Known Performance Issues Are Prevalent in Consumer WiFi Routers

Bjørn Ivar Teigen*†, Kai Olav Ellefsen*, Tor Skeie*, Jim Torresen*
*Dept. of Informatics, University of Oslo, Oslo, Norway
†Domos AS, Oslo, Norway
bjorn@domos.no, kaiolae@ifi.uio.no, tskeie@ifi.uio.no, jimtoer@ifi.uio.no

***Abstract*—WiFi is a crucial part of internet infrastructure. Performance issues are common in WiFi networks, but well-tested solutions exist for some known problems. In this work, we look for evidence of known WiFi issues in chipsets commonly found in consumer WiFi routers. We document serious performance problems in most of the tested WiFi chipsets and point to existing solutions for the detected problems. The prevalence of these problems has implications for network management, network research, and for users and creators of performance-sensitive network applications running over WiFi. To our knowledge, this is the most comprehensive documentation of these issues to date.**

I. Introduction

WiFi carries a significant portion of home network traffic, and therefore WiFi is a crucial part of the overall end-to-end performance of the internet. As reported in [1]–[3] WiFi often contributes significantly to the end-to-end latency and is frequently the throughput bottleneck. In this work, we diagnose home WiFi router chipsets for problems with known solutions. A diagnosis is achieved by analyzing data gathered using open source network measurement tools without requiring data collection software on the router.

Installing software on home routers is not always possible without the support of the device manufacturer. In addition, differences between WiFi driver interfaces make collecting detailed statistics on WiFi routers prohibitively complicated, especially when comparing different WiFi chipset vendors. In this work we use readily available open-source tools to measure network performance across the WiFi link. We design a test procedure and accompanying analysis tools based on [4] and [5] to detect performance issues without requiring software on the WiFi routers.

Our main contribution is reference measurements for several WiFi chipsets commonly found in consumer WiFi routers. We show that serious WiFi performance problems are prevalent in chipsets that make up a large portion of the home router market.

II. Background

There are many possible reasons why WiFi networks fail, and not all of the issues are under the control of the WiFi router. This work focuses on detecting problems that can be solved through software updates of already deployed home WiFi hardware. We have selected two problems based on these criteria:

1) The problem is under the control of the WiFi access point
2) The problem has at least one known solution
3) The solution is not in conflict with the IEEE 802.11 standard
4) The problem can be detected by measurements across the WiFi link

A. Bufferbloat

Buffering refers to queuing of network packets at an interface. "Bufferbloat" describes excessive buffering in network components and the resulting unnecessary latency induced for packets waiting in line [6]. Some buffering is necessary to deal with bursts of traffic that exceed the link capacity. However, if the router fails to drop packets when its capacity is overloaded for an extended period and instead keeps all packets in a very long queue, the effect is unnecessary latency. The queue can grow until it eventually overflows, at which point packets are dropped, and the traffic sources scale back their transmissions.

Bufferbloat is conceptually similar to a denial of service attack. The presence of too much traffic causes a line to form, which effectively blocks latency-sensitive users from receiving satisfactory service. Many types of internet traffic are not very latency-sensitive. An example is video streaming services that use application-layer buffers. These applications can work well even in the presence of bufferbloat, which means they have weak incentives for tuning back their throughput when bufferbloat occurs. Trusting the end-points to behave well is therefore not a good strategy to deal with bufferbloat. Additionally, buffers can fill on smaller time scales than the typical round-trip times on the internet. This discrepancy in time scales limits the utility of endpoint-based congestion control methods. For these reasons, reducing bufferbloat must at least in part be handled at each queue within the network, and this is especially important at the throughput bottlenecks. Several queuing algorithms for reducing bufferbloat have been developed [7], [8].

B. The WiFi performance anomaly

The WiFi performance anomaly is the phenomenon of throughput for all stations reducing to the level of the slowest station when different stations use different PHY rates. It was

This work is partially supported by The Research Council of Norway (RCN) as a part of the industrial PhD program under grant number 297244.

978-3-903176-36-2

first reported in [4]. The WiFi distributed coordination function was designed to allow all stations to transmit the same number of packets on average. Letting stations send the same number of packets works well as long as all stations spend the same amount of time transmitting each packet. It stops working well when some stations transmit more slowly than others because the slow stations spend more airtime; that is, they keep the WiFi channel busy for longer every time they transmit a packet. The result is that no station can transmit with higher throughput than the slowest station.

In the original paper on the WiFi performance anomaly [4], the authors detect it in a test-bed setup by running TCP uploads from four stations connected to the same Access Point. This test detects the performance anomaly in the upstream direction but does not capture the behavior of the Access Point in the downstream direction. In [9] the authors detect the performance anomaly by measuring PHY rate from within the WiFi driver on the Access Point. Measuring directly in the WiFi driver is likely the most accurate method. However, it requires access to the router software and in some cases altering the driver code to enable the necessary measurements.

C. Known solutions

In [9] the authors present a solution to both bufferbloat and the WiFi performance anomaly, with an implementation available in the mainline Linux kernel. We benchmark one router which has these fixes to serve as a representative of state of the art, see section V.

III. Method

A. Data collection

We use the FLExible Network Tester (Flent) to take measurements [5]. Flent is a wrapper for common network tools such as netperf, fping and IRTT. We designed the testing procedure to be as robust and simple as possible by directly producing the situations in which bufferbloat and the WiFi performance anomaly occur.

The test is Realtime Response Under Load (RRUL, [10]) modified to start TCP flows in the downstream direction only because the router controls downstream but not upstream queuing. ACKs still flow in the upstream direction. This change makes the test minimally dependent on the properties of the stations (STAs). The test consists of four parallel TCP streams to each STA and UDP measurements of round-trip latency. There are three UDP flows, marked voice, best effort, and background respectively, using DSCP. The UDP and ping measurements also record packet loss. All TCP streams are marked best effort using DSCP.

Tests are performed towards two stations (STAs) in parallel, as shown in figure 1. Each test consists of two phases. First, one run of the test procedure is performed with both STAs located close to the router and at approximately the same RSSI. See the ”Same RSSI” condition in figure 1. Once the first phase has been completed, one of the STAs is moved further away from the router until its RSSI is significantly lower. We repeat the test with the stations in their new positions, see the ”Different RSSI” case in figure 1. WiFi performance depends on the radio environment. To minimize the impact of environmental variations, we run the experiments at several different points in time in each configuration.

B. Diagnosing problems

1) Diagnosing Bufferbloat: Bufferbloat is characterized by large latency values under load. We plot the cumulative distribution function (CDF) of latency for the best-effort traffic class of both STAs for both RSSI conditions. The loss-rates are also reported, see Table I. We compare the results from each router to measurements taken on a router running OpenWRT 19, which has the solution from [9] implemented. We consider this a comparison to the state-of-the-art because CAKE [7], the newest queue management solutions in the Linux kernel, has not been implemented specifically for WiFi.

Latency is measured using a UDP flow running alongside the TCP downloads. Flent uses the tool Isochronous Round-Trip Tester (IRTT) to measure latency with UDP. IRTT sends UDP packets with a fixed interval, and the default for Flent is to configure IRTT to send a packet every 200ms. We use these default settings in this work. By using a separate low-volume flow to measure latency and packet loss, we capture the extent to which high-volume TCP traffic can affect the performance of applications such as gaming and video conferencing.

In this work, we operationalize detection of bufferbloat through a comparison to the newest implementation in the Linux networking stack. To keep the diagnostics simple we set a cutoff at a 90th percentile latency of 200ms, which is 5.5% higher than STA0 in the same-RSSI condition. When diagnosing bufferbloat we look exclusively at the same-RSSI measurements because they are the most stable.

2) Diagnosing the WiFi performance anomaly: If the WiFi performance anomaly is present, the STA with the lowest WiFi rate determines the achieved throughput of both stations. In our scenario this means that STA0, which has the same RSSI for both test runs, will achieve lower throughput when the RSSI of STA1 decreases.

In the paper which originally reported the WiFi anomaly problem [4] the authors find that all stations achieve the same useful throughput when the anomaly is present. This means that in order to detect the anomaly, we only require that the useful throughput for the station that we move is reduced significantly. If the two STAs have similar throughput in both the same-RSSI and different-RSSI cases then we conclude the performance anomaly has not been fixed.

Thus, the diagnosis analysis consists of two steps:

1) Check that the moved STA has significantly lower throughput in the different-RSSI condition
2) Measure how similar the throughput of the two stations are. If the WiFi performance anomaly is present they will be similar for both RSSI conditions.

We plot the results and visually determine if the difference between the two cases is significant. If the moved STA does not obviously receive lower throughput in the different-RSSI

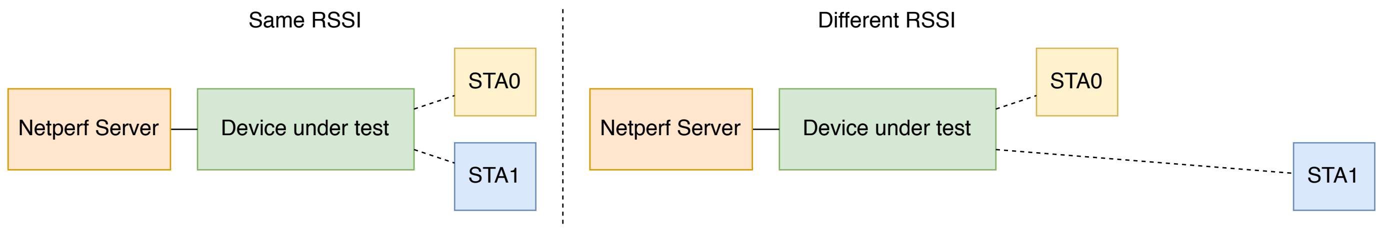

Fig. 1. The testbed setup. A dotted line indicates a wireless connection, and a full drawn line indicates an ethernet connection.

condition, then it was not moved far enough away from the router. Automating detection will be a topic of future work.

3) Detecting dropped connections: In preliminary testing we found that some routers cause one or both of the stations to disconnect during the test. This can not be a feature of the stations used for testing because it only happens for some routers. To detect this failure mode we look at the number of samples collected. For each run we count the number of 200ms intervals where no packets was received at the Flent server for any of the flows going to a single station. This serves as an indication that the connection was dropped for the station in question.

C. Verifying the stability of test results

We run the test 10 times on a router with OpenWRT 19 on which the bufferbloat and WiFi anomaly fixes from [9] are implemented. We expect the results from the 10 tests to be very similar, and specifically for the variation between runs of the same phase to be smaller than the variation between the different phases.

IV. Experiments

We now run the benchmark on a selection of routers which are common in European households. The routers have been selected to span several WiFi chipset vendors. Broadcom, Quantenna, Intel, and Qualcomm Atheros are represented. Each chipset is tested 5 times in each RSSI configuration. The specific chipsets and summary results are listed in table I.

V. Results

Figure 2(a) shows the test of an Atheros-based router running OpenWRT 19. This router has the bufferbloat and WiFi anomaly fixes from [9]. The x-axis shows the sum of throughput for all four download TCP flows. Each of the ten runs has a duration of 60 seconds. The results clearly show that the stationary STA0 is not adversely affected by the fact that STA1 is moved to a position with lower RSSI because the throughput of STA0 is slightly higher in the different-RSSI condition. The latency of STA0 is not affected much either.

Figure 2(c) shows an example where both the WiFi performance anomaly and bufferbloat are present. STA0, which was in the same place for both the same (green) and different (red) RSSI conditions, shows a much lower TCP throughput for the different-RSSI case. Also, notice that both stations have very similar TCP throughput for both conditions.

The latency results indicate that this chipset has the potential for several hundred milliseconds of bufferbloat. The CDF shows that for both STAs, and in both RSSI conditions, more than 75% of the packets receive a latency of more than 500 milliseconds. This chipset does not cause dropped connections, and packet loss rates are consistently less than 0.1%.

Table I shows a summary of results for several common WiFi chipsets. The bufferbloat results are averages for each STA of the 50th, 90th, and 99th percentile from each of the five runs in each configuration and the average loss value in percentages. The reason for this level of detail is to show how a reduction of the RSSI of STA1 impacts STA0. In some cases, moving one STA causes increased latency for the other STA. BCM63168 is an example of this. For the WiFi performance anomaly, we state whether or not it is present. For dropped connections, we report the total number of 200 ms periods with no arrived packets overall tests.

VI. Discussion

Out of the 12 non-OpenWRT chipsets tested, we found evidence of the WiFi performance anomaly in 2 of them. Ten of the 12 routers display bufferbloat of more than 100ms above the state-of-the-art implementation across all measurements for the most stable same-RSSI condition. The worst case is the BCM6755 in the 2.4GHz band, a brand new WiFi 6 chipset, which has more than 3.3 seconds of buffering for the 50th percentile latency.

Our results also point to a trade-off in the Active Queue Management (AQM) algorithms used to solve the bufferbloat problem. The UDP flow used to measure latency sends one packet every 200ms, and so should see a minimal amount of packet loss due to the per-flow queuing structure described in [9]. Our results show a packet loss rate for the latency-measuring UDP flow of 1-2% under load for the station with good RSSI.

The diagnostics method does not require that stations are the same. The stations used here are not, which is the reason for differences between them in the same-RSSI condition.

VII. Conclusion

Bufferbloat was first reported in 2011 [6], and the WiFi performance anomaly was first reported in 2003 [4]. Solutions have been proposed for both issues, but deployments seem to be lagging. Airtime fairness solutions seem to have resolved the WiFi performance anomaly on most of the tested chipsets.

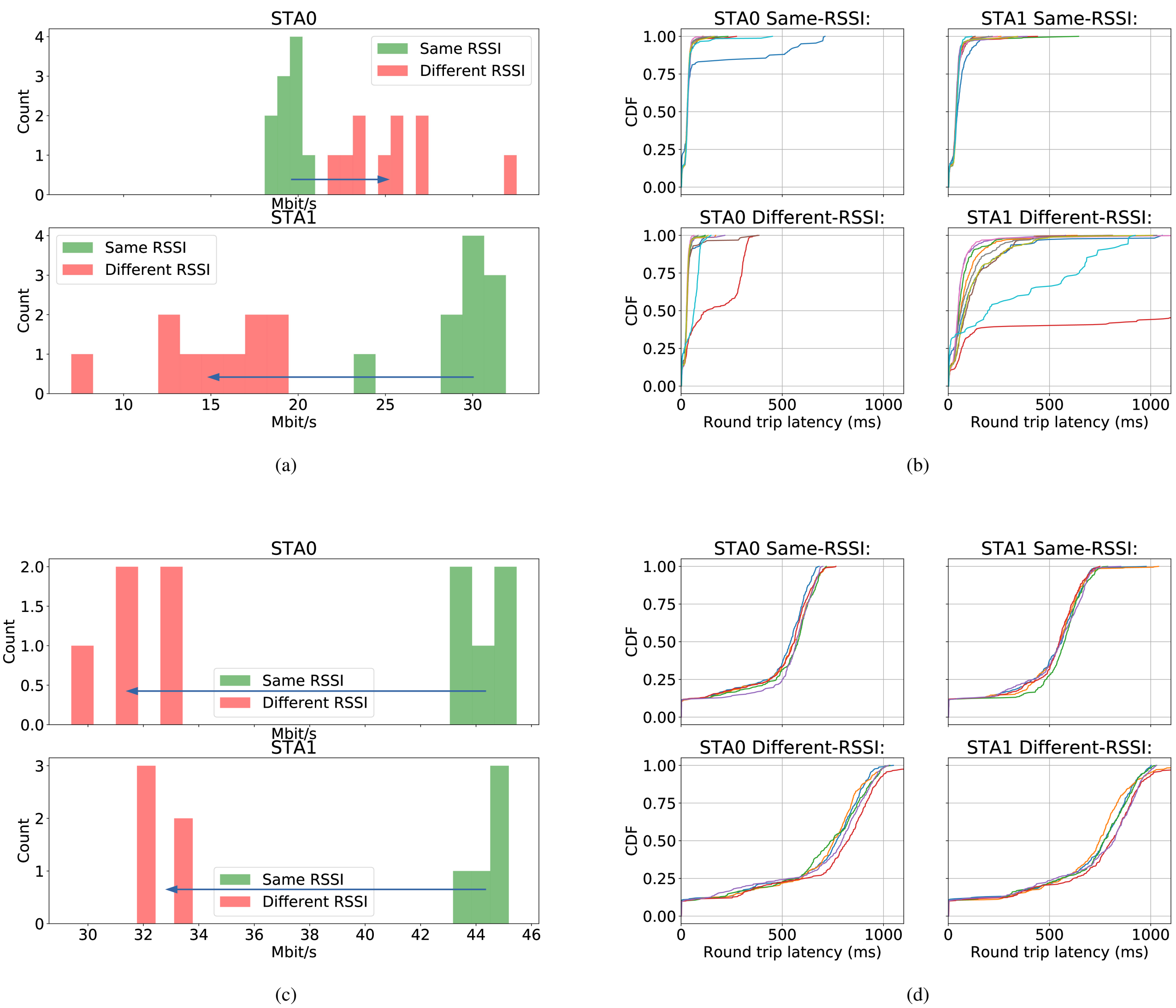

Fig. 2. Results for the OpenWRT reference router (a and b) and the BCM43602 chipset (c and d) in the 2.4GHz band.

However, bufferbloat is still a serious problem with most of the WiFi chipsets tested.

Considering how vital internet connectivity has become for most people, these results are troubling. Our results show significant quality issues with regular home WiFi equipment. It is feasible to resolve the quality problems because they can be solved with software updates of currently deployed hardware. The method presented in this work enables rapid diagnosis of common WiFi problems, and we have shown results for many common WiFi chipsets.

References

[1] J. Kennedy, G. Armitage, and J. Thomas, "Household bandwidth and the 'need for speed' Evaluating the impact of active queue management for home internet traffic," *Australian Journal of Telecommunications and the Digital Economy*, vol. 5, pp. 113–130, 6 2017.

[2] C. Pei, Y. Zhao, G. Chen, R. Tang, Y. Meng, M. Ma, K. Ling, and D. Pei, "WiFi can be the weakest link of round trip network latency in the wild," in *Proceedings - IEEE INFOCOM*, vol. 2016-July, 7 2016.

[3] K. Suiy, M. Zhou, D. Liu, M. Ma, D. Pei, Y. Zhao, Z. Li, and T. Moscibroda, "Characterizing and improving WiFi latency in large-scale operational networks," in *MobiSys 2016 - Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 347–360, 6 2016.

[4] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proceedings - IEEE INFOCOM*, vol. 2, pp. 836–843, 2003.

[5] T. Høiland-Jørgensen, C. A. Grazia, P. Hurtig, and A. Brunstrom, "Flent: The FLExible network tester," in *ACM International Conference Proceeding Series*, pp. 120–125, 12 2017.

[6] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet: Networks without effective AQM may again be vulnerable to congestion collapse," *Queue*, vol. 9, pp. 40–54, 11 2011.

[7] T. Hoiland-Jorgensen, D. Taht, and J. Morton, "Piece of CAKE: A Comprehensive Queue Management Solution for Home Gateways," in *IEEE Workshop on Local and Metropolitan Area Networks*, vol. 2018-June, pp. 37–42, 9 2018.

[8] K. Nichols and V. Jacobson, "Controlling queue delay," *Queue*, vol. 10, 5 2012.

Chipset/Driver	Band	WiFi Anomaly	STA/RSSI	Latency percentiles (Bufferbloat) 50th — 90th — 99th — loss	Missing samples
Ath9k	2.4GHz	Fixed	STA0/same	61.7ms — 189.6ms — 404.8ms — 1.087%	8
			STA0/different	82.8ms — 151.9ms — 319.5ms — 2.081%	51
			STA1/same	80.0ms — 135.7ms — 401.2ms — 0.436%	0
			STA1/different	421.9ms — 2293.5ms — 3189.0ms — 3.262%	43
BCM63269	2.4GHz	Fixed	STA0/same	333.7ms — 1843.2ms — 3333.6ms — 4.595%	718
			STA0/different	285.8ms — 1978.0ms — 3686.7ms — 2.987%	894
			STA1/same	271.9ms — 3225.1ms — 4980.5ms — 4.231%	426
			STA1/different	516.2ms — 2750.2ms — 5798.3ms — 2.735%	357
BCM43602	5GHz	Not fixed	STA0/same	561.4ms — 661.2ms — 698.6ms — 0.016%	29
			STA0/different	776.3ms — 942.5ms — 1046.1ms — 0.051%	3
			STA1/same	558.4ms — 676.0ms — 763.1ms — 0.054%	24
			STA1/different	782.3ms — 941.5ms — 1100.8ms — 0.050%	0
BCM4360	5GHz	Fixed	STA0/same	151.6ms — 198.4ms — 237.3ms — 0.281%	3
			STA0/different	125.5ms — 216.5ms — 349.0ms — 0.002%	13
			STA1/same	75.9ms — 103.5ms — 156.8ms — 0.002%	0
			STA1/different	154.6ms — 299.3ms — 661.9ms — 0.004%	9
BCM43217	2.4GHz	Fixed	STA0/same	358.3ms — 444.1ms — 619.8ms — 0.010%	9
			STA0/different	357.7ms — 554.7ms — 674.3ms — 0.007%	6
			STA1/same	395.4ms — 489.6ms — 630.0ms — 0.011%	3
			STA1/different	657.8ms — 2321.0ms — 5756.4ms — 0.893%	2
BCM4366	5GHz	Fixed	STA0/same	188.9ms — 213.5ms — 237.4ms — 0.001%	6
			STA0/different	292.6ms — 386.8ms — 499.7ms — 0.002%	5
			STA1/same	336.5ms — 373.8ms — 435.2ms — 0.006%	2
			STA1/different	446.9ms — 609.6ms — 730.6ms — 0.031%	0
BCM43217	2.4GHz	Fixed	STA0/same	696.2ms — 850.5ms — 931.5ms — 0.03%	5
			STA0/different	624.0ms — 1150.7ms — 1665.1ms — 0.04%	14
			STA1/same	378.9ms — 452.8ms — 539.0ms — 0.009%	2
			STA1/different	448.2ms — 1799.4ms — 3387.2ms — 0.69%	4
BCM63168	2.4GHz	Not fixed	STA0/same	380.8ms — 472.7ms — 543.5ms — 0.003%	0
			STA0/different	721.0ms — 2338.3ms — 3269.2ms — 0.010%	1
			STA1/same	387.7ms — 496.0ms — 591.7ms — 0.010%	1
			STA1/different	861.4ms — 2965.8ms — 4323.0ms — 0.059%	12
QT3840BC	5GHz	Fixed	STA0/same	809.2ms — 1015.6ms — 1146.7ms — 0.012%	3
			STA0/different	687.3ms — 1057.3ms — 1345.7ms — 0.004%	14
			STA1/same	328.5ms — 457.9ms — 624.7ms — 0.004%	0
			STA1/different	441.1ms — 1068.5ms — 2595.2ms — 0.093%	9
BCM6755	2.4GHz	Fixed	STA0/same	3374.4ms — 3937.5ms — 4574.0ms — 0.000%	3
			STA0/different	3308.1ms — 3781.9ms — 4266.6ms — 0.000%	0
			STA1/same	556.8ms — 1145.7ms — 2016.0ms — 0.000%	2
			STA1/different	978.9ms — 1260.7ms — 1533.7ms — 0.008%	0
BCM6755	5GHz	Fixed	STA0/same	756.1ms — 916.3ms — 1004.5ms — 0.000%	4
			STA0/different	711.2ms — 921.3ms — 1223.5ms — 0.002%	4
			STA1/same	267.5ms — 308.5ms — 378.3ms — 0.000%	1
			STA1/different	477.7ms — 689.6ms — 1164.2ms — 0.025%	0
WAV513	2.4GHz	Fixed	STA0/same	550.6ms — 762.7ms — 900.3ms — 0.033%	6
			STA0/different	552.7ms — 727.9ms — 1009.9ms — 0.024%	2
			STA1/same	391.4ms — 1089.5ms — 2048.1ms — 0.310%	48
			STA1/different	448.4ms — 1294.8ms — 2373.1ms — 0.123%	33
WAV524	5GHz	Fixed	STA0/same	84.6ms — 170.6ms — 245.1ms — 0.012%	4
			STA0/different	132.5ms — 209.6ms — 321.5ms — 0.007%	5
			STA1/same	79.6ms — 196.3ms — 1144.9ms — 0.025%	6
			STA1/different	139.7ms — 342.8ms — 939.0ms — 0.050%	0

TABLE I

SUMMARY OF RESULTS FOR EACH CHIPSET. GREEN INDICATES THAT WE FOUND NO CONCLUSIVE EVIDENCE OF KNOWN PROBLEMS. RED INDICATES THAT THE RESPECTIVE PROBLEMS ARE DETECTED.

[9] T. Høiland-Jørgensen, M. Kazior, D. Täht, P. Hurtig, and A. Brunstrom, "Ending the Anomaly: Achieving Low Latency and Airtime Fairness in WiFi," in *Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference*, p. 139–151, 2 2017.

[10] "Realtime Response Under Load (RRUL) Specification - Bufferbloat.net," 4 2020.