Statistical Learning-based Grant-Free Access for Delay-Sensitive Internet of Things Applications

Muhammad Ahmad Raza, Mehran Abolhasan, *Senior Member, IEEE,* Justin Lipman, *Senior Member, IEEE,* Negin Shariati, *Member, IEEE,* Wei Ni, *Senior Member, IEEE,* and Abbas Jamalipour, *Fellow, IEEE*

***Abstract*—Mission-critical Internet-of-Things (IoT) applications require communication interfaces that provide ultra-reliability and low latency. Acquiring knowledge regarding the number of active devices and their latency-reliability requirements becomes essential to optimize resource allocation in heterogeneous networks. Due to the inherent heavy computation overheads, the conventional centralized decision-making approaches result in large latency. The distributed computing and device-level prediction of network parameters can play a significant role in designing mission-critical IoT applications operating in dynamic environments. This paper considers the medium access control (MAC) layer of heterogeneous networks employing a framed-ALOHA-based restricted transmission strategy to enhance reliability. We present a statistical learning-based device-level network exploration mechanism in which end-devices use their transmission history to predict different network parameters. The IoT devices share the learned parameters with the base station (BS) to identify different groups presented in the network. The simulation results show that the mean square error (MSE) in predicting different network parameters can be reduced by increasing the history window size. In this regard, the optimal size of the history window under the given accuracy constraints is also determined. We demonstrate that the proposed device-level network load prediction mechanism is more robust as compared to the BS-centered approach.**

***Index Terms*—5G wireless networks; Mission-critical Internet of Things; Industry 4.0; Statistical learning; Device-level learning**

I. Introduction

MISSION critical Internet-of-Things (IoT) applications require ultra-reliable and low latency communication (URLLC) interfaces to transmit delay-sensitive data. These applications form an essential dimension of IoT 2.0 systems, which includes intelligent transportation systems, unmanned aerial vehicles (UAVs), public safety communication networks, telesurgery, smart grids, and Industry 4.0, covering smart factories [1]. Different mission-critical applications can have different latency and reliability specifications; some of those are highlighted [2]. From the vehicular communication perspective, different vehicle-to-everything (V2X) communication use-cases in which a vehicle communicates with other vehicles (V2V), with wayside infrastructure (V2I) and with mobile users (V2P), can involve delay-sensitive data transmission, which requires ultra-reliability [3], e.g., self-driving vehicles. Similarly, intelligent transportation systems aided by the vehicular ad-hoc networks (VANETs) aim to exchange safety-critical messages under strict latency and reliability requirements. It becomes very challenging to fulfill the desired latency-reliability requirements for the V2X based systems in heterogeneous networks.

Latency experienced by data packets in a wireless communication system is composed of deterministic and random components. The information processing delays at the transmitter and receiver determine the deterministic component, while the delays involved in retransmissions and back-off phases define the random part of the latency [4]. The reliability of a communication system can be affected by many factors, including the time-varying nature of the wireless channel, different sources of interference causing random changes in the signal to noise ratio (SNR) at the receiver, type of a particular constellation being used, error detection and correction codes, and nature of the medium access control (MAC) mechanism [4]. In the context of mission-critical IoT applications, reliability is interpreted as the probability of meeting the prescribed latency bound [5]. The real-time processing of a massive amount of data generated by a large number of sensors in these networks requires that the data be transferred from the source to the data centers within the application-specific latency while ensuring desired levels of reliability. It becomes challenging to meet these requirements in heterogeneous networks where different groups of IoT devices can have different latency-reliability criteria and network parameters change dynamically.

For the optimal utilization of the available radio resources in heterogeneous networks to meet the application-specific latency-reliability criterion, it is essential to know the number of active devices and their latency-reliability requirements. The conventional centralized decision-making approaches in which these tasks are performed at the base station (BS) suffer from heavy computation overheads, which result in higher latency. Therefore, the use of distributed computing and device-level learning of network parameters can play a significant role in designing mission-critical IoT applications by reducing the computation burden at the BS. Fog computing is a distributed computing paradigm that aims to address the bandwidth, latency, and reliability constrained applications in heterogeneous networks by providing cloud-like functionalities near the data source [6-8]. The growing number of vehicles equipped with intelligent IoT devices makes it necessary to design such vehicular networks in which these vehicles can learn and adapt

M.A. Raza, M. Abolhasan, J. Lipman, and N. Shariati are with the Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia (email: Muhammadahmad.Raza@student.uts.edu.au; Mehran.Abolhasan@uts.edu.au; Justin.Lipman@uts.edu.au; Negin.Shariati@uts.edu.au). W. Ni is with the Commonwealth Scientific and Industrial Research Organization, Australia (email: Wei.Ni@data61.csiro.au). A. Jamalipour is with the Faculty of Engineering, University of Sydney, NSW 2006, Australia (email: a.jamalipour@ieee.org).

to the network dynamics by themselves without depending upon the additional infrastructure. Hou et al. [9] proposed a vehicular fog computing (VFC) design paradigm that uses moving and parked vehicles as infrastructure for computation and communication in vehicular networks. The VFC paradigm can better utilize the available resources, enhance the overall system performance, and support the latency-sensitive applications in vehicular networks.

The choice of a particular network access mechanism plays a major role in meeting the application specific QoS requirements. The grant-based MAC protocol in Long Term Evolution (LTE) allows the IoT devices to transmit their data over dedicated resources if they are successful in a contention-based random access channel (RACH) phase. The RACH phase introduces additional signaling overheads, and the grant-based protocols are suitable for a smaller number of IoT devices. In comparison, the data transmission in grant-free network access mechanisms is performed over shared radio resources in a random-access manner without requesting a resource grant. The grant-free network access approach has many benefits over the grant-based strategies to support the uplink connectivity for massive IoT, generating sporadic traffic [10], [11]. However, while achieving the massive connectivity target, the latency and reliability can be compromised in the grant-free MAC protocols. Several retransmission schemes have been proposed to enhance the reliability in mission-critical IoT applications [12-15]. Another important constraint is the energy consumption in critical-IoT applications. Since the IoT devices can have limited power storage capacity, the design of energy-efficient grant-free MAC protocols is indispensable [16].

While communicating over shared radio resources, the availability of knowledge regarding the number of active devices plays a crucial role in optimizing radio resource allocation and to control the congestion efficiently [17], [18]. However, in the absence of any feedback, the BS lacks the knowledge of the exact cardinality of collisions, i.e., the number of users colliding per channel. In multichannel slotted ALOHA (framed-ALOHA) based systems, the BS can estimate the number of active devices in a frame by using the number of idle channels [17], [19]. However, for heterogeneous networks with dynamically varying parameters, tracking the number of active devices at the BS gets complicated and less accurate under a higher network load.

On the other hand, in order to address the stringent requirements of URLLC for mission-critical IoT applications in heterogeneous networks where network parameters change dynamically, acquiring knowledge of probability distributions associated with these parameters is equally essential [20]. In this regard, statistical learning is a promising tool to learn the network parameters probabilistically in a dynamic environment. Therefore, a statistical learning framework has been proposed in [20] for the physical layer design of URLLC systems. In this framework, the authors considered the limited channel knowledge and model mismatch to design a transmitter that can statistically learn and adapt the transmission rate, such that the desired reliability constraint is met probabilistically. This framework uses two necessary statistical measures for URLLC systems named the average reliability (AR) and the probably correct reliability (PCR). The AR criterion is helpful in a dynamic environment, while the PCR approach is more appropriate for relatively static environments.

The above discussion highlights the fact that supporting mission-critical applications in dynamic heterogeneous networks with a large number of IoT devices is very challenging. This fact motivates us to use the statistical learning paradigm to design such mechanisms where IoT devices can assist the BS in predicting different network parameters and the associated probability distributions. This paper considers the MAC layer of the uplink communication interface in heterogeneous networks. A large number of IoT devices communicate with one BS over shared radio resources in a grant-free manner. This uplink communication follows a framed-ALOHA-based restricted transmission strategy. Following are the key contributions and novelty of this paper:

- We propose a statistical learning-based device-level network exploration mechanism at the MAC layer for delay-sensitive IoT applications. The end-devices are enabled to learn network parameters under a dynamic environment.
- The proposed mechanism uses the information available at the devices in the history of their previous transmissions and enables the devices to predict different network parameters. Consequently, the end-devices can predict the number of active devices, the probability of collision in each frame, the average number of successful devices per round, and the average behavior of random latency.
- For the optimal radio resource allocation, the statistical knowledge of dynamic network parameters learned by the end-devices is shared with the BS to identify different IoT groups present in the network. Consequently, the computation burden at the BS is reduced, which can reduce the overall latency offered by the network.
- Using the mean square error (MSE) criterion, the optimal size of the transmission history window is determined under the given accuracy constraints in predicting different network parameters.
- The probability of exception is used to measure the robustness of the proposed statistical learning-based device-level network load prediction mechanism. Results show that the proposed mechanism is more robust than the BS-centered approach of [17] under the higher network load.

The rest of the paper is organized as follows: a summary of the related literature and the critical analysis is presented in Section II. The system model is explained in Section III, and the network exploration process is presented in Section IV. The performance analysis, simulation results, and the optimal size of the history window are given in Section V. A comparison of the proposed mechanism with the BS-centered approach is also presented in Section V. The paper is concluded in Section VI while highlighting some future research directions. Different notations used in this paper are defined in Table. II. This work is the first of its kind to the best of our knowledge, which uses statistical learning at device-level for network exploration at the MAC layer.

II. Related Works

In this section, we review the related works from the perspectives of retransmission mechanisms and the estimation of the number of active devices, which are also interrelated. In order to provide ultra-reliability in mission-critical IoT applications, several retransmission mechanisms have been proposed. Abreu et al. [21] proposed a scheme in which URLLC users with similar traffic characteristics are grouped by the BS following the block error rate (BLER) criterion. Each group uses a pre-scheduled shared resource for single retransmission if the initial transmission fails. The efficiency of this scheme is primarily dependent upon the right grouping of users at the BS. Abreu et al. [12] proposed a scheme in which active devices perform T attempts such that the first transmission is performed on dedicated channels, while $(T-1)$ retransmissions are performed on shared channels, and the receiver uses successive interference cancellation (SIC) for decoding messages over shared resources. Galinina et al. [14] presented a scheme in which an active device with a certain transmission probability sends single or multiple replicas over the shared channels in one slot. In this work, an optimal control algorithm is presented that guarantees the minimal channel access delay by controlling the probability of transmission and the number of replicas. For both cases of single and multiple transmissions, corresponding practical implementations are also discussed. The probability of transmission and the number of replicas is decided at the BS level.

While considering PHY-layer abstraction, the optimum number of retransmissions depends upon the number of transmitting devices (network load), available resources, and nature of the MAC protocol being used. Most of the existing techniques either assume perfect knowledge of the network load or rely on the BS to estimate the number of transmitting devices to update the resource allocation strategy.

Astudillo et al. [22] proposed a mechanism for LTE based cellular IoT networks which enables the end-devices to estimate the number of active devices in a frame by using the information regarding the number of detected preambles at the BS in that frame. The end-devices can determine the number of detected preambles by counting the number of random access response (RAR) messages sent by the BS. This approach performs well as long as the BS is capable of transmitting the RAR messages against all the detected preambles. In order to perform the estimation under incomplete information at the device-level, which happens when the number of transmitting devices is higher, Astudillo et al. [22] proposed to use the value of Access Class Barring (ACB) probability. The ACB probability is a function of the number of active devices and number of channels, and it is broadcasted by the BS regularly. On the other hand, the computation of ACB-probability at the BS involves estimation of the number of active devices.

Oh et al. [17] proposed a mechanism to estimate the number of active devices at the BS in a given frame by computing the probability that a preamble remains idle in that frame. The proposed mechanism works as long as the probability of having an idle preamble is non-zero. However, for the higher number of active devices, the number of idle preambles in a frame can become zero with high probability. Moreover, due to the channel impairments, an RB originally selected by one or more devices can be erroneously detected as an idle one [23]. Thus, the accuracy of this estimation method deteriorates in a dynamic environment.

For the framed-ALOHA networks, Jiang et al. [23], proposed an online supervised learning method that enables the BS to predict current traffic load. In this work, the BS keeps the history of idle, successfully decoded, and collided resource blocks in previous and current frames to predict the current traffic load. This work also incorporated the case where a detection error can occur, and a resource block can be detected as an idle one with a non-zero probability. The proposed

TABLE I
List of notations used.

Notation	Definition
$A_{m,n}$	Outcome of a transmission
$\alpha_{m,n}$	Probability of collision in a frame
$\widehat{\alpha}_n$	Prediction of $\alpha_{m,n}, \forall m$
$\epsilon_r^{(i)}$	Reliability constraint of the i^{th}-group
$\mathcal{G}$	No. of groups
$\mathbf{H}$	Transmissions history matrix for network exploration
K	No. of orthogonal channels in each frame
$K_{m,n}$	No. of idle channels in the n^{th} frame of the m^{th} round
$L_{max}^{(i)}$	Group specific maximum affordable latency
L	Random component of the latency
$M^{(i)}$	No. of active IoT devices in the i^{th}-group
M	Total no. of active IoT devices
$M_{m,n}$	No. of active devices in a frame
$\widehat{M}_n$	Prediction of $M_{m,n}, \forall m$
MSE_S	MSE in prediction of S_{av}
MSE_P	MSE in prediction of P_s
MSE_μ	MSE in prediction of μ_L
N	No. of frames in one round
$\Pr(.)$	Probability measure
P_{s_m}	Probability of success in the m^{th} round
P_s	Average probability of success per round
$\widehat{P}_s$	Prediction of P_s
$R^{(i)}$	Reliability constrained no. of rounds for the i^{th}-group
$R_{max}^{(i)}$	Rounds according to affordable latency for the i^{th}-group
$\widehat{R}$	Optimal no. of rounds for network exploration
R	No. of rounds
S_{av}	Average no. of successful devices in one round
$\widehat{S}_{av}$	Prediction of S_{av}
μ_L	Average (re)transmissions for a successful transmission
$\widehat{\mu}_L$	Prediction of μ_L
ζ_S	MSE constraint to predict S_{av}
ζ_P	MSE constraint to predict P_s
ζ_μ	MSE constraint to predict μ_L

mechanism outperformed the existing method of moments (MoM) and maximum likelihood (ML) prediction techniques.

The works mentioned above perform a centralized decision-making approach in which end-devices follow a BS-driven strategy. The performance of these schemes relies on the availability of accurate knowledge of network load at the BS. These schemes also involve significant control signaling, which can cause additional latency. In addition to that, these retransmission methods do not address the network heterogeneity because of the different latency-reliability requirements, and the end-devices are unable to adapt to the network dynamics by themselves. Therefore, the IoT devices in heterogeneous networks should have the capability to adapt to network dynamics without relying much on the BS, which can be accomplished by equipping the IoT devices with reasonable computation resources, allowing them to learn the network parameters and share the knowledge with the BS [24]. Shafiq et al. [25] proposed a random access protocol for V2I communications. The protocol enabled vehicular entities to optimize their transmission probability for the given network density to maximize network throughput. The optimal transmission probability was computed using exhaustive search approach. Ye et al. [26] reviewed the potential use of different data-driven techniques on several aspects of vehicular networks. Due to the highly dynamic nature of vehicular networks, intelligent resource management is critical in these networks. This feature of vehicular networks leads to the demand for resource optimization methods capable of adapting to dynamic changes [27].

In the context of device-level learning at the MAC layer, Raza et al. [28] proposed a statistical learning-based dynamic retransmission mechanism for the homogeneous mission-critical-IoT applications. This mechanism enables the IoT devices to predict retransmissions limit under the given latency-reliability constraint by using the history of their previous transmissions. The work in [28], assumes that all the IoT devices have identical latency-reliability requirements, the number of active IoT devices remain fixed in each frame of the observation interval, and every active device has a packet to transmit in each frame. Under these assumptions, the collision probability remains fixed in each frame. In this paper, we present a statistical learning-based mechanism that enables the end-devices to predict different MAC layer parameters while covering more general heterogeneous mission-critical-IoT applications in a dynamic environment. We present the analytical modeling of the restricted transmission strategy that involves variable collision probability, also discussed in [28]; and device-level prediction of the number of active devices, the average number of successful devices per round, average latency, and optimal size of the history window. The robustness of the proposed device-level statistical learning-based network load prediction method is measured through the probability of exception, and it is compared with the BS-centered approach. The device-level network exploration helps the end-devices adapt to the network dynamics while meeting the desired levels of latency-reliability probabilistically. The proposed mechanism can also assist the BS in optimizing radio resource utilization and enhancing the performance of the existing radio resource management methods by using the knowledge available at the end-devices.

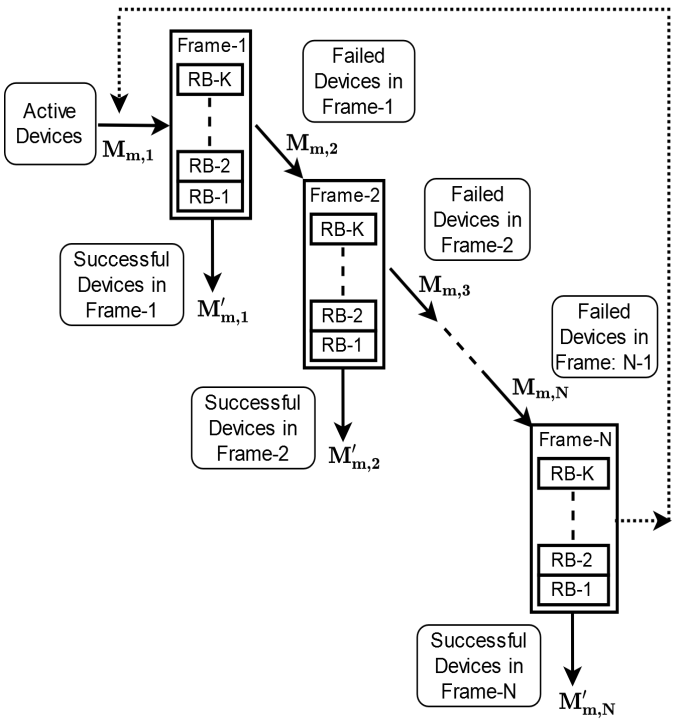

Fig. 1. Framed-ALOHA-based restricted transmission strategy over N frames in the m^{th} round of an observation interval of R rounds.

III. System Model

We consider a heterogeneous network composed of $\mathcal{J}$ IoT devices virtually partitioned into $\mathcal{G}$ groups such that each group in the network contains IoT devices with identical latency-reliability requirements. Each group of IoT devices can be part of a particular mission-critical application generating short data packets. The parameter $\mathcal{J}$ is expressed as $\mathcal{J} = \sum_{i=1}^{\mathcal{G}} \jmath^{(i)}$ where $\jmath^{(i)}$ is the number of IoT devices in the i^{th}-group, $\forall i = 1, 2, ..., \mathcal{G}$. The total number of active devices in the network is $M = \sum_{i=1}^{\mathcal{G}} M^{(i)}$, where $M^{(i)} \leq \jmath^{(i)}$ is the number of active devices in the i^{th}-group, $\forall i$. As shown in Fig. 1, the active devices from different groups communicate over K orthogonal shared resource blocks (RBs) for the transmission of their messages to a single BS in a grant-free manner by employing a framed-ALOHA based restricted transmission policy. In this protocol, each time slot is composed of multiple resource blocks (RB), called a frame, and an RB can be a time, frequency, or code-based resource. In each frame, an active device selects one of the RBs randomly such that the selection is uniform across all the RBs and independent from other active devices. If two or more IoT devices select the same RB, the transmission fails, and the colliding devices attempt again in the next frame. We consider physical layer abstraction to the MAC layer in which transmission fails only because of the collisions. We use the terms RB and channel interchangeably. All active devices begin to transmit at the start of a round only, which is composed of N-frames, and an observation interval of R independent rounds is

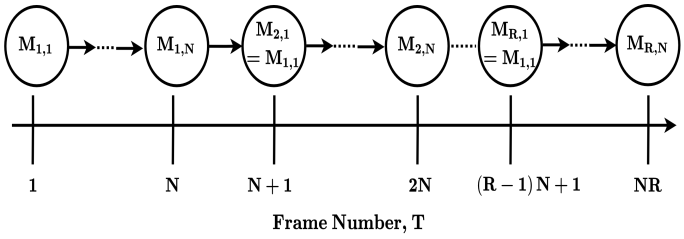

Fig. 2. Network state: number of active devices in each frame of R rounds.

considered. Upon successful transmission, the devices receive an acknowledgment from the BS, and they stop transmitting in the current round. The restricted transmission strategy helps improve the latency-reliability performance by reducing the collision probability in successive frames of any round. It is assumed that the size of the data packet is the same across all the groups and can be completely transmitted within one frame duration.

The value of parameter M can change from one observation interval to another. The end-devices do not need to know the probability distribution associated with M. The IoT devices capture the status of parameter M regularly in each observation interval and share it with the BS, as explained in the next section. As shown in Fig. 2, the time index T representing the frame number can be expressed as a function of the round number (m), and the frame number (n) as: $T = (m-1)N + n$, where $m = 1, 2, ..., R$ and $n = 1, 2, ..., N$. Fig. 2 shows the network state in terms of the number of transmitting devices in each frame of the history window. While, the number of transmitting devices in each frame of the m^{th} round is defined as:

$$M_{m,n} = \begin{cases} M, & n = 1; \\ M - \sum_{j=1}^{n-1} M'_{m,j}, & n = 2, 3, ..., N. \end{cases} \quad (1)$$

where $M^{'}_{m,n}$ denotes the number of successful devices in the n^{th} frame of the m^{th} round. Due to the restricted transmission strategy in each round, we have $M_{m,1} \geq M_{m,2} \geq, ..., \geq M_{m,N} \geq 0$, $\forall m$, and $M_{m,1} = M$, $\forall m$. The number of transmitting devices in a frame depends upon the number of transmitting and successful devices in the previous frame. Thus, two situations can arise: for the first case in which $M_{m,n-1} > K$, we have $(M_{m,n-1} - K) < M_{m,n} \leq M_{m,n-1}$, and in the second case when $M_{m,n-1} \leq K$, we get $0 \leq M_{m,n} \leq M_{m,n-1}$.

The latency-reliability requirements of the V2I communication interface can be different from other IoT devices present in the network. The number of active devices also changes when vehicular IoT entities leave or exit the coverage area of a serving BS. Therefore, the proposed framed-ALOHA-based grant-free network access is suitable for the V2I communication scenario in which vehicular IoT entities communicate with a common BS over shared radio resources under a dynamic network load. Moreover, the proposed grant-free access with a restricted transmission strategy helps end-devices reduce their energy consumption. This is because after having a successful transmission, the corresponding devices stop transmitting in the current round.

Due to the time-varying nature of the number of transmitting devices, it becomes challenging to assess the feasibility of running a particular mission-critical application in heterogeneous networks, and acquiring the statistical knowledge of the network dynamics becomes essential. The following section demonstrates how the end-devices can explore the network to learn different network parameters at the MAC layer.

IV. Device-Level Network Exploration

In this section, we present a statistical learning-based procedure to explore the network at the device-level. The number of successful devices in each round $S_m = \sum_{n=1}^{N} M^{'}_{m,n}$, is a random quantity, as a statistical measure, we are interested in getting the knowledge of average successful devices per round at the device-level which is defined as:

$$S_{av} := \frac{1}{R} \sum_{m=1}^{R} \sum_{n=1}^{N} M^{'}_{m,n}. \quad (2)$$

A related parameter is the probability of success per round (P_{s_m}) derived in the Subsection IV-A, and it depends on the number of transmitting devices in each frame of the given round. There can be possibly different patterns of the number of active devices in each round; the quantity P_{s_m} can vary from round to round. Thus, acquiring the average behaviour of P_{s_m} denoted by P_s is essential for network exploration.

The third important parameter is the latency offered by the network. In this paper, we focus on the random component of the latency (L) experienced by a data packet due to the (re)transmissions and its average behavior denoted by μ_L. For mission-critical-IoT applications, a statistical reliability constraint can be defined as: $\Pr\left(L \leq L^{(i)}_{max}\right) \geq 1 - \epsilon^{(i)}_r$, where $\epsilon^{(i)}_r$ and $L^{(i)}_{max}$ are the group-specific reliability criterion and maximum affordable latency, respectively. Under the given values of K and N, each group requires a minimum number of rounds $R^{(i)}, \forall i$ within which it can meet the desired reliability constraint probabilistically. This fourth parameter $R^{(i)}, \forall i$ is also learned dynamically at the device-level and, the end-devices share their knowledge with the BS, which can identify the number of groups present in the network. In summary, we aim to enable the end-devices to predict four important network parameters $\left(S_{av}, P_s, \mu_L, R^{(i)}\right)$ so that the BS can utilize their knowledge for the better utilization of the available resources.

A. Device-level prediction of P_s and S_{av}

The only information provided to the IoT devices is the number of channels (K) in each frame and the size (N) of a round. The number of transmitting devices $(M_{m,n})$ in each frame of a round is not known by the BS and the IoT devices a-prior. In order to explore the network, each device keeps the record of transmission outcomes from last R rounds. When a device of interest performs a transmission in the n^{th} frame of the m^{th} round, a Bernoulli random variable $A_{m,n}$ is used to show outcome of the transmission as follows:

$$A_{m,n} = \begin{cases} 1, & \text{Collision with other device/s;} \\ 0, & \text{Successful transmission.} \end{cases} \quad (3)$$

The probability that a device of interest will have a collision with at least one of the other transmitting devices in the n^{th} frame of the m^{th} round, is computed as:

$$\begin{aligned}\alpha_{m,n} &:= \Pr\left(A_{m,n}=1\right)\\ &= 1-\left(1-\frac{1}{K}\right)^{M_{m,n}-1}.\end{aligned} \tag{4}$$

The probability of a successful transmission is computed as:

$$\Pr\left(A_{m,n}=0\right) := 1-\alpha_{m,n}. \tag{5}$$

After having a successful transmission, the corresponding devices wait until the next round. The successful devices can use a constant value other than 0 and 1 to keep the history of the frames in which these devices do not perform any transmission. Thus, each element $h_{m,n}$ in the history matrix $\mathbf{H}$ is defined as follows:

$$h_{m,n} = \begin{cases} A_{m,n}, & \text{Grant-free transmission;}\\ -1, & \text{No transmission.}\end{cases} \tag{6}$$

Once a device has built its history of R rounds, it can predict different network parameters as explained below.

As shown in Fig. 1, all the active devices start transmitting at the beginning of a round, and each device can have only one successful transmission in a round. So, the probability that a device of interest remains successful in the m^{th} round is computed as:

$$\begin{aligned}P_{s_m} &:= \sum_{n=1}^{N}\left(1-\alpha_{m,n}\right)\prod_{\substack{j=1\\ n>1}}^{n-1}\alpha_{m,j}\\ &= \left(1-\alpha_{m,1}\right)+\left(1-\alpha_{m,2}\right)\alpha_{m,1}+...\\ &\quad+\left(1-\alpha_{m,N}\right)\alpha_{m,1}\alpha_{m,2}...\alpha_{m,N-1}\\ &= 1-\alpha_{m,1}+\alpha_{m,1}-\alpha_{m,1}\alpha_{m,2}+\alpha_{m,1}\alpha_{m,2}+...\\ &\quad-\alpha_{m,1}\alpha_{m,2}...\alpha_{m,N-1}\alpha_{m,N}.\end{aligned} \tag{7}$$

All the terms except first and last terms of Eq. (7), are cancelled out and Eq. (7) is reduced to:

$$P_{s_m} = 1-\prod_{n=1}^{N}\alpha_{m,n}. \tag{8}$$

Thus P_{s_m} can be computed by applying Eq. (4) in Eq. (8) and it gets the following form:

$$P_{s_m} = 1-\prod_{n=1}^{N}\left\{1-\left(1-\frac{1}{K}\right)^{M_{m,n}-1}\right\}. \tag{9}$$

As indicated in Eq. (9), for the given values of K and N, the computation of P_{s_m} at device-level requires knowledge of the number of transmitting devices ($M_{m,n}$) in each frame of a round, and this information is not available at the end-devices directly. Although, the number of active devices ($M_{1,m}$) at the start of a round is assumed to remain constant for a given observation interval, but the number of transmitting devices in the successive frames decreases randomly, and the collision probability varies accordingly. Thus, we have: $0 \le \alpha_{m,N} \le \alpha_{m,N-1} \le, ..., \le \alpha_{m,1} < 1,\ \forall m$. Consequently, the probability of a successful transmission P_{s_m}, can vary in different rounds. So, as a statistical measure of P_{s_m}, we aim to predict the average probability of success P_s in the given observation interval defined as:

$$P_s := \frac{1}{R}\sum_{m=1}^{R}P_{s_m}. \tag{10}$$

By using Eq. (9) in Eq. (10), P_s gets the following form:

$$P_s = \frac{1}{R}\sum_{m=1}^{R}\left[1-\prod_{n=1}^{N}\left\{1-\left(1-\frac{1}{K}\right)^{M_{m,n}-1}\right\}\right]. \tag{11}$$

The IoT devices can predict the quantities S_{av} and P_s by first predicting the number of transmitting devices in each frame. For that purpose, we define a parameter vector $\widehat{\mathbf{\Theta}} = \left[\widehat{M}_1, \widehat{M}_2, ..., \widehat{M}_N\right]$, where $\widehat{M}_n$ is the prediction of $M_{m,n}, \forall m$, and accordingly $\widehat{\alpha}_n$ is the prediction of $\alpha_{m,n}, \forall m$. We consider the case where the number of transmitting devices in a given frame of all rounds represents a wide-sense stationary process, and the number of transmitting devices in the n^{th} frame can be predicted as $\mathrm{E}\left[M_{m,n}\right]$, and the corresponding collision probability is predicted as $\mathrm{E}\left[\alpha_{m,n}\right]$. However, due to the random nature of the number of failures in each frame, the system can have huge number of states, and the computation of $\mathrm{E}\left[M_{m,n}\right]$ and $\mathrm{E}\left[\alpha_{m,n}\right]$ becomes cumbersome for $n > 1$.

In this paper, we enable the end-devices to predict the desired quantities S_{av} and P_s statistically from the transmissions history matrix $\mathbf{H}$ which involves the prediction of the number of transmitting devices and the associated collision probability in each frame. The proposed prediction method uses the fact that all elements in the first column of $\mathbf{H}$ are independent and identically distributed (IID) random variables, and the ML estimator of $\alpha_{1,m}, \forall m$ is given as [28]:

$$\widehat{\alpha}_1 = \frac{1}{R}\sum_{m=1}^{R}A_{m,1}. \tag{12}$$

Moreover, we can readily show that $\widehat{\alpha}_1$ is an unbiased estimator of $\alpha_{m,1},\ \forall m$, i.e., $\mathrm{E}\left[\widehat{\alpha}_1\right] = \alpha_{m,1},\ \forall m$. We can determine $\widehat{M}_1$ by applying $\widehat{\alpha}_1$ in Eq. (4) and it comes out to be:

$$\begin{aligned}\widehat{M}_1 &= 1+\frac{\ln\left(1-\widehat{\alpha}_1\right)}{\ln\left(\frac{K-1}{K}\right)}\\ &= 1+\frac{\ln\left(1-\frac{1}{R}\sum_{m=1}^{R}A_{m,1}\right)}{\ln\left(\frac{K-1}{K}\right)}.\end{aligned} \tag{13}$$

When R is large, $\widehat{\alpha}_1$ approaches to $\alpha_{m,1},\ \forall m$, and as a result $\widehat{M}_1$ approaches $M_{m,1},\ \forall m$. Thus, for the given round size, accuracy in the prediction of $M_{m,1}$ can be enhanced by using an appropriate value of R. Since the failed devices from the current frame transmit in the next frame of a given round, we use the average number of failures from the current frame as a prediction of the number of transmitting devices in the next frame:

$$\begin{aligned}\widehat{M}_n &= \widehat{M}_{n-1}\widehat{\alpha}_{n-1}, \quad n=2,3,...,N\\ &= \widehat{M}_1\prod_{j=1}^{n-1}\widehat{\alpha}_j.\end{aligned} \tag{14}$$

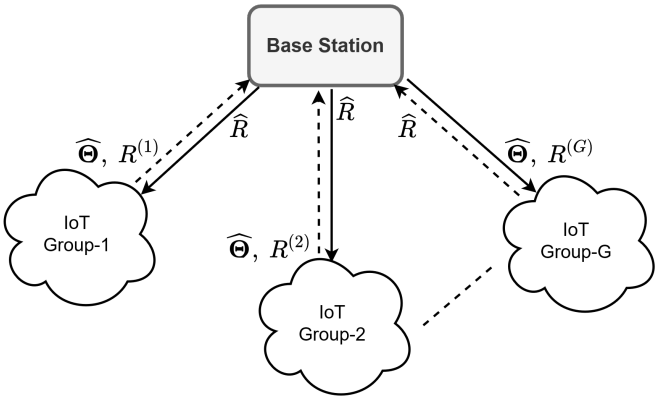

Fig. 3. Identification of different IoT groups at the BS.

The corresponding average collision probability in each frame for $n > 1$ is predicted as:

$$\widehat{\alpha}_n = 1 - \left(1 - \frac{1}{K}\right)^{\widehat{M}_n - 1}; \; n = 2, 3, ..., N. \tag{15}$$

Thus, the overall process to predict $\widehat{\boldsymbol{\Theta}}$ at the device-level is described as follows:

$$\widehat{M}_n = \begin{cases} 1 + \frac{\ln\left(1 - \frac{1}{R}\sum_{m=1}^{R} A_{m,1}\right)}{\ln\left(\frac{K-1}{K}\right)}, & n = 1; \\ \left[1 + \frac{\ln\left(1 - \frac{1}{R}\sum_{m=1}^{R} A_{m,1}\right)}{\ln\left(\frac{K-1}{K}\right)}\right] \prod_{j=1}^{n-1} \widehat{\alpha}_j, & \\ \qquad\qquad n = 2, 3, ..., N. \end{cases} \tag{16}$$

where

$$\widehat{\alpha}_n = \begin{cases} \frac{1}{R}\sum_{m=1}^{R} A_{m,1}, & n = 1; \\ 1 - \left(1 - \frac{1}{K}\right)^{\widehat{M}_n - 1}, & n = 2, 3, ..., N. \end{cases} \tag{17}$$

The average number of successful devices in the n^{th}-frame can be predicted as $\widehat{M}_n (1 - \widehat{\alpha}_n)$, thus S_{av} is predicted as:

$$\begin{aligned} \widehat{S}_{av} &:= \sum_{n=1}^{N} \widehat{M}_n (1 - \widehat{\alpha}_n) \\ &= \widehat{M}_1 (1 - \widehat{\alpha}_1) + \widehat{M}_1 \widehat{\alpha}_1 (1 - \widehat{\alpha}_2) + \widehat{M}_1 \widehat{\alpha}_1 \widehat{\alpha}_2 (1 - \widehat{\alpha}_3) \\ &\quad + ... + \widehat{M}_1 \widehat{\alpha}_1 \widehat{\alpha}_2 ... \widehat{\alpha}_{N-1} (1 - \widehat{\alpha}_N) \\ &= \widehat{M}_1 (1 - \widehat{\alpha}_1 + \widehat{\alpha}_1 - \widehat{\alpha}_1 \widehat{\alpha}_2 + \widehat{\alpha}_1 \widehat{\alpha}_2 - \widehat{\alpha}_1 \widehat{\alpha}_2 \widehat{\alpha}_3 \\ &\quad + ... + \widehat{\alpha}_1 \widehat{\alpha}_2 ... \widehat{\alpha}_{N-1} - \widehat{\alpha}_1 \widehat{\alpha}_2 ... \widehat{\alpha}_{N-1} \widehat{\alpha}_N). \end{aligned} \tag{18}$$

All the terms except first and last terms inside the parenthesis are cancelled out, and Eq. (18) is reduced to:

$$\widehat{S}_{av} = \widehat{M}_1 \left(1 - \prod_{n=1}^{N} \widehat{\alpha}_n\right). \tag{19}$$

By using Eq. (17) in Eq. (19), $\widehat{S}_{av}$ can be computed as:

$$\widehat{S}_{av} = \widehat{M}_1 \left[1 - \prod_{n=1}^{N} \left\{1 - \left(1 - \frac{1}{K}\right)^{\widehat{M}_n - 1}\right\}\right]. \tag{20}$$

We can compute $\widehat{P}_s$ as the prediction of P_s as follows:

$$\begin{aligned} \widehat{P}_s &:= \sum_{n=1}^{N} (1 - \widehat{\alpha}_n) \prod_{\substack{j=1 \\ n>1}}^{n-1} \widehat{\alpha}_j \\ &= (1 - \widehat{\alpha}_1) + (1 - \widehat{\alpha}_2) \widehat{\alpha}_1 + ... + (1 - \widehat{\alpha}_N) \widehat{\alpha}_1 \widehat{\alpha}_2 ... \widehat{\alpha}_{N-1} \\ &= 1 - \widehat{\alpha}_1 + \widehat{\alpha}_1 - \widehat{\alpha}_2 \widehat{\alpha}_2 + \widehat{\alpha}_1 \widehat{\alpha}_2 + ... - \widehat{\alpha}_1 \widehat{\alpha}_2 ... \widehat{\alpha}_{N-1} \widehat{\alpha}_N. \end{aligned} \tag{21}$$

All the terms except first and last terms of Eq. (21) are cancelled out, and Eq. (21) is reduced to:

$$\widehat{P}_s = 1 - \prod_{n=1}^{N} \widehat{\alpha}_n. \tag{22}$$

Thus $\widehat{P}_s$ can be computed by applying Eq. (4) in Eq. (22) and it gets the following form:

$$\widehat{P}_s = 1 - \prod_{n=1}^{N} \left\{1 - \left(1 - \frac{1}{K}\right)^{\widehat{M}_n - 1}\right\}. \tag{23}$$

We can see through Eq. (20) and Eq. (23) that both parameters $\widehat{S}_{av}$ and $\widehat{P}_s$ are functions of the predicted number of transmitting devices in each frame i.e., the vector $\widehat{\boldsymbol{\Theta}}$. It is worth noting that the end-devices can predict $\widehat{\boldsymbol{\Theta}}$, $\widehat{P}_s$ and $\widehat{S}_{av}$ by employing a statistical learning approach that uses the outcomes of their previous transmissions. The end-devices share the predicted network load with the BS as shown in Fig. 3, which can utilize this knowledge to optimize the radio resource allocation. Moreover, the computation burden at the BS is reduced by allowing the end-devices to predict the network load, which can result in overall latency reduction. This device-level network exploration strategy does not require any additional assistance from the BS except the values of N and K. Thus, we can use these features to design self-configuring networks where network parameters change dynamically.

B. IoT-groups identification

For URLLC based systems, the reliability is defined as the probability of satisfying a latency bound in a given network [5]. The heterogeneous IoT devices present in a network can be grouped virtually based upon their application specific statistical reliability constraints. This grouping of IoT devices can assist the BS to optimize the radio resource allocation in conditions where network parameters change dynamically. The mobile vehicular IoT entities belong to the same group as long as their latency-reliability requirements do not change and remain in the serving BS's coverage area. However, if they leave the coverage area, the total number of active devices changes, leads to changes in the resources required by different groups to meet their application-specific QoS requirements.

For the system model under consideration, we develop a statistical learning based strategy to identify different groups at the BS. This scheme involves the device level prediction of the number of rounds required to have a successful transmission such that desired statistical reliability constraint is satisfied. Each device shares this statistical knowledge with the BS which can identify the number groups and their latency-reliability constraints.

Our problem of identifying different groups at the BS reduces to predict the vector parameter $\widehat{\mathbf{\Theta}}$ and $\widehat{P}_s$ at device-level. After predicting $\widehat{\mathbf{\Theta}}$ and $\widehat{P}_s$ the end-devices can compute the optimal number of rounds needed for a successful transmission against their group-specific reliability criterion $\epsilon_r^{(i)}, \forall i = 1, 2, ..., \mathcal{G}$. In order to do that, we let the random variable X indicate the number of rounds that a device from group-i executes to get its first successful transmission, i.e., the device remains unsuccessful in $X-1$ consecutive rounds before getting a successful transmission in the round number X. The random variable X follows the geometric distribution. Under the group-specific reliability constrain $\epsilon_r^{(i)}$, the optimal value of X can be predicted as follows:

$$R^{(i)} = \inf_X \left\{ 1 \leq X \leq R_{max}^{(i)} : \widehat{P}_s \sum_{x=1}^{X} \left(1 - \widehat{P}_s\right)^{x-1} \geq 1 - \epsilon_r^{(i)} \right\}. \tag{24}$$

where $R_{max}^{(i)}$ is related to the group specific maximum affordable latency. The value of $R_{max}^{(i)}$ depends upon the nature of the environment in which communication is being carried out. If a device could not find an appropriate value of $R^{(i)}$ from Eq. (24), this indicates that the current environment cannot support the particular mission-critical communication application and the event is termed as an outage. Since the IoT devices can predict the outage event; therefore we can design an intelligent back-off mechanism which is part of our future research work.

As shown in Fig. 3, each device shares the locally learned value of $R^{(i)}$ with the BS, which uses this information to identify different groups present in the network and also their latency-reliability requirements. In addition to that, the BS can determine the number of successful devices from each group. The BS can utilize the information shared by the IoT devices to optimize the radio resource allocation based upon the latency-reliability criteria of different groups in the network. Moreover, as explained in Subsection V-B, the BS uses the information of the number of active devices to determine the optimal value of R such that the end-devices can predict different network parameters under desired prediction accuracy constraints.

C. Device-Level prediction of average latency (μ_L)

When the number of active devices vary dynamically, the random component of latency causes significant variations in the overall latency offered by the network. For example, in V2X communication scenarios, the mobile vehicles can cause random variations in the network latency. So, it becomes very essential for the heterogeneous devices to evaluate the feasibility of executing a particular mission-critical application in a dynamic environment. Acquiring the statistical knowledge of the random latency can be very useful in this regard. The average number of (re)transmissions performed by a device for a successful transmission is a measure of the average latency (μ_L), and we devise a statistical learning method to acquire knowledge of μ_L at the end-devices.

In order to have an analytical model for the prediction of average latency, which can be used at the device-level, let the random variable Y show the number of rounds a device remained failed before a successful transmission. Since we assume that the number of active devices at the start of each round remains fixed for the given observation interval, this makes all rounds independent of each other with a constant average probability of success per round predicted as $\widehat{P}_s$. Thus, the random variable Y follows the geometric distribution, and the expected value of Y is computed as: $\mathrm{E}[Y] = \frac{1-\widehat{P}_s}{\widehat{P}_s}$. By applying Eq. (23), the $\mathrm{E}[Y]$ comes out to be:

$$\mathrm{E}[Y] = \frac{\prod_{n=1}^{N} \widehat{\alpha}_n}{1 - \prod_{n=1}^{N} \widehat{\alpha}_n}. \tag{25}$$

Now given that a device of interest remains successful in the round followed by the Y failed rounds, let the random variable Z denote the number of (re)transmissions performed for the successful transmission in that round. Since the probability of collision varies in each frame of a round, the random variable Z follows a truncated geometric distribution with a variable probability of success in each frame. The probability mass function (PMF) of the random variable Z is defined as:

$$\Pr(Z = z) = \begin{cases} \dfrac{1}{1 - \prod_{n=1}^{N} \widehat{\alpha}_n} (1 - \widehat{\alpha}_z) \prod\limits_{\substack{j=1 \\ z>1}}^{z-1} \widehat{\alpha}_j, & z = 1, 2, ..., N; \\ 0, & \text{Otherwise.} \end{cases} \tag{26}$$

We can readily show that $\sum_{z=1}^{N} \Pr(Z = z) = 1$. While $\mathrm{E}[Z]$ is computed as:

$$\begin{aligned} \mathrm{E}[Z] &:= \sum_{z=1}^{N} z \Pr(Z = z) \\ &= \frac{1}{1 - \prod_{n=1}^{N} \widehat{\alpha}_n} \sum_{z=1}^{N} z (1 - \widehat{\alpha}_z) \prod_{\substack{j=1 \\ z>1}}^{z-1} \widehat{\alpha}_z \\ &= \frac{1}{1 - \prod_{n=1}^{N} \widehat{\alpha}_n} \left[1 - \widehat{\alpha}_1 + ... + N (1 - \widehat{\alpha}_N) \prod_{j=1}^{N-1} \widehat{\alpha}_j \right]. \end{aligned} \tag{27}$$

Eq. (27) is reduced to:

$$\mathrm{E}[Z] = \frac{1}{1 - \prod_{n=1}^{N} \widehat{\alpha}_n} \left(1 + \sum_{\substack{z=1 \\ N>1}}^{N-1} \prod_{j=1}^{z} \widehat{\alpha}_j - N \prod_{j=1}^{N} \widehat{\alpha}_j \right). \tag{28}$$

Thus, the average latency in terms of no. (re)transmissions per successful transmission can be predicted as follows:

$$\widehat{\mu}_L := N \mathrm{E}[Y] + \mathrm{E}[Z]. \tag{29}$$

By using Eq. (25) and Eq. (28) in Eq. (29), we get following expression of $\widehat{\mu}_L$:

$$\widehat{\mu}_L = \frac{1}{1 - \prod_{n=1}^{N} \widehat{\alpha}_n} \left(1 + \sum_{\substack{z=1 \\ N>1}}^{N-1} \prod_{j=1}^{z} \widehat{\alpha}_j \right). \tag{30}$$

We can see through (30) that the end-devices are enabled to predict average latency present in the network by using the prediction of collision probability in each frame of a round. On the other hand, as shown in (17), the computation of collision probability in each frame requires the availability of knowledge regarding the number of transmitting devices which is computed through (16). For the system model under consideration, the average latency per successful transmission is upper bounded by $\frac{N}{\widehat{P}_s}$, i.e., $\widehat{\mu}_L \leq \frac{N}{\widehat{P}_s}$.

Remarks: An interesting insight of the PMF given in Eq. (26) is that the truncated geometric distribution presented in [29] can be obtained directly from Eq. (26) as a special case when collision probability remains constant in each frame. It is also worth noting that when $N = 1$, Eq. (30) provides the expectation of a geometric random variable that shows the expected number of (re)transmissions performed for a successful transmission under constant collision probability, and the optimal number of (re)transmissions under the given latency-reliability constraint is presented in [28]. Thus, the PMF in Eq. (26) can be useful in analyzing networks in which the probability of collision varies as a result of the variable number of transmitting devices.

Algorithm 1 describes the steps of the proposed device-level network exploration mechanism, and each active device runs the algorithm every R rounds. The BS periodically broadcasts the values of K, N, and R to be used by the end-devices as the inputs for Algorithm 1. Initially, the BS broadcasts an initial value of R to explore the network. As explained in Subsection V-C, the initial value of R is selected such that the probability of exception in the network load prediction remains negligibly small for relatively low to high network load. After executing R rounds of the restricted grant free transmissions as shown in Fig. 1, the active devices predict $\widehat{\mathbf{\Theta}}$, S_{av}, P_s, $R^{(i)}$ and μ_L. The devices share their knowledge of the current network load with the BS, as shown in Fig. 3. For a given network load, the BS broadcasts the optimal value of R, which is computed according to the desired prediction accuracy constraints, as explained in Subsection V-B. Thus after every R rounds, the end-devices update their knowledge of network conditions by capturing the current network load, and the status of different QoS metrics.

Algorithm 1 Device-Level Network Exploration

Input: K, N and R
Output: $\widehat{\mathbf{\Theta}}$, $\widehat{P}_s$, $\widehat{S}_{av}$, $\widehat{\mu}_L$ and $R^{(i)}$

for $m = 1$ to R **do**
  for $n = 1$ to N **do**
    Select a channel randomly
    Transmit data
    if (success) **then**
      $A_{m,n} := 0$
      Stop transmitting in current round
      $h_{m,j} := -1;\ \forall j = n+1, n+2, ..., N$
    else
      $A_{m,n} := 1$
    end if
  end for
end for
Predict $\widehat{\mathbf{\Theta}} = \left[\widehat{M}_1, \widehat{M}_2, ..., \widehat{M}_N\right]$ from Eq. (16)
Predict $\widehat{S}_{av}$ from Eq. (20)
Predict $\widehat{P}_s$ from Eq. (23)
Predict $R^{(i)}$ from Eq. (24)
Predict $\widehat{\mu}_L$ from Eq. (30)
Share learned parameters $\widehat{\mathbf{\Theta}}$ and $R^{(i)}$ with the BS
Get the optimal $\widehat{R}$ from the BS
Update number of rounds $R := \widehat{R}$
return $\widehat{\mathbf{\Theta}}$, $\widehat{P}_s$, $\widehat{S}_{av}$, $\widehat{\mu}_L$ and $R^{(i)}$

V. Performance Analysis and Comparison

In this section we discuss the performance of the proposed statistical learning-based device-level network exploration mechanism. Since, for the given values of K and N, the only information available at the end-devices is the history of their transmissions, the performance of the proposed prediction mechanisms under the given network load is affected by the size of the history window. The value of R is also related to the amount of time required by the end-devices to learn different network parameters. So, in order to analyze the performance of the device-level network exploration, we evaluate the MSE associated with the prediction of the parameters S_{av}, P_s, and μ_L denoted by MSE_S, MSE_P, and MSE_μ respectively.

The MSE in the prediction of S_{av} is computed as follows:

$$\mathrm{MSE}_S = \mathrm{E}\left[\left\{S_{av} - \widehat{S}_{av}\right\}^2 \mid \mathbf{H}\right]. \tag{31}$$

Expectation is taken with respect to S_{av} defined in Eq. (2). By using Eq. (2) and Eq. (20), in Eq. (31), the MSE_S can be computed as:

$$\mathrm{MSE}_S = \mathrm{E}\left[\left\{\frac{1}{R}\sum_{m=1}^{R}\sum_{n=1}^{N} M'_{m,n} - \widehat{M}_1 + \widehat{M}_1 \prod_{n=1}^{N}\left\{1 - \left(1 - \frac{1}{K}\right)^{\widehat{M}_n - 1}\right\}\right\}^2\right]. \tag{32}$$

The MSE in the prediction of average probability of success per round is given as:

$$\mathrm{MSE}_P = \mathrm{E}\left[\left\{P_s - \widehat{P}_s\right\}^2 \mid \mathbf{H}\right]. \tag{33}$$

Expectation is taken with respect to P_s defined in Eq. (11). By using Eq. (11) and Eq. (23) in Eq. (33), the MSE_P can also be written as:

$$\mathrm{MSE}_P = \mathrm{E}\left[\left\{-\frac{1}{R}\sum_{m=1}^{R}\prod_{n=1}^{N}\left\{1 - \left(1 - \frac{1}{K}\right)^{M_{m,n}-1}\right\} + \prod_{n=1}^{N}\left\{1 - \left(1 - \frac{1}{K}\right)^{\widehat{M}_n - 1}\right\}\right\}^2\right]. \tag{34}$$

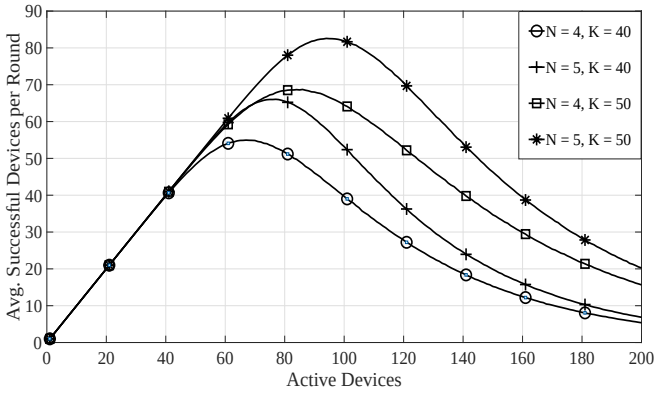

Fig. 4. Average successful devices per round with different values of K and N against active devices.

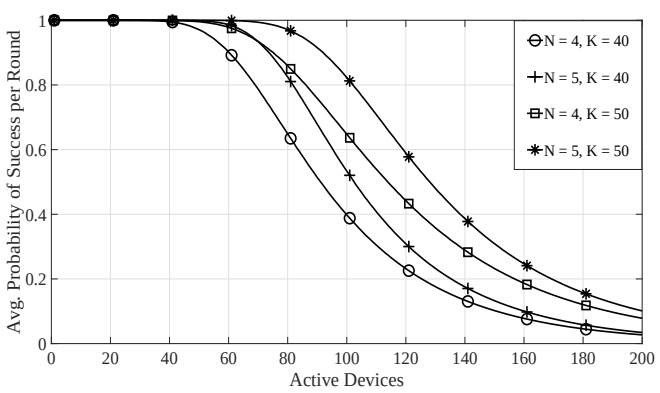

Fig. 5. Average probability of success per round with different values of K and N against active devices.

The MSE in the prediction of average latency is given as:

$$\mathrm{MSE}_{\mu} = \mathrm{E}\left[\left\{\mu_L - \widehat{\mu}_L\right\}^2 \mid \mathbf{H}\right]. \tag{35}$$

Expectation is taken with respect to μ_L defined in Eq. (37). In order to define μ_L, we use the average collision probability in each frame for the given observation interval defined as:

$$\alpha_n := \frac{1}{R}\sum_{m=1}^{R} \alpha_{m,n}. \tag{36}$$

By using Eq. (4) in Eq. (36), and replacing $\widehat{\alpha}_n$ in Eq. (30) with the resultant expression of α_n, we get the following expression for μ_L:

$$\mu_L := \frac{1 + \sum\limits_{\substack{z=1\\N>1}}^{N-1}\left[\prod\limits_{j=1}^{z}\frac{1}{R}\sum\limits_{m=1}^{R}\left\{1-\left(1-\frac{1}{K}\right)^{M_{m,j}-1}\right\}\right]}{1-\prod\limits_{n=1}^{N}\frac{1}{R}\sum\limits_{m=1}^{R}\left\{1-\left(1-\frac{1}{K}\right)^{M_{m,n}-1}\right\}}. \tag{37}$$

By applying Eq. (37) and Eq. (30) in Eq. (35), the MSE_{μ} gets the following form:

$$\begin{aligned}\mathrm{MSE}_{\mu} &= \mathrm{E}\left[\left\{\frac{1 + \sum\limits_{\substack{z=1\\N>1}}^{N-1}\left[\prod\limits_{j=1}^{z}\frac{1}{R}\sum\limits_{m=1}^{R}\left\{1-\left(1-\frac{1}{K}\right)^{M_{m,j}-1}\right\}\right]}{1-\prod\limits_{n=1}^{N}\frac{1}{R}\sum\limits_{m=1}^{R}\left\{1-\left(1-\frac{1}{K}\right)^{M_{m,n}-1}\right\}}\right.\right.\\ &\qquad \left.\left. - \frac{1 + \sum\limits_{\substack{z=1\\N>1}}^{N-1}\prod\limits_{j=1}^{z}\left\{1-\left(1-\frac{1}{K}\right)^{\widehat{M}_n-1}\right\}}{1-\prod\limits_{n=1}^{N}\left\{1-\left(1-\frac{1}{K}\right)^{\widehat{M}_n-1}\right\}}\right\}^2\right]. \end{aligned} \tag{38}$$

Due to the random nature of $M_{m,n}$ and $M'_{m,n}$ in each frame of the history window, it gets complicated to obtain the closed-form expressions of MSEs through Eqs. (32), (34) and (38). In this paper, we compute them numerically by using the Monte Carlo simulation method.

A. Simulation results

The Monte Carlo simulation method is used to analyze the performance of the proposed mechanisms for device-level network exploration. First, we analyze the behaviour of different network parameters against varying network load. For different values of N and K, we take a range of the number of active devices and use $R = 10,000$ number of independent rounds for the averaging purpose. The average number of successful devices per round defined in (2) is plotted in Fig. 4. The average probability of success per round is computed according to (10) and plotted in Fig. 5. It is observed that for lower values of M, the parameter P_s does not change much, and consequently, S_{av} increases. However, for the higher values of M, the parameter P_s decreases and S_{av} decreases accordingly. The average number of (re)transmissions for a successful transmission is presented in Fig. 6, where we can see that the parameter μ_L increases slowly under the smaller values of M, and rapidly under the larger values of M.

For IoT groups identification, three groups are considered with the respective reliability criterion $\epsilon_r^{(1)} = 10^{-3}$, $\epsilon_r^{(2)} = 10^{-4}$, and $\epsilon_r^{(3)} = 10^{-5}$. The optimal number of rounds required by these three different IoT groups to meet the required latency-reliability criteria are plotted in Fig. 7. It is observed that for the lower number of active devices at the start of each round, due to the discrete nature of the parameter R, different groups can have the same optimal number of rounds against their latency-reliability requirements. However, as the probability of success decreases as a result of an increase in the number of active devices, different groups start attaining distinct values of $R^{(i)}$.

The end-devices are enabled to predict different network parameters as explained in Section IV. In order to analyse MSE_S, MSE_P, and MSE_{μ}, associated with prediction of S_{av}, P_s, and μ_L, respectively, simulations are performed over a range of rounds for different values of M by using $K = 40$ and $N = 4$. While $N_s = 10,000$ iterations are used for each value of R to compute the MSEs numerically. The MSEs of $\widehat{S}_{av}$, $\widehat{P}_s$ and $\widehat{\mu}_L$ are demonstrated in Fig. 8, Fig. 9 and Fig. 10

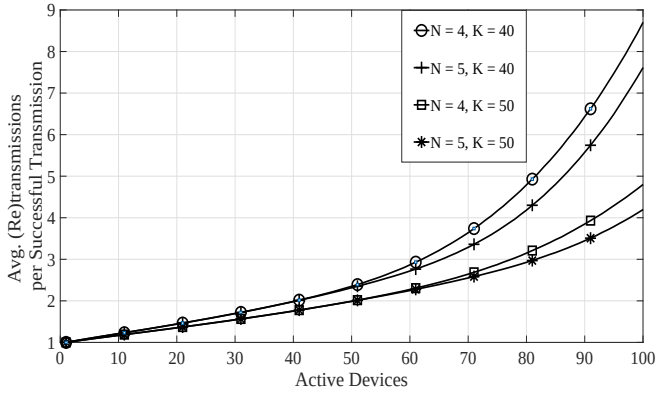

Fig. 6. Average (re)transmissions per successful transmission with different values of K and N against active devices.

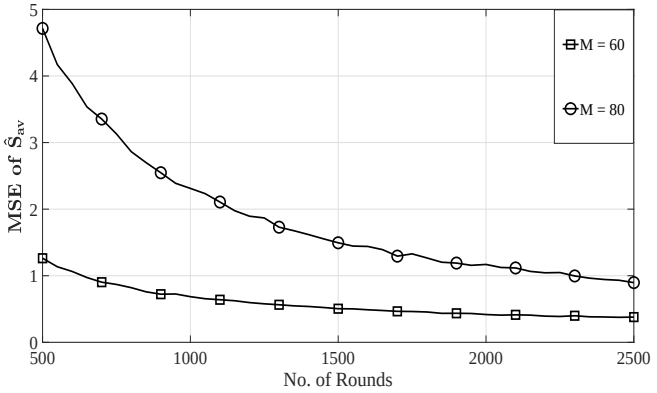

Fig. 8. MSE in the prediction of average successful devices per round with $K = 40$ and $N = 4$.

respectively. It is observed that for a given network load, these MSEs are decreased as the number of rounds is increased. Thus, the desired performance of these prediction methods can be achieved by using an appropriate value of R. In addition to that, as demonstrated in Figs. 8-10, when the number of active devices is increased, the IoT devices need to use a higher value of R to maintain the desired MSEs.

1) Network exploration delay: The time required by IoT devices for network exploration also reflects the performance of the proposed statistical learning-based prediction mechanisms. The end-devices run Algorithm 1 to predict different network parameters after each R number of rounds. Therefore, considering PHY-layer abstraction, the number of rounds executed by the end-devices measures the time required to explore the network. Since each round is composed of N frames, the network exploration delay is NR frames, while the PHY layer defines the frame duration. We can see through Figs. 8-10, that a higher prediction accuracy requires a larger value of R. In other words, the end-devices would need more time to predict different network the parameters if desired accuracy level increases. Moreover, when network load changes, the required number of rounds against the desired prediction accuracy also changes. Therefore, the network exploration delay is variable. In the following subsection, we explain the computation of the optimal value of R under the desired accuracy constraints.

B. Optimal size of the history matrix

From the perspective of mission-critical applications, the end-devices need to adapt to the network dynamics in the least possible time. Since the end-devices have limited power, computation, and memory resources, they need the minimum amount of data (transmissions history) to predict different network parameters. On the other hand, the prediction of these parameters should provide reasonable accuracy, which depends on the value of R for the fixed network load. Thus, an optimal value of R is essential to know so that the end-devices can learn different network parameters while meeting the related constraints of time to learn, storage and accuracy. For that purpose, we can use the asymptotic behavior of the above-defined MSEs against R. Let ζ_S, ζ_P and ζ_μ be the acceptable MSE in the prediction of $S_{av}, P_s,$ and μ_L respectively, the optimal number of rounds can be obtained by solving the following:

$$\widehat{R} = \min R \tag{39}$$

subject to :

$$\begin{aligned} & R \geq 1, \\ & \text{MSE}_S \leq \zeta_S, \\ & \text{MSE}_P \leq \zeta_P, \\ & \text{MSE}_\mu \leq \zeta_\mu. \end{aligned}$$

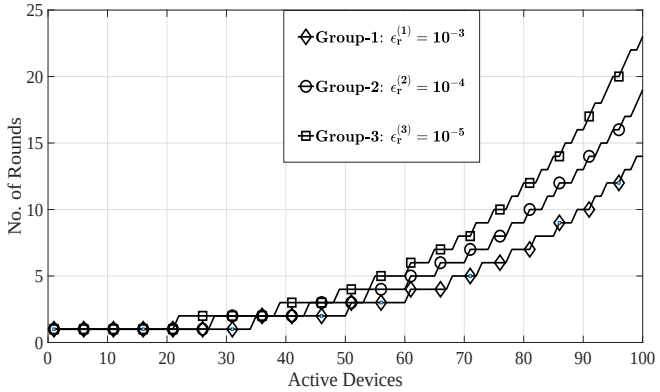

Fig. 7. No. of rounds required to meet the desired reliability with $K = 40$ and $N = 4$.

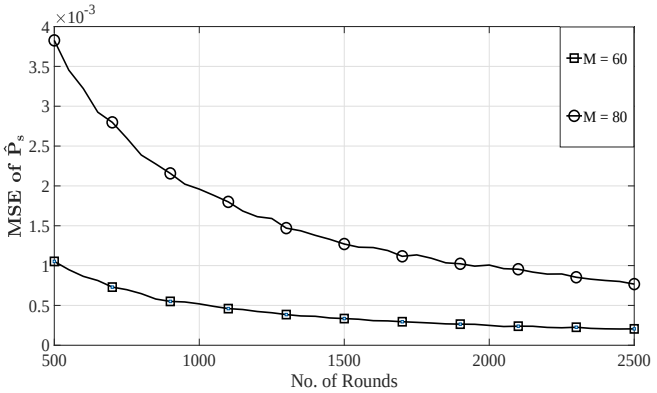

Fig. 9. MSE in the prediction of average probability of success per round with $K = 40$ and $N = 4$.

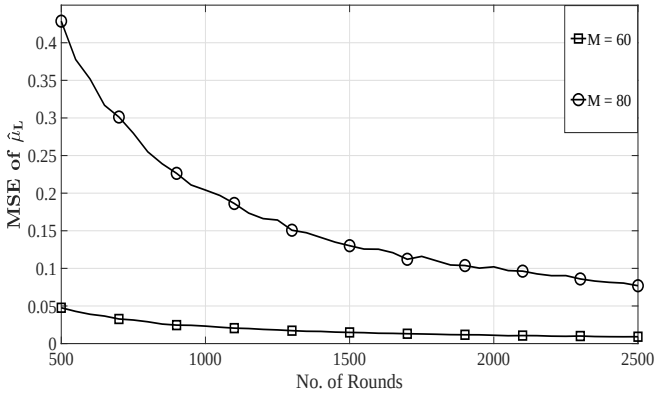

Fig. 10. MSE in the prediction of average latency with $K = 40$ and $N = 4$.

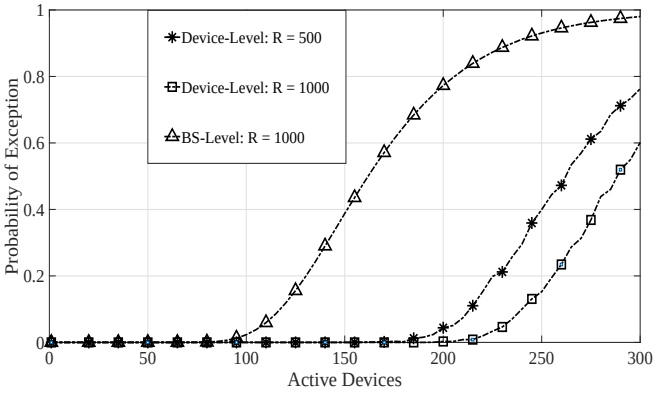

Fig. 11. Comparison of the probability of exception in the estimation of M

Since the closed-form expressions of the MSEs related to different parameters are not available, the MSEs are computed numerically for a range of R against different values of the network load. These MSEs decrease monotonically when the value of R is increased, as shown in Figs. 8-10, . Therefore, for each value of M and the given values of K, N, ζ_S, ζ_P, and ζ_L, the BS can have a lookup table to store the corresponding unique value of $\widehat{R}$. Initially, the BS broadcasts an initial value of R such that the end-devices can predict the current network load with a very small probability of exception, as demonstrated in the following Subsection V-C. After receiving the information regarding the current network load from the end-devices, the BS periodically broadcasts the optimal value $\widehat{R}$ according to the desired prediction accuracy constraints and current network load. Thus, the end-devices can update the size of their history matrix accordingly. We illustrate the impact of M on the computation of $\widehat{R}$ through an example. When $M = 60$, $K = 40$, $N = 4$, $\zeta_S = 1$, $\zeta_P = 0.001$, and $\zeta_\mu = 0.1$, by using the MSEs plotted in Fig. 8-10, we obtain $\widehat{R} \approx 650$. However, for $M = 80$, under the same prediction accuracy constraints, we get $\widehat{R} \approx 2300$.

The significance of the optimal value of R, denoted by $\widehat{R}$, has many folds. The parameter $\widehat{R}$ can be used to determine the devices' storage requirements and the minimum time required to explore and adapt to the network dynamics. In addition, the optimal value of R can be used to determine the energy requirements of the IoT devices for network exploration. In this regard, for the given network load, the transmission energy of $N\widehat{R}$ frames can be used as an upper bound for the energy consumption in device-level network exploration. Since a change in the current network load can impact the value of $\widehat{R}$, the energy consumption during network exploration varies accordingly.

C. Performance comparison

This Subsection presents a performance comparison regarding the robustness of the proposed statistical learning-based device-level network load prediction and an existing BS-level network load estimation. For the given size of an observation interval R, the performance of the proposed device-level network exploration mechanism is affected by the number of active devices $M_{m,1}$ present in the network. Thus, the accuracy in the prediction of the number of active devices $\widehat{M_1}$ plays a significant role in improving the overall performance of the proposed mechanism. The computation of $\widehat{M_1}$ through Eq. (13), requires $\widehat{\alpha}_1 < 1$ so that we can have a valid argument for the $\ln(.)$ function. We define exception as an event in which the argument of function $\ln(.)$ becomes zero which corresponds to the case when $\widehat{\alpha}_1 = 1$. The probability of exception η_{M_1} in the computation of $\widehat{M_1}$ is obtained empirically as follows:

$$\eta_{M_1} := \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{1}\left(\widehat{\alpha}_1^{(i)} = 1\right). \tag{40}$$

where

$$\mathbf{1}\left(\widehat{\alpha}^{(i)} = 1\right) = \begin{cases} 0 & \text{if} \quad \widehat{\alpha}_1^{(i)} \neq 1; \\ 1 & \text{if} \quad \widehat{\alpha}_1^{(i)} = 1. \end{cases} \tag{41}$$

For the given value of R, Eq. (40) provides the relative frequency of the exception occurring in N_s iterations (observation intervals) i.e., the number of times $\widehat{\alpha}_1$ gets value 1 in N_s iterations, while $\widehat{\alpha}_1^{(i)}$ is the prediction of $\alpha_{m,1}$ in the i^{th} iteration, and it is computed through (12).

We compare the probability of exception of the proposed device-level method with the one presented in [17, Section IV-C] to estimate the number of active devices at the BS. The estimation method in [17] uses the number of idle preambles in a frame during the contention phase in LTE-A random access procedure. By following the BS centered approach of [17], the number of transmitting devices in each frame can be estimated as follows:

$$\widehat{M_n^{(BS)}} = \frac{1}{R} \sum_{m=1}^{R} \frac{\ln\left(\frac{K_{m,n}}{K}\right)}{\ln\left(\frac{K-1}{K}\right)}. \tag{42}$$

where $\widehat{M_n^{(BS)}}$ is an estimate of number of transmitting devices and $K_{m,n}$ is the number of unused channels in the n^{th} frame. This method works well as long as the argument of $\ln(.)$ function is greater than zero i.e., $K_{m,n} > 0$. However, when value of $M_{m,n}$ gets larger, the probability of having zero idle channels becomes significantly large. Thus, an exception occurs when $K_{m,n} = 0$, and the probability of exception in

this case is computed empirically as follows:

$$\eta_{M_n}^{(BS)} := \frac{1}{N_s R} \sum_{i=1}^{N_s} \sum_{m=1}^{R} \mathbf{1}\left(K_{m,n}^{(i)} = 0\right). \tag{43}$$

where

$$\mathbf{1}\left(K_{m,n}^{(i)} = 0\right) = \begin{cases} 0 & \text{if} \quad K_{m,n}^{(i)} \neq 0; \\ 1 & \text{if} \quad K_{m,n}^{(i)} = 0. \end{cases} \tag{44}$$

where $K_{m,n}^{(i)}$ is the number of idle channels in the n^{th} frame for the i^{th} iteration. For the given value of R, Eq. (43) provides the relative frequency of the exception occurring in N_s iterations (observation intervals) i.e., the number of times $K_{m,n}$ gets value 0 in N_s iterations.

In Fig. 11, we have demonstrated η_{M_1} and $\eta_{M_1}^{(BS)}$ against a range of number of transmitting devices for different values of R with $K = 40$ and $N = 4$. For each value of M, we performed $N_s = 2500$ iterations to compute the probability of exception empirically. It is observed that both η_{M_1} and $\eta_{M_1}^{(BS)}$ are extremely small for a low to moderate network load. However, as the number of active devices is increased further, $\eta_{M_1}^{(BS)}$ becomes significantly large as compared to η_{M_1}. Moreover, for the given network load, η_{M_1} is further reduced by increasing value of R. In contrast to that, the computation of $\widehat{M}_{m,n}^{(BS)}$ only depends upon the number of idle channels in a frame, and increasing value of R does not reduce the probability of exception $\eta_{M_1}^{(BS)}$. Hence, the proposed statistical learning-based device-level network load prediction mechanism is more robust than the BS-centered approach in an environment where a large number of IoT devices communicate with a single BS over limited shared resources.

The above discussion highlights that the proposed statistical learning-based network exploration mechanism enables the IoT devices to get an insight of the network condition by predicting S_{av}, P_s, μ_L, and $R^{(i)}$. At the same time, a variation in the number of active devices impacts these network parameters. Therefore, the significance of the knowledge regarding the number of active devices available at the end-devices is further strengthened. It is noteworthy that any change in the number of active devices can be tracked and adapted accordingly by the IoT devices as long as that change remains stable for at least $\widehat{R}$ number of rounds. This feature can be used to design adaptive networks in which end-devices can learn the network dynamics with the least amount of data according to the desired accuracy. Since the BS utilizes information provided by the end-devices, this approach can yield overall latency reduction by reducing the computational overheads at the BS. Therefore, the proposed grant-free access can also improve the energy consumption in the energy-constrained delay-sensitive IoT applications.

VI. Conclusion and Future Work

Providing the URLLC interfaces for mission-critical IoT applications in dynamic heterogeneous networks is challenging, and vehicular communication is an important use case of such systems. Statistical learning is a promising tool for predicting dynamically varying parameters and learning associated probability distributions in heterogeneous networks. At the same time, the device-level network exploration can reduce the computation overheads at the BS, which results in overall latency reduction. This paper presents a statistical learning-based network exploration mechanism for heterogeneous mission-critical-IoT applications employing framed ALOHA-based restricted transmission strategy, enhancing reliability. The proposed grant-free network access mechanism is greatly suitable for designing heterogeneous networks in which mobile vehicular IoT entities communicate with other IoT devices over shared radio resources. The work presented in this paper enables the end devices to use their transmission history to predict different dynamic parameters in a probabilistic manner. Through simulations, the performance of the proposed prediction mechanisms is evaluated, and the optimal size of the history matrix is determined, enabling the end-devices to explore the network under the given accuracy constraints. Compared to the BS-centered approach, the device-level statistical learning-based network load prediction mechanism proposed in this paper is more robust against heavy network load.

This work can open new research avenues in on-device intelligence for 5G and beyond wireless communication systems. In this regard, as future research work, we aim to extend the current approach for fully decentralized heterogeneous networks while covering device-assisted radio resource management. Moreover, we also aim to design an intelligent back-off algorithm that can be executed by the IoT devices in case of an outage event.

References

[1] I. Zhou, I. Makhdoom, N. Shariati, M. A. Raza, R. Keshavarz, J. Lipman, M. Abolhasan, and A. Jamalipour, "Internet of Things 2.0: Concepts, Applications, and Future Directions," *IEEE Access*, vol. 9, pp. 70 961–71 012, 2021.

[2] *5G; Service requirements for next generation new services and markets*, 3GPP Std. TS 22.261, Jun. 2018.

[3] A. Orsino, A. Ometov, G. Fodor, D. Moltchanov, L. Militano, S. Andreev, O. N. C. Yilmaz, T. Tirronen, J. Torsner, G. Araniti, A. Iera, M. Dohler, and Y. Koucheryavy, "Effects of Heterogeneous Mobility on D2D- and Drone-Assisted Mission-Critical MTC in 5G," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 79–87, 2017.

[4] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[5] M. S. Elbamby, C. Perfecto, C. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, "Wireless Edge Computing With Latency and Reliability Guarantees," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717–1737, 2019.

[6] M. Mukherjee, L. Shu, and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[7] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[8] L. M. Vaquero and L. Rodero-Merino, "Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, p. 2732, Oct. 2014. [Online]. Available: https://doi.org/10.1145/2677046.2677052

[9] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[10] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-Free Non-Orthogonal Multiple Access for IoT: A Survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1805–1838, 2020.
[11] J. Ding, D. Qu, H. Jiang, and T. Jiang, "Success Probability of Grant-Free Random Access With Massive MIMO," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 506–516, 2019.
[12] R. Abreu, G. Berardinelli, T. Jacobsen, K. Pedersen, and P. Mogensen, "A Blind Retransmission Scheme for Ultra-Reliable and Low Latency Communications," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.
[13] T. Chiang, H. Liang, S. Wang, and S. Sheu, "On Parallel Retransmission for Uplink Ultra-Reliable and Low Latency Communications," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.
[14] O. Galinina, A. Turlikov, S. Andreev, and Y. Koucheryavy, "Multi-Channel Random Access with Replications," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2538–2542.
[15] C. Boyd, R. Kotaba, O. Tirkkonen, and P. Popovski, "Non-Orthogonal Contention-Based Access for URLLC Devices with Frequency Diversity," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.
[16] A. Azari, C. Stefanovic, P. Popovski, and C. Cavdar, "Energy-Efficient and Reliable IoT Access Without Radio Resource Reservation," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 908–920, 2021.
[17] C. Oh, D. Hwang, and T. Lee, "Joint Access Control and Resource Allocation for Concurrent and Massive Access of M2M Devices," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4182–4192, 2015.
[18] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong, "D-ACB: Adaptive Congestion Control Algorithm for Bursty M2M Traffic in LTE Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9847–9861, 2016.
[19] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement Learning for Real-Time Optimization in NB-IoT Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.
[20] M. Angjelichinoski, K. F. Trillingsgaard, and P. Popovski, "A Statistical Learning Approach to Ultra-Reliable Low Latency Communication," *IEEE Transactions on Communications*, vol. 67, no. 7, pp. 5153–5166, 2019.
[21] R. Abreu, P. Mogensen, and K. I. Pedersen, "Pre-Scheduled Resources for Retransmissions in Ultra-Reliable and Low Latency Communications," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–5.
[22] C. A. Astudillo, F. H. S. Pereira, and N. L. S. da Fonseca, "Probabilistic Retransmissions for the Random Access Procedure in Cellular IoT Networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.
[23] N. Jiang, Y. Deng, O. Simeone, and A. Nallanathan, "Online Supervised Learning for Traffic Load Prediction in Framed-ALOHA Networks," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1778–1782, 2019.
[24] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proceedings of the IEEE*, vol. 107, no. 11, pp. 2204–2239, 2019.
[25] Z. Shafiq, R. Abbas, M. H. Zafar, and M. Basheri, "Analysis and Evaluation of Random Access Transmission for UAV-Assisted Vehicular-to-Infrastructure Communications," *IEEE Access*, vol. 7, pp. 12 427–12 440, 2019.
[26] H. Ye, L. Liang, G. Ye Li, J. Kim, L. Lu, and M. Wu, "Machine Learning for Vehicular Networks: Recent Advances and Application Examples," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 94–101, 2018.
[27] M. Noor-A-Rahim, Z. Liu, H. Lee, M. O. Khyam, J. He, D. Pesch, K. Moessner, W. Saad, and H. V. Poor, "6G for Vehicle-to-Everything (V2X) Communications: Enabling Technologies, Challenges, and Opportunities," 2020.
[28] M. A. Raza, M. Abolhasan, J. Lipman, N. Shariati, and W. Ni, "Statistical Learning-Based Dynamic Retransmission Mechanism for Mission Critical Communication: An Edge-Computing Approach," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020, pp. 393–396.
[29] S. Chattopadhyay, C. Murthy, and S. K. Pal, "Fitting truncated geometric distributions in large scale real world networks," *Theoretical Computer Science*, vol. 551, pp. 22–38, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397514003521