Detecting VoIP Traffic Based on Human Conversation Patterns[*]

Chen-Chi Wu[1], Kuan-Ta Chen[2], Yu-Chun Chang[1], and Chin-Laung Lei[1]

[1] Department of Electrical Engineering, National Taiwan University
[2] Institute of Information Science, Academia Sinica
{bipa,congo}@fractal.ee.ntu.edu.tw, ktchen@iis.sinica.edu.tw,
lei@cc.ee.ntu.edu.tw

Abstract. Owing to the enormous growth of VoIP applications, an effective means of identifying VoIP is now essential for managing a number of network traffic issues, such as reserving bandwidth for VoIP traffic, assigning high priority for VoIP flows, or blocking VoIP calls to certain destinations. Because the protocols, port numbers, and codecs used by VoIP services are shifting toward proprietary, encrypted, and dynamic methods, traditional VoIP identification approaches, including port- and payload-based schemes, are now less effective. Developing a traffic identification scheme that can work for general VoIP flows is therefore of paramount importance.

In this paper, we propose *a VoIP flow identification scheme based on the unique interaction pattern of human conversations.* Our scheme is particularly useful for two reasons: 1) flow detection relies on human conversations rather than packet timing; thus, it is resistant to network variability; and 2) detection is based on a short sequence of voice activities rather than the whole packet stream. Hence, the scheme can operate as a traffic management module to provide QoS guarantees or block VoIP calls in real time. The performance evaluation, which is based on extensive real-life traffic traces, shows that the proposed method achieves an identification accuracy of 95% in the first 4 seconds of the detection period and 97% in 11 seconds.

Key words: Human Speech, Internet Measurement, Markov Model, Skype, Traffic Classification, Voice Activity Detection

1 Introduction

VoIP is becoming increasingly popular because it provides low call costs and the voice quality is comparable to that of traditional toll telephones. The trend is exemplified by the fact that one of the most widely used VoIP applications,

[*] 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.

Skype, has 246 million registrars and 100-million online users[1]. Because of the steady growth of VoIP usage, providing reliable service and satisfactory voice quality is now a high-priority for Internet and VoIP service providers.

In order to provide a dependable means of voice transmission over the Internet, it is essential that network gateways have the ability to differentiate VoIP flows from flows generated by other applications. By identifying VoIP flows, network gateways can provide QoS features, such as allocating more bandwidth or assigning a high priority to identified voice flows. Traffic management is another important application of flow classification. Enterprises often have to manage VoIP traffic in line with institutional policies, such as restricting calls to certain destinations, or blocking calls at certain times. Given these emerging needs, developing an efficient and accurate identification algorithm for VoIP flows is now of paramount importance.

To accurately identify VoIP flows in real time, we may face the following challenges:

1. **Non-standard protocols and ports.** Different VoIP applications may use different signaling protocols, such as SIP [1], H.323 [2], and several other proprietary P2P protocols [3, 4]. Additionally, many VoIP applications are peer-to-peer based and may use random port numbers rather than a fixed port number. Thus, it is difficult to detect VoIP flows by analyzing their signaling protocols.
2. **Non-standard codecs.** Modern VoIP applications may use different audio codecs to adapt to different network environments, e.g., using a wide-band codec when a broadband medium is used and a narrowband codec if a wireless connection is detected. Given the numerous audio codecs available, detecting VoIP traffic by the signatures of audio codecs is not practical.
3. **Payload encryption.** VoIP applications now tend to encrypt their packet payloads to protect privacy. Even if a VoIP application does not encrypt its packets, the packet payload may still be inaccessible because trying to obtain such information would be a violation of privacy. Thus, payload-based identification is ineffective.
4. **Silence suppression.** A large number of traffic classification methods rely on traffic patterns, e.g., the mean and variation of packet interarrival times. However, the pattern of VoIP traffic can vary a great deal over time because many audio encoders support silence suppression. In other words, they suppress a packet stream when speech is absent and resume data delivery when speech is detected. Therefore, schemes based on traffic patterns are not very effective for identifying silence-suppression-enabled VoIP flows.

In this paper, we propose *a VoIP flow identification scheme based on human conversation patterns*. We consider that the conversation pattern of VoIP traffic is *unique* compared to that of other applications. For example, a file transfer session comprises a group of unidirectional traffic flows that are very different

[1] http://seekingalpha.com/article/50328-ebay-watch-59-earnings-growth-skype-reaches-10-million-concurrent-users

from VoIP traffic, which normally comprises highly interactive speech bursts. Because the presence of speech is decided solely by the speakers and the interaction context, the conversation pattern remains constant regardless of the network dynamics. Therefore, a VoIP flow identification scheme based on human conversation patterns would be robust to *silence suppression* and *traffic dynamics* due to congestion control and packet retransmissions.

When two parties, A and B, are talking to each other, we can model their interaction by a process of four states: A talking, B talking, double-talking, and mutual silence. By so doing, we show that *the process can be well modeled by a 4-state Markov chain*, which is our basis for identifying whether a conversation is *human-like*. Our VoIP flow identification scheme comprises two phases: a training phase and an identification phase. In the training phase, a set of human conversation patterns are used to derive the transition probabilities of a Markov model and train the classifier that will be used in the next phase. Then, in the identification phase, we use the trained Markov chain to compute the likelihood value and derive the pattern features of an unknown interaction process, which indicate the *humanness* of the conversation, and apply a supervised classifier to determine whether a flow is VoIP.

One advantage of our scheme is that the detection process can be implemented in the *early stages* of a network flow; thus, it is particularly useful in traffic management and QoS provisioning. In terms of performance in detecting VoIP flows, the scheme achieves 95% identification accuracy in the first 4 seconds of the detection time and 97% within 11 seconds.

In this paper, our contribution is two-fold. 1) We propose a real-time VoIP flow identification scheme based on human conversation patterns. It is robust to silence suppression and traffic dynamics due to network congestion and protocol design. 2) We evaluate the proposed scheme with extensive real-life traces and show that it achieves a high identification rate within a short detection time.

The remainder of this paper is organized as follows. In Section 2, we review related works. We discuss the data collection methodology and summarize our traces in Section 3. In Section 4, we describe the approach for inferring speech activity. In Section 5, we discuss the intuition behind our approach, and then present a detailed description of our identification scheme in Section 6. In Section 7, we evaluate the performance of the proposed scheme with extensive traces. Then, in Section 8, we summarize our conclusions.

2 Related Work

In recent years, there has been a great deal of research in the area of network traffic classification. One traditional approach identifies traffic based on port numbers, but it is becoming less effective because dynamic ports are currently used in many applications, especially peer-to-peer applications. Another widely used approach is based on payload matching, which analyzes a packet's payload to search for the specific signature of the application. However, the approach can only be used for applications whose signature is known and cannot be applied

Table 1. Trace Summary

Category	# Connections	Duration	# Packets	Packet Rate (1/sec)	Bytes
Skype	462	2,388 (min)	4,728,240	33	4,318 (MB)
TELNET	2,008	4,729 (min)	10,559,261	37	7,331 (MB)
WoW	1,406	1,537 (min)	2,528,359	27	680 (MB)
P2P	15,845	3,334 (min)	29,220,870	146	30,500 (MB)
HTTP	2,224	120 (min)	28,264,360	3,925	59,097 (MB)

to encrypted traffic. Furthermore, examining payloads raises personal privacy concerns.

Flow statistics have also been used to classify network traffic. In [5], Moore et al. use a supervised machine learning technique, the naive Bayesian classifier, to categorize traffic by application types. In [6, 7], an unsupervised clustering algorithm is proposed for traffic identification. These works focus on offline traffic classification for the purposes of traffic trend analysis and network planning. They do not consider online traffic identification, which is essential for real-time traffic management.

Another approach used to identify network traffic is based on the specific signature in packet exchanges between hosts. In [8], Dahmouni et al. modeled the sequence signature of TCP control packets with a first-order Markov chain. They first inferred the transition probabilities of a Markov model for each known application in the learning step, and then identified traffic based on the derived transition probabilities. Our scheme is similar to that in [8] as we also adopt Markov modeling; however, instead of relying on TCP control packets, our method is based on the conversation patterns between interacting hosts.

3 Data Description

We evaluated the proposed VoIP detection scheme on real-life Internet traffic traces obtained from five types of network applications: VoIP, TELNET, HTTP, P2P, and online games. Skype, a popular VoIP software, was chosen to represent VoIP applications. The collection procedures for our traces were as follows. 1) Skype traffic was captured according to the procedures detailed in [9]. 2) TELNET traffic was captured on a gateway router for all TCP flows with port numbers 22 (SSH) and 23 (telnet); all intra-campus traffic was removed. 3) We chose World of Warcraft, a popular MMORPG (Massively Multiplayer Online Role-Playing Game), to represent online games. The traffic was captured on a gateway router for all TCP flows with port number 3274; either the source or destination address is within the network 203.66 (where the World of Warcraft server is located in Taiwan). 4) P2P traffic was captured on a dedicated PC running BitComet, a variant of BitTorrent client [10]. As the BitTorrent protocol does not use a fixed port number, we recorded all the flows that used port numbers higher than 1024.

To ensure there were sufficient packet samples in each flow, we removed flows containing less than $2,000$ packets. The collected traffic traces are summarized in Table 1.

4 Speech Activity Inference

The method used to infer speech activity from network traffic depends on whether or not *silence suppression* is employed. In this section, we discuss two methods (i.e., for applications with or without silence suppression), and present an algorithm that integrates them to detect speech activity in VoIP traffic.

4.1 Traffic with Silence Suppression

Some VoIP applications employ silence suppression, which reduces the packet sending rate when the user is not talking. The objective is to conserve network bandwidth and maximize the utilization of communication channels. We can infer the presence or absence of speech by the level of the packet rate during a short period. Specifically, a period is deemed a silence period if there are no packets longer than a threshold of 100 ms; otherwise, it is considered a speech period.

4.2 Traffic without Silence Suppression

Some VoIP applications, such as Skype and UGS [11], do not employ silence suppression. This design is intentional to ensure the UDP port bindings at the NAT and allow the background sounds to be heard all the time [12]. In this case, speech activity cannot be identified by simply observing the packet rate.

To infer speech activity from non-silence-suppressed VoIP traffic, we employ an algorithm adapted from [9], where the packet size is used to indicate whether a speech burst is present or not. The steps of the algorithm are as follows. First, we apply an exponential weighted moving average (EWMA) to remove high-frequency fluctuations in the packet size process and obtain a smoothed process. Second, we denote each peak and trough as $(t_i,\ s_i)$, where t_i denotes the occurrence time of the peak or trough and s_i denotes the smoothed packet size. For each pair of adjacent troughs on the trough list, $(t_a,\ s_a)$ and $(t_b,\ s_b)$, if there is more than one peak on the peak list between these two troughs, we take the peak with the largest packet and denote its packet size as s_p. We then draw a line from $(t_a,\ (s_a + s_p)/2)$ to $(t_b,\ (s_b + s_p)/2)$ as an adaptive threshold. Finally, we determine the state of each voice sample as ON or OFF by checking whether the size of the smoothed packet is greater than any of the adaptive thresholds defined at the time the sample was taken.

Algorithm 1 Speech activity inference from VoIP traffic

```
for each flow do
  observe the size of packets for 1 second
  if an idle period in either direction then
    infer speech activity based on packet rate
  else
    infer speech activity based on packet size
  end if
end for
```

4.3 Algorithm

We now present an integrated algorithm that can detect voice activity in VoIP flows in general, as shown in Algorithm 1. First, we observe the packet rate in either direction for a specific period, e.g., 1 second, to determine whether the flow is silence-suppressed. Because double-talk (i.e., both call parties talk at the same time) is normally short and infrequent, a period of continuous packets implies that the observed flow is not silence-suppressed. In this case, we infer voice activity in the VoIP flow based on the packet size. Otherwise, we assume the flow is generated by an application that employs silence suppression, and infer the conversation pattern embedded in the flow based on the packet rate.

5 Motivation for the Proposed Scheme

In this section, we explain the intuition behind our approach. We consider that each type of network application possesses a unique conversation pattern. In addition, since *human interaction* is different to the interaction between computer applications, VoIP traffic can be identified based on the embedded human conversation activity. We also provide a graphical illustration of the conversation patterns of a number of applications to support our argument.

5.1 Application Behavior Analysis

To understand why human conversation patterns differ from the traffic patterns of other applications, in the following, we present a behavioral analysis of common applications.

- File transfer applications, such as FTP, are comprised of unidirectional flows. That is, downloading a file corresponds to a network flow containing only server-to-client packets, except for a few control messages from the client. On the other hand, if a client uploads a file, the traffic will contain only client-to-server packets. Moreover, a file transfer session often continues for a long period and achieves a stable bit rate in the long-term.
- In web browsing, when a user clicks on a URL, the browser sends a simple HTTP request message to the web server. The response message from the

Table 2. Summary of application behavior

	File transfer	HTTP	TELNET	Online games	Video streaming	Video con.	VoIP
Unidirectionality	✓				✓		
Independence				✓		✓	
High interactivity							✓
Bulk transfer	✓	✓			✓	✓	
Large packet	✓	✓	✓	✓	✓		
CPR-like†				✓			✓

† constant packet rate

server normally comprises large objects, such as images, video clips, and document files. Therefore, HTTP traffic usually consists of small requests and large response messages.

- In TELNET applications, packets from a client to a server are normally small, as they only contain a few commands; in contrast, packets in the opposite direction often contain much more information. For example, if a client issues the command “`ls`,” the server will reply with a list of files and directories that may contain hundreds of lines of text. Moreover, a TELNET user often spends time thinking about the next step or waiting for responses from the remote server, so the inter-packet time of the client traffic is likely to be highly variable.
- In real-time interactive online games, a client issues commands to direct the virtual avatar to move, chat, or perform other actions in the game. At the same time, the game server regularly sends out the latest game states to each client to maintain the state consistency of the virtual world. Because client packets are mostly generated according to player decisions and server packets are regulated by system timers and the dynamics of the virtual environment, the game traffic in either direction is likely to be *independent*.
- In video streaming applications, the traffic pattern is similar to that of bulk transfer. In other words, a client receives multimedia content continuously and seldom sends out packets, except for control messages.
- In video conferencing applications, participating computers send out video packets continuously to update the display on the remote hosts. At the same time, they send out audio packets independently of the video traffic. The audio packets can be sent at constant or variable intervals depending on whether silence suppression is enabled. Therefore, the traffic pattern of video conferencing is bidirectional, and the traffic in either direction is likely to be independent due to the relatively large volume of the video stream.

Table 2 summarizes the behavior of the applications analyzed in this study. We consider that the conversation pattern between two people is normally *highly interactive* and *highly interdependent*. According to our analysis, VoIP traffic is the only traffic type that exhibits both properties, which is the basis of our proposed VoIP flow identification algorithm.

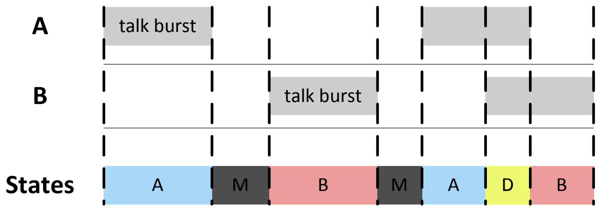

Fig. 1. Voice activity between two speakers and the conversation pattern

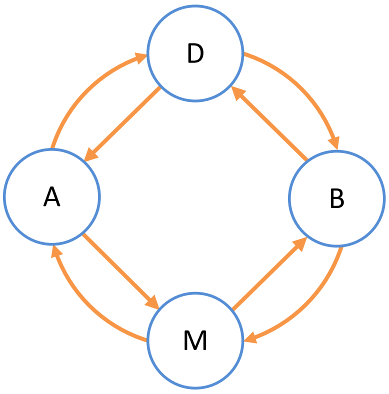

Fig. 2. Conversation model

5.2 Conversation Modeling

In some studies of human speech characteristics, conversation patterns have been modeled using *Markov chains* [13, 14]. A 4-state model for generating artificial conversational speech is proposed in [15]. For two speakers, A and B, engaged in a conversation, the four states are as follows: state $\mathcal{A}$ represents that A is talking and B is silent; state $\mathcal{B}$ represents that A is silent and B is talking; state $\mathcal{D}$ indicates double-talking; and state $\mathcal{M}$ denotes mutual silence. We illustrate the definitions of the four states in Fig. 1, and the transitions between the four states in Fig. 2. Because of its simplicity, we employ this 4-state Markov chain for human conversation modeling in this study.

5.3 Conversation Patterns: A Graphical Comparison

To verify our analysis of application behavior, we randomly select 10 flows from each of our traces and plot the conversation pattern of each flow, as shown in Fig. 3.

On the graph, each flow is divided into a number of periods and the conversation state in each period is determined by the traffic direction. The flows of HTTP, P2P and TELNET are assigned the state $\mathcal{M}$ most of time because their inter-packet times are normally large. On the other hand, VoIP flows consist

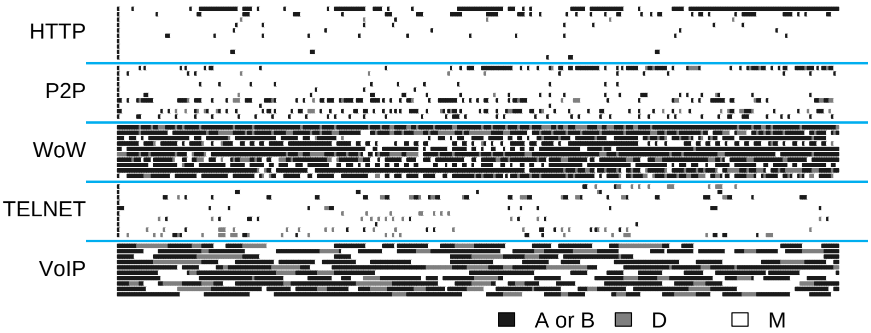

Fig. 3. The traffic patterns of the five applications considered in this study

mainly of states $\mathcal{A}$ and $\mathcal{B}$, as they reflect human conversations in which normally only one party speaks at a time. We also observe that the conversation pattern of WoW is more fragmented and disordered than that of VoIP because the interaction in online games is generally more frequent than in verbal conversations. In the following, we discuss the unique characteristics of each application.

HTTP: Normally, both the HTTP client and server remain silent for a while after a web page is downloaded, since the user needs time to read the page; this behavior is represented by state $\mathcal{A}$ or $\mathcal{B}$ followed by a long state $\mathcal{M}$. Because HTTP works in an *interactive and alternating* manner, the client and server never send packets simultaneously; thus, state $\mathcal{D}$ is not found in the pattern.

P2P: In the conversation pattern of P2P applications, we observe a trend that single-talk states occur *periodically*. Intuitively, a host running P2P file-transfer applications must communicate regularly with its peers in order to send or receive the up-to-date block table for the files currently being transferred.

WoW: Although WoW is a real-time interactive application like VoIP, its traffic pattern is more *fragmented and chaotic* than that of VoIP for the following reasons: 1) the interaction in online games generally occurs in smaller time scales, where game actions are decided in *sub-seconds* and speech bursts usually last for a number of seconds; and 2) the commands from game clients and the responses from servers are nearly *independent*, so the traffic pattern of WoW frequently alternates between the four states.

TELNET: Like HTTP, the traffic pattern of TELNET applications consists mainly of state $\mathcal{M}$, which represents the time users spend reading information on the screen and thinking about the next action. However, the pattern of TELNET is different from that of HTTP because the length of consecutive non-$\mathcal{M}$ states is short and $\mathcal{D}$ states may occur in TELNET.

VoIP: We have identified the following unique characteristics in the VoIP conversation pattern: 1) each of the four states, tends to hold for a period, e.g., longer than one second; 2) the frequency and duration of single-talk states are often higher and longer than those of double-talk states; 3) there are two types of $\mathcal{M}$ states: short states, which may indicate gaps between words or sentences; and long states, which represent periods of silence when each speaker is thinking or waiting for the other to speak. To sum up, the VoIP traffic pattern accurately reflects human conversation patterns, i.e., *high interactivity, bi-directionality, and the interdependency between both directions.*

Through graphical comparisons, we show that the VoIP traffic pattern is very different from that of other applications. In addition, the 4-state model is effective for representing the differences between the applications' conversation patterns.

6 Methodology

In this section, we propose a VoIP flow identification scheme based on the unique human speech conversation patterns embedded in voice traffic. Our scheme comprises two phases: a training phase and an identification phase. In the training phase, we learn the parameters of a Markov model and train the classifier based on a set of existing traces. Then, in the identification phase, we use a supervised classification approach to detect VoIP flows.

6.1 Training Phase

In this phase, we use a 4-state Markov chain to model the human speech conversation pattern, and then apply a naive Bayesian classifier to distinguish VoIP flows from other flows.

Markov Chain: Given a set of known VoIP flows, we compute the initial probabilities of states $S_i \in \{\mathcal{A}, \mathcal{B}, \mathcal{D}, \mathcal{M}\}$ based on the proportion of their mean sojourn times, and compute the transition probabilities based on the empirical transition frequency between the states. We treat the trained Markov chain as representative of *typical* speech conversation patterns. Any conversation pattern that can be well-modeled by this approach is considered human-like and will be considered as a VoIP flow in the identification phase.

Naive Bayesian Classifier: To determine how well the model fit indicates that a traffic pattern is a human conversation, we employ a naive Bayesian classifier [5], which is a supervised machine learning tool. We use the flows from all the applications in our data set (Section 3) to train the classifier. For each training flow, we compute the likelihood of its conversation pattern generated by the trained Markov chain. Given a state sequence $S_1, S_2, \ldots S_n$, where $S_i \in \{\mathcal{A}, \mathcal{B}, \mathcal{D}, \mathcal{M}\}$, we compute the log-likelihood of the sequence as

$$\log(P_{1,2} \times P_{2,3} \times \ldots \times P_{(n-1),n}), \tag{1}$$

where $P_{i,j}$ is the transition probability of the state sequence $S_i S_j$. Since flows may vary in length, we define the *normalized log-likelihood value* as

$$\log(P_{1,2} \times P_{2,3} \times \ldots \times P_{(n-1),n})/N, \tag{2}$$

where N is the length of the sequence. For VoIP flows, the computed log-likelihood tends to be large as the Markov chain represents typical human conversation patterns. Because non-VoIP flows normally exhibit non-human-like

Table 3. The features used in the naive Bayesian classifier

Features
Normalized log-likelihood value based on the Markov chain
Speech period of party A or B (mean, standard deviation)
Sojourn time in each states[†](mean, standard deviation)
Ratio of sojourn time in each states[†]
Alternation rate between states[†]

[†] states $\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{M}$

behavior, such as being non-interactive, independent, and unidirectional, they likely lead to low log-likelihoods as their behavior does not fit the Markov chain well. Therefore, we use the computed log-likelihood value as one of features to train the naive Bayesian classifier.

In addition, based on the conversation pattern, we derive other features for each training flow, as shown in Table 3. We compute the mean and standard deviation of the period that party A (resp. B) speaks each time, which reflects the bidirectional behavior of VoIP flows. To reveal the interactive behavior, we infer the summary of the sojourn time for four states $\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{M}$. The last feature, the state alternation rate, which is the transition rate between the four states, reflects the fragmented and disordered level of traffic patterns. Since the VoIP traffic pattern is unique, these features and the log-likelihood values reveal the distinct behavior of VoIP applications; thus, we employ them to train the classifier that will be used in the classification stage.

6.2 Identification Phase

In the flow identification phase, we extract the conversation pattern from each flow, compute the normalized log-likelihood of the pattern, and determine whether the flow was generated by VoIP applications with the naive Bayesian classifier in the training phase.

7 Performance Evaluation

In this section, we first consider the effect of the order of the Markov model on the performance of flow identification, and compare the conversation patterns generated by different models to find the most appropriate order. Next, we evaluate the effect of detection time on the detection accuracy.

7.1 Effect of the Order of the Markov Model

Under our proposed scheme, the order of the Markov chain used to model human conversation patterns could affect the identification accuracy. In a first-order Markov chain, the next state only depends on the current state, which implies a

Table 4. Transition probabilities of the 1st-order markov chain

	A	B	D	M
A	0.9022	0.0028	0.0380	0.0571
B	0.0029	0.9030	0.0391	0.0550
D	0.0607	0.0592	0.8763	0.0038
M	0.0465	0.0439	0.0019	0.9078

Table 5. Transition probabilities of the 2nd-order markov chain

	A	B	D	M
AA	0.9067	0.0034	0.0380	0.0519
AB	0.0000	1.0000	0.0000	0.0000
AD	0.0000	0.0781	0.9219	0.0000
AM	0.0000	0.0635	0.0000	0.9366
BA	1.0000	0.0000	0.0000	0.0000
BB	0.0032	0.9115	0.0346	0.0507
BD	0.0775	0.0001	0.9224	0.0000
BM	0.0641	0.0000	0.0000	0.9359
DA	0.9486	0.0000	0.0000	0.0514
DB	0.0000	0.9518	0.0000	0.0482
DD	0.0633	0.0678	0.8651	0.0037
DM	0.0000	0.0000	0.0000	1.0000
MA	0.9586	0.0000	0.0414	0.0000
MB	0.0000	0.9562	0.0438	0.0000
MD	0.0000	0.0005	0.9995	0.0000
MM	0.0549	0.0526	0.0025	0.8901

memoryless process. On the other hand, a second- or higher-order Markov chain considers the current state as well as the previous states. Finding an appropriate order for the Markov chain is essential to obtain a good fit of human conversation patterns and achieve high identification accuracy in our approach.

To examine the impact of the order of the Markov chain, we compare real human conversation patterns with patterns generated by different Markov models. Based on real human conversation traces, we first derive the transition probabilities, as shown in Tables 4 and 5. We then generate conversation patterns using the first- and second-order Markov chains respectively, as shown in Fig. 4. The *real* series shows the patterns of empirical conversations in our trace; the *2nd* and *1st* patterns are generated by the second- and first-order Markov chains, respectively. In addition, we generate a pattern with the *indep.* model, where the states are generated in an independent and identically-distributed (IID) manner.

Compared to the *real* case, the state alternation of the *indep.* model is rapid and disordered. Because of the large difference from true human behavior, the *indep.* model is clearly not suitable for describing human conversations. On the other hand, the patterns of the *1st* and *2nd* models are similar to those of the

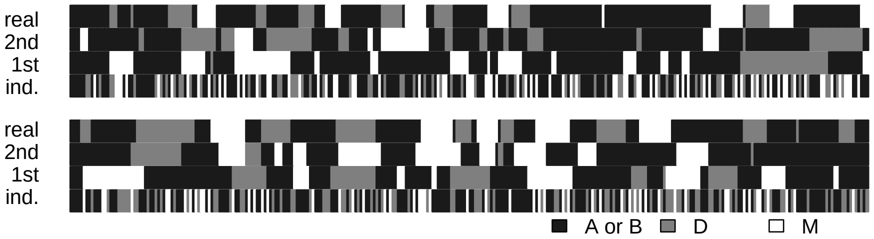

Fig. 4. Comparison of real human conversation patterns and artificial patterns generated by an nth-order Markov chain

Table 6. Summary of the mean sojourn times (sec.) for the pattern in Fig. 4

	A	*B*	*D*	*M*
real	1.03	1.15	0.75	0.86
2nd	1.21	1.17	0.66	0.88
1st	1.18	1.02	1.21	0.71
indep.	0.13	0.13	0.13	0.12

real case. For a detailed comparison, Table 6 lists the mean sojourn time for each state. Compared with the *1st* model, the mean state sojourn time in the *2nd* model is quite close to that of the *real* case, which indicates that the second-order Markov chain provides a better fit for human conversations. This also supports the intuition that a model with a larger memory can represent the evolution of a process more exactly.

While higher-order Markov chains generally provide better goodness-of-fit, the computational overhead is higher due to the model's complexity. In terms of the trade-off between computation time and identification accuracy, we consider that a second-order Markov chain provides the right balance. Thus, we adopt the second-order Markov chain in our proposed scheme.

7.2 Effect of Detection Time

Since our goal is to detect VoIP flows in real time, the detection time is a major concern. As shown in Fig. 5, the proposed scheme achieves a identification accuracy of 95% in the first 4 seconds of the detection time and 97% within 11 seconds. For a detailed identification performance, we plot the impact of the detection time on the true positive rate and true negative rate in Fig. 6. The true positive rate is estimated as

$$\text{TPR} = \frac{\text{The number of VoIP flows correctly identified}}{\text{The number of total VoIP flows}}, \tag{3}$$

while the true negative rate is

$$\text{TNR} = \frac{\text{The number of non-VoIP flows correctly identified}}{\text{The number of total non-VoIP flows}}. \tag{4}$$

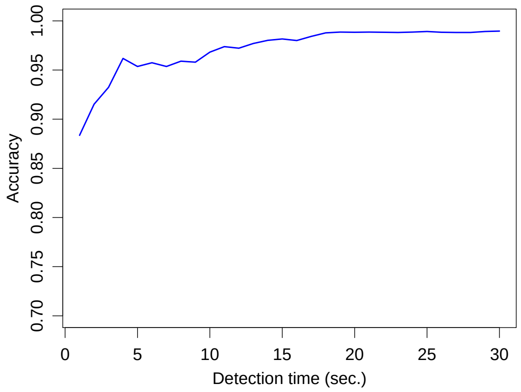

Fig. 5. Influence of the detection time on accuracy

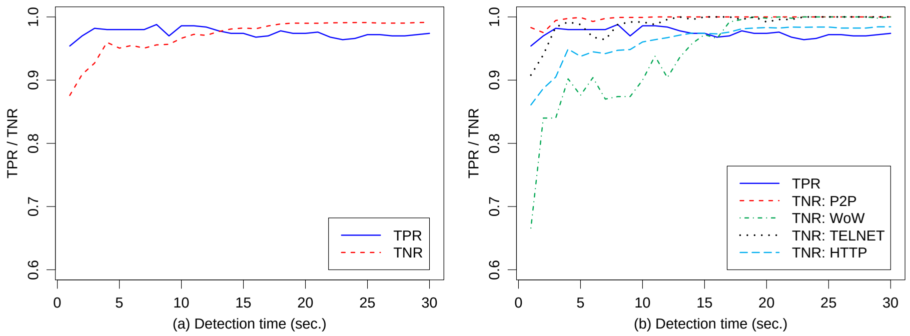

Fig. 6. Influence of the detection time on true positive rate (TPR) and true negative rate (TNR)

As shown in Fig. 6(a), the true positive rate is 95% in the first seconds of the detection time. Specifically, the rate is higher than 97% for a detection time longer than 2 seconds. On the other hand, the true negative rate is 95% in the first 4 seconds of the detection time and 97% within 11 seconds. Fig. 6(b) shows the true negative rate for each non-VoIP application. We find that, among the non-VoIP applications, the flows of WoW tend to be mis-identified as VoIP. One possible explanation is that clients may be idle in some sessions so that the traffic consists mainly of server-to-client packets; therefore, the traffic pattern will be less disordered and close to that of VoIP. The true negative rate for WoW can achieve 90% with a detection time longer than 10 seconds.

In Fig. 7, we plot ROC curves for four classifiers that are based on different length of the detection time. The performance of a classifier is better than another while the true positive rate is close to 1 for a small false positive rate (i.e., TPR is higher and FPR is lower). From these curves, we observe that the

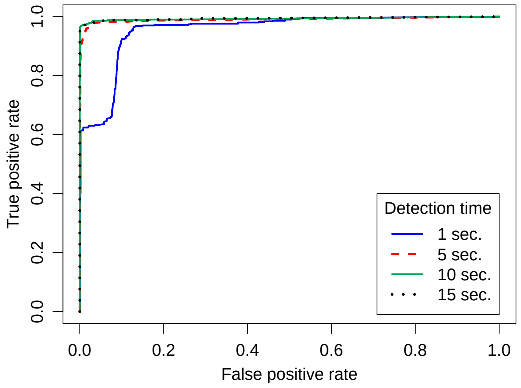

Fig. 7. ROC curves for four classifiers with different detection time

classifier achieves a high TPR and an extremely low FPR with a short detection time (5 seconds). Therefore, the results evidence that our proposed scheme can identify VoIP fows in a short time with a high accuracy.

8 Conclusion

In this paper, we propose a real-time VoIP identification scheme based on human conversation activity. By analyzing the human conversation patterns of common network applications, we show that the pattern embedded in VoIP traffic is distinct; hence it can serve as a unique signature for VoIP flow identification. Based on this finding, we propose a Markov chain-based algorithm to detect VoIP flows in real time. Through a performance evaluation based on extensive Internet traces, we show that the proposed scheme is very effective in terms of classification accuracy and detection time. Specifically, the identification accuracy is 95% within 4 seconds and 97% within 11 seconds.

References

1. TelTel: `http://www.teltel.com/`
2. XMeeting: `http://xmeeting.sourceforge.net/`
3. Skype: `http://www.skype.com/`
4. iVisit: `http://www.ivisit.com/`
5. Moore, A.W., Zuev, D.: Internet traffic classification using bayesian analysis techniques. In: Proceedings of the ACM SIGMETRICS'05, Banff, Alberta, Canada (2005) 50–60
6. Erman, J., Mahanti, A., Arlitt, M.F.: Internet traffic identification using machine learning. In: Proceedings of the IEEE GLOBECOM'06, San Francisco, California, USA (2006) 1–6

7. Erman, J., Mahanti, A., Arlitt, M., Williamson, C.: Identifying and discriminating between web and peer-to-peer traffic in the network core. In: Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada (2007) 883–892
8. Dahmouni, H., Vaton, S., Rossé, D.: A markovian signature-based approach to ip traffic classification. In: Proceedings of the 3rd annual ACM workshop on Mining network data, San Diego, California, USA (2007) 29–34
9. Chen, K.T., Huang, C.Y., Huang, P., Lei, C.L.: Quantifying Skype user satisfaction. In: Proceedings of ACM SIGCOMM'06, Pisa, Italy (Sep 2006) 399–410
10. Qiu, D., Srikant, R.: Modeling and performance analysis of bittorrent-like peer-to-peer networks. In: Proceedings of ACM SIGCOMM'04, Portland, OR, USA (August 2004) 367–378
11. Hou, F., Ho, P.H., Shen, X.S.: Performance evaluation for unsolicited grant service flows in 802.16 networks. In: Proceedings of the 2006 international conference on Wireless communications and mobile computing, Vancouver, British Columbia, Canada (July 2006) 991–996
12. Baset, S.A., Schulzrinne, H.G.: An analysis of the skype peer-to-peer internet telephony protocol. In: Proceedings of the IEEE INFOCOM'06, Barcelona, Spain (2006) 1–11
13. Brady, P.: A model for generating on-off speech patterns in two-way conversation. The Bell System Technical Journal **48**(9) (September 1969) 2445–2472
14. Stern, H.P., Wong, K.K.: A modified on-off model for conversational speech with short silence gaps. In: Proceedings of the 25th Southeastern Symposium on System Theory, Tuscaloosa, Alabama, USA (1993) 581–585
15. International Telecommunication Union: Artificial conversational speech. (1993) ITU-T Recommendation P.59.