Protocol Violations in the WiFi Distributed Coordination Function of Raspberry Pi 4

1st Bjørn Ivar Teigen
Department of Informatics
University of Oslo
Oslo, Norway
bjornite@ifi.uio.no

2nd Neil Davies
Predictable Network Solutions Limited
Stonehouse, United Kingdom
neil.davies@pnsol.com

3rd Kai Olav Ellefsen
Department of Informatics
University of Oslo
Oslo, Norway
kaiolae@ifi.uio.no

4th Tor Skeie
Department of Informatics
University of Oslo
Oslo, Norway
tskeie@ifi.uio.no

5th Jim Torresen
Department of Informatics
University of Oslo
Oslo, Norway
jimtoer@ifi.uio.no

***Abstract*—Low latency wireless communications is a growing research area with wide-ranging applications in many areas of technology. Modeling latency in the unlicensed wireless spectrum is challenging because many devices, different protocols, and both persistent and on-demand networks coexist using the same radio resources. Useful modeling depends on a detailed understanding of how each protocol works in order to capture and quantify interaction effects. While validating a model of WiFi latency, we discovered protocol violations in the WiFi implementation on the Raspberry Pi 4. In this work, we document how the WiFi protocol on Raspberry Pi 4B does not conform to the 802.11 WiFi standard. Our results have implications for researchers using the Raspberry Pi 4 to measure WiFi performance. These findings may be symptomatic of a blind spot in the WiFi certification process. If similar protocol violations are common in WiFi devices, this may be important to understand the performance of WiFi networks and their interactions with other wireless protocols.**

I. Introduction

In September 2020, the Broadband Forum published a new standard for measuring network quality [1]. The standard is based on the concept of *Quality attenuation*, which is a measure of the latency and packet loss performance of packet-switched networks. The quality attenuation concept has been developed through several decades of academic work [2], [3], and it is most comprehensively treated by Thompson and Davies [4]. We use "ΔQ" to abbreviate quality attenuation.

This recent interest and standardization related to ΔQ motivated us to investigate the ΔQ of WiFi networks. Models exist of the WiFi protocol, but most previous work on modeling the 802.11 protocol has focused on analyzing throughput and system utilization [5], [6]. Throughput analysis can be used to calculate the *average* latency of a WiFi link [7], but the ΔQ methodology requires the entire distribution of latency.

To address the lack of a WiFi protocol model with detailed predictions of latency distributions, we have developed a novel WiFi MAC layer protocol model. Our model is based on the mathematics of ΔQ. During the validation phase of the model development, we discovered a mismatch between the model predictions and our testbed measurements. On closer inspection, we were surprised to find that a large part of the prediction error was due to protocol violations in the WiFi implementation on the Raspberry Pi. This paper describes how the WiFi implementation on Raspberry Pi 4 differs from the IEEE 802.11 standard in ways that increases the chance of collisions between transmitting stations and increase latency and jitter in certain cases. We do not describe the details of the ΔQ WiFi model in this paper because of the short form factor. Model details have been published as a preprint [8], and a version of that paper will be published elsewhere.

II. Related work

A model for translating protocol specifications into latency distributions is useful for detecting the kind of protocol violations we report on without looking in detail at low-level behavior. In this section, we describe some of the most relevant prior work on WiFi protocol modeling and argue why protocol violations in the Raspberry Pi are especially relevant to the scientific community.

Bianchi [5] models the WiFi distributed coordination function (DCF) using a Markov chain, and in doing so, makes a few key simplifications. From our perspective, the most important simplification is the abstraction from time to time steps. A time-step in Bianchis model is defined by the value of the back-off counters. This simplification comes at the cost of discarding timing details because time-steps may be of different magnitude. Bianchi also points this out [5, Section IV, A].

Tinnirello et al. [6] extend the methodology of Bianchi [5]. Here, the authors solve a Markov chain for the steady-state *distribution of back-off timer values*. This approach is closer to modeling latency distributions.

Kostopoulos et al. conducted a large-scale WiFi performance measurement study using Raspberry Pis as active probes [9]. The use of Raspberry Pis as active probes in scientific studies adds to the importance of correctly understanding their WiFi protocol behavior.

III. WiFi protocol background

The WiFi protocol is “listen before talk”, which means that a WiFi station must check that the radio frequency is idle for a certain amount of time before starting a transmission. When a WiFi station begins a transmission procedure, it first selects a random number in the range 0 to CW_{min} and assign that number to a *back-off counter* [10, Section 10.2.2]. “CW” here stands for contention window. As long as the back-off counter is not zero, the station will listen for a single “slot time” at a time. If the radio frequency is idle during the “slot time”, the back-off counter is decremented by one. After the radio frequency has been busy with an active transmission, the station must wait for a minimum delay of AIFS before counting down. The duration of “slot time” and AIFS depends on the protocol version. For the setup in this paper, the values are 20 μs for “slot time” and 70 μs for AIFS. When the back-off counter becomes zero, the station starts transmitting. Whenever two stations count down to zero simultaneously, a collision occurs, which lasts for the duration of the longest colliding transmission.

When a station is involved in a collision, the station chooses a new random value for its back-off counter. The interval size from which the station selects random values is a function of how many times a packet has been retried. The back-off window size doubles with each new retransmission of the same packet, up to a maximum value of CW_{max}.

IV. Model description

In this section, we present a high-level overview of the model. The model is relevant to the results in this paper because it is a tool for translating the IEEE 802.11 protocol description into predictions about the latency distribution of a WiFi network. These predictions are then tested against real-world network measurements to check if the model matches reality. When the model does not match the measurements, it is either because the model does not capture the protocol specification correctly or because the real-world implementation deviates from the protocol description.

Our model is based on the ΔQ mathematics described by Thompson and Davies [4]. ΔQ is the delay and loss introduced by a network path or a network element. It describes how far the network is from delivering the “perfect service” of zero delay and no loss. ΔQ can be described by an improper random variable describing the probability distribution over the possible latency, with loss captured as infinite delay.

Modeling the ΔQ of the WiFi protocol consists in describing the latency distribution and loss potential of every possible path through the protocol state machine. In practice, we do this by associating each transition in the protocol state-machine with a ΔQ value. ΔQ properties allow us to add up the ΔQ contributions from each state transition. We can also combine the contributions of each possible path through the

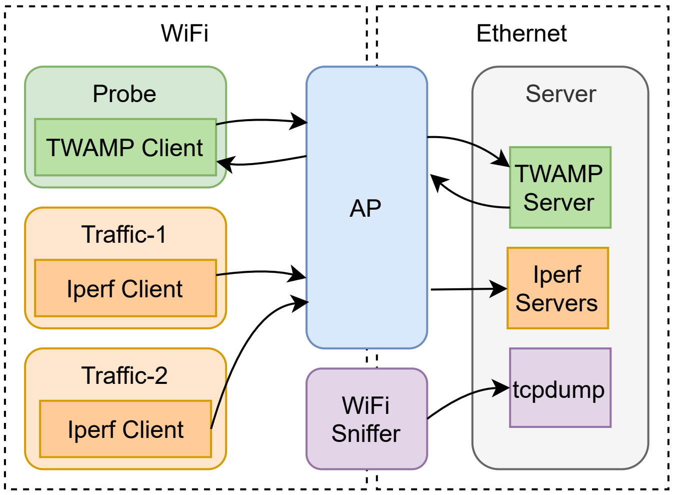

Figure 1. The testbed setup

state machine so that the latency distribution of the complete system can be described accurately [4].

Because of the large number of possible states in a complex WiFi network, the practical way to compute latency distributions is to do a Monte Carlo evaluation of the model.

V. Testbed details

In this section, we describe the testbed setup and how measurements are taken.

The testbed consists of four Raspberry Pi 4B machines and a WiFi access point, arranged as shown in figure 1. One machine, called the Server, is connected to the WiFi AP with a 1 Gbit/s Ethernet cable. The Server runs a Two-Way Active Measurement Protocol (TWAMP) server, two iperf servers, and sniffs WiFi packets using an Alfa AWUS036ACH WiFi adapter. Two of the machines, called Traffic-1 and Traffic-2, run UDP iperf uploads to the iperf servers. The final Raspberry Pi, called the Probe, runs the TWAMP client and is our source of latency measurements. All the Raspberry Pis run NixOS and use the Broadcom BCM4345/6 WiFi chipset with version 7.45.229 of the brcmfmac43455 firmware.

We designed the experiment to measure round-trip WiFi latency when there are either one or two “always-on” stations competing for access to the frequency. TWAMP measures one-way latency in each direction, as well as round-trip delay. We control the number of competing stations by using UDP uploads because UDP avoids transport-level ACKs in the reverse direction. The UDP uploads are configured to transmit more traffic than the WiFi link can support so that the competing stations can be considered to be “always-on”. The Iperf clients send packets with a payload of 200 bytes. Each experiment is run for 250 seconds, and the TWAMP client samples one round-trip every 500 ms. Experiments are conducted in an environment where no other WiFi stations are active on the relevant channel. The WiFi modulation and coding scheme is set to a constant 1 Mbit/s.

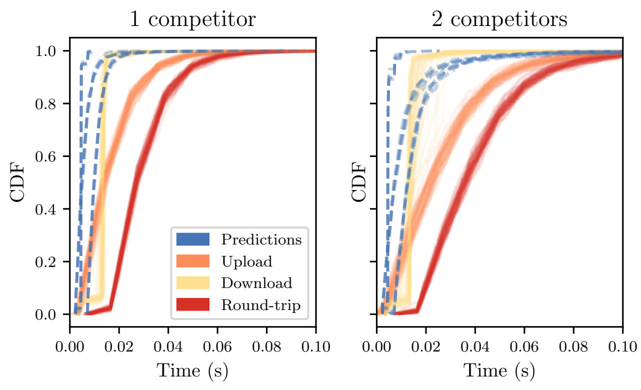

Figure 2. Initial predictions and testbed measurements

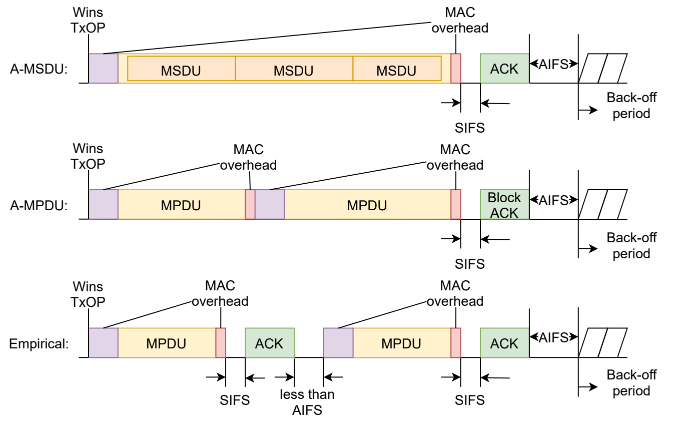

Figure 3. The A-MSDU and A-MPDU aggregation methods from the 802.11 protocol compared to what we empirically observe for the Raspberry Pi 4B

We configure the model to approximate the same setup as closely as possible, including modeling the effects of packet sizes, the WiFi modulation and coding scheme, competition for the channel, and the 500 ms TWAMP sampling interval. Then, we compare model predictions to measurements from the testbed.

VI. Results and iteration

Our first results showed a large discrepancy between model predictions and measured latency. Figure 2 shows predictions and measurements for the upload, download, and round-trip of TWAMP plotted as CDFs. Figure 2 shows 50 runs of the testbed and five runs of the model starting from different random seeds. The difference between model predictions and measurements led us to investigate both the model and the behavior of the testbed computers.

A. Investigating aggregation behavior

Traces from our testbed show that the WiFi protocol implementation on the Raspberry Pi 4 does not match the IEEE 802.11 protocol specification. We found that the Pi 4 aggregates packets by making the time interval between an ACK and the next packet less than the AIFS interval specified by the protocol. This is not the same as A-MSDU or A-MPDU aggregation as specified in the WiFi standard. The Raspberry Pi transmits a single MPDU packet, waits for the ACK, and then transmits another MPDU packet without waiting for the required AIFS duration. Figure 3 illustrates the difference between A-MSDU, A-MPDU, and the aggregation used by the Raspberry Pi.

Using a modulation and coding scheme of 1 Mbit/s and a payload size of 200 bytes, we see that the Raspberry Pi typically transmits four packets back to back before invoking the back-off procedure. The total time spent transmitting these four packets is 10,536 milliseconds.

Aggregating packets in this way is a breach of the WiFi protocol because the protocol requires a random back-off procedure after each successful transmission [10, Section 10.3.3]. The observed behavior is not consistent with the specification for multiple packet transmissions in a single EDCA TxOP [10, Section 10.22.2.7] because the TxOP limit is set to 0 for the best effort traffic class. With this configuration, only one transmission is allowed in each TxOP [10, Section 10.22.2.8].

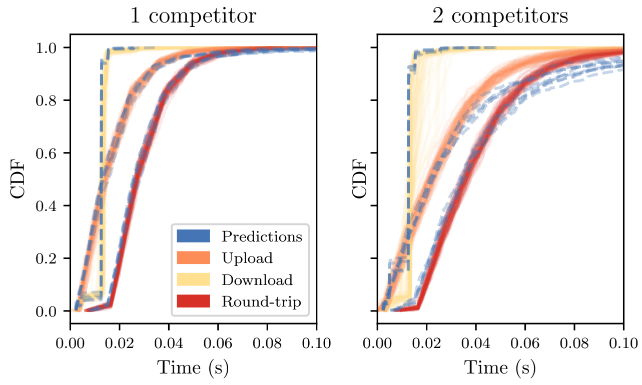

Figure 4. Predictions and test bed measurements with packet aggregation

This unforeseen aggregation behavior affects the latency results measured in the testbed. It may explain the observed difference between the testbed measurements and our model predictions. We implement the same aggregation behavior in the model to see how it affects our predictions.

B. Modeling packet aggregation

We changed our model to capture the aggregation behavior we observed in traces from the testbed. Figure 4 show the predictions of our model with the new modifications. As we can see, the prediction now much more closely matches the observed behavior.

Figure 4 shows that we over-estimate the long tail of the latency distribution, and especially so when there are two competing stations. One plausible reason for this is that the probability of packet collisions between the two competitors is higher than our model predicts it to be. If this is correct, it could be due to errors in the back-off sampling procedure, which might make collisions more common.

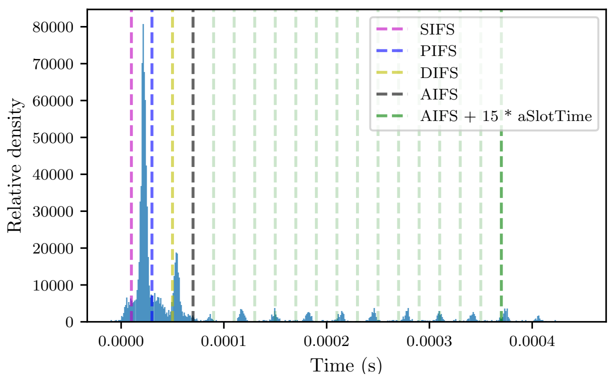

Figure 5. Inter-packet delays measured by the WiFi sniffer for packets of size 200 bytes and rate 1 Mbit/s.

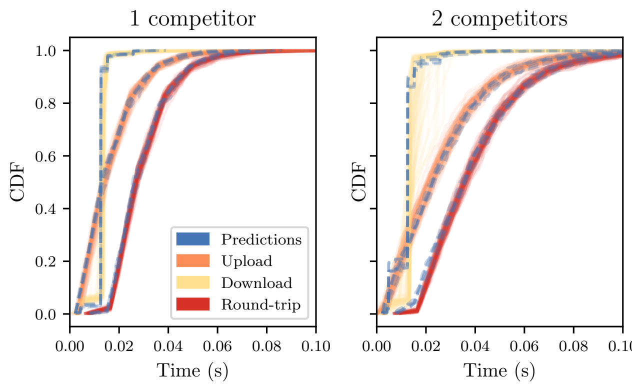

Figure 6. Predictions and test bed measurements with both packet aggregation, a smaller contention window and delayed incrementation of the retry counter

C. Investigating back-off behavior

We use logs from the WiFi sniffer to calculate the distribution of waiting times between subsequent transmissions from the same station for the particular case when that station is alone on the WiFi frequency. Because there is no competition on the frequency in this scenario, the timings between subsequent packets should be entirely due to the random back-off counter of the station itself.

Figure 5 shows the resulting distribution of inter-packet delays. As we can see, the station frequently waits for a period shorter than AIFS. When the station waits for longer than AIFS, we count 11 peaks in the distribution. Eleven is a lower number than we were expecting. According to the standard, there should be $2^4 = 16$ distinct peaks in the distribution when the contention window exponent is four.

Testbed traces also show that back-offs are not immediately exponentially incremented when collisions occur. We measure the waiting time between retransmitted packets and the previous transmission when a station is alone on the channel. Sometimes, our WiFi sniffer captures packets that are not ACKed by the AP so that we observe the back-off behavior of retransmissions. This analysis suggests that stations choose from the same 11-slot contention window for the first retry attempt, but they exponentially increase the contention window after more collisions.

D. Modeling altered back-off behavior

Figure 6 shows the effect on our predictions of reducing the contention window from what the protocol specifies (with 16 distinct values) to what we empirically observe (with 11 distinct values) and delaying the incrementation of the retry counter. Our predictions now almost perfectly match the empirically observed delays. This suggests we are close to capturing the correct behavior. More work is needed to conclude that the changes we have modeled are the only changes made to the WiFi protocol in the Raspberry Pi 4 and that all changes are modeled correctly.

VII. Conclusion

We have presented results showing that WiFi protocol implementations do not always match the 802.11 specifications. Our primary contributions are to document that the WiFi driver on the Raspberry Pi 4B aggregates packets in a way that violates the WiFi standard and that the Raspberry Pi uses too few distinct values for its back-off counter. Our findings point to a possible blind spot in the WiFi certification procedure. It also has implications for work on modeling or simulating WiFi performance and the interactions between WiFi and other protocols. We will investigate whether the same kind of protocol changes are present in other WiFi devices in future work.

References

[1] B. Forum, "TR-452.1 Quality Attenuation Measurement Architecture and Requirements," Tech. Rep., 2020. [Online]. Available: https://www.broadband-forum.org/download/TR-452.1.pdf

[2] L. Leahu, "Analysis and predictive modeling of the performance of the ATLAS TDAQ network," Ph.D. dissertation, Bucharest, Tech. U., 1 2013.

[3] D. C. Reeve, "A New Blueprint for Network QoS," Ph.D. dissertation, University of Kent, University of Kent, Canterbury, Kent, UK, 8 2003. [Online]. Available: http://www.cs.kent.ac.uk/pubs/2003/1892

[4] P. Thompson and N. Davies, "Towards a RINA-Based Architecture for Performance Management of Large-Scale Distributed Systems," *Computers*, vol. 9, no. 2, p. 53, 6 2020. [Online]. Available: https://www.mdpi.com/2073-431X/9/2/53

[5] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 3 2000.

[6] I. Tinnirello and G. Bianchi, "Rethinking the IEEE 802.11e EDCA performance modeling methodology," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 540–553, 2010. [Online]. Available: http://ieeexplore.ieee.org.

[7] G. Bianchi and I. Tinnirello, "Remarks on IEEE 802.11 DCF performance analysis," *IEEE Communications Letters*, vol. 9, no. 8, pp. 765–767, 8 2005.

[8] B. I. Teigen, N. Davies, K. O. Ellefsen, T. Skeie, and J. Torresen, "A Model of WiFi Performance With Bounded Latency," 1 2021. [Online]. Available: http://arxiv.org/abs/2101.12512

[9] N. Kostopoulos, S. Gjeci, K. Baumann, P. V. Vuletic, and K. Stamos, "WiFiMon: Combining Crowdsourced and Probe Measurements for Wi-Fi Performance Evaluation," in *16th Conference on Wireless On-Demand Network Systems and Services, WONS 2021*, 2021.

[10] The Institute of Electrical and Electronics Engineers, "IEEE Standard for Information technology – Telecommunications and information exchange between systems LAN and MAN – Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2016," pp. 1–3534, 2013. [Online]. Available: http://ieeexplore.ieee.org/document/7786995/