2024 23rd ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)

Demo Abstract: An Interpretable and Trainable CTC Framework

Haoyu Wang†
haoy_w@njfu.edu.cn
Nanjing Forestry University
Nanjing, China

Jiazhao Wang†
jiazhao_wang@mymail.sutd.edu.sg
Singapore University of Technology and Design
Singapore

Xin Lv
lx10144336@163.com
Nanjing Forestry University
Nanjing, China

Demin Gao
dmgao@njfu.edu.cn
Nanjing Forestry University
Nanjing, China

Wenchao Jiang*
wenchao_jiang@sutd.edu.sg
Singapore University of Technology and Design
Singapore

ABSTRACT

Cross-technology communication (CTC) enables seamless interactions between diverse wireless technologies. Most existing work is based on reversing the transmission path to identify the appropriate payload to generate the waveform that the target devices can recognize. However, this method suffers from many limitations, including dependency on specific technologies and the necessity for intricate algorithms to mitigate distortion. To address these challenges, we present NNCTC, a Neural-Network-based Cross-Technology Communication framework which can achieve reliable and interpretable Cross-Technology Communication through a training process with an example of WiFi (OFDM and CCK) to both known and unknown modulation schemes.

KEYWORDS

Cross-Technology Communication, Neural Network

1 INTRODUCTION

The Internet of Things (IoT) has rapidly developed, and different wireless technologies are designed for IoT connections. The huge number of IoT devices has lead to ubiquitous interference, bringing opportunities for collaboration between different wireless communication technologies, represented by cross-technology communication (CTC) [3], CTC achieves direct communication between different wireless communication technologies in the physical layer (PHY). However, existing CTC schemes encounter two fundamental challenges that Technology-specific emulation and Hand-crafted parameters.

In this work, inspired by the recent advances in applying AI in physical layer communication[1, 2, 4], we propose Neural-Network-based Cross-Technology Communication (NNCTC), a general framework to construct CTC with the help of neural networks. We reinvestigate classic CTC strategies to demonstrate how to fit them into such a framework and bring their scalability, flexibility, and learning ability to the next level. Without loss of generality, we take the physical layer CTC from WiFi to ZigBee as an example to introduce in detail how NNCTC improves physical layer CTC with an end-to-end neural network. For a better demonstration of NNCTC, we consider the OFDM-based Wi-Fi schemes, those applying IEEE 802.11a/g/n/ac, which are more complex than CCK-based IEEE 802.11b. It's worth pointing out that NNCTC does not have to be limited to certain WiFi transmission schemes, as we design the NNCTC based on a generalized and recognized idea to emulate the time-domain waveforms.

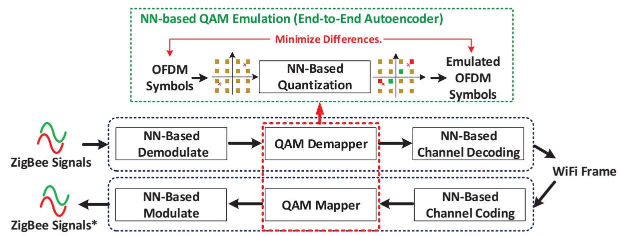

Figure 1: Workflow of NNCTC

2 SYSTEM DESIGN

Fig. 1 depicts our emulation process based on Neural Network (NN). We have devised a bidirectional emulation, commencing with standard ZigBee signals, and "ZigBee Signals*" representing the output of the end-to-end model. Training of the encoder is accomplished by minimizing the gap between "ZigBee Signals*" and "ZigBee Signals," ultimately yielding the desired WiFi Frame.

2.1 The Transformation of the WiFi Physical Layer into Neural Networks

Following the procedures in Fig. 1, we convert the processing blocks into neural networks, including DFT/IDFT, QAM mapper, and QAM demapper.

Convert DFT and IDFT processes to neural networks. DFT/IDFT processes share similar formulas as

$$
\begin{aligned}
DFT: \quad X[n] &= \sum_{i=1}^{N-1} S[n] e^{-\frac{j2\pi ni}{N}}, \quad 0 \le n \le N-1 \\
IDFT: \quad S[n] &= \sum_{i=1}^{N-1} X[n] e^{\frac{j2\pi ni}{N}}, \quad 0 \le n \le N-1
\end{aligned} \tag{1}
$$

†Co-primary student authors. * Corresponding author.

979-8-3503-6201-5/24/$31.00 ©2024 IEEE
DOI 10.1109/IPSN61024.2024.00027

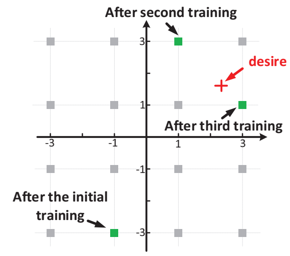

Figure 2: Schematic diagram of NN-Based Quantization training in NNCTC

where $X[n]$ and $S[n]$ represent the frequency-domain components and the time-domain signals, respectively. $e^{-\frac{j2\pi ni}{N}}$ and $e^{\frac{j2\pi ni}{N}}$ are the corresponding basis functions of DFT/IDFT.

Convert QAM Demapper/Mapper to neural network. QAM mapper maps bits to complex symbols. To achieve differentiable operations, we managed to build a floating-point one-hot vector that supports gradient transfer, and then operate it with a linear layer with standard constellation point weight parameters to achieve differentiable Mapper or DeMapper operations.

Convert quantization process to neural network. The quantization process is crucial because it restricts the results of the DFT to standard constellation points to ensure that we can derive the corresponding data bits. We use basic operations in PyTorch to implement the corresponding QAM demodulation step by step. The difference is that we embed neural network layers within the standard steps to add learning capabilities. For example, inserting a network layer to learn the scaling factor in the "quantization" step.

2.2 QAM Emulation via Neural Network

As shown in Fig. 1, we can stack these models to form an autoencoder-like emulation model. Because the goal of the autoencoder is to reconstruct the input at the output, we can employ the model to conduct different kinds of emulation processes by configuring the different inputs. Quantification is the most important step, which determines what kind of WiFi signal we generate to simulate the required ZigBee signal.

NN-Based Analog Emulation. We aim to generate a time domain signal that approximates the desired signal. Assume that $u(t)$ is the desired time domain signal and $v(t)$ is the simulated signal generated by the autoencoder used for QAM emulation. The corresponding discrete forms of $u(t)$ and $v(t)$ signals are u[n] and $v[n]$ respectively, where n is the sampling point. According to Parseval's theorem, the total energy of the signal in the time domain is equal to the total energy of the signal in the frequency domain. We have:

$$\sum_{n=0}^{N-1} |u[n] - v[n]|^2 = \frac{1}{\mathrm{N}} \sum_{k=0}^{N-1} |U[k] - V[k]|^2. \quad (2)$$

$$\mathrm{Loss}\,(\mathrm{u}[\mathrm{n}], \mathrm{v}[\mathrm{n}]) = \frac{1}{N^2} \sum_{k=0}^{N-1} |U[k] - V[k]|^2. \quad (3)$$

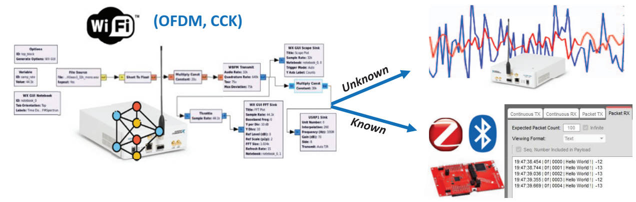

Figure 3: Demo set-up

Therefore, combining the MSELoss function and Equ. 2, we can get the relationship between the loss value and the frequency domain signal, that is, Equ. 3. According to Equation. 3, we know that gradient descent is performed based on the loss value in the neural network, which ultimately reduces the absolute value of the difference between the desired frequency domain component $U[n]$ and the simulated frequency domain component $V[n]$. So the nearest constellation point is selected, as shown in Fig. 2.

2.3 Demonstration

In Fig. 3, we show our setup to demonstrate the learning and emulation capabilities of NNCTC. On the transmitter side, we use a USRP to simulate a base device of WiFi. The USRP uses GNURadio to implement pipelines for the two most popular modulation schemes as in previous CTC works: OFDM (used in 802.11a/g/n/ac) and CCK (used in 802.11b). But unlike the earlier CTC works, parameters in the GNURadio pipelines have to be determined manually through math derivative, NNCTC supports another mode to learn these parameters through a training process. In the demo, we will show such a training process can deal with both known signals like ZigBee and unknown signals like a general QAM signal or PSK signals. It will demonstrate NNCTC is a versatile AI framework that can create an interpretable CTC pipeline without expert knowledge.

ACKNOWLEDGMENTS

This research/project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20221-0017). This research/project is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme. This work was supported by The Future Network Scientific Research Fund Project (Grant No. FNSRFP-2021-YB-17).

REFERENCES

[1] Hengtao He, Shi Jin, Chao-Kai Wen, Feifei Gao, Geoffrey Ye Li, and Zongben Xu. 2019. Model-driven deep learning for physical layer communications. *IEEE Wireless Communications* 26, 5 (2019), 77–83.

[2] Jakob Hoydis, Sebastian Cammerer, Fayçal Ait Aoudia, Avinash Vem, Nikolaus Binder, Guillermo Marcus, and Alexander Keller. 2022. Sionna: An open-source library for next-generation physical layer research. *arXiv preprint arXiv:2203.11854* (2022).

[3] Zhijun Li and Tian He. 2017. Webee: Physical-layer cross-technology communication via emulation. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 2–14.

[4] Timothy O'shea and Jakob Hoydis. 2017. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking* 3, 4 (2017), 563–575.