

Cross-layer Active Queue Management for Optimizing Mice Flows

Christoph Lindemann, Frank Martin Mehlhose, and Linus Curt Schulze
Department of Computer Science
University of Leipzig
04109 Leipzig, Germany
{cl, mehlhose, schulze}@rvs.informatik.uni-leipzig.de

Abstract — When competing with bulk traffic such as file download and video streaming, short-lived TCP and QUIC flows suffer from severe performance degradation. This holds in particular for wireless internet access over IEEE 802.11ax. To tackle this problem, we propose cross-layer Active Queue Management, cross-layer AQM, as a novel scheme for an integral management of the traffic control buffer and the MAC buffer of IEEE 802.11ax routers that provide wireless access to the internet. The main idea of cross-layer AQM lies in distinguishing between mice traffic and bulk traffic in the traffic control layer. We show that for mice traffic cross-layer AQM outperforms the current state-of-the-art scheme, FQ-CoDel, for important scenarios. Mice traffic with cross-layer AQM achieves almost 80% higher goodput than with FQ-CoDel. Furthermore, cross-layer AQM reduces the flow completion time for mice traffic by more than 10%. We also show that cross-layer AQM can achieve these performance improvements for mice traffic without degrading the performance of bulk traffic. These quantitative results are obtained by a measurement study of a prototype implementation in an IEEE 802.11ax testbed and by a ns-3 discrete-event simulation model.

Keywords — *Wireless and mobile networks, network performance and optimization, network testbeds, IEEE 802.11ax, FQ-CoDel.*

I. INTRODUCTION

Wireless access networks based on the IEEE 802.11 technology, commonly known as WiFi, have become central entry points to the Internet. According to the Cisco Annual Internet Report 2023, 55% of all Internet connections worldwide are carried over WiFi access networks. A large fraction of internet traffic flows, approximately 90 to 95%, are short-lived and transmit only small amounts of data. A significant and growing share of this short-lived traffic consists of web traffic delivered over QUIC. However, with commonly used active queue management mechanisms in WiFi routers, short-lived TCP and QUIC flows (a.k.a. “mice traffic”) are significantly disadvantaged compared to long-lived TCP flows (a.k.a. “bulk traffic”) such as file downloads or video streams. Due to its elastic nature, TCP fully utilizes the available bandwidth as well as the available buffer space most of the time. This phenomenon is known as the soaking effect. As a result, bulk traffic dominates the buffer space in the router and short-lived TCP and QUIC flows suffer from severe performance degradation.

Previous work [10] showed that for internet access via wired networks mice traffic achieves the best performance results with the active queue management scheme Fair Queueing - Controlled Delay, FQ-CoDel [7] regardless of the considered application scenario. However, these performance results may not hold for wireless internet access due to its considerably different technical properties. This holds in

particular for the current IEEE 802.11ax standard [2], [8] due to its significantly higher transmission speed in the downlink compared to internet access via legacy WiFi networks.

In this paper, we propose cross-layer Active Queue Management, cross-layer AQM, as a novel scheme for an integral management of the traffic control buffer and the net device queue of IEEE 802.11ax routers that provide wireless access to the internet. The main idea of cross-layer AQM lies in distinguishing between mice traffic and bulk traffic and buffering them in two separate FQ-CoDel [7] queues. Subsequently, cross-layer AQM uses a weighted round robin to prioritize mice traffic over bulk traffic but avoids starvation of the bulk traffic. This prioritized service also ensures that the queue of the MAC driver is not overloaded with bulk traffic. Thus, mice traffic can be transmitted with little delay. By prioritizing at the traffic control queue, cross-layer AQM optimizes the utilization of the buffer at the net device queue below. Distinguishing mice traffic from bulk traffic can be efficiently performed on the TCP flow based on the traffic patterns without any administrative effort.

To demonstrate the benefit of cross-layer AQM, we consider two application scenarios. The first scenario constitutes a home network with the bottleneck at the wired link connecting the IEEE 802.11ax router to the internet service provider and many devices starting short-lived TCP connections in regular intervals to send status updates. As second scenario, we consider an eHealth scenario with the bottleneck at the wireless downlink between the IEEE 802.11ax router and the mobile devices. In the home network, mice traffic with cross-layer AQM achieves up to almost 80% higher goodput than with FQ-CoDel. In the eHealth scenario cross-layer AQM yields more than 50% higher goodput and an almost 10% reduction in flow completion time (FCT) than FQ-CoDel. We also show that cross-layer AQM can achieve these performance improvements for mice traffic without degrading the performance of bulk traffic. These quantitative results are obtained by a measurement study of a prototype implementation in an IEEE 802.11ax testbed and by a ns-3 discrete-event simulation model.

The remainder of this paper is organized as follows. Section II summarizes related work on active queue management and on classifying mice traffic. In Section III, we introduce the proposed active queue management scheme, cross-layer AQM. Section IV describes the considered application scenarios: home network and eHealth application. Section V introduces the 802.11ax testbed and the simulation model. Furthermore, we present a comparative performance study for illustrating the benefits of the cross-layer AQM FQ-CoDel. Finally, concluding remarks are given.

II. RELATED WORK

Incorrect estimation of end-to-end bandwidth can lead to long queues at network bottlenecks, a phenomenon commonly referred to as *bufferbloat* [5]. To address this problem in a centralized manner, active queue management mechanisms have been proposed, to control buffer occupancy and prevent excessive queueing delays.

To address the performance degradation of bufferbloat Nichols and Jacobson propose the Controlled Queue Delay (CoDel) algorithm [12], which uses the minimum queueing delay as an indicator of congestion. The underlying assumption is that buffer occupancy up to a predefined threshold is acceptable. Once this threshold is exceeded, packets are dropped in a manner similar to Random Early Detection (RED) [4]. Today, the extended variant FQ-CoDel is predominantly used, as it additionally enforces fairness in buffer utilization among different flows [7]. Pan, Nataraja, Baker, and White propose the Proportional Integral Controller Enhanced (PIE) algorithm [13], which estimates queueing delay from average arrival rates and derives a packet drop probability accordingly. Kua, Nguyen, and Branch present a performance study of several modern active queue management algorithms [10]. They show that FQ-CoDel achieves the best performance results for both smart home and eHealth scenarios.

Opposed to [4, 7, 12, 13], the proposed research focuses on high-speed networks that consist of a wireless IEEE 802.11ax access network combined with a wired high-speed backbone, rather than exclusively wired high-speed networks. Similar to [10], we aim to present performance results obtained from measurement studies conducted in a real network, considering both a home network and an eHealth scenario.

Pokhrel, Vu and Cricenti proposed a model for a dynamically admission control approach, which sorts packets into the WiFi router's queue in such a way that all flows are treated approximately the same and compensates for WiFi data transmission deficiencies in networks without RTS/CTS [14]. This improved the throughput and delay of IoT devices in simulation and testbed.

Opposed to [14], the proposed cross-layer AQM implements closed-loop AQM that not only compensates for channel errors but also allows high goodput and low delay for mice traffic when competing with bulk traffic. Furthermore, we demonstrate the performance benefits of cross-layer AQM not only by simulation, but also a measurement study of a software prototype in an IEEE 802.11ax testbed.

Recently, the use of machine learning (ML) for improving WiFi performance and fairness has become a popular research area. The survey paper [15] mentions several approaches that try to improve performance in WiFi networks by adjusting the parameters based on traffic prediction using ML techniques.

Tang, H. Zhang, L. T. Yang and L. Chen [16] proposed the Efficient Sampling and Classification Approach (ESCA) for efficiently classifying short-lived and long-lived TCP flows. We adopt ESCA in Section III.A for classifying IoT flows.

III. DESCRIPTION OF THE CROSS-LAYER ACTIVE QUEUE MANAGEMENT FRAMEWORK

In this section, we present our proposed framework, cross-layer AQM. First, we outline how to classify short-lived TCP and QUIC flows. Then, we describe in detail the structure and operation of active queue management. Finally, we present how cross-layer buffer optimization can be achieved through the framework.

A. Classifying Mice Flows

In order to optimize the performance of short-lived TCP and QUIC flows, they must first be reliably detected and clearly distinguished from bulk flows. Each flow can be uniquely identified by the 5-tuple consisting of the source IP address, source port number, destination IP address, destination port number, and transport-layer protocol.

Following [14], the classification works as follows: Each new flow is initially classified as mice flow. Subsequently, the classification is refined dynamically according to the number of bytes transmitted during the lifetime of the flow. As in [14], we define a bulk flow as a flow whose transmitted data volume exceeds a multiple of the average size of all active flows in the system. The size of a flow, denoted by b , is the total number of bytes transmitted by this flow. The number of flows is denoted by K . B is the sum of bytes of all K flows. The average size of all flows, which are currently active, is denoted as \bar{B} , where $\bar{B} = \frac{B}{K}$.

A flow f is then classified as a bulk flow, if its size b_f satisfies:

$$\frac{b_f}{\bar{B}} \geq z \quad (1)$$

In Equation (1), z is a multiplicative factor representing the threshold relative to the average flow size. Note that a high value of z will increase the time required for a bulk flow to be classified correctly, because many packets must first be accumulated. To select an appropriate value for z that balances early bulk-flow classification with avoiding misclassification of mice flows, we analyzed two datasets: the Dartmouth Campus WiFi dataset [9] and the most recent CAIDA Internet traces [3]. Based on this empirical analysis, we determined values of z equal to 4 and 6 respectively. For the performance study presented in Section V, we deliberately choose a smaller value for $z = 4$. As outlined in Section III.B, mice flows, which spend their entire lifetime in TCP Slow Start, are particularly disadvantaged. Therefore, selecting a smaller value for z primarily prioritizes such flows and is closer to what is observed in real-world IEEE 802.11 data traces.

The interpretation of Equation (1) follows the same reasoning as in [16]. That is, if a flow's byte volume b is at least z times larger than the mean size of all flows \bar{B} , the flow is classified as a long-lived bulk flow. Otherwise, it remains categorized as mice flow. This ratio-based classification enables dynamic, self-adjusting distinction and automatically scales with changing flow statistics.

If a flow is classified as bulk flow, it is entered into a corresponding lookup table. The classification of an active flow is repeated regularly based on the updated data. Thus, it is possible that the class membership changes over time and

any misclassifications can be corrected. We collect the size data directly on the router and transfer them in aggregated form to the queue management system. Capturing these metrics can lead to performance problems, if the WiFi router does not possess high-performance CPUs. Therefore, cross-layer AQM is intended for deployment in WiFi-6 routers with high-performance CPUs, which are now widely available.

B. Active Queue Management

Before the data packets are forwarded in the WiFi router, they must be buffered accordingly. If the data rate at the outgoing link is lower than the data rate of the incoming packets, this buffer fills up over time, which can lead to various problems. The elasticity of TCP flows leads to both bandwidth and buffer space being fully utilized, the so-called *soaking effect* [10]. The heavy utilization of the buffers is denoted by *bufferbloat* [5]. This causes an unnecessarily high delay in data transmissions.

Despite its benefits, FQ-CoDel alone is insufficient to address low throughput and high delay for short-lived TCP and QUIC flows, since such flows commonly remain in the TCP Slow Start phase for their entire duration. In high-speed networks such flows require multiple round-trip times (RTTs) to transmit their data, even when the available bandwidth would be sufficient to complete the transmission within fewer RTTs. The effect is further amplified in IEEE 802.11 networks, where with a small congestion window fewer packets are combined into aggregated frames. Therefore, the per-packet overhead increases and channel efficiency is reduced, as shown in prior work on TCP over IEEE 802.11 [6].

We address these problems by active queue management in the traffic control (TC) buffer. For incoming packets, an automatic assignment into the two categories mice data and bulk data is performed as described in the previous section. Based on the source IP address, source port number, destination IP address, destination port number, and transport-layer protocol (i.e., TCP or UDP), the incoming packets can be uniquely assigned to an active flow.

To prevent mice packets from being queued in a buffer already containing many bulk packets, mice packets are separated and then further processed in a prioritized manner. Thus, separate FQ-CoDel queues are available for both the mice and bulk traffic. For those queues cross-layer AQM uses the default configuration of the Linux kernel. So, each FQ-CoDel queue has 1.024 subqueues and has a queueing limit of 10.240 packets. This ensures per-flow fairness within each class as active flows are separated into subqueues. Cross-layer AQM automatically assigns flows by dynamically adjusting the appropriate TC filters on the router's network interface. The last queue in the router is the net device queue (NDQ). Packets from the TC buffers are automatically transferred to the NDQ, provided it can accommodate packets, for subsequent transmission via the net device. The NDQ is typically structured as a ring buffer, which cannot be easily influenced. We extract the packets from the TC buffer using a weighted round robin. By adaptively controlling the weights for the two classes, we can optimize the performance of the mice data flows while ensuring that bulk flows do not experience significant performance degradation.

We assume that the mice flows are mostly TCP CUBIC or QUIC flows using CUBIC as congestion control as introduced in [11]. Note that cross-layer AQM is designed in a way that classification and prioritization are based solely on the transmission patterns and, therefore, works independently from the utilized transport protocol.

C. Cross-layer Optimization

As described in Section III.B, cross-layer AQM sorts packets into two different FQ-CoDel queues according to their flow type. By strictly separating the packets of both traffic types we can apply differentiated processing in the form of prioritization for mice flows. Since FQ-CoDel gives almost every flow its own subqueue, we mitigate the soaking effect.

The soaking effect is particularly problematic for mice flows, which transmit in short bursts with long idle periods. Thus, frequently encounter queues that are already filled by bulk flows sending in parallel. By separating them into different FQ-CoDel queues, cross-layer AQM achieves that TCP flows with high traffic can now only fill their own subqueue, but have no impact on the available queue capacity of the other flows. Since FQ-CoDel processes the subqueues in round robin, head-of-line blocking by a flow is avoided. This prevents a flow, which has already filled up its subqueue, from blocking packets of a newly buffered flow. The use of FQ-CoDel queues simultaneously mitigates the bufferbloat problem at the TC level for the mice flows. Since they are no longer sorted into a queue with the bulk flows, the fill level of the bulk queue no longer has a negative impact on the queueing delay of the mice packets in that queue. Algorithm 1 illustrates how bulk and mice packets are dequeued from their respective FQ-CoDel queues and transferred to the NDQ. Packets are scheduled using weighted round robin (WRR), with higher weighting for mice packets by a factor K .

The NDQ is usually implemented as a simple droptail queue. Basically, the same TC queue problems also exist in the NDQ. Since the mice flows are only short-lived, this queue will usually be filled with packets from the bulk flows (i.e., soaking effect). If a short-lived TCP or QUIC flow starts, its

Algorithm 1: Dequeue for two-class WRR of cross-layer AQM

Input: QUEUE q_1 , QUEUE q_2 , int n_bulk , int n_mice ;

```

1:  int rem1 = 0, rem2 = 0, turn = 0, K = 3, N1 = 0;
2:  while true do
3:      int N1 =  $q_1 \rightarrow count$ 
4:      int N2 =  $q_2 \rightarrow count$ 
5:      if (rem1 + rem2 = 0) or (N1 + N2 = 0) then
6:          int n1 =  $n\_bulk$ 
7:          int n2 =  $n\_mice$ 
8:          rem1 = min(N1, n1)
9:          rem2 = min(N2,  $[K \cdot n2]$ )
10:         turn = 1 - turn
11:      end if
12:      if turn = 0 then  $q_2 \rightarrow dequeue(rem2)$ 
13:      else  $q_1 \rightarrow dequeue(rem1)$ 
14:      if N1 = 0 then rem1 = 0
15:      if N2 = 0 then rem2 = 0
16:  end while

```

packets are placed at the back of an already bloated queue (i.e., bufferbloat). Since all packets of the FQ-CoDel queue are processed in round robin, the large buffer delay built up by the bulk flows remains and is transferred to the newly sorted mice and bulk packets. The sole use of FQ-CoDel changes little in this respect. FQ-CoDel does only mitigate bufferbloat in the TC queue level.

By prioritizing mice flows in cross-layer AQM, the NDQ can reduce the occupied buffer, such that after a short time the queue primarily contains mice packets. Due to the small number of packets in mice flows, they do not saturate the NDQ during their transmission period, consequently no significant bufferbloat occurs and the delay is greatly reduced. In summary, cross-layer AQM leverages flow information from the transport layer to optimize scheduling at the link layer, thereby mitigating bufferbloat for short-lived flows.

D. Determining Weights for Weighted Round Robin

Once the packets have been successfully classified into packets of bulk flows and packets of mice flows, they are forwarded to the network device queue. Strict prioritization of mice flows risks starving bulk flows when many mice flows are active concurrently. Therefore, we propose utilizing a weighted round robin scheme for scheduling the transmission of bulk and mice packets.

Let c_1 be the class of bulk packets and c_2 the class of mice packets. In order to respond adaptively to the current number of active data flows, we consider also the total number of active bulk flows and mice flows n_k and let N_k be the number of buffered bulk packets and mice packets with $k = 1, 2$.

Subsequently, we propose to set the maximum number of transmitted bulk and mice packets during a loop pass of the round robin scheme as:

$$\#packets(c_1) = \min(N_1, n_1) \quad (2)$$

$$\#packets(c_2) = \min(N_2, K \cdot n_2) \quad (3)$$

where K is a scaling factor. Following Section III.B, the value of K should be chosen sufficiently large to compensate the disadvantages experienced by mice traffic. These disadvantages are bufferbloat in the net device queue, TCP slow start, and the overhead caused by small frame aggregates. The optimal value of K depends on the startup behavior of the congestion control algorithm as well as on the IEEE 802.11 standard. For the slow start of TCP CUBIC over IEEE

802.11ax, we determine the optimal value of K for the considered scenarios in Section V.D.

As bulk and mice packets are serviced fairly within its queue, the throughput $T(f_k)$ can be guaranteed for the data flow f_k with $k = 1, 2$ with a given bottleneck bandwidth r :

$$T(f_1) \geq \frac{1}{n_1 + K \cdot n_2} r$$

$$T(f_2) \geq \frac{K}{n_1 + K \cdot n_2} r$$

Note that the tailored Weighted Round Robin introduced in (2) and (3) can be extended to any number of packet classes.

Using our proposed active queue management in the TC buffer of the IEEE 802.11ax router, we achieve that both buffers in the MAC layer are optimally utilized during the transmission periods when mice flows are active to ensure a fast transmission of the mice packets. Bulk flows are indeed treated adversely in these transmission periods. Due to the responsiveness of TCP, these data flows can recover quickly. During the long periods, in which only bulk flows are transmitted, we achieve an optimal buffer utilization for these flows through the separate FQ-CoDel queue for bulk traffic.

IV. APPLICATION SCENARIOS

Performance problems for mice flows over IEEE 802.11 networks mainly occur in scenarios where the link capacity of the access point reaches its limits and overfills its buffer. Based on this insight, we consider two realistic congestion scenarios in which IoT applications open short-lived TCP connections at regular intervals with long idle periods in between. In these scenarios, mice flows are particularly disadvantaged. In the following, we introduce these two scenarios that are later used to test and validate cross-layer AQM and describe the performance issues that arise.

A. Smart Home Scenario

In this scenario, we assume a modern small home network. In this network, IoT devices, mobile phones or personal computers run applications with short-lived transmission that compete for limited buffer capacity at an uplink with a node running a bandwidth-intensive application, such as a video conference call. This scenario is closely related to a home network scenario presented in [10].

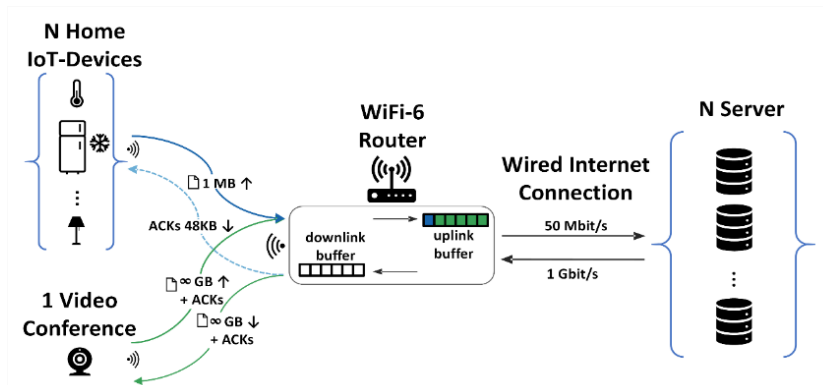


Figure 1: Illustration of Smart Home scenario

An overview of this scenario can be seen in Fig. 1. A node in the network continuously sends and receives large amounts of data by participating in a video conference. We assume that the video conference utilizes a common streaming protocol and sends data via TCP. Correspondingly, there are also flows of ACKs in both directions, which can be neglected in terms of size compared to the streaming data. At the same time, there are many devices and applications generating short-lived transmissions, such as smart light bulbs, thermostats, refrigerators, mobile phones, personal computers, and other smart home devices.

These devices generate short-lived TCP transmissions by periodically sending status or measurement data to remote servers for further processing. These servers are not on the same local IEEE 802.11ax network, so all data must be sent over the router's wired uplink. We assume that each device transmits a data packet of 1 MB every 1.5 seconds. As the data is transmitted using TCP, each packet additionally incurs approximately 48 KB of downstream ACK traffic. Since this scenario is a home network, we assume a connection to the ISP with 1 Gbit/s downlink and 50 Mbit/s uplink, that is quite common for industrialized countries.

As shown in Fig. 1, the low uplink speed causes a bottleneck in the uplink buffer at the IEEE 802.11ax router. Due to the elasticity of the bulk flows, the uplink buffer is always filled with data from the video conference. If a device generating mice traffic starts a transmission, that short-lived flow competes with the bulk data for the uplink buffer capacity. Due to the short transmission time, TCP cannot establish fairness in resource usage, so mice flows are disadvantaged here. Since wireless data transfer in the Gbit range is possible in IEEE 802.11ax, the downlink buffer should always be empty, as all data can be forwarded to the home network without delay. For the evaluation in the IEEE 802.11ax testbed and in the ns-3 simulation model, we use a dynamic number of devices for this scenario, which generate mice TCP flows, as well as one device, which permanently generates bulk traffic in order to map the video conference.

B. eHealth Scenario

This scenario is intended to illustrate the role of short-lived transmissions in healthcare. In hospitals, numerous applications are conceivable, for example, for monitoring patients. At the same time, the use of telemedicine increases other types of data traffic, which compete directly with short-lived transmissions and can jeopardize the required QoS properties.

An overview for this scenario is provided in Fig. 2. We assume that there are numerous active devices monitoring patients' vital signs, such as continuous monitoring of body temperature or heartbeat. Just as in the home network scenario, we assume that these devices send a small amount of data of 1 MB to a central server every 1.5 seconds, and accordingly TCP ACKs are upstreamed via IEEE 802.11ax. At the same time, there are physicians in the hospital who have a video stream with their patients as part of a telemedicine treatment and may be streaming other necessary data in parallel.

In this scenario, we assume two physicians treating patients in parallel. That is two concurrent video streams connected to the same IEEE 802.11ax access point. Similar to the home network scenario, these video streams cause large amounts of bulk data in both upstream and downstream directions. Since all connections use TCP, ACKs also flow in the opposite direction, but these can be neglected due to the small amount of data. In a modern hospital, it can be assumed that it is connected to the Internet via a high-performance fiber optic Internet connection. We assume a speed of 10 Gbit/s up- and downstream. Therefore, the uplink buffer is not a bottleneck. The bottleneck is the slow wireless downlink. The bulk data and the mice ACKs accumulate in the downlink buffer at the IEEE 802.11ax router.

Therefore, this scenario is fundamentally different from the first one. In the home network scenario, mice data competes with bulk data for scarce buffer capacity. Here, on the other hand, the mice ACKs compete with the bulk data in the buffer.

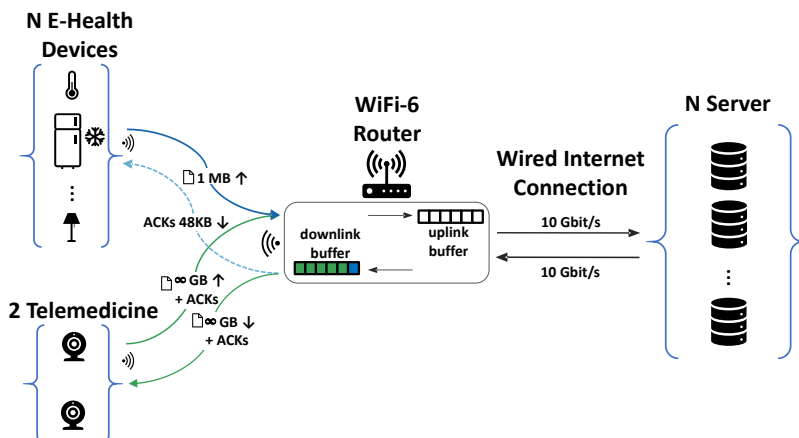


Figure 2: Illustration of eHealth scenario

V. QUANTITATIVE EVALUATION

A. The IEEE 802.11ax Testbed

We build our testbed with Intel NUC small computers as nodes and a central high-capacity server for monitoring the nodes, handling connections and aggregating measurements. We install an Intel AX200 WiFi adapter in the nodes for wireless communication. As IEEE 802.11ax router, we choose UniFi Lite 6 from Ubiquiti with default settings for DCF, EDCA and DTIM. We choose the open source operating system OpenWRT version 21.02.2. OpenWRT is a lightweight Linux distribution specifically adapted for network hardware [18]. This allows us to run cross-layer AQM as described in Algorithm 1, as a C-module in memory on the router as well as to capture and evaluate the network traffic on the router. All devices in the testbed are additionally connected via Ethernet. This subnet is used for time synchronization via the network time protocol (NTP) as well as for the exchange of control messages between server and clients during the experiment run. The setup of the testbed is illustrated in Fig. 3. In both subnets, the devices communicate via IPv4 and TCP CUBIC. In the testbed, the devices are placed about 3 meters from the WiFi access point.

The traffic patterns of devices generating mice flows, as considered in our scenarios, is characterized by periodic transmissions that resemble realistic application behavior. Such traffic patterns require a traffic generator which is capable of explicitly modeling transmission intervals. To generate traffic for both mice and bulk flows, we use the MGEN traffic generator [1]. For the short-lived TCP flows MGEN is configured to periodically open a TCP connection and transmit 1 MB of data every 1.5 seconds. For the bulk flows a single TCP connection is kept open for the entire duration of an experiment run with the transmission limited only by the available bandwidth.

Fig. 4 to Fig. 8 present a comparative performance study for illustrating the benefits of the cross-layer AQM versus FQ-CoDel. The quantitative performance results are derived from measurements in the testbed. For the initial transient, each parameter combination is repeated multiple times and confidence intervals are determined by independent replicates. For the steady state analysis, the devices generating mice traffic (i.e. mice traffic sources) start transmitting at random times uniformly distributed within the first 10 seconds. Each measurement study is performed until 55,000 packets are successfully transmitted, and results are divided into 11 batches of 5,000 packets each. The first batch is discarded as the initial transient, while performance metrics are computed from the remaining batches using the batch-means method with 95 % confidence intervals.

In the smart-home scenario, the ethernet capacity between the access point and the central server is reduced to 50 Mbps to examine the queuing behavior under contention between mice and bulk TCP traffic. It should be noted that cross-layer AQM throttles bulk flows only during the mice transmission phase. Due to the responsiveness of TCP CUBIC congestion control, bulk flows rapidly recover and utilize the full available bandwidth once mice transmissions pause. Because our goal is to evaluate the performance of mice flows rather than the applications and devices generating them, mice flow

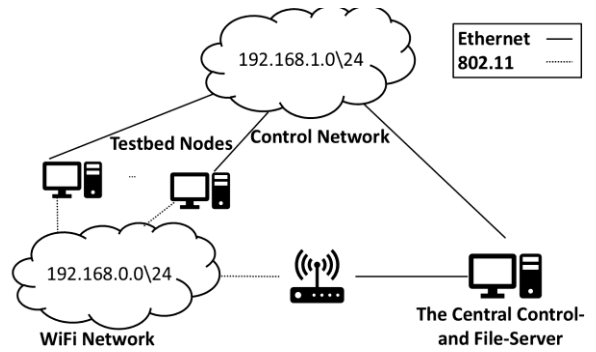


Figure 3: Illustration of IEEE 802.11ax testbed

goodput is evaluated only during active transmission periods, excluding idle times.

Fig. 4 shows the evolution of goodput over the first five mice transmission rounds. Goodput stabilizes after a single transmission round, indicating that the router is correctly classifying the data flows after the initial phase. Compared to FQ-CoDel, cross-layer AQM achieves up to 48% higher goodput for mice flows while maintaining equal goodput for bulk flows once the steady state is reached. This improvement over FQ-CoDel indicates that the adaptive mechanism quickly converges and efficiently utilizes the available bandwidth.

Fig. 5 depicts the flow completion time (FCT) as corresponding performance measure. Cross-layer AQM achieves up to 41% lower FCT for mice traffic compared to FQ-CoDel, confirming that the improved classification

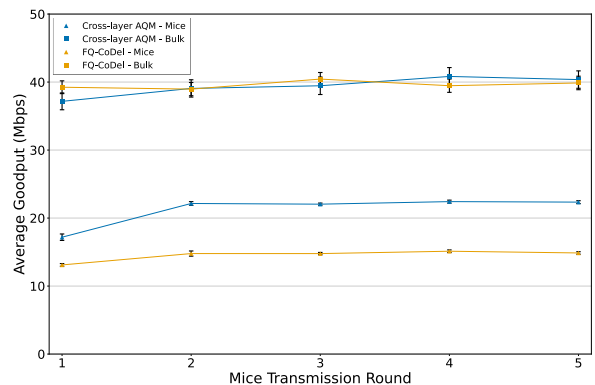


Figure 4: Measurement results for eHealth scenario: Initial transient goodput of Cross-layer AQM vs. FQ-CoDel

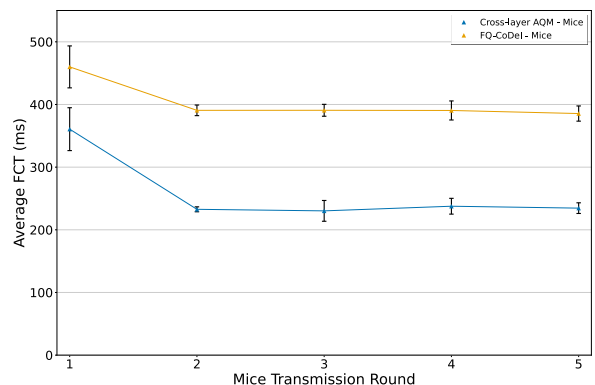


Figure 5: Measurement results for eHealth scenario: Initial transient mice FCT of Cross-layer AQM vs. FQ-CoDel

accuracy effectively reduces queuing latency for time-sensitive flows.

To further analyze scalability, Fig. 6 and 7 present measurement results for goodput and flow completion time (FCT) for a varying number of concurrent mice traffic sources in the eHealth scenario. Fig. 6 shows the goodput of mice flows increases significantly compared to FQ-CoDel, while the goodput of bulk flows remain nearly constant. In fact, cross-layer AQM achieves up to 59% higher goodput for mice traffic, demonstrating its ability to allocate bandwidth more efficiently under increasing load without penalizing best-effort traffic. This effect becomes more evident as the number of mice traffic sources grows, indicating that the adaptive

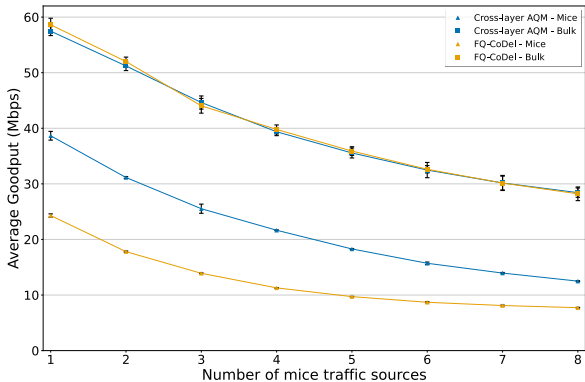


Figure 6: Measurement results for eHealth scenario: Steady state goodput of Cross-layer AQM vs. FQ-CoDel

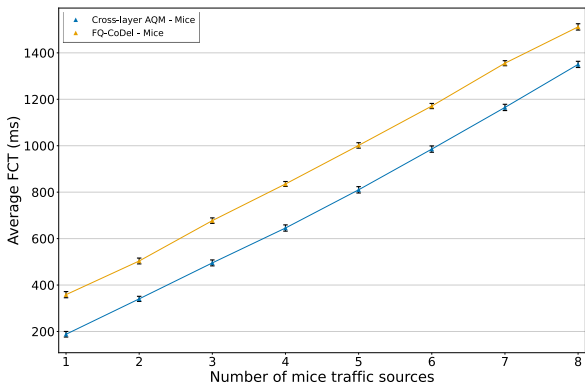


Figure 7: Measurement results for eHealth scenario: Steady state mice FCT of Cross-layer AQM vs. FQ-CoDel

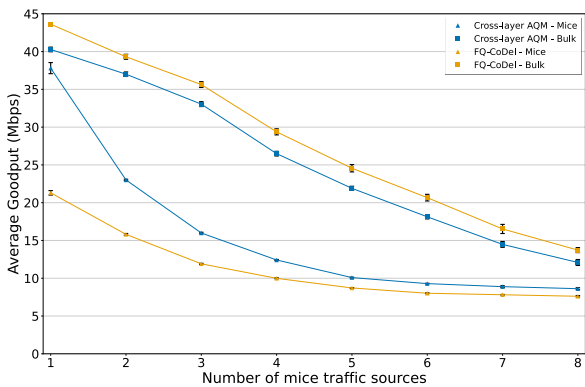


Figure 8: Measurement results for smart home scenario: Steady state goodput of Cross-layer AQM vs. FQ-CoDel

classification mechanism scales effectively with network density. Since the bulk throughput remains stable, cross-layer AQM does not compromise fairness.

Fig. 7 presents the corresponding FCT results. Across all tested numbers of mice traffic sources, cross-layer AQM consistently maintains lower FCT than FQ-CoDel, achieving an average FCT reduction of approximately 190 ms for mice flows. These results confirm that cross-layer AQM effectively and preserves low latency even for many concurrent mice flows.

Fig. 8 shows the goodput for a varying number of concurrent mice traffic sources in the smart home scenario. Here, cross-layer AQM achieves up to 77% higher goodput for mice traffic compared to FQ-CoDel. However, these performance gains diminish as the number of mice traffic sources increases and are accompanied by a decrease in bulk flow goodput. This indicates that prioritization of mice flows in the smart home scenario can lead to starvation of bulk traffic. This behavior is consistent with the discussion in Section III, as in this scenario the bottleneck is not the wireless link between clients and the access point. In summary, both the goodput and FCT results demonstrate that the proposed AQM provides substantial performance advantages, maintaining stability and responsiveness across a wide range of operating conditions.

B. Validation of NS-3 Simulation by Measurements Results in Testbed

As software package for discrete-event simulation, we use the network simulator package ns-3 [17]. Note that ns-3 has already implemented the IEEE 802.11ax standard as well as the FQ-CoDel queue management algorithm. We implemented cross-layer AQM as described in Algorithm 1 as a C++ module for the ns-3 simulation. Furthermore, we implement the smart home network and hospital network scenarios as described in Section IV in the ns-3 simulation. We use a varying number of mice traffic sources and two devices generating bulk traffic. As in the testbed, for each device the distance from the access point is set to about 3 meters in the simulation model using the uniform disc allocator of ns-3. All clients run IPv4 and TCP CUBIC.

For each scenario, we perform multiple simulation runs with different independently random seeds. During each run, we record the end-to-end packet transmission time, as well as the total application-layer delay for mice traffic nodes. Goodput statistics for bulk nodes were collected at 25 ms intervals. For each experiment, confidence intervals with 95% confidence level are derived by batch means. The first batch is discarded as initial transient.

Fig. 9 and Fig. 10 compare measurement results from the testbed with corresponding performance measures of the ns-3 simulation for both mice and bulk traffic. We observe a mean relative error for mice traffic goodput of 7%, and a mean relative error for bulk traffic goodput of 3%. The slightly higher deviation observed for mice traffic arises from its shorter transmissions with greater sensitivity to contention and timing variations in real wireless environments.

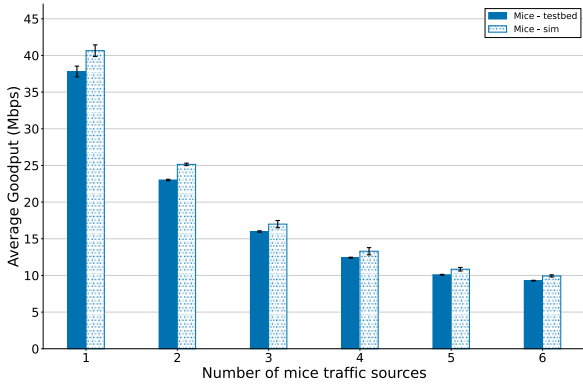


Figure 9: Validation of simulation results for Smart Home scenario: Goodput for mice traffic of cross-layer AQM

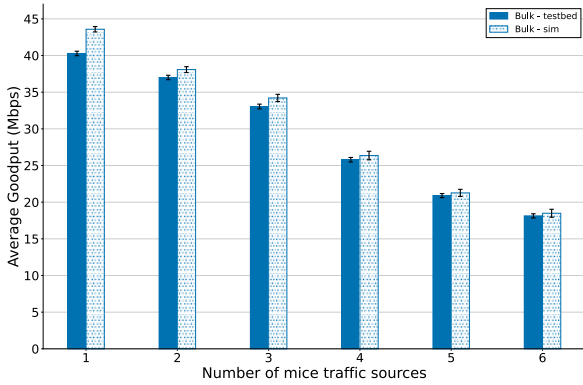


Figure 10: Validation of simulation results for Smart Home scenario: Goodput for bulk traffic of cross-layer AQM

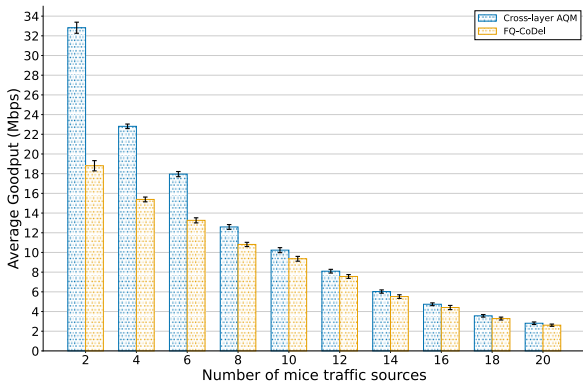


Figure 11: Simulation results for eHealth scenario: Mice goodput of Cross-layer AQM vs FQ-CoDel

C. NS-3 Simulation Study

We investigate the network scalability for increasing mice flows using the ns-3 simulation for both the eHealth and the smart home scenarios. Fig. 11 depicts the steady state goodput results for an increasing number of mice traffic sources in the eHealth scenario. Here, cross-layer AQM consistently achieves higher goodput for mice flows compared to FQ-CoDel.

Fig. 12 depicts the corresponding Jain fairness results. Compared to FQ-CoDel, consistently higher fairness is achieved with cross-layer AQM and the fairness scores converge more quickly, reaching an average fairness score of up to 0.85. Fig. 13 shows results for mice traffic goodput in

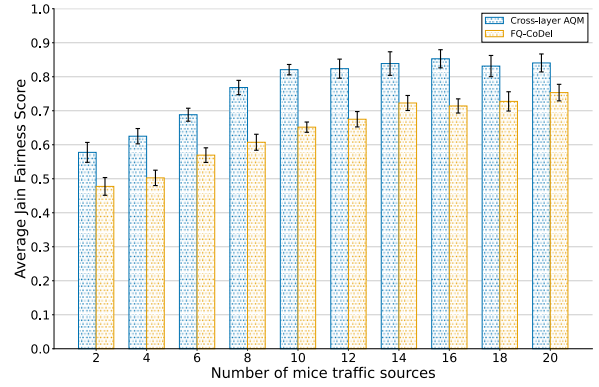


Figure 12: Simulation results for eHealth scenario: Jain fairness of Cross-layer AQM vs FQ-CoDel

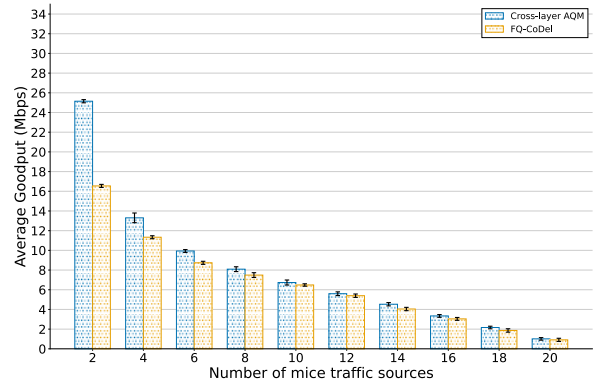


Figure 13: Simulation results for Smart Home scenario: Mice goodput of Cross-layer AQM vs FQ-CoDel

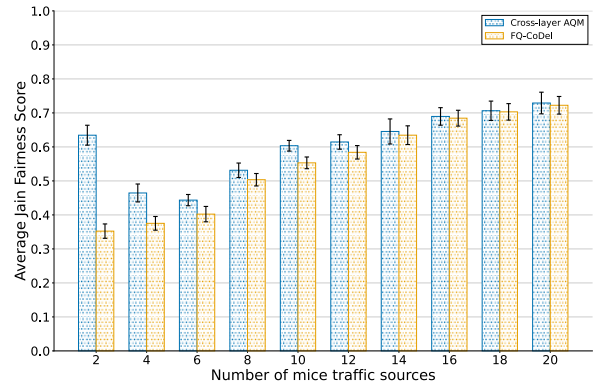


Figure 14: Simulation results for Smart Home scenario: Jain fairness of Cross-layer AQM vs FQ-CoDel

the smart home scenario. In this scenario, convergence to goodput levels comparable to FQ-CoDel occurs much more quickly than in the eHealth scenario. The simulation results for the Jain fairness shown in Fig. 14 indicate that fairness is only higher for up to 12 concurrent mice traffic sources. This indicates that, most performance gains for mice traffic arise from a shift in bandwidth usage away from bulk flows. In summary, the results show that cross-layer AQM provides significant performance improvements for mice flows in the eHealth scenario while maintaining high fairness. In contrast, in the smart-home scenario, performance gains are limited to lower concurrency levels and are achieved at the expense of bulk traffic.

D. Sensitivity of Scaling Factor for Weighted Round Robin

To determine the appropriate weighting between mice and bulk traffic, we conduct a sensitivity analysis for identifying the value of K that yields the best balance between goodput and delay. Table 1 shows goodput and delay values for mice traffic sources in the smart-home scenario.

Table 2 presents the corresponding results for the eHealth scenario. The results presented in Table 1 and Table 2 indicate that setting $K = 3$ as most suitable scaling factor. Thus, this value for the scaling factor was used in all performance experiments presented above.

K	2	2.5	3	3.5	4	5
Goodput (Mbps)	11.4	11.5	11.9	11.8	11.6	11.5
Delay (ms)	18.6	19.0	18.5	19.2	19.4	19.0

Table 1: Sensitivity Analysis of K in Smart Home

K	2	2.5	3	3.5	4	5
Goodput (Mbps)	5.1	5.2	5.6	5.7	5.5	5.4
Delay (ms)	17.0	16.7	15.1	15.7	16.2	16.5

Table 2: Sensitivity Analysis of K in eHealth

VI. CONCLUSIONS

We proposed cross-layer AQM, as a novel scheme for an integral management of the traffic control buffer of IEEE 802.11ax routers. The main idea of cross-layer AQM lies in distinguishing between mice traffic and bulk traffic in the traffic control buffer. We showed that cross-layer AQM limits the impact of the TCP soaking effect and prevents bufferbloat, hence, significantly improves the performance of short-lived TCP and QUIC flows.

We presented a comparative performance study for illustrating the performance improvements of the cross-layer AQM versus the current state-of-the-art queue management algorithm, FQ-CoDel. This performance study shows that for mice traffic with cross-layer AQM achieves up to almost 80% higher goodput than with FQ-CoDel. Furthermore, cross-layer AQM reduces the flow completion time for mice traffic by 10%.

In future work, we would like to explore the performance and fairness of cross-layer AQM under more diverse conditions, with particular emphasis on scalability in IEEE 802.11be networks. Furthermore, we would like to explore how machine-learning can be utilized in active queue management for IEEE 802.11be routers to yield further performance improvements for short-lived TCP and QUIC flows competing with bulk traffic and other traffic types.

REFERENCES

- [1] O. A. Adeleke, N. Bastin, and D. Gurkan, "Network Traffic Generation: A Survey and Methodology", *ACM Comput. Surv.*, 55, pp. 1-23, 2022.
- [2] B. Bellalta, "IEEE 802.11ax: High Efficiency WLANs." *IEEE Wireless Comm.*, 23, pp. 38-46, 2016.
- [3] CAIDA, "Anonymized Two-Way Traffic Packet Header Traces" CAIDA, 2024, https://www.caida.org/catalog/datasets/passive_100g_dataset
- [4] S. Floyd, and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Trans. on Networking*, 1, pp. 397-413, 1993.
- [5] J. Gettys, and K. Nichols, "Bufferbloat: Dark Buffers in the Internet", *Comm. ACM*, 55, pp. 57-65, 2012.
- [6] C. A. Grazia, N. Patriciello, T. Høiland-Jørgensen, M. Klapez and M. Casoni, "Aggregating Without Bloating: Hard Times for TCP on Wi-Fi", *IEEE/ACM Trans. on Networking*, 30, pp. 2359-2373, 2022.
- [7] T. Hoeiland-Joergensen, P. McKeeney, D. That, J. Gettys, and E. Dumazet, "The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm", *IETF, RFC 8290*, Jan. 2018, <https://www.rfc-editor.org/rfc/rfc8290.html>
- [8] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A Tutorial on IEEE 802.11ax High Efficiency WLANs", *IEEE Comm. Surveys & Tutorials*, 21, pp. 197-216, 2019.
- [9] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD dartmouth/campus", *IEEE Dataport*, 2022, <https://iee-dataport.org/open-access/crawdaddartmouthcampus-v-2004-11-09>
- [10] J. Kua, S. H. Nguyen, G. Armitage, and P. Branch, "Using Active Queue Management to Assist IoT Application Flows in Home Broadband Networks", *IEEE Internet of Things Journal*, 4, pp. 1399-1407, 2017.
- [11] A. Langley et al., "The QUIC Transport Protocol: Design and Internet-Scale Deployment", *Proc. SIGCOMM '17*, pp. 183-196, 2017.
- [12] K. Nichols, and V. Jacobson, "Controlling Queue Delay: A Modern AQM is just one Piece of the Solution to Bufferbloat", *ACM Queue*, 10, pp. 20-34, 2012.
- [13] R. Pan, P. Natarajan, F. Baker, and G. White, "Proportional Integral Controller Enhanced (PIE): A Lightweight Control Scheme to Address the Bufferbloat Problem", *IETF, RFC 8033*, Feb. 2017, <https://www.rfc-editor.org/rfc/rfc8033.html>
- [14] S. R. Pokhrel, H. L. Vu, and A. L. Cricenti, "Adaptive Admission Control for IoT Applications in Home WiFi Networks", *IEEE Trans. on Mobile Computing*, 19, pp. 2731-2742, 2019.
- [15] S. Szott, K. Kosek-Szott, P. Gawłowicz, J. T. Gómez, B. Bellalta, A. Zubow, and F. Dressler, "WiFi Meets ML: A Survey on Improving IEEE 802.11 Performance with Machine Learning", *IEEE Comm. Surveys & Tutorials*, 24, pp. 1843-1893, 2022.
- [16] F. Tang, H. Zhang, L. T. Yang and L. Chen, "Elephant Flow Detection and Load-Balanced Routing with Efficient Sampling and Classification", *IEEE Trans. on Cloud Computing*, 9, pp. 1022- 1036, 2021.
- [17] The NS-3 Network Simulator, <https://www.nsnam.org/>
- [18] The OpenWrt project, <https://openwrt.org/>