EFFICIENT AND QUALITY-AWARE DATA ACCESS IN MOBILE OPPORTUNISTIC NETWORKS

A Dissertation in
Computer Science Engineering
by
Xiaomei Zhang

© 2016 Xiaomei Zhang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2016
The dissertation of Xiaomei Zhang was reviewed and approved* by the following:

Guohong Cao  
Professor of Computer Science and Engineering  
Dissertation Advisor, Chair of Committee

George Kesidis  
Professor of Computer Science and Engineering  
Professor of Electrical Engineering

Sencun Zhu  
Associate Professor of Computer Science and Engineering

Zhenhui Li  
Assistant Professor of Information Science and Technology

Mahmut Kandemir  
Professor of Computer Science and Engineering  
Chair of Graduate Program for Computer Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

Recent advances in hand-held mobile devices have sparked the emergence of mobile opportunistic networks. These networks enable direct communication between mobile devices by utilizing the short-range communication interfaces embedded on mobile devices, such as Bluetooth or Wi-Fi Direct. Since mobile opportunistic networks rely on peer-to-peer connections between mobile devices, the data access does not require support from network infrastructures (e.g., cellular towers or access points). Therefore, mobile opportunistic networks have been widely used to facilitate data access in circumstances where network infrastructures are not available, such as during natural disasters or military attacks.

The specific goal of this dissertation is to develop comprehensive solutions for providing efficient and quality-aware data access in mobile opportunistic networks. In particular, two research problems are addressed in this dissertation, i.e., realizing efficient data transmission between nodes and improving the quality of data received by those nodes. Due to the dynamic network topologies and the unexpected failure of mobile devices, data transmission in mobile opportunistic networks is extremely difficult. To address this problem, this dissertation explores the social behavior patterns of mobile users and proposes social-aware solutions to achieve efficient data transmission. In addition, the data received by mobile devices may suffer from quality-awareness problem. For example, the data may be of bad quality because of the existence of unreliable mobile users, who provide low-quality data due to their limited capability to provide data, or some other malicious or subjective factors. Solutions for addressing this quality-awareness problem are also provided in this dissertation.

First, we propose a data transmission strategy by exploiting and utilizing a social structure called transient community, i.e., the community with associated temporal information. Most existing community detection methods fail to detect transient communities due to their aggregation of contact information into a sin-
gle weighted or unweighted network. Therefore, we propose a novel clustering method to detect transient communities by exploiting the pairwise contact processes between mobile users. Then, we propose a transient community-based data transmission strategy by utilizing the detected transient communities as the data transmission unit.

Second, we design data transmission strategies by considering the diverse connectivity characteristics in mobile opportunistic networks. Previous works fail to consider diverse connectivity characteristics, since they neglect the ubiquitous existence of Transient Connected Components (TCCs), where nodes inside a TCC can reach each other by multi-hop wireless communications. Exploiting the special characteristics of TCCs can increase contact opportunities and significantly improve data transmission performance.

Third, to improve the quality of data received by the destination, we provide solutions to adaptively collect data from mobile users through mobile opportunistic networks, especially when the existing data are not enough to ensure data quality or credibility. Considering the requirement on data credibility and the constraint of network resources, we propose two optimization problems to quantify the tradeoff between the enhanced data credibility and the increased network overhead. We then design resource-aware approaches for the proposed problems.

Finally, we design an approach to improve data quality by considering the expertise of the mobile users, which is based on the fact that a user may only have expertise on some problems in some domains, but not others. The proposed expertise-aware approach relies on an optimization problem to maximize the probability that data are requested from users with the right expertise, while ensuring the work load does not exceed the processing capability of each user. A maximum likelihood estimation (MLE)-based method is further proposed to estimate the truth of data, so that data quality can be improved.
# Table of Contents

List of Figures x

List of Tables xiii

Acknowledgments xiv

Chapter 1  
Introduction 1  
1.1 Motivation and Challenges 2  
1.1.1 Inefficient Data Transmission 2  
1.1.2 Quality-Awareness Problem 3  
1.2 The State of the Art 3  
1.2.1 Data Transmission in Mobile Opportunistic Networks 3  
1.2.2 Data Quality Improvement 4  
1.3 Focus of This Dissertation 5  
1.3.1 Transient Community-based Data Transmission 5  
1.3.2 Data Transmission in Networks with Diverse Connectivity Characteristics 6  
1.3.3 Resource-Aware Approaches for Truth Analysis 7  
1.3.4 Expertise-Aware Truth Analysis and Task Allocation 7  
1.4 Organization 8

Chapter 2  
Transient Community-based Data Transmission 9  
2.1 Introduction 9  
2.2 Transient Community Detection 12  
2.2.1 Preliminary 12
3.3.2 Enhanced TCC-aware Data Transmission Strategy .......................... 54
  3.3.2.1 Motivation ........................................ 54
  3.3.2.2 Set Centrality ...................................... 55
  3.3.2.3 Selecting the Optimal Set of Data Carriers ......................... 56
  3.3.2.4 Transmission Strategy .................................. 56
3.4 Performance Evaluations .................................................. 57
  3.4.1 Strategies in Comparison ....................................... 58
  3.4.2 Simulation Setup ............................................. 58
  3.4.3 Simulation Results .............................................. 59
    3.4.3.1 Comparisons with Compare-and-Forward ...................... 60
    3.4.3.2 Comparisons with R3 ..................................... 61
    3.4.3.3 Comparisons between the TCC-aware data transmission strategies ........................................ 61
3.5 Related Work ............................................................ 61
3.6 Conclusions ............................................................... 63

Chapter 4
Resource-Aware Approaches for Truth Analysis in Crowdsourcing 64
4.1 Introduction ............................................................... 64
4.2 Preliminaries .............................................................. 66
  4.2.1 Problem Formulation ........................................... 66
  4.2.2 Models ............................................................ 69
    4.2.2.1 Network Model ........................................... 69
    4.2.2.2 Reliability of Users ..................................... 69
4.3 Max-Credibility with Network Resource Constraint ...................... 70
  4.3.1 User Selection ................................................... 70
    4.3.1.1 Network reachability .................................... 70
    4.3.1.2 Reliability ............................................... 71
  4.3.2 Finding Truth using MLE ......................................... 73
    4.3.2.1 Background for MLE ..................................... 73
    4.3.2.2 Mathematical Formulation and Derivation .................. 73
4.4 Achieving Specified Credibility with Min-Overhead ..................... 75
  4.4.1 An Overview of the Iterative Approach ................................ 75
  4.4.2 Selecting Users and Estimating Truth ................................ 76
  4.4.3 Evaluating Data Credibility ..................................... 77
4.5 Performance Evaluations ................................................ 78
  4.5.1 Trace-Driven Simulations ........................................ 78
    4.5.1.1 Results on the Max-Credibility Approach .................. 78
    4.5.1.2 Results on Min-Overhead ................................. 80
# Chapter 5

**Expertise-Aware Truth Analysis and Task Allocation in Crowdsourcing**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>88</td>
</tr>
<tr>
<td>5.2 Preliminary</td>
<td>90</td>
</tr>
<tr>
<td>5.2.1 Background</td>
<td>90</td>
</tr>
<tr>
<td>5.2.2 An Overview of ETA²</td>
<td>91</td>
</tr>
<tr>
<td>5.2.3 Distribution of Random Observation</td>
<td>92</td>
</tr>
<tr>
<td>5.2.4 Expertise Model</td>
<td>93</td>
</tr>
<tr>
<td>5.3 Task Expertise Identification</td>
<td>93</td>
</tr>
<tr>
<td>5.3.1 Basic Idea</td>
<td>93</td>
</tr>
<tr>
<td>5.3.2 Semantic Information Extraction</td>
<td>94</td>
</tr>
<tr>
<td>5.3.3 Dynamic Hierarchical Clustering</td>
<td>95</td>
</tr>
<tr>
<td>5.3.3.1 Hierarchical Clustering</td>
<td>95</td>
</tr>
<tr>
<td>5.3.3.2 Dynamic Hierarchical Clustering</td>
<td>96</td>
</tr>
<tr>
<td>5.4 Expertise-Aware Truth Analysis</td>
<td>97</td>
</tr>
<tr>
<td>5.4.1 Estimation of Truth and User Expertise</td>
<td>97</td>
</tr>
<tr>
<td>5.4.2 Dynamic Update of User Expertise</td>
<td>99</td>
</tr>
<tr>
<td>5.4.2.1 Adding new tasks to existing expertise domain</td>
<td>99</td>
</tr>
<tr>
<td>5.4.2.2 Creating a new expertise domain</td>
<td>100</td>
</tr>
<tr>
<td>5.4.2.3 Merging existing expertise domains</td>
<td>101</td>
</tr>
<tr>
<td>5.5 Expertise-Aware Task Allocation</td>
<td>101</td>
</tr>
<tr>
<td>5.5.1 The Optimization Problem</td>
<td>101</td>
</tr>
<tr>
<td>5.5.1.1 Objective function</td>
<td>101</td>
</tr>
<tr>
<td>5.5.1.2 Constraints</td>
<td>102</td>
</tr>
<tr>
<td>5.5.2 Heuristic Based Algorithm</td>
<td>103</td>
</tr>
<tr>
<td>5.6 Performance Evaluations</td>
<td>105</td>
</tr>
<tr>
<td>5.6.1 Datasets</td>
<td>105</td>
</tr>
<tr>
<td>5.6.1.1 Survey-based Dataset</td>
<td>105</td>
</tr>
<tr>
<td>5.6.1.2 SFV Dataset</td>
<td>106</td>
</tr>
<tr>
<td>5.6.2 Experimental Setting</td>
<td>106</td>
</tr>
<tr>
<td>5.6.3 Approaches in Comparison</td>
<td>107</td>
</tr>
<tr>
<td>5.6.4 Evaluation Results</td>
<td>108</td>
</tr>
<tr>
<td>5.6.4.1 The Effects of Parameters</td>
<td>108</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.6.4.2 Overall Performance of ETA²</td>
<td>110</td>
</tr>
<tr>
<td>5.6.4.3 Evaluation of Different Modules in ETA²</td>
<td>112</td>
</tr>
<tr>
<td>5.7 Related Work</td>
<td>113</td>
</tr>
<tr>
<td>5.8 Conclusion</td>
<td>114</td>
</tr>
</tbody>
</table>

Chapter 6

Conclusions and Future Work

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Conclusions</td>
<td>116</td>
</tr>
<tr>
<td>6.2 Future Directions</td>
<td>118</td>
</tr>
<tr>
<td>6.2.1 Mobile Opportunistic Networks</td>
<td>118</td>
</tr>
<tr>
<td>6.2.2 Crowdsourcing</td>
<td>119</td>
</tr>
</tbody>
</table>

Appendix

The Proof of Theorem 1 in Section 3.2.4

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibliography</td>
<td>124</td>
</tr>
</tbody>
</table>
List of Figures

2.1 False mixture: (a) $TC_1$ appears during daytime and $TC_2$ appears at night. (b) The two TCs are falsely mixed into one community using a traditional method. 10
2.2 False separation: (a) The network has one TC. (b) The TC is falsely separated into two communities using a traditional method. 11
2.3 Three contact bursts. An arrow represents a contact between two nodes. 14
2.4 A connected graph built on contact bursts. Six contact bursts have similar contact durations, and they may be from one TC, circled by red line. The other four have different contact durations, which are not in this TC. 15
2.5 For two clusters to merge, they must share at least one common node. (a) No common node, nodes $U,V$ and nodes $X,Y$ may come from two different TCs which appear at similar time. (b) Two contact bursts with common node $U$, and these three nodes form a TC. 17
2.6 The number of communities (left, blue) and the average size of the communities (right, green) with varying $\gamma$ (UCSD trace) 18
2.7 Percentage (PCT) of intra-community contacts with varying $\gamma$ 18
2.8 Comparisons among CMP, AFOCS and our algorithm CCM 24
2.9 The performance of distributed CCM as the variation of running time (number of running days) 26
2.10 Evaluation of distributed CCM by comparing with CCM: (a) comparison with CCM on the proportion of intra-community contacts; (b) CDF of the similarities of distributed TCs to centralized TCs 27
2.11 CDF of TCs’ appearance times. 29
2.12 The PDF of a TC’s start time on a daily basis and the PDF of the TC’s duration 30
2.13 Data delivery ratio of various methods based on three traces 36
2.14 Data transmission overhead measured by the number of data copies 36
5.3 The three types of changes for expertise domains (clusters) after new tasks (grey) are added. Solid circles represent existing expertise domains. Dotted circles represent new or updated expertise domains.

5.4 The effects of $\alpha$ and $\gamma$ on the performance (survey-based dataset).

5.5 The effects of $\alpha$ and $\gamma$ on the performance (SFV dataset).

5.6 Comparison of estimation error (a)(b) in different days and (c)(d) with different processing capability.

5.7 The effects of User expertise on observation error in (up) survey-based dataset and (down) SFV dataset.

5.8 The comparison between dynamic hierarchical clustering (DHC) and k-means.

5.9 The comparison between expertise-aware truth analysis and reliability-based truth analysis.

5.10 The comparison between expertise-aware task allocation and reliability-based task allocation.
List of Tables

2.1 Trace summary .................................................. 13
2.2 How contact bursts can represent contact processes ($\lambda = 1$) 14
2.3 The parameters of normal and exponential distribution . 31
2.4 Effect of $T_w$ on Data Delivery Ratio ....................... 39

3.1 Trace Summary .................................................. 45
3.2 TCC-contacts & Direct Contacts ............................... 50

4.1 The parameters of log-normal distributions ................... 83
Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor Dr. Guohong Cao, for his consistent support and supervision. I am very lucky to be able to work with him during my PhD study and have truly learned a lot from him. During my Ph.D study, he is very patient and has spent considerable time and effort in training me as a more matured researcher. It is because of his unreserved assistance that I could go through many difficult times either as a Ph.D student or in my life. Also, his attitude, spirit, and experience on research have influenced me profoundly, and will guide me as a researcher in my future career.

High tribute shall be paid to Dr. George Kesidis, Dr. Sencun Zhu, and Dr. Zhenhui Li, who are among my thesis committee. It is my great privilege to have these wonderful professors in my committee. Without their generous help and encouragement, the dissertation could not have reached the present form.

The work in this dissertation was supported in part by the National Science Foundation (NSF) under grant CNS-1320278 and CNS-1421578. Therefore, I would also like to express my gratitude to NSF for providing the support.

I also would like to thank all my fellow labmates in MCN Lab who share with me the various resources and give me valuable comments during my Ph.D study. It is a pure joy to work with them. I also thank them for being my friend and share much great time with me. Special gratitude also goes to the brothers and sisters in my local churches. They helped me such a lot during my life in Penn State. Their prayer upholds me and their love impels me.

My gratitude also extends to my beloved parents, who have been assisting and caring me all my life and also been such a helper after I have my son. I have received too much from them. I owe a special debt of gratitude to them.

I also wish to express my deepest gratitude and love to my husband Shuangxi Ren and my son Mark Ren. I would like to thank them for coming to my life, sharing so many wonderful memories in my life, and continuing to help me become stronger and more mature as a wife and a mom.
Dedication

To my beloved parents, husband and my precious son.
Chapter 1

Introduction

Mobile devices\(^1\) like smartphones, laptops, and tablets have become essential to people’s daily life. Mobile devices are able to communicate with each other and connection to the Internet through network infrastructures, including cellular network base stations and the WiFi access points. Issues arise when the connections to these network infrastructures become constrained or unavailable, which is very common in many extreme environments, like during natural disasters, military attacks, or simply when network infrastructures are broken or congested. In order to transmit data between mobile devices in the absence of network infrastructures, the mobile devices need to form an ad-hoc network, namely \textit{mobile opportunistic network}\(^2\), where the nodes communicate with each other through opportunistic peer-to-peer connections.

Facilitating data access in mobile opportunistic networks is crucial, since the more extreme and challenging the environments are, the more requirements are placed on the efficient transmission of high-quality information. Successful data access in mobile opportunistic networks not only requires efficient transmission of data, but also depends on the quality of the received data. For example, in a disaster recovery scenario, outdated or inaccurate data about the building/road damage

\(^1\)For the remainder of this dissertation, the terms “device”, “node”, and “user” are used interchangeably.

\(^2\)Mobile opportunistic network [1] is also alternatively referred to as mobile social network [2], delay tolerant network [3][4], opportunistic mobile network [5] or pocket-switched network [6]. Since they all represent the network in which mobile nodes have opportunistic peer-to-peer connections, this dissertation uniformly uses the term mobile opportunistic network to represent such networks.
condition and human injury/loss may totally ruin the rescue efforts. Therefore, the specific goal of this dissertation is to develop comprehensive solutions for providing efficient and quality-aware data access in mobile opportunistic networks.

1.1 Motivation and Challenges

Facilitating efficient and quality-aware data access in mobile opportunistic networks is extremely difficult due to inefficient data transmission and the quality-awareness problem.

1.1.1 Inefficient Data Transmission

Data transmission in mobile opportunistic networks relies on peer-to-peer contacts (i.e., when mobile users move into the communication range of their devices). Therefore, mobile opportunistic networks can also be generalized into the well known Mobile Ad-hoc Networks (MANETs). However, traditional data transmission strategies in MANETs [7][8] cannot be effectively applied here, since mobile users are only intermittently connected in mobile opportunistic networks and MANETs require mobile users to be fully connected. This intermittent connectivity among mobile users is caused by the following two reasons:

- **Dynamic Topology**: Due to the low node density and unpredictable node mobility, the network topology is highly dynamic in mobile opportunistic networks. Because of this dynamic topology, connectivity among mobile users is very intermittent and dynamic. It is difficult to maintain consistent end-to-end transmission paths between nodes.

- **Unexpected Failures**: Intermittent connectivity is also caused by the unexpected failures of mobile devices, such as power depletion, unexpected functional failure, or malicious attacks. When nodes are removed from the network due to these unexpected failures, the network may become even more disconnected.
1.1.2 Quality-Awareness Problem

In addition to data transmission, successful data access in mobile opportunistic networks also depends on the quality of the received data. However, in the real world, data quality is hard to ensure for many reasons. First, the mobile users who provide data (i.e., the data sources), may have limited capabilities in observing or sensing the data. For example, a person may report incorrect information about building damage if the person is not in the vicinity of the building. Second, the data source may be malicious, and may intentionally provide wrong data. Third, some uncontrollable subjective factors (e.g., selfishness) exist, which may prevent users from providing correct data. For example, a user may intentionally generate data instead of collecting the data himself to save time, effort, or resources, such as battery life.

1.2 The State of the Art

In the literature, various research efforts have been made to address the two research problems on efficient data transmission and data quality improvement. The existing research is summarized in this section.

1.2.1 Data Transmission in Mobile Opportunistic Networks

Because there are no consistent end-to-end paths between nodes, a “carry-and-forward” method [9][10] is commonly used to forward data between nodes. With “carry-and-forward”, mobile nodes physically “carry” the data and forward the data when contacting a node with “higher” forwarding capability to the destination. The key problem is how to select the appropriate relays that have “higher” forwarding capabilities. Traditional solutions to this problem include predicting users’ contacts by capturing and formulating their mobility patterns in different ways, such as using a Kalman filter [11], semi-Markov chains [12], or Hidden Markov Models [13]. However, these solutions are ineffective for relay selection and data transmission in mobile opportunistic networks due to the inaccurate formulation of users’ mobility patterns. This inaccuracy is essentially because these solutions are incapable of capturing the social behavior patterns of mobile users.
To address this problem, researchers have further explored the social behavior patterns of mobile users and proposed approaches to exploit social network concepts, such as community [14][15][16][17] and centrality [10][18][19], in order to decide on the appropriate relays.

Community has been widely used for data transmission in mobile opportunistic networks [14][15][16][17]. The basic idea is that, in order to transmit data to a specific destination, data are first transmitted to the nodes that belong to the same community with the destination. Centrality is another concept that receives considerable attention because of its assistance in data transmission. The centrality of a node represents its capability of contacting other nodes. A node with high centrality is usually the common acquaintance of other nodes, and thus act as the communication hub. In the centrality-based data transmission strategy [10][18][19], transmission decisions are made upon two nodes’ contact, and the data are always transmitted to the node with higher centrality.

1.2.2 Data Quality Improvement

Since data from a single provider may be highly unreliable and deviate from the truth, people usually request data from multiple sources and regard the majority or average of the answers as truth. In the broad field of networking, collecting data from multiple sources simultaneously is called crowdsourcing [20].

To more accurately estimate truth from multiple answers rather than using the majority or average of answers, researchers have taken the reliability of users into consideration, i.e., high-reliability users tend to provide high-quality data. By inferring and utilizing user reliability, the truth can be better identified. Based on this approach, an important problem revolves around how to assess the reliability of mobile users and use it to better identify the truth from all reported data. To address this problem, various truth analysis techniques have been proposed by researchers. For example, [21][22][23] designed techniques based on a basic heuristic, i.e., iteratively estimating the truth of the information by analyzing the reliability of the mobile users, and estimating the reliability of mobile users based on the correctness of their provided information. Other researchers also investigated statistical models for truth finding, such as bayesian inference [24][25][26]
and expectation-maximization (EM) [27].

1.3 Focus of This Dissertation

Despite the fact that many solutions have been proposed to address these two research problems, many important issues have been overlooked in the exiting solutions. The specific goal of this dissertation is to develop more effective solutions for realizing efficient data transmission and improving data quality in mobile opportunistic networks. The following four subsections present the four problems addressed in this dissertation, respectively.

1.3.1 Transient Community-based Data Transmission

Even though community has been widely used for data transmission in mobile opportunistic networks, it is still a challenge to detect communities in a large network, especially considering the temporal information associated with community. For example, a class community consisting of students attending a class may only appear during daytime, and a dormitory community consisting of students at a dormitory may only appear at night. Since these communities normally appear during a time period and disappear thereafter, they are referred to as transient communities (TCs). Although many community detection methods have been proposed in the literature, there are hardly any methods for TC detection.

In this work, we propose a Contact-burst-based Clustering Method (CCM) to detect TCs by exploiting the pairwise contact processes. This method is based on the detailed pairwise contact information between nodes instead of networks with unweighted on-off edges or weighted edges. In CCM, we formulate each pairwise contact process as regular appearance of contact bursts, during which most contacts between the pair of nodes appear. Based on such formulation, we detect TCs by clustering the pairs of nodes with similar contact bursts together. Since it is hard to collect global contact information at individual nodes, we also propose a distributed method. In addition to TC detection, we also propose a new data transmission strategy for mobile opportunistic networks, in which TCs serve as the data transmission unit. Evaluation results show that our strategy
can achieve a much higher data delivery ratio than traditional community-based strategies with comparable network overhead.

1.3.2 Data Transmission in Networks with Diverse Connectivity Characteristics

Mobile opportunistic networks have been assumed to be highly sparse. However, after analyzing four real-world traces on mobile opportunistic networks, we find that the connectivity inside a network is actually diverse. This is because there is ubiquitous existence of *Transient Connected Components (TCCs)*. Inside each TCC, nodes have transient contacts with each other and form a connected component. For example, students in a classroom have transient connections with each other, and vehicles on highways form platoons and have transient connections inside a platoon [28]. Outside TCCs, nodes are still opportunistically connected. Since most existing strategies for mobile opportunistic networks are based on the assumption that the networks are sparsely connected, the diverse connectivity characteristics make the existing strategies far from optimal. In this work, we address the problem of data transmission in networks with diverse connectivity characteristics by proposing TCC-aware data transmission strategies.

More specifically, we first identify the existence of TCCs and analyze their properties based on four traces. From both network analyses and theoretical analyses, we find that the contact opportunities can be significantly increased in all traces, if the multi-hop wireless communications inside TCCs are treated as indirect contacts. Then, we propose two TCC-aware data transmission strategies that exploit TCCs to improve the performance of data transmission in mobile opportunistic networks. In the first solution, nodes inside a TCC exchange their forwarding metrics (e.g., centrality) through multi-hop wireless communications, and the node with the highest forwarding metric is selected to receive a replicated data copy. Although this strategy can increase the data delivery ratio, it also increases the data copies in the network. To address this problem, we further propose an enhanced strategy by selecting an optimal set of nodes in the TCC to avoid overlap in their contacts and maximize the data transmission opportunity with a small number of nodes. Trace-driven simulations show that our TCC-aware
data transmission strategies outperform existing data transmission strategies with much higher delivery ratio and less network overhead.

1.3.3 Resource-Aware Approaches for Truth Analysis

Based on crowdsourcing, an effective truth analysis technique is needed to infer truth from multiple data, so that data quality can be improved. Even though there exist a lot of work on truth analysis, these existing techniques only depend on the current available or observed data. However, if the available data are limited or have large conflicts, it is difficult to identify the truth or ensure the data credibility (quality). In this work, we address this problem by utilizing the communication networks, i.e., mobile opportunistic networks, to adaptively collect data from mobile users, especially when the existing data are not enough to ensure data credibility. By doing so, data credibility can be improved by utilizing communication networks to collect extra data. However, it also increases communication overhead for data collection, which can be a significant issue in mobile opportunistic networks. This is because the network resources are limited in mobile opportunistic networks due to the dynamics of network topology and the lack of end-to-end routing paths. Considering the requirement on data credibility and the constraint of network resources, we quantify the tradeoff between the enhanced data credibility and the increased network overhead, and propose resource-aware approaches for truth analysis. Specifically, we formalize two problems in resource-constrained mobile opportunistic networks: max-credibility, which aims to maximize data credibility with some network overhead, and min-overhead, which aims to achieve a specified data credibility while minimizing the network overhead. Simulation and experimental results demonstrate the effectiveness of the proposed solutions in terms of data credibility and network overhead.

1.3.4 Expertise-Aware Truth Analysis and Task Allocation

In this work, we further propose solutions to improve data quality by considering the expertise of mobile users, which is also based on crowdsourcing. Existing techniques identify truth from noisy data by inferring and utilizing the reliability of users. When a user requests data, a crowdsourcing task is created and allocated to
the users with high reliability, so that high-quality data can be collected. However, these techniques neglect the fact that a user may only have expertise on some problems (in some domains), but not others. Neglecting this diversity in expertise may cause two problems: low estimation accuracy in truth analysis and ineffective task allocation. First, by applying a reliability-based truth analysis technique, a user inferred to have high reliability may actually have very low expertise in some domains. Then, for tasks in these domains, the estimated truth may not be correct, since the user with “high-reliability” may provide wrong data due to lack of expertise in these domains. Second, task allocation will also be affected, since identified “high-reliability” users may not actually provide accurate data in all expertise domains. A more severe problem is related to unfair task allocation, i.e., all tasks are assigned to a few users with “high-reliability”, while the remaining majority of users are not assigned any task.

Considering these problems, we propose an Expertise-aware Truth Analysis and Task Allocation (ETA²) approach to address the aforementioned challenges. ETA² can effectively infer user expertise and then allocate tasks and estimate truth based on the inferred expertise. ETA² relies on a novel semantic analysis method to identify the expertise domains of tasks and user expertise, an expertise-aware truth analysis solution to estimate truth and learn user expertise, and an expertise-aware task allocation method to maximize the probability that tasks are allocated to users with the right expertise, while ensuring the work load does not exceed the processing capability at each user. Experimental results based on two real-world datasets demonstrate that ETA² significantly outperforms existing solutions.

1.4 Organization

The rest of this dissertation is organized as follows. Chapter 2 presents our strategy for transient community-based data transmission. Chapter 3 focuses on data transmission in networks with diverse connectivity characteristics. Chapter 4 presents our solution for resource-aware truth analysis. In Chapter 5, we present the approach for expertise-aware truth analysis and task allocation. Finally, we conclude the dissertation with final summaries and discuss future directions in Chapter 6.
Chapter 2

Transient Community-based Data Transmission

2.1 Introduction

Community has received considerable attention because of its applications to data transmission in mobile opportunistic networks, worm containment [29], etc. However, it is a challenge to detect communities in a large network [30] [31], especially when considering the temporal information associated with community. For example, a class community consisting of students attending a class may only appear during daytime, and a dormitory community consisting of students at a dormitory may only appear at night. Since these communities normally appear during a time period and disappear thereafter, they are referred to as transient communities (TCs). Although many community detection methods have been proposed in the literature, there is hardly any method for TC detection.

Existing community detection methods are generally based on weighted networks or unweighted networks. For example, algorithms have been proposed in [32][33] to detect communities in weighted networks. Methods like label propagation [34] have been proposed to detect communities in unweighted networks. To detect communities in both weighted and unweighted network, Clique Percolation Method (CPM) or called K-clique [35] has been proposed. Recently, AFOCS [16] has been proposed to detect static communities and track community dy-
Figure 2.1. False mixture: (a) $TC_1$ appears during daytime and $TC_2$ appears at night. (b) The two TCs are falsely mixed into one community using a traditional method.

namics based on unweighted network snapshots. However, it aggregates contact information into a weighted or unweighted network, and then important contact information such as the time when nodes contact is lost. Losing such temporal information may result in two problems related to TC detection: false mixture and false separation as shown in Figure 2.1 and Figure 2.2.

- **False Mixture** There are originally two TCs in the network, as shown in Figure 2.1 (a). $TC_1$ is a class community happening during daytime, and $TC_2$ is a dormitory community happening at night. The two communities share several students, who take classes together and live together. Since there is a large overlap between them, traditional community detection methods may falsely mix these two communities as one. For example, CPM (K-clique) and AFOCS fail to distinguish the two communities when the overlap is larger than some threshold.

- **False Separation** Figure 2.2 (a) shows one TC in the network. Because the network is not strongly connected (i.e., only one node connects the two parts), a traditional method may separate them into two communities. However, they should not be separated since they may indeed attend the same class at the same time.

With false mixture, two highly-overlapping TCs cannot be distinguished by ignoring the temporal contact information. With false separation, one large TC may be falsely separated if node connections are not dense enough. Therefore, traditional community detection methods fail to detect TCs.
In this chapter, we propose a Contact-burst-based Clustering Method (CCM) to detect TCs by exploiting the pairwise contact processes. It is based on the detailed pairwise contact information between nodes, instead of networks with unweighted on-off edges or weighted edges. In CCM, we formulate each pairwise contact process as regular appearance of contact bursts, during which most contacts between the pair of nodes appear. Based on such formulation, we detect TCs by clustering the pairs of nodes with similar contact bursts together. Since it is hard to collect global contact information at individual nodes, we also propose a distributed method. In addition to TC detection, we also apply TCs to data transmission in mobile opportunistic networks. The contributions of this work are summarized as follows:

1. We propose a CCM method to detect TCs. Compared with existing methods such as CPM and AFOCS that do not consider the temporal information of communities, our method has much less false mixtures and false separations.

2. Since it is hard to collect global contact information, we further propose a distributed CCM method to detect TCs. Trace-driven simulation results show that the distributed CCM can effectively detect TCs.

3. A TC may periodically appear, and hence we propose techniques to identify this appearance pattern, which is useful in many applications.

4. We propose a data transmission strategy in mobile opportunistic networks based on TCs, where data is transmitted to TCs with better relaying capabil-
ity to the destination considering the time constraint of the data. Evaluation results show that our approach outperforms other existing data transmission approaches in mobile opportunistic networks.

The rest of the chapter is organized as follows. Section 2.2 presents our CCM method and distributed CCM method for TC detection. In Section 2.3, we present the TC-based data transmission strategy. Section 2.4 gives an overview of the related work, and Section 2.5 concludes the chapter.

2.2 Transient Community Detection

In this section, we first give some preliminaries, and then present our CCM method and the distributed CCM method for TC detection. Finally, we compare our TC detection method with other community detection methods and compare the distributed CCM method with CCM.

2.2.1 Preliminary

In this section, we describe the contact traces and introduce some terms that will be used in this chapter.

2.2.1.1 Contact trace

We study the social contact patterns on three sets of traces, Dartmouth campus trace [36], MIT reality trace [37] and UCSD campus trace [38]. These traces record contacts among users carrying mobile devices on campus. In Dartmouth and UCSD traces, devices are WiFi-enabled. The Dartmouth trace uses SNMP logs from Access Points (APs). The original trace contains records over several thousands of WiFi-enabled devices and lasts almost 4 years. In this chapter, we focus on the data collected from September 2004 to December 2004. A contact is recorded when two devices detect the same AP simultaneously. The UCSD trace records the WiFi association of human-carried PDA with APs, and a contact is recorded when two devices detect the same AP. In MIT Reality trace, the devices periodically detect their peers via Bluetooth interfaces, and a contact is recorded
Table 2.1. Trace summary

<table>
<thead>
<tr>
<th>Trace</th>
<th>Dartmouth</th>
<th>MIT Reality</th>
<th>UCSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network type</td>
<td>WiFi</td>
<td>Bluetooth</td>
<td>WiFi</td>
</tr>
<tr>
<td>Number of devices</td>
<td>425</td>
<td>97</td>
<td>275</td>
</tr>
<tr>
<td>Number of contacts</td>
<td>394254</td>
<td>114046</td>
<td>201923</td>
</tr>
<tr>
<td>Durations(days)</td>
<td>80</td>
<td>246</td>
<td>78</td>
</tr>
<tr>
<td>Granularity(secs)</td>
<td>300</td>
<td>300</td>
<td>20</td>
</tr>
</tbody>
</table>

when two devices move into the communication range of each other. The details of these three traces are shown in Table 2.1.

2.2.1.2 Contact Burst

Simply extracting communities from weighted or unweighted network is not enough to detect TCs. Therefore, instead of simplifying each pairwise contact process to a weight, our method processes the pairwise contact information directly.

From the trace summary, we can see that each trace has hundreds of thousands of contacts and detecting TCs from these contacts directly will be hard. The time complexity will be extremely large, and the opportunistic nature of the contacts hardly provides any clue on how they are related with TCs. Thus, we propose a different solution. We model the pairwise contact process using simple units which can represent the contact information between nodes and can be directly processed. We formulate each pairwise contact process as a series of contact bursts during which most contacts between the pair of nodes appear. A contact burst is defined as follows:

**Definition 1:** A contact burst $T_B = [t_s, t_e]$ between two nodes is a time period when contacts frequently appear between these two nodes. Two adjacent contacts belong to one contact burst if and only if the inter-contact time between them is shorter than $\lambda$, where $\lambda$ is a pre-defined threshold.

Figure 2.3 shows three contact bursts of two nodes, where each vertical arrow indicates a contact and $T_{B_i}$ denotes the $i^{th}$ contact burst. A single contact is not considered as a contact burst because it is more like a random contact.

With the concept of contact burst, each pairwise contact process between two nodes is modeled as a series of contact bursts $T_{B_i} = [t_{s_i}, t_{e_i}], i = 1, 2, 3, \ldots$ To
verify if the contact bursts can really represent node contacts, we have done some experiments based on the three traces to evaluate how many contacts can be represented by contact bursts. Table 2.2 shows the percentage of contacts that are within contact bursts and the length of the contact bursts as a percentage of the total trace duration. The value of $\lambda$ is set empirically based on different traces. We have found that in all of our three traces, with $\lambda = 1$ hour, most contacts happen within some contact bursts which only account for a small portion of the total time. We have also tested other values of $\lambda$, and the results show that the performance is not very sensitive to the change of $\lambda$. For simplicity, we set $\lambda = 1$ hour.

Table 2.2. How contact bursts can represent contact processes ($\lambda = 1$)

<table>
<thead>
<tr>
<th>Trace</th>
<th>Contacts in bursts (%)</th>
<th>Bursts’ length (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dartmouth</td>
<td>66.52%</td>
<td>6.65%</td>
</tr>
<tr>
<td>MIT</td>
<td>77.88%</td>
<td>10.36%</td>
</tr>
<tr>
<td>UCSD</td>
<td>81.22%</td>
<td>4.93%</td>
</tr>
</tbody>
</table>

Contact bursts are used to find TCs. A contact burst $[t_s, t_e]$ between a pair of nodes is a time interval in which the two nodes frequently contact. In a TC with duration $[t_s, t_e]$, its members usually have frequent contacts with each other within this interval; i.e., their contact bursts are all with duration similar to $[t_s, t_e]$. Therefore, if we are able to find a group of contact bursts with similar duration, and they can mutually form a connected graph and form a TC. For example, in Figure 2.4, there are 10 contact bursts, where a double arrow represents a contact burst. Among them, six contact bursts start around 1pm and end around 2pm. Then, they can form a connected graph, and are most likely from the same TC with duration around $[1, 2]$ pm. On the other hand, other four contact bursts in the example are with different contact durations, and are thus not in this TC. Although this example is simple, in a complex network where contact bursts may
Figure 2.4. A connected graph built on contact bursts. Six contact bursts have similar contact durations, and they may be from one TC, circled by red line. The other four have different contact durations, which are not in this TC.

not have the exact same starting and ending time, we need techniques to measure their similarity.

2.2.1.3 The Similarity of Contact Bursts

In order to detect TC, our first objective is to find out contact bursts with similar time periods.

Definition 2: The similarity of two contact bursts $T_{B1} = [t_{s1}, t_{e1}]$ and $T_{B2} = [t_{s2}, t_{e2}]$ is defined as the Jaccard similarity coefficient

$$S(T_{B1}, T_{B2}) = \frac{T_{B1} \cap T_{B2}}{T_{B1} \cup T_{B2}}$$

$$= \begin{cases} \frac{\min(t_{e1}, t_{e2}) - \max(t_{s1}, t_{s2})}{\max(t_{e1}, t_{e2}) - \min(t_{s1}, t_{s2})}, & \text{if } \min(t_{e1}, t_{e2}) > \max(t_{s1}, t_{s2}) \\ 0, & \text{if } \min(t_{e1}, t_{e2}) \leq \max(t_{s1}, t_{s2}) \end{cases}$$

Jaccard similarity coefficient is commonly used to measure the similarity between two equal length “0-1” sequences; i.e., the total number of positions where both sequences have 1 divided by the total number of positions where at least one sequence have 1. Similar technique can be applied to our case when two time intervals are compared. The Jaccard similarity coefficient for two time intervals
are the intersection between them divided by the union of them.

### 2.2.2 TC Detection with CCM

Based on contact bursts and their similarity, we cluster similar bursts together, from which we find TCs. There are many clustering methods in the literature, like *K-means* [39], *spectral clustering* [40] and *hierarchical clustering* [41]. All these methods are aimed to cluster a set of subjects, with the similarity well defined. Using *K-means* algorithm, the number of clusters should be pre-known before clustering. *Spectral clustering* partitions nodes into clusters by using the eigenvectors of a pairwise similarity matrix. Among them, hierarchical clustering is effective and efficient with only one parameter – minimum allowed similarity ($\gamma$) that can be flexibly set. Thus, our CCM is based on hierarchical clustering.

With a set of $n$ contact bursts, CCM runs as follows:

1. **Initialization**: Each of the $n$ contact bursts starts to form its own cluster.

2. **Merge clusters**: Pick two clusters with the largest similarity and merge them together. The similarity of two clusters is defined as the average of all pairwise *Jaccard similarity coefficient* of the contact bursts in these two clusters. This step repeats until the termination condition is satisfied, as defined in the next step.

3. **Termination**: The algorithm terminates if the largest similarity between all clusters in one round is smaller than the minimum allowed similarity $\gamma$.

In order for the algorithm to generate TCs, we need to modify the cluster merging phase. With this algorithm, two clusters with similar happening time may be merged as a TC. However, this is not always true. For example, if two clusters don’t have any common node, it is more likely that they are from two TCs which happen at similar time but at different locations. Thus, we add one condition for cluster merging; that is, the picked clusters must have at least one common node (as shown in Figure 2.5). For all pairs that satisfy this condition, those with the largest similarity are merged.

With CCM, contact bursts with similar time periods are clustered together. One cluster corresponds to one TC, where nodes attached to the contact bursts
Figure 2.5. For two clusters to merge, they must share at least one common node. (a) No common node, nodes $U, V$ and nodes $X, Y$ may come from two different TCs which appear at similar time. (b) Two contact bursts with common node $U$, and these three nodes form a TC.

will be in this TC. When the algorithm terminates, there may be some “tiny” clusters, which only include one contact burst. One contact burst only consists of two nodes, and it is more like a personal meeting rather than a TC. For a contact burst $[t_s, t_e]$, its duration is defined as $t_e - t_s$. For a cluster which includes multiple contact bursts, its duration is defined as the average of the durations of these contact bursts. A short cluster duration may represent an occasional encountering of several nodes, which should not be counted as a community. Thus, we should delete clusters that only have one contact burst, and delete clusters with durations shorter than $T_{min}$, which is a parameter and we set $T_{min} = 0.25$ hour.

**Parameter $\gamma$:** The minimum allowed similarity $\gamma$ will have impacts on the number of TCs and their average size. We have run CCM on the UCSD trace to see the impacts of $\gamma$, and the results are shown in Figure 2.6. As can be seen, decreasing $\gamma$ results in a smaller number of TCs and a larger size. When $\gamma$ is small, the minimum allowed similarity between clusters becomes smaller, and then more clusters will be merged, resulting in fewer TCs with larger size. If $\gamma$ is too large, the clustering process ends quickly, and contact bursts that should belong to the same TC may not be clustered before termination. Thus, it is important to find the right value for $\gamma$.

In a mobile network, contacts that happen between members in the same community are called *intra-community contacts*, and contacts that happen between
members in different communities are called *inter-community contacts*. A community structure is preferred if there are more intra-community contacts and less inter-community contacts. Since TC also has temporal information, a contact is called an intra-community contact in a TC only when the contact happens between members in the same TC and the contact happens during this TC’s existing period.

To obtain the percentage of intra-community contacts, we re-run the experiment with the TCs’ information and count the number of intra-community contacts. The results are normalized by dividing them by the total number of con-
tacts. The percentages of intra-community contacts in a network using different TC threshold $\gamma$ are shown in Figure 2.7, and a larger percentage is preferred. As can be seen, the percentage of intra-community contact is the largest when $\gamma$ is 0.4, and we choose $\gamma = 0.4$ in the rest of the chapter.

### 2.2.3 TC Detection with Distributed CCM

CCM is a centralized detection method, and it needs all the pairwise contact information between nodes in the network. Since it is hard to collect the global contact information at individual nodes, we further propose a distributed CCM. In this subsection, we first give the basic idea of the distributed CCM and then present the detailed algorithm.

#### 2.2.3.1 Basic Idea

Similar to CCM, the distributed CCM also detects TCs by clustering the contact bursts. Since it is hard to collect all the contact bursts at individual nodes, each node only records the contact bursts with its neighbors, which are referred to as local contact bursts. For a node $v$, if it has frequent contacts with node $w$ in the recent time period and the adjacent contacts have intervals smaller than $\lambda = 1$ hour, there exists a contact burst between $v$ and $w$. An indicator $I_{e(v,w)}(t) = \{true, false\}$ is used to indicate whether the contact burst between $v$ and $w$ exists at time $t$.

In addition to local contact bursts, each node also maintains the TCs it has detected locally, which are referred to as local TCs or local clusters (a TC is actually a cluster of contact bursts). The local clusters detected at node $v$ should include $v$ as a member and have pairwise similarities smaller than the minimum allowed similarity $\gamma = 0.4$; otherwise, similar clusters should be further merged. With this rule, node $v$ adds new clusters to its local clusters in the following two ways:

- **Adding local contact bursts:** As node $v$ finishes its contact burst with another node $w$ (i.e., the last contact between $v$ and $w$ happens $\lambda = 1$ hour before), the contact burst $T_B^{(v,w)}$ is treated as a new cluster $C_{new} = \{T_B^{(v,w)}\}$ and added to the local clusters of $v$. Since the new cluster may have similar duration
with the existing local clusters (with similarity larger than \( \gamma \)), the similar clusters should be merged.

- Adding local clusters of other nodes: Simply using the local contact bursts to detect TCs is not enough. In addition, node \( v \) learns about the local clusters of other nodes upon the pairwise contacts. Specifically, as node \( v \) contacts node \( w \), it exchanges the local clusters with node \( w \). Afterwards, node \( v \) selects some clusters from \( w \) and merges with the local clusters at \( v \).

### 2.2.3.2 The Distributed Algorithm

When a node \( v \) initializes the TC detection process, it has no local contact bursts and local clusters detected. Whenever node \( v \) encounters another node \( w \), it updates the local contact bursts and exchanges the local clusters with \( w \). Then node \( v \) uses the new information to update its local clusters. Algorithm 1 outlines the process of updating the local clusters as node \( v \) contacts node \( w \) at time \( t \). We first give the notations used in the algorithm, and then discuss the detailed process and the functions used in the algorithm.

At node \( v \), \( T_B^{(v,k)} = [t_s^{(v,k)}, t_e^{(v,k)}] \) is used to denote the contact burst with node \( k \) \((k \neq v)\), and \( I_e^{(v,k)}(t) \in \{true, false\} \) indicates whether \( T_B^{(v,k)} \) exists or not at time \( t \). The set of local clusters at \( v \) is denoted as \( C^v \). Each cluster \( C \) in \( C^v \) consists of a set of contact bursts. \( S(C_1, C_2) \) denotes the similarity of two clusters. If \( C_1 \) and \( C_2 \) do not share any common node, the similarity is 0; otherwise, it is computed as the average of all pairwise Jaccard similarity coefficient of the contact bursts in the two clusters. \( N \) denotes the set of nodes in the network.

Algorithm 1 consists of two steps in general. In the first step (Line 1-16), node \( v \) updates the local contact bursts at time \( t \). Specifically, it first updates if each of the local contact bursts exists at time \( t \). If \( I_e^{(v,k)}(t) = true \) and the last contact in \( T_B^{(v,k)} \) happened \( \lambda \) before \((t - t_e^{(v,k)} > \lambda)\), it means \( T_B^{(v,k)} \) has finished and \( I_e^{(v,k)}(t) \) is changed to \( false \). Corresponding to each of the finished contact bursts, a new cluster is created and added to the local clusters. As the new cluster may have similar duration with existing local clusters, we should check if it should be merged with these clusters. The function \( NewCluster \), which is shown in Algorithm 2 and will be discussed later, is used to determine how the new cluster is merged with

...
Algorithm 1 Updating the local clusters when node $v$ contacts node $w$ at time $t$

1: /* Update the contact bursts with other nodes. */
2: for $k \in N \setminus v$ do
3: if $I_e^{(v,k)}(t) = \text{true}$ and $t - t_e^{(v,k)} \geq \lambda$ then
4: \quad $I_e^{(v,k)}(t) \leftarrow \text{false};$
5: \quad $C_{\text{new}} \leftarrow \{T_B^{(v,k)}\};$
6: \quad $C' \leftarrow C' \cup \{C_{\text{new}}\};$
7: \quad NewCluster($C', C_{\text{new}}$);
8: end if
9: end for
10: /* Update the contact burst between $v$ and $w$. */
11: if $I_e^{(v,w)}(t) = \text{false}$ then
12: \quad $I_e^{(v,w)}(t) \leftarrow \text{true};$
13: \quad $t^{(v,w)} \leftarrow t, t_e^{(v,w)} \leftarrow t;$
14: else
15: \quad $t_e^{(v,w)} \leftarrow t;$
16: end if
17: /* Exchange the local clusters between $v$ and $w$ */
18: $C' \leftarrow$ the set of local clusters in node $w$;
19: MergeClusters($C', C'$);

the existing clusters.

Node $v$ then updates the contact burst with the encountered node $w$ by incorporating the contact at time $t$. If $I_e^{(v,w)}(t) = \text{false}$, a new contact burst begins at time $t$. Otherwise, the contact burst $T_B^{(v,w)}$ is extended to time $t$ by including the current contact ($t_e^{(v,w)} \leftarrow t$).

In the second step (Line 17-19), node $v$ exchanges the local clusters with node $w$ and checks if the local clusters of $w$ can be merged with the local clusters of itself. Node $v$ uses a function $\text{MergeClusters}$ (which is shown in Algorithm 3) to decide how to merge the clusters in node $w$ to the local clusters. Generally, with $\text{MergeClusters}$, only clusters having high similarities with the local clusters are merged, since the clusters with low similarities usually have no relation with node $v$.

Algorithm 1 only shows the updating process at node $v$ as two nodes $v$ and $w$ contact. The updating process at node $w$ is similar to the process at node $v$ and therefore skipped here. Next, we discuss the two functions utilized in Algorithm 1: $\text{NewCluster}$ and $\text{MergeClusters}$. 
**Algorithm 2** Function 1 - *NewCluster*

1: function NewCluster($\mathcal{C}, C_{\text{new}}$)  
2: /* Find in $\mathcal{C}$ the most similar cluster $C_m$ to $C_{\text{new}}$. */  
3: $S_m \leftarrow 0$;  
4: for $C \in \mathcal{C} \setminus \{C_{\text{new}}\}$ do  
5: if $S(C, C_{\text{new}}) > S_m$ then  
6: $S_m \leftarrow S(C, C_{\text{new}})$;  
7: $C_m \leftarrow C$;  
8: end if  
9: end for  
10: if $S_m < \gamma$ then return  
11: else  
12: $C_m \leftarrow C_m \cup C_{\text{new}}$;  
13: $\mathcal{C} \leftarrow \mathcal{C} \setminus \{C_{\text{new}}\}$;  
14: NewCluster($\mathcal{C}, C_m$);  
15: end if

**Algorithm 3** Function 2 - *MergeClusters*

1: function MergeClusters($\mathcal{C}_1, \mathcal{C}_2$)  
2: $S_m \leftarrow 1$;  
3: while $S_m \geq \gamma$ do  
4: /* Find the pair of clusters with highest similarity. */  
5: $S_m \leftarrow 0$;  
6: for $C_1 \in \mathcal{C}_1$ do  
7: for $C_2 \in \mathcal{C}_2$ do  
8: if $S(C_1, C_2) > S_m$ then  
9: $S_m \leftarrow S(C_1, C_2)$;  
10: $C_m \leftarrow C_2$;  
11: end if  
12: end for  
13: end for  
14: $C_{\text{new}} \leftarrow C_m$;  
15: $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \cup \{C_{\text{new}}\}$;  
16: NewCluster($\mathcal{C}_1, C_{\text{new}}$);  
17: end while

- **NewCluster**: The process of *NewCluster* is outlined in Algorithm 2. The objective is to check if the new cluster $C_{\text{new}}$ should be merged with one of the existing clusters. It works by finding the cluster $C_m$ that has the maximum similarity with $C_{\text{new}}$. If the similarity is smaller than $\gamma$, nothing needs to be done and the function returns. Otherwise, $C_{\text{new}}$ is merged with $C_m$ by adding the contact bursts in $C_{\text{new}}$ to $C_m$. $C_{\text{new}}$ is then deleted from the local clusters.
Since $C_m$ has been updated by adding more contacts bursts, it is treated as another new cluster and the function $NewCluster$ will be recursively called to check whether the updated cluster should be merged with other clusters.

- **MergeClusters**: The process of $MergeClusters$ is outlined in Algorithm 3. Given two sets of local clusters $C_1$ and $C_2$, $MergeClusters$ is used to select the similar clusters in $C_2$ and merge with the clusters in $C_1$. The function works in an iterative way. In each iteration, the pairwise similarities between clusters in $C_1$ and $C_2$ are calculated. Then, for the pair of clusters that has the largest similarity, if the similarity is not smaller than $\gamma$, the cluster from $C_2$ is selected and added to $C_1$ as a new cluster. Afterwards, the function $NewCluster$ is used to check how the new cluster should be merged with the existing clusters in $C_1$.

After all contacts have been processed, we use the same procedures as CCM to delete the clusters that only have one contact burst and delete clusters with duration shorter than $T_{min}$. Finally, each cluster corresponds to a TC and the nodes attached to the contact bursts will be in this TC.

### 2.2.4 Evaluations

#### 2.2.4.1 Evaluations of CCM

We compare the performance of our TC detection algorithm CCM with existing commonly used community detection algorithms: CPM (K-clique) and AFOCS. The performance is compared based on six metrics, covering various properties of community. Here, we only show the results based on the UCSD trace, since results on other traces are similar.

**Number of communities and community size**: Figure 2.8 (a) shows that CCM can detect much more TCs than communities detected by CPM and AFOCS. This indicates that CPM and AFOCS may have mixed some TCs to one community. Figure 2.8 (b) shows that communities detected by CPM are usually bigger than those detected by AFOCS and CCM. This further confirms that CPM may have mixed some TCs, making the overall community size larger.
Proportion of nodes involved in community: Community structure is usually used to help data transmission in mobile opportunistic networks; thus it is preferable if more nodes can be included in the community. Figure 2.8 (c) shows that CCM usually involves more nodes than CPM and AFOCS. In CPM and AFOCS, the node that has infrequent contacts with others may be ignored.

Number of associated communities for one user: We use this metric to show how much communities are overlapped. From Figure 2.8 (d), we can see that a node is normally attached to one community in CPM or AFOCS. This means communities are almost mutually disjoint, which does not achieve the objective of detecting overlapping communities. However, TCs detected in CCM have good overlapping property where a node belongs to an average of three TCs. Because TCs are detected using temporal information, it has no limitation on how much communities overlap. Other methods have limits on how much two communities
can overlap.

**Proportion of intra-community contacts:** We prefer a community structure which can incorporate more contacts. For CPM and AFOCS, a contact is considered as an intra-community contact when it appears between two members in the same community. We need another temporal requirement for a contact in TC to be considered as an intra-community contact; that is, it must happen within the community’s duration. Therefore, we can see that TC incorporates less intra-community contacts than CPM and AFOCS as shown in Figure 2.8 (e). CPM has the highest number of intra-community contacts. This can be explained by the fact that there are many large communities detected by CPM, which makes more contacts counted as intra-community contacts.

**Location distortion:** It measures how much is the difference among nodes’ locations within a community. Intuitively, a smaller location distortion means that users in the same community are near each other. On the contrary, a larger location distortion means that users may appear at multiple places in this community. A community’s location distortion is the standard deviation of the locations of all intra-community contacts. A contact’s location is calculated by averaging two node locations at the contact time.

\[
\text{Loc}_{<i,j>} = \frac{\text{Loc}_{i,t_{<i,j>}} + \text{Loc}_{j,t_{<i,j>}}}{2}
\]

A community’s mean location is calculated by averaging the locations of all intra-community contacts.

\[
\text{Loc}(C) = \frac{\sum_{<i,j> \in C} \text{Loc}_{<i,j>}}{\sum_{<i,j> \in C} 1}
\]

The location distortion within a community \( C \) is defined as the standard deviation of all intra-community contacts’ positions.

\[
\text{Distortion}(C) = \text{stddev}\{\text{Loc}_{<i,j>}\} \quad (<i,j> \in C)
\]

\[
= \sqrt{\text{var}\{\text{Loc}_{<i,j>}\}} \quad (<i,j> \in C)
\]

\[
= \sqrt{\frac{\sum_{<i,j> \in C} (\text{Loc}_{<i,j>} - \text{Loc}(C))^2}{\sum_{<i,j> \in C} 1}}
\]
The UCSD trace does not record users’ location information, so we estimate user’s locations through the detected APs’ locations at that time. From Figure 2.8 (f), we can observe that the location distortion in TCs detected by CCM is smaller than that by CPM and AFOCS; i.e., nodes usually gather at one place in a TC. In a traditional community-based data transmission approach, data is intended to be transmitted to the communities that include the destination node. In such an application, a smaller location distortion is preferred, because this means nodes are near to each other in one community. Once the data reaches the destination community, it must be near the destination node. In communities detected by CPM and AFOCS, nodes do not have a gathering period, so the contacts between them can appear at any time and any place, which results in a large location distortion.

2.2.4.2 Evaluations of Distributed CCM

We next evaluate the performance of the distributed CCM based on the mobility traces.

**Influence of running time:** Using distributed CCM, TCs are updated upon pairwise contacts. With a longer running time, each node encounters more nodes and more TC information can be learned. In the first experiment, we vary the running time and examine whether increasing the running time will enhance the performance of distributed CCM. Here, the performance is evaluated based
Comparing with CCM: We next compare the distributed CCM with CCM. The performance is first compared by measuring the proportion of intra-community contacts. As shown in Figure 2.10 (a), distributed CCM incorporate 80% – 90% as much contacts as CCM in all three traces. This demonstrates that the TCs detected by the distributed CCM (referred to as distributed TC) can well represent the TCs detected by the CCM (referred to as centralized TCs).

We further evaluate the distributed TCs by measuring their similarities to the centralized TCs. To measure the similarity of two communities, previous work [42] calculates the the Jaccard similarity coefficient between the members in the two communities. Since TCs also consider the duration of the communities, the similarity on duration should also be considered. Similarly, the similarity on durations is calculated using the Jaccard similarity coefficient. The similarity between two TCs $TC_1$ and $TC_2$ is calculated as the average of similarities on members and
durations:

\[ S_{m,t}(TC_1, TC_2) = \frac{S_m(TC_1, TC_2) + S_t(TC_1, TC_2)}{2} \] (2.1)

where \( S_m(TC_1, TC_2) \) and \( S_t(TC_1, TC_2) \) respectively denote the similarities on members and durations.

To measure the similarity of a distributed TC \( TC_d \) to the centralized TCs, we find the corresponding centralized TC \( TC^*_d \) that has the highest similarity to \( TC_d \). It is preferable if every distributed TC can find a corresponding centralized TC with high similarity. Otherwise, it usually means the detected distributed TC is inaccurate or unnecessary.

The distribution (CDF) of the similarities of distributed TCs to centralized TCs is displayed in Figure 2.10 (b). As can be seen, for the traces of MIT and Dartmouth, more than 80% distributed TCs can find corresponding centralized TCs with similarity larger than 0.7. While for the traces of UCSD, about 50% distributed TCs have similarities larger than 0.7. Thus, most of the detected distributed TCs can find the corresponding centralized TCs with high similarity. In the meantime, for all three traces, there exist some distributed TCs with very low similarities to the centralized TCs, which implies that the distributed CCM may detect some TCs that are inaccurate or unnecessary.

2.3 Application to Data Transmission

Community has been widely used for data transmission in mobile opportunistic networks. However, ignoring the community appearance time may lead to non-optimal transmission paths. For example, in community-based data transmission, data is always intended to be transmitted to a node within the destination’s community. This is not optimal when considering two problems. First, because traditional community usually has a large location distortion, delivering data to destination’s community does not mean it is getting closer to the destination node. Second, considering the appearance time of communities, some of the destination communities to which data are transmitted may not even appear before data expire. Our \textit{TC-based data transmission strategy} solves these problems by utilizing TCs
as the transmission unit and always transmitting data to TCs that have a better capability of relaying data to the destination node within a short time constraint.

Utilizing TCs to data transmission requires the knowledge on TC appearance patterns. In this section, we first study the periodic appearance patterns of TCs and discuss how to compute the relaying capabilities, and then present the TC-based data transmission strategy.

2.3.1 Periodic Appearance of Transient Communities

With the proposed CCM, we can detect TCs on a daily basis. We run the algorithm based on the traces and find that there are many TCs sharing similar members and appearing at different times. These TCs represent one social group which appears periodically, like a class. In addition to these periodically appearing TCs, the majority of TCs only appear once. Such randomly appearing communities are usually formed due to the opportunistic meeting of unfamiliar nodes. Later, we will no longer consider these opportunistically formed TCs. In this subsection, we will identify the periodic appearance of TCs based on the traces and exploit their appearance patterns.

2.3.1.1 Identifying the periodic appearances of TCs

The algorithm we use to cluster similar TCs is the same hierarchical clustering as we presented in Section 2.2.2. TCs are clustered according to the similarity of
their members, which is also based on the Jaccard similarity coefficient.

We are interested in knowing how many times each TC appears. We run the clustering algorithm on three traces, each with 75 days of data. The CDFs of TCs’ appearance times are shown in Figure 2.11. TCs that only appear once are not included in the figure. As can be seen, there are more than 10% TCs in the Dartmouth trace and about 5% TCs in the MIT and UCSD traces which appear more than 10 times. Meanwhile, there are also some TCs appear more than 30 or 50 times. Given the frequent appearances of TCs, the next question is how to find the appearance patterns of TCs.

2.3.1.2 Appearance patterns of TCs

In this subsection, we study the periodic appearance patterns of TCs on three sets of traces. We formulate the appearance patterns of TCs on daily basis, based on the fact that many social groups are formed daily such as families, offices, and classes. Although there are many social groups formed on a weekly or monthly base or on a more complex pattern, we will not consider these patterns here and leave them as future work. The appearance pattern is formulated in two aspects. One is the distribution of the TC starting time, and the other is the distribution of TC duration.

We have two observations. First, the starting time of a TC within a day can be well approximated by a normal distribution. We take one TC in the Dartmouth trace as an example, shown in Figure 2.12 (a), and the parameters of the
distribution is shown in Table 4.1. The TC usually appears at around 1 to 3 pm. Second, the duration of a TC can be approximated by an exponential distribution, as shown in Figure 2.12 (b). The parameter $\lambda$ is 0.374, which means that the TC has an average duration of $1/\lambda = 2.674$ hours. In our traces, a TC only appears with limited number of times, and hence the samples used to train the distributions are limited. Therefore, the approximation does not seem perfect, especially for the normal distribution. We believe the approximation should be better if more data are used.

<table>
<thead>
<tr>
<th>Table 2.3. The parameters of normal and exponential distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>normal($\mu, \sigma^2$)</td>
</tr>
<tr>
<td>$\mu = 14.256$</td>
</tr>
</tbody>
</table>

**Determining the probability of TC appearance within a time interval:**
Given a time interval $[t_s, t_e]$, we are interested in determining the probability that a TC will appear. It is affected by three factors: the probability that the TC appears in that day, the distribution of its start time in that day, and the distribution of its duration. The probability that a TC appears in a day is simply calculated as the percentage of days when the TC has appeared during the warm-up period. Based on this, whether the TC appears within the pre-defined interval includes two possibilities. The first possibility is that the TC starts in the time period $[t_s, t_e]$, and the second is that the TC starts before the time interval but lasts until the start of the time interval. Suppose the start time is represented by a normal distribution with parameters $\mu$ and $\sigma$, and the duration is represented by the exponential distribution with parameter $\lambda$. The probability of the first possibility is:

$$P_1[t_s, t_e] = \int_{t_s}^{t_e} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-t_s)^2}{2\sigma^2}} dt$$

$$= \text{normcdf}(t_e, \mu, \sigma^2) - \text{normcdf}(t_s, \mu, \sigma^2)$$

where $\text{normcdf}(t, \mu, \sigma^2)$ is the CDF of the normal distribution with mean $\mu$ and
standard deviation $\sigma$. The probability of the second possibility is:

$$P_2[t_s, t_e] = \int_0^{t_s} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(t-s)^2}{2\sigma^2}} * e^{-\lambda(t_s-t)} dt$$

$$= e^{-\lambda(t_s-\mu)+\frac{2\sigma^2}{2}} \int_0^{t_s} \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(t-s+\frac{2\sigma^2}{2})^2}{2\sigma^2}} dt$$

The part under the integral is actually the probability density function of the distribution $\text{norm}(\mu + \lambda \sigma^2, \sigma^2)$. Thus, the integral can be easily computed by $\text{normcdf}(T_1, \mu + \lambda \sigma^2, \sigma^2) - \text{normcdf}(0, \mu + \lambda \sigma^2, \sigma^2)$.

Then, the probability of the TC appearance within this time interval is:

$$P_{TC}[t_s, t_e] = p_d * (P_1[t_s, t_e] + P_2[t_s, t_e]) \quad (2.2)$$

where $p_d$ is the probability that the TC appears in the day.

### 2.3.2 Relaying Capability

By estimating the current TC’s capability of transmitting data to the destination node within a future time period $[t, t+T]$, we can choose users in TCs with better transmission capability as the data carriers. We denote the destination node as $d$ and the current TC as $TC_c$. The destination TCs that $d$ belongs to are denoted as $TC_d^1, ..., TC_d^n$. Specifically, we first compute the relaying capability of $TC_c$ to each of $TC_d^1, ..., TC_d^n$ within $[t, t+T]$ respectively. Then we can obtain the relaying capability from $TC_c$ to $d$ by summing the computed relaying capabilities to $TC_d^1, ..., TC_d^n$.

We next discuss in detail how to compute the relaying capability from $TC_c$ to one of the destination TCs ($TC_d^i$). The relaying capability from $TC_c$ to $TC_d^i$ within the time period $[t, t+T]$ is computed by summing the probability that each node of $TC_c$ will appear in $TC_d^i$ in $[t, t+T]$. It is determined by the number of common nodes $k$ between them and the probability that $TC_d^i$ will appear in $[t, t+T]$. When we compute the relaying capability from $TC_c$ to $TC_d^i$ within time period $[t, t+T]$, it is implied that $TC_c$ has appeared at time $t$. The probability that $TC_d^i$ will appear before $t+T$, denoted as $P_{TC_d^i}[t, t+T]$, is computed in Equation (2.2). The relaying
capability from $TC_c$ to $TC_d^i$ is computed as:

$$R_{TC_c \rightarrow TC_d^i}[t, t + T] = k \times P_{TC_d^i}[t, t + T]$$ (2.3)

The relaying capability from the current TC $TC_c$ to the destination node $d$ is decided by accumulating the relaying capability from $TC_c$ to all the destination TCs $d$ belongs to, which are $TC_d^1, ..., TC_d^{n_d}$. The relaying capability from $TC_s$ to $d$ is computed as:

$$R_{TC_c \rightarrow d}[t, t + T] = \sum_{i=1}^{n_d} R_{TC_c \rightarrow TC_d^i}[t, t + T]$$ (2.4)

2.3.3 TC-based Data Transmission Strategy

In this work, we focus on unicast in mobile opportunistic networks, where the source node $s$, the destination node $d$, data’s initialization time $T_{in}$ and data’s expiration time $T_{ex}$ are given. The objective is to transmit the data item from $s$ to $d$ within the time period $[T_{in}, T_{ex}]$. Our TC-based data transmission strategy is based on the relaying capability of TCs in the recent time period $[t, t + T]$. That is, our strategy always transmits data to a TC that has better relaying capability. Transmission decisions are made upon node contacts. If a node meets another node which is in a TC that has larger relaying capability to destination, data will be transmitted. Since whether to transmit data depends on the current TC it belongs to, it’s important for the node to know which TC it’s currently in.

2.3.3.1 Finding the Current TC

To keep track of the TCs that a node is in, each node keeps a queue ($Q_w$) of the recently met nodes and the contact time within the last time window ($T_w$). $T_w$ is a constant and how to set the value of $T_w$ will be discussed in Section 2.3.5.2. Based on $Q_w$ and all TCs that it has ever known, a matching score is assigned to each TC. The node will consider the TC with the largest matching score as the TC it is currently in. The matching score is computed as follows:

$$Score(TC) = \frac{\sum_{i \in Q_w \land i \in TC} (T_w - (t - t_i))}{|TC|}$$ (2.5)
where $t$ is the current time, and $t_i$ is the node’s contact time with node $i$ in queue.

Even though the distributed CCM method can also detect TCs distributedly, we do not use it to identify the current TC in the experiment, since it cannot detect TCs promptly considering the updates on contact bursts usually have delays. Therefore, we use the CCM or distributed CCM to detect TCs during the warm-up periods. While during the experiment, the aforementioned method is used to quickly identify the current TC by matching to the TC detected in the warm-up periods.

2.3.3.2 Data Transmission based on TCs

As TC is the data transmission unit, data is always transmitted to the TC with better relaying capability to the destination. Therefore, once data reaches a new TC with larger relaying capability, the data is distributed to all nodes met in the TC. However, nodes do not carry the data permanently, and they only carry the data for a time period $T$. If it no longer contacts any node with larger relaying capability or it reaches a TC with larger relaying capability, the data will be deleted at the end of $T$. The detailed data transmission process is shown in the following steps:

When node $v_1$ contacts node $v_2$ at time $t$,

1. Update $Q_w$ for $v_1$, and find the TC that $v_1$ is currently in: $TC_1$.

2. For each data that $v_1$ carries.

   (a) Check if the data should be deleted from $v_1$’s buffer. The data should be deleted when the node has neither contacted a node in TC with larger relaying capability nor gone to a TC with larger relaying capability in the period $[t - T, t]$. Once the data item is deleted, check the next data item.

   (b) Check if the data is carried by $v_2$. If so, skip this data item.

   (c) Check $v_2$’s current TC, $TC_2$. If $TC_2$ and $TC_1$ are the same, $v_1$ transmits data to $v_2$ so that the data can be distributed in the current TC. If $TC_2$ has better relaying capability to $d$, $v_1$ also transmits data to $v_2$. 
2.3.4 Performance Evaluations

We evaluate the performance of the TC-based data transmission strategy with the Dartmouth trace, the MIT Reality trace, and the UCSD trace, as presented in Section 2.2.1.1. The second half of trace is used for data transmission experiments. The value of the time period $T$, with which we evaluate TCs’ relaying capability and decide when to delete data, is empirically set at each trace. We set $T = 1$ hour in all three traces with which we can achieve high delivery ratio with small number of data copies as can be seen in the following results. For each trace, we use the first half as the warm-up period, during which the TCs and TCs’ appearing patterns are detected and related information are transmitted among nodes. The second half of the trace is used to evaluate the performance. For each data item, sources and destinations are picked randomly and the generation time is randomly chosen in the daytime, since nodes’ activity remains low at night which may results in inaccurate comparison. Each experiment is repeated 1000 times for statistical convergence.

Our TC-based data transmission approach is compared with three traditional community-based transmission approaches and the Epidemic approach which serves as the upper bound. A brief overview of these approaches is shown below:

- **Epidemic** [43]: The data item is always transmitted to another node if it does not have the data. This method has the best data delivery ratio and the highest network overhead, which is used as an upper bound for comparison.

- **Label** [15]: This strategy is the first proposed community-based data transmission strategy. In this strategy, the data item is transmitted to nodes that are within at least one common community with the destination node. CPM (K-clique) is used to detect communities.

- **Bubble Rap** [14]: This strategy uses both centrality and community. CPM (K-clique) is used to detect communities. The data item is always transmitted to a higher centrality node, until it reaches a node that belongs to the same community as the destination node. When the data item reaches the destination community, it is transmitted to higher-centrality node within the community’s scope, until the destination node is reached.
Figure 2.13. Data delivery ratio of various methods based on three traces

Figure 2.14. Data transmission overhead measured by the number of data copies

- **AFOCS**: The strategy was proposed in [16], and it is used to evaluate the communities detected by the AFOCS method. This method is based on how many common communities a node has with the destination node. The data item is only transmitted when the contacted node has more common communities with the destination node than the original carrier.

The performance is measured with two metrics: one is the data delivery ratio and the other is the network overhead. Data delivery ratio is the proportion of data items successfully delivered before the data expires. The network overhead is the average number of data copies existing in the network at each moment.

In the first experiment, we comprehensively compare the performance of the community-based strategies. The communities and TCs used in these strategies are all detected using the centralized method. The results are shown in Figure 2.13 and Figure 2.14. Generally speaking, our TC-based method has much better performance than the other three community-based methods by achieving much larger delivery ratio with comparable network overhead.
From Figure 2.13 we can see TC-based strategy consistently achieves better performance than the other community-based data transmission algorithms and even get comparable delivery ratio with Epidemic when the time constraint is small. The good performance in small time constraints is due to the fact that we have studied and predicted users’ behavior in TCs within a short time period $T$ and transmit data to nodes in TCs that have good relaying capability to destination within $T$.

Figure 2.14 shows the overhead generated by each approach. Epidemic always has the highest overhead among all the approaches. Our TC-based strategy generates comparable overhead as other three community-based approaches. Since the TC-based strategy removes data when there is no better TC formed or contacted within $T$, the network overhead is kept low.

In the second experiment, we evaluate the performance of the TC-based strategy where the TCs are detected using distributed CCM (referred to as distributed $TC$-based strategy). The strategy is compared with the TC-based strategy where TCs are detected using the centralized CCM (referred to as $TC$-based strategy). We also compare it with the Bubble Rap strategies where communities are respectively detected using the centralized method and distributed method (referred to as Bubble Rap and distributed Bubble Rap). The results based on the MIT trace are shown in Figure 2.15, and the performances in other traces are similar.

As shown in Figure 2.15, the distributed TC-based strategy achieves compa-
rable delivery ratio with the TC-based strategy. This confirms our observation that most centralized TCs can be represented by the distributed TCs. However, the overhead of the distributed TC-based strategy is higher. The possible reason is that there are some unnecessary TCs detected by the distributed CCM, which causes more data copies created inside TCs. Compared with Bubble Rap and distributed Bubble Rap, our distributed TC-based strategy has much higher data delivery ratio. For network overhead, the distributed TC-based strategy has higher overhead as the time constraint becomes shorter. As time constraint increases, the overhead of the distributed TC-based strategy decreases and it can be even smaller than the Bubble Rap strategies.

2.3.5 Discussions

2.3.5.1 Effect of Prediction on Performance

In our TC-based data transmission strategy, predicting when and whether a TC will happen in the future is important. With prediction, we can compute the relaying capability of each TC to the destination node and then use the nodes in TCs with good relaying capability to carry data. To evaluate how the prediction affects the performance, we compare our strategy with a TC-based strategy without prediction (i.e., only distributing data when a node finds that the contacted node or itself is within the same TC of the destination node). Figure 2.16 compares their data delivery ratios. We can clearly see the superiority of the strategy with prediction.

Figure 2.16. The impact of prediction in the TC-based data transmission approach.
2.3.5.2 Effect of Time Window \( T_w \)

In Section 2.3.3.1, we used the information of recently met nodes in the last time window \( T_w \) to identify the current TC. To find the optimal value of \( T_w \) in each trace, we conduct an experiment to see how \( T_w \) impacts the data delivery ratio. Here, the data delivery ratio is recorded with a time constraint of 12 hours. The results are shown in Table 2.4. As can be seen, \( T_w = 0.5 \) hour has the best data delivery ratio for all traces.

<table>
<thead>
<tr>
<th>( T_w )</th>
<th>0.5</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dartmouth</td>
<td>0.248</td>
<td>0.217</td>
<td>0.205</td>
</tr>
<tr>
<td>MIT</td>
<td>0.115</td>
<td>0.114</td>
<td>0.112</td>
</tr>
<tr>
<td>UCSD</td>
<td>0.261</td>
<td>0.26</td>
<td>0.26</td>
</tr>
</tbody>
</table>

2.4 Related Work

Complex networks usually consist of communities. Lots of research has been done on detecting communities by interdisciplinary researchers from physics, biology and computer science. They originally aimed at identifying communities in a static network. There are many methods focused on detecting disjoint communities such as the modularity-based methods [44][32] and label propagation [34]. Techniques have also been proposed to detect overlapping communities such as CPM [35], link partitions [45] and AFOCS [16], and a survey of existing community detection methods is given in [30]. The aforementioned community detection methods are all centralized methods, which requires the global network information is known a priori. Considering the global network information is hard to obtain in the mobile networks such as mobile opportunistic networks, Hui et al. [42] proposed distributed community detection methods based on the existing centralized methods. In addition, Clauset et al. [46] also proposed methods to detect local communities by exploring the graph near an individual vertex.

Although community detection has been well studied in static networks, it remains difficult to detect time-varying communities in a dynamic network such as
a social network in which people’s behavior is highly dynamic. Recently, more research has been focused on studying evolving community structures based on the network structures at successive network snapshots. Palla et al. [47] developed an algorithm to identify evolving communities by first detecting overlapping community structure at each network snapshots and then mapping similar communities at different snapshots. Nguyen et al. [16] proposed AFOCS, which can adaptively update the community structure at each network snapshot based on history without detecting communities again at each snapshot. Similar methodologies are also utilized in [48][49][50] to detect evolving communities based on the networks at different snapshots. These methods are intended to analyze the long-term evolution of community structures caused by the permanent change of humans’ behaviors or habits. There is still a dearth of work examining transient communities caused by the periodic change of human behavior.

Although Gao et al. proposed the concept of transient community in [18], they only investigated the community relationship between individual pairs of nodes without providing the complete knowledge about the transient community structure. Pietilanen et al. [51] proposed algorithms to detect temporal communities that are similar to transient communities. The basic idea is to extract static community structure from each network snapshot. However, the time interval of the network snapshot is hard to determine; thus it is hard to accurately detect transient communities. Our work is not based on network snapshot but a careful study of the pairwise nodes’ contact processes, and thus can accurately detect transient communities.

Community structure has been extensively utilized to address the problem of data transmission in mobile opportunistic networks. It is believed that nodes within the same community have higher chance to contact each other. Pan et al. in [15] and [14] have proposed to transmit data to the nodes that are within the destination’s community based on the static community structure. By introducing dynamic community structure, AFOCS [16] further considered community evolution in the community-based data transmission. Transient communities in [18] were only used to determine the scope for evaluating node centrality. [51] studied how temporal communities contribute to data dissemination and concluded that temporal communities tend to limit data dissemination in mobile opportunistic
networks. As far as we know, our work is the first work to fully utilize transient communities for data transmission.

2.5 Conclusions

In this chapter, we proposed a contact-burst-based clustering method (CCM) to detect TCs by exploiting the pairwise contact processes. We formulated each pairwise contact process as regular appearances of contact bursts during which most contacts between the pair of nodes appear. Based on such formulation, we detect transient communities by clustering the pairs of nodes with similar contact bursts. Trace-driven simulations showed that CCM can detect TCs more effectively compared with existing community detection methods. We also proposed a distributed CCM method to make the TC detection feasible in mobile opportunistic networks, and demonstrated that it can effectively detect TCs. Finally, the concept of TCs are applied to data transmission in mobile opportunistic networks, where data is transmitted to TCs that have better relaying capability to the destination node. Trace-drive simulations showed that our approach outperforms traditional community-based data transmission approaches.
Data Transmission in Networks with Diverse Connectivity Characteristics

3.1 Introduction

Mobile opportunistic networks with diverse connectivity characteristics is a combination of sparsely connected network and mobile ad hoc network (MANET). This is because there is ubiquitous existence of *Transient Connected Components (TCCs)*. Inside a TCC, a MANET is formed where nodes can reach each other by multi-hop wireless communications. Outside of TCC, nodes contact each other opportunistically through “carry-and-forward”. However, existing data transmission strategies are all based on the assumption that mobile opportunistic networks are highly sparse, which makes these strategies far from optimal in networks with diverse connectivity characteristics. For example, Figure 3.1 (a) shows a TCC of five nodes and Figure 3.1 (b) shows their forwarding metrics which represent the capability of forwarding data to other nodes (e.g., node centrality [17]). Node $A$ has the data which will be forwarded to the destination through “carry-and-forward”. Based on existing techniques, the data should be forwarded to a contacted node with higher forwarding metric. In this example, node $A$ has two possible contacts: $B$ and $E$ (contacts are represented by lines). Since node $B$ has lower forwarding metric (2.1) than $A$’s (3.0), $B$ will not get the data. $E$ has higher forwarding metric (4.5) than $A$, and thus $A$ forwards the data to $E$. After $E$ receives the data, its
Figure 3.1. The left figure shows a TCC of five nodes. The right table shows their forwarding metrics. A line between two nodes means that they are within communication distance.

contact $D$, which has higher forwarding metric (5.8), will get the data. $D$ will not forward the data to its contact $B$ which has a lower forwarding metric. Although $C$ has a much higher forwarding metric (7.0) and much higher chance of reaching the destination, the data will not be forwarded to $C$ since it is not the contact of $D$ (i.e., no line between them). However, since $C$ and $A$ are within the same TCC, it is better for them to exchange their forwarding metrics through multi-hop wireless communication, and then it will be possible for $C$ to forward the data to the destination.

In this chapter, we address the problem of data transmission in mobile opportunistic networks with diverse connectivity characteristics by proposing two TCC-aware data transmission strategies. Since a TCC is a MANET, it is treated as one component and data carriers in this TCC are selected for opportunistic data transmission. More specifically, this work has two contributions.

1. We identify the existence of TCCs and analyze their properties based on four traces. We find that there are significant number of TCCs in mobile opportunity networks, and the distributions of TCC size and node degree follow exponential distribution. By treating multi-hop wireless communications inside TCCs as indirect contacts, through theoretical analyses, we show that the contact opportunities can be significantly increased in all traces.

2. We first propose a TCC-aware data transmission strategy to exploit TCCs to improve the performance of data transmission in mobile opportunistic networks with diverse connectivity characteristics. In our solution, nodes
inside a TCC exchange their forwarding metrics through multi-hop wireless communications, and the node with the highest forwarding metric is selected to get a replicated data copy. Although the TCC-aware data transmission strategy can increase the data delivery ratio, it increases the data copies in the network. To address this problem, we enhance the TCC-aware data transmission by selecting an optimal set of nodes in the TCC to avoid overlap in their contacts and maximize the data transmission opportunity with a small number of nodes. Trace-driven simulations show that our TCC-aware data transmission strategies outperform existing data transmission strategies with less network overhead.

The rest of the chapter is organized as follows. In Section 3.2, we identify the properties of TCCs based on four traces. Section 3.3 presents our TCC-aware data transmission strategies. In Section 3.4, we evaluate the performance of the TCC-aware data transmission strategies. Section 3.5 reviews related work, and Section 3.6 concludes the chapter.

### 3.2 Trace-based TCC analysis

In this section, we identify the existence of TCCs and analyze their properties based on four realistic traces about mobile opportunistic networks.

#### 3.2.1 Traces

We study the properties of TCCs based on four traces: *Social Evolution* [52], *Friends & Family* [53], *Reality Mining* [37], and *UCSD* [38]. These traces record contacts among users carrying different kinds of mobile devices. The first three traces are collected by the MIT Reality group based on Bluetooth on smartphones. Among them, *Social Evolution* records the contacts among students in an undergraduate dormitory. *Friends & Family* records the contacts among members of a young-family residential community. *Reality Mining* tracks the contacts between individuals in research labs. In these three traces, mobile devices periodically detect their peers via Bluetooth interfaces. A contact is recorded when two mobile devices move into the detection range of each other. The *UCSD* trace is collected
at a campus scale, where the devices are WiFi enabled PDAs. These devices search for nearby WiFi Access Points (APs), and a contact is detected when two devices detect the same AP. The details of these four traces are shown in Table 3.1, with **SE** representing *Social Evolution*, **FF** representing *Friends & Family*, and **RM** representing *Reality Mining*.

**Table 3.1. Trace Summary**

<table>
<thead>
<tr>
<th>Trace</th>
<th>SE</th>
<th>FF</th>
<th>RM</th>
<th>UCSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Cell Phone</td>
<td>Cell Phone</td>
<td>Cell Phone</td>
<td>WiFi</td>
</tr>
<tr>
<td>Network type</td>
<td>Bluetooth</td>
<td>Bluetooth</td>
<td>Bluetooth</td>
<td>WiFi</td>
</tr>
<tr>
<td>Number of devices</td>
<td>80</td>
<td>84</td>
<td>97</td>
<td>275</td>
</tr>
<tr>
<td>Number of contacts</td>
<td>200,841</td>
<td>47,774</td>
<td>114,046</td>
<td>201,923</td>
</tr>
<tr>
<td>Durations(days)</td>
<td>243</td>
<td>142</td>
<td>246</td>
<td>78</td>
</tr>
<tr>
<td>Granularity(secs)</td>
<td>300</td>
<td>300</td>
<td>300</td>
<td>20</td>
</tr>
</tbody>
</table>

### 3.2.2 Properties of the Contact Graph

Based on the collected traces, we can draw contact graphs which consist of mobile devices (nodes) and their contacts (edges), with which we study the transient contact status between nodes. A contact graph can be drawn at each time point, i.e., there is an edge between two nodes if they are contacting each other at that time point. Here, a contact graph is drawn every 10 minutes in the traces.

The extracted contact graphs have some interesting properties, and one of them is related to the distribution of the node degree. The Complementary Cumulative Distribution Function (CCDF) of the node degree $P(K > k)$, which represents the probability that a node has more than $k$ contacts with other nodes, follows exponential distribution for $k \geq 1$:

$$P(K > k) = e^{-\frac{k}{k^*}} \quad k \geq 1$$

where $k^*$ is the exponential constant. By recognizing that $e^{-\frac{k}{k^*}} = 1$ when $k = 0$, we have the following:

$$P(K > k) = e^{-\frac{k}{k^*}} \quad k \geq 0 \quad (3.1)$$

The plot of CCDF $P(K > k)$ is shown in Figure 3.2. The CCDF curve is drawn in
a semi-log plot (log scale on y axis), where the exponential distribution is a straight line. The Least Square approach [54] is applied to fit the degree distribution with the exponential distribution. As we can see in Figure 3.2, the degree distributions can be well approximated by exponential distributions in most traces.

Since the contact graphs satisfy exponential degree distributions, most nodes have low degrees. This is because the contact graph is partitioned into connected components, where nodes are only connected to some nodes in the same connected component, so that most nodes have low degrees. For example, in the trace of Friends & Family, people in the residence community form small connected components when they stay with their families. In the trace of Reality Mining, students form connected components when they are in the same laboratory. Since the connected components only appear temporarily, they are referred to as Transient Connected Components (TCCs).
3.2.3 Properties of TCC

We study the properties of TCCs in this subsection. The first property is related to Giant Connected Component (GCC), which is the largest connected component of the contact graph. The GCC is commonly used to represent the overall connectivity of the network [55], and the property of the GCC is closely related to the degree distribution [56][57][58]. Based on [58], in a random graph with exponential degree distribution, if the exponential constant $k^*$ is larger than 1, there exists a GCC with size that scales linearly with the size of graph. With the increase of $k^*$, the size of the GCC also increases. Since $k^*$ in all traces is larger than 1 as shown in Figure 3.2, we can expect that there are large GCCs inside these contact graphs.

Figure 3.3 uses boxplot to show the size of the largest TCC as a proportion of the network size, where the largest TCC is detected every 10 minutes. Here, the network size is the number of nodes that are in contact with some nodes in the trace. Other nodes may turn off their devices or contact nodes that are not included in the trace, and these nodes are not included in our analysis. The results in Figure 3.3 are consistent with our conclusion that a larger exponential constant $k^*$ leads to a larger GCC. For example, the Family & Friends trace has the smallest GCC, because $k^*$ is only slightly larger than 1.
Figure 3.4. The distribution of TCC size can be well approximated by exponential distribution for most of the traces. $s^*$ is the exponential constant.

Even though the size of GCC implies the overall connectivity of the network, it is not enough to characterize the complete TCC structure. Therefore, we are also interested in the distribution of the TCC size. Figure 3.4 plots the distribution of the TCC size. Similar to the degree distribution, the TCC size distribution also follows exponential distribution. The CCDF of the TCC size $P(S > s)$ is exponential for $s \geq 2$:

$$P(S > s) = \begin{cases} e^{-s^*} & s \geq 2 \\ 1 & 0 \leq s \leq 1 \end{cases}$$  \hspace{1cm} (3.2)$$

where $s^*$ is the exponential constant.
3.2.4 Increasing Contacts with TCC

Inside a TCC, nodes can communicate with each other through multi-hop wireless communication, which is much faster than “carry-and-forward”. Therefore, as long as two nodes are within a TCC, they have TCC-contact, which can be direct contact or indirect contact. Direct contacts are contacts between two nodes within one-hop communication distance, and indirect contacts are contacts between nodes within the same TCC but multi-hop away. Next, we analyze whether and how contact opportunities are increased by considering TCC-contacts. We find that the increase of contact opportunities is related to the distributions of node degree and TCC size. Specifically, we have the following theorem:

**Theorem 1.** For a contact graph, the contact opportunities can be increased by considering TCC-contacts, if the following two conditions are satisfied:

1. The distribution of node degree and the distribution of TCC size both follow exponential distributions with exponential constants $k^*$ and $s^*$ respectively. (Equations (3.1) and (3.2))

2. $k^* < \hat{k}^*$, where $\hat{k}^*$ is a function of $s^*$:

$$\hat{k}^* = \frac{1}{\log \frac{2(e^{-\frac{1}{s^*}}+(1-e^{-\frac{1}{s^*}})^3)}{3e^{-\frac{1}{s^*}}-1+(1-e^{-\frac{1}{s^*}})^3}}$$ (3.3)

The proof of Theorem 1 can be found in Appendix. In the proof, we also compute the ratio of TCC-contacts to direct contacts, which can be determined by two exponential constants $k^*$ and $s^*$:

$$\frac{m_t}{m_d} = 2(1-e^{-\frac{1}{s^*}}) \cdot \frac{e^{-\frac{1}{s^*}}+(1-e^{-\frac{1}{s^*}})^3}{1-e^{-\frac{1}{s^*}}+(1-e^{-\frac{1}{s^*}})^3}$$ (3.4)

where $m_t$ is the number of TCC-contacts and $m_d$ is the number of direct contacts. With Equation (3.4), we can estimate the increased contact opportunities as long as the node degree and TCC size follow exponential distributions.

Table 3.2 lists $k^*$, $s^*$ and the value of $\hat{k}^*$ in terms of $s^*$ in four traces. We also did experiments to verify if the estimated value $\frac{m_t}{m_d}$ in Equation (3.4) (i.e., Est.
is consistent with the actual value (i.e., Act. $\frac{m_t}{m_d}$) and the results are shown in Table 3.2. As can be seen, $k^*$ is smaller than $\hat{k}^*$ in all traces, which indicates that the contact opportunities are increased in all these traces. We also find that the error of the estimated $\frac{m_t}{m_d}$ compared to the actual value is very small, which proves the accuracy of our estimation. Thus, with the distributions of node degree and TCC size, we can accurately estimate the increase of contact opportunities.

Table 3.2. TCC-contacts & Direct Contacts

<table>
<thead>
<tr>
<th>Trace</th>
<th>$k^*$</th>
<th>$s^*$</th>
<th>$\hat{k}^*$</th>
<th>Est. $\frac{m_t}{m_d}$</th>
<th>Act. $\frac{m_t}{m_d}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Evolution</td>
<td>1.70</td>
<td>5.25</td>
<td>8.80</td>
<td>4.14</td>
<td>4.33</td>
</tr>
<tr>
<td>Friends &amp; Family</td>
<td>1.14</td>
<td>1.52</td>
<td>1.57</td>
<td>1.24</td>
<td>1.38</td>
</tr>
<tr>
<td>Reality Mining</td>
<td>1.55</td>
<td>3.13</td>
<td>4.56</td>
<td>2.42</td>
<td>2.00</td>
</tr>
<tr>
<td>UCSD</td>
<td>2.36</td>
<td>4.09</td>
<td>6.47</td>
<td>2.41</td>
<td>2.31</td>
</tr>
</tbody>
</table>

3.2.5 Durations of TCC-contacts

Because TCCs are formed when nodes have direct contacts with each other, the duration of the TCC-contact is smaller than the duration of direct contact. The comparisons between the durations of TCC-contacts and direct contacts in four traces are shown in Figure 3.5. As shown in the figure, even though TCC-contacts have smaller durations than direct contacts, the median durations for TCC-contacts are still several minutes. Therefore, by considering both direct and indirect contacts inside TCCs, contact opportunities are increased and the contact durations are still pretty long. This property can be exploited to design more efficient data transmission strategies, as shown in the next section.

3.3 TCC-aware Data Transmission Strategies

Data transmission in mobile opportunistic networks is difficult due to the opportunistic nature of the network. Being aware of the existence of TCCs, we propose better data transmission strategies by utilizing the contact opportunities in TCCs. Inside a TCC, nodes can reach each other by multi-hop wireless communications, which will significantly increase the contact opportunities and increase
the chance of transmitting data to the destination. In this section, we first present our TCC-aware data transmission strategy and then improve its performance with an enhanced version.

### 3.3.1 TCC-Aware Data Transmission Strategy

The objective of the TCC-aware data transmission strategy is to utilize TCCs to improve the performance of data transmission.

#### 3.3.1.1 Identifying the TCC

To utilize TCCs, the first step is to identify the TCCs. A node can detect the nodes in the current TCC by broadcasting inside the TCC, and each node receiving the broadcast sends an acknowledgement. In order to know what data items are being transmitted in the current TCC, the node receiving the broadcast message replies.

---

**Figure 3.5.** The durations of direct contacts and TCC-contacts.
with an acknowledgement that carries the information about the data items in its buffer. By collecting acknowledgements from other nodes in the TCC, the original sender can identify the nodes and the transmitted data in its current TCC.

### 3.3.1.2 Transmission Strategy

For each data item created in the network, we intend to transmit it from the source to the destination within the time constraint $T$. In the process of transmission, data can be replicated and transmitted to other nodes in accordance with specific strategies. The transmission is successful as long as one data copy of this data item arrives at the destination.

The traditional data transmission strategies for mobile opportunistic networks are normally based on social-aware forwarding metrics such as centrality [10][14][17], which quantify mobile node’s capability of contacting others in the network. When a node carrying data contacts another node, the data packet is transmitted based on Compare-and-Forward [10][9]; i.e., the data carrier forwards the data if and only if the contacted node has a higher centrality and does not have this data. The original data carrier also keeps a copy after forwarding the data.

In our TCC-aware data transmission strategy, a similar method is used. However, the data transmission decisions are not limited to these two contacted nodes, but among all nodes in the TCC. A TCC can be formed by merging two existing TCCs or by adding new nodes to existing TCCs. For example, nodes $A$ and $B$ are originally from two different TCCs before they contact. After $A$ contacts $B$, the two TCCs are merged to a new TCC.

When two nodes contact, they first check if they are already in the same TCC before the contact. They detect their original TCCs by using broadcast messages described in Section 3.3.1.1. As long as one of them receives acknowledgement from the other one, they are already within the same TCC. If the detected TCCs are different for the two nodes, these two TCCs are merged when they contact.

After a new TCC is formed, a simple compare-and-forward strategy is used to transmit data to nodes with higher centrality. However, this strategy creates many data copies. To reduce redundancy, data is only transmitted to the node with the highest centrality, and then at most one additional data copy is created in a TCC.
Algorithm 4 TCC-aware Data Transmission Strategy when two Nodes $A, B$ Contact

1: $M_A \leftarrow$ the set of nodes in $A$’s original TCC
2: $M_B \leftarrow$ the set of nodes in $B$’s original TCC
3: if $M_A = M_B$ then
4:   Do NOTHING /*$A, B$ are already in the same TCC*/
5: else
6:   /*Transmission decision inside the TCC will be made centrally at a command node $C$, which is chosen from $A, B$.*/
7:   if $|M_A| \leq |M_B|$ then
8:      $C \leftarrow B$
9:   else
10:      $C \leftarrow A$
11: end if
12:   /*$C$ will identify the nodes inside the new TCC and the data items carried by them, and make the transmission decision.*/
13: $M \leftarrow M_A \cup M_B$
14: $D \leftarrow$ the set of unique data items in the new TCC
15: for each data item $d \in D$ do
16:   if $M$ includes $d$’s destination then
17:      $d$ is transmitted to $d$’s destination
18:      Go to next data item
19: end if
20: $H_d \leftarrow$ node with the highest centrality
21: if $H_d$ has the data then
22:   Do NOTHING and go to the next data item
23: else
24:   $d$ is transmitted to $H_d$
25: end if
26: end for
27: end if

3.3.1.3 The Algorithm

The whole process of the TCC-aware data transmission strategy is outlined in Algorithm 4. When two nodes contact, they first check if they belong to the same TCC (Lines 3 ~ 5). If so, the data packet has already been transmitted in the TCC. If not, a new TCC is formed, and the data items will be transmitted inside the new TCC.

The transmission decision inside this TCC is made centrally at a node, denoted as “command” node, which is chosen from the two nodes in contact. The node that has more nodes in its original TCC will be the command node (Lines 6 ~ 10).
Then, the command node checks all the nodes and their carried data items in the new TCC and makes the transmission decisions (Lines 11 ∼ 24). Specifically, for each data item that exists in the TCC, if its destination is inside this TCC, it is directly transmitted to the destination (Lines 14 ∼ 17). Otherwise, the command node checks if the node with the highest centrality has the data. If the node has the data, nothing needs to be done. Otherwise, one data copy is created and transmitted to the node with the highest centrality (Lines 18 ∼ 23).

### 3.3.2 Enhanced TCC-aware Data Transmission Strategy

#### 3.3.2.1 Motivation

In the TCC-aware data transmission strategy, the node with the highest centrality in the TCC gets one data copy and the original data carriers also keep their data copies. However, this may not be the best option in many cases. For example, Figure 3.6 (a) shows a TCC of three nodes and Figure 3.6 (b) lists each node’s contact probability with others in the network. If the centrality metric is based on the Cumulative Contacting Probability (CCP) [17], the CCP of node A is $1 + 1 + 0.7 + 0.6 + 0.3 + 0.2 = 3.8$, the CCP of node B is $1 + 1 + 0.1 + 0.1 + 0.6 + 0.5 = 3.3$, and the CCP of node C is $1 + 1 + 0.5 + 0.6 + 0.1 + 0.1 = 3.3$. If C originally carries the data, A will receive one data copy since it has the highest centrality. However, nodes A and C have similar contact probabilities to other nodes B, D, E, F and G; i.e., they both have high contact probabilities with B, D and E, and low contact probabilities with F and G. Thus, keeping data copies at A and C may not help
too much since their contact capabilities to other nodes have a large “overlap”.

To deal with this problem, it is better to choose A and B as data carriers, because B has high contact probabilities to F and G, which is complement to A. Even though B and C have the same CCP centrality, using A and B as data carriers is better than using A and C. Thus, we propose an enhanced TCC-aware data transmission strategy, where data carriers are selected to maximize the data transmission opportunity. Different from the previous TCC-aware data transmission strategy, which selects the highest centrality node as the data carrier, we select a set of nodes as data carriers which can maximize the data transmission opportunity.

### 3.3.2.2 Set Centrality

We first define the concept of set centrality to quantify the data transmission capability of a set of nodes. Given that a data item is created at time 0, and expires at time T, the set centrality of a node set S at time t is defined as follows:

**Definition 1.** The set centrality of a node set S at time \( t < T \) is defined as the summation of their overall probability to contact each of the remaining nodes in \( N \setminus S \) before time T.

\[
C_S(t) = \sum_{i \in N \setminus S} \left( 1 - \prod_{j \in S} (1 - p_{ji}(T - t)) \right)
\]  

(3.5)

where \( N \) is the set of nodes in the network, and \( p_{ji}(T - t) \) is the probability that node j will contact node i within time \( T - t \).

Assume symmetric contacts between nodes i and j, and then we have \( p_{ij}(T - t) = p_{ji}(T - t) \). Since the inter-contact time between node i and node j has been experimentally validated in [17] to follow an exponential distribution with rate parameter \( \lambda_{ij} \), the probability that the two nodes will contact before T can be calculated as:

\[
p_{ij}(T - t) = p_{ji}(T - t) = 1 - e^{-\lambda_{ij}(T-t)}
\]

(3.6)
3.3.2.3 Selecting the Optimal Set of Data Carriers

We assume that there are \( k \) data copies carried by nodes in a TCC (denoted as \( \mathcal{M} \)), where \( k < |\mathcal{M}| \). Our objective is to choose an optimal node set \( S^* \) of size \( k \) with the highest set centrality, where \( S^* \subset \mathcal{M} \). Nodes in the optimal node set \( S^* \) will be the new data carriers for the \( k \) data copies. The detection of the optimal node set \( S^* \) can be formalized as an optimization problem:

\[
\max \sum_{i \in \mathcal{N}} (1 - x_i)(1 - \prod_{j \in \mathcal{N}} (1 - x_j \cdot p_{ji}(T - t))) \quad (3.7)
\]
\[
s.t. \quad x_i \in \{0, 1\}, \quad \forall i \in \mathcal{M} \quad (3.8)
\]
\[
x_i = 0, \quad \forall i \in \mathcal{N} \setminus \mathcal{M} \quad (3.9)
\]
\[
\sum_{i \in \mathcal{M}} x_i = k \quad (3.10)
\]

where \( x_i \in \{0, 1\} \) indicates whether the node \( i \) is selected to the optimal node set \( S^* \). Formula (3.7) maximizes the set centrality of the selected nodes. Since \( S^* \) is selected from \( \mathcal{M} \), \( x_i \in \{0, 1\} \) for nodes inside \( \mathcal{M} \), and \( x_i = 0 \) for nodes outside of \( \mathcal{M} \) as depicted in (3.8) and (3.9) respectively. Formula (3.10) indicates that the number of selected nodes is \( k \).

The optimization problem can be easily solved using dynamic programming similar to the knapsack problem [59]. With dynamic programming, the optimization problem can be solved with time complexity \( O(|\mathcal{M}|) \).

3.3.2.4 Transmission Strategy

The enhanced TCC-aware data transmission strategy is presented in Algorithm 5. Based on the original TCC-aware data transmission strategy, the enhanced TCC-aware data transmission strategy adds an extra step that redistributes data copies inside a TCC (Lines 18 \sim 24). Similar to the original strategy, one extra data copy is created if the node with the highest centrality does not have the data (Lines 18 \sim 22). Then, all data copies are redistributed in the TCC, to make sure they are carried by a set of nodes with the optimal transmission capability. Specifically, the command node first computes an optimal node set with the highest set centrality using the method in Section 3.3.2.3 (Lines 23). Afterwards, the data copies are
Algorithm 5 Enhanced TCC-aware Data Transmission Strategy when two Nodes A, B Contact

1: $\mathcal{M}_A \leftarrow$ the set of nodes in A’s original TCC
2: $\mathcal{M}_B \leftarrow$ the set of nodes in B’s original TCC
3: if $\mathcal{M}_A = \mathcal{M}_B$ then
4:  Do NOTHING /*A, B are already in the same TCC*/
5: else
6: /* Transmission decision inside the TCC will be made centrally at a command node C, which is chosen from A, B.*/
7: if $|\mathcal{M}_A| \leq |\mathcal{M}_B|$ then
8:  $C \leftarrow B$
9: else
10:  $C \leftarrow A$
11: end if
12: /*C will identify the nodes inside the new TCC and the data items carried by them, and make the transmission decision.*/
13: $\mathcal{M} \leftarrow \mathcal{M}_A \cup \mathcal{M}_B$
14: $\mathcal{D} \leftarrow$ the set of unique data items in the new TCC
15: for each data item $d \in \mathcal{D}$ do
16: if $\mathcal{M}$ includes d’s destination then
17:  $d$ is transmitted to d’s destination
18:  Go to next data item
19: end if
20: $k_d \leftarrow$ number of copies of $d$ in $\mathcal{M}$
21: $H_d \leftarrow$ node with the highest centrality
22: if $H_d$ does not have $d$ then
23:  $k_d \leftarrow k_d + 1$ /*Add one extra data copy*/
24: end if
25: /*Decide $S^*$ as discussed in Section 3.3.2.3.*/
26: /*Nodes in $S^*$ will be new carries of data item $d$*/
27: end for
28: end if
29: Data are transmitted from old carriers to nodes in $S^*$
30: end if

transmitted from the original data carries to the nodes in the optimal node set (Lines 24).

3.4 Performance Evaluations

In this section, we evaluate the performance of the TCC-aware data transmission strategies by comparing them with existing data transmission strategies.
3.4.1 Strategies in Comparison

In TCC-aware data transmission strategies (TCC and Enhanced TCC), the centrality metric is based on the Cumulative Contacting Probability (CCP) [17]. We compare it with the following four transmission strategies:

- **Compare-and-Forward**: When a node carrying data contacts a node without data, the data carrier forwards the data if the contacted node has a higher forwarding metric (CCP). This strategy has been utilized in [10][9][60].

- **Epidemic** [43]: Upon contact, the data carrier forwards data to the contacted node if it does not have the data. This method has the best data delivery ratio but the highest network overhead, which is used as the upper bound.

- **Wait**: The data source does not transmit data to any other node until it contacts with the destination. This strategy has the worst delivery ratio but the lowest network overhead, which is used as the lower bound.

- **R3** [61]: R3 selects a fixed number of transmission paths which together achieve the minimum expected delay. For each path, one data copy is created and then transmitted along the path.

3.4.2 Simulation Setup

We evaluate all the data transmission strategies based on the four traces listed in Table 2.1. For each trace, the first half is used for warm up, based on which we determine the centrality metrics, and the second half is used to evaluate the performance.

For each data item, the source and destination are picked randomly, and the data generation time is randomly chosen in the daytime, since node’s activity remains low at night and high at daytime. The experiment is repeated 1000 times for statistical convergence.
Figure 3.7. Comparisons based on data delivery ratio under different time constraints.

### 3.4.3 Simulation Results

Figure 3.7 and Figure 3.8 compare TCC-aware data transmission strategies (TCC and Enhanced TCC) and other existing strategies. Figure 3.7 shows data delivery ratio under different time constraints, where delivery ratio is the proportion of data items successfully delivered to the destination before data expire. Figure 3.8 shows the network overhead under different time constraints, where the overhead is measured by the average number of data copies created in the network for each data item. Generally speaking, TCC-aware data transmission strategies have the best performance by achieving comparable delivery ratio with Epidemic with much lower network overhead. In the following, we discuss the simulation results in detail by first comparing our TCC-aware data transmission strategies with Compare-and-Forward and R3, and then comparing these two TCC-aware data transmission strategies.
Figure 3.8. Comparisons based on the number of data copies (overhead) created under different time constraints.

### 3.4.3.1 Comparisons with Compare-and-Forward

As shown in Figure 3.7 and Figure 3.8, both TCC and Enhanced TCC have better performance than Compare-and-Forward, with higher data delivery ratio and lower network overhead. Specifically, the TCC-aware data transmission strategies achieve 10% – 40% higher delivery ratio than Compare-and-Forward in all four traces. This is because by utilizing the contact opportunities inside TCCs, nodes have higher probabilities to transmit data to the destination. Moreover, the TCC-aware data transmission strategies consume 15% – 50% less network overhead than Compare-and-Forward. The decrease in network overhead is because there is at most one data copy created when a new TCC is formed. However, in Compare-and-Forward, data copies can be created upon every contact. These results demonstrate the effectiveness of TCC-aware data transmission strategies.
when compared with strategies designed for opportunistic networks.

### 3.4.3.2 Comparisons with R3

To compare with R3 [61], we set the number of data copies (paths) to be 5 (R3-5copies) in R3. As shown in Figure 3.7, the data delivery ratio for R3 with 5 copies is about 50% less than that of TCC and Enhanced TCC. We also test R3 with 10 copies and 20 copies, and find that the delivery ratio does not increase much as the number of data copies increases. This observation is consistent with the results in [61], which also found that the performance does not increase much as the number of data copies is larger. The reason for R3’s low delivery ratio is that R3 is based on source routing, and source routing is extremely time-consuming when utilized in networks with opportunistic feature.

### 3.4.3.3 Comparisons between the TCC-aware data transmission strategies

From Figure 3.7 and Figure 3.8, we can also see that Enhanced TCC consumes less overhead than TCC but achieves similar delivery ratio. This is because, in the enhanced strategy, data copies can be transmitted from the original data carriers to a set of nodes with the highest set centrality; i.e., more effective nodes are selected as data carriers. By choosing a small number of effective nodes to carry data, the enhanced strategy requires less data carriers than the original strategy. Therefore, the number of data copies can be decreased with Enhanced TCC.

### 3.5 Related Work

Early data transmission algorithms designed for mobile opportunistic networks are mostly based on Epidemic [43], where data is flooded upon contacts with other nodes. Later solutions attempt to reduce the number of data copies created by Epidemic, and these strategies are known as controlled flooding [62]. For example, Compare-and-Forward is commonly used to control the data copies created, and the data carrier only forwards data to another node with a higher forwarding metric. The forwarding metric measures node’s capability of forwarding data to the desti-
nation. In some research, the forwarding metrics are determined based on node’s contact probability with the destination, such as PROPHET [9] and MaxProp [63]. By further aggregating the networks to social graphs [64], social properties of mobile nodes are analyzed. Based on this, node centrality [14][60][19] or community based solutions[15][16][65][4] are used as forwarding metrics. However, these strategies are based on the assumption that nodes are sparsely connected in mobile opportunistic networks, and they are not the best solutions considering the diverse connectivity characteristics, since they neglect the multi-hop communication opportunities inside TCCs.

Phe-Neau et al. considered the multi-hop communication opportunities around a node’s vicinity in [66]. However, in their approach, multi-hop communication opportunities around a node are only utilized when the destination of the data is within the node’s vicinity, which is basically a WAIT strategy and then it does not contain mechanisms to make the data reach more nodes and get closer to the destination.

The work by Tie et al. [61] considered data transmission in networks with diverse connectivity characteristics; however, their solution is different from ours and they do not consider the effects of TCCs. They identified packet replication to be the key difference between protocols designed for well-connected networks and sparsely-connected networks, and designed a routing protocol called R3, which determines the number of data copies to be created based on the predicted delays along network paths. R3 is based on source routing; i.e., data copies are transmitted along the pre-determined transmission path. However, protocols based on source routing are not suitable for networks with opportunistic features, because it is extremely time-consuming to transmit data along the pre-determined paths. Moreover, R3 does not consider the effects of TCCs on data transmission, which is the key contribution of our work.

Other existing algorithms try to modify well-known MANET protocols to make them more adaptive to networks with diverse connectivity characteristics. For example, Raffelsberger et al. [67] integrated store-and-forward to MANET protocols. In MANET protocols, a data item is dropped when the routing table does not contain an entry for the destination. With store-and-forward, data is buffered until a route to the destination can be found using MANET protocols. However, their
solution lacks a mechanism to choose effective data carriers to deliver data to destination.

There exists some work on analyzing TCCs. For example, [51] demonstrated the size of the giant connected components changes over time. [56][57][58] proved that the property of connected component is closely related with the distribution of node degree. However, they did not examine the detailed structure of TCCs using real traces, and did not consider how to use them to increase the contact opportunities and improve the performance of data transmission, which is the focus of our work.

3.6 Conclusions

In this chapter, we designed efficient data transmission strategies for mobile opportunistic networks with diverse connectivity characteristics, by exploiting the existence of TCCs. We first identified the existence of TCCs and analyzed their properties based on four traces. By treating multi-hop wireless communications inside TCCs as indirect contacts, through theoretical analyses, we showed that the contact opportunities can be significantly increased in all traces. Based on this observation, we designed a TCC-aware data transmission strategy to improve the performance of data transmission. Then, we enhanced the TCC-aware data transmission by selecting an optimal set of nodes in the TCC to avoid overlap in their contacts and maximize the data transmission opportunity with a small number of nodes. Trace-driven simulations showed that our TCC-aware data transmission strategies outperform existing data transmission strategies with much higher delivery ratio and less network overhead.
4.1 Introduction

In this chapter, we consider data are collected based on crowdsourcing. Specifically, when a user is requiring some data, a crowdsourcing task is created and multiple sources (i.e., mobile users) are queried for this task. Then the sources collect data and transmit the data back to the destination (i.e., the user who is requiring data) through the mobile opportunistic network.

Although crowdsourcing can provide a large amount of information to mobile users, the reliability of these information sources is usually unknown a priori and the information provided by them may be inaccurate. An important problem in crowdsourcing is to assess the reliability of the mobile users and identify the truth from the reported data to improve data quality. To address this problem, various truth analysis techniques have been proposed by researchers. For example, [21][22][23] designed techniques based on a basic heuristic, i.e., iteratively estimating the truth of the information by analyzing the reliability of the information source and estimating the reliability of information sources based on the correctness of their provided information. Some other researchers also investigated statistical models for truth finding, such as bayesian inference [24][26] and expectation-maximization (EM) [27].
The aforementioned truth analysis techniques only depend on the currently available or observed data. However, if the available data are limited or have large conflicts, it is difficult to identify the truth or ensure the data credibility (quality). In this chapter, we address this problem by utilizing the communication networks (i.e., mobile opportunistic networks) to adaptively collect data from mobile users, especially when the existing data are not enough to ensure data credibility. Specifically, based on a feedback loop between information and communication networks, we are able to utilize the communication networks to collect the right information from mobile users and make sure enough data are collected to improve the data credibility.

Although data credibility can be improved by utilizing communication networks to collect extra data, it also increases the communication overhead. The communication overhead for data collection can be significant in mobile opportunistic networks. This is because the network resources are limited due to the dynamics of network topology and the lack of end-to-end routing path in mobile opportunistic networks.

Considering the requirement on data credibility and the constraint of network resources, we quantify the tradeoff between the enhanced data credibility and the increased network overhead, and propose resource-aware approaches for truth analysis. Specifically, we formalize two problems in resource-constrained mobile opportunistic networks: the max-credibility problem which aims to maximize data credibility with some network overhead, and the min-overhead problem which aims to achieve a specified data credibility while minimizing the network overhead. Overall, the contributions of this work are summarized as follows:

- Different from existing truth analysis approaches, we utilize the communication network to adaptively collect data from some users, so that high-quality data are collected to achieve the required data credibility.

- We formalize the max-credibility problem and propose a solution to select users based on their data collection capabilities such as network reachability and user reliability. A maximum likelihood estimation (MLE)-based technique is also proposed to identify truth from the collected data.

- We formalize the min-overhead problem and propose a solution which iter-
atively selects users for data collection until the required data credibility is achieved.

- The proposed solutions are evaluated based on trace-driven simulations and a testbed consisting of 20 Bluetooth-enabled smartphones. Both simulation and experimental results demonstrate the effectiveness of the proposed solutions in terms of data credibility and network overhead.

The rest of the chapter is organized as follows. In Section 4.2, we present the problem formalization and the network model. Section 4.3 presents the max-credibility problem and our proposed solutions. Section 4.4 presents the min-overhead problem and our proposed solutions. In Section 4.5, we evaluate the performance of the proposed solutions. Section 4.6 reviews related work, and Section 4.7 concludes the chapter.

4.2 Preliminaries

In this section, we first formulate the problems addressed in the chapter, and then describe the network model and user reliability.

4.2.1 Problem Formulation

The system model includes a server (i.e. the user who requests data) and a set of mobile users who communicate with each other through a mobile opportunistic network, as shown in Figure 4.1. Similar to [68], we assume the server knows the mobile users existing in the network. In this work, we focus on the sensing tasks requiring real-valued measurements. First, the server creates sensing tasks specifying the data required and the time constraint, and sends each sensing task to a set of selected mobile users through the mobile opportunistic network. Then, the queried users collect data as specified by the sensing task if they have the sensing capability, and send the collected data back to the server. Note that the time constraint is at a much larger time scale, and it does not mean real time. It is used to reduce the unnecessary traffic due to the large delay variation in mobile opportunistic networks.
Since the reliability of mobile users is usually unknown a priori, the data provided by them may be inaccurate and the data credibility cannot be ensured. To ensure the data credibility, the server usually queries multiple users to collect data, and then estimates the true value based on truth analysis techniques. However, in mobile opportunistic networks, querying more nodes also induces more network overhead. This problem becomes worse considering that mobile users may have to send back extra information to improve the data provenance; e.g., voice or video related to the collected data or the context (temporal and spacial information) related to this data collection. This will significantly increase the data size and hence increase the network overhead. Considering the requirements on data credibility and the constraint of network resources in mobile opportunistic networks, our main objective is to quantify the tradeoff between the improved data credibility and the increased network overhead, and propose resource-aware approaches to improve data credibility.

To formulate the problem, we first explain how to quantify data credibility and network overhead.

**Definition 2.** Data credibility measures the quality of the data for a sensing task. We use the estimation accuracy to represent the data credibility. A smaller estimation error $|\mu - \hat{\mu}|$ stands for a higher data credibility, where $\mu$ is the ground truth and $\hat{\mu}$ is the estimated value from the collected data.

In mobile opportunistic networks, due to the sparse node connectivity, the performance of data collection is limited by the transmission opportunities which
only happen when nodes move into the communication distance. A large data item may have to be cut into small packets to make sure that they can be transmitted during a short contact time. Therefore, we use the total number of transmissions (i.e., transmission hops) for all these data packets to quantify the network overhead.

**Definition 3.** The **network overhead** of a sensing task is quantified as the total number of transmission hops used for collecting data from mobile users. If the number of transmission hops for collecting data from user $i$ is $h_i$, the network overhead is represented as $\sum_{i \in V_s} h_i$, where $V_s$ is the set of queried users.

In this definition, the transmission hops for collecting data from a user include the round-trip transmission hops for transmitting the task to the user and sending back the collected data to the server.

To quantify the tradeoff between data credibility and network overhead, we formulate two problems: the **max-credibility** problem and the **min-overhead** problem.

**Problem 1. Max-credibility:** finding the set of mobile users to query so that the data credibility is maximized while the network overhead is below a specific value.

The max-credibility problem can be formalized as follows:

\[
\begin{align*}
\text{min} & \quad |\mu - \hat{\mu}| \\
\text{s.t.} & \quad \sum_{i \in V} s_i \cdot h_i \leq \beta \\
& \quad s_i \in \{0, 1\}, \quad \forall i \in V
\end{align*}
\]

where $s_i \in \{0, 1\}$ denotes whether $i$ is selected for collecting data, $V$ denotes the set of mobile users in the system, and $\beta$ denotes a specific value for network overhead. Maximizing data credibility equals to minimizing estimation error as depicted in Formula (4.1). Formula (4.2) ensures that the network overhead is below $\beta$.

**Problem 2. Min-overhead:** using minimum network overhead to collect data so that a specified data credibility is achieved.

The min-overhead problem can be formalized as follows:

\[
\begin{align*}
\text{min} & \quad \sum_{i \in V} s_i \cdot h_i
\end{align*}
\]
\[ s.t. \quad |\mu - \hat{\mu}| \leq \epsilon \quad (4.5) \]

\[ s_i \in \{0, 1\}, \quad \forall i \in V \quad (4.6) \]

The objective of the optimization is to minimize the network overhead (Formula (4.4)), while achieving the specified data credibility, i.e., limiting the estimation error to be smaller than \( \epsilon \) as shown in Formula (4.5).

Here, the collected data for sensing tasks are assumed to be numerical values. The magnitude of the data may vary tremendously for different sensing tasks. To ensure different sensing tasks can be processed using the same statistical model, the data value for a sensing task is normalized by a base number \( \sigma \), which is a constant and relevant to the data magnitude specific to each sensing task. It can be learned using MLE as presented in the next section.

With data normalization, the estimation error \( |\mu - \hat{\mu}| \) in the two problems should be modified to the normalized estimation error \( \frac{|\mu - \hat{\mu}|}{\sigma} \). This does not affect the objective (Formula (4.1)) in the max-credibility problem, since minimizing the estimation error is equal to minimizing the normalized estimation error. For the min-overhead problem, the constraint of limiting the estimation error by \( \epsilon \) (Formula (4.5)) becomes limiting the normalized error by \( \epsilon \): \( \frac{|\mu - \hat{\mu}|}{\sigma} \leq \epsilon \)

### 4.2.2 Models

#### 4.2.2.1 Network Model

The network studied here is a mobile opportunistic network \( G(V, E) \), where \( V \) is the set of nodes (i.e., mobile users) and \( E \) is the set of edges. The edge \( e_{u,v} \) in \( E \) represents the pairwise contact process between nodes \( u \) and \( v \) (\( u, v \in V \)). The inter-contact time between nodes \( u \) and \( v \) has been experimentally validated in [17] to follow an exponential distribution with rate parameter \( \lambda_{uv} \). The contact process between \( u \) and \( v \) follows a homogeneous Poisson process with rate \( \lambda_{uv} \).

#### 4.2.2.2 Reliability of Users

The mobile users in the system have different reliability. The reliability determines the quality of the data provided by the user. Here, we denote the reliability of a user \( i \) as \( r_i \) (\( 0 < r_i \leq 1 \)). A reliability of 1 means that the user is totally reliable and
likely to provide high-quality data. A user with lower reliability is likely to provide
data deviating more from the ground truth. Similar to [26], we assume that the
random observation of user \( i \) for task \( j \) follows normal distribution \( N(\mu_j, (\sigma_j/r_i)^2) \),
where \( \mu_j \) is the ground truth of task \( j \) and \( \sigma_j/r_i \) is the standard deviation \( \sigma_j 
\) is the base number of task \( j \). The assumption of normal distribution has been
experimentally validated in [69]. The distribution has a smaller variance as the
reliability \( r_i \) is higher. Even if \( r_i = 1 \), the random observation of user \( i \) still
has variance \( \sigma_j^2 \), which is caused by the unavoidable random errors in real-world
observations.

4.3 Max-Credibility with Network Resource Constraint

In this section, we first study the problem of maximizing data credibility with
limited network resource. The objective of the max-credibility problem is to min-
imize the estimation error as depicted in Formula (4.1). However, it is impossible
to directly calculate the estimation error since the ground truth of the task is
unavailable to server. Therefore, we propose a heuristic-based max-credibility ap-
proach to address the problem. In the approach, the server first selects users that
are more likely to provide accurate data with limited network overhead. After
data are collected from these users, an MLE-based technique is further utilized to
identify the truth from these data.

4.3.1 User Selection

To ensure the selected users can provide data that achieve high data credibility,
our max-credibility approach prioritizes the users that are more likely to provide
accurate data with limited network overhead. Specifically, users are selected based
on their data collection capabilities such as network reachability and user reliability.

4.3.1.1 Network reachability

The network reachability measures if the data can be successfully transmitted
within a time constraint. The transmission includes the round-trip transmission
between the server (denoted as s) and user i. To transmit data between two nodes in the network, we simply choose the path with the minimum expected delay. If the time constraint for task j is $\mathcal{T}_j$, we try to ensure the one-way transmission delay between the server and the user is smaller than $\mathcal{T}_j/2$. Based on the assumption that the pairwise contact processes follows Poisson processes which are mutually independent, the delay on the transmission path in a mobile opportunistic network satisfies hypo-exponential distribution [17]. The probability that the data can be successfully transmitted from the server to user i within $\mathcal{T}_j/2$ is:

$$P(D_{s\to i} < \mathcal{T}_j/2) = \sum_{i=1}^{l} C_i^{(l)} \cdot (1 - e^{-\lambda_i \mathcal{T}_j/2}) \quad (4.7)$$

where the coefficient $C_i^{(l)} = \prod_{k=1, k\neq i}^{l} \frac{\lambda_k}{\lambda_i}$. In this formula, l is the number of hops on the transmission path, and the rate parameter on each hop is $\lambda_i$ ($1 \leq i \leq l$). Since the transmission delay from user i back to server s has the same hypo-exponential distribution as the transmission delay from s to i, we have

$$P(D_{i\to s} < \mathcal{T}_j/2) = P(D_{s\to i} < \mathcal{T}_j/2) \quad (4.8)$$

Then, we simply compute the probability that the data can be successfully transmitted before $\mathcal{T}_j$ as:

$$p_{i,j}^{r} = P(D_{s\to i} < \mathcal{T}_j/2) \cdot P(D_{i\to s} < \mathcal{T}_j/2) \quad (4.9)$$

4.3.1.2 Reliability

If user i has the capability to collect data, the reliability of user i determines the accuracy of the data it collects. The collected data value is accurate if the normalized error from the ground truth is smaller than $\epsilon$, i.e., $|\frac{x_{ij} - \mu_j}{\sigma_j}| < \epsilon$, where $x_{ij}$ is the data collected by i. Based on the assumption that the random observation of user i has normal distribution $N(\mu_j, (\sigma_j/r_i)^2)$, where $\sigma_j$ is the base number for task j and $r_i$ is the reliability for user i, the probability that the collected data value is accurate is:

$$p_{i,j}^{r} = P(\frac{|x_{ij} - \mu_j|}{\sigma_j} < \epsilon)$$
\begin{align*}
\Phi(\epsilon r_i) - \Phi(-\epsilon r_i) & \quad (4.11) \\
\end{align*}

where \( \Phi(*) \) is the cumulative distribution function (CDF) of the standard normal distribution. Even though the unknown ground truth \( \mu_j \) and base number \( \sigma_j \) are used in the deviation, the result \( \Phi(\epsilon r_i) - \Phi(-\epsilon r_i) \) is not influenced by \( \mu_j \) and \( \sigma_j \).

Then, the server integrates the two properties and computes the overall probability for user \( i \) to provide accurate data before task expires, i.e., \( p_{i,j} = p_{i,j}^a \cdot p_{i,j}^r \).

The probability \( p_{i,j} \) is also referred to as the success probability for user \( i \).

In the beginning, the server usually does not know users’ reliability in priori. In this case, it can simply assume users are reliable and use \( r_i = 1 \) when computing \( p_{i,j}^r \). As more data have been collected from users, the server may have some knowledge about users’ reliability based on our MLE-based estimation. In this case, the inclusion of \( p_{i,j}^r \) in \( p_{i,j} \) can assist the selection of more reliable users.

Afterwards, the server selects the users to query based on the following optimization problem:

\begin{align*}
\max & \quad 1 - \prod_{i \in V} (1 - p_{i,j})^{s_i} \\
\text{s.t.} & \quad \sum_{i \in V} s_i \cdot h_i \leq \beta \\
& \quad s_i \in \{0, 1\}, \quad \forall i \in V
\end{align*}

where \( h_i \) denotes the network overhead for collecting data from user \( i \), and \( s_i \) denotes whether \( i \) is selected. The optimization problem maximizes the probability that at least one selected user can provide accurate data, with the constraint that the total network overhead is less than \( \beta \). By solving the optimization problem, the server can prioritize the nodes that not only have a higher probability to provide accurate data but also consume less network overhead. This optimization problem can be solved using dynamic programming similar to a knapsack problem within polynomial time.
4.3.2 Finding Truth using MLE

After data are collected from the selected users, the truth can be identified with an
MLE-based truth analysis technique. Specifically, the MLE-based truth analysis
can be used to learn the unknown parameters like ground truth and the base
number of each task, and the reliability of each user.

4.3.2.1 Background for MLE

The maximum-likelihood estimation (MLE) is commonly used to estimate the
unknown parameters of a statistical model. In general, given a set of observed data
and the underlying statistical model, MLE selects the set of values for the model
parameters that maximize the likelihood function, which is the probability that
the data are observed under the resulting distribution. Given the set of observed
data $X = \{x_1, x_2, ..., x_n\}$ and the set of parameters $\Theta$, the likelihood function is
represented as

$$L(\Theta; X) = f(x_1, x_2, ..., x_n|\Theta) = \prod_{i=1}^{n} f(x_i|\Theta)$$ (4.15)

To maximize the likelihood function, the log-likelihood function is used to simplify
the process of derivation. Then, the MLE for $\Theta$ is computed as

$$\hat{\Theta} = \text{argmax} \log L(\Theta; X) = \text{argmax} \sum_{i=1}^{n} \log f(x_i|\Theta)$$ (4.16)

4.3.2.2 Mathematical Formulation and Derivation

To utilize MLE in our problem, we first set up the statistical model. In our statisti-
cal model, the observed data $X$ is the set of data collected from the selected users.
Assuming the server has collected data for $m$ tasks, we have $X = \{X_1, X_2, ..., X_m\}$,
where $X_j$ is the set of data values observed for task $j$. The unknown parameters
in the model include the ground truth $\mu_j$, the base number $\sigma_j$ for each task $j$, and
the reliability $r_i$ of each user $i$, i.e., $\Theta = \{\mu_1, \mu_2, ..., \mu_m, \sigma_1, \sigma_2, ..., \sigma_m, r_1, r_2, ..., r_n\}$.

Here, the ground truth of each task is treated as unknown parameters in
the statistical model, so that we can estimate the true value by estimating the set
of unknown parameters $\Theta$ using MLE.
Let $d_{ij} = \{0, 1\}$ denote if user $i$ has provided data for task $j$, and $x_{ij}$ denote the data user $i$ has provided for task $j$. The probability density function (pdf) that $x_{ij}$ is observed is:

$$f(x_{ij}|\Theta) = \frac{1}{\sigma_j/r_i \sqrt{2\pi}} e^{-\frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2/r_i^2}}$$  \hspace{1cm} (4.17)

From Equation (4.17), the pdf that the data in $X_j$ are observed for task $j$ is calculated as:

$$f(X_j|\Theta) = \prod_{i=1}^{n} (f(x_{ij}|\Theta))^{d_{ij}} = \prod_{i=1}^{n} \left( \frac{1}{\sigma_j/r_i \sqrt{2\pi}} e^{-\frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2/r_i^2}} \right)^{d_{ij}} \hspace{1cm} (4.18)$$

Here, $x_{ij}$ is only valid as $d_{ij} = 1$, i.e., when user $i$ has observation $x_{ij}$ for task $j$. As $d_{ij} = 0$, user $i$ has no observation, so $x_{ij}$ is meaningless. We can simply let $x_{ij} = 0$ as $d_{ij} = 0$ to make (4.18) a legal equation.

Then, the pdf that $X = \{X_1, X_2, ... X_m\}$ is observed can be computed based on Equation (4.18):

$$f(X|\Theta) = \prod_{j=1}^{m} f(X_j|\Theta) = \prod_{j=1}^{m} \prod_{i=1}^{n} \left( \frac{1}{\sigma_j/r_i \sqrt{2\pi}} e^{-\frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2/r_i^2}} \right)^{d_{ij}} \hspace{1cm} (4.19)$$

From Equation (4.19), the log-likelihood function $\log L(\Theta; X)$ is given by:

$$\log L(\Theta; X) = \log f(X|\Theta) = \sum_{j=1}^{m} \sum_{i=1}^{n} d_{ij} \cdot \left[ \log \left( \frac{1}{\sigma_j/r_i \sqrt{2\pi}} \right) - \frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2/r_i^2} \right] \hspace{1cm} (4.20)$$

To get $\hat{\Theta}$ that maximize $\log L(\Theta; X)$, we set the derivatives to be zero, i.e., $\frac{\partial \log L}{\partial \mu_j} = 0$, $\frac{\partial \log L}{\partial \sigma_j} = 0$, and $\frac{\partial \log L}{\partial r_i} = 0$, which yields,

$$\mu_j = \frac{\sum_{i=1}^{n} d_{ij} r_i x_{ij}}{\sum_{i=1}^{n} d_{ij} r_i^2} \hspace{1cm} (4.21)$$

$$\sigma_j = \left( \frac{\sum_{i=1}^{n} d_{ij} (x_{ij} - \mu_j)^2 r_i^2}{\sum_{i=1}^{n} d_{ij}} \right)^{\frac{1}{2}} \hspace{1cm} (4.22)$$
\[ r_i = \left( \frac{\sum_{j=1}^{m} d_{ij}}{\sum_{j=1}^{m} d_{ij}} \left( x_{ij} - \mu_j \right)^2 / \sigma_j^2 \right)^{\frac{1}{2}} \]  

(4.23)

It is impossible to directly solve these equations to obtain the MLEs \( \hat{\mu}_j \), \( \hat{\sigma}_j \) and \( \hat{r}_i \). Instead, we approach it in an iterative manner: given the values \( \mu_j^{(k)} \), \( \sigma_j^{(k)} \) and \( r_i^{(k)} \), we use the equations (4.21)-(4.23) to obtain the values \( \mu_j^{(k+1)} \), \( \sigma_j^{(k+1)} \) and \( r_i^{(k+1)} \); the iterative process continues until the results converge. The initial values of the iterative process is set as:

\[ \mu_j^{(0)} = \frac{\sum_{i=1}^{n} x_{ij} d_{ij}}{\sum_{i=1}^{n} d_{ij}}, \quad \sigma_j^{(0)} = \frac{\sum_{i=1}^{n} |x_{ij} - \mu_j^{(0)}| d_{ij}}{\sum_{i=1}^{n} d_{ij}}, \quad r_i^{(0)} = 1 \]

### 4.4 Achieving Specified Credibility with Min-Overhead

In this section, we study another important problem which aims to achieve a specified data credibility with minimum network overhead. User selection in this problem is more challenging since it is difficult to evaluate whether the data credibility can be satisfied with the selected users. For example, if only a small group of users are selected to keep the overhead in a low level, the required data credibility may not be guaranteed; if many users are selected to ensure data credibility, a large amount of network overhead may be incurred.

To address this problem, we propose an iterative user selection approach: in each iteration, only a limited group of users are selected, and the data credibility are evaluated based on all collected data; the iterative process continues selecting users until the required data credibility is achieved.

In this section, we first have an overview of the iterative approach, and then discuss in detail how to select users and evaluate credibility in each iteration.

#### 4.4.1 An Overview of the Iterative Approach

Figure 4.2 shows an overview of the iterative approach. In each iteration, the server first selects a group of users from the unselected users to collect data. Then, the true value is estimated based on all the collected data values including those collected from previous iterations. Afterwards, the server evaluates the data cred-
ibility and checks if the required data credibility is satisfied. If so, the iterative process ends. Otherwise, the server starts a new iteration and selects another group of users.

In this iterative approach, we propose solutions to select the users to query in each iteration and estimate the true value from the collected data, and find techniques to evaluate the data credibility.

4.4.2 Selecting Users and Estimating Truth

To minimize the network overhead, we limit the network overhead in each iteration when selecting users. Specifically, the network overhead in each iteration is limited by $\beta^o$, which is a pre-defined parameter and can be set flexibly. With the limitation on network overhead, the problem becomes selecting a group of users that are more likely to provide high-quality data, which is equivalent to the max-credibility problem studied in Section 4.3.

When collecting data from users, there is a time constraint $T_j$ for each sensing task $j$. Here, we set the time constraint for collecting data in one iteration to be $\tau^o$ with $\tau^o < T_j$. The parameter $\tau^o$ is also pre-defined and can be set flexibly. As $\tau^o$ is smaller, more iterations may be performed within $T_j$.

To estimate the truth from the collected data, the MLE-based truth analysis approach presented in Section 4.3.2 is applied here. The observed data applied to MLE include all data values collected in the current and previous iterations.
4.4.3 Evaluating Data Credibility

Since the ground truth is unknown, it is impossible to directly calculate data credibility from the collected data. Instead, we calculate data credibility in a probabilistic manner, and ensure that the specified data credibility reaches a confidence level.

For each task $j$, instead of achieving a specified data credibility by constraining the normalized error by $\epsilon$, $|\mu_j - \hat{\mu}_j| - \sigma_j \leq \epsilon$, we ensure the normalized error to be smaller than $\epsilon$ with confidence $1 - \alpha$, i.e., $P\left(\frac{|\mu_j - \hat{\mu}_j|}{\sigma_j} \leq \epsilon\right) > 1 - \alpha$, which is equal to:

$$P(\mu_j \in [\hat{\mu}_j - \epsilon \sigma_j, \hat{\mu}_j + \epsilon \sigma_j]) > 1 - \alpha$$ (4.24)

In this formula, $[\hat{\mu}_j - \epsilon \sigma_j, \hat{\mu}_j + \epsilon \sigma_j]$ is in fact the $1 - \alpha$ confidence interval for the ground truth $\mu_j$. As long as we find the $1 - \alpha$ confidence interval for $\mu_j$ is smaller than $[\hat{\mu}_j - \epsilon \sigma_j, \hat{\mu}_j + \epsilon \sigma_j]$, the inequity in formula (4.24) can be satisfied, so that the requirement on data credibility can be ensured. Thereafter, evaluating data credibility has become calculating the confidence interval for $\mu_j$.

The confidence interval for $\mu_j$ can be calculated based on one of the asymptotic properties of MLE, i.e., asymptotic normality [70]:

**Theorem 1. Asymptotic Normality:** Distribution of MLE estimators for a parameter $\theta$ is asymptotically normal with mean $\theta$ and variance $\text{var}(\theta)$, which can be approximated by the inverse of the Fisher information $I(\theta)$:

$$I(\theta) = E_\theta((\frac{\partial}{\partial \theta} \log f(X|\theta))^2) = -E_\theta(\frac{\partial^2}{\partial \theta^2} \log f(X|\theta))$$ (4.25)

Accordingly, the MLE $\hat{\mu}_j$ is asymptotically normal with mean $\mu_j$, and its variance $\text{var}(\mu_j)$ is approximated by:

$$\frac{1}{I(\mu_j)} = -E_{\mu_j}(\frac{\partial^2}{\partial \mu_j^2} \log f(X|\mu_j)) = \frac{\sigma_j^2}{\sum_{i=1}^n d_{ij} r_i^2}$$ (4.26)

From the property of normal distribution, it is easy to obtain the $1 - \alpha$ confidence interval for $\mu_j$, i.e.,

$$[\hat{\mu}_j - Z_{\alpha/2}(\frac{1}{\sqrt{I(\mu_j)}}), \hat{\mu}_j + Z_{\alpha/2}(\frac{1}{\sqrt{I(\mu_j)}})]$$
\begin{equation}
\hat{\mu}_j - Z_{\alpha/2} \frac{\sigma_j}{\sqrt{\sum_{i=1}^{n} d_{ij}^2}} \leq \mu_j \leq \hat{\mu}_j + Z_{\alpha/2} \frac{\sigma_j}{\sqrt{\sum_{i=1}^{n} d_{ij}^2}}
\end{equation}

where $Z_{\alpha/2}$ is the $\alpha/2$ quantile of the standard normal distribution.

As long as the $1 - \alpha$ confidence interval for $\mu_j$ is smaller than $[\hat{\mu}_j - \epsilon \sigma_j, \hat{\mu}_j + \epsilon \sigma_j]$, the requirement on data credibility is satisfied. Otherwise, the server will start another iteration.

### 4.5 Performance Evaluations

In this section, we evaluate the performance of the proposed max-credibility approach and min-overhead approach using trace-driven simulations and testbed-based experiments.

#### 4.5.1 Trace-Driven Simulations

In our simulation, the mobile opportunistic network is based on a real network trace Infocom 06 [71] which records the pairwise contact information between 98 mobile users in a conference environment. The server is chosen as the one with the highest centrality in the network, where centrality measures the popularity of a node in the network [10]. The reliability $r_i$ of user $i$ is randomly generated within $[0, 1]$ with uniform distribution.

In the experiments, we generate 100 sensing tasks. The ground truth $\mu_j$ for task $j$ is randomly generated within $[0, 10]$. The base number $\sigma_j$ is generated within $[0.2, 5]$. For each sensing task $j$, the data value observed by user $i$, i.e., $x_{ij}$, follows normal distribution $N(\mu_j, (\sigma_j/r_i)^2)$.

#### 4.5.1.1 Results on the Max-Credibility Approach

The proposed max-credibility approach is compared with the following approaches:

- **Max-credibility without considering network reachability**: This is based on our proposed max-credibility approach but does not consider the factor of network reachability when selecting users.

- **Max-credibility without considering user reliability**: It is also based on the
proposed max-credibility approach, but does not consider the user reliability in the statistical model. All users are assumed to be reliable.

In addition to the above two approaches, we also compare our max-credibility approach with the existing approaches for truth analysis. Since existing approaches are all designed for categorical data, to be applicable to numerical data, we assume that two data values are equal as long as their difference is within a fixed value (set to $\epsilon$). These approaches use different methods to calculate reliability and credibility, which are defined as follows.

• **Hubs and Authorities (Sums) [72]**: The reliability of a source is the sum of the credibility of the data items it provides, and the credibility of a data item is the sum of the reliability of sources that provide the data.

• **Average-Log [23]**: Different to Hubs and Authorities, it computes the reliability of each source by multiplying the average credibility of its provided data item and the logarithm of the number of its provided data item.

• **TruthFinder [21]**: The credibility of an observed data item is the probability that it is accurate and the reliability of the source is the probability that it provides accurate data. The credibility of a data item is computed as the probability that at least one source can provide accurate data. A source’s reliability is computed by averaging the credibility of its provided data.

To evaluate the performance of the max-credibility approach, we measure the average normalized estimation error of the truth value. The normalized estimation

...
error measures the deviation of the estimated value from the ground truth, and may be larger than 1. Since maximizing data credibility is equal to minimizing the estimation error, a smaller estimation error is always preferred.

In the experiments, we respectively vary the two parameters, i.e., the limit on network overhead $\beta$ and the time constraint of tasks. The first experiment varies $\beta$ from 15 to 50 while setting time constraint to two hours, and the second experiment varies time constraint from one to five hours and limits the network overhead to 20. The results are respectively shown in Figure 4.3 (a) and Figure 4.3 (b).

From the results we can see the max-credibility approach has less estimation error compared with the approaches that eliminate the factor of network reachability or the factor of reliability, indicating the importance of both factors on the proposed approach. We also find that eliminating the factor of reliability results to higher performance downgrade than eliminating the factor of network reachability, highlighting the significance of reliability in our approach.

Our max-credibility approach also achieves much less estimation error compared with the three existing approaches on truth analysis. There are two reasons. First, by incorporating the factors of network reachability and reliability, more effective users can be selected. Second, the truth can be estimated effectively by applying the MLE-based truth analysis technique.

### 4.5.1.2 Results on Min-Overhead

Basically, the min-overhead approach is an iterative approach based on the max-credibility approach. Therefore, in respect to the five approaches in comparison as listed in Section 4.5.1.1, there are also five corresponding min-overhead approaches in comparison. For the two approaches that eliminate one factor from max-credibility, we determine when the iteration ends by computing confidence interval. For the three existing approaches in comparison, there is not a straightforward way to compute confidence interval, so we simply let the iteration ends as there are a fixed number of users sending results back.

In these experiments, 100 sensing tasks are generated. The time constraint is set to 12 hours. The requirement on the data credibility is $\frac{|\mu - \hat{\mu}|}{\sigma} < 0.5$ and we set the confidence level to be $1 - \alpha = 95\%$. Since the confidence level is high,
we find the requirement on data credibility $|\mu - \hat{\mu}| / \sigma < 0.5$ can be satisfied in most cases, which means the success ratio is near 100%. The objective of min-overhead is to minimize the network overhead while achieving the required data credibility. Thereafter, we mainly measure the average network overhead consumed for each task.

In the first experiment, we compare their performance with the variation of the parameter $\beta^o$, the limit on network overhead in each iteration. The time constraint in each iteration $\tau^o$ is set to two hours. As can be seen from Figure 4.4 (a), the min-overhead approach consumes the minimum network overhead in all cases, illustrating the superiority of our approach in minimizing overhead. Moreover, we also notice that the network overhead of our approach reaches minimum when $\beta^o = 20$. As $\beta^o$ is very small, the MLE estimator usually has bad estimation in the first few iterations, since only a few data values are collected. Therefore, it needs more iterations to collect data in order to re-estimate and the resulted overhead is high. As $\beta^o$ is very large, the performance is also bad, since many unnecessary data values are collected and much network overhead is wasted.

The second experiment changes the other parameter $\tau^o$, the time constraint in each iteration. $\beta^o$ is set to 20. From Figure 4.4 (b), we can observe a clear superiority of our approach over other approaches. An interesting observation is that less network overhead is consumed as $\tau^o$ is larger. There are two reasons. One is that only limited iterations can be completed before the time constraint. For example, as $\tau^o$ is three hours, only four iterations can be completed before the
12 hours time constraint, and the network overhead consumed is constrained by \(4 \ast \beta^o = 80\). The other reason is that when \(\tau^o\) is small, there is not enough time for the requested users to send data back in the first few iterations, resulting to many unnecessary users being requested in the following iterations.

When compared with the three existing approaches, our min-overhead approach consumes much less network overhead, demonstrating the effectiveness of our approach in realizing high data credibility with less network overhead.

4.5.2 Testbed-based Experiment

4.5.2.1 Testbed and Experiment Setting

To further evaluate the performance, we also deployed a testbed of mobile opportunistic network with 20 graduate students in two departments on our campus [73]. In the system, students are distributed with Bluetooth-enabled Android smartphones, with which two students are able to communicate and share data with each other whenever they opportunistically move into the Bluetooth communication range. Finally, about seven months of data on networking are collected.

These 20 students are also required to provide answers to 89 questions about their everyday life and basic knowledge in various topics, e.g., the availability of the parking lots on campus, the estimated driving hours to another city in the local state, or the average salary for the software engineers in US. Since the provided answers are noisy, these data can be ideally used for the evaluation of the truth analysis techniques. To evaluate the proposed approaches, we assume that these questions are the sensing tasks provided by the server in the mobile opportunistic network. If a user is queried with a sensing task, the collected data value is his or her answer to the corresponding question.

Since data from 20 students may not be enough for truth analysis, we also surveyed 40 students in other departments on campus to finish the questions. Their rough locations are also recorded. Based on the networking information we have collected in the testbed, we have captured the contact features between the students in the same department and between students from different departments. First, we experimentally verify the claim in [17] that the pairwise contact process follows Poisson process. Moreover, we find the inverse of the pairwise contact rate
$\frac{1}{\lambda}$, i.e., the average inter-contact time, follows log-normal distributions either for the node pairs inside the same department or the node pairs in different departments, with which we are able to generate the contact rates between nodes. Let $\lambda_{in}$ and $\lambda_{out}$ respectively denote the pairwise contact rate between nodes inside the same department and that between nodes in different departments, we have

$$\ln \frac{1}{\lambda_{in}} \sim N(\mu_{in}, \sigma_{in}), \text{ and } \ln \frac{1}{\lambda_{out}} \sim N(\mu_{out}, \sigma_{out})$$

(4.28)

The four parameters $\mu_{in}, \sigma_{in}, \mu_{out}, \sigma_{out}$ are estimated based on our collected networking information in the testbed. Table 4.1 shows the estimation on the parameters and the 95% confidence interval of the estimation (the fourth row). A smaller confidence interval means a better fit to the log-normal distribution. As we can see, the parameter estimations have relatively small confidence intervals with range less than 0.5. This result demonstrates the inverse of pairwise contact rate (i.e., the average inter-contact time) can be well approximated by the log-normal distributions. We also believe if there are more students participated in the testbed, the distribution fit can be even more accurate.

<table>
<thead>
<tr>
<th>Inside department ($\ln \frac{1}{\lambda_{in}}$)</th>
<th>Between department ($\ln \frac{1}{\lambda_{out}}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{in}$</td>
<td>$\sigma_{in}$</td>
</tr>
<tr>
<td>2.09</td>
<td>1.15</td>
</tr>
<tr>
<td>[1.86, 2.33]</td>
<td>[1.01, 1.34]</td>
</tr>
</tbody>
</table>

Based on the extracted contact features, we generate synthetic contact information (contact rate) among the 40 students as well as between the 40 students and 20 students in the testbed. Specifically, for each pair of students $u, v$, their contact rate $\lambda_{uv}$ is generated based on the log-normal distributions in (4.28). Then the detailed contact process between them is generated to follow Poisson process with rate $\lambda_{uv}$. Finally, a mobile opportunistic network of 60 mobile users is built by integrating the synthetic contact information and the authentic networking information collected in the testbed.


Figure 4.5. Comparison of normalized estimation error for the max credibility problem in testbed-based experiment: (a) with different constraint on network overhead $\beta$, and (b) with different time constraint.

4.5.2.2 Results

Based on the testbed and the collected data, we first evaluate our approaches by comparing the performance on the max-credibility problem. The approaches listed in Section 4.5.1.1 are compared. The results are shown in Figure 4.5. The first experiment varies the limit on the network overhead $\beta$ and set the time constraint to be four hours (Figure 4.5 (a)). The second experiment varies the time constraint and set the limit on network overhead to be 40 (Figure 4.5 (b)). As can be seen from the results, our approach achieves smaller estimation error than the other approaches, similar to what we observed in the trace-driven simulation. These results further demonstrate the effectiveness of our approach on maximizing credibility in real network implementation.

Since the most important part of our min-overhead approach is its ability to iteratively and adaptively collect data from users. We next evaluate our min-overhead approach by comparing the iterative approach with the non-iterative approach. In this experiment, we set a stringent requirement on data credibility: $|\mu - \hat{\mu}| < 0.5$ and set the confidence level to be 90%. The performance is evaluated based on the success ratio, which is the percentage of tasks that can successfully satisfy the requirement on data credibility. The result is shown in Figure 4.6. As can be seen, the iterative approach consistently achieves higher success ratio than the non-iterative approach under different network overhead limit. The superior performance on success ratio is because the iterative approach can adaptively collect data from the mobile opportunistic networks when the requirement on data
credibility is not satisfied, but a non-iterative approach is not flexible and cannot collect data adaptively. Even though the non-iterative approach may achieve a high success ratio by requesting a lot of users, it is still difficult to specify what and how many users are queried. Our approach addresses this problem through adaptive collection, achieving high data credibility with low network overhead.

4.6 Related Work

Truth analysis in social sensing or crowdsourcing has received considerable attention recently. Hubs and Authorities [72] is one of the earliest techniques, which iteratively evaluates the correctness of information by evaluating the reliability of data sources. Pasternack et al. [23] extended these frameworks to a more general scenario by incorporating prior knowledge of the information. TruthFinder proposed by Yin et al. [21] is also based on an iterative method to infer the information correctness. The aforementioned techniques are all based on a simple heuristic: the correctness of information is estimated based on the reliability of the information sources, and then the reliability of the information source is estimated based on the correctness of their provided information.

Other than the heuristic-based approaches, researchers also investigated statistical models for truth finding in order to achieve more convincing results. For example, the bayesian inference is utilized in [24][25][26]. An expectation-maximization (EM) based scheme is proposed in [27]. Other truth analysis techniques enhance
the basic framework by considering additional properties and topics, e.g., several techniques [74][75] have discussed the dependency of data sources and how it affects the decision of each other.

Unfortunately, all these prior techniques are passively based on the data that are already available, which may not be enough to obtain satisfactory truth analysis results. Different from the existing techniques, our approach can adaptively collect data through mobile opportunistic networks, so that high-quality data can be collected to ensure the required data credibility.

The research on crowdsourcing mostly focused on the design of incentive or privacy-preserving schemes. For example, [20][76][77] designed incentive schemes for the smartphone-based crowdsourcing and intended to recruit users to minimize expenses, and[78][79] [80] considered the privacy threatens to the crowdsourcing participants and designed privacy-preserving schemes for data collection. Most recently, some work [81][82][83] started to realize the issue on information quality and designed schemes to recruit users to achieve high information quality and minimize total expenses or resources. However, they failed to consider how user reliability influences the quality of information, and lacked a recruiting mechanism to collect enough data to assure high information quality. Our work addresses this problem by adaptively collecting data from users until the required information quality is achieved.

4.7 Conclusion

An important problem in crowdsourcing is to assess the reliability of the mobile users and identify the truth from the reported data. Different from existing truth analysis approaches, we utilize the communication network (i.e. mobile opportunistic network) to adaptively collect data from some users, so that high-quality data are collected to achieve the required data credibility. Considering the requirement on data credibility and the constraint of network resources, we quantify the tradeoff between the enhanced data credibility and the increased network overhead as two problems: max-credibility which aims to maximize data credibility with some network overhead, and min-overhead which aims to achieve a specified data credibility while minimizing the network overhead. For max-credibility, we
propose a solution to select users based on their data collection capabilities such as network reachability and user reliability, and propose a MLE-based technique to identify truth from the collected data. For Min-overhead, we propose a solution which iteratively selects users for data collection until the required data credibility is achieved. The proposed solutions are evaluated based on trace-driven simulations and a testbed. Both simulation and experimental results demonstrate the effectiveness of the proposed solutions in terms of data credibility and network overhead.
5.1 Introduction

Previous techniques on truth analysis and task allocation in crowdsourcing focus on the reliability of users; i.e., high-reliability users tend to provide high-quality data. Specifically, by inferring and utilizing user reliability, the truth can be better identified. For example, researchers in [21][23][27][25] propose reliability-based truth analysis techniques to estimate the truth by assigning higher weights to users with higher reliability. Similarly, by considering user reliability, tasks can be allocated to users with higher reliability. For example, researchers in [84][68][85] provide solutions to identify and recruit high-reliability or high-quality data sources in various mobile crowdsourcing scenarios, such as vehicle-based crowdsourcing [84], crowdsourcing over opportunistic networks [68], and urban sensing through mobile phones [85].

However, these existing techniques are based on the assumption that the reliability of a user does not change with tasks, while neglecting the fact that a user may only have expertise on some problems (in some domains), but not others. Neglecting this expertise diversity may cause two problems: low estimation accuracy in truth analysis and ineffective task allocation. First, by applying a reliability-based truth analysis technique, a user inferred to have high reliability
may actually have very low expertise in some domains. Then, for tasks in these domains, the estimated truth may not be correct since the user with “high-reliability” may provide wrong data due to lack of expertise in these domains. Second, task allocation will also be affected since the identified “high-reliability” users may not actually provide accurate data in all expertise domains. A more severe problem is related to unfair task allocation, i.e., all tasks are assigned to a few users with “high-reliability” while the remaining majority of users are not assigned any task.

Considering these problems, it is important to design expertise-aware solutions for truth analysis and task allocation. However, it is a challenge to design expertise-aware solutions due to the following two reasons. First, it is hard to identify user expertises and the expertise domains of the tasks without any prior knowledge on user behaviors and the ground truth of the tasks. The available information only includes the task descriptions and the collected sensing data from users. Second, allocating tasks to users with the highest expertise is hard to achieve in many cases. This is because some users may have high expertise in multiple domains, but only have limited processing capability, and then can not finish all the assigned tasks within the time limit.

In this chapter, we propose an *Expertise-aware Truth Analysis and Task Allocation (ETA²)* approach to address the aforementioned challenges. ETA² can effectively infer user expertise and then allocate tasks and estimate truth based on the inferred expertise. More specifically, we have the following contributions:

- We first propose techniques to identify the expertise domains of the tasks. Specifically, we design a novel semantic analysis method to extract and quantify the semantic information of tasks based on the task descriptions. Then, we propose a *dynamic hierarchical clustering* approach to cluster tasks based on their extracted semantic information, so that each cluster corresponds to one expertise domain and the tasks inside the cluster belong to the corresponding expertise domain.

- For *expertise-aware truth analysis*, we build expertise-aware statistical models by combining the expertise models of users and tasks, and apply *maximum likelihood estimation (MLE)* to estimate the truth and learn user expertise.

- For *expertise-aware task allocation*, we formalize an optimization problem
which maximizes the probability that tasks are allocated to users with the right expertise, while ensuring that the workload at each user does not exceed its maximum processing capability.

The rest of the chapter is organized as follows. Section 5.2 introduces some background and gives an overview of ETA\(^2\). Section 5.3 presents how to identify the expertise domains of the tasks. Section 5.4 and Section 5.5 present expertise-aware truth analysis and expertise-aware task allocation, respectively. Evaluation results are discussed in Section 5.6. Section 5.7 reviews the related work and Section 5.8 concludes the chapter.

5.2 Preliminary

5.2.1 Background

The system we consider includes a crowdsourcing server who is requiring data and multiple users who are able to communicate with the server using their mobile devices (e.g., through mobile opportunistic networks). A group of tasks are created at the server at each time step (e.g., each day). After a task is created, the server allocates it to a set of selected users specifying the required data, the task deadline and the estimated processing time required for completing the task. Multiple users may be queried for each task considering that the data collected from a single user may be inaccurate. Then, the selected users collect data as specified by the task descriptions and send data back to the server. The collected data item is accepted by the server only if it is sent back within the time limit. After receiving all provided data for a task, the server estimates the truth using a truth-estimation technique.

Without loss of generality, we assume the processing capability is limited at each user, i.e., only limited time can be used for collecting data during each time step. Therefore, only limited number of tasks can be performed at each user.

Here, the collected sensing data are considered to be numerical values. Then, the magnitude of the data may vary tremendously for different tasks. To ensure different tasks can be processed using the same statistical model, the data value for a task is normalized.
5.2.2 An Overview of ETA²

Figure 5.1 shows an overview of ETA², which has the following three main modules:

- **Identifying Task Expertise**: This module is used to find the expertise domains of the tasks. The basic idea is to extract the semantic information of each task by applying a semantic analysis method. Then, the tasks are clustered based on their extracted semantic information, so that each cluster represents one expertise domain, and the tasks inside the cluster belong to the corresponding expertise domain.

- **Expertise-aware Truth Analysis**: This module is used to infer the truth after data have been collected from users. The basic idea is to build a statistical model where the ground truth and the user expertise are treated as unknown parameters. By applying statistical inference based on the collected data, the truth can be estimated and the user expertise can be identified. The identified user expertise can be used for task allocation.

- **Expertise-aware Task Allocation**: This module is used to assign the newly created tasks to users according to their expertise. The basic idea is to ensure that each task is allocated to users with high expertise.

The process as shown in Figure 5.1 starts with a warm-up period, after the **Start** button. When the system first starts, there is no prior knowledge about user
expertise. Thus, in the warm-up period, the tasks are allocated to users randomly. After data have been collected from users, user expertise can be learned. The learned user expertise can be further utilized for task allocation.

After the warm-up period, ETA^2 starts an iterative process, as shown by the dotted square in Figure 5.1. When new tasks are created in each iteration (time step), the server first finds the expertise of each task. Then, the server applies the expertise-aware task allocation technique to assign tasks to users with the right expertise. Then, users collect data as specified by the task description and send the data back to the server. After receiving the data, the server estimates the truth by applying expertise-aware truth analysis techniques and updates user expertise by incorporating the newly collected data.

5.2.3 Distribution of Random Observation

We assume the random observations of users for a task follow normal distribution. To validate this assumption, we conduct an experiment to find the distribution of the observation error based on real-world datasets. More information about the datasets can be found in Section 5.6. The observation error is simply computed as the difference between the observation and the ground truth divided by the standard deviation among all observations. Figure 5.2 shows the result. The figure also shows the probability density function of the standard normal distribution. As can be seen, the error of the observed data follows the standard normal distribution.
very well. This result also implies that the random observations of users can be approximated by normal distribution, with the mean to be the ground truth of the tasks.

5.2.4 Expertise Model

In our expertise model, there are $D$ expertise domains, where $D$ is not fixed and may be increased when new tasks are added to the system. Each sensing task $j$ belongs to one expertise domain, denoted as $d_j$. The expertise profile of a mobile user $i$ is represented by a $D \times 1$ vector:

$$ U^i = [u^i_1, u^i_2, ..., u^i_D]^T \quad (5.1) $$

where $u^i_k$ is user $i$'s expertise in domain $k$ ($u^i_k \geq 0$). A higher $u^i_k$ means more expertise. $u^i_k = 0$ means $i$ has no expertise in domain $k$. A user may have expertise in multiple domains.

The expertise of user $i$ for task $j$, represented as $u^i_{d_j}$, determines the quality of the data provided by the user. Having higher expertise means that the user is likely to provide higher-quality data for the task. A user with lower expertise is likely to provide data deviating from the ground truth. Specifically, we assume that the observation of user $i$ for task $j$ follows normal distribution $N(\mu_j, (\sigma_j/u^i_{d_j})^2)$, where $\mu_j$ is the ground truth of task $j$, and $\sigma_j/u^i_{d_j}$ is the standard deviation. Here, $\sigma_j$ is the base number of task $j$, which is used to normalize the data value of task $j$. $\sigma_j$ is unknown but can be learned as presented in Section 5.4.

5.3 Task Expertise Identification

5.3.1 Basic Idea

To design expertise-aware truth analysis, we need to first identify the expertise domains of the tasks. However, there is no existing expertise information, and then we cannot simply classify tasks to existing expertise domains. In addition, there is no explicit information to quantify or categorize a task. The only information available for a task is the task description. Although there are many existing
techniques [86][87] to identify the expertise domains or topics of documents by statistically analyzing the appearance of words in the documents, they cannot be directly applied here, because they require the documents to be long enough for effective statistical results. However, the task descriptions are usually short, and hence the expertise domains cannot be effectively identified by applying existing techniques.

To address this problem, we design a semantic analysis method, called pair-word, to extract and quantify the semantic information of the tasks based on the task descriptions. With the extracted semantic information, we are able to measure the distance (or similarity) between tasks. Then, we can cluster these tasks based on their distance so that each cluster represents one expertise domain and the tasks inside this cluster belong to the corresponding expertise domain. Specifically, a dynamic hierarchical clustering approach is proposed to cluster tasks and identify expertise. As new tasks arrive, the dynamic hierarchical clustering approach can dynamically identify their expertise domains by creating new clusters or merging to existing clusters. In the rest of the section, we first discuss how to extract the semantic information with our pair-word method, and then present the dynamic hierarchical clustering approach.

### 5.3.2 Semantic Information Extraction

To effectively extract the semantic information from the task description, we design a pair-word based method to identify two types of important terms within each description sentence: *Query* term, which refers to the words or phrases that describe the requirement of a specific task, and *Target* term, which contains the desired information. For example, the following shows two tasks and their identified *Query* and *Target* terms.

- **Task 1**: What is the noise level around the municipal building?
  
  *Query*: noise level; *Target*: municipal building

- **Task 2**: How many students have attended the seminar today?
  
  *Query*: students; *Target*: seminar
We utilize distributed semantics of \(< Query, Target >\) to capture the meaning of each task description. Word embedding is an efficient technique to map each word or phrase to a low-dimensional vector based on their global contexts. We use the Continuous Skim-gram model [88] to learn lexical representation for each single word from the entire Wikipedia dump (August 11, 2014). For multi-word terms, a simple element-wise additive model \((V = x_1 + x_2 + ... + x_i)\) [88] is exploited, where \(V\) represents a phrase embedding and \(x_1, x_2, ..., x_i\) represent the individual embeddings of the words in \(V\). We concatenate the vector representation of Query term \(V_Q\) and Target term \(V_T\) for each task description and use Euclidean distance metric to measure the distance between two tasks \(i\) and \(j\) based on their semantic vectors:

\[
E(i, j) = \frac{1}{2} \| [V_Q^i, V_T^i] - [V_Q^j, V_T^j] \|^2 \tag{5.2}
\]

where \([V_Q, V_T]\) denotes the concatenation of two vectors \(V_Q\) and \(V_T\) for each task. With this pair-word extraction method, we can efficiently capture the semantic information that two tasks shared.

5.3.3 Dynamic Hierarchical Clustering

5.3.3.1 Hierarchical Clustering

Based on the distance metric between tasks, tasks are clustered together. Although there are many clustering techniques in the literature [89], we select hierarchical clustering based on the following two reasons. First, with hierarchical clustering, the number of clusters is not fixed. As a result, clusters can be updated and new clusters can be added when new tasks are added. Second, hierarchical clustering is effective and simple with only one parameter \(\gamma\), which is used to quantify the minimum allowed distance between clusters. Here, the cluster distance is calculated as the average distance between tasks in the two clusters. After the hierarchical clustering process, the distance between any two clusters should be equal or larger than the minimum allowed distance. Assume the longest distance between all existing tasks is \(d^*\). The minimum allowed distance can be represented as \(\gamma \cdot d^*\), where \(d^*\) is a fixed value and \(\gamma \in [0, 1]\) is the parameter, which can be flexibly set according to specific requirements.

With a set of \(m\) tasks from the warm-up period, the basic hierarchical clustering
works as follows:

1. **Initialization:** Each of the \( m \) tasks starts its own cluster.

2. **Merging clusters:** Pick two clusters that are closest and merge them. This step repeats until the termination criterion is satisfied, as defined in the next step.

3. **Termination:** The algorithm terminates if the distance between the closest clusters in one round is equal to or larger than the *minimum allowed distance* between clusters, \( \gamma \cdot d^* \).

### 5.3.3.2 Dynamic Hierarchical Clustering

The above algorithm can be directly applied to identify the expertise domains of the tasks in the warm-up period. As new tasks are created, they should be classified to some existing clusters, and the dynamic hierarchical clustering method is proposed to achieve this goal.

Dynamic hierarchical clustering only differs from hierarchical clustering in the initialization step. Assume \( m' \) new tasks are created. For each new task, a new cluster is created, and then \( m' \) new clusters are created. If there are \( M \) existing clusters before new tasks are created, there will be \( M + m' \) clusters in the initialization step. Then, the \( M + m' \) clusters are merged following the same “Merging clusters” process, and it terminates when the termination criterion is satisfied. Recall that each cluster corresponds to one expertise domain. Figure 5.3 lists three types of changes that may happen to existing clusters (expertise domains):

- **Adding new tasks to the existing expertise domain:** If some of the new tasks are close to one existing cluster, after the clustering process, they are simply added to the corresponding expertise domain.

- **Creating a new expertise domain:** If some of the new tasks are close to each other but faraway from any existing clusters, a new cluster will be formed and a new expertise domain is created.

- **Merging existing expertise domains:** If some of the new tasks are close to two or more existing clusters, after the clustering process, these corresponding
expertise domains may be merged to one expertise domain, and the new tasks are added to the merged expertise domain.

5.4 Expertise-Aware Truth Analysis

In this section, we present our expertise-aware truth analysis. Specifically, a statistical model is built based on the expertise models, which treats the truth associated with each task and user expertise as parameters. By applying the technique of Maximum Likelihood Estimation (MLE), both the truth and user expertise in the statistical model can be estimated. We first present the statistical models and the MLE method used to infer the truth and user expertise. Then, we discuss how to dynamically update user expertise when new observations are made from tasks in subsequent time steps.

5.4.1 Estimation of Truth and User Expertise

Based on the collected data for the tasks in the warm-up period, we can find the user expertise and the truth associated with each task using MLE. With MLE, the unknown parameters of a statistical model can be estimated given the observed data. In our statistical model, the observed data is the data provided by the users for the sensing tasks. The set of observed data is denoted as

\[ X = \{X_1, X_2, ..., X_m\}, \]
where \( m \) is the number of tasks, and \( X_j \) is the set of data provided for task \( j \). The unknown parameters of our statistical model include the expertise of each user \( i \) in all of the \( D \) domains, i.e., \( u_{1i}^j, u_{2i}^j, ..., u_{Di}^j \), the truth for each task \( j \), i.e., \( \mu_j \), and the base number for each task \( j \), i.e., \( \sigma_j \). Overall, the set of unknown parameters is represented as

\[
\Theta = \bigcup_{i=1}^n \bigcup_{j=1}^m \{ u_{1i}^j, u_{2i}^j, ..., u_{Di}^j, \mu_j, \sigma_j \}
\]

We use \( \omega_{ij} \in \{0, 1\} \) to denote whether user \( i \) has provided data for task \( j \). If \( \omega_{ij} = 1 \), user \( i \) has provided data for task \( j \), and the data is denoted as \( x_{ij} \); otherwise, \( \omega_{ij} = 0 \).

If \( \omega_{ij} = 1 \), the probability density function (pdf) of \( x_{ij} \) is

\[
f(x_{ij} | \Theta) = \frac{1}{\sigma_j/u_{dj}^j \sqrt{2\pi}} e^{-(x_{ij} - \mu_j)^2 / 2\sigma_j^2/u_{dj}^j}.
\] (5.3)

From the above equation, we can compute the pdf (or likelihood function) that \( X \) is observed as

\[
f(X | \Theta) = \prod_{j=1}^m f(X_j | \Theta) = \prod_{j=1}^m \prod_{i=1}^n (f(x_{ij} | \Theta))^{\omega_{ij}}
= \prod_{j=1}^m \prod_{i=1}^n \left( \frac{1}{\sigma_j/u_{dj}^j \sqrt{2\pi}} e^{-(x_{ij} - \mu_j)^2 / 2\sigma_j^2/u_{dj}^j} \right)^{\omega_{ij}}.
\] (5.4)

With \( \omega_{ij} \) being the exponent of \( f(x_{ij} | \Theta) \), \( f(X | \Theta) \) only multiplies \( f(x_{ij} | \Theta) \) with \( \omega_{ij} = 1 \).

Given the likelihood function, MLE estimates the parameters by computing the parameter set \( \hat{\Theta} \) that maximize the likelihood function. Maximizing the likelihood function is equal to maximizing the log-likelihood function, which is

\[
\log L(\Theta; X) = \log f(X | \Theta)
= \sum_{i=1}^m \sum_{i=1}^n \omega_{ij} \left[ \log \left( \frac{1}{\sigma_j/u_{dj}^j \sqrt{2\pi}} \right) - \frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2/u_{dj}^j} \right]
\] (5.5)

Then, \( \hat{\Theta} \) is computed by setting the derivatives of the log-likelihood function
log $L(\Theta; X)$ over each parameter to be 0. With some derivation, we get

$$
\mu_j = \frac{\sum_{i=1}^{n} \omega_{ij} u_{ij}^k x_{ij}}{\sum_{i=1}^{n} \omega_{ij} u_{ij}^k},
$$

(5.6)

$$
\sigma_j = \left( \frac{\sum_{i=1}^{n} \omega_{ij} (x_{ij} - \mu_j)^2 u_{ij}^k}{\sum_{i=1}^{n} \omega_{ij}} \right)^{\frac{1}{2}},
$$

(5.7)

$$
u_{ik} = \left( \frac{\sum_{j=1}^{m} I(d_j = k) \omega_{ij}}{\sum_{j=1}^{m} I(d_j = k) \omega_{ij} (x_{ij} - \mu_j)^2 / \sigma_j^2} \right)^{\frac{1}{2}},
$$

(5.8)

where $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, m\}$ and $k \in \{1, 2, \ldots, D\}$. In Equation 5.8, $I(d_j = k) = 1$ if $d_j = k$ is true, and $I(d_j = k) = 0$ otherwise. Though it is hard to get a close-form solution for each of the parameters, we can get the estimation of the parameters by iteratively computing the values based on Equations 5.6, 5.7 and 5.8. To start the iterative process, we first set the initial values for the user expertise to be 1 ($\nu_{ik} = 1$, $\forall i, k$), and then use the three equations to iteratively compute the values of $\mu_j$ and $\sigma_j$ and $\nu_{ik}$ until the calculated values converge.

### 5.4.2 Dynamic Update of User Expertise

With the aforementioned method, user expertise is estimated based on the observations of the existing tasks. After new tasks are created, user expertise should be updated based on new observations. As discussed in Section 5.3.3.2, after new tasks are added, there are three types of changes to the expertise domains as shown in Figure 5.3. Corresponding to these three types of changes, there are also three types of updates to user expertise.

#### 5.4.2.1 Adding new tasks to existing expertise domain

When new tasks are added to an existing expertise domain, user expertise in this domain should be updated by incorporating users’ observed data for these new tasks. User expertise is updated based on Equation 5.8, which is used to compute the user expertise $\nu_{ik}$ in the warm-up period. For the part inside the parentheses, which is $(\nu_{ik})^2$, the numerator and the denominator have different meanings. The numerator $\sum_{j=1}^{m} I(d_j = k) \omega_{ij}$, denoted as $N(\nu_{ik})$, represents the number of tasks
from expertise domain $k$ that have been finished by user $i$. The denominator
$\sum_{j=1}^{m} I(d_j = k) \omega_{ij} (x_{ij} - \mu_j)^2 / \sigma_j^2$, denoted as $D(u_k^i)$, represents the sum of squared estimation errors for those tasks.

To update user expertise $u_k^i$, we maintain the numerator $N(u_k^i)$ and the denominator $D(u_k^i)$ of the entity $(u_k^i)^2$. Suppose the current time step is $T$, and the $t$ is the length of one time step. After new tasks from expertise domain $k$ are finished by user $i$ during the new time step, $N(u_k^i)$ and $D(u_k^i)$ are updated as follows:

$$N(u_k^i)^{T+t} = \alpha \cdot N(u_k^i)^T + \sum_{j=1}^{m'} I(d_j = k) \omega_{ij},$$

$$D(u_k^i)^{T+t} = \alpha \cdot D(u_k^i)^T + \sum_{j=1}^{m'} I(d_j = k) \omega_{ij} (x_{ij} - \mu_j)^2 / \sigma_j^2,$$

where $\alpha \in [0, 1]$ is the decaying factor placed on the original value to undermine the influence of the historical tasks, and $m'$ is the number of tasks created in the current time step. Then the user expertise is updated based on $N(u_k^i)^{T+t}$ and $D(u_k^i)^{T+t}$:

$$u_k^i^{T+t} = \left( \frac{N(u_k^i)^{T+t}}{D(u_k^i)^{T+t}} \right)^{\frac{1}{2}}$$

When computing $D(u_k^i)^{T+t}$ with Equation 5.10, the true value $\mu_j$ and base number $\sigma_j$ for new task $j$ are unknown a priori. To address this problem, $\mu_j$ and $\sigma_j$ are first estimated using Equations 5.6 and 5.7, in which the user expertise is initialized to the original values in time $T$. Then, we can update user expertise based on Equation 5.11. Since the values of $\mu_j$ and $\sigma_j$ computed from Equations 5.6 and 5.7 may be changed after the user expertise is updated, we apply the same iterative process to update $\mu_j$, $\sigma_j$ and $(u_k^i)^{T+t}$ until they converge. At the end of the iterative process, the user expertise is updated, and the truth values $\mu_j, j \in \{1, ..., m'\}$ for the new tasks are identified.

5.4.2.2 Creating a new expertise domain

If there exist some new tasks that are close to each other but far from any existing expertise domains, a new expertise domain $D + 1$ is created for them. The user expertise in the new domain, the truth and the base number of the new tasks in
this domain are estimated using Equations 5.6–5.8, just as those in the warm-up period. The number of expertise domain \( D \) is added by 1.

5.4.2.3 Merging existing expertise domains

If two existing expertise domains \( k_1 \) and \( k_2 \) are merged by including new tasks that are close to both \( k_1 \) and \( k_2 \), the user expertise in the two domains should also be updated. Specifically, the user expertise in domain \( k_1 \) is updated by incorporating tasks in \( k_2 \), and \( k_2 \) is deleted. Specifically, \( N(u_{k_1}^i) \) and \( D(u_{k_1}^i) \) in domain \( k_1 \) are updated by adding those in domain \( k_2 \):

\[
N(u_{k_1}^i) = N(u_{k_1}^i) + N(u_{k_2}^i),
\]

\[
D(u_{k_1}^i) = D(u_{k_1}^i) + D(u_{k_2}^i).
\]

After merging these two expertise domains, \( N(u_{k_1}^i) \) and \( D(u_{k_1}^i) \) are updated by incorporating tasks in the new time step. The updating process is identical to adding new tasks to existing expertise domain as shown in Section 5.4.2.1.

5.5 Expertise-Aware Task Allocation

As a task is created, it should be allocated to users with higher expertise for that task. However, since each user has limited processing capability, i.e., only limited time is available for completing tasks each day, it is possible that a user cannot finish all the assigned tasks. To address this problem, we formalize an optimization problem which maximizes the possibility that tasks are allocated to users with the right expertise, while ensuring the work load does not exceed the processing capability at each user. By proving the optimization problem to be NP-hard, we further propose a heuristic based algorithm as the approximation solution.

5.5.1 The Optimization Problem

5.5.1.1 Objective function

We first compute the probability that at least one user can provide accurate data for task \( j \):
where $p_{ij}$ is the probability that user $i$ can provide accurate data for task $j$. We consider the observed data to be accurate if its normalized error is smaller than $\epsilon$, where the normalized error is computed as the error to the ground truth divided by the base number and $\epsilon$ is a small constant and set to 0.1 in this work. Then, the probability that user $i$ can provide accurate data for task $j$ can be computed as follows:

$$p_{ij} = P\left(\frac{|x_{ij} - \mu_j|}{\sigma_j} < \epsilon\right)$$

$$= \int_{\mu_j - \epsilon \sigma_j}^{\mu_j + \epsilon \sigma_j} \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(x_{ij} - \mu_j)^2}{2\sigma_j^2}} dx_{ij}$$

$$= \Phi(\epsilon u_{d_j}^i) - \Phi(-\epsilon u_{d_j}^i)$$

The objective function; i.e., the sum of the probability that at least one user can provide accurate data for each task, is as follows:

$$\sum_{j=1}^{m} p_j = \sum_{j=1}^{m} \left[1 - \prod_{i=1}^{n} (1 - p_{ij})\right]$$

$$= \sum_{j=1}^{m} \left[1 - \prod_{i=1}^{n} (1 - \Phi(\epsilon u_{d_j}^i) + \Phi(-\epsilon u_{d_j}^i))\right]$$

By maximizing the objective function, the users with high expertise are selected with high priority, since users with high expertise are more likely to provide accurate information.

### 5.5.1.2 Constraints

The processing capability of each user $i$ is denoted as $c_i$, which is the available time for user $i$ to spend on processing tasks. The processing time for each task $j$ is denoted as $t_j$. $s_{ij} \in \{0, 1\}$ is used to denote whether task $j$ will be allocated to user $i$. Then, the constraints on the processing capability at each user can be
represented as:
\[
\sum_{j=1}^{m} t_j \cdot s_{ij} < c_i, \quad \forall i \in \{1, 2, ..., n\} \quad (5.17)
\]

Thus, the optimization problem can be formalized as follows:

\[
\begin{align*}
\max & \quad \sum_{j=1}^{m} \left[ 1 - \prod_{i=1}^{n} \left( 1 - \Phi(\epsilon u_{d_j}^i) + \Phi(-\epsilon u_{d_j}^i) \right)^{s_{ij}} \right] \\
\text{s.t.} & \quad \sum_{j=1}^{m} t_j \cdot s_{ij} < c_i, \quad \forall i \in \{1, ..., n\} \\
& \quad s_{ij} \in \{0, 1\}, \quad \forall i \in \{1, ..., n\}, j \in \{1, ..., m\} 
\end{align*}
\]

(5.18)

The optimization problem is NP-hard due to the following reasons. Considering the case where there is only one user, the problem can be easily reduced to a knapsack problem, where the knapsack is identical to the user with limited processing capability, and the items are identical to the tasks to be assigned. Since the knapsack problem has non-integer weight and item values, the problem is NP-hard [59], from which we prove the optimization problem with single user is NP-hard. Therefore, the optimization problem with \( n \) users is also NP-hard.

5.5.2 Heuristic Based Algorithm

Since the optimization problem is NP-hard, we propose a heuristic based algorithm. The basic idea is to greedily select user-task pairs which can add more value to the objective function while ensuring the tasks consume less processing time. For each selected user-task pair, the task will be assigned to that user.

In the heuristic based algorithm, we first define a concept \textit{efficiency} for each user-task pair (denoted as \textit{efficiency}(i, j) for user \( i \) and task \( j \)) to quantify its importance in increasing the value of the objective function. Let \( c_i' \) denote the remaining processing capability of user \( i \). The \textit{efficiency} concept is formally defined as follows:

\textbf{Definition 4.} The \textit{efficiency} of user-task pair \((i, j)\) is calculated as the value increase of the objective function divided by the processing time \( t_j \) of task \( j \) if task \( j \) is assigned to user \( i \). However, if the remaining processing capability \( c_i' \) of user \( i \)
is not enough to finish task \( j \), i.e., \( t_j > c'_i \), the efficiency is set to zero.

Specifically, the value increase of the objective function is computed as

\[
p_j^{\text{new}} - p_j = \left[ 1 - (1 - p_j)(1 - p_{ij}) \right] - p_j = p_{ij}(1 - p_j), \tag{5.19}
\]

where \( p_j^{\text{new}} \) is the updated \( p_j \) after user \( i \) is added. Note that adding a user for task \( j \) does not change other tasks, so we do not need to consider other tasks in the objective function. As a result, the efficiency of \((i, j)\) is computed as:

\[
\text{efficiency}(i, j) = \begin{cases} 
  \frac{p_j^{\text{new}} - p_j}{t_j} = \frac{p_{ij}(1 - p_j)}{t_j}, & \text{if } c'_i \geq t_j \\
  0, & \text{if } c'_i < t_j
\end{cases} \tag{5.20}
\]

The heuristic based algorithm works by greedily adding the user-task pair \((i, j)\) which achieves the maximum efficiency. The greedy process terminates when the maximum efficiency that can be achieved becomes zero, which means the users have used up all their processing capabilities. The general flow of the heuristic algorithm is outlined in Algorithm 6. First, the algorithm computes \( \text{efficiency}(i, j) \) for all user-task pair (Line 2). At the same time, the algorithm maintains the max efficiency \( \text{maxeff}_j \) that can be achieved for each task \( j \), and the user that achieves \( \text{maxeff}_j \), denoted as \( \text{user}_j \) (Lines 3 ~ 6), so that when the greedy algorithm selects the maximum efficiency in each round, it can simply select from the maintained \( \text{maxeff}_j \) for the \( m \) tasks (Line 8). Then, the algorithm greedily selects the user-task pair that achieves the maximum efficiency and updates the efficiency for other unselected pairs (Lines 8 ~ 16). The algorithm terminates when the max efficiency is equal to zero (Lines 10 ~ 12).

To select each user-task pair, \( O(m) \) time is used to find the pair with the highest efficiency. After selecting the user-task pair, \( O(m + n) \) time is used to update the efficiency of the remaining unselected user-task pairs, because only the pairs associated with the selected user or the selected task need to be updated. As a result, the total time used for selecting each user-task pair is \( O(m) + O(m + n) = O(m + n) \). Assuming \( K \) user-task pairs are selected, the time complexity of the
Algorithm 6 Expertise-aware Task Allocation

**Input:** \( u_k, c_i, \forall i, k \), and \( d_j, t_j, \forall j \)

**Output:** \( s_{ij}, \forall i, j \)

1: **Initialize:** \( s_{ij} \leftarrow 0 \)
2: Compute \( \text{efficiency}(i, j) \) according to Equation 5.20
3: **for** Each task \( j \in \{1, 2, ..., m\} \) **do**
4: Find the max efficiency that can be achieved for task \( j \): \( \text{maxeff}_j \)
5: Simultaneously find the user that achieves \( \text{maxeff}_j \): \( \text{user}_j \)
6: **end for**
7: **while** \( \text{true} \) **do**
8: Find max efficiency from \( \text{maxeff}_j \), \( \forall j \in \{1, ..., m\} \)
9: Simultaneously find the task that achieves the max efficiency: \( j^* \)
10: **if** max efficiency = 0 **then**
11: **break**
12: **end if**
13: Select the user-task pair: \( s(\text{user}_{j^*}, j^*) \leftarrow 1 \)
14: Update efficiency for the user-task pairs associated with \( \text{user}_{j^*} \)
15: Update efficiency for the user-task pairs associated with task \( j^* \)
16: Simultaneously update \( \text{maxeff}_{j^*} \) and \( \text{user}_{j^*} \) for task \( j^* \)
17: **end while**

whole process is \( O(K(m + n)) \).

### 5.6 Performance Evaluations

In this section, we evaluate the performance of ETA\(^2\) by conducting experiments based on two real-world datasets.

#### 5.6.1 Datasets

##### 5.6.1.1 Survey-based Dataset

The first dataset is collected through a survey including 60 participants on our campus, after IRB approval. In the survey, each of the participants is required to answer 89 questions about their daily life and basic knowledge on various topics. Some sample questions are listed as follows:

- *How many parking lots on campus are open to students in this semester?*
- *What is the estimated driving hours to another city in the local state?*
• What is the average salary for an entry-level software engineers in United States?

The answers to some questions may depend on specific time and location. For example, the available parking lots to students may be different during weekdays and weekends, and the driving hours may be different during early morning and late afternoon in each day. Therefore, some questions are replicated to consider the conditions at different time and locations. As a result, there are actually 150 questions in the dataset.

The provided answers are noisy, and then can be used for evaluating our approaches. Assuming these questions are the sensing tasks provided by the server, the content of the question is the task description. If a user is queried with a sensing task, the collected data value is the answer to the corresponding question.

5.6.1.2 SFV Dataset

The second dataset [90] is extracted from the Slot-Filling Validation (SFV) task of the 2013 Text Analysis Conference (TAC) Knowledge Base Population (KBP) track. In the task, 18 slot-filling systems are required to answer a set of questions about 100 entities, including famous persons or organizations. The questions are about various properties of the entities, like the age, birthday and name. There are about 2,000 questions for these 100 entities in the original dataset. Similar to the survey-based dataset, each question is treated as a sensing task. The 18 slot-filling systems are treated as users, and theirs answers to the questions are treated as the collected sensing data. In the dataset, documents that describe the entities and their properties are also given, from which we can easily compose the descriptions of tasks. The ground truth of the tasks is also included in the original dataset, which can be used for performance evaluations.

5.6.2 Experimental Setting

In these two datasets, some important information like the processing time required for each task and the processing capability of each user is not provided. Without loss of generality, the processing time required for each task and the processing
capability of each user are generated using uniform distribution. For the survey-based dataset, the processing time required for each task is randomly generated between [2, 4] hours. For the SFV dataset, it is randomly generated between [1, 2] hours. The processing capability of each user is randomly generated between [τ - 4, τ + 4] hours, where τ is a variable representing the average processing capability. τ is set to 12, but can be changed to evaluate how the processing capability affects the performance. The length of the time step is a day in the experiment. The sensing tasks are assumed to be generated and evenly distributed during five days. In the beginning of each day, a group of sensing tasks are generated and allocated to the selected users. At the end of the day, the truth for each task is estimated based on all the collected data values.

In order to have statistically converging results, we set different seeds to randomly select tasks in each day and take the average as the results. Specifically, every experiment is run 100 times by setting different seeds to achieve statistical convergence.

5.6.3 Approaches in Comparison

We compare our proposed expertise-aware solution with three traditional approaches which estimate truth by calculating source reliability, and a baseline approach serving as the lower bound. These approaches in comparison are as follows:

- **Hubs and Authorities** [72]: The reliability of a source is the sum of the credibility of the data items it provides, and the credibility of a data item is the sum of the reliability of sources that provide the data.

- **Average·Log** [23]: Different to Hubs and Authorities, it calculates the reliability of each source by multiplying the average credibility of its provided data item and the logarithm of the number of its provided data item.

- **TruthFinder** [21]: The credibility of an observed data item is the probability that it is accurate and the reliability of the source is the probability that it provides accurate data. Specifically, the credibility of a data item is computed as the probability that at least one source can provide accurate data. A
source’s reliability is calculated by averaging the credibility of its provided data.

- **Baseline**: The truth is estimated as the mean value of the observed data.

These approaches in comparison only specify the method for truth analysis. To allocate new tasks to users, the first three reliability-based approaches greedily allocate tasks to users with high reliability. Considering users only have limited processing capability, we prioritize the tasks with lower sensing time to be allocated to users with high-reliability, so that these high-reliability users can finish as many tasks as possible. For the baseline approach, the tasks are allocated to users randomly.

### 5.6.4 Evaluation Results

In the evaluations, we first show how the parameters $\gamma$ and $\alpha$ affect the performance of ETA$^2$ approach. Then, we evaluate the overall performance of ETA$^2$ by comparing it with existing approaches. Finally, we evaluate the effectiveness of the three modules of ETA$^2$. In the evaluation, the performance metric is the estimation error, which is computed as the average of the normalized estimation error for all sensing tasks.

#### 5.6.4.1 The Effects of Parameters

ETA$^2$ has two parameters $\alpha$ and $\gamma$. When updating user expertise to include the new tasks, a decaying factor $\alpha \in [0, 1]$ is placed on the historical tasks to reduce the influence of the historical tasks. A smaller $\alpha$ means less emphasis is placed on the historical tasks. If $\alpha = 0$, user expertise is estimated based on the new tasks. If $\alpha = 1$, historical tasks are treated equally to the new tasks. Parameter $\gamma \in [0, 1]$ is used to determine the minimum allowed distance between clusters, which determines when the merging process terminates. With a higher $\gamma$, more clusters are merged, resulting in less number of clusters, but the tasks inside the same cluster may be far away from each other. With a smaller $\gamma$, the merging process terminates quickly, the tasks that should belong to the same cluster may not be clustered before termination. Therefore, it is crucial to set appropriate parameters so that the performance of our approach can be optimized.
The first set of experiments evaluate how the two parameters affect the performance of ETA\textsuperscript{2}. We first have an overall evaluation of estimation error under different parameter settings and find the set of parameters which result in better performance. The results are shown in Figure 5.4 (a) for the survey-based dataset and Figure 5.5 (a) for the SFV dataset using 3D-plot. In these two figures, the z axis is upside down for better visualization, so that the point with the smallest estimation error is shown in the upmost place.

For the survey-based dataset, the best performance can be achieved when $\alpha = 0.5$ and $\gamma = 0.6$. Figure 5.4 (b) and Figure 5.4 (c) further show how the estimation error changes with different $\alpha$ and $\gamma$ in a 2D plane. As can be seen, when $\gamma$ is fixed to 0.6, the estimation error is the smallest as $\alpha$ is 0.5, which indicates that a decaying factor of 0.5 to historical tasks can help improve the performance. When $\alpha = 0.5$, $\gamma = 0.6$ has the lowest estimation error.

For the SFV dataset, the best performance is achieved when $\alpha = 0.1$ and
\[ \gamma = 0.5 \text{ as shown in Figure 5.5.} \] From Figure 5.5 (b), we can see that a higher \( \alpha \) degrades the performance, indicating that including the historical tasks does not help too much for estimating user expertise in the SFV dataset. In the rest of the evaluation, \( \alpha \) and \( \gamma \) are set to the values that achieve the best performance.

### 5.6.4.2 Overall Performance of ETA\(^2\)

In this subsection, we evaluate the performance of ETA\(^2\) by comparing it with existing approaches. First, we keep track of the estimation error in different days. The results for the two datasets are shown in Figure 5.6 (a) and (b), respectively. As shown in the figures, the estimation error of ETA\(^2\) drops overtime in both two datasets. This is because in the beginning, there is no initial knowledge on user expertise, and the tasks are randomly allocated to users. After more tasks have been finished, user expertise is better estimated, which can be used in the following days to make better decisions on task allocation. Also, ETA\(^2\) outperforms other approaches. Specifically, for the survey-based dataset, its estimation error is 15%
Second, we change the average processing capability $\tau$. A small $\tau$ means that users have limited processing capability and less users can be assigned for each task. A large $\tau$ means that users have more time to complete the tasks and therefore more users can be selected for each task. As shown in Figure 5.6 (c) and (d), the estimation error decreases as the average processing capability increases. When the processing capability is limited, ETA$^2$ underperforms TruthFinder in the survey dataset and Hubs and Authority in the SFV dataset. This is because there are very limited users assigned for each task, and then the user expertise cannot be accurately estimated. As the processing capability increases, ETA$^2$ significantly outperforms other approaches.

**Verifying the importance of expertise:** To further verify the importance of the expertise information, we conduct a small experiment to figure out how user expertise affects the data they observe. Specifically, the experiments measure the observation error, i.e., the error of the observed data, under different user expertise. The results are shown in Figure 5.7. As we can see, with the increase of user expertise, there is a clear decrease of the observation error. When the user expertise is larger than 2, most observation errors are close to zero (the red line inside the box indicates the median of observation errors). This result demonstrates that user expertise can be exploited to improve data quality in ETA$^2$. 

![Figure 5.7. The effects of User expertise on observation error in (up) survey-based dataset and (down) SFV dataset](image-url)
In this subsection, we evaluate the effectiveness of the three modules of ETA\(^2\).

**Task Expertise Identification:** In ETA\(^2\), the expertise domain of the task is identified by applying a dynamic hierarchical clustering (DHC) method over the task descriptions. We evaluate the effectiveness of DHC by comparing it with another well-known clustering method, *k*-means. The number of clusters is fixed to be \(k\) in *k*-means, and we test different values of \(k\). As shown in Figure 5.8 (a) and 5.8 (b), the estimation error using DHC is much smaller than those using *k*-means. This is because our DHC method is more flexible and can dynamically update the expertise domains as new tasks arrive.

**Expertise-aware Truth Analysis:** In this experiment, tasks are randomly allocated to users. Then, the truth associated with each task is identified by
applying the expertise-aware approach or the reliability-based approach. For the reliability-based approach in comparison, we test all three reliability-based truth analysis approaches listed in Section 5.6.3, but only show the result of the approach that achieves the smallest estimation error.

The results for the two datasets are shown in Figure 5.9 (a) and Figure 5.9 (b), respectively. As we can see, when there are different sensing tasks, the expertise-aware approach consistently outperforms the reliability-based solutions. This is because the expertise-aware truth analysis assigns higher weights to users with higher expertise (as shown in Equation 5.6). Since high-expertise users usually observe more accurate data, the estimated truth is more accurate. For reliability-based truth analysis, even though it can assign higher weight to users that are inferred to have “high reliability”, these users may not actually provide accurate data.

**Expertise-aware Task Allocation:** In addition to truth analysis, we also do experiments to evaluate if the expertise-aware task allocation can effectively allocate tasks to users, compared with the reliability-based solution. In the experiments, we fix the number of users to be assigned for each task, and apply the expertise-aware solution or the reliability-based solution to allocate tasks to users. As shown in Figure 5.10 (a) and Figure 5.10 (b), the expertise-aware solution outperforms the reliability-based solution. This is especially true when there is less number of users to be assigned to each task in the survey-based dataset. The possible reason is that with the expertise-aware task allocation, the tasks are mainly assigned to the users with high expertise, so that data with better quality are always collected.

### 5.7 Related Work

In crowdsourcing, task allocation determines how to select appropriate users to complete tasks. Researchers have designed solutions to address this problem by focusing on different perspectives. Specifically, the authors in [91][92][93] focus on location-dependent mobile crowdsourcing and design task allocation mechanisms considering the movement of users. For example, He et al. [91] consider the budget on user’s traveling distance and formalize an optimization problem to maximize
the platform’s total reward. Sheng et al. [92] and Zhao et al. [93] consider energy-efficient task allocation which recruits users to reduce the energy consumption while achieving the objective on spacial coverage. In addition, Wang et al. [94] consider image sensing with smartphones and propose user selection approach to maximize the photo utility and minimize the resource consumption. Cheung et al. [95] propose a technique that enable mobile users to select tasks in a distributed manner, with the objective to collect time-sensitive and location-dependent information. Researchers in [68][84][85] study how the quality of data source affects the provided information and provide solutions to identify and recruit high-quality data sources.

Truth analysis in crowdsourcing has also received considerable attention recently. Since the related work on truth analysis has been summarized in the related work of Chapter 4, it will not be repeated in this chapter.

5.8 Conclusion

In this chapter, we proposed an ETA$^2$ approach for expertise-aware truth analysis and task allocation in crowdsourcing. Specifically, we first identified the expertise domains of the tasks by designing a novel semantic analysis method to extract the semantic information of tasks and proposing a dynamic hierarchical clustering approach to cluster tasks based on their semantic information. Then, we proposed an expertise-aware truth analysis solution, in which we built an expertise-aware statistical model and applied maximum likelihood estimation to estimate truth
and learn user expertise. Finally, we designed an expertise-aware task allocation technique by formalizing an optimization problem to maximize the probability that tasks are allocated to users with the right expertise while ensuring the work load does not exceed the processing capability at each user. Experimental results based on two real-world datasets demonstrate that ETA$^2$ has much lower estimation error compared with the existing approaches.
Conclusions and Future Work

6.1 Conclusions

In this dissertation, we developed comprehensive solutions for providing efficient and high-quality communications between mobile devices. Specifically, in Chapter 2 and Chapter 3, we proposed solutions to realize efficient data transmission in mobile opportunistic networks. Then, in Chapter 4 and Chapter 5, we proposed techniques for truth analysis to improve the data quality based on the paradigm of crowdsourcing. These techniques are summarized as follows.

In Chapter 2, we proposed a transient community (TC)-based data transmission strategy in mobile opportunistic networks. To be specific, we first proposed a contact-burst-based clustering method (CCM) to detect TCs by exploiting the pairwise contact processes. In CCM, each pairwise contact process is formalized as regular appearances of contact bursts during which most contacts between the pair of nodes appear. Based on such formulation, transient communities are detected by clustering the pairs of nodes with similar contact bursts. We also proposed a distributed CCM method to make the TC detection feasible in mobile opportunistic networks, and demonstrated that it can effectively detect TCs. Finally, the concept of TCs are applied to data transmission in mobile opportunistic networks, where data is transmitted to TCs that have better relaying capability to the destination node. Trace-drive simulations showed that our approach outperforms traditional community-based data transmission approaches.

In Chapter 3, we designed efficient data transmission strategies for mobile op-
portunistic networks with diverse connectivity characteristics, by exploiting the existence of Transient Connected Components (TCCs). We first identified the existence of TCCs and analyzed their properties based on four traces. By treating multi-hop wireless communications inside TCCs as indirect contacts, through theoretical analyses, we showed that the contact opportunities can be significantly increased in all traces. Based on this observation, we designed a TCC-aware data transmission strategy to improve the performance of data transmission. Then, we enhanced the TCC-aware data transmission by selecting an optimal set of nodes in the TCC to avoid overlap in their contacts and maximize the data transmission opportunity with a small number of nodes. Trace-driven simulations showed that our TCC-aware data transmission strategies outperform existing data transmission strategies with much higher delivery ratio and less network overhead.

In Chapter 4, we proposed resource-aware approaches for truth analysis based on crowdsourcing. Different from existing truth analysis approaches, we utilized the communication network (i.e. mobile opportunistic network) to adaptively collect data from some users, so that high-quality data are collected to achieve the required data credibility. Considering the requirement on data credibility and the constraint of network resources, we quantified the tradeoff between the enhanced data credibility and the increased network overhead as two problems: max-credibility which aims to maximize data credibility with some network overhead, and min-overhead which aims to achieve a specified data credibility while minimizing the network overhead. The proposed solutions are evaluated based on trace-driven simulations and a testbed. Both simulation and experimental results demonstrate the effectiveness of the proposed solutions in terms of data credibility and network overhead.

In Chapter 5, we proposed an ETA² approach for expertise-aware truth analysis and task allocation in crowdsourcing. Specifically, we first identified the expertise domains of the crowdsourcing tasks by designing a novel semantic analysis method to extract the semantic information of tasks and proposing a dynamic hierarchical clustering approach to cluster tasks based on their semantic information. Then, we proposed an expertise-aware truth analysis solution, in which we built an expertise-aware statistical model and applied maximum likelihood estimation to estimate truth and learn user expertise. Finally, we designed an expertise-aware
task allocation technique by formalizing an optimization problem to maximize the probability that tasks are allocated to users with the right expertise while ensuring the work load does not exceed the processing capability at each user. Experimental results based on two real-world datasets demonstrate that ETA\(^2\) has much lower estimation error compared with the existing approaches.

### 6.2 Future Directions

There are many interesting topics worth further investigation. Generally, the further work of this dissertation lies in two broad directions.

#### 6.2.1 Mobile Opportunistic Networks

The first direction is about the further development on mobile opportunistic networks, which is summarized as follows.

- **Implementation of Mobile Opportunistic Networks:** Our current work focuses on the development of data transmission algorithms or strategies. Another important area of research centers on deploying and implementing the proposed strategies in data transmission in real mobile opportunistic systems. Current mobile opportunistic systems [37][71][38] mostly record and analyze the contact patterns between mobile users. However, these works lack the mechanisms to practically transmit data among mobile users. Even though Gao *et al.* in [73] have developed a mobile opportunistic system to transmit data based on an Android app, the data they are transmitting are limited to news from CNN.com or NYTimes.com. It is more interesting and useful if the system can be extended to transmit more types of data, like audio and video, or share personal messages between mobile users.

- **Exploration of Online Social Networks:** In current social-aware data transmission strategies, the social network is extracted from collected mobile opportunistic networks by analyzing the contact patterns between mobile users. However, this extracted social network may not be able to accurately characterize the real social relations between users. Therefore, it is more
helpful if we can collect information about the social network through other means, such as online social networks, e.g., Facebook, Twitter, etc. In [96], we collected data from an online event-based social network called Meetup, with the objective to predict human mobility based on the collected data on event attendance. However, this work lacks further application to data transmission in mobile opportunistic networks.

It is very difficult to apply the data collected from the online social networks to data transmission. First, it is challenging to match virtual identities in online social networks to the real persons in mobile opportunistic networks. However, correct identity matching is necessary if the information in online social networks can be exploited and utilized for data transmission. Second, the social information collected from online social networks may be incomplete. For example, some persons may be very inactive in online social networks. If this is true, the social information and social relations associated with these persons are missing, which impairs the performance of social-aware data transmission. Such data deficiencies must also be addressed with effective mechanisms, so that online social information can be effectively applied to data transmission.

- **Load Balancing:** In our current strategies, especially in TCC-aware data transmission strategies, data are prioritized to transmit to the nodes with higher forwarding metrics, i.e., centrality. A possible consequence is that the nodes with high centrality are responsible for transmitting a lot of data, which is unfair for these nodes and is also a waste of resources for the other nodes. When considering the limits on battery power, these nodes may fail very frequently, which will ultimately impair data transmission performance. In the future work, we need to further develop a mechanism for load balancing, ensuring each node receives comparable loads for data transmission, so that the resources at mobile nodes can be fairly and adequately utilized.

### 6.2.2 Crowdsourcing

The second direction is towards crowdsourcing, especially crowdsourcing based on mobile devices, which is summarized as follows.
• **Removing Data Redundancy:**

Crowdsourcing requires data collection from multiple users. However, the data collected from users may have a very large redundancy. This problem becomes even more severe when dependency between users exists [74][75] and such users share or copy data with one another. To address this problem, Wang et al. [94] have proposed solutions to remove redundancy in photo-based crowdsourcing (i.e., removing redundancy between photos with the same target and similar angles). However, considering all of the forms of data collected through crowdsourcing, this solution is not enough. For example, if the data are numerical values and dependency exists between users, it is challenging to identify the redundancy (i.e., which data have been copied). Based on the intuition that friends or acquaintances may share data with each other, a possible solution is studying and exploiting the social relations between users to identify dependencies and remove data redundancies.

• **Design of Incentive Mechanism for User Recruitment:** In crowdsourcing systems, multiple users should be recruited for each crowdsourcing task. To ensure that enough users are recruited, many incentive mechanisms have been developed to encourage user participation while minimizing total expenses. However, none of the existing mechanisms consider the reliability or expertise of users. A more effective incentive mechanism should prioritize recruiting high-reliability or high-expertise users by paying more to such users and paying less to other users. Therefore, possible future work may consider the reliability or expertise of users and design more effective incentive mechanisms for user recruitment.
The Proof of Theorem 1 in Section 3.2.4

Proof. Based on the condition that the distributions of node degree and TCC size follow exponential distributions, we compute the number of direct contacts and TCC contacts.

The number of direct contacts, denoted as $m_d$, can be determined by the network size $N$ and the distribution of node degree with exponential constant $k^*$. Let $p(k)$ denote the probability mass function (PMF) of degree $K$. $p(k)$ can be computed from the CCDF $P(K > k)$ as

$$p(k) = P(K > k - 1) - P(K > k) = e^{-rac{k-1}{k^*}} - e^{-rac{k}{k^*}} = e^{-rac{k-1}{k^*}} (1 - e^{-rac{1}{k^*}}).$$

for $k \geq 1$. Then, $m_d$ can be computed as half of the number of total degrees:

$$m_d = \frac{1}{2} \sum_{k=1}^{\infty} (N \cdot p(k) \cdot k) = \frac{N}{2(1 - e^{-\frac{1}{k^*}})}. \quad (1)$$

We next compute the number of TCC-contacts, $m_t$, with the network size $N$ and the distribution of TCC size which has exponential constant $s^*$. The PMF $p(s)$ of TCC size $S$ can be calculated from CCDF $P(S > s)$ as

$$p(s) = P(K > s - 1) - P(K > s) = \begin{cases} e^{-\frac{s-1}{s^*}} (1 - e^{-\frac{1}{s^*}}) & s \geq 3 \\ 1 - e^{-\frac{1}{s^*}} & s = 2. \end{cases}$$
We assume the total number of TCCs inside the network is $N_{TCC}$. With the distribution of TCC sizes $p(s)$, the number of TCC with size $s$ is $N_{TCC} \cdot p(s)$. The number of TCC-contacts inside a TCC with size $s$ is $\binom{s}{2} = \frac{s(s-1)}{2}$. The total number of TCC-contacts inside the network is:

$$m_t = \sum_{s=2}^{\infty} (N_{TCC} \cdot p(s) \cdot \frac{s(s-1)}{2})$$

$$= N_{TCC} \cdot \frac{E(S^2) - E(S)}{2}$$

(2)

The value of $N_{TCC}$ can be determined by the network size $N$. With the distribution of TCC size $p(s)$, the number of TCCs with size $s$ is $N_{TCC} \cdot p(s)$. Since a node belongs to one TCC, the summation of all TCC sizes is the network size $N$. Thus, we have

$$N = \sum_{s=2}^{\infty} (s \cdot N_{TCC} \cdot p(s)) = N_{TCC} \cdot E(S).$$

Therefore, $N_{TCC} = \frac{N}{E(S)}$. With

$$E(S) = 2(1 - e^{-\frac{2}{s^*}}) + \sum_{s=3}^{\infty} (s \cdot e^{-\frac{s-1}{s^*}} (1 - e^{-\frac{1}{s^*}}))$$

$$= \frac{1}{1 - e^{-\frac{1}{s^*}}} + 1 - e^{-\frac{1}{s^*}},$$

$$E(S^2) = 4(1 - e^{-\frac{2}{s^*}}) + \sum_{s=3}^{\infty} (s^2 \cdot e^{-\frac{s-1}{s^*}} (1 - e^{-\frac{1}{s^*}}))$$

$$= \frac{1 + e^{-\frac{1}{s^*}}}{(1 - e^{-\frac{1}{s^*}})^2} + 3 - 3e^{-\frac{1}{s^*}},$$

Equation (2) becomes

$$m_t = \frac{N \cdot E(S^2) - E(S)}{2} = N \cdot \frac{E(S^2) - E(S)}{2E(S)}$$

$$= N \cdot \frac{e^{-\frac{1}{s^*}} + (1 - e^{-\frac{1}{s^*}})^3}{1 - e^{-\frac{1}{s^*}} + (1 - e^{-\frac{1}{s^*}})^3}. $$

(3)
The ratio of TCC-contacts to direct contacts is

\[ \frac{m_t}{m_d} = 2(1 - e^{-\frac{1}{k^*}}) \cdot \frac{e^{-\frac{1}{s^*}} + (1 - e^{-\frac{1}{s^*}})^3}{1 - e^{-\frac{1}{s^*}} + (1 - e^{-\frac{1}{s^*}})^3}. \]  

(4)

which is determined only by the two exponential constants \( k^* \) and \( s^* \), independent of the network size \( N \). With \( \frac{m_t}{m_d} > 1 \), we have

\[ k^* < \frac{1}{\log \left( \frac{2(e^{-\frac{1}{s^*}} + (1 - e^{-\frac{1}{s^*}})^3)}{3e^{-\frac{1}{s^*}} - 1 + (1 - e^{-\frac{1}{s^*}})^3} \right)}. \]  

(5)

If \( \hat{k}^* = 1/\log \left( \frac{2(e^{-\frac{1}{s^*}} + (1 - e^{-\frac{1}{s^*}})^3)}{3e^{-\frac{1}{s^*}} - 1 + (1 - e^{-\frac{1}{s^*}})^3} \right) \), as long as \( k^* < \hat{k}^* \), the number of TCC-contacts is more than the direct contacts, which increases the contact opportunities. \( \square \)
Bibliography


Vita
Xiaomei Zhang

Xiaomei Zhang is a Ph.D Student at the Department of Computer Science and Engineering in the Pennsylvania State University. She received her B.E degree from the University of Science and Technology of China in Electronic Engineering and Information Science. Her research interest is in the area of wireless communication, mobile computing and network science. She is a student member of IEEE.

Publications during the Ph.D. study:

- Xiaomei Zhang, Jing Zhao, and Guohong Cao, “Who Will Attend?–Predicting Event Attendance in Event-Based Social Network,” IEEE International Conference on Mobile Data Management (MDM), 2015.


- Xiaomei Zhang and Guohong Cao, “Transient Community Detection and Its Application to Data Forwarding in Delay Tolerant Networks”, IEEE International Conference on Network Protocols (ICNP), 2013.