ReLy: Machine Learning for Ultra-Reliable, Low-Latency Messaging in Industrial Robots

Kunal Sankhe, Dheryta Jaisinghani, and Kaushik Chowdhury

Abstract

Robotic factory floors are transforming the manufacturing sector by delivering an unprecedented boost to productivity. However, such a paradigm raises questions on safety and coordination, especially when in the presence of unexpected events. Time-critical communication messages for such industrial robots mandate the requirement of ultra-reliable low-latency communication (URLLC). Classical WiFi-connected industrial robots often suffer from the traditional dense network problems prevalent in production WiFi networks, where transmission of an emergency notification packet is "best effort," devoid of time guarantees. In this work, we propose a machine-learning-based framework called ReLy that intelligently embeds the time-critical messages in the preamble of outgoing frames at the transmitter. These messages are inferred from the channel state information variations at the receiver. As ReLy is implemented entirely at the physical layer, the transmitter is able to deliver information within 5 ms latency and ultra-high reliability of 99 percent. We experimentally demonstrate the feasibility of achieving URLLC with moving robots in a busy workshop with a number of other peer robots, equipment, desks, and robotic arms, as expected in a typical factory setting.

Introduction

We are living in an exciting era of unprecedented connectivity, where tens of billions of sensors, devices, and machines will require rapid information sharing and actuation. 5G wireless networks specifically call for latency on the order of a few milliseconds (< 10 ms) and ultra-high reliability (> 99 percent) [1]. Many industries are transforming themselves with complete automation in their daily operations, wherein tiny robots perform a plethora of manufacturing, testing, and packaging tasks. As an example, Amazon has deployed over 200,000 robots to streamline warehouse tasks, totaling over ~\$40 million in investment [2].

Problem

A key requirement for these robots to work seamlessly is to have an efficient communication/control plane. However, the focus of existing industries to provide necessary communication for automation is mostly on time-sensitive networking (TSN), which aims to bring industrial-grade robustness and reliability to Ethernet [3]. TSN is mainly designed to provide deterministic messaging on standard Ethernet networks and is part of the IEEE 802.11Q family of standards. Industrial Ethernet systems — such as PROFINET [4] and EtherCAT [5] — are designed for communicating with industrial equipment and capable of real-time control of robotic equipment. Unfortunately, these systems require physical cabling where cables must be attached to machines and robots, increasing the risk of failure and the need for maintenance. Recent advances in 5G New Radio [6] in providing ultra-reliable low-latency (URLLC) services have generated significant interest in extending TSN capabilities over wireless [1, 7]. Unfortunately, establishing 5G infrastructure inside every factory warehouse is economically inviable for cellular operators as well as factory owners. On the contrary, WiFi access points (APs) are easier to deploy and maintain, and are scalable to a large number of devices, thereby making it the predominant choice for indoor environments. Furthermore, production WiFi deployments are commonplace in factory settings. Thus, we believe that any URLLC for robot–robot and robot–infrastructure communication in such cases must occur over WiFi. On the other hand, as robotic deployments become dense, the well-known problems related to contention and congestion that affect WiFi networks [8] cannot be neglected. As these factories of the future will work in the complete absence of human supervision, the WiFi network must ensure the robots have an always available association with the AP. Specifically, in an emergency event, it becomes critically important to meet hard latency and reliability constraints for notifications. With default WiFi standards, meeting both these constraints is a challenging task, but there could also be many legacy devices in the neighborhood that work perfectly with the current state of intermittent WiFi channel access. Thus, any solution that addresses URLLC must also be compatible with pre-existing links used by legacy WiFi devices.

Solution

We propose a novel approach called ReLy that neither requires a pre-established association with an AP nor suffers from low probability of channel access due to node density. For the specific URLLC use case of industrial automation with mobile robots, 5G demands end-to-end latency of < 1 ms and reliability of > 99.9999 percent [7]. ReLy has taken a step forward in this direction to meet the *latency* and *reliability* requirements for

The authors propose a machine-learning-based framework called ReLy that intelligently embeds the time-critical messages in the preamble of outgoing frames at the transmitter. These messages are inferred from the channel state information variations at the receiver. As ReLy is implemented entirely at the physical layer, the transmitter is able to deliver information within 5 ms latency and ultra-high reliability of 99 percent.

Kunal Sankhe and Kaushik Chowdhury are with Northeastern University; Dheryta Jaisinghani is with the University of Northern Iowa.

Digital Object Identifier: 10.1109/MCOM.001.2000598

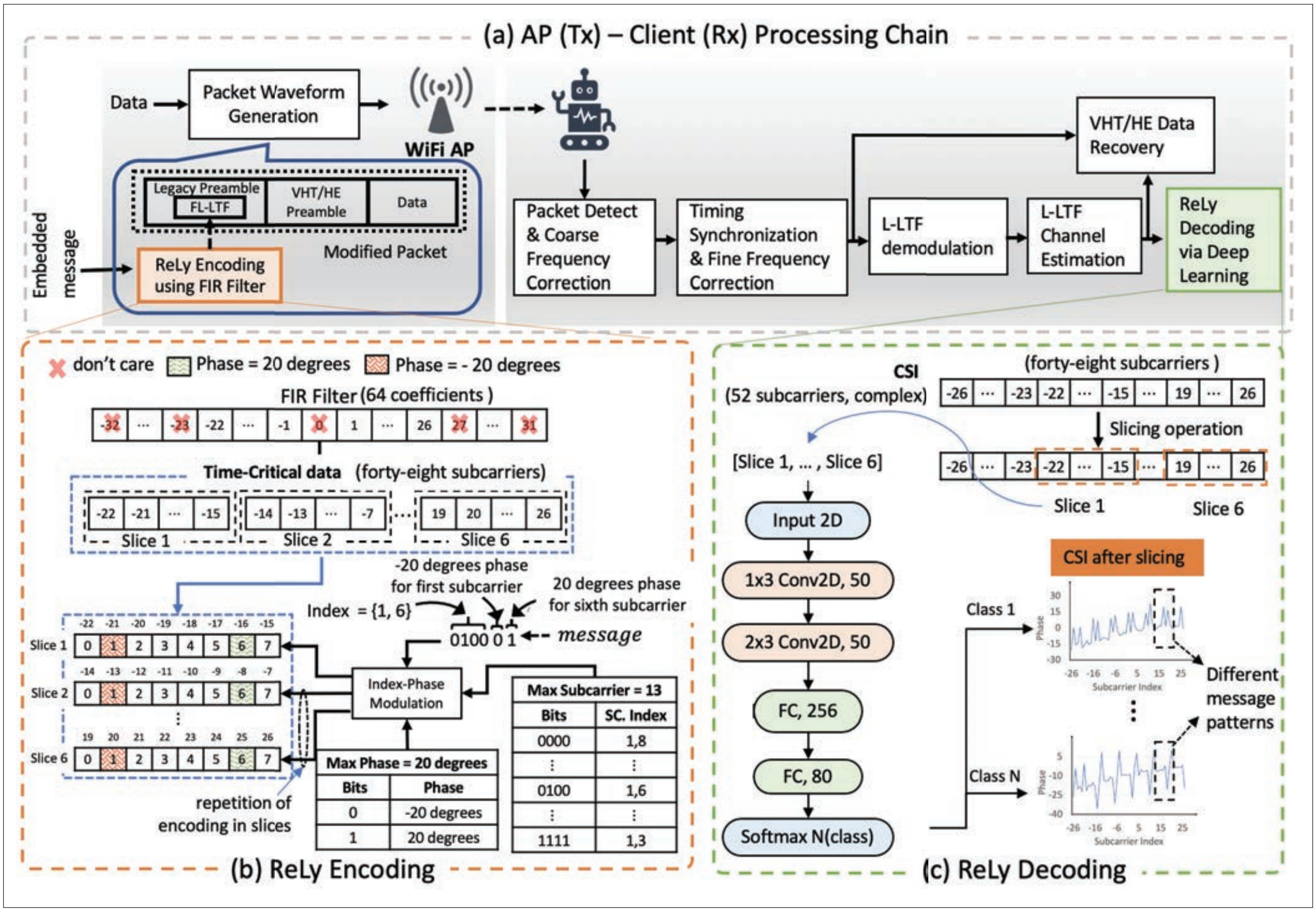

FIGURE 1. Illustrating the operation of ReLy. The AP or the transmitter modifies the preamble with introducing perturbations in phases. As an example, the information bits 010001 are mapped in subcarriers with indexes ranging from –22 to –15. First four bits 0100 indicate locations of the subcarrier, that is, the first and sixth subcarriers within the group of eight subcarriers. Subsequent bit 0 represents the phase shift of –20°, while bit 1 represents phase shift of +20° at the first and sixth subcarriers, respectively. The client or the robot decodes this information with a CNN. Each introduced pattern is mapped to a particular type of message being transmitted.

sending the time-safety-critical messages in emergency industrial communications.

ReLy uses an intelligent encoding scheme that embeds information by subtle distortions in the signal phases within the preamble of an outgoing WiFi frame from the AP. In particular, ReLy exploits the legacy portion in the preamble to embed the information and thus supports its operation in all next-generation (IEEE 802.11ax/ac) as well as legacy (IEEE 802.11a/n) WiFi devices. Any ReLy receiver uses our proposed deep learning framework to decode the embedded information from the changes in channel state information (CSI). The bits encoded in the preamble are mapped to pre-defined emergency messages. Since the logic of data transfer is implemented entirely in the physical layer, it does not require upper-layer involvement or a priori active connection with the AP.

Detailed Overview

The intuition behind ReLy is that the changes in CSI are observed at the client as part of the channel estimation process. CSI aids in capturing an aggregate effect of distortions caused due to fading, shadowing, and Doppler. To measure CSI in an orthogonal frequency-division multiplexing (OFDM)-based WiFi system, a transmitter sends long training symbols (LTFs) that include pre-defined symbols for each OFDM subcarrier (Fig. 1a). After receiving LTFs, the receiver estimates CSI using the received signals and original LTFs through conventional channel estimation methods.The legacy LTF (L-LTF) is pre-pended in outgoing physical layer frames of existing standards IEEE 802.11a/n/ac and even in the latest standard IEEE 802.11ax (marketed as WiFi 6) to support backward compatibility. Leveraging the mandatory channel estimation step and L-LTF for data transmission makes ReLy a standard-independent message delivery method. The AP, being the transmitter in this case, handles the ReLy encoding module. Next, the client handles the ReLy decoding module post channel estimation. By virtue of its principle of operation, carefully injected ReLy encodings can be a part of any outgoing frame from the AP, highlighting a crucial implicit advantage that time-critical message transmission neither needs to wait for its respective frame for transmission nor waste time in unnecessary protocol overheads. Together, these features result in reducing inessential transmission delays. However, an important consideration here is to ensure the bit error rate (BER) of these ongoing frames should not be hampered by these encodings. ReLy handles this case in a sophisticated manner. We explain the transmitter and receiver blocks in the rest of this article.

We summarize the main contributions below:

- We design a novel encoding scheme to deliver fixed length emergency messages (within 5 ms latency bound) by modifying the preamble of WiFi frames.
- We propose a deep learning framework that decodes the emergency messages (with accuracy > 99 percent) at the client by learning CSI of the received signal.
- We show the proof-of-concept testbed on USRP-based AP and clients as mobile robots in a realistic robotic-indoor environment.

Encoding and Decoding Messages for URLLC

Modifying Transmissions through FIR Filters

ReLy modifies the L-LTF in real time using discrete causal finite impulse response (FIR) filters. There are several advantages here:

- Causal filters do not depend on future inputs, but only on past and present ones.
- They are represented as a weighted and finite term sum enabling accurate prediction of the output of the FIR for any given input. As L-LTF symbols are in the time domain, ReLy first converts the filter into its time domain impulse response using the inverse fast Fourier transform (FFT) operation, and then convolves it with L-LTF symbols.

The FIR filter has complex coefficients that only distort the phase of L-LTF symbols at the subcarrier level but not their amplitude. This is required, as amplitude distortion changes the average power of the transmitted signal, which adversely affects the ongoing communication. We explain this encoding procedure in detail below.

Format of Time-Critical Messages: The core innovation in ReLy is to map each of the deliberate, FIR-induced perturbations of the transmitted signals to emergency messages. ReLy generates these perturbations by introducing different levels of phase distortions on a per-subcarrier basis for any chosen group of subcarriers used in the legacy communication channel. Thus, the distinct combination of message bits that is communicated via the L-LTF, indicated by the message on the left (transmitter) side in Fig. 1a, determines how much distortion is introduced and in which subset of subcarriers. The number of distortion patterns (i.e., messages) depend on the number of bits used for encoding in the L-LTF. Specifically, the number of messages grow as exponential base two and number of bits at the exponent. A representative set of 4 messages can be START, PAUSE, TURN-LEFT, and STOP for the mobile robot. Choosing the number of bits is a design choice, and we explain how it is done next.

Waveform Distortion via FIR Filter: We define a FIR filter with 64 complex coefficients, ensuring that ReLy matches the indexing scheme to that of 802.11 OFDM subcarrier mapping as the legacy preamble in 802.11ac/ax is always constructed using 64 subcarriers. The approach ensures backward compatibility in a 20 MHz OFDM channel. Each complex coefficient is defined as exponential of phase in degrees, ensuring its amplitude as one. L-LTF contains time-domain samples that are composed of two OFDM symbols. Each symbol is constructed by mapping a known sequence of positive and negative ones into 52 out of 64 subcarriers in an inverse FFT (IFFT) operation, whereas the remaining 12 are null subcarriers. ReLy intentionally injects phase distortions in this sequence at the subcarrier level by carefully choosing coefficients of the FIR filter. Since any level of phase distortion in symbols positioned at null subcarriers will have no impact in the observed CSI at the client, ReLy treats null subcarriers as don't cares. In its current version, ReLy leverages 48 out of remaining 52 subcarriers for choosing the FIR filter coefficients. To map a higher number of messages, additional four subcarriers can be used. These subcarriers are sequentially arranged into six groups with eight subcarriers per group, called slices. A distortion pattern is embedded within a slice with different phase shifts in various subcarrier locations, such that the maximum number of subcarriers and phase distortions are within the allowed limits.

Keeping the BER of Ongoing Transmission under Control: A major goal of ReLy is to keep the BER of an AP's ongoing transmission under control such that the frame is not corrupted. To achieve that, an optimal limit on the maximum number of subcarriers and the maximum amount of phase changes allowed is to be known. As we highlight later, this is an open research problem, ReLy's current version empirically defines a static value for maximum of number of subcarriers as 12 and phase changes as 20°. These values guarantee no significant change in BER observed at the associated client. We confirm this by measuring BER at a USRP-based receiver placed in factory-similar lab settings. We compute the average of BER over 10,000 captured WiFi packets, where each packet is composed of 8192 bits. Despite variations in signal-to-noise ratio (SNR) and modulation and coding scheme (MCS), we observe that the increase in BER with FIR filtering varies between 0.000747 to 0.000902. This minimal change in BER due to FIR filtering proves the minor impact of preamble modification on WiFi AP-client communication, even in real deployments.

In Fig. 1b, we illustrate the encoding scheme with an example. We choose a total of 12 subcarriers with two subcarriers in each slice, wherein ReLy intentionally injects phase distortion. The allowed amount of phase distortions range from negative to +20°. Next, we map the block of information bits 010001 into subcarrier indexes and phase shifts. The first four bits decide the indexes of two subcarriers. For each selected subcarrier, we choose one bit to select the phase shift to be introduced in L-LTF symbols. Information bits 0100 are mapped to the first and sixthsubcarrier indexes in the first slice. The next subsequent bit 0 is mapped to phase shift of –20° for the symbol of the first subcarrier, whereas the next 1 bit is mapped to phase shift of +20° for the symbol of the sixth subcarrier. The mapping logic stays in a mapping table, as shown in the figure. Correspondingly, the FIR filter coefficients are varied from +20° to –20°. The remaining subcarriers are assigned zero degree phase shifts.

Enhancing Reliability: ReLy ensures reliability through redundancy, that is, by repeating the encoding process in each slice. As shown in Fig.

A major goal of ReLy is to keep the BER of an AP's ongoing transmission under control such that the frame is not corrupted. To achieve that, an optimal limit on the maximum number of subcarriers and the maximum amount of phase changes allowed need to be known.

Visualizing the intentional modifications in the transmitted preambles at the receiver is a challenging task, given that the received CSI is nonlinear and discontinuous, as it varies periodically between positive and negative limits of 180°. A well-known process named "phase-unwrapping" helps in restoring the continuity in received CSI.

1b, we repeat the encoding of bits 010001 in the subsequent slices by selecting phase distortions of –20° in the first subcarrier and +20° in the sixth subcarrier of each slice.

Receiver Design with CNN-Based CSI Decoding

Visible Effect of Preamble Modification at CSI: For decoding the packet, recall that a WiFi client first estimates CSI using filtered L-LTF signal, as shown in Fig. 1c. Since ReLy perturbs only phase in L-LTF symbols, its effect is not visible in CSI amplitude. Therefore, we analyze the visuals of CSI phases only to show the effect of preamble modifications.

Visualizing the intentional modifications in the transmitted preambles at the receiver is a challenging task, given that the received CSI is nonlinear and discontinuous, as it varies periodically between positive and negative limits of 180°. A well-known process named "phase-unwrapping" helps in restoring the continuity in received CSI [9]. Phase unwrapping helps in observing the minor jumps — the "patterns" — in CSI phases at certain subcarriers. Figure 2 highlights the patterns post unwrapped CSI phases. The presence of phase jumps indicate the indexes of the subcarriers where the L-LTF symbol is intentionally distorted. Note that this operation is conducted only to visually demonstrate the distortion effect. ReLy does not specifically need to perform unwrapping.

We first validate the hypothesis that the ReLy encoding module produces a unique pattern that is repeatable under varying channel conditions (channel-invariance) and second, that it creates distinct patterns even in similar channel conditions (uniqueness). To demonstrate this, consider CSI phases collected at two different locations, as shown in Fig. 2, for two different patterns. CSI phase plots 1 and 2, indicated by Pattern 1, Channel 1 and Pattern 1, Channel 2 show nearly similar patterns within CSI phases even when the channel is completely changed (i.e., Location 1 to Location 2). On the other hand, Pattern 2, Channel 1 shows a distinct pattern within CSI phases collected at the same location by selecting different FIR filters. This confirms our hypothesis of ReLy's ability to introduce a unique pattern within CSI phases that are also channel-invariant.

Need for Learning CSI: While we clearly observe the distortion patterns in CSI phases, it is not always possible to get the clear, crisp patterns in mobile robots placed in an industrial environment in the presence of fading and shadowing with a multitude of blockages. Signal processing of CSI phases to decode the pattern is not hard, but has challenges in achieving high reliability of detection. On the other hand, deep learning [10], particularly the convolutional neural network (CNN), has shown remarkable performance in solving the pattern recognition problem in image and speech applications [11] and is steadily gaining traction in applications within the wireless domain [12]. Furthermore, CNNs have proven to be useful in learning CSI for various sensing applications [13]. Varied use cases of CNN used for learning CSI motivated us to assess its applicability in identifying the valid patterns embedded at the AP.

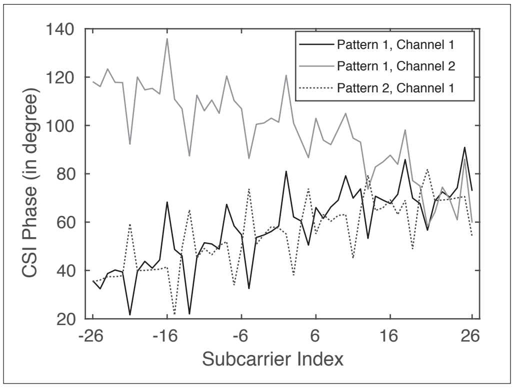

FIGURE 2. CSI phase patterns generated by two distinct FIR filters under the same or different channel conditions. Pattern 1 with Channel 1 and Channel 2 shows the channel-invariance of the patterns, while Patterns 1 and 2 show uniqueness of the pattern with two distinct FIR filters under similar channel conditions as Channel 1.

Input Preprocessing and Output Classes for CNN: The estimated CSI at the client consists of a series of complex values representing channel frequency response for 52 subcarriers. ReLy processes raw CSI using a "slicing operation," wherein it tries to recreate the slices of 48 subcarriers used in the encoding process as shown in Fig. 1c. Each slice is structured as a 2D real-valued tensor of size 2 × 8 and is fed as input to the CNN. Output classes for CNN are the distinct distorted patterns, represented with message bits that were encoded at AP. In this article, we consider four distinct messages bits, Msg1, Msg2, Msg3, and Msg4, that are our output class labels.

CNN Classifier: We use the CNN architecture depicted in Fig. 1c, which consists of four layers, with two 2D convolution layers and two fully connected layers. The input is fed to the first convolutional layer (Conv2D), which consists of 50 filters, each of size 1 × 3.

Similarly, the second Conv2D layer has 50 filters, each of size 2 × 3. Each Conv2D layer is followed by a rectified linear unit (ReLU) activation. Output of the second Conv2D layer is first flattened and then fed to the first fully connected (FC256) layer having 256 neurons. A second fully connected layer (FC80) is added to extract nonlinear combinations of high-level features extracted from previous layers, which are finally passed to a Softmax classifier layer. Through cross-validation, we carefully choose the hyperparameters of CNN including the number of Conv2D filters and their size, as well as depth of the model to ensure its generalization. Our training set consisted of 80,000 training, 10,000 validation, and 10,000 testing examples. Thus, we were able to obtain less biased estimate of the performance of our model. Further, we set the dropout rate to 50 percent at the fully connected layers to overcome overfitting. The model weights are trained using an Adam optimizer with a learning rate of 0.0001.

We predict the phase distortion pattern within CSI by following the methodology of "probability sum." Essentially, Softmax layer generates probability outcome for each pattern for each input slice. We add these probabilities for each pattern across all slices and declare the predicated pattern for which sum of probability quantity is maximum.

CNN Training and Adapting to Unseen Environments: The accuracy of a deep learning model is known to improve with the amount of training data [14]. However, in the wireless domain, the channel has an inherent dynamism involved that makes it hard to port a machine learning model as it is, trained in a certain environment to a new unseen environment. Thus, irrespective of the amount of data used for training in one environment, the model is not expected to perform at its best in another environment. More specifically, there are two related challenges here:

a. Generating enough training data such that a decent training accuracy can be obtained
b. Adapting the learned model to the new and unseen channels such that the threshold for expected prediction accuracy can be met

We address challenge a by artificially creating a simulated dataset with the MATLAB WLAN Toolbox that allows creation of variations in channel for the latest WiFi standards with an instance of `wlanTGacChannel`. With this approach, we created a dataset with 200,000 channel variations in an indoor environment with NLoS signal propagations and SNR ranging from 0 to 30 dB.

Next, we address challenge b by leveraging the machine learning approach of "domain adaptation" wherein a classifier is partially trained with a different but related dataset [15]. Our approach here is to transfer the knowledge learned with simulated channel environments to real and unseen channel environments. For this purpose, we use the popular supervised *fine-tuning* approach, wherein we freeze the first *l* layers of the model that is trained with source data and retrain it with target data to fine-tune its last few layers using backpropagation. For ReLy, we first train a CNN classifier with a large amount of simulated data and then freeze two convolution layers of the classifier and fine-tune other layers by retraining with real data.

Performance Evaluation

Experiment Details

Methodology: Our methodology to evaluate ReLy is largely governed by a robot in an industrial scenario. The floor map of the experiment is shown in Fig. 3. The floor has the live WiFi network of a university with three APs per lab. We consider a robot as the WiFi client that can be either stationary or mobile. The environment in an industrial setting can have regions with and without obstacles, resulting in line-of-sight (LoS) and NLoS signals from the AP. Accordingly, we cover four cases:

a. Client-1 is stationary and in LoS from the AP at a distance of 8 ft.
b. Client-2 is stationary and in NLoS from the AP at a distance of 12ft.
c. Client-3 is mobile in the floor covering both LoS and NLoS regions. To stress test ReLy's performance in extreme cases of blockages with poor reception of WiFi signal, we consider an additional case (Fig. 4).
d) Client-4 is stationary and in NLoS from the AP with many blockages and at a distance of nearly 30 ft.

In all cases, the AP continuously broadcasts the four time-critical messages — Msg1 = "100000," Msg2 = "010001," Msg3 = "001010," and Msg4 = "000111" — that are received by the clients. While transmission of these messages encounter channel (CSI) variations in all four cases, case c has the most variations due to the mobility involved.

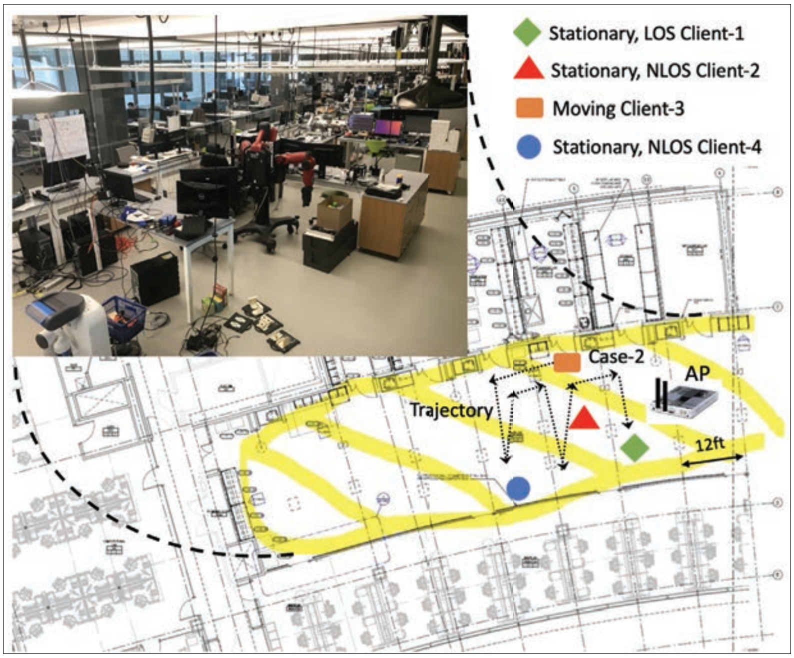

FIGURE 3. Floor map where the experiment was conducted showing the placement of AP, stationary, and mobile clients.

Metric Measurements: We evaluate the efficacy of ReLy for two metrics: accuracy and latency. We report two metrics for accuracy: per-slice accuracy and per-CSI accuracy. We define per-slice accuracy as the fraction of correctly predicted patterns within slices to the actual patterns sent by the transmitter in those slices. Second, we evaluate ReLy's reliability performance by defining per-CSI accuracy as the fraction of correctly predicted patterns sent within the entire CSI (with repetition in slices) to the actual pattern sent by the transmitter. We define latency as how much time it takes for a frame to be processed once it has been received at the receiver. We measure latency with respect to the packet reception pipeline at the receiver (i.e., time for CSI estimation and prediction with CNN). Note that we assume the typical propagation time will be negligible.

Device Configuration: The setup, as shown in Fig. 3, has an AP and four clients: stationary and mobile clients in LoS and NLoS from the AP. We set up the AP with USRP X310 software defined radio (SDR) and the clients with USRP B210 SDRs. The output power of the radio is set to 20 dBm, the same as commercial off-the-shelf (COTS) WiFi APs. The AP transmits 802.11ac-compliant WiFi frames. The mobile client's SDR is kept on an experiment cart that is moved through the floor aisles at human walking speed.

Results

Accuracy: We first report the accuracy of the CNN model when it is trained and tested with the simulated data. In this case, we achieve a per-slice accuracy of 87.43 percent and per-CSI accuracy of 97.77 percent. The improvement in

There is a trade-off in increasing the level of distortions (which improves detection of the encoded message) and the potential degradation in the ongoing communication of the other legacy clients. We need an optimization procedure to determine the number of subcarriers and levels of phase distortion to balance these contrary outcomes.

per-CSI accuracy over per-slice accuracy emphasizes the importance of redundancy achieved with slicing operation. A reason for per-CSI accuracy below 99 percent is that the simulated dataset also includes data for extremely poor channel conditions that we included to improve the robustness of the trained model in realistic environments. We consider this accuracy as our baseline.

Next, we report the accuracy results after employing domain adaptation in the real experiment setup. For case a, where the client is stationary and in LoS of the AP, the per-slice accuracy is 96.45 percent and per-CSI accuracy of 99.81 percent. For case b, where the client is stationary and in NLoS, the per-slice accuracy is 95.22 percent and per-CSI accuracy is 99.22 percent. For case c, where the client is mobile, the per-slice accuracy is 94.55 percent and per-CSI accuracy is 98.58 percent. Lastly, for case d, which covers the worst case performance, the per-slice accuracy and per-CSI accuracy drops to 87 and 95.21 percent, respectively. Thus, ReLy achieves best case accuracy of 99.22 percent for the stationary LoS client and 98.58 percent for the mobile client. Accuracy drops for distant clients due to severe drop in SNR values; this problem needs further research for improving the accuracy.

For fine-tuning the trained CNN model, we use nearly 20,000 training examples, as compared to 200,000, and thus need only 10 percent of the total training data used during simulation. This results in reduction of training time by 64 ms on average.

Latency: For default WiFi operation, the frame is first processed completely at the physical layer for packet detection, channel estimation, equalization, and decoding; then it is processed by the medium access control (MAC) layer, and later, upper layers. However, for ReLy, the processing ends immediately post channel estimation, and the message can be handled later by upper layers. In ReLy, the processing latency is incurred in estimating CSI and invoking a deep learning engine to identify a valid pattern. Our evaluation reports a processing latency of 4 ms for ReLy as compared to the latency of 12.8 ms while processing the entire receiver chain. Our setup of USRP devices is not an optimized one, thus introducing avoidable processing delays. An implementation with COTS devices would provide an optimized hardware and software configuration that would prevent unnecessary processing delays. Given that CSI can now be decoded with COTS Android smartphones, we are working on developing an edge device capable of receiving messages over ReLy.

Open Research Challenges

Need for Security

While an AP is able to convey messages to clients, this stage is susceptible to replay attacks. Methods like RF fingerprinting, which detects a specific emitter using IQ samples, can be used to distinguish the authorized AP from rogue APs. However, this approach needs further validation as the fingerprint itself can be eclipsed by intentional phase distortions at the transmitter side.

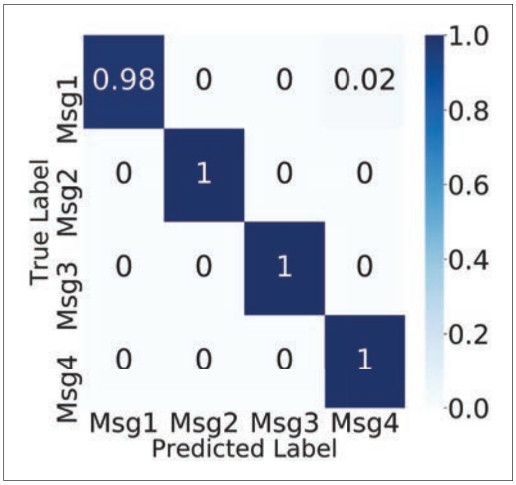

FIGURE 4. Confusion matrices for stationary, NLoS client, Client-3 placed in a live environment at a distance of 12 ft from AP. Prior to domain-adaptation, CNN achieves an overall per-CSI accuracy of 97.77 percent with simulation data. Post domain adaptation, it achieves per-CSI accuracy of 99.22 percent in a live environment.

Optimal Selection of Upper Bounds on Number of Subcarriers and Level of Phase Distortion

In highly mobile scenarios, nonlinearity in the observed CSI phase can lower the ability to discriminate small changes in the CSI due to intentional distortions. This makes learning difficult. There is a trade-off in increasing the level of distortions (which improves detection of the encoded message) and the potential degradation in the ongoing communication of the other legacy clients. We need an optimization procedure to determine the number of subcarriers and levels of phase distortion to balance these contrary outcomes.

Distributed Learning of CSI at Clients

In ReLy, we train the CNN in a centralized fashion, wherein CSIs are collected at the different clients and trained separately in a central entity. This approach raises privacy concerns in the sharing of data used to train a CNN architecture. Federated learning can be a promising solution here, wherein the CNN model is trained at the clients without sharing the raw data and then disseminated.

Testing in a Real Factory Environment

While we evaluated ReLy in our robotic lab under a sufficiently complex channel environment, it is crucial to test ReLy in a real factory environment that may have different channel propagations. It will expose us to challenges that we have not perceived yet and motivate us to find solutions that are necessary to integrate into ReLy for its deployment in the real world.

Conclusion

We present ReLy, which reliably sends time-critical messages with very low latency for industrial robots. The transmitter intelligently embeds the information in the L-LTF field of the preamble. The receiver uses CNN and decodes the emergency notifications from the changes in the CSI. We demonstrated the feasibility of ReLy with live experiments with mobile robots on a busy workshop floor. ReLy is able to transmit messages within 5 ms with > 99 percent accuracy, thus validating its proposed use in industrial URLLC communications as defined by the existing 5G specifications in 3GPP Release 15.

Acknowledgments

This work is supported by the Office of Naval Research under grant N000141612651 and NSF award CNS 1923789.

References

[1] Y. Kim *et al.*, "New Radio (NR) and Its Evolution Toward 5G-Advanced," *IEEE Wireless Commun.*, vol. 26, no. 3, June 2019, pp. 2–7.

[2] "Amazon Now Has 200,000 Robots Working in Its Warehouses," Jan. 2020; https://roboticsandautomationnews.com/2020/01/21/ amazon-now-has-200000-robots-working-in-its-warehouses/28840/, accessed 21 June 2020.

[3] J. L. Messenger, "Time-Sensitive Networking: An Introduction," *IEEE Commun. Standards Mag.*, vol. 2, no. 2, 2018, pp. 29–33.

[4] R. Pigan and M. Metter, *Automating with PROFINET: Industrial Communication Based on Industrial Ethernet*, Wiley, 2008.

[5] D. Jansen and H. Buttner, "Real-Time Ethernet: The EtherCAT Solution," *Computing and Control Engineering*, vol. 15, no. 1, 2004, pp. 16–21.

[6] 3GPP, "Study on Scenarios and Requirements for Next Generation Access Technologies." TSG RAN TR38.913 R14, June 2017.

[7] G. Brown, "Ultra-Reliable Low-Latency 5G for Industrial Automation," Qualcomm tech. rep., vol. 2, 2018, p. 5206–5394.

[8] L. Ho and H. Gacanin, "Design Principles for Ultra-Dense Wi-Fi Deployments," *Proc. IEEE Wireless Commun. and Networking Conf.*, 2018, pp. 1–6.

[9] J. Tribolet, "A New Phase Unwrapping Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 25, no. 2, 1977, pp. 170–77.

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[11] W. Liu *et al.*, "A Survey of Deep Neural Network Architectures and Their Applications," *Neurocomputing*, vol. 234, 2017, pp. 11–26.

[12] S. Riyaz *et al.*, "Deep Learning Convolutional Neural Networks for Radio Identification," *IEEE Commun. Mag.*, vol. 56, no. 9, Sept. 2018, pp. 146–52.

[13] Y. Ma, G. Zhou, and S. Wang, "WiFi Sensing with Channel State Information: A Survey," *ACM Computing Surveys*, vol. 52, no. 3, 2019, pp. 1–36.

[14] X. Zhu *et al.*, "Do We Need More Training Data?" *Int'l. J. Computer Vision*, vol. 119, no. 1, 2016, pp. 76–92.

[15] M. Wang and W. Deng, "Deep Visual Domain Adaptation: A Survey," *Neurocomputing*, vol. 312, 2018, pp. 135–53.

Biographies

Kunal Sankhe is currently pursuing a Ph.D. degree in computer engineering at Northeastern University under the supervision of Prof. K. Chowdhury. His current research efforts are focused on implementing deep learning in the wireless domain and developing a cross-layer communication framework for the Internet of Things.

Dheryta Jaisinghani is an assistant professor at the University of Northern Iowa, Cedar Falls. Prior to that she was a postdoctoral research associate at Northeastern University, Boston, Massachusetts. She received her Ph.D. from IIIT-Delhi, India, in 2019 and her Master's degree from IIIT-Bangalore, India, in 2012. Her research interests are in the areas of pervasive and ubiquitous computing with particular focus on networked systems, large-scale WiFi networks, software-defined-networks, and mobile computing.

Kaushik Roy Chowdhury [M'09, SM'15] is a professor at Northeastern University. His current research interests involve systems aspects of networked robotics, machine learning for agile spectrum sensing/access, wireless energy transfer, and large-scale experimental deployment of emerging wireless technologies.

While we evaluated ReLy in our robotic lab under sufficiently complex channel environment, it is crucial to test ReLy in a real factory environment that may have different channel propagations. It will expose us to challenges that we haven't perceive yet and motivate us to find solutions that are necessary to integrate into ReLy for its deployment in the real-world.