# Millimeter Wave Wireless Network on Chip Using Deep Reinforcement Learning

Suraj Jog[†], Zikun Liu[†], Antonio Franques[†], Vimuth Fernando[†],
Haitham Hassanieh[†], Sergi Abadal[★], Josep Torrellas[†]

University of Illinois at Urbana-Champaign[†], Polytechnic University of Catalonia[★]

## ABSTRACT

Wireless Network-on-Chip (NoC) has emerged as a promising solution to scale chip multi-core processors to hundreds of cores. However, traditional medium access protocols fall short here since the traffic patterns on wireless NoCs tend to be very dynamic and can change drastically across different cores, different time intervals and different applications. In this work, we present *NeuMAC*, a unified approach that combines networking, architecture and AI to generate highly adaptive medium access protocols that can learn and optimize for the structure, correlations and statistics of the traffic patterns on the NoC. Our results show that *NeuMAC* can quickly adapt to NoC traffic to provide significant gains in terms of latency and overall execution time, improving the execution time by up to 1.69× - 3.74×.

## CCS CONCEPTS

• **Networks** → **Network protocols**; **Wireless access networks**.

## KEYWORDS

Millimeter Wave, Deep Reinforcement Learning, Wireless Network-On-Chip

## 1 INTRODUCTION

Network-on-Chip (NoC) architectures have played a fundamental role in scaling the number of processing cores on a single chip which led to unprecedented parallelism and speedups in execution time [4]. However, as the number of cores continues to increase, the gains saturate and hit a problem known as the "Coherency Wall" [5], where the speedup gained by parallelism and multithreading is outweighed by the wired network's communication cost for keeping the caches coherent.

To address the above problem, computer architects have proposed the use of millimeter wave (mmWave) wireless

links for communication between the cores of an NoC multi-processor [4]. Recent advances in mmWave technology have also led to the design and implementation of NoC mmWave transceivers and antennas that can deliver multi-Gbps links and impose modest overhead (0.4–5.6%) on the area and power consumption of a chip multiprocessor [8]. This ability to augment NoCs with mmWave wireless links benefits chip multiprocessors in two important aspects:

- **Latency:** Unlike wired NoC, wireless enables every core to reach every other core in just 1-hop without the need to go through multi-hop NoC routers which can take multiple execution cycles. This allows the architecture to scale to a large number of cores while maintaining the ability to deliver packets within the tight timing requirements of execution on the cores [3].

- **Broadcast:** Since wireless is a broadcast medium, transmitted packets are directly heard at all other cores which significantly simplifies the cache coherency protocol. In particular, any local changes in the cache of a core can instantaneously be replicated at all other cores through a single packet transmission [3]. In contrast, today's wired NoCs must send multiple parallel unicast transmissions to synchronize the caches, which leads to large overhead in the cache coherency protocol that scales poorly as the number of cores increases.

However, while the use of wireless can significantly benefit NoCs, it brings on new challenges in terms of medium access. In particular, traffic patterns in NoCs tend to change drastically across applications and even during the execution of a single application [1, 3], making it very difficult to design efficient medium access protocols.

Fig. 1 shows examples of traffic traces for three common benchmark applications on a 16-core multiprocessor. For clarity, we only show a portion of the execution spanning ten thousand cycles. Some applications, like *PageRank* (Fig. 1(a)), have almost constant traffic on all cores and can benefit from a contention-free protocol like TDMA. Other applications, like computing the *Shortest Path in a Graph* (Fig. 1(b)), have bursty traffic and can benefit from a contention based protocol like CSMA. Moreover, in most applications, the traffic pattern changes drastically within a single execution of the application, as seen in *BodyTrack* (Fig. 1(c,d)). Past work on wireless NoC use *contention-free* schemes like TDMA and CDMA [2, 7], or *contention-based* schemes like CSMA [6]. However, such static protocols are unable to adapt to highly dynamic traffic patterns.
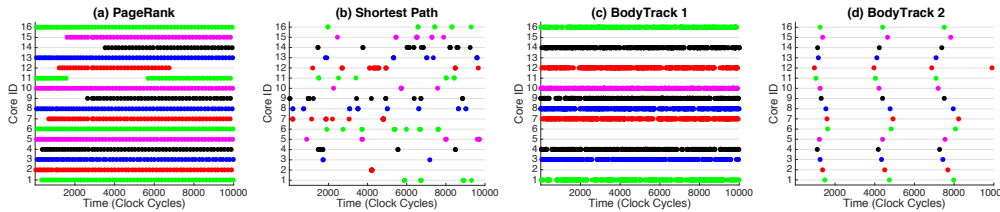
**Figure 1: Traffic Pattern on a 16-core multiprocessor for different applications. The X-axis shows clock cycles, and the Y-axis corresponds to each of the 16 cores. The figures depict the scatter plots representing the packet injections into the buffer of each core.**
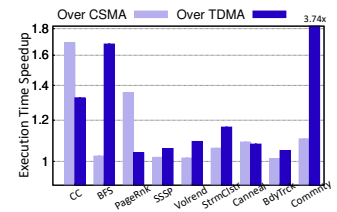
**Figure 2: Speedup in application execution time over Baselines (y axis in logscale)**
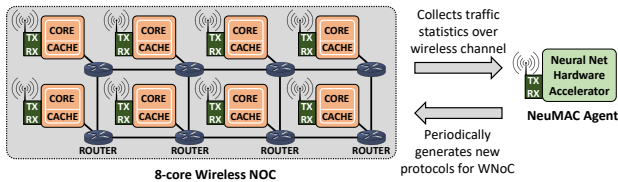


**Figure 3: NeuMAC Overview**

We introduce *NeuMAC* as a first step towards designing highly adaptive medium access protocols for wireless NoCs. *NeuMAC* leverages a reinforcement learning framework with deep neural networks to generate new MAC protocols that can learn traffic patterns, dynamically adapt the protocol to handle different applications running on the multicore, and implicitly account for the intricate dependencies between execution on the cores and packet delivery times.

RL enables *NeuMAC* to make better decisions by learning from experience. Many basic functions, like FFT, graph search, sorting, shortest path, etc., tend to repeatedly appear in many applications. Past work also shows that different programs produce unique periodic traffic pattern and as the number of cores increases, the traffic patterns show increasing spatiotemporal correlations [1]. *NeuMAC* learns these statistics and correlations of the traffic patterns, to be able to both predict future traffic patterns based on traffic history and adapt its protocol to best suit the predicted future traffic.

## 2 NEUMAC OVERVIEW

*NeuMAC* consists of two components as shown in Fig. 3. (1) A standard NoC multicore processor with $N$ cores where each core has been augmented with a wireless transceiver. (2) A *NeuMAC* agent that periodically generates new medium access policies based on the traffic patterns it sees on the wireless NoC. The *NeuMAC* agent is equipped with a wireless transceiver through which it listens on the channel, and collects traffic data about core transmissions, collisions and idle slots. It, then, feeds this data to a trained RL neural network that implicitly predicts the future traffic patterns and generates a new policy to be used as the medium access protocol on the wireless NoC. This process is repeated periodically so that *NeuMAC* can adapt the protocol with time varying traffic.

To allow *NeuMAC* to adapt to dynamic workloads, we need to design a policy that can span a wide range of protocols, all the way from contention-free protocols like TDMA to contention-based protocols like CSMA. Towards this end, we adopt a two-layer protocol design. The first layer consists of a deterministic underlying TDMA schedule, where each core is assigned a unique time slot for transmission. The second layer consists of a probabilistic transmission schedule like CSMA, where each core $i$ is assigned a contention probability $p_i$. Specifically, during its assigned time slot, core $i$ transmits on the channel with probability 1. During other cores' assigned time slots, core $i$ can transmit with probability $p_i$. In the event of a collision, exponential backoff is implemented similar to CSMA. Such a design allows for flexibility since $p_i = 0, \forall i$ emulates pure TDMA, whereas $p_i > 0$ mimics a CSMA-like protocol with varying degrees of aggressiveness on the channel. The design also gives the flexibility to control each core individually, so that the *NeuMAC* can potentially increase contention probabilities for cores that observe high traffic density.

*NeuMAC*'s agent is trained using deep reinforcement learning, where the high level objective is to minimize total execution time of the application. We set our reward signal to be -1 for each time step, thus penalizing every additional time step that the application executes. The state space is defined as the traffic statistics summary collected by the *NeuMAC* agent on the wireless channel, and the action space constitutes the contention probabilities of the cores as described above. We train our policy network end-to-end in an episodic setting using the popular REINFORCE algorithm with baseline subtraction.

## 3 IMPLEMENTATION AND RESULTS

We evaluate *NeuMAC* for processors with core count $n = 64$ on 9 different applications using Multi2sim, which is a cycle-level execution-driven architectural simulator. We use k-fold cross validation, thus ensuring that the *NeuMAC* agent is never explicitly trained on the application it is being evaluated on, and our results show that *NeuMAC* can generalize well to different applications. From Fig 2, *NeuMAC* can speedup the total execution time by up to 1.69× and 3.74× over CSMA and TDMA respectively. Further, *NeuMAC* improves median packet latency due to queuing by 4.11× compared to CSMA, and by 9.18× compared to TDMA. These results demonstrate that *NeuMAC* is able to learn concepts that allow it to adapt to different workloads and translate into gains in total execution time and packet latency for applications.

# REFERENCES

[1] S. Abadal, A. Mestres, R. Martínez, E. Alarcon, and A. Cabellos-Aparicio. Multicast on-chip traffic analysis targeting manycore noc design. In *PDP*, 2015.

[2] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo. Wireless noc as interconnection backbone for multicore chips: Promises and challenges. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2012.

[3] V. Fernando, A. Franques, S. Abadal, S. Misailovic, and J. Torrellas. Replica: wireless manycore for communication-intensive and approx data. *ASPLOS*, 2019.

[4] A. Karkar, T. Mak, K.-F. Tong, and A. Yakovlev. A survey of emerging interconnects for on-chip efficient multicast and broadcast in many-cores.

[5] R. Kumar, T. Mattson, G. Pokam, and R. V. D. Wijngaart. The case for message passing on many-core chips. In *Multiprocessor System-on-Chip*. Springer, 2011.

[6] A. Mestres, S. Abadal, J. Torrellas, E. Alarcón, and A. Aparicio. Mac protocol for reliable broadcast communications in wireless network-on-chip. *NoCArc*, 2016.

[7] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly, and A. Kwasinski. Cdma enabled wireless network-on-chip. *ACM JETC*, 2014.

[8] X. Yu, H. Rashtian, S. Mirabbasi, P. Pande, and D. Heo. An 18.7-gb/s 60-ghz ook demodulator in 65-nm cmos for wireless network-on-chip. *IEEE TCAS-I*, 2015.

*IEEE CIRC SYST MAG*, 2016.