Inferring Speech Activity from Encrypted Skype Traffic

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

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

***Abstract*—Normally, voice activity detection (VAD) refers to speech processing algorithms for detecting the presence or absence of human speech in segments of audio signals. In this paper, however, we focus on speech detection algorithms that take VoIP traffic instead of audio signals as input. We call this category of algorithms network-level VAD.**

Traditional VAD usually plays a fundamental role in speech processing systems because of its ability to delimit speech segments. Network-level VAD, on the other hand, can be quite helpful in network management, which is the motivation for our study. We propose the first real-time network-level VAD algorithm that can extract voice activity from encrypted and non-silence-suppressed Skype traffic. We evaluate the speech detection accuracy of the proposed algorithm with extensive real-life traces. The results show that our scheme achieve reasonably good performance even high degree of randomness has been injected into the network traffic.

***Index Terms*—QoS Provisioning, Traffic Classification, User Satisfaction, VoIP, Voice Activity Detection**

I. Introduction

Traditionally, voice activity detection (VAD) refers to speech processing algorithms for detecting the presence or absence of human speech in segments of *audio signals*. One of the most well-known applications of VAD is called silence suppression. To implement silence suppression, a speech coder needs to incorporate a VAD module so that it only outputs sound signals when human speech is present and the signal length is therefore reduced. By so doing, silence suppression can reduce the network bandwidth used by voice packets and achieve higher communication channel utilization. In addition, VAD has many applications in speech processing systems, such as speech encoding, echo cancellation, and speech recognition. Hereafter, we call this category of speech detection algorithms *source-level VAD*, as they operate on audio signals directly.

In this paper, we focus on speech detection algorithms that take *VoIP traffic* instead of audio signals as input. We call this category of algorithms *network-level VAD*. The location where the algorithm is implemented depends on the type of signal that each category of VAD algorithms process. Because source-level VAD deals with audio signals, the VAD module usually resides in end-users' PCs or phones. In contrast, network-level VAD infers speech activity from network traffic, so it can run on any network node. From the perspective of applications, source-level VAD usually plays a fundamental role in speech processing systems because of its ability to delimit speech segments. Network-level VAD, on the other hand, can be helpful in network management. The differences between source-level and network-level VAD are summarized in Table I.

TABLE I
The differences between source-level and network-level voice activity detection

	source-level	network-level
input	audio signal	network traffic
location	speaker's host	network node
purpose	silence suppression, echo cancellation	traffic management, QoS measurement

Motivation. Our proposed network-level VAD scheme, which infers speech activity from network traffic, is motivated by its potential applications in *network management*. In this paper, we consider one of those applications, namely *VoIP flow identification*. Flow identification is an essential component in network traffic and QoS management. In business enterprises, there is often a need to manage VoIP flows due to institutional policies, such as restricting calls to certain destinations, or blocking calls at certain times. Providing better QoS is another motivation for identifying VoIP flows. While such flows are difficult to recognize due to proprietary protocols, non-standard port numbers, and encrypted payloads, the *human conversation pattern* embedded in the traffic can be a unique signature of VoIP flows. The discrepancy arises because the conversation activity between two people often occurs on a *multi-second time scale*, while the traffic patterns of other applications, e.g., web traffic or online game traffic, are usually on a sub-second time scale [3].

Challenges. Intuitively, it should be easy to achieve network-level VAD by simply examining the payload of VoIP packets. In ideal circumstances, we would be able to extract speech signals from the packets' payloads, and then apply source-level VAD to the signals. However, *payload encryption* is becoming a common design feature to preserve privacy, so that parties other than call participants cannot know the content of a conversation. Even if one VoIP application does not encrypt its packets, the packets' payloads may be inaccessible because obtaining such information would be a violation of privacy. An alternative solution is to determine speech activity based on the packet rate, as VoIP applications normally use silence suppression for channel multiplexing and more efficient communications [1]. However, an increasing number

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council of the Republic of China under the grants NSC 97-2219-E-001-001, NSC 97-2219-E-011-006, and NSC 96-2628-E-001-027-MY3.

of VoIP applications, such as Skype and UGS [2], *do not support silence suppression* in order to obtain better voice quality and maintain UDP bindings at the NAT. This indicates that we cannot rely on either the packet payload or the packet rate to determine the presence or non-presence of speech bursts. Specifically, the traffic generated by Skype, one of the most popular VoIP applications, raises both challenges—Skype's traffic is encrypted and the system generates packets no matter whether a speaker is talking or not. Thus, we need a more sophisticated means than intuitive methods to infer speech activity from *encrypted and non-silence-suppressed VoIP traffic*.

Contributions. Given the above difficulties, in this paper, we propose a scheme that can infer speech activity from encrypted Skype traffic with no silence suppression. We chose Skype as our study subject because its traffic exhibits both difficulties—encryption and a constant packet rate. The proposed scheme is based on our observation that, in Skype traffic, *speech activity is highly correlated to packet size*, as more information will be encoded in a voice packet while a user is speaking. To demonstrate this point, we show the sound volume of an audio segment and the sizes of the voice packets corresponding to the segment in Fig. 2. The graph reveals that the packet size and speech volume are highly correlated as they fluctuate in tandem.

The contribution of this paper is two-fold. 1) Traditional VAD algorithms work on audio signals that are only available on the speakers' devices, whereas we propose using network-level VAD to infer speech activity from VoIP traffic. 2) We propose the first real-time network-level VAD algorithm, which is able to deal with encrypted and non-silence-suppressed VoIP traffic.

The remainder of this paper is organized as follows. Section II summarizes earlier studies on speech detection. We describe our trace collection methodology and summarize our traces in Section III. In Section IV, we propose a network-level VAD algorithm for encrypted and non-silence-suppressed VoIP traffic. The speech detection accuracy of the proposed algorithm is evaluated in Section V. We then summarize our conclusions in Section VI.

II. Related Work

When designing source-level VAD algorithms, it is difficult to determine whether a speech burst is present because of background noise. As a result, a large number of source-level VAD algorithms have been proposed to reduce the influence of background sound. For example, in [5], Hoyt and Wechsler used concave or convex formant patterns to detect the presence of speech. The algorithm proposed in [4] is based on a pattern recognition approach, in which the matching phase follows six fuzzy rules and is trained by a new hybrid learning tool. In [7], Prasad, Vijay, and Shankar used an information theoretic measure, called spectral entropy, to differentiate silence segments from speech segments.

Our approach differs from earlier studies in a number of respects, as shown in Table I. We consider that detecting speech activity based on network traffic is more difficult than that based on audio signals. This is because network traffic takes the form of audio signals compressed by an encoder, which may adjust the coding level, redundancy level, quantization levels, and packetization period at any time to achieve more efficient voice transmission. All these factors introduce more randomness into VoIP traffic and thus make the design of network-level VAD algorithms more challenging.

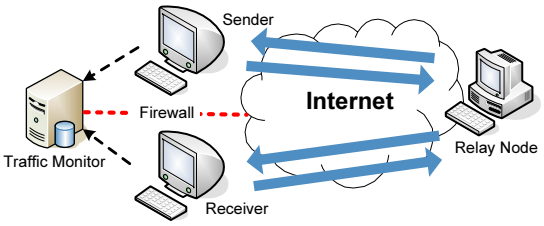

Fig. 1. Network setup for measuring processing delays at any relay node on the Internet.

III. Data Description

In this section, we describe the data set we used to develop, test, and validate our proposed VAD algorithm. The data set is comprised of two traces: 1) VoIP traffic, and 2) the corresponding speech signals, to verify the correspondence of VoIP traffic and the speech activity inferred by our scheme. In the following, we first explain the experiment setup and the method used to collect the traffic and speech signal traces. We then describe how we extract speech activity from audio recordings for use as the ground-truth in subsequent performance evaluations.

A. Experiment Setup

We conducted a number of Internet experiments to collect Skype traffic because it mimics the real-life traffic that a network node would encounter. The reason is that the structure and pattern of network traffic may change due to network impairment, i.e., network delay and packet loss, or the application itself. For example, Skype may adjust its encoding factor, redundancy level, and packetization intervals to adapt to the host and network load.

Our trace collection mechanism comprises three commodity PCs deployed in the way shown in Fig. 1. One serves as the VoIP sender, one serves as the receiver, and the third is a relay node that collects information about the traffic of both call parties. The collection procedure is as follows:

1) When the measurement program is initialized, we block the sender from reaching the receiver directly with a firewall program `ipfw`.
2) The sender initiates a VoIP call to the receiver (via the receiver's Skype name). Because of the firewall setting, the sender will be connected to the receiver host via one of its relay nodes.
3) If a VoIP call is established, we know that Skype has found a relay node to relay voice packets between the sender and the receiver.
   Occasionally, after a few retries, a VoIP call still cannot be established because the sender fails to find a usable

TABLE II
SUMMARY OF VOIP TRACES

Total # of traces	# TCP	# UDP
1839	1427	412
# Relay node	Mean packet size	Mean time period
1677	109.6 bytes	612.5 sec

relay node from the candidate list. In this case, our daemon will restart the Skype program and re-attempt to establish a VoIP call. The Skype program will retrieve a new list of relay nodes from the central server at the startup, so further VoIP calls should be successful.
In addition, we sometimes find that voice packets from the sender to receiver and vice versa are relayed through different delay nodes. In this case, we simply drop the call, block both relay nodes, and re-dial.

4) To simulate a conversation, a WAV file comprised of all the English recordings in the Open Speech Repository[1] is played continuously for both parties during a call. At the same time, we record the received voice data in MP3 files at the receiver by using a Skype plugin program `Pamela`.
5) After a call has lasted 10 minutes, we block the current relay node at the sender by `ipfw` and terminate the call. We then wait for 30 seconds before re-iterating the loop from Step 2.

We ran the trace collection procedures over a two-month period in mid-2007 and collected $1,839$ calls, which are summarized in Table 2. Of the $1,839$ traces, $1,427$ were based on TCP, and 412 were based on UDP.

B. Extracting Speech activity from Audio Recordings

To evaluate the performance of our VAD algorithm, in addition to network traffic, we need the "true" speech activity that was "heard" by the receiver. Therefore, we recorded the audio signals that were played by the receiver host's output device into WAV files and applied a source-level VAD algorithm to the speech activity in the sound recordings.

Our source-level VAD scheme is based on the sound volume, which is a commonly used indicator of voice activity. We apply the function

$$volume = 10 * log(\sum_i S_i^2) \quad (1)$$

in [6] to compute the volume of each sound sample, S_i, in a segment of audio signals. The quantity computed is referred to as the *log energy* in units of decibels and the series we obtain is called the *volume process*.

To determine whether speech is present at a given time, we apply a static thresholding method to the volume process. First, we construct a window of 5 samples in the volume process and observe whether the difference between a local extreme and any other samples is larger than 50 db. Then, we collect all the local maximums and local minimums in the volume process, where the former indicate the volumes of speech bursts, and the latter indicate the volume when speech is not present. We found that the volumes of local maximums are in the range of 210 db and 250 db where the most volumes of local maximums lie in 230 db, and the volumes of local minimums are in the range of 110 db and 160 db where the most volumes of local minimums lie in 136 db. Then we computed the threshold for determining whether a speech burst is present by the half-way point between two extremes, and obtained 183 db as our static threshold.

[1] `http://www.voiptroubleshooter.com/open_speech/`

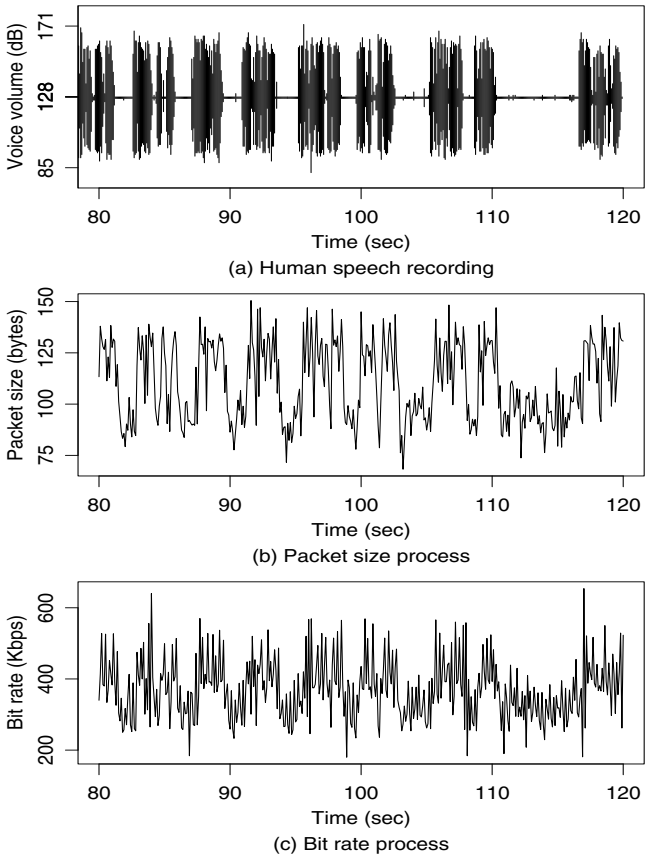

Fig. 2. The volume process of a human speech recording and the packet size and bit rate process of the corresponding VoIP traffic.

Based on the computed static threshold, we classify each speech sample as speech or silence depending on whether the volume of the sample is higher than the threshold. We define an *ON period* as a segment of speech signals whose volume is higher than the threshold, and define an *OFF period* as a segment of speech signals that is considered silence. In Section V, we use the inferred ON and OFF periods to evaluate the accuracy of our VAD algorithm, which infers speech activity from network traffic.

IV. THE PROPOSED SCHEME

In this section, we present our VAD algorithm for extracting conversation activity from encrypted and non-silence-suppressed VoIP traffic. We first explain our choice of packet size as the indicator of voice activity, after which we discuss the design of our voice activity detection scheme, which comprises two phases—smoothing and dynamic thresholding.

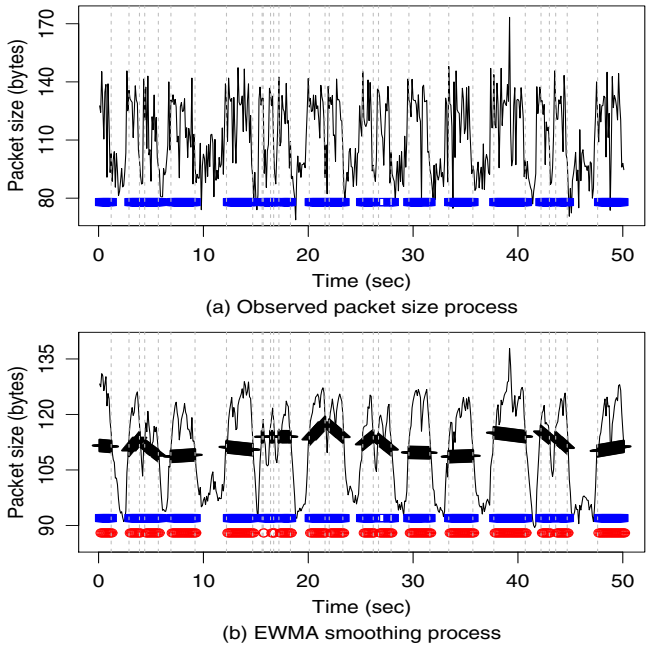

Fig. 3. The packet size process and its smoothed process, along with the computed adaptive thresholds and the true and estimated speech activity.

From our preliminary analysis, we find that both the *packet size* and the *bit rate* are indicators of speech activity, even though the packet payload is encrypted. Fig. 2 shows a comparison of the volumes, packet sizes, and bit rates that correspond to a segment of speech signals. The graph reveals that both the packet size and the bit rate correlate, to some extent, with the volume process. We use correlation coefficients to quantity the strength of their relationships. On average, the correlation coefficient between the volume and the packet size process is 0.78. Between the volume and the bit rate process, it is 0.59; and between the volume and the packet rate process, it is only -0.02. The result suggests that the packet rate is nearly independent of user speech. Although the bit rate exhibits a reasonable correlation with the volume process, it may contain randomness due to packet retransmission or congestion control mechanisms. Therefore, we adopted the packet size process as the basis for inferring voice activity.

As there is a strong correlation between packet size and speech activity, intuitively, a static threshold should be sufficient to determine whether speech is present. However, we find that a static threshold is not feasible because the packet size process is not stationary as its mean may change over time. This may occur because Skype adjusts the encoding bit rate, redundancy factor, and the packetiziation delay according to the host's CPU load, the congestion level, and the bandwidth of the network path. To address these challenges, we apply a smoothing procedure to remove high-frequency variations in the packet size process, and then apply an *adaptive thresholding* mechanism to deal with the non-stationarity of the packet sizes.

A. Smoothing

We apply an exponentially weighted moving average (EWMA) on the packet size process to remove high-frequency fluctuations. The EWMA is defined as

$$P_i = \lambda Y_i + (1 - \lambda) P_{i-1}, \tag{2}$$

where Y_i denotes the observed packet size in the ith time unit (a time unit of 0.1 second is used in this study) of the process, and P_i denotes the smoothed packet size in the ith time unit. We find that setting the weight λ to 0.2 achieves the best performance (in terms of voice activity detection accuracy, which is discussed in the next section), while λ within a range of 0.1 to 0.4 yields a similar performance.

B. Adaptive Thresholding

We now introduce the adaptive thresholding algorithm, which tries to find a reasonable threshold for determining the presence of speech given a non-stationary packet size process. The steps of the algorithm are as follows:

1) In the smoothed packet size process, we first find all the local maximums and local minimums within a window of 5 samples. In a window, if the difference between a local extreme and any other sample is greater than 25 bytes, we call it a "peak" if it is a local maximum, and a "trough" if it is a local minimum. The detected peaks and troughs are collected in a peak list and a trough list, respectively.
2) We denote each peak and trough as (t_i, s_i), where t_i means the occurrence time of the peak or trough and s_i refers to the smoothed packet size. For each pair of adjacent troughs, (t_a, s_a) and (t_b, s_b), on the trough list, if there are one or more peaks on the peak list between these two troughs, we take the peak with the largest packet size and denote the packet size as s_p. We then draw an imaginary line from $(t_a, (s_a + s_p)/2)$ to $(t_b, (s_b + s_p)/2)$ as an adaptive threshold.
3) We determine the state of each voice sample as ON or OFF period by checking whether the smoothed packet size is greater than any of the adaptive thresholds defined at the time the sample was obtained.

V. Performance Evaluation

In this section, we evaluate our proposed VAD algorithm based on the collected traces described in Section III. We begin by explaining how the algorithm works. Fig. 3 shows a comparison of the observed packet size process and the smoothed packet size process. On the upper graph, the blue squares mark the times a speech burst is present. On the lower graph, the lines formed by black crosses are the adaptive thresholds computed by our algorithm and taken as the boundary between speech and silence periods. From the figure, we observe that packet size smoothing removes high-frequency fluctuations without affecting the correlation between packet size and speech activity. The lower graph shows the level of agreement between the estimated speech activity (red circles) and the ground truth (blue squares). The graph indicates that the extracted ON/OFF periods reflect the true voice activity.

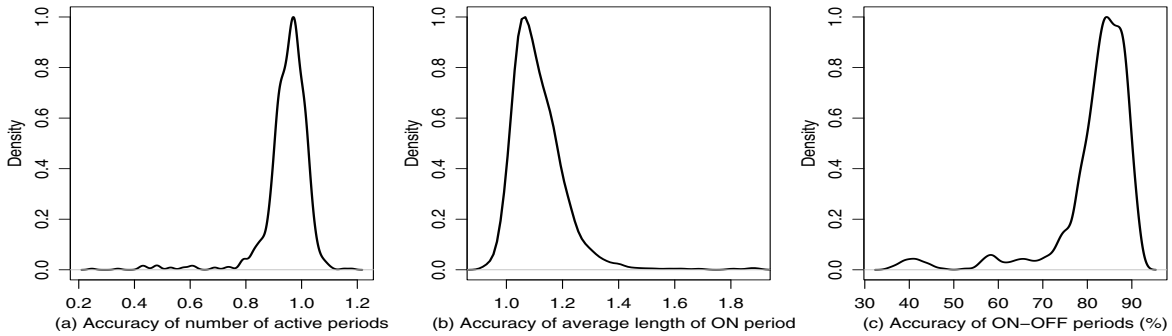

Fig. 4. Three figures show (a) The ratio of the number of active periods between true and estimated speech activity, (b) The ratio of the average length of active periods between true and estimated speech activity, and (c) The correctness of the inferred speech activity.

We use three metrics to evaluate the performance of our VAD algorithm:

- **Number of Active Periods**: We compare the number of active periods in the true and estimated ON/OFF periods. Let the number of estimated ON periods be S_{est} and the number of true ON periods be S_{true}. Then, the performance metric is defined as
$$\frac{S_{est}}{S_{true}}.$$
The left graph of Fig. 4 shows that the ratio of the number of active periods is around 1 for most calls. The result implies that our proposed VAD algorithm can extract approximately the same number of words or phases from network traffic as source-level VAD algorithms extract from speech signals.
- **Average Length of ON Periods**: An ON period indicates a continuous burst of speech. We use the ratio between the average length of true ON periods and that of estimated ON periods to evaluate whether VAD can correctly infer speech bursts. Let the mean length of the estimated ON periods be M_{est} and the mean length of true ON periods be M_{true}. Then, the accuracy metric can be defined as
$$\frac{M_{est}}{M_{true}}.$$
As shown in the middle graph of Fig. 4, the accuracy with respect to the length of ON periods is around 1 for most calls. The result suggests that the inferred speech bursts, normally generated by whole sentences or word phases, are approximately identical to the real speech activity.
- **State Correctness**: We use 1 to denote an ON period in a time unit and 0 to denote an OFF period. Assume that M is an ON/OFF sequence of speech activity in the 0-1 representation, and N is the estimated speech activity. We then compute the state correctness $\hat{\alpha}$ as
$$\frac{|M \cap N|}{|M \cup N|}.$$
The right graph of Fig. 4 shows the distribution of $\hat{\alpha}$. An $\hat{\alpha}$ value approaching 100% indicates that the estimated ON/OFF periods are nearly the same as the true ON/OFF periods. For most calls in our traces, $\hat{\alpha}$ ranges between 70% and 90%, with the median around 85%. As network-level VAD applications are mainly concerned about the presence of speech bursts and silence periods, rather than the exact state in each time unit (0.1 second in this study), we consider that the correctness rate reasonably accurate for most calls.

We believe that the detection accuracy of our method is quite good given the high amount of randomness injected into network traffic by the Skype application and network dynamics.

VI. Conclusion

In this paper, we propose the concept of network-level VAD, which infers speech activity from network traffic instead of audio signals, which are used in source-level VAD. Extracting voice activity from network traffic is more difficult because VoIP traffic can be seen as a compressed audio signal with additional randomness injected, such as redundancy, network congestion, and retransmissions. Besides proposing network-level VAD, we propose a VAD algorithm that can extract voice activity from encrypted and non-silence-suppressed VoIP network traffic. Taking Skype as the subject of our study, we show that inferring voice activity from Skype traffic achieves a reasonable level of detection accuracy even there has been high degree of randomness in the network traffic.

References

[1] "Fine - tuning voice over packet services," http://www.protocols.com/papers/voip2.htm.
[2] I. S. 802.16-2004, "Ieee standard for local and metropolitan area networks part 16: Air interface for fixed broadband wireless access systems," Oct 2004.
[3] 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.
[4] B. Francesco, C. Salvatore, and C. Alfredo, "A robust voice activity detector for wireless communications using soft computing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, 1998.
[5] J. Hoyt and H. Wechsler, "Detection of human speech in structured noise," in *Proceedings of ICASSP '94*, vol. ii. ACM Press, 1994, pp. 237–240.
[6] J.-S. R. Jang, "Audio signal processing and recognition," http://www.cs.nthu.edu.tw/ jang.
[7] V. Prasad, M. R., S. Vijay, H. Shankar, P. Pawelczak, and I. Niemegeers, "Voice activity detection for voip-an information theoretic approach," in *Proceedings of GLOBECOM '06*, vol. ii. ACM Press, 2006.