DOI: 10.11591/ijeecs.v5.i1.pp 11-18

Packet Delay Distribution Model for Investigating Delay of Network Speech Recognition

Asril Jarin*[1], Suryadi[2], Kalamullah Ramli*[3]
[1,3]Department of Electrical Engineering, Faculty of Engineering, Universitas Indonesia
[2]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia
Depok 16424, Indonesia
[1]Center for Information and Communication Technology, BPPT, PUSPIPTEK, Serpong, Tangerang Selatan 15314, Indonesia
*Corresponding author, e-mail: asril21@ui.ac.id[1], kalamullah.ramli@ui.ac.id[3]

Abstract

Unlike multimedia streaming applications that require a smooth playback at the client, application of network speech recognition (NSR) that recognizes speech signal in a sentence-by-sentence manner might tolerate an acceptable delay. The acceptable delay is a user-defined time in which the entire sentence data should be received by the server. We proposed a calculation method to investigate the acceptable delay of network speech recognition that employs a speech segmenter to send speech signal sentence-by-sentence over TCP channel to the server. The calculation multiplies the mean packet delay of TCP flow at steady-state with the number of created packets. For validation we implemented a MATLAB program and solved it using 2500 Indonesian speech sentences. The results were then compared with the results of our previous model that used a transient analysis method. It was found that this calculation method is not appropriate due to the transient behavior of the streaming sentences.

Keywords: *network speech recognition, packet delay distribution, TCP delay performance*

Copyright © 2017 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

The speech recognition technology has become one of cloud services on Internet, such as Google Speech Recognition and Apple Siri [1]. This has been supported by growing provision of bandwidth enabled by technology advances in network, such as fiber- and 4G mobile-broadband. However, in some network conditions, there is still possibility that low throughput delay the data delivery, especially for applications that use the transmission control protocol (TCP).

In the case of network speech recognition (NSR) [2] using TCP, the speech transmission will experience delays caused by two main mechanisms of TCP against network congestion, i.e. Additive Increase Multiplicative Decrease and Timeout [3]. Nevertheless some real-time applications consider the delayed packets as the lost packets. Both packet loss and packet delay contributes to the degradation of the application performance. We address a problem of continuous speech recognition application on Internet using TCP by considering that speech sentence is based on a language model which has the rules of grammar. Hence, this kind of system requires to receive the speech sentence entirely. For that purpose, a scheme of speech recognition with a speech segmenter at the client side has already been proposed by our previous work [4]. The speech segmenter cuts the speech signal into sentences based on pause marked by a silent signal in a zero-crossing threshold and then sends through the TCP channel to the server where the speech recognition is performed.

Unlike the multimedia streaming application that requires a constant speed for a smooth playback [5] or VoIP that require high interactivity, a speech recognition application has an automatic speech recognition engines (ASR) that works depending on the availability of data. The faster the data is available the faster the ASR works. On the other hand, TCP might cause delay beyond the limits of an acceptable delay. The acceptable delay is a user-perceived time in which the application can meet its satisfactory performance. Our previous work [4] analytically investigated acceptable delay using stored streaming method and identified a working region

Received August 2, 2016; Revised October 15, 2016; Accepted November 1, 2016

which can lead satisfactory performance of application for a distribution of speech sentence from one up to twenty-two seconds length.

In this paper, we proposed a method in investigating the acceptable delay of a TCP-based speech recognition by multiplying a mean packet delay with the number of packets created from data. The mean packet delay is obtained from a discrete-time Markov model of Brosh [6] at the steady-state condition. Our hypothesis stated that the result of this calculation is equal to the end-to-end delay of the entire data streamed using TCP. In order to examine the hypothesis, a MATLAB program was implemented to solve the TCP delay of the same distribution of utterance length as in our previous work. The calculation results were eventually compared with the results of our previous work [4] that used a method of transient analysis. We have proved that the calculation was not valid and appropriate in investigating the delay of the streaming via TCP in a sentence-by-sentence manner. The speech sentences distributed in the range of one to twenty-two seconds need a short period in which the steady-state has never been able to reach.

The remainder of this paper is organized as follows. Section 2 presents our work methodology including a problem setting, a review of the model of delay distribution of a real-time TCP flow [6], an implementation in a MATLAB program, and an experimentation. Section 3 discusses the experimentation results. Finally, we make concluding remarks in section 4.

2. Research Method

2.1. Problem Setting

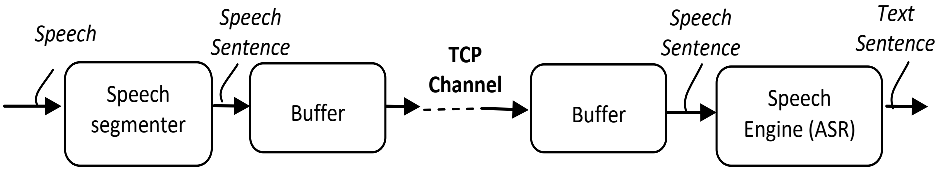

Figure 1. Scheme of Sentence Streaming Over TCP

In this work, a proposed research problem comes from a scheme of speech recognition system that receives the speech signal in a sentence-by-sentence manner with the help of a speech segmenter at the client, as shown in Figure 1. The speech sentences are sent through TCP channel to the server where recognition is carried out by an automatic speech recognition (ASR). To gain a satisfactory performance in this scheme, the sentence data should have been received by ASR within an acceptable delay. Since this scheme is addressed to an Internet service of automatic meeting transcription in Bahasa Indonesia, we used 2500 Indonesian spoken sentences that we obtained from a speech in constitutional court. Their lengths are distributed from one to twenty-two seconds, as shown in Figure 3. The sentences are encoded at a constant sampling rate of 16 kHz and 16 bits per sample, that is, one-second sentence has 32 kB. Thus, a range of speech sentences is in between 32 kB and 704 kB.

In order to investigate whether all speech sentences can be delivered within an acceptable delay calculation based on a discrete-time Markov model of the delay distribution of a real-time TCP flow [6] is used. We assumed that the speech recognition is a continuous application and the end-to-end delay of its TCP flow could be obtained from a multiplication between the mean packet delay at the steady-state and the number of packets of data.

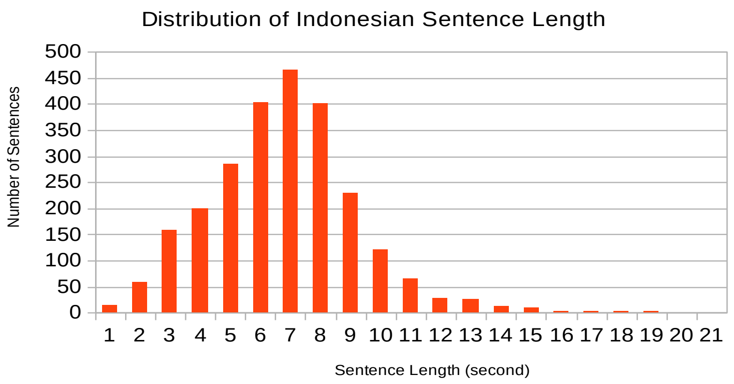

Figure 2. Length Distribution of Indonesian Sentences

2.2. Model of Delay Distribution [6]

The model of delay distribution is a discrete-time Markov model used to investigate analytically the delay-friendliness of TCP for real-time media applications that deliver data timely and continuous data, such as VoIP and live video streaming. A common difference between VoIP and live video streaming is their tolerance of end-to-end delay and their packet sizes. VoIP often uses small payload (e.g.160-bytes packets) and a low delay of no more than 400 ms. Live video streaming can send bigger packets to accelerate the streaming and afford a few seconds of start-up delay for smoothing its playback.

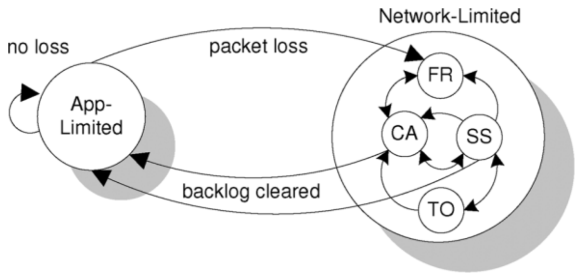

Figure 3. Model for TCP connection with a CBR source [6]

Additionally, the model of delay distribution emphasizes on the dynamics of TCP buffer, i.e. send buffer and receive buffer, to predict the delay performance of TCP. With assumptions that the sender uses TCP New Reno [3, 7] and CBR source, and the average throughput provided by TCP satisfies the data generation, the model has two main states: *application-limited* and *network-limited* as shown in Figure 3. The sender alternates between application-limited and network-limited periods. In a network limited period, the state moves between several congestion control algorithms: slow-start (SS), congestion avoidance (CA), fast retransmission (FR), and timeout (TO).

In the model, CBR source is characterized by two parameters: the data generation rate in packets per second, f and the size of a generated packet in bytes, a. Data generation rate in packets per round-trip time is denoted by r. The behavior of TCP is described by a discrete-time Markov chain with a finite state space $S = \{(w,b,l)\}$ and a probability transition matrix $Q = [q_{s;s'}]$, $s,s' \in S$. Each state represents a triple, (w,b,l), where w is the current congestion window size in segments, b is the current backlog size in bytes, and l indicates whether a loss has been detected and data need to be recovered.

Overall, all definition of state conditions are listed in Table 1. Whereas the definition of state transition probabilities are found in [6].

Table 1. State Classification

Classification	Condition
AL (Application-limited)	$r \le w$, $b = 0$, $l = 0$
NL (Network-limited)	$0 \le w$, $b \neq 0$ or $w < r$, $b = 0$
CA (Congestion avoidance)	$r/2 < w$, $l = 0$
SS (Slow start)	$w \le r/2$, $l = 0$
FR (Fast recovery)	$0 < w$, $l = 1$
TO (Timeout)	$w = 0$, $l = 1, \dots ,6$

For our solutions, the model provides us these following calculations:

a) TCP delay of the i-th packet sent in a transition from state s to state s' , $d_{s;s'}$:

$$d_{s;s'}^{(i)} = L + \begin{cases} b/(fa) + RTT + (3+i)/f, & \text{if } s' \in FR \\ b/(fa) & \text{otherwise} \end{cases} \tag{1}$$

where L is the one-way sender-to-receiver network delay.

b) The number of source packets sent in a transition from s to s', $n_{s;s'}$:

$$n_{s;s'} = \begin{cases} 1, & \text{if } s \in AL,\ s' \in AL \\ \lfloor \min(b, wMSS)/a \rfloor, & \text{if } s' \in \{CA|AL\} \\ \lfloor \min(b, (w+3)MSS)/a \rfloor, & \text{if } s' \in FR \\ 0, & \text{if } s' \in TO \end{cases} \tag{2}$$

In the application-limited state, the model evolves at the packet-level granularity, so that there is a single packet sent in a loss-free transition from an application-limited state, as shown by the first item of (2).

c) Steady-state delay distribution of a TCP connection with a CBR source, D, over some finite interval A:

$$P(D=d) = \lim_{t \to \infty} \frac{N_t(d)}{N_t}, \forall d \in A \tag{3}$$

where N_t is the number of packets successfully sent in some time interval $[0,t]$ and $N(d)$ is the number of packets out of N_t that experience delay d.

Using renewal theory (ref), the steady-state delay distribution is calculated by

$$P(D=d) = \frac{\sum_{s \in S} \pi_s \sum_{s' \in S} q_{s;s'} \sum_{i=1}^{n_{s;s'}} I_{d_{s;s'}=d}}{\sum_{s \in S} \pi_s \sum q_{s;s'} n_{s;s'}} \forall d \in A \tag{4}$$

where π_S is the stationary distribution of the Markov chain for the TCP source, I is the indicator function, π_S is the steady-state distribution of the chain, while $d_{s;s'}$ and $n_{s;s'}$ are given in (1) and (2), respectively.

In order to solve the Markov chain of the model, the size of the state space are limited by the maximum value of congestion window size, w_m and backlog size, b_m, respectively.

2.3. Implementing Model in MATLAB

In this session, we implemented the calculation model using a MATLAB program and derived a mean packet delay at the steady-state. Figure 4 shows us a flowchart which has several stages to gain the mean packet delay of the steady-state distribution. In the first stage, some variables are declared and initialized according to the experimental parameters, such as maximum segment size (MSS), round-trip-time (RTT), loss rate (p), constant bit rate (cbr), size of a generated packet (a), data generation rate in packet per second (f), data generation rate in packet per round-trip time (r) etc. The second stage defines the state space as specified by

Table 1. The size of the state space is limited by maximum values of congestion window (w_{max}), backlog (b_{max}) and back-off timeout (l_{max}). The third stage generates a Markov chain matrix using data transition probabilities, which are found in Appendix of [6]. Furthermore, the number of CBR packets and TCP delay of the i-th packet sent in each state transition can be subsequently calculated. Stationary distribution of the Markov chain, π_s, is obtained by finding the eigenvalue and eigenvector of the matrix. Finally, the steady-state delay distribution of a TCP flow is computed for some finite interval A. Based on steady-state delay distribution, the mean delay of the distribution, $E[D]$, is obtained and used further to investigate the acceptable delay of the speech recognition.

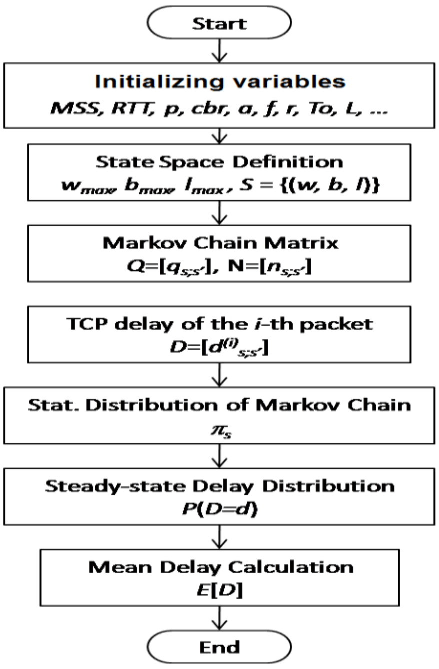
Start
Initializing variables
MSS, RTT, p, cbr, a, f, r, To, L, ...
State Space Definition
w_{max}, b_{max}, l_{max}, $S = \{(w, b, l)\}$
Markov Chain Matrix
$Q=[q_{s;s'}]$, $N=[n_{s;s'}]$
TCP delay of the i–th packet
$D=[d^{(i)}_{s;s'}]$
Stat. Distribution of Markov Chain
π_s
Steady-state Delay Distribution
$P(D=d)$
Mean Delay Calculation
$E[D]$
End

Figure 4. A Flowchart for Calculation of TCP Delay

2.4. Experimentation

In order to examine whether the calculation is valid and appropriate for our model: (i) we run the MATLAB program under several parameter settings to obtain the mean packet delays at the steady-state; (ii) we collected data of mean values in a spread sheet and multiplied with the number of packets depending on the sentence length to obtain the end-to-end delay of each streaming sentence; and finally (iii), we verified the results based on real-time-factor, working region and made comparison with the results of our previous work [4].

The calculation parameters were set based on packet size of {640 bytes, 1200 bytes}, loss rate of {0.001 sec, 0.005 sec, 0.01 sec, 0.02 sec, 0.025 sec, 0.05 sec}, round trip time (RTT) of {100 ms and 200 ms} and expected acceptable delay of {4 sec, 6 sec, 8 sec, 10 sec, 12 sec}. The constant parameters, such as maximum segment size (*MSS*) and source rate (*cbr*) were set 1500 bytes and of 32 Kbps, respectively. While, the maximum values of congestion window (w_{max}), backlog (b_{max}) and back-off timeout (l_{max}) were set as recommended by Brosh [6].

In this experiment, the end-to-end delays of each streaming sentences were calculated for a length distribution of 2500 speech sentences in Bahasa Indonesia, i.e.{1 sec,.., 22 sec}. The real time factor and the working region were examined. In analyzing, the experiment results were compared with the results of our previous work [4] that used a discrete-time Markov model based on Padhye [8] and Figure [9], and also a method of transient analysis. The measurements of the transient analysis are obtained using a modeling tool of TANGRAM-II [10] that applied a reward calculation, such as rate reward and impulse reward, to solve the model [11, 12].

3. Results and Discussion

To investigate the acceptable delay against the streaming of Indonesian sentences we need a calculation of delay difference between the acceptable delay and TCP delay of the streaming sentences. The difference is negative when TCP delay of the sentence is bigger than its acceptable delay. A negative difference indicates a tardines of the streaming sentences against its acceptable delay. Meanwhile, TCP delay is obtained from the multiplication between mean packet delay at the steady-state and the number of sentence packets.

Figure 5a shows the investigation results of five samples of the acceptable delays, i.e. 4 sec, 6 sec, 8 sec, 10 sec, and 12 sec, against one- to eight-second sentences with packet size of 1200 bytes, loss rate of 0.014 and RTT of 100 ms. All designated number specified for the acceptable delay in this work can not meet the user acceptable limit except for the 1-second and 2-second sentences. Meanwhile, the results of our previous work [4] that use transient analysis, as shown in Figure 5b, indicated that the acceptable delay of 8 seconds is not longer fulfilled in a sentence with a length of 16 seconds.

Regarding the timeliness of speech recognition, Platek [13] determined that the performance of speech decoding speed is measured with a Real Time Factor (RTF). RTF is the ratio of the decoding time of a speech recognizer to the length of pronunciation. RTF indicates how quickly automatic speech recognition decodes the speech signal. In this work, we define the real-time-factor of the network speech recognition using TCP is a comparison between the end-to-end delay of the mean sentence against a specified acceptable delay. Based on the results of this experiment, the real-time-factor is bigger than one. Hence, the timeliness or the satisfactory performance is not fulfilled at packet size of 1200 bytes with loss rate of 0.014 and RTT of 100 ms.

In Figure 6, there is a very significant difference in the comparison of TCP delay on a working region. The working region is identified as a function of loss rate and packet size. TCP delay with packet size of 640 bytes is greater than with packet size of 1200 bytes. This is caused by the aggregation of all packet delays in streaming sentences, although the smaller packets are relatively faster in delivery.

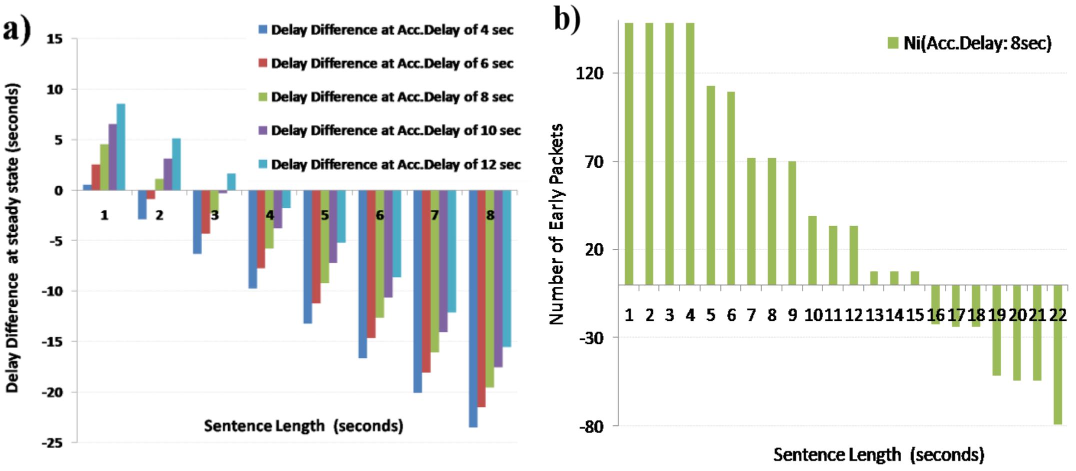

Figure 5. Acceptable Delay Investigation using: a) Packet Delay Distribution, b) Transient Analysis

Comparing the results of this work and our previous study [4] the difference of TCP delay is large (Figure 6). The reasons are threefolds: First, the steady-state analysis is not appropriate for a short-term or intermittent process. The length of our sample sentences is not more than 22 seconds. If the streaming of the longest sentence needs not more than 22 seconds and the round-trip-time is constant at a value of 100 ms, then the number of state transitions would be not more than 220. This is too small compared to the state space at the maximum value of the state variables, which has more than two thousands states. Therefore, the steady-state would not be achieved in this application; Second, the use of two different TCP base model. This calculation was modeled based on an assumption that the sender is using TCP NewReno [7], while our previous work used stochastic model of TCP Reno [8, 9]. TCP NewReno is an implementation of TCP that modifies TCP Reno in order to make it faster and more efficient, such as the detection of multiple packet losses in a congestion window; Third, packet delay distribution is obtained using two levels of granularity, i.e. packet granularity in the application-limited period and window granularity in the network-limited period. Hence, we argue that the calculation of packet delay distribution at the steady-state is not accurate to investigate the acceptable delay. However, when we notice the slow trend of TCP delays based on the increase of loss rate at the specified packet size and RTT, the calculation could qualitatively be an estimation of the tardiness without making the loss rate as a parameter.

Concerning the phenomenon in Figure 7, the increase in TCP delay caused by the increase in the round-trip-time is relatively larger than by the increase in the loss rate. The increase in TCP delay is also shown by the difference in the packet size. The smaller packet size on a larger RTT the more slowly TCP deliver its data.

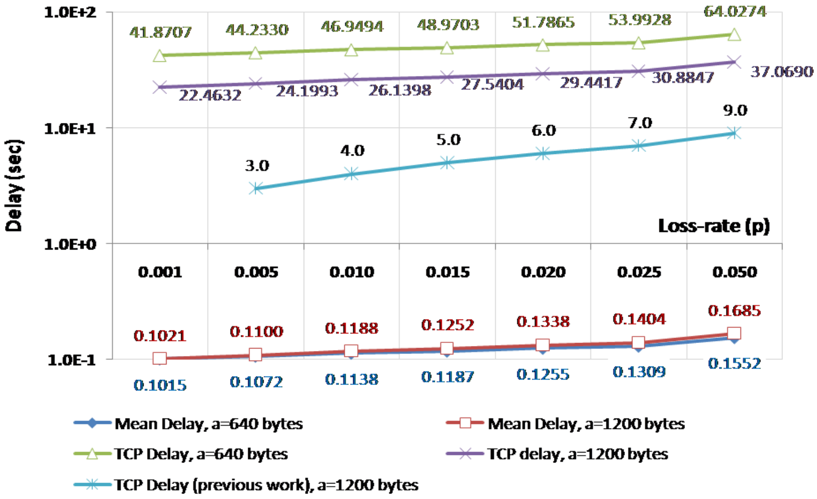

Figure 6. Mean Packet Delay and TCP delay of 8-second Sentence with RTT=100 ms

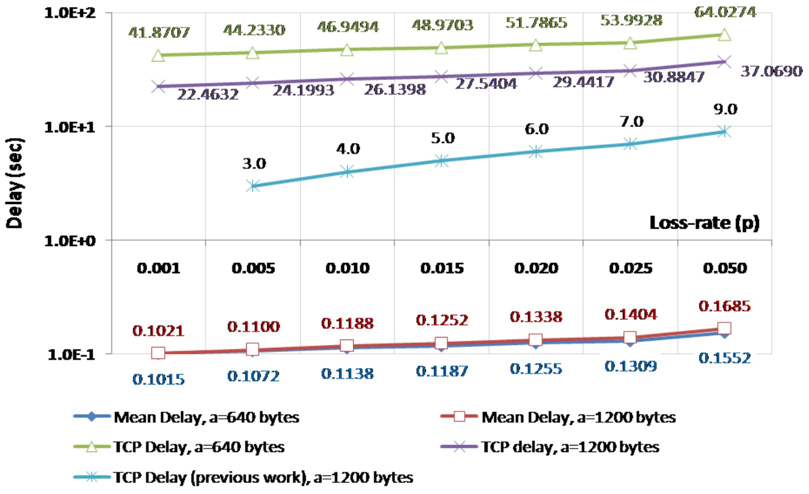

Figure 7. TCP Delay of 8-second Sentence

Overall, we conclude that the calculation method at the steady-state applied in analyzing the non-continuous system such as the speech recognition using a speech segmenter at the client that transmits data intermittently in a sentence by sentence manner is not appropriate. Thereby, our hypothesis that we pointed in the introduction of this paper is not valid.

4. Conclusion

This paper proposed a calculation method to investigate the acceptable delay of network speech recognition that places a speech segmenter for sending speech signal to the server in a sentence-by-sentence manner over TCP channel. This method refers to a discrete-time Markov Model of packet delay distribution of TCP flow at the steady-state. The mean delay was multiplied with the sum of data packets in order to obtain the end-to-end delay of TCP. Through experimentations using MATLAB program, the model was realized. The results were then compared with the results of our previous work [4] by examining the real time factor and the working region of the application.

It is concluded that this calculation is not appropriate in investigating the acceptable delay of streaming sentences due to the transient behavior of the streaming sentences. The streaming of speech recognition using a speech segmenter is not continuous and the distribution of sentence-lengths only ranges from one second to twenty-two seconds, which is too short for the steady-state condition to achieve.

Provide a statement that what is expected, as stated in the "Introduction" chapter can ultimately result in "Results and Discussion" chapter, so there is compatibility. Moreover, it can also be added the prospect of the development of research results and application prospects of further studies into the next (based on result and discussion).

Acknowledgments

The work is partly supported by the research grant of PITTA 2016 of Universitas Indonesia (Contract No. 1753/UN2.R12/PPM.00.00/2016).

References

[1] Assefi M, Wittie M, Knight A. *Impact of Network Performance on Cloud Speech Recognition*. International Conference on Computer Communication and Networks (ICCCN). Las Vegas, Nevada, USA. 2015: 1-6.

[2] Peinado A, Segura JC. Speech recognition over digital channels robustness and standards. Chichester, England: John Wiley. 2006.

[3] Stevens WR, Wright GR. TCP/IP illustrated. Reading, Mass.: Addison-Wesley. 1994: 1-3.

[4] Jarin A, Fahmi H, Suryadi S, Ramli K. Development of Modified Analytical Model for Investigating Acceptable Delay of TCP-Based Speech Recognition. *Advanced Science Letters*. 2016;

[5] Wang B, Kurose J, Shenoy P, Towsley D. Multimedia streaming via TCP: An analytic performance study. *ACM Trans Multimedia Comput Commun Appl*. 2008; 4(2): 1-22.

[6] Brosh E, Baset SA, Misra V, Rubenstein D, Schulzrinne H. The Delay-Friendliness of TCP for Real-Time Traffic. *IEEE/ACM Transactions on Networking*. 2010; 18(5): 1478-91.

[7] Henderson T, Floyd S, Gurtov A, Nishida Y. *The NewReno Modification to TCP's Fast Recovery Algorithm*. RFC 6582. 2012.

[8] Padhye J, Firoiu V, Towsley DF, Kurose JF. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking*. 2000; 8(2): 133-45.

[9] Figueiredo DR, Liu B, Misra V, Towsley D. On the autocorrelation structure of TCP traffic. *Computer Networks*. 2002; 40(3): 339-61.

[10] de Souza e Silva E, Figueiredo DR, Leão RM. The TANGRAM II integrated modeling environment for computer systems and networks. *ACM SIGMETRICS Performance Evaluation Review.* 2009; 36(4): 64-9.

[11] Souza e Silva Ed, Richard Gail H. An algorithm to calculate transient distributions of cumulative rate and impulse based reward. *Communications in statistics Stochastic models*. 1998; 14(3): 509-36.

[12] de Souza e Silva E, Gail HR, Campos RV. Calculating transient distributions of cumulative reward: *ACM*; 1995.

[13] Platek O. Automatic Speech Recognition using KALDI. Master Thesis. Prague: Charles University; 2014.