SCHC over DtS-IoT: Performance of SCHC Confirmation Modes in Satellite IoT

Rodrigo Munoz-Lara
Dept. of Electrical Engineering
University of Chile
Santiago, Chile
https://orcid.org/0009-0002-6888-1206

Sandra Céspedes
Dept. of Computer Science
& Software Eng.
Concordia University
Montreal, QC, Canada
https://orcid.org/0000-0003-1686-2644

Marcos Diaz
Dept. of Electrical Engineering
University of Chile
Santiago, Chile
https://orcid.org/0000-0002-7701-5839

2024 IEEE 10th World Forum on Internet of Things (WF-IoT) | 979-8-3503-7301-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/WF-IoT62078.2024.10811196

Abstract—**Static Context Header Compression and Fragmentation (SCHC) is a standard defined as an adaptation layer for supporting IPv6, UDP, and CoAP protocols in low power wide area network (LPWAN) technologies. SCHC has fragmentation modes that use confirmation mechanisms to deliver reliable communications. Direct-to-satellite Internet of Things (DtS-IoT) directly communicates between end devices and the satellite in an IoT environment. If SCHC is implemented in a DtS-IoT environment, applications using IPv6 could have global coverage through the Internet of Things (IoT). Unfortunately, it is not known whether SCHC can operate with the current acknowledgement mechanisms in a DtS-IoT environment, and if so, it is not known which one performs better. This work evaluates SCHC acknowledgement modes for ACK-on-Error fragmentation mode, including the new SCHC Compound ACK mode. It determines which mode has the lowest transfer delay for different LoRa data rates and frame error rates in a DtS-IoT environment with LEO satellites.**

Index Terms—**DtS-IoT, fragmentation, IoT, IPv6, satellite, SCHC**

I. Introduction

Direct-to-satellite IoT (DtS-IoT) refers to the direct communication between end devices and satellites within an IoT environment, bypassing the need for a terrestrial LPWAN gateway (such as a radio gateway for LoRaWAN or an eNodeB for NB-IoT). In DtS-IoT, the LPWAN gateway is integrated into the satellite. LPWAN technologies feature a Maximum Transmission Unit (MTU) of only a few tens of bytes and lack link layer fragmentation. Consequently, application protocols like MQTT or CoAP, which rely on UDP and IP for transport and network layers, are not feasible in this context.

The *Internet Engineering Task Force (IETF)* through the *lpwan* working group, developed a standard called Static Context Header Compression and Fragmentation or SCHC [1] to compress and fragment IPv6, UDP and CoAP protocol headers for Low Power Wide Area Networks (LPWANs) technologies. This standard allows communicating over IPv6 and related protocols using compressed and fragmented SCHC packets in devices with bandwidth constraints.

DtS-IoT uses LPWAN technologies to communicate with end devices. Therefore, it would make sense to implement SCHC over LPWAN in a DtS-IoT environment to provide end devices with global coverage and support for superior protocols. Regrettably, it is unclear if SCHC can function with the current acknowledgement mechanisms in a DtS-IoT environment, and if it can, it is unknown which mechanism performs more effectively. This study aims to validate the performance of SCHC in a DtS-IoT environment by using SCHC in the uplink. It identifies the SCHC acknowledgement mechanism that results in the lowest transfer delay, considering packet errors in an optimal DtS-IoT scenario.

The work is organized as follows. Section II-A briefly overviews the SCHC standard operation. Section II-B shows the primary research on using IPv6 in a terrestrial and satellite IoT environment. Section III describes the main issues of using SCHC in a DtS-IoT environment. Section IV shows the possible confirmation scenarios and methods for determining the transfer delay. Section V presents and discusses the results for transfer delay as a function of the LoRaWAN data rates and frame error rate (FER). Finally, section VI shows the main points obtained from the work results.

II. Previous Works

A. SCHC Background

SCHC is considered an adaptation layer between a network layer protocol such as IPv6 and a data link layer such as LPWAN-type technologies. This adaptation layer comprises two sublayers called compression and fragmentation. Figure 1 shows the transmission process of an IPv6 packet as it passes through the SCHC sublayers. First, the IPv6 packet is compressed using fixed rules. The compression sublayer delivers an SCHC packet (compressed IPv6 packet). If the size of the SCHC packet is less than or equal to the MTU of the LPWAN technology, it is transmitted without fragmentation; otherwise, the SCHC packet is delivered to the fragmentation sublayer.

In the fragmentation sublayer, the SCHC packet is divided into several fixed bytes called ***tiles***. A fixed number of tiles form a SCHC window. When all the tiles of an SCHC window

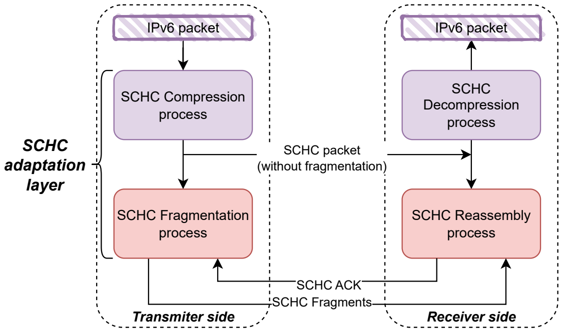

Fig. 1: The flow that a message follows as it enters the SCHC at both the transmitter and receiver of the message.

have been sent, the receiver sends an acknowledgement message to the transmitter indicating which tiles arrived correctly and which did not. Tiles are sent from the transmitter to the receiver in messages called ***SCHC regular fragments*** (hereafter, regular fragment). Figure 2 shows the division of a SCHC packet into SCHC tiles and SCHC windows.

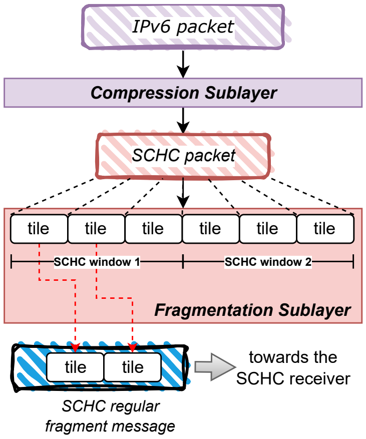

Fig. 2: Representation of an SCHC packet divided into tiles and windows within the fragmentation sublayer (Transmitter side).

The fragmentation sublayer can operate in one of three modes for tile transmission: No-ACK, ACK-Always, and ACK-on-error. Also, the ACK-on-error mode can implement one of three tile *acknowledgement mechanisms*. Figure 3 shows graphically the three confirmation mechanisms.

- **Acknowledgement at the end of the SCHC window:** The receiver sends an SCHC ACK message to the transmitter exactly at the end of the reception of an SCHC window. The ***SCHC ACK*** message indicates which tiles were successfully received and which were not.
- **Acknowledgement at the end of the SCHC session:** The receiver sends the transmitter an SCHC ACK message only at the end of the SCHC session. For each window, the receiver must send an SCHC ACK message. Because the acknowledgement of each window will be sent at the end of the session, the transmitter can send the tiles of window i+1 without waiting for the SCHC ACK message of the tiles of the i-th window.
- **Acknowledgement with SCHC Compound ACK:** The receiver can acknowledge both at the end of a window or session. The receiver uses an SCHC Compound ACK message defined in RFC 9441 [2] to acknowledge. The Compound ACK mechanism can group the acknowledgements of each SCHC window into a single SCHC compound ACK message. This avoids sending an acknowledgement for each SCHC window.

B. State of the Art

IPv6 support in a DtS-IoT environment is a developing area of research. This section organizes the works in two aspects: IPv6 in terrestrial IoT networks and IPv6 in satellite IoT networks. In the case of satellite networks, since there are no studies of IPv6 over LPWAN, the studies of CoAP and MQTT are shown because they are protocols oriented to IoT that must operate over IPv6.

1) IPv6 in terrestrial IoT: In [3], Moons et al. compare the fragmentation mechanism for SCHC and 6LoWPAN. The comparison is from the overhead and the number of fragments for transmitting a fixed-size packet for both standards. The analysis shows that SCHC is the only protocol supporting end-to-end IPv6 connectivity for a SigFox, LoRa, and DASH-7 compliant device.

In [4], Ayoub et al. compare two IETF-standard solutions to compress IPv6 over constrained nodes: SCHC and 6LoWPAN. The authors implement the SCHC mechanism in the NS3 network simulator. They show that the SCHC protocol is better regarding header compression by providing a smaller header size than 6LoWPAN.

In [5], Aguilar et al. present an overview of SCHC fragmentation modes and evaluate them through simulations. The results show that the No-ACK mode has the lowest total channel occupancy, highest goodput, and lowest total delay, but it lacks a reliability mechanism. ACK-Always and ACK-on-Error modes offer the same total delay and similar total channel occupancy as the No-ACK mode. ACK-on-Error offers higher goodput than the three modes.

Aguilar et al. present in [6] a model to evaluate the performance of the SCHC ACK-on-Error mode, applying it to the standard version of SCHC (RFC 8724). The model considers the number of SCHC ACK messages on the downlink to evaluate four metrics: ACK message overhead, ACK bit overhead, ACK bit overhead with L2 headers, and the percentage of bits used per fragment. The theoretical results show that channel occupancy directly relates to the SCHC specification parameters: tile and window sizes.

Munoz-Lara et al. present in [7] a model to determine the channel occupancy efficiency based on the transmission times of SCHC messages in the LoRaWAN uplink channel. The

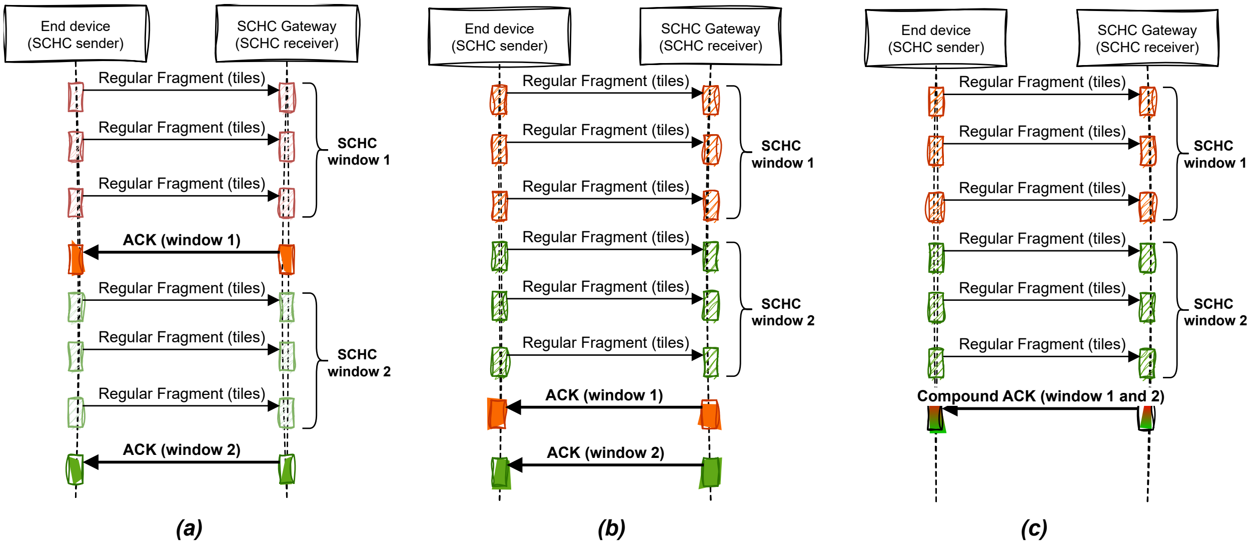

Fig. 3: Acknowledgement Mechanisms for SCHC standard. (a) ACK at the end of each window. (b) ACK for each window at the end of the SCHC session. (c) Compound ACK at the end of the SCHC session.

model is compared with experimental data obtained from the transmission of fragmented packets using an SCHC implementation over LoRaWAN. This modelling provides a relationship between channel occupancy efficiency, LoRa's spreading factor, and the error probability of an SCHC message.

2) IPv6 in satellite IoT: Giotti et al. present in [8] an evaluation of the performance of the CoAP protocol on a satellite link based on DVB-S2 and DVB-RCS. The results show that when the payload of CoAP is equal to 1024 bytes, fragmentation starts, and therefore, the end-to-end delay increases drastically with payloads higher than this value. On the other hand, increasing the payload leads to higher protocol efficiency until the payload size reaches the fragmentation threshold.

Collina et al. present in [9] a quantitative analysis of the performance of CoAP and MQTT protocols on high-delay links. The results show that, with the default protocol parameters, MQTT offers better performance in terms of throughput in any of the scenarios considered. Regarding latency, CoAP slightly outperforms MQTT in case of low offered traffic, low loss probability, and high delay.

III. Issues of SCHC over Satellite Links

For the operation of an LEO satellite, two-time windows are identified:

- **Visibility window** (*visibility time*): is the time during which a device on the ground can communicate with a satellite.
- **Pass-to-pass window** (*revisit time*): is the time between the end of a visibility window (pass i) and the beginning of the next visibility window (pass $i+1$) for the same device on the ground.

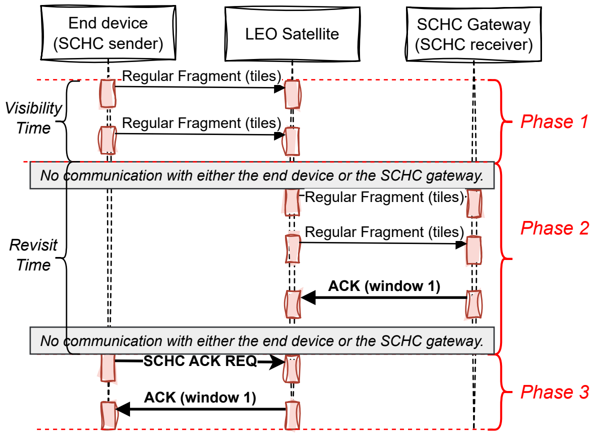

Fig. 4: Message flow example for a SCHC window transmission in DtS-IoT environment

When the SCHC window is **smaller** than the visibility window, the visibility window may be large enough to allow sending one or more SCHC windows in a satellite pass. In this case, the problem of using SCHC technology is reduced to determine which acknowledgement mechanism minimizes the transfer delay of an SCHC packet. If an acknowledgement mechanism is used at the end of the session, the end device can start transmitting the tiles of the next SCHC window without waiting for the acknowledgement of the receipt of the tiles of the current SCHC window. In Figure 3.a, the SCHC sender must wait for an acknowledgement before sending the tiles to the next SCHC window. Figures 3.b and 3.c shows that the SCHC sender can continue transmitting the tiles of the next SCHC window.

IV. Methodology

The work presented in [6] determines the number of SCHC fragments to be retransmitted each time an SCHC window is sent. The authors use a probabilistic model based on a geometric distribution. Munoz-Lara et a. [7] take that work and define the message flows by adding the time on air to each SCHC message. This research takes the input from both works and creates a model for the transfer delay of an SCHC packet using an LEO satellite's revisit and visibility times. With the model, the research determines the most efficient acknowledgement mechanism when using SCHC in a DtS-IoT environment. Therefore, the first step is to define the transfer delay.

a) Transfer Delay: Time elapsed between the sending of the first message associated with an SCHC session and the reception of the last SCHC ACK of the session. It is calculated from the time on air (ToA) of each message involved in a SCHC session and the visibility and revisit times of a LEO constellation. For LoRa and LR-FHSS modulations, the calculation of the ToA is defined in [10] and [11], respectively.

A. SCHC message flow in DtS-IoT

The SCHC message flow is defined in RFC 8724 [1] and RFC 9441 [2]. Figure 4 shows the transmission of only an SCHC window adapted to a DtS-IoT environment. This transmission can be divided into 3 phases.

- **Phase 1:** Sending from the end device to the satellite of the tiles carried by SCHC regular fragments messages. The messages are stored in the satellite and delivered to the SCHC gateway in phase 2.
- **Phase 2:** Reception at the SCHC gateway of the SCHC regular fragments. In addition, the SCHC gateway responds with an SCHC ACK acknowledgement message. This message is stored in the satellite and sent to the end device in phase 3.
- **Phase 3:** Reception at the end device of the SCHC ACK.

B. Transfer Delay for each acknowledgement mechanism

For the acknowledgement mechanism *at the end of each SCHC window*, the tiles will be transmitted by the end device to the satellite in a visibility window (phase 1), retransmitted by the satellite to the SCHC gateway in a revisit window (phase 2), and finally confirmed to the end device in a second visibility window (phase 3). This process is the same for both the transmission and retransmission of tiles.

The process is slightly different for the acknowledgement mechanisms *at the end of each SCHC session* and *SCHC compound ACK*. In both processes, the tiles will be transmitted by the end device to the satellite in a visibility window (phase 1) and retransmitted by the satellite to the SCHC gateway in a revisit window (phase 2). In this case, an acknowledgement message does not exist when finishing the SCHC window.

Therefore, to calculate the transfer delay, it is only necessary to count the number of SCHC transmission windows and retransmission windows. Table I shows the transfer delay for each acknowledgement mechanism.

TABLE I: Summary of transfer delay models.

Acknowledgement mode	Transfer delay
After each window	$\left[\sum_{j=1}^{n_W+N_{sum}} (V_j + R_j)\right] + T_{ar} + RX_1$
At the end of the session	$\left[\sum_{j=1}^{nV+N_{sum}} (V_j + R_j)\right] + T_{ar} + RX_1$
SCHC Compound ACK	$\left[\sum_{j=1}^{nV+N_i} (V_j + R_j)\right] + T_{ar} + RX_1$

where:

- n_W: Number of SCHC windows needed to deliver a SCHC packet message successfully. The number of SCHC windows is calculated in (1).

$$n_W = \left\lceil \frac{n_T}{window_{size}} \right\rceil , \tag{1}$$

where n_T is the number of tiles into which an SCHC packet is divided. It is determined by (2). $window_{size}$ is the number of tiles allowed in a SCHC window. For LoRaWAN, $window_{size}$ is equal to 63 according to RFC 9011 [12].

$$n_T = \frac{P}{tile_{size} \cdot 8} , \tag{2}$$

where P is the size of the *SCHC packet* in bits. $tile_{size}$ is the size of the tile in bytes defined in each *profile*. For LoRaWAN, $tile_{size}$ is equal to 10 bytes according to RFC 9011 [12].

- N_{sum}: is the total number of retransmission windows within an SCHC session and is given by the sum of retransmissions occurring after each SCHC window. It is calculated by (3).

$$N_{sum} = \sum_{j=0}^{3} N_i , \tag{3}$$

where N_i is the number of retransmission windows for the i-th SCHC window.

- nV is the number of visibility windows in which the Full regular fragments are transmitted.
- V_j is the duration of the visibility window j
- R_j is the duration of the revisit window j.
- T_{ar}: Time to transmit the bits of a *SCHC ACK REQ* message.
- RX_1: Time the LoRa radio module uses to detect a preamble in the downlink.

V. Results and Analysis

This section shows the numerical evaluation of the model described in section IV and formulated in Table I for each

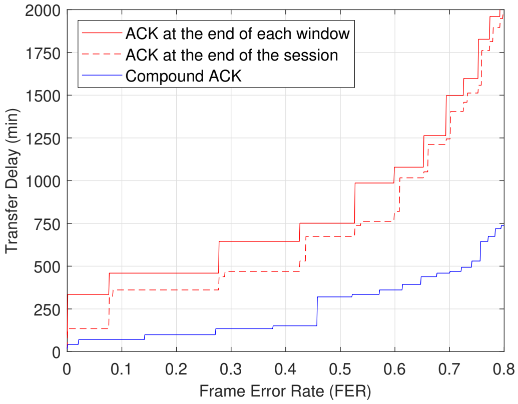

Fig. 5: Transfer Delay vs FER for Data Rate 0 (SF12)

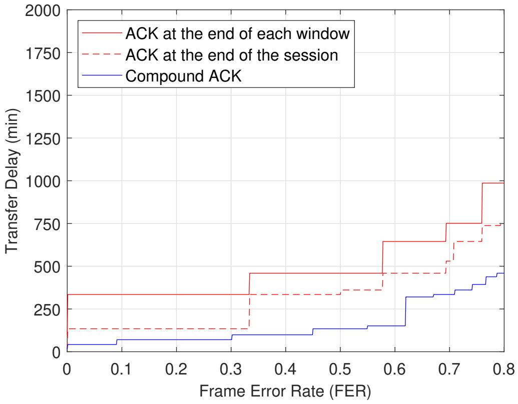

Fig. 6: Transfer Delay vs FER for Data Rate 5 (SF7)

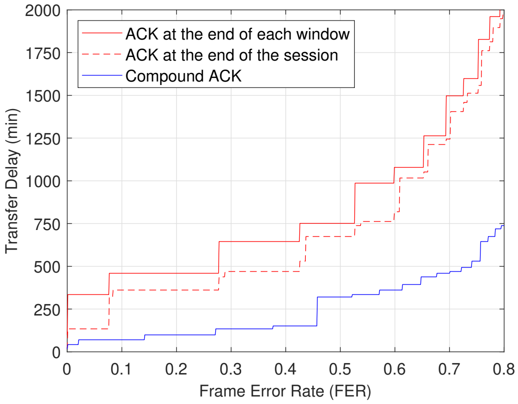

Fig. 7: Transfer Delay vs FER for Data Rate 7 (LR-FHSS: CR1/3)

confirmation mechanism. The numerical evaluation uses Table II parameters and visibility and revisit times of the Swarm constellation obtained from the forecast of passes at position 33.4867° S, 070.5720° W [13].

TABLE II: Parameters to model evaluation

Param	Value
LPWAN Technology	LoRaWAN
Frequency band	AU915-928
SCHC Window size ($window_{size}$)	63 tiles
SCHC Tile size ($tile_{size}$)	10 bytes
SCHC Packet size (P)	20160 bits
Duration of the visibility window j (V_j)	Constellation forecast [13]
Duration of the revisit window j (R_j)	Constellation forecast [13]

Figures 5, 6, and 7 show the transfer delay as a function of frame error rate (FER) for the LoRaWAN Data Rate 0 (DR0), Data Rate 5 (DR5), and Data Rate (DR7) respectively for the three acknowledgement mechanisms. The graph shows that for all three data rates, the SCHC Compound ACK mechanism always has the lowest transfer delay independently of the FER. From the same analysis, the mechanism with confirmation at the end of each SCHC window always has the highest transfer delay. All graphs show that the transfer delay increases when the FER increases for all three confirmation mechanisms. On the other hand, increasing the FER increases the number of retransmission windows. The retransmission windows are integers. Therefore, the transfer delay increases stepwise as the FER increases. Each step is a discrete increase in the number of retransmission windows.

Figures 8, 9, and 10 show the transfer delay as a function of frame error rate (FER) for Acknowledgement at the end SCHC Window, Acknowledgement at the end SCHC Window, and Acknowledgement with SCHC Compound respectively for all LoRaWAN Data Rates. The graph shows that data rates 5 and 4, independent of FER, always have the lowest transmission delay for all acknowledgement mechanisms. Data rates 0, 1, 2 and 7, independent of FER, always have the highest transmission delay.

The results show that, from a transfer delay point of view, LR-FHSS modulation is equal to or worse than LoRa modulation (for 125 KHz LoRa channels). Furthermore, the results show that to decrease the transfer delay of an SCHC packet, the best option is to use the SCHC Compound ACK at the end of the SCHC session.

VI. Conclusions

Satellite networks using LPWAN can benefit from the SCHC standard for IP applications, but no studies have evaluated SCHC in satellite IoT networks. Without these evaluations, SCHC's performance in satellite scenarios remains unknown, hindering the use of IP-enabled IoT applications. This work evaluates the confirmation mechanisms for the fragmentation modes of the SCHC standard in a satellite scenario. The results show that the mechanism with the lowest transmission time and latency is SCHC Compound ACK while using SCHC ACK at the end of each window maximizes transmission times and latency. Future research could also benefit from conducting empirical tests of confirmation mechanisms

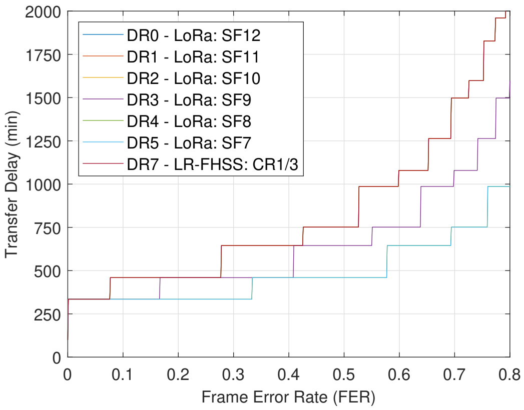

Fig. 8: Transfer Delay vs FER for Acknowledgement at the end of the SCHC window.

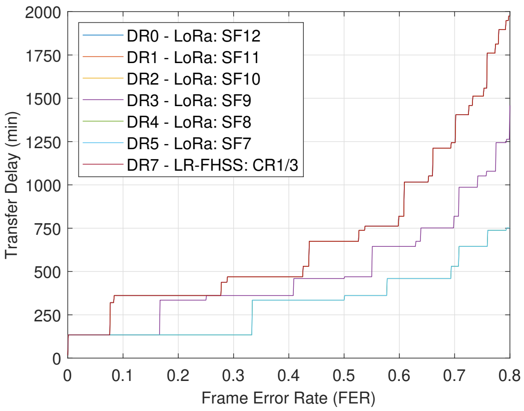

Fig. 9: Transfer Delay vs FER for Acknowledgement at the end of the SCHC Session.

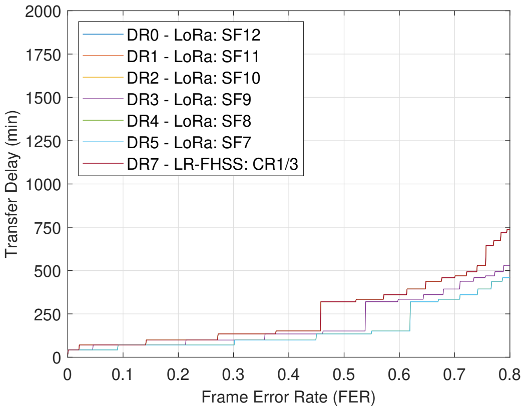

Fig. 10: Transfer Delay vs FER for Acknowledgement with SCHC Compound

and comparing them with the results obtained in this work, for example, using an SCHC stack in a satellite environment.

VII. Acknowledgments

Rodrigo Muñoz would like to thank the *Agencia Nacional de Investigacion y Desarrollo (ANID)* for funding his doctoral studies through a scholarship ANID BECAS/DOCTORADO NACIONAL 21221482. He also thanks the Department of Electrical Engineering of the University of Chile for the scholarship to participate in international congresses. The authors of this article also would like to acknowledge ANID Project Basal FB0008, NSERC Canada Discovery Grant RGPIN-2024-05730, and ANID FONDEF IDEA ID23I10360.

References

[1] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and J.-C. Zúñiga, *SCHC: Generic Framework for Static Context Header Compression and Fragmentation*, RFC 8724, Apr. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8724

[2] J.-C. Zúñiga, C. Gomez, S. Aguilar, L. Toutain, S. Cespedes, and D. S. W. L. Torre, *Static Context Header Compression (SCHC) Compound Acknowledgement (ACK)*, RFC 9441, Jul. 2023. [Online]. Available: https://www.rfc-editor.org/info/rfc9441

[3] B. Moons, A. Karaagac, J. Haxhibeqiri, E. D. Poorter, and J. Hoebeke, "Using SCHC for an optimized protocol stack in multimodal LPWAN solutions," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, April 2019, pp. 430–435.

[4] W. Ayoub, F. Nouvel, S. Hmede, A. E. Samhat, M. Mroue, and J.-C. Prévotet, "Implementation of SCHC in NS-3 and Comparison with 6LoWPAN," in *2019 26th International Conference on Telecommunications (ICT)*, April 2019, pp. 432–436.

[5] S. Aguilar, A. Marquet, L. Toutain, C. Gomez, R. Vidal, N. Montavont, and G. Z. Papadopoulos, "LoRaWAN SCHC Fragmentation Demystified," in *Ad-Hoc, Mobile, and Wireless Networks: 18th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2019, Luxembourg, Luxembourg, October 1–3, 2019, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2019, p. 213–227. [Online]. Available: https://doi.org/10.1007/978-3-030-31831-4_15

[6] S. Aguilar, P. Maillé, L. Toutain, C. Gomez, R. Vidal, N. Montavont, and G. Z. Papadopoulos, "Performance Analysis and Optimal Tuning of IETF LPWAN SCHC ACK-on-Error Mode," *IEEE Sensors Journal*, vol. 20, no. 23, pp. 14 534–14 547, Dec 2020.

[7] R. Munoz-Lara, J. Saez Hidalgo, F. Canales, D. Dujovne, and S. Céspedes, "SCHC over LoRaWAN Efficiency: Evaluation and Experimental Performance of Packet Fragmentation," *Sensors*, vol. 22, no. 4, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/1531

[8] D. Giotti, L. Lamorte, R. Soua, M. R. Palattella, and T. Engel, "Performance Analysis of CoAP under Satellite Link Disruption," in *2018 25th International Conference on Telecommunications (ICT)*, June 2018, pp. 623–628.

[9] M. Collina, M. Bartolucci, A. Vanelli-Coralli, and G. E. Corazza, "Internet of Things application layer protocol analysis over error and delay prone links," in *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, Sep. 2014, pp. 398–404.

[10] Semtech, *AN1200.13 and AN1200.17: LoRa Modem Design*, 2013, [Accessed 01-08-2024]. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-connect/sx1272

[11] (2022) AN1200.64 LR-FHSS System Performance. [Accessed 01-08-2024]. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-edge/lr1120

[12] O. Gimenez and I. Petrov, *Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN*, RFC 9011, Apr. 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9011

[13] (2024) Swarm - Satellite Pass Checker. [Accessed 01-08-2024]. [Online]. Available: https://bumblebee.hive.swarm.space/pass-checker/