A NEW SMOOTHING JITTER ALGORITHM FOR VOICE OVER AD HOC NETWORKS

Mouna Benaissa and Vincent Lecuire
CRAN, CNRS UMR 7039, Henri Poincaré-Nancy I University
Campus Scientifique, BP 239, 54506 Vandoeuvre-les-Nancy Cedex, France
{mouna.benaissa,vincent.lecuire}@cran.uhp-nancy.fr

Abstract Voice over IP applications require playout buffer at the receiver side to smooth network delay variations. Existing algorithms for dynamic playout adjustment used in Internet do not operate correctly in wireless ad hoc networks because they estimate end-to-end delay based on set of previous received audio packets. Mobility in ad hoc networks leads to topology changing and estimate based on past history is not appropriate. In this paper, we propose a new algorithm for playout delay adjustment based on *Route Request* AODV control messages to provide more accurate delay estimation. The performance evaluation shows that this algorithm outperforms existing playout delay adjustment algorithms. Performance criteria are the loss late percentage (reliability criterion), averaged playout delay (interactivity criterion) and playout delay variation (stability criterion).

Keywords: packet audio, playout delay, delay estimation, ad hoc network, AODV.

1. Introduction

One of the challenges of transmitting real-time voice on packet networks is how to overcome the variable inter-packet delay -the jitter- encountered as packets move on the transmission path through the network. In order to compensate these variable delays, packets are buffered at the receiver side and their *playout* is delayed for a period of time. Thus, most of the packets will be received before their scheduled *playout* times [Moon et al., 1998][Clark et al., 1992]. The playout delay must take into account three constraints. The first one is the interactivity constraint which requires playout delay below a certain value considered to be quite acceptable in human conversation (less than 300ms but 100ms is recommended to obtain excellent interactivity [ITU-T, 2001]). Second, the reliability constraint which requires little packet loss rate (generally less than 5%). Third, the stability constraint requires no large playout delay variation (this constraint is effective when playout delay is adjusted dynamically).

Extensive research work has been done to adjust dynamically the playout delay according to delay variation through the duration of an audio session. Existing algorithms estimate the end-to-end delay using collected delays measures of the more recent received audio packets. These measures can involve one packet (autoregressive algorithms)[Ramjee et al., 1994][Kansal and Karandikar, 2001], L packets [Moon et al., 1998][Leon and Sreenan, 1999][Agrawal et al., 1998][Liang et al., 2001], M talkspurts [Pinto and Christensen, 1999][Ramos et al., 2003] or all received packets [Fujimoto et al., 2002]. Generally, the playout delay is computed per-packet but adjusted per-talkspurt. However, these algorithms do not operate correctly in wireless ad hoc networks: when the topology changes, end-to-end delay estimation based on past history (delays of audio packets which arrived by obsolete route) becomes inappropriate.

This paper highlights effect of network reconfigurations on audio traffic transfer in ad hoc networks and then presents a new algorithm for playout delay adjustment appropriate to such environnement. End-to-end delay estimation is based on an ad hoc routing event: The *Route request* control message (RREQ) of AODV routing protocol. Of course, we suppose that AODV is adopted for the deployment of voice applications over ad hoc networks. However, it is not a drawback since we show in our prior work given in [Benaissa et al., 2003] that it is more appropriate for such applications.

The paper is organized as follows: In the section II, we present ad hoc network characteristics which can have a particular effect on audio traffic transfer, compared to wireline networks. In section III, we show that RREQ-AODV message provides an accurate estimation for end-to-end delay. Then, we describe our new playout delay algorithm based on RREQ-AODV delay messages. Section IV provides performance evaluation and comparison results obtained by simulation using ns-2. The performance criteria are loss late percentage (reliability criterion), averaged playout delay (interactivity criterion) and playout delay variation (stability criterion). Section V concludes the paper.

2. Ad hoc reconfiguration phases: A typical disturbing event for VoIP

In ad hoc network, mobile nodes communicate with others using multi-hop wireless links. There is no stationary infrastructure such as base stations. Each node in the network also acts as a router to forward data packet to other nodes [Perkins, 2001]. There is two approaches for existing routing protocols in mobile ad hoc networks: the proactive approach such as OLSR [Jaquet et al., 2002] and the reactive approach such as AODV[Perkins, 2001]. The proactive approach consists in every node emitting hellos messages periodically in order to learn the network topology. Reactive protocols invoke a route determination procedure on demand only.

Packets audio streaming over mobile ad hoc networks distinguishes clearly communication phases and network reconfiguration phases in an audio session. During a reconfiguration phase, the audio stream is interrupted because of the delay caused by a routing protocol to establish a new route towards the destination. The receiver identifies this phase when a sudden interruption occurs on packets arriving, generally followed by series of packets arriving with high end-to-end delays. Indeed, packets waiting for the new route, arrive with long delays at the destination if they are not dropped in the network queues. These interruptions disturb the played out audio speech at the receiver and can be long, generally of some seconds [Benaissa et al., 2003]). After this event, the new route for the new communication phase can present different or similar network conditions (traffic load and number of hops) compared to the previous one. We say *strong*, a reconfiguration which leads to different network conditions. We say *light*, a reconfiguration which leads to similar network conditions. Prior work given in [Benaissa et al., 2003][Jaquet and Viennot, 2000] have shown that:

- OLSR causes more reconfiguration phases than AODV, thus, AODV is more adapted for packet audio applications.
- Delays can vary significantly after a reconfiguration phase: Playout delay adjustement is needed.
- Reconfiguration phase causes disruption on the audio speech: It is possible to benefit from this interruption to adjust playout delay whithout any additionnal disturb.

Based on these works, we propose a new playout delay adjustment algorithm for voice over ad hoc networks. This algorithm presents a new approach for playout delay estimation and considers new event leading to its adjustment, appropriate to ad hoc environment.

3. RREQ-AODV algorithm

Jitter control required by voice applications faces a typical problem of spontaneous changes which occur on ad hoc network topology. When such a change happens, the delay on the new topology must be correctly estimated to be able to adjust the playout delay in accordance with current network conditions. The existing strategies used to estimate future delay are not efficient in ad hoc networks because they are based on past delay measures. The proposed approach is different: it uses AODV routing information to predict network conditions. In the following section, we show that RREQ message generated by AODV during a route discovery process provides pertinent information about future audio packet delay.

3.1 Delay indication using RREQ-AODV messages

The AODV routing protocol generates control messages to establish a new route towards the destination. The source node initializes a route discovery process just before sending data. It is achieved using a RREQ message which is broadcasted across the network. When the destination node receives the first RREQ message, it carries back the route in a RREP message to the source and ignores the next received RREQ messages for this route. The route established is the network path built by the RREQ message. Thus, RREQ message and audio packets use the same path from the source to the destination. This path is known by the receiver because AODV source uses the ones built by the first RREQ message which reaches the destination. So, the end-to-end delay achieved by the RREQ message presents for the receiver a pertinent indicator about delay audio packets to be received through this route.

At the receiver side, this indicator is available and updated dynamically before receiving audio packets of each communication phase. A new RREQ message is received during a reconfiguration phase from a new route discovery cycle, since the source maintains at most one route per destination. This can happen in several cases:

- **The beginning of an audio session**: As any reactive routing protocol, AODV initiates a route discovery process to start an audio session. This involves the sending of the first RREQ message which provides to the receiver a delay indication from the beginning of the session.

- **Mobility**: When a node moves from an active audio path, a new reconfiguration phase begins. The node which detects the link failure sends a route error (ERR) message to the source. If the audio source node still desires the route, it reinitiates route discovery process. So, the destination receives a new RREQ message which provides a new delay indication appropriated to the new topology.
- **Long silence period**: AODV maintains a timer-based states in each node, about the usage of individual routing table entries. A routing table entry is expired if not used recently. Thus, the audio route expires during a silence period which is longer than a route expiry time. Due to this long silence period, the source needs to initiate route discovery process at the next talkspurt. This provides a new RREQ message to the audio destination and updates the delay indicator.
- **High traffic load**: AODV maintains topology information via *HELLO* messages. If a node does not receive any *HELLO* message from its known neighbor, the link is considered broken. This can occur when network traffic load is high, even if there is no mobility. So, a new route discovery process starts and presents a new RREQ message to the receiver. This message provides a new delay indication appropriated to the network load conditions.

In our work, the receiver requires always to be notified about route changing by RREQ messages. It is obtained in the following way: Firstly, the procedure of RREQ message broadcasting is modified. In the initial procedure, the intermediate node receiving the RREQ message may send a RREP message if it has a route to the destination and stop broadcasting. Thus, the audio destination does not receive any *RREQ* message and can not detect this reconfiguration. To notify the receiver about this reconfiguration, this procedure is modified. The intermediate node having a route to the destination unicasts the received RREQ message to the final destination which reply a RREP message to the source. Secondly, a local repair procedure is not used. Using this procedure, AODV attempts to repair localy a failed link instead of informing the source and initiates a new route discovery process. To notify the destination about this reconfiguration, this procedure is not used in our work. Note that in 50 nodes networks, this approach does not have any significant performance advantage while it is recommended in larger networks to increase scalability [Lee et al., 2003].

3.2 RREQ-AODV algorithm description

In this section, we describe a new approach to adjust the playout delay in voice over ad hoc networks. This approach is based on a typical event of AODV routing protocol to estimate correctly the end-to-end delay even if network topology was changed: the RREQ control message arrival on the receiver side.

3.2.1 Playout delay estimation. The playout delay estimation is based on delay indication provided by RREQ messages. Let $Drreq_n$ be the end-to-end delay achieved by the RREQ message received during the n^{th} reconfiguration phase. The playout delay estimation $\hat{d}_{n,k}$ for packets of the k^{th} talkspurt belonging to the n^{th} normal phase is computed as follows:

$$\hat{d}_{n,k} = Drreq_n + \beta_{n,k} \qquad (1)$$

where $\beta_{n,k}$ is a safety factor, added to ensure that the estimated end-to-end delay is greater than the actual network delay. To get more accurate playout delay estimation, this factor is adjusted dynamically. This is discussed in the next section.

3.2.2 Playout delay adjustment. The delay indication $Drreq_n$ is updated at every RREQ message arrival and used at the beginning of a new normal phase (adjusted during reconfiguration phase) or a new talkspurt (adjusted during silence period): If arriving packet i is the first packet of talkspurt k or a normal phase n, the playout delay is computed as given in equation 1. This delay is preserved for each subsequent packet j. The playout times p_i and p_j are computed as below:

$$p_i = t_i + \hat{d}_{n,k} \qquad (2)$$

$$p_j = p_i + (t_j - t_i) \qquad (3)$$

where t_i and t_j are (respectively) times at which packets i and j are generated at the sender.

3.3 Safety factor adjustment

We distinguishes two events for adaptation of the safety factor $\beta_{n,k}$: the beginning of a new normal communication phase (a RREQ message is received) and the beginning of a new talkspurt (a new talkspurt begins in the same normal phase while no new RREQ message is received). The adaptation of $\beta_{n,k}$ is performed in the following way for each case:

- ***RREQ* message is received:** The reception of a new RREQ message indicates that a new network topology is established. Then, the new delay indication $Drreq_n$ is updated in equation (1). To identify the type of the occurred reconfiguration (light or strong), the algorithm computes the difference δ between the current delay indication $Drreq_n$ and the previous one $Drreq_{n-1}$ as follows:

$$\delta = |Drreq_n - Drreq_{n-1}| \qquad (4)$$

The algorithm compares this difference to a certain threshold $threshold\_req$ and adjust $\beta_{n,k}$ accordingly:

1 **δ is large enough ($\delta > threshold\_req$):** The result of this comparison indicates that the network conditions on the new topology have changed significantly; it was a strong reconfiguration. For the new normal phase, the delay estimation is based on the new delay indication $Drreq_n$ and $\beta_{n,k}$ is set to its primary value β_{min}:

$$if\ |Drreq_n - Drreq_{n-1}| > threshold\_req$$

$$then\ \beta_{n,k} \leftarrow \beta_{min}; \tag{5}$$

2 **δ is small ($\delta <= threshold\_req$):** In this case, the algorithm considers that the network conditions are similar on the new topology; it was a light reconfiguration. The safety factor $\beta_{n,k}$ preserves its previous value $\beta_{n-1,k}$:

$$if\ |Drreq_n - Drreq_{n-1}| <= threshold\_req$$
$$then\ \beta_{n,k} \leftarrow \beta_{n-1,k}; \tag{6}$$

- **No *RREQ* message is received while a new talkspurt begins**: The absence of RREQ message indicates that no changes happen on the ad hoc network topology. In this case, the delay estimation for this talkspurt can be based on the recent delay past history. We propose to adapt $\beta_{n,k}$ as a function of loss percentage q_{k-1} achieved on the more recent talkpsurt $(k-1)$:

$$\beta_{n,k} = f(\beta_{n,k-1}, q_{k-1}) \tag{7}$$

To keep a certain stability of the estimated playout delay, the adjustment of $\beta_{n,k}$ in equation (7) is performed in a gradual way as follows:

1 **No loss observed on the previous talkpsurt** ($q_{k-1} = 0$): This means that the safety factor is large and it can be decreased to improve interactivity without degrading reliability. Then, $\beta_{n,k}$ is decreased by a factor r:

$$if\ q_{k-1} = 0\ then\ \beta_{n,k} = (1-r)\beta_{n,k-1} \tag{8}$$

2 **Loss percentage is less or equal than the user tolerable limit q_{ref}** ($q_{k-1} \leq q_{ref}\%$): This means that the algorithm gets a good reliability but there is no margin on the safety factor to improve interactivity. Then, $\beta_{n,k}$ preserves its previous value for the next talkspurt in order to maintain the same level of reliability and interactivity:

$$if\ q_{k-1} \leq q_{ref}\ then\ \beta_{n,k} = \beta_{n,k-1} \tag{9}$$

Note that in our work, q_{ref} is set to 3%.

3 **Loss percentage exceeds the user tolerable limit** ($q_{k-1} > q_{ref}\%$): This means that the safety factor is small. It must be enlarged at the next talkspurt to increase reliability; $\beta_{n,k}$ is increased by a multiple of factor r as function of observed loss percentage q_{k-1}, as follows:

$$\begin{array}{lll} if\ q_{ref}. < q_{k-1} < 10\% & then & \beta_{n,k} = (1+2r)\beta_{n,k-1} \\ if\ 10 < q_{k-1} < 20\% & then & \beta_{n,k} = (1+4r)\beta_{n,k-1} \\ if\ 20 < q_{k-1} < 30\% & then & \beta_{n,k} = (1+6r)\beta_{n,k-1} \\ if\ q_{k-1} > 30\% & then & \beta_{n,k} = 2\beta_{n,k-1} \end{array} \tag{10}$$

> The equation 10 is defined as to avoid adjusting $\beta_{h,k}$ with strong value from talkspurt to another. Indeed, we aim to keep a certain stability of the playout delay. For larger value of r, the playout delay adjustment is larger.

To keep acceptable interactivity (end-to-end delay less than $300ms$ including delay required to collect audio samples), the adjustment of $\beta_{h,k}$ is bounded between $\beta_{min} = 40$ and $\beta_{max} = 200$. Parameters $threshold\_req$ and r are chosen in way they give the better tradeoff between interactivity, reliability and stability. $threshold\_req$ is set to 80 an r is set to 0.05. Figure 1 shows that the playout delay follows suitably end-to-end delays and presents a good stability.

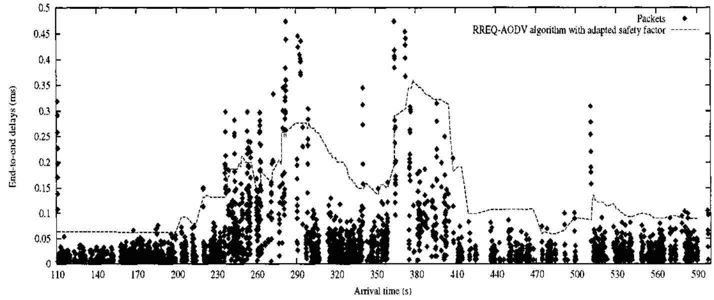

Figure 1. Playout delay using algorithm RREQ-AODV with adapted safety factor β.

4. Performance evaluation

In this section, we evaluate and compare the RREQ-AODV algorithm performances to autoregressive based algorithms 1 and 4 reported in [Ramjee et al., 1994] (refered as mean delay algorithm and spike algorithme in this paper) and L packets statistics based algorithm reported in [Moon et al., 1998] (refered as *Moon* algorithm in this paper). The results shown in this section are evaluated on six audio traces obtained by simulation.

4.1 Performance metrics

To measure the obtained audio quality Q at the receiver when applying a playout delay algorithm, we take into account three criteria: Interactivity (averaged playout delay I), reliability (percentage of loss due to late arrivals F) and stability (averaged playout delay jitter S). The E-model predicts the subjective quality Q of a telephone call based on its characterizing transmission parameters. It combines impairment caused by these parameters into a single rating Q. According to the ITU-T recommendations, the rating value range of Q corresponds to a speech transmission category, as follows: Best for range of $[90, 100]$, High for range of

$[80, 90]$, Medium for range of $[70, 80]$, Low for range of $[60, 70]$ and Poor for range of $[0, 60]$. The rating Q is given by:

$$Q = Q_0 - E \tag{11}$$

where Q_0 takes into account the effects of noise. The default value of Q_0 is 94.2. E combines impairment of different transmission parameters. In our work, E groups the impairment relative to interactivity $E(I)$, the impairment relative to reliability $E(F)$ and the impairment relative to stability $E(S)$. $Q(I, F, S)$: $\mathcal{R}^+ \times \mathcal{R}^+ \times \mathcal{R}^+ \longrightarrow [0, 100]$ is given by:

$$Q(I, F, S) = 94{,}2 - E(I) - E(F) - E(S) \tag{12}$$

Let p_i be the playout delay of packet i, N be the total sent audio packets, A be the total received audio packets, L be the total played out audio packets during the audio session. A packet i, sent at time t_i and received at time a_i, is played out if it arrives before its playout time tp_i, ie: $a_i \leq tp_i$ (where $tp_i = t_i + p_i$).

4.1.1 Interactivity metric. The averaged playout delay I provides indication about the interactivity level. I is given by:

$$I = \frac{1}{L}\sum_{i=1}^{L}(p_i) \tag{13}$$

In human conversation, end-to-end delay must not exceeds $110ms$ for a good interactivity but tolerates degraded audio quality for end-to-end delay between $110ms$ and $260ms$ (In our work 40ms are required to collect samples of one audio packets). When end-to-end delay exceeds $260ms$, audio quality is poor. Considering these bounds, $E(I)$ is given by [Boutremans and Le Boudec, 2003]:

$$E(I) = \begin{cases} 0.001\, I & for\, I \leq 110 \\ 18.89\, tanh(0.02\, (I - 185)) + 17.1 & for\ 110 < I \leq 260 \\ 0.01 I + 32 & for\, I > 260 \end{cases} \tag{14}$$

4.1.2 Reliability metric. The loss late percentage F indicates the reliability level. F is given by:

$$F = (\frac{A - L}{A})100 \tag{15}$$

Independently of the codec in use, $E(F)$ is given by [Boutremans and Le Boudec, 2003]:

$$E(F) = 34.3 ln(1 + 12.8F) \tag{16}$$

According to equation 16, when the percentage of loss is less than 3%, the audio quality is good but tolerates degraded audio quality for loss percentage between 3% and 15%. When loss percentage exceeds 15%, audio quality is poor.

4.1.3 Stability metric. The jitter S on the playout delay during the audio session provides indication about the stability of the playout delay. S is given by:

$$S = \frac{\sum_{i=2}^{L} |p_i - p_{i-1}|}{(L-1)} \qquad (17)$$

In our work, $E(S)$ is given as follows:

$$E(S) = 2 \times S \qquad (18)$$

As S increases, E(S) increases and playout delay is less stable.

4.2 Reference traces

We selected six reference traces which characterize different ad hoc network conditions (load traffic and mobility speed). These traces were obtained by simulation using ns-2. Our network model consisted of 50 nodes in a 1000 × 1000 meter flat, square area. The nodes moved according to the random way point mobility model. All nodes communicated with 802.11 based wavelan wireless radios, which have a bandwidth of 11Mbps and a propagation radius of 250m (See [Benaissa, 2004] for complete details on the simulation environment and methodology). We use AODV protocol for routing and PCM codec (Pulse Codec Modulation- See Recommendation G.711 in [ITU-T, 2001]) to generate audio traffic. For each trace, we get sending and receiving time of all audio packets and all RREQ messages transfered from the audio source to the audio destination. Principals characteristics of these traces are:

- Reference trace 1, 2 and 3: They present normal node mobility ($1m/s$ to $2m/s$) and normal load traffic conditions. We consider that network conditions on these three traces are favorable to VoIP applications.
- Reference trace 4: It presents high mobility ($8m/s$) and high load traffic. We consider that network conditions on this trace are difficult for VoIP applications. Such a trace is useful to study the behavior of playout delay algorithm in difficult situation.
- Reference trace 5: This trace presents light load traffic and high mobility (6m/s). End-to-end delays are very small during normal phases and very high after reconfiguration phases. When route reconfiguration occurs, packets waiting for new route arrive with very high delays because network queues are not loaded and thus these packets are not discarded. This trace is used to show particularly the effect of mobility on end-to-end delays packets.
- Reference trace 6: This trace presents normal load traffic conditions without any mobility. We consider that this trace is appropriate for mean delay algorithm, spike algorithm and *Moon* algorithm.

4.3 Performance comparison

In this section, we evaluate and compare RREQ-AODV algorithm to mean delay algorithm, spike algorithm and *Moon* algorithm on the six reference audio traces. We give a table 1, summarizing the obtained results: I, F, S, and $Q(I, F, S)$.

4.3.1 Reference trace 1, 2 and 3: Normal mobility with normal load conditions. Results given in table 1 show that RREQ-AODV algorithm outperforms other algorithms on the three traces. Mean delay algorithm obtains good stability and fiability at the cost of degraded interactivity. Impairment relative to stability is very important with spike algorithm. *Moon* algorithm gives degraded reliability. Algorithm RREQ-AODV obtains the better tradeoff between interactivity, reliability and stability. When observing results in details from the traces, we remark that RREQ-AODV algorithm follows more suitably the delays tendency than the others.

4.3.2 Reference trace 4: High mobility with high load traffic. The four algorithms lead to poor audio quality. Mean delay algorithm outperforms others when considering reliability ($F = 4.4\%$) but it obtains poor interactivity (I is 10 times higher than with RREQ-AODV algorithm). Indeed, RREQ-AODV algorithm provides good interactivity but at the cost of poor reliability ($F = 20.91\%$). Spike algorithm and *Moon* algorithm give degraded stability ($S > 10ms$) and reliability. When observing results in more details, we remark that most of lost packets are those arriving with great delays, which are not useful to the VoIP application. These packets are played out when using mean delay algorithm which leads to excessive playout delay. In difficult network conditions, playout delay adjustment algorithms cannot give a good tradeoff between interactivity and reliability. In this case, mechanisms for the quality of service must be deployed in the network.

Trace	**Algorithm**	**I**	**F**	**S**	**Q(I,F,S)**	**Quality**
1	Mean delay	241.46	6.01	0.32	41.57	poor
	Spike	194.29	6.72	3.63	45.08	poor
	Moon	161.46	8.75	1.10	58.52	poor
	RREQ-AODV	131.04	5.33	0.32	73.91	medium
2	Mean delay	143.27	4.40	0.21	74.23	medium
	Spike	72.66	6.74	1.45	69.89	low
	Moon	92.45	8.25	0.93	68.46	low
	RREQ-AODV	86.31	3.45	0.24	81.09	good
3	Mean delay	181.35	5.30	0.21	60.29	low
	Spike	117.08	8.73	1.88	64.16	low
	Moon	115.41	10.79	0.86	63.16	low
	RREQ-AODV	124.66	6.43	0.34	71.65	medium
4	Mean delay	1467.28	4.40	2.43	27.33	poor
	Spike	361.42	13.50	11.21	1.74	poor
	Moon	832.55	11.75	10.79	11.60	poor
	RREQ-AODV	148.80	20.91	1.83	40.48	poor
5	Mean delay	958.15	5.35	2.74	29.24	poor
	Spike	208.63	9.53	9.01	23.41	poor
	Moon	568.03	10.06	9.03	19.10	poor
	RREQ-AODV	84.91	6.74	0.28	72.22	medium
6	Mean delay	44.32	3.8	0.04	80.48	good
	Spike	37.55	6.70	0.66	71.60	medium
	Moon	36.17	6.17	0.18	73.99	medium
	RREQ-AODV	61.91	1.3	0.19	88.53	good

Table 1. Result obtained on the six audio traces

4.3.3 Reference trace 5: high mobility with light load traffic.

RREQ-AODV algorithm provides medium audio quality and then outperforms other algorithms which lead to poor audio quality. Mean delay algorithm, spike algorithm and *Moon* algorithm do not give good stability and interactivity. This is due to the fact that these algorithms consider long delays caused by reconfiguration phases in their estimation, while such delays do not give an appropriate indication about future delays. However, RREQ-AODV algorithm leads to more stable playout delay ($S = 0.28$) and excellent interactivity at the cost of degraded reliability. A more careful analysis of the results reveals that most part of the lost packets are those arriving with great delays. Such packets are considered lost by the audio application. Thus, the algorithm RREQ-AODV does not increase playout delay if there is no additional advantage. These results confirm that algorithms which are based on delay past history are not appropriate to adjust playout delay in the presence of mobility even with light load traffic. In these conditions, algorithm RREQ-AODV reacts correctly.

4.3.4 Reference trace 6: no mobility with normal load conditions.

Results show that all algorithms provide excellent interactivity and stability but lead to different reliability levels. When observing results in more details, we remark that mean delay algorithm and spike algorithm underestimate playout delay and thus lose packets arriving with acceptable delays. As algorithm RREQ-AODV considers a minimum bound for the safety factor ($\beta_{n,k} > 40$), it looses less packets than the others.

5. Conclusion

In this paper, we have proposed a new playout delay algorithm specially designed for voice over ad hoc networks. Its first strength is in the way that it estimates the end-to-end delay in the presence of mobility which leads to route reconfiguration. The algorithm uses RREQ-AODV message delay as a delay indicator. This is appropriate because the receiver is sure that the audio packet will go through the same path. Its second strength is the adaptation strategy which gives the same importance to interactivity, reliability and stability constraints. The performance evaluation results show that our algorithm outperforms existing algorithms in all cases when considering simultaneously: the interactivity, the reliability and the stability criteria, as well as when considering only interactivity and reliability criteria. A drawback of our solution is that the methodology is tied to AODV. However, a general methodology for reactive protocols can perhaps be derived from the proposal. In conclusion, our algorithm will contribute to improve the quality of voice application running on ad hoc networks. Of course, other mecanisms, such as FEC and network differenciated services, should be also used for supply enough QoS for user of voice applications.

6. References

[Agrawal et al., 1998] Agrawal, P., Chen, J., and Sreenan, C. (1998). "use of statistical methods to reduce delays for media playback buffering". In *International Conference on Multimedia Computing and Systems*, pages 259–263.

[Benaissa, 2004] Benaissa, M. (2004). *Dynamic playout delay adjustment for voice over ad hoc network.* PhD thesis, Henri Poincaré - Nancy I University.

[Benaissa et al., 2003] Benaissa, M., Lecuire, V., Lepage, F., and Schaff, A. (2003). "analysing end-to-end delay and loss in mobile ad hoc networks for interactive audio applications". *Workshop on Mobile Ad Hoc Networking and Computing MADNET'2003*, pages 27–33.

[Boutremans and Le Boudec, 2003] Boutremans, C. and Le Boudec, J. Y. (2003). "Adaptive joint playout buffer and fec adjustment for internet telephony". In *Proceedings of IEEE INFO-COM'2003*, San-Francisco, CA.

[Clark et al., 1992] Clark, D., Shenker, S., and Zhang, L. (1992). "Supporting real-time applications in an integrated services packet network architecture and mechanism". *SIGCOMM'92*, pages 14–26.

[Fujimoto et al., 2002] Fujimoto, K., Ata, S., and Murata, M. (2002). "Adaptive playout buffer algorithm for enhancing perceived quality of streaming applications". In *IEEE Globecom.*

[ITU-T, 2001] ITU-T (2001). "List of itu-t recommendations". http://www.itu.int/publications/itu-t/itutrec.htm.

[Jaquet et al., 2002] Jaquet, P., Muhletaler, P., Quayyum, A., Laouiti, A., Clausen, T., Viennot, L., and Minet, P. (2002). "Optimized link state routing protocol". *Internet Draft draft-ietf-manet-olsr-06.tx, Internet Engineering Task Force.*

[Jaquet and Viennot, 2000] Jaquet, P. and Viennot, L. (2000). "Overhead in mobile ad hoc network protocols". *Technical Report, INRIA.*

[Kansal and Karandikar, 2001] Kansal, A. and Karandikar, A. (2001). "Adaptive delay estimation for low jitter audio over internet". *in proccedings of IEEE Globcom*, pages 17–18. San Antonio, USA.

[Lee et al., 2003] Lee, S.-J., Royer, E. M., and Perkins, C. E. (2003). "Scalability study of the ad hoc on-demand distance vector routing protocol". *ACM/Wiley International Journal of Network Management*, 13(2):97–114.

[Leon and Sreenan, 1999] Leon, P. D. and Sreenan, C. (1999). "An adaptive predictor for media playout buffering". In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3097–3100.

[Liang et al., 2001] Liang, Y., Farber, N., and Girod, B. (2001). "Adaptive playout scheduling and loss concealment for voice communications over IP networks". *IEEE Transactions on Multimedia.*

[Moon et al., 1998] Moon, S., Kurose, J., and Towsley, D. (1998). "Packet audio playout delay adjustment: Performance bounds and algorithms". *ACM/Springer Multimedia Systems*, 6:17–28.

[Perkins, 2001] Perkins, C. (2001). "Ad hoc networking". Addison Wesley Longman.

[Pinto and Christensen, 1999] Pinto, J. and Christensen, K. (1999). "An algorithm for playout of packet voice based on adaptive adjustment of talkspurt silence periods". In *Proceedings of the IEEE 24th Conference on Local Computer Networks*, pages 224–231.

[Ramjee et al., 1994] Ramjee, R., Kurose, J., Towsley, D., and Schulzrinne, H. (1994). "Adaptive playout mechanisms for packetized audio applications in wide-area networks". In *Proceedings of the IEEE Infocom*, pages 680–688.

[Ramos et al., 2003] Ramos, V., Barakat, C., and Altman, E. (2003). "A moving average predictor for playout delay control in voip". Technical report, Nice-Sophia Antipolis University, Mistral and Planète research team, INRIA.