Lightweight Random Linear Coding over Wireless Mesh Networks

Pablo Garrido[1(✉)], David Gómez[1], Jorge Lanza[1], Ramón Agüero[1], and Joan Serrat[2]

[1] University of Cantabria, Santander, Spain
{pgarrido,dgomez,jlanza,ramon}@tlmat.unican.es
[2] Universitat Politecnica de Catalunya, Barcelona, Spain
serrat@tsc.upc.edu

Abstract. We propose an enhanced version of an *intra-flow* Network Coding protocol, which was conceived to offer a reliable communication service, by means of the combination a Random Linear Coding (RLC) scheme with the UDP protocol. We reduce the overhead that was originally required in the protocol header and we assess, through an extensive campaign carried out over the `ns-3` framework, the performance gain that is brought by this enhancement, comparing it to the TCP protocol, as the mainstream transport-level solution to offer a reliable service. We study the impact of the various configuration parameters of the solution. Afterwards, we challenge the proposed scheme over random topologies (Wireless Mesh Networks or WMNs). The results show a remarkable gain (approximately 20 times higher) of the performance over an ideal channel, thanks to the aforementioned overhead reduction.

Keywords: Random linear coding · Network coding · Simulation · Wireless mesh networks

1 Introduction

Wireless networks have become the most widespread communication alternative. The fast roll-out of new technologies, such as LTE, and the strong consolidation of other alternatives, being WiFi the most outstanding example, together with the growing presence of advanced end-users devices (smartphones, tablets, etc.), are some of the main reasons behind this fact.

Despite the evolution that we have witnessed on the access to the Internet, the most used transport layer protocol, TCP, exhibits a poor performance over wireless networks. This problem has been tackled by the research community, which has made a great effort to come up with new approaches to provide a reliable end-to-end transport solution suitable for the hostile conditions of wireless networks. Several proposals have been made, ranging from modifications of the legacy TCP protocol to novel transport and cross-layer solutions that address the problem from different perspectives.

R. Agüero et al. (Eds.): MONAMI 2015, LNICST 158, pp. 164–178, 2015.
DOI: 10.1007/978-3-319-26925-2_13

Amongst them, *Network Coding* (NC) appears as one of the most promising techniques. Its basic principle is allowing nodes to process and code packets before sending them again, opposed to the classical *store-and-forward* paradigm.

In a previous work [9], we studied the joint operation of a Random Linear Source Coding scheme and the UDP protocol to provide a reliable service over a single wireless link. The main goal of this paper is to extend that initial contribution; first, we propose an enhancement of such scheme, which allows reducing the required overhead, since we minimize the information that needs to be sent for coding and decoding purposes. We analytically study the gain that is brought by this modification and, afterwards, we use an implementation over the `ns-3` platform [1] to validate such analysis and to broaden the comparison. The simulation results show that the throughput of the proposed solution is remarkably higher than the one observed with the legacy TCP protocol. In addition, we also assess the performance of the proposed solution over *Wireless Mesh Networks (WMN)*, which lead to random topologies.

This paper is structured as follows: Sect. 2 discusses some related works that address the topics covered herein. Section 3 briefly describes the RLC scheme and details the proposed enhancement; it also depicts the analytical study of the throughput over ideal channels. Afterwards, Sect. 4 discusses the results that were obtained after an extensive simulation campaign carried out over the `ns-3` platform to corroborate the previous analysis and to broaden it, considering more complex network deployments. Finally, Sect. 5 concludes the paper, identifying a number of issues that will be tackled in our future research.

2 Related Work

TCP was originally designed with the assumption that packet losses were mostly caused by the congestion of intermediate network routers, which was a sensible belief, since almost all communications happened over wired links. The little attention that was paid to other types of errors leads to a remarkable performance decrease. In particular, this is the case over WMNs [3], which are believed to play an important role in the forthcoming wireless networking realm. Their main limitation comes from the severe impact that wireless communications (due to the appearance of errors and the interference) might have over the performance.

Several works studied the impact that the error-prone characteristic of wireless networks has over the TCP performance [15,18]. In addition, a number of proposals have been made to overcome the limitations exhibited by TCP over wireless networks. One of the most promising techniques amongst them is NC.

The term *Network Coding* was originally coined by Ahlswede *et al.* in [2]. They discussed the suitability of the classic *store-and-forward* paradigm in IP networks, advocating that the integration of some additional functionalities at the intermediate nodes could yield remarkable performance enhancements. Since then, several works have proposed the use of this technique, to get either performance enhancements or more reliable communications.

One of the most relevant contributions is the work presented by Katti *et al.* [12]. They proposed the COPE protocol, where the forwarding nodes code packets that belong to different flows, combining them by means of a simple *XOR* operation. They exploited the broadcast nature of the wireless medium, assuming that the neighboring nodes were able to overhear packets not directly addressed to them; this would eventually reduce the number of transmissions, yielding a significant performance gains. However, it has been shown [8] that when the conditions of the wireless channels are poor, the performance gain of this approach is not so relevant.

Chachulski *et al.* proposed in [5] the protocol *MAC-independent opportunistic routing protocol* (MORE). MORE combines a *Random Linear Coding* (RLC) scheme with opportunistic routing; the source combines (codes) information belonging to the same flow. Besides, the authors also proposed a number of additional mechanisms to avoid unnecessary retransmissions by the relaying nodes. Nodes estimate the quality of each link by means of echo messages and, using this information, they decide whether to forward a packet. However, the authors did not consider the interplay with any transport protocol, and their analysis is mostly focused on the lower layers.

The approach fostered by the MORE protocol can be referred to as an *intra-flow* coding scheme. There are various works that follow a similar approach: for instance, [16] demonstrates that by linearly combining various packets, the maximum capacity bounds for multicast traffic can be assessed. On the other hand, Ho *et al.* [11] demonstrated that the use of random linear coefficients over a finite field was the best solution.

In our previous work [9] we proposed the combined use of RLC and the UDP protocol. We studied the performance over a single wireless link. In this paper we extend that initial contribution, by modifying the protocol specifications, so as to reduce the protocol required overhead, since we avoid transmitting the complete vector of random coefficients, only adding the corresponding (random) *seed* within the header. Using such seed, the destination is able to generate the same random coefficient vector, with which it can decode the original information. We analytically study the additional gain brought about by this enhancement, using an approach similar to that of used by Trullols *et al.* in [17].

This improvement might be rather relevant, since there is an impact of the finite field size over the system performance. Heidi *et al.* [10] showed that the overall complexity and the required overhead increase with the size of the finite field. In [9] we also concluded that the longer the finite field size the less likely is to have linearly dependent combinations, leading to a performance boost. However, there exists a clear trade-off between this improvement and the required overhead, that might even lead to lower throughputs. With the modification proposed in this work we make the overhead independent of the size of the finite field.

A complete different approach is the one by Kim *et al.* in [13], where they proposed a reliable solution that combined TCP and NC, the so-called Network Coded TCP (CTCP). Although they showed that CTPC outperforms the legacy

TCP, the behavior over networks with large Round Trip Time (RTT) was not appropriate. They also proposed sending a seed to alleviate the overhead that would have been otherwise needed for the decoding process, as we propose herewith.

In a more recent work, Krigslund *et al.* [14] have proposed an integration of COPE and MORE, to exploit the benefits brought by each of them. The *intra-flow* scheme provides a more reliable communication, reducing the impact of packet losses over the wireless channel. However, the basic principles of both schemes are unaltered.

Last, we should also highlight other works that have focused on the improvement of the traditional routing solutions. These, which are usually based on the shortest route goal, might not lead to the highest performances over wireless networks. One of the most relevant works in this group, carried out by De Couto *et al.* in [6], proposes the Expected Transmission Count (ETX) metric to estimate the quality of the wireless links, to boost the performance over multi-hop wireless networks.

3 Model and Design

As was already said, in [9] we proposed the combined use of RLC scheme and the UDP protocol to offer a reliable communication service. In this section we introduce an enhancement of such solution, allowing the use of larger Galois Field, $GF(2^q)$, and block sizes. We analytically prove the potential gain that can be achieved with these improvements.

3.1 RLC Protocol

Before depicting the proposed changes we briefly describe the basic operation of the RLC scheme. At the source, an RLC entity, placed between the IP and UDP layers, divides the upper layer information, which is stored in its own RLC *transmission buffer*, into fixed blocks of K packets. It sends linear combinations (we refer to them as coded packets) of these K packets belonging to the same block, which are built according to the following expression $p' = \sum_{i=0}^{K-1} c_i \cdot p_i$, where $\boldsymbol{c} = \{c_0\ c_1 \ldots c_{K-1}\}$ represents the corresponding coefficient vector. It is worth highlighting that the random coefficients c_i are generated from a finite Galois Field $GF(Q = 2^q)$. The procedure followed by the transmitted is depicted in Fig. 1.

When the destination receives a coded packet, it checks whether the corresponding coefficient vector is linearly independent from the previously received ones, by appending it to the $\mathbf{C}(K \times K)$ matrix and performing a rank calculation operation. In such case, the corresponding coded packet is said to be *innovative* and is stored at the RLC *reception buffer*. When $\boldsymbol{c}$ is linearly dependent, the packet is silently discarded.

Once the destination has received K *innovative* packets, it is able to decode the whole block, obtaining the original information; in addition, it sends an

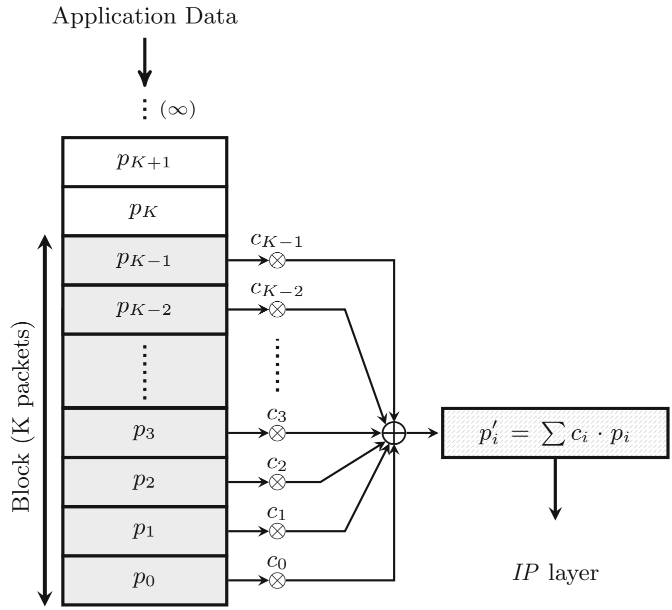


Fig. 1. Codification procedure at the source node

acknowledgment back to the source node, so that it can remove the corresponding block and start with the next one. If the acknowledgment was lost, the source would keep on sending packets of the previous block, which would cause the destination node to retransmit the corresponding acknowledgments, until the source realizes it needs to move forward to the next block.

Figure 2 shows the original RLC header, which is divided into two parts. The first one, with a fixed length of 9-bytes, contains all the information that needs to be exchanged from the source towards the destination. The second part of the header carries the corresponding random coefficients (i.e. **c**), whose length depends on the particular configuration of the coding processes.

- Type of message (1B): This field indicates the packet type: data packet ('0') or an acknowledgment ('1').
- Block Size K (1B): Number of packets per block. The maximum block size is 256, since the latency for larger blocks would be probably too high.
- Galois Field size - $GF(Q)$ (1B): The linear combination coefficients are randomly obtained from the Galois Field $GF(Q)$. In order to carry out the required operations, we have integrated the M4RIE [4] library into the `ns-3` platform; it imposes a limit of $Q = 2^8$.
- Fragment Number (2B): This field identifies the block that is being sent by the source and allows identifying spurious transmissions of an already received block.

– UDP source and destination ports (4B): A flow is identified by the source and destination IP address/UDP port tuples. Since the UDP header is coded, these ports need to be included in the RLC header.
– Coefficient vector, $\boldsymbol{c}$ (1-256B): Each coefficient, c_i, requires q bits and the header must include all the K coefficients.

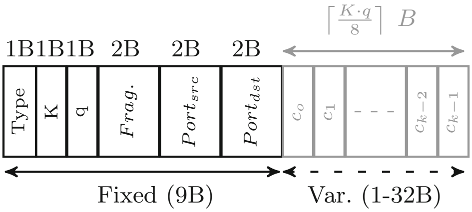


Fig. 2. Original RLC header

The overall header length in the first version of the protocol was therefore $9 + \lceil \frac{K \cdot q}{8} \rceil$, since all the coefficients need to be transmitted within every coded packet.

In order to avoid such large overhead, we propose keeping track of the *seed* used by the source node to generate the random coefficients by means of a *pseudo random number generator* and to actually send such seed in the RLC header, rather than the complete vector. When the destination node receives a coded packet it uses the seed to generate the same coefficient vector that the one used by the source entity. Taking this into account, we propose a new header, which is depicted in Fig. 3, where we substitute the variable length part of the original header by a fixed 4-byte field that indicates the random seed. The new header has a fixed length (no matter the values of K and q are) of 13 bytes.

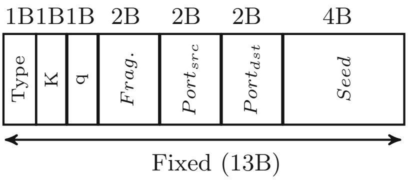


Fig. 3. Enhanced RLC header

3.2 Analytical Model

Following an approach similar to the one we already used in [9] we aim at analytically finding the performance gain brought by the proposed enhancement. We want to obtain the throughput that is perceived at the RLC layer, i.e. the one offered by the UDP protocol, which mostly depends on two main factors: the *spurious* transmissions (linearly dependent coefficient vectors) and the backwards acknowledgments that are sent by the destination.

On the first hand, the *spurious* transmissions are consequence of the transmission of linearly dependent coefficient vectors $\boldsymbol{c}$, which are silently dropped by the receiver, as was already said. We can establish their impact over the overall performance using the model proposed by Trullols-Cruces *et al.* in [17]. They derived the probability that a block of K packets can be successfully decoded after N receptions, as a function of the Galois Field $GF(Q)$ size, as can be seen in Eq. 1, where $\mathcal{P}_{K,Q}(K)$ is the probability of an ideal block transmission, i.e. the destination node got K *innovative packets* after K receptions, Eq. (2), without any spurious transmissions.

$$\begin{aligned}\mathcal{P}_{K,Q}(N > K) =&\mathcal{P}_{K,Q}(K)\cdot \\ &\cdot\left(\begin{bmatrix} N \\ N-K \end{bmatrix}_Q + \sum_{t=1}^{N-K} (-1)^t \binom{N}{t} \begin{bmatrix} N-t \\ N-K-t \end{bmatrix}_Q\right)\end{aligned} \tag{1}$$

$$\mathcal{P}_{K,Q}(K) = \frac{Q^{K^2}}{(Q^K - 1)^K} \cdot \prod_{j=1}^{K} \left(1 - \frac{1}{Q^j}\right) \tag{2}$$

Therefore, we can calculate the average number of *spurious* transmissions, as can be seen in Eq. 3 and the ratio between the number of excess packets and the overall transmissions, as is shown in Eq. 4. This ratio, ϵ, is the factor that jeopardizes the performance, being $0 \leq \epsilon \leq 1$.

$$E[N] = \sum_{i=N-K}^{\infty} i \cdot P_{K,Q}(i) \tag{3}$$

$$\epsilon = \frac{E[N]}{K + E[N]} \tag{4}$$

As a second factor, the penalization caused by the acknowledgments sent by the destination, ϵ_{ack}, is expressed as the ratio between the time required to send such confirmation packet, τ_{ack}, and the average transmission time of a data packet, τ_{data}, as defined in Eq. (5). Finally, we can model the *goodput*, i.e. the throughput perceived by the application layer, $\overline{Thput_{RLC}}$, as shown in Eq. (6).

$$\epsilon_{ACK} = \frac{\overline{\tau_{ack}}}{(K + \epsilon) \cdot \overline{\tau_{data}} + \overline{\tau_{ack}}} \tag{5}$$

$$\overline{Thput_{RLC}} = \overline{Thput_{UDP}} \cdot (1 - \epsilon) \cdot (1 - \epsilon_{ack}) \tag{6}$$

Section 4 discusses the results that are obtained applying this model. We will study the influence of both finite field and block sizes, i.e. q and k, over the achieved throughput, both with the original and the enhanced RLC header. This is of outer relevance to better understand the relationship between the system performance and configuration of the RLC scheme (block and Galois Field size), and to assess the gain that is brought by the new proposed coding solution.

Table 1. Simulation parameters

Feature	Value
Physical link	*IEEE 802.11b* (11 *Mbps*)
Error Model	Fixed FER (memoryless)
FER Values	[0:0.1:0.6]
RTX IEEE 802.11	1(RLC), 3(TCP)
Transport Layer	*UDP* (*NC*)/*TCP*
Application Data Rate	Fixed (Saturation)
MTU	1500 *Bytes* at *IP* layer

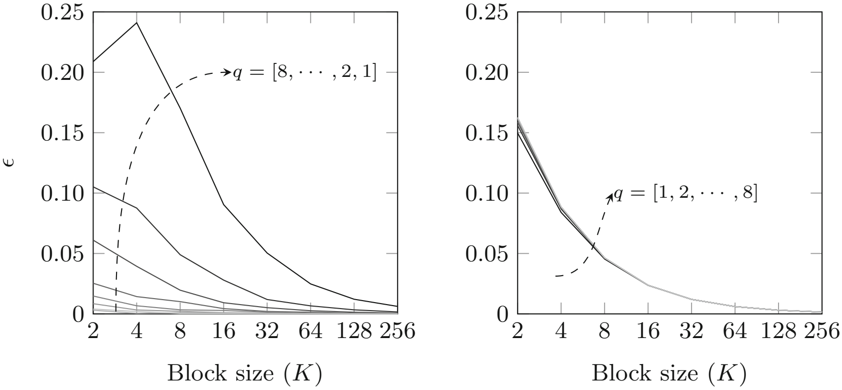
4 Results

In this section we outline the process that was followed to assess the performance of our proposal. Starting from the analytical model presented in Sect. 3, we analyze the potential benefits of using larger field ($Q = 2^q$) and block (K) sizes.

Afterwards, we carry out an extensive simulation campaign over the `ns-3` simulator to study the performance over the same scenario, allowing us to validate the analytical model. In addition, we exploit the implementation that was done to evaluate the behavior of the RLC scheme over more complex scenarios (Wireless Mesh Networks), comparing the achieved throughput (measured at the received application) with that of exhibited by the traditional reliable transport protocol, TCP (in particular, using the New Reno [7] version).

Table 1 summarizes the most relevant parameters of the scenario that was used throughout the simulations. The physical and Medium Access Control (MAC) layers follow the IEEE 802.11b recommendation. The retransmission scheme is disabled for the RLC protocol, since it does not provide any additional gain [9], while we fix a maximum of 3 retransmissions when the legacy TCP is used. Regarding the application layer at source nodes, we use a constant transmission rate, ensuring a saturated situation, so that the system bottleneck stays at the wireless channel.

First, we study the impact of the Galois Field and block sizes, q and K, respectively, over the system performance. As was already derived from Eqs. 4 and 5, the higher the K, the lower the performance penalization, while when q gets lower, the probability of having linearly dependent coefficient vectors increases (more spurious transmissions), with the consequent throughput reduction. Figures 4a and b show the evolution of these two penalization factors as a function of both parameters (block and Galois field sizes, K and q, respectively). Figure 4a yields that the use of a higher q greatly reduces the penalization on the performance, regardless of the block size. This stresses the relevance of the proposed enhanced header, since it would allow using a higher Galois Field without requiring an increase of the overhead. On the other hand, Fig. 4b shows that the acknowledgment penalization is mostly affected by the block size (the influence of the finite field is rather negligible), and this again would be solved with the



(a) Linear combinations (ϵ) (b) Acknowledgments (ϵ_{ack})

Fig. 4. *RLC* performance penalization factors

proposed new header, since in the original scheme, a higher block size would lead to an increased overhead and the throughput (as will be seen later) could be even reduced.

In order to complement the previous results, we can analytically establish the region where the use of the new header format would be worthy, if only the overhead, and not the penalization factors, was taken into account. If the previous header had a size of $\lceil 9 + \frac{K \cdot q}{8} \rceil$ bytes, and the new header has a fixed length of 13 bytes, Eq. 7 establishes the condition in which the use the new proposed format would lead to a lighter overhead. Figure 5 graphically represents this region; as can be seen, regardless of the q value, with a block size greater than 32 packets the new header is always beneficial; in addition, by increasing q we broaden the range of K values that are worth using.

$$\left\lceil \frac{K \cdot q}{8} \right\rceil > 4 \tag{7}$$

Once we have assessed the impact of the new header, we carry out an extensive simulation-based analysis to better understand the behavior of various configurations of the RLC scheme. We deploy two nodes and we represent the throughput that was observed when using different combinations of $q = [2, \cdots, 8]$ and $K = [2, \cdots, 256]$. Figures 6 and 7 shows the results that were observed for the original (variable-length) RLC header and for the one proposed in this paper, respectively. For the former case (see Fig. 6), the impact of both penalization factors is clearly seen. For a binary Galois Field ($q = 1$), the performance increases with K; however, when $q > 1$, the overhead required to transmit the corresponding coefficients leads to a performance reduction for large K values. As can be seen, the performance of the RLC scheme surpasses the TCP throughput either when $K > 16$ and $q \leq 1$, or for $K > 4$ and $q > 2$.

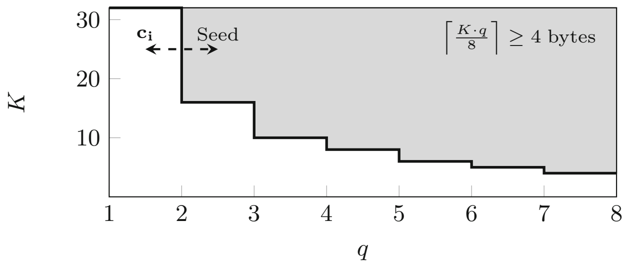


Fig. 5. Region where the use of the new RLC header is worthy

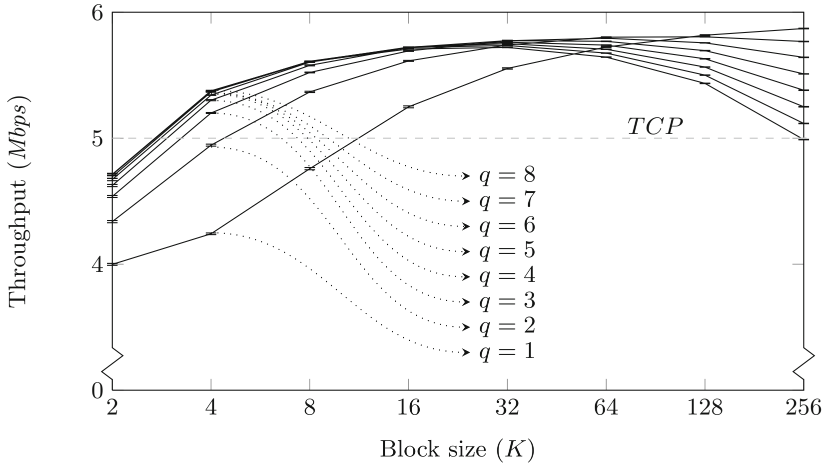


Fig. 6. Throughput observed for the RLC long header over an ideal single hop transmission

Figure 7 shows the throughput that was observed when the new header was used; in order to ease the comparison it also includes the curve corresponding to the original header and $q = 1$. It is worth highlighting that in this case, the new format allows maintaining the performance even for $K > 64$, while the throughput was severely jeopardized in the original solution. Thanks to this enhancement, we can take full advantage of the configuration that reduces the impact of the two aforementioned penalization factors, without increasing the overhead. The throughput gain, compared with the legacy TCP, under ideal conditions and for $K > 128$, is $\approx 18\,\%$, regardless of the q value.

Figure 8 shows the throughput that was observed when the quality of the wireless link between the source and the destination gets worse[1]. We increase the FER between 0 and 0.6, and we analyze the behavior of two configurations of the proposed scheme: $q = 1, K = 64$ and $q = 6, K = 256$, since the were

[1] For the sake of simplicity, we assume that the acknowledgment messages will never get lost in the inverse link.

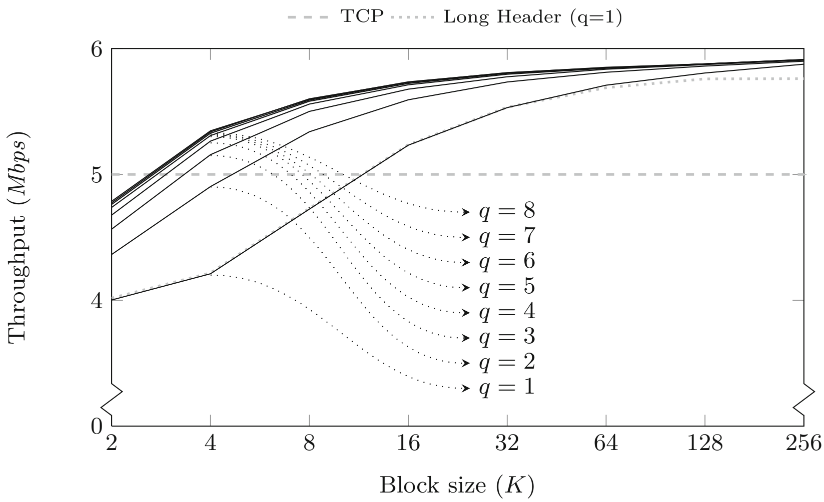


Fig. 7. Throughput observed for the RLC short header over an ideal single hop transmission

the ones leading to the optimum performance over ideal channels for the short and long headers, respectively. We can see that the performance of the proposed scheme is remarkably higher than the one observed when using the traditional TCP protocol, especially when the conditions of the wireless link are worse (the throughput is for instance approximately 10 times higher when the FER is 0.5). The figure also yields that there is not a relevant difference between the performances assessed with the various configurations of the RLC scheme. As can be seen, only the configuration with a slight lower throughput is the one using the original header and the larger Galois Field and block, since the required header length jeopardizes its performance, as was also seen over the ideal channel (see Fig. 6).

After assessing the potential benefits of the combination of RLC and UDP to provide a reliable communication service, and having measured the additional gain that can be achieved by modifying the original coding scheme, we exploit the framework that was integrated into the `ns-3` simulator in order to study the performance over wireless mesh networks, which lead to random topologies. In particular, we consider the characteristics that are enumerated below.

- A square area of 100 $m \times 100$ m.
- 32 Nodes randomly deployed throughout the scenario, following a *Poisson Point Process*, i.e. the x and y coordinates are selected with a uniform random variable between 0 and 100.
- We ensure that all network deployments are connected, discarding those whose underlying graph does not fulfill this constraint.
- Each node has a coverage area of 20 m, using the well-known disk radius model.
- We randomly select the source and the destination nodes.

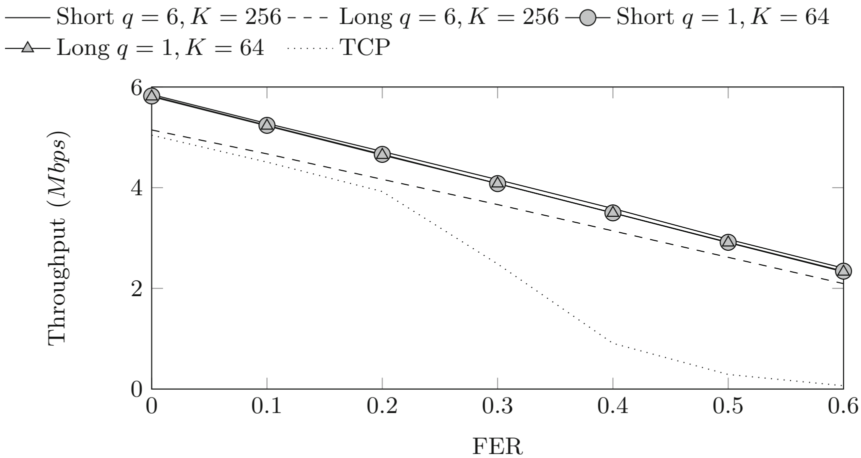


Fig. 8. Throughput observed for various configurations as a function of the wireless link quality

- We set the quality of the existing links (always with an euclidean distance lower than 20 m), by means of a FER randomly selected within the interval $[0.0 \ldots 0.6]$.
- The route between the source and the destination is established with the *Dijkstra* algorithm, which minimizes the overall cost, based on the ETX metric.
- We carry out a Monte-Carlo process, by randomly generating 1000 scenarios.

We use the following configuration for the RLC: $q = 6$ and $K = 256$. Figure 9 shows the throughput *cumulative distribution function* (cdf). As can be seen, the new header brings a performance gain of $\approx 16\,\%$ as compared with the original format, while the throughput is $\approx 1.7\times$ the one that was observed for the legacy TCP protocol.

It is worth highlighting the bad behavior exhibited by TCP either when the number of hops is increased and when the qualities of the wireless channels get worse. In fact, TCP is able to successfully finalize the connection (throughput greater than 0) in only $\approx 40\,\%$ of the scenarios.

To conclude the analysis, Fig. 10 shows the evolution of the throughput as a function of the number of hops of the route that is found between the source and destination nodes. First of all, we can again see the poor behavior exhibited by TCP when the quality of the wireless channels gets worse; for 1-hop routes, the performance of the RLC scheme is $\approx 3x$ the one observed for the legacy TCP case. On the other hand, the results also yield that the throughput that was measured with the new RLC header always surpasses the one observed for the initial RLC implementation.

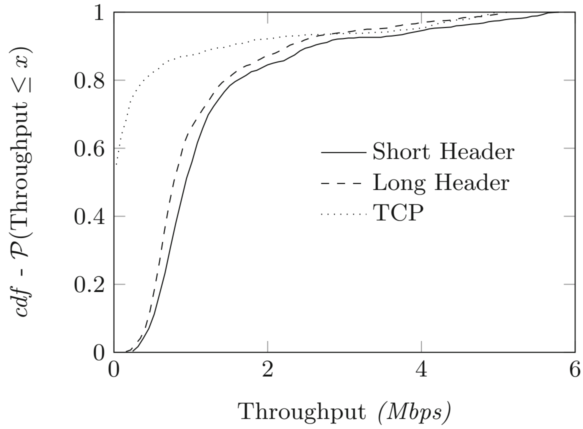


Fig. 9. *cdf* of the throughput observed over random wireless networks

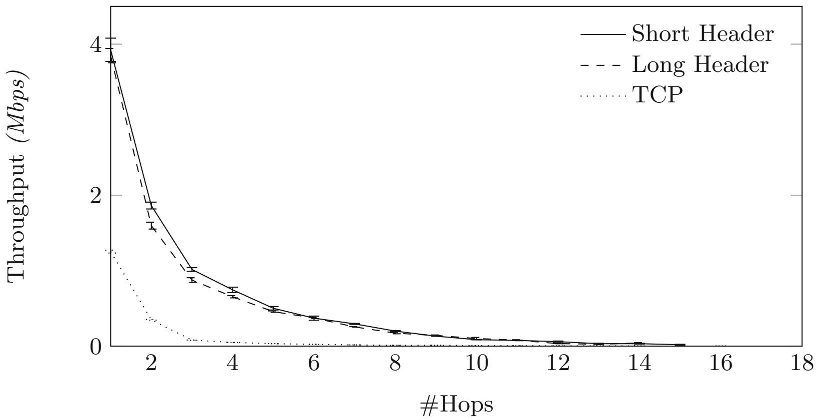


Fig. 10. Throughput vs. the # of hops

5 Conclusions

The use of Network Coding techniques to enhance the performance over wireless networks has received significant attention during the last years. The related research has covered various aspects, ranging from the analysis of the efficiency of coding/decoding procedures to the proposal of novel protocols able to promote these solutions. Some of the existing proposals belong to the so-called *Intra-flow* techniques group, since they are based on performing random linear combinations of packets that belong to the same data flow. Some of these works have discussed the relevance of using large Galois Fields for the coding/decoding purposes.

In a previous work [9], we proposed the combination of a RLC scheme and the UDP protocol to offer reliable communications over wireless links, and now, we propose a modification in the header format to alleviate the required overhead. Instead of sending the full coefficient vector, we include the random seed that was used by the source to generate it; this allows having a fixed-length header and using larger block and Galois Field sizes, K and q, respectively. By means of an analytical model we have evaluated the impact of the performance and the influence of K and q; we also assessed the relevance of using larger field sizes, which is favored by the use of the novel header format.

Afterwards, we have analyzed the performance of the two approaches over a single wireless link, comparing it with that observed with the legacy TCP protocol, which always showed much lower performances. The new header allows keeping the throughput regardless of the block size, while for the initial RLC scheme a strong reduction was observed. Finally, we have also assessed the performance of the RLC scheme over random network topologies (wireless mesh networks). We can highlight the low TCP performance over either low quality links or long routes. The throughput observed for the RLC scheme is, in average, approximately 1.7 times higher.

This work has also opened some new items that will be taken in our future research. Special attention will be paid to the integration of the RLC scheme with opportunistic routing techniques, which have been shown to provide relevant benefits over wireless mesh networks [5,14]. In addition, we will explore the possibilities that are brought about if the forwarding nodes take a more active role, by *re-coding* the packets previously stored previously.

Acknowledgments. This work has been supported by the Spanish Government by its funding through the project **COSAIF**, *"Connectivity as a Service: Access for the Internet of the Future"* (TEC2012-38754-C02-01).

References

1. The ns-3 network simulator. http://www.nsnam.org/
2. Ahlswede, R., Cai, N., Li, S.Y., Yeung, R.: Network information flow. IEEE Trans. Inf. Theory **46**(4), 1204–1216 (2000)
3. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh networks: a survey. Comput. Netw. **47**(4), 445–487 (2005). http://www.sciencedirect.com/science/article/pii/S1389128604003457
4. Albrecht, M.R., Bard, G.V., Hart, W.: Efficient multiplication of dense matrices over GF(2). CoRR abs/0811.1714 (2008). http://arxiv.org/abs/0811.1714
5. Chachulski, S., Jennings, M., Katti, S., Katabi, D.: Trading structure for randomness in wireless opportunistic routing. SIGCOMM Comput. Commun. Rev. **37**(4), 169–180 (2007). http://doi.acm.org/10.1145/1282427.1282400
6. De Couto, D.S.J., Aguayo, D., Bicket, J., Morris, R.: A high-throughput path metric for multi-hop wireless routing. Wirel. Netw. **11**(4), 419–434 (2005). http://dx.doi.org/10.1007/s11276-005-1766-z
7. Floyd, S., Gurtov, A., Henderson, T.: The NewReno modification to TCP's fast recovery algorithm (2004)

8. Gomez, D., Hassayoun, S., Herren, A., Aguero, R., Ros, D.: Impact of network coding on TCP performance in wireless mesh networks. In: 2012 IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), pp. 777–782, September 2012
9. Gomez, D., Rodriguez, E., Aguero, R., Munoz, L.: Reliable communications over lossy wireless channels by means of the combination of UDP and random linear coding. In: 2014 IEEE Symposium on Computers and Communication (ISCC), pp. 1–6, June 2014
10. Heide, J., Pedersen, M.V., Fitzek, F.H., Médard, M.: On code parameters and coding vector representation for practical RNLC. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–5. IEEE (2011)
11. Ho, T., Medard, M., Shi, J., Effros, M., Karger, D.R.: On randomized network coding. In: Proceedings of the Annual Allerton Conference on Communication Control and Computing, vol. 41, pp. 11–20. The University 1998 (2003)
12. Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M., Crowcroft, J.: XORs in the air: practical wireless network coding. SIGCOMM Comput. Commun. Rev. **36**(4), 243–254 (2006). http://doi.acm.org/10.1145/1151659.1159942
13. Kim, M., Cloud, J., ParandehGheibi, A., Urbina, L., Fouli, K., Leith, D., Médard, M.: Network coded TCP (CTCP) (2012). arXiv preprint arXiv:1212.2291
14. Krigslund, J., Hansen, J., Hundeboll, M., Lucani, D., Fitzek, F.: CORE: COPE with MORE in wireless meshed networks. In: 2013 IEEE 77th Vehicular Technology Conference (VTC Spring), pp. 1–6, June 2013
15. Lefevre, F., Vivier, G.: Understanding TCP's behavior over wireless links. In: 2000 Symposium on Communications and Vehicular Technology, SCVT-2000, pp. 123–130 (2000)
16. Li, S.Y., Yeung, R.W., Cai, N.: Linear network coding. IEEE Trans. Inf. Theory **49**(2), 371–381 (2003)
17. Trullols-Cruces, O., Barcelo-Ordinas, J., Fiore, M.: Exact decoding probability under random linear network coding. IEEE Commun. Lett. **15**(1), 67–69 (2011)
18. Zorzi, M., Chockalingam, A., Rao, R.: Throughput analysis of TCP on channels with memory. IEEE J. Sel. Areas Commun. **18**(7), 1289–1300 (2000)