Internet of Things and Cyber-Physical Systems 3 (2023) 37–44

Contents lists available at ScienceDirect

Internet of Things and Cyber-Physical Systems

journal homepage: www.keaipublishing.com/en/journals/internet-of-things-and-cyber-physical-systems

RTP transport in IoT MQTT topologies

Rolando Herrero

College of Engineering, Northeastern University, Boston, MA, USA

ARTICLE INFO

Keywords:
MQTT
6LoWPAN
AMR-WB
RTP
IEEE 802.15.4

ABSTRACT

Media transmission in the context of constrained IoT devices is of critical importance to support several solutions that enable interaction with the physical environment. This includes solutions that range from scenarios of occupancy estimation in building automation to traditional audio and image processing in remote sensing. Many modern access side IoT networks follow *Event Driven Architecture* (EDA) topologies that rely on brokers to forward messages between endpoints. The *Message Queuing Telemetry Transport* (MQTT) is one such protocol that can be used to encapsulate well known *Real Time Protocol* (RTP) audio and media packets. Unfortunately, MQTT relies on TCP transport that is not well suited in IoT constrained environments. This paper introduces a scheme that leverages some of the MQTT features to enable the reliable transmission of media in the context of *Lossy Low Power Networks* (LLNs). Specifically, this scheme is presented, analyzed, optimized and compared against other state-of-the-art mechanisms.

1. Introduction

EDA topologies provide an alternative to traditional *REpresentational State Transfer* (REST) scenarios by natively enabling asynchronous communication between devices and applications [1].

Consider the EDA topology shown in Fig. 1. An access side device transmits sensor readouts to an application on the network core. The topology includes an additional component, a broker, that forwards traffic from the device to the application and from the application to the device. Note that, when compared to REST mechanisms, the EDA broker increases the end-to-end latency as two endpoints are separated by two hops as opposed to one. The broker, however, improves reliability but supporting redundancy and buffering among many other features. The scenario is purely wireless with an IEEE 802.15.4 access side and an IEEE 802.11 core side. The *Message Queue Telemetry Transport* (MQTT) protocol is the most popular of all EDA mechanisms and has become the default standard for this type of deployments [2]. MQTT is intended to support end-to-end IoT connectivity and is, therefore, transmitted over the *Transport Control Protocol* (TCP) to enable firewall traversal. Applications subscribe to topics (i.e. temperature) associated with specific events (i.e. temperature readouts) and devices publish the events associated with these topics. Because of this, MQTT and EDA protocols in general are known as publish/subscribe protocols. TCP transport guarantees reliable data transmission such that MQTT messages arrive at the destination uncorrupted and in order.

In the context of this paper, the focus is on IoT devices that generate audio streams. In REST architectures, audio is typically transmitted over *Real Time Protocol* (RTP) media sessions. Because RTP was conceived in the 1990s to support *Voice over IP* (VoIP) applications on constrained handheld Internet phones, is highly optimized and a really good fit for IoT scenarios. Typically, out-of-band protocols like the *Session Initialization Protocol* are used to negotiate RTP sessions (SIP) [3]. Many various adaptations have been proposed to support SIP in the context of IoT because SIP was not designed for constrained devices [4]. However, the focus of this paper is on RTP traffic, and it is assumed that every session is started by some out of band method. Unfortunately, RTP does not comply with the EDA model very well. For starters, RTP is a one-hop protocol that assumes direct connectivity between devices and applications. Moreover, RTP relies on *User Datagram Protocol* (UDP) transport. Fig. 2 shows a traditional REST-based RTP stack and our proposed EDA-adapted version. Both stacks rely on the same low level layers. Specifically, IEEE 802.15.4 is used to implement the physical and link layers [5]. With frames no bigger than 127 bytes, this well-known low power IoT technology achieves nominal transmission rates of 250 Kbps. The transmission takes place over *Instrumental Scientific and Medical* (ISM) bands that are not licensed. IPv6 datagrams can be encapsulated over IEEE 802.15.4 if a 6LoWPAN layer is available as a middle layer. The numerous important features introduced by 6LoWPAN include (1) header compression of upper-layer protocols like IPv6 and UDP, (2) mesh forwarding to increase connectivity's physical range, (3) multicasting and (4) fragmentation to accommodate MTU sizes as small as those of IEEE 802.15.4 [6]. For the traditional REST stack, RTP frames are transmitted

E-mail address: r.herrero@northeastern.edu.

https://doi.org/10.1016/j.iotcps.2023.02.001
Received 16 January 2023; Received in revised form 4 February 2023; Accepted 5 February 2023
Available online 10 February 2023

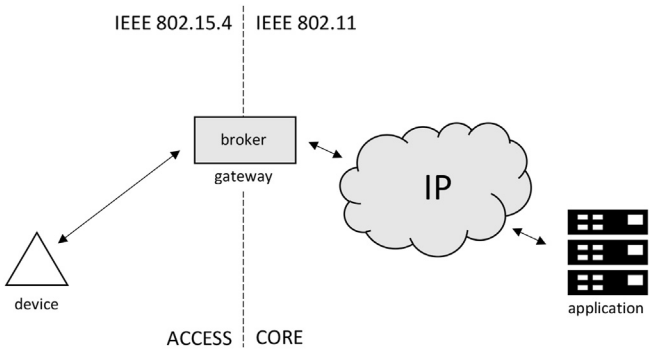


Fig. 1. EDA topology.

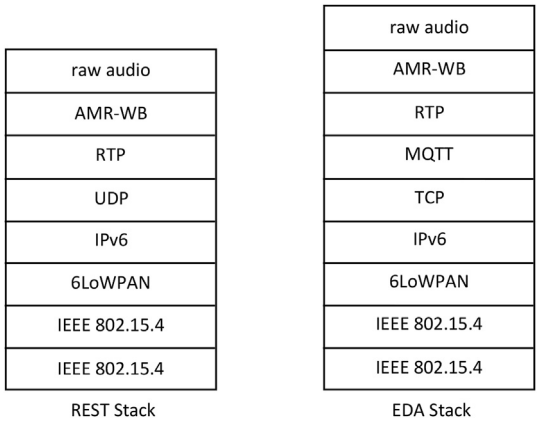


Fig. 2. RTP stack over EDA

directly over UDP segments, while for the proposed EDA mechanism, RTP frames are transmitted over MQTT messages. These MQTT messages are, in turn, carried by TCP segments. The *Adaptive Multi Rate Wideband* (AMR-WB) speech codec is used to compress the audio frames that are carried in RTP packets [7]. Widely utilized in mobile networks, AMR-WB is a well-known linear prediction-based codec. It should be noted that despite the fact that AMR-WB was created to compress speech, it enables high transmission rate modes that also support generic audio compression. The actual frame size can be quite large, depending on the sampling rates and transmission mode set on the codec. RTP packets, that are transmitted while the session is established, are typically protected by a security mechanism that enables *Secure RTP* (SRTP). In the context of the topology presented in this paper, security is provided by MQTT. Specifically, and although the protocol stack in Fig. 2 does not include a security layer, MQTT relies on a standard *Transport Layer Security* (TLS) layer between the TCP and the MQTT layers.

AMR-WB frames are encapsulated over RTP as indicated in Fig. 3. RTP headers, in turn, are encapsulated over MQTT. The MQTT header starts by a 4-bit type field that is set to PUBLISH, a 2-bit QoS field that indicates *fire-and-forget* (no retransmissions), an 8-bit message length field and a variable length topic string that is used to identify the event type. Similar to this, the RTP header starts with a 2-bit version field and then has a padding bit to determine whether or not the payload is extended with more padding bytes. The padding's size is indicated by its final byte. An extension bit, a 4-bit *Contributing Source* (CSRC) count field, a marker bit, a 7-bit payload type field identifying the codec being used, a 16-bit sequence number, and a 32-bit timestamp used for synchronization and jitter estimations are among the other fields included in the header. A list of 32-bit CSRCs for the sources that contributed to the packet is optionally included in the headers, along with a 32-bit *Synchronization Source* (SSRC) number that identifies the stream. As described in RFC 4867, "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs" [8] packetization headers for AMR-WB frames are prepended. These headers support multiple frames per packet and describe how speech frames are packed over RTP. Note that RTP packets are normally transported over UDP to minimize end-to-end latency and avoid retransmissions. In other words, packets that are lost are not retransmitted. Under MQTT, QoS 0 (fire-and-forget) provides the closest match to this behavior. MQTT QoS level 0, however, is not exactly like UDP as it relies on TCP transport and it is, therefore, subject to congestion. This means that for severe network layer losses, the RTP packets transmitted over MQTT may be affected by increased latency due to the TCP retransmissions.

Fig. 4 shows the structure of the 6LoWPAN datagrams that are transmitted on the IEEE 802.15.4 based access side of the scheme. It relies on 6LoWPAN *IP Header Compression* (IPHC) that enables the transmission of AMR-WB media frames encapsulated over RTP, MQTT, TCP and compressed IPv6. Similarly, Fig. 5 shows the structure of the IPv6 datagrams that are transmitted on the IEEE 802.11 based core side of the scheme. It supports the transmission of AMR-WB media frames encapsulated over RTP, MQTT, TCP and IPv6. The big different between both scenarios is the fact that 6LoWPAN *IP Header Compression* (IPHC) compresses IPv6 headers by around 12 times when compared to uncompressed IPv6.

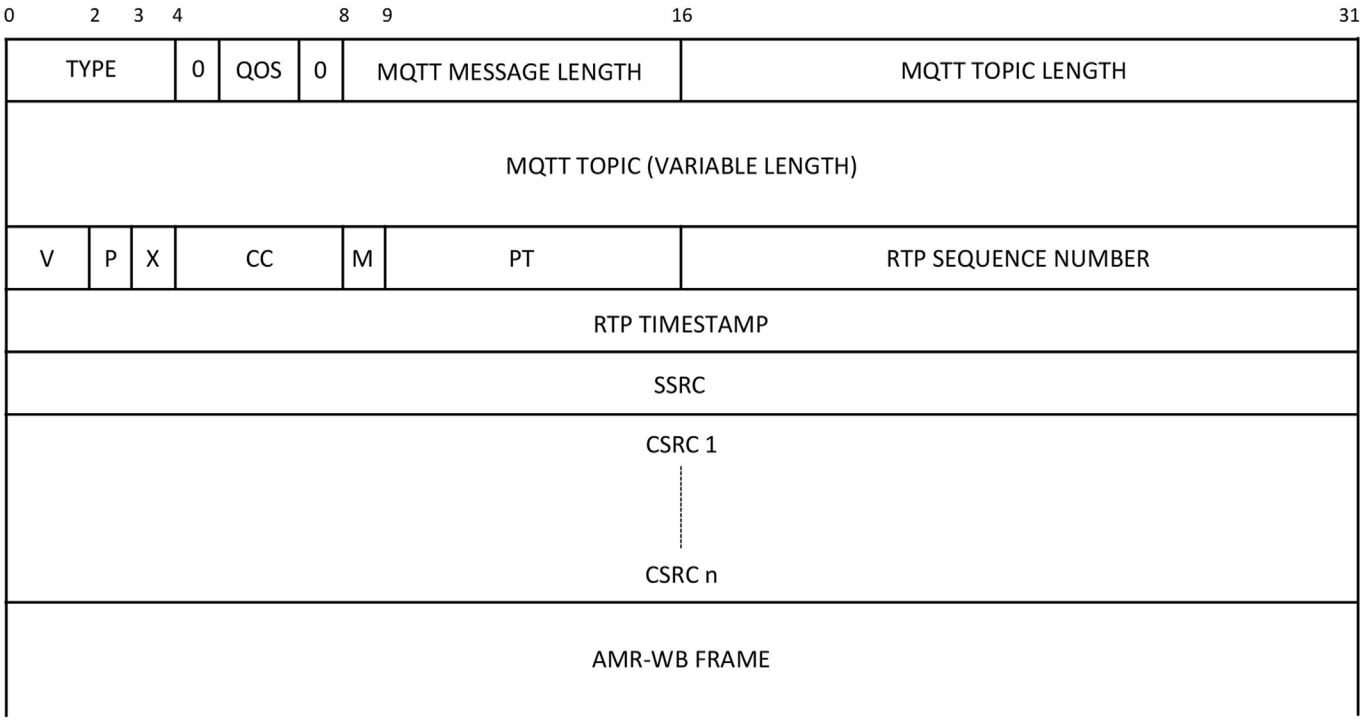
MQTT traffic, by relying on firewall-enabled TCP transport, is subject to *Head-of-Line* (HOL) blocking that can lead to application layer packet loss as TCP retransmissions delay the arrival of MQTT messages [9]. Essentially, MQTT messages that arrive too late, are lost from the perspective of the application layer. Moreover, due to the stream nature of TCP transport, one a message arrives late, all subsequent messages may arrive late due to congestion. This paper introduces a mechanism that splits the transmission of RTP traffic over multiple MQTT sessions (and TCP connections) to minimize the effect of HOL blocking. It relies on an algorithm that senses network layer impairments to dynamically estimate the number of simultaneous MQTT sessions that can lead to a specific application layer loss goal.

The remaining Sections of the paper are structured as follows: In Section 2, a review of the literature is provided. In Section 3, specifics about the solution, the mathematical model, and the suggested algorithm are described. In Section 11, a framework for conducting experiments to assess the effectiveness of the solution is presented. Section 5 offers conclusions and suggestions for future work.

2. Motivation and literature review

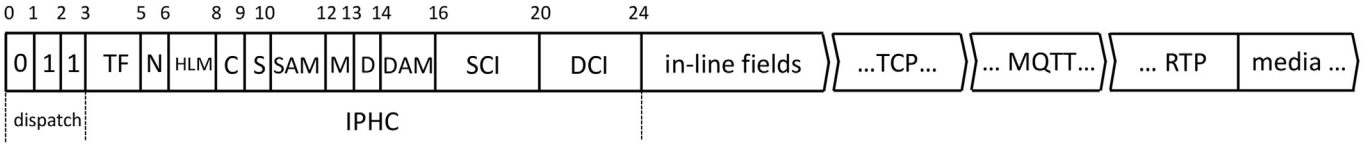
The study of the reliable transmission of MQTT traffic over IEEE 802.15.4 is a topic of great interest as MQTT is the industry standard that enables the transmission of sensor data to the core side of the IoT networks. In the context of constrained devices, MQTT is not a popular access side protocol as, traditionally, IEEE 802.15.4 has relied on other session layer mechanisms like the *Constrained Application Protocol* (CoAP) to enable access side connectivity [10]. In all, hybrid scenarios where CoAP provides session management from the device to the edge and MQTT does the same from the edge to the cloud are common. The end-to-end MQTT topology discussed in this paper removes most of the *Application Layer Gateway* (ALG) functionality overhead that is carried out at the edge. Specifically, all CoAP-to-MQTT and the MQTT-to-CoAP session adaptation is replaced by a scenario where MQTT is present at both the device and the cloud. The fact that devices are becoming less constrained, as a result of the latest developments in low power processor manufacturing, is helping enable the support of end-to-end MQTT topologies.

Support of end-to-end MQTT transported media in the context of IEEE 802.15.4 based access is an area of limited research. Most of the focus has been around the transmission of non-real time sensor readouts and, in particular, on the deployment of applications and use cases that rely on



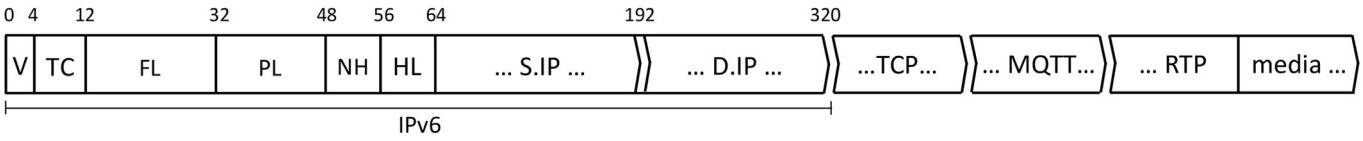
TYPE MQTT type (PUBLISH)
QOS MQTT QoS Level 0 (fire-and-forget)
V RTP version
P RTP padding
X RTP extension
CC RTP CSRC count (n)
M RTP marker
PT RTP payload type
SSRC RTP synchronization source
CSRC RTP contributing sources

Fig. 3. AMR-WB encapsulation.



TF traffic class and flow label encoding
N = 0 NHC doesn't follow
HLM hop limit encoding
C include context information fields
S context based source address encoding
SAM source address mode
M multicast destination address
D context based destination address encoding
DAM destination address mode
SCI source context identifier (if C = 1)
DCI destination context identifier (if C = 1)

Fig. 4. Access side encapsulation.



V version
TC traffic class
FL flow label
PL payload length
NH next header
HL hop limit
S.IP source IP address
D.IP destination IP address

Fig. 5. Core side encapsulation.

these scenarios. In Ref. [11] the authors introduce an indoor localization mechanism that relies on MQTT over IEEE 802.15.4. Similarly, in Refs. [12,13], a traffic light control solution built around MQTT, 6LoWPAN and IEEE 802.15.4 is presented. An air monitoring station that involves an MQTT over IEEE 802.15.4 stack is introduced in Ref. [14]. In Ref. [15], MQTT and 6LoWPAN are combined to support home automation scenarios. IPv6 adaptation to support the transmission of media and other more general traffic in the context of IoT has been a widely explored topic of research. This involves many physical layer technologies including IEEE 802.15.4 [16–18], NB-IoT [19], BLE [20,21] and generic ultrasound [22]. In the context of media over MQTT, the authors, in Ref. [23], analyze the effect of impairments and throughput on audio. Similarly, in Ref. [24], the performance of MQTT against other session layer protocols is compared. In Ref. [25] the authors introduce a mechanism that propagates speech by means of MQTT in an end-to-end solution that integrates embedded devices with cloud based Natural language processing.

This paper has some advantages with respect to these previous work as it addresses some missing areas. Specifically, none of the aforementioned papers address the reliable support of end-to-end MQTT to enable media transmission. More importantly, none of them introduce a model that can be used to predict traffic behavior and dynamically configure transport parameters to improve application layer quality scores.

3. Reliable EDA mechanism

In order to enhance the reliability when transmitting media stream, we introduce a *Forward Error Correction* (FEC) mechanism that sends multiple copies of each RTP frame over a set of separate MQTT sessions. The number of sessions in each set is dynamically calculated based on the estimation of network impairments. This section introduces a mathematical model that is the building block of an algorithm that enables this computation.

Fig. 6 shows the flow of RTP frames between a device and an application where n RTP frames are transmitted over k MQTT sessions. There is a trade-off between reliability and the transmission rate. Note that more advanced mechanisms of redundancy can be used applied to this scheme. For instance, the overall performance of the scheme could be enhanced by using FEC block codes rather than transmitting frame copies. The analysis and use of these mechanisms is outside the scope of this paper.

3.1. Theoretical background

The loss is quite bursty and affected by dynamic multi-path fading associated with wireless networks. The Gilbert-Elliot two-state Markov channel model shown in Fig. 7 [26] is used to accurately model this

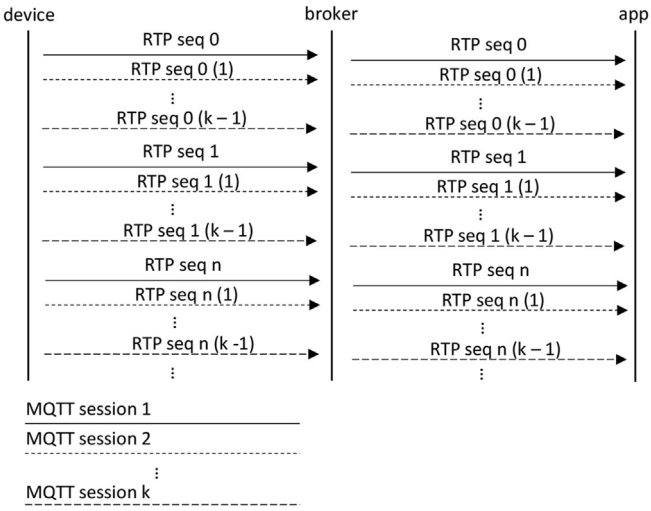


Fig. 6. RTP over MQTT

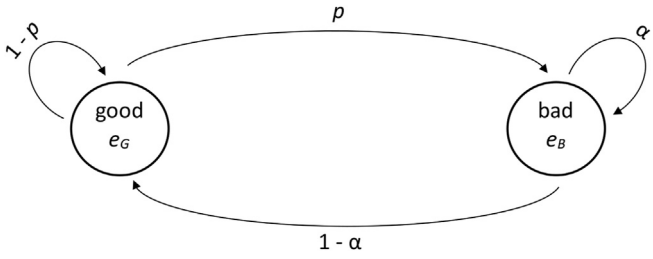


Fig. 7. Gilbert-Elliot channel model.

bursty loss. The model incorporates *em good* and *em bad* states, which are related to *em low* and *em high* packet losses with probabilities e_G and e_B respectively. The model has four parameters: (1) p, the probability of the channel changing from good to bad; (2) *alpha*, the likelihood that the channel will stay in bad condition; (3) e_G, the likelihood of packet loss when the channel is in good condition; and (4) e_B, the likelihood of packet loss when the channel is in bad condition. Network packet loss and loss burstiness are controlled by the parameters p and *alpha*, respectively. The differences between physical layer technologies and topologies can be associated with different values of e_G and e_B. In this context, considering the message flow in Fig. 8, the steady state probabilities of the channel being in the bad and good states, respectively named P_b and P_g, are given by

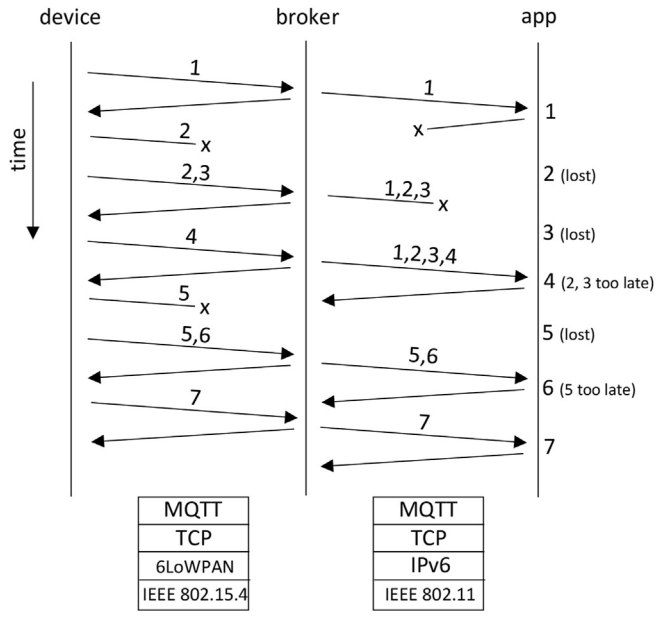


Fig. 8. MQTT message flow.

$$P_{\mathrm{b}} = \frac{p}{1-\alpha+p} \tag{1}$$

and

$$P_{\mathrm{g}} = \frac{1-\alpha}{1-\alpha+p} \tag{2}$$

Due to the chain rule, the probability of successful acknowledgments and transmissions is given by

$$P_{\mathrm{A,T}} = P_{\mathrm{A|T,b}} P_{\mathrm{T|b}} P_{\mathrm{b}} + P_{\mathrm{A|T,g}} P_{\mathrm{T|g}} P_{\mathrm{g}} \tag{3}$$

where $P_{\mathrm{A|T,b}}$ and $P_{\mathrm{A|T,g}}$ are the probabilities of acknowledgment given that a successful transmission has occurred when the channel is in bad and good states respectively and $P_{\mathrm{T|b}} = 1 - e_B$ as well as $P_{\mathrm{T|g}} = 1 - e_G$ are the probabilities of successful transmission for the same bad and good channel states. Note that because acknowledgments are independent from transmissions, the $P_{\mathrm{A|T,b}}$ and $P_{\mathrm{A|T,g}}$ probabilities are respectively given by

$$P_{\mathrm{A|T,b}} = P_{\mathrm{A|b}} P_{\mathrm{b\to b}} + P_{\mathrm{A|g}} P_{\mathrm{b\to g}} \tag{4}$$

and

$$P_{\mathrm{A|T,g}} = P_{\mathrm{A|g}} P_{\mathrm{g\to g}} + P_{\mathrm{A|b}} P_{\mathrm{g\to b}} \tag{5}$$

where $P_{\mathrm{b\to b}} = \alpha$, $P_{\mathrm{b\to g}} = 1 - \alpha$, $P_{\mathrm{g\to b}} = p$ and $P_{\mathrm{g\to g}} = 1 - p$ represent the probabilities of transitioning between the different channel states.

Considering a symmetric channel, there is no difference between transmissions and acknowledgments and, therefore, $P_{\mathrm{A|b}} = P_{\mathrm{T|b}} = 1 - e_B$ and $P_{\mathrm{A|g}} = P_{\mathrm{T|g}} = 1 - e_G$. This leads to the following acknowledgment probabilities given the successful transmission over channels in bad and good states

$$P_{\mathrm{A|T,b}} = (1-e_B)\alpha + (1-e_G)(1-\alpha) \tag{6}$$

and

$$P_{\mathrm{A|T,g}} = (1-e_G)(1-p) + (1-e_B)p \tag{7}$$

that, when replaced in Equation (3), results in the probability of successful acknowledgment and transmission

$$P_{\mathrm{A,T}} = [(1-e_B)\alpha + (1-e_G)(1-\alpha)](1-e_B)\frac{p}{1-\alpha+p} + [(1-e_G)(1-p) + (1-e_B)p](1-e_G)\frac{1-\alpha}{1-\alpha+p} \tag{8}$$

where e_B and e_G, as previously indicated, are dependent on the wireless technology and topology.

The overall packet loss, defined as P_{loss}, is therefore given by

$$P_{\mathrm{loss}} = [1 - P_{\mathrm{A,T,1}} P_{\mathrm{A,T,2}}]^n \tag{9}$$

where $P_{\mathrm{A,T,1}}$ and $P_{\mathrm{A,T,2}}$ respectively represent the aforementioned probability of successful acknowledgment and transmission for IEEE 802.15.4 as well as for IEEE 802.11 and n accounts for the number of simultaneous MQTT sessions.

Figs. 9 and 10 show the application layer message loss probability as a function of the number MQTT sessions and the network layer packet loss probability for low and high packet loss burstiness. It is more than clear, that regardless of the network loss scenario, the application layer message loss decreases dramatically with an increasing number of sessions. Note that for most cases $e_B \approx 1$, $e_G \approx 0$ and therefore

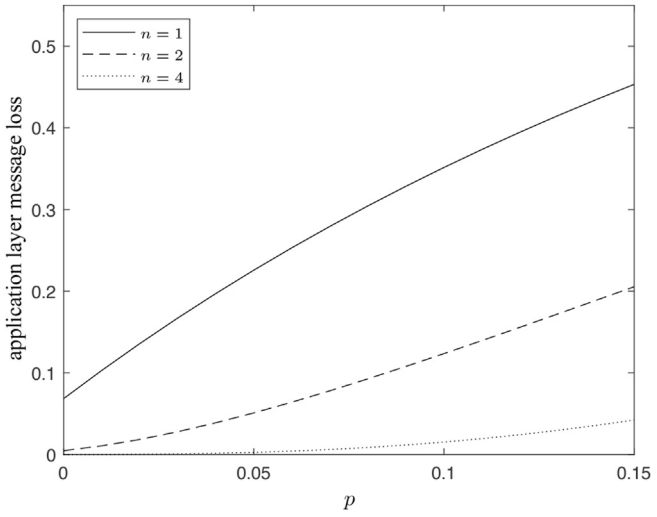


Fig. 9. Application layer message loss (for low loss burstiness).

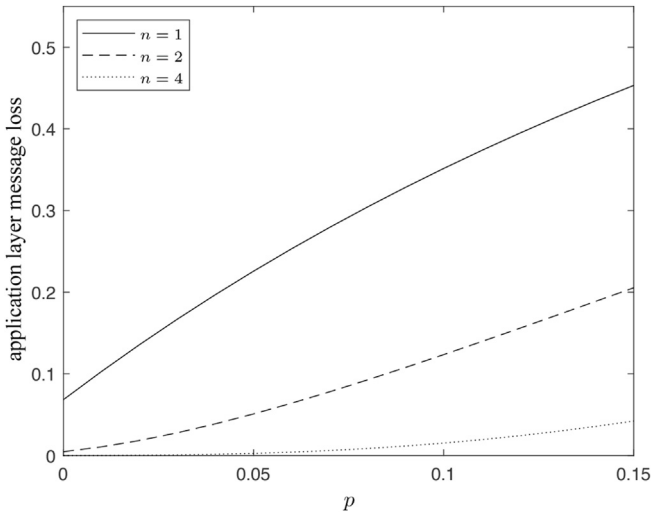


Fig. 10. Application layer message loss (for high loss burstiness).

$$P_{\mathrm{A,T}} \approx \frac{(1-\alpha)(1-p)}{(1+\alpha+p)} \tag{10}$$

and the overall packet loss becomes

$$P_{\mathrm{loss}} = \left[1 - P_{\mathrm{A,T}}^2\right]^n \tag{11}$$

where there is no more need to differentiate between the different physical layers.

3.2. Algorithm

That analytical model introduced in Section 3.1 can be used to design an algorithm that dynamically adjusts the number of MQTT session based on overall network application layer message loss probability (P_{loss}) goals. By measuring network layer impairments that result from TCP metrics, it is possible to estimate the parameters of the Gilbert-Elliot channel model and, in turn, determine their effect on the application layer. In this context, consider the following steps to support the dynamic estimation of the number of MQTT sessions ($\hat{n}$):

1. Set $\hat{n}_{\mathrm{avg}} = 1$
2. Calculate the p and α estimators using *Maximum Likelihood Estimation* (MLE) as

$$\hat{\alpha} = \frac{J_{H,H}}{J_{H,H} + J_{H,L}}$$

and

$$\hat{p} = \frac{J_{L,H}}{J_{L,H} + J_{L,L}}$$

where $J_{H,H}$, $J_{L,H}$, $J_{H,L}$ and $J_{L,L}$ respectively, reflect the transition counts between low and high loss probability states throughout the analysis period. These parameters are the outcome of a TCP segment analysis.

3. Compute $\hat{n}$ as

$$\hat{n} = \max\left[1, \min\left(\log_{1-\hat{P}_{\mathrm{A,T}}^2} P_{\mathrm{loss}}, n_{\max}\right)\right]$$

where $\hat{P}_{\mathrm{A,T}}$ is given by $\hat{P}_{\mathrm{A,T}} = \frac{(1-\hat{\alpha})(1-\hat{p})}{(1+\hat{\alpha}+\hat{p})}$, P_{loss} is the packet loss goal and $n_{\max}$ is maximum number of supported MQTT sessions.

4. Update $\hat{n}_{\mathrm{avg}}$ as

$$\hat{n}_{\mathrm{avg}} = \beta\hat{n}_{\mathrm{avg}} + (1-\beta)\hat{n}$$

where $\beta = 0.875$ controls weight of the instantaneous $\hat{n}$ estimation in the moving average.

5. $\lceil \hat{n}_{\mathrm{avg}} \rceil$ represents the number of MQTT sessions needed to support the transmission of messages.

Algorithm 1. Target Application Selection

Algorithm 1: Target Application Selection

Function `SessionCount`($\hat{n}_{\mathrm{avg}}$, $J_{L,L}$, $J_{H,L}$, $J_{L,H}$, $J_{H,H}$, $n_{\max}$, P_{loss}, β):

$\hat{\alpha} = \frac{J_{H,H}}{J_{H,H}+J_{H,L}}$

$\hat{p} = \frac{J_{L,H}}{J_{L,H}+J_{L,L}}$

$\hat{P}_{\mathrm{A,T}} = \frac{(1-\hat{\alpha})(1-\hat{p})}{(1+\hat{\alpha}+\hat{p})}$

$\hat{n} = \max\left[1, \min\left(\log_{1-\hat{P}_{\mathrm{A,T}}^2} P_{\mathrm{loss}}, n_{\max}\right)\right]$

return $\beta\hat{n}_{\mathrm{avg}} + (1-\beta)\hat{n}$

Note that β can be used to regulate the convergence of the instantaneous samples into the overall average. Algorithm 1 shows the function *SessionCount* that implements the algorithm to determine the number of MQTT sessions.

4. Experimental framework

Fig. 11 shows the experimental framework that is used to validate the efficiency of the algorithm introduced in Section 3.2. The scenario includes an array of N devices that publish RTP packets into an MQTT broker. The broker, in turn, publishes the packets into an application that calculates speech quality scores and application layer network loss. All wireless paths between devices, broker and application are subjected to impairments that follow the Gilbert-Elliot channel model and are controlled by the application.

The devices are set up to support AMR-WB transcoding for speech sampled at 16 KHz at a bitrate of 23.85 Kbps. Each RTP packet contains one AMR-WB speech frame and is sent out every 20 ms. ITU-T Rec. P.863 [27], which specifies a speech quality estimate technique called *Perceptual Objective Listening Quality Assessment* (POLQA) is used to calculate speech quality scores. This process is based on a perceptual model, which is used to extract key factors that both describe the reference and the speech sequence being examined, and which are then compared to produce a quality score between 1 (poor) and 5. (excellent). The listening quality scores that are generated by POLQA, when applied on test speech samples, are known as *Mean Opinion Score Listening Quality Objective* (MOS-LQO). The speech reference in this case is made up of a 300-s sequence that is created by concatenating several ITU P.501 Annex B sample voice files [28].

Four test cases are considered: (1) low loss burstiness with a 1% packet loss goal, (2) high loss burstiness with a 1% packet loss goal, (3) low loss burstiness with a 5% packet loss goal and (4) high loss burstiness with a 5% packet loss goal. Each test cases consists of $N = 10$ devices transmitting the aforementioned 300-s sequence. The algorithm described in Section 3.2 is used to minimize the number of simultaneously active MQTT that the devices use to propagate media traffic (with $n_{\max} = 7$). For each scenario the following average metrics are collected: (1) the number of MQTT sessions, (2) the actual application layer packet loss probability and (3) the corresponding MOS-LQO score. Table 1 shows these three metrics for low ($\alpha = 0.1$) and high ($\alpha = 0.4$) packet loss burstiness and application layer loss probability goals of 1% and 5%.

Figs. 12 and 13 respectively show the number of MQTT sessions per RTP stream for 5% and 1% application layer packet loss goals as a function of time. Specifically, the Figures present a short 30-s snippet of the number of MQTT sessions captured on a random device during each of the test cases. Note that to accomplish the 5% packet loss goal, two and six sessions per RTP stream are typically needed to support low and high network packet loss burstiness respectively. When further restricting the media quality by imposing a 1% application layer packet loss goal, six and seven sessions per RTP stream are required for low and high network

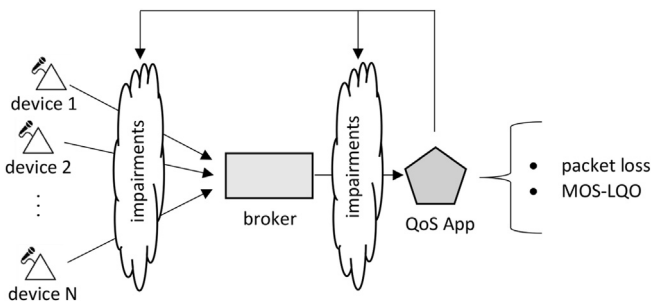


Fig. 11. Experimental framework.

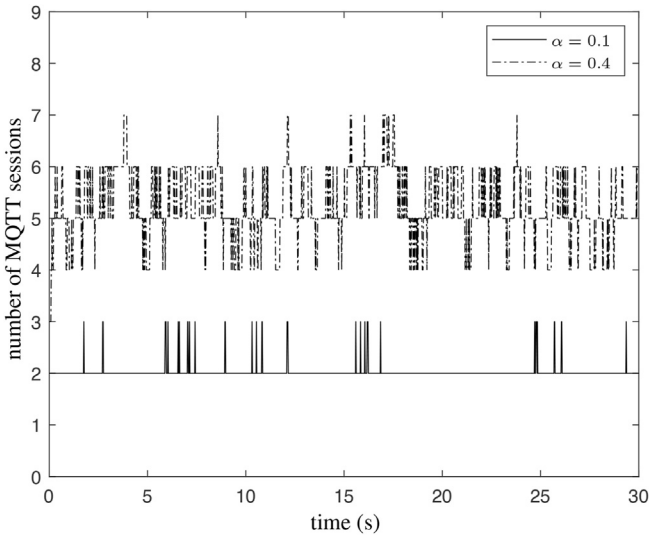


Fig. 12. Number of MQTT session per RTP stream (5% packet loss).

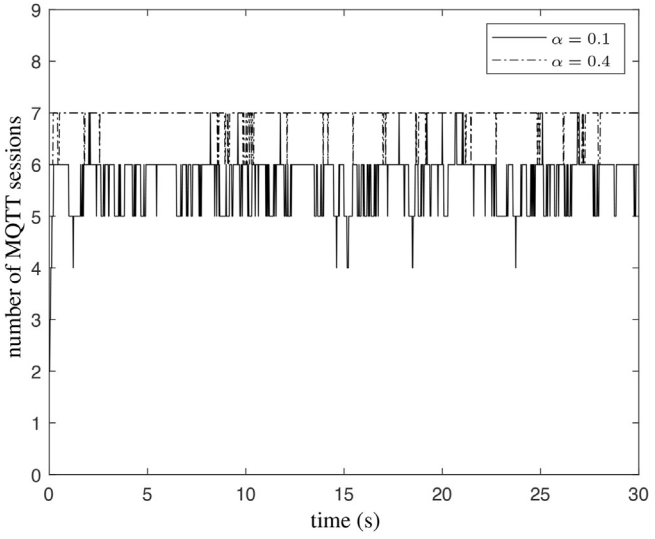


Fig. 13. Number of MQTT session per RTP stream (1% packet loss).

packet loss burstiness respectively.

By comparing the results in Table 1 with the plots in Figs. 12 and 13 a few observations can be drawn:

- The algorithm meets 1% application layer packet loss goals (with a maximum average error of 17%) and it also meets 5% application layer packet loss goals (with a maximum average error of 5%). The algorithm follows goals better when they are less restrictive.
- For the same application packet loss goals, higher network layer packet loss burstiness causes the algorithm to be less accurate.
- The relationship between application packet loss goals and POLQA is that, as expected, lower application packet loss goals lead to higher MOS-LQO scores.

Table 1
Metrics.

loss goal	α	sessions	$\hat{P}_{\text{loss}}$	MOS-LQO
1%	0.1	5.66	1.09%	3.46
	0.4	6.98	1.26%	3.14
5%	0.1	2.01	4.91%	3.02
	0.4	5.17	5.15%	2.86

- Similarly, the relationship between application packet loss goals and the number of MQTT sessions per RTP stream is that, as expected, lower application packet loss goals lead to a larger number of sessions.

All in all, the application layer packet loss goal can be used to indirectly determine the optimal number of MQTT sessions per RTP stream and estimate the overall media quality.

5. Conclusion and future work

This paper introduces a mechanism that lowers the effects of the network layer impairments on the application layer by dynamically assigning multiple EDA sessions to a single media stream. The mechanism relies on an algorithm that monitors the network traffic to estimate the optimal number of MQTT sessions that accomplishes a specific packet loss goal. In this context, it can be seen that more restrictive goals lead to a larger number of sessions per RTP stream as indicated in Section 11. Moreover, these more restrictive goals also result in better POLQA media quality scores. Of course, the trade-off between media quality and available resources must be taken into account since additional MQTT sessions affect computational and networking performance.

There are quite a few topics for future study that are associated with the material presented in this paper. They range from the integration of standard error correction technologies that support more efficient controlled redundancy to the use of the architecture with other session layer protocols like CoAP.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Herrero, Fundamentals of IoT Communication Technologies, Textbooks in Telecommunication Engineering, Springer International Publishing, 2021. URL, https://books.google.com/books?id=k70rzgEACAAJ.
[2] K.B. Andrew Banks, Ed Briggs, R. Gupta, Mqtt version 3.1.1 oasis committee specification, Apr. 2014. URL, http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.
[3] E. Schooler, J. Rosenberg, H. Schulzrinne, A. Johnston, G. Camarillo, J. Peterson, R. Sparks, M.J. Handley, SIP: session initiation protocol, RFC 3261 (Jul. 2002), https://doi.org/10.17487/RFC3261. URL, https://rfc-editor.org/rfc/rfc3261.txt.
[4] S. Cirani, M. Picone, L. Veltri, Cosip: a constrained session initiation protocol for the internet of things, in: C. Canal, M. Villari (Eds.), Advances in Service-Oriented and Cloud Computing - Workshops of ESOCC 2013, Málaga, Spain, September 11-13, 2013, Revised Selected Papers, Vol. 393 of Communications in Computer and Information Science, Springer, 2013, pp. 13–24, https://doi.org/10.1007/978-3-642-45364-9_2, 10.1007/978-3-642-45364-9_2. URL.
[5] Ieee standard for low-rate wireless networks, IEEE Std 802.15.4-2020 (2020) 1–800. Revision of IEEE Std 802.15.4-2015.
[6] G. Montenegro, J. Hui, D. Culler, N. Kushalnagar, Transmission of IPv6 packets over IEEE 802.15.4 networks, RFC 4944 (Sep. 2007), https://doi.org/10.17487/RFC4944. URL, https://rfc-editor.org/rfc/rfc4944.txt.
[7] 3GPP, Ts 26.190 : speech codec speech processing functions; adaptive multi-rate - wideband (amr-wb) speech codec; transcoding functions, in: Tech. Rep., 1, 2008, pp. 1–42. TS 26.190, 3rd Generation Partnership Project.
[8] J. Sjoberg, A. Lakaniemi, M. Westerlund, Q. Xie, RTP payload Format and file storage Format for the adaptive multi-rate (AMR) and adaptive multi-rate wideband (AMR-WB) audio codecs, RFC 4867 (Apr. 2007), https://doi.org/10.17487/RFC4867. URL, https://rfc-editor.org/rfc/rfc4867.txt.
[9] M. Scharf, S. Kiesel, Head-of-line Blocking in tcp and sctp: Analysis and measurements, 2007, pp. 1–5, https://doi.org/10.1109/GLOCOM.2006.333.
[10] Z. Shelby, K. Hartke, C. Bormann, The constrained application protocol (CoAP), RFC 7252 (Jun. 2014), https://doi.org/10.17487/RFC7252. URL, https://rfc-editor.org/rfc/rfc7252.txt.
[11] P. Puspitaningayu, N. Funabiki, Y. Huo, K. Hamazaki, M. Kuribayashi, W.-C. Kao, Application of fingerprint-based indoor localization system using ieee 802.15.4 to two-floors environment, in: 2022 IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech), 2022, pp. 239–240, https://doi.org/10.1109/LifeTech53646.2022.9754790.
[12] R. Zitouni, J. Petit, A. Djoudi, L. George, Iot-based urban traffic-light control: modelling, prototyping and evaluation of mqtt protocol, in: 2019 International

Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2019, pp. 182–189, https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00051.

[13] J. Petit, R. Zitouni, L. George, Prototyping of urban traffic-light control in iot, in: 2018 IEEE International Smart Cities Conference (ISC2), 2018, pp. 1–2, https://doi.org/10.1109/ISC2.2018.8656717.

[14] V. Tanyingyong, R. Olsson, M. Hidell, P. Sjoedin, B. Ahlgren, Implementation and deployment of an outdoor iot-based air quality monitoring testbed, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 206–212, https://doi.org/10.1109/GLOCOM.2018.8647287.

[15] S. Biju, N.M. Shekokar, Security approach on mqtt based smart home, in: 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 1106–1114, https://doi.org/10.1109/ICPCSI.2017.8391883.

[16] R. Herrero, Mobile shared resources in the context of iot low power lossy networks, Internet of Things 12 (2020), 100274.

[17] J. Tournier, F. Lesueur, F.L. Mouel, L. Guyon, H. Ben-Hassine, A survey of iot protocols and their security issues through the lens of a generic iot stack, Internet of Things 16 (2021), 100264, https://doi.org/10.1016/j.iot.2020.100264. URL, https://www.sciencedirect.com/science/article/pii/S2542660520300986.

[18] J. Higuera, J. Polo, Understanding the ieee 1451 standard in 6lowpan sensor networks, in: 2010 IEEE Sensors Applications Symposium (SAS), 2010, pp. 189–193, https://doi.org/10.1109/SAS.2010.5439427.

[19] W. Ayoub, M. Mroue, F. Nouvel, A.E. Samhat, J.-c. Prevotet, Towards ip over lpwans technologies: lorawan, dash7, nb-iot, in: 2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC), 2018, pp. 43–47, https://doi.org/10.1109/DINWC.2018.8356993.

[20] R. Tabish, A. Ben Mnaouer, F. Touati, A.M. Ghaleb, A comparative analysis of ble and 6lowpan for u-healthcare applications, in: 2013 7th IEEE GCC Conference and Exhibition (GCC), 2013, pp. 286–291, https://doi.org/10.1109/IEEEGCC.2013.6705791.

[21] J. Decuir, Bluetooth smart support for 6lobtle: applications and connection questions, IEEE Cons. Electron. Mag. 4 (2) (2015) 67–70, https://doi.org/10.1109/MCE.2015.2392955.

[22] R. Herrero, Ultrasonic physical layers as building blocks of iot stacks, Internet of Things 18 (2022), 100489, https://doi.org/10.1016/j.iot.2021.100489. URL, https://www.sciencedirect.com/science/article/pii/S2542660521001281.

[23] Y. Chen, T. Kunz, Performance evaluation of iot protocols under a constrained wireless access network, in: 2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT), 2016, pp. 1–7, https://doi.org/10.1109/MoWNet.2016.7496622.

[24] R. Herrero, Mqtt-sn, coap, and rtp in wireless iot real-time communications, Multimed. Syst.doi:10.1007/s00530-020-00674-5. URL https://doi.org/10.1007/s00530-020-00674-5.

[25] C. Lai, Y. Hwang, The voice controlled internet of things system, in: 2018 7th International Symposium on Next Generation Electronics (ISNE), 2018, pp. 1–2, https://doi.org/10.1109/ISNE.2018.8394640.

[26] O. Hohlfeld, R. Geib, G. Hasslinger, Packet loss in real-time services: markovian models generating qoe impairments, in: 2008 16th International Workshop on Quality of Service, 2008, pp. 239–248, https://doi.org/10.1109/IWQOS.2008.33.

[27] ITU-T Recommendation P.863, Tech. Rep., International Telecommunication Union, Geneva, Switzerland (Sep.)..

[28] ITU-T recommendation P.501, Test signals for use in Telephonometry, Feb. 2009.