Design of Multi-tier Wireless Mesh Networks

Thesis

Submitted in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

by

Raghuraman Rangarajan

Roll No. 99429001

Advisor

Prof. Sridhar Iyer

K.R. SCHOOL OF INFORMATION TECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY - BOMBAY

MUMBAI - 400 076

2009

APPROVAL SHEET

Thesis entitled "**Design of Multi-tier Wireless Mesh Networks**" by **Raghuraman Rangarajan** is approved for the degree of **DOCTOR OF PHILOSOPHY**.

Examiners

____

____

____

Supervisor

____

Chairman

____

Date: ____________

Place: ____________

INDIAN INSTITUTE OF TECHNOLOGY, BOMBAY, INDIA

CERTIFICATE OF COURSE WORK

This is to certify that Mr. Raghuraman Rangarajan was admitted to the candidacy of the Ph.D. Degree in January, 2004 after successfully completing all the courses required for the Ph.D. Degree programme. The details of the course work done are given below.

Sr. No.	**Course Code**	**Course Name**	**Credits**
1.	IT 620	Seminar	4
2.	IT 642	Data warehousing and Data mining	6
3.	IT 601	Object-oriented technologies	6
4.	IT 661	IT Entrepreneurship	6

I.I.T Bombay

Dy. Registrar (Academic)

Date:

Abstract

In this thesis, we investigate the issue of automated design of capacity-constrained Wireless Mesh Networks (WMN). We argue for the necessity of applying network design methodologies from wired and cellular network fields in wireless network design scenarios and present algorithms for Wireless Local Area Networks (WLANs) and backbone topology design. The deployment scenario we envision is a campus of office buildings requiring wireless connectivity. The client nodes to be deployed in each office, their application traffic requirements and the deployment layout are given. We identify three main stages in the design of such wireless networks: 1) Association of clients to access points, 2) WLAN topology construction and 3) Backbone topology construction. Capacity provisioning and network cost minimisation are the two constraints imposed on the design problem.

In the first stage, we define the AP-assignment problem, that is, the problem of associating client nodes with the nearest Access Point (AP) and investigate various access point association scenarios. We compute the performance of 802.11 WLANs under homogeneous realtime application deployments with theoretical and OPNET simulation results for various voice and video codecs. We capture the performance, in terms of number of flows supported for an application, as the capacity of a WLAN. We then examine heterogeneous application deployments and show the inability of 802.11 DCF mechanism to handle them. We propose an extended DCF for handling scenarios where delay-sensitive and delay-tolerant applications are deployed together. Next, we propose a novel approach called sub-optimal AP-assignment and show that the utilisation of 802.11 DCF can be increased by up to 75%. We show how the solutions to the AP-assignment problem can then be used for abstract representations of the 802.11 DCF MAC in wireless design problems.

In the second stage, we define a network design problem for constructing WLAN topologies. The scenario we consider is intra-office connectivity for client nodes. We present a recursive bottom-up algorithm for capacity-constrained topology construction. The topology construction algorithm considers the deployment scenario, the client nodes deployed and their application scenarios as inputs. The AP-assignment solutions are used as a construction mechanism in the form of link specification functions. We introduce a new object, called composite unit, as an abstract building block for network topology construction. The generated topology is then validated with simulations (OPNET Modeler).

In the third stage, we define a wireless mesh network (WMN) design problem for constructing a mesh topology for a campus-like scenario. The design problem is defined as a case of traditional network design problem for optimal node location and topology construction. These problem form Mixed Integer Linear Programming (MILP) formulations and are solved with an MILP solver (CPLEX).

We have built a tool to implement our multi-tier wireless design solution. The tool, the Wireless Infrastructure Deployment tool (WIND), designs topologies for both WLANs as well as WMNs. WIND takes information about nodes deployed, their properties and deployment layouts to construct logical network topologies. The input parameters and output topology use XML schemas compatible with data formats of the OPNET Modeler network simulator. This allows the topology to be input directly to the simulator for validation. Using simulation we show that the constructed topologies satisfy the given constraints on application scenarios, protocols and deployment scenario. We present case studies of constructing topologies for both WLAN and WMN scenarios.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The standardisation of wireless technologies and the perceived ease of use of wireless communication has played an important role in the adoption of wireless devices. The main usage of wireless networks has emerged to be broadband internet access. Any deployment scenario where wired network deployment proves to be costly (or infeasible), or where mobility is an important requirement, becomes a candidate scenario for deploying wireless networks. One main aspect driving this growth has been the introduction of the IEEE 802.11 standard. The advent of wireless networks has allowed the replacement of wired networks with wireless at the last hop and this effort has also moved towards replacing last mile connectivity.

We first discuss present-day wireless networks deployments in Section 1.1 and discuss deployment issues in networks in Section 1.2. We then present the problem of wireless network design as a multi-tiered network design problem in Section 1.3.1. We discuss the design issues and the design stages in this problem, an outline of our solution approach and details of the tool we have developed. In Section 1.4, we discuss the contributions of this thesis and present the chapter organisation in Section 1.5.

1.1 Present-day wireless deployments

Wireless device usage has increased with the standardisation of wireless technologies especially, IEEE 802.11 [80203a]. For wired networks, Ethernet is widely deployed for inter-office and intra-office connectivity and is the defacto standard [Eth05]. But, Ethernet networks suffer some disadvantages: they are cumbersome and costly to deploy, re-configuration of clients is difficult and they do not provide connectivity for mobile devices. Wireless networks, have the following aims to alleviate the disadvantages of wired connectivity: a) Remove physical connectivity to the network, b) Allow users to connect to a network from any location in an area instead of fixed points, c) Allow re-configuration of network topology at little cost and d) Remove cost of deployment of wired media.

A common illustration for wireless deployment is an office space with fixed clients such as PCs and mobile clients like laptops and PDAs. An office wireless network deployment involves providing wireless hubs, like Access Points (AP), to which the clients connect. The APs, in turn, are connected to the backhaul (core network) using wired infrastructure. This form of deployment is called single-hop infrastructure mode deployment. It allows wireless clients to move at will in a deployment area as long as they are able to establish a connection with at least one AP.

Ad hoc network, or peer-to-peer network, is another deployment methodology that allows clients to directly connect to each other, without any infrastructure, as long as they are able to establish connections with each other. This allows networks to be formed (and broken) without any infrastructure support. The advantages of ad hoc networks, over infrastructure mode networks, are the ability of mobile clients to dynamically form a network and the use of a multi-hop network architecture. Figure 1.1 illustrates single-hop infrastructure and ad hoc deployment scenarios.

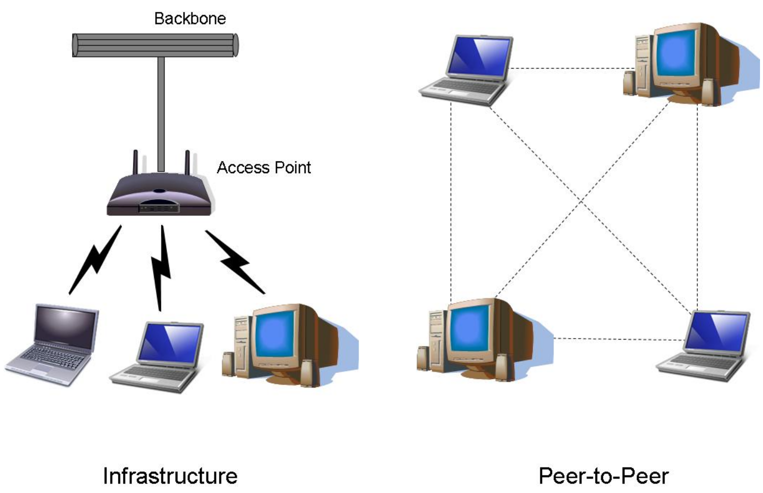

Figure 1.1: Infrastructure and Peer-to-peer mode for WLANs.

The 802.11 standard has been used for wireless connectivity in both office and home scenarios in the form of Wireless Local Area Networks (WLANs) [Var03]. In public scenarios, hotspots provide single-hop or adhoc connectivity [BVB03]. Also, in scenarios like disaster management, where network infrastructure is unavailable or cannot be built, adhoc wireless networks are used [Rap02]. Portable wireless multi-function devices with inbuilt bar scanners and other functionalities for book-keeping can be used in retail, warehouse, shipping and other commercial scenarios [Mot].

The main challenge in initial infrastructure deployment approaches was the appropriate placement of infrastructure nodes (like routers and APs). The APs needed to be placed such that clients in any given area are assured of connectivity to *at least* one AP. This signal coverage planning approach, or Radio Frequency (RF) planning, involved the study of RF characteristics of a particular deployment area and computing the number of APs required to adequately serve the given number of clients [ZPK03].

Wireless networks are also being used for backhaul networks. Short to medium range backhauls can be used to connect campus and metropolitan Wide Area Networks (WANs). Long distance point to point backhauls can be used in areas where deployment of a wired backhaul proves to be expensive. Wireless mesh networks is an example of a campus WAN and is being widely deployed as a backhaul for WLANs [AWW05]. IEEE 802.16 [80204], and its flavours like WiMAX [WiM], are also being used for point to multi-point connectivity.

However, due to the wireless nature of these networks, some issues unique to this domain affect the deployment of these networks. In the next section, we discuss these issues and argue that an integrated approach towards design is required to address them.

1.2 Issues in wireless deployments

The deployment of wireless networks has the following issues that affect the cost of a network deployment:

1. Network capacity provisioning.
2. Network modification cost.
3. Wired nature of backhaul network.

We now examine these issues in detail.

1.2.1 Network capacity provisioning

Network capacity provisioning is defined as the issue of provisioning a network to satisfy the capacity requirements of client nodes in the network. Network provisioning in wireless networks is an important issue in comparison to Ethernet. Clients on Ethernet networks are provisioned with at least 10 Mbps connectivity [Eth05].

Ethernet bandwidth of 100 Mbps have become common while Gigabit Ethernets are also being deployed [Tan02]. On the other hand, wireless networks are limited by their physical characteristics [KMK05]. The 802.11b standard, which is the most commonly used physical standard, provides only up to 11 Mbps of bandwidth [80203a]. Bandwidth in the unlicensed band, depending on the physical layer, can reach up to 54 Mbps using 802.11g [80203b]. However, the actual available bandwidth is lower due to physical layer characteristics [RIG05].

Moreover, WLAN designers focus mainly on coverage and not on capacity provisioning [McL03]. Coverage addresses the issue of any node in the deployment area being able to connect to at least one AP. But, even when all clients can connect to APs, sufficient APs have to be provided such that individual client traffic requirements are satisfied. Simply providing additional APs may also not solve the problem as such an ad hoc deployment approach affects bandwidth provisioning [YCG+03]. Also, unlike a wired link, the non-deterministic nature of the wireless link itself affects such provisioning [KMK05]. Hence, AP-client associations have to be studied carefully.

Another issue is that the contention-based access of 802.11 MAC results in difficulty in network provisioning [RIG05]. The main mode of access in the IEEE 802.11 standard is the Distributed Co-ordination Function (DCF) which uses a Collision Sense Multi-Access with Collision Avoidance (CSMA/CA) to access the wireless channel [80203a]. CSMA/CA ensures equal opportunity access to all nodes in the system. While this makes the standard simple to implement and use, it is difficult to provide any kind of bandwidth provisioning. Other techniques in the standard, like Quality of Service (QoS) extensions [80205], that help in provisioning have not found wide usage. Hence, intelligent capacity provisioning needs is also an important issue.

1.2.2 Network modification cost

Related to network provisioning is the cost of network deployment. Any modification in the system, like addition of new clients or re-arrangement of office layout, may result in AP locations being changed. Additional APs may also need to be installed in the network. The coverage planning may have to be re-evaluated as the existing results may prove to be experimentally invalid. The wired backhaul installed to connect the APs may also have to be reconfigured. All these issues, and especially the wiring of backhaul connectivity, drastically affects the costs incurred for any modifications in deployment. Hence, any network deployment has to factor these issues at the design stage.

1.2.3 Wired nature of backhaul

The wireless infrastructure network is in effect a single-hop wireless network. While WLANs do alleviate the last-hop connectivity problem by having a wireless link, the issue of connecting the APs still exists. The APs are still connected to the gateway through a wired backhaul. The issue then is of the disadvantages of the wired backhaul. As discussed above, any reconfiguration of network is costly, due to removal and rewiring.

Increasingly, the backhaul of the network is also being replaced with a wireless network. Modified versions of IEEE 802.11 have been used for backhaul networks and Wireless Mesh Networks (WMNs) have emerged as an important paradigm [AWW05]. Also, other standards like IEEE 802.16 have emerged to provide backhaul connectivity [80204]. Hence, like for WLANs, design and provisioning of wireless backhaul networks are also issues to be studied.

1.3 Design of wireless mesh networks

With the increasing demand for QoS and the expansion in deployment scenarios, the issue of designing wireless networks to suitably handle traffic loads has become important. While such issues have been handled in wired networks for backhaul traffic, the problem of *wireless* network topology design and provisioning still remains in its infancy. The issues discussed in the previous section can be considered together as the problem of *wireless network design*. Any wireless network to be deployed has to address one or more of these issues.

Design is a well studied area in the field of wired [PM04] and cellular networks [WH03]. The issue of provisioning wired and cellular networks as well as network topology design is well studied. In cellular networks, additionally, the issue of coverage has also been studied. As far as wireless data networks are concerned, except for coverage planning, the issue of design deserves more attention.

1.3.1 Problem scope: Multi-tiered network design

We consider a deployment scenario of a campus of offices, with WLANs in each office, connected to each other and the gateway through a WMN backhaul network (See Figure 2.1 for an example). Given the deployment scenario, client nodes properties and their demands, we generate an appropriate wireless network topology by constructing intermediate backhaul nodes and links, and provisioning capacity on each wireless link.

The scenario we envision is that of constructing WMNs with WLAN as client access networks (tier 1) and a mesh backhaul network (tier 2) while provisioning for capacity and minimising cost. The backhaul network provides both inter-WLAN and gateway connectivity.

We identify and define three distinct stages in this problem: 1) associating clients with APs, 2) constructing the logical WLAN topology and 3) constructing the backhaul wireless network. First, we discuss the issues in network design.

1.3.2 Design issues considered

We consider three issues in the network design problem: 1) Capacity provisioning, 2) Cost minimisation and 3) Integrated network design.

Capacity provisioning

The main aim of this work is to study capacity provisioning in wireless networks with special emphasis on IEEE 802.11 DCF WLANs. As we discuss in Chapter 2, other studies concentrate on establishing coverage. We study the issue of provisioning under different cases: 1) provisioning in the presence of homogeneous application scenarios in a WLAN environment, 2) provisioning in the presence of heterogeneous application scenarios in a WLAN environment and 3) provisioning of packet traffic in backhaul wireless mesh networks.

Cost minimisation

In each network design stage, we minimise the network deployment cost. The deployment cost involves the cost of deploying infrastructure nodes and links in the network.

Integrated network design

We propose an integrated approach for wireless network design. The network model we consider is a wireless mesh network as backhaul network connecting infrastructure mode WLANs to the gateway. We consider a bottom-up approach towards the design of the network from the stage of client association with AP to constructing a WLAN topology to constructing the backhaul mesh topology (See Figure 3.1 for solution approach flowchart).

1.3.3 Solution approach

The three distinct stages in our wireless network design solution are as follows:

Provisioning in 802.11 WLANs

The first stage involves the study of the AP-client association in infrastructure IEEE 802.11 WLANs. This problem, defined as AP assignment problem, involves the study of AP-client mappings such that, individual client application requirements are satisfied.

In an infrastructure mode deployment, client nodes access the network through access points (APs) with a single-hop. The capacity of an AP, in an infrastructure WLAN, can then be defined in terms of the number of client nodes that associate with it. The maximum number of clients connected to an AP now depends on the type of clients associated with it. The client node type is characterised by its application scenarios. Homogeneous clients run a single application scenario while heterogeneous nodes may run many applications. We investigate the AP-assignment problem for homogeneous and heterogeneous client application deployments.

We first analyse the capacity of 802.11 WLANs for homogeneous real-time application deployments (with voice and video as examples). Using heuristics for calculating the backoff time in the DCF mechanism, we compute theoretical capacities in terms of number of application flows and compare these with simulation results. We make the observation that the DCF mechanism is adequate in providing *some* guarantees in homogeneous deployments and is inadequate for handling heterogeneous traffic.

We then extend the AP-assignment problem to heterogeneous deployments and analyse joint deployment scenarios of realtime and non-realtime applications. For heterogeneous application traffic scenarios, we show that DCF performs poorly without additional functionalities for assisting priority traffic. We then show with simulation that a simple extension to DCF, using separate contention windows, can be used to tackle this issue. We then present a novel approach for solving the AP-assignment problem called sub-optimal application deployment. This approach uses prioritisation of applications scenarios as a mechanism for improving the system performance. We show that this approach results in significant improvement of the number of heterogeneous flows in the system and also improves system utilisation up to 75%. We also show how the above techniques can be used as an Access Control Limit (ACL) scheme for WLAN APs.

Constructing the WLAN topology

We then design a network for connectivity within an office. This involves building a topology to connect the clients together to form a WLAN. For this, we use an abstract graph representation of the deployment layout, list of nodes to be deployed and their traffic properties. We define a concept, called affinity factor, to capture node deployment scenarios. We then use the solutions to the AP-assignment problem to construct AP-client mappings. We constraint this network with the client capacity requirements and use a bottom-up approach to construct logical topologies for WLANs.

Constructing the wireless backhaul mesh topology

Next, we design a mesh backhaul network topology to connect and form the backhaul network. This backhaul provides inter-WLAN connectivity as well as gateway connectivity to WLAN clients. Also, we constrain this backhaul in order to, 1) consider the client capacity requirements, and 2) minimise the number of backhaul nodes/APs required to connect to the network. This problem falls under the class of optimal node location and traffic provisioning problems and is a Mixed Integer Linear Programming (MILP) problem [PM04]. We solve this problem and compute mesh node locations and the backhaul topology using the CPLEX solver [Sol]

1.3.4 Wireless Infrastructure Deployment tool

We have developed a tool for designing multi-tier wireless mesh networks. The Wireless Infrastructure Deployment tool (WIND) implements our techniques for designing multi-tier networks - WMNs with WLANs as clients. There are many capacity and coverage provisioning tools available for cellular networks [EDXb, Ato]. For WLANs and WMN, while there are tools in the closed domain [EDXa, Mes], to the best of our knowledge no such capacity provisioning tool exists for WLAN + WMN design in the open domain.

WIND takes its input information, modeled on simulator input formats, about client node demands and layout of deployment area. WIND also uses information on AP-client mappings from a fact table. The client demands are handled as follows: 1) WIND attempts to build a topology to satisfy intra-WLAN demands 2) WIND aggregates inter-WLAN and gateway traffic demands at the root node of the WLAN. The topology is constructed using an all-pairs shortest path algorithm for finding the shortest paths between a source-destination pair of any demand. The output is presented in a simulator-friendly format for ease of integration with a simulator.

In the next stage of WIND, a backhaul network is constructed with the root nodes of WLANs as client nodes. The demands of the root nodes are the aggregated demands of inter-WLAN and gateway traffic demands of WLAN clients. In this stage, locations of potential mesh nodes and number of available links are given. This problem falls under the category of a facility location problem in optimization and is a Mixed Integer Linear Programming (MILP) problem. WIND uses an MILP solver to evaluate the potential mesh nodes and links and uses its output to construct the topology.

1.4 Thesis contributions

The main contributions of this thesis are as follows:

1. Study on designing wireless network with emphasis on wireless mesh and local area networks. The features of this design approach are: 1) Integrated network design of WLANs and WMNs, 2) Capacity provisioning in wireless networks and 3) Network deployment cost minimisation.

2. Study of AP-assignment problem of association of client nodes to APs in infrastructure networks with capacity provisioning. The contributions are: 1) Theoretical and simulation study of homogeneous application deployments in 802.11 DCF WLANs, 2) Simulation study of heterogeneous application deployments with emphasis on joint deployment of realtime and non-realtime applications, 3) Extending the DCF mechanism for alleviation of provisioning issues in 802.11 DCF mechanism, 4) Study of prioritisation of applications and sub-optimal application deployment techniques as extensions to DCF. For heterogeneous deployments, we show system utilisation improvement up to 75% over normal DCF and 5) An Access Control Limit (ACL) mechanism for management of APs infrastructure WLANs using sub-optimal application deployment techniques. This mechanism is useful for managing the AP at optimal system operating points in heterogeneous application deployment

scenarios.

3. A framework for WLAN and WMN network design. The framework enables: 1) Capacity constrained design of WLANs and WMNs, 2) Link characterisations for heterogeneous application deployments based on solutions to the AP-assignment problem and 3) WMN network design as a joint node locationing and dimensioning optimisation problem.

4. The Wireless Infrastructure Deployment (WIND) tool that implements the above framework for design of two-tiered networks: WMN with client WLANs. The tool features are: 1) Integrated approach for design of AP-assignment, WLAN and WMN topology construction and 2) Input parameter and output topology XML formats in conformance with OPNET Modeler for integration as a design tool to the simulator.

To the best of our knowledge, no other open domain tool for capacity-constrained design of multi-tier WLAN + WMN wireless networks are available.

1.5 Chapter organisation

The organization of the thesis is as follows. In Chapter 2, the motivation that derives the rationale and the need for advanced wireless network design is presented along with the related work. In Chapter 3, we define the wireless design problem statement and outline our solution approach. In Chapters 4 and 5, we consider the problem of bandwidth provisioning in a WLAN, with homogeneous and heterogeneous application deployments. We study, with theoretical and simulation results, the deployment of realtime (voice and video) traffic along with non-realtime applications. In Chapter 6, we present the details pertaining to the framework of the design tool WIND for WLAN. In Chapter 7 we study design of the backhaul WMN ad present the extension to the WIND framework for WMNs. In Chapter 8, we discuss the issues that can be pursued to extend this work and conclude.

Chapter 2

Motivation and related work

In this chapter, we discuss wireless networks and network planning. Wireless technology has been used for communication since the development of radio by Tesla and Marconi in 1897 [Rad]. In the last few decades, various wireless networks have been developed for transmission of data and voice. Wireless networks brought along advantages of tetherless connectivity, allowing users to be mobile and reducing infrastructure cost. The advent of IEEE standards for Wireless Local Area Networks (WLANs) has resulted in wide spread data communication using personal computers and handheld devices.

In Section 2.1 we provide an introduction to wireless networks with emphasis on data networks and WLANs. In Section 2.2, we illustrate an example deployment scenario to highlight the issues involved in planning. We discuss current approaches for designing and deploying wireless networks and also discuss their drawbacks. We use the example to establish, a) the class of wireless networks under investigation and b) the advantage of planning in designing and deploying networks. In Section 2.3, we then discuss the literature in related areas.

2.1 Introduction to wireless networks

Wireless networks are increasingly being used for voice and data connectivity. Since the cellular concept was developed in the 1960s, there has been a rapid growth in the development and the deployment of wireless networks. The aim to reduce network infrastructure costs, support mobile networking applications and provide improved connectivity has driven the growth of wireless networks. Wireless networks have also been deployed where wired networks did not previously exist or would have been difficult to deploy.

Wireless networks for voice connectivity have been in use for a few decades and are widely prevalent. They have mostly replaced wired telephony with mobile connectivity. Cellular networks have evolved from being pure voice networks in the initial generations to voice networks with some data (2G and 2.5G) to now provide both voice and data capabilities (3G and beyond) [Rap02].

In the data networks space, wireless networks are also being designed for various deployment scenarios. *Fixed wireless* standards for *wide area networks* (WANs) are now used for replacement of fibre optic cables or copper lines between fixed points from a few hundred to a few kilometres apart. WLANs are being used for replacement of wires from a few feet to a few hundred feet. *Wireless Personal Area Networks* (WPANs), like Bluetooth [Gro07], are used to remove wires from the personal workspace with wireless connections.

In this thesis, we concentrate mainly on wireless data networks and their application and deployment scenarios. Hence, our focus in this chapter and discussion on the related work will be on issues in wireless data networks. The main applications therefore are data oriented and voice is analysed as voice over data networks (in the form of voice over IP (VoIP))

2.1.1 Classification of wireless data networks

The classification of wireless data networks can be done based on their application and deployment scenarios.

Infrastructure and Peer-to-peer networks

Wireless networks in local area networking are of two types: Peer-to-peer and Infrastructure. A peer-to-peer, or *adhoc*, network is composed of two or more computers connected directly to each other without the use of an AP. Such networks can be rapidly deployed and are cost effective. Examples of such networks are hotspots and mobile ad hoc networks [RT99].

An infrastructure-based network connects computers to the network through a system of APs. These APs are connected to the backhaul[1] by a wired infrastructure. APs can also be connected to the backhaul using multiple hops of wireless links. Infrastructure-based networks provide a direct mapping of hub-based Ethernet. Current deployments typically have a single-hop architecture. Clients are connected to APs through a wireless hop. The APs are in turn connected to the rest of the network through the wired medium [80203a].

The type of system used depends on various factors. Peer-to-peer networks can be easily setup but suffer from bandwidth restrictions. Also, coverage depends on the proximity of the client devices. Peer-to-peer networks are good for rapid deployments where presence of *some* network is a requirement rather than *quality* of the network (with respect to bandwidth constraints). Peer-to-peer networks perform well for environments like hotspots, disaster management scenarios or conference hall connectivity. The disadvantage of not providing any guarantees of bandwidth or connectivity makes peer-to-peer networks a bad candidate for an office network.

[1] We define the term backhaul network to be the core of the network and edge or access network to be the subnetwork to which the client devices are connected.

Also, while a peer-to-peer network has only temporal connectivity as long as the computers are connected to each other (or within range of each other), an infrastructure-based network has a network of APs to provide fixed connectivity to the backhaul. Infrastructure networks are useful for medium to large scale deployments which need to cover a large area as well as serve a larger number of client nodes. Such a deployment is also useful, or required, for an environment with QoS requirements.

Wireless networks for backhaul

Wireless Mesh Networks (WMNs) has arisen as an interesting paradigm for backhaul networks [AWW05]. WMN provides a mechanism to provide backhaul wireless connectivity to APs (See Figure 2.1 for an example network). The ease of deployment, which is the main advantage of a WLAN, can now be extended to the backhaul from the level of the single-hop AP.

2.1.2 Advantages of wireless networks

This progress towards wireless networks for data networking addresses many drawbacks of wired networks [Gei99]. The main issue that wireless networks address is that of tetherless connectivity which brings about many advantages. We discuss some of these issues below.

Mobility

The mobility of an user is severely challenged in a wired environment with the user having to physically disconnect a device, move to the new location and connect the device again in order to access the network. The user is not able to access the network while on the move. This need is especially alleviated with the advent of devices like the Personal Digital Assistant (PDA) and mobile phones with data networking capability. The user may not just move inside a building but may also move in between them. Seamless connectivity not only between floors in a building is required but also between the buildings in the campus. A wireless network, due to its nature, can address this.

Cost

The installation costs of a wireless network, due to its lack of tethered connectivity, is lower than corresponding costs of installing a wired network. Also, a wired network, due to its physical nature of deployment of wires, consumes more time and human effort in the installation process.

Savings also accrue in long term costs. Organisations continually upgrade and modify their plans: new devices may be added to the network, office floor plans may be modified and employees may be added or removed. Each time the network needs to be remodeled. A wireless network is far easier to remodel as the only connectivity involved is the power supply (which is usually widely available). A wired network, on the other hand, requires far more effort in removing the old connections and rewiring the system.

Installation and rewiring

An installation of a wireless network requires no data cabling, other than connecting the wireless hub devices to the backhaul. Since this forms a substantial part of the installation effort, a wireless network is also quicker to install. A wireless network is also a suitable option while installing a network in areas with physical obstructions. If natural obstructions separate buildings in a campus then, a wireless solution is probably more feasible and cheaper.

As discussed with costs, rewiring of a wired network is costly and a wireless network proves more economical when considering modification.

The above issues have resulted in the increased use of wireless networks as far as last hop access is concerned.

Bandwidth advantage of wired networks

On the other hand, one advantage of wired network over wireless networks, till now, is the difference in available bandwidth. Wired networks, as mentioned before, can now provide Gigabit speeds at the desktop. Wireless networks are far behind as far as data rates are concerned. The most common IEEE 802.11 deployments, for last hop connectivity, have a data rate of 11 Mbps and the best rate in that standard is 54 Mbps. Data rates of hundreds of Mbps are only now being developed in wireless standards (for example, IEEE 802.11n [Gro09])

While this is a significant issue, there has been a constant shift to wireless networks and increased wireless usage. These facts show that users consider the advantages of tetherless connectivity to outweigh the bandwidth advantage of wired networks.

2.2 Motivation

Consider the campus scenario shown in Figure 2.1. The campus has a collection of buildings spread over an area. Each building typically has one or more floors with rooms and corridors.

Currently, in such a setup, most academic or office campuses already have some kind of internetworking setup. Typically an end to end network consists of the following elements: applications and services, the backhaul core network, the last-mile access networks and user devices. The user devices or clients are connected through the access networks to the backhaul network and access application services.

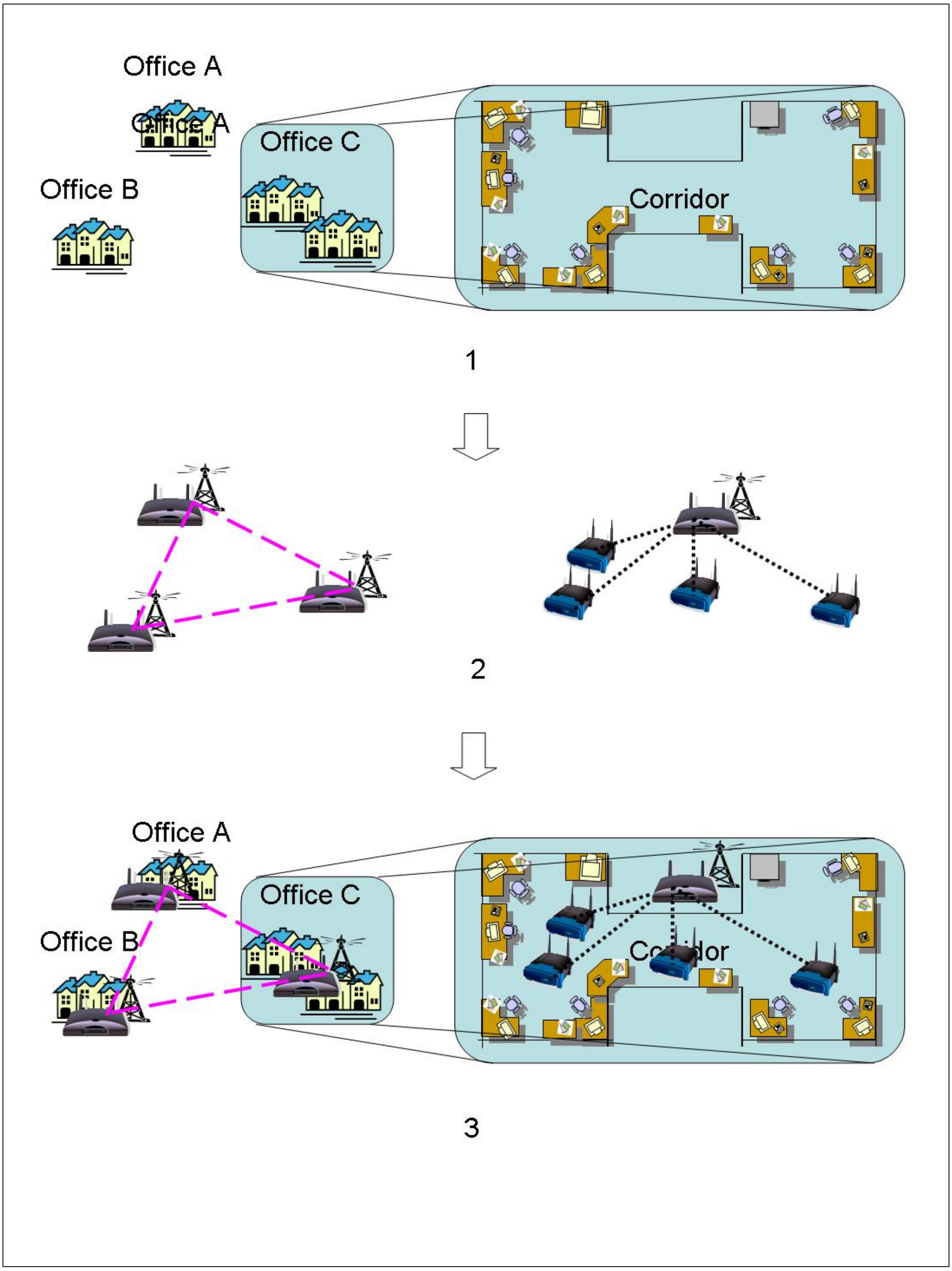

Figure 2.1: An example campus scenario: 1) Deployment scenario, 2) Suggested topology and 3) Topology deployment.

Now, in our example, each laboratory or office floor has workstations or other devices where Ethernet connectivity is provided [Eth05]. This forms a Local Area Network (LAN) in a building and all the LANs are connected together using a Wide Area Network (WAN). This is usually how wired networks, in an office environment, have been deployed. Ethernet data speeds up to 1 Gbps are now easily available at the desktop, which allows a user to access the network at high speeds.

In this section, we first discuss how networks are deployed and the issues in wireless deployments. Next, we discuss current approaches towards wireless network design, present their drawbacks and state the need for an integrated capacity-constrained design approach. Using that discussion, we state our motivation to study the following three issues:

1. Provisioning of 802.11 WLANs in heterogeneous application scenarios.
2. Capacity-constrained design of wireless networks.
3. Minimisation of network infrastructure cost.
4. Integrated design of local area and backhaul wireless networks.

2.2.1 Deploying a network

Coming back to our example, consider that a wired network is deployed in the campus. The following steps have to be taken. First, for each building, a study is done of floor plans, the type of devices to be deployed (client devices, servers and others) and the application scenarios. A LAN network topology is then constructed to satisfy these constraints. A similar study needs to be done for the campus layout itself. Each building's data networking needs have to be studied and the campus WAN topology is built. Together, the issues addressed to build a network is defined as the *network design problem* (NDP).

The network needs to be "designed" for two reasons: 1) devices have to be physically connected and 2) the network has to be provisioned based on application requirements. This problem is simple when considering a LAN. As mentioned before, wired network speeds at the client device are greater than what is currently required and high speed switches to connect these devices are available [Tan02]. This results in bandwidth provisioning being a non-issue in LANs. The design of a LAN is then an issue of physically connecting the devices together, considering information on floor layout and devices, to construct a topology. But, design still remains a relevant issue considered at the level of a WAN as provisioning here still remains an important issue (as this network connects and serves many LANs) [PM04].

Suppose that this wired campus network is to be replaced with a wireless network (for one or more of the reasons mentioned in Section 2.1.2). The inputs to the design problem remain the same as mentioned above for wired networks: the layout information of buildings and campus (called deployment layout), the devices deployed and their application scenarios. Additionally, the network design problem now has to address a set of issues due to the wireless nature of the network. We elaborate on this in the next section.

2.2.2 Issues in wireless deployments

The tetherless connectivity of a wireless network brings in some issues along with its advantages. Consider the case of a single floor in a building. First, the wireless nodes in the deployment area (be they client or infrastructure nodes) have to be "connected" to each other in order to form a network. Second, on being connected, the network has to provision bandwidth for application scenarios. The first is defined as coverage issue and the second as capacity issue. Also, other issues are the application scenarios, the various access technologies used and the cost of the system. We elaborate on these issues below.

Coverage

Coverage is defined as the study of RF characteristics of the deployment area and computing the number of infrastructure nodes (APs) required to adequately cover the given area. Coverage is a problem that does not occur in wired networks and is unique to wireless. It is dependent on the physical properties of the deployment area (properties of materials used, physical obstructions and others).

Approaches taken to address this issue are to study the deployment area, its radio frequency (RF) properties and a tentative deployment of wireless nodes to compute or estimate the coverage. These issues together are called the RF planning of a wireless network. The coverage of a wireless network is not a static property like for a wired network and can be temporal in nature. Physical changes in the deployment area (remodeling) or movement of objects (mobility and fading effect) can affect the coverage. Changes in coverage then affects the topology of the wireless network as nodes may go out of transmit range of each other. Hence, this requires careful design is required. We discuss these issues in detail while discussing network planning in Section 2.3.

Capacity

The capacity of a wireless network depends on coverage and wireless technology used. First, because coverage is non-static and nodes can be mobile, capacity of the network can also change. The network, therefore, has to be carefully provisioned.

Next, nearly all wireless deployments for last-hop client access are IEEE 802.11 WLAN deployments. Hence, a basic assumption of last-hop 802.11 access has to be made. The main mode of access in the IEEE 802.11 standard is the Distributed Coordination Function (DCF) which uses a Collision Sense Multi-Access with Collision Avoidance (CSMA/CA) to access the wireless channel [80203a]. This ensures equal opportunity access to all nodes in the system. While this makes the standard simple to implement and use, it is difficult to provide any kind of bandwidth provisioning on it. Other techniques in the standard like quality of service extensions help in provisioning but none of them have found any wide usage [80205]. Hence, if the standard has to be used, intelligent provisioning needs to be done to improve the capacity.

Also, the issue of interference and channel allocation has to be considered. In a wired network, switches are used to segregate the different parts of a network such that they can simultaneously function. A wireless network on the other hand acts as a hub where all nodes can listen to each other (and communicate with each other). The advantage of this is the simplicity of the access mechanism. The disadvantage is that they can then interfere with each other and bring down the capacity of the system (interference). While multiple channels can be used to alleviate this issue but this is one of the reasons why the capacity of a wireless network is less than a wired network.

We study the issue of capacity provisioning in 802.11 WLANs for homogeneous and heterogeneous application deployments in Chapters 4 and 5. In Chapter 7, we study capacity provisioning in wireless mesh networks.

Application scenarios

Provisioning a network needs to consider the application scenarios of each client in the network. The application scenarios in a wireless network, for the sake of classification of the network, can be broadly divided into voice and data applications. Each network is designed with parameters based on its main application scenario. For example, cellular networks is concerned with providing voice connectivity with user mobility over long distances. But, there has also been a distinct convergence of data and voice networks as the application scenarios themselves become heterogeneous in usage. Carrying data over cellular networks is an important issue as is carrying voice over data networks. In our case, wireless data networks, it becomes important then to study joint deployments of realtime applications like voice and video along with non-realtime applications.

Heterogeneous networking technologies

IEEE 802.11 WLAN have become the standard for last-hop access in wireless data networks. The client devices, in infrastructure mode, are connected to the network through access points (AP) [80203a]. These APs in turn are usually connected to the backhaul of the network using wired networking. This is mainly due to the single-hop infrastructure mode of access in the IEEE 802.11 standard and hence, only wireless connectivity in the last-hop is provided.

Increasingly, the "backhaul" of the network is also being replaced with a wireless network. Modified versions of IEEE 802.11 have been used for backhaul networks with wireless mesh networks emerging as an important paradigm [AWW05]. Also, other standards like IEEE 802.16 have emerged to provide backhaul connectivity [80204]. As wireless deployments become heterogeneous in nature, different types of access technologies may be used for last-hop client connectivity and backhaul network. Studying this issue and provisioning across these networks then becomes important.

We discuss the issue of heterogeneous wireless networks in Chapter 6 and 7 where we investigate the design of wireless networks with WLANs as access networks and mesh networks as backhaul. In that context, we study the issue of capacity provisioning and topology design.

Cost

Finally, as with any network deployment, the cost of a deployment also has to be minimised. The cost of a wired network involves the cost of the infrastructure[2] and the planning involved. The costs in a wireless network are not just the fixed costs of node deployments. A node may have one or more radio links. The links deployed have a fixed cost of deployment and a variable cost which depends on the transmit power used. This cost becomes a significant part of total costs when considering medium to long range transmit distances (for example, backhauls).

We study the issue of network infrastructure cost minimisation in design of wireless mesh networks in Chapter 7.

2.2.3 Current approaches towards wireless network design

WLAN networks, especially IEEE 802.11 networks, have generally been deployed using simple rules of thumb. An ad hoc planning for setting up an infrastructure-based wireless network may take the following steps [All99]:

1. The number (and type) of users (or client devices) are considered.
2. The number of clients that can associate with an AP is then calculated based on thumb rules. Based on client usage requirements, each AP can support a fixed number of client devices. Also, different types of users may have different requirements.
3. The number of APs required is then calculated.
4. The network is deployed.

Additionally, one or more of the following steps may be taken.

[2] Only the cost of the infrastructure nodes which are used to connect these clients are taken into consideration (like switches, hubs and cables) The cost of client nodes or servers deployed in the system are usually not considered as cost of network deployment.

Site survey

A simple survey of the deployment layout is made before deployment [ZPK03]. This involves the physical inspection of the deployment layout and identifying obstacles (such as walls, doors) that may affect the coverage of the network. The positions of APs are then calculated and the network deployed.

Simulations

Simulators like OPNET Modeler [OPN] or ns2 [ns2] are suitable for rapidly configuring a network and test its performance. However, this deployment technique has drawbacks: firstly, the design process requires knowledge of the node, link and application QoS characteristics; secondly, a simplistic association of clients to APs (either manually or using some automated tool like rapid configuration in Modeler) may not optimally utilise the network. The designer has to then perform a significant number of simulations, analysing various scenarios, before arriving at a suitable deployment scenario. While simulators provide good support for design validation, this process is still a significant manual effort.

Test measurements

In this step, the network is deployed after a site survey [ZPK03]. The deployed network is then tuned using performance measurements. If certain areas of the deployment layout lack coverage, additional infrastructure nodes are added to the network.

Signal strength measurements

Signal strength measurements are used to measure channel interference and signal strength levels at various points in the deployment environment. The measurement of actual coverage of the network is done by placing temporary APs and measuring performance metrics such as signal strength, signal to noise ratio (SNR) and packet error ratio. These measurements can then be used to plan placement of APs in the network [Air, Net].

RF planning

RF planning is the measure of signal strength at various points in the deployment environment, often using simulations. There are various computer aided planning tools available for simulating radio propagation in the design environment [Spe, CIN, FGK+95]. These tools model the deployment of APs in an area and project the signal characteristics of the deployment. They can be used to simulate the signal strength and attenuation characteristics of the deployed APs and also their interference behaviour. Figure 2.2 shows an example RF plan for a two room office deployment with 6 APs. The advantage of this approach over test and signal strength measurements is that, the time required for planning is reduced by performing various AP placement scenarios without actual deployment. But, using such tools and generating an AP placement plan requires domain knowledge. We discuss the models used for RF planning and the tools further in Section 2.3.

Backhaul deployment

In most current WLAN deployments, the backhaul is considered to be a wired network. The client devices are connected with a single-hop wireless access to an AP. The AP is then further connected through a wired network. As we discussed in Section 2.1.2, the cost and network re-organisation issues in a wired network are present for backhaul networks too. With wireless backhaul networks, similar test and signal measurement approaches have been used for WAN deployments like mesh networks [EDXa, Mes].

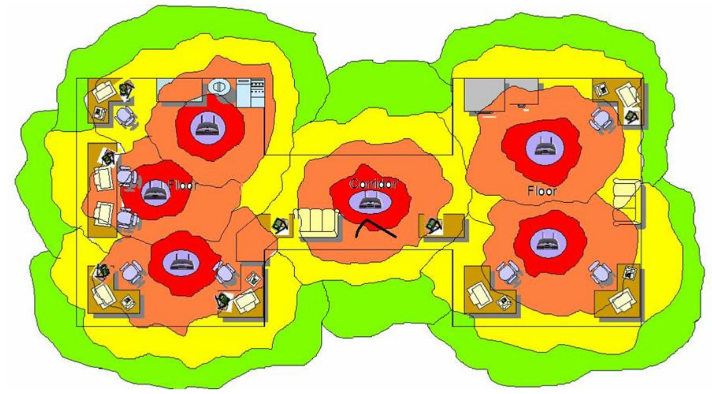

Figure 2.2: RF plan for an example office layout.

2.2.4 Drawbacks with current approaches

Ad hoc planning, site survey and test measurements are suitable only for deployment of small sized networks. Any large network cannot be deployed in such a manner when no guarantee can be given for required performance due to these rudimentary deployment strategies. As the network is deployed without any analysis on coverage or capacity constraints, any change required in network topology, in the form of addition or deletion of nodes, may prove costly.

Signal strength measurements and RF planning addresses the coverage issues by measuring (simulating) the RF characteristics of deployment environment and the signal strength at various points in the environment. But, the issue of capacity still remains. The number of APs deployed also depends on how the network is used. That is, the number of users in the network and their applications will determine some capacity constraint and in turn the number of APs required. For small scale deployments, such a calculation is usually based on thumb rules. The size of such deployments allow reconfigurations with minimal cost incurred. But, for large scale deployments, capacity constraints need to be looked at carefully.

Additionally, the IEEE 802.11 DCF mechanism is a standard for single-hop infrastructure based wireless connectivity. The DCF channel access mechanism does not provide any QoS guarantees and hence provisioning on an such access technology is a difficult task.

In summary, for a WLAN, designers do not plan their networks sufficiently by concentrating on providing only adequate coverage while ignoring the provisioning for sufficient bandwidth [McL03]. In general, for any wireless deployment, reconfiguration for large scale deployment maybe be costly and such a system has to be designed carefully. A rule of thumb approach may result in a sub-optimal system being deployed. Deploying APs without addressing this aspect of planning may result in the system topology being redesigned based on actual usage statistics. This then becomes a costly and time consuming process.

The above issues and drawbacks are present not only in WLAN deployment but also for the backhaul wireless networks. Study of a wireless backhaul network will also have the issues of capacity and coverage. The above approaches for coverage are also applicable for a wireless backhaul. The issue of capacity in *backhaul wired networks* has been widely studied and there exists a body of literature on provisioning in backhaul networks [PM04]. Similar studies for heterogeneous wireless networks are lacking. Additionally, in most current WLAN deployments, the backhaul is considered to be a wired network. The client devices are connected with single-hop wireless access to a wired network. Heterogeneous wireless access or the presence of wireless at the backhaul is not considered in the deployment process.

2.2.5 Need for an integrated approach for capacity-constrained design

Together, the issues discussed in Section 2.2.4 can be considered as the problem of wireless network design and any wireless network to be deployed has to address one or more of these issues.

Designers, till now, have been mostly considering the problem from a coverage point of view. This addresses only one half of the problem and networks are not designed with capacity issues in mind. Next, in 802.11 networks the issue of application deployments becomes important due to the limitations of the access mechanism and the lack of good provisioning techniques. And finally, the consideration of backhaul wireless networks and their interaction with the WLAN access networks have to be studied. Any one or more of the above issues may result in sub-optimally provisioned networks. Both these affect the cost of deployments.

From the discussion in the previous section, we then state the following four issues as our motivation:

1. Provisioning of 802.11 WLANs in heterogeneous application scenarios.

2. Consideration of a capacity-constrained approach towards designing wireless networks.

3. Minimisation of network infrastructure cost.

4. Integrated design of local area and backhaul wireless networks.

We discuss these issues further when we define our problem in Chapter 3. Now, in the rest of the chapter, we survey the various network planning approaches in wired and wireless networks detail.

2.3 Network planning in wired and wireless networks

Planning encompasses many different areas of networking. Planning, or design, involves aspects ranging from tools for designing RF plans for node deployment and measuring signal strength to optimization techniques for AP placement and topology construction. The issue of planning has been studied for wired and cellular networks. These problems, due to their similarity, have in turn been adapted from the field of operations research [PM04, GMW00]. Many of these techniques can be used in our case for capacity-constrained design of wireless networks. But, wireless networks involve certain issues, especially the channel access mechanism for WLANs and the characteristics of the wireless medium, which necessitates the modification of these techniques to suit the needs.

In this section, we discuss related work and wireless network design problems in related areas. We first discuss a simple wireless network model and construct a generic network design problem for it. Our design problem definition is useful in understanding the basic approach towards topology design and is general enough for us to adapt it to different classes of network design. We use this problem as a base case for constructing our network design problems. We then survey the work done in wired and cellular networks and provide a overview of general planning issues. We then survey the work done in coverage and capacity planning in WLANs. Other issues, like power control, which occur while discussing design are also presented. Then, we discuss the issue of design of wireless backhaul networks and especially wireless mesh networks.

2.3.1 Wireless design: A generic approach

We first discuss a generic network model for wireless networks. Our model is generic enough to capture local and wide area networks (LANs and WANs). We assume some basic information on the network deployment (like nodes, node properties). We then present a capacity-constrained design problem for generating logical topologies of a wireless network.

Network model

Consider the example wireless network shown in Figure 2.3. This network consists of two distinct categories of network elements. One category of network elements are the *client nodes*. Client nodes are nodes representing the application users of a network like workstations and notebooks. Server workstations in the network, for the purpose of this discussion, can also be considered as client nodes. The second category of nodes are the *infrastructure nodes*. These nodes connect the client nodes to form the topology and are put in place with the purpose of aggregating and transporting traffic for client nodes (including servers). Nodes like APs, routers or switches are examples of infrastructure nodes.

We assume that the following basic information on properties of the network to be deployed is available:

- The number and type of client nodes deployed.
- The characteristics of the client nodes. Characteristics are properties such as, number of links, bandwidth constraints, application traffic scenarios and other node properties.
- An abstract representation of the deployment area.

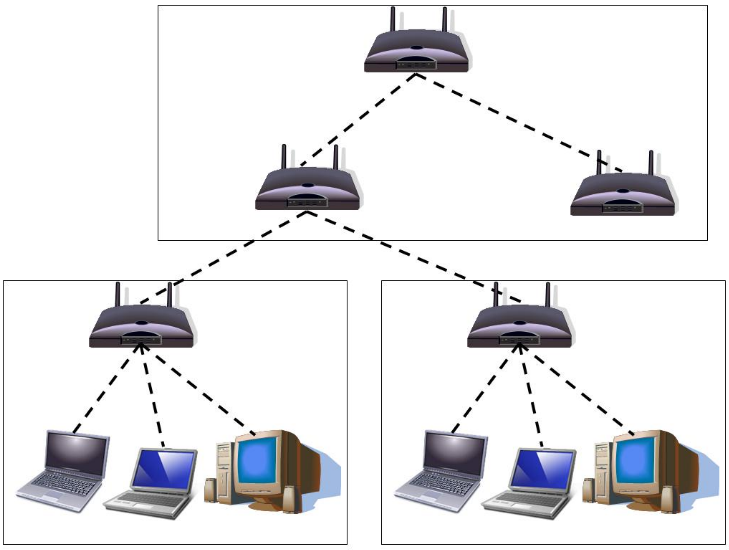

Figure 2.3: An example wireless network.

Given these properties, the aim is to generate appropriate logical topologies of network. We now state the generic design problem for topology construction.

Design problem statement

The design problem formulation is along the lines of the design heuristic defined for cellular networks by Jabari and others [JCNK95]:

Network design problem (NDP)

Given client nodes to be deployed, their characteristics and deployment layout,

Construct network topology,

Subject to capacity constraints,

While minimizing network infrastructure cost.

A **client node** is an end-user network element. The client nodes is a set of network elements to be deployed in the network. We define a client node's application scenario as its characteristics. The traffic properties of each application running on a node forms the application scenario. The **deployment layout** is an abstract deployment layout map (in the form of a graph). The layout also contains the position of nodes. We define **capacity constraints** as the collective demand constraints placed on the network by client nodes, which have to be satisfied. Additionally, we impose a cost constraint that the network infrastructure cost, while generating the topology, is to be minimised. This cost, also called deployment cost, is the total cost of infrastructure deployed in the network. Cost of deploying infrastructure nodes like APs, backhaul mesh nodes and the wireless links placed in the system to connect these nodes and construct the network topology form the **network infrastructure cost**.

To illustrate the above problem with an example, consider the deployment shown in Figure 2.4 (a concise version of Figure 2.3). Figure 2.4(1) shows the deployment layout of both the buildings and the offices (floor plan). Also shown are the client nodes to be deployed in each office (on R.H.S.). Each of these client nodes is assumed to have some applications running on it, for example VoIP-enabled PDA phones or FTP + HTTP applications on a Workstation. The capacity constraints now are the constraints placed on the network topology to be constructed and is a collection of the application demand constraints of all client nodes. An example network infrastructure topology is shown in Figure 2.4(2) for both the network backhaul (L.H.S.) and the access network (R.H.S.). The network infrastructure cost, to be minimised, is the cost of deploying these infrastructure nodes (and the links connecting these nodes).

This problem definition is general enough for us to adapt it to different types of wireless networks. We show in the rest of this chapter how this definition can be adapted for design of both local and wide area wireless networks (WLANs and WANs).

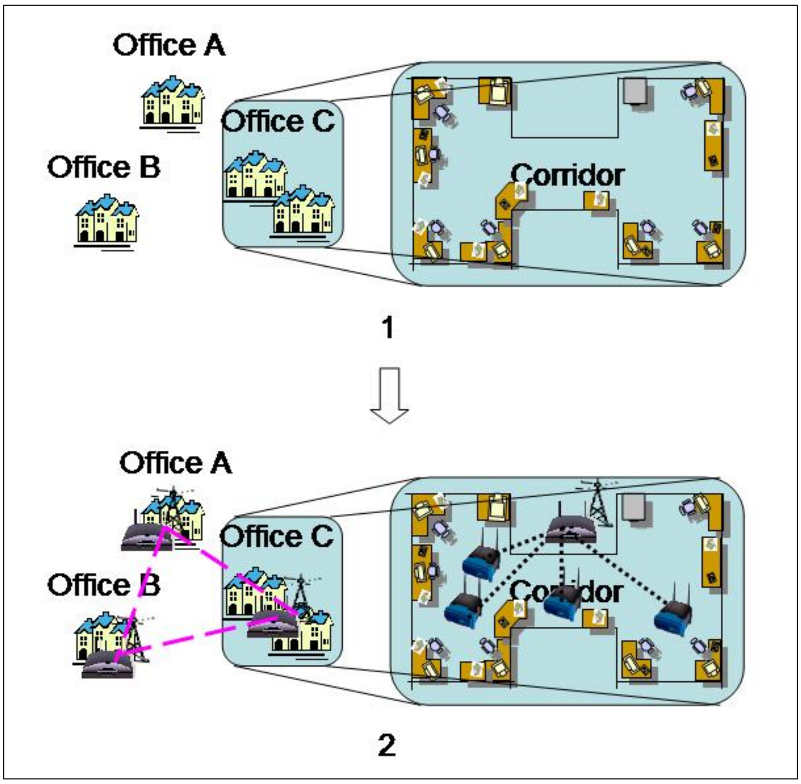

Figure 2.4: An example campus scenario (concise): 1) Deployment scenario, 2) Suggested topology and 3) Topology deployment.

Since, the issue of planning is well studied in wired networks, we survey it first before discussing the issue of design in wireless networks.

2.3.2 Planning in wired networks

Location and topology design problems have been studied for wired networks for some time. Most problems in this class are *dimensioning* problems where nodes (and links) are already present and installed in the network. The cost of the network then depends only on the capacity assigned to links.

At early stages of network design, decisions must be made on where to install network nodes and links interconnecting the located nodes. This must be done on an economical basis, that is, it should also take into account the *fixed* costs.[3] Since

[3]Here, fixed costs are the cost of establishing node locations and links.

the location of nodes and links determines the topology of the resulting network graph, these problems are called topological design problems. There are two types:

- Location problems: where demand is not considered (might not be known) and an assurance on a certain degree of connectivity is provided.
- Demand volume plus location problems: where volume is considered along with capacity dependent costs. The positions of nodes (and links) are determined simultaneously with flow and capacity allocation.

Pioro and Medhi have detailed out this entire class of problems and describe various solutions for wired network scenarios in [PM04].

For the basic node location problem the following information have to be defined first: 1) The cost information for connecting each area to each possible node location, 2) Any node site, if chosen, can handle up to a certain number of areas and 3) Each area is connected to only one node.

The node location problem can then be stated as: Connect N access regions or areas through a list of potential node locations M such that total cost is minimised. This design problem belongs to a class of network design problems that has been studied for various cases of wired as well as cellular networks. It falls under the class of topology computation and optimal node location problems.

2.3.3 Planning in cellular networks

A communication network, for telephony or data traffic, can be defined as a topology of networks made up of different nodes connecting a user to his data source [Tan02, PM04]. Planning has been studied for wired networks for the issue of design of backhaul connectivity. Medhi and Pioro study the various approaches in topology design in computer and communication networks [PM04]. They primarily address the various network models, design problems and optimization algorithms used in planning of backhaul networks for IP, ATM, digital switching and optical transmission.

Cellular network design has been studied and used for various fixed and mobile services. These include, GSM, UMTS and CDMA technologies [Rap02]. Whitaker and Hurley provide a survey on planning in wireless communication systems [WH03]. They cover approaches for network design, models and techniques for automated planning in 2G and 3G cellular networks. Planning for a cellular transmission infrastructure generally falls under the category of the facility location problem [PM04].

Given information such as node locations, network element specification, traffic capacity model and propagation models, the aim in cellular networks is to optimally design cells (organisation of cells, cell size) and assign frequencies such that adjacent cell interference is minimised. While cellular networks are designed with coverage, number of users per cell and power consumption in mind, WLAN design is basically for data traffic capacity and compatibility with LANs. In general, wireless networks has evolved into voice-oriented and data-oriented networks and their goals differ [PK02]. Cellular networks differ from wireless networks in channel characteristics, access mechanisms and capacity requirements.

2.3.4 Coverage and capacity planning for WLANs

Wireless networks based on the 802.11 standards [80203a, 80203b] continue to proliferate, with widespread deployments in many domains like hotspots, local area networks, homes and public venues like conferences [Var03, YPC05, Kap02].

The successful deployment of 802.11 WLANs has demonstrated that internet/intranet connectivity remains the primary application driver. With the increase in capability of the network elements, there are many network applications that demand throughput provisions, such as multimedia, collaborative work and distributed applications. The high bandwidth demand of such applications makes infrastructure support a necessity.

Currently, planning for wireless networks is done either in an adhoc manner or through complex site survey with emphasis on RF propagation studies [ZPK03]. For example, in an 802.11 network the APs are distributed regularly across the site. The distribution of APs is uniform or based on RF propagation studies. The fixed clients are also uniform in their distribution, while mobile clients are assumed to be located in a random manner.

A sub-optimal, and naive, approach is to place the AP nodes in a regular grid topology. This is simple to implement and can be used as a reference point for other coverage planning techniques [KU02].

While such an approach is valid for small networks, providing one-hop backhaul connectivity for all nodes and site surveys of large deployment areas restrict the scalability of such network planning techniques [YCG+03]. Also, such approaches look at wireless network planning from the angle of *coverage* rather than *capacity*[4] [KU02]. That is, the area in which the network is to be deployed is analysed for connectivity requirements rather than capacity. As discussed in Section 2.2, while coverage remains an important issue, with increasing demands on bandwidth, capacity requirements become important too [PPK+00, KU02].

Hills discusses the general issues in designing a large scale WLAN network with emphasis on coverage [Hil01]. This is based on the work done for the the Andrews wireless network project [Hil99]. Kamenetsky and Unbehaun survey the various coverage planning techniques for positioning of APs [KU02]. The basic goal behind

[4]Coverage is defined as the area around a wireless node where its received signal can be decoded. By capacity, we mean throughput requirements or bandwidth demands.

these techniques is to maximise the coverage area of each AP and to increase the signal quality.

Various optimization techniques have been used to solve the coverage planning problem. Most optimization algorithms take the approximation route to reduce the complexity of the problem. Some of the techniques used are pruning, neighbourhood search and simulated annealing. Pruning [KU02] is a greedy approach which starts with N APs, one for each available AP deployment site. The algorithm iteratively removes each AP and the objective function is re-computed. The AP removal causing the lowest objective function is then permanently removed and the algorithm repeats. Pruning can also be used to provide a good starting solution for other techniques due to its simplicity. Neighbourhood search [PM04], or local search, is a stochastic heuristic which finds the local optimum of the objective function. Neighbourhood search requires a good starting solution and hence has to be used in combination with other techniques like pruning. Simulated annealing [OL96] is also a type of local search algorithm which reduces its acceptance probability of change in objective function until a local optimum is found.

Prasad and others discuss issues like achievable throughput, coverage and interference in planning WLANs [PPK+00]. They state that even with a high bit rate, the net effective throughput of 802.11 is reduced due to protocol overheads.[5] They observe a drop in range with increasing *constriction* of the office environment and also with increasing data rates. They also note that a network deployed with coverage as the main criteria has larger cells due to the lower number of APs but have low aggregate throughput. If the criteria was throughput instead, the cell sizes would be larger (still providing for coverage) while requiring more APs and hence may be an expensive implementation. Their observations are important to note for

[5] This is a well known result [Bia00] and we have also examined this in [RIG05]. We study the issue of capacity of WLANs and discuss provisioning in Chapters 4 and 5.

our work as we consider capacity as the criteria in network design.

Rodrigues and others also discuss the coverage problem in WLAN [RML00]. They formulate the coverage problem as an integer linear programming problem to maximise signal strength at each traffic demand point (or client) in the layout and solve for AP placement and channel assignment.

Adhoc networks

At the other end of the spectrum, Mobile Adhoc Networks (MANET) topologies are constructed *after* the network is deployed and is more of a best-effort attempt [CCL03, RT99, ACG04]. Once the nodes are deployed as such, research has mainly concentrated on routing issues [RT99]. By definition, no planning issues arise for these networks. Adhoc networks are more interesting as a class of multi-hop wireless networks and we examine that aspect in Section 2.3.7 when we discuss wireless backhaul networks.

2.3.5 Site survey and RF planning

There are two methods to perform coverage planning for WLANs, 1) site survey, and 2) RF planning to study AP signal propagation [ZPK03]. Site survey is a costly and time consuming process involving the measure of information to determine the number and placement of APs to provide adequate coverage. It involves the deployment of temporary APs in candidate locations to examine coverage. Tools to measure network parameters such as received signal strength, propagation delay, throughput and packet loss are usually provided as part of the wireless device vendor toolkits [Sym, Cis, Lin, Int] or freely available [Eth, Kis, Net, Wra, Air].

RF planning, or propagation modeling, is a simulation-based approach to decide AP placements in a designed environment. RF planning can be classified into two types: 1) Deterministic methods and 2) Empirical methods [Gol05]. Deterministic methods use wave propagation theory to simulate the physics of wave propagation in an environment. Ray tracing [Gol05], based on computational geometry, is one commonly used technique which, given the deployment layout characteristics and position of a wireless node, computes the signal strength (or the path loss) at different locations in the layout. Ray tracing involves significant computation effort and there are many ray tracing tools available [Spe, CIN, FGK+95].

Empirical methods [Gol05] use parameters from statistics collected from representative measurements (that is, data from similar scenarios) [iST96]. These methods are simple and fast to apply but due to approximation of parameters, deployment specific accuracy cannot be achieved.

2.3.6 Other approaches: Power control

Studies have also concentrated on adjusting the transmission range (that is, power control) for constructing a topology satisfying a given throughput [Hu93, HL86]. Jia and others [JLD04], attempted to create a QoS aware topology using power control. Even though they deal with meeting the overall QoS requirements, they do this by adjusting the transmission range of the nodes. Marsan and others presented a technique in [MCN+02] to optimise the topology of Bluetooth, which aims at minimizing the maximum traffic load of nodes (and thus minimizing the power consumption). Detailed discussion of such approaches is beyond the scope of this thesis.

2.3.7 Wireless network backhaul

To remove wired media from the backhaul of a wireless network, APs in a single-hop wireless network need a mechanism to connect to each other and to access the gateway wirelessly. In order to do so, we have to evaluate a multi-hop wireless network architecture. Multi-hop networks have been studied under the area of ad hoc networks and there are many results presented as far as self-organising networks are concerned [CCL03]. Adhoc networks have been developed for large scale military applications like thousands of sensor nodes or specialised civilian applications like disaster recovery and conferences. A common aspect in all these applications is the lack of infrastructure and self-organisation of networks [BCE05].

Recently, efforts have been made to look at multi-hop wireless networks for enabling wireless broadband access to build backbones for WLANs [KSK04]. The advantages of wireless networks with a mesh backhaul topology are that it brings all the benefits of adhoc networks. It allows multi-hop wireless access, support for self-forming and rapid reconfiguration of topologies. Also, the use of multiple radios as well as multiple-channels at the mesh and AP nodes considerably improves the capacity and range of the network [Gol05]. The multi-hop network also allows for spatial reuse of channels allowing for increase in system capacity. The use of directional antennas at the mesh router results in greater coverage due to increased transmission range [AWW05].

Wireless Mesh Networks (WMN) [Gro04, AWW05, BCE05], presents one approach to providing a multi-hop wireless backhaul. WMN has been suggested as a technology for applications like broadband home networking, community and neighborhood networks, and enterprise networking [AWW05]. Such mesh topologies are being used widely in urban and semi-urban scenarios to connect campus environments and apartment complexes. Some examples of deployments are the MIT Roofnet [Roo], Microsoft neighbourhood wireless mesh [Mic], mesh networks [Mes] and others [Mer].

WMNs, using upcoming standards like IEEE 802.11s [Gro08] and other non-standard implementations [Mes], are also anticipated to significantly improve the performance of ad hoc networks, wireless personal area networks (WPAN), and wireless metropolitan area networks (WMAN).

WMN is suitable for building large scale networks while keeping the application scenario requirements (that is, appropriate backhaul connectivity) in mind. While WMNs result in larger wireless coverage, the pitfall of this approach is increased channel contention due to wireless links [LBB04] and reduction in channel capacity [JS03]. Hence, network planning of such networks has to take the above factors into consideration. There has also been a lot of support towards standardisation of mesh networks: IEEE 802.16 standard, also called Wireless MAN, presents the formal specifications for deployment of broadband wireless metropolitan networks [80204]. This standard has also been adopted by the industry in the form of the Worldwide Interoperability for Microwave Access (WiMAX) standard [WiM]. Additional standards, similar to, or extending the IEEE 802.16 standard, have also come up:

- HIPERMAN: The High Performance Radio Metropolitan Network, an European effort designed to inter-operate with 802.16 [ETS05];
- WiBro: Wireless Broadband, a Korean standard compatible with the mobile version of 802.16, the 802.16e standard [80206];
- IEEE 802.11s: The unapproved extension to the 802.11 standard which presents the Extended Service Set(ESS) for mesh networking [Gro08];
- WiFire: WiFi for Rural Extension, a standard from CeWIT for long range rural wireless communication [CeW06]. WiFire takes an interesting approach by building a 802.16-like MAC on top of a WiFi PHY to take advantage of the license-free nature and easy availability of WiFi chipsets.

There have also been other implementations of WMNs, especially as attempts to build co-operative community networks [Roo, Mic].

Recently, So and Liang have attempted to address the issue of design of WMN by proposing optimization based solutions similar to our proposed approach [SL06a, SL06b]. They present a joint formulation for the optimal number and placement of relay nodes in a WMN by using Bender's decomposition and Lagrangian approaches but only look at the WMN design problem.

2.4 Comments

We discussed the issue of network planning for wireless networks in this chapter. We noted that current approaches typically:

- use thumb rule solutions for network deployment.
- look at planning from a coverage perspective:
- consider solutions for homogeneous networks (WiFi or WiMAX deployment).

As discussed in Section 2.2 thumb rule solutions for deployment of APs result in under-provisioning of APs and traffic demands of clients not being met. While deploying a WLAN, the issue of AP-assignment has to be looked at more carefully. In Chapter 5, we present various techniques to address the AP-assignment problem.

The above approaches are also inadequate given the evolving hybrid nature of networks, which take advantage of multiple system capabilities at different levels/stages in the network.

To conclude, this chapter primarily deals with the present day wireless network design, their limitations and an attempt to justify the primary motivation in our effort towards advanced wireless network planning and design.

The current approaches of design and related issues are discussed and a subset of relevant related work is considered herein. With exponential growth and rate of advancement with respect to introduction of newer technologies in the field of wireless networking, planning, design and deployment deserve special attention. With these issues as the baseline, and with an intention to fill the gaps identified in this chapter, we extract design goals in wireless network planning and define the design problem statement in Chapter 3.

Chapter 3

Problem definition and solution approach

In this chapter, we define the multi-tier wireless mesh network problem and outline the solution approach. In Section 2.4, we had discussed the issue of wireless network deployment and current approaches to wireless network design. We discussed the drawbacks of current approaches and the need for an integrated framework for wireless design.

In Section 3.1, we first discuss the problem scenario with an example to outline the issues involved. Then, in Section 3.2, we present the problem as a multi-tier wireless network design problem. We discuss the constraints involved in this problem and the various stages involved in Section 3.3. In Section 3.4, we discuss a framework for an integrated approach towards network design. The framework, called the Wireless Infrastructure Deployment (WIND) tool, is for capacity-constrained design of the class of multi-tier wireless networks.

3.1 Problem scenario

Consider the campus shown in Figure 2.4(1). The campus consists of a set of buildings. Each building has a floor plan as shown on the R.H.S. On each floor, there is a deployment of client devices like PCs and PDAs which require connectivity. Client devices may have demands originating and/or designated to any device inside the office, the campus or the outside world through the network gateway. These source or sink demands are to be captured as the traffic demands of a client.

The problem is now that of designing a wireless network to provide connectivity to client nodes in each office building in the campus as well as connecting the campus together. This network can be thought of as a collection of local area networks (LAN in each building) connected together and to a gateway for external connectivity (campus WAN). Instances of such networks are: a campus of offices, with a LAN in each office and the offices connected with a backhaul network; a residential apartment complex with a WLAN in each building block and a backhaul connecting the apartments together.

Based on the requirements, two distinct networks are to be built. One, the office WLAN which connects the client devices inside each building. And two, a backhaul network to connect the buildings to provide inter-office building and gateway connectivity. The constraints placed on the network are capacity and cost constraints. The demands of each client in the network forms the capacity constraints. The cost of infrastructure placed in the network form the cost constraints. Breaking down the capacity constraints we can state that, the WLANs in each building must have adequate number of APs to satisfy the client demands. Similarly, the backhaul network must have adequate connectivity and capacity to satisfy inter-WLAN and gateway traffic. Also, the cost of both WLAN and backhaul infrastructure must be minimised.

3.2 Multi-tiered approach to design: Problem statement

Building a wireless network for the above example can be thought of as a multi-tiered problem. In one tier lie WLAN networks that connect the client nodes. Here, infrastructure nodes (APs) enable connectivity for clients (see Figure 2.4 for example network). The WLAN is built to provide intra-office connectivity and a gateway to the backhaul for client devices. In the other tier lie the infrastructure nodes forming the backhaul Wireless Mesh Network (WMN). These infrastructure nodes (mesh nodes) are connected wirelessly to each other and to the network gateway and form a backhaul for WLANs. The backhaul provides inter-WLAN and external connectivity through the gateway. While the WLANs connect the client devices, the WMN connects the WLANs.

Figure 3.1 presents a flowchart describing an overview of our wireless network design problem. The flowchart depicts the three-stage design of WMNs. The aim of this design approach is to start with a minimal set of input parameters describing the network to be deployed and construct the network topology in various stages. The input parameters (shown as Input in figure) are information about the network elements to be deployed and their application properties, the network deployment scenario, the traffic demands, the capacity constraints imposed on the design problem and the optimisation parameters.

In the first stage, *Compute AP-client mapping*, the AP-client association is computed.

This problem, called the AP-assignment problem, is formally defined as follows (using the generic network design problem definition in Section 2.3.1):

AP-assignment problem

Given set of client nodes,

Compute number of APs required,

Subject to capacity constraints,

While minimizing number of APs.

In the second stage, *Generate WLAN Topology*, the WLAN topology is constructed. Here, we use the solutions to the AP-assignment problem from the first stage for computing the topology. Again, using the generic network design problem definition in Section 2.3.1, we can formally state this problem as:

WLAN topology design problem

Given set of client nodes and deployment area,

Construct WLAN topology,

Subject to capacity constraints,

While minimizing network infrastructure (number of APs).

In the third stage, *Generate WMN topology*, a backhaul topology is constructed (for the WLANs constructed in the second stage). Using the generic network design problem definition in Section 2.3.1, we can formally state the mesh network design problem as:

Mesh network design problem

Given deployment layout, AP nodes deployed and their characteristics,

Construct backhaul topology,

Subject to capacity constraints,

While minimizing network infrastructure (mesh nodes and links).

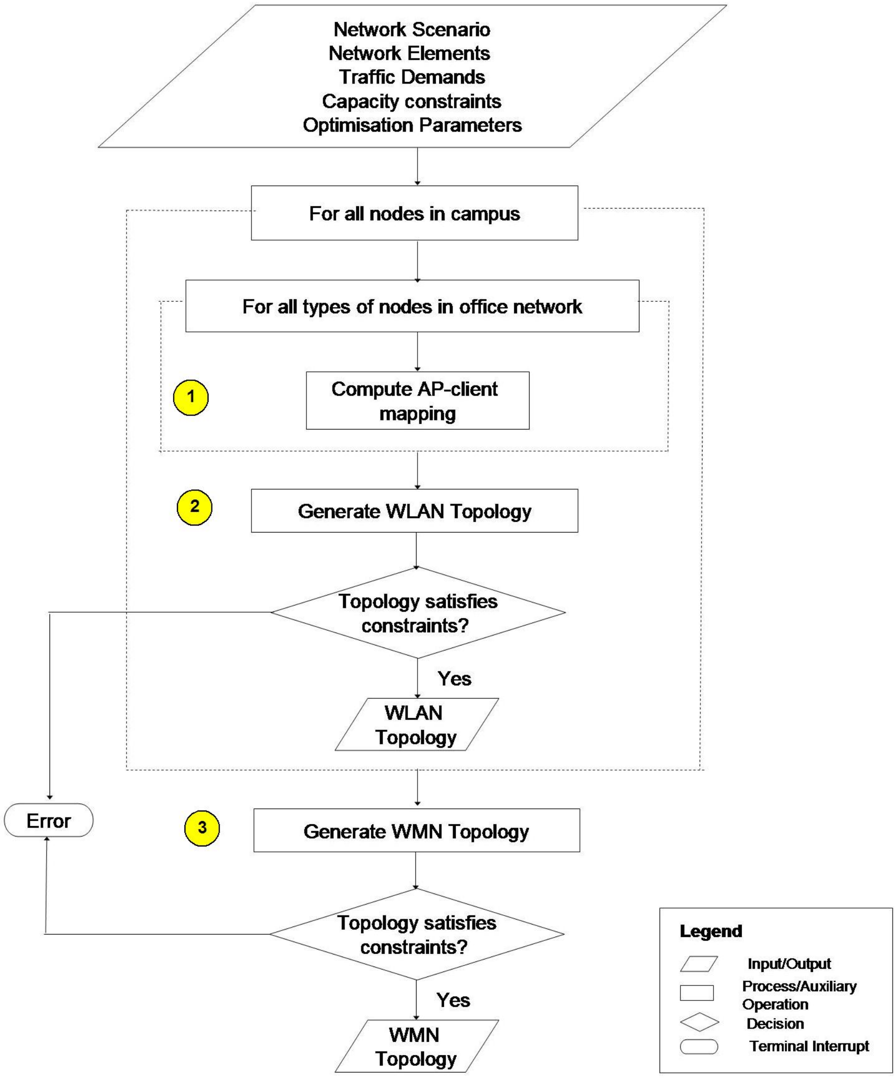

Figure 3.1: Design stages flowchart.

There are three design issues we consider at each stage:

1. **Capacity provisioning**: Provisioning for capacity is an important part of network design. Any realistic deployment of applications in a network requires some guarantee on performance. In the scenario described above, capacity has to be provisioned at all three stages: 1) While associating clients with APs, 2) While constructing the WLAN topology and 3) While constructing the backhaul.

2. **Cost minimisation**: In every stage in the network design, minimising network deployment cost is an important design issue. The deployment cost of a network involves the cost of deploying infrastructure nodes and links in the network.

3. **Integrated network design**: The network model under consideration is a WMN as the backhaul network connecting infrastructure mode WLANs to the gateway. We consider the design of a network from the stage of client association with AP to constructing a WLAN to constructing the backhaul mesh topology. Each stage is inter-connected with each other with respect to system requirements and constraints. Hence, our design approach takes an integrated view of the design problem.

We now outline our solution approach.

3.3 Design stages of solution

As stated before, building a wireless network for the above problem has three distinct stages (shown in Figure 3.1):

1. AP assignment problem: construct the association of clients to APs;

2. WLAN topology construction: construct the WLAN topology for each office in the network. Each of these WLANs may be different from each other based on client device demands in each office;

3. Backhaul topology construction: construct a backhaul network topology to connect the WLANs as well as provide gateway connectivity.

We now elaborate on each of these solution steps.

3.3.1 AP assignment problem

As noted before, the network is constructed with capacity satisfaction rather than coverage as a constraint. As mentioned in Section 1.2, even when all clients can connect to APs, sufficient APs have to be provided such that individual client traffic requirements are satisfied.

The first stage involves the study of the AP-client association in infrastructure IEEE 802.11 WLANs. This problem, defined as AP-assignment problem, involves the study of AP-client mappings such that individual client capacity requirements are satisfied while keeping characteristics of the wireless channel in mind.

In an infrastructure mode deployment, client nodes access the network through APs with a single-hop. The capacity of an AP, in an infrastructure WLAN, can then be defined in terms of the number of client nodes that associate with it. We investigate the AP-assignment problem for homogeneous and heterogeneous client application deployments. The maximum number of clients connected to an AP now depends on the type of clients associated with it. The client node type is characterised by its application scenarios. Homogeneous clients run a single application scenario while heterogeneous nodes may run many different applications.

In Chapter 4, we first analyse the capacity of 802.11 WLANs for homogeneous application deployments using real-time applications (with voice and video as examples). Using heuristics for calculating the backoff time in the DCF mechanism, we compute theoretical capacities in terms of number of flows and compare these with simulation results. Our aim in this experiment is to show that the DCF mechanism performance is adequate only in homogeneous environments.

In Chapter 5, we then extend the AP-assignment problem to heterogeneous deployments and analyse joint deployment scenarios of realtime and non-realtime applications. For heterogeneous application traffic scenarios, we first need to show that DCF performs poorly without additional functionalities for assisting priority traffic. We then present novel techniques to improve upon DCF performance. First, we show with simulations that a simple extension to DCF, using separate contention windows, can be used to tackle this issue. We then discuss an approach for solving the AP-assignment problem called sub-optimal application deployment. This approach uses prioritisation of applications scenarios as a mechanism for improving the system utilisation. We show that this approach results in significant improvement of the number of heterogeneous flows in the system and also improves system utilisation.

3.3.2 WLAN topology construction

The office WLAN topology is built to provide client devices with intra-office connectivity and a route (through some gateway node) to the backhaul network to provide inter-WLAN and external connectivity. The AP assignment solutions computed before provides the starting point for the construction algorithm. The topology construction additionally depends on factors such as:

- the layout of the deployment area;
- the node deployment strategy of each type of client device.

We discuss WLAN topology construction in Chapter 6. We present an approach to capture the abstract representation of the deployment layout as a graph. Each node in this graph represents an area in the deployment layout, where client nodes and APs are deployed. The deployment layout as well as the client devices deployed (and their demands) may be different for each office. Client device deployment strategy is captured by making use of the density of users in each node in the graph.

This requires making use of the AP-assignment solutions and adding network infrastructure to construct the logical topology of the WLAN. Two important considerations which enable this static generation of logical topologies are: 1) abstraction of wireless links using the AP-client mapping generated using the solutions to the AP-assignment problem and 2) abstract representation of the deployment layout as a graph. Using these, we construct logical topologies for WLANs using a recursive bottom-up greedy algorithm.

As inputs to the backhaul network construction stage, the WLAN gateway node assumes the aggregated inter-WLAN and external connectivity demands of the clients in the WLAN *underlying* it. This gateway node of the WLAN takes the form of a client node in the backhaul network.

3.3.3 Backhaul topology construction

Next, we design a mesh backhaul network topology to connect and form the backhaul network. This backhaul provides inter-WLAN connectivity as well as gateway connectivity to WLAN clients. Also, we constrain this backhaul in order to: 1) consider the client bandwidth requirements and 2) minimise the number of backhaul nodes/APs required to connect to the network.

The backhaul network takes the form of a wireless mesh network. The problem here is similar to the WLAN construction problem but differs in some crucial aspects:

- WLAN gateway (or root) node acts as a client in the backhaul network.
- Root node aggregates client demands outgoing from the WLAN.
- Position of potential mesh node locations are given. The design problem now involves finding a subset of the given set of suitable locations.
- Cost of both mesh nodes (routers) and links have to be minimised.

The cost to minimise is the deployment cost of mesh nodes and links. The constraints are:

- Demand constraints forced by the aggregated demands of client APs.
- Bandwidth constraints of the mesh links.
- Upper bounds on available mesh nodes and links.

In Chapter 7, we discuss these issues and present the optimization problem. We state the WMN construction problem as a node location and topology design problem. This WMN construction problem is an example of a facility location problem and requires a Mixed Integer Linear Programming (MILP) formulation. We use the CPLEX optimiser to solve this MILP formulation.

3.4 Wireless Infrastructure Deployment tool

We have developed a tool box, the Wireless Infrastructure Deployment tool (WIND), to implement the three stages in the WMN design problem. As discussed in Section 2.3.4 there are various tools available for coverage planning of WLANs. While

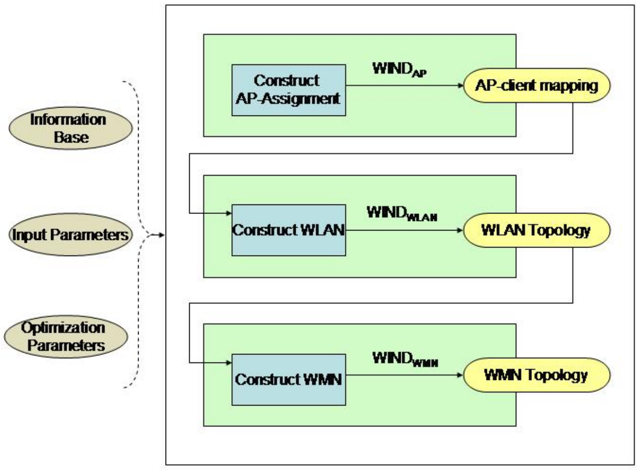

Figure 3.2: WIND framework for multi-tier mesh network design.

there are similar capacity provisioning tools for cellular networks, to the best of our knowledge no such tool exists for WLAN + WMN, especially in the open domain.

The aims of WIND are as follows:

1. Plan for capacity in the design stage.

2. Automate design process by computing topology of the entire network starting from client node demand requirements.

3. Ease validation process by using inputs and outputs in simulator readable formats.

Figure 3.2 shows the framework of WIND and it has three modules for: 1) AP-assignment, 2) WLAN topology design and 3) WMN topology design. The modules interact to generate the final topology of the network.

WIND(AP) generates the AP-assignment solutions. It considers heterogeneous application scenarios and takes as input client node and AP properties to generate AP-client assignments (Chapter 5). These assignments are used by WIND(WLAN) to construct the WLAN topology for each office deployment area (Chapter 6).

WIND(WLAN) takes as its input information client node demands and layout of deployment area. WIND also uses information on AP-client mappings from an information base. WIND uses a recursive bottom-up algorithm to construct logical topologies for WLANs. The client demands are handled as follows: 1) WIND attempts to build a topology to satisfy intra-WLAN demands and 2) WIND aggregates inter-WLAN and gateway traffic demands at the root node of the WLAN. The output is presented in a simulator-friendly format for ease of integration with a simulator.

WIND(WMN) then computes a WMN topology to connect the WLANs (Chapter 7). In this stage, a backhaul network is constructed with the root nodes of underlying WLANs as client nodes. The demands of the root nodes are the aggregated demands of the WLAN clients and form the capacity constraints. Locations of potential mesh nodes and number of available links are given, and the cost of network infrastructure is to be minimised. The design problem then is to compute the optimal mesh node locations and backhaul topology while satisfying capacity constraints. This problem falls under the category of facility location problems in optimization and is a Mixed Integer Linear Programming (MILP) problem. WIND uses an MILP solver to evaluate the potential mesh nodes and links and uses its output to construct the topology.

3.5 Comments

We have proposed a multi-tier wireless mesh network problem for design of WMN networks with WLANs as clients. We have considered an integrated network design approach and identified three distinct stages in this problem: 1) AP-assignment problem, 2) WLAN topology construction and 3) Backhaul WMN topology construction. The constraints imposed are capacity provisioning and cost minimisation.

We have also built a tool WIND that implements the design stages. To the best of our knowledge, the approach and the tool are novel and not found in open domain literature.

Chapter 4

Capacity of WLANs

As discussed in problem definition (Chapter 3), the first issue in design of 802.11 WLANs is to study AP-client association in WLANs. In an infrastructure mode deployment, client nodes access the network through APs with a single-hop. The capacity of an AP, in an infrastructure WLAN, can then be defined in terms of the number of client nodes that associate with it. The maximum number of clients connected to an AP now depends on the type of clients associated with it. The client node type is characterised by its applications. Homogeneous clients run a single application scenario while heterogeneous nodes may run many applications. Typical applications would be Voice over IP (VoIP) and web browsing.

We analyse the capacity of 802.11 WLANs for homogeneous application deployments using real-time applications (with voice and video as examples). We compute theoretical capacities in terms of number of flows and compare these with simulated results. We observe that the DCF mechanism is adequate only in providing some guarantees in homogeneous deployments and is inadequate in handling heterogeneous traffic. Many similar studies of capacity of WLANs has been done for various traffic scenarios [KMK05]. While this study is not singular in its presentation here, we use it as a base case for our analysis of heterogeneous deployments and for establishing the problem of AP-client association.

In Section 4.1, we first present a generic theoretical capacity equation for the 802.11 DCF mechanism. In Sections 4.2 and 4.3, we then compare the theoretical number of flows with simulation results, under different data rates, for various voice and video codecs. In the subsequent discussions in Section 4.4, we present the issues with DCF mechanism and discuss how presence of other application flows may affect the system performance.

4.1 Theoretical capacity

The *Distributed Coordination Function* (DCF) protocol in the 802.11 MAC specifications is a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism [80203a] and is a distributed random access mechanism. Figure 4.1 shows a fragment in the timeline of events during a data transfer in the 802.11 DCF MAC protocol. The figure shows, under steady load, DCF events during a data exchange between two nodes (a transmitter-receiver pair). At the start of the fragment, it is assumed that a previous transmission has just finished and the channel is sensed idle. At the beginning of the fragment, nodes with data to transmit start a *DCF InterFrame Space* (DIFS) timer. When the DIFS timers expire, each node enters a random backoff phase. In the backoff phase, to avoid contention, nodes pick a random number of backoff slots from their contention window (between 0 and CW). They then defer data transmission for backoff slot units of time. When this backoff is completed, a transmitter node attempts to transmit its data.[1] All other nodes, on sensing this data transmission, freeze their backoff counters and defer transmission. The transmitter node then transmits its data. At the end of the data transmission is a *Short InterFrame Space* (SIFS) that allows a receiver to

[1]In this discussion we do not consider the RTS-CTS handshaking mechanism for channel reservation.

Figure 4.1: DCF mechanism: Timing diagram for events during data transfer between two nodes in the 802.11 DCF MAC protocol .

receive the data and turn its radio from receive to transmit mode. The receiver then transmits an ACK packet.

If more than one node finishes its backoff at the same time, then a collision occurs on data transmission. The collision results in corrupted data packet(s) received by the receiver node(s). The receiver nodes then do not transmit an ACK after SIFS. The transmitter nodes infer that a retransmission is required after a timeout (called ACK timeout). These nodes then double their contention window (until a maximum value of CW_{max}) to reduce collision probability for the next transmission. All other nodes, not involved in the collision, continue with their residual backoff timers (which were frozen) for the next round of the DCF protocol.

We calculate the theoretical throughput by first calculating the time required to transmit one packet. The terms used in the calculation and the MAC parameters are given in Tables 4.1 and 4.2 respectively. The packet size (pkt) is equal to the payload of the application data packet (in our example, the voice or video codec) plus the overhead due to application and IP protocols (given in Table 4.3).

The time taken for transmission of a packet is determined by calculating the following:

1. time spent waiting to acquire the channel: DIFS + average back off.
2. time required for the actual transmission of the data along with its overhead (PHY, MAC and higher layer headers)
3. time required to transmit the acknowledgment + SIFS.

The calculation proceeds as follows (time is in microseconds (μS) and through-

Term	Definition
pkt	Packet size (at the MAC)
ACK	Size of ACK packet (14 bytes)
$MSDU_{max}$	Maximum size of MAC Service Data Unit (2048 bytes for 802.11)
$frag\_size$	Fragmentation size (in bytes)
r	Data rate (in $Mbps$)
$DIFS$	DIFS time (in μS)
$SIFS$	SIFS time (in μS)
$slot$	Slot time (in μS)
$backoff$	Backoff
PHY	PHY overhead (in μS)

Table 4.1: Terms used in theoretical capacity calculation.

Parameter (in μS)	802.11b	802.11g
Slot time	20	9
SIFS	10	10
DIFS (= SIFS + 2 * Slot time)	50	28
PHY preamble	192	20
Signal extension	-	6

Table 4.2: 802.11 b and g MAC parameters: timing, preamble transmission time and signal extension.

Overhead	Value (in bytes)
RTP	12
UDP	8
IP	20
MAC	34

Table 4.3: RTP, UDP, IP and MAC stack overheads.

put in megabits per second (Mbps)):

- time taken to send only the packet (without calculating ACK, PHY and backoff) is:

$$t_{pkt} = \frac{(pkt + MAC) * 8}{r} \tag{4.1}$$

- time taken to send ACK (size is 14 bytes) is

$$t_{ack} = \frac{ACK * 8}{r} = \frac{14 * 8}{r} = \frac{112}{r} \tag{4.2}$$

- total time taken is now equal to the time taken for DIFS, backoff, packet transmission plus its PHY overhead, SIFS and ACK transmission plus its PHY overhead:

$$\begin{aligned} t_{total} &= DIFS + \frac{backoff}{2} * slot + PHY + t_{pkt} \\ &\quad + SIFS + PHY + t_{ack} \qquad (4.3) \\ &= DIFS + SIFS + 2 * PHY + \frac{backoff}{2} * slot + t_{pkt} + t_{ack} \qquad (4.4) \end{aligned}$$

Now throughput is given as:

$$\begin{aligned} T &= \frac{Payload}{t_{total}} \qquad (4.5) \\ &= \frac{pkt * 8}{DIFS + SIFS + 2 * PHY + \frac{backoff}{2} * slot + t_{pkt} + t_{ack}} \qquad (4.6) \end{aligned}$$

Backoff, due to its random nature, is difficult to model. While there are various analytical models which try to accurately model the backoff behaviour in 802.11, we show that the above approximation is sufficient, for modeling purposes, by comparing theoretical results with simulations. This has similarly been shown by capacity calculations in [Sys04, MGFK04, WLL05]. The common denominator in all these calculations is the assumption on backoff value in order to simplify the capacity equation.

In equation (4.5), the 802.11 scheme used defines the data rate and timing values. The application scenario gives us the payload (pkt) and hence the time taken to send a packet. Given the 802.11 scheme and the payload, we can derive the throughput of a WLAN and also determine the maximum number of application flows. We show that now with an example.

4.1.1 Example: Number of G.711 voice calls in 802.11b

We calculate the number of VoIP calls in an 11 Mbps 802.11 WLAN using the G.711 codec [IT88]. The 802.11b timing parameters are given in Table 4.2. The G.711 voice codec packets are sent along with the RTP, UDP, IP and the MAC overheads. These protocol overheads are given in Table 4.3.

Now, the G.711 codec sends 100 packets/second in a duplex connection (50 in each direction). The bandwidth (b), required for one duplex call, is then given by: $b = 100 * pkt * 8 = 100 * (160 + 12 + 8 + 20) * 8 = 100 * 200 * 8 = 160000\ bps$. The number of possible calls is now given by $floor(T/b)$. The G.711 codec parameters are given in Table 4.4. The time taken to send a packet is now given from equation (4.1):

$$
\begin{aligned}
t_{pkt} &= \frac{(pkt + MAC) * 8}{r} = \frac{((160 + 20 + 8 + 12) + 34) * 8}{11} && (4.7)\\
&= \frac{234 * 8}{11} = 170.18 && (4.8)
\end{aligned}
$$

The time taken to ACK this packet is then (equation (4.2))

$$
t_{ack} = \frac{112}{r} = \frac{112}{11} = 10.18 \quad (4.9)
$$

Substituting into equation (4.5), time for packet transmission (t_{pkt}) from equation (4.7), time for ACK (t_{ack}) from equation (4.9), *pkt* = G.711 packet size + protocol overheads (from Table 4.3) = 200 bytes and *backoff* = 31, we get:

Parameters	Value
Bit rate	64 Kbps
Framing interval	20 ms
Payload	160 bytes

Table 4.4: G.711 voice codec parameters.

$$T = \frac{200 * 8}{DIFS + SIFS + 2 * PHY + \frac{31}{2} * slot + 170.18 + 10.18} \tag{4.10}$$

$$= \frac{1600}{934.36} = 1.712\ Mbps \tag{4.11}$$

The number of possible calls is now given by $\lfloor T/b \rfloor = \lfloor 1.712/.16 \rfloor = 10\ calls$.

4.1.2 Extension: Handling large payloads

The 802.11 MAC has an upper bound on MAC payload size (MSDU size). If the size of a higher layer packet is larger than the maximum MSDU size allowed by the IEEE 802.11 WLAN standard (2304 bytes) then such a packet will not be transmitted by the MAC and is discarded. To handle large packets, such as in the case of video applications, fragmentation is used to divide the application payload into manageable packets. The following changes are made to the capacity calculations to capture the effect of large packet processing.

In equation (4.1), for time taken to send a packet, *pkt* is equal to the application payload plus the application and IP protocol overheads given in Table 4.3. For large packets, if *pkt* is greater than $MSDU$ (maximum size of MAC Service Data Unit), the packet is fragmented. The packet fragments are of fragmentation size ($frag\_size$) where, $frag\_size \leq MSDU$ (see Table 4.1 for definitions). Changing *pkt* in the equation to pkt_{frag} ($= frag\_size$), the time taken to send the packet fragment is now:

$$t_{pkt_{frag}} = \frac{(pkt_{frag} + MAC) * 8}{r} \tag{4.12}$$

The total time taken per fragment is obtained from modifying equation (4.3) as follows:

$$\begin{aligned} t_{frag} &= DIFS + PHY + t_{pkt_{frag}} + SIFS + PHY + t_{ack} && (4.13) \\ &= DIFS + SIFS + 2 * PHY + t_{pkt_{frag}} + t_{ack} && (4.14) \end{aligned}$$

Note that the backoff is not counted in equation (4.13) as it is present only for the first fragment. Hence, we have to take care that in the throughput we add backoff only once (to the first fragment's time). The total number of fragmented packets to be sent is given by $frag\_num = \lceil pkt/pkt_{frag} \rceil$. The throughput (equation (4.15)) is now suitably modified:

$$\begin{aligned} T_{frag} &= \frac{Payload}{\frac{backoff}{2} * slot + t_{frag} * frag\_num} && (4.15) \\ &= \frac{pkt * 8}{\frac{backoff}{2} * slot + (DIFS + SIFS + 2 * PHY + t_{pkt_{frag}} + t_{ack}) * frag\_num} && (4.16) \end{aligned}$$

4.2 Voice capacity

We now use the analysis from Section 4.1 to determine the maximum number of voice calls in a WLAN for different scenarios (802.11 schemes and voice codecs). We then simulate the same scenarios and compare these results with theoretical results.

The different data rates used are given in Table 4.5. We simulate a total of 6 data rates, 3 each in 802.11b and 802.11g. The data rates are only a sample of the data rates provided in each scheme. Two common data rates, 1 and 11 Mbps, in both 802.11b and 802.11g allow us to compare the two schemes.

Theoretical calculations

We examine four voice codecs from the ITU standards, G.711, G.723.1, G.729 and GSM [IT88, IT06, IT07a, Sta94]. The codec parameters are given in Table 4.6. For each of these codecs, for each data rate and scheme, we compare the theoretical calculation with simulation results. The theoretical maximum number of flows (or calls) for the various schemes is given in Table 4.8. The detailed calculation of the theoretical number of calls using the throughput equations from Section 4.1 is given in Table 4.7.

Simulations

In simulations, we impose two constraints to define a valid voice call. One, throughput requirements should be satisfied. That is, no voice packet gets dropped.[2] Second, a delay guarantee is imposed as an upper bound on the total time taken for the voice call. We assume that the wireless link is the bottleneck in the delay analysis. The total end-to-end delay allowed for a voice call is 150 *ms*, as specified by the ITU standard G.114 [IT03]. As we simulate only the single-hop wireless link, a permissible *delay of 75 ms* is used as a heuristic.

The simulation was done using OPNET Modeler [OPN]. Each of the voice scenario was set up as an application definition and a voice user profile was created to model a voice call. Each simulation was run for 5 seconds. In order to determine

[2] It is possible though to build some amount of tolerance into the traffic in order to allow it to tolerate some loss of information. For simplicity, we consider only the above case.

Scheme	Data rate (in *Mbps*)
802.11b	1, 5.5, 11
802.11g	1, 11, 54

Table 4.5: 802.11 schemes used for maximum number of flow calculation.

↓ Parameters / Codecs →	G.711	G.723.1	G.729	GSM
Bit rate (in kbps)	64	6.4	8	13.2
Framing interval (in ms)	20	20	20	20
Payload (in bytes)	160	24	20	33

Table 4.6: Voice codec parameters.

the maximum number of calls we increased the number of flows until one of the above constraints failed.

4.2.1 Observations

The maximum number of voice calls for each scheme is shown in Table 4.9. The delay experienced by each scheme is given in Table 4.10. Comparing the simulation results with the theoretical calculations in Table 4.8, we observe that the simulation results closely follow the theoretical results with the theoretical results forming an upper bound on the number of calls. For ease of visualisation, Figures 4.2, 4.3, 4.4 and 4.5 compare the simulation and theoretical results for G.711, G.723.1, G.729 and GSM respectively.

At 1 Mbps data rate, the number of calls in both 802.11b as well as 802.11g are nearly the same. The effect of shorter timings in 802.11g is seen in the 11 Mbps data rate, where 802.11g outperforms 802.11b. With increase in data rates it can be seen that, for different codecs, the maximum number of calls do vary by a large extent. Also, there is not much difference between the codecs as far as the number of calls are concerned. This shows that contention in 802.11 CSMA/CA mechanism is the main limitation.

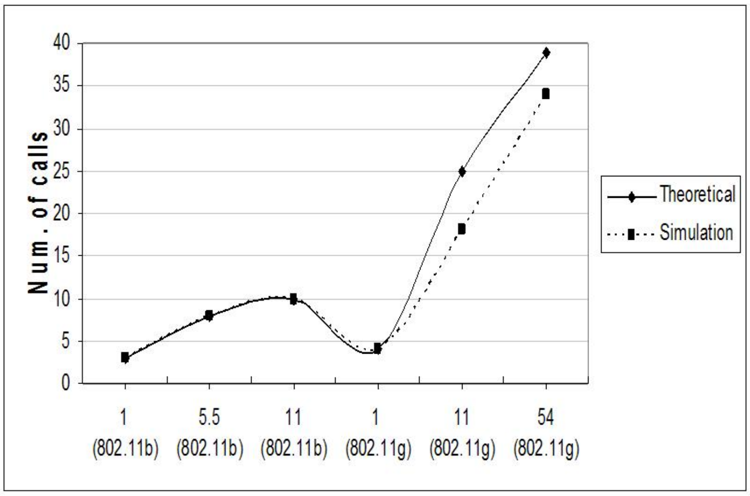

Figure 4.2: Maximum G.711 voice calls: theoretical vs simulation results.

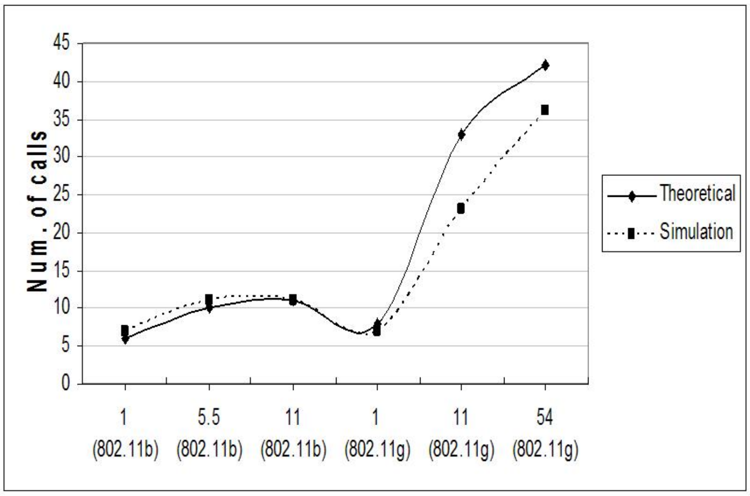

Figure 4.3: Maximum G.723.1 voice calls: theoretical vs simulation results.

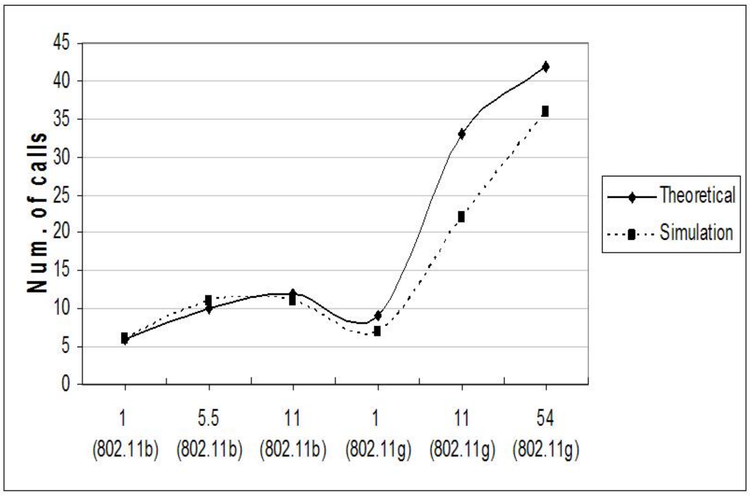

Figure 4.4: Maximum G.729 voice calls: theoretical vs simulation results.

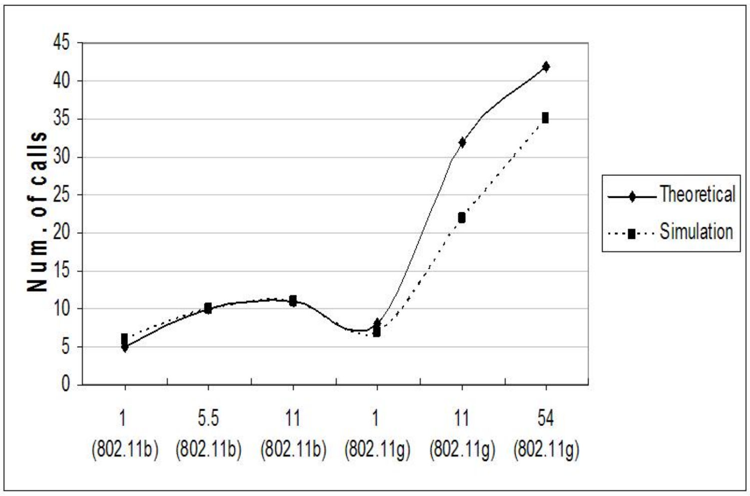

Figure 4.5: Maximum GSM voice calls: theoretical vs simulation results.

Scheme $\longrightarrow$ Data rate (r)		802.11b 1	5.5	11	802.11g 1	11	54
pkt (in bytes)	G.711	200	200	200	200	200	200
	G.723.1	64	64	64	64	64	64
	G.729	60	60	60	60	60	60
	GSM	73	73	73	73	73	73
DIFS		50	50	50	28	28	28
SIFS		10	10	10	10	10	10
PHY		192	192	192	20	20	20
backoff		31	31	31	31	31	31
slot		20	20	20	9	9	9
t_{pkt} (in μs)	G.711	1872	340.364	170.182	1872	170.182	34.667
	G.723.1	784	142.546	71.273	784	71.273	14.519
	G.729	752	136.727	68.364	752	68.364	13.926
	GSM	856	155.636	77.818	856	77.818	15.852
t_{ack}	(in μs)	112	20.364	10.182	112	10.182	2.074
Throughput (T) (in Mbps)	G.711	0.584	1.435	1.712	0.727	4.022	6.293
	G.723.1	0.310	0.558	0.613	0.460	1.713	2.187
	G.729	0.297	0.527	0.577	0.444	1.621	2.056
	GSM	0.339	0.628	0.694	0.493	1.912	2.481
Bandwidth (b) (in Mbps)	G.711	0.160	0.160	0.160	0.160	0.160	0.160
	G.723.1	0.051	0.051	0.051	0.051	0.051	0.051
	G.729	0.048	0.048	0.048	0.048	0.048	0.048
	GSM	0.058	0.058	0.058	0.058	0.058	0.058
Number of calls	G.711	3	8	10	4	25	39
	G.723.1	6	10	11	8	33	42
	G.729	6	10	12	9	33	42
	GSM	5	10	11	8	32	42

Table 4.7: Number of voice calls: voice capacity calculations.

Figure 4.6 compares the delay experienced by the various schemes. The delays experienced by the nodes in the system are far lower than the constraints imposed, with the largest delay being around 18 μs. Further delay can be tolerated in some cases if some data loss is acceptable. For example, in the case of G.711 codec under 54 Mbps 802.11g scheme, the maximum number of calls is 34 (Table 4.9). The theoretical maximum specified in Table 4.7 is 39 calls. Simulating this scenario we see that delay, while high, is still bounded (Figure 4.7). But, the throughput is not

Scheme ⟶	802.11b			802.11g		
Data rate (r)	1	5.5	11	1	11	54
G.711	3	8	10	4	25	39
G.723.1	6	10	11	8	33	42
G.729	6	10	12	9	33	42
GSM	5	10	11	8	32	42

Table 4.8: Maximum number of voice calls: theoretical results.

Scheme ⟶	802.11b			802.11g		
Data rate (r)	1	5.5	11	1	11	54
G.711	3	8	10	4	18	34
G.723.1	7	11	11	7	23	36
G.729	6	11	11	7	22	36
GSM	6	10	11	7	22	35

Table 4.9: Maximum number of voice calls: simulation results.

Scheme ⟶	802.11b			802.11g		
Data rate (r)	1	5.5	11	1	11	54
G.711	0.0086	0.0072	0.006	0.0086	0.0104	0.0187
G.723.1	0.0079	0.0094	0.0069	0.0068	0.0105	0.0126
G.729	0.004	0.0063	0.004	0.0044	0.0054	0.0138
GSM	0.007	0.007	0.007	0.0085	0.0088	0.0098

Table 4.10: Delay experienced at maximum number of voice calls (in seconds).

satisfied and nearly 20 percent of data (offered load) is dropped (Figure 4.8).

This is again because of the CSMA/CA contention limitation mentioned above due to which the system does not perform in an optimal fashion. In the next chapter we expand on this issue and present an extension to the DCF mechanism to increase the capacity of an 802.11 system.

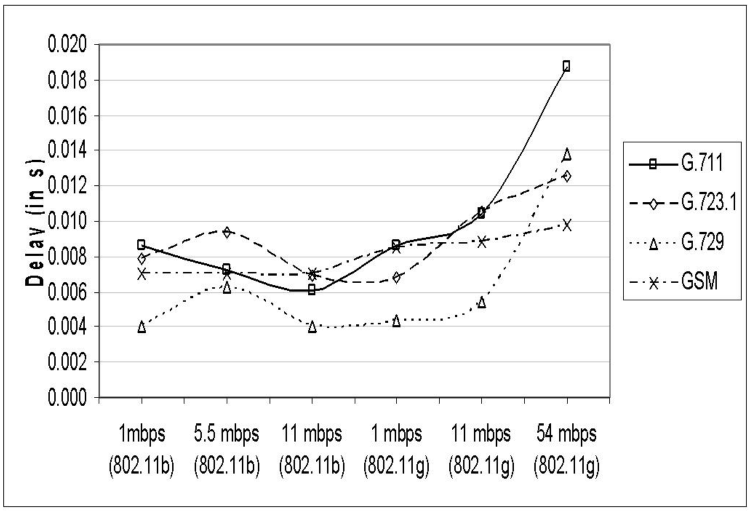

Figure 4.6: Delay for voice schemes in 802.11b/g.

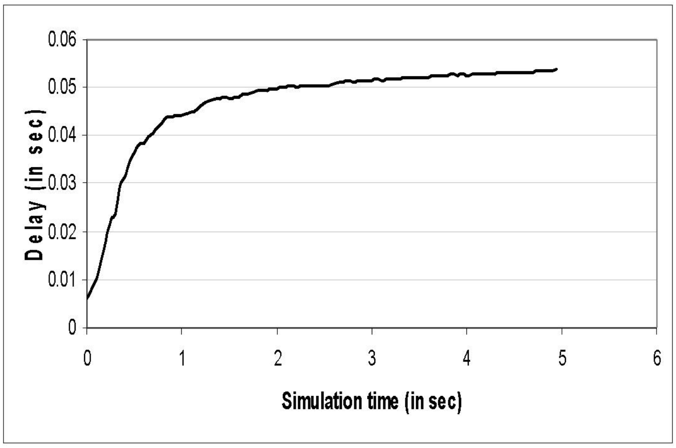

Figure 4.7: Delay for G.711, 54 mbps 802.11g - 39 call scenario.

4.3 Video capacity

The analysis for maximum number of video flows is similar to the voice analysis in Section 4.2. For different scenarios (802.11 schemes and video codecs), we determine the maximum number of video flows. We then simulate these scenarios and compare them with the theoretical results.

The data rates used are the same as in Table 4.5. H.264 is a standard for video compression, also known as MPEG-4 AVC (Advanced Video Coding), jointly writ-

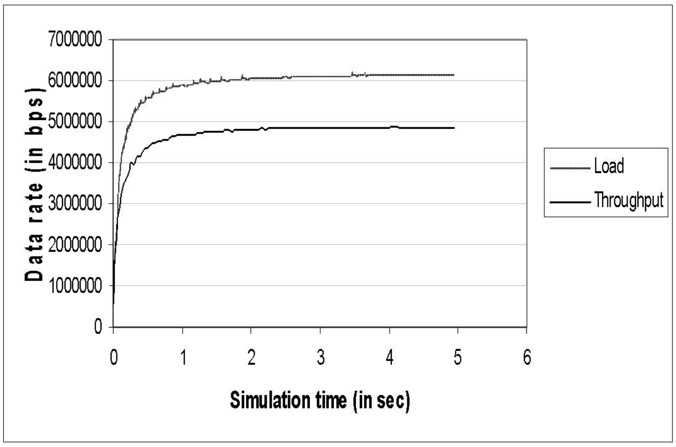

Figure 4.8: Load and throughput for G.711, 54 mbps 802.11g - 39 call scenario.

↓ Parameters / Codecs →	SQCIF	QCIF	CIF
Resolution	128x96	176x144	352x286
frame rate (in fps)	30	15	10
Framing interval (in ms)	33.333	66.667	100.000
Payload (in bytes)	264	1072	3216
Bit rate (in kbps)	64	128	384

Table 4.11: Video codec parameters.

ten by ITU-T Video Coding Experts group and Motion Pictures Standards Group [IT07b, ISO05]. We simulate three commonly used resolutions in video conferencing scenarios from the H.264 standard: Common Intermediate Format (CIF), Quarter CIF (QCIF) and Sub Quarter CIF (SQCIF). The formats' parameters are given in Table 4.11.

The delay and throughput constraints as defined for the voice codecs are also imposed on the video codecs. Throughput constraints should be satisfied, that is no video packet should be dropped and permissible delay is 75 ms. The simulation setup is shown in Figure 4.9.

The theoretical maximum number of flows (or calls) for the various schemes are given in Table 4.13. The detailed calculations using the throughput equations from Section 4.1 are given in Table 4.12. The simulated maximum number of video flows

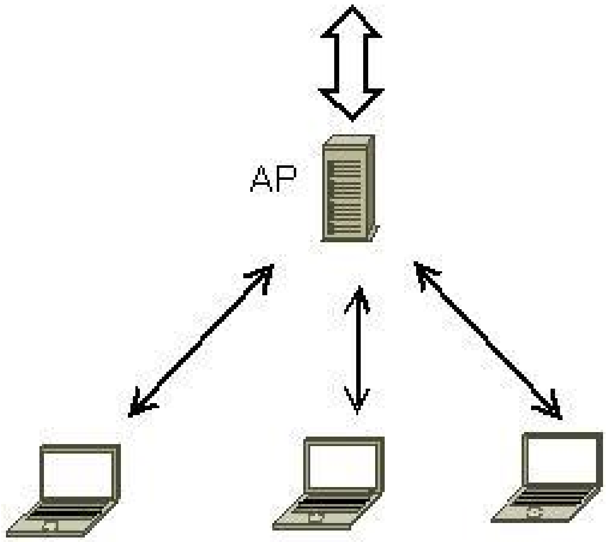

Figure 4.9: Video conferencing simulation setup.

for each scheme is shown in Table 4.14. The delay experienced by each scheme is given in Table 4.15. Figures 4.10, 4.11 and 4.12 compare the simulation and theoretical results for SQCIF, QCIF and CIF respectively.

4.3.1 Observations

At 1 Mbps data rate, the number of flows in both 802.11b as well as 802.11g are nearly the same. At 11 Mbps data rate the number of flows is slightly greater in 802.11g. With increase in data rates it can be seen that, for different codecs, the maximum number of calls do vary by a large extent. In voice codecs we noticed that the various codecs performed similarly (number of calls). This was largely due to the small packet size of the voice codecs, hence contention in the 802.11 CSMA/CA mechanism is acting as the main limitation. Here we find that due to the large packet size of video packets (as compared to voice packets) there is a significant performance difference between the various codecs especially at higher data rates. Contention remains a concern but the larger packet size allows for a more efficient use of the channel.

Scheme → Data rate (r)		802.11b 1	802.11b 5.5	802.11b 11	802.11g 1	802.11g 11	802.11g 54
pkt (in bytes)	SQCIF	304	304	304	304	304	304
	QCIF	1112	1112	1112	1112	1112	1112
	CIF	3256	3256	3256	3256	3256	3256
$frag\_size$		1500	1500	1500	1500	1500	1500
fragments per pkt	SQCIF	1	1	1	1	1	1
	QCIF	1	1	1	1	1	1
	CIF	3	3	3	3	3	3
DIFS		50	50	50	28	28	28
SIFS		10	10	10	10	10	10
PHY		192	192	192	20	20	20
backoff		31	31	31	31	31	31
slot		20	20	20	9	9	9
t_{pkt} (in μs)	SQCIF	2704	491.636	245.818	2710	251.818	56.074
	QCIF	9168	1666.909	833.455	9174	839.455	175.778
	CIF	12000	2181.818	1090.909	12006	1096.909	228.222
t_{ack}	(in μs)	1.978	0.36	0.18	7.978	6.18	6.037
Throughput (T) (in Mbps)	SQCIF	0.681	1.921	2.408	0.799	5.009	8.635
	QCIF	0.887	3.644	5.568	0.936	8.290	22.165
	CIF	0.686	3.281	5.267	0.709	7.016	24.065
Bandwidth (b) (in Mbps)	SQCIF	0.146	0.146	0.146	0.146	0.146	0.146
	QCIF	0.267	0.267	0.267	0.267	0.267	0.267
	CIF	0.521	0.521	0.521	0.521	0.521	0.521
Number of calls	SQCIF	4	13	16	5	34	59
	QCIF	3	13	20	3	31	83
	CIF	1	6	10	1	13	46

Table 4.12: Number of video flows: video capacity calculations.

4.4 Comments

In this chapter, we discussed the issue of capacity of 802.11 networks. We analysed homogeneous application deployments using real-time applications such as voice and video as examples. We presented theoretical capacities in terms of number of flows and compared these with simulated results. As discussed, many similar studies of capacity of WLANs has been done for various traffic scenarios [KMK05]. We use this study to establish a base case for our analysis of heterogeneous deployments

Scheme $\longrightarrow$ Data rate (r)	802.11b 1	5.5	11	802.11g 1	11	54
SQCIF 128x96, 30fps	4	13	16	5	34	59
QCIF 176x144, 15fps	3	13	20	3	31	83
CIF 352x286, 10fps	1	6	10	1	13	46

Table 4.13: Maximum number of video flows: theoretical results.

Scheme $\longrightarrow$ Data rate (r)	802.11b 1	5.5	11	802.11g 1	11	54
SQCIF 128x96, 30fps	4	13	16	5	29	76
QCIF 176x144, 15fps	3	13	20	3	27	94
CIF 352x286, 10fps	1	6	10	1	14	52

Table 4.14: Maximum number of video flows: simulation results.

and for establishing the AP-assignment problem.

The voice and video capacity experiments show that it is possible to both theoretically study and provide some amount of provisioning in 802.11 DCF. This is true when homogeneous applications, like in the above experiments, are deployed. We show in Chapter 5 that if applications like voice or video, which require QoS guarantees, are deployed along with best-effort services like FTP, the performance of such a heterogeneous system becomes unpredictable. That is, QoS guarantees cannot be given and, in general, real-time applications get severely affected. The results presented in this chapter then do not hold for heterogeneous cases. The main reason for this is the equal opportunity approach of the contention-based DCF mechanism which gives equal access priority to all nodes in the network. Also, one side-effect of this equal priority is bandwidth starvation of the AP. This starvation occurs due to the skewed nature of infrastructure mode operation in 802.11 where the AP has to handle, in the average case, half the load in the system. The AP then becomes the bottleneck in the system.[3]

[3]This issue, defined as AP starvation, is an important problem in WLANs. A version of this

Scheme ⟶ Data rate (r)	802.11b 1	5.5	11	802.11g 1	11	54
SQCIF 128x96, 30fps	0.01	0.015	0.014	0.055	0.02	0.0303
QCIF 176x144, 15fps	0.064	0.032	0.023	0.049	0.03	0.013
CIF 352x286, 10fps	0.0478	0.022	0.0236	0.0448	0.0446	0.0081

Table 4.15: Delay experienced at maximum number of video flows (in seconds).

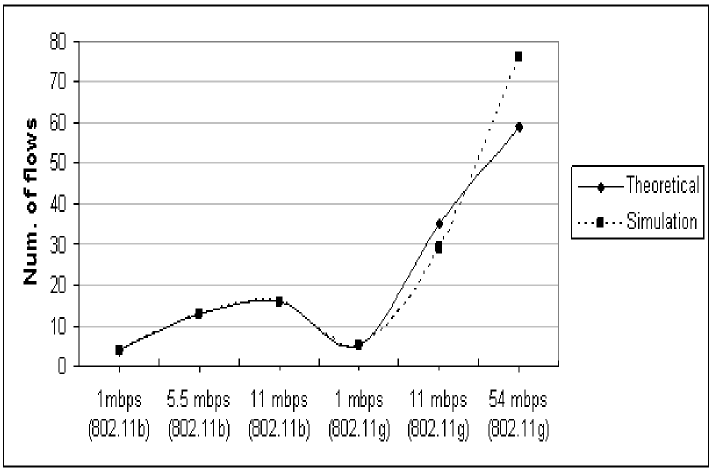

Figure 4.10: Maximum SQCIF video flows: theoretical vs simulation results.

The IEEE 802.11e standard [80205] was proposed to address such issues and provide mechanisms for differentiating various traffic classes based on their priorities. In the next chapter, we discuss the QoS mechanisms available in 802.11e and argue that the complexity of 802.11e makes it a practically difficult standard for implementation.

Also, in the next chapter, we group the above issues together as the problem of association of nodes to APs. We define this as the AP-assignment problem and present a novel modification to DCF as a QoS extension scheme and show its usefulness in prioritising applications in heterogeneous traffic scenarios. We then discuss a different approach called the sub-optimal application deployment. With

chapter and AP starvation has been discussed by us in [RIG05]. A mechanism to alleviate this by prioritising the AP is discussed by Wang and others in [WLL05].

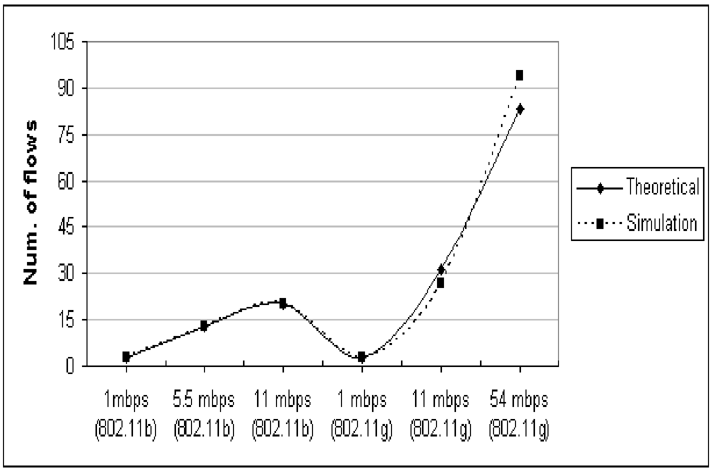

Figure 4.11: Maximum QCIF video flows: theoretical vs simulation results.

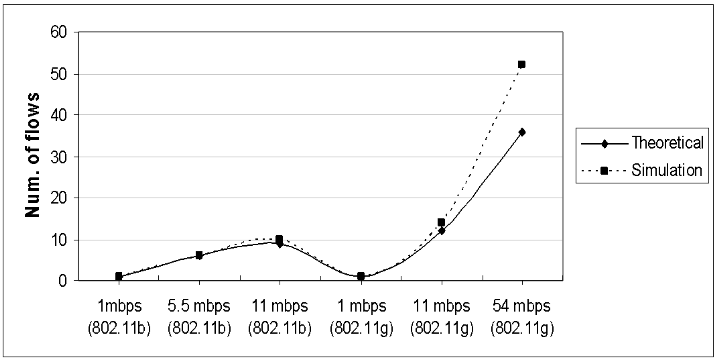

Figure 4.12: Maximum CIF video flows: theoretical vs simulation results.

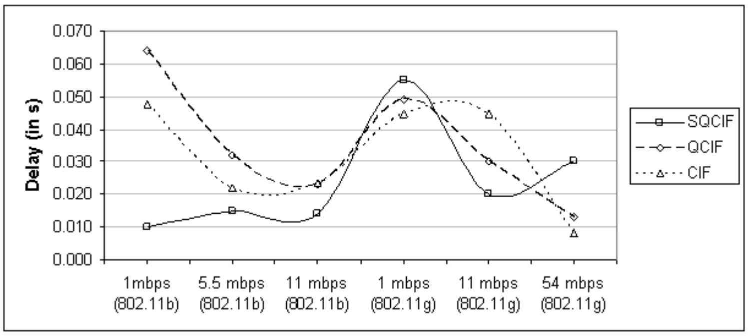

Figure 4.13: Delay for video schemes in 802.11b/g.

this approach we show that, for the AP-assignment problem, the overall system utilisation can be significantly improved.

Chapter 5

AP-assignment problem

In this chapter, we extend the association problem discussed for homogeneous application in Chapter 4. In that chapter, we had computed the number of homogeneous clients that could associate with a single AP. In this chapter, we extend the association problem to heterogeneous deployments and analyse joint deployments of realtime and non-realtime applications. As discussed in Chapter 3, addressing this association problem is the first step in the WLAN design solution.

In Section 5.1, we discuss the proposed QoS extensions to 802.11 and discuss their support for provisioning. We argue that these extensions have found little support due to their complexity. In Section 5.2, we discuss the issue of associating heterogeneous clients. We define this association problem as the AP-assignment problem (Section 5.3). For heterogeneous application traffic scenarios, we show that DCF performs poorly without additional functionalities for assisting priority traffic. In Section 5.4, we then show with a simple extension to DCF that this problem can be tackled. Finally, in Section 5.5, we discuss a unique approach for solving the AP-assignment problem called sub-optimal application deployment. We show that this approach results in significant improvement in the number of heterogeneous flows in the system and system utilisation.

5.1 Survey of QoS mechanisms in 802.11 - 802.11e and WMM

IEEE 802.11e is an amendment to the IEEE 802.11 standard that defines quality of service enhancements through modifications to the MAC layer [80205].

The 802.11e enhances DCF and PCF [1], through a new coordination function: the Hybrid Coordination Function (HCF). Within HCF, there are two methods of channel access: Enhanced Distributed Channel Access (EDCA) and HCF Controlled Channel Access (HCCA).

It introduces the concept of Traffic Classes (TC) which allows the setting of different priorities to different application scenarios. For example, VoIP could be assigned to a high priority class, and FTP could be set to a low priority class.

EDCA uses two mechanisms to provide differentiation among traffic classes [80205]. Each traffic class can have a different contention window, hence providing higher priority to traffic having a smaller contention window value. With EDCA, with prioritisation using TCs, a high priority traffic has a probabilistically higher chance of being sent than low priority traffic: this is because a station with high priority traffic waits for a smaller amount of time, on average, than a station with low priority traffic. The intuitive idea behind this being that a node with a smaller contention window is able to access the channel faster.

Also, EDCA allows different traffic classes to use different inter-frame spaces, called *Arbitration Inter-Frame Space (AIFS)*, instead of just the *DCF Inter-Frame Space (DIFS)* in DCF. This allows higher priority to nodes with smaller AIFS values.

The HCCA mechanism is similar to PCF but differs mainly in how Contention-

[1] The Point Co-ordination Function (PCF) is another mechanism specified in the original 802.11 standard [80203a]. PCF allows for contention-free access to the medium under infrastructure mode. This mechanism has not been widely adopted and very few APs or clients implement it.

Free Periods (CFP) are initiated. It is the most advanced co-ordination function in the 802.11 standards and allows for session level priorities to be assigned. But it has not found widespread use.

The 802.11e also has a Transmit Opportunity (TXOP) mechanism, attached to each TC, which allows a station to transmit as many frames as possible within a bounded time interval. The use of TXOP reduces the problem of low rate stations affecting the performance of a system, as found in DCF, by allowing higher-rate stations to send multiple frames in a single access. EDCA also uses admission control to provide a check on the number of admitted flows of the same priority.

Wi-Fi Multimedia (WMM) is an interoperability certification based on the IEEE 802.11e standard developed by the Wi-Fi Alliance [WMM04]. It provides basic QoS features to IEEE 802.11 networks and is a restricted version of EDCA.

WMM prioritises traffic according to four AC (Access Categories) - voice, video, best effort and background. Table 5.1 shows the contention window values for WMM (and 802.11e). The values aCWmin and aCWmax in the table are PHY specific. WMM certified APs must be enabled for EDCA and TXOP. All other enhancements of 802.11e are optional.

But, 802.11e is not widely adopted as a standard implementation. The main reasons are the size and complexity of implementation in order to be 802.11e compliant. Also, the widespread use of legacy 802.11 devices preclude such an adoption. As a subset of 802.11e, WMM has found some implementation support and for this purpose Atheros chipsets with open source madwifi drivers are used to some extent [Ath, Mad].

We now discuss the problem of associating heterogeneous clients to APs (as an extension to the homogeneous association studied in Chapter 4) and propose QoS extensions to DCF in order to support provisioning.

Access category	CWmin	CWmax
AC_VO (Voice)	(aCWmin + 1)/4 - 1, [7]	(aCWmin + 1)/2 - 1, [15]
AC_VI (Video)	(aCWmin + 1)/2 - 1, [15]	aCWmin, [31]
AC_BE (Best effort)	aCWmin, [31]	aCWmax, [1023]
AC_BK (Spare)	aCWmin, [31]	aCWmax, [1023]

Table 5.1: 802.11e/WMM contention window parameters per access category (values in brackets are for 802.11b). The values aCWmin and aCWmax in the table are PHY specific. For 802.11b, aCWmin = 31 and aCWmax = 1023. For 802.11g, aCWmin = 31 (for rates upto 11mbps) and aCWmin = 15 (otherwise), and aCWmax = 1023.

5.2 Associating clients to APs

Consider, as an example, deploying two types of nodes in the network (shown in Figure 5.1). The PDA and Workstation (WS) nodes run VoIP and FTP clients respectively. For FTP traffic, as it is a best-effort service, throughput satisfaction is an adequate constraint. VoIP additionally requires delay to be within some bound.

Definition: We define the *AP-assignment* problem as the problem of finding the *capacity* of an AP in terms of number of clients attached.

The clients maybe heterogeneous in nature, with respect to their applications. Now, our goal is to maximise the number of client, that is PDA and WS nodes, in any combination, that can be connected to a single AP in infrastructure mode while keeping their constraints satisfied.

In Chapter 4, we calculated the maximum voice and video flows in an infrastructure BSS. That was an instance of the AP-assignment problem for a homogeneous application deployment. In this chapter, we now extend the analysis to the heterogeneous case. Our intention is to state a restricted version of the problem and analyse it for certain practical traffic conditions.

Coming back to our example, assume that an AP allows n VoIP and m FTP

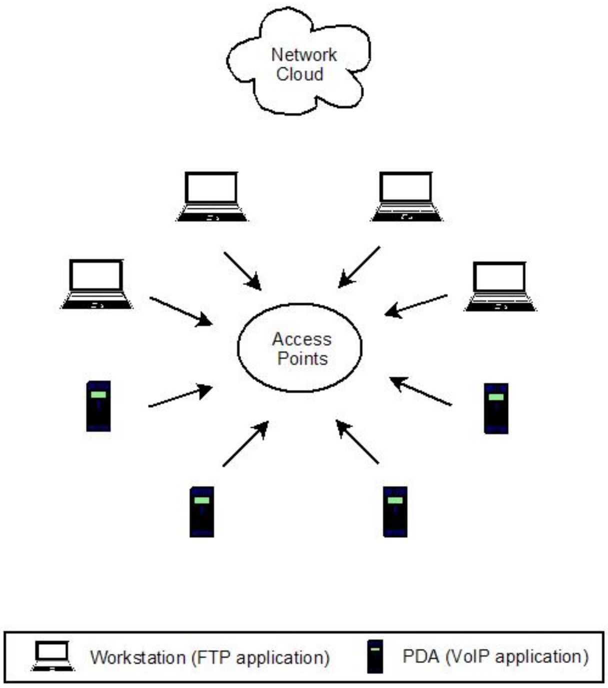

Figure 5.1: AP-assignment problem.

flows. The following issues then arise:

- Providing throughput and delay guarantees.
- Effect of additional flows on the system with respect to delay and throughput.
- If the system ran only one homogeneous application, say VoIP calls, and if n maximum calls were possible:
  - State of system under k ($< n$) calls.
  - Effect of additional FTP flows on k ($< n$) system state.

The first issue is that of arriving at a solution for the AP-assignment problem. The media access mechanism in DCF makes it difficult to provide guarantees on throughput and delay as its performance deteriorates exponentially with increase in traffic [Bia00]. If traffic is homogeneous (hence all nodes have similar traffic scenarios) some guarantees on throughput can be given, as seen by our capacity analysis in Chapter 4. Extending this to heterogeneous traffic proves to be difficult because of the nature of the DCF mechanism.

A more practical approach would be to try to optimise the deployment for a designated main application and analyse the effect of this on other applications. This is an approach we investigate further in this chapter.

5.3 Formal definition

We now formally state the AP-assignment problem.

Let $\boldsymbol{N}$ be the set of heterogeneous client nodes and $\boldsymbol{R}$ be the constraints imposed on the AP-assignment derivation. Based on the generic design problem stated in Section 2.3.1, the AP-assignment problem is defined as the set of APs $\boldsymbol{AP}$ required for $\boldsymbol{N}$ such that $\boldsymbol{R}$ is satisfied. Formally:

AP-assignment problem

Given $\boldsymbol{N}$,

Compute $\boldsymbol{AP}$,

Subject to $\boldsymbol{R}$,

While minimizing $|\boldsymbol{AP}|$.

As an example of constraints, consider a deployment of VoIP and FTP applications. The constraints $\boldsymbol{R}$ can then be throughput constraints for VoIP and FTP. Additionally, delay constraints can be imposed on VoIP.

5.4 Extending DCF to provide guarantees

To implement such a system, with respect to our above example, we need to restrict the number of VoIP calls in order to provide QoS guarantees. That is, we need to specify an Access Control Limit (ACL) mechanism to control the number of VoIP calls associated with an AP.

First, we make one assumption. As compared to 802.11e, in our case, only one queue per node is allowed due to adherence to DCF. Hence, our assumption is that only one application, either VoIP or any other traffic, per node is run. While this assumption is restrictive, it allows us to compute capacity of a network in terms of number of application flows instead of nodes.

The AP-assignment problem is then solved as follows (Figure 5.2):

1. Implement a scheme to differentiate between various application traffic flows.

2. Impose an ACL mechanism on the number of priority flows supported by AP (In our case, VoIP calls)

3. Add additional flows of the best effort service (FTP and VoIP) to test the capacity of the system.

Our scheme uses ACL for ensuring, with some guarantee, service to admitted VoIP calls. The scheme does this using a contention window-based service differentiation mechanism, similar to EDCA in 802.11e, to *prioritise* VoIP calls (Figure 5.3). The prototype was implemented in OPNET by extending the legacy DCF MAC. The contention window values at each node can be set to application specific values. Contention window parameters are given in Table 5.2. The setting of these values is an implementation choice. These values can either be assumed by the node (by learning from traffic) or the node can be managed by the AP.

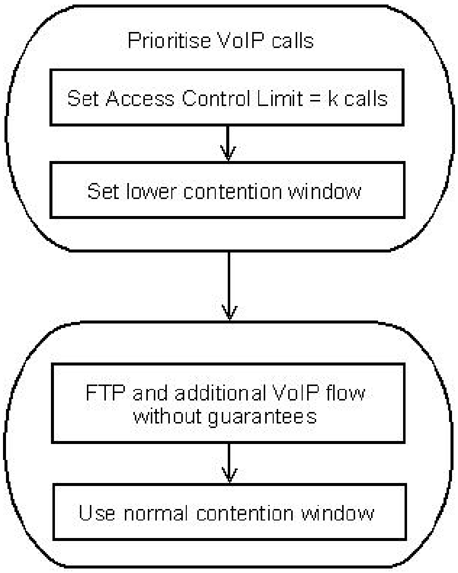

Figure 5.2: Modified DCF for priority VoIP traffic.

Application	CWmin	CWmax
VoIP (priority)	15	31
FTP and VoIP w/o priority	31	1023

Table 5.2: Contention window parameters for extended DCF.

We simulate the G.711 voice codec on an 802.11b network. The voice codec parameters are given in Table 4.6 and FTP simulation parameters are given in Table 5.3.

5.4.1 Simulation results

We simulated, using OPNET and scenarios in Figure 5.3, both normal and extended DCF with the same number from the maximum voice calls study (10 VoIP calls). The number of FTP flows were increased until the VoIP delay constraint was affected. A maximum number of 4 FTP flows were possible in extended DCF, without affecting the delay constraint on VoIP ($<$ 75 msec). VoIP delay in this case was 0.062 seconds. Running the same number of VoIP and FTP flows in normal DCF, the VoIP delay was out of bound (0.1 seconds). Table 5.4 compares FTP

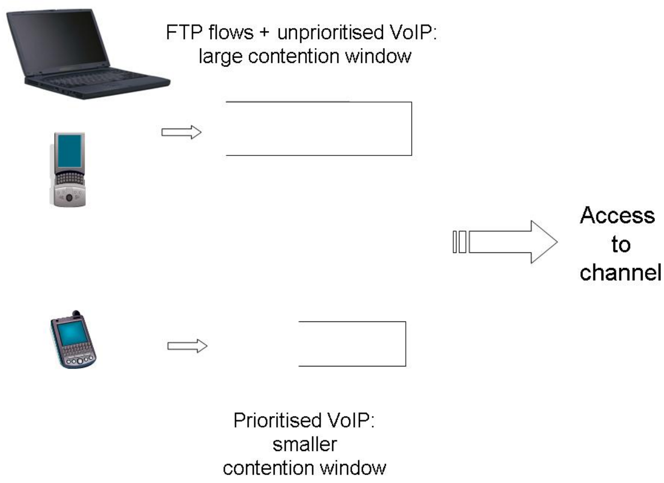

Figure 5.3: Contention window for ACL scheme - VoIP + FTP flows.

Attribute	Value
Command mix (get/total)	50%
Inter-request time (s)	exponential(60)
File size (bytes)	constant(125000)
Fragmentation size (bytes)	1500
Type of service	Best Effort (AC_BE)

Table 5.3: Extended DCF: FTP simulation parameters.

performance between normal and extended DCF. Extended DCF, while having increased delay per FTP flow, manages to improve the average throughput of the network and hence increases the utilisation of the network.

We use this modification of the DCF algorithm as a starting point and, in the rest of this chapter, we further examine the effect of running heterogeneous traffic on a network. As we noted in Chapter 4, DCF performs adequately in homogeneous application scenarios. Now, if the AP-assignment solution is optimal, additional application flows may affect the system. But, in homogeneous deployments, if a system were to operate under the k ($< n$) calls constraint, where n is the optimal AP-assignment solution, then the system state can be said to be *sub-optimal*. Such

Scheme	Load (in bps)	Throughput (in bps)	Delay (in sec)
DCF	81265	73712	0.009
Extended DCF	89575	85612	0.019

Table 5.4: Comparison of FTP performance in normal and extended DCF.

a system can be further investigated by adding other applications to study the systems performance in heterogeneous conditions. In the next section we elaborate on such sub-optimal application deployments. We especially investigate heterogeneous scenarios where realtime and non-realtime applications run together. And, for such deployments, we present techniques to maximise the network utilisation.

5.5 Sub-optimal application deployment

The DCF CSMA mechanism does not allocate bandwidth per station or allow any other mechanism to provide QoS to an application flow. As discussed before, this is the main cause for low capacity of an 802.11 WLAN. In Chapter 4, we calculated the maximum number of flows of an application for a given data rate. In Section 5.4, we extended this to prioritise a single application and provide better QoS. The maximum number of flows calculation allowed us to define the capacity of a WLAN in an homogeneous application scenario.

In this section, we extend the investigation of ACL-based access in a heterogeneous deployment. Our aim is to study the system utilisation of an heterogeneous deployment of application scenarios and the effect of deployment of a restricted number of a primary application.

Scenario overview

We observe that a common assumption made while calculating the capacity of a WLAN is to evaluate the network at its maximum capacity in terms of an application scenario. Now, the scenario we envision is as follows: consider an infrastructure mode WLAN, with VoIP (or Video or any other real-time application) as the prioritised application. The network, on the average case, will not be running the maximum number of VoIP flows. In that case, we can state some properties of the system in terms of additional number of flows of a non-real time application (like FTP). This allows us to study the number of heterogeneous application flows (VoIP and FTP) in the system and also comment on the system utilisation. Finally, these properties can then be used by an AP (as say a management utility) to set ACL limits dynamically on a system.

Sub-optimal heterogeneous application deployment

We compute the capacity of a heterogeneous deployment of applications, in terms of number of flows of each application scenario, by deploying a restricted number of the ACL application (instead of maximum n) and investigate the capacity further in terms of number of *other* application flows. If a sub-optimal (say $k < n$) number of flows of prioritised ACL application is deployed, then our interest is in defining capacity in terms of number of other possible application flows. These flows maybe non-prioritised applications or the prioritised application itself running under non-prioritised mode (a VoIP call running without priority). We define such a deployment as a *sub-optimal heterogeneous application deployment* in a WLAN, where an optimal solution would be the deployment of n homogeneous applications.

Solution approach

As an approach for network planning, evaluation of sub-optimal heterogeneous application deployments may provide more insights on DCF performance and may also improve the overall utilisation of the system. Note that the sub-optimality is only with respect to the prioritised application. We still attempt to maximise the number of flows, now of heterogeneous nature, in the system. System utilisation is also maximised.

The aim behind this experiment is to find the capacity of DCF under a heterogeneous application environment, that is, compute the number of prioritised and un-prioritised application flows. There are four issues here:

1. Effect of *under using* a network in terms of its priority traffic.
2. Effect of prioritising and un-prioritising heterogeneous application traffic.
3. Effect of prioritised traffic on un-prioritised traffic.
4. Effect of un-prioritised traffic on prioritised traffic.

The first step is to extend the ACL-based scheme from Section 5.4. The ACL-based scheme allows the maximum number of flows/calls of an application for a given data rate. In Section 4.1, we calculated the theoretical capacity of a WLAN. We now device a scheme to *virtually* divide this capacity. One part, determined by an ACL factor "k", is used by the ACL-based scheme to compute the reserved number of flows for a targeted application. The rest of the capacity is *allocated* to other applications flows. The DCF mechanism, by definition, does not have any notion of bandwidth reservation, hence this division of capacity is *virtual* in nature.

First, we present an extension of the theoretical calculations in Chapter 4 for sub-optimal voice flows.

5.5.1 Example: Sub-optimal G.711 voice calls in 802.11b

In the example in Section 4.1.1, we calculated the theoretical capacity of a single application (G.711 codec VoIP on an 11 Mbps WLAN). Throughput was given by equation 4.10. The bandwidth (b) required for one duplex call is 160000 bps =

k ↓	**Number of calls: $\lfloor kT/b \rfloor$** **802.11b (in mbps)**			**802.11g (in mbps)**		
	1	**5.5**	**11**	**1**	**11**	**54**
$\frac{T}{b} \rightarrow$	3.805	9.135	10.818	4.781	25.782	39.651
1.0	3	9	10	4	25	39
0.9	3	8	9	4	23	35
0.8	3	7	8	3	20	31
0.7	2	6	7	3	18	27
0.6	2	5	6	2	15	23
0.5	1	4	5	2	12	19
0.4	1	3	4	1	10	15
0.3	1	2	3	1	7	11
0.2	0	1	2	0	5	7
0.1	0	0	1	0	2	3

Table 5.5: k value vs Number of voice calls for G.711 codec: theoretical results. Values for Throughput (T) and bandwidth (b) are from Table 4.7.

.16 $Mbps$. Adding factor k to the equation and rewriting it to accommodate fractional bandwidth assignment gives us:

$$\lfloor k.T/b \rfloor = \lfloor 1.73k/.16 \rfloor \tag{5.1}$$

As an example, if we assume that 30% of bandwidth is reserved for G.711 calls then $k = 0.3$. The number of calls are $\lfloor 1.73 * 0.3/.16 \rfloor = 3$ *calls*. Note that, the maximum calls by simulation was $\lfloor T/b \rfloor = 10$ *calls* and therefore $\lfloor 10 * 0.3 \rfloor = 3$ *calls*

Table 5.5 shows theoretical calculations for different rates of 802.11b/g. Tables 5.6, 5.7 and 5.8, present theoretical throughput calculations for the other voice codecs: G.723.1, G.729 and GSM respectively. Additional codec parameters used in these tables are from Table 4.7.

k ↓	**Number of calls: $\lfloor kT/b \rfloor$**					
	802.11b (in mbps)			**802.11g (in mbps)**		
	1	**5.5**	**11**	**1**	**11**	**54**
$\frac{T}{b} \rightarrow$	6.494	11.15	12.115	9.966	34.608	43.093
1.0	6	11	12	9	34	43
0.9	5	10	10	8	31	38
0.8	5	8	9	7	27	34
0.7	4	7	8	6	24	30
0.6	3	6	7	5	20	25
0.5	3	5	6	4	17	21
0.4	2	4	4	3	13	17
0.3	1	3	3	2	10	12
0.2	1	2	2	1	6	8
0.1	0	1	1	0	3	4

Table 5.6: k value vs Number of voice calls for G.723.1 codec: theoretical results. Values for Throughput (T) and bandwidth (b) from Table 4.7.

k ↓	**Number of calls: $\lfloor kT/b \rfloor$**					
	802.11b (in mbps)			**802.11g (in mbps)**		
	1	**5.5**	**11**	**1**	**11**	**54**
$\frac{T}{b} \rightarrow$	6.631	11.222	12.157	10.294	34.96	43.204
1.0	6	11	12	9	34	43
0.9	5	10	10	9	31	38
0.8	5	8	9	8	27	34
0.7	4	7	8	7	24	30
0.6	3	6	7	6	20	25
0.5	3	5	6	5	17	21
0.4	2	4	4	4	13	17
0.3	1	3	3	3	10	12
0.2	1	2	2	2	6	8
0.1	0	1	1	1	3	4

Table 5.7: k value vs Number of voice calls for G.729 codec: theoretical results. Values for Throughput (T) and bandwidth (b) from Table 4.7.

5.5.2 Problem definition

We now state two types of sub-optimal application deployment for AP-assignment problems to capture heterogeneity of application scenarios and different sub-optimality

k ↓	**Number of calls: $\lfloor kT/b \rfloor$** **802.11b (in mbps)** **1**	**5.5**	**11**	**802.11g (in mbps)** **1**	**11**	**54**
$\frac{T}{b} \rightarrow$	6.204	10.989	12.019	9.298	33.841	42.847
1.0	6	10	12	9	33	42
0.9	5	9	10	8	30	38
0.8	4	8	9	7	27	34
0.7	4	7	8	6	23	29
0.6	3	6	7	5	20	25
0.5	3	5	6	4	16	21
0.4	2	4	4	3	13	17
0.3	1	3	3	2	10	12
0.2	1	2	2	1	6	8
0.1	0	1	1	0	3	4

Table 5.8: k value vs Number of voice calls for GSM codec: theoretical results. Values for Throughput (T) and bandwidth (b) from Table 4.7.

conditions.

First, we define three application classes:

- *Alpha* (α): Prioritised applications running under ACL and subject to sub-optimality condition.
- *Beta* (β): Applications run with normal priority.
- *Gamma* (γ): Applications of the same class as *Alpha*, but running under un-prioritised mode.

Assumptions:

Each application class corresponds to one application scenario (like VoIP, FTP, video and others). *Alpha* is the primary application in the system and the system capacity is optimised for this application; that is, utilisation of the system is calculated after deciding the number of *Alpha* flows. *Beta* and *Gamma* are mutually exclusive classes of applications. The *Gamma* class is defined to capture the effect of running *Alpha* class applications without priority.

Hypothesis:

We expect classification of applications and prioritisation of some of the applications to allow us better control over the number of application flows. Prioritisation of some applications is by providing them better channel access, that is, by using shorter queues and contention windows.

Definitions:

We can now define the sub-optimal application deployment AP-assignment problems based on the absence or presence of *gamma* applications. Both problem definitions are based on the generic design problem stated in Section 2.3.1.

Given a system can support n *Alpha* flows and $\boldsymbol{R}$ be the constraints imposed, the first Sub-Optimal application deployment AP-assignment (`SOAP`) problem is defined as follows. :

> `SOAP1`
>
> *Given* k *Alpha* flows ($|\alpha| = k$),
> *compute* number of *Beta* flows ($|\beta|$),
> *subject to* constraints $\boldsymbol{R}$.

Given a system can support n *Alpha* flows and $\boldsymbol{R}$ be the constraints imposed, the second SOAP problem is defined as follows:

> `SOAP2`
>
> *Given* k *Alpha* flows ($|\alpha| = k$),
> *compute* number of *Beta* flows ($|\beta|$), and number of *Gamma* flows ($|\gamma|$),
> *subject to* constraints $\boldsymbol{R}$.

The set of constraints $\boldsymbol{R}$ is defined as follows: *Alpha* and *Gamma* flows have delay and throughput constraints while *Beta* flows have throughput as constraint.

Both `SOAP1` and `SOAP2` study the four issues stated in Section 5.5 on effect of traffic prioritisation and application of sub-optimality condition. `SOAP1` studies the problem for two classes of traffic, prioritised class *Alpha* and un-prioritised class *Beta*. `SOAP2`, additionally, studies the effect of adding *Gamma* class applications.

We can consider the example explained in Section 5.3. As an example of constraints, consider a deployment of VoIP and FTP applications. The constraints $\boldsymbol{R}$ can then be throughput constraints for VoIP and FTP applications. Additionally, delay constraints can be imposed on VoIP applications.

5.5.3 Implementation

The implementation is similar to that of extended DCF in Section 5.4. To implement such a system, we first need to restrict the number of VoIP calls in order to satisfy constraints $\boldsymbol{R}$. Hence, we need to specify an Access Control Limit (ACL) mechanism to control the number of α VoIP calls associated with an AP.

SOAP was implemented in OPNET by extending the legacy DCF MAC. The implementation is as follows:

1. Implement scheme to differentiate between various application traffic flows. The scheme is a contention window-based service differentiation mechanism, similar to EDCA (shown in Table 5.2)

2. Impose an ACL mechanism on the number of α flows supported by AP (In our case, VoIP calls)

3. Add additional β and/or γ flows as best effort service (FTP and VoIP) to compute the capacity of the system.

Table 5.9 shows the application scenarios used in the simulation. With respect to definitions in the previous section, `SOAP1` assumes some given number of prioritised VoIP flows (α) and computes the number of FTP flows (β). For `SOAP2`, we consider the results of `SOAP1` as the starting point (for number of prioritised VoIP and FTP

Application class	Application	
α	VoIP -	G.711
β	FTP -	Avg. Load - 250 and 500kbps
γ	VoIP -	G.711

Table 5.9: Application classes used in SOAP simulation.

Attribute	Value
Command mix (get/total)	100%
Inter-request time (s)	exp(1)
File size (bytes)	FTP 250 - cons(31250), FTP 500 - cons(62500)
Fragmentation size (bytes)	1500
Type of service	Best Effort (AC_BE)

Table 5.10: SOAP simulation parameters for FTP - Average load 250 and 500 kbps.

flows). In addition, we add un-prioritised VoIP flows (γ). If this addition fails the constraints $\boldsymbol{R}$, that is, delay and throughput constraints of VoIP and FTP flows, then the number of FTP flows is reduced. This reduction is performed until *at least* one un-prioritised VoIP call is possible while $\boldsymbol{R}$ remains satisfied. The number of prioritised VoIP flows remains the same.

The simulation setup assumes an 802.11g system at 11 Mbps. The voice codec is G.711. G.711 parameters are given in Table 4.4, FTP parameters are given in Table 5.10.

Note that the simulated maximum number of G.711 calls was 18 (Table 4.9). We simulate SOAP1 with 18 calls as the starting point ($|\alpha_{k=1.0}| = 18$). For each value of k in Table 5.5, we find the maximum number of β applications. Delay constraint in $\boldsymbol{R}$ is: *delay* of $\alpha_k < 75ms$.

5.5.4 Results

`SOAP1`

The results are given in Tables 5.11 and 5.12. For $k = 1$, no flows are possible in either FTP scenarios (results shown in italics). In both cases, VoIP delay goes above bound in the presence of FTP flows with *delay* $(\alpha_{1.0}) \sim 86ms$ in both FTP cases. From $k = 0.9$ *to* 0.5, the number of FTP flows steadily increases with decreasing number of VoIP flows. VoIP remains within the delay bound in all cases. At $k = 0.4$, the effect of β applications on α becomes negligible and α delay is far less than the bound. This is the effect of reduction in number of α applications also reducing the contention between the flows. The next simulation point, $k = 0.3$, proves this effect and we stop the analysis.

The system utilisation improves with a reduction in k. From $k = 0.9$ *to* 0.3, the utilisation improves from 30% to above 50% in both FTP cases. At $k = 0.8$, the number of α flows is around 75% of the maximum number of flows. But, the number of β flows are 7 and 4 in the two FTP scenarios. In terms of number of flows, this is a significant improvement in utilisation with a small loss in number of α flows. Figures 5.4 and 5.5 show the number of α_k and β_k flows for 250 and 500 Kbps FTP respectively. These graphs are helpful in setting the ACL limits and determining the *operating point* of an AP.

`SOAP2`

In `SOAP2`, we extend `SOAP1` results for additional γ applications. The steps taken in `SOAP2` are:

1. α_k^{soap2} is computed the same way as α_k^{soap1}.

2. For each α_k^{soap2}, with `SOAP1` β results as the starting point, compute `SOAP2`.

k	$\lvert\alpha_k\rvert$	α_k delay	$\lvert\beta_k\rvert$	β_k throughput	β_k delay
1.0	18	*0.086*	*1*	*63672*	*0.007*
0.9	16	0.067	3	713658	0.010
0.8	14	0.074	7	1447647	0.012
0.7	12	0.075	10	2181612	0.013
0.6	10	0.071	13	2920485	0.015
0.5	9	0.071	16	3306418	0.017
0.4	7	0.027	20	4134720	0.023
0.3	5	0.009	24	5002889	0.021

Table 5.11: `SOAP1` results for FTP 250 Kbps with G.711 codec on 11 Mbps 802.11g. α_k and β_k delays are in seconds. β_k throughput is in bps.

k	$\lvert\alpha_k\rvert$	α_k delay	$\lvert\beta_k\rvert$	β_k throughput	β_k delay
1.0	18	*0.086*	*1*	*101247*	*0.008*
0.9	16	0.070	2	758230	0.105
0.8	14	0.073	4	1481418	0.013
0.7	12	0.073	5	2229776	0.015
0.6	10	0.072	7	2969675	0.015
0.5	9	0.071	9	3386316	0.016
0.4	7	0.038	12	4293402	0.022
0.3	5	0.011	15	5179227	0.021

Table 5.12: `SOAP1` results for FTP 500 kbps with G.711 codec on 11mbps 802.11g. α_k and β_k delays are in seconds. β_k throughput is in bps.

That is, for each value of α_k^{soap2}, we range β_k^{soap2} (from β_k^{soap1} to 1) and find γ_k^{soap2}.

As an example, from Table 5.11, $|\alpha_{0.6}^{soap1}| = 10$ (for $k = 0.6$). Therefore, $|\alpha_{0.6}^{soap2}| = |\alpha_{0.6}^{soap1}| = 10$. Next, we range $|\beta_{0.6}^{soap2}|$ from $|\beta_{0.6}^{soap1}| = 7$ to 1.

The constraints $\boldsymbol{R}$ imposed are: *delays* $\alpha_k^{soap2}, \gamma_k^{soap2} < 75ms$.

The number of data points in this experiment are $|\alpha_k^{soap2}| * \sum_k |\beta_k^{soap2}|$. Instead of performing such exhaustive simulations, we consider only the *best* k case in `SOAP1`. We present results for $k = 0.8$ in Table 5.13 for $|\alpha_{0.8}^{soap2}| = 14$ & $|\beta_{0.8}^{soap2}| = 7$.

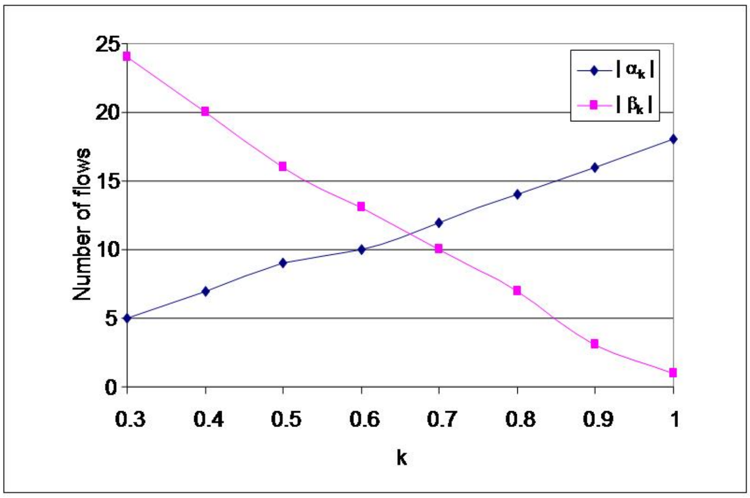

Figure 5.4: SOAP1 FTP 250 kbps: k vs Number of α_k and β_k flows.

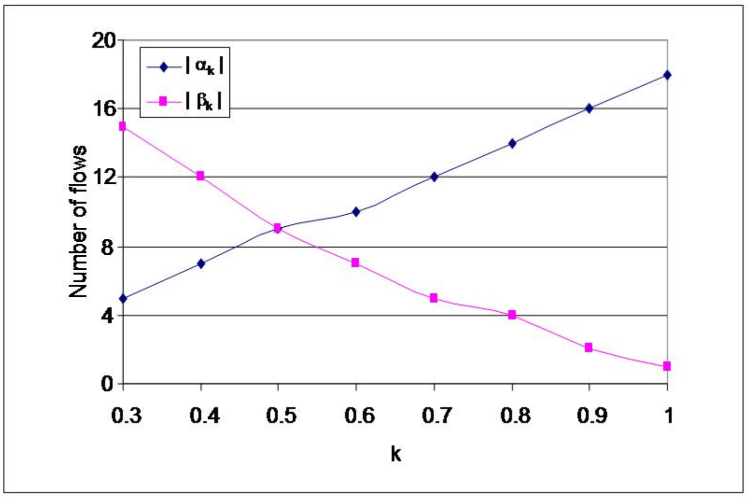

Figure 5.5: SOAP1 FTP 500 kbps: k vs Number of α_k and β_k flows.

The simulation results show that, from 7 to 3 FTP flows no additional γ flows were possible. At $|\beta_{0.8}^{soap2}| = 2$, 2 γ flows were possible, this makes the total number of VoIP flows = $|\alpha_{0.8}^{soap2}| + |\gamma_{0.8}^{soap2}| = 18$ flows. This is the same as number of VoIP flows in $|\alpha_{1.0}^{soap1}|$. But, in $|\alpha_{1.0}^{soap1}|$ the number of β flows was 1.

$\lvert\beta_k\rvert$	$\lvert\gamma_k\rvert$	γ_k delay	α_k delay
7 to 3	0	-	-
2	2	0.033	0.074
1	4	0.337	0.012

Table 5.13: SOAP2 results for $k = 0.8$ ($\lvert\alpha_{0.8}\rvert = 14$), FTP 250 Kbps with G.711 codec on 11 Mbps 802.11g. α_k and γ_k delays are in seconds.

5.5.5 Observations

The first technique, `SOAP1`, showed significant improvement in system utilisation up to 75%. The second technique, `SOAP2`, did not show any significant improvement in performance over `SOAP1`. Both the techniques take advantage of the non-deterministic nature of the DCF mechanism and improve system capacity by providing priority flows with a better chance of accessing the channel. Additionally, controlling the ratio of prioritised and non-prioritised flows allows the problem to be studied as an ACL mechanism. Figures 5.4 and 5.5 show the number of α_k and β_k flows vs k. Graphs like these can be used to determine the *operating point* of the AP and setting of access control limits. The assignment information can also be used as a simple look-up table while planning.

An extension of the work done by Bianchi for modeling of the DCF mechanism [Bia00], can be used to construct a formal model for `SOAP`. However, our main aim was to show that the capacity constraint problem can be addressed for heterogeneous traffic using extension mechanisms for DCF. Therefore, our focus is only on applying these techniques for addressing the issue of AP-assignment in wireless network design.

5.6 Summary

We first defined the AP-assignment problem as the problem of finding the capacity of an AP in terms of the clients associated. We then proposed a simple extended DCF as a QoS extension to 802.11 and showed its usefulness in prioritising applications in heterogeneous traffic scenarios. This extension showed a small improvement in performance over DCF.

We then presented a case for sub-optimal deployment of a primary application in heterogeneous deployments. The SOAP techniques significantly improved the system utilisation by up to 75%. We also showed that with an insignificant drop in number of primary realtime application flows (like voice), the capacity of a system can be improved in an heterogeneous deployment.

Recalling the design problem in Chapter 3, the first step in constructing a WLAN topology was to consider the problem of AP-assignment. In this chapter, we have shown how heterogeneous applications, especially combinations of realtime and non-realtime applications, can be deployed in an 802.11 network with minor modifications to the access mechanisms and the use of ACL. We show in the next chapter how the solutions to the AP-assignment problem can be used as inputs while constructing the topology in a WLAN design problem.

Chapter 6

WIND framework for WLAN deployment

In Chapter 5, we discussed the issue of AP-assignment in WLANs. The AP assignment problem involved finding mappings between heterogeneous client nodes and APs. We had discussed ACL-based schemes to improve performance of DCF in WLANs so as to maximise the utilisation of APs. We also discussed Sub-optimal AP-assignment (SOAP) techniques to improve the utilisation of a WLAN. In this chapter, we take this work further by employing it in the larger scenario of WLAN deployment.

In Section 6.1, we discuss the problem of topology construction in WLANs and present an overview of the design approach. In Section 6.2, we discuss the inputs and outputs of the tool. WIND uses an information base as its main source of input data and additionally takes various user-defined inputs such as capacity constraints, optimisation criteria and node deployment scenarios. In Section 6.3, we define and discuss composite unit which is the basic building block for topology construction in WIND. In Section 6.4, we present the WIND framework and discuss the algorithm details with an example. Sections 6.5 and 6.6 discuss the implementation details and results.

6.1 Introduction

As discussed in Section 2.2, WLANs are generally deployed to alleviate costs of last-hop wired connectivity. This is done by providing the clients single-hop wireless access through APs. The APs themselves have wired connectivity to the backhaul. APs can also be connected to the backhaul using multiple hops of wireless links. WLANs have become extremely useful because client deployments need not be strictly planned and nodes can be placed in the deployment area in any manner as long as connectivity to at least one AP is available.

From the perspective of a traditional LAN, APs replace last-hop relay nodes and act as a hub for clients. APs have to be placed over the deployment area and clients connect to the *nearest* AP. Here, finding the nearest AP is dependent on many factors: signal strength available at the client from an AP, manual AP-client mapping or ACL mechanism, availability of APs on the channel allocated to client and the capacity of AP. The association decision is usually made taking one or more of the above issues into consideration. All these issues place constraints on placement and number of APs.

The AP-client association is the first step in the network planning process. We discussed before how using the AP-assignment techniques from Chapter 5, the system utilisation can be improved and AP-client mapping maximised. Associating more client nodes with the given APs reduces the number of APs required for connectivity, hence reducing costs.

We discussed the two network planning approaches, coverage and capacity, in Section 2.2.2. In Section 2.2.4, we discussed the drawbacks of current WLAN designers providing only coverage planning instead of bandwidth provisioning.

A typical campus 802.11 network of APs is deployed based on a simple site survey. The network is only later tested for satisfactory performance [ZPK03]. Additional APs are deployed as and when the network QoS metrics are found inadequate. As discussed in Section 2.2.3, simulators can also be used for design validation but the process is still a significant manual effort.

Also, while the above approach can be said to be adequate for deploying best-effort services, any application requiring guarantees, like voice, will suffer if their requirements are not considered at the time of design. As seen in our AP-assignment problem, provisioning for QoS-aware deployment in a contention-based system such as the 802.11 DCF is difficult.

6.1.1 Design approach

In this chapter, we take the approach of designing networks based on their capacity requirements and develop a tool to construct topologies. We explain the framework of this design tool, the Wireless Infrastructure Network Deployment tool for WLANs ($\texttt{WIND}_{\texttt{wlan}}$), that constructs topologies for deployment of WLANs in office environments. $\texttt{WIND}_{\texttt{wlan}}$ takes the AP-assignment solutions as a starting point and constructs topology graphs for given deployment scenarios.

The topology is built to provide client devices with intra-office connectivity and routes (through gateway nodes) to the backhaul network to provide inter-WLAN and external connectivity. The topology further takes into account the layout of the deployment area and the node deployment strategy of each type of client device during construction.

$\texttt{WIND}_{\texttt{wlan}}$ allows a network designer to plan for capacity at the design stage itself and automate the design process by computing the topology of an entire network starting from client node demand requirements. In other words, the tool allows us

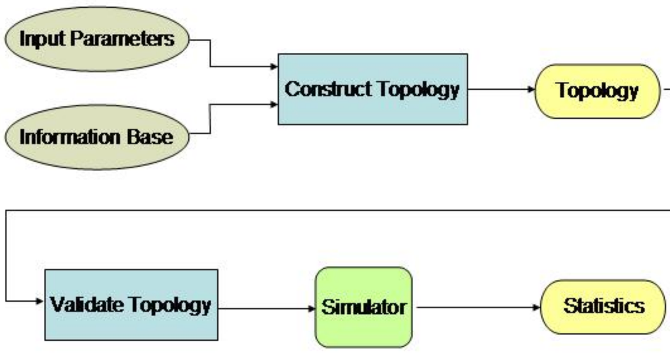

Figure 6.1: Generic topology construction framework.

to automate the process of network planning from a bandwidth provisioning perspective. It then eases validation by using inputs and outputs in simulator readable formats. $\texttt{WIND}_{\texttt{wlan}}$ allows rapid construction of topologies for different application and deployment scenarios and reduces the time taken for network deployment and modification.

Figure 6.1 shows the generic approach taken by $\texttt{WIND}_{\texttt{wlan}}$ for topology construction. It involves two steps: 1) construct the topology and 2) validate the topology.

Two inputs are now required: a database containing information about network elements and input parameters describing the required deployment scenario and network requirements. The deployment scenario defines the deployment area and the network elements to be deployed. The requirements define the constraints imposed on the topology construction. The output of the topology constructor is a network graph and it is then validated using a simulator. We now discuss the input and output parameters in $\texttt{WIND}_{\texttt{wlan}}$.

Term	Definition
Node Unit (NU)	Abstract representation of a network element. All client and infrastructure nodes are considered as network elements.
Relay nodes	All infrastructure nodes (APs, routers and switches) are considered as relay nodes.
C	Capacity of a link in bytes/sec.
Addr	Address of an *NU*
Load	Average load in bytes/sec.

Table 6.1: Terms used in WIND network model.

6.2 Input and output parameters

`WIND`$_{\texttt{wlan}}$ takes two main inputs: an information base of network elements and input parameters defining the network requirements. It also takes as input, the rules and constraints to be applied on the network. The terms used in `WIND`$_{\texttt{wlan}}$ are defined in Table 6.1.

6.2.1 Wireless information base

`WIND`$_{\texttt{wlan}}$ makes use of an *information base (IB)* containing parametric information on network elements and network properties. This database contains node and link type information for the network under consideration. Table 6.2 shows part of an *IB* for a network deploying some Workstation Node Units (NU_{WS}) and mobile PDAs (NU_{PDA}). Other example NUs shown represent servers (NU_S) and relay nodes (NU_{Relay}). As shown in the table, a relay node is not the source or destination of any *AS*. The *IB* stores the following information:

NU Type

Type of an NU deployed in the network. Type information consists of information on applications running on NU, links available and other parameters. The *NU* *types* of all network elements, including relay nodes, are defined in the *IB*. *NU Type* parameters are:

- *Application scenarios (AS)*: The set of applications running on NU. *AS* is represented by a tuple consisting of outgoing and incoming load, the source and the destination addresses:

  $$AS = < Traffic_{out},\ Traffic_{in},\ Addr_{src},\ Addr_{destn} >$$

  where,
  $Traffic_{out}$, $Traffic_{in}$ = outgoing and incoming traffic *load* respectively.
  $Addr_{src}$, $Addr_{destn}$ = source and destination address respectively.

  The rows in Table 6.2 represent application scenarios. NU_{PDA}, for example, has $Traffic_{out}$ = 10 Kbps and $Traffic_{in}$ = 100 Kbps. The $Addr_{destn}$ of the *AS* is NU_S. An *NU* can have one or more application scenarios in the *IB*. As shown in the table, a relay node NU_{Relay} is not the source or destination for any *AS*.

- *Total Load*: Total of all *AS traffic loads.* This is used for link budgeting and statistics.

- *Links*: Type and number of links present in an NU. In Table 6.2, NU_{PDA} and NU_{WS} have wireless links while NU_S has an Ethernet link. The NU_{Relay} has both links present.

- *AS-Link mapping*: The mapping determines the preferred link on which *AS traffic* is carried. This is useful in the case where more than one link is available at a node (multiple paths to a destination are available). In Table 6.2, an NU_{Relay} has two links available with it and can choose between them based on traffic priorities (the choice here is link 2).

- Mobility: Defines a mobile or a static node. In Table 6.2, an NU_{PDA} is a mobile node.

Link type

Type of a link in an NU. Properties are:

- Link name.
- Type: A link can be wired or wireless.
- *Capacity* of link, C.
- Link specification function (*maxNodes()*): This function is used to capture the characteristics of the underlying MAC and abstract an 802.11 wireless link.

For the link specification function, a link is specified in terms of the nodes sharing the link and their application traffic. The AP-assignment solutions for homogeneous and heterogeneous traffic from Chapter 5 are used to define this function. The function, defined for each link type, takes a list of NUs and returns a subset of NUs that can share that link (that is, associate with *an* AP). This is done by computing the number of NUs that can share a link given application scenario properties of NUs. The NUs as well as the application scenarios can be heterogeneous in nature.

For example, using Table 5.12 and the discussion in Section 5.5 for `SOAP1`, we can define a *maxNodes()* function for the following parameters: 1) set of PDAs (running VoIP) and Workstations (running FTP), 2) 802.11g 11 Mbps link and 3) FTP rate 500 Kbps. We then state *maxNodes()* = 14 PDAs and 4 Workstations.

Figure 6.2 shows the XML schema for information base. The $\texttt{WIND}_{\texttt{wlan}}$ tool uses the schema to verify an *IB* for correctness and conformance. The XML file and an example information base is given in Appendix A.1.

NU	***Traffic**$_{out}$*	***Traffic**$_{in}$*	***Addr**$_{src}$*	***Addr**$_{destn}$*	**Link**	**AS-Link Map.**	**Mobility**
NU_{PDA}	10000	100000	$< N_{PDA} >$	$< N_S >$	1	1	Yes
NU_{WS}	10000	1000000	$< N_{WS} >$	$< N_S >$	1	1	No
NU_S	10^6	100000	$< N_S >$	Undefined	2	2	No
NU_{Relay}	$5 * 10^5$	$5 * 10^5$	Undefined	Undefined	1,2	2	No
...	...	...	...	...	...	...	...

Table 6.2: Example information base. Traffic$_{out}$, Traffic$_{in}$ are in bits per second. Link type 1 represents a 802.11 10 Mbps wireless link and type 2 represents a 10 Mbps Ethernet link.

6.2.2 Node and deployment layout parameters

Now, we describe the deployment layout parameters and a strategy for node deployments called *affinity factor*.

Deployment layout graph (DLG)

Generally, a network deployment starts with an office floor plan. An example layout for an office floor plan is shown in Figure 6.3. In RF-based coverage network planning, this plan is evaluated for correct placement of APs. In our case of capacity planning, we abstract the floor plan into a deployment layout.

Definition: A deployment layout graph is an abstract representation of the actual physical layout. The layout is represented as a graph $\mathcal{F} = (\mathcal{W}, \mathcal{I})$, where $\mathcal{W}$ is the set of nodes representing areas where network elements are deployed and $\mathcal{I}$ is the set of edges between nodes indicating the physical connection between areas.

In Figure 6.3, each floor and the corridor is abstracted as a node in the DLG. The actually physical connections between the office areas are represented by edges between nodes in the DLG.

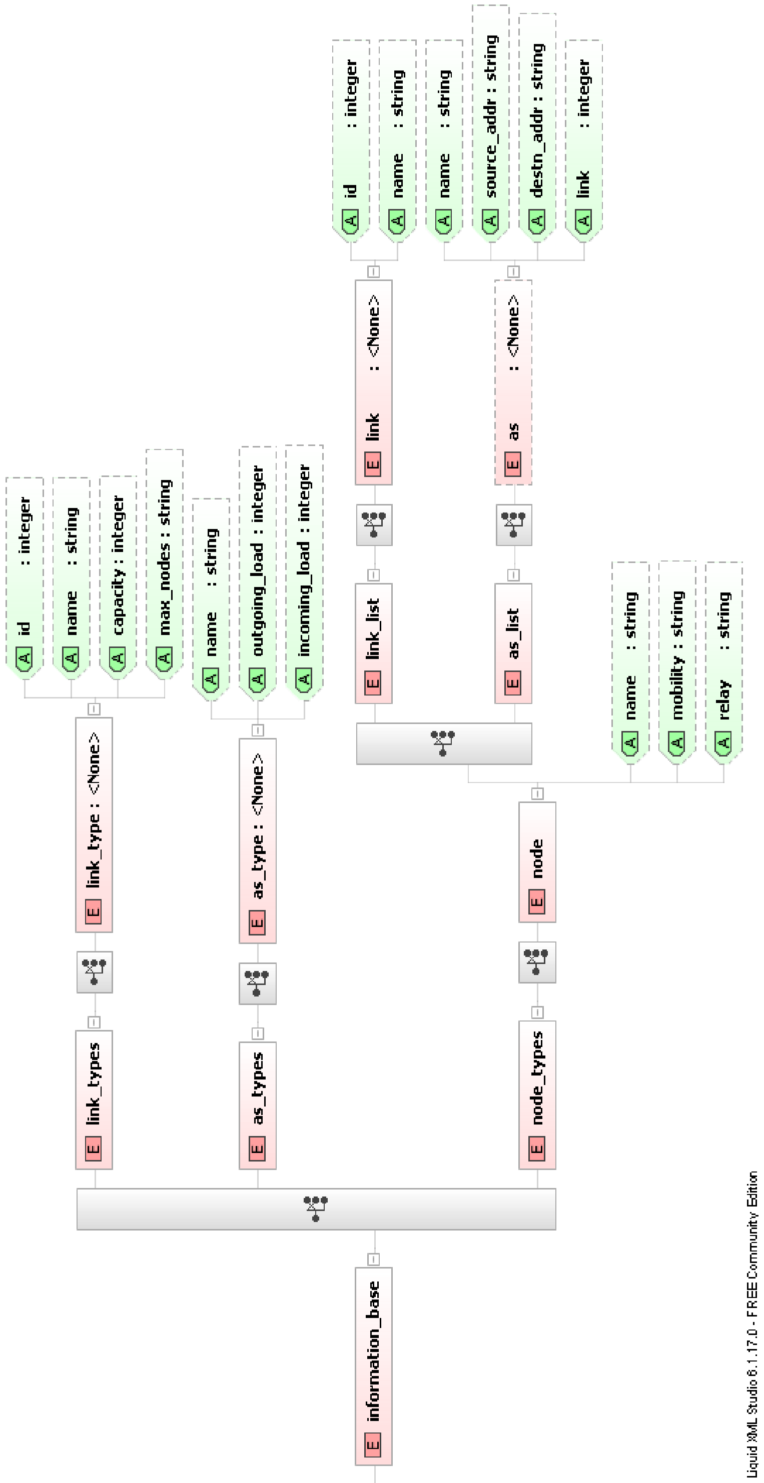

Figure 6.2: XML schema for information base.

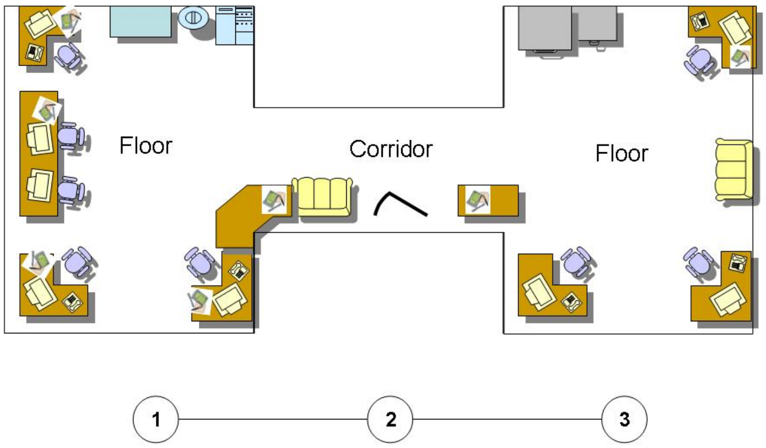

Figure 6.3: Example: (a) floor plan, (b) corresponding deployment layout.

Affinity factor (af)

Given the type of NUs to be deployed, the number of NUs and the DLG where NUs are to be deployed, we still require a heuristic to automatically assign NUs to each node in the DLG. There can be many schemes to do this assignment.

One, we physically allocate NUs to each graph node. As the size of floor plan grows, this process becomes cumbersome. Two, we can statically assign each type of NU to a particular type of graph node. For example, we can assign workstation NUs to be deployed only on office floors (and not, say, in labs or in the corridors). This is an improvement over the previous scheme as it is more general in application and is hence an useful heuristic. The third approach, the one we use, is adapted from the attraction points characteristic defined in [FHR04]. This approach uses a mixture of static and probabilistic assignment.

Definition: Affinity factor is defined as the probability with which an NU type

Node type ↓, Vertex →	**1**	**2**	**3**
NU_{PDA}	0.6	0.2	0.2
NU_{WS}	0.6	0	0.4
NU_S	1	0	0
...	...	...	...

Table 6.3: Affinity factors.

is present (or attracted) to a node in the DLG. For non-mobile NUs, *af* represents the areas where such nodes are deployed and is an assignment of NUs to DLG nodes. A mobile NU may have more than one area of deployment due to its mobility. For mobile NUs, *af* captures the probability of the NU visiting a particular node in DLG. We define, for each NU type, either a static allocation of number of NUs per graph node (or nodes) or we attach to each NU type a probability with which it would *exist* in some node.

Nodes in the deployment graph are typed with *af* of each NU type. Table 6.3 shows an example listing of affinity factors for NUs (PDAs and workstations) for the deployment layout in Figure 6.3. In the case of static deployments, *af* is represented as a fraction of the total nodes deployed. In the mobile case, *af* represents a probability. So, for mobile nodes, the sum of all the *af* for an NU need not be equal to the number of nodes.

Figure 6.4 shows the XML schema for input parameters. The $\texttt{WIND}_{\texttt{wlan}}$ tool uses the schema to verify input parameters for correctness and conformance. The XML file and example input parameters are given in Appendix A.2.

6.2.3 Network and optimization constraints

Constraints in our tool are of two types: constraints imposed by NU application scenarios and constraints imposed on topology. The first type of constraints are used to check whether application scenarios in each NU is satisfied. These constraints can

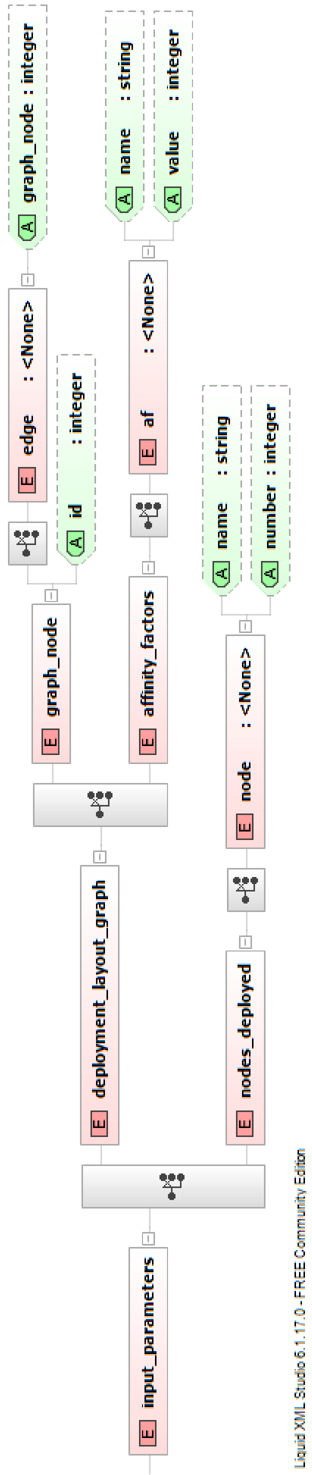

Figure 6.4: XML schema for input parameters.

be bandwidth requirements demanded by the applications running in the network. Or, delay constraints in the form of number of hops. As $\texttt{WIND}_{\texttt{wlan}}$ is a static topology planning tool, we can only use hop count as a measure of delay accumulated in the path from source to destination for an application.

Second, we have constraints that are imposed on the topology construction algorithm. The AP-assignment problem for heterogeneous traffic discussed in the previous chapter gives us the mechanism to develop an abstraction function, or link specification function. This is used to capture the capacity of links in terms of number of traffic flows. Other constraints such as a bound on the number of infrastructure or relay NUs deployed (infrastructure cost) can also be enforced. Another constraint is the optimisation of relay NUs across adjacent nodes in DLG in order to reduce costs.

6.2.4 Outputs: Network topology graphs and statistics

The network topology is output as a graph. Each node in the graph is an NU and every edge in the graph corresponds to a link between two or more NUs. Edges are annotated with their usage and amount of bandwidth available. Each NU_{Relay} is also annotated with its usage. Our aim is to integrate the tool (described later) with a simulator for automatic validation of the topology. Hence, the tool outputs topology in a format that is readable by the simulator.

Figure 6.8 shows an example output topology. Switches [1, 2] and AP [1] correspond to DLG node 1 (in Figure 6.3). Switch [4] and AP [4] correspond to DLG node 2. Switch [3] and APs [2, 3] correspond to DLG node 3.

6.3 Composite Unit (Building block)

We now define a *Composite Unit (CU)* which is the basic building block in the topology construction algorithm.

Definition: A CU is a virtual network element constructed for the purpose of aggregating nodes, or a branch of a network, and their properties. A CU is recursively defined as an amalgamation of one or more CUs and/or NUs. The properties of a CU are the function of properties of its child CU/NUs. Formally, a CU is defined as:

$$CU = (CU' \mid NU)^{+}$$

where, CU and CU' are composite units. NU a node unit.

CU is a tool for network abstraction. This is also useful in understanding the effect and properties of the subnetwork of network elements. The CU construction mechanism is explained with an example in Section 6.4.1.

We consider topologies to be in the form of graphs. The network topology is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where, $\mathcal{V}$ are client and relay NUs and $\mathcal{E}$ are links connecting NUs to form the network. A CU is then a sub-graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ of network topology $\mathcal{G}$ where, $\mathcal{V}'$ ($\subseteq \mathcal{V}$) are NUs forming the CU and $\mathcal{E}'$ ($\subseteq \mathcal{E}$) are associated links. Note that a *leaf* CU in the topology graph will be formed out of only NUs and each CU can be recursively expanded to form a graph of only NUs. From an implementation perspective, CU is an analogy of Container classes in Object-Oriented Programming [Str91]. In our implementation, a CU class is used to capture CU properties and store its child objects. In Section 6.4.3, we present the class definition of CU and discuss its properties.

6.4 Topology construction

We first discuss the topology construction with an example. Then, we present a module-level overview $\texttt{WIND}_{\texttt{wlan}}$. We then discuss the functionalities of CU before discussing the details of the algorithm.

6.4.1 Illustrative example

We explain the topology construction with an example office deployment. The example builds a network topology for seven PDAs and seven Workstations deployed in an office. The node properties are defined in Table 6.2. The office floor plan and its corresponding deployment layout are shown in Figure 6.3. The office layout forms a 3-node DLG with one node each for the two floors and one node forming the connecting corridor. The Workstations are deployed only on the floors. The PDAs may be present both on the floors as well as in the corridor as PDA nodes are mobile. The affinity factors for the nodes are defined in Table 6.3. Both type of nodes run an application accessing some service from a server NU_S.

Figure 6.5 shows the progression of the *computeCU*() algorithm (in Algorithm 1). First, NUs are deployed per node in the deployment layout graph based on affinity factor and number of nodes. The algorithm goes through three recursions. In the first stage, CUs for each DLG node are formed. Relay nodes (in our case, APs) are deployed as required. In the second stage, CUs in a graph node are consolidated. In graph node 3, all CUs from stage one are collapsed into one CU. In graph node 1, CU_2 and CU_3 are merged together to form CU_7. The server node NUs in CU_1 has exhausted all available bandwidth on the link and is not consolidated at this stage. In stage 3, all CUs are brought together into root CU_{10} to arrive at the final topology of the network.

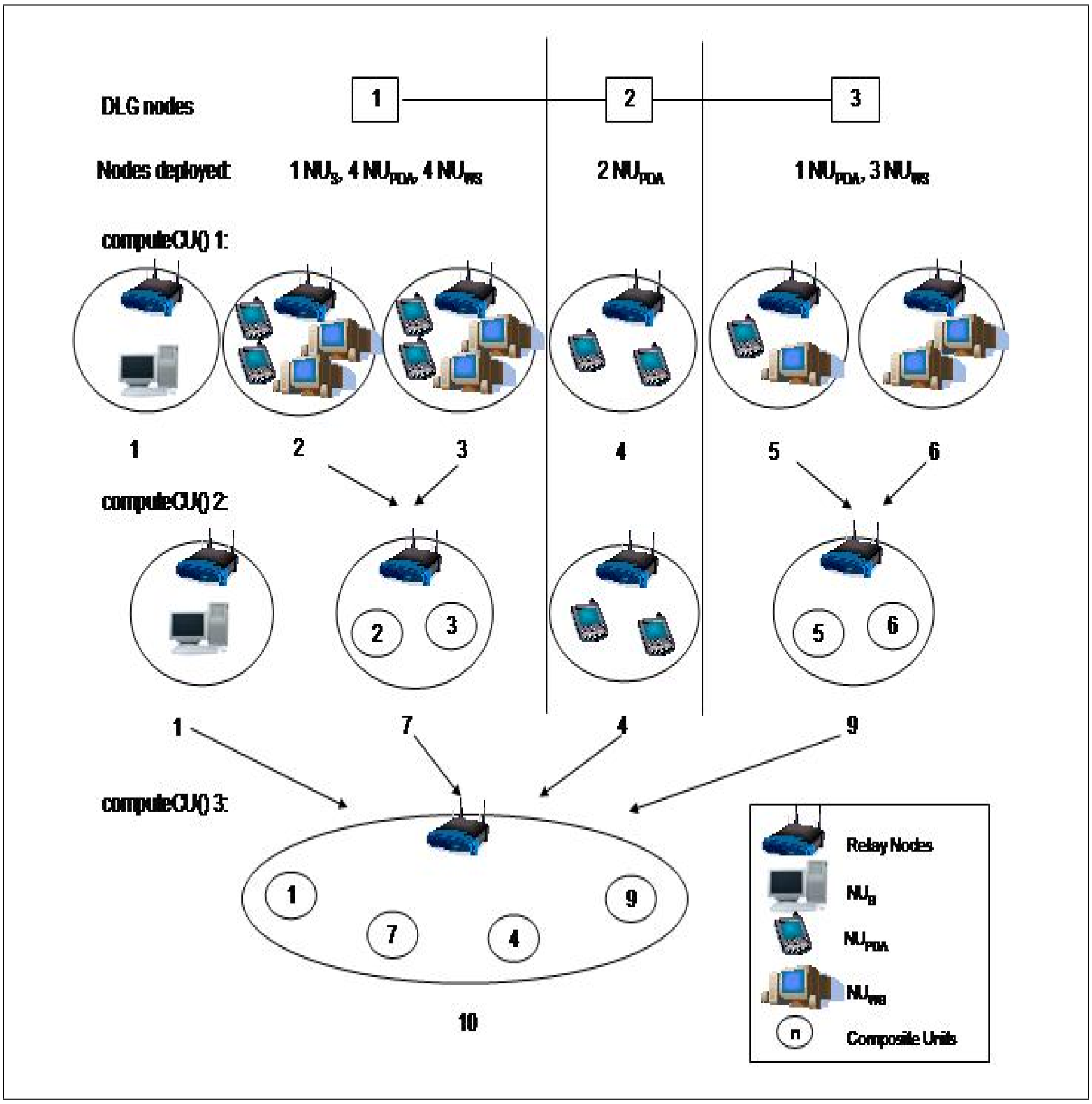

Figure 6.5: Function computeCU() example.

6.4.2 Module-level overview

The tool consists of three modules (Figure 6.6): 1) *Preprocessor* for input configuration, 2) *Compute CU* for topology construction and 3) *Output topology* for printing topology and statistics.

1) The *Preprocessor* module configures the inputs for use by the topology generator. It takes as input:

- information base for NU and link models,
- deployment layout of the network,
- affinity factors of NUs and
- number of elements of each NU to be deployed..

The *Preprocessor* constructs the following for the *Compute CU* module:

- *NU list* : Using the NU models, the deployment layout graph and the affinity factors associated with each NU, the list of NUs to be deployed in each node in the DLG is calculated.
- *CU rules*: The topology generator also requires a set of rules while generating the topology. The rules define the criteria for CU formation and relay node selection.

2) The *Compute CU* module is the topology generator which uses NU list and CU rules to construct the topology. The *Compute CU* module works on a list of NUs/CUs and uses a set of rules during CU construction. The module takes in one more input, the optimization criteria, which defines the QoS constraints on the construction process. At each stage in the construction process, the module considers the CU rules and checks for criteria satisfaction. The module works as follows :

1. Given an NU list, the function calculates the NUs to be used for construction of the CU (using CU rules).

2. The properties of the constructed CU and the NUs are set using the criteria defined for optimization.

3. The algorithm continues until all nodes in the list are exhausted.

4. The generated list of CUs is then recursively fed to the module again, until the root CU of the topology is generated or the list of CUs cannot be processed further.

3) The constructed topology is then processed by the *Output topology* module for input to a simulator for validation. The simulation is done using OPNET Modeler and the output topology formats are modeled after the topology input formats of the Modeler [OPN]. This allows us to directly use the output of $\texttt{WIND}_{\texttt{wlan}}$ as input to the simulator.

We now discuss the properties of the CU class before presenting the algorithm details.

6.4.3 CU properties

Figure 6.7 shows the CU class. Each CU class has an id and a name (in the case of an NU). The CU also stores the relationships between child objects. The CU class maintains a list of child CUs and properties of application scenarios and links. These properties, or relationships, are useful during topology construction. The properties are:

- *childList*: The list of child NUs and CUs.

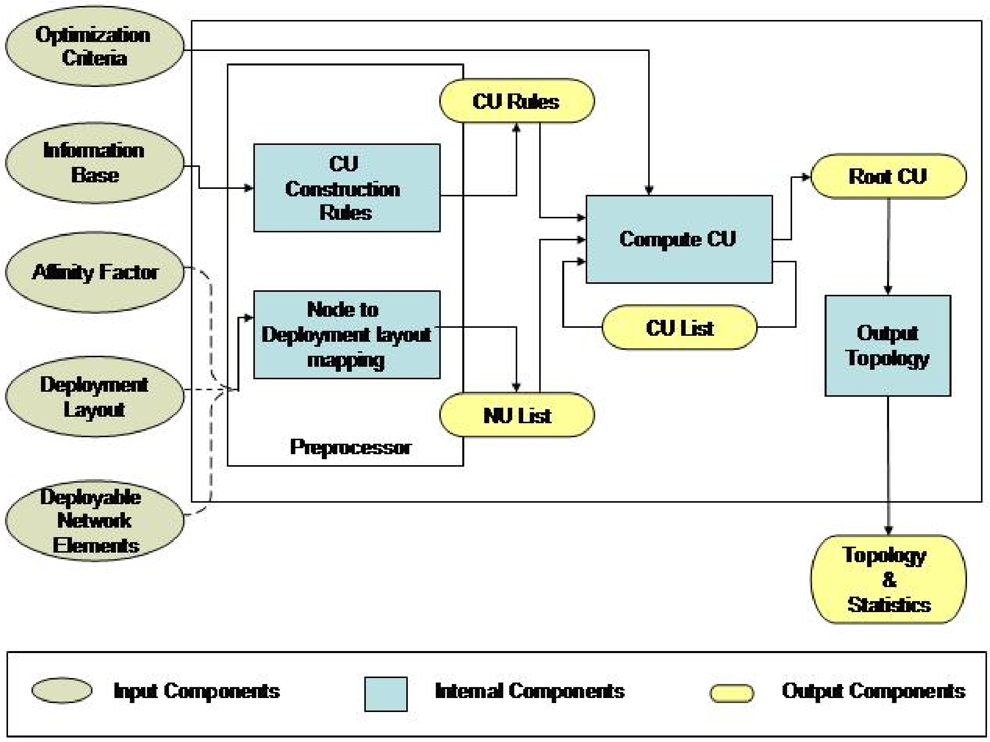

Figure 6.6: $\texttt{WIND}_{\texttt{wlan}}$ framework.

- *linkList* = *getUnusedLinks* $((CU' \mid NU)^k)$, where, k is the number of CU/NU units in CU. Function *getUnusedLinks()* returns the unused links of underlying network elements (NUs and CUs in *childList*). Once the capacity on a link is exhausted, a link is considered to be used (or reserved). Hence, the function computes the links of child CU/NUs on which capacity is still available. It also stores information on the amount of bandwidth available.

- *asList* = *getUnfulfilledTraffic* $((CU' \mid NU)^k)$, where, k is the number of CU/NU units in CU. Function *getUnfulfilledTraffic()* returns the unfulfilled AS traffic of underlying network elements (NUs and CUs in *childList*).

For each AS, the Compute CU module finds the shortest path between source and destination network elements. In the process, we need to keep track of

```
class CU

private:

 int id;
 string name;

 double outLoadTotal;
 double inLoadTotal;

 LinkList* linkList;
 ASList* asList;
 CUList* childList;

public:

  void print(int tab);
  CU(NodeType* nt);//NU constructor
  CU(); //CU constructor

  ASList* getASList();
  LinkType* getBestLink();
  void addChild(CU* cu);
  void rstChildProperty(LinkType*);
  void setProperty();
  void resetLinks(LinkType*);
  void resetTraffic();

  LinkList* getUnusedLinks();
  ASList* getUnfulfilledTraffic();
```

Figure 6.7: CU class definition.

AS traffic that is *not fulfilled* yet. Unfulfilled traffic is defined as traffic which is not destined for a descendant of the current CU. Traffic of underlying CUs which have found a path (that is, both source and destination CUs are descendants of the same CU), are considered to be satisfied traffic. In other words, the function computes a list of *AS* bound for some NU (not a descendant of this CU) which the CU construction process has not yet considered in topology construction. We describe this further while discussing the CU construction process. As discussed in Section 6.2.1, relay nodes do not have any traffic of their own and hence they subsume the traffic of their underlying network elements.

- *outLoadTotal and inLoadTotal*: The unfulfilled load of child CUs is assigned to their parent as its load (using function *setProperty()*).

6.4.4 Algorithm details

The pseudo-code for *computeCU()* is given in Algorithm 1. The *computeCU()* algorithm uses a recursive bottom-up approach for topology generation. Starting from a list of NUs, the algorithm recursively builds the topology by adding relay nodes while keeping constraints satisfied. Given a list of NUs/CUs as a list of CUs (*cuList*), *computeCU()* module does the following (step numbers are given with respect to Algorithm 1):

1. Find common link types among all NUs/CUs in *cuList*. This results in a list of lists; that is, an unordered list of NUs/CUs for each link type (Step 1.3).

2. Calculate $cuList'$ ($\subseteq cuList$) such that, $cuList'$ is the optimal number of nodes that can be merged together. The MAC abstraction function - *maxNodes()*, defined for each link type in *IB*, is used to compute $cuList'$ (Step 1.5). This function calculates the optimal number of nodes that can share a link or be *brought together* given link capacity and traffic characteristics. The function uses AP-assignment solutions from Chapter 5 to compute capacity of the link in terms of a set of heterogeneous nodes. Function *maxNodes()* is defined in Section 6.2.1.

3. Create an object CU (Step 1.9). Add $cuList'$ as children (Step 1.10). Set properties of child CUs and newly formed CU. Assign NU_{Relay} if required.

   Once n NU/CUs children are added to the newly created CU, an appropriate NU_{Relay} is selected from the information base and added to CU as a child (Step 1.12-13). $\texttt{WIND}_{\texttt{wlan}}$ selects infrastructure nodes from a set of relay nodes in the *IB* based on their link properties and constraints.

   A child CU with a relay node having a suitable link can also be selected as a relay node. For example, nodes having multiple links may act as relays.

We first reset the properties of the child CUs, before setting the properties of CU. This is done in order to reflect the traffic scenario at CU. The function *resetProperty()* computes the following properties (Steps 1.14 and 1.15) :

- *Used links*: For all child NU/CUs, set corresponding links used in the merge criterion as a used link.
- *Fulfilled traffic*: Consider traffic from any NU in one child CU destined for NU in another child CU as fulfilled traffic. A source-destination link is formed between the two NUs and this traffic is removed from further consideration.

Now, function *setProperty()* computes the properties of CU as follows (Step 1.16):

- $links$ = Unused links of NU/CUs + Outgoing link of Relay Node. All child CU links that remain after function *resetProperty()* has computed the used links are collected together as $links$. If a relay node has been added as a child CU, its outgoing links are also added.
- *traffic* = Traffic of child NUs/CUs. All child CU traffic that remain after function *resetProperty()* has computed the fulfilled traffic are collected together as *traffic*.
- $AS\_load = \sum_{i=1}^{k} traffic$. The sum of all traffic scenarios at this CU is collected as a statistic.

4. The created CU is added to a new cuList (Step 1.17) and the function recursively calls itself until the root CU is computed or the current cuList cannot be reduced further (Step 1.20). The function then returns the root (or the final cuList).

```
input  : cuList,ib: info base
if sizeOf(cuList) = 1) then return cuList
newCUList ← NULL
L ← linktypes_present(cuList)
forall lt ∈ L do
    cuList_lt ← cuList_lt + {cuList[i], cuList[i].linktype = lt}
    while cuList_lt NOTEMPTY do
        cuList' = lt.maxNodes(cuList_lt)
        // Average load
        t ← (Σ_∀j cuList'[j].total_load) / sizeOf(cuList')
        new cu'
        cu'.child(cuList')
        cuList_lt.remove(cuList')
        newcu_relay = findRelayNode(lt, t)
        cu'.child(cu_relay)
        for cu ∈ cuList' do cu.resetProperty()
        cu_relay.resetProperty()
        cu'.setProperty()
        newCUList.add(cu')
    end
end
return computeCU(newCUList, ib)
```

Algorithm 1: Psuedo-code for computeCU()

The function *computeCU()* is used by *wind()* to compute the network topology. The pseudo-code is given in Algorithm 2. Function *wind()* works as follows:

1. Compute a NU list, *nuList*, of deployable NUs for each node in the deployment layout graph and pass it to function *computeCU()*. The function returns a CU (or a list of CUs); this is collected in another list of CUs, *cuList*.

2. Compute root CU of network using the above list of CUs, *cuList*. If no root CU found, *cuList* forms the top level of topology graph.

3. Print topology by performing a Depth First Search (DFS) on root CU (or each CU in *cuList*).

input : ib: info base, ip: input parameters
$cuList \leftarrow NULL$
`//` G_{DL}`:Deployment layout`
forall $v \in V(ip.G_{DL})$ **do**
    `// af:affinity factor`
    $deployedList \leftarrow \bigcup_{\forall i}(v.af_i * ip.num_{NU_i}).NU_i$
end
$cuList \leftarrow computeCU(cuList, ib)$
$printTopology(cuList)$

Algorithm 2: Pseudo-code for $\texttt{WIND}_{\texttt{wlan}}$

6.5 Implementation

We have implemented $\texttt{WIND}_{\texttt{wlan}}$ using C++. The description language for input parameters and information base input files is based on the XML description formats defined in OPNET. The XML Schema for information base and input parameters are given in Figures 6.2 and 6.4, respectively. The XML files or the schemas and examples used are given in Appendix xml-ex. The Xerces-C++ parser, from the Apache project, is used for processing the XML files [Xer]. This is a validating parser which checks a input XML file with a given schema and is useful for automatically checking inputs. We impose a bandwidth constraint for AS traffic and use the link specification function for AP-assignment.

6.6 Validation

The experiment parameters are as follows:

- 5 PDAs running a VoIP call with an average load of 100 Kbps
- 5 workstations running an FTP client with an average load of 1000 Kbps.
- Deployment layout is as shown in figure 6.3.
- The input parameters are as defined in Appendix A.2.

- Information base are defined in Appendix A.1.
- The affinity factors are defined in Table 6.3.

The topology generated by $\texttt{WIND}_{\texttt{wlan}}$ is shown in figure 6.8. The topology is given as input to the OPNET Modeler (along with node and link models from information base) to validate the generated topology. The average bandwidth was 98.77 Kbps for PDAs and 998.38 Kbps for Workstations.[1] The generated topology satisfies the given constraints and shows the feasibility of the $\texttt{WIND}_{\texttt{wlan}}$.

The use of composite unit as an abstraction mechanism is useful in analysing the generated logical topologies. Since CU represents every object in the network from an NU to the entire topology, subset of the deployment layout (represented by sub-trees of the topology as CU) can be analysed separately. Also, this allows scaling for larger networks by allowing reuse of the generated topology.

6.7 Comments

The $\texttt{WIND}_{\texttt{wlan}}$ tool was used to construct WLAN topologies using the AP-assignment solutions. $\texttt{WIND}_{\texttt{wlan}}$ uses a bottom-up heuristic approach to construct the topology using a small set of inputs. The tool constructs topologies up till the relay or gateway node of the WLAN. This is appropriate as then each WLAN can be designed in a modular manner. With a deployment area with similar floor plans, the WLAN designs can then be reused. The AP-assignment solutions from Chapter 5 were used as a basic construct mechanism in the form of link specification functions. We introduced composite unit as an abstract building block for topology construction. CU is also useful in understanding the properties of subnetworks.

An important aspect of the tool were the XML-based inputs and outputs. The

[1] The $\texttt{WIND}_{\texttt{wlan}}$ framework and WLAN experiments were presented at ICPWC [RI05] and Infocom [RI06a].

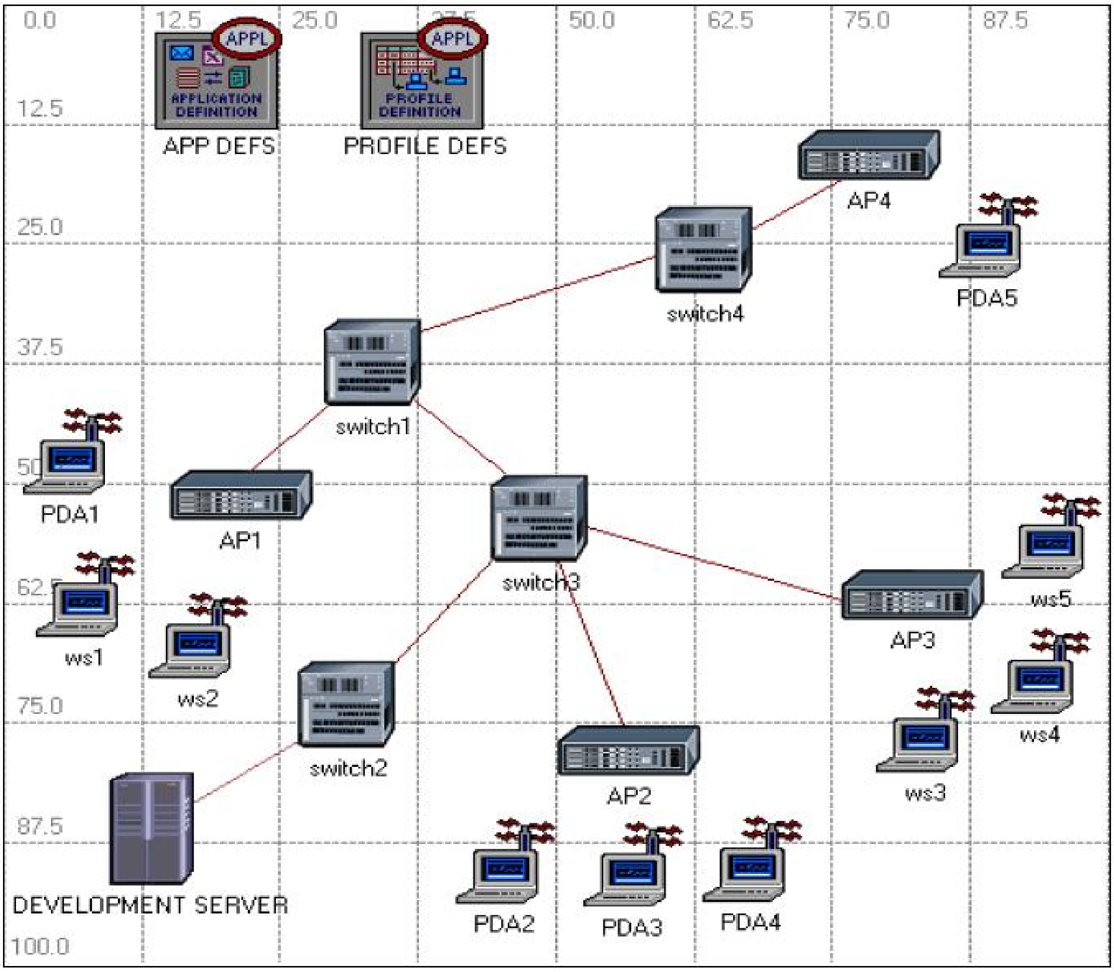

Figure 6.8: Constructed WLAN topology.

schemas for these data are based on the OPNET Modeler data formats. This allows us to connect the tool to the simulator, with topology as intermediate data, and validate topologies.

In the next chapter, we extend the $\texttt{WIND}_{\texttt{wlan}}$ module to design wireless mesh networks. This is then used to construct the backhaul topology for WLANs. We show that with some modifications to the topology constructor and using an external optimiser, the scope of $\texttt{WIND}_{\texttt{wlan}}$ can be extended to designing wireless mesh networks.

Chapter 7

Backhaul infrastructure design for Wireless Mesh Networks

In this chapter we discuss the design of wireless mesh networks under the ambit of backhaul networks for WLANs. In Chapter 6, we had discussed capacity-constrained topology design of WLANs. The tool, $\mathtt{WIND}_{\mathtt{wlan}}$, constructed topologies up till the relay or gateway node of the WLAN. Providing external connectivity to the WLAN, that is, extending the design problem to beyond the gateway and backhaul connectivity planning, is the next important step in the design process.

Backhaul, for office WLANs, have generally been through wired legacy connections (Ethernet) which connected with the rest of the network infrastructure. In this work, we make an important assumption that the backhaul[1] connectivity is provided with the help of a wireless infrastructure. The reasons and advantages that were presented in Chapters 1 and 2 for a WLAN also hold good for backhaul wireless networks. But additionally, a suitable technology is necessary to adequately replace the large bandwidth capability of wired networks.

Wireless mesh networks, in different forms, are increasingly being deployed as a

[1]In this context, we use the term backhaul to refer to a backbone network or a wireless mesh network.

backhaul infrastructure solution. In this chapter, we examine the case of backhaul infrastructure design for a campus wide network of WLANs.

We present an overview of the problem and discuss mesh networks and the issues in planning in Section 7.1. We then present the mesh network design problem and design approach in Section 7.2. Next, we present the design problem formulation in Section 7.3. We formulate the problem as a mixed-integer linear programming problem and present the assumptions and algorithmic details. We have extended the WIND tool to encompass the design of mesh networks and, in Section 7.4, we detail out the WIND for WMN module and its interaction with the WLAN module from Chapter 6. In Section 7.5, we present the implementation details and discuss results.

7.1 An example wireless mesh network

Wireless Mesh Networks (WMN) are gaining popularity as a solution to provide wireless backhaul in campus and metropolitan environments.

Figure 7.1 shows a WMN in an urban scenario. Each cluster of buildings can be considered as office buildings or apartment societies. The WMN connects these clusters to the base station which is connected through a gateway to the external cloud (for this scenario, the base station can also be considered to be co-located with the gateway). Each cluster is also connected with others using a multihop wireless network architecture. Each building in a cluster is assumed to have a WLAN. In the architecture figure, this WLAN is abstracted and considered as *one node*. This node, called the AP node, is the *root AP node* in the topology of the building's WLAN network. The node represents the properties of its underlying WLAN (demands and links) and acts as the *building's gateway*.

Router nodes, termed here as mesh nodes, are placed in the network to provide

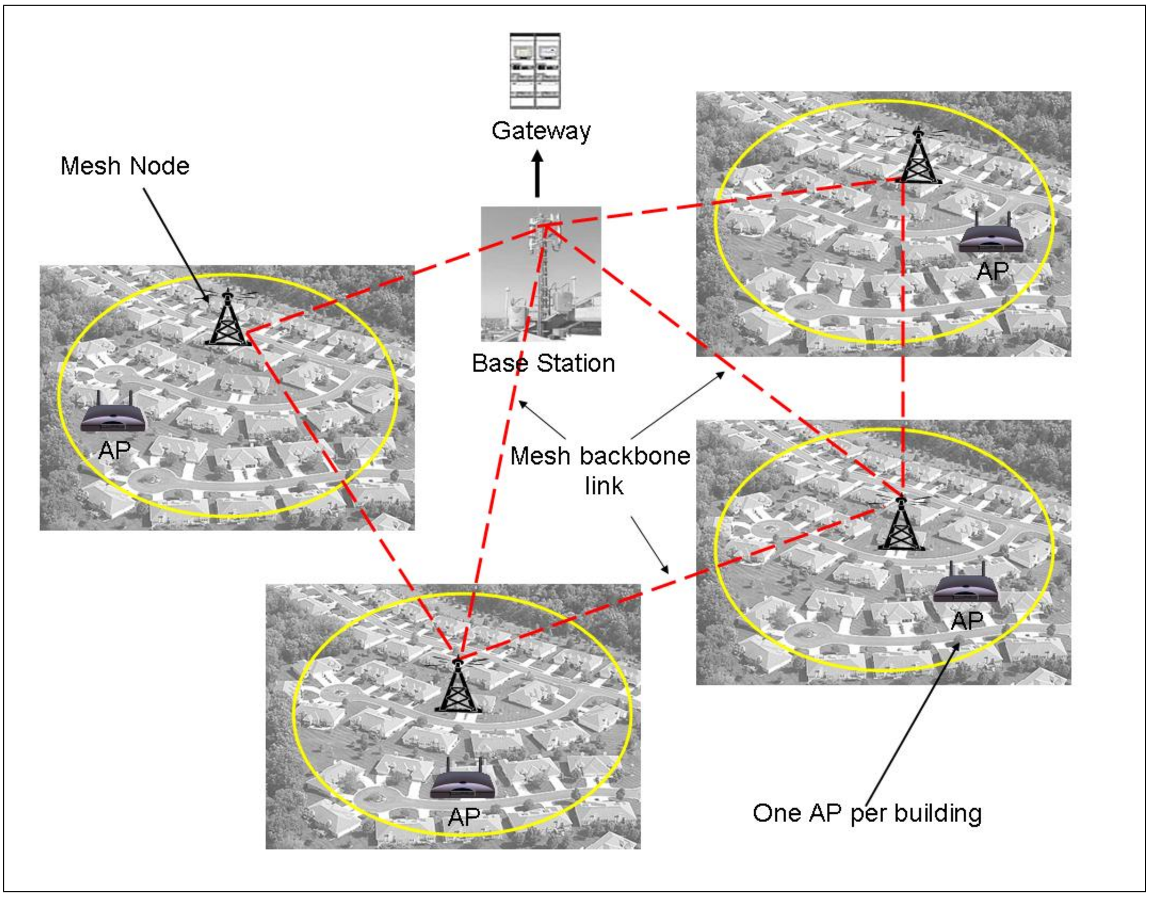

Figure 7.1: An example mesh network. The ovals designate the coverage area of the mesh nodes.

gateway connectivity to APs. This architecture is different from the single-hop architecture of WLANs as each AP is assumed not to have connectivity directly with the base station (in the case of WLAN, each client node is connected to the AP with a single hop). The APs are connected through the mesh nodes to the base station in a two-tiered architecture. One or more mesh nodes may be present in each cluster to provide connectivity to APs in the cluster depending on requirements. Some mesh nodes may even act purely as routers for the mesh nodes themselves. That is, no AP is directly connected to them. Each mesh node is connected to other mesh nodes and/or the base station with one or more mesh backhaul links. These links, called *mesh links*, are for inter-mesh node traffic. Links connecting APs to mesh nodes are called *mesh-AP links*.

This topology may also be slightly modified to accommodate other scenarios. The APs may themselves be connected to the mesh nodes through multiple AP hops. More than one gateway may be present in the network, hence providing multiple routes to the external cloud. The APs and the mesh nodes may have one or more radios for segregating connectivity and improving the capacity of the system. An AP may use multiple radios to communicate with the end-users and the mesh nodes.

The mesh network deployment scenario considered in our example is generic enough to be applied to different usage scenarios. This network can be considered as a backhaul for adhoc networks itself and hence will be useful as a backhaul for hotspot scenarios. Considering the base station as a sink node, the network can also be thought of as a backhaul for a sensor network.

Another important advantage is that, since a mesh network itself can be considered as a special case of an adhoc network, routing protocols for adhoc networks are also applicable in this case. This is because a mesh network can be considered as a static adhoc network of routing nodes with a self-formed topology. Most practical routing algorithms are directly taken from work done for adhoc networks. A comprehensive survey of deployment scenarios, routing and other issues in mesh networks is given by Akyildiz and others [AWW05].

Now, some important issues arise when we consider a WMN deployment: the number of mesh nodes to be placed, the placement of mesh nodes, the backhaul topology constructed and the constraints placed by the underlying WLANs. All these issues affect the cost of the network deployment as well as its performance. It is important then to have mechanisms for the design of such networks. We discuss this in the next section and frame a design problem for a class of backhaul mesh networks.

7.2 Mesh network design problem

With respect to the example in Figure 7.1, given an AP deployment scenario as discussed, the aim of network design is to provide *optimal* backhaul connectivity in the form of a mesh network.

In order to do this, we need to define two things: the properties of the mesh network to be designed and the optimality criteria for the network. There are a few constraints now in the design, or planning, of mesh networks. The first issue is the connectivity provided by the mesh backhaul to each AP; that is, the availability of a mesh node for an AP to connect to the backhaul network. The second issue is that the backhaul mesh network should have the capacity to satisfy the demands placed by underlying networks. Both issues together lead to the following constraints on network design:

- Connectivity constraint: mesh nodes should be deployed such that all APs in the deployment area are connected.
- Capacity constraints: APs and their underlying networks place demand constraints on the backhaul which have to be satisfied.
- Optimisation criteria: Optimisation of number of mesh nodes deployed to reduce costs.

A related issue to connectivity constraint is the issue of power control; that is, the mesh nodes should also be deployed close enough to APs in order to satisfy constraints on transmit power. Transmit power and distance are significant issues when the considering cost of wireless deployments.

As part of capacity constraints, APs and their underlying networks may have demands between different WLANs (inter-AP demands) or demands may be with the external world through the gateway. Both these demands have to be addressed.

Deployment of each mesh node has its associated fixed and variable deployment costs. The cost of each mesh node includes the fixed cost of the mesh node itself plus the cost of establishing mesh links. Each mesh link, which may require a separate radio or channel, has a fixed cost for establishing the link. Also, the transmission distance of a mesh link (distance between the mesh nodes forming the link) plays a role, as transmit power determines the variable cost. Both these link costs have to be captured by a link cost function. The transmit power required adds up to a significant cost over long term deployments and so forming short distance links is a preferable design constraint.

The optimality criteria should then capture these node and link costs. The criteria is defined as follows: 1) Minimise number of mesh nodes and 2) Minimise number of mesh links formed. While these costs have to be minimised it should be noted that this is to be done while preserving the capacity and connectivity constraints. Adapting the definition from Chapter 3, we can now state the problem formally:

Mesh network design problem

Given deployment layout, AP nodes deployed and their characteristics,

Construct backhaul topology,

Subject to demand constraints,

While minimizing network infrastructure (mesh nodes and links).

7.2.1 Design approach

While planning a network, with respect to our wireless example, the following scenarios can be envisioned:

- Given locations of possible APs, where should the mesh nodes be located?

- Where and how should the interconnection between the mesh nodes be, taking into account the locations of APs?

- Once the location of mesh nodes and topology graph is computed, how can the links be expanded such that the traffic demand is met?

There is an available body of work in wired network design that can be applied in this context (discussed in Section 2.3). While this design problem falls in the same class of network design problems as encountered in wired as well as cellular networks [PM04], the difference arises when we consider the node capabilities and the associated constraints and cost-functions. For example, the wireless nature of the links (including backhaul links) gives rise to cost-functions not encountered in other types of networks, and also the use of multi-hop wireless transmission results in scheduling constraints.

We frame the network design problem for WMNs as an optimization problem. Our aim is to construct the WMN backhaul topology as well as find the optimal location of mesh nodes while satisfying imposed constraints. The deployment scenario is a Typical Urban (TU) setting. In the deployment layout, the AP node locations and the demands are given. Potential mesh node locations are also given. Under various user defined constraints (discussed below) optimal mesh node locations and links are chosen and a topology is formed.

Next, we describe the network model which captures the network details and assumptions described in the above example more formally. Then, we describe the problem formulation and implementation details and discuss the WIND extensions developed to address this problem.

7.2.2 Network model

For the design problem stated above, we consider the following urban scenario (Figure 7.1). Each building in the deployment area, has an AP which provides the connectivity between the clients inside the building and mesh backhaul.

A traffic demand is defined as in Section 6.2 for WLANs. Demands at each WLAN under an AP node can be of three types. One, it can be a demand source (demands generated within the underlying WLAN) which requires external[2] nodes (servers) to service it. Two, it can be a demand sink (demand services provided) which services demands for external clients nodes. Three, it can be a peer-to-peer demand in which client nodes in different WLANs require connectivity (for say a VoIP call).

An AP has two radio links, one for internal connectivity to the WLAN and one providing connectivity to the mesh backhaul. The internal link is assumed to be an 802.11 device, while the external link is a mesh-AP link. Note that, we are mainly concerned with the demand generated at each AP. Hence the type of the internal link or the underlying sub-network topology is *irrelevant* to the problem as long as there is no overlap in the frequency allocated to the links (in order to avoid interference). We will discuss later how this assumption goes with our $\texttt{WIND}_{\texttt{wlan}}$ discussion in Chapter 6. The mesh-AP link is created by the AP associating itself with a *suitable* mesh node.[3] The *capacity* of a link is defined the same way as defined for $\texttt{WIND}_{\texttt{wlan}}$ in Section 6.2 (Table 6.1), that is, as number of bytes per second.

The mesh nodes may have multiple antennas in order to communicate with both APs as well as other mesh nodes. Each AP associates itself with only one mesh node. Multiple APs may associate with a mesh node and there may be an upper

[2]Any node in the rest of this mesh network or accessible through the gateway is considered to be an external node

[3]The suitable node may not be the nearest mesh node and is determined by overall system cost constraints.

bound set on number of APs per mesh node. All associated APs are assumed to share a single mesh-AP link and hence its bandwidth. There is a similar upper bound on the number of mesh links per mesh node. The base station is considered to be a mesh node with gateway connectivity.

7.3 Node location and topology construction problem

We first present the assumptions and definitions used in the formulation. The *optimal mesh node placement* requirement makes this problem a Mixed-Integer Linear Programming (MILP) problem. The MILP formulation for this problem is given in algorithm 4.

7.3.1 Assumptions and definitions

First we make some assumptions on node deployments:

- The AP node locations are given.
- The traffic demand at each AP node is given.
- Potential mesh node locations are given.
- Mesh links are directional links with each mesh link connecting only two mesh nodes.

Definitions of nodes and their properties are as follows (refer Figure 7.2):

- An AP is defined as $\{w \mid w \in W, w = (x, y, r, D)\}$. W is the set of APs in the deployment layout. The properties of node w are the co-ordinates (x, y), the transmission radius r and the demands D.

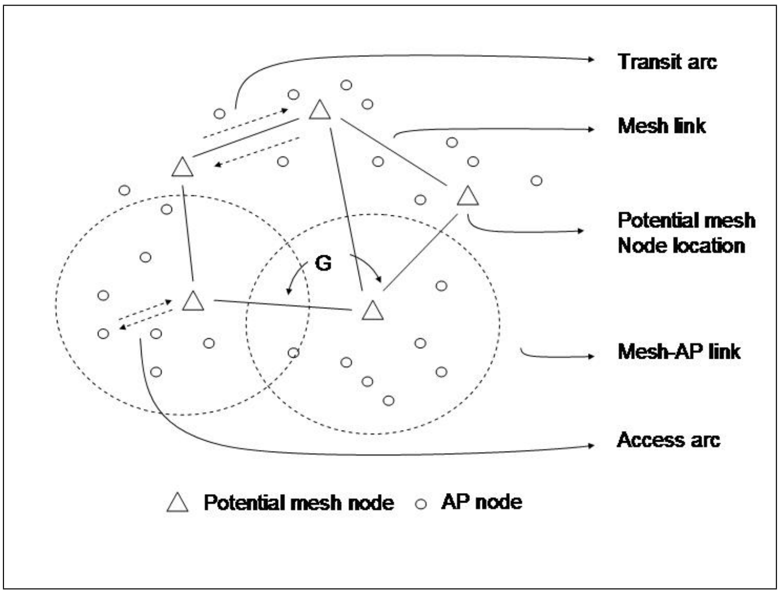

Figure 7.2: Mesh network: definitions.

- Demand from AP w to AP w' is defined as $h_{ww'}$. The atomic unit in each demand between a pair of APs is considered to be discrete (for example, an IP packet) and this information is used for routing purposes. The sum of all demands from AP w is then defined as, $D = H_w = \sum_{w'} h_{ww'}, \forall\ w'\ in\ W$.

- A mesh node is defined as $\{v \mid v \in V, v = (x', y', r', G)\}$. V is the set of potential mesh nodes from which the optimal number of mesh nodes are chosen. The properties of node v are the co-ordinates (x', y'), the transmission radius r' and the bound on number of links G. As the mesh nodes merely act as relay nodes they do not have any traffic parameters.

- A potential link is defined as $\{e \mid e \in E, e = (M)\}$. E is the set of all possible links in the system. Each link is computed as follows: The transmission ranges of AP and mesh nodes are given. For each AP node, all mesh nodes within its transmission range are considered to be potential for formation of a mesh-AP link. For each mesh node, all other mesh nodes in its transmission range are considered to be potential for formation of a mesh link. These links are formed only between mesh-AP and mesh-mesh pairs. AP-AP links, even if possible, *are not considered*. Each link is *bidirectional* and M is given as an upper bound on capacity of link e.

- Each link e is split into two directional arcs for computational purposes. This is then used to compute the direction of flow of each demand. A link can be between an AP and a mesh or between two mesh nodes. Mesh-AP links are split as *access arcs*, $\{f \mid f \in F\}$, where F is the set of all possible access arcs. Mesh links are split as *transit arcs*, $\{t \mid t \in T\}$, where T is the set of all possible transit arcs. Therefore, $|T| + |F| = 2 * |E|$.

7.3.2 Determining potential links

Our first assumption was a distribution of AP and mesh nodes in the deployment layout. We now need to determine the potential links between the nodes, that is, the set of links E. The design algorithm will require these potential links in order to and compute the final topology. We define a simple function that uses the transmission distance and the co-ordinates of each node to compute the list of potential links.

Consider two nodes $z = (x, y, r, \ldots)$ and $z' = (x', y', r', \ldots)$. Parameters of the nodes are as discussed in assumptions and consist of only those that are common for both AP and mesh nodes.[4]

If z and z' are mesh nodes, a mesh link $e_{zz'}$ is formed between the two nodes and added to the set of potential links E if and only if, $\sqrt{(x - x')^2 + (y - y')^2} < r\ (or\ r')$. If z is an AP and z' a mesh node, an mesh-AP link $e_{zz'}$ is formed between the two nodes and added to the potential set of links E if and only if, $\sqrt{(x - x')^2 + (y - y')^2} < r$ (assuming $r < r'$, that is, a mesh node has a longer

[4] Ignoring rest of the node parameters does not affect this discussion.

transmission range that an AP)

We had assumed before that the transmit radius of each node is available. A different approach can be taken by using transmit power. In a perfect channel with no errors, we can compute the transmission distance by computing the path loss. Or, we can extend it for different channel conditions by characterising the signal attenuation.[5] This may change the set of potential links but the design algorithm itself is not affected.

7.3.3 Node and link costs

The costs associated with installing nodes and links are given as:

- φ_v: cost of installing mesh node v.
- κ_e: cost of installing link e.

The objective of the design algorithm is to minimise the cost of node and link deployments and it uses the above costs for computing the total cost of deployment.

Now, we need to specify a link cost function for computing κ_e. This function captures the cost of establishing a link at any node. The cost of a link has two parts: 1) the cost of the hardware for establishing a link and 2) the cost of the transmit power required to establish the link (which in turn determines the power requirements). The transmit power required by a node to establish a link can be either fixed or variable.

So, we define two link cost functions to capture the effect of power control. Assuming the establishing cost of a link e is σ_e, one function uses fixed transmit power

[5] An equation which includes path loss and shadowing is given by (in dB): $(\Psi_{rcv})_{dB} = (\Psi_{xmt})_{dB} - 10\eta \log_{10}(d/d_0) - \xi$ where, $(\Psi_{rcv})_{dB}$ and $(\Psi_{xmt})_{dB}$ are the received and transmit powers respectively. Given d as the transmit distance, d_0 as the reference distance and η as the path loss exponent, the second factor on R.H.S. is the path loss. ξ is the shadowing component. Now, given a received SNR threshold β, we can compute potential links between two nodes [KMK05].

and the other variable transmit power (depending on the transmission distance) to compute the cost. They are defined as follows:

- Fixed transmit power: In this function we assume a fixed cost for each link based on the transmit radius of the node. Cost of link is then:

  $\kappa_e = \sigma_e \ + \ ceil \ (\ r_e^2/\rho_e \ )$

  where,

  r_e is the transmit radius of node forming link e,

  and ρ_e is a multiplicative cost factor.

- Variable transmit power: In this function we assume a variable cost for each link based on the actual transmission distance between the nodes forming link e. Cost of link is then:

  $\kappa_e = \sigma_e \ + \ ceil \ (\ tx\_dist_e^2/\rho_e \ )$

  where,

  tx_dist_e is the transmission distance,

  and ρ_e is a multiplicative cost factor.

7.3.4 Objective function

The objective function is defined as follows (Algorithm 4):

minimize $\boldsymbol{F} = \sum_e \kappa_e u_e + \sum_v \varphi_v s_v$

where,

u_e is a binary variable specifying whether link e is `ON` or `OFF`.[6]

s_v is a binary variable specifying whether node v is `ON` or `OFF`.

The objective function $\boldsymbol{F}$, which is the sum of cost of all nodes and links that are `ON`, needs to be minimised.

[6] A link/node is `ON` if it is selected for the final mesh topology (turned `ON`) and hence is considered when calculating the total cost. It is `OFF` otherwise.

7.3.5 Demand and link constraints

Two types of constraints are imposed on the objective function. One, demand constraints which guarantee demand satisfaction for traffic in the network and two, bound constraints on the number of links per node. The constraint numbers below are with reference to Algorithm 4.

The demand constraints are as follows:

- The volume of traffic demand flowing on each link should not exceed the capacity of the link. This is expressed by constraints 1 and 5 in Algorithm 4.
- Traffic demands of each AP should be satisfied by the computed mesh network. That is each demand should have a path from source AP to destination and entire demand should be satisfied. This is expressed by constraints 2, 3 and 4 in Algorithm 4.

Also, for each mesh node, there is an upper bound on the number of links present at the node. This is expressed by constraint 6 where the number of links cannot exceed G. There may also additionally be an upper bound on the number of demands per AP.

7.3.6 Example: 6 AP and 5 mesh nodes

Figure 7.3(a) presents an example 6 AP node deployment scenario. The number of potential mesh nodes is 5. The transmission range of AP and mesh nodes are 1.5 and 2 respectively. The upper bound on mesh links is $G = 4$. There are 7 demands between the nodes (each of 100 Kbps): $< 1-2 >, < 2-5 >, < 2-6 >, < 3-4 >$ $, < 3-6 >, < 4-6 >$ and $< 5-2 >$.

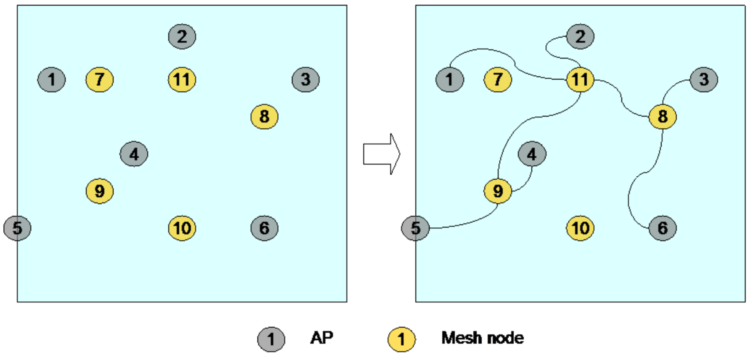

Figure 7.3: WMN topology: (a) 6 AP, 5 mesh node deployment scenario. (b) Output topology

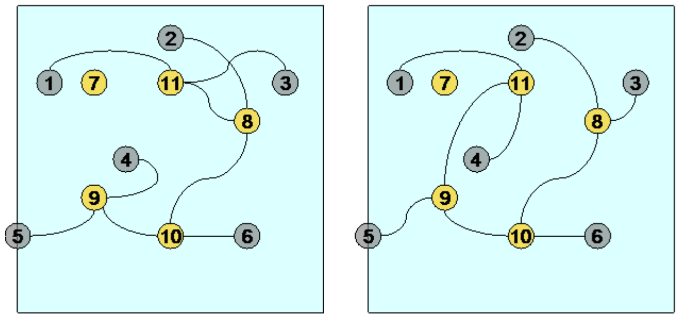

Figure 7.4: Output topologies for a fixed power cost function with change in demands.

The output topology is shown on the right in Figure 7.3(b). From the given 5 potential nodes, the network is connected using just 3 nodes. Using a fixed power cost function, Figure 7.4 shows the changes in topologies with change in demands.

7.3.7 Remarks

The approach the optimiser takes in solving this problem has been discussed in Chapter 2 and Section 7.2. Modeling of nodes and links as binary variables (taking values `ON/OFF`), for finding the optimal combination, results in this problem being a Mixed-Integer Linear Programming (MILP) problem. Solvers such as CPLEX directly support executing such programs [Sol].

The approach taken by nearly all solvers is the Branch and Bound (BB) method to solve MILP problems (or its enhancement, the Branch and Cut) [PM04]. The BB algorithm enumerates all candidate solutions (the search tree) by using upper and lower estimated bounds of the objective function being optimised.

Scheduling and routing

The issues of scheduling and routing traffic over the mesh links still remains. Scheduling a multi-hop wireless network is an np-hard problem [KMK05]. As discussed before, the node location and topology construction is also an np-hard problem [PM04]. Hence, heuristics have to be used to study scheduling along with our design problem. We make an assumption that each directional link between the mesh nodes can transmit and receive simultaneously hence simplifying the scheduler. Further to this, a brief discussion on scheduling constraints for our problem formulation is given in Appendix B.2.

The design algorithm finds a route for each demand flow of an AP. As each demand flow is considered to be in discrete units, a single demand between a pair of nodes can be routed through multiple paths in the network. In our case, this depends on the link capacity and cost constraints. The routing pseudo-code is given in Algorithm 3.

The algorithm uses an all pairs[7] shortest path approach to find the mesh nodes and links which result in the lowest cost (using function cost_of_shortest_path()). At each iteration, the constraints specified above are checked. An important heuristic is the initialisation $cost_{min} \leftarrow COSTMIN$. This is used by the optimiser for pruning the search tree and is discussed in Section 7.5.

[7]All pairs of source and destination AP nodes for all demands.

```
cost_min ← COSTMIN
forall ON/OFF combination of mesh_nodes do
    // on mesh nodes which have been switched ON
    forall ON/OFF combination of links & num_of_mesh_links < max_links
    do
        forall demands do
            if demand < remaining_link_capacity() then
            cost ← cost_of_shortest_path() if cost < cost_min then
            cost_min ← cost adjust_link_capacity()
        end
    end
end
```

Algorithm 3: Psuedo-code for mesh routing.

7.4 Extending the WIND framework

We present the details of the framework that extends the `WIND` tool for WMN design. The framework is similar to that discussed for $\texttt{WIND}_{\texttt{wlan}}$ and we designate this part of the tool as WIND for WMN ($\texttt{WIND}_{\texttt{wmn}}$). The core algorithm of $\texttt{WIND}_{\texttt{wmn}}$ is now an MILP formulation. While the rest of the modules in the tool remains the same, we add an optimization pre-processor to $\texttt{WIND}_{\texttt{wmn}}$ which processes the inputs into the required format of the optimiser. An external optimiser is then invoked to solve the MILP problem. The details of the tool, shown in Figure 7.5, is given below.

$\texttt{WIND}_{\texttt{wmn}}$ takes the following inputs:

1. Network elements: Number of AP nodes and potential mesh nodes to be deployed in the layout.

2. Network element properties: Properties of nodes and their associated links (properties of mesh and AP nodes are as discussed in Section 7.3.1).

3. Network scenario strategy: Properties of deployment layout and distribution of nodes in the layout.

4. Traffic demands: User generated traffic demands for each AP.

5. Link cost functions: The cost functions for fixed and variable transmit powers. These are used by $\texttt{WIND}_{\texttt{wmn}}$ to construct the potential links (mesh links and mesh-AP links).

6. Optimizer parameters and heuristics: Heuristics and initial settings for the optimiser.

The first four input parameters are common between $\texttt{WIND}_{\texttt{wlan}}$ and $\texttt{WIND}_{\texttt{wmn}}$. The link cost functions are as discussed in section 7.3.3 and are used to attach a cost with each link. The optimizer parameters specify additional inputs to the optimizer. Two such parameters are used. One, an upper bound on the objective function, and two, the cuts performed by the optimizer to reduce execution time. We discuss these when describing the implementation details in Section 7.5.

The modules in $\texttt{WIND}_{\texttt{wmn}}$ are:

- Network scenario generator: A network scenario is created based on the deployment layout parameters and number of nodes. It creates locations of AP nodes and potential mesh nodes.

- Link constructor: Uses heuristics to generate the list of potential links using node and link properties. It calculates the feasibility of each mesh-mesh and mesh-AP link based on the link cost heuristics mentioned in Section 7.3.3. It also generates the cost of establishment of each link. Output is a list of potential links.

- Optimization preprocessor: Uses the network scenario and the list of potential links to construct the optimizer input. It also generates the list of demands, present in the network as a demand matrix, to be used by the constraint verifier. Outputs are the input for the optimiser and an demand matrix for the constraint verifier.

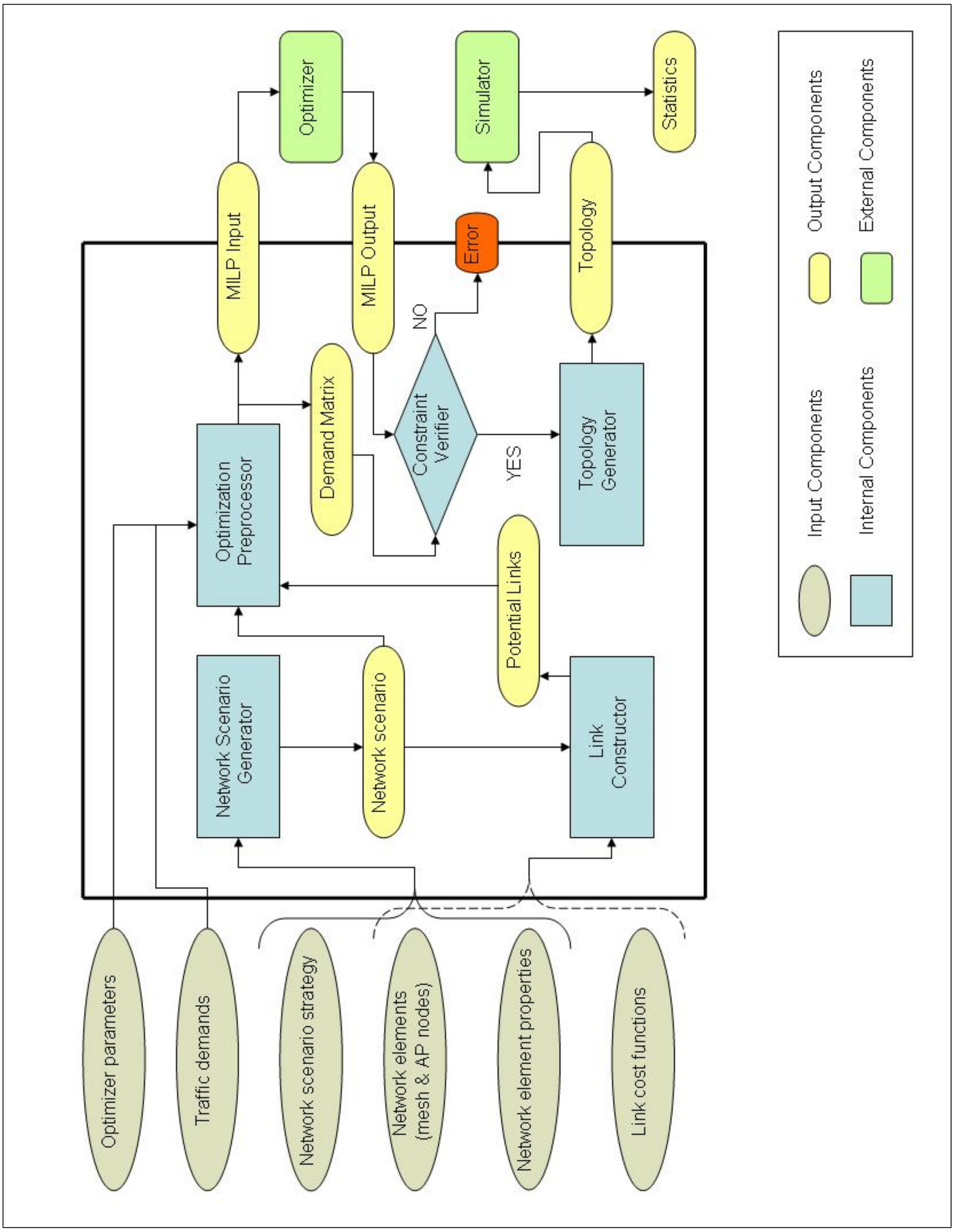

Figure 7.5: `WIND`$_{\texttt{wmn}}$: Tool overview.

- Optimizer: An external optimizer which is invoked to solve the MILP problem. The outputs are the total cost of the optimal topology, the topology graph and the route of each demand (as MILP output).
- Constraint verifier: Verifies the capacity constraints imposed on the scenario by comparing the optimizer output with the demand matrix. Invokes the topology generator once constraints are satisfied.
- Topology generator: The corresponding capacity-constrained topology is generated from the optimizer output files.
- Simulator: An external simulator is invoked to validate the topology generated. It does this by checking whether all demands are satisfied by the constructed topology.

7.5 Implementation and results

$\mathtt{WIND}_{\mathtt{wmn}}$ is implemented using perl and ILOG's Optimization Programming Language (OPL) [Sol]. The network scenario generator, link constructor and optimisation preprocessor were implemented in perl. The function call flow of the optimisation preprocessor is shown in Figure 7.6. The preprocessor (called Optimiser input generator) takes node and link information and cost functions to prepare the optimiser inputs in ILOG OPL formats. It in turn uses four programs for node distribution, demand generation, optimiser heuristic generation and solver input generation.

The preprocessor additionally generates the cutoff heuristic for the objective function based on the upper bound of the cost. This heuristic allows us to prune the search tree and results in significant improvement in execution time.[8] Additionally,

[8]**Heuristic for upper cut off on *F***: An upper bound can be set for the objective function

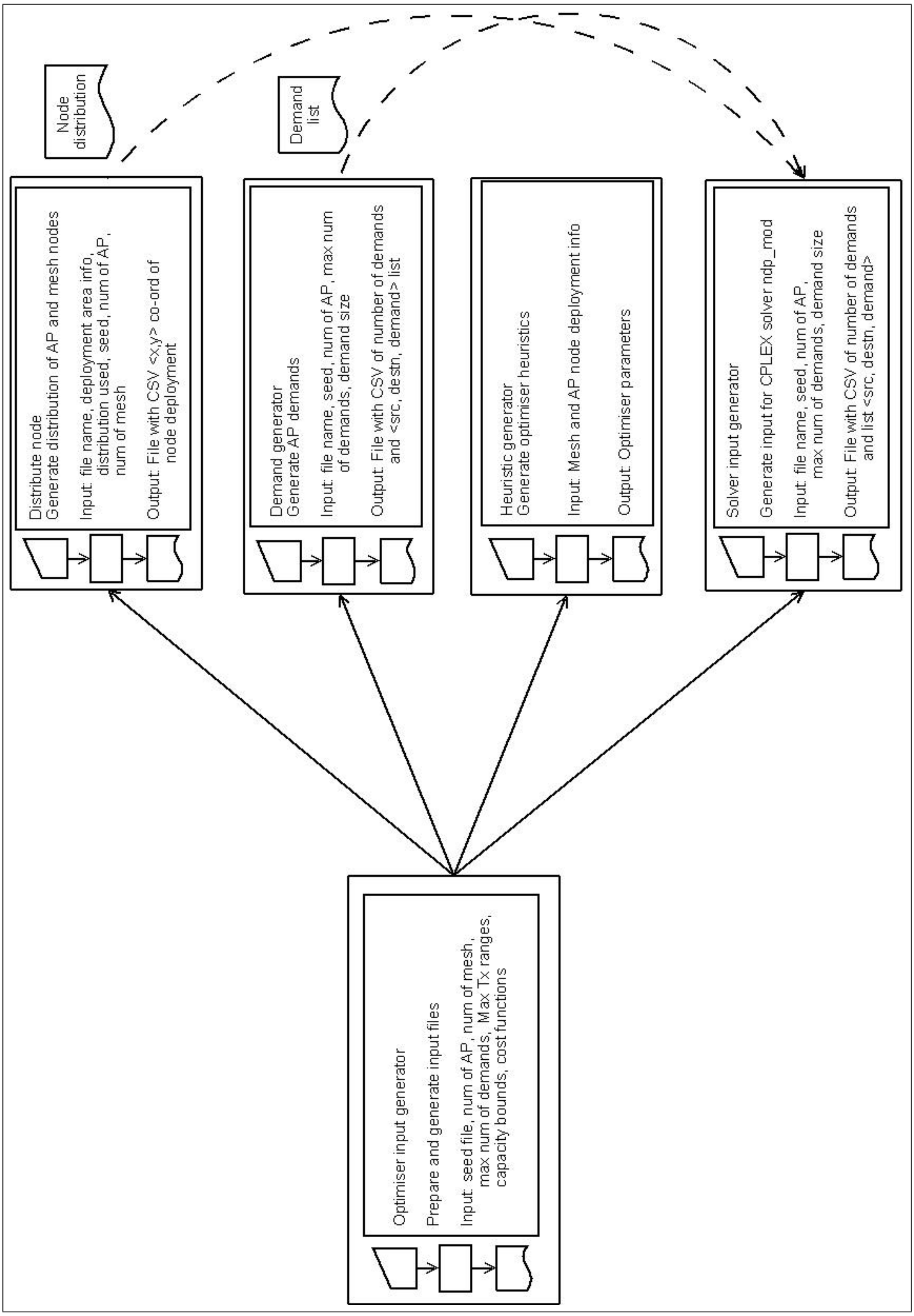

Figure 7.6: $\texttt{WIND}_{\texttt{wmn}}$ - Optimiser input generator.

the preprocessor also checks whether each AP is in transmission range of at least one mesh node (in order to output an error signal).

The design problem formulation is written in OPL for the CPLEX solver.

7.5.1 Experimental setup

The experimental setup is as follows. The deployment area is assumed to be a 100mX100m square. The AP nodes are deployed over the given area using a uniform random distribution and there is an upper bound on the number of demands per AP (This is taken as an input parameter and default value is 2). The potential mesh nodes are also deployed using a uniform random distribution. The parameters used in the experiment are given in Table 7.1. In order to simulate a varied deployment scenarios, we use two node distributions. For each network scenario, we varied the demands (11 artificially generated loads) to analyse topology changes and effect on the cost of the objective function (that is, number of nodes and links selected). A desirable effect would be that the changes in the topology are minimal in nature.

7.5.2 Results

The solver took a significantly longer time to solve problems with the fixed transmit power cost function. For example, we found that for a 10 AP and 8 mesh node scenario the solver took 6528.5 seconds for a solution. We found that the fixed power cost function is ineffective as the number of potential links found is exhaustive (due to the transmission ranges of mesh and AP nodes). The timing for a variable transmit power cost function for the similar scenario was 69.92 seconds. Hence,

F as follows. Assuming G as bound on number of links per mesh node, φ as cost of each link, r^2/ρ as the link cost factor: The upper bound is then the assumption that the mesh topology is a fully connected graph with all potential mesh nodes and links being `ON`. Removing the links with which the AP nodes are connected to mesh from the equation (in the first term on R.H.S.), we get the upper bound as $(|V| * G - |W|).r^2/\rho \ + \ |V| * \varphi$.

Parameter	Value
Area	100mx100m
AP/Mesh Tx Range	70m
Mesh node cost φ	1000
Mesh link cost factor ρ	10
Max. Links G	4
Link capacity	10 Mbps
Demand	1 Mbps

Table 7.1: Design parameters.

AP	Potential mesh	Exec time (s)	Mesh nodes (min,max)	Links (min,max,avg)
8	5	< 1	2, 3	8, 10, 10
10	7	50.93	3, 4	10, 13, 12
10	8	69.86	3, 4	10, 13, 12
12	7	178.12	3, 6	12, 16, 15
12	8	854.51	3, 5	12, 16, 15

Table 7.2: Topology construction and node location: results for network scenarios using variable power cost function. Execution time is average of 22 scenarios. Average number of links: avg = ceil(average of all scenarios).

Table 7.2 shows the results only with the variable transmit power cost function for various node deployment scenarios. Appendix B.1 contains a complete example of an 8 AP, 5 mesh node scenario describing node positions, demand graph, optimiser inputs and outputs.

The use of a variable power cost function to establish a link proves to be a better estimate (Figure 7.3(b)). This reduces the transmission power required by forcing nodes to choose nearer nodes. The topology computed remains on an average invariant with change in demand. This can be seen from the average number of links in Table 7.2.

We noticed that the CPLEX solver's parameters needed to be tuned in order to reduce the solution time. We found that, providing upper cutoff values for the objective speeds up the solver.

7.6 Comments

We first presented an overview of mesh networks and the classification of the design problem in the context of classical network design problems for wired networks. Next, we presented the design problem formulation as a mixed-integer linear programming problem for locationing and dimensioning. We then presented the assumptions and algorithmic details. We have extended the WIND tool to encompass the design of mesh networks and we detailed out the $\texttt{WIND}_{\texttt{wmn}}$ module and its interaction with the WLAN module from Chapter 6.

We make some observations regarding the WMN design experiment.[9] We used the CPLEX solver to generate various topologies under varying loads with a variable transmit power cost function. As expected, the problem scales exponentially with a small increase in node numbers due to the increase in search space required by the formulation. Also, the fixed power cost function proves ineffective as a cost function.

[9] We presented results on the WIND framework and the WMN design problem in WWC [RI06b] and Infocom [RI06c].

Indices:

$w = 1, 2, \ldots, W$: APs
$v = 1, 2, \ldots, V$: mesh nodes
$e = 1, 2, \ldots, E$: links
$f = 1, 2, \ldots, F$: directed access arcs (between AP & mesh nodes)
$t = 1, 2, \ldots, T$: directed transit arcs (between mesh nodes)

Constants:

$h_{ww'}$: volume of demand from AP w to w'
$H_w = \sum_{w'} h_{ww'}$: total demand outgoing from AP w
$\beta_{ev} = 1$ if link e is incident with mesh node v; 0, otherwise
$\beta_{fw} = -1$ if access arc f is incoming to AP w
$= 1$ if access arc f is outgoing from AP w
$= 0$ otherwise
$\beta_{fv} = -1$ if access arc f is incoming to mesh node v
$= 1$ if access arc f is outgoing from mesh node v
$= 0$ otherwise
$\beta_{tv} = -1$ if transit arc t is incoming to mesh node v
$= 1$ if transit arc t is outgoing from mesh node v
$= 0$ otherwise
$w_{ef} = 1$ if access arc f is realised on link e; 0, otherwise
$w_{et} = 1$ if transit arc t is realised on link e; 0, otherwise
κ_e : cost of installing link e
M_e : upper bound on the capacity of link e
φ_v : cost of installing mesh node v
G_v : upper bound on the number of radios of mesh node v

Variables:

x_{fw}: flow realising all demands originating at AP w on access arc f
x_{tw}: flow realising all demands originating at AP w on transit arc t
y_e: capacity of link e
$u_e = 1$ if link e is provided; 0, otherwise
$s_v = 1$ if mesh node v is installed; 0, otherwise

Objective function:

minimize $\boldsymbol{F} = \sum_e \kappa_e u_e + \sum_v \varphi_v s_v$

Constraints:

$\sum_t w_{et} \sum_w x_{fw} + \sum_f w_{ef} \sum_w x_{fw} \leq y_e$, $e = 1, 2, \ldots, E$ **- *(1)***

$\sum_f \beta_{fw} x_{fw} = H_w$, $w = 1, 2, \ldots, W$ **- *(2)***

$\sum_f \beta_{fw'} x_{fw} = -h_{ww'}$ **- *(3)***

$\sum_t \beta_{tv} x_{tw} + \sum_f \beta_{fv} x_{fw} = 0$ **- *(4)***

$y_e \leq M_e u_e$ **- *(5)***

$\sum_e \beta_{ev} u_e \leq G_v s_v$ **- *(6)***

Algorithm 4: Mesh network design problem formulation

Chapter 8

Conclusion and future work

In this thesis, we have motivated the need for capacity-constrained design of wireless networks. The class of wireless networks under consideration was a wireless mesh backhaul network with WLANs as clients.

We first identified two important issues in design of such wireless data networks: 1) Network capacity provisioning and 2) Wireless networks for backhaul. Towards this end, we approached this issue as an integrated multi-tiered network design problem. We identified three important stages in the design of such networks and presented techniques to construct capacity-constrained topologies. The following issues were addressed:

1. Provisioning of 802.11 WLANs in homogeneous and heterogeneous application scenarios.
2. Consideration of capacity-constrained approach towards design of wireless networks.
3. Integrated design of local area and backhaul wireless networks.

We first presented theoretical and simulation results for homogeneous applications and discussed the difficulty of provisioning in IEEE 802.11 DCF infrastructure mode WLANs. We then presented three techniques to alleviate this issue of AP-assignment, extended DCF, `SOAP1` and `SOAP2`. Then, for heterogeneous application deployments, we analysed the issue of prioritising applications for an ACL mechanism. Using sub-optimal application deployment as a technique, we showed that system utilisation improvement up to 75% was achievable (`SOAP1`).

We then presented an approach to construct WLAN topologies using the AP-assignment solutions. We developed a tool, $\texttt{WIND}_{\texttt{wlan}}$, that uses a bottom-up heuristic approach to construct the topology using a small set of inputs. The AP-assignment solutions from Chapter 5 were used as a basic construct mechanism in the form of link specification functions. We introduced composite unit as an abstract building block for network topology construction. An important aspect of the tool was the XML-based inputs and outputs. The schemas for these data are based on the OPNET Modeler data formats (given in Appendix A). This allowed us to connect the tool to the simulator, with topology as intermediate data, and validate topologies.

Then, we discussed the issue of backhaul wireless mesh networks and presented a design problem formulation. The formulation was a mixed-integer linear programming problem for locationing and dimensioning of a mesh backhaul. We extended the WIND tool to encompass the design of mesh networks and we detailed out the $\texttt{WIND}_{\texttt{wmn}}$ module and its interaction with the WLAN module.

The `WIND` tool can be individually used for design of WLAN or WMN networks, with capacity provisioning and minimising network infrastructure cost as main constraints. Taken together, this tool can also be used to design a consolidated view of a multi-tier heterogeneous wireless network with a backhaul WMN deployment and underlying WLANs.

8.1 Future work

In this section, we present some ideas on extending the work done in this thesis.

Bringing together capacity and coverage

Bringing together both capacity and coverage constraints can result in more accurate topologies. Including RF planning as an extension to the AP-assignment problem will give us a more accurate prediction of AP-client assignments.

Extensions to AP-assignment problem

The work done for AP-assignment solutions can also be extended to study cases of interaction of various application scenarios or devices operating together. These results can be presented as an information base which can be used as a quick reference during WLAN deployments.

Extensions to WMN design

Further performance analysis case studies for wireless mesh network deployment scenarios can be undertaken.

Further work on link cost functions and computing topologies for large network scenarios is envisaged. We also plan to automate the design process by integrating it with the wireless infrastructure design tool proposed in [RI05].

The scheduling and routing issues that arise in wireless mesh networks can be brought into the dimensioning problem. We assume non-interfering point-to-point directional links between infrastructure nodes in the WMN. Scheduling issues which arise from relaxing this assumption can also be studied. A short overview of various scheduling approaches is given in Appendix B.2.

Currently, we generate topologies without specifying the number of hops as a constraint. One approach would be to set a k-hop to back-bone node bound for each traffic (k may be different for different clients).

Extending optimization goals

The cost v/s capacity trade off would be another interesting study. For any given network scenario, the tool tries to construct the optimal topology. On the solution being infeasible, the tool generates an error report. Extensions to the tool such that it can suggest alternate topologies at different costs may be analysed.

Use of tool in other areas

The current version of the tool guarantees capacity constraints for WMNs and WLANs. Extensions to this in other areas can be considered by accommodating constraints such as: delay, lifetime (in the case of sensor networks), reachability (in the case of sparse networks) and limited mobility.

Appendices

Appendix A

WIND XML input files

A.1 Information base

XML XSD representation of information base schema shown in figure 6.2.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="information_base">
    <xs:complexType>
      <xs:all>
        <xs:element minOccurs="1" maxOccurs="1" name="link_types">
          <xs:complexType>
            <xs:all>
              <xs:element name="link_type">
                <xs:complexType>
                  <xs:attribute name="id" type="xs:integer" />
                  <xs:attribute name="name" type="xs:string" />
                  <xs:attribute name="capacity" type="xs:integer" />
```

```
          <xs:attribute name="max_nodes" type="xs:string" />
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element minOccurs="1" maxOccurs="1" name="as_types">
  <xs:complexType>
    <xs:all>
      <xs:element name="as_type">
        <xs:complexType>
          <xs:attribute name="name" type="xs:string" />
          <xs:attribute name="outgoing_load" type="xs:integer" />
          <xs:attribute name="incoming_load" type="xs:integer" />
        </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element minOccurs="1" maxOccurs="1" name="node_types">
  <xs:complexType>
    <xs:all>
      <xs:element name="node">
        <xs:complexType>
          <xs:all>
            <xs:element minOccurs="1" maxOccurs="1" name="link_list">
              <xs:complexType>
```

```
            <xs:all>
              <xs:element minOccurs="1" name="link">
                <xs:complexType>
                  <xs:attribute name="id" type="xs:integer" />
                  <xs:attribute name="name" type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="1" maxOccurs="1" name="as_list">
          <xs:complexType>
            <xs:all>
              <xs:element minOccurs="0" name="as">
              <xs:complexType>
              <xs:attribute name="name" type="xs:string" />
              <xs:attribute name="source_addr" type="xs:string" />
              <xs:attribute name="destn_addr" type="xs:string" />
              <xs:attribute name="link" type="xs:integer" />
              </xs:complexType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
      <xs:attribute name="name" type="xs:string" />
      <xs:attribute name="mobility" type="xs:string" />
```

```
                  <xs:attribute name="relay" type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.1.1 Example

Information base for example and simulation in Section 6.5.

```
<information_base>


<link_types>
<link_type id="1" name="802.11b DCF"  capacity="11" max_nodes="">
</link_type>
<link_type id="2" name="10mbps ethernet" capacity="10" max_nodes="">
</link_type>
</link_types>


<as_types>
<as_type name="voip" outgoing_load="10" incoming_load="1000">
</as_type>
<as_type name="ftp heavy" outgoing_load="100" incoming_load="10000">
</as_type>
<as_type name="server" outgoing_load="120000" incoming_load="1200">
</as_type>
</as_types>


<node_types>


<node name="Switch" mobility="no" relay="yes">
<link_list>
<link id="1" name="10mbps ethernet"></link>
<link id="2" name="10mbps ethernet"></link>
```

```
</link_list>
<as_list></as_list>
</node>

<node name="Access Point" mobility="no" relay="yes">
<link_list>
<link id="1" name="802.11b DCF"></link>
<link id="2" name="10mbps ethernet"></link>
</link_list>
<as_list></as_list>
</node>

<node name="PDA" mobility="yes" relay="no">
<link_list>
<link id="1" name="802.11b DCF"></link>
</link_list>
<as_list>
<as  name="voip" src_addr="PDA" destn_addr="server" link="1">
</as>
</as_list>
</node>

<node name="Workstation" mobility="no" relay="no">
<link_list>
<link id="1" name="802.11b DCF"></link>
</link_list>
<as_list>
```

```
<as name="ftp heavy" src_addr="Workstation" destn_addr="server" link="1">
</as>
</as_list>
</node>

<node name="server" mobility="no" relay="no">
<link_list>
<link id="1" name="10mbps ethernet"></link>
</link_list>
<as_list>
<as name="server" src_addr="server" destn_addr="" link="1">
</as>
</as_list>
</node>

</node_types>

</information_base>
```

A.2 Input parameters

XML XSD representation of information base schema shown in figure 6.4.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:element name="input_parameters">
    <xs:complexType>
      <xs:all>
        <xs:element minOccurs="1" maxOccurs="1" name="deployment_layout_graph">
          <xs:complexType>
            <xs:all>
              <xs:element minOccurs="1" name="graph_node">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="edge">
                      <xs:complexType>
                        <xs:attribute name="graph_node" type="xs:integer" />
                      </xs:complexType>
                    </xs:element>
                  </xs:all>
                  <xs:attribute name="id" type="xs:integer" />
                </xs:complexType>
              </xs:element>
              <xs:element name="affinity_factors">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="af">
```

```
                  <xs:complexType>
                    <xs:attribute name="name" type="xs:string" />
                    <xs:attribute name="value" type="xs:integer" />
                  </xs:complexType>
                </xs:element>
              </xs:all>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
    <xs:element minOccurs="1" maxOccurs="1" name="nodes_deployed">
      <xs:complexType>
        <xs:all>
          <xs:element name="node">
            <xs:complexType>
              <xs:attribute name="name" type="xs:string" />
              <xs:attribute name="number" type="xs:integer" />
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
  </xs:all>
 </xs:complexType>
</xs:element>
</xs:schema>
```

A.2.1 Example

Input parameters for example and simulation in Section 6.5.

```
<input_parameters>


<nodes_deployed>
<node name="PDA" num="10"></node>
<node name="Workstation" num="10"></node>
<node name="Ftp Server" num="1"></node>
</nodes_deployed>

<deployment_layout_graph>

<graph_node id="1">
<edge graph_node="2"></edge>

<affinity_factors>
<af name="PDA" value="0.6"></af>
<af name="Workstation" value="0.6"></af>
<af name="Ftp Server" value="1"></af>
</affinity_factors>

</graph_node>

<graph_node id="2">
<edge graph_node="1"></edge>
<edge graph_node="2"></edge>
```

```
<affinity_factors>
<af name="PDA" value="0.6"></af>
</affinity_factors>

</graph_node>

<graph_node id="3">
<edge graph_node="2"></edge>

<affinity_factors>
<af name="PDA" value="0.1"></af>
<af name="Workstation" value="0.4"></af>
</affinity_factors>

</graph_node>

</deployment_layout_graph>

</input_parameters>
```

Appendix B

Example mesh scenario and scheduling issues in WMNs

B.1 Example: 8 AP - 5 mesh scenario

This section describes example input parameters and results for the 8 AP, 5 mesh node problem. The output of Node distributor and demand generator is shown in Section B.1.1. The output of optimiser input generator is shown in Section B.1.2. The output of the CPLEX solver is shown in Section B.1.3.

B.1.1 Node position and demand graph

Node are randomly distributed on a 100mX100m square deployment area. Each node position is recorded as an <x, y> tuple. The number of demands is chosen randomly based on user specified maximum number of demands. The demands between two APs are randomly generated. Each demand is recorded as an <srcAP, destnAP, demand> tuple.

`Node positions:`

```
AP, 8, 70,
4.16, 45.44,
83.48, 33.59,
56.54, 0.17,
18.75, 99.04,
75.04, 36.62,
35.12, 57.33,
13.25, 6.41,
95.08, 15.35,
Mesh, 5, 70,
58.46, 21.65,
80.65, 14.04,
62.2, 21.08,
0.65, 57.32,
93.26, 34.03,

Demand list:

Demands, 11,
 5, 2, 1000,
 8, 4, 1000,
 2, 5, 1000,
 5, 4, 1000,
 7, 2, 1000,
 1, 5, 1000,
 6, 3, 1000,
```

```
 2, 5, 1000,
 4, 5, 1000,
 8, 5, 1000,
 3, 4, 1000,
End
```

B.1.2 Optimiser input

The input to the CPLEX solver takes input in the following example format. The first 5 lines specifies the number of AP, potential mesh nodes, potential links, access arcs and transit arcs respectively. The array `M` specifies the capacity of each link. The array `kappa` specify the link costs. The arrays `accessSrc`, `accessDestn` and `accessLink` define the access arcs. The arrays `meshSrc`, `meshDestn` and `meshLink` define the transit arcs. The array `h` specifies the demands, `varphi` the cost of mesh nodes and `G` the upper bound on the number of links per mesh node.

```
OPL input:


W =  8;
V =  5;
E = 36;
Z = 58;
T = 14;
linkSrc = [ 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 5, 5, 5, 5, 6, 6, 6, 6, 6, 7, 7,
 7, 8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 11];
linkDestn = [ 9, 11, 12, 9, 10, 11, 13, 9, 10, 11, 13, 12, 9, 10, 11, 13, 9, 10,
11, 12, 13, 9, 10, 11, 12, 9, 10, 11, 13, 10, 11, 12, 13, 11, 13, 13];
M = [ 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 1000
```

```
 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000,
 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000, 10000,
 10000, 10000, 10000];
kappa = [ 352, 397, 16, 77, 40, 61, 10, 47, 78, 47, 250, 207, 50, 55, 41, 34, 182,
 395, 205, 119, 393, 228, 461, 262, 276, 139, 21, 112, 36, 56, 2, 462, 137, 39, 56,
 114];
accessSrc = [ 1, 9, 1, 11, 1, 12, 2, 9, 2, 10, 2, 11, 2, 13, 3, 9, 3, 10, 3, 11, 3,
 13, 4, 12, 5, 9, 5, 10, 5, 11, 5, 13, 6, 9, 6, 10, 6, 11, 6, 12, 6, 13, 7, 9, 7,
 10, 7, 11, 7, 12, 8, 9, 8, 10, 8, 11, 8, 13];
accessDestn = [ 9, 1, 11, 1, 12, 1, 9, 2, 10, 2, 11, 2, 13, 2, 9, 3, 10, 3, 11, 3,
 13, 3, 12, 4, 9, 5, 10, 5, 11, 5, 13, 5, 9, 6, 10, 6, 11, 6, 12, 6, 13, 6, 9, 7,
 10, 7, 11, 7, 12, 7, 9, 8, 10, 8, 11, 8, 13, 8];
accessLink = [ 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11,
 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 16, 17, 17, 18, 18, 19, 19, 20, 20, 21,
 21, 22, 22, 23, 23, 24, 24, 25, 25, 26, 26, 27, 27, 28, 28, 29, 29];
meshSrc = [ 9, 10, 9, 11, 9, 12, 9, 13, 10, 11, 10, 13, 11, 13];
meshDestn = [ 10, 9, 11, 9, 12, 9, 13, 9, 11, 10, 13, 10, 13, 11];
meshLink = [ 30, 30, 31, 31, 32, 32, 33, 33, 34, 34, 35, 35, 36, 36];
h = [
[0,0,0,0,1000,0,0,0],
[0,0,0,0,2000,0,0,0],
[0,0,0,1000,0,0,0,0],
[0,0,0,0,1000,0,0,0],
[0,1000,0,1000,0,0,0,0],
[0,0,1000,0,0,0,0,0],
[0,1000,0,0,0,0,0,0],
[0,0,0,1000,1000,0,0,0],
```

```
];
varphi = [1000, 1000, 1000, 1000, 1000];
G = [4, 4, 4, 4, 4];
```

B.1.3 Optimiser output

The array `Nodes` and `Links` shows the potential mesh nodes and links. The selected nodes and links of the constructed topology are set to `1`. The `Links` array is mapped to the arrays `linkSrc` and `linkDestn` in the optimiser input (in Section B.1.2) to construct the final set of links. The two-dimensional arrays `X_FW` and `T_FW` show the demand flow on each access and transit arcs (see Section 7.3.1 for definitions).

```
Results:
Nodes:  [1 1 0 1 0]
Link:  [0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 1
0 0 0 0 1 0 0 1 0 1 0 0 0 0]

X_FW:  [[0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [1000 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 2000 0 0 0 0 0 0]
        [0 0 0 0 1000 0 1000 0]
        [0 0 0 0 0 0 0 0]
```

```
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 1000 0 0 0 0 0]
[0 0 0 0 0 1000 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 1000 0 0 0 0]
[0 0 1000 0 1000 0 0 1000]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 2000 0 0 0]
[1000 2000 0 1000 0 0 0 1000]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0]
```

```
        [0 0 0 0 0 1000 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 1000 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 4000 0 0 2000]
        [0 0 0 0 4000 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]]

X_TW:  [[1000 0 0 1000 0 0 1000 0]
        [0 0 0 0 1000 0 0 1000]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 1000 0 1000 0 0 1000]
        [1000 0 0 1000 0 1000 0 0]
```

```
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]
        [0 0 0 0 0 0 0 0]]
main returns 0
<<< main, at 0.85s, took 0.85s
```

B.2 Scheduling issues in wireless mesh design

We discuss the basic issues in scheduling WMNs. The assumption in Chapter 7 was that of a simple scheduler that allows each link to operate simultaneously. This assumption was made to simplify the design problem. Our focus was towards addressing the problem of node location and topology construction and not finding a suitable scheduling algorithm. As pointed out, the current problem definition itself is an np-hard problem. Even assuming that we are given node locations and topology (that is, the network), scheduling is also a np-hard problem [KMK05]. In this section, we elaborate on the different classes of Time Division Multi-Access (TDMA) schedulers that can be built over and above our topology design.

First case: Each link operates independent of all other links. In this case there is no scheduling involved and all links operate simultaneously. The capacity constraints in our optimization algorithm are sufficient to capture the properties of the links.

Second case: If we assume a centralized scheduler, then we can construct a

TDMA scheduling algorithm. Let the total of time slots available be T. Now the system (the base station) has to allocate time slots to each node in the system (mesh + AP). Assuming that we know the total flow on each link, we can impose the following `TDMA constraint` on the resulting topology:

```
TDMA Constraint:
for each node
  for each link
    //time taken by flows on link l in slots
    t(l) = sum(time taken for all flows on link l)
  //time taken by flows on all links at node n in slots
  t(n) = sum(t(l))

//total time taken by all flows on all links at every node in slots
T_dash = sum(t(n))

if(T_dash <= T)
  //We have a successful schedule
```

Third case: We schedule transmissions by considering interference between nodes and consider Spacial TDMA (STDMA) to take advantage of non-interfering simultaneous transmissions. We first define the terms.

A transmission scenario (TS) is defined as a snapshot of the system in which non-interfering nodes are either in transmit or receive mode. Interference can be defined in many ways [VM06]. We define a non-interfering TS as follows:

- All links are point to point directional links. That is, all transmitter-receiver pairs are unique.
- No transmitting node can receive simultaneously.

Thus, if any two links (i,j) and (k,l) are active simultaneously, then { i,j,k,l } must be distinct. Since each link is considered to be a directional link, we discount *inter-node interference*. Also, each node can use only one link at a time either to transmit or receive so we accommodate *inter-link interference* at a node.

A Transmission Scenario Set (TSS) is defined as the set of all possible TSs.

The problem is now an optimization sub-problem for the design problem involving the computation of a TSS (set of non-interfering snapshots of the system). So, we can impose the following constraint on the TSS: each link switched `ON`, on each node switched `ON`, should appear in atleast one TS in the TSS.

The order of transmission (which link is transmitting/receiving) in the TSS does not concern us. What concerns us is:

- Each link switched `ON` on each node switched `ON` should appear in at least one TS in the TSS in order to have a source to destination path.
- Throughput should be maximised.

Computing TSS: After the optimizer chooses a set of nodes and links. We compute the TSS. Finding all possible TSs is a hard problem. Instead we use a greedy algorithm.

```
GOAL: Generate TSS
Approach: Maximise simultaneous links.
```

We now have to check whether slot time to be allocated among all nodes is less that `T`, `sum(slot time) < T`. We do that as follows: First note that each link in a TS can transmit simultaneously.

```
STDMA Constraint:
for each TS in TSS
```

```
 for each link in TS
   // time taken by one TS = maximum time taken by any link,
   since all nodes are transmitting simultaneously
   t(TS) = max(sum(time taken for all flows on link l))
t(TSS) = sum(all t(TS))

if(t(TSS) < T)
  //We have a successful schedule
```

So maximum throughput is achieved when we allocate the minimum number of slots.

Publications

Following are the papers published during the course of this work.

- Automatic topology generation for a class of wireless networks. *IEEE International Conference On Personal Wireless Communications*, 2005.

  Joint work with: Sridhar Iyer.

- Automated design of VoIP-enabled 802.11g WLANs. *OPNETWORK*, 2005.

  Joint work with: Sridhar Iyer and Atanu Guchhait.

- Designing multi-tier wireless mesh networks: Capacity-constrained placement of mesh backbone nodes. *World Wireless Congress*, 2006.

  Joint work with: Sridhar Iyer.

- Capacity-constrained design of resilient multi-tier wireless mesh networks. *IEEE Infocom Student Workshop*, 2006.

  Joint work with: Sridhar Iyer.

- WIND: A Tool for capacity-constrained design of resilient multi-tier wireless mesh networks. *IEEE Infocom Poster Session*, 2006.

  Joint work with: Sridhar Iyer.

- Bridging the gap between reality and simulations: An Ethernet case study. *IEEE International Conference on Information Technology*, 2006.

  Joint work with: Punit Rathod and Srinath Perur.

- VoIP-based intra-village teleconnectivity: An architecture and case study. *First annual workshop on Wireless Systems: Advanced Research and Development (WISARD)*, 2006.

  Joint work with: Janak Chandarana, K. Sravana Kumar, Srinath Perur, Sameer Sahasrabuddhe and Sridhar Iyer.

Bibliography

[80203a] IEEE wireless LAN medium access control (MAC) and physical layer (PHY) specifications. 2003.

[80203b] IEEE wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Further higher date rate extension in the 2.4 ghz band. 2003.

[80204] IEEE standard for local and metropolitan area networks - part 16: Air interface for fixed broadband wireless access systems. 2004.

[80205] IEEE wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Medium access control (MAC) quality of service (QoS) enhancements. 2005.

[80206] IEEE standard for local and metropolitan area networks part 16: Air interface for fixed and mobile broadband wireless access systems - Amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands. 2006.

[ACG04] G. Anastasi, M. Conti, and E. Gregori. IEEE 802.11 ad hoc networks: Protocols, performance and open issues. In *Ad Hoc Networking*, chapter 3. Wiley-Interscience, 2004.

[Air] AiroPeek: Wireless LAN analyser. *www.wildpackets.com*.

[All99] Wi-Fi Alliance. Five steps to creating a wireless network. Technical report, 1999.

[Ath] Atheros communications.

[Ato] Atoll - network design platform. *www.forsk.com.*

[AWW05] Ian Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks*, 2005.

[BCE05] R. Bruno, M. Conti, and E. Gregori E. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3), 2005.

[Bia00] Guiseppe Bianchi. Performance analysis of the IEEE distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[BVB03] Anand Balachandran, Geoffrey M. Voelker, and Paramvir Bahl. Wireless hotspots: current challenges and future directions. In *ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH)*, pages 1–9, 2003.

[CCL03] I. Chlamtac, M. Conti, and Jennifer Liu. Mobile ad hoc networking: imperatives and challenges. *Ad Hoc Networks 1*, 2003.

[CeW06] CeWIT. Broadband wireless for rural areas - WiFiRe: Medium access control (MAC) and physical layer (PHY) specifications. 2006.

[CIN] Computer tool for planning and design of wireless systems in enclosed spaces (CINDOOR). *www.gsr.unican.es/cindoor.*

[Cis] Cisco systems. *www.cisco.com.*

[EDXa] EDX signalMX mesh network planning tool. *www.edx.com/products/products.html*.

[EDXb] EDX signalpro network planning tool. *www.edx.com/products/products.html*.

[Eth] Ethereal network protocol analyser. *www.ethereal.com*.

[Eth05] IEEE 802.3 Ethernet standard. 2005.

[ETS05] ETSI. High performance metropolitan area networks (HiperMAN) standard. *portal.etsi.org/radio/HiperMAN/HiperMAN.asp*, 2005.

[FGK+95] Steven J. Fortune, David M. Gay, Brian W. Kernighan, Orlando Landron, Reinaldo A. Valenzuela, and Margaret H. Wright. WISE design of indoor wireless systems: Practical computation and optimization. *IEEE Computational Science & Engineering*, 2(1):58–68, 1995.

[FHR04] M. Feeley, N. Hutchinson, and S. Ray. Realistic mobility for ad hoc network simulation. *ADHOC-NOW, LNCS*, 3158, 2004.

[Gei99] Jim Geier. *Wireless LANs: Implementing Interoperable Networks*. MacMillan Technical Publishing, 1999.

[GMW00] Philip Gill, Walter Murray, and Margaret Wright. *Practical Optimization*. Academic Press, 2000.

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[Gro04] Farpoint Group. Wireless LAN infrastructure mesh networks: Capabilities and benefits. 2004.

[Gro07] Bluetooth Special Interest Group. Specification of the bluetooth system. 2007.

[Gro08] Mesh Networking Task Group. IEEE 802.11s unapproved standard for extended service set (ESS) mesh networking. 2008.

[Gro09] IEEE 802.11 Task Group. IEEE Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Enhancements for higher throughput. 2009.

[Hil99] A. Hills. Wireless Andrew. *IEEE Spectrum*, 36(6):49–53, 1999.

[Hil01] A. Hills. Large-scale wireless LAN design. *IEEE Communications Magazine*, 39(11):98–107, 2001.

[HL86] T. Hou and Victor O.K. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(1), 1986.

[Hu93] L. Hu. Topology control for multihop packet radio networks. *IEEE Transactions on Communications*, 41(10), 1993.

[Int] Intel. *www.intel.com*.

[ISO05] ISO/IEC. Mpeg-4 avc: Information technology - coding of audio-visual objects - part 10: Advanced video coding. 2005.

[iST96] European Cooperation in Science and Technology. Digital mobile radio towards future generation systems, COST 231. 1996.

[IT88] ITU-T. G.711: Pulse code modulation (pcm) of voice frequencies. 1988.

[IT03] ITU-T. International telephone connections and circuits - general recommendations on the transmission quality for an entire international telephone connection: One-way transmission time. 2003.

[IT06] ITU-T. G.723.1 : Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s. 2006.

[IT07a] ITU-T. G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (cs-acelp). 2007.

[IT07b] ITU-T. H.264 : Advanced video coding for generic audiovisual services. 2007.

[JCNK95] B. Jabbari, G. Colombo, A. Nakajima, and J. Kulkarni. Network issues for wireless communications. *IEEE Communications Magazine*, 33(1), 1995.

[JLD04] X. Jia, D. Li, and D. Du. Topology control in ad hoc wireless network. In *IEEE Infocom*, 2004.

[JS03] Jangeun Jun and M.L. Sichitiu. The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, 10(5):8–14, 2003.

[Kap02] Steve Kapp. 802.11: leaving the wire behind. *IEEE Internet Computing*, 6(1):82–85, 2002.

[Kis] Kismet: Layer2 wireless network detector, sniffer, and intrusion detection system. *www.kismetwireless.com*.

[KMK05] Anurag Kumar, D. Manjunath, and Joy Kuri. *Communication Networking: An Analytical Approach*. Morgan Kaufmann, 2005.

[KSK04] R. Karrer, A. Sabharwal, and E. Knightly. Enabling large-scale wireless broadband: the case for TAPs. *ACM SIGCOMM Computer Communication Review*, 34(1):27–32, 2004.

[KU02] M. Kamenetsky and M. Unbehaun. Coverage planning for outdoor wireless LAN systems. In *International Zurich Seminar on Broadband Communications: Access, Transmission, Networking*, 2002.

[LBB04] S. Lee, S. Banerjee, and B. Bhattacharjee. The case for a multi-hop wireless local area network. In *IEEE Infocom*, 2004.

[Lin] Linksys technologies. *www.linksys.com.*

[Mad] Atheros chipsets with madwifi WMM support.

[McL03] Michelle McLean. How to design a wireless LAN. *Communication News*, September, 2003.

[MCN+02] M.A. Marsan, C.F. Chiasserini, A. Nucci, G. Carello, and L. De Giovanni. Optimizing the topology of bluetooth wirelesspersonal area networks. In *IEEE Infocom*, 2002.

[Mer] Meraki networks. *www.meraki.com.*

[Mes] Mesh networks. *www.motorola.com/mesh.*

[MGFK04] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama. Voice capacity of IEEE 802.11b, 802.11a and 802.11g wireless LANs. In *IEEE GLOBECOM*, 2004.

[Mic] Self-organizing neighborhood wireless mesh networks. *research.microsoft.com/mesh.*

[Mot] Motorola handheld computers. *www.motorola.com/Business/US-EN/Business+Product+and+Services/Mobile+Computers/Handheld+Computers.*

[Net] Netstumbler. *www.netstumbler.com.*

[ns2] Network simulator (ns2). *www.isi.edu/nsnam/ns.*

[OL96] I.H. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, pages 63:513–628, 1996.

[OPN] OPNET modeler. *www.opnet.com.*

[PK02] Kaveh Pahlavan and Prashant Krishnamurthy. *Principles of Wireless Network - A Unified Approach.* Pearson Education Asia, 2002.

[PM04] Michal Pioro and Deepankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann, 2004.

[PPK+00] A.R. Prasad, N.R. Prasad, A. Kamerman, H. Moelard, and A. Eikelenboom. Indoor wireless LANs deployment. In *Vehicular Technology Conference (VTC)*, 2000.

[Rad] History of radio. *www.wikipedia.org/wiki/History_of_radio.*

[Rap02] Theodore S. Rappaport. *Wireless Communications: Principles and Practice: second edition.* Prentice Hall, 2002.

[RI05] Raghuraman Rangarajan and Sridhar Iyer. Automatic topology generation for a class of wireless networks. In *IEEE International Conference On Personal Wireless Communications*, 2005.

[RI06a] Raghuraman Rangarajan and Sridhar Iyer. Capacity-constrained design of resilient multi-tier wireless mesh networks. *IEEE Infocom Student Workshop*, 2006.

[RI06b] Raghuraman Rangarajan and Sridhar Iyer. Designing multi-tier wireless mesh networks: Capacity-constrained placement of mesh backbone nodes. In *World Wireless Congress*, 2006.

[RI06c] Raghuraman Rangarajan and Sridhar Iyer. WIND: A tool for capacity-constrained design of resilient multi-tier wireless mesh networks. In *IEEE Infocom Poster Session*, 2006.

[RIG05] Raghuraman Rangarajan, Sridhar Iyer, and Atanu Guchhait. Automated design of VoIP-enabled 802.11g WLANs. In *OPNETWORK*, 2005.

[RML00] R.C. Rodrigues, G.R. Mateus, and A.A.F. Loureiro. On the design and capacity planning of a wireless local area network. In *Network Operations and Management Symposium*, 2000.

[Roo] MIT RoofNet project. *pdos.csail.mit.edu/roofnet*.

[RT99] E. Royer and C. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications*, 1999.

[SL06a] Aaron So and Ben Liang. A lagrangian approach for the optimal placement of wireless relay nodes in wireless local area networks. In *Proc. of the International IFIP-TC6 Networking Conference*, 2006.

[SL06b] Aaron So and Ben Liang. Optimal placement of relay infrastructure in heterogeneous wireless mesh networks by Bender's decomposition. In *IEEE QShine*, 2006.

[Sol] ILOG CPLEX Solver. www.cplex.com.

[Spe] SpectraGuard planner. *www.airtightnetworks.*

[Sta94] European Telecommunication Standard. GSM 06.10 (ETS-300-580-2): Digital cellular telecommunications system; Full rate speech transcoding. 1994.

[Str91] Bjarne Stroustrup. *The C++ Programming Language: Second edition.* Addison-Wesley, 1991.

[Sym] Symbol technologies. *www.symbol.com.*

[Sys04] Cisco Systems. Maximum throughput calculations for 802.11b WLAN - Implementation design guide. 2004.

[Tan02] Andrew S. Tanenbaum. *Computer Networks.* Prentice Hall, 2002.

[Var03] U. Varshney. The status and future of 802.11-based WLANs. *IEEE Computer*, 36(6):102–105, 2003.

[VM06] H. Viswanathan and S. Mukherjee. Throughput-range tradeoff of wireless mesh backhaul networks. *IEEE JSAC*, 24(3), 2006.

[WH03] Roger M. Whitaker and Steve Hurley. Evolution of planning for wireless communication systems. In *IEEE Hawaii International Conference on System Sciences (HICSS)*, 2003.

[WiM] WiMAX forum. *www.wimaxforum.org.*

[WLL05] Wei Wang, S. C. Liew, and Victor O. K. Li. Solutions to performance problems in VoIP over a 802.11 wireless LAN. *IEEE Transactions on Vehicular Technology*, 54(1), 2005.

[WMM04] WMM specification v1.1. *www.wi-fi.org*, 2004.

[Wra] Wireless research API (WRAPI). *sysnet.ucsd.edu/pawn/wrapi.*

[Xer] Xerces-C++ validating xml parser. *xerces.apache.org/xerces-c/.*

[YCG+03] Yang, Conner, Guo, Hazra, and Zhu. Common wireless adhoc network usage scenarios. Technical report, 2003.

[YPC05] Mark Yarvis, Konstantina Papagiannaki, and W. Steven Conner. Characterization of 802.11 wireless networks in the home. In *Wireless Network Measurement Workshop (WiNMee)*, 2005.

[ZPK03] Stanislav Zvanovec, Pavel Pechac, and Martin Klepal. Wireless LAN networks design: Site survey or propagation modeling? *Radio Engineering*, 12(4), 2003.