AODV Performance Evaluation and Proposal of Parameters Modification for Multimedia Traffic on Wireless Ad hoc Networks

Teresa Albero-Albero[1], Salvador Santonja-Climent[1], Víctor-M. Sempere-Payá[1], Jordi Mataix-Oltra[2],

[1]Instituto Tecnológico de Informática
Universidad Politécnica de Valencia,
Camino de Vera s/n, 46022 Valencia, Spain
{maalal0, vsempere}@dcom.upv.es, salsancl@upvnet.upv.es

[2]Universidad Politécnica de Cataluña (UPC)
Av. del Canal Olímpic, s/n, 08860
Castelldefels. Spain.
jordi.mataix@upc.edu

***Abstract*—This paper presents experiments on the performance and the viability of the AODV protocol (Ad hoc On-demand Distance Vector protocol) in multi-hop ad hoc networks, for use with multimedia traffic. This type of traffic presents some particular characteristics which make it difficult for AODV to serve fluently and reliably. After this experimental study of the problems that may arise in this type of communication and its origins, we conclude that better performance is possible with the modification of some of the protocol´s parameters, without modifying the protocol algorithm. Applying these changes, experimental tests show an improvement in the performance of the protocol with video streaming.**

***Index Terms*— AODV, multi-hop, ad hoc networks, video streaming, multimedia.**

I. Introduction

Ad hoc networks represent a new doorway in the future of telecommunications, allowing wireless devices to be interconnected, thus forming communication networks without infrastructure, where each new node works as both client and router, extending the network range. On first consideration, this concept may seem unnecessary, but the reality is that the use of decentralized protocols allows networks to be formed with mobile nodes (extremely difficult for classic routing protocols) which change position continuously, and therefore change the network topology. Examples of this include vehicle networks (VANET – Vehicular Ad hoc Networks) or mobile-phones networks.

The AODV protocol (Ad hoc On-demand Distance Vector protocol) [1] allows dynamic, self initiating, multi-hop routing between the wireless nodes that participate in the Ad hoc network. It enables networks with hundreds of nodes and different mobility rates, as well as a great variety of levels of traffic. This protocol, in its experimental phase, has shown in real tests that it is capable of finding reliable routes in a short time, as well as recovering links that are lost because of node failure or node movement. However, AODV is a "Best effort" protocol, and does not assure, in any case, parameters of reaction speed or route resolution. The protocol has no support for "quality of service", however is precisely the real time and multimedia applications that need more than best effort from a protocol to find a route and to solve topology problems. In multimedia traffic, a traffic flow must be achieved which, although it can tolerate losses, must be constant (with a throughput that is equal to or higher than the coding bitrate of the stream coding, and without breaks), that is to say, it must fulfill certain QoS[1] parameters. On this point, AODV has deficiencies that generate problems for multimedia traffic, given that it is not capable of maintaining a stable flow and the user experience deteriorates. This study is organized as follows: in section II we evaluate the behavior of the AODV protocol through tests designed to allow us to analyze the problems which may affect multimedia traffic. In section III we study a modification to the parameters, which the test results show can solve the problems that may arise. In section IV, we test the modifications in a real video streaming transmission, comparing the results with those of the same transmission without the proposed modifications. Finally, the conclusions are presented in V.

II. AODV Behavior

A. AODV overview

AODV is an experimental routing protocol, defined in 2003 in RFC 3561, for mobile ad hoc networks. These are characterized by rapid adaptation to dynamic conditions in the links, low processing and memory needs (an important consideration for small devices such as sensor networks), and low network use, which together mean high scalability and performance.

This is in part due to its reactive behavior (a node only initiates a route search to a destination when it needs to transmit something, but does not conserve routes to all the

[1] Parameters recommended by CISCO: error rate (5% in video-streaming and 1% in VoIP); latency (up to 4 seconds in video-streaming and 150 ms in VoIP); throughput (21 - 320 Kbps in VoIP, and depending on the coding in video-streaming); and jitter (30 ms in VoIP, and insignificant in video-streaming depending on the buffering capacity).

possible destinations), preserving the medium for those who really need it, and avoiding large routing tables for nodes that are never going to communicate with each other. Once a route has been obtained, it will not change as long as the communication between the nodes involved is not lost, assuring the use of stable routes for the maximum possible time and continuity in the multimedia flow.

The use of sequence numbers in the route, a technique inherited from the DSDV protocol (Destination Sequenced Distance Vector Routing Protocol) [2] ensures constantly updated information, avoiding flooding of the network with packets that have been broadcast in this network area, and avoiding routing loops.

B. Experimental methodology

The tests were carried out using the AODV protocol of the University of Uppsala (version 0.9.5) [3], which is one of the most up to date and tested versions [4 –7] and is totally compatible with RFC 3561 and provides the source code to carry out modifications, which is a vital aspect for this project. We used a test environment with an IEEE 802.11b network. This technology allows the network to be set up easily, and there are a multitude of free tools to evaluate test results from systems of this type.

For the scenario, 6 nodes were used, all operating with a Ubuntu 7.10[2] system (kernel version 2.6.22-14) and the necessary tools to test, and analyze results. Conti states in [8], it is unrealistic to centre the research on networks with hundreds of mobile nodes. Normally, real testbeds with few nodes are used (5-10 or 10-20), see [5 - 7].

All cards are used with the same frequency, with default parameters, with RTS/CTS disabled, and without any type of encryption. Different topologies were simulated, depending on the test being carried out, using MAC filters with *iptables* Linux tool [9]. This allowed us to eliminate visibility between nodes while still allowing route reachability. In this way, the tests were carried out in a more controllable enviroment, the laboratory, although in this environment, it is difficult to represent exactly the degradation of signals in a field scenario. The nodes have static positions, and in the tests, we did not simulate node movement, but rather node failure and the consequent loss of links.

C. Preliminary Results

The protocol performed at a high level in the tests, showing the capacity to find routes almost instantaneously (13 ms in topologies of 2 hops, 16 ms with 3 hops…). The route search mechanism when faced with node failure in an active link is different, and in general, these situations were resolved in even less time, thanks to the re-use of previous values and the local repair function used in the protocol.

Each new node added to the linear topology formed in the laboratory increased the end to end latency by only 3 ms. However, using IEEE 802.11b technology, these types of

[2] Newer kernels showed some problems compiling the Uppsala code, because of a change in used libraries. Since a newer kernel is not necessary for our studies, we prefer not to modify the Uppsala source code, which it is not an objective of this research.

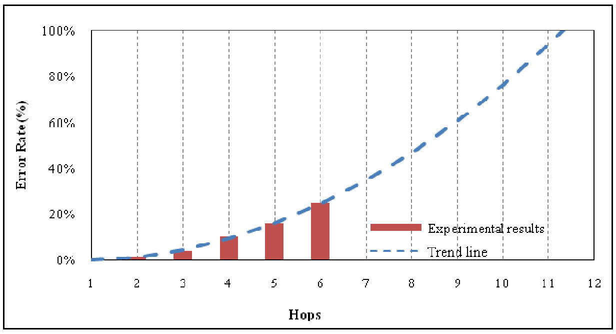

Figure 1. Error rate during 2 min ping.

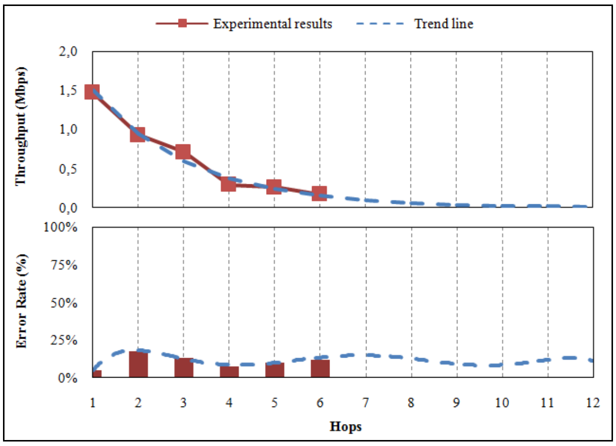

Figure 2. Throughput and error rate during FTP transfer.

networks have been shown to be not very scalable, not because of the protocol, but because of the wireless technology used and the noise in the medium.

We can conclude from the tests carried out that in applications without error control (for example ICMP, through the linux 'ping' tool), the error rate increases exponentially, see Figure1. Calculating the trend of the experimental results, we can approximate that from 12 hops onwards, the error rate reaches 100%, meaning that transmission is not possible.

Carrying out a FTP transfer, we can find an exponential decrease in the throughput alongside the number of hops (figure 2, upper graphic). The main reason for this is that each new hop represents a new fight to gain access to the medium and a full transmission between peers. Moreover, the medium is shared and as there are various nodes transmitting at the same time, longer delays in access to the medium are generated along with greater interference, which cause a decrease in the size of the TCP window. If the window is reduced in order to keep the error rate stable, (figure 2, lower graphic), the transfer rates are also decreased, according to the equation:

$$\boxed{\textit{Max. Data Transfer rate} = \textit{Window size} / \textit{RTT}}$$

It is unusual to find ad hoc networks where routes with a high number of hops are necessary [10], although it is a restriction that should be taken into account, imposed by the physical layer protocol used. Using technology with more solid modulations (such as OFDMA, Orthogonal Frequency Division Multiple Access, instead of DSSS, Direct-Sequence Spread Spectrum), with assignment of the medium by time slots (TDD, Time Division Duplex), frequency (FDD, Frequency Division Duplex), or with multi-channel technology (MIMO, Multiple Input Multiple Output), it would be possible to increase the number of hops. However, it is not the aim of the present work to increase the size of the network, but rather to improve the performance of the protocol when used with a network size that is viable for this technology.

Furthermore, during the experimental tests, we saw a delay in establishing routes in topologies with 4 or more hops. While in 2 and 3 hops the route was established in 13 and 16 ms respectively, for 4 hops, it needed 339 ms, and for 6 hops 845 ms. These values are similar to those obtained in other studies, such as that of Gupta [7], where the average with 2 hops is 7 ms, 10 ms for 3 hops, and 331 ms for 4 hops, and seems to be a time convergence problem for long routes. This behavior is due to the AODV mechanisms for avoiding flooding of the network when the distance to the destination node or to a node which knows the route is small: progressive TTL increment. The first RREQ is generated with a value of 2 in the TTL field of the IP header (see figure 3). If after a "RING_TRAVERSAL_TIME", the source node has not received a reply, it continues sending RREQ messages, increasing the TTL and RING_TRAVERSAL_TIME values, until it receives a reply, (RREP, Route Replay) or reaches the limit for retries (7 by default). Therefore, this system creates considerable delays in resolving routes to remote nodes, needing to run through the timer various times in order to establish a route, sending various RREQ messages, when the first could have found the route (with the right TTL). Tests have been carried out avoiding this system through the use of a high starting TTL, such as NET_DIAMETER-1, obtaining much faster route discovery times. However, this system is suitable for a multimedia application, as the delay is only produced the first time the route is established, and therefore is not dealt with in this paper. Once the transmission has begun, losses of connection and rerouting are corrected directly with an adapted TTL.

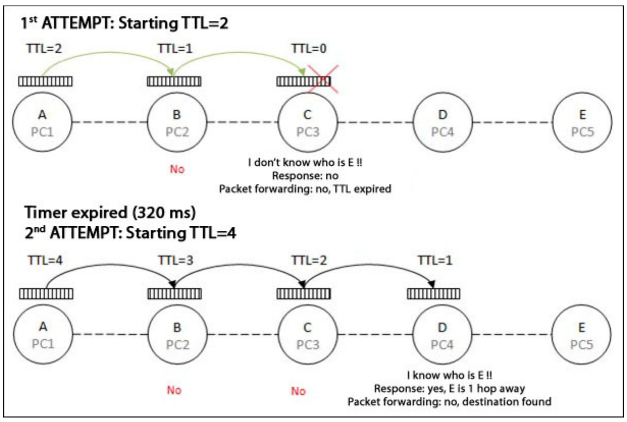

Figure 3. Tx. of several RREQ to establish a route of 4 hops.

However, during the tests carried out in the laboratory, we found two drawbacks that do affect directly the performance of multimedia applications. The first problem is the "*route intermittency phenomenon*". This effect is not usually considered in theoretical studies and simulations, but the experiments have shown that it has a significant influence on the results. AODV uses a HELLO message system to know the state of the links with its neighbors [11]. In this way, each node creates and maintains a route table with its neighbors, and those nodes with which it maintains active communication. Using this system, a node can know if it has lost a link with one of its neighbors, if it does not receive a particular number of consecutive HELLO packets from this neighbor. In an ideal situation, not receiving packets from a neighbor means that this node is no longer available (whether it is due to distance or problems with a neighboring node). However, in a real scenario with IEEE 802.11b technology, noise has a significant influence on communications. If the noise on the medium invalidates a particular number of HELLO packets, or if these packets arrive outside of the waiting time, the node will think that the link has failed, and will eliminate it from its routing table. If this happens to a node belonging to an active route, this will force the protocol to request a new route to the destination in order to continue the communication, with the consequent delay, especially if the route management packets (RREQ and RREP) also suffer from interference, and require several timer expirations and resets, increasing latency and control overhead. This problem has often meant that in previous studies AODV has been unfavourably compared to other protocols, as it generates higher losses and latency rates, which in some cases are attributed to the protocol itself. In [12] we can see a loss rate of 50% in a 3 hop link using AODV, while with the OLSR proactive routing protocol the loss rate is only 0,1%. In [13], with 3 hops, a loss rate of 51% was produced with AODV and 28% with OLSR. Moreover, the delays generated by AODV are much greater.

However, all of this is due to the problem of *route intermittency*, a problem which is not usually considered in theoretical studies and simulations, and which is caused by:

- The use of a shared medium and the consequent losses, meaning HELLO packets will also suffer collisions. If consecutive "ALLOWED_HELLO_LOSS" HELLO packets are lost, the link with the neighbor is lost.
- Given that various stations fight for use of the medium, with variable waiting times if it is already busy, it is not always possible to deliver the packets when desired and so the HELLO packets can suffer delays and arrive outside of the expected time, also generating losses of connectivity.

Both cases cause an unreal loss of connectivity, given that the topology has not changed, but the characteristics of the medium and its access protocol generate this effect arising from the HELLO mechanism. For example, during a ping of 120 s in a topology of only 2 hops, we observed 8 route changes; and in a FTP transfer of 2 minutes in a topology of 4 hops, the route changed 23 times in our experiments, changes that were unnecessary given that the topology had not changed. In [13] the average overhead of AODV protocol during an active communication is analyzed, and contrary to OLSR, the graph (see Figure 2 in [13]) shows control traffic peaks throughout the communication. These peaks correspond to the route search traffic which is continuously generated due to alternation between routes.

This effect is damaging to the continuity of the multimedia flow, because it introduces random delays which can not be anticipated, and annuls the recovery capacity of the protocol used.

The second problem detected in the tests is the reaction time of the protocol when dealing with changes in topology, which was approximately 2s. Depending on the level of mobility of the nodes within a network, this reaction time could be unacceptable, and lead to breaks in communication during the change of position of the nodes.

Due to these two problems, the AODV protocol as it is defined in the RFC, is not capable of maintaining a constant multimedia flow while fulfilling QoS parameters such as latency and jitter, and is also exposed to the intermittency phenomenon in the route as well as long reaction times when dealing with changes in topology and node mobility.

III. SOLUTION PROPOSAL

A. *Study of AODV parameters*

In the RFC there is a list of parameters that characterize the behavior of the protocol: the starting TTL of the route request packets, the route lifetimes or the number of route request retries before giving up the destination as unreachable. Among all these parameters, two are directly related to the problems observed in the experimental tests:

i) The HELLO_INTERVAL defines the time between sending two consecutive HELLO packets, and its default value is 1 s. Better reaction times are obtained by reducing this value [14], but there are two negative effects: i) collisions between data and HELLO messages, and ii) the inability to send HELLO messages at the right instants, especially for the lowest HELLO_INTERVAL values, which results in higher HELLO message intervals than expected, thus leading to false link failure detections.

ii) The ALLOWED_HELLO_LOSS defines the number of HELLO packets that can be lost before the system assumes that the link to this neighbor is currently lost. The lower this value, the faster the protocol will react to real link failures, although the HELLO packet loss due to interference or the arrival out of time of packets may also cause the elimination of a valid path. If we increase the value of this parameter, the system is more resistant to losses due to noise, but at the expense of an increase in the reaction time to real failures. Its default value is 2.

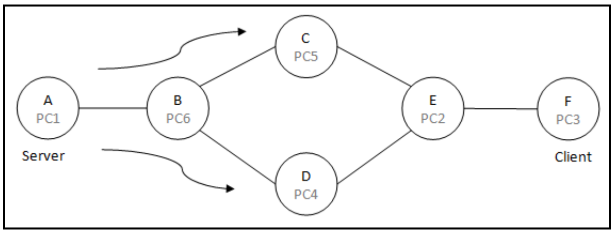

Figure. 4 Topology with the routes used in the test

The maximum reaction time of the protocol is calculated as ALLOWED_HELLO_LOSS * HELLO_INTERVAL, which is the 2s on average the protocol takes to react. The solution is to find an optimal relation between both parameters, increasing the density of HELLO packets to be received per unit time, making the protocol more robust against noise. In this way we obtain better reaction times, minimizing the *route intermittency phenomenon*. This increased density causes an increase in control packet traffic that should be analyzed.

B. *Scenario*

To check the behavior of the protocol after the proposed solutions a scenario with a topology of 4 hops and 2 possible routes between source and destination has been used, see Figure 4. All nodes are placed in a same collision domain, so the demand for media access of all nodes degrades the final service, generating the same waiting times and interference that occur in a noisy environment, such as industry or areas with wifi hotspots. Since the environmental noise is highly variable, a radio scanner and a high repeatability in the tests to achieve stable values were needed.

The tests were carried out varying the values of ALLOWED_HELLO_LOSS between 1 and 7, and the values of HELLO_INTERVAL of 1000, 200, 100, and 50 ms. We used an FTP transfer of 2 minutes, and also monitored the change of routes using the *ping* tool with the –R modifier, which informs of the route followed and the changes made during the communication. Each test was repeated 10 times and the average was calculated.

C. *Final results*

Figure 5 shows the route changes during the transfer. The lower the value of ALLOWED_HELLO_TIME, the greater the alternation between routes that occurs, due to the sensitivity of the protocol to delays and interference. For example, with HELLO_INTERVAL = 200 ms and ALLOWED_HELLO_LOSS = 2, there are 35 route changes. These changes generate cuts and communication delays that can exceed the stipulated parameters of QoS. However, from ALLOWED_HELLO_LOSS = 5, we can observe the re-

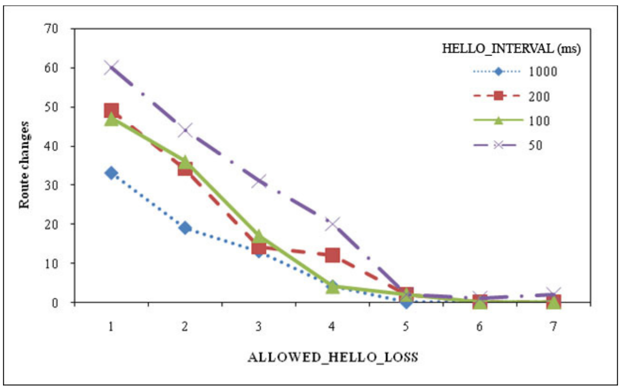

Figure 5. Route changes effected.

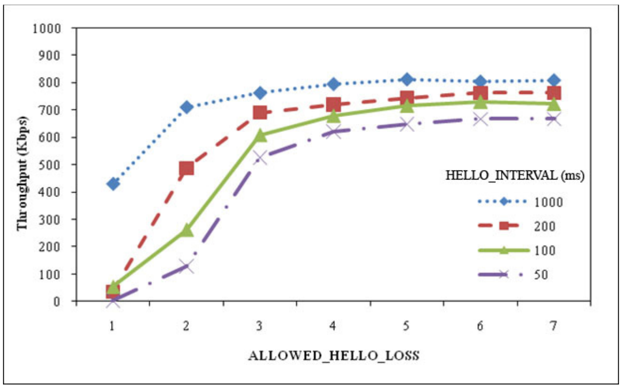

Figure 6. Throughput.

routing reach its first minimum, remaining around 0.

The test shows that this is an optimum value which provides sufficient strength to avoid the *route intermittency phenomenon* in similar scenarios. This configuration greatly benefits multimedia applications, provides a more constant flow and the recovery of lost data can be scheduled in advance with protocols such as RTP. To reduce the reaction time, this value can be combined with HELLO_INTERVAL = 100 ms (a value more stable than 50, which has presented a higher typical deviation in tests), which allows the detection of a link failure in 500 ms, instead of the 2 s default. But does this new configuration penalize communication throughput?

Figure 6 shows that the communication throughput during the test increases steadily, before levelling of after ALLOWED_HELLO_LOSS = 5. Communication with ALLOWED_HELLO_LOSS = 1 is virtually impossible, because HELLO packets that are sent so often suffer delays or interference too easily, so the system is continuously requesting routes and does not transmit data.

With the parameter values defined in the RFC (HELLO_INTERVAL = 1000 ms, and ALLOWED_HELLO_LOSS = 2), the throughput is 710 Kbps. If the ALLOWED_HELLO_LOSS value is increased to 5 and the HELLO_INTERVAL is reduced to 100 ms, the throughput obtained is 715 Kbps. This does not mean that a better througput is obtained with this combination, since tests are experimental and therefore the results have a margin of variation. However, these results show that the new configuration eliminates the *route intermittency phenomenon* and improves the reaction time without penalizing the throughput. Therefore, the experimental results show that the optimal values of these parameters for multimedia applications in the scenario used are:

ALLOWED_HELLO_LOSS = 5
HELLO_INTERVAL = 100 ms

IV. Application in Video-Streaming

To verify that the changes proposed work properly with multimedia applications, a video streaming transmission for 5 minutes has been used with the default settings of AODV. Another transmission has been used with the values obtained in the previous section. The software chosen for the streaming server and client is VLC (VideoLan media player) which enables the transcoding of the video during its broadcast, to adjust the bitrate parameters to what is needed.

Transmission is made over UDP, commonly used for this type of service because of its low header and control data overhead, and RTP as the session protocol. The video codec used is H.264, the most powerful video compression format at the moment. It offers low bitrates with high video quality, has been especially optimized for moving scenes, and has special support for streaming applications. The audio codec used is a52, known as AC3 or Dolby Digital. The system is carried to its limits of throughput using a bitrate of 704 Kbps (512Kbps for video and 96Kbps x 2 audio channels), and we evaluate the performance in the most compromised situation.

With the default settings (ALLOWED_HELLO_LOSS = 2 and HELLO_INTERVAL = 1000 ms) the video in the client suffers continuous micro-cuts in video and audio, as well as long pauses and occasional decoding failures (incomplete information) that hinder viewing. The player records the loss of 123 video frames and 199 audio buffers. The maximum difference recorded by the sniffer between two consecutive RTP packets is 4438.86 ms; the maximum jitter is 271.96ms and the average jitter is 3.55ms. Figure 7 shows the graph of the instantaneous jitter during playback of the video. Significant fluctuations that make it difficult for the player to predict the recovery of scenes generating random cuts are observed.

With the change of parameters (ALLOWED_HELLO_LOSS = 5 and HELLO_INTERVAL = 100 ms) the video in the client is fluent, without any significant cut in video or audio, which allows us a correct view of the entire stream. The player has registered the loss of 4 video frames and no loss of audio. The sniffer has shown a maximum difference between RTP packets of 341 ms, a maximum jitter of 24 ms and an average jitter of 2.44 ms. Figure 8 shows a more relaxed jitter. Although there are more peaks, due to increased control packets, the graph is much more limited, with less abrupt changes, allowing the player to make the necessary predictions about the delay of the packets and recover scenes on time.

Other parameters and bitrates tested can be found in Table 1. As we can observe, there is a better performance for higher reaction times, so it's possible to increase network throughput if an increase of reaction time is feasible. This is due to the

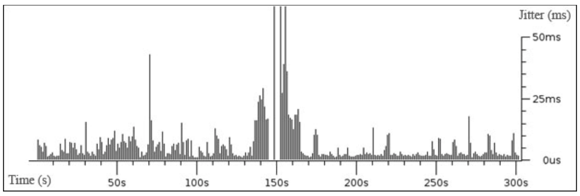

Figure 7. Instantaneous jitter with default AODV parameters.

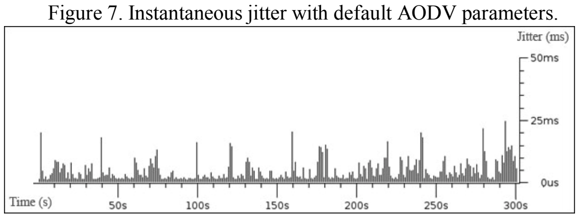

Figure 8. Instantaneous jitter with modified AODV parameters.

Test Number	AODV Parameters		Reaction Speed (s)	Bitrate (Kbps)				Delays (ms)			Losses				Route changes
	ALLOWED HELLO LOSS	HELLO INTERVAL (s)		Max. Theoretical	Video	Audio (x2)	Measured	Max. delta	Max. Jitter	Mean jitter	Sequence errors	RTP packets (%)	Video frames	Audio buffers	
1	5	1	5	384	256	64	382	216,98	16,48	2,29	9	0,00	0	0	0
2	5	1	5	576	384	96	539	133,73	19,86	2,39	29	0,07	0	0	0
3	5	0,1	0,5	576	384	96	542	261,86	14,44	2,61	31	0,05	0	0	0
4	5	0,1	0,5	704	512	96	604	341,12	24,02	2,44	47	0,12	4	0	0
5	5	1	5	768	512	128	612	148,72	20,34	2,86	31	0,00	1	0	0
6	5	0,2	1	768	512	128	634	206,24	32,11	3,54	54	0,17	8	2	1
7	5	1	5	896	512	196	725	168,17	17,19	3,11	87	0,22	11	0	0
8	5	1	5	896	768	64	734	681,91	19,46	2,95	47	0,20	163	261	1
9	5	0,2	1	896	768	64	752	908,79	106,28	9,56	224	3,05	322	507	11
10	5	1	5	1024	768	128	877	456,43	34,81	5,26	159	0,61	667	1638	1
11	5	1	5	1152	1024	64	866	608,09	3024,17	7,93	600	5,87	1191	3287	3

Table 1. Relation of reaction speeds, bitrates and route changes in tests made.

reduction of concentration of Hello packets per unit time, increasing HELLO_INTERVAL and maintaining the same ALLOWED_HELLO_LOSS for the same reliability, decompressing the network. This is useful for networks that need a fast deployment and have a temporary stationary structure, such as laptops in a meeting, where each one has its own position and there is a low movement rate.

V. CONCLUSIONS

The results obtained in laboratory tests show the difficulty of providing multimedia services with constant flow (see [15]) that fulfil the QoS parameters such as latency, jitter, error rate or constant bitrate in ad hoc networks. Specifically, the AODV protocol does not perform well when the default values of its parameters are used. This is due to the high sensitivity that wireless technology has to noise (which causes the *route intermittency phenomenon*, generating continuous delays and losses of connectivity) and high reaction time to topology changes, which affects communications with mobile nodes (MANET - mobile ad hoc networks) or constantly changing topologies negatively. Although, due to the characteristics of such networks, we cannot guarantee the QoS of multimedia applications, it is possible to make changes that bring added value to "Best effort". With the values proposed for certain parameters of the AODV protocol which were obtained from experimental tests in a noisy environment, the strength of the protocol when facing the *route intermittency phenomenon* is increased, reducing delays and increasing the reaction time when dealing with topology changes, without affecting the throughput of the communication. This improves the behavior of multimedia applications, allowing a cleaner, more fluent communication.

ACKNOWLEDGMENT

This work was supported by the MCyT (Spanish Ministry of Science and Technology) under the projects TSI2007-66637-C02-01/02, which are partially funded by FEDER.

REFERENCES

[1] C. Perkins and E. Royer: Ad-Hoc On-Demand Distance Vector Routing (AODV). July 2003. RFC 3561. http://www.ietforg/rfc/rfc3561.txt

[2] C. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers”, in Proc. ACM SIGCOMM’94, London, U. K., pp. 234–244.

[3] AODV-UU Implementation. Department of Information Technology, Uppsala University (Sweden), http://core.it.uu.se/core/index.php/AODV-UU

[4] Josip Lorincz, Nenad Ukic, Dinko Begusic, "Througput comparison of AODV-UU and DSR-UU protocol implementations in multi-hop static environments", 9th International Conference on Telecommunications - ConTEL 2007, June 13-15, 2007, Zagreb, Croatia.

[5] Eleonora Borgia, Marco Conti, Franca Delmastro, Luciana Pelusi, "Lessons from an ad hoc network testbed: middleware and routing issues", Ad hoc & sensor wireless networks, February 14, 2005, Vol. 1, pp. 125-157

[6] Eleonora B.: Experimental Evaluation of Ad Hoc Routing Protocols. In: 3rd International Conference On Pervasive Computing and Communications Workshops. (2005)

[7] Abhinav Gupta, Ian Wormsbecker, Carey Williamson: Experimental evaluation of TCP performance in multi-hop wireless Ad hoc networks. In: IEEE Computer SocietyÂ´s 12th Annual International Symposium on Modelling, Analysis, and simulation of Computer and Telecommunications Systems (2004)

[8] Marco Conti, Silvia Giordano: Multihop Ad Hoc Networking: The theory. IEEE Communications Magazine, vol. 45, Iss. 4, pp 78--86. (2007).

[9] Linux iptables tool homepage, http://www.netfilter.org/

[10] Christian Tschudin, Per Gunningberg, Henrik Lundgren, Erik Nordstrom: Lessons from Experimental MANET Research. Ad Hoc Networks Journal, special issue on Ad Hoc Networking for Pervasive Systems. M. Conti, E. Gregori (Editors).

[11] C. Gomez, M.Catalan, X. Mantecon, J. Paradells, A. Calveras, “Evaluating Performance of Real Ad hoc Networks Using AODV with Hello Message Mechanism for Maintaining Local Connectivity” IEEE 16th Symposium on Personal, Indoor and Mobile Radio Communications (2005)

[12] Eleonora Borgia and Franca Delmastro, “Effects of Unstable Links on AODV Performance in Real Testbeds”. EURASIP Journal on Wireless Communications and Networking, Vol. 2007, Iss. 1, 2007

[13] Eleonora Borgia, Marco Conti, Franca Delmastro, Enrico Gregori, Andrea Passarella, “MANET perspective: current and forthcoming technologies”, in Proceedings of the 15th IST Mobile & Wireless Communicat ions Summit, Mykonos, Greece, June 2006

[14] Ian D. Chakeres, Elizabeth M. Blding-Royer, “The Utility of Hello Messages for Determining Link Connectivity” 5th International Symposium on Wireless Personal Multimedia Communications (WPMC) (2005)

[15] S. Xu, T. Saadawi, Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? IEEE Communications Magazine vol. 39, Issue 6, pp. 130--137 (2001)