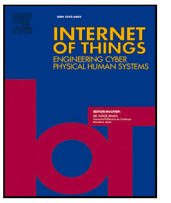
Internet of Things 18 (2022) 100489

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot



Research article

Ultrasonic physical layers as building blocks of IoT stacks

Rolando Herrero

College of Engineering, Northeastern University, 360 Huntington Ave., Boston, MA, USA

ARTICLE INFO

Keywords:
6LoWPAN
Ultrasound
ITU V.23
CoAP
CSMA/CA
Layers

ABSTRACT

Instrumental, Scientific and Medical (ISM) bands are essential to the transmission of IoT traffic as most of the physical layers of the *Wireless Personal Area Network* (WPAN) and *Low Power Wide Area Network* (LPWAN) families rely on them. ISM bands, however, are associated with a myriad of problems responsible for signal degradation and pollution that lead to application *Quality of Service* (QoS) issues. In this context, the use of alternative transmission mechanisms serve as backup to traditional schemes. One of such mechanisms relies on sub-ultrasonic and ultrasonic signals for the propagation of IoT device traffic. In this paper, we recycle a well known acoustic modulation scheme intended to be used in legacy *Public Switching Telephone Networks* (PSTNs) to support sub-ultrasonic and ultrasonic channels. This scheme becomes the basic building block of a physical layer that is combined with other well known elements of the IoT layered architecture. This not only includes a customized link layer but also an adaptation layer that enables full integration with upper layers.

1. Introduction

The two main families of low power IoT technologies LPWANs and WPANs represent a trade-off between transmission rate and coverage. LPWANs are typically associated with low transmission rates as well as long coverage and WPANs are characterized by high transmission rates and short distances. WPANs are widely used in the context of indoor IoT solutions that include home, building automation and connected health among others. Examples of WPAN physical and link layers include IEEE 802.15.4, BLE, ITU-T G.9959 and *Digital Enhanced Cordless Telecommunications Ultra Low Energy* (DECT ULE) [1]. All these technologies rely on the transmission of frames over non-licensed ISM bands that are becoming increasingly more polluted. This increases the media access contention, that these devices are typically subjected to, and it degrades latency as well as overall application *Quality of Service* (QoS).

Sub-ultrasonic and ultrasonic physical layers provide a valid alternative to traditional WPANs over ISM bands. As presented in Section 2, there have been several attempts to providing physical and link layer IoT support over sub-ultrasonic and ultrasonic channels. In this paper, we present a new *Near Field Communication* (NFC) IoT physical layer, named *Ultrasonic V.23* (UV23), that relies on ITU V.23 acoustic modulation and demodulation adapted to transmission over sub-ultrasonic and ultrasonic carriers [2]. This mechanism is supported by traditional CSMA/CA *Media Access Control* (MAC) and a simple frame format that are responsible for the UV23 link layer. In this context, 16-bit MAC addresses, configured on network deployment, provide basic link connectivity.

Because the UV23 frame size is small, when compared to the minimum MTU size requirements of IPv6, adaptation is needed. One of the most popular IPv6 adaptation mechanisms, 6LoWPAN, enables the compression of IPv6 header fields as well as the lowering of MTU size requirements to make it compatible with well-known IoT link layers like IEEE 802.15.4 and BLE [1]. In this paper we use 6LoWPAN to adapt the UV23 frames in order to support end-to-end IPv6 connectivity [3]. In order to evaluate the performance of this scheme, we present an experimental framework where the application-to-device interaction is carried out by means of the

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

https://doi.org/10.1016/j.iot.2021.100489
Received 15 August 2021; Received in revised form 10 December 2021; Accepted 12 December 2021
Available online 4 January 2022

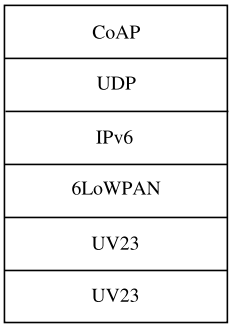


Fig. 1. Ultrasonic IoT Stack.

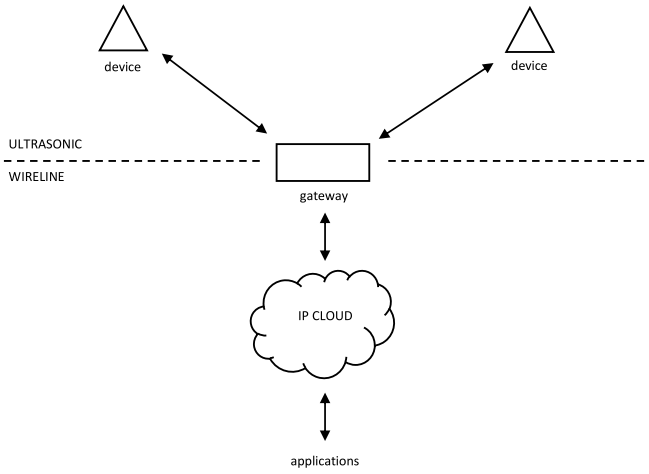


Fig. 2. Ultrasonic IoT Topology.

Constrained Application Protocol (CoAP) [4]. Essentially, we analyze the application layer throughput, packet loss and latency of the proposed stack and compare them against those of a mathematical model. The model, in turn, enables real-time estimation of latency and throughput as a function of network conditions that can be used to perform dynamic stack configuration. The full stack proposed in this paper is shown in Fig. 1. From a topology perspective, shown in Fig. 2, the scenario addressed in this chapter consists of a central gateway that forwards traffic back-and-forth to and from two devices by means of IoT based NFC supported by the aforementioned stack. A solution is said to be NFC if the signal coverage is less than 20 centimeters [5]. Moreover, in the context of real-time applications like those supported by VoIP and IoT scenarios, a packet loss of 5% is considered the threshold for acceptable service [6].

The contributions of this paper are several: (1) it provides a full IoT stack that relies on a sub-ultrasonic and ultrasonic physical layer based on the successful ITU V.23 standard of acoustic modulation, (2) it introduces a novel link layer that is optimized to minimize throughput over the sub-ultrasonic and ultrasonic channel, (3) it presents a network layer that, based on 6LoWPAN, enables efficient wide end-to-end IPv6 support in this scenario, (4) it introduces a mathematical model that supports real-time latency and throughput estimations that enable dynamic stack configuration and (5) it evaluates the performance of the overall scheme by considering the effect of packet loss and latency over a CoAP based session layer.

The remainder of the paper is organized as follows: a full description of other sub-ultrasonic and ultrasonic based IoT solutions is given in Section 2. Details of the sub-ultrasonic and ultrasonic physical and link layers are introduced in Section 3. The IPv6 adaptation scheme and the session layer mechanism are described in Section 4. In Section 5, an analytical model of the relationship between latency, packet loss and throughput is presented. An experimental framework that is used to evaluate the performance of the communication framework is introduced in Section 6. Conclusions and future work are provided in Section 7.

2. Review of literature

The use of sub-ultrasonic and ultrasonic channels to provide NFC communications have evolved throughout the years to support applications ranging from plain point-to-point scenarios to full network connectivity. All these approaches, however, are based on proprietary mechanisms and none of them complies with the IoT paradigm. For the most part these solutions involve legacy

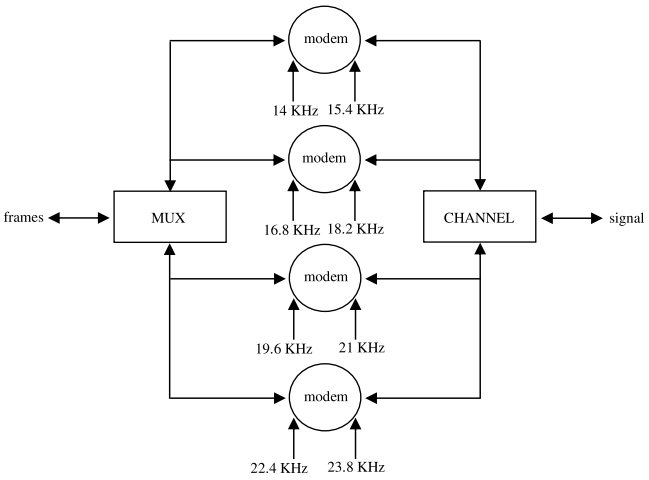


Fig. 3. Physical Layer Diagram.

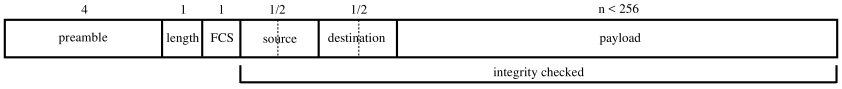


Fig. 4. Link Layer Frame.

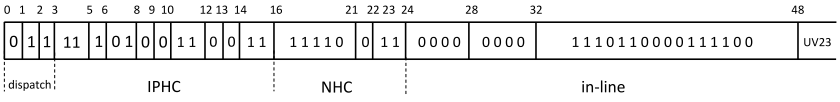
mechanisms that accomplish very low transmission rates on dedicated hardware [7–11]. Most other solutions focus on using smartphones that natively rely on audio interfaces in order to support ultrasonic audio playback and recording. An analysis of the characteristics of ultrasound generation in smartphones is presented in [12]. In [13] the authors provide a *Room Area Network* (RAN) based on ultrasonic communications. A similar approach that embeds ultrasonic data streams in the audio transmitted by television signals is explored by several authors in [14,15]. Using ultrasonic modulated data to complement traditional radio based communications is presented in [16,17]. A modulation approach that relies on adapting an acoustic modem is introduced in [18]. A similar *Orthogonal Frequency Division Multiplexing* (OFDM) mechanism that relies on multiple channels is detailed in [19]. Finally, in [20] the authors present a physical and link layer mechanism for ultrasonic communications in the context of IoT. Note that neither of these papers really address ultrasonic communications from a perspective of IoT. All of them present physical and link layers and ignore upper layers where a lot of the IoT innovation exists. Moreover, they present proprietary mechanisms that ignore the main requirement of any IoT solution: end-to-end IPv6 support. Additionally, none of them model the characteristics of the proposed schemes to make real-time decisions like changing system parameters. This paper addresses all these deficiencies by introducing a full IoT stack that relies on an ultrasonic physical and link layer. Moreover, it explores the implications of the stack from a performance perspective by modeling and analyzing latency, network packet loss and throughput. By relying on standard IoT upper layer protocols, it enables the integration with traditional radio based IoT infrastructure.

In reviewing the literature, besides the aspects associated with the ultrasonic physical and link layers, the use of CoAP as a mechanism for the establishment of IoT session has been an area of extensive study in the context of traditional radio based communications. Specifically, in [21] the authors measure application loss and latency in a CoAP based LLN. The transmission of CoAP over different transport protocols is analyzed in [22]. In [23], latency, loss and network bandwidth are measured in a constrained network where CoAP and other IoT mechanisms, are compared. In [24] the authors examine CoAP in the context of IoT media streaming. In [25] the authors analyze the congestion control mechanisms introduced for CoAP and measure latency, loss and throughput for different network scenarios. Similarly, in [26] CoAP is analyzed when utilized with service discovery mechanisms.

3. Physical and link layers

ITU V.23 defines a now obsolete *modulation–demodulation* (modem) standard for transmission of low rate data over PSTN lines. It enables nominal transmission rates of up to 1200 bps. The modulation scheme relies on basic Binary Frequency Shift Keying (BFSK) with one bit per symbol where carriers are centered at 1.3 KHz and 1.7 KHz for the transmission of zeros and ones respectively. ITU-T V.23 was not defined to work in a networking environment so there are no specific MAC mechanisms associated with it. In this paper, the ITU V.23 scheme is further enhanced to support the transmission over sub-ultrasonic and ultrasonic bands as illustrated in Fig. 3.

The mechanism is carried out by an OFDM scenario where link layer frames are multiplexed for transmission over four different ITU V.23 modems tuned at separate sub-ultrasonic and ultrasonic frequencies. Specifically, OFDM provides a generic arrangement of multiple carriers that rely on particular modulation schemes. The idea is to divide the available channel bandwidth W into a



TF	11	neither traffic class not flow label encoding
N = 1	1	NHC follows
HLM	01	hop limit is 64
C	0	no context information
S	0	no context based source address encoding
SAM	11	link layer source address
M	0	unicast destination address
D	0	no context based destination address encoding
DAM	11	link layer destination address
UDP	11110	UDP transport
K	0	transmit checksum in-line
P	11	4-bit source and destination port encoding
source	0000	UDP source port (61440)
destination	0000	UDP destination port (61440)
checksum	11 .. 00	checksum

Fig. 5. 6LoWPAN IPHC and NHC headers.

Table 1
MAC address encoding.

Initial	Final	Length (bytes)	Addresses
00	7F	1	0 - 127
80 00	FF FF	2	128 - 32767

number of subchannels of equal bandwidth Δf such that around $K = {}^{W}/_{\Delta f}$ subchannels are available for modulation. Under OFDM, the carrier waves are orthogonal, that is

$$\int_0^T S_i(t)S_j(t)dt = 0$$

where S_n is one of the M symbols associated with the modulation, T is the symbol length and $i \neq j$. Orthogonality improves resilience against packet loss caused by channel noise and multipath phenomenon. If $T = {}^{1}/_{\Delta f}$, where Δf is the frequency difference between symbol sinusoidal waves, the carrier waves are orthogonal for any possible value of the carrier phase. Because orthogonality restricts the period length of each symbol, it also imposes a new limitation of the transmission rate in each subchannel. In this ultrasonic scenario, where all carries are BFSK ITU V.23 compliant, the orthogonality of the carrier separation is set in accordance with a symbol duration of 714.29 microseconds.

Because the transmission rate supported by each modem is 1400 bps, the maximum nominal transmission rate of the whole scheme is 5600 bps. The 1400 bps speed associated with the proposed scheme has to do with a slightly increased transmission rate that is possible because the carriers are shifted to a higher frequency to comply with ultrasonic communication. The symbol period is shorter so it is possible to increase the transmission rate accordingly. Note that firmware implementations of the ITU V.23 modem are widely available and highly optimized to run on constrained hardware [27]. To adapt a regular ITU V.23 modem to the scheme presented in this paper, it is only necessary to change the sampling rate from 8000 to 48000 samples per second in order to comply the with range of the sub-ultrasonic and ultrasonic carriers indicated in Fig. 3.

Each modem shown in Fig. 3 accesses the sub-ultrasonic and ultrasonic channel by means of CSMA/ CA [28]. If modem has a frame to send and the channel is free (identified by detecting the lack of a signals), it transmits it right away. Alternatively, if the channel is busy, the modem monitors for ongoing frame transmissions waiting for the channel to become available. When the channel is finally free, the modem waits for a random amount of time, known as backoff period. The purpose of the backoff timer is to randomize channel access and prevent collisions.

Fig. 4 shows a UV23 frame. It starts with a 4-byte 0xAA55AA55 preamble and it follows with 1-byte length and *Frame Checksum* (FCS) fields. The length field specifies the size of the payload (not larger than 256 bytes). The FCS field is calculated over the MAC addresses and the payload as a simple *eXclusive OR* (XOR) 1-byte operation. The FCS is generated by the transmitter and verified by the receiver. The MAC addresses can be one or two bytes long and support values between 0 and 32767. As shown in Table 1, 1-byte MAC addresses encode values between 0 and 127, while 2-byte MAC addresses encode values between 128 and 32767. MAC addresses with zero value are used for broadcast transmissions. At nominal transmission rates, an empty frame takes 11.43 ms to be transmitted. On the other hand, a maximum size of 2120-bit frame, takes 378.6 ms to be transmitted.

4. IPv6 adaptation and session layer

Because IoT is intended to connect billions of devices, end-to-end IPv6 support is clearly one of the main requirements. In the end, the use of IPv6 provides a very large address space that guarantees support for this connectivity. Long addresses associated with a large address space imply long headers that are not typically compatible with the low transmission rates of most IoT scenarios. In

Table 2
Stack performance.

d_1 (cm)	Transmission rate (bps)	Throughput (bps)	Message loss (%)	Latency (s)
5	4974.41	4860.99	2.33	56.16
10	4977.96	4752.16	4.91	57.56
15	4921.43	4499.93	10.69	60.8
20	4942.31	3760.32	32.63	72.54
25	4998.24	3266.82	53.54	101.32
30	4819.01	2612.68	67.12	144.2

Table 3
SNR_{dB} at receivers.

d_1 (cm)	SNR_{dB} at application	SNR_{dB} at gateway	SNR_{dB} at sensor
5	53.24	51.49	53.52
10	50.96	50.14	51.32
15	48.07	46.50	48.32
20	40.02	38.21	40.46
25	28.19	26.97	28.02
30	20.75	18.47	20.08

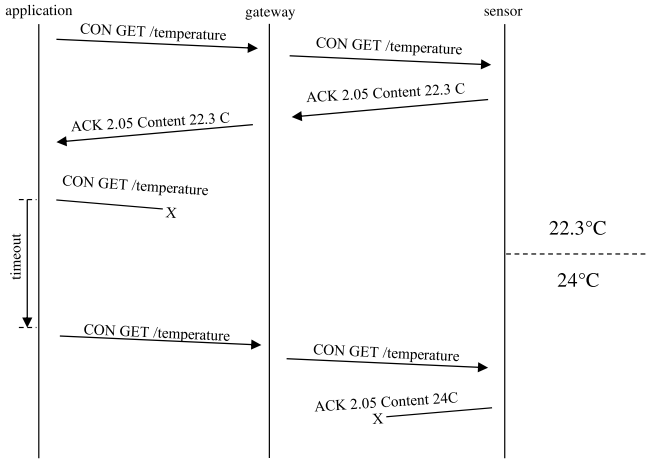


Fig. 6. Session Message Flow.

this context, IPv6 is responsible of frame size requirements that are characterized by a minimum MTU size of around 1280 bytes. This prevents and limits the direct use of IPv6 over the UV23 physical and link layers. In this scenario, IPv6 adaptation by means of 6LoWPAN becomes key. 6LoWPAN was initially designed to provide adaptation over IEEE 802.15.4 but it has been extended to support other physical and link layer technologies. 6LoWPAN provides, among other things, IPv6 and UDP header compression and efficient fragmentation. As in most IoT scenarios, IPv6 relies on *Stateless Address Autoconfiguration* (SAA) to generate unicast network addresses that are derived from MAC addresses [29]. This provides the ultimate IPv6 header compression as it is not necessary to transmit addresses. In the context of this solution, we rely on traditional stateful 6LoWPAN *IP Header Compression* (IPHC) and *Next Header Compression* (NHC) for IPv6 and UDP header compression respectively. Fig. 5 shows a 6LoWPAN datagram carrying a UV23 frame. The IPHC and NHC headers are 6 bytes long, they remove some of the fields of the IPv6 header like the traffic class and flow label and compress others like the hop limit. UDP ports are compressed and encoded as 4-bit fields that support 16 dynamically out-of-band allocated ports. Because under normal circumstances IPv6 and UDP headers are 40 and 8 bytes long respectively, the use of 6LoWPAN leads to a compression rate of 1/8.

Interaction with IoT devices is done by means of *REpresentational State Transfer* (REST) message exchanges. Specifically, REST requests and responses support connectivity between clients and servers. CoAP is a very good candidate to complete the stack as it provides a generic scenario of REST interaction between devices and applications/gateways. Specifically, CoAP is an application layer protocol that enables session management on constrained devices. To deal with some of the shortcomings associated with UDP, CoAP introduces two modes of operation; (1) *non-confirmable* where a packet is sent as "fire-and-forget" without any guarantee of successful delivery and (2) *confirmable* where a packet is considered delivered once a far-end initiated acknowledgment is received. Specifically, confirmable CoAP introduces a much simpler retransmission mechanism than that of TCP.

Fig. 6 shows the interaction between an application requesting temperature readouts and a sensor. The requests are forwarded by a gateway that sits in between the application and the sensor. Note that this is a generic scenario that can be applied to other type of sensing solutions. The gateway acts as a proxy that supports a single CoAP session for which it forwards application layer

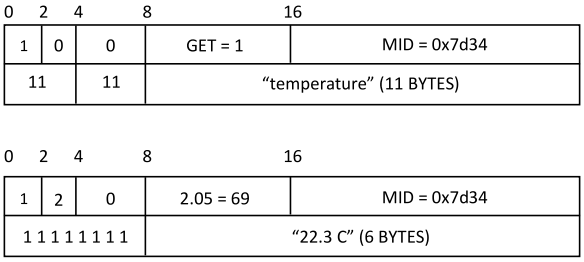


Fig. 7. CoAP Encoding.

request and responses but that regenerates all lower layers packets including the datagrams that result from separate IPv6 addresses derived from the assigned UV23 MAC addresses. Note that this is in opposition to a more complex scenario where the gateway fully regenerates the stack such that there are two CoAP sessions that are terminated and restarted at the gateway. Under confirmable CoAP, the application transmits a GET request to retrieve a single readout that is piggybacked in the ACK response from the sensor. If either message, GET or ACK, is lost, retransmissions occur at exponentially increasing intervals in accordance with RFC 7252 "The Constrained Application Protocol (CoAP)" [4]. The CoAP message encoding, shown in Fig. 7, is quite simple. Messages include a 2-byte *Message Identifier* (MID) with the asset identifier (i.e. temperature) and readouts encoded as plain-text in their payloads. In this context, the nominal lengths of the CoAP requests and responses are 17 and 12 bytes long respectively. When combining all the other layers (physical, link, network and transport), requests and responses become 31 and 26 bytes long respectively. Assuming a maximum rate, the overall transmission latencies are 44.28 and 37.14 ms respectively.

5. Analytical model

An analytical model of the flow in Fig. 6 can be used to determine the relationship between latency, packet loss and throughput. Essentially, for each session, the total amount of traffic transmitted between the application and the sensor through the gateway, defined as $\bar{L}$, is given by

$$\bar{L} = 2L_{\text{req}} + 2L_{\text{resp}} \tag{1}$$

where L_{req} and L_{resp} are the request and response sizes respectively. The network packet loss, defined as P_{L}, is given by

$$P_{\text{L}} = 1 - P_{\text{S}} = \bar{L}P_{\text{e}} \tag{2}$$

where P_{e} is the bit error probability.

The latency, defined as Δ and associated with a single session, results from considering up to $M = 3$ exponentially increasing retransmissions. It is given by

$$\begin{aligned} \Delta &= \frac{\bar{L}}{R}\sum_{n=0}^{M} 2^n P_{\text{S}} P_{\text{L}}^n \\ &= \frac{\bar{L}}{R}P_{\text{S}}\sum_{n=0}^{N}\left(2P_{\text{L}}\right)^n \\ &= \frac{\bar{L}}{R}P_{\text{S}}\frac{\left(2P_{\text{L}}\right)^{M+1}-1}{2P_{\text{L}}-1} \end{aligned} \tag{3}$$

where R is the transmission rate of the scheme and $P_{\text{L}} < 1/2$ for the geometric series above to converge. In turn, the throughput, defined as T, is given by

$$T = \frac{\bar{L}}{\Delta}. \tag{4}$$

6. Experimental framework

The performance of the NFC IoT communication stack can be evaluated by implementing the interaction between the devices and the gateway shown in the message flow in Fig. 6. Specifically, the application transmits confirmable CoAP requests that are forwarded by the gateway and arrive at the sensor. The sensor generates readouts that are, in turn, forwarded to the application. In order to implement this scenario, we created a plugin for VPS+ [30] that supports the UV23 physical and link layers presented in Section 3. VPS+ is general IoT tool that enables the fast prototyping of layered architecture stacks including their fast deployment into virtual and physical devices. The UV23 plugins and the standard 6LoWPAN, IPv6, UDP and CoAP layers are deployed on two Android Galaxy S6 smartphones playing roles of application and sensor and on a regular Dell Latitude E5470 laptop that provides

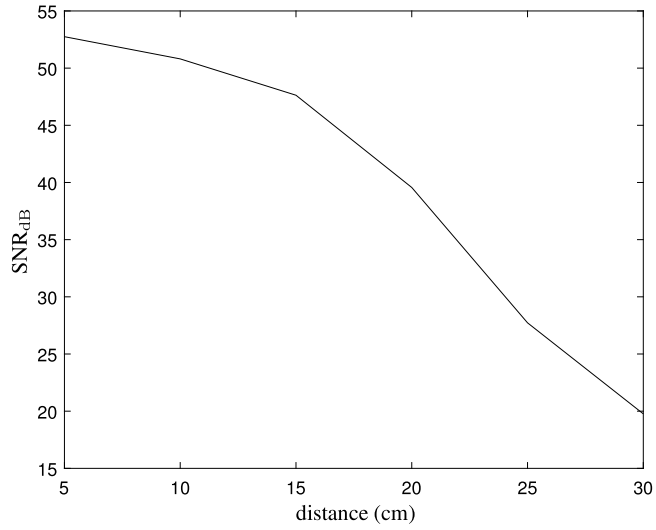


Fig. 8. SNR_{dB} vs d_1.

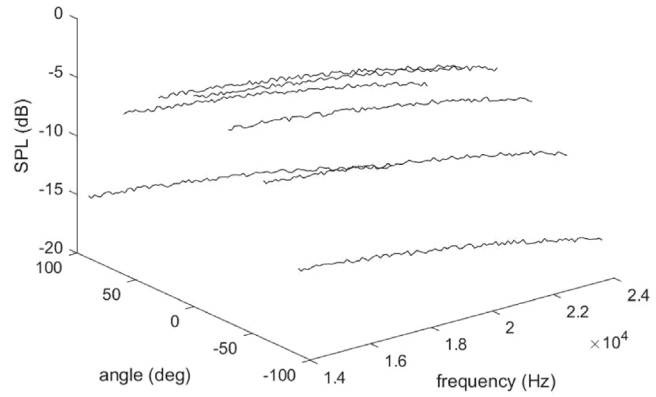


Fig. 9. Polar Response Gateway.

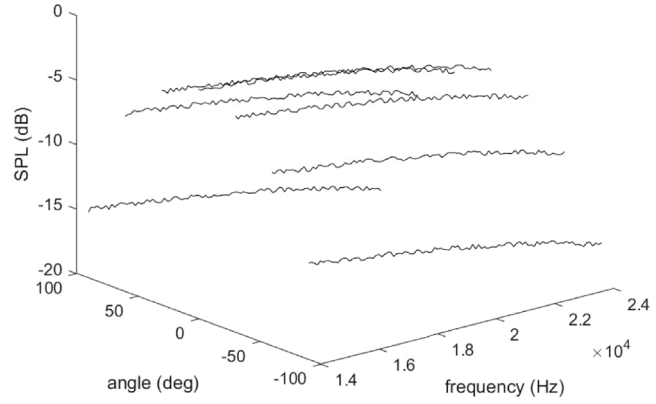


Fig. 10. Polar Response Application/Sensor.

gateway functionality. Fig. 13 shows a photograph of the setup. The selection of smartphones and a laptop is based on the fact that these devices natively support sub-ultrasonic and ultrasonic audio playback and recording interfaces. The modem implementation of the physical layer is based on the well known open source linmodem [27] (see Fig. 18).

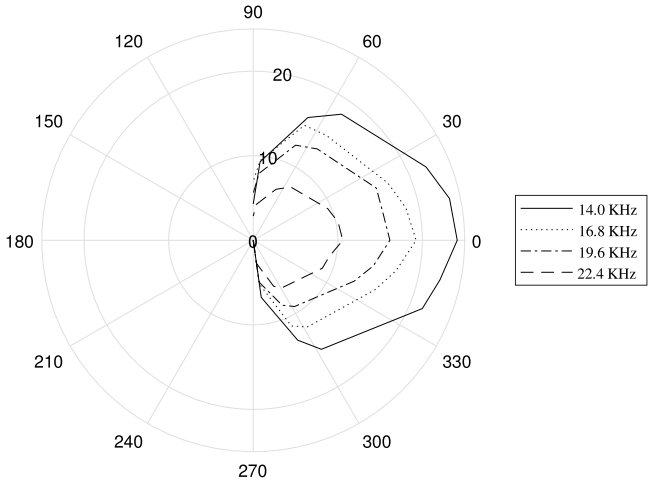


Fig. 11. Zenithal Response Gateway.

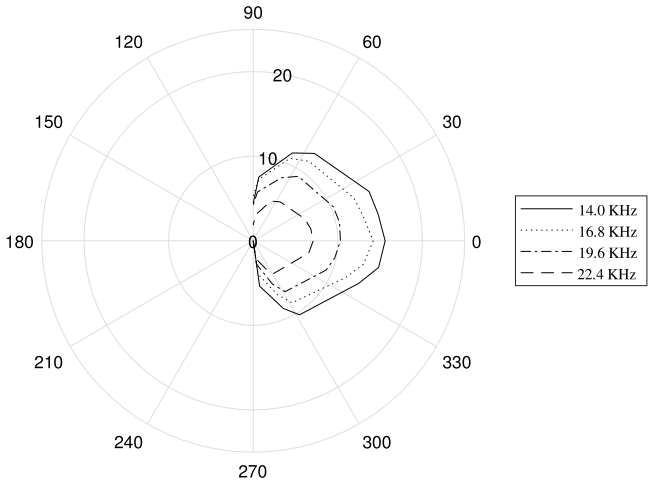


Fig. 12. Zenithal Response Application/Sensor.

In this experimental scenario, a VPS+ script drives the application and the emulated temperature sensors to generate CoAP requests and responses as well as to accurately measure performance parameters. Each test case consists of requesting and receiving 300 temperature readouts such that the distance d_1 between the laptop and the phones is the control variables. The separation between phones is fixed at $d_2 = 30$ centimeters. A few parameters are measured in each case, the effective transmission rates, throughput, message loss and overall latency of the scheme.

The Table 2 shows the performance of the scheme for different values of the distance d_1. The transmission rate and throughput result from averaging them for all 300 sessions. The transmission rate is lower than the nominal transmission rate due to MAC contention. The throughput is lower than the transmission rate due to the packet loss that results in additional retransmissions. The packet loss is measured as a percentage of the number of transmitted packets that are lost while the latency is measured as the time it takes for all 300 readouts to arrive to the application. Note that the performance quickly degrades as the distance d_1 increases. Specifically, for d_1 above 30 centimeters the degradation is such that the use of the scheme becomes impractical as it dramatically lowers the overall throughput. This is compliant, however, with the characteristics of NFC scenarios. Moreover, as with any NFC solution, the assumption is *Line-of-Sight* (LoS) communication with no blockages. Fig. 14 compares network packet loss and throughput as a function of distance while Fig. 15 compares network packet loss and latency as a function of distance. Table 3 shows the received SNR_{dB} measured on reception at each of the devices as the second power of the RMS value of the sampled signals. Again, there is heavy correlation between d_1 and SNR. Note that the SNR levels at the application and sensor devices are of the same order as they are identical from a hardware and software perspective. Similarly, the SNR level at the gateway, which runs on a laptop, is slightly higher as hardware and software are different. Fig. 8 shows the average SNR_{dB} measured at the receivers as a function of the distance d_1. Figs. 9 and 10 show the polar response of the *Sound Pressure Level* (SPL) measured at 30 centimeters of the emitter as a function of the angle of arrival and the frequency for the gateway and application/sensor respectively. Figs. 11 and 12 show

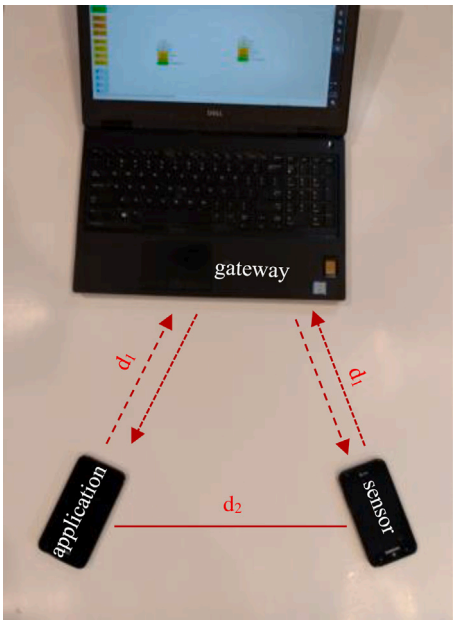


Fig. 13. VPS+ Setup.

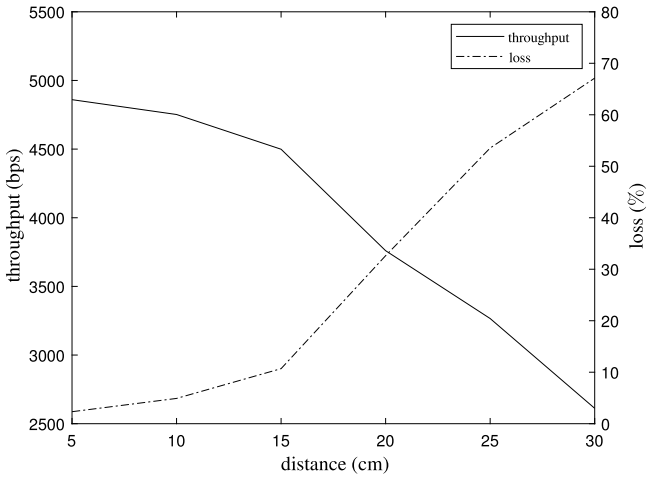


Fig. 14. Throughput vs Packet Loss vs Distance.

the SPL responses (in dB) measured at 30 centimeters of the emitter as a function of the compliment of the zenith angle for the gateway and application/sensor respectively. These SPL values are obtained at 14 different angles for four main carrier frequencies: 14 KHz, 16.8 KHz, 19.6 KHz and 22.4 KHz. Fig. 16 combines the results shown in Figs. 14 and 15 by comparing both latency and throughput as a function of the network packet loss. In turn, Fig. 17 shows both theoretical and experimental throughput as a function of loss. Similarly, Fig. 18 shows both theoretical and experimental latency as a function of loss. The theoretical throughput is obtained from Eq. (4). Note that theoretical and experimental throughputs differ from each one because, among multiple factors, the mathematical model does not take the effect of CSMA/CA into consideration. The overall relative errors between the model and the experimental scenarios are 5.08% and 7.82% for latency and throughput respectively.

7. Conclusion and future work

This paper introduced a full protocol stack that supports IoT scenarios in absence of traditional radio based communications. This protocol stack relies on novel physical and link layers that combine multiple ITU V.23 modems tuned to transmit over ultrasonic bands. An important requirement of IoT solutions, that is, the support of end-to-end IPv6 connectivity is carried out by means of 6LoWPAN adaptation. The scheme is tested through a setup that includes a sensor, an application and an gateway for which

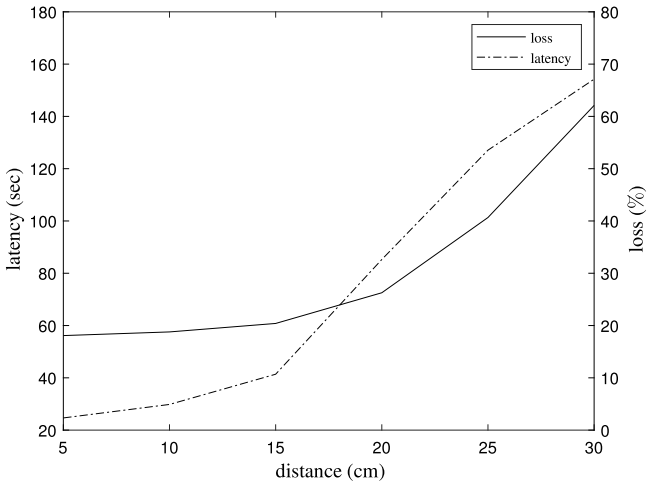


Fig. 15. Latency vs Packet Loss vs Distance.

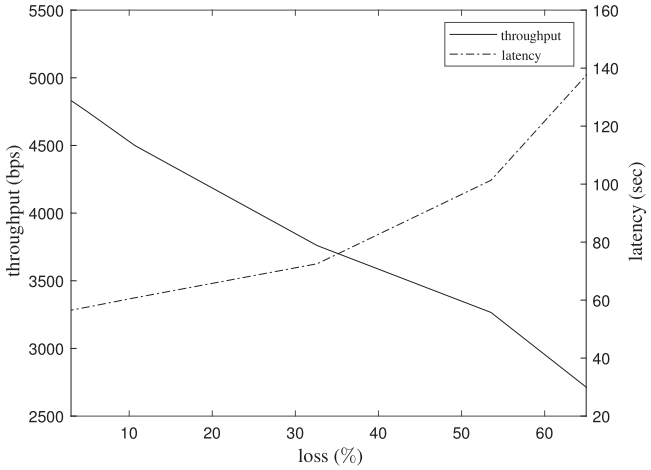


Fig. 16. Throughput vs Packet Loss vs Latency.

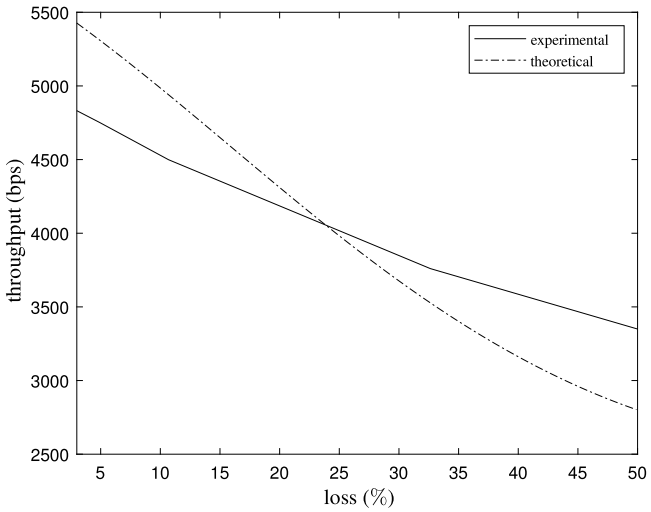


Fig. 17. Throughput vs Packet Loss.

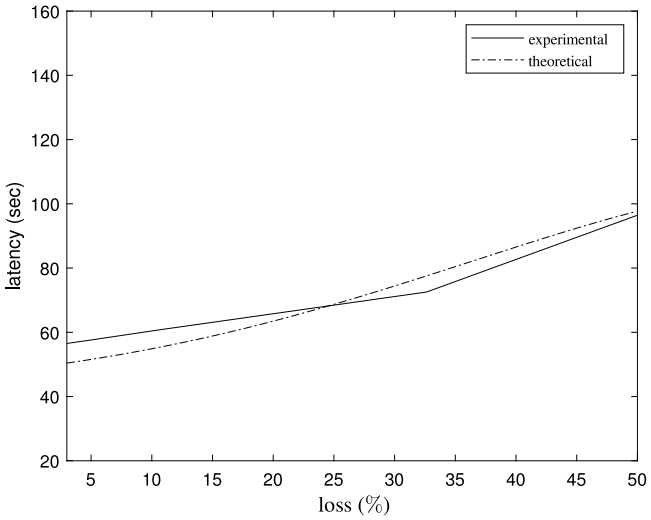


Fig. 18. Latency vs Packet Loss.

performance parameters are obtained. The results indicate that the distance between the devices is critical as degradation is heavily affected by coverage. In this context, the protocol stack presented in this paper can serve for a wide range of NFC applications ranging from physical security and access control to proximity sensors. To summarize, the main contributions of this paper are the following: (1) it introduces physical and link layers that rely on ultrasonic communications to enable NFC, (2) it integrates these physical and link layers with well-known IoT network, transport and application layers in order to support a full NFC IoT stack and (3) it provides an analytical model for the evaluation the performance of the overall scheme by considering the effect of packet loss and latency over a generic CoAP-based session layer. This paper provides the bases for future work that includes: (1) the use of other ultrasonic physical layer mechanisms like ITU V.34, (2) the analysis of multiple devices and how MAC contention affects performance as well as scalability and (3) the modeling and the effect of physical layer impairments on the application layer.

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, in: Textbooks in Telecommunication Engineering, Springer International Publishing, 2021, URL https://books.google.com/books?id=k70rzgEACAAJ.

[2] ITU-T Recommendation V.23: 600/1200-Baud modem standardized for use in the general switched telephone network, 1988.

[3] C. Bormann, Z. Shelby, S. Chakrabarti, E. Nordmark, Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), RFC 6775, in: Request for Comments, RFC Editor, 2012, http://dx.doi.org/10.17487/RFC6775, URL https://rfc-editor.org/rfc/rfc6775.txt.

[4] C. Bormann, K. Hartke, Z. Shelby, The Constrained Application Protocol (CoAP), RFC 7252, in: Request for Comments, RFC Editor, 2015, http://dx.doi.org/10.17487/rfc7252, URL https://rfc-editor.org/rfc/rfc7252.txt.

[5] R. Want, An introduction to RFID technology, 5 (1) (2006).

[6] M. Rizal, S. Taheri, D. Hogrefe, Empirical performance analysis of anonymizing VoIP over the onion router (TOR) network, in: 2013 International Conference on Privacy and Security in Mobile Systems (PRISMS), 2013, pp. 1–4, http://dx.doi.org/10.1109/PRISMS.2013.6927177.

[7] V. Gerasimov, W. Bender, Things that talk: Using sound for device-to-device and device-to-human communication, IBM Syst. J. 39 (3.4) (2000) 530–546, http://dx.doi.org/10.1147/sj.393.0530.

[8] A. Madhavapeddy, R. Sharp, D. Scott, A. Tse, Audio networking: the forgotten wireless technology, IEEE Pervasive Comput. 4 (3) (2005) 55–60, http://dx.doi.org/10.1109/MPRV.2005.50.

[9] A. Madhavapeddy, D. Scott, R. Sharp, Context-aware computing with sound, in: A.K. Dey, A. Schmidt, J.F. McCarthy (Eds.), UbiComp 2003: Ubiquitous Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 315–332.

[10] W. Jiang, W.M.D. Wright, Ultrasonic wireless communication in air using OFDM-OOK modulation, in: 2014 IEEE International Ultrasonics Symposium, 2014, pp. 1025–1028, http://dx.doi.org/10.1109/ULTSYM.2014.0251.

[11] M. Uddin, T. Nadeem, A2PSM: Audio assisted wi-fi power saving mechanism for smart devices, in: Proceedings of the 14th Workshop on Mobile Computing Systems and Applications, in: HotMobile '13, Association for Computing Machinery, New York, NY, USA, 2013, http://dx.doi.org/10.1145/2444776.2444782, URL https://doi.org/10.1145/2444776.2444782.

[12] Q. Wang, K. Ren, M. Zhou, T. Lei, D. Koutsonikolas, L. Su, Messages behind the Sound: Real-Time Hidden Acoustic Signal Capture with Smartphones, in: MobiCom '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 29–41, http://dx.doi.org/10.1145/2973750.2973765.

[13] P.A. Iannucci, R. Netravali, A.K. Goyal, H. Balakrishnan, Room-area networks, in: Proceedings of the 14th ACM Workshop on Hot Topics in Networks, in: HotNets-XIV, Association for Computing Machinery, New York, NY, USA, 2015, http://dx.doi.org/10.1145/2834050.2834113.

[14] A.S. Nittala, X.-D. Yang, S. Bateman, E. Sharlin, S. Greenberg, Phoneear: Interactions for mobile devices that hear high-frequency sound-encoded data, in: Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, in: EICS '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 174–179, http://dx.doi.org/10.1145/2774225.2775082.
[15] H. Lee, H.T. Kim, W.J. Choi, S. Choi, Chirp signal-based aerial acoustic communication for smart devices, IEEE Int. Conf. Comput. Commun. (2015).
[16] L. Zhang, X. Zhu, X. Wu, No more free riders: Sharing WiFi secrets with acoustic signals, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), 2019, pp. 1–8, http://dx.doi.org/10.1109/ICCCN.2019.8847081.
[17] B. Zhang, Q. Zhan, S. Chen, M. Li, K. Ren, C. Wang, D. Ma, $ssrPriWhisper$: Enabling Keyless secure acoustic communication for smartphones, IEEE Internet Things J. 1 (1) (2014) 33–45, http://dx.doi.org/10.1109/JIOT.2014.2297998.
[18] M. Hanspach, M. Goetz, On covert acoustical mesh networks in air, CoRR, 2014, arXiv:1406.1213.
[19] H. Matsuoka, Y. Nakashima, T. Yoshimura, Acoustic Communication System Using Mobile Terminal Microphones, 2007.
[20] E. Novak, Z. Tang, Q. Li, Ultrasound proximity networking on smart mobile devices for IoT applications, IEEE Internet Things J. 6 (1) (2019) 399–409, http://dx.doi.org/10.1109/JIOT.2018.2848099.
[21] S. Thombre, R.U. Islam, K. Andersson, M.S. Hossain, Performance analysis of an IP based protocol stack for WSNs, in: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016, pp. 360–365, http://dx.doi.org/10.1109/INFCOMW.2016.7562102.
[22] R. Herrero, Analysis of the constrained application protocol over quick UDP internet connection transport, Internet of Things 12 (2020) 100328.
[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, http://dx.doi.org/10.1109/MoWNet.2016.7496622.
[24] R. Herrero, Analysis of IoT mechanisms for media streaming, Internet Things 9 (2020) 100168, http://dx.doi.org/10.1016/j.iot.2020.100168.
[25] M. Collina, M. Bartolucci, A. Vanelli-Coralli, G.E. Corazza, Internet of things application layer protocol analysis over error and delay prone links, in: 2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2014, pp. 398–404, http://dx.doi.org/10.1109/ASMS-SPSC.2014.6934573.
[26] R. Herrero, Mobile shared resources in the context of IoT low power lossy networks, Internet of Things 12 (2020) 100274.
[27] F. Bellard, Linux Modem, https://bellard.org/linmodem/.
[28] Z. Dahham, A. Sali, B.M. Ali, M.S. Jahan, An efficient CSMA-CA algorithm for IEEE 802.15.4 wireless sensor networks, in: 2012 International Symposium on Telecommunication Technologies, 2012, pp. 118–123, http://dx.doi.org/10.1109/ISTT.2012.6481575.
[29] D.T. Narten, T. Jinmei, D.S. Thomson, IPv6 Stateless Address Autoconfiguration, RFC 4862, in: Request for Comments, RFC Editor, 2007, http://dx.doi.org/10.17487/RFC4862, URL https://rfc-editor.org/rfc/rfc4862.txt.
[30] VPS+, Visual Protocol Stack Emulator, URL https://www.l7tr.com.