Abstract—With the advent of the internet of things, the number of wireless devices and the amount of wireless data traffic is increasing at an unprecedented rate. By 2020, the number of connected devices per person is expected to exceed 6.58. Due to the ubiquitous availability of IEEE 802.11n/ac (i.e., WiFi) in most modern homes and enterprise environments, the majority of the new home and enterprise-focused IoT devices that are entering the market use IEEE 802.11n/ac to connect to the internet. Our literature survey revealed that there are no prior measurement studies on the performance of IEEE 802.11n/ac’s MAC in dense IoT environments. Thus, in this paper, we conduct a comprehensive measurement study of various aspects of IEEE 802.11n/ac’s MAC using real IoT devices and realistic IoT workloads and present several useful observations that demonstrate the need to revise some of the key aspects of IEEE 802.11n/ac’s MAC to make it suitable for dense IoT environments. We conducted our study on a real testbed comprising a large number of Raspberry Pis deployed in a real-world environment and studied the impact of the density and type of IoT traffic on the throughput and frame aggregation sizes of the wireless system. We further studied the impact of TCP’s congestion control mechanism on the performance of IEEE 802.11n/ac’s MAC.

I. INTRODUCTION

The number of devices connected to the internet is increasing at an unprecedented rate. It is estimated that by the year 2020, the number of devices connected to the internet will exceed 24 billion [1]. While these devices will include conventional devices such as laptops, mobile phones, and web servers, the major portion of these devices will be contributed by the Internet of Things (IoT). IEEE 802.11n/ac, more commonly known as WiFi, is the most popular and convenient wireless technology to connect devices to the internet, especially in home and enterprise networks.

As more and more devices are connecting to the internet over WiFi, maintaining reliable WiFi connectivity is becoming problematic due to the distributed and uncoordinated nature of the medium access control (MAC) of IEEE 802.11n/ac. The direct effect of this limitation is that the usable share of the wireless bandwidth for individual clients decreases, which results in the degradation of performance metrics such as loss in throughput and increase in latency. This, in turn, leads to poor quality of experience for users in home networks and financial and functional losses for users in enterprise networks.

As the wireless environments are becoming increasingly dense due to the proliferation of the IoT devices and as IEEE 802.11n/ac’s current MAC is facing problems in dense IoT deployments, a need for a MAC protocol that is tailored for IoT devices is rising. In order to identify the aspects of the existing MAC that should be updated, it is imperative to first identify how various aspects of current MAC are holding up in the new dense IoT environments, and what are the root causes behind the deterioration of the aspects that do not perform well in the dense IoT environments. To study these aspects, and to identify the causes for any deterioration, in this paper, we conduct a comprehensive measurement study of various aspects of IEEE 802.11n/ac’s MAC using real IoT devices and realistic IoT workloads. More specifically, we characterize the throughput of individual IoT devices as well as the aggregate wireless system for five different emulated classes of IoT devices, which range from simple sensors that generate as little as 10kbits data to complex CCTV security camera networks that generate data at the rate of 5Mbps. We further study the impact of TCP’s congestion control protocol on the performance of IEEE 802.11n/ac’s MAC.

II. EXPERIMENTAL SETUP

A. IoT Traffic

Real world network applications transmit information across a spectrum of data sizes to and from the server. Since the purpose of this study is to observe the performance of MAC in dense IoT networks, we have only used uplink traffic where multiple clients send data to the server via a single AP. The particulars of the traffic rates are tabulated in Table I. We used the Eclipse-Paho [2] implementation of MQTT protocol to generate traffic to capture application overhead.

B. Testbed Setup

Our testbed is comprised of 45 Raspberry Pi 3 (RPi) [3] IoT prototype devices, deployed inside a 25ft $\times$ 16ft room. The motivation behind putting them in a single, albeit, a large room was to create an environment that is very densely populated with the IoT devices. The TCP variant that Raspbian Linux uses is TCP Cubic, which is also the default version of TCP in the latest releases of almost all flavors of Linux [4]. While RPIs come equipped with built-in WiFi radios, we used an external

---

Table I: Bit rates for different classes of IoT traffic

<table>
<thead>
<tr>
<th>Class #</th>
<th>App. Bit Rate</th>
<th>Examples IoT Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10 kbps</td>
<td>Stove, Lights, Dishwasher, Garage, HVAC</td>
</tr>
<tr>
<td>2</td>
<td>50 kbps</td>
<td>Smoke detector, Smart refrigerator</td>
</tr>
<tr>
<td>3</td>
<td>200 kbps</td>
<td>Audio streaming</td>
</tr>
<tr>
<td>4</td>
<td>2.5 Mbps</td>
<td>Smartphone, Laptop, Video conferencing</td>
</tr>
<tr>
<td>5</td>
<td>5 Mbps</td>
<td>Video Streaming, CCTV monitoring</td>
</tr>
</tbody>
</table>
Edimax N150 WiFi adapter [5] on each RPi to increase the wireless communication range and reliability. Our testbed also contains a dedicated RPi with the Nexmon patch [6], acting as a wireless sniffer. Every experiment contains a setup phase where we configure the experiment parameters (number of clients, traffic class), a IoT device communication phase (when the actual data transfer happens), and a log transfer phase (to query the collected log files to a central node).

III. EVALUATIONS

A. Throughput

From the throughput share per client in Figure 1, we observe that the per client throughput stays constant across different network sizes for class 1, 2, and 3 traffic. This happens because the combined application data rate of all clients for these traffic classes is much less compared to the channel capacity (which is 72.2 Mbps in our case). On the contrary, for class 4 and 5 traffic, the average per client throughput degrades drastically with the increase in network size. This shows that with increasing amount of offered traffic, IEEE 802.11n/ac's MAC starts to waste more and more of its available bandwidth, which can prove to be a major bottleneck in near future as consumers adopt more and more IoT devices that generate high network traffic. This last observation empirically validates the analytical prediction that Bianchi made in [7].

![Fig. 1: Avg. throughput of each class for network sizes](image)

**Fig. 1:** Avg. throughput of each client

![Fig. 2: Avg. aggregate lengths of AMPDUs](image)

**Fig. 2:** Avg. aggregate lengths of AMPDUs

B. Frame Aggregation

Figure 2 plots the average aggregate length of AMPDUs for the RPi clients transmitted for each traffic class and different network sizes. We calculated the average aggregate length of the AMPDUs from the frame bitmap in block ACKs. We make three important observations from this figure. First, for any given network size, aggregate lengths increase as the traffic class increases. This happens because higher traffic classes put more number of frames into the transmit buffer, and therefore, upon medium access, a client has more number of frames to aggregate. Second, for lightly loaded networks such as for class 1, 2, and 3 traffic, the average aggregate length slightly increases with the size of the network. This happens because an increase in the number of clients in the network induces larger backoff delays. Consequently, the number of frames that queue into the transmit buffer while the client waits for channel access also increases, which results in a net increase in the number of frames it has to transmit in a single aggregate. Third, for heavily loaded networks such as for class 4 and 5 traffic, we see an opposite behavior: the average aggregate length decreases with the networks size. This is counter intuitive because the increase in the medium access delay for heavily loaded networks is even larger compared to the increase for the lightly loaded networks, which should give rise to even larger sizes of AMPDUs. On further investigation, we found out that as MQTT uses TCP as the transport layer protocol, the TCP on RPi clients started perceiving the medium access delay as congestion in the network and kicked in its congestion control mechanism to decongest the network, which significantly reduced the rate at which it hands data to the lower layers in the network stack. As fewer amount of data arrives in the transmit buffer at the link layer, the sizes of the aggregates continue to diminish, which results in lost opportunity at the time when the client gets access to the medium, and thus experiences low throughput. Had TCP offered more packets to the lower layers, the average aggregate size would have increased, resulting in higher throughput. This shows that while IEEE 802.11n/ac's MAC benefits from frame aggregation in all scenarios, when paired with TCP, TCP reduces the usefulness of the frame aggregation. This, in turn, implies that conventional TCPs are not well suited for dense IoT networks with heavy traffic.

IV. CONCLUSION

The widespread adoption of IEEE802.11n/ac to deploy IoT networks pose several challenging problems in ensuring throughput performance reliability. The interplay between current transport control mechanisms and IEEE802.11n/ac MAC has a detrimental effect on system throughput in dense environments and therefore, the next generation network protocols have to be mindful of this interdependence.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation under grants # CNS 1616273 and CNS 1616317.

REFERENCES


