IMPROVING CAPACITY AND ROBUSTNESS OF WIRELESS NETWORKS

A Dissertation in
Computer Science and Engineering
by
Li Qiu

© 2018 Li Qiu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2018
The dissertation of Li Qiu was reviewed and approved* by the following:

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

Sencun Zhu  
Associate Professor of Computer Science and Engineering  
Associate Professor of Information Sciences and Technology

Zhibiao Zhao  
Associate Professor of Statistics

Bhuvan Urgaonkar  
Associate Professor of Computer Science and Engineering

Chita R. Das  
Distinguished Professor of Computer Science and Engineering  
Department Head of Computer Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

Wireless ad hoc networks enable communications among mobile nodes without any infrastructure support, as nodes themselves relay and forward packets for each other. Due to wireless interference and network dynamicity, they suffer from low network capacity and unreliable network connections. Specifically, due to the interference between concurrent transmissions, the per-node capacity decreases with the increasing number of nodes. Moreover, it is difficult to maintain stable end-to-end connections, especially between faraway nodes, since the communication links may experience high dynamicity. To address these challenges, we propose resource management strategies to improve the network capacity and the link robustness. To improve the network capacity, we design caching techniques, where nodes retrieve desired contents from close neighbors rather than the faraway server, to reduce the transmission distance and improve the network capacity. To maintain reliable communication between important social pairs, we devote the network resources to these important social links, rather than focus on improving the performance of data forwarding for all nodes in the network. We propose two strategies, amplify-and-forward and direct link placement, to connect faraway nodes and improve the data forwarding performance.

The goal of this dissertation is to design efficient resource management strategies such as content caching, transmission scheduling and link placement to improve the network capacity and maintain important social links. First, we utilize caching techniques to improve the network capacity. When the content popularity follows a uniform distribution, we study the capacity of wireless networks with caching considering the cache size of each node, the total size of unique content in the network, and the number of nodes in the network. We present an upper bound on network capacity, and present an achievable capacity lower bound. Our results suggest that the capacity of wireless ad hoc networks with caching can remain constant even as the number of nodes in the network increases. Second, when the contents
have skewed popularity, we evaluate how the distribution of the content popularity affects the per-node capacity, and design popularity-aware caching strategies that maximizes the per-node capacity for wireless ad hoc networks. Based on the popularity-aware caching strategies, we derive different capacity scaling laws based on the skewness of the content popularity. Our results suggest that for wireless networks with caching, when contents have skewed popularity, increasing the number of nodes monotonically increases the per-node capacity. Third, to maintain the communication links between important social pairs, we adopt a cooperative amplify-and-forward strategy, where nodes (relays) cooperate to improve the signal strength at the destination. We formulate and study two optimization problems for maintaining the required link throughput: Min-Energy and Min-Relay, where the goal of Min-Energy is to minimize the power consumption of the relays, and the goal of Min-Relay is to minimize the number of active relays. Evaluation results show that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption. Forth, to improve the robustness of the social links, we propose to proactively place some reliable communication links (e.g., satellite links) to the given network to improve the social link quality. We formulate and study the problem of maintaining social links through direct link placement (called MSL). Due to the complexity and the NP-hardness of the MSL problem, we propose solutions to bound it by two submodular functions, and derive an approximation algorithm with provable approximation ratio based on the sandwich approximation strategy. Evaluation results based on both synthetic graphs and a real-world social network dataset demonstrate the effectiveness of our proposed algorithms.
# Table of Contents

List of Figures ix  
List of Tables x  
Acknowledgments xi  

Chapter 1  
Introduction 1  
1.1 Focus of This Dissertation 2  
1.1.1 Capacity of Wireless Networks with Caching 3  
1.1.2 Popularity-Aware Caching in Wireless Networks 4  
1.1.3 Maintaining Social Links through Amplify-and-Forward 4  
1.1.4 Maintaining Social Links through Direct Link Placement 5  
1.2 Organization 6  

Chapter 2  
Capacity of Wireless Networks with Caching 7  
2.1 Introduction 7  
2.2 Related Work 8  
2.3 Model 9  
2.3.1 Network Model 9  
2.3.2 Content Access Model 10  
2.3.3 Capacity and Request Satisfaction Rate 11  
2.4 An Upper Bound on Network Capacity 11  
2.4.1 Capacity and Transmission Distance 12  
2.4.2 Minimum Expected Transmission Distance 13  
2.4.3 Capacity Upper Bound 16
2.5 Achievable Capacity Lower Bound ............................................ 19
  2.5.1 Voronoi Tessellation ...................................................... 19
  2.5.2 Finding cells that can cache all content .............................. 20
  2.5.3 Transmission Scheduling ............................................... 23
  2.5.4 Achievable Lower Bound ............................................... 25
2.6 Numerical Results ............................................................... 27
  2.6.1 Approximation Validation and Numerical Results .................. 27
  2.6.2 Comparisons to Existing Work ......................................... 27
  2.6.3 Discussions .................................................................. 30
2.7 Conclusion .......................................................................... 30

Chapter 3

Popularity-Aware Caching in Wireless Networks 31
3.1 Introduction ........................................................................... 31
3.2 Related Work ......................................................................... 33
3.3 Preliminaries ......................................................................... 35
  3.3.1 Network Model ............................................................... 35
  3.3.2 Content Access and Zipf Distribution ............................... 35
  3.3.3 Capacity ....................................................................... 36
3.4 Problem Formulation .............................................................. 38
3.5 Capacity Upper Bound ............................................................ 40
  3.5.1 Optimal Content Densities .............................................. 40
  3.5.2 Minimum Transmission Distance ................................. 42
    3.5.2.1 $\gamma < \frac{3}{2}$ ...................................................... 42
    3.5.2.2 $\gamma = \frac{3}{2}$ ..................................................... 45
    3.5.2.3 $\gamma > \frac{3}{2}$ ..................................................... 46
  3.5.3 Capacity Scaling Laws ..................................................... 50
3.6 Analytical Results ................................................................. 52
  3.6.1 Capacity and the Number of Nodes .................................. 52
  3.6.2 Effects of Local Cache ................................................... 55
  3.6.3 Influence of Other Parameters ....................................... 57
3.7 Caching Algorithms ............................................................... 58
  3.7.1 Algorithm Description .................................................... 59
  3.7.2 Theoretical Results ....................................................... 60
3.8 Evaluations ........................................................................... 63
  3.8.1 Effects of Various Parameters ....................................... 63
  3.8.2 Comparisons to Existing Work ..................................... 66
  3.8.3 Evaluation of the Distributed Caching Algorithm .......... 67
3.9 Conclusion ........................................................................... 70
### Chapter 4
**Maintaining Social Links through Amplify-and-Forward**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>Preliminaries</td>
<td>73</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Model</td>
<td>73</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Link Throughput</td>
<td>75</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Problem Formulation</td>
<td>75</td>
</tr>
<tr>
<td>4.3</td>
<td>The Min-Energy Problem</td>
<td>77</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Important Properties</td>
<td>77</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Semidefinite Relaxation</td>
<td>79</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Optimal Solution</td>
<td>81</td>
</tr>
<tr>
<td>4.4</td>
<td>The Min-Relay Problem</td>
<td>83</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Relay Selection</td>
<td>83</td>
</tr>
<tr>
<td>4.4.2</td>
<td>The Min-Relay Algorithm</td>
<td>87</td>
</tr>
<tr>
<td>4.5</td>
<td>Performance Evaluations</td>
<td>89</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Simulation setup</td>
<td>89</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Power Consumption</td>
<td>91</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Comparing Min-Energy and Min-Relay</td>
<td>93</td>
</tr>
<tr>
<td>4.6</td>
<td>Related Work</td>
<td>95</td>
</tr>
<tr>
<td>4.7</td>
<td>Conclusion</td>
<td>96</td>
</tr>
</tbody>
</table>

### Chapter 5
**Maintaining Social Links through Direct Link Placement**

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>Problem Formulation</td>
<td>99</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Network Model</td>
<td>99</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Important Social Links</td>
<td>99</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Problem Formulation</td>
<td>100</td>
</tr>
<tr>
<td>5.3</td>
<td>MSL with a Common Node</td>
<td>101</td>
</tr>
<tr>
<td>5.3.1</td>
<td>NP-Hardness</td>
<td>101</td>
</tr>
<tr>
<td>5.3.2</td>
<td>A Solution to the MSL-CN Problem</td>
<td>103</td>
</tr>
<tr>
<td>5.4</td>
<td>Maintaining Social Links (MSL)</td>
<td>104</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Submodularity of MSL</td>
<td>104</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Sandwich Approximation Strategy</td>
<td>104</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Lower Bound Function</td>
<td>106</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Upper Bound Function</td>
<td>106</td>
</tr>
<tr>
<td>5.5</td>
<td>Performance Evaluations</td>
<td>107</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Evaluation Setup</td>
<td>107</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Theoretical Approximation Ratio</td>
<td>108</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Comparing with the Randomized Selection Solution</td>
<td>110</td>
</tr>
</tbody>
</table>
Chapter 6
Conclusions and Future Work

6.1 Summary

6.2 Future Directions

Bibliography
List of Figures

2.1 Approximation validation for $E(Y^i_{\xi})$ and $E(X^i_{\xi})$ (i.e., formula (2.7) and formula (2.10)) . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.2 Capacity and request satisfaction rate . . . . . . . . . . . . . . . . . 28
2.3 Comparisons to existing work . . . . . . . . . . . . . . . . . . . . . . 29

3.1 Effects of various parameters on capacity . . . . . . . . . . . . . . . 64
3.2 Comparisons to existing work . . . . . . . . . . . . . . . . . . . . . . 65
3.3 Content densities vs. content popularity . . . . . . . . . . . . . . . . . 66
3.4 Results of the Distributed Caching Algorithm vs. theoretical value . 68

4.1 Selection order of the relays . . . . . . . . . . . . . . . . . . . . . . . 85
4.2 The power consumption of Min-Energy . . . . . . . . . . . . . . . . . . 91
4.3 The power consumption of Min-Relay . . . . . . . . . . . . . . . . . . 91
4.4 Relay Selection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 93
4.5 Min-Relay vs. Min-Energy ($n = 30$) . . . . . . . . . . . . . . . . . . . 94

5.1 Shortcut Edge Placement . . . . . . . . . . . . . . . . . . . . . . . . . . 110
5.2 Proposed Solution vs. Randomized Selection Solution . . . . . . . . 112
List of Tables

3.1 Notation ................................................. 37

5.1 $\frac{\sigma(F_{\nu})}{\nu(F_{\nu})}$ for Random Geometric graph ................ 109

5.2 $\frac{\sigma(F_{\nu})}{\nu(F_{\nu})}$ for Gowalla Dataset ......................... 109

5.3 Important Social Pairs .................................. 111
Acknowledgments

First of all, I would like to express my sincere gratitude to my advisor Prof. Guohong Cao, for his patient guidance, valuable suggestions and warm encouragement during my Ph.D. study. I am especially lucky to have an advisor with such multidisciplinary expertise, who can provide valuable insights in any areas I am working in. I also have also benefited a lot from his visionary thinking and rigorous attitude towards research. Without his guidance and support, I could not have completed my Ph.D. study.

Besides my adviser, I would also like to thank the rest of my doctoral committee: Prof. Bhuvan Urgaonkar, Prof. Sencun Zhu, and Prof. Zhibiao Zhao. It is my great privilege to have these wonderful professors in my doctoral committee. Without their generous help and insightful comments, the dissertation could not have reached the present form.

My sincere thanks also goes to my co-authors, Dr. Liang Ma from IBM research, and Dr. Jing Zhao in MCN Lab, for their inspiring thoughts, sharp opinions and valuable comments on our works. I would also like to thank all my fellow labmates, especially Yibo Wu and Yi Yang, for their sincere friendship and genuine help during the past six years. They have made my study and life at Penn State an enjoyable and memorable journey.

Last but not least, I am deeply indebted to my parents for their unconditional love. Their meticulous care and strong support have helped me through all the difficulties in my life. This dissertation is simply impossible without their love and support. For that, I dedicate this dissertation to them.

This work was supported in part by the National Science Foundation (NSF) under grants CNS-1320278, CNS-1421578 and CNS-1526425, and by Network Science CTA under grant W911NF-09-2-0053. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or the Network Science CTA.
Dedication

To my beloved parents.
Chapter 1

Introduction

Wireless ad hoc networks enable communications among mobile nodes without any infrastructure support, as nodes themselves relay and forward packets for each other. Due to wireless interference and network dynamicity, they suffer from low network capacity and unreliable network connections.

Specifically, due to the interference between concurrent transmissions, the per-node capacity decreases with the increasing number of nodes. This was first found by Gupta and Kumar \[1\]. They showed that for a wireless ad hoc network with \( n \) nodes, each node can at most transmit at a rate of \( \Theta(\frac{1}{\sqrt{n}}) \) to its destination, even with optimal scheduling of transmissions from various nodes.

Caching can be used to improve network capacity. With caching, contents are stored close to the users, which shortens the transmission distance and improves the network capacity. However, existing studies on caching in wireless networks are limited to protocol design, and the associated performance evaluations are based on simulations. The fundamental performance limits of caching in wireless ad hoc networks have rarely been studied in an analytical manner, and it is unclear how contents with various popularity should be cached so that the per-node capacity is maximized.

Moreover, it is difficult to maintain stable end-to-end connections in wireless ad hoc networks, especially between faraway nodes. Although social cognitive techniques can be exploited to enhance the robustness and performance of mobile ad hoc networks, existing research \[2, 3, 4, 5\] focuses on improving the performance of data forwarding (or routing) for all nodes in the network, without differentiating
which node pairs are more important. In many real scenarios, the communications between some nodes are more important than others. For example, in a battle field, for a platoon of soldiers consisting of three squads, the commander of the platoon should maintain good connections with the squad leaders, but not necessarily with all other soldiers. Then, it is more important to maintain the social links between the commander and the squad leaders, even though this may be at the cost of sacrificing some communication performance with other nodes.

There are many research challenges on maintaining important social links in wireless ad hoc networks, especially when the nodes are far away from each other. Although multi-hop transmission can be used, if two nodes in the routing path are out of the transmission range, a network partition is possible. Increasing the transmission power level can solve part of the problem, but this approach has its limitations since there is always a maximum transmission range, out of which two nodes will not be able to communicate. Thus, even with the maximum transmission power level, it is possible that the social links cannot be maintained because two nodes along the routing path are far away from each other.

To address these challenges, we propose resource management strategies to improve the network capacity and maintain important social links. To improve the network capacity, we design caching techniques, where nodes retrieve desired contents from close neighbors rather than the faraway server, to reduce the transmission distance and improve the network capacity. To maintain reliable communication between important social pairs, we devote the network resources to these important social links, rather than focus on improving the performance of data forwarding for all nodes in the network. We propose two strategies, amplify-and-forward and direct link placement, to connect faraway nodes and improve the data forwarding performance.

1.1 Focus of This Dissertation

The goal of this dissertation is to design efficient resource management strategies such as content caching, transmission scheduling and link placement to improve the network capacity and maintain important social links. Specifically, we focus on four aspects, i.e., deriving capacity scaling laws for content caching in wireless ad hoc
networks, designing popularity-aware caching strategies to maximize the network capacity, maintaining social links through amplify-and-forward, and maintaining social links through direct link placement. We briefly explain them in the following four subsections.

### 1.1.1 Capacity of Wireless Networks with Caching

Over the past few years, caching in wireless ad hoc networks has attracted lots of attention. By caching contents at multiple nodes, network traffic and content access delay can be reduced as users can access contents from near neighbors instead of the faraway server. Although various caching techniques [6, 7, 8, 9] have been proposed to improve the performance of wireless ad hoc networks, these studies are limited to protocol design and the associated performance evaluations are based on simulations. The fundamental performance limits of caching in wireless ad hoc networks have rarely been studied in an analytical manner.

We study the capacity of wireless networks with caching considering the cache size of each node \( s \), the total size of unique content in the network \( m \), and the number of nodes \( n \) in the network. As various caching schemes may lead to totally different performance, we first investigate how the employed caching scheme will affect the network capacity. Based on the effect of the caching schemes, we derive a capacity upper bound for caching in wireless ad hoc networks. However, this upper bound only specifies the maximum capacity that the network could possibly support, instead of the capacity that is actually achievable. To derive the achievable capacity, we also design a caching scheme and show that by carefully caching distinct contents at specific nodes, a capacity of \( \Theta \left( W \frac{s}{m} \right) \) can always be achievable.

Our analytical results suggest that for wireless ad hoc networks with caching, the network capacity can remain constant even if the number of nodes increases. This is in sharp contrast to previous results on capacity of wireless ad hoc networks without caching, in which the network capacity decreases quickly as the number of nodes increases [1].
1.1.2 Popularity-Aware Caching in Wireless Networks

Our previous study focuses on designing caching strategy and deriving capacity scaling laws when content popularity follows a uniform distribution. In practice, some contents are accessed much more frequently than others, which requires a more complicated caching strategy to maximize the per-node capacity, and may lead to totally different capacity scaling laws.

To address this problem, given the content popularity, we first study the optimal caching strategy, i.e., how the contents with various popularity should be cached so that the per-node capacity is maximized. Based on the optimal caching strategy, we then evaluate the effect of content popularity on per-node capacity, and derive different capacity scaling laws for networks with different content popularity skewness. For all the different capacity scaling laws, we analytically investigate how the per-node capacity is affected by various parameters, including the number of nodes ($n$), the cache size ($s$), and the number of unique contents ($m$). Basically, as the distribution of the content popularity changes from uniform distribution to more skewed distributions, the per-node capacity increases from $\Theta(\sqrt{\frac{s}{m}})$ to roughly $\Theta(\sqrt{s})$. Furthermore, we propose a distributed caching algorithm which enables nodes to optimally cache contents and maximize the per-node capacity only based on local knowledge.

1.1.3 Maintaining Social Links through Amplify-and-Forward

In mobile ad hoc networks, the communication between some nodes are more important than others, and the network resources should be devoted to these important social links, rather than focus on improving the performance of data forwarding for all nodes in the network. To maintain the important social links, we adopt a cooperative amplify-and-forward strategy, where nodes around the node that cannot reach the next hop node (or link source for simplicity) cooperate to transmit towards the next node in the routing path (or link destination for simplicity). More specifically, the source first broadcasts the data to its nearby nodes, which simultaneously amplifies and forwards their received signal to the destination. In this way, the destination can receive a much stronger signal, from which it can decode and obtain the data from the source, and accordingly the
social links can be maintained.

We formulate and study two optimization problems for maintaining the required link throughput: Min-Energy and Min-Relay, where the goal of Min-Energy is to minimize the power consumption of the relays, and the goal of Min-Relay is to minimize the number of active relays. Since the Min-Energy problem is a non-convex problem, we solve it based on an approximation technique and prove that our solution is a feasible, in fact optimal solution. We formulate the Min-Relay problem as an integer programming problem, and propose a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. Evaluation results show that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.

1.1.4 Maintaining Social Links through Direct Link Placement

In some cases, even when the network resources are devoted to the important social links, it is still possible that these social links are unable to achieve the desired quality of service (QoS). To address this problem, in addition to exploiting the existing network resources, we propose to proactively place some reliable communication links (e.g., satellite links) to the given network, to improve the data forwarding performance. These reliable communication links are referred to as shortcut edges.

We focus on simultaneously maintaining multiple social links (referred to as the MSL problem), through direct placement of shortcut edges. Since shortcut edges (e.g., satellite links) are valuable and expensive resources, we constrain ourselves to the case where the number of shortcut edges is limited. Thus, the real challenge of the problem lies in the effective placement of the shortcut edges, such that each shortcut edge can benefit more social links. Due to the complexity and the NP-hardness of the MSL problem, we first study a special case of MSL, where all important social links share a common node (called MSL-CN). We show that MSL-CN can be formulated in terms of a submodular function, for which we present a greedy algorithm with guaranteed high approximation ratio. Then, for the general
MSL problem, we propose solutions to bound it by two submodular functions, and derive an approximation algorithm with provable approximation ratio based on the sandwich approximation strategy. Evaluation results based on both synthetic graphs and a real-world social network dataset demonstrate the effectiveness of our proposed algorithms.

1.2 Organization

The remainder of the dissertation is organized as follows. Chapter 2 presents the caching strategy and capacity upper/lower bounds when the content popularity follows a uniform distribution. Chapter 3 studies the caching strategy for contents with skewed popularity, and derives capacity scaling laws based on the skewness of content popularity. Chapter 4 discusses the problem of maintaining social links through amplify-and-forward, and proposes optimal solutions for both Min-Energy and Min-Relay problems. Chapter 5 presents algorithms on maintaining social links through direct link placement. Finally, we conclude the dissertation and discuss the future work in Chapter 6.
Chapter 2

Capacity of Wireless Networks with Caching

2.1 Introduction

Over the past few years, caching in wireless ad hoc networks has attracted lots of attention. By caching contents at multiple nodes, network traffic and content access delay can be reduced as users can access contents from near neighbors instead of the faraway server. Although various caching techniques [6, 7, 8, 9] have been proposed to improve the performance of wireless ad hoc networks, these studies are limited to protocol design and the associated performance evaluations are based on simulations. The fundamental performance limits of caching in wireless ad hoc networks have rarely been studied in an analytical manner. Although there are some analytic performance studies on caching in Internet such as modeling the cache hit ratio [10, 11, 12], modeling the steady-states of cache networks [13], performance analysis of optimal routing and content caching [14], etc., these results cannot be directly applied to wireless ad hoc networks.

There are some well known analytical results related to the capacity of wireless ad hoc networks, which is constrained by the mutual interference of concurrent transmissions between nodes. Gupta and Kumar [1] have proved that each node can transmit at most $\Theta(\frac{W}{\sqrt{n}})$ bits per second, where $n$ is the number of nodes and $W$ is the channel throughput. That is, the network capacity decreases as the
number of nodes increases. Later, Grossglauser and Tse [15] show that the per-user throughput can increase dramatically when nodes are mobile rather than static. It is possible for network capacity to remain constant even if the number of nodes increases, but at the cost of long data transmission delay. However, cache is not considered in their studies, and it is not clear how cache affects the capacity of wireless ad hoc networks.

In this chapter, we study the capacity of wireless ad hoc networks with caching with respect to the cache size of each node \((s)\), the total size of unique content in the network \((m)\), and the number of nodes \((n)\) in the network. As various caching schemes may lead to totally different performance, we first investigate how the employed caching scheme will affect the network capacity. Based on the effect of the caching schemes, we derive a capacity upper bound for caching in wireless ad hoc networks. However, this upper bound only specifies the maximum capacity that the network could possibly support, instead of the capacity that is actually achievable. To address this problem, we also design a caching scheme and show that by carefully caching distinct contents at specific nodes, a capacity of \(\Theta \left(W \frac{s}{m}\right)\) can always be achievable. Most importantly, our analytical results suggest that for wireless ad hoc networks with caching, the network capacity can remain constant even if the number of nodes increases. This is in sharp contrast to previous results on capacity of wireless ad hoc networks without caching, in which the network capacity decreases quickly as the number of nodes increases [1].

The rest of the chapter is organized as follows. We review the related work in Section 2.2. We give the models and definitions in Section 2.3. In Section 2.4, we derive a capacity upper bound for wireless networks with caching. We construct an achievable capacity lower bound in Section 2.5. In Section 2.6, we present the numerical results and discuss their implications. Section 2.7 concludes the chapter.

### 2.2 Related Work

Caching in wireless networks has been studied from various aspects. Yin et al. [6, 7] have examined several cooperative caching schemes to determine where to cache the data. Jin and Wang [16] have proposed solutions to determine the optimal placement of replicas in the network. Fiore et al. [8] have proposed techniques to
determine whether a node should cache the data to reduce data redundancy among neighbors. However, none of them have studied the fundamental performance limits of caching in wireless networks in an analytical manner.

Niesen et al. [17] studied the content delivery problem in wireless networks from an information-theoretical point of view. They focused on deriving the region of feasible request serving rate by knowing where and what data has been cached, while we focus on the scaling laws of network capacity based on the cache size and the number of nodes in the network.

In [18], Gitzenis et al. studied the asymptotic laws for joint replication and delivery in wireless networks. They derived the minimum throughput on each link so that every node is able to satisfy one request per second. However, in practice, the transmissions on various links may interfere with each other. Thus it is unknown whether their throughput can be supported by the network, and it is unclear what is the network capacity actually achievable. Furthermore, we present more interesting results of caching which show that the network capacity can remain constant even if the number of nodes increases.

Azimdoost et al. [19] studied the capacity of wireless networks with caching when each content has a limited lifetime. They presented a capacity upper bound for both grid and random networks. In [20], the authors derived capacity upper bounds for two specific content access schemes considering the number of nodes $n$ and the cache size $s$. However, in both [19] and [20], the authors only consider the capacity upper bound, but fail to investigate what capacity is actually achievable. In addition, both assume that the wireless transmission range is in the order of $\sqrt{\frac{\log n}{n}}$, which will significantly restrict the network capacity.

2.3 Model

2.3.1 Network Model

We consider a wireless ad hoc network where $n$ nodes are independently and uniformly distributed on the surface of a unit sphere. Similar to [1], we analyze the capacity of wireless networks with caching on the surface of the sphere $S^2$ rather than on a disk so as to eliminate the edge effects; i.e., nodes near the edge have
much fewer neighbors than nodes near the center.

Assume the nodes are homogeneous, i.e., each node can cache $s$ bits of contents, and all transmissions employ the same amount of power $P$. All nodes transmit on a common wireless channel which can support $W$ bits per second. Let $\tau(t)$ denote the set of nodes simultaneously transmitting at time $t$. Suppose a node $i \in \tau(t)$ sends data to node $j$, according to [1], the transmission rate can reach $W$ bits per second if:

$$\frac{P}{X_{i,j}^\alpha} \geq \beta,$$

where $\beta$ is the minimum SIR for successful reception, $X_{i,j}$ is the great-circle distance (i.e., the shortest distance between two points on the surface of a sphere) between $i$ and $j$, $N_0$ is the white noise and $\alpha$ is a parameter larger than 2 describing how the signal strength scales with distance.

### 2.3.2 Content Access Model

Let $\Phi = \{\rho_i\}_{1 \leq i \leq m}$ denote the set of $m$ unique content throughout the network. To simplify the analysis, we assume each content has one bit. Note that our capacity analysis results can be applied to cases where the content has various sizes, as our analysis only depends on the total content size rather than the size of individual content. These contents are cached throughout the network where each node $i$ caches a subset $\phi_i$ of $\Phi$ locally. The cache size constraint requires $|\phi_i| \leq s$.

When each content has one bit, $m$ contents have $m$ bits. Apparently, when $s \geq m$, the problem is trivial since each node can cache all $m$ contents and then all content requests can be satisfied by local cache. On the other hand, to guarantee that at least one copy of each content exists in the network, the total cache size should be larger than $m$, i.e., $ns \geq m$. Thus, we assume $\frac{m}{n} \leq s < m$.

At each node, there is always a content request, and a new request will arrive after the previous request has been served. The content requests follow uniform distribution. If the requested content has been cached locally, it will be directly served. Otherwise, the node receiving the request has to contact other nodes for the content either directly or through multiple hops.
2.3.3 Capacity and Request Satisfaction Rate

Similar to [1, 15], network capacity describes node’s capability to transmit or retrieve contents. Under our content access model, the capacity of wireless networks \( \lambda(s, m, n) \) is the number of bits that nodes can receive from others per second.

In addition to capacity, we also analyze the request satisfaction rate \( \mu(s, m, n) \), which is the number of requests that can be satisfied per second under our content access model. As the request satisfaction rate depends on both local cache and network capacity, it provides a more complete description of how the cache size may affect the caching performance in wireless networks.

The achievable capacity and request satisfaction rate may depend on the location of nodes. As the nodes are uniformly distributed, there is always a small probability for them to be extremely oddly located, and then the capacity and the request satisfaction rate may be largely deviated. Therefore, when defining the upper and lower bounds, we allow for vanishingly small probability that the actual value deviates from the upper or lower bounds.

**Definition 1** (Upper Bound). The capacity (request satisfaction rate) is upper bounded by \( \hat{\lambda}(s, m, n) \) (\( \hat{\mu}(s, m, n) \)), if there exists a constant \( c \) (\( c' \)) such that for any caching scheme and transmission schedule

\[
\lim_{n \to \infty} \Pr \left( \lambda(s, m, n) \leq c\hat{\lambda}(s, m, n) \right) = 1
\]

\[
\lim_{n \to \infty} \Pr \left( \mu(s, m, n) \leq c'\hat{\mu}(s, m, n) \right) = 1.
\]  

(2.2)

**Definition 2** (Achievable Lower Bound). A capacity (request satisfaction rate) of \( \tilde{\lambda}(s, m, n) \) (\( \tilde{\mu}(s, m, n) \)) is achievable, if there exists a constant \( c \) (\( c' \)), and a caching scheme and transmission schedule, such that

\[
\lim_{n \to \infty} \Pr \left( \lambda(s, m, n) \geq c\tilde{\lambda}(s, m, n) \right) = 1
\]

\[
\lim_{n \to \infty} \Pr \left( \mu(s, m, n) \geq c'\tilde{\mu}(s, m, n) \right) = 1.
\]  

(2.3)

2.4 An Upper Bound on Network Capacity

In this section, we present an upper bound on the capacity of wireless networks with caching. We first show that nodes prefer the contents cached at closest nodes
to achieve a higher capacity. Then we derive the network capacity for an ideal scenario where each node has contents cached as close as possible, which will be the upper bound.

2.4.1 Capacity and Transmission Distance

For a node $i$, after receiving requests for content $\rho_j$ not cached locally, it has to retrieve $\rho_j$ from the node that caches the content. Under different caching schemes, $i$ may contact nearby nodes or faraway nodes to retrieve $\rho_j$, and then lead to different performance. To derive a capacity upper bound for all possible schemes, we first discuss what may affect the capacity.

In a wireless network with $n$ nodes, where each node transmits to an arbitrarily chosen destination, Gupta and Kumar [1] prove that the per node capacity is upper bounded by $\frac{W}{L\sqrt{n}}$, where $L$ is the average distance between the source and destination pairs. This result indicates that as nodes communicate with more distant nodes, the capacity decreases. A similar result has also been found in [21, 15], where the authors suggest that the capacity can be improved by shortening the transmission distance between the source node and the destination node. This is because as the transmission distance increases, more intermediate nodes will be affected and then less number of simultaneous transmissions are allowed.

Based on the above results, each node prefers the other $m - s$ contents ($s$ contents can be cached locally) to be cached at nodes as close as possible to achieve higher capacity. Thus, to find an upper bound, we assume each node $i$ caches $s$ contents locally, and the closest $\left\lceil \frac{m-s}{s} \right\rceil$ nodes cache the remaining $m - s$ contents. Let $\{X_{(1)}^i, \ldots, X_{(n-1)}^i\}$ denote the ordered distance between $i$ and any other node, i.e., $X_{(j)}^i$ is the distance between $i$ and its $j$-th closest neighbor. For simplicity, we also use $X_{(j)}^i$ to represent the $j$-th closest neighbor of $i$. Then $i$ and $\{X_{(1)}^i, \ldots, X_{\left\lceil \frac{m}{2} - 2 \right\rceil}^i\}$ will each cache $s$ distinct contents, and $X_{\left\lceil \frac{m}{2} - 1 \right\rceil}^i$ will cache the remaining $m - s \left\lceil \frac{m}{s} - 1 \right\rceil$ contents. Although such an ideal scenario where every node has the remaining contents cached at its closest neighbors may not always be practical, there may be special cases when such requirement can be satisfied, and thus this ideal scenario can be used for calculating the capacity upper bound.
2.4.2 Minimum Expected Transmission Distance

In this subsection, we focus on deriving the average data transmission distance in the aforementioned ideal scenario. As all nodes are uniformly and independently distributed on the surface of the sphere with radius 1, given location of node $i$, the distance between $i$ and any other node $k$ follows the distribution shown below:

$$f(x) = \frac{\sin x}{2} (0 \leq x \leq \pi),$$

$$F(x) = \frac{1 - \cos x}{2} (0 \leq x \leq \pi),$$

(2.4)

where $f(x)$ and $F(x)$ are the pdf and cdf, respectively.

Node $i$ has $n - 1$ neighbors, and the distance between $i$ and its neighbors will be $n - 1$ i.i.d random variables following the above distribution. Therefore, the ordered random variables $\{X_{(1)}^i, ..., X_{(n-1)}^i\}$ are equivalent to $n - 1$ ordered random variables drawn from distribution (2.4).

For $i$’s neighbors $X_{(j)}^i$ ($1 \leq j \leq \lceil \frac{m}{s} - 2 \rceil$), $i$ will retrieve contents from any of them with equivalent probability $\frac{s}{m-s}$ as each of them caches $s$ distinct contents. Specifically, $X_{\lfloor \frac{m}{s} - 1 \rfloor}^i$ may cache fewer useful contents for $i$, and $i$ might retrieve contents from it less frequently. Let $\xi = \lceil \frac{m}{s} \rceil - 1$, and let $L_i$ denote the average transmission distance of the contents sent to $i$, then the expectation of $L_i$ will be

$$E(L_i) = E\left( \frac{s \sum_{j=1}^{\xi-1} X_{(j)}^i + (m - s\xi) X_{(\xi)}^i}{m-s} \right),$$

(2.5)

where $X_{(j)}^i$ is the distance between $i$ and its $j$-th closest neighbor. Based on Theorem 2.5 of [22], given $X_{(r)}^i = x_r$ and $X_{(s)}^i = x_s$ ($r < s$), the conditional distribution of $X_{(r+1)}^i$, ..., $X_{(s-1)}^i$ is the distribution of $s - r - 1$ random variables drawn from $f(x)/[F(x_s) - F(x_r)]$ ($x_r \leq x \leq x_s$). Thus, given $X_{(\xi)}^i$, variables $X_{(1)}^i$, ..., $X_{(\xi-1)}^i$ are equivalent to $\xi - 1$ random variables drawn from the distribution of

$$g_{X_{(\xi)}^i}(x) = \frac{f(x)}{F(X_{(\xi)}^i)} \quad (0 \leq x \leq X_{(\xi)}^i).$$

(2.6)

Suppose a random variable $Y_{(\xi)}^i$ follows the distribution $g_{X_{(\xi)}^i}$. Then, the expec-
The expectation of \( Y_i(\xi) \) is
\[
E(Y_i(\xi)) = E(E(Y_i(\xi) | X_i(\xi))) = E\left( \int_0^{X_i(\xi)} g_{X_i(\xi)}(x) x \, dx \right) = E\left( \frac{2}{3} X_i(\xi) - \frac{1}{90} X_i(\xi)^{3} + \ldots \right),
\]
where the result on the last line is the Taylor series of the integral on the second line. To simplify the results, we approximate \( E(Y_i(\xi)) \) to the first term of the Taylor series, which leads to
\[
E(Y_i(\xi)) \approx \frac{2}{3} E(X_i(\xi)).
\]
(2.7)

Note that \( E(X_i(\xi)) \) is smaller than \( \pi \), thus the truncation error is insignificant (see simulation results). Combining formula (2.5) with formula (2.7), we can get
\[
E(L_i) = \frac{s}{m-s} E(Y_i(\xi)) + \frac{m - s \xi}{m-s} E(X_i(\xi)) \\
\approx \frac{2s}{3(m-s)} E(Y_i(\xi)) + \frac{m - s \xi}{m-s} E(X_i(\xi)) \\
\geq \frac{2}{3} E(X_i(\xi)).
\]
(2.8)

Unfortunately, \( E(X_i(\xi)) \) is not easy to compute. As \( X_i(\xi) \) is the \( \xi \)-th smallest among \( n-1 \) random variables drawn from distribution (2.4), \( X_i(\xi) \) follows the distribution of
\[
f_{X_i(\xi)}(x) = \binom{n-1}{\xi-1} (F(x))^{\xi-1} \\
\times \binom{n-\xi}{n-\xi-1} (1-F(x))^{n-\xi-1} f(x).
\]

The expectation of the random variable is
\[
E(X_i(\xi)) = \int_0^\pi x f_{X_i(\xi)}(x) \, dx.
\]
(2.9)

The above expectation is extremely difficult to obtain. The exact expectation of a similar distribution can be found in [23], where the only difference is that they consider a distribution defined in \([-\frac{\pi}{2}, \frac{\pi}{2}]\), while we consider a distribution defined...
in \([0, \pi]\). The exact expression is presented in theorem 2.1 of their paper which contains both Beta function and Gamma function, and is apparently too evolved for later analysis. Thus, we exploit the David and Johnson series to approximate \(E\left(X_i^\xi\right)\) [22]. David and Johnson state that given cdf \(F(x)\), the expected value of \(X_i^\xi\) can be approximated by:

\[
E\left(X_i^\xi\right) = Q(c_\xi) + \frac{c_\xi d_\xi}{2(n+1)}Q''(c_\xi) + O\left(\frac{1}{(n+1)^2}\right),
\]

where \(Q = F^{-1}\), \(c_\xi = \frac{\xi}{n}\), and \(d_\xi = 1 - c_\xi\). As \(F(x) = \frac{1}{2}(1 - \cos x)\), then

\[
Q(x) = F^{-1}(x) = \arccos(1 - 2x)
\]

\[
Q'(x) = \frac{dQ}{dx} = \frac{1}{\sqrt{x - x^2}}
\]

\[
Q''(x) = \frac{d^2Q}{dx^2} = \frac{2x - 1}{2(x - x^2)^{3/2}}.
\]

Then the expectation of \(X_i^\xi\) is

\[
E(X_i^\xi) = \arccos\left(1 - \frac{2\xi}{n}\right) + \frac{1 - \frac{2\xi}{n}}{2(n+1)\sqrt{\frac{\xi}{n} - \frac{\xi^2}{n^2}}} + \ldots
\]

For simplicity, we approximate \(E(X_i^\xi)\) to its first term of David and Johnson series:

\[
E(X_i^\xi) \approx \arccos\left(1 - \frac{2\xi}{n}\right). \quad (2.10)
\]

Note that the truncation error is insignificant. Even in the extreme case of \(\xi = 1\) (i.e., when the David and Johnson series is most prone to approximation error), the exact value of \(E(X_{(1)}^i) = \text{Beta}(n - \frac{1}{2}, \frac{1}{2})\) [24] which is \(\frac{\sqrt{\pi}}{\sqrt{n}}\) when \(n\) is large (based on Stirling’s approximation), while our approximated result is \(\frac{2}{\sqrt{n}}\). These two only differ by a small constant factor, and is already good enough for our analysis. We will also use simulations to verify this approximation later.

Combining formula (2.8) with formula (2.10), we can get

\[
E(L_i) \geq \frac{2}{3} \arccos\left(1 - \frac{2\xi}{n}\right). \quad (2.11)
\]
Note that $i$ is a node arbitrarily chosen, thus the above inequality actually applies to all nodes in the network.

### 2.4.3 Capacity Upper Bound

Recall $L_i$ is the average transmission distance of contents sent to $i$, the average transmission distance for all contents will be

$$\bar{L} = \frac{\sum_{i=1}^{n} L_i}{n}. \quad (2.12)$$

Then, $E(\bar{L}) = E(L_i)$, as $L_1, L_2, L_3, \ldots, L_n$ are identically distributed.

**Lemma 1.** The average transmission distance $\bar{L}$ converges in probability to $E(\bar{L})$ as $n \to \infty$, i.e., for any $\epsilon > 0$,

$$\lim_{n \to \infty} \Pr(|\bar{L} - E(\bar{L})| \geq \epsilon) = 0.$$

**Proof.** Based on formula (2.5),

$$\text{Var}(L_i) = \text{Var}\left(\frac{s \sum_{j=1}^{\xi-1} X_{(j)}^i + (m - s\xi) X_{(\xi)}^i}{m - s}\right).$$

Let $a_1 = a_2 = \ldots = a_{\xi-1} = \frac{s}{m-s}$, and $a_{\xi} = \frac{m-s\xi}{m-s}$, then

$$\text{Var}(L_i) = \text{Var}\left(\sum_{j=1}^{\xi} a_j X_{(j)}^i\right)$$

$$= \sum_{j=1}^{\xi} \text{Var}(a_j X_{(j)}^i) + \sum_{j=1}^{\xi} \sum_{k=1}^{\xi} \text{Cov}(a_j X_{(j)}^i, a_k X_{(k)}^i).$$

Based on David and Johnson series [22],

$$\text{Var}(X_{(j)}^i) = \frac{c_j d_j}{n+1} Q'(c_j)^2 + O\left(\frac{1}{n^2}\right).$$
Recall that \( Q'(c_j) = \frac{1}{\sqrt{c_j - c_j^2}} \) and \( d_j = 1 - c_j \), therefore for any \( i, j \)

\[
\text{Var} (X^i_{(j)}) = \frac{1}{n+1} + O \left( \frac{1}{n^2} \right) = O \left( \frac{1}{n} \right).
\]

In addition, for any \( i, j \) and \( k \)

\[
\text{Cov} (X^i_{(j)}, X^i_{(k)}) \leq \sqrt{\text{Var} (X^i_{(j)}) \text{Var} (X^i_{(k)})} \leq \max \left( \text{Var} (X^i_{(j)}), \text{Var} (X^i_{(k)}) \right).
\]

In this way, the variance of \( L_i \) is bounded by

\[
\text{Var} (L_i) \leq \left( \sum_{j=1}^{\xi} a_j^2 + \sum_{j=1}^{\xi} \sum_{k=1}^{\xi} a_j a_k \right) \max_{1 \leq j \leq \xi} \left( \text{Var} (X^i_{(j)}) \right)
\]

\[
= \left( \sum_{j=1}^{\xi} a_j \right)^2 \max_{1 \leq j \leq \xi} \left( \text{Var} (X^i_{(j)}) \right) = O \left( \frac{1}{n} \right).
\]

The above formula holds for all \( i \), thus the variance of \( \bar{L} \) is

\[
\text{Var} (\bar{L}) = \frac{1}{n^2} \left( \sum_{i=1}^{n} \text{Var} (L_i) + \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \text{Cov} (L_i, L_j) \right)
\]

\[
\leq \frac{n + n(n-1)}{n^2} \max_{1 \leq i \leq n} \left( \text{Var} (L_i) \right) = O \left( \frac{1}{n} \right).
\]

Based on Chebyshev’s inequality [25], for any \( \epsilon > 0 \),

\[
P \left( |\bar{L} - E(\bar{L})| \geq \epsilon \right) \leq \frac{\text{Var}(\bar{L})}{\epsilon^2} \leq \frac{1}{n\epsilon^2}.
\]

As \( n \to \infty \), the above probability goes to 0.

**Theorem 1.** Under our network and content access model, the per node capacity is upper bounded by

\[
\hat{\lambda}(s, m, n) = W \sqrt{\frac{s}{m}}.
\]  

(2.13)

**Proof.** In the proof of Theorem 5.2 [1], the authors identified that for nodes uni-
formally distributed on the surface of the sphere, if the average transmission distance is $\bar{L}$, then the capacity for each node can not be larger than \( \sqrt{\frac{8}{\pi}} \beta \frac{W}{\beta \frac{1}{2} - 1} \frac{1}{L \sqrt{n}} \). We have shown that even in the ideal scenario where each node retrieves contents from closest neighbors, the average transmission distance $\bar{L} \overset{p}{\rightarrow} \mathbb{E}(\bar{L}) \geq \frac{2}{3} \arccos \left( 1 - \frac{2e}{n} \right)$, thus per node capacity

$$\lambda(s, m, n) \leq \sqrt{\frac{8}{\pi}} \frac{W}{\beta \frac{1}{2} - 1} \frac{1}{L \sqrt{n}}$$

$$\leq \sqrt{\frac{8}{\pi}} \frac{W}{\beta \frac{1}{2} - 1} \frac{3}{2 \arccos(1 - \frac{2e}{n})}$$

$$= O \left( W \sqrt{\frac{s}{m}} \right)$$

(as $\left\lceil \frac{m}{s} \right\rceil \geq 2$ and $\arccos(1 - 2x) \approx 2\sqrt{x}$).

\[ \square \]

**Corollary 1.1.** Under our network and content access model, the per node request satisfaction rate is upper bounded by

$$\hat{\mu}(s, m, n) = W \sqrt{\frac{s}{m}} \frac{m}{m - s}. \quad (2.14)$$

**Proof.** Each node caches $s$ out of $m$ contents locally. Since the content requests follow the uniform distribution, at each node, $\frac{s}{m}$ of all requests are served by local cache, while the remaining $\frac{m-s}{m}$ of requests are served by other nodes. It has been shown that no more than $\hat{\lambda}(s, m, n)$ bits can be received at every node per second. Thus no more than $\hat{\lambda}(s, m, n)$ requests where the requested contents not in local cache can be satisfied per second. Thus, the per node request satisfaction rate is upper bounded by $\frac{m}{m-s} \hat{\lambda}(s, m, n)$ (note that requests come one after another). Therefore,

$$\hat{\mu}(s, m, n) = \frac{m}{m-s} \hat{\lambda}(s, m, n) = W \sqrt{\frac{s}{m} \frac{m}{m-s}}.$$

\[ \square \]
2.5 Achievable Capacity Lower Bound

In this section, we construct a caching scheme to show that at least a capacity of $\Theta\left(\frac{W_s}{m}\right)$ is achievable. Basically, we partition the surface of the sphere $S^2$ into cells with similar sizes. Then we show that by carefully choosing the contents to cache at each cell, most of the nodes can retrieve any content from nodes within the same cell. Additionally, we prove that transmission within the same cell will only affect a constant number of neighboring cells. Based on the above results, we prove that a capacity proportional to the number of cells can be achieved.

2.5.1 Voronoi Tessellation

In our constructed caching scheme, to partition the sphere into cells, we use Voronoi Tessellation [26], so that the divided cells have similar area. Given the $t$ initial points $\{b_1, b_2, ..., b_t\}$, Voronoi cell $V_i$ ($1 \leq i \leq t$) consists of all the points that are closer to $b_i$ than to any other $b_j$:

$$V_i = \left\{x \in S^2 \mid |x - b_i| \leq |x - b_j| \text{ for all } 1 \leq j \leq t \right\},$$

where $|x - b_i|$ is the great circle distance between $x$ and $b_i$, and $x$ can be any point on the surface of the sphere.

As shown in Lemma 4.1 of [1], for any $\rho > 0$, there is a Voronoi tessellation which can partition the sphere into cells such that each cell contains a disk of radius $\rho$ and the cell is contained in a disk of radius $2\rho$. With such a property of Voronoi tessellation, we set $\rho$ as the following:

$$\rho := \text{radius of the disk with area } \frac{8\pi}{n} \left\lceil \frac{m}{s} \right\rceil \text{ on } S^2. \quad (2.15)$$

Since each Voronoi cell contains a disk of radius $\rho$, and it is contained in a disk of radius $2\rho$, its area is at least $\frac{8\pi}{n} \left\lceil \frac{m}{s} \right\rceil$, and at most $c_1 \frac{32\pi}{n} \left\lceil \frac{m}{s} \right\rceil$, where $c_1$ is a constant close to 1, since the area of a disk with radius $2\rho$ on the surface of the sphere is different from $4\pi \rho^2$. As the surface of the sphere has an area of $4\pi$, the number of cells $t$ must satisfy

$$\frac{4\pi}{c_1} \frac{32\pi}{n} \left\lceil \frac{m}{s} \right\rceil \leq t \leq \frac{4\pi}{c_1} \frac{8\pi}{n} \left\lceil \frac{m}{s} \right\rceil$$
\[ \frac{n}{8c_1 \left\lfloor \frac{m}{s} \right\rfloor} \leq t \leq \frac{n}{2 \left\lfloor \frac{m}{s} \right\rfloor}. \]  \tag{2.16}

For the special case when \( \left\lfloor \frac{m}{s} \right\rfloor \geq \frac{n}{4} \), we have the surface of the sphere as one cell and it contains all \( n \) nodes. The number of cells \( t = 1 \), and it is still in the order of \( n/\left\lfloor \frac{m}{s} \right\rfloor \).

### 2.5.2 Finding cells that can cache all content

In our caching scheme, we want to make sure that most nodes can retrieve content from nodes in the same cell, which means that nodes in the cell can cache all content. Thus, we want to find the cells which contain at least \( \left\lfloor \frac{m}{s} \right\rfloor \) nodes, since only these cells can cache all content. To simplify the notation, let \( \eta = \left\lfloor \frac{m}{s} \right\rfloor \).

**Lemma 2.** For any cell \( V_i \) (\( 1 \leq i \leq t \)),

\[
\Pr (V_i \text{ contains at least } \eta \text{ nodes}) \geq 1 - e^{-\frac{1}{4}}. \tag{2.17}
\]

**Proof.** As nodes are uniformly distributed, and \( V_i \) has an area of at least \( \frac{8\pi}{n} \eta \), the probability \( p_i \) for any node \( j \) to fall into \( V_i \) is:

\[
\Pr (\text{Node } j \text{ falls in } V_i) = p_i \geq \frac{8\pi}{n} \eta / 4\pi = \frac{2}{n} \eta.
\]

Let \( v_i \) denote the number of nodes in cell \( V_i \), as \( n \) nodes are independently distributed, \( v_i \) follows a binomial distribution \( B(n, p_i) \). For the cumulative probability \( \Pr (v_i \leq v) \), Chernoff bound [27] states that the probability for variable \( v_i \) to be smaller than a constant \( v \) is bounded by

\[
\Pr (v_i \leq v) \leq \exp \left( -\frac{(np_i - v)^2}{2np_i} \right), \text{ for any } v \leq np_i.
\]

By setting \( v = \eta \) (note \( v < 2\eta \leq np_i \)), we can get

\[
\Pr (v_i \leq \eta) \leq \exp \left( -\frac{(np_i - \eta)^2}{2np_i} \right) \leq \exp \left( -\frac{(np_i - \frac{np_i}{2})^2}{2np_i} \right) = \exp \left( -\frac{np_i}{8} \right)
\]
\[
\leq \exp \left( -\frac{\eta}{4} \right) \leq \exp \left( -\frac{1}{4} \right).
\]

Then the probability that \( V_i \) contains at least \( \eta \) nodes is

\[
\Pr (v_i \geq \eta) \geq 1 - \Pr (v_i \leq \eta) \geq 1 - e^{-\frac{1}{4}}.
\]

As we have not made any assumption on cell \( V_i \), the above inequality actually holds for all cells. \( \square \)

Let Bernoulli random variables \( \{u_i\}_{1 \leq i \leq t} \) denote if each cell has more than \( \eta \) nodes (i.e., \( u_i = 1 \) if \( v_i \geq \eta \), and \( u_i = 0 \) otherwise). Let \( \sigma = (\sum_{i=1}^{t} u_i)/t \), then

\[
E (\sigma) = \frac{\sum_{i=1}^{t} \Pr (v_i \geq \eta)}{t} \geq 1 - e^{-\frac{1}{4}}.
\]

(2.18)

**Lemma 3.** For any \( \epsilon > 0 \),

\[
\lim_{n \to \infty} \Pr (|\sigma - E (\sigma)| \geq \epsilon) = 0.
\]

(2.19)

**Proof.** There are two cases.

(i) \( \eta \leq 50 \log n \):

For any \( u_i \) and \( u_j \),

\[
\text{Cov} (u_i, u_j) = E (u_i u_j) - E (u_i) E (u_j)
\]

\[
= \Pr (u_j = 1 | u_i = 1) \Pr (u_i = 1)
\]

\[
- \Pr (u_j = 1) \Pr (u_i = 1).
\]

Conditional on \( u_i = 1 \), the probability \( \Pr (u_j = 1 | u_i = 1) \) will not be as large as \( \Pr (u_j = 1) \). Because \( u_i = 1 \) implies that at least \( \eta \) nodes are known to fall into \( V_i \), and fewer nodes might fall into \( V_j \). Thus, the above covariance

\[
\text{Cov} (u_i, u_j) = (\Pr (u_j = 1 | u_i = 1) - \Pr (u_j = 1))
\]

\[
\times \Pr (u_i = 1) \leq 0.
\]
Therefore, the variance of $\sigma$ is

\[
\text{Var}(\sigma) = \text{Var}\left(\frac{\sum_{i=1}^{t} u_i}{t}\right) = \frac{\sum_{i=1}^{t} \text{Var}(u_i) + \sum_{i=1}^{t} \sum_{j \neq i} \text{Cov}(u_i, u_j)}{t^2} \leq \frac{\sum_{i=1}^{t} \text{Var}(u_i)}{t^2} \leq \frac{\sum_{i=1}^{t} \Pr(u_i = 1)}{t^2} \leq \frac{1}{t}.
\]

Based on Chebyshev’s inequality, $\forall \epsilon > 0$,

\[
\Pr\left(|\sigma - \mathbb{E}(\sigma)| \geq \epsilon\right) \leq \frac{\text{Var}(\sigma)}{\epsilon^2} \leq \frac{1}{t \epsilon^2}.
\]

Note that $t = \Theta(n/\eta)$ and $\eta \leq 50 \log n$, thus $t \geq c_2 \frac{n}{\log n}$, and above probability approaches 0 as $n$ goes to infinity.

(ii) $\eta > 50 \log n$:

If $\eta \geq \frac{n}{4}$, there is only one cell and it contains all $n$ nodes. As $n \geq \eta$, $\Pr(\sigma = 1) = 1$.

Otherwise, based on Vapnik-Chervonenkis Theorem [28], for a set of subsets $U$ with finite VC-dimension, and $n$ i.i.d. random variables $\{X_i\}$ with common probability distribution $G$,

\[
\Pr\left(\max_{D_i \in U} \left|\frac{I(D_i)}{n} - G(D_i)\right| \leq \epsilon\right) > 1 - \delta
\]

if $n$ satisfies

\[
n > \max \left\{ \frac{VC - d(U)}{\epsilon} \log \frac{16\epsilon}{\epsilon^2}, \frac{4}{\epsilon}, \log \frac{2}{\delta} \right\}.
\]

Here $I(D_i)$ is the number of random variables that falls in $D_i$, and $VC - d(U)$ is the VC-dimension of $U$. Based on Lemma 4.7 in [1], the VC-dimension of the set of disks on $S^2$ strictly smaller than hemisphere is 3. Thus, if we let $U$ be the set of disks with radius $\rho$ on $S^2$, and

\[
\epsilon = \frac{\eta}{n}, \text{ and } \delta = \frac{50 \log n}{n},
\]

\[
\text{and } \theta = \frac{\log n}{n},\text{ and } \eta = \frac{50 \log n}{n}.
\]
then the following inequality is always satisfied:

$$\Pr \left( \max_{D_i \in U} \left| \frac{I(D_i)}{n} - \frac{2\eta}{n} \right| \leq \frac{\eta}{n} \right) > 1 - \frac{50 \log n}{n}. $$

It can be rewritten as

$$\Pr (\text{for every } D_i \in U, I(D_i) \geq \eta) > 1 - \frac{50 \log n}{n}.$$  \hfill (2.20)

As each cell $V_i$ contains a disk of radius $\rho$,

$$\Pr (v_i \geq \eta, \text{for every } 1 \leq i \leq t) > 1 - \frac{50 \log n}{n}$$

$$\Pr (\sigma = 1) > 1 - \frac{50 \log n}{n}.$$  

The above lemma states that most of the cells can cache all contents when $\rho$ is set up based on formula (2.15).

### 2.5.3 Transmission Scheduling

In this subsection, we propose a transmission schedule which shows how the contents can be retrieved in our caching scheme. We show that under the proposed schedule, for a time period of $T$, $cWT$ can be transmitted within each cell where $c$ is a constant. A similar fact with slightly different settings has already been established in Lemma 4.4 of [1], yet we still include the proof here for completeness.

In what follows, we assume that at most one node is transmitting data at any time in each cell.

**Definition 3** (Interfering Neighbors). Two cells are interfering neighbors if there is one point in one cell and another point in another cell that is at most $\delta \rho$ away.

**Lemma 4.** For sufficiently large transmission power $P$, there exists a constant $\delta$ such that if there are no concurrent transmissions from two interfering neighbors, any transmission between nodes within the same cell can always be successfully received.
Proof. Consider a node $i$ transmitting to a node $j$ in the same cell. As each cell must be contained in a disk of radius $2\rho$, the transmission distance between $i$ and $j$ is at most $4\rho$. The signal power received at node $j$ is at least $P_{(4\rho)^\alpha}^{4\rho\alpha}$. 

Since there is no concurrent transmissions from interfering neighbors, any two nodes transmitting simultaneously must be separated by a distance of at least $\delta\rho$. Therefore, disks of radius $\delta\rho/2$ centered at each transmitter must be disjoint. Consider the transmitters that are within a distance between $a$ and $b$ from node $j$. The disk of radius $\delta\rho/2$ centered at each transmitter must be contained within an annulus of all points lying within a distance between $a - \frac{\delta\rho}{2}$ and $b + \frac{\delta\rho}{2}$ from receiver $j$. Such an annulus has an area of 

$$c_3\pi \left( \left( b + \frac{\delta\rho}{2} \right)^2 - \left( a - \frac{\delta\rho}{2} \right)^2 \right).$$

The disks of radius $\delta\rho/2$ centered at each transmitter are disjoint, therefore the number of transmitters contained in the annulus can not be more than 

$$\frac{c_3\pi((b + \frac{\delta\rho}{2})^2 - (a - \frac{\delta\rho}{2})^2)}{c_4\pi(\frac{\delta\rho}{2})^2}.$$ (2.21)

In addition, the received power at $j$ from any of the above transmitters can not exceed $\frac{P}{\delta\alpha}$. By setting $a = k\delta\rho$, and $b = (k + 1)\delta\rho$ for $k = 1, 2, 3,...$, we can get the SIR at $j$ is 

$$N_0 + c_5 \sum_{k=1}^{\infty} \frac{((k+1)\delta\rho + \frac{\delta\rho}{2})^2 - (k\delta\rho - \frac{\delta\rho}{2})^2}{(\frac{\delta\rho}{2})^2} \frac{P}{(\delta\rho)^{\alpha}}$$

$$= \frac{P}{\delta\alpha} N_0 + c_5 \frac{P}{\delta\alpha} \sum_{k=1}^{\infty} \frac{16k}{k^\alpha} + \frac{8}{k^\alpha}$$

$$\geq \frac{P}{(4\rho)^\alpha N_0 + c_5 P_{4\rho^\alpha}^{4\rho\alpha} \left(24 + \frac{16}{\alpha-2} + \frac{8}{\alpha-1}\right)} \text{ (as } \alpha > 2).$$

The above SIR is larger than $\beta$, i.e., the transmission can be successfully re-
ceived, when \( P \) is sufficiently large and \( \delta \) is chosen to satisfy:

\[
\delta > 4 \left( c_5 \beta \left( 24 + \frac{16}{\alpha - 2} + \frac{8}{\alpha - 1} \right) \right)^{\frac{1}{n}}.
\]  

(2.22)

**Lemma 5.** For a time period of \( T \), nodes in each cell can get a total period of \( \frac{T}{c_6} \) for transmission, and the transmission can always be successfully received by any node within the cell.

**Proof.** For Voronoi cell \( V_i \) and its interfering neighbor \( V'_i \), there must be two points, one in \( V_i \) and one in \( V'_i \), that are no more than \( \delta \rho \) apart. Thus, \( V_i \) and all of its interfering neighbors can be contained in a large disk of radius \( (6 + \delta) \rho \). This large disk can not contain more than \( c_6 (6 + \delta)^2 \) disks of radius \( \rho \), thus \( V_i \) can not have more than \( c_6 (6 + \delta)^2 - 1 \) interfering neighbors.

Consider coloring all the cells such that no two interfering neighbors have the same color. A well-known fact about the vertex coloring of graphs is that a graph of degree no more than \( c_7 \) can be colored by using no more than \( (1 + c_7) \) colors [29]. Thus the cells can be colored by using \( c_6 (6 + \delta)^2 \) colors. We allocate a period of \( \frac{T}{c_6 (6 + \delta)^2} \) for each color, during which the cells of that color will transmit simultaneously. As no interfering neighbors have the same color, for each period of \( \frac{T}{c_6 (6 + \delta)^2} \), the transmission in each cell can be successfully received based on Lemma 4.

\[
\square
\]

### 2.5.4 Achievable Lower Bound

Based on the above results, we now present achievable lower bounds on network capacity and request satisfaction rate.

**Theorem 2.** Under our network and content access model, an achievable lower bound on network capacity (averaged over all nodes) is

\[
\tilde{\lambda}(s, m, n) = \frac{sW}{m}.
\]  

(2.23)

**Proof.** Lemma 5 shows that for a time period of \( T \), each cell can have a period of \( \frac{T}{c_6} \) during which the transmission in the cell is always successfully received. Thus,
for those cells that can cache all contents, during the period of $\frac{T}{c_7}$, nodes within
the cell can receive $W\frac{T}{c_7}$ bits of contents regardless of which contents have been
requested. Additionally, Lemma 3 has shown that at least $c_8\frac{n}{\eta}$ cells can cache all
contents. Summing over all the cells that can cache all the contents, the total
number of bits received is

$$c_8\frac{n}{\eta} \times W\frac{T}{c_7} = c_9\frac{WnT}{\eta}.$$ (2.24)

Then on average each node can achieve a capacity of

$$\tilde{\lambda}(s, m, n) = c_9\frac{WnT}{\eta}/nT = c_9\frac{W}{\eta} \approx \frac{sW}{m},$$ (2.25)

as $\eta = \lceil \frac{m}{s} \rceil$.

**Corollary 2.1.** *Under our network and content access model, an achievable lower
bound on request satisfaction rate (averaged over all nodes) is*

$$\tilde{\mu}(s, m, n) = \frac{sW}{m - s}.$$ (2.26)

*Proof.* As each node caches $s$ contents locally, $\frac{s}{m}$ of incoming requests will be sat-
sfied by local cache, and the remaining requests require nodes to retrieve contents
from other nodes. Theorem 2 has shown that a capacity of $\tilde{\lambda}(s, m, n)$ is achiev-
able, which means that $\tilde{\lambda}(s, m, n)$ requests for contents not cached locally can be
satisfied per second on average. An additional $\frac{s}{m - s} \tilde{\lambda}(s, m, n)$ requests for contents
cached locally can be received and directly satisfied. Then,

$$\tilde{\mu}(s, m, n) = \tilde{\lambda}(s, m, n) + \frac{s}{m - s} \tilde{\lambda}(s, m, n) = \frac{sW}{m - s}.$$

$\square$


2.6 Numerical Results

2.6.1 Approximation Validation and Numerical Results

We first validate the two approximations we made in section 2.4. Fig. 2.1(a) verifies the approximation in formula (2.7), i.e., $E(Y_i(\xi))$ is approximated to the first term of Taylor series: $\frac{2}{3}E(X_i(\xi))$. We compare the actual value of $E(Y_i(\xi))$ with its approximated value based on the value of $\xi$ and the number of nodes. Fig. 2.1(a) shows that under all three values of $n$, the approximation error is quite small for all possible values of $\xi$ ($1 \leq \xi \leq n - 1$). Fig. 2.1(b) validates the approximation in formula (2.10), i.e., $E(X_i(\xi))$ can be approximated to the first term of David and Johnson series: $\arccos(1 - \frac{2\xi}{n})$. From the figure, the difference between the actual value and the approximated value is negligible for various $n$ and $\xi$.

Fig. 2.2 shows the effect of cache size on network capacity and request satisfaction rate based on different total sizes of unique content, where $n = 10000$. As shown in the figure, the achievable capacity lower bound grows linearly with cache size, while the capacity upper bound grows quickly when the cache size is small, and much slower when the cache size is large. We can also see that the network capacity decreases with the increase of the total size of unique content, because at this time contents have to be retrieved from faraway nodes. Fig. 2.2(b) shows how the cache size affects the request satisfaction rate. As can be seen, the upper bound of the request satisfaction rate grows quickly when the cache size is small, and almost linearly when the cache size is large. On the other hand, its lower bound grows linearly when the cache size is small, and grows faster when cache size is large, especially with small $m$ ($m = 3000$). Therefore, increasing the cache size may significantly improve the request satisfaction rate when the initial cache size is either quite small or quite large. In other cases, the request satisfaction rate generally grows linearly with the cache size.

2.6.2 Comparisons to Existing Work

In this subsection, we compare our results with the analytical results presented in [1, 15], and show that our results are consistent with theirs when our caching scenario matches theirs.
(a) $E(Y^i_\xi)$

(b) $E(X^i_\xi)$

Figure 2.1: Approximation validation for $E(Y^i_\xi)$ and $E(X^i_\xi)$ (i.e., formula (2.7) and formula (2.10)).

Figure 2.2: Capacity and request satisfaction rate

(i) In case of $m = sn$, the total cache size is just enough to store all contents. To find the capacity upper bound, each content is only cached at one node. For any node $i$ receiving a request, the requested content may be cached at any node with equal probability. Thus, if the requested content is not cached locally, it is equally likely for $i$ to retrieve the requested content from any other node. This scenario is similar to the scenario of random network in [1], where all nodes randomly choose a destination for transmission. Our derived upper bound does conform to their results, as they present an upper bound of $O\left(\frac{W}{\sqrt{n}}\right)$ while our $\lambda(s,m,n)$ is also
upper bounded by \( W \sqrt{\frac{2}{m}} = \frac{W}{\sqrt{n}} \) when \( m = sn \).

(ii) In case of \( \frac{m}{2} \leq s < m \), two nodes will be enough to cache all the contents. If contents are carefully cached, each node only need to retrieve contents from its closest neighbor. This scenario is similar to the scenario constructed in [15], as communications are restricted to closest neighbors. Under this special case, we are also able to obtain an achievable capacity of \( \Theta(W) \), which conforms to their results.

(iii) The most important implication of our results is that, the network capacity and the request satisfaction rate (averaged over all nodes) can remain constant even if the number of nodes increase. This is because although more nodes bring in more interferences, high node density also helps each node retrieve contents from closer neighbors.

Fig. 2.3 compares our results with [1, 15] based on numerical results. Fig. 2.3(a) shows how the capacity changes with \( \frac{s}{m} \) when \( n = 10^6 \). Apparently, the capacity results of Gupta-Kumar [1] and Grossglauser-Tse [15] will not change with \( \frac{s}{m} \). When \( \frac{s}{m} \) is very small, our result is comparable to Gupta-Kumar. However, as \( \frac{s}{m} \) increases, our capacity result quickly approach to Grossglauser-Tse’s result. Fig. 2.3(b) shows the capacity as a function of the number of nodes when \( \frac{s}{m} = \frac{1}{10} \). Grossglauser-Tse shows that the network capacity can remain constant through node mobility. Similarly, caching also helps achieve constant network capacity. For wireless networks without considering mobility and caching, as shown in Gupta-
Kumar, the capacity decreases quickly as the number of nodes increases. Note that although Grossglauser-Tse can achieve higher network capacity than ours, they are at the cost of long delay. On the other hand, caching can increase the network capacity, and reduce the content access delay.

2.6.3 Discussions

Although our analysis has been based on the assumption that all contents have the same size of 1 bit, the proposed capacity upper and lower bounds are still valid even if contents have various sizes. For the upper bound, in the ideal scenario, each node still caches $s$ bits of contents locally, and has its closest neighbors cache the remaining $m - s$ bits of contents. Thus, the average distance to retrieve contents from others ($\bar{L}$) is unchanged. Recall that the capacity upper bound is $\Theta(\frac{1}{\sqrt{nL}})$. Since $L$ is unchanged, the capacity upper bound still holds. For the achievable lower bound, we only need to make sure that $m$ bits can be cached in most of the Voronoi cells, which is irrelevant to the size of individual content.

Even when the nodes are distributed on a planar disk rather than the surface of the sphere as considered in this chapter, we believe similar capacity upper and lower bounds can still be derived. For the capacity upper bound, the average distance to retrieve a content is still in the order of $\Theta(\frac{\sqrt{\pi}}{m})$; while for the capacity lower bound, the parameters to construct Voronoi tessellation are almost the same. In addition, when the radius of the sphere increases, both the upper bound and the achievable lower bound will remain unchanged.

2.7 Conclusion

In this chapter, we have studied the capacity of wireless networks with caching. We proved that for nodes uniformly distributed on the surface of sphere, the network capacity is upper bounded by $\Theta(W\sqrt{\frac{\pi}{m}})$. We also propose a caching scheme, based on which a capacity of $\Theta(W\frac{\sqrt{\pi}}{m})$ is achievable. More importantly, our results suggest that through caching, it is possible for nodes to obtain constant capacity even if the number of nodes increases.
Chapter 3

Popularity-Aware Caching in Wireless Networks

3.1 Introduction

Wireless ad hoc networks enable communications among mobile nodes without any infrastructure support, as nodes themselves relay and forward packets for each other. Due to the interference between concurrent transmissions, the per-node capacity generally decreases with the increasing number of nodes in the network. This was first found by Gupta and Kumar [1]. They showed that for a wireless ad hoc network with $n$ nodes, each node can at most transmit at a rate of $\Theta(\frac{1}{\sqrt{n}})$ to its destination, even with optimal scheduling of transmissions from various nodes.

Later, Grossglauser and Tse [15] examined the capacity of a mobile ad hoc network. They proved that with node mobility, the per-node capacity can be kept constant even when the number of nodes in the network grows. In their proposed scheme, two nodes wait until they move close enough to transmit data. In this way, the per-node capacity can be significantly improved, since the transmissions are limited to nearby neighbors and the consequent interference will affect much less number of nodes. However, the delay of their proposed scheme is extremely long, because nodes need to wait until they are close enough to transmit.

Caching can be used to improve the network capacity. With caching, contents are stored close to the users, which shortens the transmission distance and im-
proves the per-node capacity. Although theoretical study of caching has attracted considerable attention, most existing research focuses on caching in the Internet [10, 13, 30, 31], and the results can not be directly applied to wireless ad hoc networks. It was only recently that researchers have become interested in studying the fundamental performance limits of caching in wireless ad hoc networks. Liu et al. [20] have derived capacity upper bounds for two specific content access schemes, and examined how the per-node capacity is affected by the cache size and the number of nodes. In [32], the authors proved that for wireless networks with caching, the per-node capacity will remain constant even when the number of nodes grows. However, these works simply assume that content popularity follows a uniform distribution. They ignore the fact that in reality some contents are accessed much more frequently than others, which requires a more complicated caching strategy to maximize the per-node capacity, and may lead to totally different capacity scaling laws.

In this chapter, we quantify the effect of popularity-aware caching on the capacity of wireless ad hoc networks. To maximize the per-node capacity given the content popularity, we first study the optimal caching strategy; i.e., how frequently contents with various popularity should be cached so that the per-node capacity is maximized. Based on the optimal caching strategy, we then evaluate the effect of content popularity on per-node capacity, and derive different capacity scaling laws for networks with different content popularity skewness. For all the different capacity scaling laws, we analytically investigate how the per-node capacity is affected by various parameters, including the number of nodes \((n)\), the cache size \((s)\), and the number of unique contents \((m)\). Basically, as the distribution of the content popularity changes from uniform distribution to more skewed distributions, the per-node capacity increases from \(\Theta \left( \sqrt{\frac{s}{m}} \right)\) to roughly \(\Theta (\sqrt{s})\). These results suggest that for wireless ad hoc networks with caching, when contents have skewed popularity, increasing the number of nodes monotonically increases the per-node capacity. Besides the analytical results, we also propose a distributed caching algorithm which enables nodes to optimally cache contents and maximize the per-node capacity only based on local knowledge.

The rest of the chapter is organized as follows. Section 3.2 reviews the related work. Section 3.3 introduces the network model, and Section 3.4 formulates the
problem. We derive the capacity scaling laws in Section 3.5, and analyze the effects of various parameters in Section 3.6. We present the distributed caching algorithm in Section 3.7. Section 3.8 presents evaluation results, and Section 3.9 concludes the chapter.

3.2 Related Work

Caching in wireless networks is a traditional topic that has been studied from various aspects. In [6], Yin and Cao have designed and evaluated cooperative caching schemes to support data access in wireless ad hoc networks. In [7], the authors have implemented and examined a caching scheme for wireless P2P networks. Fiore et al. [8] have proposed an algorithm to help users decide whether a content should be cached, so as to reduce the data redundancy among neighbors. Recently, various content caching and cache replacement algorithms have also been proposed for information-centric networks [33, 34]. However, these algorithms are only based on heuristic and the associated performance evaluations are based on simulations. None of them has analytically investigated the fundamental performance limits of caching in wireless networks.

The problem of optimal caching has only been investigated in a few existing works. When the content popularity is known, Cohen and Shenker [35] have studied the problem of how the contents with various popularity should be cached in P2P networks, so that the number of searches to retrieve the content is minimized. Besides, Jin and Wang [16] have proposed techniques to determine how contents should be cached at various nodes in wireless networks. When the content popularity is unknown, content caching at the base stations has been examined in [36, 37]. The authors have proposed learning-based approaches, where the base station learns the content popularity based on user requests or context information, and then caches the contents according to the observed popularity distribution. However, since these algorithms require the exact content popularity, they may not work well in a mobile ad hoc network, which does not have a base station to observe and process all content requests. Furthermore, these works [35, 16, 36, 37] only proposed caching schemes, and the effect of caching on network capacity was not investigated.
Recently, researchers have become more interested in studying the fundamental property of caching in wireless networks, based on different models and assumptions. In [38, 39], Ji et al. presented theoretical bounds as well as simulations results on per-node capacity, but their results were limited to single-hop data transmission. In [40], the authors presented theoretical results on network throughput and content access delay for a heterogenous network, which consists of base stations, relays, and device-to-device pairs. Niesen et al. [17] focused on deriving the feasible request serving rate by knowing where and what data has been cached. In [18], Gitzenis et al. examined the asymptotic laws for joint replication and delivery in wireless networks. They derived the minimum throughput on each link so that every node can satisfy one request per second. However, these works have assumed different communication or content access models, thus leading to results different from ours.

Following the model in [1, 15], the capacity of wireless networks with caching has also been examined in [20, 32, 41]. Liu et al. [20] have derived capacity upper bounds for a wireless ad hoc network under two specific content access schemes. In [32], the authors have derived the asymptotic bounds of per-node capacity, which suggests that for contents with uniform popularity, the per-node capacity will remain constant even when the number of nodes grows. Yet, in both works [20, 32], the results are under the assumption that content access pattern follows a simple uniform distribution, and the fundamental problem of how the distribution of content popularity affects the scaling laws of the per-node capacity has not been explored. In [41], although the authors investigated the per-node capacity based on different Zipf distributions, their results are based on a specific random caching algorithm, where nodes uniformly cache a few popular contents locally. Since caching more frequently accessed contents at more nodes can further improve the per-node capacity, their capacity result does not explore the full potential of caching. In this chapter, we address this issue by presenting a capacity upper bound when contents are optimally cached. Moreover, our result is more interesting, which suggests that the per-node capacity will not diminish even when the number of nodes grows.
3.3 Preliminaries

3.3.1 Network Model

We consider a wireless network consisting of \( n \) nodes that are independently and uniformly distributed on the surface of a unit sphere. As in [1], we analyze the per-node capacity when nodes are located on the surface of sphere \( S^2 \) rather than on a disk so as to eliminate the edge effects; i.e., nodes near the edge have much fewer neighbors than nodes near the center.

Assume all nodes employ the same amount of power for transmission, and they transmit over a common wireless channel which can support \( W \) bits per second. According to the physical model in [1], when a node \( i \) sends data to node \( j \), the transmission rate can reach \( W \) bits per second, if the signal-to-interference-plus-noise ratio (SINR) at node \( j \) is greater than or equal to \( \beta \), where \( \beta \) is the minimum SINR for successful reception.

3.3.2 Content Access and Zipf Distribution

Let \( m \) denote the number of unique contents throughout the network. To simplify the analysis, we assume each content has one bit. These \( m \) contents are cached by various nodes. Suppose each node has a cache size of \( s \). For any node \( i \), it caches \( s \) out of \( m \) contents locally. In case each content has \( b \) bits rather than one bit, the capacity results can be obtained by replacing \( s \) with \( \frac{s}{b} \). The per-node capacity decreases as the content size \( b \) increases, since a larger content size results in a longer average distance for the nodes to retrieve the content.

We assume the three parameters \( n, s \) and \( m \) are independent. In wireless networks with caching, generally each node can only cache a small portion of the \( m \) unique contents, in this way, \( m \gg s \). On the other hand, to guarantee that all content requests can be served by the nodes in the network, the total cache size of all nodes should be greater than or equal to \( m \), i.e., \( ns \geq m \). Thus, we assume \( \frac{m}{n} \leq s \ll m \).

At each node, there is always a content request, and a new request will arrive after the previous request has been served. If the requested content has been cached locally, the request can be directly served. Otherwise, the node will retrieve the
requested content from other nodes via multi-hop transmission. We assume the content requests are independent. The popularity of the contents follows a Zipf distribution, i.e., the probability that the $i$-th most popular content being requested is proportional to $\frac{1}{i^\gamma}$. This content access pattern has been used by previous studies, and existing works [42, 43] have shown that the content access pattern in the Internet follows a Zipf distribution. We focus on the scenario when the content popularity changes slowly or even remains unchanged. Typical examples include music files, and uploaded videos, which may remain popular for a long period of time. There are $m$ unique contents, and the probability that the $i$-th most popular content will be requested, denoted as $\rho_i$, will be $\rho_i = \frac{1}{i^{1+H_{m,\gamma}}}$, where $H_{m,\gamma}$ is the generalized harmonic number, given by

$$H_{m,\gamma} = \sum_{j=1}^{m} \frac{1}{j^\gamma}.$$  

For $H_{m,\gamma}$, it must satisfy

$$\int_{1}^{m+1} \frac{1}{x^\gamma} \, dx \leq H_{m,\gamma} = \sum_{j=1}^{m} \frac{1}{j^\gamma} \leq 1 + \int_{1}^{m} \frac{1}{x^\gamma} \, dx.$$ 

Accordingly, we have

$$\begin{cases} \log(m+1) \leq H_{m,\gamma} \leq \log m + 1, & \text{if } \gamma = 1 \\ \frac{(m+1)^{1-\gamma} - 1}{1 - \gamma} \leq H_{m,\gamma} \leq \frac{m^{1-\gamma} - 1}{1 - \gamma} + 1, & \text{otherwise} \end{cases}$$

As $m \to \infty$, $H_{m,\gamma}$ converges if and only if $\gamma > 1$.

### 3.3.3 Capacity

As in [1, 15], the per-node capacity describes each node’s capability to transmit or retrieve contents. In a wireless network with caching, we define the per-node capacity as the number of bits each node utilizes per second to satisfy content requests, which includes the amount of contents it receives from others, and the contents in the local cache that have been used to serve requests.

We study the scaling behavior of the per-node capacity $C$ based on four param-
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>The per-node capacity</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>The Zipf distribution exponent</td>
</tr>
<tr>
<td>$H_{m,\gamma}$</td>
<td>The generalized harmonic number</td>
</tr>
<tr>
<td>$L$</td>
<td>The average distance to retrieve a content</td>
</tr>
<tr>
<td>$L^*$</td>
<td>The lower bound on $L$</td>
</tr>
<tr>
<td>$L_i$</td>
<td>The average distance to retrieve content $i$</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td>The average distance to retrieve contents from others</td>
</tr>
<tr>
<td>$\mathcal{L}_i$</td>
<td>The average distance to retrieve content $i$ from others</td>
</tr>
<tr>
<td>$l(i)$</td>
<td>The distance to retrieve content $i$ in Algorithm 1</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>The number of contents with density $\frac{1}{m}$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>The number of contents with density $\frac{1}{s}$</td>
</tr>
<tr>
<td>$m$</td>
<td>The number of unique contents</td>
</tr>
<tr>
<td>$n$</td>
<td>The number of nodes</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>The feasible region of the content densities</td>
</tr>
<tr>
<td>$P$</td>
<td>The vector of content density</td>
</tr>
<tr>
<td>$p_i$</td>
<td>The density of content $i$</td>
</tr>
<tr>
<td>$P^*$</td>
<td>The optimal content density</td>
</tr>
<tr>
<td>$p_i^*$</td>
<td>The optimal density of content $i$</td>
</tr>
<tr>
<td>$\mathcal{P}$</td>
<td>The cache hit ratio</td>
</tr>
<tr>
<td>$\rho_i$</td>
<td>The popularity of content $i$</td>
</tr>
<tr>
<td>$s$</td>
<td>The cache size of each node</td>
</tr>
<tr>
<td>$T(i)$</td>
<td>Value maintained for each content cached locally</td>
</tr>
<tr>
<td>$t$</td>
<td>The time period in the steady state</td>
</tr>
<tr>
<td>$t_i$</td>
<td>The total time that nodes in the network cache content $i$</td>
</tr>
<tr>
<td>$t_{i,j}$</td>
<td>The total time that node $j$ caches content $i$</td>
</tr>
<tr>
<td>$u$</td>
<td>The dual variable</td>
</tr>
<tr>
<td>$u^*$</td>
<td>The optimal value of the dual variable</td>
</tr>
<tr>
<td>$V$</td>
<td>The inflation value maintained at each node</td>
</tr>
<tr>
<td>$W$</td>
<td>The number of bits transmitted per second</td>
</tr>
</tbody>
</table>

Table 3.1: Notation

Parameters: the number of nodes $n$, the number of unique contents $m$, the cache size $s$, and the Zipf parameter $\gamma$. The per-node capacity is studied under the assumption, $n \rightarrow \infty$, $m \rightarrow \infty$ and $\frac{m}{n} \leq s \ll m$. Table 3.1 lists the notations used in the chapter.
3.4 Problem Formulation

When content popularity is known, the per-node capacity depends on the caching strategy, i.e., for content $i$ with popularity $\rho_i$, how many nodes in the network should cache that content? Let $P = [p_1, p_2, \ldots, p_m]$ denote the densities of the contents, where $p_i$ is the density of content $i$ (i.e., the fraction of content $i$ among all $ns$ contents). We aim to find the optimal content densities, such that when the contents are cached according to the optimal densities, the per-node capacity is maximized.

Note that the content density $P$ has some practical constraints. First, the sum of the densities of all contents must be less than or equal to 1:

$$\sum_{i=1}^{m} p_i \leq 1. \quad (3.3)$$

Second, it is meaningless for any node to cache more than one copy of the same content locally, thus $p_i$ must satisfy

$$p_i \leq \frac{1}{s}, \text{ for } i = 1, \ldots, m. \quad (3.4)$$

Third, to guarantee that all content requests can be served, each content must have at least one copy in the network, that is:

$$p_i \geq \frac{1}{ns}, \text{ for } i = 1, \ldots, m. \quad (3.5)$$

To compute the per-node capacity based on the densities $P$, we review one important result from [1]. In that paper, the authors proved that for $n$ nodes uniformly distributed on the surface of a sphere, the per-node capacity is upper bounded by $\frac{W}{L\sqrt{n}}$, where $L$ is the average transmission distance between the source nodes and destination nodes. This result implies that the per-node capacity will increase as nodes retrieve contents from closer neighbors. Hence, to obtain the highest capacity, we should find the densities $P$ that can minimize the average transmission distance $L$, where $P$ is subject to constraints (3.3), (3.4) and (3.5).

Consider a content $i$ with popularity $\rho_i$ and density $p_i$. Since $n$ nodes with cache size $s$ can in total cache $ns$ contents, $i$ is cached at $nsp_i$ nodes in the network (note
that each node caches at most one copy of $i$ locally). For the remaining $n - nsp_i$ nodes that have not cached $i$, they have to retrieve $i$ from their neighbors. Thus, each node that caches $i$ will on average be responsible for requests of $i$ from $\frac{1}{p_i} - 1$ other nodes. As all $n$ nodes are uniformly and independently distributed, for any node $j$ that caches $i$, based on [32], its average distance to itself and its closest $\frac{1}{p_i} - 1$ neighbors is approximately:

$$L_i \approx \frac{2}{3} \arccos \left(1 - \frac{2}{n} \left(\frac{1}{p_i} - 1\right)\right) \geq \sqrt{\frac{1}{p_i sn} - \frac{1}{n}} \geq \sqrt{\frac{1}{p_i sn} - \frac{1}{n}}.$$  

Therefore, the average transmission distance of content $i$ is at least $\sqrt{\frac{1}{p_i sn} - \frac{1}{n}}$. Applying the inequality to all $m$ contents, the transmission distance $L$ averaged over all contents will be

$$L = \sum_{i=1}^{m} L_i \cdot \rho_i \geq \sum_{i=1}^{m} \left(\sqrt{\frac{1}{np_i s}} - \sqrt{\frac{1}{n}}\right) \rho_i.$$  \hspace{1cm} (3.6)

Based on the above inequality, the optimal value of the following nonlinear program $L^*$, will be a lower bound on the average transmission distance $L$.

$$\min_{P \in \Omega} L = f(P) = \sum_{i=1}^{m} \left(\frac{1}{\sqrt{p_i sn}} - \frac{1}{\sqrt{n}}\right) \rho_i$$  \hspace{1cm} (3.7)

s.t.  \hspace{0.5cm} g(P) = \sum_{i=1}^{m} p_i - 1 \leq 0$$

where $\Omega = [\frac{1}{ns}, \frac{1}{s}]^m$. The above convex nonlinear program is referred to as the primal problem, and it has the same constraints (3.3), (3.4) and (3.5).

To obtain $L^*$, let us consider the dual of the above nonlinear program:

$$\max \hspace{0.5cm} d = D(u)$$  \hspace{1cm} (3.8)

s.t.  \hspace{0.5cm} u \geq 0,$$
where the dual objective function $D(u)$ is given by

$$D(u) = \min_{P \in \Omega} (f(P) + u \cdot g(P))$$

$$= \min_{P \in \Omega} \left( \sum_{i=1}^{m} \left( \frac{1}{\sqrt{p_i s_i}} - \frac{1}{\sqrt{n}} \right) \rho_i + u \left( \sum_{i=1}^{m} p_i - 1 \right) \right).$$

The above nonlinear program is referred to as the dual problem.

**Definition 4.** A pair $(P^*, u^*)$ with $P^* \in \Omega$ and $u^* \geq 0$ satisfies the global optimality conditions for the primal problem, if

\begin{align*}
(i) & \quad f(P^*) + u^* g(P^*) = \min_{P \in \Omega} (f(P) + u^* g(P)) \quad (3.9) \\
(ii) & \quad u^* g(P^*) = 0 \quad (3.10) \\
(iii) & \quad g(P^*) \leq 0 \quad (3.11)
\end{align*}

**Lemma 6.** If a pair $(P^*, u^*)$ satisfies the global optimality condition given in Definition 4, $P^*$ is optimal in the primal problem.

**Proof.** The detailed proof can be found in [44]. Basically, given Eq. (3.10), $f(P^*) = D(u^*)$. Weak duality theorem [44] states that $f(P) \geq D(u)$ for any feasible $P$ and $u$, therefore $P^*$ optimizes the primal problem. \qed

In the above lemma, $P^*$ optimizes the primal problem, i.e., $P^*$ represents the optimal content density that maximizes the per-node capacity. The above lemma states that, the optimal content density $P^*$ can be obtained by solving (3.9), (3.10) and (3.11). Then, the minimum transmission distance is $L^* = f(P^*)$, and accordingly the per-node capacity is upper bounded by $\Theta \left( \frac{W}{L^* \sqrt{n}} \right)$.

### 3.5 Capacity Upper Bound

#### 3.5.1 Optimal Content Densities

In this subsection, we discuss the optimal content densities. Based on Eq. (3.9), the optimal $P^*$ minimizes $f(P) + u^* g(P)$. We compute the partial derivative of
\( f(P) + u^* g(P) \) over any \( p_i \), which is

\[
\frac{\partial (f(P) + u^* g(P))}{\partial p_i} = u^* - \frac{1}{2} \frac{\rho_i p_i^{-\frac{2}{3}}}{\sqrt{sn}}.
\] (3.12)

The above derivative monotonically increases with \( p_i \), and as \( u^* \geq 0 \) (the constraint of the dual), the function \( f(P) + u^* g(P) \) will first decrease with \( p_i \), and then increase with \( p_i \) when \( p_i \) is larger than \( \left( \frac{\rho_i}{2u^*} \right)^{\frac{2}{3}} \). Thus, the function \( f(P) + u^* g(P) \) is minimized, when \( p_i = \left( \frac{\rho_i}{2u^*} \right)^{\frac{2}{3}} \). Recall that \( p_i \) is restricted by \( \frac{1}{sn} \leq p_i \leq \frac{1}{s} \), hence the optimal content densities are given as follows:

\[
p_i^* = \begin{cases} 
\frac{1}{s}, & \text{if } \left( \frac{\rho_i}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}} \geq \frac{1}{s} \\
\left( \frac{\rho_i}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}}, & \text{if } \frac{1}{sn} < \left( \frac{\rho_i}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}} < \frac{1}{s} \\
\frac{1}{sn}, & \text{if } \left( \frac{\rho_i}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}} \leq \frac{1}{sn}
\end{cases}
\] (3.13)

Basically, the above equation indicates that: popular contents (with densities equal to \( \frac{1}{s} \)) are cached by all nodes in the network, and unpopular contents (with densities equal to \( \frac{1}{sn} \)) are only cached at one node. For the remaining contents, their densities are proportional to \( \rho_i^{\frac{2}{3}} \).

Due to the convexity of the primal problem and the condition \( \frac{m}{n} \leq s \ll m \), \( P^* \) exists and is unique. To simplify the notations, let \( \mu \) denote the number of contents with density \( \frac{1}{s} \), and let \( \lambda \) denote the number of contents with density \( \frac{1}{sn} \). Then, among all \( m \) contents, the most popular \( \mu \) contents each has a density of \( \frac{1}{s} \), the most unpopular \( \lambda \) contents each has a density of \( \frac{1}{sn} \), and the density for each of the remaining contents is \( \left( \frac{\rho_i}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}} \). The constraint of the primal problem must be tight, thus the densities of all contents add up to 1, which leads to:

\[
\frac{\mu}{s} + \left( \frac{1}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}} \sum_{i=\mu+1}^{m-\lambda} \left( \frac{1}{H_{m,\gamma}} \right)^{\frac{2}{3}} + \frac{\lambda}{sn} = 1.
\] (3.14)

Specifically, for the value of \( \mu \), when \( \left( \frac{\rho_i}{2u^* \sqrt{sn}} \right)^{\frac{2}{3}} \geq \frac{1}{s} \), \( \mu \) is the non-negative
integer satisfying
\[
\left( \frac{\rho_{\mu+1}}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} < \frac{1}{s} \leq \left( \frac{\rho_\mu}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}}.
\] (3.15)

When \( \mu \) is reasonably large, \( \left( \frac{\rho_\mu}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} \approx \frac{1}{s} \), and
\[
\mu \approx \left( \frac{1}{\sqrt{n}} \frac{s}{2u^* H_{m,\gamma}} \right)^{\frac{1}{\gamma}}.
\] (3.16)

While for the value of \( \lambda \), when \( \left( \frac{\rho_m}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} \leq \frac{1}{s n} \), \( \lambda \) is the non-negative integer that satisfies
\[
\left( \frac{\rho_{m-\lambda}}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} \leq \frac{1}{s n} < \left( \frac{\rho_{m-\lambda}}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}}.
\] (3.17)

If \( \lambda \) is not too close to \( m \), \( \left( \frac{\rho_{m-\lambda}}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} \approx \frac{1}{s n} \), and
\[
m - \lambda \approx \left( \frac{s n}{2u^* H_{m,\gamma}} \right)^{\frac{1}{\gamma}}.
\] (3.18)

In the next subsection, we derive the minimum transmission distance, by solving the above equation where \( \mu \) and \( \lambda \) are restricted by inequalities (3.15) and (3.17), respectively.

3.5.2 Minimum Transmission Distance

We solve the primal problem based on various values of Zipf parameter \( \gamma \): \( \gamma < \frac{3}{2} \), \( \gamma = \frac{3}{2} \) and \( \gamma > \frac{3}{2} \).

3.5.2.1 \( \gamma < \frac{3}{2} \)

**Lemma 7.** In case of \( \gamma \leq \frac{3}{2} \), \( \mu = 0 \).

**Proof.** We use contradiction to prove \( \mu = 0 \). Assume \( \mu \geq 1 \), based on inequality (3.15), for the most popular content, we have \( \left( \frac{\rho_{\mu}}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} \geq \frac{1}{s} \), which leads to:
\[
\left( \frac{1}{2u^* / \sqrt{s n}} \right)^{\frac{2}{\gamma}} \geq \left( H_{m,\gamma} \right)^{\frac{2}{3}} / s.
\] (3.19)
Consequently,
\[
\frac{\mu}{s} + \sum_{i=\mu+1}^{m-\lambda} \left( \frac{\rho_i}{2u^*\sqrt{s/n}} \right)^{\frac{2}{3}} \geq \sum_{i=1}^{m-\lambda} \left( \frac{\rho_i}{2u^*\sqrt{s/n}} \right)^{\frac{2}{3}}
\geq \frac{(H_{m,\gamma})^{\frac{2}{3}}}{s} \geq \frac{H_{m-\lambda,\frac{2\gamma}{3}}}{s}.
\]

Combining inequality (3.19) and (3.17), when \( \lambda \geq 1 \), \( (m - \lambda + 1) \frac{2\gamma}{3} \geq n \), which means \( m - \lambda + 1 \geq n^{\frac{2}{3\gamma}} \); otherwise (i.e., \( \lambda = 0 \)), \( m - \lambda + 1 = m + 1 \). Combining the two cases of \( \lambda \geq 1 \) and \( \lambda = 0 \), we have \( (m - \lambda + 1) \geq \min(m + 1, n^{\frac{2}{3\gamma}}) \). As \( m \to \infty \) and \( n \to \infty \), \( m - \lambda \to \infty \). Since \( \frac{2\gamma}{3} \leq 1 \), \( H_{m-\lambda,\frac{2\gamma}{3}} \) goes to infinity, and
\[
\frac{\mu}{s} + \sum_{i=\mu+1}^{m-\lambda} \left( \frac{\rho_i}{2u^*} \right)^{\frac{2}{3}} \gg 1.
\]

The above inequality contradicts Eq. (3.14), which suggests that our assumption of \( \mu \geq 1 \) when \( \gamma \leq \frac{3}{2} \) cannot be true.

For the value of \( \lambda \) we have the following lemma.

**Lemma 8.** When \( \gamma < \frac{3}{2} \), \( \lambda = 0 \) if and only if \( m < (1 - \frac{2}{3\gamma}) sn \).

**Proof.** Based on Lemma 7, when \( \gamma < \frac{3}{2} \), \( \mu = 0 \). By assuming \( \lambda = 0 \), Eq. (3.14) becomes
\[
\left( \frac{1}{2u^*\sqrt{s/n}} \right)^{\frac{2}{3}} \sum_{i=1}^{m} \left( \frac{1}{H_{m,\gamma}} \right)^{\frac{2}{3}} = 1.
\]

(3.20)

Since no content has a density of \( \frac{1}{sn} \), according to Eq. (3.13), even for the most unpopular content, we have
\[
\left( \frac{\rho_m}{2u^*\sqrt{s/n}} \right)^{\frac{2}{3}} \geq \left( \frac{1}{2u^*\sqrt{s/n}} \right)^{\frac{2}{3}} \left( \frac{1}{H_{m,\gamma}} \right)^{\frac{2}{3}} \geq \frac{1}{sn}.
\]

(3.21)

Combining Eq. (3.20) and inequality (3.21) leads to
\[
\left( \frac{1}{2u^*\sqrt{s/n}} \right)^{\frac{2}{3}} \left( \frac{1}{H_{m,\gamma}} \right)^{\frac{2}{3}} = \left( \frac{1}{m} \right)^{\frac{2\gamma}{3}} \frac{H_{m,\frac{2\gamma}{3}}}{H_{m,\frac{2\gamma}{3}}} = 1 - \frac{2}{3\gamma} \frac{m}{sn}.
\]
which is the same as $m < \left(1 - \frac{2\gamma}{3}\right)sn$.

On the other hand, when both Eq. (3.20) and inequality (3.21) hold, it is guaranteed that $\lambda = 0$. Hence, when $\gamma < \frac{3}{2}$, $\lambda = 0$ if and only if $m < \left(1 - \frac{2\gamma}{3}\right)sn$.

Based on the above two lemmas, we have the following:

**Proposition 1.** When $\gamma < \frac{3}{2}$, a lower bound on the average transmission distance is

$$L^* = \begin{cases} 
\frac{1}{\sqrt{sn}} \left(\frac{H_{m,\gamma}}{H_{m,\gamma}}\right)^{\frac{3}{2}}, & \text{if } m < \left(1 - \frac{2\gamma}{3}\right)sn \\
\frac{(m - \lambda)^{1 - \gamma}}{(1 - \frac{2\gamma}{3})H_{m,\gamma}} + 1 - \frac{H_{m-\lambda,\gamma}}{H_{m,\gamma}}, & \text{otherwise}
\end{cases}$$

where $m - \lambda = (sn - m)(\frac{3}{2\gamma} - 1)$.

**Proof.** Based on Lemma 7, when $\gamma < \frac{3}{2}$, $\mu = 0$.

(i) $m < \left(1 - \frac{2\gamma}{3}\right)sn$

In this case, $\lambda$ is also 0. Then $p_i^* = \left(\frac{\rho_i}{2n\lambda^{\gamma}}\right)^{\frac{2}{3}}$ for all $i = 1, ..., m$, and the optimal value of the primal problem is

$$L^* = \sum_{i=1}^{m} \left(\frac{1}{\sqrt{p_i} sn} - \frac{1}{\sqrt{n}}\right) \rho_i = \left(\frac{2u^*}{sn}\right)^{\frac{1}{3}} \left(\frac{H_{m,\gamma}}{H_{m,\gamma}}\right)^{\frac{2}{3}} - \frac{1}{\sqrt{n}}. \quad (3.22)$$

As $\lambda = \mu = 0$, Eq. (3.14) becomes

$$\left(\frac{1}{2u^*\sqrt{sn}}\right)^{\frac{2}{3}} \sum_{i=1}^{m} \left(\frac{1}{H_{m,\gamma}}\right)^{\frac{2}{3}} = 1. \quad (3.23)$$

Combining Eq. (3.22) and Eq. (3.23), we can get the optimal value of the primal problem is

$$L^* = \frac{1}{\sqrt{sn}} \left(\frac{H_{m,\gamma}}{H_{m,\gamma}}\right)^{\frac{3}{2}} - \frac{1}{\sqrt{n}} = \frac{1}{\sqrt{sn}} \left(\frac{H_{m,\gamma}}{H_{m,\gamma}}\right)^{\frac{2}{3}}.$$ 

Here $\frac{1}{\sqrt{n}}$ can be ignored as $\left(\frac{H_{m,\gamma}}{H_{m,\gamma}}\right)^{\frac{2}{3}} \gg 1$.

(ii) $m \geq \left(1 - \frac{2\gamma}{3}\right)sn$
When $m \geq (1 - \frac{2\gamma}{3})sn$, according to Lemma 8, $\lambda \neq 0$. Then, the optimal value is

$$L^* = \sum_{i=1}^{m-\lambda} \left(1 - \frac{1}{\sqrt{p_i^* sn}} - \frac{1}{\sqrt{n}}\right) \rho_i + \sum_{i=m-\lambda+1}^{m} \left(1 - \frac{1}{\sqrt{n}}\right) \rho_i$$

$$= \left(\frac{2u^*}{sn}\right)^{\frac{1}{3}} \frac{H_{m-\lambda, \frac{2\gamma}{3}}}{(H_{m, \gamma})^{\frac{2}{3}}} + \frac{H_{m, \gamma} - H_{m-\lambda, \gamma}}{H_{m, \gamma}} - \frac{1}{\sqrt{n}}$$

$$= \frac{(m - \lambda)^{1-\gamma}}{(1 - \frac{2\gamma}{3})H_{m, \gamma}} + 1 - \frac{H_{m-\lambda, \gamma}}{H_{m, \gamma}},$$

where the last step of the equality is due to $m - \lambda \approx \left(\frac{sn}{2u^* H_{m, \gamma}}\right)^{\frac{1}{3}}$ in Eq. (3.18), and $H_{m-\lambda, \frac{2\gamma}{3}} \approx \frac{3}{3-2\gamma}(m - \lambda)^{1-\frac{2\gamma}{3}}$ in Eq. (3.2). The remaining task is to find the actual value of $m - \lambda$. Since $\mu = 0$ and $\lambda \neq 0$, Eq. (3.14) becomes

$$\left(\frac{1}{2u^* \sqrt{sn}}\right)^{\frac{2}{3}} \sum_{i=1}^{m-\lambda} \left(\frac{1}{H_{m, \gamma}}\right)^{\frac{2}{3}} + \frac{\lambda}{sn} = 1.$$

As $m - \lambda \approx \left(\frac{sn}{2u^* H_{m, \gamma}}\right)^{\frac{1}{3}}$ and $H_{m-\lambda, \frac{2\gamma}{3}} \approx \frac{3}{3-2\gamma}(m - \lambda)^{1-\frac{2\gamma}{3}}$, the above Eq. (3.25) leads to

$$m - \lambda = (sn - m) \left(\frac{3}{2\gamma} - 1\right).$$

$\square$

### 3.5.2.2 $\gamma = \frac{3}{2}$

Similar to Proposition 1, we have the following:

**Proposition 2.** When $\gamma = \frac{3}{2}$, a lower bound on the average transmission distance is

$$L^* = \begin{cases} 
\frac{1}{\sqrt{sn}} \frac{(H_{m, 1})^{\frac{3}{2}}}{H_{m, \frac{3}{2}}}, & \text{if } m \ln m < ns \\
\frac{2(\kappa^{-\frac{1}{2}} - m^{-\frac{1}{2}}) + \kappa^{-\frac{1}{2}} \ln \kappa}{H_{m, \frac{3}{2}}}, & \text{otherwise}
\end{cases}$$

where $\kappa$ satisfies $\kappa(\ln \kappa - 1) = sn - m$.

**Proof.** (i) $m \ln m < ns$

When $m \ln m < ns$, $\lambda = 0$. Then $p_i^* = \left(\frac{\rho_i}{2u^* \sqrt{sn}}\right)^{\frac{3}{2}}$ for all $i = 1, ..., m$. Similar to
Eq. (3.22), we can get the optimal value of the primal problem is

\[ L^* = \frac{1}{\sqrt{sn}} \frac{(H_{m,\frac{3}{2}})^{\frac{3}{2}}}{H_{m,\gamma}} = \frac{1}{\sqrt{sn}} \frac{(H_{m,1})^{\frac{3}{2}}}{H_{m,\frac{3}{2}}} . \]

(ii) \( m \ln m \geq ns \)

In this case, based on Eq. (3.18), we have \( m - \lambda \approx \left( \frac{sn}{2u^2 H_{m,\gamma}} \right)^{\frac{1}{2}} \). Let \( \kappa = \left( \frac{sn}{2u^2 H_{m,\gamma}} \right)^{\frac{1}{2}} \), then Eq. (3.14) leads to

\[ \frac{\kappa}{sn} \left( H_{m,\frac{3}{2}} \right)^{\frac{2}{3}} \frac{H_{H_{m,1}}}{(H_{m,\gamma})^{\frac{2}{3}}} + \frac{m}{sn} - \frac{\kappa}{sn} = 1. \quad (3.26) \]

After simplifying the above equation, we can get

\[ \kappa (\ln \kappa - 1) = sn - m. \]

Since \( m - \lambda \approx \kappa \), by combining the above equation with Eq. (3.24), the optimal value of the primal problem is

\[ L^* = \left( \frac{1}{\kappa^{\frac{3}{2}} H_{m,\frac{3}{2}}} \right)^{\frac{1}{2}} \ln \kappa \left( H_{m,\frac{3}{2}} \right)^{\frac{2}{3}} + \frac{\kappa^{-\frac{1}{2}} - m^{-\frac{1}{2}}}{(H_{m,\frac{3}{2}})^{\frac{2}{3}}} \frac{1}{H_{m,\frac{3}{2}}} \]

\[ = \frac{\kappa^{-\frac{1}{2}} \ln \kappa + 2 \left( \kappa^{-\frac{1}{2}} - m^{-\frac{1}{2}} \right)}{H_{m,\frac{3}{2}}}. \]

\[ 3.5.2.3 \quad \gamma > \frac{3}{2} \]

Before the derivation of \( L^* \), we present the conditions for \( \mu = 0 \) or \( \lambda = 0 \) in the following lemma.
Lemma 9. When $\gamma > \frac{3}{2}$,

\[
\begin{align*}
\mu &= 0 \text{ and } \lambda = 0, \quad \text{if } s < \chi_1 \text{ and } sn > \chi_2 \\
\mu &= 0 \text{ and } \lambda \neq 0, \quad \text{if } s < \chi_3 \text{ and } sn \leq \chi_2 \\
\mu &\neq 0 \text{ and } \lambda \neq 0, \quad \text{if } s \geq \chi_3 \text{ and } sn \leq \chi_4 \\
\mu &\neq 0 \text{ and } \lambda = 0, \quad \text{if } s \geq \chi_1 \text{ and } sn > \chi_4
\end{align*}
\]

where $\chi_1 = H_{m, \frac{2\gamma}{3}}$, $\chi_2 = m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$, $\chi_3 = \frac{m}{n} + \frac{2\gamma}{2\gamma - 3}$ and $\chi_4 = \frac{2\gamma}{2\gamma - 3} \frac{m}{n\frac{2\gamma - 1}{2\gamma}} + m$.

Proof. (i) $\mu = \lambda = 0$

When $\mu = 0$, we have $p_1^* = \left(\frac{\rho_1}{2u_1\sqrt{sn}}\right)^{\frac{3}{2}} < \frac{1}{s}$. Combining this inequality with Eq. (3.23) leads to $s < H_{m, \frac{2\gamma}{3}}$. When $\lambda = 0$, we have $p_m^* = \left(\frac{\rho_m}{2u_m\sqrt{sn}}\right)^{\frac{3}{2}} > \frac{1}{sn}$. By combining this inequality with Eq. (3.23), we can get $sn > m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$.

(ii) $\mu = 0$ and $\lambda \neq 0$

First, note that when $s < H_{m, \frac{2\gamma}{3}}$ and $sn \leq m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$, $\mu = 0$ and $\lambda \neq 0$ hold. Second, when $s \geq H_{m, \frac{2\gamma}{3}}$ and $sn \leq m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$, if the solution to the primal problem satisfies $\mu = 0$ ($s < \frac{m}{n} + \frac{2\gamma}{2\gamma - 3}$), then $\mu = 0$ and $\lambda \neq 0$ also hold.

Combining the above two feasible regions(i.e., combining $s < H_{m, \frac{2\gamma}{3}}$ and $sn \leq m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$ with $H_{m, \frac{2\gamma}{3}} \leq s < \frac{m}{n} + \frac{2\gamma}{2\gamma - 3}$ and $sn \leq m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$), we can get the condition for $\mu = 0$ and $\lambda \neq 0$ is

\[
s < \frac{m}{n} + \frac{2\gamma}{2\gamma - 3}, \text{ and } sn \leq m \frac{3\gamma}{2} H_{m, \frac{2\gamma}{3}}.
\]

(iii) $\mu \neq 0$ and $\lambda \neq 0$

To achieve that $\mu \geq 1$ in the solution of the primal problem, we have $s \geq H_{m, \frac{2\gamma}{3}}$. Similarly $\lambda \geq 1$ leads to $sn \leq m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$.

(iv) $\mu \neq 0$ and $\lambda = 0$

The proof here is similar to the proof of case (ii) (i.e., $\mu = 0$ and $\lambda \neq 0$). First, when $s \geq H_{m, \frac{2\gamma}{3}}$ and $sn > m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$, $\mu \neq 0$ and $\lambda = 0$ hold. Second, when $s \geq H_{m, \frac{2\gamma}{3}}$ and $sn \leq m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$, for $\lambda = 0$ to hold, $sn$ must be greater than $\frac{2\gamma}{2\gamma - 3} \frac{m}{n\frac{2\gamma - 1}{2\gamma}} + m$. 


Combining the above two feasible regions, the condition for $\mu \neq 0$ and $\lambda = 0$ is

$$s \geq H_{m, \frac{2\gamma}{3}} \text{ and } sn > \frac{2\gamma}{2\gamma - 3} \frac{m}{n^{\frac{2\gamma}{3} - 1}} + m.$$ 

Based on the above lemma, we now derive the optimal value of the primal problem, when $\gamma > \frac{3}{2}$.

**Proposition 3.** When $\gamma > \frac{3}{2}$, a lower bound on the average transmission distance is

$$L^* = \begin{cases} 
\frac{1}{\sqrt{sn}} \frac{(H_{m, \frac{2\gamma}{3}})}{H_{m, \gamma}}^{\frac{3}{2}}, & \text{if } s < \chi_1 \text{ and } sn > \chi_2 \\
\frac{1}{\sqrt{sn - m}} \frac{(H_{m, \frac{2\gamma}{3}})}{H_{m, \gamma}}^{\frac{3}{2}}, & \text{if } s < \chi_3 \text{ and } sn \leq \chi_2 \\
\frac{(2\gamma - 3)}{2\gamma} \left(\frac{s - m}{n}\right)^{1-\gamma} \sqrt{nH_{m, \gamma}(\frac{2\gamma}{3} - 1)}, & \text{if } s \geq \chi_3 \text{ and } sn \leq \chi_4 \\
\frac{(2\gamma - 3)}{2\gamma} \left(\frac{s}{\chi_1^{\gamma}}\right)^{1-\gamma} \sqrt{nH_{m, \gamma}(\frac{2\gamma}{3} - 1)}, & \text{if } s \geq \chi_1 \text{ and } sn > \chi_4
\end{cases}$$

where $\chi_1 = H_{m, \frac{2\gamma}{3}}$, $\chi_2 = m \frac{2\gamma}{3} H_{m, \frac{2\gamma}{3}}$, $\chi_3 = \frac{m}{n} + \frac{2\gamma}{2\gamma - 3}$ and $\chi_4 = \frac{2\gamma}{2\gamma - 3} \frac{m}{n^{\frac{2\gamma}{3}}} + m$.

**Proof.**

(i) **$\mu = \lambda = 0$**

When both $\mu$ and $\lambda$ are 0, similar to Eq. (3.22), we have

$$L^* = \frac{1}{\sqrt{sn}} \frac{(H_{m, \frac{2\gamma}{3}})}{H_{m, \gamma}}^{\frac{3}{2}}.$$

(ii) **$\mu = 0$ and $\lambda \neq 0$**

In this case, based on Eq. (3.24) and Eq. (3.18), we can get

$$H_{m, \frac{2\gamma}{3}}(m - \lambda)^{\frac{2\gamma}{3}} - \frac{2\gamma}{2\gamma - 3} (m - \lambda) = 1.$$ 

Combining the above equality with the objective of the primal problem in Eq.
(3.24), the optimal value of the primal problem is
\[
L^* = \frac{1}{\sqrt{sn - m}} \left( \frac{H_{\frac{m}{\gamma}}}{H_{m,\gamma}} \right)^{\frac{2}{3}}.
\]

(iii) \( \mu \neq 0 \) and \( \lambda \neq 0 \)

If both \( \mu \) and \( \lambda \) are larger than 0, combining Eq. (3.16), Eq. (3.18) and Eq. (3.14) leads to
\[
\frac{\mu}{s} + \frac{\mu - (m - \lambda)n^{-1}}{s(\frac{2}{3}\gamma - 1)} + \frac{\lambda}{sn} = 1.
\]
Based on Eq. (3.16) and Eq. (3.18), \( \frac{m-\lambda}{\mu} = n^{\frac{3}{2\gamma}} \), then the value of \( \mu \) and \( \lambda \) are given by the following equation:
\[
\begin{cases}
\mu = \frac{2\gamma - 3}{2\gamma} \left( s - \frac{m}{n} \right) \\
\lambda = \frac{2\gamma - 3}{2\gamma} \left( s - \frac{m}{n} \right) n^{\frac{3}{2\gamma}}
\end{cases}
\]
Take the value of \( \lambda \) and \( \mu \) to the objective function of the primal problem yields the results in the proposition.

(iv) \( \mu \neq 0 \) and \( \lambda = 0 \)

When \( \mu \neq 0 \) and \( \lambda = 0 \), Eq. (3.14) is equivalent to
\[
\frac{\mu}{s} + \left( \frac{1}{2u^*\sqrt{sn}} \right)^{\frac{2}{3}} \sum_{i=\mu+1}^{m} \rho_i^{\frac{2}{3}} = 1.
\]
Combining above equation with Eq. (3.16), we can get
\[
\mu = s \frac{2\gamma - 3}{2\gamma}.
\]
Take the value of \( \mu \) in the above equation to the objective function of the primal problem, we can get
\[
L^* = \frac{\left( \frac{2\gamma - 3}{2\gamma} s \right)^{1-\gamma}}{\sqrt{n}H_{m,\gamma}(\frac{2}{3}\gamma - 1)}.
\]
3.5.3 Capacity Scaling Laws

Based on previous results on the average transmission distance, we now present an upper bound on the per-node capacity. Recall that given the average transmission distance $L$, the per node capacity is upper bounded by $\sqrt{\frac{8}{\pi}} \frac{W}{\beta} \frac{1}{L^{\frac{1}{\gamma}}}$. Combining this result with the lower bounds on average transmission distance given in propositions 1, 2 and 3, we have the following theorem.

**Theorem 3.** Under our network and content access model, an upper bound $C$ on the per-node capacity (with constants $\sqrt{\frac{8}{\pi}} \frac{W}{\beta} \frac{1}{L^{\frac{1}{\gamma}}}$ omitted) is given as follows.

(i) $\gamma < \frac{3}{2}$

If $m < (1 - \frac{2}{3} \gamma)sn$:

$$C = \begin{cases} \frac{\sqrt{s}(1 - \frac{2}{3} \gamma)^{\frac{3}{2}}}{m^{1-\gamma}}, & \text{if } \gamma < 1 \\ 3^{-\frac{3}{2}} \frac{\sqrt{s}}{m} \ln m, & \text{if } \gamma = 1 \\ H_{m,\gamma} \frac{\sqrt{s}}{m^{\frac{3}{2}-\gamma}} \left(1 - \frac{2}{3} \gamma\right)^{\frac{3}{2}}, & \text{if } 1 < \gamma < \frac{3}{2} \end{cases}$$

If $m \geq (1 - \frac{2}{3} \gamma)sn$:

$$C = \begin{cases} \frac{1}{\sqrt{n}} \ln \frac{2m}{sn-m} + 3, & \text{if } \gamma < 1 \\ \frac{1}{\sqrt{n}} \ln \frac{m}{\frac{2m}{sn-m} - 1} - \frac{1}{\gamma} \left(\frac{sn}{m} - 1\right)^{1-\gamma} - 1, & \text{if } 1 < \gamma < \frac{3}{2} \end{cases}$$

(ii) $\gamma = \frac{3}{2}$

$$C = \begin{cases} \frac{\sqrt{s}}{(\ln m)^{\frac{3}{2}}} H_{m,\frac{3}{2}}, & \text{if } m \ln m < sn \\ \frac{1}{\sqrt{n}} \left(\frac{3}{2\gamma} - 1\right)^{\gamma} \left(\frac{sn}{m} - 1\right)^{1-\gamma} - 1, & \text{if } m \ln m \geq sn \end{cases}$$

where $\kappa$ satisfies $\kappa (\ln \kappa - 1) = sn - m$. 
(iii) $\gamma > \frac{3}{2}$

$$C = \begin{cases} 
\sqrt{s} \frac{H_{m,\gamma}}{(H_{m,\frac{2\gamma}{2}})^{\frac{3}{2}}}, & \text{if } s < \chi_1 \text{ and } sn > \chi_2 \\
\sqrt{s - \frac{m}{n}} \frac{H_{m,\gamma}}{(H_{m,\frac{2\gamma}{2}})^{\frac{3}{2}}}, & \text{if } s < \chi_3 \text{ and } sn \leq \chi_2 \\
\frac{2}{3} \left( s - \frac{m}{n} \right)^{\gamma-1} \frac{\gamma H_{m,\gamma}}{(2^{\frac{\gamma-3}{2\gamma}})^{-\gamma}}, & \text{if } s \geq \chi_3 \text{ and } sn \leq \chi_4 \\
\frac{2}{3} s^{\gamma-1} \frac{\gamma H_{m,\gamma}}{(2^{\frac{\gamma-3}{2\gamma}})^{-\gamma}}, & \text{if } s \geq \chi_1 \text{ and } sn > \chi_4 
\end{cases}$$

where the values of $\chi_1$, $\chi_2$, $\chi_3$ and $\chi_4$ are given in Proposition 3.

**Proof.** Recall $p_i$ is the density of content $i$, then $1 - sp_i$ is the fraction of nodes that have not cached content $i$ locally. Let $L_i$ represent the average distance for those nodes to retrieve $i$ from others. Then, the average distance for nodes to retrieve contents from others (averaged over all contents), denoted by $L$, is

$$L = \frac{\sum_{i=1}^{m} \rho_i (1 - sp_i) L_i}{\sum_{i=1}^{m} \rho_i (1 - sp_i)}.$$

According to [1], the number of bits each node can receive from others per second is upper bounded by $\frac{1}{L^{\sqrt{n}}}$. The cache hit ratio (i.e., the probability that a request is served by local cache) is $P = \sum_{i=1}^{m} \rho_i sp_i$. Then, besides the bits received from others, $\frac{P}{1-P} \frac{1}{L^{\sqrt{n}}}$ bits from local cache are also used to serve requests. Combining the bits from others and the bits from local cache, we can get an upper bound on per-node capacity as follows:

$$C = \frac{1}{L^{\sqrt{n}}} \cdot \left( 1 + \frac{P}{1-P} \right) = \frac{1}{\sqrt{n} \sum_{i=1}^{m} \rho_i (1 - sp_i) L_i}. \quad (3.27)$$

Recall that $L_i$ is the transmission distance of content $i$ averaged over all nodes (including the nodes that have cached $i$ locally), then $L_i = L_i \cdot (1 - sp_i) + 0 \cdot sp_i$, for $i = 1, ..., m$. The transmission distance averaged over all contents is $L =$
\[ \sum_{i=1}^m L_i \rho_i = \sum_{i=1}^m L_i (1 - sp_i) \rho_i. \] Then, Eq. (3.27) can be rewritten as
\[ C = \frac{1}{L \sqrt{n}}. \]

Combining the above equation with propositions 1, 2 and 3 yields the results in the theorem.

\[ \square \]

### 3.6 Analytical Results

#### 3.6.1 Capacity and the Number of Nodes

In this subsection, we analyze how the capacity of wireless networks with caching changes with the number of nodes. According to Theorem 3, given a specific \( \gamma \), the capacity result \( C \) is a piecewise function based on \( n, s \) and \( m \). We first show \( C \) is continuous for any \( \gamma > 0 \).

**Lemma 10.** For any \( \gamma > 0 \), the per-node capacity function \( C \) is continuous on \( n \geq \frac{m}{s} \).

**Proof.** To show that \( C \) is continuous, we only need to show \( C \) is continuous at the critical points (i.e., the point that connects two sub-functions of the piecewise function).

(i) \( \gamma < \frac{3}{2} \)

In this case, there is only one critical point \( n_0 = \frac{m}{s(1 - \frac{2\gamma}{3})} \). The comparisons of the right-hand limit and the actual value of the capacity at the critical point for different \( \gamma \) are given as follows:

\[
\lim_{n \to n_0^-} C = \sqrt{s} \frac{\left(1 - \frac{2\gamma}{3}\right)^{\frac{3}{2}}}{m} \frac{1}{1 - \gamma} = \frac{1}{\sqrt{n_0}} \frac{1}{1 - \frac{2\gamma}{3}}, \quad \gamma < 1
\]

\[
\lim_{n \to n_0^-} C = 3^{-\frac{3}{2}} \sqrt{s} m \ln m = \sqrt{\frac{s}{3m \ln \frac{2m}{3m - m + 3}}}, \quad \gamma = 1
\]

\[
\lim_{n \to n_0^-} C = \frac{H_{m, \gamma} \sqrt{s}}{m^{\frac{1}{2} - \gamma}} \left(1 - \frac{2\gamma}{3}\right)^{\frac{3}{2}} = \frac{1 - \frac{2\gamma}{3}}{\sqrt{n_0}} H_{m, \gamma} \frac{m^{1 - \gamma}}{m^{1 - \gamma}}, \quad \gamma > 1.
\]

(ii) \( \gamma = \frac{3}{2} \)
At the only critical point \( n_0 = \frac{\ln m}{s} \), we have
\[
\lim_{n \to n_0^+} C = \frac{\sqrt{s}H_{m,\frac{3}{2}}}{(\ln m)^{\frac{3}{2}}} = \frac{\sqrt{s}}{\sqrt{m \ln m}} \frac{H_{m,\frac{3}{2}}}{m^{\frac{3}{2} \ln m}}.
\]

(iii) \( \gamma > \frac{3}{2} \)

There are two critical points. At the first critical point \( n_0 = \frac{m^{\frac{2}{3}H_m}}{s^{\frac{2}{3}}}, \) we have \( m \ll n_0 \), and the right-hand limit and the actual value are equal
\[
\lim_{n \to n_0^+} C = \frac{\sqrt{s}H_{m,\gamma}}{\left(H_{m,\frac{2}{3}}\right)^{\frac{3}{2}}} = \lim_{n \to 0} \frac{\sqrt{s - \frac{m}{n}H_{m,\gamma}}}{\left(H_{m,\frac{2}{3}}\right)^{\frac{3}{2}}}.\]

The other critical point denoted by \( n'_0 \), satisfies \( n'_0 = \frac{m}{s} (1 + \frac{2\gamma}{2\gamma - 3} n_0^{1-\frac{2}{\gamma}}) \). At \( n'_0 \), we have \( m \ll n'_0 \) and the right-hand limit and the actual value are equal
\[
\lim_{n \to n'_0^+} C = \frac{2\gamma s^{\gamma-1}H_{m,\gamma}}{(2\gamma-3)^{-\gamma}} = \lim_{n \to n'_0^+} \frac{2\gamma (s - \frac{m}{n})^{\gamma-1}H_{m,\gamma}}{(2\gamma-3)^{-\gamma}}.
\]

Since in all three cases, the right-hand limit and the actual value are always equal at the critical points, \( C \) is continuous.

Regarding the monotonicity of the capacity \( C \), we have the following lemma.

**Lemma 11.** For any \( \gamma > 0 \), the capacity \( C \) monotonically increases with the number of nodes \( n \).

**Proof.** We have shown in Lemma 10 that \( C \) is continuous, to show \( C \) monotonically increases, we just need to prove each sub-function of \( C \) monotonically increases with \( n \).

(i) \( \gamma < 1 \)

When \( m < (1 - \frac{2}{3} \gamma) s n \), \( C \) is independent of \( n \). We just need to prove \( C \) monotonically increases with \( n \) when \( m \geq (1 - \frac{2}{3} \gamma) s n \). Let \( t = \sqrt{m(1 - \frac{1}{2}(\frac{3}{2\gamma - 3})^{-\gamma}(\frac{m}{n} - 1)^{1-\gamma})} \), which is the denominator of \( C \). Then
\[
\frac{dt}{dn} = \frac{n^{-\frac{1}{2}}}{4 t^2},
\] (3.28)
where \( \tilde{t} = \frac{1}{2} - \frac{1}{4}((\frac{3}{2\gamma} - 1)(\frac{sn}{m} - 1))^{-\gamma}(\frac{sn}{m} - 1 + \frac{2sn}{m}(1 - \gamma)) \). Let \( x = (\frac{sn}{m} - 1) \), then \( 0 \leq x = \frac{sn}{m} - 1 \leq \frac{2\gamma}{3-2\gamma} \). The derivative of \( \tilde{t} \) over \( x \) is

\[
\frac{d\tilde{t}}{dx} = -\left(3\left(\frac{2\gamma}{3} - 1\right)\right)^{-\gamma}(1 - \gamma)((3 - 2\gamma)x - 2\gamma)x^{-\gamma - 1} \geq 0.
\]

As \( \frac{dt}{dx} \geq 0 \), \( \tilde{t} \) monotonically increases with \( x \), and the maximum value of \( \tilde{t} \) is 0 when \( x = \frac{2\gamma}{3-2\gamma} \). Combining \( \tilde{t} \leq 0 \) with Eq. (3.28) leads to \( \frac{dt}{dn} \leq 0 \). Recall \( C = \frac{1}{t} \), then \( \frac{dC}{dn} \geq 0 \), and we have \( C \) monotonically increases with \( n \).

(ii) \( \gamma = 1 \)

When \( \gamma = 1 \) and \( m < (1 - \frac{2}{3}\gamma)sn \), the capacity \( C \) is independent of \( n \). Hence, to show \( C \) monotonically increases, we only need to prove \( C \) increases with \( n \) when \( m \geq (1 - \frac{2}{3}\gamma)sn \). Let \( t \) denote the denominator of \( C \), \( t = \sqrt{n}(ln \frac{2m}{sn - m} + 3) \). Then

\[
\frac{dt}{dn} = \frac{1}{2}n^{-\frac{1}{2}}\left(ln \frac{2m}{sn - m} + 3\right) - \frac{\sqrt{ns}}{sn - m}
= \frac{1}{2\sqrt{n}}\left(ln \frac{2m}{sn - m} + 1 - \frac{2m}{sn - m}\right).
\]

As \( m \geq \frac{1}{3}sn \), \( \frac{2m}{sn - m} \geq 1 \). Also note \( x - \ln x \geq 1 \) holds for all \( x \geq 1 \), thus \( \frac{dt}{dn} \leq 0 \). In this way, \( t \) monotonically decreases with \( n \), and \( C \) increases with \( n \).

(iii) \( 1 < \gamma < \frac{3}{2} \)

When \( m < (1 - \frac{2}{3}\gamma)sn \), \( C \) is independent of \( n \). When \( m \geq (1 - \frac{2}{3}\gamma)sn \), the numerator of \( C \) is independent of \( n \), hence we only focus on the denominator of \( C \), and let \( t = \sqrt{n}(\frac{1}{2}(\frac{3}{2\gamma - 1})^{-\gamma}(\frac{sn}{m} - 1)^{1 - \gamma} - 1) \). The proof here is similar to the proof of case \( \gamma < 1 \), as \( t \) in both cases only differ by a negative sign. Consequently, the only difference in the proof is when showing \( \frac{dt}{dx} \geq 0 \). Previously we have \( 1 - \gamma > 0 \) for \( \gamma < 1 \), and now we have \(- (1 - \gamma) > 0 \) for \( 1 < \gamma < \frac{3}{2} \).

(iv) \( \gamma = \frac{3}{2} \)

When \( sn > m \ln m \), the capacity \( C \) is independent of \( n \). For \( sn \leq m \ln m \), let \( t \)
denote the denominator of $C$, then

$$\frac{dt}{dn} = \frac{1}{\sqrt{n}} \left( (\kappa^{-\frac{1}{2}} - m^{-\frac{1}{2}}) + \frac{\kappa^{-\frac{1}{2}}}{2} \ln \kappa \right) - \sqrt{n} \left( \frac{\kappa^{-\frac{3}{2}}}{2} \ln \kappa \right) \frac{d\kappa}{dn}. \quad (3.29)$$

Recall that $\kappa$ satisfies $\kappa(\ln \kappa - 1) = sn - m$, then we have

$$\frac{d(\kappa \ln \kappa - \kappa)}{d\kappa} \frac{d\kappa}{dn} = \frac{d(sn - m)}{dn} \quad (3.30)$$

Combining Eq. (3.29) with Eq. (3.30) leads to

$$\frac{dt}{dn} = \frac{1}{2\sqrt{n}} \left( 2\kappa^{-\frac{1}{2}} - 2m^{-\frac{1}{2}} + \kappa^{-\frac{1}{2}} \ln \kappa - nsk^{-\frac{3}{2}} \right) = \frac{1}{2\sqrt{n}} \left( 2\kappa^{-\frac{1}{2}} - 2m^{-\frac{1}{2}} - m\kappa^{-\frac{3}{2}} \right).$$

As $sn \leq m \ln m$ and $\kappa(\ln \kappa - 1) = sn - m$, $\kappa \leq m$. Then the above derivative $\frac{dt}{dn} \leq 0$, $t$ decreases with $n$, and $C$ monotonically increases with $n$ when $\gamma = \frac{3}{2}$.

(v) $\gamma > \frac{3}{2}$

In this case, based on Theorem 3, the capacity $C$ which is a piecewise function has four sub-functions. Apparently, for the first and forth sub-function, the per-node capacity is independent of $n$. In the other two sub-functions, only the term $(s - \frac{m}{n})$ is affected by $n$, since $-\frac{m}{n}$ increases as $n$ increases, $C$ increases with $n$. □

This result suggests that the per-node capacity will not decrease when the number of nodes increases. More interestingly, in some cases, it is even possible for the per-node capacity to increase as $n$ increases.

3.6.2 Effects of Local Cache

In this subsection, we analyze the cache hit ratio $P$, i.e., the probability that a request is served locally, when contents have been optimally cached.
Lemma 12. When contents have been optimally cached to maximize the per-node capacity, for any $\gamma \leq \frac{3}{2}$, $P \rightarrow 0$.

Proof. Given $P^*$, the probability that requests being served by local cache is $P = \sum_{i=1}^{m} \rho_i p_i^*$. Based on Lemma 7, $\mu = 0$ for $\gamma \leq \frac{3}{2}$. As $\mu = 0$, Eq. (3.14) is

$$1 = \sum_{i=1}^{m-\lambda} \left( \frac{p_i}{2u^* \sqrt{sn}} \right)^\frac{2}{3} + \frac{\lambda}{sn} \geq \sum_{i=1}^{m} \left( \frac{p_i}{2u^* \sqrt{sn}} \right)^\frac{2}{3}. \quad (3.31)$$

Since $\sum_{i=1}^{m} \rho_i^\frac{2}{3} = \frac{H_{m, \frac{2}{3}}}{(H_{m, \gamma})^\frac{2}{3}}$, the above inequality (3.31) is equivalent to $\frac{(H_{m, \gamma})^\frac{2}{3}}{H_{m, \frac{2}{3}}} \geq \left( \frac{1}{2u^* \sqrt{sn}} \right)^\frac{2}{3}$. Combining the previous inequality and the value of $p_i^*$ given in Eq. (3.13), and let $\tilde{p}_i^* = \frac{(\rho_i H_{m, \gamma})^\frac{2}{3}}{H_{m, \frac{2}{3}}}$, we can get

$$\tilde{p}_1^* - p_1^* \geq \tilde{p}_2^* - p_2^* \geq \ldots \geq \tilde{p}_m^* - p_m^*. \quad (3.32)$$

Based on the definition of Zipf distribution, the popularity of contents satisfies $\rho_1 \geq \rho_2 \geq \ldots \geq \rho_m > 0$. Combining this content popularity inequality with inequality (3.32), and as $\sum_{i=1}^{m} \tilde{p}_i^* - p_i^* = 0$, we have $\sum_{i=1}^{m} \rho_i (\tilde{p}_i^* - p_i^*) s \geq 0$, which leads to:

$$\sum_{i=1}^{m} \rho_i \tilde{p}_i^* s \geq \sum_{i=1}^{m} \rho_i p_i^* s.$$

The left-hand side of the above inequality is

$$\sum_{i=1}^{m} \rho_i \tilde{p}_i^* s = \sum_{i=1}^{m} \frac{s}{\gamma H_{m, \gamma}} \frac{1}{H_{m, \frac{2}{3}}} = \frac{s H_{m, \frac{2}{3}}}{H_{m, \gamma} H_{m, \frac{2}{3}}}.$$

Since $\frac{s H_{m, \frac{2}{3}}}{H_{m, \gamma} H_{m, \frac{2}{3}}} \rightarrow 0$ for any $\gamma \leq \frac{3}{2}$, $\sum_{i=1}^{m} \rho_i p_i^* s = P \rightarrow 0$. \hfill $\square$

The above lemma states that when $\gamma \leq \frac{3}{2}$, only a small portion of requests are served locally, and the effect of the local cache is negligible. On the other hand, when $\gamma > \frac{3}{2}$, the content access is mainly focused on a few popular contents which may have been cached locally, and the local cache will have a more significant effect on per-node capacity. In the following lemma, we give two sufficient conditions for a constant portion of requests being served by local cache.
Lemma 13. When contents have been optimally cached to maximize the per-node capacity, for any $\gamma > \frac{3}{2}$, if $\mu \neq 0$ or $\mu = \lambda = 0$, there exists a constant $c$ independent of $m$, $s$ and $n$, such that $P \geq c$.

Proof. (i) $\mu \neq 0$ In case of $\mu \geq 1$, $p_1 = \frac{1}{s}$ and every node caches the most popular content locally. Then, requests for that content is always served by the local cache, and $P$ must satisfy

$$P > \rho_1 = \frac{1}{H_{m,\gamma}} > \frac{1}{1 + \frac{1}{\gamma - 1}} = \frac{\gamma - 1}{\gamma}.$$

(ii) $\mu = \lambda = 0$

When both $\lambda$ and $\mu$ are 0, we have

$$P = \sum_{i=1}^{m} \rho_i p_i^* s = \frac{sH_{m, \frac{5\gamma}{3}}}{H_{m, \gamma} H_{m, \frac{2\gamma}{3}}} \geq \frac{H_{m, \frac{5\gamma}{3}}}{H_{m, \gamma} H_{m, \frac{2\gamma}{3}}}.$$

As $m \to \infty$, $\frac{H_{m, \frac{5\gamma}{3}}}{H_{m, \gamma} H_{m, \frac{2\gamma}{3}}}$ converges to a constant which only depends on $\gamma$. Therefore, $P$ is larger than a constant when $\lambda = \mu = 0$. □

3.6.3 Influence of Other Parameters

Similar to the analysis in subsection 3.6.1, we now analyze how the cache size, the number of unique contents, and the Zipf parameter affect the per-node capacity. If we ignore the insignificant terms in the capacity results of Theorem 3, a capacity upper bound is roughly

1. $C = O\left(\sqrt{\frac{s}{m}}\right)$, if $\gamma < 1$.
2. $C = O\left(\sqrt{\frac{s}{m} \ln m}\right)$, if $\gamma = 1$.
3. $C = O\left(\frac{\sqrt{s}}{m^{\frac{3}{2} - \gamma}}\right)$, if $1 < \gamma < \frac{3}{2}$.
4. if $\gamma = \frac{3}{2}$,
   (a) $C = O\left(\frac{\sqrt{s}}{(\ln m)^{\frac{3}{2}}}\right)$, if $m \ln m < sn$;
   (b) $C = O\left(\frac{1}{\sqrt{n} \kappa^{\frac{1}{2} + 1/2 \ln \kappa}}\right)$, if $m \ln m \geq sn$, where $\kappa = \kappa - 1 = sn - m$. 
5. if $\gamma > \frac{3}{2}$

(a) $C = O\left(\sqrt{s - \frac{m}{n}}\right)$, if $s < H_{m,2\gamma/3}$;
(b) $C = O\left((s - \frac{m}{n})^{\gamma - 1}\right)$, if $s \geq H_{m,2\gamma/3}$.

We have the following observations. First, as cache size $s$ grows, the per-node capacity increases. This is because when the cache size increases, more requests can be served by local cache or nearby nodes. Our results also show that the capacity scales like $s^{\gamma - 1}$ with cache size $s$ when $\gamma > \frac{3}{2}$, and scales like $\sqrt{s}$ when $\gamma \leq \frac{3}{2}$. Thus, when $\gamma$ is large, increasing the cache size can significantly improve the per-node capacity.

Second, the per-node capacity decreases when the number of unique contents $m$ increases, since a larger $m$ reduces the probability for the requests to be served locally or by close neighbors. However, as $\gamma$ increases from 0 to more than $\frac{3}{2}$, $m$ has a smaller effect on per-node capacity. When $\gamma < 1$, $m$ is somewhat equally important as $s$, as $s$ needs to increase at the same speed with $m$ to keep the capacity constant. On the other hand, when $\gamma > \frac{3}{2}$, for some cases the capacity is almost irrelevant to $m$.

Third, $\gamma$ also plays an important role in per-node capacity. When $\gamma = 0$, contents follow uniform distribution. As $\gamma$ grows from 0 to more than $\frac{3}{2}$, the per-node capacity is increased drastically as the exponent of $m$ grows from $-\frac{1}{2}$ to 0. This result suggests that caching is more effective when contents have skewed popularity.

### 3.7 Caching Algorithms

In the previous sections, we have presented the optimal content densities, and derived an upper bound on the per-node capacity when contents have been optimally cached. In practice, the exact content popularity is usually unknown a priori. To show that the optimal content densities are achievable in real scenarios, we present a distributed algorithm which achieves the optimal content densities only based on each node’s local information.
3.7.1 Algorithm Description

According to Eq. (3.13), each of the most unpopular \( \lambda \) contents only has one copy in the network. On the other hand, each of the most popular \( \mu \) contents has \( n \) copies and is cached at all the nodes in the network. For any remaining content \( i \), its popularity is proportional to \( \rho_i^2 \), where \( \rho_i \) is the popularity of content \( i \).

To guarantee that all content requests can be served, a permanent copy for each content is randomly cached at some node in the network, before each node runs the distributed caching algorithm. To achieve the \( p_i \propto \rho_i^2 \) law for contents with moderate popularity, we use the GreedyDual-Size algorithm [45, 46]. The algorithm was originally proposed to reduce the access cost in web caching, and it had been proved to be \( k \)-competitive for web caching (where \( k \) is defined to be ratio of the cache size to the content size). However, we found that for caching in wireless networks, such an algorithm does not perform well especially when contents have very skewed popularity (i.e., \( \gamma \) is large). Thus, we propose a modified version of the GreedyDual-Size algorithm, so as to achieve the desired densities.

In our algorithm, each node maintains a value \( T(i) \) for any content \( i \) cached locally. Each node also maintains an “inflation” value \( V \), which is initially set to 0. For any content request, if the requested content \( i \) is cached locally, then \( T(i) \) is increased by \( l(i) \), where \( l(i) \) is the latest physical distance to retrieve content \( i \) from others. On the other hand, if content \( i \) is not in the local cache, \( i \) will be cached locally, and \( T(i) \) is set to \( V + l(i) \) (\( l(i) \) is the physical distance to retrieve \( i \)). For the value of \( V \), if the memory is full, \( V \) equals to the smallest \( T(j) \); otherwise, \( V \) equals to 0. Note that when the memory is full, to cache \( i \) locally, the content with the smallest \( T(j) \) will be evicted. Let \( S \) denote the set of contents cached locally, the Distributed Caching Algorithm running at each node is given as follows.

In practice, the value of \( l(i) \), i.e., the physical distance to retrieve a content, can be obtained based on the geographic coordinates of the source node and the destination node. Since each node only maintains \( T(i) \) and \( l(i) \) for contents cached locally, the space complexity of our algorithm is \( O(s) \). We also have the following theorem on the time complexity of the proposed algorithm.

**Theorem 4.** For the Distributed Caching Algorithm, the time complexity for processing each content request is \( O(\log s) \).
**Algorithm 1** Distributed Caching Algorithm

1: $V \leftarrow 0$
2: for content request for any content $i$ do
3: if $i$ is in local cache then
4: $T(i) \leftarrow T(i) + l(i)$ \> $l(i)$ is the latest physical distance to retrieve content $i$
5: else
6: if the memory is full then
7: $V \leftarrow \min\{T(j)|j \in S\}$
8: evict content $j$ such that $j = \arg\min_{k \in S} T(k)$
9: end if
10: cache $i$ locally, $T(i) \leftarrow V + l(i)$
11: end if
12: end for

Proof. In Line 3, it takes $O(1)$ time to determine whether a content has been cached locally, if a hash table is used. In Line 7, it takes $O(\log s)$ time to find the minimum $T(i)$, if the value $T(i)$ are stored in a min-heap. All the other operations inside the for loop take constant time. Hence, it takes $O(\log s)$ time to perform the operations inside the for loop (i.e., process each content request).

### 3.7.2 Theoretical Results

In this subsection, we prove that the Distributed Caching Algorithm can achieve the optimal content density $P^*$.

In the lemma below, we show that the content densities achieved by the Distributed Caching Algorithm are upper bounded by $\frac{1}{s}$ and lower bounded by $\frac{1}{sn}$.

**Lemma 14.** The Distributed Caching Algorithm achieves $\frac{1}{sn} \leq p_i \leq \frac{1}{s}$ for any content $i$, where $p_i$ is the density of content $i$.

Proof. Each content has at least one permanent copy in the network, and thus we have $p_i \geq \frac{1}{sn}$. On the other hand, each node caches at most one copy of each content locally, which leads to $p_i \leq \frac{1}{s}$.

Based on the above lemma, we present the content densities achieved by the Distributed Caching Algorithm in the following theorem.
Theorem 5. There exist two integers \( k \) and \( l \), and a positive constant \( u \), with \( 1 \leq k \leq l \leq m \), \( u \rho_k^\frac{2}{3} < \frac{1}{s} \leq u \rho_{k-1}^\frac{2}{3} \), and \( u \rho_{l+1}^\frac{2}{3} \leq \frac{1}{sn} < u \rho_l^\frac{2}{3} \), such that in the steady state, the content densities achieved by the Distributed Caching Algorithm satisfy the following equation:

\[
\begin{align*}
    p_i &= \frac{1}{s}, & \text{for } 1 \leq i < k \\
    p_i &= u \cdot \rho_i^\frac{2}{3}, & \text{for } k \leq i \leq l \\
    p_i &\sim \frac{1}{sn}, & \text{for } l < i \leq m
\end{align*}
\]

where \( \sum_{i=1}^n p_i = 1 \).

Proof. We first show that for any content \( i \) with moderate popularity (i.e., \( k \leq i \leq l \)), its density \( p_i \) is proportional to \( \rho_i^\frac{2}{3} \).

Consider a time period \( t \) in the steady state. For any node \( j \) and any content \( k \leq i \leq l \), let \( t_{i,j} \) denote the total time that \( j \) caches \( i \) locally. Since the content requests at each node are independent, in the steady state, the “inflation” value \( V \) at each node almost grows at constant speed.

Then, based on the Distributed Caching Algorithm, whenever content \( i \) is requested at node \( j \), it increases \( t_{i,j} \) by \( \theta l(i) \), where \( \theta \) is a constant. This is because \( T(i) \) is increased by \( l(i) \) when \( i \) is requested, and \( i \) is evicted when \( V = T(i) \).

Accordingly, we have that \( t_{i,j} \) is proportional to \( l(i) \), and it is proportional to the number of times \( i \) being requested during this time period. Recall that \( l(i) \) is the physical distance to retrieve \( i \), then we have \( l(i) \propto \frac{1}{\sqrt{p_i}} \). Besides, the number of times \( i \) being requested is proportional to its popularity and the length of the time period. Hence, we have

\[
t_{i,j} \propto \rho_i \cdot t \cdot \frac{1}{\sqrt{p_i}}. \tag{3.33}
\]

Let \( t_i = \sum_{j=1}^n t_{i,j} \), which is the total time that nodes in the network cache \( i \). As the proportional relationship specified in Eq. (3.33) holds for all the nodes in the network, we have

\[
t_i \propto \rho_i \cdot \frac{n}{\sqrt{p_i}} t. \tag{3.34}
\]

On the other hand, \( \frac{t_i}{T} \) gives the average number of copies of content \( i \) during
this time period, therefore we have \( \frac{t}{t} = nsp_i \). Combining this result with Eq. (3.34), we can get that

\[
\text{tnsp}_i \propto \rho_i \frac{n}{\sqrt{p_i}} t \\
\rho_i \propto \left( \frac{\rho_i}{s} \right)^{\frac{2}{3}}.
\]

Since the cache size \( s \) is a constant, the above equation yields the result that \( p_i \propto \rho_i^{\frac{2}{3}} \) for \( k \leq i \leq l \).

For \( 1 \leq i < k \) (i.e., the more popular contents), we have \( u \cdot \rho_i^{\frac{2}{3}} > \frac{1}{s} \). Based on Lemma 14, \( p_i \) is upper bounded by \( \frac{1}{s} \), hence the previous proof does not apply to the case of \( 1 \leq i < k \). When \( 1 \leq i < k \), content \( i \) is requested frequently and \( T(i) \) grows faster than \( V \). Then, \( T(i) \) is always greater than \( V \), and content \( i \) is always cached at all the nodes in the network. Accordingly, we have \( p_i = \frac{1}{s} \) for \( 1 \leq i < k \).

For \( l < i \leq m \) (i.e., the less popular contents), we have \( u \cdot \rho_i^{\frac{2}{3}} < \frac{1}{sn} \). Note that each content has a permanent copy, hence for \( l < i \leq m \), \( p_i \) is dominated by the term of \( \frac{1}{sn} \), and we have \( p_i \sim \frac{1}{sn} \).

When comparing the results in Theorem 5 and \( P^* \) in Eq. (3.13), it can be seen that for each of the most popular \( k - 1 \) (\( \mu \)) contents, the Distributed Caching Algorithm achieves the optimal density of \( \frac{1}{s} \); while for each of the most unpopular \( m - l \) (\( \lambda \)) contents, the proposed algorithm achieves the optimal density of \( \frac{1}{sn} \); for each of the remaining content, the proposed algorithm achieves the optimal density of \( u \cdot \rho_i^{\frac{2}{3}} \) (where \( u \) is a constant). Therefore, the proposed algorithm achieves the optimal content density \( P^* \).

Regarding the convergence of the algorithm, we have the following theorem.

**Theorem 6.** The Distributed Caching Algorithm converges to a steady state (i.e., the content densities in Theorem 5).

**Proof.** Consider the increasing rate of \( p_i \) in the steady state. For the partial derivative of \( p_i \) over time period \( t \), the increasing rate of \( p_i \) is proportional to the physical distance to retrieve content \( i \) (\( l(i) \)) and the frequency that content \( i \) is requested (i.e., \( n\rho_i \)), and the diminishing rate of \( p_i \) is proportional to the number of copies of \( i \). Hence, we have the following differential equation for \( p_i \):

\[
\frac{\partial p_i}{\partial t} = \phi n \rho_i \frac{1}{\sqrt{p_i}} - \psi snp_i,
\]
where $\phi$ and $\psi$ are constants. In the steady state, the content density is $p_i^*$, and $\frac{\partial p_i}{\partial t} = 0$, which leads to

$$\phi n \rho_i \frac{1}{\sqrt{p_i}} - \psi sp_i^* = 0.$$ 

For any $p_i > p_i^*$, $\frac{\partial p_i}{\partial t}$ will be

$$\frac{\partial p_i}{\partial t} = \phi n \rho_i \frac{1}{\sqrt{p_i}} - \psi sp_i < \phi n \rho_i \frac{1}{\sqrt{p_i}} - \psi sp_i^* = 0.$$ 

The above equation indicates that when the actual content density $p_i$ is larger than $p_i^*$, $p_i$ will decrease. On the other hand, for any $p_i < p_i^*$, $\frac{\partial p_i}{\partial t}$ will be

$$\frac{\partial p_i}{\partial t} = \phi n \rho_i \frac{1}{\sqrt{p_i}} - \psi sp_i > \phi n \rho_i \frac{1}{\sqrt{p_i}} - \psi sp_i^* = 0.$$ 

That is, when the actual content density $p_i$ is smaller than $p_i^*$, $p_i$ will increase. Hence, the algorithm converges to the steady state.

### 3.8 Evaluations

#### 3.8.1 Effects of Various Parameters

Based on above theoretical analysis, we now present some numerical results to illustrate how various parameters affect the capacity. Fig. 3.1(a) shows how the per-node capacity varies with the number of nodes ($n$) given different values of $\gamma$, where $s = 10$ and $m = 10^4$. As shown in the figure, the per-node capacity increases as $\gamma$ increases. When $\gamma$ is small, the content access is more like a uniform distribution (note that $\gamma = 0$ corresponds to the uniform distribution). When $\gamma$ is large, the content access is focused on some hot (frequently accessed) content which may have been cached, and then improving the per-node capacity. Under all five values of $\gamma$, the per-node capacity first increases with $n$, and then remains constant. This is different from Gupta-Kumar’s result where cache is not considered and the per-node capacity decreases with increasing $n$. With caching, the per-node capacity will not decrease as $n$ increases. Moreover, it shows that when $n$ is relatively small, it is possible that increasing $n$ will improve the per-node capacity. This is because
when $n$ is relatively small, most contents only have one replica, and nodes need to traverse the whole network to obtain the contents. On the other hand, when $n$ is relatively large, some of the contents could be cached by nearby nodes, which makes content retrieval easier and then improves the per-node capacity.

Fig. 3.1(b) illustrates how the number of unique contents $m$ affects the per-node capacity. Generally, increasing $m$ results in a reduction in per-node capacity. This is because with more unique contents, nodes will be more likely to retrieve contents from further away nodes, which leads to more interference and less capacity. More interestingly, when $\gamma$ increases, $m$ has a diminishing effect on capacity.

Figure 3.1: Effects of various parameters on capacity
For example, when $\gamma = 0$, the capacity decreases quickly with $m$; while for $\gamma = 2$, the capacity almost remains constant for small $m$. This is because the unpopular contents will be less frequently requested when $\gamma$ increases. When $\gamma$ is relatively large, increasing $m$ will only add a few extremely unpopular contents, and will hardly affect what contents will be cached and requested.

Fig. 3.1(c) shows how the cache size $s$ affects per-node capacity. As can be seen, as the cache size increases, the per-node capacity increases. When $\gamma$ is relatively small, the contents have similar popularity, and caching a few more contents will not result in a large increase in the network capacity. When $\gamma$ is relatively large, the popular contents are requested more frequently, and only caching a few more popular contents can significantly improve the per-node capacity.

Fig. 3.1(d) illustrates how the Zipf parameter $\gamma$ affects the per-node capacity. Compared to the three parameters $(s, m, n)$ discussed above, $\gamma$ has the largest impact on per-node capacity. As $\gamma$ grows from 0 to 2.5, the capacity dramatically increases from $10^{-3}$ to roughly $10^2$. Increasing $\gamma$ can significantly improve the capacity, since a larger $\gamma$ results in more skewed content popularity where a few popular contents are more frequently requested, which makes caching more effective.
Comparisons to Existing Work

In Fig. 3.2, we compare our capacity results with previous work [1, 15, 32] based on numerical results. From [1], for wireless networks without caching, the per-node capacity scales like $\frac{1}{\sqrt{n}}$, which is shown in the figure by the green dashed line. Grossglauser and Tse [15] have proved that when nodes are mobile, the per-node capacity can remain constant when $n$ grows, and their result is shown by the red dashed line in the figure. Qiu and Cao [32] have derived that the per-node capacity scales like $\sqrt{s/m}$ when the content popularity follows a uniform distribution (i.e., $\gamma = 0$), which is shown by the blue dashed line in the figure. Since our capacity result at $\gamma = 0$ (i.e., $C = \Theta(\sqrt{s/m})$) conforms to their result, this blue line also represents our capacity at $\gamma = 0$. The remaining two solid lines show our capacity results at $\gamma = 1.25$ and $\gamma = 2$, respectively.

In Fig. 3.2(a), we show how the per-node capacity changes with the number of nodes, where $s = 100$ and $m = 10^7$. As shown in the figure, when the number of nodes $n$ increases, the capacity result of Gupta-Kumar drops quickly, and the capacity of Grossglauser-Tse and Qiu-Cao remain unchanged. In our approaches, the per-node capacity will not decrease when $n$ increases. This is a significant improvement compared to the Gupta-Kumar’s result; that is, with caching, increasing the number of nodes will not reduce the per-node capacity.

Fig. 3.2(b) illustrates the per-node capacity as a function of the cache size,
where \( n = m = 10^6 \). Since Gupta-Kumar and Grossglauser-Tse do not consider caching, their results do not change with the cache size. When the cache size is extremely small \((s = \frac{m}{n})\), under all three values of \( \gamma \), our per-node capacity is comparable with Gupta-Kumar results. When \( s \) approaches \( \frac{m}{n} \), each content only has one replica in the network. Then, retrieving a content is like to communicate with a random node, which is identical to the communication scenario in [1].

As the cache size increases, the per-node capacity of our approach with various \( \gamma \) increases quickly, and significantly higher than Gupta-Kumar. When compared to Grossglauser-Tse, the result depends on \( \gamma \). When \( \gamma \) is large (i.e., \( \gamma = 2 \)), the content access is focused on some hot (frequently accessed) content which may have been cached, and then improving the per-node capacity. As a result, the per-node capacity of our approach with \( \gamma = 2 \) significantly outperforms that of Grossglauser-Tse. When \( \gamma \) is relatively small (i.e., \( \gamma = 0 \)), the content access is more like a uniform distribution and the caching advantage is not very high, and our per-node capacity is lower. However, Grossglauser-Tse has much longer delay since nodes have to wait until they move close to the destination. While compared to Qiu-Cao, our capacity conforms to theirs at \( \gamma = 0 \). As \( \gamma \) increases (e.g. \( \gamma = 1.25 \)), caching becomes more effective due to more skewed content popularity, and our capacity grows much higher than Qiu-Cao.

### 3.8.3 Evaluation of the Distributed Caching Algorithm

In this subsection, we evaluate the performance of our Distributed Caching Algorithm. In the simulations, \( n \) nodes are uniformly and independently placed on the surface of a unit sphere, where each node can cache \( s \) contents locally. There are \( m \) unique contents in the network, where the content popularity follows a Zipf distribution with parameter \( \gamma \). Suppose there is always a content request at each node. Nodes send requests sequentially to the nearest node that caches the requested content. After the content has been retrieved, it will be cached locally. If the memory is full, one content will be evicted based on the Distributed Caching Algorithm.

Fig. 3.3 plots content density as a function of content popularity, where \( n = 10000, m = 10000 \) and \( s = 10 \). In the figure, the blue line shows the simulations
results of the proposed algorithm, while the red line shows the theoretical optimal value. In Fig. 3.3(a), the Zipf parameter $\gamma$ is set to 0.8. As can be seen, popular contents have higher densities, and the simulation and theoretical results match quite well, except for several extremely popular contents. In Fig. 3.3(b), with $\gamma = 2$, the simulation results and the theoretical results are almost identical. Note that when $\gamma = 2$, some contents are accessed much more frequently than others, and both theoretical and simulation results show that the two most popular contents should almost be cached everywhere (i.e., with density close to $\frac{1}{s} = 0.1$), while for some unpopular contents, their densities are close to $\frac{1}{ns} = 1 \times 10^{-5}$. 

Figure 3.4: Results of the Distributed Caching Algorithm vs. theoretical value
Fig. 3.3 compares the average transmission distance in the simulations with the theoretical minimum transmission distance. The theoretical distance is shown by the solid line, while the distance based on simulations is shown by the dashed line. Recall that the per-node capacity scales like $O\left(\frac{1}{\sqrt{L}}\right)$ with the transmission distance $L$, hence, a smaller distance implies higher capacity. Fig. 3.3(a) shows the distance as a function of the number of nodes, where $\gamma$ ranges from 0.1 to 1.6, $m = 10000$, and $s = 5$. When $ns = m$ (i.e., $n = 2000$), both the theoretical approach and the proposed algorithm require that the each content has exactly one copy in the network. Therefore, the simulation results conform quite well to the theoretical results. As $n$ increases, the simulation results slightly deviate from the theoretical optimal value. One possible explanation is that the content density $p_i$ in the theoretical analysis can take any value between $\frac{1}{sn}$ and $\frac{1}{s}$, while in simulations $sp_i$ has to be a integer (since the number of copies has to be an integer). Thus, the proposed algorithm performs a bit worse than the theoretical value. Besides, as $\gamma$ grows, the transmission distance decreases.

Fig. 3.3(b) shows the effect of $m$ on the transmission distance, given different $\gamma$, when $n = 2500$ and $s = 10$. As the number of unique contents grows, the transmission distance increases, and then the per-node capacity decreases. This is because with more unique contents, it becomes less likely for the content requests to be served by local cache or nearby nodes.

Fig. 3.3(c) presents the transmission distance as a function of the cache size, given different $n$, when $m = 10000$ and $\gamma = 1.1$. When the cache size increases, more contents can be cached locally, and more requests can be served by nearby nodes, and the transmission distance decreases. Note that the result at the point of $n = 2000$ and $s = 25$, is identical to the result at the point of $n = 10000$ and $s = 5$, since the total cache size at the two points are the same (i.e., $ns = 50000$).

Fig. 3.3(d) shows how the transmission distance is affected by $\gamma$, given different $s$, when $n = 2000$ and $m = 10000$. Not surprisingly, the transmission distance decreases with $\gamma$ and $s$. When $\gamma = 2.6$, all three cases (i.e., $s = 10$, $s = 15$ and $s = 30$) perform similarly. This is because for large $\gamma$, increasing the cache size will only cache a few less popular contents, and may not significantly affect the system performance.


3.9 Conclusion

In this chapter, we have studied scaling laws of network capacity based on the skewness of content popularity. We found that as the distribution of the content popularity changes from uniform distribution to more skewed distributions, the network capacity quickly increases from $\Theta(\sqrt{\frac{s}{m}})$ to roughly $\Theta(\sqrt{s})$. Moreover, our results suggest that for wireless networks with caching, when contents have skewed popularity, increasing the number of nodes monotonically increases the per node capacity.
Chapter 4

Maintaining Social Links through Amplify-and-Forward

4.1 Introduction

In the past decade, many researchers have designed various routing algorithms for mobile ad hoc networks [47, 48, 49]. In sparse mobile ad hoc networks, where the node density is low, and contacts between the nodes in the network do not occur frequently, it is hard to maintain end-to-end connections. As a result, “carry-and-forward” is used, where mobile nodes physically carry the data, and forward the data when contacting a node with higher forwarding capability. To improve the performance of data forwarding, recent research [2, 3, 4, 5] focuses on exploiting social knowledge for data forwarding, because social knowledge is more reliable and less susceptible to the randomness of human mobility.

Although social cognitive techniques can be exploited to enhance the robustness and performance of mobile ad hoc networks, existing research focuses on improving the performance of data forwarding (or routing) for all nodes in the network, without differentiating which node pairs are more important. However, in many real scenarios, the communications between some nodes are more important than others. For example, in a battle field, for a platoon of soldiers consisting of three squads, the commander of the platoon should maintain good connections with the squad leaders, but not necessarily with all other soldiers. Then, it is more
important to maintain the social links between the commander and the squad leaders, even though this may be at the cost of sacrificing some communication performance with other nodes.

There are many research challenges on maintaining important social links in wireless ad hoc networks, especially when the nodes are far away from each other. Although multi-hop transmission can be used, if two nodes in the routing path are out of the transmission range, a network partition is possible. Increasing the transmission power level can solve part of the problem, but this approach has its limitations since there is always a maximum transmission range, out of which two nodes will not be able to communicate. Thus, even with the maximum transmission power level, it is possible that the social links cannot be maintained because two nodes along the routing path are far away from each other.

To address this problem, we adopt a cooperative amplify-and-forward strategy, where nodes around the node that cannot reach the next hop node (or link source for simplicity) cooperate to transmit towards the next node in the routing path (or link destination for simplicity). More specifically, the source first broadcasts the data to its nearby nodes, which simultaneously amplifies and forwards their received signal to the destination. In this way, the destination can receive a much stronger signal, from which it can decode and obtain the data from the source, and accordingly the social links can be maintained.

There is some existing work [50, 51, 52, 53, 54] on amplify-and-forward. However, most of them focused on maximizing the data throughput and optimizing the power allocation; i.e., given the total transmission power at the relay nodes, how to optimally allocate the power for each relay, so that the data throughput between the source and destination is maximized. Different from existing work, maintaining social link only requires that the data throughput of the social link is greater than a threshold. Furthermore, to save energy, relays should transmit at the lowest power level that can achieve the required link throughput. Due to these differences, existing solutions on amplify-and-forward can not be directly applied.

In this chapter, we study the problem of maintaining social links (i.e., link throughput larger than a threshold) while minimizing the power consumption of the relays, which is referred to as the Min-Energy problem. The difficulty of this problem lies in the fact that with many relays simultaneously transmitting
to the destination, it is not obvious how increasing the transmission power at one relay will affect the link throughput between the source and destination. We first formulate the Min-Energy problem as a non-convex quadratic programming problem by exploiting the rate-distortion theory, and then solve it based on an approximation technique called semidefinite relaxation (SDR) [55]. We also prove that our solution can minimize the power consumption of the relays.

In our solution to the Min-Energy problem, all relays are involved. This will generate significant synchronization overhead, since relays have to synchronize their clocks for their signals to arrive at the destination simultaneously. To minimize the synchronization overhead, we also study the Min-Relay problem, which aims to minimize the number of active relays while maintaining the social link. We formulate the Min-Relay problem as an integer programming problem, and propose a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. Evaluation results show that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.

The rest of the chapter is organized as follows. Section 4.2 presents the model and the problem formulation. We study the Min-Energy problem in Section 4.3 and the Min-Relay problem in Section 4.4. We present performance evaluations in Section 4.5. Section 4.6 reviews related work, and Section 4.7 concludes the chapter.

## 4.2 Preliminaries

### 4.2.1 Model

Consider a social link, where the source node $S$ wants to transmit data to a distant destination node $D$. As these two nodes are out of the normal wireless transmission range, node $S$ relies on $n$ other relays ($\{R_1, R_2, \ldots, R_n\}$) around it to help forward the data towards the destination. Let $d_{1,k}$ denote the distance between $S$ and any relay node $R_k$, and let $d_{2,k}$ denote the distance between $R_k$ and $D$.

Suppose the source node $S$ sends out a signal $s$ towards $D$, which follows a Gaussian distribution with zero-mean and unit variance. $S$ uses a fixed power $P$
for transmission, while each of the relays can at most transmit at power $Q$. We consider a Gaussian channel, where the received signal is the sum of the faded transmitted signal from all other nodes and additive white Gaussian noise. That is, the received signal $Y_j$ at node $j$ is given by:

$$Y_j = \sum_{i \in \tau} \frac{X_i}{d_{ij}} + W_j,$$

where $\tau$ is the set of nodes that are simultaneously transmitting, and $X_i$ is the signal transmitted by node $i$, and $d_{ij}$ is the distance between node $i$ and $j$, $r$ is a positive parameter describing how the signal strength scales with distance, and $W_j$ is the white noise at node $j$ following a normal distribution $\sim \mathcal{N}(0, N)$. To simplify the notations, let $\alpha_k$ denote the signal scaling coefficients from $S$ to the relay node $R_k$, i.e., $\alpha_k = 1/d_{1,k}^r$ for $k = 1, 2, \ldots, n$. Besides, let $\beta_k$ denote the signal scaling coefficient from the $R_k$ to $D$, which is given by $\beta_k = 1/d_{2,k}^r$, where $k = 1, 2, \ldots, n$.

In this chapter, we consider an amplify-and-forward strategy, which takes two steps to transmit the signal from the source node to the destination node. In the first step, the source node broadcasts the signal to all relay nodes, where the signal received at a relay node $R_i$ is $Z_i = s\alpha_i\sqrt{P} + W_i$.

In the second step, relay nodes amplify their received signal and forward to the destination. If $R_i$ employs power $P_i$ for transmission, the signal transmitted by $R_i$ is

$$X_i = \sqrt{\frac{P_i}{\alpha_i^2 P + N}} Z_i = \sqrt{\frac{P_i}{\alpha_i^2 P + N}} \left(s\alpha_i\sqrt{P} + W_i\right).$$

The signal at the destination node is the sum of the signals from all the relay nodes, which is

$$Y = \sum_{i=1}^{n} \beta_i X_i + W_{n+1} = \sum_{i=1}^{n} \beta_i \sqrt{\frac{P_i}{\alpha_i^2 P + N}} \left(s\alpha_i\sqrt{P} + W_i\right) + W_{n+1},$$

where $W_{n+1} \sim \mathcal{N}(0, N)$ is the white noise at the destination.
4.2.2 Link Throughput

In this subsection, we quantify how the transmission power at the relay node affects the link throughput between the source and destination.

Based on [56], for a Gaussian source (i.e., the signal that the source sends out follows a Gaussian distribution), the rate distortion function for a squared-error distortion measure is $R(D) = \frac{1}{2} \log \frac{P}{D}$. That is, when the source node transmits at power $P$ and the distortion at the destination node is $D$, the link throughput between the source and destination is at least $\frac{1}{2} \log \frac{P}{D}$. According to [50], the squared-error distortion between the original signal $s$ and the scaled signal received at the destination node is:

$$\tilde{D} = \frac{PN}{\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2 P + N}, \quad (4.1)$$

where $x_i = \beta_i \sqrt{\frac{P}{\sigma_i^2 P + N}}$. Note that it takes two time slots to transmit the signal under the amplify-and-forward strategy, hence the link throughput between the source and destination is at least

$$R(\tilde{D}) = \frac{1}{4} \log \frac{P}{\tilde{D}}, \quad (4.2)$$

where the distortion $\tilde{D}$ is given in Eq. (4.1).

4.2.3 Problem Formulation

In the **Min-Energy** problem, we seek to minimize the total transmission power at the relays, such that the link throughput between $S$ and $D$ is greater than or equal to a threshold $C$. Based on Eq. (4.2), the constraint that the throughput is greater than or equal to $C$ (i.e., $R(\tilde{D}) \geq C$) is

$$\frac{\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2}{1 + \sum_{i=1}^{n} x_i^2} \geq (2^4 - 1) \frac{N}{P}, \quad (4.3)$$

where $x_i = \beta_i \sqrt{\frac{P}{\sigma_i^2 P + N}}$. To simplify the notations, let $T = (2^4 - 1) \frac{N}{P}$. Then, the problem of minimizing the total transmission power at the relay nodes while
maintaining the required link throughput is given as follows:

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} P_i \\
\text{s.t.} & \quad -\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2 + T \left(1 + \sum_{i=1}^{n} x_i^2\right) \leq 0 \\
& \quad 0 \leq P_i \leq Q, \text{ for } i = 1, \ldots, n,
\end{align*}
\]

where \(x_i = \beta_i \sqrt{\frac{P_i}{\alpha_i^2 P + N}}\). Since \(P_i = \frac{x_i^2}{\beta_i^2} (\alpha_i^2 P + N)\), we simplify the above optimization problem as following:

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} \frac{x_i^2}{\beta_i^2} (\alpha_i^2 P + N) \\
\text{s.t.} & \quad -\left(\sum_{i=1}^{n} \alpha_i x_i\right)^2 + T \left(1 + \sum_{i=1}^{n} x_i^2\right) \leq 0 \\
& \quad 0 \leq x_i \leq \beta_i \sqrt{\frac{Q}{\alpha_i^2 P + N}}, \text{ for } i = 1, \ldots, n.
\end{align*}
\]

For Problem (4.5), \(P, N, Q, \alpha_i\) and \(\beta_i\) are known parameters, and \(x_i\) \((i = 1, \ldots, n)\) are unknown variables.

Although the Min-Energy problem provides a theoretical lower bound on the transmission power to maintain the link throughput, it has some strong assumptions. First, the solution involves all relays. This will generate significant synchronization overhead, since the relays have to be synchronized to have their signals arrive at the destination simultaneously. Second, each relay can transmit at any power level below the maximum power \(Q\). However, in practice, there are only limited number of power levels, and it is hard to continuously adjust the power level to all possible values.

To address these limitations, we study a more practical problem called the Min-Relay problem. In the Min-Relay problem, the goal is to minimize the number of active relays, where each active relay transmits at power \(Q\), such that the link throughput between \(S\) and \(D\) is greater than a threshold \(C\). Let \(I_i \in \{0, 1\}\) denote whether relay \(R_i\) has been chosen to forward the signal \((I_i = 1\) represents \(R_i\) is chosen, and \(I_i = 0\) represents otherwise\), then it is straightforward to formulate the Min-Relay problem as an integer programming problem. Similar to Eq. (4.3),
the constraint that the throughput is greater than or equal to $C$ is

$$\frac{(\sum_{i=1}^{n} I_i \theta_i)^2}{1 + \sum_{i=1}^{n} I_i \delta_i^2} \geq \mathcal{T},$$  \hfill (4.6)$$

where $\delta_i = \beta_i \sqrt{\frac{Q}{\alpha_i^2 P + N}}$, $\theta_i = \alpha_i \delta_i$. Since minimizing the number of active relays is equivalent to minimizing the sum of $I_i$, the Min-Relay problem can be formulated as follows:

$$\begin{align*}
\text{min} & \quad \sum_{i=1}^{n} I_i \\
\text{s.t.} & \quad L = \left(\sum_{i=1}^{n} \theta_i I_i\right)^2 - \mathcal{T} \sum_{i=1}^{n} \delta_i^2 I_i \geq \mathcal{T}, \\
& \quad I_i \in \{0, 1\}, \text{ for } i = 1, \ldots, n,
\end{align*}$$ \hfill (4.7)$$

where $\delta_i = \beta_i \sqrt{\frac{Q}{\alpha_i^2 P + N}}$, $\theta_i = \alpha_i \delta_i$, $\mathcal{T} = (2^4C - 1) \frac{N}{P}$, and $I_i \in \{0, 1\}$ ($i = 1, \ldots, n$) is the unknown variable to denote whether relay $R_i$ has been chosen to help forward the signal.

### 4.3 The Min-Energy problem

In this section, we study the Min-Energy problem, and propose an optimal solution which minimizes the transmission power of the relays while maintaining the required link throughput.

#### 4.3.1 Important Properties

Before solving the Min-Energy problem, we first discuss some of its important properties.

**Lemma 15.** When the maximum transmission power of the relay nodes $Q$ goes to infinity, a feasible solution to Problem (4.5) exists if and only if:

$$\sum_{i=1}^{n} \alpha_i^2 > \mathcal{T},$$

where $\mathcal{T} = (2^4C - 1) \frac{N}{P}$. 
Proof. According to Cauchy Schwarz inequality [57], we have
\[
\sum_{i=1}^{n} \alpha_i^2 \geq \frac{(\sum_{i=1}^{n} \alpha_i x_i)^2}{\sum_{i=1}^{n} x_i^2} > \frac{(\sum_{i=1}^{n} \alpha_i x_i)^2}{1 + \sum_{i=1}^{n} x_i^2}.
\]

To satisfy the constraint (4.3) of Problem (4.5), it is required that \(\sum_{i=1}^{n} \alpha_i^2 > T\).

Given \(T\), the above lemma indicates that even when each relay can employ infinite power to forward its received signal, to maintain a higher throughput, more relays or relays closer to the source node should be involved in the transmission.

For the more practical case of \(Q \ll \infty\), the above condition is necessary but not sufficient for the existence of a feasible solution. Then, we have the following lemma.

**Lemma 16.** For the optimization Problem (4.5), when a feasible solution exists (i.e., when \(\sum_{i=1}^{n} \alpha_i^2 > T\)), the problem is non-convex.

**Proof.** The Hessian matrix of constraint (4.3) is given as follows:
\[
H = \begin{bmatrix}
2(T - \alpha_1^2) & -2\alpha_1\alpha_2 & \cdots & -2\alpha_1\alpha_n \\
-2\alpha_1\alpha_2 & 2(T - \alpha_2^2) & \cdots & -2\alpha_2\alpha_n \\
\vdots & \vdots & \ddots & \vdots \\
-2\alpha_1\alpha_n & -2\alpha_2\alpha_n & \cdots & 2(T - \alpha_n^2)
\end{bmatrix}.
\]

Let \(H = A + B\), where \(A = 2T \ast I_n\) (\(I_n\) is the identity matrix of size \(n\)), and \(B = -2[\alpha_1, \alpha_2, \ldots, \alpha_n] \times [\alpha_1, \alpha_2, \ldots, \alpha_n]\). Matrix \(B\) is the product of two vectors, and its rank cannot exceed the rank of a vector. Then the rank of matrix \(B\) is 1, and 0 is an eigenvalue of \(B\), and the algebraic multiplicity of eigenvalue 0 is \(n - 1\). With \(H = 2T \ast I_n + B\), the eigenvectors of \(B\) are also eigenvectors of \(H\). For each of the eigenvectors, its corresponding eigenvalue in \(B\) is \(2T\) smaller than its eigenvalue in \(H\). Because 0 is an eigenvalue of \(B\) and the eigenvalues of \(B\) are \(2T\) smaller than the eigenvalues of \(H\), \(2T\) is an eigenvalue of \(H\), and the algebraic multiplicity of eigenvalue \(2T\) is \(n - 1\). Matrix \(H\) is of size \(n \times n\), and its eigenvalue \(2T\) has multiplicity of \(n - 1\), then \(H\) has only one more eigenvalue. Let \(\lambda\) denote the remaining eigenvalue, based on [57], the trace of \(H\) equals to the sum of all the eigenvalues of \(H\), then \(\lambda\) can be obtained by:
\[ \lambda = \sum_{i=1}^{n} 2(\mathcal{T} - \alpha_i^2) - 2\mathcal{T} \ast (n - 1) = 2\mathcal{T} - 2\sum_{i=1}^{n} \alpha_i^2. \]

Based on lemma 15, a feasible solution exists only when \( \sum_{i=1}^{n} \alpha_i^2 > \mathcal{T} \), thus eigenvalue \( \lambda \) must be smaller than 0. Accordingly, the Hessian matrix \( H \) of constraint (4.3) is not positive-semidefinite, and Problem (4.5) is non-convex.

Based on the above lemma, the Min-Energy problem is a non-convex problem. Compared to convex problems, non-convex problems are generally much more difficult to solve. This is because for non-convex problems, a local optimal point is not guaranteed to be the global optimal point. Consequently, simple algorithms that stop at a local optimal point can not provide optimal solution to non-convex problems. To address such difficulties, we apply semidefinite relaxation (SDR) [55] technique to Problem (4.5), and prove that our solution can minimize the power consumption of the relays.

### 4.3.2 Semidefinite Relaxation

SDR is a computationally efficient technique to provide approximated solutions for difficult optimization problems. It has been widely used to solve non-convex quadratically constrained quadratic programs. Basically, SDR first relaxes the non-convex constraints of the original problem, and transforms the original difficult non-convex problem to a simpler convex problem. By solving the simpler convex problem, an approximated solution to the original problem can be obtained.

To apply SDR to our Min-Energy problem, we present the canonical form of Problem (4.5) as following:

\[
\begin{align*}
\min & \quad x^T U x \\
\text{s.t.} & \quad x^T V x \leq -\mathcal{T}, \\
& \quad 0 \leq x \leq W,
\end{align*}
\]

where \( x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n \), \( V = -[\alpha_1, \ldots, \alpha_n]^T \times [\alpha_1, \ldots, \alpha_n] + T \times \mathcal{I}_n \), \( W = \left[ \sqrt{\frac{\beta_2^2 Q}{\alpha_1^2 P + N}}, \ldots, \sqrt{\frac{\beta_2^2 Q}{\alpha_n^2 P + N}} \right]^T \), and \( U \) is a \( n \times n \) diagonal matrix with diagonal elements \( \left[ \frac{\alpha_1^2 P + N}{\beta_1^2}, \ldots, \frac{\alpha_n^2 P + N}{\beta_n^2} \right] \). To isolate the convex constraints from the non-
convex constraints, we introduce a new variable $X = xx^T$. Since $x \in \mathbb{R}^n$, $X$ must be a symmetric positive semidefinite matrix and $\text{rank}(X) = 1$. For the objective function and the constraint in Problem (4.8), we have

$$
\begin{align*}
x^T U x &= \text{Tr}(U xx^T) = \text{Tr}(UX), \\
x^T V x &= \text{Tr}(V xx^T) = \text{Tr}(VX),
\end{align*}
$$

where $\text{Tr}(\cdot)$ is the trace of a matrix. Then, Problem (4.8) is equivalent to the following problem:

$$
\begin{align*}
\min_{X \in \mathbb{S}^n} & \quad \text{Tr}(UX) \\
\text{s.t.} & \quad \text{Tr}(VX) \leq -\mathcal{T}, \\
& \quad \text{diag}(X) \leq W \odot W, \\
& \quad X \succeq 0, \\
& \quad \text{rank}(X) = 1.
\end{align*}
$$

Here $\mathbb{S}^n$ is the set of symmetric matrices of size $n \times n$, $\text{diag}(X)$ is the vector formed by the diagonal elements of $X$, and $W \odot W$ is the Hadamard product (i.e., the pairwise product that produces a vector of the same size as the original vector), and $X \succeq 0$ represents matrix $X$ is positive semidefinite.

For the objective function and the first two constraints of Problem (4.9), they only involve linear functions, thus they are all convex. The third constraint (i.e. $X \succeq 0$) is also convex, since for any $0 < \lambda < 1$, $X_1 \succeq 0$ and $X_2 \succeq 0$, we have $\lambda X_1 + (1 - \lambda)X_2 \succeq 0$. Hence, the difficulty of solving Problem (4.9) lies in the last non-convex constraint, which requires the rank of matrix $X$ equals to 1. Based on SDR, we remove this non-convex constraint and have the following relaxed convex problem:

$$
\begin{align*}
\min_{X \in \mathbb{S}^n} & \quad \text{Tr}(UX) \\
\text{s.t.} & \quad \text{Tr}(VX) \leq -\mathcal{T}, \\
& \quad \text{diag}(X) \leq W \odot W, \\
& \quad X \succeq 0.
\end{align*}
$$

The above convex problem can be easily solved. In the next subsection, we convert a globally optimal solution $\hat{X}^*$ to Problem (4.10) into a feasible solution $X^*$ to Problem (4.8). We then prove that the feasible solution $X^*$ is in fact an
optimal solution to Problem (4.8).

### 4.3.3 Optimal Solution

Let \( n \times n \) symmetric matrix \( \hat{X}^* \) be an optimal solution to Problem (4.10), where the element at the \( i \)-th row and \( j \)-th column of \( \hat{X}^* \) is denoted by \( \hat{x}_{ij}^* \). As it is required that \( \hat{X}^* \succeq 0 \), we have \( \hat{x}_{ii}^* \geq 0 \) for \( 1 \leq i \leq n \). Based on \( \hat{X}^* \), we construct \( X^* = \left[ \sqrt{\hat{x}_{11}^*}, \sqrt{\hat{x}_{22}^*}, \ldots, \sqrt{\hat{x}_{nn}^*} \right]^T \times \left[ \sqrt{\hat{x}_{11}^*}, \sqrt{\hat{x}_{22}^*}, \ldots, \sqrt{\hat{x}_{nn}^*} \right] \). Let \( x_{ij}^* \) denote the element in the \( i \)-th row and \( j \)-th column of \( X^* \), then we have \( x_{ij}^* = \sqrt{\hat{x}_{ii}^*} \sqrt{\hat{x}_{jj}^*} \).

The following lemma compares the value of \( \hat{x}_{ij}^* \) and \( x_{ij}^* \).

**Lemma 17.** \( x_{ij}^* \geq \hat{x}_{ij}^* \), \( \forall 1 \leq i \leq n \) and \( 1 \leq j \leq n \).

**Proof.** When \( i = j \), based on the definition of \( X^* \), \( x_{ij}^* = \hat{x}_{ij}^* \).

When \( i \neq j \), we prove the lemma by contradiction. Assume \( \hat{x}_{ij}^* > x_{ij}^* \), consider a \( n \times 1 \) vector \( z \) with all elements equal to 0, except the \( i \)-th element \( z_i = \sqrt{\hat{x}_{ii}^*} \) and the \( j \)-th element \( z_j = -\sqrt{\hat{x}_{jj}^*} \). Then we have

\[
z^T \hat{X}^* z = 2 \sqrt{\hat{x}_{ii}^*} \sqrt{\hat{x}_{jj}^*} \left( \sqrt{\hat{x}_{ii}^*} \sqrt{\hat{x}_{jj}^*} - \hat{x}_{ij}^* \right).
\]

Recall that \( x_{ij}^* = \sqrt{\hat{x}_{ii}^*} \sqrt{\hat{x}_{jj}^*} \) and it is assumed that \( \hat{x}_{ij}^* > x_{ij}^* \). Then, the right-hand side of above equation is less than 0. For this specific \( z \), we have \( z^T X^* z < 0 \), which contradicts the condition that \( \hat{X}^* \) is positive semidefinite. \( \square \)

Based on the above lemma, we obtain an interesting property of the constructed matrix \( X^* \), which is stated as follows.

**Lemma 18.** \( X^* \) is a feasible, in fact optimal, solution to Problem (4.10).

**Proof.** First, we show that \( X^* \) satisfies the first constraint of Problem (4.10). Recall that \( V = -[\alpha_1, \ldots, \alpha_n]^T \times [\alpha_1, \ldots, \alpha_n] + T \times I_n \), and let \( Y = -[\alpha_1, \ldots, \alpha_n]^T \times [\alpha_1, \ldots, \alpha_n] \) and let \( Z = T \times I_n \). Note that the diagonal elements of \( \hat{X}^* \) and \( X^* \) are the same, then we have \( \text{Tr}(Z X^*) = \text{Tr}(Z \hat{X}^*) \). Because \( \alpha_i \geq 0 \) for all \( i \), all the elements of \( Y \) are less than or equal to 0. Combining this property of \( Y \) with the results of lemma 17 (i.e., \( x_{ij}^* \geq \hat{x}_{ij}^* \)), we have \( \text{Tr}(Y X^*) \leq \text{Tr}(Y \hat{X}^*) \). Consequently, \( \text{Tr}(V X^*) = \text{Tr}(Z X^*) + \text{Tr}(V X^*) \leq \text{Tr}(V \hat{X}^*) = \text{Tr}(Z \hat{X}^*) + \text{Tr}(V \hat{X}^*) \). Since \( \hat{X}^* \) is a feasible solution to Problem (4.10), \( \text{Tr}(V X^*) \leq \text{Tr}(V \hat{X}^*) \leq -T \).
Second, we prove that $X^*$ also satisfies the second and third constraints of Problem (4.10). Since the diagonal elements of $X^*$ and $\hat{X}^*$ are the same, $X^*$ must satisfy the second constraint. For the third constraint, based on the definition of $X^*$, for any $n \times 1$ vector $z$, we have

$$
z^T X^* z = \left( \left[ \sqrt{\hat{x}_{11}^*}, \sqrt{\hat{x}_{22}^*}, \ldots, \sqrt{\hat{x}_{nn}^*} \right] z \right)^T \times \left( \left[ \sqrt{\hat{x}_{11}^*}, \sqrt{\hat{x}_{22}^*}, \ldots, \sqrt{\hat{x}_{nn}^*} \right] z \right) \geq 0.
$$

Thus, $X^*$ is positive semidefinite, and $X^*$ satisfies the third constraint.

Lastly, we show that by replacing $\hat{X}^*$ with $X^*$, the value of the objective function is unchanged. Based on problem formulation, $U$ is a diagonal matrix, the condition that the diagonal elements of $X^*$ and $\hat{X}^*$ are the same is sufficient to guarantee that $\text{Tr}(UX^*) = \text{Tr}(U\hat{X}^*)$.

In conclusion, $X^*$ is a feasible solution to Problem (4.10), and $\text{Tr}(UX^*)$ is equal to the optimal value of the problem. Thus, $X^*$ is an optimal solution to Problem (4.10).

Based on the above lemma, we now show that the constructed matrix $X^*$ is actually optimal to Problem (4.9).

**Theorem 7.** $X^*$ is a feasible, in fact optimal, solution to Problem (4.9).

**Proof.** Since $X^* = \left[ \sqrt{\hat{x}_{11}^*}, \sqrt{\hat{x}_{22}^*}, \ldots, \sqrt{\hat{x}_{nn}^*} \right]^T \times \left[ \sqrt{\hat{x}_{11}^*}, \sqrt{\hat{x}_{22}^*}, \ldots, \sqrt{\hat{x}_{nn}^*} \right]$, $\text{Rank}(X^*) = 1$. Combining this result with lemma 18, $X^*$ is a feasible solution to Problem (4.9).

To show $X^*$ is optimal to Problem (4.9), we assume there exists another feasible solution $\tilde{X}^* \neq X^*$ to Problem (4.9), such that $\text{Tr}(U\tilde{X}^*) < \text{Tr}(UX^*)$. Because Problem (4.10) is a relaxed problem of Problem (4.8), $\tilde{X}^*$ must also be feasible to Problem (4.10). Then, the optimal value of Problem (4.10), which is $\text{Tr}(UX^*)$, can not exceed $\text{Tr}(U\tilde{X}^*)$. This contradicts the condition that $\text{Tr}(U\tilde{X}^*) < \text{Tr}(UX^*)$, thus $X^*$ must be optimal to Problem (4.9).

Problem (4.9) and the original Problem (4.8) are equivalent, where the only difference is that the unknown variable is $x$ in Problem (4.8) and the unknown variable is $X = xx^T$ in Problem (4.9). In theorem 7, we have shown that $X^*$ is optimal to
and it is known that $X^* = [\sqrt{x_{11}^*}, \sqrt{x_{22}^*}, \ldots, \sqrt{x_{nn}^*}]^T \times [\sqrt{x_{11}}, \sqrt{x_{22}}, \ldots, \sqrt{x_{nn}}]$, and then $[\sqrt{x_{11}^*}, \sqrt{x_{22}^*}, \ldots, \sqrt{x_{nn}^*}]^T$ must be optimal to the original problem (4.8).

**Theorem 8.** Given a solution accuracy $\epsilon > 0$, the Min-Energy problem can be solved within $O(n^{4.5} \log(1/\epsilon))$ time, where $n$ is the number of relays in the network.

**Proof.** Based on [55, 58], the time complexity of solving Problem (4.10) is $O(n^{4.5} \log(1/\epsilon))$. Obtaining the solution to Problem (4.8) based on the solution to Problem (4.10) takes $O(n)$ time. Therefore, the Min-Energy problem can be solved within $O(n^{4.5} \log(1/\epsilon))$ time.

### 4.4 The Min-Relay Problem

In this section, we study the Min-Relay problem, where the goal is to minimize the number of active relays while maintaining the social link.

#### 4.4.1 Relay Selection

The Min-Relay problem (4.7) is an integer programming problem which chooses the minimum number of relays, until the constraint $L \geq T$ is satisfied. Let each relay $R_i$ be represented by the parameter $(\theta_i, \delta_i)$ as defined in Problem (4.7). Then the $n$ relays in the Min-Relay problem can be viewed as $n$ pairs $(\theta_1, \delta_1), \ldots, (\theta_n, \delta_n)$. The problem of choosing the minimum number of relays becomes equivalent to choosing the minimum number of pairs, such that the constraint $L \geq T$ is satisfied.

The basic idea of our solution is based on a related problem. Consider $n$ packets each with weight $W_i$, the problem of finding the minimum number of packets, such that their total weight is greater than $M$ can be formulated as follows:

$$\min \sum_{i=1}^{n} I_i \quad \text{s.t.} \quad \sum_{i=1}^{n} W_i I_i \geq M,$$

(4.11)

For the above problem, the solution is to choose packets in the decreasing order of their weights, until their total weight is larger than $M$. To have a larger total
weight, heavier packets are chosen before lighter packets. If the packets are ranked in the descending order of their weights, packets can be chosen one by one until the total weight grows larger than $M$. For Problem (4.7), if a similar ordering of the relays can be obtained, relays can be chosen one by one until constraint $L \geq T$ is satisfied.

For the integer programming problem (4.7), it may have multiple optimal solutions; i.e., there may exist more than one $I = \{I_1, I_2, \ldots, I_n\}$ that are optimal to Problem (4.7). We focus on one specific optimal solution $I^*$, which is defined as follows:

**Definition 5.** Let set $S$ include all optimal solutions of Problem (4.7). $I^* = \{I_1^*, I_2^*, \ldots, I_n^*\}$ is the optimal solution that maximizes $L$, that is

$$I^* = \underset{I \in S}{\text{argmax}}(L) = \underset{I \in S}{\text{argmax}} \left( \sum_{i=1}^{n} \theta_i I_i \right)^2 - T \sum_{i=1}^{n} \delta_i^2 I_i \right).$$ (4.12)

If any $I_i^*$ in $I^*$ is changed, it leads to either a sub-optimal solution or a decrease of $L$. Based on $I^*$, we formally define the selection order of the relays.

**Definition 6.** For any two relays $R_i$ and $R_j$, $R_i$ should be selected before $R_j$, denoted by $R_i \geq R_j$, if $I_i^* = 0$ and $I_j^* = 1$ cannot both hold.

In this definition, since $I_i^* = 0$ and $I_j^* = 1$ cannot both hold, it is impossible for $R_j$ to be selected while $R_i$ is not selected. Hence, $R_i$ should always be selected before $R_j$. However, determining the selection order of the relays is not easy. When a relay $R_i$ is added, the improvement of the link throughput not only depends on the parameter of $R_i$, but also depends on the relays that have already been selected. Hence, we can not obtain a simple selection order as in Problem (4.11), where the order only depends on the parameter of $R_i$. For the selection order of the relays, the order might be different when different relays have already been selected. Let $\mathcal{K}^* = \sum_{i=1}^{n} \theta_i I_i^*$, based on Definition 6, the following lemma states how to order two arbitrary relays.

**Lemma 19.** Consider any two relays $R_i$ and $R_j$ with parameters $(\theta_i, \delta_i)$ and $(\theta_j, \delta_j)$
(assume $\theta_j > \theta_i$), and let $\mathcal{K}_{ij} = \frac{\tau(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}$. If $\mathcal{K}^* > \mathcal{K}_{ij}$, then relay $R_j$ is selected before $R_i$; otherwise, relay $R_i$ is selected before $R_j$.

**Proof.** We prove the lemma by contradiction. Suppose $I^*_i = 1$ and $I^*_j = 0$, and we replace $R_i$ with $R_j$ to transmit the data (i.e., let $I^*_i = 0$ and $I^*_j = 1$). Then, the value of the objective function remains unchanged, while $L$ differs by:

$$\Delta L = (K^* + \theta_j - \theta_i)^2 - K^*^2 - \tau (\delta_j^2 - \delta_i^2)$$

$$= 2K^* (\theta_j - \theta_i) + (\theta_j - \theta_i)^2 - \tau (\delta_j^2 - \delta_i^2).$$

If $\Delta L > 0$, the new solution has a larger $L$ than $I^*$, which contradicts the definition of $I^*$. Thus, when $\Delta L > 0$, $I^*_i = 1$ and $I^*_j = 0$ never hold, and consequently $R_j$ is selected before $R_i$. Given $\theta_j > \theta_i$, the condition $\Delta L > 0$ is equivalent to

$$
\mathcal{K}^* > \frac{\tau(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)} - \frac{1}{2} (\theta_j - \theta_i).
$$

Then, when $\mathcal{K}^* > \mathcal{K}_{ij} = \frac{\tau(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}$ (note that $\theta_j > \theta_i > 0$), $R_j \geq R_i$ ($R_j$ is selected before $R_i$).

Similarly, suppose $I^*_i = 0$ and $I^*_j = 1$, replacing $R_j$ with $R_i$ will increase $L$ when

$$\mathcal{K}^* < \frac{\tau(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)} - \frac{1}{2} (\theta_i - \theta_j).$$

That is, when $\mathcal{K}^* < \mathcal{K}_{ij}$, we have $R_i \geq R_j$. \hfill \Box

The above lemma states that for any two relays $R_i$ and $R_j$, based on $\mathcal{K}_{ij} = \frac{\tau(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}$, their selection order can be determined as follows: when $\mathcal{K}^* > \mathcal{K}_{ij}$, the relay with larger $\theta$ value should be chosen first; otherwise, the relay with smaller $\theta$
value should be chosen first. If we calculate $K_{ij}$ for any two of the $n$ relays, there will be $\frac{n(n-1)}{2}$ different $K_{ij}$. Those different $K_{ij}$ will divide the feasible region of $K^*$ into $\frac{n(n-1)}{2} + 1$ small regions, where in each region we know the selection order between any two relays.

Fig. 4.1 gives an example for the case of three relays $R_1$, $R_2$ and $R_3$ ($\theta_1 < \theta_2 < \theta_3$). When $K^*$ resides in the leftmost region, $K^* < K_{12}$ leads to $R_1 \geq R_2$, $K^* < K_{13}$ leads to $R_1 \geq R_3$ and $K^* < K_{23}$ leads to $R_2 \geq R_3$. The selection order of the relays in other three regions are also shown in the figure, which is obtained according to the value of $K^*$, $K_{12}$, $K_{13}$ and $K_{23}$.

However, given the selection order between any two relays, a complete ordering of all relays may not exist. For example, if the ordering is like $R_1 \geq R_2$, $R_2 \geq R_3$, and $R_3 \geq R_1$, then $R_1 \geq R_2$ and $R_2 \geq R_3$ lead to $R_1 \geq R_3$ which contradicts $R_3 \geq R_1$. To show that a complete ordering of the relays always exists, we prove the ordering given in lemma 19 is transitive.

**Lemma 20.** For a given $K^*$, $R_i \geq R_j$ and $R_j \geq R_k$ will lead to $R_i \geq R_k$.

**Proof.** Based on the value of $\theta_i$, $\theta_j$ and $\theta_k$, there are 6 different cases. Consider the first case of $\theta_i > \theta_j > \theta_k$. $R_i \geq R_j$ and $\theta_i > \theta_j$ lead to $K^* \geq K_{ij}$, which is

$$K^* \geq \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)}.$$  \hspace{1cm} (4.13)

Similarly, with $R_j \geq R_k$ and $\theta_j > \theta_k$, we have

$$K^* \geq \frac{\mathcal{T}(\delta_j^2 - \delta_k^2)}{2(\theta_j - \theta_k)}.$$  \hspace{1cm} (4.14)

For $K_{ik}$, we have

$$K_{ik} = \frac{\mathcal{T}(\delta_i^2 - \delta_k^2)}{2(\theta_i - \theta_k)} = \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)}{2(\theta_i - \theta_j)} + \frac{\mathcal{T}(\delta_j^2 - \delta_k^2)}{2(\theta_j - \theta_k)}$$

$$= \frac{\mathcal{T}(\delta_i^2 - \delta_j^2)(\theta_i - \theta_j)}{2(\theta_i - \theta_j)(\theta_i - \theta_k)} + \frac{\mathcal{T}(\delta_j^2 - \delta_k^2)(\theta_j - \theta_k)}{2(\theta_j - \theta_k)(\theta_i - \theta_k)}.$$
Combining the above equation with Eq. (4.13) Eq. (4.14), we can get

\[ K_{ik} \leq K^* \frac{(\theta_i - \theta_j)}{\theta_i - \theta_k} + K^* \frac{(\theta_j - \theta_k)}{\theta_i - \theta_k} = K^*. \]

Thus \( R_i \geq R_k \). The other five cases can be similarly proved by comparing the value of \( K_{ik} \) and \( K^* \) based on \( K_{ij} \) and \( K_{jk} \).

For each of the small region bounded by two consecutive \( K_{ij} \) or infinity, we can get a complete ordering of the relays. For example, in the leftmost region of Fig. 4.1, the ordering will be \( R_1 \geq R_2 \geq R_3 \). That is, \( R_1 \) should be chosen first, and \( R_2 \) should be chosen second, and \( R_3 \) should be chosen third.

To sum up, we can find the selection order of the relays based on the value of \( K^* \). If the value of \( K^* \) is known, relays can be selected one by one according to the selection order, until the constraint is met. Unfortunately, the value of \( K^* \) is not known a priori, and we have to try each of the \( \frac{n(n-1)}{2} + 1 \) selection order. By comparing all the different solutions, we can obtain an optimal solution to the Min-Relay problem. Next, we present the details of this solution, which is referred to as the Min-Relay algorithm.

### 4.4.2 The Min-Relay Algorithm

Algorithm 1 shows the formal description of the algorithm. Recall that \( \delta_i \) and \( \theta_i \) are calculated based on the signal scaling factor \( \alpha_i \) and \( \beta_i \), where \( \delta_i = \beta_i \sqrt{\frac{Q}{\alpha_i^2 + \gamma}} \), \( \theta_i = \alpha_i \delta_i \), and we have \( \delta_i > 0 \) and \( \theta_i > 0 \). Parameter \( T \) is related to the throughput between the source and destination, which is given by \( T = \frac{2^{4C} - 1}{N} \).

In the algorithm, \( S_{opt} \) records the relays chosen in the optimal solution, and \( Q_K \) stores all the small regions divided by \( K_{ij} \), and \( Q_{Relay} \) stores the current ordering of relays. The algorithm first calculates \( K_{ij} \) for each pair of relays, and push the tuple \((i, j, K_{ij})\) into the priority queue \( Q_K \), where the tuple with smaller \( K_{ij} \) has higher priority. After that, one more tuple is pushed into the queue (in Line 7). It corresponds to the rightmost region. Line 8-10 gives the ordering of the relays for the first region (i.e., the leftmost region), where simply relays with lower \( \theta_i \) are first chosen. The while loop between Line 11-22 finds the solution when \( K^* \) resides in each of the small region. Basically, in each small region, the algorithm chooses
Algorithm 1 Min Relay Algorithm

1: \( S_{\text{opt}} \leftarrow \emptyset \), \( Q_{\text{Relay}} \leftarrow \text{initQueue()} \)
2: \( Q_K \leftarrow \text{initPriorityQueue()} \)
3: for any \( R_i \) and \( R_j \) with \( i \leq j \) do
4: \( K_{ij} \leftarrow T(\delta_i^2 - \delta_j^2)/2(\theta_i - \theta_j) \)
5: \( Q_K.\text{push}({R_i, R_j, K_{ij}}) \quad \triangleright \text{tuple with smaller } K_{ij} \text{ has higher priority} 
6: \text{end for} 
7: Q_K.\text{push}({\text{NULL, NULL, inf}}) 
8: for each \( R_i \), in the increasing order of \( \theta_i \) do 
9: \( Q_{\text{Relay}}.\text{push}({R_i, \theta_i, \delta_i}) \)
10: \text{end for} 
11: while \( Q_K.\text{notEmpty()} \) do 
12: \( \{R_i, R_j, K_{\text{upper}}\} \leftarrow Q_K.\text{poll()} \), \( Q_{\text{ranking}} \leftarrow Q_{\text{Relay}} \)
13: \( K \leftarrow 0, B \leftarrow 0, S_{\text{temp}} \leftarrow \emptyset \)
14: while \( Q_{\text{ranking}}.\text{notEmpty()} \) and \( K \leq K_{\text{upper}} \) and \( K^2 - TB < T \) do 
15: \( \{R_k, \theta_k, \delta_k\} \leftarrow Q_{\text{ranking}}.\text{poll()} \)
16: \( S_{\text{temp}} \leftarrow S_{\text{temp}} \cup \{R_k\}, K \leftarrow K + \theta_k, B \leftarrow B + \delta_k^2 \)
17: \text{end while} 
18: if \( K^2 - TB \geq T \) and \((|S_{\text{temp}}| < |S_{\text{opt}}| \text{ or } S_{\text{opt}} = \emptyset) \) then 
19: \( S_{\text{opt}} \leftarrow S_{\text{temp}} \)
20: \text{end if} 
21: Call update-ordering \((Q_{\text{relay}}, R_i, R_j)\) 
22: \text{end while} 

Procedure 1 Update-Ordering \((Q_{\text{relay}}, R_i, R_j)\)

1: \( Q_{\text{temp}} \leftarrow \text{initQueue()} \)
2: while \( Q_{\text{relay}}.\text{notEmpty()} \) do 
3: \( \{R_k, \theta_k, \delta_k\} \leftarrow Q_{\text{relay}}.\text{poll()} \)
4: if \( R_k = R_i \) then 
5: \( Q_{\text{temp}}.\text{push}({R_j, \theta_j, \delta_j}) \)
6: \text{else if } R_k = R_j \text{ then} 
7: \( Q_{\text{temp}}.\text{push}({R_i, \theta_i, \delta_i}) \)
8: \text{else} 
9: \( Q_{\text{temp}}.\text{push}({R_k, \theta_k, \delta_k}) \)
10: \text{end if} 
11: \text{end while} 
12: \( Q_{\text{relay}} \leftarrow Q_{\text{temp}} \)
relays one by one until either the constraint \( L \geq T \) is met, or for the relays already chosen, \( K \) grows out of the bounds of the region (i.e., \( K = \sum_{i=1}^{n} \theta_i I_i > K_{upper} \)). If a solution is obtained, in Line 18-20, the algorithm compares the solution with the current best solution, and updates the current best solution if needed (\(| \cdot | \) in Line 18 is the cardinality of a set). In Line 21, the algorithm calls the Update-Ordering Procedure to update the ordering of the relays. In this way, as the while loop between Line 11 and 22 starts over (i.e., compute the solution for the next region), \( Q_{relay} \) will store the correct ordering for the next region. The details of the update-ordering procedure are given in Procedure 1.

Note that in Procedure 1, what we have achieved is simply exchange \( R_i \) and \( R_j \) in the ordering. Because in the next region, as \( K^* \) crosses \( K_{ij} \), only the ordering between \( R_i \) and \( R_j \) is reversed, and the ordering of all others are kept unchanged. For instance, in Fig. 4.1, when comparing the leftmost region and the second left region, the only difference is the ordering of \( R_1 \) and \( R_2 \) is reversed. Hence, procedure 1 gives the correct ordering for the next region.

**Theorem 9.** The time complexity of the Min Relay Algorithm is \( O(n^3) \), where \( n \) is the number of relays in the network.

**Proof.** For the while loop between Line 11 and Line 22, it repeats at most \( O(n^2) \) times, since there are in total \( O(n^2) \) small regions. The Update Ordering Procedure takes \( O(n) \) time, and the while loop between Line 14 and Line 17 repeats at most \( O(n) \) times. Hence, it takes \( O(n^3) \) to run the algorithm. \( \square \)

### 4.5 Performance Evaluations

In this section, we evaluate the performance of the proposed Min-Energy solution and the Min-Relay algorithm in terms of power consumption and the number of active relays.

#### 4.5.1 Simulation setup

Our evaluation is based on the UCSD trace [59], which is collected at a campus scale with 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
trace records the contacts between 275 users every 20 seconds for a period of 77 days. Each user can detect multiple APs, and the user location is inferred based on the location of the detected APs and the AP signal strength [60].

Because the social relationship between the users is not reported in the trace, to identify the important social links, we construct a weighted directed social graph upon the nodes in the network similar to [61]. Let \( f \) denote the total contact frequency of the whole trace, \( f_i \) denote the total contact frequency of node \( i \), and \( f_{ij} \) denote the contact frequency between node \( i \) and \( j \). Since users connected by stronger ties are more likely to contact frequently with each other, and users with more social ties are more likely to meet with others [62], the social graph is built as follows. First, we generate the node degrees following the power-law distribution, where the power-law coefficient is set to 1.6. Second, we assign the node degrees to the nodes in the network. For the largest node degree, it is assigned to node \( i \) with a probability of \( \frac{f_i}{f} \). Then, this process is repeated for the remaining degrees and nodes. Third, for the social ties of each node, we generate weight for each of the social tie, where the weight is uniformly distributed in \([0,1]\). Finally, we connect the social ties of each node to other nodes. Specifically, for the strongest tie of node \( i \), it is connected to another node in a way that node \( j \)'s probability to be connected is \( \frac{f_{ij}}{f_i} \), and this is repeated for the remaining social ties and the nodes have not been connected to \( i \).

In the simulation, node locations are generated based on a snapshot of the UCSD trace; i.e., we use the location of \( n \) nodes in that trace at a specific time to generate the node location. The important social link corresponds to link with the strongest social tie in the network. In the simulations, we focus on cases where multi-hop transmission can not be maintained due to network partition, and amplify-and-forward is used to help transmit the data. The link source is the node in the routing path that can not maintain the required link throughput between itself and its next hop node (called link destination), even with the highest transmission power. To simplify the notations, we use S-D to represent the distance between the link source and the link destination.

In the simulation, the link source uses power \( P = 2W \) to transmit the data to all relay nodes. For Min-Energy, each relay can employ any power level below the maximum power \( Q = 2W \) for data transmission. For Min-Relay, all relays use
4.5.2 Power Consumption

Fig. 4.2 evaluates how various parameters such as the link throughput, the number of relays, and the distance between the source and destination affect the power consumption of Min-Energy. Fig. 4.2(a) shows the effects of link throughput $C$ on the total power consumption. The noise at each node follows a normal distribution $\sim \mathcal{N}(0, N)$, where $N = 10^{-10} W$.

For power $Q = 2W$ for data transmissions. The noise at each node follows a normal distribution $\sim \mathcal{N}(0, N)$, where $N = 10^{-10} W$. 

Figure 4.2: The power consumption of Min-Energy

Figure 4.3: The power consumption of Min-Relay
and the number of relays \( n \). As expected, a higher link throughput requires higher transmission power at the relays. We can also see that the power consumption decreases when the number of relays increases. This is because with more relays, the SNR at the destination can be improved, and consequently the relays can use lower power to achieve the required link throughput.

Fig. 4.2(b) shows the power consumption as a function of \( P \) and the distance between the source and destination. As shown in the figure, the transmission power at the relays increases when the distance between the source and destination increases. This is because the signal strength at the destination drops quickly with increasing distance. To maintain the link throughput, the relays have to employ higher power for data transmission.

We can also see that the power consumption decreases as \( P \) grows. This is because increasing \( P \) increases the signal strength at the relays, and consequently improves the SNR at the destination. When \( P \) is large, further increasing \( P \) hardly affects the transmission power. This is because for large \( P \), the noise at the relay is negligible compared to \( P \), and then the SNR at the destination is only determined by the relay transmission power and the white noise at the destination.

Fig. 4.3 illustrates how the power consumption of Min-Relay is affected by the parameters including the link throughput, the relay transmission power, and the distance between the source and destination. Fig. 4.3(a) shows how the transmission power at the relays varies with link throughput, given different relay transmission power \( Q \). As can be seen, when \( Q \) increases, the total transmission power increases and the number of active relays decreases. For example, when \( C = 1.5 \) and \( Q = 2W \), the total transmission power is 28W, and there are 28/2 = 14 active relays; when \( C = 1.5 \) and \( Q = 8W \), the total transmission power is 56W, and 7 relays are involved in the transmission. The number of active relays decreases with increasing \( Q \), because with higher \( Q \), each node can provide a stronger signal strength at the destination, and consequently less number of relays are required to maintain the link.

Fig. 4.3(b) shows the power consumption as a function of \( P \), when the distance between the source and destination varies. As can be seen, increasing \( P \) reduces the transmission power at the relays, and the number of active relays. When the distance between the source and destination increases, the transmission power
increases and more relays are involved.

### 4.5.3 Comparing Min-Energy and Min-Relay

Fig. 4.4 illustrates what relays will be selected when the link throughput $C$ increases from 0.46 to 0.68 in Min-Energy and Min-Relay. In the figure, the red circle near $(0, 0)$ denotes the source, and the blue cross near $(400, 400)$ denotes the destination. There are 15 relays in the network (i.e., $n = 15$), which are denoted by the solid circles. The color of the solid circle represents the transmission power of the relay, where dark color means high power and light color means low power.
For Min-Relay, as shown in Fig. 4.4(a), when $C = 0.46$, only two relays are active, each with a transmission power of 2W. In Fig. 4.4(b), $C$ increases to 0.68. As a result, one more relay is selected to support higher link throughput. For Min-Energy, as shown in Fig. 4.4(c), when $C = 0.46$, all relays are involved, although most of them only use a relatively low power (about 0.25W) for transmission. When $C$ increases to 0.68 (see Fig. 4.4(d)), relays employ higher power for transmission, e.g., most of them transmit with a power of 0.5W. From the figure, we can easily see that the number of active relays in Min-Relay is much less than that in Min-Energy.

In Fig. 4.5(a), we compare the power consumption of Min-Relay and Min-Energy as the link throughput $C$ increases, given different distance between the source and destination. The results of Min-Energy are shown by the dashed lines, while the results of Min-Relay are shown by the solid lines. As can be seen, the power consumption of Min-Energy and Min-Relay is comparable. Min-Energy always performs better than Min-Relay, because it has less constraints and has the minimum power consumption for maintaining the required link throughput. We can also see that the power consumption of Min-Relay converges to Min-Energy as $C$ and S-D both increase (e.g., the teal and blue line at $C = 1.25$). This is because when $C$ and S-D are large, all relays are required to transmit at their highest power to maintain the required link throughput.

Fig. 4.5(b) compares the number of active relays in Min-Energy and Min-Relay.
The results for Min-Relay are shown by the dashed lines, and the results for Min-Energy are shown by the solid lines. For Min-Energy, all 30 relays are involved in the transmission, and the three lines overlap. For Min-Relay, when the throughput increases or the distance between the source and destination increases, more relays are involved. In general, Min-Relay has much less number of active relays.

### 4.6 Related Work

In the past few years, researchers have proposed various routing and data forwarding algorithms for mobile ad hoc networks based on social knowledge. For example, in [4], the authors have exploited the community and centrality social metrics, and proposed a social-based forwarding algorithm to improve the performance of data forwarding. In [2], the authors have identified the existence of transient connected components in mobile social networks, and proposed a data forwarding strategy exploiting the user contacts in transient connected components. However, these existing works focus on improving the performance of data forwarding for all nodes in the network, and none of them considers maintaining important social links through amplify-and-forward.

Amplify-and-forward is a cooperative transmission strategy which has been used to transmit data to faraway nodes. Compared to the other well-known cooperative transmission approach such as decode-and-forward [63, 64] where data is decoded at the relays, amplify-and-forward requires less processing and it saves energy at the relays since the relays simply amplifies and transmits their received signal to the destination without decoding. Moreover, as shown in [65], amplify-and-forward has lower bit error rate since it does not decode the data at the relays.

The power allocation problem in amplify-and-forward has been studied by many researchers recently. Given the total transmission power at the relays, Zhao et al. derived the transmission power for each relay, such that the link throughput between the source and destination is maximized [54]. In [51], Ding et al. proposed a distributed power allocation strategy which determines the transmission power of each relay, to maximize the data throughput between the source and destination. These existing work [50, 51, 53, 54] focuses on maximizing the link throughput, where relays transmit data at their maximum power. Different from them, in our
Min-Energy problem, our goal is to minimize the power consumption of the relays while maintaining the required link throughput.

In these aforementioned solutions, all relays are involved. Jing et al. [66] studied the problem of selecting a group of relays to help transmit the data to the destination, such that the SNR at the destination is maximized. Different from them, in our Min-Relay problem, our goal is to minimize the number of active relays, while maintaining the required link throughput. Also, [66] only gives sub-optimal solutions, while we propose a polynomial-time algorithm that can efficiently select the minimum number of relays while maintaining the required link throughput.

4.7 Conclusion

In this chapter, we applied the amplify-and-forward strategy to maintain social links in wireless networks. To save energy at the relays, we formulated and studied the Min-Energy problem. We formulated it as a non-convex quadratic programming problem by exploiting the rate-distortion theory, and solved it based on an approximation technique. We also proved that our solution can minimize the power consumption of the relays. The Min-Energy problem involves all relays, which could generate significant synchronization overhead. To minimize the synchronization overhead, we also studied the Min-Relay problem which aims to minimize the number of active relays while maintaining the social link. We formulated the Min-Relay problem as an integer programming problem, and proposed a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. Evaluation results showed that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.
5.1 Introduction

In mobile ad hoc networks, due to node mobility and transmission interference, it is hard to maintain end-to-end connections. To provide more reliable data forwarding, various routing techniques have been proposed [48, 4, 2, 5]. However, most of these existing works focus on improving the data forwarding performance for all nodes in the network. They ignore the fact that the communication between some node pairs might be more important than others, and it is critical to maintain the social links between these important social pairs. For instance, in a battlefield, for a platoon of soldiers consisting of several squads, the commander of the platoon should maintain good connections with the squad leaders, but not necessarily with all other soldiers. As another example, during disaster recovery, it is critical to maintain the social links between the command center and the rescue teams; while it is secondary to have the command center connect to each individual in the area.

In the aforementioned scenarios, in case of resource shortage, the communication resources should be devoted to maintain the most important social links, instead of maintaining the communication of other less critical links. Even when the communication resources are devoted to the important social links, it is still possible that these social links are unable to achieve the desired quality of ser-
vice (QoS). For example, consider maintaining the social link between two distant nodes in a mobile ad hoc network. Since the data has to be forwarded via multiple unstable wireless links (i.e., multihop transmission), the probability that the data is successfully delivered is quite low. Although multipath routing or even flooding can be used to improve the data forwarding performance, where data is transmitted along several designated redundant paths towards the destination, each path still experiences a high failure probability. Moreover, such redundant transmission may further degrade the communication of other social pairs, and thus it is likely that the overall data delivery ratio cannot meet the desired QoS requirement.

To address this problem, in addition to exploiting the existing communication resources, we proactively place some reliable communication links (e.g., satellite links) to the given network, to improve the data forwarding performance. More specifically, each reliable link enables the direct communication between two selected nodes in the network (even if they are far away from each other), and has a failure probability close to 0. These links are referred to as shortcut edges in the rest of the chapter. Furthermore, besides the pairs of nodes directly connected by the shortcut edges, other node pairs may also benefit from these shortcut edges if transmitting data via the shortcut edges improves the data forwarding performance.

Existing works on direct link placement are limited to improving the connectivity or reducing the distance of all node pairs [67, 68, 69, 70], and thus cannot be directly applied to our problem of maintaining the connection between important social pairs (which are a subset of all node pairs in the network). Such social link maintenance problem is also investigated in [71]. However, their work is limited to maintaining one social link through physical layer techniques such as amplify-and-forward, while we focus on simultaneously maintaining multiple social links (referred to as the MSL problem), through direct placement of shortcut edges. Since shortcut edges (e.g., satellite links) are valuable and expensive resources, we constrain ourselves to the case where the number of shortcut edges is limited. Thus, the real challenge of the problem lies in the effective placement of the shortcut edges, such that each shortcut edge can benefit more social links.

To reveal the difficulty of our problem, we first consider a special case of our MSL problem, where all important social pairs share a common node (referred
to as MSL-CN problem). By proving that the MSL-CN problem is NP-hard, we are able to prove that the general MSL problem is also NP-hard. Then, based on the Sandwich Approximation strategy [72], we propose an efficient approximation algorithm for the MSL problem. Moreover, we prove that the performance of the proposed algorithm can be theoretically bounded, and present various experimental results to verify its effectiveness.

The rest of the chapter is organized as follows. Section 5.2 presents the model and the problem formulation. We discuss the MSL-CN problem in Section 5.3 and focus on the MSL problem in Section 5.4. The performance evaluation results are presented in Section 5.5. Section 5.6 reviews related work and Section 5.7 concludes the chapter.

5.2 Problem Formulation

5.2.1 Network Model

We consider a wireless network with $n$ nodes, where $m$ important social links should be maintained ($m \leq \binom{n}{2}$). The network is modeled as an undirected graph $G = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ is the vertex set, and edge $e_{i,j} \in E$ represents an edge between $v_i$ and $v_j$.

Let $p_{i,j}$ denote the link failure probability of edge $e_{i,j}$. Then for any path $P = v_1, v_2, \ldots, v_q$ (where $v_i \in V$ for $1 \leq i \leq q$), its failure probability is

$$p = 1 - \prod_{i=1}^{q-1} (1 - p_{i,i+1}) , \quad (5.1)$$

where $p_{i,i+1}$ is the failure probability of edge $e_{i,i+1}$.

5.2.2 Important Social Links

Let set $S = \{\{u_1, w_1\}, \ldots, \{u_m, w_m\}\}$ ($u_i, w_i \in V$) denote the $m$ important social pairs. We aim to maintain social links connecting these $m$ node pairs, i.e., for each node pair, there exists one path with the failure probability less than or equal to a threshold $p_t$. For simplicity, for any pair $\{u_i, w_i\} \in S$, if there exists a path with
the failure probability no larger than \( p_t \), we say the social link connecting the pair \( \{u_i, w_i\} \) meets the **connectivity requirement**.

The wireless links are relatively unstable, under the current network topology \( G = (V, E) \) and link failure probability \( p_{i,j} \), some important social links may fail to meet the connectivity requirement. To maintain these social links, we consider adding more reliable links (e.g., satellite links) inside the network. Each of these links can directly connect two arbitrary nodes in the network, and is assumed to have a failure probability close to 0. We refer to these reliable links as shortcut edges in the sequel.

Note that it is unnecessary to consider any pair in \( S \) that already meets the connectivity requirement. Hence, without loss of generality, we assume that none of the social pairs in the given set \( S \) satisfies its connectivity requirement.

### 5.2.3 Problem Formulation

In this chapter, we aim to maximize the number of important social pairs that meet the connectivity requirement, by adding at most \( k \) shortcut edges to \( G \). Note that the path failure probability is calculated based on Eq. (5.1). To simplify the computation, for each edge \( e_{i,j} \), we define its length \( l_{i,j} = -\ln (1 - p_{i,j}) \) (here \( \ln \) is the natural logarithm). Based on the definition of \( l_{i,j} \), the failure probability of any path \( P = v_1, v_2, \ldots, v_q \) can be rewritten as:

\[
p = 1 - \prod_{i=1}^{q-1} (1 - p_{i,i+1}) = 1 - \prod_{i=1}^{q-1} e^{\ln (1-p_{i,i+1})} = 1 - e^{\sum_{i=1}^{q-1} \ln (1-p_{i,i+1})} = 1 - e^{-\sum_{i=1}^{q-1} l_{i,i+1}}.
\]

(5.2)

Note that \( \sum_{i=1}^{q-1} l_{i,i+1} \) is the length of path \( P \). Then, the problem of finding the path with the lowest failure probability for the pair \( \{u_i, w_i\} \) is equivalent to the problem of finding the shortest path between \( u_i \) and \( w_i \). Accordingly, for any pair of nodes \( \{u_i, w_i\} \), it meets the connectivity requirement, if and only if the length of the shortest path between \( u_i \) and \( w_i \) is no larger than \( -\ln (1 - p_t) \). Since the failure probability of each shortcut edge is assumed to be 0, adding a shortcut edge between any two nodes is equivalent to adding an edge of length 0 between those
two nodes (since its edge length is \( l = -\ln(1 - 0) \)).

In this way, our original problem is equivalent to maximizing the number of important social links with distance no larger than \( d_t = -\ln(1 - p_t) \), by adding at most \( k \) shortcut edges (i.e., edges with length 0) to the graph. For simplicity, we say a pair \( \{u_i, w_i\} \) satisfies the distance requirement, if the distance between \( u_i \) and \( w_i \) is no larger than \( d_t \). Given a set \( F \subseteq V \times V \) of shortcut edges with length 0, we define function \( \sigma(F) \) as the number of pairs in \( S \) that meet the distance requirement in \( G' = (V, E \cup F) \). Then, the problem of Maintaining Social Links (MSL problem) is defined as follows:

**Definition 7.** Given an undirected graph \( G = (V, E) \), length \( l_{i,j} \) on each edge \( e_{i,j} \in E \), a distance threshold \( d_t \) (i.e., distance requirement), and a set of \( m \) important social pairs \( S \), find a set \( F \subseteq V \times V \) of at most \( k \) shortcut edges with length 0, such that \( \sigma(F) \) is maximized.

Note that in the case where \( m \leq k \) (i.e., when the number of important social pairs is less than the number of shortcut edges), this problem is trivial. This is because we can directly connect each important social pair by one shortcut edge. Hence, in this chapter, we consider the problem under the constraint that \( m > k \), where each shortcut edge is devoted to potentially benefit the communication of more than one important social pair.

### 5.3 MSL with a Common Node

To prove the NP-hardness of the MSL problem, in this section, we consider a special case of the general MSL problem: maintaining social links with a shared common node (MSL-CN), which is proved to be NP-hard. Since the MSL problem is at least as hard as the MSL-CN problem, the MSL problem is also NP-hard. The discussion of the MSL-CN problem also provides insights into understanding the approximation solution to the MSL problem.

#### 5.3.1 NP-Hardness

In MSL-CN, all important social pairs share a common node. For example, if \( S = \{\{u, w_1\}, \{u, w_2\}, \{u, w_3\}\} \), all important social pairs share the common node
Let node $u$ denote the shared common node among the important social pairs, we have the following lemma:

**Lemma 21.** For every instance of MSL-CN, there exists an optimal solution $F^*$, where every shortcut edge $f \in F^*$ is incident to $u$. Furthermore, for any pair $\{u, w_i\} \in S$, there exists a shortest $uw_i$-path which only uses at most one shortcut edge in $F^*$.

The proof of above lemma is similar to the proof of Lemma 1 in [68], thus omitted for conciseness. Basically, the above lemma states that there exists an optimal solution $F^*$, where all shortcut edges in $F^*$ are incident to the common node $u$. Hence, constructing $F^*$ with $k$ shortcut edges is equivalent to finding optimal $k$ distinct endpoints, each of which connects to $u$ to form a shortcut edge in $F^*$.

Let $F^* = \{\{u, f_1\}, \ldots, \{u, f_k\}\}$ denote the aforementioned optimal solution, where $f_i \in V$ for $1 \leq i \leq k$. Let $W = \{w_i | \{u, w_i\} \in S\}$, i.e., $W$ consists of all the endpoints of the important social pairs except for the common node $u$. For any node $w_i \in W$, if there exists $\{u, f_j\} \in F^*$, such that the distance between $f_j$ and $w_i$ is no larger than $d_i$, then the pair $\{u, w_i\}$ meets the distance requirement.

Consider placing a shortcut edge between node $v_i$ and the common node $u$. Let set $C_i$ denote all the nodes in $W$ whose distance to $v_i$ is no larger than $d_i$. Then the MSL-CN problem is equivalent to finding at most $k$ sets from $C = \{C_1, \ldots, C_n\}$ to cover the maximum number of nodes in $W$.

Based on the above argument, the MSL-CN problem is precisely the maximum coverage problem [73]. The maximum coverage problem is proved to be NP-hard, and accordingly the MSL-CN problem is also NP-hard. Furthermore, the following theorem holds.

**Theorem 10.** The MSL problem is NP-hard.

*Proof.* Since the MSL-CN problem is a special case of the MSL problem, the MSL problem is at least as hard as the MSL-CN problem. Recall MSL-CN is NP-hard; therefore, MSL is also NP-hard. $\square$
5.3.2 A Solution to the MSL-CN Problem

In this subsection, we present an approximation solution to the MSL-CN problem. We first show that MSL-CN can be formulated in terms of submodular functions, and then present a greedy algorithm which can achieve \((1 - \frac{1}{e})\)-approximation of the optimal.

Before presenting the details of the greedy algorithm, we give the definition of submodular functions. A set function \(\psi: 2^C \rightarrow \mathbb{R}\) is submodular, if for any \(X \subseteq Y \subseteq C\) and any \(x \in C \setminus Y\), the following inequality is always satisfied:

\[
\psi(X \cup \{x\}) - \psi(X) \geq \psi(Y \cup \{x\}) - \psi(Y).
\]

In other words, for submodular functions, if \(X\) is a subset of \(Y\), the marginal benefit of adding an element \(x\) to \(X\) is at least as large as the marginal benefit of adding \(x\) to \(Y\).

Now we show that MSL-CN can be formulated in terms of submodular functions. For any \(X \subseteq C\) (recall \(C = \{C_1, C_2, \ldots, C_n\}\)), let \(\psi(X)\) denote the number of nodes covered by \(X\). For any \(X \subseteq Y \subseteq C\), the nodes covered by \(X\) is always a subset of the nodes covered by \(Y\). Then for any \(C_i \notin Y\), the number of overlapping nodes between \(C_i\) and \(X\) is always less than or equal to the number of overlapping nodes between \(C_i\) and \(Y\). Hence, the number of additional nodes covered by adding \(C_i\) to \(X\), is at least as large as the number of additional nodes covered by adding \(C_i\) to \(Y\). Thus, we can get \(\psi(X \cup \{x\}) - \psi(X) \geq \psi(Y \cup \{x\}) - \psi(Y)\), and MSL-CN can be formulated in terms of submodular functions.

The greedy algorithm performs quite well in solving the problems that can be formulated in terms of submodular functions. For our MSL-CN problem, the greedy algorithm works as a multi-round selection process. At each round, for each unselected set \(C_i\) from \(C\), we calculate the number of nodes it can cover in \(W\). After that, the set that covers the maximum number of nodes is selected. Then, the selected set is removed from \(C\), and the covered nodes are removed from \(W\). This selection process runs until \(k\) sets have been selected or all nodes have been covered. We have the following theorem, which states that the performance of the greedy algorithm is theoretically bounded.

**Theorem 11.** Let \(\psi(F^*)\) denote the number of nodes covered in the optimal solu-
tion of MSL-CN problem, and let $\psi(\tilde{F})$ denote the number of nodes covered in the solution given by the greedy algorithm. We have

$$\psi(\tilde{F}) \geq (1 - \frac{1}{e}) \cdot \psi(F^*)$$

That is, the solution of the greedy algorithm achieves $(1 - \frac{1}{e})$-approximation of the optimal solution. The proof of the theorem is similar to the proof in [74], thus omitted for conciseness.

## 5.4 Maintaining Social Links (MSL)

In this section, we present our solution to the MSL problem, which can no longer be formulated in terms of submodular functions. Consequently, the previous approximation results in Theorem 11 may not be directly applicable. Nevertheless, we exploit the Sandwich Approximation strategy (SA) [72] to derive an efficient algorithm with provable approximation ratio, even when the problem is not submodular.

### 5.4.1 Submodularity of MSL

In this subsection, we show that the MSL problem may not be formulated in terms of submodular functions. Consider a network $G = (V, E)$, where $V = \{v_1, v_2, v_3\}$, and $E = \emptyset$. Let the set of important social pairs $S = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}$, and distance threshold $d_t = 1$. Consider the following counter-example, where $x = f_{1,2}$ (i.e., the shortcut edge between $v_1$ and $v_2$), $X = \emptyset$, and $Y = \{f_{2,3}\}$. Then, we have $\sigma(X \cup \{x\}) - \sigma(X) = 1 < \sigma(Y \cup \{x\}) - \sigma(Y) = 2$, which contradicts the definition of submodular functions in Eq. (5.3).

### 5.4.2 Sandwich Approximation Strategy

Since the MSL problem is not submodular, in this subsection, alternatively, we exploit the Sandwich Approximation (SA) strategy [72] to propose an approximation algorithm. Suppose we have two submodular functions $\mu$ and $\nu$, which upper and lower bounds function $\sigma$ everywhere, i.e., $\mu(F) \leq \sigma(F) \leq \nu(F)$ for all $F \subseteq V \times V$
(recall $\sigma$ is the mapping between solution $F$ and the number of pairs in $S$ that meet the distance requirement). Apparently, for problems with objective function $\mu$ or $\nu$, greedy algorithm gives a solution that achieves $(1 - \frac{1}{e})$-approximation of the optimal. Then, exploiting $\nu$ and $\mu$, we aim to derive an efficient approximation algorithm for $\sigma$ (i.e., for MSL problem).

Formally, let $\mu$ and $\nu$ be submodular functions defined on the same ground set $V \times V$, such that $\mu(F) \leq \sigma(F) \leq \nu(F)$ for all $F \subseteq V \times V$. We consider the problem of maximizing $\sigma(F)$, with cardinality constraint $k$ on $F$ (i.e., $|F| \leq k$). Let $\tilde{F}_\mu$, $\tilde{F}_\sigma$ and $\tilde{F}_\nu$ be the solution of the greedy algorithm for functions $\mu$, $\sigma$ and $\nu$, respectively. We then select our solution to the MSL problem as

$$F_{\text{sand}} = \arg\max_{F \in \{\tilde{F}_\mu, \tilde{F}_\sigma, \tilde{F}_\nu\}} \sigma(F). \tag{5.4}$$

For $F_{\text{sand}}$, we have the following theorem which states that its performance can be theoretically bounded, as proved in [72].

**Theorem 12.** For the solution $F_{\text{sand}}$ defined in Eq. (5.4), $\sigma(F_{\text{sand}})$ satisfies

$$\sigma(F_{\text{sand}}) \geq \max \left\{ \frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}, \frac{\mu(F^*_\sigma)}{\sigma(F^*_\sigma)} \right\} \cdot \left( 1 - \frac{1}{e} \right) \cdot \sigma(F^*_\sigma), \tag{5.5}$$

where $F^*_\sigma$ is the optimal solution maximizing $\sigma$ subject to cardinality constraint $k$.

Note that in the above theorem, the data-dependent approximation ratio involves the optimal solution $F^*_\sigma$, which may not be obtained in polynomial time. Hence, in practice, we have $\sigma(F_{\text{sand}})$ bounded by the term $\frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}$ rather than

$$\max \left\{ \frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}, \frac{\mu(F^*_\sigma)}{\sigma(F^*_\sigma)} \right\},$$

that is

$$\sigma(F_{\text{sand}}) \geq \frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)} \cdot \left( 1 - \frac{1}{e} \right) \cdot \sigma(F^*_\sigma). \tag{5.6}$$

The real challenge of applying SA to our MSL problem lies in finding the submodular function $\nu$ and $\mu$ that upper and lower bounds $\sigma$ everywhere. Furthermore, the achieved theoretical approximation ratio is also closely related to how tightly $\sigma$ is bounded by $\nu$ and $\mu$. We will present the upper and lower bound function in the next two subsections.
5.4.3 Lower Bound Function

In this subsection, we present the lower bound function $\mu$. We first show that $\mu$ lower bounds $\sigma$ everywhere, then prove that function $\mu$ is submodular.

Recall that function $\sigma$ maps $F$ to the number of pairs in $S$ that meet the distance requirement. To construct the lower bound function $\mu$, we add another restriction on top of $\sigma$, i.e., each path can use at most one shortcut edge. After adding this restriction, for the pairs that originally go through multiple shortcut edges in their shortest path, their pairwise distance increases. Thus, after adding this restriction, less number of important social pairs meet the distance requirement. Accordingly, $\mu$ lower bounds $\sigma$ everywhere.

On the other hand, under function $\mu$, each pair in $S$ can use at most one shortcut edge. For an arbitrary shortcut edge $f_{i,j}$, let set $S_{i,j} \subseteq S$ denote the pairs in $S$ that satisfy the distance requirement after adding $f_{i,j}$ to $G$. Then, for any set of shortcut edges $X \subseteq V \times V$, after adding $X$ to $G$, the set of pairs that meet the distance requirement is:

$$\theta(X) = \bigcup_{f_{i,j} \in X} S_{i,j}.$$  

Based on the above definition, for any two sets of shortcut edges $X$ and $Y$ satisfying $X \subseteq Y \subseteq V \times V$, we have $\theta(X) \subseteq \theta(Y)$. Consider adding one more shortcut edge $f_{i,j}$, where $f_{i,j} \in V \times V \setminus Y$. Since $\theta(X) \subseteq \theta(Y)$, we have

$$\theta(\{f_{i,j}\}) \cap \theta(X) \subseteq \theta(\{f_{i,j}\}) \cap \theta(Y).$$

Based on the definition of $\theta$ and $\mu$, $\mu(X) = |\theta(X)|$ holds for all $X \subseteq V \times V$. Consequently, we have $\mu(\{f_{i,j}\} \cup X) - \mu(X) \geq \mu(\{f_{i,j}\} \cup Y) - \mu(Y)$, and function $\mu$ is submodular.

5.4.4 Upper Bound Function

We present the upper bound function $\nu$ in this subsection. Basically, the upper bound function $\nu$ corresponds to a maximum coverage problem, where we try to cover the endpoints of the important social pairs by adding shortcut edges.

Consider a set of shortcut edges $F$, and a pair $\{u_i, w_i\} \in S$. We say node $u_i$ ($w_i$) is covered by $F$, if there exists an endpoint $f_j$ of a shortcut edge in $F$, such
that the distance between \( f_j \) and \( u_i \) (or \( w_i \)) is less than \( d_i \). Recall that the distance between any pair in \( S \) is greater than \( d_i \). Then for any pair \( \{u_i, w_i\} \in S \) that meets the distance requirement, both \( u_i \) and \( w_i \) must be covered by \( F \).

For any pair \( \{u_i, w_i\} \in S \), define the weight of \( u_i \) (or \( w_i \)) to be half of the number of times \( u_i \) (or \( w_i \)) appears in \( S \). For example, when \( S = \{\{u_1, w_1\}, \{u_1, w_2\}\} \), \( u_1 \) has a weight of 1, while both \( w_1 \) and \( w_2 \) have a weight of 0.5. The weights are specifically defined in this way, such that when the endpoints of the pairs in \( S \) are all distinct, the weighted sum of node pairs that meet the distance requirement is exactly the number of pairs that meet the distance requirement.

We first show that \( \nu \) upper bounds \( \sigma \) everywhere. We begin by considering a very simple case: suppose for one specific \( F \), only one pair \( \{u_i, w_i\} \in S \) meets the distance requirement. Then at least \( u_i \) and \( w_i \) are covered by \( F \), and \( \nu(F) \) must be larger than or equal to the weighted sum of \( u_i \) and \( w_i \). That is, \( \nu(F) \geq 0.5 + 0.5 = 1 \). Since only one pair meets the distance requirement, we have \( \sigma(F) = 1 \) and \( \nu(F) \geq \sigma(F) \). Furthermore, consider an arbitrary set of shortcut edges \( F \). For all the pairs that meet the distance requirement, let \( \kappa(F) \) denote the weighted sum of their endpoints. Based on the definition of the weight of the node in this subsection, we have \( \kappa(F) \geq \sigma(F) \). Also recall \( F \) at least covers the endpoints of the pairs that meet the distance requirement, we have \( \nu(F) \geq \kappa(F) \geq \sigma(F) \).

Then we show function \( \nu \) is submodular. Note that function \( \nu \) corresponds to the weighted maximum coverage problem. In Subsection 5.3.2, we have proved that the maximum coverage problem (or equivalently the MSL-CN problem) can be formulated in terms of submodular functions. Since the maximum coverage problem is a special case of the weighted maximum coverage problem, i.e., the weight of each node equals 1, the previous proof can be easily extended here. Hence, function \( \nu \) is submodular.

### 5.5 Performance Evaluations

#### 5.5.1 Evaluation Setup

We perform extensive simulations on synthetic graphs as well as a real-world social network dataset from the SNAP project [75]. To validate how tightly we can
bound the performance of our proposed algorithm, various results on the theoretical approximation ratio are presented. In addition, we also compare our solution with a randomized selection solution.

For synthetic graphs, we select the Random Geometric (RG) model, which has been widely used to model the topology of wireless ad hoc networks [76]. The RG graph also resembles a social network, since it spontaneously demonstrates the community structure and displays the degree assortativity (i.e., popular nodes are more likely to be connected to other popular nodes). In a RG graph, nodes are uniformly distributed in a square. Two nodes are connected by an edge if their distance is smaller than a threshold. In our simulation, all nodes are uniformly distributed in a 500 × 500 square-meter area. Two nodes are connected by an edge if their distance is less than 100 meters. For any edge $e_{i,j}$, its link failure probability $p_{i,j}$ is uniformly distributed between 0 and 0.7. The $m$ important social pairs are randomly selected from the nodes in the network.

Besides the RG graph, we also evaluate on a real-world social network dataset: the Gowalla dataset from the SNAP project. The dataset is collected from a location based social network called Gowalla, where users share their locations by check-ins. The dataset contains 196,591 users, 950,327 undirected friendships, and 6,442,890 check-ins of these users over the period of Feb. 2009 - Oct. 2010. For our simulations, we focus on the nodes that have a check-in between 6pm and midnight of Oct. 1st, near Austin, Texas. Two nodes are connected by an edge, if their distance is less than 200 meters based on the locations of their check-ins. The failure probability of each edge is assumed to be uniformly distributed between 0.15 and 0.7. Besides, all node pairs with friendships are chosen to be important social pairs.

5.5.2 Theoretical Approximation Ratio

To evaluate how tightly we can bound the performance of the proposed solution $F_{\text{sand}}$, we present results on the theoretical approximation ratio in this subsection. Specifically, we compute the value of the term $\frac{\sigma(F_{\nu})}{\nu(F_{\nu})}$. Basically, given $\frac{\sigma(F_{\nu})}{\nu(F_{\nu})}$, our solution $F_{\text{sand}}$ at least achieves $\frac{\sigma(F_{\nu})}{\nu(F_{\nu})} \cdot (1 - \frac{1}{e})$-approximation of the optimal.

Table 5.1 presents the theoretical approximation ratio on the RG graph, when
Table 5.1: $\frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}$ for Random Geometric graph

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_t$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.04</td>
<td>0.3636</td>
<td>0.2353</td>
<td>0.1905</td>
<td>0.1600</td>
<td>0.1379</td>
</tr>
<tr>
<td>0.08</td>
<td>0.1333</td>
<td>0.0909</td>
<td>0.0769</td>
<td>0.0667</td>
<td>0.0588</td>
</tr>
<tr>
<td>0.11</td>
<td>0.3000</td>
<td>0.2222</td>
<td>0.1935</td>
<td>0.1714</td>
<td>0.1538</td>
</tr>
<tr>
<td>0.14</td>
<td>0.2727</td>
<td>0.1935</td>
<td>0.1714</td>
<td>0.1538</td>
<td>0.1500</td>
</tr>
<tr>
<td>0.18</td>
<td>0.4348</td>
<td>0.3750</td>
<td>0.3333</td>
<td>0.3000</td>
<td>0.3000</td>
</tr>
</tbody>
</table>

Table 5.2: $\frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}$ for Gowalla Dataset

<table>
<thead>
<tr>
<th>$\nu$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_t$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.23</td>
<td>0.2128</td>
<td>0.1954</td>
<td>0.1762</td>
<td>0.1683</td>
<td>0.1683</td>
</tr>
<tr>
<td>0.27</td>
<td>0.2577</td>
<td>0.2269</td>
<td>0.2269</td>
<td>0.2067</td>
<td>0.2024</td>
</tr>
<tr>
<td>0.31</td>
<td>0.3626</td>
<td>0.3291</td>
<td>0.3291</td>
<td>0.3135</td>
<td>0.3077</td>
</tr>
<tr>
<td>0.35</td>
<td>0.5707</td>
<td>0.5330</td>
<td>0.5330</td>
<td>0.5128</td>
<td>0.5081</td>
</tr>
</tbody>
</table>

$n = 100$, $m = 17$, the path failure probability $p_t$ ranges from 0.04 to 0.18, and the number of shortcut edges $k$ ranges from 2 to 10. As reported in Table 5.1, the value of $\frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}$ is mostly greater than 0.1, where the largest value is around 0.43 (marked in bold). Furthermore, with the increase of $k$, the approximation ratio decreases. Intuitively, this is because as $k$ increases, the MSL problem becomes more complicated, and it is more likely for the function $\mu$ and $\nu$ (i.e., the upper and lower bound function) to deviate from $\sigma$. Consequently, the approximation ratio declines as $k$ grows.

Table 5.2 reports the approximation ratio on the Gowalla dataset. We focus on the nodes that have a check-in between 6pm and midnight of October 1st, near Austin, Texas. Two nodes are connected by an edge if their distance is smaller than 200 meters. The corresponding network contains 134 nodes and 1886 edges. It consists of 63 friendships, and thus there are 63 important social pairs. The table presents the value of $\frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}$, when $p_t$ ranges from 0.23 to 0.35, and $k$ ranges from 2 to 10. We observe that our proposed algorithm works even better on this real-world dataset. The value of $\frac{\sigma(\tilde{F}_\nu)}{\nu(\tilde{F}_\nu)}$ is greater than 0.2 for most cases, and can be as large as 0.57 (marked in bold).
5.5.3 Comparing with the Randomized Selection Solution

In this subsection, we compare our solution with a randomized selection solution. In this randomized selection solution, $k$ shortcut edges are randomly added to the graph, and the number of important social pairs that meet the connectivity requirement (or equivalently distance requirement) is recorded. Then, this process of randomly adding shortcut edges is repeated for 500 times, and the one with the best performance is selected to be the final solution.

Fig. 5.1 compares the shortcut edge placement of our proposed algorithm with the randomized selection solution based on a snapshot of the Gowalla dataset, where $n = 23$ and $p_t = 0.25$. The indices of some nodes are given in the graph, the 7 important social pairs are given in Table 5.3, where each column in the table represents one important social pair. The black lines in the figure are the original wireless links, while the green dashed lines are the shortcut edges. In Fig. 5.1(a), after adding the shortcut edges, pairs $\{v_2, v_4\}$, $\{v_2, v_8\}$, $\{v_7, v_4\}$ and $\{v_7, v_8\}$ are connected by shortcut edges, while pair $\{v_2, v_{10}\}$ is connected by one shortcut edge and one wireless link. Our proposed algorithm is able to maintain 5 important social pairs. In contrast, in Fig. 5.1(b), under the randomized selection solution, 3 important social pairs meet the connectivity requirement, where pairs $\{v_4, v_{22}\}$ and $\{v_2, v_8\}$ are connected by shortcut edges, and pair $\{v_2, v_{10}\}$ is connected by one shortcut edge and one wireless link.
Table 5.3: Important Social Pairs

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_2$</td>
<td>$v_2$</td>
<td>$v_2$</td>
<td>$v_2$</td>
<td>$v_4$</td>
<td>$v_4$</td>
<td>$v_7$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>$v_8$</td>
<td>$v_{10}$</td>
<td>$v_{11}$</td>
<td>$v_7$</td>
<td>$v_{22}$</td>
<td>$v_8$</td>
</tr>
</tbody>
</table>

Fig. 5.2 shows the number of important social pairs that meet the connectivity requirement as a function of path failure probability $p_t$ and the number of shortcut edges $k$. The solid lines stand for the proposed solution, while the dashed lines represent the randomized selection solution. Fig. 5.2(a) presents the results on the RG graph, when $n = 100$ and $m = 17$. As expected, the number of important social pairs that meet the connectivity requirement increases with the number of shortcut edges, and our solution always outperforms the randomized selection solution. In our solution, when $p_t$ is relatively small (e.g., $p_t \approx 0.03$), the number of pairs that meet the connectivity requirement equals the number of shortcut edges added to the graph. This is because when $p_t$ is small, the failure probability of each edge is larger than the threshold of the path failure probability. Then, each pair can only meet the connectivity requirement when its endpoints are directly connected by a shortcut edge. Accordingly, each shortcut edge may only enable one pair to meet the connectivity requirement. As $p_t$ increases, more edges in the original graph can be utilized, and each shortcut edge can benefit more important social pairs. Consequently, the number of important social pairs that meet the connectivity requirement increases quickly as $p_t$ increases.

Fig. 5.2(b) shows the results on the Gowalla dataset when $n = 134$ and $m = 63$. As can be seen, our solution always outperforms the randomized selection solution. Besides, the number of important social pairs that meet the connectivity requirement increases with $p_t$ and $k$. When $p_t$ is relatively small, the number of pairs that meet the connectivity requirement is close to the number of shortcut edges. As $p_t$ increases, this number grows rapidly. When $p_t$ is relatively large (i.e., $p_t = 0.4$), in our proposed solution, by only adding 8 shortcut edges, almost all the important social pairs (58 out of 63) can meet the connectivity requirement.
5.6 Related Work

Due to the node mobility and the transmission interference, the wireless links in mobile ad hoc networks are relatively unstable. To facilitate the user communication and data forwarding, various routing techniques have been proposed. For example, multipath routing has been considered in [77, 78], which selects a few redundant paths to forward the data, so as to increase the delivery reliability and resilience to node failures. For sparse mobile ad hoc networks, a “carry-and-forward” strategy is used to forward data to nodes with higher forwarding capabilities [49, 79]. Social knowledge has also been exploited to improve the performance of data forwarding [4, 2, 5], where the data is forwarded to nodes that have a closer social relationship with the destination. However, in all these previous works, the authors only considered supporting the data forwarding process with existing resources based on current network topology. In real scenarios, it is possible that the underlying communication network might be unable to support the desired QoS for the nodes in the network.

There are some existing works that have considered proactively adding shortcut edges to the existing communication network to improve the network connectivity or shortening the pairwise distance. For example, the authors discussed the problem of augmenting the edge-connectivity between all pairs of nodes, by adding new edges to the graph [67]. In [68], the authors investigated the problem of minimizing
the diameter of the network (i.e., the longest distance between any two nodes in the network), by adding at most $k$ shortcut edges. In [69], the authors studied the problem of minimizing the average shortest path distance over all pairs of nodes in the network, by adding at most $k$ shortcut edges. In all these previous works, the shortcut edges are devoted to benefit the communication between all pairs of nodes in the network. The authors fail to realize that some of pairs are much more important than the others, and the shortcut edges should be used to maintain the connection between these important social pairs. Furthermore, different from previous works, we aim to maximize the number of important social pairs that can meet the desired QoS requirement, rather than minimizing the average distance/diameter. Hence, the previous results may not be directly applied.

In [71], the authors investigated the problem of maintaining important social links through an amplify-and-forward strategy. Although their strategy also aims to maintain the communication between important social pairs by altering the underlying communication network, it is limited to maintain the connection between only one important social pair. While in this chapter, we present a more practical strategy, which simultaneously maintains the connection between multiple important social pairs.

## 5.7 Conclusion

In this chapter, we focused on maintaining social links through direct link placement. We formulated and studied the problem of maintaining social links using a limited number of shortcut edges, and proposed an approximation algorithm with provable approximation ratio based on the sandwich approximation strategy. Evaluation results based on synthetic graphs as well as a real-world social network dataset validate the effectiveness of the proposed solution. Furthermore, although our algorithm was proposed to solve the problem of maintaining social links, our solution could provide insights into the general shortcut edge addition problem in any graphs.
Conclusions and Future Work

6.1 Summary

In this dissertation, we designed efficient resource management strategies such as content caching, transmission scheduling and link placement to improve the network capacity and maintain important social links. We summarize the main results as follows.

In Chapter 2, we studied the capacity of wireless networks with caching. We proved that for nodes uniformly distributed on the surface of sphere, the network capacity is upper bounded by $\Theta(W\sqrt{s}/m)$. We also proposed a caching scheme, based on which a capacity of $\Theta(W^{\frac{s}{m}})$ is achievable. More importantly, our results suggest that through caching, it is possible for nodes to obtain constant capacity even if the number of nodes increases.

In Chapter 3, we studied scaling laws of network capacity based on the skewness of content popularity. We found that as the distribution of the content popularity changes from uniform distribution to more skewed distributions, the network capacity quickly increases from $\Theta(\sqrt{s})$ to roughly $\Theta(\sqrt{s})$. Our results suggest that for wireless networks with caching, when contents have skewed popularity, increasing the number of nodes monotonically increases the per node capacity. We also proposed a distributed caching algorithm which enables nodes to optimally cache contents and maximize the per-node capacity only based on local knowledge.

In Chapter 4, we applied the amplify-and-forward strategy to maintain social links in wireless networks. To save energy at the relays, we formulated and
studied the Min-Energy problem. We formulated it as a non-convex quadratic programming problem by exploiting the rate-distortion theory, and solved it based on an approximation technique. We also proved that our solution can minimize the power consumption of the relays. The Min-Energy problem involves all relays, which could generate significant synchronization overhead. To minimize the synchronization overhead, we also studied the Min-Relay problem which aims to minimize the number of active relays while maintaining the social link. We formulated the Min-Relay problem as an integer programming problem, and proposed a polynomial-time algorithm which can select the minimum number of relays to maintain the social link. Evaluation results showed that Min-Relay can significantly reduce the number of active relays compared to Min-Energy, while achieving comparable power consumption.

In Chapter 5, we focused on maintaining social links through direct link placement. We formulated and studied the problem of maintaining social links using a limited number of shortcut edges, and proposed an approximation algorithm with provable approximation ratio based on the sandwich approximation strategy. Evaluation results based on synthetic graphs as well as a real-world social network dataset validate the effectiveness of the proposed solution. Furthermore, although our algorithm was proposed to solve the problem of maintaining social links, our solution could provide insights into the general shortcut edge addition problem in any graphs.

6.2 Future Directions

Our work to date has provided efficient resource management strategies to improve the network capacity and maintain important social links. Besides the research issues we have discussed, there are other research directions worth in-depth investigation. Next, we outline several directions for future work that one could pursue.

- Content access delay for caching in wireless ad hoc networks: So far, our work on caching in wireless ad hoc networks has focused on the effect of caching on network capacity. Another interesting direction that has yet been
explored is the effect of caching on content access delay. Intuitively, caching can significantly reduce the content access delay, since nodes can access their desired contents from neighbors rather than the congested server. Although the benign effect of caching on the content access delay is well expected, analytically quantifying the effect of caching on content access delay requires the accurate modeling of the request processing at each node and the content transmission in the wireless network. To be more specific, the content access delay is composed of two parts: the queueing delay while waiting for the content request to be processed, and the transmission delay when the content is being retrieved. The queueing delay at each node can be derived based on the request arrival rate and processing rate at each node. While for the transmission delay, given the transmission scheduling strategy, it can be computed according to the number of active transmissions in the network. Although the analysis of content access delay is quite challenging, it can provide valuable insights into the design of caching strategies for delay-sensitive applications, like video streaming.

• Caching strategy for dynamic content popularity: In our studies, the analytical results are derived when the content popularity changes slowly or even remains unchanged. Yet, a more interesting and practical question is: what is the optimal caching strategy to maximize the network capacity under dynamic content popularity? The real challenge of this problem lies in the fact that the computational cost to accurately model the behavior of a caching system is exponential in both the cache size and the number of contents. Hence, in most of the cases, it will be computationally infeasible to derive the optimal caching strategy. One way to approach this problem might be start from some simplified assumptions (e.g., assume the content popularity follows the shot noise model) and utilize approximation techniques (e.g., Che’s approximation) to design an efficient and accurate solution. Then, try to apply and extend the results to more general scenarios.

• Maintaining social links to meet both throughput and robustness requirements: In our work on maintaining social links through direct link placement, the objective is to improve the link robustness, i.e., to improve the probabil-
ity that data is successfully delivered on the social links. Yet, in some cases, it is also critical to take the throughput requirements of the social links into consideration. To be more specific, although sharing the same communication link among multiple social links with low throughput requirements may not degrade the performance, sharing the communication link among social links with high throughput requirements will result in congestion and longer delay. Hence, a shortcut edge placement algorithm which jointly considers the link robustness and throughput requirement will be a promising direction for future work.


[76] **Ma, L., T. He, K. K. Leung, A. Swami, and D. Towsley (2013)** “Identifiability of link metrics based on end-to-end path measurements,” in *ACM IMC*.


Vita

Li Qiu

Li Qiu received the B.E. degree in Electronic Engineering from Tsinghua University, Beijing, China, in 2012. He enrolled in the Ph.D. program in Computer Science and Engineering at The Pennsylvania State University in August 2012.

Publications during the Ph.D. study:


- Li Qiu and Guohong Cao, “Popularity-Aware Caching Increases the Capacity of Wireless Networks”, IEEE International Conference on Computer Communications (INFOCOM), 2017.

- Li Qiu and Guohong Cao, “Cache Increases the Capacity of Wireless Networks”, IEEE International Conference on Computer Communications (INFOCOM), 2016.