High Throughput Low Coupling Multipath Routing for Wireless Multimedia Sensor Networks

NAÉRCIO MAGAIA*, PAULO ROGÉRIO PEREIRA†, ANTÓNIO GRILO‡

IST/INESC-ID/INOV, Rua Alves Redol, n^o 9, 1000-029 Lisboa, Portugal

Received October, 23 2012; Reviewed March, 19 2013; Accepted: March, 20 2013

This paper proposes a new high throughput low coupling multipath routing protocol for Wireless Multimedia Sensor Networks and describes its design, implementation and performance analysis.

Our approach attempts to improve throughput by means of the Expected Transmission Count (ETX) metric and chooses routes with minimum interference among them by using a modified version of the Correlation Factor (CF).

The ETX metric represents the expected total number of transmissions (including retransmissions) required to successfully deliver a packet to the destination node. ETX allows finding high-throughput paths on a multi-hop wireless network. ETX incorporates the effects of link loss ratios, asymmetry in the loss ratios between the two directions of each link, and the interference among the successive links of a path.

The radio interference when using multiple simultaneous routes can greatly degrade the network performance. The CF is determined using neighbor information, allowing to find routes with minimum interference among them.

Our simulation results show that the proposed high throughput low coupling Multipath extension to the Dynamic Source Rout-

---

* email: `naercio.magaia@ist.utl.pt`

† email: `prbp@inesc.pt`

‡ email: `antonio.grilo@inov.pt`

ing (HTLC-MeDSR) protocol improves Quality of Service (QoS) metrics like throughput, delay, packet loss and energy efficiency in comparison with our preliminar Multipath extension to the Dynamic Source Routing (MeDSR) protocol.

Key words: Multipath Routing, Correlation Factor, Expected Transmission Count, Wireless Sensor Networks, Quality of Service.

1 INTRODUCTION

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 nodes to the sink nodes and vice versa. Sensor nodes consist of sensing, data processing, communication, storage and energy components. In order to minimize the cost, these sensor nodes are tiny devices with significant resource constrains.

Traditionally, WSNs have been used for monitoring applications based on low-rate data collection with short periods of operation. Recent advances in image sensor technology have enabled the use of audio and video sensors, which have the ability to retrieve, store, process, fuse and/or correlate multimedia data, originated from heterogeneous sources (multimedia sensor nodes, scalar sensor nodes, etc), resulting in a new class of WSNs, called Wireless Multimedia Sensor Networks (WMSN) [2]. WMSN can be designed for real-time applications, which demand strict deadline, low delay, high throughput and reliability, and non-real time applications, which usually require medium to high bandwidth, low losses, etc. The transmissions in real time and non real time of multimedia and/or scalar data may have different Quality of Service (QoS) requirements, such as bounded latency or delay, throughput, jitter, energy efficiency, and reliability, depending on the application. Some examples of real-time mission critical and monitoring applications are search and rescue, security surveillance, patient monitoring [3].

Multipath routing allows building and using multiple paths for routing between a source and a destination, by exploiting the resource redundancy and diversity in the underlying network to provide benefits such as fault tolerance, load balancing, bandwidth aggregation, as well as improvement in QoS metrics such as throughput, delay and losses [4]. As sensor nodes can be used to monitor events, where applications may require high data rates (e.g. multimedia applications) not achievable with a single path, multiple flows may be used, through different paths, so that the effective data rate at each path is reduced, and the application can be supported. Another example is splitting a large burst of data, resulting for example from an I-frame into several small fragments preventing buffer overflow at intermediate sensor nodes. These applications make the use of multiple paths in WMSN recommended [2].

The use of multiple paths between a source and a destination node, despite its benefits, has some disadvantages like the route coupling. The latter is due to the radio interference or contention among routes, and may have a serious impact on the performance of the multipath routing protocol even if routes are node disjoint, when routes are not carefully selected. We propose several mechanisms to minimize the route coupling issue.

The routing metric most commonly used by ad hoc routing protocols is minimum hop-count. By minimizing the hop-count, the distance traveled in each hop is maximized. This may contradict the performance requirements of WMSN applications, since longer distances reduce the signal strength, resulting in additional transmission errors.

It has been stated in [5] that packet loss due to collisions are often misinterpreted as link failures, which are called False Routing Failure (FRF). As a consequence, the routing protocol attempts to find an alternative path even though the current path is still valid.

In this paper, we present the design, implementation, and evaluation of the High Throughput Low Coupling Multipath extension to the Dynamic Source Routing (HTLC-MeDSR) protocol. We propose a cross layer approach to the FRF problem involving the Medium Access Control (MAC) and the routing layers. At the routing layer, we propose several enhancements to our preliminary Multipath extension to the Dynamic Source Routing (MeDSR) [6] protocol, by allowing it to use the Expected Transmission Count (ETX) [7] metric and by including second degree neighbors (neighbor of neighbor's) information in the Correlation Factor (CF) calculation. ETX finds paths with the fewest expected number of transmissions (including retransmissions) required to deliver packets all the way to their destination. The primary goal of the ETX design is to find paths with high throughput, despite some losses [7]. The CF allows finding paths with low route coupling if they exist. We apply a scheme to distinguish collisions from route failures. By increasing the maximum number of MAC retransmissions (Short Retry Limit - SRL), we reduce packet loss, and we use probe packets to identify routing failures. This scheme reduces unnecessary route maintenance operations. In order to demonstrate the effectiveness of our approach, this paper presents simulations comparing HTLC-MeDSR with the Dynamic Source Routing (DSR)

[8], MeDSR and Ad hoc On-demand Multipath Distance Vector (AOMDV) [9], and other routing protocols. These measurements show that our approach improves metrics such as throughput, delay and loss rate in comparison with those routing protocols.

In terms of WMSN QoS requirements, HTLC-MeDSR addresses throughput (by using the ETX), bounded latency and reliability (by means of CF and SRL modification), and energy efficiency (by using probe packets to reduce unnecessary route maintenance operations).

The paper proceeds in Section 2 with a description of related work. Section 3 evaluates the SRL in MeDSR. Sections 4 and 5 describe the ETX and Correlation Factor metrics respectively. Section 6 describes the design and implementation of our HTLC-MeDSR protocol. Section 7 evaluates this approach using the NS-2* simulator. And finally Section 8 presents conclusions and further work topics.

2 RELATED WORK

The Dynamic Source Routing (DSR) [8] protocol is an on-demand source routing protocol. In source routing, each data packet contains complete routing information to reach its destination. When a node wants to send a data packet, and no route to that destination is available on its route cache, it starts a route discovery process. 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. 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 is a single path routing protocol.

The Robust Multipath Source Routing (RMPSR) [10] protocol is a multipath extension to DSR. The basic idea behind RMPSR protocol 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 [11] include 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

---

* http://www.isi.edu/nsman/ns/

number of links connecting the two paths. A route set consists of a primary route and several alternative routes. The primary route connects a source and a destination node, and alternative routes connect an intermediate node to a destination. The destination node collects multiple copies of RREQ packets of the same session within a time window, then builds multiple nearly disjoint route sets, and returns primary routes to the source node, and alternative routes to corresponding intermediate nodes.

The RMPSR protocol uses a per-packet allocation scheme to distribute video packets over two primary routes of two route sets. If one transmitting primary route is broken, the intermediate node that detects a broken link will send a Route Error (RERR) packet to the source node. Upon receiving the RERR packet, the source node removes the broken primary route from its route cache, and switches the transmission to another primary route.

The RMPSR protocol was designed taking into account the QoS requirements for video applications. It does not deal with issues like route coupling, as it selects nearly disjoint routes. In addition, the traffic allocation algorithm does not take into account the link quality while selecting the routes to send data, which means that routes with high packet loss may be selected, further degrading performance.

The Ad hoc On-demand Multipath Distance Vector (AOMDV) [9] protocol offers a multipath, loop-free extension to Ad hoc On-demand Distance Vector (AODV) [12]. It ensures that alternate paths at every node are disjoint. Therefore it achieves path disjointedness without using source routing.

To support multipath routing, route tables in AOMDV contain a list of paths for each destination. All the paths to a destination have the same destination sequence number. Once a route advertisement with a higher sequence number is received, all routes with the old sequence number are removed. Two additional fields, hop count and last hop, are stored in the route entry to help address the problems of loop freedom and path disjointedness, respectively.

Because AOMDV implements multipath discovery, the loop freedom guarantee from AODV no longer holds. AOMDV addresses this issue as follows. The hop count field contains the length of the longest path for a particular destination sequence number, and is only initialized once, at the time of the first advertisement for that sequence number. Hence, the hop count remains unchanged until a path for a higher destination sequence number is received. It follows that loop freedom is ensured as long as a node never advertises a route shorter than one already advertised, and never accepts a route longer than one already advertised.

To ensure that paths in the route table are link-disjoint, a node discards a path advertisement that has either a common next hop or a common last hop relative to one already in the route table. It was observed that, as long as each node adheres to this rule, all paths for the same destination sequence number are guaranteed to be link-disjoint. Node-disjoint paths can be obtained with the following additional restriction: for a particular destination sequence number, every node always advertises the same designated path to other nodes. Route maintenance in AOMDV and AODV are similar.

The AOMDV protocol attempts to find link disjoint or node disjoint routes between a source-destination pair. AOMDV does not address the route coupling issue even though it uses disjoint routes, because it does not guarantee that the routes are non-interfering. Furthermore, it does not take into account the link quality of the selected routes, similarly to RMPSR.

The Interference-Aware Multipath Routing for Video Delivery (IAMDV) [13] protocol is an on-demand interference-aware routing protocol for a single source-destination pair, that tries to build two disjoint paths without the need of any special hardware support for locatization. IAMDV consists of two rounds of route request/replies. In the primary round, the protocol discovers the shortest path between the source-destination pair, and while the RREP packet travels back to the source node, the protocol tries to block neighbors of the nodes in the shortest path, that are not supposed to participate in the routing process. In the second round, the protocol attempts to find two paths that are distant from the shortest path by twice the radio communication range.

IAMDV prioritizes video traffic by splitting video streams into I-frames, P-frames and B-frames, and sending the most important frames through the path with the smallest hop count, and the less important frames (P and B) through an alternate path. IAMDV takes into account the route coupling issue during the route discovery process, but it does not consider the link quality over the discovered/selected routes, as the protocol only considers the hop count metric for the selection of the path to forward the most important traffic.

In a previous paper we have proposed a Multipath Extension to Dynamic Source Routing protocol (MeDSR) [6]. The basic idea is to build disjoint route sets for the source-destination pair. As in [11], to increase the probability of discovering multiple disjoint routes, 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 relative to 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. The route sets are built at the destination node, since the destination node knows the entire path of all available routes. The route sets consist of a primary route, connecting a source-destination pair, and the information about all the neighbors of the nodes in the route. The neighborhood information is collected by all nodes upon the route discovery process and is added to the RREP packet as it travels back to the source node.

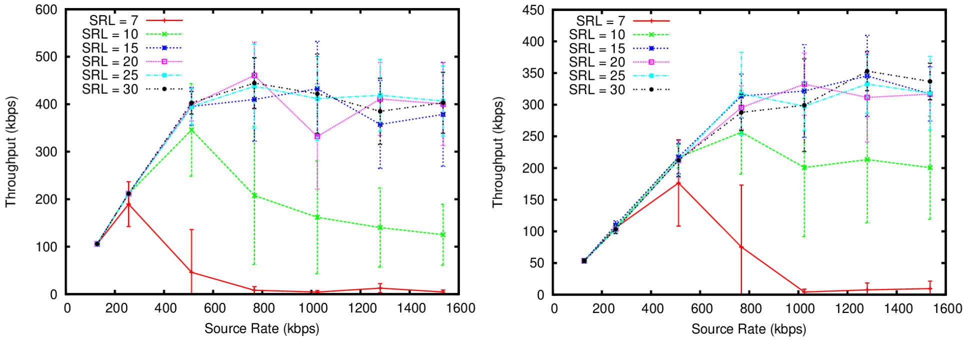
Some procedures were implemented to manage RREQ and RREP packets, at the destination and source node respectively. The destination node collects RREQ packets and builds route sets, and the source nodes collects paths from received RREP packets and uses them during the multipath selection process. During the multipath selection process, the paths are grouped according to their correlation factor, and the ones with the smallest correlation are selected. The MeDSR protocol is a multipath routing protocol that takes into account the route coupling issue, but it does not consider link quality over the selected routes, nor does it attempt to distinguish collisions from routing failures.

None of the above multipath routing protocols takes into account the link quality during the route discovery process or while selecting the routes to send data.

There is a technique used at the MAC layer that can improve routing in WSN. The Short Retry Limit (SRL) is the limit of Request-To-Send (RTS) retransmissions for a data packet over a wireless link at the MAC layer. If a node fails to transmit a RTS packet SRL times, then the corresponding data packet is discarded, and the routing maintenance process is triggered. In [14], a scheme for adaptive SRL is proposed. Upon a collision, the SRL is increased to reduce collision losses. Upon a link failure, the SRL is decreased to avoid unnecessary retransmissions in subsequent packet transmission attempts which will eventually fail. This scheme distinguishes collisions from routing failures, and treats them differently. This study only evaluates the impact of this scheme on the performance of single path routing protocols.

A common metric used in Mobile Ad Hoc Networks (MANETs) is the ETX. In [7], the design and implementation of ETX as a metric for the DSR and DSDV routing protocols is described. In addition, the modifications necessary to allow these routing protocols to use ETX are also described. Measurements taken from their test-bed demonstrated the poor performance of minimum hop-count metric, and confirmed that ETX improved throughput performance in a factor of two or more for longer paths. As a matter of fact, by avoiding the use of paths with high packet error rates, packet loss is reduced and the achieved throughput is increased. These studies were only conducted for single-path routing protocols.

This and other issues presented will be addressed by our HTLC-MeDSR



(a) CBR traffic (b) Pareto traffic

FIGURE 1
MeDSR throughput analysis in various static scenarios

protocol, as described in the following sections.

3 SHORT RETRY LIMIT IN MULTIPATH ROUTING PROTOCOLS

The IEEE 802.11 DCF (Distributed coordination Function) employs the RTS/CTS mechanism to reduce the effect of collisions due to the hidden terminal problem. It is known that the RTS/CTS exchange is useful in heavily contending environments, where many transmissions might fail due to collisions [15].

As mentioned before, if a node fails to send a RTS packet a certain number of times, called SRL, it discards the corresponding packet, and considers that the destination node is no longer available. Then, it triggers the route maintenance procedure. In the current IEEE 802.11 standard, the SRL is statically set to seven tries.

According to [14], the problem of static SRL arises from the fact that a node cannot distinguish between collision losses and mobility-induced errors. Consequently, MANETs suffer from unnecessary overhead from routing maintenance. To examine the effect of the routing instability due to the FRF, we compare the throughput achieved by the MeDSR protocol varying the SRL. We perform one simulation run over 5 scenarios with 50 nodes randomly distributed over an area of 500m x 300m using 6Mbps links, with SRL = 7, 10, 15, 20, 25 and 30, respectively, and observe the effect of SRL on

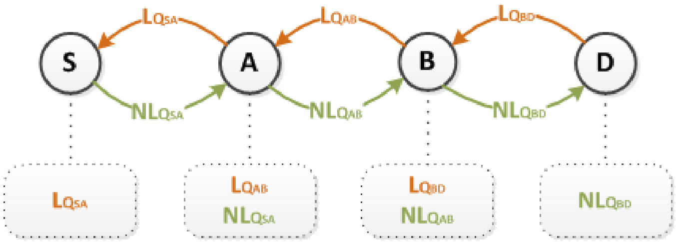


FIGURE 2
The ETX concept.

the achieved throughput. We vary the source rate from 128 to 1536 kbps, for Pareto On/Off (50% average On time) and CBR (Constant Bit Rate) traffic. In Figure 1, we present the throughput analysis for various static scenarios together with 95% confidence intervals.

For the standard value (SRL = 7), the throughput tends to zero for source rates above 768kbps, as the number of FRF is so high that almost no user traffic goes through. It is observed that throughput increases till SRL = 25 for CBR traffic (see Figure 1a), and SRL = 15 for Pareto traffic (see Figure 1b). This results show that a large SRL can improve throughput in static routing topologies. An optimal value for SRL for both CBR and Pareto traffic is 15. For Pareto, this achieves the maximum throughput and for CBR it is the third best value.

For SRL values above 30, we were not able to obtain higher gains in throughput.

4 THE ETX METRIC

The ETX metric design goal is to choose routes with high end-to-end throughput. The ETX of a link is considered as the average number of data transmissions (including retransmissions) required to send a packet over that link. For a route, the sum of the ETX for each link in the route is considered (the accumulated ETX).

The ETX of a link is calculated using the link quality and the neighbor link quality (see Figure 2). Taking node S for reference, the link quality, LQ, is defined as the measured probability for a successful data packet transmission from node A to node S. So LQ says how good a given link from a neighbor to the reference node is. As can be seen from Figure 2, the source node S computes the link quality from node A to itself, LQ_{SA}, and for the route from node S to destination D, we have 3 LQ values: LQ_{SA}, LQ_{AB} and LQ_{BD}. It is also important to know the quality of the link in the opposite direction, i.e. how many of the data packets that were sent from node S are received by node A. So, we are not only interested in the LQ of a given link, but also in the corresponding neighbor's perception of the LQ. The latter is what we call the Neighbor Link Quality, NLQ. The NLQ says how good a given link between the reference node to the neighbor is. From Figure 2, node D computes neighbor link quality from node B to itself, NLQ_{BD}, and for the route from nodes S to D we have 3 values for NLQ: NLQ_{SA}, NLQ_{AB} and NLQ_{BD}. The expected probability for a successful data packet round trip, (i.e. the probability that we successfully send a data packet to our neighbor and, on receiving it, our neighbor successfully replies with a response data packet) is $LQ \times NLQ$. Because each attempt to transmit a packet can be considered a Bernoulli trial, the expected number of transmissions is:

$$ETX = \frac{1}{LQ \times NLQ} \tag{1}$$

We propose the use of dedicated link probe packets, designated HELLO packets, to measure the LQ and NLQ of every link. Each node broadcast this link probe packets, at an average period τ (one second in our implementation). The probe packets are jittered by up to $\pm 0.1\tau$, to reduce the probability of collisions (accidental synchronization). Every node remembers the probes it receives during the last ω seconds (five seconds in our implementation), allowing it to calculate the link quality. Because the probes are broadcast, 802.11a does not acknowledge or retransmit them as in [7].

In order to calculate the ETX for a link to a neighbor, a node needs to know also the NLQ, as it can only determine the LQ by itself. For each link listed in such a probe packet, the originator of the probe packet also includes the link quality. So, node A puts the LQ values that it has determined in the probe packets, which from the reference node S are NLQ values. So, with this mechanism, every node has all the information to calculate the ETX for each link between itself and every one of its neighbors. A probe packet can include direct neighbors' information, link quality information and ETX information (if it exists). The ETX of a route is the sum of the link metrics along the path, i.e., $ETX_{SA} + ETX_{AB} + ETX_{BD}$ for Figure 2.

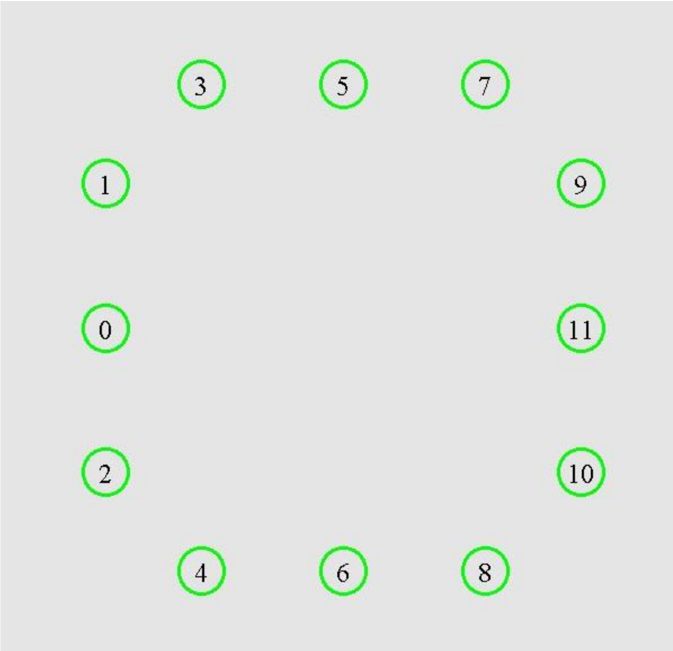


FIGURE 3
The correlation factor concept for a 12 nodes scenario.

5 THE CORRELATION FACTOR

The correlation factor (α) between two paths is defined as the number of shared neighbor nodes between the two. If there are no shared neighbor nodes between two paths, we say the two paths are unrelated ($\alpha = 0$). Otherwise, the two paths have a correlation factor of α.

Consider Figure 3, where there are two node disjoint paths from the source node 0 to the destination node 11. Two primary routes can be used to forward packets. Nodes on the first primary route are 0, 1, 3, 5, 7, 9 and 11 and on the other 0, 2, 4, 6, 8, 10 and 11. The neighborhood information for the first path is $\{0, 1, 3, 5, 7, 9, 11\}$ and for the other path is $\{0, 2, 4, 6, 8, 10, 11\}$. Excluding the source and destination nodes, these paths do not share nodes. The correlation factor for this case is $\alpha = 0$ as these two paths are unrelated.

Now consider Figure 4, where there are 100 nodes randomly distributed

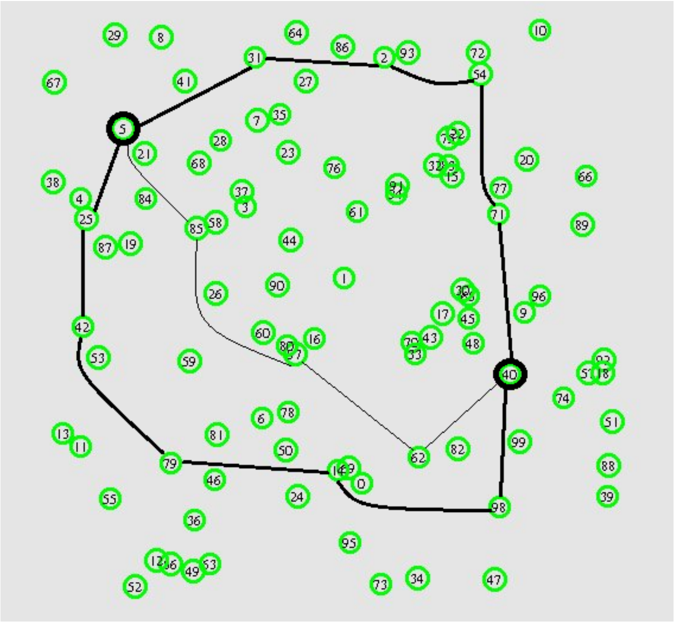


FIGURE 4
The correlation factor concept for a randomly distributed scenario with 100 nodes.

over an area of 550x550m. We assume that each node has information about its first degree neighbors (direct neighbors) and about its second degree neighbors (neighbors of its neighbors) on its neighbors cache. This information can be collected using the HELLO packets described in section 4. We choose nodes 5 and 40 as our source-destination pair. We applied a MeDSR route discovery process, and assumed that the received RREQ packets contained information about the first and second degree neighbors of each node of each discovered route. By running Dijkstra's Algorithm [14] at the destination node to find node disjoint paths from node 40 to node 5, the following paths are found: {40 17 1 3 5}, {40 33 44 7 5}, {40 43 60 19 5}, {40 9 15 23 5}, {40 30 61 28 5}, {40 45 32 35 5}, {40 48 16 37 5}, {40 70 80 58 5}, {40 62 97 85 5}, {40 65 22 27 8 5}, {40 71 54 2 31 5}, {40 89 75 86 41 5}, {40 96 77 76 68 5}, {40 82 0 6 26 4 5}, {40 98 14 79 42 25 5}, {40 39 47 73 50 90 84 5}.

Despite all found paths being node disjoint, if the selected pair of paths to use for multipath forwarding is not carefully chosen, the interference between routes can greatly degrade the performance of multipath routing. The best option would be to select first the most unrelated pair of paths. There are not unrelated paths between the ones that were found. The combinations of paths with the smallest correlation factor are $Combination1$: $\{\{$40 62 97 85 5$\}$, $\{$40 71 54 2 31 5$\}\}$ and $Combination2$: $\{\{$40 71 54 2 31 5$\}$, $\{$40 98 14 79 42 25 5$\}\}$, with correlation factors of 18 and 9, respectively. The best option is $Combination2$ because it has the lowest value of correlation factor as can be seen in Figure 4. The paths on $Combination2$ have fewer common neighbors and less interference between each other, if compared to those of $Combination1$.

There are cases where the Combinations of paths have the same Correlation Factor. In these cases, our correlation factor metric takes into account information from second degree neighbors too. We apply the same principle as before, but we consider as path's nodes the first degree neighbors of all the nodes in the path. The neighborhood information is obtained from the second degree neighbors. By doing so, we obtain the **Second degree Correlation Factor** (**SdCF**). For example, for $Combination1$ and $Combination2$, the second degree correlation factors are 76 and 68 respectively, as they have different sets of second degree neighbors. The best possible combination of discovered paths is $Combination2$, as it has the lowest SdCF.

6 HIGH THROUGHPUT LOW COUPLING MULTIPATH EXTENSION TO THE DYNAMIC SOURCE ROUTING (HTLC-MEDSR)

We propose a High Throughput Low Coupling Multipath Extension to DSR (HTLC-MeDSR) protocol, which inherits desirable features of other routing protocol approaches, and applies some new features to address WMSN QoS requirements.

The HTLC-MeDSR implementation is based on our MeDSR protocol as in [6]. MeDSR is a reactive routing protocol, in which a node issues RREQ packets only when it has data to send. RREQ packets are flooded through the network, each node appending its own address to each request it receives, and then re-broadcasting it. The request originator issues a new RREQ packet for the same destination after an exponential back-off time if no RREP packet is received.

The destination issues a RREP packet in response to the RREQ packets received. The RREP includes the route which was accumulated as the request was forwarded through the network. The RREP is source-routed back to the originator along the reverse route.

In MeDSR, FRF due to collisions are misinterpreted as link failures. HTLC-MeDSR uses probe packets to detect link failures. We modified the standard SRL to 15, as for both CBR and Pareto traffic, we could obtain a higher throughput, as explained on section 3. These two mechanisms allow the reduction of packet loss and the increase of throughput.

In order to increase the accuracy in determining FRF, each node overhears packet transmissions from other nodes, and all transmitted packets from neighboring nodes are considered probe packets, meaning that that neighbor is alive.

6.1 Obtaining neighborhood information

Each node has a neighbor's cache where the neighborhood information is kept. Our neighborhood information consists of first and second degree neighbors.

The direct neighbors (first degree neighbors) are those nodes from which we receive HELLO packets. If no probe packet is received from a given node over the last ω seconds (5 seconds in our implementation), we assume that node as failed and remove it from our first degree neighbors cache.

The first degree neighbor's information is added to the HELLO packet and broadcast. Each node that receives the first degree neighbor's information adds it to its second degree neighbor's cache. This information is used to calculate SdCF as explained in Section 5.

6.2 Obtaining ETX information

To support ETX, each node maintains a link quality and a neighbor link quality caches.

Each node sends sends periodically HELLO packets which are used to determine the link quality, as explained in Section 4. The link quality information obtained is added to the HELLO packet. Each node receiving the link quality information, adds it to its neighbor link quality cache, and uses it to calculate the ETX. The ETX information is also added to the HELLO packet. As HELLO packets are broadcast, each node has the ETX information of its first and second degree neighbors, which is added to the RREQ packet during the route discovery process.

6.3 Building Route Sets

During the route discovery process, each RREQ packet collects neighborhood and ETX information for each node in the path. With this information we can build a graph and use available path finding algorithms like Dijkstra's [16], A* [17], etc. We assume the destination (sink node) is powerful enough to run these algorithms.

HTLC-MeDSR runs Dijkstra's Algorithm at the destination node, to find the shortest path between the source and destination pair. If the algorithm returns a path, we remove all the nodes in the path except the first degree neighbors of the source and destination nodes. Then we run Dijkstra's algorithm again to find a new path. If a new path is found, and it contains the first degree neighbors of the source and destination nodes that have been previously considered, those nodes are removed together with the other nodes in the path in the succeeding runs. Therefore, any first degree neighbor is used at most twice by the algorithm. This increases the possibility of finding more node disjoint paths. This process is repeated until the algorithm does not find more paths.

After finding all the available paths, the ETX information is added to them. Then the CF is determined as described in section 5. The resulting pairs of routes are sorted according to formula 2, where ε is the path ETX value, CF is the Correlation Factor, SdCF is the Second degree Correlation Factor, and n is the number of node disjoint paths found between the source-destination pair.

$$sort\_metric = \sum_{i=1}^{n} \varepsilon_i \times \sum_{i=1}^{n} (CF + \frac{SdCF}{2}) \tag{2}$$

The first term of formula 2, corresponds to the sum of the paths ETX values. We can see from the second term that the SdCF does not have the same weight as the CF. The SdCF is useful for situations where we have some pairs of routes with the same value of CF.

The $sort\_metric$ takes into account both ETX and CF which enabled us to find the best pairs of routes, both in terms of smaller expected number of transmissions to reach the destination and smaller cross-interference.

A route set is created with at most eight routes from the best pairs of routes. The ETX and neighborhood information for all nodes in these routes is added to the route set. The route set is inserted in the RREP packet that is sent back to the source node for every route in the route set to improve tolerance to transmission errors.

6.4 Data Transmission

The source node selects the best pair of routes using the $sort\_metric$ explained in the previous section, and transmits the data packets in a round robin

FIGURE 5
Neighborhood information structure.

manner. In other words, the source node will send one data packet through the path with the smallest ETX value, and then send the next packet through the other path. In case of route failure, a new pair of paths is selected to transmit data packets. By distributing data packets among the selected paths, the energy usage is balanced.

In case of data prioritization, the most important data in sent through the path with the smallest ETX value, as it is the path with the highest throughput.

6.5 Cross-Layer Approach

HTLC-MeDSR uses a cross layer approach to address the False Routing Failure problem involving the MAC and routing layers. At the routing layer, we use the ETX and the CF values to find high throughput paths with low route coupling between them, if they exist. At the MAC layer, we modified SRL and use probe packets to identify routing failures, which reduces packet loss and unnecessary route maintenance operations.

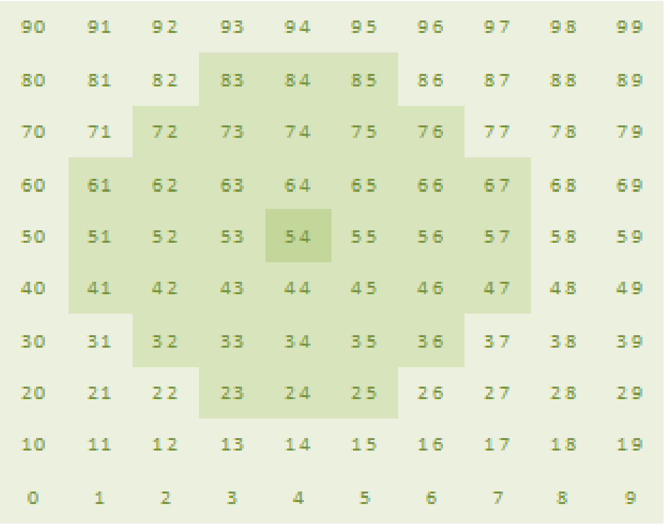
6.6 Scalability

It was mentioned before that the RREQ packets sent by the source node collect neighborhood and ETX information while traveling to the destination node. For small networks, this information is small, but for large networks it may exceed the maximum IEEE802.11 [18] frame body size of 2312 bytes.

When a packet exceeds the maximum frame body size, it is fragmented. As RREQ packets are sent in broadcast without retransmission, sending RREQ fragments would reduce the probability of finding routes at the destination node because of collisions and contention occuring during the Route Discovery process.

We use two schemes in order to reduce the size of the collected information: (1) use of node identifiers (2) data compression.

These techniques can be used individually or combined depending on the situations, and are explained on the following subsections.



90	91	92	93	94	95	96	97	98	99
80	81	82	83	84	85	86	87	88	89
70	71	72	73	74	75	76	77	78	79
60	61	62	63	64	65	66	67	68	69
50	51	52	53	54	55	56	57	58	59
40	41	42	43	44	45	46	47	48	49
30	31	32	33	34	35	36	37	38	39
20	21	22	23	24	25	26	27	28	29
10	11	12	13	14	15	16	17	18	19
0	1	2	3	4	5	6	7	8	9

FIGURE 6
10x10 Grid network.

Use of Node Identifiers

Figure 5 shows the neighborhood information structure, and Figure 6 shows a 10x10 dense grid network. If we consider that the distance between nodes and the transmission range allows node 54, for example, to have the following 36 direct neighbor nodes: 23, 24, 25, 32, 33, 34, 35, 36, 41, 42, 43, 44, 45, 46, 47, 51, 52, 53, 55, 56, 57, 61, 62, 63, 64, 65, 66, 67, 72, 73, 74, 75, 76, 83, 84, 85; node 54 neighbor information has 148 bytes[†] . By applying the same logic to all nodes in this particular network (Figure 6), i.e. from node 0 to node 99, the neighborhood information has 10880 bytes.

Our scheme assigns a node identifier to every node in the network of 8/16 bits. Here we consider 8 bits node identifiers. As all nodes in the network have 1 byte identifier, that accounts for 100 bytes. Node 54 neighborhood information occupies now just 37 bytes. This allowed a reduction of 75%. Applying the same logic to the entire network, for nodes 0 to 99, we have 2820 bytes. If we collect neighborhood information of all the network, we gain a reduction of 74.1%.

[†] An IPv4 node address has 32bits (4 bytes).

Data Compression

Even by applying the scheme described in the previous subsection, for larger networks, the addition of compression mechanisms is required to allow a RREQ packet to collect neighborhood and ETX information while it travels through the network until it reaches the destination node.

The authors in [19] survey practical data compression algorithms for wireless sensor networks, i.e. algorithms that are based on real world requirements (e.g. minimizing power consumption). According to them, data compression can be classified as (1) distributed data compression approach, which is usually applied in dense sensor networks, and (2) local data compression approach, which performs data compression locally on each node without distributed collaboration among other sensor nodes.

The local data compression approach can be divided in two techniques: lossless compression algorithms, which ensure that information is correct during the processes of compression and decompression; and lossy compression algorithms, in which some information may be lost.

Each node that deals with a RREQ packet must be able to compress neighborhood and/or ETX information, or decompress it in case of the destination node.

In our routing protocol approach, we assume the use of a lossless compression algorithm such that the information always fits in a single packet. The use of further additional compression mechanisms, such as lossy compression by using RSSI information to discard some low quality links, is left for future study.

7 SIMULATION MODEL

We use a simulation model based on NS-2. We study the performance of HTLC-MeDSR protocol in comparison with a single path and multipath routing protocols. The channel capacity of every node 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 [20] as MAC protocol simulator from NS-2 LANs, because it uses an additive interference model [6].

In our simulation, we consider two different network scenarios, with 12 (see Figure 3) and 100 (see Figure 6) nodes distributed over an area of 700m x 700m. The 12 nodes scenario has two non-interfering paths. In the 100 nodes scenario, nodes are distributed on a 10 x 10 grid with separation of

50m between them. We only consider static scenarios because they are more common in WMSN.

The simulation duration is of 210s. We perform ten runs of simulations for both scenarios, and the results are averaged. In order to get non-deterministic results across runs, different seeds were used. This will influence the received noise, as well as the traffic pattern, where applicable.

We use two simulation traffic types: Constant Bit Rate (CBR) and Pareto On/Off with transmission periods of 50% of the time, on average. 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: 128, 256, 512, 768, 1024, 1280 and 1536kbps. We present the average throughput, end-to-end delay, packet loss, control overhead, energy efficiency and path length analysis for both scenarios together with 95% confidence intervals.

Each node contains an initial energy of 1MJ. Considering a group of 12 and 100 nodes, it gives amounts of 12MJ and 100MJ overall. Since nodes participate in the sensor network, they use their energy to transmit or to receive packets, as well as while operating in standby mode. They can only participate in the network if they have energy. The values of power consumption in the four radio states, including idle, transmit, receive and sleep state are set according to the study results in [21]: P_{idle} = 1.08 mW, P_{tx} = 1.875 mW, P_{rx} = 1.3 mW, and P_{slp} = 0.045 mW.

Table 1 show the simulation parameters.

8 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.

- *Packet Loss Ratio*: 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.

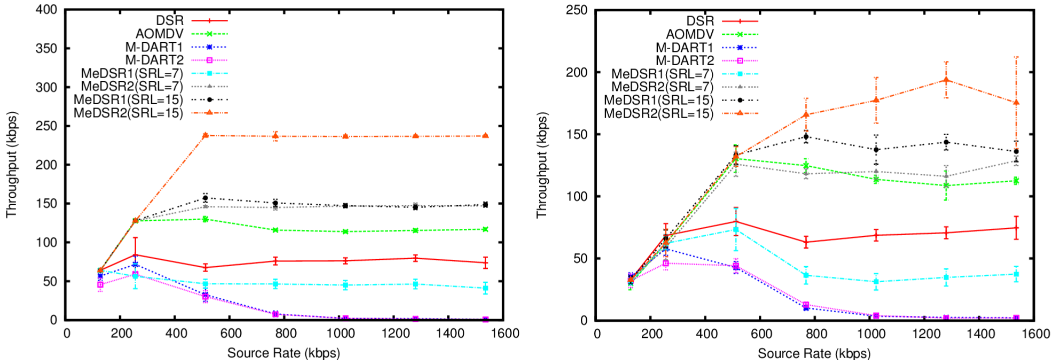
TABLE 1
Simulation Parameters

Network field	700m x 700m
Number of Sinks/Number of Sources	1/1
Packet Size	210 bytes
Idle power	1.08mW
Receive power	1.3mW
Transmit power	1.875mW
Sleep power	0.045mW
Radio Propagation Model	Two Ray Ground
Traffic types	CBR, Pareto On/Off
MAC Layer	IEEE 802.11a
Physical Layer data rate	6 Mbps
Simulation time	210 seconds
Number of Sensor	12, 100
Node Energy	1MJ
Source Data rates	128 – 1536kbps

- *Energy Efficiency*: The energy efficiency measures the energy dissipated per transmitted bit by the nodes throughout the entire simulation.
- *Path Length*: The path length is averaged over all surviving packets from the source to the destination.

We evaluate and compare the performance of the following protocols:

- DSR: Dynamic Source Routing protocol which is a single path routing protocol.
- AOMDV: Ad hoc On-demand Multipath Distance Vector which uses only one path to forward traffic, keeping the alternative paths as a backup if the main path fails.
- MDART1: Multipath Dynamic Address Routing protocol [20], using a single path.
- MDART2: MDART using two simultaneous paths to forward traffic.



(a) CBR traffic

(b) Pareto traffic

FIGURE 7
Throughput as function of source rate on scenario 1

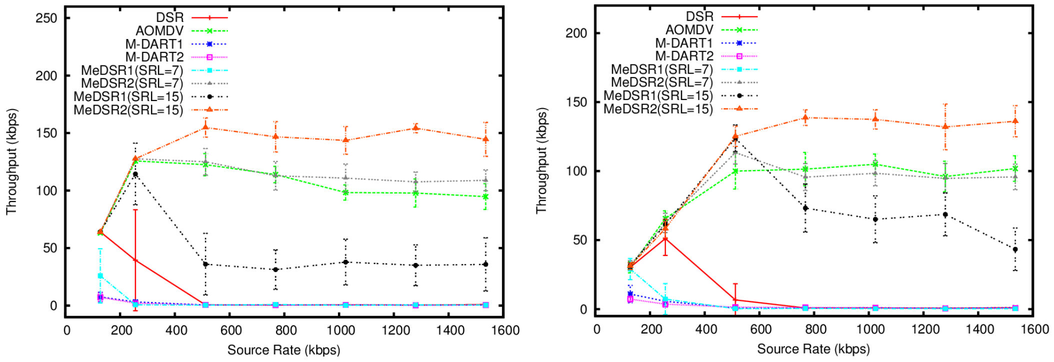
- MeDSR1: Our preliminary Multipath Extension to Dynamic Source Routing protocol [6].
- MeDSR2: High Throughput Low Coupling Multipath Extension to the Dynamic Source Routing (HTLC-MeDSR) protocol.

8.1 Throughput

It has been stated before that the simultaneous use of multiple paths to forward traffic may incur in a performance degradation due to the route coupling issues, if additional care is not taken during the multipath selection process.

Multipath routing protocols in fault tolerance configuration, like M-DART1, use only one path to forward traffic and another path for backup. For small source rates, single path protocols behave better than the ones that use multiple paths simultaneously (e.g. MDART2), because their nodes suffer less contention and collision while forwarding traffic through a single path. The same happens between DSR and MeDSR1 with normal SRL. These two last protocols use SRL to detect link failure.

In Scenario 1, MeDSR1 with normal SRL uses two independent paths to forward traffic. It has to deal with contention and collisions over these two routes, and with the increase of source rate, link failures in both paths cause additional performance degradation. Figure 7 shows that, if we increase SRL as in MeDSR1 with SRL = 15, more MAC level retransmissions are permitted



(a) CBR traffic

(b) Pareto traffic

FIGURE 8
Throughput as function of source rate on scenario 2

per node. Consequently, there is a throughput increase since more packets are delivered to the destination. This confirms the results of section 3. As AOMDV and MeDSR2 use probe packets to detect link failure, they behave better than all other protocols, since this mechanism reduces FRF. Figure 7 also shows that using multiple paths to forward traffic increases throughput, if the routes have no cross-interference. With the increase of SRL in MeDSR2, there is also a throughput increase for both CBR and Pareto traffic as for MeDSR1 with SRL = 15. For Pareto traffic (see Figure 7b), the throughput magnitude is smaller than the one achieved with CBR traffic, because Pareto traffic is less intense due to burst and pause times.

The results for scenario 2 are shown in Figure 8. Considering only normal SRL, we can see that MeDSR2 behaves almost similarly to AOMDV, performing even better if compared with other routing protocols for both CBR and Pareto traffic. The difference in performance is because, in Scenario 2, the routing protocols are able to find node disjoint routes between the source-destination pair, but the selected routes are not completely independent as the ones of Scenario 1. Consequently, the selected routes in Scenario 2, suffer more route coupling even for MeDSR2.

Figure 8 also shows that MeDSR1 with normal SRL behaves worse than DSR. This happens because of the route coupling among the selected pair of multipath routes, as the protocol has to deal with a lot of contention and collision, causing unnecessary new route discovery processes that severely degrade performance. For scenario 2, as for scenario1, the increase of SRL in MeDSR1 also causes an increase in terms of throughput in comparison with other routing protocols. The same happens to MeDSR2.

MDART performs worst, as it is not able to find usable paths with the increase of the source rate, due to excessive collisions.

TABLE 2
MeDSR2 Throughput Gains for normal SRL (%)

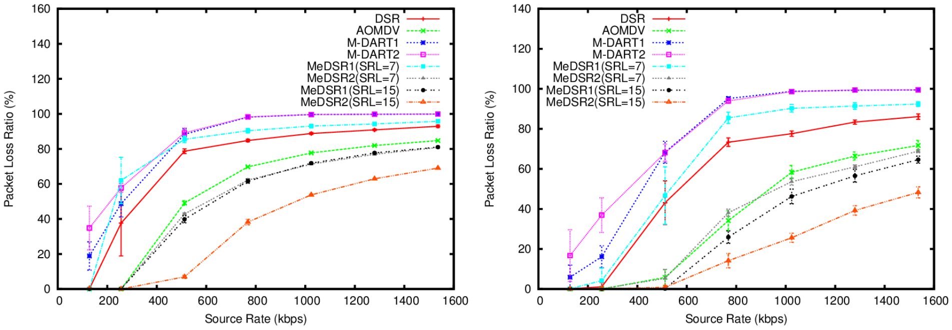
		DSR	AOMDV	M-DART1	M-DART2	MeDSR1
S1	CBR	77.1	17.8	433	531	158
	Pareto	52.9	2.7	355	389	120
S2	CBR	611	5.6	5737	6609	2513
	Pareto	536	-1.3	2774	3734	1457

Table 2 shows the average throughput gain comparison for SRL=7 between MeDSR2 and other routing protocols on Scenario 1 (S1) and Scenario 2 (S2). As can be seen, MeDSR2 is on average 1147% better than protocols that use one path to forward traffic (e.g. DSR, AOMDV and M-DART1) for CBR traffic and 620% better for Pareto traffic. It is also on average 2453% better than protocols that use two paths to forward traffic (e.g. M-DART2 and MeDSR1) for CBR traffic and 1425% for Pareto traffic.

TABLE 3
MeDSR2 Throughput Gains for SRL = 15 (%)

		DSR	AOMDV	MeDSR1		MeDSR2
		SRL=7	SRL=7	SRL=7	SRL=15	SRL=7
S1	CBR	164	75.6	284	46.1	49.0
	Pareto	105	37.7	195	17.8	34.1
S2	CBR	779	30.6	3132	164	23.6
	Pareto	716	26.5	1895	65.1	28.2

Table 3 shows the average throughput gain comparison between MeDSR2



(a) CBR traffic

(b) Pareto traffic

FIGURE 9
Packet Loss Ratio as function of source rate on scenario 1

with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both Scenarios. As can be seen, MeDSR2 with SRL = 15 is on average 616% better than other MeDSR implementations for CBR traffic and 373% better for Pareto traffic.

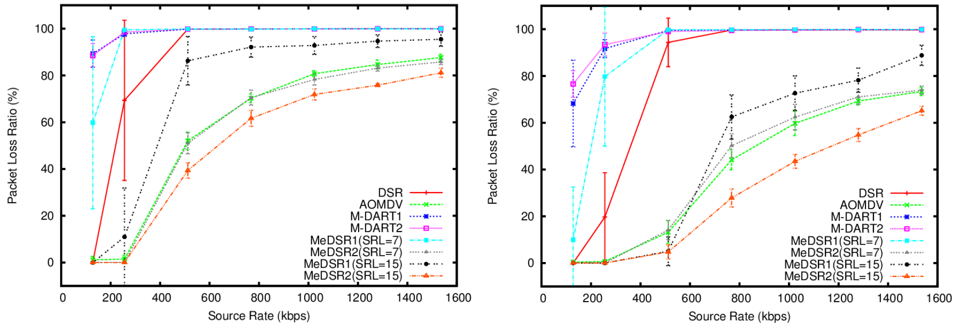
In comparison with DSR, MeDSR2 is on average 472% better for CBR traffic and 411% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 53.1% better for CBR traffic and 32.1% better for Pareto traffic.

8.2 Packet Loss Ratio

Figure 9 shows the packet loss ratio for scenario 1. For normal SRL, MeDSR2 presents less packet loss than all other protocols, because it uses node independent routes and probe packets to detect link failure. With the increase of SRL, MeDSR1 and MeDSR2 use the additional attempts to successfully deliver more packets to the next hop, at the expense of an additional delay.

Figure 10 shows the packet loss ratio for scenario 2. The results show that single path routing protocols tend to perform better than multipath routing protocols regarding the packet loss ratio. As examples, we have DSR and MeDSR1 with normal SRL, and both version of MDART. We also noticed that for smaller source rates, single path routing protocols like DSR and MDART1 suffer less from concurrent transmissions than multipath routing protocols like MeDSR1 with normal SRL and MDART2.

AOMDV and MeDSR2 with normal SRL, despite being multipath routing protocols, behave similarly in terms of packet loss, since they use a similar



(a) CBR traffic

(b) Pareto traffic

FIGURE 10
Packet Loss Ratio as function of source rate on scenario 2

mechanism to avoid FRFs. The increase of SRL enables the protocols to reduce even more packet loss, since there are more transmission attempts.

TABLE 4
MeDSR2 Packet Drop Ratio Gains for normal SRL (%)

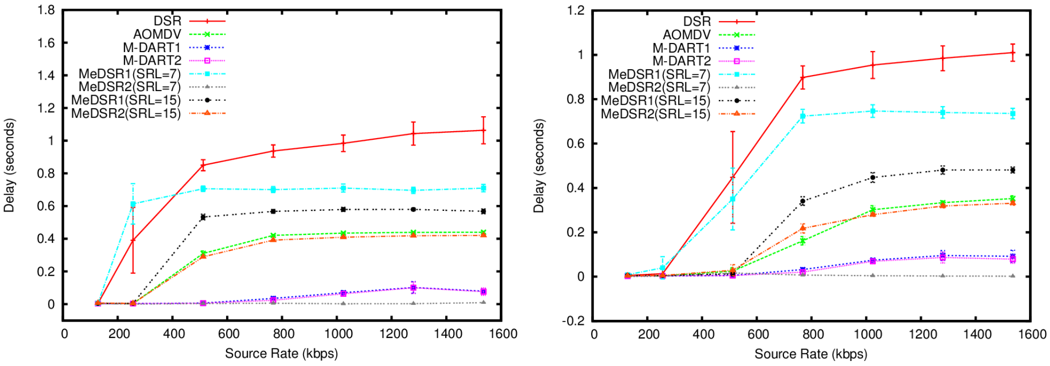
		DSR	AOMDV	M-DART1	M-DART2	MeDSR1
S1	CBR	29.4	8.0	39.2	42.3	34.8
	Pareto	37.7	3.9	53.0	55.7	43.7
S2	CBR	35.2	2.4	46.3	46.4	44.0
	Pareto	46.5	4.2	58.7	59.5	53.9

Table 4 shows the average packet loss gain comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. As can be seen, MeDSR2 is on average 26.7% better than protocols that use one path to forward traffic for CBR traffic and 34% better for Pareto traffic. It is also on average 27.9% better than protocols that use two paths to forward traffic for CBR traffic and 53.2% for Pareto traffic.

Table 5 shows the average packet loss comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15,

TABLE 5
MeDSR2 Packet Drop Ratio Gains for SRL = 15 (%)

		DSR	AOMDV	MeDSR1		MeDSR2
		SRL=7	SRL=7	SRL=7	SRL=15	SRL=7
S1	CBR	51.2	36.4	54.9	30.4	30.8
	Pareto	64.8	45.7	68.2	33.7	43.5
S2	CBR	42.0	12.7	49.9	30.1	10.5
	Pareto	61.4	24.8	66.7	35.4	27.8



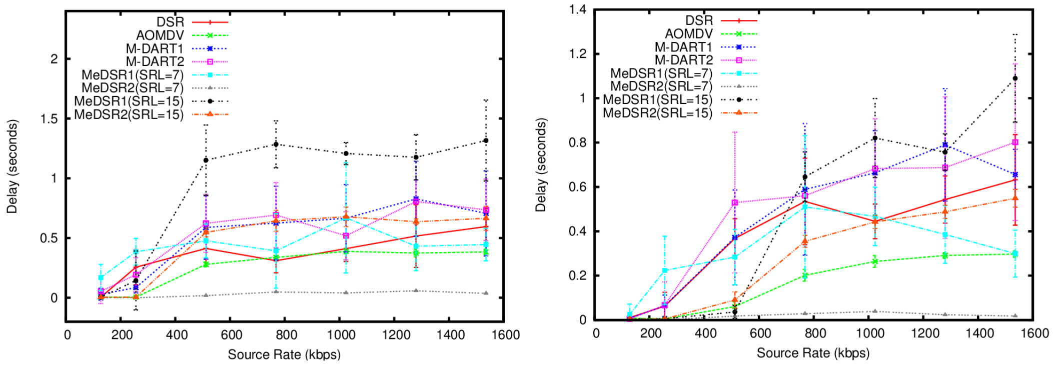
(a) CBR traffic

(b) Pareto traffic

FIGURE 11
Average End-to-End Delay as function of source rate on scenario 1

DSR and AOMDV for both scenarios. As can be seen, MeDSR2 with SRL = 15 is on average 34.4% better than other MeDSR implementations for CBR traffic and 45.9% better for Pareto traffic.

In comparison with DSR, MeDSR2 is on average 46.6% better for CBR traffic and 63.1% better for Pareto traffic. If compared to AOMDV, it is on average 24.6% better for CBR traffic and 35.3% better for Pareto traffic.



(a) CBR traffic

(b) Pareto traffic

FIGURE 12
Average End-to-End Delay as function of source rate on scenario 2

8.3 Average End-to-End Delay

For scenario 1, DSR presents a low end-to-end delay for small values of source rates for both CBR and Pareto traffic (see Figure 11). Above these values, DSR presents a considerable increase in the end-to-end delay. This happens because it uses only one path to forward traffic and, if that path becomes broken, salvaged packets suffer additional delay, since intermediate nodes might not have alternative paths to the destination node. In these situations, the route maintenance procedure initiates a route discovery process attempting to discover new routes to the destination node.

MeDSR1 behaves better because it uses multiple paths to forward traffic. By using multiple paths alternatively, there is less load in each path, resulting in a smaller end-to-end delay. We can see also that MeDSR1 with SRL = 15 behaves better than the normal SRL one, on scenario 1, due to the use of more MAC retransmissions, as explained before.

MDART versions have less end-to-end delay than AOMDV and MeDSR2 with SRL = 15, because only a small subset of packets sent by the source node is received by the destination node, and these received packets have small end-to-end delay. The increase of SRL despite its benefits, like the reduction of packet loss, has some disadvantages, such as the increase of end-to-end delay. This is caused by the additional MAC level retransmissions over sucessive nodes in a path. This can be seen in Figure 11 and Figure 12, as

MeDSR2 with normal SRL has the smallest values of end-to-end delay. Since MeDSR2 with normal SRL and AOMDV have similar values of throughput (see Figure 8 or Table II), MeDSR2 is a good choice for small end-to-end delay applications. If we compare MeDSR2 with normal SRL on both scenarios, we can see that the set of changes proposed were useful in selecting paths with smaller end-to-end delay.

Table 6 shows the average delay gain comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 is on average 94% better than protocols that use one path to forward traffic for both CBR and Pareto traffic. It is also 94.3% better on average than protocols that use two paths to forward traffic for CBR traffic and 93.9% for Pareto traffic.

TABLE 6
MeDSR2 Average End-to-End Delay Gains for normal SRL (%)

		DSR	AOMDV	M-DART1	M-DART2	MeDSR1
S1	CBR	99.4	98.7	91.0	90.3	99.3
	Pareto	99.1	96.9	88.3	86.6	98.9
S2	CBR	92.0	88.7	94.3	94.5	93.3
	Pareto	95.3	88.7	95.9	96.2	94.2

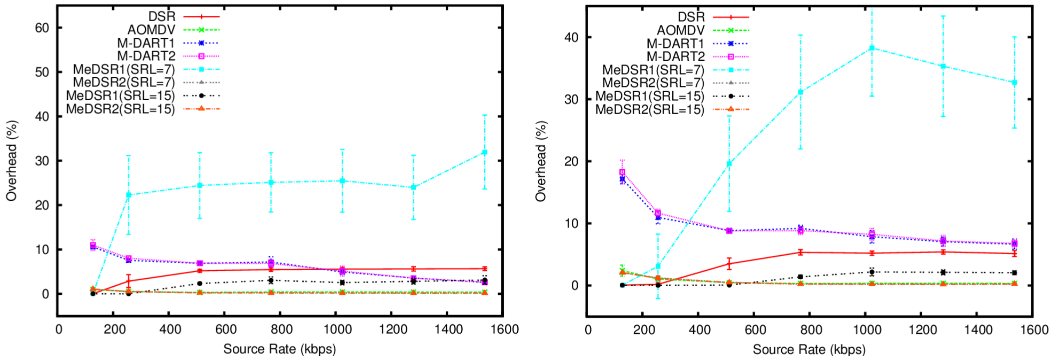
Table 7 shows the average delay gain comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. As can be seen, MeDSR2 with SRL = 15 is on average 31.5% better than other MeDSR1 versions for CBR traffic and 36.9% better for Pareto traffic.

MeDSR2 with SRL = 15 despite having a considerable gains in comparison with both MeDSR1 implementations, does not perform as good, in terms of delay, if compared with the normal SRL version. MeDSR2 with normal SRL is on average 4304% better than SRL= 15 version for CBR traffic and 2314% better for Pareto traffic.

In comparison with DSR, MeDSR2 is on average 18.1% better for CBR traffic and 50.6% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 37% worse for CBR traffic and 35.9% worse for Pareto traffic.

TABLE 7
MeDSR2 Average End-to-End Delay Gains for SRL = 15 (%)

		DSR	AOMDV	MeDSR1		MeDSR2
		SRL=7	SRL=7	SRL=7	SRL=15	SRL=7
S1	CBR	63.1	5.3	52.1	31.6	-7121
	Pareto	72.5	-0.4	63.1	33.2	-3213
S2	CBR	-26.9	-79.3	-7.1	49.4	-1487
	Pareto	28.6	-71.3	11.9	39.6	-1415



(a) CBR traffic

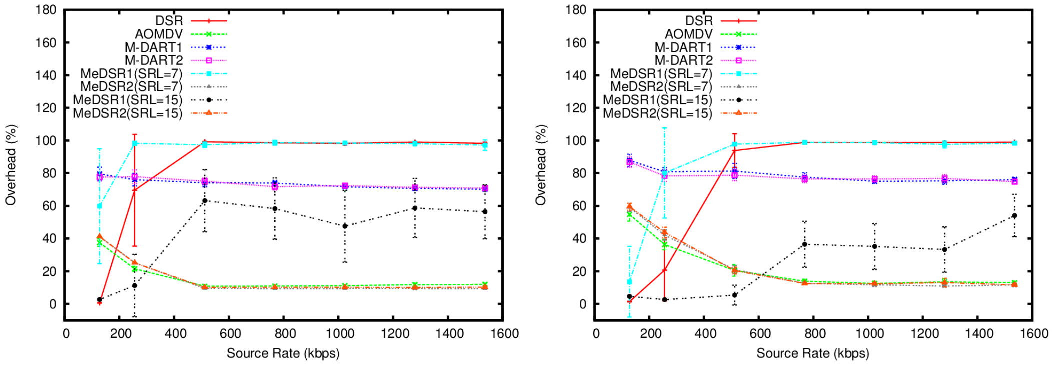
(b) Pareto traffic

FIGURE 13
Control Overhead as function of source rate on scenario 1

8.4 Control Overhead

Figure 13 shows, for scenario 1, that AOMDV and MeDSR2 have less control overhead, as they use probe packets to detect link failures. For small source rates, DSR has smaller control overhead. But as the source rate increases, collisions and contention increase too, and so does the control overhead. MeDSR1 with normal SRL has a higher amount of control overhead, because it uses multiple paths to forward traffic, and has to deal with contention and collisions in both paths.

In Figure 14, we see that MeDSR2 and AOMDV for smaller source rates



(a) CBR traffic (b) Pareto traffic

FIGURE 14
Control Overhead as function of source rate on scenario 2

have more overhead as the number of control packets tends to be higher than the data packets. The number of generated data packets for small source rates is quite low, but as the source rate increases, the number of data packets also increases, reducing considerably the percentage of control overhead. MDART behaves similarly, but has a higher control overhead, as the number of delivered packets is very small.

By analyzing MeDSR1 with normal SRL, we conclude that the number of FRF is very high. FRFs are directly responsible for new route discovery processes which increase the control overhead.

TABLE 8
MeDSR2 Control Overhead Gains for normal SRL (%)

		DSR	AOMDV	M-DART1	M-DART2	MeDSR1
S1	CBR	91.3	23.7	93.9	94.0	98.2
	Pareto	81.4	12.9	93.2	93.4	96.9
S2	CBR	79.9	2.1	78.1	78.1	82.5
	Pareto	66.5	2.8	69.5	69.2	71.1

TABLE 9
MeDSR2 Control Overhead Gains for SRL = 15 (%)

		DSR	AOMDV	MeDSR1		MeDSR2
		SRL=7	SRL=7	SRL=7	SRL=15	SRL=7
S1	CBR	92.1	30.2	98.4	82.6	8.6
	Pareto	81.5	13.3	96.9	41.8	0.4
S2	CBR	79.3	-1.1	81.9	60.8	-3.3
	Pareto	65.7	-5.4	70.4	-1.2	-2.5

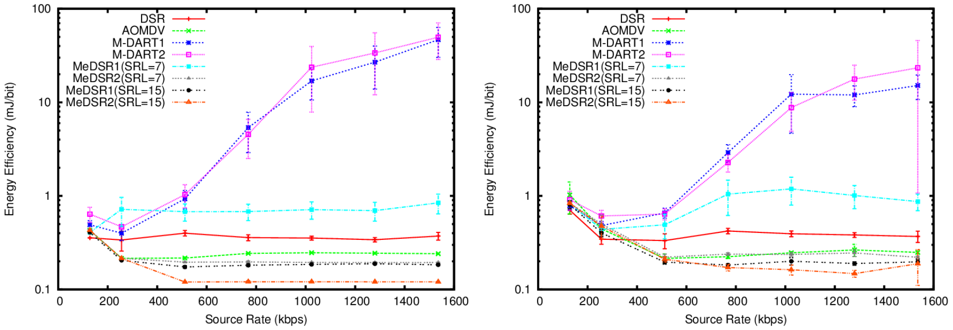
Table 8 shows the average control overhead gain comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. As can be seen, MeDSR2 is on average 62.1% better than protocols that use one path to forward traffic for CBR traffic and 54.4% better for Pareto traffic. It is also better on average 88.2% than protocols that use two paths to forward traffic for CBR traffic and 82.7% for Pareto traffic.

Table 9 shows the average control overhead comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. As can be seen, MeDSR2 with SRL = 15 is on average 54.8% and 34.3% better than other MeDSR implementations for CBR and Pareto traffic respectively.

In comparison with DSR, MeDSR2 is on average 85.7% better for CBR traffic and 73.6% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 14.6% better for CBR traffic and 3.9% better for Pareto traffic.

8.5 Energy Efficiency

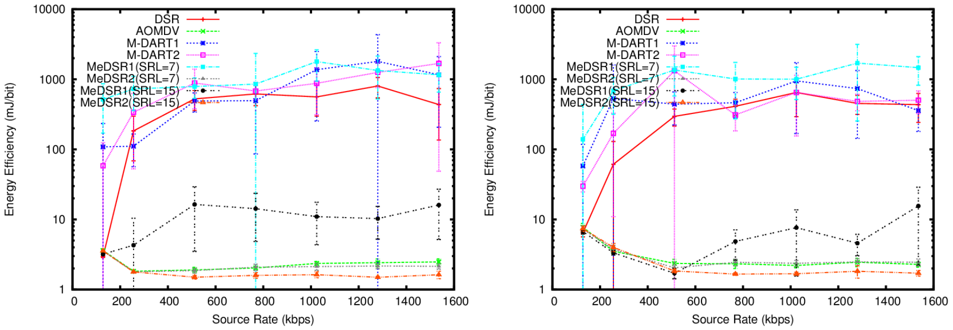
Figures 15 and 16 show that, for a source rate of 128Kbps, DSR presents the best values of energy efficiency, since a single path routing protocol is sufficient for transmission at low data rates, and it does not use additional control packets like AOMDV and MeDSR2. But with the increase of the source rate, nodes suffer more packet loss which causes an increase in energy consumption and a reduction of throughput. MeDSR1 has a similar behaviour to DSR, but the normal SRL version consumes more energy as it experiences more control overhead than other routing protocols. MeDSR2 is the most energy efficient protocol as it has similar values of energy consumption, as other routing protocols, but achieves higher values of throughput. M-DART



(a) CBR traffic

(b) Pareto traffic

FIGURE 15
Energy Efficiency as function of source rate on scenario 1



(a) CBR traffic

(b) Pareto traffic

FIGURE 16
Energy Efficiency as function of source rate on scenario 2

is the routing protocol that performs worst in both scenarios, as it presents the lowest values of throughput.

Table 10 shows the average energy efficiency gain comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 is on average 58% better than protocols that use one path to forward traffic for CBR traffic and 51.4% better for Pareto traffic. It is also on average 89.2% better than protocols that use two paths to forward traffic for CBR traffic and 85.1% for Pareto traffic.

TABLE 10
MeDSR2 Energy Efficiency Gains for normal SRL (%)

		DSR	AOMDV	M-DART1	M-DART2	MeDSR1
S1	CBR	34.4	11.7	97.8	97.9	59.9
	Pareto	15.3	3.3	93.2	93.5	49.1
S2	CBR	99.3	5.2	99.4	99.5	99.5
	Pareto	98.1	0.1	98.5	98.5	99.4

TABLE 11
MeDSR2 Energy Efficiency Gains for SRL = 15 (%)

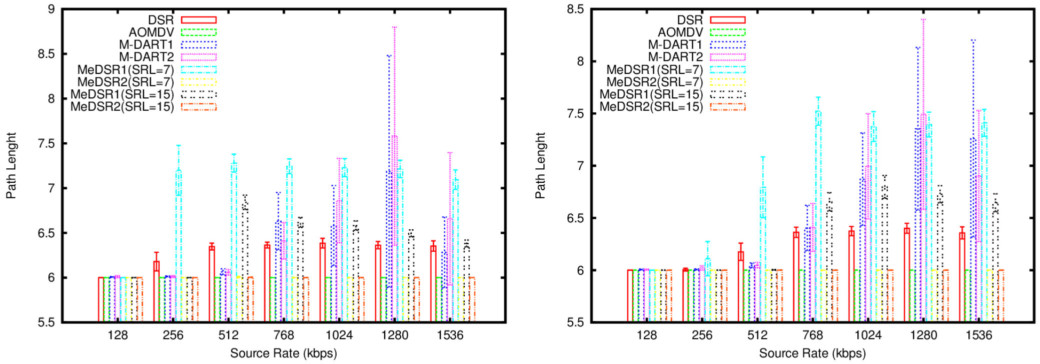
		DSR	AOMDV	MeDSR1		MeDSR2
		SRL=7	SRL=7	SRL=7	SRL=15	SRL=7
S1	CBR	49.4	31.9	69.1	18.1	22.9
	Pareto	26.9	16.6	56.1	1.6	13.7
S2	CBR	99.4	20.3	99.6	15.9	15.9
	Pareto	98.4	11.6	99.4	11.6	11.6

Table 11 shows the energy efficiency comparison between MeDSR2 with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV for both scenarios. As can be seen, MeDSR2 with SRL = 15 is on average 40.3% and 29.4% better than other MeDSR implementations for CBR and Pareto traffic respectively.

In comparison with DSR, MeDSR2 is on average 74.4% better for CBR traffic and 62.7% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 26.1% better for CBR traffic and 14.1% better for Pareto traffic.

8.6 Path Length

Routing protocols that use SRL to detect link failures tend to initiate a route discovery process more often, if intermediate nodes are not able to forward



(a) CBR traffic

(b) Pareto traffic

FIGURE 17
Path Length as function of source rate on scenario 1

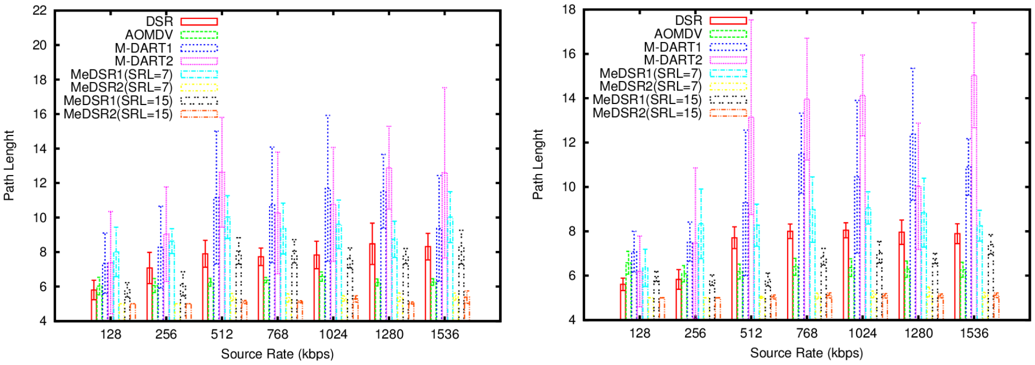
packets. If we consider scenario 1 (see Figure 3), the routes discovered by intermediate nodes do not have equal length, e.g. for node 5, one path has length 3 and another path has length 9 (which includes going back through the source node 0). As intermediate nodes attempt to salvage packets, they tend to use paths with longer path lengths. This happens even more with the increase of the source transmission rate, as more collisions occur, resulting in an increased average path length.

MDART, in case of link failure, attempts to forward packets using other neighbor nodes that have paths with higher costs, resulting in higher average path lengths.

Protocols with high control overhead tend to have also a high path length, as can be seen in Figures 17 and 18. MeDSR2 with SRL=15 achieves the best average path length, for the reasons mentioned in previous sections.

Table 12 shows the average path length gain comparison for normal SRL between MeDSR2 and other routing protocols for both scenarios. MeDSR2 path length is on average 17.4% shorter than that of other routing protocols that use one path to forward traffic for CBR traffic and 18.3% shorter for Pareto traffic. It is also on average 29.1% better than that of the routing protocols that use two paths to forward traffic for CBR traffic and 28.9% for Pareto traffic.

Table 13 shows the average path length comparison between MeDSR2



(a) CBR traffic (b) Pareto traffic

FIGURE 18
Path Length as function of source rate on scenario 2

with SRL = 15 and other MeDSR implementations with SRL = 7 and SRL = 15, DSR and AOMDV in both Scenarios. MeDSR2 with SRL = 15 has an average path length 16.4% shorter than that of other MeDSR implementations for CBR traffic and 13.6% shorter for Pareto traffic.

In comparison with DSR, MeDSR2 is on average 18.5% better for CBR traffic and 17.1% better for Pareto traffic. If compared to AOMDV, MeDSR2 is on average 8.9% better for CBR traffic and 9.9% better for Pareto traffic.

9 CONCLUSION

We have developed a high throughput low coupling multipath routing protocol for wireless multimedia sensor networks and studied its performance through simulation. In our approach, at the routing layer, nodes use ETX in conjunction with the first and second degree correlation factors between paths to find multiple high throughput paths with minimal interference between them. At the MAC layer, we modify SRL and use probe packets to identify routing failures, which reduces packet loss and unnecessary route maintenance operations.

Simulation results show that HTLC-MeDSR presents considerable gains in comparison to other routing protocols in terms of the studied QoS metrics (throughput, delay, packet loss, control overhead, energy efficiency and path

TABLE 12
MeDSR2 Path Length Gains for normal SRL (%)

		DSR	AOMDV	M-DART1	M-DART2	MeDSR1
S1	CBR	4.5	0.0	6.2	7.9	14.4
	Pareto	3.8	0.0	8.6	8.4	13.6
S2	CBR	30.1	15.9	47.4	51.2	42.8
	Pareto	29.9	19.3	48.4	55.3	38.6

TABLE 13
MeDSR2 Path Length Gains for SRL = 15 (%)

		DSR	AOMDV	MeDSR1		MeDSR2
		SRL=7	SRL=7	SRL=7	SRL=15	SRL=7
S1	CBR	4.5	0.0	14.4	6.5	0.0
	Pareto	3.8	0.0	13.6	6.3	0.0
S2	CBR	32.4	17.9	44.2	30.6	2.5
	Pareto	30.3	19.8	39.0	21.8	0.6

length).

By increasing the number of MAC level retransmissions (SRL) from 7 to 15, in conjunction with the mechanisms implemented at the routing layer, HTLC-MeDSR is 40%, 590% and 30% better in terms of throughput, in comparison with DSR, AOMDV and MeDSR1, respectively. This shows the advantage of increasing the SRL, despite the fact that certain caution should be taken, as high SRL values can cause an overall performance degradation.

We also noticed that in terms of delay, HTLC-MeDSR with normal SRL is 180%, 160% and 180% better in comparison with DSR, AOMDV and MeDSR1, respectively. The increase of SRL, despite its throughput improvements, increases the end-to-end delay, as more packets with additional latency are sucessfully delivered.

By increasing SRL from 7 to 15, our routing protocol is 50%, 550% and 10% better in terms of packet loss, and 10%, 300% better and 40% worst in terms of energy efficiency in comparison with DSR, AOMDV and MeDSR1, respectively.

The current solution is targeted to networks with low mobility, which is expected to be the most common situation in Wireless Multimedia Sensor Networks. For 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 the received signal strength indicator (RSSI) to complement ETX, since having a low ETX value does not mean that the nodes are close to each other, and the distance among nodes has direct impact on the achievable throughput. A third topic is to add priority schemes, allowing the protocol to adjust its behavior according to the QoS requirements that multimedia traffic (real or non-real time) may have. Finally, scalability issues also deserve further study.

10 ACKNOWLEDGMENTS

This work was supported in part by national funding from FCT - Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) through the PIDDAC program funds under project PEst-OE/EEI/LA0021/2011 and by the MPSat project (FCT Ref. PTDC/EEA-TEL/099074/2008).

REFERENCES

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

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

[3] S. Ehsan and B. Hamdaoui. A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks. IEEE Communications Surveys & Tutorials, vol.14, no.2, pp.265-278, Second Quarter 2012.

[4] J. Tsai and T. Moors. A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks. ACoRN Early Career Researcher Workshop on Wireless Multihop Networking, pp. 17-18, 2006.

[5] K. Nahm, A. Helmy and C.-C. J. Kuo, TCP over Multihop 802.11 Networks: Issues and Performance Enhancement. in Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'05), Urbana-Champaign, Illinois, May 2005.

[6] Naércio Magaia, Paulo Pereira and Antonio Grilo. A Multipath Extension to the Dynamic Source Routing Protocol for Wireless Multimedia Sensor Networks. Proc. 11ª Conferência sobre Redes de Computadores, Coimbra, Portugal, pp. 49-56, 17-18 November 2011. ISBN: 978-989-96001-6-4.

[7] D. Couto, D. Aguayo, J. Bicket and R. Morris. A high-throughput path metric for multi-hop wireless routing. In Proc. of the 9th Annual Int. Conf. on Mobile Computing and Networking (MobiCom '03). ACM, pp. 134–146, 2003.

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

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

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

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

[12] C. Perkins, E. Belding-Royer and S.Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental). IETF, 2003.

[13] Z. Wang and Jun Zhang. Interference aware multipath routing protocol for wireless sensor networks. IEEE GLOBECOM Workshops (GC Wkshps), pp.1696-1700, Dec. 2010.

[14] S. Kim, J. Lee and I. Yeom. Neighbor-aware Adaptive Retry Limit for IEEE 802.11 based Mobile Ad Hoc Network. Wireless Communication and Networking Conference (WCNC-2008), IEEE, pp. 1402 – 1407, 2008.

[15] G. Bianchi. Performance Analysis of the IEEE 802.11 Distributed Coordination Function. IEEE Journal of Selected Areas in Communications, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. Section 24.3: Dijkstra's algorithm. Introduction to Algorithms (Second ed.). MIT Press and McGraw-Hill, pp. 595–601, 2001. ISBN: 0-262-03293-7.

[17] P. E. Hart, N. J. Nilsson, B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.

[18] IEEE-SA. IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2007 revision), 12 June 2007.

[19] T. Srisooksai, K. Keamarungsi, P. Lamsrichan and K. Araki. Pratical data compression in wireless sensor networks: A survey. Journal of Network and Computer Applications, vol. 35, pp. 37-59, 2012.

[20] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi and H. Hartenstein. 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.

[21] E. Shih, P. Bahl and M. J. Sinclair. Wake on wireless: an event driven energy saving strategy for battery operated devices. In Proceedings of the 8th annual international conference on Mobile computing and networking (MobiCom '02). ACM, New York, NY, USA, 2002.

[22] M. Caleffi and L. Paura. M-DART: Multi-path dynamic address routing. Wireless Communication and Mobile Computing, pp. 392-409, 2011.