Aalborg Universitet

AALBORG UNIVERSITY

Random Access Protocols for Correlated IoT Traffic Activated by Semantic Queries

Kalør, Anders E.; Popovski, Petar; Huang, Kaibin

Published in:
2023 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt 2023

DOI (link to publication from Publisher):
10.23919/WiOpt58741.2023.10349893

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Kalør, A. E., Popovski, P., & Huang, K. (2023). Random Access Protocols for Correlated IoT Traffic Activated by Semantic Queries. In *2023 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, WiOpt 2023* (pp. 643-650). IEEE (Institute of Electrical and Electronics Engineers). https://doi.org/10.23919/WiOpt58741.2023.10349893

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: July 04, 2025

Random Access Protocols for Correlated IoT Traffic Activated by Semantic Queries

Anders E. Kalør*†, Petar Popovski†, and Kaibin Huang*
*Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
Email: {aekaloer,huangkb}@eee.hku.hk
†Department of Electronic Systems, Aalborg University, Denmark
Email: {aek,petarp}@es.aau.dk

***Abstract*—As IoT devices become increasingly advanced and equipped with sensors such as cameras and microphones, the collection of massive data streams, produced in real-time, becomes challenging. In many cases only a small fraction of the collected data might be relevant, e.g., if cameras are used to search for a specific object. In this paper, we introduce and analyze a set of random access protocols in which the transmitting IoT devices are activated by semantic queries. This can be seen as a semantic data sourcing random access: each device computes a matching score that characterizes the relevance of its current observation and, if the matching score exceeds a threshold, the device transmits its observation over a random access collision channel to an edge node. We study two random access transmission policies. The first is the classical slotted ALOHA policy, while the other is able to exploit semantic correlation between the device observations. Furthermore, we show how the protocol can be integrated with machine learning-based query and matching score functions to capture the semantic content of, say, images. The numerical results show that the proposed protocol is able to effectively filter the device observations, such that mostly relevant data is received. Overall, the protocol is promising for collecting data in real-time from massive IoT networks based on the semantic content of sensor observations.**

I. Introduction

Wireless Internet of Things (IoT) devices are becoming increasingly advanced and equipped with a multitude of sensors, allowing them to monitor complex and often multi-modal signals, such as images, video, and audio/speech. Combined with advanced processing in the cloud or on an edge server, usually relying on machine learning techniques, a fleet of IoT devices can be used for a wide range of applications, such as taking control decisions, surveillance, detection of accidents, fault detection in smart grids and water networks, and others. However, due to the limited capacity of the wireless channel as well as possible power constraints, collecting the massive, continuous stream of data that the sensors produce in real-time necessitate that the data are carefully filtered [1], [2].

Recently, the problem of collecting real-time data has motivated the definition of the Age of Information (AoI) metric, which characterizes the average freshness of information at the destination subject to the communication constraints [3], [4]. Subsequently, the AoI of numerous communication schemes has been analyzed, and a multitude of protocols that aim to minimize the AoI have been proposed, see e.g., [4] and references therein. An inherent assumption in AoI is that all data are relevant as long as they are fresh. While this may be the case in some monitoring scenarios, such as when the data are collected for recording purposes, many applications have other relevance criteria. For this reason, the concept of AoI has been generalized to various variants that consider more general relevance metrics, such as Age of Incorrect Information [5], Urgency of Information [6], Semantics of Information [7], and Query AoI [8].

Regardless of the specific AoI variant, the metric is designed for *proactive* communication, which is the typical paradigm for IoT. In proactive communication, the monitored data are collected before they are requested by an application, and the application has access only to the data that have been collected prior to the request. Alternatively, data can be collected *reactively* (or pull-based), i.e., collected *after* they have been requested by an application [2]. Reactive communication has been considered in e.g., in-network query processing [9], [10], and in content-based wake-up radios [11].

The examples mentioned above study the case where data are requested based on declarative queries, such as data that fall within a given range. However, these requests are not particularly useful for complex signals such as images, in which the content is more challenging to extract. Instead, such signals require *semantic* queries, which can query, e.g., objects in an image. This problem was addressed in [12] under the framework *semantic data sourcing* (SEMDAS), which leverages machine learning to reactively request data that are relevant for a given task, such as classification. SEMDAS can thus be seen as an instance of semantic communications [13], [14] in which the edge server broadcasts a semantic query in the downlink. Based on the query and their current observations, the devices compute *matching scores*, which are sent back to the edge server. Based on the received matching scores, the edge server schedules a subset of the devices to transmit their full observations in the uplink. Compared to proactive communication, this approach has the advantage that only relevant data are collected, but it comes at the cost of increased latency. Nevertheless, it is an attractive scheme when only a small fraction of the generated sensor data are relevant and/or the relevance is hard to predict.

As a concrete example where the reactive strategy is beneficial, consider an IoT network comprising a large number of cameras deployed within a city. Suppose there have been reportings of a wild boar, and we would like to track the boar

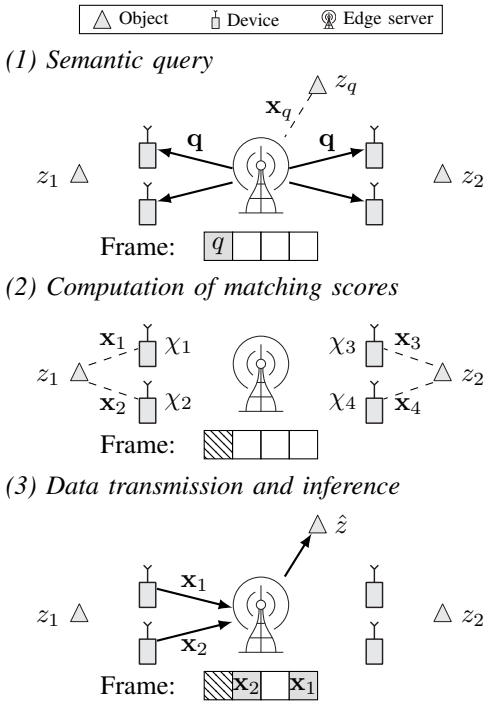

Fig. 1. The proposed protocol: *(1)* The edge server broadcasts a semantic query, $\mathbf{q}$, in the first slot based on the query observation $\mathbf{x}_q$; *(2)* Each device computes a matching score χ_m characterizing how well its observation $\mathbf{x}_m$ matches the query; *(3)* The devices whose matching score exceeds a threshold transmit their observations in the remaining slots using a random access protocol, and the edge server performs inference based on the received observations.

with the cameras. In the proactive case, we would have to hope that the boar has been captured and transmitted by some of the cameras, which is unlikely if, for instance, we collect an image from each camera every 60 seconds. On the other hand, if we can request images only from the cameras that currently observe a wild boar, as in the reactive case, then we would not only be much more likely to receive images of the boar, but also to receive the most fresh observations that will allow us to estimate its location with high accuracy. Furthermore, the sensors can operate in a low-power mode between queries. Under the SEMDAS framework, the query could be an image of a wild boar obtained from some existing database. After receiving the relevant observations, the edge server could perform classification to identify the specific boar, assess the risks and determine an appropriate action. Nevertheless, SEMDAS requires the edge server to collect matching scores from all devices in order to determine which ones to be scheduled for transmission. This results in a significant overhead, especially if the number of devices is large and only a small fraction of their observations are relevant.

To address this issue, in this paper we propose a novel semantically activated random access protocol inspired by SEMDAS: the devices determine independently, based on their matching scores, whether their data are relevant and subsequently transmit them over a shared random access channel, see Fig. 1. We study two threshold-based random access policies under the slotted collision model: (1) classical slotted ALOHA policy and (2) protocol that exploits the semantic correlation between the device observations. We evaluate their performance through both a toy scenario and a more realistic scenario that rely on machine learning to compute the matching scores as in SEMDAS. It is seen that the proposed protocol is able to retrieve a large number of relevant observations while filtering out the irrelevant ones, making it a promising strategy for semantic queries in IoT. Furthermore, the random access policy that exploits correlation generally performs better than the classical slotted ALOHA policy.

The remainder of the paper is organized as follows. Section II introduces the system model and the overall problem definition. We present the two random access policies for SEMDAS in Section III, and demonstrate how they can be integrated with a machine learning-based SEMDAS model in Section IV. Numerical results are presented in Section V, and finally the paper is concluded in Section VI.

II. System Model and Problem Definition

A. Observation Model

We consider a scenario with M IoT devices and a single edge server. Each IoT device, indexed by $m = 1, 2, \ldots, M$, is equipped with a sensor, such as a camera, which generates an observation feature vector $\mathbf{x}_m \in \mathbb{R}^d$ of some object $z_m \in \mathcal{Z} = \{1, 2, \ldots, |\mathcal{Z}|\}$ in the environment according to the distribution $p(\mathbf{x}_m|z_m)$. The observations are assumed to be semantically correlated due to, e.g., spatial proximity, for instance caused by having multiple cameras observing overlapping views from different angles (this is often referred to as a multi-view scenario [15]). We model the correlation by grouping the devices into G disjoint clusters, so that the devices that belong to the same cluster produce observations of the same object. Denoting the IoT devices that belong to each cluster by $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_G$, where $\bigcup_{i=1}^{G} \mathcal{C}_i = \{1, 2, \ldots, M\}$, we have $z_m = z_i \forall m \in \mathcal{C}_i, i = 1, \ldots, G$. Using the clusters, the joint distribution of the observations can be factorized as

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_M) = \prod_{i=1}^{G} \sum_{z_i \in \mathcal{Z}} \left(\prod_{m \in \mathcal{C}_i} p(\mathbf{x}_m|z_i) p(z_i) \right). \quad (1)$$

In this initial work, we will make the simplifying assumption that the number of devices in each cluster is the same, i.e., $|\mathcal{C}_1| = |\mathcal{C}_2| = \cdots = |\mathcal{C}_G| \overset{\text{d}}{=} C$. While we will assume that at least the expected fraction of devices that observe the query object is known, the full cluster structure may or may not be known a priori. Out of the two random access policies that we will introduce in Section III, only one requires knowledge of the cluster structure. However, as we will show in the numerical results, the one that actively exploits the cluster structure performs significantly better than the one that does not, suggesting that such knowledge can be highly beneficial.

B. Channel Model

The IoT devices and the edge server are connected through a shared wireless link. The wireless interface is divided into frames comprising one downlink slot followed by L uplink slots, in which the devices can transmit their observations. We consider a collision channel, so that the only sources of error are collisions between multiple transmissions in the uplink, i.e., a transmission is correctly received whenever it is the only transmission in a given slot. Under this assumption, the received transmissions can be characterized by the distribution $p(\hat{\mathcal{X}}|\mathcal{X})$, where $\mathcal{X} \subseteq \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}$ denotes the set of transmitted observations and $\hat{\mathcal{X}} \subseteq \mathcal{X}$ is the set of received observations.

C. Semantic Data Sourcing

As illustrated in Fig. 1 and in line with SEMDAS [12], we assume that the edge server broadcasts a semantic query $\mathbf{q} \in \mathbb{R}^l$ to the IoT devices in the first slot of each frame. The semantic query requests observations similar to some query observation $\mathbf{x}_q$ of a (possibly unknown) query object z_q. The dimension l of the query is assumed to be much smaller than that of the observed features, i.e., $l \ll d$. Using the received query, each device computes a matching score $\chi_m = \chi(\mathbf{x}_m, \mathbf{q}) \in [0, 1]$ that aims to quantify the relevance of its current observation $\mathbf{x}_m$ given the query. For instance, the matching score could be designed to approximate the likelihood that $\mathbf{x}_m$ represents the query object z_q, i.e., $\chi(\mathbf{x}_m, \mathbf{q}) \approx \Pr(z_m = z_q|\mathbf{x}_m, \mathbf{q})$, or some other semantic similarity metric. We return to the problem of designing $\chi(\mathbf{x}_m, \mathbf{q})$ in Section IV. Based on the matching score, each device independently determines whether to transmit or not, according to some predefined transmission policy. In this initial work, we limit our focus to threshold policies, so that a device transmits its full feature vector whenever $\chi_m \geq \tau$ for some threshold $\tau \in [0, 1]$. Note that both z_m and z_q are random quantities from the perspective of the device, since z_m must be inferred from $\mathbf{x}_m$ and z_q from $\mathbf{q}$. Hence, $\mathbf{q}$ can be seen as a compressed representation of $\mathbf{x}_q$ suitable for determining whether an observation $\mathbf{x}_m$ is similar to $\mathbf{x}_q$. We will refer to observations of the query object as *positives* and observations of different objects as *negatives*.

Our end goal is to perform joint classification of the received observations. Specifically, given a set of received observations $\hat{\mathcal{X}}$ we aim to classify the observations into one of the $|\mathcal{Z}|$ object classes. To this end, we define the classifier $\mathcal{F}(\hat{\mathcal{X}})$ that takes a list of received observation features $\hat{\mathcal{X}}$ and outputs the most likely observed class $\hat{z}$ from the set $\mathcal{Z} = \{1, 2, \ldots, |\hat{\mathcal{Z}}|\}$.

III. Random Access based on Semantic Matching

We present two general random access strategies. To keep the optimization of the policies tractable, we will assume that the matching scores are conditionally independent across the devices given whether they observe the query object or not, and that we have access to (estimates of) the probability distributions of these conditional matching scores. We define the *true positive* (TP) and *false positive* (FP) probabilities as

$$\Pr(\mathrm{TP}|\tau) = \Pr(\chi_m \geq \tau | z_m = z_q), \tag{2}$$

$$\Pr(\mathrm{FP}|\tau) = \Pr(\chi_m \geq \tau | z_m \neq z_q). \tag{3}$$

These distributions can be obtained analytically or empirically (e.g. via training of a machine learning-based matching function).

Using the derived TP and FP probabilities, we can obtain the distribution of the number of transmitting devices $|\mathcal{X}|$ under the assumption that one of the clusters observe the query object. To simplify the notation, let us denote by $P = C$ the number of devices that observe the query object (positives) and by $N = M - C$ the number of devices that do not observe the query object (negatives). The expected number of transmitting positive and negative devices are then $\mathbb{E}[|\mathcal{X}_{\mathrm{TP}}| \mid \tau] = P \Pr(\mathrm{TP}|\tau)$ and $\mathbb{E}[|\mathcal{X}_{\mathrm{FP}}| \mid \tau] = N \Pr(\mathrm{FP}|\tau)$, respectively, and $\mathbb{E}[|\mathcal{X}| \mid \tau] = \mathbb{E}[|\mathcal{X}_{\mathrm{TP}}| \mid \tau] + \mathbb{E}[|\mathcal{X}_{\mathrm{FP}}| \mid \tau]$.

A. Slotted ALOHA (SA)

We start by considering the classical slotted ALOHA (SA) policy, in which each device with a matching score that exceeds the threshold τ transmits in a slot selected independently and uniformly at random among the L uplink slots. This policy makes no use of the assumed cluster structure, and therefore can be applied when the structure is unknown (although P and N must be known, at least approximately). Because the dependency between the number of transmissions in each slot makes the analysis hard, we will refrain from performing exact analysis of the channel distribution $p(\hat{\mathcal{X}}|\mathcal{X})$. Instead, we apply the Poisson approximation by assuming that the number transmissions in each slot is independent and Poisson distributed with mean $\lambda = (1/L)\,\mathbb{E}[|\mathcal{X}| \mid \tau]$. Under this assumption, the probability that a slot contains a successful transmission is given as

$$p_{\mathrm{succ}} = \frac{\mathbb{E}[|\mathcal{X}| \mid \tau]}{L} e^{-\mathbb{E}[|\mathcal{X}||\tau]/L}, \tag{4}$$

which is maximized when $\mathbb{E}[|\mathcal{X}| \mid \tau] = L$, resulting in a success probability of $p_{\mathrm{succ}} = 1/e$.

However, instead of maximizing the overall success probability, a more reasonable strategy is to pick the threshold τ so that the number of received *true positive* observations is maximized, which is equivalent to maximizing the probability of receiving a true positive in a given slot. Since collisions cause uniform erasures, the probability that a true positive is received is simply the fraction of transmitted true positives multiplied by the probability of success,

$$p_{\mathrm{TP}} = \mathbb{E}\left[\frac{|\mathcal{X}_{\mathrm{TP}}|}{|\mathcal{X}|}\middle|\tau\right] p_{\mathrm{succ}} \tag{5}$$

$$= \mathbb{E}\left[\frac{|\mathcal{X}_{\mathrm{TP}}|}{L} e^{-|\hat{\mathcal{X}}|/L}\middle|\tau\right] \tag{6}$$

$$\approx \frac{\mathbb{E}[|\mathcal{X}_{\mathrm{TP}}| \mid \tau]}{L} e^{-\mathbb{E}[|\mathcal{X}||\tau]/L}, \tag{7}$$

where the approximation comes from assuming that $|\mathcal{X}_{\mathrm{TP}}|$ and $|\mathcal{X}|$ are independent and then performing a first-order Taylor approximation of $\mathbb{E}[e^{-|\mathcal{X}|/L} \mid \tau]$ around $\mathbb{E}[|\mathcal{X}| \mid \tau]$. The value of τ that maximizes this expression can be obtained numerically. Note that the optimal value is in general different from the one that maximizes p_{succ}.

B. Correlation-Aided Slot-Assigned ALOHA (SAA)

We now consider an alternative policy, termed slot-assigned ALOHA (SAA), which exploits the semantic correlation arising from having multiple views of each object, as captured by the cluster model. In this policy, the devices are pre-assigned to slots in such a way that the number of devices assigned to the same slot within the same cluster is minimized. This approach has previously shown to be efficient in random access scenarios with strong correlation [16]. The resulting allocation depends on the ratio between the number of slots, L, and the number of devices in each cluster, C. Specifically, if $L \leq C$, then each device is assigned to exactly one slot, and each slot is shared by an average of C/L devices. On the other hand, if $L > C$, then each device is assigned to an average of L/C slots, and selects one for transmission uniformly at random. To keep the assignment simple, we assign the slots in a round-robin manner until all devices have been assigned at least one slot and all slots have been assigned to at least one device. For instance, if $L = 7$ and $C = 10$, then 4 slots will be assigned to exactly one device within a given cluster, and 3 slots will be assigned to two devices within the cluster. The order of the round-robin sequence is randomized for each cluster to ensure a fair allocation.

In addition to the slots, the policy also requires a threshold τ. However, the expression for the number of received true positives depends on the relation between L and C, which makes it complicated to apply in practice. Instead, noticing that the average number of transmitting devices per slot is again $\lambda = (1/L)\,\mathbb{E}[|\mathcal{X}| \mid \tau]$ as in the previous policy, we can again obtain a reasonable threshold by maximizing Eq. (7). This threshold is approximately optimal unless $L \gg C$, since is does not take into account that at most C devices observe the query object. However, in our case we are primarily concerned about the case in which L and C are relatively close, since this regime has the highest efficiency.

IV. Machine Learning-Aided Semantic Matching

In practice, extracting the semantic content from the sensor data, e.g., in the case of image data, is often complex, making the design of the query vector and the computation of matching scores difficult. An attractive alternative solution is to instead *learn* the query and the matching score function using machine learning.

Since our ultimate goal is to perform classification of the objects that match the query object, the matching score should ideally capture both how similar an observation is to the query object, and how likely it is to contribute to classifying the object. To achieve this goal, we use a deep multi-view attention-based mechanism similar to the one in SEMDAS [12] and inspired by the approach in When2com [17]. Attention mechanisms work by fusing a set of feature vectors using a weighted sum, whose weights depend on the importance of each feature vector. In this paper, we will focus on the dot product attention, in which the weight of a feature vector $\mathbf{x}_m$ is computed as the inner product between the query vector $\mathbf{q}$ and a local observation-dependent key vector $\mathbf{k}_m \in \mathbb{R}^l$:

$$\mathbf{w}_m = \mathbf{q}^T \mathbf{k}_m. \tag{8}$$

Using the weights, a set of vectors can be fused into a single feature vector $\bar{\mathbf{x}}$ as

$$\bar{\mathbf{x}} = \sum_{m=1}^{M} \mathbf{x}_m \left(\frac{e^{\mathbf{w}_m}}{\sum_{n=1}^{M} e^{\mathbf{w}_n}} \right), \tag{9}$$

where the weights are normalized using the softmax function. The fused feature vector $\bar{\mathbf{x}}$ can then be used as input to a predictor, such as a classifier.

Usually, both the query $\mathbf{q}$ and the keys $\mathbf{k}_m$ are generated using neural networks, and learned as part of the predictor. In our case, since we are interested in classification, we do so by decomposing the classifier $\mathcal{F}(\mathcal{X})$ into three feed-forward neural networks, namely query and key encoders

$$\mathbf{q} = \mathcal{Q}(\mathbf{x}_q), \tag{10}$$

$$\mathbf{k}_m = \mathcal{K}(\mathbf{x}_m), \tag{11}$$

as well as a classifier $\tilde{\mathcal{F}}(\bar{\mathbf{x}})$ that outputs the predicted class based on the fused feature vector. Using these functions, $\mathcal{F}(\mathcal{X})$ can be written

$$\mathcal{F}(\mathcal{X}) = \tilde{\mathcal{F}} \left(\sum_{m=1}^{M} \mathbf{x}_m \left(\frac{e^{\mathcal{Q}(\mathbf{x}_q)^T \mathcal{K}(\mathbf{x}_m)}}{\sum_{n=1}^{M} e^{\mathcal{Q}(\mathbf{x}_q)^T \mathcal{K}(\mathbf{x}_m)}} \right) \right). \tag{12}$$

Due to the feed-forward structure of $\mathcal{F}(\mathcal{X})$, the three neural networks can be trained jointly using backpropagation in an end-to-end manner. Note that this classifier ignores the effects of the channel, which will only be considered during the inference phase.

The feature weights in Eq. (8) quantify how important a given feature vector is in predicting the class, and can be computed using only the query vector $\mathbf{q}$ and the observation itself $\mathbf{x}_m$ (through the key $\mathbf{k}_m$). Therefore, the weight is a natural choice as the matching score, i.e.,

$$\chi(\mathbf{x}_m, \mathbf{q}) = \frac{e^{\mathbf{q}^T \mathcal{K}(\mathbf{x}_m)}}{e^{\mathbf{q}^T \mathcal{K}(\mathbf{x}_m)} + 1} \tag{13}$$

$$\stackrel{\mathrm{d}}{=} \sigma\left(\mathbf{q}^T \mathcal{K}(\mathbf{x}_m)\right), \tag{14}$$

where $\mathbf{q} = \mathcal{Q}(\mathbf{x}_q)$ and $\sigma(x) = e^x/(e^x + 1)$ is the sigmoid function. Note that we use the sigmoid function for normalization instead of softmax as in Eq. (9), since softmax requires access to the weights of all devices in order to be computed. Note also that all devices use the same key encoder function. Using the computed matching scores, the devices can apply the transmission policies presented in the previous section. The set of received observations $\hat{\mathcal{X}}$ (after passing through the

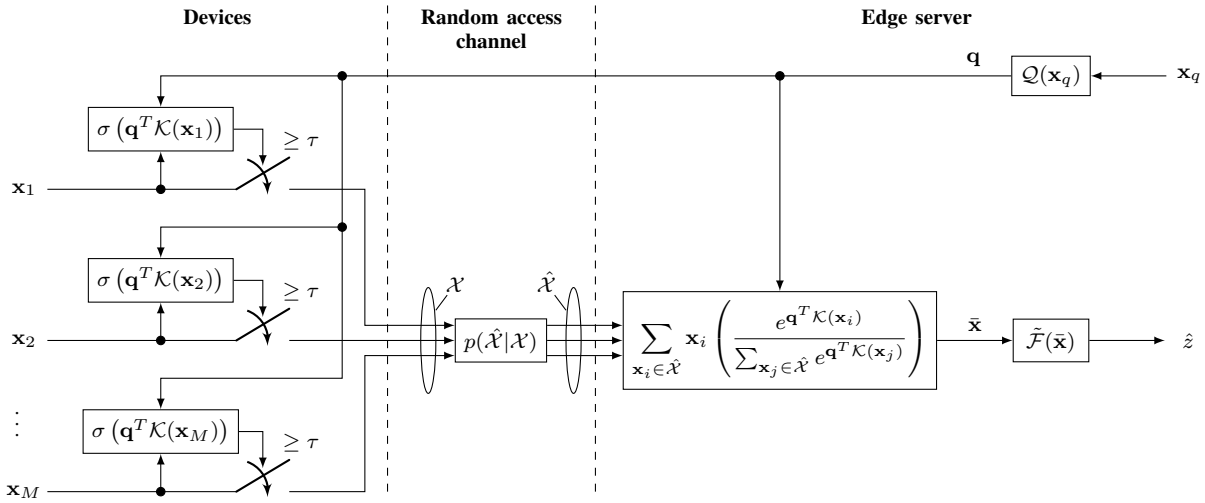

Fig. 2. Schematic illustration of the classifier, including the matching score computation and the wireless channel. Here, $\sigma(x) = e^x/(e^x+1)$ is the sigmoid function.

channel) will then be used as input to the model in Eq. (12), which re-computes the weights and normalizes them using softmax before fusing the features and passing them to the classifier $\tilde{\mathcal{F}}(\bar{\mathbf{x}})$, which then outputs the predicted class. An illustration of the proposed model in the inference phase is shown in Fig. 2.

Finally, we note that empirical estimates of the conditional matching score probability distributions required to optimize the threshold τ can be obtained during training/validation of the neural networks.

V. Numerical Results

A. Gaussian Matching Scores

To gain insight into the problem, we first study a Gaussian setting, in which the matching scores of positive examples ($z_m = z_q$) are independently distributed as $\chi_m = \sigma(X_{\text{pos}})$, where $X_{\text{pos}} \sim \mathcal{N}(\frac{\delta}{2}, 1)$, and the matching scores of negative examples ($z_m \neq z_q$) are independently distributed as $\chi_m = \sigma(X_{\text{neg}})$ with $X_{\text{neg}} \sim \mathcal{N}(-\frac{\delta}{2}, 1)$. Here, the parameter δ controls the classification margin: the larger the value of δ, the more polarized the matching scores become, making it possible to better discriminate between positive and negative examples. In the considered SEMDAS scenario, δ would depend on both the classification margin of the considered classification task and on the size of the query l. The true positive and false positive probabilities are then

$$\Pr(\text{TP}|\tau) = Q\left(\ln\left(\frac{\tau}{1-\tau}\right) - \frac{\delta}{2}\right), \tag{15}$$

$$\Pr(\text{FP}|\tau) = Q\left(\ln\left(\frac{\tau}{1-\tau}\right) + \frac{\delta}{2}\right), \tag{16}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2}\,\mathrm{d}u$. The probability distributions for the two matching score distributions are illustrated in Fig. 3 for various choices of δ.

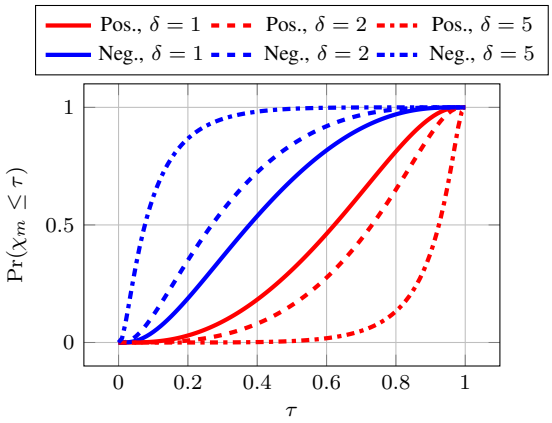

Fig. 3. Conditional distributions of the matching scores conditioned on being a positive or negative example in the Gaussian model.

We consider a scenario with $G = 10$ clusters, each containing $C = 10$ devices, so that the total number of devices is $M = 100$. The query object is assumed to be observed by one of the clusters, while the remaining clusters observe different objects. We start by studying the average precision and recall of the two random access policies described in Section III, defined as

$$\text{precision} = \mathbb{E}\left[\frac{|\hat{\mathcal{X}}_{\text{TP}}|}{|\hat{\mathcal{X}}_{\text{TP}}| + |\hat{\mathcal{X}}_{\text{FP}}|}\right], \tag{17}$$

$$\text{recall} = \mathbb{E}\left[\frac{|\hat{\mathcal{X}}_{\text{TP}}|}{P}\right], \tag{18}$$

where $|\hat{\mathcal{X}}_{\text{TP}}|$ and $|\hat{\mathcal{X}}_{\text{FP}}|$ refer to the *received* number of true/false positives, i.e., the ones that remain after the random access channel, and $P = C$ is the total number of query object observations (positives) in the scenario. Thus, precision quantifies the fraction of the received observations

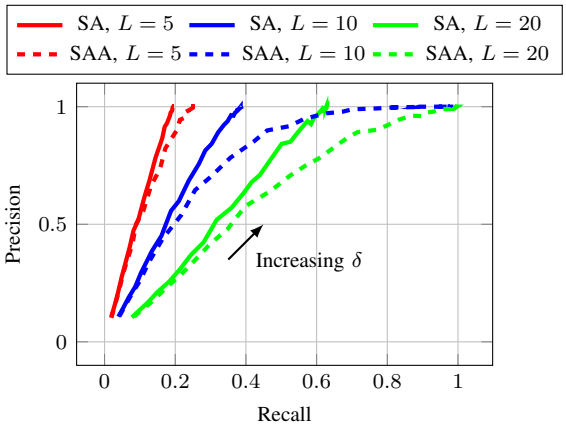

Fig. 4. Precision and recall for the studied random access policies under the Gaussian model when the classification margin parameter δ is varied.

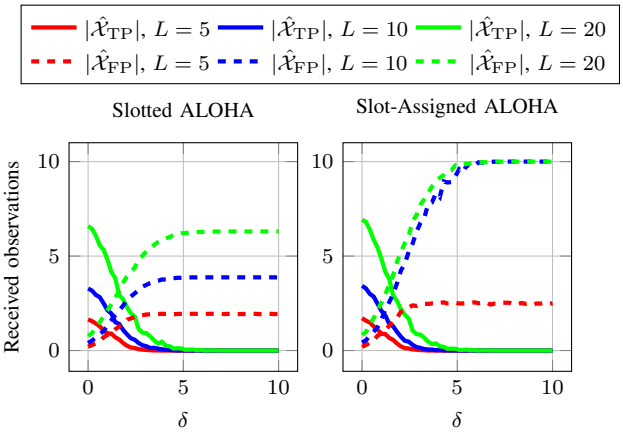

Fig. 5. Number of received true positive ($|\hat{\mathcal{X}}_{\mathrm{TP}}|$) and false positive ($|\hat{\mathcal{X}}_{\mathrm{FP}}|$) observations vs. δ for the two random access policies under the Gaussian model.

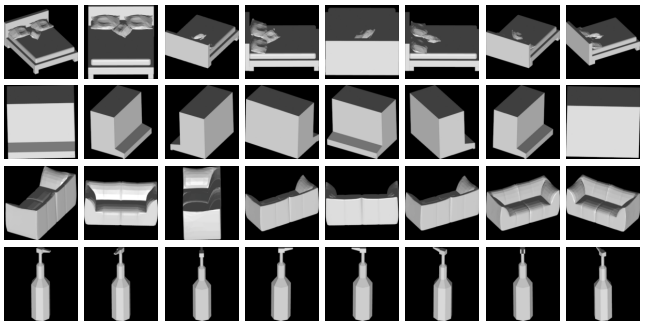

Fig. 6. Samples from the ModelNet40 dataset. Each row shows a specific object and each column is a view, corresponding to an observation of the object.

that are relevant, and recall quantifies the fraction of relevant observations that are received. The results are shown in Fig. 4 for $L \in \{5, 10, 20\}$ transmission slots. In general, increasing the classification margin δ increases both precision and recall, and the same is true when the number of transmission slots is increased. The precision can reach a value of 1 with both policies, but due to collisions the SA policy attains a maximum recall that is bounded from 1. On the other hand, the SAA policy can approach a recall of 1 as long as the number of slots is at least equal to the number of devices within a cluster. This demonstrates the main benefit of SAA, namely its ability to exploit knowledge about the observation and activation correlation to avoid collisions.

The number of received true positive and false positive observations is shown in Fig. 5. As can be seen, the fact that the number of potential false positives is much larger than the number of potential true positives has a significant influence when d is small (only $C = 10$ out of $M = 100$ devices observe the query object). In particular, false positive observations dominate the received observations in both random access policies. If the observations were to be used for inference, this is likely to lead to significant performance degradation, suggesting that achieving a sufficiently large classification margin is crucial to the performance of the system.

B. Edge Classification with Machine Learning-Aided Matching

We now turn our attention to machine learning-aided semantic matching. We consider the case in which observations are drawn from the multi-view ModelNet40 dataset [18], and we will assume that the edge server performs classification of the $|\mathcal{Z}| = 40$ categories in the dataset. A sample of the dataset is shown in Fig. 6, where each row contains observations of a given object, assumed to be observed from different angles. The observation feature vectors $\mathbf{x}_m$ are generated using the feature extraction layers of the VGG11 model [19], which results in $d = 25088$ features. $\mathcal{Q}(\mathbf{x}_q)$ and $\mathcal{K}(\mathbf{x}_m)$ are both neural networks of a single hidden layer with ReLU activations, and the classifier $\mathcal{F}(\bar{\mathbf{x}})$ is constructed using the classifier layers of the VGG11 model. Both the VGG11 feature extraction layers and the classifier layers are pre-trained on the ImageNet dataset, and then fine-tuned along with the query and key encoder networks on the ModelNet40 dataset with the cross-entropy loss function. The fine-tuning is performed in two stages. First, the classifier network $\tilde{\mathcal{F}}(\mathbf{x})$ is trained in isolation using single ModelNet40 observations. Then, in the second stage, the classifier network is plugged into the full neural network $\mathcal{F}(\mathcal{X})$, comprising in addition to the classifier also the query and key encoder neural networks. The full neural network is then trained in an end-to-end fashion on samples $\mathcal{X}$ containing 2 positive observations and 10 negative observations, but without considering the random access channel. Finally, after training the network, empirical distributions of the matching scores for positive and negative examples are obtained using the training data. We then evaluate its performance in a scenario with $G = 10$ clusters of $C = 10$ devices as before, and include the random access policies and

TABLE I
PERFORMANCE OF THE MACHINE LEARNING-AIDED SEMANTIC MATCHING RANDOM ACCESS POLICY

		$l = 16$				$l = 32$				$l = 64$				$l = 128$			
		$\lvert\hat{\mathcal{X}}_{\mathrm{TP}}\rvert$	$\lvert\hat{\mathcal{X}}_{\mathrm{FP}}\rvert$	Prec.	Acc.	$\lvert\hat{\mathcal{X}}_{\mathrm{TP}}\rvert$	$\lvert\hat{\mathcal{X}}_{\mathrm{FP}}\rvert$	Prec.	Acc.	$\lvert\hat{\mathcal{X}}_{\mathrm{TP}}\rvert$	$\lvert\hat{\mathcal{X}}_{\mathrm{FP}}\rvert$	Prec.	Acc.	$\lvert\hat{\mathcal{X}}_{\mathrm{TP}}\rvert$	$\lvert\hat{\mathcal{X}}_{\mathrm{FP}}\rvert$	Prec.	Acc.
$L = 5$	SA	0.482	0.130	0.276	0.272	0.826	0.158	0.472	0.447	0.902	0.213	0.502	0.475	1.043	0.177	0.568	0.534
	SAA	0.455	0.110	0.229	0.220	0.779	0.132	0.395	0.372	0.918	0.193	0.458	0.429	1.107	0.170	0.523	0.478
$L = 10$	SA	1.309	0.581	0.416	0.422	1.941	0.660	0.630	0.619	2.044	0.778	0.650	0.631	2.286	0.642	0.712	0.690
	SAA	2.288	0.460	0.438	0.425	3.356	0.582	0.659	0.624	3.522	0.752	0.683	0.649	3.971	0.618	0.736	0.695
$L = 20$	SA	2.616	2.130	0.489	0.515	3.347	2.119	0.637	0.647	3.482	2.143	0.647	0.651	3.746	1.951	0.688	0.680
	SAA	3.499	2.327	0.513	0.537	4.422	2.245	0.664	0.673	4.746	2.705	0.641	0.652	5.059	2.212	0.706	0.707

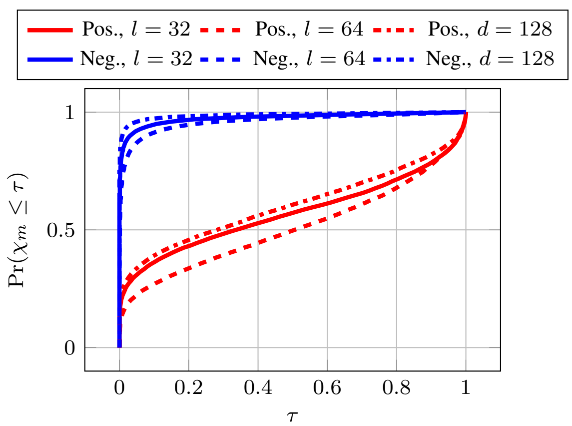

Fig. 7. Conditional distributions of the matching scores conditioned on being a positive or negative example in the machine learning-aided matching scenario.

the channel as well.

The matching score distributions are shown in Fig. 7 for query dimensions $l \in \{32, 64, 128\}$. In general, the figure shows that the learned model is able to discriminate between positive and negative observations. However, there is a significant number of positive observations that produce a low matching score. This can either be because the query and matching neural networks fail to match the observations, or because some observations are of bad quality and unlikely to contribute significantly to the classification task. In general, although there is some variation between the distributions for different query dimensions l, there is no clear tendency.

The average number of true positives, false positives, the precision, and classification accuracy of various configurations are provided in Table I. Here it can be seen that despite no tendency in the matching score distributions, increasing l consistently leads to an increase in the number of received true positives, which suggests that the query gets better at matching the relevant observations. This is clearly reflected in the classification accuracy, which also increases with l. As expected, increasing the number of slots L also leads to improved accuracy. However, since it also results in a lower τ and thereby increases the number of received false positives, it does not always result in higher precision. The fact that the precision is reduced or unchanged but the accuracy increases suggests that the attention mechanism at the edge server is able to efficiently filter out the false positives. Nevertheless, filtering out false positives at the individual devices is crucial to avoid congestion in the channel, which would prevent the true positives from being delivered to the edge server.

The results in Table I also show that the SAA protocol generally outperforms the SA protocol, except in the case with $L = 5$. This is a direct consequence of having fewer collisions in the channel as long as $L \geq C$, which can be seen by comparing the total number of received observations between the two random access policies. On the other hand, when $L < C$ the randomness in the SA policy can lead to higher performance. For instance, if all devices within a cluster activate in the case with $L = 5$, then all packets will collide with SAA and no observations will be delivered. On the other hand, although there will also be many collisions in the SA policy, there is still a small probability that some observations will be delivered. The results suggest that more work is required in order to design efficient policies for this regime.

VI. CONCLUSION

In this paper, we have considered the problem of collecting relevant semantic data from correlated IoT devices. Compared to traditional IoT communication, where the devices transmit their observations periodically or sporadically, in the considered framework the devices transmit reactively, and only if their observations are relevant for the destination. We have proposed a two-step semantic data sourcing random access protocol, in which an edge server first broadcasts a semantic query. Then, each device computes a matching score characterizing how relevant its current observation is for the edge server, and transmits its observation over a random access channel only if the matching score exceeds a threshold. We have considered two random access policies, namely a classical slotted ALOHA (SA) policy and a correlation-aided slot-assigned ALOHA (SAA) policy that exploits spatial correlation in the observations between devices. Furthermore, we have demonstrated how the policies can be integrated with a machine learning-based query design. The performance of the proposed protocol has been studied both in a toy example, using a Gaussian observation model, and using the machine learning-based query design. The results show that

the protocol effectively collects relevant observations under both the SA and the SAA policies while filtering out irrelevant observations. While the SAA policy generally performs better when the number of transmission slots is comparable to the number of relevant observations, the SA policy is best when only a small number of slots are available. In conclusion, the proposed protocol presents a promising direction for collecting data in real-time from massive IoT networks based on the semantic content of sensor observations.

ACKNOWLEDGMENT

This work was supported by the Independent Research Fund Denmark under Grant 1056-00006B, and by the Villum Investigator grant "WATER" from the Velux Foundation, Denmark.

REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[2] P. Popovski, F. Chiariotti, K. Huang, A. E. Kalør, M. Kountouris, N. Pappas, and B. Soret, "A perspective on time toward wireless 6G," *Proceedings of the IEEE*, vol. 110, no. 8, pp. 1116–1146, 2022.

[3] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2731–2735.

[4] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.

[5] A. Maatouk, S. Kriouile, M. Assaad, and A. Ephremides, "The age of incorrect information: A new performance metric for status updates," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2215–2228, 2020.

[6] X. Zheng, S. Zhou, and Z. Niu, "Urgency of information for context-aware timely status updates in remote control systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7237–7250, 2020.

[7] P. Agheli, N. Pappas, and M. Kountouris, "Semantics-aware source coding in status update systems," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2022, pp. 169–174.

[8] F. Chiariotti, J. Holm, A. E. Kalør, B. Soret, S. K. Jensen, T. B. Pedersen, and P. Popovski, "Query age of information: Freshness in pull-based communication," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1606–1622, 2022.

[9] S. Kapadia and B. Krishnamachari, "Comparative analysis of push-pull query strategies for wireless sensor networks," in *Distributed Computing in Sensor Systems*, Springer. Springer Berlin Heidelberg, 2006, pp. 185–201.

[10] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Rec.*, vol. 31, no. 3, pp. 9–18, sep 2002.

[11] J. Shiraishi, H. Yomo, K. Huang, Č. Stefanović, and P. Popovski, "Content-based wake-up for top-k query in wireless sensor networks," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 1, pp. 362–377, 2020.

[12] K. Huang, Q. Lan, Z. Liu, and L. Yang, "Semantic data sourcing for 6g edge intelligence," *arXiv preprint arXiv:2301.00403*, 2023.

[13] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is semantic communication? a view on conveying meaning in the era of machine intelligence," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, 2021.

[14] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2021.

[15] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[16] A. E. Kalør, O. A. Hanna, and P. Popovski, "Random access schemes in wireless systems with correlated user activity," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[17] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.