ABSTRACT

Title of Dissertation: Heterogeneous Wireless Networks: An Analysis of Network and Service Level Diversity

Ulaş C. Kozat, Doctor of Philosophy, 2004

Dissertation directed by: Professor Leandros Tassiulas
Department of Electrical and Computer Engineering

Future wireless systems will be a collection of symbiotic and hierarchical networks that address different aspects of communication needs. This architectural heterogeneity constitutes a network level diversity, where wireless domains can benefit from each other's spare resources in terms of bandwidth and energy. The dissertation investigates the network diversity through particularly interesting scenarios that involve capacity-limited multi-hop ad hoc networks and high-bandwidth wired or wireless infrastructures.

Heterogeneity and infrastructures not only exist at the level of networking technologies and architectures, but also at the level of available services in each network domain. Efficient discovery of services across the domains and allocation of service points to individual users are beneficial for facilitating the actual communication, supplying survivable services, and better utilizing the network resources. These concepts

together define the service level diversity, which is the second topic studied in our dissertation.

In this dissertation, we first focus on a large-scale hybrid network, where a relatively resource abundant infrastructure network overlays a multi-hop wireless network. Using a random geometric random graph model and defining appropriate connectivity constraints, we derive the overall transport capacity of this hybrid network.

In the sequel, we dwell upon hybrid networks with arbitrary size and topology. We develop a Quality of Service (QoS) based framework to utilize the joint resources of the ad hoc and infrastructure tier with minimal power exposure on other symbiotic networks that operate over the same radio frequency bands. The framework requires a cross-layer approach to adequately satisfy the system objectives and individual user demands. Since the problem is proven to be intractable, we explore sub-optimal but efficient algorithms to solve it by relying on derived performance bounds.

In the last part of the dissertation, we shift our attention from network level diversity to service level diversity. After investigating possible resource discovery mechanisms in conjunction with their applicability to multi-hop wireless environments, we present our own solution, namely Distributed Service Discovery Protocol (DSDP). DSDP enables a highly scalable, survivable, and fast resource discovery under a very dynamic network topology. It also provides the necessary architectural and signaling mechanisms to effectively implement resource allocation techniques.

Heterogeneous Wireless Networks: An Analysis of Network and Service Level Diversity

by

Ulaş C. Kozat

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Professor Leandros Tassiulas, Chair
Professor Anthony Ephremides
Professor John Baras
Professor Udaya A. Shankar
Professor Richard La

DEDICATION

Anneme, babama ve eşim Burçak'a

for my beloved wife Burçak and my parents

ACKNOWLEDGMENTS

I would first like to thank Prof. Leandros Tassiulas for giving me the opportunity to pursue a career in wireless networks area. He contributed to my academic formation in many different ways, however, I am most thankful to him for providing me the academic freedom to determine my own research direction and for his timely feedbacks that assist me in finishing my Ph.D. program successfully. I am equally thankful to Prof. Can Korman, who supported my academic development from the start of my graduate life. I also want to note here that Prof. Pickholtz, Prof. Ephremides, Prof. Narayan, and Prof. Baras inspired my academic and professional life with their discipline as well as their dedication to the telecommunications field.

I would like to thank George Kondylis, Bong Ryu, and Srikanth Krishnamurthy for giving me the first real collaborative research environment at HRL Laboratories that extraordinarily contributed to my future research during a short and yet productive time period. I would also like to thank Mariusz Fecko and Sunil Samtani from Telcordia Technologies who provided me with the opportunity to implement the service discovery portions of my thesis research as a real-time test-bed and for supporting my research initiatives.

I shall also acknowledge friends who made my graduate student years more enjoyable. Fatih Alagöz is the first of my colleagues and my close friends in the U.S.

Gün Akkor has been there with me all the time and deserves a special thanks. Iordanis Koutsopoulos has always been a very good friend and a valuable colleague with whom I had the most fruitful discussions.

Most importantly, I would like to thank my family for their greatest support and encouragement. Burçak, your love is the source of my strength.

TABLE OF CONTENTS

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Wireless Hybrid Networks 3

1.1.1 Overlaid Wireless Systems 4

1.2 Service Discovery 9

1.2.1 Client-Server Paradigm 11

1.2.2 Directory Systems 12

1.3 Organization of the Thesis 14

2 Asymptotic Throughput Capacity of Large Hybrid Networks 18

2.1 Introduction 18

2.2 Overview 21

2.2.1 Capacity of Wireless Ad Hoc Networks 22

2.2.2 Capacity of Hybrid Networks 26

2.3 System Model 27

2.4 Capacity Improvement with Infrastructure Layer 30

2.4.1 Throughput Upper-bound under Strong Connectivity 30

2.4.2 Achievability of $\Theta[W/\log(N)]$ 32

2.5 Looser Connectivity Conditions and Achievability of Constant Capacity per Node 40

2.6 Achievability of the Capacity in the case of Weak Connectivity 50

2.7 Relaxing the Assumptions 52

2.7.1 Capacity Bound for the Constrained Infrastructure 53

2.7.2 Revisiting the Assumption $K = \Theta(N)$ 55

2.7.2.1 Strong Connectivity Results vs. Access Point Population 55

2.7.2.2 Weak Connectivity Results vs. Access Point Population 57

2.8 Summary 59

3 Hybrid Networks as Finite Arbitrary Graphs 62

3.1 Cross-layer Interaction 64

3.2 Related Works 67

3.3 System Model 70

3.3.1 Channel Model 71

3.3.2 SINR threshold and Feasibility of Concurrent Transmissions . . . 74

3.3.3 Notion of Virtual Links 77

3.3.4 Hybrid Network Communication Scenarios 78

3.4 Joint Power Allocation, Schedule and Path Assignment Problem in two tier Hybrid Networks 80

3.4.1 Formal Problem Statement 80

3.4.2 Intractability of the Jointly Feasible Schedule and Power Allocation 84

3.4.3 Performance Bounds 86

3.5 Sub-Optimal Approximation Algorithms for $P2$ 89

3.5.1 Pseudo-polynomial Approximation: Solving $P1$ 89

3.5.1.1 Algorithm A 89

3.5.1.2 Algorithm B 92

3.5.2 Polynomial Approximation Algorithm for $P2$ 94

3.6 Experimental Evaluation 97

3.6.1 Solving $P1$: Algorithm A versus Algorithm B 97

3.6.2 Solving $P2$: Pseudo-polynomial versus Polynomial Algorithms . 103

3.7 Summary 109

4 Service Discovery in Wireless Hybrid and Ad Hoc Networks 111

4.1 Service Discovery for Multi-hop Wireless Networks 113

4.2 Network Model and Notation 117

4.2.1 Network Model 117

4.2.2 Notation and Definitions 117

4.3 A Directory Architecture Solution for Service Discovery 119

4.3.1 BackBone Management (BBM) Phase 120

4.3.1.1 Backbone Selection 121

4.3.1.2 Mesh Formation 125

4.3.1.3 Backbone Maintenance 127

4.3.2 Distributed Service Discovery (DSD) Phase 128

4.4 Simulation Environment 134

4.4.1 Performance Metrics 134

4.4.2 Simulation Model 135

4.4.3 Simulation Results 139

4.5 Integration of DSDP Solutions with Other Layers 150

4.5.1 Application of DSDP to Reliable Server Pooling 151

4.5.2 Integration with Routing Protocols: Case Study with OLSR . . . 153

4.6 Summary 155

5 Conclusion 158

5.1 Suggestions for Further Study 161

A Proofs of Lemmas in Chapter 2 164

A.1 Proof of Lemma 3 164

A.2 Proof of Theorem 3 165

Bibliography 177

LIST OF TABLES

4.1 Pseudo-code for backbone selection phase. 122

LIST OF FIGURES

1.1 Abstraction of a multi-tier hybrid network architecture. Enumerated bidirectional arrows indicate possible wireless interfaces established between wireless domains via hybrid devices that carry multiple radio technologies. 8

1.2 Architectural models and signaling for (a) Directoryless and (b) Directory systems. 11

2.1 Two-tier network topology: Circles and squares represent the ad hoc nodes and access points respectively. Ad hoc nodes can access to the infrastructure network at random locations via access points. Access points are connected via an infrastructure network that is capable of carrying any amount of traffic between any pair of access points. 19

2.2 Triangular inequality requires that any two receivers must be separated by at least Δr_T, i.e. the disks with radius $\Delta r_T/2$ and centers located at receivers must be disjoint. 29

2.3 Formation of a Voronoi tessellation on a disk domain with radius R. Each Voronoi cell can be sandwiched between disks of radius ϵ and 3ϵ. 35

2.4 Representing non-increasing sequences by differentiable functions with continuous first order derivatives. 48

3.1 Sample topology and scheduling for concurrent multi-hop sessions. . . . 73

3.2 SINR threshold over different paths for the same session depends on the specific coding, modulation level and path length. Above we plot the SINR threshold ratios against $-\ln(BER_2)$ when different paths with hop distance ratios h_2/h_1 are utilized with fixed modulation levels and uncoded transmissions under independence assumption. BER_2 is the per link bit error rate requirement that corresponds to the SINR threshold γ_2 under the same end-to-end packet error rate requirement. 75

3.3 Block Diagram for Algorithm A. 90

3.4 Block Diagram for Algorithm B. 92

3.5 Block Diagram for Algorithm C. 95

3.6 An assignment instance generated by Algorithm C in a multi-hop cellular topology with fixed base-station locations at the center of square cells and randomly distributed relay/source nodes over a 1000mx1000m topology. The scenario involves 15 sessions with varying error rate requirements and outdoor shadowing environment. 96

3.7 Ratio of feasible scenarios for 7 sessions and minimum-hop routing. . . . 100

3.8 Total transmit power averaged over the feasible scenarios for 7 sessions and minimum-hop routing. 100

3.9 Ratio of jointly feasible scenarios for 7 sessions and minimum-power routing. 101

3.10 Total transmit power averaged over the feasible scenarios for 7 sessions and minimum-power routing. 101

3.11 Ratio of the feasible scenarios for 15 sessions and minimum-hop routing. 102

3.12 Total transmit power averaged over the feasible scenarios for 15 sessions and minimum-hop routing. 102

3.13 Ratio of jointly feasible scenarios for 10 sessions: Algorithms A and B are applied onto a single path, which is the same as the minimum-power path to the overlay tier. 104

3.14 Total transmit power averaged over the feasible scenarios. 105

3.15 Total transmit power averaged over the feasible scenarios common to all algorithms at the load of 10 sessions. 106

3.16 Ratio of jointly feasible scenarios for 10 sessions. 107

3.17 Ratio of jointly feasible scenarios for 7 sessions. 108

3.18 Total transmit power averaged over the feasible scenarios common to all algorithms at the load of 10 sessions. 108

4.1 An instance of virtual backbone formation. 126

4.2 Source based multicast tree formation on virtual backbone with service registration message. 131

4.3 General view of service discovery with TTL-limited registration and request messages. 133

4.4 Control message overhead for 10-user case. 140

4.5 Ratio of successful requests for 10-user case. 140

4.6 Average delay comparison for 1-server/10-user case. 141

4.7 Average delay comparison for 3-server/10-user case. 141

4.8 Average delay comparison for 5-server/10-user case. 142

4.9 Control message overhead for 3-server case. 142

4.10 Ratio of successful requests for 3-server case. 143

4.11 Average delay comparison for 3-server/20-user case. 143

4.12 Average delay comparison for 3-server/30-user case. 144

4.13 Control message overhead for different TTL values with mean request inter-arrival times of 4 and 8 seconds. 146

4.14 Control message overhead for different TTL values with 100 mobile nodes and mean request inter-arrival time of 4 seconds. 147

4.15 Ratio of successful requests for different TTL values with mean request inter-arrival times of 4 and 8 seconds. 148

4.16 Ratio of successful requests for different TTL values with 100 mobile nodes and mean request inter-arrival time of 4 seconds. 148

4.17 Average delay for different TTL values with mean request inter-arrival time of 4 and 8 seconds. 149

4.18 Average delay for different TTL values with 100 mobile nodes and mean request inter-arrival time of 4 seconds. 149

Chapter 1

Introduction

As wireless networks become more popular and new application areas emerge, we see an ever-increasing number of radio-technology, protocol, architecture, and service level proposals that address specific needs of military, education/research institutes, industry, enterprises and individuals. Sometimes, as we see in the evolution of WLANs and micro-cell cellular data networks [1], these technologies can be seen as competitors for the same market segment. Most of the time, however, they are developed as complementary technologies that co-exist together. Although the seamless operation among these complementary networks is important for ubiquitous communications [2], the ultimate question is about how much we can gain by making the resources of each network available to each other and how we can utilize the existing heterogeneity.

In this dissertation, we see bandwidth and services -former at the lower layers and the latter at the higher layers- as the two main resources that any network segment can offer to their own users as well as to other segments. Bandwidth in bits/sec is a more tangible resource than the services, which may assume highly abstract forms. For instance, when two wireless networks that can interchange data packets through multiple-interface nodes co-exist in a locality, depending on their individual average and instantaneous traffic demand, one could have spare capacity to share with the other one. How much traffic can be carried out over such hybrid system, to what level of coordination is required to provide an adequate level of quality of service, how the interaction among different tiers should develop are all relevant questions that are addressed in the dissertation.

On the other hand, as services are becoming the focal points of the data communication, independent of the networking technologies, they flourish as a new layer of infrastructure to be efficiently utilized by wireless users. The abstractness of services as a resource stems from the difficulty in classifying what they are, how they should be located, assigned, or even accessed. From our perspective, services -some of which has yet to be defined- will be very generic. Nonetheless, some concrete server examples that are essential for the operations of a real network can be listed as configuration servers (e.g. DHCP [3] or DRCP [4]), public key managers, bandwidth brokers, and domain gateways. Servers do not necessarily belong to a specific wireless domain or are not necessarily related to the functioning of the network. They can be as diverse as multi-media libraries, printers, and combat command centers. In this dissertation, we mainly address locating services in terms of the network level identifiers, i.e. IPv4 addresses, of the specific servers and setting a framework that will allow to implement server allocation mechanisms in a distributed and localized way.

The rest of this chapter provides a brief overview of the most relevant hybrid wireless network proposals as well as some preliminaries on service discovery mechanisms. We then outline the organization of the remaining chapters by highlighting their specific contributions to the wireless literature.

1.1 Wireless Hybrid Networks

Wireless hybrid networks can be conceptualized as symbiotic or overlaid systems. Symbiotic systems refer to the availability of multiple wireless domains that can operate stand-alone, but they can also access to each other's available bandwidth resources in a symmetric relation. Two ad hoc networks that operate on IEEE 802.11b radio interfaces but at different frequencies with some nodes carrying both interfaces at the same time or Bluetooth piconets may establish such a symbiotic relation.

Overlaid systems, however, exhibit a hierarchical structure and an asymmetric relation among different wireless tiers. Lower tiers have limited bandwidth resources and they utilize the relatively abundant resources of upper tiers via specific access points. This architectural model is the main focus of our dissertation. The design philosophy of overlaid networks may furthermore follow two different perspectives: (i) Ad hoc networks as extensions to the wireless access networks and (ii) overlay networks to improve the performance and connectivity of ad hoc network partitions. Although we adopt the latter design philosophy, our problem formulations and findings can also be generalized to the former after minor modifications. Below, we provide an overview of the existing proposals for wireless hybrid networks.

1.1.1 Overlaid Wireless Systems

Significant amount of research is invested in scenarios, where multi-hop wireless ad hoc networks are visualized at the edge of wireless access networks to shift their traditional single-hop wireless architectures toward multi-hop wireless architectures. The motivation can be expressed as the expectations over significant improvements in performance in terms of system throughput, user capacity, wireless coverage and power consumption [5, 6, 7, 8].

In one of the earliest works on the throughput performance of multi-hop wireless hybrid networks [5], the authors adopt the classical hexagonal cell architecture with base stations positioned at the center of each cell. Within each domain, it is assumed that medium access is resolved via the distributed coordination function (DCF) of IEEE 802.11, which belongs to the CSMA/CA family of protocols. Multi-hop architecture is formed by simply reducing the transmission power at base stations and mobile nodes by a constant factor of k_p compared to the original transmission power (P_t) of nodes in the single-hop access. Each cell is assumed to have non-interfering channel reuse. Different from standard WLANs and cellular networks, nodes are permitted to communicate with other nodes within the same cell without going through the base station. However, any packet transmitted for a different cell has to pass through the base stations of the source and destination nodes respectively. When the transmit power is not enough to reach the base station, nodes use multiple hops, using other users as their relays. Authors obtain analytical results for per-hop and end-to-end throughput performance with the following conventions: (i) Only the contention for the channel determines the successful packet transmission and (ii) there always exists a relay node toward the base station at a distance of transmission range. Noting that this is an over-simplified model and that it does not fully capture the real WLAN environments for which IEEE 802.11 is developed [9], the results indicate significant improvements in throughput as k_p increases. In other words, multi-hop wireless architecture performs better than single-hop wireless architectures.

In a similar vein, but more concerned with the actual system implementation, authors in [7] outline a multi-hop architecture for next generation *Global System for Mobile Communication (GSM)* networks. They propose to enhance the existing GSM protocol stack and signaling such that a native multi-hop ad hoc extension can be realized. Their primary motivation is that especially in the urban areas, there exist a lot of dead spots where very weak or no signal at all can be received from the base stations. By allowing GSM devices to use other ad hoc enabled GSM devices, which may have direct or indirect linkage with a base station, as relay nodes, they argue that the coverage, network utilization, and end-to-end throughput can be significantly improved. Authors define a beaconing mechanism that enables ad hoc nodes to discover each other and their association with any base stations. These Adhoc-GSM nodes closely monitor the channel quality with other Adhoc-GSM nodes and the base station. They also specify a handover mechanism for altering between base stations and relay nodes. Results shown in [7] shows that an end-to-end throughput (i.e. ratio of the packets that are successfully reached to base station) improves significantly (8-17 percent) as the number of dead spots increases. The proposal does not consider or prohibit an ad hoc communication mode where Adhoc-GSM nodes can directly communicate with each other locally without using the base station.

With a more elaborate model, SOPRANO proposal [8] works on a CDMA-TDD based system, where wireless routers are placed in each cell to facilitate cell-splitting without any wires. Similar to other system models, upstream and downstream transmissions are allowed to pass through multiple hops until packets reach to the base station and mobile host respectively. To avoid self-interference, wireless routers are prohibited to simultaneously transmit and receive in the same frequency band. Accordingly, time division duplexing (TDD) is used at wireless routers, e.g. base station and wireless routers transmit in different slots for the down-stream traffic. The authors develop joint power control and routing strategies under the assumption that instantaneous channel capacity is achieved [10]. They show as much as 33% throughput gain over the single hop transmissions to the best base station in terms of path gain.

In another interesting proposal, authors suggest to use CDMA-based cellular networks to solve the last mile problem for WLANs [6]. In this respect, a hybrid device, which supports both cellular and WLAN radio interfaces, acts as an access point for WLAN users. A certain amount of IP address pool is allocated to each of these wireless gateways (or bridges), and PPP connection is established over the cellular link layer. This system model effectively exposes a 2-hop wireless architecture that increases the coverage area and the utilization of cellular networks while solving the last mile problem for locations that lack broadband connections in a cost-effective way.

There are also other proposals that unite the single-hop cellular architectures with non-native multi-hop wireless ad hoc networks such as UCAN [11] and iCAR [12]. In

UCAN, mobile clients are assumed to have both 3G cellular link and IEEE 802.11-based peer-to-peer links. Routes are established between mobile hosts and the base station by using the channel gain over the cellular link as the path metric. The routing path is selected as the one with the maximum channel gain over the first link from base station to proxy node, which is the only link that actually uses 3G interface and subsequent transmissions are routed using IEEE 802.11 interfaces. This opportunistic routing strategy is accompanied with scheduling and secure crediting mechanisms to enforce fairness in the system. iCAR, on the other hand, focuses on the placement of fixed wireless routers to reduce call blocking probability in a circuit-switched system in addition to circumventing blind spots. The users in the congested cells are routed through wireless relays to uncongested cells to increase user capacity of the system.

The proposals summarized so far mainly exploit ad hoc networks to enhance the capabilities of fixed wireless access networks. The opposite view, where the stand-alone ad hoc networks benefit from the existing wired or wireless (even mobile) infrastructures, is also explored especially in the context of tactical networks and sensor networks. For tactical networks, the main objective is to connect different MANET partitions with each other and to command centers via long range-high bandwidth radios, whereas in sensor networks, the general goal is to send as much information as possible to the command centers again via special nodes that constitute a resourceful overlay (e.g. see [13] for SENMA project).

Despite of the numerous publications on hybrid networks, most of the proposals are very recent and a lot of important issues remain uncharted. This dissertation will take up

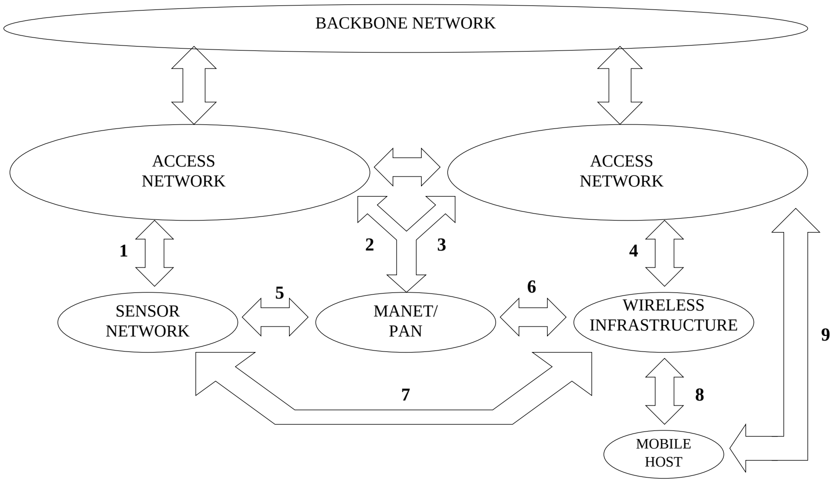


Figure 1.1: Abstraction of a multi-tier hybrid network architecture. Enumerated bidirectional arrows indicate possible wireless interfaces established between wireless domains via hybrid devices that carry multiple radio technologies.

the most essential problems within these unexplored fields and assert that:

1. Analyzing *Transport Capacity* of hybrid architectures over all power control, scheduling, and routing decisions is of a major interest to understand the fundamental limitations on such systems.

2. *Handoff or base station assignment problem* cannot be done by a simple measure of the path gain as in the case of [11] and [7] or cell call blocking probabilities, because even the moderate size of ad hoc network renders the ad hoc tier as the bottleneck. Base station assignments become a critical tool to avoid the bottleneck situations.

3. When the symbiotic systems that operate over the same frequency band co-exist in close proximity, e.g. nearby cells, the goal should not be a brute-force throughput maximization but rather to reduce the interference that emanates outside each ad hoc partition. Such objectives should be accompanied with a level of QoS agreed

upon between the users and network operator. The existence of the infrastructure permits to implement more sophisticated cross-layered approaches to attain the performance goals.

1.2 Service Discovery

Flexibility and minimum user intervention are essential for communication networks that are to be easily deployed and reconfigured automatically when extended with new hardware and/or software capabilities. Service discovery, which allows devices to automatically locate network services with their attributes and to advertise their own capabilities to the rest of the network, is the key technology for such self-configurable networks. Since hybrid networks may consist of infrastructures and ad hoc networks with time-varying topologies, they are the primary customers for service discovery technologies to facilitate any type of interaction/communication within and across the network tiers.

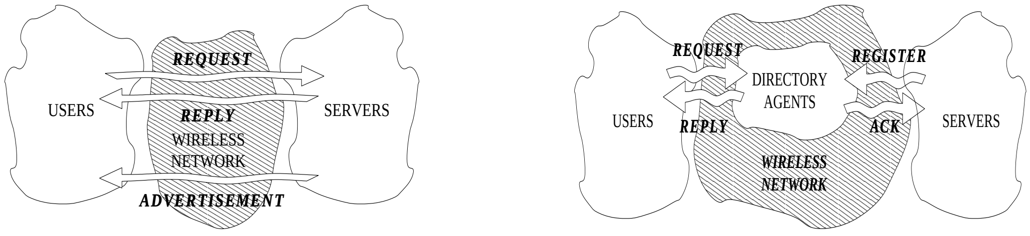
Several different (yet overlapping) industrial consortiums and organizations are established to standardize various service discovery protocols- such as *Service Location Protocol (SLP)* of IETF [14], Sun's *Jini* [15], Microsoft's *Universal Plug and Play (UPnP)* [16], IBM's *Salutation* [17], and Bluetooth's *Service Discovery Protocol (SDP)* [18]. Nonetheless, these standardization endeavors do not directly dwell on mobile ad hoc network (MANET) or hybrid wireless network environments that stretch over multiple wireless hops.

Ad hoc nodes may have very little or no knowledge at all about the identities and capabilities of each other. There can in fact be a high degree of variety in terms of the capabilities of each individual device (e.g. support of multiple physical interfaces, processing power, printing capability, multi-media libraries, etc.) and such a heterogeneity renders it even more attractive to establish an ad hoc network.

In this dissertation, we adopt an abstract view on service discovery without paying particular attention to how the service types and attributes are defined. Instead, we concentrate on the available design choices and evaluate them both in terms of their limitations and their interactions with the lower layers to assess the overall performance in the context of multi-hop wireless networks. The following definitions clarify how services and service discovery must be understood in our work.

- Service is any hardware or software feature that can be utilized or benefited by a wireless user.
- Server is any ad hoc node that has at least one service to offer to the other nodes.
- User (or Client) is any ad hoc node that wants to utilize a specific service offered in the network.
- Service Discovery is a mapping from a service class and an attribute list to a single IP address or a group of IP addresses.

Hence, a gateway node that inter-connects a wireless network to other wireless/wireline networks, a back-up storage device, or a special purpose sensor node (e.g. surveillance camera) may all be regarded as servers of different types under this generic view.



(a) Directoryless Architecture.

(b) Directory Architecture.

Figure 1.2: Architectural models and signaling for (a) Directoryless and (b) Directory systems.

Service discovery is traditionally performed in the application layer transparent from the lower layers. Nevertheless, the architectural design and mandatory control message signaling of a specific protocol impose certain requirements on the network layer and below. In return, the applicability and the performance of a specific protocol in multi-hop wireless networks at a large extent depend on these requirements. Our objectives in this dissertation in the context of service discovery are two-fold: (i) to carefully examine the implications of various design choices on the overall communication cost and (ii) to design a scalable, fast and survivable service discovery protocol for wireless networks.

The following section presents a brief background on how the service discovery is performed in current proposals.

1.2.1 Client-Server Paradigm

Most service discovery protocols include the client-server paradigm as a mode of operation. In this paradigm, service users (i.e. clients) reactively send out service request messages and servers listen to such messages at a well-determined network interface and port. If the requested service is supported, then a reply message is generated and sent back to the clients. Service users can also passively learn about the available services by listening to service advertisements that are pro-actively generated at the servers. We refer to such client-server paradigm based protocols as *directoryless* systems in regard to the directory systems as defined in the next section.

The simplicity of directoryless systems is often cited to attribute the *light-weight* feature to them [19]. Indeed, these architectures do not carry the burden of introducing intermediary agents and they are quite efficient in networks that consist of a few nodes. Nevertheless, all of these proposals require network layer support in terms of broadcasting and multicasting, which may be quite costly to implement as network size gets larger especially in wireless environments.

1.2.2 Directory Systems

The alternative scheme involves service brokers (or directory agents) which reside between clients and servers as a logical entity. Clients direct their requests to well-known service brokers whereas servers register their services with these brokers. In return, service brokers send back service reply messages to the clients and registration acknowledgments to the servers. Since the location of directory agents may be unknown initially, a hunting procedure for the directory agents must be engaged. For example, IETF’s service discovery proposal SLP relies on multicasting to find out the actual identities of the directory agents. Service requests are sent to individual agents to retrieve the actual service records and directory agents are queried in sequence by the users until the request is resolved. Directory agents may have more functionality than keeping a database of available services. For instance, they may also supply the objects to access a particular service as in Jini [15] or provide bridging services between domains as in Salutation [17].

Directory systems are preferred in general because of the following advantages they offer: 1) Scalability is achieved when network size becomes larger. 2) Response time for locating services decreases. 3) Servers are not flooded with service requests when there is a high demand for certain type of services. 4) Directory agents can apply simple load balancing techniques before sending back a reply message. This further reduces the load on individual servers and enhances the service performance.

The main disadvantage of the existing proposals that support directory architecture is that they are mainly designed for wired networks with infrequent topology changes. Thus, they treat directory agents as fixed, pre-configured devices. Multi-hop wireless environments, especially when mobility exists, cannot make such an assumption. In the evolution of time, wireless domains may partition or merge and each connected domain should be able to access the available services. Along these lines, implementing a survivable directory system in multi-hop wireless networks requires making directory agent overlay adaptive against the topology changes. The means of accomplishing this task as a viable alternative to the directoryless systems is one of the main subjects of the dissertation.

Figure 1.2 summarizes the architectural and signaling choices for the service discovery protocols. The detailed discussion on how to implement them in wireless and mobile network scenarios is further supplemented in chapter 4.

1.3 Organization of the Thesis

Most of our results in this dissertation have been presented previously. Chapter 2 was presented in part at the Ninth Annual International Conference on Mobile Computing and Networking (ACM MobiCom 2003) [20] and also has been accepted for publication in ACM/Kluwer Wireless Networks (WINET) Journal [21]. Chapter 3 will appear in part at the 23rd Conference of the IEEE Communications Society Infocom 2004 [22]. Chapter 4 was presented in part at IEEE International Conference on Communications [23] and at the 22nd Conference of the IEEE Communication Society Infocom 2003 [24]. It was also published in Ad Hoc Networks Journal [25]. The service discovery proposal in the thesis has also been adapted for Reliable Server Pooling at Telcordia Technologies, where a real time test-bed was built. The same test-bed was exhibited at Military Communications Conference Milcom 2003. Below, we provide a brief outline of the main chapters of the dissertation.

In chapter 2, we consider the transport capacity of ad hoc networks with a random flat topology under the present support of an infinite capacity infrastructure network. Such a network architecture allows ad hoc nodes to communicate with each other by purely using the remaining ad hoc nodes as their relays. In addition, ad hoc nodes can also utilize the existing infrastructure fully or partially by reaching any access point (or gateway) of the infrastructure network in a single or multi-hop fashion. Using the same tools as in [26], we show that the per source node capacity of $\Theta(W/\log(N))$ can be achieved in a random network scenario with the following assumptions: (i) The number of ad hoc nodes per access point is bounded above, (ii) each wireless node, including the access points, is able to transmit at W bits/sec using a fixed transmission range, and (iii) N ad hoc nodes, excluding the access points, constitute a connected topology graph. This is a significant improvement over the capacity of random ad hoc networks with no infrastructure support which is found as $\Theta(W/\sqrt{N\log(N)})$ in [26]. Although better capacity figures may be obtained by complex network coding or by exploiting mobility in the network, infrastructure approach provides a simpler mechanism that has more practical aspects. We also show that even when less stringent requirements are imposed on topology connectivity, a per source node capacity figure that is arbitrarily close to $\Theta(1)$ cannot be obtained. Nevertheless, under these weak conditions, we can further improve per node throughput significantly. We also provide a limited extension on our results when the infrastructure is topologically constrained or when the number of ad hoc nodes per access point is not bounded.

In chapter 3, we turn our attention to finite networks with arbitrary topologies and arbitrary number of access points. In this new network model, we also allow individual users to define their quality of service demand in terms of desired end-to-end bandwidth resources and packet losses. Considering the fact that the efficient use of energy is of paramount importance in multi-hop wireless networks both because of the exposed interference and battery powered wireless nodes, our objective becomes minimum power emanation outside each ad hoc domain. Since power expenditure and connection quality depend on mechanisms that span several communication layers due to the existing co-channel interference among competing flows that must reuse the limited radio spectrum, our solution framework is characterized by a synergy between the physical and the medium access control (MAC) layers with a view towards inclusion of higher layers as well. More specifically, we address the joint problem of power control, scheduling, and access point assignment with the objective of minimizing the total transmit power subject to the end-to-end quality of service (QoS) guarantees for sessions in terms of their bandwidth and bit error rate guarantees. Bearing to the NP-hardness of this combinatorial optimization problem, we propose our heuristic solutions that follow greedy approaches.

In chapter 4, we look at more practical aspects of hybrid networking in terms of locating network resources within and across the wireless domains. Our study as presented in the dissertation is probably the first extensive examination of service discovery problem in the context of wireless ad hoc and hybrid networks. In this chapter, we discuss possible service discovery architectures along with the required network support for their implementation, and we propose a distributed service discovery architecture which relies on a virtual backbone for locating and registering available services within a dynamic network topology. Our proposal consists of two independent components: (i) formation of a virtual backbone and (ii) distribution of service registrations, requests, and replies. The first component creates a mesh structure from a subset of a given network graph that includes the nodes acting as service brokers and a subset of paths (which we refer as *virtual links*) connecting them. Service broker nodes

(SBNs) constitute a dominating set, i.e. all the nodes in the network are either in this set or only one-hop away from at least one member of the set. The second component establishes sub-trees rooted at service requesting nodes and registering servers for efficient dissemination of the service discovery probing messages. We provide extensive simulation results for comparison of performance measures, i.e. latency, success rate, and control message overhead, when different architectures and network support mechanisms are utilized in service discovery. Our results indicate that directory systems, which can be discarded as heavy weight at first sight, can in fact be an effective and efficient mean of providing resource discovery.

Finally, in chapter 5, we draw the future direction of research in hybrid wireless networks and conclude the dissertation.

Chapter 2

Asymptotic Throughput Capacity of Large Hybrid Networks

2.1 Introduction

This chapter investigates the theoretical gains of a two-tier hybrid wireless network in terms of its throughput capacity and scalability. Along these lines, hybrid network consists of an ad hoc component with limited resources and a wireless or a wired infrastructure with relatively abundant resources in terms of bandwidth, energy, buffer space, and processing power. The transactions between the two tiers are carried out by a set of access points that are equipped with both ad hoc and infrastructure network interfaces (see figure 2.1).

We define our problem on a disk domain as it is widely accepted in the literature [26, 27, 28]. Both the ad hoc nodes and the access points of the infrastructure network are assumed to be randomly distributed on this disk domain. Furthermore, these nodes can transmit up to a fixed transmission range. On the other hand, the infrastructure topology is conditioned to be capable of matching even the highest bandwidth demand

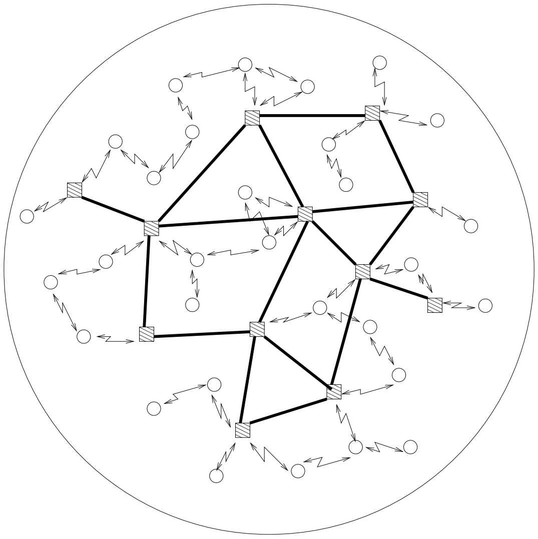


Figure 2.1: Two-tier network topology: Circles and squares represent the ad hoc nodes and access points respectively. Ad hoc nodes can access to the infrastructure network at random locations via access points. Access points are connected via an infrastructure network that is capable of carrying any amount of traffic between any pair of access points.

that can occur between any pair of access points.

The choice of random location for ad hoc nodes is a natural one. However, it is a good practice to question the appropriateness of imposing the same assumption on the access points. As a contrary example, consider the case where we have a cellular network overlay. Then, the access points are simply the base stations located at the center of hexagonal cells and they are connected to each other by a wire-line network. Hence, the locations of the access points are deterministic by construction. On the other hand, if we have wireless local area networks (WLANs) as the infrastructure, the shape of the serving areas as well as the location of each access point are not well-determined [9]. Furthermore, when we consider the access points to be mobile/wireless routers with broadband connections to the infrastructure network, our randomness assumption becomes more sound. Although we do not have control over the location of access points, we may have control over their population. For most part of our analysis, we will explicitly use the number of access points in the derivations. Although it is equally interesting to investigate the region where the number of ad hoc nodes per access points is unbounded, our attention will be on the systems where the number of access points is in the same order as the number of ad hoc nodes.

Along these lines, we define three layers of graphs $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$ as follows:

$\mathcal{G}_1(V_1; r_T)$ is the graph formed by taking V_1 -the set of ad hoc node positions- as the vertex set and including edges between each pair x, y of distinct points in V_1, which has an Euclidean distance less than or equal to r_T.

$\mathcal{G}_2(V_2; r_T)$ is the graph formed by taking V_2 -the set of ad hoc node and access point positions- as the vertex set and including edges between each pair x, y of distinct points in V_2, which has an Euclidean distance less than or equal to r_T.

$\mathcal{G}_3(V_3; r_T)$ is the graph formed by denoting V_3 -the set of ad hoc nodes, access points, and the internal nodes of the infrastructure network- as the vertex set. The edges given by distinct node pairs include all the edges in $\mathcal{G}_2(V_2; r_T)$ plus all the edges between internal nodes of the arbitrary infrastructure tier.

The analysis in this chapter can be divided into two parts. In the first part, we obtain the throughput capacity under a notion of *strong connectivity* condition, which mandates that $\mathcal{G}_1$ becomes a connected topology graph almost surely. In other words, we want to have a stand-alone ad hoc network that can provide connection between any pair of ad hoc nodes with probability arbitrarily close to one and without the support of any existing infrastructure. This certainly is a very cautious constraint and does not entirely take advantage of the existing infrastructure. For instance, there can be partitions in the ad hoc tier, but when the overall topology construct is visualized, any pair of ad hoc nodes can still be connected. Therefore, at the expense of partitions, ad hoc nodes can reduce their transmission range below the value enforced by the strong connectivity. This eliminates excessive interference of ad hoc nodes on each other and increases the number of simultaneous transmissions in the ad hoc tier while improving the upper bound of the transport capacity. Hence, in the second part, we introduce the second notion of connectivity, *the weak connectivity*, that requires the overall network topology graph $\mathcal{G}_3$ to be connected and we further explore the throughput capacity of hybrid networks under this new notion.

For a comprehensive treatment of the subject, it is essential to discuss the existing models and architectures before describing our own system model and analysis framework.

2.2 Overview

Although the transport capacity in the context of wireless ad hoc networks and single hop cellular networks has been widely investigated, it remains relatively untouched in the case of hybrid networks. This section will critically engage with the nascent literature on the capacity of wireless ad hoc and hybrid networks as it relates to the objectives and methods of this thesis.

2.2.1 Capacity of Wireless Ad Hoc Networks

Transport capacity of wireless ad hoc networks has been a major research interest since the landmark paper of Gupta and Kumar [26]. In that paper, authors prove that per node throughput values of $\Omega(1/\sqrt{N})$ bit-meters/sec and $\Omega(1/\sqrt{N \log N})$ bits/sec are attainable for arbitrary and random networks respectively both on a planar disk domain and on the surface of a sphere. Achieving the throughput figure for arbitrary networks involves the freedom of placing the nodes and choosing the traffic patterns. On the other hand, random network scenarios encompass a uniform distribution of the nodes on the topology area as well as a random destination for each ad hoc node. Therefore, authors show the achievability results for random networks in the asymptotic sense by designing proper routing and transmission scheduling mechanisms. In [26], two different models are considered for determining the successful transmissions in the same channel: the protocol model and the physical model. In the protocol model, a given transmitter-receiver pair has an acceptable level of communication, only if they are within the transmission range of each other and no other transmitter exists within an interference disk centered at the receiver. The physical model on the other hand demands a certain signal to interference and noise (SINR) ratio threshold for successful transmissions in the multiple access channel. The upper bounds that are derived for both transmission models in arbitrary network and for protocol model in random networks are found to be in the same order of the constructed lower bounds; hence capacity of ad hoc networks as modelled becomes $\Theta(1/\sqrt{N})$ and $\Theta(1/\sqrt{N \log N})$ correspondingly.

Whereas Gupta and Kumar consider only the case of stationary nodes, with the rationale that mobility can only deteriorate the capacity, Grossglauer and Tse [29] demonstrate that mobility can achieve higher rates asymptotically as the number of nodes increases. They assume a stationary and ergodic distribution for the node positions, where the location of a node is uniformly distributed on a disk, and the SINR based physical model for deciding on the successful transmissions. The key point in their analysis is that, when each source or relay node transmits to the closest receiver, SINR requirement for each transmission pair is asymptotically satisfied with a positive probability value. Hence, given that θN nodes are randomly selected as transmitters (where $0 < \theta < 1$), transmitters always choose the closest receiver to send. Since all transmitter-receiver pairs are equally likely to be scheduled, each link is activated with the probability at the order of $\Theta(1/N)$. Authors define a two-round scheduling policy. In the first round, source nodes transmit the pending packets to their closest receiver, which can be a relay or the destination node. In the second round, transmitters, which can be the source or the relay node, forward the packets that have the same destination as their closest receiver. Thus, for any source-destination pair, $(N-2)$ relay nodes receive and transmit packets at rate $\Theta(1/N)$ while source nodes also transmit directly to the destination with $\Theta(1/N)$. Summing over all paths, each flow identified by the source-destination pair acquires a fixed rate, i.e. $\Theta(1)$, that constitutes a significant improvement over the results of Gupta and Kumar.[1] Nevertheless, their result is achieved

[1] Note that this improvement is achieved by effectively reducing the hop distance between source-destination pairs to a constant. Gupta and Kumar's upper-bound result in bits-meter holds under any scenario that is mobile or stationary. If the average minimum distance travelled at each hop is $1/\Theta(\sqrt{N})$, then the upper-bound becomes $\Theta(1)$ without any conflict with Grossglauer and Tse.

at the expense of possibly excessive delays.

Extending on their previous work, Gupta and Kumar also follow an information theoretical perspective to find the sufficient conditions for achieving a rate region by allowing arbitrarily complex network coding [30]. Authors group relay nodes in disjoint sets for each source-destination pair and order them such that lower order sets can only forward data to higher order sets, hence defining a forwarding graph. All possible forwarding graphs are considered to determine the achievable rates. Although their approach is not proved to yield a capacity result, they nevertheless demonstrate that a specific wireless network of N nodes located in a region of unit area can indeed achieve a network throughput of $\Theta(N)$ bit-meters/sec or $\Theta(1)$ bits/sec data rate per node, which is again a remarkable gain over their original capacity results that is inherently limited by the assumed point-to-point communication.

Gastpar and Vetterli too tackle the information theoretical asymptotic capacity of wireless networks, but for the simpler relay case [27]. Different from previous works, they consider only one source-destination pair in their problem setting and remaining $(N-2)$ nodes purely act as relays helping the source node to convey as much information as possible to the destination by repeating the received signal. To make things analytically tractable, authors introduce a slotted scheme, where source node transmits in the even slots and relays repeat the received signal with proper amplification in the odd slots. Unlike [30], the total transmit power of the relays is constrained to be in the same order of the number of ad hoc nodes and no individual relay is allowed to transmit at an unbounded power level as N goes to infinity. Thus, the transmit powers of the relay nodes must be coordinated. The slotted scheme allows to use the separation principle for the source and channel coding despite of the fact that this principle does not hold in general for multi-user communication systems. It is proved that channel capacity behaves at best as $\log(N)$ after imposing an additional constraint of an arbitrarily small but positive separation between the ad hoc nodes.

In a more recent work, Duarte-Melo and Liu address a many-to-one communication paradigm in multi hop sensor networks [31]. They first consider a flat network architecture, in which sensor nodes are assumed to be uniformly distributed on a planar disk domain with a single base station located at the center of the disk. All sensors generate data traffic at the same rate towards this single base station. They adopt the protocol model for packet transmissions and find out the conditions, under which the trivial upper bound $O(W/N)$ cannot be achieved for a given channel bandwidth of W bits/sec. Under the same conditions, they demonstrate that $O(W/2N)$ is asymptotically feasible. Authors then introduce clustering where the base stations are now placed on equally separated grid points. Each sensor directs its traffic towards the closest base station. Base stations forward the sensory data again to a central node using a wireless channel non-interfering with the transmissions within the clusters. Furthermore, assuming that there is no interference between the clusters, authors illustrate that the trivial upper bound can be asymptotically achieved.

As it is clear from our overview, network capacity can be drastically improved, when mobility, network coding, redundant relay nodes and/or clustering are effectively exploited. However, we instead work on a new perspective that searches for the achievable wireless network capacity when an infrastructure network support is available at random ingress and egress points to the ad hoc users. Such provisioning reduces the burden on the ad hoc tier in terms of the coordination overhead, in comparison to its alternatives such as complex network coding, adding redundant ad hoc nodes, and clustering.

2.2.2 Capacity of Hybrid Networks

In a very recent work [28], authors investigate the throughput capacity of a hybrid network architecture. In their proposed architecture, the infrastructure network is depicted as a cellular network, where the access points are located at the center of hexagonal cells and are inter-connected via a broadband wireline network. Authors are mainly interested in how the number of access points (hence the hexagonal cells) should scale with the number of ad hoc nodes to gain substantial network capacity improvement over the pure ad hoc operations. They impose different routing strategies that segment the randomly distributed ad hoc nodes into two groups depending on whether they use the cellular network to reach the destination or not. The decision criteria in forming the groups rely on heuristic arguments and are not necessarily the optimum routing strategies. Under such circumstances, they show that the number of access points should grow faster than $\sqrt{N}$ to have a noticeable gain. Their results also reveal that if all the bandwidth resources are allocated to the communication through the infrastructure network and the number of access points is in the same order of ad hoc network size, then $\Theta(NW)$ bits/sec can be achieved as the total transport capacity. Note that such an allocation does not support all the source nodes and this capacity is mainly shared among the nodes that are routed through the infrastructure as determined by the routing layer.

Although there is a significant overlap between our network model and that of [28], there are also major differences that underline our own contributions: (1) First of all, as we have already mentioned at the beginning of this chapter, the type of the infrastructure network may not allow a hexagonal cell structure. Assuming random locations for access points can give us a better capacity budget estimate of the scenarios, where the access point locations are not on regular grid points. In fact, we will demonstrate in the following sections that the network capacity of $\Theta(NW)$ bits/sec is not attainable in our random network model. (2) We specify the upper bound of throughput capacity over all routing and transmission strategies. After then, we design a specific routing and transmission scheme to achieve this upper bound. (3) Our constraints in terms of the connectivity requirements on the ad hoc network pose a different problem. (4) We show that the network throughput capacity can be achieved by a fair allocation of bandwidth among all users regardless of their destinations.

Having finished the overview of the related works and identified the distinguishing features of our problem, we may now proceed with the details of our system model.

2.3 System Model

We consider a two-tier architecture, where an ad hoc network is overlaid with an infrastructure network. Ad hoc nodes can communicate with each other along the paths that may reside entirely in the ad hoc tier, i.e. they cross only the ad hoc nodes.

However, they are also allowed to utilize the infrastructure network such that the flow paths can be partially overlapped with the infrastructure nodes and links. We assume that the infrastructure network has a relatively abundant bandwidth and the transmissions within each tier do not interfere with the other one. The access between two tiers is achieved through special nodes, which we refer to as access points or gateway nodes. Without loss of generality and for clarity, access points are assumed only to relay the packets between each tier and they do not generate any data traffic themselves.

We limit our attention to a random network scenario, in which ad hoc nodes and access points are uniformly distributed on a disk of area $A_R = \pi\mathcal{R}^2$, where R is the disk radius[2]. Each ad hoc node generates data traffic of rate $\lambda(N, K)$ bits/sec for a random destination in the ad hoc tier. Here, N and K refer to the number of ad hoc nodes and access points respectively. We assume that the number of ad hoc nodes per access point is bounded and $\lim_{N\to\infty}(N/K) = \alpha$ where $\alpha \in (0, \infty)$. Although the transmission radius of ad hoc nodes is assumed to be fixed, it can be arbitrarily small as N goes to infinity subject to the connectivity constraints.

We assume a total available bandwidth of W bits/sec, which can be carried over multiple orthogonal channels (i.e. frequency band and/or code). The contention over the same channel is resolved in time and space. As a simple interference scheme, we adopt the *protocol model*. Due to this model, transmission from node i to node j in a specific

[2] Although the access points are also physically a part of the ad hoc tier, we functionally treat them different. Unless otherwise is explicitly specified, when we call *ad hoc nodes*, we exclude the access points.

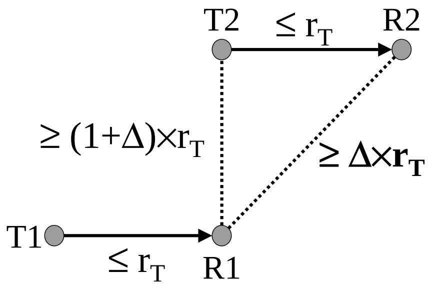


Figure 2.2: Triangular inequality requires that any two receivers must be separated by at least Δr_T, i.e. the disks with radius $\Delta r_T/2$ and centers located at receivers must be disjoint.

combination of ad hoc channel and time slot is called *interference-free* if the following two conditions are satisfied:

(i) Euclidean distance between i and j is smaller than or equal to r_T, i.e. $|X_i - X_j| \leq r_T$, where X_l represents the position vector of node l.

(ii) There are no other transmitters around j at a distance of $r_I = (1 + \Delta) \times r_T$ in the same channel and time slot, where $\Delta \geq 0$.

These two conditions along with the triangle inequality imply that disks of radius $\Delta r_T/2$ centered at the receivers must be disjoint in order to be able to schedule them simultaneously in the same channel and time slot (see figure 2.2). [26]. In the rest of the chapter, r_T and r_I will be used interchangeably with *transmission range* and *interference range* respectively.

The throughput capacity is computed over all possible time-space scheduling of transmissions and flow paths. A per node throughput of $\lambda(N, K)$ is called *feasible* if there exist satisfying time-space scheduling and routing paths with unlimited buffering capabilities in the intermediate nodes. We call the per node throughput capacity of the random network as described to be in the order of $\Theta(f(N,K))$ bits/sec if there are deterministic constants $0 < c < c' < \infty$, such that;

$$\lim_{N\to\infty} Prob(\lambda(N,K) = cf(N,K)\ is\ feasible) = 1\ ,$$

$$\liminf_{N\to\infty} Prob(\lambda(N,K) = c'f(N,K)\ is\ feasible) < 1\ .$$

In the next section, we provide the asymptotic results that capture the benefits of using an infrastructure network even in random scenarios under strong connectivity constraints.

2.4 Capacity Improvement with Infrastructure Layer

The tools to derive the capacity result for our network model will not be very different from the ones already engaged in [26]. We first start with establishing the upper bounds.

2.4.1 Throughput Upper-bound under Strong Connectivity

Using the interference-free transmission model, we can bound the number of simultaneously successful transmissions by the number of disks with radius $\Delta r_T/2$ that can be packed inside the disk of area A_R. However, the boundary effects require modification in our argument: When a receiver is close to the boundary of the disk domain such that the disk with radius $\Delta r_T/2$ is not completely inside the domain, we only need to take into account the fraction that overlaps with the domain. The smallest of such fraction is 0.25 which occurs exactly when the receiver is located on the boundary and $\Delta r_T/2$ is equal to the domain diameter, i.e. $2R$. Hence, the number of simultaneous transmissions must be smaller than $16A_R/(\pi\Delta^2 r_T^2)$. In this respect, given the average number of hops $\bar{h}(N,K)$ within the ad hoc tier, total bandwidth W, and per node throughput $\lambda(N,K)$, the following inequality holds:

$$N\lambda(N,K)\bar{h}(N,K) \leq \frac{16A_R W}{\pi\Delta^2 r_T^2} \ . \tag{2.1}$$

Here, the dependence of $\bar{h}$ on N is a natural consequence of letting transmission range to be smaller as N gets larger, while its dependence on K is the result of routing decisions which may be based on the location and number of the gateway nodes. In the above expressions, maximizing $\lambda(N,K)$ amounts to minimizing both r_T and $\bar{h}(N,K)$, where the latter is clearly dependent on the former in stationary ad hoc networks. Suppose for now that $\bar{h}(N,K) = 1$, which is the best situation we can have, and let us focus on the transmission range. We want to minimize r_T subject to the strong connectivity condition. At this point we can directly use the results from [32] or [33]. To provide a more general picture of the minimal connectivity problem, we present the one by Penrose below.

Theorem 1 (by Penrose [32]). *Suppose $X_1, X_2, \ldots$ are independent random points in $\mathbb{R}^d$, $d \geq 2$, with common density f, having connected compact support Ω with smooth boundary $\partial\Omega$. Assume also that the discontinuity set of f is restricted to Ω is Lebesque null and contains no element of $\partial\Omega$. Let M_N denote the smallest $r_T(N)$ such that the union of balls of diameter r centered at the first N points is connected. Let Θ denote the volume of the unit ball. Then as $n \to \infty$, with probability of one following limit holds:*

$$\frac{N\Theta M_N^d}{\log N} \to \max\left\{\frac{1}{\min_\Omega f}, 2(1-1/d)\frac{1}{\min_{\partial\Omega} f}\right\} \ .$$

In our model, the support for the density function is a disk and it complies with the hypothesis of the theorem. The value of the density function is simply $1/A_R$ over the support and on the boundary. In 2-D, Θ is equal to π and

$$r_T(N) \geq M_N = \sqrt{\frac{A_R \log(N)}{\pi N}} \ , \tag{2.2}$$

to have *strong connectivity* almost surely. Because of the inequality (2.2) and the fact that $\bar{h}(N,K) \geq 1$, with probability of one (as N goes to ∞), the following upper bound is valid under any routing and scheduling decision:

$$\lambda(N,K) \leq \frac{16W}{\Delta^2 \log(N)} \ . \tag{2.3}$$

Next, we will show that $\Theta[W/\log(N)]$ is the actual per node throughput capacity.

2.4.2 Achievability of $\Theta[W/\log(N)]$

Achievability of the upper-bound in (2.3) requires to show the existence of a temporal and spatial scheduling as well as a routing scheme that asymptotically attain the same dependence on N almost surely. The following steps are involved in the construction of this jointly *optimal* scheduling and routing scheme:

(1) We create a Voronoi tessellation[3] on our topology domain, where each Voronoi cell completely covers an area of $100A_R \log(N+K)/(N+K)$. We also set the

[3] Voronoi tessellation on a region is formed by a set of construction points on this region. Each construction point identifies a unique Voronoi cell and all the remaining points on the region are partitioned into disjoint Voronoi cells by assigning each point to the Voronoi cell that has the closest construction point to its own position [35].

transmission range such that any node can directly reach to the other nodes in the same Voronoi cell.

(2) We show that the number of Voronoi cells that interfere with the transmissions of a specific cell is bounded above by a constant C.

(3) We prove that the total number of ad hoc nodes and access points in each Voronoi cell is indeed less than $O(\log(N+K))$.

(4) We demonstrate that each Voronoi cell includes at least one access point.

(5) Finally, we show that the number of destination nodes per access point within a Voronoi cell is $\Theta(1)$.

Before explaining each of these steps in detail, let us jump ahead and first examine their implications in our construction. Suppose that time is divided into slots with fine granularity and that each node utilizes the whole bandwidth W in the time slot it transmits. When steps 2 and 3 are considered together, we can schedule each node in a Voronoi cell, including the access points, without any conflict by assigning $W/[(C+1)\log(N+K)]$ amount of bandwidth to that node. On the other hand, steps 1 and 4 provide us the routing algorithm we search for: (i) If both the source and the destination nodes are in the same Voronoi cell, the source node transmits to the destination node in single hop by using its own share of bandwidth. (ii) Otherwise, the source node can use its share of bandwidth to reach any access point in its own cell. Once the data packets reach to the selected access point, they can be relayed up to one of the access points that share the same Voronoi cell as the destination node without any packet loss. Step 5 ensures that we can assign bounded number of destination nodes to each access point. Hence, each access point divides its bandwidth share further by a constant value. The access points before the destination nodes become the throughput bottleneck; nevertheless, an end-to-end rate of $W/[C_1(\log(N)+\log(1+K/N))]$ per source node is supported. Since this result is asymptotic and $\lim_{N\to\infty} K/N = 1/\alpha$, we have constructed the following lower bound which implies that per node throughput capacity for random network with infrastructure becomes $\Theta(W/\log(N))$:

$$\lambda(N,K) \geq \frac{W}{C_1\left[\log(N)+\log(1+\frac{1}{\alpha})\right]} \ . \tag{2.4}$$

Now, we are ready to proceed with the individual steps to under-fill the result as found in (2.4).

STEP 1:

We repeat various procedures that are already established in [26] for the sake of completeness. Recall that the Voronoi tessellation of a closed region on $\mathcal{R}^2$ is defined by a set of points p on the region. Each Voronoi cell is identified by a point $p_i \in p$ and consists of the set of all nodes that are closer to p_i than any other point in p. Here on, the distance is measured simply in Euclidean distance. We provide a modified version of the lemma from [26] to make it directly applicable to disks in $\mathcal{R}^2$.

Lemma 1. *For every $R \geq \epsilon > 0$, there is a Voronoi tessellation of a disk of radius R in $\mathcal{R}^2$ with the property that each Voronoi cell contains a disk of radius ϵ and is contained in a disk of radius 3ϵ.*

Proof. Let $D(x,\epsilon)$ denotes the disk centered at point x with radius ϵ. We form the tessellation in two rounds. We start the first round with a construction point p_1, which is

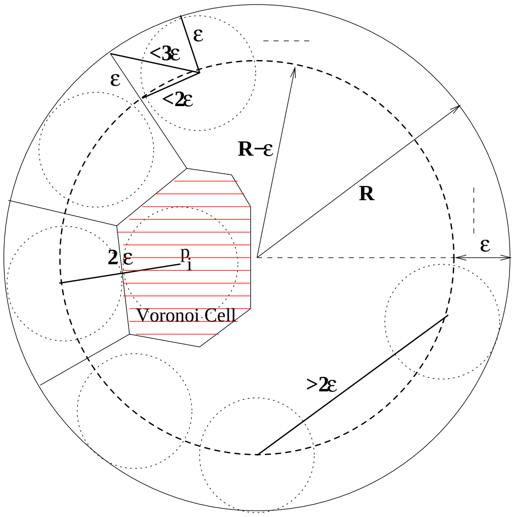


Figure 2.3: Formation of a Voronoi tessellation on a disk domain with radius R. Each Voronoi cell can be sandwiched between disks of radius ϵ and 3ϵ.

exactly at a distance of ϵ from the disk domain boundary (see figure 2.3). Given the first $(i-1)$ points, the next construction point p_i is selected such that the distance between p_i and the disk domain boundary is exactly ϵ while the distance between p_i and any previously selected point is at least 2ϵ. Since these points lie on the finite perimeter of a circle that is concentric with domain disk and has a radius $(R-\epsilon)$, the first round terminates eventually. In the second round, we add a new construction point p_j only on the inner disk of radius $(R-\epsilon)$ and only if $D(p_j,\epsilon)$ does not intersect $D(p_i,\epsilon)$ for the already selected p_is. Since we have a bounded area and each addition of a point removes a finite portion of the available area, from which we can select another point, second round eventually halts. The Voronoi tessellation generated by points p_i satisfies the properties of the lemma. To be precise, suppose that point x is closer to the construction point p_i than to any other construction point. If x lies on the inner disk of radius $(R-\epsilon)$, it is at most 2ϵ away from p_i. Otherwise, it would be at a distance larger than 2ϵ from all construction points and the disk $D(x, \epsilon)$ would not intersect with the disks $D(p_j, \epsilon)$ contradicting to our construction. On the other hand, if x lies outside the disk of radius $(R - \epsilon)$, from triangular inequality, it must be at most 3ϵ away from p_i. It is also clear from our construction that each Voronoi cell covers a disk of radius ϵ; otherwise, at least two disks $D(p_i, \epsilon)$ and $D(p_j, \epsilon)$ for $i \neq j$ would intersect by again violating our construction. □

Thus, when we choose ϵ and the transmission range r_T such that

$$\pi\epsilon^2 = 100 A_R \log(N + K)/(N + K) \text{ and } r_T = 6\epsilon ,$$

lemma-1 guarantees us a tessellation, where each Voronoi cell covers at least an area of $100 A_R \log(N + K)/(N + K)$ and each node can reach to other nodes in the same cell in a single hop. The following steps will provide the basis of designing a joint routing and scheduling scheme built upon this particular tessellation.

STEP 2:

Any Voronoi cell V' interferes with another Voronoi cell V, if V' and V include points that are at most $(r_T + r_I) = (2 + \Delta)r_T$ apart. In the worst case condition, these points can be just on the boundaries of each cell and since the Voronoi cells have a diameter less than or equal to 6ϵ, any interfering cell for V must be located in a region with a radius of $9\epsilon + (2 + \Delta)r_T$. Using the facts that each cell area is lower bounded by $\pi\epsilon^2$ and that we have already set $r_T = 6\epsilon$, there can be at most

$$C = \left\lfloor \frac{\pi(9\epsilon + (2 + \Delta)r_T)^2}{\pi\epsilon^2} \right\rfloor - 1 = \left\lfloor (21 + 6\Delta)^2 \right\rfloor - 1$$

interfering cells in the neighborhood of V. Notice that C is a constant that depends only on the medium access protocol specific parameter Δ. Now, it is a straight-forward application of the graph theory to demonstrate that $(C + 1)$ slots are enough to schedule one transmission for each cell in a conflict-free manner. When each Voronoi cell is represented by a vertex and an edge between any two vertices represents the mutually interfering cells, we encounter a graph coloring problem, where each color corresponds to a different time slot. Since this graph has a maximum degree of C, we can color it with $(C + 1)$ colors at most. The corollary of this result is that we have a scheduling of length $(C + 1)$ slots that can allocate an exclusive slot for each Voronoi cell in a round robin fashion. In each slot, the corresponding cell utilizes the entire bandwidth. We can then introduce sub-slots within each time slot to further allocate an equal amount of bandwidth among the ad hoc nodes and the access points over the ad hoc channels in the same cell. The order of the number of these sub-slots will be the same as the order of the number of users in the same cell, which is obtained in the next step.

STEP 3:

We use the *Vapnik-Chervonenkis Theorem* and a lemma from [26] to prove that each Voronoi cell includes less than $O(\log(N + K))$ nodes.

Theorem 2 (The Vapnik-Chervonenkis Theorem). *If $\mathcal{F}$ is a set of finite VC-dimension VC-d($\mathcal{F}$), and $\{X_i\}$ is a sequence of i.i.d. random variables with common probability distribution P, then for every $\epsilon, \delta > 0$,*

$$Prob\left(\sup_{F\in\mathcal{F}} |\frac{1}{L}\sum_{i=1}^{L} I(X_i \in F) - P(F)| \leq \epsilon\right) > 1 - \delta\ ,$$

whenever

$$L > \max\left(\frac{8VC-d(\mathcal{F})}{\epsilon}\log\frac{16e}{\epsilon}, \frac{4}{\epsilon}\log\frac{2}{\delta}\right).$$

Lemma 2. *The Vapnik-Chervonenkis dimension (VC-d) of the set of disks in $\mathcal{R}^2$ is 3.*

Then, by letting the sequence $\{X_i\}$ be the random positions of ad hoc nodes and access points, L equal to $N+K$, and $\mathcal{F}$ be the set of disks in $\mathcal{R}^2$ with area $900A_R\log(N+K)/(N+K)$ so that the disk area entirely covers a Voronoi cell, we obtain:

$$Prob\left(\sup_{D\in\mathcal{F}}|\frac{Number\ of\ nodes\ in\ D}{N+K} - P(D)| \leq \epsilon\right) > 1-\delta\ , \tag{2.5}$$

whenever

$$N+K > \max\left(\frac{24}{\epsilon}\log\frac{16e}{\epsilon}, \frac{4}{\epsilon}\log\frac{2}{\delta}\right)\ . \tag{2.6}$$

Equation (2.5) implies that;

$$Prob\left(\frac{Number\ of\ nodes\ in\ D}{N+K} \leq \sup_{D\in\mathcal{F}}[P(D)] + \epsilon\right) > 1-\delta\ . \tag{2.7}$$

Evidently, $\sup_{D\in\mathcal{F}}[P(D)] = 900\log(N+K)]/(N+K)$ and setting ϵ and δ equal to $100\log(N+K)/(N+K)$ satisfy (2.6) at least for large $N+K$. Hence, we have

$$Prob\{Number\ of\ nodes\ in\ any\ Voronoi\ cell\ \leq 1000\log(N+K)\} > 1-\delta(N+K)\ . \tag{2.8}$$

We have basically proved that, with the probability of one, total number of access points and ad hoc nodes within each Voronoi cell in the constructed tessellation is $O(\log(N+K))$ as $(N+K) \to \infty$. At this point, we also need to prove that there are enough number of access points in each Voronoi cell to be able to route the packets from source nodes to the infrastructure[4] and from access points to the destination nodes without effecting the order of bandwidth allocated to each transmitter.

STEP 4 & 5:

Steps 4 and 5 are again straightforward applications of the Vapnik-Chervonenkis Theorem and lemma 2. But, this time, we let the sequence $\{X_i\}$ be the random positions of *access points*, $\mathcal{F}$ be the set of disks in $\mathcal{R}^2$ with area

$$100 A_R \log(N+K)/(N+K) \ ,$$

and we set

$$\epsilon = \delta = 50 \log(N+K)/(N+K)$$

to obtain the following result as $N \to \infty$.

$$Prob\left\{Number\ of\ access\ points\ in\ any\ Voronoi\ cell \geq \frac{50\log(N+K)}{(1+\alpha)}\right\} > 1-\delta\ . \quad (2.9)$$

Asymptotic lower bound as given in equation 2.9 is also valid for number of ad hoc nodes if we substitute α with $1/\alpha$. These lower bounds and step 2 together imply that both number of ad hoc nodes and access points belonging to the same Voronoi cell are

[4] Actually, one access point per cell is enough for the uplink transmissions, i.e. from source nodes to the access points.

asymptotically in the same order, i.e. $\Theta(\log(N+K))$. Hence, the number of distinct destination points per access point is bounded by $\Theta(1)$ for large (N+K). However, since the source-destination pairs are selected randomly, different source nodes can generate packets for the same destination with a finite probability. This reserve in fact turns out to be a small technicality in the asymptotic results. Suppose that Y_i denotes the position vector of the destination node corresponding to the source node i in our disk domain. Then, $\{Y_i\}$ is a sequence of uniformly distributed i.i.d. random variables. This allows us to use the same $\mathcal{F}$ and $\epsilon = \delta = 100 \log(N+K)/(N+K)$ as in step 3, except for now we have upper-bounded the number of destination points with $O(\log(N+K))$.

Thus, we have completed all the steps required for deriving the lower bound as given in inequality (2.4). Upper and lower bounds in (2.3) and (2.4) state that the throughput capacity for each ad hoc node is $\Theta(W/\log(N))$. This also becomes the first major result of this chapter. In the next section, we modify our connectivity assumption to investigate the full benefits of having the infrastructure network in terms of the throughput capacity.

2.5 Looser Connectivity Conditions and Achievability of Constant Capacity per Node

The strong connectivity condition that is previously imposed on our network model aims at achieving a fully functional ad hoc network without having any infrastructure assistance. Nevertheless, this constraint does not fully capture the benefits of the two-tier architecture. Accordingly, we should relax our connectivity condition as follows: Each ad hoc node should be connected to at least one access point almost surely. This is equivalent to considering the ad hoc network and the infrastructure as a single topology graph and defining the connectivity in accordance with this broader topology. We refer to this specific definition of connectivity as *connectivity in the weak sense* or *weak connectivity*. In this section, we obtain the necessary and sufficient conditions on the transmission range to achieve the weak connectivity. Our results will reveal that even under weak connectivity condition, we cannot have a per node transport capacity of $\Theta(1)$ as it is widely witnessed under different network scenarios [28, 29, 30].

In the simplest form of weak connectivity, there exists at least one access point within the transmission range of any ad hoc node. Hence, given that there are K gateway nodes; X_i denotes the location of node i, which is uniformly distributed on disk domain; each node i has a capture area $A_c^i(X_i)$, where its neighbors can be located; and A_ϵ denotes the disk area with radius $\epsilon = r_T$; the following relations hold:

$$\begin{aligned} &Prob[Node\ i\ connected\ to\ any\ access\ point \mid X_i = x] \\ &\qquad \geq Prob[Node\ i\ has\ an\ adjacent\ access\ point \mid X_i = x] \\ &\qquad \overset{(a)}{=} 1 - \left(1 - \frac{A_c^i(x)}{A_R}\right)^K \\ &\qquad \overset{(b)}{\geq} 1 - \left(1 - \frac{A_\epsilon}{4A_R}\right)^K . \end{aligned} \tag{2.10}$$

Here, step (a) follows directly from the case, where no access point is located in the capture area of node i and step (b) follows from the boundary effect of the disk domain, i.e. at least quarter of a disk centered at the ad hoc node with radius equal to transmission range must be totally covered by the capture area. Integrating both sides of

(2.10) over the disk domain and taking the limit, we find the asymptotic lower bound as:

$$Prob[Node\ i\ connected\ to\ any\ access\ point] \geq 1 - \lim_{K\to\infty}\left(1 - \frac{A_\epsilon}{4A_R}\right)^K . \quad (2.11)$$

We can also obtain an upper bound similar to the right hand side of the expression in (2.11) for the probability of connectivity. Let N denote the number of ad hoc nodes. The event that node i is not connected to an access point includes the event that i is isolated. Hence, the upper bound can be derived as follows.

$$\begin{aligned} &Prob[Node\ i\ disconnected\ from\ access\ points \mid X_i = x] \\ &\geq Prob[Node\ i\ is\ isolated \mid X_i = x] \\ &\overset{(a)}{=} \left(1 - \frac{A_c^i(x)}{A_R}\right)^{N+K-1} \\ &\overset{(b)}{\geq} \left(1 - \frac{A_\epsilon}{A_R}\right)^{c_2 K} . \end{aligned} \quad (2.12)$$

Step (a) is again the result of having no other nodes, including the access points, within the capture area that is uniquely defined by the position of node i on the disk domain and the transmission radius. And step (b) comes from the observation that $A_\epsilon/4 \leq A_c^i(x) \leq A_\epsilon$ in addition to the initial assumption $K = \Theta(N)$. Again integrating both sides of inequality (2.12) over the disk domain, we get rid of the conditional probability;

$$Prob[Node\ i\ disconnected\ from\ access\ points] \geq \left(1 - \frac{A_\epsilon}{A_R}\right)^{c_2 K} .$$

By simple manipulations and taking the limit, we obtain an asymptotic upper bound for weak connectivity;

$$Prob[Node\ i\ connected\ to\ any\ access\ point] \leq 1 - \lim_{K\to\infty}\left(1 - \frac{A_\epsilon}{A_R}\right)^{c_2 K} \quad . \tag{2.13}$$

Next, we introduce a lemma that will assist us to compute the limits in the lower and upper bound expressions given in (2.11) and (2.13) respectively.

Lemma 3. *Let $a(x)$ and $b(x)$ be differentiable functions such that following properties are satisfied: (i) There exists x_1 such that $1/b(x) \neq 0$ for all $x \in (x_1, \infty)$, (ii)* $\lim_{x\to\infty} a(x) = \pm\infty$ *and* $lim_{x\to\infty} b(x) = \pm\infty$. *Then*

$$\lim_{x\to\infty}\left(1 + \frac{1}{a(x)}\right)^{b(x)} = \exp\left[\lim_{x\to\infty}\left(\frac{b(x)^2}{a(x)^2}\frac{\dot{a}(x)}{\dot{b}(x)}\right)\right],$$

provided that the limit on the right hand side exists in $\mathcal{R}^+ = \mathcal{R} \cup \{\infty, -\infty\}$. Above, $\dot{a}(x)$ and $\dot{b}(x)$ represent the derivatives of $a(x)$ and $b(x)$ with respect to x.

Proof. See appendix A.1. □

To apply lemma-3, we need to overcome an obvious technicality. Our upper and lower bound expressions are sequences with non-negative integer indices, but the lemma considers continuous functions. For that reason, we will consider the sequence $A_\epsilon(K)$ as a sampling from a continuous function that captures the desired features of transmission range r_T as a function of number of nodes in the network.

Definition 1. $A_\epsilon(K) = \int_0^\infty A_\epsilon(x)\delta(x-K)dx$ *where $\delta(x-K)$ is the Dirac-Delta function, $A_\epsilon(x)$ is a non-increasing differentiable function of x and* $\lim_{x\to\infty} A_\epsilon = 0$.

From the definition, it is clear that $A_\epsilon(K)$ and $r_T(K)$ are assumed to be monotonically non-increasing sequences with limits 0. The underlying rationale of this assumption is as follows: We are looking for the necessary and sufficient conditions on the sequence $A_\epsilon(K)$, which will ensure the asymptotic probability of connectivity to be arbitrarily large. Yet, we also want to minimize $A_\epsilon(K)$ so that we can pack as many transmission as we can in the same channel maximizing the upper bound. Putting more access points while keeping the r_T same would increase the probability of connectivity as seen from (2.11). Then, we can reduce r_T at a smaller pace than the increase in the number of access points, and at the same time, improve the probability of connectivity. Our next lemma introduces the sufficiency condition for the existence of the limits in the upper and lower bound expressions.

Lemma 4. *Let* $\Gamma_K = [1 - a_1 A_\epsilon(K)]^{a_2 K}$ *and* $\Gamma(x) = [1 - a_1 A_\epsilon(x)]^{a_2 x}$. *If* $\lim_{x \to \infty} \Gamma(x)$ *exists, then*

$$\lim_{K \to \infty} \Gamma_K = \lim_{x \to \infty} \Gamma(x) \; .$$

Proof. From the definition of limit, $\forall \epsilon$, $\exists K_0$ such that if $x > K_0$ then $|\Gamma(x) - \Gamma^*| < \epsilon$. Substituting K_0 with $\lfloor K_0 \rfloor$ and x with K completes the proof. □

Since we have established a relation between Γ_K and $\Gamma(x)$, we are ready to apply lemma-3 to compute the limit of $\Gamma(x)$. To do this, we set $a(x) = -1/a_1 A_\epsilon(x)$ and $b(x) = a_2 x$. Since conditions of lemma-3 are satisfied, we have the following relations given that the limit on right hand side of the equation exists in the set of extended real numbers.

$$\begin{aligned} \lim_{x \to \infty} \left[\Gamma(x) = (1 - a_1 A_\epsilon(x))^{a_2 x}\right] &= \exp\left[\lim_{x \to \infty} \left(\frac{a_2^2 x^2}{1/a_1^2 A_\epsilon^2(x)} \frac{(-1/a_1)\dot{A_\epsilon^{-1}}(x)}{a_2 \dot{x}} \right)\right] \\ &= \exp\left[\lim_{x \to \infty} \left(a_1 a_2 x^2 \dot{A}_\epsilon(x) \right)\right] \; . \qquad (2.14) \end{aligned}$$

Equation (2.14) provides us valuable insights about the necessary and sufficient conditions for connectivity in the weak sense as stated below in theorem-3. But, before the theorem, we first provide some useful properties of $\dot{A}_\epsilon(x)$.

Property 1. $\dot{A}_\epsilon(x) \leq 0$ *for all x.*

Proof. Follows from non-increasing property of $A_\epsilon(x)$. □

Property 2. *If there exists a* X_0 *such that* $\dot{A}_\epsilon$ *is continuous for all* $x \geq X_0$, *then* $\lim_{x\to\infty} \dot{A}_\epsilon(x) = 0$.

Proof. Suppose that limit does not exist or it is not zero. Then there exists $\epsilon_i > 0$ for all $X_i \geq X_0$ such that $|\dot{A}_\epsilon(x)| \geq \epsilon_i$ in a non-zero length interval (x_i, x_{i+1}) where $x_{i+1} > x_i \geq X_i$. Here, non-zero length interval is a consequence of continuity. Since this statement is true for all $X_i = x_{i+1}$, there are infinitely many finite intervals where $\dot{A}_\epsilon(x) \leq -\min_i \epsilon_i$ and in other intervals $\dot{A}_\epsilon(x)$ is at most 0 (using property-1), the integral (hence $A_\epsilon(x)$) diverges to -∞. This contradicts with the definition of $A_\epsilon(x)$. □

Theorem 3. *Given that* $\dot{A}_\epsilon(x)$ *is continuous, the network is asymptotically connected in the weak sense with probability approaching to one if and only if*

$$\lim_{x\to\infty} \left(x^2 \dot{A}_\epsilon(x) \right) = -\infty \; .$$

Proof. If we show that $\lim_{x\to\infty}(x^2\dot{A}_\epsilon(x))$ exists in $\mathcal{R}^+ = \mathcal{R} \cup \{\infty, -\infty\}$, then by using relation (2.14) and lemma-4, we prove the existence of limits for upper and lower bounds. Clearly, these limits are equal to 1 if and only if $\lim_{x\to\infty}(x^2\dot{A}_\epsilon(x)) = -\infty$. Thus, for completing the proof of the lemma, we are bound to demonstrate the existence of $\lim_{x\to\infty}(x^2\dot{A}_\epsilon(x))$ in the set of extended real numbers. We will use the way of contradiction to show it.

Suppose that there is no limit, then for every real number x^*, there exists $x_0 > X_0$ and $\epsilon_0 > 0$ for all X_0 such that,

$$|x_0^2\dot{A}_\epsilon(x_0) - x^*| \geq \epsilon_0 \ .$$

Otherwise, the limit would exist and be equal to x^*. Using our freedom of choosing any x^*, let us set $x^* = 0$. Accordingly,

$$|x_0^2\dot{A}_\epsilon(x_0)| \geq \epsilon_0 \iff |\dot{A}_\epsilon(x_0)| \geq \epsilon_0/x_0^2 \ ,$$

for some $x_0 > X_0$, $\epsilon_0 > 0$ and any X_0. However, we know by property-2 that $\lim_{x\to\infty}\dot{A}_\epsilon(x)$ is 0. Therefore, for all $\epsilon_1 > 0$, there exists an X_1 such that $|\dot{A}_\epsilon(x)| < \epsilon_1$ when $x > X_1$. By setting $\epsilon_1 = \epsilon_0/x_0^2$ and $X_0 = X_1$, we have our contradiction.

Note that, when we replace x^2 in the limit with any non-negative function $\Phi(x)$, the above derivation steps to show the existence of a unique limit point hold by simply substituting x^2 by $\Phi(x)$ and x_0^2 by $\Phi(x_0)$. □

Corollary 1. *Given that $\dot{A}_\epsilon(x)$ is continuous, the network is asymptotically disconnected in the weak sense with the probability approaching to one if and only if*

$$\lim_{x\to\infty}\left(x^2\dot{A}_\epsilon(x)\right) = 0 \ .$$

Corollary 2. *The network is* not *asymptotically connected in the weak sense with the probability approaching to one if*

$$A_\epsilon(K) \leq c_3/K$$

for any positive finite number c_3.

Proof. First, observe that if the network is disconnected in the weak sense for the disk area $A_\epsilon(K)$, it is also disconnected for any other disk area $A_{\epsilon'}(K) \leq A_\epsilon(K)$. Suppose that $A_\epsilon(K) = c_3/K$, then clearly $A_\epsilon(x) = c_3/x$ satisfies the definition-1 as well as the hypothesis of theorem-3. Since $x^2 \dot{A}_\epsilon(x) = -c_3 > -\infty$, theorem-3 states that we do not have weak connectivity with arbitrarily high probability. Thus, it is also true for all $A_{\epsilon'}(K) \leq c_3/K$. □

We can actually prove a more stringent requirement by conditioning connectivity on all nodes, i.e. instead of any node i, all the ad hoc nodes in the network must be asymptotically connected to the infrastructure access points with the probability of one.

Theorem 4. *Let $\mathcal{Y}$ denote the number of nodes that are connected to at least one access point. Then, the expected value of $\mathcal{Y}$, i.e. $E[\mathcal{Y}]$, becomes $\Theta(N)$ for large N if $\lim_{x\to\infty}(x^2 \dot{A}_\epsilon(x)) < 0$. In addition, if any node i is connected to at least one access point with arbitrarily high probability as increasing N (or K), it is also true that all nodes are asymptotically connected to at least one access point with arbitrarily high probability.*

Proof. See appendix A.2. □

We may now state the main result of this section by revisiting the upper bound expression as given in (2.1). The corollary-2 necessitates that $r_T > c_4/\sqrt{\pi N}$, therefore;

$$\lambda(N, K) < \frac{16 A_R W}{c_4^2 \Delta^2} \ , \tag{2.15}$$

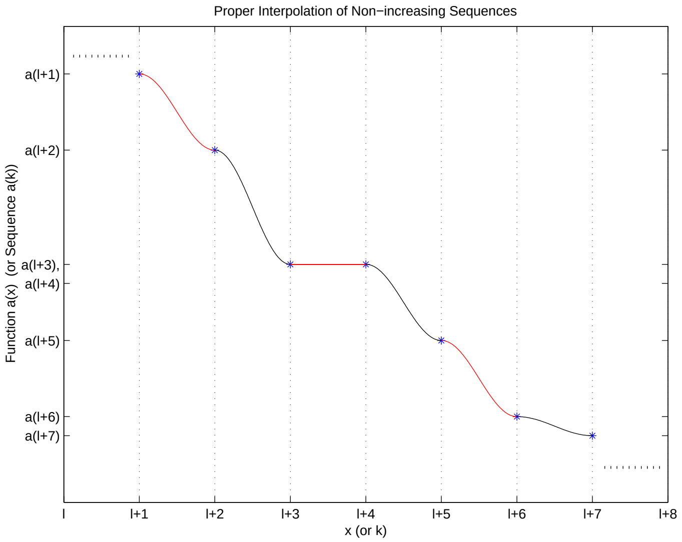


Figure 2.4: Representing non-increasing sequences by differentiable functions with continuous first order derivatives.

for any positive finite number c_4. In other words, per node throughput cannot reach to $\Theta(1)$ as $N \to \infty$. Hence, we have the following theorem.

Theorem 5. *Even under the weak connectivity condition, per node transport capacity of* $\Theta(1)$ cannot *be achieved with the probability of one.*

Now, there remains one subtle point to make the arguments that we made so far rigorous. We have started from non-increasing sequences as an index of the number of nodes, then we have found the necessary and sufficient conditions in terms of any non-increasing differentiable function $A_\epsilon(x)$ with the following conditions: $A_\epsilon(x)$ (i) has samples at non-negative integer points that are equal to the sequence of interest, (ii) has a limit 0, and (iii) has a continuous derivative function.[5] Let us define the set of all such

[5] Note that, since we are mainly interested in the asymptotic behavior, we can modify the statements of definitions, lemmas, and theorems in this section by requiring continuity and monotonicity features only for large K or x values.

functions as $S_\epsilon = \{A_\epsilon(x)\}$. Our results are general in the sense that we can pick any function from S_ϵ and yet use the result given in theorem-3, corollary-1, corollary-2 and theorem-4. The question is whether we can find at least one such function for every sequence of interest. We pictorially demonstrate below that it is indeed the case. Thus, the set S_ϵ represents all possible sequences, in which we are interested.

In figure 2.4, we interpolate any two different valued consecutive sequence points with a cosine function with period 2 in the interval $[0, pi]$, where cosine is a monotonically decreasing function. The amplitude of cosine is shifted in time and amplitude such that it exactly fits into the corresponding interval. If two consecutive points are the same, we interpolate between these two points with a straight horizontal line. Obviously, this piecewise defined function is differentiable. Moreover, the derivative is equal to zero at integer points and behaves as a sine function in between preserving continuity.

The necessary and sufficient conditions as stated in theorem-3 provide us with the mechanisms to check the order of transmission radius and -consequently- the upper bound, above which per node transport capacity cannot be achieved with the probability of one. The question of whether one can find a minimal function on the order of transmission radius (equivalently the maximum upper bound) that conforms with these conditions is not addressed in this chapter. Instead, we show that any upper bound conforming with the necessary and sufficient conditions can indeed be achieved with the probability of one as N goes to ∞.

2.6 Achievability of the Capacity in the case of Weak Connectivity

The design steps to show the achievability of any upper bound that is derived from a transmission area $A_\epsilon(N)$ satisfying the requirements of the weak sense connectivity with the probability of one, are exactly the same as the steps in section-2.4. There are however two nuances: First, the disk areas covered by Voronoi cells in the new tessellation are different; and second, we cannot apply Vapnik-Chervonenkis Theorem for any disk area of interest.

Without loss of generality, let us define $A_\epsilon(N)$ as $g(N)/N$ and suppose that $A_\epsilon(N)$ satisfies the hypothesis of theorem-3. Thus, using equation (2.1) and assuming that $r_T \geq \sqrt{A_\epsilon(N)/\pi}$, the upper bound for per node throughput capacity becomes

$$\lambda(N, K) \leq \frac{16 A_R W}{\Delta^2 g(N)} \ . \tag{2.16}$$

To show that the upper bound given in (2.16) is achievable, we form the tessellation such that $\pi\epsilon^2 = A_R g(N)/N$ and $r_T = 6\epsilon$ (see step 1 in section-2.4). As before, each Voronoi cell is confined between two disks of radii ϵ and 3ϵ respectively. Hence, we need to prove that each Voronoi cell includes $\Theta(g(N))$ ad hoc nodes, access points, and destination points with arbitrarily high probability as $N \to \infty$.

Again, let X_i denote the position of node i in the disk domain. Note that we do not differentiate among node i being a source node, an access point, or a destination node, because $\{X_i\}$ are i.i.d. random variables with uniform distribution across the disk domain in all cases. Define $Y_L \triangleq \sum_{i=1}^{L} I(X_i \in \mathcal{V})$, where $\mathcal{V}$ is a particular Voronoi cell.

Here, L may be either N or K, and we have $\lim_{N\to\infty}(L/N) = \Theta(1)$. Thus, we can compute the mean and variance of Y_L as $\overline{Y} = E[Y] = LP(X_i \in \mathcal{V})$ and $\sigma_Y^2 = Var[Y] = LP(X_i \in \mathcal{V})(1 - P(X_i \in \mathcal{V}))$. Since $P(X_i \in \mathcal{V}) = \Theta(g(N)/N)$, we can use the well-known Chebyshev's inequality [36] as follows:

$$P\left[|Y - L\Theta(\frac{g(N)}{N})| < \gamma\right] \\ \geq 1 - \frac{L\Theta(g(N)/N)(1 - \Theta(g(N)/N))}{\gamma^2} \; .$$

But, here γ can assume any positive value and setting $\gamma = \Theta(g(N))$ simplifies the inequality above further as;

$$P\left[Y = \Theta(g(N))\right] \geq 1 + \frac{1}{\Theta(N)} - \frac{1}{\Theta(g(N))} \; .$$

The results from the previous section require that $g(N)$ cannot be bounded above with a finite value and $g(N)/N$ must be defined for all positive integers N. Therefore, $\lim_{N\to\infty} g(N) = \infty$. In other words, the number of regular ad hoc nodes, access points, and destination nodes in any Voronoi cell is asymptotically in the order of $\Theta(g(N))$ with the probability of one. This means that we can actually achieve any upper bound that conforms to the condition given in theorem-3.

This section concludes our tight results on per node throughput capacity of hybrid networks. Illustrative examples that signify the strength of the results presented in the last two section are given below before we proceed with our final remarks of this chapter.

Example 1. *Let $g(N)$ be $N^{1/p}$ where $p > 1$ is a constant number. Then, $A_\epsilon(N)$ becomes $N^{1/p}/N = N^{1/p-1}$. Trivially choosing $A_\epsilon(x) = x^{1/p-1}$ provides us a continuously* *differentiable and monotonically decreasing function for $x > 0$. Since*

$$\lim_{x\to\infty} x^2 \dot{A}_\epsilon(x) = \lim_{x\to\infty} \left(\frac{1}{p} - 1\right) x^{1/p} = -\infty \ ,$$

$A_\epsilon(x) = x^{1/p-1}$ satisfies the weak connectivity condition with probability one. Thus, the corresponding upper bound $\Theta(1/N^{1/p})$ by selecting $A_\epsilon(N) = \Theta(N^{1/p-1})$ is achievable.

Example 2. *Let $g(N)$ behave as a recursive logarithm function [37] for large N, i.e. $g(N) = \ln^{(m)}(N)$ for $N \geq N_0$ where m, N_0 are positive finite numbers and $\ln^{(m)}(\cdot)$ denotes taking natural logarithm of the argument m times. Then, $A_\epsilon(N)$ becomes $\ln^{(m)}(N)/N$. Simply substituting the discrete variable N with the continuous variable x gives us a continuously differentiable function $A_\epsilon(x)$, which is also monotonically decreasing for sufficiently large x. Since*

$$\lim_{x\to\infty} x^2 \dot{A}_\epsilon(x) = \lim_{x\to\infty} \left[\frac{1}{\prod_{i=1}^{m-1} \ln^{(i)}(x)} - \ln^{(m)}(x)\right] = -\infty,$$

$A_\epsilon(N) = \ln^{(m)}(N)/N$ satisfies the weak connectivity constraint with probability one. Moreover, $\lim_{N\to\infty} \ln^{(m)}(N) = \infty$, therefore, per node throughput $\Theta[1/\ln^{(m)}(N)]$ is feasible with the probability of one for any constant $m > 0$.

2.7 Relaxing the Assumptions

In our asymptotical results, we have made two major assumptions: (i) An infrastructure network that is capable of carrying any load between access point pairs and (ii) an access point population that is in the same order as the population of ad hoc nodes.

The first assumption can be easily satisfied if any pair of access points were inter-connected directly with capacity $C = \alpha\lambda(N, K)$ for some constant $\alpha > 0$.[6] However, this may be a very expensive investment. In the first part of this section, we demonstrate that the bottleneck condition may in fact be observed on the infrastructure network under very reasonable topology constraints.

In the second part of this section, we relax the $K = \Theta(N)$ assumption for each connectivity model and extend our findings to the scenarios where number of access points are $O(N)$.

2.7.1 Capacity Bound for the Constrained Infrastructure

Consider an infrastructure network with the following constraints:

(1) The size of infrastructure network is in the order of K.

(2) Each node in the infrastructure has the maximum capacity of C bits/sec with any of its neighbors.

(3) Each node may have at most M neighbors.

(4) Each access point i generates the same aggregate amount of traffic $\lambda_{ij} = \lambda^*$ in its own cell to any other access point j.

Among these constraints, the first and last conditions are necessary for complying with our hybrid network architecture. The second constraint is valid for any network model. And the third constraint simply limits the topology graph under a more realistic model.

[6] Note that we already proved there were $\Theta(1)$ destination nodes and source nodes per access point in each Voronoi cell.

Given any traffic pattern and route P_{ij} between access point pairs $\{i,j\}$, we can define the total traffic load Λ in bits/sec through the infrastructure networks as:

$$\Lambda = \sum_i \sum_j \sum_l \lambda_{ij} I(l \in P_{ij}) = \sum_i \sum_j \lambda^* h_{ij} \ ,$$

where $I(\cdot)$ is the indicator function and h_{ij} is the hop distance between access points i,j. We can find an upper-bound for Λ for any topology that satisfies constraints (2) and (3) as follows:

$$\Lambda \leq \text{Sum of the capacities of each link} \leq \frac{CKM}{2} \ .$$

We can express Λ in a slightly different way:

$$\Lambda = \sum_i \lambda^*(K-1)\frac{1}{(K-1)}\sum_j h_{ij} = \sum_i (K-1)\lambda^* \bar{h}_i = \lambda^{**} \sum_i \bar{h}_i \ ,$$

where $\lambda^{**} = (K-1)\lambda^*$ should be understood as the total traffic accumulated at access point i from the ad hoc nodes in i's Voronoi cell and $\bar{h}_i$ is the average hop distance from access point i to other access points. Graph theory tells us that smallest $\bar{h}_i$ is $\Theta(\log(K))$ and it is achieved when we have a balanced P-tree rooted at node i [37]. Hence,

$$\lambda^{**} K \Theta(\log(K)) \leq \Lambda \leq \frac{CKM}{2}$$

and

$$\lambda^{**} \leq \Theta\left(\frac{CM}{2\log(K)}\right).$$

In previous sections, we demonstrate that the number of ad hoc nodes per access points is constant. When $K = \Theta(N)$, this means that each access point can serve at most $\Theta(1/\log(N))$ bits/sec for each ad hoc node. In other words, asymptotically, the infrastructure network as constrained in this section is able to match the capacity only under the strong connectivity condition and it becomes the bottleneck under weak connectivity conditions. This is at first a counter-intuitive argument, because it is generally assumed that the wireless network is the bottleneck tier. However the result is due to the following facts:

(i) As the size of wireless network increases indefinitely, we are able to reduce the transmission power and to mitigate the effect of increased network size.

(ii) Any infrastructure network that does not match the spatial re-use factor of the wireless network will eventually become the bottleneck. In the network we considered, we kept C and P as constants. To meet the traffic demand from the wireless nodes, either of them must be an increasing function of the network size.

Having underlined the importance of a proper infrastructure design to avoid bottlenecks in that tier, we continue next by revisiting another major assumption that directly constrains the size of the infrastructure network.

2.7.2 Revisiting the Assumption $K = \Theta(N)$

2.7.2.1 Strong Connectivity Results vs. Access Point Population

The upper bound given by (2.3) is established by finding the minimum r_T that satisfies the strong connectivity condition and the minimum $\bar{h}(N, K)$. Since strong connectivity is independently defined from the population of access points, the only term that explicitly depends on K becomes the average ad hoc hop distance $\bar{h}(N, K)$ between source-destination pairs. The effect of introducing the infrastructure on the throughput capacity amounts to reducing this quantity. When $K = \Theta(N)$, our achievability results suggest that $\bar{h}(N, K) = \Theta(1)$, i.e. capacity is independent of K. In the worst case scenarios, e.g. $K = \Theta(1)$, our capacity results reduce to that of pure ad hoc networks. Although it is quite interesting to investigate what happens in the interval between $\Theta(1)$ and $\Theta(N)$, it is outside the scope of this paper. However, our achievability results can directly allow us to state the following corollary:

Corollary 3. *Throughput capacity under strong connectivity condition is*

$$\lambda(N, K) = \Theta(W/\log(N)) ,$$

provided that $K = \Omega(N/\log(N))$. In other words, strong connectivity dominates the capacity results whenever the access point population satisfies;

$$K \in (\Theta(N/\log(N)), \Theta(N)) .$$

Proof. The steps involved in bounding the population of ad hoc and destination nodes remain same. In step-5 of section-2.4, using relation (2.9), we can show that if $K = \Omega(N/\log(N))$, there exists at least one access point in each Voronoi cell. By allocating half of the bandwidth, i.e. $W/2$, to uplink[7] transmissions and the other half to downlink transmissions, we can effectively carry $\Theta(W/\log(N))$ bits/sec per node in each direction even if there is only one access point available in each cell. □

The corollary also implies that $\bar{h}(N, K) = \Theta(1)$ whenever $K = \Omega(N/\log(N)$.

[7] Uplink refers to the transmissions from ad hoc nodes to access points and downlink refers to the transmissions from access points to ad hoc nodes.

2.7.2.2 Weak Connectivity Results vs. Access Point Population

We can generalize our weak connectivity results by decoupling necessary and sufficient conditions. Suppose $K = h(N)$, where $h(N)$ is a monotonically increasing function of N and $K \leq N$. Following almost the same derivation steps of section-2.5, we obtain:

$$1 - \lim_{K\to\infty}\left(1 - \frac{A_\epsilon}{4A_R(N)}\right)^{h(N)} \leq$$

$$Prob[weak\ connectivity]$$

$$\leq 1 - \lim_{N\to\infty}\left(1 - \frac{A_\epsilon}{A_R(N)}\right)^{\Theta(N)} .$$

Left and right hand sides of the above expression provide respectively the sufficient and the necessary conditions on weak connectivity. By representing our discrete sequences by continuously differentiable functions as before, we have the following corollary:

Corollary 4. *The sufficient condition for weak connectivity is:*

$$\lim_{x\to\infty}\left(\frac{h(x)^2}{\dot{h}(x)}\dot{A}_\epsilon(x)\right) = -\infty \ , \tag{2.17}$$

and the necessary condition is:

$$\lim_{x\to\infty}\left(x^2\dot{A}_\epsilon(x)\right) = -\infty \ . \tag{2.18}$$

The necessity condition is exactly same with the case when $K = \Theta(N)$. This is an evidence of having a very loose necessity condition. The sufficiency result is on the other hand explicitly a function of the dependency between K and N. Note that this sufficiency condition is computed by simply checking the probability that there exists at least one access point within the coverage area of any node. Hence, almost surely any ad hoc node can transmit to or receive from an access point. This also indicates that our sufficiency condition may be too tight and the probabilities of higher order of events must be taken into account for closing the gap between necessary and sufficient conditions.

We can show the achievability of the throughput capacity upper-bound that is subject to condition (2.17) with slight modifications to our previous arguments:

1) We form a tessellation with $A_\epsilon(N) = g(N)/N$. The derivations for upper bounds on the number of ad hoc nodes and destination nodes per Voronoi cell from section-2.6 remain valid, i.e. there are $O(g(N))$ ad hoc and destination nodes per cell.

2) We allow each Voronoi cell to receive and send packets to other cells within transmission range. The number of such cells is bounded from above by a constant. Therefore, an access point can serve a constant number of cells in which there exists no access point in addition to its own cell. The number of nodes to be served remains as $O(g(N))$ and every node is guaranteed to communicate with at least one access point due to the condition (2.17).

3) We divide the available bandwidth into two equal chunks of $W/2$ bits/sec exclusively used for uplink and downlink transmissions.

We finalize our discussion with an example.

Example 3. *Let* $A_\epsilon = N^{\frac{1}{p}-1}$ *with* $p > 1$ *and* $K = h(N) = N^\alpha$*. Substitute* N *with* *continuous variable x. Since*

$$\begin{aligned}\lim_{x\to\infty}\left(\frac{h(x)^2}{\dot{h}(x)}\dot{A}_\epsilon(x)\right) &= \lim_{x\to\infty}\frac{1}{\alpha}\left(\frac{1}{p}-1\right)x^{\alpha+\frac{1}{p}-1} \\ &= \begin{cases} -\infty & ;\ \alpha+\frac{1}{p}>1 \\ c>-\infty & ;\ \alpha+\frac{1}{p}\leq 1\end{cases},\end{aligned}$$

the throughput capacity of $\Theta(W/N^{\frac{1}{p}})$ is achievable when $K=N^\alpha$ such that $\alpha+\frac{1}{p}>1$. In the special case where $p=2$ and $\alpha=\frac{1}{2}+\epsilon$, where ϵ is arbitrarily small positive real constant, we achieve $\Theta(W/\sqrt{N})$ by using $K=N^{\frac{1}{2}+\epsilon}$ access points. This throughput result is equivalent to the asymptotic capacity of arbitrary ad hoc networks.

2.8 Summary

In this chapter, we addressed the benefits of using a hybrid network architecture over pure ad hoc wireless networks with no infrastructure support in terms of per node throughput capacity. We showed that adding an infrastructure, which provides access to the ad hoc users at random locations, improves the per node throughput significantly over the infrastructureless operation. Such a hybrid network model is adequate especially when the access points of the infrastructure network are not placed on regular grid points. Supporting examples can be given from a wide span of scenarios, e.g. sensor networks formed by scattering the sensors, some of which have long-range and high bandwidth radio trancievers, over a terrain, cellular/WLAN networks with wireless/mobile routers, ad hoc networks with airborne communication node (ACN) support, etc.

We have started with a strict connectivity constraint, under which ad hoc tier must preserve the connectivity with arbitrarily high probability for stand-alone operations. The asymptotic capacity figures are derived under this regimen. Our results reveal that $\Theta(\sqrt{N/\log(N)})$ folds better performance than what the pure ad hoc operations may obtain, despite of the randomness imposed on the locations of the access points. The gain in performance is mainly due to the fact that the mean number of hops from source to destination in the ad hoc tier is effectively reduced to a constant factor as opposed to the case of pure ad hoc networks, where the mean number of hops increases as a function of N.

In the second part of the chapter, we relaxed the connectivity constraint to fully utilize the infrastructure network. Under this weak connectivity constraint, the combined network topology graph is required to be connected. We devised an analytical tool to find the necessary and sufficient conditions on the radio transmission range, which effectively determines the upper bound on the per node throughput capacity. As a consequence of the necessary conditions, even under the weak connectivity, per node throughput asymptotically goes to zero in contrast to the constant rates obtained under different problem constructions reported in the literature. Nonetheless, the rate of convergence to zero can be made remarkably small at the expense of increased confidence interval for weak connectivity. Although we could not provide a minimal function on the transmission radius, which effectively leads us to the maximum upper bound on capacity without compromising the weak connectivity condition, we proved that this maximum upper bound can in fact be achieved with the probability of one.

In the last section, we also relaxed our assumptions on the network model in terms of unconstrained infrastructure tier and access point population. In the first part, we primarily showed that under very general topology and capacity constraints, a wired network can in fact be the bottleneck part of the overall architecture. In the second part, we generalized our capacity derivations to the scenarios where we have unbounded number of ad hoc nodes per access point. We showed that in fact the strong connectivity condition limits the capacity figures when $K = \Omega(N/\log{(N)})$ and throughput capacity remains same for such K values. For weak connectivity, our upper and lower bounds turn out to be loose unlike the case when $K = \Theta(N)$. Despite of this, they can still provide useful necessity and sufficiency conditions to examine a large class of transmission radii and access point populations.

Chapter 3

Hybrid Networks as Finite Arbitrary Graphs

The previous chapter has focused on a random geometric graph model of two-tier hybrid networks and we presented asymptotical capacity figures under a probabilistic measure for this model. Although such an analysis provides us an understanding of how network throughput scales for large network size, it does not capture more practical scenarios. In reality, we usually have an overlaid network with the following properties:

- An arbitrary network topology with finite number of nodes (e.g. $\sim 10 - 100$).
- A finite number of feasible access points that can provide access to the infrastructure for each logically or geographically defined domain.
- A physical layer that is capable of adapting parameters such as transmit power, modulation level, coding rates, antenna beam coefficients and spreading codes to achieve a certain level of link quality measured in terms of signal to interference and noise (SINR) ratio or bit error rate.
- An interference model that is defined at the signal level.
- A well-defined medium access scheme that resolves time and space conflicts on

the same channel.

- A finite number of sessions that are simultaneously active between deterministic source-destination pairs. The destination can be another wireless host in the same wireless network domain or a wired node connected to the global data network. Each session may have different rate and packet error rate requirements.

- The nodes that are part of the network service provider -such as domain wireless routers and access points- and do not generate payload data sessions, but merely assist in relaying packets to the destination.

- Co-existing symbiotic systems, each of which has to minimize its own total radio signal power emanation.

- Energy-limited wireless users.

When all these factors are taken into account, instead of trying to find and attain the network capacity, a more sensible approach would be to try satisfying the quality of service (QoS) demands of the individual sessions at minimal total power emanation. Minimizing the total power emanation reduces the impairments on other logically separated domains or symbiotic systems. It also improves the power savings of energy limited nodes.

QoS is interpreted quite differently depending on the particular communication layer. At the lowest level, i.e. physical layer, QoS is synonymous to an acceptable *bit error rate (BER)* or *signal to interference and noise ratio (SINR)*, whereas at the MAC layer or higher layers, QoS is usually expressed in terms of minimum rate or maximum delay guarantees. For the multi-hop communications, network layer QoS pertains to an end-to-end provisioning of the guaranteed QoS for each session. In accordance with these different interpretations at different layers, it is natural to use a QoS policy that is explicitly based on both minimum short-term rate requirements and maximum tolerable BERs of the sessions. Such a QoS policy also helps classifying the applications as high bandwidth or low bandwidth and as error prone or error resilient.

Although QoS definitions differ among layers, its value can be determined only after all the layers finalize their decisions. Therefore, a QoS guarantee in real terms can be satisfied if the decisions at each layer converge to the desired QoS value or else a joint control mechanism that crosses multiple layers is enforced. In the first section of this chapter, we summarize the interactions among the lowest three layers in terms of how they affect each other's decisions, energy consumption, and QoS values.

3.1 Cross-layer Interaction

Wireless transmissions mainly suffer from channel impairments and other user interference operating in the same frequency band. Multi-hop wireless operation merely exacerbates the existing conditions. Unless a coordination spanning to multiple layers and multiple hops exists, either the session QoS requirements are not satisfied or they are probably satisfied at a significantly higher energy consumption than the necessary. Once the set of sessions with their source-destination pairs and QoS requirements are given, three layers together impact the contention for network resources: physical layer, medium access control (MAC) layer, and routing layer. For a cross-layer design that satisfactorily enhances the network performance, it is essential to dissect the interactions among these layers.

Physical layer with its key parameters- such as transmit power, modulation, coding rate, antenna beam coefficients- has a direct impact on multiple access of nodes in wireless channels through affecting the interference at receivers and the susceptibility to it. Local adaptation of these parameters to achieve a target BER restraints both routing and MAC decisions by altering the directed topology graph, feasible transmission schedules, and payload transmission rates. Physical layer features -such as transceiver complexity, power required to drive the RF modules, and the transmit power- accumulatively govern the energy expenditure of transmitters, receivers, and idle nodes.

MAC layer is responsible for scheduling the transmissions and allocating the wireless channels. While the concurrent transmissions create mutual interference, the time evolution of the scheduled transmissions ultimately determines the bandwidth allocated to each transmitter and the packet delays. The interference imposed by simultaneous transmissions naturally affects the performance of the physical layer in terms of successfully separating the desired signals from the rest. On the other hand, as a result of transmission schedules, high packet delays and/or low bandwidth can occur, forcing the routing layer to change its route decisions. MAC layer influences the energy expenditure in two ways: (i) It mainly controls the interference level at any time instance that may lead to transmit power adaptation in the physical layer. (ii) Depending on the transmission schedules, nodes may switch to a power-saving mode, turning off all or some of their RF components.

Routing layer selects the wireless links that will eventually carry the data packets. Different routing decisions alter the set of links to be scheduled, and thereby influence the performance of MAC layer. For instance, if the routing protocol chooses flow paths that are closer to each other among the alternatives, the subsequently higher interference and contention levels in the network make it harder for MAC to resolve the transmission conflicts. Similarly, higher interference levels force the adaptation of physical layer parameters to achieve the target BER. However, as the number of independent sessions with distinct source-destination pairs increases, the routing criterion is expected to play a less important role in contention resolution as compared to the physical layer adaptations and MAC decisions. [1] When QoS requirements are ignored and link costs that accurately quantify the energy consumption can be assigned, routing layer becomes the sole determinant of energy consumption. These link costs, however, depend on the transmit power, which is a function of decisions in all three layers. Therefore, the layer interactions necessitate iterative approaches to find the most energy efficient communication scenario.

In our two-tier network architecture, we have an additional degree of freedom: the choice of access point to reach the infrastructure. This is conceptually the same as using different routing paths when the overall network topology is considered. But, the complexity of the decision is much lower while the impact on the wireless tier is more significant. The complexity is low because of the number of access points is generally

[1] One should exclusively consider the cases where each session or the routing layer has a degree of freedom in selecting the destination point among a set of functionally equivalent nodes.

much smaller than the number of available paths. The impact is higher because as long as the destination is same, the paths have to collapse at the same point limiting the degree of separation from other session paths, whereas availability of multiple end points do not have such restriction. This is especially important for the nodes that are approximately located at the same distance from the access points.

Although the ideal network design for energy-efficient and QoS-based communication requires jointly computing the session paths, transmission schedules and physical layer parameters, we assume that a set of routing paths between source-destination paths are given to us a priori. Therefore, our focus will be on joint scheduling, power control and access point assignment problems.

3.2 Related Works

Power control has been the focus of single-hop multi-user wireless networks for more than a decade [38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48]. The popularity of the topic stems from the facts that it can be exploited in suppressing multi-user interference, increasing system user or throughput capacity, and reducing the transmission power hence extending the battery life of the wireless devices. Later on, power control has also been adopted as an efficient protocol design technique for ad hoc wireless networks in different layers as joint or isolated problems [49, 50, 51, 52, 53, 54, 55, 56, 57]. Among these highly diverse works, it is essential to dwell upon two recent studies, [49] and [50], in order to elucidate our own contribution in this chapter.

In [49], Elbatt and Ephremides investigate the problem of scheduling maximum number of links in the same time slot. In other words, authors try to maximize the per hop throughput of the network. They adapt the transmit powers to their minimum required levels such that all transmissions achieve a target SINR threshold. They show that this particular system model is actually equivalent to uplink power control in cellular networks and the iterative algorithms developed for cellular networks can be employed in ad hoc wireless networks. In the case where the set of links that have buffered packets cannot be scheduled in the same time slot, these solutions do not converge and authors suggest to remove one link at a time until a feasible set of links is achieved. However, the criterion for removing the link is not precisely addressed; especially in the case of varying target SINR thresholds for each link. Also, the system model does not cover a multi-hop wireless environment.

A closer approach to our own is followed by Cruz and Santhanam in [50], where authors provide long term end-to-end rate guarantees to a set of sessions at the minimum possible long term average of the total transmit powers. Their main assumption is that the system operates at significantly low SINR values and that the link rates can be approximated as linearly dependent on SINR. Hence, the transmit power is not used for giving a quality of service guarantee in bit error rate (BER) but rather directly used as a throughput guarantee constraint. Instead of solving the relatively difficult problem of minimizing the long term average transmit power sum with the constraints on the power vector and on the long term session rates, they define and solve a dual problem that does not have a *duality gap* with the primary problem [58]. Their results reveal that all the links scheduled in a particular time slot must transmit at the maximum allowed power

P_{max} rather than in more number of slots at a lower power level. The solution method to determine the set of links that must be activated simultaneously as well as the existence of schedules to achieve the rate requirements are established in the paper. Under certain continuity conditions on the optimum dual objective function, authors also extend cross-layering to the routing layer, where each small increment in session rates is routed dynamically abiding by the path costs as determined by the rate of change in dual objective function. Hence, the optimal joint routing, scheduling, and power control policy is obtained.

Our system model differs from that of Cruz and Santhanam in several respects. First of all, we want to satisfy the rate requirements of the sessions not only in the long term but also in the short term within a well-defined frame duration. This prevents the sessions with low jitter or bounded delay requirement suffering from the ambiguity of the *long term* guarantees. Secondly, the end-to-end rate constraint used in [50] is actually the end-to-end throughput constraint, i.e. the number of bits that are successfully reached to the destination. We instead decouple the end-to-end throughput constraints into the transmission rate and the BER constraints, which better differentiate the applications and which are more amenable to actual system implementation. By this way, we also avoid the artificial assumptions such as approximating the rate as a linear function of SINR values.

3.3 System Model

We consider a wireless network of N wireless ad hoc nodes and K access points. Each node is capable of transmitting at a power value less than or equal to P_{max}. A *directed link* exists between nodes i and j if the signal to noise ratio (SNR) at receiver j, when i transmits at this maximum power, is above a threshold γ_{ij}, i.e. $G_{ij}P_{max}/\sigma_j^2 \geq \gamma_{ij}$, where G_{ij} represents the path gain from i to j and σ_j^2 is the ambient noise at receiver j. The infrastructure network topology is constrained such that each node is assumed to have a symmetric link with at least one access point. Furthermore, we have S sessions and each session i is characterized by:

- A $\{$source,destination$\}$ pair.
- A set of paths between $\{$source,destination$\}$ pairs that are represented by a sequence of directed links.
- A minimum short-term end-to-end rate requirement in bits/sec.
- Maximum BER requirement for each directed link along the specific session path.

The end-to-end rate requirement for a session dictates that the designated session rate must be supported across *all* links that constitute the session path. BER requirements are derived as a link budget estimation using the information on the total error tolerance of the session and its path length. A simple budgeting can be done by assuming that: (1) Each hop maintains the same packet error rate $P_\epsilon^{packet,h} = \Psi(BER)$, where $\Psi(\cdot)$ is a monotonically increasing function of its argument with inverse function $\Psi^{-1}(\cdot)$ and (2) the error probabilities of subsequent transmissions of the same packet over different hops are independent. Hence, end-to-end packet error rate P_ϵ can be written as:

$$P_\epsilon = 1 - (1 - P_\epsilon^{packet,h})^h ,$$

where h is the path length in number of hops. Equivalently, BER can be expressed as:

$$BER = \Psi^{-1}\left[1 - (1 - P_\epsilon)^{\frac{1}{h}}\right] . \tag{3.1}$$

The rest of the section explains the specific details of how session requirements are satisfied.

3.3.1 Channel Model

The data packets are transmitted over the same wireless channel, which explicitly refers to the same frequency band. To prevent self-interference, half-duplex operation is enforced, i.e. a node cannot transmit and receive at the same time. We also limit ourselves only to point-to-point transmissions and no node is permitted to send multiple packets (for the same receiver or not) at the same time. The payload rate R of link l over the data channel is given by

$$R(l) = \frac{b_{sym}^l \times R_c^l}{T_{sym}^l} ,$$

where b_{sym}^l is the number of bits per symbol, R_c^l is the coding rate, and T_{sym}^l is the symbol duration for the transmissions over l. Time domain is divided into slots of length T_{slot} and time slots are further grouped into frames of L slots. We do not have control over the physical layer parameters b_{sym}^l, R_c^l, and T_{sym}^l, but we assume that they can be altered only before the start of each frame and that they are kept fixed throughout the frame. Hence, for link l, each slot has a constant payload rate, i.e.

$$r_l = \frac{b^l_{sym} \times R^l_c}{L \times T^l_{sym}} \ .$$

The scheduling is performed per frame basis and each link is assigned to a number of slots in a given frame. More precisely, the *short-term* rate requirement r_i of each session i, which traverses directed link l, necessitates allocating

$$k^l_i = \left\lceil \frac{r_i}{r_l} \right\rceil$$

time slots for link l. Here, $\lceil \cdot \rceil$ stands for the ceiling operation. Note that, in reality, we assign the time slots to the transmitter of a link and different links may have the same transmitter. As it will be clear later on, transmitters can utilize the same time slot assigned to them for different sessions only if the sessions have the same BER constraint and they traverse the same directed link. Therefore, the actual number of time slots k^l assigned to a directed link l can be bounded as;

$$\left\lceil \frac{1}{r_l} \sum_i I(l \in \mathcal{P}_i) r_i \right\rceil \leq k^l \leq \sum_i I(l \in \mathcal{P}_i) \left\lceil \frac{r_i}{r_l} \right\rceil \ ,$$

where $\mathcal{P}_i$ represents the flow path of session i and $I(\cdot)$ is the indicator function that is equal to one if its argument is true and zero otherwise. k^l satisfies the left hand side of the above expression when all sessions traversing link l have the same BER requirement and are able to be multiplexed together onto the same slot. If no multiplexing is possible, the upper-bound on the right hand side holds with equality. Obviously, the lower and upper bounds become the same when r_i's are integer multiples of r_l. Here on, without

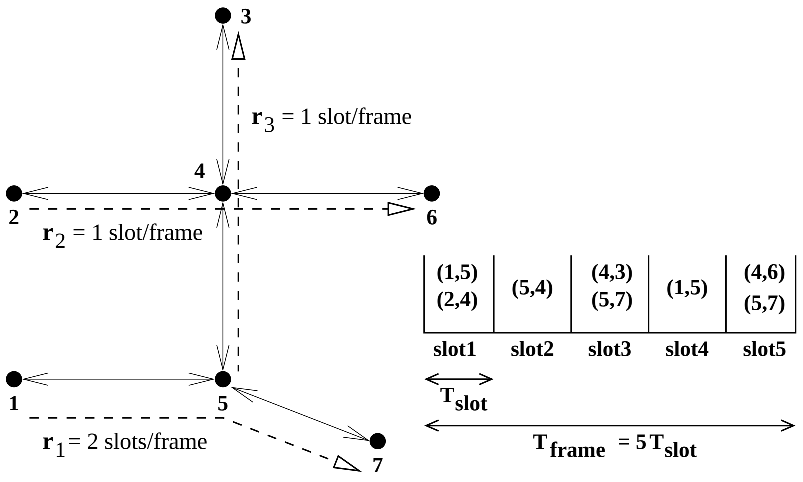


Figure 3.1: Sample topology and scheduling for concurrent multi-hop sessions.

loss of generality, we restrict our attention to the session rates that are integer multiples of r_l.

Let us examine our system model as described so far on Fig.3.1. In the figure, bidirectional arrows show the existence of directed links between node pairs they connect. The frame length is set to 5 slots. There are three sessions initiated at nodes 1, 2, and 5 with flow paths depicted by dashed directional arrows. Session 1 has a bandwidth requirement of 2 slots per frame, whereas sessions 2 and 3 both require 1 slot per frame. Thus, total end to end bandwidth requirement becomes 8 slots per frame. Since the total figure is above the frame length, different links have to be activated at the same time. Also, the BER requirements at each receiver must be satisfied in all time slots. A sample link scheduling is given in the figure. Due to the bandwidth requirements or overlapping flows, the same link can be activated more than once during a frame period. For instance, ordered vertex pairs (1,5) and (5,7) must both be scheduled twice in the sample scenario.

3.3.2 SINR threshold and Feasibility of Concurrent Transmissions

We now elaborate on how the BER constraints of the concurrent transmissions can be satisfied using the proper schedules and transmission powers. For this purpose, we look at the relation among the modulation level, coding rate, BER and SINR.

Our main assumption on BER is that it must be a one-to-one monotonically decreasing function of SINR around the receiver's operating point. Accordingly, a maximum tolerable BER can be mapped onto a minimum SINR threshold for a successful reception. In general, transceiver pairs may support multiple modulation levels (e.g. M-QAM with $M \in \{1, 2, \ldots, M_0\}$) and code rates (e.g. R_c = 1/2, 3/4, 7/8, 1). In the presence of time-varying link quality, the objective of modulation and coding rate adaptations is to increase transmission rate and to maintain an acceptable BER at the receivers. Lower modulation levels and coding rate can sustain more interference or equivalently assist in lowering average transmitted signal power at the same interference level.

For instance, when M-QAM modulation is used for the transmissions over link l, e.g. $b^l_{sym} = \log_2 M$, the BER is approximated as BER $\approx 0.2 \exp[-1.5(\text{SINR})/M - 1]$ [59].[2] For a maximum acceptable BER of ϵ, the SINR should satisfy

$$SINR \geq \frac{-\ln(5\epsilon)}{1.5}(M-1)\ . \tag{3.2}$$

Thus, we map each modulation level b_{sym} and maximum acceptable BER to the SINR threshold γ_l, which is equal to the right-hand side of (3.2).[3] Clearly, decreasing b_{sym} or

[2] This is under the assumption that interference can be approximated as a Gaussian random variable.

[3] In general, the SINR thresholds for each transmission (even over the same link) differ from each other;

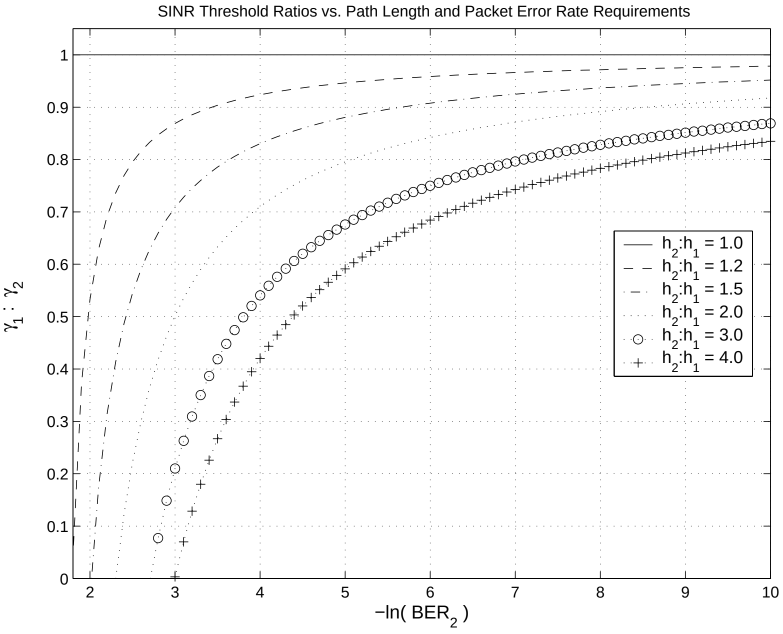


Figure 3.2: SINR threshold over different paths for the same session depends on the specific coding, modulation level and path length. Above we plot the SINR threshold ratios against $-\ln(BER_2)$ when different paths with hop distance ratios h_2/h_1 are utilized with fixed modulation levels and uncoded transmissions under independence assumption. BER_2 is the per link bit error rate requirement that corresponds to the SINR threshold γ_2 under the same end-to-end packet error rate requirement.

R_c also reduces the SINR threshold. From equations (3.1) and (3.2), we can further relate the end-to-end packet error rate (i.e. P_ϵ) requirements of each session with the SINR requirements along a particular path (see figure 3.2). On the other hand, the left-hand side of (3.2) is determined by channel gains, noise power, and the transmit powers of the links assigned to the same time slot. We allow the adaptation of transmit powers between consecutive time slots. Since we have assumed that the coding rate and modulation level are kept fixed throughout the frame, transmit powers and slot assignments are the only controls we have to satisfy the BER constraints.

because either different modulation or coding schemes are used for different links of sessions or each session is characterized by its own BER requirement.

Suppose that $\mathcal{C}(n)$ denotes the set of links that are assigned to slot n; $T(l)$ and $R(l)$ are the transmitter and receiver end points of directed link l; P_l is the transmission power at node $T(l)$; and $s(l,n)$ is the session for which the transmissions over link l in slot n are reserved. Then, for each link $l \in \mathcal{C}(n)$, at the given modulation level and coding rate, BER requirements of $s(l,n)$'s are mapped onto the following set of constraints:

$$\frac{G_{T(l)R(l)}P_l}{\sum\limits_{\substack{j \neq l \\ j \in \mathcal{C}(n)}} G_{T(j)R(l)}P_j + \sigma^2_{R(l)}} \geq \gamma_l \ ; \ \forall l \in \mathcal{C}(n) \ . \tag{3.3}$$

Constraints in (3.3) can be put into matrix form by defining $|\mathcal{C}(n)|$ by $|\mathcal{C}(n)|$ matrix $\tilde{\mathbf{G}}$ and the column vector β with entries:

$$\tilde{G}_{ij} = \frac{\gamma_j}{1+\gamma_j}\frac{G_{T(i)R(j)}}{G_{T(j)R(j)}} \ \ ; \ \ \beta_i = \frac{\gamma_i}{1+\gamma_i}\frac{\sigma^2_{R(i)}}{G_{T(i)R(i)}} \ . \tag{3.4}$$

Then, we obtain:

$$\boldsymbol{P} \geq \tilde{\boldsymbol{G}}\boldsymbol{P} + \boldsymbol{\beta} \ . \tag{3.5}$$

Here, $\mathbf{P}$ is simply the transmit power vector for the links assigned to slot n. $\mathcal{C}(n)$ is a *feasible* assignment for slot n if (3.5) is satisfied for a non-negative and finite $\mathbf{P}$.

Matrix $\tilde{\mathbf{G}}$ is non-negative and irreducible. From Perron-Frobenius theorem, $\tilde{\mathbf{G}}$ has exactly one positive real eigenvalue ρ with $\rho = \max\{|\lambda_i|\}_{i=1}^{M}$, where $\{\lambda_i\}_{i=1}^{M}$ are the eigenvalues of $\tilde{\mathbf{G}}$. ρ is called the Perron-Frobenius eigenvalue of $\tilde{\mathbf{G}}$. It is well-established that (3.5) is satisfied for a non-negative and finite $\mathbf{P}$ if and only if $\rho < 1$ [46]. Hence, the feasibility of $\mathcal{C}(n)$ is solely determined by the maximum eigenvalue of $\tilde{\mathbf{G}}$, which is contingent upon the channel gains and the sessions' BER requirements.

It is important to note that, in our model, channel gains of different links remain constant within a time frame. Thus, our approach applies primarily to quasi-stationary or fully stationary wireless networks, when the link gain G_{ij} of each link (i, j) captures mainly path loss and shadowing effects.

Next, we present the notion of *virtual links* to simplify our system model.

3.3.3 Notion of Virtual Links

Virtual links are defined to avoid dealing with the bandwidth and BER requirements of the sessions explicitly. Let's denote the index set of active links[4] with $\Lambda^a = \{1, 2, \dots, E\}$. As the same link can be scheduled more than once (in different slots), we index each instance of such links separately and denote them as virtual links, because they physically constitute the same link. Thus, we have a populated index set $\Lambda^v = \{1, 2, \dots, M\}$ for virtual links where $M = \sum_{i=1}^{S} h_i(r_i/r_l)$ and h_i is the number of hops that ith session traverses. Each virtual link i is further labelled by a two-tuple $\{s(i), h(i)\}$, where $s(i)$ is the session number the virtual link is allocated to and $h(i)$ is the hop distance of the physical link from the source node of ith session. The SINR constraints of each virtual link i is derived from the end-to-end error requirements of session $s(i)$, session path distance and parameters of the underlying physical link. We continue to use $T(i)$ and $R(i)$ notation to denote the actual transmitting and receiving end points of the virtual link i. Before defining our problem over these virtual links, we need to elaborate on one more subtle point.

Our channel model restricts us to half duplex operation and point-to-point communication with one packet transmission at a time. The former condition is violated

[4] This is the set of links which carry payload traffic as a result of routing decisions.

if two virtual links i and j that are scheduled in the same slot have the property of $T(i) = R(j)$ and the latter is violated if $T(i) = T(j)$. These properties suggest that the set of links scheduled for the same time slot must be a *matching set* in the corresponding topology graph. Nonetheless, we can simply absorb the *matching set* constraint into the SINR constraints by setting $G_{T(i)T(i)} = \infty$ and letting the γ_i's to be high enough. In other words, when node i is scheduled to receive and to transmit at the same time, the SINR at node i is driven to zero, violating its positive SINR requirement as a receiver. In a similar way, if two virtual links with the same transmitter are simultaneously scheduled, they will be strong interferers for each other, leading to unsatisfied SINR constraints

Until now, we have not referred to any specifics of a hybrid network scenario and our system model fully capture a flat ad hoc network topology with no infrastructure support. Nevertheless, the system model does not require a modification in the problem formulation even in the case of hybrid network scenarios.

3.3.4 Hybrid Network Communication Scenarios

Our formulation is completely based on sessions, their requirements, and a set of paths for each session. This provides a very general framework that captures various communication paradigms well for both flat ad hoc and hybrid wireless networks. In two-tier network topologies, we typically have the following communication paradigms:

(i) Wireless hosts communicate with the third parties that are connected to global network in a different (wired or wireless) domain. The goal of wireless hosts in such a scenario becomes being connected to an access point that can satisfy their session requirements. Then, the set of paths for each session corresponds to the routes that traverse different feasible access points. As other wireless hosts or sessions join or the link gain matrix alter, switching over a different path may result in a change of access point. Therefore, our system model also generalize the *handoff* problem of classical cellular or WLAN systems to multi-hop wireless infrastructure systems.

(ii) Wireless hosts communicate with the other wireless nodes in the same domain. This task can be achieved using the paths that completely reside in the ad hoc tier or using the paths that partially overlap with the infrastructure network, where the ingress and egress points correspond to the feasible access points. This scenario can be transformed into the first scenario by simple considering the wireless destination node as another feasible access point for the specific session. The solution provides the answer to the question of *to use or not to use* the infrastructure.

(iii) We can also have a mixed scenario that has wireless hosts with any of the first two communication paradigm. However, as in the second case, it is trivial to transform the scenario into the first case.

Hence, without loss of generality, we can focus only on the first paradigm and formally state our problem in that context in the following sections.

3.4 Joint Power Allocation, Schedule and Path Assignment Problem in two tier Hybrid Networks

3.4.1 Formal Problem Statement

We want to minimize the total transmit power as summed over all time slots and links while satisfying the minimum rate and SINR constraints of the sessions. Rate requirements of individual sessions are satisfied if and only if we can place all of the virtual links into a single frame in a specific order. This part constitutes the scheduling component of the problem. Suppose that there are π_i different paths available for session i. Then, we have to choose a path assignment from the set $\Pi = \pi_1 \times \ldots \times \pi_S$. Given that there are B access points to reach the infrastructure tier, a particular path assignment $m \in \{1, 2, \ldots, |\Pi|\}$ results in an access point assignment $\chi(m) \in \{1, 2, \ldots, B\}^S$ and a set of virtual links $\Lambda^v(m)$. This part corresponds to the path assignment component of the problem. And the power allocation component tries to satisfy the SINR requirements while minimizing the total transmit power at each slot by adjusting the transmit powers at each scheduled virtual link. For the path assignment m, joint scheduling and power allocation problem $P1$ is expressed as:

$$\Xi(m) = \min_{\mathcal{A}(m), \underline{P}(m)} \sum_{i \in \Lambda^v(m)} P_i \ , \tag{3.6}$$

subject to the constraints:

$$\frac{G_{T(i)R(i)}P_i}{\sum\limits_{\substack{j\neq i\\ c(j)=c(i)}} G_{T(j)R(i)}P_j + \sigma^2_{R(i)}} \geq \gamma_i \quad ; \quad \forall i \in \Lambda^v(m), \tag{3.7}$$

$$c(i) \in \{1, 2, \ldots, L\} \quad ; \quad \forall i \in \Lambda^v(m), \tag{3.8}$$

$$0 \leq P_i \leq P_{max} \quad ; \quad \forall i \in \Lambda^v(m), \tag{3.9}$$

$$\sum_{k=1}^{c} N_k(s(i), h(i)) < \sum_{k=1}^{c} N_k(s(i), h(i)-1) \quad ; \quad \forall i \in \Lambda^v(m)\ ,\ c \in \{1, \ldots, L\}, \tag{3.10}$$

where P_i is the transmit power of node $T(i)$, $c(i)$ is the time slot virtual link i is assigned to, and $N_x(y, z)$ is the number of virtual links that are assigned to slot x and that are labelled with $\{y, z\}$. The last constraint simply states that *any link that is closer to the session source must be scheduled more than the further ones along the session path*. It guarantees that whenever a virtual link is scheduled to transmit, it indeed has a packet to transmit.[5] On the other hand, constraints (3.7), (3.8), and (3.9) correspond to the SINR, frame length, and power requirements respectively. Together, they define the constraint set

$$\Omega(m) = \{\mathcal{A}(m), \underline{P}(m) : 0 \geq \underline{P}(m) \geq P_{max} \; and \; \underline{P}(m) \geq \Gamma_m \mathcal{H}_m \underline{P}(m) + \underline{\beta}(m)\} \ .$$

Here, $\mathcal{A}(m) : \Lambda^v(m) \to \{1, \ldots, L\}^{|\Lambda^v(m)|}$ is the time slot assignment of virtual links; $\underline{P}(m)$ is the $|\Lambda^v(m)| \times 1$ column vector with ith entry P_i; Γ_m is the $|\Lambda^v(m)| \times |\Lambda^v(m)|$ diagonal matrix with diagonal entries $\Gamma_{i,i} = \gamma_i$; $\mathcal{H}_m$ is the $|\Lambda^v(m)| \times |\Lambda^v(m)|$

[5] This assumes that no packet loss occurs due to channel errors and buffer overflows. The first one dominates the packet losses and it has been taken into account within the error rate guarantee of each session.

interference matrix with entries

$$H^m_{i,j} = \begin{cases} \delta_{ij}\frac{G_{T(i)R(j)}}{G_{T(i)R(i)}} & \text{for } i \neq j \\ 0 & \text{for } i = j \end{cases} ;$$

$\underline{\beta}(m)$ is the $|\Lambda^v(m)| \times 1$ column vector with ith entry $\gamma_i\sigma^2_{R(i)}/G_{T(i)R(i)}$; and δ_{ij} is the assignment function that is equal to one if $c(i) = c(j)$, otherwise to zero. Whenever we have a pair $(\mathcal{A}(m), \underline{P}(m)) \in \Omega(m)$, we will refer to them as *jointly feasible* schedule and power allocation. Among all such pairs, we are interested in the ones that minimize (3.6), which we will call *jointly optimal* schedule and power allocations.

Bearing to the definition of $P1$, joint path, schedule and power assignment problem $P2$ is defined below as:

$$\min_m \Xi(m) . \tag{3.11}$$

Given the assignment instances m and $\mathcal{A}(m)$, our problem reduces to classical power control problem in cellular networks and we may check if there exists a feasible solution by investigating Perron-Frobenius eigenvalue of $\Gamma_m\mathcal{H}_m$ [46]. Moreover, we can find the optimum power allocation at each slot centrally or iteratively. In fact, the optimum power allocation is *Pareto optimal*, i.e. all the links transmit at their minimum feasible power, and the constraint (3.7) is satisfied with equality [43]. However, finding the jointly optimum transmit power and time slot allocation is not straight-forward extension to the continuous transmission scheme as in the cellular voice services [48]. Since our constraint set does not satisfy the necessary monotonicity feature of the standard function [6], the existing iterative solutions cannot solve our problem. Besides, the

[6] See [45] for details of the standard function and its convergence proof.

constraint set is not a convex set in general and we cannot also apply standard techniques that minimize a linear function over a convex set. Adding one more degree of freedom in terms of the routing paths simply makes our problem even harder. Only under special conditions -such as when frame length L is larger than the number of virtual links $|\Lambda^v(m)|$ under any m- the optimum joint path, schedule and power assignment is trivial, e.g. each virtual link is placed on a distinct time slot and transmission power is set to the value that is just enough for combating the ambient noise.

In the rest of this chapter, we pursue two of the strongest candidates to solve $P2$:

(1) Benefiting from the fact that most paths can be discarded due to the topological setting, the cardinality of Π becomes small and we can exhaustively solve $P1$ for each $m \in \Pi$ to solve $P2$. Then, the main question is whether we can find a feasible allocation in $\Omega(m)$ and an efficient optimal solution for $P1$ under general circumstances, which makes this strategy a *pseudo-polynomial* one. However, as it will be proved in the next section, the feasibility problem of $P1$ is indeed NP-complete [60], which requires devising near-optimal approximation algorithms.

(2) Since optimization problem $P1$ is NP-hard, $P2$ that is even a harder problem also becomes NP-hard.[7] As a more direct strategy, we will pursue a suboptimal approximation algorithm that jointly searches for the best feasible paths and schedules to minimize the total transmit power.

[7] More formally, any instance of problem $P2$ can be 1-to-1 mapped onto an instance of $P1$ in polynomial time by setting the unique path assignment that corresponds to ordered virtual links in $P2$ as the only path assignment instance in $P1$. Hence, $P1$ is harder than $P2$.

3.4.2 Intractability of the Jointly Feasible Schedule and Power Allocation

Let us first define the following problem.

FP1(feasibility problem): Given the gain matrix G of virtual links, frame length L, session rate and SINR constraints, is there a schedule and power assignment that satisfy both the rate and SINR constraints?

To show the NP-completeness of FP1, we provide an alternative formulation of our optimization problem and its corresponding feasibility question. The alternative formulation assumes that each session has the same BER (or SINR) requirement and the virtual link notion is put aside. Naturally, the gain matrix G and the SINR constraint define a super-set $\mathcal{X}$ of activation vectors $X_1, X_2, \ldots, X_\kappa$ where each X_i have exactly E entries from the binary set $\{$0,1$\}$. The entries with value 1 correspond to the indices of simultaneously transmitting active links while each transmission satisfies the given SINR constraint. Clearly, the vectors majorized by any X_i are also the members of $\mathcal{X}$. Suppose that we also know the power vectors that achieve the minimum total transmit power $P^*_{X_i}$ for each X_i. Then, our objective becomes:

$$\min_{\underline{m}=[m_1 \ldots m_\kappa]^T} \sum_{i=1}^{\kappa} m_i P^*_{X_i} \ , \tag{3.12}$$

subject to

$$[X_1 X_2 \ldots X_\kappa]\,\underline{m} \ \geq \ [\rho_1 \ldots \rho_E]^T \tag{3.13}$$

and to

$$[1 \ldots 1]\underline{m} \leq L \ . \tag{3.14}$$

Here, ρ_i is the total flow rate through link i as determined by routing decisions and session rate requirements, m_i is the number of slots that activation set X_i is used, and L is the number of slots in a frame as before. (3.13) is the short-hand representation of the rate requirements, whereas (3.14) simply states the total number of slots cannot be larger than the frame length. Objective function (3.12) and constraints (3.13)-(3.14) constitute an integer programming problem. Hence, its feasibility problem given below is NP-complete [60].

FP2(alternative feasibility problem): Given the finite set of $\mathcal{X}$ with the associated minimizing power assignments and SINR threshold, is there a E-tuple $\underline{m}$ of integers such that constraint (3.14) is satisfied for fixed L and the rate constraints in (3.13) hold?

Now, we can easily prove that FP1 is also NP-complete.

Lemma 5. *FP1 is NP-complete.*

Proof. Given any schedule and power allocation, e.g. an instance for FP1, it takes $O(M^2)$ time steps to check if the session rates and SINR constraints are satisfied. Therefore, FP1 is in NP.

Consider the following mapping: (i) Since we know the SINR threshold γ of FP2, set γ_i in FP1 as γ. (ii) Starting from the members of $\mathcal{X}$ that has the least number of active link, compute the elements of gain matrix using γ and minimizing power vectors. (iii) For the entries of G that cannot be computed, enter ∞. (iv) Create virtual links for the physical links that have a rate more than 1 slot/frame. (v) Keep the frame length same.

This procedure takes $O(\kappa^2)$ time steps and an instance of FP2 is mapped onto an instance of FP1 in polynomial time. Since FP1 solves exactly the same problem, FP2 reduces to FP1 in polynomial time. However, FP2 is NP-complete and FP1 is in NP, which makes FP1 NP-complete. □

Notice that we have singled out the constraint (3.10) in defining $FP1$. Adding this constraint makes our problem harder in the sense that each virtual link i can be mapped onto a unique virtual source; then, any instance of $FP1$ is m-to-1 mapped in polynomial time onto an instance of the general feasibility problem with single hop paths and number of sessions equal to the number of virtual sources. This suffices to show that original feasibility problem of $P2$ is even harder than $FP1$.

This intractability result demands sup-optimal but efficient algorithms to perform the joint scheduling and power allocation in order to follow our first strategy. Before proceeding with our algorithmic proposals, we need to derive some upper and lower bounds that will provide the guidelines for our heuristics.

3.4.3 Performance Bounds

We want to derive bounds on the total transmit power in a specific slot n in terms of path gains given that the virtual link assignment is feasible. Since the assignment is feasible, the transmit power of link i in slot n satisfies the inequality:

$$P_i \geq \frac{\gamma_i \sigma^2_{R(i)}}{G_{T(i)R(i)}} + \sum_{\substack{j \neq i \\ c(j)=c(i)}} \gamma_i \frac{G_{T(j)R(i)}}{G_{T(i)R(i)}} P_j \ . \tag{3.15}$$

Summing up both sides of inequality (3.15) over all links in slot n and rearranging the terms in the second summation, we obtain:

$$\sum_{i\in s_n} P_i \geq \sum_{i\in s_n} \frac{\gamma_i \sigma^2_{R(i)}}{G_{T(i)R(i)}} + \sum_{i\in s_n} \sum_{\substack{j\neq i\\ j\in s_n}} \gamma_j \frac{G_{T(i)R(j)}}{G_{T(j)R(j)}} P_i \ , \tag{3.16}$$

where s_n denotes the set of virtual links in slot n. Further, we define the following:

$$\Theta_i(n) \triangleq \sum_{\substack{j\neq i\\ j\in s_n}} \gamma_j \frac{G_{T(i)R(j)}}{G_{T(j)R(j)}} \ \ ; \ \ \alpha(n) = \sum_{i\in s_n} \frac{\gamma_i \sigma^2_{R(i)}}{G_{T(i)R(i)}} \ .$$

Note that $\Theta_i(n)$ can be understood as the *effective* interference of virtual link i on other users in the same slot and $\alpha(n)$ represents the capability of slot n to combat the noise term. Thus, we have the inequality:

$$\sum_{i\in s_n} \left(1 - \Theta_i(n)\right) P_i \geq \alpha(n) > 0 \ \ . \tag{3.17}$$

It follows from (3.17) that

$$\max_{i\in s_n}(1 - \Theta_i(n)) \sum_{i\in s_n} P_i \geq \alpha(n) \ . \tag{3.18}$$

Remember that $\alpha(n) > 0$ and this implies $\max_{i\in s_n}(1 - \Theta_i(n)) > 0$ or $\min_{i\in s_n} \Theta_i(n) < 1$. In other words, in order to have a feasible power allocation, the minimum *effective interference* in a slot must be strictly less than one. We will refer to the link which has the minimum effective interference on other links as *minimum interferer*. Hence, we obtained the following lower bound on the total transmit power of a specific slot assignment:

$$\Sigma_n \geq \frac{\alpha(n)}{1 - \min\limits_{i\in s_n} [\Theta_i(n)]} \ , \tag{3.19}$$

where $\Sigma_n = \sum_{i \in s_n} P_i$. When we consider the trivial upper bound for Σ_n using the feasibility constraint $P_i \leq P_{max}$, minimum effective interference must satisfy the additional necessary condition of

$$\min_{i \in s_n} [\Theta_i(n)] \leq 1 - \frac{\alpha(n)}{|s_n| P_{max}} \; . \tag{3.20}$$

In (3.20), $|s_n|$ is the number of links assigned to time slot n. It is also straight-forward to see that the inequalities (3.15)-(3.17) are satisfied with equality at the optimum power allocation. [8] Let Σ_n^* be the optimum total transmit power of slot n, then it must satisfy the following upper bound provided that $\max_{i \in s_n} [\Theta_i(n)] < 1$:

$$\Sigma_n^* \leq \frac{\alpha(n)}{\min\limits_{i \in s_n} (1 - \Theta_i(n))} = \frac{\alpha(n)}{1 - \max\limits_{i \in s_n} [\Theta_i(n)]} \; . \tag{3.21}$$

We can infer from (3.21) that minimizing $\max_{i \in s_n} [\Theta_i(n)]$ both decrements the upper bound and traps the total transmit power within tighter intervals. In addition, if the variation between maximum and minimum effective interference is sufficiently small, the upper bound also becomes a tight one. Quite intuitively, both the upper and lower bounds suggest that we should minimize $\alpha(n)$, i.e. choose a set of links, in which each link has a good channel gain or low SINR requirements. These observations are the main ingredients in the design of our heuristic algorithms which are revealed in the next section.

[8] Otherwise, at the optimal power allocation, some links experience SINR values higher than they require and they can reduce their transmit powers until the SINR threshold is reached without violating the feasibility of other link powers. This is certainly in contradiction with the initial optimality assumption.

3.5 Sub-Optimal Approximation Algorithms for $P2$

In this section, we work on two different strategies to attack the joint path, schedule, and power assignment problem $P2$. The first strategy investigates approximation algorithms to solve $P1$ to provide a pseudo-polynomial solution for $P2$. The second strategy, on the other hand, directly attempts to solve $P2$ by deciding on paths, schedules, and power levels concurrently.

3.5.1 Pseudo-polynomial Approximation: Solving $P1$

We explore two greedy suboptimal algorithms to solve the joint power allocation and schedule assignment problem $P1$. We refer to each algorithm as A and B respectively.

3.5.1.1 Algorithm A

Algorithm A follows a top-down design strategy. It starts with the feasibility problem and searches for the minimum frame length L^* to satisfy the rate and SINR requirements. Clearly, *P1* has a feasible solution if and only if $L^* \leq L$. Once the problem instance is identified to be feasible, the links from congested time slots are shifted to the empty or less congested ones to further reduce the transmit powers of the virtual links. The decision criterion on *which link to be shifted to which time slot* is explained in details below.

Block diagram in Fig.3.3 summarizes Algorithm A. As an initial condition, we are given a sequence of empty time slots and a set of virtual links, each of which has to be scheduled for only once throughout the frame duration. Starting from the first slot, we

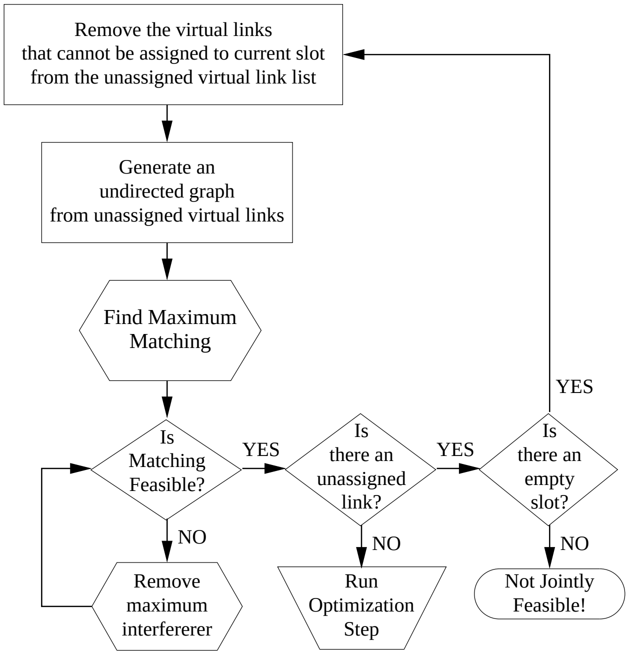

Remove the virtual links that cannot be assigned to current slot from the unassigned virtual link list
Generate an undirected graph from unassigned virtual links
Find Maximum Matching
Is Matching Feasible?
YES
NO
Remove maximum interfererer
Is there an unassigned link?
YES
NO
Run Optimization Step
Is there an empty slot?
YES
NO
Not Jointly Feasible!

Figure 3.3: Block Diagram for Algorithm A.

want to pack as many virtual links among the unscheduled links as possible into a single slot. First, we filter out the virtual links with labels that do not satisfy the constraint (3.10). The remaining unassigned virtual links with their transmitters and receivers form a directed graph that possibly has multiple directional edges between the same vertex pair. Because of the point-to-point and half duplex communication assumptions, we cannot assign any two of these directional edges connecting the same vertex pair to the same slot. Hence, we can replace directional edges with un-directional edges and prune the extra edges connecting the same vertex pairs. In this way, we obtain an undirected graph. The same assumptions further render only simultaneous scheduling of *matching* edges[9] possible. Then, putting as many links as possible in the same slot becomes *maximum matching* problem, which is solvable in polynomial time [60].

[9] These are the edges that do not share a common vertex.

Next step in the algorithm involves (i) one-to-one mapping of the maximum matching back to virtual links and (ii) checking if we have a feasible power allocation for this set of virtual links. When an undirected link in the matching set corresponds to the same directed link, we pick the one that has a smaller SINR threshold, because it has a better chance to satisfy the slot feasibility. In the case where the undirected link corresponds to the links with opposite polarities, we pick any of them. If the maximum matching fails to be feasible, we remove the link with maximum interference on the matching set. This process continues until the matching set is reduced to a feasible one. The matching set is infeasible provided that: (1) Perron-Frobenius eigenvalue ρ is larger than or equal to one or (2) ρ is smaller than one[10], but any of the links fails to satisfy maximum power constraint. Removal of the maximum interferer is beneficial not only in limiting the total transmit power of the matching set (see (3.21)), but also for avoiding the ambiguity in case, where successive removals lead to infeasibility as a result of having $\rho \geq 1$. The virtual links in the resulting matching are pruned from the directed graph and we continue with the next time slot until all virtual links are assigned to a feasible slot.

If we cannot assign all the virtual links for a given frame length L, we declare the problem instance as *not jointly feasible*. In the situations, where all the links are assigned to a number of slots less than L, we run an optimization step to shift the links to non-utilized/under-utilized slots. A greedy approach would be as follows. For a link reassignment a that involves reassignment of link i from slot s to a feasible slot s', we

[10] Then, we can compute the optimal power allocation by matrix inversion.

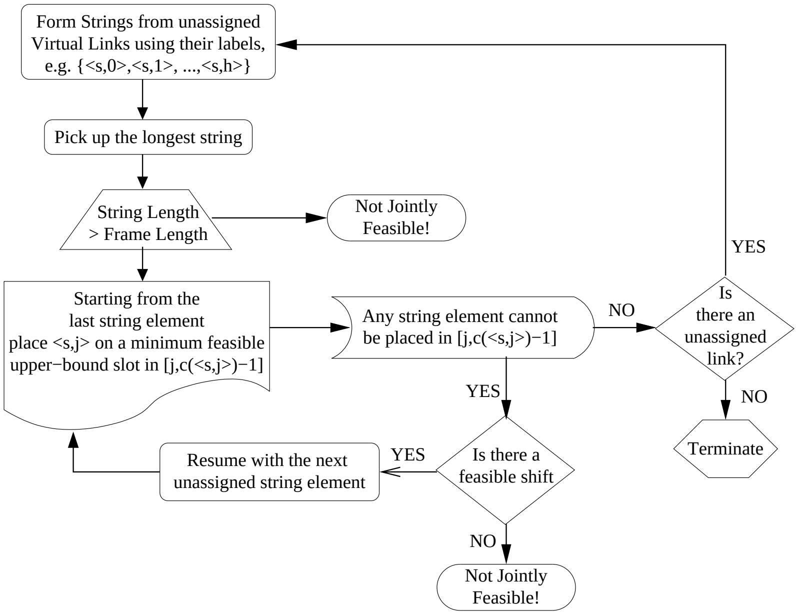

Form Strings from unassigned Virtual Links using their labels, e.g. {<s,0>,<s,1>, ...,<s,h>}
Pick up the longest string
String Length > Frame Length
Not Jointly Feasible!
YES
Starting from the last string element place <s,j> on a minimum feasible upper−bound slot in [j,c(<s,j>)−1]
Any string element cannot be placed in [j,c(<s,j>)−1]
NO
Is there an unassigned link?
YES
NO
Terminate
Resume with the next unassigned string element
YES
Is there a feasible shift
NO
Not Jointly Feasible!

Figure 3.4: Block Diagram for Algorithm B.

compute the factor $\Delta P(a) = P(before) - P(after)$, where P(before) is the total power consumption before the reassignment and P(after) is the total power consumption after the reassignment. The link that is selected for reassignment is the one that causes the maximal power consumption decrease $\Delta P(a)$. The algorithm terminates when no further link reassignments can cause power consumption decrease, i.e. when $\Delta P(a) < 0$ for all reassignments a of links from slots s to slots s'. Evidently, we restrict the re-assignments to the ones that ensure the joint feasibility.

3.5.1.2 Algorithm B

The second strategy on the other hand follows a bottom-up approach (see figure 3.4 for the block diagram of the algorithm). Initially, we form strings of virtual links such that each virtual link labelled by $\{s, h\}$ is appended by another virtual link labelled by

$\{s, h+1\}$ until there exists no virtual link labelled by $\{s, h+n\}$ for some $n \geq 0$.[11] Algorithm B iterates over these strings that are sorted in descending order by their length. At iteration i -i.e. we place the ith longest string onto the time slots-, we perform the following steps:

1. For all elements of the string, rank each feasible slot with respect to the upper-bound as computed by (3.21). The rank increases as the upper-bound decreases.

2. Starting from the end of the string, place the string element labelled by $\{s, j\}$ onto the highest ranked time slot in the interval $[j, \min(c(\{s, j+1\}) - 1, L)]$, where $c(\{s, j+1\})$ is the slot to which the previous string element is assigned to.

3. If there is no feasible slot in the given interval, find the first feasible slot in $[c(\{s, j+1\}), L]$ and shift the already assigned but now violated string elements towards their next best slot in ascending order. The next best slot is defined as the highest ranked slot that does not cross the boundary of the next string element or the first feasible slot after the boundary, whichever is satisfied first. If no such slot is found, algorithm terminates early declaring that no feasible solution exists.

4. If more than one string has the same length, repeat the steps 1 to 3 for each string. Place the string that results in minimum maximum upper-bound, where the maximum is over all slots. Continue to place the next best string provided it does not have a common slot with the previous string.

[11] Each of these strings can be interpreted as virtual paths or circuits.

5. Algorithm terminates with success if all the strings are exhausted.

The intervals in the algorithm are defined to strictly satisfy the constraint (3.10). If that condition is relaxed, Algorithm B simplifies to the case where the iterations are performed over the virtual links rather than the strings. Then, at each iteration, a virtual link-slot pairing that causes the minimum maximum upper-bound is selected as the next assignment.

3.5.2 Polynomial Approximation Algorithm for $P2$

Different from the previous two solutions, we also propose a greedy heuristic (hereon Algorithm C) that performs the route and schedule assignments jointly. In addition to the notion of *virtual links*, we introduce *virtual sources* to distinguish between different paths from the same source node. For instance, if node i has two different paths to reach the infrastructure, we will partition node i into virtually two different nodes, e.g. $i1$ and $i2$. Effectively, we will act as if we have $\sum_i \pi_i$ source nodes with well known virtual links. The block diagram of the algorithm is given in figure 3.5 and the details of each block is explained below:

1. Form the virtual links and prepare the strings as in Algorithm B. Bundle the strings that start with the virtual source nodes of the same physical node together. All the strings are initially included in a list of unassigned strings.

2. Rank the strings in each bundle. Rank decreases as the overall interference-free power sum increases.

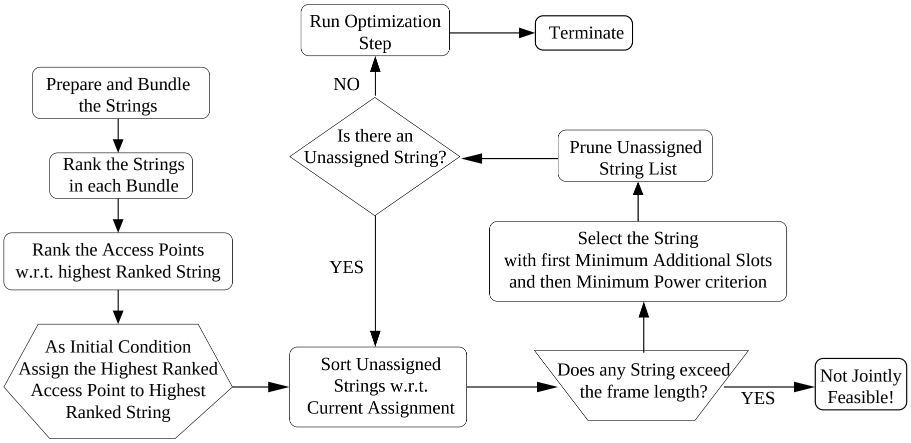

Run Optimization Step
Terminate
Prepare and Bundle the Strings
NO
Is there an Unassigned String?
Prune Unassigned String List
Rank the Strings in each Bundle
Select the String with first Minimum Additional Slots and then Minimum Power criterion
Rank the Access Points w.r.t. highest Ranked String
YES
As Initial Condition Assign the Highest Ranked Access Point to Highest Ranked String
Sort Unassigned Strings w.r.t. Current Assignment
Does any String exceed the frame length?
YES
Not Jointly Feasible!

Figure 3.5: Block Diagram for Algorithm C.

3. Count the occurrences of each access point as the last string element in each highest ranked string of all bundles.

4. Rank the access points, i.e. an access point has a higher rank if it has a higher count.

5. Assign the lowest ranked access point with non-zero count to the highest ranked string that has lowest interference-free power-sum among all bundles. Place the virtual links that correspond to the string elements onto adjacent time slots starting from the first slot. Remove all the strings in the same bundle of the assigned string from the list of unassigned strings.

6. Mark the current number of utilized slots. Iterate over each unassigned string:

   (a) Place the string elements on the first feasible slots while preserving their order in the string and using minimum number of additional slots.

   (b) Find the string that keeps the total number of used slots minimal. If more

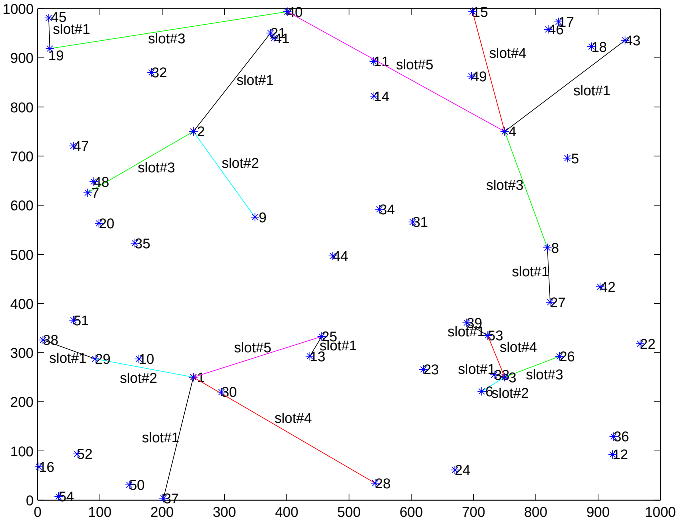


Figure 3.6: An assignment instance generated by Algorithm C in a multi-hop cellular topology with fixed base-station locations at the center of square cells and randomly distributed relay/source nodes over a 1000mx1000m topology. The scenario involves 15 sessions with varying error rate requirements and outdoor shadowing environment.

than one string exist, take the one that results in minimum total power.

(c) Remove all the strings in the same bundle as the assigned string from the unassigned string list.

(d) If no feasible assignment is found, terminate by declaring the joint assignment as infeasible.

7. If the number of utilized slots is larger than the frame length, terminate by declaring the joint assignment as infeasible. Otherwise, run the optimization step as in algorithm A.

3.6 Experimental Evaluation

We have investigated the performance of our heuristic proposals on a $1000m \times 1000m$ square topology. The network is partitioned into four square cells and four nodes are positioned at the center of each cell. These nodes at the cell centers can be viewed as cluster heads that concentrate traffic in each ad hoc domain to relay to the other domains or access points/base-stations of an infrastructure/overlay network. The remaining wireless nodes are randomly distributed over the whole topology and the source nodes are also randomly selected among them. We have allocated the same amount of bandwidth for each session, but used different bit error rate (or equivalently SINR) constraints. As performance metrics, we are interested in: (i) the success rate of each proposal in finding a feasible solution and (ii) the mean total transmit power, where averaging is performed only over the feasible solutions.

In this section, we provide two sets of experiments. The first set evaluates the performance of pseudo-polynomial algorithms A and B in solving $P1$, since their success in solving $P2$ relies on their performance in solving $P1$. The second set, on the other hand, directly compares the performance of all three heuristics to solve $P2$.

3.6.1 Solving $P1$: Algorithm A versus Algorithm B

We can summarize the simulation environment and how the parameters are set as follows. Each session as identified by its source node has a fixed rate requirement of 1 slot/frame. They are randomly assigned SINR threshold values from the set $\{4, 5, 6, 7, 8\}$. The noise power is assumed to be same at each receiver and transmit powers are normalized with respect to the noise power. The channel gains are computed by only taking the path loss factor into account with the path loss exponent of two for transceiver pairs close than 100 meters and of four otherwise (i.e. two-ray ground reflection model with distance cross-over [61]). We set the maximum normalized transmission power to be 31.25, which corresponds to a transmission range of 250 meters at the highest SINR requirement. Each wireless node is assigned to a base station that is closest to its location in the Euclidean sense. We have considered two different shortest path routing schemes for each given scenario with link costs equal to either the unit value (i.e. minimum-hop routing) or to the transmission power just enough to combat the noise for the specific session (i.e. minimum-power routing).

We use the frame length in number of slots and number of sessions as the variable system parameters, while keeping the session requirements fixed. Note that altering the frame length is essentially equivalent to keeping the frame duration fixed and altering the traffic load in terms of the session rate requirements. For cases where the system parameters make the size of problem instances manageable, we have also computed the performance of optimal solutions that are found by exhaustive search.

Figures 3.7 and 3.8 show the average performance for the scenarios where we limit the number of sessions to seven and use the minimum-hop routing. In the plot legends, when there is an *upper-bound* label next to the algorithms A and B, it indicates that the upper-bound in (3.21) is used in the heuristics instead of the actual total transmission powers of the slots. Similarly, the *actual power* label corresponds to the utilization of the actual power levels in the greedy heuristics to rank the slots. Quite interestingly, we observe that Algorithm A, which is specifically designed for first finding a feasible solution, is actually outperformed by Algorithm B, which relies on the upper-bound formulation we derived. In other words, a water-filling argument with a proper cost function can actually be more successful than a top-down design strategy such as Algorithm A. However, an inadequate cost function that does not assist in distributing the links, which exhibit high interference to each other, onto different slots results with a degraded performance as seen in figure 3.7. Algorithm B also matches the performance of optimal solution within 10% margin in finding a feasible solution.

On the other hand, when the total power consumption as summed over all virtual links is observed, Algorithm A executes much better than the other heuristics. Algorithm B also performs comparable to optimal solution when the actual power values are utilized to rank the slots and links. However, due to the lower success rate in identifying feasible solutions, Algorithm B with the actual power heuristic can resolve only the less constrained scenarios and the lower power consumption figures should not be misleading. Clearly, the optimal solution performs better at each problem instance. The overall suggestion of the power consumption results is that greedy approaches, which directly operate on the objective function, have an advantage in minimizing the objective function.

Figures 3.9 and 3.10 present the same topology and session requirements, but at a different routing strategy, i.e. minimum-power routing. Since the problem size is quite large, we did not compute the optimal values. Nevertheless, we know that the optimum strategy when L gets large enough is to schedule one link at a time. Therefore, we show

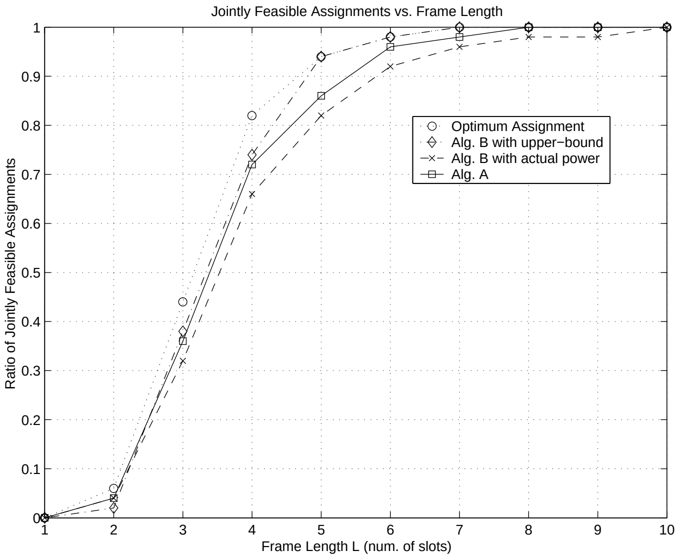


Figure 3.7: Ratio of feasible scenarios for 7 sessions and minimum-hop routing.

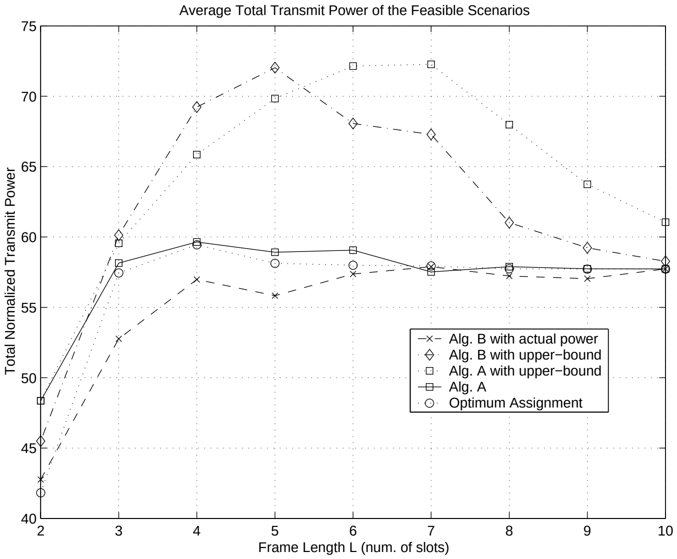


Figure 3.8: Total transmit power averaged over the feasible scenarios for 7 sessions and minimum-hop routing.

this asymptotic result in total power consumption figures. The relative performances have similar tendencies as in the minimum-hop routing except for the following points: (1) Using power as an explicit factor in link costs for routing protocols significantly ameliorates the overall power consumption. (2) Higher number of active links forces the

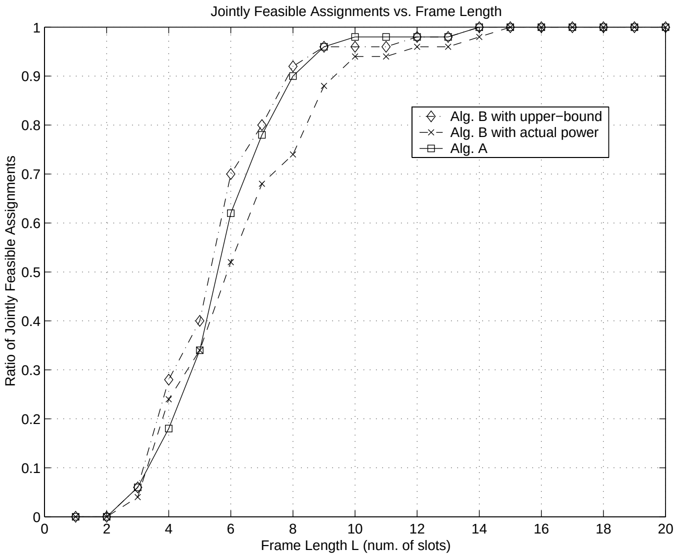


Figure 3.9: Ratio of jointly feasible scenarios for 7 sessions and minimum-power routing.

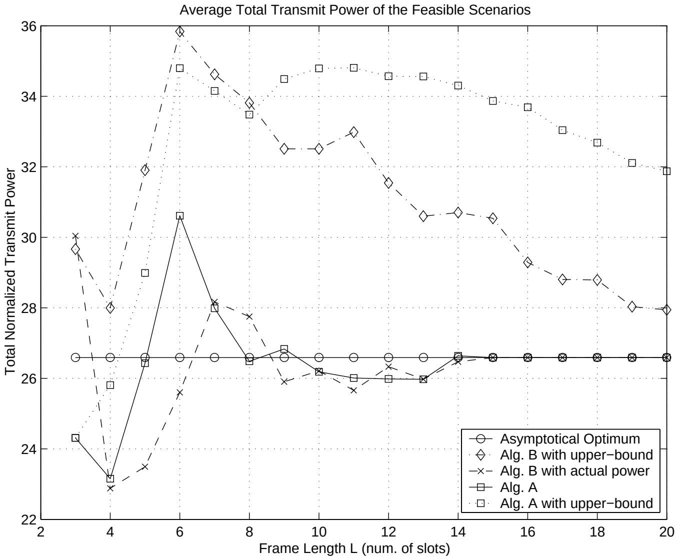


Figure 3.10: Total transmit power averaged over the feasible scenarios for 7 sessions and minimum-power routing.

system to use longer frame lengths to satisfy the session requirements. Thus, reducing the power consumption in the routing layer often fails to satisfy the session QoS requirements even at moderate frame lengths.

Figures 3.11 and 3.12 give more insights when the network load is increased by

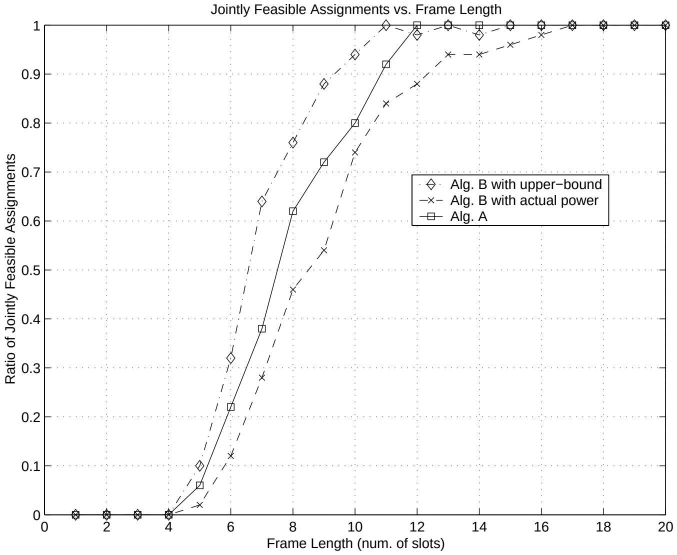


Figure 3.11: Ratio of the feasible scenarios for 15 sessions and minimum-hop routing.

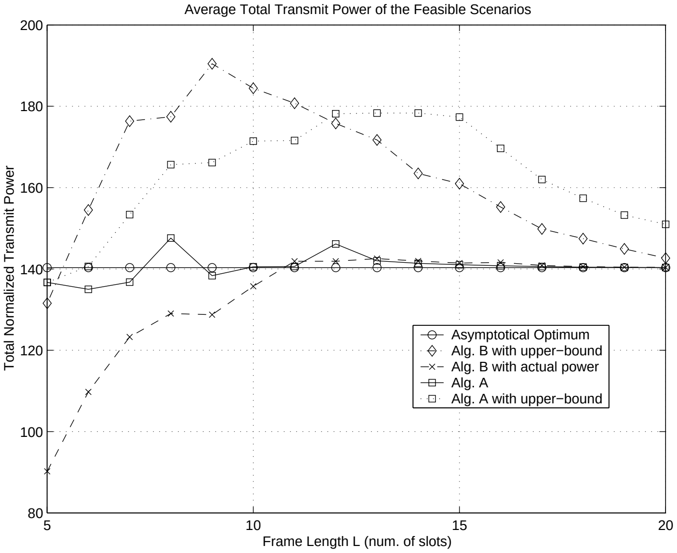


Figure 3.12: Total transmit power averaged over the feasible scenarios for 15 sessions and minimum-hop routing.

changing the number of sessions from 7 to 15. The relative performances remain same with wider performance gaps and the nominal values of the operating points get worse both in terms of the required frame length to satisfy the session requirements in majority of the scenarios and the settled down total power consumption.

3.6.2 Solving $P2$: Pseudo-polynomial versus Polynomial Algorithms

Different from the previous simulations, we are now interested in the performance of our algorithmic proposals in solving $P2$. We use the same $1000m \times 1000m$ square topology and uniform bandwidth request among the users in our experiments with the following nuances: (1) Instead of using a channel gain that depends only on the inverse powers of the distance between transceiver pairs, we employ shadowing model:

$$\left[\frac{P_r(d)}{P_r(d_0)}\right]_{dB} = -10\beta \log \frac{d}{d_0} + X_{dB} \ ,$$

where d is the distance between transceiver pairs, d_0 is the reference distance the receiver power P_r is measured, β is the path loss exponent, and X_{dB} is a gaussian random variable with zero mean and standard deviation σ_{dB} (i.e. shadowing deviation). We set values as $d_0 = 1m$, $\beta = 3$, and $\sigma_{dB} = 4$, which reflects an outdoor environment [61]. (2) Since each session has alternative paths with different hop lengths, the BER constraints shall map onto different SINR constraints (see figure 3.2). Therefore, instead of selecting an SINR value, we picked an error exponent n randomly from the set $\{4, \ldots, 9\}$ that corresponds to a bit error rate of 10^{-n} for each session. (3) We selected the alternative paths as the minimum-power paths toward each feasible access point.[12]

[12] Minimum-power in the sense that link costs are assigned to transmit powers just enough to combat the noise power. Note that the link cost depends on the SINR threshold, which depends on the path length. Thus, in a typical dynamic programming method such as Dijkstra's algorithm, the links costs are updated at each step. Nevertheless, as the path length increases, the link costs also increase. This is sufficient for the principle of optimality to hold and we can still find the optimal paths by applying Dijkstra's algorithm with link cost updates at each iteration.

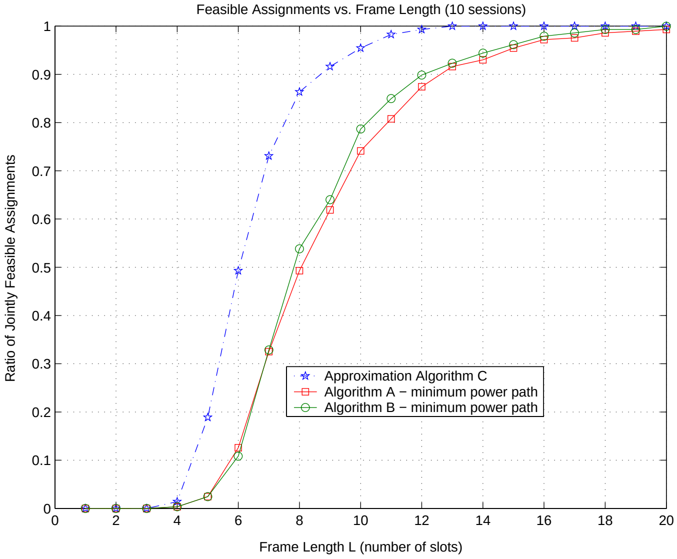


Figure 3.13: Ratio of jointly feasible scenarios for 10 sessions: Algorithms A and B are applied onto a single path, which is the same as the minimum-power path to the overlay tier.

Unlike the previous section, we have an additional degree of freedom in terms of access point assignment. To illustrate how such a cross-layering can be quite effective, we compared Algorithm C, which performs path, schedule, and power assignment simultaneously, against Algorithm A and B that are restricted to a single path assignment instance, where each wireless node is assigned to the access point that has minimum routing cost when minimum-power routing is applied. Figure 3.13 plots the performance in satisfying the QoS constraints of the wireless users for the case of 10 sessions. Although the base station assignment strategy is different from the previous section, we observe that relative performance of Algorithms A and B remain same. However, what is significant is that Algorithm C outperforms as much as 40% better than other heuristics. If the strategy is to give a performance margin of 90% success in satisfying the user requirements, Algorithm C achieves the goal in a frame-length of 9 slots, whereas

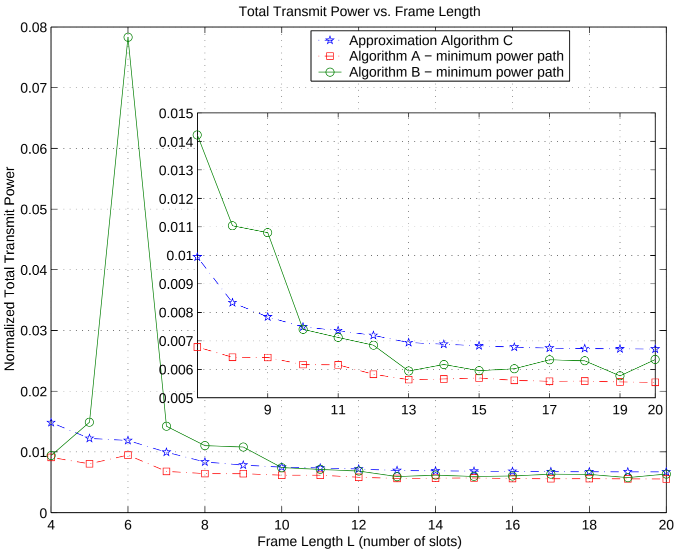


Figure 3.14: Total transmit power averaged over the feasible scenarios.

Algorithm B and C achieves the same goal in 12 and 13 slots respectively. A very tight margin such as 99% is achieved in 12 slots by Algorithm C and 19 slots by other heuristics with layer separation. This is a clear illustration of how a simple cross-layering can drastically ameliorate the QoS provisioning. Our observation holds under different session loadings such as 7 and 20 sessions.

In terms of minimizing the objective function, we first examined the total power dissipation of each heuristic as averaged over the scenarios that they find a feasible solution. However, each algorithm may find a feasible solution for a different subset of the scenarios. For this reason, we calibrated the results such that the total power dissipation of each heuristic is averaged over the scenarios, where all three algorithms agree on their feasibility. Figures 3.14 and 3.15 show both situations again over a 10 session scenario with power axis normalized with P_{max}. As the frame length increases, the overlap between the heuristics increases and ultimately becomes the same when they

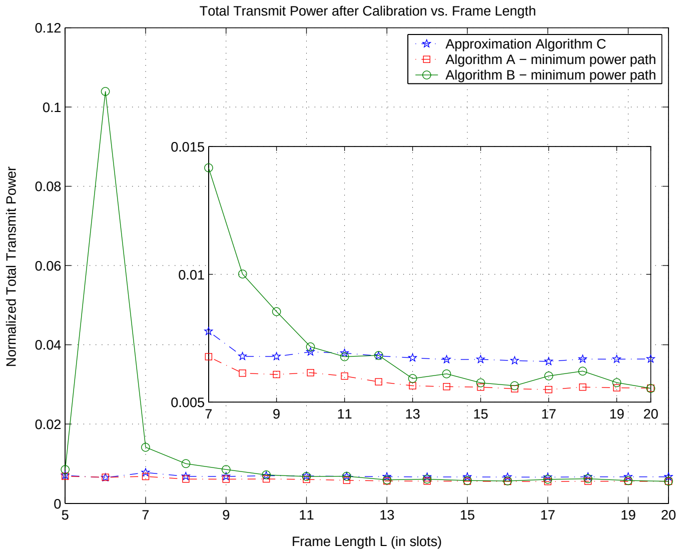


Figure 3.15: Total transmit power averaged over the feasible scenarios common to all algorithms at the load of 10 sessions.

show 100% success in finding a feasible solution. For both cases, Algorithm A outperforms the other heuristics in compliance with the results of the previous section. Algorithm B, which relies on the upper-bound expression, performs better as the success rate of finding a feasible solution crosses 80% line. According to calibrated results, Algorithm C settles down on a suboptimal solution without sacrificing more than 15% of performance. The situation is different for the non-calibrated results, where initially as much as 50% performance difference is observed for short frame lengths, because Algorithm C finds solutions for harder scenarios that Algorithm A cannot solve. Evidently, these scenarios are subject to consume significantly more power.

As we exhaustively search all feasible access point assignment instances and apply Algorithms A and B, we expect much better performance in terms of identifying a feasible solution and reducing the power consumption. This expectation is also supported by our experiments. Figure 3.16 shows that the gap between algorithms reduces significantly from 45% to less than 10%. Nevertheless, Algorithm C still performs better than the other heuristics. As we reduce the session load further, pseudo-polynomial approaches take the lead (e.g. see figure 3.17 for the case of 7 sessions). This suggests that pseudo-polynomial algorithms that are manageable for small size problems can be exploited to obtain better results in both satisfying the QoS requirements and minimizing the objective function. However, Algorithm C which has much faster execution time than the pseudo-polynomial approach can be effectively used for larger problem sizes.

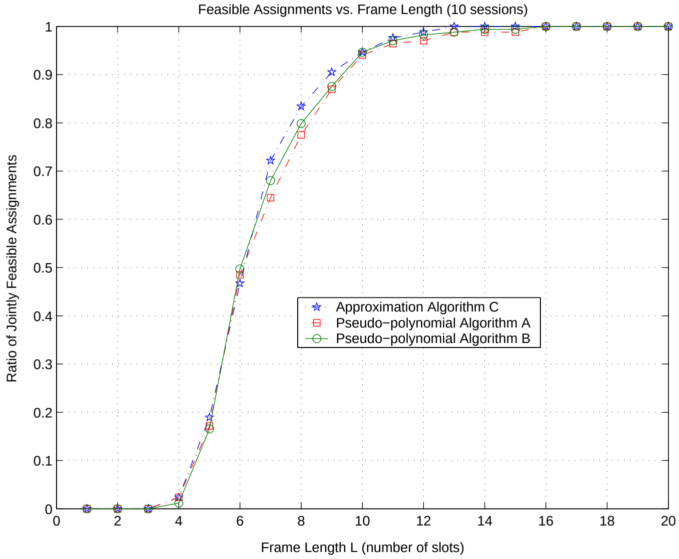


Figure 3.16: Ratio of jointly feasible scenarios for 10 sessions.

Even when we restrict algorithms A and B to the minimum-power routing, their performance exceeds that of Algorithm C. Figure 3.18 shows the calibrated power consumption trends when algorithms A and B used with pseudo-polynomial search that eliminates the infeasible access point assignments. Originally, Algorithm B was trailing behind for frame lengths less than 11 slots when the access point assignment instance is

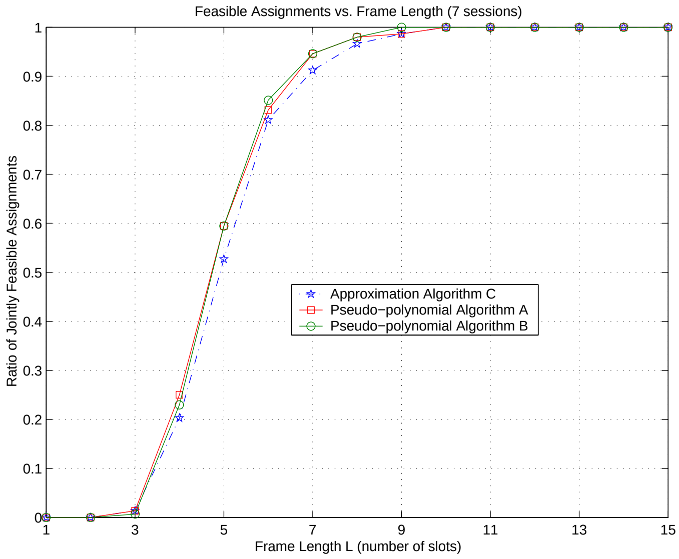


Figure 3.17: Ratio of jointly feasible scenarios for 7 sessions.

determined independently. With the exhaustive search, this level reduces to 7 slots under the same load. Algorithm A consistently offers the best results, however Algorithm B demonstrates more smooth behavior by positively benefiting from access point assignments.

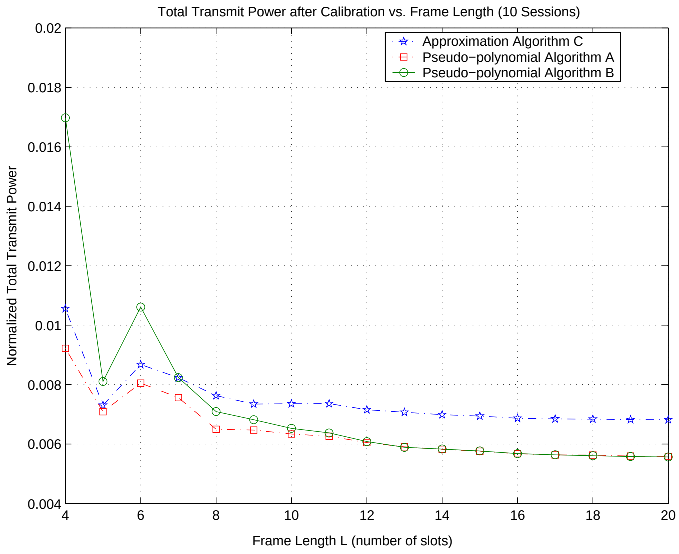


Figure 3.18: Total transmit power averaged over the feasible scenarios common to all algorithms at the load of 10 sessions.

3.7 Summary

In this chapter, we concentrated on a small scale networking scenario, where we have a finite number of feasible access points for each wireless user to utilize a high-bandwidth overlay network. Since multiple such small scale domains potentially operate over the same radio frequencies and wireless users are subject to limited battery life, we formulated our problem to suppress the energy emanation outside of each domain while satisfying the minimum connectivity requirements of the users in terms of bandwidth and error rate. Given the fact that the independent decisions on different layers for achieving a local objective would deteriorate the performance of other layers and lead to a failure in achieving the main goal, we followed a cross-layered framework that mainly addresses the joint transmission power, link schedule and path assignment problem.

Our main assumptions within this framework have been that (i) there exists a one-to-one mapping between BER requirements and SINR thresholds and (ii) channel conditions are slowly varying or stationary. By introducing the notion of virtual links, we decoupled the joint optimization problem from the underlying session based requirements. We proved the NP-completeness of the feasibility problems associated with the joint scheduling and power control problem as well as the joint scheduling, power control and path assignment problem. Therefore, we introduced three heuristic solutions as suboptimal but efficient algorithms. The performance of these heuristic algorithms established the following points:

(1) Water-filling techniques that rely on appropriate metrics perform better in solving the feasibility problem for joint scheduling and power control problem than the

top-down designed greedy algorithms that attack the feasibility problem first. The upper-bound expression developed in section 3.4.3 proves to be such a reliable metric, which eliminates strong interferers evenly across the available channels.

(2) Top-down algorithms that greedily operate on minimizing the total power consumption at each iteration turn out to be more effective in terms of minimizing the objective function.

(3) Routing layer plays a dominant role in reducing power consumption, but it happens at the expense of increasing failures in QoS provisioning.

(4) Coupling access point assignment into the joint scheduling and power control problem significantly improves the QoS provisioning (as much as 45% in our experiments) while reducing the power consumption.

(5) A simple mixed strategy[13] that concurrently does the path, power and schedule assignment implements a very fast algorithm while attaining better performance in finding feasible solutions as opposed to pseudo-polynomial algorithms that solve each path assignment instance via approximation algorithms. However, this comes at the expense of performance loss in terms of minimizing the total power sum. Our experiments indicate that the loss can be as much as 22% mainly due to the suboptimal access point assignments.

[13] Mixed in the sense that it employs both top-down design strategy by attacking first at the feasibility problem and bottom-up design strategy by placing a whole path over the slots at each iteration.

Chapter 4

Service Discovery in Wireless Hybrid and Ad Hoc Networks

In this chapter, we shift our focus from network level diversity to the service level diversity in multi-hop wireless networks. Our goal is to investigate the architectural choices along with their required network layer support and to propose a comprehensive framework to discover and allocate the services in wireless and hybrid networks. Before providing further details, let us examine the steps that are involved in a typical (i.e. cellular networks and WLANs) wireless network operation to start communication from any node.

In its simplest form of service discovery, nodes have to establish a connectivity with an access point or base station. Nodes, then, reach a configuration server such as DHCP [3] or DRCP [4] to configure their network interfaces with routable IP addresses.[1] The same agents can also be used to locate gateway nodes and DNS servers. The next step is to find out communication end points that are in particular interest of the wireless user. The common way of performing this task in the application level is via DNS servers or

[1] The identities of DRCP/DHCP servers are retrieved from the access points in general if not hard coded.

RServPool name servers. One step beyond is to establish acceptable level of connectivity with the end points. To increase scalability, security, and robustness, this requires a communication with special purpose nodes such as bandwidth brokers or public key managers. Since we have a single hop wireless network with well defined domain boundaries, this step by step process to establish a real communication is straight-forward to implement. However, when we shift toward multi-hop wireless environments with fuzzy domain boundaries and mobile routers/gateways (e.g. tactical networks, future combat systems, multi-hop wireless infrastructure networks, etc.), the same steps cannot be directly applied because of the following reasons:

- In general, mobile nodes are not within direct communication range of any configuration agents.

- These agents are stationed on a fixed number of and possibly mobile, resource abundant nodes. Depending on the mobility of ad hoc nodes, there can be partitions that have access to only a sub-set of the services, e.g. a partition may have DRCP servers, but may have no DNS servers. To decrease the vulnerability against single point of failures, the critical services are distributed among the nodes rather than locating them on a single node.

- These services are usually neither designed nor suitable for transferring their functionality to other nodes.

- In most cases, more than one agent is available for a particular service and the decision on which agent to use has a direct impact on the network resources.

Moreover, we need a service-centric network operation as the actual communication demand itself becomes more and more oriented to accessing a particular service rather than talking with a peer node. Service discovery is the key technology to enable such a networking environment.

4.1 Service Discovery for Multi-hop Wireless Networks

Service discovery protocols can be evaluated under a few different contexts. For us, the distinguishing characteristic of these protocols is their applicability to highly dynamic, multi-hop wireless ad hoc networks. To be applied directly to such networks, a service discovery protocol should operate smoothly when (i) no fixed infrastructure is available, (ii) there are multiple servers offering and multiple clients requesting the same type of services, and (iii) no direct link between clients and servers exists.

The first condition is desirable, because as the network topology changes and partitions occur, we want the wireless ad hoc network to reconfigure itself automatically and communicate independently in each partition. This has a direct impact on the architectural choice of the service discovery protocols, i.e. *directoryless architectures*, which does not require intermediary directory agents (DAs), become more favorable. Unless directory architectures comply with the following two properties, they are not suitable for implementation in a dynamic networking environment.

(i) *Simplicity*: The first property concerns about the nodes that can carry out the functions of DAs. If DAs are very specialized nodes having requirements of bridging between different communication media (e.g. *salutation manager (SLM)* in Salutation protocol [17]) or containing the objects to access the services (e.g. *lookup server* in JINI [15]), these functions can be impossible or very costly to relocate to another node in the network without any user intervention. Therefore, these DAs constitute an infrastructure contrary to the initial requirements. On the other hand, if DAs simply contain records of services registered dynamically by the servers, any node with enough resources (e.g. battery power, memory, processing power, etc.) can assume the responsibility of being a DA, which is essential for infrastructure-less operation.

(ii) *Adaptivity*: Though the simplicity of DA functions is desired, it is not self-sufficient. We also need mechanisms to dynamically (s)elect the nodes that DAs will reside in and make their locations known to other nodes as the topology changes. In principle, these mechanisms should be as same as forming clusters or virtual backbones in mobile ad hoc networks at the network layer except for the close coupling with service discovery in the application layer [62, 63, 64, 23, 65, 66, 67, 68].

The second and third conditions entail the provisioning of an efficient and yet satisfactory network layer support to distribute the actual service discovery control messages under a dynamic topology. If there are multiple servers (DAs) and no direct links exist between clients and servers (DAs), the network support must be in the form of broadcasting, multicasting, or anycasting. For multi-hop wireless networks, broadcasting may have excessive control message overhead which is quite crucial in shared wireless channels and unintended nodes have to receive, process, and re-transmit these packets which wastes the scarce network resources. Even if efficient broadcasting mechanisms that intelligently select a subset of nodes to relay the packets are used, they require a coordination among the mobile nodes. Nevertheless, one can do more than just broadcasting with the associated coordination overhead in the context of service discovery. We will therefore not consider broadcasting in this work as a viable alternative.

In multicasting, server and DA nodes (depending on which architecture is being used) are assigned well-defined multicast addresses so that they can be reached by all client nodes[2]. The formation and maintenance of multicast groups in terms of clients, servers, and DAs may incur significant costs in network operations. Thus, even directoryless systems, which are preferred because of their light-weight, may in fact turn out to be heavy-weight in the overall cost when multicasting is used. On the other hand, anycasting [69] can provide a simpler framework, since in practice it can be simulated by any unicast routing protocol [70], which is clearly less demanding than multicasting. We should however point out that anycasting does not differentiate the attributes of the services. Service requests will eventually be received by only one of the servers that may not satisfy the client request[3]. Evidently, assigning a different anycast address for each service type and attribute combination is not a feasible approach. Hence, anycasting has a limited scope as compared to the cases where service attributes may exhibit a high

[2] In case directory system is used, DAs themselves actually become servers providing directory services. Then other servers are treated just like the remaining clients from the perspective of DAs.

[3] One may also consider sending application data directly to the anycast address reserved for a service type rather than first discovering the server location. But then, subsequent data packets will possibly be routed to different servers and still the server attributes are not distinguished.

variety and clients have preferences[4]. It is noteworthy to state that anycasting can be extended to provide quality of service by differentiating services in terms of server performances and client requests [71, 72], but it would again require special address resolvers or service brokers to monitor the network resources and relevant activities. This brings us back to the directory system models and associated overheads.

We have already pointed out the similarities between virtual backbone or cluster formation and implementing a directory architecture. If the control message distribution and (s)election process of DA nodes are de-coupled from each other, clients and servers must first discover the DA nodes as if a directoryless system is in place. Otherwise, the second and third conditions basically impose a control message distribution support on top of the virtual backbone or clusters as well as a close interaction between the backbone management sublayer in the network layer and the service discovery agents in the application layer. We prefer the latter approach, because the already collected information during the (s)election of DA nodes can be effectively reused to form a mesh of DA nodes and to locate them. In the following sections, we unravel our own algorithmic solutions to implement a directory system for highly dynamic wireless networks.

[4] For instance, a client may request a color printer in a close location with low number of jobs queued. If all the printers are assigned a single anycast address, the user preference will be ignored in the service query.

4.2 Network Model and Notation

In this section, we present the network model and the notation that we employ in the rest of the chapter.

4.2.1 Network Model

We assume that all the nodes in the network have an omni-directional antenna and have the same transmission power. All links are assumed to be symmetric, i.e. if node A can hear node B, then node B can also hear node A. Nodes share the same communication channel (e.g. same frequency band, same spreading code or frequency hopping pattern) to transmit and receive the control packets. Hence, no node is allowed to transmit and receive at the same time to avoid self-interference. No particular assumption is made on medium access control (MAC) and access scheme can be random, reservation based, or any variant of both. Without loss of generality, partitioning in the network is not allowed, because each partition may then be treated as an independent network.

4.2.2 Notation and Definitions

We use a color convention in determining the roles of each node in the network. DAs are represented by *black* color. If a node is not part of the virtual backbone and it has at least one DA (i.e. black) neighbor, then it is called to be *associated* with the virtual backbone and it is represented by *green* color. When a node is neither DA nor associated with a DA, it is represented by *white* color. The remaining notation and the definitions are as follows.

- N: Set of all the nodes in the network.

- $N_i^{(d)}$: Set of nodes that are at most *d* hops away from node i excluding node i itself.

- W, G: Set of white and green nodes in N.

- c_i: Color of node $i \in N$, which can be black, green, or white.

- VAP_i: Virtual Access Point (VAP) of green node i. This node is used by node i as its access point to the backbone. If node i is a server, it always registers its service with the DA residing on node VAP_i.

- d_i, dw_i: Degree information for node i, i.e. total number of neighbors (or *degree*) and total number of white neighbors (or *effective degree*) of node i in the given network topology.

- $NLFF_i$: Normalized link failure frequency of node i. This parameter represents the total number of link losses for node i in a fixed time window normalized by d_i at the end of the observation window.

- $nlff_{th}$: System threshold that sets the preferred level of normalized link losses for the backbone nodes.

- ID_i: Network identifier for node i.

- T_w, T_l, T_s, T_h: Waiting time, long time, short time, and hello beacon periods which are ordered as $T_w > T_l > T_s > T_h$.

- $\overset{(l)}{>}$: lexicographical comparison, i.e. $[a_1, a_2, \ldots, a_n] \overset{(l)}{>} [b_1, b_2, \ldots, b_n]$ if and only if $a_1 > b_1$ or $a_1 = b_1$ and $a_2 > b_2$ or $a_1 = b_1$ and $a_2 = b_2$ and $a_3 > b_3$, so on...

We will also use the terms black nodes, VAP and backbone nodes interchangeably throughout the chapter.

4.3 A Directory Architecture Solution for Service Discovery

Our network level solution to support a directory architecture consists of two parts: BackBone Management (BBM) phase and Distributed Service Discovery (DSD) phase.

The first part, BBM phase, selects a subset of the network nodes to form a relatively stable *dominating set*, discovers the paths between dominating nodes, and adapts to the topology changes by adding/removing network nodes into/from this dominating set. The formation algorithm for the dominating set is very similar to the *backbone selection phase* used in our earlier work VDBP [23]. But, the way in which we incorporate the effect of link failures and interconnect the VAP nodes are quite different. In VDBP, the node with minimum *NLFF* selects itself as a backbone node. Instead, we only eliminate the nodes with *NLFF* values higher than a given threshold (i.e. stability constraint) and use the degree (or effective degree) as the selection criterion for the remaining nodes. Note that $NLFF_i$ is simply the proportion of the link losses at node i. Relying on a threshold value eliminates the extreme cases as seen in VDBP, which selects a node i with no link losses but very few neighbors as the backbone node rather than a node j with very high degree but a few link losses. Unlike most of the other backbone or clustering algorithms, BBM utilizes only a 1-hop local broadcast control message (*Hello Beacon*) for forming the backbone, creating virtual links between backbone nodes, and maintaining the backbone. Hello beacons are also light-weight, because they do not carry all the neighborhood information of the transmitting node.

After the first part is successfully carried out, we have a virtual backbone that constitutes a mesh structure with the backbone nodes and the virtual links connecting them. The second part, DSD phase, is used to efficiently distribute the request and registration messages from the service discovery agents to the DAs. These messages assist in forming multicast trees rooted at client and server nodes on top of the backbone mesh to further increase the signaling efficiency.

The detailed descriptions of both parts are provided in the subsequent subsections.

4.3.1 BackBone Management (BBM) Phase

The goal of the BBM algorithm is to obtain a small size (not necessarily minimum size) and relatively stable backbone. The algorithm is highly distributed and is based on local decisions which enable it to rapidly react back to the changes in the network topology. BBM algorithm can be described in three components: (i) initial selection of backbone nodes, (ii) mesh formation by finding the paths between backbone nodes, (iii) and maintenance against topology changes. All the components rely on the periodically broadcasted *hello beacons* which bear the following information about the transmitting node i: $\{ID_i$, d_i, dw_i, $NLFF_i$, c_i, $VAP_i, flags, routing\ information\}$. Each node creates a *neighborhood information table (NIT)* and a *routing table* using the information carried by these beacons.

4.3.1.1 Backbone Selection

Initially, e.g. when first powered on, every node is assigned white color. Before deciding on their role in the network, white nodes collect hello messages and built up their own neighborhood information table (NIT) for a time period T_w.[5] Each node also caches its own degree, effective degree, and NLFF information as advertised in its last hello beacon and uses the *cached parameters* rather than the actual ones in decision step for a synchronized view between the neighbors. At the end of the waiting period, any white node k, which complies with the stability constraint (e.g. $NLFF_k \leq nlff_{th}$), joins the virtual backbone and becomes black if it has the highest *effective* degree among the other white nodes in its NIT that also satisfy the stability constraint. Ties are broken by giving strict priority to higher ID nodes. Checking the normalized link loss threshold helps to avoid the nodes with a lot of link losses relative to their total number of links becoming backbone nodes. If no white node in 1-hop neighborhood of node k including itself has a link loss rate lower than the threshold, then node k decides as if its $nlff_{th}$ is set to ∞. Effective degree information is checked to force undecided subnets in the network to continue the process independent from the decided components. If a node is still white after its waiting period because it does not have the best effective degree among its white

[5] As a high level link failure detection, an entry for a specific neighbor is erased from NIT unless any hello beacon is received from that node for an integer multiple of T_h.

```
\\Constants: $nlff_{th}$, $T_w$, $T_h$, $\alpha$

\\Initialization:
1. $\forall i \in N$ if ($c_i == \phi$)
  1.1. $c_i = white$ ;
  1.2. set timer_ = $T_w$ ;

\\Waiting Period:
2. $\forall i \in W$ if (*HELLO* received from k)
  2.1. Update NIT ;
  2.2. Update $NLFF_i$ ;
  2.3. if (($T_w \neq 0$) && ($c_k$ == black))
    2.3.1. $c_i = green$ ;
    2.3.2. $VAP_i = k$ ;
    2.3.3. Expire timer_ ;

\\Timer Expires
3. $\forall i \in W$ if (timer_ == 0)
  3.1. if ($(N_i^{(1)} \cap W) \neq \phi$)
    3.1.1. if ($\nexists k \in ((N_i^{(1)} \cap W) \cup \{i\})$ s.t. $NLFF_k \leq nlff_{th}$)
      3.1.1.1. $nlff_{th} = \infty$ ;
    3.1.2. if $NLFF_i > nlff_{th}$
      3.1.2.1. timer_ = $\alpha T_h$ ;
      3.1.2.2. continue from step 2;
    3.1.3. if $([dw_i, ID_i] \overset{(l)}{>} [dw_k, ID_k])$ $\forall k$ s.t.
           $((k \in (N_i^{(1)} \cap W))\&\&(NLFF_k \leq nlff_{th}))$
      3.1.3.1. $c_i = black$ ;
  3.2. else if ($(N_i^{(1)} \cap W) == \phi$)
    3.2.1. if ($\nexists k \in (N_i^{(1)} \cap G)$ s.t. $NLFF_k \leq nlff_{th}$)
      3.2.1.1. $nlff_{th} = \infty$ ;
    3.2.2. $VAP_i = \underset{\substack{k\in(N_i^{(1)}\cap G)\\ NLFF_k\leq nlff_{th}}}{\arg\max} [d_k]$ ;

\\Arbitration:
4. $\forall i \in G$ if ((HELLO received from k) && ($ID_i == VAP_k$))
  4.1. $c_i$ = black;
```

Table 4.1: Pseudo-code for backbone selection phase.

neighbors, it extends its waiting period to receive more hello messages. This extra waiting time must be in the order of hello beacon interval T_h. At any point in the waiting time period, if a white node k receives a hello message from a black node l, k associates itself with l and l becomes VAP node of k. Thus k becomes a *green* node. If a white node remains as the only white node in its neighborhood at the end of the waiting time, then this node must select a green node as its VAP node by giving strict priority to the nodes that first satisfy the $nlff_{th}$ requirement and secondly have a higher degree. When any green node i receives a hello beacon from node j with VAP field set to its own id, node i must become a backbone node and turn into black. Table-4.1 presents the pseudo-code of the backbone selection process. Each white node runs the algorithm asynchronously. When a node decides to be black or green, it no longer stays in the selection phase and starts immediately mesh formation and maintenance steps. Thus, while portions of the network are still in the selection phase, the rest can be in the mesh formation and maintenance steps.

The following lemmas show that all nodes decide in a finite amount of time, we end up with a dominating set, and each VAP node has other VAP nodes within 3-hop distance if the network radius is large enough.

Lemma 6. *Given a subgraph G^s of white nodes, $nlff_{th}$, and $NLFF_i$ for all $i \in G^s$ at time instant t, there exists at least one white node which identifies itself as the backbone node if it decides at time t.*

Proof. We can further partition G^s into two sub-graphs G^s_1 where all the nodes satisfy the $nlff_{th}$ constraint and G^s_2 where none of the nodes satisfies the $nlff_{th}$ constraint. If

G_1^s is non-empty then lexicographically the white nodes can be strictly ordered and highest ordered node in G_1^s identifies itself as a backbone node. Otherwise $G_2^s = G^s$ and again lexicographically there is a unique node (but this time among all white nodes) which identifies itself as a backbone node. □

Lemma 7. *[Time-boundedness and Correctness of Selection Algorithm]: Initial selection part of BBM terminates in a finite amount of time and the set of black nodes constitutes a dominating set under the assumptions that (1) the network graph is connected and the network size is bounded but larger than one, (2) hello beacons are transmitted error-free with zero delay, (3) hello beacons are transmitted at a much higher rate than the rate of topology changes, (4) nodes use the cached parameters to decide, (5) decided portions maintain the dominating set feature.*

Proof. Without loss of generality, suppose that at time t_0, there is a connected subgraph G^s where any node $i \in G^s$ is white, all nodes have already waited for time T_w, and $1 < |G^s| < \infty$. If we show that at least one node in G^s decides to be black in a finite amount of time τ, then $|G^s|$ becomes a monotonically decreasing function when observed at time instants $t_0 + k \times \tau$ until $|G^s|$ becomes one. But at this moment, we have an isolated white node which selects another green node as a VAP node and it turns into green. Since all the white nodes are exhausted, we end up with a set of black nodes that constitutes a dominating set.

That all nodes have finished initial waiting time of T_w, they are now in the extended waiting time period. Suppose a topology change occurred at $t_1 \geq t_0$ and no white node decided to be black yet. Then, the assumption about the topology changes and hello beacons dictates that in $[t_1 + m_1T_h, t_1 + m_2T_h)$ all nodes have stationary NITs where m_1 and m_2 are integer constants. Assumption (4) ensures a consistent set of parameters among neighbors. By lemma-6 and letting extended waiting time to be αT_h, at any time in $[t_1 + m_1T_h, t_1 + (m_1 + \alpha)T_h)$ where $\alpha < m_2$, there exists at least one white node that decides to be black. Thus, in the worst case, choosing $\tau = t_1 + (m_1 + \alpha)T_h - t_0$ suffices to complete the proof. □

Lemma 8. *Assuming that network graph is connected and the maximum distance in that graph (i.e. network radius) is greater than or equal to 3 hops, there exists a black node i for each black node j such that $i \neq j$ and $i \in N_j^{(3)}$ after the selection part of BBM.*

Proof. We will prove the lemma by using the way of contradiction. Let's assume that there is not any black node $i \in N_j^{(3)}$. Because of the assumptions of the lemma, there is a green node k which is 2 hops away from node j. Since there is no black node in 3 hops of node j, green node k cannot have black neighbors, either. This is a contradiction to the definition of a green node. Hence, the lemma follows. □

Lemmas 7 and 8 provide the necessary framework for completing the backbone formation. If each VAP node discovers the paths to other VAP nodes within 3 hops (i.e. *VAP neighbors*), then we obtain a mesh structure, which will later be utilized in distributing control messages between VAP nodes.

4.3.1.2 Mesh Formation

Hello beacons convey enough information for finding paths between VAP neighbors. We will refer to these paths as *virtual links*. A virtual link can be 2-hop or 3-hop long, i.e.

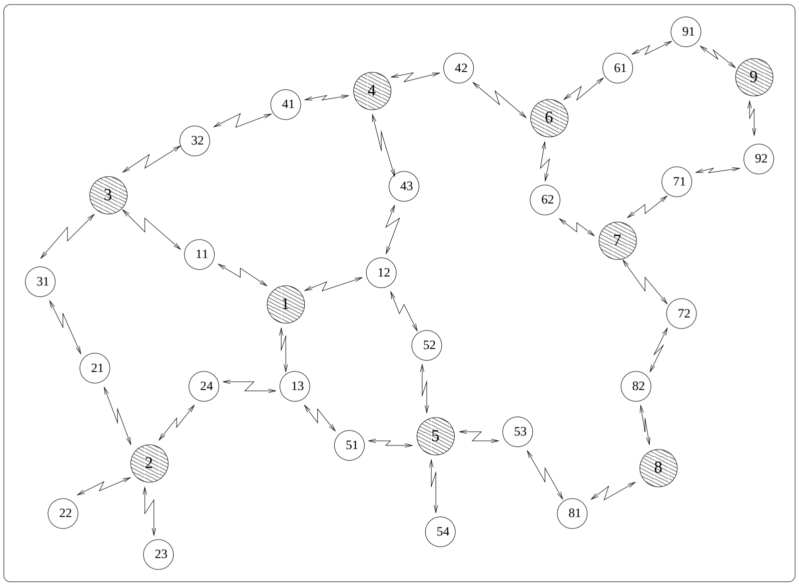


Figure 4.1: An instance of virtual backbone formation.

there may be one or two green nodes between VAP neighbors respectively. Both situations are outlined in figure 4.1. In the figure, dashed nodes are our black nodes, and non-dashed smaller size nodes represent the green nodes. Black nodes are identified by single digit numbers, whereas green nodes are identified by two-digit numbers with most significant digit indicating their VAP node.

For instance of a 2-hop long virtual link, we can look at the case where green node 11 lies between black nodes 1 and 3. Node 3 sees in the hello beacons of node 11 that node 11 has a VAP node other than itself. Since node 11 also sees that node 3 is black and it is not its own VAP node, it includes this routing information in its hello beacons. Thus, both black nodes 1 and 3 have the information that they can reach to each other via node 11 and they update their routing tables.

When there exist two green nodes along the virtual link (i.e. 3-hop long virtual link), the situation is slightly different. For example, we have two green nodes between black nodes 1 and 2. Nodes 13 and 24 recognize from each other's hello beacons that they have different VAP nodes. Therefore, node 13 caches node 24 as the next hop for node 2 and node 24 caches node 13 as the next hop for node 1. They also include this routing information as an extension in their hello beacons so that nodes 1 and 2 will know that nodes 13 and 24 are next hop nodes respectively toward each other. Hence, green nodes play the major role in discovering the virtual links between VAP neighbors. To reduce the size of hello beacons, routing extensions that carry no new information are avoided for a time-out period.

4.3.1.3 Backbone Maintenance

The maintenance of the dominating set feature of the backbone is a very important task against frequent topology changes. BBM phase gracefully handles three events that may happen mainly due to the node mobility: (i) Green node loses its VAP, (ii) black node is deserted by its green nodes, and (iii) black nodes become overpopulated.

The first situation is resolved by forcing new nodes to join the backbone. If green nodes do not receive hello messages for a time period from their VAP nodes, they choose another neighbor as their new VAP node by giving strict priority to black nodes and then to green nodes that comply with the stability constraint and the highest degree criterion. Thus, no node is left without a VAP node.

On the other hand, in the second situation, deserted black nodes autonomously decide to leave the backbone. Desertation may happen either because a black node may migrate to a location where none of the green nodes has this node as a VAP node or because all green nodes associated with the same black node may move out of range or have failed to communicate. Therefore, upon detecting that it is deserted (i.e. when no hello message indicating itself as a VAP node has been received for a time period T_l) a black node must turn into a green node and follow the same actions that a green node take when it is left without a VAP node.

In the third situation, black nodes can be grouped together in the same locality depending on their mobility patterns. To resolve such cases, when a black node i notices other black nodes in its 1-hop neighborhood, it transmits its hello message with a flag indicating that it will change its color to green. When green neighbors, which have node i as their VAP node, receive node i's hello message, they compute the best black neighbor from their own NIT again using stability constraint and highest degree criterion. If the best black neighbor is not node i, they simply assign the best black neighbor as their new VAP node. Otherwise, they set the flag in their own hello messages indicating i as the best node. As long as black node i receives hello messages from its green neighbors indicating itself as the best VAP node, node i remains black. If no such messages are received for a time period T_s, node i turns into green and leaves the backbone.

4.3.2 Distributed Service Discovery (DSD) Phase

Now, we have the virtual backbone and DAs are co-located with the VAP nodes. However, we still need mechanisms to let servers register their services with one or more DAs and clients request the services. This is done in the following fashion. When a server located on node i wants to register its service, it has to register with the DA located on VAP_i assuming node i is a non-backbone node. VAP_i is referred to as *source VAP node*. If the node i is already a black node, then the service is registered with the DA on the same node and node i itself becomes the source VAP node. Server may register with more DAs (even maybe with all DAs). We then need a multicast or broadcast mechanism to distribute the registration messages to other DAs located on other VAP nodes. Any time the VAP node of a server changes, it must renew its registration with the DA operating on the new VAP node. Also, the server should be able to keep the scope of its registration messages local by bounding the number of black nodes that the registration messages could traverse. Similar arguments hold true for the service requests. When a client on node j requests for a service, node j forwards the request to VAP_j provided node j is not already a black node and VAP_j passes the request to the collocated DA. If node j is black, then the request is passed to the DA on node j. In case DAs do not have any fresh registration for the service, the service should be requested from other DAs again by multicasting or broadcasting.

Wireless bandwidth is scarce because of the shared medium and the wireless channel impairments. Although backbone itself helps reduce the overhead in disseminating broadcast or multicast messages by using simple mechanisms like flooding the backbone, it is not sufficient when we consider the increasing frequency of multicast events and the topologies where backbone with virtual links exhibits lots of loops and high average degree. To make things simple, scalable and efficient, we propose a source-based multicast tree algorithm that is triggered by service discovery requests and registration messages sent to the backbone management layer by clients and servers.

In our algorithm, every backbone node keeps a forwarding list among their VAP neighbors for each multicast tree uniquely identified by the source VAP node. As the initial condition, forwarding lists include all of the VAP neighbors except for the source VAP node. Multicast messages contain the following fields: {*source node, source VAP node, sequence number, last-hop VAP node, next-hop VAP node, next hop node, time-to-live (TTL), options, payload data*}. *Source node* indicates the client or the server which initiated the request or registration process. Each multicast message is uniquely identified by the 2-tuple {*source VAP node, sequence number*}. Multicast messages flow from last-hop VAP node to next-hop VAP node. Last hop VAP node and the green nodes along the virtual link compute the *next hop node* from their routing tables using *next-hop VAP node* as the destination point. Note that these routing tables are generated and updated by BBM phase. When *next-hop VAP node* receives the message, it prunes the *last-hop VAP node* from the forwarding list of the particular tree. If the message is received for the first time, the replicas are sent to each node in the forwarding list. When a duplicate multicast message is received from a pruned VAP node to which the same message has not been sent, an explicit PRUNE message must be sent to that VAP node to force it to prune the same link. This algorithm guarantees a multicast tree after a convergence time, given that topology changes slower than the convergence time. In the worst case situation, multicasting to backbone nodes becomes the same as flooding among the backbone nodes. Options field basically defines the type of the encapsulated payload data, e.g. service registration, service request, etc. Depending on the options field, a VAP node waits for a feedback from upper layer protocols, where the payload

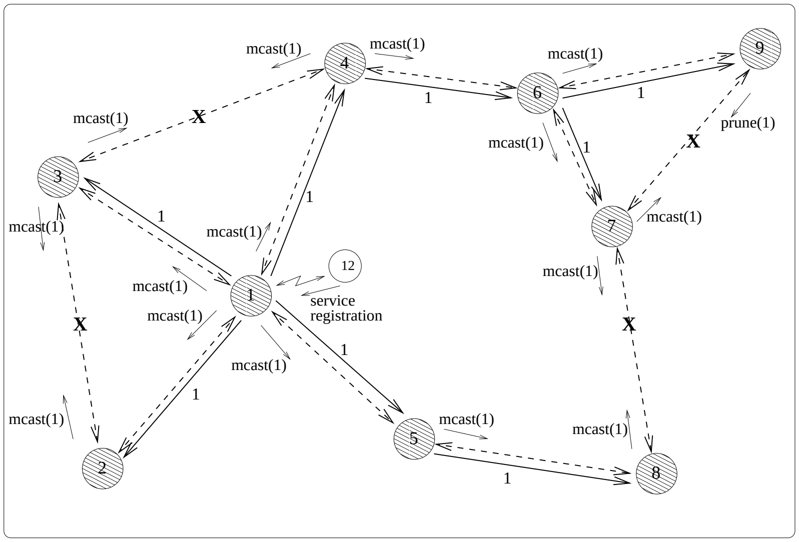


Figure 4.2: Source based multicast tree formation on virtual backbone with service registration message.

data is handled, in order to stop or proceed with the forwarding operation. TTL field can be used to further limit the depth of forwarding for a particular multicast message (i.e. the information is explicitly kept local).

Forwarding list members basically can be understood as *child nodes*, and the VAP node from which a multicast message is received for the first time is then designated as the *parent node*. When a VAP node loses its parent node, it sends an UNPRUNE message to its VAP neighbors. Upon receiving an UNPRUNE message from their parent, child nodes generate their own UNPRUNE message and send it to other VAP neighbors except for their parent node. All VAP nodes, which receive an UNPRUNE message, add the sender to the forwarding list of the particular multicast tree as specified in the UNPRUNE message.

To give more insights about the algorithm, we provide two examples in figure 4.2 and figure 4.3. These figures essentially show the same topology as in figure 4.1 except for the fact that the links and green nodes between VAP nodes are replaced by bidirectional dashed lines to denote the virtual links. Suppose server on node 12 wants to register its service to all VAP nodes (nodes 1 to 9 in the figure). Node 12 sends the registration message to its VAP node 1. Then node 1 initiates multicasting process by first unicasting the copies of the registration message to its VAP neighbors. Nodes 2, 3, 4, and 5 receive a multicast message originated by node 1 for the first time. So, they forward the copies of the message to their own VAP neighbors except for the node from which that message is received and the originator of the message. In the figure, node 2 receives the same multicast message from node 3 as it received from node 1. Thus node 2 knows at that moment that node 3 has already received the same multicast message. As a result node 2 stops forwarding multicast messages originated from node 1 to node 3. Similarly, node 3 sees duplicates via node 2 and 4, so it stops forwarding multicast messages originated from node 1 to these nodes. A different case happens at node 9. Node 9 first receives the multicast message via 6 and then via 7 at almost the same time. The message from node 7 is duplicate, and since node 7 is pruned from the forwarding list of node 9, node 9 explicitly sends a PRUNE message to node 7. Hence, node 7 stops forwarding multicast messages originated from node 1. Explicit PRUNE mechanism is also used when duplicates are received persistently from a pruned node. This can happen in cases where multicast or PRUNE messages are lost. At the end a multicast tree is formed on top of the virtual backbone. The source VAP node, i.e. node 1, becomes the root of this multicast tree. In figure 4.2, directed solid lines labelled with node 1 represents this tree. Dashed lines with a cross represent the pruned links. If node 13 also

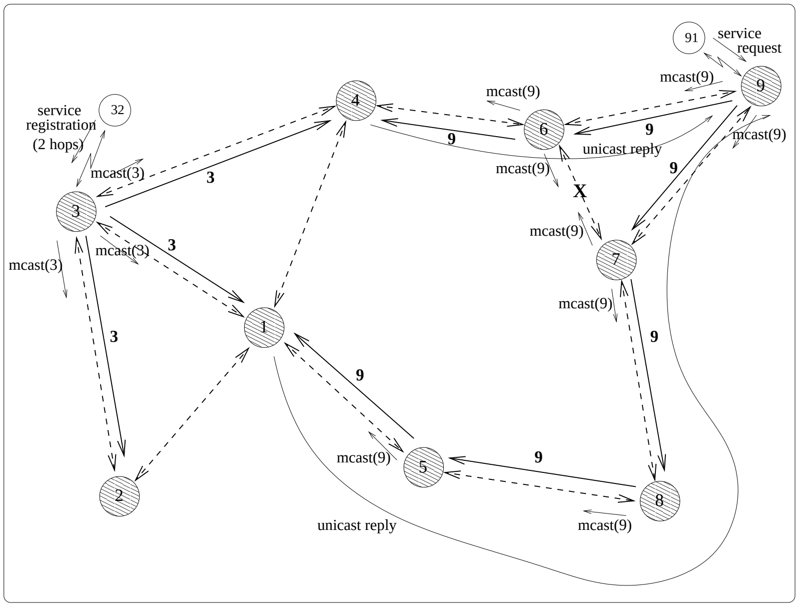


Figure 4.3: General view of service discovery with TTL-limited registration and request messages.

wants to initiate a service registration or request, since it has the same VAP node as 12, it uses the same multicast tree represented by source VAP node 1. Note that the multicast tree can be logically viewed as two separate trees rooted at nodes 12 and 13.

Figure 4.3 shows a general scenario when TTL field is set to 2 hops for multicasting service registrations from node 32. TTL field includes the link from node 32 to its VAP node 3. The solid one directional links show the multicast tree and the numbers on them indicate the root of the tree. Suppose that this service is only provided by node 32 and that node 91 wants to utilize it. In that case, node 91 sends a service request message to its VAP node 9 and the request is multicasted until it reaches nodes 1 and 4, which already have the information. Therefore, nodes 1 and 4 discontinue forwarding the request message and using reverse link information, they reply back to node 9; node 9 then replies back to node 91. Here, it is interesting to note the interaction between service discovery agent and multicasting in forwarding decisions. Unless the query is resolved, a service request is propagated further.

4.4 Simulation Environment

In this section, we describe the performance metrics that are used to evaluate different architectural and network support choices. We then present the simulation framework and the simulation results.

4.4.1 Performance Metrics

Three performance criteria are considered in our simulations. The first performance metric is the total mean *control message overhead* of each service discovery mechanism which measures the load of the algorithms on network resources in terms of the number of packets.

The second performance metric is the mean *hit ratio* of these mechanisms. In the generic service discovery algorithm used in our simulations, a client does not repeat the request until it receives a successful reply. This is simply because we want to see how many original requests are successfully replied and we label these requests as successful attempts. Hit ratio is simply the ratio of the total number of successful attempts to the total number of requests. When hit ratio and control message overhead are combined together, it reflects the efficiency of each approach.

Our last performance metric is the *average time delay* between the time any successful request sent from a client and the time corresponding reply received by the same client. This metric measures the promptness of the service discovery and it is particularly important when we have real time applications waiting for timely response for each service query.

4.4.2 Simulation Model

We simulate four different service discovery mechanisms using ns-2 with CMU wireless extensions [61]. One of these mechanisms is our proposal for the directory architecture, and other three mechanisms are based on directoryless architectures. As network support, we consider multicasting and anycasting as two major contenders for the directoryless architecture.

There exists a rich literature on multicasting for mobile ad hoc networks [73, 74, 75, 76, 77, 78, 79]. In our scenarios, we consider the general case of multiple senders (i.e. clients) and multiple receivers (i.e. servers). We do not include the case, where servers advertise their services and clients learn about the services passively. Instead, we focus only on the request-reply and registration mechanisms for they cover the majority of the applications. Since clients issue service requests at will, the multicast protocol should be sender based rather than receiver based. We also want to have a multicast protocol that does not depend on any particular unicast routing protocol, which restricts its use. Given these choices, *on demand multicast routing protocol (ODMRP)* [73] quite well satisfies the features that we seek for. We have implemented ODMRP in ns-2 without using its mobility adaptive part, which requires GPS receivers on mobile devices. In this implemented version, when a sender initiates a multicast session for the first time, it floods the network with JoinData control messages. JoinData messages generate reverse path information as next hop to the multicast sender. As JoinData messages reach to receivers, each receiver forms a JoinTable message which includes a tabulation of multicast senders and the corresponding next hop nodes towards senders. Each intermediate node checks if it is a next hop node for any multicast sender. In that case, it joins to the forwarding group for that particular multicast group and re-transmits JoinTable message after filtering out the portions of the tabulation for which the intermediate node is not a next hop node. When a multicast data packet is received, the nodes, which are in the forwarding group for the particular multicast address, locally re-broadcast the data packet. To maintain the routing (or forwarding group) entries, each multicast sender periodically floods the network with JoinData messages as long as the multicast session continues. We set the time-out period for routing entries as 10 seconds and for JoinData refresh messages from senders to receivers as 3 seconds.

We have implemented anycasting by modifying the existing ns-2 code of the two very popular ad hoc unicast routing protocols, namely DSR [80] and AODV [77]. Specifically, we have defined a virtual server node that is uniquely identified by the IP anycast address, for which only the actual server nodes have routing entries [70]. We refer to these two modified algorithms as *anycast-DSR* and *anycast-AODV* respectively. The choice of these protocols also stems from the fact that they are reactive algorithms and quite efficient in terms of control message overhead. Both of them support similar mechanisms like flooding route request messages and obtaining replies. Nevertheless, in addition to their differences in creating, caching, and maintaining the routing entries, these protocols mainly diverge in the way they place the routing decisions: In DSR, data packets themselves carry all the necessary routing information whereas in AODV the intermediate nodes use their own routing tables to determine the next hop to forward the data packets.

For our proposal, BBM and DSD are implemented below the routing and above the link layer with direct interfaces with the service discovery protocol again in ns-2. Initially, the TTL field for service registration messages is kept fixed at 1 so that each server registers with only one DA. Later on, the impact of using different TTL values is also examined. We refer to the overall proposal as *distributed service discovery protocol (DSDP)*.

We use the distributed coordination function (DCF) of IEEE 802.11 as the underlying MAC protocol. DCF in IEEE 802.11 is a random access scheme and belongs to the CSMA/CA family. The radio interface is based on Lucent's WaveLan technology with 250 meters of nominal propagation range and 2 Mbps of nominal bit rate. Radios use omni-directional antennas and we assume a two-ray ground propagation model. To compare the directoryless and directory solutions, the network size is fixed to 50 nodes. Both square (1000mx1000m) and rectangular (1500mx300m) topologies are considered, but since the relative results are very similar to each other, we only provide the results for the rectangular topology.

We use two different mobility models: pure random-way point model and mixed random-way point model. In pure random-way point model, which is the most widely used one in the literature, nodes select a random destination point and a speed value from a uniform distribution $U[0, V_{max}]$ after a pause time P. When they reach the destination, they repeat the same process. A higher V_{max} or a lower P corresponds to a higher mobility level. Five different mobility patterns are used for each $\{V_{max}, P\}$ pair. In the experiments, where V_{max} is varied, P value is kept at 0 sec., i.e. nodes are always in motion. When we vary the P value, we had fixed the V_{max} value at 20 m/sec. On the other hand, in mixed random-way point, we randomly selected a subset of the nodes[6] stationary and apply random-way point model for the rest of the nodes. Again, since the relative results turn out to be same as a function of average mobility level, we do not present our results that are obtained by altering pause times and number of stationary nodes in the thesis.

Apart from mobility, the other crucial scenario parameters that we can analyze are the number of clients and the number of servers in the system. The number of clients is selected as 10, 20, and 30, whereas the number of servers is varied between 1, 3, and 5. For each mobility pattern, again five different random set of clients and servers are used. Thus, each point in the simulation plots is averaged over 25 random scenarios. Each server periodically registers its service every 10 seconds and whenever its VAP node changes for the directory architecture. Without any loss of generality, we assumed that only one service class is offered by all the servers in the network. Clients, on the other hand, send their requests such that the inter-arrival time is a random process $\zeta = T_0 + \tau$ where T_0 is deterministic time set to 6 seconds and τ is exponential random variable

[6] In the batches of 10,20,30, and 40 nodes.

with mean 2 seconds.

Clearly, TTL variable may have a significant impact on the performance of DSDP. We observe the effect of TTL on system performance by testing it against different mean request inter-arrival times (e.g. 4 and 8 seconds) and different topologies (e.g. 1500mx300m with 50 nodes vs. 2400mx1500 with 100 nodes). To this end, we have altered the TTL value between 1 and 3 in these experiments.

The main results of these extensive simulations are presented in the sequel.

4.4.3 Simulation Results

In the first set of experiments, we intend to capture the effect of the number of servers as the number of users is kept constant. Control message overheads of on-demand anycast protocols are found to be very sensitive to the number of servers[7]. ODMRP tends to have more overhead while other protocols have a lower overhead as the number of servers increases. This is not an unusual outcome when we consider the main mechanisms of these approaches. In multicasting, all the servers receive the requests, and then all of them have to reply back. But, in anycasting only one of the servers receives the message regardless of the number of servers. Since it is highly likely that the closer server replies back and the average shortest distance between clients and servers gets smaller with more servers in the network, the control message overhead declines. Higher number of servers also narrows down the depth of the query trees in DSDP, leading to a slight reduction in the control overhead.

[7] In the plots, arrow directions indicate the increasing number of servers or users.

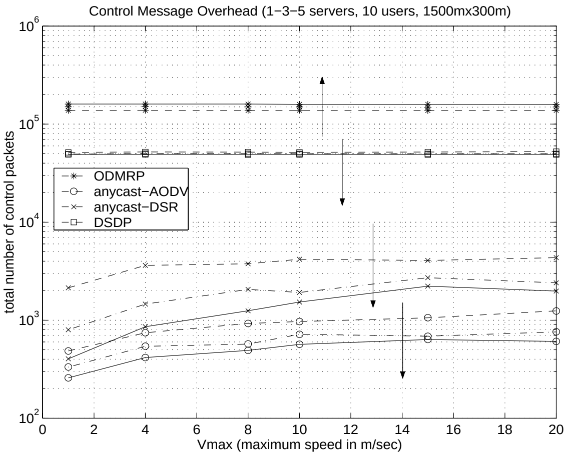


Figure 4.4: Control message overhead for 10-user case.

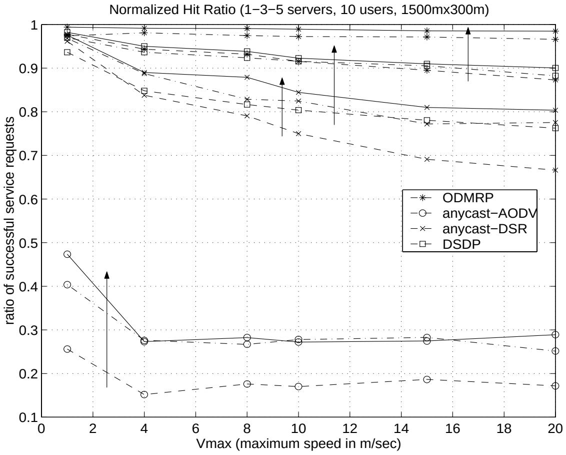


Figure 4.5: Ratio of successful requests for 10-user case.

End-to-end delays for successful service discovery ameliorate with the increasing number of servers. This is expected again due to the fact that the average distance between users and servers decreases as the number of servers increases. Anycast-DSR and ODMRP show more rapid improvements, whereas anycast-AODV and DSDP offer consistently lower delays. Although delay performance of ODMRP reaches to that of anycast-AODV and DSDP, anycast-DSR cannot compete in terms of delay.

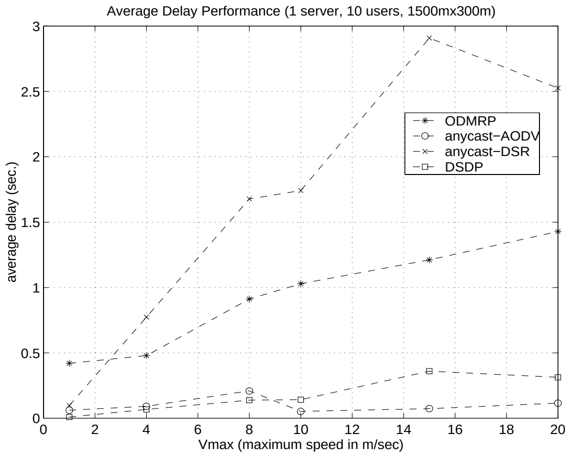


Figure 4.6: Average delay comparison for 1-server/10-user case.

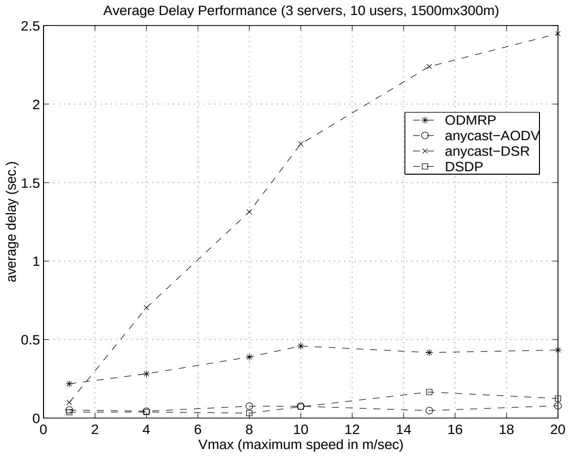


Figure 4.7: Average delay comparison for 3-server/10-user case.

The hit ratio also improves with the number of servers. Anycast-AODV performs inferior compared to other protocols, whereas ODMRP has consistently the best hit ratio. DSDP outperforms anycast-DSR more significantly as the mobility level and/or the number of servers increase. Note that all performance metrics become worse as the mobility in the network increases, because the link failures occur more often.

The second set of simulations addresses the question of how increased load in terms

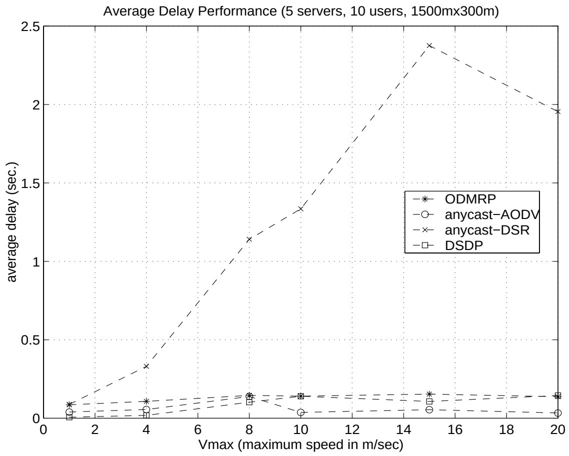


Figure 4.8: Average delay comparison for 5-server/10-user case.

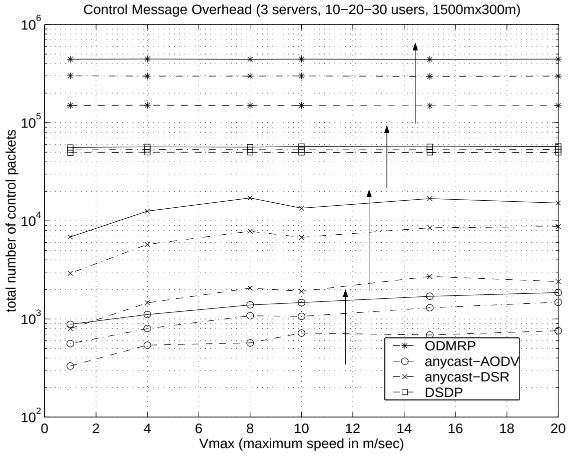


Figure 4.9: Control message overhead for 3-server case.

of number of users affects the performance of the protocols. The results are plotted in figures 4.7, 4.9, 4.10, 4.11, and 4.12. The number of servers are kept at three and the number of users are varied between 10 and 30.

The overhead of each protocol increases with the number of users with DSDP being the least sensitive one. Although ODMRP is a heavy-weight protocol, it is quite sensitive to the increased number of users rather than the servers. This is a direct consequence of

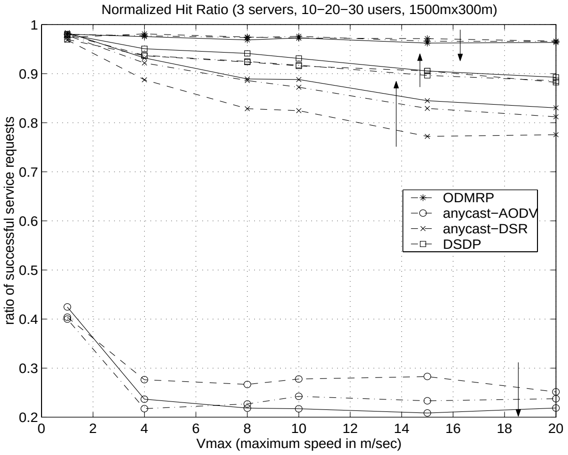


Figure 4.10: Ratio of successful requests for 3-server case.

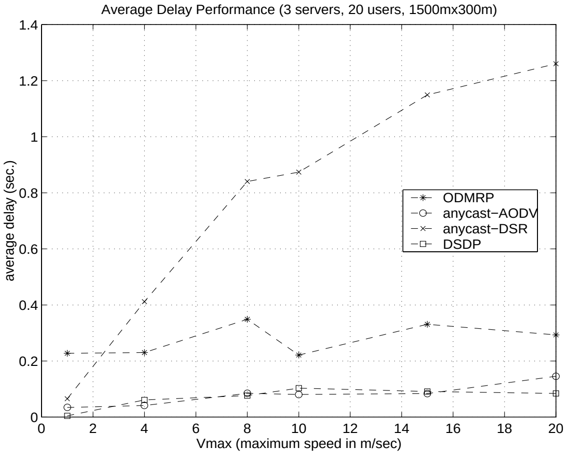


Figure 4.11: Average delay comparison for 3-server/20-user case.

the facts that ODMRP is a sender-based multicast scheme and the senders broadcast periodic refresh messages to maintain the multicast group.

The delay of each protocol tends to decrease as the number of users increases. At first sight, this can be regarded as a counter-intuitive result; because an increase in the number of users creates a higher load on the network that may result in congestion and packet collisions, and may consequently deteriorate the delay values. On the other hand,

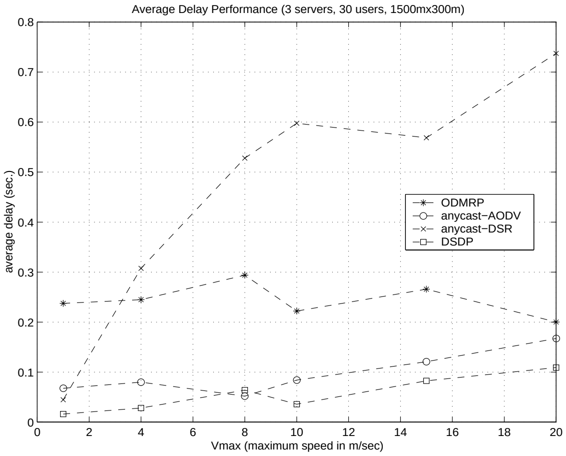


Figure 4.12: Average delay comparison for 3-server/30-user case.

increasing the number of users can have the effect of reducing the average distance from the servers, and can thereby enhance the end-to-end delays. It also helps on-demand anycast algorithms to discover the routes in advance when users share more common links. For networks operating below capacity, these positive effects suffice to obtain better delay values.

ODMRP and DSDP do not show much response to the changes in the number of users in terms of successful service requests. Anycast-DSR shows significant improvements in higher mobility cases whereas anycast-AODV suffers from further performance loss.

We further run a third set of simulations to understand how sensitive DSDP is against the TTL parameter for different request rates and network topology. Figures 4.13, 4.15, and 4.17 show the effect of TTL field on a topology of 1500mx300m with 50 nodes at a mean request inter-arrival time per user of 4 and 8 seconds. On the other hand, figures 4.14, 4.16, and 4.18 present the impact of TTL on a topology of 2400mx1500m with 100 nodes at a mean request inter-arrival time of 4 seconds. Unlike in the previous plots, the overhead of hello beacons is excluded from the total overhead, because it is a fixed cost solely determined by the network size and T_h.

In figure 4.13, the top three plots correspond to request inter-arrival time of 4 seconds and the bottom three correspond to inter-arrival time of 8 seconds. Evidently, more frequent requests boost up the overhead. We also find out that higher service request rates may significantly benefit from using higher TTL values in terms of the control message overhead. Higher values in TTL field increase the depth that the service registrations reach along the backbone[8]. In turn, the depth that service requests traverse along the backbone reduces. Therefore, if the total number of service registration messages is less than the total number of service requests, higher TTL value is expected to reduce the overhead until TTL value saturates. Saturation point occurs when the registration trees rooted at different servers cover the whole backbone so that any further increment in TTL field no more affects the depth of the request trees, which becomes one-hop from clients to their own DA nodes. For request inter-arrival time of 4 seconds, this saturation point is observed when TTL equals to 2. The number of service registrations is directly proportional to the handoff rate of servers between different DAs and the number of servers, whereas it is inversely proportional to the service registration lifetime. Mobility increases the handoff rate and hence the advertisement rate. The interesting point is that mobility also multiplies the number of DAs, to which the same server registers, as the server handoffs. Thus, the depth of request trees again reduces.

[8] This is true provided that the network diameter is large enough.

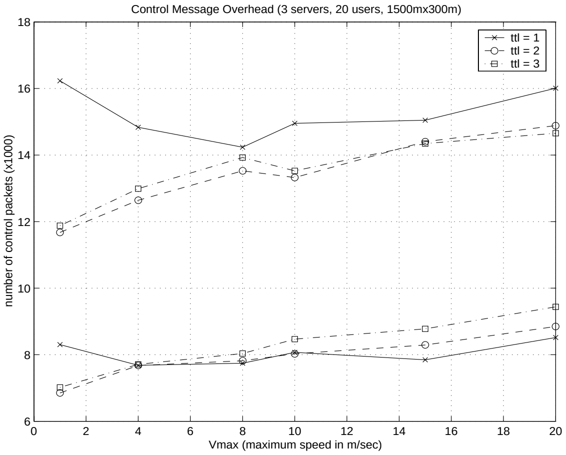


Figure 4.13: Control message overhead for different TTL values with mean request inter-arrival times of 4 and 8 seconds.

When mobility level becomes too high, the stability of the backbone collapses. Accordingly, the life-time of staying as a DA node gets shorter and this makes the previous registrations stale in a premature way. All these observations are well reflected in our simulation results. In this respect, higher TTL fields do not improve overhead for request inter-arrival time of 8 seconds especially for higher mobility levels. The *saturation* TTL value of 2 provides a significant reduction in the overhead; as mobility increases, the performance gap against the TTL value of 1 first closes down, but after a certain point, it gradually gets wider. For the larger network topology, the gain becomes even more significant (see figure 4.14).

For the success rate and delay performance, we see similar trends with the control message overhead. In figures 4.15 and 4.17, better performance curves (i.e. higher hit ratio and smaller delays) for the same TTL value correspond to the mean inter-arrival time of 4 seconds. Increasing TTL value until the saturation point improves both the hit ratio and the delay values, because it curtails the distance between the clients and DA

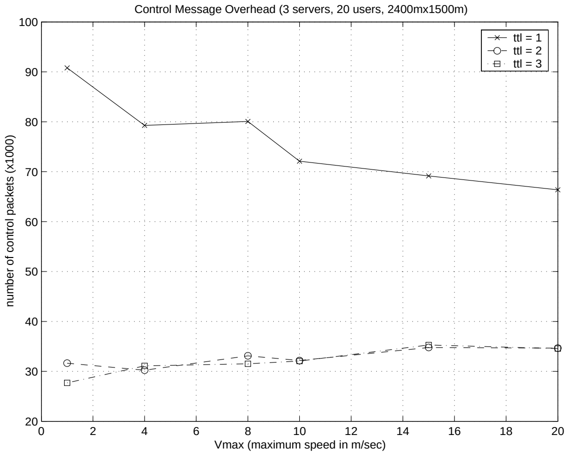


Figure 4.14: Control message overhead for different TTL values with 100 mobile nodes and mean request inter-arrival time of 4 seconds.

nodes that have a fresh service registration. After the saturation point, this distance remains the same, but the overhead increases until TTL becomes equal to the backbone diameter. Higher overhead increases the probability of collisions that causes higher transmission delays and packet losses. Again, larger topology shows significant improvements over TTL value of 1 with diminishing returns as TTL gets larger as seen in figures 4.16 and 4.18. The saturation value of TTL for 2400mx1500m turns out to be 3 hops.

We can summarize our simulation results as follows: 1) Under various mobility scenarios with different number of users and servers, the relative performances of the protocols remain the same in general for a fixed number of nodes and persistent service requests. 2) In terms of overhead, ODMRP is the most heavy-weight protocol and on-demand anycast protocols are the most light-weight. An increase in the number of users almost linearly affects the message overhead of ODMRP as a result of its sender-driven multicasting feature. On the other hand, DSDP is not as sensitive as the

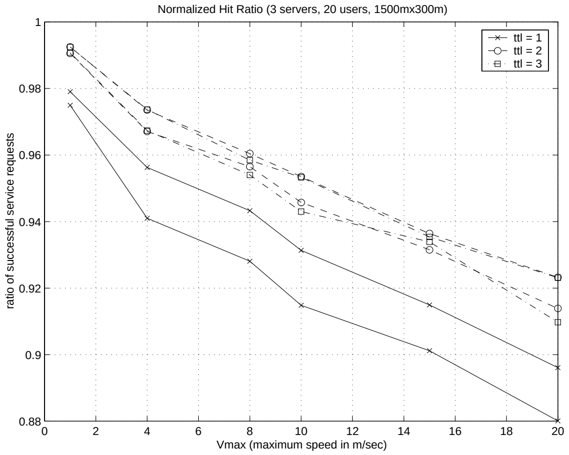


Figure 4.15: Ratio of successful requests for different TTL values with mean request inter-arrival times of 4 and 8 seconds.

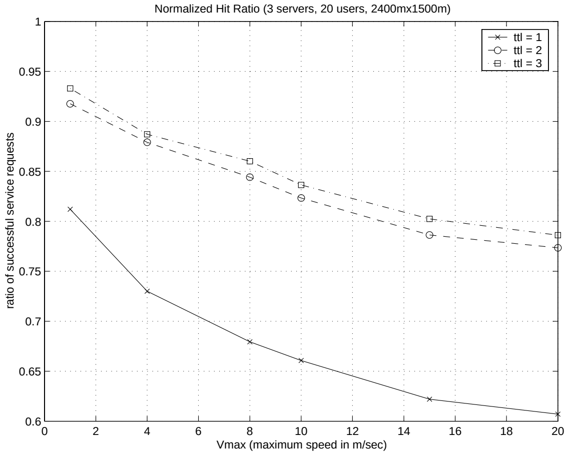


Figure 4.16: Ratio of successful requests for different TTL values with 100 mobile nodes and mean request inter-arrival time of 4 seconds.

other protocols against mobility, number of servers or number of clients, since the bulk of its message overhead is generated by periodically transmitted hello beacons. In our simulations, each node transmits a hello beacon every 1 second. Considering that we run simulations with 50 nodes and for 900 seconds, the overall cost of these beacons amounts to 45,000 packets, i.e. 75% to 94% of the overall overhead. 3) In terms of hit ratio, anycast-AODV performs poorly except for almost stationary scenarios. It cannot

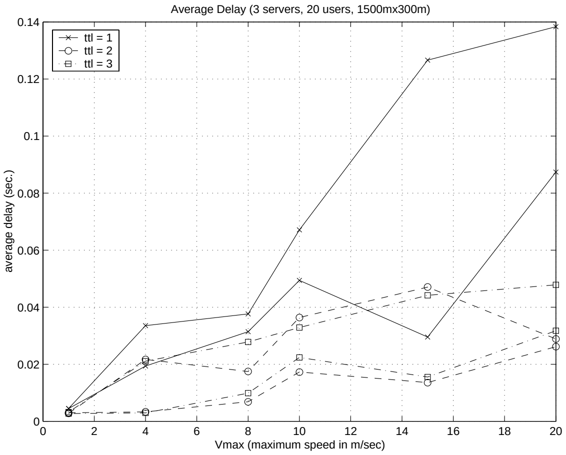


Figure 4.17: Average delay for different TTL values with mean request inter-arrival time of 4 and 8 seconds.

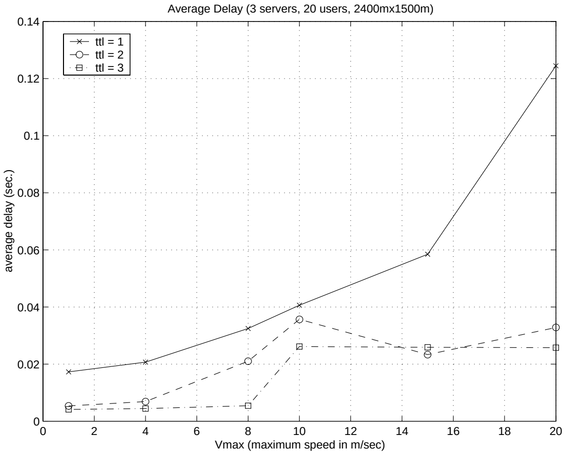


Figure 4.18: Average delay for different TTL values with 100 mobile nodes and mean request inter-arrival time of 4 seconds.

therefore be a good candidate in general even though its delay and overhead performances are much better than ODMRP and anycast-DSR. ODMRP has consistently the best hit ratio performance as much as 18% above the next best protocol DSDP at very mobile environments. Anycast-DSR catches ODMRP and DSDP for low mobility cases, but the performance difference goes up to more than 10% with second best DSDP.

4) Delay performance of DSDP is consistently better than other choices. This is due to

the fact that lower numbers of nodes are involved in search queries and that registration of services shortens the average distance in number of hops. 5) Performance of DSDP proves to be sensitive to the choice of TTL field in the registration messages. When higher request rate and network size or sparser topologies are used, a TTL field of more than one improves performance values in all three metrics. Depending on the number of servers and network diameter, results indicate that there is an optimum TTL value for control message overhead. After this optimum value, the gains in terms of delay and successful service requests experience a negative drift, which eventually leads to a decline in the performance values. Limited mobility is observed to help in disseminating the server information to a larger set of backbone nodes which limits the average depth that the request messages are propagated. However, mobility artificially increases the rate of service registrations. Since higher TTL values increase the depth that the service registrations are propagated, mobility negatively impacts the overhead. The net effect turns out to be in favor of the control message overhead for mobility levels and TTL values below a certain threshold.

Reducing the signaling overhead of DSDP is possible by integrating its control messages with or opening its signaling support to other layers. Next, we discuss about the two strong candidates to demonstrate how this can be actually performed.

4.5 Integration of DSDP Solutions with Other Layers

We see two important collaboration opportunities between our service discovery proposal and the other proposals in: (i) Reliable Server Pooling and (ii) Routing.

4.5.1 Application of DSDP to Reliable Server Pooling

Although we have developed DSDP for implementing a directory based service discovery for wireless networks, it is also capable of providing the core mechanisms for realizing *Reliable Server Pooling* (RSP) [81] in mobile ad hoc networks. RSP addresses the reliability issue by introducing redundancy in the number of servers available to a client. It also provides abstraction of all the functionally equivalent servers, whereby the client can access these servers as a single entity, termed *server pool*. In the RSP, the *Name Servers* (NSs) are responsible for maintaining server pools, load balancing, and server discovery. The client resolves the mapping from a server-pool handle to the addresses of servers registered in this pool by querying its *Primary Name Server (PNS)*. Under this scheme, whenever a server fails, clients that utilize that server should transparently switch over to another server in the pool.

Traditional methods such as IETF RSerPool proposals are shown to be inapplicable to very dynamic networking environments [82, 83]. However, this is not an unexpected result, because the original architecture is designed for fixed networks, where connectivity of the network topology is rarely disrupted. On the other hand, the entities such as name servers and primary name servers in RSP are exact equivalents of the directory agents and virtual access points in DSDP. Hence, in principle, we should be able to utilize the architectural and signaling solutions of this chapter to effectively build a survivable server pooling in wireless networks, especially in tactical network environments, after the following modifications:

- From the DSDP point of view, the goal is to discover at least one server that matches the service query. If there exists other servers that are located behind a particular DA that has a single server match, since the query is interrupted at that point, only one server is discovered. This reduces the overhead, but in RSP clients generally request a minimum number of matching servers, which is typically more than one. The solution involves defining the number of pool elements, i.e. servers, and placing the already discovered server elements into the query message. Once, a DA observes that the minimum threshold is exceeded already, the query propagation is halted.

- DSDP does not support switch-over functionality. Clients must issue a query and, upon receiving a response, must re-establish a connection with the new server by itself. Nevertheless, this process and switch-over decisions are orthogonal concepts. From the RSP perspective, DSDP should be adapted only as a stub that builds the core architecture and provides signaling support. Just like application layer service discovery agents, an RSP sub-layer above the transport layer must be defined separately to accomplish switch-over functions. Then, applications talk to the RSP sub-layer on the same node and RSP sub-layers on different nodes communicate over the DSDP procedures.

- DSDP does not establish a reliable communication for server registrations, however this is just a small technicality and the registration with the VAP node can be made reliable by requesting receipt acknowledgements.

4.5.2 Integration with Routing Protocols: Case Study with OLSR

DSDP does not need any routing layer support for its functionality. However, it is not devised or proposed as a routing layer alternative and the integration between two layers will be beneficial in suppressing signaling overhead. The first straight-forward application is to share the routing tables that are built by DSDP with the routing agents. However, even stronger integration is possible by embedding the hello beacons into the existing control messages of the routing protocols. We believe that especially link state protocols (e.g. OLSR) or reactive protocols that have their own neighbor discovery procedures (e.g. AODV) have a strong potential in that pursuit. To show an instance of such a signaling integration, we use Optimized Link State Routing (OLSR) protocol (IETF-RFC3626) as our case study in the following.

OLSR is a link state protocol that suppresses the control message flooding by limiting the retransmissions to a limited set of forwarding nodes. Its hello-beacons are transmitted at a period of 2 seconds, and carries the following information:

- Node willingness to relay broadcast messages.
- Link code field that has 8 bits total. However, only the least significant 4 bits are defined to specify whether (1) the neighbor links are symmetric, asymmetric, lost, or unknown; and (2) whether the neighbors are of types symmetric, relay-node, or not-a-neighbor.
- 1-hop neighborhood information (e.g. IP addresses) of the transmitting node.

From the information carried by OLSR beacons, we can infer the following information that can be exploited by the BBM algorithms: (1) degree information, (2) the number of link losses of any neighbor, and (3) relay willingness. Of this information, the relay willingness is currently not used in the backbone-selection phase; however, our decisions are lexicographical and adding such a field do not violate the convergence of the algorithms and it may be useful to filter out the incapable nodes in the selection process. In addition to the information that we can infer from the OLSR beacons, the BBM phase requires the following data: (1) Effective degree (the number of undecided neighbors); (2) Neighbors color; (3) VAP nodes of neighbors; And (4) other backbone nodes in 3-hop neighborhood.

The first three entries can be included in the OLSR hello-beacon headers by extending the semantics of a neighbor type. In addition to the symmetric, relay-node, and not-a-neighbor type, we can introduce a new type as VAP. This new type allows us to identify the interfaces that belong to the VAP (if any). To declare the color of the transmitting node, we can use 2-bits of the most significant four bits of the first link code in the hello-beacon header, which is not used in the existing OLSR implementations. At this stage, the only missing piece of information is the list of VAPs in a backbone node's 3-hop neighborhood. One possibility is to modify the backbone-formation algorithm by letting each VAP advertise itself via an IP packet (with TTL set to 3 hops). OLSR can then effectively limit the overhead of such local flooding. This approach avoids substantial modifications to OLSR. However, a more efficient option is to include 2-hop, VAP-only neighbors in the OLSR hello-beacons. A third -and we believe the best-option is to utilize the information gathered by green nodes. When a green node discovers in its 2-hop neighborhood a VAP node that is different from its own VAP, it can send a unicast message directly to its VAP (instead of a beacon). Since OLSR is link-state based, a failure of any path to virtual neighbors can be detected by checking the routing tables. Thus, unicast messages should only be created when a new NS is detected by a green node. To summarize, we can easily integrate OLSR and virtual backbone through modest modifications to OLSR that (1) exploit the link codes in the hello beacons and (2) make the OLSR hello-beacons visible to DSDP through some file I/O operation. Most modifications have to be made in the DSDP itself for the purpose of information collection and to support associated signaling changes.

4.6 Summary

In this chapter, we have explored the possible architectural and network support choices for service discovery in multi hop wireless and mobile networks. We provided our original service discovery mechanism to support a directory architecture. We also implemented ODMRP, anycast-DSR, and anycast-AODV protocols along with our proposal to compare (i) directory and directoryless architectures and (ii) to compare different network support options, i.e. anycasting vs. multicasting. We used control message overhead, mean success rate and the average delay as three performance metrics for our comparisons. We changed the parameters such as number of clients, number of users, mobility level, topology size, service request rates, and TTL value for a comprehensive analysis.

It is the general idea that since no cost of selecting and maintaining DAs are involved, a directoryless architecture would be the least expensive and easiest to implement in a mobile ad hoc network. This view does not however take into account the operational costs of the lower layer support that is required for such an implementation. Our results demonstrate that if the required network support is multicasting, then maintaining multicast trees can be very expensive in terms of control message overhead. Hence, overall cost of directoryless architecture with multicast support requirement can in fact be more than that of the directory architectures. On the other hand, if anycasting is used as a network support, we can have a very light-weight directoryless service discovery. Nonetheless, this reward comes at the expense of significantly reduced performance in terms of average hit ratio. The level of hit ratio may in fact drop to unacceptable levels as seen in our simulation scenarios with anycast-AODV. Anycast-DSR shows a more competitive level in terms of hit ratio, but then its mean delay values are largely compromised even under mild mobility conditions. These problems are put aside, the main restriction for anycast support arises from the fact that it can only be utilized in a limited number of service classes. Therefore, multicast support displays a more robust, reliable, and general framework for directoryless service discovery architectures.

Results also reveal that directory architecture supported by a virtual backbone structure can perform quite well under various mobility conditions in addition to its inherent advantages, e.g. resource allocation, load balancing, localization, etc. The most dominant figures are observed in the average delay performances. Our proposal consistently has the best delay values and achieves very competitive mean hit ratio values in comparison to the best values obtained by ODMRP in directoryless service discovery. Furthermore, the performance results show relatively little sensitivity against the mobility, the load on the network, and the number of servers. This suggests that DSDP is not only feasible, but also a very good candidate for real-time service discovery scenarios, where a prompt and low jitter response is essential. Although virtual backbone approach is not as light-weight as anycasting solutions in terms of message complexity, when backbone is exploited by multiple stack of higher level protocols and light-weight hello messages are piggybacked behind data packets or other layer control beacons, the overhead of forming and maintaining the backbone can be quite justified. Thus, on contrary to the general view, we demonstrated in this paper that directory architecture is a compelling solution especially for medium to large scale MANETs.

Besides comparing our directory architecture based DSDP protocol with the alternative solutions, we also examined its internal dynamics with respect to the TTL value. Depending on the service advertisement and request rates as well as the number of users, the number of servers, and the backbone size, we showed that further improvements could be achieved by fine-tuning the TTL field. Experiments also support the following intuition: Servers that frequently change their VAP points should suppress their registrations by keeping the TTL value small to keep the service discovery overhead small, since mobility helps in dissemination of fresh service records among multiple DAs.

We also illustrated over reliable server pooling and routing applications that DSDP signaling and architectural mechanisms can be readily integrated with other layer's solutions. This is particularly important in increasing the overhead efficiency of DSDP.

Chapter 5

Conclusion

This thesis concentrated on the network and service level heterogeneity as two crucial pillars of wireless networking. Accordingly, we treated bandwidth resources and services as the main commodities that different network segments can offer to each other as well as to their own users. Chapter 2 and 3 was dedicated to the analysis of network level heterogeneity and what it implies in terms of utilizing the overall bandwidth resources. On the other hand, chapter 4 investigated the fundamentals of locating network services from the architectural and signaling perspectives.

In evaluating the network level heterogeneity, we used a two-tier network architecture, where a transport capacity-limited ad hoc network is overlaid by a resourceful infrastructure. We further refined our model by introducing necessary topology, connectivity, interference, and channel models to specifically address two main-stream scenarios: (i) large scale hybrid network environments, where the traffic pattern and topology observations look more random, and (ii) small scale hybrid network environments, where the traffic pattern and topology can be viewed more arbitrary. Our models contribute to the wireless literature by clearly setting forth the basic guidelines about how to analyze the hybrid network architectures and how to utilize the bandwidth resources of different tiers.

More specifically, in chapter 2, we proved that per node throughput capacity scales with $\Theta(1/logN)$ for large scale hybrid networks with random ingress and egress points between the tiers, while preserving *strong connectivity* of the ad hoc network. This result indicates a $\Theta(\sqrt{N/logN})$-fold improvement over the pure ad hoc network operations. One of the important findings in this chapter also states that strong connectivity condition is the main limiting constraint on the capacity rather than the access point population whenever it scales faster than $\Theta(N/logN)$. We later worked on a *weak connectivity* constraint to take full advantage of the infrastructure support. We provided the mathematical rules to decide on the satisfiability of weak connectivity condition. We also demonstrated that any upper-bound that satisfies the weak connectivity condition can in fact be achieved. Showing the achievability of upper-bounds requires developing a jointly optimum routing and scheduling algorithm in the sense that a tight lower bound can be attained. We used Voronoi tessellation based clustering by appropriately limiting the Voronoi cell areas, which further allowed us to decouple the scheduling and routing decisions from each other. Our major finding is that -unlike the deterministic hybrid networks- a constant throughput per node cannot be obtained asymptotically in a random network scenario. This pessimistic result is compensated by establishing a subsequent result that the decay or per node throughput capacity to zero can be made arbitrarily slow.

In chapter 3, our view of global hybrid network is partitioned into smaller domains that include a locally manageable number of ad hoc nodes. For such small scale finite networks, our basic argument has been that any communication is subject to a quality of service demand and that the objective of the network design must be the suppression of emanated radio interference in each wireless domain. Because the independent consideration of communication layers can provide neither quality of service guarantees nor enough power suppression, we provided a cross-layered framework to address the problem. Our results reveal that water-filling techniques that rely on appropriate metrics perform better in solving the feasibility problem for joint scheduling and power control problem than the top-down designed greedy algorithms that attack the feasibility problem first. The upper-bound expression developed in chapter 3 proves to be such a reliable metric, which eliminates strong interferers evenly across the available channels. On the other hand, algorithms that greedily operate on minimizing the total power at each iteration turn out to be more effective in terms of the objective function. When a degree of freedom introduced in terms of access point (or equivalently path) assignment, a simple mixed strategy, which performs the path, schedule, and power assignment simultaneously, is shown to attain significant improvement over the case where the base station assignments are decided independently. This mixed strategy turns out to perform slightly better in solving the feasibility problem even if we employ pseudo-polynomial techniques that exhaustively search over the feasible base-station assignments and rely on joint schedule&power assignment heuristics. This illustrates us that with some performance sacrifice in terms of total power consumption, we can design efficient algorithms that effectively satisfy the wireless user QoS demands by concurrently performing path, power, and schedule assignments.

The last part of the dissertation tackled down the issue of service level heterogeneity. We investigated main architectural models in conjunction with their network layer support alternatives. We provided our own distributed service discovery protocol, which accomplishes the implementation of a survivable and adaptive directory system under a highly dynamic networking environment. Besides its performance advantages in scalability and fast service discovery, the supported directory system also enables a strong framework to apply server allocation techniques for reducing server overloads and network congestions.

Although this thesis covered significant amount of issues related to heterogeneous wireless networks, the investigated topics emerged quite recently and there remain equally important open problems to dwell upon. The next section highlights possible outgrowths of the ideas developed in the dissertation.

5.1 Suggestions for Further Study

The following issues remain as the critical subjects of further research:

- Instead of asymptotical results, what is the network capacity of a finite size hybrid network for arbitrary number of ad hoc nodes and access points? When a meticulous network planning is not possible and the only control we have is the population of the access points, the answer to this question would provide us a valuable asset.

- In reality, there are also topological constraints on the infrastructure network. It is important to find a minimal cost infrastructure network that does not suffer from being the bandwidth bottleneck as exemplified in chapter 3.

- To obtain our results, we exploited nice mathematical features of simple and yet intuitive interference models. However, it is essential to understand the impact of more realistic channel models on the capacity figures in comparison to the simple and deterministic models.

- One of the assumptions in chapter 2 is that ad hoc nodes have the same transmission range -which can be interpreted as fixed transmission power for all nodes- and a uniformly distributed traffic pattern. A more interesting direction is to allow the power adaptation with respect to the traffic patterns other than the uniform models in the network.

- In chapter 3, our algorithms are centralized in the sense that they are executed by a central agent that has global network knowledge. Although this itself is not a big obstacle for ad hoc networks with infrastructure assistance, a more desirable solution would be to devise partially or fully distributed algorithms based on only local node information. Such algorithms would be executed independently at each node, yet the transmission schedules and transmit powers should converge to an optimal or near-optimal solutions. It is also desirable to find the guaranteed performance gaps of the heuristic solutions from the optimal one.

- In chapter 4, we did not investigate the effects of several load balancing and resource allocation techniques on server and network performances when implemented on our service discovery proposal. We believe as the communication paradigm shifts from any-to-any communication toward service-centric communication, the next cycle of wireless protocols must be redesigned to build efficient systems. The topic is however beyond the scope of this dissertation.

Appendix A

Proofs of Lemmas in Chapter 2

A.1 Proof of Lemma 3

The proof follows from the L'Hospital Rule and properties of the $\log$ function. We can express $\lim_{x\to\infty}(1+1/a(x))^{b(x)}$ as

$$exp\left(\lim_{x\to\infty}\frac{\ln(1+1/a(x))}{1/b(x)}\right) \ .$$

We have an indeterminate form of $\frac{0}{0}$ and conditions (i)-(ii) in the lemma allow us to apply L'Hospital Rule, which states that the above limit exists in set of extended real numbers if

$$\begin{aligned} & exp\left(\lim_{x\to\infty}\frac{d(\ln(1+1/a(x)))/dx}{d(1/b(x))/dx}\right) \\ & = exp\left(\lim_{x\to\infty}\frac{-\dot{a}(x)/a(x)^2}{1+1/a(x)}(-b(x)^2/\dot{b}(x))\right) \\ & = exp\left(\lim_{x\to\infty}\frac{b(x)^2}{a(x)^2}\frac{\dot{a}(x)}{\dot{b}(x)}\right) \ . \end{aligned} \tag{A.1}$$

exists and it is equal to the limit in (A.1). Hence, we proved the lemma.

A.2 Proof of Theorem 3

We can express $\mathcal{Y}$ as,

$$\mathcal{Y} = \sum_{i=1}^{N} I(i\ is\ connected\ to\ an\ access\ point)\ , \tag{A.2}$$

where I is the indicator function. Clearly, $\mathcal{Y} \leq N$, hence;

$$\begin{aligned} N \geq E[\mathcal{Y}] & \\ &= E\left[\sum_{i=1}^{N} I(i\ is\ connected\ to\ an\ access\ point)\right]\ , \\ &= \sum_{i=1}^{N} E[I(i\ is\ connected\ to\ an\ access\ point)]\ , \\ &\overset{(a)}{=} N \times Prob[i\ is\ connected\ to\ an\ access\ point] \\ &\overset{(b)}{\geq} N \times \left[1 - \left(1 - \frac{A_\epsilon(K)}{4A_R}\right)^K\right]\ . \end{aligned} \tag{A.3}$$

Here, step (a) follows from the fact that each node has the same marginal distribution of being connected to an access point, though they are not independent. And, step (b) is a direct application of the lower bound as given by relation (2.10). Define $\beta(K) = [1 - (1 - A_\epsilon(K)/4A_R)^K]$ and suppose $\beta(K)$ has a limit $\beta^* > 0$. Then, for all $\epsilon > 0$, there exists a real number K_0 such that $|\beta(K) - \beta^*| < \epsilon$ for all $N > K \geq K_0$. Choose $\epsilon = 1/N^2$, thus we have $N\beta > N(\beta^* - \epsilon) = N\beta^* - 1/N$. Or, equivalently,

$$N \geq E[\mathcal{Y}] > \beta^* N - \gamma\ ,\ \forall N \geq K_0\ ,\ \gamma > 0\ ,$$

where γ is arbitrarily small. Corollary-1 implies the existence of $\beta^* > 0$ completing the first part of the theorem.

Proving the second statement of the theorem is again a brute-force application of theorem-3. The weak connectivity of node i with arbitrarily high probability forces β^* to be 1 and $E[\mathcal{Y}]$ becomes arbitrarily close to N. Considering this result along with the observation $E[\mathcal{Y}] = N$ if and only if Prob[all nodes are connected to an access point] = 1 suffices to prove the second part of the theorem.

BIBLIOGRAPHY

[1] Girish Patel and Steven Dennett. The 3GPP and 3GPP2 Movements Toward an All-IP Mobile Network. *IEEE Personal Communications*, pages 62–64, Aug. 2000.

[2] Joseph P. Macker, Vincent D. Park, and M. Scott Corson. Mobile and Wireless Internet Services: Putting the Pieces Together. *IEEE Communications Magazine*, 39(6):148–155, June 2001.

[3] R. Droms. Dynamic Host Configuration Protocol. *IETF RFC 2131*, March 1997.

[4] Anthony McAuley, Subir Das, Sunil Madhani, Shinichi Baba, and Yasuro Shobatake. Dynamic Registration and Configuration Protocol (DRCP). *Internet Draft, draft-itsumo-drcp-01.txt*, July 2000.

[5] Ying-Dar Lin and Yu-Ching Hsu. Multihop Cellular: A New Architecture for Wireless Communications. In *IEEE INFOCOM 2000*, pages 1273–1282, March 2000.

[6] John W. Noerenberg II. Bridging Wireless Protocols. *IEEE Communications Magazine*, 39(11):90–97, Nov. 2001.

[7] George Neonakis Aggélou and Rahim Tafazolli. On the Relaying Capability of Next-Generation GSM Cellular Networks. *IEEE Personal Communications*, pages 40–47, Feb. 2001.

[8] A.N. Zadeh, B. Jabbari, R. Pickholtz, B. Vojcic. Self-organizing packet radio ad hoc networks with overlay (SOPRANO). *IEEE Communications Magazine*, 40:149–157, June 2002.

[9] Jiang Xie and Ian F. Akyildiz. A Distributed Dynamic Regional Location Management Scheme for Mobile IP. In *Proceedings of IEEE Infocom*, June 2002.

[10] C.A. St Jean, A.N. Zadeh and B. Jabbari. Combined Routing, Channel Scheduling, and Power Control in Packet Radio Networks with Cellular Overlay. In *IEEE VTC2002 Spring*, volume 4, pages 1960–1964, 2002.

[11] Haiyun Luo, Ramachandran Ramjee, Prasun Sinha, Li (Erran) Li, Songwu Lu. UCAN: A Unified Cellular and Ad-Hoc Network Architecture. In *ACM Mobicom-2003*, pages 353–367, 2002.

[12] Hingyi Wu. *iCAR: and Integrated Cellular and Ad hoc Relaying System*. PhD thesis, State University of New York at Buffolo, 2002.

[13] L. Tong, Q. Zhao, and S. Adireddy. Sensor networks with mobile agents. In *MILCOM'03*, October 2003.

[14] E. Guttman, C. Perkins, J. Veizades, and M. Day. Service Location Protocol, Version 2. *IETF RFC 2608*, June 1999.

[15] Sun Microsystems, JINI Architecture Specification, Nov. 1999.

[16] Microsoft Corporation. Universal Plug and Play: Background, www.upnp.org/resources/UPnPbkgnd.htm, 1999.

[17] Salutation Consortium. Salutation Architecture Specification, www.salutation.org/specordr.htm, 1999.

[18] Specification of the Bluetooth System. *www.bluetooth.com*, Dec. 1999.

[19] Michel Barbeau. Service Discovery Protocols for Ad Hoc Networking. *CASCON 2000 Workshop on ad hoc communications*, 2000.

[20] Ulaş C. Kozat and Leandros Tassiulas. Throughout Capacity of Random Ad hoc Networks with Infrastructure Support. In *Proc. ACM MOBICOM 2003*, September 2003.

[21] Ulaş C. Kozat and Leandros Tassiulas. Throughput Scalability of Wireless Hybrid Networks over a Random Geometric Graph. *to appear in Wireless Networks (WINET)*, 2004.

[22] Ulaş C. Kozat, Iordanis Koutsopoulos, Leandros Tassiulas. A Framework for Cross-layer Design of Energy-efficient Communication with QoS Provisioning in Multi-hop Wireless Networks. In *Proc. IEEE INFOCOM 2004*, March 2004.

[23] Ulaş C. Kozat, G. Kondylis, and M.K. Marina. Virtual Dynamic Backbone for Mobile Ad hoc Networks. In *Proc. IEEE ICC-2001*, June 2001.

[24] Ulaş C. Kozat and Leandros Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *Proc. IEEE INFOCOM 2003*, March 2003.

[25] Ulaş C. Kozat and Leandros Tassiulas. Service Discovery in Mobile Ad Hoc Networks: An Overall Perspective on Architectural Choices and Network Layer Support Issues. *AdHoc Networks Journal*, 2(1):23–44, January 2004.

[26] P. Gupta and P.R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[27] M. Gastpar and M. Vetterli. On the capacity of wireless networks: The relay case. In *Proceedings of the IEEE Infocom*, 2002.

[28] Benyuan Liu, Zhen Liu, and Don Towsley. On the capacity of hybrid wireless networks. In *Proceedings of the IEEE Infocom*, 2003.

[29] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proceedings of the IEEE Infocom*, 2001.

[30] P. Gupta and P.R. Kumar. Towards an information theory of large networks: An achievable rate region. In *IEEE Int. Symp. Info. Theory, Washington DC*, June 2001.

[31] Enrique J. Duarte-Melo and Mingyan Liu. Data-gathering wireless sensor networks: Organization and capacity. *Computer Networks (COMNET)*, 43(4):519–537, November 2003. Special Issue on Wireless Sensor Networks.

[32] Mathew D. Penrose. A strong law for the longest edge of the minimal spanning tree. *The Annals of Probability*, 27(1):246–160, 1999.

[33] P. Gupta and P.R. Kumar. Critical power for asymptotic connectivity in wireless networks. *Stochastic Analysis, Control, Optimization and Application: A Volume in Honor of W.H. Fleming*, March 1998.

[34] P.T. Olivier Dousse and M. Hasler. Connectivity in ad-hoc and hybrid networks. In *Proceedings of the IEEE Infocom*, 2002.

[35] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[36] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Inc., 1991.

[37] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[38] J. Zander. Distributed co-channel interference control in cellular radio systems. *IEEE Trans. Veh. Technol.*, 41(3):305–311, August 1992.

[39] J. Zander. Performance of optimum transmitter power control in cellular radio systems. *IEEE Trans. Veh. Technol.*, 41:57–62, February 1992.

[40] S.A. Grandhi, R. Vijayan, and D.J. Goodman. Distributed algorithm for power control in cellular radio systems. In *Proc. 30th Annual Allerton Conf. Comm., Control Computing, Urbana, Illinois*, September 1992.

[41] S.A. Grandhi, R. Vijayan, D.J. Goodman, and J. Zander. Centralized power control in cellular radio systems. *IEEE Trans. veh. Technol.*, 42:466–468, November 1993.

[42] G.J. Foschini and Z. Miljanic. A simple distributed autonomous power control algorithm and its convergence. *IEEE Trans. veh. Technol.*, 42(4):641–646, November 1993.

[43] R.D. Yates. A framework for uplink power control and base station assignment. *IEEE J. Select. Areas Commun.*, 13:1341–1347, September 1995.

[44] N. Bambos, S. Chen and G. Pottie. Radio Link Admission Algorithms for Wireless Networks with Power Control and Active Link Quality Protection. In *Proc. Infocom-1995*, pages 97–104, 1995.

[45] C. Huang and R.D. Yates. Rate of convergence for minimum power assignment algorithm in cellular radio systems. *Wireless Networks*, 4:223–231, 1998.

[46] Sennur Ulukus and R.D. Yates. Stochastic Power Control for Cellular Radio Systems. *IEEE Trans. on Commun.*, 46(6):784–798, June 1998.

[47] S. Hanly and D. Tse. Power control and capacity of spread-spectrum wireless networks. *Automatica*, 35:1987–2012, 1999.

[48] Fredrik Berggren, Seong-Lyun Kim, and Jens Zander. Joint Power Control and Intracell Scheduling of DS-CDMA Nonreal Time Data. *IEEE JSAC*, 19(10):1860–1870, October 2001.

[49] T. Elbatt and A. Ephremides. Joint Scheduling and Power Control for Wireless Ad Hoc Networks. In *Proc. Infocom-2002*, June 2002.

[50] R.L. Cruz and A.V. Santhanam. Optimal Routing, Link Scheduling and Power Control in Multi-hop Wireless Networks. In *Proc. Infocom-2003*, April 2003.

[51] Randeep Bhatia and M. Kodialam . On Power Efficient Communication over Multi-hop Wireless Networks: Joint Routing, Scheduling and Power Control . In *Proc. of IEEE Infocom-2004*, March 2004. to appear.

[52] R.L. Cruz and A.V. Santhanam. Optimal Link Scheduling and Power Control in CDMA Multihop Wireless Networks. In *Proc. Globecom-2002*, November 2002.

[53] R. Wattenhofer, Li Li, Paramvir Bahl, and Yi-Min Wang. Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks. In *Proc. Infocom-2001*, 2001.

[54] R. Ramanathan and R. Rosales-Hain. Topology Control of Multihop Wireless Networks using Transmit Power Adjustment. In *Proc. Infocom-2000*, 2000.

[55] J. Monks, V. Barghavan and W. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In *Proc. Infocom-2001*, April 2001.

[56] V. Rodoplu and Teresa Meng. Minimum energy mobile wireless networks. *IEEE JSAC*, 17(8):1333–1344, 1999.

[57] S. Singh, M. Woo and C.S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. of Fourth Annual ACM/IEEE Conference on Mobile Computing and Networking*, pages 181–190, October 1998.

[58] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[59] A.J. Goldsmith and S.-G. Chua. Variable-rate variable-power MQAM for fading channels. *IEEE Trans. on Commun.*, 45(10):1218–1230, October 1997.

[60] Michael R. Garey and David S. Johnson. *Computers and Intractability A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[61] Kevin Fall and Kannan Varadhan (Eds.). ns notes and documentation. available from http://www.isi.edu/nsnam/ns/.

[62] D.J. Baker and A. Ephremides. The architectural organization of a mobile radio networks via a distributed algorithm. *IEEE Transactions on Communication*, 29:1694–1701, Nov. 1981.

[63] Raghupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan. CEDAR: A core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8):1454–1465, Aug. 1999.

[64] Raghupathy Sivakumar, Bevan Das, and Vaduvur Bharghavan. Spine routing in ad hoc networks. *Cluster Computing 1*, pages 237–248, 1998.

[65] Ben Liang and Zygmunt J. Haas. Virtual Backbone Generation and Maintenance in Ad Hoc Network Mobility Management. In *IEEE INFOCOM 2000*, March 2000.

[66] C.R. Lin and M. Gerla. Adaptive Clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, Sep. 1997.

[67] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multi-hop mobile wireless networks for quality-of-service support. *Mobile Networks and Applications*, 3(1):101–119, 1998.

[68] M. Chatterjee, S.K. Das, and D. Turgut. An on-demand weighted clustering algorithm (WCA) for ad hoc networks. In *Proceedings of IEEE Globecom 2000*, pages 1967–1701, Nov. 2000.

[69] C. Partridge, T. Mendez, and W. Milliken. Host Anycasting Service. *Internet RFC 1546*, Nov. 1993.

[70] V. Park and J. Macker. Anycast Routing for Mobile Services. In *Proc. Conference on Information Sciences and Systems (CISS) '99*, Jan. 1999.

[71] S. Bhattacharjee, M. Ammar, E. Zegura, V. Shah and Z. Fei. Application-Layer Anycasting. In *Proc. INFOCOM '97*, April 1997.

[72] Dongyan Xu, Klara Nahrstedt, and Duangdao Wichadakul. QoS-Aware Discovery of Wide-Area Distributed Services. In *1st International Symposium on Cluster Computing and the Grid*, May 2001.

[73] S-J. Lee, W. Su, and M. Gerla. On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks. *Internet Draft*, *draft-ietf-manet-odmrp-02.txt*, Jan. 2000.

[74] C-C. Chiang, Mario Gerla, and L. Zhang. Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks. *Cluster Computing*, 1998.

[75] M.S. Corson and S.G. Batsell. A Reservation-Based Multicast (RBM) Routing Protocol for Mobile Networks: Initial Route Construction phase. *ACM/Baltzer Wireless Networks*, pages 427–450, 1995.

[76] J.J. Garcia-Luna-Aceves and E. Madruga. The Core-Assisted Mesh Protocol. *IEEE Journal on Selected Areas in Communications*, Aug. 1999.

[77] C.E. Perkins, E.M. Royer, and S.R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. *Internet Draft, draft-ietf-manet-aodv-05.txt*, 2000.

[78] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast Core-Extraction Distributed Ad hoc Routing. In *IEEE Wireless Communications and Networking Conference*, Sep. 1999.

[79] C.W. Wu and Y.C. Tay. Ad hoc Multicast Routing Protocol utilizing Increasing id-numbers. *Internet Draft, draft-ietf-manet-amris-spec-00.txt*, Nov. 1998.

[80] David B. Johnson et al. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. *Internet Draft, draft-ietf-manet-dsr-05.txt*, March 2001.

[81] M.U. Uyar, J. Zheng, M.A. Fecko, S. Samtani, and P.T. Conrad. Evaluation of architectures for reliable server pooling in wired and wireless environments. *IEEE Journal of Selected Areas of Communications*, 2003. invited paper.

[82] M.U. Uyar, J. Zheng, M.A. Fecko, and S. Samtani. Performance study of reliable server pooling. In *Proc. of IEEE Int'l Symp. Network Comput. Appl. (NCA)*, pages 205–212, Cambridge, MA, 2003.

[83] M.U. Uyar, J. Zheng, M.A. Fecko, and S. Samtani. Reliable server pooling in highly mobile wireless networks. In *Proc. IEEE Int'l Symp. Comput. Commun. (ISCC)*, pages 627–632, Kemer-Antalya, Turkey, 2003.