Impact of user density increase on 802.11ax based Network Optimization

SIETZE VAN DER VINNE, University of Twente, The Netherlands

The number of networked devices is increasing rapidly with numbers reaching 29.3 billion by 2023, up from 18.4 billion in 2018. This increasing user density requires new technologies to provide reliable connectivity. 5G and IEEE 802.11ax are among these technologies. Deployment of Artificial Intelligence (AI) in wireless networks has been proven useful in optimizing dense networks, however optimization frameworks proposed in literature lack analysis of user density variation over optimization time and performance improvements. In this paper, Contention Window (CW) optimization in 802.11ax networks has been analysed because it has been shown to improve network performance significantly. The analysis shows that throughput varies from 33.9 Mbits/s with 5 users in the network to 32.7 Mbit/s with 50 users in the network while delay and jitter variations remain within 45 and 14 milliseconds, respectively. The analysis reveals CW optimization can help improve performance in dense networks and this analysis can help network administrators to deploy cost-optimized networks.

Additional Key Words and Phrases: Deep Reinforcement Learning, IEEE 802.11ax, Throughput, Wireless LAN, Contention Window, DRL based Optimizations, User density in WLANs

1 INTRODUCTION

As can be seen from figure 1, the number of connected devices grows by billions every year [5]. One of the biggest growing categories is Machine-to-Machine (also called IoT) devices. It is forecasted that 50 per cent of total connected devices will be M2M devices by the end of 2023, of which around half is wireless. That is why

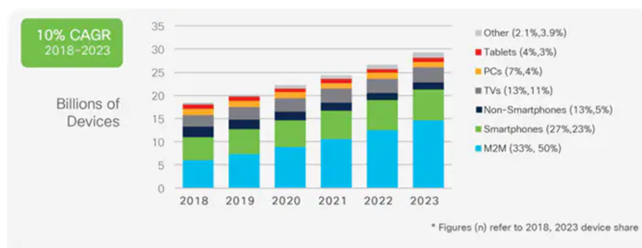

Fig. 1. Global device and connection growth. Source: [5]

researchers are trying to find ways for Access Points (AP) and edge devices to work more and more efficient. Internet standard IEEE 802.11ax is a result of such research and released in 2020. This new standard is more efficient than its predecessors, especially when it comes to higher user density. It employs OFDMA to support high user density however, it still follows the Contention-based channel access mechanism.

When user density grows channel access becomes very complex and induces a lot of collisions. To get access to the channel the station checks if the channel is idle and then transmits its data frame. The station waits for an acknowledgement to see if there was no collision before it proceeds with the next frame. If there was no acknowledgement there was most likely a collision and it waits a random number of time slots between 0 and the Contention Window (CW) to avoid another collision. When the channel does not get an acknowledgement it doubles the CW and this continues until the station gets an acknowledgement and it will then send the rest of the frames. This way of choosing CW can be very inefficient in networks with a high user density and setting the right CW is

TScIT 37, July 8, 2022, Enschede, The Netherlands

very important to the performance of a WiFi network [3]. That is why researchers are trying to find techniques to optimize the CW value.

Optimization of CW can be efficiently done with the help of Machine learning-based algorithms like DRL [11] or supervised learning (SL) [8] and throughput improvements have been achieved compared to conventional optimization techniques. Now with the increase of devices, it is important to understand all the consequences of user density increase in such optimized networks like latency or the convergence speed of the DRL. This can help network administrators decide on the employment of CW optimization solutions and design more robust and efficient networks while reducing infrastructure costs.

The objective of this research is to find out how user increase affects CW optimization time and overall network performance. This analysis will help design WiFi networks keeping in view anticipated user density. The analysis can also be used to design cost-effective networks with minimal infrastructure to support a given user density. In order to perform this analysis, network simulations were carried out in NS3 with the CW optimization framework used in [11].

2 RELATED WORK

A lot of research has been done on network optimization with ML. The articles in the introduction [11](using ML and neural networks) and [8](using ML without neural networks) are examples of CW optimization. But there are other parameters that can be optimized using ML. Like frame length optimization using Supervised Learning [7], they were able to achieve an improvement of 18.36% in throughput by optimizing the length of the packet frames using Supervised Learning. Applying RL to optimize the data transmission rate [4] is another example, here the data sending rates of nodes is controlled with a DRL agent and higher throughput was achieved. Or this research [1], where DRL is used to solve time and resource allocation problems in OFDMA wireless networks. Higher throughput was achieved here as well. ML research in other wireless networks like Long-Term Evolution (LTE) has also been done [9], this research also achieved higher throughput.

Some of these papers have an analysis of how varying user density affects throughput, but other effects are left out. This paper will show how CW optimization in 802.11ax networks ...

- affects jitter with varying user density.
- affects delay with varying user density.
- affects throughput with varying user density.
- can best be deployed.
- can reduce network costs.

3 METHODOLOGY

To study the effects of increasing user density on CW optimization networks simulations have been used. IOT sensors and M2M devices are represented as nodes with varying traffic requirements. Traffic generation is done from nodes towards AP and from AP towards nodes. Parameters like bandwidth, packet size and CW are set up as per IEEE 802.11ax standard. Network statistics like throughput, Delays, Jitter and CW values are monitored and collected for the DRL algorithm to perform CW optimization.

3.1 Ns-3 and Ns3-gym

For the simulations, a combination of Ns-3 and ns3-gym is used. Ns-3 is an open-source network simulator developed in C++ using object orienting programming model and is widely used in network research. Ns-3 can simulate the latest 802.11ax network models and allows sophisticated tracing and monitoring of the network. Ns3-gym uses the OpenAI Gym[2] RL toolkit. Given numerical data of observations, actions and rewards OpenAI Gym can train a RL agent. Ns3-gym is a framework that uses this toolkit specifically for RL research in networking.

In this paper, an open-source framework [11] has been used which already has a DRL algorithm for CW optimization.

3.2 Data

In these simulations, user density has been increased and the effects on DRL and certain aspects of the network have been analyzed. All simulations have been run twice, once with CW optimization and once with standard 802.11ax. Later the results are analyzed and compared for improvement. So in this quantitative research, the following data on networks with varying number of stations from 5,10,....,N has been collected:

- Optimization time (the time it takes for the DRL algorithm to learn the optimal CW value).
- Throughput of the network (successfully transferred data in Mbits/second).
- Latency (Time it takes for a packet to be received).
- Jitter (delay variation in seconds).

The gathered data will be presented in graphs. From these graphs, conclusions will be drawn and a real-life environment will be sketched to see if network costs can be reduced.

3.3 Network and study model

The simulated network consists of one AP and varying number of stations from 5,10,....,N. These stations will constantly send UDP packets with a fixed size of 1500B to the AP. The CW-value is calculated with the DRL algorithm and then broadcasted to all stations in the neighbourhood. After receiving these broadcasts all stations update their CW accordingly. The network topology is shown in figure 2 and while network parameters used in simulations are given in Table 1. Around 20% of the connected stations are stationary, and the rest of the stations are mobile. All stations are connected wirelessly.

3.4 DRL Framework

The DRL framework used in [11] has been used in this study. In RL there is an agent who can perform certain actions in an environment. In this framework, the RL agent is centralized at the Access Point (AP). The environment is the wireless network and the action the agent can take is to broadcast a certain CW.

The goal of the agent is to optimize its parameters, in this case, the CW. The agent learns how to reach this goal by means of rewards and punishments. Throughput is a good measure of the performance of

# AP	1
# Stations	5 up to N
IEEE	802.11ax
Modulation	1024-QAM
Channel width	20 MHz
Packet protocol	UDP
Packet size	1500B

Table 1. Network settings

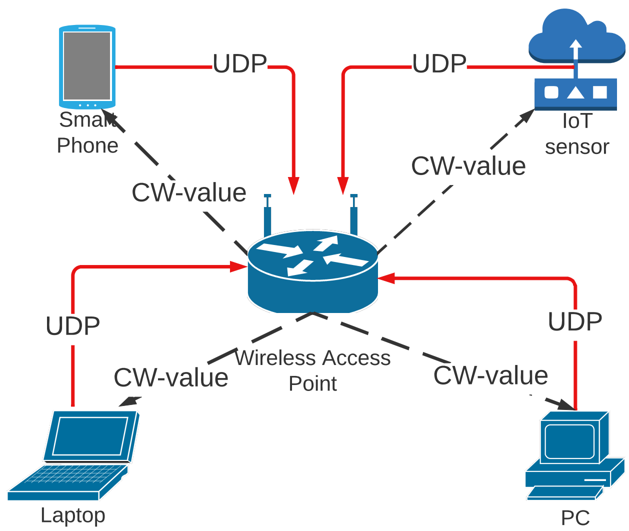

Fig. 2. Network topology: stations sent UDP packets, AP broadcasts CW.

the network and is therefore used as a reward factor. The algorithm works in steps, in every step, a CW is chosen and the throughput is measured. If the throughput has increased or decreased since the last step the agent is rewarded or punished respectively. The framework's algorithm uses Neural Networks to learn from its experience and determine the CW, which makes this a DRL algorithm. The framework has two options for the algorithms: discrete (DQN) and continuous (DDPG). In this paper, only the DDPG method is used, since the research from this paper[11] shows that this method has higher throughput than the discrete method. The settings of the DRL algorithm are shown in table 2 [11]. DDPG has two neural networks, one for the actor who gives an action directly from a state and one for the critic who takes the state and action as input and outputs the expected reward. Both the critic and the actor network had the following layers structure: 8 × 128 × 64. The hyper-parameters in table 2 and the network configuration are all important for the DRL

Parameter	Value
Interaction period	10 ms
History length h	300
DQN's learning rate	4×10^{-4}
DDPG's actor learning rate	4×10^{-4}
DDPG's critic learning rate	4×10^{-3}
Batch size	32
Reward discount γ	0.7
Replay buffer B size	18,000
Soft update coefficient τ	4×10^{-3}

Table 2. DRL settings

algorithm to work optimally. They have been chosen like this by means of trial and error after many simulations.

To help the agent explore more possible actions a certain randomization factor is added to the decision of the agent. This so-called noise makes sure the agent does not get stuck on the first decision it gets rewarded on. The noise decreases every step. When the noise is zero the agent will always make the best-known decision.

4 RESULTS

Jitter (variation in time delay), Throughput (received bits by the AP per second) and Delay (the time it takes for a packet to reach its destination) are measured because it gives a good representation of the performance of the network. DRL optimization time has been measured to give an insight on how long it would take before the network can perform optimally. In the simulations all devices stay connected and are constantly sending data, this is different from real-world networks. This means that the results may also be different from real-world networks. A few extra connected devices that send little to no data would however have comparable effects on standard 802.11ax as to an optimized network. This is because they would rarely request channel access.

4.1 Throughput

As you can see figure 3 shows the network throughput for three different situations. Throughput (no optimization), which is the average throughput of the standard 802.11ax. Then Throughput (total), which is the average throughput during and after training. And Throughput (after training) which is the average throughput after the training is done. At 50 stations the throughput is 23% higher in the network with optimization after training.

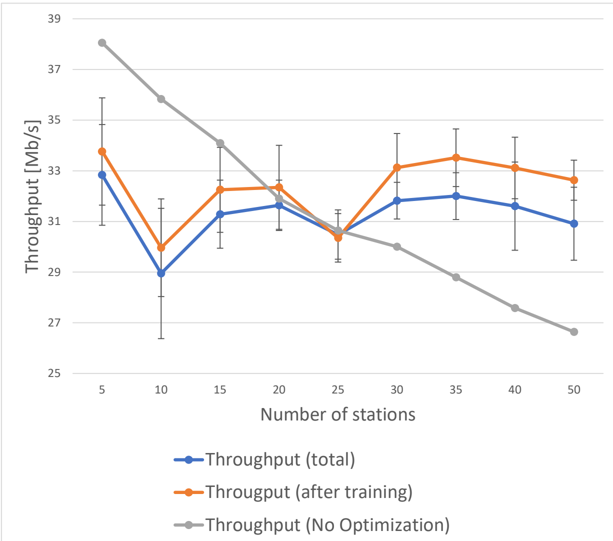

Fig. 3. Throughput in Mb/s

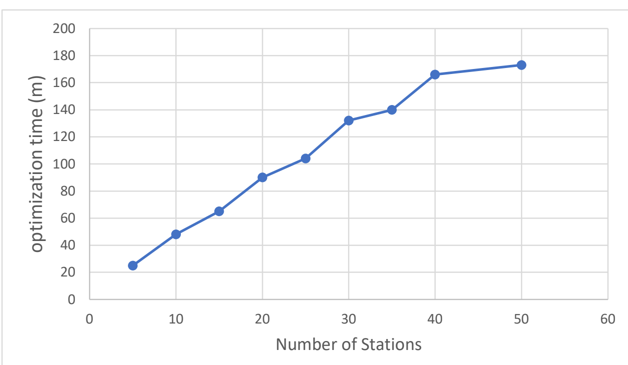

Fig. 4. optimization time in minutes

4.2 Optimization Time

Figure 4 shows how long it takes for the DRL agent to optimize the CW in minutes. The training was done with a i7-9750H CPU at 2.60 GHz. The training time increases from 25 minutes for 5 stations to 173 minutes for 50 stations. These optimization times are high, but once an agent is trained it does not need to be retrained until the network topology changes a lot. This is because the neural network is already optimal for that network topology.

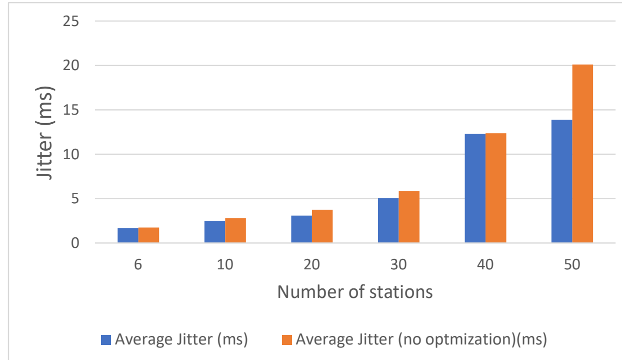

Fig. 5. Jitter in milliseconds

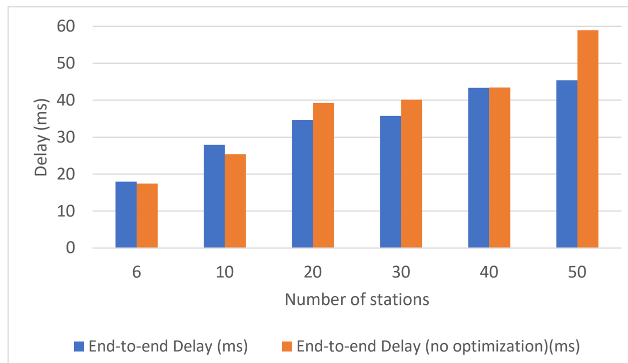

Fig. 6. Delay in milliseconds

4.3 Jitter and Delay

And figure 5 shows the average jitter in he network in milliseconds. The jitter is approximately the same for 6 up to 40 stations. At 50 stations the optimized network has 45% less jitter.

Lastly, figure 6 shows the average of all end-to-end delays of the received packets. As with jitter the delay is approximately the same for 6 up to 40 stations. At 50 stations the delay is 30% less for the optimized network.

5 ANALYSIS AND DISCUSSION

In this section, an example will be given of how a network administrator can reduce the cost of a network using CW optimization. An example of a network with a lot of wirelessly connected sensors and devices is the network of a hospital.

5.1 throughput

Table 3 shows some data rates of typical devices in such a network according to [10]. Now as an example suppose a hospital floor uses 5 EMGs, 10 digital audio stethoscopes and 10 ECGs for their patients. This would need only 4.35 Mb/s of throughput. If 5 physicians are

Digital device	Data rate
Digital audio stethoscope (heart sound)	∼120kbps
Electromyogram EMG	∼600kbps
Electrocardiogram ECG	∼15kbps
Medical video for teleconsultation (e.g.,ophthalmoscope, proctoscope, etc.)	∼1.544Mb/s
Voice/video/chat communication of commuting physicians	384kbps to 1.544Mb/s
Digital radiography (DICOM)	6MB (image size)
Mammogram (DICOM)	24MB (image size)

Table 3. Typical medical data rates

having a teleconsultation and 5 communicating over video, this would need 15.44 Mb/s of throughput. Now suppose one of those physicians wants to download a digital radiography (6 MB = 48 Mbits) at the same time within 4 seconds, this would need another 12 Mb/s of throughput. In total 31.79 Mb/s of throughput is needed for 30 devices. From 3 we can conclude that this would need either 2 APs without CW optimization or one AP with CW optimization.

If instead of two physicians having a teleconsultation there would be 22 extra ECGs, then the required throughput would be 30.601 Mb/s. This would be even less throughput, but because the user density now has increased to 50 stations an AP without CW optimization would have even more problems handling the data.

Given that the AP has the same bandwidth (maximum rate of data transfer) as the AP from the simulation. This is of course a very specific example but it does show how, if the CW optimization was free, the cost of a network can be halved.

5.2 Jitter and delay

In this article [6], requirements for future healthcare applications are given. Examples are "remote pervasive monitoring" and "Mobile-health wearables" which both allow for a maximum jitter of 25 ms. The average jitter of an AP without CW optimization with 50 connected stations is 20.1 ms, but in some connections, jitter did exceed 25 ms. For the optimized network, this was not the case and the average jitter was 45% less. For 40 stations or less CW optimization can not make a difference, but when the user density grows and too much jitter is becoming a problem, CW optimization can be considered as a solution.

The same goes for delay (30% improvement at 50 stations) which could be of importance when it comes to telesurgery [12].

5.3 Cost reduction

From the example, it can be concluded that the number of APs can be halved in some situations. Assume a network administrator has to provide the wireless network of a hospital with three floors. He could either choose two standard 802.11ax routers per floor which are around €60 or one with a bit more computing power to better handle the DRL training which would be around €80 at the time of writing. For three floors this would give a cost reduction of €120 which is 33%

6 CONCLUSION

The IEEE 802.11ax network standard was designed to cope with high-density networks. From this research, it may be concluded that this network standard can be improved when it comes to throughput, delay and jitter. This can be done with the help of Contention Window optimization. Because of this improvement, the cost of high-density networks can be reduced by 33%.

REFERENCES

[1] Ravikumar Balakrishnan, Kunal Sankhe, V. Srinivasa Somayazulu, Rath Vannithamby, and Jerry Sydir. 2019. Deep Reinforcement Learning Based Traffic- and Channel-Aware OFDMA Resource Allocation. In *2019 IEEE Global Communications Conference (GLOBECOM)*. 1–6. https://doi.org/10.1109/GLOBECOM38437.2019.9014270

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *CoRR* abs/1606.01540 (2016). arXiv:1606.01540 http://arxiv.org/abs/1606.01540

[3] Yunli Chen and Dharma Agrawal. 2004. Effect of Contention Window on the performance of IEEE 802.11 WLANs. (01 2004).

[4] Soohyun Cho. 2020. Rate Adaptation with Q-Learning in CSMA/CA Wireless Networks. *Journal of Information Processing Systems* 16 (10 2020), 1048–1063. https://doi.org/10.3745/JIPS.03.0148

[5] Cisco. 2018, updated 2020. *Cisco Annual Internet Report (2018–2023) White Paper*. Technical Report. Cisco.

[6] Giulia Cisotto, Edoardo Casarin, and Stefano Tomasin. 2020. Requirements and Enablers of Advanced Healthcare Services over Future Cellular Systems. *IEEE Communications Magazine* 58, 3 (mar 2020), 76–81. https://doi.org/10.1109/mcom.001.1900349

[7] Estefanía Coronado, Abin Thomas, and Roberto Riggio. 2020. Adaptive ML-Based Frame Length Optimisation in Enterprise SD-WLANs. *J. Netw. Syst. Manage.* 28, 4 (oct 2020), 850–881. https://doi.org/10.1007/s10922-020-09527-y

[8] Yalda Edalat and Katia Obraczka. 2019. Dynamically Tuning IEEE 802.11's Contention Window Using Machine Learning. In *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (Miami Beach, FL, USA) *(MSWIM '19)*. Association for Computing Machinery, New York, NY, USA, 19–26. https://doi.org/10.1145/3345768.3355920

[9] Ahmed M. El-Shal, Badiaa Gabr, Laila H. Afify, Amr El-Sherif, Karim G. Seddik, and Mustafa Elattar. 2021. Machine Learning-based Module for Monitoring LTE/WiFi Coexistence Networks Dynamics. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. 1–6. https://doi.org/10.1109/ICCWorkshops50388.2021.9473865

[10] Dimitris Komnakos, Demosthenes Vouyioukas, Ilias Maglogiannis, and Philip Constantinou. 2008. Performance evaluation of an enhanced uplink 3.5G system for mobile healthcare applications. *Int. J. Telemed. Appl.* 2008 (Dec. 2008), 417870.

[11] Witold Wydmański and Szymon Szott. 2021. Contention Window Optimization in IEEE 802.11ax Networks with Deep Reinforcement Learning. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–6. https://doi.org/10.1109/WCNC49053.2021.9417575

[12] Song Xu, Manuela Perez, Kun Yang, Cyril Perrenot, Jacques Felblinger, and Jacques Hubert. 2014. Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer(®) simulator. *Surg. Endosc.* 28, 9 (Sept. 2014), 2569–2576.