Computer Communications 33 (2010) 269–282

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# Survey on diversity-based routing in wireless mesh networks: Challenges and solutions

Raffaele Bruno [a], Maddalena Nurchis [b,*]

[a] Italian National Research Council (CNR), Institute for Informatics and Telematics (IIT), Via G. Moruzzi 1, 56124 Pisa, Italy
[b] IMT Lucca Institute for Advanced Studies, Piazza S. Ponziano 6, 55100 Lucca, Italy

ARTICLE INFO

*Article history:*
Received 10 June 2009
Received in revised form 6 August 2009
Accepted 12 September 2009
Available online 17 September 2009

*Keywords:*
Wireless mesh networks
Multi-user diversity
Opportunistic routing
Network coding

ABSTRACT

Wireless multi-hop networks often experience severe performance degradations when legacy routing algorithms are employed, because they are not optimized to take advantage of the peculiarities of wireless links. Indeed, the wireless channel is intrinsically a broadcast medium, making a point-to-point link abstraction not suitable. Furthermore, channel conditions may significantly differ both in time and space, making routing over predetermined paths inadequate to adapt the forwarding process to the channel variability. Motivated by these limitations, the research community has started to explore novel routing paradigms and design principles dealing with the wireless diversity as an opportunity rather than a shortcoming. Within this large body of research, opportunistic routing and network coding are emerging as two of the most promising approaches to exploit the intrinsic characteristics of multi-hop wireless networks, such as multi-user diversity. The aim of this survey is to examine how opportunistic forwarding and network coding can achieve performance gains by performing hop-by-hop route construction and by encoding data packets at intermediate nodes. To this end, we present a taxonomy of existing solutions, and we describe their most representative features, benefits and design challenges. We also discuss open issues in this research area, with a special attention to the ones most related to wireless mesh networks.

## 1. Introduction

In multi-hop ad hoc networks, wireless devices cooperate in forwarding traffic between non-adjacent nodes. In this way, multi-hop network paths can be established between any pair of nodes without relying on a pre-existing network infrastructure or dedicated network devices (i.e., routers, switches, servers, etc.) [1]. This distributed networking paradigm is not a novel concept, but it has been proposed more that two decades ago for tactical and military networks. However, the recent advances in wireless technologies, as well as the advent of new mobile devices (e.g., smartphones), have promoted its utilization for a variety of innovative application domains, ranging from sensor networks to vehicular networks and mesh networks [2]. In particular, wireless mesh networks are static ad hoc networks consisting of dedicated nodes (called mesh routers) that form a multi-hop wireless backbone used to share a limited number of fixed Internet connections with a potentially large number of static or nomadic users [3].

Due to their attractive features, such as low cost and ease of deployment [4], as well as the wide range of possible application scenarios, spanning from public safety communications to community-based networks and metro scale municipal networks [5], wireless mesh networks have received increasing attention and stimulated a large body of research activities. Indeed, wireless mesh networks inherit most of the traditional challenges of ad hoc networks [6]. In particular, it is widely recognized that performance and reliability of wireless multi-hop communications significantly depend on the ability of the routing protocol to properly select network paths, given the current network conditions. A natural design approach for dealing with the complexities of the routing problem is to simply apply to the mesh domain the routing paradigms traditionally conceived for wired networks. This design choice implicitly assumes that wireless links are similar to wired links, and that they can be represented as point-to-point connections. For example, most of the routing schemes proposed for generic ad hoc networks (such as DSR [7], AODV [8] and OLSR [9]) select a shortest path between a source and destination pair, and forward each packet through a predetermined sequence of network devices, while assuming that link-layer retransmissions provide a reasonable level of communication reliability. Henceforth, we refer to this category of networking protocols as *legacy routing* solutions. However, wireless links are fundamentally different from wired links. First of all, the wireless channel is an intrinsic *broadcast medium* that has

[*] Corresponding author. Tel.: +39 340 2848776.
*E-mail addresses:* raffaele.bruno@iit.cnr.it (R. Bruno), maddalena.nurchis@imtlucca.it (M. Nurchis).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.09.003

not clearly observable boundaries outside of which nodes are always unable to communicate. This implies that wireless links with intermediate packet loss rates, even higher that 50%, are quite common in typical outdoor mesh environments [10,11]. Furthermore, wireless medium has time-varying and asymmetric propagation properties due to a variety of phenomena, including interference from external signals, wireless propagation impairments and fading [12].

The above considerations on the peculiarities of the wireless communications suggest that, in order to improve the performance of wireless mesh networks, it is necessary to consider link qualities when choosing the best route between a source–destination pair. Indeed, a large body of research has been carried out in this area and different routing metrics have been proposed. The first metric proposed for wireless mesh networking is the ETX [13], which defines the cost of a link between a node and one of its neighbors as the expected number of transmissions that node requires to successfully deliver a packet to its neighbor. However, the implementation of this metric has shown poor performance in multi-rate environments, and an extension, called ETT [14], has been proposed, which defines the link cost as the time a data packet requires to be transmitted successfully. On the other hand, recent work has established that to correctly represent the quality of a link in a multi-hop environment, a routing metric should be able to capture other aspects of the wireless domain, such as the location-dependent nature of the link-layer contention (for instance, see CATT [15] and ETP [16] proposals), or the inter-flow and intra-flow interference (e.g., IRU [17]).

Some of the proposed link-aware routing metrics have been implemented and tested in real network deployments, and experiments have shown that they can achieve significantly higher performance compared to a classical shortest-path routing algorithm. However, all these legacy routing protocols *pre-compute* one or more minimum-cost paths (see, for instance, multi-path schemes described in [18–20]) for each source–destination pair. Experimental evidence [23,25,38] has also proved that using predetermined paths can be ineffective in dealing with unreliable and varying wireless environments. For these reasons, recently researchers have been investigating radically new routing approaches, which exploit the multiple transmission opportunities that the broadcast nature of the wireless medium creates. More precisely, whenever a packet is transmitted, it is simultaneously received by multiple nodes, which may experience significantly different channel conditions. This property is called *multi-user diversity* because it refers to a type of spatial diversity existing across multiple receivers (or users) [21,22]. This intrinsic diversity of the wireless environment is not a drawback per se, but it may cater for new design principles and alternative routing paradigms. Several protocols can be included in this novel class of routing strategies that exploit receptions of the same packet at multiple nodes to increase network performance compared to legacy routing. In this survey, we give a comprehensive review of two of the most promising design approaches: *opportunistic forwarding* and *network coding*.

Opportunistic routing algorithms implement forwarding decisions in a hop-by-hop fashion, and they defer the selection of the next hop for a packet until they have learnt the set of nodes which have actually received that packet [23]. This permits to optimize the selection of the packet forwarder(s) and to discover on the fly the best network path. This strategy clearly departs from the design principles of legacy routing, which assigns a predetermined next hop to each packet. It is also important to note that the term “opportunistic” refers to a wider class of routing algorithms based on the common idea of leveraging any transmission opportunity rather than imposing the packet transmission along a predetermined path. For instance, opportunistic routing is also used in intermittently connected networks [24]. However, in that context, communication opportunities are generated by mobility, which enables pair-wise contacts between nodes. In contrast, in this survey we limit ourselves to static networks, where transmission opportunities rely on the variability of channel conditions and the broadcast nature of the wireless medium.

The second design principle we analyze in this survey is wireless network coding, which allows the network nodes to combine/encode the data packets they receive, so as to compress data information and to increase the innovative content carried into each packet [25]. At the same time, network coding may increase reliability of packet transmissions because each encoded packet mix information about multiple packets, thus increasing the probability that they would reach their destination. It is also useful to note that the boundary between network coding and opportunistic forwarding may be blurred in some cases, when both approaches are jointly used. In these cases, we will prefer the term *hybrid* routing to point out that network coding and opportunistic forwarding are integrated into a unified routing scheme.

The above discussion provides only a brief insight into the reasons of performance gains achievable with opportunistic forwarding and network coding. The objective of this survey is to analyze in a thorough way the various conditions in which these two routing paradigms may provide the most significant performance improvements. To this end, not only we use simple illustrative network scenarios, but we also build a comprehensive classification of the main approaches that can be adopted to implement such strategies. Then, we use our classification as a roadmap to analyze the design challenges that diversity-based routing paradigms need to address, and to describe the features, advantages and disadvantages of the most representative solutions proposed in the literature. Finally, we discuss the open issues in this research area, with a special attention to the ones most related to the wireless mesh scenario.

The rest of this paper is organized as follows. Section 2 describes the three main classes of routing approaches that are the focus of this survey, and it introduces their classification. Section 3 overviews some of the most representative proposals for each class of approaches. Section 4 presents a comparison of the main features of the reviewed solutions. Finally, section 5 draws conclusions and highlights the main open issues in this research area.

## 2. Background and taxonomy

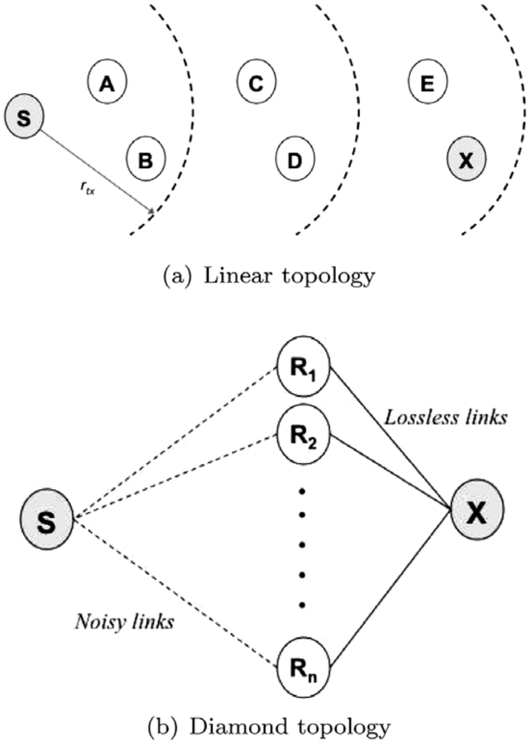
In this section, we overview the general routing approaches that can be adopted to take advantage of opportunistic forwarding and network coding in wireless mesh networks. Specifically, we introduce three main routing categories and several related sub-categories. Then, we describe the representative features, benefits and design challenges of these three classes of routing approaches.

### 2.1. Opportunistic routing

The opportunistic-based routing concept considered in this study is characterized by two main features: (i) any node overhearing a packet transmission is involved in the forwarding process, and (ii) the selection of the next forwarding node(s) is deferred after packet reception [23]. As previously explained, legacy routing algorithms rely on transmitters that select one or more designated next hops before delivering the packets, which implies that each packet must know a priori its next relay(s). However, this design principle borrowed from the routing protocols for wireline networks, does not appear suitable for wireless networks. Indeed, it masks the broadcast property of wireless communications under an artificial point-to-point link abstraction [21]. On the contrary, opportunistic routing fully embraces the broadcast nature of wireless medium because whenever a node is willing to deliver a packet, it performs a broadcast transmission and, then, the nodes that successfully received the packet autonomously select the "best" next forwarder. This allows each packet to dynamically construct the optimal route to reach its intended destination according to the link conditions at the actual time a packet transmission is to be performed.

Intuitively, the main benefit of opportunistic-based routing is to leverage transmission opportunities that unexpectedly reach far nodes, taking advantage of any transmission progress while mitigating the negative impact of failed transmission attempts. Moreover, this strategy allows the destination node to receive packets that have been forwarded by different relays, thus traversing different network paths, fully exploiting the multi-user diversity property. In contrast, legacy routing would require retransmitting any packet that does not reach the next hop for which it was intended, as well as the preliminary construction of one or multiple network paths connecting the source-destination pair.

For a better understanding of the inherent benefits associated to opportunistic forwarding, let us consider the simple topology illustrated in Fig. 1, firstly analyzed in [23]. Let us suppose that the best route from source node *S* to destination node *X* selected by a traditional shortest-path routing is *S*–*B*–*D*–*X*. If a packet sent by *S* is correctly received by node *A* but not node *B*, then it has to be retransmitted by *S* until it reaches the intended next hop *B*. This is the case of a transmission falling unexpectedly short. Another possible situation is a packet sent by *S* that is correctly received by both node *B* and node *C*. Although node C is closer to packet destination than node *B*, it is not allowed to relay the packet. On the contrary, opportunistic routing techniques take advantage of any of these situations to maximize the progress towards the packet destination that each transmission may provide. Moreover, retransmissions are avoided whenever possible, i.e., if there is any alternative forwarding possibility. Thus, by avoiding wasting of network resources through useless transmissions, it is possible to significantly increase the overall network throughput.



(a) Linear topology

(b) Diamond topology

**Fig. 1.** Illustrative topologies clarifying the inherent benefits of opportunistic-based forwarding.

Another interesting benefit of opportunistic routing is the ability of combining many weak physical links into one stronger *virtual* link. As shown in Fig. 1 for a diamond-shaped topology, the sender has a low delivery probability to all its *n* neighbors, while they have a high probability to successfully deliver packets to the destination. In this setting, it may be more advantageous to broadcast a packet in order to increase the probability that at least one of the possible one-hop neighbors correctly receives it. After the packet reception, the "best" node among those that received the packet will be responsible for further forwarding it to the destination. On the other hand, legacy unicast routing strategies would select in advance one of the available neighbors as the exclusive relay for the communications between the source and the destination. This may lead to many retransmissions before the source would be able to successfully deliver a packet to the intended next hop.

At this point, it should be clear that this capability of taking advantage of any transmission opportunity that arises in the network is the principal root of the performance improvements provided with opportunistic forwarding. However, this flexibility comes at the cost of an increased design complexity. In particular, the key technical challenges to be addressed when designing a new opportunistic protocol for wireless mesh networks are the following.

### 2.1.1. How to select the next forwarding node(s)?

In principle, all the network nodes may cooperate in the forwarding process, and be considered as candidate relays. The selection of the next forwarding node(s) among the candidate relays should maximize the transmission benefits, measured in terms of the selected performance metric (e.g., reliability, throughput or the end-to-end delay of the flow). However, the selection of the best forwarding node(s) requires the implementation of a *coordination* process among the candidate relays, which may require explicit exchange of state information. It is intuitive to note that the coordination overheads and complexities increase with the number of candidate relays involved in the coordination process. For these reasons, most of the existing solutions for opportunist-based routing relax the constraint of "pure" opportunism, and assume that the flow source specifies in advance a *subset* of *candidate relays* for each packet or block of packets, which are the only nodes allowed to participate in the forwarding process. Various schemes have been proposed for the selection of candidate relays based on some notion of "*closeness*" of the nodes to the packet destination [23,26,27]. In general, these mechanisms assume that there is an underlying link-state routing protocol that constructs a map of the link qualities in the network, permitting to compute the approximate cost of using a node as forwarder to reach the intended packet destination (according to a given routing metric). Then, the identities of the selected candidate forwarders are listed in the packet header. Thus, whenever a node receives a packet, it first checks if it is in the forwarder list of that packet, in which case it further processes the packet; otherwise it discards it. This strategy keeps the coordination overhead limited, as far as the number of participants is kept small. To further facilitate the selection of the next forwarding node(s), the list of candidate forwarders may be also ordered by assigning to each node a fixed *priority* value to be used during the forwarding process. A common design choice is to derive the node priority from the distance between that node and the packet destination. In the following, we will further elaborate on the role of prioritization in the implementation of an opportunistic forwarding process

### 2.1.2. When relays should forward a packet?

After selecting the candidate forwarders, it is necessary to establish the time at which the packet should be forwarded. In fact, differently from classical routing, where packet forwarding at the next hop node should immediately follow the reception of the packet, in general opportunistic routing solutions introduce forwarding delays. The main reason for this design choice is that imperfect coordination among candidate relays, as well as packet losses, may cause multiple duplicate transmissions of the same packet from multiple nodes. To avoid this unnecessary waste of network resources, a simple solution is to establish a *scheduling* among the candidate relays, and to set different forwarding timers at the selected forwarders. For instance, many schemes use the differentiation/prioritization of the candidate relays to assign fixed and constant forwarding timers to each of the potential forwarders [23,28]. Although the overhead for such a scheduling is high, each node may be aware of its own forwarding time, without the need of a real-time agreement with the other nodes. Then, overhearing of other nodes transmissions or explicit exchange of state information can be used to cancel transmissions of packets already delivered by higher priority nodes. To summarize, scheduling techniques ensure the effective suppression of most duplicate transmissions at the cost of an increase in packet delays and protocol complexity, which are necessary to establish node scheduling without requiring explicit signalling before each transmission. An alternative approach more appropriate for delay-constrained traffic is to implement randomized schedulers, permitting each node to probabilistically decide if continuing forwarding the packet towards the destination [29]. However, in this case, additional mechanisms are needed to control the level of redundancy in packet transmissions, such as rate control mechanisms or methods to limit the maximum number of forwarders for each packet traversing the network.

#### 2.1.3. How to acknowledge packet reception?

Broadcast transmissions are the basis of any opportunistic routing scheme. However, broadcast frames do not implement link-layer acknowledgements. Thus, acknowledgment mechanisms should be introduced, either at the routing or link layer, to provide transmission reliability. Two options can be considered: end-to-end acknowledgements generated by the final destination [23], or hop-by-hop acknowledgements generated by the forwarders [28]. The former reduces overhead but it may lead to higher delay since the forwarding progress depends on acknowledgements generated by the destination. The latter is in contrast with the broadcast transmission principle, even though it contributes to reducing delay and ensuring correct packet reception. Note that for schemes operating on blocks of packets rather than individual packets, acknowledgment information can be easily grouped together, limiting the number of needed acknowledgment messages. Usually, in this case map-based approaches are adopted to implement selective acknowledgment for group of packets [23]. However, these techniques add significant overhead to packet headers, and they require a careful design and tuning.

#### 2.1.4. How to control congestion?

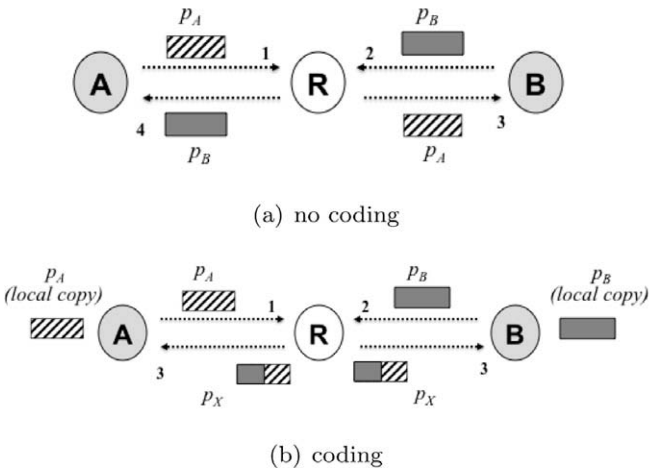
Generally, most of the link-layer technologies transmit broadcast frames without using congestion-aware access methods. Furthermore, duplicate transmissions and multiple flows may exacerbate the collision problem. Thus, the surge of congestion is a critical aspect to be taken into account during the forwarding process. This is still an open issue because most of the current work on opportunistic routing has focused on the design of mechanisms to avoid redundant transmissions, rather than controlling the broadcasting rate. For instance, credit-based schemes have been proposed to control the spreading of packets throughout the network [29], or basic window-based techniques have been used to limit the rate with which new packets are injected in the network [28]. However, opportunistic routing solutions generally forward data packets based only on link conditions and routing metrics that reflect the cost to reach the packet destination. On the other hand, the forwarding process should take into account typical burstiness of data flows and the contention among multiple sessions [28]. This would require up-do-date congestion information to be spread across the network, which arises additional issues.

Keeping in mind the main points discussed above, we believe that a key aspect in the design of opportunistic routing protocols is the strategy adopted for the coordination of the candidate relays during the forwarding process. In our view, this is an appropriate aspect to be considered when classifying existing solutions. Thus, we divide opportunistic routing schemes into *scheduled* and *not-scheduled* algorithms. The first category of solutions identifies a prioritized subset of potential forwarders and specify their scheduling. This list specifies not only the nodes allowed to participate in the forwarding process, but also the order in which they have to transmit, thus their scheduling. On the contrary, not-scheduled schemes allow each node to decide autonomously whether to forward a packet and when to do it. Clearly, a subset of potential forwarders may be still provided in order to simplify the forwarding process, but without establishing in advance a prioritization. The most representative schemes belonging to the above two sub-categories are briefly described in Section 3.1.

### 2.2. Coding-based routing

The basic principle behind network coding is that routers can combine the information to be transmitted so as to deliver multiple data packets through a single transmission. More precisely, let us denote as *native* packets the original non-coded packets that are initially generated by the source node. Then, a *coded* packet is a combination of the native packets, which the destination node can decode to reconstruct the set of initial packets. Potential advantages of network coding were first demonstrated in the pioneering paper by Ahlswede et al. [30], which considers multicast transmissions in wired networks. In this paper it is shown that network utilization can be enhanced if network nodes do not act as classical switches, i.e., routing or replicating packets, but as encoders that mix the information they receive from all the input links and send it to all the output links. The authors demonstrate that network coding allows to achieve the multicast capacity, which is the maximum rate at which a sender can communicate common information to a set of receivers. Moreover, Li et al. [31] show that linear coding is sufficient for the above condition to hold, while Ho et al. [32] demonstrate that this is true also when nodes pick random codes.

Network coding benefits are not confined to multicast transmissions in networks with point-to-point links. Network coding techniques naturally extend to wireless networks by taking advantage of the broadcast nature of the wireless channel [33]. In wireless networks, nodes can overhear neighbors' transmissions. Hence, each node may be able to collect many packets to code together, thus increasing the efficiency of the forwarding process in many cases. To better explain the performance gains obtained by employing network coding techniques in the context of wireless networks, in the following we illustrate a simple coding example. To this end, let us consider the chain topology depicted in Fig. 2, where node $A$ wants to send packet $p_A$ to node $B$, and node $B$ wants to send packet $p_B$ to node $A$. In this case, intermediate node $R$ must forward both packets received by node $A$ and node $B$ because they can not directly communicate to each other. Thus, with legacy routing, four transmissions are needed in order to deliver one packet to both destinations. On the other hand, as shown in Fig. 2, network coding allows node $R$ to broadcast a single coded packet, say $p_X$, generated by applying the *XOR* operator to the native packets (i.e., $p_X = p_A \oplus p_B$). Then, node $A$ can easily recover packet $p_B$ since it locally stores a copy of packet $p_A$ and $p_A \oplus p_X = p_B$,



(a) no coding

(b) coding

**Fig. 2.** A simple example of achievable coding gain in wireless environments.

while node *B* can reconstruct packet $p_A$ with analogous operations. In this way, three transmissions are required instead of four, with a 33% improvement of network capacity. Note that the overall coding gain depends on both the network topology and the traffic patterns. For instance, a similar reasoning can be applied to a cross-based topology, in which four flows intersect the central node *R*. Then, node *R* can combine the four packets received by its neighbors into one coded packet. Assuming overhearing among the neighboring nodes, four nodes are able to exchange packets in five total transmissions instead of eight, with a 60% improvement of network capacity. Hence, coding gain is more significant in larger networks, where more *coding opportunities* arise. In general, a coding opportunity may be defined as the possibility of creating a coded packet that can be successfully decoded from the intended destinations of the native packets.

Another important feature of network coding is the ability of providing reliability with low complexity, which is particularly relevant in lossy environments. Indeed, packet loss is a not negligible issue in wireless networks, and retransmissions are the simplest method to ensure reliability. However, simplicity comes at the cost of increased congestion and higher collision probability due to retransmitted packets. In this context, network coding offers a more convenient alternative to retransmissions of original packets by spreading information about several packets through their combination. This leads to a certain level of reliability in a more efficient manner. Even in case of packet loss, nodes may still be able to recover the original packets without asking for any retransmission.

Although the basic principles of network coding are intuitive, there are critical aspects to be considered when developing a coding-based routing solution, which can be summarized as follows.

### 2.2.1. Which packets should be coded together?

To maximize coding gains it is necessary that each node receiving coded packets is able to recover all the original native packets. This goal can be achieved by imposing some constraints on coding decisions taken by every node. The coding process regards *which* packets must be coded together and *how many* coded packets must be sent (i.e., how much *redundancy* must be guaranteed). A primary basic distinction is between intra-flow and inter-flow coding styles. If a network coding technique is intra-flow, then each node must encode packets together only if they belong to the same flow [34]. Thus, packet selection is mainly driven by the flow membership. On the other hand, if a node can select packets intended for different next hops, the choice is more complex. Whenever a node is willing to send data, it must select the subset of native packets that maximize a certain metric, which should reflect the possibility for each neighbor to recover native packets [25,35]. As explained later, this is strictly related to the encoding scheme used to code packets together. Finally, several techniques have been proposed to improve encoding efficiency. For instance, a common approach is to group packets into blocks and to permit only the coding of packets belonging to the same block. This solution aims to find a trade-off between network coding benefits and complexity.

### 2.2.2. How to code packets together?

Computational complexity is a crucial issue in network coding, and selection of coding techniques must consider the impact both in terms of encoding and decoding complexity, and in terms of minimum number of coded packets needed to recover the original flow. In the example presented above, coding is performed through *XOR* operations, which are easy to implement. However, the most frequently used approach for encoding packets is through *random linear codes* [31,32]. Specifically, given a set of $k$ native packets $p_1,\dots,p_n$, the coded packet $p^{'}$ can be created as $p' = \sum_{j=1}^{k} c_j p_j$, where $c_j$ are random coefficients extracted from a certain finite field $\mathbb{F}_q$.[1] Random linear codes have some nice properties. First of all, checking for independence between coded packets requires only simple matrix algebra, and decoding can be done inverting the matrix of coding vectors. Furthermore, a linear combination of coded packets is also a linear combination of the corresponding native packets, which greatly simplifies the re-encoding process at intermediate forwarders. Several theoretical studies on properties of random linear coding have been conducted, demonstrating the potentiality of this technique, both in lossless and in lossy environments [36].

### 2.2.3. When coded packets should be generated?

Many factors affect the selection of the time at which coded packets should be generated. In general, a coded packet should be created only when there is a coding opportunity, i.e., the node has enough packets to code together. However, a node may have packets to send but no coding opportunities, thus it may decide either to forward native packets or to further delay transmissions waiting for receiving additional packets. Clearly, this design choice represents a trade-off between delay and achievable coding gain. Note that also buffer constraints must be taken into account to decide how long packets useful for encoding should be stored by each node. Furthermore, the coding algorithm should ensure that intermediate nodes have received enough coded packets to decode their corresponding native packets [25,37]. For instance, intermediate nodes may want to reconstruct native packets to refresh the packet stream by replacing coding coefficients and re-encoding incoming packets. In addition, coded packets may include packets from multiple flows, and intermediate nodes may want to decode incoming packets to avoid that data is forwarded to areas where there are no interested receivers. Thus, the broadcast rate of coded packets should be adjusted to ensure that the decoding probability is sufficiently high not only at intended destinations but also at intermediate forwarders. Finally, transmissions of coded packets can also be driven by a trade-off between the desired level of data redundancy and the achievable coding gain [35]. Specifically, coding is generally used to minimize the total number of transmissions needed to carry packets across each wireless hop. However, in case of high loss rates, it may be desirable to increase redundancy by injecting more coded packets in the network, so as to ensure that the next hop forwarders receive enough packets to be used during

[1] A finite field $\mathbb{F}_q$, or Galois Field $GF(q)$, contains a finite number $q$ of elements, where $q = p^n$, $p$ is a prime number and $n$ is a positive integer. In general, for network coding purposes $p = 2$.

the decoding process, even at the cost of increasing the number of transmissions required to communicate the same information.

Several solutions exist to deal with the various issues described above, and to fully exploit the coding benefits. We believe that the key characteristic pertinent to network coding that can be used to discriminate between coding-based routing schemes is the set of rules employed to decide which packets code together. To this end, the distinction between the two complementary approaches of *intra-flow* and *inter-flow* network coding, represents an essential principle for the network coding classification. The most representative schemes belonging to these two sub-categories are briefly described in Section 3.2.

### 2.3. Hybrid routing

From the above discussion we can conclude that opportunistic forwarding and network coding are two complementary means of taking advantage of the broadcast nature of wireless channel, as well as to exploit the multi-user diversity of typical wireless environments. It is also intuitive to anticipate that coupling these two approaches into a *hybrid* paradigm may permit to obtain significant improvements, originated by joining advantages of both techniques. Moreover, this coupling can offer an implicit solution to some limitations of the two paradigms. For instance, one of the main issues of opportunistic routing is the scheduling overhead for node coordination. Classical opportunistic forwarding deal with this issue by introducing node prioritization and forwarding delays, or exchanging state information between candidate relays. In contrast, network coding may provide an elegant method to partially eliminate this complexity. In principle, nodes do not need to know exactly which packets are stored by each neighbor and which packets are sent by the other forwarders. Indeed, if $n$ native packets have to be sent, then any set of $n$ different coded packets is sufficient to recover the original set. Hence, every forwarder may autonomously generate its own coded packets, since any of them contains information about several native packets, and it may contribute to the flow progress towards the destination. Clearly, this solution carries also some network coding issues to the opportunistic setting. In theory, each forwarder can create and broadcast coded packets, but this may lead to a high number of unnecessary transmissions. A possible solution is to allow node to code and forward only *innovative* packets. However, the formulation of the innovative property depends on the specific scheme, and we describe it in details in Section 3.3.

In order to better clarify the advantage of the hybrid paradigm, let us explain it with an example. In the chain topology shown in Fig. 3 source node $A$ should send two packets, $p_1$ and $p_2$, to the destination node $B$. Let us assume that a routing protocol has selected the two-hop path $A$–$R$–$B$ because the direct communication $A$–$B$ is too "weak", e.g., affected by a high loss probability or using a slow transmission rate. Thus, packets $p_1$ and $p_2$ will be sent to intermediate node $R$, which should further relay them to the intended destination. Now, let us assume that relay $R$ receives both packets $p_1$ and $p_2$ correctly, and that node $B$ can directly overhear at least packet $p_1$ from node $A$. With legacy routing $p_1$ correct reception at node $B$ is useless, because the packet is discarded. In contrast, opportunistic routing permits to take advantage of this unexpected reception, reducing the number of packets that relay $R$ must forward to node $B$. However, if only the opportunistic paradigm is employed, node $R$ would first need to communicate with $B$ to know which packet(s) it misses and, then, send it(them). On the other hand, by exploiting coding techniques, $R$ can broadcast linear combinations of the two packets, allowing $B$ to recover the missing packets potentially requiring a smaller number of transmissions and without exchanging additional control messages. For instance, if $R$ broadcasts $p_1 \oplus p_2$, then only one transmission from $R$ is sufficient to successfully complete data exchange. Clearly, after recovering the missing packet, $B$ has to send an ACK to notify $R$ of its successful reception, as it is required also in the legacy case. In summary, this basic case illustrates the main principle of a hybrid scheme. Nodes perform *broadcast* transmissions of *coded* packets without having a designated next hop. Clearly, specific details depend on the chosen approach, but the above considerations hold in general.

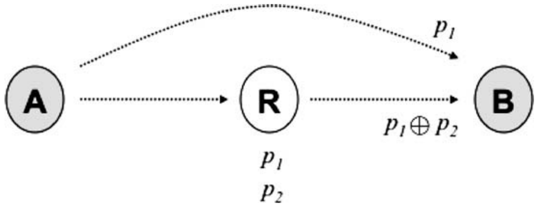


**Fig. 3.** An example of potential advantages of hybrid schemes.

When designing a hybrid scheme, most of the technical issues are inherited from the individual techniques. Below we focalize only on the critical issues that are specifically related to the hybrid paradigm.

#### 2.3.1. Which packets should be coded together?

Although this issue has been already discussed in the network coding context, it should be revisited by taking into account the peculiarities of opportunistic forwarding. More precisely, using legacy routing each packet transmission has a designated next hop node. In this case, the coding process may easily discover, through probabilistic considerations and overhearing or explicit signalling, which are the packets available at the different destination nodes. Thus, each relay node can locally decide which coded and native packets should be transmitted to maximize some metric (e.g., throughput, packet delivery rate, etc.) across all the intended next hop nodes. In contrast, when the path is constructed hop-by-hop at the packet recipients, the concept of designated next hop is not valid anymore, thus different coding strategies should be employed. In general, when opportunistic routing and network coding are used together, the aim should be to spread coded packets across the network in order to increase coding opportunities rather than sending (native) packets to a specific subset of nodes [38]. To simplify this coding decision process, intra-flow coding is commonly used in existing hybrid routing schemes.

#### 2.3.2. When a node should stop sending packets?

From the point of view of opportunistic routing, a relay node should keep transmitting a packet until it is sure that at least one node closer to the destination has received it. On the other hand, network coding imposes a more demanding constraint because a minimum number of independent coded packets must be received at the destination node for ensuring correct decoding [38]. In principle, a forwarder may keep transmitting stored packets to increase levels of redundancy and improve successful decoding probability at the destination node. However, an efficient stopping rule is needed to achieve those goals while ensuring tolerable delays and overheads limited. To facilitate the protocol design, usually the coding process operates on *blocks* of packets.[2] Thus, the stopping rule reduces to the policy used to stop processing a certain group of packets and start with the next block. Two general approaches can be identified. The first idea is that the destination directly sends a message to the sender when it receives enough packets from a certain block, so that the sender starts processing a new

[2] The block here is intended as a group of consecutive packets belonging to a certain flow. In practice, each coding approach has its own "grouping" policy.

block and informing all the other nodes to stop coding packets of the previous block [38]. Thus, the destination drives coding decisions at each hop, according to its decoding goals. An alternative strategy entails that every node is in charge of ensuring decoding at its own neighborhood. The key point is that the source and all the intermediate nodes move from one block to the next one based on the current situation of their neighbors, thus each node contributes to ensure the local complete and correct reception of a block, which in turn guarantees correct decoding at the destination itself [39]. The details of different solutions and possible enhancements are discussed in Section 3.3.

The classification of hybrid-based routing solutions is slightly trickier than in the other two routing categories previously presented. Clearly, key features are derived from the basic "components", since a hybrid solution is intrinsically opportunistic and coding-based. Thus, in principle we could think to derive a taxonomy for this class of solutions from the same criteria used for its building blocks. From the opportunistic side, the categorization in *scheduled* and *not-scheduled* approaches loses its validity in the hybrid case, since network coding partially eliminates the need of node scheduling. Thus, as for coding-based approaches, we could opt for a classification based on flow-membership constraints in the native packets selection process. However, as stated above, inter-flow coding is perhaps harder in a hybrid context, where the notion of predetermined next hop node is not valid. In our vision, hybrid-based routing can be better classified based on the used stopping rule. Specifically, we categorize as *destination-based* the hybrid schemes that rely on the destination to drive coding decisions at each hop. In contrast, we denote as *neighborhood-based* the hybrid schemes that move the focus of coding decision to the neighborhood rather than the destination. Hence, the key difference between *destination-based* and *neighborhood-based* coding is that the former requires notifications from the destination about its current situation, while the latter rely on local context discovered through overhearing and, in some cases, explicit signalling. The most representative schemes belonging to these two sub-categories are briefly described in Section 3.3.

## 3. Overview of representative solutions

In this section, for each of the three routing categories analyzed in Section 2, we overview the design choices and operations of some of the most representative schemes. For the sake of clarity, Fig. 4 illustrates the proposed taxonomy and lists the solutions that will be analyzed more in details in the following.

### 3.1. Opportunistic routing

As reported in Fig. 4, opportunistic-based routing solutions are divided into two sub-categories: *scheduled* approaches and *not-scheduled* approaches. The following two sections outline the most relevant schemes proposed for both approaches.

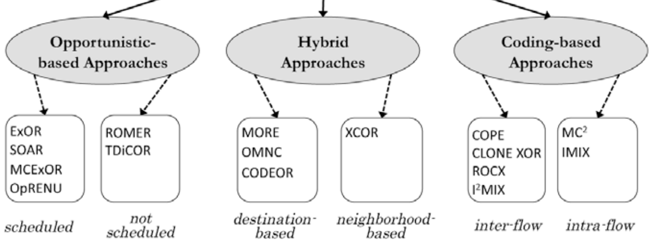


**Fig. 4.** Taxonomy of routing approaches for wireless mesh networks taking advantage of multi-user diversity.

#### 3.1.1. Scheduled schemes

Schedule-based opportunistic routing originates from the seminal paper from Biswas and Morris, who have proposed the *Extremely Opportunistic Routing* protocol (*ExOR*) [23]. To reduce the coordination overhead between candidate relays, in ExOR the packets to be transmitted are grouped into batches according to their destination node. For each packet of the same batch, the source node selects a subset of optimal candidate forwarders, which are prioritized by closeness to the destination. The closeness property of a node is evaluated employing the ETX metric [13], i.e., estimating the average number of retransmissions needed to reach the destination from that node along the lowest-ETX path. Thus, the implicit assumption underlying ExOR design is that a link state routing protocol is also running in parallel to the opportunistic routing to efficiently collect links' delivery probabilities.

The list of selected forwarders, ordered by node priority, is added to the header of each packet broadcasted by the source. Hence, each node receiving a packet knows whether it has to participate in the forwarding process or not, and its position in the forwarding schedule. Due to inter-node loss rates, each candidate forwarder will successfully decode only portions, called *fragments*, of the packet batch it has received. In order to distribute information on which fragments each forwarder has received and rebroadcasted, each packet also contain a *batch map*. For each packet in the batch, this map indicates the highest-priority node known to have received a copy of that packet [23]. Then, as the packet progresses towards the destination the batch map contained in the packet is used to update the local batch maps stored in the receiving nodes, which list the IDs of the node closest to the destination known to have transmitted that packet. A forwarder is allowed to broadcast only received packets that its local batch map indicates have not been forwarded by any other higher priority node. Moreover, to avoid simultaneous or duplicated transmissions by different nodes, whenever a forwarder receives a packet it sets a timer, called *forwarding timer*. This timer is an estimate of the time that would be necessary to higher priority nodes to transmit the remaining packets in the batch. Then, only when the node's forwarding timer elapses, it can rebroadcast the packets it has received, and which its local batch map does not indicate as received by higher priority nodes.

It is now evident that the batch maps contained in the headers of transmitted packets play a crucial role in ExOR. On the one hand, they are used as a sort of gossip mechanism to disseminate reception information from higher priority nodes to lower priority nodes. On the other hand, batch maps are used also for the acknowledgement process. More precisely, when the destination receives a new packet it sends back to the source its batch map using a legacy unicast routing. In this way, the source knows when the destination has received most of the current batch, so that it can pass to transmit a new batch. However, ExOR only guarantees to transmit 90% of a batch using opportunistic forwarding, while the remaining packets are sent with legacy unicast routing.

The strict schedule ExOR establishes between candidate forwarders is somehow equivalent to a slotted polling system. As shown in [23] through experiments in an urban mesh trial, the ExOR scheduling is effective in ensuring that each packet is retransmitted a minimal number of times, and in limiting the probability that multiple forwarders rebroadcast the same packet. However, the simplicity of this scheme comes at the cost of its inefficiency. First of all, the largest the set of candidate forwarders and the longer is the cycle of the scheduler. Furthermore, since candidate forwarders can be out of each other radio range, ExOR fixes a minimum value for the forwarding timers. Thus, even if a node has no batch fragments to transmit, the scheduler blocks the lower priority nodes. Finally, the scheduling duration is not dependent on the number of packets to transmit. Thus, the scheduling overhead would be excessive for a relatively small number of packets. This is the reason why in the original ExOR design the last 10% of packets in a batch are routed to the destination using legacy link-state routing. In addition, ExOR mandates the use of large batches of the order of tens of packets. This also implies that ExOR works well only with persistent flows, which always generate the minimum number of packets needed to fill a batch. In addition, the use of ExOR-style scheduling with multiple concurrent flows, which can induce conflicting forwarding timers, is not specified in [23]. Nevertheless, ExOR approach has raised very high interest in the research community, where many groups started to work on solutions based on the same principle: to abandon the concept of predetermined paths in favor of paths constructed hop-by-hop.

An interesting variant of the ExOR solution is the *Simple Opportunistic Adaptive Routing* protocol (*SOAR*) [28]. Similarly to ExOR, SOAR employs a scheduling scheme relying on priority-based forwarding timers to avoid duplicate and simultaneous transmissions by different nodes, being ETX the metric used to estimate node's closeness to the destination, and its related priority. However, the strategy used by SOAR to establish the schedule among the candidate forwarders is radically different from ExOR. First of all, SOAR does not use batch maps to explicit signal among candidate forwarders on packets' reception status, but it employs overhearing to coordinate forwarders' transmissions. More precisely, whenever a node overhears a transmission from higher priority nodes, it will cancel its forwarding timer and remove that packet from its queue, thus avoiding duplicate transmissions. To ensure that the candidate forwarders are close enough to overhear each other with a high probability, SOAR avoids diverging paths and uses only network paths in close proximity to the shortest route between the source and destination. Moreover, SOAR abandons the use of packet batches and operates on individual packets. Since packet are not organized in batches, also the computation of forwarding timers is simpler in SOAR than in ExOR because the former can use constant timers proportional to the node priority, while the latter employed variable timers whose duration depends on the number of packets buffered in higher priority nodes and the receiving data rate.

Another aspect that differentiates SOAR from ExOR is the use of hop-by-hop retransmissions, which are driven by network-layer ACKs generated by the highest priority forwarder that received the packet. However, to increase the reliability of the forwarding process and minimize useless retransmissions, SOAR uses a combination of various ACK mechanisms, including selective ACK to acknowledge all recently received packets, as well as piggyback ACKs and ACK compression to reduce ACK overhead. Finally, although SOAR does not employ packet batches, it allows each forwarder to transmit a new packet even if there are other outstanding unacknowledged packets. To this end, SOAR uses a classical sliding-window protocol to control the maximum number of outstanding data packets. Note that in [28] it is proposed to use a small window (only three packets) to limit the transmission delays.

The *Multi-Channel ExOR* protocol (*MCExOR)* [40] applies opportunistic routing to multi-channel wireless networks. Similarly to ExOR, this protocol uses a prioritized set of candidate forwarders. However, the multi-channel extension requires the computation of one set for each radio channel. To decouple routing from channel assignment, in MCExOR it is assumed that the assignment of channels to nodes is carried out independently of packets flows. In this case, each node is simply characterized by its *home channel*, i.e., the channel it is operating on.

Since MCExOR leverages the ExOR design principles, the most important task it has to perform is the selection of the candidate forwarder set. However, while ExOR employs a simple and centralized selection rule, i.e., a candidate forwarder is *any* node in the network that would be able to transmit at least 10% of the packets in a batch, MCExOR defines a more sophisticated and localized heuristic. First of all, in MCExOR a source of a data flow selects among its neighbors the nodes that have an expected cost of delivering a packet along the lowest-ETX path to the destination lower than its own. Then, the selected neighbors are grouped according to their home channels. Finally, among each group, all the possible combinations of candidate forwarders are considered. Each of these subsets has an associated cost, which depends on the average number of transmissions that would be needed to reach the destination in case that set of forwarders would be used.[3] However, the optimal set is not the one that simply has the minimum metric, but the one that also minimizes self-interference, which is caused by the use of the same radio channel at each hop along the path. Thus, a multi-channel environment introduces a further dimension in the routing process because the goal now is to reduce not only the number of data transmissions, but also interference among packets belonging to the same flow.

Once the source has selected its set of best forwarders, it broadcasts the packet, whose header contains the forwarder list. To select which candidate forwarder must carry on with packet transmission, MCExOR relies on slotted link-layer acknowledgments. Specifically, candidate forwarders that received the packet send their ACK in order of decreasing priority, all separated by a delay of *SIFS*. In case of ACK missing, other nodes willing to send data may sense the medium idle for a *DIFS* period, which allows them to start a new transmission. Thus, in order to avoid collisions, the mechanism is defined as a compressed slotted acknowledgement, where each node sends prematurely its own ACK whenever it detects a missing acknowledgment from the previous (higher priority) forwarder. Obviously, the highest-priority candidate forwarder to have sent the ACK is elected as the next relay node. Then, the new forwarding node must identify its optimal set of candidate forwarders applying the same algorithm used by the source. Thus, differently from ExOR, in MCExOR the candidate forwarder list is recomputed after each transmission by the node that has been selected to rebroadcast the received packet. It is intuitive to note that this is possible only if MCExOR operates on individual packets, as SOAR, and not on packet batches.

A different approach from the ones presented so far is introduced in [41]. The basic idea is that nodes closer to the destination are not the only valid potential forwarders, as in many other opportunistic solutions, and an utility framework is proposed to estimate the benefit of the successful delivery of a packet. More precisely, a benefit value is attributed to each packet originated at source *S* and heading to destination *D*. Then, an expected *utility* value can be associated to the packet delivery on each multi-hop network path between the source and destination, computed as the packet benefit minus the path cost. It is also possible to compute the *residual expected network utility* (*RENU*) for each node on the network path, which represents the utility to use that node as relay for packet destination *D*. In other words, RENU parameter reflects the node's closeness to the destination in terms of utility. Then, the optimal route that maximizes the utility will depend not only on the topology, but also on the chosen benefit value, which is a unique property of utility-based routing [42]. In case of opportunistic routing the utility metric is reformulated as *OpRENU* [41] to take into account that there are multiple candidate forwarders and not a single next hop. In other words, the utility of a node depends not only on the utility of that node selected as relay

[3] To compute the optimal set in a more efficient way, MCExOR assumes that only the first hop in path is opportunistic.

to reach the destination, but also on the utility of all the nodes belonging to the same relay set. Due to this mutual dependency, determining the optimal relay set that maximizes the network utility is a complex problem that requires an exhaustive analysis of all paths from source to destination. Thus, in [41] an heuristic solution is proposed to select relays and to determine priorities among them. With this heuristic, the relay selection procedure operates only on a restricted subset of the possible paths, according to the order obtained through RENU values.

#### 3.1.2. Not-scheduled schemes

In contrast with the approaches described above, the family of not-scheduled schemes mitigate the design complexity of opportunistic routing by avoiding strict scheduling among candidate relays. One of the first examples of such approach is the *ROMER* protocol [29], which introduces a credit-based forwarding scheme. More precisely, when a packet is generated by the source node, it receives an amount of credits that it can spend during the forwarding process. The assigned credits are equal to the sum of the minimum cost from the source to the destination (i.e., the shortest path cost), plus extra credits necessary to expand the path while being forwarded. Then, whenever a node receives a packet, it decides if it is an appropriate forwarder according to the remaining credits of the packet and the cost of the shortest path from itself to the destination. To some extent, this extra credits reflects the level of resilience demanded to the forwarding process. This strategy permits to create a forwarding mesh on-the-fly centered around the minimum-cost path from the source node to the destination node. Conceptually, this approach is similar to the one adopted in SOAR, which uses the ETX metric to control the width of the forwarding mesh, while ROMER credit mechanism provides a finer control on a per-packet basis. With this approach, a critical aspect is how to distribute packet credits along multiple candidate forwarders. For instance, more credits can be assigned to nodes closer to the source than to the destination. This has the effect of permitting a faster initial expansion of the forwarding mesh, and to remain closer to the shortest path when approaching the destination. However, other strategies taking into account interference or load distributions are equally valid design choices. To select the optimal forwarders among the nodes that received a packet, as well as to reduce the number of duplicate transmissions, ROMER employs a *probabilistic* strategy: the forwarding probability is set proportional to the link's current transmission rate and to the desired level of packet redundancy, so as to assign a higher forwarding probability to intermediate nodes that use higher transmission rates. This probability-based forwarding scheme permits to exploit the best-rate links that are dynamically identified by the rate adaptation algorithms [43], but it is also beneficial to improve the routing resiliency to randomized packet losses, and to quickly adapt to the varying conditions of the wireless links.

A different approach for minimizing the coordination overhead is proposed in [44], which presents the *Transmit Diversity-based Cooperative Opportunistic Routing* scheme (*TDiCOR*). Basically, TDiCOR employs passive listening for the forwarder selection: all the selected candidate forwarders that successfully received a packet try to forward it by contending simultaneously for the medium access. Then, the first candidate relay that gains access to the channel assumes the *forwarding responsibility*, which is the responsibility of continuing the packet forwarding to the destination, and retransmitting the packet if necessary. The other candidate forwarders overhearing this transmission will cancel their own. Thus, forwarder selection does not rely on any form of prioritization or cost-based scheduling, but it leverages only on random medium access. However, since the forwarders selection requires packet overhearing among candidate nodes, it is important that potential forwarders have high-quality links between each other. To this end, TDiCOR uses the ETX metric to evaluate the closeness between nodes, which is somehow similar to SOAR [28]. Finally, to improve the reliability of frame transmissions, TDiCOR exploits the *transmit diversity* property. Specifically, after packet reception all the selected candidate forwarders transmit their link-layer ACK frames simultaneously, so that the sender receives multiple identical copies of the same ACK frame. This cooperative acknowledgment has the effect of increasing received signal strength and mitigating fading effects, while the next hop selection comes automatically from the medium access protocol without requiring any additional control traffic nor complex scheduling. However, the feasibility of transmission diversity depends on the level of timing accuracy. Finally, in TDiCOR transmit diversity is used not only for cooperative acknowledgements but also for cooperative data transmissions. Specifically, if the RTS/CTS option is active, when a node overhears a CTS frame it can check its interface queue to search if it has an identical frame to transmit (i.e., identical source address, sequence number, candidate set, and retry bit). In this case, that node can act as cooperative relay and it can transmit the frame simultaneously with the node that has sent the initial RTS frame.

### 3.2. Coding-based routing

As reported in Fig. 2, coding-based routing solutions are divided into two sub-categories: *inter-flow coding* approaches and *intra-flow coding* approaches. The following two sections outline the most relevant schemes proposed for both approaches.

#### 3.2.1. Inter-flow coding schemes

A fundamental approach proposed for inter-flow network coding in mesh networks is the *COPE* protocol [25]. COPE relies on a legacy routing protocol to select a minimum-cost path (according to some metric) between nodes. In this sense, COPE is not an opportunistic routing protocol as defined in this study because the sequence of next hops that each packet, encoded or not, should follow is fixed and known a priori. However, COPE also allows intermediate forwarders to mix packets from multiple unicast flows. To this end, each node implements opportunistic listening to overhear packets that are not intended to it but that can be used for efficient coding. The overheard packets are stored in a local buffer for a limited time period. Then, whenever the MAC protocol grants a node the permission to transmit, this node selects from its local buffers the packets to code together in such a way that all next hops of encoded packets will be able to reconstruct their corresponding native packets. More precisely, each node combines, using the *XOR* operator, $n$ distinct packets headed for $n$ different next hop relays only if it is sure that every intended next hop has already all the $n-1$ packets required to decode the native packets encoded together.

It is now clear that the critical aspect of the COPE design is to ensure that each node can learn the state of neighbors' buffers to know which packets they have. In practice, neighbors' buffer information is obtained through neighbors' notifications and "guessing". Specifically, each node broadcasts *reception reports* listing the packets it has stored. To some extent, reception reports are equivalent to the batch maps used in ExOR. The problem with the reception reports is that these messages can be lost or, even more important, arrive too late for the coding purposes. For these reasons, each node may anticipate if a particular packet has been received by a certain neighbor based on the delivery probability between that neighbor and the packet previous hop, i.e., the node from which it has received that packet. Furthermore, the packet coding algorithm in COPE is based on the principle of never delaying packets whenever the wireless medium is available. Thus, the node transmits a combination of packets if a coding opportunity exists, giving preference to packets of the same length, otherwise it simply forwards the native packet, if any, at the head of its transmission queue.

An interesting design choice of COPE is the use of *pseudo-broadcast* transmissions instead of conventional broadcast. More precisely, the destination MAC address of the encoded packet is set to one of the intended next hops, while an additional COPE-header specifies all the next hops of the native packets mixed together. By setting radio interfaces in promiscuous mode, COPE enables each node to overhear multiple encoded packets, while, at the same time, unicast transmissions ensure a higher level of reliability. To further increase the communication reliability, a local recovery strategy is performed through hop-by-hop ACKs sent asynchronously by all the next hops to which the encoded packets were headed. If any of the encoded native packets was not acknowledged within a given timeout, the packet is retransmitted, possible encoded again, but with a different set of native packets. Note that decoding and re-encoding packets at each intermediate node is important to avoid diverging paths, resulting into packets that move away from their destinations. Indeed, this problem is originated by the use of inter-flow coding that allow to mix together packets even if they are headed for different areas in the network.

Experimental results shown in [25] indicate that the coding gain provided by COPE is highly dependent on a set of factors, including traffic patterns and congestion levels. To quantify these interdependencies in arbitrary topologies, authors in [45] elaborate a linear programming formulation to model the maximum throughput achievable by COPE-style coding. An interesting contribution of this paper is also the notion of *coding-aware* and interference-aware routing for selecting routes that maximizes the coding gain while minimizing the interference due to the coded transmissions. However, the scheme proposed in [45] requires an exhaustive analysis of all possible coding opportunities that may arise after a given routing decision. A simpler and more practical coding-aware routing approach, called *ROCX*, is described in [37]. The ROCX scheme is based on a new routing metric, called *ECX*, which captures the expected number of coded transmissions needed to successfully deliver packets between two nodes communicating through a relay. Then, a linear programming problem is formulated to find paths between node pairs with the minimum ECX cost, i.e., which minimizes the expected total number of coded packets for a successful exchange of packets.

A common feature of the schemes described above is that the coding process tries to minimize the number of coded transmissions needed to successfully delivery a set of packets. A different approach is adopted in [35], which proposes a series of algorithms for *network coding with loss-awareness* (*CLONE*), based on the idea that coding must provide higher levels of redundancy in lossy wireless environments. In other words, coding decisions aim to introduce an adequate redundancy in network coding operations in order to achieve higher reliability. Specifically, in [35] the binary (i.e., involving at most two native packets) network coding problem is modeled through a graph-based formulation. Then, various coding strategies are defined by imposing different constraints on the coding process. For instance, the CLONE-MultiXOR heuristic tries to maximize the number of ways a native packet can be decoded by the intended next hop. However, the high complexity required for packet selection limits the applicability of these algorithms beyond binary coding.

A somehow simpler approach to implement inter-flow network coding is proposed in [50] with the *Intra-flow*&*Inter-flow MIXing* protocol (*I* $^2$*MIX*). The basic idea behind I$^2$MIX is that each node with packets buffered in its transmission queue creates random linear combinations of the same subset of the stored packets until all the respective next hops acknowledge their correct reception. Intuitively, each receiving node that is a next hop sends an ACK as soon as it is able to recover the original data from the coded packets it has received. Then, the decoded packets are stored in the receiver's transmission buffer and are used to generated new coded packets. In contrast to COPE, I$^2$MIX generates random linear combinations of stored packets. This permits to take advantage of any existing coding opportunity, thus simplifying the coding process and avoiding the use of reception reports. However, in I$^2$MIX the sender can stop to send combinations of the same set of packets, and to move to the next one, only if it receives an acknowledgment from the next hop of each flow. In addition, the coding decisions cannot be optimized because the status of neighbors' buffers is unknown. Both these simplifications of the coding process can easily produce a number of transmissions much higher than the one that would be generally needed by COPE to deliver the same number of native packets.

### *3.2.2. Intra-flow coding schemes*

In principle, intra-flow coding would permit to avoid the problem of discovering the state of neighbors' buffers. However, coding packets flowing between the same source and destination pair could give rise to a limited number of coding opportunities. To address this issue, *Multipath Code Casting* protocol (*MC*$^2$) [34] proposes to integrate coding with multi-path routing. Specifically, MC$^2$ relies on a legacy routing protocol to find a set of multiple, not necessarily disjoint, paths between a source and a destination node. This should ensure that multiple next hops exist for each node on a path. Then, relay nodes broadcast encoded packets generated using either native packets received by the source or other encoded packets received by neighbor nodes. Intermediate decoding is not allowed, and only the destination node collects encoded packets and reconstructs the original packets when it has received a sufficient number of linearly independent coded packets. Note that, similarly to ExOR, MC$^2$ performs its coding decisions on block of packets, also called *generations*, for limiting decoding overheads and state size at intermediate nodes. Finally, in order to provide reliability, two error control mechanisms are defined. First, a hop-by-hop local recovery is performed by each sending node, which overhears its neighbors' transmissions and retransmits some missed packets if necessary. Second, on a timeout the destination sends a unicast request for additional coded packets to the source, which can also be intercepted and managed by any intermediate node holding the missed packets.

The critical part of the MC$^2$ scheme is how to assign coding rates to the multiple paths, i.e., how to decide which next hop a packet must be sent to, and how many encoded packets should be generated along each path. To this end, a credit-based algorithm is proposed in [34]. Specifically, the source associates to each packet generation a given amount of *credits*, which represents the total number of packets (i.e., including coded packets) that should be used to transfer that block of native packets to the destination. Moreover, the number of packets the source is allowed to send per time unit is specified as a function of the generated credits. Then, whenever a packet is successfully transmitted to a next hop, a credit is also transferred to that node. Based on the credits associated to each node, forwarders are aware of the amount of packets that are waiting to be sent over each link. Then, to determine the best next hop to which a node should forward the packet it has received, the routing protocol applies the back-pressure algorithm [46] to the total credits accumulated by each node. In other words, the packet transmission is scheduled on the broadcast link with the maximum difference in the queued credits. The drawback of this approach is that this optimal scheduling is hard to implement, and the required node state grows exponentially with the number of neighbors.

An alternative approach to implement intra-flow coding by exploiting node overhearing is proposed in the *Intra-flow MIXing* (*IMIX*) protocol [50]. The basic principles are similar to the ones used in $I^2$MIX: the sender keeps coding the same set of *n* packets from the same data flow with random linear coding until receiving an acknowledgement from the next hop. Obviously, the next hop can generate an acknowledgment only after having received *n* linearly independent coded packets. Then, the final destination will recover the native data packets from the coded packet, while intermediate nodes only recode the received packets. It is important to note that with traditional acknowledgments, one ACK message is needed per every received data packets. On the contrary, with intra-flow coding only one acknowledgement is needed for every *n* packets. Thus, using linear coding it is possible to reduce the overhead of the acknowledgment process without the cost of increased protocol complexity. To some extent, IMIX can be viewed as a basic intra-flow solution in which every node simply codes and broadcasts packets stored in its buffer, irrespective of those packets received by its neighbors. To maximize the coding gain, which can be low for intra-flow coding since coding opportunities may be scarce, IMIX employs a coding-aware routing protocol, called *OSPR*, which selects network paths with least EXT value, taking into account overhearing opportunities. Compared to classical shortest-path routing, OSPR network paths generally include more hops to provide more overhearing opportunities.

### 3.3. Hybrid routing

As reported in Fig. 4, hybrid routing solutions are divided into two sub-categories: *destination-based* approaches and *neighborhood-based* approaches. The following two sections outline the most relevant schemes proposed for both approaches.

#### 3.3.1. Destination-based schemes

The (*MAC-independent Opportunistic Routing and Encoding*) protocol (*MORE*) proposed in [38] is the first practical system to combine intra-flow random linear coding with ExOR-style opportunistic routing. As in $MC^2$ (see Section 3.2.2) packets are grouped into blocks, which are now called *batches*, and coding is restricted to linear combinations of packets of the same batch. Similarly to ExOR [23], opportunistic routing is performed by letting the sender specify a prioritized list of candidate forwarders. However, in contrast to ExOR there is not a structured transmission schedule among the forwarders, but the coding permits random transmissions regulated through the 802.11 MAC protocol. More precisely, the source breaks up the file to be transmitted into batches of native packets, creates random linear combinations and broadcasts the resulting packets after adding a MORE header containing the forwarder list. Each receiving node discards a packet if it is not innovative, i.e., not linearly independent from the other packets stored in the node's local buffer, or if the node does not appear in the associated forwarder list. Otherwise, it refreshes the packet stream by linearly combining the received coded packets and rebroadcasting the newly encoded packets. Note that a linear combination of coded packets is also a linear combination of the corresponding native packets [38]. As soon as the destination is able to decode the whole batch, it sends an ACK to the source using shortest-path routing, causing the sender to stop forwarding packets from that batch and start processing the next batch. The intermediate nodes stop coding/sending packets from a certain batch as soon as they intercept the ACK for that batch sent by the destination, or they receive a packet belonging to a new batch. This strategy leads to a faster synchronization among nodes without requiring complex coordination procedures.

Although network coding reduces the number of required transmissions for successfully delivering a packet, the opportunistic paradigm allows any potential forwarder to send many coded packets. However, uncontrolled generation of coded packets results in redundant transmissions. The trade-off is between transmitting a sufficient number of coded packets to guarantee that the destination has enough innovative packets to reconstruct the native packets, and avoiding to inject in the network unnecessary packets that may cause congestion. To address this issue, MORE uses a heuristic algorithm to estimate the maximum number of transmissions that each node can perform after receiving a packet from an upstream node, which is a node farther from the destination than itself. This limit is computed by each node considering the loss probability in sending a packet to its neighbors, and the probability that the packet to be transmitted has not been yet overheard by downstream nodes, which are nodes closer to the destination than itself. Experimental results obtained in an indoor wireless testbed indicate that MORE's throughput gain over ExOR can be relevant when there is a chance of spatial reuse, because MORE allows multiple forwarders to access the channel simultaneously, which is hindered in ExOR [38].

As pointed out above, determining the acceptable rate for intermediate nodes represents a critical issue for hybrid schemes, especially in scenarios involving multiple flows. Thus, a number of recent papers have proposed optimization approaches for broadcast rate control, which maximize the benefit of network coding and broadcast transmissions while mitigating congestion. For instance, the *Optimized Multipath Network Coding* scheme (*OMNC*) proposed in [47] formulates the throughput-maximization problem as a linear programming problem, whose outcome is the optimal encoding and broadcasting rate for all nodes. A variant of this approach, called *DICE*, is proposed in [48] by taking a game-theoretic perspective. However, these enhancements of MORE require the exchange of a large amount of state information, which may be quite inefficient in lossy environments. A different and more practical scheme to improve routing efficiency is the *CodeOR* protocol proposed in [49]. The design idea of CodeOR originates from the observation that the coding process in MORE, and subsequent enhancements, is similar to a 'stop-and-wait' protocol because the source keeps coding packets of the same block, also called *segment* in CodeOR's notation, until it receives an explicit signal from the destination. In contrast, CodeOR allows the source to transmit a *sliding window* of multiple segments so that data coding/broadcasting is not limited to only one block of packets. Moreover, each intermediate node locally decides when it should start processing a new segment within the allowed window. Hence, we can say that coding decisions depend globally on the destination, but may adapt locally to the current situation. In practice, this adaptive behavior is implemented using two different ACK messages: an end-to-end ACK (called E-ACK) sent by the destination directly to the source node, and a hop-by-hop ACK (called H-ACK) sent by intermediate nodes. The former message is used to indicate that a segment of data packets have been received at the destination, and it regulates the window-based flow control similarly to TCP ACKs. The latter message is generated by a node to inform its upstream nodes that it has received a sufficient number of coded packets of the current segment so that it can continue the coding process on behalf of the source node, while the upstream nodes can move to the next segment. The peculiarity of the hop-by-hop support proposed in CodeOR is that the number of required coded packets is not constant but depends on the receiving rate of each node. Specifically, each node computes a *receiving threshold* proportional to its receiving rate, which specifies when it can assume that it has received enough packets in a segment. This rate-based threshold aims to balance the number of packets generated for a certain segment with the reception and decoding rate at the various downstream nodes.

#### 3.3.2. Neighborhood-based schemes

All the previous hybrid schemes adopt intra-flow coding because with opportunistic routing each packet has more than one possible next hop, and coordinating the coding process through multiple flows is probably less intuitive. In contrast, the hybrid scheme proposed in [39], called *XCOR*, is an attempt to combine inter-flow coding with opportunistic routing. To achieve this goal, XCOR abandons the approach followed in destination-driven coding strategies (e.g., MORE [38] and, partially, CodeOR [49]), to adopt a COPE-style coding, where the coding decisions are driven by neighbors' notifications and overhearing of neighbors' transmissions. More precisely, in XCOR the source constructs the set of nodes allowed to participate in the forwarding process of a certain packet starting from the shortest path, and sorting the candidate next hops in terms of ETX-proximity to the destination. Then, after each transmission, the nodes on the shortest path are allowed to rebroadcast immediately the packets they receive, while the other relays set a forwarding timer in proportion to their priority. In this way, if they overhear a transmission from a higher priority node, they can cancel their timers. Furthermore, similarly to COPE [25] each node periodically sends reception reports to inform its neighbors about the packets it has received. The most innovative aspect of XCOR scheme is the way these reports are used to regulate the mixing of packets. Specifically, let assume that a node is crossed by $m$ different flows. Then, the node computes the utility of each possible combination of packets belonging to these $m$ flows, in order to find the one that gives the largest utility. However, the number of possible combinations increases exponentially with the number of flows to code together. For this reason, XCOR applies an heuristic that examines the flows in a sequential order, giving higher priority to flows that are heavily loaded, so that the packet dropping probability is minimized.

## 4. Discussion

In this survey, we have examined the key challenges associated to the design of routing algorithms that use opportunistic forwarding and network coding to take advantage of the multiuser diversity and the broadcast nature of the wireless medium. To this end, we have presented a taxonomy of existing solutions relying on these novel routing paradigms, and we have analyzed their most representative features, relative strengths and weaknesses. From this overview, it is easy to identify some common functionalities and mechanisms that can be considered as basic building blocks for each solution. Thus, in the following we integrate the previous discussion by summarizing in Table 1 the spe-

**Table 1**
Summary of the key design choices of the wireless diversity-based routing approaches presented in this survey

| Protocol | Routing | Scheduling | Node priority | Coding | Duplicate suppression | ACK strategy | Prototype |
|---|---|---|---|---|---|---|---|
| ExOR | Opportunistic | Yes | ETX-based distance to destination | No | Batch map and priority-based forwarding timers | Batch map | MIT Roofnet testbed |
| MCExOR | Opportunistic | Yes | ETX-based distance to destination and home channel | No | Priority-based link-layer ACKs | Hop-by-hop slotted ACKs | No |
| SOAR | Opportunistic | Yes | ETX-based distance to destination + ETX-based proximity to minimum-cost path | No | Overhearing and priority-based forwarding timers | Hop-by-hop network-layer selective ACKs | No |
| OpRENU | Opportunistic | Yes | Residual network utility | No | Priority-based link-layer ACKs | Link-layer ACKs | No |
| ROMER | Opportunistic | No | Minimum-cost path to destination + link data rate | No | Overhearing and randomized forwarding | – | No |
| TDiCOR | Opportunistic | No | ETX-based distance to destination and other candidate forwarders | No | Overhearing | Cooperative acknowledgement | No |
| COPE | Legacy with pseudo-broadcast transmissions | – | – | Inter-flow XOR with reception reports/ guessing | – | Link-layer ACKs | 20-Node indoor testbed |
| ROCX | Legacy and coding-aware | – | – | Inter-flow RLC | – | – | No |
| CLONE XOR | Legacy with broadcast transmissions | – | – | Inter-flow loss-aware XOR with reception reports/ guessing | – | Link-layer ACKs | 12-Node testbed |
| $I^2MIX$ | Legacy with broadcast transmissions | – | – | Inter-flow RLC | – | Link-layer ACKs | No |
| $MC^2$ | Legacy multipath with broadcast transmissions | Yes | Transmission credits | Intra-flow RLC | – | Hop-by-hop overhearing, end-to-end retransmissions | MIT Roofnet testbed |
| IMIX | Legacy and coding-aware | – | – | Intra-flow RLC | – | MAC-layer unicast ACKs | No |
| MORE | Opportunistic | No | ETX-based distance to destination | Intra-flow RLC | Packet innovativeness + overhearing of a new batch + ACK overhearing | End-to-end ACKs | MIT Roofnet testbed |
| OMNC | Opportunistic | No | ETX-based distance to destination | Intra-flow RLC | Packet innovativeness + overhearing of a new generation + ACK overhearing | End-to-end ACKs | Emulation testbed |
| CodeOR | Opportunistic | No | ETX-based distance to destination | Intra-flow RLC | Packet innovativeness + hop-by-hop ACKs | End-to-end ACKs | No |
| XCOR | Opportunistic | Yes | ETX-based distance to destination + ETX-based proximity to minimum-cost path | Inter-flow XOR with reception reports | Overhearing and priority-based forwarding timers | Reception reports | No |

cific design choices made by the presented solutions for each of these key components. The aim of this schematic illustration is to further clarify the main features of the various schemes, as well as to permit an easier comparison between the various approaches. To this end, the first column specifies the solution name, as reported in the proposed taxonomy (see Fig. 2), while the other columns are dedicated to the most representative features of diversity-based routing approaches, which are a subset of the key challenges illustrated and discussed for each category. Hence, by looking at this concise description, basic differences and common aspects among the various approaches are immediately noticeable. In the following we briefly explain the meaning of the columns fields:

*Routing:* This field specifies if the routing approach is opportunistic or legacy. The former implies broadcast transmissions at each hop, while the latter can perform either unicast or broadcast/pseudo-broadcast transmissions, as explained in Section 3.2. Moreover, legacy routing can be either single-path or multi-path.

*Scheduling:* This field specifies if candidate forwarders coordinate their transmissions by establishing an ordering among them, or if each node autonomously decides whether to carry on with the forwarding process. In legacy routing approaches, a node can transmit only after receiving a packet from the previous hop along the predetermined path, thus the scheduling is not a task to be performed at the routing layer. On the other hand, this aspect is particularly important in the opportunistic routing case, where different solutions have been proposed.

*Node priority:* This field specifies the metric used for establishing node ordering. Some schemes propose new metrics, such as ECX [37], or use existing ones, such as ETX. Other schemes rely on path costs for routing operations, but they are not forced to use a specific routing metric. It is important to note that node priority is not necessarily associated to scheduling. For instance, in some cases node priority is used to prune some nodes from the forwarding process, e.g., if the ETX of a node is greater than that of the sending node. Finally, in legacy routing node priority is in general provided implicitly by the predetermined sequence of next hops in the selected network path(s).

*Coding:* This field reports, when applicable, the coding technique and the coding strategy. Generally, the coding techniques used in the presented schemes can be either the *XOR* operation or the random linear coding (*RLC*). On the other hand, the coding strategy refers to the possibility of combining packets belonging only to the same data flow (intra-flow coding) or to different flows (inter-flow coding).

*Duplicate suppression:* This field specifies the method used for minimizing the duplicate transmissions. This issue is particularly relevant for opportunistic-based routing, since many potential forwarders may send the same packets, leading to unnecessary transmissions. In case of network coding applied to legacy routing, duplicate suppression is not required because the routing protocol allows only some predetermined nodes to send data, each one towards its respective next hop(s). In principle, the redundancy provided by network coding is intended to favor packets decoding, thus duplicate suppression is not an issue. In contrast, opportunistic forwarding suffers from duplicate transmissions because there is a looser control on packets dissemination.

*ACK strategy:* This field indicates the method used to acknowledge packet transmissions. Most of the network coding-based solutions rely on link-layer ACKs, while hybrid approaches mainly use end-to-end ACKs, although some hybrid schemes may also use some kind of local recovery mechanism. The opportunistic-based routing schemes offer a higher variety of solutions, ranging from hop-by-hop network-layer ACKs to priority-based link-layer ACKs, and batch maps. In a few cases, the acknowledgment approach is not specified, thus it is not reported in the table.

*Prototype:* This field states whether a real implementation of the solution exists, "no" meaning that the protocol evaluation has been carried out based only on simulations.

## 5. Conclusions

In this survey, we have presented the wireless diversity-based routing paradigm by illustrating the main features, discussing the key challenges and presenting some of the most representative existing solutions. Although very promising results have been obtained in terms of throughput and reliability improvements, many research issues are still open. To conclude this article, we discuss these key issues with a special attention to the ones most related to the wireless mesh scenario.

(1) *Systematic analysis of coding and opportunistic gains:* Existing studies usually point out the advantages of the proposed solutions with respect to others developed with the same purpose, considering a specific testbed or a simulation environment created for the evaluation. However, higher performance improvements could be achieved by clarifying the impact of opportunistic, coding-based and hybrid routing primitives on more general scenarios. The motivation behind this requirement is that each of the above paradigms implies a different balance between design trade-offs and achievable gains. Therefore, a more systematic study is necessary to quantify the impact of different design approaches on various network settings. For instance, many theoretical research papers have shown the capability of the network coding paradigm to achieve the maximum flow capacity in arbitrary random networks, whereas analogous results are not available in the opportunistic nor in the hybrid area. Thus, an important direction for future work is the characterization of limitations, trade-offs and gains provided by the approaches described in this survey, with particular attention to the design parameters that mainly affect these results. This analysis will also be useful to reveal the capacity bounds achievable with each technique in different network scenarios.

(2) *Impact of traffic patterns:* To the best of our knowledge, opportunistic and coding-based approaches proposed so far do not take into account the variety of possible traffic patterns and characteristics. The strong assumption behind their design is that an application can tolerate a certain delay in exchange for throughput gain. In other words, throughput improvement has been considered the main objective in this research field, while only a little attention has been directed to delay reduction. However, wireless mesh networks are expected to provide advanced communication services supporting real-time traffic. In this perspective, it is clear that stronger guarantees must be offered to network users also in terms of maximum delay and/or delay variability. Moreover, the distinction among different types of traffic flowing across a network is not confined only to delay-tolerance. In a wireless mesh network, a traffic flow can be headed to a node either within the network or belonging to an external network, which imposes a differentiated traffic management for intra-mesh and extra-mesh traffic. Hence, diversity-based routing protocols should be designed to adjust their forwarding policies to meet different traffic requirements.

(3) *Cross-layer solutions design:* Generally speaking, the local situation of a node provided by lower layers represents a fundamental information during routing/coding processes. Clearly, designing diversity-based routing protocols by abstracting from MAC layer details seems to be hard and highly inefficient. On the contrary, a cross-layer design involving MAC/routing interactions appears to be more convenient, since routing strongly relies on MAC-related

aspects. For instance, transmission rate and routing metrics definition, as well as congestion-aware access methods for broadcast frames, and several other protocol features can greatly benefit from the cross-layer interaction, which permits several optimizations. Thus, cross-layer interaction should be considered as a key attribute in the development of cooperative routing protocols.

## Acknowledgments

This work was supported in part by the European Commission through FP7 Project EU-MESH, ICT-215320, and by the Italian National Project PRIN-WORLD, under Grant No. 2007R989S.

## References

[1] M. Conti, S. Giordano, Multihop ad hoc networking: the theory, IEEE Communications Magazine 45 (4) (2007) 78–86.
[2] M. Conti, S. Giordano, Multihop ad hoc networking: the reality, IEEE Communications Magazine 45 (4) (2007) 88–95.
[3] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.
[4] R. Karrer, A. Sabharwal, E. Knightly, Enabling large-scale wireless broadband: the case for TAPs, ACM SIGCOMM Computer Communication Review 34 (1) (2004) 27–32.
[5] R. Bruno, M. Conti, E. Gregori, Mesh networks: commodity multihop ad hoc networks, IEEE Communications Magazine 43 (3) (2005) 123–131.
[6] I. Chlamtac, M. Conti, J.-N. Liu, Mobile ad hoc networking: imperatives and challenges, Ad Hoc Networks Journal 1 (1) (2003) 13–64.
[7] D. Johnson, Y. Hu, D. Maltz, The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for IPv4, RFC 4728, February 2007. Available from: <http://www.ietf.org/rfc/rfc4728.txt/>.
[8] C. Perkins, E. Belding-Royer, S. Das, Ad Hoc On-Demand Distance Vector (AODV) routing, RFC 3561, July 2003. Available from: <http://www.ietf.org/rfc/rfc3561.txt/>.
[9] T. Clausen, P. Jaquet, Optimized Link State Routing Protocol (OLSR), RFC 3626, October 2003. Available from: <http://www.ietf.org/rfc/rfc3626.txt/>.
[10] D. Aguayo, J. Bicket, S. Biswas, G. Judd, R. Morris, Link-level Measurements from an 802.11b mesh network, SIGCOMM Computer Communication Review 34 (4) (2004) 121–132.
[11] J. Camp, J. Robinson, C. Steger, E. Knightly, Measurement driven deployment of a two-tier urban mesh access network, in: Proc. ACM MobiSys'06, Uppsala, Sweden, 2006, pp. 96–109.
[12] T. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 2002.
[13] D. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, in: Proc. ACM MobiCom'03, 2003, pp. 134–146.
[14] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: Proc. ACM MobiCom'04, 2004, pp. 114–128.
[15] M. Genetzakis, V. Siris, A contention-aware routing metric for multi-rate multi-radio mesh networks, in: Proc. IEEE SECON 2008, 2008, pp. 242–250.
[16] V. Mhatre, F. Baccelli, H. Lundgren, C. Diot, Joint MAC-aware routing and load balancing in mesh networks, in: Proc. ACM CoNEXT'07, 2007, pp. 1–12.
[17] Y. Yang, J. Wang, R. Kravets, Load-balanced Routing For Mesh Networks, ACM SIGMOBILE Mobile Computing and Communications Review 10 (4) (2006) 3–5.
[18] Y. Ganjali, A. Keshavarzian, Load balancing in ad hoc networks: single-path routing vs. multi-path routing, in: Proc. IEEE INFOCOM'04, vol. 2, 2004, pp. 1120–1125.
[19] N. Nandiraju, D. Nandiraju, D. Agrawal, Multipath routing in wireless mesh networks, in: Proc. IEEE MASS'06, 2006, pp. 741–746.
[20] M. Mosko, Garcia-Luna-Aceves, Multipath routing in wireless mesh networks, in: Proc. IEEE WiMesh'05, 2005.
[21] S. Diggavi, N. Al-Dhahir, A. Stamoulis, A. Calderbank, Great expectations: the value of spatial diversity in wireless networks, Proceedings of the IEEE 92 (2) (2004) 219–270.
[22] X. Qin, R. Berry, Exploiting multiuser diversity for medium access control in wireless networks, in: Proc. IEEE INFOCOM'03, vol. 2, 2003, pp. 1084–1094.
[23] S. Biswas, R. Morris, ExOR: opportunistic multi-hop routing for wireless networks, SIGCOMM Computer Communication Review 35 (4) (2005) 133–144.
[24] L. Pelusi, A. Passarella, M. Conti, Opportunistic networking: data forwarding in disconnected mobile ad hoc networks, IEEE Communications Magazine 44 (11) (2004) 134–141.
[25] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, XORs in the air: practical wireless network coding, IEEE/ACM Transactions on Networking 16 (3) (2008) 497–510.
[26] H. Dubois-Ferriere, M. Grossglauser, M. Vetterli, Least-cost opportunistic routing, in: Proc. Allerton Conference, 2007, pp. 994–1001.
[27] Z. Zhong, J. Wang, S. Nelakuditi, G.-H. Lu, On selection of candidates for opportunistic any path forwarding, ACM SIGMOBILE Mobile Computing and Communications Review 10 (4) (2006) 1–2.
[28] E. Rozner, J. Seshadri, Y. Mebta, L. Qiu, Simple opportunistic routing protocol for wireless mesh networks, in: Proc. IEEE WiMesh 2006, 2006, pp. 48–54.
[29] Y. Yuan, H. Yang, S. Wong, S. Lu, W. Arbaugh, ROMER: resilient opportunistic mesh routing for wireless mesh networks, in: Proc. IEEE WiMesh'05, 2005.
[30] R. Ahlswede, N. Cai, S. Li, R. Yeung, Network Information Flow, IEEE/ACM Transactions on Information Theory 46 (4) (2000) 1204–1216.
[31] S. Li, N. Yeung, R.W. Cai, Linear network coding, IEEE/ACM Transactions on Information Theory 49 (2) (2003) 371–381.
[32] T. Ho, M. Médard, J. Shi, M. Effros, D. Karger, On randomized network coding, in: Proc. 41st Allerton Conf., 2003, pp. 11–20.
[33] C. Fragouli, D. Katabi, A. Markopoulou, M. Médard, H. Rahul, Wireless network coding: opportunities & challenges, in: Proc. MILCOM 2007, 2007, pp. 1–8.
[34] C. Gkantsidis, W. Hu, P. Key, B. Radunovic, P. Rodriguez, S. Gheorghiu, Multipath code casting for wireless mesh networks, in: Proc. ACM CoNEXT'07, 2007, pp. 1–12.
[35] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, S. Sengupta, Loss-aware network coding for unicast wireless sessions: design, implementation and performance evaluation, ACM SIGMETRICS Performance Evaluation Review 36 (1) (2008) 85–96.
[36] D. Lun, M. Médard, R. Koetter, Efficient operation of wireless packet networks using network coding, in: Proc. IWCT 2005, 2005.
[37] B. Ni, N. Santhapuri, Z. Zhong, S. Nelakuditi, Routing with opportunistically coded exchanges in wireless mesh networks, in: Proc. IEEE WiMesh 2006, 2006, pp. 157–159.
[38] S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading structure for randomness in wireless opportunistic routing, in: Proc. ACM SIGCOMM'07, 2007, pp. 169–180.
[39] D. Koutsonikolas, Y.C. Hu, C.-C. Wang, XCOR: synergistic interflow network coding and opportunistic routing, in: Proc. ACM MobiCom'08 SRC, 2008.
[40] A. Zubow, M. Kurth, J.-P. Redlich, Multi-channel opportunistic routing, in: Proc. European Wireless 2007, 2007.
[41] J. Wu, M. Lu, F. Li, Utility-based opportunistic routing in multi-hop wireless networks, in: Proc. ICDCS'08, 2008, pp. 470–477.
[42] M. Lu, J. Wu, Social welfare based routing in ad hoc networks, in: Proc. ICPP'06, 2006, pp. 211–218.
[43] E. Ancillotti, R. Bruno, M. Conti, Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks, in: Proc. ACM PE-WASUN'08, 2008, pp. 7–14.
[44] A. Zubow, M. Kurth, J.-P. Redlich, Cooperative opportunistic routing using transmit diversity in wireless mesh networks, in: Proc. INFOCOM'08, 2008, pp. 1310–1318.
[45] S. Sengupta, S. Rayanchu, S. Banerjee, An analysis of wireless network coding for unicast sessions: the case for coding-aware routing, in: Proc. INFOCOM'07, 2007, pp. 1028–1036.
[46] L. Tassiulas, A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, IEEE Transactions on Automatic Control 37 (12) (1992) 1936–1948.
[47] X. Zhang, B. Li, Optimized multipath network coding in lossy wireless networks, in: Proc. IEEE ICDCS'08, 2008, pp. 243–250.
[48] X. Zhang, B. Li, Dice: a game theoretic framework for wireless multipath network coding, in: Proc. ACM MobiHoc'08, 2008, pp. 293–302.
[49] Y. Lin, X. Zhang, B. Li, CodeOR: opportunistic routing in wireless mesh networks with segmented network coding, in: Proc. IEEE ICNP'08, 2008, pp. 1092–1648.
[50] C. Qin, Y. Xian, C. Gray, N. Santhapuri, S. Nelakuditi, I$^2$MIX: integration of intra-flow and inter-flow wireless network coding, in: Proc. SECON'08, 2008, pp. 1–6.