Towards A Machine Learning-Based Framework For Automated Design of Networking Protocols

Hannaneh Barahouei Pasandi
Dept. of computer science
Virginia Commonwealth University
Richmond,VA, USA
barahoueipash@vcu.edu

I. Introduction

Today, due to the increasing demands on wireless communications and new emerged technologies, new protocols will be designed faster than before. By evolving network technologies as well as increasing demands of modern applications, general-purpose protocol stacks are not always adequate and need to be replaced by application tailored protocols. To cope with the emergence of various device characteristics and application requirements, complex and custom design of high performance networking protocols is needed. The current methods for protocol design are mainly human-based and thus are burdened with various limitations. Design of new protocols is time-consuming and requires a specialized knowledge that is not trivial to acquire. This is especially limiting in the context of modern networking domain, i.e., IEEE 802.11 protocol that is continuously evolving nowadays to meet new requirements and conditions through the addition of new amendments. Furthermore, once a protocol is designed, it lacks optimal adaptability and flexibility to changes in the environment, since contemporary communication scenarios display dynamic and non-stationary properties. Changes in network are so fast and frequent that no human-based mechanisms can follow them accurately. Finally, current approaches are limited to human perception and understanding of this field, thus limiting the potential for extracting new and unexpected insights during the protocol design process. *Therefore, replacing this inefficient human-based protocol designing process by a novel paradigm that enables rapid design of efficient, flexible, and high performance protocols that intelligently adapt to different device characteristics, application requirements, user objectives, and network conditions is highly desired.*

Sharing limited resources is the main concern in wireless networks. Different medium access control (MAC) protocols have been proposed in the literature to allocate resources of the system between the endpoints. One of the main roles of MAC protocols is to decide when a node should access the shared link to transmit a packet. Given the various network and environment conditions, designing a flexible and efficient MAC protocol that fits with all possible different conditions is a challenge especially when these conditions are not available or known ahead of time. Reinforcement learning (RL) is an interactive optimization tool that doesn't require any prior knowledge of the system/environment. Inspired by recent advances in reinforcement learning and deep learning, *in my thesis work, we aim at building an autonomous, scalable, and robust intelligent system using Deep Q-Learning (DQL) that is capable of real-time design of adaptive and self-managing protocols only by interacting with and learning from the environment, without having any prior knowledge.*

Probably the closet work to our framework is [3]. Authors propose and evaluate Remy tool that generates congestion control algorithms to run at the endpoints rather than manually formulate each endpoints reaction to congestion signals. Remy is a heuristic search algorithm, which maintains rule tables in which states are mapped to actions. Although our idea of automating the protocol design using ML is similar, the novelty of our approach resides at the way our framework constructs a protocol from a set of building blocks. Moreover, in our approach we benefit from model-free online learning scheme by considering the network as a black box with no prior assumptions about the network, Since it is hard to accurately model the network due to variation uncertainty. In particular, we optimize the protocols by choosing the best set of core functionalities (e.g., CW, Carrier Sensing) than just tuning specific parameters (e.g., adjusting congestion window in case of congestion control).

II. Research Path

The objective of my research is to design a novel framework that automates the design of MAC protocols using machine-learning. Fig 1 represents an overview of such a framework. In this framework, we decompose IEEE 802.11 MAC protocols into core functionalities called *building blocks* (e.g., carrier sensing, backoff, timers, etc,.). The RL agent learns the optimum set of blocks that should be included in the protocol design. In particular, the input of RL agent is the combination of binary representation of the blocks and a measurement signal (e.g,. throughput). After designing the protocol using the selected set of blocks, a reward (e.g., average throughput, number of the collisions) is calculated for the designed protocol, and sent back to the agent as the feedback signal. The goal of the agent is to maximize/minimize the reward in each state. A preliminary version of this system has been implemented in [1]. In the following, we outline the main implementation parts of the proposed framework and highlight the corresponding challenges and proposed solution.

978-1-5386-9151-9/19/$31.00

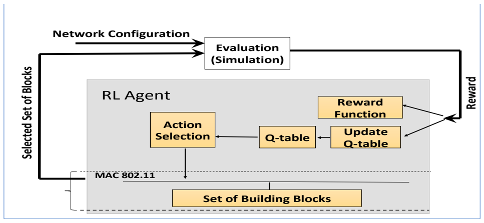

Fig. 1: Proposed Framework

Develop the basic modules and components of IEEE 802.11 standards for automatic network protocol design. In our framework, we consider the modular design principle that decouples a MAC protocol into independent blocks.This principle allows fast reconfiguration and adaptation of the MAC design. In the literature for MAC decomposition two approaches are used: FSM and data flow [2]. However, the main question is what is the best level of granularity of the decomposed building blocks and how to evaluate different set of building blocks with the same functionality but different level of granularity when designing the protocol. To address these challenges we propose using similarity metrics in order to check if a change in building blocks leads to a modification of the content and increasing the diversity of information embedded in them.

Design and develop a Deep Q-Learning (DQL) approach for centralized and distributed agents. After finalizing the building block set, DQL agent uses these blocks as well as a batch of previous experiences as input. The output is a vector of corresponding Q-values for all the possible actions from the current state. We expand this architecture for centralized and distributed approach. In centralized single-agent scenario, a single DQL agent is responsible for managing the protocol design task. In other words, a centralized agent designs a protocol and the nodes in the network use the same designed protocol. There is no universal ML model to solve all tasks, which is especially vivid in the context of deep learning, where a number of free parameters must be selected. One of the vital challenges in deep learning lies in the possible strong temporal correlations embedded in the agent learning procedure. Additionally, in our framework we faced long-range time dependencies that will materialize only after many transitions of the environment. Therefore, we alleviated this by embedding Long Short-Term Memory (LSTM) networks within our DQL architecture. In Multi-agent scenario, we will model actual protocol design task, where multiple protocols must be created at the same time. Multi-agent reinforcement learning (MARL) is an emerging paradigm in RL. We assumed a cooperative scenario, where all agents work together towards a common goal. We propose to develop a mechanism for communication between agents, using bidirectional and all-to-all channels. Bidirectional channels will allow two agents working on similar protocols to exchange their information, speeding-up the learning process and improving the chances of fulfilling the time constraints. We finally evaluate our framework in terms of DQL agent convergence and adaptation to network conditions, as well as comparing the designed protocol by our agent against series of state of the art MAC protocols. Our results in our preliminary work [1] demonstrate that the agent is able to converge in a reasonable amount of time and perform comparably similar or better in some scenarios to conventional protocols.

III. Preliminary Work

We have evaluated a simplified version of our framework hereafter *AlphaMAC* for automating the design of MAC protocols. We wanted to see if AlphaMAC is able to design a simple ALOHA, and also to see if AlphaMAC is able to pick other components that can help in maximizing its objective (average link throughput). In the first step, We decoupled ALOHA into a set of coarse-grained building blocks using a data flow model. We also added other blocks from other contention-based protocols to see whether the agent find them useful in the protocol design. The decomposed set of building blocks included: ACK, Backoff, Contention Window (CW), Carrier Sensing, Access Resolution (the probability whether the agent transmit in a particular time-slot or not), and Transmission Rate. In this framework, the input of the agent is the binary representation of building blocks and the throughput of the link the previous iteration as a network measurement signal. The objective of the agent is to select the best set of blocks to maximize average throughput of the link. For evaluation, we developed an event-driven simulator using C++. The designed protocols by ML agent were evaluated in different scenarios varying in terms of number of nodes, traffic load, and level of present noise. We compared the performance of designed protocols by AlphaMAC against Pure ALOHA. Our result showed AlphaMAC achieves higher throughput than ALOHA in all the scenarios. For example, in scenarios where the network load is low, AlphaMAC removes ACK and Backoff components to gain a higher throughput, while in scenarios with the high traffic load and noise, it not only includes ACK and Backoff but also CW component to enhance the number of successful transmissions.

In this research, we motivated the importance of a shift from human-driven protocol design process to a machine-based design. We proposed and evaluated a novel, self-managing and self-adaptive framework for automating MAC protocol design.

IV. Acknowledgement

I would like to thank my advisor Dr. Tamer Nadeem for his support and valuable feedback.

References

[1] H. barahouei Pasandi and T. Nadeem. Challenges and limitations in automating the design of mac protocols using machine-learning. In *Artificial Intelligence in Information and Communications (ICAIIC), 2019 IEEE*. IEEE, 2019.

[2] P. Wang, M. Petrova, and P. Mähönen. Dmdl: A hierarchical approach to design, visualize, and implement mac protocols. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*, pages 1–6. IEEE, 2018.

[3] K. Winstein and H. Balakrishnan. Tcp ex machina: computer-generated congestion control. In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 123–134. ACM, 2013.