

# Improving Latency Using Codes in Mission-Critical Communication

Sunghyun Kim\* and Jee-Hyeon Na\*

\*Electronics Telecommunications Research Institute, Daejeon, South Korea

koishkim@etri.re.kr, jhna@etri.re.kr

**Abstract**—Delays are crucial in mission-critical applications such as safety networks that exchange information in a short period of time to prevent impending accidents. When such applications are implemented in wireless systems, delays are inevitable, due to a defining characteristic of unstable wireless channels. Information signals sent over the air are inherently prone to distortion, thus often need to be retransmitted to ensure their successful decoding. It is crucial in mission-critical applications that the delays be short enough, although it may come with a cost in other aspects of communication, for example, reduced data rates. In this paper, we propose a novel technique that improves latency using codes. We consider the established cellular systems that are based on orthogonal frequency-division multiplexing (OFDM), and employ maximum-distance separable (MDS) codes for signals that are sent via parallel orthogonal carriers. As our main result, we show that compared to a naïve approach that applies simple repetition coding schemes, our technique can achieve both better latency and data rate. This result suggests that by employing sophisticated coding schemes, one can improve latency without making much sacrifice in data rate.

**Keywords**—Latency-rate trade-off, Mission-critical communication, MDS codes, OFDM systems, Packet failures

## I. INTRODUCTION

In mission-critical communication scenarios, such as vehicular safety networks that can automatically detect and prevent impending accidents by exchanging small amounts of information among vehicles, shorter latencies are a critical requirement to meet, perhaps much more than higher data rates. The amount of information to convey may be small, but it is crucial that the information be delivered in time.

As more and more mobile devices are being connected with wireless communication capabilities, a variety of mission-critical applications, most likely regarding safety, will emerge. Dealing with safety, they will need to be reliable anytime to users who can be anywhere. This makes current established cellular systems, thanks to their wide and active deployment, a viable candidate to play a central role in such applications.

The cellular systems, which are based on orthogonal frequency-division multiplexing (OFDM) where streams of information (data packets) are delivered in parallel via multiple orthogonal carrier signals, have been a great success. But they are not flawless. Fluctuating unstable channels are a defining characteristic in wireless communication, thus signals sent over the air are fundamentally susceptible to distortion.

This leads to packet failures even with error-correcting codes. To cope with inevitable packet failures, the systems employ retransmission schemes to ensure packet delivery, and it is clear that retransmissions lead to undesirable latency.

A naïve but effective way to achieve shorter latencies would be to employ simple repetition coding schemes for data packets. One can make multiple copies for each data packet and send them in parallel. As long as at least one of the copies is successfully decoded at the destination, there is no need to retransmit the data packet, thereby achieving short latencies. This latency gain from improved reliability obviously comes with a cost: reduction in data rate.

One question naturally follows: is there any other way to achieve good latencies without crippling data rates to a serious extent? In this paper, we answer this question in the affirmative. To that end, we consider a point-to-point channel which consists of multiple parallel subchannels. To represent packet failures in wireless channels in a simple yet intuitive way, we assume that a packet sent through each subchannel results in a decoding failure with probability  $1-p$  and a successful decoding with probability  $p$ . We also assume that such failures take place independently over the subchannels. In this setting, we investigate three scenarios: in one, we do not employ any coding scheme over subchannels; in another, we employ a simple repetition coding scheme; in the other, we employ a coding scheme that is more sophisticated than repetition. We examine the data rate (packet transmission rate) and latency in all scenarios and compare them in detail.

As our main result, we first demonstrate that a simple repetition coding scheme can lead to improved latency, with reduced data rate being the price to pay. But more importantly, we show that by using more sophisticated coding schemes than mere repetition, yet not too complex to be implemented in practice, one can not only achieve better latency, but also sacrifice less in terms of data rate (similar techniques can be found, e.g., in [1]). This result suggests that employing advanced coding schemes for signals sent through orthogonal carriers can be very useful, opening up opportunities to strike a balance between data rate and latency.

**Related work:** The ideas of introducing redundancy to data without coding (simple uncoded replication) and coding (for example, MDS codes used in this work) in order to gain in latency have appeared in various contexts where accessing multiple nodes for data retrieval takes place. In [2], in the

context of distributed data centers, using erasure codes such as MDS codes to data chunks stored in a distributed manner can reduce the waiting time for a data access request in the queue to be processed. In [3], in the context of downloading large contents via limited bandwidth in peer-to-peer (P2P) networks, it has been shown that storing coded data fragments in the nodes can be more beneficial than uncoded ones. In addition to these applications, in this paper, we propose yet another promising one in which employing coding schemes to data whose parts can be accessed in parallel is helpful.

## II. PROBLEM FORMULATION

Figure 1 describes the bursty  $M$ -parallel point-to-point channel which consists of  $M$  orthogonal subchannels. The source node (transmitter) wishes to convey information reliably to the destination node (receiver). In this paper, we consider signals transmitted for a certain period of time to be equivalent to one packet sent in one time slot for the sake of simplicity. When the signals are distorted to a serious extent due to the channel instability during the time of transmission, the corresponding packet results in a decoding failure.

To represent decoding failures of packets in the simplest manner, we assume a transmitted packet results in a successful decoding with probability  $p$  and a failure with probability  $1-p$ . For ease of demonstration, from now we interchangeably denote the failure probability by  $q$ . We also assume that the failure probabilities are independent over the subchannels.

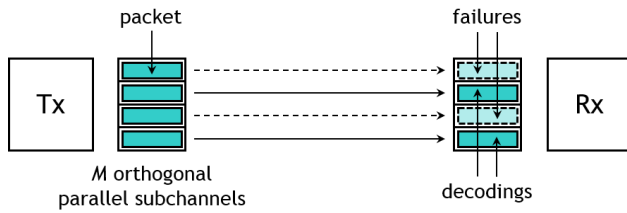


**Figure 1.** Bursty  $M$ -parallel point-to-point channel. Packets sent through subchannels are subject to decoding failures probabilistically.

Our primary interest of this work is to examine latency and data rate achievable in various scenarios. We define the packet transmission rate (which we refer to as “rate” from now interchangeably) as the expected number of data packets successfully delivered to the receiver per time slot. It is important for us to note that in counting the number of data packets, we do not include redundant packets that provide no further information to the destination node. We also define the latency as the expected number of time slots required until a designated number of data packets are decoded at the receiver. For  $k$  data packets, we refer to the latency as  $k$ -latency.

## III. MAIN RESULTS

In this section, we present our main results, deferring all relevant proofs to Section IV. First, for concreteness, in the form of a theorem, we express the packet transmission rate and latency achievable when we do not employ any coding scheme over the subchannels for the channel of interest.

**Theorem 1 (No coding scheme):** In the bursty  $M$ -parallel point-to-point channel, when no coding is employed over the

subchannels, the following packet transmission rate and  $k$ -latency are achievable<sup>1</sup>.

$$R_{No} = Mp, \quad L_{No} = \sum_{t=1}^{\infty} tP_{No}(t).$$

*Proof:* See Section IV-A.

Next, when we employ simple repetition coding, achievable rate and latency are given as follows.

**Theorem 2 (Repetition coding):** In the bursty  $M$ -parallel point-to-point channel, when repetition coding is employed over the subchannels, the following packet transmission rate and  $k$ -latency are achievable<sup>2</sup>.

$$R_{Rep} = k \left( 1 - (1-p)^{\frac{M}{k}} \right), \quad L_{Rep} = \sum_{t=1}^{\infty} tP_{Rep}(t).$$

*Proof:* See Section IV-B.

Last, when we employ a scheme that makes use of an  $(n, k)$ -MDS code, achievable rate and latency are given as follows.

**Theorem 3 (MDS coding):** In the bursty  $M$ -parallel point-to-point channel, when  $(M, k)$ -MDS coding is employed over the subchannels, the following packet transmission rate and  $k$ -latency are achievable<sup>3</sup>.

$$R_{MDS} = \sum_{s=0}^{k-1} w_s \left[ k \left( 1 - \sum_{r=s}^{k-1} \binom{M-s}{r-s} p^{r-s} (1-p)^{M-r} \right) + sp \right],$$

$$L_{MDS} = \sum_{t=1}^{\infty} tP_{MDS}(t).$$

*Proof:* See Section IV-C.

It is instructive to visualize the scheme used in each of the three scenarios of interest with examples. It can highlight key ideas of this paper, and also make the proofs in Section IV easier to follow. Now let us consider a channel that consists of four parallel subchannels.

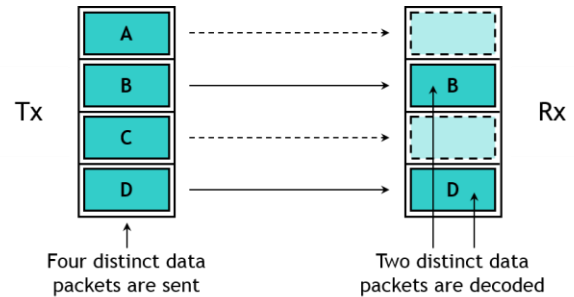


**Figure 2.** No coding employed over the subchannels.

Figure 2 illustrates the scenario in which no coding scheme is employed. Four distinct data packets are sent through the four subchannels. When a packet results in a decoding failure, then the destination node cannot recover the content in it. In order for the system to deliver the failed packet, say packet A, it has to retransmit the packet until its successful decoding.

<sup>1</sup>  $P_{No}(t) := (1-q^t)^k - (1-q^{t-1})^k$ .

<sup>2</sup>  $P_{Rep}(t) := \left(1 - q^{\frac{M}{k}t}\right)^k - \left(1 - q^{\frac{M}{k}(t-1)}\right)^k$ .

<sup>3</sup>  $P_{MDS}(t) := \sum_{i=0}^{k-1} \binom{M}{i} \left[ (q^{t-1})^{M-i} (1-q^{t-1})^i - (q^t)^{M-i} (1-q^t)^i \right],$

$w_s := \frac{\sum_{r=0}^{s-1} w_r \binom{M-r}{s-r} p^{s-r} q^{M-s}}{1 - q^{M-s}}, \forall s \in [1 : k-1], w_0$  is set to meet  $\sum_{s=0}^{k-1} w_s = 1$ .

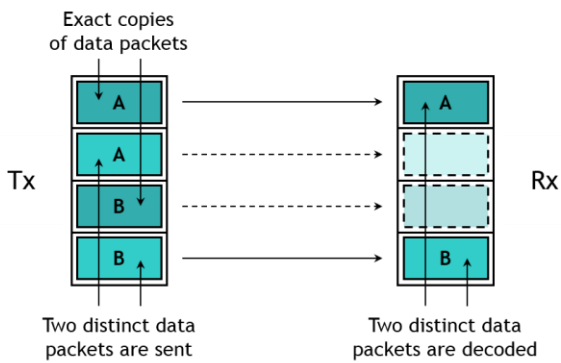


Figure 3. Repetition coding employed over the subchannels.

Figure 3 illustrates the scenario in which a simple repetition coding scheme is employed. Of the four subchannels, two are used to send distinct data packets, while the other two are used to send the exact copies of them. As a result, although some packets may result in a decoding failure, due to the introduced redundancies, the destination node can recover the contents of the failed packets. It is important to note that if all the copies of one data packet are decoded at the destination node, it leads to a waste of precious bandwidth resources. It is a cost in rate reduction to pay for improved latency.

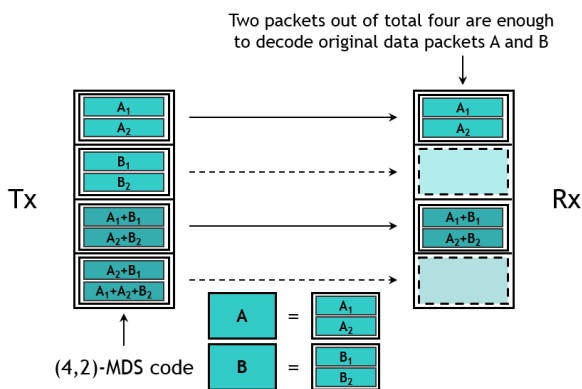


Figure 4. An  $(4, 2)$ -MDS code ([4]) employed over the subchannels.

Figure 4 illustrates the scenario in which a more advanced coding scheme is employed, compared to mere repetition. The system uses a  $(4, 2)$ -MDS code. When an  $(n, k)$ -MDS code is used, information contained in  $k$  data packets can be recovered as long as any  $k$  packets from the  $n$  total packets are intact. This figure is a clear example. Two original data packets A and B are divided into two subpackets,  $(A_1, A_2)$  and  $(B_1, B_2)$ . With appropriate linear combinations of the subpackets, two redundant packets are created. It is important to note that with at least any two intact packets out of the four total packets, the original data packets A and B can be recovered. As a result, in this example, the destination node can decode packets A and B retrieving all information the source node intends to convey.

What is it that could make using MDS codes lead to better performance? Suppose the third and fourth packets are not successfully decoded at the destination node in Figure 3. Then all copies of packet B are not available, thus it is impossible to recover data contained in packet B. On the other hand, when

an MDS code is used as in Figure 4, the original data can be recovered as long as at least two packets are available.

To examine the two performance measures (latency and rate), and the trade-offs between them in each scenario with varying degrees of channel stability, it is useful to look into latency and rate plots. In doing so, we continue to consider the same channel described previously. To make fair comparisons, in computing latency, we stick to 2-latency, the expected time slots needed to decode two distinct data packets A and B.

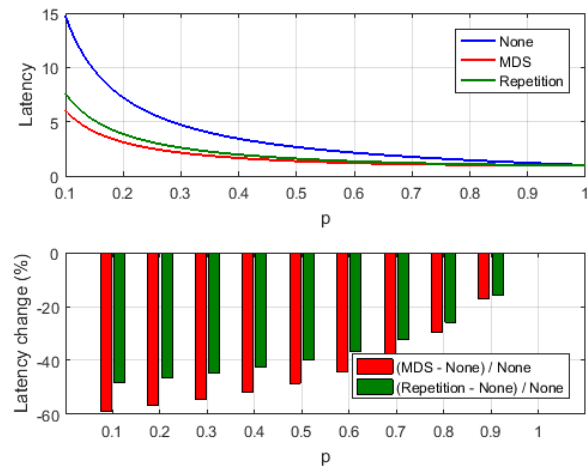


Figure 5. (Top) Latencies attained when employing (1) no coding (blue); (2) MDS coding (red); (3) repetition coding (green). (Bottom) Considering the case where no coding is employed as the baseline, latency changes in terms of percentage.

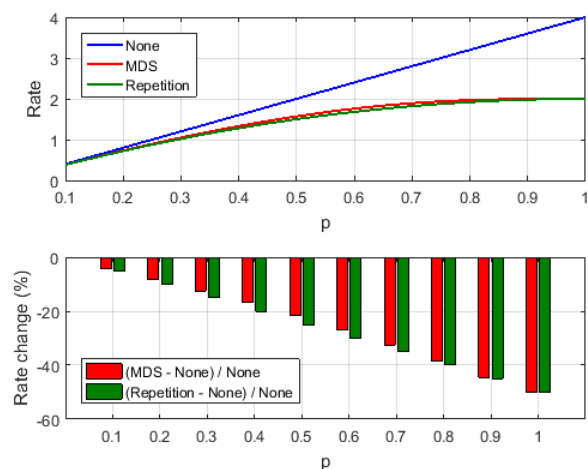


Figure 6. (Top) Rates attained when employing (1) no coding (blue); (2) MDS coding (red); (3) repetition coding (green). (Bottom) Considering the case where no coding is employed as the baseline, rate changes in terms of percentage.

Figure 5 illustrates the latencies in the three scenarios (top), and the changes in latency (bottom) in employing two coding schemes. First, one can see that both repetition coding and MDS coding reduce latencies to a great extent, possibly up to around 50% and 60% respectively for low  $p$ . In words, when the wireless channel is very unreliable causing frequent decoding failures, it is desirable to employ coding schemes in

parallel in OFDM wireless systems, particularly so in mission-critical applications where short delays are necessary. Furthermore, comparing the relative latency changes attained by repetition coding and MDS coding, with respect to the case where no coding is used, MDS coding always outperforms repetition coding. This result clearly demonstrates that MDS coding, which is more sophisticated than mere repetition, can be more gainful in terms of latency.

Figure 6 illustrates the rates in the three scenarios (top), and the changes in rate (bottom) in employing two coding schemes. One can see that MDS coding suffer from less rate reductions, compared to repetition coding. This figure, together with Figure 5, shows that by employing MDS coding, a more sophisticated scheme than a rather simple scheme based on repetition, one can achieve improved latency at a lower cost of rate reduction, especially for low  $p$ .

#### IV. PROOFS

In this section, we provide proofs for the theorems stated in Section III. We choose to provide concise versions.

##### A. Proof of Theorem 1

It is straightforward to prove Theorem 1. Since we do not introduce any redundancy in transmitting data packets, each packet to be sent is distinct, meaning that the information in each packet is independent of that in the other packets sent at the same time. Hence, of all total  $M$  distinct data packets to be sent through the  $M$  parallel subchannels, on average,  $Mp$  data packets are decoded at the destination node at any given time slot. Therefore, the achievable rate is  $Mp$ .

Our definition of  $k$ -latency is the expected number of time slots required in order to ensure  $k$  data packets to be decoded at the destination node. Let us denote by  $T$  the time slot in which all intended  $k$  data packets are decoded.

$$T = \max(T_1, T_2, \dots, T_k).$$

Then, we derive the cumulative distribution of an event in which all  $k$  data packets are decoded by time slot  $t$ .

$$\begin{aligned} \Pr[T \leq t] &= \Pr[T_1 \leq t] \times \Pr[T_2 \leq t] \times \dots \times \Pr[T_k \leq t] \\ &= (1 - \Pr[T_1 > t])^k \\ &= (1 - (1 - p)^t)^k. \end{aligned}$$

We get the distribution of  $T$  by computing the difference of the cumulative distribution.

$$\begin{aligned} \Pr[T = t] &= \Pr[T \leq t] - \Pr[T \leq t - 1] \\ &= (1 - (1 - p)^t)^k - (1 - (1 - p)^{t-1})^k. \end{aligned}$$

The latency is the expectation of  $T$ . This finishes the proof.

##### B. Proof of Theorem 2

Once the proof of Theorem 1 is clear, it is easy to extend it to prove Theorem 2. For simplicity, let us assume  $M$  is divisible by  $k$ . Then, intended  $k$  data packets are sent, each of which with  $M/k$  copies through  $M/k$  subchannels. In order for each data packet to be decoded at the destination node, it suffices for at least one copy of the packet to be intact. The probability of such an event is given as follows.

$$p' = 1 - (1 - p)^{\frac{M}{k}}.$$

Replacing  $M$  and  $p$  in the proof of Theorem 1 used to compute the achievable rate by  $k$  and  $p'$  respectively finishes the proof of the achievable rate.

Also, changing properly the probability of the event of at least one copy of each data packet is decoded by time slot  $t$  in the proof of Theorem 1 finishes the proof of the latency.

##### C. Proof of Theorem 3

We first explain the scheme that makes use of MDS codes in detail. In doing so, we set aside the issue of the existence of an  $(M, k)$ -MDS code, as it is out of scope in this paper.

In the scheme, at any given time, when less than  $k$  packets have been decoded, say  $l$ , than the subchannels that were used to send the  $l$  packets are now used to send  $l$  new data packets without coding. The purpose of this operation is (1) not to waste precious bandwidth resources, and (2) to wait until all intended  $k$  data packets are recovered. Once all intended  $k$  data packets are recovered, another set of  $k$  data packets can then be encoded into total  $M$  packets using an  $(M, k)$ -MDS code, and the operation starts from the beginning.

We now prove the latency. The line of reasoning follows exactly that in the previous proofs. We derive the cumulative distribution of an event in which  $k$  packets are decoded by time slot  $t$ , compute its difference to get the distribution, and the latency is the expectation of  $T$ . Here, we only derive the cumulative distribution. It is easy to obtain the distribution from it, and then to finish the proof of the latency.

$$\Pr[T \leq t] = 1 - \Pr[T > t]$$

$$= 1 - \sum_{i=0}^{k-1} \binom{M}{i} ((1 - p)^t)^{M-i} (1 - (1 - p)^t)^i.$$

We now prove the achievable rate. It is an elaborate proof that involves a finite Markov chain. As mentioned, we provide a concise version of it. There are total  $k$  states, each labelled from 0 to  $k-1$  according to the number of packets that have been decoded at the destination node so far. For example, state  $s$  indicates that  $s$  packets have been decoded, thus there are at least  $k-s$  packets yet to be decoded to deliver the information contained in the original  $k$  data packets. From state  $s$ , the transition to initial state 0, meaning the original  $k$  data packets being recovered, occurs when at least extra  $k-s$  packets are decoded. The transition to state  $r$  occurs when extra  $r-s$  packets are decoded. Constructing the Markov chain based on the transitions explained, since it is aperiodic and irreducible, one can compute the unique stationary state distribution.

For each state  $s$ ,  $s$  new data packets are sent through the  $s$  subchannels. They can be decoded with probability  $p$ . Also,  $M-s$  packets encoded in the beginning are still sent through the remaining subchannels. The original  $k$  data packets can be recovered when at least  $r-s$  packets of them are decoded. This event occurs with the following probability.

$$1 - \sum_{r=s}^{k-1} \binom{M-s}{r-s} p^{r-s} (1 - p)^{M-r}.$$

Putting all together finishes the proof of the rate.

## V. CONCLUSION

We investigated the trade-offs between latency and data rate in employing coding schemes in the bursty parallel point-to-point channel where packets transmitted in parallel may result in decoding failures due to signal distortion. Compared to a naïve coding scheme that simply sends multiple copies of one original data packet for improved latency, we showed that by employing more sophisticated coding schemes such as MDS codes, which are not too sophisticated to lose their practicality, than repetition, one can gain in latency with less loss in data rate. This result suggests that for mission-critical applications where short delays (or high reliability) are crucial, employing codes in parallel transmission can be helpful.

## ACKNOWLEDGMENT

This work was supported by an Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (No. R0101-16-244, Development of 5G Mobile Communication Technologies for Hyper-connected Smart Services).

## REFERENCES

- [1] H. Weatherspoon and J. D. Kubiatowicz, "Erasure coding vs. replication: A quantitative comparison," in *Proceedings of International Workshop Peer-to-Peer Systems*, 2002.
- [2] N. B. Shah, K. Lee, and K. Ramchandran, "The MDS queue: Analysing the latency performance of erasure codes," *arXiv:1211.5405 [cs.IT]*, 2013.
- [3] M. Sathiamoorthy, A. G. Dimakis, B. Krishnamachari, and F. Bai, "Distributed storage codes reduce latency in vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 2016 – 2027, Sep. 2014.
- [4] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures," *IEEE Transactions on Computers*, vol. 44, no. 2, pp. 192 – 202, Feb. 1995.



**Sunghyun Kim** received his B.S. degree and M.S. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2013 and 2015. He is currently a research associate at Electronics and Telecommunications Research Institute (ETRI). His research interests are in information theory and its applications in various fields, including wireless communication and machine learning.



**Jee-Hyeon Na** is a director in Radio Access Network SW Section at ETRI, where she has been employed since 1989 after receiving her B.S. degree in computer science from Cheonnam National University. She received her M.S. and Ph.D. degrees in computer science from Chungnam National University in 2002 and 2008. She is a member of IEEE and IEICE communications. Her research interests are 4G, 5G Small Cell related radio access networks, MBMS and Self-Organizing Networks.