Quality of Service Improvement for Voice Streaming over Wireless Ad-hoc Networks using an Adaptive Playout Adjustment Algorithm

Maral Salehi*, Mehdi Dehghan

Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran

Received 26 August 2009; revised 28 January 2010; accepted 9 February 2010

Abstract

Providing a high-quality service for transmission and playing real-time voice conversations (voice streaming) over wireless ad-hoc networks is no mean feat. Buffering together with adjusting the playout time of the packets is a receiver-side solution to overcome this challenge. In this paper, a new adaptive playout adjustment algorithm is proposed to stream the voice conversations over wireless ad-hoc networks. This algorithm always tries to be aware of the network's conditions, adapts itself with these conditions and adjusts the playout time of the voice packets as efficiently as possible. It is required that not only most of the packets be received before their playout time, as scheduled in the receiver, but also that the playout time not be too long so as to adversely affect the interactivity between the sender and the receiver. The main features of the presented method are: adjusting the threshold adaptively with respect to the varying conditions of the network in order to determine the state of system; calculating the mean network jitter dynamically based on the current conditions of the network in order to calculate the playout delay for the current packet; being optimistic about the future state of the network and not using the delay history in order to calculate the mean network delay. Simulation results show that the proposed algorithm adapts itself with the network's dynamics and adjusts the playout delay for voice packets better than the other algorithms.

Keywords: Wireless Mobile Ad-hoc Networks, Voice Streaming, Adaptive Playout Algorithm, Quality of Service, Multimedia.
.

1. Introduction

The A mobile ad-hoc network (MANET) is a collection of mobile nodes that can communicate with each other over radio in the absence of any infrastructure. In fact, inside a MANET all nodes act as routers and forward the received packets to nodes within their radio range. Indeed, mobile ad-hoc networks are characterized by the mobility of all nodes, a bandwidth-limited channel, an unreliable wireless transmission medium, etc. The MANET's properties render the provision of packet-based conversational applications such as packet-based voice conversation into a challenging task. Indeed, such applications require a bounded end-to-end delay and jitter, but can tolerate a limited packet loss ratio. The transmission of voice over packet-switched networks differs from that over conventional circuit-switched networks since in the former, each VoIP packet will experience variations in delay. This kind of jitter variation is an inherent characteristic of the packet-switched networks. To address this problem, a playout jitter buffer is used at the receiver side to hold the incoming voice packets for a short period of time (jitter buffer delay) to synchronize the packet stream before their scheduled playout time. Packets that are not received before their scheduled playout time are considered lost (jitter buffer packet loss). Improved synchronization quality will be gained when the jitter buffer scheduler imposes longer delay. However, this will also lead to the increased overall delay of the voice stream that may be perceived by the user. A simplified schematic for the transmission of voice packets over IP networks is shown in Figure 1.

From the network transport perspective, the VoIP speech quality is primarily impaired by packet loss, total end-to-end delay (mouth-to-ear delay) and the jitter. The loss of some packets will generate gaps in the continuous voice stream, resulting in degraded voice quality. The total packet loss includes network transmission loss due to network congestion and jitter buffer loss due to packet arrival latency. Various strategies may be employed to deal with

* Corresponding Author. Email: maralsalehi@aut.ac.ir

packet loss such as, repeating the last received packet, insertion of silence, insertion of noise and interpolation. The total packet end-to-end delay does not cause a reduction in voice quality but it affects the interactive nature of the conversations. A network propagation delay of less than 100ms is not noticeable to users, whereas delays over 200 ms begin to cause some degree of disruption during a conversation. The introduction of the VoIP playout jitter buffer affects both the total end-to-end delay time and the total packet loss rate. With a small jitter buffer size, a larger number of overdue packets are likely to be dropped, essentially increasing the jitter buffer packet loss rate which in turn gives rise to a higher total packet loss rate. When the jitter buffer size increases, fewer packets will be dropped due to arrival latency, resulting in an improved synchronization and thus higher voice quality. However, a large buffer size will increase the total end-to-end delay. The increase in both packet loss rate and the total end-to-end delay results in the degradation of the conversation quality. Finding the tradeoff between the total end-to-end delay and the total packet loss rate is the key issue in designing a VoIP playout jitter buffer scheduler. An efficient playout algorithm should be able to minimize the buffering delay and packet loss, thus improving the loss-delay tradeoff.

In this paper, we propose a new adaptive playout adjustment algorithm to stream the voice conversations over wireless ad-hoc networks. This algorithm always tries to be aware of the network's conditions, adapts itself with these conditions and adjusts the playout time for the voice packets as good as possible. The rest of the paper is structured as follows: Section 2 first introduces the current QoS improvement schemes for voice transmission over wireless networks and then classifies the existing playout adjustment algorithms designed for real-time packet-based voice conversation. Section 3 describes the proposed algorithm which has been designed specifically to play voice packets transferred over a MANET. The simulation scenarios and the performance results are discussed in Section 4. Section 5 concludes the paper.

2. Related Works

2.1. QoS Improvement Schemes for voice transmission over wireless IP networks

A wide assortment of techniques has been proposed for improving the QoS for voice transmission over wireless IP-based networks. The existing schemes can be classified into the following four categories: 1) Techniques for improving bandwidth utilization, such as: alternating between different voice CODECs, silence suppression and reducing the IP/UDP/RTP header overhead. 2) Mechanisms for improving the underlying network protocol efficiency for real-time voice packets, such as: giving higher priorities to the real-time voice packets, using admission control

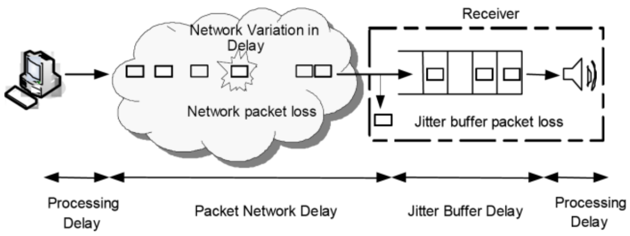

Fig. 1. Transmission of voice packets over an IP network [2].

schemes and improving the bandwidth and throughput for 802.11 wireless LAN. 3) Mechanisms for recovering and concealing the lost packets. 4) Algorithms for alleviating the packet jitter and for improving the playout quality. Our proposed solution in this paper is based on a similar perspective as that of the techniques in the fourth category. In the following section, we briefly survey the prior art in the context of our adopted strategy.

2.1.1. Techniques for Improving the Bandwidth Utilization

In order to come up with a satisfactory QoS level for the Internet telephony system under a limited communication bandwidth, researchers have proposed several ways to improve the bandwidth utilization efficiency.

- Different voice codecs

Some researchers have proposed methods for alternating between different voice codecs to achieve flow control and congestion control for the limited bandwidth situation [13],[14],[15]. When the bandwidth is limited and the data rate needs to be lowered, a higher compression rate codec will be adopted to increase the amount of compression. This will usually result in the degradation of the speech quality, but it is still better than having lots of lost packets and large delay. Given the current demand for high quality, it is clear that choosing the lowest bit rate codec will not suit a large proportion of today’s VoIP market. Therefore, most systems offer G.711 (PCM) and at least one low-bit-rate codec. This gives the operator some flexibility to establish a trade-off between the quality and bandwidth. Table 1 lists some of the standard codecs used in VoIP systems. Figure 2 shows the quality of speech passed through some of the standard codecs.

- Silence suppression

To further improve the bandwidth utilization of the Internet telephony system, a silence suppression method can be applied. Silence suppression is the process in which nothing is coded or transmitted during the periods the user is not speaking. By using the silence suppression method, the total required bandwidth can be reduced up to 50%. However, it is quite challenging to have the voice activity detector (VAD) detect the silent period accurately, while also keeping the algorithm fast enough for real-time communication.

Table 1
Current voice codec schemes for VoIP systems

Codes	Bit Rate (*Kbps*)	Sample Period	Payloud Size	Coding Technique
G.711	64	20 ms	160 bytes	Pulse Code Modulation
G.726	40 to 16	20 ms	80 bytes	Adaptive Different PCM (ADPCM)
G.728	16	10 ms	20 bytes	Low-Delay Code Excited Linear Prediction (LD-CELP)
G.729	8	10 ms	10 bytes	Algebraic Code Exited Linear Prediction (ACELP)
G.723.1	6.3/5.3	30 ms	24/20 bytes	Multi-Pulse Max Likelihood Quantization (MP-MLQ)/ACELP
GSM FR	13	20 ms	32.5 bytes	Regular Pulse-Excited Long-term Predictor (RPE-LPT)
GSM EFR	12.2	20 ms	30.5 bytes	Algebraic Code-Excited Linear Prediction (ACELP)

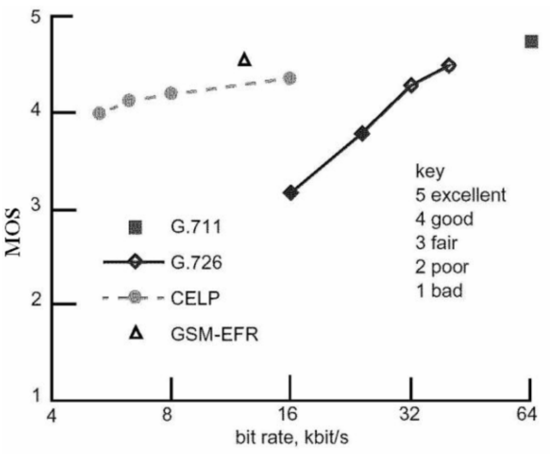

Fig. 2. Codec speech qualities under different bandwidths [13].

- The reduction of the IP/UDP/RTP header overhead

Voice data is transported over the Internet using RTP/UDP/IP packet encapsulation. The overhead induced by packet encapsulation is very large, rendering the process quite inefficient. For example, if we rely on G.729a codecs and use one RTP packet to carry 20 ms of compressed voice information in the payload, then the payload size is 20 bytes. However, the IP/UDP/RTP header itself will occupy 40 bytes, as shown in Figure 3.

IP Header 20 bytes	UDP Header 8 bytes	RTP Header 12 bytes	Voice Payload 20 bytes

Fig. 3. Typical RTP packet structure using G.729a in an IP network.

This is very inefficient since the IP header overhead takes almost 67% of the total required bandwidth. Thus, some researchers have proposed using a point-to-point RTP header compression technique [16] to reduce the IP/UDP/RTP header overhead of 40 bytes into 2 to 4 bytes to further improve the bandwidth utilization.

2.1.2. Mechanisms for Improving the Protocol Efficiency for Voice Packets

To improve the QoS for real-time voice packet delivery in the IP switched networks, it has been typically proposed to give higher priorities to the real-time voice packets so that they can be handled faster compared to other time-insensitive data packets. These priorities can be given at different layers of the network, such as Media Access Control (MAC) or the transport layer. The original media access control scheme of the current 802.11 wireless LAN protocol has been designed with two modes of communication for wireless stations [17]: the Distributed Coordination Function (DCF), the Point Coordination Function (PCF). Neither DCF nor PCF differentiates between traffic types or sources. Thus, the original 802.11 MAC protocol cannot support QoS, and is thus, quite inefficient and has difficulty achieving the service quality demanded by real-time VoIP in which voice and data traffic coexist [6]. In order to enhance the current 802.11 MAC protocol in terms of QoS, two different schemes in line with the two different coordination modes of the original 802.11 MAC protocol have been proposed; viz. Enhanced Distribution Coordination Function (EDCF) and Hybrid Coordination Function (HCF).

Besides quality of services, to improve the access fairness and efficiency, an admission control scheme is an integral part of the solution. The admission control works as follows. When an additional real-time packet stream requests for a higher-priority QoS service, the admission control module will check the Basic Service Set (BSS)'s current available resource budget. If the budget is enough and provisioning for the requesting real-time stream will not affect the current real-time high-priority sessions, it will approve the request. If the BSS lacks enough resources, it will reject the request, and simply treats the requested real-time packet stream as a regular data stream. In this way, the QoS of the current high-priority real-time stream will not be affected and will still have very good quality. Thus, by using the admission control, we can protect the high-priority traffic from the competing.

Another option to provide better QoS for real-time applications over wireless LAN is to improve the bandwidth and throughput. In late 2003, IEEE started to organize the 802.11n task group to develop a new standard for the next generation of wireless LANs, which is expected to achieve throughputs over 100 Mbps. Table 2 lists some important data on the IEEE 802.11 series wireless LAN.

Table 2
Specifications of the IEEE 802.11 series wireless LAN

IEEE WLAN Standard	Over-the-Air (OTA) Estimates	Media Access Control Layer, Service Access Point (MAC SAP) Estimates
802.11b	11 Mbps	5 Mbps
802.11g	54 Mbps	25 Mbps (when 11b is not present)
802.11a	54 Mbps	25 Mbps
802.11n	600 Mbps	300 Mbps

2.1.3. Mechanisms for the Recovery and Concealment of Lost Packets

The voice packets transferred in the IP network may become lost due to the following reasons: overflow of the queuing buffer during competing media accesses, drop-outs during the congestion period in the routers along the path, late arrivals past the scheduled play-out time, etc. The lost packets induce a significant impairment to the perceived voice quality by the end user. To recover or repair the lost packets, many real-time loss concealment schemes have been proposed to improve the quality of the VoIP system [9],[18].

Current wireless LANs offer sufficient bandwidth (at least 1 Mbps, and typically 10 Mbps or 54 Mbps), but they suffer from a higher packet-loss rate, fluctuating link status, sender collision avoidance buffering delays, etc. For real-time voice communication, delay is generally more of a concern compared to packet loss. In the WiFi MAC scheme, bandwidth is not a major concern, especially for real-time voice packets, but the packet drop rate, the media access contention, and the back-off retry times, which increase the overall delivery delay, may result in a serious problem.

Therefore, from the perspective of the entire system, in the wireless Wi-Fi environment, it may not be worthwhile to pursue low-bit-rate concealment schemes at the expense of extra computing complexity and delay; in effect, one would better off adopting a simple concealment scheme to minimize the loss concealment processing delay. For example, when the network drop rate is high, the system can simply duplicate the entire previous voice frame in every sent packet; when the network drop rate is low, on the hand, the system can simply use noise padding or interpolation techniques to conceal packet loss.

2.1.4. Algorithms for Packet Jitter Removal and Voice Playout Schemes

From the network transport perspective, VoIP speech quality is primarily impaired by packet loss, total end-to-end delay (mouth-to-ear delay) and the delay jitter. The loss of packets generates gaps in the continuous voice stream, resulting in degraded voice quality. The total packet loss includes network transmission loss due to network congestion and jitter buffer loss due to packet arrival latency. The total packet end-to-end delay does not cause a reduction in voice quality but it does affect the interactive nature of conversations. A network propagation delay of less than 100 ms is not noticeable to users, whereas delays over 200 ms begin to cause some degree of disruption during a conversation. The introduction of the VoIP playout jitter buffer affects both the total end-to-end delay time and the total packet loss rate. With a small jitter buffer size, more overdue packets are likely to be dropped, essentially increasing the jitter buffer packet loss which in turn contributes to the total packet loss rate. When the jitter buffer size increases, fewer packets are dropped due to arrival latency, resulting in improved voice quality. However, a large buffer size increases the total end-to-end delay. The increase in both packet loss rate and the total end-to-end delay results in the degradation of conversation quality. Finding the tradeoff between the total end-to-end delay and the total packet loss rate is the key issue for designing the VoIP playout jitter buffer scheduler. An efficient playout algorithm should be able to minimize both the buffering delay and packet loss, thus improving the loss-delay trade-off.

It is very common to witness bursts or spikes in a trace of network packet delays. A spike indicates a sudden and very large increase of delay experienced by packets in the network. The normal adaptive playout algorithms do not work well in the presence of such spikes. Thus, most of the current adaptive playout algorithms provide for some spike detection mechanism so as to be able to switch between a fast adaptation mode for spikes and a regular working mode for normal conditions.

2.2. Classification of Playout Adjustment Algorithms for Real-time Packet-based Voice Conversations

Real-time packet-based voice conversations require bounded end-to-end delay and delay jitter. To this end, a playout jitter buffer and a playout adjustment algorithm are used at the receiver side. As shown in Figure 4, the design strategy for the VoIP playout jitter buffer can be classified into two groups: static playout buffer design and adaptive jitter playout buffer design. The fixed-size jitter buffer playout algorithms are usually inadequate since they do not adapt to the dynamic network conditions. Thus, in practice, most current VoIP playout jitter buffer schedulers can be adaptive to network traffic changes to some extent. Adaptive playout algorithms dynamically adjust the playout delay throughout a conversation. Also, the normal adaptive playout algorithms do not work well in the presence of spikes. Thus, most of the current schemes provide for some form of a spike detection mechanism so as to be able to put up with spike-like conditions.

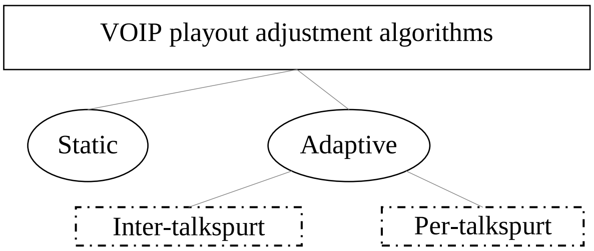

Fig. 4. Classification of playout adjustment algorithms.

Adaptive approaches, in turn, can be sub-classified into two categories: Inter –talkspurt adjustment algorithms and Per-talkspurt adjustment algorithms. In both categories, play-out adaptation is carried out according to the measured performance metrics, and with regards to the features of the played signal such as talkspurts in the case of voice conversations and type of frame in the case of video conversation.

2.2.1. Per-talkspurt Playout Adjustment Algorithms

Adaptive algorithms in this category calculate and adjust the playout delay only at the beginning of the talkspurts. The performance of the per-talk-spurt playout algorithms are tightly related to several configuration parameters. Indeed, the distribution of talk-spurts is strongly related to the CODEC used which typically includes a built-in VAD (voice activity detector) functionality. The VAD algorithm enables the detection of the talkspurt occurrences to efficiently schedule the packet transmissions and to assist the playout algorithm.

In [1], Ramjee et al. have proposed an adaptive playout jitter buffer algorithm which has recently become a standard adaptive playout mechanism for VoIP systems. This adaptive playout algorithm is based on Jacobson's work [3] on TCP round-trip time estimation (an exponentially weighted moving average estimator). This algorithm estimates two statistics: the packet delay and the delay variance, and then uses those two estimates to calculate the playout time:

$$T_p^i = T_s^i + \hat{T}_{net}^i + \beta \times \hat{V}^i, \quad (1)$$

$$\hat{T}_{net}^i = \alpha \hat{T}_{net}^{i-1} + (1-\alpha) T_{net}^i, \quad (2)$$

$$\hat{V}^i = \alpha \hat{V}^{i-1} + (1-\alpha) \left| T_{net}^i - \hat{T}_{net}^i \right|, \quad (3)$$

T_p^i and T_s^i represent, respectively, the playout and the sending time of the ith packet, and $\hat{T}_{net}^i$ is the weighted average of network delay upon the arrival of the ith packet. $\hat{V}^i$ is the mean delay variation which is updated upon receipt of each packet. The coefficients α and β are considered to be 0.99802 and 4 respectively. However, the practicality of Ramjee's adaptive playout algorithm is hindered by the sensitivity of the performance to the proper tuning of the α parameter.

In [5], Narbutt et al. have conducted a series of experiments with different values of α under different network conditions and have found out that it is not feasible to tune the α parameter to an optimal value so that the adaptive algorithm works well for all network conditions. Thus, Narbutt have proposed a dynamic α playout algorithm based on Ramjee's scheme. The selection of α is based on an empirical function derived from a series of experiments on a large set of network traces. In the dynamic α algorithm, α is dynamically adjusted according to the observed delay variations. When these variations are high, the parameter is set low, and vice-versa. The performance evaluation results for this extension reveal that the adaptive playout algorithm enables achieving a more desirable loss-delay trade-off compared to the conventional playout algorithms.

Besides Ramjee's standard adaptive playout algorithm and its variants which draw on linear recursive filtering for the estimation of packet delays, other researchers have instead proposed algorithms based on the packet-delay histogram. In [6], Moon et al. calculate a given percentile point q in the distribution function of the packet delays for the latest w packets, and then use it as the playout delay for the new talkspurt. The advantage of this histogram-based estimation approach is that the user can specify the acceptable packet loss rate ε, and the algorithm automatically adjusts the delay accordingly. Thus, the trade-off between the buffering delay and the loss rate associated with late arrivals can be controlled explicitly. However, the histogram-based estimation can only provide the distribution information and would not reveal much on the dynamic relationship between the consecutive packet delays. That is to say, the histogram-based estimation treats the entire history of packet delays indiscriminately, without considering that the latest packet delay may be more correlated with the next packet delay. This may result in a slow adaptation to the dynamic network conditions, especially for the non-stationary wireless Internet environment.

Indeed, all Per-talkspurt playout adjustment algorithms adjust the jitter buffer size only at the beginning of the talkspurts. With this type of "per-talkspurt" mechanism, any variation in the playout delay will introduce artificially compressed or expanded silent periods between consecutive talkspurts. The efficacy of the per-talkspurt mechanism is limited when talkspurts are long and the network delay variation is high. In order to overcome these problems, inter-talkspurt playout algorithms have been proposed.

2.2.2. Inter-talkspurt Playout Adjustment Algorithms

The inter-talkspurt algorithms calculate and adjust the playout delay not only at the beginning of the talkspurts, but also throughout the entire talkspurt for each receiving packet. In this way, the algorithm can be more reactive to the changes in network conditions, achieving a better tradeoff between the buffering delay and the loss rate due to late arrival.

In [7], Liang et al. have proposed an adaptive playout algorithm which can adjust the jitter buffer size within a talkspurt. The proper reconstruction of the continuous playout speech is achieved by scaling individual voice packets using a time-scale modification technique based on the Waveform Similarity Overlap-Add (WSOLA) algorithm. Their subjective listening experiments have shown that the voice packets could be scaled from 50 percent to 200 percent of the original size without degrading the sound quality. In Liang's adaptive playout scheme, the packet delay prediction was also based on a histogram approach.

In [2], Liu has proposed a novel playout buffer adjustment algorithm based on the dynamic estimation of network delays using time series modeling techniques. One of the major contributions of this algorithm is the adoption of time-series analysis methods to model and forecast the dynamics of the non-stationary WLAN end-to-end packet delay and the jitter series. This algorithm provides an explicit relationship between the packet loss rate due to late arrival and the playout buffering delay. The algorithm can effectively come up with a minimum playout buffer delay, while also guaranteeing that the packet loss rate will be below a certain level.

In [8], Jelassi has proposed an adaptive playout algorithm which adjusts the playout delay according to node mobility. This algorithm operates in two possible modes: During the occurrence of a path loss due to mobility, the playout delay is calculated on a per-packet basis, while aiming to maximize the perceptual quality insofar as possible. However, during normal periods, it plays the voice packets according to a baseline per-talk-spurt playout algorithm. One of the advantages of this algorithm is that it allows for gaps below 80 ms caused by delayed packets. When the gap is above 80 ms, however, it will play the late packet and the playout delay is increased consequently. The introduced delay during the last talk-spurt is then reduced in the next silence period. We have leveraged a similar initiative in our proposed design.

Besides its advantages that allow for a smooth operation over MANETs, Jelassi's algorithm is also associated with a number of weaknesses. Although it calculates the network mean delay using Ramjee's third algorithm [1], but the calculation is actually done without considering whether or not the packet of interest is received on time and before its playout time. Given that an erroneously calculated network mean delay is then used to determine the packet's playout delay, Jelassi's algorithm might not be able to correctly adjust the playout delay for packets. In addition, this algorithm uses (3) to calculate the network mean jitter and considers a fixed value for α. This is while a static α is not a reasonable choice for the calculation of the mean jitter in ad-hoc networks that are characterized by the mobility of nodes and variable network condition. Jelassi's algorithm also uses a constant threshold to determine the amount of network activity, which is not changed in response to the variable conditions. In sum, Jelassi's algorithm does not perform optimally and is unable to properly keep up with the network's dynamics. The proposed algorithm in the next section is intended to address these problems.

3. The Proposed Algorithm

In this paper, a novel algorithm is proposed to manage the playout buffer and adjust the playout delay associated with the voice packets belonging to a real-time voice conversation. This algorithm monitors the network condition and behaves accordingly to the current state of the network. To this end, the quality of service parameters like jitter will be under constant examination so that the algorithm is able adapt to the current state of the network and adjust the packet's playout time as efficiently as possible.

The key QoS parameters include network delay, packet loss, network jitter and the ratio of out-of-order packets. In order to determine which of these parameters best reflect the impact of node mobility and of the network condition, it is mandatory to study the effect of mobility on these parameters in an ad-hoc network. To serve this purpose, we have set up a simulated ad-hoc network which is discussed later in the subsequent section.

Figure 5 and Figure 6 show the cumulative distribution functions (CDF) derived from both the observed network delays and the experienced network jitters corresponding to all the ten simulated voice conversations over the ad-hoc network. *CDF(x)* represents the percentile of the received packets with a network delay (and also network jitter) of less than x.

As can be seen in Figure 5, a large percentile of the received packets (70%) experience a very low network delay (10 milliseconds). Also, Figure 6 shows that only a small percentile of the received packets (30%) are associated with a very low network jitter (10 ms). It is of note that the delay variations plotted in this figure have been calculated according to $jitter_i = |delay_i| - meandelay_i$ for each voice packet, where $jitter_i$, $delay_i$ and $meandelay_i$ represent the network delay variation, network delay and mean network delay for the *i*th packet, respectively. It can thus be inferred that the network jitter is a more pertinent metric to mirror the impact of node mobility.

On the other hand, as argued in [8], the ratio of out-of-order packets is highly sensitive to the occurrence of path switching. Therefore, the packet out-of-order parameter is considered as a metric to reflect the impact of node mobility in this work.

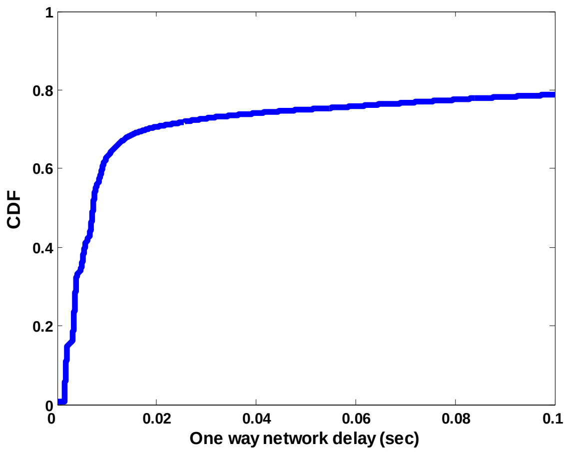

Fig. 5. The cumulative distribution function of the network delays orresponding to all the ten voice conversations over the ad-hoc network.

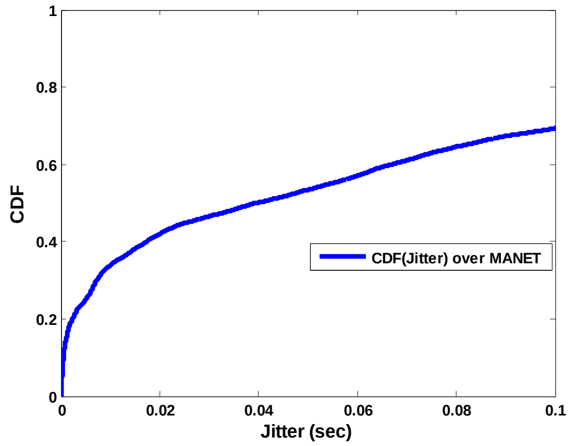

Fig. 6.The cumulative distribution function of the network jitters corresponding to all the ten voice conversations over the ad-hoc network.

Given that the node mobility and the variant network topology are the main characteristics of the wireless ad-hoc networks, with the jitter and packet out-of-order happening to be the most reflective metrics of network condition, our design leverages these metrics to determine the network's current state. The proposed algorithm runs in the active mode when both the jitter (inter packet delay difference: IDD) and the packet out-of-order (POR) values are relatively high. IDD is calculated as follows:

$$IDD = (T_a^{i+1} - T_s^{i+1}) - (T_a^{i} - T_s^{i}) \quad (4)$$

T_s^i and T_a^i represent the sending time and the arrival time of the *i*th packet, respectively. POR is calculated based on the packet's sequence number. A packet is considered out-of-order if it reaches the source with a sequence number p_i smaller than p_{max}. Otherwise, the value of p_i is assigned to p_{max}. The p_{max} represents the maximum sequence number received so far. The algorithm works in normal mode in the absence of out-of-order packets.

Following the calculation of the IDD and POR metrics, the state of the network should be detected. It is thus necessary to determine whether the sampled values are high or not. For this purpose, much in the same way as discussed in [9], the relative sample deviation (RSD) method is applied. RSD is a statistical algorithm used to judge the level of a sample value relative to the recent history of records. The RSD algorithm returns a value between 0 and 1 which can be used to classify the level of a sample as HIGH or not HIGH. RSD works as follows: Assuming sample values of x in the range [0, R], RSD divides the range R into N intervals $I_1, I_2, \ldots, I_N$ where the interval I_i holds sample values within [(i - 1)R/N, iR/N). Denote the total number of samples as S and the number of samples within interval I_i as $s(I_i)$. Given x, its corresponding interval is $I_x = x/[R/N] + 1$ [8]. To decide how HIGH x is, RSD calculates the ratio of sample values below x to the total number of samples:

$$RSD(x) = \sum_{i=1}^{x} s(I_i) / S \quad (5)$$

This corresponds to the CDF value at x. Given RSD(x), we can tell what percentage of sample values is lower than x. An RSD value close to one implies that x is HIGH with respect to the history records.

RSD(IDD) and RSD(POR) should be compared against their corresponding thresholds, denoted respectively by IDD-threshold POR-threshold. Rather than using constant thresholds to determine the amount of network activity, we dynamically regulate the IDD-threshold and POR-threshold with respect to the varying conditions of the network.

Figure 7 shows the state determination function of the proposed algorithm. In this figure, the IDD and POR are

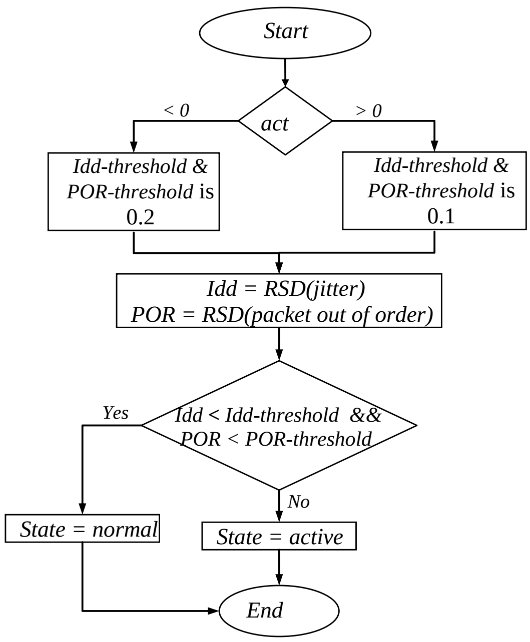

Fig. 7. State determination function.

between 0 and 1, which corroborates with the result of RSD(jitter) and RSD(packet out-of-order). As can be seen, in the critical situation in which the nodes have high mobility and in histories during which the network is in active state for a long time, the thresholds are decreased and set to 0.1. Conversely, when the network is in normal mode and the nodes have low mobility for a long time, thresholds are increased and set to 0.2. The history of the network's state is maintained by using a global variable denoted by act.

It is worth mentioning that during mobility-induced path switching, the receiver will experience a relatively high network delay variation together with a large ratio of out-of-order packets. In contrast to path switching, a congestion state will experience a high network delay variation but a low ratio of out-of-order packets.

Given that in ad-hoc networks, the main reason for decreasing voice quality is node mobility but not the congestion in intermediate nodes, the proposed algorithm primarily tackles with the issue of node mobility. We have accordingly applied the *AND* operator between *Idd* < *Idd-threshold* and *Poo* < *Poo-threshold*.

As for the calculation of the mean network jitter, it is common in the playout adjustment algorithms to estimate the mean jitter by using (3) with α considered as being a constant. However, a static α is not a reasonable choice for the calculation of the mean jitter in ad-hoc networks characterized by mobility of nodes and variable network conditions.

In this paper, we also use the equation described by (3) in order to calculate the mean jitter, but the alpha parameter will be regulated dynamically based on the current condition of the network, as there is no optimal fixed value of α when network condition varies in time.

As shown in Figure 8, the studies conducted in [5] suggests that the voice packet playout time can be adjusted properly by selecting a suitable value for α in different conditions.

Indeed, when the jitter is small and the fluctuations in the end-to-end delays are large, the best results are achieved for the small values of α. On the other hand, when the jitter is large but the average network delay is constant, the best results are achieved when α = 0.998002. Therefore, the proposed algorithm adjusts the alpha parameter according to the average delay and jitter as calculated in the following:

```
If  (jitter is small) and
   (fluctuations in the end-to-end delays are large)
       α = 0.7;
Else
       α = 0.998002;
```

As for the calculation of the mean network delay, it is very common in the playout adjustment algorithms to estimate the mean delay based on the equation described by (2). However, given the relatively high variations in the delay and the inherent unpredictability of the ad-hoc networks, the use of history for the mean network delay calculation does not stand to reason. In the context of our proposed algorithm, we have come up with an optimistic approach with respect to the future state of the network. It uses the minimum network delay associated with the last talkspurt as the mean network delay in the next talkspurt. As shown in Figure 9, we select the minimum delay value of the packets received on time before their playout time.

Figure 9 summarizes how the proposed algorithm processes the received packet in order to compute their optimum playout time. The proposed algorithm is triggered by the reception of a new voice packet. The state variable corresponds to the system state in which the playout time of

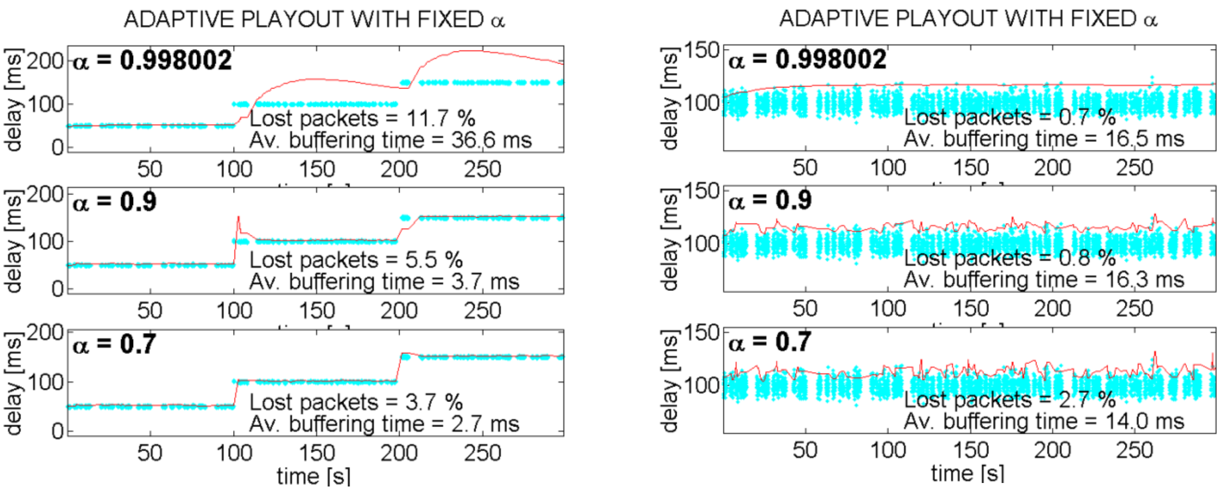

Fig. 8. The calculated playout times for various values of α [5].

the last packet has been adjusted.

Simulation results, discussed in the subsequent section, reveal that the proposed algorithm is capable of adapting itself with the network's dynamics and that it outperforms the existing schemes in terms of adjusting the playout delay of the voice packets.

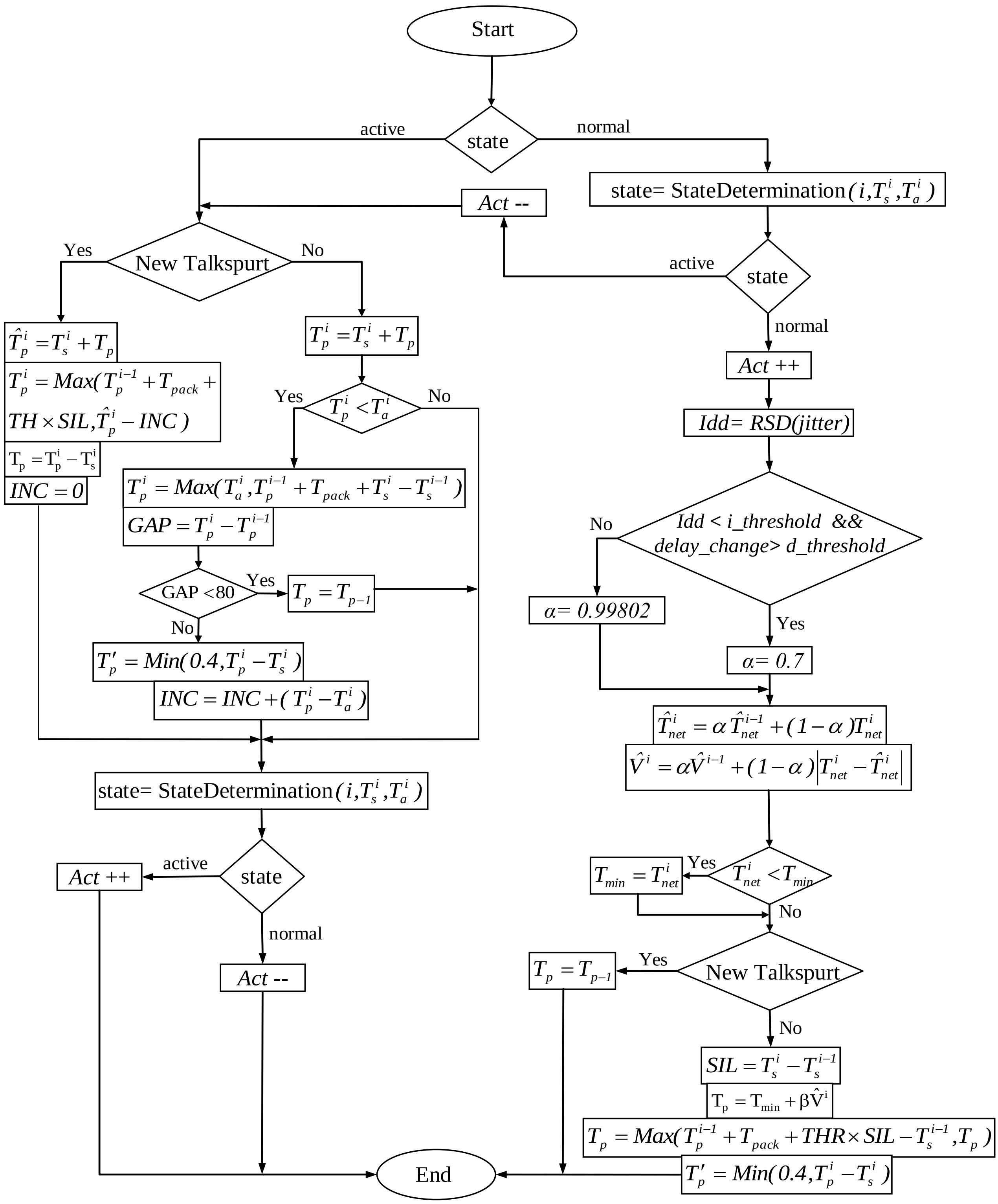

Fig. 9. The proposed algorithm.

4. Performance Evaluation

Usually there are two ways to evaluate the proposed algorithm: mathematical analysis and simulation. The first method is more reliable but is more complex and also in many cases it is not possible to model the given setting in terms of mathematical equations. Here, we have adopted the simulation approach to verify the presented algorithm.

4.1. Parameters Settings

We used NS2 running on Linux to simulate packet-based voice conversations over a MANET. In this work, we use the Gauss–Markov (GM) mobility model to generate the network topology. GM models the node movement according to the speed and direction values which are updated at discrete time intervals. This model avoids the unnatural abrupt movement of nodes and assures a uniform distribution of nodes over the simulated area. Each simulated scenario involves 25 mobile nodes roaming freely, i.e., without obstacles, over a rectangular area of 900 m × 300 m. The transmission range of each node is set to 250 m. Node velocities are randomly selected from the range 1–5 m/s to mimic node mobility in an urban environment.

As for the network traffic pattern, five bidirectional voice conversations have been established during the simulation run. Each voice conversation lasted for a randomly selected duration according to an exponential distribution with a mean value of 3 minutes. The start time of the sessions has been chosen uniformly. The generated voice packet stream of each session has mean active and silence periods of 1 s and 1.5 s respectively. In the active period, one packet is transmitted every 20 ms, each of length 160 bytes. We only consider delay traces having a packet loss ratio below 10% in each direction of a voice session. The other simulation parameters are summarized in Table 3.

Table 3

Simulation parameters

Parameter	Value
MAC protocol	IEEE 802.11b
Routing protocol	DSR
Transport protocol	UDP
Simulation duration	600 s
Wireless link bandwidth	2 Mbps
Codec	G.711

ITU-T E-Model is an applied method to validate the proposed algorithm. Figure 10 shows different schemes for the verification of the quality of voice conversation.

Generally, the voice quality verification methods are categorized into two groups: subjective and objective

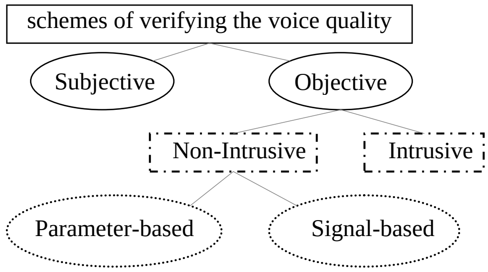

Fig. 10. Classification of the methods for the verification of the quality of voice conversation.

methods. The subjective methods are based on the listener's actual perception of the received voice; however, given that these methods are time consuming, expensive and unrepeatable, their use is prohibited for the verification of the voice quality. The objective methods, on the other hand, try instead to simulate the listener's perception. E-Model is one of the well known objective methods. The objective methods are classified into two groups: intrusive and non-Intrusive methods. The former yields a high precision but is in need of reference data and proves overwhelming in terms of the network bandwidth. On the contrary, the non-Intrusive methods do not need reference data and are suitable for real time network monitoring. Depending on the type of the input parameter, the non-Intrusive methods can be further sub-classified into two groups: signal-based and parameter-based methods.

The ITU-T E-Model is an objective, non-intrusive and a parameter-based method that is used to evaluate the voice quality. More specifically, the E-Model is a parametric computational algorithm which enables the objective derivation of the rating factor using a set of gathered measures throughout the mouth-to-ear path.

In this paper, the conversational quality of voice services is estimated through a rating factor denoted by R. The rating factor is based on the ITU-T E-Model which is a scalar ranging from 0 to 100, corresponding respectively to the worst and best transmission quality. Table 4 shows the relationship between the range of R and the voice quality. As can be seen, a rating factor value smaller than 60 corresponds to an unsatisfactory transmission quality.

Table 4

The relationship between the range of the rating factor and the voice quality

The range of R	Quality
50-60	Week
60-70	Bad
70-80	Not Bad
80-90	Good
90-100	Excellent

The reduced formula to derive the adequate rating factor R for assessing a VoIP conversation is given by [10]:

$$R = 93.2 - I_d(T_a) - I_e(CODEC, plr) + A, \qquad (6)$$

where I_d models the impairments affecting the interactivity such as the absolute propagation delay and echoes, I_e models the impairments affecting the intelligibility of voice conversations such as low bit-rate CODEC and packet losses, and A represents an advantage factor that accounts for a user's willingness to put up with some quality degradation in return for ease of access (e.g., cell phone). T_a and plr correspond respectively to the mean absolute propagation delay and packet loss rate. The distortion effects of I_d can be given by the following equation:

$$I_d(T_a) = 0.024T_a + 0.11(T_a - 177.3)H(T_a - 177.3), \qquad (7)$$

$$where \begin{cases} H(x) = 1 & if\ x < 0 \\ H(x) = 0 & if\ x \geq 0 \end{cases}$$

where T_a represents the mean absolute propagation delay including: framing, buffering, and network delays. In contrast to I_d, models of I_e should be developed and calibrated specifically for each CODEC. Typically, the proposed models in the literature have the following form:

$$I_e(CODEC, plr) = a + b \times ln(1 + c \times plr), \qquad (8)$$

where plr corresponds to the end-to-end packet loss rate, and the constant coefficients a, b, and c are selected according to the behavior of each CODEC [11,12]. For instance, the adequate coefficients of the G.711 CODEC with packet loss concealment (PLC) capability are $a = 0$, $b = 30$, and $c = 15$.

Given the dynamic and variable conditions of the MANETs during the conversation and in order to calculate the rating factor more accurately, it is recommended in [8] that the conversation duration be divided into fixed intervals (e.g., 10 s) to be assessed independently. The resultant rating factor is termed as the "instantaneous rating factor". It is also desirable to calculate the "overall rating factor" which is associated with the entire voice conversation.

4.2. Results

We have conducted a series of simulation experiments to evaluate the performance of the proposed algorithm and monitor its behavior in different conditions over MANET. We have also contrasted our results against those derived from two important adjustment algorithms discussed in [1] and [8] proposed for wired IP-based and wireless ad-hoc networks, respectively.

The behaviors of the proposed algorithm and the other algorithms in response to varying delay values over MANET are shown in Figure 11. It is revealed that the proposed algorithm tracks the variations in network delay more closely during the occurrence of a path switching and comes up with a faster adaptation of the playout delay compared to the other algorithms.

Figure 12 demonstrates the experienced one-way network delays over the simulated MANET during the 7th voice conversation. Figure 13 shows the instantaneous rating factor of the cited algorithms associated to this conversation. As mentioned before, the instantaneous rating factor is calculated every 10 seconds during the conversation in order to monitor the behavior of the algorithms more accurately.

Figure 14 shows the experienced one-way network delays

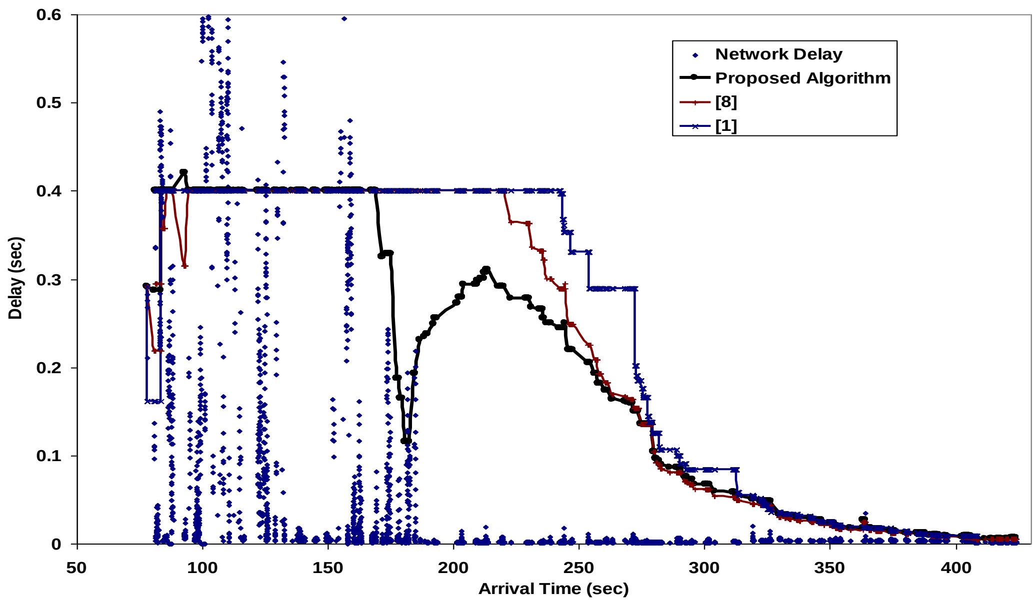

Fig. 11. The algorithms' behavior in face of delay variations over MANET.

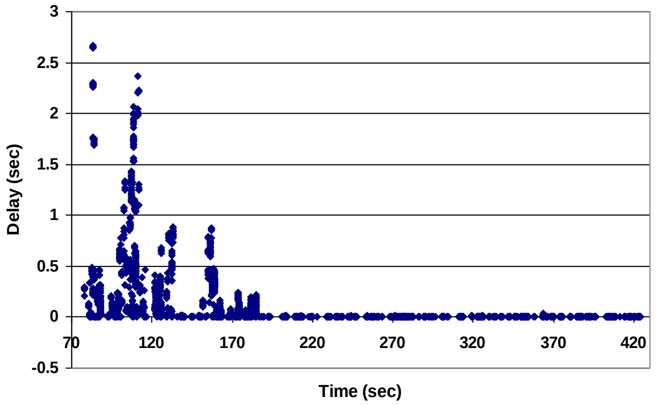

Fig.12. The 7th voice conversation.

over the simulated MANET during the 9th voice conversation. Figure 15 plots the instantaneous rating factor of the cited algorithms with reference to this conversation.

As shown in Figure 13 and Figure 15, the presented algorithm outperforms all its counterparts with respect to the instantaneous rating factor. Since the rating factor is derived from delay and packet loss (according to Equation (6)), our algorithm also improves the trade-off between the total end-to-end delay and the total packet loss rate.

Finally, at the end of each conversation, the "overall rating factor" is computed and used for calculating the "end-of-call rating factors".

In order to evaluate the algorithm's performance in the most general sense, the "end-of-call rating factors" that correspond to all the ten simulated voice conversations have been calculated and listed in Table 5. It is of note that the "end-of-call rating factor" for each algorithm is obtained by averaging over all "overall rating factors".

Table 5

The overall rating factors

Algorithm's Name	$R_{\text{end-of-call}}$
The [1]'s 2nd algorithm	59.3610316965707
The [1]'s 4th algorithm	64.9858081856080
The [1]'s 1st algorithm	67.5491730222304
The [1]'s 3rd algorithm	67.9647500651344
The[8]'s algorithm	71.0410115409502
The Proposed Algorithm	73.6013571098519

As can be seen in Table 5, the proposed algorithm achieves the highest estimated quality at the end of the processed call and can improve the performance by 10% to 14%. Indeed, the calculated rating factor indicates that the presented algorithm can minimize both the end-to-end

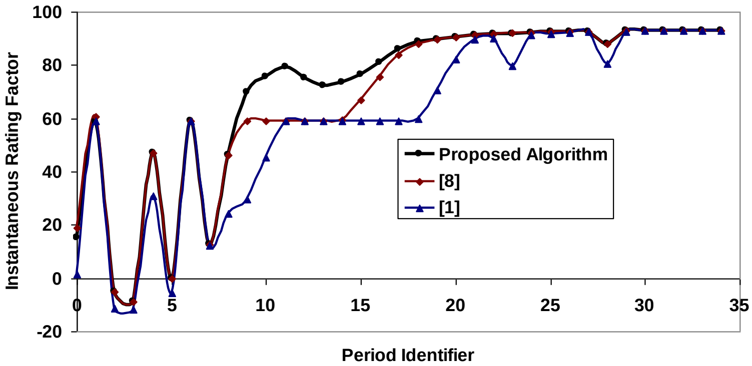

Fig. 13. The instantaneous rating factor of the cited algorithms w.r.t. the 7th conversation.

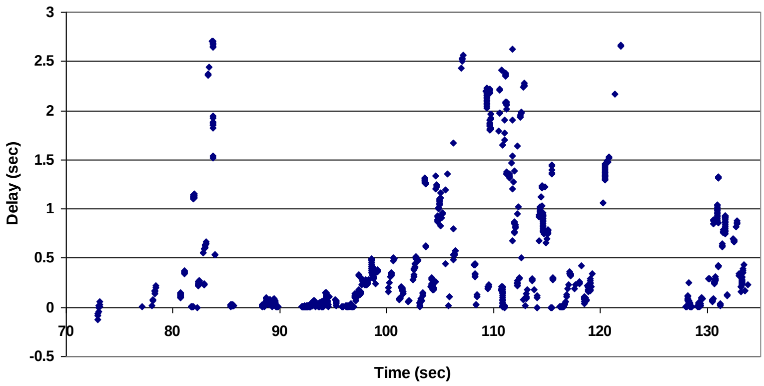

Fig. 14. The 9th voice conversation.

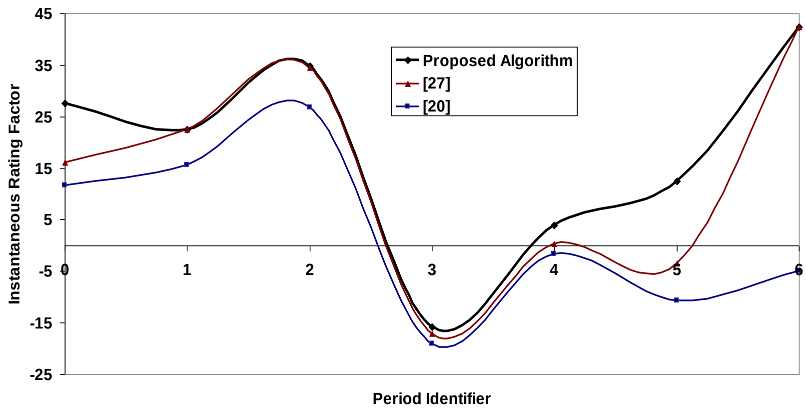

Fig. 15. The instantaneous rating factor of the cited algorithms w.r.t. the 9th conversation.

delay and packet losses due to late arrivals.

The key parameters of the voice quality can be calculated quantitatively to show the performance of the proposed algorithm more explicitly. To that end, the mean end-to-end delay and also the mean packet loss rate are computed. Figure 16 shows the cumulative distribution function associated with the end-to-end delays monitored during all the ten real-time simulated conversations. In this figure, the voice packet playout times are adjusted by the proposed algorithm.

As can be seen in the figure, most of the packets have admissible delay. Also, the mean end-to-end delay is 183 milliseconds. Furthermore, the packet loss rate is calculated during each conversation with the average rate of 18.2 percent.

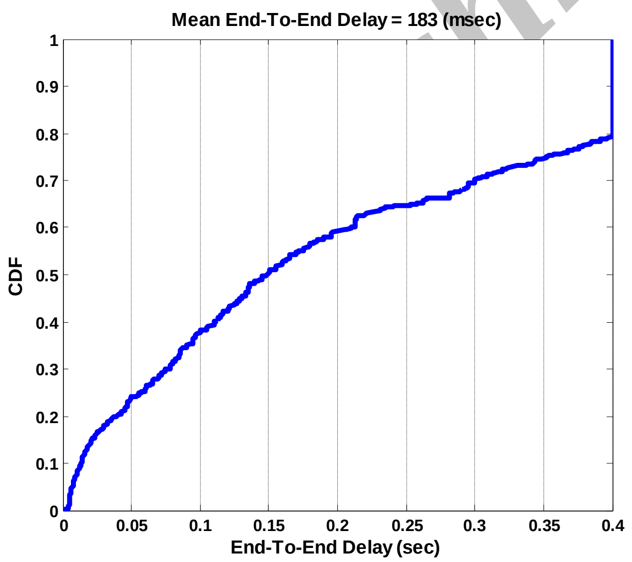

Fig. 16. The cumulative distribution function associated with the end-to-end delays monitored during all the ten voice conversations.

5. Conclusion

In this article, we have designed and developed a new adaptive playout adjustment algorithm for streaming the voice conversations over wireless ad-hoc networks. The designed playout algorithm adjusts the playout delay for each talk-spurt in response to node mobility. The major contribution of this paper is a novel playout buffer adjustment algorithm for adjusting the threshold adaptively with respect to network dynamics. Our proposed solution calculates the playout delay of a recently received packet by the dynamic estimation of the mean network jitter and especially the alpha parameter based on the current conditions of the network. We have favored an optimistic design with respect to the future state of network and have not based our calculation of the mean network delay on the delay history. Consequently, the proposed algorithm provides a minimum playout buffer delay, while also guaranteeing that the packet-loss rate remains below than a certain level. Playout buffer adjustment is handled within the talkspurts so as to be more reactive to the changing network conditions, especially when bursts or spikes occur. The results of our simulation experiments show that the proposed adaptive playout algorithm adapts itself with the network's dynamic conditions and adjusts the playout delay of voice packets more efficiently compared to the other algorithms; in particular, we have shown results suggesting that our design outperforms two representative adaptive playout buffering schemes (discussed in [1] and [8]) for all of the tested traces. In terms of the E-model R-factor, the algorithm has achieved a performance improvement of 14% and 10% compared to the works discussed in [1] and [8], respectively. More explicit calculations on the loss-delay trade-off have shown that the "Mean end-to-end Delay" with reference to ten real-time simulated voice conversations is 183 milliseconds, with the associated "Mean packet loss rate" of 18.2 percent. As part of our plans for future work, it is intended to improve the performance of the state determiner function.

References

[1] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, Adaptive playout mechanisms for packetized audio applications in wide area network, in: Proc. of IEEE INFOCOM, Toronto, Canada, vol. 2, pp. 680–688, 1994.

[2] M. Liu, QOS Improvement for real-time wireless VoIP, PhD. Thesis, Arizona University, 2006.

[3] V. Jacobson , M. J. Karels, Congestion avoidance and control, in: Proc. of ACM SIGCOMM Conf., Stanford, 1988.

[4] M. Salehi, QOS improvement for voice transmission over wireless LAN networks, MSc. Thesis, Amirkabir University of Technology, 2009.

[5] M. Narbutt, L. Murphy, VOIP playout buffer adjustment using adaptive estimation of network delays, in: Proc. of 18th Int. Teletraffic Congress (ITC-18), Berlin, Germany, pp. 1171–1180, September 2003.

[6] S. Moon, J. Kurose, D. Towsley, Packet audio playout delay adjustment: performances bounds and algorithms, ACM/Springer Multimedia Systems, vol. 6, pp. 17–28, 1998.

[7] Y. Liang, N. Farber, B. Girod, Adaptive playout scheduling and loss concealment for voice communication over IP networks, IEEE Trans. on Multimedia, vol. 5, no. 4, pp. 532-543, 2003.

[8] S. Jelassi, H. Youssef, Adaptive playout scheduling algorithm tailored for real-time packet-based voice conversations over wireless ad-hoc networks, in: Proc. of Elsevier Computer Networks, vol. 52, pp. 2473–2488, 2008.

[9] Z. Fu, X. Meng, S. Lu, TCP friendly rate adaptation for multimedia streaming in mobile ad hoc networks, in: Proc. of IEEE Int. Conf. on Management of Multimedia Networks and Services (MMNS'03), Belfast, Northern Ireland, UK, pp. 115–127, September 2003.

[10] M. Narbutt, M. Davis, Predicting the impact of application-layer adaptation mechanisms and MAC-layer parameter tuning on speech transmission quality in VoIP systems, Available on-line at http:// www.cnri.dit.ie/publications/, visited on January 2008.

[11] R.G. Cole, J.H. Rosenbluth, Voice over IP performance monitoring, computer communication review, ACM SIGCOMM, vol. 31, No. 2, 2001.

[12] L. Sun, E. Ifeachor, New models for perceived voice quality prediction and their applications in playout buffer optimization for VOIP networks, in: Proc. of IEEE Int. Conf. on Communication (ICC'04), vol. 27, No. 1, June 2004.

[13] N. Shacham, P. McKenney, Packet recovery in high-speed networks using coding and buffer management, In Proc. of the IEEE 1990 INFOCOM, vol.1, pp.124-131, June 1990.

[14] R.A Valenzuela, C.N. Animalu: a new voice-packet reconstruction technique, In Proc. of the IEEE 1989 Int. Conf. on Acoustic, Speech, and Signal Processing, vol.2, pp. 1334- 1336, 1989.

[15] A.K. Anandakumar, A.V. McCree, V.Viswanathan, Efficient CELP-based diversity schemes for VoIP, In Proc. of IEEE 2000 Int. Conf. on Acoustic, Speech, and Signal Processing, vol. 6, pp. 3682-3685, 2000.

[16] R.C.F. Tucker, J.E. Flood, Optimizing the performance of packet-switch speech, In Proc. of IEEE 1985 Conf. on Digital Processing of Signals in Communications, pp. 227-234, Apr. 1985.

[17] D.L. Stone, K. Jeffay, An empirical study of delay jitter management policies, Multimedia Systems 2 (1) (1995) 3–27.

[18] H. Schulzrinne, Voice communication across the Internet: a network voice terminal, Technical Report TR 92-50, Department of Computer Science, University of Massachusetts, Amherst, Massachusetts, July 1992.