Reliability and scalability evaluation with TCP/IP of IEEE802.11ah networks

Amina Šljivo, Dwight Kerkhove, Ingrid Moerman, Eli de Poorter, Jeroen Hoebke
Ghent University - imec
IDLab, Department of Information Technology
Technologiepark-Zwijnaarde 15, 9052 Ghent, Belgium
{amina.sljivo, dwight.kerkhove, ingrid.moerman, eli.depoorter, jeroen.hoebke}@ugent.be

ABSTRACT

In Europe, devices operating in the license-free 868MHz ETSI SRD band must comply with a maximum duty cycle limit of 2.8%, provided that they also support Listen Before Talk (LBT) and Adaptive Frequency Agility (AFA) features. Further, in scenarios where reliable bidirectional traffic is required, such as firmware updates over the air, it is a natural choice to use TCP. In spite of duty cycle limitations and given the fact that IEEE 802.11ah supports high data rates up to 7.8Mbps, TCP-based scenarios become more realistic. Therefore we evaluated the feasibility of bidirectional TCP traffic on top of 802.11ah in two scenarios: a reliable throughput scenario that analyses which configurations have the best attainable throughput, and a high scalability scenario where we push the limits of the number of stations the network can support. In order to accurately model bidirectional traffic, we have expanded the existing ns-3 802.11ah module. Both changes to the module and preliminary performance results are presented.

CCS CONCEPTS

• **Networks** → Network performance evaluation; Sensor networks; Network Protocols; Network management

Keywords

IEEE802.11ah, ns-3, IoT, TCP

1 INTRODUCTION

The IEEE 802.11ah or Wi-Fi HaLow specification is a recently finished addition to the growing family of 802.11 standards. Its primary goal is to significantly extend wireless range by bringing the carrier frequency into the sub-GHz domain, more precisely the 868MHz band for Europe, and to significantly reduce power usage by utilizing a station (STA) grouping mechanism. Stations are divided into 2 types: TIM (Traffic Indication Map) and non-TIM stations. TIM is used for stations which need both downlink and uplink access. Non-TIM STAs are designed to periodically transmit small chunks of data and will negotiate the interval with the Access Point (AP) during association. In this model, stations only have specific time slices between beacon intervals where they are allowed to communicate with the AP. A 13bit Association ID (AID) is assigned to each STA and organizes stations into a hierarchical structure, as illustrated in Figure 1. This hierarchy enables the AP to indicate whether a station has buffered downlink data at the AP on multiple levels. For example, if the AP indicates there is data pending for TIM group 1, all STAs listening will check their own TIM group based on their assigned AID. Only the stations that belong to TIM group 1 will need to continue listening to hear which STA index has data to receive, while all the remaining STAs from other TIM groups can sleep until the next AP announcement. Such AP announcements are sent according to beacon intervals: the first beacon sent contains the Distributed Traffic Indication Map (DTIM). All TIM stations have to listen to the DTIM beacon which indicates which TIM groups can expect data from the AP. The next beacons sent are the beacons for each individual TIM group. Each TIM beacon has a map that indicates which STA indices have data pending at the AP as well, analogous to the DTIM beacon, but now at a lower level in the AID hierarchy. Stations of the corresponding TIM group will listen to their TIM beacons if the DTIM specified there was data to be received, the others will go to sleep for the entire

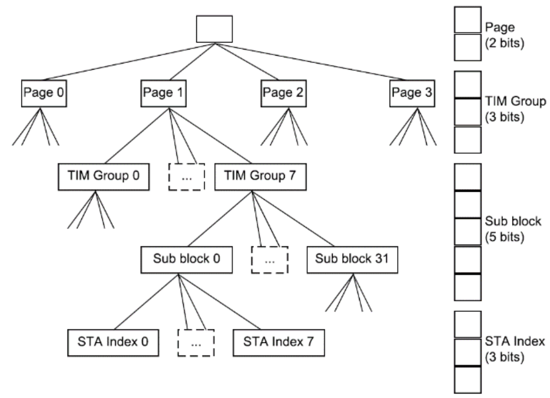

Figure 1: Hierarchical structure of STAs based on AIDs assigned by the AP

DTIM cycle until the next DTIM beacon is expected (unless they have data to transmit). Similarly, stations receiving TIM beacons will immediately know whether to continue listening for data or to re-enter sleep mode as well.

Every group of STAs within the same TIM group only has channel access in its corresponding Restricted Access Window (RAW) period. Raw periods are further divided in slots using a sub-slotting mechanism. The STAs can be distributed evenly over the slots based on their AID to reduce collisions and increase the sleep time, but no RAW grouping algorithm is specified by the draft. If the STAs have data to transmit, they will wait for their corresponding RAW slot to make the transmission. If they sense busy channel during the RAW slot, they will start exponential backoff algorithm within the slot (CSMA/CA[1]) as previously defined in other 802.11 standards. All parameters that stations need to know are either deduced from the association response, containing AID, or are presented in each beacon, namely beacon interval, number of slots, slot duration count etc.

To reduce the overhead of the frame exchange, shorter headers and null data frames are introduced. Frames are sent with the shorter AID instead of the full MAC addresses and frames such as RTS, CTS and ACK are shortened to null data frames.

In this paper we explore the feasibility of bidirectional TCP traffic on top of 802.11ah. For this, we continue the work done on the 802.11.ah ns-3 module by Tian et al. [1], which is implemented as a simplified version of the described model.

2 NS-3 IMPLEMENTATION

2.1 Simplified model

In the module implemented by Tian et al., the AP assigns AIDs to all stations in an incremental fashion. The hierarchy is flattened to just ranges of AIDs. The stations are divided over the slots within the RAW period with modulo operation on their AIDs. TIM beacons contain both start and end AID values for the TIM group, so the stations can determine if the beacon is relevant to them. Shortened frames are not implemented and all frames still use full MAC addresses. Only RAW periods are defined within beacon intervals and stations have unrestricted access. They can both send and receive frames within appropriate RAW slot. The module has previously been used to evaluate throughput and delay with unidirectional UDP traffic from the STAs to the AP, restricting the channel access for uplink communication to the respective TIM group full RAW period. This, however, did not restrict when the stations could receive data. DTIM beacons were also ignored and the radio was never put to sleep because it did not affect the throughput.

2.2 Enhancements for bidirectional traffic

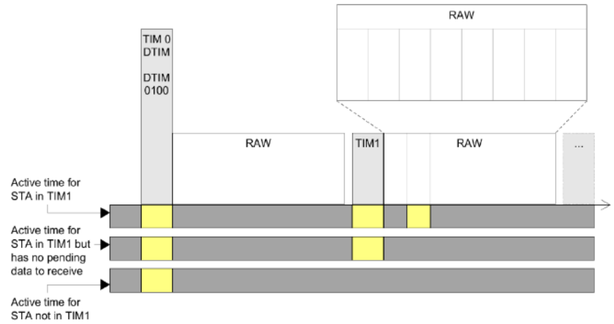

Figure 2: Timeline between beacon intervals. Almost all time between beacons is used as RAW period.

In order to evaluate TCP traffic, we have addressed the issues mentioned in section 2.1. We have implemented DTIM beaconing by sending a 32bit bitmap which indicates which TIM groups have data pending at the AP. Based on the beacon content, STAs can calculate if and how long they can sleep. In case they can enter sleep mode, the radio is turned off at the physical layer and all the frames sent or received during that time are dropped. STAs only wake for DTIM beacons, their TIM group beacon and their assigned slot in the RAW period (if pending data, both uplink and downlink) as shown in Figure 2. The TIM beacon period and RAW slot period are configured to be equivalent.

Next, we have added the transmission scheduling for frames going from the AP to the stations. These packets must be sent during the RAW slot when the STA is actively listening, otherwise it will be dropped.

We have also implemented the possibility for an AP to send an immediate reply: when the AP has data to transmit – usually to reply - it can check whether the reply can still fit within the current RAW slot window. If it does not fit, the AP must schedule the transmission for the slot after the next DTIM beacon. If it fits and the AP deduced the STA is actively listening, it can transmit the frame immediately, significantly reducing round trip time.

Transmissions at the end of a RAW slot also pose problems. While the radio will only go to sleep once the transmission is complete, it will never receive ACK frame that the AP responds with, and will have to retransmit anyway when the next channel access is granted. To prevent transmissions exceeding RAW slot, the DCA queue is also notified about the RAW slot duration. Before starting the transmission of a frame, the STA will now first ensure that the transmission will not cross the boundary of the RAW slot based on the calculated transmission time and remaining RAW slot duration.

TCP is designed to avoid network congestion and treats all losses as such, even if the cause is due to network propagation delay. This packet losses cause the congestion window size to be reset on each drop, resulting in the accumulation of retransmissions to a level that is too large for the network to handle, causing further network degradation and reduction in throughput. There are TCP implementations, such as TCP Peach [2] of TCP Noordwijk

[1] CSMA/CA – Carrier Sense Multiple Access/Collision Avoidance

[3], which can gracefully handle high latency networks and which would be useful to deploy in 802.11ah environment. Sadly, today the ns-3 simulator does not provide these implementations yet. Instead we used TCP Westwood [4], which is suitable for lossy wireless connections.

3 EXPERIMENTS

Most of ns-3's visualization tools only track the final result once the simulation has been finished. To prevent network congestion, TCP adjusts retransmission timeout and congestion window parameters based on RTT estimations. In order to analyze the evolution of these parameters, we have created snapshots of multiple metrics that are taken each second of simulation time. These are then sent to our custom coded visualizer where we can inspect the evolution of each metric per station and see the overall distribution of these values for all stations in real time during the simulation.

We have tested two scenarios: the first one represents IP camera traffic where consistent reliable throughput is necessary, whereas the second scenario tests the scalability of a sensor network that sends measurements in specific intervals. We have used a log propagation loss model for outdoors scenario. We have run over 20000 simulations, per scenario, distributed over 30 servers[2].

3.1 Reliable high throughput

In this scenario STAs implement an IP motion camera that can be configured with the motion probability, duration to record if motion was detected and the data rate of the resulting stream to transmit. For performance testing, we varied data modes, amount of contention per RAW slot, TCP segment sizes, number of TIM groups and number of RAW slots.

To measure the maximum attainable bit rate we set the data rates of the IP cameras to 2, 4, 8, 16, 32, 48, 64, 96 and 128 kbit/s and measured both the sending rate at the STA and receiving rate at the AP.

Results have shown that the coding scheme and TCP segment size are crucial for sustaining higher data rates. A higher MCS will transmit frames much quicker, reducing the drops due to the radio already transmitting/receiving. Bigger TCP segment sizes make better use of the available channel bandwidth before waiting for TCP ACKs, which can clearly be seen in the rise in possible data rate (as shown in Table 1 and Table 2). Adding contention in RAW slots reduces maximum attainable data rate by more than half due to exponential backoff algorithm which on average halves the available RAW slot time when the STA senses the medium to be busy. Halving or doubling a number of TIM groups also doubles or halves the maximum attainable data rate. However, increasing the number of TIM groups has a serious impact on latency: doubling the number of TIM groups at least doubles the RTT[3] which can pose

Table 1: Maximum attainable data rates (kbit/s) of IP camera streams with 4 TIM groups, 5 RAW slots and no contention (20 STAs)

	TCP Segment Size					
MSC	536	1072	1608	2144	2680	3216
MCS0	10.9	12.7	N/A	N/A	N/A	N/A
MCS4	52.4	69.5	80.4	89.0	75.1	81.4
MCS8	67.4	107.8	129.3	135.6	140.4	160.0

Table 2: Maximum attainable data rates (kbit/s) of IP camera streams with 4 TIM groups, 5 RAW slots and contention with 1 station (40 STAs)

	TCP Segment Size					
MSC	536	1072	1608	2144	2680	3216
MCS0	4.9	N/A	N/A	N/A	N/A	N/A
MCS4	24.7	33.2	34.5	38.5	34.1	36.8
MCS8	31.6	51.2	61.3	67.3	61.9	64.1

problems for TCP retransmissions. In case the streams have to be reliable but are generally unidirectional, this is a better choice than increasing the contention inside a RAW slot. Varying the number of RAW slots has a similar effect as changing the contention, both change the channel access time available to stations, leading to comparable results.

3.2 High scalability

One of the main features of 802.11ah is the ability to have a large number of STAs associated with the AP. In this scenario we analyzed how many STAs can be supported by a single AP over an area up to 500m with MCS0 at 2MHz bandwidth. Each station acts as a sensor and periodically sends its measurements to a TCP server hosted on the AP. Each measurement has a payload of 100 bytes,

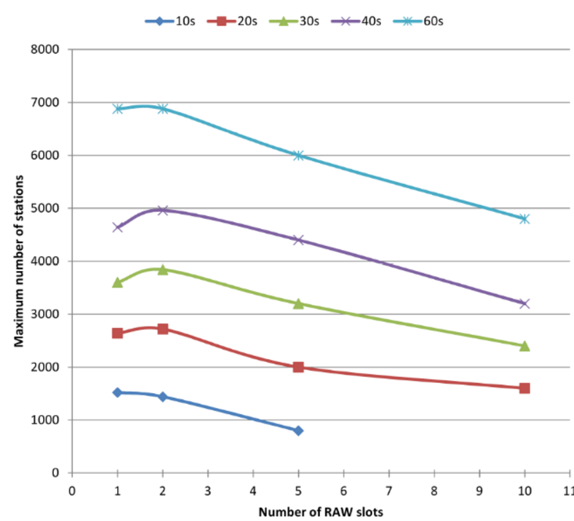

Figure 3: Maximum number of stations for 16 TIM groups

[2] Detailed information and hardware specifications of used testbed is available on http://ilabt.iminds.be/iminds-virtualwall-overview

[3] If the frame was queued in an empty EDCA queue during the RAW slot of the station and the AP has to schedule the reply in the next DTIM cycle.

corresponding to 190 bytes on the wire in total. The payload size is also increased up to 400 bytes to evaluate the effect of grouping multiple measurements together and sending them at slower rate. Each STA starts randomly within the measurement interval to prevent bursts of traffic and from the interval boundary to simulate clock drift.

To determine the upper limit of possible number of STAs, we gradually increased contention per RAW slot to increase the amount of traffic. If the amount of traffic is to too large, the EDCA queues will grow until they reach their maximum size and will start dropping frames.

Due to lower transmission size, we chose a TCP segment size of 536. Smaller segment size enables evaluation of scenario with 10 RAW slots, leaving only $10220\mu s$ of channel access time each DTIM cycle per STA.

With a $10s$ traffic interval, around 1500 stations are possible in various configurations. Decreasing the traffic interval to $20s$ allows up to 2800 stations. A 30, 40 and 60 second interval result in 3800, 4900 and 6900 station limit on average. Figure 3 shows how increasing the number of RAW slots reduces the available channel time and thus the maximum number of stations. Opting to increase the TIM groups instead improves the maximum contention possible but also increases the send and receive time at the AP considerably. Non-time-critical sensors could tolerate this, but some TCP tuning should be done to address the longer duration in the calculation of the retransmission timeout.

Increasing the number of TIM groups significantly reduces the active time, increasing the number of RAW slots has a comparable effect. With 16 TIM groups, 5 RAW slots and a traffic interval of 10 seconds all 800 stations had their radio turned off 97% of the total simulation time.

Increasing the packet size along with the transmission interval results in the same overall reception rate at the server, but maximum possible contention rises considerably. Larger packets make better use of the available channel time: it reduces the overhead, reduces the amount of TCP ACKs and rarer, though longer, transmissions reduce collisions. Compared to the 100 bytes, 10 second traffic interval, doubling the payload and interval can reach a 140-150% higher contention in all RAW slot configurations. However, packets exceeding TCP segment size would cause more traffic due to fragmentation.

To maximize the efficiency of a sensor network the best approach would thus be:

1. Determine the number of stations there will be.
2. Choose as high as possible, but acceptable transmission interval. If possible, delay transmissions by grouping multiple measurements together into a bigger packet.
3. In the chosen interval, maximize the number of TIM groups that the limit will allow.
4. In the chosen interval and TIM group, maximize the number of RAW slots that the limit will allow.

4 CONCLUSIONS AND FUTURE WORK

In this paper we described our contribution to the already existing implementation of the IEEE 802.11ah in ns3. We evaluated the reliability of the network for a continuous high throughput scenario as well as the scalability for a monitoring scenario as a function of the number of TIM groups, RAW slots and packet size.

We presented a selection of obtained results regarding both evaluated scenarios and we proceed with further strategies for additional performance improvements of the 802.11ah network by reducing message size. To this end, we added IPv6 and 6LoWPAN as an option to the ns-3 802.11ah module with the aim to implement the CoAP protocol, designed for constrained devices and networks on top. CoAP on top of UDP and IPv6 with 6LowPAN will result in a vast decrease in message size. We also envision to use CoAP for network management, since constrained devices need to be managed in automatic fashion to handle the large quantity of devices that are expected in future installations. The messages between devices need to be as small and infrequent as possible. As part of this future work, we will evaluate the 802.11ah network performance by using protocol stack shown in Figure 4 in similar fashion as for bidirectional TCP traffic as elaborated in this paper.

Application	CoAP
Transport	UDP
Network	IPv6 with 6LoWPAN
Data Link	IEEE802.11ah
Physical	IEEE802.11ah

Figure 4: Protocol stack to be implemented

6 REFERENCES

[1] Tian, L., Deronne, S., Latré, S., 2016. Evaluation of the IEEE 802.11ah Restricted Access Window Mechanism for dense IoT networks. In *Proceedings of Seventeenth International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE.

[2] Akyildiz, I. F., Morabito, G. and Palazzo, S., 2001. TCP-Peach: a new congestion control scheme for satellite IP networks. IEEE/ACM Transactions on Networking (ToN), 9(3), pp.307-321.

[3] Roseti, C. and Kristiansen, E., 2008, June. TCP Noordwijk: TCP-based transport optimized for web traffic in satellite networks. In *Proceedings of the 26th International Communications Satellite Systems Conference (ICSSC'08).*

[4] Gangadhar, S., Nguyen, T.A.N., Umapathi, G. and Sterbenz, J.P., 2013, March. TCP Westwood (+) protocol implementation in ns-3. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (pp. 167-175).* ICST.