EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

DEMON: preemptive route recovery for AODV in multi-hop wireless networks based on performance degradation monitoring

Miguel Catalan-Cid[1,3*], Carles Gomez[2,3], Josep Paradells[1,3] and Jose Luis Ferrer[1,3]

Abstract

Reactive routing protocols like *ad hoc* on-demand distance vector (AODV) have been widely proposed for multi-hop wireless networks (MWNs). However, these types of routing protocols only recover routes after route break detection. Considerable research efforts have been devoted to minimizing the route recovery delay or to anticipating link breaks by using preemptive mechanisms. To the best of our knowledge, state-of-the-art solutions in this space are mainly focused on mobility prediction. However, as we argue in this paper, other phenomena like interference and congestion should also be taken into account, since they can also negatively affect the performance of routes and even cause disconnections. This article presents a novel preemptive solution for AODV called *performance degradation monitoring*, which allows preemptive route recovery based on passive estimation of link data loss rate. Extensive simulations demonstrate that our proposal, which is sensitive to link quality degradation regardless of its nature, improves network performance in a wide range of scenarios.

Keywords: Multi-hop wireless networks; AODV; IEEE 802.11; Preemptive route recovery; Routing metrics

1 Introduction

Routing in multi-hop wireless networks (MWNs) requires two main mechanisms: route discovery and route maintenance. Route discovery is carried out by using a routing algorithm and a routing metric. The former aims at finding available routes, while the latter seeks to select the optimal route based on certain criteria such as hop count, link loss rate, link data rate, and node congestion or interference caused by active flows [1,2]. However, the routing protocol cannot assure that the optimality of the selected route will be maintained in time. Uncontrolled phenomena like fading, external interference, and node mobility could severely degrade link performance and even cause route disconnections [3,4]. In addition, the unpredictable arrival of new flows in the network may cause congestion or inter-flow interference on the flows being routed [5,6]. Thus, route maintenance mechanisms are required.

Route maintenance in on-demand routing protocols is limited to link break detection in order to recover disconnected routes. For instance, most implementations of *ad hoc* on-demand distance vector (AODV) detect link breaks when a defined number of consecutive Hello messages are lost [7]. By using this recovery procedure, a flow may undergo a large interval of performance degradation until the detection of the link break and the rediscovery of the new route [8].

The aforementioned problem can be significantly mitigated by means of preemptive route maintenance [9]. This technique allows nodes to monitor the quality of the links that participate in active routes, decide when the quality is no longer acceptable, and, in that case, trigger route recovery immediately, before an actual route break occurs. To the best of our knowledge, state-of-the-art solutions in this area are limited to using link break predictors and anticipating route recovery with the objective of reducing route disconnections due to node mobility [9-17]. Nevertheless, as we argue in this paper, a comprehensive solution is required which considers other causes of route performance degradation like interference or congestion.

* Correspondence: miguel.catalan@entel.upc.edu
[1]Wireless Networks Group, Telematics Engineering Department, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB), Jordi Girona 1-3, Barcelona 08034, Spain
[3]i2CAT Foundation, Gran Capita 2–4 (Nexus building), Barcelona 08034, Spain
Full list of author information is available at the end of the article

In this paper, we present the following contributions:

- A comprehensive study of the route maintenance mechanisms used by on-demand routing protocols for MWNs and the main preemptive route recovery proposals of the literature.
- The design of the DEMON solution, which performs preemptive route recovery by using information about the data loss rate of the links. We have designed and evaluated DEMON as an extension of AODV, although it could also be applied to other reactive protocols.
- An evaluation of DEMON by means of extensive simulations in a wide range of scenarios. Results show that DEMON significantly improves network performance in all the scenarios considered.

The remainder of the paper is organized as follows: Section 2 surveys existing route maintenance mechanisms of reactive routing protocols for MWNs. Section 3 describes our solution and its design principles, Section 4 presents the evaluation of DEMON by means of extensive simulations, focusing on its performance in different IEEE 802.11 MWN scenarios. Finally, Section 5 concludes the paper with the main remarks from our work.

2 Route maintenance in reactive routing protocols for MWNs

This section reviews the route maintenance mechanisms used by the AODV routing protocol and the main preemptive route recovery extensions designed for on-demand routing protocols. As previously introduced, some of the analyzed preemptive solutions can be also applied to other protocols, such as the Dynamic Source Routing protocol (DSR) [9,13,14,18].

Section 2.1 introduces AODV and discusses its route maintenance mechanisms. Section 2.2 reviews the state of the art of preemptive extensions of on-demand routing protocols for MWNs.

2.1 Route discovery and maintenance in AODV

In AODV, route discovery is carried out on demand by broadcasting route request (RREQ) messages. The routing metric is computed each time a node receives a RREQ message, and the corresponding cost is added to the total cost of the path. If upon reception of a RREQ the node has a valid route entry to the requested destination or the node is the destination itself, the node sends a route reply (RREP) message back to the source node. Otherwise, the node broadcasts the RREQ again. In any case, a RREQ receiver creates a backward path routing table entry that indicates the sender of the RREQ as the next hop in the path towards the source node. Every node that receives a RREP creates and maintains a forward path routing table entry with next hop information that expires after a specified time if the path becomes inactive.

When a link breaks along an active path, the node that detects the link break transmits a route error (RERR) message which lists the set of destinations that have become unreachable. Upon reception of the RERR message, the sources affected by the link break start a new route discovery, provided that they still have data to transmit to the same destination. If alternative paths exist, the source and destination of a broken route will remain disconnected from the moment of transmission of the RERR until the reception of a new RREP by the source. The duration of this disconnection interval (or route change latency) may be extremely significant, especially in highly loaded, mobile, or large networks [8,19]. This problem can be alleviated by means of an option called local repair [7]: the intermediate node which noticed the link break initiates route recovery by broadcasting RREQ messages with a time-to-live (TTL) set to the last known distance to the destination, plus an increment value. In this way, the route is recovered faster and the mechanism prevents the entire network from being flooded again [19].

For connectivity maintenance purposes, each node can periodically broadcast Hello messages within a one-hop radius. By default, the interval between Hello messages is 1 s and the loss of three consecutive Hello messages is understood as a link break [7]. Increasing the frequency of Hello messages or decreasing the number of allowed Hello losses is a simple way to improve the network sensitivity to mobility [8]. However, the detection of link breaks based on Hello messages is not optimal [19-21]. Hello packets are short, and in 802.11-based MWNs, they are broadcasted at the lowest and most robust data rate. There are therefore cases in which Hello packets can be correctly received, while data packets cannot, leading to large periods of unsuccessful data packet transmission.

There are other strategies for link failure detection, such as link layer notification and passive acknowledgement, based on the absence of link layer ACK messages and the overhearing of forwarded packets, respectively [7]. However, these techniques need link layer feedback, which makes the routing protocol implementation dependent on the link layer mechanisms. Therefore, almost all real AODV implementations are based on the exchange of Hello messages for connectivity maintenance, which allows a node to carry out one-hop neighborhood discovery and maintenance regardless of the specific link layer used.

2.2 Preemptive extensions of on-demand routing protocols for MWNs

One approach that mitigates the connectivity maintenance limitations of AODV mentioned above is the maintenance of routes using preemptive mechanisms. Preemptive route

recovery comprehends the following two components: link quality monitoring and route recovery [9]. Nodes monitor the quality of the links that participate in active routes and decide when the quality is no longer acceptable. In such cases, the node which detects the performance degradation warns the source of the route (which then can start the discovery of a new route) or starts a local repair process in order to find an alternative route.

In this subsection, we first review link quality monitoring, and second, we focus on route recovery techniques used in preemptive extensions of on-demand routing protocols.

2.2.1 Link quality monitoring

Most state-of-the-art link quality monitoring solutions for preemptive route recovery define link quality as a function of the received signal strength, in order to detect node mobility and anticipate link breaks. Since this mechanism was first introduced [9], it has also been adopted by subsequent proposals [10,13-15,17]. When a node receives a packet, the node compares the received power with a defined power threshold. If the received power is smaller than the threshold, the node assumes that the sender has left the region where communication was safe (i.e., now, it is located in the so-called preemptive region). Thus, it may soon be unreachable due to mobility (i.e., be located in the out-of-range region). In that case, therefore, the receiver triggers route recovery preventing an incoming link break and route disconnection. Figure 1 illustrates the different communication regions.

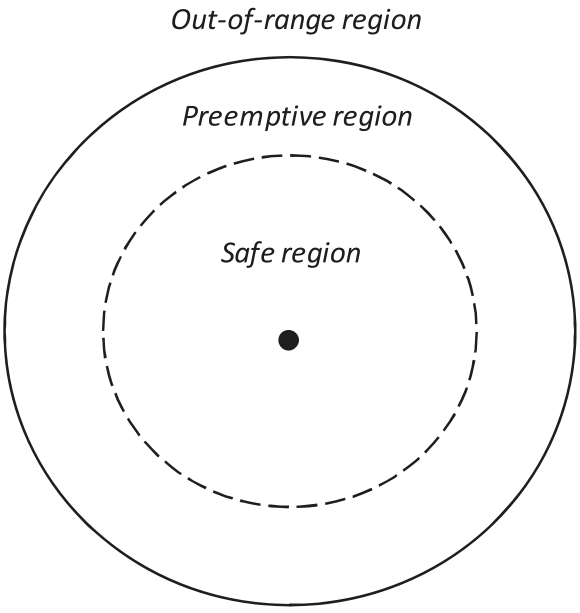

Figure 1 Illustration of the regions involved in the preemptive recovery mechanism based on signal strength. The black dot at the center of the circles represents a receiver that determines whether the received power is below the threshold.

A significant drawback of preemptive solutions based on link quality monitoring is that they need successful packet reception in order to monitor the signal strength of the packets sent by a particular neighbor. Thus, these solutions are not able to detect (and react to) congestion and interference, since such phenomena cause packet drops and corrupt receptions due to collisions (e.g., as in the case of a hidden terminal) and do not affect the received signal power of successfully received packets. Furthermore, these solutions must have access to nodes' physical layer parameters, which limit their implementation to specific network interface card (NIC) drivers. In fact, a wide majority of routing protocol implementations for MWNs use connectivity maintenance mechanisms which operate at the network layer, in order to avoid dependence on specific NIC drivers [22].

Instead of using received signal strength, other preemptive protocols use GPS to obtain the position and speed of the nodes and compute the link expiration time (LET), i.e., the predicted amount of time during which the link will remain active [11]. When the LET of a link falls below a defined threshold, a route recovery procedure is started. This approach requires a GPS receiver on each node and clock synchronization, thus limiting its application in real deployments. In addition, this solution is only aware of link degradation due to mobility.

Another type of preemptive solution is based on the rate adaptation mechanisms used by some drivers of IEEE 802.11 interfaces [12]. These adaptation algorithms increase or decrease the data rate of the nodes (i.e., decrease or increase the robustness of the packet transmission, respectively) according to the measured packet loss. A transmission rate decrease has been considered as an indicator of mobility [12]. According to the neighbors' data rates and to its own, a node can detect when the distance between itself and the neighbors increases and initiate a route recovery procedure. However, the behavior of the rate adaptation algorithm under congestion and interference could be counterproductive due to uncontrolled rate decrease, which might lead to a greater amount of congestion and performance degradation [12,23].

Link break prediction can also be based on packet loss monitoring carried out by potential relay nodes. In a proposed routing protocol [16], relay nodes, which are neighbors of the sender and receiver of the monitored link, overhear packet transmissions. When a relay node notices that a packet has reached its maximum number of retransmissions, it assumes that, due to mobility, the transmitter and the receiver are now out of their coverage ranges. The relay node then warns the transmitter in order to recover the route and proposes itself as an

alternative next hop to the receiver. However, in real deployments, it is difficult to assure the presence of a relay node for each pair of nodes. In addition, although this mechanism can detect packet losses due to any phenomena, under high-load conditions (where collisions and retransmissions are very frequent), it may lead to network instability due to a high amount of route recoveries. Packet overhearing under these conditions could also burden the resources of the nodes. The mechanism has only been evaluated under mobility and very low load conditions [16].

Note that the preemptive solutions introduced in this subsection depend on a threshold in order to determine when the quality of a link is no longer acceptable and therefore when route recovery should be initiated. The definition of the threshold is critical, since a threshold that is too high could lead to a late reaction to link quality degradation, whereas if the threshold is too low, it may trigger unnecessary recoveries (e.g., due to temporary wireless channel quality fluctuation). Thus, some protocols use additional mechanisms to assure that the link is really degraded. For example, the nodes that form the degraded link can use a ping-pong procedure [9]: if more than a certain number of pings (i.e., probe packets requiring a reply) are lost, degradation is confirmed and the route recovery is initiated. Another technique is based on comparing the power of the last two received packets in order to estimate the relative speed between the two nodes that form the link [14]. This estimation is used to update the value of the power threshold. In this way, as relative speed increases, the preemptive region becomes greater.

Finally, note that all the state-of-art link quality monitoring mechanisms considered, apart from [11] which uses GPS, are based on the reception of data packets. Therefore, if there are large intervals without data packets (for instance, due to Transmission Control Protocol (TCP) congestion control), link quality estimation could become poor or even unfeasible. In such a case, route recovery should be assumed by the default mechanisms of the routing protocol (i.e., route maintenance based on Hello losses in basic AODV).

2.2.2 Route recovery

Depending on which node aims at repairing a broken route, route recovery techniques can be classified in the following two categories: source-initiated route recovery and local repair-based route recovery.

In source-initiated route recovery, the node which notices the degradation notifies this fact to the source by using a special control packet. Then, the source initiates a new route discovery for the same flow. In order to avoid the selection of degraded links during route recovery, RREQ messages should carry additional information. For instance, the RREQ messages can transport the minimum allowed link quality threshold [9,17]. Using this information, the nodes will discard the RREQ received through a link of a quality lower than the threshold.

The main drawbacks of source-initiated route recovery are the related overhead and delay. Nevertheless, such drawbacks can be minimized if preemptive route recovery protocol extensions are applied to DSR [13,14]. In DSR, the nodes can store in a route cache different route alternatives for the same destination. Then, if preemptive recovery (or a link break) occurs, the nodes are able to use alternative routes stored in their caches without executing the whole route discovery procedure.

On the other hand, local repair-based route recovery mechanisms allow intermediate nodes to start the route recovery procedure. The node which notices link degradation may broadcast RREQ messages as per the classical local repair procedure [11,12]. Other proposals use a relay node, which is a neighbor of both nodes forming the degraded link, to maintain the connectivity between the two link ends [10,15,16]. However, these solutions may fail to find alternative routes in sparse networks. In addition, if performance degradation is due to congestion or interference, route recovery based on using nearby nodes could be inefficient, since it may reproduce the problem that led to performance degradation. In contrast, source-initiated route recovery does not suffer the same problem, since it allows to completely change the route for a flow once significant performance degradation has been found.

2.3 Requirements for a preemptive route recovery for AODV

From the analysis of the state of the art, we conclude that a preemptive solution for AODV should satisfy the following requirements:

- The solution should use a comprehensive link quality monitoring mechanism which allows the protocol to take into account not only mobility but further reasons for performance degradation like congestion or interference.
- The link quality monitoring mechanism should be based on information available at the network layer, so that the solution does not depend on the link layer implementation and is not limited to particular network interface card drivers or hardware.
- The route recovery mechanism should be source-initiated in order to allow appropriate flow distribution in the network.

In the following section, we present the DEMON extension, which has been designed in order to satisfy the aforementioned requirements.

3 DEMON

This section describes our preemptive solution, called DEMON, which stands for *performance degradation monitoring*. We have designed DEMON as an extension of AODV, although it could be applied to other reactive protocols. DEMON comprises two main mechanisms: link quality estimation and route recovery. In this section, we first introduce the motivation for DEMON. We then present and discuss the link quality estimation mechanism. Subsequently, we introduce the link quality thresholds used in DEMON, which allow us to determine when the quality of a link is no longer acceptable. We then describe the procedures for route recovery. Finally, we illustrate its performance in the scenario of the example provided to show the motivation of its design.

3.1 Motivation

In order to illustrate the purpose of our solution, we introduce an example of performance degradation which is not caused by mobility. For this example, a 64-node grid using the basic version of AODV was simulated[a] (see Figure 2). At second 200, a flow *f1* with a data rate of 1 Mbps was initiated and routed through *initial route*. Then, at second 250, a highly loaded flow of 5 Mbps, called *f2*, was initiated. Note that in such a scenario, regardless of the routing metric in use, *f2* would degrade the performance of *f1*, since the source of *f2* was a hidden node for several nodes traversed by *f1*. Thus, in order to avoid the interference of *f2*, *f1* should be rerouted after *f2* activation. However, as previously mentioned, basic AODV only performs route recovery after the detection of a link break.

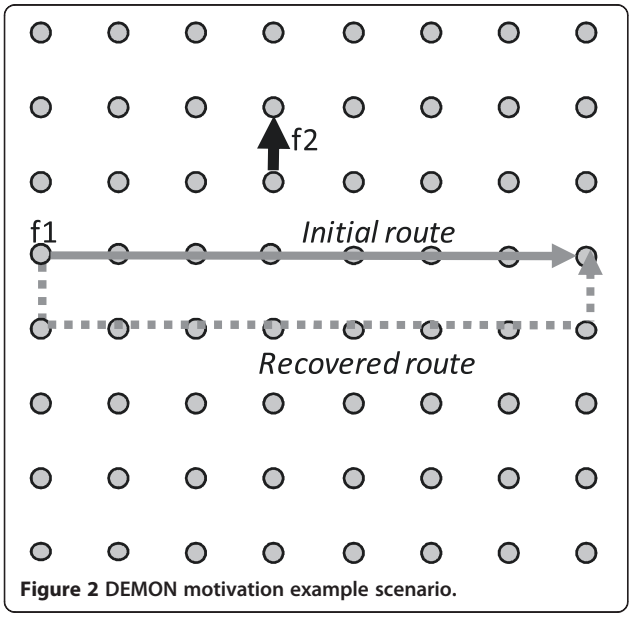

Figure 2 DEMON motivation example scenario.

After repeating the experiment 20 times with different seeds, we obtained two different types of results. In the first one, *f1* traversed the initial route during all the simulation times. As shown in Figure 3a, due to the high contention and interference of *f2*, the throughput of *f1* became reduced approximately in one half. In the second one, the initial route became disconnected after a long period of performance degradation due to the interference of *f2* and AODV finally re-routed *f1* through *recovered route* (see Figure 3b). The latter route did not suffer the influence of the interfering flow. However, route recovery was only triggered after three consecutive Hello message losses, which occurred after roughly 1 min of performance degradation.

Note that in the presented scenario, where the reasons for performance degradation are contention and interference, preemptive solutions based on received signal strength [9,10,13-15,17], node position [11], or link data rate [12] would behave analogously to basic AODV, since the interfering flow does not affect the signal strength of successfully received packets, while the position and the link data rate of the nodes are fixed. Therefore, we designed DEMON as a preemptive route recovery extension of AODV which is aware of flow performance degradation due to any reason and aims at allowing a fast route recovery. In the following subsections, we will describe its components in detail.

3.2 Link quality estimation mechanism

In order to detect performance degradation caused by a range of reasons wider than mobility alone, we choose link loss rate as the criterion for determining link quality. A high loss rate may be a sign not only of mobility, but also of interference or congestion. In addition, due to retransmissions, lossy links also impact negatively on the performance of nearby nodes. For these reasons, we define a link quality estimation mechanism based on the expected transmission count (ETX) metric [24], which we describe below.

The ETX metric, which estimates the expected number of transmission attempts for a packet through a link, was one of the first means for increasing performance in MWNs to be proposed as an alternative to the hop count metric. Since that time, ETX has been widely adopted. In fact, most of subsequent routing metrics that consider the link loss rate have been designed on the basis of ETX [1,2,25,26].

A node that implements ETX computes the packet delivery probability in each direction of the link between nodes i and j to obtain the link cost, ETX_{ij}, as shown in Equation (1), where d_i and d_j, denote the packet delivery probability in directions ij and ji, respectively. Both link directions are considered, since successful frame transmission requires the reception

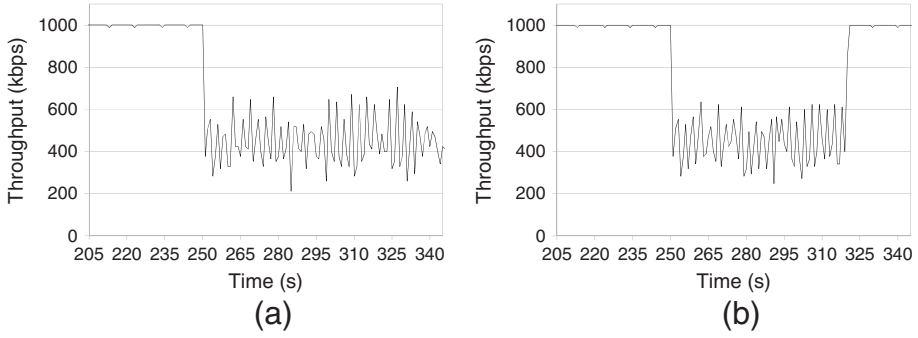

Figure 3 DEMON motivation example results. (a) Throughput of *f1* using basic AODV in a simulation whereby route disconnection does not occur. **(b)** Throughput of *f1* using basic AODV in a simulation whereby initial route becomes disconnected and a new route is discovered.

of an ACK frame in many link layer protocols, as in 802.11 CSMA/CA.

$$\mathrm{ETX}_{ij} = \frac{1}{d_i} \times \frac{1}{d_j}. \quad (1)$$

ETX computation is usually implemented by taking into account the number of Hello messages which have been lost in a given period of time [24]. However, as introduced in Section 2.1, the estimation of the link error rate based on Hello messages is not optimal. Recent studies also show that ETX estimation based on probing messages is not reliable if the links are interfered by active flows [27].

In order to obtain a better estimation of the link performance than the one provided by the ETX metric, we redefine ETX by using performance measurements of actual data traffic instead of Hello messages. Our metric is called passive ETX (pETX). The pETX metric of a link *ij*, denoted pETX_{ij}, represents the expected number of data packet transmission attempts required for successful data packet delivery in the link *ij* and is computed as follows:

$$\mathrm{pETX}_{ij} = \frac{1}{d_i} \times \frac{1}{d_j} = \frac{\mathrm{NS}_{ij}}{\mathrm{NR}_{ij}} \times \left(\frac{\mathrm{NR}_{ij} + \mathrm{ND}_{ij}}{\mathrm{NR}_{ij}} \right), \quad (2)$$

where

- NS_{ij} is the number of packets that the network layer of *i* has passed to its link layer in order to be transmitted to *j* during a certain interval. Note that since NS_{ij} is computed at the network layer, the packets that originated from the upper layer retransmissions contribute to NS_{ij} as new packets.
- NR_{ij} is the number of non-duplicated packets that the network layer of *j* has received from node *i* during a certain interval. Note that these packets may be forwarded to the next hop (if *j* is an intermediate node) or processed by the upper layers (if *j* is the destination node).
- ND_{ij} is the number of duplicated packets received by the network layer of node *j* from node *i* during a certain interval. Duplicated packets can be detected by comparing the last two received packets from a neighbor. Reception of two consecutive identical packets implies that the link layer of the transmitter has performed a retransmission due to the loss of an ACK.

Remarkably, nodes compute pETX by using information available at the network layer. While parameters NR_{ij} and ND_{ij} can be monitored by node *j*, this node needs to know the number of packets sent by node *i* during a certain period (which we call *cycle*) in order to calculate NS_{ij}. For this purpose, node *i* includes the number of packets sent since the start of the current cycle in its Hello messages, making it possible for node *j* to compute pETX_{ij}. The duration of the cycle is equal to the number of allowed Hello losses of the routing protocol. The frequency of Hello messages is a configurable parameter, which affects the time required to detect performance degradations (see Section 4.2).

The parameters NS_{ij}, NR_{ij}, and ND_{ij} are recomputed by node *j* upon reception of a Hello message sent by node *i*. In order to provide robustness to the pETX computation even when Hello messages are lost, sequence numbers are included in the Hello messages. Figure 4 shows an example of pETX_{ij} computation, whereby sequence numbers are reset after a cycle of three Hellos. Intervals without transmitted packets (during which NS_{ij} is equal to zero) are not included in the pETX computation. Finally, if there is a complete cycle without transmitted packets, the pETX computation of the link is cancelled until data packets are received and can be monitored again.

In order to filter out the influence of temporary fluctuations of the radio channel and avoid spurious detection of link quality degradation (which would trigger spurious route recoveries), we apply an exponentially

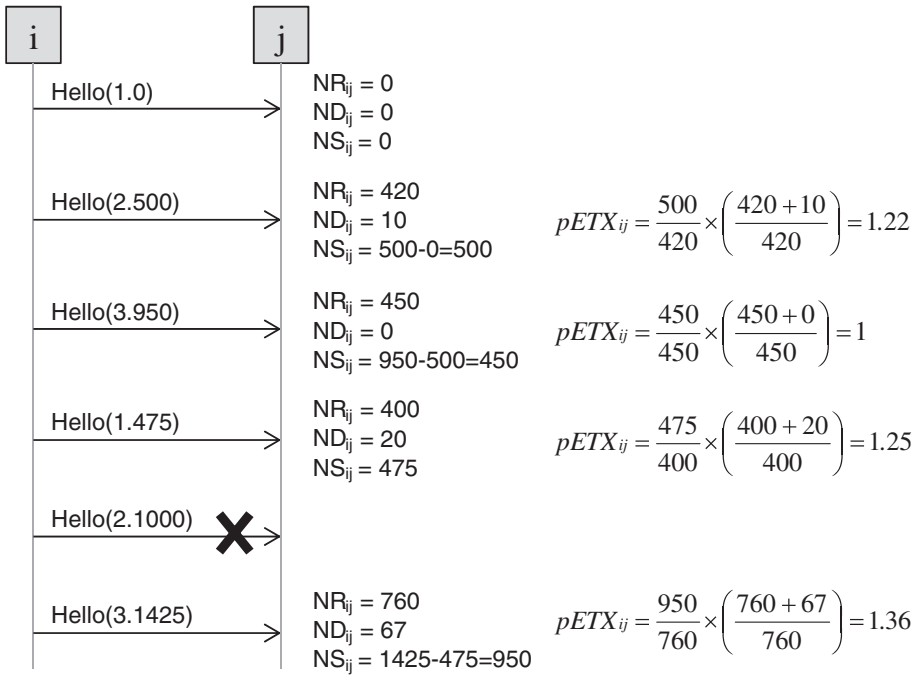

Figure 4 Example of $pETX_{ij}$ computation performed by node *j* with a cycle of three Hellos. The notation of the Hello messages is *Hello (sequence number and number of packets transmitted in the current cycle)*. Note that the fifth Hello is lost. However, by means of the sequence number, node *j* knows that the sixth Hello belongs to the same cycle as the one previously received (otherwise, a link break would have occurred), and thus node *j* can correctly compute the pETX metric.

weighted moving average (EWMA) to pETX, as shown in Equation (3). This filter has been called the time averaged filter or window mean EWMA (WMEWMA) and has been recommended for link quality estimation since it offers a good trade-off between reaction time and stability [28]. We define the smoothed pETX of a link *ij* at instant *n*, $\mathrm{spETX}_{ij}(n)$, as follows:

$$\mathrm{spETX}_{ij}(n) = \alpha \times \mathrm{spETX}_{ij}(n-1) - (1-\alpha) \times \mathrm{pETX}_{ij}(n), \quad (3)$$

where α is a coefficient between 0 and 1 which controls the contribution of a new $\mathrm{pETX}_{ij}(n)$ sample to the smoothed $\mathrm{spETX}_{ij}(n)$. The latter is updated when a new $\mathrm{pETX}_{ij}(n)$ sample is available.

Finally, node *j* computes the link success rate (LSR) of link *ij*, LSR_{ij}, by using Equation (4). The LSR of a link is then compared with a link quality threshold to determine whether the link performance is sufficient, as explained in the next subsection:

$$\mathrm{LSR}_{ij}(n) = 100 \times \frac{1}{\mathrm{spETX}_{ij}(n)}. \quad (4)$$

3.3 Link quality threshold

As explained in the previous section, in DEMON, every node computes the LSR of each link it belongs to. In order to assure that only links of sufficient quality are used, we have designed the following mechanism: If the LSR of a link *ij* falls below a given threshold, node *j* starts route recovery procedure (which is described in the next subsection).

The threshold is a parameter that represents the minimum allowed LSR of a link, expressed by a percentage. In order to use a particular threshold in the network, it is necessary to consider the trade-off between sensitivity and stability. If it is not possible to find routes with a high success rate due to congestion, a high threshold may lead to continuous recovery of routes. On the other hand, in the presence of mobility, a low threshold will give rise to a slow reaction to performance degradation. In Section 4, we evaluate the performance of different fixed values for the link quality threshold.

In addition to the option of using a fixed threshold for all links and flows in the whole network, we have designed a dynamic solution in which a particular threshold $\delta_{k,ij}$ is used for each flow *k* and traversed link *ij*. After route discovery, the threshold for each flow and link is set to the LSR of the link minus a safe margin, β, as follows:

$$\delta_{k,ij} = \mathrm{LSR}_{ij}(t_k) - \beta, \quad (5)$$

where t_k is the instant when the new route for the flow *k*

traversing the link *ij* becomes active, and the margin β is used in order to avoid spurious route recoveries due to small variations of the LSR. β is a percentage between 0% (immediate route recovery) and the LSR of the link (route recovery disabled). Once the threshold is set as per Equation (5), if the LSR of the link falls below the threshold, a route recovery is triggered. When a route is recovered, a new threshold is set for the affected flow for each traversed link.

3.4 Route recovery mechanism

In DEMON, once the LSR of a link *ij* falls below the corresponding threshold, node *j* starts route recovery. The route recovery mechanism comprises two components: (1) the selection of a flow that will be recovered and (2) the transmission of a warning message to the source of the flow, so that the source triggers the AODV route discovery.

We next describe the rationale of the algorithm used in our solution for selecting the flow to be re-routed. Since link quality degradation may not only be due to mobility, it may be counterproductive to re-route all the flows traversing a degraded link. For instance, in case of congestion, re-routing all the affected flows at the same time could reproduce the same problem in another zone of the network and cause instability. In order to avoid this problem, if node *j* notices link degradation, it selects one single flow k^* to be re-routed, by using Equation (6):

$$k^* = \arg\max_{k \in K} \{\min(d_k(\text{source}_k, j), d_k(j, \text{destination}_k))\}, \tag{6}$$

where K denotes the set of flows that traverse the degraded link, and $d_k(a,b)$ denotes the number of hops between nodes a and b through the route of flow k. The algorithm expressed in Equation (6) selects the flow k^* whose minimum distance in hops from node j to one of the endpoints of its route is the greatest one among the flows in K. As shown in Figure 5, the algorithm attempts to re-route the most appropriate flow at a greater distance from the degraded link.

Once the flow is selected, the node warns the source of that flow about the degradation, and then the source carries out a new route discovery to determine a new route for the flow. This way, a new path is created between the source and the destination (which is the same for both the forward and the backward directions). The warning message is unicast and has a size of 12 bytes.

The route recovery mechanism encounters a performance limitation due to the fact that AODV is *destination-based*. In AODV, all flows with the same destination that traverse the same common intermediate node are forced to follow the same route between the intermediate node and the destination node (Figure 6a). Therefore, using AODV, the recovery of one route could also cause an implicit recovery of other routes to the same destination. This would limit the benefits of the route selection algorithm, since as mentioned above it might reproduce the problems that led to quality degradation in another zone of the network.

In a prior work, we dealt with this problem of AODV by designing a *flow-based* extension of AODV (FB-AODV) which finds routes individually for each flow [26]. Using FB-AODV, routes are determined on the basis of the destination address, the source address, and also the type of service (ToS) for the flow. As shown in Figure 6b, this enables a better distribution of flows in the network, since flows with a common destination can follow different paths, thereby improving the overall performance of the network [26]. We have therefore chosen FB-AODV as the basis of our preemptive route recovery proposal, since it allows us to exploit to the full how DEMON redistributes flows in the network after performance degradation. Finally, in order to illustrate the complete DEMON process,

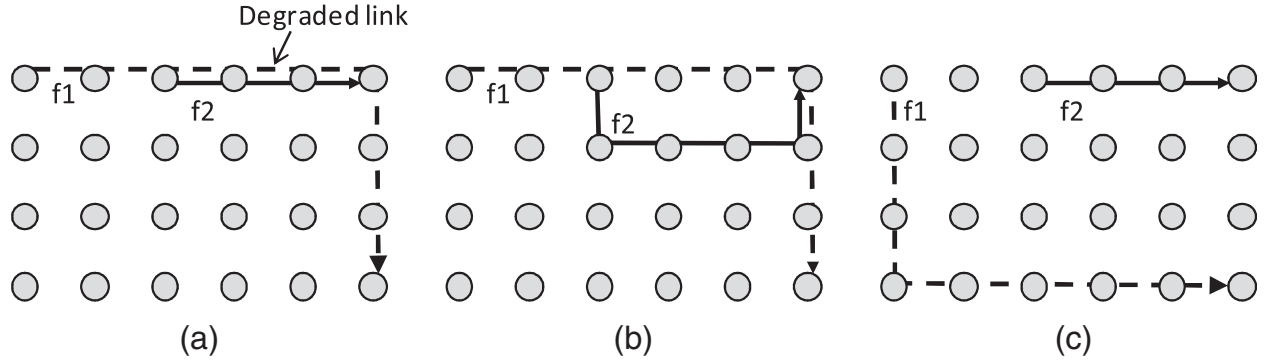

Figure 5 Flow selection and route recovery in case of degradation. (a) Initially, flows *f1* and *f2* are being routed through a link which becomes degraded. **(b)** If the degradation is due to congestion and *f2* is selected for being re-routed, the new route might reproduce the same problem. **(c)** As per the algorithm shown in Equation (6), *f1* is selected for being re-routed, which leads to a better flow distribution in the network, whereby flows affect each other to a significantly smaller extent.

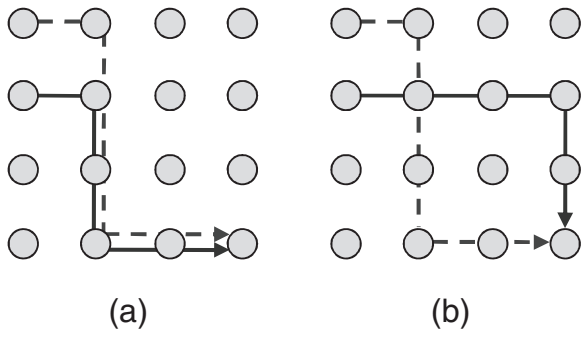

Figure 6 Example of destination-based routing vs. flow-based routing. There are two flows with the same destination. **(a)** In destination-based routing, both flows share the same path segment from the first common intermediate hop to the destination; **(b)** flow-based routing allows a better flow distribution in the network.

Figure 7 shows a flow chart that includes all DEMON components and their sequencing.

3.5 Performance of DEMON in the motivation example scenario

Revisiting the example scenario described in Subsection 3.1 and using DEMON instead of basic AODV, we observed a significant improvement of the performance of the interfered flow, as shown in Figure 8.

Since the interference of flow *f2* caused the loss of about the 50% of the packets, the LSR of the degraded link fell very fast under the threshold (in this case, we defined a fixed threshold of 90%). Once a node of the path monitored that degradation had occurred, it warned the source of *f1* which rerouted the flow through recovered route. As shown in the figure, in this scenario, the whole DEMON route recovery procedure took around 5 s from the beginning of the interference until the flow became rerouted, thus reducing dramatically the duration of the performance degradation period (see Figure 3 for comparison). In the next section, we will extensively evaluate DEMON under a wide variety of conditions.

4 Evaluation

In this section, we analyze by simulation the performance of DEMON. Section 4.1 introduces the simulation platform and the main parameters used in the evaluation. Then, Section 4.2 studies the performance of DEMON when mobility is the main cause of link quality degradations. Section 4.3 analyzes the performance improvement of DEMON under interference in a wide range of scenarios. Finally, in Section 4.4, we study the performance of DEMON when data flows use TCP as the transport protocol.

4.1 Simulation platform and scenario

Our evaluation is carried out by simulation, using OMNeT++ v3.4b2 (Andras Varga, Technical University of Budapest), a discrete event simulator [29]. We chose OMNeT++ because its wireless physical model is based on the *Additive SNIR Model*, which simulates carrier sensing and interference more accurately than other models, such as the *Capture Threshold Model* implemented by default in NS-2 [30,31]. OMNeT++ uses random number generators based on the Mersenne Twister with a period of 2^{19937}-1, which ensures statistical validity of the results

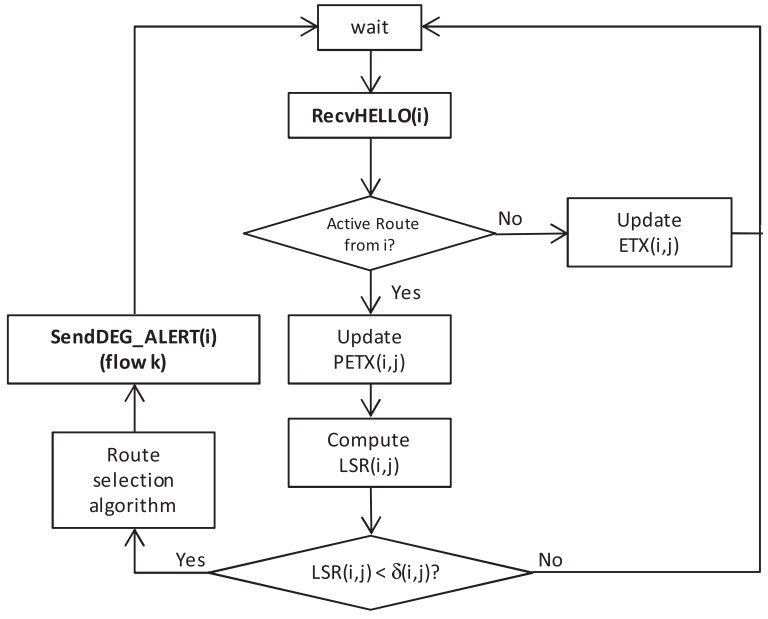

Figure 7 DEMON flow chart of node *j* and link *ij*.

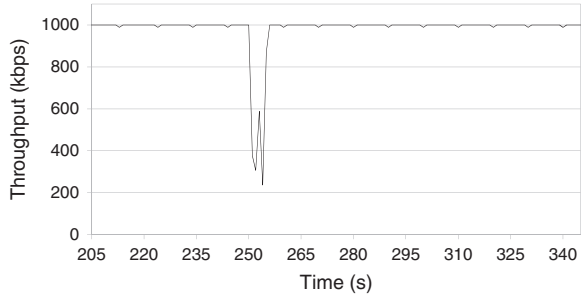

Figure 8 DEMON motivation example results. Throughput of *f1* using DEMON.

[32]. Table 1 summarizes the main parameters of the simulations. The PHY/MAC layer parameters of the simulated nodes are based on the specification of the IEEE 802.11a/g standard [33].

For the evaluation, we implement DEMON as an extension of the AODV simulation code used in a previous work [26]. Hereafter, we will refer to the AODV routing protocol without the DEMON extension as *default AODV*, while the AODV routing protocol with the DEMON extension will be simply denoted as *DEMON*. With regard to the DEMON link quality threshold, we analyze the performance of three fixed thresholds (whose values are set to 90%, 80%, and 70%) and of the dynamic (DYN) one.

Table 1 Simulation fixed parameters

Parameter	Value or configuration
Propagation model	Two-ray propagation model
Fading	Ricean fading with factor 5
Transmission power	30 mW
Transmission rate of broadcast, preambles, and ACKs	6 Mbps
Minimum sensitivity at 6 Mbps (carrier sensing range using preamble detection)	−82 dBm in reception (197 m)
Minimum sensitivity at 12 Mbps (max. communication range at 12 Mbps)	−79 dBm in reception (167 m)
Noise level (max. interference range)	−95 dBm in reception (418 m)
RTS/CTS mode	Off
Type of flows	UDP, constant-bit-rate, unidirectional, TCP Reno, bulk transfer
Random number generators (RNG)	Four independent RNGs: application, routing, MAC, and PHY layer
Simulation time	460 s
Flows initialization	First flow starts at second 100. Then, a new flow starts every 5 s.
Flows ending	At second 450
Number of simulations per obtained average result	50
Number of nodes	64
Simulation area	980 m × 980 m
α (EWMA)	0.5
β (LSR)	10%

In the evaluation, two different state-of-the-art routing metrics are used for route discovery. These routing metrics are the expected transmission time metric (ETT) [25] and the weighted contention and interference routing metric (WCIM) [26]. ETT improves the link awareness of the ETX metric by taking into account the data rate of the links, thus favoring the selection of fast links with low error rates. On the other hand, WCIM is both link- and load-aware, and improves ETT by estimating the available bandwidth of the links. This way, we can analyze the behavior of DEMON when two different route discovery strategies are used. For a detailed performance comparison between ETT and WCIM routing metrics, the reader may refer to the literature [26].

For the evaluation, we chose the setting of α and β shown in Table 1, which offers a reasonable trade-off between avoiding spurious route recovery and fast reaction to degradation. More specifically, low α values (i.e., higher impact of the last link pETX sample on the link pETX estimation) could trigger unnecessary route changes due to temporary link degradation, whereas high α values could excessively delay route recovery. On the other hand, setting the β parameter to low values (i.e., the adaptive threshold approaches the LSR), will lead to immediate route recovery, whereas high β values will lead rapidly to low threshold values, which can delay the reaction to performance degradation. Nevertheless, both α and β parameters should be fine tuned for each particular scenario.

In the next subsections, we use goodput (i.e., number of bits correctly received at the destination per time unit) as the main performance indicator. Each result depicted in the figures presented hereinafter illustrates the average value obtained from 50 simulations. We have computed the 95% confidence interval of each result. These intervals are below 3% and 5% of the results obtained in stationary and in mobile scenarios, respectively. Thus, for the sake of illustration clarity, we do not include the confidence intervals of the results of this section in the corresponding figures.

4.2 Analysis of DEMON performance with mobility

The scenario analyzed in this subsection consists of six active User Datagram Protocol (UDP) and constant bit rate (CBR) flows, each with randomly selected source and destination. Starting from a random location, the nodes move according to the random waypoint model (RWP). In this model, each node remains static during a random pause interval, then it selects a random destination point in the simulation area and moves towards

the destination at a random speed. Each time the node reaches its destination, the process is repeated. In the scenario considered in this subsection, the speed and the pause time of each node are determined using a uniform random distribution within the intervals [0, v_{max}] m/s and [0, 20] s, respectively. The total traffic load in the network is 1,620 kbps.

Our evaluation starts by analyzing the impact of two critical parameters of the Hello message-based connectivity maintenance mechanism on the performance of the flows in a MWN, for different node speeds. The results are used to determine the settings for these parameters in the next subsection. The parameters under consideration are the frequency of Hello messages and the number of allowed Hello losses. In default AODV, these parameters determine the maximum Hello wait time, i.e., the time interval that a node will wait for a Hello message from a neighbor before deciding that the corresponding link has become broken [7,8]. In addition, in DEMON, since Hello messages are needed to compute the pETX metric of the links, their frequency determines the time required to detect performance degradation. Thus, adequate tuning of these parameters is critical for achieving good network performance.

Figure 9a,b show the results obtained when using default AODV with ETT and WCIM metrics for route discovery, respectively. When mobility increases, performance improves by means of using low allowed Hello losses and low Hello interval (which determine a low Hello wait time). On the other hand, under low mobility conditions, the best results are obtained when Hello wait times are high for all the routing metrics and mechanisms. This is because when mobility is low, link breaks are mainly caused by congestion or interference between active flows. In these conditions, Hello packets, which are neither acknowledged nor retransmitted at the link layer, are more likely to be unsuccessfully delivered than data packets. According to the results, despite the Hello packet losses, it is better to continue using a link of moderate quality for data exchange than declaring the link as broken and incurring the overhead of finding a new route, which may suffer the same problems as those of the old route.

The AODV specification defines by default a time between Hello messages of 1 s and an allowed loss of three consecutive Hellos [7]. These settings yield good results in the considered scenario. Note that setting the allowed Hello loss to a value smaller than three might lead to spurious link break detection, given the variable link quality that is characteristic of wireless links.

Figure 10 shows the goodput obtained by using DEMON. All the different thresholds led to similar results in this scenario; the figure shows the average of the results obtained by using all of the different thresholds. As speed increases, performance is more sensitive to the frequency of Hello messages than to the number of allowed Hello messages lost. Since link quality monitoring uses Hello messages for exchanging message transmission count information between neighbors, a higher frequency of these messages allows a faster reaction to performance degradation. For example, in high-mobility cases, a faster reaction is necessary in order to recover routes before a link break occurs. Similarly to default AODV, under low mobility conditions, better results are obtained when Hello wait times are high.

As shown in Figure 11, in average, DEMON achieves up to 80% to 90% goodput increase with respect to default AODV, when both use ETT or WCIM.

Table 2 shows a qualitative comparison of the evaluation results for the state-of-the-art preemptive proposals. In general, topology and mobility conditions used in the

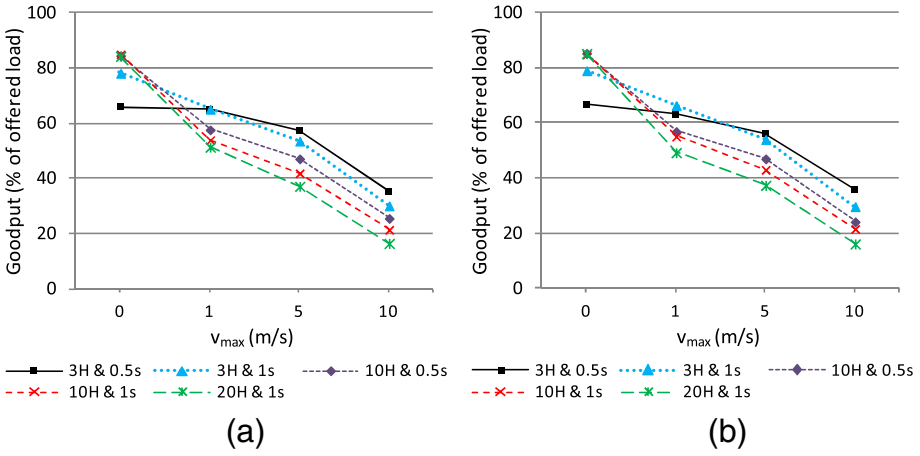

Figure 9 Study of Hello message connectivity maintenance parameters using default AODV. (a) Goodput obtained by using the ETT metric for route discovery; **(b)** goodput obtained by using the WCIM metric for route discovery. The notation '*x*H & *y*s' denotes an allowed loss of *x* Hellos and a time between Hello messages of *y* seconds.

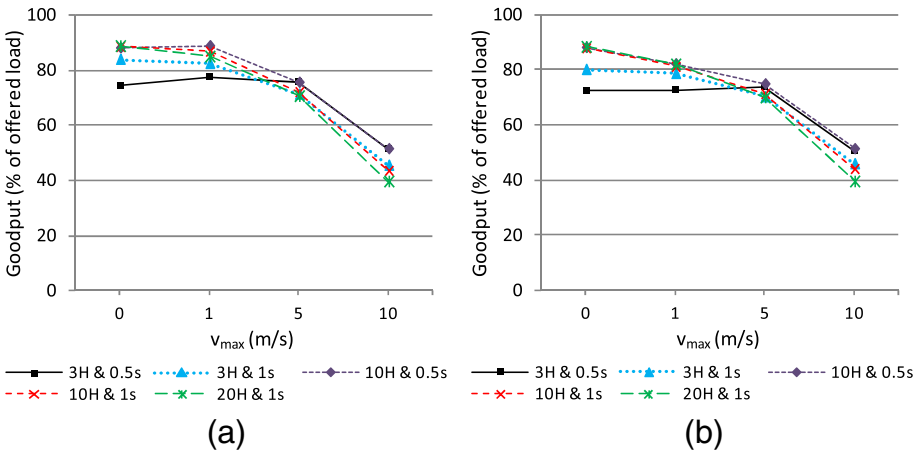

Figure 10 Study of Hello message connectivity maintenance parameters using DEMON. (a) Goodput obtained by using the ETT metric for route discovery. **(b)** Goodput obtained by using the WCIM metric for route discovery. The notation '*x*H & *y*s' denotes an allowed loss of *x* Hellos and a time between Hello messages of *y* seconds.

evaluation of state-of-the-art proposals are comparable to those of this scenario, although some works analyze other conditions such as higher mobility or lower load. As shown in Table 2, improvement in state-of-the-art preemptive solutions with regard to the basic versions of the corresponding routing protocols is similar to or lower than the one obtained by DEMON.

4.3 Analysis of DEMON performance under interference

We next analyze the performance of DEMON in six scenarios, where the main cause of performance degradation is the interference caused by the increasing load of the active UDP, CBR flows. Each scenario has particular conditions in terms of offered load, number of flows, spatial node distribution, link rates, node mobility, and use of single- or multi-radio nodes. Table 3 summarizes the main characteristics of the six scenarios.

Based on the results of Section 4.2, we set the allowed Hello loss to 10 and the Hello interval to 1 s for both default AODV and DEMON in all scenarios, except for scenario 3. In this latter scenario, where node mobility is high, we set an allowed Hello loss of three for default AODV, since it provides better performance, while the allowed Hello loss of ten was still used in DEMON.

As previously analyzed, the state-of-the-art preemptive solutions are insensitive to performance degradation due to reasons other than mobility, such as interference and congestion. Therefore, in scenarios without mobility (i.e., scenarios 1, 2, 5, and 6), we can assume that these solutions would obtain results similar to those of default AODV.

4.3.1 Scenario 1: stationary grid

In the first scenario, the nodes are stationary and are located in a regular grid topology of 8 × 8 nodes, with a distance between consecutive nodes in the same row or column of 140 m. Thirty-two flows are present in this scenario, each with randomly chosen source and destination.

Figure 12 shows the results for this scenario in terms of goodput. In this scenario, performance degradation is caused by congestion and interference between active flows. In low load conditions, the improvement of DEMON is minor, since flows suffer hardly any performance degradation. Under high load, congestion and interference degrade the quality of links. In this case, DEMON outperforms default AODV by properly reallocating flows in the network.

We next analyze the behavior of the different thresholds used in DEMON. The highest fixed threshold (i.e., 90%) obtains good results under low-load conditions. However, as load increases, the performance of this threshold

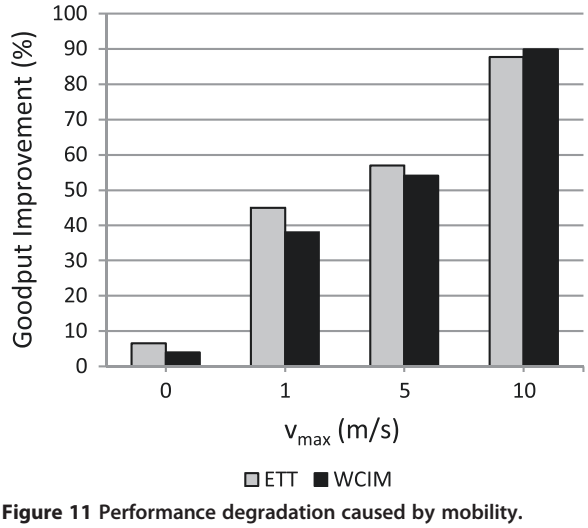

Figure 11 Performance degradation caused by mobility. Goodput improvement obtained using the DEMON extension.

Table 2 Comparison of the improvement and evaluation details of state-of-the-art preemptive solutions

Solution name	Number of nodes	Mobility	Load conditions	Goodput gain of preemption
H-AODV [10]	80	RWP	CBR: 40 flows of 16 kbps and 30 s of duration	10% to 25%
		Max. speed 10 m/s		
		Min. pause time 7 s	Simulation time of 300 s	
CPDSR [13]	50	RWP	TCP: 50 FTP connections	20% to 30%
		Max. speed 30 m/s		
		Min. pause time 1 s	Simulation time of 200 s	
LAW [14]	50	RWP	TCP: 50 FTP connections	20% to 30%
		Max. speed 20 m/s		
		Min. pause time 0 s	Simulation time of 500 s	
AODV-PSR [15]	50	RWP	CBR: 40 flows of 12 kbps	10% to 15%
		Max. speed 20 m/s	Simulation time of 900 s	
		Min. pause time 0 s		
RELREC [16]	200	RWP	CBR: two flows of variable load: 8 to 64 kbps	0% to 5%
		Max. speed 10 m/s		
		Min. pause time 20 s	Simulation time of 5,000 s	

The works [9], [11], [12], and [17] do not include an evaluation of the solutions proposed therein in terms of goodput.

decreases, and even becomes similar to that of default AODV. As shown in Figure 13, when using this threshold, the number of route recoveries per flow significantly increases with the offered load. This is due to the fact that because of congestion and interference, the number of links with an LSR below 90% increases significantly; hence, the threshold of 90% becomes unsuitable for this scenario. On the other hand, the lower fixed thresholds yield a better performance, since they lead to a smaller amount of route recoveries.

Under moderate to high load, the dynamic threshold performs better than the highest fixed threshold, but worse than the lower ones. Because the dynamic threshold is set on the basis of link quality at the instant of route discovery, it underperforms, since link qualities in this scenario are highly variable due to the presence of a high number of flows. Nevertheless, the dynamic threshold achieves the objective of avoiding the high number of route recoveries that occur with the highest threshold.

Figure 14 shows the average packet end-to-end delay for this scenario. As observed in the goodput results, in this scenario, the DEMON extension also outperforms basic AODV in almost all considered cases. Again, the highest fixed threshold obtains the best results under low-load conditions, but the performance of this threshold degrades as interference increases. In contrast, the rest of the evaluated thresholds perform similarly and better than the highest fixed threshold under high load. According to the results depicted in Figure 14, we conclude that contention and interference affect significantly the packet end-to-end

Table 3 Simulation study scenarios

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Type of scenario	Stationary grid	Stationary grid with a gateway	High mobility	Low mobility	Stationary random distribution	Stationary multi-radio grid
Node location	Grid	Grid	Random	Grid	Random	Grid
Number of flows	32	32	6	6	6	32
Sources	Random	Random	Random	Random	Random	Random
Destinations	Random	One, fixed	Random	Random	Random	Random
Minimum offered load (Mbps)	1.60	1.60	0.54	1.62	0.54	1.60
Maximum offered load (Mbps)	4.00	4.00	3.24	3.24	2.16	6.40
Link rate (Mbps)	12	12	12	12	12	6
Multi-radio	No	No	No	No	No	Yes
Mobility	No	No	Yes	Yes	No	No

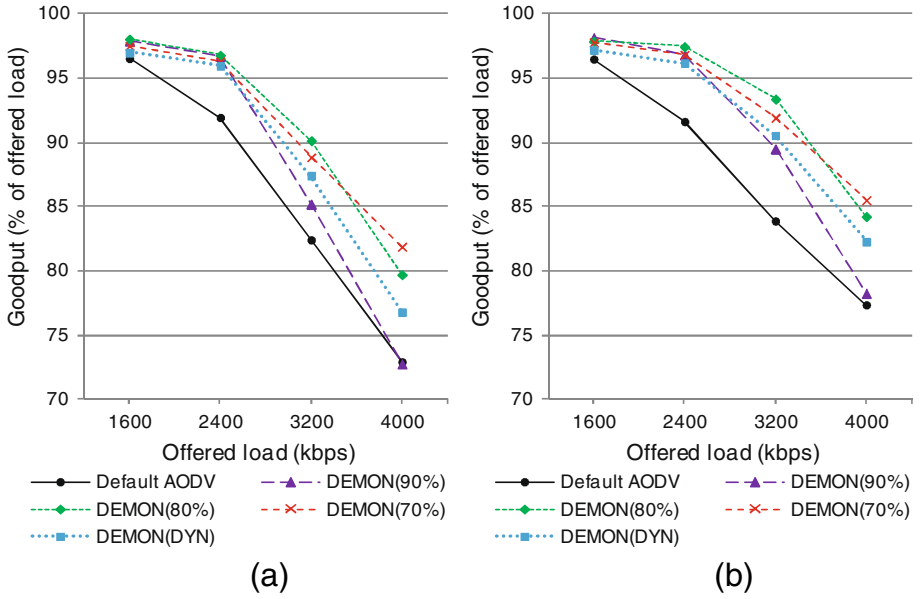

Figure 12 Scenario 1 results: goodput. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery. DEMON(*X*) denotes the use of DEMON, where *X* is a percentage or 'DYN' if a fixed or the dynamic threshold is used, respectively.

delay. DEMON rerouting strategy improves performance in terms of delay, even though it generally leads to longer routes in order to avoid congestion. On the other hand, we have found that, in this scenario and in the following ones, the end-to-end delay performance of both basic AODV and DEMON is almost complementary to their goodput performance (i.e., when a configuration provides high goodput, it provides also low delay, and vice versa). Therefore, for the sake of avoiding redundancy, henceforth we focus the analysis of DEMON on the basis of goodput results.

Finally, note that as shown in Figure 13, the use of the DEMON extension and ETT as the routing metric leads generally to a higher number of route recoveries than DEMON with WCIM. Since the WCIM routing metric is load-aware, it takes into account interference and congestion, which provides a better route selection than ETT, for both route discovery and route recovery. For this reason,

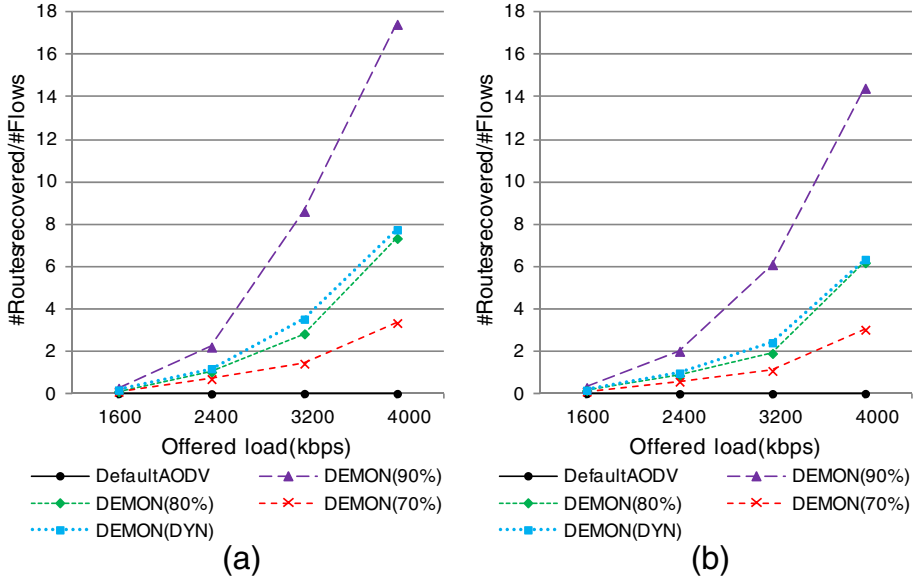

Figure 13 Scenario 1 results: route recoveries per flow. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

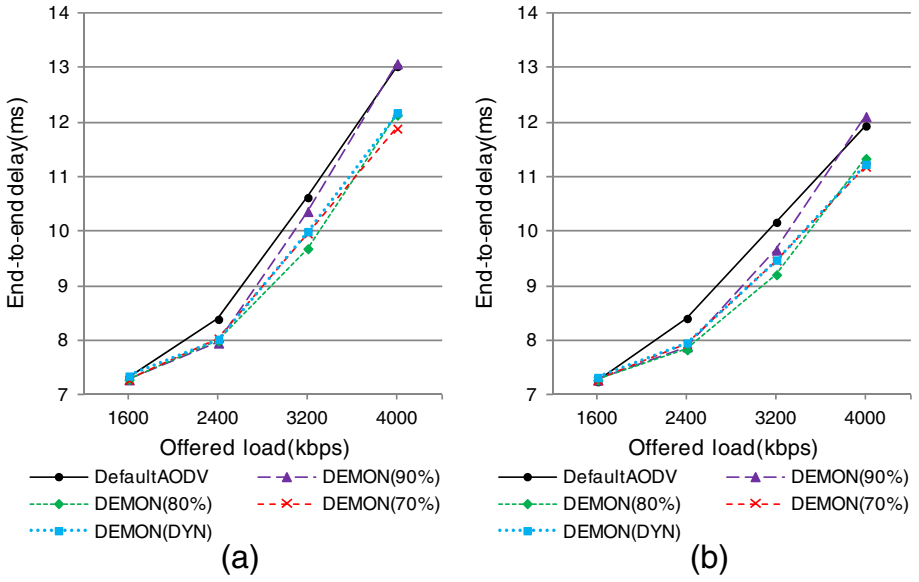

Figure 14 Scenario 1 results: average packet end-to-end delay. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

in this scenario and also in the following ones, DEMON achieves usually better performance using WCIM than with ETT.

4.3.2 Scenario 2: stationary grid with a gateway

In the second scenario, we analyze a network in which a node acts as a gateway (e.g., offers connectivity to the Internet). Thirty-two flows with a randomly selected source are again present in the network, but in this scenario, the destination is the same for all these flows (the destination is the node at the bottom right end of the regular 8×8 grid). The remaining scenario characteristics are the same as those in the first scenario.

Figure 15 shows the results for the second scenario. Once again, DEMON outperforms default AODV. In this scenario, the difference between the performances provided by using the different thresholds is minor. The reason for this result is that the destination for all the flows is the same node. Therefore, the route alternatives for avoiding interference in the area close to the gateway are limited.

4.3.3 Scenario 3: high mobility

In this scenario, the nodes are initially placed in a randomly chosen location and then move according to the RWP. The speed and the pause time of each node are determined using a uniform random distribution within the intervals [0, 5] m/s and [0, 20] s, respectively. There are six flows with random source and destination.

As shown in Figure 16, DEMON clearly outperforms default AODV, achieving from 30% up to 90% increase in goodput, depending on the offered load, for both ETT and WCIM routing metrics. As the load increases, the goodput gain of DEMON compared with default AODV decreases; this is because default AODV suffers from long disconnections due to route change latency. Thus, a small amount of data traffic is actually transmitted, and the degree of congestion is low regardless of the load offered to the network. In contrast, DEMON allows a greater fraction of data traffic to be transmitted, but the goodput, expressed as a percentage of offered load, decreases as load increases due to congestion.

On the other hand, Figure 16 shows that the different thresholds used in DEMON lead to similar goodput results, especially when the WCIM metric is used. In the case of the ETT metric, the fixed threshold of 90% offers a slightly better performance than the other thresholds. This phenomenon occurs because, in this scenario, link performance degradation is mainly caused by mobility, and therefore a high threshold provides a fast reaction to link breaks.

4.3.4 Scenario 4: low mobility

In the fourth scenario, the nodes are first located in the same grid used in scenario 1. However, after initialization, the nodes select a random direction and move linearly in that direction. The speed of each node is determined by using a uniform random distribution within the interval [0, 0.1] m/s. Under the low speed conditions of this scenario, the number of link breaks is low. Note that the unicast transmission rate of the nodes is fixed at 12 Mbps, while neighbor discovery is based on Hello messages, which are broadcasted at 6 Mbps. Compared to the previous scenario, low mobility causes some nodes to remain for a longer time in the zone where links are unable to transmit data packets successfully, but are considered

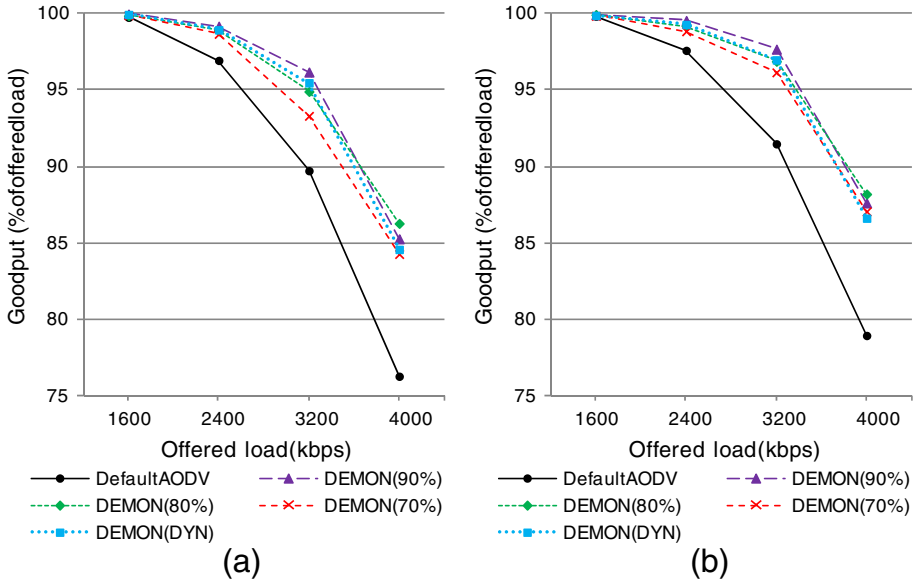

Figure 15 Scenario 2 results: goodput. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

active links by default AODV, since Hello messages are not lost (this zone has been called the *gray zone* [21]). The number of flows in the scenario is six, each with randomly selected source and destination.

As shown in Figure 17, the gray zone problem causes significant performance degradation of default AODV, even under low offered load. In contrast, since DEMON detects link performance degradation based on the link data loss rate, it is able to detect when a link becomes degraded and recover the corresponding routes.

With regard to the link quality thresholds tested for DEMON, in contrast to previous scenarios, under high load, the dynamic threshold obtains the best results using WCIM. This is due to the fact that in this scenario, the network conditions are less variable than those in the previous ones because of the smaller number of flows and the lower mobility. Link qualities are therefore more stable, and the dynamic thresholds, which are set during route discovery, are consistent with the network conditions for long time intervals. Nevertheless, the dynamic threshold

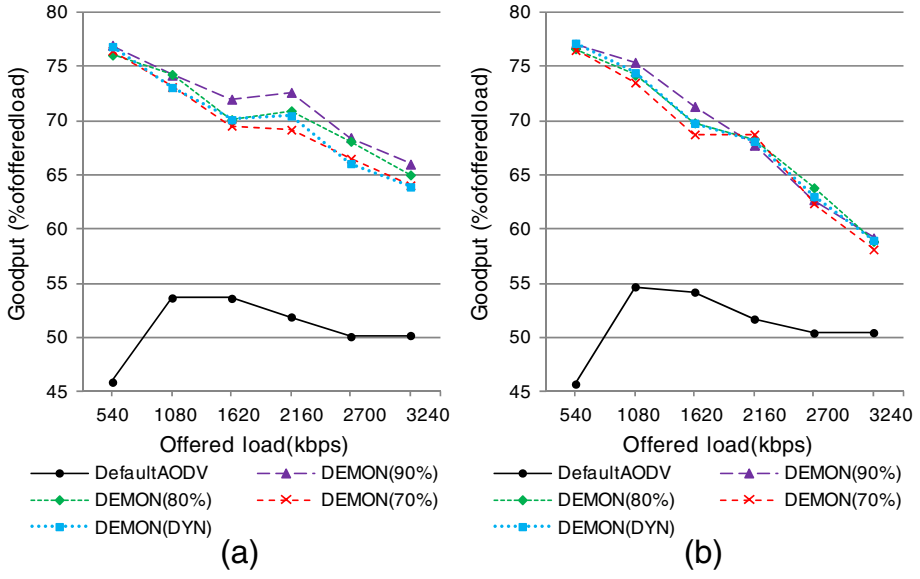

Figure 16 Scenario 3 results: goodput. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

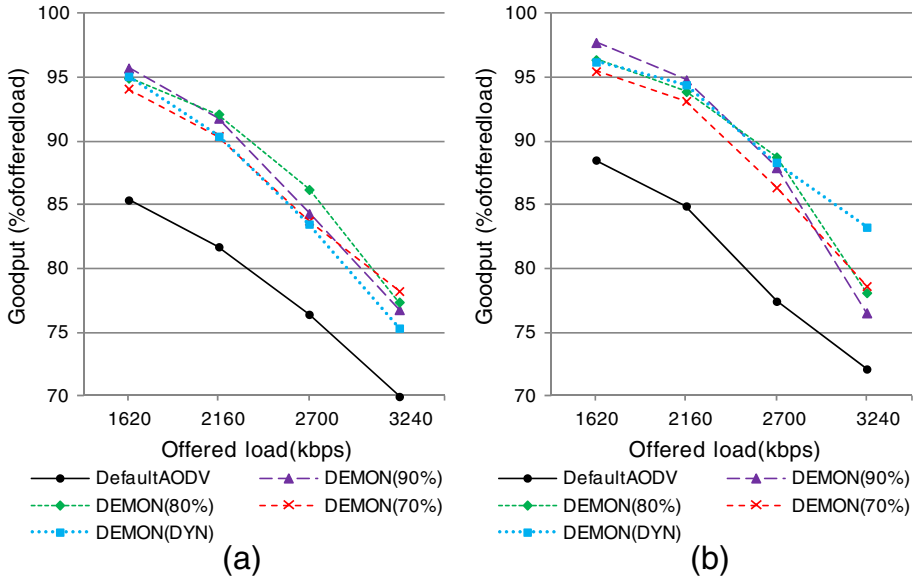

Figure 17 Scenario 4 results: goodput. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

underperforms when ETT is used. When using ETT, DEMON leads to a high number of rediscoveries due to its load unawareness (see Figure 13). This problem restricts the benefits of the dynamic threshold when the ETT metric is used.

4.3.5 Scenario 5: stationary random node spatial distribution

The fifth scenario has the same characteristics as the previous one, except for the fact that the nodes are statically located in the simulation area using a uniform random distribution. Since in this scenario the number of neighbors of a node is variable and sometimes scarce, the number of alternative routes that can be used is also limited. Thus, network congestion dramatically increases with the offered load. For this reason, the range of offered loads is smaller than the one evaluated in previous scenarios.

Figure 18 shows the obtained results; once again, DEMON outperforms the default AODV. Since network congestion is high, the fixed threshold of 80% gives better results than the 90% threshold, which gives rise to a large number of route recoveries even under low load conditions. Furthermore, in this case, the dynamic threshold yields moderate to good performance. On the other hand, for the highest load tested, the best threshold is the 70% one, since the LSR of several links is below 80% due to congestion.

4.3.6 Scenario 6: stationary multi-radio grid

The sixth scenario simulates a multi-radio network. Each node has two radio interfaces and randomly chooses two channels from three available orthogonal channels. In this way, we assure that each pair of neighbors shares at least one common channel. The data rate of the links is set to 6 Mbps. As in scenario 1, there are 32 flows with random source and destination, and the nodes are stationary and located in a grid topology. The results obtained in this scenario are plotted in Figure 19.

Figure 19 shows that DEMON again provides considerably higher goodput than default AODV. Compared with scenario 1, the offered load can be increased significantly without leading to congestion or inter-flow interference. Due to the use of two different radios (with orthogonal channels) by each node, route recovery has more alternative routes for avoiding interference and congestion than in single-radio scenarios. As in the previous scenarios, the different thresholds lead to a similar performance in low-load conditions, while in high-load conditions, the low fixed thresholds give better results since they are better suited to the actual LSR of the links and lead to a lower number of route recoveries.

4.4 Analysis of DEMON performance with TCP

In the previous scenarios, we analyzed the behavior of DEMON using UDP as the transport protocol, where the sources of the flows sent data packets at a constant rate, regardless of the network state. TCP, on the other hand, implements different mechanisms in order to adapt the data rate to its view of the congestion state of the network [34]. In this section, we study the performance of DEMON when TCP is used as the transport protocol of the data flows.

We consider two scenarios. In the first one, we simulate an 8×8 static grid with a varying number of TCP flows, where congestion and interference are the main

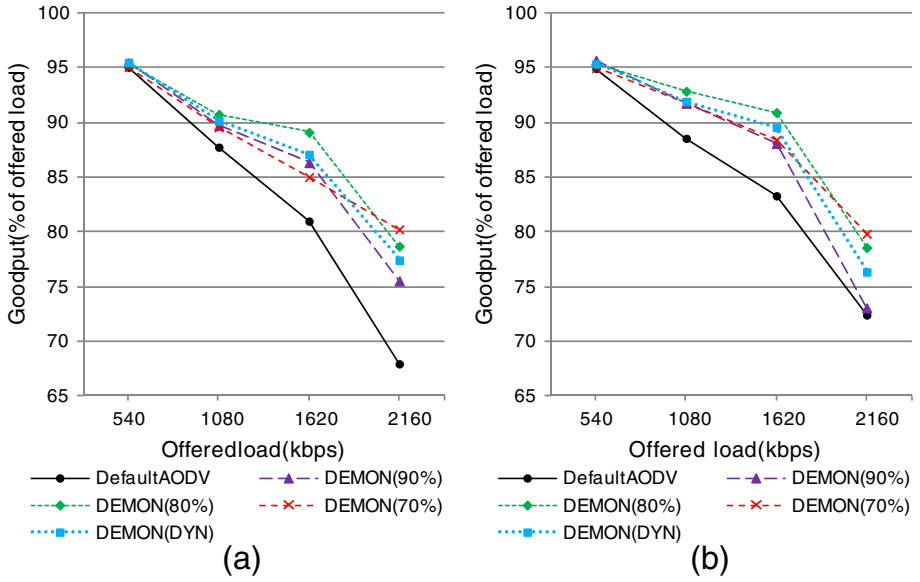

Figure 18 Scenario 5 results: goodput. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

causes of performance degradation. In the second one, 64 nodes are initially placed in a randomly chosen location and then move according to the random waypoint model (RWP). In this latter scenario, the number of TCP flows is fixed to four, while the maximum speed of the nodes is varied. We have used the ETT routing metric in this evaluation. Performance is analyzed in terms of goodput, and the results of both scenarios are shown in Figure 20a,b, respectively.

Figure 20 shows that in both scenarios, DEMON improves performance of the TCP flows, achieving in average 5% to 10% of goodput increase. However, compared to the UDP scenarios analyzed in the previous subsection, the improvement of DEMON under mobility is less remarkable. We found that two of the main mechanisms of the TCP congestion control, the congestion window limit (CWL) and the retransmission timeout (RTO), limit the benefits of DEMON. The negative impact of these two mechanisms on performance of multi-hop wireless networks is a well-known problem [35]. Due to the inter- and intra-flow interference, the CWL should be set to low values in order to avoid heavy congestion at the MAC layer [36]. In fact, in all our simulations, we have obtained the best performance using a CWL of one maximum

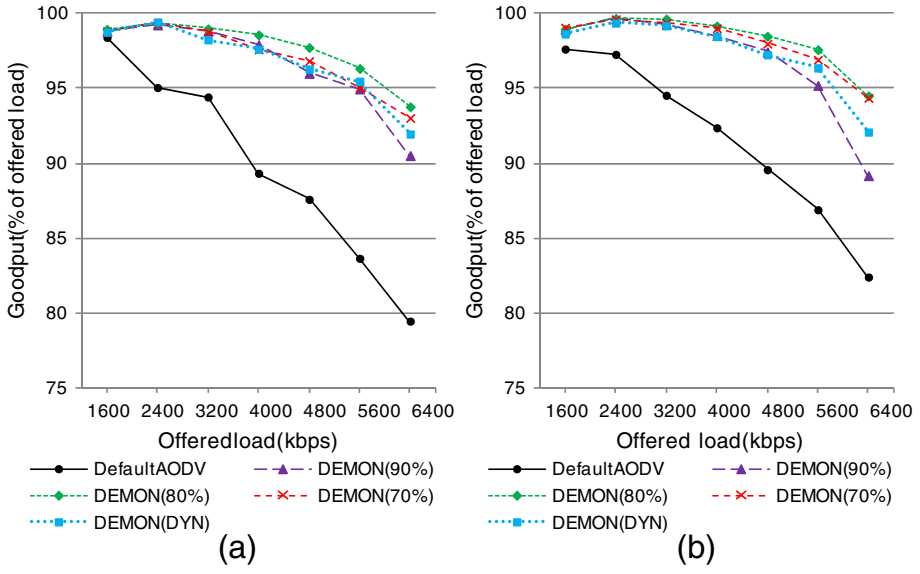

Figure 19 Scenario 6 results: goodput. (a) ETT metric for route discovery. **(b)** WCIM metric for route discovery.

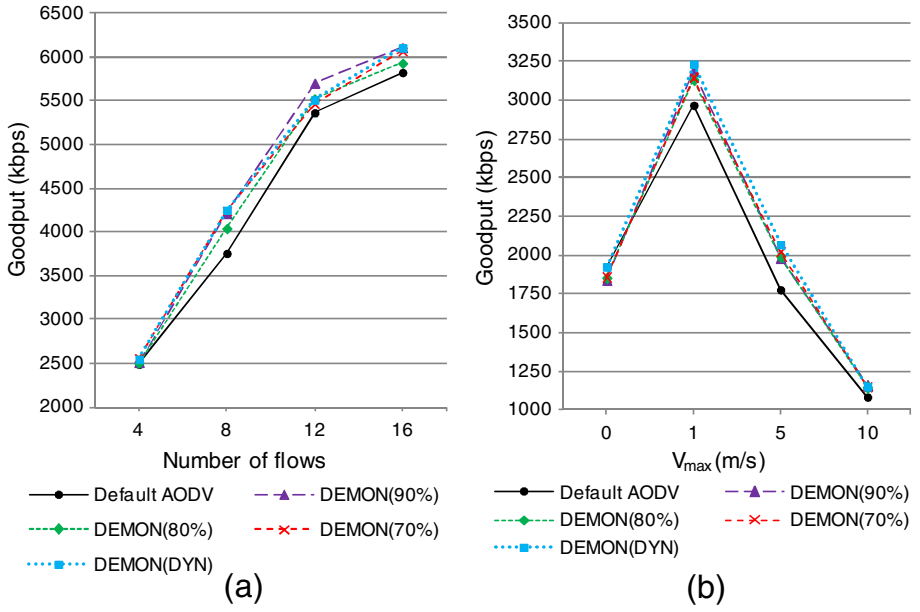

Figure 20 TCP analysis results: goodput. (a) Increasing number of flows. **(b)** Increasing maximum speed.

segment size, both for basic AODV and DEMON. This was also the case in other state-of-the-art studies [36,37]. On the other hand, TCP RTO is frequently triggered in wireless multi-hop networks due to transmission errors. In such case, the exponential increase of TCP RTO as a congestion control mechanism is unnecessarily activated, which reduces significantly the sending rate of the TCP sources despite the absence of congestion [38].

In our simulations, the combination of the two described phenomena causes the source node sending rate to decrease down to values in the order of one packet per second (i.e., the minimum TCP RTO). This situation limits the applicability and accuracy of DEMON, which is based on traffic monitoring, since very few or even no packets are transmitted in some Hello cycles. This is especially detrimental in mobile scenarios, whereby detecting the presence of nodes in a gray zone cannot be done as rapidly as in the UDP scenarios.

We found that in mobile scenarios, DEMON results are slightly worse than those of CPDSR and LAW (see Table 2). Since the link quality estimation mechanism of CPDSR and LAW is based on the received power of data packets, they are less affected than DEMON by the data rate decrease due to TCP congestion control: a single received packet under the received power threshold is taken as a sign of mobility and triggers route recovery. However, both CPDSR and LAW are applied to the DSR routing protocol and use cache-enabled route recovery, which improves their results [13].

From the analysis of the scenarios considered in our evaluation, we conclude that in order to improve the performance of DEMON in the presence of TCP flows, additional cross-layer mechanisms are needed. One possibility is providing DEMON with mechanisms at the end nodes capable of monitoring the state of TCP algorithms, in order to identify signs of performance degradation, such as the number of TCP retransmissions or RTO variance. On the other hand, it would be interesting to evaluate DEMON when improved TCP mechanisms are used. These include state-of-the-art TCP modifications designed for wireless multi-hop networks, like the use of adaptive CWLs [39] or algorithms to distinguish the cause of RTO expirations [38]. These studies are left for future work.

5 Conclusions

In this paper, we present a novel preemptive route recovery extension of AODV called DEMON. The design of DEMON was motivated by the fact that, to the best of our knowledge, state-of-the-art preemptive route recovery solutions for on-demand MWN routing protocols are only designed with the objective of minimizing route disconnections due to node mobility. However, a comprehensive preemptive solution should consider other causes of route performance degradation, such as inter-flow interference or congestion, and have appropriate route maintenance mechanisms in order to re-route active flows if their performance becomes compromised.

DEMON uses pETX, a link quality metric based on a passive estimation of link data loss rate for monitoring the performance of active links. When a node detects that link quality falls below a threshold (which can be statically or dynamically configured), route recovery is triggered in order to find a new route for the affected

flow whose recovery optimizes network performance. DEMON performs preemptive, source-based route recovery due to any cause of link quality degradation. In contrast with state-of-the-art solutions, DEMON can operate regardless of the link layer implementation since it only uses information available at the network layer.

We study the performance of DEMON in IEEE 802.11-based MWNs by means of simulation, to which end we consider a wide range of scenarios, each of them with particular characteristics in terms of offered load, number of flows, spatial node distribution, link rates, mobility, and use of single- or multi-radio nodes. According to the results, DEMON outperforms default AODV in all the scenarios used in the evaluation due to its preemptive recovery of degraded routes. In mobile scenarios, DEMON can provide an increase of up to 90% of goodput by anticipating link breaks and minimizing route disconnections. The improvement is equal to or greater than that achieved by state-of-the-art preemptive solutions according to the literature. In stationary networks, the average improvement is smaller, about 10% to 30%, since as load increases, congestion and interference leave little margin for improving performance by changing the flow distribution in the network. In the absence of mobility, the performance of state-of-the-art preemptive solutions would be analogous to that of default AODV.

With regard to the link quality thresholds used in DEMON, fixed thresholds set to high values obtain good results under low-load or high-mobility conditions, since they provide a fast reaction to performance degradation. However, in congested networks, fixed thresholds set to high values may lead to excessive route recoveries, which become counterproductive. On the other hand, fixed thresholds set to low values are less sensitive to performance degradation under low load or high mobility; however, under high load, they are more consistent with actual link qualities and thus give rise to a low number of route recoveries, which leads to good performance. In most considered cases, the best performance is obtained by using a fixed threshold appropriately tuned to the scenario. Nevertheless, the dynamic threshold performs reasonably well in all the scenarios and load conditions. Even so, we plan to study enhancements to the dynamic threshold in order improve its adaptability to the characteristics of a scenario. In particular, networks of varying conditions constitute a challenge.

We analyze the behavior of DEMON when two different state-of-the-art routing metrics are used for route discovery, ETT and WCIM, which are link quality and load aware, respectively. In most cases, WCIM obtains better results than ETT, since its load awareness leads to better route selection. However, results show that in some cases, the performance of DEMON improves by using the ETT metric for route discovery. This is an interesting feature, since the use of load-aware metrics can lead to network instability under highly variable network conditions. By using DEMON, the routing metric can remain load-unaware, as is ETT, while route recovery can be load-aware by means of the link quality estimation mechanism based on pETX. In fact, according to the obtained results, we can conclude that the route recovery mechanisms have a greater influence on the performance of the routed flows than the routing metrics used for route discovery. This is remarkable, since the state-of-the-art on preemptive and route recovery solutions is scarce compared to the wide literature on routing metrics and route discovery solutions.

Finally, we evaluate the performance of DEMON in combination with TCP congestion control mechanisms. DEMON can provide an average increase of up to 5% to 10% of network goodput in the presence of TCP flows. However, we find that in mobile scenarios, TCP congestion control mechanisms dramatically decrease the packet sending rate, which reduces the accuracy of DEMON and limits its benefits. A future work item is the evaluation of DEMON with state-of-the-art TCP improvements for wireless multi-hop networks. Also, we will study the addition of cross-layer mechanisms for DEMON at the end nodes, making use of transport layer information in order to detect performance degradation.

Future work will also include investigating strategies for using DEMON as the basis of a channel assignment mechanism for multi-radio MWNs. In this case, instead of recovering the whole route when performance becomes degraded, it is possible to assign a new channel for the degraded link by exploiting the multi-radio capabilities of the nodes.

Endnotes

[a]The main parameters of the simulation are summarized in Table 1 (see Section 4). The frequency of Hello messages is one packet per second.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This work is supported by the Spanish Government through the MICINN project TEC2009-11453, the Ministerio de Economía y Competitividad project TEC2012-32531, the FPU MEC fellowship, and FEDER.

Author details
[1]Wireless Networks Group, Telematics Engineering Department, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB), Jordi Girona 1-3, Barcelona 08034, Spain. [2]Wireless Networks Group, Telematics Engineering Department, Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels (EETAC), Esteve Terradas 7, Castelldefels, Barcelona 08860, Spain. [3]i2CAT Foundation, Gran Capita 2–4 (Nexus building), Barcelona 08034, Spain.

Received: 22 April 2013 Accepted: 2 December 2013
Published: 17 December 2013

References

1. Y Yang, J Wang, R Kravets, Designing routing metrics for mesh networks, in *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)* (Santa Clara, 2005)
2. MEM Campista, PM Esposito, IM Moraes, LHMK Costa, OCMB Duarte, DG Passos, CVN de Albuquerque, DCM Saade, MG Rubinstein, Routing metrics and protocols for wireless mesh networks. IEEE Network **22**, 6–12 (2008)
3. D Aguayo, J Bicket, S Biswas, G Judd, R Morris, Link-level measurements from an 802.11b mesh network. SIGCOMM Comput. Commun. Rev. **34**, 121–132 (2004)
4. T Clausen, P Jacquet, L Viennot, Comparative study of routing protocols for mobile ad-hoc networks, in *Proceedings of the First Annual Mediterranean Ad Hoc Networking Workshop* (Sardegna, 2002)
5. J Li, C Blake, DSJD Couto, HI Lee, R Morris, Capacity of ad hoc wireless networks, in *International Conference on Mobile Computing and Networking* (Rome, 2001)
6. J Lee, S-J Lee, W Kim, D Jo, T Kwon, Y Choi, Understanding interference and carrier sensing in wireless mesh networks. IEEE Commun. Mag. **47**, 102–109 (2009)
7. C Perkins, E Belding-Royer, S Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561 (The Internet Society, Reston, 2003)
8. C Gomez, M Catalan, X Mantecon, J Paradells, A Calveras, Evaluating performance of real ad-hoc networks using AODV with Hello message mechanism for maintaining local connectivity, in *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications* (Berlin, 2005), pp. 1327–1331
9. T Goff, NB Abu-Ghazaleh, DS Phatak, R Kahvecioglu, Preemptive routing in ad hoc networks, in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)* (Rome, 2001), pp. 43–52
10. P Srinath, P Abhilash, I Sridhar, Router handoff: a preemptive route repair strategy for AODV, in *IEEE International Conference on Personal Wireless Communications* (New Delhi, 2002), pp. 168–171
11. S Crisostomo, S Sargento, P Brandao, Improving AODV with preemptive local route repair, in *International Workshop on Wireless Ad-Hoc Networks* (Oulu, 2004), pp. 223–227
12. E Weiss, G Hiertz, B Xu, S Hischke, B Walke, S Gross, Improving routing performance in wireless ad hoc networks using cross-layer interactions. Ad Hoc Netw. **5**, 579–599 (2007)
13. W Zhu, X Zhang, Improve preemptive routing performance in mobile ad hoc networks with Cache-enabled method, in *Third International Conference on Communications and Networking in China* (Hagzhou, 2008), pp. 732–736
14. W Zhu, X Zhang, N Li, Improve TCP performance with link-aware warning method in mobile ad hoc networks, in *4th International Conference on Wireless Communications Networking and Mobile Computing* (Dalian, 2008), pp. 1–4
15. H Soliman, M AlOtaibi, An efficient routing approach over mobile wireless ad-hoc sensor networks, in *6th IEEE Consumer Communications and Networking Conference* (Las Vegas, 2009), pp. 1–5
16. Z Liang, Y Taenaka, T Ogawa, Y Wakahara, Pro-reactive route recovery with automatic route shortening in wireless ad hoc networks, in *Tenth International Symposium on Autonomous Decentralized Systems* (Tokyo, 2011), pp. 57–64
17. ZK Lee, G Lee, HR Oh, H Song, QoS-aware routing and power control algorithm for multimedia service over multi-hop mobile ad hoc network. Wirel. Commun. Mob. Comput. **12**, 567–579 (2012)
18. D Johnson, D Maltz, *The Dynamic Source Routing Protocol (DSR)*. RFC 4728 (The Internet Society, Reston, 2007)
19. SJ Lee, EM Belding-Royer, CE Perkins, Scalability study of the ad hoc on-demand distance vector routing protocol. Int. J. Netw. Manag. **13**, 97–114 (2003)
20. ID Chakeres, EM Belding-Royer, The utility of hello messages for determining link connectivity, in *The 5th International Symposium on Wireless Personal Multimedia Communications* (Honolulu, 2002), pp. 504–508
21. H Lundgren, E Nordströ, C Tschudin, Coping with communication gray zones in IEEE 802.11b based ad hoc networks, in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM '02)* (Atlanta, 2002), pp. 49–55
22. C Gomez, P Salvatella, O Alonso, J Paradells, Adapting AODV for IEEE 802.15.4 mesh sensor networks: theoretical discussion and performance evaluation in a real environment, in *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (New York, 2006)
23. D Xu, R Bagrodia, Impact of complex wireless environments on rate adaptation algorithms, in *IEEE Wireless Communications and Networking Conference* (Cancun, 2011), pp. 168–173
24. D Couto, D Aguayo, J Bicket, R Morris, A high-throughput path metric for multi-hop wireless routing, in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)* (San Diego, 2003), pp. 134–146
25. R Draves, J Padhye, B Zill, Routing in multi-radio, multi-hop wireless mesh networks, in *International Conference on Mobile Computing and Networking* (Philadelphia, 2004)
26. M Catalan-Cid, JL Ferrer, C Gomez, J Paradells, Contention- and interference-aware flow-based routing in wireless mesh networks: design and evaluation of a novel routing metric. EURASIP J. Wirel. Commun. Netw. **2010**, 313768 (2010)
27. ZR Zaidi, S Shastry, What is wrong with broadcast probing based ETX estimation for wireless links? in *Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '12)* (Cyprus, 2012)
28. A Woo, T Tong, D Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in *Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys '03)* (Los Angeles, 2003)
29. Omnetpp, OMNeT++ simulator. http://www.omnetpp.org. Accessed 1 Sept 2012
30. M Takai, J Martin, R Bagrodia, Effects of wireless physical layer modeling in mobile ad hoc networks, in *International Symposium on Mobile Ad Hoc Networking & Computing* (Long Beach, 2001)
31. A Iyer, C Rosenberg, A Karnik, What is the right model for wireless channel interference? IEEE Trans. Wireless Commun. **8**, 2662–2671 (2009)
32. TR Andel, A Yasinsac, On the credibility of MANET simulations. Computer **39**, 48–54 (2006)
33. IEEE, IEEE Std 80211–2007, *IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements*. Part 11 (Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (IEEE, Piscataway, 2007)
34. J Postel, *Transmission control protocol*. RFC 793 (IETF Network Working Group, Fremont, 1981)
35. AM Al-Jubari1, M Othman, BM Ali, N Hamid, TCP performance in multi-hop wireless ad hoc networks: challenges and solution. EURASIP J. Wirel. Commun. Net. **2011**, 198 (2011)
36. M Gerla, K Tang, R Bagrodia, TCP performance in wireless multi-hop networks, in *Proceedings of the IEEE International Workshop on Mobile Computing Systems and Applications (WMCSA'99)* (New Orleans, Louisiana, 1999)
37. Z Fu, X Meng, S Lu, How bad TCP can perform in mobile ad hoc networks, in *Proceedings of the IEEE International Symposium on Computers and Communications (ISCC'02)* (Taormina, Italy, 2002)
38. M-Y Park, S-H Chung, Distinguishing the cause of TCP retransmission timeouts in multi-hop wireless networks, in *12th IEEE International Conference on High Performance Computing and Communication* (Melbourne, 2010)
39. K Chen, X Yuan, K Nahrstedt, On setting TCP's congestion window limit in mobile ad hoc networks, in *IEEE International Conference on Communications* (Alaska, 2003)

doi:10.1186/1687-1499-2013-286
Cite this article as: Catalan-Cid *et al.*: DEMON: preemptive route recovery for AODV in multi-hop wireless networks based on performance degradation monitoring. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:286.