Coexistence of Pull and Push Communication in Wireless Access for IoT Devices

Coexistence of Pull and Push Communication in Wireless Access for IoT Devices

Sara Cavallero*, Fabio Saggese†, Junya Shiraishi†, Shashi Raj Pandey†, Chiara Buratti*, Petar Popovski†

*WiLab/CNIT and University of Bologna, Italy. †Department of Electronic System, Aalborg University, Denmark.
*{s.cavallero, c.buratti}@unibo.it, †{fasa, jush, srp, petarp}@es.aau.dk.

***Abstract*—We consider a setup with Internet of Things (IoT), where a base station (BS) collects data from nodes that use two different communication modes. The first is *pull-based*, where the BS retrieves the data from specific nodes through queries. In addition, the nodes that apply pull-based communication contain a wake-up receiver: upon a query, the BS sends wake-up signal (WuS) to activate the corresponding devices equipped with wake-up receiver (WuDs). The second one is *push-based* communication, in which the nodes decide when to send to the BS. Consider a time-slotted model, where the time slots in each frame are shared for both pull-based and push-based communications. Therein, this coexistence scenario gives rise to a new type of problem with fundamental trade-offs in sharing communication resources: the objective to serve a maximum number of queries, within a specified deadline, limits the transmission opportunities for push sensors, and vice versa. This work develops a mathematical model that characterizes these trade-offs, validates them through simulations, and optimizes the frame design to meet the objectives of both the pull- and push-based communications.**

***Index Terms*—Internet of things, goal-oriented communications, pull-based communications, wake-up radio, medium access control.**

I. Introduction

The transition from 5G to 6G communication affects significantly the Internet of Things (IoT) domain, where intelligence will play an increasingly large role in the overall system setup, supporting diverse applications [1]. The increasing intelligence within the communication nodes and devices directs the evolution towards semantic and goal-oriented communication, fostering collaboration and context-aware information sharing [2]. Within this IoT framework, two distinct communication modes take place: *pull-based* communication, where the receiver determines who can send information, leading to scheduled transmissions; and *push-based* communication, where each device autonomously decides when to transmit to the receiver, following the principles of random access. Depending on the specific application requirements, one communication method may be more suitable than the other; however, in a heterogeneous scenario with different types of IoT devices, these two communication methods can coexist. This coexistence can result in a balanced and adaptive system, optimizing and enhancing the overall performance.

This work was partly supported by the Villum Investigator Grant "WATER" from the Velux Foundation, Denmark, partly by the Horizon Europe SNS "6G-GOALS" project with grant 101139232, and partly by the Horizon Europe SNS "6G-XCEL" project with Grant 101139194.

Only a limited number of studies in the literature have explored their interaction. In [3], a solution to a distributed convex optimization problem is presented using the push-pull gradient method, involving the push of information about gradients from agents to neighbours and the pull of decisions in the opposite direction. The authors in [4] proposed an analytical model for optimizing push- and pull-based communication, involving interaction between the actuator and the BS to share their state information. While these works treat the two methods of communication as interdependent, where the action to be performed on the push depends on the result of the pull and vice versa, we approach them as separate processes that must coexist within the same system.

In our study, we examine a network featuring a 5G-enabled base station (BS) managing wireless traffic from two device classes: the nodes operating with pull-based communication and those utilizing push-based communication. For pull-based communication, this paper considers the system integrating wake-up radio, in which ultra low-power wake-up receiver [5] is installed into IoT devices. The wake-up receiver keeps active monitoring of the wake-up signal (WuS) while each node turns off the main radio, saving the wasteful energy consumption of nodes during the idle period. This paper focuses on the well-investigated identity-based wake-up (IDWu) [5, 6], which enables the BS to retrieve the data from the target devices based on its ID. This is suitable for a variety of practical scenarios where the external entity sends a query request to the BS through the cloud. For push-based communication, devices initiate data transmission with random access to the channel once a data packet is available in their queue. Therein, pull-based communication prioritizes goal-oriented behaviour to maximize query satisfaction, while push communication aims to maximize the success probability and, consequently, the throughput of push sensors. To the best of our knowledge, this is the first work that exploits the integration of these two communication modes and identifies possible trade-offs between their performance metrics. The performance of this coexistence communication is assessed mathematically, and the validity of the implemented model is evaluated by comparing it with simulations.

Notation: $k \sim \text{Poiss}(\mu)$ is Poisson distributed random variable (r.v.) with mean value μ, i.e., $k \in \mathbb{N}$ has probability mass function (pmf) $\mathcal{P}(k,\mu) = \frac{\mu^k e^{-\mu}}{k!}$; $\boldsymbol{B}(m,E)$ denotes the Erlang-B formula with m servers and normalized ingress load

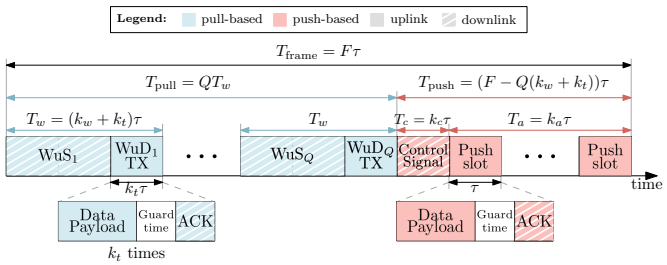


Fig. 1: Example of time-frame structure for pull and push communication coexistence, with Q scheduled communications for WuDs, k_a slots for push-based access, $k_w = 2$, $k_t = k_c = 1$.

E. $\lfloor x \rfloor$ denotes the closest integer lower than x.

II. System Model

We consider an IoT scenario where a BS manages access control for two type of devices, or nodes, communicating over a shared wireless link: some nodes are tailored for pull-based communication, while others are for push-based communication. Besides receiving uplink (UL) data from various devices within the network, the BS also manages queries coming through the cloud, requesting data from specific pull-based devices equipped with wake-up receiver (WuDs). To simplify the analysis and to show a possible trade-off of the coexistence of pull-push communication, we assume a collision channel and a narrow-band transmission, where a single frequency channel is reserved for both UL and downlink (DL) communication[1].

As depicted in Fig. 1, the system operates in a time-slotted framework organized into frames, where the duration of each frame, denoted as T_{frame}, is split into two segments: T_{pull} and T_{push} dedicated to pull and push-based communication, respectively. Each frame is further divided into F slots, each of duration equal to τ [s], such that $\tau F = T_{\text{frame}}$. In order to be compliant with the 3rd Generation Partnership Project (3GPP) standard [8], τ is assumed to be a multiple of an Orthogonal Frequency Division Multiplexing (OFDM) symbol, as in [9, 10]. Moreover, we assume that a single slot has been designed to accommodate an UL data transmission, the corresponding Acknowledgement (ACK) reception, plus an additional guard time to switch from UL to DL transmissions. In the following, the details of pull and push communications are given.

A. Pull-based communication

Regarding the pull-based communication, we assume that data collection is subject to the reception of specific queries coming through the cloud. These queries contain an identifier (ID) of the device to pull data from (selected depending on the task to be performed), and a deadline that needs to be satisfied, denoted as L_{th} [frames]. If a query is not *successfully* served before the deadline, it is *discarded*. The queries are coming through the cloud according to a Poisson Point Process (PPP) with an average arrival rate λ_q [queries/s]. Accordingly, the number of queries in a frame is $n_q \sim \text{Poiss}(\overline{n}_q)$ with $\overline{n}_q = \lambda_q T_{\text{frame}}$. During each frame, the BS accumulates the queries, and, in the subsequent frame, it tries to serve them retrieving the data from a subset of WuDs. To retrieve the data from the subsets of target nodes without collisions, this paper applies IDWu, in which a unique wake-up ID is embedded into a WuS, allowing the BS to collect data from individual nodes based on its own scheduling. The WuSs are sent based on the order of queries' arrival at the BS – First In First Out (FIFO) principle. Once the corresponding WuD has received the WuS, it switches on its primary radio and transmits its data. Without loss of generality, we assume that k_w slots are needed for the transmission of the WuS and the activation of the WuD primary radio, while the WuD data transmission occupies the subsequent k_t slots. Therefore, for each WuD, the scheduled communication has a duration $T_w = (k_w + k_t)\tau$. Consequently, the time dedicated to the pull-based communication is

$$T_{\text{pull}} = QT_w = Q(k_w + k_t)\tau, \tag{1}$$

where Q represents the number of queries that can be effectively served within that frame. Remark that Q is a fundamental design parameter of the system, representing the amount of resources reserved for pull-based communication. For simplicity, we assume that each query received in the t-th frame must be addressed before the end of the $(t+1)$-th frame, meaning queries have a one-frame deadline ($L_{\text{th}} = 1$). Hence, if the BS cannot fulfill a query for a specific WuD within the current frame due to insufficient resources, it is discarded.

B. Push-based communication

The sensors involved in the push-based communication operate intermittently, taking measurements of the environment according to a PPP and deciding autonomously whether to transmit the measured data. Accordingly, the generated packets follow a PPP with mean arrival rate λ_p [packet/s], and the number of packets in a frame $n_p \sim \text{Poiss}(\overline{n}_p)$, with $\overline{n}_p = \lambda_p T_{\text{frame}}$.

Unlike pull-based communication, push-based communication employs contention-based grant-free access, within the portion of the frame T_{push}, using a Framed-ALOHA scheme. At the beginning of T_{push}, a subset of k_c slots is dedicated to the transmission of a control message – e.g., a beacon – from the BS to signal readiness to receive push data[2]; this control message has a duration of $T_c = k_c\tau$. The remaining time, $T_a = k_a\tau$, is available for nodes to access the channel randomly selecting a subset of the k_a slots to transmit their queued packets, one per slot. If two or more nodes choose the same slot, a collision occurs, and all the packets transmitted in that slot are lost[3]. Accordingly, the time reserved for push

[1]Remark that this time-frequency structure is equivalent to the one used in 5G New Radio (NR) definition, assuming a single resource block (RB) is used [7]. The extension to the multi-carrier system is left for future studies.

[2]From the 3GPP standard, this can be seen as similar to the transmission of the Synchronization Signal (SS) of the RACH procedure [8].

[3]For the sake of simplicity, no re-transmission technique such as hybrid automatic repeat request (HARQ) is considered.

communication is $T_{\text{push}} = (k_c + k_a)\tau$ which can be re-written as a function of Q, following

$$T_{\text{push}} = T_{\text{frame}} - T_{\text{pull}} = (F - Q(k_w + k_t))\tau. \quad (2)$$

C. Pull and push trade-off

In the proposed scenario, we need to manage the finite time resources of the Medium Access Control (MAC) frame shared between pull and push nodes taking into account the different requirements of the two traffic: pull-based communication aims to maximize the average number of successfully served queries, while push-based communication wants to maximize the throughput. To reach these two objectives, it is necessary to find a trade-off on the time resources allocated for the two communications, considering that T_{frame} remains fixed. A high number of queries collected in the previous frame requires reserving more slots for pull communication to wake up all the intended WuDs and ensure their transmissions. In this case, it becomes essential to increase T_{pull}, which simultaneously decreases T_{push}. In turn, this increases the collision probability for UL push transmissions, because of the lower number of available slots, leading to a lower throughput of the push devices. These communication requirements can be translated into a unique metric, accounting for both, the *success probability* of serving the queries and the *success probability* of push access (see Section III). The goal is to determine the values of T_{pull} and T_{push} able to achieve a trade-off for the communication coexistence. Since $T_{\text{frame}} = T_{\text{pull}} + T_{\text{push}}$, and both terms depends on the number of queries that can be served (see eq. (1) and (2)), our goal translates into finding the values of Q striking a trade-off between success probability of queries and push packets, taking into account that $k_a = F - k_c - Q\,(k_w + k_t)$ and $Q = \left\lfloor \frac{F - k_c - k_a}{k_w + k_t} \right\rfloor$.

III. Performance Evaluation

Throughout this section, we aim to relate the metrics of pull- and push-based communications with the value of Q controlling the portion of time reserved to the two traffic.

A. Success probability of serving the queries

In pull-based communication, the BS must timely collect fresh data from the WuD, meeting a single-frame deadline. If the BS cannot obtain data from a specific WuD by the deadline, it discards the query request. This occurs when the available resources for pull-based communication, denoted as Q, are insufficient to handle the number of queries received in the previous frame.

To analyze the success probability of serving the queries, we remark that the current setting is equivalent to a multi-server queuing model with null queuing capacity. Following Kendall's notation, this can be represented by a $M/D/Q/0$ queue model, according to the fact that only the first Q queries can be served, and the others are discarded. Under this model, the Erlang-B formula represents the discarding probability when the number of sources is infinite [11, 12]. Hence, the success probability of serving the queries is approximated as:

$$p_s^{(q)} \approx 1 - \boldsymbol{B}(Q, \overline{n}_q) = 1 - \frac{\frac{\overline{n}_q^Q}{Q!}}{\sum_{i=0}^{Q} \frac{\overline{n}_q^i}{i!}}. \quad (3)$$

The approximation becomes exact when the average number of queries $\overline{n}_q$ (pull sources), grows to infinity.

From this metric, it is possible to find the average number of successfully served queries. Since each query has a success probability independent from query to query, and from the thinning property of PPP [13], the number of successfully served queries, $n_s = n_q\, p_s^{(q)}$, is still Poissonian, with mean:

$$\overline{n}_s = \sum_{n_q=0}^{\infty} n_q\, p_s^{(q)}\, \mathcal{P}(n_q, \overline{n}_q) = \overline{n}_q\, p_s^{(q)}. \quad (4)$$

B. Average Throughput for push-based communication

For the push-based communication, we are interested in analyzing the average throughput, S_{push} [packets/s], defined as the average number of packets per unit of time successfully received by the BS from the push devices. Consequently, S_{push} is related to the success probability of push access $p_s^{(p)}$.

In order to formalize the latter, let us denote the probability of successful access in a frame conditioned to the knowledge of the number of transmitted push packets n_p as $p_{s|n_p}^{(p)}$. Under the Framed ALOHA scheme and collision channel assumptions, $p_{s|n_p}^{(p)}$ depends on the available slots for the push k_a, with the assumption that at least one slot is available for push access, i.e., $k_a > 0$. If $k_a = 1$, the probability of access is 1 if $n_p \leq 1$, and 0 otherwise, due to the unavoidable collisions; if $k_a \geq 1$, the probability of accessing the channel is exactly 1 with $n_p = 0$, because no collisions can occur, while it is $(1 - 1/k_a)^{n_p - 1}$ for $n_p \geq 1$, representing the case that only a single packet selects a slot. Hence, we can write:

$$p_{s|n_p}^{(p)} = \begin{cases} 0, & \text{if } k_a = 1, n_p > 1, \\ \left(1 - \frac{1}{k_a}\right)^{n_p - 1}, & \text{if } k_a > 1, n_p \geq 1, \\ 1, & \text{if } \begin{cases} k_a = 1, \\ n_p \leq 1, \end{cases} \text{ or } \begin{cases} k_a > 1, \\ n_p = 0. \end{cases} \end{cases} \quad (5)$$

From (5), it is possible to compute the success probability of push access considering two different cases: $k_a = 1$ and $k_a > 1$. If $k_a = 1$, the access probability is equal to the probability that 0 or 1 packets are transmitted, i.e., $p_s^{(p)} = \mathcal{P}(0, \overline{n}_p) + \mathcal{P}(1, \overline{n}_p)$. If $k_a > 1$, the probability of access can be obtained by applying the law of total probability, i.e., $p_s^{(p)} = \sum_{n_p=0}^{\infty} p_{s|n_p}^{(p)}\, \mathcal{P}(n_p, \overline{n}_p) = \mathcal{P}(0, \overline{n}_p) + \sum_{n_p=1}^{\infty} \left(1 - \frac{1}{k_a}\right)^{n_p - 1} \mathcal{P}(n_p, \overline{n}_p)$. This results in:

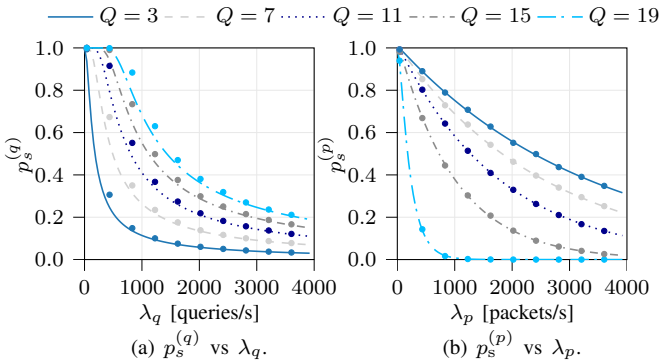
$$p_s^{(p)} = \begin{cases} (1 + \overline{n}_p)\, e^{-\overline{n}_p}, & \text{if } k_a = 1, \\ e^{-\overline{n}_p}\, \frac{k_a\, e^{\frac{(k_a - 1)\,\overline{n}_p}{k_a}} - 1}{k_a - 1}, & \text{if } k_a > 1. \end{cases} \quad (6)$$

Similarly to the previous case, based on (5), we can derive the average throughput for push nodes as in [14, 15], obtaining

$$S_{\text{push}} = \frac{1}{T_{\text{frame}}} \sum_{n_p=0}^{\infty} n_p\, p_{s|n_p}^{(p)}\, \mathcal{P}(n_p, \overline{n}_p) = \frac{\overline{n}_p}{T_{\text{frame}}}\, e^{-\frac{\overline{n}_p}{k_a}}. \quad (7)$$

TABLE I: Simulation parameters.

Parameter	Description	Value
τ	Slot duration [s]	0.25 ms[4]
F	Slots in a frame	101
T_{frame}	Frame duration [s]	25.25 ms
k_w	Slots for WuS	4[5]
k_t	Slots in for WuD TX	1
k_c	Slots for control part	1
k_a	Slots for Framed ALOHA TX	1



(a) $p_s^{(q)}$ vs λ_q. (b) $p_s^{(p)}$ vs λ_p.

Fig. 2: Validation of analytical models with different Q values.

C. Success probability trade-off

Instead of jointly maximizing both pull and push success probabilities, we characterize a performance trade-off defining the weighed average success probability for the communication coexistence as

$$p_s = w_q\, p_s^{(q)} + w_p\, p_s^{(p)}, \tag{8}$$

where $w_q \in [0,1]$ and $w_p \in [0,1]$ are positive weights useful to target requirements of pull- and push-based communications, respectively. Since both $p_s^{(q)}$ and $p_s^{(p)}$ are functions of Q, it is possible to find the value of Q that maximizes p_s, under a specific set of weights and system settings. As a show case, we select $w_q = \frac{\lambda_q}{\lambda_q+\lambda_p}$ and $w_p = \frac{\lambda_p}{\lambda_q+\lambda_p}$, considering the fairness in terms of traffic load for both pull and push nodes.

IV. Numerical Results

In this section, we numerically benchmark the performance of the proposed system. We build a custom simulator that implements both the pull and the push-based communications, according to the system model described in Section II. Each simulation lasts 10^5 T_{frame}; the averaged values are plotted in the following figures. Unless stated otherwise, Table I reports the parameter values used for numerical results.

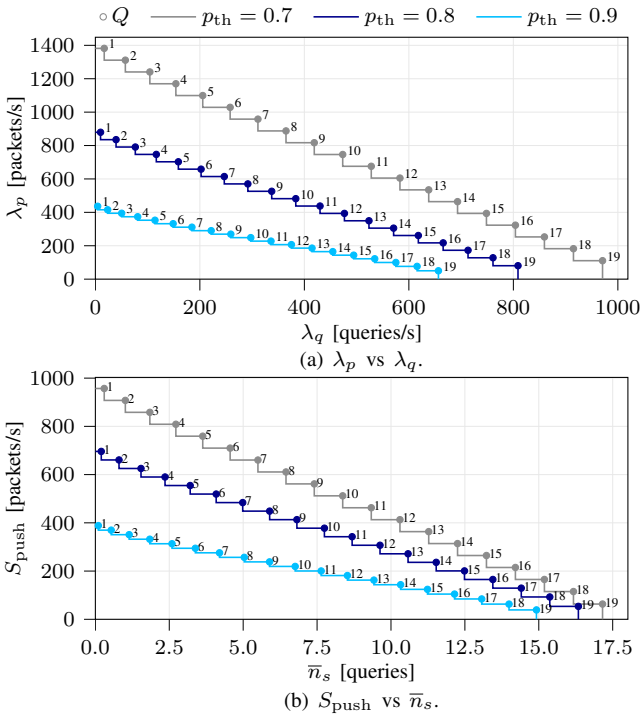
A. Validation of the analytical model

Fig. 2a shows the success probability of serving the queries, $p_s^{(q)}$ versus the average arrival rate of the queries, λ_q, while Fig. 2b depicts the success probability $p_s^{(p)}$ against the average arrival rate for the push-based communication, λ_p. These results are obtained from both our theoretical analysis (lines) and computer simulation (circular symbols) considering different Q values. Fig. 2b shows a perfect correspondence between the theoretical analysis and the numerical simulation, validating our results. In Fig. 2a, there is a slight difference between the analytical and simulation results for a range of low values of λ_q, due to the infinite population of sources assumption of the Erlang-B formula. Nevertheless, eq. (3) is a tight lower bound of the actual success probability $\forall \lambda_q$, justifying the use of our analytical model to characterize the system performance.

Finally, the results already show that a fundamental trade-off on time resources exists: the higher is Q, the greater $p_s^{(q)}$, whereas the lower the $p_s^{(p)}$, and vice versa. This result indicates the importance of the choice of the parameter Q considering both types of traffic.

[4]According to 3GPP concept of mini-slots [8], we assume a slot= 7 OFDM symbols and a sub-carrier spacing $\Delta_f = 30$ kHz, working in FR1.

[5]Value set according to 3GPP [6, Table 7.1.2.2-5].



(a) λ_p vs λ_q. (b) S_{push} vs $\overline{n}_s$.

Fig. 3: Design guidelines for pull and push communications when $p_s^{(q)} = p_s^{(p)} = p_{\text{th}}$.

B. Design guidelines for coexistence of pull and push-based communication

Fig. 3 shows the achievable performance of the system constrained to a target success probability for pull and push traffic. To obtain the plot, we set a requirement on the success probabilities: $p_s^{(q)} = p_s^{(p)} = p_{\text{th}}$. For each Q value, we compute the maximum arrival rates $\lambda_q^{(\max)}$ and $\lambda_p^{(\max)}$ that can be supported satisfying this constraint. Knowing the arrival rates, for each Q value, we determine the maximum number of successfully served queries $\overline{n}_s(\lambda_q^{(\max)})$ and the maximum throughput $S_{\text{push}}(\lambda_p^{(\max)})$ using (4), and (7), respectively.

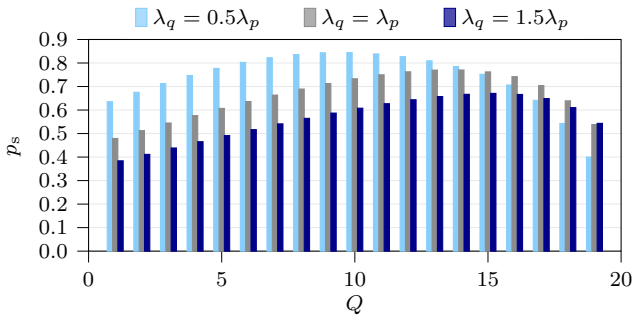


Fig. 4: p_s vs Q for λ_p = 500 packets/s and different λ_q values.

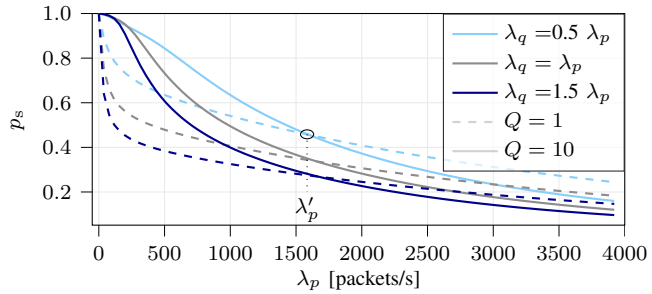


Fig. 5: p_s vs λ_p for different λ_q (color coded) and Q values (solid and dashed lines).

Fig. 3a displays the acceptable traffic λ_p vs λ_q for $p_{\text{th}} = [0.7; 0.8; 0.9]$. Increasing Q generally leads to a decrease in the acceptable λ_p, as λ_q rises, accommodating higher pull traffic to meet the p_{th} requirement. Lower p_{th} allows for higher acceptable arrival rates. Fig. 3b shows the throughput S_{push} as a function of $\overline{n}_s$ for the same p_{th} values. The behaviour of this graph aligns with the previous one, where increasing Q and/or decreasing p_{th}, directly correlates with an increase in λ_q and therefore a higher number of satisfied queries $\overline{n}_s$. This corresponds with a simultaneous decrease in the supported input traffic for push communication, resulting in a lower achieved throughput. For example, considering $p_{\text{th}} = 0.8$ and $Q = 2$, the success requirement is met up to $\lambda_q = 39$ queries/s, and $\lambda_p = 835$ packets/s. This results in an achievable throughput $S_{\text{push}} = 660$ packets/s and an average number of successfully served queries $\overline{n}_s = 0.5$. If λ_q increases, raising Q to 3 becomes necessary, leading to a decrease in input push traffic to $\lambda_p = 791$ packets/s corresponding to $S_{\text{push}} = 625$ packets/s. The region under each curve indicates the set of achievable performance for push and pull at each Q value, facilitating the selection of suitable parameters based on specific requirements.

C. Trade-off pull/push-based communication

Fig. 4 shows the behaviour of the weighted average success probability p_s of eq. (8) as a function of Q for $\lambda_p = 500$ packets/s and $\lambda_q = [0.5; 1; 1.5]\lambda_p$. Results show that the highest p_s is obtained with $Q = [10; 14; 15]$ when $\lambda_q = [0.5; 1; 1.5]\lambda_p$, respectively. As expected, increasing λ_q, the maximum of p_s is achievable increasing Q, meaning that more slots dedicated to the pull-based communication are needed.

Finally, in Fig. 5, we evaluate p_s as a function of λ_p for $\lambda_q = [0.5; 1; 1.5]\lambda_p$ and $Q \in \{1, 10\}$. This graph shows that p_s is a monotonically decreasing function of both λ_p and λ_q. Notably, when λ_q is held constant, the curves for $Q = 1$ and $Q = 10$ intersect at a specific arrival rate λ'_p. When $\lambda_p < \lambda'_p$, it is advisable to maintain $Q = 10$, allocating more slots for pull communication. Conversely, for $\lambda_p > \lambda'_p$, reducing Q to 1 is preferable. In this scenario, the impact of push communication traffic is more significant on system performance, and allocating more slots to push results in an enhancement of overall performance. For clarity, the graph shows only two Q values, but it can be extended to all Q values, facilitating the identification of the optimal solution based on the system's input load (λ_p and λ_q).

V. Conclusions

This paper studies the integration of pull-enabled WuDs and intermittently active push sensors operating within the same time frame structure in an IoT setting. A mathematical model, validated through simulations, highlights the interdependence of both communications and the need of adjusting the time allocation reserved for pull and push to achieve optimal performance. Based on the knowledge of the incoming push and pull traffic, the time allocation that optimally balances the overall performance in a single frame has been found. Future works will investigate the optimal performance considering a longer time horizon, as well as learning that is capable to capture the correlations among the traffic arrivals and data.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, 2020.

[2] E. Calvanese Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *Comput. Netw.*, vol. 190, p. 107930, 2021.

[3] S. Pu *et al.*, "Push–pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, 2021.

[4] P. Talli *et al.*, "Push- and pull-based effective communication in cyber-physical systems," *arXiv preprint arXiv:2401.10921*, 2024.

[5] R. Piyare *et al.*, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2117–2157, 2017.

[6] 3GPP, "Study on low-power Wake-up Signal and Receiver for NR," 3rd Generation Partnership Project (3GPP), Tech. report (TR) 38.869, 10 2023, release 18.

[7] ——, "NR; Physical layer procedures for data," 3rd Generation Partnership Project (3GPP), Tech. specification (TS) 38.214, 10 2022, version 15.0.0.

[8] ——, "Study on New Radio (NR) access technology," 3rd Generation Partnership Project (3GPP), Tech. report (TR) 38.912, 07 2017, version 14.1.0.

[9] G. Cuozzo *et al.*, "Enabling URLLC in 5G NR IIoT networks: A full-stack end-to-end analysis," in *Joint Eur. Conf. Netw. and Commun. & 6G Summit (EuCNC/6G Summit)*, 2022.

[10] J. Zhang *et al.*, "Low latency DWBA scheme for mini-slot based 5G new radio in a fixed and mobile converged TWDM-PON," *J. Lightw. Technol.*, vol. 40, no. 1, pp. 3–13, 2022.

[11] M. U. Thomas, "Queueing systems. volume 1: Theory (leonard kleinrock)," Jul 1976.

[12] T. Bonald, "The erlang model with non-poisson call arrivals," in *Joint Int. Conf. Meas. and Model. Comput. Syst.* ACM, 2006, p. 276–286.

[13] W. Weil, Ed., *Spatial Point Processes and their Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–75.

[14] S. Cavallero *et al.*, "Applying carrier sense multiple access to industrial IoT at terahertz frequencies," *IEEE Internet Things J.*, pp. 1–1, 2023.

[15] S. Mignardi and C. Buratti, "Modeling UAV-based IoT clustered networks for reduced capability UEs," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10 266–10 279, 2023.