IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 26, 2020, accepted March 17, 2020, date of publication March 19, 2020, date of current version March 27, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2982120*

# Cooperative and Energy-Efficient Strategies in Emergency Navigation Using Edge Computing

**HUIBO BI, (Member, IEEE), WEN-LONG SHANG, AND YANYAN CHEN**
Beijing Key Laboratory of Traffic Engineering, College of Metropolitan Transportation, Beijing University of Technology, Beijing 100124, China

Corresponding author: Wen-Long Shang (shangwl_imperial@bjut.edu.cn)

This work was supported in part by the National Key Research and Development Program for the 13th-Five-Year Plan of China under Grant 2018YFF0300305 and Grant 2018YFF0300300.

**ABSTRACT** Nowadays, public transportation junctions (PTJs) such as metro stations and railway stations can involve up to half of urban traffic and account for a large portion of energy usage in urban areas. The tremendous traffic and energy flows in PTJs have induced severe problems in safety and efficiency. Thus, in this paper, we present a comprehensive edge computing enabled navigation framework to aid the emergent evacuation processes within PTJs with respect to energy and safety restrictions. From the energy-saving aspect, due to the resource restraints of the on-site Internet of Things (IoT) based environmental monitoring system, a queueing network model is utilised to balance the energy utilisation and reduce the congestion of the sensing and navigation process during emergency. From the safety aspect, three edge computing aided cooperative strategies are proposed to dynamically assign evacuees into several groups to adapt their course of action with regard to their physical conditions and immediate environments. Simulation results show that the use of the queueing network model can reduce and balance the energy utilisation of the on-site IoT network. Experiments also show that the use of cooperative strategies to adjust the evacuees' category and the associated routing algorithm can achieve higher survival rates.

**INDEX TERMS** Cognitive packet network, cooperative strategies, edge computing, emergency navigation, energy-efficiency, G-network.

## I. INTRODUCTION

With the rapid progress in urbanisation, informatisation, as well as the large-scale construction of transportation infrastructures [1], public transportation junctions are gradually becoming the nexus of energy flows, information flows and traffic flows. The necessity of energy flows to support the electrified transportation systems and the concentration of traffic flows have brought new challenges to emergency navigation (also known as evacuation path planning) in both system optimisation and evacuee behaviour management perspectives.

From the system optimisation aspect, the advancement in sensing, computing, information and communications technologies has greatly improved the efficiency of emergency navigation systems. However, the huge amount of generated sensory data could also aggravate the burden on real-time computation and has led to the continuous evolution in computing paradigms, including desktop computing, client/server computing, networks computing, cluster computing, grid computing and eventually cloud computing [2]. In the recent years, the prosperity of Internet of Things (IoT) and the rich cloud services has spawned a new computing paradigm—edge computing. Compared to cloud computing which pulls all the computations to the remote cloud servers, edge computing is able to perform computation [3], caching [4], data storage, relay and resource allocation [5] at the edge of the network, which is nearer to IoT data sources [6]. Thus, edge computing has become a promising technology since it can efficiently shorten task response time, reduce network burden, prolong IoT battery life time, and improve data safety and privacy [7]. In this paper, we take advantage of these features, and propose an edge computing enabled system framework to reduce the energy usage of an emergency navigation system.

From the evacuee behaviour management aspect, congestion or stampede of passengers in transportation junctions during disasters or huge public events, can be a significant

The associate editor coordinating the review of this manuscript and approving it for publication was Sanaa Sharafeddine.

threat to public safety. Destructive crowd behaviours such as clogging, pushing and trampling may block evacuees and induce unnecessary fatalities and injuries. Recent studies related to crowd behaviour modelling indicate that cooperative collective behaviours can mitigate the aforementioned harmful behaviours and benefit an evacuation process. For instance, by employing a cellular automata model to mimic grouping behaviours, the work in [8] shows that the evacuation time decreases with the increasing numbers of groups. Similarly, the work in [9] employs a cellular automata model to investigate the emergence of cooperative behaviours during emergency with an evolutionary game approach; the experiments indicate that the cooperation level of evacuees and evacuation efficiency will grow with the increase in escape aspiration level. However, although various cellular automata and agent based models have been designed to mimic the empirical collective behaviours of human beings in emergency, little work has been conducted to apply some of these ideas to the field of emergency navigation, which aims to compute proper paths.

Most of the previous emergency navigation algorithms [10] use a single metric to select paths for all evacuees without considering their characteristics in terms of age, mobility, level of resistance to hazard etc., and therefore, may require diverse services during an emergency. For instance, evacuees such as sick people or aged people need to choose the safest paths that will be well ahead of the spreading hazard while for others may prefer the quickest paths without hazard. Although our previous work [11], [12] have proposed a routing algorithm to guide diverse evacuees with respect to their requirements and capabilities, this algorithm sticks to a single routing algorithm during an entire evacuation process and are insensitive to sudden changes in the hazard environment such as abrupt congestion or injury of civilians. The subsequent work in [13] presents a health-aware dynamic grouping strategy and suggests that the adaptation of groups as well as the associated routing metric with respect to the on-going health conditions and mobility of evacuees can improve the efficiency of an evacuation process. In this paper, we extend the previous work and develop three dynamic grouping mechanisms to adjust the behaviours of evacuees in terms of their on-going physical condition and surrounding environment. We borrow the concept of the cognitive packet networks (CPN) [14], [15] to customise escape paths for the diverse evacuees in a building based on the time-oriented and safety-oriented metrics in [12]. Since cognitive packets play a vital role in route discovery and information collection of the CPN framework [16] and their behaviours contribute significantly to the efficiency of path-finding as well as the energy usage. A G-network model [17] based optimisation algorithm, which is deployed in the off-site edge computing center, is utilised to manage the dropping probability of cognitive packets when traffic is under overload conditions. Specifically, the main contributions of this paper are summarized as follows.

- We present an edge computing enabled queueing network model for the energy and delay optimisation in a given IoT environment which is utilised to provide emergency navigation services for civilians. This comprehensive model takes the packet-routing behaviours of the IoT network as well as interactions between the IoT and the back-end edge computing center into considerations. The model gradually optimises the IoT network by dynamically tuning the dropping probability of packets at each node of the IoT network. Since the tuning process is computationally-intensive, it is offloaded to the remote edge computing center. Hence, the on-site IoT network can be energy and latency efficient by only updating the dropping probability periodically.
- We present three edge computing enabled cooperative strategies to aid the on-site path-finding algorithm to optimise egress paths for evacuees, which is innovative in the research field of emergency management. The first strategy enhances the "sensitivity" of the path-finding algorithm by shifting path type of evacuees with respect to their physical attributes. The second strategy enhances the "congestion-adaptation" of the the path-finding algorithm by distributing evacuees to alternative paths when sudden congestion occurs. The third strategy enhances "resilience" of the path-finding algorithm to the initial distribution of evacuees by assigning them to specific staircases and exits. It is shown that these cooperative strategies can significantly improve the survival rate of the evacuees.

The remainder of this paper is organised as follows. In the next section we review the literature relevant to our work. Section III presents the fundamental theories underlying our research problem and formulates the system model. In Section IV, the optimal packet-dropping probability is obtained by minimising the proposed compound cost function via a gradient descent algorithm. Next, we propose several cooperative strategies for crowd movement optimisation in Section V. The simulation models and assumptions are then described in Section VI. The experimental results and discussions are presented in Section VII. Finally, we draw conclusions in Section VIII.

## II. LITERATURE REVIEW

In accordance with the aforementioned contributions of this paper, we review literature from both system optimisation and emergency cooperative strategies aspects.

### A. SYSTEM OPTIMISATION IN EMERGENCY NAVIGATION

Due to their long term monitoring ability, wireless sensor network (WSNs) have been involved in a number of security related applications such as emergency navigation, intrusion detection, medical care and criminal hunting. Since WSNs are generally formed by battery-powered devices, various protocols are designed to improve the energy efficiency in information acquisition and transmission. Many routing protocols that are specifically designed for WSNs have been presented due to power constraints and unattended operations in life time of WSNs. Data delivery models play a vital role in WSNs related protocols and can be divided into continuous models, event-driven models and query-driven models [18]. In continuous models, sensor nodes transmit data periodically while in the latter two models data are sent when a trigger event is generated. This review focuses on literature related to continuous models in which data rate or packet life time has a significant impact on reducing traffic or saving energy. By taking imperfect channel conditions into account, the work in [19] presents an adaptive rate selection algorithm to adjust the throughput to the closest nominal data rate options; each sensor node chooses the highest data rate based on the pilot signal-to-noise ratio and improved overall throughput are observed in simulations. The study in [20] considers the Quality of Service (QoS) of video or imaging sensors and categorises data packets into real-time traffic and non-real-time traffic; each node maintains one queue for each type of traffic and a queueing model [21] is employed to estimate the delay; the bandwidth ratio between two types of traffic is dynamically adapted to satisfy the time latency requirements for both categories of data. By separating data packets into routine packets and unusual packets, the research in [22] presents a Randomized Re-routing (RRR) mechanism to divert routine data to secondary routes and keep the high QoS paths for unusual data; the occurrence of unusual events is determined by monitoring the significant changes in traffic flows; the experiments which are performed in the ns-2 network simulator show that this algorithm provides remarkably improvements on the QoS of unusual data while fulfills the QoS requirements of routine data. A Brownian motion model [23], [24] is presented in [25] to evaluate the effect of time-out to the average packet travel time in a wireless network where packets search the destination without the aid of routing table; common phenomena such as packet loss are considered and the results indicate that a judicious choice can effectively minimise the average packet travel time. The major drawback of the WSN based emergency navigation systems is the limited computing and battery capacity. Hence, some emergency navigation systems have integrated cloud computing technologies that are accessible via on-site WSN, to offload intensive computations to remote cloud servers. For instance, the research in [26] proposes a route recommendation system to provide safe and less-congested paths for vehicles and evacuees during a disaster; the system is composed of a front-end intelligent transportation system (ITS) and a back-end cloud-based parallel computing cluster; the ITS is responsible for real-time traffic load monitoring while the cloud computing cluster is utilised to compute the routes with the shortest potential time to destination; when a road segment is occupied by emergency vehicles, the cloud cluster will provide secondary routes for evacuee vehicles to avoid possible congestion. With the recent prosperity of IoT and mobile computing techniques, in [27], an evacuation path planning system is designed to calculate safety-oriented evacuation paths for civilians to appropriate shelters with a relationship-sensitive artificial potential field algorithm; evacuees with relationships can be guided to the same shelter; the path planning system consists of an IoT enabled software application to gather location data of evacuees and a cloud service center to perform route planning.

### B. COOPERATIVE STRATEGIES IN EMERGENCY NAVIGATION

Crowd management issues, which are likely to arise in congested sites of modern urbanised societies, has aroused a new interest in recent decades due to their interactive effect on evacuations. Destructive crowd behaviours such as stampede, which can prolong the clearance time of an evacuation process, may lead to serious fatalities [28]. Hence, in order to optimise the design of crowded sites and estimate the evacuation time, much work has been dedicated to investigate and design crowd behaviour models such as cellular automata models [8], social force models [29], fluid-dynamic models [30] and agent-based models [31] to simulate the crowd movements in reality and prevent destructive crowd behaviours from occurring by improving the design of built environments, which are also known as off-line emergency navigation algorithms [32]. Although much research on multi-agent systems [33], robotic systems [34] and autonomous systems [35], [36] has shown the benefits of cooperative grouping behaviours in improving the system performance, little work has been done in the field of real-time emergency navigation. The work in [37], [38] present a resilient emergency support system (ESS) to disseminate emergency messages among evacuees with the aid of opportunistic communication; evacuees choose the shortest paths to exits and exchange hazard information to evacuees in proximity; the impact of ‘‘passive’’ cooperation of information-sharing among evacuees is evaluated with different communication ranges. The research in [39] proposes an infrastructure-less emergency navigation system to guide evacuees with the aid of smart handsets and cloud servers; to balance the remaining battery power among portable devices, evacuees are organised in loose groups to increase the possibility of multi-hop relaying. The study in [40] presents an emergency rescue evacuation support system (ERESS) with specific mobile terminals that can maintain communication with the aid of a mobile ad hoc network during disasters; an ERESS holder and the surrounding non-holders can form a group and follow the evacuation route with the shortest time to exit; the moving speed of a group is predicted based on the number of evacuees in the group. Since the evacuation efficiency can be considerably affected by the actions of evacuee leaders, the work in [41] designs a leader-following strategy to evacuate evacuees in an urban railway transit station; a bi-level optimisation model is utilised to optimise the number and initial locations of leaders, as well as the their routes with a variant of A* algorithm and a multi-leader coordination mechanism. The research in [42] proposes a dynamic grouping mechanism to improve the evacuation efficiency during building evacuation, evacuees are grouped with respect to the

**TABLE 1. Comparison with existing relevant contributions in emergency management. EBM: evacuee behaviour modelling, EPP: evacuation path planning, WSN: wireless sensor networks, PD: Portable devices.**

| References | Focus | Infrastructure Type | Modelling Approach | Multiple User Type Supporting | Cooperative Strategies |
|---|---|---|---|---|---|
| Helbing et al.(2002) [30] | EBM | Not applied | Fluid-dynamic | × | × |
| Müller et al.(2014) [8] | EBM | Not applied | Cellular automata | ✓ | × |
| Cheng et al.(2018) [9] | EBM | Not applied | Cellular automata | ✓ | ✓ |
| Rozo et al.(2019) [31] | EBM | Not applied | Multi-agent | ✓ | × |
| Jiang et al.(2020) [29] | EBM | Not applied | Social force | × | × |
| Gelenbe et al.(2012) [37] | EPP | WSN + PD | Multi-agent | × | × |
| Bi et al.(2014) [11] | EPP | WSN + PD | Multi-agent | ✓ | × |
| Akinwande et al.(2015) [13] | EPP | WSN + PD | Multi-agent | ✓ | ✓ |
| Khalid et al.(2016) [26] | EPP | ITS + Cloud computing | Unspecified | × | × |
| Xu et al.(2018) [27] | EPP | IoT + Mobile cloud computing | Prototype system | × | × |
| Liu et al.(2018) [42] | EPP | Unspecified | Social force | × | ✓ |
| Zhou et al.(2019) [41] | EPP | WSN + PD | Social force | × | ✓ |
| Our Proposed work | EPP | IoT + Edge computing | Multi-agent | ✓ | ✓ |

selected exits and the distance to other evacuees; an artificial bee colony algorithm is employed to calculate desired paths for each group.

In conclusion, as shown in Table 1, most of the previous emergency management research that has considered heterogeneous evacuee categories or cooperative strategies concentrates on designing proper crowd behaviour models to mimic real emergency situations rather than designing navigation algorithms to provide appropriate evacuation routes. Although several emergency navigation algorithms have introduced cooperative strategies, these cooperative strategies are limited to provide resilient communications, leader-following or grouping adjacent evacuees with respect to their desired destinations. Additionally, to the best of our knowledge, few cloud or edge computing aided emergency navigation systems have considered the energy optimisation problem of the on-site WSN or IoT network.

## III. PROBLEM FORMULATION AND SYSTEM MODEL

This section firstly presents the descriptions and assumptions of the investigated fire-related evacuation navigation problem, which focuses on developing navigation algorithms and cooperative strategies to direct evacuees out of a confined built environment safely and efficiently when a disaster occurs. Secondly, based on the realistic evacuation scenario, a bi-level system approximation model is abstracted and formulated based on a lower level CPN model and a upper level G-network model, respectively. The CPN model is utilised to search appropriate evacuation paths for evacuees while the G-network model is employed to balance the energy consumption and reduce the packet network congestion of the IoT network. A list of notations used in this section is summarised in Table 2.

### A. PROBLEM DESCRIPTIONS AND ASSUMPTIONS

To provide appropriate egress routes for evacuees, we assume that an edge computing enabled two-tier emergency response system is deployed in a transportation junction as shown in Figure 1. The system is composed of an on-site IoT network and an off-site edge computing center as illustrated in Figure 2. The IoT network is utilised to search paths for evacuees via performing CPN operations while the edge computing center is employed to conduct intensive computations including the algorithms in Section IV and V. The IoT network is pre-deployed in the built environment and consists of edge nodes (ENs) and sensor nodes (SNs). ENs, which employ the edge-computing concept and function as CPN nodes, emit SPs to search and measure paths to exits and

**TABLE 2. List of notations used in Section III.**

| Notation | Definition |
|---|---|
| $k$ | Represent a type of packets (including SPs and ACKs) or the associated category of evacuees |
| $K$ | Represent the number of types of packets |
| $N$ | Represent the number of ENs in the network |
| $n_i$ or $n_j$ | Equally represent an EN |
| $\Lambda_{n_i,k}$ | Represent the birth rate of packets of type $k$ at EN $n_i$ |
| $P(n_i, n_j, k)$ | Represent the probability that a packet of type $k$ to be relayed to EN $n_j$ from EN $n_i$ |
| $P^e_{d_{n_i},k}$ | Represent the probability that a packet of type $k$ to be dropped at $n_i$ at a given time. It equals either 1 or 0 since all packets of type $k$ only travel over a bench of deterministic paths at a given time. |
| $Q(n_i, n_j, k)$ | Represent the probability that a packet of type $k$ to be rerouted to EN $n_j$ |
| $q_{n_i,k}$ | Represent the probability that EN $n_i$ has one or more packets of type $k$ |
| $\lambda^+_{n_i,k}$ | Represent the total arrival rate of packets of type $k$ to EN $n_i$ |
| $r_{n_i,k}$ | Represent the service rate of EN $n_i$ to packets of type $k$ |
| $\lambda^-_{n_i,k}$ | Represent the arrival rate of re-routing decisions of packet class $k$ from the edge computing center |
| $q_{n_i}$ | Represent the probability that EN $n_i$ has one or more packets |
| $N_{n_i}$ | Represent the average number of packets at EN $n_i$ |
| $\lambda^s_{n_i}$ | Represent the arrival rate of SPs at EN $n_i$ |
| $\lambda^a_{n_i}$ | Represent the arrival rate of ACKs at EN $n_i$ |
| $P_{d_{n_i},k}$ | Represent the probability that a SP of type $k$ to be dropped at $n_i$ |
| $Q_s(n_i, n_j, k)$ | Represent the probabilistic choice of a SP of type $k$ at EN $n_i$ |
| $P_d$ | Represent the drift parameter of CPN |
| $N^a_n$ | Represent the number of neighbouring ENs of a given EN |
| $n_r$ | Represent the RNN's advice of an EN |
| $n_p$ | Represent the previous hop of a SP before arriving the current EN |

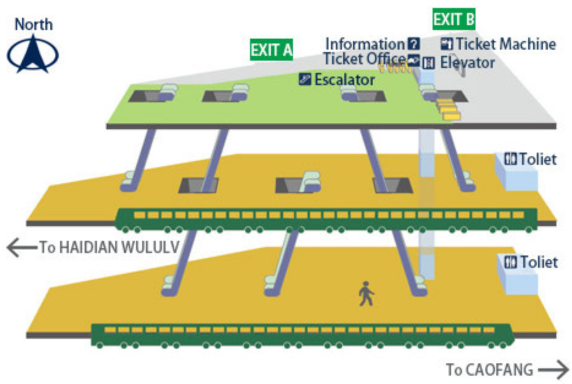


**FIGURE 1. A schematic diagram illustrating the layout of the Nanluoguxiang transportation junction.**

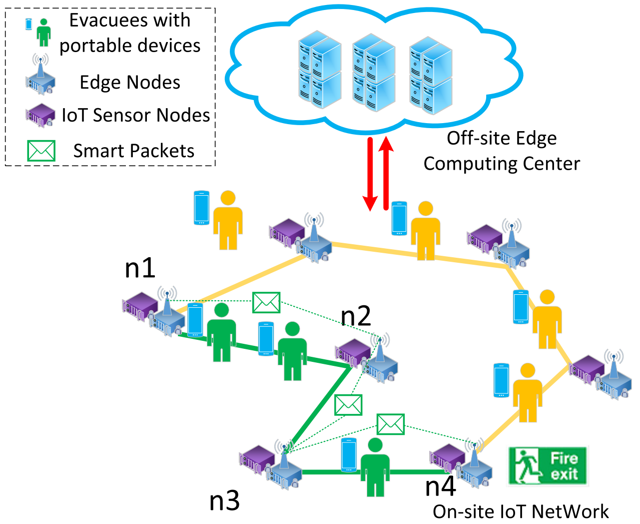


**FIGURE 2.** **The framework of the proposed edge computing enabled emergency navigation system.**

provide appropriate paths for evacuees in proximity. Hence, ENs form a CPN and act in a cooperative manner via learning others' experience, although each EN possesses limited computing power, they together can solve desired paths rapidly and adaptively. Real-time decisions can be sent to evacuees via transmitting suggested paths to portable devices carried by the evacuees. ENs also work as relay nodes to exchange information among evacuees and the remote edge computing center, while providing distributed and lightweight computing services. SNs are employed to monitor the situation of the surrounding area and gather hazard information. To calculate the desired paths for evacuees, our algorithm requires the estimated number of evacuees nearby and the severity level of the hazard as input parameters. To count the number of evacuees in the monitored area, various sensors such as WI-FI probe [43] can be used to sniff the presence of mobile devices carried by evacuees via capturing the features of the mobile devices such as MAC address and strength of the signal. To detect the severity level of the hazard, visual sensors such as camera can be used as a general way to detect the hazard type and severity level. For a specific hazard such as a fire hazard, temperature sensors can also be implementable. ENs and SNs are deployed at the significant locations such as junctions of two or more paths.

We have divided evacuees into two categories and customise one specific type of paths for each category of evacuees under the CPN framework. Class 1 evacuees represent prime-aged evacuees with high mobility and physical strength. Class 2 evacuees represent evacuees with low mobility and weaker physical conditions, such as children, aged people or sick people. As shown in Figure 2, the CPN framework can distribute packets including smart packets (SPs) and acknowledgements (ACKs) among ENs to search paths for evacuees. Each class of evacuees is associated with a type of packets. We have used type 1 packets with time-oriented routing metric to search paths with the shortest time to exit and type 2 packets with safety-oriented metric to search safest-shortest paths for Class 1 and Class 2 evacuees, respectively.

### B. MODEL FORMULATION

To direct evacuees out of the hazardous areas efficiently, we utilise an adaptation of the widely tested network routing protocol, namely the Cognitive Packet Network [44], [45]. The routing operations of CPN and its variations for Evacuee Routing Problems can be found in [12]. The time metric and safety metric we used to search paths for evacuees are derived from the previous work [12]. The routing metrics are the quality of service (QoS) goals that are pursued by SPs and optimised by the random neural networks (RNN) algorithm [46]. When a SP reaches an exit, it will be converted into an ACK and bring back the discovered path as well as all the collected sensory data. After an ACK backtrack to an EN, collected information will be extracted and then measured by routing metrics, the result will be used as the input of RNNs.

On the other hand, since the IoT network suffers from limited battery power, the battery of the ENs, especially the nodes on the backbone, may be drained rapidly due to excessive communication and computation demand. Hence, in this paper, we propose a packet discard mechanism to "drop" certain SPs to reduce the burden of the network. Instead of continuing to search the network, a "dropped" SP is abandoned at the EN, which can introduce two benefits to the CPN: (1) it can increase the "sensitivity" of the CPN by avoiding the RNN from over-training; (2) it can reduce the energy consumption and network congestion caused by the redundant packets. A G-network model [47] with additional control capabilities (e.g. positive customers, triggers and negative customers) is utilised to capture the dynamics of this behaviour and reduce the energy utilisation as well as congestion of the IoT network. In our treatment, positive customers and triggers represent network packets (SPs and ACKs) and re-routing decisions of packets, respectively. Negative customers are not employed since they have no practical interpretation in the model. Since the CPN-based emergency navigation algorithm is a multi-path routing algorithm, we introduce several types of SPs to search paths for the associated categories of evacuees with respect to the pre-defined goal functions. When a fire breaks out, ENs start to generate and emit SPs of type $k$ at the rate (known as external arrival rate in the G-network model) $\Lambda_{n_i,k}$, and also relay SPs and ACKs from other ENs. A packet of type $k$ that is departing an EN will either be relayed to another linked EN $n_j$ with probability $P(n_i, n_j, k)$ or be dropped at $n_i$ with probability $P^e_{d_{n_i,k}}$, where $P^e_{d_{n_i,k}} + \sum_{j=1}^{N}[P(n_i, n_j, k)] = 1$. Term $P^e_{d_{n_i,k}}$ and $P(n_i, n_j, k)$ equals either 1 or 0 since all packets of a specific type only travel over a bench of deterministic paths at a given time. For instance, for a given path $\pi(n_1, n_2, n_3, n_4)$ from EN $n_1$ to EN $n_4$ for packets of type $k$ in Figure 2, we can describe it by setting $P(n_1, n_2, k) = 1$, $P(n_2, n_3, k) = 1$, $P(n_3, n_4, k) = 1$. Since EN $n_4$ is an exit, term $P^e_{d_{n_4,k}}$ is also set to 1 to describe the fact that EN $n_4$ is the destination for all the packets traversing along the path. Term $N$ is the number of ENs in the network. Additionally, in the context of CPN, a departing packet of class $k$ heads to a connected EN $n_j$ with the probability $Q(n_i, n_j, k)$ under the impact of re-routing decisions from the edge computing center or the local RNN, where $\sum_{j=1}^{N} Q(n_i, n_j, k) = 1$. Under these assumptions, the probability that an EN $n_i$ has one or more packets of class $k$ can be derived from [17].

$$q_{n_i,k} = \frac{\lambda^+_{n_i,k}}{r_{n_i,k} + \lambda^-_{n_i,k}} \tag{1}$$

where $\lambda^+_{n_i,k}$ is the total arrival rate of packet class $k$ to EN $n_i$, including the SPs that are initially generated at $n_i$ and the SPs and ACKs that are arrived from other ENs. Term $r_{n_i,k}$ represents the service rate of the EN $n_i$ to the packet class $k$. We assume that packets of different classes are processed in a first-come-first-served order. Term $\lambda^-_{n_i,k}$ is the arrival rate of re-routing decisions of packet class $k$, which is equivalent to the arrival rate of control signals from the edge computing center in our treatment. Because $q_{n_i,k}$ is related to all the other $q_{n_j,k}$ in the network and also (1) is a nonlinear equation, therefore we can solve it numerically.

The total arrival rate of packet class $k$ to EN $n_i$ is given by:

$$\lambda^+_{n_i,k} = \Lambda_{n_i,k} + \sum_{j=1}^{N} q_{n_j,k}[r_{n_j,k}P(n_j, n_i, k) + \lambda^-_{n_j,k}Q(n_j, n_i, k)] \tag{2}$$

The probability that EN $n_i$ has one or more packets is given by:

$$q_{n_i} = \sum_{k \in K} q_{n_i,k} \tag{3}$$

where term $K$ stands for the number of types of SPs. This probability can be derived under the assumption that the average service rates $r_{n_i,k}$ of different types of packets at a given EN are equal. This is reasonable because the packets are served at the same EN and only the routing metrics are different for different types of packets. Since the packets are processed in a first-come-first-served order, and the average arrival rate of re-routing decisions $\lambda^-_{n_i,k}$ are also equal, then we have $q_{n_i} = \frac{\sum_{k\in K} \lambda^+_{n_i,k}}{r_{n_i,k}+\lambda^-_{n_i,k}} = \sum_{k\in K} q_{n_i,k}$.

Hence, the average number of packets at each EN $n_i$ is given by:

$$N_{n_i} = \frac{q_{n_i}}{1 - q_{n_i}} \tag{4}$$

The probabilistic choice $Q(n_j, n_i, k)$ of a packet of class $k$ sends from EN $n_j$ to EN $n_i$ can be calculated by:

$$Q(n_j, n_i, k) = \frac{1}{\lambda^s_{n_j} + \lambda^a_{n_j}}\big[\lambda^s_{n_j} Q_s(n_j, n_i, k)(1 - P_{d_{n_j,k}}) + \lambda^a_{n_j} Q_s(n_i, n_j, k)\big] \tag{5}$$

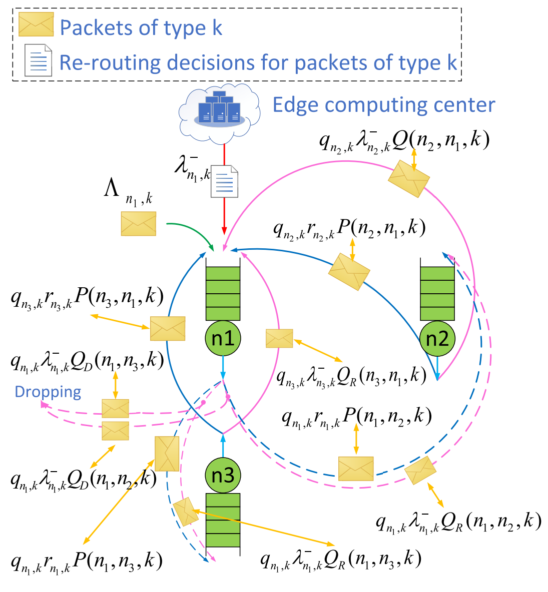


**FIGURE 3. G-network model representation of packet flows of type $k$ at EN $n_1$ in the proposed system shown in Figure 2. Positive customers and triggers in the G-network model represent network packets (SPs and ACKs) and re-routing decisions of packets, respectively. Term $Q_R(n_j, n_i, k) = \frac{1}{\lambda^s_{n_j}+\lambda^a_{n_j}}[\lambda^s_{n_j} Q_s(n_j, n_i, k) + \lambda^a_{n_j} Q_s(n_i, n_j, k)]$ represents the probability for a packet of type $k$ to traverse from EN $n_j$ to EN $n_i$. Term $Q_D(n_j, n_i, k) = \frac{1}{\lambda^s_{n_j}+\lambda^a_{n_j}}[\lambda^s_{n_j} Q_s(n_j, n_i, k)P_{d_{n_j,k}}]$ represents the probability for a packet of type $k$ that tends to traverse from EN $n_j$ to EN $n_i$ to be dropped at EN $n_j$.**

where term $\lambda^s_{n_j}$ represents the arrival rate of SPs at EN $n_j$, and term $\lambda^a_{n_j}$ represents the arrival rate of ACKs at EN $n_j$. These two values can be measured by the EN. Term $P_{d_{n_j,k}}$ represent the probability that a SP of type $k$ to be dropped at $n_j$ to relieve the network burden. Term $Q_s(n_j, n_i, k)$ stands for the probabilistic choice of a SP of class $k$ at EN $n_j$ towards EN $n_j$, which is determined by the "drift" parameter of the CPN as shown in (6). The equation above is derived based on the assumption that only SPs can be dropped. This is because it is relatively costly to generate ACKs which carry discovered route and environmental information. Besides, dropping SPs can naturally reduce the number of ACKs since only the SPs that reach the destination can generate ACKs. A schematic diagram of the G-network model based packet flow representation of EN $n_1$ is shown in Figure 3.

The "drift" parameter is defined as an SP's probability to choose the next hop at random over the RNN's advice.

$$Q_s(n_i, n_j, k) = \begin{cases} 1 - P_d + \dfrac{P_d}{N^a_n - 1} & \text{if } n_j = n_r \\ \dfrac{P_d}{N^a_n - 1} & \text{if } n_j \neq n_r \ or \ n_p \\ 0 & \text{if } n_j = n_p \end{cases} \tag{6}$$

where term $P_d$ stands for the drift parameter. Term $N^a_n$ represents the number of neighbouring nodes of EN $n_i$. Term $n_r$ is the RNN's advice on the EN $n_i$, and $n_p$ stands for the previous hop of this packet before reaching $n_i$.

**TABLE 3. List of notations used in Section IV.**

| Notation | Definition |
|---|---|
| $F_t$ | Represent the compound objective function to be optimised |
| $E_t$ | Represent the energy oriented sub-objective function |
| $D_t$ | Represent the network delay oriented sub-objective function |
| $R_t$ | Represent the path-finding ratio oriented sub-objective function |
| $E_n^{n_i}$ | Represent the current remaining battery power of EN $n_i$ |
| $\overline{E_n}$ | Represent the average remaining battery power of all ENs in the IoT network |
| $E_c^{n_i}$ | Represent the energy consumption during each dropping probability update interval |
| $t_{cu}$ | Represent the updating time interval of the dropping probability |
| $E_s$ | Represent the static power consumption of an EN |
| $P^s_{n_i}$ | Represent the empirical probability for a SP that traverses through EN $n_i$ to discover an egress path |

## IV. GRADIENT OPTIMISATION FOR PACKET FLOWS

To balance the energy consumption and reduce congestion in the IoT network, we propose a gradient based algorithm to select the appropriate dropping probability $P_{d_{n_i,k}}$, a list of notations used in this section is shown in Table 3. To achieve energy and delay efficiency while maintaining an acceptable level of path-finding probability, we define a compound objective function that contains energy balancing $E_t$, network delay $D_t$ and the path-finding ratio $R_t$. This path-finding ratio $R_t$ of an EN is simply defined as the number of traversed SPs (including SPs generated at the EN and SPs that passed through the EN) that have successfully discovered a path over the number of SPs generated at this EN.

$$F_t = E_t + \epsilon D_t + \frac{\psi}{R_t} \tag{7}$$

where $\epsilon$ and $\psi$ are constants that determine the relative importance of the three metrics. The above objective function is derived from the G-network representation of the network, and is optimised periodically during the routing operations of the CPN by using a gradient descent algorithm that performs on the edge computing center.

The energy balancing function can be expressed as:

$$E_t = \sum_{i=1}^{N}(E_n^{n_i} - E_c^{n_i} - \overline{E_n})^2 \tag{8}$$

where term $E_n^{n_i}$ represents the current remaining battery power of EN $n_i$ and $\overline{E_n}$ is the mean remaining battery power of all ENs in the network. Term $E_c^{n_i}$ represents the energy consumption during each dropping probability update interval $t_{cu}$ during which the latest dropping probabilities $P_{d_{n_i,k}}$ of ENs are determined and returned from the edge computing center.

$$E_c^i = \Big\{E_s + \sum_{k=1}^{K}\big[e_l\lambda^+_{n_i,k} + e_b r_{n_i,k}B[q_{n_i,k} < 1] + e_b\lambda^+_{n_i,k}B[q_{n_i,k} \geq 1]\big]\Big\}t_{cu} \tag{9}$$

where $E_s$ is the static power consumption of an EN and $e_b = e_c d^a + e_t$, the detailed energy utilisation model of ENs can be found in Table 5. $B[X]$ is a function that takes the value zero if $X$ is true or 1 if $X$ is false, respectively. If $q_{n_i,k} < 1$, it means packets of class $k$ is not buffered at this EN. Hence, the throughput is determined by the packet arrival rate $\lambda^+_{n_i,k}$. Otherwise, the throughput is determined by the service rate $r_{n_i,k}$.

To estimate the average delay for a packet experienced in the network, we consider the network as a whole and utilise the Little's formula.

$$D_t = \frac{\sum_{i=1}^{N} N_{n_i}}{\sum_{i=1}^{N} \Lambda_{n_i}} \tag{10}$$

where the numerator is the total average number of packets in the network, and denominator is the total rate at which packets (SPs) join the network.

On the other hand, the path-finding ratio $R_t$ can be expressed as:

$$R_t = \frac{\sum_{i=1}^{N} \lambda^s_{n_i} N_{n_i} P^s_{n_i}}{\sum_{i=1}^{N} \Lambda_{n_i}(\lambda^s_{n_i} + \lambda^a_{n_i})} \tag{11}$$

where $P^s_{n_i}$ is the empirical probability for a SP that traverses through EN $n_i$ to discover an egress path.

By substituting (8), (10) and (11) with (2), (3), (5) and into (7), we can express (7) as a function of $q_{n_i,k}$:

$$F_t(q_{n_i,k}) = \sum_{i=1}^{N}(E_n^{n_i} - \sum_{k=1}^{K}\widetilde{E_c^{n_i}}(q_{n_i,k}) - \overline{E_n})^2 + \epsilon\sum_{i=1}^{N}\widetilde{N^a_{n_i}}(q_{n_i,k}) + \frac{\psi}{\sum_{i=1}^{N}\widetilde{N^b_{n_i}}(q_{n_i,k})} \tag{12}$$

where term $\widetilde{E_c^{n_i}}(q_{n_i,k}) = \big\{\frac{1}{K}E_s + \big[e_l\lambda^+_{n_i,k} + e_b r_{n_i,k}B[q_{n_i,k} < 1] + e_b\lambda^+_{n_i,k}B[q_{n_i,k} \geq 1]\big]\big\}t_{cu}$, term $\widetilde{N^a_{n_i}}(q_{n_i,k}) = \frac{\sum_{k=1}^{K} q_{n_i,k}}{(1-\sum_{k=1}^{K} q_{n_i,k})\sum_{i=1}^{N}\Lambda_{n_i}}$ and term $\widetilde{N^b_{n_i}}(q_{n_i,k}) = \frac{\lambda^s_{n_i}P^s_{n_i}\sum_{k=1}^{K} q_{n_i,k}}{(1-\sum_{k=1}^{K} q_{n_i,k})\sum_{i=1}^{N}\Lambda_{n_i}(\lambda^s_{n_i}+\lambda^a_{n_i})}$.

Since $q_{n_i,k}$ can be expressed by a function of $P_{d_{n_i,k}}$, by utilising the chain rule, taking the partial derivative of the above expression (12) with respect to a specific $d_{n_x,y} (x \in N, y \in K)$ yields:

$$\frac{\partial F_t}{\partial P_{d_{n_x,y}}} = \sum_{i=1}^{N}\sum_{k=1}^{K}\frac{\partial F_t}{\partial q_{n_i,k}}\frac{\partial q_{n_i,k}}{\partial P_{d_{n_x,y}}} \tag{13}$$

Finally, we can calculate $P_{d_{n_i,k}}$ by using the following iterations:

$$P^{n+1}_{d_{n_i,k}} = P^n_{d_{n_i,k}} - \eta\frac{\partial F_t}{\partial P^n_{d_{n_i,k}}} \tag{14}$$

where $\eta > 0$ is the learning rate.

## V. COOPERATIVE STRATEGIES FOR EVACUEE FLOW

Although much research has demonstrated that cooperative strategies can benefit cyber-physical systems, most studies focus on large-scale robotic systems [34]–[36] and few has been directed to the field of emergency navigation. Due to the complexity and diversity of hazard environments, an overall well-performed algorithm which normally integrates several metrics may be insensitive to sudden local changes such as abrupt congestion. Hence, in this section, we present three dynamic grouping mechanisms to assist the emergency navigation algorithms to optimise the crowd behaviour.

### A. HEALTH-AWARE GROUPING

To the best of our knowledge, most of the previous studies in emergency navigation persist on a single decision algorithm during the whole evacuation process and do not adjust in accordance with individuals' physical conditions and their immediate environments. Although we have proposed a health-aware grouping in previous research [13] to adapt injured evacuees to safer paths, we believe it is more beneficial for evacuees to be able to switch back and forth between groups during an evacuation when certain conditions are triggered. For instance, when an individual which belongs to the "Class 1 group" gets injured or get exhausted, it should be adapted to the safer "Class 2 group" due to the reduced mobility and injury. On the other hand, if its virtual health value has exceeded a certain threshold of the mean health value of the overall population, it should be switched back to the "Class 1 group".

This health-aware grouping mechanism enables evacuees to change groups and the associated algorithms rather than sticking to the pre-defined groups: any evacuee in the first category whose health level has dropped below a certain threshold of the mean health value will be immediately considered as a Class 2 evacuee. Similarly, any evacuee in the second category whose virtual health level has exceeded a certain threshold of the mean health value will be immediately considered as a Class 1 evacuee. The detailed process of the mechanism is shown in Pseudocode 1 and a list of symbols used is summarised in Table 4. The health value of an evacuee is affected by the exposure to the hazard and the traveled distance. In reality, it can be calculated by a portable device carried by evacuees. The impact of hazard can be evaluated by the hazard intensity of the adjacent sensor, which can be updated when reaching an EN. The mean health value is calculated by the back-end edge computing center in accordance with the uploaded individual health values.

### B. CONGESTION-AWARE GROUPING

The second mechanism is proposed based on the observation that certain vertices located in broad areas such as a hall can still have a long queue and the connected paths are not sufficiently used. These vertices are normally linked with "bottlenecks" such as staircases where stampede or continuous congestion may occur. Although both the time metric and safety metric take predicted congestion level of a route into consideration, the effect of other factors in the metrics such as "effective length" may induce evacuees at a vertex to choose an identical path. To exclude the influence of other factors, we consider the potential number of congestion encountered on a path as a secondary QoS metric and propose a congestion-alleviate mechanism to balance the main QoS requirement and the secondary QoS need. This congestion-ease policy makes use of the mailbox of CPN nodes and chooses a less congested path with acceptable main QoS value (safety level or egress time) rather than the top ranked path when the congestion level is high. The process of the mechanism is shown in Pseudocode 2 and the symbols used are also listed in Table 4.

**TABLE 4.** List of symbols used in the Pseudocode 1, 2 and 3.

| Notation | Definition |
|---|---|
| $G_{id}$ | Represent the group ID of an evacuee |
| $G_{one}$ | Represent the group ID of "Class 1" evacuees |
| $G_{two}$ | Represent the group ID of "Class 2" evacuees |
| $G_{con}$ | Represent the group ID of "congestion-ease" evacuees, which is a temporary group for congestion alleviation |
| $\overline{Y_c}$ | Represent the current mean virtual health value of the overall population |
| $Y_c$ | Represent the current virtual health value of an evacuee |
| $B_u$ | Represent a Boolean value to indicate if an evacuee can be assigned to the congestion-ease group |
| $V_n$ | Represent the number of linked neighbour nodes of a vertex |
| $V_t$ | Represent a threshold value of $V_n$ |
| $N_e$ | Represent the total number of evacuees in the vicinity of a node |
| $R_o$ | Represent the routing list in the mailbox of a CPN node |
| $R_o^i$ | Represent the $i$-th item in $R_o$ |
| $Q$ | Represent the QoS value of a path |
| $M_t$ | Represent the movement depth of evacuees |
| $N_{hop}^c$ | Represent the number of hops that has been traversed by an evacuee since being assigned to the "congestion-ease" group |
| $\alpha$ | Represent a constant value and is set to 0.7 |
| $\beta$ | Represent a constant value and is set to 2 |
| $e(i)$ | Represent an evacuee $i$ |
| $b(j)$ | Represent a bottleneck $j$ in the environment, such as a staircase or an exit |

**Pseudocode 1** The Process of Switching an Evacuee Between "Class 1" and "Class 2". EN, Edge Node

When an evacuee reaches the vicinity of an EN, obtain $G_{id}$ of the evacuee
**if** $G_{id} \in G_{one}$ **then**
  gain the virtual health value $Y_c$ of the evacuee and the mean health value $\overline{Y_c}$
  **if** $Y_c < \alpha \overline{Y_c}$ **then**
    $G_{id} \leftarrow G_{two}$
  **end if**
**else**
  **if** $Y_c > \frac{\overline{Y_c}}{\alpha}$ **then**
    $G_{id} \leftarrow G_{one}$
  **end if**
**end if**

**Pseudocode 2** The Process of Switching From "Class 1" or "Class 2" Group to the Congestion-Ease Group. EN, Edge Node

When an evacuee reaches a vertex in the vicinity of an EN
Examine $G_{id}$ of the evacuee
**if** $G_{id} \notin G_{con}$ **then**
  Set $B_u \leftarrow False$
  **if** $V_n \geqslant V_t$ **then**
    Sense the total number of evacuees $N_e$ in proximity.
    /* If the queue length at a vertex is larger than or equals to a certain value $N_t$, then switch the newly arrived evacuee to congestion-ease group. */
    **if** $N_e \geqslant N_t$ **then**
      $G_{id} \leftarrow G_{con}$
      Access $R_o$ in the mailbox of the EN
      Obtain the QoS value $Q_t$ of the top ranked path in $R_o$
      Re-rank $R_o$ with respect to the potential congestion level of routes
      **for** each item $R_o^i$ in $R_o$ **do**
        Obtain the QoS value $Q_i$ of $R_o^i$
        **if** $Q_i \leqslant \beta Q_t$ **then**
          Set $B_u \leftarrow True$
          Break;
        **end if**
      **end for**
    **end if**
  **end if**
  /* If a suitable path exists */
  **if** $B_u = True$ **then**
    Choose the path associated with $Q_i$ as the decision
  **else**
    Choose the original top ranked path associated with $Q_t$ as the decision
  **end if**
**else**
  **if** $N_{hop}^c = M_t$ **then**
    Re-assign the evacuee to its original group
  **end if**
**end if**

### C. TRAFFIC-BALANCE-AWARE GROUPING

Although distributed emergency navigation systems are more robust in hazard environments in comparison with centralised counterparts, they may not lead to a global optimum in terms of load-balancing by pursuing individuals' goal and may induce unexpected stampede or congestion. Furthermore, when a disaster occurs, no matter what metrics (e.g. distance, congestion level, safety level) are involved, emergency navigation algorithms tend to provide paths with the shortest distance to exits in the beginning. This is because destructive crowd behaviours have not yet been generated and the initial intensity of the hazard is low. Hence, initial distribution of evacuees has a significant impact on the result of an evacuation process. Inspired by "uniformity principle" [48], we present a load-balancing grouping mechanism to disperse evacuees among staircases or exits when a hazard occurs. This mechanism can be used during the whole evacuation process but currently we only employ it in the initial stage.

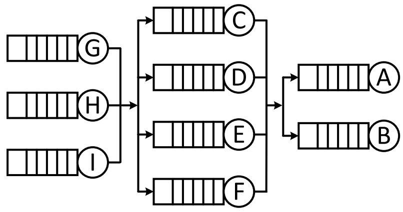


**FIGURE 4.** Queueing model of the building. Each server represents a staircase or exit. Server A and B represent two exits on the ground floor, server C, D, E and F represent the four staircases that connect the ground floor to the basement one, server G, H and I represent the three staircases that connect the basement one to the basement two.

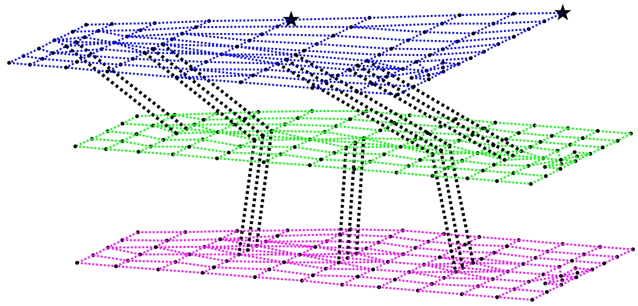

**FIGURE 5.** The graph model of the NANLUOGUXIANG public traffic junction.

As is stated in the "uniformity principle", in order to achieve minimal building evacuation time, all evacuation routes should be saturated and clear their last evacuee at the same time. Hence, the number of evacuees to each evacuation route should be assigned in proportional to the flow rate of the evacuation route. Since staircases where evacuees may congregate due to reduced speed can be considered as the bottlenecks, we simplify the graph representation of the PTJ as shown in Figure 5 (This junction contains two exits on the ground floor, four staircases that connect the ground floor to the basement one, and three staircases that connect the basement one to the basement two) and establish a queueing model as shown in Fig. 4 based on the following assumptions:

1) The evacuees on lower floors do not affect the clearance time of evacuees on upper floors;
2) The staircases or exits will be operated at near-full capacity before civilians on this floor have all evacuated.

The above assumptions can match the realistic situation: for instance, an evacuee $E_3^x$ on the basement two has low likelihood to pass or disturb an evacuee $E_2^y$ on the basement one, on the contrary, $E_2^y$ may affect $E_3^x$ as it may be involved in congestion and block $E_3^x$; For the second assumption, as staircases are bottlenecks owing to the reduced speed and limited space, evacuees will soon congregate at these locations when a disaster occurs. Hence, we can consider each bottleneck as a "server" and calculate the clearance time for $N$ evacuees to traverse an evacuation path $\pi$ using the algorithm introduced in [49].

$$T_c^{\pi}(N) = L(\pi) + \frac{N}{\mu(\pi)} \tag{15}$$

where $T_c(N)$ represents the time to evacuate $N$ evacuees from an evacuation path $\pi$. Term $L(\pi)$ depicts the shortest time for an evacuee to traverse path $\pi$ (which also refers as "lead time"). On the basement one or basement two, $L(\pi)$ is the transmission time for an evacuee to traverse the shortest route from the staircase to the nearest exit, whereas on the ground floor, $L(\pi)$ is zero. Term $\mu(\pi)$ represents the service rate of the bottleneck on path $\pi$, which is the capacity of bottlenecks such as staircases.

According to the "uniformity principle", all evacuation paths should have the same clearance time to achieve the shortest building evacuation time. Hence, the number of evacuees assigned to each staircase or exit can be solved by using equation set (16).

$$\begin{cases} T_c^{\pi}(N_{b(i)}) = T_c^{\pi}(N_{b(j)}) \quad \forall b(i), b(j) \in B \\ \sum_{i=1}^{|B|} N_{b(i)} = N_t \end{cases} \tag{16}$$

where $b(i)$ represents the $i$-th staircase or exit. Term $B$ represents the set of staircases or exits on a floor while $|B|$ depicts the number of staircases or exits. Term $N_t$ represents the total number of evacuees on a floor and $N_{b(i)}$ depicts the number of evacuees assigned to the $i$-th staircase or exit. Since $N_{b(i)}$ represents the number of evacuees, rounding operation is required to obtain closest integer to $N_{b(i)}$. If $N_{b(i)}$ is smaller than zero, this means that no evacuees will be assigned to this bottleneck. Hence, $N_{b(i)}$ will be set to zero and a re-assigning process will be performed to allocate evacuees to other bottlenecks.

The next step after determining the number of evacuees to each staircase or exit is to assign each evacuee to a specific staircase or exit. Because the employed model assume that each staircase or exit are operated at near-full capacity, evacuees should be assigned in a way that the overall distance to staircases or exits is minimised while fulfilling the number restrictions to each staircase or exit solved in (16). This problem is similar to the NP-hard "knapsack problem" but is even more complex. The formulation of this problem is shown in (17).

$$\begin{aligned} \text{minimize} \quad & D_a = \sum_{i=1}^{N_t} \sum_{j=1}^{|B|} K_{b(j)}^{e(i)} D(e(i), b(j)) \\ \text{subject to:} \quad & \sum_{i=1}^{N_t} K_{b(j)}^{e(i)} = N_{b(j)} \\ & \sum_{i=1}^{N_t} \sum_{j=1}^{|B|} K_{b(j)}^{e(i)} = N_t \end{aligned} \tag{17}$$

where $D_a$ denotes the overall distance from evacuees to staircases or exits. Term $K_{b(j)}^{e(i)} \in \{0, 1\}$, if evacuee $e(i)$ uses staircase or exit $b(j)$, $K_{b(j)}^{e(i)} = 1$, otherwise, $K_{b(j)}^{e(i)} = 0$.

Since assigning evacuees to staircases or exits while fulfilling the above conditions is a NP-hard problem, we propose a greedy algorithm to approximate the near-optimal solution. Because most of parameters involved in the algorithm can be calculated in advance, the algorithm can be performed efficiently. The details of the mechanism are shown in Pseudocode 3 and the symbols used are listed in Table 4. We assume that the computations of this algorithm is offloaded to the remote edge computing center, which can rapidly return the results in the preliminary stage of the evacuation process.

**Pseudocode 3** The Process of Assigning Evacuees on One Floor to Each Staircase or Exit (Bottleneck) When a Disaster Occurs

When a disaster breaks out, send the location of evacuees on the floor to the appointed edge node
Retrieval the shortest paths $D(e(i), b(j))$ from each evacuee $e(i)$ to each bottleneck $b(j)$
Generate a row vector $D_{b(j)} = [D(e(1), b(j)), D(e(2), b(j)), \ldots](i = 1 \ldots N_t)$ for each bottleneck $b(j)$
**for all** the evacuees $e(i)$ on the floor **do**
    Find the shortest $D(e(i), b(j))$ and label the associated evacuee $e(i)$
**end for**
**for all** the bottlenecks $b(j)$ on the floor **do**
    Compare the number of labeled evacuee $L_{b(j)}$ in row vector $D_{b(j)}$ with $N_{b(j)}$
    **if** $L_{b(j)} > N_{b(j)}$ **then**
        Label the associated row vector $D_{b(j)}$
    **else**
        Continue
    **end if**
**end for**
**for all** the labeled row vectors **do**
    **for all** the labeled evacuees **do**
        /* Retrieval the associated $D(e(i), b(j))$ and minus the corresponding value in the unlabeled row vector */
        $D^{absolute}(e(i), b(j)) = |D^{labeled}(e(i), b(j)) - D^{unlabeled}(e(i), b(j))|$
    **end for**
**end for**
Sort all the $D^{absolute}(e(i), b(j))$ in ascending order
Replace the bottlenecks in $D^{labeled}(e(i), b(j))$ with the bottlenecks in $D^{unlabeled}(e(i), b(j))$ until the number of evacuees to each bottleneck reaches the desired value

After assigning each evacuee to a specific bottleneck, rather than using the top-ranked path, ENs will choose the best path that contains the specific bottleneck for the evacuee. If no such path is discovered, the edge computing center will perform the Dijkstra's shortest path algorithm to calculate a route (from the source node to the specific bottleneck) for the evacuee. Evacuees will stick to the given paths until satisfying the restriction of movement depth.

## VI. SIMULATION MODEL AND ASSUMPTIONS

We employ a Python based simulation tool, namely the smart environment simulator (SES), which is inspired by the Distributed Building Evacuation Simulator (DBES) [50], [51], to evaluate the effectiveness of the proposed algorithm in fire-related scenarios. It can simulate different aspects of emergency scenarios in PTJs and evaluate diverse courses of actions such as running, walking and stopping. The SES works as a client to send control commands to the open-source simulation platform ''Simulation of Urban MObility'' (SUMO) [52]. As a multi-agent simulator, each entity in the SES is represented by an intelligent agent to interact with the SUMO platform. For instance, evacuees are modelled as pedestrians to realise self-observation, inter-competition and cooperation; fire agent is responsible for simulating the breakout and spreading of fire; floor agents, which depict floors of a building, act as containers for pedestrians as well as maintain the environment data.

### A. BUILDING MODEL

The building model in our experiments simulates the three floors of the NANLUOGUXIANG public transportation junction in Beijing. The basement one and basement two have a dimension of 210 $m$ by 50 $m$ while the ground floor has identical dimension of 180 $m$ by 50 $m$. The height between each floor is approximately 12 $m$. Fig. 5 shows the graph model of this transportation junction.

### B. CIVILIANS MODEL

In this paper, initially, simulated evacuees are considered under two categories based on their diverse mobility and resistance to hazard: (1) the Class 1 evacuees represent prime-aged healthy people with high mobility and resistance to hazard; (2) the Class 2 evacuees represent evacuees with low mobility and resistance to hazard, including aged people, children, or people may have been impaired during the disaster.

The health level of each evacuee is initialized to a value of 100 and decreased based on exposure to the hazard. Each category of evacuees is characterized by their speed and their resistance to the hazard. In the simulations we set the probability of a created civilian agent belonging to either of the two categories to 0.5. A civilian agent is randomly initialized at a particular location on the building graph.

### C. ENERGY CONSUMPTION MODEL

To validate the proposed energy-saving mechanism, we construct an energy usage model of the IoT network, which consists of SNs and ENs. The energy consumption of SNs are omitted under the assumption that SNs will periodically sniff and gather sensory data in a fixed rate. Hence, no matter with or without the energy-saving mechanism, the energy consumption of SNs will stay constant. On the other hand, the energy usage model of ENs considers the energy consumption in communication and computation. The energy consumption during communication processes is further divided into (a) the energy utilisation of data transmission and reception among the on-site ENs, and (b) the energy utilisation of uploading and downloading data to and from the edge computing center. The energy model for (a) is borrowed from [53] while energy model for (b) is borrowed from [39] under the assumption that ENs can communicate with the edge computing center through 3G wireless technology at a data rate of 2 *MB/s*. The detailed parameters in the energy model of ENs is shown in Table 5.

**TABLE 5. The energy utilisation model of ENs which is reconstructed from [53] and [39].**

| Notation | Formula | Value |
|---|---|---|
| Energy utilisation for packet transmission with neighboring ENs | $e_c b d^\alpha + e_t b$ | Term $e_c$ is the energy dissipated per bit per $m^2$ and is chosen to be $10 * 10^{-11}$, $e_t$ is the energy spent by transmission circuitry per bit and is chosen to be $5 * 10^{-8}$. Term $b$ is the number of bits to transmit or receive and $d$ is the distance from transmitter to receiver. Term $\alpha$ is a constant which depends on the attenuation the signal will suffer in the environment and is set to 2. |
| Energy utilisation for packet reception with neighboring ENs | $e_l b$ | Term $e_l$ is the energy spent by reception circuitry per bit and is chosen to be $5 * 10^{-8}$. |
| Energy utilisation for uploading data to the edge computing center | $e_u b$ | Term $e_u$ is the energy spent for uploading per bit and is chosen to be $4.21875 * 10^{-5}$. |
| Energy utilisation for downloading data from the edge computing center | $e_d b$ | Term $e_d$ is the energy spent for downloading per bit and is chosen to be $1.53 * 10^{-4}$. |
| Energy utilisation for an EN in idle state for a time interval of $t$ | $E_s t$ | Term $E_s$ is the power of an EN in idle state and is set to $9.6 * 10^{-3}$ watts. |

The energy consumption of ENs in computation is calculated by an analogy of the code-transformation algorithm introduced in [54], which can simulate the number of CPU cycles executed by each EN. By using this algorithm, we can then obtain the CPU active time of an EN by dividing the recorded total CPU cycles of the EN during the simulation by the CPU frequency. Finally, the energy consumption of the EN can be calculated by multiplying the CPU active time by the CPU power, which equals voltage times current. In our simulation, we presume that the current consumption of CPU in active mode and idle mode is 8.0 $mA$ and 3.2 $mA$, respectively. The CPU frequency is set to 16 $MHz$ and the power supply is set to 3.0 $V$. The Instructions per cycle (IPC) of the CPU is set to 1. The detailed procedures of this algorithm is as follows:

1) Convert operations of each EN (modelled by computer code) to simple CPU instructions;
2) Convert involved decimal numbers in the operations to binary numbers;

3) Perform bitwise operations and sum the number of instructions used in computation;
4) Transform the number of instructions to CPU cycles counts in accordance with the Instructions per Cycle;
5) Convert CPU cycles counts to CPU active time with regard to CPU frequency;
6) Calculate energy consumption of CPU with respect to time cost on each power state.
7) Sum the energy consumption of CPU in each EN to obtain the total energy consumption of the IoT network in computation.

## VII. EXPERIMENTS, RESULTS AND DISCUSSION

We conduct three experiments to investigate the performance of the proposed routing algorithms. In the first experiment, we evaluate the performance of the packet-dropping mechanism by observing the energy consumption and average delay of the ENs. To avoid unnecessary interferences, we run the emergency response system for 5 minutes without introducing evacuees. For latter two experiments, we run 20 simulations each under different levels of occupancy in the aforementioned PTJ model to represent light (100 and 500 evacuees), medium (1000 and 2000 evacuees) and heavy (5000 and 6000 evacuees) traffic load. In the second experiment, we compare three scenarios under the same conditions: in the first two scenarios we use a single metric to guide all the evacuees and do not employ the cooperative strategies; while in the third scenario, we use two metrics, one for each category of evacuees and the three cooperative strategies are employed. In the third experiment, we concentrate on evaluating the effect of the proposed cooperative strategies by comparing with scenarios without these strategies.

Our results are averages over the 20 simulations and are presented as bar charts with error bars, which indicate the average over 20 simulations and the maximum/minimum value in any of the 20 simulations, respectively.

### A. THE PERFORMANCE OF THE PACKET-DROPPING MECHANISM

We assume that 300 ENs are deployed in the given PTJ. ENs exchange necessary information with the edge computing center in every 2 seconds. The uploaded information includes actual number of packets at each EN, the steady-state probability that an EN has one or more type-$k$ packet $q_{n_i,k}$, the remaining battery power of each EN $E_n^i$, the arrival rate of SPs at each EN $\lambda_{n_i}^s$, the arrival rate of ACKs at each EN $\lambda_{n_i}^a$, number of evacuees at each EN, the hazard value at each EN, the probability that SPs of class $k$ that travel from EN $n_i$ to EN $n_j$ $P(n_i, n_j, k)$. The downloaded information is the dropping probability of SPs at each EN. We also assume that the data rate between ENs are $25mbits/s$ and the size of the address of each EN is kept at 16 bits.

As can be seen from Fig. 6, the use of the packet-dropping mechanism can remarkably reduce the CPU cycles in the EN. This is because a SP involves in information-updating at each traversed EN. In addition, the ACK generated by a SP after reaching the destination will trigger the training process of the RNN at each EN along the reverse route. Fig. 7 shows the total energy usage during a 5 minutes long simulation when using the original CPN and the edge computing aided packet-dropping mechanism. As can be seen clearly, although the updates of information with the edge computing center can cost extra 600 Joules of energy, the introducing of the packet-dropping mechanism can significantly reduce the total energy usage.

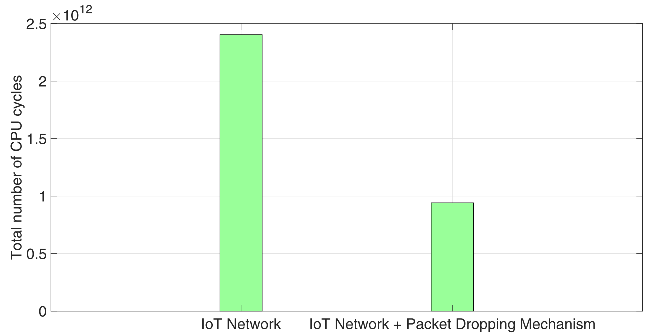


**FIGURE 6.** The total number of CPU cycles of all the ENs in the IoT network with and without the edge computing aided packet dropping mechanism in a 5 minutes simulation run.

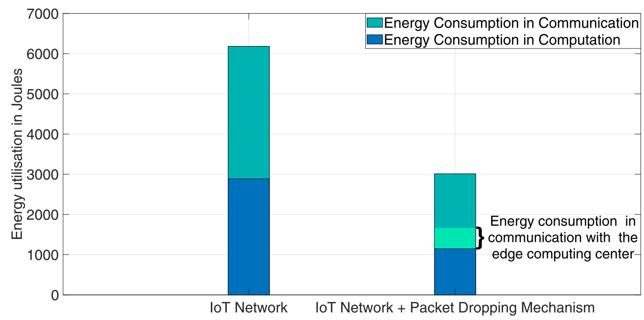


**FIGURE 7.** The total energy consumption of all the ENs in the IoT network with and without the edge computing aided packet dropping mechanism in a 5 minutes simulation run.

By comparing Fig. 9 with Fig. 8, it is clearly shown that the proposed packet-dropping mechanism (PDM) can remarkably reduce the delay at each EN in the network, especially at backbone nodes. The horizontal planes of the two figures represent the geographical locations of ENs deployed on the ground floor of the network. Since the two exits are also on this floor, the ENs on the ground floor not only have to handle the generated packets from the same storey, but also packets from basement one and two.

### B. THE PERFORMANCE OF THE COOPERATIVE STRATEGIES

#### 1) AVERAGE SURVIVAL RATE

As shown in Figure 10, for low levels of occupancy (100 and 500 evacuees), the time metric (TM) achieves better performance than the safety metric (SM), which gives the worst performance overall. This is because unlike TM that distributes evacuees among various available paths, and sometimes even may take the risk to traverse potential hazard areas, SM are congestion-insensitive and tends to guide

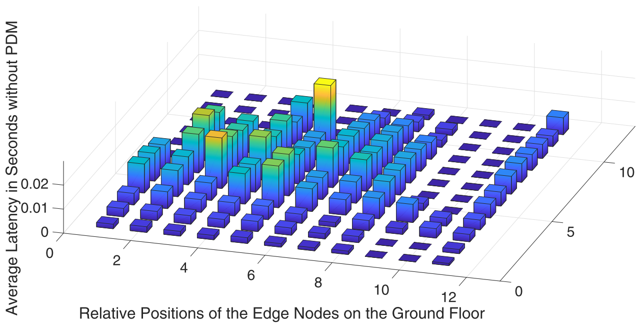


**FIGURE 8.** The average delay in seconds of each EN on the ground floor without the packet-dropping mechanism in a 5 minutes simulation run.

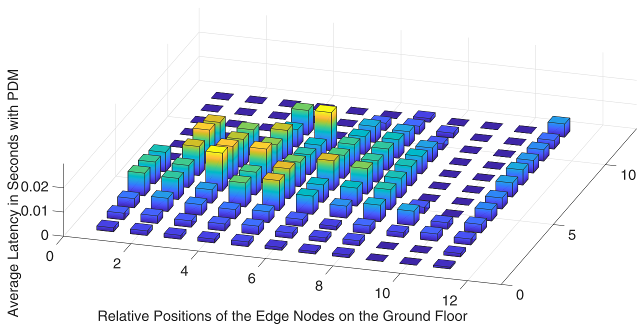


**FIGURE 9.** The average delay in seconds of each EN on the ground floor with aid of the packet-dropping mechanism in a 5 minutes simulation run.

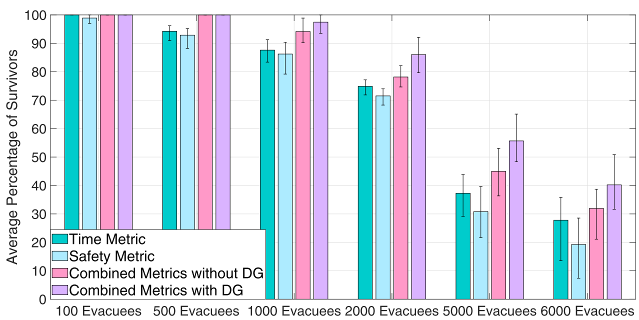


**FIGURE 10.** The average percentage of survivors for each of the decision algorithms. The results are the average of 20 randomized simulation runs, and error bars show the min/max result in any of the 20 simulation runs.

all the evacuees to the safest paths. However, some evacuees may perish due to the long evacuation time caused by congestion. The combined metrics without the dynamic grouping mechanisms (CMNODG) and combined metrics the with the dynamic grouping mechanisms (CMDG) achieve the equally-best performance because they can naturally ease congestion by generating separate channels for two categories of evacuees. Moreover, for CMDG, rather than sticking to the quickest path with higher risk, injured evacuees can switch to the safest path due to the benefit of dynamic grouping and therefore reduce fatalities.

It is more apparent in scenarios with medium and heavy population densities (1000, 2000, 5000 and 6000 evacuees), CMDG apparently achieves the best performance among all the algorithms. This is because the congestion level has a considerable impact on the system performance in

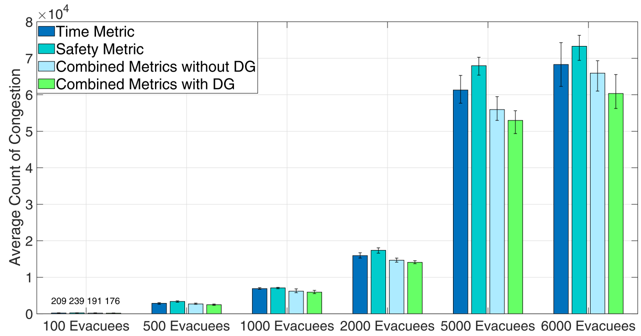


**FIGURE 11.** The average count of congestion for each of the decision algorithms. The results are the average of 20 randomized simulation runs, and error bars show the min/max result in any of the 20 simulation runs. "Count of congestion" is defined as the number of congestion encountered by evacuees during an entire evacuation process. Congestion is considered to occur when an evacuee reaches a node where one or more evacuees have queued up for service. In other words, an evacuee is considered to encounter congestion when it reaches a node with a non-zero queue.

densely-populated scenarios. Because CMDG can assign evacuees to a third group when severe congestion occurs (queue length $> 5$) and suggest a less congested path with an acceptable QoS level, paths in broad areas are more sufficiently used and evacuees can reach exits with less latency. Moreover, the traffic-balance-aware grouping mechanism that initially distributes evacuees to staircases or exits can significantly reduce the likelihood for a path to be overused. Additionally, at this stage, TM can be seen to perform remarkably better than SM. This is because TM can alleviate congestion more efficiently in comparison with SM.

In summary, CMDG gains better results than other algorithms because of customising different evacuees with desirable paths as well as the use of dynamic grouping mechanisms. Towards comparison algorithms, SM gains acceptable performance under light traffic load but bad performance under medium and heavy traffic load. This is because in order to keep all the evacuees far away from the hazard, SM only directs evacuees to safest paths and has the potential to cause jamming. When the population is low, this issue is negligible but can induce continuous congestion in densely-populated scenarios. Conversely, TM achieves gradually worse performance with the increase of occupancy rates. This is due to fact that the quickest paths tend to traverse areas with potential risks and evacuees with low health level may perish owing to the fast spreading of the hazard as well as significantly reduced mobility. In high population densities, the embedded congestion ease mechanism of TM is less effective because injured evacuees with significantly reduced mobility become "obstacles" for other civilians and induce continuous congestion.

### 2) AVERAGE COUNT OF CONGESTION

Figure 11 presents the average count of congestion in each scenario. As expected, CMDG gains the least average count of congestion in all scenarios. This is due to fact that by dispersing evacuees with respect to their physical condition

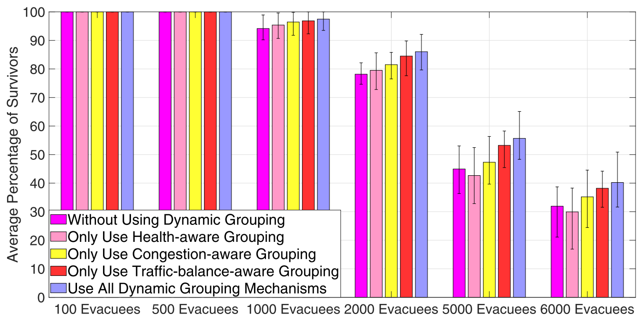


**FIGURE 12. The average percentage of survivors for different dynamic grouping mechanisms. The results are the average of 20 randomized simulation runs, and error bars show the min/max result in any of the 20 simulation runs.**

especially mobility, evacuees can be assigned to different paths and reduce congestion. Moreover, the use of the congestion-aware grouping and the traffic-balance-aware grouping can also reduce congestion. SM achieves the highest congestion level during the evacuation processes as it tends to congregate all the evacuees to the safest path and cause continuous congestion. TM, on the other hand, achieves lower congestion level than SM because of the embedded congestion-ease mechanism.

### C. THE EFFECT OF DYNAMIC GROUPING

Figure 12 shows the average percentage of survivors for CM with different dynamic grouping mechanisms. As can be seen clearly, compared with CMNODG, the use of each individual mechanism can significantly improve the survival rates under medium population densities. However, under the heavy traffic load, since congestion becomes the dominating factor for evacuation efficiency, CMDG with only health-aware grouping mechanism malfunctions due to the quick formation of bottlenecks at the staircases and exits mainly caused by Class 2 evacuees. On the other hand, CMDG with only congestion-aware grouping and CMDG with only traffic-balance-aware grouping mechanism are more efficient in using the interior space and can ease the formation of bottlenecks. Among the three scenarios of using only one grouping mechanism, the traffic-balance-aware grouping gains the best performance, because it is the most efficient in easing the formation of bottlenecks at the staircases and exits. The results also indicate that integrating the three mechanisms can achieve better results than only using one single mechanism.

## VIII. CONCLUSION AND FUTURE WORK

In this paper, we propose an edge computing enabled emergency navigation system that can direct evacuees out of a confined built environment in a cooperative manner while fulfilling the energy consumption and latency requirements of the on-site IoT network. Two cooperative strategies which adjust the type and the associated decision algorithm of evacuees are presented with regard to evacuees' physical conditions and surrounding environments. Furthermore, since the congestion-ease mechanisms do not function at the beginning of an evacuation process owing to the fact that the evacuees have not congregated yet, a load-balancing strategy is proposed to distribute evacuees among all the staircases and exits to ease congestion. In addition, a packet discard mechanism is employed in combination with a G-network model to balance the energy consumption and network latency of the IoT network. The results indicated that this QoS driven routing algorithm with the use of cooperative strategies can achieve higher survival rates. The simulation results also imply that the proposed packet discard mechanism can significantly improve the performance of the IoT network by reducing energy consumption and latency. Since mobile agents can migrate seamlessly through multiple clouds, different portable devices and the edge of the cellular network [55], future research will be directed to design a mobile agent-based emergency response system that has the potential to reduce communication costs and ease network congestion in large-scale emergency evacuations by dynamically optimising the locations of mobile agents and using the mobile edge computing technologies.

## REFERENCES

[1] Z. He, G. Qi, L. Lu, and Y. Chen, "Network-wide identification of turn-level intersection congestion using only low-frequency probe vehicle data," *Transp. Res. C, Emerg. Technol.*, vol. 108, pp. 320–339, Nov. 2019.
[2] H. Bi and E. Gelenbe, "Cloud enabled emergency navigation using faster-than-real-time simulation," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 475–480.
[3] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable IoT architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, Aug. 2017.
[4] E. E. Ugwuanyi, S. Ghosh, M. Iqbal, T. Dagiuklas, S. Mumtaz, and A. Al-Dulaimi, "Co-operative and hybrid replacement caching for multi-access mobile edge computing," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 394–399.
[5] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for edge computing-empowered industrial IoT," *IEEE Internet Things J.*, early access, Dec. 31, 2019, doi: 10.1109/JIOT.2019.2963371.
[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[7] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, May 2016.
[8] F. Müller, O. Wohak, and A. Schadschneider, "Study of influence of groups on evacuation dynamics using a cellular automaton model," *Transp. Res. Procedia*, vol. 2, pp. 168–176, Jan. 2014.
[9] Y. Cheng and X. Zheng, "Emergence of cooperation during an emergency evacuation," *Appl. Math. Comput.*, vol. 320, pp. 485–494, Mar. 2018.
[10] H. Bi and E. Gelenbe, "A survey of algorithms and systems for evacuating people in confined spaces," *Electronics*, vol. 8, no. 6, p. 711, 2019.
[11] H. Bi and E. Gelenbe, "Routing diverse evacuees with cognitive packets," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM WORKSHOPS)*, Mar. 2014, pp. 291–296.
[12] H. Bi, "Routing diverse evacuees with the cognitive packet network algorithm," *Future Internet*, vol. 6, no. 2, pp. 203–222, 2014. [Online]. Available: http://www.mdpi.com/1999-5903/6/2/203
[13] O. J. Akinwande, H. Bi, and E. Gelenbe, "Managing crowds in hazards with dynamic grouping," *IEEE Access*, vol. 3, pp. 1060–1070, 2015.
[14] E. Gelenbe, R. Lent, and Z. Xu, "Design and performance of cognitive packet networks," *Perform. Eval.*, vol. 46, nos. 2–3, pp. 155–176, Oct. 2001.
[15] E. Gelenbe, "Steps toward self-aware networks," *Commun. ACM*, vol. 52, no. 7, p. 66, Jul. 2009.
[16] L. Wang and E. Gelenbe, "Adaptive dispatching of tasks in the cloud," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 33–45, Jan. 2018.
[17] E. Gelenbe and A. Labed, "G-networks with multiple classes of signals and positive customers," *Eur. J. Oper. Res.*, vol. 108, no. 2, pp. 293–305, Jul. 1998.

[18] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 325–349, May 2005.
[19] M. Yavuz and D. W. Paranchych, "Adaptive rate control in high data rate wireless networks," in *Proc. IEEE Wireless Commun. Netw., . WCNC .*, Mar. 2003, pp. 866–871.
[20] K. Akkaya and M. Younis, "An energy-aware QoS routing protocol for wireless sensor networks," in *Proc. 23rd Int. Conf. Distrib. Comput. Syst. Workshops*, May 2003, pp. 710–715.
[21] E. Gelenbe, G. Pujolle, and J. Nelson, *Introduction to Queueing Networks*. Chichester, U.K.: Wiley, 1987.
[22] E. Gelenbe and E. C.-H. Ngai, "Adaptive QoS routing for significant events in wireless sensor networks," in *Proc. 5th IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, Sep. 2008, pp. 410–415.
[23] E. Gelenbe, "On approximate computer system models," *J. ACM*, vol. 22, no. 2, pp. 261–269, Apr. 1975.
[24] E. Gelenbe and G. Pujolle, "The behaviour of a single queue in a general queueing network," *Acta Inf.*, vol. 7, no. 2, pp. 123–136, 1976.
[25] E. Gelenbe, "A diffusion model for packet travel time in a random multihop medium," *ACM Trans. Sensor Netw.*, vol. 3, no. 2, p. 10, Jun. 2007.
[26] O. Khalid, M. U. S. Khan, Y. Huang, S. U. Khan, and A. Zomaya, "EvacSys: A cloud-based service for emergency evacuation," *IEEE Cloud Comput.*, vol. 3, no. 1, pp. 60–68, Jan. 2016.
[27] X. Xu, L. Zhang, S. Sotiriadis, E. Asimakopoulou, M. Li, and N. Bessis, "CLOTHO: A large-scale Internet of Things-based crowd evacuation planning system for disaster management," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3559–3568, Oct. 2018.
[28] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, pp. 487–490, Sep. 2000.
[29] Y. Jiang, B. Chen, X. Li, and Z. Ding, "Dynamic navigation field in the social force model for pedestrian evacuation," *Appl. Math. Model.*, vol. 80, pp. 815–826, Apr. 2020.
[30] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," *Pedestrian Evacuation Dyn.*, vol. 21, no. 2, pp. 21–58, 2002.
[31] K. Rendón Rozo, J. Arellana, A. Santander-Mercado, and M. Jubiz-Diaz, "Modelling building emergency evacuation plans considering the dynamic behaviour of pedestrians using agent-based simulation," *Saf. Sci.*, vol. 113, pp. 276–284, Mar. 2019.
[32] H. Bi, "Emergency navigation systems and algorithms in confined spaces," in *Proc. 3rd Int. Conf. Inf. Commun. Technol. Disaster Manage. (ICT-DM)*, Dec. 2016, pp. 1–8.
[33] B. Horling and V. Lesser, "A survey of multi-agent organizational paradigms," *Knowl. Eng. Rev.*, vol. 19, no. 4, pp. 281–316, Dec. 2004.
[34] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 865–875, Oct. 2004.
[35] E. Gelenbe, N. Schmajuk, J. Staddon, and J. Reif, "Autonomous search by robots and animals: A survey," *Robot. Auto. Syst.*, vol. 22, no. 1, pp. 23–34, Nov. 1997.
[36] E. Gelenbe and Y. Cao, "Autonomous search for mines," *Eur. J. Oper. Res.*, vol. 108, no. 2, pp. 319–333, Jul. 1998.
[37] E. Gelenbe and G. Gorbil, "Wireless networks in emergency management," in *Proc. 1st ACM Int. Workshop Practical Issues Appl. Next Gener. Wireless Netw. (PINGEN)*. New York, NY, USA: ACM, 2012, pp. 1–6.
[38] G. Gorbil and E. Gelenbe, "Resilient emergency evacuation using opportunistic communications," in *Computer and Information Sciences III*. London, U.K.: Springer, 2013, pp. 249–257.
[39] E. Gelenbe and H. Bi, "Emergency navigation without an infrastructure," *Sensors*, vol. 14, no. 8, pp. 15142–15162, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/8/15142
[40] M. Hamdan and M. E. El-Hawary, "A novel genetic algorithm searching approach for dynamic constrained multicast routing," in *Proc. Can. Conf. Electr. Comput. Eng. Toward Caring Humane Technol. (CCECE)*, vol. 2, May 2003, pp. 1127–1130.
[41] M. Zhou, H. Dong, Y. Zhao, P. A. Ioannou, and F.-Y. Wang, "Optimization of crowd evacuation with leaders in urban rail transit stations," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4476–4487, Dec. 2019.
[42] H. Liu, B. Xu, D. Lu, and G. Zhang, "A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 68, pp. 360–376, Jul. 2018.
[43] L. Oliveira, D. Schneider, J. De Souza, and W. Shen, "Mobile device detection through WiFi probe request analysis," *IEEE Access*, vol. 7, pp. 98579–98588, 2019.
[44] E. Gelenbe, Z. Xu, and E. Seref, "Cognitive packet networks," in *Proc. 11th Int. Conf. Tools Artif. Intell.*, Nov. 1999, pp. 47–54.
[45] E. Gelenbe, R. Lent, and A. Nunez, "Self-aware networks and QoS," *Proc. IEEE*, vol. 92, no. 9, pp. 1478–1489, Aug. 2004.
[46] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Comput.*, vol. 1, no. 4, pp. 502–510, Dec. 1989.
[47] E. Gelenbe, "G-networks by triggered customer movement," *J. Appl. Probab.*, vol. 30, no. 03, pp. 742–748, Sep. 1993.
[48] R. L. Francis, "A 'uniformity principle' for evacuation route allocation," *J. Res. Nat. Bur. Standards*, vol. 86, pp. 509–513, Sep./Oct. 1981.
[49] G.-H. Chen and Y.-C. Hung, "On the quickest path problem," *Inf. Process. Lett.*, vol. 46, no. 3, pp. 125–128, Jun. 1993.
[50] A. Filippoupolitis and E. Gelenbe, "A distributed decision support system for building evacuation," in *Proc. 2nd Conf. Human Syst. Interact.*, May 2009, pp. 323–330.
[51] N. Dimakis, A. Filippoupolitis, and E. Gelenbe, "Distributed building evacuation simulator for smart emergency management," *Comput. J.*, vol. 53, no. 9, pp. 1384–1400, Nov. 2010.
[52] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO—Simulation of urban MObility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation (SIMUL)*, Barcelona, Spain, vol. 42, 2011, pp. 55–60.
[53] E. J. Duarte-Melo and M. Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks," in *Proc. Global Telecommun. Conf., . GLOBECOM 02. IEEE*, vol. 1, Nov. 2002, pp. 21–25.
[54] V. Shnayder, M. Hempstead, B.-R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*. New York, NY, USA: ACM, 2004, pp. 188–200.
[55] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," *ETSI White Paper*, vol. 11, no. 11, pp. 1–16, 2015.



**HUIBO BI** (Member, IEEE) received the B.Eng. degree from the Qingdao University of Science and Technology, China, in 2009, the M.Eng. degree from Chongqing University, China, in 2012, and the Ph.D. degree in electrical and electronic engineering from Imperial College London, in 2017, under the supervision of Prof. E. Gelenbe. He is currently a Lecturer with the College of Metropolitan Transportation, Beijing University of Technology. His research interests include intelligent-networked systems, machine learning, emergency management, and distributed computing.



**WEN-LONG SHANG** received the Ph.D. degree from the Centre for Transport Studies, Department of Civil and Environmental Engineering, Imperial College London. He is currently a Lecturer with the College of Metropolitan Transportation, Beijing University of Technology. He has already published several academic articles in peer-reviewed journals and conferences. His research interests include but not limit to: analysis and processing of urban road network data, complex network theory, robustness and resilience of urban road networks, intelligent transport systems, safety science and smart city, large scale optimization, and so on.



**YANYAN CHEN** received the Ph.D. degree in civil engineering from the Harbin University of Technology, China, in 1997. Since 1999, she has been with the Beijing University of Technology, where she is currently the Dean of the College of Metropolitan Transportation. She has published more than 100 articles in journals and six academic books. Her research interests include urban transport planning, travel demand management policies, transport system reliability, and big data mining. She is the Co-Chair of Urban Transportation Committee in China Highway and Transportation Society.