

# VIoT: Voice over Internet of Things

Abdulkadir Karaagac  
Ghent University - imec, IDLab,  
Ghent, Belgium  
abdulkadir.karaagac@ugent.be

Pieterjan Camerlynck  
Televic Healthcare  
Izegem, Belgium  
p.camerlynck@televic.com

Pieter Crombez  
Televic Healthcare  
Izegem, Belgium  
p.crombez@televic.com

Jeroen Hoebeke  
Ghent University - imec, IDLab,  
Ghent, Belgium  
jeroen.hoebeke@ugent.be

**Abstract**—These days, the Internet of Things (IoT) is everywhere with a significantly increased number of devices connected to the Internet. Besides, we have also witnessed the broad adoption of the Internet telephony technologies in the last decade. In this regard, this paper investigates the integration of these two domains in order to enable voice and telephony services in IoT, resulting in a new paradigm that we named Voice over IoT (VIoT). To do so, a novel, efficient and low-cost integration architecture is introduced in order to connect IoT devices with voice capabilities to the Voice over Internet Protocol (VoIP) ecosystem and to enable people to interact with them. Also, a validation and evaluation study is presented in order to show the applicability of the proposed system for VIoT applications in industrial and consumer domains.

**Index Terms**—IoT, VoIP, IP telephony, streaming, SIP, LwM2M.

## I. INTRODUCTION

In recent years, the Internet of Things (IoT) is being considered as the technological revolution that consolidates various emerging concepts and enabling technologies to bridge the physical and virtual world [1]. IoT provides access to and interaction with a broad range of devices, systems and objects and enables a wide range of applications including smart healthcare, smart home, smart metering, and smart city [2].

Although most of the focus in IoT has been on the communication of a typical set of data (i.e. sensing, actuation), IoT applications emerge which present new and innovative ways to collaborate, communicate and interact, both from a human and machine perspective [3]. In this sense, as the natural mode of communication, the integration of Speech/Voice and Telephony into IoT applications can offer a versatile method to provide human interaction, communication, and control [3].

We believe that the successful integration of IP Telephony and IoT technologies is mutually beneficial and even essential for both fields. Such an advancement would enable numerous new IoT applications and products with voice-awareness, which can provide a more flexible user experience in a more economical way than traditional methods. Also, IP Telephony can gain more extensity and applicability by overcoming its limitations in supporting cyber-physical systems. Ultimately, all these new applications and features will result in a broader and smarter ecosystem. However, despite the significant potential of voice-enabled IoT applications, the integration of voice and telephony services and devices into the IoT still stays an unexplored phenomenon with several challenges.

Therefore, the primary goal of this work is the integration of voice and telephony services with *Smart* devices to make a smarter way of living possible. It first highlights the potential benefits, opportunities, and challenges of such an integration. It also explores the role of voice and telephony in IoT by outlining the multitude of IoT applications and use cases that would

benefit from voice integration. Then, it studies the first conceptual design of efficient and standard-based integration of Voice over IP (VoIP) and IoT ecosystems by taking the capabilities of IoT devices and the requirements of telephony applications into consideration, which enables the paradigm of Voice over IoT (VIoT). For this, it leverages on one of our previous works, which studied a standard-based streaming protocol for Internet of Multimedia Things (IoMT) [4]. Additionally, the technical capabilities and characteristics of the proposed architecture are validated and evaluated by means of practical analysis and a real-life implementation, which interconnects a VoIP client and a voice-enabled IoT device.

The remainder of the paper is organized as follows. Section II highlights the benefits, challenges, and opportunities for VIoT applications. Then, Section III provides technical background on the relevant concepts and technologies along with an overview of related work. Next, Section IV provides details on the proposed approach for VoIP and IoT integration and the design of the resultant VIoT paradigm, which is followed by a validation and evaluation study of the proposed architecture in Section V. Finally, Section VI concludes the paper.

## II. APPLICATIONS, OPPORTUNITIES AND CHALLENGES

With the emergence of the VIoT, voice and telephony services can start to play a pivotal role in various IoT segments by offering interesting properties and benefits. Primarily, voice integration can potentially challenge the need for a touch screen on many devices by providing similar functionalities in a more cost-effective way, especially for devices that will be dormant for the majority of the time. Furthermore, voice can provide productivity gains for the enterprise and industrial applications by better management of smart devices and machines over voice commands. For consumers, the voice can offer a better user experience in several applications by providing simpler means of user interaction, which creates a potential for voice to become the “*Touch*” of the IoT world. Besides these new functionalities and performance gains, as illustrated in *Fig. 1*, the realization of VIoT will also allow new applications and use cases, especially in wearables, healthcare, smart home and building fields [3].



Fig. 1 Voice over Internet of Things: Vision

However, these voice-enabled IoT applications have distinct characteristics and requirements that should be considered during IoT-VoIP integration. First, there are applications that require bi-directional voice communication for achieving human to human interaction on IoT devices, such as Intercom systems and alarm panels. Industrial or healthcare wearables, activity trackers and smartwatches can also make use of voice communication for remote interaction with the user in any cause (e.g. emergency, training, mentoring). The bandwidth and latency requirements of these voice applications are relatively high, but they are now more viable with the latest advancements in communication technologies [4]. While, some applications require only unidirectional voice communication, such as public address or notification systems. Also, IoT devices (actuators, household appliances) accepting voice commands over the phone can be considered as an example. By means of voice-enablement, a user can call in and instruct IoT gateways or home assistants to remotely control various connected appliances. These applications have softer communication requirements because the voice is transmitted in one direction towards or from the IoT device. Next, there can be also applications where only voice recognition is sufficient to achieve the targeted functionality, such as door entry controls/authentication or hands-free access to a device through voice recognition. Finally, the VoIP systems provide also support for video telephony, so IoT imaging devices can be integrated with VoIP so that a user can view the video feed from that device on a VoIP telephone.

In the pursuit of achieving all of these functionalities, there are a number of major issues. Initially, the voice and telephony applications rely on data flows that possess distinct characteristics as compared to typical traffic flows envisioned for IoT (sensing, actuation) [5], [6]. Therefore, voice/telephony services require special treatment (e.g. jitter compensation, packet reordering), while still meeting the constraints of IoT devices and networks. Secondly, due to the lack of a common architecture or standard that provides the interaction of voice systems and IoT devices, the structural, syntactical and semantic integration and interoperation of IoT and VoIP devices and components need to be realized. Finally, security vulnerability and privacy concerns should be addressed sufficiently.

### III. TECHNICAL BACKGROUND AND RELATED WORK

#### A. Voice over IP

VoIP, also known as IP Telephony, is the transmission of voice and multimedia content over IP networks [7]. Compared to traditional telephony, known as Public Switched Telephone Network (PSTN), VoIP is offering a much more cost-efficient telephony solution along with improved call quality, accessibility, flexibility, scalability and advanced services including security, conferencing and call routing. For achieving all these telephony functionalities over the Internet, VoIP uses a group of enabler technologies, such as Session Initiation Protocol (SIP) [8] and Real-time Transport Protocol (RTP) [9]. SIP is a signaling protocol for initiating, maintaining, and terminating communication sessions. Whereas, RTP is a transport protocol for real-time applications for delivering audio and video over IP networks. Fig. 2 presents a common architecture and protocol stack of SIP/RTP-based VoIP systems.

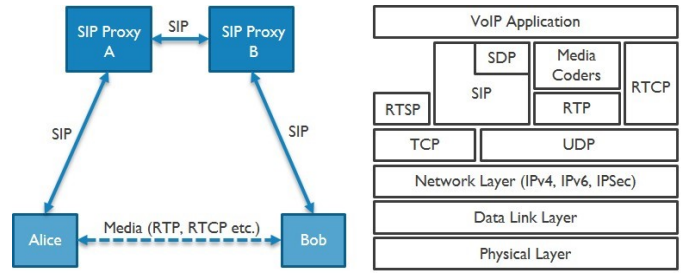


Fig. 2. Architecture and Protocol Stack for SIP-based VoIP

#### B. IoT Technologies and Protocols

IoT refers to a network of internet-connected things and smart embedded objects are considered an important part of this ecosystem. Therefore, there have been many efforts (6LoWPAN, 6lo, CoAP) to enable IP-based connectivity for constrained embedded devices and tiny objects. A common architecture and protocol stack for IoT are provided in Fig. 3. In addition, there are also several standardization initiatives and research efforts targeting interoperability and Machine-to-Machine (M2M) understandability in IoT. The Lightweight M2M (LwM2M) protocol, specified by the OMA Alliance, is one of these leading M2M communication specifications, which offers several functionalities for managing resource-constrained devices on a variety of networks [10].

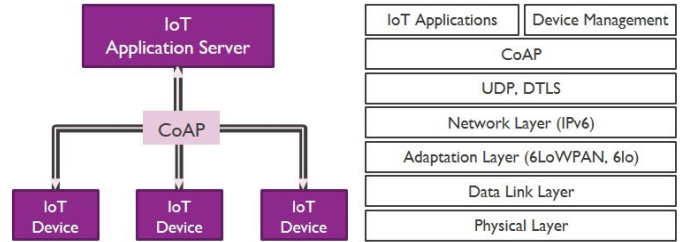


Fig. 3. IoT Architecture and Protocol Stack

#### C. Related Work

Despite the potential of voice applications in the IoT ecosystem, the integration of voice into IoT applications is still an issue, largely due to the constrained nature of IoT devices and technologies, i.e. CPU, memory, battery, limited/unpredictable bandwidth [5][11]. In this regard, there are some previous attempts that targeted the adaptation of the VoIP enabler protocols for IoT devices. RTP-IoT [11] extends RTP/RTCP protocols to make them more suitable for IoT. Similarly, CoSIP [12] creates a light-weight session initiation mechanism, based on CoAP, aiming at allowing constrained devices to instantiate communication sessions in a standardized fashion. Besides, there are also studies that extend the CoAP protocol to better support media streaming features in IoT [13], [14]. In [4], we investigated an efficient streaming protocol for IoMT applications and services, in which heterogeneous multimedia things can interact seamlessly with each other and with other connected devices. The proposed solution uses open IoT protocols and creates a uniform representation for the media semantics combined with interaction models that can be used for dealing with various multimedia technologies and streaming applications. This solution also defines the means of managing voice-related parameters and monitoring streaming statistics.

Regarding the voice-enablement in IoT, these prior works only study the transfer of media data along with limited session and transport messaging capabilities. But, no full structural and architectural design is investigated that enables telephony and voice services in IoT devices. There are also no existing efforts that implement or evaluate any telephony operation in IoT.

#### IV. VOIP AND IOT INTEGRATION

The seamless connectivity and spontaneous interoperability of IoT and VoIP technologies will eliminate boundaries between two enormous ecosystems and enable numerous new IoT and Telephony applications with more capabilities, where things can communicate with other things or people over a telephone network. In this sense, the joint usage of the Internet and UDP protocols as the network and transport protocol by both technologies simplifies the integration process.

However, there are issues to be addressed in order to achieve this integration. First, VoIP and IoT use different protocols for application, authentication, security and device management purposes. So, these protocols need to be translated to or integrated with each other in order to create a single network for all kinds of data. Another issue is related to device identity and addressing. SIP-based VoIP uses the combination of usernames and domain names in order to uniquely identify the end devices, whereas IoT uses IP addresses and unique endpoint names. In addition, SIP defines client to client addressing, whereas IoT generally defines server-client communication and no scheme is defined for client-to-client interactions.

Based on the characteristics of IoT and VoIP technologies, we propose an integration mechanism by means of VIoT Gateways (GW), which can be typically operated by VIoT service providers or privately. Therefore, it is similar to the broadly adopted PSTN GW approach, also known as VoIP GW, which converts telephony traffic into IP for transmission over a data network [15]. Only in this case, the conversion will happen between VoIP and IoT traffic instead of PSTN.

##### A. Architecture

For the realization of the VIoT GW, we opted a scalable and light-weight design that virtualizes SIP clients for VIoT devices by means of containers. As illustrated in Fig. 4, the proposed VIoT GW resides between the IoT and VoIP domains and acts as the core unit that is responsible for all of the integration and translation processes. The VIoT GW virtualizes and manages SIP clients, performs device mapping/addressing and call/stream forwarding. The GW may virtualize a SIP client for every IoT streaming device or maybe only in case of call requests depending on the application scenario. The integration process is transparent to the SIP domain and SIP clients will see IoT devices as connected SIP clients in the same domain.

The VIoT GW consists of three main components. The first and most important one is the Integration and Translation Logic (ITL) entity. This component is responsible for performing several fundamental integration operations including client virtualization, address management, device mapping, presence and call management, QoS management, media processing, and transcoding if needed. The second component is the cluster of virtualized SIP clients that interacts with the SIP server and other

SIP clients for achieving connectivity with the VoIP world. These SIP clients are virtualized (via containers) and managed by the ITL entity. They operate based on the commands (call initiation, answering) and voice data coming from the ITL.

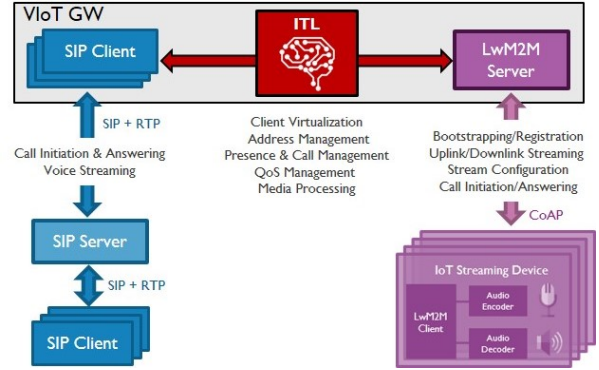


Fig. 4 VIoT GW - System Architecture

The last component is the LwM2M Server that is responsible for the interaction with IoT devices and realizing essential management operations (bootstrapping, registration, security) along with telephony functionalities (streaming, call initiation termination) in IoT. Thanks to the usage of a single standardized interface for all operations in the IoT domain, the proposed architecture boosts the memory efficiency for the end devices while eliminating any further effort for the interoperability even with classical IoT devices and systems.

##### B. Virtualization and Address Management

The integration process starts with a registration message from an IoT device to the LwM2M Server. This message includes the device's connection properties (lifetime, queue mode) along with an optional *Endpoint Client Name*. Upon the registration request of a connected VIoT device, the VIoT GW will virtualize a SIP client, with configuration and addressing, for that IoT device. Next, the virtual SIP client registers to the SIP server and notifies the presence of the IoT device on a certain SIP address. As long as the IoT device stays connected, the VIoT GW needs to maintain the mapping between the virtual SIP client and the attached IoT device, so it can forward the telephony requests to the right IoT device. Regarding the addressing of the virtual SIP clients, the VIoT GW can assign an SIP address as a combination of the *Endpoint Client Name* (string) and the particular domain name (or simply an IP address) used by the Server. This will result in a SIP address that is structured as *sip:endpointname@domain:port*.

##### C. Presence and Call Management

Upon the reception of the first registration message and the virtualization of the SIP client, the IoT device becomes connected to the SIP ecosystem and can be reached for any telephony operation. As long as the IoT device stays connected to the LwM2M server, it needs to maintain its registration by performing a periodical *Update* of its registration information to the LwM2M Server. When shutting down or discontinuing the use of the VIoT GW, the LwM2M Client performs a *De-register* operation, which results in the removal of the registration of that LwM2M Client. Similarly, if the lifetime of the registration expires without receiving an update from the LwM2M Client,

the LwM2M Server will consider it as a de-registration. Then, in case of a de-registration event, the VIoT GW will inactivate the corresponding SIP client. At that moment, the IoT device is no longer present in the VoIP domain and thus not reachable for any telephony operation. The details of the presence management process are presented in Fig. 5.

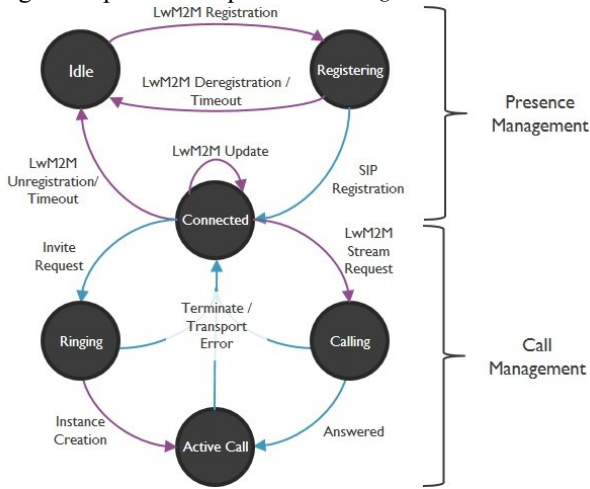


Fig. 5 VIoT GW - Presence and Call Management

During the connected state, the IoT device can initiate a call towards any VoIP client and any VoIP client can send a call invitation to any IoT device. In case of a call request towards the IoT device, upon the reception of the *Invite* message, the VIoT GW will first check if the corresponding IoT device is present, connected and available. If not, the gateway responds to the invite request with an associated SIP error message. If the IoT device is available, the VIoT GW sends a *Ringing* response. Next, it sends *Create* messages to the corresponding IoT device for relevant streaming objects. The IoT device can accept or reject this request. If accepted, the VIoT GW informs the SIP server about the confirmation and an active call is initiated with bi-directional voice streaming. Until a termination request from any side or any transport error, the data streaming will last. In case of the termination of the active call, the gateway needs to delete created streaming object instances. The details of a calling process from a SIP client to an IoT device is provided in Fig. 6.

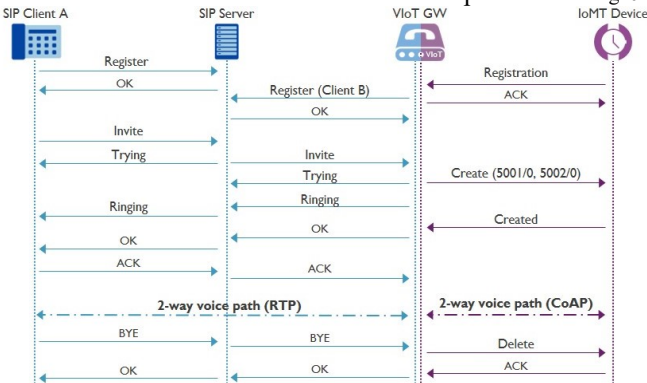


Fig. 6 VIoT GW - Telephony operation

#### D. QoS Management and Media Processing

An intelligible telephony communication depends on the continuous and steady transmission of voice packets and it might be extremely sensitive to insufficient bandwidth, packet drops,

excessive delay and jitter. Therefore, it is crucial to verify and maintain QoS for the VIoT voice and media traffic. However, the exchanged voice packets hold audio frames with varying sizes (typically between 10 and 60 ms), periods and packet sizes (from a few bytes to hundreds of bytes) depending on the encoding format and settings. In addition, the communication technologies used for IoT applications have remarkably distinct characteristics and bitrates starting from a few bps (e.g. SigFox) to several Mbps (e.g. Bluetooth LE). Even, the quality of the IoT network may dynamically change over time. Therefore, another envisioned functionality for the VIoT GW is the active monitoring of streaming statistics and dynamic configuration of streaming parameters based on application requirements (coding format, bitrate) and communication resources.

Moreover, further media processing operations (transcoding, compression, etc.) can be performed for a variety of causes. For instance, transcoding might be crucial in cases where a target IoT device does not support the incoming format or has a limited storage capacity that mandates a reduced file size or to convert incompatible or obsolete data to a better-supported format. Or it can concatenate or fragment audio segments, based on the supported Maximum Transmission Unit (MTU), in order to achieve target communication performance. In this case, sending fewer packets will reduce the bandwidth requirements due to the reduced packet headers. However, it will increase latency and sensitivity to packet losses, as losing one packet will result in a loss of a larger chunk of audio.

#### E. System Implementation

In order to realize the proposed VIoT architecture, we developed a VIoT platform, which connects voice-enabled IoT devices with SIP networks and realizes full-duplex voice communication. As presented in Fig. 7, for the VIoT GW implementation, we used an open-source LwM2M Server platform, named Leshan, with required extensions (Media Sensor/Actuator objects, Streaming API) as defined in [4]. For the virtual SIP clients, we used Linphone application and extended it with Control and Stream APIs for establishing interactions with the ITL to place and receive calls, inform presence, and configure codecs. The details of the VIoT GW implementation are provided in Fig. 7. On one side, it has Control and Stream interfaces, which are used to interact with and control the SIP clients. On the other side, it exposes the IoT API which interacts with the Leshan Server in order to receive LwM2M registration and timeout notifications, send control messages, perform management operations and also streaming operations. Apart from these interfaces, the ITL entity includes a number of components that perform the fundamental integration functionalities defined in the previous section, such as call control, presence management, and media processing.

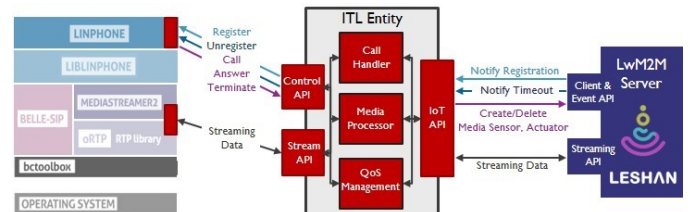


Fig. 7 VIoT Gateway - Internal Architecture

## V. VALIDATION AND EVALUATION

In order to validate and evaluate the VIoT platform, we created a flexible experimental setup where a VoIP softphone can make calls with a voice-enabled IoT device in various network settings. As illustrated in Fig. 8, the experimental setup can connect an IoT device to the VIoT GW via a configurable emulated link in addition to a Bluetooth LE and Narrowband IoT (NB-IoT) connection. For the emulated link, we made use of the Click Modular Router [16] and its traffic shaping functionalities. First, we took the specific characteristics (bandwidth, delay, jitter, etc.) and limitations (e.g. MTU) of the target technologies and shaped the particular link to emulate similar conditions.

In this setup, the IoT device was a Raspberry Pi that contains LWM2M client application (extended for telephony functionalities) along with audio input/output capabilities. The SIP client was a Raspberry Pi that runs the Linphone softphone application. The VIoT GW was a powerful PC that runs the VIoT Gateway module. Considering the capabilities of the IoT network technologies, Codec2, a low-bitrate audio coding format (450-3200 bit/s), is used. Other coding formats can also be used as long as they offer bitrates that can be transferred with the available network resources.

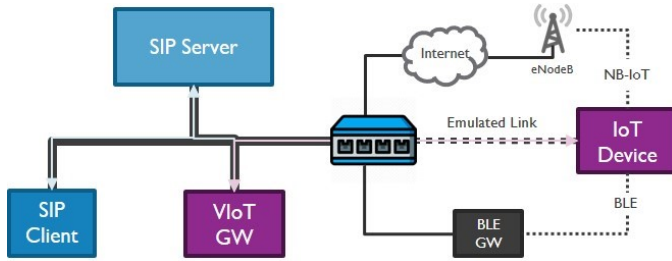


Fig. 8 Experimentation Setup with Configurable Emulated Link

### A. Experimental Results

1) *Latency*: In order to investigate the effect of the network latency on the quality of VIoT applications, we emulated a variable delay for the IoT device connection. Our observations in these experiments show that latency of 20ms is not noticeable, whereas latencies of 100-150ms are barely noticeable and therefore still acceptable. For higher values, the call quality starts to diminish and starting from 300ms calls become relatively slow and slightly interrupted. Around 700ms, call quality becomes unacceptable for phone conversations. But push-to-talk or intercom systems might be still feasible.

Besides, only unidirectional voice streaming or voice recognition applications can be realized for latencies above 1s.

2) *Jitter*: Jitter, the packet latency variation, is one of the most common VoIP call quality problems. Our experiments with emulated jitter show that the effect of network-related jitter on the VIoT call quality is not noticeable when it is below 30ms. The call quality starts to diminish around 40-50ms and leads to choppy conversations. Above 100ms, the VIoT conversation becomes unintelligible. Jitter buffers can be used to eliminate jitter by queuing a number of packets to absorb the delay differences. But, this will lead to larger playout delays that must be considered together with the latency requirements.

3) *Media Throughput*: We also performed experiments to investigate the achievable end-to-end media throughput in case of various communication frequencies and voice segment sizes. Fig. 9a illustrates the outcome of one of these experiments where we realized a VIoT connection over an emulated link with a variable communication period. In this test, we used Codec2 mode 3200, which converts 20ms frames to 64 bits to be sent over a data channel. The figure shows that, in case of no aggregation, only technologies with a communication period of 24 ms or smaller can provide sufficient throughput. However, with aggregated frames, even a 100ms communication period is still sufficient. These results show that lower communication frequencies may result in insufficient throughput, thus inconsistent call quality, especially in case of small voice segment sizes. Considering the typical packetization time (10-80ms) of VoIP applications, several IoT networking technologies may not be able to provide a sufficient communication frequency. However, by aggregating audio frames in voice segments, the media throughput can be increased, which can enable several IoT technologies to be used for VIoT applications, with the cost of larger delays.

### B. NB-IoT and BLE Measurements

In order to evaluate the VIoT platform, we also performed tests with real networks: NB-IoT and BLE. In theory, NB-IoT offers lower limits for the communication periods (24-28ms for downlink, 21-45ms for uplink single-tone) based on the network quality and utilized transport block size (TBS) [4]. Similarly, BLE v4.2 specification offers connection interval ranges from 7.5 ms to 4 seconds. This means that NB-IoT and BLE, for many connection interval settings, are not expected to provide a sufficient packet rate for Codec2 and many audio encoding

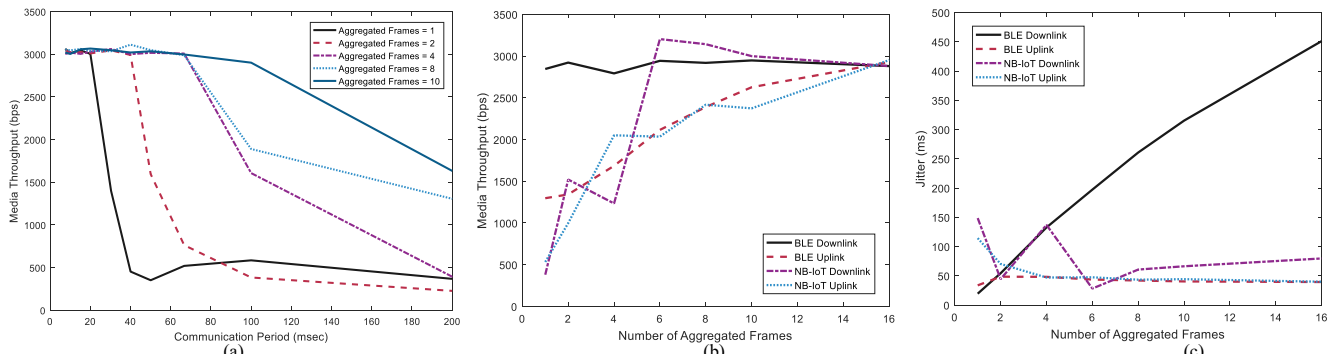


Fig. 9 (a) Media Throughput for varying Communication Period. (b) Throughput and (c) Jitter Measurements for BLE and NB-IoT.

formats without any frame aggregation. This is validated by our experimental results in Fig. 9b and 9c. In these measurements, BLE was configured to use a 7.5 ms connection interval, while the NB-IoT network was using a TBS size of 208 bits. As reaching 3.2 kbps media throughput was the target, both NB-IoT and BLE technologies were able to provide sufficient throughput only if a certain number of frames are aggregated, depending on the technology properties. Meanwhile, the measured jitter values were also acceptable, except for the BLE downlink, probably due to scheduling errors in the BLE firmware. Also according to Table I, which presents the best case (BC) and worst-case (WC) performances of VIoT operations over BLE and NB-IoT, both technologies can provide sufficient performance only with the right level of frame aggregation.

TABLE I. MEASUREMENTS FOR NB-IoT AND BLE

Technology - Scenario	Aggregation	Bandwidth (kbps)	Jitter (ms)	Voice Quality
BLE-Downlink-BC	2	3.159	46.4	Good
BLE-Downlink-WC	1	2.249	25.3	Poor
BLE-Uplink-BC	16	3.192	37.6	Good
BLE-Uplink-WC	2	1.254	51.6	NA
NB-IoT-Downlink-BC	6	3.197	28.9	Good
NB-IoT-Downlink-WC	1	0.264	223.5	NA
NB-IoT-Uplink-BC	16	3.044	39.5	Good
NB-IoT-Uplink-WC	1	0.521	118.4	NA

### C. Discussion

As these experiments show, the quality of voice depends severely on the performance and characteristics of the network underneath. Especially, considering the diverse and dynamic nature of the IoT networks, the network technologies should be selected and configured carefully based on the application requirements and device capabilities for the proper functioning of VIoT applications. For instance, powerful communication protocols, like Wi-Fi, can easily provide high-quality VIoT calls. While, relatively constrained IoT protocols, such as BLE and NB-IoT, can provide low-bitrate VIoT applications (e.g. unidirectional streaming, voice recognition) in case of careful configuration and network adaptation. Besides, due to limited bandwidth and duty-cycle limitations, it is not practical to achieve voice communication (with any coding format of today) in certain IoT technologies, such as LoRa and SigFox.

Furthermore, VIoT systems should also adapt to the level and variation of network performance ) to achieve target functionalities. But this QoS management (transcoding, compression, frame aggregation, etc.) come with a cost of higher latency, jitter and complexity. VIoT GW can also interact with network management entities to ensure the required communication resources. Or it can ask the end device to use the technology which can provide the required QoS, in case of multiple technology support on the device.

## VI. CONCLUSION

In this work, we investigated the lightweight, efficient and standard-based integration of the VoIP and IoT ecosystems in order to enable voice and telephony services in IoT, to connect Smart IoT devices to VoIP systems and to enable people to speak to IoT. We also showed that telephony services are suited to a wide range of IoT applications and use cases, and can enable

numerous new IoT applications in consumer, enterprise and industrial domains. Moreover, the value proposition of VoIP will also broaden together with IoT by enabling communication and interaction with billions of things. We believe this work can provide a baseline, for the Telephony developers, about how to use IoT protocols and platforms in order to integrate telephony products in IoT applications. It can also help IoT system providers to understand the characteristics and needs of telephony technologies in order to realize their semantic and structural integration and interoperation in the context of IoT. Moreover, this work can also serve as a starting point for further research on self-adaptive VIoT platforms including device and network-aware media coding and processing for IoT applications, and adaptive voice-aware network configuration.

## ACKNOWLEDGMENT

This research was funded in part by the ICON project MAGICiAN. MAGICiAN is realized in collaboration with imec, with project support from VLAIO and Innoviris. Project partners are imec, Orange Belgium, Televic Healthcare, Restore and Citymesh. Part of this research was funded by the FWO-Flanders under grant agreement #G084177N.

## REFERENCES

- [1] A. Al-Fuqaha et al., "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials* 17 (4), pp. 2347–2376, 2015.
- [2] L. D. Xu, W. He and S. Li, "Internet of things in industries: a survey," in *IEEE Transactions on Industrial Informatics*, vol. 10(4), Nov. 2014.
- [3] A. Brown, "White paper: the role of voice in IoT applications," *Strategy Analytics*, December 2015.
- [4] A. Karaagac et al., "Light-weight streaming protocol for the internet of multimedia things: voice streaming over NB-IoT," in *Elsevier Pervasive and Mobile Computing*, 2019.
- [5] C. Lewis, S. Pickavance, "Implementing Quality of Service Over Cisco MPLS VPNs," in *Selecting MPLS VPN Services*, Cisco Press, 2006.
- [6] A. Duresi and R. Jain, "RTP, RTCP and RTSP for Real Time Applications," in: R. Zurawski (Ed.), *The Industrial Information Technology Handbook*, CRC Press, Ch. 28, 2005.
- [7] S. Karapantazis and F.-N. Pavlidou. "VoIP: A comprehensive survey on a promising technology," *Computer Networks*, 53 (12), 2009.
- [8] J. Rosenberg, H. Schulzrinne et al., "SIP: Session Initiation Protocol", RFC 3261, IETF, June 2002.
- [9] H. Schulzrinne et al., "RTP: A transport protocol for real-time applications," RFC 3550, IETF, July 2003.
- [10] Open Mobile Alliance, "Lightweight Machine to Machine Technical Specification: Core595v1.0", February 2017.
- [11] O. Said, Y. Albagory, M. Nofal, F. A. Raddady, "IoT-RTP and IoT-RTCP: Adaptive protocols for multimedia transmission over internet of things environments", *IEEE Access* 5, July 2017.
- [12] S. Cirani, M. Picone, L. Veltri, "CoSIP: A Constrained Session Initiation Protocol for the Internet of Things," in *Proc. of Advances in Service-Oriented and Cloud Computing*, 2013.
- [13] Guang Lu et al., "Mechanisms to Support Adaptive Constrained Application Protocol (CoAP) Streaming for Internet of Things (IoT) Systems", WO 2016/210109A1, 2016.
- [14] A. Bhattacharyya et al., "Adaptive RESTful Real-time Live Streaming for Things (A-REaLiST)", IETF Internet Draft, Oct. 2018.
- [15] M. Hamdi et al., "Voice service interworking for PSTN and IP networks," in *IEEE Communications Magazine*, vol. 37(5), May 1999.
- [16] E. Kohler et al., "The click modular router," *ACM Transactions on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.