



HAL
open science

Real-Time QCSP Communication System Prototyping

Camille Monière, Bertrand Le Gal, Emmanuel Boutillon

► **To cite this version:**

Camille Monière, Bertrand Le Gal, Emmanuel Boutillon. Real-Time QCSP Communication System Prototyping. Colloque GDR SoC², 2022, Strasbourg, France. . hal-03880799

HAL Id: hal-03880799

<https://hal.science/hal-03880799>

Submitted on 1 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

Real-Time QCSP Communication System Prototyping

Camille Monière
Lab-STICC, CNRS, UMR 6285
Université de Bretagne Sud
 Lorient, France
 camille.moniere@univ-ubs.fr

Bertrand Le Gal
IMS, CNRS, UMR 5218
Bordeaux INP
 Talence, France
 bertrand.legal@ims-bordeaux.fr

Emmanuel Boutillon
Lab-STICC, CNRS, UMR 6285
Université de Bretagne Sud
 Lorient, France
 emmanuel.boutillon@univ-ubs.fr

Abstract—In wireless communications, frame detection and synchronization are usually performed using a preamble, consuming bandwidth and resources that are not negligible for small packets. Recently, a new kind of preamble-free frame called Quasi Cyclic Small Packet (QCSP) have been proposed. This paper present a real-time implementation of a QCSP transmission, at both transmitter and receiver sides. A credible QCSP use case is depicted. A real-world experimentation is also detailed and demonstrates that the emission/reception process of a QCSP frame is feasible at low hardware cost, which make the QCSP frame very attractive for Low Power Wide Area Networks.

Index Terms—Real-Time, CCSK, Small Packets, Prototyping, Internet of Things, Low Power Wide Area Network.

I. INTRODUCTION

Transmitting small amount of data in an unsupervised communication network is a real challenge, especially for the Internet of Things (IoT). IoT frames usually consists of a redundant error-tolerant payload, headed by a “preamble”, a standardized waveform used for signal synchronization. However, the shorter the packets get, the less negligible the preamble is regarding the payload (e.g. In Ultra Reliable Low Latency context [1]). Particularly since resources allocated to preamble transmission are wasted from a communication point of view [2]. There is preamble-less strategies in the literature [3]–[5], but demonstrated only with a signal energy greater than the noise one (positive SNR in dB).

A recently introduced preamble-less frame called Quasi Cyclic Small Packet (QCSP) [6], [7], associating Cyclic Code Shift Keying (CCSK) modulation and Non-Binary Error Control Code (NB-ECC), has proven to work for very low SNR (down to -10 dB), relevant in the IoT context.

In [8], the authors demonstrated the low complexity of the QCSP transmitter, ideal for IoT devices constrained by power usage. They also described a method to implement efficiently the QCSP receiver detection task. This article extends this work, presenting a first proof of concept in the form of a real-time complete QCSP system.

II. CONTEXT

This section briefly reviews the context of the paper. First, the QCSP communication system described in [7] and [6]

is summarized. Then, the settings of the experimentation are introduced.

The communication system is depicted in Fig. 1. A message M to be transmitted is first encoded using a Non-Binary Low Density Parity Check (NB-LDPC) encoder into a codeword C of N symbols. The codeword is then CCSK modulated using a pseudo random binary sequence P_0 of size q [9]. The resulting CCSK frame is modulated by Binary Phase-Shift Keying (BPSK) and over-modulated [10]. The QCSP frame F is finally filtered by a half raised cosine filter and transmitted using an RF device. In reception, the same filter is applied on an over-sampled signal. Oversampling implies the use of several detectors in parallel, one per sampling hypothesis. Each one uses a CCSK based detection method [7]. It mainly consists in comparing a score function to a threshold value U_0 . This function is similar to an N -long cumulative sum of the maximum of correlation between P_0 and the last received “frame”, so $N \times q$ samples divided in N consecutive vectors of q samples. Using [8], it can be implemented efficiently on a multicore processor. Data associated with the highest-scoring sampling hypothesis are then synchronized. The first step of synchronization uses the same algorithm as in detection but with a finer frequency resolution. Then it uses the information of the NB-LDPC code to remove time inaccuracies and remove remaining phase-frequency errors by a linear regression on the time-synchronized data correlation maxima [7]. The synchronized frame is CCSK demodulated and the result fed to a NB-LDPC decoder [11] to retrieve the received message.

The center frequency has been set to 433.95 MHz (Industrial Scientific and Medical (ISM) range), used by other IoT standards. A payload of 20 symbols of 6 bits is sent

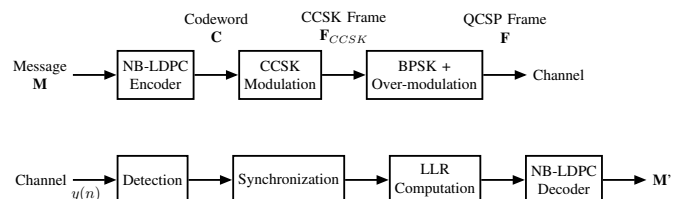


Fig. 1. QCSP communication system model

with an effective (modulation and coding) rate of $\frac{1}{32}$ and an oversampling factor of 8. A sampling rate of 2 Msps has been targeted, to reach a bit rate of around 8 kbps. To fulfill ISM band restrictions, a delay of 15 seconds is inserted between each transmission.

III. COMPLETE REAL-TIME SYSTEM

This section presents the current proof of concept. First, it details the platforms used to implement the transmitter and the receiver. Then, the experimentation and its results are presented.

The real-time transmitter is implemented on a Raspberry Pi 4 (ARM, 4 cores, 1.5 GHz, RAM 4 GB) using C/C++ codes on a 64-bits Linux, without any optimization other than classical compilation flags. The Pi 4 is powered by a 5 kAh external battery. It is equipped with a low-cost GPS module wired on the GPIO and an Universal Software-defined Radio Peripheral (USRP) B205 mini-1 connected by USB-3.0. The USRP only performed the RF tasks, with a transmission power measured to around 4 dBm. The gathered GPS coordinates are embedded in the payload which is sent in the QCSP frame. The receiver platform consists of a plugged-in laptop (Intel Core i5-8265U, 4 cores, 3.5 GHz, RAM 16 GB) running a 64-bits Linux connected to an USRP X310 which contains a Kintex 7 Xilinx FPGA. Its implementation is hybrid:

- the filter is done directly on the FPGA using USRP specification Radio Frequency Network-on-Chip (RFNoC),
- the detection is implemented using OpenMP to process oversampling phases in parallel, with handmade optimized C/C++ codes (using [8]) to implement QCSP specific tasks,
- the remaining tasks are performed by slightly optimized MATLAB/Octave program.

The USRP communicates to the laptop using the dedicated USRP Hardware Driver (UHD) [12] and the detection transmits data to MATLAB/Octave using a Linux FiFo. The threshold was tuned to limit the number of false alarms to one every 30 seconds in average, to avoid wasting CPU time performing synchronization and decoding on noise.

The experimentation took place in an urban area (Lorient, France), the receiver antenna installed in height while the transmitter was placed in a moving car, with antennas (one for the GPS and one for the QCSP transmitter) fixed on its roof. The GPS track was also saved by an independent device in

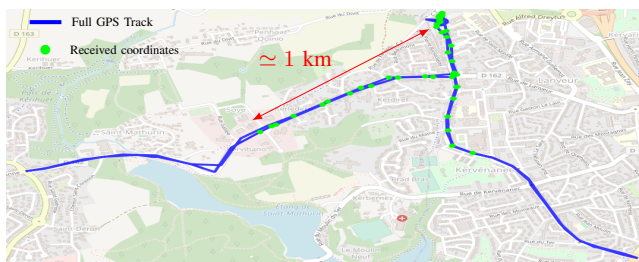


Fig. 2. QCSP transmitter successive locations (green circles) extracted from successfully received frames and the actual path (blue line) followed by the vehicle drawn on OpenStreetMap.

the car. The GPS coordinates extracted from the successfully decoded frames and vehicle GPS track have been plotted on a map using data from OpenStreetMap [13]. Results show that current QCSP transmissions are reliable for a distance up to 1 km between a moving transmitter and a static in-height receiver in an urban area. Such an area represents a particular challenge, each building being both a reflective surface and an obstacle. Despite these difficulties, the QCSP real-time system managed to successfully transmit data, for an energy of $0.5 \mu\text{J}$ per information bit.

IV. CONCLUSION

The reported work demonstrated that the QCSP emission and reception tasks can be implemented in real-time and at reasonable costs, allowing reliable transmissions at low SNR without the need of a preamble. The transmitter's first real-time implementation reached a throughput of a few kbps, compatible with the LPWAN context, on an embedded system. The first real-time implementation of the receiver required more effort and more expensive devices, however it achieved already compliant performances. It reaffirms the suitability of the QCSP waveform in contexts such as wireless sensor networks, where low-end sensor nodes send data to higher-end server nodes. In the future, a hardware implementation of the detection will be embedded on the USRP X310, to reduce the amount of data streamed from the USRP to the laptop and leverage parallelization techniques developed in [8]. In parallel, the remaining stages of the QCSP receiver will be ported to C/C++ to improve its performances and stability.

REFERENCES

- [1] G. Durisi and al., "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets," *Proceedings of the IEEE*, vol. 104, no. 9, 2016.
- [2] Y. Polyanskiy, "Asynchronous Communication: Exact Synchronization, Universality, and Dispersion," *IEEE Trans. on Information Theory*, vol. 59, no. 3, March 2013.
- [3] A. Azari and al., "Grant-Free Radio Access for Short-Packet Communications over 5G Networks," in *Proceedings of GLOBECOM*, 2017.
- [4] B. Bloessl and F. Dressler, "mSync: Physical Layer Frame Synchronization without Preamble Symbols," *IEEE Trans. on Mobile Computing*, vol. 17, no. 10, 2018.
- [5] P. Walk and al., "MOCZ for Blind Short-Packet Communication: Practical Aspects," *IEEE Trans. on Wireless Communications*, vol. 19, no. 10, October 2020.
- [6] C. Monière, K. Saied, B. Le Gal, and E. Boutillon, "Time sliding window for the detection of CCSK frames," in *Proceedings of SiPS*, 2021, IEEE.
- [7] K. Saied, *Quasi-Cyclic Short Packet (QCSP) Transmission for IoT*, Theses, Université Bretagne Sud, Mar. 2022.
- [8] C. Monière, B. Le Gal, and E. Boutillon, "Efficient implementations of a QCSP communication system," in *Proceedings of DASIP "in press"*, 2022, .
- [9] G.M. Dillard and al., "Cyclic code shift keying: A low probability of intercept communication technique," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 3, 2003.
- [10] K. Saied, Ali Ghouwayel, and E. Boutillon, "Time-synchronization of CCSK short frames," in *Proceedings of WiMob*, 2021.
- [11] O. Abassi, L. Conde-Canencia, M. Mansour, and E. Boutillon, "Non-binary low-density parity-check coded cyclic code-shift keying," in *Proceedings of WCNC*, Shanghai, China, April 2013.
- [12] Ettus Research, *USRP Hardware Driver and USRP Manual*, 4.1.0 edition, 2022.
- [13] "OpenStreetMap," <https://www.openstreetmap.org/copyright>.