JOKER: A Novel Opportunistic Routing Protocol

Ramon Sanchez-Iborra and Maria-Dolores Cano, *Member, IEEE*

***Abstract*—The increase in multimedia services has put energy saving on the top of current demands for mobile devices. Unfortunately, batteries' lifetime has not been as extended as it would be desirable. For that reason, reducing energy consumption in every task performed by these devices is crucial. In this work, a novel opportunistic routing protocol, called JOKER, is introduced. This proposal presents novelties in both the candidate selection and coordination phases, which permit increasing the performance of the network supporting multimedia traffic as well as enhancing the nodes' energy efficiency. JOKER is compared in different-nature test-benches with BATMAN routing protocol, showing its superiority in supporting a demanding service such as video-streaming in terms of QoE, while achieving a power draining reduction in routing tasks.**

***Index Terms*—Opportunistic routing, QoE, ad-hoc networks, JOKER.**

I. Introduction

OPPORTUNISTIC networks have emerged as a new networking-paradigm that is attracting the research community's interest due to its potential for enhancing communications between mobile smart devices [1], [2]. These networks are an evolution of the MANETs (Mobile Ad-hoc NETworks), including new functionalities that make them more efficient than their precursors. Specifically, opportunistic networks take advantage of the broadcast nature of the wireless networks, i.e., direct communications between two nodes can be overheard by nearest neighbors. In ad-hoc multi-hop networks, traditional routing protocols such as OLSR (Optimized Link State Routing) [3], AODV (Ad-hoc On-Demand Distance Vector) [4], or BATMAN (Better Approach To Mobile Ad-hoc Networking) [5] calculate a unique route between transmitter and receiver. Thus, each node just considers one single neighbor as the next hop to reach a given destination. However, with opportunistic routing protocols each node selects a set of its neighbors, referred to as candidates, as the potential next hops towards the final destination. The way each node selects its candidates and how they coordinate each other to pick the most proper candidate as the actual forwarder are the two key challenges in opportunistic routing. These characteristics determine the effectiveness of the opportunistic routing algorithm [2], [6]. Having different candidates to forward a packet may provoke multiple copies of the same packet in the network, causing extra overhead, collisions, etc. Thus, adequate candidate coordination schemes are needed in order to achieve an efficient synchronization among the possible forwarders. Furthermore, depending on the topology and the characteristics of the services flowing through the network, considering the greatest number of candidates is not always advantageous. In next sections this fact is clearly manifested and, under some conditions, considering a low number of candidates permits reaching higher levels of quality (Quality of user Experience) than employing a higher number of potential forwarders.

Another important point affecting ad-hoc networks is the terminals' energy consumption. As these devices are usually battery powered, the development of power-efficient techniques to diminish energy consumption in communication networks is a compelling need. Wireless card energy consumption has a remarkable weight in mobile devices' power draining [7], thereby, the development of efficient networking protocols and procedures is necessary in order to extend battery lifetime. Given the higher efficiency of opportunistic protocols compared to traditional proposals, the former may represent a real alternative for reducing the energy consumed in routing tasks.

In this work, a novel opportunistic routing algorithm is presented. This protocol, called JOKER (auto-adJustable Opportunistic acKnowledgment/timEr-based Routing), gets some functional features from the pro-active ad-hoc routing protocol BATMAN [5]. The main basis taken from BATMAN is its simplicity regarding routing tasks, which entails low computational and memory needs in the nodes, making it suitable for mobile devices with limited processing power. In a previous work [8], authors showed the capability of BATMAN to support multimedia traffic in low-consumption nodes by tweaking some of its configuration parameters. Thus, JOKER represents a step forward aiming at improving the capability of ad-hoc networks for transmitting heavy and demanding traffic such as multimedia content while reducing energy consumption in routing tasks. Concretely, it has been designed to work in IEEE 802.11 mobile devices, as this is the end-users' preferred way for accessing multimedia services. To the authors' knowledge, there is not any prior work addressing the tradeoff between the provision of quality for multimedia services and the reduction of nodes' energy consumption in opportunistic networks. Hence, the main contributions of this paper are the following:

- A detailed description of the novel opportunistic routing protocol JOKER is presented. Within this description, the main features of this protocol are compared with those of BATMAN.
- A new metric for selecting candidates is developed. This new algorithm takes into account both the packet-delivery reliability of the links and the distance-progress towards

Manuscript received July 31, 2015; revised December 6, 2015; accepted February 23, 2016. Date of publication March 22, 2016; date of current version May 19, 2016. This work was supported by the MINECO/FEDER project Grant TEC2013-47016-C2-2-R (COINS).

The authors are with the Department of Information and Communication Technologies, Universidad Politécnica de Cartagena, Antiguo Cuartel de Antigones, Cartagena 30202, Spain (e-mail: ramon.sanchez@upct.es; mdolores.cano@upct.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2016.2545439

the final destination. To the authors' knowledge, the presented metric is the first routing metric from the related literature including the fade margin value within its calculations [9].

- Two different implementations of JOKER are presented. The first one has been coded to work in a network simulation environment, while the other has been developed to work on real devices.
- A performance evaluation of JOKER in terms of QoE (Quality of user Experience), QoS (Quality of Service), and energy consumption is carried out. This study includes results for several JOKER parameter characterizations in order to analyze their impact on the energy consumed by the wireless cards as well as the quality of a highly demanding multimedia service such as video-streaming. These outcomes are compared with those attained by BATMAN.

The remainder of the paper is organized as follows. Section II reviews the related literature, focusing on those recent works presenting novel proposals regarding opportunistic routing algorithms. A deep description of JOKER is included in Section III. Section IV details the implementation of JOKER and the test-benches employed to obtain the presented results. These outcomes are showed in Section V, which presents a performance comparison of JOKER against BATMAN supporting video traffic. Finally, the conclusions are drawn in Section VI.

II. Related Work

During the last years, several works proposing novel opportunistic protocols for ad-hoc networks have been presented. Although most of them are based on the opportunistic-routing foundations, different works introduced interesting proposals, some of them even addressing QoS and/ or energy efficiency issues (for a more extensive review, please refer to [6]).

Lin and Chen [10] presented two different spectrum-map-empowered opportunistic routing protocols for regular and large-scale cognitive-radio ad-hoc networks. In this work, two different state-of-art network architectures are gathered together: cognitive radio and opportunistic routing. Thus, after obtaining a spectrum map indicating the available spectrum within the geographic area, the proposed protocols make use of a packet-delivery strategy based on network coding. Therefore, the candidate coordination phase is avoided. In turn, the candidates are chosen and ordered by using the end-to-end delay and ETX (expected number of transmissions) metrics. Additionally, for large-scale scenarios, geographical information is also considered for selecting the best forwarders towards the final destination. The proposed protocols showed improved performance (in terms of end-to-end delay) in comparison with other opportunistic protocols such as MORE (MAC-independent Opportunistic Routing & Encoding) [11] and a shortest-path greedy algorithm.

Following a similar approach, authors of [12] developed a routing protocol based on cognitive networking with opportunistic routing for wireless sensor networks. In this case, the candidate set selection criterion is the distance between each neighbor and the destination, enhanced with information about the network density. In turn, instead of using network coding, the RTS/CTS (Request to Send / Clear to Send) strategy is adopted for the candidate coordination. The proposed protocol was compared with a simple opportunistic spectrum access protocol and a geographic opportunistic routing protocol similar to GeRaF (GEographicRandom Forwarding) [13], showing to be the most efficient in terms of throughput, delay and energy consumption in an indoor environment.

Other schemes using the network coding strategy are those presented in [14]–[16]. In the case of [14], a multicast protocol was proposed. Authors employed different strategies: i) LP-based opportunistic routing structure, which defines the candidate set by taking into account the tradeoff between opportunistic forwarding and the contention probability, ii) opportunistic feeding, which permits different nodes to serve as cooperative sources for feeding other multicast receivers, iii) fast batch moving, by which the original source can quickly start sending coded packets of the next batch, allowing the coexistence of different batches in the network, and iv) inter-batch coding that permits improving the overall throughput by using RLNC (Random Linear Network Coding) coding operations instead of the traditional XOR coding scheme. Results showed how the proposed algorithm outperformed MORE [11] and Pacifier [17] in terms of energy efficiency, throughput, and response time. Likewise, works in [15], [16] proposed unicast and multicast algorithms, respectively, to improve the performance of real-time video communication in wireless networks. Both algorithms presented network-coding-based proposals aiming at improving the bandwidth utilization and throughput of the network. The provided results showed how the proposed protocols overcame the performance of other opportunistic routing algorithms. However, the test-benches employed in both works seem to be limited considering that just 1 simultaneous video flow was streamed to evaluate static and pre-defined topologies composed of up to 7 nodes. Gathering the concepts of network coding and content-based prioritization Seferoglu and Markopoulou proposed a more complete approach to opportunistic video coding for video-streaming taking into account the importance of video packets in network code selection [18]. By improving the application-level throughput, their proposal reaches higher levels of quality (in terms of PSNR (Peak Signal-to-Noise Ratio)) than other routing protocols.

From a different perspective, Wu *et al.* studied the problem of selfish behavior in multi-rate opportunistic networks, i.e., the case in which one node could manipulate its input/output metrics to lead its neighbors to take routing decisions for benefiting itself [19]. This fact could be especially harmful in the presence of highly demanding services. In this work, authors introduced a new opportunistic routing protocol that ensures the faithfulness of each node in the network, maximizing the end-to-end throughput. By employing an incentive protocol, the presented scheme maximizes each node's payoff. Besides, the overall network throughput is also improved with the presence of selfish nodes.

Work in [20] addressed the impact of link correlation in the candidate set selection. Thus, a link correlation aware metric was proposed aiming at enhancing the performance of the routing tasks by selecting the nodes with low correlated links as forwarder candidates. The performance of the proposed protocol was evaluated in a wireless sensor network, showing a great efficiency against a correlation unaware protocol in terms of energy consumption, number of transmissions and delivery ratio. Taking into account spatial reusability, Meng *et al.* introduced the concept of spatial reusability-aware (single path and any path) routing in [21]. They not only investigate the two approaches but also proposed an algorithm for node selection, cost calculation, and forwarding list determination aiming to maximize the end-to-end throughput in multi-hop 802.11 wireless networks. Their performance evaluation results evidence a notable improvement compared to the SAF (Shortest Any-path First) algorithm.

Regarding energy efficiency, work in [22] presented an opportunistic routing protocol for minimizing the energy consumption of the nodes composing a fixed wireless sensor network. Focused on a multi-hop chain topology, the proposed algorithm takes into account two key metrics to select the candidate set: the distance of the sensor nodes to the sink and the residual energy of each node. The candidate-coordination phase is carried out following an ACK strategy: each candidate successfully receiving the data-packet replies an ACK to the sender after a given period determined by its priority. This operation is performed only if no other ACK from a higher-priority node has been overheard. After the ACK reply is completed, the data-packet is forwarded. The proposed protocol was tested in both simulation and realistic test-benches, showing better performance, in terms of energy consumption, than GeRaF [13] and a MTE (Minimum Transmission Energy) protocol. Work in [23] also focuses on improving energy efficiency but from a different perspective, by using the ant colony gossiping and geographic forwarding. Results showed that their proposal increases the wireless sensor network lifetime and maintains strict packet delay needs, too.

Finally, from a QoS perspective, work in [24] proposed to exploit GOR (Geographic Opportunistic Routing) for multi-constrained QoS provisioning in wireless sensor networks. Thus, reliability and end-to-end delay QoS constraints were considered. To this end, the proposed scheme makes use of both the packet reception ratio and the distance progress towards the final destination provided by each one-hop neighbor as metrics to configure the candidate set. The candidate coordination is performed by employing a similar ACK strategy like that followed in [22]. Results showed a great performance of the proposed protocol in terms of end-to-end delay, communication cost, and delivery ratio, which are greatly valued metrics in scenarios with strict QoS requirements.

Although the discussed works propose a wide range of performance improvements to different-nature services, there is still a lack regarding the trade-off between multimedia-service QoE and power consumption in ad-hoc networks. Additionally, most of the proposed opportunistic routing protocols need to make deep modifications to the wireless card protocol stack by modifying or replacing the MAC layer protocol (e.g., 802.11). Thus, the algorithm proposed in this work tries to cover these gaps by i) improving the QoE of streaming services while reducing the energy consumed in routing tasks, and ii) being a ready-to-use piece of software compatible with other layers' protocols and without needing the modification of the off-the-shelf implementations of these protocols.

III. JOKER

JOKER (auto-adJustable Opportunistic acKnowledgment/timEr-based Routing) is an opportunistic routing protocol that takes some of its features from the architecture of the BATMAN ad-hoc routing protocol (for a deeper explanation of BATMAN, please refer to [5], [25]). JOKER is a proactive algorithm, designed to be as simple as possible to be suitable to work in a variety of devices with computational or energetic constraints, e.g., IEEE 802.11 mobile devices. Several JOKER's configuration parameters are adjustable so that JOKER is highly adaptive to different network conditions. In the following, the structure and the operational insights of JOKER are presented.

A. Main Characteristics

JOKER works between the link and the network layers of the OSI protocol stack. All the regular traffic in a node, as well as its self-generated control messages, are (de-)encapsulated, processed, and forwarded by JOKER. Thus, any protocol on higher layers (IPv4, IPv6, UDP, TCP, etc.) can run over JOKER without any modification. Additionally, as the node-addressing regarding routing tasks is based on MAC-addresses, there is no need for a network-layer addressing management system, e.g., IP, which simplifies the network setting-up operations. On the other hand, the link protocol also remains unmodified; so that, the original IEEE 802.11 protocol (or any other) can be used just by setting the network card to promiscuous mode, as JOKER is in charge of handling all the received packets (or frames).

All the traffic, except for the routing control messages that have their own format as discussed below, is encapsulated with the JOKER header (please, see Fig. 1). The header includes the following fields: a packet type that informs about the nature of the encapsulated packet (as explained later, three types of packets have been defined: unicast, ACK, or Forwarding packet); the packet TTL (Time To Live), defined as the number of hops that it is permitted to travel before being discarded (this value is initially set to the usual figure of 32); a packet-id, which is a 4-byte code that uniquely identifies each packet and it is obtained by calculating the CRC-32 of the payload; the final destination address, i.e., the MAC address of the packet's final receiver; and the candidate x address, which is the MAC address of each potential candidate to forward the packet, and in total there are $N_{candidates}$ candidates (which is a protocol configuration parameter).

This header is placed between those corresponding to the network and the link layers, adding $12 + 6 \cdot (N_{candidates} - 1)$ bytes of extra overhead. Note that the address of the candidate with the highest priority is not placed here but in the MAC header. This fact will be deeply discussed in the next sub-section. Also observe that, with the information included in the JOKER header, it is not necessary to pass the packet up to higher layers for routing or forwarding purposes. Furthermore, the inclusion of the packet's final destination address in the

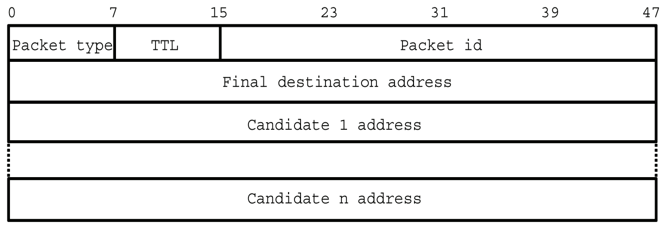


Fig. 1. JOKER header.

header allows the final receiver to accept and process lucky long transmissions (i.e., when a transmission is overheard by the final destination, although it should be handled by a prior hop). This situation, which will be discussed in next sections, reduces the transmission delay and avoids discarding valid packets in the receiving extreme of the communication.

B. Candidate Selection

As stated previously, JOKER takes inherited from BATMAN the way nodes interact with each-other regarding routing operations. Hence, the concept of transmitting a tiny packet (called OGM (OriGinator Message) in BATMAN) as the unique control packet employed is kept. By means of the reception and selective broadcasting of these messages, nodes are able to discover other nodes composing the network and establishing routes towards them. The original BATMAN metric made use of the number of OGM messages received from a given node via each link-local neighbor to estimate the quality of a route. In order to be able to find the best route to a particular end node, BATMAN counted the OGM messages received from each node in the network and logged which link-local neighbor relayed the greatest amount of messages. Currently, BATMAN's metric (Transmission Quality, TQ) has incorporated additional parameters regarding the asymmetry of the links and the number of hops in the routes towards the destination to obtain a more accurate assessment of the end-to-end path quality (please see (1)).

$$TQ = TQ_{local} \cdot TQ_{recv} \cdot f_{asym} \cdot hop\_penalty \tag{1}$$

TQ_{local} is the transmission quality locally calculated towards a direct neighbor to reach the final destination, TQ_{recv} is the transmission quality computed by the direct neighbor towards the final destination, f_{asym} is an asymmetric links penalization, and $hop\_penalty$ is a penalization for longer paths.

Previous works have demonstrated that BATMAN performs better than other ad-hoc routing protocols supporting real-time communications [8], [26]. So that, the metric that JOKER employs to select and sort the candidate set is based on the BATMAN's metric but including a fine tuning. This adjustment regards the distance between nodes, penalizing the nearest neighbors against the most distant ones. The aim of this tuning is achieving a greater distance progress to the final destination in each hop, as the geographic routing protocols propose [13], [27]. Thus, the link quality (LQ) that a node assigns to a given direct-link to reach a distant node is calculated as showed in (2),

$$LQ = TQ \cdot \frac{TQ_{max} - Distance\_penalty}{TQ_{max}} \tag{2}$$

TABLE I
Distance_penalty MAPPING

Progress towards the final destination	Fade margin	$Distance\_penalty$
HIGH	<10 dB	1
MEDIUM	[10 - 20] dB	3
LOW	> 20 dB	5

where TQ is the transmission quality calculated following the BATMAN algorithm (please, see (1)), TQ_{max} is the maximum value for the TQ (255 by default), and the $Distance\_penalty$ factor is calculated based on the distance between the nodes, in other words, the progress that the neighbor provides towards the final destination. This value is estimated by assessing the fade margin (FM) of the packets received from the corresponding neighbor. The FM is defined as the difference between the power of the received signal and the sensitivity of the wireless card. Many routing protocols that make use of the link-strength as unique metric, select the best neighbors trying to maximize this value. These approaches tend to choose the nearest neighbors, which has the drawback of calculating routes formed by many short links. Adding more hops to the path causes additional transmissions, with its corresponding possible collisions, and adds more delay to the communication. For those reasons, the $Distance\_penalty$ factor rewards the longest links against the shortest ones. Progress regions (Table I) have been defined according to different wireless card manufactures [28], [29]. Three different regions have been established: high progress (low level of received power, FM < 10 dB), medium progress (medium level of received power, 10 dB $\leq$ FM $\leq$ 20 dB), and low progress (high level of received power, FM > 20 dB), with associated $Distance\_penalty$ values of 1, 3, and 5, respectively. Observe that the tuning introduced by this factor is so mild that it does not have enough weight in the LQ calculation to modify very much the value of TQ. Thus, what it is achieved with this tuning is selecting the most distant nodes among the best qualified candidates; therefore, JOKER multi-metric keeps the reliability of BATMAN's metric but also incorporates a beneficial factor rewarding those nodes that provide a greater progress to the end extreme of the communication without injecting additional control-packets in the network or making use of GPS information. As mentioned previously, each node generates and broadcasts at a certain pace a simple packet to make the other nodes in the network know about its existence. Although the control messages sending interval (CMSI) is usually fixed in almost all the ad-hoc routing protocols, as demonstrated in previous works [8], [30], the adaptation of this period to the actual network conditions is crucial for the good performance of the routing protocol; thus, this also has a noticeable impact on the quality level of the service flowing through the network. Thereby, JOKER incorporates a dynamic adjustment of the routing control-messages broadcasting time-interval. This period depends on the throughput that is momentarily crossing each node; thus, nodes locally calculate its own CMSI following (3),

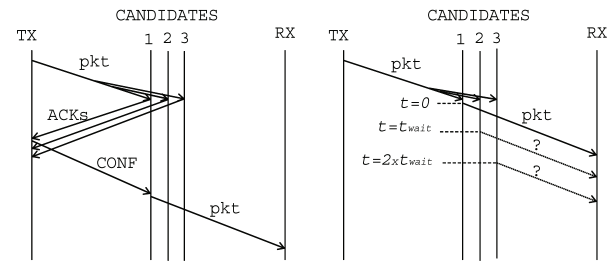
$$CMSI = 0.006 \cdot TP + 1.5 \tag{3}$$

where TP is the throughput crossing the node in kbps. The validity of this dynamic tuning of the CMSI was demonstrated in [31], showing to be especially beneficial for networks loaded with heavy traffic such as those supporting multimedia transmissions. Observe that the candidate selection is finally computed from a cross-layer perspective. The *LQ* value gathers network-layer metrics (end-to-end link reliability), link-layer metrics (link symmetry taken into account in the BATMAN's original metric), and physical-layer metrics (distance between nodes). Additionally, the dynamic CMSI regards the application layer by adjusting the broadcasting of control traffic to the throughput generated by top-layer services. According to this multi-layer metric, JOKER selects the best $N_{candidates}$ as potential forwarders for a packet destined to a given node. As mentioned above, $N_{candidates}$ is a protocol parameter and its impact on the performance of JOKER will be evaluated in following sections.

Finally, to avoid size-overspending in the JOKER header and to permit a wide compatibility with lower layer protocols the first candidate's address is not included in the JOKER header but is passed down to the link layer as an input parameter, letting this layer to include this address in its header's most appropriate field. This also permits JOKER to work as a traditional ad-hoc routing algorithm, making use of just one candidate that will perform as the unique potential next hop in the path towards the final destination.

C. Candidate Coordination

In JOKER, two different candidate coordination schemes, namely ACK-based or timer-based, can be employed depending on the requirements of the service flowing through the network. For applications that do not have stringent delay-constraints and with some reliability needs, the ACK-based candidate coordination methodology has been designed. Following this scheme, when the candidates receive the packet to forward, they extract its packet-id and return an ACK message with this identifier to the packet transmitter (Fig. 2(a)). ACK messages are sent as unicast packets without any candidates in the JOKER header in order to i) reduce its size and ii) avoid control packet storms. To reduce the delay in this process, there is not any priority rule for the order in which the candidates send their ACK messages, i.e., there is no distinction among candidates: once the packet is received, the ACK message is generated and sent. In turn, the packet transmitter answers with a Forwarding message packet (again, without candidates in the JOKER header) to the sender of the first received ACK. Therefore, this coordination scheme first makes use of the established candidate priorities computed by the candidate selection algorithm, and finally, it relies on the current state of the links between transmitter and candidates. In other words, the best candidate among all the best potential candidates is selected in the forwarding-operation moment. Once the Forwarding message is received by the chosen candidate, it automatically forwards the data packet. As mentioned in the previous section, this methodology adds more control traffic and delay to the transmission; nevertheless, the number of duplicated packets is reduced to 0, and the election of the forwarder is made based on the freshest information about the status of the relaying nodes and the links between them and the transmitter.



(a) ACK-based coordination (b) Timer-based coordination

Fig. 2. Communication diagram of both candidate coordination schemes.

In order to provide a control-message-free candidate coordination methodology that meets the requirements of heavy-traffic applications with low tolerance to delay, e.g., multimedia-content distribution, a timer-based coordination has been developed. By using this coordination scheme, when a candidate receives a packet, it waits a period of $t_{wait} \times (priority - 1)$ ms before forwarding the packet; thus, the highest priority candidate automatically forwards the packet once it is fully received and the rest of candidates listen to the medium before its forwarding timer is expired aiming at determining whether the packet has been forwarded by other candidates or not (Fig. 2(b)). The packet is only forwarded if no other previous packet-relaying is heard. As explained above the waiting timer can be tuned depending on the network and applications characteristics. This methodology does not inject any control packet into the network and reduces the extra delay of the ACK-based methodology. However, if a lower priority candidate does not hear the transmission of a higher priority candidate, duplicated packets can appear. The other drawback of this scheme is the extra delay added to the communication as the priority of the actual packet forwarded decreases.

D. Theoretical Analysis

Aiming at exploring the impact of the new JOKER's features in comparison with BATMAN, in this sub-section a theoretical analysis is presented. First, the advantages provided by the opportunistic paradigm adopted by JOKER will be discussed by considering the communication between the transmitter (TX) and the receiver (RX) in the network illustrated in Fig. 3; this figure also presents the delivery probability of each link. Based on these probabilities, BATMAN would compute the following path TX → node 2 → RX to establish the communication between the end nodes. Therefore, 3.5 transmissions would be needed on average to reach the RX node from the TX node: $1/0.4 = 2.5$ tries to transmit a packet from TX to node 2, and an additional transmission to reach RX from the forwarder. This way of calculating routes is inefficient as explained below:

1) By employing JOKER, configured to consider the three intermediate nodes (1, 2, and 3) as candidates to reach the RX from the TX, a "virtual link" would be generated with a delivery ratio of 0.73, $(1-(1-0.4)(1-0.33)(1-0.33) \approx 0.73)$; thus, only 2.37 transmissions would be necessary on average to communicate both end points (RX and TX): $1/0.73 \approx 1.37$ tries to transmit a packet

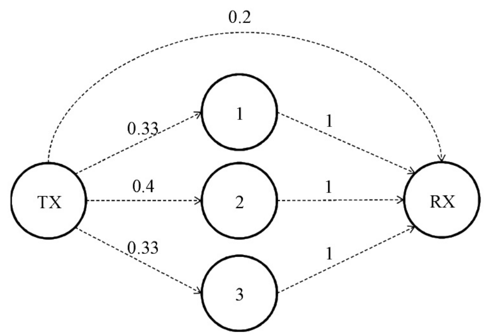


Fig. 3. Example network.

from TX node to the candidates, and another transmission to reach the final destination.

2) 1 out of 5 packets directly reaches the RX node ($1/0.2 = 5$). As stated previously, this phenomenon is known as lucky long transmission. BATMAN does not take advantage of it, so these valid packets would be discarded in the RX node, which also has to wait for the intermediate node 2 to forward the packet. Besides the wrong decision of dropping a valid packet, the communication would also suffer a longer delay due to the processing time added by the forwarding node. By using JOKER, the receiver would take this packet as valid improving the communication performance.

3) In case of a link drop, due to link failures or nodes motion, BATMAN has to recalculate the route after noticing these troubles, which consumes extra time, adding more delay to the communication. By employing JOKER, at least one backup node would be ready to automatically forward the packet if the primary forwarder fails.

In the following, the impact of the dynamic CMSI on the number of control packets in the network is explored. For this analysis, a 2-dimensional bidirectional Manhattan network is considered [32]. This symmetric network is defined by its size parameter (number of nodes in a row/column) n, hence having a total number of nodes of $N = n^2$. Assuming a uniform successful transmission probability for every link in the network, p, the number of control packets that each node broadcasts each CMSI, CP, can be modeled as (4) indicates (please, see [33]).

$$n_{even} : CP_{even} = 1+\sum_{d=1}^{\frac{n}{2}-1} 4dp^d + (2n-2)p^{\frac{n}{2}} + \sum_{\frac{n}{2}+1}^{n-1} 4(n-d)p^d + p$$

$$n_{odd} : CP_{odd} = 1 + \sum_{d=1}^{\frac{n-1}{2}} 4dp^d + \sum_{\frac{n+1}{2}}^{n-1} 4(n-d)p^d \tag{4}$$

Disregarding the even or odd nature of n, and considering the default BATMAN's CMSI (1 s) the number of control packets per second in the whole network (overhead), O, is calculated by (5),

$$O_{BATMAN} : \frac{CP \cdot N}{CMSI} = \frac{CP \cdot N}{1} = CP \cdot N$$
$$O_{JOKER} : \frac{CP \cdot N}{CMSI} = \frac{CP \cdot N}{0.006 \cdot TP + 1.5} \tag{5}$$

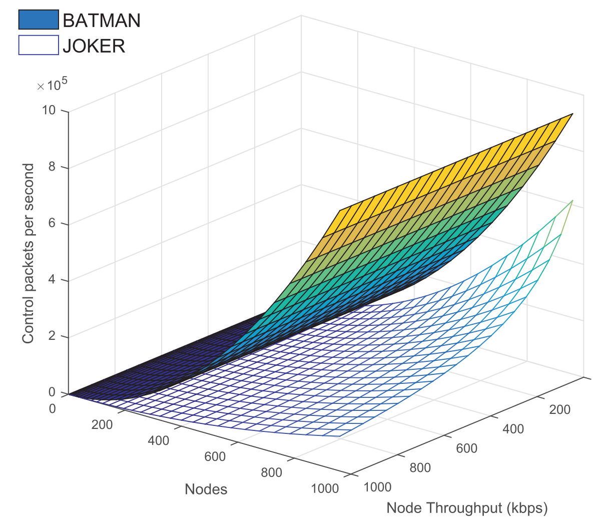


Fig. 4. Comparison of the overhead introduced by BATMAN and JOKER.

where TP is the average throughput crossing the nodes. Thus, Fig. 4 presents a comparison of the overhead introduced by JOKER and BATMAN in a worst case scenario with an ideal probability of successful transmission for every link in the network, i.e., $p = 1$. Notice that reducing this probability would decrease the absolute number of control packets in the network, but the comparative behavior between JOKER and BATMAN remains the same. Observe the uncontrolled increase in the number of control packets attained by BATMAN, whereas JOKER presents a more controlled flooding scheme, especially with high load in the network. In the less advantageous scenario for JOKER, in which the average throughput crossing the nodes is 0 kbps (no traffic in the network), the CMSI is 1.5 s. This enlargement of the minimum CMSI (recall that BATMAN's default interval is 1 s) means, at least, a 33% reduction in the number of control packets in the network.

IV. Development Details & Test-Bench Description

JOKER has been coded to work in both simulation and experimental test-benches. A JOKER version in a C++ implementation has been developed to be integrated in the InetManet Framework v.2.2 of the Omnet++ v.4.4.1 network simulator [34]; meanwhile, the JOKER version developed to work on real devices was coded in C, and has been tested in Ubuntu 12.04 (Linux Kernel 3.2.0) real computers.

JOKER's input parameters that can be tuned are: i) the number of potential candidates considered when transmitting a packet ($N_{candidates}$), ii) the candidate coordination scheme (ACK-based or timer-based), iii) the aforementioned waiting timer employed in the timer-based coordination (t_{wait}), and iv) the number of retransmissions that a node attempts if a packet is not received by the next hop.

A. Simulation Test-Bench

The simulated scenario in Omnet++ to evaluate the performance of JOKER against BATMAN was an IEEE 802.11g wireless ad-hoc network at 54 Mbps. The network was composed by 25 nodes with both static and dynamic mobility configurations. The nodes were randomly spread in a square area of 500 m × 500 m, thus having a node density of 100 nodes/km^2. Aiming at evaluating different-topology scenarios, 10 independent simulation instances with different seeds have been run for every protocol and scenario characterization. Hence, in order to generate the results, the average value for every measured parameter was taken, avoiding non-representative singularities. The 95% confidence intervals for every measurement have been also calculated. Each simulation-run lasted 75 s.

The traffic pattern employed was generated by the InetManet Framework's UDPVideoServer2, which emulates the transmission of a video flow from a video server to an end-user, i.e., emulating a video-streaming application. In this case, a realistic 30 s video-trace (available in [35]) has been employed, corresponding to a video encoded at 30 frames per second (fps) with the H.264/SVC single layer VBR (Variable Bit Rate) codec. As the considered devices are cell-phones, the video frame size was 176 × 144 px (QCIF format), the GoP (Group of Pictures) 16, and the number of B frames between I and P frames is 1 (G16B1). The encoding system generated VBR frames with an average size of 485.5 Bytes, transmitting them at 118 kbps. With this video coding configuration, it was intended to simulate a handset-oriented video-streaming application. As the 802.11 MAC layer's MTU (Maximum Transmission Unit) was set to its default value (2304 Bytes), it was not necessary to fragment each video frame into different packets. The starting time for each multimedia communication was generated randomly according to a Poisson distribution in a time interval of (0, 10 s), after the convergence time for each routing protocol.

Aiming to obtain the most realistic results, the PHY layer has been simulated by means of the Nakagami-*m* propagation model (*m* factor has been set to 5), considering the effect of fading in the wireless transmissions. Results have been also collected when characterizing the physical layer with the Free Space propagation model. Network performance has been evaluated in both static and dynamic conditions. Focusing on the nodes movement, it imitates the motion of a pedestrian, following a Random Way Point mobility pattern with a speed characterized by a Gaussian distribution with a mean value of 1.34 m/s and a standard deviation of 0.26 m/s. The stop period was set to a uniform value between 2 s and 5 s [36].

Regarding the node's hardware configuration, the normal operation of a mobile handset has been simulated. Thus, each node's wireless card has been configured following the real parameters of the widely used Broadcom's BCM4330 IEEE 802.11 a/b/g/n chip [37]. This chipset has been installed in different brand's cell-phones and tablets, namely, iPhone 4s, iPad 3, Samsung Galaxy S 2, or Samsung Galaxy Tab 2, among others. Thereby, the transmission power was set to 20 mW (13 dBm), with sensitivity at 54 Mbps of −74 dBm and a SNIR (Signal to Interference and Noise Ratio) threshold of 4 dB. The employed energy-consumption model is based on that proposed by Feeney *et al.* [38]. This model registers the wireless card's state (transmit, receive, idle, or sleep), calculating the momentarily consumption according to the current state and the card's consumption associated to this state. In this case, the

TABLE II
WIRELESS CARD ENERGY CONSUMPTION(EXTRACTED FROM [37])

State	Current Consumption
Transmit	250 mA
Receive	60 mA
Idle	1.25 mA
Sleep	0.18 mA

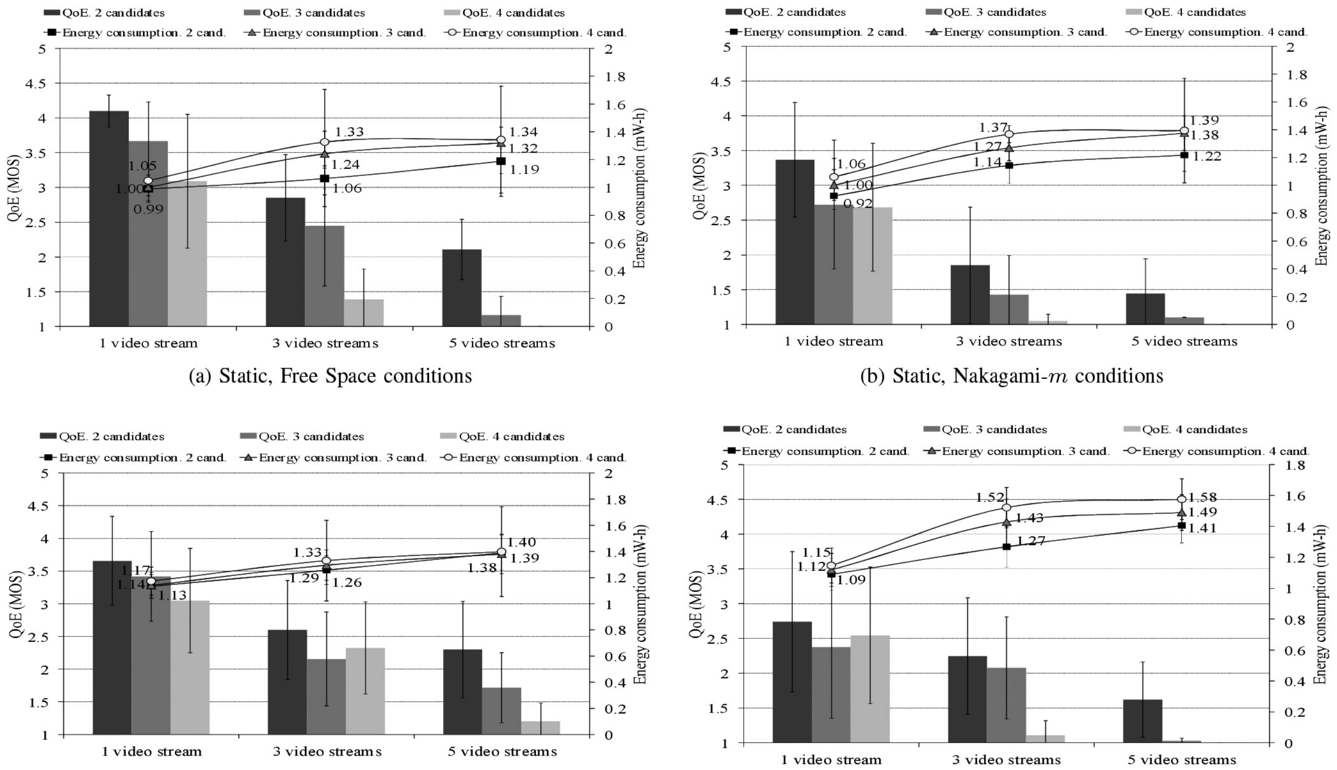
TABLE III
EMULAB NODES CHARACTERISTICS

Feature	Value
Emulab PC type	pc2400w
Processor	Core 2 Duo E6600
Processor speed	2.4 GHz
RAM	2048 MB
Wireless Card	Netgear WAG311 802.11a/b/g (Atheros)
Operative System	Ubuntu 12.04. 64 bits
Linux Kernel version	3.2.0
BATMAN version	Batman-adv 2014.2.0

energy-consumption values presented in the BCM4330 chip's data-sheet [37] have been employed to characterize the simulated wireless card (please, see Table II). It was assumed that the battery's output voltage is 3.6 V. All the parameters and configurations of the link and network layers' protocols, namely, IEEE 802.11 MAC and IP, respectively, have been set to their default values. As mentioned above, JOKER does not need any modification in other layers' protocols for its proper functioning.

B. Experimental Test-Bench

The evaluated scenario for the real implementation was similar to that showed in Fig. 3 (please, note that the link reliability values showed in that picture are only for the theoretical exemplification presented in previous sections and do not correspond to the actual delivery probabilities of the employed test-bench). This scenario was deployed by using the Emulab platform [39]. This network test-bench provides both wired and wireless nodes with high level of software customization. The Emulab nodes' characteristics are showed in Table III. In order to ensure that the TX node was not able to reach the RX node directly but by employing one of the relays (please, see Fig. 3), the wireless cards transmission power was set to 1 mW, since this is the minimum power value supported by the card [40]. The IEEE 802.11 channel 10, which was free of other transmissions during the experiment measurements, was employed, thus avoiding undesired interferences and collisions. The test traffic used in the Emulab test-bench was generated by the Distributed-Internet Traffic Generator v2.8.1 [41], which allows sending video traffic by generating bursty traffic characterized by a given distribution and bit-rate. The video packets were fixed to 512 Bytes and the average throughput was set to 150 kbps. Three independent and consecutive 60 s transmissions were established between TX node and RX node. 5 independent tests were performed in order to avoid non-representative results, calculating the 95% confidence intervals for every measurement, too. These values are not shown due to their low significance. Finally, aiming at emulating disconnection issues, the relay's network cards (node 1, node 2, and node 3) have



(a) Static, Free Space conditions

(b) Static, Nakagami-m conditions

(c) Dynamic, Free Space conditions

(d) Dynamic, Nakagami-m conditions

Fig. 5. Comparison of QoE (MOS) and energy consumption for a time-period of 75 s attained by JOKER-ACK for a variable number of simultaneous video streams and different number of opportunistic candidates. Results attained in static ((a) and (b)) and dynamic ((c) and (d)) conditions by considering the Free Space ((a) and (c)) and Nakagami-m ((b) and (d)) propagations models.

been randomly switched off during 5 second-periods during the transmission between the TX node and RX node. The beginning time from each disconnection event was scheduled following a uniform distribution from 30 s to 60 s.

C. Performance Metrics

Common to both test-benches, QoE and QoS metrics have been employed to evaluate the performance of the routing protocols. Video QoE has been measured in terms of MOS (Mean Opinion Score) following the model presented in [42] and the corrections to this model introduced in [43]. This model allows obtaining accurate video-MOS estimations by considering the PLR (Packet Loss Ratio), the coding scheme (k), the resolution (a) and the video-coding bit-rate (Br) as shown in (6).

$$V_q = 1 + 4k\left(1 - \frac{1}{1+\left(\frac{a\cdot Br}{v_1}\right)^{v_2}}\right)e^{-\frac{PLR}{v_3}} \tag{6}$$

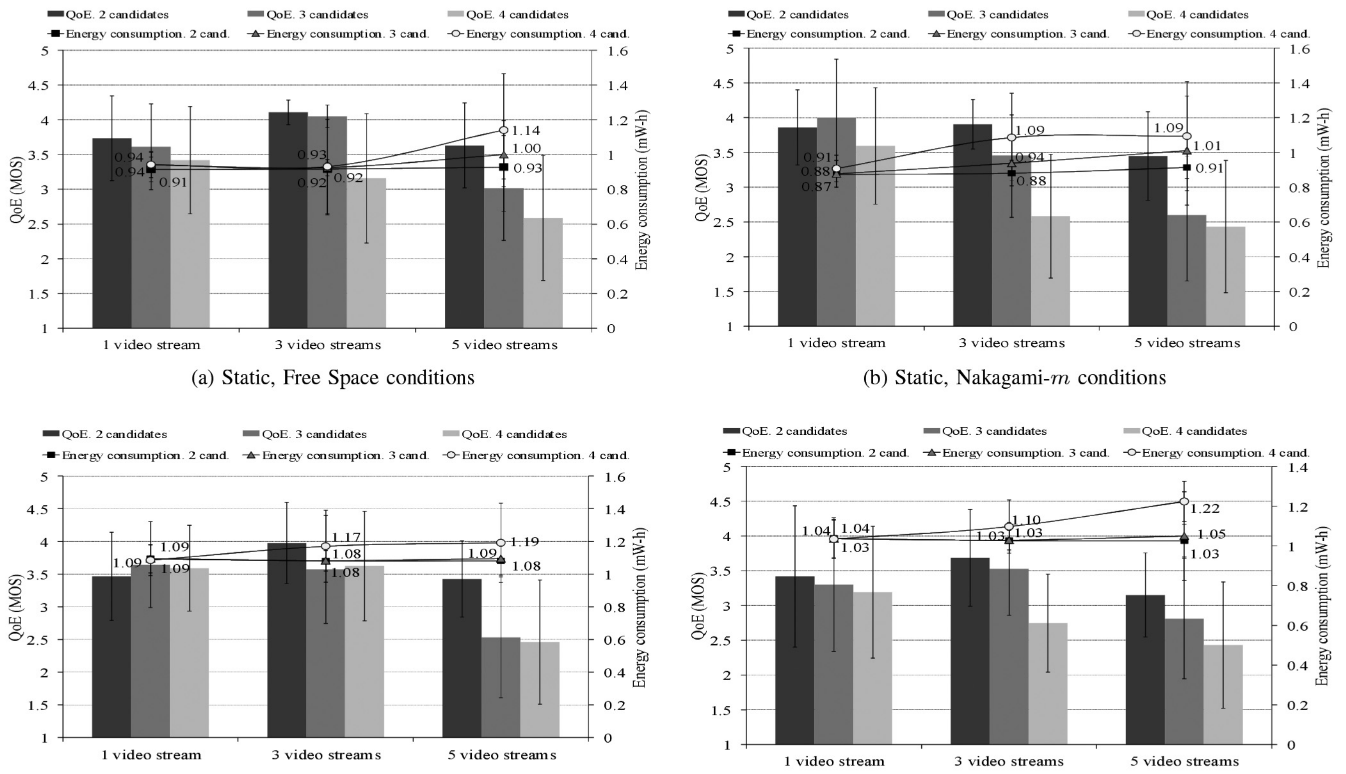
Besides, $v1$, $v2$, and $v3$, are experimental factors that depend on the coding scheme, the resolution applied, and the motion characteristics of the video. We have used the H.264 codec for the video-streaming transmission in QCIF format, so the values assigned to these factors were: $k = 1.12$, $a = 10.8$, $v1 = 0.366$, $v2 = 1.32$, and $v3 = 3.5$, consistent with the figures recommended by the model's authors [42], [43]. From the QoS perspective, the metrics considered to accurately assess the network performance have been: number of hops between video server and video client, PDR (Packet Delivery Ratio), throughput, and PLR. The energy consumed by the wireless card in routing tasks has been taken into consideration, too.

V. Results

In this section, the performance of JOKER supporting video-streaming traffic is explored. Different subsections showing the response of JOKER when varying its tunable parameters are presented. All these outcomes are compared with those attained by BATMAN. Results for this complete evaluation have been extracted from the simulator due to its easiness for tuning the environmental and network conditions. At the end of this section, additional outcomes extracted from the realistic test-bench are also exhibited aiming at showing the functionality of the JOKER implementation for real machines.

A. Protocol Adjustment

Fig. 5 shows the average QoE (MOS) and the average wireless card's energy consumption for a variable number of simultaneous video streams transmitted through the network managed by JOKER with the ACK coordination (in the following JOKER-ACK). Results have been extracted from different scenario configurations, namely, static and dynamic, characterizing the wireless transmission channel with the Free Space and Nakagami-m ($m = 5$) propagation models. It is clear that the best results in terms of both metrics under consideration



(a) Static, Free Space conditions

(b) Static, Nakagami-m conditions

(c) Dynamic, Free Space conditions

(d) Dynamic, Nakagami-m conditions

Fig. 6. Comparison of QoE (MOS) and energy consumption for a time-period of 75 s attained by JOKER-timer for a variable number of simultaneous video streams and different number of opportunistic candidates. Results attained in static ((a) and (b)) and dynamic ((c) and (d)) conditions by considering the Free Space ((a) and (c)) and Nakagami-m ((b) and (d)) propagations models.

(QoE and power consumption) are obtained when considering just 2 opportunistic candidates, no matter the scenario conditions. This fact has sense considering that the ACK coordination injects more packets in the network. Therefore, when more candidates are involved in the routing activities, more ACK packets are transmitted, which increases i) the probability of collision and ii) the channel-accessing waiting-time, hence, harming the video-streaming service. Regarding the impact of the transmission channel conditions, observe that there is an overall decrease on the attained QoE in hostile scenarios (Nakagami-m conditions, Fig. 5(b) and Fig. 5(d)) compared to that obtained in Free Space situations (Fig. 5(a) and Fig. 5(c)). This can be attributed to the impact of the fading channels on the transmitted packets. The channel variations increase the BER (Bit Error Rate), generating more corrupted packets and more retransmissions are necessary, provoking additional collisions and increasing the PLR. Focusing on nodes' motion conditions, no significant decrease in the attained QoE (MOS) is noticeable, but there is a logical increase on the battery drained by the nodes (please, compare Fig. 5(a) and Fig. 5(b) against Fig. 5(c) and Fig. 5(d)). Due to the continuous changes in the routes between transmitters and receivers, the routing tasks become more intense and hence the energy draining too.

Regarding the timer-based candidate coordination method (in the following JOKER-timer), Fig. 6 presents a similar comparison to that discussed above for the case of JOKER-ACK. Note that in these experiments the waiting timer (t_{wait}) has been fixed to 50 ms. Unlike the JOKER-ACK case, now there is not an evident superiority of a specific number of candidates. Although it is clear that the worst results are obtained when 4 candidates are considered, the highest quality levels are attained when employing 2 or 3 candidates depending on the scenario under study. Under heavy traffic conditions, i.e., with 5 simultaneous video streams, the best results are obtained with the lowest number of candidates, 2. In turn, when just 1 video connection is established, the greatest levels of QoE (MOS) are obtained when considering 3 candidates as potential forwarders. This behavior is explained as follows: as the number of candidates grows, the probability that one of them was not able to overhear the highest priority candidate also rises; consequently, more duplicated transmissions happen, harming the overall performance of the network.

Focusing on the impact of the wireless transmission channel conditions, in this case the protocol seems to be more robust than in the JOKER-ACK case. This can be explained by the lack of control-packets exchange in JOKER-timer. Whereas the effect of fading channels can affect JOKER-ACK's control messages, in the case of JOKER-timer, once the data packet is correctly received by any of the candidates, it will be certainly forwarded after a given time-interval (depending on the candidate priority). This forwarding operation is done without the need of waiting for a confirmation message, which is also prone to be affected by the channel conditions.

Comparing the performance of both candidate coordination methodologies, observe the overall improvement of QoE of JOKER-timer (Fig. 6) against JOKER-ACK (Fig. 5). The

TABLE IV
QoE (MOS) FOR A VARIABLE NUMBER OF VIDEO-STREAMS. RESULTS FOR JOKER-TIMER CONFIGURED WITH DIFFERENT t_{wait} VALUES

Video Streams	Static				Dynamic			
	25 ms	50 ms	75 ms	100 ms	25 ms	50 ms	75 ms	100 ms
1	3.9	3.9	3.9	3.8	3.3	3.4	3.2	3.4
3	3.3	3.9	3.8	3.3	3.4	3.7	3.2	3.4
5	2.8	3.4	3	2.9	3	3.2	2.8	2.7

simplicity of the former adapts better to the needs of the service under consideration. Video-streaming services require quick routing operations, capable of delivering the video packets as soon as possible to the end-extreme of the communication. Although JOKER-ACK ensures the hop-by-hop data transmission and no duplicated packets at the receiver, the introduced overhead harms the real-time communication in terms of extra delay and higher collision probability. Besides, the energy consumed by JOKER-timer is also lower than that drained by JOKER-ACK (please compare Fig. 5 and Fig. 6). This is because of the lack of control messages exchange in JOKER-timer that permits the wireless cards working in low-consumption mode for longer time than in the case of JOKER-ACK.

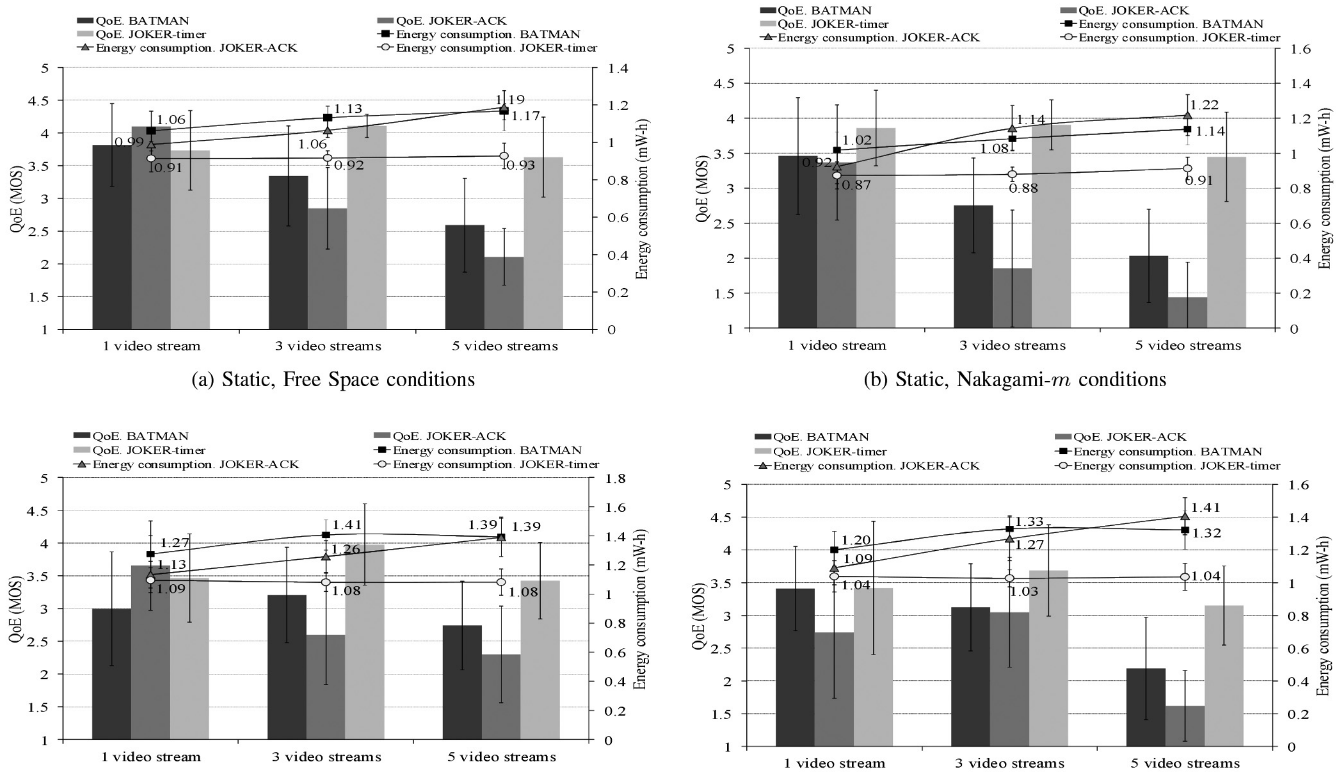
Another important configurable parameter in JOKER-timer is the time that one candidate overhears the channel before forwarding a packet (waiting timer, t_{wait}). An excessively short waiting timer could cause the multiple forwarding of the same data packet, reducing the network efficiency. In turn, assigning longer values to this parameter introduces extra delay in the case that the candidate with the highest priority fails to forward the packet. Thus, Table IV depicts the reached QoE (MOS) when tuning the t_{wait} parameter for a variable number of simultaneous video streams in the system. The number of considered candidates was set to 2 and the considered environment is that characterized by the realistic Nakagami-m propagation model. Observe how the best results in terms of QoE (MOS) are achieved by setting the waiting timer to 50 ms. This figure is a usual timer employed to configure timer-based candidate coordination schemes [44]. For that reason, hereafter the waiting timer is fixed to this value.

B. *JOKER vs. BATMAN*

In the following, an exhaustive comparison between JOKER and BATMAN is provided. To this end, the number of potential opportunistic candidates has been fixed to 2 and the JOKER-timer's twait interval has been set to 50 ms. In turn, BATMAN's configuration parameters have been set to their recommended values [5]. Thus, Fig. 7 presents a comparison of the QoE and power consumption obtained for BATMAN, JOKER-ACK, and JOKER-timer. Observe the superior quality values reached by JOKER, especially JOKER-timer, in comparison with those achieved by BATMAN. Focusing on the latter, it is interesting how the network performance (in terms of QoE) decreases when the traffic load increases (from 1 to 5 simultaneous video streams). This behavior is similar in the case of JOKER-ACK. In turn, JOKER-timer shows a greater strength supporting high traffic load. Note that QoE (MOS) sometimes increases with increasing load. These outcomes are obtained because for every simulation-run, different seeds for initializing the random variables, e.g., initial node positions, were employed. Depending on the TX and RX positions, the QoE of the video-streaming service could vary. Furthermore, please notice that this MOS increase only happens in the case of JOKER-timer. This exposes the great capability of this version of the protocol to adapt itself to support increasing traffic in the network. Regarding the impact of motion and the wireless channel hostility, observe how JOKER-timer seems to be less affected by these environmental difficulties compared with BATMAN and JOKER-ACK (observe the different scenario conditions in Fig. 7). It is highly remarkable that, using JOKER-timer, the QoE (MOS) level always remains over the notable value of 3, which is a recommended value by the ITU-T for ensuring an acceptable QoE for the end-users.

Focusing on the energy consumed by the wireless cards, again JOKER-timer overcomes the other options. Even in some scenarios, JOKER-ACK also presents lower battery consumption than BATMAN. The reasons for this improvement in the protocol energy efficiency are manifold. First, an important factor for decreasing the power consumption is the reduction of the control-packets broadcasts. By means of the dynamic CMSI, the pace at which the routing-packets are broadcasted is severely reduced, so that the wireless card can be in sleep mode for longer periods, permitting energy saving. Another key point for the showed reduction of battery draining is the new metric employed by JOKER for selecting the candidate set, which benefits from the nodes farther away from the transmitter. By using the $Distance\_penalty$ factor, the distance progress towards the final destination is increased at each hop, so the routes calculated by JOKER tend to be shorter than those calculated by BATMAN. Table V presents the average number of hops composing the path from the transmitter to the final receiver calculated by both BATMAN and JOKER. First, notice that the routes calculated by JOKER-ACK and JOKER-timer have almost the same length. This outcome makes sense as the candidate selection algorithm is not related with the coordination phase. Thus, observe how the routes calculated by JOKER are always shorter than those computed by BATMAN. Besides the new metric employed by JOKER that improves the distance progress in each hop, note that JOKER allows the final destination to take advantage of lucky long transmissions, hence, reducing the number of hops. One interesting result is that, despite the extra-difficulties added by node motion, the routes calculated in dynamic scenarios are sometimes shorter than those computed in the static ones. This behavior is attributed to the good performance of both protocols; with nodes motion, new opportunities for calculating more efficient routes than in the static case arise.

Another important factor for the network scalability and the node's energy efficiency is the number of retransmissions needed to deliver a packet from the transmitter to the final receiver. The ideal routing protocol would be that one able to successfully deliver a packet to its destination with the lowest number of transmissions. Thereby, Fig. 8 depicts the PDR of BATMAN, JOKER-ACK, and JOKER-timer in a static scenario when tuning the IEEE 802.11 retry limit. Note that for the previous experiments, this value was fixed to the standard



(a) Static, Free Space conditions

(b) Static, Nakagami-m conditions

(c) Dynamic, Free Space conditions

(d) Dynamic, Nakagami-m conditions

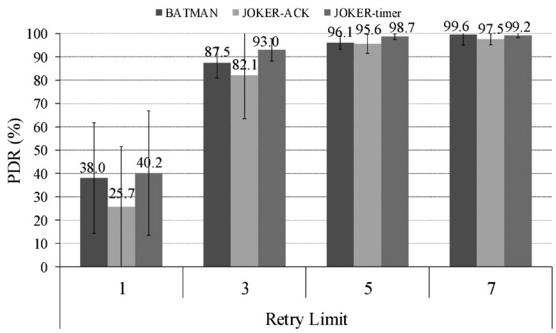
Fig. 7. Comparison of QoE (MOS) and energy consumption for a time-period of 75 s attained by BATMAN, JOKER-ACK, and JOKER-timer for a variable number of simultaneous video streams. Results attained in static ((a) and (b)) and dynamic ((c) and (d)) conditions by considering the Free Space ((a) and (c)) and Nakagami-m ((b) and (d)) propagations models.

TABLE V
AVERAGE NUMBER OF HOPS

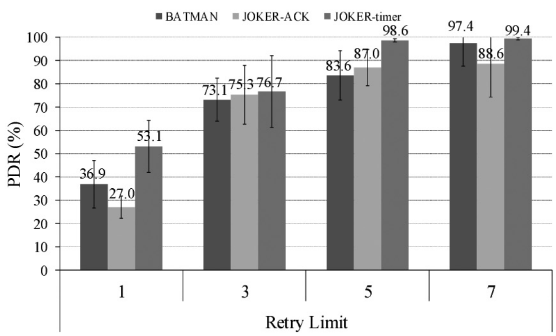
Video Streams	BATMAN		JOKER-ACK		JOKER-timer	
	Static	Dynamic	Static	Dynamic	Static	Dynamic
1	2.38	1.72	2.17	1.26	2.16	1.24
3	2.2	1.55	1.98	1.52	2.04	1.51
5	2.4	1.92	2.37	1.8	2.35	1.8

short retry limit, i.e., 7, which is widely used as default value. The better performance of JOKER-timer, in terms of delivered packets, compared to the other schemes is noticeable. In turn, BATMAN-ACK presents some difficulties with a low retry limit. The presented results have been extracted from a Nakagami-*m* hostile scenario, so JOKER-ACK control packets are impacted by fading channels. If the ACK or Forwarding packets do not finally reach their destinations, the involved data packet is dropped by the forwarder, hence reducing the PDR. Regarding the necessary retransmissions for successfully delivering data packets, observe that reducing the retry limit from 7 down to 5, JOKER-timer reaches PDR levels over the notable figure of 95%. On the other hand, BATMAN still needs 7 retransmissions to reach even lower PDR values. This reduction in the number of necessary retransmissions emphasizes that JOKER-timer presents much more efficient and scalable performance than BATMAN, permitting the improvement of the quality of the streaming service flowing through the network, as well as reducing the energy consumed by the nodes.

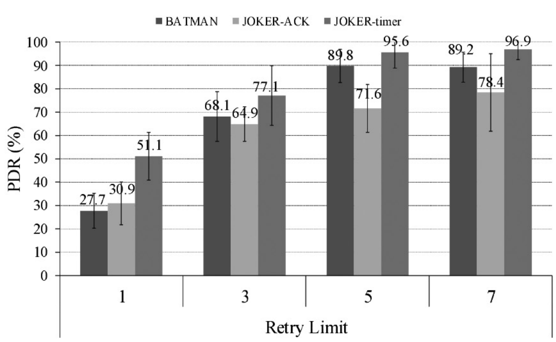
As aforementioned, an evaluation test has been also conducted by employing the BATMAN and JOKER (3 candidates) implementations in real machines. A similar topology to that shown in Fig. 3 was used. Each test consisted of the transmission of 3 independent 60 s video streams at 150 kbps between the RX node and the TX node by means of the relay nodes. To simulate link failures provoking disconnection events, the relay nodes' wireless cards were randomly switched off for a time interval of 5 s during the transmission between the TX node and the RX node (please, see the experiment description in Section IV). Fig. 9 presents the attained throughput and PLR for the 5 performed tests. Observe how both JOKER-ACK and JOKER-timer overcome BATMAN in all the conducted tests in terms of both metrics under consideration. Focusing on the throughput achieved in the RX node (Fig. 9(a)), JOKER-timer presents a minimum value of 135 kbps, which is a 10% reduction from the original stream, and for JOKER-ACK is 130 kbps (less than the 15% of reduction from the transmitted flow). In turn, the maximum throughput reached by BATMAN is just 116 kbps (a 22.6% decrease with respect to the sent traffic). Regarding PLR, (Fig. 9(b)), the differences between both algorithms are also notable. Observe that JOKER maintains low levels of packet loss (almost all values around 0% but a maximum value of 4% for JOKER-ACK in test 3); meanwhile BATMAN is not able to obtain values lower than 9%, with several tests achieving a PLR above 20%. The superior performance of JOKER can be attributed to the ready-to-use backup routes that the TX node employs when the first candidate is



(a) 1 video stream

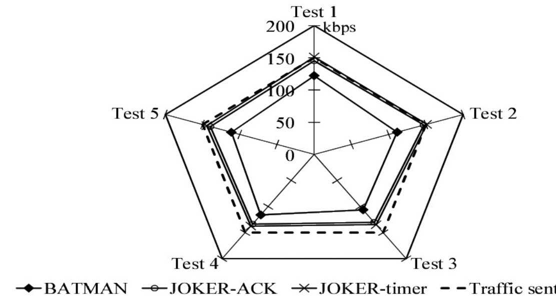


(b) 3 video streams

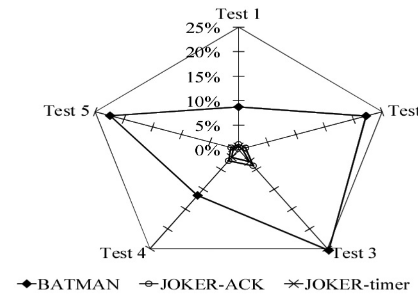


(c) 5 video streams

Fig. 8. PDR attained by BATMAN, JOKER-ACK, and JOKER-timer when tuning the IEEE 802.11 retry limit value. Results from a static scenario characterized by the Nakagami-*m* propagation model with 1 (a), 3 (b), and 5 (c) simultaneous video streams in the network.



(a) Throughput



(b) PLR

Fig. 9. Performance comparison of JOKER and BATMAN implementations for real machines. Throughput (a) and PLR (b).

TABLE VI
QoE (MOS) VALUES ATTAINED IN THE EXPERIMENTAL TEST-BENCH

	BATMAN	JOKER-ACK	JOKER-timer
Test 1	1.28	3.46	3.85
Test 2	1	3.39	3.84
Test 3	1	2.05	2.42
Test 4	1.1	2.5	3
Test 5	1	3.3	3.6

down. When this happens, the next candidate automatically forwards the packet, so the transmission continues without undesirable cuts. On the other hand, with BATMAN, the TX node needs to re-calculate a new route each time that the selected next- hop towards the RX node is down. Comparing both candidate-selection algorithms, similar to the simulation environment, the best results are attained by JOKER-timer. Nevertheless the performance of JOKER-ACK seems to be sufficient to support video traffic in the studied topology. From a QoE perspective, the high levels of PLR attained by BATMAN decrease video quality, achieving QoE (MOS) values near 1 in all the conducted tests (please, see Table VI). Recall that the QoE estimations in terms of MOS have been computed by employing (6). In turn, JOKER presents acceptable QoE (MOS) values above 3 in most of the performed tests. Again, the best results are attained by JOKER-timer that achieves remarkable QoE (MOS) values equal or greater than 3.6 in three out of five tests.

VI. CONCLUSION

In this work, an opportunistic routing protocol, called JOKER, addressing the trade-off between QoE in multimedia transmissions and energy consumption has been presented. Following the opportunistic paradigm, JOKER presents novelties in both the candidate selection, where a new metric that gathers the packet-delivery reliability of the links with the distance-progress towards the final destination has been introduced, and the candidate coordination, where two different procedures were included, namely, ACK-based and timer-based coordination schemes. Additionally, a dynamic adjustment of the protocol's control-message sending-interval was developed aiming at adapting JOKER to the actual network conditions and reducing energy consumption as well. Two different JOKER implementations, for simulation and experimental test-benches, were introduced and tested, allowing comparing the performance of JOKER supporting video-streaming traffic with that presented by the prominent ad-hoc routing algorithm BATMAN. From the attained results, the superior performance of JOKER in comparison with BATMAN was shown in terms of both QoE and energy efficiency. We highlight the performance of JOKER-timer that increases the network reliability and hence the multimedia service quality without adding extra overhead. These outcomes were confirmed in several scenarios evaluated in both simulation and experimental testbenches. In the lights of the presented results, it can be concluded that JOKER is an efficient proposal for the distribution of streaming traffic in ad-hoc networks, considering the energy constraints imposed by these systems, too. As future work, it is planned to continue exploring the opportunistic routing approach. To the authors' opinion, it permits an efficient management of wireless networks by taking advantage of the inherent characteristics of these systems, improving the overall network performance as well as the node's energy efficiency.

REFERENCES

[1] S. Liu and A. D. Striegel, "Exploring the potential in practice for opportunistic networks amongst smart mobile devices," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom'13)*, Sep. 2013, pp. 315–326.

[2] I. Woungang, S. K. Dhurandher, A. Anpalagan, and A. V. Vasilakos, *Routing in Opportunistic Networks*. New York, NY, USA: Springer, 2013.

[3] T. Clausen and P. Jacquet, "Optimized link state routing (OLSR) protocol," *IETF RFC 3626*, 2003.

[4] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *IETF RFC 3561*, 2003.

[5] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better approach to mobile ad-hoc networking (B.A.T.M.A.N.)," *IETF Draft*, 2008 [Online]. Available: http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00

[6] A. Boukerche and A. Darehshoorzadeh, "Opportunistic routing in wireless networks: Models, algorithms, and classifications," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 1–36, Nov. 2014.

[7] S. Chiaravalloti, F. Idzikowski, and Ł. Budzisz, "Power consumption of WLAN network elements," TKN, TU, Berlin, Germany, Tech. Rep. TKN-11-002, 2011.

[8] R. Sanchez-Iborra, M.-D. Cano, and J. Garcia-Haro, "Performance evaluation of BATMAN routing protocol for VoIP services: A QoE perspective," *IEEE Trans. Wireless Commun.*, vol. 13, no. 9, pp. 4947–4958, Sep. 2014.

[9] M. Youssef, M. Ibrahim, M. Abdelatif, L. Chen, and A. V. Vasilakos, "Routing metrics of cognitive radio networks: A survey," *IEEE Commun. Surv. Tuts.*, vol. 16, no. 1, pp. 92–109, Jan. 2014.

[10] S.-C. Lin and K.-C. Chen, "Spectrum-map-empowered opportunistic routing for cognitive radio ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 6, pp. 2848–2861, Jul. 2014.

[11] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 169–180, Oct. 2007.

[12] P. Spachos and D. Hantzinakos, "Scalable dynamic routing protocol for cognitive radio sensor networks," *IEEE Sensors J.*, vol. 14, no. 7, pp. 2257–2266, Jul. 2014.

[13] M. Zorzi and R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Trans. Mobile Comput.*, vol. 2, no. 4, pp. 337–348, Oct. 2003.

[14] P. Li, S. Guo, S. Yu, and A. V. Vasilakos, "CodePipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3264–3273, Dec. 2014.

[15] H. Shen, G. Bai, Z. Tang, and L. Zhao, "Flow-correlation-aware opportunistic network coding for video streaming over wireless networks," in *Proc. IEEE 8th Int. Conf. Commun. Netw. China (CHINACOM)*, Aug. 2013, pp. 155–160.

[16] K. Choumas, I. Syrigos, T. Korakis, and L. Tassiulas, "Video-aware multicast opportunistic routing over 802.11 two-hop mesh networks," in *Proc. 11th Annu. IEEE Int. Conf. Sens. Commun. Netw. (SECON)*, Jun. 2014, pp. 486–494.

[17] D. Koutsonikolas, Y. C. Hu, and C.-C. Wang, "Pacifier: High-throughput, reliable multicast without "crying babies" in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1375–1388, Oct. 2012.

[18] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in *Proc. IEEE Packet Video*, Nov. 2007, pp. 191–200.

[19] F. Wu, K. Gong, T. Zhang, G. Chen, and C. Qiao, "COMO: A game-theoretic approach for joint multirate opportunistic routing and forwarding in non-cooperative wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 948–959, Feb. 2015.

[20] S. Wang, A. Basalamah, S. M. Kim, S. Guo, Y. Tobe, and T. He, "Link-correlation-aware opportunistic routing in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 1, pp. 47–56, Jan. 2015.

[21] T. Meng, F. Wu, Z. Yang, G. Chen, and A. Vasilakos, "Spatial reusability-aware routing in multi-hop wireless networks," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 244–255, Jan. 2016.

[22] J. Luo, J. Hu, D. Wu, and R. Li, "Opportunistic routing algorithm for relay node selection in wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 112–121, Feb. 2015.

[23] Y. Yao, Q. Cao, and A. V. Vasilakos, "EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 810–823, Jun. 2015.

[24] L. Cheng, J. Niu, J. Cao, S. K. Das, and Y. Gu, "QoS aware geographic opportunistic routing in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1864–1875, Jul. 2014.

[25] Better Approach to Mobile Ad Hoc Networking (B.A.T.M.A.N.) [Online]. Available: http://www.open-mesh.org, accessed on Nov. 30, 2015.

[26] D. Seither, A. Konig, and M. Hollick, "Routing performance of wireless mesh networks: A practical evaluation of BATMAN advanced," in *Proc. IEEE 36th Conf. Local Comput. Netw.*, Oct. 2011, pp. 897–904.

[27] A. Darehshoorzadeh and L. Cerda-Alabern, "Distance progress based opportunistic routing for wireless mesh networks," in *Proc. IEEE 8th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2012, pp. 179–184.

[28] Radius. *Radio Fade Margin* [Online]. Available: http://www.radius.net/eng/services/radio-networking-tools/radio-fade-margin/, accessed on Dec. 10, 2015.

[29] UltraElectronics, Wood&Douglas. *What is fade margin* [Online]. Available: http://ultra-woodanddouglas.com/FAQ/what-is-fade-margin, accessed on Dec. 1, 2015.

[30] R. Sanchez-Iborra and M.-D. Cano, "An approach to a cross layer-based QoE improvement for MANET routing protocols," *Netw. Protocols Algorithms*, vol. 6, no. 3, pp. 18–34, Aug. 2014.

[31] R. Sanchez-Iborra and M. D. Cano, "Evaluation of the trade-off between energy consumption and QoE in video transmission with MANET routing protocols," in *Proc. XII Jornadas de Ingenieria Telematica (JITEL'15)*, Palma de Mallorca, Spain, 2015, pp. 145–152.

[32] F. N. Maxemchuk, "The Manhattan street network," in *Proc. IEEE Global Telecommun. Conf.*, New Orleans, LA, USA, 1985, pp. 255–261.

[33] D. Furlan, "Analysis of the overhead of B.A.T.M.A.N. routing protocol in regular torus topologies," University of Trento, Italy, Tech. Rep., 2011.

[34] Omnet++ [Online]. Available: http://www.omnetpp.org, accessed on Nov. 30, 2015.

[35] Video Trace Library [Online]. Available: http://trace.eas.asu.edu, accessed on Nov. 27, 2015.

[36] D. Helbing, "Traffic and related self-driven many-particle systems," *Rev. Modern Phys.*, vol. 73, no. 4, pp. 1067–1141, Dec. 2001.

[37] Broadcom. (2011). BCM4330 preliminary data sheet.

[38] L. M. Feeney and D. Willkomm, "Energy framework: An extensible framework for simulating battery consumption in wireless networks," in *Proc. 3rd Int. ICST Conf. Simul. Tools Techn. (ICST)*, Mar. 2010 [Online]. Available: http://dl.acm.org/citation.cfm?id=1808143.1808170

[39] B. White *et al.*, "An integrated experimental environment for distributed systems and networks," in *Proc. 5th Symp. Oper. Syst. Des. Implement. (OSDI'02)*, 2002, vol. 36, no. SI, pp. 255–270.

[40] F. Abdesslem, L. Iannone, M. Dias de Amorim, K. Kabassanov, and S. Fdida, "On the feasibility of power control in current IEEE 802.11 devices," in *Proc. IEEE 4th Annu. Int. Conf. Pervasive Comput. Commun. Workshops (PERCOMW'06)*, 2006, pp. 468–473.

[41] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Netw.*, vol. 56, no. 15, pp. 3531–3547, Oct. 2012.

[42] K. Yamagishi and T. Hayashi, "Parametric packet-layer model for monitoring video quality of IPTV services," in *Proc. IEEE Int. Conf. Commun.*, 2008, pp. 110–114.

[43] J. Joskowicz and J. C. L. Ardao, "Enhancements to the opinion model for video-telephony applications," in *Proc. 5th Int. Latin Amer. Netw. Conf. (LANC'09)*, Sep. 2009, pp. 87–94.

[44] A. Darehshoorzadeh and A. Boukerche, "Opportunistic routing protocols in wireless networks: A performance comparison," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2504–2509.

Ramon Sanchez-Iborra received the B.Sc. degree in telecommunication engineering and the M.Sc. and Ph.D. degrees in information and communication technologies from the Universidad Politécnica de Cartagena, Cartagena, Spain, in 2007, 2013, and 2016, respectively. He is an Associate Researcher and Teaching Assistant with the Department of Information Technologies and Communications, Universidad Politécnica de Cartagena. He has been a Visiting Researcher at the NetGNA Group, Universidade da Beira Interior, Covilhã, Portugal, and the Quality and Usability Laboratory, Technische Universität Berlin/Telekom Innovation Laboratories, Berlin, Germany. His research interests include evaluation of QoE in multimedia services, management of wireless mobile networks, green networking techniques, and IoT/M2M architectures. He has authored several papers in national and international conferences and journals. He has collaborated as a TPC member and Reviewer for international journals and conferences such as the IEEE ICC, EAI ADHOCNETS, or IEEE COMMUNICATIONS LETTERS, and is a member of many IEEE technical committees.

Maria-Dolores Cano (M'10) received the telecommunications engineering degree from the Universidad Politécnica de Valencia (UPV), Valencia, Spain, in 2000, and the Ph.D. degree from the Universidad Politécnica de Cartagena (UPCT), Cartagena, Spain, in 2004. She joined the UPCT in 2000, where she is currently an Associate Professor with the Department of Information Technologies and Communications. She has authored numerous papers in international journals and conferences. Her research interests include quality of service (QoS)/quality of user experience (QoE), security provisioning in telecommunications networks, and engineering education. She received the Fulbright Grant as a Postdoctoral Researcher at Columbia University, New York, NY, USA, in 2006, and also collaborates since 2007 with several universities in South America. She is a member of the Editorial Board of the *IET Journal on Wireless Sensor Systems*, as well as of the technical program committee of several the IEEE, IFIP, and IARIA international conferences. She has participated in several national and international research projects and also collaborates as a Reviewer for international funding research programs and international journals included in the JCR. She was the recipient of the Best Paper award at the 10th IEEE International Symposium on Computers and Communications ISCC'05 and was recognized as an Exemplary Reviewer by the IEEE COMMUNICATION LETTERS journal in 2010.