

Spatial Clustering in Slotted ALOHA Two-Hop Random Access for Machine Type Communications

by

Ziwen Zhao, B.Sc.

A thesis submitted to the
Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
August, 2016

©Copyright
Ziwen Zhao, 2016

The undersigned hereby recommends to the
Faculty of Graduate and Postdoctoral Affairs
acceptance of the thesis

**Spatial Clustering in Slotted ALOHA Two-Hop Random
Access for Machine Type Communications**

submitted by **Ziwen Zhao, B.Sc.**

in partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

External Examiner

Professor Halim Yanikomeroglu, Thesis Co-Supervisor

Professor Sebastian Szyszkowicz , Thesis Co-supervisor

Professor Yvan Labiche, Chair,
Department of Systems and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
August, 2016

Abstract

The LTE random access procedures proposed in 3GPP for Machine Type Communication (MTC) in current cellular systems may become overwhelmed when too many machine devices attempt to upload their data. In this thesis, we propose a two-hop slotted ALOHA-based cluster random access method to improve the performance of MTC. Due to the offloading from the random access channel to the slotted ALOHA, the number of contending devices is reduced, which alleviates the collision problem and results in a better performance. The utilization of the slotted ALOHA method also simplifies the access procedure, which can decrease the energy consumption of machine devices. Compared with previous literature, the reason we select the slotted ALOHA method for the first hop communication is that it is easy to implement and more efficient in resources and energy utilization.

We introduce a clustering geometry model for machine locations. In this model, the machines join different clusters according to their location proximity, and the clustering of machine locations can be adjusted. We also define a clustering metric to measure the clustering of the locations. Some literature on different clustering algorithms is studied in the thesis. We conduct some mathematical analysis on the average energy consumption of machine devices in terms of the collision probability of message transmission.

We examine the impact of location clustering on the system performance. Our results show that as the machine locations become more clustered, the overall performance metrics are improved. It can also be seen that the energy consumption can be dramatically decreased using the proposed scheme due to the simplified access procedure proposed. We also examine the impact of the number of devices and the number of clusters on the system performance with and without a fixed amount of clustering. Generally speaking, as the number of devices increases, the performance worsens, and as the number of clusters increases, the performance improves.

Acknowledgments

First, I would like to express my extreme gratefulness to my two co-supervisors, Halim Yanikomeroglu and Sebastian Szyszkowicz. It is professor Halim who invited me to work with him when I took one of his courses in 2013 September. He permits my flexible working schedule and provides me with a broad range of knowledge in the research area. Professor Halim is such a gentleman who is always nice and considerate to people. He is very generous to students and tries his best to help them in work and life. If it's not him I may not set forth on a journey as a researcher. I could not wished for a better supervisor.

I also need to show my great appreciations to Dr. Sebastian Szyszkowicz. Without him, I cannot make all the achievements in my research work. He offers me great help in my thesis and paper publishing in all details. He makes contributions on the aspects of materials filling, contents organization, simulation coding, results presentation, format correction, and proofreading, basically everything in my work. He spends a lot of time and effort to help me complete my research work. I learn a lot from him on how to be a good researcher and make academic publishing. His supervision not only covers me but also many other researchers in our group, and it gives us a lot of confidence to make more achievements working with him.

I also want to express my thankfulness to my research buddy Tamer Beitalmal. He kindly offers me all the help I need whenever I ask. He continuously motivates and encourages me to complete my work. Sometimes I have some technical difficulties impeding my work moving on, and he always has some solutions there and is willing to tutor me. He really saves a lot of time and makes some shortcuts in my research work.

I also want to express my gratitudes to some other researchers in our group who grant me much help from different aspects, such as Faraj Lagum, Irem Bor Yaliniz, Meisam Mirahsan, Ramy Gohary, etc. It is them who makes me feel like working as a team and motivated to move on in my work.

On the personal side, I want to express my gratefulness to my parents. It is them who always back me up unconditionally and supports me all the way to lead a life and finish my Master Degree in Canada. I will love them forever.

This work is supported in part by Huawei Canada Co., Ltd., in part by Telus Corp., and in part by the Ontario Ministry of Economic Development and Innovation's ORF-RE (Ontario Research Fund-Research Excellence) Program.

Table of Contents

Abstract	iii
Acknowledgments	iv
Table of Contents	vi
List of Tables	ix
List of Figures	x
Nomenclature	xii
1 Introduction	1
1.1 Overview	1
1.2 Applications of MTC	3
1.2.1 Smart Grid Applications	3
1.2.2 Bridge Monitoring Applications	5
1.2.3 Vehicle-Related Applications	6
1.2.4 Other Applications	7
1.3 Contributions and Organization	7
2 Background	10
2.1 Summary of Symbols	10
2.2 3GPP Random Access Procedure	12
2.2.1 LTE Random Access Procedure	12
2.2.2 RAN Overload Control	15
2.2.2.1 Push-Based Methods	15
2.2.2.2 Pull-Based Scheme	16

2.2.3	Evaluation of Push-Based RAN Overload Control Schemes	16
2.2.3.1	ACB Schemes	17
2.2.3.2	Separate RACH Resources for MTC	17
2.2.3.3	Dynamic Allocation of RACH Resources	17
2.2.3.4	MTC Specific Backoff Scheme	18
2.2.3.5	Slotted Access	18
2.2.3.6	Conclusion and Proposal	18
2.3	Two-Hop Cluster-Based Random Access in Literature	19
2.4	Clustering Metric	20
2.4.1	Voronoi Tessellation and Delaunay Triangulation	20
2.4.1.1	Voronoi Tessellation	20
2.4.1.2	Delaunay Triangulation	21
2.4.2	CoV and Voronoi Cell Area Based Clustering Metric	22
2.5	Clustering Algorithms in Literature	24
2.5.1	Clustering Algorithms Overview	24
2.5.2	Clustering Algorithms in Detail	26
2.5.2.1	Distance and Similarity Measures	27
2.5.2.2	Hierarchical Clustering	29
2.5.2.3	Partitional Clustering	39
2.5.2.4	DBSCAN Algorithm	45
2.5.3	Choice of Clustering Algorithm	50
3	Proposed System Model	51
3.1	3GPP-Compliant LTE Random Access Simulator	51
3.2	Location Generator	57
3.3	Slotted ALOHA-Based Cluster Random Access	58
3.4	Analysis of Energy Consumption	60
3.4.1	Energy Consumption in Slotted ALOHA	60
3.4.2	Energy Consumption in RACH	61
3.5	C_V and the Inputs of the Location Generator	62
4	Simulation Results	64
4.1	Performance as a Function of Clustering	64
4.2	Performance as a Function of the Number of Machines	66
4.3	Performance as a Function of the Number of Clusters	67

4.4	Energy Consumption as a Function of Collision Probability	69
5	Conclusion and Future Work	71
5.1	Conclusion	71
5.2	Future Work	73
5.2.1	Analytical Results	73
5.2.2	Implementation of Other Communication Schemes within the Cluster	73
5.2.2.1	CSMA/CA	73
5.2.2.2	Reference LTE Random Access Method	73
5.2.3	Cluster Discovery	74
	List of References	75

List of Tables

2.1	Summary of Symbols in Chapter 2	11
2.2	Summary of the evaluation on push-based RAN overload control schemes [31].	19
2.3	Some examples of distance measures [47].	28
2.4	Different schemes of coefficients values $\alpha_i, \alpha_j, \beta$, and γ in (2.7). n_i, n_j and n_l are the number of data points in cluster C_i, C_j , and C_l , respectively [40, 42, 52].	32
2.5	Two features f_i and f_j with values of 0 and 1.	36
3.1	Simulation parameter setup [6, 27, 28].	53
3.2	Validation of LTE random access simulator with 3GPP results [6].	55

List of Figures

1.1	Comparison between three schemes: (a) LTE one-hop reference scheme, (b) Clustered two-hop scheme in previous literature, and (c) Proposed slotted ALOHA-based clustered two-hop scheme.	8
1.2	Comparison between our research and previous research.	9
2.1	PRACH configuration [26].	12
2.2	(a) Random access procedures four messages exchange (b) Procedure details assuming Msg1 and Msg3 both have one collision [5, 26].	13
2.3	Voronoi Tessellation (dashed lines) and Delaunay Triangulation (solid lines) for a set of points [33].	21
2.4	Voronoi tessellation of clustered points.	22
2.5	Heterogeneous machine locations with different cluster radius R and C_V , with the number of nodes $N=2500$, the probability of isolated node $P_{isolated}=0.01$, and the number of clusters $M=50$	23
2.6	Hierarchical clustering dendrogram. The agglomerative method is from bottom to top while the divisive method is the opposite. Two clusters are obtained by cutting the dendrogram at an appropriate level [44].	30
2.7	Flowchart of the agglomerative hierarchical clustering algorithm [44].	31
2.8	Flowchart of the divisive hierarchical clustering algorithm [44].	37
2.9	Simulation results of hierarchical clustering algorithm on spatial traffic patterns with different C_V values.	38
2.10	Simulation Results of K -means algorithm on heterogeneous geometry with different C_V values.	46
2.11	Neighborhood ϵ for points p and q	47
2.12	Examples of core, border, and noise points in DBSCAN algorithm.	48
2.13	Simulation results of DBSCAN algorithm on spatial traffic patterns with different C_V values.	49
3.1	Flowchart of LTE random access procedure in one subframe.	54

3.2	An example of the track of a machine's (ID 235) states in our simulator.	56
3.3	The states of four machines (ID 704, 944, 1045, 1668) from subframe 4620 to subframe 4740	57
3.4	Clustering geometry model. The blue links are the slotted ALOHA communication between the CM and the CH, and the red links are the LTE random access procedure between CH and BS. The inputs of our location generator are $N = 200$, $M = 10$, $R = 25$, and $P_{isolated} = 0.01$	59
3.5	(a) Random access procedure details assuming Msg1 and Msg3 both have one collision (b) slotted ALOHA procedure details assuming only one collision occurs.	60
3.6	Relationship between C_V and the inputs of location generator.	63
4.1	Flowchart of the complete simulation system.	65
4.2	Performance vs. C_V with 95% confidence interval. C_V is varied by tuning the cluster radius R . The other inputs of location generator are: the number of clusters $M = 50$, the number of devices $N = 2000$, and the probability of isolated node $P_{isolated} = 0.01$	66
4.3	Performance vs. number of devices with 95% confidence interval. The other three inputs of the location generator are fixed, i.e. $M = 50$, $R = 50$, and $P_{isolated} = 0.01$. However, the C_V increases as the number of devices grows.	67
4.4	Performance vs. number of devices. The C_V is maintained around 5 by tuning R from the lookup table. The other inputs of the location generator are $M = 50$, and $P_{isolated} = 0.01$	68
4.5	Performance vs. number of clusters. The other three inputs of the location generator are fixed, i.e. $N = 2000$, $R = 50$, and $P_{isolated} = 0.01$. However, the C_V decreases as the number of devices grows.	68
4.6	Performance vs. number of clusters M . The C_V is maintained around 5 by tuning R from the lookup table. The other inputs of location generator are $N = 2000$ and $P_{isolated} = 0.01$	69
4.7	Comparison of average energy consumption of machine devices in slotted ALOHA vs. RACH, varying the collision probability. For the random access procedure, the collision probability of Msg1 is varied while for Msg3 only four cases are considered: the collision probability is equal to 0.02, 0.05, 0.1, and 0.2.	70

Nomenclature

ACB	Access Class Barring
BI	Backoff Indicator
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
BS	Base Station
CH	Cluster Head
CM	Cluster Member
CN	Core Network
CoV	Coefficient of Variation
CR	Cognitive Radio
C-RNTI	Cell Radio Network Temporary Identifier
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CURE	Clustering Using Representatives
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EAB	Extended Access Barring
eNB	eNode B
HARQ	Hybrid Automatic Repeat Request
H2H	Human-to-Human
IoT	Internet of Things
ISODATA	Iterative Self-Organizing Data Analysis Technique
LOS	Line-of-Sight
LTE	Long-Term Evolution
MAC	Medium Access Control
MTC	Machine Type Communication
M2M	Machine-to-Machine

NLOS	Non-Line-of-Sight
PDCCH	Physical Downlink Control Channel
PPP	Poison Point Process
PRACH	Physical Random Access Channel
PSD	Power Spectral Density
PUSCH	Physical Uplink Shared Channel
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RAO	Random Access Opportunity
RAR	Random Access Response
RA-RNTI	Random Access Radio Network Temporary Identifier
ROCK	Robust Clustering using Links
SG	Smart Grid
SINR	Signal to Interference plus Noise Ratio
TDMA	Time Division Multiple Access
UE	User Equipment
WSN	Wireless Sensor Network
3GPP	3rd Generation Partnership Project

Chapter 1

Introduction

1.1 Overview

An unprecedented market growth of machine devices is expected in the coming years. Many machines will connect to the Internet, forming the Internet of Things (IoT), which empowers a full automatic communication system between machines without necessary human intervention [1,2]. Machine-to-machine (M2M) communications has a wide range of applications, such as smart metering, tracking and tracing, automatic payment, eHealth, surveillance and security, and infrastructure management [3].

The realization of M2M communications could rely on current long-term evolution (LTE) cellular networks, because of the easier infrastructure installation, and also the possibility of reliable long distance M2M communications, especially for those with mobility [4]. However, traditional cellular networks designed for human-centric device communications may not support M2M communications effectively, which have distinctive features such as a large number of terminals, small data transmissions, low mobility, time tolerance, and group-based features [3]. A major challenge for M2M communications is to handle the traffic generated by massive simultaneous access requests. A medium access control (MAC) procedure for M2M is proposed in [5]. The machines send connection requests through the physical random access channels (PRACHs), which have limited resources. If two requests use the same channels concurrently, there will be a collision. The overload access signalling in random access channel (RACH) can lead to large delay, packet loss, and even service unavailability. Thus, a tailored solution is needed to adapt cellular networks to M2M characteristics and accommodate a large number of machines.

Several mechanisms are proposed to protect the network from massive RACH overload from M2M traffic on the radio access network (RAN) level [6]. These mechanisms include access class barring (ACB) and backoff schemes. In ACB, the access load is well controlled by the restriction of access requests, while in backoff, the retransmission of each failure access request is delayed by a random period. Both methods can spread peak traffic in time and prevent batch arrivals. Other solutions, like RACH resources separation and dynamic allocation, divide and adapt the resources for M2M and human-to-human (H2H) traffic in order to protect H2H communications from the surge of M2M communications.

To efficiently adapt to the characteristics of M2M traffic, several studies proposed random access methods based on clustering methods [7–11]. An improved random access method is proposed in [7] in response to group paging, where the paging intervals are separated into three periods for data aggregation and random access procedures to decrease the number of machines that perform direct access attempts and thus alleviate RACH congestion. In [8], a cluster head (CH) performs the random access procedure on behalf of all the machines in an access group, and cluster members (CMs) take turns to transmit data on the channels allocated for the CH. In [9, 10], a CH is elected in each cluster to process the access attempts from the CMs. The network spatially reuses the random access resources to increase the supported number of machines. The inter-cluster interference is also analyzed, with the conclusion that it does not affect the results; in fact, full reuse of random access resource among clusters is feasible. In [11], cognitive radio technology is used for the CH gathering the traffic from multiple machines to avoid interference with the primary link between users and the base station (BS). The cluster-based methods mentioned above use either the LTE random access procedure or the time division multiple access (TDMA) method to implement the communication within a cluster. However, the former method puts high requirements on the function of the CH, and the TDMA method is inconsistent with the nature of bursty Machine Type Communication (MTC) traffic and may waste channel resources.

In this thesis, we propose a slotted ALOHA-based two-hop cluster random access method. In this method, the CH aggregates data from CMs through slotted ALOHA and then initiates the LTE random access procedure to the BS. Due to the offload from the RACH to the slotted ALOHA, the number of contending machines in the RACH is reduced, resulting in fewer collisions and a better performance. We also

introduce a clustering geometry model for M2M locations. This model is more suitable for current LTE networks with random and cluster features, such as body area networks, coffee bars, personal residences, and commercial buildings. We examine the impact of location clustering on the system performance. Our results show that as the machine locations become more clustered, the overall performance metrics are improved. It can also be seen that the energy consumption can be dramatically decreased using the proposed scheme due to the simplified access procedure proposed. We also examine the impact of the number of devices and the number of clusters on the system performance with and without a fixed amount of clustering. Generally speaking, as the number of devices increases, the performance worsens, and as the number of clusters increases, the performance improves. In summary, the proposed scheme outperforms the reference scheme as long as the machine locations are sufficiently clustered.

1.2 Applications of MTC

There are a variety of applications of MTC or wireless sensor networks (WSNs) in the current communication networks.

1.2.1 Smart Grid Applications

MTC can be applied in smart grid (SG) [12–14]. SG is an emerging technology which is designed to integrate advanced communication technologies into electrical power grids. Due to the low cost and high function, MTC is widely used in SG applications to achieve an automated operation.

In [12], the authors provide an overview of various applications of WSNs in SG and a discussion of existing issues. In SG, the WSN comprehensively senses the specific objects, collects the useful data and controls devices. The WSN in SG enables a two-way information exchange, monitoring, control, and maintenance in real time. Due to the two-way communication mode, users can interact extensively with the network, both in reporting the power consumption data, and even feeding back domestically produced energy into the SG. The reported data is in digital form instead of electromechanical, which enables real-time pricing and net metering. The use of WSN also shifts the work from manual operation to remote, automated supervision. With the assistance of WSN, smart power control can be achieved with pro-active

protection, and the outages can be avoided before they start. Finally, the smart system will be self-healing. It is noted that in SG, WSN can be used to sense and monitor the transmission line, substation, home, etc. Based on these observations, some intelligent function can be realized, such as 3D visualization, smart home, smart inspection, etc. However, some issues in the SG need to be addressed, such as energy support, security, reliability, and standardization.

In [13], the authors give a general overview of the applications of MTC in SG. It is noted that the application of M2M communication in the field of SG has already been well developed. An electric grid equipped with a smart system is able to produce, distribute, and consume power in the most efficient and intelligent way. The communication only occurs among machine devices, such as smart meters and sensors. The privacy and security need to be guaranteed since the message might flow through multiple devices and the entire system runs automatically without too much human supervision. A complex connection between a huge number of electrical appliances needs to be supported so they can share the information efficiently. This allows for an efficient power generation system, such as local solar and wind generators. The applications will also be developed in the area of home-based devices with the requirement of scalability and fast communication. The authors provide a detailed M2M communication model for SG and verify the effectiveness of different adopted communication technologies such as WiFi, ZigBee, and Bluetooth. They stressed choosing ZigBee for the enabler of M2M communications, as its performance is far much better than the others'. Additionally, the authors propose a technique to improve the traditional ZigBee-based M2M communications. It is also noted that the combination of different M2M communication technologies is worth studying in the future.

In [14], the authors first demonstrate the general architecture and business characteristics of the IoT. Subsequently, a layered architecture of IoT applied in the area of SG is proposed from the perspective of business demand in four areas, namely electricity transmission, electric substation, electricity distribution and electricity usage. A comparison is also conducted between IoT and the traditional electric power communication network performance. An application-based wireless sensor is proposed for the production processes for SG. The smart electricity networking solutions in the physical layer and the interaction in SG are also analyzed. In summary, this paper proposes a solution to build IoT for SG considering the features and demand in SG.

1.2.2 Bridge Monitoring Applications

The WSN can also be used for bridge monitoring and diagnosis [15–17]. In [15], the author proposed an event detection strategy for a WSN deployed for railway bridge health monitoring. The proposed scheme considers a set of constraints of sensor networks and makes an efficient use of the limited resources such as battery power and memory. A simple but robust event detection algorithm is proposed, which takes care of both accuracy as well as the low delay. Lower power consumption is achieved by operating the nodes only when there is a train on the bridge. In [16], the authors develop a bridge diagnosis system for daily bridge maintenance. The system utilizes a few technologies, including WSN, signal processing, and structure analysis of a bridge. Some external forces, such as wind pressure and running vehicles, can cause the bridge to vibrate, while the vibration data is collected through the WSN. After the data collection, independent component analysis and spectral analysis is used to extract character frequency. The independent component analysis method can separate an independent signal from mixed input signals. The diagnosis system can understand some phenomena like deterioration and corrosion of a bridge, by which the bridge's health condition can be assessed and a repair or reinforcement work can be considered. The system's hardware is composed of several sensor modules, a data storage device, and a data acquisition device. Experiments were conducted to test the diagnosis system at Kokura, in the city of Kitakyushu in Japan - the required data was successfully obtained. In [17], the authors designed a structural health monitoring system for the bridge deck according to the measurement of temperature and humidity through WSN. The system could also predict various possible damage. Two sensor nodes were used to measure the temperature and humidity on both sides of the bridge deck. The collected data were analyzed by LABVIEW-based software and stored in a database. The monitoring system has many benefits, including low costs, increased efficiency of the bridge service, and reduced maintenance costs. In addition, the system can issue proactive and precise warnings in terms of the place of occurrence and the severity of possible damage. The results show that this system is qualified to be used as a structural health monitoring system in medium to large bridges.

1.2.3 Vehicle-Related Applications

In the applications associated with vehicles, WSNs also has a wide utilization [18–20]. In [18], the authors propose an adaptive navigation method with WSNs, which uses the technology of WiMAX multihop relay networks for the communications between vehicles in order to increase the reliability and efficiency of inter-vehicle communications. Traditional navigation systems use the information broadcasted by a traffic information center. However, the information gathered by the centre may not be sufficient to provide an optimal route, since the monitoring devices which report the traffic information may have a limited coverage of roads. Hence, the traditional navigation system may have some drawbacks, such as providing only partial traffic information on some roads, long update periods, and low reliability by using the traffic information center. Traditional navigation can usually provide only the shortest distance route, which may not be the optimal one in terms of travel time and gasoline consumption. The authors therefore propose a novel, WSN-based adaptive navigation approach. Sensors are installed on vehicles, which can provide real-time traffic information and exchange traffic information among neighboring vehicles. As a result, three metrics can be obtained by the navigation system: road traffic density, road class, and road distance, which can be used to compute an optimal route. The simulation results show that the proposed scheme has a better performance than other traditional methods from the perspective of average travel time and average gasoline consumption, while also providing a competitive average travel distance.

In [19], the authors study a WSN-based application of online-site tracking. After identifying the problem for single target case, the problem of multiple targets case can be generalized. The online vehicle search and tracking algorithms are based on a WSN method. The simulation results show that these algorithms have good performance in terms of tracking time, without so much satisfaction of energy consumption. The authors conclude that the vehicle tracking system can also be developed to an advanced system which has the ability to track automobiles, regulate traffic, take photographs of traffic violations, and also send the data to a BS, thereby notifying it of the violations.

In [20], the authors propose to seismically detect vehicles by using unpowered geophone sensors. The event triggering is captured in different environments, such as loose earth or concrete approaches. The geophones are connected with wires with the ability to detect event and analyze data. However, the application has a challenge

in the feature extraction. The paper proposes a new feature extraction algorithm called spectral statistics and wavelet coefficients characterization. By virtue of this algorithm, the feature is extracted from both the frequency and the time-frequency domain analysis of the seismic signals, including the spectrum, the power spectral density (PSD) and the wavelet coefficients.

1.2.4 Other Applications

MTC is also widely used in other areas of application such as e-healthcare [21], traffic control [22], parking service [23], light control [24], smart home [25].

In summary, MTC has already been widely used in a variety of applications. It will be popularized further in the next generation of communication networks, with more and more requirements for fully automated communication systems free from the assistance from human beings. Therefore, more studies should be done to overcome the challenges in MTC, especially regarding the overloading issues caused by the massive number of machine devices accessing networks. This should eventually enable the global utilization of MTC in the future networks.

1.3 Contributions and Organization

In this thesis, we propose a clustered slotted ALOHA-based two-hop random access procedure for MTC. We introduce a clustered pattern of machine locations and study the impact of the clustering on the system performance.

In 3GPP, it is proposed to use the LTE one-hop reference scheme. Some previous literature modifies it into a clustered two-hop scheme. On the basis of this literature, we propose a slotted ALOHA-based clustered two-hop scheme. The differences between the three schemes are illustrated in Fig. 1.1. The reason we select the slotted ALOHA method for the first hop communication is that it is easy to implement and more efficient in terms of resources and energy utilization.

This thesis mainly makes two contributions. The first is that it uses the slotted ALOHA method in the communication within clusters. The second is the utilization of the clustered pattern of machine locations. The comparison between our research and previous research is summarized in Fig. 1.2.

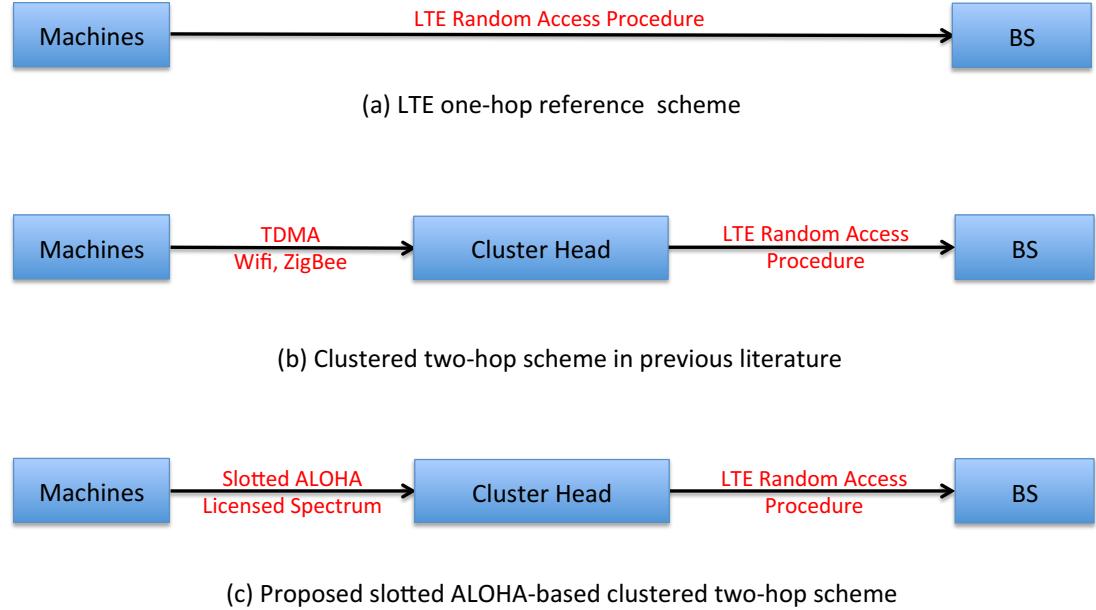


Figure 1.1: Comparison between three schemes: (a) LTE one-hop reference scheme, (b) Clustered two-hop scheme in previous literature, and (c) Proposed slotted ALOHA-based clustered two-hop scheme.

The remainder of the thesis is organized as follows:

- Chapter 2 introduces the standard 3GPP random access procedure. It also introduces and evaluates several overloading control methods in the RACH.
- Chapter 3 introduces some background knowledge. First, we demonstrate the general idea of the two-hop cluster-based random access method. Then we define a clustering metric based on the coefficient of variation (CoV) and the Voronoi tessellation. Next, we present a review of the literature on clustering algorithms and discuss some of them in detail. The key point in this section is that we select to use the hierarchical clustering algorithm in this work according to our needs.
- Chapter 4 demonstrates the system model of our proposed scheme. First, we explain the entire system of slotted ALOHA-based cluster random access. Then we present a specific analysis of the energy consumption in both procedures of slotted ALOHA and one-hop LTE random access. Next, we examine the

relationship between the clustering metrics C_V and the inputs of our location generator.

- Chapter 5 shows the simulation results against different parameters.
- Chapter 6 concludes the work of the thesis, and propose some future work.

A manuscript developed in part based on the work in this thesis is:

Ziwen Zhao, Sebastian Szyszkowicz, Tamer Beitalmal, and Halim Yanikomeroglu, “Spatial clustering in slotted ALOHA two-hop random access for machine type communication”, to appear in *IEEE Global Communications Conference (Globecom) 2016*, 4–8 December 2016, Washington, DC, USA.

	Previous literature	This thesis
Intra-Cluster Communication	Scheduled Scheme	Slotted ALOHA
Resources for Intra-Cluster Communication	Unlicensed Resource (WiFi, ZigBee)	Resources are migrated from PRACH, All-licensed spectrum
Spatial Traffic Pattern	Homogeneous (PPP)	Clustered

Figure 1.2: Comparison between our research and previous research.

Chapter 2

Background

We first give a list of the main symbols used in this chapter. Then we have a literature review on the 3GPP random access procedure in Sec 2.2. We introduce the general idea of two-hop cluster-based random access in Sec 2.3. Our research is also based on clustered communication and we intend to study the impact of the clustering of machine locations on the system performance metrics; therefore, we need to define a metric for the clustering in Sec 2.4. In our system model, the machines are separated into different clusters; therefore, we also need a clustering algorithm to do this. We study and expand the topic of clustering algorithms in Sec 2.5, where we have a lot of discussions on different algorithms in detail. But a key point in this section is that we select the hierarchical clustering algorithm in our simulation work.

2.1 Summary of Symbols

The main symbols in this chapter are summarized below in Table 2.1.

Table 2.1: Summary of Symbols in Chapter 2

C_V	scaled coefficient of variation
k	scaled factor of C_V used to normalize C_V to 1 when it's poisson point process
σ	the variation of Voronoi cell areas
μ	the mean of Voronoi cell areas
$\alpha_i, \alpha_j, \beta$, and γ	coefficients of clustering linkage equation 2.7
f_i, f_j	an example of two binary features
J	criterion function in partitional clustering
$P(N, K)$	computational complexity for partitional clustering with N objects and K clusters
Γ	partition matrix
γ_{ij}	element of partition matrix, $\gamma_{ij}=1$ if $x_j \in$ cluster i , $\gamma_{ij} = 0$ otherwise
M	the array of cluster centroids
m_i	cluster centroid of i th cluster
S_T	total scatter matrix
S_W	within-cluster scatter matrix
S_B	between-cluster scatter matrix
$tr()$	trace function
ρ	a threshold which is used when a new cluster is needed
$\delta()$	binary mismatch function of features
ϵ	the size of epsilon neighborhood
m	minimum points needed within the epsilon neighborhood

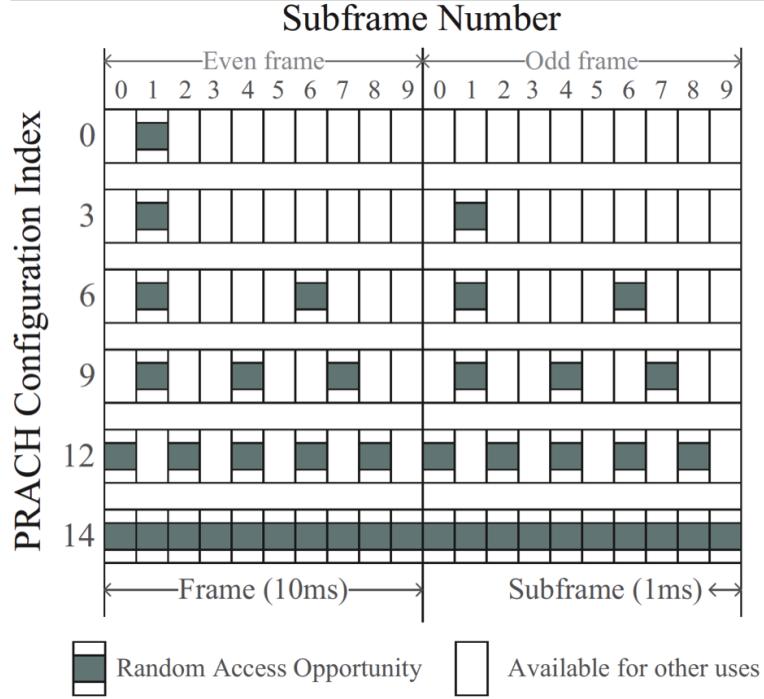


Figure 2.1: PRACH configuration [26].

2.2 3GPP Random Access Procedure

2.2.1 LTE Random Access Procedure

In an LTE network, a machine device will trigger a random access procedure whenever it needs to set up connection with the eNodeB. In this random access procedure, all machines directly connect to the eNodeB in one hop. They send access requests to the eNodeB through the random access channel. The RACH consists of a number of subframes for random access opportunities (RAOs) which are released every few subframes according to the specific configuration, as depicted in Fig. 2.1 [26]. The handshake consists of a four-message signaling exchange depicted in Fig. 2.2 [5, 26].

Msg1 is the preamble transmission. The device waits for the next random access slot and randomly selects a preamble from the set of 54 ($N_{preamble}$) available random access preambles and transmits it in the PRACH; the resources are broadcasted by the eNodeB periodically. The duration of the preamble transmission is $T_{MSG1} = 1\text{ s}$. If two or more devices choose the same preambles simultaneously, a collision occurs.

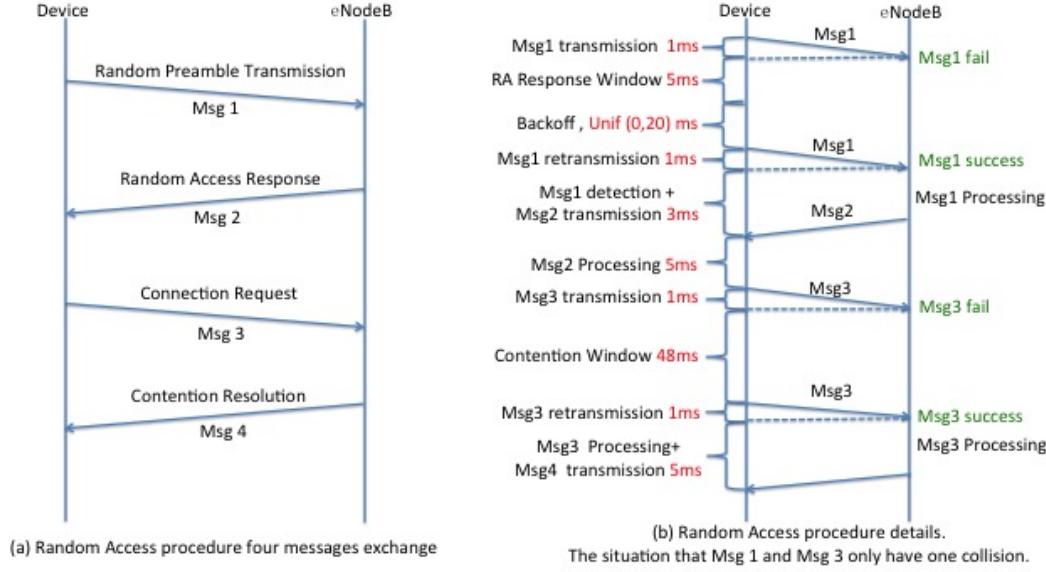


Figure 2.2: (a) Random access procedures four messages exchange (b) Procedure details assuming Msg1 and Msg3 both have one collision [5, 26].

If a preamble is transmitted by only one device, it can be successfully received and decoded by the eNodeB. After the device transmits Msg1, it waits for the Random Access Response (RAR), i.e. Msg2, until a time window $T_{RAR} = 5\text{ ms}$ expires.

Msg2 is the response to the random access preambles transmitted in Msg1. For each successfully decoded preamble, the eNodeB calculates its Random Access Radio Network Temporary Identifier (RA-RNTI) based on the PRACH where Msg1 was transmitted. The message of the RAR can include the following information:

- Identification of the preamble.
- Timing adjustment to synchronize the data transmission.
- Temporary Cell Radio Network Temporary Identifier (C-RNTI).
- The uplink grant channel for Msg3 transmission.

The duration of the preamble detection at the eNodeB and Msg2 transmission time is $T_{MSG2} = 3\text{ ms}$. If multiple devices select the same preamble in the same random access slot, there will be a collision. The detection of the collision depends on the difference of arrival time. If the gap between different arrival is detected, the eNodeB will be aware of the collision and no information will be provided related

to that preamble in the next RAR. In this case, the collision devices will not see the identifier of the used preamble in the corresponding RAR message. They will back off for a random period between 0 and the backoff indicator (*BI*) notified in the RAR message after the expiration of T_{RAR} , and retry the random access. The failure to decode Msg1 can also be caused by the power fading, so the machines need to increase the transmission power for each retransmission. The ramping power is simplified assuming that the success probability of each Msg1 transmission is $(1 - e^{-i})$, where i denotes the transmission times. There is a maximum number of preamble transmission times ($Max_{preamble}$), after which the packet should be dropped and an error is reported to the upper layer. However, if the same preambles are received constructively without the collision of detection, the devices transmitting it in Msg1 will receive the same RAR which will result in a collision in Msg3.

Msg3 is the connection request message. Devices that successfully receive RAR associated with their transmitted preambles will send Msg3 to request connection in the uplink channel granted in Msg2. The processing time between the reception of Msg2 and the transmission of Msg3 is $T_{MSG3} = 5\text{ ms}$, and the Msg3 transmission time is $T_{TranMSG3} = 1\text{ ms}$. After the transmission of Msg3, the device waits for the acknowledgment until the contention resolution timer (48 ms) expires. The undetected collision devices will use the same uplink resources to transmit Msg3. This will result in a collision, meaning that no acknowledgment will be sent back to them. The failure of Msg3 can also result from the channel fading. We assume the success probability of Msg3 delivery is 90%. After not receiving any acknowledgment, the failure devices will retransmit Msg3 after the expiration of a waiting window unless it exceeds the maximum number of allowed retransmission times ($N_{MSG3max}$).

Msg4 is an acknowledgment message to Msg3. After successfully decoding Msg3, the eNodeB sends Msg4 to the devices. Upon the reception of Msg4, the random access procedure is regarded as complete. The duration for processing Msg3 and sending Msg4 is $T_{MSG4} = 5\text{ ms}$. If the device does not receive Msg4 within the maximum allowed transmission times of Msg3, a random access problem is reported to the upper layer and a new access attempt is scheduled after a random backoff period.

2.2.2 RAN Overload Control

A potential number of MTC devices is expected to be located in the same area, and the network may have to face the surge of MTC traffic. Signalling network congestion may occur when massive access attempts occur simultaneously, which could lead to large delays, packet loss or even service unavailability. 3GPP [6] proposes several mechanisms to protect the RAN for RACH overload at the RAN level, i.e., mechanisms to handle MTC access load without a significant impact on H2H traffic. Generally speaking, there are two types of control methods: push-based methods and pull-based methods. In push-based methods, it is the user equipment (UE) devices that initiate access requests to the network, while in pull-based methods, it is the network that sends paging messages to MTC devices. Upon receiving the paging messages, the MTC devices will perform the connection establishment.

2.2.2.1 Push-Based Methods

Access Class Barring Schemes Access Class Barring (ACB) schemes can be used to prohibit the devices of some classes from accessing the network, in which the access loads in RACH can be controlled. Each specific MTC class can be barred or not barred by ACB. Additionally, a barring factor can be used to control the probability of barring the access.

Individual ACB Scaling In this method, the network broadcasts the access control parameter, which can be adjusted on a per UE basis. Each individual UE has the same priority to scale the access control parameter. The parameter can also be adjusted to conduct different control levels based on the current severity of congestion.

Extended Access Barring The Extended Access Barring (EAB) method assigns some low-priority devices to the group of UE configured for EAB, in which no new access classes are introduced. Those low-priority devices are considered more tolerant to access restrictions than any other UEs. When severe congestion occurs, the network can restrict the access from ‘UE configured for EAB’ while still permitting the access of other UEs.

Separate RACH Resources for MTC This method separates the resources for M2M and H2H traffic in order to prevent the impact of MTC access overload on the

traditional H2H communications.

Dynamic Allocation of RACH Resources In the scenario that the network has the ability to predict the surge of M2M access loads, it can dynamically allocate additional RACH resources from the resources for data transmission to alleviate the MTC congestion problem.

MTC Specific Backoff Scheme When the access attempt of a UE fails, it can retransmit the access request after a period. If all the collided UEs back off in the same period, they will retransmit in the same subframe, which results in a high possibility of another collision. Therefore, a random backoff scheme can be implemented to spread the access attempts and avoid batch arrivals.

Slotted Access In this method, each MTC device only accesses during its dedicated access slot, which is assigned by the network. The scheduled method can completely prevent the occurrence of collision.

2.2.2.2 Pull-Based Scheme

If the network is aware that MTC devices have data to transmit or the network needs information from MTC devices, the core network (CN) can page the MTC devices. Upon receiving the paging message, the MTC devices will perform random access procedures to establish the connections. The eNodeB can control the paging message considering the current access load condition. The paging method also involves a backoff time for MTC devices, which indicates the time of access after the reception of the paging message.

2.2.3 Evaluation of Push-Based RAN Overload Control Schemes

3GPP has done some simulation and evaluation towards the push-based RAN overload control schemes [27, 28].

2.2.3.1 ACB Schemes

ACB schemes can control the potential surge of access attempts from MTC and alleviate the overload problems efficiently. Compared to other push-based schemes, ACB can provide a more effective solution to the RACH congestion problem. ACB can also introduce an access class barring factor to allow different control granularity. Using large barring times and barring factor can handle high RACH congestion levels, but the access delay will also be increased significantly, which may not be acceptable for some time-sensitive MTC devices. Furthermore, the ACB schemes cannot respond promptly to some congestion problems that occur in a very short time period.

2.2.3.2 Separate RACH Resources for MTC

When M2M and H2H communication shares the same RACH resources, they have the same access success probability. So if we want to eliminate the impact of M2M traffic on H2H, separate resources can be allocated for each type of communication. In LTE, the separation of resources can be achieved either by splitting the preambles into two groups or allocating different channels in time and frequency. However, the method of resource separation may not be very effective when one group of resources cannot be fully used. For example, when there is little H2H traffic and heavy M2M traffic, the resources for H2H are free while the resources for M2M are already exhausted. Therefore, the RACH resources separation method is not so attractive. The simulation results show that this method does reduce the impact on H2H devices, but the improvement is limited when the congestion level is very high.

2.2.3.3 Dynamic Allocation of RACH Resources

For some applications like smart metering, which reports data periodically, the network can predict the surge of access attempts from those M2M applications. Therefore, the network can dynamically allocate more RACH resources in order to handle the surge of this access load. Generally speaking, this method is always beneficial for resolving the overload problems of M2M traffic without any bad influence on other indicators. The only drawback is that the additional RACH resources will occupy PUSCH resources. Hence, the method should only be used when the congestion happens in RACH and the PUSCH resources are not overloaded.

2.2.3.4 MTC Specific Backoff Scheme

A backoff scheme can be used to delay the retransmissions of access requests. Longer backoff windows can improve the collision probability of M2M access load in RACH, while the side effect is a considerable access delay. However, considering the backoff scheme has a similar effect to ACB scheme, which is more powerful, this approach seems to be less attractive. The simulation results show that the backoff scheme does have some potential to alleviate the congestion problem, but the improvement is limited when the intensity of access attempts is high.

2.2.3.5 Slotted Access

The network assigns dedicated access slots for MTC devices, and these devices can only send access attempts at their own slots, in which the collision situation can be completely eliminated. In theory, the slotted access method can be most efficient if an ideal distribution algorithm can be designed. However, without an optimized solution, many access opportunities might be wasted. Therefore, this method seems not very attractive until a good solution can be found.

2.2.3.6 Conclusion and Proposal

In most cases, when the congestion level is low, the network should configure low control on the access load from MTC devices in order to maintain the system efficiency.

Then, when there are a huge number of MTC devices sending access requests in a very short period, the eNodeB will face the surge of M2M access attempts. The load increase will typically span several seconds. During this transition from a low load level to a high level, only dynamic RACH allocation schemes can increase the RACH resources in a timely fashion, based on the specific congestion level, and maintain the efficiency of the system. If the congestion level exceeds the control ability of the dynamic RACH allocation schemes, other time spreading control schemes can be utilized to further improve the congestion. Therefore, 3GPP [27] proposes that dynamic RACH allocation scheme should be used as a basic solution to address RACH overload, and possibly other schemes as supplementary solutions.

3GPP [28] has summarized some evaluation of the push-based RAN overload control schemes mentioned above in Table 2.2.

Table 2.2: Summary of the evaluation on push-based RAN overload control schemes [31].

RAN overload control scheme	Attractive?	Reason
Access Class Barring	Yes	The only scheme that could control the potential surge of access attempts, which could effectively avoid/alleviate the RACH congestion.
Separate RACH resources	No	Not efficient, because it will split the RACH resources into 2 separate pools.
Dynamic allocation of RACH resources	No	It is mostly like implementation specific without any stage-3 impacts.
MTC Specific Backoff scheme	No	It has quite similar effects with the Access Class Barring scheme, but the Access Class Barring scheme is better.
Slotted access	Not clear	It is a competitive solution in theory, however, good solution is expected in the subsequent RAN2 discussions.

2.3 Two-Hop Cluster-Based Random Access in Literature

To efficiently adapt to the characteristics of a network like LTE, cluster-based random access is proposed for MTC. The machines gather into logical clusters and a CH is selected within each cluster as a relay to aggregate data from CMs and forward it to the eNodeB. This method is particularly suitable for, e.g., smart metering applications, where nodes are static or low-mobility, and therefore the structure of a cluster can be maintained for a long time without the computational expense of frequent cluster re-organization. Most of the data from sensors is delay-tolerant, hence the data can wait for the accumulation in the CH buffer and be forwarded to the eNodeB in one package. In this method, the traffic can be shaped into a small number of bulky loads, which is more appropriate for cellular networks.

Cluster-based schemes have many benefits for MTC communication. Most machines upload data to the CH and only the CHs contend in the RACH. Hence the

access load can be dramatically decreased in the RACH, which can significantly alleviate the collision problems. Additionally, a crowd of neighbor sensors associated with the same event may upload similar information and the CH can perform data compression to reduce redundancy. The link between CM and CH is usually short-distance and line-of-sight (LOS), which has a better channel gain than the long-distance and non-line-of-sight (NLOS) link with the eNodeB. Due to the better channel gains and fewer retransmissions, the energy of machines can be conserved.

2.4 Clustering Metric

In this section, we intend to define a metric to measure the clustering of a set of given points. There are two main considerations in the determination of the metric. One is the measure such as the distance between nearest neighbors or the density of an area, and here we select the cell areas of the Voronoi tessellation, which is introduced in Sec 2.4.1.1. Another is the statistics of the measure such as mean or variance, here we select a scaled CoV as the statistics, which is defined in Sec 2.4.2.

2.4.1 Voronoi Tessellation and Delaunay Triangulation

2.4.1.1 Voronoi Tessellation

The Voronoi tessellation is defined as follows [29]:

Definition 2.4.1. *Voronoi tessellation.* Given a point pattern $P = \{p_1, p_2, \dots, p_n\}$, the Voronoi tessellation $T = \{C_{p1}, C_{p2}, \dots, C_{pn}\}$ is the set of cells such that every location, $y \in C_{pi}$, is nearer to p_i than any other point in P . This can be expressed formally as

$$C_{pi} = \{y \in \mathbb{R}^d : \|y - p_i\| \leq \|y - p_j\| \text{ for } i, j \in 1, \dots, n\}. \quad (2.1)$$

Fig. 2.3 gives an example of Voronoi tessellation (dashed lines) [30].

Fig. 2.4 gives an example of Voronoi tessellation for clustered points.

Generally speaking, the Voronoi tessellation is a partition of a plane according to a set of given points. The given points are called seeds, and for each seed there is

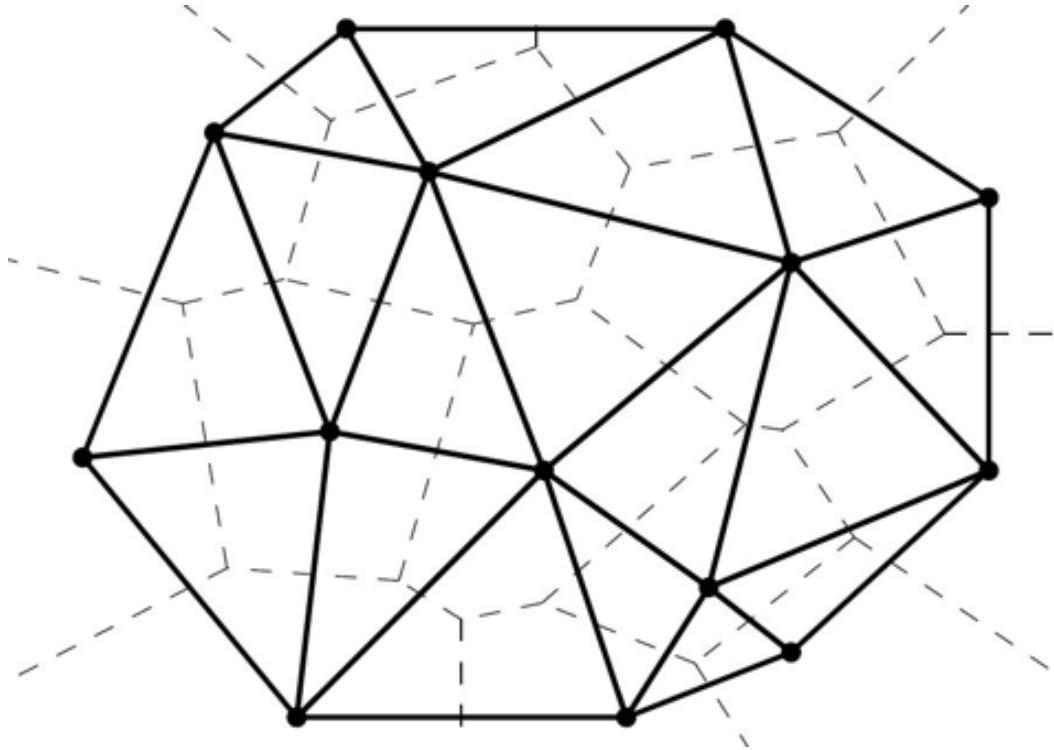


Figure 2.3: Voronoi Tessellation (dashed lines) and Delaunay Triangulation (solid lines) for a set of points [33].

a corresponding region, within which any point is closer to this seed than any other seed.

Voronoi tessellations are associated with a wide range of applications. For example, in wireless communication, the region serviced by a BS is a Voronoi cell and the collection of these regions constitute a Voronoi tessellation, provided that each device is served by its nearest BS. Voronoi tessellations are also applied in other fields such as astrophysics, biology, geography, ecology, industrial engineering, and seismology.

2.4.1.2 Delaunay Triangulation

The Delaunay Triangulation is defined as follows [31]:

Definition 2.4.2. *Delaunay Triangulation.* In two dimensions, after obtaining the Voronoi tessellation, each of its vertices is given by an intersection of exactly $d+1$ Voronoi cells. The corresponding $d+1$ nuclei define a *Delaunay cell*, and all the Delaunay cells constitute the *Delaunay Triangulation*.

The Voronoi tessellation and Delaunay triangulation are said to be dual, since the

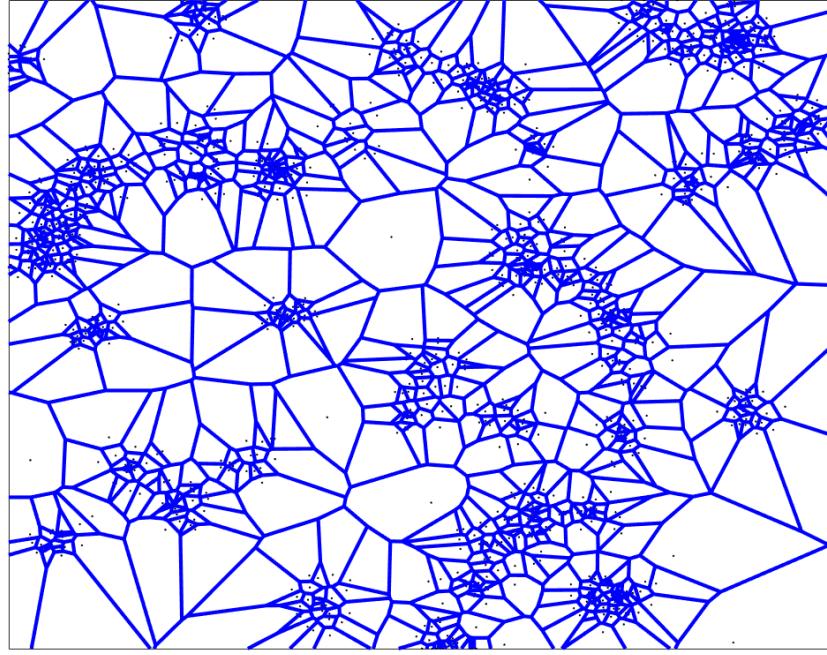


Figure 2.4: Voronoi tessellation of clustered points.

vertices of the Voronoi tessellation correspond to the Delaunay cells. Fig. 2.3 shows an example of Delaunay triangulation (solid lines).

2.4.2 CoV and Voronoi Cell Area Based Clustering Metric

In [32], it is proposed to use a scaled coefficient of variation (CoV) of the Voronoi cell areas to measure traffic heterogeneity. The CoV of a random value is defined as the ratio of its standard deviation to its mean. The CoV-based metric of Voronoi cell areas is defined as

$$C_V = \frac{1}{k} \cdot \frac{\sigma}{\mu}, \quad k \approx 0.529, \quad (2.2)$$

where μ is the mean and σ is the standard deviation of the Voronoi cell areas, and k is a factor to normalize C_V to 1 when the points are taken from a Poisson point process (PPP) [32]. Fig. 2.5 demonstrates different points patterns and their C_V . The first three subfigures show clustered points, with a cluster radius of 10, 50, and 100 meters, respectively. The last subfigure shows homogeneous (PPP) points. As

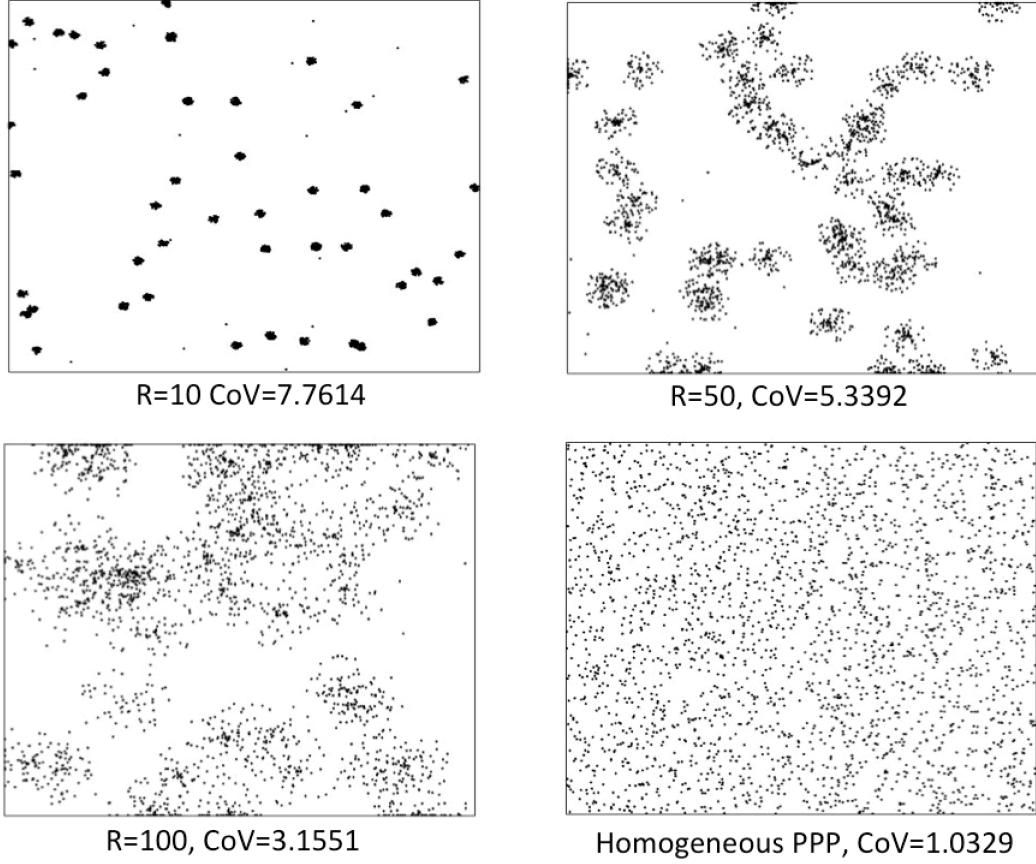


Figure 2.5: Heterogeneous machine locations with different cluster radius R and C_V , with the number of nodes $N=2500$, the probability of isolated node $P_{isolated}=0.01$, and the number of clusters $M=50$.

can be seen from Fig. 2.5, as the geometry of the points become more clustered, the value of C_V increases. The geometry is clustered if $C_V > 1$, while it is repulsive if $C_V < 1$ [32, 33].

2.5 Clustering Algorithms in Literature

2.5.1 Clustering Algorithms Overview

Every day, people need to deal with a huge amount of information which is represented as data. In order to analyze these data, one important method is to classify them into different categories or clusters. Actually, classification plays a very important role in the activities of human beings to understand objects. In order to observe and understand a new object, people usually need to extract its features, which can be used to compare with other known objects and thereby a comprehensive cognition can be obtained by recognizing the similarity and dissimilarity. Classification system can be viewed as a mapping from a set of input variables x_1, \dots, x_d , to an output variable y representing the class label [34]. Classification methods can be divided into two types, supervised and unsupervised, depending on whether they assign new inputs to one of a finite number of discrete supervised classes or unsupervised categories [34–36]. In supervised classification, a number of samples (the objects which are already labeled in different classes) are used to train the system to classify new objects. The system can be modeled as a mathematical function:

$$y_k = y_k(x; w), \quad (2.3)$$

where x is the input data and w is adjustable parameters which are determined by the set of training samples. In unsupervised classification, which can also be called clustering, no training samples can be used for the empirical-based classification. Clustering algorithms can separate these unlabelled data into a number of “natural” sets which do not have a known and accurate definition. As pointed out by Backer and Jain [37], “in cluster analysis a group of objects is split up into a number of more or less homogeneous subgroups on the basis of an often subjectively chosen measure of similarity (i.e., chosen subjectively based on its ability to create interesting clusters), such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups”.

Clustering algorithms divide objects into different clusters. The definition of clusters is not globally agreed. Most researchers select the criteria of clustering based on internal homogeneity and external heterogeneity [38, 39]. A general criterion is that the features of objects in the same cluster should be similar to each other and

in different clusters should not. In [38], some types of clustering are described as below:

- (Hard) partitional clustering attempts to seek a K -partition of \mathbf{X} , $C = \{C_1, \dots, C_K\}(K \leq N)$, such that
 1. $C_i \neq \emptyset, i = 1, \dots, K$;
 2. $\bigcup_{i=1}^K C_i = \mathbf{X}$;
 3. $C_i \cap C_j = \emptyset, i, j = 1, \dots, K$ and $i \neq j$.
- Hierarchical clustering attempts to construct a tree-like nested structure partition of \mathbf{X} , $H = \{H_1, \dots, H_Q\}(Q \leq N)$, such that $C_i \in H_m, C_j \in H_l$, and $m > l$ imply $C_i \subset C_j$ or $C_i \cap C_j = \emptyset$ for all $i, j \neq i, m, l = 1, \dots, Q$.

For hard partitional clustering, an object can only belong to one cluster. However, in fuzzy clustering, an object is allowed to belong to many clusters with a membership variable $u_{i,j} \in [0, 1]$, which represent the membership degree of the j th object in the i th cluster and it needs to satisfy two constraints:

$$\sum_{i=1}^c u_{i,j} = 1, \forall j \quad \text{and} \quad \sum_{j=1}^N u_{i,j} < N, \forall i \quad (2.4)$$

as introduced in fuzzy set theory [40].

Clustering method usually has four basic steps [41].

1. *Feature selection or extraction.* As introduced in [42, 43], feature selection is choosing a prominent feature from a group of candidates while feature extraction transforms the original data into some meaningful and representative features. Both of them are crucial to the effectiveness of clustering procedures. Ideal features should obviously distinguish objects belonging to different clusters, be immune to noise and easy to extract.
2. *Clustering algorithm design or selection.* This step consists of proximity measure and clustering construction. Proximity measure is directly used to connect similar objects and separate different objects by virtue of the data representation. In clustering applications, it usually requires an accurate mathematical definition of proximity measure. There are a variety of cluster algorithms in the

literature. However, no algorithm can universally solve all the problems. Therefore, it is important to clarify the characteristics of the problems beforehand, in order to select or design an appropriate clustering algorithm.

3. *Cluster validation.* Given the same data set, different clustering algorithms may generate different results. And even for the same algorithm, varying the internal parameters and the input order of the original data may lead to different outputs. Hence, a standard evaluation criterion is important to assess the results and examine the validity of the used algorithms. These criteria should be objective and unbiased to all candidates. Also, they can do some analysis like the number of clusters hidden in the data, whether the result is meaningful or just an artifact, or the advantages and disadvantages of the algorithm. Generally, there are three types of cluster validation criteria: external indices, internal indices, and relative indices. External indices are based on prespecified structures. It compares the results derived from the used algorithm with the prior structures, and this criterion is usually used as a standard to validate the clustering method. Internal tests are irrelevant with external knowledge but only depends on the original data. Relative criteria focus on the comparison of different clustering structures in order to select one which may best reflect the feature of the original data.
4. *Results interpretation.* After obtaining the division structures of data, further analysis and interpretation may be needed to provide some insights of the original data.

The clustering process needs trial-and-errors. No universal rules can determine the selection of data features and clustering algorithms. Validation criteria can provide some assessment about algorithm performances, but even the choice of criteria is not globally agreed and remains as an issue in the research.

2.5.2 Clustering Algorithms in Detail

The taxonomies of clustering algorithms can be different according to different criteria. Generally, clustering algorithms are classified into two categories: hierarchical clustering and partitional clustering according to the cluster generation process. Hierarchical clustering either recursively splits data from one cluster or combines data

from singleton clusters. While partitional clustering directly divides data into a certain number of clusters without the hierarchical structure. We first introduce the distance and similarity measure, which is the basis of most clustering algorithms. And then we discuss three types of clustering algorithms in detail namely the hierarchical, partitional and DBSCAN clustering algorithm.

2.5.2.1 Distance and Similarity Measures

In order to determine the closeness or proximity between objects, we usually need a standard to measure and define the distance or similarity. Generally, a cluster centroid may be required to represent the cluster so that we can treat a cluster as an object as well.

A data object is usually expressed as a vector, each element of which refers to one of the features. The features can be continuous or binary, nominal or ordinal, quantitative or qualitative, which determine the corresponding measure mechanisms.

A dissimilarity function is defined with some satisfied constrains [44]:

- Symmetry: $D(x_i, x_j) = D(x_j, x_i);$
- Positivity: $D(x_i, x_j) \geq 0$ for all $i, j.$

If it also satisfies two additional conditions,

- Triangle inequality: $D(x_i, x_j) \leq D(x_i, x_k) + D(x_k, x_j)$ for all $i, j, k;$
- Reflexivity: $D(x_i, x_j) = 0 \iff x_i = x_j,$

it is called a metric.

Likewise, a similarity function is defined with satisfied constrains as below:

- Symmetry: $S(x_i, x_j) = S(x_j, x_i);$
- Positivity: $0 \leq S(x_i, x_j) \leq 1,$ for all $i, j.$

If it also satisfies two additional conditions,

- $S(x_i, x_j)S(x_j, x_k) \leq [S(x_i, x_j) + S(x_j, x_k)]S(x_i, x_k)$ for all $i, j, k;$
- $S(x_i, x_j) = 1 \iff x_i = x_j,$

Table 2.3: Some examples of distance measures [47].

Measures	Forms
Minkowski distance	$D_{ij} = (\sum_{l=1}^d x_{il} - x_{jl} ^{1/n})^n$
Euclidean distance	$D_{ij} = (\sum_{l=1}^d x_{il} - x_{jl} ^{1/2})^n$
City-block distance	$D_{ij} = \sum_{l=1}^d x_{il} - x_{jl} $
Sup distance	$D_{ij} = \max_{1 \leq l \leq d} x_{il} - x_{jl} $
Mahalanobis distance	$D_{ij} = (x_i - x_j)^T S^{-1} (x_i - x_j),$ where S is the within-group covariance matrix.
Pearson correlation	$D_{ij} = (1 - r_{ij})/2,$ where $r_{ij} = \frac{\sum_{l=1}^d (x_{il} - \bar{x}_i)(x_{jl} - \bar{x}_j)}{\sqrt{\sum_{l=1}^d (x_{il} - \bar{x}_i)^2} \sqrt{\sum_{l=1}^d (x_{jl} - \bar{x}_j)^2}}$
Point symmetry distance	$D_{ir} = \min_{j=1, \dots, N \text{ and } j \neq i} \frac{\ (x_i - x_r) + (x_j - x_r)\ }{\ (x_i - x_r)\ + \ (x_j - x_r)\ }$
Cosine similarity	$S_{ij} = \cos \alpha = \frac{x_i^T x_j}{\ x_i\ \ x_j\ }$

it is called a similarity metric.

Assume there are N data objects, We can then build an $N \times N$ symmetric matrix, in which the $(i, j)^{th}$ element refers to the similarity/dissimilarity of i^{th} and j^{th} objects ($i, j \in 1, \dots, N$).

The similarity measure is usually used for qualitative variables while dissimilarity/distance measure is more suitable for continuous variables. Table 2.3 [44] summarizes some typical distance measures. The selection of measures is problem dependent. For binary variables, it usually uses similarity measures. Suppose we use two binary subscripts to count absent or present features in two objects. For example, n_{00} and n_{11} refer to the number of features which are absent and present in both two objects, respectively. n_{01} and n_{10} refer to the number of features which exists only in one object. Then two commonly used similarity measures of the two objects are defined as below [44]:

- $S_{ij} = \frac{n_{11} + n_{00}}{n_{11} + n_{00} + \omega(n_{10} + n_{01})},$
 $\omega = 1$, simple matching coefficient,
 $\omega = 2$, Rogers and Tanimoto measure,
 $\omega = 1/2$, Gower and Legendre measure.

Where ω is the weight of unmatched features, and it determines the contribution of these features.

- $S_{ij} = \frac{n_{11}}{n_{11} + \omega(n_{10} + n_{01})}$,
- $\omega = 1$, Jaccard coefficient,
- $\omega = 2$, Sokal and Sneath measure,
- $\omega = 1/2$, Gower and Legendre measure.

This measure emphasizes the co-existence feature while neglecting the role of co-absence.

If nominal features have more than two states, they can be mapped into a set of binary features [45]. A matching criterion can be defined as below:

$$S_{ij} = \frac{1}{d} \sum_{l=1}^d S_{ijl}, \quad (2.5)$$

where

$$S_{ijl} = \begin{cases} 0, & \text{if } i \text{ and } j \text{ do not match.} \\ 1, & \text{if } i \text{ and } j \text{ match.} \end{cases} \quad (2.6)$$

[46]. Ordinal features order states based on some standards demonstrated in [45].

Generally, a common method to compute the distance is like this: first, map all the variables of objects into the range of $[0, 1]$. Next, we can use the measures in Table 2.3. Alternatively, binary variables can be converted from them and binary similarity function can be used afterwards. It is noted that these methods can result in the loss of information. To fix this problem, an improved method is proposed by Gower in [47]: $S_{ij} = (\sum_{l=1}^d \eta_{ijl} S_{ijl}) / (\sum_{l=1}^d \eta_{ijl})$, where S_{ijl} refers to the similarity of the l th feature and η_{ijl} is a 0-1 coefficient determined by the presence of the measure of the two objects.

2.5.2.2 Hierarchical Clustering

In this paper, we use the hierarchical clustering algorithm [41] to cluster the spatial traffic. There are two types of hierarchical clustering algorithms. One is the agglomerative method which groups data from singleton clusters to a whole cluster, while the other one is the divisive method which groups data in an opposite way. Both types of clustering algorithms organize data objects in a hierarchical structure based on their

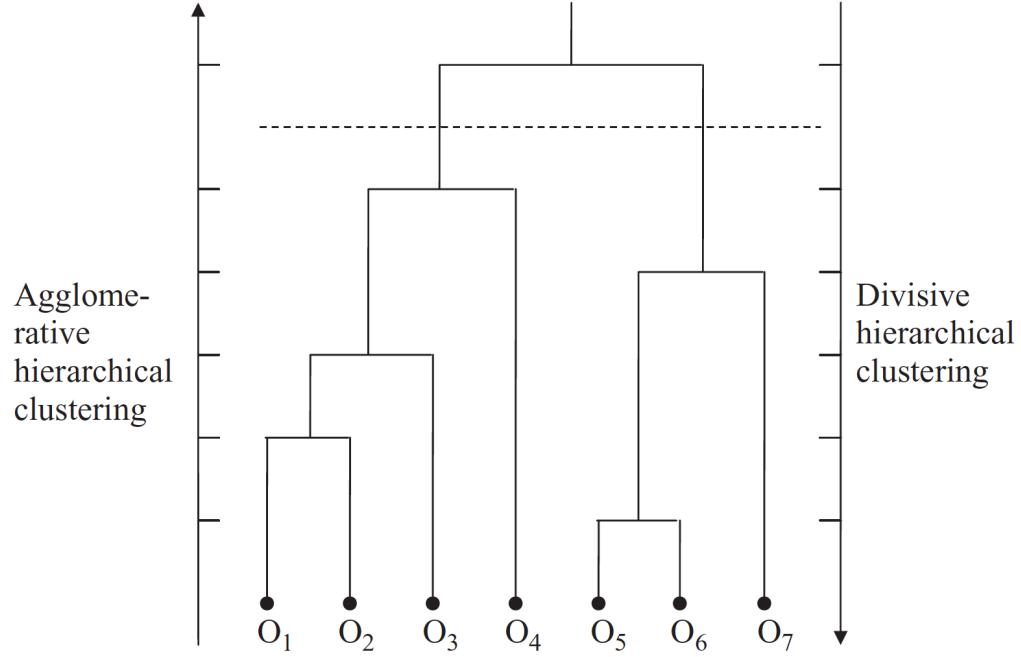


Figure 2.6: Hierarchical clustering dendrogram. The agglomerative method is from bottom to top while the divisive method is the opposite. Two clusters are obtained by cutting the dendrogram at an appropriate level [44].

similarities. The hierarchical structure is usually illustrated in a form of binary tree or dendrogram, as depicted in Fig. 2.6 [41]. The top node of the dendrogram is the one cluster consisting of all data and each node at the bottom represents a single data object. The intermediate nodes in the dendrogram describe the extent to which the objects are proximal to each other, and the height of the dendrogram indicates the distance between two objects or clusters. The final results of clustering algorithms can be obtained by cutting the dendrogram at a certain level as required (the dashed line in Fig. 2.6). The presentation of hierarchical structure provides an informative and visualized description. It is very effective when the hierarchical relations really exist in the data. Different from agglomerative methods, the divisive method is computationally expensive because in each step it has to consider $2^{N-1} - 1$ division possibilities of a cluster consisting of N data objects. Therefore, agglomerative methods are more commonly used for convenience.

Agglomerative Hierarchical Clustering

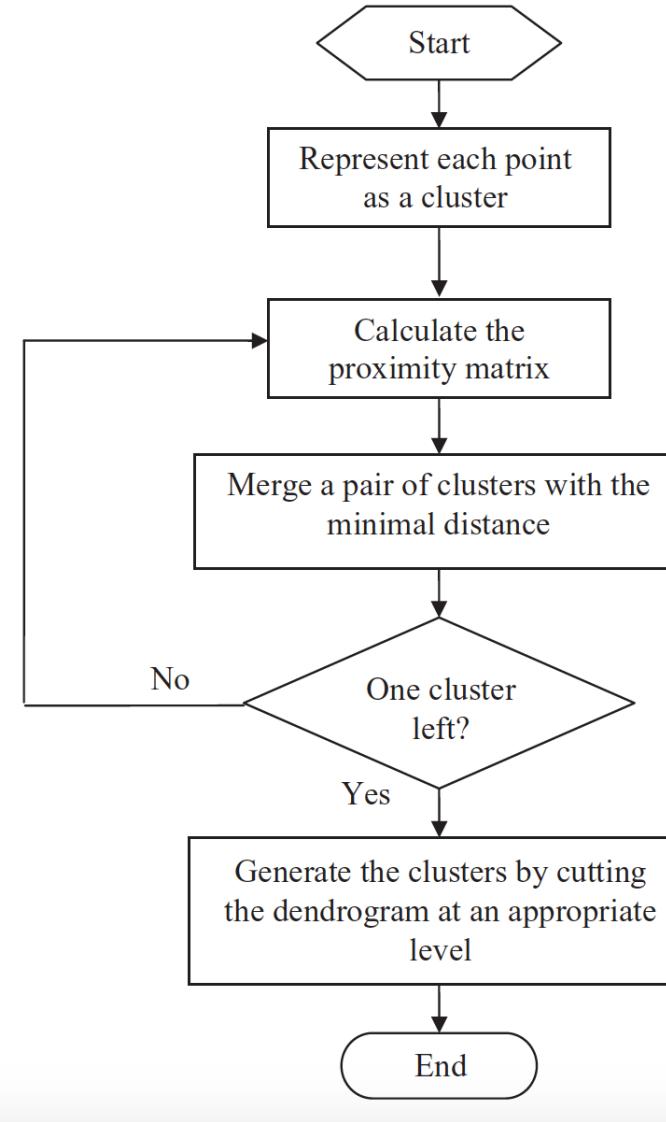


Figure 2.7: Flowchart of the agglomerative hierarchical clustering algorithm [44].

General Agglomerative Hierarchical Clustering Agglomerative clustering starts with N clusters, each one is made up of a singleton point. A series of recursive merging procedures are operated that eventually all data points are merged into one cluster. Fig. 2.7 depicts the procedures of general agglomerative method [41]. Algorithm 1 demonstrates the procedures of the agglomerative method. An alternative stopping condition for this algorithm can also depend on the number of clusters required.

Algorithm 1: Agglomerative hierarchical clustering algorithm

-
- 1 Assign each node to a cluster. Compute the distance between each pair of clusters.
 - 2 Find the closest pair of clusters and merge them into one cluster.
 - 3 Compute distances between the new cluster and each of the old clusters.
 - 4 Repeat step 2 and 3 until the minimum distance is larger than a predefined value.
-

Table 2.4: Different schemes of coefficients values $\alpha_i, \alpha_j, \beta$, and γ in (2.7). n_i, n_j and n_l are the number of data points in cluster C_i, C_j , and C_l , respectively [40, 42, 52].

Clustering algorithms	α_i	α_j	β	γ
Single linkage (nearest neighbor)	1/2	1/2	0	-1/2
Complete linkage (farthest neighbor)	1/2	1/2	0	1/2
Group average linkage (UPGMA)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted average linkage (WPGMA)	1/2	1/2	0	0
Median linkage(WPGMC)	1/2	1/2	-1/4	0
Centroid linkage(UPGMC)	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i n_j}{(n_i+n_j)^2}$	0
Ward's method	$\frac{n_i+n_l}{n_i+n_j+n_l}$	$\frac{n_j+n_l}{n_i+n_j+n_l}$	$\frac{-n_l}{n_i+n_j+n_l}$	0

Clustering Linkage It is the definition of distance that determines the merging procedures of clusters. There are a variety of definitions of the distance between a cluster C_l and a new cluster $C_{(ij)}$ merged by cluster C_i and C_j , which can be generally formulated by Lance and Williams [48].

$$D(C_l, (C_i, C_j)) = \alpha_i D(C_l, C_i) + \alpha_j D(C_l, C_j) + \beta D(C_i, C_j) + \gamma |D(C_l, C_i) - D(C_l, C_j)|, \quad (2.7)$$

where $D()$ is the distance function and $\alpha_i, \alpha_j, \beta$, and γ are the coefficients whose value is dependent on the specific schemes utilized. The values of these coefficients are summarized in Table 2.4, which are also given in [37, 39, 49].

- The single linkage algorithm. For single linkage, the distance between the merged cluster C_{ij} and the third cluster C_l depends on the closest one of the two clusters C_i and C_j to the third one C_l . Therefore, it is also called the nearest

neighbor linkage and following the parameters in Table 2.4. (2.7) becomes:

$$D(C_l, (C_i, C_j)) = \min(D(C_l, C_i), D(C_l, C_j)). \quad (2.8)$$

Single linkage method tends to generate elongated clusters. It may cause the problem that clusters with different properties join together. However, if clusters are far away from each other, the single linkage method works well.

- The complete linkage algorithm. In contrast to single linkage method, complete linkage computes the distance based on the maximum one between the two clusters C_i , C_j and the third cluster C_l . and (2.7) becomes

$$D(C_l, (C_i, C_j)) = \max(D(C_l, C_i), D(C_l, C_j)). \quad (2.9)$$

It is effective in uncovering small and compact clusters.

- The group average linkage algorithm. The distance is defined as the average of the distance between all pairs of data points, (2.7) becomes

$$D(C_l, (C_i, C_j)) = \frac{1}{2}(D(C_l, C_i) + D(C_l, C_j)). \quad (2.10)$$

The distance is the average of $D(C_l, C_i)$ and $D(C_l, C_j)$.

- The weighted average linkage algorithm. Similar with the group average method, the calculation of the distance also takes the average. The difference is that two weights are given to them based on the number of nodes included in two clusters. (2.7) becomes

$$D(C_l, (C_i, C_j)) = \frac{n_i}{n_i + n_j} D(C_l, C_i) + \frac{n_j}{n_i + n_j} D(C_l, C_j). \quad (2.11)$$

- The centroid linkage algorithm. The distance between the centroids of two clusters determines whether they merge or not. The distance is defined as below,

$$m_i = \frac{1}{n_i} \sum_{x \in C_i} x, \quad (2.12)$$

where n_i is the number of points in cluster C_i . (2.7) becomes,

$$D(C_l, (C_i, C_j)) = \frac{n_i}{n_i + n_j} D(C_l, C_i) + \frac{n_j}{n_i + n_j} D(C_l, C_j) - \frac{n_i n_j}{(n_i + n_j)^2} D(C_i, C_j). \quad (2.13)$$

This definition is equivalent to the calculation of the squared Euclidean distance between the centroids of the two clusters,

$$D(C_l, (C_i, C_j)) = \|m_l - m_{(ij)}\|^2. \quad (2.14)$$

- The median linkage algorithm. The median linkage algorithm is similar with the centroid linkage, except that equal weight is given to the merged two clusters. (2.7) becomes

$$D(C_l, (C_i, C_j)) = \frac{1}{2} D(C_l, C_i) + \frac{1}{2} D(C_l, C_j) - \frac{1}{4} D(C_i, C_j) \quad (2.15)$$

The median linkage can be regarded as a unique case of centroid linkage when there are an equal number of nodes in two merged clusters.

- Ward's method, also known as the minimum variance method. The Ward's method aims to minimize the increase of the within-class sum of the squared errors,

$$E = \sum_{k=1}^K \sum_{x_i \in C_k} \|x_i - m_k\|^2, \quad (2.16)$$

where K is the total number of clusters and m_k is the centroid cluster C_k as defined in (2.12), after the merging of two selected clusters. The change only occurs to the merged cluster while all the others remain the same. Therefore, the computation only considers the merged clusters, and can be expressed as

$$\Delta E_{ij} = \frac{n_i n_j}{n_i + n_j} \|m_i - m_j\|^2, \quad (2.17)$$

(2.7) then becomes

$$\begin{aligned} D(D_l, (C_i, C_j)) &= \frac{n_i + n_l}{n_i + n_j + n_l} D(C_l, C_i) + \frac{n_j + n_l}{n_i + n_j + n_l} D(C_l, C_j) \\ &\quad - \frac{n_l}{(n_i + n_j)^2} D(C_i, C_j). \end{aligned} \quad (2.18)$$

The above mentioned methods can be divided into two types. The first type includes single linkage, complete linkage, and average linkage which considers all points of the merged two clusters when computing the distance. They are also called the graph methods. The other methods belong to the second type which uses a geometric centroid to represent the whole cluster when calculating the distance. And they are also called geometric methods.

Divisive Hierarchical Clustering On the contrary of agglomerative hierarchical clustering, the divisive hierarchical clustering operates in an opposite way. At the very beginning, all points belong to the same cluster. A procedure iteratively divides clusters until all clusters are singletons. Assuming a cluster has N points, there are $2^{N-1} - 1$ possible divisions of the cluster without any empty ones. This method can be very computationally expensive even when the cluster size is small which makes the divisive method the second option after the agglomerative one. However, the divisive method can provide a clearer view of the main structure since larger clusters are generated in the early stages and it's also less likely to have problems from the accumulated erroneous decisions, which cannot be corrected by the successive process [50]. Heuristic methods have been proposed, such as the algorithm DIANA (Divisive Analysis), based on the earlier work of [51], which only considers a part of all possible divisions.

According to [50], at each step, divisive clustering has some iterative steps to move the closer objects to the splinter group which is seeded with an object that is farthest from the others in the cluster to be divided. The divided cluster is the one who has the largest diameter which is defined as the largest distance between any pair of points in the cluster. Assuming cluster C_l is going to be divided into two clusters C_i and C_j . The iterative clustering procedures are described as below and summarized in Fig. 2.8 [41].

1. Start with C_i equal to C_l and C_j as an empty cluster.
2. For each data object $x_m \in C_i$,
 - a) For the first iteration, compute its average distance to all other objects:

$$d(x_m, C_i \setminus \{x_m\}) = \frac{1}{N_{ci} - 1} \sum_{x_p \in C_i, p \neq m} d(x_m, x_p) \quad (2.19)$$

- b) For the remaining iterations, compute the difference between the average distance

Table 2.5: Two features f_i and f_j with values of 0 and 1.

$f_j \backslash f_i$	0	1
0	n_{00}	n_{01}
1	n_{10}	n_{11}

to C_i and the average distance to C_j :

$$d(x_m, C_i \setminus \{x_m\}) - d(x_m, C_j) = \frac{1}{N_{ci} - 1} \sum_{x_p \in C_i, p \neq m} d(x_m, x_p) - \frac{1}{N_{cj}} \sum_{x_q \in C_j} d(x_m, x_q) \quad (2.20)$$

3. a) For the first iteration, move the data object with the maximum value into C_j .
- b) For the remaining iterations, if the maximum value of (2.20) is greater than 0, move the data object with the maximum difference into C_j . Repeat step 2 b) and 3 b). Otherwise stop.

In each step of division, it uses all features. Therefore, the divisive hierarchical clustering algorithm is also called polythetic. On the other hand, if it only uses one feature in each divisive step, then it is called monothetic which is usually used for data objects with binary features [50]. The selection of the feature for the cluster division depends on its similarity with other features, through the measures of association. Given the table 2.5 including two binary features f_i and f_j . The association can be calculated as follows:

$$a_1 = |n_{11}n_{00} - n_{10}n_{01}|, \quad (2.21)$$

$$a_2 = (n_{11}n_{00} - n_{10}n_{01})^2, \quad (2.22)$$

$$a_3 = \frac{(n_{11}n_{00} - n_{10}n_{01})^2 N}{(n_{11} + n_{01})(n_{11} + n_{10})(n_{00} + n_{01})(n_{00}) + n_{10}} \quad (2.23)$$

As an example, a_3 is a statistic that has approximately a chi-squared distribution with 1 degree of freedom [50]. The larger the value of a_3 is, the closer is the relation and the distance between the two variables.

Simulation Results of Hierarchical Clustering Algorithm Fig. 2.9 shows the results of hierarchical clustering algorithm on spatial traffic patterns with different

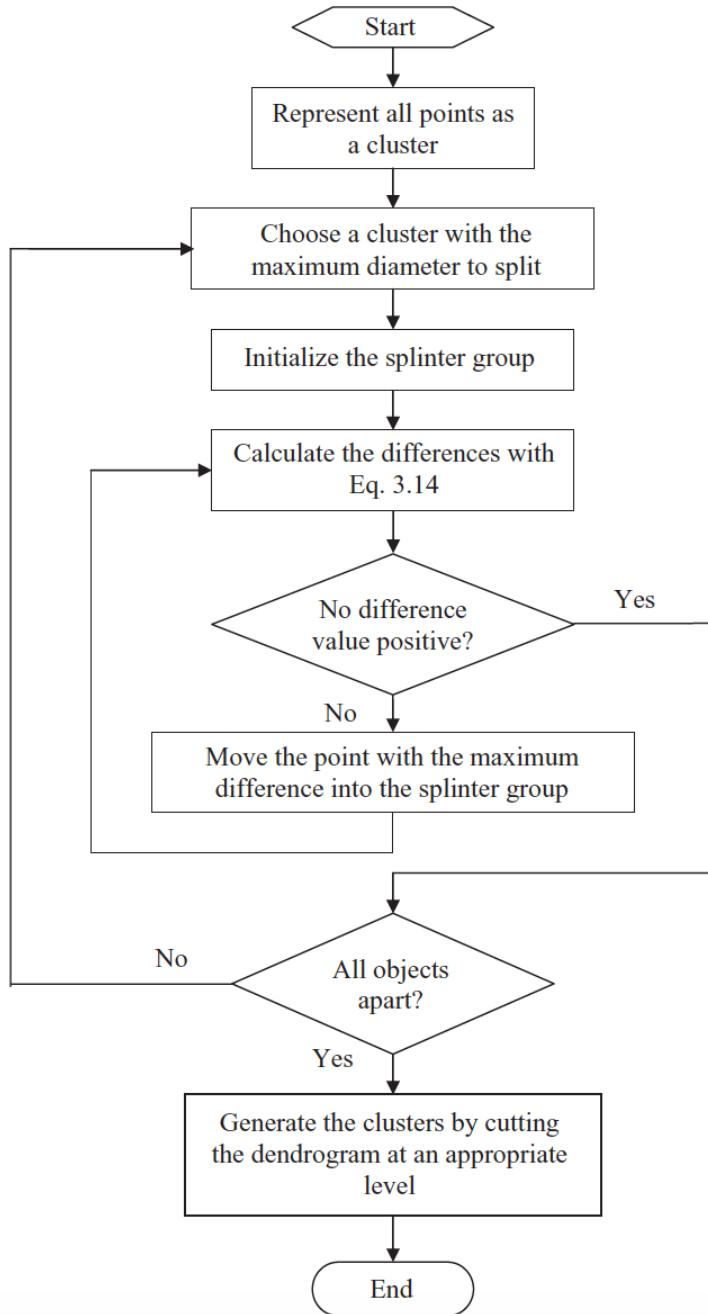


Figure 2.8: Flowchart of the divisive hierarchical clustering algorithm [44].

C_V values.

Lines are used to indicate cluster grouping. A group of nodes linking to the same point constitute a cluster, and that point is the CH, selected according to the best

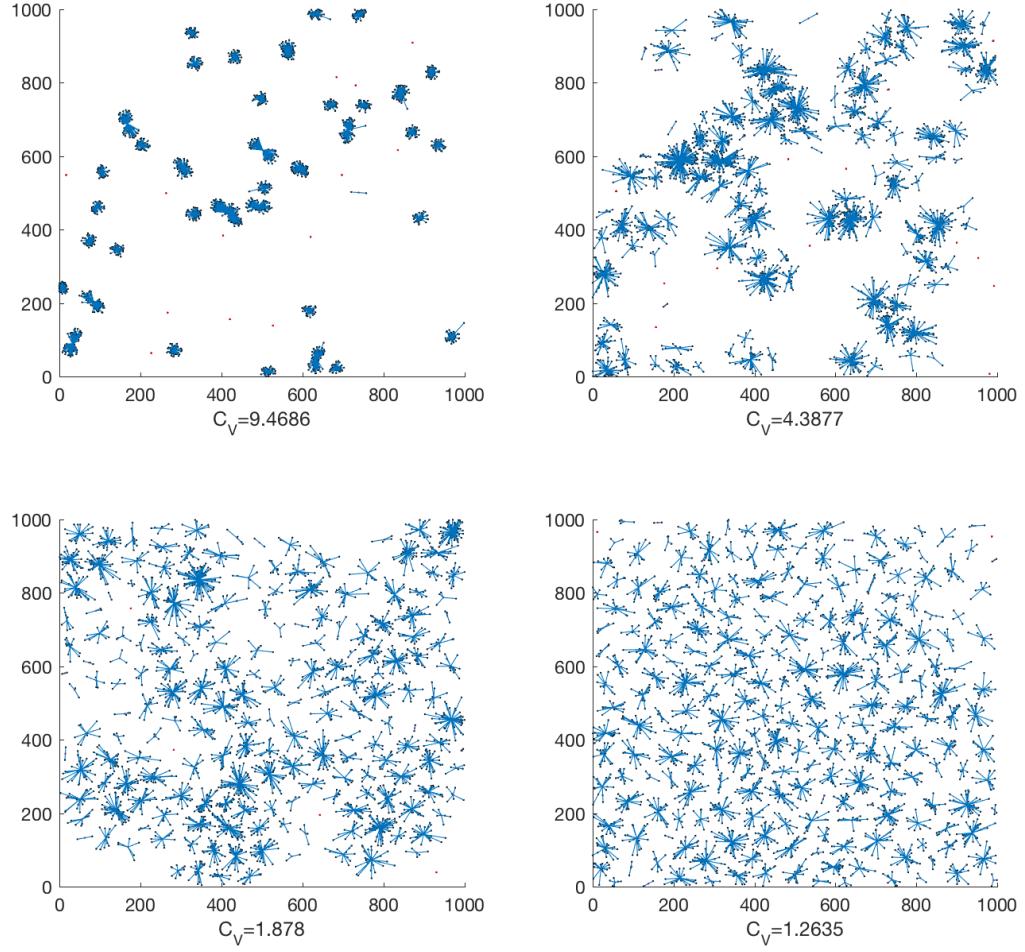


Figure 2.9: Simulation results of hierarchical clustering algorithm on spatial traffic patterns with different C_V values.

channel gain. In the result, we can also observe that an individual node can become a cluster itself when it is far away from all the other machines and becomes isolated. From the figure, it can be well seen that the hierarchical clustering algorithm works well for clustered traffic pattern.

Recent Advances There are some criticisms of classical hierarchical clustering algorithms such as the lack of robustness and their sensitivity to noise and outliers. Hierarchical clustering algorithms cannot correct misclassification because once a point

is classified into a cluster, it will not be considered again. The computational complexity is also a defect of this algorithm which is at least $\mathcal{O}(N^2)$ and cannot deal with large-scale data sets in data mining. To meet these requirements, some new hierarchical clustering algorithms are proposed such as BIRCH in [52], CURE in [53], ROCK in [54], and Chameleon in [55].

2.5.2.3 Partitional Clustering

In contrast to hierarchical clustering, partitional clustering classifies objects into K clusters with no hierarchical structure. This algorithm usually accompanies the optimization of a criterion function. Given a set of points $x_i \in \Re^d, i = 1, \dots, N$, partitional clustering algorithm classifies them into a certain number of clusters while maximizing or minimizing a prespecified criterion function J [56]. Theoretically, the optimization of criterion function J can be found by the enumeration of all clustering possibilities. However, this enumeration method is quite computationally expensive and impractical, as given by the formula [56]:

$$P(N, K) = \frac{1}{K!} \sum_{m=1}^K (-1)^{K-m} \binom{m}{k} m^N \quad (2.24)$$

It can be seen from (2.24) that even for a small-scale clustering problem, the computational complexity is too high. For example, in order to divide 30 objects into 3 clusters, the total number of division possibilities is approximately 2×10^{14} . Therefore an efficient optimization method is needed.

Clustering Criteria The goal of clustering is to divide objects into different clusters so that the objects belonging to the same cluster are homogeneous while the objects belonging to different clusters are well separated. The homogeneity and separation are defined through criterion functions. One of the common criterion function is the sum-of-squared-error criterion [39, 50].

Given a set of objects $x_j \in \Re^d, j = 1, \dots, N$, and we want to divide them into K

clusters C_1, \dots, C_K . The sum-of-squared-error function is defined as below,

$$\begin{aligned} J_S(\Gamma, M) &= \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} \|x_j - m_i\|^2 \\ &= \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} (x_j - m_i)^T (x_j - m_i), \end{aligned} \quad (2.25)$$

where, $\Gamma = \{\gamma_{ij}\}$ is a partition matrix, $\gamma_{ij}=1$ if $x_j \in$ cluster i , $\gamma_{ij} = 0$ otherwise, with $\sum_{i=1}^K \gamma_{ij} = 1 \forall j$; $M = [m_1, \dots, m_k]$ is the cluster prototype or centroid (means) matrix; and $m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \gamma_{ij} x_j$ is the sample mean for the i^{th} cluster with N_i objects.

The goal is to minimize the sum-of-squared-error criterion to obtain the optimal partition which is called minimum variance partition. The sum-of-squared-error criterion can well divide clusters that are compact and well separated. However, this criterion might be sensitive to outliers and may incorrectly divide a large cluster into small pieces, as indicated by [57].

The sum-of-squared-error criterion can also be derived from the scatter matrices defined in multi-class discriminant analysis [57].

$$S_T = S_W + S_B, \quad (2.26)$$

where

$S_T = \sum_{j=1}^N (x_j - m)(x_j - m)^T$ is the total scatter matrix;
 $S_W = \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} (x_j - m_i)(x_j - m_i)^T$ is the within-cluster scatter matrix;
 $S_B = \sum_{i=1}^K N_i (m_i - m)(m_i - m)^T$ is the between-cluster scatter matrix; and
 $m = \frac{1}{N} \sum_{i=1}^K N_i m_i$ is the total mean vector for the entire data set.

It can be seen that the total scatter matrix is independent of the partition while the other two matrices are related to each other. In the case of univariate analysis, the decrease of S_w is equivalent to the increase of S_B . The minimization of the sum-of-squared-error criterion is equivalent to the minimization of the trace of S_W , since,

$$tr(S_W) = \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} (x_i - m_i)^T (x_j - m_i) = J_s(\Gamma, M). \quad (2.27)$$

Therefore, it can also be derived that the minimization of the sum-of-squared-error criterion is also equal to the maximization of the trace of S_B , since $tr(S_T) = tr(S_W) + tr(S_B)$.

K-means Algorithm

K-means Algorithm *K*-means algorithm [58] is one of the well-known clustering algorithms [57, 59]. *K*-means aims to obtain an optimal partition by minimizing (2.25) with an iterative procedure. The *K*-means algorithm can be described as below:

- Initialize a *K*-partition randomly or based on some prior knowledge. Calculate the cluster prototype matrix $M = [m_1, \dots, m_K]$
- Assign each object in the data set to the nearest cluster C_l , i.e.,

$$x_j \in C_l, \text{ if } \|x_j - m_l\| < \|x_j - m_i\| \text{ for } j = 1, \dots, N, i \neq l, \text{ and } i = 1, \dots, K; \quad (2.28)$$

- Recalculate the cluster prototype matrix based on the current partition,

$$m_i = \frac{1}{N_i} \sum_{x_j \in C_i} x_j \quad (2.29)$$

- Repeat steps 2 and 3 until there is no change for each cluster.

The partition of step 2 is based on the nearest-neighbor principal and it is a Voronoi partition [60]. Space is separated into Voronoi cells based on the prototype Voronoi vectors. Each point in a Voronoi cell is closer to the corresponded Voronoi vector than any other vectors.

This algorithm follows a batch mode learning, since the new centroids are generated after the assignment of all data points. Correspondingly, the on-line or incremental mode *K*-means recalculate the centroids each time a data point is processed,

$$m^{new} = m^{old} + \eta(x - m^{old}), \quad (2.30)$$

where η is the learning rate.

Similar algorithms are proposed such as the Lloyd algorithm [61], which was further generalized for vector quantization, known as generalized Lloyd algorithm or Linde-Buzo-Gray algorithm [60, 62], used to compress data and signal. In the context of vector quantization, the prototype vectors are called code words constituting a code book.

K -means is quite popular in the application of clustering algorithms due to its ease of implementation. For many practical problems, it works very well, particularly when the clusters are compact in shape. The computation complexity of K -means is $\mathcal{O}(NKdT)$, where T is the number of iterations. K, d and T are usually much smaller than N , therefore N dominates the complexity calculation which makes it an approximate linear function. The complexity of K -means is much less than hierarchical clustering which makes it more suitable for big data processing. Additionally, several methods are proposed to speed up K -means algorithm [63, 64]. A spherical K -means is proposed to address sparse data objects, particular for documents clustering [65].

There are some disadvantages of K -means as well particularly the inherent limitations when the optimization are done through the hill-climbing methods. A lot of effort are made to deal with these drawbacks and many variants of K -means are proposed to address these issues.

Advancement of K -means *Convergence and Initial Partition* The final centroids obtained by the iterative K -means procedures may not be the global optimum [66]. Since the optimization of K -means algorithms can drop to a local optimum, the selection of the initial points plays a very important role in leading to a better convergence centroids because different selections may result in different local minimum which represents different partition solutions. However, there are no efficient and universal methods to determine the initial points in the literature so far.

Stochastic optimal search techniques, such as simulated annealing and genetic algorithm, can find a global optimum effectively. In [67], it proposes the genetic K -means algorithm to have a fast convergence. In [68], ELBG algorithm utilizes a roulette mechanism to achieve the near-optimum; therefore it is not sensitive to the initialization. In [69], it compares the reference random initialization methods with three other initial partition methods proposed by [50, 58, 70] from the perspectives of effectiveness, robustness, and convergence speed criteria. In [58], Forgy's randomly select K points as prototypes and then clustering the remaining points based on their proximity. Similarly in [70], MacQueen's method also randomly selects K points from the given set, and the difference is that it works in an online mode, therefore, the presentation order of points also affect the results. In [50], Kaufman and Rousseeuw's method locates the centroids one by one. The first centroid is located at the position

where it has the smallest sum of distance with all the other points. The second centroid is located at the position where it can give rise to the most decrease in the cost function. The other centroids are located with the same principal one by one until all K centroids are determined. The four methods are compared on the same set of data samples. The results show that in terms of effectiveness and sensitivity, the random method and Kaufman and Rousseeuw's method work much better than the other two while considering convergence speed, MacQueen's method overtake the others. In summary, the Kaufman and Rousseeuw's method is most recommended.

In [71], Likas proposed a successive K -means clustering procedures with the number of clusters varying from 1 to K . First a centroid is selected among all given points assuming there is only one cluster, then at each iteration k , $k = 2, \dots, K$, the previous $k - 1$ centroids remain the same and the new centroid is selected considering all data points which are regarded as initial centroids. It is claimed that this method is irrelevant with the initial partitions and can provide accelerating strategies. The actual number of clusters can also be estimated by the intermediate results. There is a defect of this algorithm that the computation complexity is very high, because for each iteration, it has to execute K -means N times.

The Number of K K -means algorithm need to know the value of K beforehand. However, in most cases, it is impractical. Similar to the decision of initialization of centroids, there are no efficient or global methods to determine the value of K . Therefore the selection of K becomes a very import issue in the K -means algorithm [72].

Ball and Hall [73] develops the ISODATA (Iterative Self-Organizing Data Analysis Technique) algorithm to dynamically estimate the value of K . ISODATA adjusts the number of clusters by merging and splitting clusters according to some predefined thresholds. The merging of two clusters depends on the distance of their centroids and the splitting depends on the within-class variability. The new K is used for the next iteration of clustering.

For the cases that K is increased, a new cluster is needed when no existing cluster can represent the new object. This can be done with a threshold. Specifically, the cluster C_l is not only the nearest cluster to the new data object x_j , but also follows a constraint, i.e.

$$\|x_j - m_l\| \leq \rho \quad (2.31)$$

where ρ is the threshold. It can be seen that as ρ decreases, there will be more clusters and few points in each cluster, and vice versa. If (2.31) is not satisfied, it generates

a new cluster to represent the data.

Robustness K -means is sensitive to outliers and noise. Even if a point is far away from the cluster centroids, it will also be considered when calculating the new centroids. Therefore some noise or outlier points will distort the shape of the cluster.

ISODATA [73] and PAM [50] both consider the effect of outliers in K -means algorithm. ISODATA discards the clusters inside which the number of points is less than a predefined threshold. And the splitting mechanism in this method avoids generating elongated clusters. PAM method selects real point as the medoids instead of generating a cluster centroid by calculation, in which it eliminates the effect of outliers to the resulting prototypes. Medoid is the point who has the minimum average distance with all other points in the same cluster. Inspired by the same strategy, a K -medoids based algorithm is proposed in [74]. It uses discrete medians as the cluster centroids. The time complexity of this algorithm is $\mathcal{O}(N \log N)$ and it's slower than K -means algorithm.

Extension of the Definition of Means The definition of means limits the application of K -means only to the numerical variables. Moreover, the calculated means may not have a physical interpretation. The aforementioned medoid method is a natural choice when the computation of means is not available because the medoids always exist [50].

[75] discussed K -means in binary data clustering and suggested three variants. The categorical data can also be represented by binary data. [76] and [77] defined different dissimilarity measures to extend K -means to categorical variables. In [76], Huang defines the distance between two categorical data object as the number of their mismatches features,

$$D(x_i, x_j) = \sum_{i=1}^d \delta(x_{il}, x_{jl}), \quad (2.32)$$

where

$$\delta(x_{il}, x_{jl}) = \begin{cases} 1 & x_{il} \neq x_{jl} \\ 0 & x_{il} = x_{jl} \end{cases} \quad (2.33)$$

or the weighted total number of mismatches,

$$D(x_i, x_j) = \sum_{i=1}^d \frac{n_{x_{il}} + n_{x_{jl}}}{n_{x_{il}} n_{x_{jl}}} \delta(x_{il}, x_{jl}), \quad (2.34)$$

where $n_{x_{il}}$ and $n_{x_{jl}}$ are the frequencies of the occurrence of x_{il} and x_{jl} .

Mathematically, The goal of clustering is to minimize the cost function as below:

$$J(\Gamma, Q) = \sum_{i=1}^K \sum_{j=1}^N \gamma_{ij} D(x_j, Q_i), \quad (2.35)$$

where Q is a set of d -dimensional vectors $Q = \{Q_1, \dots, Q_K\}$, and $Q_j = (Q_{j1}, \dots, Q_{jd})$, which represents the clusters instead of the traditional mean centroids. Each Q_j is defined to minimize the sum of distances:

$$J(Q) = \sum_{i=1}^N D(x_i, Q_j). \quad (2.36)$$

Simulation Results of K-means Algorithm Fig. 2.10 shows the output of standard K -means algorithm to heterogeneous geometry with different C_V values. It can be seen from the result that basically K -means algorithm can classify data points into different clusters according to their proximity. The execution speed is faster than hierarchical clustering algorithm. However, the drawbacks are obvious as well. Some clusters in the result are supposed to merge into one cluster because they are close enough to each other. But the K -means algorithm force them to be split into different clusters to achieve the number of K clusters. Besides, it cannot distinguish outliers. The noise nodes are also forced to join a cluster which makes the shape and centroid of the cluster distorted.

2.5.2.4 DBSCAN Algorithm

Density-based spatial clustering of applications with noise (DBSCAN) is a commonly used algorithm for clustering application. In this algorithm, it regards that dense regions of data objects are separated by sparse regions of low density (noise). The shape of these regions can be arbitrary. The key parameters in this method are density and connectivity both associated with the number of neighbors. The main reason why this algorithm works is that the density within a cluster area is higher than that of the outside. And for the area of noise points, the density is also quite low.

There are two crucial concepts in DBSCAN: *density reachability* and *density connectivity*. The two concepts are associated with two input parameters: the size of

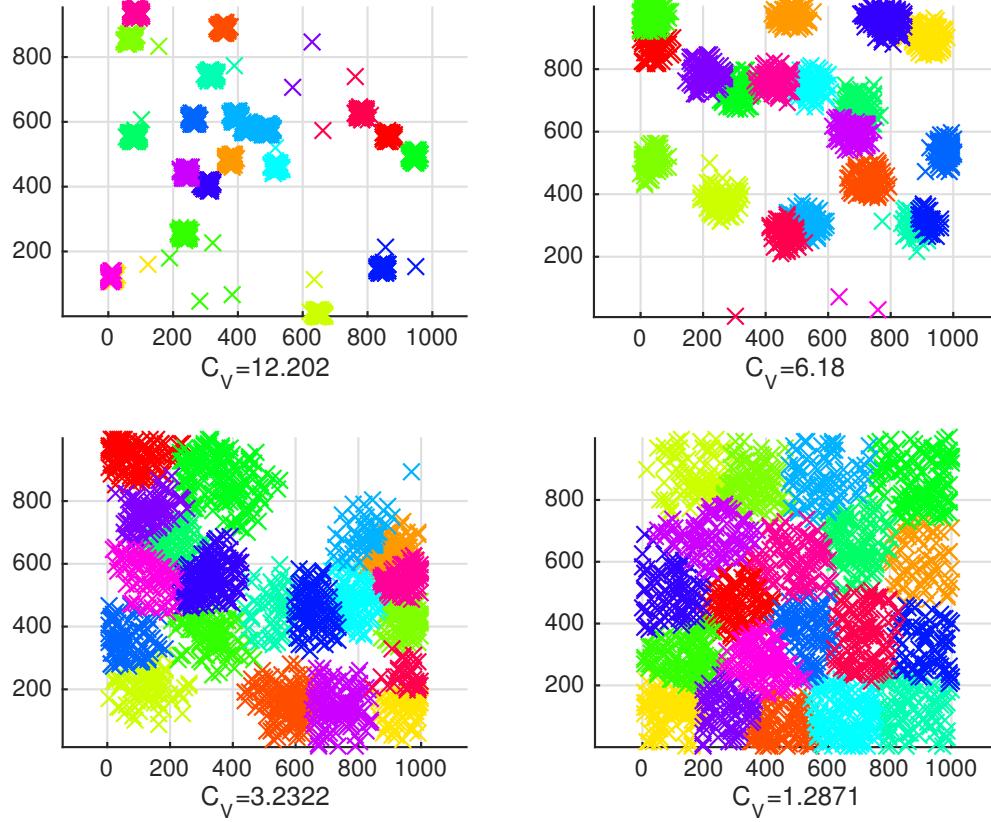


Figure 2.10: Simulation Results of K -means algorithm on heterogeneous geometry with different C_V values.

epsilon neighborhood (ϵ) and the minimum points needed within the epsilon neighborhood (m). The main principle for DBSCAN is that, for a point, only if its neighborhood of a given radius ϵ contains a minimum number of m points, they can be classified into the same clusters. In other words, the density in the neighborhood has to exceed a predefined threshold.

Density reachability decides whether two points belong to the same cluster. A point p_1 is regarded as directly reachable from p_2 if two conditions are satisfied:

1. p_1 is within the circle of p_2 with radius of ϵ ,
2. the number of points within the epsilon neighborhood of p_2 is over m .

Fig. 2.11 is an example demonstrating the concept of *Density reachability*. Assum-

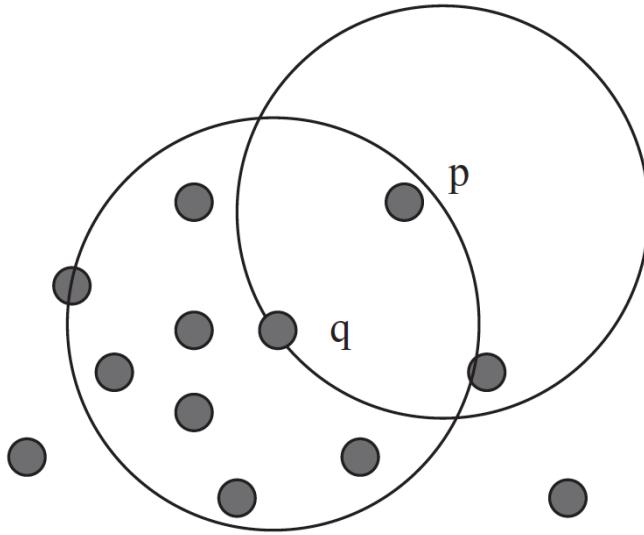


Figure 2.11: Neighborhood ϵ for points p and q .

ing the threshold of the minimum number of points m is 6. It can be seen that p is contained in the epsilon neighborhood of q , and the number of points within the neighborhood is over 6. Both two conditions are satisfied, therefore point p is regarded directly reachable from point q . However, it is not true the other way around because the number of points in the epsilon neighborhood of p is less than the required threshold 6, therefore q is not directly reachable from p . *Density connectivity* is next building step of DBSCAN. Point p_0 and p_n is density connected if there is a path p_0, \dots, p_n where each p_{i+1} is directly density reachable from p_i .

There are three types of points in DBSCAN algorithm, the core points, border points and noise points.

- A point is a core point if at least m points are within distance ϵ of it (including p).
- A border point is reachable from a core point but it is not a core point.
- A noise point is not reachable from any other points.

Figure 2.12 is an example to show the three types of points.

The result is dependent on the selection of parameters ϵ and m . However, there is no easy method to know this information in advance for different cluster pattern.

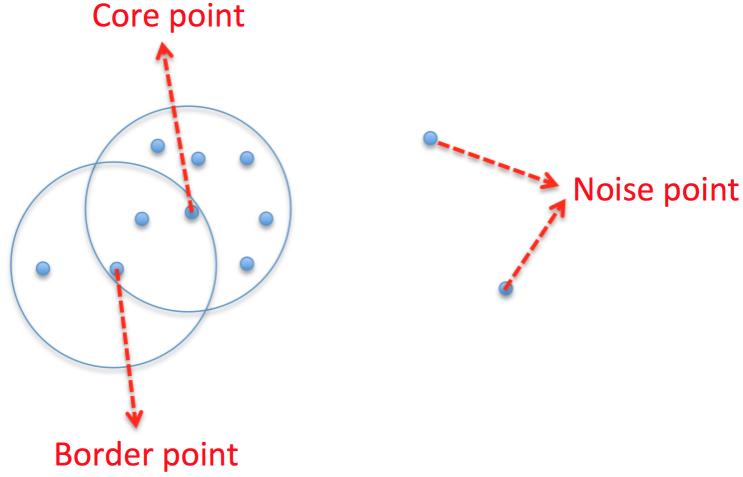


Figure 2.12: Examples of core, border, and noise points in DBSCAN algorithm.

Therefore, DBSCAN method uses global values for ϵ and m for all clusters. From the numerical results, it also indicates that the output of DBSCAN algorithm when $m > 4$ does not differ a lot from the case that $m = 4$, but the computation burden increase a lot. Therefore we usually set $m = 4$ for low-dimensional databases. The procedures of DBSCAN algorithm can be described as follows:

- Arbitrarily select a point p .
- Retrieve all points density-reachable from p with respect to ϵ and m .
- If p is a core point, a new cluster is formed or existing cluster is extended.
- If p is a border point, no points are density-reachable from p , and DBSCAN visits the next point of the database.
- Continue the process with other points in the database until all of the points have been processed.
- Since global values for ϵ and m are used, DBSCAN may merge two clusters into one cluster, if two clusters of different density are “close” to each other. They are close if the distance between clusters is lower than ϵ .

Fig. 2.13 shows the result of DBSCAN algorithm on the spatial traffic patterns with different C_V values. It can be seen that DBSCAN works well for clustered

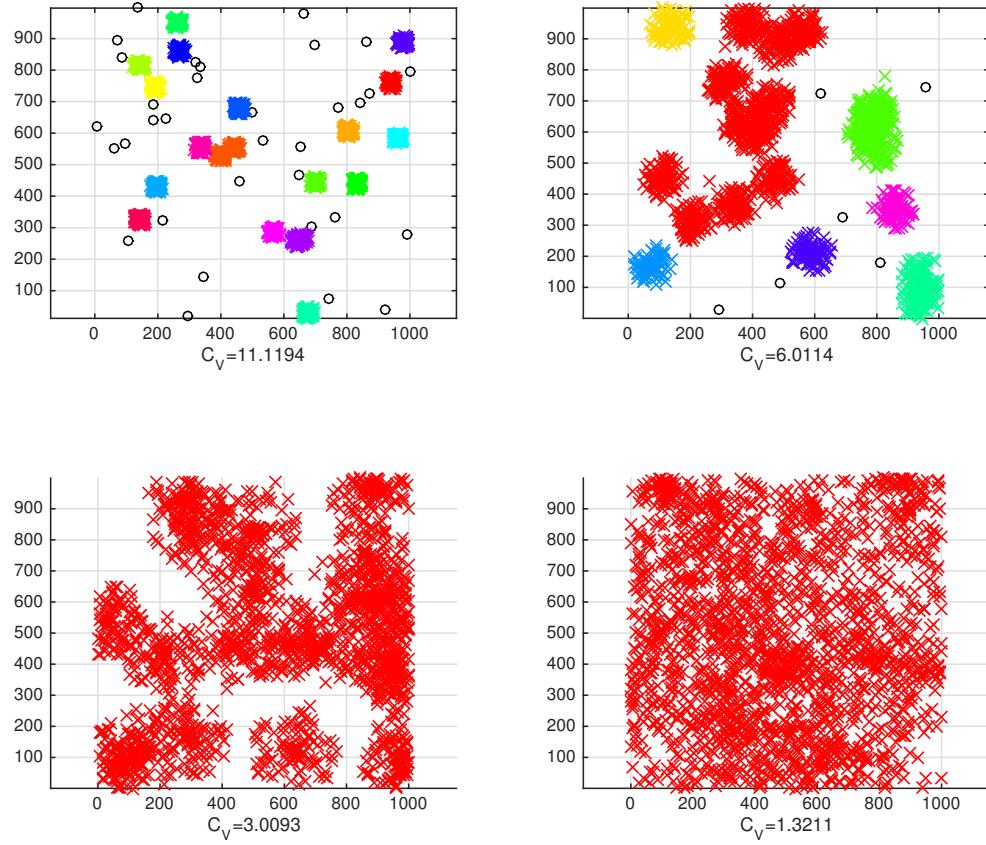


Figure 2.13: Simulation results of DBSCAN algorithm on spatial traffic patterns with different C_V values.

traffic pattern. Different clusters can be distinguished and noise points can also be recognized.

The main advantage of the DBSCAN algorithm are as follows:

1. Unlike K -means algorithm, DBSCAN does not need the information of cluster number in advance.
2. DBSCAN can recognize arbitrarily shaped clusters.
3. DBSCAN can recognize noise points and outliers.
4. DBSCAN only requires two input parameters and is insensitive to the ordering of the data points.

There are also some drawbacks of DBSCAN. The computation complexity of DBSCAN algorithm is quite high with time complexity of $\mathcal{O}(n \times \log n)$. The calculation of distance is based on Euclidean distance, and the algorithm may find the curse of dimensionality problem for high-dimensional data sets. Hence, DBSCAN algorithm is usually used for low-dimensional applications.

2.5.3 Choice of Clustering Algorithm

In this thesis, we decide to use the agglomerative hierarchical clustering algorithm due to the following reasons:

1. The agglomerative hierarchical clustering algorithm provides a good result for most of the cases.
2. The number of clusters K is not predetermined.
3. The hierarchical clustering algorithm can still form some clusters when the locations are homogeneous, while DBSCAN can only generate one cluster in this case.
4. Within the hierarchical clustering algorithms, the agglomerative method computes much faster than the divisive method.

Chapter 3

Proposed System Model

3.1 3GPP-Compliant LTE Random Access Simulator

We build a simulator of LTE one-hop random access procedure based on the parameters from 3GPP [6, 78]. The parameters are presented in Table 3.1. In our simulator, the random access procedure is run subframe by subframe. Fig. 3.1 shows the flow chart of our simulator in each subframe. Table 3.2 shows the consistency between our simulation and 3GPP [6].

Our comprehensive simulator tracks the states of all the machines in each subframe, and Fig. 3.2 gives an example of the track of a machine's (ID 235) states from the beginning to the completion of the random access procedure. The machine experiences all the possible states in the procedure. It can be seen that the machine transmits Msg1 for the first time at subframe 6827, and before that it remains at the idle state. Msg1 has a collision and the machine backs off for a random period (subframe 6833) after the response window (subframe 6828–6832). Then it turns active (subframe 6834–6836) and retransmits Msg1 at the next access subframe 6837. After that it waits for a period of Msg1 detection, Msg2 transmission and processing (subframe 6838–6845). Msg3 is transmitted at subframe 6846. It fails due to channel fading and the machine waits for a contention window period (subframe 6847–6894) before the retransmission of Msg3 at subframe 6895. It succeeds this time and waits for a period of Msg3 processing and Msg4 transmission (subframe 6896–6900), after which it completes the entire random access procedure. Fig. 3.3 shows the states of four machines within the same period from subframe 4620 to subframe 4740. It can

be seen that they all start from the idle state and end with the completion state. Machine 704 and 944 (red and blue) send access request simultaneously at subframe 4647, and they select the same preamble to transmit. Therefore, there is a collision and neither of them get through. They are both followed with a response window state and a backoff state. But the period of their backoff state are different. Then they are back to the active state and retransmit the access request at subframe 4667 and 4672 respectively. The blue machine also experiences a Msg3 failure and spans a long period of contention window state.

Table 3.1: Simulation parameter setup [6, 27, 28].

Symbol	Parameter	Value
B	Cell bandwidth	5 MHz
-	PRACH configuration index	6
$N_{preamble}$	Total number of preambles	54
$Maxpreamble$	Maximum number of preamble transmissions	10
-	Number of CCEs allocated for PDCCH	16
-	Number of CCEs per PDCCH	4
T_{RAR}	Ra-Response WindowSize	5 ms
T_{CR}	Mac-contentionResolutionTimer	48 ms
BI	BackoffTimer	20 ms
$P_{MSG3coll}$	Probability of successful delivery for both MSG3 & MSG4 (non-adaptive HARQ)	90%
$N_{MSG3max}$	Maximum number of MSG3 transmissions	5
-	Number of MTC devices	5k, 10k, 30k
-	Number of available subframes over the distribution period	10k, 60k
-	Period of PRACH opportunities	5 ms
T_{MSG1}	MSG1 transmission time	1 ms
T_{MSG2}	Preamble detection at eNodeB & MSG2 trans. time	3 ms
T_{MSG3}	Device processing time before sending MSG3	5 ms
$T_{TransMSG3}$	MSG3 transmission time	1 ms
T_{MSG4}	Time of processing MSG3 & sending MSG4	5 ms
T_{Tx}	Time of packet transmission in slotted ALOHA	1 ms
T_{RESP}	Response window size in slotted ALOHA	5 ms
T_{Rx}	Time of packet processing & acknowledgement transmission time	3 ms
P_{idle}	Power consumption in idle state	0.025 mW [79]
P_{Rx1}	Power consumption of processing and Rx in RACH	50 mW [79]
P_{Tx1}	Power consumption of Tx in RACH	50 mW [79]
P_{Rx2}	Power consumption of processing and Rx in slotted ALOHA	25 mW [80]
P_{Tx2}	Power consumption of Tx in slotted ALOHA	25 mW [80]

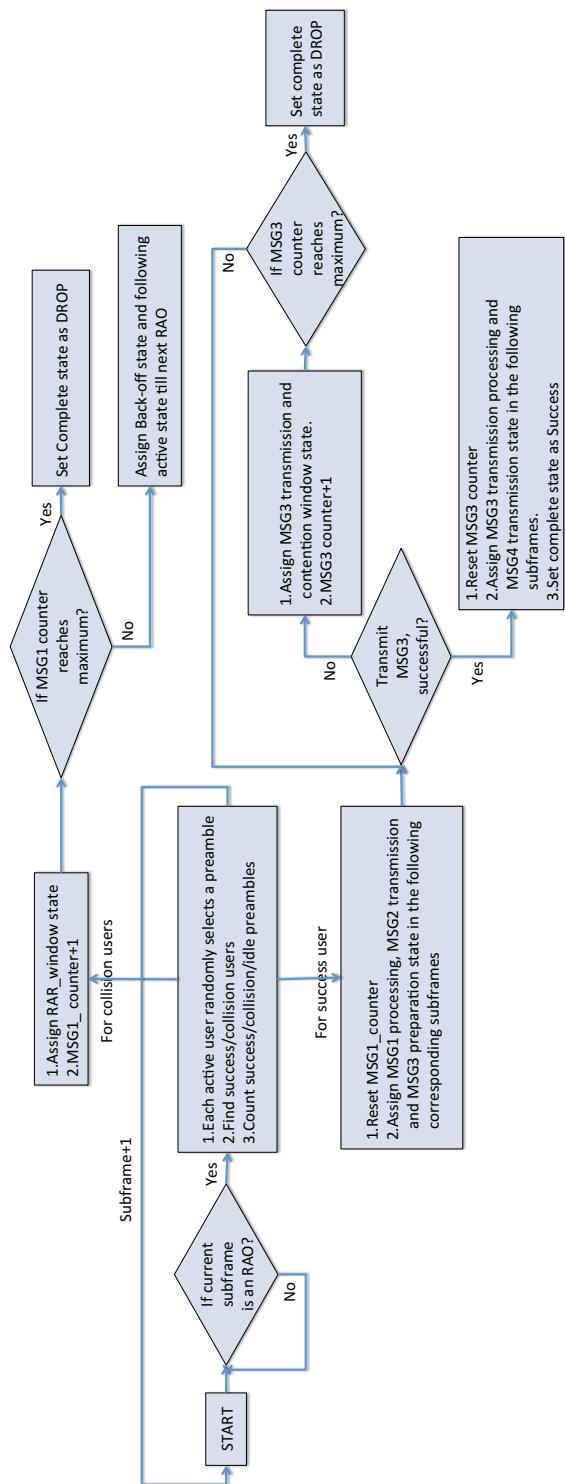


Figure 3.1: Flowchart of LTE random access procedure in one subframe.

Table 3.2: Validation of LTE random access simulator with 3GPP results [6].

Performance Measures	Number of MTC devices			
	5k	10k	30k	Result Origin
Collision Probability (%)	0.01	0.03	0.22	3GPP [6]
	0.01	0.03	0.23	Simulation
Success Probability (%)	1.09	2.18	6.49	Simulation
Idle Probability (%)	98.90	97.79	0.93	Simulation
Access Success Probability (%)	100	100	100	3GPP [6]
	100	100	100	Simulation
Average Access Delay (ms)	25.60	26.05	27.35	3GPP [6]
	28.23	28.58	29.63	Simulation
Average Preamble Transmission (%)	1.43	.145	1.50	3GPP [6]
	1.43	1.45	1.50	Simulation

0 : Idle state 1 : Active state or preamble transmission

2 : Msg1 Response window

3 : Msg1 backoff

4 : Msg1 detection, Msg2 transmission and processing

5 : Msg3 transmission and processing, Msg4 transmission

6 : Access completion

5 : Msg3 transmission and pr
7 : Msg3 contention window

Figure 3.2: An example of the track of a machine's (ID 235) states in our simulator

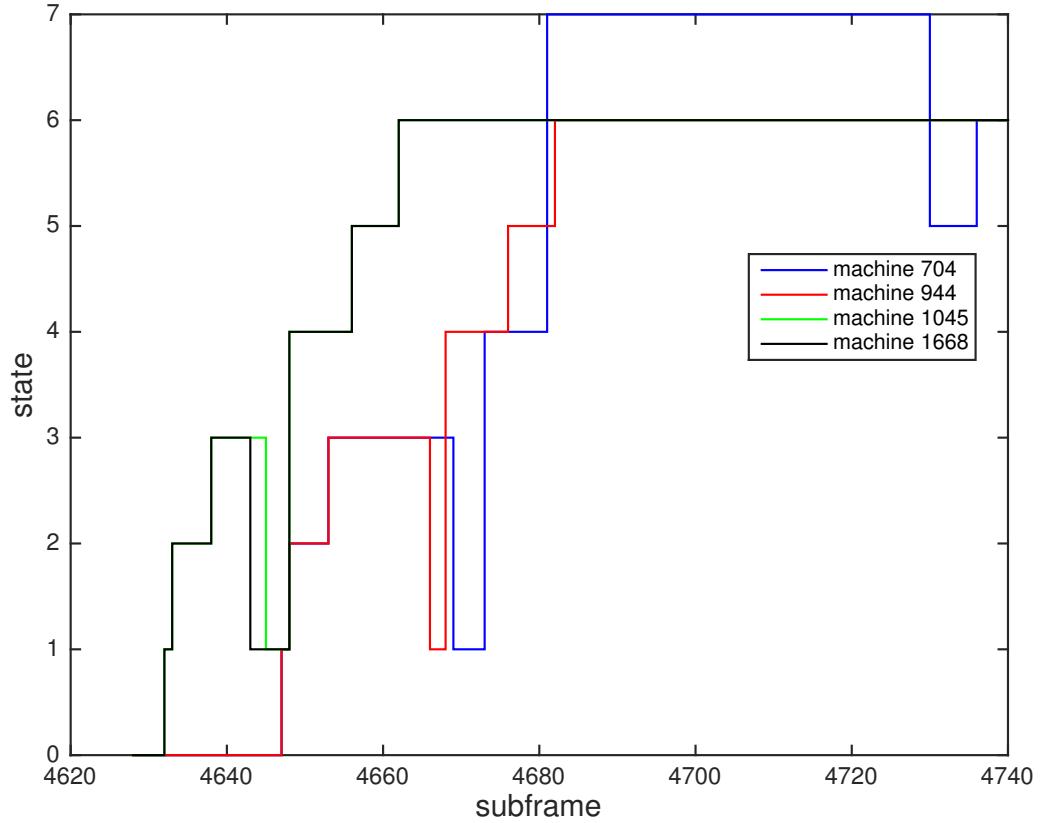


Figure 3.3: The states of four machines (ID 704, 944, 1045, 1668) from subframe 4620 to subframe 4740.

3.2 Location Generator

In our proposed scheme, we introduce a clustering location generator. In our generator, there are two types of machines. A machine is either a member of a cluster or is an independent node isolated from any clusters. Our location generator has four inputs, namely:

- number of devices (N),
- radius of cluster (R),
- number of clusters (M),

- proportion of isolated nodes ($P_{isolated}$).

Algorithm 2 demonstrates our procedure of generating heterogeneous locations for machines. Fig. 3.4 shows a realization of our clustering location generator.

Algorithm 2: Algorithm for clustering location generator

- 1 Generate M cluster centroids $\{C_i\}$, $i = 1, \dots, M$ through a Poisson Point Process.
 - 2 Generate N points:
 - For each point, uniformly selects a random variable $X \sim \mathcal{U}(0, 1)$.
 - If $0 < X \leq P_{isolated}$, randomly choose a location for the point.
 - If $P_{isolated} < X \leq 1$,
 - randomly select a centroid C_k , $k \in [1, M]$.
 - randomly locate the point within a circle with radius R and center C_k .
-

3.3 Slotted ALOHA-Based Cluster Random Access

After obtaining the machine locations, we use a hierarchical clustering algorithm to cluster the machines and select a CH in each cluster based on the channels between the machines and the BS. In the time domain, we assume that each machine has one packet to transmit and the arrival time is randomly distributed over the whole random access period (10 s). There are two hops of communication in our model. In the first hop, CMs upload packets to their CH through slotted ALOHA communication, and in the second hop, once the buffer of the CH reaches a certain level, the CH performs a random access procedure to set up the connection with the BS. Fig. 3.5 shows the details of random access procedures between machine and BS, and slotted ALOHA procedures between CM and CH. We assume all packet transmissions can be finished within one subframe, and if a transmission fails, the device backs off for a random period before the retransmission. It is also assumed that the buffer in CH has no

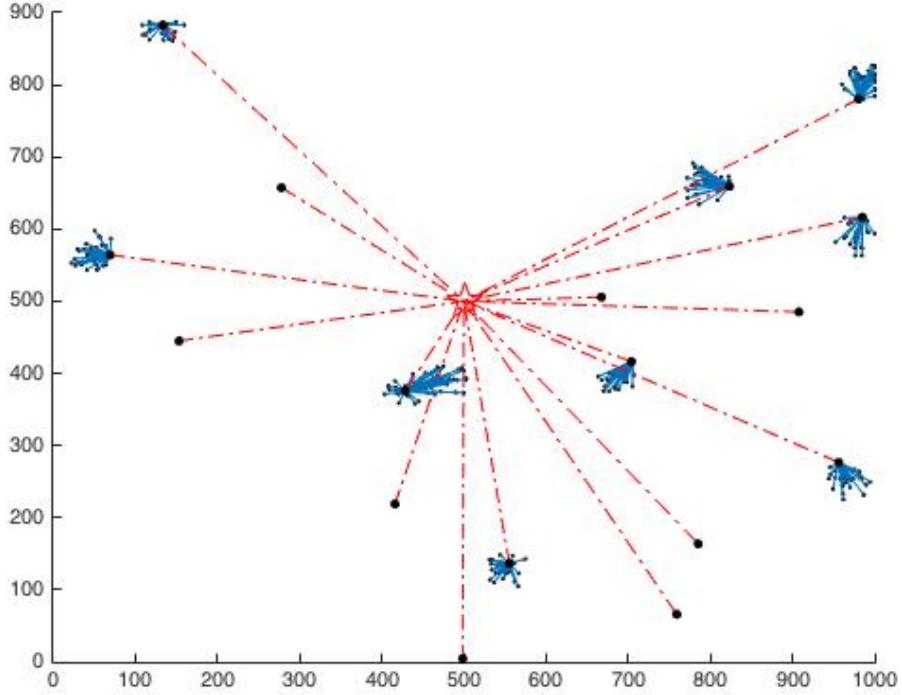


Figure 3.4: Clustering geometry model. The blue links are the slotted ALOHA communication between the CM and the CH, and the red links are the LTE random access procedure between CH and BS. The inputs of our location generator are $N = 200$, $M = 10$, $R = 25$, and $P_{isolated} = 0.01$.

limit so that the uploaded packets cannot be dropped by the CH due to insufficient storage. The resources (slots) for slotted ALOHA are migrated from the original RACH resources in order to keep a fair comparison with the reference method. From the original resources, we allocate one subframe for RACH every 100 RAOs. All the clusters fully reuse the resources for slotted ALOHA communication. Hence, interference exists, and it can arise from both inside and outside the cluster. We assume the packet can be successfully decoded by the CH only if the SINR is greater than 20 dB.

The SINR is calculated on the basis of the M2M channel model in [80]. The path loss is $48.9 + 40 \log(D)$, $D \geq 0.3$, where D is the distance between two machines in meters. The shadowing standard deviation and noise figure is 8 dB and 9 dB, respectively, and the maximum transmission power is 14 dBm. It is worth noting that the channel model between the machine and the BS is not needed in this work, because

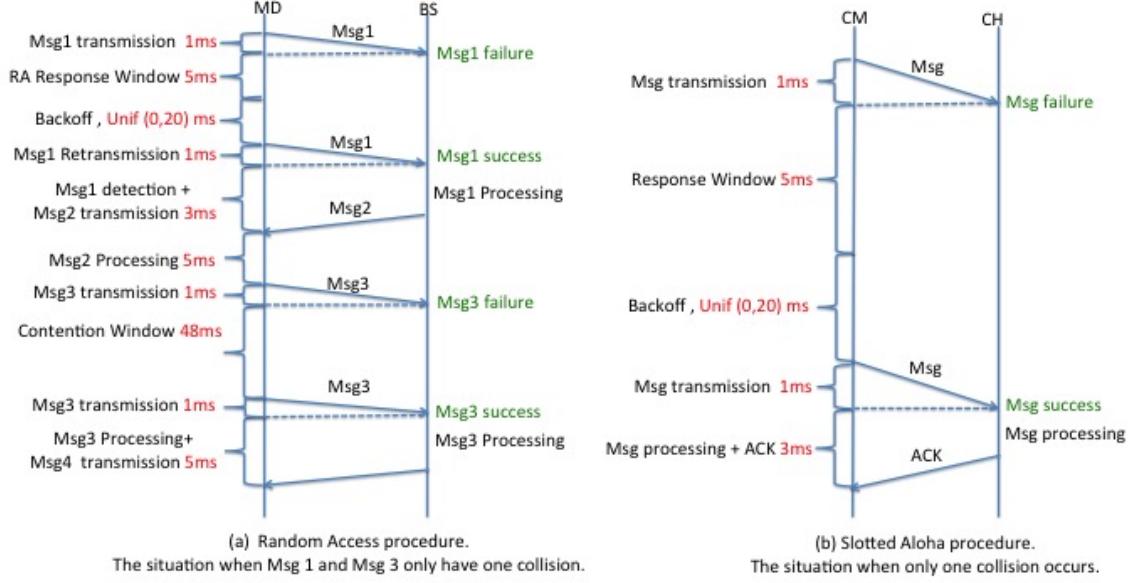


Figure 3.5: (a) Random access procedure details assuming **Msg1** and **Msg3** both have one collision (b) slotted ALOHA procedure details assuming only one collision occurs.

collisions in RACH only depend on the selection of the random access channels.

3.4 Analysis of Energy Consumption

3.4.1 Energy Consumption in Slotted ALOHA

From Fig. 3.5, it can be seen that the slotted ALOHA procedure is the repetition of transmission, response waiting window and backoff period. The backoff duration can be excluded when calculating the energy consumption due to its small contribution. The energy calculation here does not include the overhead for device location identification. Assuming the collision probability of packet transmission in slotted ALOHA is $P_{collision}$. From Table 3.1, the power consumption for transmission is P_{Tx2} , and the power for receiving and processing is P_{Rx2} . The transmission period, response waiting window, and receiving period are T_{Tx} , T_{RESP} , and T_{Rx} , respectively. The probability that a CM needs to transmit N times until success is

$$P_{collision}^{N-1} (1 - P_{collision}). \quad (3.1)$$

According to the time period of each step in Fig. 3.5 (b), the total elapsed time excluding backoff is

$$NT_{Tx} + (N - 1)T_{RESP} + T_{Rx}, \quad (3.2)$$

and the energy consumed is

$$NT_{Tx}P_{Tx2} + [(N - 1)T_{RESP} + T_{Rx}]P_{Rx2}. \quad (3.3)$$

Therefore, the expected number of transmissions is

$$\sum_{N=1}^{\infty} N(P_{collision})^{N-1}(1 - P_{collision}) = \frac{1}{1 - P_{collision}}. \quad (3.4)$$

The expected energy consumption can be formulated as:

$$\begin{aligned} & \sum_{N=1}^{\infty} (P_{collision})^{N-1}(1 - P_{collision})\{NT_{Tx}P_{Tx2} + [(N - 1)T_{RESP} + T_{Rx}]P_{Rx2}\} \\ &= \frac{1}{1 - P_{collision}}T_{Tx}P_{Tx2} + \left[\left(\frac{1}{1 - P_{collision}} - 1 \right)T_{RESP} + T_{Rx} \right]P_{Rx2}. \end{aligned} \quad (3.5)$$

Noted that associated with Fig. 3.5 (b) and Table. 3.1, the values of parameters T_{Tx} , T_{RESP} , T_{Rx} , P_{Tx2} , and P_{Rx2} in this equation are 1 ms, 5 ms, 3 ms, 25 mW, and 25 mW respectively.

3.4.2 Energy Consumption in RACH

Similarly, we calculate the energy consumption for random access procedure. Using the parameters in Table. 3.1, the collision probability of Msg1 transmission is denoted as $P_{Msg1coll}$, while the failure probability of Msg3 transmission is $P_{Msg3coll}$. The transmission power is P_{Tx1} and the receiving power & processing is P_{Rx1} . The Msg1 response window and Msg3 contention window duration are denoted as T_{RAR} and T_{CR} , respectively.

The probability that a machine needs to transmit Msg1 N times is

$$P_{Msg1coll}^{N-1}(1 - P_{Msg1coll}). \quad (3.6)$$

The probability of transmitting Msg3 M times is

$$P_{\text{Msg3coll}}^{M-1}(1 - P_{\text{Msg3coll}}). \quad (3.7)$$

The total consumed energy, excluding backoff, is

$$\begin{aligned} & (NT_{MSG1} + MT_{TransMSG3})P_{Tx1} + \\ & [(N - 1)T_{RAR} + T_{MSG2} + T_{MSG3} + (M - 1)T_{CR} + T_{MSG4}]P_{Rx1}. \end{aligned} \quad (3.8)$$

The expected energy consumption can be formulated as:

$$\begin{aligned} & \left(\frac{1}{1 - P_{\text{Msg1coll}}}T_{MSG1} + \frac{1}{1 - P_{\text{Msg3coll}}}T_{TransMSG3} \right)P_{Tx1} + \\ & \left[\left(\frac{1}{1 - P_{\text{Msg1coll}}} - 1 \right)T_{RAR} + T_{MSG2} + T_{MSG3} \right. \\ & \left. + \left(\frac{1}{1 - P_{\text{Msg3coll}}} - 1 \right)T_{CR} + T_{MSG4} \right]P_{Rx1}. \end{aligned} \quad (3.9)$$

3.5 C_V and the Inputs of the Location Generator

In this part, we examine the relationship between the location generator and the clustering metric C_V , so that we can control C_V by tuning the inputs of our location generator. Fig. 3.6 shows the behavior of C_V versus the inputs N , M , R , and $P_{isolated}$. As it can be seen in Fig. 3.6(a), varying the number of clusters M can only provide a C_V range of approximately over 5. In Fig. 3.6(b) the number of devices is not fixed, which cannot satisfy our needs either. In Fig. 3.6(c) and Fig. 3.6(d), varying the radius of clusters and probability of isolated node can offer a proper range of C_V starting from 1. From Fig. 3.6(c) it can be observed that when the radius of the cluster is small, the geometry is highly clustered, with a larger C_V . As the radius of the clusters increases, different clusters will overlap with each other, and will merge together and lose their distinct boundaries. The locations will become homogeneous and the value of C_V will drop to 1 (as in the PPP). Similar to $P_{isolated}$, it can also provide a proper range of C_V . In this work, we choose R as the tuning parameter for C_V .

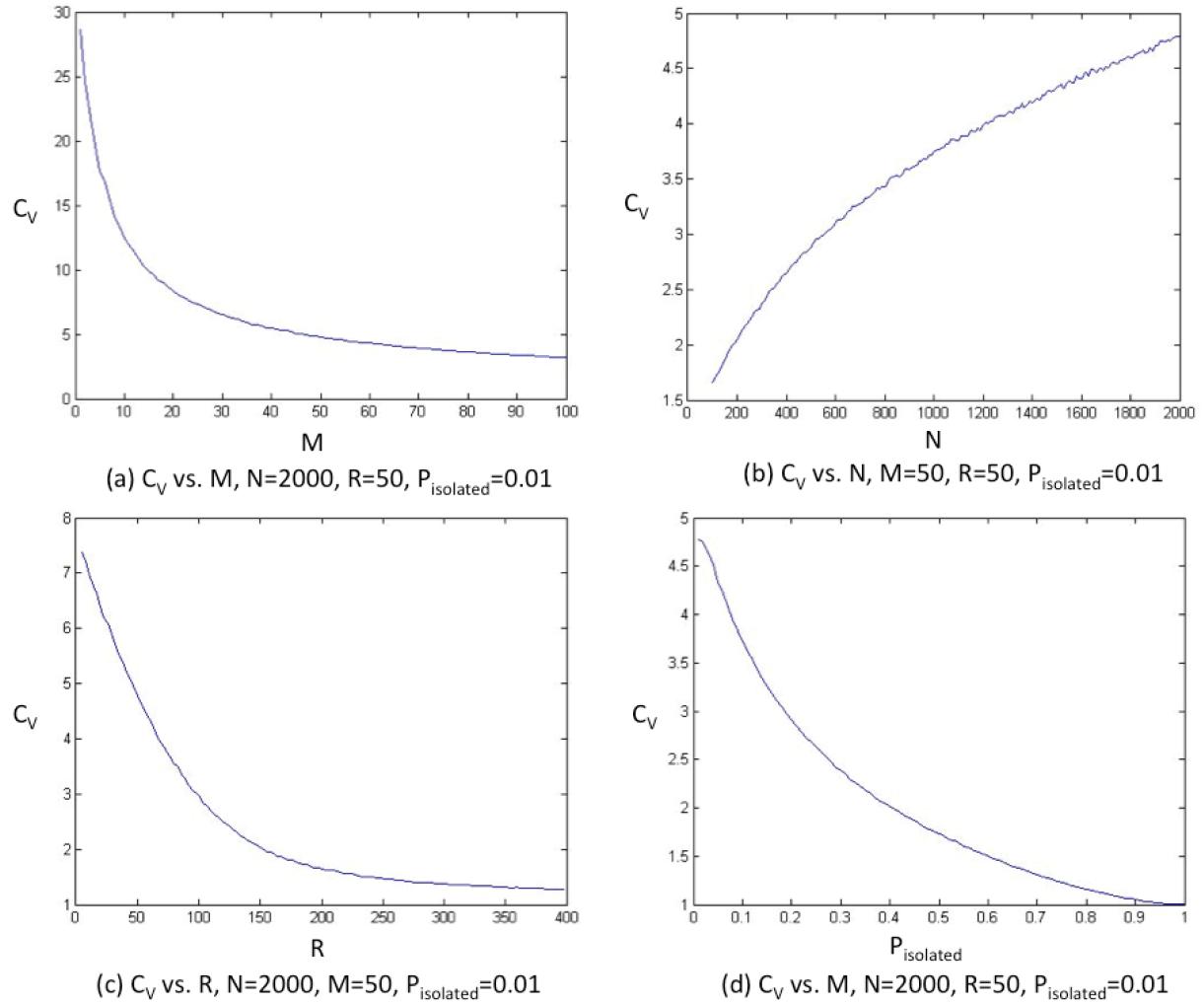


Figure 3.6: Relationship between C_V and the inputs of location generator.

Chapter 4

Simulation Results

Based on our reference simulator, we integrate the clustering location generator and the slotted ALOHA protocol. The simulator is run subframe by subframe until all machines complete their access procedure. Fig. 4.1 illustrates the flowchart of the complete simulation system. In our simulations, the overhead for device location identification is omitted, as such, the performance results are indicative for static or very slowly moving networks.

4.1 Performance as a Function of Clustering

We examine the impact of spatial clustering metric on a group of performance metrics: total transmission time, average access delay, collision/success probability, and energy consumption. We compare the reference one-hop method (Sec. 2.2.1) with the proposed two-hop method (Sec. 3.3). Fig. 4.2 shows the results of performance as a function of clustering.

A 95% confidence interval is provided in the results. It can be observed that, as C_V increases, the overall performance remains the same for the reference method, while it improves for the proposed method. This is because in the reference method it is the number of machines rather than the geometry of their locations that is relevant to the performance. Whereas, in the proposed method, when C_V is higher, the geometry of the machines is more clustered and different clusters are farther away from each other, which results in less interference in the slotted ALOHA communication, and therefore the overall performance is improved. It can also be observed that when C_V is greater than about 3, our proposed method outperforms the reference method.

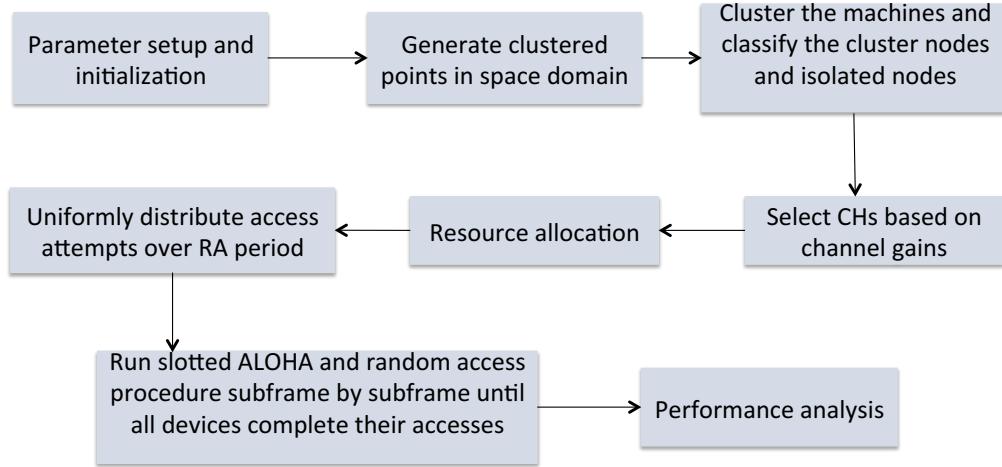


Figure 4.1: Flowchart of the complete simulation system.

In Fig. 4.2(d), the energy consumed by the machines in the proposed method is only 20% of that in the reference one. Here the energy consumption in the proposed scheme only considers the CM devices, because CH could be assigned as a special device with sufficient battery power, the energy of which is not a critical issue. This difference mainly results from the different access procedures in the two methods. In the reference method, a machine device experiences a four-message exchange, while in the proposed method, most of the devices go through the slotted ALOHA procedure. According to Fig. 3.5, it can be seen that the slotted ALOHA procedure only consists of the repetitions of transmission, a response waiting window, and a backoff period, which are less complicated than the random access procedures. Therefore, the energy consumption can be dramatically decreased.

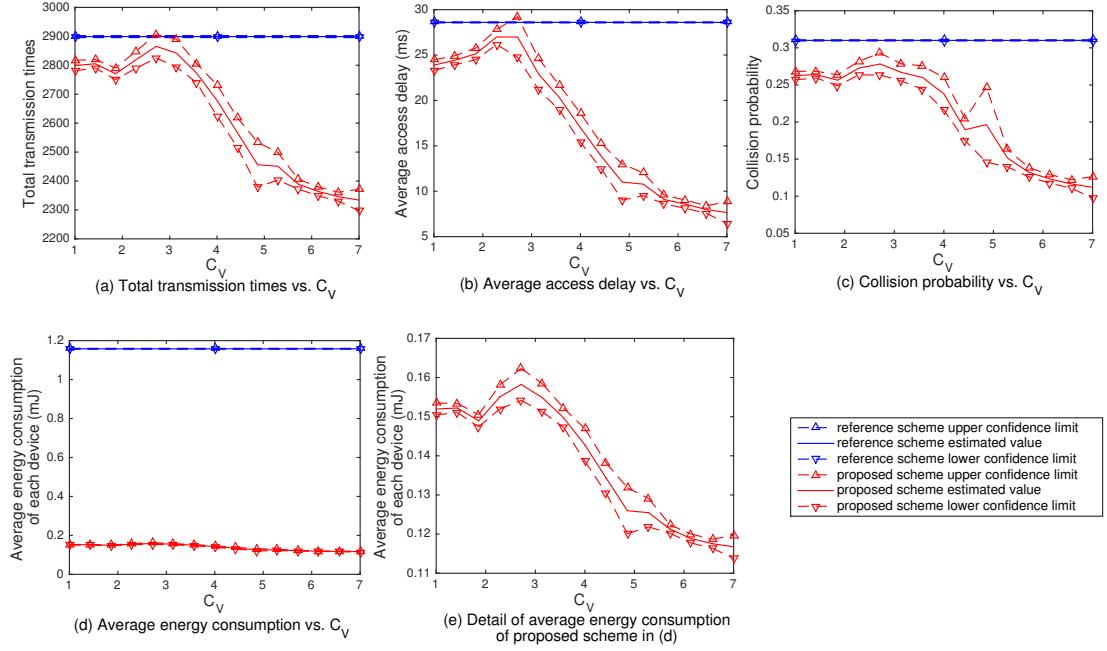


Figure 4.2: Performance vs. C_V with 95% confidence interval. C_V is varied by tuning the cluster radius R . The other inputs of location generator are: the number of clusters $M = 50$, the number of devices $N = 2000$, and the probability of isolated node $P_{isolated} = 0.01$.

4.2 Performance as a Function of the Number of Machines

We examine the impact of the number of devices (N) on the same set of performance metrics, while the other three inputs of location generator are fixed. The comparison is also done between the reference one-hop method and the proposed two-hop method. Fig. 4.3 shows the results.

It can be seen that, generally speaking, as the number of devices increases, more collisions will happen, which results in an overall worse performance. The proposed two-hop method still outperforms the reference one-hop method from all perspectives.

We also examine the impact of N on the performance metrics when C_V is maintained constant at 5. Fig. 4.4 shows the results. Curves are fitted to the scattered points. Here, we maintain a fixed C_V by tuning the cluster radius R with the help of a lookup table. As can be seen in Fig. 4.4 (e), C_V can be controlled to be approximately

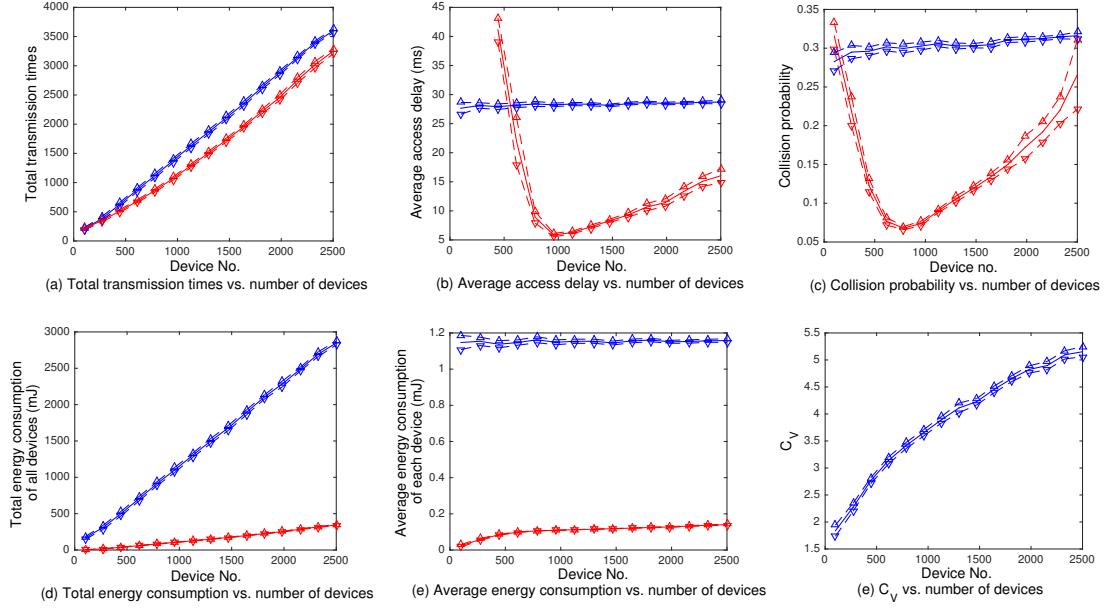


Figure 4.3: Performance vs. number of devices with 95% confidence interval. The other three inputs of the location generator are fixed, i.e. $M = 50$, $R = 50$, and $P_{isolated} = 0.01$. However, the C_V increases as the number of devices grows.

equal to 5. Fig. 4.4 shows the required R to adjust to the changed N to maintain the C_V . It also can be seen in (a–d) that the proposed scheme still provides a better performance than the reference scheme.

4.3 Performance as a Function of the Number of Clusters

Fig. 4.5 shows the impact of the number of clusters M on the performance metrics, while the other three inputs of our location generator are fixed. It can be seen that as M increases, the overall performance is almost constant. This is because the interference in slotted ALOHA communication arises from all the other machines inside and outside the cluster, who transmit their packets simultaneously. The interference is related to the distances and the number of contending machines, but is less affected by the number of clusters. It also can be seen in Fig. 4.5 (f) that, as M increases while keeping the other generator inputs fixed, the C_V goes down and the machine

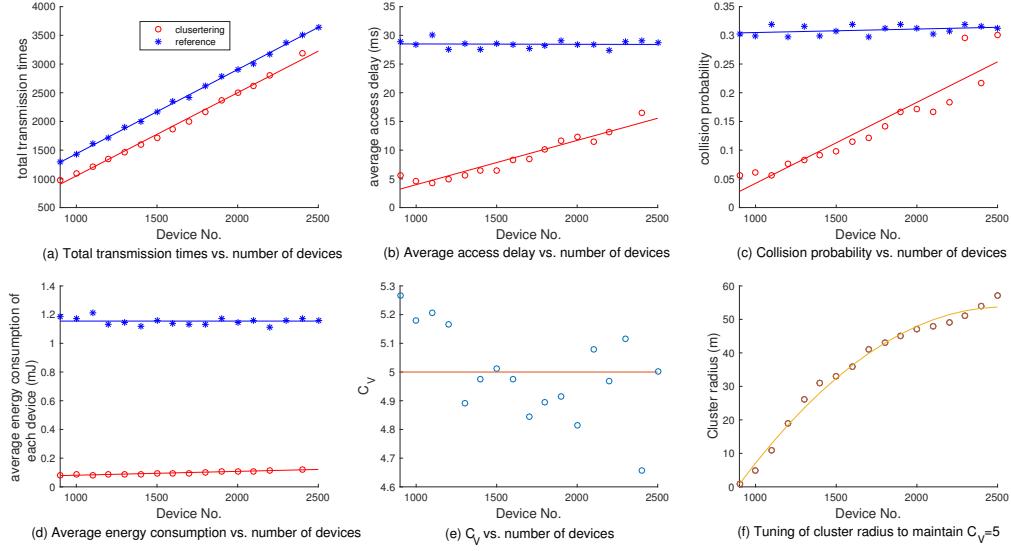


Figure 4.4: Performance vs. number of devices. The C_V is maintained around 5 by tuning R from the lookup table. The other inputs of the location generator are $M = 50$, and $P_{isolated} = 0.01$.

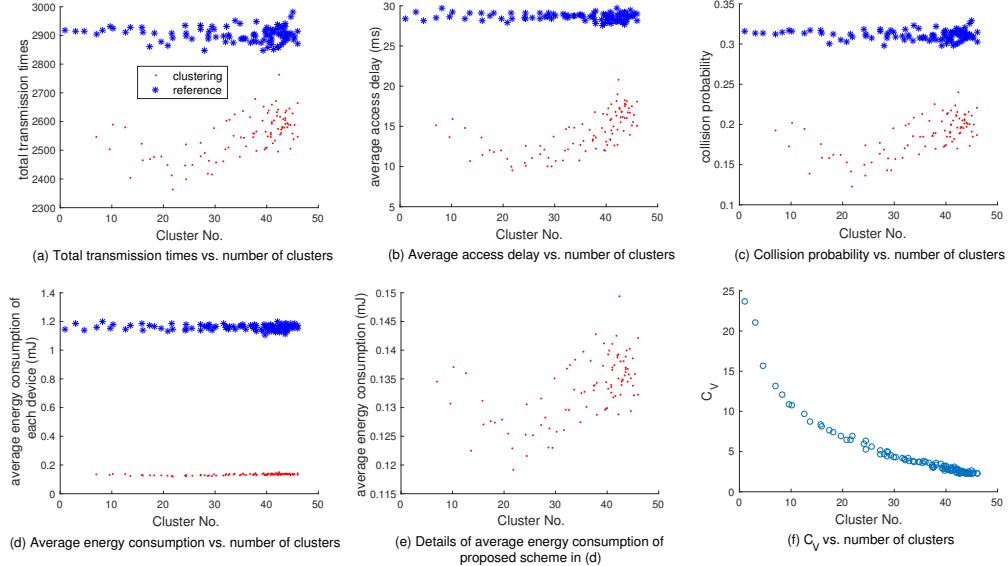


Figure 4.5: Performance vs. number of clusters. The other three inputs of the location generator are fixed, i.e. $N = 2000$, $R = 50$, and $P_{isolated} = 0.01$. However, the C_V decreases as the number of devices grows.

locations become less clustered.

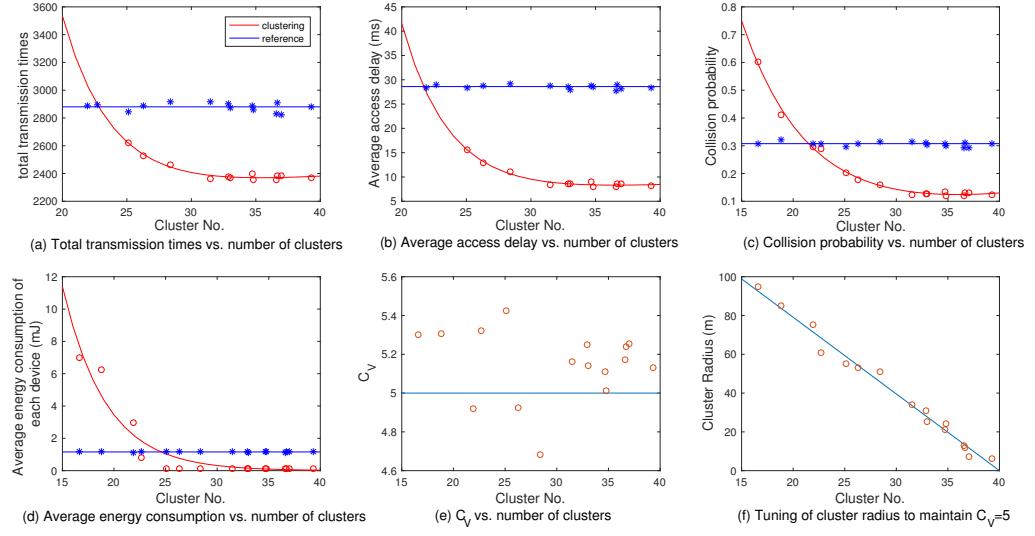


Figure 4.6: Performance vs. number of clusters M . The C_V is maintained around 5 by tuning R from the lookup table. The other inputs of location generator are $N = 2000$ and $P_{isolated} = 0.01$.

Fig. 4.6 shows the results when varying the number of clusters M while maintaining C_V fixed at 5. Fig. 4.6 (f) shows the required R to adjust to the changed M to maintain the C_V . The relationship between R and M is approximately linear. It can also be seen in Fig. 4.6 (a–d) that, as M is increased, the overall performance becomes better, and the proposed scheme outperforms the reference scheme when M is approximately larger than 22, which means there are less than 90 machines per cluster. If the clustering level is fixed, more clusters means a smaller cluster radius, which decreases the interference between machines, therefore giving better performance.

4.4 Energy Consumption as a Function of Collision Probability

According to (3.5) and (3.9), Fig. 4.7 compares the average energy consumption of machines in slotted ALOHA vs. RACH, varying the probability of collision. In the slotted ALOHA, the collision probability of packet transmission is varied. In the RACH, the collision probability of Msg1 is varied, while for Msg3 only four cases

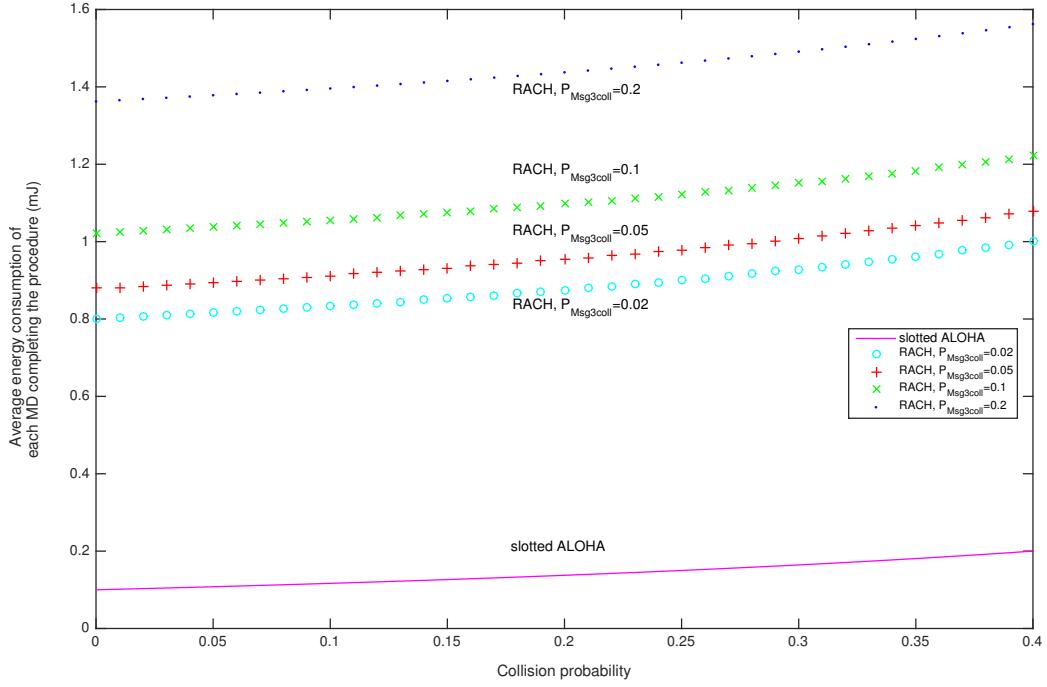


Figure 4.7: Comparison of average energy consumption of machine devices in slotted ALOHA vs. RACH, varying the collision probability. For the random access procedure, the collision probability of Msg1 is varied while for Msg3 only four cases are considered: the collision probability is equal to 0.02, 0.05, 0.1, and 0.2.

are considered when the collision probability is equal to 0.02, 0.05, 0.1, and 0.2. It can be observed that, for all situations, as the collision probability increases, more retransmissions are needed, which leads to more energy consumption. And for the collision probability of Msg3 in RACH, even if it is very small (e.g. 0.02), the slotted ALOHA method still gains a lot in terms of energy conservation.

Associated with Fig. 4.2 (c–e) and Fig. 4.7, the simulation results of average energy consumption against C_V can be validated in some way. For example, from Fig. 4.2 (e), it is known that when C_V is 3, the average energy consumption of each device in proposed scheme is around 0.15 mJ. From Fig. 4.2 (c), we can see when C_V is equal to 3, the collision probability is around 0.3. And from Fig. 4.7, we can observe that when the collision probability is around 0.3, the average energy consumption of each device is approximately 0.15 mJ, which is in line with the results in Fig. 4.2 (e).

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we propose a slotted ALOHA-based two-hop cluster random access method to improve the performance of MTC. In this method, the CH aggregates data from CMs through slotted ALOHA and then initiates the LTE random access procedure to the BS after the buffer reaches a certain threshold. Due to the offloading from RACH to the slotted ALOHA communication, the number of contending devices is greatly reduced in the RACH, which results in fewer collisions and a better performance. The utilization of the slotted ALOHA method also simplifies the access procedure, which can decrease the energy consumption of machine devices. Compared with previous literature, the reason we select the slotted ALOHA method for the first hop communication is that it is easy to implement and more efficient in terms of resources and energy utilization.

We also introduce a clustering geometry model for machine locations. In this model, most of the machines join different clusters according to their location proximity; there are also some isolated machines far away from any other cluster. Machines that belong to a cluster upload packets to the corresponding CH through slotted ALOHA, while the isolated machines directly access BS through the LTE random access procedure. In our model, the clustering of machine locations can be adjusted by tuning the input parameters.

We define a clustering metric C_V based on the cell areas of the Voronoi tessellation and the coefficient of variation statistics. This metric C_V can capture the clustering of machine locations well. The geometry of machine locations is clustered if $C_V > 1$, while it is repulsive if $C_V < 1$. We study the relationship between the clustering

metric C_V and the inputs of our location model. We find that by tuning the cluster radius R and the probability of isolated node $P_{isolated}$, a required range of C_V can be obtained. We can also build a C_V lookup table with ergodic discrete values of the inputs of our location model. Thus if some inputs are given, we can find the values of the other inputs in the lookup table to obtain the required C_V of a certain value.

We conduct some literature reviews on different clustering algorithms such as the hierarchical, K-means, and DBSCAN clustering algorithms. According to our needs, we finally decide to use the agglomerative hierarchical clustering algorithm in our work.

We conduct some mathematical analysis on the energy consumption of machine devices. We derive the mathematical expressions for the average energy consumption of each machine in both slotted ALOHA and RACH in terms of the collision probability of message transmission.

We examine the impact of location clustering on the system performance. Our results show that as the machine locations become more clustered, the overall performance metrics improve. It can also be seen that the energy consumption could be dramatically decreased using the proposed scheme, which is our main contribution. We explain that such a large gain in energy conservation is obtained mainly due to the simplified access procedure proposed, i.e., the two-hop slotted ALOHA-based random access procedure. The differences between the proposed procedure and the LTE random access procedure are illustrated in detail. We also examine the impact of the number of devices and the number of clusters on the system performance with and without a fixed C_V . Generally speaking, as the number of devices increases, the performance worsens, and as the number of clusters increases, the performance improves. In summary, the proposed scheme can outperform the reference scheme when the key parameter values satisfy some conditions, and a dynamic operation should be used to switch between the two schemes depending on configuration of the parameters.

We also validate our simulation results of average energy consumption with the analytical expressions in some way. It can be proved that the simulation and mathematical analyses match each other.

5.2 Future Work

5.2.1 Analytical Results

The research work of this thesis is mostly based on simulations in Matlab. The only analytical results are the expressions of energy consumption against collision probability in both slotted ALOHA and RACH in Sec. 3.4. As such, in future work we intend to conduct more mathematical analyses about the system performance - for example, analyses of the relationship between the clustering metric C_V and the inputs of our location generators, and of the expressions of different performances against C_V . After obtaining these analytical results, we will be able to compare them with the simulation results and thus validate the performances of our proposed system. It is worth noting that no mathematical analysis has been done about the clustering metric of C_V so far, since the metric was defined in [30]. Therefore, it could be meaningful to study a selection of analytical results about it in the future.

5.2.2 Implementation of Other Communication Schemes within the Cluster

In this thesis, we used slotted ALOHA, the easiest implementable random access method, for the communication within the cluster. Other methods can also be used for this tier of communication.

5.2.2.1 CSMA/CA

Carrier sense multiple access with collision avoidance (CSMA/CA) is also a random access protocol which can be used for the communication within the cluster. Compared with slotted ALOHA, it is more complicated, which may lead to more power consumption, but the collision rate will be reduced because of the ability of channel sensing. It would be interesting to compare this method with the slotted ALOHA method in future work.

5.2.2.2 Reference LTE Random Access Method

Another idea of realizing the communication within the cluster is to utilize the same method as the second hop. So the two hop communication will both use the LTE random access method to access the network. In this way, it is interesting to study how

to allocate RACH resources for both tiers. If they share the same resources, interference could happen. However, if they use separate resources, the optimal allocation strategy would be worthy of study. Compared with slotted ALOHA, the two-hop LTE random access method is more complicated and puts a high requirement on the CH, which should have some functions similar to those of a BS. However, it is in line with the current 3GPP LTE network and may facilitate implementation.

5.2.3 Cluster Discovery

In this thesis, the clustering discovery is based on the assumption that the BS is aware of the locations of all machines. It requires that all the machines are equipped with GPS and report their locations to the BS, which increases the cost and is energy consuming. A more practical method to do the cluster discovery is dependent on the neighborhood discovery through the interaction between machines without the assistance of the BS. A specific algorithm is proposed to achieve cluster discovery on the machine level in [81]. More research on this topic is worth study in the future work.

List of References

- [1] M. Hasan, E. Hossain, and D. Niyato, “Random access for machine-to-machine communication in LTE-Advanced networks: Issues and approaches,” *IEEE Commun. Mag.*, vol. 51, pp. 86–93, June 2013.
- [2] T. Taleb and A. Kunz, “Machine type communications in 3GPP networks: Potential, challenges, and solutions,” *IEEE Commun. Mag.*, vol. 50, pp. 178–184, Mar. 2012.
- [3] “Service requirements for machine-type communications (MTC),” TS 22.368, 3rd Generation Partnership Project (3GPP), Dec. 2015.
- [4] “System improvements for machine-type communications (MTC),” TR 23.888, 3rd Generation Partnership Project (3GPP), Sept. 2012.
- [5] “Evolved universal terrestrial radio access (E-UTRA); medium access control (MAC) protocol specification,” TS 36.321, 3rd Generation Partnership Project (3GPP), Dec. 2015.
- [6] “RAN improvements for machine-type communications,” TR 37.868, 3rd Generation Partnership Project (3GPP), Sept. 2012.
- [7] T. Deng, X. Wang, and W. Jiang, “A random access scheme based on device-to-device communications for machine-type communications,” in *IEEE International Conference on Communication Technology (ICCT)*, pp. 573–577, Nov. 2013.
- [8] G. Farhadi and A. Ito, “Group-based signaling and access control for cellular machine-to-machine communication,” in *IEEE Vehicular Technology Conference (VTC Fall)*, pp. 1–6, Sept. 2013.
- [9] S. H. Wang, H. J. Su, H. Y. Hsieh, S. P. Yeh, and M. Ho, “Random access design for clustered wireless machine to machine networks,” in *First International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 107–111, July 2013.
- [10] S. H. Wang, “Self-organizing scheme for machine-to-machine networks based on cross-entropy method,” in *IEEE International Conference on Internet of Things (iThings) and Green Computing and Communications (GreenCom) and Cyber, Physical and Social Computing(CPSCom)*, pp. 562–566, Sept. 2014.

- [11] H. K. Lee, D. M. Kim, Y. Hwang, S. M. Yu, and S. L. Kim, "Feasibility of cognitive machine-to-machine communication using cellular bands," *IEEE Wireless Commun.*, vol. 20, pp. 97–103, Apr. 2013.
- [12] Y. Zhang, X. Li, S. Zhang, and Y. Zhen, "Wireless sensor network in smart grid: Applications and issue," in *Information and Communication Technologies (WICT), Congress on 2012 World*, pp. 1204–1208, Oct. 2012.
- [13] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Commun. Mag.*, vol. 49, pp. 60–65, Apr. 2011.
- [14] L. Zheng, S. Chen, S. Xiang, and Y. Hu, "Research of architecture and application of internet of things for smart grid," in *2012 International Conference on Computer Science Service System (CSSS)*, pp. 938–941, Aug. 2012.
- [15] S. K. Ghosh, M. Suman, R. Datta, and P. K. Biswas, "Power efficient event detection scheme in wireless sensor networks for railway bridge health monitoring system," in *2014 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–6, Dec. 2014.
- [16] J. Cheng, M. Nakayama, P. an Ruan, J.-I. Cheon, M.-Y. Hsieh, H. Ogai, H. Inujima, and N. Yamauchi, "Bridge diagnosis system by using wireless sensor network and independent component analysis," in *SICE Annual Conference, 2008*, pp. 1683–1686, Aug. 2008.
- [17] H. N. Rad, S. S. Sani, and M. N. Rad, "A new inexpensive system for SHM of bridge decks using wireless sensor networks based on measurements of temperature and humidity," in *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, pp. 998–1009, Nov. 2015.
- [18] B. J. Chang, B. J. Huang, and Y. H. Liang, "Wireless sensor network-based adaptive vehicle navigation in multihop-relay WiMAX networks," in *22nd International Conference on Advanced Information Networking and Applications (aina 2008)*, pp. 56–63, Mar. 2008.
- [19] B. Zeng and L. Yao, "Study of vehicle monitoring application with wireless sensor networks," in *11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015)*, pp. 1–4, Sept. 2015.
- [20] N. Sharma, A. K. Jairath, B. Singh, and A. Gupta, "Detection of various vehicles using wireless seismic sensor network," in *2012 International Conference on Advances in Mobile Network, Communication and its Applications (MNCAPPS)*, pp. 149–155, Aug. 2012.
- [21] K. Saleem, A. Derhab, and J. Al-Muhtadi, "Low delay and secure M2M communication mechanism for ehealthcare," in *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 105–110, Oct. 2014.

- [22] A. C. M. Kumar, and P. Kumar, "City traffic congestion control in Indian scenario using wireless sensors network," in *2009 Fifth IEEE Conference on Wireless Communication and Sensor Networks (WCSN)*, pp. 1–6, Dec. 2009.
- [23] G. Yan, M. C. Weigle, and S. Olariu, "A novel parking service using wireless networks," in *2009. SOLI '09. IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, pp. 406–411, July 2009.
- [24] H. Park, J. Burke, and M. B. Srivastava, "Design and implementation of a wireless sensor network for intelligent light control," in *2007 6th International Symposium on Information Processing in Sensor Networks*, pp. 370–379, Apr. 2007.
- [25] X. Yang and H. H. Wang, "The design and implement of embedded M2M smart home system," in *2011 IEEE 3rd International Conference on Communication Software and Networks (ICCSN)*, pp. 134–137, May 2011.
- [26] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? a survey of alternatives," *IEEE Commun. Surveys Tuts.*, vol. 16, pp. 4–16, first quarter 2014.
- [27] "MTC simulation results with specific solutions," R2-104662, 3rd Generation Partnership Project (3GPP), Aug. 2010.
- [28] "Evaluation on push based RAN overload control schemes," R2-112071, 3rd Generation Partnership Project (3GPP), Apr. 2011.
- [29] C. D. Barr, *Applications of Voronoi Tessellations in Point Pattern Analysis*. ProQuest, Ann Arbor, MI, USA, 2008.
- [30] M. Mirahsan, Z. Wang, R. Schoenen, H. Yanikomeroglu, and M. St-Hilaire, "Unified and non-parameterized statistical modeling of temporal and spatial traffic heterogeneity in wireless cellular networks," in *IEEE International Conference on Communications Workshops (ICC)*, pp. 55–60, June 2014.
- [31] J. Møller and D. Stoyan, "Stochastic geometry and random tessellations," tech. rep., Department of Mathematical Sciences, Aalborg University, 2007.
- [32] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, "HetHetNets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, pp. 2252–2265, Oct. 2015.
- [33] F. Lagum, S. S. Szyszkowicz, and H. Yanikomeroglu, "CoV-based metrics for quantifying the regularity of hard-core point processes for modeling base station locations," *IEEE Wireless Commun. Lett.*, vol. 5, pp. 276–279, June 2016.
- [34] C. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- [35] V. Cherkassky and F. Mulier, *Learning From Data: Concepts, Theory, and Methods*. New York: Wiley, 1998.
- [36] R. Duda, P. Hart, and D. Stork, *Pattern Classification, 2nd ed.* New York: Wiley, 2001.

- [37] E. Backer and A. K. Jain, “A clustering performance measure based on fuzzy set decomposition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, pp. 66–75, Jan. 1981.
- [38] P. Hansen and B. Jaumard, “Cluster analysis and mathematical programming,” *Math. Program.*, vol. 79, pp. 191–215, 1997.
- [39] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs: NJ: Prentice Hall, 1988.
- [40] L. Zadeh, “Fuzzy sets,” *Inf. Control*, vol. 8, pp. 338–353, 1965.
- [41] R. Xu and D. Wunsch, *Hierarchical Clustering*, pp. 31–62. Wiley-IEEE Press, 2009.
- [42] A. K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 4–37, Jan. 2000.
- [43] A. Jain, M. Murty, and P. Flynn, “Data clustering: A review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
- [44] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Trans. Neural Netw.*, vol. 16, pp. 645–678, May 2005.
- [45] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.
- [46] B. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. London:Arnold, 2001.
- [47] J. Gower, “A general coefficient of similarity and some of its properties,” *Biometrics*, vol. 27, pp. 857–872, 1971.
- [48] G. N. Lance and W. T. Williams, “A general theory of classificatory sorting strategies,” *Computer Journal*, vol. 9, pp. 373–380, 1967.
- [49] F. Murtagh, “A survey of recent advances in hierarchical clustering algorithms,” *Computer Journal*, vol. 26, pp. 354–359, 1983.
- [50] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: NY: John Wiley & Sons, 1990.
- [51] P. Macnaughton-Smith, W. Williams, M. Dale, and L. Mockett, “Dissimilarity analysis: A new technique of hierarchical sub-division,” *Nature*, vol. 202, pp. 1034–1035, 1964.
- [52] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An efficient data clustering method for very large databases,” in *ACM Sigmod Record*, vol. 25, pp. 103–114, ACM, 1996.
- [53] S. Guha, R. Rastogi, and K. Shim, “CURE: An efficient clustering algorithm for large databases,” in *ACM Sigmod Record*, vol. 27, pp. 73–84, ACM, 1998.

- [54] S. Guha, R. Rastogi, and K. Shim, “ROCK: A robust clustering algorithm for categorical attributes,” in *15th International Conference on Data Engineering, 1999. Proceedings.*, pp. 512–521, Mar. 1999.
- [55] N. B. Karayiannis and P.-I. Pai, “Segmentation of magnetic resonance images using fuzzy algorithms for learning vector quantization,” *IEEE Trans. Med. Imag.*, vol. 18, pp. 172–180, Feb. 1999.
- [56] R. Xu and D. Wunsch, *Partitional Clustering*, pp. 63–110. Wiley-IEEE Press, 2009.
- [57] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, 2nd edition*. New York, NY ,USA: John Wiley & Sons, 2001.
- [58] E. Forgy, “Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications,” *Biometrics*, vol. 21, pp. 768–780, 1965.
- [59] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, 3rd*. San Diego, CA, USA: Academic Press, 2006.
- [60] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1992.
- [61] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–137, Mar. 1982.
- [62] Y. Linde, A. Buzo, and R. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Jan. 1980.
- [63] D. Pelleg and A. Moore, “Accelerating exact K-means algorithms with geometric reasoning,” in *Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 277–281, ACM, 1999.
- [64] K. Stoffel and A. Belkoniene, “Parallel K-means clustering for large data sets,” in *European Conference on Parallel Processing*, pp. 1451–1454, Springer, 1999.
- [65] I. Dhillon and D. Modha, “Concept decompositions for large sparse text data using clustering.,” *Machine Learning*, vol. 42, no. 1, pp. 143–175, 2001.
- [66] S. Z. Selim and M. A. Ismail, “K-means-type algorithms: A generalized convergence theorem and characterization of local optimality,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, pp. 81–87, Jan. 1984.
- [67] K. Krishna and M. N. Murty, “Genetic K-means algorithm,” *IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics)*, vol. 29, pp. 433–439, June 1999.
- [68] G. Patané and M. Russo, “The enhanced LBG algorithm,” *Neural Networks*, vol. 14, no. 9, pp. 1219–1237, 2001.
- [69] J. M. Pena, J. A. Lozano, and P. Larrañaga, “An empirical comparison of four initialization methods for the K-means algorithm,” *Pattern Recognition Letters*, vol. 20, no. 10, pp. 1027–1040, 1999.

- [70] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, Oakland, CA, USA., 1967.
- [71] A. Likas, N. Vlassis, and J. J. Verbeek, “The global K-means clustering algorithm,” *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.
- [72] G. Punj and D. W. Stewart, “Cluster analysis in marketing research: Review and suggestions for application,” *Journal of Marketing Research*, pp. 134–148, 1983.
- [73] G. H. Ball and D. J. Hall, “A clustering technique for summarizing multivariate data,” *Behavioral Science*, vol. 12, no. 2, pp. 153–155, 1967.
- [74] V. Estivill-Castro and J. Yang, “Fast and robust general purpose clustering algorithms,” in *Pacific Rim International Conference on Artificial Intelligence*, pp. 208–218, Springer, 2000.
- [75] C. Ordonez, “Clustering binary data streams with K-means,” in *the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 12–19, ACM, 2003.
- [76] Z. Huang, “Extensions to the K-means algorithm for clustering large data sets with categorical values,” *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283–304, 1998.
- [77] S. Gupta, K. S. Rao, and V. Bhatnagar, “K-means clustering algorithm for categorical attributes,” in *International Conference on Data Warehousing and Knowledge Discovery*, pp. 203–208, Springer, 1999.
- [78] “Feasibility study for further advancements for E-UTRA (LTE-Advanced),” TR 36.912, 3rd Generation Partnership Project (3GPP), Dec. 2015.
- [79] M. Gerasimenko, V. Petrov, O. Galinina, S. Andreev, and Y. Koucheryavy, “Energy and delay analysis of LTE-Advanced RACH performance under MTC overload,” in *2012 IEEE Globecom Workshops*, pp. 1632–1637, Dec. 2012.
- [80] K. Zheng, F. Hu, W. Wang, W. Xiang, and M. Dohler, “Radio resource allocation in LTE-Advanced cellular networks with M2M communications,” *IEEE Commun. Mag.*, vol. 50, pp. 184–192, July 2012.
- [81] W. Zhou, H. M. Chen, and W. H. Xie, “Cluster merging algorithm with link optimization for wireless sensor networks,” in *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Sept. 2006.