Voice over AODV

Durga Prasad Pandey
MIT Media Lab
dpsmiles@mit.edu

ABSTRACT
This paper studies the performance of voice transmission on a wireless ad-hoc network running the AODV(Ad-hoc On-demand Distance Vector) routing protocol. We use the MOS(mean opinion score) score to carry out Perceptual Evaluation of Voice Quality(PESQ). Repeated tests show that voice over wireless has a high MOS score over one hop and degrades as the number of hops is increased to two. However, the deterioration of voice quality is not as much due to increase in the number of hops, as due to the instability of the routing protocol at hop edges. AODV selects routes based on minimizing the number of hops, leading to oscillations at the boundary between one and two hops. We use our experimental results to point out inadequacies in AODV, and suggest measures for improvement.

Keywords
Ad-hoc, Wireless, VoIP, Multihop

1. INTRODUCTION

Given the huge growth of wireless networking in recent years, and the expectation that it'll continue growing at a very fast rate because of the obvious benefits it offers, the issue of optimizing the usage of wireless links is a critical one. With the clear success of Skype[1] as a VoIP application over the Internet, there is considerable interest and faith in the packet switched network to take over the world of telephony.

While most of the protocols on the Internet were designed for and work well with wired networks, they don't perform quite as well on wireless networks[2]. There is a simple reason behind this: radios are not wire-like links. Rather, radio communication involves using a shared medium with very little shielding from other radio transmissions. Ad-hoc wireless networking takes this contrast a step further, by extending the architecture based on an assumption of an access point connecting different nodes, all one hop away, to an architecture where there is no access point, nodes themselves forward packets like routers, and multiple hops are the norm.

Our work will directly address the routing subproblem, which critically affects the performance of multihop wireless networks. We draw inspiration from the FluidVoice[3] project, which argues that voice being a reasonably bandwidth demanding as well as latency and delay sensitive application, is best suited for stressing wireless networks and revealing their weaknesses so that they can be made more robust and reliable. Sensor networks, for example, do not stress the network for bandwidth as data from sensors is relatively low-bandwidth. Video, on the other hand, can be buffered, so it isn't as delay sensitive as voice.

We have chosen the Ad hoc On Demand Distance Vector (AODV)[4] routing algorithm for this study, which is a routing protocol designed for ad hoc mobile networks. AODV is one of the most prominent among a large number of competing ad-hoc routing protocols[5]. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

We start by asking ourselves: are current routing protocols, in this instance AODV, best suited for voice applications over ad-hoc wireless networks? If not, what changes do we need to incorporate to make these protocols better. For this study, we have deliberately chosen to use a real voice application setup, rather than resorting to the conventional method of using simulations. While simulation has its own benefits, we believe that testing a real system will reveal issues and offer insights that might not be as visible in a simulation study.

2. EXPERIMENTAL SETUP

While carrying out this experiment, we identified Linphone[6] as a better alternative to Kphone[7], which we had originally proposed to use. Kphone is often slow to start up, sometimes taking upto a minute, at times not popping up at all. Linphone and Kphone are both open source Voice over IP softphones for Linux. They are based on the Session Initiation Protocol(SIP)[8], an open standard for Internet telephony.

We carried out tests with a 3-node setup as shown below:

[A] <--------> [B] <---------> [C]

In our initial setup, A played an audio wav file(duration:27 seconds, 16-bit encoded at 16 KHz) repeatedly. A microphone was placed near the speakers, which captured the audio being played out. A simultaneously ran Linphone, so that the audio being captured by A's microphone was transmitted to C. All 3 laptops ran the AODV protocol. B would be expected to forward packets between A and C when AODV discovered that that was the best current route. C played out the received audio using Linphone. It also recorded the received audio using ReZound[9].

Wireless routing typically picks up the shortest path(lower number of hops) over one that minimizes loss rate. This has the advantage of voice packets avoiding the latency involved in going through the processing and forwarding by an intermediate node. However, a highly lossy link might lead to loss of a large number of packets leading to degradation of voice quality at the other end. Thus, there is clearly a tradeoff involved here. We intended to study their relative importance to perceived voice quality as part of this project.

We chose AODV-UU, which is a popular open source AODV implementation developed at the University of Uppsala for our experiments. AODV-UU optimizes paths on the basis of minimum hop count.

We started off by relying on our sense of hearing to judge voice quality. However, the human ear isn't really a consistent and reliable instrument of scientific measurement. The need was thus felt for an objective measure of the perceived sound quality. Based on a suggestion from Prof Katabi, we decided to use ITU-T recommendation P.862, also known as Perceptual Evaluation of Voice Quality, or PESQ[10] for our study. PESQ measures the effects of distortions such as noise, delay and front-end clipping to model and predict subjective quality. PESQ is considered to be the most accurate speech quality standard[11], and correlates highly with subjective test results. The measure used for this evaluation is a subjective scoring called MOS(Mean Opinion Score), originally designed by the Bell Companies to quantify the quality of a voice call; with 1 being unacceptable and 5 being superlative.[ban] "Toll quality" sound is associated with a MOS score of at least 4. G.711 starts with a MOS score of 4.4. G.729, which performs compression, has an average MOS score of 4.1. With VoIP, the MOS score will be further reduced when there is packet loss, excessive delays, etc.

Our aim was to generate the following four graphs:

1) Plot the MOS score as a function of time over a single hop wireless path.
2) Plot the Average MOS score over one-hop wireless path as a function of distance between sender and receiver. Each point on your graph would be the average MOS score taken over multiple samples that have the same distance between the sender and receiver.
3) Plot the MOS score as a function of time over a 2-hop wireless path.
4) Plot MOS as a function of time for the following experiment. Start by positioning C close to A so that there is a one-hop path. After 5 minutes, move the receiver to a new position so that the path become 2-hop length. The receiver is to be moved very slowly. For instance, each time, take two steps quickly and then stay 60 seconds in the same place, then take again two steps, and again wait for 60 seconds. The plot we get should reveal how the MOS score changes as a function of distance. At some threshold, we should see a major change that occurs because the routing is moving to two-hop paths from one-hop path.

3. INITIAL OBSERVATION

One of the first observations was the low MOS value we obtained even on single hop paths. Some thinking and experimentation helped us establish that the low MOS score was due to the loss of audio quality in the process of playing it out and picking it up from the microphone. We found that the MOS score was low even between the original file and a file recorded while playing it on the same computer. So what we were feeding into Linphone itself was low quality audio.

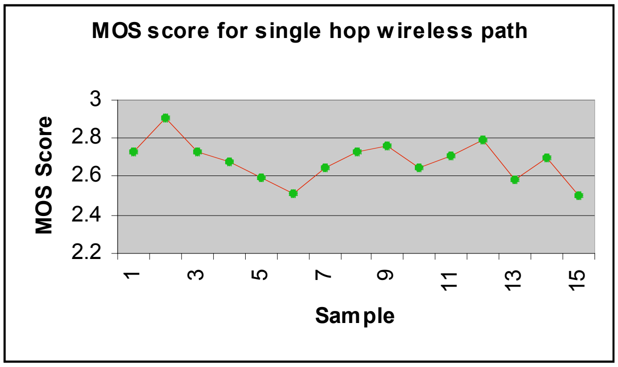
To fix this problem, we modified the experimental setup, after a former colleague suggested using JACK. JACK is a low-latency audio server, written for POSIX conformant operating systems such as GNU/Linux and Apple's OS X. It can connect a number of different applications to an audio device, as well as allowing them to share audio between themselves.

In the new setup, the audio file was played out by Alsaplayer, and the output was connected through JACK to Linphone's input. Alsaplayer is a PCM player, and is among an increasing number of JACK-enabled music players.

We immediately observed a significant increase in the MOS score with a sound quality that was clearly better from the speaker-microphone setup. The next section describes this and other observations in greater detail.

4. OBSERVATIONS AND ANALYSIS

On single hop paths, the MOS score was pretty consistent. Figure 1 shows the MOS score variation over time on the speaker-microphone setup.



Once the setup was changed to use JACK, there was a significant improvement in the MOS score which more accurately reflects the slight degradation over a single hop wireless path. Figure 2 shows the MOS score for a single hop wireless path when using JACK.

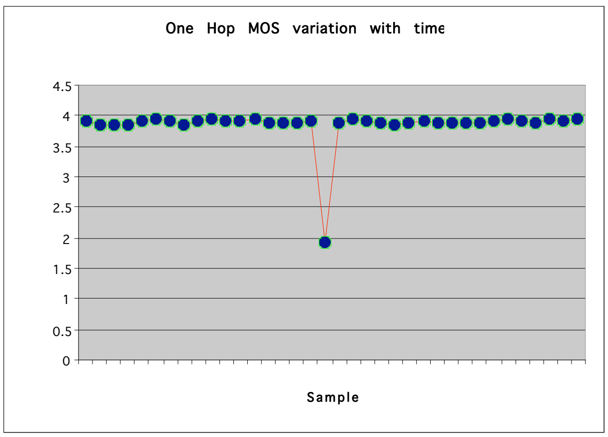
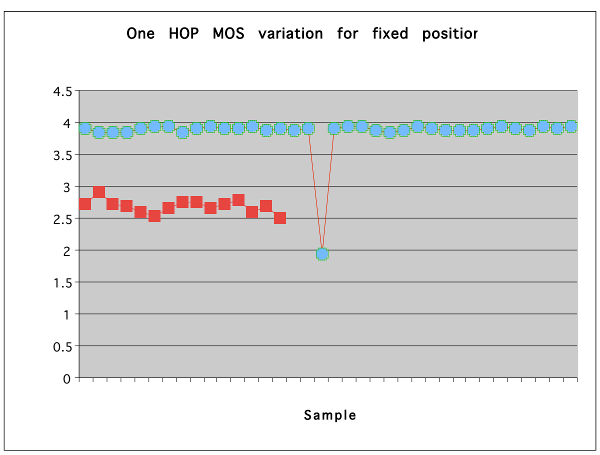
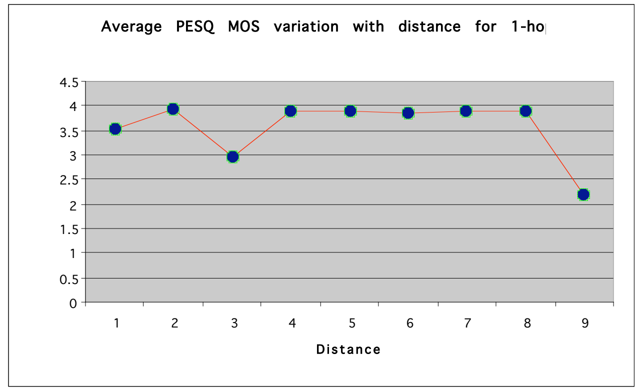


Figure 3 is a comparision between the cases using speaker-microphone and JACK.

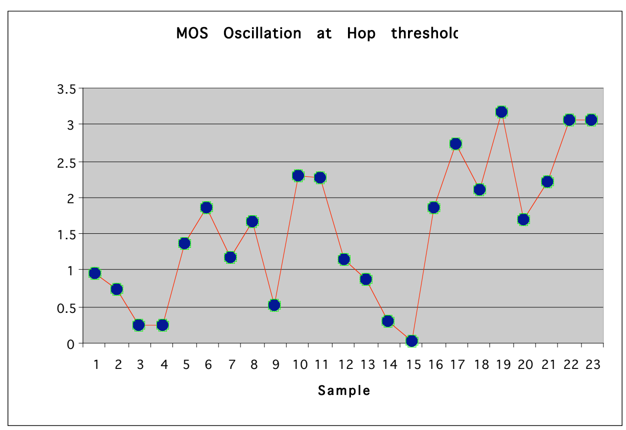


Next, we set out to measure the MOS score variation with distance. The experiment was conducted over the length of the 4th floor of the Media Lab, and the total length was divided into ten approximately equal sub-lengths. One of the interesting observations was that the MOS score varied slightly until a critical distance at which it dropped sharply, sometimes totally losing connectivity for seconds at a time. Figure 4 shows the variation in MOS score for a single hop wireless link with equal size increments in distance.



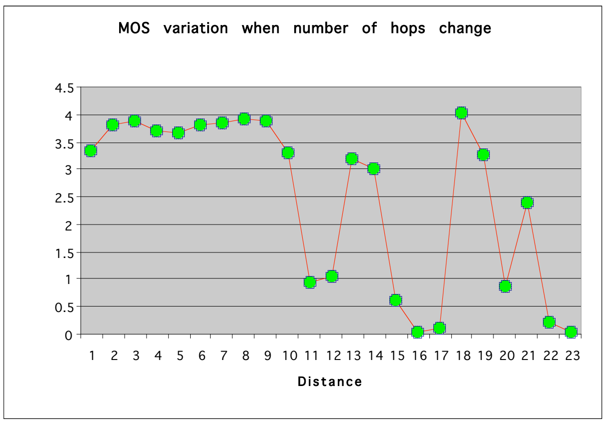
Next, we set out to measure the MOS score as a function of time for a 2 hop wireless path. However, we found using ping that the wireless path was continuously oscillating between one and two hops and we were not able to obtain a stable two hop wireless link. The path was also often asymmetric, a ping request would go on one hop on the forward link, but the reply would take 2 hops, or vice versa. AODV tried to minimize the number of hops as much as it could, so it would select a one-hop path even when a 2-hop path might have offered better voice quality.

Figure 5 shows the MOS score variation over time at a fixed distance which we consider to be the *critical distance* that represents the transition from a path reachable completely by one hop to one that oscillates continuously between one and two hops.



We surmise that the degradation in MOS score is not attributed simply to 2-hop, but to the loss of link when the path switches hops. To confirm this assumption, we will shortly carry out an evaluation of MOS score for a hardwired 2-hop link. If the average MOS score is found to be significantly better than what we obtained on our experiment, that would confirm our belief that AODV itself is responsible for the degradation of voice quality over 2 hops.

Our final experiment involved measuring the MOS score with equal increments in distance such that the path went from one to 2 hops. As expected, the path kept switching between one and two hops even far beyond the critical distance, though asymmetric paths occurred more frequently than single hop paths. Figure 5 shows the variation in the MOS score as the distance is increased by 2 steps and a sample of the full audio file is recorded at every point.



5. CONCLUSIONS AND FUTURE WORK

One of the key observations that came up again and again in our study was that obstructions are more lethal to radio signals than distance by itself. We also saw numerous examples of assymetric signalling, where a ping packet would go on one hop on the forward path and on two on the reverse path, and vice versa. However, we are also not confident of ping being an accurate indicator of the path condition, or of the flow of voice packets themselves. Many times during the experiments, it was observed that the audio quality was working fine even though the pings would stop getting replies for tens of seconds even. This happened particularly in the region when the path osciallated between one and two hops: sometimes the audio was much better than what the ping session indicated.

Interestingly, we also found that the voice quality stabilised pretty well when 2-hop was occasionally achieved. This calls for further study of voice quality using hardwired 2 and more hops. An attempt into finding the cause of oscillations between number of hops(which we believe is the minimum path metric) and fixing it and evaluating voice quality would be a valuable next step.

As pointed out by the AODV-UU team, AODV suffers from certain gray zones in which the protocol doesn't work well even though it should be expected to. This is understood to be primarily due to the different sending rates of control and data packets, so that a path that is considered workable because a Hello packet was received might just not be usable for data transmission. A solution into fixing this anomaly would help the performance of AODV.

6. ACKNOWLEDGEMENTS

I'd like to give my heartfelt thanks to the following people whose help and feedback was instrumental in carrying out this project: Dina Katabi, Rob Beverly, Sachin Katti, Kwan Lee and Hector Yuen.

7. REFERENCES

[1] Skype homepage: http://www.skype.com
[2] Hari Balakrishnan, Wireless Channel Access Protocols, Lecture notes, Chapter 11.
[3] The *FluidVoice* project, Web: http://viral.media.mit.edu/index.php?page=FluidVoice
[4] AODV homepage: http://moment.cs.ucsb.edu/AODV/
[5] Ad hoc routing protocol list, Wikipedia page: http://en.wikipedia.org/wiki/Ad_hoc_routing_protocol_list
[6] Linphone, Web: http://www.linphone.org/
[7] Kphone, Web: http://sourceforge.net/projects/kphone
[8] Session Initiation Protocol working group. Web: http://www.ietf.org/html.charters/sip-charter.html
[9] ReZound Web: http://rezound.sourceforge.net/
[10] AODV-UU, AODV routing protocol implementation created at Uppsala University. Web: http://core.it.uu.se/core/index.php/AODV-UU
[11] PESQ homepage http://www.pesq.org/
[12] Psytechnics PESQ page: http://www.psytechnics.com/site/sections/products/pesq.php
[13] Mean Opinion Score(MOS), Web: http://www.bandwidth.com/wiki/article/MOS_score
[14] JACK audio server, Web: http://jackaudio.org/
[15] ALSA homepage: http://www.alsa-project.org/
[16] Henrik Lundgren, Erik Nordström and Christian Tschudin, Coping with Communication Gray Zones in IEEE 802.11b based Ad hoc Networks, Proceedings of the 5th ACM international workshop on Wireless mobile multimedia table of contents Atlanta, Georgia, USA , 2002