Σχεδιασμός, Υλοποίηση και Αξιολόγηση ενός Multicast Πρωτοκόλλου Δρομολόγησης Βίντεο σε Ασύρματα Δίκτυα

από τον

Ηλία Συρίγο

Μεταπτυχιακή εργασία στα πλαίσια του μεταπτυχιακού προγράμματος
"Επιστήμη και Τεχνολογία Ηλεκτρολόγου Μηχανικού και Μηχανικού Υπολογιστών"
του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας
2014

Επιβλέπων:

Αθανάσιος Κοράκης, Επίκουρος Καθηγητής

Συνεπιβλέποντες:

Λέανδρος Τασιούλας, Καθηγητής
Αντώνιος Αργυρίου, Λέκτορας

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της μεταπτυχιακής μου εργασίας κ. Αθανάσιο Κοράκη, Επίκουρο Καθηγητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας, για την ευκαιρία που μου έδωσε να εργαστώ πάνω σε ένα ιδιαίτερα ενδιαφέρον θέμα και για την εμπιστοσύνη που μου έδειξε όλον αυτό τον καιρό.

Επίσης, θα ήθελα να ευχαριστήσω τους κυρίους Αντώνη Αργυρίου και Λέανδρο Τασιούλα, Λέκτορα και Καθηγητή αντίστοιχα του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας, για την τιμή που μου έκαναν να συνεπιβλέπουν την μεταπτυχιακή μου εργασία.

Ακόμη, θα ήθελα να ευχαριστήσω ιδιαίτερα τον Κωνσταντίνο Χούμα, Υποψήφιο Διδάκτορα του Τμήματος μας για την πολύ καλή συνεργασία που είχαμε, την υπομονή που επέδειξε, την εμπιστοσύνη που μου έδειξε στα δύσκολα κομμάτια της εργασίας και την σωστή καθοδήγηση καθ' όλη τη διάρκεια της.

Θέλω να ευχαριστήσω επίσης όλα τα παιδιά από το NITLab για τη συνεργασία τους σε καθημερινή βάση καθώς και για την προθυμία τους να επιλύσουν οποιαδήποτε προβλήματα προέκυπταν κατα τη διάρκεια της υλοποίησης της εργασίας μου. Ιδιαίτερα ευχαριστώ τον καλό μου φίλο, Βιργίλιο Πασσά, για τη συμπαράσταση του καθ' όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Τέλος, ευχαριστώ από τα βάθη της καρδιάς μου την οικογένεια μου για την αγάπη, την καθοδήγηση, την υπομονή και τη στήριξη της όλα αυτά τα χρόνια.

TABLE OF CONTENTS

ΕΥΧΑΡΙΣΤΙΕΣ ii

LIST OF FIGURES v

ΠΕΡΙΛΗΨΗ vii

ABSTRACT viii

CHAPTER

I. Introduction 1

1.1 Introduction 1

II. Related Work 4

2.1 Related Work 4

III. ViMOR Design 7

3.1 ViMOR Design 7
3.1.1 Rejection of the acknowledgment mechanism 11
3.1.2 Redesign of the transmissions policy 11
3.1.3 Classification and prioritization of the video packets . . . 14

IV. Implementation on Click Modular Router 17

4.1 Click Modular Router 17
4.2 ViMOR implementation 19
4.3 Click configuration 20
4.3.1 Input Path 20
4.3.2 Output Path 21

V. Experimentation Results 23

5.1 Experimentation Results 23
5.1.1 First class of experiments 24
5.1.2 Second class of experiments 31

VI. Conclusion 33

6.1 Conclusion 33

APPENDICES 34
A.1 Convexity of Algorithm 1 35

BIBLIOGRAPHY 37

LIST OF FIGURES

Figure

3.1 A topology where all destinations are at most two-hop away from source. . . 9

3.2 The main differences between MORE and ViMOR. In MORE, the source and the relay are competing for the medium during the whole period of the batch forwarding, which ends when the source receives the acknowlegment. The source has the same transmission opportunities with the relay, although it is more susceptible to transmission errors and needs more. ViMOR replaces the acknowlegment with a time counter (gaining the time spent for the acknowlegment forwarding). It gives more transmission opportunities to the source and disables the collisions between the source and the relay by applying the *first-decode-then-transmit* policy. Finally, the relay uses PLC enabling the forwarding of the I-frame even if the P-frame is lost. 15

4.1 Input path of ViMOR's Click configuration. 21

4.2 Output path of ViMOR's Click configuration. 21

5.1 Two different topologies with 4-nodes (a) and 7-nodes (b) used in our algorithm evaluation. 24

5.2 Evaluating ViMOR in the 4-nodes experimentation topology of Figure 5.1(a). 25

5.3 PSNR evaluation of the received video stream. 28

5.4 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 values. The marker of its line indicates the e_1 - e_2. The solid lines correspond to the first policy and the dashed lines to the second one. 28

5.5 The performance of ViMOR for $k = 64$, $\alpha = 1/3$, $e_1 \approx 0.001$ and $e_2 = 2/3$. 30

5.6 Video performance comparison between ViMOR and MORE in the 7-nodes topology of Figure 5.1(b). The dashed lines correspond to the PSNR evaluation of the receipt video of each individual destination under ViMOR. 31

ΠΕΡΙΛΗΨΗ

Σχεδιασμός, Υλοποίηση και Αξιολόγηση ενός Multicast Πρωτοκόλλου Δρομολόγησης Βίντεο σε Ασύρματα Δίκτυα

από τον

Ηλία Συρίγο

Στη μεταπτυχιακή αυτή εργασία, προτείνουμε και χαρακτηρίζουμε την απόδοση ενός καινοτόμου αλγορίθμου Οπορτουνιστικής Δρομολόγησης (ΟΔ), προσαρμοσμένου στις ανάγκες του video, για multicast σε ασύρματα δίκτυα 802.11. Η ΟΔ εκμεταλλεύεται την broadcast φύση της ασύρματης επικοινωνίας και προσαρμόζεται πολύ καλά στο ασύρματο περιβάλλον που συχνά προκαλεί απώλειες. Στην εργασία αυτή προεκτείνουμε τον πλέον αναγνωρισμένο αλγόριθμο ΟΔ, ονομαζόμενο MORE, ο οποίος προσφέρει υποστήριξη για multicast αλλά δεν μπορεί να εφαρμοστεί αποτελεσματικά σε μετάδοση video. Ο μηχανισμός μας μπορεί να υποστηρίξει εφαρμογές σε πραγματικό χρόνο με αυστηρούς χρονικούς περιορισμούς. Βελτιώνουμε τη λαμβανόμενη ποιότητα video των τελικών χρηστών κατηγοριοποιώντας και δίνοντας προτεραιότητα στη video κίνηση και ενορχηστρώνουμε αποδοτικά τους πολλαπλούς εκπομπούς που συμμετέχουν στη multicast δρομολόγηση. Για την αξιολόγηση του προτεινόμενου σχήματος, διεξάγαμε πειράματα σε ένα μεσαίας κλίμακας ασύρματο testbed. Τα αποτελέσματα δείχνουν ότι ο προτεινόμενος αλγόριθμος αυξάνει τη λαμβανόμενη ποιότητα video έως και 270% η κατά 175% σε μέσο όρο, σε σχέση με τον MORE αλγόριθμο.

ABSTRACT

Design, Implementation and Evaluation of a Video Specific Multicast Routing Protocol over Wireless Network

by

Ilias Syrigos

Opportunistic Routing (OR) exploits the inherent broadcast nature of the wireless communication and adapts very well to the lossy wireless environment, especially in case of multicast. In this thesis, we propose and characterize the performance of a novel video-aware OR algorithm for multicast used in 802.11 two-hop mesh networks. We extend a state of the art OR scheme, namely MORE, that offers multicast support but is not efficiently applicable to video streaming applications. Our scheme is able to support real-time applications with hard time-constraints. We improve the video-perception quality of the end users by classifying/prioritizing the video traffic and efficiently orchestrating the multiple transmitters involved in multicast routing. In order to evaluate the proposed scheme, we conducted experiments in a medium-scale wireless testbed. Our results show that the proposed scheme increases the average video-perception quality by up to 270% in some cases or up to 175% in average, compared to the MORE algorithm.

CHAPTER I

Introduction

1.1 Introduction

As the need for Internet access has grown enormously nowadays, wireless connectivity seems to be the most appropriate solution for low-cost and efficient network coverage. Deployment of wireless networks is more affordable compared to the past and the available speeds are now up to 600 Mbps (802.11n), rendering the wireless access as the most appropriate option for physical interconnection. Although the wired access seems to be more stable and high-bandwidth, it is impracticable in cases of public areas with mobile devices that opportunistically arrive or leave. The recent proliferation of these devices (laptops, tablets, smartphones) has been instrumental in bridging even more the improvement of the wireless access to the attention of networking researchers. There are multiple open issues and challenges in order to stream efficiently over the wireless medium. For example, the broadcast nature of the wireless medium should be exploited in cases that devices are delivering the same multicast stream.

Representative examples are large scale events in public areas (audio concerns, football matches, airports, etc.), where the majority of the requests made is for the same multicast and real-time stream. Although the wireless access can backhaul this kind of scenarios, the deployment of many wireless gateways in order to cover the whole area is often impossible [16]. The exploitation of easily placed wireless relays could fill in the gap between the gateways' coverage subareas. Moreover, the collaborative processing and retransmission of overheard information at some end devices, could also make them play the role of wireless relays and create spatial diversity and throughput improvement for all devices. In this case, the routing from the gateway to each end device is the most challenging issue.

ExOR [4] is the first Opportunistic Routing (OR) protocol for wireless mesh networks that takes advantage of the wireless broadcast nature and does not follow the traditional routing approach of choosing the best sequence of relays between the gateway and each device. It creates cooperative diversity, leveraging broadcast transmissions in order to send information through multiple relays concurrently. MORE [6] is an enhanced version of the ExOR protocol, supporting also multicast traffic and utilizing Network Coding (NC) [9] to improve throughput up to three times. Moreover, MORE is a MAC independent protocol as compared to ExOR, running directly on top of 802.11 CSMA/CA instead of the strict scheduler that ExOR deals with. 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.

In this work we extend and fine tune the work made in MORE in order to meet and satisfy the necessities and requirements of video multicast. It is worth to mention that *video traffic should be delivered without delay, even if this implies that some information may get lost.* Inferring from the above, we introduce the Video-aware Multicast Opportunistic Routing protocol (ViMOR) [7], focusing in topologies where the end devices are one-hop or two-hop away from a gateway. In particular, our augmentation is threefold: i) support for time-constrained routing process, ii) enhancements to the transmissions policy regarding the opportunistic selection of the relays and the orchestration of the transmission opportunities of the gateways and the relays and iii) video-perception quality improvement by classifying and prioritizing the video traffic. In contrast to MORE, ViMOR addresses the demanding video challenges, enjoys high throughput performance and increases the quality of the video perception in all end devices of each multicast group.

The rest of the thesis is organized as follows. In Chapter II we introduce related work. Chapter III introduces OR concepts and provides the design and the keystones of the proposed scheme. The implementation of our scheme is discussed in Chapter IV. In Chapter V we evaluate the performance of the proposed protocol by conducting appropriate experiments in a wireless testbed. We conclude in Chapter VI.

CHAPTER II

Related Work

2.1 Related Work

ExOR [4] was proposed by Biswas and Morris, introducing the OR approach. OR belongs to a general class of wireless algorithms that exploit the broadcast nature of the wireless transmission, utilizing the overheard information at multiple nodes to increase wireless throughput. These algorithms could either relay the received signal acting as a multi-antenna system, or combine the bits received at different nodes to correct wireless transmission's errors [10], or optimize the choice of the next relay from the nodes that received a transmission. ExOR belongs to the third category and was the first OR implementation that demonstrated cases, where the more relaxed choice of next-hop achieves significant throughput gains. More specifically, in ExOR the source separates the packets in batches in order to send them collectively. Then, it does not try to send the packets of each batch to a specific next-hop host (expecting for an acknowledgment), but broadcasts the packets for a specific number of retries, and each potential receiver also retransmits them for a specified number of times, until the destination finally receives the whole batch and sends a batch acknowledgment. The scheduling of the transmissions among the source and the potential relays is based on a modified MAC layer, that specifies the intervals when nodes send their packets avoiding contentions/collisions.

MORE [6] is the enhanced version of ExOR, introducing a NC approach that randomly mixes packets before forwarding them. The source and the relays do not forward the identical packets of the batch, but linear combinations with arbitrary multipliers of the original packets. The newly generated packets have the corresponding multipliers encapsulated in a specific header, thus reproducing the original packets in destinations is feasible by executing the inverse process. The scheduling of the transmissions of all involved nodes is arbitrary, based on the 802.11 CSMA/CA, making the protocol MAC-independent and more easily applied. However, even under the impact of the resultant contentions/collisions, the throughput performance of MORE is significantly better than that of ExOR. It is also worth to mention that MORE, in the same way that ExOR does, enforces the source and the relays to retransmit until the destination successfully sends an acknowledgment. The main difference with ExOR is that MORE imposes the source to transmit continuously, while each potential relay has a credit value, which is the number of transmissions that it will attempt for each received packet. Finally, the architecture of MORE makes the multicast case a natural extension of the unicast one, in comparison to ExOR that supports only unicast.

Some open issues and weaknesses of the MORE protocol regarding the multicast case have been addressed in some other works [14, 12, 11]. For example, in MORE the source requires an acknowledgment from each destination of the multicast group before proceeding to the next batch, resulting in performance degradation for some receivers if others exist that have poor connections. Pacifier [11] addressed this weakness of MORE and suggested a round-robin mechanism that enables the source to move to the next batch every time that one receiver acknowledges the current batch. After proceeding with a predefined number of batches, the source will repeat the transmission process for each of the previous batches to finally receive acknowledgments from all multicast destinations. This is a very interesting approach, since it suppresses the annoying variation on the batch forwarding duration of MORE. However, it does not succeed in eliminating this phenomenon, since it targets again at 100% reliable forwarding. To the best of our knowledge, ViMOR is the first scheme that introduces the total denial of the acknowledgment mechanism, redesigning appropriately the transmissions policy and enabling a time-constrained forwarding process.

OR-PLC [13] is another work that focus on video traffic, enabling the partial reproduction of a batch, when the full reproduction is not feasible yet. Instead of using the Random Linear Coding (RLC) of MORE and Pacifier, this work introduces a Priority (or progressive) Linear Coding (PLC) to mitigate the error propagation and provide high bandwidth utility. More specifically, with OR-PLC the source generates some network coded packets as a linear combination of only the most important original packets, that correspond to video intra-frames. The intra-frames are encoded by only removing spatial redundancy in the frame, while inter-frames are encoded by removing temporal redundancy in successive frames. The loss of an intra-frame is much more crucial than the loss of an inter-frame, since the intra-frame is also required for decoding all successive frames. PLC enables the earlier retrieval of the intra-frames comparing to RLC, even if some inter-frames get lost, provisioning at least a low quality video sequence to a poorly connected destination. However, OR-PLC adopts the same acknowledgment mechanism with MORE. ViMOR implements PLC and evaluates its efficiency, when it is activated in parallel with the aforementioned video-aware extensions.

CHAPTER III

ViMOR Design

3.1 ViMOR Design

In this work, an innovative and enhanced multicast OR protocol is proposed, extending the MORE philosophy to adapt to the video traffic requirements. Video multicast streaming uses UDP that suffers from unreliable streaming comparing to TCP, increasing the negative effect of the high probability of packet loss in wireless networks. We strongly believe that OR integrates well with the video traffic requirements and the packet loss environment, since in case of video streaming, delivering on-time is of greater importance than delivering reliably. In case of traditional routing, the duration of each wireless transmission cannot be easily estimated, since the occasional but not rare variations of channel conditions may cause an unknown number of MAC retransmissions, until the MAC acknowledgment is successfully received. Subsequently, the time of a packet forwarding process through a specific route is unpredictable and may exceed the time constraints of a specific video sequence, since it is equal to the aggregate duration of the individual time-varying transmissions.

On the other hand, in case of OR, the transmissions are broadcasted without MAC retransmissions and acknowledgments, enabling the duration of the packet forwarding process to be upper limited, depending only on the controlled number of transmissions that the source and each relay attempts. Subsequently, OR does not provide reliability in packet delivery, since there are no MAC acknowledgments, but as we already mentioned this is of less importance in case of video streaming. It is worth to mention that some OR algorithms, like MORE, implement an application layer acknowledgment mechanism to provide reliability in cost of their capability for time constrained streaming, coping with similar inconvenience with the traditional routing.

Based on the aforementioned analysis, we propose a new OR protocol based on the design of MORE, named Video-aware Multicast Opportunistic Routing (ViMOR), and we summarize its main differences as compared to the MORE protocol:

- **Rejection of the acknowledgment mechanism** since the video traffic should be delivered on-time and not necessarily reliably.
- **Redesign of the transmissions policy** concerning the scheduling and the number of transmissions that the source and the relays perform.
- **Classification and prioritization of the video packets** according to their content, adopting an enhanced NC policy.

Due to the first two differences, ViMOR achieves high throughput video streaming satisfying the video requirement for maximum time duration of packet forwarding process and giving more transmission opportunities to the most error-susceptible wireless links. The third one improves even more the video streaming performance by enhancing the quality of the delivered video, increasing the probability of successful delivery of the most important video packets.

As it is depicted in Figure 3.1, ViMOR focuses on multicast scenarios where **all destinations are at most two-hop away from the source**. The rationale behind this decision is twofold: i) the performance of video wireless streaming over paths of three or more hops is degraded due to the fluctuations that increase as the paths get longer, and ii) the source is not able to apply the transmissions policy if it serves more than two-hop away destinations, since it requires the link evaluations that should be on-line and updated. At this point, it is useful to mention that MORE supports broader topologies, however, based on off-line link evaluations

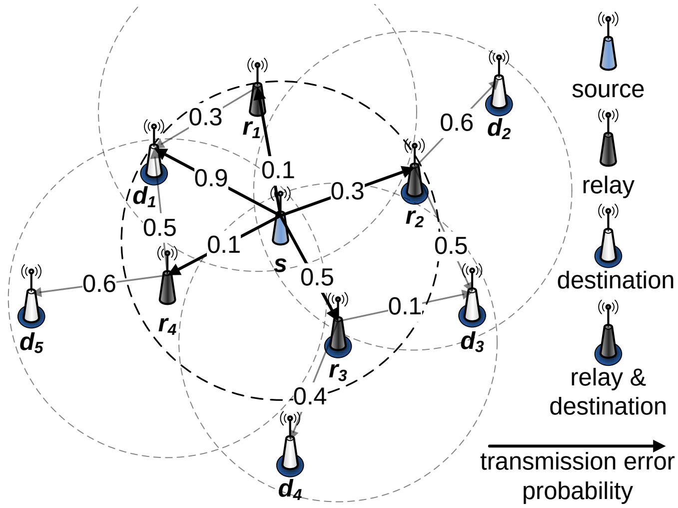


Figure 3.1: A topology where all destinations are at most two-hop away from source.

that were collected in the past. It is infeasible for one central point to gather on-line measurements in these broader topologies with more than two-hop away destinations. This feature of MORE's design is not desirable, since studies have shown that link metrics are sensitive and should be frequently updated [8].

On the other hand, a mechanism inspired by the ETX estimation algorithm of Roofnet [3] is able to provide on-line link evaluations for the aforementioned topologies of our focus. More specifically, this mechanism enforces nodes to periodically send broadcast packets, estimate the number of the corresponding received packets from each neighbor and report these numbers among them. Through this process, each node calculates the transmission error probabilities of its adjacent links, while a periodical report informs its neighbors about these evaluations. At the end, every node (including the source node that applies the transmission policy) knows the quality of its adjacent links and its neighbors' adjacent links. It is worth to mention that the flooding mechanism and the statistics from previous packets going to the reverse direction, which are used by a Roofnet node to evaluate the links that are more than two-hop away, are impracticable for every multicast and single-source algorithm, like MORE and ViMOR.

Before proceeding, we introduce some notations further explaining the key points of the NC policy adopted by both ViMOR and MORE. They are summarized in Table 3.1, together

Table 3.1: Variables description

Variable	Description
s, $\mathcal{R}$, $\mathcal{D}$	source and sets of relays and destinations respectively
N	total number of nodes
k	number of initial packets included in a batch
b	packet size (payload and headers)
ρ	utilized basic physical transmission rate
f, g	video frame ratio and number of GOP frames respectively
l	packets needed for a GOP transmission
τ	time given for the forwarding of one batch (slot)
c	total number of transmissions in a slot (total credit)
c_1, c_2	credits of source and each relay respectively
E	average prob/ty of unsuccessful packet delivery among all destinations $d \in \mathcal{D}$
$\mathcal{O}$	packet classes with different priority
k_o	number of class $o \in \mathcal{O}$ packets in a batch
o_h, o_l	high and low priority classes including intra-frames and all frames respectively
α	the intra-frames size proportion of the whole batch size

with all other notations that will be introduced later. Regarding a single multicast stream, imagine a source s that is supported by a set $\mathcal{R}$ of R relays and serves a set $\mathcal{D}$ of D destinations. The network consists of $N = |\mathcal{R} \cup \mathcal{D}| + 1$ nodes. In both routing schemes, source s breaks up the stream to *batches* of k equal-sized packets of size b. Each time the source forwards a batch, it generates and transmits broadcast packets that are linear combinations of the k initial batch packets. The coefficients of each linear combination are encapsulated to the corresponding generated packet. Once a relay $r \in \mathcal{R}$ receives a packet, it linearly combines this packet with the previously received ones of the same batch and forwards the generated packet for transmission. When a destination $d \in \mathcal{D}$ receives k linearly independent packets, it is able to decode the batch and retrieve the k initial packets of this. Both source and relays utilize the basic/lowest physical rate ρ for all packet transmissions, in order to extend as much as possible their coverage areas.

The following Subsections 3.1.1, 3.1.2 and 3.1.3 will explain further the outlines of ViMOR differentiation, as compared to MORE.

3.1.1 Rejection of the acknowledgment mechanism

In MORE, an acknowledgment mechanism gives a signal to the source for the expiration of a batch forwarding and the initiation of a new one. More specifically, during a batch forwarding process, the source and the relays generate and transmit packets continuously and for an unlimited number of times, until source receives an application layer acknowledgment from each of the involved destinations. It is obvious that this mechanism cannot provide any guarantee for maximum time duration of a batch multicast forwarding.

On the other hand, ViMOR overcomes this challenge enforcing the source and the relays to transmit for a fixed number of times. The source does not wait for an acknowledgment, but keeps a timer and the batch forwarding is limited within a specified time period, called *slot*. The slot duration is estimated by the source, according to the video stream characteristics. After the expiration of the slot interval, the source proceeds to the next batch. Assuming that a video stream features a frame ratio f, a Group Of Pictures (GOP) with g frames should be delivered in a time interval equal to g/f. So, if a GOP needs l packets or l/k batches to be encapsulated, then the forwarding of one batch should be completed during a slot $\tau = (g/f)/(l/k) = gk/fl$.

The slotted mechanism does not provide reliability in batch forwarding, but every batch that is successfully delivered is always on-time. As we already mentioned before, this is a desirable feature, since it is a waste of time and energy for source and relays to keep forwarding a batch, that is already obsolete and useless for the destinations.

3.1.2 Redesign of the transmissions policy

The second most important difference in ViMOR's approach is the enhanced transmissions policy. In MORE, as it is already mentioned in Section **??**, source generates and transmits packets continuously and for unlimited number of times before proceeding to the following batch. Once a node receives a packet, it generates and transmits a number of new packets equal to its assigned *credit*, which is estimated by taking into account the quality of all network links.

Each node that is ``charged'' with a non-zero credit is a potential relay. In ViMOR, the credit of a node is interpreted in a different way, representing the number of packet transmissions this node will attempt during a batch forwarding, independent of the number of the received packets. The aggregate credit of source and relays is upper bounded by a c integer value that depends on the utilized slot τ, since the number of transmissions that can be performed in a slot interval is obviously limited by $c < \rho\tau/b$. Source initially estimates the value of $c = \lfloor \rho\tau/b \rfloor$ based on the other known parameters ρ, τ and b. Later, it hears the transmissions happened at the past slots and estimates again a more accurate value of c, based on this history. Outside interference and unmodelled factors in wireless transmissions are the main reasons for the c variation.

Actually, ViMOR adopts a new transmissions policy that is presented below and aims at increasing the individual throughput of each one-hop or two-hop away destination host, **maximizing the average probability of successful batch reception among all destinations**. From now on, assume that E is this probability. The challenges appear in i) selecting the most appropriate one-hop relays; and ii) charging source and these relays with suitable credits. Regarding the relays selection, the source can either choose them or utilize a fixed and dedicated set of relays. In case that the set of relays is not fixed and predefined, $\mathcal{R}$ is retrieved by source building a multicast tree that connects the source to all two-hop away destinations. The tree is similar to that of Pacifier and is a shortest-ETX tree, constructed at the source by taking the union of all shortest-ETX paths to the two-hop away destinations. At the end, the set of relays $\mathcal{R}$ consists of all one-hop connected nodes to the source, belonging in this tree.

To overcome the second challenge, regarding the credit charging, we need to address two orthogonal sub-challenges. The first has to do with providing the source with the highest possible credit, equal to $c_1 \geq k$, in order to satisfy all one-hop away destinations, while the second aims at sharing appropriately the credit c among source and relays, providing also a credit $c_2 \geq k$ to each relay, in a way that satisfies the two-hop away destinations. Source and relays need at least k transmissions to forward k independent packets that is the minimum

Algorithm 1 Computing c_1, where $E(x)$ is the probability E for $c_1 = x$ and $c_2 = (c-x)/R$.

```
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 |E(y_l) − E(y_r)| > 0.01 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)
```

required for the batch decoding. We also choose to share the same credit among the relays, following the same approach with other works [2] and enabling the estimation of the two variables with low-computational cost. As follows, $c_1+Rc_2 = c$ for avoidance of slot violation or underutilization, thus $c_1 \in \{k, k+1, ..., c-Rk\}$ [1]. The balance between these two sub-challenges is related to the aforementioned system objective.

In order to satisfy this objective, the source shares the total credit c in a way that maximizes the aimed probability, however, for a packet and not for a batch. This is an approximation followed also by MORE. The c_1 and c_2 credits are retrieved at the source by applying the ``Golden section'' search of Algorithm 1, since probability E is a convex function of c_1. Appendix A.1 proofs that E is a convex function of c_1. The source knows the transmission error probabilities of all links due to the ETX estimation mechanism, described in detail before. Moreover, the relays learn the c_2 value by the source through the periodical broadcasts, which are used for the estimation of the transmission error probabilities. The complexity of this algorithm is $O((R+1)D\log c)$ (the estimation of E requires at most $(R+1)D$ calculations for

[1] If $k > c - Rk$, then we satisfy only the one-hop away destinations giving all credit to $c_1 = c$.

a specified couple of c_1 and c_2 values), while the complexity of the corresponding algorithm of MORE is $O(DN^2)$ for the case of multicast forwarding [6]. It is worth to mention that R is limited, since in most cases there is no need for more than 4 or 5 relays supporting the two-hop away destinations. Moreover, for large values of c, the algorithm converges rapidly in less iterations than $\log c$, since the maximum value of E is close to 1 and is given for many c_1 values. Subsequently, the complexity of this algorithm is apparently better than this of MORE.

Our experimentation shows that this transmissions policy outperforms the behavior of MORE by giving more transmission opportunities over the lowest quality links. Actually in MORE, the source does not stop transmitting and competing with the one-hop relays for the medium access during the whole period of a batch forwarding. This approach results to equal transmission opportunities among the source and its one-hop relays, regardless of the links quality and the corresponding MORE's credit assignment, since the CSMA/CA mechanism of 802.11 statistically distributes equally the channel access among the competing transmitters. In ViMOR, the contentions/collisions are reduced by enforcing relays to apply the *first-decode-then-transmit* policy. When applying this policy, the relays are imposed to start forwarding a batch only after the successful decoding of this batch and the retrieval of the corresponding k initial packets, thus the contentions/collisions are reduced. This policy is also applied for a second reason; the relays should not spend transmission opportunities of the source for transmission of packets, which are not linear combinations of all k initial packets and thus contain less information.

3.1.3 Classification and prioritization of the video packets

The last difference of ViMOR is the implementation of NC using Priority Linear Coding (PLC), which classifies the packets to $\mathcal{O}$ priority classes and replaces the Random Linear Coding (RLC) of MORE. Our scheme focuses on video streaming, which inherently consists of packets of varied significance. For example, the packets that include segments of the intra-

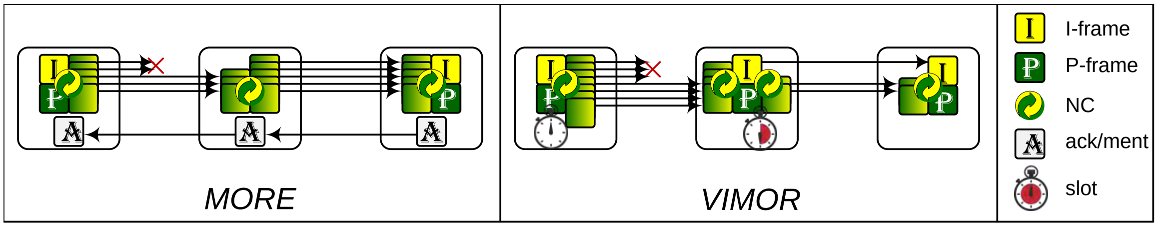


Figure 3.2: The main differences between MORE and ViMOR. In MORE, the source and the relay are competing for the medium during the whole period of the batch forwarding, which ends when the source receives the acknowlegment. The source has the same transmission opportunities with the relay, although it is more susceptible to transmission errors and needs more. ViMOR replaces the acknowlegment with a time counter (gaining the time spent for the acknowlegment forwarding). It gives more transmission opportunities to the source and disables the collisions between the source and the relay by applying the *first-decode-then-transmit* policy. Finally, the relay uses PLC enabling the forwarding of the I-frame even if the P-frame is lost.

encoded frames (I-frames) are more important than the packets that include segments of the inter-decoded ones (P-frames and B-frames). The latter P/B-frames cannot be decoded without having the corresponding I-frame. 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. More specifically, we utilize a high and a low priority o_h and o_l class respectively. The o_h packets include the segments of the intra-frames, having always $k_{o_h} = \alpha k$, while the o_l class contains all batch packets and $k_{o_l} = k$. We assume that the intra-frames of a batch need a proportion of the whole batch size less or equal to α. The credit of each relay is shared proportionally to each class. The o_h packets take a proportion α of the whole credit c_2, and the o_l packets take the whole credit. This means that each relay generates and transmits the first αc_2 packets as linear combinations of the most important o_h packets, while the rest ones are linear combinations of all packets. The source does not change its behavior, doing the same as with the RLC mechanism.

The receiver performs two parallel decoding processes; the first one is fed with the packets generated from the coding of the o_h packets, while the second one is fed with all received packets. The two decoding processes are executed simultaneously, hence enabling the suc-

Table 3.2: 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 interfaces	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

cessful decoding of the o_h packets with higher or equal probability. Even if the decoding of the whole batch is infeasible, a receiver may be capable to decode the most important packets of this batch. This enables the reception of a video sequence of tolerable quality, in case that the reception of a high quality video is infeasible. However, this comes at the cost of higher CPU utilization, since the encoding/decoding process is the one with the highest CPU usage.

Figure 3.2 summarizes and depicts the main differences between MORE and ViMOR in a representative example with a source, one relay and one destination.

CHAPTER IV

Implementation on Click Modular Router

4.1 Click Modular Router

The Click Modular Router [15] is a software architecture for building flexible and configurable routers. A Click router is assembled from packet processing modules called elements. Individual elements implement simple router functions like packet classification, queuing, scheduling and interfacing with the network devices. A router configuration is a directed graph with elements at the vertices with packets flowing along the edges of the graph. Click configurations are modular and easy to extend supporting user defined abstractions.

A Click element represents a unit of router processing. An element represents a conceptually simple computation, such as decrementing an IP packet's time-to-live field, rather than a large, complex computation, such as IP routing. The most important properties of an element are:

- **Element class.** Each element belongs to one element class. This specifies the code that should be executed when the element processes a packet, as well as the element's initialization procedure and data layout.

- **Ports.** An element can have any number of input and output ports. Every connection goes from an output port on one element to an input port on another. Different ports can have different semantics; for example, second output ports are often used to emit erroneous packets.

- **Configuration string.** The optional configuration string contains additional arguments that are passed to the element at router initialization time. Many element classes use these arguments to set per-element state and fine-tune their behaviour.

- **Method interfaces.** Each element supports one or more method interfaces. Every element supports the simple packet-transfer interface, but elements can create and export arbitrary additional interfaces. For example a queue might export an interface that reports its length. Elements communicate at run time through these interfaces, which can contain both methods and data.

Click provides two kinds of connections between elements, push and pull. In a push connection, the upstream element hands a packet to the downstream element. In a pull connection, the downstream element asks the upstream element to return a packet. Apart from implementing push or pull methods an element can be implemented with agnostic ports, meaning it can work as either push or pull depending on its context in the router. Elements not participating in packet processing but rather storing and maintaining data are referred to as Information elements.

Besides being activated upon a push or pull request, an element can be put on the scheduling queue of the Click router by registering a task or scheduling a timer at a precise time.

Click elements can add the possibility to interact with them once they are running through so called handlers. This can for example be done to change the configuration of the element, to reset or change values or to add or remove entries from Information elements. Handlers can either be read or write handlers, used to retrieve information and return it or to change settings inside the router, if necessary with passing of parameters.

4.2 ViMOR implementation

The implementation of ViMOR routing scheme is based on the Click framework. In this work, we extend and modify the Click based implementation of the MORE routing algorithm, introducing the aforementioned contributions for video streaming. More specifically, the elements utilized in our implementation are listed below along with a short description of their functionality.

VIMOR:

The main element implementing the push and pull functions. It receives packets either from the upper layer through the tun interface to which packets are passed from the kernel, or directly from the wireless interface operating in monitor mode. In this element we implemented a timer function which initiates, in case of source node, the encoding process for a new batch ensuring that the duration of this batch's transmission is limited to the time slot passed as an argument from the configuration string. Additionally, in case of a destination node, packets are passed to the elements responsible for the decoding of a batch, while in case of a relay node packets are passed for encoding and forwarding. Output ports of VIMOR element are connected either to the tun interface for promoting decoded packets at the destination to the IP layer, or to the wireless interface of the source and relays for the transmission of packets through the wireless medium.

VIMORFlow:

The element responsible for the control of the flow. Currently, only a single flow is supported. More particularly, this element implements the *first-decode-then-transmit* policy on relays in which the encoding process starts only after the successful decoding of a batch received from source. In addition, the credit assignment for each node is implemented inside this element and via the use of Click handlers. The addition of a header containing the batch sequence number is also implemented here. MORE's original code for this element contained a function for the creation and forwarding of an ACK after the successful decoding of a batch. This function has been disabled according to the video streaming needs discussed previously.

VIMORBatch:

The element responsible for the management of a batch. It contains functions for inserting newly arrived packets to the matrices used for storage as well as taking already encoded packets for forwarding.

VIMORCodedBlock:

The VIMORCodedBlock element represents an encoded packet along with its coefficients and implements functions for the setup and linear operations on packet's bytes.

VIMORMatrix:

The VIMORMatrix element implements a matrix for the storage of the coded blocks of a batch along with functions for the linear combinations of the decoding process. We define an instance of this element for each priority class of packets.

VIMORMatrixMgr:

This element inherits VIMORMatrix and adds functions implementing the encoding of a new packet by generating random coefficients and linearly combining contents of the matrix.

4.3 Click configuration

In this section we present a sample configuration that a router implementing our protocol is running. We schematically display the elements used and discuss the procedures taking place prior to the entrance of a packet in the VIMOR element and past the exit.

4.3.1 Input Path

The input path consists of two routes. The first route is the shortest one where an IP packet coming from the tun interface which is part of the elementclass *LinuxIPHost* enters immediately the *VIMOR* element. The second route consists of a *FromDevice* element that receives packets coming from an interface and in our case from the wireless interface. A *Switch* element used for allowing or blocking incoming packets follows. It is useful for blocking the flow after a specific time period. *RadiotapDecap* then decapsulates the radiotap header containing

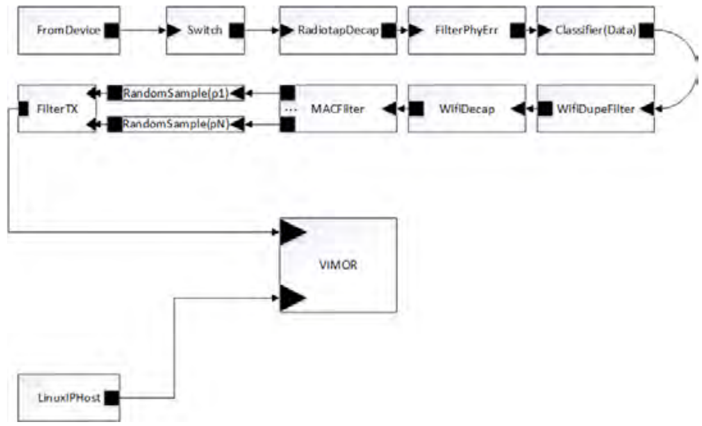


Figure 4.1: Input path of ViMOR's Click configuration.

useful information from the wireless card driver. Thereafter *FilterPhyErr* filters out packets that failed the 802.11 CRC check. The following *Classifier* separates data from control packets while the *WifiDupeFilter* filters out duplicate packets. Afterwards *WifiDecap* turns 802.11 packets into ethernet packets. In order to emulate links with various transmission error probabilities we operate in perfect conditions and filter out packets with a given probability and from a specific mac address with the next two elements (*Classifier*, *RandomSample*). Finally, *FilterTX* filters out transmission feedback packets before passing packets to *VIMOR* element.

4.3.2 Output Path

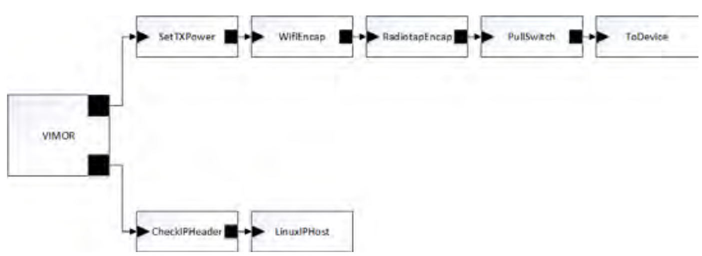


Figure 4.2: Output path of ViMOR's Click configuration.

The *VIMOR* element has two output ports. The first one for passing encoded packets for forwarding while the second one for passing the decoded packets up to the tun interface. After exiting the first port of *VIMOR*, the packet is handled by the *SetTXPower* element which sets the transmission power. Afterwards the *WifiEncap* element converts the ethernet packets to 802.11 packets with an LLC header. Then the *RadiotapEncap* pushes a radiotap header on the packet. A *PullSwitch* is used for blocking transmissions, when set, in a similar manner to the *Switch* included in the input path. Finally, the packet is passed to the *ToDevice* element for trasmission from the wireless interface. Packets coming from the second port of *VIMOR* after being checked by the *CheckIPHeader* element for a valid IP header are passed to tun interface for delivery to the IP layer.

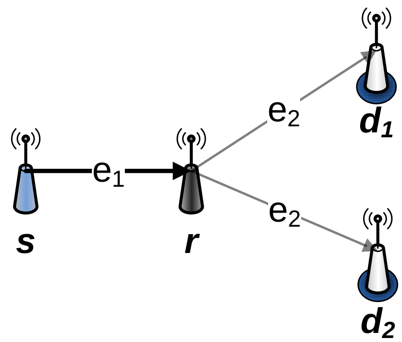
CHAPTER V

Experimentation Results

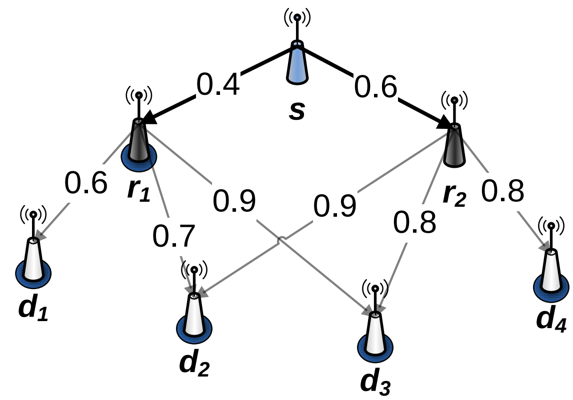
5.1 Experimentation Results

The deployment and evaluation of ViMOR took place at the NITOS testbed [1], where we conducted experiments under various topologies with specific features. NITOS is a a non-RF-isolated wireless outdoor testbed, so we used 802.11a to eliminate interference, since commercial 802.11 products in Greece use only 802.11b/g. The specifications of the NITOS nodes used for the experiments are depicted in Table 3.2.

The thorough evaluation of ViMOR required the experimentation under different topologies with several connectivity conditions. Since it is impossible to find the desired conditions in a testbed with stationary nodes, we reproduce them with the use of a distributed packet filtering mechanism, that we further explain. More particularly, we selected NITOS nodes that are close to each other, shaping a full mesh connected topology with robust links (transmission error probabilities very close to zero). Then, we applied a packet filter to each one of these nodes, allowing a received packet to pass through with a specific probability, according to the transmitter's identifier. This mechanism enabled the full control of the connectivity map, providing us with the ability to replicate any lossy link. The topologies of our experimental setups are illustrated in Figure 5.1. Each link represents a communication channel for direct transmission from a given node to another one, and is labeled by its corresponding transmission error rate.



(a) $\mathcal{R} = \{r\}, \mathcal{D} = \{d_1, d_2\}$



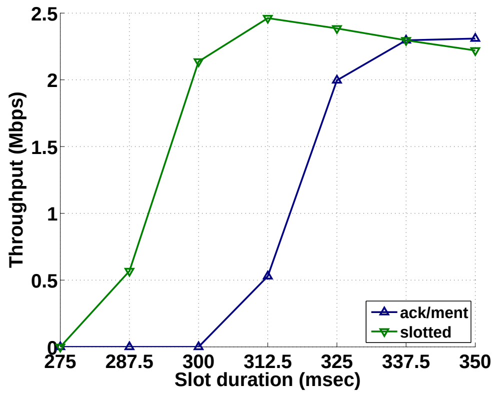
(b) $\mathcal{R} = \{r_1, r_2\}, \mathcal{D} = \{r_1, d_1, d_2, d_3, d_4\}$

Figure 5.1: Two different topologies with 4-nodes (a) and 7-nodes (b) used in our algorithm evaluation.

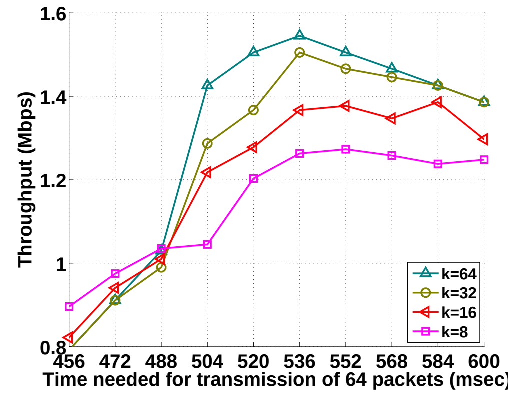
5.1.1 First class of experiments

The first class of our experiments is conducted using the topology of Figure 5.1(a), where the source is s, $\mathcal{R} = \{r\}$ and $\mathcal{D} = \{d_1, d_2\}$, while the transmission error probabilities e_1 and e_2 are adjusted appropriately. The performance of both MORE and ViMOR is expected to be highly insensitive to different batch sizes ($k = 8, 16, 32, 64$), as it is presented in [6]. However, as we explain later and conclude in our experimentation, $k = 64$ seems to be the best choice for ViMOR. The main configuration parameters are that RTS/CTS is disabled, as it happens in most real networks, and all nodes use $\rho = 6$ Mbps as physical transmission rate. Finally we configure the packet payload to be equal to 1470 bytes. The packet size is $b = 1556 + k$ bytes, after adding the WiFi, IP and UDP headers/trailers, as well as the MORE header that is also adopted by ViMOR. The MORE header features $22 + k$ bytes length, where the k bytes are used for holding the coefficients that linear coding uses to generate the corresponding packet.

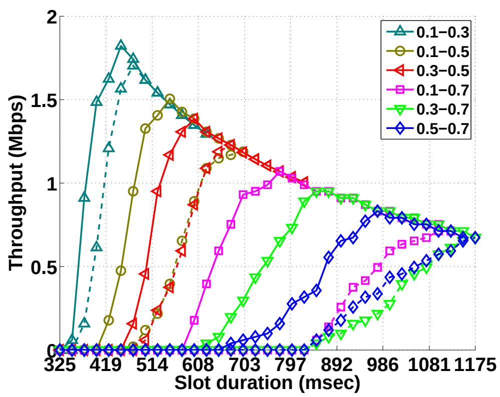
In the following lines, we present the evaluation of ViMOR. ViMOR's proposed contributions have been evaluated individually, conducting three separate sets of experiments in order to explore the individual benefits of each contribution.



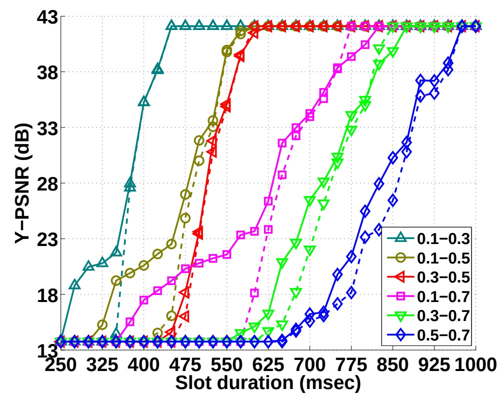
(a) The slotted mechanism of ViMOR compared to the acknowledgment mechanism of MORE for $k = 64$ and $e_1 = e_2 \approx 0.001$.



(b) The VIMOR's credit assignment performance for $k = 8, 16, 32, 64$, $e_1 = 0.1$ and $e_2 = 0.5$.



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



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

Figure 5.2: Evaluating ViMOR in the 4-nodes experimentation topology of Figure 5.1(a).

5.1.1.1 Slotted vs. acknowledgment mechanism

In the first set of experiments, the throughput performance of the proposed video-aware slotted mechanism of ViMOR (details in Subsection 3.1.1) is compared to the one of the acknowledgment mechanism of MORE. We perform the comparison using the first topology under transmission error probabilities close to zero, in particular $e_1 = e_2 \approx 0.001$, and $k = 64$, since this is the best value for k as we will see later. The performance of the slotted mechanism is quite insensitive to the k value in this experiment. When using MORE, the source transmits continuously, while the relay retransmits a specific number of packets for each one received. In our case, this number is equal to one. The source proceeds to the next batch after receiving an aggregate acknowledgment from both destinations. On the other hand, under the slotted mechanism of ViMOR, the source proceeds to the next batch after the expiration of the current slot, even if the destinations have not yet decoded the current batch.

The plots in Figure 5.2(a) depict the average throughput of the on-time decoded packets between the two destinations for the two mechanisms. On-time decoded packets are only these that have been delivered in a time interval less than the slot duration τ. The traffic load sent from the source may be larger than the corresponding throughput, since it also includes packets that either got lost, as it happens in the slotted mechanism, or received too late, that happens in the acknowledgment mechanism. The horizontal axis represents the slot duration in milliseconds, while on the vertical axis we depict the measured throughput in Mbps. It is obvious that for long time slots the performance of the two mechanisms is similar, or the acknowledgment mechanism performs better, due to the underutilization of the wireless medium that the slotted mechanism imposes as the slot duration increases. Both mechanisms achieve to forward frames on-time, while the acknowledgment one succeeds in pre-buffering more and more as the slot period increases. However, as the slot period decreases, it is evident that the proposed mechanism achieves a significant performance improvement, delivering video in cases that the acknowledgment mechanism is completely inefficient ($\tau \leq 300$ msecs). This is a remarkable result, since it enables transmission of higher quality video sequences, that feature high frame ratios (high f) or high definition frames (high l) and subsequently require low slot duration $\tau = gk/fl$.

5.1.1.2 Evaluation of the transmissions policy

In the second set of experiments, we evaluate the proposed transmissions policy by configuring the nodes connectivity and applying the suggested credit assignment mechanism of Subsection 3.1.2. Initially, we configure the transmission error probabilities $e_1 = 0.1$ and

Table 5.1: Credits c as a percentage of slot duration τ for physical transmission rate $\rho = 6$ Mbps and $k = 8, 16, 32, 64$

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

$e_2 = 0.5$ for selecting the best k. The selection of these error rates is the result of extensive experimentation, where we have observed the largest differentiation in the performance of the proposed policy for multiple values of k. Figure 5.2(b) shows the performance of the proposed credit assignment for $k = 8, 16, 32, 64$. The horizontal axis represents the time needed in milliseconds for delivering a sequence of 64 packets, while on the vertical axis we depict the measured throughput in Mbps. The interval represented in horizontal axis is equal to $\tau \cdot 64/k$, where τ is the slot duration that a batch needs to be delivered. As it is clearly depicted, $k = 64$ is the best choice. Although $k = 64$ imposes the largest overhead in packet transmission, since it uses longer headers, it enables the most accurate estimation of the redundancy packets that a transmitter should use. Therefore, for the rest of the experiments presented, we use $k = 64$.

The next step is to configure the transmission error probabilities e_1 and e_2, using different pairs of probability values. Figure 5.2(c) shows the performance of the proposed credit assignment compared to the performance of a simple and equally distributed credit assignment ($50 - 50\%$), where $c_1 = c_2 = c/2$ independently of the e_1 and e_2 values. The horizontal axis represents the slot duration τ in milliseconds, while the vertical axis represents the achieved throughput in Mbps. The solid lines depict the throughput performance of the ViMOR policy, and the dashed lines the one of the equally distributed assignment policy. The c value depends on the slot duration τ, as we have already mentioned, and it is presented in Table 5.1. Each pair of same colored solid and dashed plots corresponds to a different couple of probability pairs $e_1 - e_2$. It is worth to mention that both assignment policies succeed the same results if we swap the values of e_1 and e_2.

We compare the ViMOR policy with the $50 - 50\%$ one, because in MORE the transmission opportunities among source and one-hop relays are equally shared, due to the 802.11

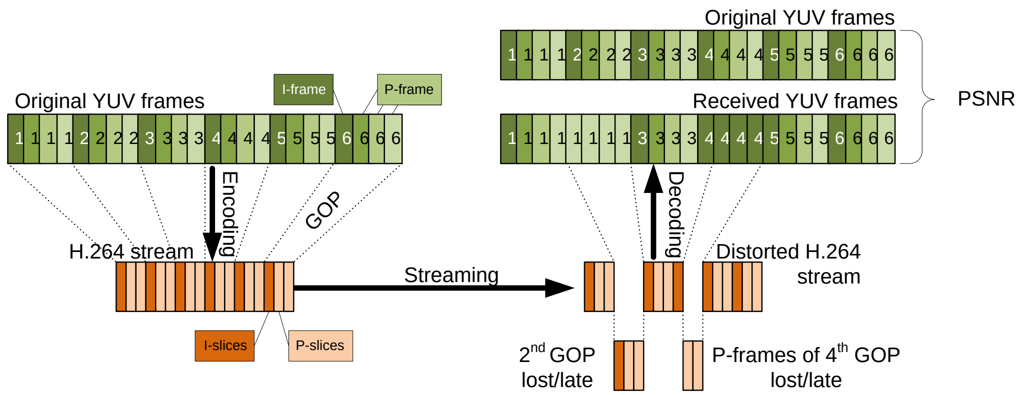


Figure 5.3: PSNR evaluation of the received video stream.

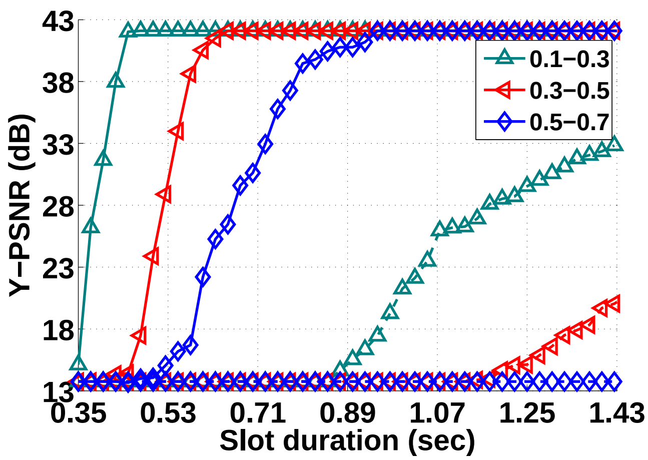


Figure 5.4: The *first-decode-then-transmit* policy compared to the *partial-decode-then-transmit* one, for $k = 64$, $\alpha = 1/3$ and multiple e_1 and e_2 values. The marker of its line indicates the e_1 - e_2. The solid lines correspond to the first policy and the dashed lines to the second one.

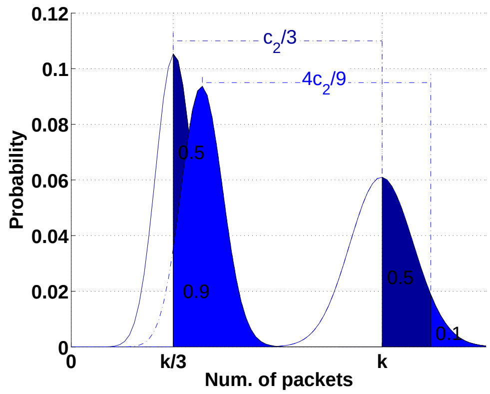
MAC protocol. Subsequently, although MORE applies a more sophisticated credit assignment policy, the result is the same with applying the $50-50\%$ one. In ViMOR, the first-decode-then-transmit policy applies an indirect scheduling that reduces the contentions/collisions and allows a proportional sharing of the transmission opportunities. It is noticeable that the proposed policy succeeds in delivering higher throughput traffic in all the cases that $e_1 \neq e_2$. As it is expected, the throughput gain of the proposed policy is high in cases that $|e_1 - e_2|$ is large enough. Moreover, it is worth to mention that the performance of the equally distributed credit assignment depends only on the lowest quality link, since it is the same for all probability pairs that feature the same $\min(e_1, e_2)$.

5.1.1.3 PLC vs. RLC

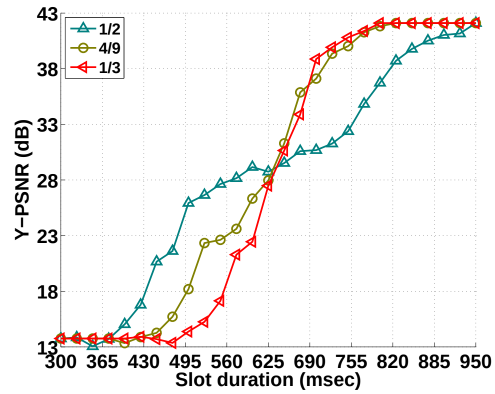
Finally, in our third set of experiments we examine the behavior of our proposed PLC mechanism as compared to the RLC mechanism, with respect to the PSNR metric. We replace the plain data streams with video ones and collect the received videos from each destination under both mechanisms. Each lost or late frame is replaced by the previous video frame, that could be replaced by the frame before the previous one for the same reason, etc. (we always provide the first frame to all destinations). Subsequently, in the extreme case that nothing is received on-time from a destination, the corresponding perceived video corresponds to a sequence of repeated frames that are the same with the first one. Obviously, if an inter-frame is not lost or late but the corresponding intra-frame is, then the inter-frame is useless. Figure 5.3 illustrates how we evaluate the received video stream using the PSNR metric.

We conduct the experiments in almost lossless links by configuring the transmission error probabilities as in the previous experiment. We configure $\alpha = 1/3$ and we use the video sequence of *foreman* with CIF resolution, encoded in H.264 with GOP size $g = 10$ and only I/P-frames (no B-frames). The quality of the H.264 compression (in particular quantization) is such as the average size of a compressed GOP to be almost equal to the batch size ($k/l \simeq 1$), while the size of each I-frame is approximately the $\alpha = 1/3$ proportion of the whole GOP size. For different frame ratios f, our scheme utilizes different time slots equal to $\tau = g/f$. In Figure 5.2(d), we observe how our enhanced PLC mechanism prioritizes the decoding of I-frames, outperforming the simple RLC mechanism. The horizontal axis represents the slot duration τ in milliseconds, while the vertical axis represents the perceived video quality in destinations, measured in PSNR, comparing with the original YUV sequence. We notice that the lowest PSNR value of 13.4 corresponds to the video sequence that results from no batch reception, while the largest PSNR value of 42.1 corresponds to the video sequence that results from no occurrence of lost batch. Moreover, the PSNR gain of PLC is high in cases that $|e_1 - e_2|$ is large enough, as it happens in the previous experiment.

After replacing RLC with PLC, our first approach was to relax the *first-decode-then-*



(a) The probability of successful decoding of the most important and all packets, when $\tau = 525ms$. The two left PDFs show the probability the most important packets to be decoded, when the relay transmits $c_2/3$ or $4c_2/9$ packets as linear combinations of these packets, and the right PDFs correspond to the probability all packets to be decoded.



(b) The PSNR performance of ViMOR when the relay transmits $c_2/3$, $4c_2/9$ and $c_2/2$ packets as linear combinations of the most important packets.

Figure 5.5: The performance of ViMOR for $k = 64$, $\alpha = 1/3$, $e_1 \approx 0.001$ and $e_2 = 2/3$.

transmit policy at the relays, enabling them to start transmitting even if they have decoded only the most important packets of each batch. We named this policy *partial-decode-then-transmit*. Although this policy seems to be more efficient in case that the delivery of the whole batch to the relay is impossible (enabling at least the forwarding of the most important packets), our experimental results show that it is inefficient due to the increased collisions between the source and the relay. As we see in Figure 5.4, the PSNR performance of the *partial-decode-then-transmit* policy is worser than this of the *first-decode-then-transmit*.

We also experimented to share non-proportionally the credit among the priority classes of PLC, giving more than αc_2 credit to the relays for the transmission of the most important packets of the batch. In this way, we increase the probability that the destination successfully decode the most important packets of each batch, however, at the cost of reducing the probability for successful decoding of the whole batch. This practice may offer better perceptual experience to the destinations, prioritizing even more the packets of the I-frames. As we see in Figure 5.5(a), by enabling the relay to transmit more than αc_2 packets as linear combinations of the most important packets, the probability of successful decoding of these packets

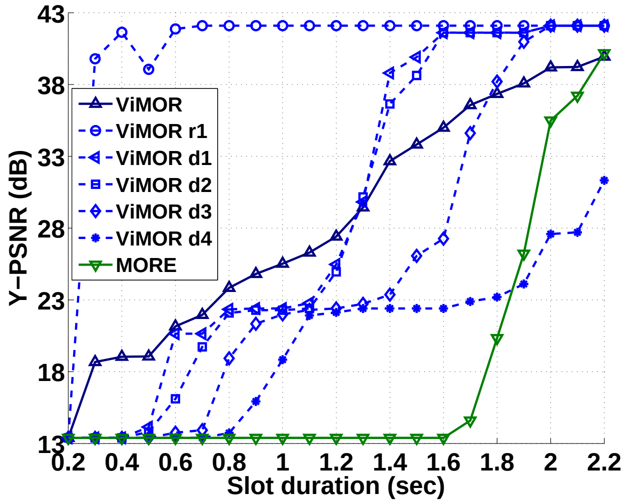


Figure 5.6: Video performance comparison between ViMOR and MORE in the 7-nodes topology of Figure 5.1(b). The dashed lines correspond to the PSNR evaluation of the receipt video of each individual destination under ViMOR.

increases from 0.5 to 0.9, however, in cost of the probability of decoding all packets, which decreases from 0.5 to 0.1. As we see in Figure 5.5(b), by increasing the number of packets that relay produces as combinations of the most important packets, the destinations deliver better video quality for short slots, but worse quality for longer slots.

5.1.2 Second class of experiments

The second class of our experiments aims at comparing the performance of ViMOR to MORE in terms of PSNR, evaluating all contributions together (slotted mechanism, enhanced transmissions policy and PLC). The experiments were conducted in the 7-nodes topology of Figure 5.1(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 other configuration variables are the same as in the previous experiments, since $k = 64$, RTS/CTS is disabled, $\rho = 6$ Mbps, 1470 bytes is the payload size and the video-specifics $\alpha = 1/3$ and $g = 10$.

In Figure 5.6, ViMOR obviously enables the r_1 node to enjoy high quality video for $\tau > 0.6$ sec, while the other 2-hop destinations start receiving a satisfying quality of video after some slots. In particular, all destinations receive a video stream with PSNR greater or equal to 22.4 for $\tau > 1.1$ sec, which corresponds to a video sequence where all I-frames are almost received and P-frames are not. This happens when the destinations are able to decode only the high priority o_h packets of each forwarded batch, that approximately include the I-frame of the corresponding GOP. The average PSNR value among all destinations, under the ViMOR scheme, is increasing constantly for all slot durations $\tau > 0.3$ sec, while the corresponding PSNR value of the MORE scheme is increasing after slot $\tau > 1.7$ sec. Obviously, ViMOR enables video streaming, even in a subset of the destinations, with slot durations up to 5.3 times smaller than the corresponding of MORE. Moreover, the PSNR gain is up to 270% for a slot $\tau = 1.6$ sec, while the average gain is 175%.

CHAPTER VI

Conclusion

6.1 Conclusion

In this thesis, we presented ViMOR, the first practical algorithm that efficiently forwards multicast video over wireless networks. To the best of our knowledge, this is the first implementation of a video-aware multicast OR algorithm for 802.11 mesh networks. The potential of this researching effort is well promising, since the results of our experimentation depict a PSNR gain up to 270%. Of course, there are many open issues for further research. For example, a rate control algorithm that enables the utilization of larger rates than the basic one may allow higher throughput and perceived video quality. However, this comes at the cost of reducing the network coverage area. Moreover, another policy that imposes less strict scheduling would enable relays to transmit even if they have decoded only the packets of the high priority class, allowing in this way the delivery of a video even in smaller slots. On the other hand, for longer slots there would be a degradation in the perceived video quality because of the increased probability of contentions/collisions. A third point for further research is the effect of an increased number of priority classes, as well as a different way of sharing the credit c among the priority classes. These all are challenging issues and subjects for our ongoing research.

APPENDICES

APPENDIX A

Convexity of Algorithm 1

A.1 Convexity of Algorithm 1

Let $\mathcal{P}^d$ be the set of one-hop and two-hop paths connecting the source s with the destination $d \in \mathcal{D}$. Let e_{xy} be the transmission error probability of the link connecting node x to node y, thus $e_{xy}^z \in (0, 1]$ is the probability of z successive transmission errors over this link. We define E_p to be the probability of unsuccessful packet delivery to the destination d through a path $p \in \mathcal{P}^d$, when the source and each relay are charged with a credit c_1 and $c_2 = (c - c_1)/R$ respectively. We show that this quantity is always a convex function of c_1.

For example, the probability of unsuccessful packet delivery through the one-hop path $p' \in \mathcal{P}^d$ is $E_{p'} = e_{sd}^{c_1}$, that is a convex function over all legitimate values of c_1. Furthermore, the corresponding probability of a two-hop path $p'' \in \mathcal{P}^d$, that utilizes a relay $r \in \mathcal{R}$, is a convex function of c_1 as well, equal to $E_{p''} = 1 - (1 - e_{sr}^{c_1})(1 - e_{rd}^{(c-c_1)/R})$. In particular, $E_{p''}$

is convex since its second derivative is always non-negative, as it is depicted in (A.1).

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

The packet is not delivered to d, if each of the paths of $\mathcal{P}^d$ fails to do it. So $E^d = \prod_{p \in \mathcal{P}^d} E_p$ is the probability of unsuccessful packet delivery to d over all paths of $\mathcal{P}^d$. As follows, E^d is a convex function of c_1 as well, since E_p is a positive and convex function for all $p \in \mathcal{P}^d$ [5], as we proved before. Finally, the average probability $E = 1 - \sum_{d \in \mathcal{D}} E^d/D$ is a convex function of c_1 again, which means that at most two c_1 integer values exist that maximize this probability and achieve system objective.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Nitlab: Network implementation testbed laboratory, `http://nitlab.inf.uth.gr/NITlab`.

[2] Ozgu Alay, Thanasis Korakis, Yao Wang, Elza Erkip, and Shivendra S. Panwar. Layered wireless video multicast using omni-directional relays. *IEEE ICASSP 2008*.

[3] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. *ACM MobiCom 2005*.

[4] Sanjit Biswas and Robert Morris. ExOR: opportunistic multi-hop routing for wireless networks. *ACM SIGCOMM 2005*.

[5] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. *Cambridge University Press*, 2004.

[6] Szymon Chachulski (now Jakubczak), Michael Jennings, Sachin Katti, and Dina Katabi. Trading Structure for Randomness in Wireless Opportunistic Routing. *ACM SIGCOMM 2007*.

[7] Kostas Choumas, Ilias Syrigos, Thanasis Korakis, and Leandros Tassiulas. Video-aware Multicast Opportunistic Routing over 802.11 two-hop mesh networks. *IEEE SECON 2014*.

[8] Saumitra M. Das, Himabindu Pucha, Konstantina Papagiannaki, and Y. Charlie Hu. Studying wireless routing link metric dynamics. *ACM IMC 2007*.

[9] Christina Fragouli, Dina Katabi, Athina Markopoulou, Muriel Médard, and Hariharan Rahul. Wireless Network Coding Opportunities & Challenges. *MILCOM 2007*.

[10] Sachin Katti, Dina Katabi, Hari Balakrishnan, and Muriel Medard. Symbol-Level Network Coding for Wireless Mesh Networks. *ACM SIGCOMM 2008*.

[11] Dimitrios Koutsonikolas, Y. Charlie Hu, and Chih-Chun Wang. Pacifier: high-throughput, reliable multicast without "Crying babies" in wireless mesh networks. *IEEE/ACM Trans. Netw. 2012*.

[12] Dimitrios Koutsonikolas, Chih-Chun Wang, and Y. Charlie Hu. CCACK: Efficient Network Coding Based Opportunistic Routing Through Cumulative Coded Acknowledgments. *IEEE INFOCOM 2010*.

[13] Zhi Li, Limin Sun, Xinyun Zhou, and Liqun Li. Priority Linear Coding Based Opportunistic Routing for Video Streaming in Ad Hoc Networks. *IEEE GLOBECOM 2010*.

[14] Yunfeng Lin, Baochun Li, and Ben Liang. CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding. *IEEE ICNP 2008*.

[15] Robert Morris, Eddie Kohler, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM SOSP 1999*.

[16] Yuan Yuan, Hao Yang, Starsky H. Y. Wong, Songwu Lu, and William Arbaugh. ROMER: Resilient opportunistic mesh routing for wireless mesh networks. *IEEE WiMesh 2005*.