Mucchi *et al. EURASIP Journal on Advances in Signal Processing* 2012, **2012**:170
http://asp.eurasipjournals.com/content/2012/1/170

EURASIP Journal on
Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

A robust cross-layer metric for routing protocol in mobile wireless ad hoc networks

Lorenzo Mucchi[1]*, Luigi Chisci[2], Luca Fabbrini[1] and Giulio Giovannetti[1]

Abstract

In a mobile ad-hoc network (MANET) where Mobile Nodes (MNs) self-organize to ensure the communication over radio links, routing protocols clearly play a significant role. In future MANETs, protocols should provide routing under full mobility, power constraints, fast time-varying channels, and nodes subject to high loading. In this article, a novel robust routing protocol, named distributed X-layer fastest path (DXFP), is proposed. The protocol is based on a cross-layer metric which is robust against the time-variations of the network as far as topology (mobility), congestion of the nodes and channel quality (fading, power constraints) are concerned. All these features are integrated in a single physical cost, i.e., the network crossing time, which has to be minimized. Furthermore, several routes from source to destination are stored for a given data flow to efficiently face the disconnections which frequently occur in MANETs. It is shown that the DXFP protocol, though locally operating in a fully distributed way within the MNs, provides, for each data flow, the optimum routes according to the considered metric. The DXFP protocol has been compared with two of the most commonly used routing protocols for MANETs, i.e., dynamic source routing and ad hoc on-demand distance vector, showing significant improvements in performance and robustness.

Keywords: Ad hoc network, Routing protocol, Cross-layer metric, Robustness, Wireless communication

Introduction

Wireless ad hoc networks are typically seen as networks without a fixed infrastructure, where the mobile terminals cooperate to assure the correct work flow of the network communications [1].

In ad hoc networks, the mobile terminals communicate with each other directly with no central unit that coordinates the overall traffic. Each mobile terminal must implement routing functionalities for the other terminals in the network, thus allowing communication also between terminals that do not have a direct link. When no infrastructure is present, it is extremely important to select the sequence of terminals which allow the communication path from the source to the destination node properly. The so-called routing protocols handle the problem of choosing and maintaining the paths through time, even when changes of the network topology occur.

Routing protocols have different classifications in the literature [2]. We refer here to the IETF MANET Working Group classification [3] which subdivides protocols into proactive (table-driven routing or source routing) and reactive (on demand or distributed routing) and considers a flat logic organization of the network where all the terminals have the same functionalities. In addition, hybrid schemes (or hierarchical routing) can be designed. Reactive protocols are very interesting for MANET applications because they send less control packets than proactive ones when the network topology changes frequently as typically occurs in MANETs. A survey of the current routing protocols based on routing philosophy structure can be found in [4,5].

In the literature, the first protocol to be introduced was dynamic source routing (DSR) [6] in which each packet transmitted by the source includes the complete path to the destination. Afterward, the ad hoc on-demand distance vector (AODV) protocol [7,8] was designed. This protocol uses a routing table and performs better than DSR, but is more difficult to implement because it uses advanced features like timers, sequence numbers, and

*Correspondence: lorenzo.mucchi@unifi.it
[1] Department of Electronics and Telecommunications, University of Florence, via Santa Marta 3, 50139, Italy
Full list of author information is available at the end of the article

promiscuous-mode listening. The dynamic MANET on-demand (DYMO) [9] protocol is a tradeoff between the low complexity of DSR and the higher performance of AODV. Normally, in all these protocols the mobile terminals store only one path from source to destination, but evolutions of both DSR [6] and AODV [10] protocols were proposed to have multiple routes stored in the terminals so that the secondary routes can be used as backup paths or to balance loads. In all the above-mentioned protocols, the adopted metric is the number of hops which are needed to reach the destination node, i.e., a network layer metric.

A routing metric based only on the number of hops, however, is not able to take into account all the features of the future mobile terminals operating in ad hoc networks. If it is believed that more information has to be taken into account to make the routing protocol smarter and more efficient.

Simple protocols, which use information of other layers to decide the routing (e.g., ABR, Associativity-Based Routing) [11] and System Stability-based Adaptive (SSA) routing [12]) have been proposed more recently. Given the improvements related to cross-layer approaches, more advanced metrics, taking into account many other network parameters, have been proposed. In particular, expected transmission count (ETX) [13], expected transmission time (ETT) [14], weighted cumulative ETT (WCETT) [14], and Metric of Interference and Channel-switching (MIC) [15] have been designed for fixed ad hoc networks but not much is known about their performance in MANETs.

Several routing protocols have been previously proposed, e.g., [16-26]. In particular, in [23] a new protocol, named Flow-Oriented Routing Protocol (FORP), for route rebuilding was proposed. FORP takes into account topology changes due to node mobility. In [24], the Zone Routing Protocol (ZRP) still for facing network topology changes due to node mobility was proposed. In [25], a protocol which takes into account the channel quality (large and small scale fading, and thus SNR) on a hop-by-hop basis to select the best route from source to destination was presented. In [26], a protocol for maximizing the network lifetime in terms of energy consumption was proposed and discussed. The algorithm selects the best position for the nodes and thus the network topology to minimize the energy consumption for relaying the information of other nodes. The metric makes use of real parameters such as distances between nodes, mobility rates, and energy consumptions for the relaying of packets by assuming that the larger the queue (node congestion) the higher is the cost to transmit them.

In this article, a new routing protocol, called Distributed X-layer Fastest Path (DXFP), is proposed. DXFP is designed to overcome the limitations of the current routing protocols for ad hoc networks as well as to fit the requirements for continuity and quality of service of the future wireless networks. The main features of DXFP are

- adoption of a cross-layer metric;
- implementation of a distributed algorithm which guarantees that the selected path is optimum according to the considered metric;
- use of backup routes to give continuity to the connection;
- implementation of a robust method for path discovery.

The proposed protocol is analyzed theoretically. Its optimality is demonstrated with respect to a generic additive metric. Moreover, simulation results show that the proposed DXFP protocol outperforms the most commonly used protocols DSR and AODV, especially in scenarios with congestions and mobile nodes in a low channel quality environment.

The rest of this article is organized as follows. In section "System model", the system model is presented. In section "DXFP protocol", the proposed protocol is introduced and the route discovery and maintenance are discussed. In section "Local connectivity management", the optimality of the DXFP algorithm is discussed and in section "Routing metric", the new cross-layer metric is presented. Section "Performance evaluation" shows a comparison between DXFP and existing protocols DSR and AODV by means of realistic simulation experiments. Finally, section "Conclusions" concludes the article.

System model

This section presents the network model which allows to mathematically formulate the routing problem.

The network is represented by a weighted graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes representing terminals and $\mathcal{L}$ is the set of links among terminals. Moreover, a metric function $c : \mathcal{L} \rightarrow \mathbb{R}^+$ that associates a nonnegative cost to each arc is introduced. Given two nodes $i, j \in \mathcal{N}$, $\mathcal{P}_{i,j}$ will denote the set of paths from source i to destination j. A path $p \in \mathcal{P}_{i,j}$ is represented by an ordered sequence of nodes, i.e., $p = \{n_1, \ldots, n_{|p|}\}$ where $n_1 = i$ is the source, $|p|$ is the number of path's nodes (cardinality of p), $n_{|p|} = j$ is the destination and any other nodes n_i $(i = 2, \ldots, |p| - 1)$ are called *intermediate*. Given the metric function $c(\cdot)$, the following cost is associated to the path p:

$$C(p) = \sum_{i=1}^{|p|-1} c\,(n_i, n_{i+1}). \tag{1}$$

The routing problem from source i to destination j consists of finding, in the graph $\mathcal{G}$, the path p^* from i to j of minimal cost, i.e.

$$p^* = \underset{p \in \mathcal{P}_{i,j}}{\arg\min}\ C(p). \tag{2}$$

DXFP protocol

DXFP is a reactive protocol, i.e., it looks for the path toward a destination node only when it needs to send data packets toward that node. Thanks to the periodical sending of short signalling messages each node is aware of its neighbors, i.e., the nodes which can be directly reached by a data packet. Each node calculates the so-called reachability parameter with respect to each neighbor. In particular, the reachability between two nodes connected by a single hop can be conventionally characterized by the packet error rate (PER) of the channel.

Whereas most existing routing protocols are destination-oriented, DXFP is flow-oriented in that packets are routed taking into account both source and destination. Moreover DXFP is a multipath routing protocol as it maintains multiple paths for the same source-destination flow and chooses to forward the packets of the given flow through the path of minimal cost. Flow-oriented routing has been adopted so as to cope in a better way with the high time variance of the network topology and of the link quality. According to the flow-oriented paradigm, whenever a new traffic session is initiated, a Route Discovery procedure is carried out on the basis of the most recent available network information. Moreover, it must be pointed out that the installation of a path modifies its quality, worsening the congestion of the terminals along the route. Hence, it is of paramount importance, for a correct operation, that each new traffic session probe the network state so as to be routed on the path of minimal delivery delay.

When a node S wants to reach a node D in the network, it starts the Route Discovery procedure by broadcasting a request packet which carries the address of D. The nodes receiving the request just forward it. When a request reaches the destination node, D answers by unicasting an answer packet towards the node from which it has received the request. The node which receives the answer writes its address in the packet, updates the packet metric, and forward the answer to all the neighbor nodes. Then, the node installs a path toward the destination for the flow identified by the pair (S,D). When an answer packet reaches the source S, S starts to send the data packets using the path associated to the received answer.

An intermediate node can receive more than one answer related to the same flow (S,D). In this case, the procedure of the intermediate node is to send the first received answer, minimizing the time needed to find a route between S and D. A subsequent answer will be sent toward the source S only if it yields an improvement in the metric field of the answer message, i.e., the value corresponding to the cost-to-go to destination D. All remaining answers will be stored in the routing table. Once the source receives the first route, it starts to send data packets through it. Then, S will evaluate the metric of each successive answer it will receive. If the metric of the incoming answer is better than the one it was using, S starts to send data packets using the new route.

It is important to note that in the routing table of each intermediate node i, associated to a flow (S,D), only the cumulative metric of the links of the sub-path between i and D and the address of the next hop are stored.

The process of maintenance of the paths is called route maintenance and starts whenever a node i detects a link failure. In this case, the current route is deleted and node i tries to use the best backup route for the same flow (S,D). Before being used, the backup route is tested by sending a short control packet. If the control packet does not reach the destination, the tested route is deleted and the subsequent one is tested. In case all the backup routes do not work, an error packet is sent back to the source. A node receiving the error packet starts to test its own backup routes. If the error packet reaches the source and no working routes are found, then the source will start a new Route Discovery procedure.

It is important to note that the proposed protocol can work with any path metric provided that it is additive with respect to the links of the path. The metric used by DXFP is the (statistical) mean latency $\bar{T}$ that a packet undergoes from S to D. Given two intermediate nodes i, j and the link between them (i,j), the metric can be defined as $\overline{T}_{i,j} = \overline{W}_i + \overline{R}_{i,j}$ where $\overline{W}_i$ is the average waiting time in the queue at node i and $\overline{R}_{i,j}$ is the average time to send a packet through the link (i,j) successfully.

The following sections describe in detail how the proposed DXFP builds and maintains the routes.

Messages

This section describes the messages used by DXFP to fulfill the assigned routing requirements. These messages are referred to as control messages or signalling messages. The feature they have in common is that they are sent with a random delay uniformly distributed (*jitter*[a]), in order to avoid that the messages generated by nodes having synchronous behavior collide in a random channel access network.

The messages of DXFP are:

- Route REQuest (RREQ): Request message generated by the source. It goes through the network to inform the destination that a data flow must reach it.

- Route REPly (RREP): Answer message generated by the destination of an RREQ. It crosses the network updating the routing tables. When an RREP reaches the source, the source is informed about the route to be used for sending the data flow.
- Route ERRor (RERR): Error message generated by a node which tries to forward packets to some destination but has no available route.
- Route TEST (RTEST): Test message generated by a node which has detected a broken link while sending packets. The message is forwarded to the destination to check its reachability.
- Route TEST Ack (RTEST_Ack): Answer message generated by the destination of the RTEST. The message retraces back the path of the RTEST to inform the node that has generated the RTEST that the path is checked and available for the data flow.
- HELLO: Broadcast message periodically generated by all the nodes. It informs a node about the presence of neighboring nodes. DXFP uses this message to measure the SINR and calculate the PER accordingly.

In Table 1, all the messages used by DXFP are listed, while in Table 2 the description of the functionality of each field is reported.

Local connectivity management

The local connectivity management consists of building a table in which, for each neighbor, the IP address and a measure of the quality of the associated link are stored.

DXFP uses the PER to characterize the link quality. This measure is obtained by means of the periodic HELLO messages. Each node A which receives a HELLO message from node B measures the signal to interference and noise ratio (SINR) of the link to estimate the associated PER. Each node, to carry out this calculation (which will be detailed in section "Average transmission time $\overline{R}$"), needs to know the packet length of the HELLO message, the modulation, the coding, and the measured SINR, which are assumed to be available at MAC level. It is important to point out that all the previous mentioned parameters are physical layer parameters, but we suppose to forward their values at MAC layer in order to implement the cross-layer approach.

Each node considers as neighbors only the nodes for which the associated link satisfies $\text{PER} < \text{PER}_{\text{th}}$, where PER_{th} is the maximum PER that a receiver can tolerate to guarantee a correct reception of messages.

In order to face the high time-variability of the channel, the measure of the PER is averaged over a suitable time window. In our experiments, every time a HELLO packet is received from a node, the mean PER is averaged over previously calculated PERs. The time window over which the average of PER is calculated is limited to 300 s, to follow the actual network state, i.e., any 300 s (any 60 receptions of the HELLO packet[b]) the PER information is reset.

Using an average PER to characterize the quality of each neighbor link, it is important to manage the case that a HELLO message is not received during the preset time. So, in the case of missed HELLO packets, it has been decided heuristically to penalize the link by associating to it a $\text{PER}_{\text{penalty}} = 2 \cdot \text{PER}_{\text{th}}$. This choice is to strongly penalize the use of such a link. The fact that a HELLO message is not received can be due to several reasons, for example, the mobility of the neighbor terminal, the fact that it is switched-off, its malfunction or a collision of the packet with other ones transmitted at the same time.

Routing table

Hereafter, a formal description of the routing table to be used in the subsequent algorithmic procedures is provided. The routing table $\mathcal{E}(i)$ of node $i \in \mathcal{N}$ consists of K entries representing different paths passing through node i and relative to active source-destination flows. The kth entry of the routing table ($k = 1, \ldots, K$) is a quadruple of the form $e_k(i) = (S_k(i), D_k(i), H_k^+(i), C_k(i))$ where the pair $(S_k(i), D_k(i))$ specifies the source-destination flow relative to the path associated to $e_k(i)$, the node $H_k^+(i)$ is the next hop of node i along the path associated to $e_k(i)$ and the nonnegative scalar $C_k(i)$ is the cost-to-go, from node i to the destination $D_k(i)$, of the path associated to $e_k(i)$. The

Table 1 DXFP messages and related fields

Message	Field (`header` and `type` are common to all messages)						
	`seq_num`	`src_addr`	`dest_addr`	`sender`	`hop`	`metric`	`route`
RREQ	✔	✔	✔	✘	✘	✘	✘
RREP	✘	✔	✔	✘	✔	✔	✔
RTEST	✔	✔	✔	✔	✘	✘	✔
RTEST_Ack	✔	✔	✔	✔	✘	✘	✔
RERR	✘	✔	✔	✘	✘	✘	✘
HELLO	✘	✘	✘	✘	✘	✘	✘

Table 2 Message fields description

Field	Length	Description
`header`	8 bit	General purpose currently not used
`type`	3 bit	Indicating the type of control message among the 6 possibilities listed above
`seq_num`	8 bit	Used to identify a message which has the same purpose
`src_addr`	32 bit	Carrying the IP address of the source
`dest_addr`	32 bit	Carrying the IP address of the destination
`sender`	32 bit	Used to define the node which has started a path testing
`hop`	8 bit	Storing the number of hops that the control message has undergone
`metric`	64 bit	Carrying the metric related to the path that the control message has done
`route_record`	(32 · `hop`) bit	The nodes append here their IP address

routing table $\mathcal{E}(i)$ is organized so that all path entries referring to the same source-destination flow are contiguous and ordered according to the increasing cost-to-go.

Route discovery

This section describes in detail the Route Discovery procedure. Just for the sake of simplicity but without any loss of generality, this section will make reference to a single source-destination flow (*S*,*D*). From the previous developments, it is clear that the algorithm consists of two steps: a *Forward Step* in which RREQ messages are propagated from the source to the destination and a *Backward Step* in which RREP messages are propagated backward from the destination to the source.

First, it is convenient to introduce the following notation:

- $\mathcal{V}(i) \subseteq \mathcal{N}$ is the set of neighbors of node *i*, i.e. the subset of nodes directly connected to node $i \in \mathcal{N}$;
- $T_{i,j} \in \mathbb{R}^+$ is the transition cost from node i to node $j \in \mathcal{V}(i)$;
- $J^h(i) \in \mathbb{R}^+$ is the minimal cost to reach the destination D from the node i after the h^{th} reception of an RREP message;
- $p(i)$ is the path travelled by the RREP forwarded by node *i*.

It is assumed that RREQs and RREPs include all information necessary to be uniquely associated to a given flow (i.e., source, destination, and sequence number).

Forward step

The Route Discovery starts whenever the source *S* broadcasts an RREQ with destination *D*. Any intermediate node *i*, receiving the first RREQ, performs the actions described in Algorithm 1. If a given node receives multiple RREQ messages for the same flow (*S*, *D*), it processes only the one with highest sequence number and discards the others.

```
Algorithm 1 - Forward Step
if $i = D$ then
    go to Backward Step
else
    send RREQ (broadcast)
end if
```

Therefore, the main task of this *Forward Step* is that the source, by propagating an RREQ from *S* to *D*, communicates forward to the destination the need to start the *Backward Step* which performs the search of the shortest path. The Forward Step (RREQ propagation) is used only to inform the destination *D* about a request from the source *S*. Hence, only the first RREQ received from *D* is useful. All the other requests for the same flow (*S*,*D*) and with the same (or lower) sequence number are discarded by *D*. Moreover, to limit the Forward Step propagation, if an intermediate node has already received an RREP for the same flow (*S*,*D*) and with the same (or higher) sequence number, any RREQ of the corresponding forward step is discarded.

Backward step

D sends in unicast to all nodes $j \in \mathcal{V}(D)$ an RREP with $J(D) = 0$ and $p(D) = \{D\}$.

Any node *i*, receiving from a node $l \in \mathcal{V}(i)$ an RREP containing $J(l)$ and $p(l)$ carries out the steps of Algorithm 2.

```
Algorithm 2 - Backward Step
if $h = 1$ then
    $J^h(i) \leftarrow T_{i,l} + J(l)$
else
    $J^h(i) \leftarrow \min\left\{T_{i,l} + J(l), J^{h-1}(i)\right\}$
end if
if $J^h(i) \neq J^{h-1}(i)$ or $h = 1$ then
    $H_1^+(i) \leftarrow l$
    if $i \neq S$ then
```

```
            send (unicast) to all $j \in \mathcal{V}(i) \setminus \{p(l)\}$ an RREP
            with $J^h(i)$ and $p(i) = \{i, p(l)\}$
        end if
    end if
```

The main operation, allowing the selection of the minimum cost path, is

$$J^h(i) \leftarrow \min\left\{T_{i,l} + J(l), J^{h-1}(i)\right\}. \tag{3}$$

Notice that a node i sends an RREP to the only nodes $j \in \mathcal{V}(i) \setminus \{p(l)\}$ so as to prevent the formation of cycles in the backward propagation of RREPs.

It is also worth to recall that in most MAC protocols for MANETs, unicast packets are acknowledged by means of a Link Layer Acknowledgment while broadcast packets are not acknowledged by receiving nodes. Hence, sending an RREP in unicast makes the Route Discovery procedure more robust as the information required for the choice of the best path is acknowledged and, in case of loss, retransmitted. Once the source S receives the first route through the reception of an RREP, it starts to send data packets through it. Then, S will evaluate the metric of each successive answer it will receive, which univocally corresponds to a new route. If the metric of the incoming answer is better than the previous one, S switches to the new route. Hence, just like in other reactive protocols, the source S uses the first route received to send initial data because, although suboptimal, it is certainly a good quality path and, in this way, it is possible to reduce network latency.

Route maintenance

Route maintenance is the mechanism by which the protocol reacts to the failure of a link along the path from the source to the destination, trying to recover the communication. To this end, the link diagnosis is carried out by exploiting the Link Layer Acknowledgment messages available in the most common MAC protocols. In reactive routing protocols, which do not have spare routes, whenever a node detects the failure of a link, it forwards an RERR toward the source which, in turn, will restart the Route Discovery procedure.

Due to the highly dynamic nature of MANETs, the activation of a new Route Discovery every time the used path fails, implies an excessive usage of the network resources as well as a drastic reduction of throughput during the recovery phases. A possible approach to reduce the Route Discovery rate is to store multiple paths for the same source-destination pair (possibly ranked in terms of path costs) and replace the best path currently used (whenever it becomes unusable) with the next path in the ranking; see, e.g., [27].

Hereafter, $\mathcal{E}(i)$ will be used to denote, with a slight abuse of notation, just the portion of the routing table of node i relative to the data session (*S*,*D*). Furthermore, let us introduce:

- the node $H^-(i)$ preceding i along the path relative to the data session (*S*,*D*);
- the path $f(i)$ covered by an RTEST forwarded by node i;
- the path $r(j)$ covered by an RTEST_Ack sent by node j;
- the maximum waiting time T allowed for the reception of an RTEST_Ack.

The Route Maintenance process is composed of four algorithms. The main one is *Route Backup Test*, which tests the reliability of backup routes, while the others (*RTEST Management*, *RTEST_Ack Management*, and *RERR Management*) support the management and propagation of the relative messages. These algorithms are described below.

Route backup test

Whenever a node i detects the failure of a link engaged in the data flow (*S*,*D*), the Algorithm 3 is performed.

```
Algorithm 3 - Route Backup Test
delete $e_1(i)$
for $k = 2$ to $|\mathcal{E}(i)|$ do
    send RTEST (unicast) with $f(i) = \{i\}$ to $H_k^+(i)$
    if RTEST_Ack received in time then
    $T < NewTestDelay$
        break
    else
        delete $e_k(i)$
    end if
end for
if $i \neq S$ then
    send (unicast) RERR to $H^-(i)$
else
    go to Forward Step
end if
```

RTEST management

The intermediate node j, receiving an RTEST containing the path $f(i)$, carries out the operations shown in Algorithm 4.

```
Algorithm 4 - RTEST Management
if $j \neq D$ then
    send (unicast) RTEST with $f(j) = \{f(i), j\}$ to $H_1^+(j)$
else
    send (unicast) RTEST_Ack with $r(i) = f(i)$ to last
    element of $f(i)$
end if
```

RTEST_Ack management

The intermediate node j, receiving an RTEST_Ack containing the path $r(i)$, performs the actions described in Algorithm 5.

Algorithm 5 - RTEST_Ack Management

```
if $j \neq$ first element of $r(i)$ then
    delete $j$ from $r(i)$
    send (unicast) RTEST_Ack containing $r(i)$ to the
    new last element of $r(i)$
end if
```

RERR management

The intermediate node j, receiving a RERR, executes the *Route Backup Test* (Algorithm 3).

In order to send an RERR to the preceding node $H^{-}(i)$ along the path, a node i will store the IP address associated to the MAC address of the node from which it received the last packet for that specific flow. Since the information of the preceding node is continuously updated, it is guaranteed that the RERR packet reaches the source also if there have been many link failures along the path from S to D.

It is important to point out that the Backward Step procedure starts when the destination D sends an RREP to its neighboring nodes. In the RREP message, D sets to zero the "metric" field of the message, storing the cost to reach destination. Then, receiving this RREP, a neighbor i of D updates the metric field and sends the updated RREP to its neighbors, except D. It should be noticed that node i has all necessary information to compute the cost-to-go to D because it knows both the mean waiting time $\bar{W}_i$ of its queue and the mean time $\bar{R}_{i,j}$ to transmit a packet to D. Then, each intermediate node receiving an RREP performs the same procedure of node i, until an RREP reaches the source S thus making the data flow start.

Optimality analysis

This section analyzes the optimality properties of the proposed DXFP routing protocol. The following result holds.

Theorem 1. *The DXFP protocol provides a fully distributed solution of the shortest path problem in that among all paths $p \in \mathcal{P}_{S,D}$ connecting the given source $S \in \mathcal{N}$ and destination $D \in \mathcal{N}$, DXFP selects the one, p^*, with minimum cost $C(p^*)$, i.e.,*

$$p^* = \underset{p \in \mathcal{P}_{S,D}}{arg\,min}\ C(p). \tag{4}$$

Before giving a formal proof of the previous theorem, which relies on *dynamic programming* [28], a brief overview on dynamic programming applied to the solution of the shortest-path problem will be provided below.

Dynamic programming approach to shortest-path

In this article, the well-known Bellman's dynamic programming method is recalled and adopted to mathematically prove that our DXFP finds in a distributed way the optimal source-destination path according to the considered metric. Dynamic programming [29] has been introduced by Bellman [28] in 1957 for optimal control of dynamical systems. Hereafter, its application to the shortest path problem will be recalled. Let $c_{i,j}$ denote the cost to pass from node i to node j and let us admit $c_{i,j} = \infty$ to take into account the possibility that i and j are not connected. The objective is to find the shortest path from each node i to a given destination D. It is assumed that there are no cycles with negative cost through the graph, i.e., a path $\{i, j_1, j_2, \ldots, j_k, i\}$, starting and ending with the same node i, cannot have a negative cost.

As the graph $\mathcal{G}$ has $N+1$ nodes, any path (and hence also the optimal one) cannot have more than N hops, the case of exactly N hops corresponding to the situation in which the message crosses all nodes before reaching the destination. Hence, it is convenient to formulate the problem by considering only paths with exactly N connections by allowing also self-connections in node i with associated cost $c_{ii} = 0$. In other words, a path with less than N hops will contain self-transitions in it; clearly each self-connections is associated to a null cost so that its inclusion cannot reduce the cost. Hereafter, let $J_k(i)$ denote the optimal cost-to-go from node i to the destination D in N-k hops, for $k = 0, 1, \ldots, N-1$; clearly $J_0(i)$ is the optimal cost from i to D.

It is, therefore, possible to iteratively formulate the shortest path problem by dynamic programming as follows:

$$J_k(i) = \min_{j \in \mathcal{N}} \left[c_{i,j} + J_{k+1}(j) \right] \tag{5}$$

for $k = 0, 1, \ldots, N-1$ and for all $i \in \mathcal{N}$. The initial condition is given by:

$$J_N(i) = \begin{cases} 0 & , \; i = D \\ \infty & , \; i \neq D \end{cases} \quad \forall i \in \mathcal{N} \tag{6}$$

The optimal strategy at the node i after k hops is therefore to move towards the node j^* minimizing $c_{i,j} + J_{k+1}(j)$ with respect to $j \in \mathcal{N}$. As previously observed, if the algorithm finds self-transitions, it means that the optimal path contains less than N hops.

Proof of DXFP optimality

Proof. DP, being a centralized algorithm, applies (5) to all nodes $i \in \mathcal{N}$ and for any step $k \in \{0, 1, \ldots, N-1\}$. Conversely in DXFP, which is an algorithm fully distributed over the network nodes, (3) is applied from a given node $i \in \mathcal{N}$ only if a reply (RREP) message is received. Hereafter, it is shown how Equation (3) is the distributed implementation of the centralized optimization in (5).

Clearly, for a fixed node *i*, (5) needs only to be applied to the nodes *j* for which $c_{i,j} \neq \infty$.

Hence, (5) becomes

$$J_k(i) = \min_{l \in \mathcal{V}(i) \cup i} \left[c_{i,l} + J_{k+1}(l) \right]. \tag{7}$$

In a similar way, (7) needs only to be applied to the nodes *l* for which $J_{k+1}(l) \neq \infty$, from which (7) becomes

$$J_k(i) = \min_{l \in \mathcal{L}_{k+1}(i) \cup i} \left[c_{i,l} + J_{k+1}(l) \right], \tag{8}$$

where $\mathcal{L}_{k+1}(i) \subseteq \mathcal{V}(i)$ is the set of nodes *l* for which a path to D of length $N-k-1$ exists. Remind that the index *k* is reduced from $N-1$ down to 0. Performing the minimization (8) to get $J_k(i)$ is equivalent to executing the cycle of operations shown in Algorithm 6. □

```
Algorithm 6 - Minimization
$t \leftarrow 0$
$J^t(i) \leftarrow J_{k+1}(i)$
for all $l \in \mathcal{L}_{k+1}(i)$ do
    $t \leftarrow t + 1;$
    $J^t(i) \leftarrow \min\left[c_{i,j} + J(l), J^{t-1}(i)\right]$
end for
$J_k(i) \leftarrow J^{|\mathcal{L}_{k+1}(i)|}(i)$
```

Notice that the main operation of the above cycle is

$$J^t(i) \leftarrow \min\left[c_{i,j} + J(l), J^{t-1}(i)\right] \tag{9}$$

which coincides with the main optimization step (3) of the DXFP algorithm. Assuming that, for DXFP, a node *i* receives in finite time all replies relative to the paths which have the same length $N-k-1$, then the nodes $l \in \mathcal{V}(i)$ from which node *i* receives replies (RREP) form the set $\mathcal{L}_{k+1}(i)$ and it is easy to see that the operations carried out by the DXFP procedure are equivalent to the cycle described above. This, in turn, proves that DXFP is nothing but a distributed implementation of the dynamic programming algorithm.

Routing metric

Let us now consider in more detail the routing metric. As previously mentioned, DXFP, from an operating point of view, is completely independent of the adopted metric provided that this is additive with respect to the link costs, as this guarantees the determination of the *shortest path*.

The metric proposed in this work considers, for a given hop (link), the *average hop crossing time*

$$\overline{T} = \overline{W} + \overline{R} \tag{10}$$

given by the sum of the *average waiting time* $\overline{W}$ at the link origin plus the *average transmission time* $\overline{R}$ over the link, where $\overline{T} \triangleq E[T]$, $\overline{W} \triangleq E[W]$ and $\overline{R} \triangleq E[R]$.

Average queue waiting time $\overline{W}$

As far as the average waiting time $\overline{W}$ is concerned, the well known Little's law [30] yields $\overline{Q} = \overline{\lambda}\, \overline{W}$ where $\overline{Q}$ is the expected number of users in the queue, $\overline{\lambda}$ is the average arrival rate in the queue and $\overline{W}$ is the expected waiting time of an user in the queue.

Hence, $\overline{W}$ can be simply computed by monitoring the queue evolution in time. In fact,

$$\overline{W}(t) = \frac{\sum_{k=1}^{\alpha(t)} W^k}{\alpha(t)} = \frac{\int_0^t Q(\tau)\, d\tau}{\alpha(t)}, \tag{11}$$

where W^k is the waiting time in the queue of the *k*th packet, $\alpha(t)$ is the number of packets arrived at time *t* and $Q(t)$ is the number of queued packets at time *t*. Notice that $Q(t)$ is a piecewise-constant function and its integral can be easily calculated by the sum of the areas of rectangles while the number of arrivals $\alpha(t)$ can be simply counted. The value provided by (11) is just a time average which, assuming ergodicity and considering steady-state behavior ($t \to \infty$), converges to the expected value. In (11), it is necessary to assume that the queue buffer is large enough to avoid overflows; otherwise, an effective number of packets should be considered by taking into account the packet loss probability due to buffer overflow.

Average transmission time $\overline{R}$

The average transmission time $\overline{R}$ depends on the link PER, which is the probability of not receiving a packet correctly

Table 3 DXFP simulation parameters

Parameter	Value	Description
PER_{th}	0.1	Used to evaluate neighbor nodes
$\text{PER}_{\text{penalty}}$	0.2	Used when a HELLO is not received from a neighbor during the set period
HelloPeriod	5 s	Transmitting period of a HELLO message
NewPDDelay	0.1 s	Maximum waiting time for an RREP by the source. It is exponentially (base 2) incremented.
NewTestDelay	0.1 s	Maximum waiting time for an RTEST_Ack by a node which is performing a *Route Backup Test*

Table 4 Performance comparison in a scenario with channels of different quality

	DSR	AODV	DXFP
Mean delay (ms)	2.618	3.834	2.127
Standard deviation (ms)	1.117	53.340	0.554
MAC loss rate (%)	7.36	6.03	1.32
Retransmissions	610	277	77
Signalling	3.66	3.00	3.30
Periodic signalling	✘	✘	50.99

and, supposing channel bi-directionality[c], is assumed to be equal for the reverse link. As stated in section "Local connectivity management", the PER is measured by means of the periodic HELLO messages. Each node A which receives a HELLO message from node B, measures the SINR of the link in order to analytically calculate the associated PER by means of a formula of the type PER $=$ F(SINR) where the function F depends on the modulation, the transmitted energy per bit, the coding, etc. (all the typical parameters of a radio communication channel).

The total transmission time of a packet is equal to the time taken to physically transmit the packet for the number of times required for a correct reception of the packet, i.e., $R = X\frac{L}{B}$ where L is the length (in bits) of the packet at the physical layer, B is the transmission rate (in bit/s), $\frac{L}{B}$ is therefore the nominal time to transmit a packet and X is the number of packet retransmissions. Since X is a random variable with geometric distribution modified by the parameter $p = 1 - \mathrm{PER}$, while L and B are deterministic, the expected value of R is

$$\overline{R} = \mathrm{E}\left[X\frac{L}{B}\right] = \mathrm{E}[\,X]\,\frac{L}{B} = \frac{1}{p}\cdot\frac{L}{B} = \frac{1}{1-\mathrm{PER}}\frac{L}{B}. \quad (12)$$

Notice from (12) that the average transmission time decreases if PER decreases, tending to the nominal transmission time L/B for PER $\rightarrow 0$.

Considerations on the metric $\overline{T}$

Considering nodes i and j, the metric associated to the *link* (i,j) is given by

$$\overline{T}_{i,j} = \overline{W}_i + \overline{R}_{i,j} = \frac{\int_0^t Q_i(\tau)\,d\tau}{\alpha_i(t)} + \frac{1}{1-\mathrm{PER}_{i,j}}\frac{L}{B}, \quad (13)$$

where $\overline{W}_i$ is the average waiting time in the queue of node i; $\overline{R}_{i,j}$ is the average transmission time on the link (i,j); $\mathrm{PER}_{i,j}$ is the PER associated to the link (i,j). It is worth to point out that the DXFP metric provides a better estimate of the network crossing time than ETT mainly for two reasons. First of all it takes into account the waiting time which a packet spends in each node. Second, the analytical calculation of the PER based on the measured SINR of the channel is preferable as it provides a better characterization of the channel quality, for ranking purposes, compared to the simple counting of missed HELLO packets.

Notice that the proposed cross-layer metric (13) takes into account power consumption of the nodes indirectly in the terms $\overline{W}_i$ (the lower is the queue, the lower is the energy required to transmit the queued packets) and $\overline{R}_{i,j}$ (the lower is the link quality, the higher the energy required to correctly transmit the packets). As a matter of fact, energy requirements could also be directly taken into account in the metric, e.g., including an additive penalty inversely proportional to the residual energy of the nodes, provided that such a quantity is available.

In this section, it has been shown how the link metric defined by (13) allows to account for the effects of congestion, node mobility, and channel quality simultaneously. In fact:

- Congestion: the average waiting time $\overline{W}$ allows to penalize congestioned nodes characterized by high values of $\overline{W}$;
- Mobility: averaging PER over a fixed time window, it is possible to avoid choosing *links* associated to mobile nodes, the quality of which is going to get worse because of mobility;
- Channel Quality: PER provides a good measure of the link quality so that the presence of $(1 - PER)^{-1}$ in (13) penalizes links with low channel quality.

Table 5 Mean delay and number of route discoveries in full mobility scenario

	1 m/s Mean delay (ms)	RD	Throughput (%)	2.5 m/s Mean delay (ms)	RD	Throughput (%)	5 m/s Mean delay (ms)	RD	Throughput (%)
DSR	1.548	2	90.69	3.998	32	31.38	1.922	56	9.48
AODV	5.450	8	98.96	3.021	10	98.62	58.752	242	60.00
DXFP	1.663	1	100.00	2.844	3	98.45	4.539	100	59.66

Table 6 Performance comparison in partial mobility scenario

	AODV	DXFP
Mean delay (ms)	5.213	2.839
Standard deviation (ms)	43.445	19.375
Signalling	54.84	5.37
Periodic signalling	✘	203
Route Discovery	52	1
Throughput (%)	97.44	100.00

Performance evaluation

The DXFP protocol was implemented in the OMNeT++ network simulator. The simulations are based on a network composed of terminals implementing a Constant Bit Rate (CBR) application, the UDP protocol as transport layer, IP at network level and IEEE 802.11 g as the only radio technology available for ad hoc networks.

Each simulation was carried out by varying the speed and the congestion degrees of the nodes as well as the quality of the channels. For each simulation, DXFP was compared with DSR and AODV which are nowadays the main routing protocols for wireless ad hoc networks. OMNeT++ has built-in implementations of these two protocols [31] while in [32] the complete DXFP source code can be found. DXFP simulation parameters are shown in Table 3, while for AODV and DSR the settings are those suggested in the relative RFCs.

Channel quality

The first experiment took into account a network with 23 fixed terminals, disposed as a grid (matrix). The terminals can have three different transmitting power levels: 1.60, 0.85, or 0.40 mW.

This case study allows to evaluate a network with channels of different qualities. A robust routing protocol should select the path through the terminals with highest power levels. In all the simulations carried out, it was shown that only DXFP always selects the path with highest channel quality, thus giving the best performance in terms of QoS.

In Table 4, it is shown that the DXFP provides the lowest delivery delays, in terms of mean and variance, since it is characterized by the lowest number of retransmissions and MAC loss rate, where the MAC loss rate is calculated as the number of received packets with error over the total number of packets. In particular, DXFP exhibits a number of retransmissions four times lower than AODV and nine times lower than DSR.

For what concerns the signalling load, defined as the average number of control packets generated per node, it is clear from Table 4 that the DXFP performance is comparable to the ones of DSR and AODV. Upon a total number of 3,760 data packets DXFP shows less than 55 protocol-delayed packets per node (1.46%), while AODV and DSR show less than 4 protocol-delayed packets (0.11%). The exceeding 50 packets of DXFP are due to the periodic signalling required to maintain local connectivity. These control packets are only 11-bit long at network level so that the overall network load is only slightly increased.

Mobility

In this section, the performance of the protocols are compared when the network nodes are characterized by high mobility. The scenario consists of 22 mobile

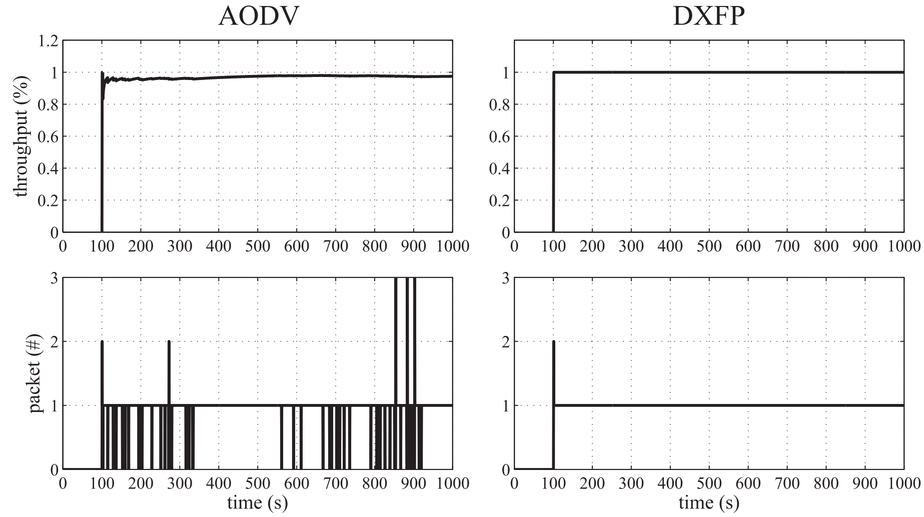

Figure 1 Throughput as a function of time in the high mobility scenario with fixed terminals.

Table 7 Performance comparison in congestioned scenario

	DSR	AODV	DXFP
Mean delay (s)	12.916	1.184	0.065
Standard deviation (s)	7.090	0.216	0.593
Signalling	104.40	6.04	128.24
Periodic signalling	✘	✘	38.96
Throughput (%)	9.49	79.31	95.70

terminals. Each node can have a speed equal to 1, 2.5, or 5 m/s.

As shown in Table 5, the performance of DSR, in terms of percentage throughput, decreases significantly as the speed of the nodes increases. The percentage throughput is defined as the number of packets received by the destination over the number of packets sent by the source. DSR is totally inefficient for node speeds greater than 1 m/s. AODV and DXFP, on the contrary, show higher performance. DXFP, compared to AODV, is able to deliver packets with less route discoveries (see Table 5), because of using backup routes and the selection of more robust paths. Moreover, DXFP always provides the lowest mean delays and, in particular, in the case of node speeds equal to 5 m/s, DXFP provides a mean delay about 14-times smaller than AODV.

The fact that DXFP is able to select more robust paths becomes clearer if we insert a possible path between source and destination through fixed terminals in the scenario where nodes have 5 m/s speed. In this new context (partial mobility scenario), DXFP is able to select the path composed by fixed terminals only at the first path discovery, compared to the 52 of AODV (Table 6). Moreover, the performance of DXFP in terms of throughput is optimal (Table 6), as all the packets (100%) are correctly delivered.

AODV has good performance too, although the analysis of the throughput as a function of time (Figure 1) shows that the paths selected by AODV have continuous losses during the overall simulation. This means that the selected paths are not the best. Moreover, DXFP exhibits the lowest delays (Table 6). In this case, the performance of DSR is significantly worse than the ones of DXFP and AODV, and, therefore, was not included in the comparison.

When a segment of fixed terminals is present in the network, the signalling load of DXFP is 10-times smaller than the AODV's one (Table 6), apart from the periodic component.

Moreover, DXFP runs only one Route Discovery procedure, compared to more than 50 of AODV, showing as AODV does not select the most robust paths.

Congestion

In the last scenario, proposed to compare the performance of the protocols, the network has a zone where the nodes are congested due to high queue loads. In this case, simulation results show that DXFP is able to select the path which avoids such a zone.

The performance in terms of throughput is shown in Table 7. DXFP yields the best performance, allowing the delivery of over 95% correct packets with the lowest delays (Table 7), 20-times less than AODV.

The signalling of DXFP is higher (Table 7) because the network has a high density of terminals and so the answer messages have to be transmitted to a lot of neighboring nodes during the Route Discovery procedure.

The throughput as a function of time is shown in Figure 2. As it can be seen, once DXFP has obtained the access to the channel it does not undergo any loss, while AODV continuously loses packets, showing asymptotically a maximum throughput of approximately 80%.

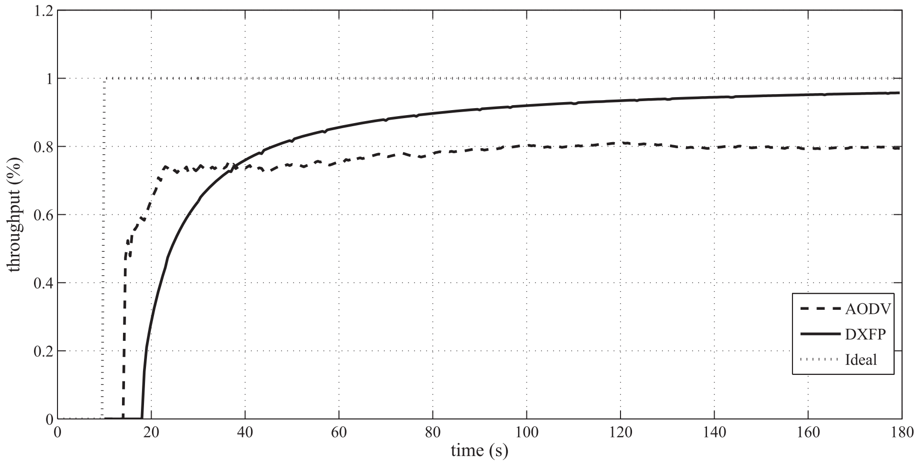

Figure 2 Throughput as a function of time in the scenario with congestion.

Conclusions

In this article, a new routing protocol for wireless ad hoc networks, called Distributed X-Layer Fastest Path (DXFP), has been presented. The proposed protocol was designed to overcome the limitations of most existing routing protocols which typically use network layer metrics, like the number of hops, that are not adequate for emerging wireless ad hoc networks. Full mobility, extreme environments, high time-variability of the channel, high loads of the nodes, are some of the main issues that the future wireless ad hoc networks have to cope with. Maintaining the quality of service in highly critical and time varying scenarios is clearly of extreme importance and a hard task to handle. So, robustness is one of the key features of the design of the new protocol. For these reasons, special emphasis on robustness has been posed in the design of the proposed DXFP protocol.

The specific cross-layer metric adopted for DXFP is able to take into account—in a single quantity to be minimized (the latency of the network)—fundamental issues such as user mobility, node congestion and channel quality thus allowing to deliver packets with less delay and higher throughput.

It was demonstrated analytically that DXFP, despite of its fully distributed nature, is able to select the path with minimum cost.

Simulations extensively carried out by means of OMNeT++ show that the DXFP always performs better than the existing protocols DSR and AODV, especially in scenarios with congestions and mobile nodes in a low channel quality environment. The only price to be paid for the performance enhancement is the higher signalling load, but it is important to point out that the signalling messages consist of a few bits, and therefore they only add a negligible load to the network.

Thanks to all the above-mentioned features, DXFP could be a promising candidate as routing protocol for future wireless ad hoc networks.

Endnotes

[a] In our experiments the jitter was set as a random variable uniformly distributed between 0 and 3 ms.

[b] It should be noticed that this value is a typical routing entry time-to-live value of DSDV protocol or ACTIVE-ROUTE-TIMEOUT value of AODV protocol.

[c] The assumption of bidirectional links is needed in our implementation since the quality indicator $\mathrm{PER}_{i,j}$ of the channel (i, j) between node i and j is measured on the reverse link (j, i) by means of the received HELLO packet. This assumption allows us to implement an effective connectivity management system. Nevertheless, we are aware that in general $\mathrm{PER}_{i,j} \neq \mathrm{PER}_{j,i}$ but in wireless communications channel reciprocity is a rather frequent assumption (see, e.g., [33]).

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

Author details
[1]Department of Electronics and Telecommunications, University of Florence, via Santa Marta 3, 50139, Italy. [2]Department of Systems and Computer Engineering, University of Florence, via Santa Marta 3, 50139, Italy.

Received: 23 November 2011 Accepted: 26 July 2012
Published: 16 August 2012

References

1. E Hossain, RP Palit, P Thulasiraman, Clustering in mobile wireless ad-hoc networks. in *Wireless Communications Systems and Networks*, ed. by Guizani M. (Kluwer, New York, 2004). pp. 383–424
2. N Saeed, MF Abbod, HS Al-Raweshidy, Manet routing protocols taxonomy. in *Future Communication Networks (ICFCN), 2012 International Conference*, 123–128, (April 2012) doi:10.1109/ICFCN.2012.6206854
3. S Corson, J Macker, Mobile ad-hoc network (MANET): routing protocol performance issues and evaluation considerations. in *RFC 2501 Informational*, (January 1999). [http://www.ietf.org/rfc/rfc2501.txt]
4. M Abolhasan, T Wysocki, E Dutkiewicz, A review of routing protocols for mobile ad hoc networks. Ad Hoc Netw. **2**(1), 1–22 (2004). [http://www.sciencedirect.com/science/article/pii/S157087050300043X]
5. E Royer, C-K Toh, A review of current routing protocols for ad hoc mobile wireless networks. Personal Commun. IEEE. **6**(2), 46–55 (1999)
6. D Johnson, Y Hu, D Maltz, The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. in *RFC 4728 Experimental*, (February 2007). [http://www.ietf.org/rfc/rfc4728.txt]
7. C Perkins, E Royer, Ad-hoc on-demand distance vector routing. in *IEEE Workshop on Mobile Computing Systems and Applications 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, 25-26 February 1999, pp. 90–100, doi:10.1109/MCSA.1999.749281
8. C Perkins, E Royer, S Das, Ad hoc on-demand distance vector (AODV) routing. in *RFC 3561 Experimental*, (July 2003). [http://www.ietf.org/rfc/rfc3561.txt]
9. C Sommer, I Dietrich, F Dressler, Simulation of ad hoc routing protocols using OMNeT++: a case study for the DYMO protocol. ACM/Springer Mob. Netw. Appl (MONET) (Special Issue on Simulation Techniques & Tools for Mobile Networking) (2009). doi:10.1007/s11036-009-0174-5
10. S Lee, M Gerla, AODV-BR: backup routing in ad hoc networks. in *IEEE Wireless Communications and Networking Conference 2000. WCNC. 2000 IEEE*, 1311–1316, (vol. 3, September 2000), doi:10.1109/WCNC.2000.904822
11. C Toh, A novel distributed routing protocol to support ad-hoc mobile computing. in *IEEE International Conference on Computers and Communications 1996, Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on*, 480–486, (27–29 March 1996), doi:10.1109/PCCC.1996.493675
12. R Dube, CD Rais, K yeh Wang, SK Tripathi, Signal stability based adaptive routing (SSA) for ad hoc mobile networks. IEEE Personal Commun. **4**, 36–45 (1997)
13. DSJ De Couto, D Agueyo, JBR Morris, A high throughput path metric for multihop wireless routing. in *MobiCom 03*. (San Diego, California, USA, September 2003) pp. 14–19
14. JPR Draves, B Zill, Routing in multi-radio, multi-hop wireless mesh networks. in *MobiCom 04*. (Philadelphia, Pennsylvania, USA, September 26–October 1 2004)
15. Y Yang, J Wang, R Kravets, Designing routing metrics for mesh networks. in *IEEE Workshop on Wireless Mesh Networks (WiMesh)*, (Sept. 2005)
16. CE Perkins, P Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. in *Proceedings of the conference on Communications architectures, protocols and applications, ser. SIGCOMM '94*. (New York: ACM, 234–244, 1994). [http://doi.acm.org/10.1145/190314.190336]
17. S Murthy, JJ Garcia-Luna-Aceves, An efficient routing protocol for wireless networks. Mob. Netw. Appl. **1**(2), 183–197 (1996). [http://dx.doi.org/10.1007/BF01193336]
18. DB Johnson, DA Maltz, Dynamic source routing in ad hoc wireless networks. in *Mobile Computing*, ed. by Imielinski T and Korth HF, (The

Kluwer International Series in Engineering and Computer Science, vol. 353. Springer, New York, 153–181, 1996), doi:10.1007/978-0-585-29603-6-5. [http://dx.doi.org/10.1007/978-0-585-29603-6-5]

19. VD Park, MS Corson, A highly adaptive distributed routing algorithm for mobile wireless networks. in *INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 1405–1413, (vol. 3, April 1997), doi:10.1109/INFCOM.1997.631180
20. ZJ Haas, A new routing protocol for the reconfigurable wireless networks. in *IEEE 6th International Conference on Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, 562–566, (vol. 2, 12–16 October 1997), doi:10.1109/ICUPC.1997.627227
21. C-K Toh, Associativity-based routing for ad hoc mobile networks. Wireless Personal Commun. **4**, 103–139 (1997). doi:10.1023/A:1008812928561
22. R Sivakumar, P Sinha, V Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm. IEEE J. Sel. Areas Commun. **17**(8), 1454–1465 (1999)
23. W Su, M Gerla, IPv6 flow handoff in ad hoc wireless networks using mobility prediction. in *Global Telecommunications Conference, 1999. GLOBECOM '99, vol. 1A, 1999.* 271–275, (1999), doi:10.1109/GLOCOM.1999.831647
24. M SreeRangaRaju, J Mungara, ZRP versus AODV and DSR: a comprehensive study on ZRP performance using QualNet simulator. in *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010.* 1–10, (23–25 September 2010), doi:10.1109/WICOM.2010.5600909
25. MR Souryal, N Moayeri, Channel-adaptive relaying in mobile ad hoc networks with fading. in *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005*. 142–152, (September 2005), doi:10.1109/SAHCN.2005.1557070
26. S Basagni, A Carosi, E Melachrinoudis, C Petrioli, ZM Wang, A new MILP formulation and distributed protocols for wireless sensor networks lifetime maximization. in *IEEE International Conference on Communications, 2006. ICC '06*, 3517–3524, (vol. 8, June 2006), doi:10.1109/ICC.2006.255617
27. A Nasipuri, R Castaneda, SR Das, Performance of multipath routing for on-demand protocols in mobile ad hoc networks. Mob. Netw. Appl. **6**(4), 339–349 (2001)
28. R Bellman, *Dynamic Programming*. (Princeton, Princeton University, 1957)
29. DP Bertsekas, *Dynamic Programming and Optimal Control*. 1st edn, Vol. 1. (Belmont, Athena Scientific, 1995). pp. 1–37, 50–54
30. L Kleinrock, *Queueing Systems*. 1st edn, Vol. 1. (New York, Wiley, 1975). sec. 2.1, pp. 10–19
31. ECA Ariza-Quintana, A Triviño-Cabrera, Implementation of MANET routing protocols on OMNeT++. in *Proc. First Int. Conf. on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools)*. (Marseille, France, 2008) pp. 1–4
32. L Fabbrini, G Giovannetti, DXFP Project Home Page, 2009, April. [http://code.google.com/p/dxfprouting/]
33. K Hugl, K Kalliola, J Laurila, Spatial reciprocity of uplink and downlink radio channels in FDD systems. in *Proc. of COST 273, Technical Report*, (2002)

doi:10.1186/1687-6180-2012-170
Cite this article as: Mucchi *et al.*: **A robust cross-layer metric for routing protocol in mobile wireless ad hoc networks.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:170.