Ultra-Mini Slot Transmission for 5G+ and 6G URLLC Network

Wonjun Kim and Byonghyo Shim
Institute of New Media and Communications and Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
Email: {wjkim, bshim}@islab.snu.ac.kr

Abstract—Ultra-reliable and low-latency communication (URLLC) is a new service category to accommodate emerging services requiring low end-to-end latency. In the 4G LTE/5G NR, it is very difficult to satisfy the URLLC requirements since multiple OFDM symbols are processed as a bundle. In this paper, we propose a novel low-latency packet transmission scheme, referred to as ultra-mini slot transmission (UMST), suitable for the short packet transmission in URLLC scenario. Key idea of the proposed scheme is to transform the transmit information into subcarrier positions and then decode it using a small amount of time-domain received samples. In particular, in the UMST decoding, we put forth an entirely different approach based on a deep neural network (DNN). From the numerical evaluations on realistic channel models, we demonstrate that the UMST scheme outperforms the conventional transmission schemes in terms of the block error rate (BLER) and transmission latency.

I. INTRODUCTION

Ultra-reliable and low-latency communication (URLLC) is a new service category introduced in 5G to accommodate emerging services requiring low end-to-end latency called mission-critical applications [1], [2]. To support this service, 3GPP sets a stringent requirement that a packet should be delivered with $10^{-5}$ packet error rate within 1 msec end-to-end latency [3]. In many URLLC applications such as autonomous vehicles, remote emergency surgery, and smart factory, an amount of information to be transmitted is tiny. For example, information to be exchanged in robots in the smart factory or autonomous vehicles is in a form of control and command type information such as turn on/off, move forward/backward, speed up/down, and rotate left/right. In this application, required packet size is typically in the order of 100 bits or less.

In the current 4G LTE systems, it is very difficult to satisfy the URLLC requirements since multiple OFDM symbols are processed as a bundle. In fact, a group of 7 symbols spanning 12 subcarriers (0.5 msec × 180 kHz) called a resource block (RB) is used in LTE as a basic scheduling unit (see Fig. 1(a)) [4]. In order to decode RB, a mobile device has to receive 7 OFDM symbols and it takes 0.5 msec just for the buffering samples. Since it takes almost 0.5 msec to perform the control signaling (e.g., PDCCH transmission), it is highly likely that the URLLC latency requirement cannot be satisfied with an ordinary receiver processing in LTE systems [5], [6]. This situation is relaxed in 5G NR due to the short transmission mode referred to as the minislot transmission but it is not enough to support the latency-critical applications since the time for buffering samples is still large to satisfy the stringent latency requirement (e.g., less than 0.1 msec).

In this paper, we propose a ultra low-latency packet transmission scheme suitable for the short packet transmission in URLLC scenarios. The main idea of the proposed scheme, henceforth referred to as ultra-mini slot transmission (UMST),
is to transform the short-sized information into a sparse OFDM symbol vector and then decode it using a small number of received time-domain samples. Specifically, the base station (BS) chooses a small number of subcarriers in the frequency-domain symbol and then encodes the URLLC information into positions of the chosen subcarriers. In doing so, the transmit information is converted into the sparse frequency-domain symbol vector. This together with the fact that the submatrix of inverse discrete Fourier transform (IDFT) matrix serves as a sensing matrix allows us to use the compressed sensing (CS) technique in the decoding of sparse (OFDM) vector. It is now well-known that as long as the sensing mechanism preserves the input information, the input sparse vector can be recovered with a small number of measurement [7]. This means that the mobile device can accurately decode the UMST packet using only small portion of time-domain samples. By choosing early arrived samples, we can achieve a significant reduction of the receiver processing latency (i.e., latency of transmission, buffering, and decoding).

In the decoding process, instead of using the conventional sparse recovery algorithms, we use the deep neural network (DNN) technique, an approach to learn the nonlinear and complicated mapping between the received signal vector and support of transmit sparse vector using the training dataset. In the proposed decoding technique, called deep UMST (D-UMST), sparse recovery problem (more accurately, support identification problem) can be formulated as a multi-label classification problem.

From the numerical evaluations, we demonstrate that the proposed UMST scheme outperforms the conventional transmission schemes by a large margin, achieving 1.5 dB gain in the block error rate (BLER) performance and 52% reduction in the receiver processing latency over the 5G NR transmission scheme.

II. RECEIVER PROCESSING LATENCY IN DOWNLINK TRANSMISSION

In this section, we briefly review the receiver processing latency \( T_{\text{Rx}} \) in the downlink 4G LTE/5G NR systems. At the mobile device, the duration from the beginning of the sample transmission to the end of the decoding process can be expressed as the sum of three distinct latency components (see Fig. 1):

\[
T_{\text{Rx}} = T_{\text{prop}} + T_{\text{buff}} + T_{\text{dec}}. \tag{1}
\]

- \( T_{\text{prop}} \) is the propagation latency, which corresponds to the time for a signal to travel from the BS to the mobile device
- \( T_{\text{buff}} \) is the time to receive the transmitted signal
- \( T_{\text{dec}} \) is the time to decode the transmit information

Among these delay components, we primarily focus on the reduction of the buffering latency \( T_{\text{buff}} \) since \( T_{\text{buff}} \) is much higher than \( T_{\text{prop}} \) and \( T_{\text{dec}} \).\(^1\)

\(^1\) \( T_{\text{prop}} \) is in the order of tens \( \sim \) hundreds of nanoseconds (\( \text{nsec} \)) in the NR cell and \( T_{\text{dec}} \) is a few microseconds (\( \mu\text{sec} \)) using the parallel decoding technique [2].

As mentioned, when delivering a packet in a form of RB in LTE system, a mobile device needs to receive 7 OFDM symbols. In this case, \( T_{\text{buff}} \) equals to one slot period (i.e., 0.5 msec), which is too large to meet the URLLC latency requirement due to the time-consuming control signaling (e.g., 0.5 msec for the PDCCH transmission). In order to reduce \( T_{\text{buff}} \), NR system supports the short transmission mode, called minislot transmission (see Fig. 1(b)). In the minislot transmission, the mobile device needs to buffer the time-domain samples corresponding to 2 \~ 4 symbols to initiate the decoding process and the buffering time \( T_{\text{buff}} \) is 0.15 \~ 0.3 msec. One can expect from this fact that when supporting the latency-critical applications (e.g., less than 0.1 msec for the smart factory), conventional transmission scheme is not a viable option. Therefore, an entirely new transmission scheme to achieve a significant reduction of \( T_{\text{buff}} \) is required.

In the following section, we describe the proposed UMST scheme that enforces \( T_{\text{buff}} \) being less than one symbol period.
III. ULTRA-MINI SLOT TRANSMISSION

A. System Description of UMST

Fig. 2(a) depicts the overall description of the UMST scheme. The key operation of UMST is to transform the short-sized packet into a sparse vector. When encoding the transmit information into the sparse signal vector $s$, a small number of subcarriers (say $k$ out of $N$) are chosen. When the first and fourth subcarriers are picked, for example, then $s = [s_1 \ 0 \ 0 \ s_2 \ \cdots \ 0]^T$ ($s_1$ and $s_2$ are the symbols) and the support (nonzero positions) of $s$ is $\Omega = \{1, 4\}$. Distinctive feature of UMST over the conventional transmission scheme is that positions as well as symbols can be employed to convey the information. When we choose $k$ nonzero elements in $N$ positions ($k \ll N$), we have $\binom{N}{k}$ choices and thus $\lfloor \log_2 \left( \binom{N}{k} \right) \rfloor$ bits information can be encoded into the position of $s$. Suppose the modulation order in the same for all nonzero positions (say $b_s$ bit per symbol), then $kb_s$ bits can be encoded to the active symbols (symbols in the nonzero positions). Therefore, one UMST block conveys $\lfloor \log_2 \left( \binom{N}{k} \right) \rfloor + kb_s$ bits of information in total.

After the UMST encoding process, the inverse fast Fourier transform (IFFT) is applied (i.e., the time-domain sample bits information can be encoded into the position of $s$). Distinctive feature of UMST over the conventional transmission scheme is that positions as well as symbols can be employed to convey the information. When we choose $k$ nonzero elements in $N$ positions ($k \ll N$), we have $\binom{N}{k}$ choices and thus $\lfloor \log_2 \left( \binom{N}{k} \right) \rfloor$ bits information can be encoded into the position of $s$. Suppose the modulation order in the same for all nonzero positions (say $b_s$ bit per symbol), then $kb_s$ bits can be encoded to the active symbols (symbols in the nonzero positions). Therefore, one UMST block conveys $\lfloor \log_2 \left( \binom{N}{k} \right) \rfloor + kb_s$ bits of information in total.

Thus, we have

$$y = \sum_{i=1}^{k} F_i \cdot s_i + w$$

where $H \in \mathbb{C}^{N \times k}$ is the channel matrix and $w$ is the additive Gaussian noise vector. After removing the cyclic prefix, the channel matrix $H$ becomes a circulant matrix and hence it can be eigen-decomposed by the DFT matrix (i.e., $H = F^* \Sigma F$ where $\Sigma$ is the diagonal matrix whose diagonal entry \(\sigma_{ii}\) corresponds to the frequency channel of the $i$-th subcarrier). Thus, we have

$$y = (F^* \Sigma F) F^* s + w$$

$$= F^* \Sigma s + w.$$ 

By letting $x = \Sigma s$, we have

$$y = F^* x + w.$$ 

We note that the supports of $s$ and $x$ are the same (i.e., nonzero positions of $s$ and $x$ are the same).

Based on the theory of CS, as long as the sensing mechanism preserves the energy of an input sparse vector, $k$-sparse vector can be recovered with a small number of measurements $m = ck \log N$ ($c$ is a constant) [7]. In our system model, $x$ and $F^*$ correspond to the sparse vector and the sensing matrix so that $x$ can be recovered from $y$ with a small number of measurements. This means that a small portion of *firstly arrived* samples in $y$ is enough to decode the transmit information (see Fig. 2(b)). The corresponding partial measurement vector $\hat{y} \in \mathbb{C}^{m \times 1} (m \ll N)$ is

$$\hat{y} = P \hat{y}$$

$$= PF^* x + \hat{w}$$

$$= \Phi x + \hat{w}$$

where $P = [I_m \ 0_{m \times (N - m)}]$ is the selection matrix to take the first $m$ samples among $N$ time-domain samples, \(\hat{w} = \hat{w}P\), and $\Phi = PF^*$ is the IDFT submatrix constructed from the first $m$ consecutive rows of $F^*$.

Since the information is conveyed by both subcarrier positions (support) and symbols, two operations (i.e., support identification and symbol detection) are needed for the decoding of the UMST packet. First, to find out the nonzero positions of $s$, we need to identify the support of $x$, which is done by the sparse signal recovery algorithm [7], [8]. We will say more about this in Section III-B. After identifying the support $\Omega$, rest of information can be decoded by detecting the symbol vector $\hat{s}_ii$. Note that, by removing columns of $\Phi$ corresponding to the zero entries in $s$, we can convert the underdetermined system into over-determined one ($m > k$) and thus can find out the solution using the standard technique (e.g., the linear minimum mean square error (LMMSE) estimator followed by the symbol slicer).

The advantages of UMST can be summarized as follows. First and foremost, the decoding process is done with a small number of time-domain samples. When compared to the RB-based and minislot-based transmission, $T_{buff}$ of the UMST scheme can be reduced substantially (more accurately, $T_{buff}$ is less than one symbol period). For example, when half of the transmitted samples is used in decoding (e.g., $m = 128$ and $N = 256$), the UMST achieves 92% and 75% reduction in $T_{buff}$ over the LTE RB (7 symbols) and the NR minislot (2 symbols), respectively. Second, in the support identification, the channel information is unnecessary. This is because the sensing matrix $\Phi$ in (8) is constructed only by the submatrix of IDFT matrix and what we need to do is to find out the nonzero positions of $x = \Sigma s$, not the actual values. Third, the transmit power can be saved considerably. Noting that the required number of samples in the receiver is small ($m \ll N$), the BS does not need to transmit whole samples, resulting in a significant reduction of the transmit power by the factor of $m/N$. For example, if $m = 128$ and $N = 512$, then the transmit power is reduced by 75%.

B. UMST Decoding using Deep Neural Network

In the UMST decoding, we can basically use any sparse recovery algorithm. In many sparse recovery algorithms, such as orthogonal matching pursuit (OMP), an index of a column in $\Phi$ that is maximally correlated to the measurement $\hat{y}$ is chosen as an estimate of the support element [9]. Thus, if two columns of $\Phi$ are highly correlated and only one of these contributes to $\hat{y}$, then it might be difficult to find out the right column from wrong one. One can clearly see from this observation that the support identification performance
depends heavily on the column correlation of $\Phi$. Indeed, when only a few measurements are used, underdetermined ratio $N \ll M$ of the system will increase sharply, causing a severe degradation in the decoding performance.

To handle this issue, we exploit the deep neural network (DNN), a learning-based tool to perform the desired operation [10]. By using the training data as an input and then updating the network parameters using backpropagation mechanism, DNN is trained to represent the nonlinear mapping $Q$ between the input and output. In this context, the received sample vector $\hat{y}$ and the support of $s$ serve as the input and output, respectively. Then, the support identification problem in D-UMST can be formulated as

$$\hat{\Omega} = Q(\hat{y}; \Theta),$$  \hspace{1cm} (9)

where $\Theta$ is the set of weights and biases of DNN.

Fig. 3 depicts the structure of the D-UMST network. The proposed network consists of the fully-connected (FC) layer, batch normalization layer, rectified linear unit (ReLU) layer, and softmax layer with the residual connection. In each training process, we use $D$ training data $\hat{y}^{(1)}, \cdots, \hat{y}^{(D)}$. The output vector $z^{(d)} \in \mathbb{R}^{1 \times 1}$ of the first FC layer is

$$z^{(d)} = W^{in}\hat{y}^{(d)} + b^{in}, \quad \text{for } d = 1, \cdots, D \hspace{1cm} (10)$$

where $W^{in}$ and $b^{in}$ are the initial weight and the initial bias, respectively. After passing the FC layer, $D$ output vectors are stacked in the batch $B = [z^{(1)} \cdots z^{(D)}]^T$. Then, in order to ensure that the batch has a constant mean and variance, the normalization process (often called the batch normalization) is performed [11]. In doing so, variation in $\hat{y}$ is reduced so that the D-UMST can readily obtain the internal features in $\hat{y}$.

After the batch normalization, the output vector $\tilde{z}$ passes through $L$ hidden layers\(^2\). Each hidden layer consists of the FC layer, batch normalization layer, ReLU layer with a residual connection (see Fig. 3). The output vector of the $l$-th FC layer $\tilde{z}^{[l]}$ is given by

$$\tilde{z}^{[l]} = W^{[l]} \left( \tilde{z} + \sum_{i=1}^{l-1} \tilde{z}^{[i]} \right) + b^{[l]}, \hspace{1cm} (11)$$

where $W^{[l]}$ and $b^{[l]}$ are the weight and bias of the $l$-th FC layer, respectively, and $\tilde{z}^{[l]}$ is the output vector of the previous hidden layer. Then, in the same way of the previous batch normalization, $\tilde{z}^{[l]}$ is normalized. After that, an activation function $g$ is applied to $\tilde{z}^{[l]}$ to determine whether each hidden node (unit component of hidden layer) is activated or not. In our network, the ReLU function $f_{\text{ReLU}}(x) = \max(0, x)$ is used as the activation function [12].

After passing through the $L$ hidden layers, the final FC layer produces $N$-dimensional output vector $z^{out}$. It is worth noting that the dimension of $z^{out}$ is matched with that of the sparse vector $s$. In the final FC layer, $z^{out}$ is given by

$$z^{out} = W^{out} \left( z + \sum_{i=1}^{L} \tilde{z}^{[i]} \right) + b^{out}, \hspace{1cm} (12)$$

where $W^{out}$ and $b^{out}$ are the corresponding weight and bias, respectively. Then, the softmax layer generates $N$ probabilities $(\tilde{p}_1, \cdots, \tilde{p}_N)$ representing the likelihood of being the true support element using the softmax function given by

$$\tilde{p}_i = \frac{e^{z^{out}_i}}{\sum_{j=1}^{N} e^{z^{out}_j}}, \quad \text{for } i = 1, \cdots, N. \hspace{1cm} (13)$$

Finally, an estimate of the support $\hat{\Omega}$ is obtained by choosing $k$ elements having the largest probabilities:

$$\hat{\Omega} = \arg \max_{|\Omega|=k} \sum_{i \in \Omega} \tilde{p}_i, \hspace{1cm} (14)$$

Note that the D-UMST decoding is essentially the same as the support identification and all the channel components are contained in $x = \Sigma s$. This means that, in the training phase, the D-UMST decoder only needs to exploit the IDFT submatrix $\Phi$ (known a priori), not the channel statistics. Hence, the proposed D-UMST is robust to the variation of the received vector $\hat{y}$, caused by the channel condition variation.

IV. NUMERICAL RESULTS

In this section, we present the simulation results to evaluate the decoding performance and receiving latency of the proposed UMST. In our simulations, the OFDM systems (with
N = 96 subcarriers) under the fading channels are used. In the D-UMST network, we set L = 6 (the number of hidden layers), α = 500 (the width of hidden layer), D = 1000 (batch size), and η = 10^{-3} (learning rate). When training the D-UMST, we use an Adam optimizer, a well-known optimization tool to maintain the robustness of learning process [13]. As performance measures, BLER and the reception latency in (1) are considered.

In Fig. 4, we evaluate the BLER performance of the proposed UMST scheme as a function of SNR. For comparison, we examine the performance of the conventional physical downlink shared channel (PDSCH) transmission and OMP-based UMST. We observe that the D-UMST scheme outperforms the conventional schemes by a large margin. For example, the D-UMST scheme achieves around 1.5 dB gain over the PDSCH transmission at BLER = 10^{-5}.

Next, we evaluate the latency performance of the UMST and minislot-based PDSCH transmission. In Fig. 5, we plot the distribution of $T_{Rx}$ which corresponds to the time from the initial transmission to the successful packet decoding at the mobile terminal. From the result, we observe that $T_{Rx}$ of UMST is much smaller (52% and 82% on average) than that of minislot transmission and RB transmission, respectively.

V. CONCLUSION

In this paper, we proposed a novel low-latency transmission scheme suitable for the short packet transmission in URLLC scenarios. The key idea behind the proposed UMST scheme is to transform the URLLC information into the sparse symbol vector and then to exploit the DNN architecture in decoding. As long as the number of subcarriers is small enough and the measurements contain enough information to figure out the transmit information, accurate decoding of the UMST-encoded short packet can be guaranteed. We demonstrated from the numerical evaluations that the proposed UMST scheme is very effective in terms of both the reliability and latency.

ACKNOWLEDGMENT

This work was supported by ‘The Cross-Ministry Giga KO-REA Project’ grant funded by the Korea government(MSIT) (No. GK19P0500, Development of Ultra Low-Latency Radio Access Technologies for 5G URLLC Service).

REFERENCES