Rapid Detection of Constant-Packet-Rate Flows

Kuan-Ta Chen and Jing-Kai Lou
Institute of Information Science, Academia Sinica
ktchen@iis.sinica.edu.tw kaeaura@iis.sinica.edu.tw

***Abstract*—The demand for effective VoIP and online gaming traffic management methods continues to increase for purposes such as QoS provisioning, usage accounting, and blocking VoIP calls or game connections. However, identifying such flows has become a significant administrative burden because many of the applications use proprietary signaling and transport protocols. The question of how to identify proprietary VoIP traffic has yet to be solved.**

In this paper, we propose using a deviation-based classifier to identify VoIP and gaming traffic, given that such real-time interactive services normally send out constant-packet-rate (CPR) traffic with a fixed interval, in order to maintain *real-timeliness* and *interactivity*. Our contribution is two-fold: 1) We show that scale-free variability measures are more appropriate than scale-dependent ones for quantifying the network variability injected into CPR traffic. 2) Our proposed classifier is particularly lightweight in that it only requires a few inter-packet times to make a decision. The evaluation results show that by only analyzing 10 successive inter-packet times, we can distinguish between CPR and non-CPR traffic with approximately 90% accuracy.

***Index Terms*—Online Gaming, Traffic Analysis, Traffic Classification, Traffic Identification, VoIP**

I. Introduction

VoIP and online gaming have become increasingly popular in recent years. Meanwhile, the demand for ways to manage the traffic of these applications has continued to grow for purposes such as QoS provisioning, usage accounting, and blocking certain VoIP calls or game connections in enterprises. For management purposes, the flows generated by such applications must first be *identified*.

VoIP flow identification was not a problem until recently because, in the past, most of the dominant VoIP solutions employed standard signaling protocols, such as H.323 [14] or SIP [10]. As a result, traffic from standardized VoIP systems could be easily identified using publicly-available monitoring tools [7]. However, some recent VoIP solutions, which are mostly based on peer-to-peer (P2P) technology, do not follow standardized signaling and/or transport protocols. The most well-known examples are probably Skype [1] and GoogleTalk[1] [9]. Specifically, Skype uses proprietary signaling protocols and randomly-assigned TCP/UDP port numbers. Moreover, all messages and voice data in Skype communications are encrypted. Traffic from these applications continues to cause a significant administrative burden for ISPs and enterprises, as it can bypass firewalls without audit controls. The question of how to identify proprietary VoIP traffic has yet to be solved.

Rather than provide a general flow classification framework based on flow characteristics or application signatures, this study considers a particular type of traffic, namely, *constant-packet-rate (CPR) traffic*. To maintain *real-timeliness* and *interactivity*, VoIP and some real-time network games, such as Counter-Strike [8], send out packets to interacting parties at regular intervals. In VoIP applications, the continuous human voice is encoded and streamed into packets for transmission with a typical interval ranging from 20 ms to 50 ms[2]. For real-time game playing, the updated game states at the server are sent to each game client at regular intervals in order to keep the clients' states up-to-date. These observations form the motivation of this study: *If we can detect CPR flows, then we should be able to detect VoIP and online gaming flows, as these two applications normally generate CPR traffic*.

Challenges: Intuitively, determining whether a packet stream is CPR should not be a difficult task, since the inter-packet times (IPT) would be a constant value. While this property might be correct when we observe the stream at the sender host, the IPTs become much more variable, instead of remaining constant, once the packets have been input to the network.

Assume a series of successive and equally-spaced packets have been generated by an application with a fixed interval T. The packets may incur the following types of network impairment that would weaken their constant-IPT property after they have been injected to networks :

- **Host Delay**: Because modern operating systems are normally multi-tasking, even if an application wishes to send out packets regularly, it can only issue a packet transmission request when it is served by the CPU. Consequently, the inter-packet times immediately after packets leave the sending host might be slightly different from T, depending on the CPU load at that instant.
- **Channel Delay**: To send a segment on a CSMA/CD-like network, a host must wait for a period to detect whether other stations are transmitting data simultaneously. When a transmission collision occurs, a host should wait for a random period so as to prevent further collisions. Such delays would alter the inter-packet times, especially in wireless networks, where collisions and interference are

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council of the Republic of China under the grants NSC 96-2219-E-001-001, NSC 96-2219-E-011-008, and NSC 96-2628-E-001-027-MY3.

[1]GoogleTalk actually adopts a hybrid approach comprised of a standard signaling protocol (SIP) and a proprietary transport protocol [7].

[2]Many recently developed P2P-based VoIP applications, including Skype, do not support silence suppression; that is, lowering the packet sending rate while the user is not talking. This design is deliberate, as it maintains UDP port bindings at the NAT and ensures that environmental sounds can be heard all the time.

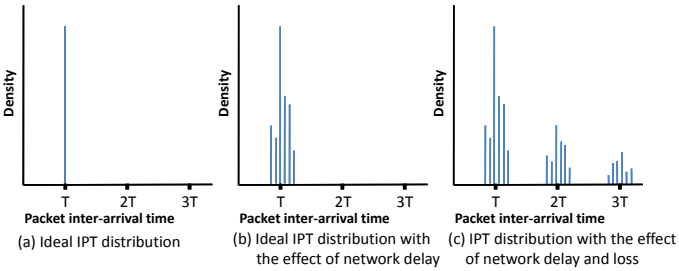

Fig. 1. The distribution of IPTs with and without disturbance caused by network impairment (a) Ideal (b) Delay introduced (c) Delay and loss introduced

much more serious than in wired networks.

- **Network Queueing**: A packet may be queued for a while when it passes through a congested link. The queueing delay depends on the congestion level of the link. Since the queueing delays incurred by different packets may not be the same, the IPTs between packets that have been queued would be altered.
- **Network Packet Loss**: A packet may be dropped when passing through a congested link if the packet queue is full, which will lead to IPTs of multiples of T. For example, assume a series of three packets, with two IPTs (T, T), are to be transmitted over the Internet. If the middle packet is dropped by a router, then the IPT between the remaining two papers will be $2T$. Similarly, the IPT value will be kT if a packet loss burst of length $(k-1)$ occurs.

The graph in Fig. 1 illustrates the effect of network delay and loss on the distribution of IPTs in CPR traffic. As shown in the figure, even though the packets leave the sender with exactly equal inter-packet times, the IPTs disturbed by network delays tend to have a wider distribution compared to the original distribution. Fig. 1(c) shows that the distribution of IPTs disturbed by both network delays and network loss would have multiple clusters centered around $T, 2T, 3T, ..., kT$, if no packet loss bursts of length longer than $k-1$ have occurred.

Contributions: The contribution of this study is three-fold. 1) We propose using a deviation-based CPR-traffic classifier to identify real-time interactive applications, specifically VoIP and online gaming. 2) We show that scale-free variability measures, such as CoV (the coefficient of variation), are more appropriate than scale-dependent metrics, such as SD (the standard deviation), for quantifying the network variability incurred by CPR traffic. 3) The proposed classifier is particularly *lightweight* in that it requires only a few inter-packet times to make a decision. Our evaluation shows that by analyzing only 10 successive inter-packet times, we can distinguish between CPR and non-CPR traffic with approximately 90% accuracy.

The remainder of this paper is organized as follows. Section II considers related works. We discuss the measurement methodology and summarize our traces in Section III. In Section IV, we describe the design of the deviation estimators and demonstrate their ability to classify CPR and non-CPR flows. In Section V, we evaluate the performance of different deviation estimators in terms of classification accuracy, and identify the sources of misclassifications. Then, in Section VI, we present our conclusions.

II. Related Work

Traffic identification was not a serious problem until recently because most Internet flows used well-known TCP/UDP ports, such as port 80 for HTTP, and port 23 for TELNET services. Thus, a simple inspection of the TCP/UDP port number field could identify the application associated with a certain flow [18]. However, the effectiveness of port-based classification has been largely reduced due to the prevalence of P2P applications, many of which use random port numbers, or even port numbers that have been assigned to other applications. For instance, HTTP port numbers (80/443) are often used by P2P applications to traverse a firewall [16]. Thus, a simple inspection of the port number field may lead to inaccurate classification.

Payload-based Classification To identify a variety of P2P applications whose protocols are often proprietary, ill-defined, and even encrypted, a payload-based classification approach has been proposed in [11, 23].

Payload-based classification relies on a unique signature carried by application packets, so the identification accuracy can be very high. Even so, the approach has some major drawbacks: 1) The packet payload may be inaccessible in a carrier network because obtaining such information would be a violation of privacy. 2) Payload pattern matching usually relies on the flexibility of regular expressions; however, software implementations of regular expression matching incur a heavy computational load, while hardware implementations may restrict the expression flexibility and the state number of the transformed DFA (deterministic finite automata). 3) A proprietary application's signature may be subject to change at any time, so payload-based classifiers require frequent updates of the signature database to detect newly released software.

Flow-Information-based Classification Another traffic classification approach is based on a summary of flow statistics, such as the flow duration, number of packets, flow inter-arrival times, and packet inter-arrival times. Machine learning [15] or statistical clustering techniques [17, 19, 22] are usually employed to process flow information for classification purposes. Among the numerous studies, [15] extracts representative features from bulk data flows, such as HTTP, FTP, and SMTP, to create a workload model for network simulators. In [22], traffic is classified into three classes, namely, bulk data transfer (e.g., FTP), conversational communications (e.g., TELNET), and streaming traffic, to provide different levels of QoS for each class. Meanwhile, [19] uses a reference pattern approach, which designates the packet size and inter-packet time as flow features, to identify VoIP flows for management purposes and QoS provisioning.

A number of studies do not belong to either of the above categories. The approach proposed in [12] identifies P2P traffic based on the transport layer characteristics, as P2P applications usually exhibit distinct connection patterns; for example, they create both TCP and UDP connections using the same port number, and establish several connections simultaneously. BLINC [13] considers the association between Internet hosts

TABLE I
SUMMARY OF THE COLLECTED TRAFFIC TRACES

Trace	Flow #	Packet Rate	IPT CoV	Path Diversity	Source
VoIP	1739	(20, 33, 33) pkt/sec	0.37	1106 hosts / 1641 paths	National Taiwan University
Counter-Strike	1016	(11, 22, 26) pkt/sec	0.32	271 hosts / 270 paths	mshmro.com [3]
TELNET	276	(3, 12, 89) pkt/sec	1.53	140 hosts / 93 paths	National Taiwan University
HTTP	409	(3, 13, 117) pkt/sec	1.54	474 hosts / 325 paths	WIDE backbone [6]
P2P	1303	(3, 7, 29) pkt/sec	1.63	645 hosts / 644 paths	Academia Sinica
World of Warcraft	1611	(4, 5, 10) pkt/sec	0.71	52 hosts / 39 paths	National Taiwan University

† The (.05, .50, .95) quantiles of the observed packet rates in one second.

and the applications running on them. Instead of inspecting individual flows, it looks at all the flows generated by specific hosts. BLINC is therefore able to accurately associate hosts with the services they provide or use.

The present study differs from the above works in the following respects: 1) Rather than provide a general framework to classify all Internet traffic, we identify a specific kind of traffic—constant-packet-rate (CPR) traffic, which is usually associated with real-time interactive applications, notably VoIP and fast-paced online games. 2) Our approach only operates on network-level information and uses a) the five-tuples to identify a flow, and b) the intervals between a few successive packets, without transport-level information, such as port numbers, even application-level information. While our approach cannot classify network flows into specific applications, it can efficiently identify CPR flows for management and QoS provisioning purposes. Moreover, it can also be used as a lightweight front line for an application-specific flow classifier to reduce the processing overhead.

III. TRACE COLLECTION

We evaluate the proposed deviation-based classifiers for CPR and non-CPR flows based on Internet traffic traces. As packet inter-spacing times are highly sensitive to the capacity, router configurations, and varying cross-traffic along a network path, the trace collection was conducted in a way that includes as much path diversity as possible.

We select six network applications, two of which are CPR-based and four are non-CPR-based. Skype [2], a popular VoIP software, and Counter-Strike [8], a popular first-person shooting game, are selected as representative of CPR applications. Skype is known to use either iLBC and iSAC codec depending on the network conditions [4]. The iSAC codec may use different encoding bit rates and packetization frequencies depending on the host's CPU usage and network congestion levels; however, as the adaptation of the packet rate is infrequent, Skype traffic can be seen as CPR in a short time-scale. On the other hand, we select TELNET, HTTP, P2P, and World of Warcraft, a popular MMORPG (Massively Multiplayer Online Role-Playing Game) as representative of non-CPR applications.

The collection procedures for specific traces were as follows. 1) Skype traffic was captured according to the procedures detailed in [4]. 2) TELNET traffic was captured on a gateway router for all TCP flows using port 22 (SSH) and port 23 (telnet); all intra-campus traffic was removed. 3) World of Warcraft traffic was captured on a gateway router for all TCP flows with port number 3274 and either the source or destination address within the network 203.66 (where the World of Warcraft server in Taiwan resides). 4) P2P traffic was captured on a dedicated PC running BitComet, a variant of BitTorrent client [21]. As the BitTorrent protocol does not use a fixed port number, we only recorded flows that used port numbers higher than 1024.

The collected traffic traces are summarized in Table I. To ensure there were sufficient packet samples in each flow, we removed flows containing less than $2,000$ packets. The IPT CoV field denotes the overall coefficient of variation of inter-packet times in each trace (see Section IV-B1). It indicates that CPR applications have lower IPT variability than non-CPR applications. The path diversity field summarizes the number of hosts and host-pairs involved in each trace. It shows that most of our traces contained a considerable number of network paths ($> 1,000$). The degree of path diversity is especially important to our study, as the network dynamics is one of the key factors that can make CPR traffic less distinguishable from non-CPR traffic. In view of this factor, we believe that, to a certain degree, our traces are appropriate for classifying CPR and non-CPR traffic in terms of traffic variability inherited from the Internet dynamics.

IV. DEVIATION-BASED CLASSIFICATION

In this section, we discuss how we classify CPR and non-CPR traffic based on their *short-term deviation measures*. By definition, CPR traffic should be easily identifiable, since its inter-packet times (IPTs) are perfectly constant. However, CPR traffic may inherit variability, such as delay and loss, from the network and applications (in the form of dispersed IPTs), and become more similar to non-CPR traffic.

To assess whether IPT variability can be an effective indicator of CPR traffic, we first analyze the IPT distributions of different applications, which can be inferred from their respective long-term packet arrival processes. Then, we discuss the design of a robust estimator to infer IPT deviations from a short packet arrival series. We conclude this section with a graphical presentation to illustrate how the derived short-term IPT deviations are used to classify CPR and non-CPR traffic.

A. Inter-Packet Time Distribution

To identify discrepancies in the distribution of IPTs in different applications, we plot the histogram of the IPTs from each trace in Fig. 2. The IPT histogram provides an intuitive way to observe the *long-term deviation* of packet inter-arrival

times, on a time scale roughly equal to the duration of the flows. The histograms in Fig. 2 can be interpreted as follows. If the long-term IPT distribution of an application is heavily concentrated within a small range, it is reasonable to expect that the application's *short-term IPT deviations* will also be small. However, the opposite is not always true. If the long-term IPTs are widely distributed, then the corresponding short-term IPT variation might be small, moderate, or large—depending on *how the IPTs are temporally auto-correlated*. For example, if IPTs are positively auto-correlated, indicating that a small (large) IPT is likely to be followed by a small (large) IPT, then their small-scale burstiness tends to be small. In contrast, if IPTs are negatively auto-correlated, which implies that small and large IPTs are apt to occur alternately, then the short-term IPT deviations tend to be large.

As one might expect, the IPTs of CPR applications, VoIP and Counter-Strike in Fig. 2, are heavily clustered with a mode smaller than 100 ms. Moreover, most of the IPTs are not far from the cluster's center. Non-CPR applications—HTTP, P2P, TELNET, and World of Warcraft—also have a considerable number of IPTs clustered around a small value (at scales of 10 ms or smaller). They also have IPTs widely spread over the range of the x-axis, which clearly manifests the non-CPR nature of their traffic patterns. Interestingly, the IPT distributions associated with non-CPR traces are all multi-modal. We consider that the 200 ms peaks are due to TCP's Nagle algorithm and delayed-ack mechanism [24], while the remaining peaks are probably due to the application's design. In particular, we find that the pronounced peaks at 200 ms and 500 ms in the World of Warcraft trace occur because the game exchanges data with regular time intervals. This design is common in network games for maintaining the state consistency between peers [5].

We also provide the coefficient of variation (CoV) of the IPTs above each figure. The value of the CoV provides a scale-independent way to quantify the dispersion of IPTs. Clearly, the IPT CoV for the VoIP and Counter-Strike traces (< 0.7) is significantly smaller than those of non-CPR applications (> 1.5), while the IPT CoV of the World of Warcraft trace is approximately in-between the two extremes. It seems that a simple variability measure like the CoV may already be a useful indicator of whether a flow is CPR or not. Even so, whether IPT deviations inferred from short-term IPT series, rather than long-term IPT series, can achieve similar classification accuracy remains a question. In the rest of this paper, we attempt to answer this question.

B. Estimating Short-Term IPT Deviation

In Fig. 2, the long-term IPT deviation seems to be a decisive feature for distinguishing CPR flows from non-CPR flows, as the former exhibit a much smaller degree of IPT dispersion. However, in practice, IPT deviations must be computed based on *short-term observation* of packet arrivals for the following reasons:

- Many applications, such as usage accounting and traffic blocking, require the classifier to make a very quick decision (e.g., in sub-seconds).

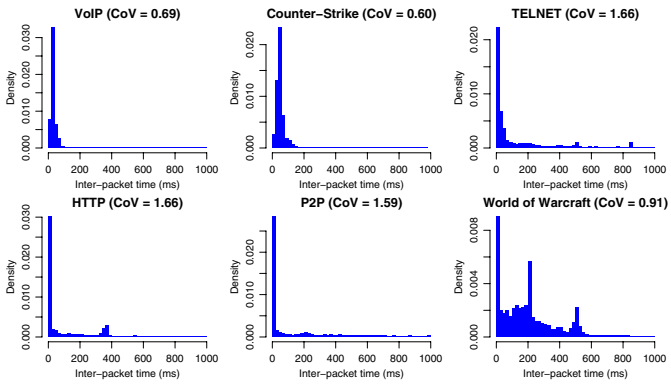

Fig. 2. IPT distribution of different applications

- The classifier must keep a flow's state in its memory before a decision can be made. Thus, the faster the detection, the less time a flow's state needs to be kept. Consequently, more flows can be served simultaneously with the same amount of memory.

Therefore, even if the dispersion of a long IPT sequence can be used to accurately identify whether a flow is CPR or not, we have to limit the length of an IPT series so that we can obtain a quick decision. The difficulty is that, while long IPT sequences are primarily governed by an application's traffic patterns, a short IPT series is less robust and less reliable for inferring the "true" traffic patterns (such as the variability) of an application. This is because the amount of traffic characteristics captured by short packet sequences depends on the sampling time and varying network conditions. In other words, short-term IPT variability, though inherited from the application traffic variability, might be dominated by *sampling randomness* and *network dynamics*.

Because of the detection time and unavoidable randomness, we need an estimator for short-term IPT deviations that has the following properties:

1) It should be able to cope with the *randomness* introduced by sampling and network dynamics.
2) It should achieve *high discriminability* between CPR and non-CPR traffic.
3) It should incur the lowest possible *storage and computation overhead*.

More specifically, the desired estimator should obtain low deviation measures for CPR traffic, and high deviation measures for non-CPR traffic, regardless of the inevitable randomness introduced in packet arrival sequences.

In the following, we discuss three aspects of designing an IPT deviation estimator, namely, the deviation metric, the sample size w, and the smoother size s. We list a number of possible designs with regard to each aspect, and the reasons why they deserve to be considered. Hereafter, we use "IPT deviation(s)" to denote short-term IPT deviation(s).

1) Deviation Metric: In statistics, there are numerous measures for quantifying the degree of spread, or variability, in a set of numbers because different metrics usually emphasize different portions of "variability" in the data. For example, some metrics, such as the median absolute deviation (MAD),

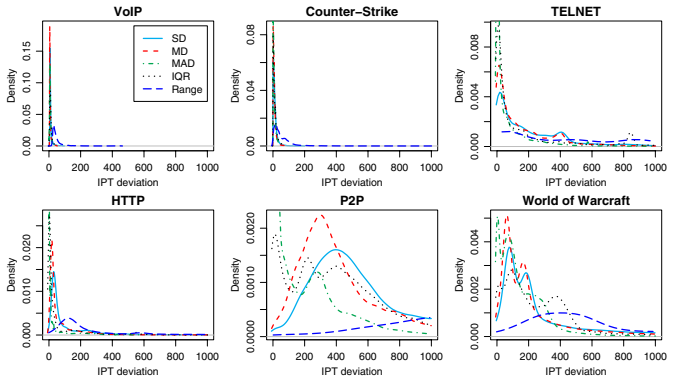

Fig. 3. IPT deviation distributions computed by different variability metrics

put more weight on the data around the center, while others, such as the range, capture the degree of spread of extreme values without considering the remaining data.

Without a priori knowledge about the variability components should we emphasis, i.e., whether the center or the tail of the IPT distribution yields a more representative measure? We list the candidate metrics below:

- **Standard deviation (SD)**: This is one of the most widely used variability estimators. It is the square root of the variance, which is approximately the average of the squared distance from the mean. Squaring the distance from the mean indicates that more weight is given to data farther from the center; thus, the SD can be greatly affected by the tail behavior.
- **Coefficient of variation (CoV)**: The CoV is defined as the ratio of the standard deviation to the mean. For this reason it is a *dimensionless* number that allows comparison between data sets with different scales.
- **Mean absolute deviation (MD)**: This measure is like the standard deviation, but it does not square the distance to the mean; hence, it is less affected by extreme values than the standard deviation.
- **Median absolute deviation (MAD)**: This measure is similar to the mean absolute deviation, except that the central tendency is computed by the median instead of the mean. Because the median is a more robust estimator of the center, the tail data have less influence on the calculation of the MAD than they have on MD.
- **Inter-quantile range (IQR)**: This measure is the value of the 75th percentile minus the value of 25th percentile. As the values in the first and fourth quantiles are not considered at all, this estimator only measures the variability of data near the center.
- **Range**: This measure is only based on the lowest and highest extreme values in the data set. In contrast to the IQR, the spread near the center of the data is not captured at all; thus, it is obviously very sensitive to extreme values.

To demonstrate the difference between the listed metrics, we divide the IPT series of each flow into a non-overlapping 30-IPT sub-series, and take a deviation measure from each sub-series. The distributions of deviation estimates based on different metrics are plotted in Fig. 3. Because of the difficulty of displaying different metrics on the same scale at the same time, the result of the CoV metric is not included. We can see from the figure that, overall, the deviation estimates of VoIP and Counter-Strike are generally small and concentrated around a certain value. Those of TELNET and HTTP have a wider distribution and a mode of slightly larger magnitude. Meanwhile, P2P and World of Warcraft have very diverse IPT deviation estimates and large modes, whose magnitude is dependent on the metric used.

In terms of the deviation metrics, MAD exhibits the most concentrated distribution of all the measures, which demonstrates its conservative nature in dispersion estimation. The sensitivity of the deviation measures to extreme values can also observed in the magnitude of the deviation distributions of the P2P trace, in that smaller deviation estimates generally indicate a metric's insensitivity to the tail. According to the figure, the susceptibility of the metrics to tail values is ordered as follows:

$$\text{Range} > \text{SD} > \text{MD} > \text{IQR} > \text{MAD},$$

which is consistent with the statistical meaning of each metric.

The different characteristics of the deviation metrics can also be observed in the plot of the World of Warcraft trace, in which the distributions of IQR and MAD estimates are bi-modal, while those of the remaining metrics are uni-modal. If an IPT sample (30 IPTs in this case) includes IPTs from both modes, the metrics that take account of tail data (such as the SD) would yield moderate dispersion measures. Meanwhile, robust estimators (such as the IQR and MAD) would yield either small or large dispersion measures depending on the exact composition of IPTs in the respective clusters.

2) Sample Size: To efficiently determine a flow's type, i.e., CPR or non-CPR, and reduce the memory required to store flow states, samples that contain only a small number of IPTs are preferred. However, small samples also lead to less reliable estimation of IPT deviations. Thus, the decision about sample size is a trade-off between classification accuracy and time/space complexity for CPR flow detection.

We find that a small sample containing only a few IPTs may not be able to capture an application's traffic patterns. That is, even for a highly variable traffic stream, there might be a non-insignificant probability that a few consecutive packets will be released with nearly constant intervals. In addition, the traffic generated by many applications usually exhibits *positive auto-correlations* to a certain degree. A positively auto-correlated packet stream possesses a characteristic whereby the IPT deviation of successive packets is generally small if the time span of packet sampling is shorter than the time scale at which the auto-correlation effect diminishes. This feature may considerably reduce non-CPR flows' traffic variability estimates and make those flows less distinguishable from CPR flows.

To obtain robust estimates of IPT variability in a short time, we consider sample sizes ranging from 3 to 30 IPTs. The maximum sample size of 30 IPTs is chosen simply because it takes approximately one second to observe 30 packets for VoIP traffic, as the VoIP packet rate normally spans 15–50

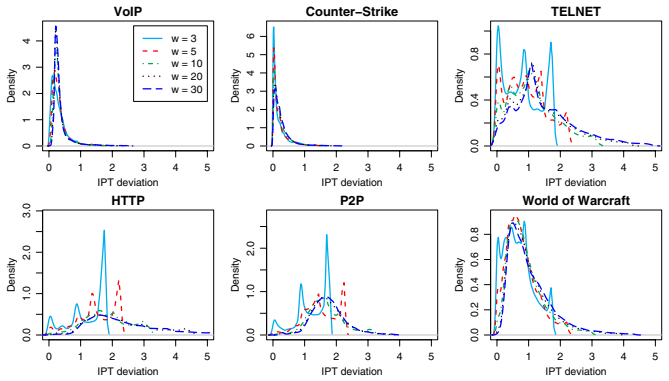

Fig. 4. IPT deviation distributions computed with various sample sizes

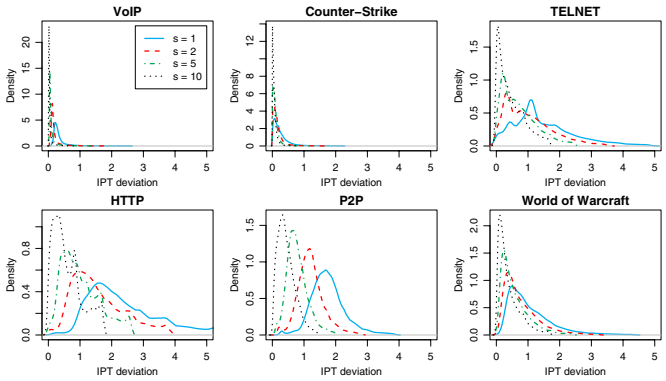

Fig. 5. IPT deviation distributions computed with various smoother sizes

packets per second [4]. With this setting, the flow states have to be kept in memory for about one second, which may incur a high overhead for a classifier under a heavy traffic load. For example, in a scenario where an average of ten thousand flows are established in one second, each flow requires 100 bytes for state maintenance at an average rate of 15 packets per second. Hence, at least a 2-MB SRAM is required to maintain the states for that have yet to be classsified flows.

In Fig. 4, we plot the distributions of IPT deviations derived from samples of different size. We make two observations from the figure. First, the IPT deviations captured by small samples (e.g., $w < 10$) are not robust enough and may differ drastically depending on the particular IPTs observed. This explains why IPT deviation distributions for small samples show multiple peaks for non-CPR traces. Second, although large samples have more consistent deviation estimates, they generally capture more variability and lead to higher estimates (identified by modes of larger magnitude). This is because longer packet sequences are more likely to include irregular packet arrivals and occasional randomness introduced by network dynamics.

3) Smoother Size: The sample size provides a means of restricting the time scale at which deviation estimates are taken; however, the granularty of deviations is always measured at the scale of packet inter-spacing times.

Consider a scenario where the application, host, and network have introduced considerable randomness into a CPR packet stream. As a result, the stream's IPT deviations become large, and thus not easily distinguishable from non-CPR traffic. In this case, we can apply a smoother to *remove short-term IPT fluctuations* before calculating the deviation estimates. For an IPT sample of size w[3], $(ipt_1, ipt_2, ..., ipt_w)$ and smoother size s, the smoothed IPT series of size w/s is computed as $(\sum_{i=1}^{s} ipt_i/s, \sum_{i=s+1}^{2s} ipt_i/s, ..., \sum_{i=w-s+1}^{w} ipt_i/s)$. With an appropriate smoother of size s, we can remove IPT fluctuations in time scales smaller than $s \times mean(ipt)$. The deviation estimates are then derived from this smoothed series.

We apply smoothers with sizes ranging from 2 to 10 IPTs and compute the IPT deviation estimates using the CoV metric. As shown in Fig. 5, the larger the smoother size, the smaller the deviation estimates we obtain. On the one hand, we expect the smoothers to remove undesirable fluctuations injected into CPR packet streams. On the other hand, smoothers that are too large may unnecessarily remove the intrinsic variability of non-CPR traffic. Thus, choosing the smoother size is also an important issue in designing an appropriate IPT deviation estimator.

[3]For simplicity and without loss of generality, we assume that $w = k \times s, k \in \mathbb{N}$.

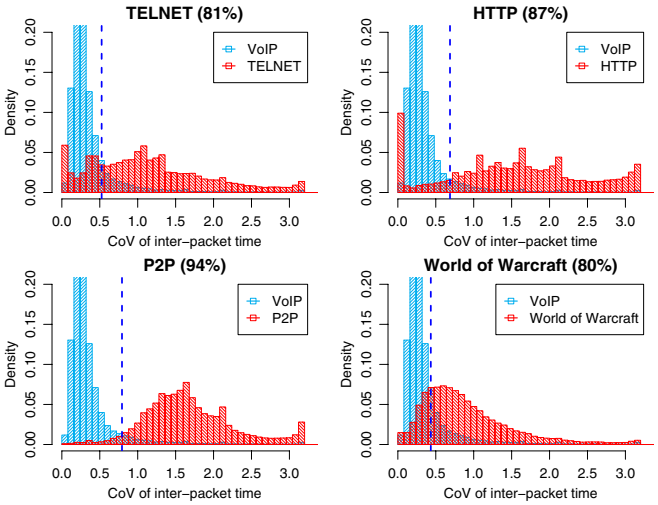

Fig. 6. Differences between our VoIP traffic and non-CPR traffic traces based on deviation estimates (CoV, $w = 10$ and $s = 1$)

C. Graphical Demonstration

Having discussed several aspects of designing an estimator for short-term deviations of IPTs, we now use a graphical demonstration to illustrate how the deviation estimates are used for the classification of CPR and non-CPR traffic.

In Fig. 6, we plot the IPT deviation distributions for VoIP and each non-CPR traffic trace, where the deviation estimator used is comprised of the CoV metric, a 10-IPT sample size ($w = 10$), and no smoothing ($s = 1$). We can see from the graphs that VoIP traffic clearly has very different IPT deviation distributions to those of non-CPR traffic. The vertical dashed line marks the threshold value that has the most discriminative power. The number following each trace name denotes the discrimination accuracy, which is computed as the number of

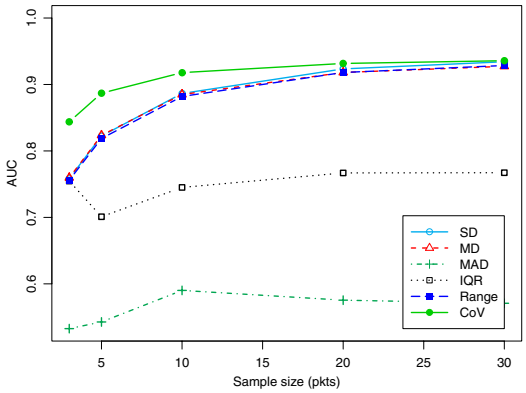

Fig. 7. Flow classification performance with different deviation metrics and sample sizes

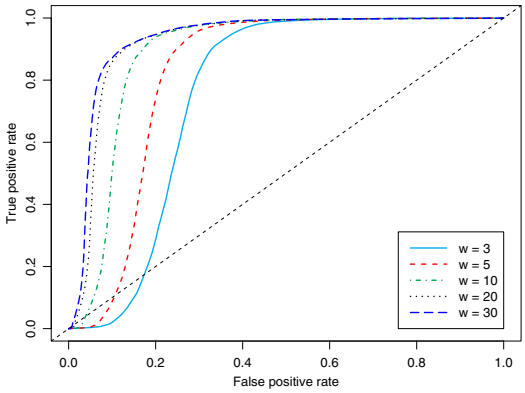

Fig. 8. ROC curves based on various sample sizes (CoV, $s = 1$

correct classifications (both false positives and false negatives) divided by the number of samples.

This demonstration shows that even a small 10-IPT sample can be used to classify CPR and non-CPR flows with an accuracy higher than 80%. The accuracy can be boosted to nearly 90% if a sample size of 30 IPTs is used. Although this suggests the feasibility of using short-term-deviation-based classification for CPR and non-CPR traffic, we still need to address the following questions: 1) "What is the limit of this classification approach?" 2) "Why and how do misclassifications occur?" We consider these issues in the next section.

V. Performance Evaluation

In this section, we evaluate the classification performance of a variety of IPT deviation estimators. Our purpose is twofold: 1) to understand the performance and limits of IPT-deviation-based classifiers; and 2) to identify the estimator that possesses the most robust discriminative power.

We begin by evaluating the effect of the deviation metric and sample size on an estimator's ability to classify CPR and non-CPR flows, after which we consider the effect of the smoother size. Then, we analyze the sources of misclassifications by breaking the evaluation into a per-application analysis. For simplicity, we use "CPR discriminability" to denote a deviation estimator's ability to discriminate between CPR and non-CPR traffic.

Performance Metric In this section, we use AUC (Area Under the Curve) to compare the classification performance of deviation estimators. AUC is the area under the ROC (Receiver Operating Characteristic) curve, which is is formed by the ratio of true positives (sensitivity) and the ratio of false negatives ($1-$specificity) over the entire range of possible cutpoints. The AUC is generally referred to as the discrimination ability of a classifier. As a rule of thumb, an AUC value higher than 0.8 indicates generally good discrimination, while a value of 0.5 is equivalent to a random guess.

A. Effect of Deviation Metric

Fig. 7 shows the effect of the deviation metric and sample size on CPR discriminability. The result indicates that more robust deviation metrics generally lead to worse discrimination, which is exemplified by the fact that MAD and IQR yield much lower discrimination values than the other metrics. This behavior implies that non-CPR traffic is undistinguishable from CPR traffic, as its intrinsic variability is underestimated by robust variability measures. However, the discrimination ability is not always higher for metrics that are more sensitive to tails, as the most sensitive metric, i.e., the range, does not yield the best performance. Generally speaking, the SD, MD, and range metrics have approximately the discriminability, although the SD performs slightly better than the other two.

The only dimensionless metric, CoV, performs better than its scale-dependent variant, SD, especially when the sample size is small. As the numerator of the CoV metric is equal to that of SD, so the discrepancy in performance must be due to the denominator of CoV, i.e., the average IPT. Our analysis shows that, for non-CPR traffic, the average IPT can be very small depending on the composition of IPTs observed (see Fig. 2), which leads to high CoV estimates. For example, while the SDs of series $(10, 20, 30)$ and $(30, 40, 50)$ are identical (both are 10), the CoVs of the two series have a ratio of 2:1 (0.5 and 0.25 respectively). With the occurrence of small average IPTs, the high deviation estimates of non-CPR flows make them more distinguishable from CPR flows. This effect diminishes with the increase in sample size, as the average IPT is less likely to be extremely small; in fact, it tends to converge to the IPT population mean.

B. Effect of Sample Size

With regard to the sample size, as shown in Fig. 7, larger samples consistently lead to better classification efficiency, except for the two most robust deviation metrics, MAD and IQR. However, the increase in CPR discrimination ability with larger samples is *sub-linear*; in other words, the marginal return decreases as the sample size increases. Thus, in view of the classification accuracy and time/space overhead required, it is not easy to choose the best sample size. In fact, the choice usually depends on the particular context. We consider that a 10-IPT sample size is generally a good choice because it achieves reasonable discriminability and does not require much time to collect IPT series.

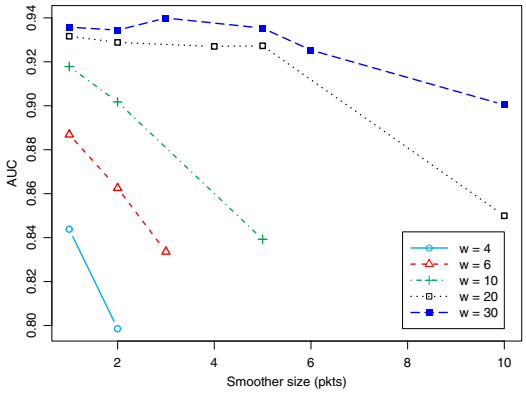

Fig. 9. Flow classification performance with different sample sizes and smoother sizes

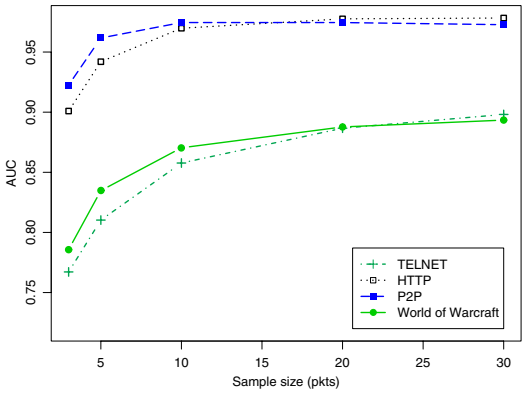

Fig. 10. Flow classification performance for CPR applications and individual non-CPR applications

In Fig. 8, we plot the ROC curves that correspond to the deviation estimators comprised of the CoV metric and samples of different size to evaluate their classification performance. In an ROC plot, the top-left area denotes higher true positive rates and lower false positive rates. We can see that the curves shift toward the top-left region as the sample size increases, along with a decrease in marginal utility. Generally, the true positive rate increases much faster than the false positive rate, which is an indication of a good classifier. The curves also reveal that the false positive rates are non-insignificant (10% for $w = 10$, and 5% for $w = 30$) when the true positive rate reaches 90%, even when the sample size is large. This indicates that non-CPR traffic may comprise CPR-like packet sequences that mislead the classifiers. We address this issue and determine the source of the false positive classifications in Section V-D.

C. Effect of Smoother Size

We now investigate the effect of the smoother size on CPR discriminability. Fig. 9 shows the classification efficiency of deviation estimators comprised of the CoV metric and smoothers of different size. The CoV metric is chosen for its generally good discrimination performance.

The result shows that, although smoothers may improve the classification performance under certain conditions, the increase is marginal. There is only an improvement when the sample size is large and the smoother size is not comparable to the sample size. One possible explanation of this phenomenon is that, while a single IPT may be affected by large disturbances from the application or network, the average IPTs across a few packets might be better able to capture the original application's traffic variability. However, smoothers reduce the AUC value when the smoother size is comparable to the sample size. As smoothers are unable to significantly improve classification efficiency and may even reduce the accuracy, we conclude that IPT smoothing prior to taking deviation estimates is not helpful.

D. Error Analysis

Fig. 7 shows that the AUC does not increase unboundedly, but converges at approximately 0.93 as the sample size increases. To determine the sources of misclassifications and understand why they occur, we evaluate the discrimination performance of CPR traffic and individual non-CPR traffic streams, as shown in Fig. 10.

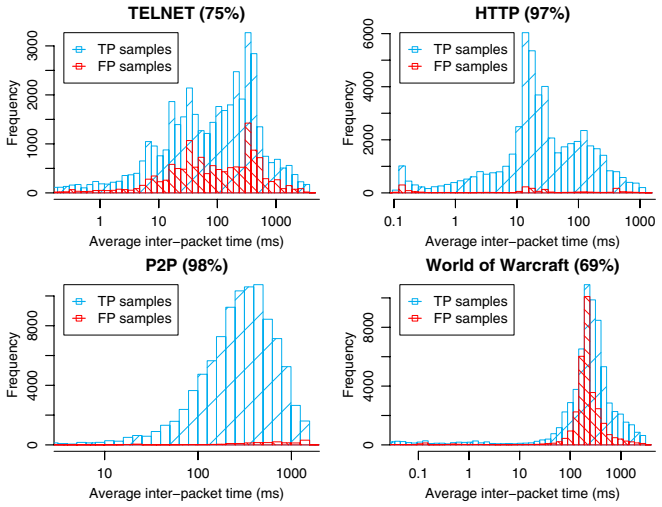

Fig. 11. Flow classification error analysis for CoV deviation estimator

The result for non-CPR applications can be divided into two groups. TELNET and World of Warcraft traffic is not easily distinguishable from VoIP traffic (AUC ≈ 0.85 with $w = 10$), while HTTP and P2P traffic is much more distinct from CPR traffic (AUC > 0.95 with $w = 10$). We note that the applications with worst classification performance are both *interactive*, while the remaining two are both *bulk transfer* applications.

What makes TELNET and World of Warcraft traffic less distinguishable from CPR traffic? To determine whether particular traffic patterns affect the classification accuracy, the histograms of the average IPTs of correctly- and mis-classified samples for each non-CPR trace are plotted in Fig. 11. The graph reveals that the false positive samples (those misclassified as CPR) for TELNET and World of Warcraft traces have no particular relationship with average IPTs, as the average IPTs of true-positive and false-positive samples have approximately identical distributions. One possible explanation for the poor discrimination for these two traces is the *appli-*

cation behavior. For TELNET traffic, because the packets are initiated by humans, whose typing activities can be modeled as exponential inter-packet times with a certain degree of auto-correlation [20], it is reasonable to find IPT samples with low burstiness, especially when the sample size is small. For World of Warcraft traffic, we find that the game peers occasionally exchange messages with 200 ms or 500 ms intervals (see Fig. 2); thus, if an IPT sample includes some of the regularly-released packets, it will probably be classified as CPR.

VI. Conclusion and Future Work

In this paper, we have proposed using inter-packet times to detect VoIP and online gaming traffic based on their distinct traffic characteristics—*constant inter-packet times*. We have shown that CPR traffic, after inheriting the randomness from the application and the network dynamics, may no longer possess the *constant-IPT property*. We have also examined various issues in the design of IPT deviation estimators for classifying CPR and non-CPR traffic. The performance evaluation shows that the dimensionless variability measure, CoV, achieves the best performance in terms of discrimination efficiency; a sample size of 10 IPTs can yield a reasonable classification accuracy ($\approx 90\%$).

Although the deviation-based CPR-traffic detector discussed in this paper achieves a reasonable performance, it generates a not insignificant number of misclassifications. Two reasons for the misclassifications are an *application's nature* and a *small sample size*. How to improve classification accuracy without increasing the sample size is an issue we will address in our future research. In addition, for packet streams passing through a lossy path (very congested Internet links or noisy wireless links), the IPT variability for CPR traffic increases substantially due to excessive packet loss. Moreover, for flow classification on a high speed link, a certain degree of *packet sampling* would be beneficial to reduce the processing overhead. In our future work, we will further improve the deviation-based classifier, so that it is *robust against both intentional or unintentional packet loss events*.

References

[1] S. A. Baset and H. Schulzrinne, "An analysis of the Skype peer-to-peer internet telephony protocol," in *Proceedings of IEEE INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[2] D. Bergström, "An analysis of skype voip application for use in a corporate environment," http://www.geocities.com/bergstromdennis/, 2004.

[3] W. chang Feng, "mshmro.com counter-strike traffic trace," http://www.thefengs.com/wuchang/work/cstrike/, 2004.

[4] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying Skype user satisfaction," in *Proceedings of ACM SIGCOMM 2006*, Pisa, Itlay, Sep 2006.

[5] K.-T. Chen, P. Huang, and C.-L. Lei, "Game traffic analysis: An MMORPG perspective," *Computer Networks*, vol. 50, no. 16, pp. 3002–3023, 2006.

[6] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the wide project," in *Proceedings of USENIX*, 2000, pp. 263–270.

[7] L. Deri, "Open source voip traffic monitoring," in *Proceedings of System Administration and Network Engineering Conference (SANE)*, May 2006.

[8] W. C. Feng, F. Chang, W. C. Feng, and J. Walpole, "A traffic characterization of popular on-line games," *IEEE/ACM Transactions on Networking*, vol. 13, no. 3, pp. 488–500, June 2005.

[9] Google Inc., "Google Talk," http://talk.google.com/, 2005.

[10] M. J. Handley, H. Schulzrinne, E. M. Schooler, and J. Rosenberg, "SIP: Session initiation protocol," Internet proposed standard RFC 2543, Mar. 1999.

[11] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy, and M. Faloutsos, "Is P2P dying or just hiding?" in *Proceedings of the GLOBECOM 2004 Conference*. Dallas, Texas: IEEE Computer Society Press, Nov. 2004.

[12] T. Karagiannis, A. Broido, M. Faloutsos, and K. C. Claffy, "Transport layer identification of P2P traffic," in *Internet Measurment Conference*, A. Lombardo and J. F. Kurose, Eds. ACM, 2004, pp. 121–134.

[13] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *SIGCOMM*, R. Guérin, R. Govindan, and G. Minshall, Eds. ACM, 2005, pp. 229–240.

[14] H. Liu and P. Mouchtaris, "Voice over ip signaling: H.323 and beyond," vol. 38. Communications Magazine, IEEE, Oct 2000, pp. 142–148.

[15] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *PAM*, ser. Lecture Notes in Computer Science, C. Barakat and I. Pratt, Eds., vol. 3015. Springer, 2004, pp. 205–214.

[16] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *PAM*, ser. Lecture Notes in Computer Science, C. Dovrolis, Ed., vol. 3431. Springer, 2005, pp. 41–54.

[17] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS*, D. L. Eager, C. L. Williamson, S. C. Borst, and J. C. S. Lui, Eds. ACM, 2005, pp. 50–60.

[18] D. Moore, K. Keys, R. Koga, E. Lagache, and K. C. Claffy, "The coralreef software suite as a tool for system and network administrators," in *LISA*. USENIX, 2001, pp. 133–144.

[19] T. Okabe, T. Kitamura, and T. Shizuno, "astatistical traffic identification method based on flow-level behavior for fair voip service." VoIP Management and Security, 2006. 1st IEEE Workshop on, apr 2006, pp. 35–40.

[20] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.

[21] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004, pp. 367–378.

[22] M. Roughan, S. Sen, O. Spatscheck, and N. G. Duffield, "Class-of-service mapping for qoS: a statistical signature-based approach to IP traffic classification," in *Internet Measurment Conference*, A. Lombardo and J. F. Kurose, Eds. ACM, 2004, pp. 135–148.

[23] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," Jan. 01 2004.

[24] R. W. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.