



ICDT 2023

The Eighteenth International Conference on Digital Telecommunications

ISBN: 978-1-68558-034-6

April 24th – 28th, 2023

Venice, Italy

ICDT 2023 Editors

Pacal Lorenz, University of Haute Alsace, France

ICDT 2023

Forward

The Eighteenth International Conference on Digital Telecommunications (ICDT 2023), held between April 24th and April 28th, 2023, continued a series of events focusing on telecommunications aspects in multimedia environments. The scope of the conference was to address the lower layers of systems interaction and identify the technical challenges and the most recent achievements.

The conference served as a forum for researchers from both the academia and the industry, professionals, and practitioners to present and discuss the current state-of-the art in research and best practices as well as future trends and needs (both in research and practice) in the areas of multimedia telecommunications, signal processing in telecommunications, data processing, audio transmission and reception systems, voice over packet networks, video, conferencing, telephony, as well as image producing, sending, and mining, speech producing and processing, IP/Mobile TV, Multicast/Broadcast Triple-Quadruple-play, content production and distribution, multimedia protocols, H-series towards SIP, and control and management of multimedia telecommunications.

High quality software is not an accident; it is constructed via a systematic plan that demands familiarity with analytical techniques, architectural design methodologies, implementation polices, and testing techniques. Software architecture plays an important role in the development of today's complex software systems. Furthermore, our ability to model and reason about the architectural properties of a system built from existing components is of great concern to modern system developers.

Performance, scalability and suitability to specific domains raise the challenging efforts for gathering special requirements, capture temporal constraints, and implement service-oriented requirements. The complexity of the systems requires an early-stage adoption of advanced paradigms for adaptive and self-adaptive features.

Online monitoring applications, in which continuous queries operate in near real-time over rapid and unbounded "streams" of data such as telephone call records, sensor readings, web usage logs, network packet traces, are fundamentally different from traditional data management. The difference is induced by the fact that in applications such as network monitoring, telecommunications data management, manufacturing, sensor networks, and others, data takes the form of continuous data streams rather than finite stored data sets. As a result, clients require long-running continuous queries as opposed to one-time queries. These requirements lead to reconsider data management and processing of complex and numerous continuous queries over data streams, as current database systems and data processing methods are not suitable. Event stream processing is a new paradigm of computing that supports the processing of multiple streams of event data with the goal of identifying the meaningful events within those streams.

We take here the opportunity to warmly thank all the members of the ICDT 2023 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to ICDT 2023. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the ICDT 2023 organizing committee for their help in handling the logistics of this event.

We hope that ICDT 2023 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of digital telecommunications.

ICDT 2023 Chairs

ICDT 2023 Steering Committee

Stan McClellan, Texas State University - San Marcos, USA
Bernd E. Wolfinger, University of Hamburg, Germany
Amalia Miliou, Aristotle University of Thessaloniki, Greece

ICDT 2023 Publicity Chairs

Laura Garcia, Universitat Politecnica de Valencia, Spain
Javier Rocher Morant, Universitat Politecnica de Valencia, Spain

ICDT 2023 Committee

ICDT 2023 Steering Committee

Stan McClellan, Texas State University - San Marcos, USA
Bernd E. Wolfinger, University of Hamburg, Germany
Amalia Miliou, Aristotle University of Thessaloniki, Greece

ICDT 2023 Publicity Chairs

Laura Garcia, Universitat Politecnica de Valencia, Spain
Javier Rocher Morant, Universitat Politecnica de Valencia, Spain

ICDT 2023 Technical Program Committee

Arsalan Ahmad, National University of Sciences and Technology, Islamabad, Pakistan / Trinity College
Dublin, Ireland
Ayad Al-Adhami, Plymouth University, UK / University of Technology, Iraq
Babak Barazandeh, University of Southern California, USA
Ilija Basicovic, University of Novi Sad, Serbia
Pierre Beausery, Université de Technologie de Troyes, France
Larbi Boubchir, University of Paris 8, France
Abhishek Das, Aliah University, Kolkata, India
Somaieh Davar, Concordia University, Canada
Laura-Maria Dogariu, University Politehnica of Bucharest, Romania
Tan Do Duy, Ho Chi Minh City University of Technology and Education, Vietnam
Zakaria Abou El Houda, University of Montreal, Canada
David Espes, Université de Bretagne Occidentale, France
Mário Ferreira, University of Aveiro, Portugal
Mohamed Fezari, Annaba University, Algeria
Rita Francese, Università di Salerno, Italy
Felix J. Garcia-Clemente, University of Murcia, Spain
Benedict R. Gaster, University of West of England, UK
Pantea Nadimi Goki, TeCIP Institute Scuola Superiore Sant'Anna, Italy
Carlos Guerrero, University of Balearic Islands, Spain
Makhlouf Hadji, Technological research Institute SystemX, France
Ruimin Hu, Wuhan University, China
Yishan Jiao, Pearson Education, USA
Mohammed Jouhari, Mohammed VI Polytechnic University, Benguerir, Morocco
Maria Kallionpaa, Hong Kong Baptist University, Hong Kong
Wen-Hsing Lai, National Kaohsiung University of Science and Technology, Taiwan
Jan Lansky, University of Finance and Administration, Czech Republic
Moonjin Lee, Korea Maritime and Ocean University / University of Science & Technology Korea /
Research Institute of Ship and Ocean engineering, Korea
Isaac Lera, University of the Balearic Islands, Spain
Xiaoli Liu, University of Helsinki, Finland

Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Vladimir Lyashev, Huawei Technologies Co. Ltd. - Moscow Research Center, Russia
Min Ma, Google speech, USA
Adnan Mahmood, Macquarie University, Australia
S. Manoharan, University of Auckland, New Zealand
Alexandru Martian, University Politehnica of Bucharest, Romania
Stan McClellan, Texas State University, USA
Amalia Miliou, Aristotle University of Thessaloniki, Greece
Ioannis Moscholios, University of Peloponnese, Greece
Dmitry Namiot, Lomonosov Moscow State University, Russia
Morteza Noshad, University of Michigan, USA
Patrik Österberg, Mid Sweden University, Sweden
Constantin Paleologu, University Politehnica of Bucharest, Romania
Euthimios Panagos, Perspecta Labs Inc., USA
Liyun Pang, Huawei Germany Research Center, Germany
Maciej Piechowiak, Kazimierz Wielki University, Poland
Luca Potì, Universitas Mercatorum | CNIT, Italy
Andy Reed, The Open University, UK
Eric Renault, IMT-TSP, France
Ravikant Saini, Indian Institute of Technology Jammu, India
Abdel-Badeeh M. Salem, Ain Shams University, Cairo, Egypt
Ana Serrano Tellería, University of Castilla La Mancha, Spain
Akbar Sheikh-Akbari, Leeds Beckett University, UK
M. Estela Sousa-Vieira, University of Vigo, Spain
Cristian Lucian Stanciu, University Politehnica of Bucharest, Romania
Da-Wen Sun, University College Dublin, Ireland
Christopher Tegho, Calipsa, London, UK
Giorgio Terracina, Università della Calabria, Italy
Tony Thomas, Indian Institute of Information Technology and Management - Kerala, India
Božo Tomas, University of Mostar, Bosnia and Herzegovina
Laszlo Toth, University of Szeged, Hungary
Mahyar Turchi Moghaddam, INRIA Grenoble-Rhône-Alpes, France
Scott Trent, IBM Research - Tokyo, Japan
Chrisa Tsinaraki, European Commission - Joint Research Centre, Italy
Ming Tu, JD AI Research, Mountain View, USA
Adriano Valenzano, CNR-National Research Council, Italy
Rob van der Mei, Centre for Mathematics and Computer Science (CWI), Amsterdam, Netherlands
Calin Vladeanu, University Politehnica of Bucharest, Romania
Sergey V. Volvenko, Peter the Great St. Petersburg Polytechnic University, Russia
Haixin Wang, Toyota Motor North America R&D InfoTech Labs, USA
Lei Wang, University of Connecticut, USA
Bernd E. Wolfinger, University of Hamburg, Germany
Qilian (Vision) Yu, University of California, Davis, USA
Zbigniew Zakrzewski, UTP University of Science and Technology, Poland
Ligang Zhang, School of Engineering and Technology - Central Queensland University, Australia
Piotr Zwierzykowski, Poznan University of Technology, Poland

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission to reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

On the Acoustic and Articulatory Characterization of the Effects of Arabic Pharyngealized Consonant on Adjacent Vowel <i>Fazia Karaoui, Rachida Djeradi, and Amar Djeradi</i>	1
On the Performance of a Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation <i>Ionut-Dorinel Ficiu, Cristian-Lucian Stanciu, Constantin Paleologu, and Felix Albu</i>	6
Comparative Performance of TCP and MQTT <i>Bishal Thapa, Bishal Sharma, and Stan McClellan</i>	10

On the Acoustic and Articulatory Characterization of the Effects of Arabic Pharyngealized Consonant on Adjacent Vowel

X-ray Study of Arabic Pharyngealized Consonant

Fazia Karaoui
URSL/AALA
LCPTS/FEI/USTHB
Algiers, Algeria
e-mail: k.fazia6cp@yahoo.fr
e-mail: fkaraoui@usthb.dz

Rachida Djeradi and Amar Djeradi
dept: Telecommunication/LCPTS/FEI/USTHB
Algiers, Algeria
e-mail: r_djeradi@yahoo.fr
e-mail: adjeradi05@yahoo.com

Abstract—The phonetic and phonological properties of the pharyngealized consonants are essential for understanding how they affect the phonological systems of the languages in which they occur. This study focuses on consonants with a primary or secondary pharyngeal articulation, which occur in Moroccan Arabic dialect. Pharyngeal or pharyngealized consonants are rare and they are understudied. The manifestation of pharyngealization is centered between the consonant and vowel as in distortion of the formant transitions in the adjacent vowel. This study used articulatory and acoustic data to characterize the pharyngealized Moroccan Arabic dialect sound and highlight their effect on neighboring vowel. The acoustic results from very specific articulatory configuration show that the vowel formants are influenced by the pharyngealized contexts and the articulatory gestures spread on the adjacent vowel. It was also found that the pharyngealized and the voiced pharyngeal consonants have similar effects on the neighboring vowel.

Keywords—pharyngealized consonants; Arabic vowels; coarticulation; x-ray images; vocal tract.

I. INTRODUCTION

During speech production, a series of articulatory gestures are realized by means of the phonatory organs movement in the vocal tract. The passage from the production of isolate phonemes to continuous speech involved an enormous articulatory and acoustic variability. Today, technological developments in medical imaging, progress in the observation of organs and muscles facilitated by new tools, have produced significant advances in the understanding of the mechanisms involved in speech production [1].

In this work, we indicate the methods and techniques used to observe the activity of organs in speech production and the possibilities offered by the vocal tract. The way of envisaging how this is done is to situate the process in a physiological theory of speech production and invoke an articulatory model. Close attention to the movement of the phonatory articulators involved in speech production can shed light on the consequences of certain actions on other organs. The great mobility of the phonatory organs offers a rapid modification of the vocal tract, leading to the

modification of the size of the oral cavity and the pharyngeal cavity.

The pharynx is a passageway linking the oral cavity with the esophagus and the nasal cavity with the larynx. It runs vertically from the base of the skull to the level of the sixth cervical vertebra [2]. It serves as the first resonator for laryngeal sound. Regarding the role of the pharynx in speech production, it is important to note that the alterations in the size and shape of the pharyngeal cavity play an important role in determining voice quality, this is why the researchers paid particular attention to the anatomical and muscular description of the pharynx. Thus the raising or lowering of the larynx alters the length of the pharyngeal cavity which leads to acoustic consequences due to the change in the vertical dimension of the pharynx; larynx lowering results in the lowering of formant values [3].

In this study, we deal with the Arabic sound which occurs in the pharyngeal zone of the vocal tract, the pharyngealized consonant /t^ʕ/ which shows the highest degree of emphaticness [2]. Laufert and Baer cited Panconcelli-Calzia (1924, pp. 48-49) who suggested processes for the emphatic consonants: contraction of the muscles of the hyoid bone, raising of the larynx, constriction of the pharynx due to the actions of the constrictor muscles [2]. According to Catford's definition, when there are two simultaneous articulations, an articulation in the pharynx is considered the secondary one [4] (the Pharyngeal constriction generally constitutes a secondary articulation). In Arabic and other Semitic languages, pharyngealization involves a secondary articulation, it is defined as an articulation performed separately in addition to the primary articulation associated with a sound. The secondary articulation according to definition is less constricted than the primary articulation. The studies of emphatics based on experimental data, mainly from x-ray as in Ghazeli's study, he conclude that the pharyngeal sounds are articulated in the lower part of the pharynx, the uvular sounds have a constriction in the upper oropharynx and the emphatics have a secondary tongue retraction midway in the pharynx between the place of articulation of uvular and pharyngeal [5].

Giannini and Pettorino [6] examined one Arabic speaker of a Baghdadi dialect and presented acoustic and radiographic data. Their acoustic results are that for the emphatics the first and second formant F1 and F2 approach each other: F1 rises and F2 lowers. The third formant F3 is almost unchanged. They interpreted their radiographic results as showing a constriction in the pharynx for the emphatic sounds.

In summary, our main aim in this paper is to study the production of the Moroccan Arabic pharyngealized sound /t^ʕ/, its effects on adjacent vowel /a/ and we compare them with the pharyngeal consonants /ʕ/. Using cineradiography films of the vocal tract, we examined the hypothesis that both the pharyngeal and pharyngealized consonants are consistently produced with a pharyngeal constriction, and with a raising larynx and that pharyngeal constriction and larynx height are thus an articulatory features associated with these sounds, we refer to the acoustic and radiographic data to examine this issue. We also explore the association of the two consonants effect on the adjacent vowel /a/ as in /ʕta/. The methods and tools used for data processing and for measuring organ displacements are given in Section 2, the obtained results are given in Section 3 and the conclusion is given in Section 4.

II. METHODS AND TOOLS

This study based on experimental data, mainly from x-ray images of the vocal tract and acoustic data, these articulatory and acoustic data were taken from a database (DONnées Cinéradiographiques VALorisées et recherches sur la Coarticulation, Inversion et évaluation de Modèles physiques) (DOCVACIM) [7]. We have drawn the main articulator contours, particularly those of the tongue, lower and upper lips, the larynx, the glottis, the jaw, the hard palate and the hyoid bone.

A. Delineating Articulator Contours

We used software, called “X-articulators” [8], enabling several tracking tools to be used according to the nature of articulators. The “X-articulators” software has been designed to exploit automatic or semi-automatic tracking tools, it provide several tools to the exploitation of contours easier [8], and to construct articulatory models from the articulator contours via Principal Component Analysis (PCA). The main articulator are approximated by the following parameters: The lip deformations are approximated by two factors (roughly aperture and protrusion), and the lower part of the pharynx (including the larynx and the epiglottis) is represented by two factors (larynx height, glottis), the tongue by three factors (apex, tongue dorsum, tongue body) [9]. It provides also several tools to make the exploitation of contours easier, since it is important to relate contours to phonemes uttered, it is possible to import a file of phonetic annotations. Thus, before the delineation of the phonatory organs contours a phonetic annotations and Synchronization of the annotation is carried out.

The radiographic film used in this study, consists of vocal tract x-ray images of a native Moroccan Arabic adult

male speaker, and the acoustic data consists of sentences in Moroccan Arabic dialect. For this study, we processed the images that correspond to the production of Arabic vowel /a/ adjacent to pharyngealized consonant /t^ʕ/. These articulatory data allowed us to extract combinations of articulators for this phonetic type.

B. Treatment of Acoustic Data

The acoustic database consists of sentences uttering by an adult male speaker in Moroccan Arabic dialect, which is the L1 of the speaker. In this study we selected sequences that contain vowel /a/ in pharyngealized environment. We used the software Praat to segment the sentences into phonemes and also for the phonetic annotation, the length of each sentence is measured and also the duration of the entire sequence. After segmentation, we synchronized each phone with the corresponding x-ray images of the speaker vocal tract. Thus, for each phoneme we have the corresponding vocal tract x-ray images, and then we processed the vocal tract x-ray images corresponding to each phoneme.

The contours are annotated and exploited for direct measurement of the articulators’ displacement.

C. Measurement of the phonatory organs displacements

The method of measurement of the displacement of the phonatory organs is based on an angular reference. Knowing that the vocal tract configurations are different for each speaker that is the reason why we make an adapted grid for our subject. An orthonormal basis is drawn beforehand on tracing paper and used with the adapted grid to measure the articulators’ displacements [10]. This method allows the observation of the displacements of the phonatory organs, therefore it is used in the analysis of numerous data collected on different subjects. a reference image which corresponds to a rest position is chosen to reproduce on the millimeter layer, where the orthonormal reference has been drawn before, the fixed elements of the vocal tract, particularly the upper incisor and the hard palate, thus, making the upper incisor particularly reliable as a reference point. The traced part will then be superimposed on each sketch as finely as possible. All the positions of the phonatory organs are therefore calculated relative to the upper incisor. In our study, we focused on the measurements of the larynx center position, the constriction opening, the constriction location and the hyoid bone position, knowing that these components are the effective gestures for producing the pharyngeal and pharyngealized consonants.

III. RESULTS

Tables 1 summarizes the position of the hyoid bone and the larynx center, the constriction opening and location, during the production of the pharyngeal /ʕ/, the pharyngealized consonants /t^ʕ/ and during the production of the short vowel /a/ adjacent to these consonants.

We focused on the vertical movement of the larynx and the hyoid bone given by the vertical coordinate (y) relative to the rest position, all measures are carried out relative to the reference point (upper incisor). The Average Values (AV) with Standard Deviation (SD) are calculated, the

values are given in centimeters. We calculate the elevation of the larynx center and the hyoid bone relative to the rest position.

TABLE I. MEASURES REGARDING THE HYOID BONE POSITION, THE LARYNX CENTER POSITION, THE CONSTRUCTION OPENING AND CONSTRUCTION LOCATION: IN THE REST POSITION, DURING THE PRODUCTION OF THE PHARYNGEAL /ʕ/, DURING THE PHARYNGEALIZED CONSONANT /tʕ/ AND FOR THE ADJACENT VOWEL /a/.

	Hyoid Bone (HB) position (y) cm	Larynx Center (LC) position (y) cm	Constriction opening (cm)	Constriction location (cm)
Rest position	4.2266	6.1732	-	-
/ʕ/ in /ʕtʕah/	2.8111	4.6995	0.9872	14.6384
/tʕ/in /ʕtʕah/	3.0527	4.6535	0.0583	1.4485
/tʕ/in /tsutʕina/	3.3362	5.5298	0.0878	1.7827
/tʕ/in/tʕwaʕen /	3.3649	5.9345	0.0877	1.9483
AV of /tʕ/ in plain contexts	3.3505 SD(0.17)	5.7321 SD(0.654)	0.0877	1.8655
/a/ in /ʕtʕah/	3.2391	5.1861	0.5847	1.1145
/a ₁ /in /ma ₁ tʕa ₂ ruʕ/	4.1298	5.9788	1.1006	12.9673
/a ₂ / in /ma ₁ tʕa ₂ ruʕ/	3.4968	5.8149	0.5756	1.4485
/a ₁ / in /ba ₁ tʕa ₂ tʕa ₃ /	3.7440	5.8270	0.9887	13.3014
/a ₂ / in /ba ₁ tʕa ₂ tʕa ₃ /	3.4715	5.61	1.0605	0.8338
/a ₃ / in /ba ₁ tʕa ₂ tʕa ₃ /	3.7609	5.6995	0.9559	13.1343
Average values for /a/ in /tʕ/ contexts	3.6123 SD(0.2909)	5.6717 SD(0.2538)	0.8163 SD (0.2697)	1.1322
/a/ in plain contexts	3.9359 SD(0.336)	5.8118 SD(0.271)	1.2164	0.9474

For the pharyngealized /tʕ/, the larynx center position is at about 5.732cm, with SD =0.654, the larynx center rises by 0.8 cm relative to the rest position. The hyoid bone is at 3.35cm, and SD=0.17, the hyoid bone rises by 0.87 cm. The constriction location is at 1.87cm and the constriction opening is about 0.09cm.

In the context of the word /ʕtʕah/, the larynx center position is 4.65cm, it rises about 1.52 cm. The hyoid bone is at 3.05cm, it raises by 1.17 cm relative to the rest position. The constriction location is at 1.45cm and the constriction opening is 0.06cm.

During the production of the pharyngeal consonant /ʕ/, the larynx center rise by about 0.87 cm relative to the rest position and the hyoid bone raise by 0.72 cm. The constriction location is at 13.76cm and the constriction opening is 1.16cm.

What is noticed in our study is the remarkable influence of the pharyngealized consonant /tʕ/ on the articulation of /ʕ/ in the word /ʕtʕah/. If we compare the larynx center and the hyoid bone position during the production of /ʕ/ in the word /ʕtʕah/ and during the production of /ʕ/ in the word /ʕlaf/ [11], we noticed that the larynx center moved up to 4.699 cm, so it rises by 1.47 cm relative to the rest position, 0.674 cm more than in the case of /ʕlaf/ and 0.601 cm more than the average value of the larynx center position during the production of /ʕ/ in the plain coronal contexts. The hyoid bone reaches the value 2.81 cm, it rises by 1.41 cm relative to the rest position; 1.40 cm more compared to the production of /ʕ/ in the word /ʕlaf/ and 0.94 cm more than the average value of the hyoid bone position in the different studied contexts of /ʕ/ [11]. Likewise, the pharyngealized /tʕ/ is influenced by /ʕ/, where the larynx center rises by about 1cm more than in the plain coronal contexts. So, there is a mutual effect of the two consonants. Regarding the constriction location and opening, the software x-articulator gives us these two parameters as follow: The constriction is given by the point of the tongue whose distance to the palate (measured by orthogonal projection) is minimal. This distance gives the opening of the constriction. To locate the position of the constriction, the outline of the palate is used and the length of the path between the upper incisor and the place of the constriction is calculated. When there is a contact between the tongue and the palate, it is the point of the most onward constriction in the vocal tract that is chosen [8][9]. We can conclude from the obtained results that the primary articulation for /tʕ/ is located at 1.45 to 1.87 cm relative to the upper incisor and the constriction opening is about 0.06 to 0.09 cm. the secondary constriction is located at 12.96 to 13.3 cm and the constriction opening is 0.98 to 1.1 cm. we noticed that the primary constriction is more constricted than the secondary one and the location of the secondary constriction for the pharyngealized /tʕ/ is close to that of the pharyngeal /ʕ, h/ which are at about 13.76 to 14.63 cm. the difference is 0.8 cm to 2.16 cm.

In the second part of this section, we explore the articulatory effects of these consonants on the adjacent vowel /a/. The larynx center position and the hyoid bone positions are measured during the production of the vowel /a/ in plain coronal contexts, the average value of the larynx center position is 5.81 cm, and the value of SD is 0.27, the larynx center rises by 0.36 cm and the hyoid bone is at 3.935 cm, and SD =0.3, the hyoid bone rises by 0.29 cm. In the pharyngealized environment /tʕ/, the larynx center rises by 0.50 cm relative to the rest position, it rises 0.14 cm

more compared to that measured in plain coronal context. The hyoid bone rises by 0.61 cm (0.32 cm more).

As mentioned earlier, the acoustic output varies according to the behavior of the active gestures and the obtained results from the articulatory study lead to additional questions about the acoustic consequences of change in the vertical position of the larynx on vowel adjacent to the pharyngealized /t^ʕ/. An acoustic study of the vowel quality was conducted in order to determine the extent to which vowels could be affected by that environment. The software Praat is used for the segmentation of the words into phonemes and for the measurement of the formants values. The table II summarizes the formants values of the vowel /a/ in pharyngealized neighboring, in plain coronal contexts and in the voiced pharyngeal neighboring.

The formants are explored for our subject: For the vowel /a/ in plain coronal environment, the average value of F1 is 519.60 Hz with standard deviation SD equal to 46.46, the value of F2 is 1573.437 Hz, and SD =94.37, F3 is equal to 2485.809 Hz, with SD =137.43, the value of F4 is 3676.291, and SD =95.33.

In /t^ʕ/ neighboring, F1 increases by 138.52 Hz, F2 decreases by -314.15 Hz, F3 and F4 undergo a moderate increase of 52.33 Hz and 23.88 Hz respectively. In the case of /ʕt^ʕa/, F1 increases by 198.98 Hz, F2 decreases by -232.48 Hz, F3 remain stable, and F4 increases by 123.72 Hz.

In this study, we compare the effects of the pharyngealized consonant / t^ʕ/ on adjacent vowel /a/with that of the pharyngeal /ʕ/. The effects of the pharyngealized /t^ʕ/ and the pharyngeal /ʕ/on the formants values of /a / are: F1 increases by about 134.79Hz in /ʕ/ environment, and by 138.52 Hz in /t^ʕ/ environment, the effect of the pharyngeal /ʕ/ is similar to that of the pharyngealized/t^ʕ/and in the context of /ʕt^ʕah/, F1 increases by about 198.98 Hz. F2 in the environment of /ʕ/ increases by about 132.78 Hz, in the environment of /t^ʕ/, it decreases by about -314.15 Hz. In the word /ʕt^ʕah/, it decreases by -232.48 Hz. This result agrees with the founding of Al-Ani [12], he found a considerable drop of F2 in vowels following pharyngealized consonants compared to plain consonants. Ghazeli also reported that all vowels have a lower F2 after pharyngealized consonants [5]. We noticed that /ʕ/ decreases the effect of /t^ʕ/ by about 81.67 Hz. F3 is less influenced by both /t^ʕ/ and /ʕ/, so in the context of the word /ʕt^ʕah/, it remains stable. F4 increases by 91.89 Hz in the environment of /ʕ/, by 23.88 Hz in the environment of /t^ʕ/, and it increases by 123.72 Hz in the word /ʕt^ʕah/. F4 is less influenced by /t^ʕ/ and a little more by /ʕ/, so in /ʕt^ʕah/, the effects of /ʕ/ and /t^ʕ/ are superimposed.

Comparing the articulatory and the acoustic results, we noticed a correlation between the degrees of the articulatory gestures spreading on the adjacent vowel /a/with the acoustic consequences. Thus, in the word /ʕt^ʕah/, the pharyngeal /ʕ/ adjacent to the pharyngealized /t^ʕ/ and both adjacent to the vowel /a/, the larynx center rises during the

production of /a/ at 0.987 cm, it is increased by 45.08% relative to the context /ʕa/ (0.542 cm) and by 40.52% relative to the context /t^ʕa/ (0.587 cm) [11].

TABLE II. FORMANTS VALUES OF THE SHORT VOWEL/A/ IN DIFFERENT CONTEXTS: (/ʕ / AND /T^ʕ/) NEIGHBORING AND IN PLAIN CONTEXTS

Formants /	F1(Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)
/a/ in /ʕ/and /t ^ʕ / neighboring				
/a/ in /ʕt ^ʕ ah/	718.58	1340.96	2487.67	3800.35
/a ₁ / in/ma ₁ t ^ʕ a ₂ ru ₁ /	659.81	1278.10	2601.01	3643.81
/a ₁ /in/ba ₁ t ^ʕ a ₂ t ^ʕ a ₃ /	674.1	1275.05	2471.15	3658.03
/a ₂ /in /ma ₁ t ^ʕ a ₂ ru ₁ /	649.26	1256.64	2489.54	3674.53
/a ₂ /in/ba ₁ t ^ʕ a ₂ t ^ʕ a ₃ /	652.2	1359.74	2548.24	3805.53
/a ₃ /in/ba ₁ t ^ʕ a ₂ t ^ʕ a ₃ /	655.24	1126.95	2580.75	3718.96
Average value of /a/ formants in /t ^ʕ / neighboring	658.12	1259.29	2538.13	3700.17
/a/ in /ʕ/ neighbouring	654.39	1706.22	1051.83	2539.67
/a/ in plain neighboring	440.50	1418.58	2566.69	3794.06
/a/ in / t ^ʕ / (%)	39.49%	-11.22%	-1.1%	-2.47%
/a/ in /ʕ/ (%)	48.55%	20.27%	-59%	-33%
/a/ in /ʕt ^ʕ ah/ (%)	63.13 %	-7.47%	-3.07%	0.16%

The value of F1 increased by about 63.13%, in the context of the word /ʕt^ʕah/ compared to plain coronal context. And it is increased by about 23.64% compared to/t^ʕ/ neighboring and by 14.64% compared to /ʕ/ neighboring. The second formant F2 drops considerably in /t^ʕ/ neighboring by about 11.22%, the effect of /t^ʕ/ on F2 is decreased in the word /ʕt^ʕah/ (7.47%), the consonant / ʕ/ halves the effect of /t^ʕ/. F3 is less influenced by /t^ʕ/ also in the word /ʕt^ʕah/, the variation does not exceed 3%. Likewise for the fourth formant F4 the variation is at about 0.16%.

IV. CONCLUSION

In this study, the articulatory and acoustic aspects of the production of the pharyngealized consonant /t^s/are explored. We compared these aspects with those of the pharyngeal consonant /ʕ/. The effects of the association of the pharyngeal consonant with the pharyngealized one as in the word /ʕt^sah/ on the adjacent vowel /a/are investigated.

We noticed that the raising of the larynx center and the hyoid bone is in the same range during the production of both /ʕ/ and /t^s/ (about 0.8 cm), and F1 increases by 48.5% in /ʕ/ neighboring and by about 39.49% in /t^s/ neighboring. The production of the two consonants successively /ʕt^s/ in the word /ʕt^sah/ the larynx center and the hyoid bone rise doubly by about 1.5 cm. So, we observe that the gesture is superimposed and the acoustic effect on F1 also is raised, we noticed a correlation between the articulatory effect and the value of the first formant F1.

In summary, /ʕ/ and /t^s/have a similar effect on F1 in CV context and in CCV context F1 undergoes a moderate increase compared to CV context about 14% to 23%. F2 increases by 132.78 Hz in /ʕ/ neighboring and drops by -314.15 Hz in /t^s/ neighboring, in /ʕt^sah/ it drops by -232.48 Hz, so the association of the two types of consonants reduced their effects on F2 and the effect of /t^s/ is dominant. /ʕ/ decreases the effect of /t^s/ on F2 by about 81.67 Hz in the word /ʕt^s ah/.

F3 is less influenced by both /t^s/ and /ʕ/, F4 is not influenced by the pharyngealized /t^s/.

ACKNOWLEDGMENT

This paper and its associated research would not have been possible without the exceptional support of Dr. Yves Laprie, who welcomed us in the research laboratory "Laboratoire Lorrain de Recherche en Informatique et ses Applications LORIA", in Henri-Poincaré University of Nancy, France. We thank him for his assistance and for providing the data with their processing tools. His knowledge and exacting attention to detail have been an inspiration and kept my work on track.

REFERENCES

- [1] M. Stone and C. H. Shadle, "A History of Speech Production Research", *Acoustics Today*, Winter 2016, volume 12, issue 4, pp.48-55.
- [2] A. Laufer and T. Baer, "The Emphatic and Pharyngeal Sounds in Hebrew and in Arabic", *Language And Speech*, 31(2), pp.181-205,1988, doi: 10.1177/002383098803100205.
- [3] A. Marchal, "From Speech Physiology to Linguistic Phonetics", John Wiley& Sons, 2010. ISBN, 0470610409, 978047610404.
- [4] J. C. Catford, "Fundamental problems in phonetics", Bloomington, IN: Indiana University Press, 1977.
- [5] S. Ghazeli, "Back Consonants and Backing Coarticulation in Arabic", Ph.D. Dissertation, University of Texas, Austin, 1977.
- [6] A. Giannini and M. Pettorino, "The emphatic consonants in Arabic", *Speech Laboratory Report*, 4,1982.
- [7] R. Sock, F. Hirsch, Y. Laprie, P. Perrier and B.Vaxelaire, B, "An X-ray Database, Tools and Procedures for the Study of Speech Production", 9th International Seminar on Speech Production (ISSP 2011), Montreal, Canada. pp.41-48, 2011. <hal 00610297>.
- [8] Y. Laprie, R. Sock, B. Vaxelaire and B. Elie" Comment faire parler les images aux rayons X du conduit voca ?!". SHS Web of Conferences, EDP Sciences, 2014, 4e Congrès Mondial de Linguistique Française, vol. 8, pp.14. <10.1051/shsconf/20140801344>. <hal- 01059887>
- [9] Y. Laprie and J. Busset, "Construction and Evaluation of an Articulatory Model of the Vocal Tract". In 19th European Signal Processing Conference (EUSIPCO), 2011.
- [10] B. Fayssal, "la Gémiation en Tarifit, Considérations Phonologiques, Etude Acoustique et Articulatoire", PhD thesis université de Strasbourg, 2014. URL : <https://theses.hal.science/tel-01162892/document>.
- [11] F. Karaoui, A. Djéradi and Y. Laprie, "The Articulatory and acoustics Effects of Pharyngeal Consonants on Adjacent Vowels in Arabic Language", *Proceedings of The Fourth International Conference on Natural Language and Speech Processing* , 12--13 November, Trento, Italy. pp. 272-279, 2021. URL: <https://aclanthology.org/2021.icnlspp-1.32>
- [12] S. H. Al-Ani, "Arabic phonology: An acoustical and physiological investigation", Paris: The Hague: Mouton, 1970, doi: <https://doi.org/10.1515/9783110878769>

On the Performance of a Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation

Ionuț-Dorinel Fîciu*, Cristian-Lucian Stanciu*, Constantin Paleologu*, and Felix Albu†

*Department of Telecommunications, University Politehnica of Bucharest, Romania

† Department of Electronics, Valahia University of Târgoviște, Romania

Emails: ionut.ficiu22@gmail.com, cristian@comm.pub.ro, pale@comm.pub.ro, felix.albu@valahia.ro

Abstract—The Stereophonic Acoustic Echo Cancellation (SAEC) setup implies the estimation of four loudspeaker-to-microphone unknown impulse responses, which generate unwanted acoustic replicas (i.e., echoes). In this context, some state-of-the-art approaches that combine the Widely Linear (WL) model with various versions of the Recursive Least-Squares (RLS) algorithm have been recently proposed. This paper focuses on the most recent one – the version that uses a Data-Reuse (DR) approach over the WL-RLS based on Dichotomous Coordinate Descent (DCD) iterations, i.e., the namely WL-DR-RLS-DCD. The target of the paper is to present how this algorithm behaves in various conditions, for input signals that present a very high correlation, such as speech or Auto-Regressive (AR) sequences. Simulations results proved that the DR approach is suitable in the SAEC context.

Index Terms—Stereophonic Acoustic Echo Cancellation (SAEC); Recursive Least-Squares (RLS) algorithm; Data-Reuse (DR); Dichotomous Coordinate Descent (DCD); Widely Linear (WL) model.

I. INTRODUCTION

Stereophonic communication creates the sensation of audio directionality by employing for each terminal two loudspeakers and two microphones, respectively [1]–[3]. When performing the Stereophonic Acoustic Echo Cancellation (SAEC), a total number of four echo paths must be estimated in order to mitigate the echo effect produced by any of the four associated loudspeaker-to-microphone pairs. The standard approach for such scenarios is the implementation of four individual adaptive systems, usually based on the Least-Mean-Squares (LMS) family of algorithms [4]–[6]. However, despite their low arithmetic costs, the LMS methods provide limited performance when working with highly correlated input signals (such as speech sequences) due to gradient noise [4].

In [7]–[9], the recursive least-squares (RLS) algorithm [4] was combined with the dichotomous coordinate descent (DCD) iterations [10], and employed in SAEC scenarios in order to improve performance, especially in terms of convergence rate and computational complexity. The widely linear (WL) model described in [1][8], was used as a framework in order to improve the handling of the system, by grouping the four adaptive filters working with real valued variables into a single adaptive system working with fewer complex valued variables (CRVs). The resulting WL-RLS-DCD method has acceptable arithmetic workloads and performance levels

comparable with other consecrated RLS versions, such as the one based on Woodburry’s identity [4].

This paper analyzes the performance of an improved version of the WL-RLS-DCD algorithm designed for superior tracking capabilities, which requires minimal extra arithmetic costs. A Data-Reuse (DR) approach for the WL-RLS-DCD adaptive method will be described, which re-uses the same input data multiples times for each of the adaptive filter’s iterations. An analysis will be performed in several scenarios in order to demonstrate the capabilities of the new WL-DR-RLS-DCD method, and several other aspects will be discussed.

This paper is organized as follows. Section II describes the mathematical model of the WL framework for RLS adaptive systems. Section III presents the WL-DR-RLS-DCD algorithm, which is suitable for improving its corresponding tracking speeds. Section IV discusses simulation results for the proposed algorithm in the SAEC setup, and the paper is concluded by a few conclusions in Section V.

II. SYSTEM MODEL

In the SAEC configuration, for each discrete-time index n , we store the last L samples corresponding to $x_{L_c}(n)$ and $x_{R_c}(n)$ (i.e., the left and right channels) in the $L \times 1$ vectors denoted by $\mathbf{x}_{L_c}(n)$, respectively $\mathbf{x}_{R_c}(n)$ [1][8]. Furthermore, the echo contributions associated with the stereo channels can be obtained using the input vectors and the four possible $L \times 1$ echo path impulse responses denoted by $\mathbf{g}_{t,L_c L_c}$, $\mathbf{g}_{t,L_c R_c}$, $\mathbf{g}_{t,R_c L_c}$, and $\mathbf{g}_{t,R_c R_c}$ [8][9]:

$$y_{L_c}(n) = \mathbf{g}_{t,L_c L_c}^T \mathbf{x}_{L_c}(n) + \mathbf{g}_{t,R_c L_c}^T \mathbf{x}_{R_c}(n), \quad (1)$$

$$y_{R_c}(n) = \mathbf{g}_{t,L_c R_c}^T \mathbf{x}_{L_c}(n) + \mathbf{g}_{t,R_c R_c}^T \mathbf{x}_{R_c}(n), \quad (2)$$

where T represents the transpose operator. As a consequence, the microphone (or reference) signals can be written as

$$d_{L_c}(n) = y_{L_c}(n) + w_{L_c}(n), \quad (3)$$

$$d_{R_c}(n) = y_{R_c}(n) + w_{R_c}(n), \quad (4)$$

with $w_{L_c}(n)$ and $w_{R_c}(n)$ being environmental noise signals.

When we apply the WL model, we can use $j = \sqrt{-1}$ to combine the inputs and the outputs of the SAEC unknown echo paths into the complex input signal

$$x(n) = x_{L_c}(n) + jx_{R_c}(n), \quad (5)$$

respectively the complex output

$$y(n) = y_{L_c}(n) + jy_{R_c}(n). \quad (6)$$

When applying this approach for the vectors $\mathbf{x}_{L_c}(n)$ and $\mathbf{x}_{R_c}(n)$, then interleaving the resulting complex valued $L \times 1$ vector with its complex conjugate version, a $2L \times 1$ input vector $\tilde{\mathbf{x}}(n)$ is obtained [1][8][9]. Similarly, the unknown echo paths can be linearly combined and interleaved in order to generate a single $2L \times 1$ complex valued unknown system denoted as $\tilde{\mathbf{g}}_t$ [1][8][9].

Furthermore, the complex valued output can be written as

$$y(n) = \tilde{\mathbf{g}}_t^H \tilde{\mathbf{x}}(n), \quad (7)$$

where H denotes the Hermitian operator.

We can write the complex valued noise signal

$$w(n) = w_{L_c}(n) + jw_{R_c}(n), \quad (8)$$

and also the expression for the complex reference as

$$d(n) = y(n) + w(n). \quad (9)$$

Thus, we express the complex error as

$$e(n) = d(n) - \tilde{y}(n), \quad (10)$$

where $\tilde{y}(n)$ is the estimate of the complex valued system's output determined with the adaptive filter's set of coefficients $\tilde{\mathbf{g}}(n)$, which is employed to approximate $\tilde{\mathbf{g}}_t$.

The WL framework improves the handling of SAEC applications using adaptive algorithms. It re-casts four unknown system identification problems into a single one. The two-input/two-output system with real random variables is expressed as a single-input/single-output system with CRVs.

III. THE WL-DR-RLS-DCD ALGORITHM

The process of generating the estimate $\tilde{\mathbf{g}}(n)$ can be approached in SAEC scenarios using RLS filters due to their increased performance when working with highly correlated signals, such as speech. In [8][10], the complex valued leading DCD was employed in combination with the RLS algorithm in order to decrease the corresponding arithmetic complexity and mitigate their prohibitive nature when approaching hardware implementations. The DCD requires only additions and bit-shifts to solve an auxiliary system of equations, by exploiting the properties of the input signal [8][10].

The RLS-DCD method applied to the WL setup (i.e., the WL-RLS-DCD) is presented in Table I, where we denoted by ϵ a small positive constant employed to initialize the correlation matrix in a non-singular form, and by $\mathbf{R}^{(:,1)}(n)$ the first column of the matrix $\mathbf{R}(n)$. Steps 1 and 2 update the complex input vector $\tilde{\mathbf{x}}(n)$, respectively the $2L \times 2L$ correlation matrix estimate $\mathbf{R}(n)$, using the time-shift property of the input signal and the forgetting factor λ ($0 < \lambda \leq 1$), which determines the memory of the algorithm [1][9]. After computing the filter output and the corresponding error in steps 3 and 4, the residual component $\mathbf{p}_0(n)$ is updated in step 5 using the residual vector $\mathbf{r}(n-1)$ [8][10]. Consequently, the

DCD iterations are employed in step 6 in order to solve an auxiliary system of equations and generate the $2L \times 1$ complex valued solution vector $\Delta\tilde{\mathbf{g}}(n)$, and an updated version of $\mathbf{r}(n)$. In the final step of the WL-RLS-DCD (i.e., step 7), the solution vector is used to update the filter coefficients $\tilde{\mathbf{g}}(n)$. The overall complexity of the algorithm is proportional to the value $2L$ multiplied by a small factor, which makes it attractive for hardware implementations.

Table I. WL-RLS-DCD ALGORITHM

Step	Action
Init.	Set: $\tilde{\mathbf{g}}(0) = \mathbf{0}_{2L \times 1}$; $\mathbf{r}(0) = \mathbf{0}_{2L \times 1}$ $\mathbf{R}(0) = \epsilon \mathbf{I}_{2L}$, $\epsilon > 0$
For $n = 1, 2, \dots$, number of iterations :	
1	Update $\tilde{\mathbf{x}}(n)$
2	Update $\mathbf{R}(n)$ using time-shift $\mathbf{R}^{(:,1)}(n) = \lambda \mathbf{R}^{(:,1)}(n-1) + \mathbf{x}^*(n)\tilde{\mathbf{x}}(n)$
3	$\tilde{y}(n) = \tilde{\mathbf{g}}^H(n-1)\tilde{\mathbf{x}}(n)$
4	$e(n) = d(n) - \tilde{y}(n)$
5	$\mathbf{p}_0(n) = \lambda \mathbf{r}(n-1) + e^*(n)\tilde{\mathbf{x}}(n)$
6	$\mathbf{R}(n)\Delta\tilde{\mathbf{g}}(n) = \mathbf{p}_0(n) \xrightarrow{\text{DCD}} \Delta\tilde{\mathbf{g}}(n), \mathbf{r}(n)$
7	$\tilde{\mathbf{g}}(n) = \tilde{\mathbf{g}}(n-1) + \Delta\tilde{\mathbf{g}}(n)$

The tracking speed of the WL-RLS-DCD can be improved using the DR approach [9][11][12], which employs the same input data multiple times per each time index n . The DCD method is run for a number of N_{it} iterations (i.e., for $\phi = 0 \dots N_{it} - 1$), and for each of the solution vectors $\Delta\tilde{\mathbf{g}}_\phi(n)$, the output signal estimate changes to $\tilde{y}_\phi(n)$ [9] (i.e., it is updated accordingly):

$$\tilde{y}_\phi(n) = \begin{cases} \tilde{\mathbf{g}}^H(n-1)\tilde{\mathbf{x}}(n), & \phi = 0, \\ \left[\tilde{\mathbf{g}}(n-1) + \sum_{k=0}^{\phi-1} \Delta\tilde{\mathbf{g}}_k(n) \right]^H \tilde{\mathbf{x}}(n), & \phi > 0. \end{cases} \quad (11)$$

It can be noticed that only the first branch of (11) is used when the new algorithm has the parameter $N_{it} = 1$. In this case, it is equivalent to the WL-RLS-DCD and only the filter coefficients from the previous time index (i.e., $n-1$) are required for any computations at time index n . However, if $N_{it} > 1$, then for any DR iteration corresponding to $\phi > 1$, the values corresponding to the filter taps have to be updated using the solution vector $\Delta\tilde{\mathbf{g}}_{\phi-1}(n)$ from the previous DR step (also, from the same time index). In a similar manner, the value of the error signal is adjusted to reflect the change in the filter coefficients [9], and to also take advantage of the results available at previous iterations:

$$e_\phi(n) = \begin{cases} d(n) - \tilde{\mathbf{g}}^H(n-1)\tilde{\mathbf{x}}(n) \triangleq e_0(n), & \phi = 0 \\ e_{\phi-1}(n) + \Delta\tilde{\mathbf{g}}_{\phi-1}^H(n)\tilde{\mathbf{x}}(n), & \phi > 0. \end{cases} \quad (12)$$

Furthermore, in order to run the DCD for each DR step, the residual contributions must be updated and can be expressed as [9]

$$\mathbf{p}_{0,\phi}(n) = \begin{cases} \lambda \mathbf{r}_{N_{it}-1}(n-1) + e_0^*(n)\tilde{\mathbf{x}}(n), & \phi = 0 \\ \mathbf{r}_{\phi-1}(n) + e_0^*(n)\tilde{\mathbf{x}}(n), & \phi > 0. \end{cases} \quad (13)$$

The first branch of (13) represents the classical transition from the previous time index, while the second branch performs the transition from the previous DR iteration. A common conclusion is also available for (11), (12), respectively (13) when considering $N_{it} = 1$: all the update expressions are simplified such that the form of the proposed algorithm reverts back to the steps corresponding to the WL-RLS-DCD method.

Table II. WL-DR-RLS-DCD ALGORITHM

Step	Action
Init.	Set: $\tilde{\mathbf{g}}(0) = \mathbf{0}_{2L \times 1}$; $\mathbf{r}(0) = \mathbf{0}_{2L \times 1}$ $\mathbf{R}(0) = \epsilon \mathbf{I}_{2L}$, $\epsilon > 0$, $\phi = 0$
For $n = 1, 2, \dots$, number of iterations :	
1	Update $\tilde{\mathbf{x}}(n)$
2	Update $\mathbf{R}(n)$ using time-shift $\mathbf{R}^{(:,1)}(n) = \lambda \mathbf{R}^{(:,1)}(n-1) + x^*(n)\tilde{\mathbf{x}}(n)$
3	$\phi = \phi + 1$
4	Determine $e_\phi(n)$ using (12)
5	Determine $\mathbf{p}_{0,\phi}(n)$ using (13)
6	$\mathbf{R}(n)\Delta\mathbf{g}_\phi(n) = \mathbf{p}_{0,\phi}(n) \xrightarrow{\text{DCD}} \Delta\tilde{\mathbf{g}}_\phi(n), \mathbf{r}_\phi(n)$
7	Determine $\tilde{\mathbf{g}}_\phi(n)$ using $\Delta\tilde{\mathbf{g}}_\phi(n)$ If $\phi < N_{it} \xrightarrow{\text{jump}}$ step 3

The resulting adaptive algorithm, namely the WL-DR-RLS-DCD, is presented in Table II. It is expected to generate improved tracking capabilities with the cost of some accuracy when the adaptive system reaches the steady-state. Moreover, the overall arithmetic effort remains proportional to the value $2L$ multiplied by a small number, which includes the contribution of N_{it} . Simulations will demonstrate that a value of N_{it} much smaller than 10 is sufficient in order to generate satisfactory performance.

IV. SIMULATION RESULTS

Simulations were performed for the SAEC context working with the WL framework. Two types of input signal sources were employed: a Gaussian noise filtered through an autoregressive (AR) system with a single pole, and a high quality speech sequence [13]. Both types are highly correlated, which challenges the convergence of the algorithms. These inputs were filtered through two distinct real acoustic impulse responses in order to achieve the effect of the *left* and *right* channels and generate the samples $x_{L_c}(n)$ and $x_{R_c}(n)$. For each acoustic channel, Gaussian noise was added to the echo signals having the signal-to-noise ratio (SNR) experimentally set to 30 dB.

The pre-distortion approach presented in [1][7][14] was also used in order to avoid the problem of the correlation between the $x_{L_c}(n)$ and $x_{R_c}(n)$ sequences. The method performs the addition of a certain level of nonlinearity to these two signals in order to obtain a unique solution and it is controlled by a parameter denoted by α_r (with $0 < \alpha_r < 1$). Consequently, starting with the so-called *half-wave rectifier* method, we can generate 2 new signals using the expressions [1][14]:

$$\mathbf{x}'_{L_c}(n) = \mathbf{x}_{L_c}(n) + \alpha_r \frac{\mathbf{x}_{L_c}(n) + |\mathbf{x}_{L_c}(n)|}{2}, \quad (14)$$

$$\mathbf{x}'_{R_c}(n) = \mathbf{x}_{R_c}(n) + \alpha_r \frac{\mathbf{x}_{R_c}(n) - |\mathbf{x}_{R_c}(n)|}{2}. \quad (15)$$

From a practical point of view, the stereo effect is not impacted if $0 < \alpha_r \leq 0.5$.

In all analyzed cases, the RLS-DCD specific parameters were chosen as $\lambda = 1 - 1/(KL)$, $H = 1$, $N_u = 4$, and $M_b = 16$. The initial convergence is not shown, as it is not considered relevant [15]. The normalized misalignment has been chosen as a performance measure:

$$\text{Mis}(n) = 20 \log_{10} \frac{\|\tilde{\mathbf{g}}_t - \tilde{\mathbf{g}}(n)\|_2}{\|\tilde{\mathbf{g}}_t\|_2} \quad [\text{dB}], \quad (16)$$

with $\|\cdot\|_2$ denoting the ℓ_2 norm [1].

In Figure 1, the input signal is a Gaussian noise filtered through an AR(1) system with the pole 0.9. For this scenario, the signs corresponding to the filter coefficients associated with the four acoustic impulse responses were suddenly changed after the adaptive system has reached steady-state. It can be noticed that, for the same values of α_r , the tracking

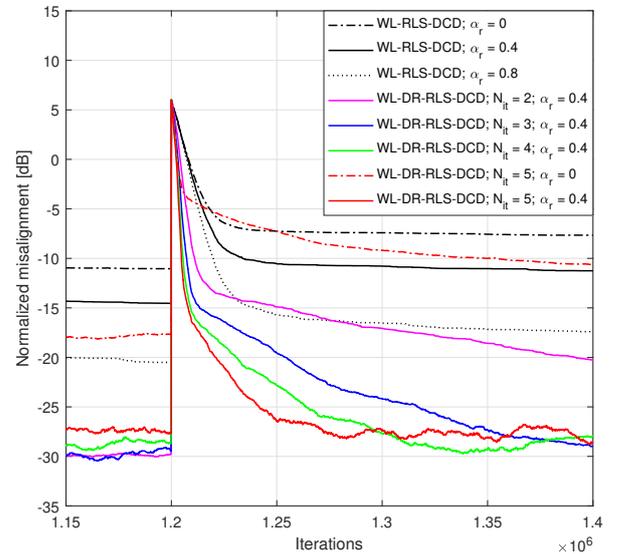


Figure 1. Performance of the WL-DR-RLS and WL-DR-RLS-DCD algorithms for different values of N_{it} and α_r . The input signal is an AR(1) sequence with the pole 0.9, and the length of the four unknown impulse responses is $L = 128$ and $\lambda = 1 - 1/(64L)$. The unknown system changes at time index 1200000.

speed is improved with the increase of N_{it} . When the input signals are not pre-distorted, the performance values are the lowest, even when using $N_{it} = 5$ DR iterations. For the value $\alpha_r = 0$, despite having good initial convergence, the WL-DR-RLS-DCD with $N_{it} = 5$, does not perform well when reaching steady-state. Moreover, when trying to compensate the extra DR iterations using $\alpha_r = 0.8$ (exceeding the recommended limit of 0.5), the tracking capabilities of the WL-RLS-DCD (i.e., the WL-DR-RLS-DCD with $N_{it} = 1$) algorithm are weaker than the ones obtained via the DR variants with $\alpha_r = 0.4$. Finally, taking into account that an additional workload is introduced by employing multiple DR

iterations, respectively that a performance cap is reached in terms of tracking capabilities for higher values of N_{it} , a value of $N_{it} = 3$ DR iterations can be considered sufficient for attaining the desired effect.

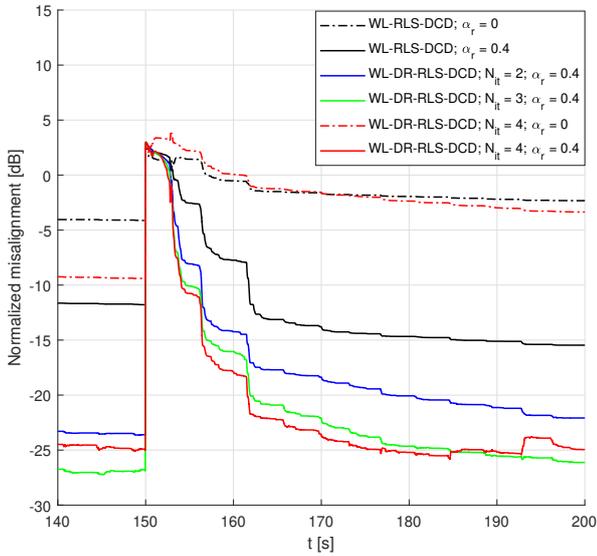


Figure 2. Performance of the WL-DR-RLS and WL-DR-RLS-DCD algorithms for different values of N_{it} and α_r . The input signal is a high quality speech sequence, and the length of the four unknown impulse responses is $L = 256$ and $\lambda = 1 - 1/(96L)$. The unknown system changes at $t = 150$ s due to the interchanging of the microphones positions.

In Figure 2, the input signal is a high quality speech sequence [13]. The four echo paths are suddenly changed at steady-state to:

$$\begin{cases} \mathbf{g}_{t,L_c L_c} = \mathbf{g}_{t,L_c R_c}, & \text{and} & \begin{cases} \mathbf{g}_{t,L_c R_c} = \mathbf{g}_{t,L_c L_c}, \\ \mathbf{g}_{t,R_c L_c} = \mathbf{g}_{t,R_c R_c}, \end{cases} \end{cases} \quad (17)$$

The adjustment is equivalent to interchanging the microphones positions from the WL model for the SAEC context. The displayed results show the performance of the WL-DR-RLS-DCD for different values of N_{it} and $\alpha_r = 0.4$. As a reference, results were also shown for $\alpha_r = 0$. As the value of the DR parameter increases, so do the tracking capabilities of the WL-DR-RLS-DCD, until a certain performance cap is revealed. It can also be noticed that the absence of the pre-processing of the input signal leads to a noticeable performance gap between the cases with $\alpha_r = 0$ and $\alpha_r = 0.4$.

V. CONCLUSIONS AND FUTURE WORK

The algorithm analyzed in this paper is an enhanced version of the combination between the exponentially weighted RLS algorithm and the DCD method [8][10]. The WL-DR-RLS-DCD is designed to approach the SAEC scenarios by exploiting the properties of the input signal and solve an auxiliary system of equations using only additions and bit-shifts. The overall complexity is proportional to the adaptive filter's length multiplied by a small number.

Simulation results were presented using Gaussian noise filtered through an AR(1) system, respectively a speech sequence, as inputs. The results have shown that, with respect to the WL-RLS-DCD, the WL-DR-RLS-DCD has a corresponding compromise between superior tracking capabilities on one side, respectively estimation accuracy at steady-state and a slight increase in arithmetic complexity on the other side. The performance trade-off is controlled from the DR portion of the algorithm through the number of iterations parameter N_{it} .

Considering the performance demonstrated by the WL-DR-RLS-DCD in simulations, the algorithm is attractive for hardware implementations in system identification scenarios, such as the SAEC configurations. A generalization to the multichannel case will be also considered.

ACKNOWLEDGEMENT

This work was supported by two grants of the Ministry of Research, Innovation and Digitization, CNCS-UEFISCDI, projects PN-III-P4-PCE-2021-0780 and PN-III-P4-PCE-2021-0438, within PNCDI III.

REFERENCES

- [1] J. Benesty, C. Paleologu, T. Gaensler, and S. Ciochina, *A Perspective on Stereophonic Acoustic Echo Cancellation*. Germany: Springer-Verlag, 01 2011.
- [2] M. Sondhi, D. Morgan, and J. Hall, "Stereophonic acoustic echo cancellation-an overview of the fundamental problem," *IEEE Signal Processing Letters*, vol. 2, no. 8, pp. 148–151, 1995.
- [3] J. Benesty, D. Morgan, and M. Sondhi, "A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 156–165, 1998.
- [4] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall information and system sciences series, Prentice Hall, 2002.
- [5] A. Sayed, *Adaptive Filters*. Wiley, New York, NY, 2008.
- [6] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Fourth Edition, New York, NY, USA: Springer-Verlag, 2013.
- [7] C. Stanciu, C. Paleologu, J. Benesty, T. Gaensler, and S. Ciochina, "An efficient RLS algorithm for stereophonic acoustic echo cancellation with the widely linear model," *41st International Congress and Exposition on Noise Control Engineering, INTERNOISE*, vol. 9, pp. 7756–7767, 01 2012.
- [8] C. Stanciu, J. Benesty, C. Paleologu, T. Gänslar, and S. Ciochină, "A widely linear model for stereophonic acoustic echo cancellation," *Signal Processing*, vol. 93, no. 2, pp. 511–516, 2013.
- [9] I.-D. Ficiu, C.-L. Stanciu, C. Paleologu, and J. Benesty, "Low-Complexity Data-Reuse RLS Algorithm for Stereophonic Acoustic Echo Cancellation," *Applied Sciences*, vol. 13, no. 4, 2023.
- [10] Y. V. Zakharov, G. P. White, and J. Liu, "Low-Complexity RLS Algorithms Using Dichotomous Coordinate Descent Iterations," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3150–3161, 2008.
- [11] C. Paleologu, J. Benesty, and S. Ciochină, "Data-Reuse Recursive Least-Squares Algorithms," *IEEE Signal Processing Letters*, vol. 29, pp. 752–756, 2022.
- [12] I.-D. Ficiu, C.-L. Stanciu, C. Elisei-Iliescu, C. Anghel, and R.-M. Udrea, "Low-Complexity Implementation of a Data-Reuse RLS Algorithm," in *2022 45th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 289–293, 2022.
- [13] "The Open Speech Repository." <https://www.itu.int/rec/T-REC-G.168/en>. Accessed: 2023-01-26.
- [14] C. Stanciu, J. Benesty, C. Paleologu, T. Gaensler, and S. Ciochină, "A novel perspective on stereophonic acoustic echo cancellation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 25–28, 2012.
- [15] Y. V. Zakharov and V. H. Nascimento, "DCD-RLS Adaptive Filters With Penalties for Sparse Identification," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3198–3213, 2013.

Comparative Performance of TCP and MQTT

Bishal Thapa

Ingram School of Engineering
Texas State University
San Marcos, TX USA
email: b_t220@txstate.edu

Bishal Sharma

Ingram School of Engineering
Texas State University
San Marcos, TX USA
email: dxa6@txstate.edu

Stan McClellan

Ingram School of Engineering
Texas State University
San Marcos, TX USA
email: stan.mcclellan@txstate.edu

Abstract— This paper compares the performance of conventional Transmission Control Protocol (TCP) with the popular Message Queueing Telemetry Transport (MQTT) protocol in private and public network settings. Higher-layer protocols, such as MQTT may be problematic for important constraints in some Internet of Things (IoT) scenarios, whereas a simpler "bare socket" TCP may be sufficient for certain scenarios. This research examines the network performance of these protocols by analyzing goodput and transmission time for a specific scenario with data security embedded at the application layer. Other features that may be suitable to be used in IoT applications are also discussed, along with their shortcomings.

Keywords—TCP/IP; MQTT; TCP Sockets.

I. INTRODUCTION

Communication between one or more devices in a network is possible due to a specific set of protocols. The use of appropriate protocols can improve application performance or can impose extra burdens for processing and transmitting overhead data in addition to application information. Different protocols function in various layers of the familiar Open Systems Interconnection (OSI) 7-layer model [1]. The Transmission Control Protocol / Internet Protocol (TCP/IP) suite is the collection of protocols used for communication between Internet-connected devices.

The TCP transport protocol (OSI layer 4) is a connection-oriented protocol that guarantees delivery of IP packets. TCP segments the data packets received from the network layer and sends them in an ordered sequence using a specified port number [2]. Many common application layer protocols rely on TCP for guaranteed data transmission. TCP is typically used in well-known client-server scenarios, and is a critical part of most Internet communications.

MQTT is a lightweight messaging protocol (OSI layers 5-7) designed for constrained devices with low bandwidth and high latency [3]. It is widely used in IoT applications, where devices communicate with each other via cloud-based servers. In MQTT, messages are organized into topics, which describe the content of the message. Clients can subscribe to one or more topics to receive messages, or they can publish messages to one or more topics. The protocol is intended to be efficient, with minimal overhead and support for offline messaging [4]. The MQTT communication structure depends on an intervening server or "broker" to distribute messages via a publish/subscribe structure.

Many messaging protocols have been used in IoT applications, including MQTT, Hypertext Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), and Advanced Message Queuing Protocol (AMQP) [2-5]. Comparing such protocols is not straightforward due to application requirements and constraints. For instance, traffic reduction, protocol efficiency improvement, communication delay reduction, and better quality of service are some obstacles to IoT implementations [5]. These considerations are important when choosing the correct protocol for specific application [5]. However, the use of additional protocol layers induces additional costs for the

overall system which may be critical in certain IoT scenarios. This paper presents findings of a comparative evaluation of MQTT (layers 5-7) with "naked" TCP (layer 4) using standardized payload data. The comparison is presented in terms of header length and total time to transfer or receive the message by the client. MQTT applications are often built using TCP as the transport layer. Thus, this work attempts to quantify the "penalty" or "cost" imposed by MQTT over TCP in a relative sense. This understanding can be very important in the implementation of efficient transport and application protocols in resource constrained scenarios, such as IoT.

The remainder of the paper is organized as follows: Section II provides the overall background including the literature review, and our approach to the experiment. Section III presents the experimental setup, while Section IV presents the data gathered and summarized results of experiments. Section V presents conclusions drawn from the experiments. Section VI discusses possible future work.

II. BACKGROUND

A. Socket

A socket is an endpoint of a two-way communication in an IP network. It is an abstract data structure provided by the operating system to establish communication to transfer messages between endpoints [7]. Each endpoint is identified by a port number and IP address. The TCP transport layer recognizes the application that data is intended to be transferred to by the port number bound to the socket. Thus, the performance of data transfer via a "naked" socket establishes a baseline for comparison of application performance via additional, higher-layer protocols.

B. MQTT

MQTT is a many-to-many messaging protocol capable of transferring messages between multiple producers and consumers [8]. MQTT is described as an efficient, bi-directional, scalable, reliable, and security enabled IoT messaging protocol which can be "scaled down" effectively and implemented on microcontrollers. MQTT may use TCP to establish communication between an end-device and the intervening server, or "broker" [4].

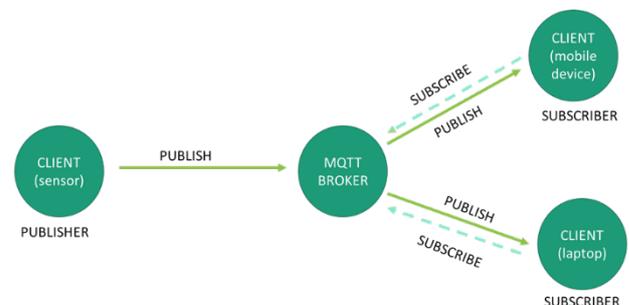


Figure 1. MQTT Publish/Subscribe Model

MQTT uses a Publish/Subscribe construct to achieve one-to-many message distribution [4]. In this structure, end-devices (clients) talk to a central authority (broker). Figure 1

presents the Publish/Subscribe architecture used by MQTT. Essential components of this architecture include:

Publisher (sender): The publisher is a client which sends data (messages) to the broker. Many publishers can connect to the same broker. The publisher may provide data as a part of a ‘topic’, to which other clients (subscribers) may subscribe in order to retrieve related messages.

Broker: The broker is the interface for the publisher and subscriber to exchange message data. The broker forwards messages to clients (subscribers) based on the topic for which the message was published, and the client is subscribed [3].

Subscriber (receiver): A client must subscribe to certain ‘topic’ to receive messages for that topic from the broker. If the topic specified in the SUBSCRIBE message from the subscriber matches the topic in the PUBLISH message from the publisher, then the broker forwards the message to the subscriber [5]. Subscribers can subscribe to several topics, and each topic’s published messages are forwarded [8].

MQTT systems can deliver messages using three different Quality of Service (QoS) classifications [5]:

QoS 0 (At most once): The message is delivered at most once. There is no guaranteed delivery of packets and no extra methods for quality checks [5]. As a result, messages can be lost and connection reliability is dependent on the transport layer (e.g., TCP) [3].

QoS 1 (At least once): The message is delivered at least once [5]. Each message is sent multiple times, and may overlap, until acknowledged by the recipient [3].

QoS 2 (Exactly once): The message is delivered exactly once, which avoids overlapping of identical messages that may occur in QoS 1 [3].

Several studies have examined the viability of MQTT protocol in IoT and how it compares with other IoT communication protocols. A brief exploration of MQTT and CoAP outlining their architecture and message transmission mechanism is presented in [9]. An extensive comparison of MQTT and CoAP concluded that the MQTT protocol is less bandwidth efficient [10].

The comparison of the different IoT messaging protocols is not straightforward. As a result, various authors have examined the effectiveness of certain protocols under different network circumstances. For example, common messaging protocols for IoT have been compared with message overhead classified as higher for MQTT than CoAP but lesser than AMQP and HTTP. Similar conclusions are presented for bandwidth, latency, power consumption and resource requirements [11].

The performance of MQTT and CoAP have been studied under different network conditions, concluding that MQTT messages suffered lower delays for lower packet loss and higher delays for higher packet loss [12]. Additionally, the overhead for MQTT was higher for different message sizes when compared to CoAP [11]. MQTT has been compared with CoAP and Open Platform Communications Unified Architecture (OPC UA) over different cellular networks, reaching similar conclusions [3]. Additionally, the transmission time of CoAP increases every 1024 bytes leading to higher transmission time than MQTT for high payload sizes [3]. A higher payload for MQTT was expected as CoAP used UDP for its transport layer, whereas MQTT used TCP. However, for larger messages, the overhead can become higher for CoAP due to inefficiencies of User Datagram Protocol (UDP) vs. TCP for connection management [12]. Sockets have been used to send text-based data (e.g., JavaScript Object Notation) in an attempt to establish communication between Android mobile applications and IoT embedded systems [13], and some work has compared connection-oriented and connectionless transports [6].

Previous research may be helpful in understanding protocol differences. However, these outcomes didn’t provide an explicit understanding of the overhead incurred by different network configurations for a range of payload sizes. Although seemingly pedestrian, this level of understanding is critical in the efficient implementation of an IoT application, which may be severely resource-constrained.

Thus, the present paper provides a comparative evaluation of MQTT and socket communications. To accomplish this comparison, two distinct experimental environments are used: LAN (controlled) and Internet (uncontrolled). Overhead for a range of message payloads is compared statistically in each environment, using TCP sockets to provide a baseline for comparison with MQTT. Further, a unique application payload is used to compartmentalize data communications, provide intrinsic security, and regularize payload structure.

C. Intelligent Cipher Transfer Object (ICTO)

ICTO is a security technology that includes mechanisms for participant authentication and authorization for access of data, which is protected by cloaking patterns. A portable dynamic rule set, which includes executable code for managing access to the protected set of participants and the protected data, is included within the ICTO. For a given user, the ICTO may provide access to some participants while preventing access to other participants based on this set of access constraints. The ICTO concept extends the conventional Authentication, Authorization, and Accounting (AAA) and Role-Based Access Control (RBAC) concepts by cloaking data at the point of generation with specific user-defined rule sets for access. The owner of data is in command to configure how or when protected data can be accessed by another party [14].

The ICTO technology is independent of what security measures are followed at the channel. Once user data has been encapsulated by a secure ICTO object, the data remains secure even in the absence of any security measures on the communication channel. As a result, the use of ICTO in IoT communications may be of particular interest. In such cases, a clear understanding of the efficiency of messaging protocols is critical. In an IoT environment, for instance, assuming a secure ICTO object has been generated, we then need an efficient data exchange protocol to reliably transmit the object from one device to another. Thus, the results of the present research may be useful in establishing an efficient framework for the exchange of ICTO objects.

III. EXPERIMENTAL SETUP

A. TCP socket

A wireless TCP socket connection was established between two PCs similar in hardware and software where one was configured as the client that sends user data of given size and the other was configured as server that receives data sent from the client. Sending and receiving machines used Intel i5 2.4 GHz processors with Debian Linux kernels with 8 Gigabytes of memory. Python scripts were used to set up the client, server, and the packet logger programs. To trace/analyze packets being exchanged between client and server, the Python library ‘‘pcapy’’ was used. Pcap is based on tcpdump [13]. The data recorded with the Python program was validated with the well-known Wireshark application-layer packet analyzer.

The machine running the client program was configured to send user data to the server and the machine running the server was configured to receive data sent by the client. Python scripts to record every packet that is seen on the network interface were run on both client and server machines during the experiments.

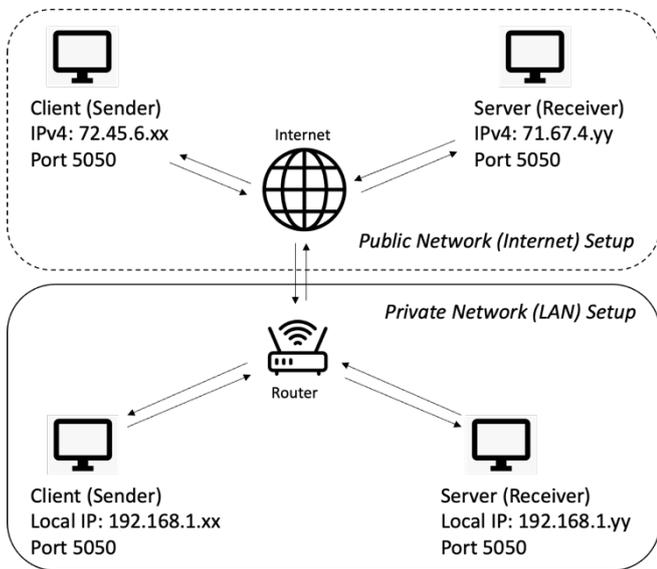


Figure 2. Experimental Setup for TCP socket communication

Experiments were performed for two different configurations of user data: file and byte sequence. In the “file” experiments, user data to be sent by the client is in a pre-existing file. In the “byte” experiments, the user data to be sent by the client is sequence of dynamically generated bytes. This procedure was repeated for two different network environments: Internet (uncontrolled) and Local Area Network (LAN) – a controlled network environment. Figure 2 illustrates the configuration in both environments.

B. MQTT

For setting up MQTT experiments, three PCs with similar hardware and software configurations were used. Since MQTT utilizes a broker to transmit information between multiple clients, the first PC was used as a sending client, the second as the receiving client, and the third as the MQTT broker, all on the wireless interface.

Python scripts were used to run the sending and receiving clients as well as the packet logger. The packet logger program executed on both the sending and receiving machines during each experiment. Other procedures were similar to the setup discussed in Section III.A regarding user data configurations (file and byte) and network environments (Internet and LAN). Figure 3 illustrates the configuration in both environments.

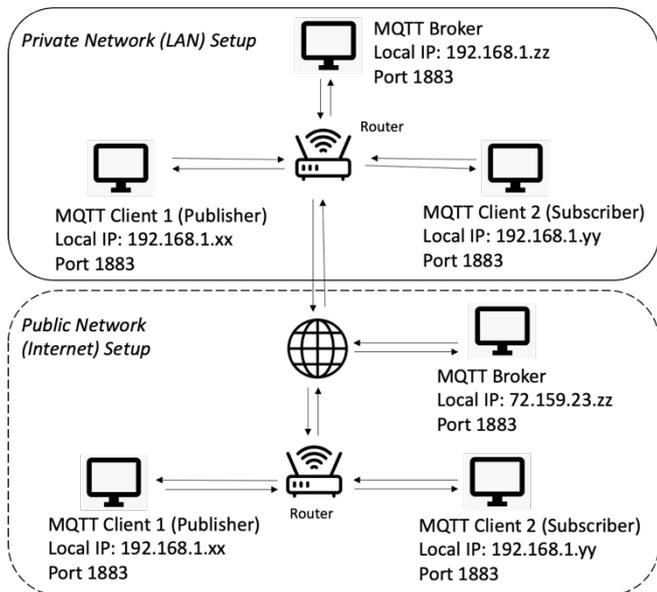


Figure 3. Experimental Setup for MQTT communication

With the experimental configurations detailed in Sections III.A and III.B, at least 150 iterations were completed for each configuration to obtain a statistically valid collection of network performance metrics. Metrics of interest included: total transmit time, total header size, header to payload ratio, and similar valuable measurements. For each experiment, total header size is calculated by summing the header lengths of TCP segment header of each IP packet exchanged between sender and receiver. User payload sizes of 1 byte, 10 bytes, 100 Kbytes, 500 Kbytes, and 1000 Kbytes were used. Figure 3 illustrates experimental setups on different network environments.

IV. EXPERIMENTAL RESULTS

The results of the experiments are divided into two sections: private network (LAN) and public network (Internet). Subsequent plots consist of dependent variables describing network performance metrics (e.g., total transfer time and cumulative header size) versus the independent variable (payload size). Each point in the plot represents the mean of at least 150 iterations of each experiment. The shaded regions surrounding the mean represent the boundaries of a 95% confidence interval with unknown population standard deviation (e.g., via the t-distribution).

A. Private Network (LAN)

Plots generated from experimental data in a controlled environment (LAN) are presented comparing the relative performance of TCP with MQTT for payloads of byte sequences as well as files/objects.

Figure 4 presents the cumulative header size required for varying sizes of user data (transferred as a file or as a sequence of bytes) for both sending and receiving agents of TCP and MQTT in a private network (LAN) setting.

As expected, MQTT requires greater header size for transmitting a user payload of given size as compared to TCP. Surprisingly, with increasing payload size, header overhead for MQTT increases dramatically. In contrast, for the TCP baseline, header overhead remains constant and at least one order of magnitude smaller than MQTT.

This result indicates that TCP is substantially more efficient than MQTT for resource-constrained environments. The convenience and apparent simplicity of using MQTT for a publish/subscribe scenario comes with a “penalty” of vastly more non-payload data transfer.

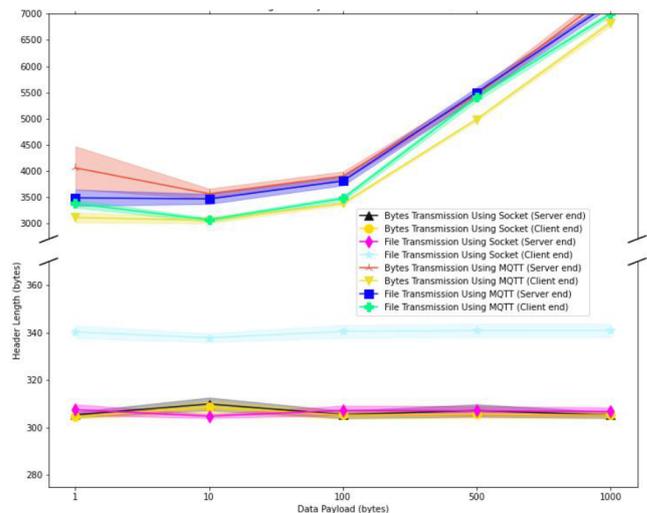


Figure 4. Cumulative Header Size vs Payload Size in LAN

This result indicates that TCP is substantially more efficient than MQTT for resource-constrained environments. The convenience and apparent simplicity of using MQTT for a publish/subscribe scenario comes with a “penalty” of vastly more non-payload data transfer.

Figure 5 presents the total transmission time for varying sizes of user data (transferred as a file or as a sequence of bytes). This data is presented for both sending and receiving agents of TCP and MQTT in a private network (LAN) environment. Although the difference between transfer times for a given payload size is relatively small, it is present and observable.

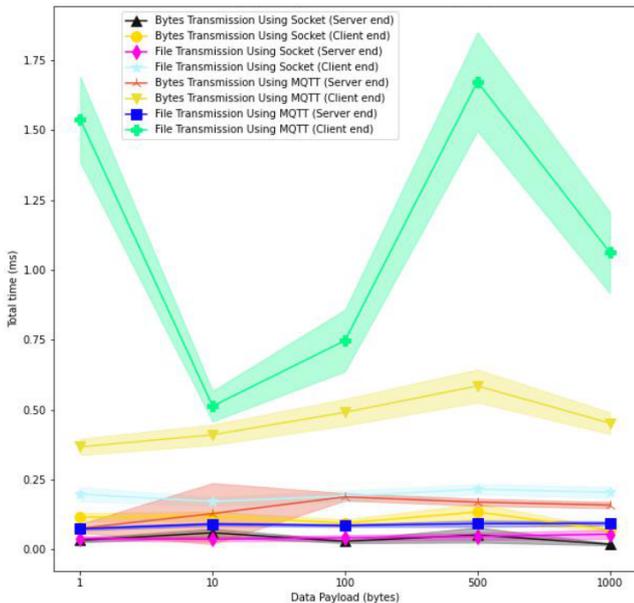


Figure 5. Total Time (ms) vs Payload Size in LAN

From Figure 5, it is clear that the time required for data exchange for most MQTT configurations is substantially higher (by a factor of 2 or more) than those for TCP. A significant difference in the cumulative header size for MQTT and TCP (observed in Figure 4) may be an intuitive reason for the observed time difference.

B. Public Network (Internet)

Plots generated from experimental data in a public network (uncontrolled) are presented comparing the relative performance of TCP with MQTT for payloads of byte sequences as well as files/objects.

Figure 6 presents the cumulative header size required for varying sizes of user data (transferred as a file or as a sequence of bytes). This data is presented for both sending and receiving agents of TCP and MQTT in a public network (Internet) environment.

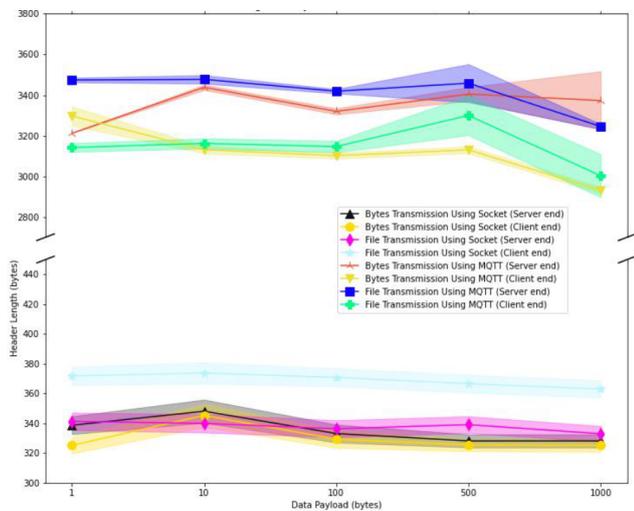


Figure 6. Cumulative Header Size vs Payload Size in Internet

As expected, and as clearly displayed in Figure 6, the header overhead for MQTT is significantly greater than TCP. However, unlike the rising trend of total header size with increasing payload size observed in Figure 4, header

overhead for MQTT seems to be steady with increasing payload size in the Internet environment, but is still an order of magnitude greater than TCP.

Figure 7 presents the total transmission time for varying sizes of user data (transferred as a file or as a sequence of bytes). This data is presented for both sending and receiving agents of TCP and MQTT in a public network (Internet) environment. Wide confidence intervals may be due to the dynamic/unpredictable nature of routing, packet loss, and other factors present in Internet traffic.

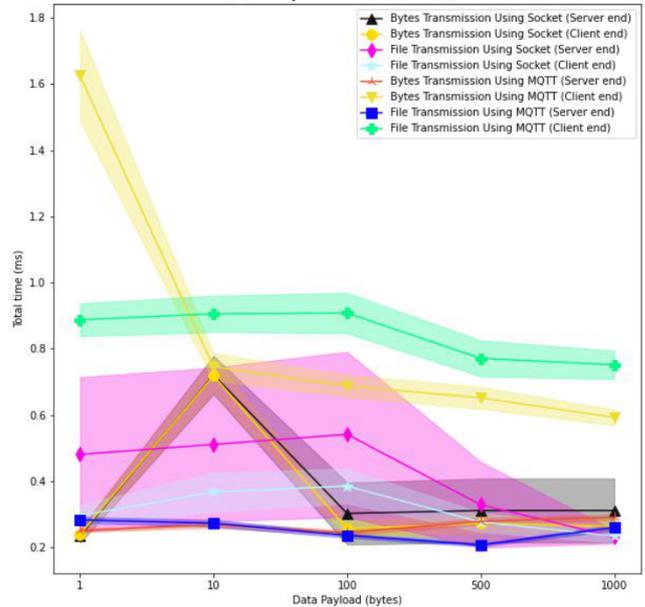


Figure 7. Total Time (ms) vs Payload Size in Internet

Regardless, a clear comparison between MQTT and TCP is evident from Figure 7 in that total transmission time for TCP is typically faster by a factor of 2 or more. This is consistent with the observations and conclusions derived from Figure 5, which performed the same experiment in the LAN (controlled) environment.

V. CONCLUSION

From the results presented in Section IV, we find that for normal data transfer, either as a file or a series of bytes, TCP to performs better in total transmit time and payload to header ratio (goodput). This is unsurprising, because MQTT leverages TCP as the transport layer. However, the overall inefficiency of MQTT is surprising, providing transmission delay of at least a factor of 2 (and typically much greater), and an overhead inefficiency of an order of magnitude, regardless of network environment. Thus, for the purpose of transmitting information, TCP sockets are substantially more efficient. The presence of a broker to moderate communication between publishers and subscribers in MQTT may provide application flexibility, but the resulting operational inefficiencies are concerning.

As in any application, additional aspects of each alternative must be considered. For instance, MQTT can operate in various QoS modes and therefore certain performance parameters like reliability and transfer time may be bounded. Additionally, the message queueing feature of MQTT enables relatively passive IoT devices to transmit/fetch data from the broker regardless of operating concurrency between the publisher and subscriber. However, MQTT typically provides minimal security through basic authentication (e.g., username, password).

As a result, in the context of IoT, a choice between the use of TCP or MQTT for ICTO object transport becomes clearer. Although MQTT's message queueing capability may be useful, the lack of security for data in-flight and at-

rest (e.g., on the broker) may be a critical consideration. As a result, the direct use of TCP could be preferable because of the substantial improvement in efficiency. However, ICTO technology offers security regardless of the security imposed by network protocols.

VI. FUTURE WORK

In the future, we intend to further assess MQTT in terms of network performance by comparing it to other popular protocols like CoAP. Further, we aim to experiment more thoroughly with MQTT by altering the QoS parameters between experiments, introducing multiple subscribers and publishers, and stress testing the message queuing feature. From the findings, we intend to further explore favorable communication protocols to transport ICTO objects.

REFERENCES

- [1] P. Gralla, "How the Internet Works," Que Publishing, 1998.
- [2] B. A. Forouzan, "TCP/IP Protocol Suite," Guide books. <https://dl.acm.org/doi/abs/10.5555/572565> (accessed Mar. 12, 2023).
- [3] L. Durkop, B. Czybik, and J. Jasperneite, "Performance evaluation of M2M protocols over cellular networks in a lab environment," in 2015 18th International Conference on Intelligence in Next Generation Networks, 2015, pp. 70–75, doi: 10.1109/ICIN.2015.7073809.
- [4] "MQTT - The Standard for IoT Messaging." <https://mqtt.org/> (accessed Mar. 16, 2023).
- [5] Y. Sueda, M. Sato, and K. Hasuike, "Evaluation of Message Protocols for IoT," in 2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD), May 2019, pp. 172–175. doi: 10.1109/BCD.2019.8884975.
- [6] M. Xue and C. Zhu, "The Socket Programming and Software Design for Communication Based on Client/Server," in 2009 Pacific-Asia Conference on Circuits, Communications and Systems, May 2009, pp. 775–777. doi: 10.1109/PACCS.2009.89.
- [7] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," in 2017 International Conference on Engineering & MIS (ICEMIS), May 2017, pp. 1–6. doi: 10.1109/ICEMIS.2017.8273112.
- [8] "MQTT Version 5.0." OASIS. [Online]. Available: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html> (accessed Mar. 10, 2023).
- [9] D. B. Ansari, A. Rehman, and R. Ali. "Internet of things (iot) protocols: a brief exploration of MQTT and COAP." International Journal of Computer Applications 179.27 (2018): 9-14.
- [10] V. Seoane, C. Garcia-Rubio, F. Almenares, and C. Campo, "Performance evaluation of CoAP and MQTT with security support for IoT environments," Computer Networks, vol. 197, p. 108338, 2021.
- [11] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in 2017 IEEE International Systems Engineering Symposium (ISSE), Oct. 2017, pp. 1–7. doi: 10.1109/SysEng.2017.8088251.
- [12] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Apr. 2014, pp. 1–6. doi: 10.1109/ISSNIP.2014.6827678.
- [13] N. Nikolov and O. Nakov, "Research of Communication Between IoT Cloud Structure, Android Application and IoT Device Using TCP Sockets," in 2019 X National Conference with International Participation (ELECTRONICA), May 2019, pp. 1–4. doi: 10.1109/ELECTRONICA.2019.8825568.
- [14] G. S. Smith, M. L. Smith-Weed, D. M. Fischer, and E.M. Ridenour, "System and methods for using cipher objects to protect data," (US US20220004649A1) USPTO, 2017.
- [15] "tcpdump(1) man page | TCPDUMP & LIBPCAP." <https://www.tcpdump.org/manpages/tcpdump.1.html> (accessed Mar. 10, 2023).