A Multipath Extension to the Dynamic Source Routing Protocol for Wireless Multimedia Sensor Networks

Naércio Magaia, Paulo Rogério Pereira, António Grilo

***Abstract* — Multipath routing allows building and using multiple paths for routing between a source-destination pair. It exploits the resource redundancy and diversity in the underlying network to provide benefits such as fault tolerance, load balancing, bandwidth aggregation, and improvement in Quality of Service (QoS) metrics such as throughput, delay and loss.**

In this work, we propose a multipath extension to Dynamic Source Routing protocol (MeDSR) appropriate for Wireless Multimedia Sensor Network (WMSN) and study its performance through simulation.

First, a multipath routing algorithm was proposed, which takes into account important aspects such as selection of paths in which the transmissions between nodes of one path do not interfere with those of the other path, and the end to end delay. As the destination node receives route request packets, it groups them and responds to the source node, a response containing a set of paths to reach it. The destination node sends as many answers as the requests received whose paths were node independent between them. This is done because it is necessary to gather information about the neighboring nodes of each node that is part of the independent path. The source node selects the best paths taking into account the lowest number of neighbors in common among the different paths received. The purpose is to minimize radio interference between the paths to be used.

Finally, we used a mechanism for interleaving packet transmissions at the source node to further reduce interference between the multiple paths.

***Index Terms* — Wireless Multimedia Sensor Network, Multipath routing, Route Coupling**

1 INTRODUCTION

Recent advances in electronics and wireless communications have led to the development of tiny, low cost, low energy and active sensors. Sensor nodes consist of sensing, data processing and communication components.

Wireless Sensor Networks (WSN) [1] are ad-hoc networks that can be established with no need for a pre-existent communications infrastructure. WSN nodes (sensors or actuators) collaborate to forward sensor data hop-by-hop from the source node to the sink nodes and vice versa.

By adding multimedia capabilities to sensors (CMOS cameras and microphones), sensor networks are able to capture multimedia contents from the environment, enabling the development of Wireless Multimedia Sensor Networks (WMSN) [2], i.e. networks of wirelessly interconnected devices that allow retrieving video and audio streams, still images and scalar data. WMSN enable several new applications such as: multimedia surveillance sensor networks, storage of potentially relevant activities, traffic avoidance, enforcement and control systems, advanced health care delivery, automated assistance for the elderly and family monitors, environmental monitoring, person locator services, industrial process control and many others.

A multimedia content such as a video stream requires a transmission bandwidth that is orders of magnitude higher than those typically necessary for a scalar sensor. So the use of a single path routing protocol may not provide enough performance for demanding applications.

Through multipath routing (i.e. using more than one path in parallel), load can be balanced and network capacity can be leveraged by increasing spatial reuse. According to [3], spatial reuse refers to the scheduling of multiple (mutually non-interfering) transmissions simultaneously when all the links are operating in the same channel.

The maximum achievable throughput improvement of multipath is determined by the achievable degree of spatial reuse rather than the number of paths used, and having more than 2 paths does not significantly improve the performance, as the source and destination nodes are common to all multiple paths [4].

The route coupling effect (i.e. interference between multiple paths) can limit the effectiveness of load balancing in wireless networks, even with node disjointed paths. So, it is highly desirable for multipath routing the use of independent paths, where transmissions along different paths do not interfere with each other (except at the end nodes) [4].

We propose a multipath extension to Dynamic Source

This work was supported in part by the FP7 WSAN4CIP (Wireless Sensor and Actuator Networks for the Protection of Critical Infrastructures) project under grant agreement no. 225186 as well as by national funding from FCT - Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC program funds and by the MPSat project (FCT Ref. PTDC/EEA-TEL/099074/2008).

N. Magaia, P. Pereira and A. Grilo are with INESC-ID/INESC INOV, Instituto Superior Técnico, Technical University of Lisbon, Rua Alves Redol, nº 9, 1000-029 Lisboa, Portugal. (phone: +351-213100345; fax: +351-213145843; e-mails: naercio.magaia@ist.utl.pt, prbp@inesc.pt, antonio.grilo@inov.pt).

Routing (DSR) [5] that addresses the route coupling issue. We address the route coupling issue by ensuring that the protocol uses independent paths to forward data packets, if they exist. We collect neighborhood information from received packets, without increasing control messages. An additional method to prevent route coupling is to interleave the transmissions for the alternate paths, so that the second node in each path has an opportunity to transmit each packet to the next hop without a collision.

The performance of the new protocol, as compared with DSR and Ad hoc On-demand Multipath Distance Vector (AOMDV) [6], is evaluated through simulation.

2 Related Work

In order to forward data from source to sink, routing protocols are used. Below we present a brief overview of two relevant routing protocols, the Dynamic Source Routing protocol (DSR) [5] and the Robust Multipath Source Routing protocol (RMPSR) [7] .

The DSR protocol is an on-demand source routing protocol. In source routing, each data packet contains complete routing information to reach its destination. DSR consists of two mechanisms: route discovery and route maintenance. Route discovery is the mechanism by which a source node discovers a route to a destination, typically by flooding Route REquest (RREQ) packets targeting the destination. That happens if it wants to send a data packet, and no route to that destination is available on its route cache. When a neighbor of a source receives a RREQ packet, it first checks whether the packet is intended for it or not. If it is the destination, it sends a reply back to the source after copying the accumulated routing information contained in the RREQ packet into a Route REPly (RREP) packet. Route maintenance is the mechanism by which a node is able to detect any change in the network topology. The DSR protocol has the disadvantage of being a single path routing protocol.

The RMPSR is a multipath extension to DSR. The basic idea is to discover multiple nearly disjoint routes between a source and a destination. To increase the probability of discovering multiple disjoint routes, the path selection criteria [8] includes the following properties: disjoint nodes, small distance between the primary (shortest) and the other paths, and small correlation factor. The correlation factor of two node disjoint paths is defined as the number of links connecting the two paths. As the destination receives multiple copies of RREQ packets of the same session within a time windows, route sets are built there. A route set consists of a primary route and several alternative routes. RMPSR triggers a new route discovery process before the connectivity is entirely lost, and increases the probability of discovering multiple disjoint routes at the expense of an increase in control overhead. The RMPSR protocol does not address the route coupling issue.

3 Multipath Extension to DSR

We propose a multipath extension to DSR (MeDSR), which inherits both desirable features of other routing protocol approaches, and applies some new features to address requirements of WMSN.

The idea of MeDSR is to build disjoint route sets for the source-destination pairs. Similar to DSR, we use an on-demand source routing approach. As in [8], to increase the probability of discovering multiple disjointed routes, we use a modified form of the RREQ packet forwarding scheme, i.e. when a node receives a RREQ packet, if it is the first time this RREQ packet is received or the path included in this message is node disjoint with the path included in a previously cached copy of the same RREQ packet, then the node will cache it and broadcast it again. In other cases, the node will discard this message.

As in [7], we choose to build the route sets at the destination node, since the destination node knows the entire path of all the available routes. The destination collects multiple copies of RREQ packets of the same session within a time window, then builds the route set, and returns it to the source. The route set consists only of a primary route and neighborhood information for each node of the primary route. We consider as primary route, the one that connects the communicating pair of nodes. We do not use alternative routes as in [7], because that information is obtained by overhearing messages.

As it is necessary to add the neighborhood information to determine the path's correlation factor, the destination node sends RREP packets for every node disjointed paths found in the route set. As the RREP packet is passed node by node until it reaches the source node, all nodes in the path add their neighborhood information. The neighborhood information is collected by every node during the route discovery process, as every node floods to its neighbors the received RREQ packets until the RREQ packet reaches the destination node. As nodes might fail, the neighborhood information is updated upon receiving a Route ERRor (RERR) packet or if the node tries to forward a packet and the intended node is unavailable. By doing that, we do not have to use additional specific control messages to keep a neighborhood table updated.

As mentioned before, in order to achieve maximum spatial reuse when using multiple independent paths, it is necessary for the traffic on one path to not interfere with the traffic on the other paths, as in the example of Figure 1. According to [4], as the transmission rate at the source is limited by the contention among the first 2 hops of the path, using 2 independent paths does not double the achievable throughput. So the maximum spatial reuse is achievable by the source when distributing the load between the 2 paths, by transmitting 1 packet every 2 hops, i.e. at times t, $3t$, $5t$, $7t$, $9t$… assuming t the necessary time to successfully forward a packet to the next hop node. So, when using two paths, the source transmits at time t through the first path, at time $3t$ through the second path, at time $5t$ through the first path and so on. Comparing with a single path, this provides a theoretical improvement of 50%

[4].

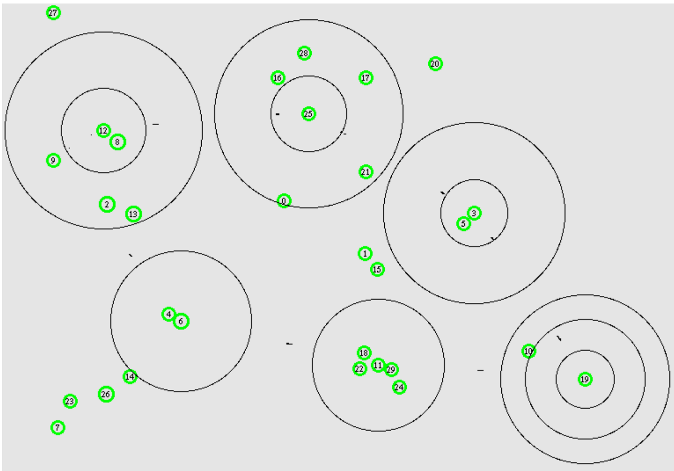

Figure 1 The use of independent node disjoint path from the source node 9 to the destination node 19. Nodes on the first path are 9, 12, 25, 3 and 19 and on the other path: 9, 6, 11 and 19.

3.1 Managing Route Request Packets

The destination node collects RREQ packets before the time window to receive RREQ packets expires. There is a tradeoff between the network size and the time window, i.e. the larger the network is, the greater the time window should be. But if we keep increasing the time window, then we would increase the route discovery latency which is not good for multimedia applications. If the time window is too short, then the destination may not have received enough RREQ packets, or if the destination issues a RREP packet too soon, it might collide with incoming RREQ packets. The collection of routes on the RREQ packets received during that time window is sorted according to the following criteria:

- Path disjointedness in relation to other paths
- Number of hops (path length)

If two or more routes in the RREQ packets with node disjoint paths are received, they are sorted according to path length. If non-disjoint paths are received too, they are discarded. A route set is built using all node disjoint routes (primary routes) received. A RREP packet is built for every node disjoint route on the route set and is sent back to the source route through that path. The route set is included in all RREP packets sent. We send RREP packets for all node disjoint paths in the route set because: (1) it is almost impossible to predict which RREP packets will reach the source node and (2) we need to collect as much neighborhood information as possible.

In case there are no node disjointed paths, and more than 2 RREQ packets were received, only 2 RREP packets are sent. They are chosen trying to minimize the number of common nodes.

3.2 Managing Route Reply Packets

We consider as expected RREP packets, all RREP packets that are sent by the destination node with the intention of collecting neighborhood information from all nodes along the path. The number of expected RREP packets is equal to the number of routes in the route set. We set a maximum of 32 neighbors that each node can add to a RREP packet. This value can be configured considering the network size.

When the source node receives a RREP packet, it waits for the time window for the arrival of other RREP packets. The RREP time window was introduced because: (1) if by the time the first RREP packet was received and the correlation factor was calculated, it would contain only little neighborhood information, (2) by starting to transmit packets with the existing routes, the risk of interference between the incoming RREP packets and outgoing traffic would be greater, causing avoidable link error messages.

After the RREP time window expires, the paths of all received RREP packets are added to the source node route cache and the neighborhood information is added to the source node neighborhood cache. If the expected RREPs are 2 or more and not all were received, we cannot ensure that the paths will not interfere with each other, even if they are node disjoint, as we did not receive all neighborhood information. If we receive RREP packets after the time window expires, the route cache will be updated with the new routes and a new multipath selection process will start.

The route and neighborhood caches are used in the multipath selection process that is explained in the following subsection.

3.3 Multipath Selection Process

The multipath selection process consists of selecting the most appropriate paths to send the data packets. According to [1], there are 2 route caches, the primary and secondary. The primary stores routes received by RREP packets and the secondary those overheard. When a route is needed to send data packets, all paths on these two caches for that destination are grouped according to the criteria described on section 3.1. The pairs are sorted according to the correlation factor. The pair with the smallest correlation factor is selected and its routes are used for multipath routing. The selected multipath routes are used until: (1) the route cache is updated or (2) a notification of dead link is received. If any of the previous events happens, the routes are discarded and new routes are selected. Figure 1 shows a situation where independent node disjoint paths are used, as discovered by our algorithms. Figure 2 shows a situation, also generated by our algorithms, where node disjoint paths are used, but the paths are not independent, as some of the intermediate nodes are too close, causing radio interference with each other. Even if node disjointed path exists, we cannot ensure that the algorithms will use them due to dropped packets.

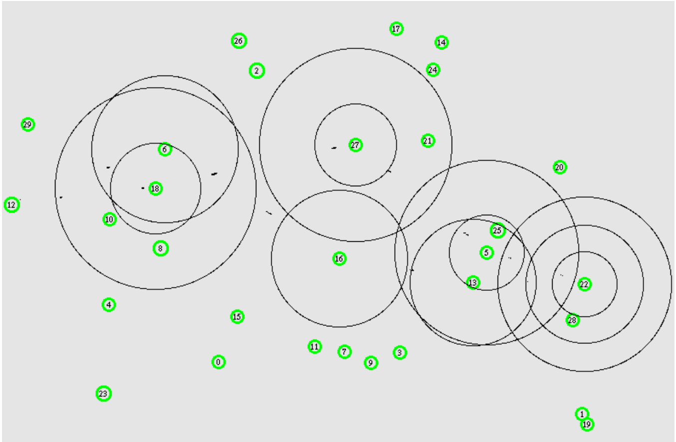

Figure 2 The use of node disjoint paths from the source node 12 to the destination node 22. Nodes on the first path are: 12, 6, 16, 13 and 22 and on the other, 12, 18, 27, 5 and 22.

4 Simulation model

We use a simulation model based on NS-2[1] [9]. We study the performance of our protocol in comparison with a single path and a multipath routing protocol, DSR and AOMDV respectively. The channel capacity of mobile hosts is set to 6Mbps. All the transmitters have the same transmission range. We use the distributed coordination function (DCF) of IEEE 802.11 for wireless LANs. We use 802.11Ext [10] as MAC protocol from NS2 as it uses an additive interference model [11], i.e. a wireless signal is decoded by treating the sum of all the other on-going signal transmissions and environmental disturbances, as noise. The default NS2 MAC protocol [12], calculates pseudo Signal to Noise Ratio (SNR) values by treating a signal that has arrived prior to the receiving signal to represent the noise on the channel. It then applies the SNR threshold (SNRT) based model to determine the successful reception of each signal. According to [12], the SNRT based model uses the SNR value directly by comparing it with an SNRT, and accepts only signals whose SNR values have been above SNRT at any time during the reception. Like that, there is no noise calculation on the default NS2 MAC protocol. The 802.11Ext interference model treats all these issues.

In our simulation, we consider 60 different network scenarios with 30 nodes randomly distributed over an area of 1500m x 1000m. The source and destination nodes were also randomly selected for each of the 60 simulation scenarios. The simulation duration is of 50s. We use 2 simulation traffic types: Constant Bit Rate (CBR) and Pareto On/Off. The size of the packets is set to 210 bytes. There is only one source/destination traffic pair. For each traffic source, we used the following data rates: 64, 128, 196, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960 and 1024kbps. For Pareto, the burst and idle average time periods are set to 250ms. A packet is dropped when no acknowledgment is received after several retransmissions or there is no buffer to hold the packet. We run all scenarios under the different protocols with the same random seed, and the results were averaged. By using the same seed, we ensured that the traffic source would behave in the same way in all similar simulation scenarios. Table I shows the simulation parameters.

[1] http://www.isi.edu/nsnam/ns/

Table I
Simulation Parameters

Network field	1500m x 1000m
Number of Sensor	30
Number of Sinks/Number of Sources	1/1
Packet Size	210 bytes
Node Energy	1J
Radio Propagation Model	Two Ray Ground
Source Data rates	64 – 1024kbps
Traffic types	CBR, Pareto On/Off
MAC Layer	IEEE 802.11a
Physical Layer data rate	6 Mbps
Simulation time	50seconds

5 Simulation Results

We evaluate the performance according to the following metrics:

- *Throughput*: This metric represents the ratio between the number of data packets that are sent by the source and received by the sink during the simulation over the simulation time.
- *Average End-to-End Delay*: The end-to-end delay is averaged over all surviving packets from the source to the destination.
- *Drops*: This metric represent the ratio between the number of dropped packets over all packets.
- *Control Overhead*: The control overhead represents the ratio between the total number of routing control packets over all packets.

We evaluate and compare the performance of the following protocols:

- DSR: Dynamic Source Routing protocol which uses a single path routing protocol.
- AOMDV: Ad hoc On-demand Multipath Distance Vector which uses a multipath routing protocol.
- MeDSR1: Normal Multipath Extension to Dynamic Source Routing protocol.
- MeDSR2: MeDSR where a 2t delay is added before sending each packet from the source node, as explained in section 3.

5.1 *Throughput*

Figure 3 and Figure 4 show the throughput comparison between DSR, AOMDV, MeDSR1 and MeDSR2 routing protocols using CBR and Pareto traffic.

We can observe that for CBR traffic, all protocols behave similarly until the source rate reaches 320kbps. This is expected as for low data rates, a single path is enough to carry all the traffic. As AOMDV is a multipath routing protocol and it is configured to find node disjoint paths, it behaves similarly to MeDSR1 and MeDSR2 until the source route reaches 448kbps. For Pareto traffic, as the traffic is less intense, as we have pause and burst periods, all protocols behave similarly until we reach a source rate of 512kbps.

As we increase the source rate, we observe that both MeDSR schemes outperform DSR. This is achieved by using simultaneously the transmission resources of multiple paths. For higher source rates, we observe a decrease on the throughput, due to collisions and contention. As for multipath we have two available paths, load can be balanced between them. But as we increase the source rate for the multipath case, the increase of throughput is not linear. As 60 different random scenarios for the 30 nodes configuration were considered, and the nodes were randomly distributed, for some scenarios it was not possible to use multiples paths as they were not available. For other scenarios, multiple paths were used, but they were not independent or in other scenarios, link disjoint paths were used. For all these situations, there was some interference between nodes, decreasing the throughput. Between the multipath routing protocols, as the source rate increases beyond 576kbps for CBR traffic, only MeDSR2 continues increasing its throughput as it tries to avoid packet collision by means of spatial reuse.

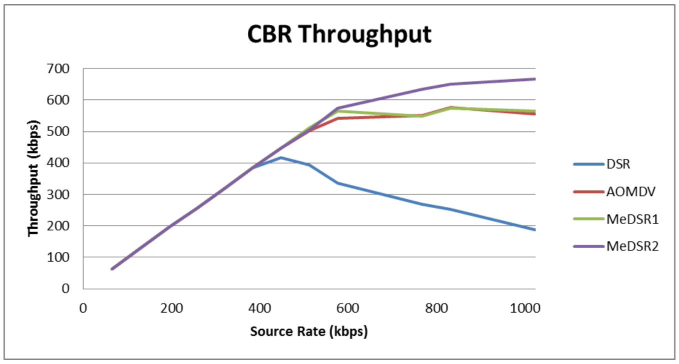

Figure 3 Throughput comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using CBR traffic.

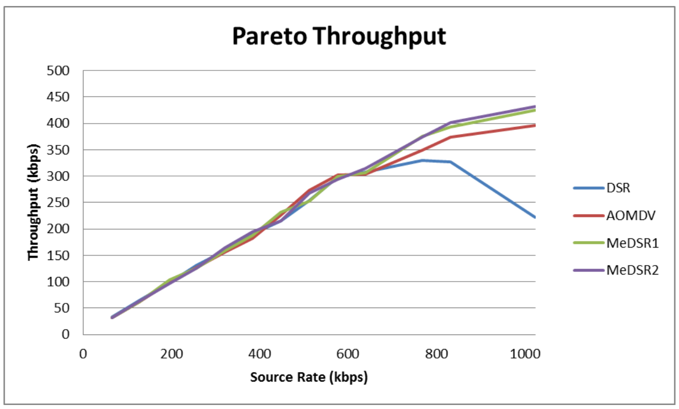

Figure 4. Throughput comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using Pareto traffic.

Table II shows a throughput gain comparison between the routing protocols. As can be seen, MeDSR2 is on average of 5,82% better the MeDSR1 for CBR traffic and 0.66% better for Pareto traffic.

Table II
Throughput Gains Comparison (%)

	CBR		Pareto	
Protocol	**MeDSR1**	**MeDSR2**	**MeDSR1**	**MeDSR2**
DSR	42.29	50.58	12.32	13.07
AOMDV	0.80	6.67	2.35	3.03
MeDSR1		5.82		0.66

5.2 Delay

Figure 5 and Figure 6 show the delay comparison between DSR, AOMDV, MeDSR1 and MeDSR2 routing protocols using CBR and Pareto traffic.

The single path routing protocol behaves better than the multiple path protocols for lower source rates. The single path routing protocol uses only the shortest path between the source and destination nodes, so it minimizes the delay.

As we increase the source rate, the single path delay increases drastically, as the number of collisions due to link saturation becomes significant. For the multipath case, as MeDSR2 adds a delay in the source to reduce collisions, for smaller source rates the delay is slightly higher than for MeDSR1. But as the source rate increases, the delay mechanism from MeDSR2 starts paying back, as it is able to reduce more collisions than MeDSR1.

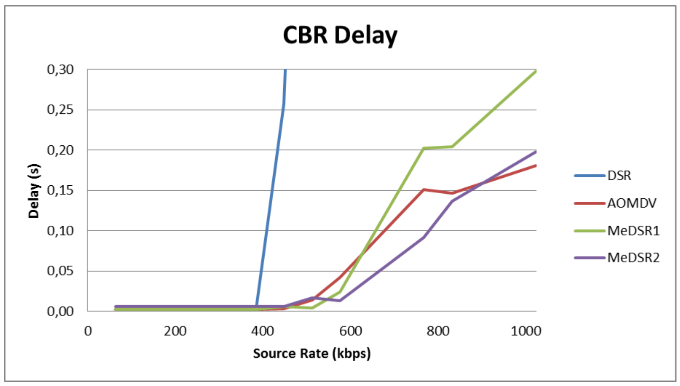

Figure 5. Delay comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using CBR traffic.

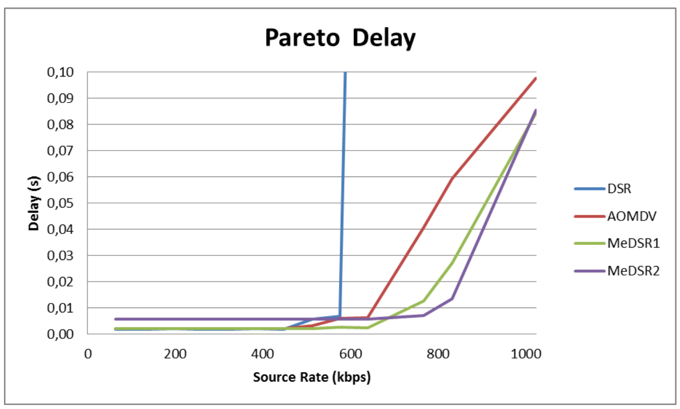

Figure 6. Delay comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using Pareto traffic

Table III shows the average delay gain comparison between the routing protocols. As can be seen, MeDSR2 is on an average of 33.88% better than MeDSR1 for CBR traffic and 12.20% worst for Pareto traffic.

Table III
Average Delay Gains Comparison (%)

	CBR		Pareto	
Protocol	**MeDSR1**	**MeDSR2**	**MeDSR1**	**MeDSR2**
DSR	93.05	95.40	93.01	92.16
AOMDV	-36.88	9.49	36.02	28.22
MeDSR1		33.88		-12.20

5.3 Packet Drop Ratio

Figure 7 and Figure 8 show the packet drop comparison between DSR, AOMDV, MeDSR1 and MeDSR2 routing protocols using CBR and Pareto traffic. The reasons for packet drops can be collisions and congestion.

By increasing the source rate above a certain threshold, the drop ratio increases too. But for multipath protocols, that increase is not as fast as for single path protocols even considering situations where we have route coupling. If we were only considering scenarios where independent node disjoint paths were used, the only limitation for multipath would be the source and destination nodes, as mentioned in section 3.

MeDSR2 outperforms both AOMDV and MeDSR1, as the source delay introduced reduces collisions and, consequently, packet drops. For smaller sources rates in both traffic patterns, our protocol performs better than AOMDV, as the latest does not take into account non interfering paths upon its route discovery process. Routing algorithms for finding node disjoint paths do not ensure non interfering paths, but routing algorithms for finding non interfering paths ensure that those paths are node disjoint when nodes on each path are distant enough to not interfere with each other.

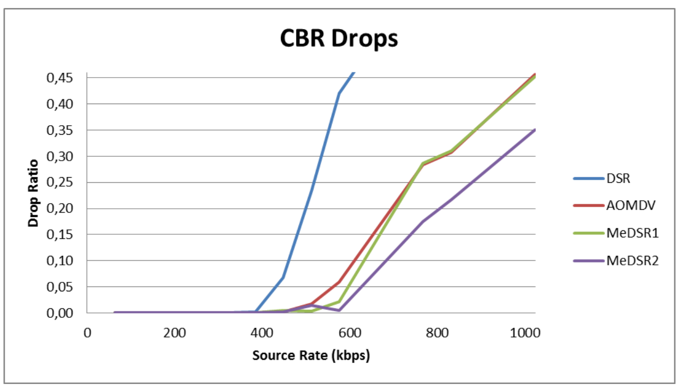

Figure 7. Drops comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using CBR traffic

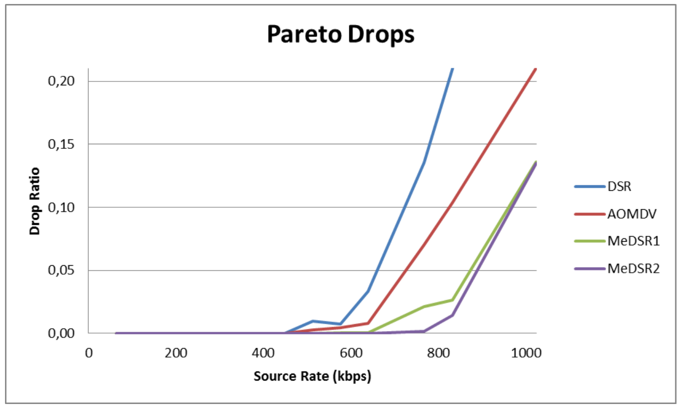

Figure 8. Drops comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using Pareto traffic

For CBR traffic, MeDSR1 tends to behave similarly to AOMDV that is also a multipath routing protocol, for higher source rates it experiences more contention. This can be concluded also by analyzing the gains it has in comparison with AOMDV. For Pareto traffic, as the traffic is less intense, as we have pause and burst periods, we obtained more gains in comparison with CBR.

Table IV shows the average packet drops gain comparison between the routing protocols. As can be seen, MeDSR2 is on an average of 29.07% better than MeDSR1 for CBR traffic and 18.35% better for Pareto traffic.

Table IV
Average Packet Drops Gains Comparison (%)

	CBR		Pareto	
Protocol	**MeDSR1**	**MeDSR2**	**MeDSR1**	**MeDSR2**
DSR	62.81	73.62	81.17	84.63
AOMDV	4.43	32.21	53.81	62.29
MeDSR1		29.07		18.35

5.4 Control Overhead

Figure 9 and Figure 10 show the control overhead comparison between DSR, AOMDV, MeDSR1 and MeDSR2 routing protocols using CBR and Pareto traffic.

For single path routing protocols, as we increase the source rate, congestion and collisions increase too. As consequence, we have an increase in control overhead as the protocol becomes unable to deliver data packets. AOMDV despite normal control messages used during route discovery operations uses too many hello messages to detect link breakages. For smaller source rates, as the number of sent data packets is small, the control overhead is considerably high. By increasing the source rate, the number of control packets becomes negligible in comparison to the total number of packets forwarded by the nodes.

Our algorithm uses neighborhood information to find non interfering paths as described in section 3.2. As we do not send periodically control messages, our protocol has less overhead even at higher source rates.

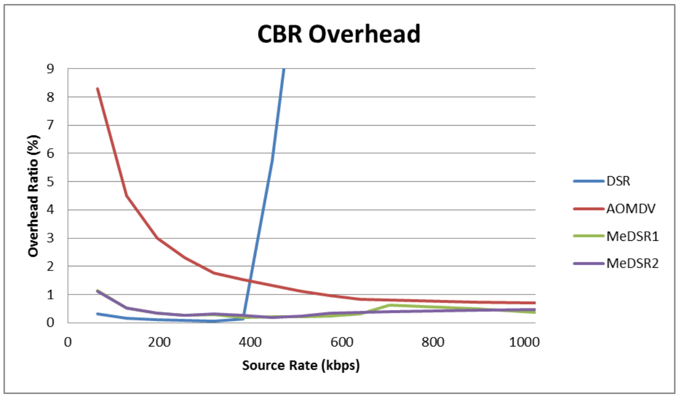

Figure 9 Control overhead comparison DSR, AOMDV, MeDSR1 and MEDSR2 using CBR traffic

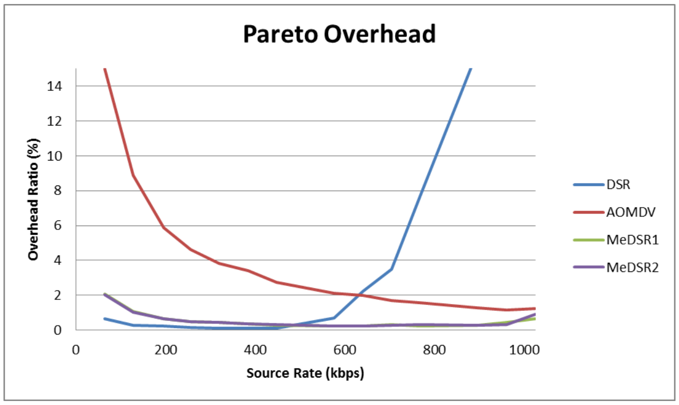

Figure 10 Control overhead comparison between DSR, AOMDV, MeDSR1 and MEDSR2 using Pareto traffic

Table V shows the average overhead gain comparison between the routing protocols. As can be seen, MeDSR2 is on an average of 0.81% better than MeDSR1 for CBR traffic and 0.99% better for Pareto traffic.

Table V
Average Overhead Gains Comparison (%)

	CBR		Pareto	
Protocol	**MeDSR1**	**MeDSR2**	**MeDSR1**	**MeDSR2**
DSR	96.98	81.30	90.29	90.19
AOMDV	96.96	81.14	85.74	85.60
MeDSR1		0.81		0.99

6 Conclusion

Multipath routing protocols have been used to enhance the performance of Wireless Multimedia Sensor Networks in different ways. Gains can be achieved in throughput, delay, drop ratio, load distribution and energy consumption.

We have developed a new multipath extension to DSR (MeDSR) and studied its performance through simulation. In MeDSR, nodes use the neighbor information they have to help choosing paths with minimal interference between them. This mechanism also avoids some overhead caused by Hello packets as in AOMDV.

As we can see by the experimental evaluation, multipath routing protocols perform better than single path ones over extreme situations, where the network load is high, as for multimedia application scenarios, since they explore the parallel use of transmission resources throughout multiple paths in the network. The average gains of MeDSR as compared with DSR are 31.83% in throughput, 93.78% in delay, 79.12% in packet drops, 85.74% in overhead and 33.88% in energy consumption.

The route coupling issue was also addressed in our work. MeDSR tries to choose the two most disjoint paths with minimal hop length. Adding a delay on the source to interleave packet transmissions, as in the MeDSR2 variant, resulted in a reduction of collisions and an improvement in terms of throughput, and a reduction on delay and packet drop ratio. The experimental evaluation has shown that MeDSR2 outperforms the normal MeDSR by an average of 3.24% in throughput, 10.84% in delay, 23.71% in packet drops, 0.90% in overhead and 9.89% in energy consumption.

The experimental evaluation has shown that MeDSR outperforms AOMDV, as we increase the network load. The average gains of MeDSR2 as compared with AOMDV are 4.85% in throughput, 18.85% in delay, 47.25% in packet drops, 83.37% in overhead and 66.8% in energy consumption.

The current solution is targeted to networks with low mobility, which is the most common situation in Wireless Multimedia Sensor Networks. As future work, we plan on evaluating and improving the stability of routes and protocol overhead in face of failures and node mobility. Another topic is using Hello messages to keep an updated view of the neighbors and study its overhead increase. A third topic is using link quality information to choose the best routes and testing higher data rates in the physical layer at the cost of reduced radio reach, but with the same interference range. A forth topic is using real multimedia traffic to evaluate the protocol performance. By using EvalVid tools-set [13] together with the NS-2[2], perceived quality and objective measure like PSNR calculation can be obtained after network simulation.

A fifth topic is using network coding. By using network coding, the nodes of a network instead of simply relaying the packets of information they receive will take several packets and combine them together for transmission. Redundant information can be sent in alternative paths and used to recover from packet loss without the need for retransmission.

Finally, it would also be interesting to evaluate protocol scalability for larger networks.

References

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E. "Wireless sensor networks: a survey." Computer Networks 38, 2002: 393–422.

[2] Akyildiz, I. F., Melodia, T., Chowdhury, K. R. "A survey on wireless multimedia sensor networks." Computer Networks 51, 2007: 921–960.

[3] Harish , A. R., Garigala, S., Raman , B., Gupta, P. “Feasibility study of spatial reuse in an 802.11 access network”. In XXVIII URSI General Assembly, New Delhi, India, October 2005.

[4] Liaw, Y. S., Dadej, A., Jayasuriya, A. "Throughput Performance of Multiple Independent Paths in Wireless Multihop Network." IEEE

[2] The interfacing code between EvalVid and NS-2 is suggested in [14]

International Conference on Communications. SA, Australia, 2004. 4157 - 4161.

[5] Johnson, D., Maltz, D., Broch, J. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), IETF, 2007.

[6] Marina, M. K., Das, S.R. “On-demand multipath distance vector routing in ad hoc networks.” Ninth International Conference on Network Protocols. 2001. 14 - 23

[7] Wei, W., Zakhor, A. "Robust multipath source routing protocol (RMPSR) for video communication over wireless ad hoc networks." IEEE International Conference on Multimedia and Expo. Taipei, 2004. 1379 - 1382

[8] Wu, K.,Harms, J. "On-Demand Multipath Routing for Mobile Ad Hoc Networks." European Personal Mobile Communications Conference EPMCC. Vienna, 2001.

[9] Fall, K., Varadhan, K. "The ns Manual (formerly ns Notes and Documentation)." The Network Simulator ns-2: Documentation. May 9, 2010. http://www.isi.edu/nsnam/ns/ns-documentation.html (accessed September 2011).

[10] Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., Hartenstein, H., “Overhaul of IEEE 802.11 Modeling and Simulation in NS-2”, Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, NY, USA, 2007

[11] Iyer, A., Rosenberg, C., Karnik, A., “What is the right model for wireless channel interference?”, Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, NY, USA, 2006.

[12] Takai, M., Martin, J., Bagrodia, R. "Effects of wireless physical layer modeling in mobile ad hoc networks". Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing. NY, USA, 2001

[13] Klaue, J.,Rathke, B. , Wolisz, A. “EvalVid - A Framework for Video Transmission and Quality Evaluation.” 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation. Urbana, Illinois, USA, 2003.

[14] Ke, C.-H.,Shieh, C.-K., Hwang, W.-S., Ziviani, A. “An Evaluation Framework for More Realistic Simulations of MPEG Video Transmission.” Journal Of Information Science And Engineering 24 [2008]: 425-440.