Video aware Multicast Opportunistic Routing over 802.11 two-hop mesh networks

Kostas Choumas†, Ilias Syrigos†, Thanasis Korakis† and Leandros Tassiulas‡

Abstract—**Opportunistic routing benefits from the broadcast nature of the wireless communication and outperforms the packets losses happen in a wireless environment, especially in case of multicast. In this work, we present and evaluate the performance of an innovative video-aware multicast opportunistic routing algorithm in 802.11 two-hop mesh networks. We enhance a state-of-the-art opportunistic routing scheme, namely MORE, that offers multicast but is not efficiently applicable to video streaming applications. Our scheme is able to support real-time applications with time-constraints. We improve the received video quality by classifying/prioritising the video traffic and efficiently orchestrating the multiple transmitters involved in multicast routing. The presented scheme, namely ViMOR, is evaluated through extended experimentation in the wireless testbed of NITOS. ViMOR succeeds in increasing the perceived video quality by up to 270% in a few scenarios or up to 175% in average, compared to MORE.**

Index Terms—**two-hop mesh network, opportunistic routing, network coding, multicast, video streaming, testbed experimentation**

I. Introduction

As the desire for Internet connection is growing rapidly, wireless access becomes more and more attractive, offering low-cost and easy-deployed network coverage. Wireless network deployment is more affordable compared to a wired network installation and the supported speeds are up to 1.3 Gbps in the widely used 802.11ac, making the wireless networking as the most preferable solution for Internet connectivity. Although the wired access is more stable and of high-bandwidth, it is unsuitable in environments where mobile devices opportunistically arrive or leave. The proliferation of these devices, such as laptops, tablets and smartphones, has been instrumental in bringing even more the development of the wireless access to the attention of networking researchers. More specifically, the routing problem remains one of the most challenging issues in the wireless access networking, while Opportunistic Routing (OR) seems to be an attractive solution to this problem.

† K. Choumas, I. Syrigos and T. Korakis are with the Dept. of ECE, University of Thessaly, Volos, Greece. E-mail: {kohoumas, ilsirigo, korakis}@uth.gr

‡ L. Tassiulas is with the Dept. of Electrical Engineering, Yale University, New Haven, USA. E-mail: leandros.tassiulas@yale.edu

Preliminary results of this work have been presented at IEEE SECON 2014 [1]. This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 671551. The European Union and its agencies are not liable or otherwise responsible for the contents of this document; its content reflects the view of its authors only.

OR is an appealing routing approach that leverages on the broadcast nature of the wireless medium and was first implemented in ExOR [2]. It does not choose the best sequence of relays between the source and the destination, as it happens in the most common routing algorithms, but it exploits broadcast transmissions and creates cooperative diversity sending packets through multiple relays simultaneously. Especially in the case of multicast scenarios, OR outperforms traditional routing by exploiting Network Coding (NC) and improving the routing efficiency [3]. Examples of these scenarios are large scale events happening in public areas, such as audio concerts or football matches, where the majority of the requests address the same multicast and real-time video stream e.g. a different view of the stadium. Although the wireless access is able to support these scenarios, the successful coverage of a public area with use of multiple WiFi gateways is impractical and inefficient [4]. The collaborative retransmission and processing of overheard information at some mobile devices, which act as wireless relays, offers coverage extension and throughput gain for all devices [5]. This is also indicated in the LTE/LTE-A standards and the upcoming 5G standardisation activities, that propose the utilisation of the end devices (User Equipment) as potential relays for streams destined to other devices. Either in WiFi mesh or cellular networks, the challenge is to forward multicast streams originated from the gateways or the cellular base stations to all destinations, by forging some devices to act as relays.

The contribution of this work is to explore OR in order to deal with this challenge and meet the necessities of video multicast streaming over two-hop wireless networks. We focus on two-hop relaying schemes over WiFi mesh networks, since more extended forwarding is impractical and inefficient, as we will explain later. This happens also in LTE/LTE-A, where the relay hop count is limited to two hops. In particular, in this work we elaborate the scheme of MORE [6], which is the state-of-the-art OR algorithm, focusing on satisfying the video multicast requirements.

It is worth mentioning that *real-time video streaming should be forwarded on time, even if this implies that some data may get lost decreasing the perceived video quality*. Based on this principle, we present the Video-aware Multicast Opportunistic Routing protocol (ViMOR), focusing on mesh networks where the end devices are one-hop or two-hops away from the source. In particular, our contribution is threefold: i) support for time-constrained streaming process, ii) improvements to the transmission policy regarding the selection of the relays and the orchestration of the transmissions of the relays and the gateways and iii) video quality enhancement by classifying and prioritising the video traffic. In contrast to MORE, ViMOR deals with the challenging video demands and improves the quality of the video perception in all end devices of each multicast group. We demonstrate the ViMOR performance through extensive experimentation in 802.11 mesh networks, paving the way for applying the same scheme to other wireless technologies, such as LTE/LTE-A.

To sum up, the paper is organised as follows: Section II presents related work in OR, starting with the state-of-the-art routing scheme of MORE and continuing with other proposed extensions of this scheme. Section III introduces the design and the concepts of the proposed ViMOR scheme. In Section IV, the performance of ViMOR is evaluated by conducting real experimentation in a wireless testbed and comparing its performance to the performance of MORE. We conclude by presenting our ideas for future work in Section V.

II. Related Work

ExOR [2] is the seminal implementation of an OR scheme. OR is included to a broader collection of routing algorithms that utilise the broadcast nature of the transmission over the wireless medium and achieves high throughput gains by exploiting the overheard information at multiple nodes. The OR concepts and ideas are currently applied to multiple contemporary schemes [7], [8], [9] for networking smartphones and portable devices. In contrast to the classic routing approach of forwarding a packet from each node to a specific next relay that is the "best" next relay according to the used routing philosophy, OR enables the node to transmit broadcast and then to optimise the choice of the next relay from the nodes that received the packet. ExOR is the first implemented OR scheme that improves throughput performance by enabling the more relaxed choice of next relay.

According to ExOR, the packets of each flow have to be divided into batches at the source node. The source does not insist on transmitting each packet to a specific next relay, waiting to receive an acknowledgment from this, but transmits broadcast packets for a limited number of times. In the same manner, each packet receiver retransmits for a specified number of times, that is different for each receiver, until the destination successfully receives all packets of a batch and sends an acknowledgment for this batch. The scheduling of the transmissions among the source and the relays relies on a modified Medium Access Control (MAC), that defines which time interval should be exploited by each node, in order to transmit its packets in a contention free environment.

MAC-independent Opportunistic Routing & Encoding (MORE) [6] is an improved ExOR version, which exploits Random Linear NC (RLNC) to mix (or encode) the batch packets before their forwarding. The source and the relays do not transmit the "native" packets of the batch, but linear combinations of them with use of randomly selected coefficients. The generated packets encapsulate the respective coefficients in a special header, which is named the MORE header. Destinations are able to reproduce the native packets by doing the inverse function of decoding, once they have received the appropriate number of mixed packets. The scheduling of the transmissions relies on the CSMA/CA of 802.11, rendering the routing protocol MAC-independent and directly applied. MORE outperforms ExOR in terms of throughput performance, even under the negative impact of the resultant contentions.

In addition, MORE imposes the source and the relays to transmit and retransmit continuously until an acknowledgment is sent by the destination. The difference between ExOR and MORE is that the source transmits unlimitedly, while each relay is assigned with a "credit", which is interpreted to the number of attempted transmissions for each received packet. Moreover, MORE makes multicast a straightforward extension of unicast, in contrast to ExOR that enables only unicast. Both protocols are mostly UDP compliant routing protocols, since ExOR needs to be better integrated with TCP and MORE supports multicast streaming that is only implemented over UDP.

The works in [10], [11], [12] have addressed some weaknesses of MORE regarding the multicast case. In MORE, the source proceeds to the next batch after receiving an acknowledgment from each multicast receiver, resulting in performance degradation in cases of one or more receivers having poor connections. This weakness is addressed by CodeOR [10] and Pacifier [12], which suggest solutions to overcome it. They enable the source to proceed with the next batch before receiving the acknowledgment of the current batch, using either a sliding window of multiple batches or a round-robin mechanism that allows the source to move to the next batch even if only one receiver sent acknowledgment for the current batch. These approaches succeed in suppressing the annoying variation on the batch forwarding duration, however, they do not eliminate this phenomenon, since they target again at 100% reliable forwarding. ViMOR is the first scheme that introduces the total rejection of the acknowledgment mechanism and enables a time-constrained forwarding process.

In OR-PLC [13], authors focus on video streaming, proposing (and not implementing) a NC technique that enables the partial reproduction of a batch, when the full decoding of the batch is not feasible. Instead of using RLNC, as MORE does, this work studies the utilisation of Priority (or progressive) Linear Network Coding (PLNC) in order to mitigate the transmission error effects. With OR-PLC, the source generates some coded packets as a combination of the highest priority native packets, which include the intra-frames of the streamed video. The intra-frames are individual and contain all of their own information, while the inter-frames require the intra-frames in order to be decoded in complete images. As follows, the loss of intra-frames is more damaging than the loss of inter-frames. PLNC allows the retrieval of intra-frames, even if some inter-frames are lost, streaming low quality video to the destinations with poor connection. To the best of our knowledge, ViMOR is the first implementation of an opportunistic routing scheme that utilises PLNC.

III. ViMOR Design

We present an enhanced and novel multicast OR scheme, improving the MORE protocol to accommodate to the video streaming necessities. According to the most common routing approach of path selection, the time needed for each wireless unicast transmission is not easily estimated, since the wireless channel variations may result in an unpredictable number of MAC retransmissions, until the transmitter receives a MAC acknowledgment. As follows, the duration of forwarding a packet through an exclusive path, which is the aggregation of the time intervals required by the individual unicast transmissions, is unconstrained and may exceed the time requirements of a video stream. On the other hand, the broadcast transmissions of OR do not rely on MAC acknowledgments and the resulting retransmissions, limiting the time needed for the packet forwarding, which now depends on the controlled number of broadcast transmissions performed by the source and the relays. In summary, OR is not reliable in packet delivery (since it does not use MAC acknowledgments), however, it always delivers on-time, which is more important in case of video streaming.

We strongly support that OR adapts very well to the video streaming requirements, especially in wireless networks with high probability of packet losses, because it can be more time-sensitive. It is worth mentioning that MORE and other related OR protocols use an application layer acknowledgment for reliable transmissions, however, in cost of their opportunity for time constrained streaming, coping with similar inconvenience with the most common routing approach of path selection. Furthermore, OR is inherently advantageous for multicast because of the utilised broadcast transmissions. Bearing all these in mind, a novel OR scheme is designed sharing some common features with MORE but, for example, it does not have any acknowledgment mechanism in any layer. The name of this new scheme is Video-aware Multicast Opportunistic Routing (ViMOR), The focus of ViMOR is on UDP streaming, since TCP is not used for multicast streaming. In the following list, we summarise the main differences of ViMOR compared to the MORE protocol:

- **Rejection of the acknowledgment mechanism** since on-time video streaming is more essential than reliability.
- **Redesign of the transmission policy** including the scheduling of the source and the relays transmissions.
- **Classification and prioritisation of the video packets** adopting enhanced PLNC instead of RLNC.

Due to the two first differentiations, ViMOR succeeds in high throughput of packets that have been neither received late nor lost. It satisfies the video requirement for time-constrained forwarding process and gives more transmission opportunities to the most error-susceptible wireless links. The third differentiation improves the video streaming performance by increasing the probability of delivery of the most crucial video packets and enhancing the quality of the delivered video.

We propose ViMOR focusing on multicast scenarios where **all destinations are at most two-hop away from the source**, as it is depicted in Figure 1. This is common to most of the existing and implemented relaying schemes, which adopt the same hop count limitation, such as LTE/LTE-A Relay. The rationale behind the two-hop count limitation is twofold: i) the performance of video wireless streaming over longer paths

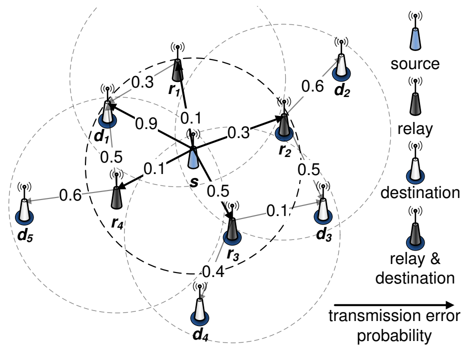


Fig. 1. The source is at most two-hop away from all destinations.

is usually degraded, due to the wireless channel variations that are more contingent as the paths get longer, and ii) the source is not able to apply the transmission policy if it serves more than two-hop away destinations, since it requires updated and on-line link evaluations (link evaluations that are done in parallel with the routing process). MORE claims that is able to support broader topologies, however, using link evaluations that were collected off-line [11]. This feature is undesired, since the link metrics are susceptible to the nodes mobility, the environmental changes, etc., thus they should be frequently updated [14]. Having in mind that a central point cannot sufficiently monitor broader topologies, with more than two-hop away destinations, there is no OR scheme that could be efficiently applied to these topologies.

On the other hand, the ETX estimation mechanism presented in Roofnet [15] is able to perform on-line link evaluations for the two-hop mesh networks of our interest. This mechanism imposes nodes to periodically i) broadcast fixed number of packets, ii) keep counter of the received packets from each neighbour and iii) report back to these neighbours the percentages of missed packets. In this way, each node is able to calculate the transmission error probabilities of its outgoing links and inform also its neighbours about these probabilities. At the end, every node has a knowledge of the quality of the outgoing links of its own and its neighbourhood, while the periodical execution of this process enables the knowledge update when the network status changes. It is worth mentioning that Roofnet enables also the statistics collection for links that are more than two-hops away from a node, however, using a flooding mechanism or information from packets going towards the collecting node, which are impractical for every multicast and single-source algorithm, like MORE and ViMOR. There are also some other metrics, such as EMTX [16], which focus on multicast streaming and claim better performance. However, their computational overhead creates a strong disincentive for being used by ViMOR.

One more issue is the utilised transmission rate for all nodes, which is the lowest physical rate in both MORE and ViMOR, for extending the coverage area, as much as possible. A rate adaptation scheme could have been considered along with a thorough analysis and the corresponding development. However, since the utilisation of high physical rates could reduce the coverage area, there is a tradeoff between these two important factors, which we prefer to not cover in this work due to space limitations. A promising future extension, towards

TABLE I
DESCRIPTION OF VARIABLES

Variable	Description
s, $\mathcal{R}$, $\mathcal{D}$	source, relays and destinations sets
N	number of nodes
k	number of packets included in batch
b	packet size (payload with headers)
ρ	utilised physical transmission rate
f, g	video frame rate and number of GOP frames
l	average number of packets needed for a GOP transmission
τ	time given for batch forwarding (slot)
c	number of transmissions in a slot (total credit)
c_1, c_2	credits of source and each relay
E	average probability of successful packet delivery among all destinations
$\mathcal{O}$	prioritised packet classes
k_o	number of class $o \in \mathcal{O}$ packets in a batch
o_I, o_{all}	priority classes of packets including intra-frames and all frames respectively
α	the intra-frames size proportion of the whole batch size

an efficient rate adaptation scheme, could be the gradual increase of the utilised physical rate, which is common in the whole network, until one of the transmission error probabilities increases. At this point the physical rate is stabilised until the one of the transmission error probabilities suddenly increases, thus the physical rate returns to its initial lowest value and the process is repeated.

Before proceeding to the ViMOR presentation, we introduce the related notation that is summarised and explained in Table I, together with some notation that will be introduced later. From now on, s is the notation from the source and $\mathcal{R}$ and $\mathcal{D}$ are the sets of relays and destinations respectively. The network consists of $R = |\mathcal{R}|$ relays, $D = |\mathcal{D}|$ destinations and $N = |\mathcal{R} \cup \mathcal{D}| + 1$ nodes. The source s breaks up the stream to *batches* of k equal-sized packets of size b. Then, the source broadcasts packets that are generated as linear combinations of the native batch packets. The coefficients of the linear combination are encapsulated to the generated packet. Once a relay $r \in \mathcal{R}$ receives a packet, it uses again linear combination to mix this packet with the previously received of the same batch and broadcasts the generated packet. When a destination $d \in \mathcal{D}$ receives k linearly independent packets, it is able to retrieve the k native packets by decoding the batch. Both source and relays broadcast with use of the lowest physical rate ρ.

The following Subsections III-A, III-B and III-C present in detail the three main differences of ViMOR compared to MORE.

A. Rejection of the acknowledgment mechanism

In MORE, the source and the relays generate and broadcast packets without limitation on the maximum number of transmissions, until an application layer acknowledgment is received by the source from each of the destinations. This acknowledgment informs the source that the current batch has been delivered and prompts this to initiate the forwarding of the next batch. Obviously, this mechanism cannot satisfy any requirement for maximum time needed for the batch forwarding, especially as the set of destinations gets increased.

On the other hand, ViMOR overcomes this issue by enforcing the source and the relays to broadcast for a bounded number of times. The source does not expect for an acknowledgment, but uses a countdown timer to the initiation of the next batch forwarding, with duration equal to a specific time interval, called *slot* τ. The source estimates the slot duration in respect to the time constraints imposed by the video stream. For example, a video stream that features a frame rate f and a Group Of Pictures (GOP) size of g frames has to be forwarded in a time period less or equal to g/f. If this GOP uses in average l packets of size b (or l/k batches) to be encapsulated, then each batch forwarding approximately requires a slot duration equal to $\tau = (g/f)/(l/k) = gk/fl$. After the expiration of the slot interval, the source proceeds to the following batch regardless of the successful or not delivery of the previous batch.

The **slotted mechanism** succeeds in forwarding always on-time, however, in cost of its inability to guarantee reliable forwarding of all batches. As we have already explained, this is a preferable characteristic, since it is a waste of energy and time for the source and the relays to insist in the forwarding of an obsolete batch, which includes frames that are out of time for the destinations.

B. Redesign of the transmission policy

A new transmission policy is applied in ViMOR concerning the orchestration of the time given in the source and the relays for transmitting. In MORE, the source continuously transmits encoded packets and proceeds to the following batch once it receives an acknowledgment for the current one. On the other hand, each relay is limited to generate and transmit encoded packets no more than its *credit* multiplied by the number of the packets it receives. The credits of all network nodes are computed based on the quality of all network links and, as follows, relays are the nodes "charged" with non-zero credit. In ViMOR, the credits have a different meaning, since the credit of each node indicates how many transmissions this node does during the forwarding of a batch, regardless of the number of the packets it receives. The total credit of the source and all relays is equal to an integer variable $c < \tau\rho/b$, which is the number of transmissions could happen in this network during a time interval τ, given that all packets share the same size b and are transmitted with the same physical rate ρ. At first, the source assumes that $c = \lfloor \tau\rho/b \rfloor$, while after the end of each slot it recalculates c based on the history of the actual number of transmissions happened at the past slots. More specifically, it estimates the new value of c as the sum of the 80% of the old value and the 20% of the number of the transmissions happened in the last slot. Of course these percentages are configurable, since they depend on the frequency of the c variation. Outside interference and unmodelled factors in wireless transmissions are the main reasons for this variation and are out of the scope of this publication.

ViMOR targets at maximising the total throughput of all destinations, **maximising the average probability of successful batch reception among all destinations**, which is denoted from now on by E. Towards this total throughput maximisation, the most challenging issues are to i) select the relays, as well as to ii) charge each of the source and the relays with the most efficient credit. We decouple these two subproblems, since the usage of a dedicated set of relays eliminates the first subproblem, and this is the most common case in most of the scenarios that this scheme could be applied (e.g. LTE/LTE-A technologies exploit a fixed set of relays consisting of femtocells that expand the wireless area of the base station, while the cellphones rarely operate as relays). However, even if a set of relays has to be selected as a subset of all network nodes, the source is able to do this by creating a multicast tree connecting all destinations to the source, and selecting all its neighbours on this tree as relays. This tree is the shortest-ETX one, similar to that exploited in Pacifier and developed by merging the shortest-ETX paths from the source to all destinations.

As for the second subproblem of charging with the most appropriate credit the source and all relays, there are two orthogonal challenges that needs to be addressed. On the one hand, the source needs to be charged with the highest feasible credit c_1, in order to forward as much as possible throughput to all one-hop away destinations. On the other hand, each relay should also be charged with the appropriate credit c_2, in a way that the two-hop away destinations are also fed with satisfying throughput. The source and the relays should transmit k times at least for forwarding the minimum number of packets required for the batch decoding, thus $c_1, c_2 \geq k$. We also propose all relays to be charged with the same credit, an approach that is also followed in other works [17] and makes the estimation of the credits assigned to the node significantly less computational complex. Obviously, the slot should not be over or under-utilised, thus $c_1 + Rc_2 = c$, which means that $c_1 \in \{k, k+1, ..., c-Rk\}$ (if $k > c - Rk$, then we satisfy only the one-hop away destinations giving all credit to $c_1 = c$). We follow the same approximation with MORE, aiming at maximising the aforementioned probability E for a batch consisting of only one packet ($k = 1$), since for $k > 1$ the corresponding problem is too complicated. The source is able to retrieve the c_1 and c_2 credits by exploiting Algorithm 1, which is a “Golden section” search algorithm and converges due to the convexity of E over $c_1 \in \{k, k+1, ..., c-Rk\}$. The proof of this convexity is given in Appendix A.

Except for the efficiency of the transmission policy, its capability to be implemented and used in a real development was a matter of serious concern. Due to our focus on topologies where the end devices are at most two-hops away from the source, the latter one is able to apply Algorithm 1, since it leverages on the ETX mechanism and has all the necessary information regarding the transmission error probabilities of the network links. Then, the source lets the relays also know the c_2 credit through periodical broadcasts, which are also exploited by the ETX mechanism for estimating the transmissions error probabilities. Last but not least, the low complexity of Algorithm 1 enables the real time estimation of the credits even under low slots, since it is $O((R+1)D\log c)$ that is much better than the corresponding $O(DN^2)$ complexity of MORE for multicast streaming. The $O((R+1)D\log c)$ complexity seems to be even more promising, having in mind that i) R usually does not contain more than 4 to 6 nodes, which sufficiently cover and extend the area surrounding the source, as well as the fact that ii) for large values of c, the algorithm converges rapidly in less iterations than $\log c$, since E is close to its maximum value for almost all values of c_1.

Algorithm 1 Computing c_1, where $E(x)$ is the probability E for $c_1 = x$ and $c_2 = (c-x)/R$. The algorithm tries to find the maximum of the convex function $E(x)$ by successively narrowing the initial range $[k, c-Rk]$ of value x by golden ratio ϕ. It converges when the length of the range including the solution is bounded by 1, after at most $\log(c)$ iterations.

```
y_l ← k
y_r ← c − Rk
φ ← (√5 − 1)/2
x_l ← y_l + (1 − φ)(y_r − y_l)
x_r ← y_l + φ(y_r − y_l)
for |y_l − y_r| > 1 do
    if E(x_l) > E(x_r) then
        y_r ← x_r
        x_r ← x_l
        x_l ← y_l + (1 − φ)(y_r − y_l)
    else
        y_l ← x_l
        x_l ← x_r
        x_r ← y_l + φ(y_r − y_l)
    end if
end for
c_1 ← ⌈arg max_{x∈{x_l, x_r}} E(x)⌉
```

Finally, our experimentation results illustrate the efficiency of the proposed policy and its significant benefits compared to MORE. ViMOR gives more transmission opportunities over the lossy links, while MORE does not, since it enforces the source to continuously transmit competing with its neighbours for the medium access and resulting in sharing equally the transmission opportunities with them. Regardless of the quality of the network links and the respective credit assignment of MORE, 802.11 CSMA/CA shares equally the channel access among the competing transmitters, since they always have to transmit such as being saturated. On the other hand, in ViMOR, the contentions are limited due to the *first-decode-then-transmit* policy followed by the relays. More specifically, the relays are enforced to begin transmitting only after decoding the received packets from the source and retrieving the k native packets of the forwarded batch. In this way, the relays does not also spend transmission opportunities that could be exploited by the source for transmission of packets that contain less information, since they are not encoded as linear combinations of all k native packets.

C. Classification and prioritisation of the video packets

The third ViMOR change is the replacement of the Random Linear Coding (RLNC) of MORE with Priority Linear Coding (PLNC), which classifies the packets using a set $\mathcal{O}$ of priority classes. Having in mind that video streaming consists of packets with different priority, PLNC follows a hierarchical encoding scheme. More specifically, some video packets include segments of base layer frames (I-frames) and are more important than the packets with segments of enhancement layer frames (B or P-frames), since the former packets can be depicted by their own, while the latter packets need the I-frames in order their segments to be depicted. The I-frames without the B/P-frames are able to support a video stream of lower quality.

In ViMOR, we define priority classes of packets, where each class $o \in \mathcal{O}$ contains the k_o most important packets of a batch. We utilise two classes o_I and $o_{all} \supset o_I$ that consist of the high priority and all packets respectively. The o_I class includes the segments of the intra-frames and contains $k_{o_I} = \alpha k$ packets per batch, since we assume that the intra-frames of a batch require a proportion of the batch size less or equal to α. The o_{all} class contains all batch packets and subsequently $k_{o_{all}} = k$. Each relay shares its credit proportionally among the classes, meaning that it transmits αc_2 and $(1-\alpha)c_2$ packets as linear combinations of the o_I (important) and o_{all} all packets respectively. Keep in mind that the first αc_2 packets can also be considered as combinations of all packets (with zero coefficients at the least important packets), thus they can also be used for the decoding of all packets. The source keeps behaving as before, using RLNC instead of PLNC. On the other hand, each destination runs two concurrent decoding processes: i) one for the incoming packets generated from the combination of the o_I packets, and ii) one for all incoming packets. The two decoding processes are performed concurrently, hence enabling the decoding of the most important o_I packets, even if the decoding of all packets is unsuccessful. This enables the delivery of a video stream of lower quality that, however, is better than delivering nothing.

IV. Experimentation Results

The ViMOR implementation is based on the framework of Click Modular Router [18]. Click consist of packet processing modules called elements, that implement specific router operations. In this work, we leverage on the Click implementation of MORE [19] and we extend it according to the three aforementioned proposals for video streaming.

We evaluated ViMOR by conducting experiments under several network topologies, using the experimental platform of the NITOS testbed [20]. NITOS offers a wireless indoor testbed enabling experimentation with no interference. The hardware and software features of the utilised NITOS nodes are presented in Table II. In our experimentation, we chose 802.11g. Our architecture can be directly applied to any 802.11 protocol without requiring modifications or influencing in any other way the performance comparison between MORE and ViMOR. We also conducted the same experiments in 802.11n, but since we tested the lower physical bit rate that is 6.5 Mbps, the performance of both algorithms was almost the same with this of 802.11g. We finally present the experimentation results on 802.11g in order to enable their comparison with other results of existing research on opportunistic routing algorithms, and is mainly done in 802.11g. Also, we conclude at the end that our main idea for future work is to extend and apply this scheme in the LTE/LTE-A Relay.

The experimental evaluation of ViMOR required the development of several network topologies with links of various packet loss rates. Since it is very difficult to find the wanted conditions in a testbed with stationary nodes, we enforced all nodes to accept a proportion of the received packets by sampling their incoming packets. In NITOS, almost all nodes shape a full mesh network with lossless links (transmission error probabilities very close to zero). In order to create lossy links with specific transmission error probabilities, we used a Click element that filters the incoming packets and allows each packet to pass through with a given probability, depending on its source MAC address. Using this distributed filtering, we gained the full control of the connectivity map, replicating any lossy link. It is worth to mention, that there is no difference if a packet is actually lost or discarded in the receiver, since all transmissions are broadcast and there are no acknowledgments. The topologies we used for our experimentation are depicted in Figure 2, where each link connecting two nodes is tagged with the corresponding transmission error probability.

TABLE II
Basic Configuration of NITOS nodes

Model	Icarus nodes
CPU	Intel i7-2600 Proc., 8M Cache, at 3.40 GHz
RAM	Kingston 4 GB HYPERX BLU DDR3
Storage	Solid State Drive 60 GB
Wireless cards	two Atheros 802.11a/b/g/n (MIMO)
OS	3.2.0-31-generic Ubuntu precise
Driver	compat-wireless version 3.6.6-1-snpc
Click	version 2.0

There are also three other important issues that we had to cope with in our implementation. Firstly, we modified the code of MORE in order to forward video packets pushed from the upper layer, rather than using dummy packets created by MORE. Secondly, we added the ETX broadcast mechanism of Roofnet, with a view to use updated link evaluations during our experimentation, enabling the source to run periodically Algorithm 1 and recompute the credits. Last but not least, we utilised the queue notification scheme of Click in order to reduce the number of empty pull requests, a.k.a. the unreplied requests made by the wireless device for coded packets to be transmitted. We notify the wireless device when there is not any packet for transmission, avoiding the CPU-consuming process of the pull request, since otherwise the CPU utilisation of Click was almost 100%.

A. Experimentation in the 4-nodes topology of Figure 2(a)

In the first class of our experiments we use the topology of Figure 2(a), where s is the source, $\mathcal{R} = \{r\}$ are the relays, $\mathcal{D} = \{d_1, d_2\}$ are the destinations and e_1 and e_2 are the configurable transmission error probabilities. The choice of the relay is trivial, since it is the only node connecting s with the destinations. The fixed parameters are that all nodes use $\rho = 6$ Mbps as physical transmission rate, all packets have the same payload equal to 1470 bytes and size $b = 1470 + 36 + 20 + 8 + (22 + k) = 1556 + k$ bytes, since their payload is encapsulated with WiFi (36 bytes), IP (20 bytes) and UDP (8 bytes) headers/trailers, as well as with the MORE header ($22 + k$ bytes) that is also adopted by ViMOR and lies between the WiFi and IP headers. The additional k bytes in the header of MORE carry the coefficients used in the process of generating the corresponding packet with linear coding.

(a) $\mathcal{R} = \{r\}, \mathcal{D} = \{d_1, d_2\}$ (b) $\mathcal{R} = \{r_1, r_2\}, \mathcal{D} = \{r_1, d_1, d_2, d_3, d_4\}$

Fig. 2. Two different topologies with 4-nodes (a) and 7-nodes (b) used in our algorithm evaluation.

In the subsequent lines, the evaluation of ViMOR is presented. Each of the aforementioned ViMOR's contributions is evaluated separately with the conduction of three different sets of experiments that aim to assess the potential improvement offered by each one. Furthermore, an additional set of experiments that examines the CPU utilisation of the source, relays and destination nodes.

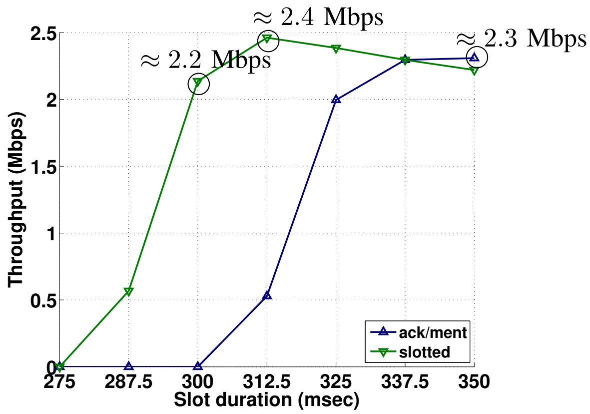
1) Slotted vs. acknowledgment mechanism: In the first experimentation set, we compared performance in terms of throughput for the slotted video-aware mechanism we proposed in ViMOR (details in Subsection III-A) with the acknowledgment mechanism in MORE. The comparison was performed using almost zero transmission error probabilities, specifically $e_1 = e_2 \approx 0.001$, and $k = 64$, which corresponds to ViMOR's optimal value of k and one of the best values for MORE [6]. As mentioned before, in MORE, the source transmits constantly, while a relay retransmits an exact number of packets for each one it received. For this topology, this exact number equals to one. The transmission of the next batch by the source occurs only after the reception of an aggregate acknowledgement from both destinations. Contrary, in ViMOR's slotted mechanism, the transmission of packets from the next batch by the source starts only when the current slot is expired, without considering if the current batch is decoded by the destinations.

Plots in Figure 3(a) depict the average throughput of the packets decoded on-time between the two destinations for both mechanisms. The decoded packets that we consider on-time are these that have been delivered within a time interval which is less than the slot duration τ. The traffic sent by the source can be larger than the corresponding throughput, since packets that are also included are either the lost ones, in case of the slotted mechanism, or the late received ones in case of the acknowledgement mechanism. At the horizontal axis the slot duration in milliseconds is represented, while on the vertical axis the corresponding measured throughput in Mbps is depicted, which can be transformed to batches per second by scaling its values by $10^6/(8kb)$ (denominator is the batch size in bits).

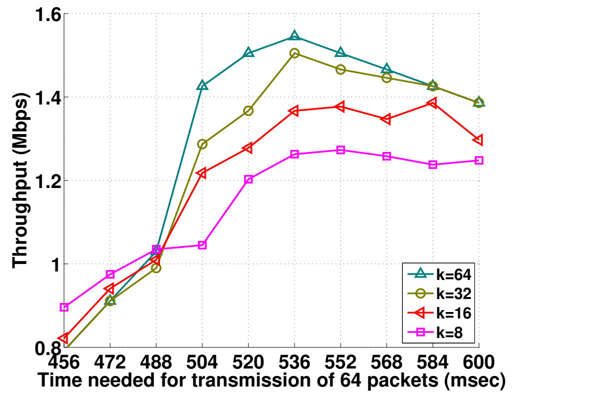
Lets assume that a video sequence, that of *foreman*, has to be forwarded from s to the destinations of $\mathcal{D}$. Assume also that this video is encoded with H.264 and has GOP size $g = 10$. The resolution is CIF and the quality of the H.264 compression (particularly the quantisation) is such, so that the average size of a compressed GOP to be almost equal to the batch size ($k/l \simeq 1$). In this case, for different values of frame rates f, our scheme utilises different time slots equal to $\tau = gk/fl = g/f$. If $f = 32$ fps, the required slot is $\tau = 10/32 = 312.5$ msec. In Figure 3(a), we see that the slotted mechanism of ViMOR delivers successfully one GOP (or one batch) per slot for $\tau = 312.5$ msec, since the throughput is equal to 2.408 Mbps or $2.408 \cdot 10^6/(8kb) = 3.2$ batches per second or $3.2 \cdot \tau = 1$ batch per slot. For less demanding videos with lower frame rates and higher slots $\tau > 312.5$ msec, the slotted mechanism succeeds again in delivering the GOP on the same duration 312.5 msec, meaning that for the rest of the time ($\tau - 312.5$ msec) the transmitters are idle and save power. In these cases, the throughput decreases, however, this is not a problem since we succeed on delivering each GOP on time. For lower slots $\tau < 312.5$ msec, the slotted mechanism does not always succeed in delivering the GOP. For example, if $\tau = 300$ msec, there is a probability 0.87 that a GOP will be delivered under the slotted mechanism, which means that the average throughput is almost $0.87/\tau$ batches per second or $(0.87/\tau) \cdot 8kb = 2.2$ Mbps, as it is depicted in the same figure. As it is illustrated, this probability of successful GOP delivery is decreasing as the slot decreases too.

On the contrary, the acknowledgment mechanism of MORE succeeds in delivering GOP on-time, when τ becomes greater than 350 msec. For $\tau \geq 350$ msec, its throughput is almost 2.3 Mbps, which is the maximum of the acknowledgment mechanism and it is less than the maximum of the slotted mechanism (2.4 Mbps). This happens because of the extra delay that the acknowledgment mechanism introduces. For $\tau > 350$ msec, the acknowledgment mechanism keeps the same high throughput, compared to the slotted one, enabling the pre-buffering of the video stream at the destinations. For $\tau = 312.5$ msec, there is a probability 0.21 that a GOP is delivered under the acknowledgment mechanism, meaning that the average throughput is almost $(0.21/\tau) \cdot 8kb = 0.5$ Mbps. As the slot decreases more and more, it is evident that the slotted mechanism achieves a remarkable improvement in performance compared to the acknowledgment one, delivering video in cases that the latter mechanism is completely inefficient ($\tau \leq 300$ msec). This is a significant result, as it enables the transmission of video sequences with higher quality, featuring high frame rates (high f) or high definition frames (high l) and thus requiring low slot duration $\tau = gk/fl$.

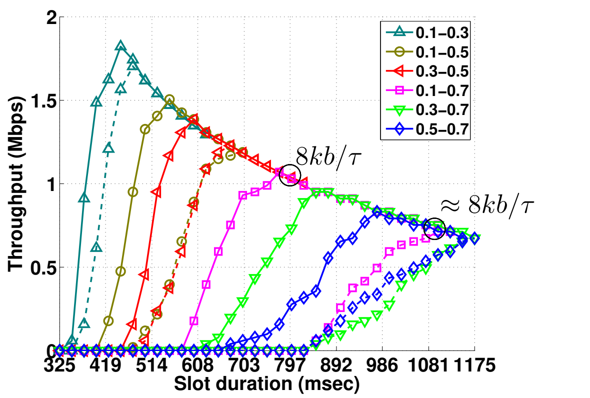
2) Evaluation of the transmission policy: In this set of experiments, our proposed transmission policy is evaluated by the configuration of the nodes connectivity and with the



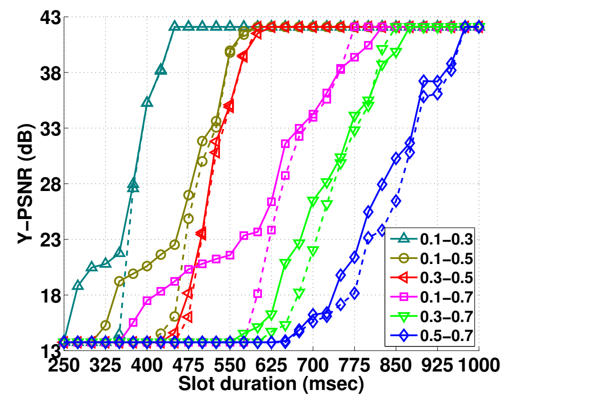
(a) The slotted mechanism of ViMOR compared to the acknowledgment mechanism of MORE for $k = 64$ and error transmission probabilities $e_1 = e_2 \approx 0.001$.



(b) The ViMOR's transmission policy performance for $k = 8, 16, 32, 64$ and error transmission probabilities $e_1 = 0.1$ and $e_2 = 0.5$.



(c) The ViMOR's transmission policy compared to the 50-50% equally distributed credit assignment for $k = 64$ and multiple e_1 and e_2 probabilities. The marker of each line indicates the e_1-e_2 values. The solid lines correspond to the ViMOR's assignment and the dashed lines to the 50-50% one.



(d) PLNC compared to RLNC for $k = 64$, $\alpha = 1/3$ and multiple e_1 and e_2 probabilities. The marker of each line indicates the e_1-e_2 values. The solid lines correspond to PLNC and the dashed lines to RLNC.

Fig. 3. Evaluating ViMOR in the 4-nodes experimentation topology of Figure 2(a).

application of the suggested credit assignment mechanism of Subsection III-B. At first, we configure the transmission error probabilities $e_1 = 0.1$ and $e_2 = 0.5$ in order to select the best k. After extensive experimentation with different e_1 and e_2 values we concluded to the selection of the aforementioned error probabilities with which we had observed the greatest differentiation in the performance of our proposed policy for various values of k. We have also seen that the throughput performance follows the same trend under all tested e_1, e_2 selections. Figure 3(b) depicts the performance of our proposed credit assignment for $k = 8, 16, 32, 64$. In the horizontal axis the time required for the delivery of a 64-packet sequence is represented, while on the vertical axis the measured throughput in Mbps is depicted.

As it is depicted, $k = 64$ is the optimal choice. Although $k = 64$ increases the length of the headers and imposes the largest overhead in the transmission of a packet, it gives us the ability for the most precise estimation of the credit values c, c_1 and c_2. For lower values of $k = 64$, there are fewer packets in each batch that need less time to be forwarded, thus the total credit c is decreased and the quantisation error introduced by its discrete estimation (credit is always integer) is proportionally higher. For example, if $k = 64$ and the batch has to be forwarded in time $\tau = 80$ msec, then $c < \lfloor \tau\rho/b \rfloor = \lfloor 0.08 \cdot 6 \cdot 10^6/(8 \cdot 1620) \rfloor = \lfloor 37.04 \rfloor = 37$, while if $k = 8$, then the smaller batch has to be forwarded in $\tau = 80/8 = 10$ msec and the total credit is $c < \lfloor \tau\rho/b \rfloor = \lfloor 0.01 \cdot 6 \cdot 10^6/(8 \cdot 1564) \rfloor = \lfloor 4.8 \rfloor = 4$. Obviously, for $k = 64$ the quantisation error is less than $1/38 = 2.63\%$, while for $k = 8$ it could be even $1/5 = 20\%$. On the other hand, higher values of $k = 64$ could enable even more accurate estimation of the credit values and be more throughput efficient, but we also conclude that for $k > 64$ the CPU utilisation of the nodes is increased rapidly, as we will see later in the related subsection. Consequently, for the rest of the presented experiments, we use $k = 64$ for the comparison between MORE and ViMOR.

TABLE III
CREDITS c AS A PERCENTAGE OF SLOT DURATION τ (IN MSEC) FOR PHYSICAL TRANSMISSION RATE $\rho = 6$ MBPS AND $k = 8, 16, 32, 64$

k	8	16	32	64
c	$\lfloor \tau \cdot 45.1\% \rfloor$	$\lfloor \tau \cdot 44.9\% \rfloor$	$\lfloor \tau \cdot 44.5\% \rfloor$	$\lfloor \tau \cdot 43.7\% \rfloor$

Our next step includes the examination of the potential performance gains offered by the proposed transmission policy, through the configuration of various pairs of values for the transmission error probabilities e_1 and e_2. Figure 3(c) depicts the performance of the credit assignment proposed in Algorithm 1 as compared to that of a naive, equally distributed credit assignment (50-50%), where $c_1 = c_2 = c/2$ independently of the e_1 and e_2 values. On the horizontal axis the slot duration τ in milliseconds is represented, while on the vertical axis, the achieved throughput in Mbps is represented. The solid lines correspond to the throughput performance of the ViMOR transmission policy, while the dashed lines correspond to the equally distributed assignment policy. The total credit c is initiated with a value depending on the duration of the slot τ, as mentioned before, and which is presented in Table III. The values of this table are estimated experimentally and show the total credit as a percentage of the slot τ, which is quite less than $\lfloor \tau\rho/b \rfloor$ due to the few time spent during the collisions that occur among the source and the relays. During the experimentation, the total credit is estimated again and again after the end of each slot. Each pair of solid and dashed lines of the same colour corresponds to a distinct pair of probabilities e_1-e_2. We should also mention that when swapping the e_1 and e_2 values, we obtain the same results for both the assignment policies, and thus we selected to present the results for $e_1 < e_2$ only.

Our rationale behind the comparison between the ViMOR and the 50-50% policy is that due to the inherent fairness of the 802.11 CSMA/CA mechanism the opportunities for transmission will be shared equally among the source and the relays in MORE. Consequently, despite the fact that MORE utilises its own policy for credit assignment, the outcome is exactly the same with that of the application of the 50-50% policy. In ViMOR, the *first-decode-then-transmit* policy introduces an indirect scheduling that allows us to reduce the contention/collisions and also forces the relays to not spend their credits in transmitting not fully decoded packets that contain less information. The inverse exponential curve that is not depicted in Figure 3(c), but all lines approach as the slot increases, is the plot of the maximum throughput that our scheme could achieve for each slot, equal to one batch per slot or $1/\tau$ batch per second or $8kb/\tau$ bps. When a line converges to this upper limit, the corresponding transmission policy delivers successfully all batches for these slots, while for lower slots there is a non-zero probability that the batch is not successfully delivered, leading to lower throughputs.

It is remarkable that our proposed policy achieves higher throughput in all the cases where $e_1 \neq e_2$. As expected, the throughput gains of the proposed policy are much greater in cases where $|e_1 - e_2|$ is high enough. Furthermore, it is worth mentioning that in cases where the credit is distributed equally, the performance is affected only by the link with the lowest quality, since it is equal for all pairs of probabilities that feature the same $\min(e_1, e_2)$. For example, if $e_1 = 0.1$ and $e_2 = 0.7$, then the proposed transmission policy achieves the highest throughput for all $\tau \geq 797$ msec, while at the same time the other policy features zero throughput, achieving the highest throughput for almost $\tau \geq 1081$ msec. The same worse performance of the equally distributed credit assignment happens whenever $e_1 = 0.1$, while our proposed credit assignment improves as e_2 decreases.

3) PLNC vs. RLNC: Finally, in this experimental set, the behaviour of the PLNC mechanism we proposed (described in Subsection III-C) is assessed by comparing it to the RLNC mechanism regarding the PSNR metric. We collect the videos received from all the destinations and for both mechanisms and we replace every late or lost frame with the previous frame, which could also be replaced by the previous one with the same logic, etc. For obvious reasons, the first frame is always provided to all the destinations during our experimentation. Subsequently, in the case, which is quite extreme, where no video frame is received on-time from a destination, the video perceived would correspond to a video sequence which consists of the first video frame being repeated to all the subsequent frames. Obviously, if a P/B-frame is received on time but the corresponding I-frame is lost, then the P/B-frame is useless. Figure 4 illustrates how we evaluate the received video stream using the PSNR metric to compare each displayed video frame with the corresponding one that would be displayed if all packets were received. The evaluation of the quality of a video sequence is based on the average PSNR of all its frames.

The experiments are conducted in links with configured transmission error probabilities, as we described in the previous experiments. The video sequence used is again that of *foreman* which is encoded with H.264, in resolution CIF, size of GOP $g = 10$ and there are not any B-frames, just I/P-frames. We configure the H.264 compression and in particularly quantisation, in such a way that the compressed GOP has an average size that almost equals the size of the batch ($k/l \simeq 1$), while every I-frame has a size that is roughly about the $\alpha = 1/3$ percentage of the whole GOP size. As we saw in our previous experimentation, for different frame rates f, our scheme utilises different time slots equal to $\tau = g/f$.

Our enhanced PLNC mechanism which gives priority to the decoding of the I-frames outperforms significantly the RLNC mechanism as it can be observed in Figure 3(d). On the horizontal axis the slot duration τ is represented in milliseconds, while on the vertical axis the video quality which is perceived in each of the destinations is represented as measured in terms of PSNR. When there is no batch reception the resulting PSNR stands at 13.4, which in that case is the lowest value, while on the other hand the maximum PSNR value that stands at 42.1 results from a video sequence that was received successfully without any lost frame. We can also notice that the gain of PLNC in terms of PSNR is high when the $|e_1 - e_2|$ is also high, a fact that we also observed in the previous experiment

Relaxing the *first-decode-then-transmit* policy: After replacing RLNC with PLNC, our first approach was to relax the *first-decode-then-transmit* policy at the relays. We recall at this moment that under the *first-decode-then-transmit* policy, the relays start transmitting when they have decoded the whole batch. Thus, we now enabled the relays to start transmitting after the decoding of only the most prioritised and important packets of every batch. We named this policy *partial-decode-then-transmit*. Although this policy seems to be more efficient in case where the delivery of the entire batch to the relay is impossible (enabling at least the forwarding of the most important packets), our experimental results show its very bad performance due to the increased contention between the source and the relay. As we see in Figure 5, the

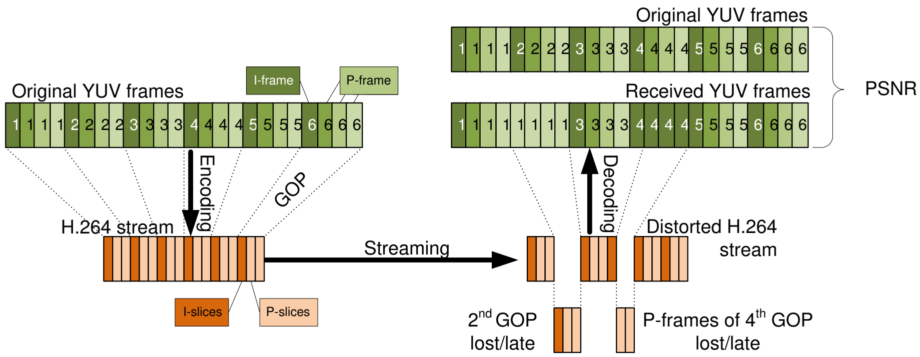


Fig. 4. PSNR evaluation of the received video stream.

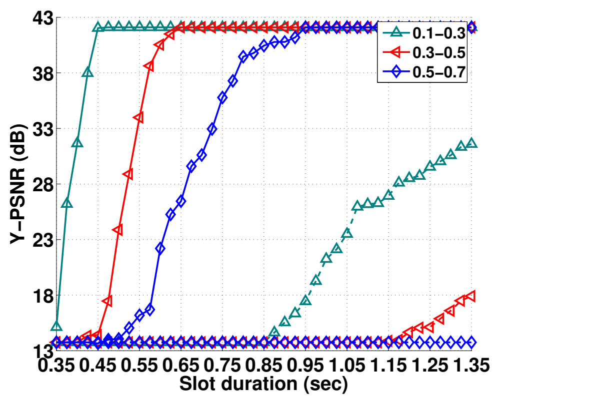


Fig. 5. The *first-decode-then-transmit* policy compared to the *partial-decode-then-transmit* one, for $k = 64$, $\alpha = 1/3$ and multiple e_1 and e_2 probabilities. The marker of each line indicates the e_1-e_2 values. The solid lines correspond to the first policy and the dashed lines to the second one.

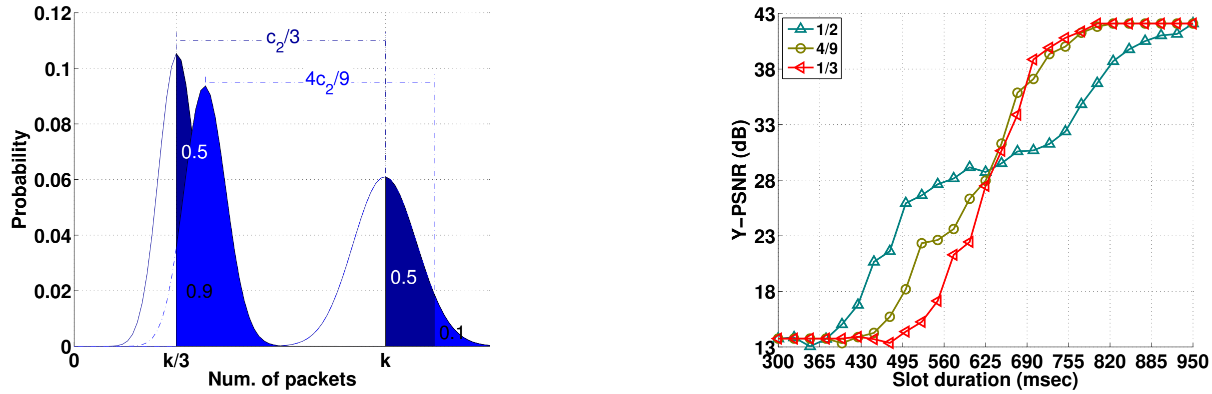
PSNR performance of the *partial-decode-then-transmit* policy is much worse than this of the *first-decode-then-transmit*, for all cases. Based on these results, we adopted the *first-decode-then-transmit* policy even together with PLNC.

Giving more credit to I-frames: According to the proposed PLNC mechanism, the credit c_2 in each relay is shared proportionally between the priority packet classes $\mathcal{O}$. More specifically, the credit given for the transmission of the o_h packets is $c_2^{o_h} = \alpha c_2$, since the packets of this class is $k_{o_h} = \alpha k$. In this section, we present our experimentation results on sharing non-proportionally the credit between the priority classes, giving more than αc_2 credit to the relays for the transmission of the o_h packets of the batch which are the most important. Following this approach, we increase the probability of successfully decoding of these packets at the destination, however, with the drawback that the probability of the entire batch successful decoding is reduced. This practice may offer better perceptual experience to the destinations, prioritising even more the packets of the I-frames, especially in case that the delivery of the whole batch has low probability.

In Figure 6(a), we see three Probability Distribution Functions (PDF). One is the probability distribution of delivering to each destination an exact number of o_l packets (right PDF) and the remaining two are the corresponding probability distributions for the most important o_h packets when $c_2^{o_h} = \alpha c_2 = c_2/3$ and $4c_2/9$ (two left PDFs). All PDFs are binomial, since each transmission is independent and successful with probability $2/3$. The slot duration is $\tau = 586$ msec, which means that $c = \lfloor \tau \cdot 43.7\% \rfloor = 256$ and $c_2 = 192$, with respect to Algorithm 1 and the fact that $e_1 \approx 0.001$ and $e_2 = 2/3$. The probability of successful delivery of the o_h packets is the cumulative probability from $k_{o_h} = k/3$ to infinity, since the number of received packets should be greater than $k/3$. When $c_{o_h} = c_2/3$, this probability is equal to 0.5 and it is shown in the figure with the dark blue area in the left PDF, while if $c_{o_h} = 4c_2/9$, the probability is 0.9 and it is depicted with the light blue area in the middle PDF. On the contrary, when $c_{o_h} = c_2/3$, the decoding of all o_l packets happens with probability 0.5 (depicted with the dark blue area in the right PDF), since the total received packets has to be greater than k, while if $c_{o_h} = 4c_2/9$, this probability changes to 0.1 because the total received packets has to be greater than $k(1-1/3)/(1-4/9) = 6k/5$. We observe that by enabling the relay to transmit more than αc_2 packets that are the linear combination of the most important packets, the probability of successful decoding of these packets increases from 0.5 to 0.9, however, with the disadvantage that the probability of decoding all packets decreases from 0.5 to 0.1.

In Figure 6(b), we see that by increasing the number of packets that relay produces as combinations of the most important packets, the destinations deliver better video quality for small slots, but worse quality for bigger slots. The best choice for a network operator to manage this trade off depends on the video sequence. Future work is to make the relay flexible to this choice, enabling the source to estimate the appropriate proportion of credit for the most important packets transmitted by the relays and propagate them this value.

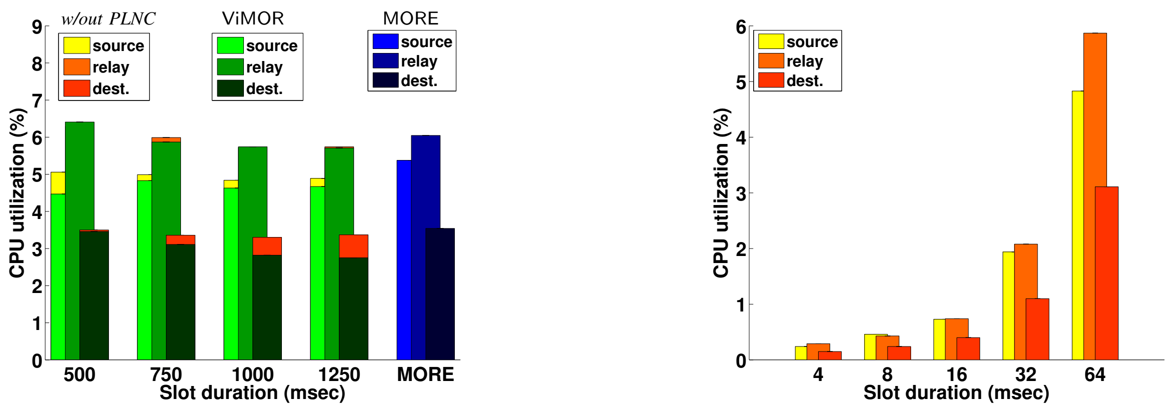
4) CPU utilisation: Our last set of experiments in this topology evaluates the performance of ViMOR and MORE in terms of CPU utilisation. In Figure 7(a), we show the CPU utilisation of each node during the whole video streaming for various slot durations. As we see, the CPU utilisation is quite similar under both protocols for small slots. However, as the slot increases, ViMOR requires less CPU cycles compared to MORE. The most CPU demanding process is this of the relays, while the source and destinations require approximately the 75% and 50% of the source's CPU cycles respectively. We also evaluated ViMOR with and without PLNC. It is highly remarkable that ViMOR with PLNC requires even less CPU cycles, since the extra encoding/decoding process does not increase the CPU usage, as we expected, but decreases it. The rationale behind this observation is that with PLNC, the number of computations that the encoding/decoding processes



(a) The Probability Distribution Functions (PDF) of delivering to each destination a specific number of o_l packets (right PDF) and o_h packets when $c_2^{o_h} = \alpha c = 2c_2/3$ or $c_2^{o_h} = 4c_2/9$ (two left PDFs) for slot $\tau = 586$ msec.

(b) The PSNR performance of ViMOR when the relay transmits $c_2^{o_h} = c_2/3$, $4c_2/9$ or $c_2/2$ packets as linear combinations of the most important o_h packets.

Fig. 6. The performance of ViMOR for $k = 64$, $\alpha = 1/3$ and error transmission probabilities $e_1 \approx 0.001$ and $e_2 = 2/3$.



(a) The CPU utilisation of the source, the relays and the destinations under MORE, ViMOR and ViMOR-*without PLNC*. The utilisation of all nodes is independent from the slot duration under MORE, while it is decreasing when the slot increases under ViMOR and ViMOR-*without PLNC*.

(b) The CPU utilisation of ViMOR for $k = 4, 8, 16, 32, 64$ and slot duration $\tau = 750$ msec.

Fig. 7. The CPU utilisation of the source, the relays and the destinations under MORE, ViMOR and ViMOR-*without PLNC*. The last scheme is ViMOR with the relays using RLNC instead of PLNC.

require is less than this of ViMOR-*without PLNC*, since some of the encodings/decodings involve a smaller set of initiative packets.

Figure 7(b) depicts the CPU utilisation of ViMOR for multiple values of $k = 4, 8, 16, 32, 64$. Comparing with Figure 3(b), we see a tradeoff between the throughput performance and the CPU utilisation of ViMOR. For higher values of k the achieved throughput increases, and this is why we choose $k = 64$. However, in case of devices with less CPU power (e.g. sensors), we remark that the protocol can be significantly less CPU-consuming by decreasing the value of k, in cost of the throughput performance.

B. Experimentation in the 7-nodes topology of Figure 2(b)

In the second class of our experiments we aim at providing a comparison between the performance of ViMOR and MORE in terms of PSNR, with an overall evaluation of all contributions (slotted mechanism, transmission policy and PLNC). We conducted the experiments in the 7-nodes topology of Figure 2(b), where source is s, $\mathcal{R} = \{r_1, r_2\}$ and $\mathcal{D} = \{r_1, d_1, d_2, d_3, d_4\}$. The nodes are arbitrary selected from the NITOS testbed, thus the packet losses are random. The two r_1 and r_2 nodes are relays, since they participate in the shortest-ETX tree connecting the source to all destinations. In particular, the shortest paths of d_1 and d_4 pass through r_1 and r_2 respectively, while the corresponding paths of d_2 and d_3 are going through r_1, since $0.4 \cdot 0.7 < 0.6 \cdot 0.9$ and $0.4 \cdot 0.9 < 0.6 \cdot 0.8$. Therefore, $p_1 = (0.4 + 0.6)/2 = 0.5$ and $p_2 = (0.6+0.7+0.8+0.9)/4 = 0.75$ (the average values of the corresponding transmission error probabilities). The credits c_1 and c_2 are given by Algorithm 1, using the probabilities p_1 and p_2 as well as the total credit c that depends on the slot duration τ. The other fixed variables remain the same as in the previous experiments. In particular, $k = 64$, $\rho = 6$ Mbps, $b = 1556 + k$ bytes and the video sequence is the same *foreman* encoded in H.264, with CIF resolution, GOP size $g = 10$ and compression configured in a way that the average size of a compressed GOP to be almost the same with the batch size ($k/l \simeq 1$), while every I-frame has a size that is roughly about the $\alpha = 1/3$ percentage of the whole GOP size.

In Figure 8, ViMOR obviously allows the r_1 node to perceive video of high quality in cases where $\tau > 0.6$ sec, while

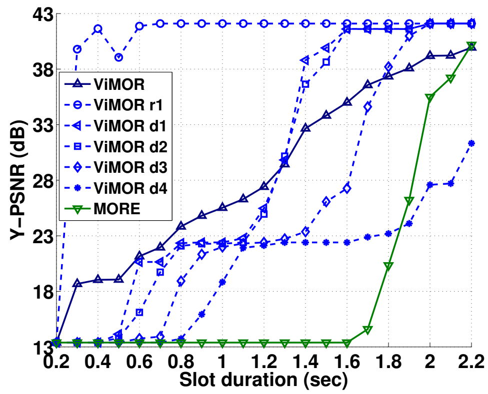


Fig. 8. Video performance comparison between ViMOR and MORE in the 7-nodes topology of Figure 2(b). The dashed lines correspond to the PSNR evaluation of the receipt video of each individual destination under ViMOR.

the rest two-hop destinations are able to receive an enjoyable quality of video after several slots. More specifically, the video stream received at all destinations has PSNR with value greater or equal to 22.4 for $\tau > 1.1$ sec. This video sequence corresponds to that where the I-frames are all successfully received while the P-frames are not received at all. This occurs when the high priority o_h packets, which include the I-frame of the corresponding GOP, of every forwarded batch are the only packets that have been decoded at the destinations. For duration of slot that is $\tau > 0.3$ sec, the PSNR averaged across all destinations is constantly increasing under ViMOR, while for MORE the corresponding PSNR increases only after slot $\tau > 1.7$ sec. Evidently, our proposed scheme, ViMOR, allows high quality video streaming, at least in some of the destinations, as the durations of slot are up to 5.3 times lower as compared to the ones of MORE. Furthermore, the gain in PSNR is close to 270% for duration of slot $\tau = 1.6$ sec, while the average gain is 175%.

V. Conclusion & Next Steps

ViMOR, the scheme presented in this paper, is the first, to the best of our knowledge, that succeeds in forwarding multicast video efficiently over wireless networks and is also the first implementation of a multicast OR scheme that is video-aware. Certainly, there are also other works proposing or simulating routing algorithms that consider some of our proposals, although there is not any of them which is actually implemented and evaluated on a realistic setup. As depicted by the PSNR gain which is up to 270% compared to the state-of-the-art algorithm of MORE, the potential of our research is very promising. However, there are still some open issues that require further investigation.

A basic issue that we need to consider is the introduction of a rate control algorithm that will allow for higher throughput and also better perceived video quality. Furthermore, part of our future work should be making the relays flexible in choosing of the number of transmitted packets, generated from the most important packets of the high priority class. Another point for further research is examining the impact of a higher number of priority classes, as well as considering various ways of sharing the credit c among the priority classes. It would also be interesting to design and evaluate algorithms for credit allocation between multiple flows, based on multiple criteria and goals. Last but not least, LTE/LTE-A is an appealing cellular access technology that promotes the use of relays (creating only two-hop paths) and seems to be an attractive research area for applying most of our concepts in ViMOR. The utilisation of multicast bearers and the capability for accurate coordination in LTE/LTE-A (since there are no contentions in the LTE technology for the medium usage) will improve the efficiency of our scheme. The link quality evaluation will be based on the inherent CQI mechanism of LTE, instead of using the ETX one of Roofnet. There are also some other open issues introduced by the LTE usage, while our ongoing research aims at tackling all these challenges.

Appendix
Proof: E is convex function of c_1

We first begin with the necessary notation, introducing some new variables. Let's assume that E_p stands for the probability of unsuccessful packet forwarding to the destination $d \in \mathcal{D}$ through a specific path $p \in \mathcal{P}^d$, where $\mathcal{P}^d$ denotes the set of the paths connecting the source s with destination d. Destination d does not receive the packet if all paths $\mathcal{P}^d$ fail to forward this, thus $E^d = \prod_{p \in \mathcal{P}^d} E_p$ express the probability of unsuccessful packet forwarding to the destination d. If we prove that E_p for all paths are positive and convex functions of c_1, then E^d is convex as well, since it is product of positive and convex functions [21]. Then, the average probability of successful packet delivery E, that is equal to $\sum_{d \in \mathcal{D}} (1 - E^d)/D$, is a convex function of c_1 again, since it is a linear combination of convex functions E^d.

Let's now assume that $e_{xy} \in (0, 1]$ denotes the probability of unsuccessful packet transmission over the link connecting node x to node y, thus e^z_{xy} is the corresponding probability of unsuccessful packet forwarding after z successive transmissions over this link. As follows, the probability of unsuccessful packet forwarding through an one-hop path $p' \in \mathcal{P}^d$ is

$$E_{p'} = e^{c_1}_{sd},$$

that is an exponential and convex function of c_1. The corresponding probability of a two-hop path $p'' \in \mathcal{P}^d$, that utilises a relay $r \in \mathcal{R}$, is a convex function of c_1 as well, equal to

$$\begin{aligned} E_{p''} &= 1 - (1 - e^{c_1}_{sr})(1 - e^{c_2}_{rd}) \\ &= 1 - (1 - e^{c_1}_{sr})(1 - e^{(c - c_1)/R}_{rd}), \end{aligned}$$

since its second derivative is always non-negative for $c_1 \in [k, c]$:

$$\begin{aligned} \partial^2 E_{p''} = &\underbrace{\ln(e_{sr})^2 e^{c_1}_{sr}(1 - e^{(c-c_1)/R}_{rd})}_{non-negative} \\ &+ \underbrace{\ln(e_{rd})^2 e^{(c-c_1)/R}_{rd}(1 - e^{c_1}_{sr})}_{non-negative}/R^2 \\ &+ 2\underbrace{\ln(e_{sr})\ln(e_{rd}) e^{c_1}_{sr} e^{(c-c_1)/R}_{rd}}_{non-negative}/R \geq 0. \end{aligned}$$

As follows, for all paths $p \in \mathcal{P}^d$ (both the one-hop and two-hop paths) the corresponding probabilities E_p are convex and positive functions of c_1, and this concludes the proof of the convexity of E over the same area.

REFERENCES

[1] K. Choumas, I. Syrigos, T. Korakis, and L. Tassiulas. Video-aware Multicast Opportunistic Routing over 802.11 two-hop mesh networks. In *Proc. IEEE SECON*, 2014.

[2] S. Biswas and R. Morris. ExOR: opportunistic multi-hop routing for wireless networks. In *Proc. ACM SIGCOMM*, 2005.

[3] S.-Y.R. Li, R.W. Yeung, and Ning Cai. Linear network coding. *IEEE Trans. on Information Theory*, 49(2):371–381, February 2003.

[4] Y. Yuan, H. Yang, S. H. Y. Wong, S. Lu, and W. Arbaugh. ROMER: Resilient opportunistic mesh routing for wireless mesh networks. In *Proc. IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[5] Pei Liu, Zhifeng Tao, S. Narayanan, T. Korakis, and S.S. Panwar. CoopMAC: A Cooperative MAC for Wireless LANs. *IEEE Journal on Sel. Areas in Comm.*, 25(2):340–354, February 2007.

[6] S. Chachulski (now Jakubczak), M. Jennings, S. Katti, and D. Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. In *Proc. ACM SIGCOMM*, 2007.

[7] S. Trifunovica, M. Kurantb, K.A. Hummela, and F. Legendrec. WLAN-Opp: Ad-hoc-less opportunistic networking on smartphones. *Elsevier Ad Hoc Networks*, 25:346–358, 2015.

[8] Open Garden. https://www.opengarden.com/.

[9] O. Turkes, H. Scholten, and P.J.M. Havinga. BLESSED with Opportunistic Beacons: A Lightweight Data Dissemination Model for Smart Mobile Ad-Hoc Networks. In *Proc. ACM MobiCom Workshop on Challenged Networks (CHANTS)*, 2015.

[10] Y. Lin, B. Li, and B. Liang. CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding. In *Proc. IEEE ICNP*, 2008.

[11] D. Koutsonikolas, C. Wang, and Y.C. Hu. CCACK: Efficient Network Coding Based Opportunistic Routing Through Cumulative Coded Acknowledgments. In *Proc. IEEE INFOCOM*, 2010.

[12] D. Koutsonikolas, Y. C. Hu, and C. Wang. Pacifier: high-throughput, reliable multicast without "Crying babies" in wireless mesh networks. *IEEE/ACM Trans. on Networking*, 20(5):1375–1388, October 2012.

[13] Z. Li, L. Sun, X. Zhou, and L. Li. Priority Linear Coding Based Opportunistic Routing for Video Streaming in Ad Hoc Networks. In *Proc. GLOBECOM*, 2010.

[14] S. M. Das, H. Pucha, K. Papagiannaki, and Y.C. Hu. Studying wireless routing link metric dynamics. In *Proc. ACM Internet Measurement Conference (IMC)*, 2007.

[15] J. Bicket, D. Aguayo, S. Biswas, and R. Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *Proc. ACM MobiCom*, 2005.

[16] X. Zhao, J. Guo, C.T. Chou, A. Misra, and S.K. Jha. High-Throughput Reliable Multicast in Multi-Hop Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, 14(4):728–741, April 2015.

[17] O. Alay, T. Korakis, Y. Wang, E. Erkip, and S. S. Panwar. Layered wireless video multicast using omni-directional relays. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[18] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click Modular Router. *ACM Trans. on Computer Systems*, 18(3):263–297, August 2000.

[19] Source code of the MORE elements. http://people.csail.mit.edu/szym/more/more-release-one.tgz.

[20] NITLab: Network Implementation Testbed Laboratory. http://nitlab.inf.uth.gr/NITlab/index.php/testbed.

[21] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Kostas Choumas received the Diploma and M.S. degrees in electrical and computer engineering from the University of Thessaly, Volos, Greece, in 2007 and 2008, respectively, and the Ph.D. degree in the same department, in 2015, under the supervision of Assistant Professor Thanasis Korakis and Professor Leandros Tassiulas. From 2015 until now, he is Postdoctoral Associate with the University of Thessaly. His research interests include the design and implementation of enhanced algorithms on wireless networks, as well as the experimentation on these algorithms performance using testbed platforms.

Ilias Syrigos received the Diploma and M.S. degrees in electrical and computer engineering from the University of Thessaly, Volos, Greece, in 2012 and 2014, respectively. He is a Ph.D. candidate at the same department under the supervision of Assistant Professor Thanasis Korakis. His research interests involve the design and implementation of mechanisms for the detection of wireless networks' pathologies along with the development and experimentation of rate adaptation algorithms for wireless networks.

Thanasis Korakis received the B.S. and M.S. degrees in computer science from the University of Athens in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the University of Thessaly, Greece, in 2005. In the summer of 2004, he was a Visiting Researcher with the CSE Department, University of California at Riverside. From 2005 to 2006, he was a Research Scientist with the ECE Department, Polytechnic University, NY, where he was a Research Assistant Professor from 2006 to 2012. He is currently an Assistant Professor with the ECE Department, University of Thessaly. His research interests are in the field of networking with an emphasis on access layer protocols, cooperative networks, quality-of-service provisioning, network management, and experimental platforms. From 2007 to 2012, he was a Voting Member of the IEEE 802.16 Standardization Group. He served as a Publication Chair for WiOpt 2007, a TPC Chair of WiNTECH 2010 and Tridentcom 2011, a General Chair of Tridentcom 2012, and a Chair of EMUTools 2013 and WINMEE 2016. He received several awards, including the best paper awards in WiNTECH 2013, GREE 2013, and CloudComp 2015.

Leandros Tassiulas (S?89?M?91?SM?05?F?07) received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 1991. He has been a Faculty Member with the NYU Polytechnic School of Engineering, Brooklyn, NY, USA, the University of Maryland, College Park, and the University of Thessaly, Volos, Greece. He is currently the John C. Malone Professor of Electrical Engineering with Yale University, New Haven, CT, USA. His most notable contributions include the max-weight scheduling algorithm and the back-pressure network control policy, opportunistic scheduling in wireless, the maximum lifetime approach for wireless network energy management, and the consideration of joint access control and antenna transmission management in multiple antenna wireless systems. His research interests include computer and communication networks with an emphasis on fundamental mathematical models and algorithms of complex networks, architectures and protocols of wireless systems, sensor networks, novel internet architectures, and experimental platforms for network research. He was a recipient of several awards, including the IEEE Koji Kobayashi Computer and Communications Award, the Inaugural INFOCOM 2007 Achievement Award for fundamental contributions to resource allocation in communication networks, the INFOCOM 1994 Best Paper Award, the National Science Foundation (NSF) Research Initiation Award (1992), the NSF CAREER Award (1995), the Office of Naval Research Young Investigator Award (1997), and the Bodossaki Foundation Award (1999).