Performance Analysis of Early-HARQ for Finite Block-Length Packet Transmission

João Nadas*, Paulo Klaine*, Lei Zhang*, Guodong Zhao*, Muhammad Imran* and Richard Souza†
*Communication, sensing and imaging (CSI) group, School of Engineering, University of Glasgow, U.K.
{j.battitsella-nadas.1, p.valente-klaine.1}@research.gla.ac.uk, {lei.zhang, guodong.zhao, muhammad.imran}@glasgow.ac.uk
†Circuits and Signal Processing Lab (LINSE), Federal University of Santa Catarina, Florianópolis-SC, Brazil
richard.demo@ufsc.br

***Abstract*—Traditional retransmission protocols require the receiver to decode the entire packet before sending feedback signals, which may not be a viable solution for ultra-reliable and low-latency communications (URLLC) as it may cause a significant delay. To address this issue, early hybrid automatic repeat request (E-HARQ) has been proposed as an alternative to reduce the processing time at the receiver and send the feedback signal as quickly as possible. In this work, we present a framework for analyzing the performance of ARQ protocols considering URLLC requirements for finite block-length packet transmission and use it to evaluate the performance of E-HARQ from an information theory perspective, comparing it to simple ARQ. The results show that this new class of retransmission protocols can significantly improve the performance of URLLC systems.**

I. Introduction

The next generation of cellular communications, 5G, is foreseen to enable several different applications and use-cases which were previously not supported by cellular networks [1]. It will not only bring improvements to current networks via enhanced mobile broadband but also provide the necessary infrastructure for mission-critical machine type communications (cMTC) by introducing the support for ultra-reliable and low-latency communications (URLLC) [2].

However, current cellular technology being used by Long Term Evolution (LTE) has been, in many cases, designed without the necessary regard for latency sensitive applications and thus protocols throughout the entire communication stack have to be redesigned in order to make URLLC a reality [3], [4]. One such example is the use of retransmissions, via automatic repeat request (ARQ) and hybrid-ARQ (HARQ) protocols, which have the potential to provide efficient use of resources [5].

In time diversity techniques, a feedback channel is commonly used such that the transmitter only sends a new copy of the message in case the receiver has failed to decode the previous attempt. This, in turn, allows for a very efficient use of resources as in most cases the average number of attempts is very close to one [5]. Therefore, when using ARQ we are exploiting the advantage of consuming only the necessary resources when compared to other diversity techniques, such as spatial or frequency diversity [5]. Moreover, retransmission protocols bring an attractive feature as they provide full diversity at low cost since often the average number of attempts used to convey a message is very close to one [5]. Nonetheless, to harvest such benefits in the context of URLLC these protocols have to be re-imagined as the diversity gains are provided at the cost of added delay, which might be intolerable in this case. An analysis of sources of delay in LTE networks has been presented by the 3rd Generation Partnership Project (3GPP) study [6] and the minimum processing delays that can be expected when using LTE with the traditional turbo encoding are on the order of 4 ms. This is intolerable for low-latency applications envisioned to be supported by 5G, such as several tasks needed for wireless factory automation [7].

Several researchers have proposed enhancements to ARQ technologies to enable their use in URLLC applications. Shariatmadari *et.al.* propose in [8] a resource allocation strategy between control and data plan in the context of URLLC with the possibility of one retransmission. They propose a sub-optimal but computationally feasible approach to improve the system performance both in ARQ and HARQ systems. In [9], on the other hand, the same authors present a scheme which uses asymmetric feedback signal detection between acknowledgment (ACK) and non-acknowledgment (NACK) for up to one retransmission. This enables a better protection of the NACK signal which ensures that failed transmissions are properly detected at the transmitter. In our previous work [5] we have shown that by using an optimal number of allowed transmission attempts it is possible to use the energy resources more effectively while still meeting stringent reliability and latency requirements. HARQ performance is compared to frequency diversity in a one-shot transmission (which does not incur in a latency increase) and we showed that Chase combining HARQ (CC-HARQ) still brings improvements in energy efficiency when properly designed.

Another possible strategy to enable the use of ARQ protocols is to use early feedback to improve latency. Traditional ARQ protocols were designed with the goal of improving the reliability of the system without major concerns for the excess latency, and so one might ask: is there anything in the protocol which could be optimized in order to improve its latency performance? With that in mind, E-HARQ was designed to improve the performance of the system while still meeting latency requirements.

E-HARQ has been studied by several works [10], [11], [12], [13]. The effect of incorrect predictions in the system performance under two different operating modes is analyzed in [10]

while in [11] the authors focus on designing a predictor to estimate the block error rate from likelihood ratios, instead of waiting for the turbo decoder to operate and study the statistics of false positives in using their predictor from a coding and modulation perspective. Meanwhile, the authors in [12] use machine learning techniques to determine the occurrence of errors using a predictor of their own design. Using different encoding mechanisms to enable early detection is another possibility to reduce the latency in HARQ communications. For instance, in [13] the authors propose using low-density parity-checks (LDPC) encoding to predict when an error will occur based on early stages of the decoder. They are able to achieve sub-millisecond latency with error rates on the order of 10^{-4}.

Unlike [11], [12], [13] which focus on designing predictors for failure rate from a coding and modulation perspective, in this paper we analyze the performance of E-HARQ in various scenarios from an information theory point of view. Moreover, we consider the effect of the finite block length (FBL) on the achievable rates which is also different from [10], as the authors analyze the impact of incorrect predictions on the performance considering practical codes. The advantage of studying the behavior from a channel capacity point of view is to devise a benchmark on achievable error rates to test practical implementations against.

The contribution of this work is a framework to evaluate ARQ techniques under URLLC constraints. Using the proposed framework, we show that E-HARQ enables the use of more transmission attempts when compared to traditional ARQ protocols while still meeting stringent latency requirements. We compare its performance with traditional simple ARQ (S-ARQ) in several different scenarios, considering varied target latencies, line-of-sight (LOS) and non-LOS (NLOS) channel conditions, and different levels of average signal to noise (SNR) ratio.

The remainder of this paper is organized as follows, Section II presents the system model, Section III presents the proposed framework, Section IV contains numerical results and Section V concludes the paper.

II. System Model

A. Communication Model

In this work, we consider a block-fading model where the received message $\boldsymbol{y}$ is expressed as

$$\boldsymbol{y} = \sqrt{\bar{\gamma}} h \boldsymbol{x} + \boldsymbol{w}, \tag{1}$$

where $\boldsymbol{x}$ is the transmitted message with unit energy, $\boldsymbol{w}$ is the additive white Gaussian noise (AWGN) with normalized power, h is a random channel fading which follows Nakagami-m distribution, and $\bar{\gamma}$ is the average input SNR.

B. Average Error Probability

In the context of cMTC, one important use-case of URLLC, short messages are exchanged, wherein the asymptotic approximation for the average error probability (*i.e.* the probability of outage) might be flawed due to assuming an infinite block length [14]. Therefore, we use the normal approximation for the achievable rate, which considers the length of the message block and is expressed as [15]

$$R = C(\gamma) - \sqrt{\frac{V(\gamma)}{n}} Q^{-1}(\epsilon) + \frac{1}{2n}\log_2(n), \tag{2}$$

where R is the code rate in bits per channel use, C is the channel asymptotic capacity, V is the channel dispersion, $\gamma = \bar{\gamma} h^2$ is the instantaneous SNR, n is the block length in channel uses, $Q(.)$ is the cumulative distribution function (CDF) of the standardized normal distribution and ϵ is the error rate.

Since the block-fading channel is conditionally ergodic with respect to h, the average error probability $\bar{\epsilon}_1$ can be determined by integrating the error rate for all possible channel realizations, similarly to what is done in [16], yielding

$$\bar{\epsilon}_1 = \int_{\mathbb{R}^+} Q\left(\sqrt{\frac{n}{V(\gamma)}}\left(C(\gamma) + \frac{1}{2n}\log_2(n)\right)\right) p_\gamma(\gamma) d\gamma, \tag{3}$$

where p_γ is the probability density function of γ which, for a Nakagami-m channel, is given by

$$p_\gamma(x) = \frac{m^2 x^{m-1}}{\Gamma(m)\bar{\gamma}^m} \mathrm{e}^{\frac{-mx}{\bar{\gamma}}}, \tag{4}$$

where m is the Nakagami-m parameter which is related to the amount of LOS, $\Gamma(.)$ is the gamma function and e is Euler's constant.

To improve the error rate performance, it is common to use diversity strategies, which consist of sending copies of the message via uncorrelated channels [17]. Since the channels are uncorrelated, the probability of all the channels yielding a low γ is smaller than that of any individual channel. One popular diversity strategy is the use of retransmissions, wherein diversity is achieved by sending copies of the message at different times. In a simple implementation, the receiver can discard any messages which it fails to decode and wait for a new transmission. Therefore, the average error probability is the product of the error probabilities of each attempt and, as the new attempts are performed in an uncorrelated channel, is

$$\bar{\epsilon}\,(z) = \bar{\epsilon}_1^z, \tag{5}$$

where z is the maximum number of allowed attempts. Fig. 1 shows the average error probability for the NLOS case (*i.e.* Rayleigh fading, $m = 1$), considering $n = 100$ channel uses and $R = 0.1$ bits per channel use, for z ranging from 1 to 4. As expected, adding diversity allows us to operate at much lower error probabilities, easily reaching error rate probabilities as low as 10^{-5} and beyond with only a few extra attempts.

C. Maximum Latency

The results presented in Fig. 1 are well known and work reasonably well in traditional systems. The trade-off here is gaining reliability at the cost of an increased latency, which is not a problem in many applications. However, this can be challenging in the context of URLLC, as both reliability and latency play important roles. In other words, some applications

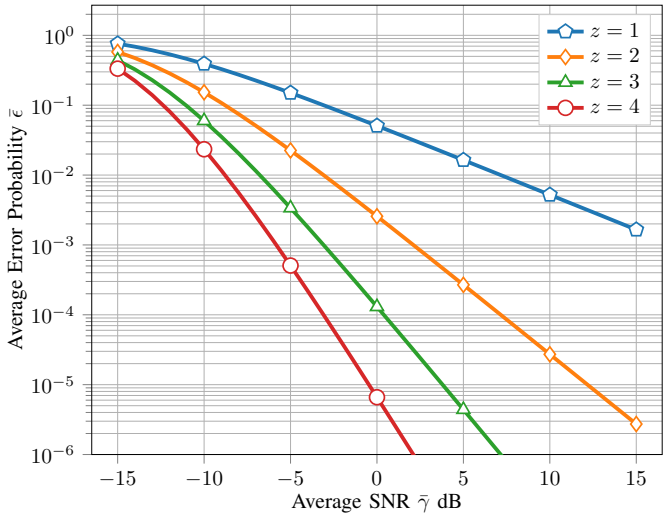

Fig. 1. Average error considering different number of attempts for $n = 100$ channel uses, $m = 1$, indicating NLOS, and $R = 0.1$ bits per channel use.

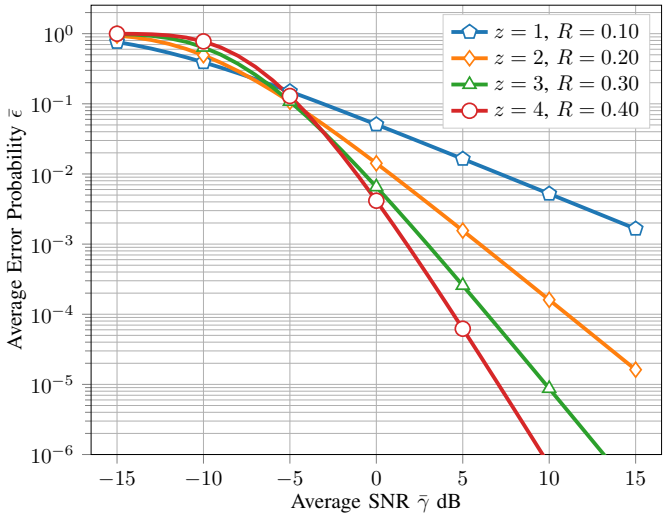

Fig. 2. Average error considering different number of attempts for $n = 100$ channel uses, $m = 1$, indicating NLOS, and increasing the rate to compensate the latency of allowing more attempts.

do not have the necessary degree of freedom to exploit a trade-off between these two parameters.

Therefore, in order to use retransmissions in this context we must compensate for the excess latency in another fashion. For instance, we could increase the bandwidth of the system as this would allow us to convey more information per channel use, shortening the time on air. This, however, is only possible if there is available spectrum, however as it is an expensive commodity in wireless communications, this might not be a viable solution. Thus, we must seek other means to enable time diversity for URLLC applications.

One interesting possibility is to increase the code rate such that it is possible to fit all z attempts within the target latency. This is what we are exploring in this work, as in our previous work [5]. In Fig. 2 we show the behavior of the error curves when we compensate the additional attempts by increasing the code rate accordingly. Note that, specially at high SNR, increasing the diversity yields a better performance in terms of average error probability even with the increase in R.

III. Performance Evaluation Framework

Considering L_{T} as the total number of bits to be transmitted, both for forward and feedback messages, we determine the minimum code rate (in bits per channel use) required to meet the target latency (λ) as

$$R = \frac{L_{\mathrm{T}}}{n(\lambda)}, \tag{6}$$

which is a function of n, the number of available channel uses for each attempt. Fig. 3 illustrates the idea for the case with $z = 3$ and considering that δ seconds are being used in total to decode messages.

The proposed framework consists of determining the number of channel uses available for each attempt according to the desired scheme and then determining the minimum communication rate according to (6). Next, the values of n and

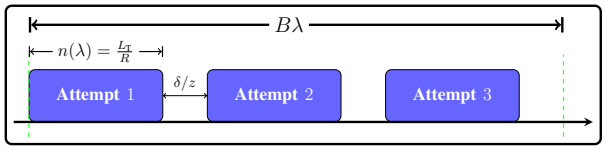

Fig. 3. Example of how using R obtained from (6) makes it possible to meet the target latency. Note that here $z = 3$ and $n(\lambda) = (\lambda - \delta)B/z$, such that in total δ seconds are being used to decode the messages.

R are used for the purpose of measuring system performance under the desired metric, for example (5). Below we show how to determine n for S-ARQ and E-HARQ in order to evaluate their performance.

In S-ARQ, when the receiver successfully decodes the message it sends back an acknowledgment (ACK), whereas if it fails it sends a non-acknowledgment (NACK) instead. Upon receiving a NACK, the transmitter sends a new copy of the message and the receiver discards the signal from the first message and repeats the process until z attempts have elapsed or it successfully decodes the message. In S-ARQ the receiver has to decode all the bits sent by the transmitter at every attempt before deciding to send an ACK or NACK, such that it has to use up to

$$\delta_{\mathrm{s}} = z\frac{L_{\mathrm{T}}\phi}{f_{\mathrm{apu}}} \tag{7}$$

seconds out of the latency budget to decode messages. Here, ϕ is the number of operations per bit required for decoding the messages and f_{apu} is the arithmetic logic unit clock frequency. Therefore, $n_{\mathrm{s}}(\lambda)$, the number of available channel uses for each S-ARQ attempt, becomes

$$n_{\mathrm{s}}(\lambda) = (\lambda - \delta_{\mathrm{s}})\frac{B}{z}, \tag{8}$$

where B is the communication bandwidth.

On the other hand, when considering E-HARQ, the receiver uses some strategy to predict whether or not an error will occur and sends an early feedback, thus saving important latency resources. We propose to measure the instantaneous SNR and if γ is below a certain threshold, the receiver sends the NACK back to the transmitter without attempting to decode the message. Since the instantaneous SNR is low, there is a high probability of failure and wasting time with complex decoding algorithms might not be the best approach, in particular when latency is so critical. This way, the payload only has to be decoded once, when γ is above this threshold. Therefore, δ_{e}, the amount of time required for decoding messages in the E-HARQ case, is determined by adding the time to decode the payload once with the amount of time to decode the remaining bits (headers, overhead and feedback signals) z times, yielding

$$\delta_{\mathrm{e}} = \frac{(z(L_{\mathrm{T}} - L_{\mathrm{P}}) + L_{\mathrm{P}})\phi}{f_{\mathrm{apu}}}, \tag{9}$$

where L_{P} is the payload length. Thus, the number of channel uses available in each E-HARQ attempt, $n_{\mathrm{e}}(\lambda)$, is

$$n_{\mathrm{e}}(\lambda) = (\lambda - \delta_{\mathrm{e}})\frac{B}{z}. \tag{10}$$

We can use the equations derived here to make certain predictions on the performance of each scheme. For instance, analyzing (8) and (10) asymptotically, when $\lambda \to \inf$, $n_{\mathrm{s}} \approx n_{\mathrm{e}}$ and the performance of both schemes will be very similar. This explains why the protocols designed without latency in mind are sub-optimal when considering URLLC. Moreover, taking the partial derivative of n_{s} with respect to z yields

$$\frac{\partial n_{\mathrm{s}}}{\partial z} = -\frac{B\lambda}{z^2}, \tag{11}$$

while for n_{e} we have

$$\frac{\partial n_{\mathrm{e}}}{\partial z} = -\frac{B\left(\lambda - \frac{L_{\mathrm{P}}\phi}{f_{\mathrm{apu}}}\right)}{z^2}. \tag{12}$$

Considering that in any system $L_{\mathrm{P}} > 0$, it is possible to see that (11) decreases faster than (12), thus proving that S-ARQ will always have access to fewer channel uses when compared to E-HARQ. Therefore, the former requires a larger coding rate to deliver the same latency performance. This provides mathematical guarantees that E-HARQ outperforms S-ARQ for any z.

Although we are comparing E-HARQ with S-ARQ in this example, the proposed framework could be used for more complex ARQ mechanisms such as CC-HARQ or incremental redundancy HARQ. In those cases, n and R would be determined in similar fashion, the difference would be in the function used to determine the error.

IV. Numerical Results

In this section, we use the proposed framework to compare the error rate performance of S-ARQ and E-HARQ via numerical simulations. This highlights the potential performance improvements that using E-HARQ can bring to URLLC applications and moreover validate the predictions made using the proposed framework. For the purpose of determining ϕ, we are assuming that the communication is done using turbo coding, as this is the standard channel coding used in current LTE technology. The parameters used in the simulations, unless otherwise stated, are the ones presented in Table I.

TABLE I
Simulation Parameters

Parameter	Value
Bandwidth (B)	200.0 KHz
Maximum Link Latency (λ)	5.0 ms
Payload Length (L_{P})	256 bits
Total Bits Exchanged (L_{T})	289 bits
Decoder Complexity (ϕ)	1536 operations/bit [18]
APU frequency (f_{apu})	900.0 MHz

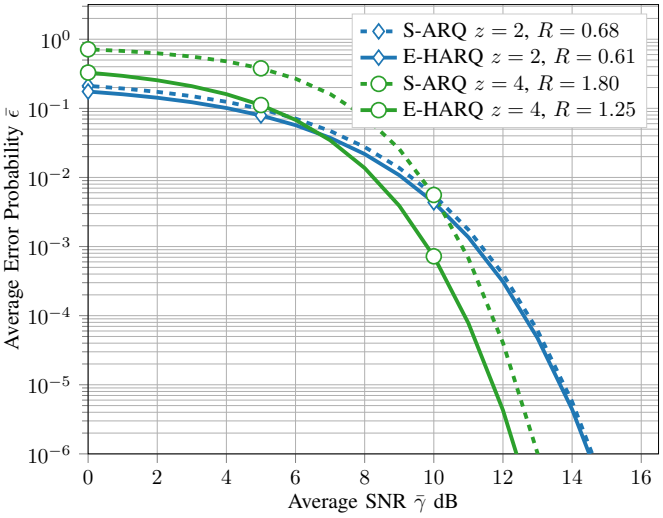

Fig. 4. Block error rate versus SNR for S-ARQ and E-HARQ for $z = 2$ and $z = 4$ and with $m = 1$.

Fig. 4 shows the average error probability as a function of the average SNR for S-ARQ and E-HARQ, for the NLOS ($m = 1$) case, while Fig. 5 has the same information when there is some LOS ($m = 3$). As we can observe, the benefits of E-HARQ are more pronounced for larger values of z, since the difference in coding rate is larger. Moreover, when m grows, meaning that there is more LOS, the performance gap also increases.

Next we compare in Fig. 6 the error rate performance of both schemes when the target latency varies, for two levels of average SNR (0 and 10 dB) and for $z = 3$. As we can observe, for a more strict latency the performance difference is greater since the latency budget is more stringent. When the target latency increases, both schemes tend to the same performance, as predicted by the asymptotic analysis. Moreover, the gap is almost equivalent regardless of the average SNR considered, showing that the gains of using the proposed scheme can be used in various different scenarios, such as applications with stringent power limitations (*e.g.* cognitive radio) or applications with access to more energy (*e.g.* cyber-physical systems).

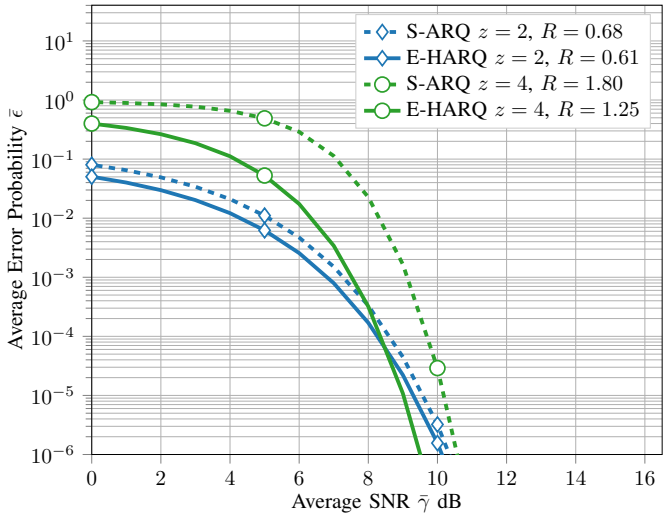

Fig. 5. Block error rate versus SNR for S-ARQ and E-HARQ for $z = 2$ and $z = 4$ and with $m = 3$.

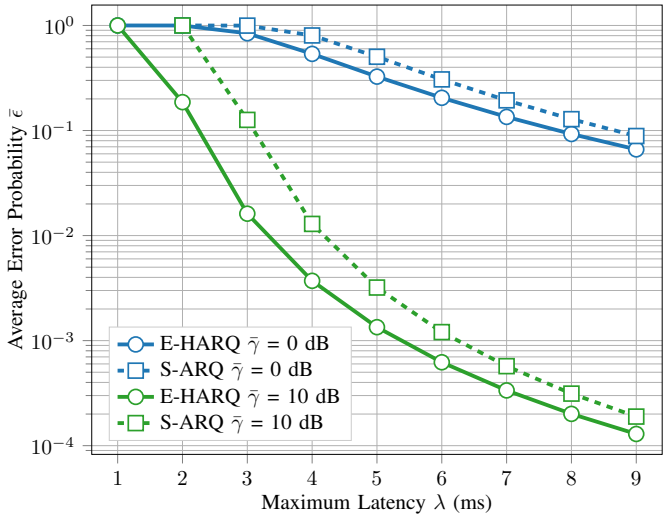

Fig. 6. Performance comparison between S-ARQ and E-HARQ for $z = 3$ transmission attempts and different values of λ.

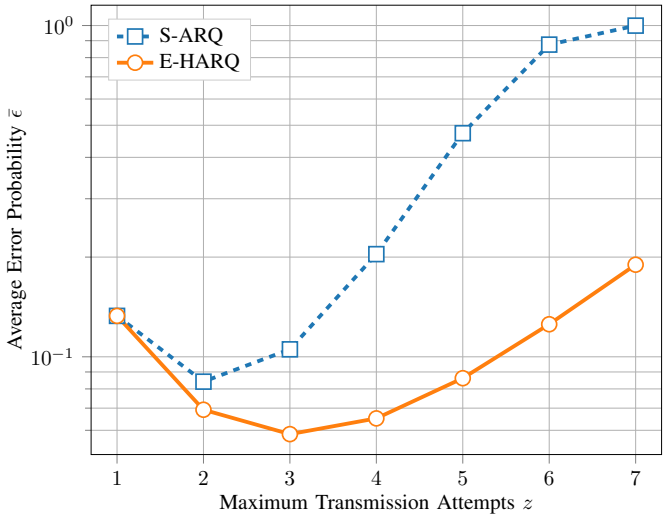

Fig. 7. Performance comparison between S-ARQ and E-HARQ for $\bar{\gamma} = 0$ dB and different values of z.

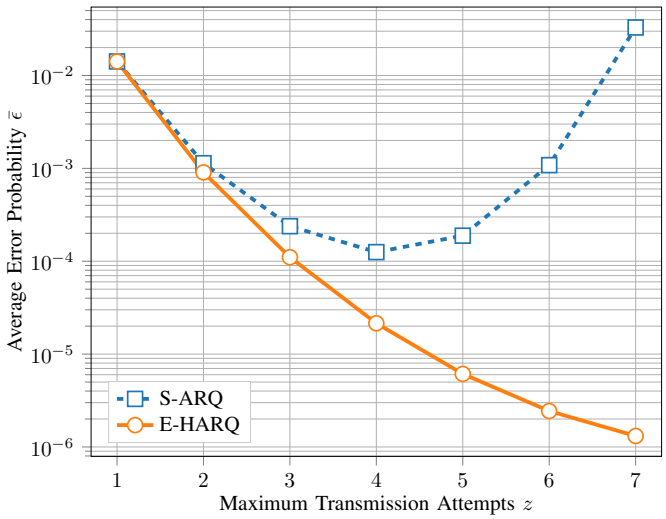

Fig. 8. Performance comparison between S-ARQ and E-HARQ at high SNR ($\bar{\gamma} = 10$ dB) and different values of z.

To show the gains when increasing the number of allowed attempts, we present in Figs. 7 and 8 the average error probabilities for both schemes when $m = 1$ and considering $\bar{\gamma} = 0$ dB and 10 dB, respectively. In Fig. 7 we can clearly observe that increasing the number of allowed attempts does not scale indefinitely, as at some point the required coding rate will overcome the added gains from increased diversity. The more stringent the link budget (*e.g.* smaller average SNR) the earlier this tipping point will occur. In Fig. 8, on the other hand, since it depicts a high SNR scenario ($\bar{\gamma} = 10$ dB), the tipping point is only plotted for S-ARQ, as it occurs for larger z in the case of E-HARQ.

V. Conclusion

In this paper we have proposed a framework for evaluating the performance of ARQ schemes considering URLLC strict latency requirements. Moreover, we have used the proposed framework to analyze the performance of E-HARQ, a retransmission strategy designed specifically for URLLC applications, and S-ARQ, a traditional ARQ protocol. Results show that significant performance improvements can be obtained by using a protocol designed specifically considering stringent latency constraints and show promising potential to enable URLLC in future 5G networks.

Acknowledgement

This work was supported by EPSRC Global Challenges Research Fund the DARE Project under Grant EP/P028764/1, and by CNPq, Brazil.

REFERENCES

[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.

[2] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view," *arXiv preprint arXiv:1804.05057*, 2018.

[3] G. Pocovi, H. Shariatmadari, G. Berardinelli, K. Pedersen, J. Steiner, and Z. Li, "Achieving ultra-reliable low-latency communications: Challenges and envisioned system enhancements," *IEEE Netw.*, vol. 32, no. 2, pp. 8–15, Mar. 2018.

[4] H. Shariatmadari, S. Iraji, R. Jantti, P. Popovski, Z. Li, and M. A. Uusitalo, "Fifth-generation control channel design: Achieving ultrareliable low-latency communications," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 84–93, Jun. 2018.

[5] J. P. B. Nadas, O. Onireti, R. D. Souza, H. Alves, G. Brante, and M. A. Imran, "Performance analysis of hybrid ARQ for ultra-reliable low latency communications," *IEEE Sensors J.*, p. 1, 2019.

[6] 3GPP, "Study on latency reduction techniques for LTE," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.881, 07 2016, version 14.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/-36881.htm

[7] S. Ashraf, Y. P. E. Wang, S. Eldessoki, B. Holfeld, D. Parruca, M. Serror, and J. Gross, "From radio design to system evaluations for ultra-reliable and low-latency communication," in *Proc. European Wireless 2017; 23th European Wireless Conf*, May 2017, pp. 1–8.

[8] H. Shariatmadari, R. Duan, S. Iraji, Z. Li, M. A. Uusitalo, and R. Jäntti, "Resource allocations for ultra-reliable low-latency communications," *Int. J. Wireless Inf. Networks*, pp. 1–11, 2017.

[9] H. Shariatmadari, R. Duan, S. Iraji, R. Jäntti, Z. Li, and M. A. Uusitalo, "Asymmetric ACK/NACK detection for ultra - reliable low - latency communications," in *Proc. European Conf. Networks and Communications (EuCNC)*, Jun. 2018, pp. 1–166.

[10] G. Berardinelli, S. R. Khosravirad, K. I. Pedersen, F. Frederiksen, and P. Mogensen, "On the benefits of early HARQ feedback with non-ideal prediction in 5G networks," in *Proc. Int. Symp. Wireless Communication Systems (ISWCS)*, Sep. 2016, pp. 11–15.

[11] ——, "Enabling early HARQ feedback in 5G networks," in *Proc. IEEE 83rd Vehicular Technology Conf. (VTC Spring)*, May 2016, pp. 1–5.

[12] N. Strodthoff, B. Göktepe, T. Schierl, C. Hellge, and W. Samek, "Enhanced machine learning techniques for early HARQ feedback prediction in 5G," *arXiv preprint arXiv:1807.10495*, 2018.

[13] B. Goektepe, S. Faehse, L. Thiele, T. Schierl, and C. Hellge, "Subcode-based early HARQ for 5G," in *Proc. IEEE Int. Conf. Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[14] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, 2010.

[15] M. Shirvanimoghaddam, M. S. Mohamadi, R. Abbas, A. Minja, B. Matuz, G. Han, Z. Lin, Y. Li, S. Johnson, and B. Vucetic, "Short block-length codes for ultra-reliable low-latency communications," *arXiv preprint arXiv:1802.09166*, 2018.

[16] P. Mary, J. M. Gorce, A. Unsal, and H. V. Poor, "Finite blocklength information theory: What is the practical impact on wireless communications?" in *IEEE Globecom Workshops*, Dec. 2016, pp. 1–6.

[17] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[18] M. Sybis, K. Wesolowski, K. Jayasinghe, V. Venkatasubramanian, and V. Vukadinovic, "Channel coding for ultra-reliable low-latency communication in 5G systems," in *Proc. IEEE 84th Vehicular Technology Conf. (VTC-Fall)*, Sep. 2016, pp. 1–5.