

A Mass-Processing Simulation Framework for Resource Management in Dense 5G-IoT Scenarios

Lusheng Wang¹, Kun Chang¹, Xiumin Wang², Zhen Wei^{1,3}, Qingxin Hu^{1,3} and Caihong Kai^{1*}

¹School of Computer and Information, Hefei University of Technology
Feicui Lake Campus 485 Danxia Road, Hefei, Anhui, China

²School of Computer Science and Engineering, South China University of Technology
381 Wushan Road, 510641, Guangzhou, Guangdong, China

³GOCOM Information & Technology Co. LTD
27 Tianzhi Road, 230088, Hefei, Anhui, China

*Corresponding author: Caihong Kai

[e-mail: wanglusheng@hfut.edu.cn, xmwang@scut.edu.cn, weizhen@hfut.edu.cn, chkai@hfut.edu.cn]

*Received March 26, 2018; revised April 23, 2018; accepted May 1, 2018;
published September 30, 2018*

Abstract

Because of the increment in network scale and test expenditure, simulators gradually become main tools for research on key problems of wireless networking, such as radio resource management (RRM) techniques. However, existing simulators are generally event-driven, causing unacceptably large simulation time owing to the tremendous number of events handled during a simulation. In this article, a mass-processing framework for RRM simulations is proposed for the scenarios with a massive amount of terminals of Internet of Things accessing 5G communication systems, which divides the time axis into RRM periods and each period into a number of mini-slots. Transmissions within the coverage of each access point are arranged into mini-slots based on the simulated RRM schemes, and mini-slots are almost fully occupied in dense scenarios. Because the sizes of matrices during this process are only decided by the fixed number of mini-slots in a period, the time expended for performance calculation is not affected by the number of terminals or packets. Therefore, by avoiding the event-driven process, the proposal can simulate dense scenarios in a quite limited time. By comparing with a classical event-driven simulator, NS2, we show the significant merits of our proposal on low time and memory costs.

This study was supported by Key Laboratory of Universal Wireless Communications (Beijing University of Posts and Telecommunications), Ministry of Education, P.R. China under grant No. KFKT-2016103, also in part supported by the National Natural Science Foundation of China under Grant Nos. 61501160, 61571178.

Keywords: 5G, Internet of things (IoTs), dense network, radio resource management (RRM), simulation

1. Introduction

With the fast development of wireless communication networks, such as Internet of Things (IoT), Internet of vehicles, and 5G communication systems, the number of wireless terminals and sensors has dramatically increased. Meanwhile, new applications in the above networks have gradually appeared, bringing us into a big data era with a massive amount of terminals in hyper-dense networks [1]. To research on such scenarios but because of limited site and funding, it is common to use computers for simulations instead of real-time tests with large-scale experimental networks [2].

Simulations for wireless communications and networks can be categorized into several levels based on the main objects to be simulated, such as link, system, and network. Link-level simulations focus on physical (PHY) layer techniques in communication links, such as modulation, channel coding, and interference cancellation using signal processing methods. They generally evaluate links' performance, such as bit error rate (BER) and block error rate, affected by pathloss, fading, and noise in communication channels [3]. System-level simulations are usually used for multi-cell and multi-terminal scenarios, which may be performed fast by simplifying the simulated links [4]. Network-level simulations can simulate routing, accessing, congestion control, etc. This category will be further summarized in the following paragraphs. In addition to the above categories, algorithm-level simulations can rapidly solve a mathematical or computational problem, whose results can be used to judge the optimality of a proposed scheme and inspire ideas for improving it. Both network-level and algorithm-level simulations are widely used for the study of radio resource management (RRM) in wireless networks. Algorithm-level simulations are simple and fast but they do not mimic the transmission of packets during the simulation. Classical network-level simulators [5]-[6] include NS, OPNET, OMNET, J-SIM, GloMoSim, QualNet, etc. They are used for simulations of schemes and protocols in various scenarios, but they are not specifically designed for simulations of RRM in wireless networks, so many researchers believe that self-developed simulators may be more suitable for their special purposes than these classical simulators. The existing self-developed simulators are summarized as follows.

Wireless sensor network is a key paradigm widely simulated by self-developed simulators, and WSNet [7] is a representative one for this purpose, which uses an event-driven mechanism to support packet transmission and propagation on a wireless medium. The platform designed in [8] is based on OMNET's environment, which uses a multi-agent model and an independent node's PHY architecture. Each terminal is an

autonomous agent and the source code is automatically translated by mapping from configured parameters to objective environment. EasiSim [9] is also an event-driven simulator, which realizes node, topology, and scenario components. Topology, time, and event queue are integrated to improve the platform's extensibility. Based on Tmote Sky sensor and Fox board platform, [10] uses a simulation model combining ZigBee and Bluetooth, which achieves low-power operation and photo transmission. Ref. [11] provides a software-defined architecture for IoT applications in heterogeneous networks, which uses a multilayer controller to achieve multiframe scheduling, network status evaluation, and scheme improvement. In [12], a platform is designed specifically for the study of energy-efficiency improvement in smart city, which supports a large number of terminals and interactive operations, but RRM or 5G scenario is not considered.

Long-term evolution (LTE) system is also a widely studied key paradigm. OAISim [13] is an open-source LTE simulator, which supports simulations on LTE standards with evolved node B (eNB), user equipment (UE), and evolved packet core (EPC). It also supports signaling and schemes standardized by 3GPP with an almost full protocol stack. Ref. [14] focuses on uplink study of modulation and coding, multiple-input-multiple-output (MIMO), multi-user scheduling, and other PHY layer key techniques. Meanwhile, this simulator uses Matlab's parallel computation toolbox for multi-core processing, which increases the computational complexity. Ref. [15] extends the above platform by pre-generating the fading parameters, which further decreases the computational complexity. LTE-Sim proposed in [16] is an open-source simulator, which can simulate multi-cell multi-user scheduling, mobility modeling, RRM, etc. Ref. [17] raises an open architecture with distributed configuration to decrease the reconfiguring time cost. In [18], a mobile gateway is designed in the scope of NS2 for the internetworking of ZigBee and mobile communication systems. Based on hierarchical addressing and network discovery techniques, routing from ZigBee to the mobile communication system is achieved. Based on OPNET, [19] proposes a simulator for the study of a mixture between universal mobile telecommunication system (UMTS) and wireless local area network (WLAN), which supports applications in UMTS, mobility models, handover, and call admission control. There are also some simulators specifically for the simulations of RRM schemes. Ref. [20] designs a policy-based distributed RRM architecture to achieve decision-sharing between networks and terminals. Ref. [21] uses OPNET to simulate multimedia applications in UMTS systems, specifically on RRM. The key RRM techniques, such as power control, handover, and load balancing are supported. LENA is an NS3-based simulator, which could simulate RRM schemes for LTE system. It supports LTE protocol stack, network structure, indoor propagation model, and RRM schemes [22]-[23].

To simulate communications in wireless networks, event-driven simulators are usually used, but they are too slow for simulations with a massive amount of nodes and packets due to the tremendous events generated during the simulation. Instead of node-by-node or packet-by-packet processing, RRM for all the nodes is achieved by a certain scheme and completed at the same time. Therefore, for RRM in hyper-dense networks with a massive amount of nodes and packets, there should be a better way to mass-process the packets. The main contribution of this article is the proposal of a mass-processing framework used for

simulations of the scenarios with a massive amount of nodes in hyper-dense networks. Different from common packet generation methods obeying a certain distribution of packet interval, the number of packets in hyper-dense networks is huge, making it a saturated buffer case when RRM allocates resource to transmit these packets. Thus, for each access point, the transmitted packets are almost continuous on the time axis. Given a certain value of period per bit and a certain probability distribution of packet length, the number of bits and the number of packets transmitted during each RRM period tend to be quite stable. Considering these features, we divide the time axis into RRM periods and each RRM period into a number of mini-slots. The length of each mini-slot is much smaller than a common packet, so each packet generally spans multiple mini-slots. After resource allocation by the RRM procedure and mini-slot assignment, packets are mapped to mini-slots they occupy. Finally, the performance of certain RRM schemes can be calculated by matrix operations on the mini-slots of an RRM period. In this way, even if we severely increase the number of nodes or packets in the simulation, the simulation time is not significantly increased, because the number of mini-slots in the simulated RRM period is fixed and the unallocated packets do not obviously affect the simulation time cost. Performance comparisons using the same scenarios show that the proposed framework achieves much lower time and memory costs than the classical event-driven simulators in hyper-dense scenarios.

The remainder of this article is organized as follows: in Section 2, the procedure of the proposed framework is described; in Section 3, the RRM techniques in the framework are explained; in Section 4, two typical 5G scenarios are designed and simulation results are shown to demonstrate the validity of the proposal. Finally, conclusions are drawn in Section 5.

2. Procedure of the Proposed Framework

The key module in event-driven simulators is the event scheduler that handles events to mimic all the processing during communications and networking. Everything happened in a simulation is described by events, such as application type events (packet generation), medium type events (transmission start, transmission end, reception start, reception end), and medium access control type events (channel detection and access backoff). Each event is defined by a data structure, including event type, event happening time, corresponding terminal ID, happening layer in protocol stack, event serial number, etc.

An event scheduler generates, sorts, and processes the events, as shown in [Fig. 1](#). Its key steps include scheduler pre-process, event add and sort, event read, processing function calling, and event delete. Since it spends a lot of time and effort to mimic each packet's detailed transmission, its merit is the accuracy for the simulations of packet processing, but it fits for the scenarios with only a small amount of terminals and not too much traffic. For simulations of hyper-dense networks, its merit leads to its main drawback instead, i.e., too time-consuming.

To solve the above problem, let us consider from another angle. During each RRM period, the number of packets transmitted from and to each base station (BS) is limited. The successful probability of each transmitted packet is highly related to the interference it suffers, which is further decided by the concurrently transmitted packets in adjacent cells. The division of a number of mini-slots could discretize the time axis to mimic packet scheduling and packet overlapping (interfering) features. Thus, multiple BSs and multiple mini-slots form the two dimensions of a matrix, making it possible to use matrix operations to simulate the transmissions of packets arranged by a certain RRM scheme.

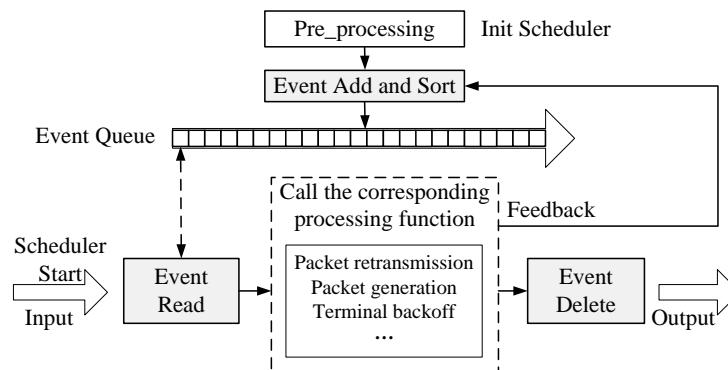


Fig. 1. The process of an event scheduler.

In this section, the design of the matrix-operation-based mass-processing framework is described, whose procedure is shown in **Fig. 2**. First, the framework is initialized and the user-configured parameters for simulated scenario, IoT application, and RRM schemes are set. Second, scenario details, such as positions of BSs and terminals in the simulation area are generated based on the configured scenario parameters. Third, all the packets during an RRM period are generated at the same time, including packet length, packet starting time, source ID, and destination ID, without payload. Note that generating all the packets at the same time is reasonable because the number of packets generated by each terminal and stored in its buffer is quite regular based on the configured IoT applications, so the ratio of the numbers of packets in different terminals' buffers is almost fixed. Meanwhile, the allocation strategy in each RRM period is actually for the packets generated before this period. Fourth, the RRM module is called to decide the BS to access for each terminal, the carrier frequency to use, and the power to allocate. Then, a mini-slot assignment procedure is used to map all the packets into the mini-slots based on each packet's transmitting time and packet length. Note that the number of mini-slots in an RRM period should not be too large or too small; otherwise, it either makes the simulation procedure too time-consuming or makes the simulation results imprecise. We set the level of RRM period to milliseconds and the level of mini-slot to microseconds in the following simulations. Finally, the performance on various metrics can be obtained.

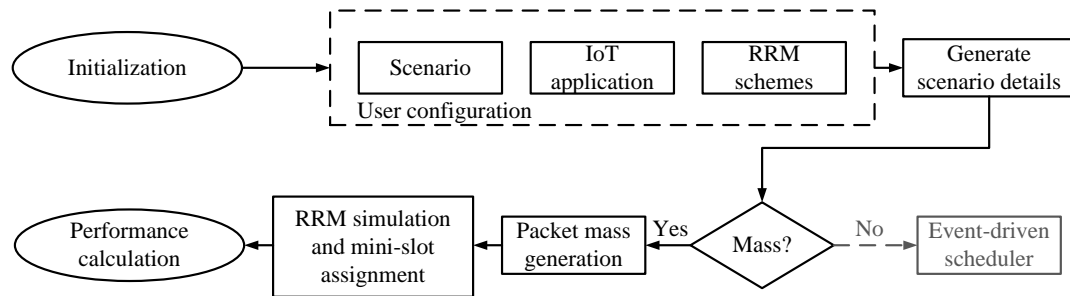


Fig. 2. Procedure of the proposed framework.

A traffic generator is used to mass-generate all the packets of a simulation. IoT traffic usually has the following features [24]–[25]: uplink more than downlink, and usually periodic with very small packets but sometimes bursty. The traffic is defined as packets with parameters including length, transmitting time, source ID, destination ID, etc. No detailed content (i.e., payload) is transmitted in each packet.

Mini-slot assignment is a major part of the procedure. The time axis is divided into very small virtual mini-slots and each packet is mapped to one or multiple mini-slots based on its transmission time and packet length. The element in each mini-slot indicates the serial number of the terminal occupying this mini-slot. Take Fig. 3 as an example, the first four elements in the second row are 6s, representing that a packet transmitted from terminal 6 to BS2 is allocated in these mini-slots. Therefore, each BS forms a vector of these serial numbers indicating the occupation of its mini-slots, and these elements for all the BSs form a matrix with the number of rows equaling the number of BSs and the number of columns equaling the number of mini-slots in one RRM period. The occupation of the mini-slots is based on RRM schemes and the mini-slot assignment procedure. RRM schemes will be further described in Section 3, which runs all the configured schemes to obtain the strategy of resource allocation to the terminals by each BS. The mini-slot assignment procedure further maps the packets with the mini-slots and records the first mini-slot and the last mini-slot of each packet for further performance calculation.

A dynamic RRM procedure running period by period is too time-consuming for the massive-node scenarios, so our framework is designed for the simulation of one period within a dynamic procedure, which is already quite time-consuming in dense scenarios. If a dynamic RRM procedure containing many periods is simulated, the simulation procedure should be repeated many times, which is actually easy to be achieved by a simple script (or a loop program) to call the main function many times from a higher level.

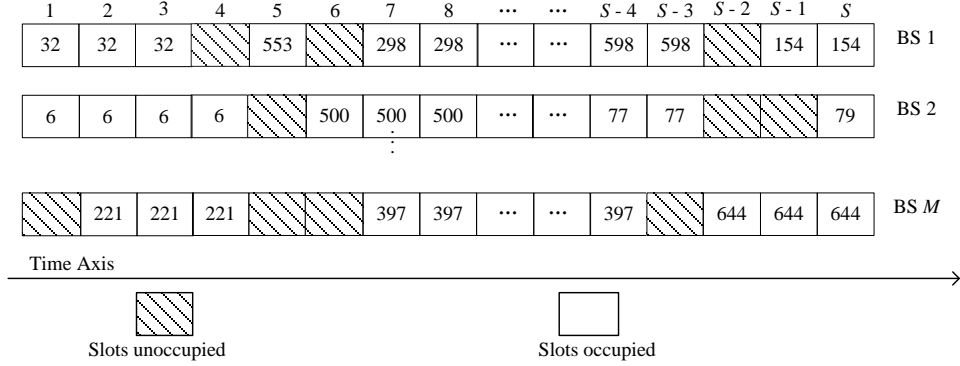


Fig. 3. Mini-slot assignment examples

The next step is the calculation of average signal-to-interference-plus-noise ratio (SINR), packet error rate (PER), and system capacity. By calculating the useful signal's strength and the interference in each mini-slot, we could obtain the SINR, PER, and system capacity for each mini-slot. Then, those performance metrics of a packet can be obtained by averaging the mini-slots it occupies, and those of each terminal can be also averagely calculated. Taking uplink transmission as an example, we consider the interference from terminals to BSs. Each mini-slot in **Fig. 3** is represented by a two-tuple $\{i, j\}$, where i denotes the index of BS and j denotes the index of mini-slot. In this way, the interference from the terminal occupying the j th mini-slot in the i th BS to the l th BS is written as

$$I_{il}^j = \frac{Pt_i^j}{PL_{il}^j * BPL_{il}^j} \quad (1)$$

where Pt_i^j is the transmitting power of the terminal occupying the j th mini-slot in the i th BS. PL_{il}^j and BPL_{il}^j are the pathloss and the building penetration loss from the terminal occupying the j th mini-slot in the i th BS to the l th BS. Therefore, the SINR of the l th BS in the j th mini-slot can be written as

$$sinr_l^j = \frac{Pr_l^j}{\sum_{i \neq l} I_{il}^j + n_0} \quad (2)$$

where n_0 is the variance of additive white Gaussian noise (AWGN), and Pr_l^j is the receiving power of the l th BS in the j th mini-slot. Therefore, the average SINR for the k th packet in the l th BS is written as

$$SINR_{lk} = \frac{\sum_{j=T_{lk}^x}^{T_{lk}^y} sinr_l^j}{T_{lk}^y - T_{lk}^x + 1} \quad (3)$$

where T_{lk}^x and T_{lk}^y are the serial numbers of the first and the last mini-slots that the k th packet occupies. Assuming that offset quadra-phase shift keying (OQPSK) is used for modulation, we get the BER of the k th packet as

$$BER_{lk} = 1 - [1 - \frac{1}{2} * \text{erfc}(2 * SINR_{lk})]^2 \quad (4)$$

and the PER of the k th packet is given by

$$PER_{lk} = 1 - (1 - BER_{lk})^{N_k} \quad (5)$$

where N_k is the total number of bits in the k th packet. The channel capacity corresponding to the SINR of the k th packet in the l th BS is given by

$$C_{lk} = B * \log_2(1 + SINR_{lk}) \quad (6)$$

where B is the bandwidth for transmitting the k th packet. Therefore, the total system capacity is given by

$$C = \sum_{l=1}^M \sum_{k=1}^{NP_l} C_{lk} \quad (7)$$

where NP_l is the total number of packets transmitted in the l th BS and M is the total number of BSs in the scenario.

The advantage of the proposed framework is mainly its high processing speed and the reasons are summarized as follows:

1) The process of an event-driven simulator is based on the time points of events (such as packet transmission) in the event queue, which can happen at any time point. In a dense scenario, the number of events may be huge, so the time points when events happen may be extremely close to one another (i.e., quite successive on the time axis). In contrast, packet transmissions in our framework are arranged to the mini-slots, and the division of time axis into mini-slots is a discretization. To sum up, the smallest unit for processing in an event-driven simulator is the event, while that in our framework is the mini-slot. By discretizing the time axis into appropriate number of mini-slots, we could guarantee the precision of the simulation and also make sure that the number of mini-slots is obviously less than the number of events.

2) For the calculation of the performance metrics in a simulation, an event-driven simulator mimics the procedure of each packet's transmission and outputs the results in a log file so that the performance metrics (such as the number of packets successfully transmitted) can be counted. In contrast, performance calculation in our framework is based on matrix operations, which processes all the mini-slots in an RRM period together, making it much faster than the one-by-one counting process of event-driven simulators.

3) When the number of terminals is very large, the number of events generated during a simulation becomes huge, making it quite time-consuming to complete a simulation by an event-driven simulator. In contrast, the number of mini-slots of each RRM period in our framework is fixed for a simulation, so the time cost for many processes during a simulation does not increase much. In detail, the time cost for the performance calculation step in [Fig. 2](#) is based on the number of mini-slots in an RRM period, which does not increase at all, even though the number of terminals may become very large. The time cost of the packet generation step is proportional to the number of packets generated, which is more or less the same as an event-driven simulator. The time cost of the RRM schemes may increase polynomially, but its increasing trend is just the same as the RRM schemes running by an event-driven simulator. To sum up, along with the increment of the number of terminals, the time cost of the proposed framework increases slower than event-driven

simulators. Since the processing time saved by the performance calculation step is comparable with event-by-event processing by event-driven simulators is large, this improvement should be quite promising.

Another advantage of the framework is its low memory cost. When the scenario is dense, event-driven simulators usually generate a huge number of events. Since the processing of the events is based on their happening time points, the huge number of events that are not processed yet should be stored in the memory, which leads to the occupation of too much memory and may further affect the processing speed during the simulation. When the simulated scenario is hyper-dense, event-driven simulators may become out of memory, which breaks the simulation directly. In contrast, the proposed framework is based on the processing on mini-slots and the mini-slots in each RRM period are fixed during a simulation. Therefore, even if we extremely increase the number of terminals, the usage of memory is not obviously increasing. In detail, the increment is only about the storage of the terminals' features and the generated packets' information, which are the same as in event-driven simulators, while the memory usage for the processing of mini-slots does not increase at all. This is a promising feature of the proposed framework for the simulations of hyper-dense scenarios.

3. RRM Techniques in the Framework

RRM in our framework is an independent module, which is composed of a pre-processing part and an algorithm processing part. The pre-processing part provides parameters for the simulated algorithms, while the algorithm processing part uses the configured algorithm to get the RRM result. Based on the network's current status and the requirements of the terminals, the RRM module is used to appropriately allocate resources to decrease inter-cell interference, increase spectral utilization, and guarantee the transmission of various applications. RRM techniques supported include cell selection, partial frequency reuse (PFR), dynamic interference coordination, and power allocation. The algorithm processing part provides interfaces to user's algorithms, so the user just needs to make sure that the designed algorithm is correct and calls it in the algorithm processing part. Algorithms already implemented include received signal strength (RSS) based scheme, Q-learning algorithm, simulated-annealing algorithm, Hungarian algorithm, genetic algorithm, etc.

3.1 Cell Selection

In the scenario of heterogeneous networks, each terminal might be covered by multiple cells (i.e., BSs) at the same time and it can select the most appropriate cell for access, called cell selection. Since the capacity of each cell is limited, when multiple terminals access one single cell, congestion may cause a decline in the performance of the whole system. Therefore, cell selection is a procedure that combines multiple factors, such as terminal properties, user preferences, network attributes, and application features, for decision making. The cell selection procedure in a simulation is divided into two steps:

- Preparation step, which first calculates the distances between the terminals and BSs, hence pathloss for each link is obtained. Then, it obtains the RSS for each terminal at the BS and its capacity based on the average SINR. In the end, the set of available cells for each terminal is obtained, which will be a key input for the cell selection algorithm.
- Decision step, which makes decision for each terminal on the cell to access and outputs the results by a log file. Taking RSS-based algorithm and Q-learning-based algorithm as examples, the former allows each terminal to select the cell with the maximum RSS, while the latter uses a Q-learning iteration process to gradually select the most appropriate cell for each terminal.

3.2 PFR

PFR divides the whole frequency band into multiple bands and divides each cell into multiple concentric circles. The bands are used by the concentric circles in an overlapped manner, so that the inter-cell interference can be decreased and the cell-edge terminals' performance can be guaranteed.

PFR schemes are involved in the design of the division of bands, the number of circles, and the reuse factors of different circles [26]. The current framework supports the simulation of PFR schemes with three or less concentric circles. As shown in Fig. 4, 61 adjacent cells are arranged regularly and their cell radii, circles, and reuse factors are set based on user configuration. The scenario that the user wants to simulate should be a subset of these 61 cells and at least one coincident should be found. For example, if a user configures a scenario with 5 cells to simulate, i.e., {A, B, C, D, E} shown on the left of Fig. 4, the simulation program will do a movement of these cells to the 61 cells on the right and try to find a coincident, such as {25, 9, 7, 16, 35} in the figure, to fix the configuration of these 5 cells.

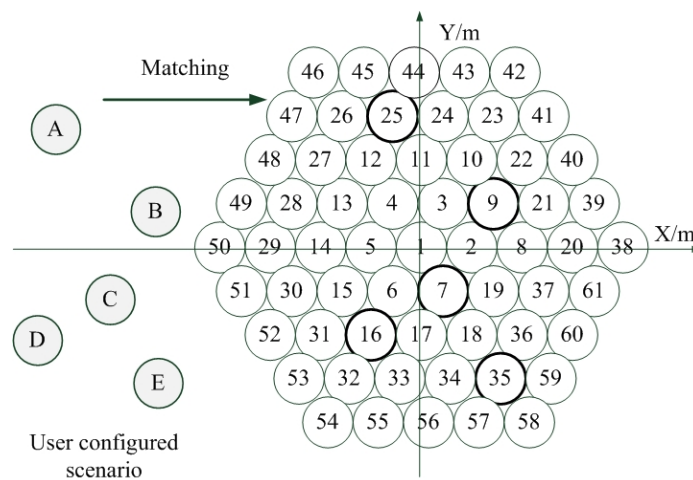


Fig. 4. PFR.

3.3 Dynamic Interference Coordination

Dynamic interference coordination divides a frequency band into frequency points (carriers) and a time period into slots, which form resource blocks. Then, a certain optimization algorithm is used to find an appropriate resource-allocation mechanism, i.e., a matching between resource blocks and terminals.

Dynamic interference coordination is used to allocate resources between circles in adjacent cells with the same frequency band in **Fig. 4**. The realization of this technique is as follows: first, based on the PFR scheme, we gather the circles using the same frequency band into the same group, hence all the circles are classified into multiple groups. Note that when the numbers of terminals are not the same in different circles within a group, we add some virtual terminals for the circles containing less terminals, so that all the circles have the same number of terminals. Then, a utility matrix is obtained representing the utilities for pairing different terminals. Finally, for each group, dynamic interference coordination is used to decide the resource blocks used for each terminal. For example, the mid-circle in cell 2 and the mid-circle in cell 4 will use the same frequency band when a 3-circle scheme is used for PFR, and dynamic interference coordination will be further used to match the terminals in these two circles to avoid severe interference. A certain optimization algorithm, such as genetic algorithm, simulated annealing algorithm, and Hungarian algorithm, is used to decrease interference and increase spectral efficiency.

3.4 Power Allocation

Power allocation adjusts the transmitting power of the terminals to avoid severe interference and increase system capacity. Power allocation decides the transmitting power values of terminals in adjacent cells, so the usage of this technique is similar to dynamic interference coordination. First, all the circles are classified into multiple groups in the same way as dynamic interference coordination. Second, for each group, virtual terminals are added to make the numbers of terminals in different circles identical. Third, for each pair of terminals, the optimal transmitting power ratio is obtained by the technique in [27]. In the end, a utility matrix is obtained and used as the input of the power allocation algorithm to decide the final results. Note that, if another power allocation scheme is simulated, we just need to add a new algorithm function with the same input and output interfaces to replace the current one, thus its corresponding RRM results will be obtained.

4. Simulation Results

Compared with the previous networks, 5G network will increase machine-to-machine applications' traffic [28], which brings about new features, such as large data rate, low latency, and low energy efficiency. It supports smart home, smart office, intelligent transportation, and many other new fields, which generate extremely large number of bursty or periodic very small packets [29]–[31]. Therefore, two typical scenarios, i.e., indoor office and outdoor grid road referring to 3GPP and MITIS-2, are designed and used for the verification of our proposed framework.

4.1 Outdoor Simulation Results

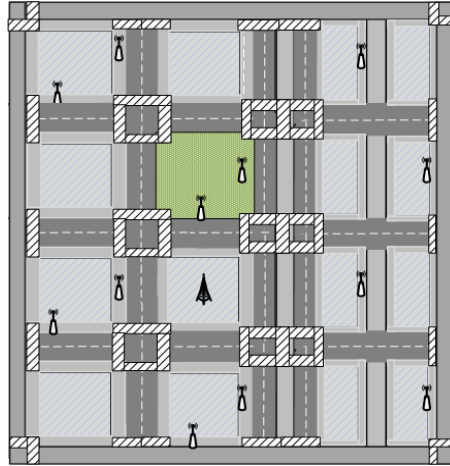


Fig. 5. Outdoor scenario.

The outdoor scenario can simulate the city's road, propagation model, and building penetration loss [29]. Meanwhile, it is also a typical heterogeneous cellular network scenario where a number of picocells are distributed in the coverage of macrocells to enhance the signal coverage in some hot-zones and to provide multiple access choices to terminals. The configuration procedure of the scenario is as follows: first, a two-dimensional Cartesian coordinate system is established for Fig. 5, whose original point is at the top left corner. Second, based on the positions of buildings and roads, we obtain their coordinates and coverage areas. Each building is modeled as a rectangle with fixed center and side length. Third, parameters such as transmitting power, antenna gain, carrier frequency, and channel bandwidth are set based on the corresponding 3GPP specifications and MITIS-2 documents [32]. Finally, we set the locations of the BSs and terminals and corresponding parameters for propagation and building penetration.

The configured parameters include simulation time, IoT application type, RRM schemes, terminal features, BS features, channel model, and other related parameters [31]-[33]. The simulation area is 387 meters * 553 meters with 1 macrocell and 12 picocells, while the number of terminals is from 100 to 10000 under uniform distribution. The transmitting power of terminal, picocell, and macrocell are set to 23, 40, and 49 dBm, while their antenna gains are set to 0, 6, and 12 dBi. The 3GPP ITU model is used for propagation with pathloss exponents and shadow standard deviations set as the above references. The total bandwidth is 20 MHz, the modulation mode is OQPSK, and the penetration loss per wall is set to 71.5 dB. The simulation time is one RRM period, 100 ms, divided into mini-slots with 10 us each. Period per bit is set to 50 ns. The simulated IoT application type is video monitoring whose packet length obeys lognormal distribution with $\mu = 6.6$ and $\sigma = 1.12$, while the packet interval is a constant (10 ms). The average packet length is obtained as $E(\text{packet_length}) = e^{\mu + \sigma^2/2} \approx 1376.4$, which is reasonable for video-monitoring

applications. The positions of the BSs are fixed as in **Fig. 5**, while the positions of terminals are randomly generated in the whole simulation area.

The packets allocated for transmission during this period are those generated in the previous periods. Before the simulation period starts, all the packets should be generated based on the configured application type. Then, cell selection in RRM module is called to associate all the terminals with their most appropriate BSs, and other RRM techniques are used to allocate the packets to the resource blocks. Finally, the mini-slot assignment procedure is called to determine the first and the last mini-slots occupied by each packet and store them for performance calculation.

Simulation results are shown in **Table 1**. We can see that the number of packets successfully transmitted is increasing along with the increment of the number of terminals. When the number of terminals is small, the number of generated packets is also small, so RRM could separate these packets into almost nonoverlapping mini-slots to avoid interference and further packet failure, leading to high packet successful rate. Along with the increment of the number of terminals, the packet successful rate gradually decreases due to interference. When the number of terminals is larger than a certain value (such as 5000 terminals in **Table 1**), the network becomes saturated, so RRM can only arrange an almost fixed number of packets to the mini-slots for each BS, leading to a stable packet successful rate. Because of the same reason, we can see that the number of successfully transmitted packets gradually increases until the network is saturated. For simulation time, it is monotone increasing, close to a linear level. As we know, time cost for the matrix operations part does not obviously change when the number of terminals is increasing. The main reason that increases the simulation time is the RRM schemes and the calculation of the channel features for each sender-receiver pair, which are related to the number of terminals.

Table 1. Performance for outdoor scenario

Num. nodes	Num. received packets	Successful rate	Time cost/S
200	1592	79.60%	26.62
500	3881	77.62%	60.74
1000	6865	75.27%	121.36
2000	9690	71.30%	248.72
5000	12018	68.80%	613.42
10000	12001	68.44%	1278.76

4.2 Indoor Simulation Results

Fig. 6 shows a typical 5G indoor scenario referring to MITIS-2 [29]. It considers the layout, furniture, and low-power BSs in a room, which is suitable for the simulations of bursty traffic generated by a massive amount of nodes in resource-limited hotspot zones.

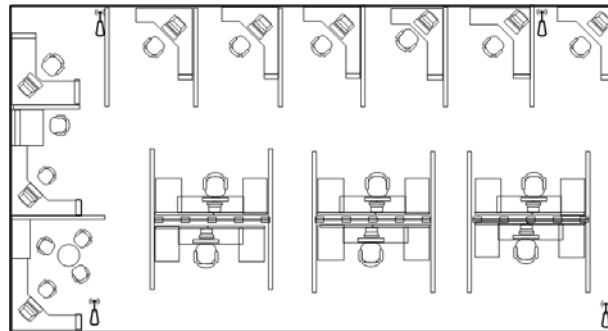


Fig. 6. Indoor scenario.

The simulation area is 20 meters * 10 meters with 4 femtocells, while the number of terminals is from 200 to 1000 under uniform distribution. Since the coverage of each BS in the indoor scenario is much smaller, as well as the number of terminals in this scenario, the time cost of RRM, including information gathering and RRM schemes, should be obviously shorter than the outdoor scenario. Therefore, the RRM period is set to 20 ms for this indoor scenario. The transmitting power of femtocells is set to 24 dBm and the antenna gain is 6 dBi. The propagation model is close-in free space reference distance path loss (CI-FSPL), the penetration loss per clapboard is 8 dB. The other parameters are the same as the outdoor scenario [31]-[33].

Each simulation result in **Table 2** is obtained by averaging the results of 10 randomly generated scenarios. The configuration on RRM is the same as the outdoor scenario, as well as the three evaluated performance metrics. Since the number of terminals for the indoor scenario is much smaller, the total number of successfully transmitted packets is generally less than the outdoor scenario. For the same reason, the packets have less chance to be interfered with each other, so the packet successful rate is much higher.

Comparing the cases with 200 terminals for both indoor and outdoor scenarios, we can see that the packet successful rate for the former is obviously better than the latter. That is because the indoor scenario has less BSs hence there is less average interference for each packet, although the average number of packets transmitted in unit time period is more or less the same. For simulation time, the cost of indoor scenario is less than that of outdoor scenario when the numbers of terminals are the same. That is because the indoor scenario has less BSs hence less repeats of RRM processing and less transmitter-receiver pairs for the calculation of channel features. Meanwhile, since the RRM period for indoor scenario is set to be 1/5 of the outdoor scenario, the size of matrices is also much less, which also decreases the time cost of the indoor scenario. Similar to the outdoor scenario, when the number of terminals is larger than a certain value (such as 800), the number of successfully transmitted packets becomes stable because almost all the resource blocks have been already occupied and almost no more packets can be further added by the RRM schemes.

Table 2. Performance for indoor scenario

No. nodes	Num. received packets	Successful rate	Time cost/S
200	362	90.50%	10.72
400	689	90.48%	21.73
600	806	88.95%	30.97
800	885	89.03%	40.46
1000	893	88.75%	50.72

4.3 RRM Simulation Results

This simulation includes cell selection, PFR, dynamic interference coordination, and power allocation. We choose Q-learning-based algorithm for cell selection. To avoid too long simulation time and make sure the results are typical for demonstration, we set 1 macrocell and 4 picocells in the scenario, numbered as 1 and 2-5, respectively. The picocells are distributed inside the macrocell. 100 terminals are uniformly distributed in the macrocell, hence some of them are also covered by picocells. For the other parameters, such as the channel model, we refer to the outdoor scenario in 4.1.

Simulation results are written into a log file, as shown in Fig. 7. Since Q-learning is used, each terminal's selection is to maximize the whole system's performance. We can see that, a large number of terminals select the macrocell because it covers a large area that the picocells may not cover.

For PFR simulation, we should assume that the cell selection procedure is completed and a PFR scheme is configured then. To make the scenario typical, we choose the cells 1-7 in Fig. 4 and uniformly distribute 15 terminals in each cell. We select a 3-circle scheme for simulation, so each cell is divided into 3 circles with reuse factors 1/3/7, representing reuse factor 1 for the inner circle, 3 for the middle circle, and 7 for the outer circle. Simulation results are shown in the log file of Fig. 8. We can see that, with the above configurations, the inner circles of the 7 cells use band numbers {1,1,1,1,1,1,1}, the middle circles use {3,4,2,4,2,4,2}, and the outer circles use {6,8,10,7,5,11,9}.

```
Cell_Selection_Result Node "6 7 9 11 12 13 14 15 16 19 21 22 23 24 25 26
29 30 31 33 35 37 38 39 41 43 44 45 46 47 49 50 51 52 53 54 55 56 57
58 59 61 62 63 64 65 66 67 69 70 71 72 75 76 77 79 81 82 86 87 88 89
90 92 93 95 97 98 100 101 102 104 105" select 【1】
Cell_Selection_Result Node "8 17 20 27 32 34 48 68 78 83 91 96 99" select 【2】
Cell_Selection_Result Node "10 40 85" select 【3】
Cell_Selection_Result Node "18 36 60 94 103" select 【4】
Cell_Selection_Result Node "28 42 73 74 80 84" select 【5】
```

Fig. 7. Cell selection results.

```

PFR Result
In BS "1" Inner_Node "10 14 68 93" Select_Frequency_Band "1"
In BS "2" Inner_Node "28 53" Select_Frequency_Band "1"
In BS "3" Inner_Node "57 59 60 94 98 100" Select_Frequency_Band "1"
In BS "4" Inner_Node "48 74 89" Select_Frequency_Band "1"
In BS "5" Inner_Node "111" Select_Frequency_Band "1"
In BS "6" Inner_Node "34" Select_Frequency_Band "1"
In BS "7" Inner_Node "24 97" Select_Frequency_Band "1"
In BS "1" Mid_Node "12 27 36 63 66 96" Select_Frequency_Band "3"
In BS "2" Mid_Node "11 15 18 32 39 69 88 91 102" Select_Frequency_Band "4"
In BS "3" Mid_Node "92" Select_Frequency_Band "2"
In BS "4" Mid_Node "23 61 99 103 104" Select_Frequency_Band "4"
In BS "5" Mid_Node "38 46 79 95 108" Select_Frequency_Band "2"
In BS "6" Mid_Node "9 25 26 31 43 56 72 106" Select_Frequency_Band "4"
In BS "7" Mid_Node "41 51 101" Select_Frequency_Band "2"
In BS "1" Outer_Node "8 17 29 50 76 82 105 107" Select_Frequency_Band "6"
In BS "2" Outer_Node "21 22 30 33 40 62 64 70 71 75 80 85 87" Select_Frequency_Band "8"
In BS "3" Outer_Node "16 35 44 52 55 81 84" Select_Frequency_Band "10"
In BS "4" Outer_Node "13 45 77 86 110" Select_Frequency_Band "7"
In BS "5" Outer_Node "58 65 78" Select_Frequency_Band "5"
In BS "6" Outer_Node "37 42 47 54 67 83 109" Select_Frequency_Band "11"
In BS "7" Outer_Node "19 20 49 73 90 112" Select_Frequency_Band "9"

```

Fig. 8. PFR results.

After cell selection and PFR, the terminals are associated with cells and the circles of cells are allocated with bands. The next step is to allocate resource to terminals in each circle, which involves in the dynamic coordination of interference between adjacent cells. To concisely show the results, we set 2 macrocells with fixed locations, such as cells 1 and 2 in Fig. 4. Fifty terminals are uniformly distributed in the two cells and associated with the cells by cell selection. Each cell is divided into 3 circles with reuse factors 1/3/7. After dynamic interference coordination with Hungarian algorithm, terminals from the 2 cells are paired. For example, as shown at the 3rd line of Fig. 9, terminals 14 and 22 are paired to use frequency 1, terminals 19 and 39 are paired to use frequency 5, etc.

Instead of only coordinating the interference, the above process can also coordinate transmitting power of terminals in adjacent cells, i.e., power allocation. Assuming that cell selection and PFR have been completed, the input matrix for Hungarian algorithm is generated, including power allocation and interference coordination effects. Seen from Fig. 10, terminals are paired and their transmitting power values are obtained. For example, the allocated power values for terminal 4 and terminal 5 are 97.9819 mw and 102.0181 mw, while those of terminal 6 and terminal 13 are 112.3668 mw and 87.6332 mw, etc.

```

Dynamic_Interference_Coordination_Result
In_BS "2" Inner_Node "22 25 26 32 39" Select_Frequency_Point "1 2 3 4 5"
Inner_Node "22 39 25 32 26" Matching_Inner_Node "14 19 24 34 37" Select_Frequency_Point "1 5 2 4 3"
In_BS "1" Mid_Node "20 27" Select_Frequency_Point "14 15"
In_BS "2" Mid_Node "3 8 10 12 13 18 23 30" Select_Frequency_Point "22 23 24 25 26 27 28 29"
In_BS "1" Outer_Node "5 7 16 17 28 29 33 35 36 38 40 41"
Select_Frequency_Point "42 43 44 45 46 47 48 49 50 51 52 53"
In_BS "2" Outer_Node "4 6 9 11 15 21 31 42" Select_Frequency_Point "54 55 56 57 58 59 60 61"

```

Fig. 9. Dynamic interference coordination results.


```

Power_Allocation_Result
Node "3 7 30 35" Allocated Power "80.4070 127.8491 158.4752 113.6863" mw
Node"9 29 39 41" Matching Node "3 7 30 35"
Allocated Power "119.5930 72.1509 41.5248 86.3137" mw
Node "5 13 16 20 21 22 24"
Allocated Power "102.0181 87.6332 96.0770 107.0468 83.5395 80.2506 88.2712" mw
Node"4 6 10 17 33 36 40" Matching Node "5 13 16 20 21 22 24"
Allocated Power "97.9819 112.3668 103.9230 92.9532 116.4605 119.7494 111.7288" mw

```

Fig. 10. Power allocation results.

4.4 Comparison on Simulation Time and Memory Costs

In this subsection, we compare the simulation time cost and memory cost of the proposed framework with the traditional even-driven simulators. These simulators usually achieve time costs in the same order of magnitude for simulating the same scenario, so one of them is chosen as a representative milestone [34]. NS2 is selected because it is mature with many manuals, widely accepted by this research domain, and easy to guarantee identical scenarios and packet transmissions with our framework because of its explicit user interfaces and tools. The RRM period is set to 10 ms, which is smaller than the previous simulations so that NS2 does not take too much time to complete the simulation. The period of each mini-slot is set to 10 us and the period per bit is 50 ns. To guarantee a generic comparison, we do not use the scenarios in the previous two subsections. Instead, we consider a scenario with a 400 meters * 400 meters simulation area with 16 BSs randomly deployed within it. A number of nodes, from 160 to 3000, are uniformly distributed in the simulation area and their transmission power is set to 20 dBm. Since the simulation time cost is correlated to the number of packets processed during the simulation, we guarantee that the number of processed packets by our proposed framework and that by NS2 are identical, while the packet length and packet interval are set to constant values 1000 bits and 1 ms, forming a constant bit rate application.

Comparison results are shown in **Fig. 11**. Along with the increment of the number of nodes, both the simulation time cost and memory cost of NS2 dramatically increase. Take the case with 1600 nodes as an example, NS2 costs almost 10 h and 6300 MB of memory to complete this simulation. Note that since the time cost of NS2 are polynomially increasing, it becomes extremely large when we further increase the number of nodes. Also note that this is only the time cost for one simulation, and researchers usually need to do a lot of simulations during the study, making event-driven simulators unsuitable for the studies of hyper-dense scenarios. In contrast, since the proposed framework uses matrix operations, especially if combining the matrix operation features of Matlab, the time cost does not increase too much when the number of nodes becomes large. For example, when 1600 nodes are simulated, our framework costs only about 34 s. Combining with **Table 1** and **Table 2**, we find that the time cost of our framework is linearly increasing with regard to the number of nodes.

For RRM simulations, when all the resource blocks are occupied by packets, there is no further resource to transmit more packets, so the increment of packet processing costs should be limited even if the number of nodes further increases. Event-driven simulators do

not show this feature because their packet processing costs are still increasing due to the fact that the number of generated events is still dramatically increasing. Instead, the packet processing cost of our proposed framework becomes gradually stable when the resource is almost fully occupied, no matter what kind of massive-node scenario is simulated. The increment of time cost of our framework along with the number of nodes is because of the increment of the simulation time of RRM module and channel feature calculation. This explains the major merit of our proposed framework on simulation time cost. Moreover, because the number of mini-slots during this simulation is fixed and Matlab generates vectors with fixed number of entries, the memory cost of our proposed framework does not increase at all. Therefore, when the number of nodes is small, the memory occupied by NS2 is smaller than our framework, but along with the increment of the number of nodes, its memory cost dramatically increases to store the events.

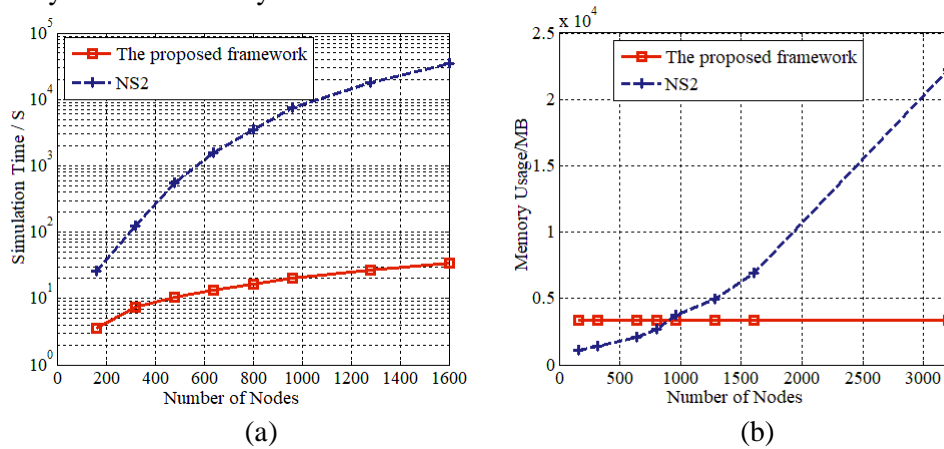


Fig. 11. Performance comparison: (a) simulation time costs, (b) memory costs

5. Conclusion

Tremendous events, as the smallest processing units, can only be processed one-by-one in traditional event-driven simulators, making them too time-consuming for simulations of hyper-dense networks. The proposal in this article is a matrix-based mass-processing framework for simulations of RRM in hyper-dense 5G-IoT scenarios, which divides the time axis into multiple RRM periods and each RRM period into multiple mini-slots. The key point is that the discretization effect of the time axis is strong, so the number of mini-slots, compared with the number of events during an event-driven simulation, decreases several orders of magnitude. Therefore, the time cost of the proposed framework is much less than the traditional event-driven simulators for the same simulation, making it much more suitable for RRM simulations in 5G-IoT scenarios. To complete the functionalities of the proposed framework, a traffic generator for IoT applications, two 5G scenarios, and an RRM module containing many techniques and algorithms are also implemented. Our near future work is to add more algorithms into the framework, so that its context of use could be widely extended.

References

- [1] W. Chin, Z. Fan and R. Haines, "Emerging technologies and research challenges for 5G wireless networks," *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 106–112, April 2014. [Article \(CrossRef Link\)](#)
- [2] Y. Wang, J. Xu and L. Jiang, "Challenges of system-level simulations and performance evaluation for 5G wireless networks," *IEEE Access*, vol. 2, pp. 1553–1561, Jan. 2015. [Article \(CrossRef Link\)](#)
- [3] C. Yahiaoui, M. Bouhali and C. Gontrand, "Simulating the long term evolution (LTE) downlink physical Layer," in *Proc. of UKSim*, pp. 531–535, March 2014. [Article \(CrossRef Link\)](#)
- [4] L. Chenand, W. Chen, B. Wang, X. Zhang, H. Chen and D. Yang, "System-level simulation methodology and platform for mobile cellular systems," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 148–155, July 2011. [Article \(CrossRef Link\)](#)
- [5] <http://www.isi.edu/nsnam/ns/>
- [6] <http://www.opnet.com/>
- [7] <http://wsnet.gforge.inria.fr/>
- [8] A. Hamzi and M. Koudil, "Agent based platform for the design and simulation of wireless sensor networks," in *Proc. of International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp. 1–5, May 2012. [Article \(CrossRef Link\)](#)
- [9] H. Chen, L. Cui, H. Zhu and C. Huang, "EasiSim: a scalable simulation platform for wireless sensor network," in *Proc. of ICC Workshop*, pp. 184–188, May 2008. [Article \(CrossRef Link\)](#)
- [10] E. Capo-Chichi, H. Guyennet, J. Friedt, I. Johnson and C. Duffy, "Design and implementation of a generic hybrid wireless sensor network platform," in *Proc. of LCN*, pp. 836–840, Oct. 2008. [Article \(CrossRef Link\)](#)
- [11] Z. Qin, G. Denker, C. Giannelli, P. Bellavista and N. Venkatasubramanian, "A software defined networking architecture for the Internet-of-Things," in *Proc. of NOMS*, pp. 1–9, May 2014. [Article \(CrossRef Link\)](#)
- [12] E. Patti and A. Acquaviva, "IoT platform for smart cities: requirement and implementation case studies," in *Proc. of IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pp. 1–6, Sept. 2016. [Article \(CrossRef Link\)](#)
- [13] <http://www.openairinterface.org/>
- [14] C. Mehlfrer, M. Wrulich, J. Ikuno and D. Bosanska, "Simulating the long term evolution physical layer," in *Proc. of EUSIPCO*, pp. 1471–1478, Aug. 2009. [Article \(CrossRef Link\)](#)
- [15] J. Ikuno, M. Wrulich and M. Rupp, "System level simulation of LTE network," in *Proc. of IEEE VTC Spring*, pp. 1–5, May 2010. [Article \(CrossRef Link\)](#)
- [16] G. Piro, L. Grieco, G. Boggia, F. Capozzi and P. Camarda, "Simulating LTE cellular system: an open-source framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498–513, Feb. 2011. [Article \(CrossRef Link\)](#)
- [17] F. Clermidy, R. Lemaire, X. Popon, D. Ktenas and Y. Thonnart, "An open and reconfigurable platform for 4G telecommunication: concepts and application," in *Proc. of Euromicro Conference on Digital System Design, Architectures, Methods and Tools*, pp. 449–456, Aug. 2009. [Article \(CrossRef Link\)](#)
- [18] Z. Lei, Z. Ying, A. Chen and C. Liu, "A simulation platform for ZigBee-UMTS hybrid networks," *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 293–296, March 2013. [Article \(CrossRef Link\)](#)

- [19] C. Wang, W. Xia, L. Shen and T. Song, "A system level simulation platform for the study of QoS provisioning in UMTS/WLAN interworking," in *Proc. of International Conference on Information Science and Engineering*, pp. 2518–2521, Dec. 2009. [Article \(CrossRef Link\)](#)
- [20] S. Horrich, S. Jamaa and P. Houze, "Policy based RRM for network terminal decision sharing," *IEEE Veh. Technol. Mag.*, vol. 2, no. 3, pp. 35–40, Sept. 2007. [Article \(CrossRef Link\)](#)
- [21] A. Eduardo and H. Garzon, "Analysis and simulation of radio resource management for quality service in universal mobile telecommunications system," in *Proc. of IEEE ANDESCON*, pp. 1–6, Sept. 2010. [Article \(CrossRef Link\)](#)
- [22] A. Virdis, G. Stea and G. Nardini, "SimuLTE - A modular system level simulator for LTE/LTE-A networks based on OMNeT++," in *Proc. of International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pp. 59–70, Aug. 2014. [Article \(CrossRef Link\)](#)
- [23] N. Baldo, M. Miozzo, M. Requena-Esteso and J. Nin-Guerrero, "An open source product-oriented LTE network simulator based on ns-3," in *Proc. of ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 293–298, Nov. 2011. [Article \(CrossRef Link\)](#)
- [24] E. Soltanmohammadi, K. Ghavami and M. Naraghi-Pour, "A survey of traffic issues in machine-to-machine communications over LTE," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 865–884, Feb. 2016. [Article \(CrossRef Link\)](#)
- [25] M. Centenaro and L. Vangelista, "A study on M2M traffic and its impact on cellular networks," in *Proc. of IEEE World Forum on Internet of Things (WF-IoT)*, pp. 154–159, Dec. 2015. [Article \(CrossRef Link\)](#)
- [26] L. Wang, F. Fang, N. Nikaein and L. Cottatellucci, "An analytical framework for multi-layer partial frequency reuse scheme design in mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7593–7605, Nov. 2015. [Article \(CrossRef Link\)](#)
- [27] L. Wang, Y. Wang, W. Chen, C. Kai and L. Wu, "Dynamic interference coordination with analytical near-optimum of power allocation toward high user fairness," *China Commun.*, vol. 13, no. 12, pp. 37–48, Dec. 2016. [Article \(CrossRef Link\)](#)
- [28] N. Osman, Y. Wang, A. Niklas, B. Nadia, A. Shehzad and S. Joachim, "Analysis of ultra-reliable and low-latency 5G communication for a factory automation use case," in *Proc. of ICC Workshop*, pp. 441–446, June 2015. [Article \(CrossRef Link\)](#)
- [29] Technical Specification, "Scenarios, requirements and KPIs for 5G mobile and wireless system," METIS project D1.1. [Article \(CrossRef Link\)](#)
- [30] I. Chih-Lin, S. Han, Y. Chen and G. Li, "Trillions of nodes for 5G!?" in *Proc. of IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 246–250, Oct. 2014. [Article \(CrossRef Link\)](#)
- [31] Y. Yuan and L. Zhu, "Application scenarios and enabling technologies of 5G," *China Commun.*, vol. 11, no. 11, pp. 69–79, Nov. 2014. [Article \(CrossRef Link\)](#)
- [32] Technical Specification, "Components of a new air interface-building blocks and performance," METIS project D2.3. [Article \(CrossRef Link\)](#)
- [33] 5G Americas, "5G channel model for bands up to 100 GHz," White Paper, Feb. 2016. [Article \(CrossRef Link\)](#)
- [34] M. A. Khan, H. Hasbullah and B. Nazir, "Recent open source wireless sensor network supporting simulators: a performance comparison," in *Proc. of International conference on Computer, Communications, and Control Technology*, pp. 324–328, Sept. 2014. [Article \(CrossRef Link\)](#)



Lusheng Wang received his B.Sc. in Communications Engineering in 2004 from Beijing University of Posts and Telecommunications (BUPT), China and his Ph.D. in 2010 in Computer Science and Networks from Telecom ParisTech (ENST), France. He worked as a Post-Doc during 2010 at INSA-Lyon and a Post-doc during 2011–2012 at Eurecom, France. Currently, he is a research professor and the vice-dean of Communications Engineering Department at Hefei University of Technology (HFUT), China. His research interests are resource and interference management in hyper-dense and heterogeneous networks. He has published about 40 refereed international journals and conference papers.



Kun Chang received the B.Sc. degree in communication engineering from Huaibei Normal University, Anhui, China in 2015. He is currently working toward the M.Sc. degree in communication and information system at HFUT, Hefei, China. His research interest is on wireless network simulator in 5G hyper-dense networks.



Xiumin Wang is currently an associate professor at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. She received her B.Sc. from the Department of Computer Science, Anhui Normal University, China, in 2006. She received her Ph.D. from both the Department of Computer Science at University of Science and Technology of China and the Department of Computer Science at City University of Hong Kong, under a joint Ph.D. program. Her research interests include mobile computing, wireless networks, and network coding.



Zhen Wei received his B.Sc. from Wireless Communication Engineering Department at Anhui University in 1985, M.Sc. in 1997 and Ph.D. in 2005 both in Computer Application Technology from HFUT. He is currently a full professor at School of Computer and Information, HFUT, and the president of HFUT High-Tech (also known as GOCOM). His research interests cover many fields, including wireless communications, industrial safety and distributed control, intelligent railway signaling and control, mining track transportation, etc. He has published over 80 journals and conference papers, and holds over 50 patents and 16 software copyrights.



Qingxin Hu received his B.Sc. from in Electronic Information Engineering in 1985 and his M.Sc. in Signal and Information Processing in 1991, both from HFUT, where he is currently an associate professor. Meanwhile, he is also a senior vice-president at HFUT High-Tech and the dean of the research institute of industrial safety technologies. His current research interest is mainly information processing and wireless communication support for intelligent railway signaling control.



Caihong Kai received her master's degree in Electronic Engineering and Computer Science from University of Science and Technology of China in 2006 and her Ph.D. in Information Engineering from The Chinese University of Hong Kong in 2010, respectively. She is now an associate professor at HFUT. Her research interests are in wireless communication and networking, network protocols and performance evaluation.