Experimental Results in Wireless LANs and Meshes

Dragoș Ștefan Niculescu,
ETTI, University Politehnica of Bucharest

Habilitation Thesis

Contents

List of Figures

List of Tables

Part I

Thesis Summary

Executive Summary

This thesis summarizes results in the larger area of **mobile computing** and includes work on positioning of mobile devices, optimization of communication procedures, and optimization of antenna configurations. Since the inception of the first IEEE 802.11 standard in 1997, networks using unlicensed bands have been growing in popularity and surpassed all initial estimates of adoption. Their success enabled some new mobile computing applications, but also spawned a host of new problems for the research community. While expecting new applications and functionality from the new wireless LAN (WLAN), users also expected the network to behave as the conventional LAN, but this was not the case. The thesis reproduces results previously published in our work [1, 2, 3, 4, 5], and shows that 802.11 based networks (informally named WiFi) are difficult to operate under high density conditions (because of complex patterns of interference), multiple hop conditions (because of self interference of flows), and under some conditions of mixed traffic (TCP and VoIP). The results summarized here are applicable to 802.11 based networks with single and multiple hops. For the single hop case, we present a method of using the network not only for communication, but also for providing an indoor positioning service. Another research theme is that of enhancing WLAN links using mobile antenna technologies. For the multiple hop case, we focused on problems related to operation of fixed WiFi meshes that are used for popular applications based on TCP, or VoIP.

Angle of arrival (AOA) has previously been used for outdoor positioning in aircraft navigation and for services like E911. For indoor positioning, the best schemes to date rely either on extensive infrastructure, or on sampling of the signal strength on a dense grid, which is subject to changes in the environment, like furniture, elevators, or people. In chapter 1, we present an indoor positioning architecture that does not require a signal strength map, simply requiring the placement of special **VOR base stations** (VORBA). While our incipient realization of the AOA using 802.11 uses a base station with a revolving directional antenna, a non mechanical implementation would yield comparable performance, even with quantized angles. Performance of positioning with VOR base stations is evaluated though experimentation, simulation, and theoretical analysis.

Performance in multihop wireless networks is known to degrade with the number of hops for both TCP and UDP traffic. For VoIP, the wireless network presents additional challenges as the perceived quality is dependent on both loss and delay. In chapter 2, we investigate several methods to improve voice quality and present experimental results from an 802.11b **testbed optimized for voice delivery**. Use of multiple interfaces, path diversity and aggregation are shown to provide a combined improvement of 13 times in number of calls supported in our 15 node 802.11 mesh system.

When supporting both voice and TCP in a wireless multihop network, there are two conflicting goals: to protect the VoIP traffic, and to completely utilize the remaining capacity for TCP. In chapter 3, we investigate the interaction between these two popular categories of traffic and find that conventional solution approaches, such as enhanced TCP variants, priority queues, bandwidth limitation, and traffic shaping do not always achieve the goals. TCP and VoIP traffic do not easily coexist because of TCP aggressiveness and data burstiness, and the (self-) interference nature of multihop traffic. We found that enhanced TCP variants (Reno, Vegas, C-TCP, CUBIC, Westwood) fail to coexist with VoIP in the wireless multihop scenarios. Surprisingly, even priority schemes, including those built into the MAC such as RTS/CTS or 802.11e, generally can not protect voice, as they do not account for the interference outside communication range. We present ***VAGP* (Voice Adaptive Gateway Pacer)** - an adaptive bandwidth control algorithm at the access gateway that dynamically paces wired-to-wireless TCP data flows based on VoIP traffic status. *VAGP* continuously monitors the quality of VoIP flows at the gateway and controls the bandwidth used by TCP flows

before entering the wireless multihop. To also maintain utilization and TCP performance, *VAGP* employs TCP specific mechanisms that suppress certain retransmissions across the wireless multihop. Compared to previous proposals for improving TCP over wireless multihop, we show that *VAGP* retains the end-to-end semantics of TCP, does not require modifications of endpoints, and works in a variety of conditions: different TCP variants, multiple flows, internet delays, different patterns of interference, different multihop topologies, different traffic patterns.

The **interference map** of an 802.11 network is a collection of data structures that can help heuristics for routing, channel assignment and call admission in dense wireless networks. The map can be obtained from detailed measurements, which are time consuming and require network down time. In chapter 4, we explore methods and models to produce the interference map with a reduced number of measurements, by identifying interference properties that help to extrapolate complex measurements from simple measurements. Actual interference in an 802.11a testbed is shown to follow certain regularities – it is linear with respect to packet rate of the source, packet rate of the interferer, and shows independence among interferers.

When multiple cards are available, they behave differently, and even different channels of the same card have different performance. We find that while current methods of gathering the interference map may be appropriate for characterizing interference in one card networks, they are unscalable for multiple card networks when considering: 802.11 characteristics (card and channel asymmetries, time variation), required downtime, and complexity of the measurement procedure.

Availability of multiple antennas enables increased capacity or increased resilience for modern radios. This advantage depends on the deployment of the antennas at the sender and receiver. But there is a performance gap between most simulated results and the actual performance obtained in practice. This is due to the rank of the channel obtained in deployments, which depends on local propagation conditions, and on the placement of the senders and receivers. Using implementation on top of USRP platform and **mobile antennas**, we show in chapter 5 that it is possible to find 'good' antenna positions within a search space of a few carrier wavelengths. This opens the possibility for adaptive methods in antenna position and coding/modulation techniques to feed back to each other to reduce the gap between theoretical and practical MIMO performance.

Finally, in chapter 6, we present a few possible research directions, some continuing research described below, and some branching out into other areas of mobile computing.

Rezumatul tezei

Această teză cumulează diverse rezultate din domeniul mai larg denumit **mobile computing**, și include tematici din: poziționarea dispozitivelor, optimizarea procedurilor de comunicație, rutare, optimizarea configuratiilor de antene MIMO.

Încă de la adoptarea standardului IEEE 802.11 (neoficial numit WiFi) în 1997, reţelele care utilizează benzi fără licenţă au fost în creştere în popularitate şi a depăşit toate estimările iniţiale. Succesul lor a permis unor noi aplicaţii mobile, dar de asemenea a dat naștere unei serii de noi probleme pentru comunitatea de cercetători. Pe de o parte utilizatorii se așteaptă la noi aplicaţii şi funcţionalități de la noul LAN fără fir (WLAN), și pe de alta se aşteaptă ca reţeaua să se comporte ca LAN-urile convenţionale. Rezultatele prezentate în această teză sunt reproduse din lucrări ale noastre deja publicate [1, 2, 3, 4, 5], și arată că reţelele bazate pe **IEEE 802.11 sunt dificil de operat** în condiţii de densitate mare de dispozitive (datorită fenomenelor complexe de interferență), în condiţii de hopuri multiple (din cauza fenomenului de auto interferenţă a fluxurilor), dar și pentru unele condiţii de trafic mixt (TCP şi VoIP). Teza dezbate problematici ale rețelelelor 802.11 cu hopuri unice(WLAN), dar și cu hopuri multiple (mesh). Pentru cazul WLAN, se prezintă o metodă alternativă de utilizare a reţelei: în afară de comunicare, rețeaua este de asemenea folosită pentru furnizarea unui serviciu de localizare în interior. O altă direcţie de cercetare abordată pentru cazul WLAN este optimizarea legăturii radio folosind tehnologii cu antenă mobilă. Pentru cazul cu hopuri multiple, ne-am concentrat pe probleme legate de funcţionarea rețelelor mesh WiFi, care sunt utilizate pentru aplicaţii populare, de multe ori bazate pe TCP sau VoIP.

Angle of arrival (AOA) este o metodă utilizată pentru poziţionarea în exterior, de exemplu pentru poziționarea aeronavelor, dar şi pentru servicii de urgență cum ar fi E911. Pentru localizare în interior, cele mai bune soluții se bazează fie pe o infrastructură de mare anvergură (mulți senzori bine poziționați și calibrați), fie pe prelevarea de probe ale semnalului radio pe o grilă densă, care este supusă unor modificări de mediu, din cauza mișcării mobilierului, a lifturilor, sau chiar a utilizatorilor. În capitolul 1, prezentăm o arhitectură de poziţionare pentru interior, care nu are nevoie de o hartă detaliată cu puterea semnalului, ci necesită doar plasarea de stații speciale de bază VOR (**VORBA**). Deși prototipul de obținere a unghiului AOA pentru 802.11 utilizează o staţie de bază cu o antenă direcţională rotativă, soluție fără mobilitate fizică ar genera performanţe comparabile, chiar şi folosind unghiuri discretizate. Performanţa de poziţionare, cu staţiile de bază VORBA este evaluată prin analiză teoretică, simulare, și experimentare.

Se știe că performanța în reţelele fără fir multihop se degradează cu numărul de hopuri atât pentru TCP, cât și pentru UDP. Pentru VoIP, reţeaua fără fir aduce probleme suplimentare deoarece calitatea percepută este dependentă și de rata de pierderi, şi întârzierile din rețea. În capitolul 2, investigăm mai multe metode de îmbunătăţire a calităţii traficului de voce şi prezentăm rezultatele experimentale dintr-un **testbed 802.11b optimizat** pentru livrarea de voce. Utilizarea mai multor interfeţe, diversitatea căilor, şi agregarea traficului VoIP sunt combinate pentru a oferi o îmbunătăţire de 13 ori a numărului de apeluri acceptate în testbedul de 15 noduri de tip 802.11.

Atunci când se transportă atât voce, cât și TCP într-o reţea fără fir multihop, există de fapt două obiective contradictorii: protejarea traficul VoIP, şi utilizarea în întregime a capacitatății rămase pentru TCP. În capitolul 3, investigăm interacţiunea dintre aceste două categorii populare de trafic şi arătăm că abordările convenţionale, cum ar fi variantele îmbunătăţite de TCP, cozile de prioritate, limitarea lăţimii de bandă, sau modelarea traficului (traffic shaping) nu atinge întotdeauna scopul. Traficul TCP şi traficul VoIP nu coexistă cu uşurinţă în principal din cauza agresivităţii TCP burstiness, dar şi din cauza fenomenului de (auto-) interferenţă din rețelele multihop. Am constatat că variantele noi de TCP (Reno, Vegas, C-TCP, CUBIC, Westwood) nu coexistă corect

cu VoIP în scenariile multihop. Surprinzător, chiar şi schemele cu priorităţi, inclusiv cele implementate în MAC, cum ar fi RTS/CTS sau 802.11e, nu pot în general proteja traficul de voce, deoarece acestea nu iau în considerare interferenţa în afara razei de comunicare. În acest capitol se prezintă **VAGP** (Voice Adaptive Gateway Pacer) - un algoritm adaptiv de control al debitului la AP, care reglează în mod dinamic fluxurile TCP pe baza stării fluxurilor VoIP. VAGP monitorizează continuu calitatea fluxurilor de voce la AP si controlează latimea de banda folosita de TCP înainte de a intra în domeniul multihop, sensibil la auto interferenţă. Pentru a oferi o utilizare ridicată şi performanţă rezonabilă și pentru TCP, VAGP are mecanisme specifice care suprimă anumite retransmisii în domeniul multihop. Comparativ cu propunerile anterioare de îmbunătăţire TCP în multihop, VAGP păstrează semantica end-to-end a TCP-ului, nu are nevoie de modificări ale clienţilor sau serverelor, şi tolerează condiţii diverse: coexistenţa cu variante diferite de TCP, prezenţa fluxurilor multiple, întârzieri din internet, diferite situaţii de interferenţă, multihop cu topologii diverse, diferite modele de trafic.

Harta interferenţei pentru o reţea densă 802.11 este o colecţie de structuri de date care pot ajuta euristicile pentru rutare, alocarea de canale şi politicile de admitere a apelurilor în reţea. Harta poate fi obţinută din măsurători detaliate, care însă sunt consumatoare de timp şi oprirea reţelei pe durata măsurătorilor. În capitolul 4, vom explora metode şi modele pentru a produce harta de interferenţă folosind un număr redus de măsurători, prin identificarea unor proprietăţi ale interferenţei care să ajute la extrapolarea unor masuratori complexe din măsurători simple. Se arată că interferenţa reală într-un testbed 802.11a ă urmeze anumite regularităţi - este liniară cu rata de emisie a sursei, liniară cu a ratei de emisie a agentului de interferenţă, şi independentă între agenţii de interferenţă.

Atunci când mai multe carduri sunt disponibile, se constată că acestea se comporta diferit, şi chiar canale diferite ale aceleaşi interfeţe pot avea performanţe diferite. Se arată că metodele actuale de colectare a hărţii de interferenţă pot fi adecvate pentru caracterizarea interferenţei în reţelele cu card unic, dar ele sunt nescalabile pentru reţelele cu mai multe carduri. Această nescalabilitate se datorează: caracteristicilor 802.11 (asimetrii de card şi de canal, variaţii în timp), şi complexitatăţii procedurilor de măsurare.

Disponibilitatea de antene multiple permite creşterea capacităţii sau a rezilienţei pentru sistemele radio moderne. Acest avantaj depinde însă de configuraţia antenelor atât la emiţător, cât și la şi receptor. Dar în practică se observă o diferenţă marcată de performanţă între rezultatele simulate şi performanţele reale obţinute. Acest lucru se datorează calităţii canalului obţinut în plasamentele reale, care depinde de condiţiile de propagare locale, precum şi poziţia efectivă a anteleor la emiţători şi receptori. Folosind o implementare bazată pe platforma USRP şi **antene mobile**, în capitolul 5 se arată că este posibilă detectarea unei configuraţii "bune" a antenei într-un spaţiu de căutare redus. Acest spaţi este de ordinul a câteva lungimi de undă ale purtătoarei folosite. Această implementare deschide posibilitatea aplicării recursive a unor metode de cu feedback circular între optimizarea configuraţiei antenei si optimizarea codării/modulării pentru un canal obtinut.

În final, în capitolul 6, se prezintă câteva direcţii posibile de dezvoltare în continuare a carierei academice, planuri de cercetare pentru viitorul apropiat, și modul în care acestea se împletesc cu activitatea didactică.

Part II

Scientific and Professional Achievements

Chapter 1

Positioning using directional antennas

1.1 Introduction

Indoor positioning is a complex engineering problem that has been approached by many computing communities: networking, robotics, vision, and signal processing. In most of the proposed solutions, certain aspects of the problem domain are so specialized that a solution applicable in one domain does not easily translate into a solution in other domains. For example, many computer vision based techniques require line of sight conditions, and achieve high accuracy. In the context of 802.11 based position aware applications, line of sight is usually not available, but a lower accuracy may be tolerable.

There has been a surge of research activity in the pursuit of finding methods for accurate and robust indoor positioning techniques. Currently, the most convenient solutions for indoor positioning using 802.11 are RADAR [6], and its derivatives [7, 8, 9]. The problem with these approaches is that they require signal strength mapping of the entire area that has to support positioning, mapping which has to be reworked when propagation conditions change. As the human body roughly halves the strength of 2.4GHz signals, a crowded building might render the measurement map unusable. The advantage of these systems however lies in the fact they use off the shelf hardware which is widely available due to the popularity of wireless LANs and 802.11 hotspots. One application in which it is not feasible to build a signal strength map is that of an ad hoc disaster network. In such a setup there is no time to do signal strength mapping, and it may be impossible to deploy such a positioning infrastructure.

Methods that rely on range measurements as a function of signal strength are subjected to variance caused by the environmental surroundings which in previous studies have been shown to be a significant factor [10]. We propose to mitigate the effects of the environment on range measurements by use of a base station (or an access point) that has a rotating directional antenna. Using a 802.11 base station with a revolving directional antenna that can provide angle of arrival (AOA) and range measurements, we can obtain a more robust estimation than that based on signal strength from a standard 802.11 access point. By moving some of the complexity of the positioning support to the base station, we can significantly reduce the work and cost of deploying a range map in order to estimate indoor positions.

In this paper, we show that there are a number of ways of determining AOA from a 802.11 base station, one of which we have actually built, tested and obtained results with. Some of the newer schemes that improved on the RADAR idea showed better performance, but still retained the requirement of building a dense signal strength map of the building. Accuracy of positions obtained by our system, of 2.1m median error, is comparable to the original RADAR, but works without requiring a map of the signal strength of the area.

We show how to use the idea of VOR (VHF Omnidirectional Ranging) for indoor positioning using 802.11 hardware, with a **customized base station that can measure both angles and ranges**. In our incipient realization, the base station is a laptop equipped with a directional antenna that is continually rotating. We are able to derive angles towards mobiles with a median error of 22° and ranges to mobiles with a median error of 2.8m. These enable the use of trilateration procedures (for ranges), triangulation (for angles), and a combination of them.

In summary, here are the main contributions of the paper: first, an indoor positioning architecture that does not require a signal strength map. Second, a more robust positioning system that uses AOA and ranges obtained from a rotating directional antenna. Third, an actual implementation of the base station using off the shelf (garage sale) hardware. Fourth, analysis of position accuracy and robustness based on experiments, simulation and theory.

The rest of the article is structured as follows: the re-

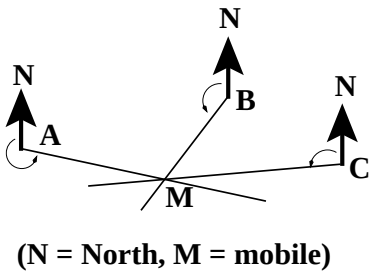


Figure 1.1: Basic triangulation and trilateration

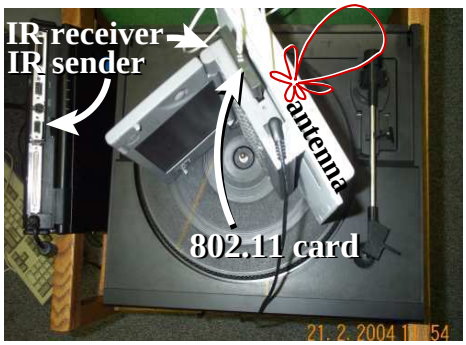


Figure 1.2: Experimental base station with revolving antenna

maining of the section reviews the basics of triangulation and trilateration; section 1.2 describes our implementation of the VORBA, how it infers angles, and a description of the measurement methodology. Section 1.3 presents several methods of positioning using VORBAs: based on discrete angles, angle distributions, quantized angles, and ranges. A theoretical analysis of the AOA based positioning is also included in this section. We review related articles in section 1.4, and conclude with some discussion and summary in section 1.5.

1.1.1 Triangulation and trilateration

Trilateration is a positioning procedure in which mobile M knows distances MA, MB, and MC, also known as ranges, in addition to the coordinates of landmarks A, B, C (Figure 1.1). By solving the nonlinear system :

$$\sqrt{(x_M - x_I)^2 + (y_M - y_I)^2} \quad = \quad MI, I = A, B, C$$

the mobile is able to find an estimate for its own position (x_M, y_M). This procedure is used by GPS with ranges obtained measuring time of flight to precisely synchronized and positioned satellites.

If no distances are known, but the mobile can find the angles under which it is seen by the landmarks, the **triangulation** procedure can be applied. Here the mobile knows the angles $\widehat{NAM}$, $\widehat{NBM}$, and $\widehat{NCM}$. These angles, together with the known positions of the respective landmarks determine half lines whose intersection is at the mobile M. Triangulation has actually been used prior to trilateration, because angles are easier to measure than ranges using simple mechanical / optical methods - for example in topometry, topography, air and sea navigation. One such example is VOR - a ground based navigation aid that still is the primary navigation system for the majority of aircrafts, even after the introduction of GPS. Its principle was the main inspiration for this paper: a landmark sends two signals, one that is periodic and omnidirectional, while the another one is directional and rotating about the landmark. The airborne equipment receives both signals, and interprets the difference between the times of the signals as an angle under which the landmark sees the aircraft. The coordinates of the landmark are known, therefore placing the aircraft anywhere on a given line. A second VOR reading provides a second line to be intersected with the first to yield a position.

In the next section, we describe VORBA, a prototype base station that measures ranges and angles, enabling mobiles to use triangulation, trilateration, and other positioning procedures.

1.2 VOR base station realization

A way to have AOA functionality on a 802.11 base station is to attach a directional antenna to a wireless access point. When this antenna is rotated, the SS (signal strength) reported by the card is higher in the direction of the mobile, and possibly in other directions as well, due to reflections. To automatize this measurement of the angle, we mounted a small Toshiba Libretto 70ct laptop on a record player (turntable) as shown from a top view in Figure 1.2. In order to obtain higher difference in the maximums, we chose an antenna that is highly directional. We linked the Lucent 2Mbps 802.11 card to a Hyperlink 14dB gain directional antenna that has the horizontal radiation pattern shown in Figure 1.3. The vertical pattern is almost identical, the main feature being that the strongest signal is spread only 30° from the center. The antenna is attached to the bottom of the laptop, so that it rotates in the horizontal plane.

One revolution of the turntable corresponds to a complete sweep of all angles in $0..2\pi$. About 500 samples of the SS can be associated with each angle at a 33RPM speed of the turntable (a period of 1.8 seconds). $SS(\alpha)$ is a function describing the strength of the signal of the

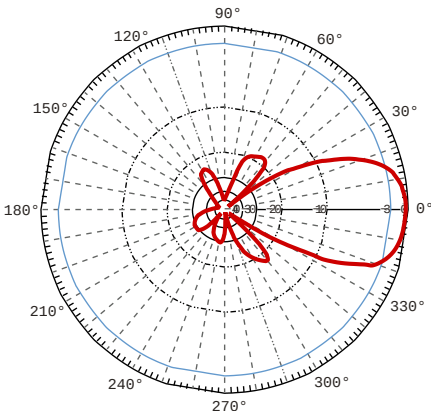


Figure 1.3: Hyperlink HG2414 directional antenna gain pattern

mobile as seen from the base station. **This function provides all the information needed to derive angles and ranges for triangulation and trilateration.** Most of section 1.3 shows how a mobile can derive angle information by using $SS(\alpha)$ from several base stations, while subsection 1.3.4 explains how ranges are derived from the same information.

If the revolution speed would be constant, $SS(t)$ measurements in time could be directly translated into angles with respect to North to obtain $SS(\alpha)$. But a problem we encountered early on during experimentation phase was the instability of the turntable period, that varied albeit slowly between 1.7 and 2.3 seconds. This would produce SS samples at variable rates, which wouldn't theoretically be a problem if we could accurately associate each SS reading with an angle. Ideally, it would be convenient to use a digital compass attached to the laptop, but this option was not available due to lack of interfaces on the laptop. Also, compasses might be disturbed near power cables, or large metal objects. We opted instead for synchronization using infrared once every revolution. A continuous IR signal is sent from the device on the left in Figure 1.2, in this case another laptop. When the revolving antenna on the laptop perfectly aligns its receiver with the fixed IR beam, the base station knows it has hit the horizontal axis of the system, which provides for a good continuous calibration of the system. The fixed IR beam indicates the horizontal axis of our coordinate system (opposite direction), so that all base stations report angles with respect to the same reference.

1.2.1 Measurements

The mobile requires an $SS(\alpha)$ from each base station, but for the purpose of experimentation, we only realized

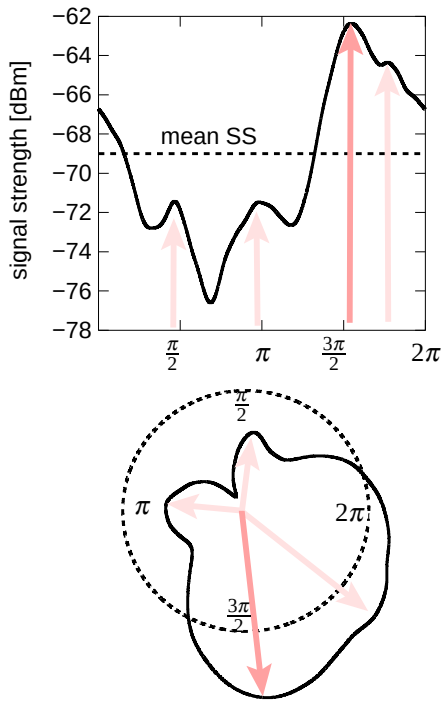
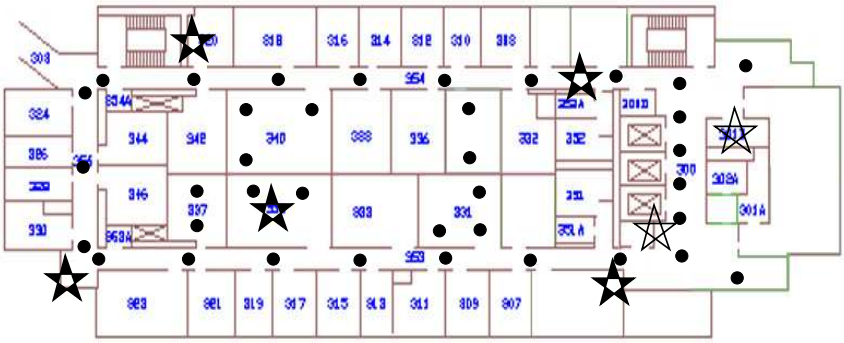


Figure 1.4: A peak in signal strength indicates a possible direction of the mobile. Cartesian (top) and polar (bottom) representation of $SS(\alpha)$.

one VOR base station and took several sets of measurements for different positions of the base station. Figure 1.5 shows the $56m \times 25m$ building in which 32 measurements points were taken (possible positions of the mobile), indicated with black dots. VOR base stations were placed at locations indicated by stars. Five base stations are sufficient to cover most of the building except the right side, where most points are separated from the base stations by three large elevators. In some experiments, we used two extra base stations indicated with hollow stars only for this side of the building.

In each measurement point, a regular 802.11 equipped laptop took four sets of measurements, one for each pose of the mobile (facing North, East, South, and West). No compass was used for orientation, the user just aiming to have the measurement laptop parallel to the walls. The position of the user was randomized in order to include in the measurements the situations in which both a human body and a laptop screen block the shortest path towards the base station. Polarization doesn't seem to matter, but we performed all the measurements keeping the 802.11 card in an horizontal plane, as is standard in most laptops. A measurement for a pose is in fact an average over three or four revolutions of the base station, in order to reduce the effect of temporary factors, such as open doors, or people moving by. Measurements were taken at various times of the day and night, including the busy morning and after-

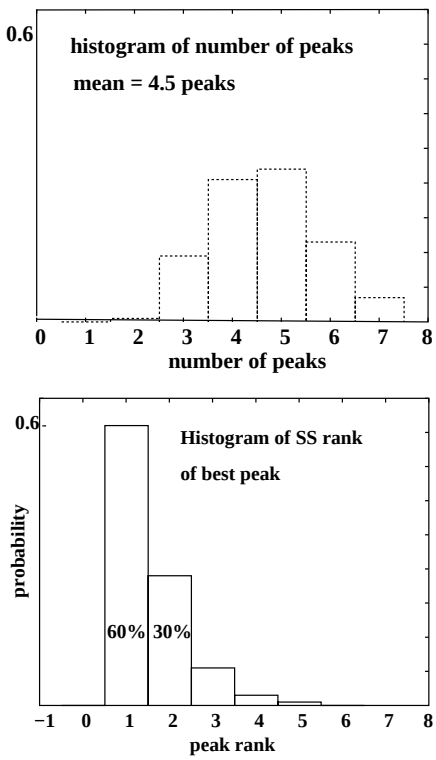


● = sample point ★ ☆ = VOR base stations

Figure 1.5: Map of VOR base stations and sample points

noon hours.

In the initial phases of experimentation, we decided to take as many SS samples per revolution as possible in order to maximize the information in $SS(\alpha)$. The current setup supports 250 pings per second, yielding almost 500 samples per revolution, and any additional traffic would probably introduce jitter in the angle measurements unless some prioritizing for the probe traffic is used. On the other hand, since VORBA is an extension for data access points, user data can be used to sample SS without the need for evenly spaced probes, when a synchronization mechanism such as NTP is available. The necessary accuracy for such a mechanism is on the order of milliseconds ($1^\circ \simeq 5ms$). Therefore, even if in the current experimental setup the positioning support is using all the bandwidth available for evenly spaced SS sampling, it is possible to ameliorate this by a number of methods: 1. employ user traffic for sampling; 2. broadcast the angle periodically from the base station (no RTS, CTS, or ACK overhead); 3. reduce the sampling rate. This last method is probably the most effective, because the high frequencies of $SS(\alpha)$ are not usable anyway. To obtain the signal in Figure 1.4, we employed a filter of 0.2 seconds to smooth the 2 second periodic signal.



(a) the optimal peak is one of the strongest two peaks in 90% of the cases.

Figure 1.6: Signal peaks are ranked based on the signal strength of the peak.

1.2.2 AOA inference

As will be shown in this section, in most cases the best direction towards the mobile is indicated by one of the strongest peaks in $SS(\alpha)$. To extract the two most powerful peaks of the signals we experimented with several heuristics, including voting between peaks produced by all four poses and averaging the functions from the four poses. Choosing the right peak is a critical part of the system and still a subject of experimentation, but for the results presented in this paper we use the signal pattern obtained through the following procedure: samples from all periods for all poses are each shifted in the interval $[0, 2\pi)$; samples are then sorted based on their corresponding angle and a smoothing Gaussian filter with the size 10% of the period is applied; the two strongest peaks are retained.

An $SS(\alpha)$ measurement is shown in Figure 1.4. As we

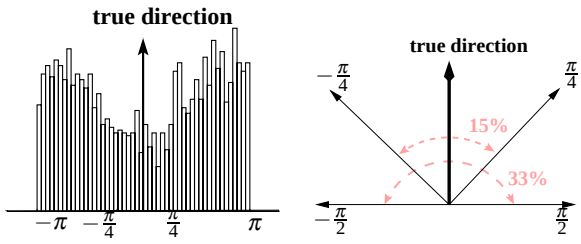


Figure 1.7: Non optimal peak distribution

expected, there are several peaks, or local maximums, indicated with arrows from the center in the polar representation. The darker arrow indicates the peak towards the actual position of the mobile, within a few degrees. In this case, the best direction is the strongest signal, but that is not always the case. We measured $SS(\alpha)$ in 32 points, each with four orientations, for five base stations, and ranked the peaks based on their SS value. As shown in the right of Figure 1.6, we found that in 90% of cases the best direction is either the first, or the second peak. If we knew the best of the two, it would have an error of 22° standard deviation from the true direction of the mobile. Not knowing which of them is best, we have to use both of them in triangulation.

It is now important to see how the non optimal peaks are distributed, because they have to be somehow integrated in the triangulation procedure. Even when working with the strongest two angles, it is not known which of them points towards the mobile. In the left Figure 1.6, we see that the total number of maximums obtained is between 2 and 8, with an average of 4.5 and a seemingly Poisson distribution. In Figure 1.7, we eliminate the best peak and find that the other maximums lie mostly away from the it. Only 15% of the times other maximums are in a 90° sector towards the mobile, and only 33% of the times in a 180° sector.

Based on these statistics, it is likely that working only with the strongest two maximums will cover most situations, while the other maximums are mostly grouped in a direction away from the mobile. The next question is how to use the two angles communicated from each base station to infer a position for the moving mobile. In fact, in one of the methods proposed below, we can use not just the two strongest directions, but the entire shape of $SS(\alpha)$.

1.3 Positioning with VORBA

In this section, we detail several methods of determining the position based on the information provided by base stations. Two of them are based on angles derived from $SS(\alpha)$, and the last one is based on ranges derived from the average signal strength $\frac{1}{2\pi}\int_0^{2\pi} SS(\alpha)d\alpha$.

1.3.1 Positioning using discrete angles

The problem of intersecting lines on a plane can be as simple as solving a linear system with one equation describing each line, were it not for the errors that can affect these lines, and for the fact that triangulation actually uses half lines. Since linear system solving typically optimizes for the sum of squares of vertical offsets, it may not be the best method to intersect lines affected by AOA errors. In our case a line is defined by the position of a known base station, and an angle affected by error. As will be shown later, this error has a normal unbiased distribution for measurements in a large departmental building.

Positioning using lines from several base stations can then be cast as an estimation problem. Let $\beta_i(\mathbf{x})$ be the true angles at which the base station i sees the mobile, α_i the measured angles, σ_a the variance of the AOA measurements, and $\mathbf{x}$ the position of the mobile. Based on the measurements available ($32\,points \times 7\,base\,stations$), the distribution of the angle error looks either Laplacian, or Gaussian, and we used the latter for the following analysis. For the implementation the Laplacian proved better for triangulation, because of the module being more robust to outliers.

Given that the measurements to different base stations are independent, the likelihood function is:

$$P\{\alpha_1,\alpha_2,\ldots|\mathbf{x}\} = \Pi P\{\alpha_i|\mathbf{x}\} = \frac{e^{-\frac{1}{2\sigma_a^2}\sum|\alpha_i-\beta_i|^2}}{\sigma_a^n\sqrt{(2\pi)^n}} \quad (1.1)$$

The maximum likelihood estimate (MLE) is the solution to the equation:

$$\frac{\partial \ln P\{\alpha|\mathbf{x}\}}{\partial \mathbf{x}} = 0$$

which leads to least square estimate that minimizes the square error in the fit to the angle data α_i. The function to minimize is therefore

$$\sum(\alpha_i - \beta_i(\mathbf{x}))^2 = \\ = \sum(\alpha_i - atan2(y-y_i, x-x_i))^2 \quad (1.2)$$

where x_i, y_i are the coordinates of base station i, $\mathbf{x} = x, y$ is the candidate solution point, and `atan2` is a function that computes the polar angle of $(x-x_i, y-y_i)$. If large

outliers are present, then m-estimators [11] can be used to optimize for a different objective giving less weight to the outliers. We optimized for the sum of modules, as opposed to the sum of squares also because the distribution of the angle errors may be Laplacian, and not Gaussian. In the implementation, to optimize for the objective we used the Nelder-Meade simplex method that is provided by the command `nmsmax` in the free software package octave-forge [12].

1.3.1.1 Choosing the best angle

Most positioning schemes, including triangulation, assume that an estimate angle towards landmarks is available with a certain error. In our case however, we have two angles, and the best peak is among them in only 90% of the cases. Even if we work with two angles per base station, the question remains which of the two angles to use. In 10% of the cases, both angles will point away from the mobile, so we need a method to identify "bad" angles. For n of base stations, using all combinations yields a maximum of 2^n possible intersections to be optimized using objective function (1.2).

Fortunately, many of these intersections happen either outside the feasible space (outside the building), or do not corroborate among all base stations. In order to avoid the exponential number of intersections, we first compute in $O(n^2)$ time a $2n \times 2n$ boolean incidence matrix A describing whether any two directions intersect inside the feasible space. An additional $2n \times n$ matrix B is obtained by adding columns of the first matrix to describe incidence between a given direction and a base station (any direction from that base station). If $B(d, j) = 0$ it means that direction d from base station $\lceil \frac{d}{2} \rceil$ does not intersect any direction from base station j, therefore it can be eliminated completely as a candidate. But most combinations are eliminated during the direction assignment phase, when the matrices indicate that an assignment conflicts with a past or future assignment. When running with 5 base stations, the number of candidates obtained with this method was between 1 and 16, with a median of 5. For 7 base stations, there were a maximum of 24 candidates, with a median of 6.

In order to choose a candidate angle, we compare the ranking of the distances to the base stations with the ranking of the SS to the same base stations. The signal strength used for ranking is obtained by averaging over all the samples, by integration of the signal in Figure 1.4 (integration of $SS(\alpha)$).

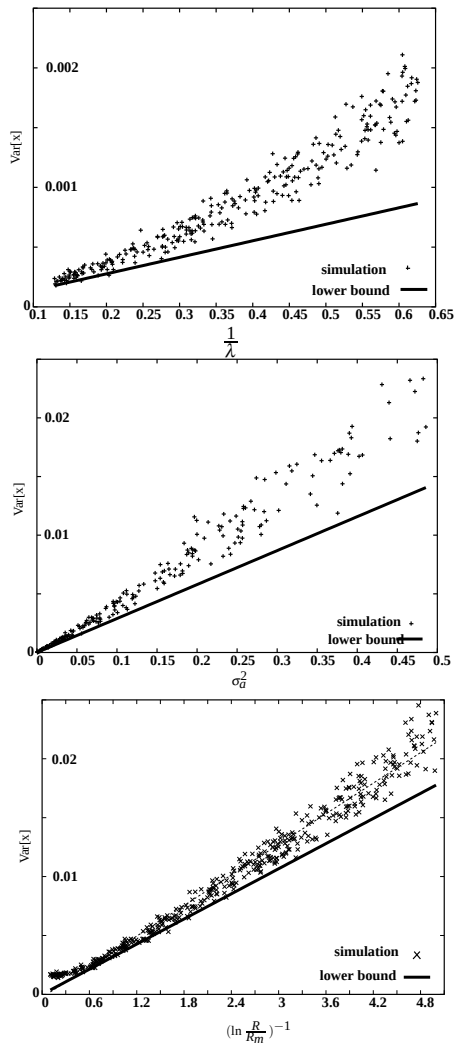


Figure 1.8: Position error variance depends linearly on σ_a^2 (angular error variance), on $1/\lambda$, and on $\ln^{-1} \frac{R}{R_m}$ (λ=base station density, R=mobile range, R_m=minimum distance to a base station).

1.3.1.2 Analysis of discrete angle positioning

An important question is how accurate a position can be obtained only with angles, and how it depends on the main parameters: density of base stations λ, and quality of the angles (variance σ_a^2). The answer helps in provisioning the deployment of the base stations in order to achieve a certain error.

To simplify the theoretical analysis, we assume the deployment area to be circular, with radius R, and base stations spread with a Poisson spatial distribution with rate λ. The first assumption is reasonable because the coverage will on average be circular in a large enough building (50-100m range for indoors 802.11). The second assumption however is more forced in the light of base station placement that might be preferential for reasons of coverage. It is nevertheless more general than the analysis of a particular placement such as in the corners of the building.

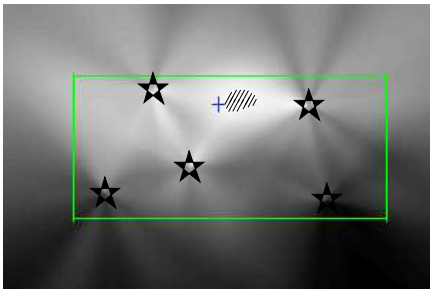


Figure 1.9: Map of position probabilities: lighter gray indicates higher likelihood.

In the appendix of this chapter (section 1.7) it is shown that the error covariance of the position obtained by triangulation is bounded by:

$$Var[x] \quad > \quad \frac{\sigma_a^2}{\lambda \pi \ln \frac{R}{R_m}} \tag{1.3}$$

where R_m is the minimum distance from the mobile to a base station. When $R_m \to 0$, the positioning error also becomes 0.

In order to verify the linearity of the positioning error with the angular error and with the inverse of the density, we ran a Montecarlo simulation in a circle with $R = 1$. Varying λ so that the expected number of base stations is between 5 and 25, for $\sigma = 0.1, R_m = 0.1$, produces the points in Figure 1.8a, showing the standard deviation in the obtained position in a linear relation with $\frac{1}{\lambda}$ together with the line corresponding to equation (1.3). In Figure 1.8b, $\lambda = \frac{15}{\pi}$, $R_m = 0.1$ while σ_a - the angle measurement error, is varied in the interval $[0, 0.7]$ radians, or $[0, 45°]$ showing the same linear dependence for the variance of the position. Each point in this experiments is obtained through averaging over 500 different positions of the base stations. The dependence on R_m is also verified in Figure 1.8c, for $n = 45$, $\sigma = 0.4$, and $R_m \in [0, 0.8]$, which places $\frac{1}{\ln \frac{R}{R_m}}$ in $[0.1, 5]$. In all the cases, the simulation verifies the trend and the bound indicated by equation (1.3).

The importance of the result implied by equation (1.3) is for dimensioning and deploying a positioning infrastructure. For example, in order to cut the deviation of the position in half, we need to either half the deviation of the angle measurement, or quadruple the base station density. Increasing the density and reducing the range is in many cases the only way to scale up the data access for more users, cheaper than upgrading all the hardware. The third factor, the minimum distance to a base station R_m, is not as useful as a control knob, as the user does not know how close he is to a base station.

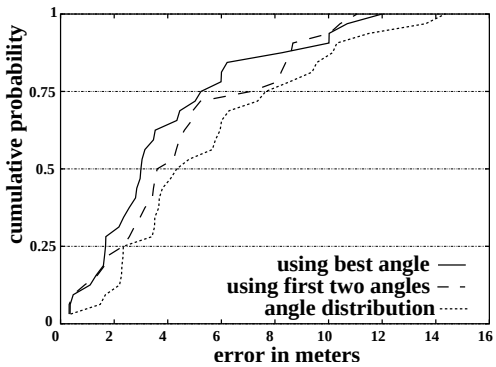


Figure 1.10: Cumulative distribution of positioning error using discrete angles and angle distribution.

1.3.2 Positioning using angle distribution

The pattern in Figure 1.4 indicates directions in which *SS* is maximum, and can be used to approximate a continuous density of probability for the direction of the mobile. By translating the *SS* values in the $[0..1]$ interval and scaling so that they sum to 1 for all angles in $[0..2\pi]$, we obtain a probability for each angle around the base station. There is a zero probability for the minimum signal, and proportional values for the other values of the signal. For each base station, the mobile computes the corresponding probabilities for each point of a 25cm grid covering the entire area of interest. The probability of a given point depends on the value of $SS(\alpha)$ in that direction. For each base station we have a probability map for virtually all the points in the area. Aggregating maps from different base stations produces a map like the one in Figure 1.9. In the map, we can distinguish the positions of the base stations, the same indicated by stars in Figure 1.5. The lighter areas have higher probability, and the true position in the upper corridor is indicated by a cross. By selecting the points with the highest probabilities, a maximum likelihood region is generated, shown as a hashed shape near the true position. The centroid of this region provides the estimated position.

The performance of positioning in the 32 points is shown as a cumulative distribution in Figure 1.10. The points in the main part of the building use the five main base stations, while the points separated by the elevators use three of the main base stations and two extra ones shown with hollow stars in Figure 1.5. The continuous line corresponds to an idealized performance that using the best measurement angle, the dashed line to using the two the angles and the heuristic in section 1.3.1.1, and the dotted line to the angle distribution method. The angle distribution method provides a slightly lower performance than the discrete method, but automatically deals

with outliers. It has a lower complexity with respect to the number of base stations, but higher with the area. It effectively takes into consideration all the possible angles, but it only builds one grid of probabilities for each base station, which are then merged in a final probability map. For the discrete method we used five base stations for the left main part of the building, and three base stations for the left part, separated by elevators, because the heuristic of choosing between two angles is more sensitive to outliers. Only the five main base stations were used for the angle distribution results.

To summarize, the idealized method using the best angle achieves a 2.9m median error, the heuristic using best two angles 3.5m, and the angle distribution method 4.1m.

1.3.3 Positioning using quantized angles

It is convenient to achieve the AOA functionality in a system without moving parts as the one we used. Steerable and switched beam 802.11 antennas are appearing on the market (www.vivato.net and www.paratek.com). They electronically steer the beam to provide preferential amplification for certain directions. In many cases when using a phased antenna array, and also in order to achieve a small form factor, the angle of arrival obtained is quantized in multiples of 45°. Another possible replacement for the mechanical part of VORBA is to just use eight different directional antennas at the base station. This would provide an angle of arrival quantized to 45°, but will also increase the total power output of the base station.

Without hardware that provides quantized angles, we are using our measurements from the VOR base stations discretized after the peak selection phase. In Figure 1.11a, there is an example of a point that is computed using readings from the five main base stations, using the strongest two peaks. The stars indicate the positions of the five main VORBAs, as represented in figure 1.5. From each base station, a continuous arrow indicates the first candidate angle, and the dashed arrow indicates the second candidate. The “o” sign indicates the true position of the mobile, while the “+” signs are the candidates considered by the optimization process. In Figure 1.11b, each angle is replaced with the closest multiple of 45° and the position is recomputed.

The quality of positions does not decrease dramatically when using quantization at 22.5° (16 directions), or at 45° (8 directions) - as shown in Figure 1.11c. The continuous line is the same in Figure 1.10 repeated here for reference - it represents the idealized case in which the best angle is provided by an oracle, not by the heuristic proposed in section 1.3.1.1. The results are somewhat predictable, since the 45° quantization introduces an error that is maximum 22.5°, similar to the measurement errors currently produced by the VOR base stations. For 90° quantization (4 directions), two of the points did not get a position, because of directions quantized to parallel lines, and we used the rest of the position errors to get a cumulative distribution. In this latter case, most of the points were optimized at values drawn from the coordinates of the base station themselves, meaning that the x coordinate is the x coordinate of some base station, and the y coordinate is drawn from possibly another one.

Using quantized measurements in 16 directions produces a 3m median error, a small degradation over the 2.9m of the idealized discrete method that uses the best of the two angles.

1.3.4 Positioning using ranges

In indoor situations, due to unpredictable propagation and fading effects, it is difficult to relate signal strength (SS) to distance. What motivated us to use SS for ranging is the fact that when using VOR base stations, SS can be more reliable being obtained from an integration over all angles, as opposed to a single arbitrary direction measurement. For a random pose of the mobile and for a fixed orientation of a regular base station, a SS sample would be a random value of $SS(\alpha)$ in Figure 1.4. Using VORBA, we can get a high resolution version of $SS(\alpha)$, and in this example there is a 15dB difference between the maximum and the minimum values that can be obtained. The mean value of $SS(\alpha)$ is a more accurate characterization of the SS from the mobile to VORBA simply because one factor (base station orientation) is eliminated.

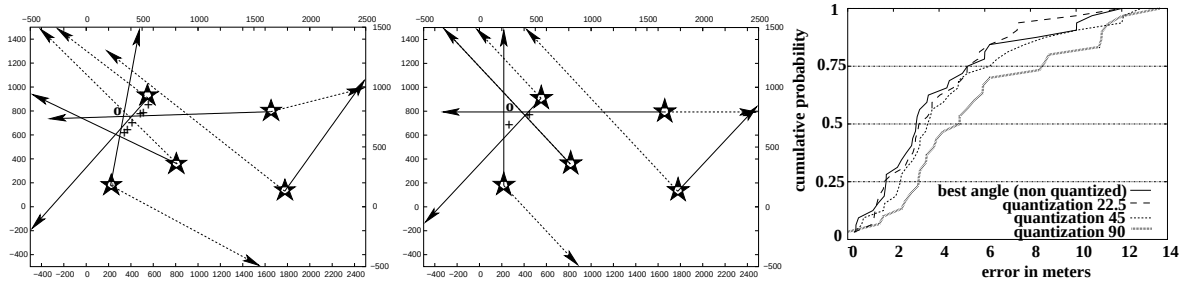
We verified this hypothesis by fitting SS values averaged over a revolution on a curve describing distance ρ as a function of SS in dB. Starting with the distance attenuation relation:

$$P[dBm] = P_0[dBm] - \log_{10}(\frac{\rho}{\rho_0})^n$$

we used the relation

$$\rho(P) = \rho_1 + \rho_0 \exp(\frac{P_0 - P}{n}) \qquad (1.4)$$

where ρ_1 is a shifting factor introduced to allow for differences in our hardware, whereas P_0 and ρ_0 are not needed both by the fitting procedure. Parameters ρ_1, ρ_0,



(a) positioning with actual angle measurements.

(b) positioning with measurements quantized to 45°

(c) cumulative errors for quantizations at 22.5°, 45°, and 90°.

Figure 1.11: Positioning with quantized angles.

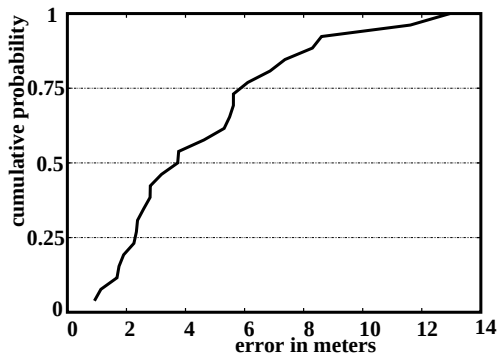


Figure 1.12: Cumulative distribution of positioning error using ranges from SS fitting

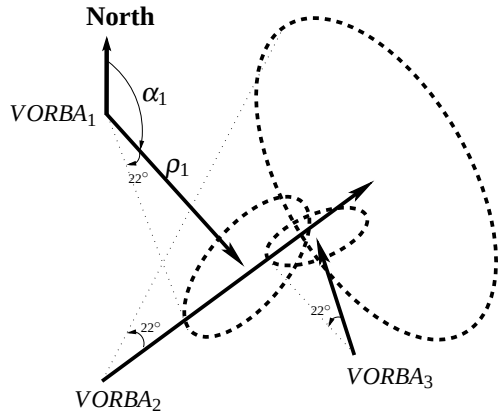


Figure 1.13: Position covariance is represented as an ellipse. Its size depends on the range ρ, and its orientation on the angle α.

and n were fairly similar for all base stations mainly because we used modulo residuals in the fitting procedure, in order to give less weight to large outliers. Especially the two base stations placed on the corridor had very large outliers - a much stronger signal with respect to distance.

The next step was five fold cross-validation: with parameters ρ_1, ρ_0, and n fitted from one base station we generate ranges from the SS readings for all the other base stations, and trilaterate using only these ranges. The procedure is repeated for the five main base stations (including the corner ones, which produce large outliers), and the positioning errors are cumulated into one distribution. Errors similar with those obtained from angles are shown in Figure 1.12.

The advantage of the method is that using some limited mapping of the signal strength, for example readings at various known points, a reliable curve (1.4) can be obtained that can be used for most other base stations. Distance measurements from signal strength can also be used to enhance angle measurements and assist candidate selection for discrete angle positioning (section 1.3.1.1).

The disadvantage is that some sampling of the signal map is needed for the fit, whereas the angle based methods provide positions just with placement of VOR base stations. This issue can be addressed using permanent stationary emitters as in [13, 9] which makes the resampling automatic.

The positioning errors for the range based methods are higher than those obtained with the angle based methods, with a median position error of 4m when fitting all the SS measurements, and 4.5m with cross validation.

1.3.5 Positioning using ranges and discrete angles

When the mobile knows both the angle α under which it is seen by a base station at (x_b, y_b), and the distance ρ to that base station, it can have an estimate of its position as

$$(x_b + \rho\cos(\alpha), y_b + \rho\sin(\alpha)) \tag{1.5}$$

What is the accuracy of this estimation? We know that the angle has an error with a standard deviation of 22°, with the current performance of VORBA. This means that even with a perfect range at 50m, the mobile can easily be placed 20m away from its true location. Ranging from integration of $SS(\alpha)$ however is not perfect, in fact errors in

ranging vary linearly with the actual range, as was found by other projects. After fitting the SS versus distance measurements as mention in section 1.3.4, we chose one of the non corridor base stations and used the fitting parameters obtained from it. Looking then at all measurements for all base stations, we found that standard deviation is 15-25% of the actual estimated range. We therefore assume $\sigma_r = 0.2$ as a relative deviation, modeling actual error as linear with the measured range $\sigma(range) = \sigma_r \rho$. The median positioning error that we get using relation (1.5) for a single base station under these conditions is 8.4m. This is an overall figure, as individual base stations provided median errors as low as 3.9m, and as high as 11m.

We can use estimations of positions from several base stations, which we can aggregate into a single estimate. In this case, it is necessary to weigh faraway base stations less than close ones, because for a fixed error in angle measurements, the uncertainty in position increases with distance. To see why that happens, it is convenient to characterize the uncertainty in the position estimation using a covariance matrix. When we consider the polar measurements (ρ, α), with standard deviations of $\sigma_r \rho = 0.2\rho$ for the range, and σ_a for the angle, the covariance matrix of the estimated position becomes:

$$U = \rho^2 \begin{bmatrix} \tan^2(\sigma_a^2) & 0 \\ 0 & \sigma_r^2 \end{bmatrix}$$

This matrix shows the particular case in which the base station is placed in $(0,0)$ and the mobile in $(0,\rho)$ so that angular error produces variation only on the x axis and range error variation only on the y axis. In reality, the covariance matrix describing the possible positions of the mobile is rotated to reflect the actual measurement α. In Figure 1.13, we see how the covariance matrix is shaped and placed depending on actual measurements. The tip of the arrow indicates the position estimate for the respective base station, and the size of the ellipse the size of the covariance. All the covariances have the same aspect ratio: the width is given by the angle error $\tan(22^\circ) = 0.4$, and the height by the range relative error $\sigma_r = 0.2$, both of them scaled with the actual measurement ρ. If each base station estimate is $[x_i y_i]$ and has covariance U_i, we can combine them in a weighted estimate using a simplified Kalman filter: $[xy] = (\sum_{i=1}^{n} [\ x_i \quad y_i]\ U_i^{-1})U$, where $U = (\sum_{i=1}^{n} U_i^{-1})^{-1}$ is the covariance of the estimated position.

In Figure 1.14, the cumulative distribution of positioning error is presented for 1,3,5, and 7 base stations. As expected, the error improves as we add base stations. For one base station, we used each individual station and cumulated the results, whereas for 3 and 5 we used configurations that favored base stations on the corner of the building. The five base stations in the main part of the building in Figure 1.5 provided a median error of 3.3m, better than the previous schemes using only angles and only ranges. The curve corresponding to 7 used all the measurements taken. The median error achieved in this latter case is of 2.1m, a clear improvement, provided the rather large error in angle and range estimation.

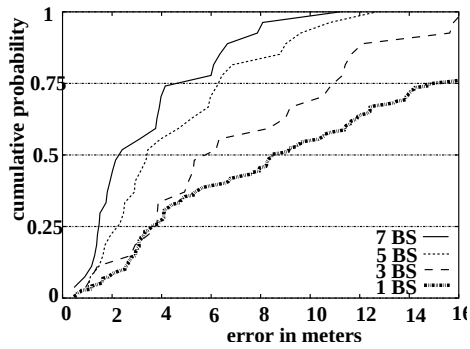


Figure 1.14: Cumulative distribution of positioning error using angles and ranges

1.4 Related work

Indoor positioning schemes can be classified based on the infrastructure they use, and on the type of measurement medium they employ. In terms of infrastructure, some systems may require specialized instrumentation of the area, and possibly line of sight to the mobiles, whereas others rely on the base stations that are already used for data access. The measurement medium is usually a choice or a combination of RF, infrared, and ultrasound.

Active Badge [14] was one of the early indoor systems, that provided each user with an IR badge that can be read by an IR station that keeps updating user's position in a central database. The position granularity is limited by the density of stations, and the reliance on light can be detrimental in the presence of spurious infrared emissions such as sunlight.

Active Bat [15] is a more recent project of the same group that uses ultrasound instead of IR, and has centimeter accuracy. The bats are supported by a grid of sensors placed on the ceiling which communicate with a central location that performs sensor fusion tasks to provide position through trilateration and also orientation.

Cricket [16] is an MIT project that makes use of ultrasound and per room infrastructure to achieve indoor positioning. It uses the six order of magnitude difference in the speeds of light and sound to achieve ranging to ceiling beacons. Mobiles then perform their own triangulation to position. Cricket compass [17], a followup project added the measurement of the orientation capability by using three ultrasound receivers on the mobile.

When compared to our VORBA system, these methods have the advantage of a higher accuracy. The disadvantage is that they require extensive support infrastructure, which translates in high cost of deployment. VORBA also requires extra infrastructure, but that is seen as an option for the data providing base stations rather than separate position providing infrastructure.

Another category of schemes relies on measurements of signal strength, and therefore can be used with existing wireless data infrastructure. RADAR [6] was the first system to propose the use of a signal map of the area. Average signal strength for each base station is stored as a fingerprint for each point in a dense grid covering the floor. When querying, a nearest neighbor match in the fingerprint space provides candidates for mobile's position. Ladd et all [7] improve on the the RADAR idea by using full distributions instead of just estimated expectations in each point. At query time, Bayesian inferences are used to search the grid for the most likely positions that fit the distributions of the query point. It improves on positioning performance of RADAR decreasing the median error.

The advantage of this class of systems is that they do not require any additional specialized infrastructure. The deployment cost however is dominated by the necessity of building the signal map of the floor. In addition, the measurements have to be re-taken when the propagation conditions change (people, furniture, etc). VORBA requires the specialized VOR base stations, but saves on the deployment and maintenance by not requiring major updates with changes in the environment. When only angles are used, no training is necessary, while for ranging, some limited training is required.

Landmarc [13] is an RFID based positioning scheme that is in a way similar to RADAR, except that the signal map is built on the fly by previously placed tags. A similar scheme of searching the nearest neighbor in signal strength space is used, but the system adapts more gracefully to changes. A similar idea is explored for 802.11 in [9]. The automatic way to sample the SS map using RFID tags in [13], and stationary emitters in [9] is applicable to VORBA for the range case, when some sampling is needed. However, because we use a parametric method (more robust because of rotation), less samples are needed, as explained in section 1.3.4.

The Lighthouse [18] location system is used for Smart Dust positioning, but is mentioned here for the similarity with our VOR approach in using a base station with a revolving antenna. Smart Dust are small sensors that only have optical communication capability, so they require line of sight to the base station. The lighthouse rotates its beam and based on the time a node senses the continuous light, it may infer its range to the base station.

Positioning in ad hoc and sensor networks [19, 20, 21, 22, 23, 24, 25, 26, 27] is the problem of positioning nodes based on ranges, angles, or simple connectivity. Most of these approaches face different issues such as the multihop nature of the ah hoc graph, and the requirement of having a distributed algorithm.

1.5 Discussion and future work

As presented in the theoretical analysis (section 1.3.1.2), the factors that affect accuracy of positioning in an idealized setup are density of base stations and quality of angles. Another factor is the actual triangulation procedure that in many cases has to deal with large outliers, therefore a question is how to eliminate an outlier base station reading from the process, without severely degrading the DOP [28]. DOP is dilution of precision due to the geometry of the base stations relative to the mobile, for example DOP is lower when the mobile is outside the convex hull of the base stations.

An aspect also related to the position resolution is the use of more than two discrete angles, but in current implementation we found it to be slower, and with reduced quality, because the extra angles introduce more noise (more candidates).

A number of facts that were found during experimentation may help identify more accurate mapping of the propagation patterns of a building using VOR base stations:

- there is no correlation between angle error and distance. We initially hoped to see such a correlation that could be used in the resolution of candidate positions.
- there is no correlation between angle error and mean signal strength. This is a direct consequence of the previous observation, since SS and distance are strongly correlated, as shown in section 1.3.4.

- corridors act as waveguides. It is better to place VOR base stations in rooms in order to achieve more accurate angles. This however would reduce "natural" amplification available in a building, which may be detrimental for data access, in order to enhance positioning.

Another way to enhance the accuracy of the $SS(\alpha)$ measurements would be to use a revolving signal at the mobile as well. This would not only simplify the self positioning procedure which currently requires measurements for four poses, but would enable a more accurate picture of the signal at both the base station and mobile.

Currently $SS(\alpha)$ is measured at the base stations because synchronization is easier. We found that the SS as a function of time has a similar shape when measured at the mobile, but with different phase. Provided that the mobile would have a solid frame of reference (compass), it would be possible to measure angles from the mobile towards the base stations, with respect to North. These are equivalent with the angles currently obtained, but the new method would offload some tasks from the base stations. On the other hand, if the mobile doesn't have a compass, it will report all angles to an arbitrary reference and a different type of triangulation must be applied. In this case the mobile only knows the angle under which it sees pairs of VORBAs, case which can be solved by a nonlinear optimization.

VORBA is supposed to be an extension to regular data access points, and not a separate infrastructure to support positioning, therefore an important issue that we are currently investigating is the data performance. The directional antenna used has an amplification of 14dB when facing the mobile, and a very small one in the opposite direction. While bit error rate is a function of signal strength, providing predictable effects on UDP traffic, it is not clear what the effects of VORBA on TCP performance are. This may be less of an issue if the sweeping antenna pattern is achieved in a non mechanical manner, and is only activated on demand.

A related issue is the time to position, and the possibility to do mobile tracking. The limiting factors are currently the need for several poses for the measurements, and the large rotation period of the base station. Currently, a user has to take four poses, waiting three revolutions of two seconds each yielding a total of $3 \times 2 \times 4 = 24$ seconds, while the triangulation time is almost negligible on current laptops. While the revolution period factor can be greatly reduced using electronic beam forming, operating with fewer poses and with fewer revolutions is less promising because of the inherent variability of the SS with respect to pose and time. If for example, the user takes the measurement only in one pose, it may happen that his body or the screen of the laptop is blocking a direct path to VORBA.

The original VOR used for aircraft navigation is a 2D scheme. VORBA has also been designed to take advantage of the rotation of antennas in a single plane to provide positioning only in that plane. However, the information provided by a single VORBA amounts to placing the mobile on a plane passing through the vertical axis of rotation of the antenna (in fact it is the half plane created by the rotation axis and the direction of the mobile). Two VORBAs are sufficient for positioning in 2D because in addition to the two planes provided by the base stations, there is a third implicit plane of the 2D setup. VORBAs rotating around other axes (other than vertical) could theoretically provide additional planes for intersection. If the angle error is independent of the altitude with respect to the base station, the problem is a simple extension of the 2D case. In the more likely situation in which the error does depend on altitude, a problem for a 3D setup is to provision the directions of the rotation axes in order to minimize positioning errors. An additional aspect in 3D is the possible variation in polarization. In the current realization, the mobile has the card always in the same horizontal plane VORBA uses for rotation, but if base stations rotate in arbitrary planes, the polarization between the base station and the terminal may change in the course of every revolution.

1.6 Conclusions

VORBA (VOR base station) is a prototype base station that provides **angle and range measurements** using 802.11 signal strength. Its basic idea is to find the strongest maximums in the signal strength, and use them as the most likely directions in which the mobile can be. The current realization uses a revolving antenna, but a non mechanical implementation, even using quantized angles, would yield similar performance. A positioning architecture using VOR base stations and triangulation has the advantage of not requiring extensive measurements of the signal strength map, while providing performance similar (2.1m median error) with systems that require such sampling. When limited sampling is acceptable, the VOR base station can provide robust range estimations that can

be used for trilateration.

1.7 Appendix: Lower bound for angle only positioning

The Cramér-Rao lower bound is method that sets a lower bound on the variance of *any* unbiased estimator. In our case the triangulation problem is cast as an estimation problem by considering the true position $\mathbf{x}$ as the parameter to be estimated.

Given a circle with radius R, n VOR base stations are Poisson distributed with density λ. Coordinates of the base stations are (x_i, y_i) Cartesian, and (ρ_i, β_i) polar. Assume the mobile is in the center of the circle, $E[\mathbf{x}] = (0,0)$, but rhe angle readings $\alpha = [\alpha_1, ..., \alpha_n]$ are described by equation (1.1). The likelihood of being at position $\mathbf{x} = [x\ y]$, after having seen bearings from n base stations is:

$$\begin{aligned} L(\mathbf{x}|\alpha) &= \ln(p(\alpha|\mathbf{x})) \\ &= -\ln(\sigma^n\sqrt{(2\pi)^n}) - \frac{1}{2\sigma^2}\sum(\alpha_i - \beta_i)^2 \end{aligned}$$

Using the likelihood, define

$$I = -\int_{-\infty}^{\infty} \begin{bmatrix} \frac{\partial^2 L}{\partial x^2} & \frac{\partial^2 L}{\partial x \partial y} \\ \frac{\partial^2 L}{\partial x \partial y} & \frac{\partial^2 L}{\partial y^2} \end{bmatrix} p(\hat{\rho}; \mathbf{x}) d\hat{\rho}$$

and the bound on the covariance of the position is given by I_{11}^{-1}. Second derivative of the likelihood $\frac{\partial^2 L(x|\alpha)}{\partial x^2}$ reduces to:

$$\sum_{i=1}^{n} \frac{2(x-x_i)(y-y_i)(\alpha_i - \beta_i) - (y-y_i)^2}{\sigma^2 \rho_i^4}$$

where $\rho_i = \sqrt{(x-x_i)^2 + (y-y_i)^2}$ is the distance form the point to the base station i.

$$-\int_{-\infty}^{\infty} \frac{\partial^2 L}{\partial x^2} p(\alpha|\mathbf{x}) d\alpha = \sum_{i=1}^{n} \frac{(y-y_i)^2}{\sigma^2 \rho_i^4} \quad (1.6)$$

$$-\int_{-\infty}^{\infty} \frac{\partial^2 L}{\partial x \partial y} p(\alpha|\mathbf{x}) d\alpha = \sum_{i=1}^{n} \frac{(x-x_i)(y-y_i)}{\sigma^2 \rho_i^4} \quad (1.7)$$

because $E[\alpha_i] = \int_{-\infty}^{\infty} p(\alpha_i|\mathbf{x}) d\alpha_i = \beta_i$.

To compute (1.6) and (1.7), we use polar coordinates (ρ_i, β_i) for all cartesian points (x_i, y_i): $\frac{y-y_i}{\rho_i} = \sin(\beta_i)$, $\frac{x-x_i}{\rho_i} = \cos(\beta_i)$, $(1.6) = \frac{n}{\sigma^2} E[\frac{\sin^2(\beta)}{\rho^2}]$ and $(1.7) = \frac{n}{\sigma^2} E[\frac{\sin(2\beta)}{2\rho^2}]$. It is known that the polar coordinates are independent, so we only need to compute $E[\frac{1}{\rho^2}]$, $E[\sin^2(\beta)]$, and $E[\sin(2\beta_i)]$. The distribution of β is uniform, with pdf $f_\beta = \frac{1}{2\pi}$, and it can be shown that $f_{\cos^2(\beta)}(s) = \frac{1}{\pi\sqrt{s(1-s)}}$, which yields $E[\sin^2(\beta_i)] = E[\cos^2(\beta_i)] = \frac{1}{2}$. Similarly, $f_{\sin(2\beta)}(s) = \frac{1}{\pi\sqrt{1-s^2}}$, with an expectation $E[\sin(2\beta_i)] = 0$.

The c.d.f. of distances to base stations is $F_\rho(s) = \frac{s^2}{R^2}$. Let $m = \frac{1}{\rho^2}$ a random variable with $m \in [\frac{1}{R^2}, \infty)$.

$$F_m(s) = P\{m < s\} = P\{\frac{1}{\rho^2} < s\} = 1 - \frac{1}{sR^2}$$

$$E[m] = \int_{\frac{1}{R^2}}^{\infty} s f_m(s) ds = \frac{1}{R^2} ln(s)|_{\frac{1}{R^2}}^{\infty} = \infty$$

The interpretation of this is that error can become arbitrarily small when the mobile is getting infinitely close to the base station. For this reason, we use R_m - the minimum distance to the base station a mobile can have.

$$E[m] = \frac{1}{R^2} ln(s)|_{\frac{1}{R^2}}^{\frac{1}{R_m^2}} = \frac{2}{R^2} \ln \frac{R}{R_m}$$

$$A = nE[\frac{\sin^2(\beta)}{\rho^2}] = nE[\sin^2(\beta)]E[\frac{1}{\rho^2}]$$

$$A = \frac{n}{R^2} \ln \frac{R}{R_m} = \pi\lambda \ln \frac{R}{R_m}$$

$B = 0$, which produces
$I = \frac{n}{\sigma^2 R^2} \ln \frac{R}{R_m} I_2$, where I_2 is the 2×2 identity matrix.

$$Var(x) > I_{11}^{-1} = \frac{\sigma^2}{\pi\lambda \ln \frac{R}{R_m}}$$

Chapter 2

VoIP in WiFi based meshes

2.1 Introduction

In the recent past, there has been a tremendous proliferation of VoIP services in both residential homes and corporate offices. For example, in the corporate sector, Infonetics data forecasts that there will be massive increase in VoIP and SIP adoption by 2015 . In addition, the Skype service [29] providing free internet calls has recorded more than 10 billion minutes of call time in its first year of inception. The cost savings achieved by VoIP by using existing data infrastructures along with easy deployment benefits are the main reasons driving the steady growth of VoIP.

At the same time, VoIP over wireless LAN (WLAN) has the potential of becoming an important application due to the ubiquity of the WLAN in homes and offices. With the advent of dual cell phone handset with WiFi capabilities and soft-phones over PDAs, carrying voice over the WLAN is gaining a significant importance. Once VoIP over WLAN becomes widespread, most cell phone or WiFi handset owners will migrate to using VoIP over WLAN inside the administrative boundaries of the enterprise buildings, campuses, public places such as airports or even in WLAN equipped homes.

Providing VoIP users with true mobile phone services having the freedom of roaming requires wide area wireless coverage, and IEEE 802.11-based multihop wireless mesh networks have been considered a practical solution for wide area coverage. The benefits of mesh network compared to wired LAN connecting WiFi access points are: i) ease of deployment and expansion; ii) better coverage; iii) resilience to node failure; iv) reduced cost of maintenance. Such a mesh network has the potential of creating an enterprise-scale or community-scale wireless backbone supporting multiple users while driving these users from using fixed phones to wireless VoIP phones. A typical usage scenario is shown in Figure 2.3.

However, supporting delay sensitive realtime applications such as VoIP over wireless mesh networks is challenging. Although convenient and cheap, voice service over WLAN faces a number of technical problems: a) providing QoS sensitive VoIP traffic in presence of best effort TCP data traffic; b) packet loss due to channel interference by using unlicensed bands (2.4GHz, 5GHz); c) high overhead of the protocol stack - 802.11/IP/UDP/RTP for each VoIP packet with 20bytes payload. The above problems become even more severe when supporting VoIP over multihop mesh networks. In a multihop wireless network operating on a single channel, the UDP throughput decreases with number of hops for properly spaced nodes and is shown to be between 1/4 and 1/7 that of single hop capacity [30]. This phenomenon of self interference is produced by different packets of the same flow competing for medium access at different nodes. When all nodes are within interference range, the UDP throughput in a linear topology can degrade to $\frac{1}{n}$, where n is the number of hops.

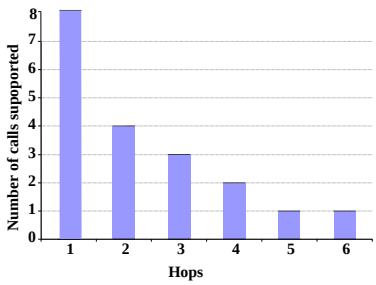


Figure 2.1: In a linear topology, capacity degrades with the number of hops.

As shown in Figure 2.1, our experiment on a real mesh testbed with G.729 encoded VoIP calls indicates that the number of supported medium quality calls decreases with the number of hops for a simple linear topology. In a mesh network with 2Mbps link speed, the number of supported calls reduces from 8 calls in single hop to one call after 5 hops. This significant reduction in the number of supported calls can be attributed to following factors: a)

decrease in the UDP throughput because of self interference; b) packet loss over multiple hops and c) high protocol overhead for small VoIP packets.In this work we focus on designing a 802.11 based wireless mesh network that can efficiently support the VoIP calls. Specifically, our main objective is to increase the number of calls that a multihop mesh network can support. We address several performance optimization issues that lead to significant benefits in capacity and in the quality of VoIP calls.

We describe our implementation of a 802.11 wireless mesh designed specifically to provide various VoIP related services. We focus on two important problems in supporting VoIP over wireless mesh network: increasing VoIP capacity and maintaining QoS under internal and external interference. We evaluate the performance of VoIP over the mesh network and provide various approaches for optimization of the overall system.

In particular, for increasing capacity in supporting more number of VoIP calls, we investigate on the following three directions: use of multiple interfaces, efficient routing, and use of multihop packet aggregation to reduce overhead. We present the individual performance benefits obtained by each of the above directions. For routing, we use label based forwarding and adaptive path selection to support fast path switching, call admission and mobility support.

We provide a multihop aggregation mechanism that uses the "natural" waiting time of packets in a loaded network. We show that our aggregation scheme does not increase delay while providing significant benefit in term of capacity increase. Each of the optimization schemes proposed are distributed in nature and does not rise scalability problems for large mesh networks. The above performance optimization techniques are implemented in a 15 node indoor wireless mesh network. The experimental results show an increase of 13 times for a six hop string when all optimizations are used (Figure 2.18).

2.1.1 Related work

Recently, with growing importance of VoIP, several research works have addressed the performance issues of supporting VoIP over Internet. The use of switching among multiple paths to reduce delay was proposed in [31] and recovering from packet loss was proposed in [32]. These strategies were used for delivering VoIP using an overlay network. When transporting VoIP over the Internet, the major factor affecting the performance is path delay as for good quality, VoIP requires 200ms or less one way delay. In wireless networks, the main factors affecting VoIP performance are the low capacity and the variable loss rate.

Some initial studies on the performance of real-time applications over 802.11 were presented by Sobrinho and Yeh in [33, 34]. References [35, 36] focused specifically on VoIP over 802.11 considering the delay and loss characteristics under PCF and DCF modes. Another recent work on VoIP over WLAN [37] presents analytical studies on the number of calls that can be supported in a single hop WLAN. The study reports that increasing the payload per frame increases the number of supported calls. Our work uses this observation for a multihop scenario to design the voice packet aggregation scheme.

Several performance optimization schemes were proposed for VoIP over WLAN: Yu et al. [38] propose the use of dual queue of 802.11 MAC to provide priority to VoIP, while Wang et al. [39] propose packet aggregation to increase capacity.

Recently, research has been conducted in the area of 802.11 based wireless multihop mesh networks. A study conducted to understand the capacity of multihop network was presented in [30]. Research on improving the end-to-end performance of application on multihop network by employing multiple radios was considered in [40] and [41]. Further work on finding better routing metrics and strategies for multihop networks was presented in [42].

2.2 VoIP basics

A VoIP system consists of an encoder-decoder pair and an IP transport network. The choice of vocoder is important because it has to fit the particularities of the transport network (loss and delay). One of the popular voice encoders is G.729, which uses 10ms or 20ms frames. It is used by some available 802.11 VoIP phones (such as the Zyxel Prestige, Senao S7800H, and by other wired VoIP phones as well). The Zyxel Prestige for example, sends 50 packets per second, of 20 bytes each, and we chose this traffic specification for experiments and simulations throughout this paper. Although a 30% utilization increase is generally expected when accounting for periods of silence when no packets are sent, we do not consider silence periods (the Zyxel phones and the Skype application also do not use silence suppression). To measure the quality of a call, we used a metric proposed in [43], which takes into account mouth to ear delay, loss rate, and the type of the encoder. Quality is defined by the *R-score*, which for

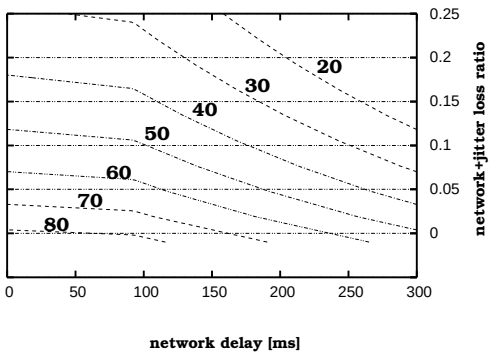


Figure 2.2: *R-score* for 60ms jitter buffer, G729, uniform loss.

medium quality should provide a value above 70:

$$\begin{aligned} R &= 94.2 - 0.024d \\ &- 0.11(d - 177.3)H(d - 177.3) \\ &- 11 - 40log(1 + 10e) \end{aligned}$$

where:

- $d = 25 + d_{jitter\_buffer} + d_{network}$ is the total ear to mouth delay comprising 25 ms vocoder delay, delay in the de-jitter buffer, and network delay;
- $e = e_{network} + (1 - e_{network})e_{jitter}$ is the total loss including network and jitter losses;
- $H(x) = 1\, if\, x > 0; 0\, otherwise$ is the Heaviside function;
- the parameters used are specific to the G.729 encoder with uniformly distributed loss.

The constants consider the delay introduced by the encoder for its lookahead buffer, and the delay introduced by the jitter buffer. We considered a jitter buffer of 60ms, which has two contradictory effects: it increases end to end delay, therefore degrading the quality, but it also reduces the jitter, which has an overall better effect. The *R-score* is finally computed only from the loss and the delay in the network, which can be measured directly in our testbed. In order to emulate the behavior of a simple jitter buffer, we assume that playout starts at the destination after the arrival of 4 packets from start (60 ms jitter buffer = 3 packets). Therefore all the deadlines for the packets at the receiving side are established at this point. Loss in the jitter buffer is computed as the fraction of packets which do not meet their deadlines. In order to compute loss probabilities and average delay in the network, all packets from all flows in an experiment are considered together by macro-averaging.

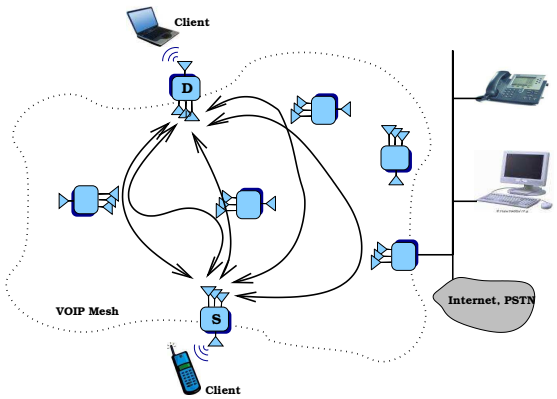


Figure 2.3: Mesh system showing two clients connected, and the paths maintained between them. Each mesh node has one separate interface for the clients, in addition to backhaul interfaces. Clients can connect across the mesh to other wireless devices, in the institution intranet to wired VOIP phones, to the internet, or to the PSTN.

Figure 2.2 shows the values of the *R-score* with respect to network delay and total loss for 60ms jitter buffer and 25ms vocoder delay. The interpretation of the iso-*R-score* curves is that for example to obtain an *R-score* of 70, the network has to deliver all packets in less than 160ms, or deliver 98% in less than 104ms. From the figure we can see that the quality is sensitive to even a couple of percents of loss, whereas the delay tolerates differences in tens of milliseconds. In 802.11, loss has a high variance, as it depends on the quality of the channels and the cards, and on the interference from external or internal sources. In a multihop setup, end to end loss is difficult to control and needs to be maintained under 2%. Using the retry mechanism of 802.11, this loss can be reduced at the cost of increasing delay.

2.3 VoIP mesh system

To evaluate performance and capacity, we deployed an 802.11b based wireless mesh network for supporting VoIP traffic. In this implementation, the mesh network is considered a multihop extension of the access point infrastructure existent in most institutions. It is useful to use the concept of a layer 2 switch to see the entire mesh as a single element that switches packets between its ports. A port is in fact a mesh node which has at least two interfaces: one in ad hoc mode for the backhaul in the mesh, and one in infrastructure mode to connect to clients (Figure 2.3). These clients can be VoIP wireless phones, laptops or other wireless handhelds. To the clients, the mesh

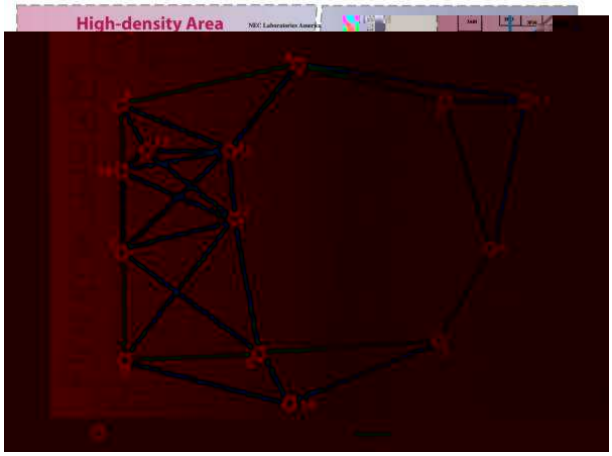


Figure 2.4: 15 node testbed in a 70m x 55m building

acts as a switch or hub in the sense that they are not concerned with the internal routing of the mesh. However, the implementation of the mesh is based on IP, even though it offers a layer 2 abstraction outside. In our implementation, the clients can have connections across the mesh to other wireless devices as shown in the figure, through the institution intranet to other wired VoIP phones, out to the internet with the help of a SIP server, or to the PSTN through a local PBX.

2.3.1 Hardware/software configuration

Our VoIP mesh testbed consists of 15 nodes based on the Stargate architecture from Intel, using the XScale processor, 32MB of RAM, and 64MB of compact flash. Each node is equipped with two 802.11b wireless interfaces (compact flash and USB 1.1) and has an open slot for a third one (PCMCIA 16bit). The testbed is spread over the third floor of NEC Research Labs in Princeton NJ, and the layout is shown in Figure 2.4. A high-density area on the left side of the building provides a proximity of nodes which allows the study of interference, and a lower density area to allow for longer paths in the network. The wireless cards are running at the fixed rate of 2Mbps which has the advantage of providing more stable results for the indoor setting. Each node operates with two interfaces: one that is used to get client traffic from the VoIP 802.11 phones, and the other one for backhaul in the mesh. If a third interface is available, it can be used to improve the capacity of the backhaul. To experiment with this setup using only two interfaces, we generated traffic locally at the nodes, in order to have both interfaces available for backhaul.

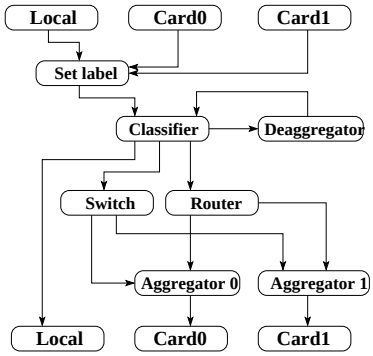


Figure 2.5: Click node components for label based forwarding, routing , aggregation

2.3.2 Mesh node

In order to provide routing, forwarding and other VoIP specific services, we used the Click modular router [44] on each mesh node. The router architecture is shown in figure 2.5. Voice packets use label based forwarding and routing, while other traffic uses regular routing. In this router configuration, when a packet is received from the cards, it may get labeled if it is a voice packet which needs to be routed over mesh network. For test purposes, labeling of traffic that is generated locally at the node is also allowed. To increase the capacity to carry VoIP traffic, we implemented a packet aggregation service which encapsulates multiple small VoIP packets into larger packets before forwarding. For each interface, a corresponding aggregator as shown in Figure 2.5 handles outgoing packets. Similarly, a de-aggregator decapsulates the aggregated packet into the original VoIP packets. A classifier decides whether a packet is destined for the local machine (signaling, aggregated packets); has to be routed (best effort traffic, signaling); or has to be label routed (voice). For aggregated packets, after the decapsulation, the resulting packets are fed back to the classifier and they may join another aggregation path.

2.3.3 Label based forwarding

In order to label the IP packets, we use TOS field of each IP packet that provides 255 labels at each node. Only packets with non zero label are forwarded based on their label. Packets with a zero label follow the underlying routing protocol (DSDV) used in the mesh network. In order to perform label based forwarding, each node maintains an additional table with an entry like:

in_label	out_label	interface	gateway

Any packet that arrives on the interface and has a non

zero label (*in_label*) is stamped with the corresponding *out_label* and sent to the *interface* to be delivered to *gateway* which is the next hop in the path. Once the outgoing label is set (by the Switch in Figure 2.5), and the outgoing interface is determined (for both routing and label based forwarding, by their respective lookups), packets are pushed to *"pull"* queues associated with each interface. These queues perform the aggregation of packets with have the same next hop (only voice and probe packets). The meaning of *"pull"* in Click terminology is that these queues are queried by the cards when the transmission is possible, so the waiting time in these queues is used for the purpose of the aggregation. A naïve implementation of aggregation would delay small packets in order to club them together in larger packets. Using this implementation feature ensures that no forced delay is introduced during forwarding.

2.3.4 VoIP call routing

Voice packets have a hard deadline of about 200ms mouth to ear in order to achieve reasonable quality, while 400ms is acceptable for intercontinental calls. This means that for the wireless leg of the setup, there is a fairly tight budget in time, so it can not be relied on the routing protocol to reconfigure paths during the call, or even search for the path when the call is placed. Our design decision was to use pinned down paths using label based forwarding to service the voice calls, where paths are continuously refreshed in the background by some routing protocol.

Up to five pre-computed paths are maintained for all voice communications of a pair of nodes, obtained from a route discovery protocol such as DSDV[45] to update the list of paths in the background. The use of pre-computed label based paths is appropriate for mobility as well. While one hop handoff can be achieved in 60ms in 802.11 networks [46], updating a path in a mesh, or using triangle routing may not satisfy delay requirements. For example, a loaded 4 hop path in our testbed has a delay of 80ms, and still provides QoS, but a signaling scheme or routing protocol might require several round trips along this path to set up a route. With pre-computed paths, after the one hop handoff, the new node has five proven good paths to choose from.

The main reason to use label based pre-computed paths is *to have several alternative paths between the same source destination pair, available at all times*, but there are other advantages to be considered as well: a) no time to update paths during calls or after handoff, b) flows may have to be switched fast to alternative paths to maintain QoS, c) fast call admission (no waiting for routing setup or reservations); d) pinned down paths allow various path selection strategies at source (e.g. aggregation, interference). Multipath routing also comes with generic advantages which are not specific to realtime traffic, providing natural load balancing, increased resilience, path diversity, increased capacity and reduced interference (when several cards are available).

To summarize, the implemented VoIP mesh system uses routing (DSDV) for signaling and best effort traffic, and label based forwarding for voice traffic. Each node maintains statistics about the popular routes used by DSDV and keeps the top five most popular routes to each destination to be used for label based forwarding. Mobility is handled using these precomputed paths so that after handoff from one node to another, several paths are always available to continue the voice forwarding.

2.4 VoIP performance optimizations

2.4.1 Evaluation methodology

In order to evaluate the performance of VoIP communication over this network, we used the **rude** UDP traffic generator/collector[47], which is able to generate CBR packet flows with given rates, packet sizes, and schedules. Detailed traces of the connection include the sending time, receiving time, flow ID, and sequence number for each packet. In order to have an accurate measure of the delay, all node are synchronized using NTP. To emulate voice conversation between two terminals, we set up two simultaneous rude sessions, one for each speaker. All calls have one minute in length, and use a traffic pattern corresponding to the G729 encoder, which produces 50 packets per second with 20 bytes of payload each.

2.4.2 Use of multiple interfaces

Referring back to Figure 2.1, we see that the main problem in a multihop network is performance degradation with increasing number of hops. A simple idea for improvement would be to just increase the number of interfaces in each node. A naïve use of multiple interfaces in a string would be to use one interface on a channel for the forward traffic and a second interface on a second channel for the reverse traffic, which should provide double capacity. We verified this in our testbed on a string of six hops.

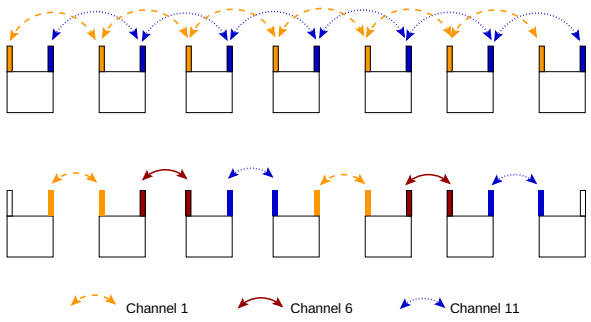


Figure 2.6: Each node with two 802.11b interface. case A: two non overlapping channels for forward and reverse direction; case B: three channels used with reduced self interference

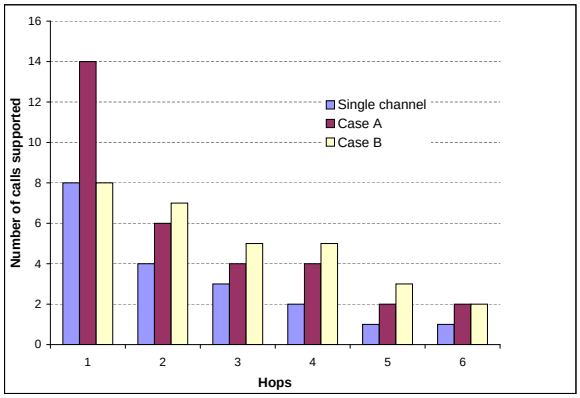


Figure 2.7: Channel diversity: use of multiple cards with independent channels.

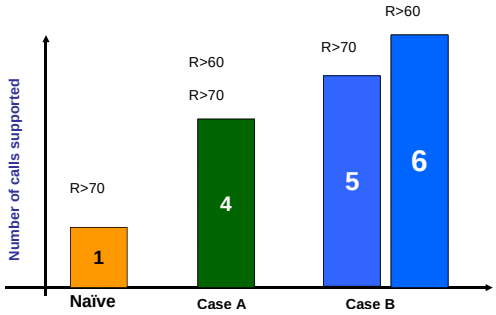


Figure 2.8: Path and channel diversity: 2 disjoint paths, each using independent channels.

However, for each of these flows, the same behavior as in Figure 2.1 is created by interference with neighbors which have cards on the same channel. An alternate method is to use more independent channels as shown in Figure 2.6. However, using 802.11b, only three channels are available, which limits the achievable improvement. Operating with only two backhaul interfaces and only three independent channels offered by 802.11b, we evaluated the following situations. Case A: two independent channels for forward and reverse traffic: (1,6)-(1,6)-(1,6)-(1,6)-(1,6)-(1,6)-(1,6)-(1,6). Case B: reduced self interference channel allocation: (1)-(1,6)-(6,11)-(11,1)-(1,6)-(6,11)-(11). The two solutions produce notable improvements, especially for longer paths (Figure 2.7). The lack of improvement for shorter paths is explained by a shortcoming of our testbed node, which only supports a limited number of interrupts per second. Using a better architecture, roughly a doubling of performance is expected with the addition of a second card, at least for the solution A. Solution B has even greater potential of improvement when more independent channels are available. In 802.11a, interference may be completely eliminated in a string, because a channel can be reused after 11 hops, which in most cases will be out of the interference range.

Further, one can use the multiple interfaces to create path diversity in addition to channel diversity. Here the forwarding path and the reverse path are disjoint, preferably at the interference level as well, except at the termination points. Also, each path independently uses channel diversity.

To evaluate the multiple path option in the testbed, we pinned down two independent paths of five hops each, which are as far from each other as possible: 10-9-8-7-11-6 and 6-5-14-3-2-10 (see Figure 2.4 for the placement) and assigned channels for the forward and reverse paths: 1-7-1-7-1 forward and 4-11-4-11-4 reverse (case A), 1-1-7-7-1 forward and 11-11-4-4-11 reverse (case B). Using configuration B, five calls were possible between nodes 6 and 10, compared with just one call in the basic case (Figure 2.8).

Using several network interfaces provides scalability to the system, while using several channels across the mesh provides frequency diversity. These factors combined actually have a more than additive effect, meaning that the capacity improvement of using them together is greater than the sum of their independent improvements. The reason is the reduction of interference, but this gain is still limited by the number of interfaces used - 2, and the number of available independent channels - 3.

Currently, we are investigating the interference properties of 802.11a cards which offer better possibilities for denser meshes. On one hand, the range is shorter and the capacity higher, and on the other hand a larger number of channels is available, so a six hop setup like the one in our building should require no frequency reuse.

2.4.3 Routing

The design of good routing schemes for supporting real-time applications over wireless mesh is an inherently challenging problem. The difficulty in routing arises from

the number of factors on which a good route depends: a) channel quality; b) dynamic condition due to interference caused by traffic inside and outside the mesh network; c) traffic load on routes in the interference range. Jointly considering the routing and channel assignment problem [40] is NP-hard even with a centralized solution and instantaneous global knowledge of network conditions. Voice calls pose additional problems because any decision taken in reaction to network conditions may affect voice quality: changes in routes, call admission and handoff, all have strict delay requirements to minimize the time during which packets are lost.

In our current setup, call admission has to be performed within seconds of placing a voice call. This is achieved by using pre-computed paths, even if the solution is suboptimal. Another important factor in this decision is that the call admission process should preferably be distributed. To achieve the above design goals, our voice call routing approach consist of two components: route discovery and adaptive path selection. To choose paths, we opted for a solution based on probing in order to cumulate all factors (interference, load, channel quality), which are otherwise hard to account for.

Route discovery: For route discovery, DSR would have been a good choice, but we need to maintain multiple source destination paths, and the implementation available with Click performs poorly on our platform in terms of CPU usage and responsiveness. DSDV is the second option, but being distance vector based, it has the undesirable effect of frequently updating paths in the middle of the call. We experimented with several metrics: hop count, end to end loss, quantized end to end loss, a threshold based loss metric, and ETX [48] (which is also based on loss). All these metrics provide unacceptable performance (Figure 2.9) for voice in our testbed. This is mainly because of the default behavior of DSDV which is aging routes and is always ready to accept new ones. The wireless environment in the building is also a factor that degrades the performance of routing. Even the fairly stable hop count metric exhibits a lot of route variance because occasionally packets may travel across the entire building long enough to make the routing algorithm believe a one hop route is available. When this long link becomes unavailable, or is beyond a certain age, DSDV is looking for alternatives based on its current metric. It is in this case, during switching between routes, when voice packets get lost and quality gets degraded.

We opted instead for using DSDV to collect frequently used routes which are then pinned down and used with

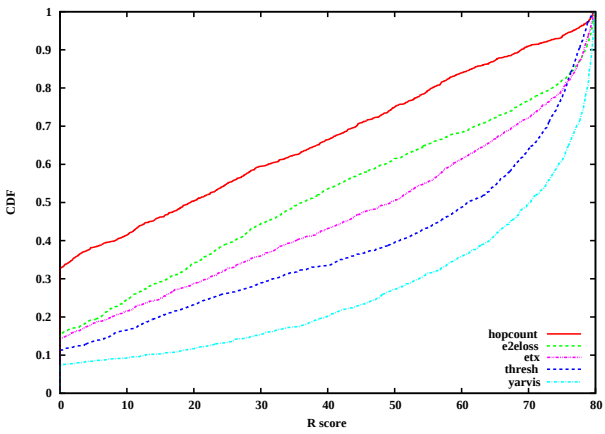


Figure 2.9: DSDV has low performance for one call across the testbed using various metrics.

label based forwarding. Each node is maintaining the top five most frequently used routes for each destination, based on measured statistics. We ran DSDV with various loss based metrics (including ETX, end to end loss, quantized end to end loss). We retained the five most frequently used paths chosen during last 24 hours. These top five paths were almost the same for the different metrics, although with differing frequencies from one metric to another. This means that from the loss point of view, a similar set of path shows a consistently better quality over time.

Adaptive path selection: In order to use the paths for voice transport over our wireless mesh network, we pinned down the paths using label based forwarding as described in the last section. We chose a pair of nodes A and B at extremities of the building to maximize distance in hops and path diversity. To make use of the alternate paths and the possibility of fast switching without losing any packets we implemented a simple strategy in which one of the nodes monitors all the paths with a low bandwidth ping (one packet per second). When a call is placed, the five paths are probed with a low bandwidth probe to evaluate the delay of each path, which is the most critical component. The probing traffic has the same characteristics and treatment as the voice traffic, namely it can be aggregated or delayed based on localized conditions in the network. The size of the packet is chosen to be the same as the voice packet so that round trip times are good estimates that can be extrapolated for voice packets. When the exponentially averaged *R-score* of the voice call stays under 70 for an extended period, the decision is taken to switch to another path based on the monitored round trip times and loss rates.

Evaluation: In order to evaluate the above strategy on

Path	R_{avg}	cdf($R > 70$)	path usage	R_{avg} used
adaptive	71.2	0.86	-	-
a	40.4	0.48	47%	72.4
b	56.3	0.69	40%	73.2
c	17.1	0.19	11%	70.8
d	28.7	0.33	1%	5.1
e	6.5	0.07	1%	52.8

Table 2.1: Adaptive path switching vs. fixed paths

our mesh testbed, a single voice call is ran for 2500s between nodes A and B, and the *R-score* is recorded every second. Each of the five available paths is measured independently. To create a repeatable pattern of disruption similar to office use of laptops, we selected a number of jammers outside the testbed that follow a predefined random, but fixed, sequence of traffic on the same channel as the testbed.

In Table 2.1, we have the path labels in the first column, from a to e. The first line corresponds to the adaptive strategy. The second column of the table shows the average *R-score* achieved, and the third column the fraction of time when $R > 70$. By using the available alternate paths, the simple adaptive strategy is able to route the voice traffic around the interference and congestion providing a good *R-score* 86% of the time, with an overall average of $R = 71.2$. The fourth column shows how paths are used by the adaptive strategy, and the fifth column the average *R-score* obtained by the respective paths on behalf of the adaptive strategy. Most service is provided by just three paths, which could help in reducing the amount of probing traffic to only proven quality paths. Probing of additional paths may be enabled only when the reduced set doesn't provide the required quality.

Knowing that the *R-score* is a function of loss and delay, the question is which of the two factors is more important in our testbed. The network loss is less than half a percent for most paths, except one, so delay must be the deciding factor. In Figures 2.10a and 2.10b, the delay distribution histogram is shown for one of the participating paths and for the adaptive case. Times over 200ms are collapsed in the rightmost bin. These distributions confirm that the quality of our paths is dominated by delay. The source of this delay is cross traffic (from the jammers) and channel conditions (802.11 retry is set to 16). Since an unloaded path experiences about 2ms-3ms per hop, and our paths have 4 and 5 hops, the measured delay was confirmed to be in the range 8-15 ms, which is

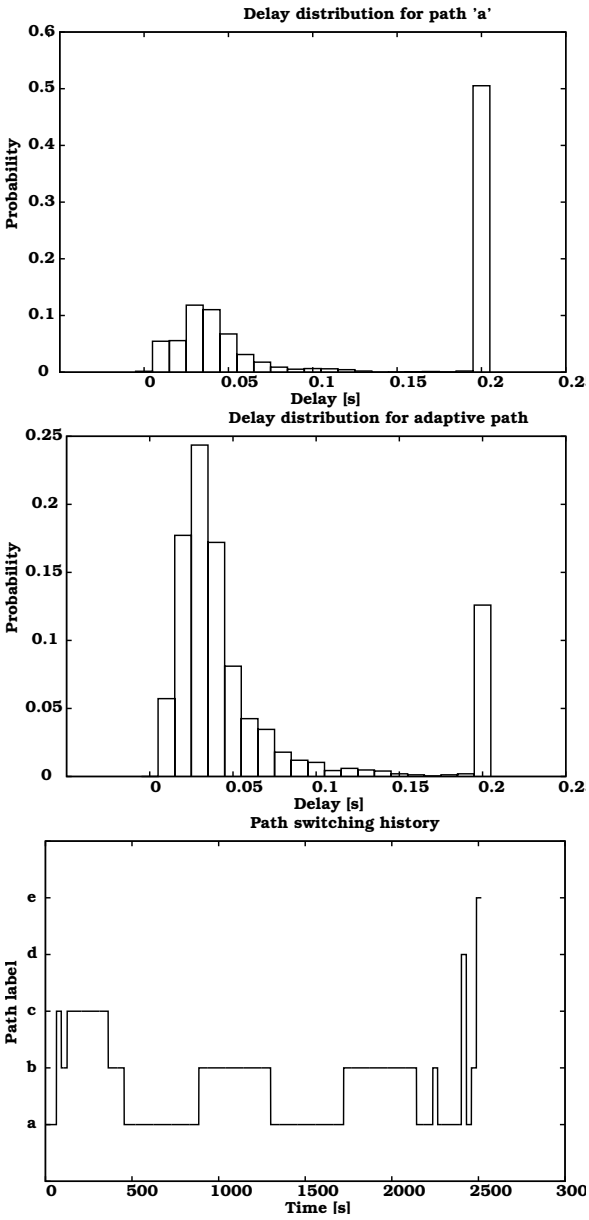


Figure 2.10: Delay distribution adaptive vs. fixed. (a) A fixed path provide delays greater than 200ms 50% of the time. (b) Only 12% of the time the delay is greater than 200ms when the path is adaptive. (c) path labels used by the adaptive scheme.

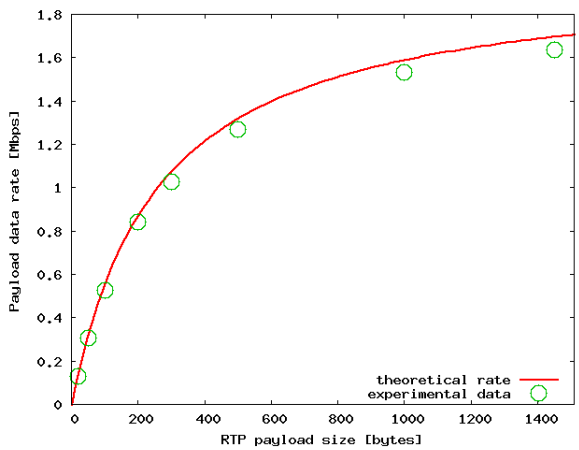


Figure 2.11: Overhead measurement confirms analysis in a 2Mbps 802.11 network.

almost negligible compared with the delays experienced by the voice traffic during the experiment. Figure 2.10c shows which paths were used during the experiment, corroborating the figures from table 2.1.

2.4.4 Aggregation

As most vocoders use samples of 10-100 ms, a mesh node is expected to get a large volume of small packet traffic. However, 802.11 networks incur a high overhead to transfer one packet, therefore small sizes of packets reduce the network utilization. The problem with small payloads is that most of the time spent by the 802.11 MAC is for sending headers and acknowledgments, waiting for separation DIFS and SIFS, and contending for the medium. For example, in order to send a 20 byte VoIP payload, a 60 byte packet is assembled from 20 bytes IP header, 12 bytes RTP header, and 8 bytes UDP header. This takes 43.6μs to send at 11Mbps, but MAC header and physical headers, trailers, inter-frame periods and ACK need a total of 444μs. That however does not consider the amount of contention which is on average of 310μs, and increases exponentially with contention. This way, to send a 20 bytes payload takes 800μs at 11Mbps, yielding approximately 1250 packets per second, which for a vocoder like G.729a means only 12 calls can be supported. At 2Mbps, a similar computation leads to 8 calls. When sending x byte voice samples, the overhead incurred is given by:

- RTP/UDP/IP 12+8+20=40 bytes
- MAC header + ACK = 38 bytes
- MAC/PHY procedure overhead = 754μs
  - DIFS(50μs), SIFS(10μs)

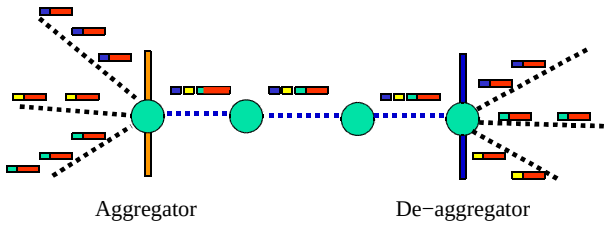


Figure 2.12: Aggregation merges small voice packets from different calls into larger packets to improve channel utilization

  - preamble + PLCP (192μs) for data and ACK
  - contention (approx 310μs)

The throughput in Mbps is given by the relation

$$T(x) = \frac{8x}{754 + (78 + x)\frac{8}{B}}$$

where x is the payload size in bytes, and B is the raw bandwidth of the channel (in Mbps - 1,2,5.5, or 11). In Figure 2.11, we measured actual data rate obtained between two nodes as a function of packet size employed, which proved to be fairly closed to the analysis of $T(x)$. When using 20 byte voice payload, and comparing with the maximum achievable with large packets, the obtained throughput is just 8.2% for 2Mbps, and 3% for 11Mbps.

Two main techniques of reducing this overhead are packet aggregation and header compression. The basic idea of aggregation (Figure 2.12) is to combine together several small packets at the aggregator in the ingress nodes and forward them with one IP, MAC and PHY header across the air.

A common problem in packet aggregation is that it increases packet delay, reducing its suitability for VoIP services. But if the network is lightly loaded, the packet aggregation techniques do not have to be used for improving network performance. Under heavy load, small size packets would experience heavy contention which lead to retransmission, and drops. The packets then spend the largest part of network delay in the queues at the intermediate nodes. The higher the contention to access wireless media, the larger the network delay becomes. These small packets waiting for media access in the queues are the candidates for packet aggregation. Therefore, packet aggregation during heavy load does not need to introduce additional forced delay to combine packets.

Packet aggregation can also be used for combining voice packets in the same flow by introducing an explicit delay at the ingress node of the call, if the extra la-

Algorithm 2.1 Aggregation logic for ingress nodes

```
P - packet being queued at a node;
P' - packet with the same next hop as P;
A - aggregation packet being prepared;
minPackets - number of packets from the same flow
  that have to be aggregated at the ingress (corresponds
  to the delay budget available for the flow);
MTU - maximum transmission unit = number of voice
  packets that can be fit in 1500 bytes;

   find queue of P;
1: if size(queue) > minPackets
     add all packets from flow(P);
     if size(A) < MTU
       find a queue with the same dest
       go to 1;
     else
       send A directly to destination;
  else
  if size(A) < MTU
    do find the flows(P') >= minPackets and add
minPackets from them
      while size(A) > MTU
  else
    send A to the next hop
```

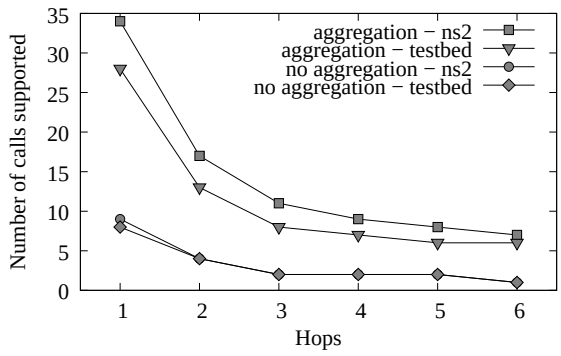


Figure 2.13: Aggregation on a string: ns-2 vs. testbed.

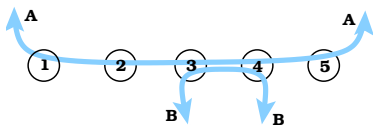


Figure 2.14: Aggregation introduces only controlled delay at the source of flows. Intermediate nodes do not delay packets to improve aggregation, but use "natural" waiting required by MAC under load.

tency does not degrade voice quality severely. This technique increases mouth to ear latency, but it also reduces queueing delay caused by network contention. On the other hand, too large forced waiting at the ingress node causes longer network delay, and lower quality. In our system, the delay budget available for aggregation is obtained from path monitoring. If measured delay on a path allows extra waiting without degrading the *R-score*, that amount is allocated to aggregator at the ingress node.

The aggregator treats differently flows which are forwarded and the ones for which it is the ingress node. At the ingress the packets can be delayed some amount of time, depending on the budget allowed by the probing of available paths (Algorithm 2.1). When forwarding however, no delay is introduced, but under higher than minimal load, packets still cumulate in the queues, waiting for medium access. During this wait, other packets may join the aggregated packet, provided that they have the same next hop or same destination. Alternatively, packets which take a different path are split and re - aggregated accordingly.

Evaluation: In *ns-2*, we simulated a string of 6 nodes with and without aggregation, and verified the results against a similar string in the testbed. In Figure 2.13, we find that for the non aggregated traffic, the simulation matches the testbed results in most points, but for aggregated traffic the testbed performs worse. The cause of this was identified in the fact that the capacity of some of the hops in 2Mbps mode was less than the optimal 1.7Mbps. The first hop for example was measured to provide a capacity of only 1.38Mbps, with 1500 bytes packets, accounting for the difference between the aggregated calls supported - 29 in testbed versus 34 in *ns-2*. However, knowing that our simulation setup performs reasonably close to the testbed, we obtain the most of the next results in simulation only.

One of the main claims of our distributed aggregation method is that it does not introduce additional delay by using the wait for the MAC availability to club together packets destined to the same next hop. To verify this claim, we place five longer calls indicated by A in Figure 2.14 and five short calls, indicated by B. Aggregation is performed for each group of flows independently at the source by introducing a controlled delay of 80ms. This means that packets from A are not merged with packets from B during their common hop 3-4. The network time for flows A is about 61ms in the absence of B and increases to 89ms after B is added. This is normal, considering the increased interference for all the nodes and the longer queues at nodes 3 and 4. After we enable aggregation between A and B at hop 3-4 the network time for flows A decreases to 79ms. Not only a delay is not added to the long flow, but the creation of larger packets reduces the contention for the hop 3-4 thus reducing the load on the network. This experiment reveals some interesting properties of our aggregation scheme: first, it only kicks in at higher load when waiting in the queues can be

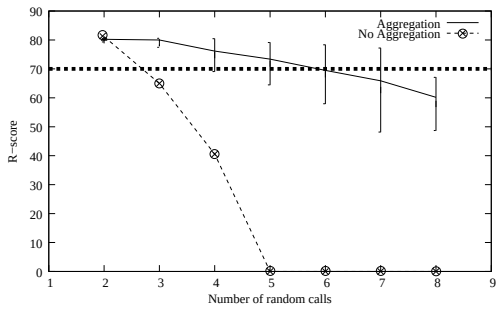


Figure 2.15: Aggregation performance for random calls in a string

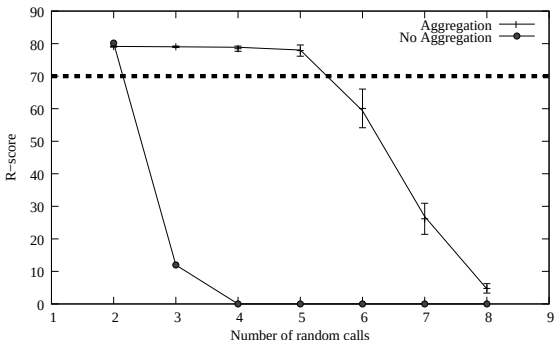


Figure 2.16: Aggregation performance for random calls in a tree

used to group packets with the same next hop. Second, it is completely distributed inside the mesh. The endpoints need to specify initial delays depending on their time budgets, but the intermediate nodes have the simpler task of looking for packets with the same next hop. Third, and the most important, **short flows do not delay long flows for the purpose of aggregation**. It is true that the mere existence of short flows increases wireless medium load and therefore increases delay for long flows, but that is inherent to the behavior of the shared medium. In network distributed aggregation reduces load without impacting the network time of existing flows.

In another experiment we consider a more randomized situation in terms of sources and destinations of calls. In an eight hop string, calls between random sources and random destinations are placed. Considering 10 random situations for each configuration of two to eight calls, we compute the minimum, maximum, and average value of the *R-score* for the offered number of calls. In Figure 2.15 these values are plotted for each offered load. For the non-aggregation case minimum and maximum only take values 0 or maximum, and are omitted from the figure. If we consider an *R-score* of 70 as a threshold of call quality, then aggregation more than doubles the capacity, even with randomized traffic.

When the mesh is used as an extension of the access point infrastructure, a popular pattern is to have voice calls forwarded to the wired infrastructure or to the PSTN. In this case paths from the clients all lead to a common root which provides access to the wired leg of VoIP. We simulated a complete binary tree with 8 leaves, and with an additional link from the root to the wired access, so that there are 4 hops to forward from the leaves. Figure 2.16 shows that performance improvement is similar to the string case, by more than a factor of two. For the non aggregated case however, the capacity is much less, mainly because of the interference that now covers larger portions of the tree (more than the 4 hops that interfere in a string).

2.4.5 Aggregation and header compression

Header compression is a complementary scheme related to aggregation. It has the same goal of reducing the amount of overhead by exploiting headers that do not change, or whose change can be predicted. For a VoIP flow RTP/UDP/IP headers take 40 bytes, but only 12 of them are changing often. Schemes such as cRTP or ROHC aim at compressing the 40 bytes into a 2 byte connection ID, but they are appropriate for one link only. In order to emulate a simpler scheme that only transmits the changing fields, we reduce the header from 40 to 14 bytes so that more voice packets can fit in an 1500 byte packet.

In Figure 2.17, we look at the performance gain given by header compression alone, aggregation alone and the combination of header compression and aggregation. Header compression by itself achieves an almost negligible improvement, because it only reduces overhead with 26 bytes out of the total 78 bytes + 754μs. Aggregation only curve is the same as in Figure 2.13, repeated here for reference. When combining header compression and aggregation we get another factor of two improvement in capacity. The reason is that once the aggregation reduces the MAC overhead of 78 bytes + 754μs, the saving of 26 bytes provided by header compression becomes significant by allowing more voice samples be stored in a 1500 byte wireless packet - 41 for header compression versus 24 for aggregation only.

2.4.6 Aggregation and multiple interfaces

To combine the advantages of using several interfaces and aggregation, we simulated in *ns-2* a string of 6 nodes that use one or two channels for the backhaul traffic. The improvement is most visible at 6 hops (Figure 2.18): a factor

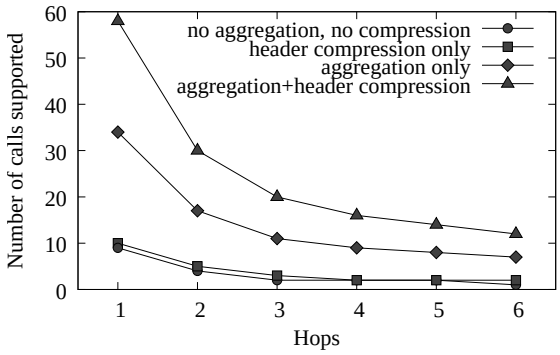


Figure 2.17: Header compression increases capacity over simple aggregation.

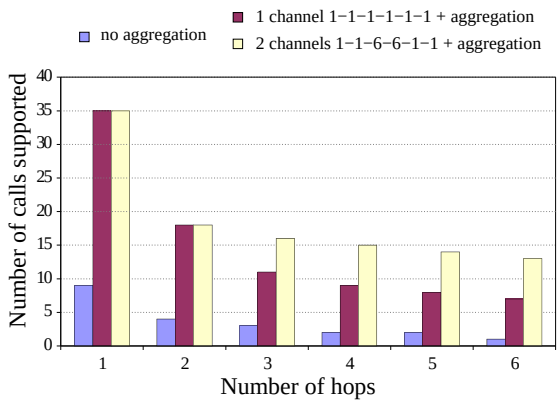


Figure 2.18: To send a 20 byte packet over 802.11, 78 bytes are used by MAC, IP, UDP and RTP headers. Aggregating voice packets from different flows provides 13 times improvement for 6 hop calls.

of 7 increase from the aggregation, and another factor of 2 from the multiple channel.

The combined results of the testbed experiments and the *ns-2* simulations show that with appropriate optimizations, the wireless mesh is appropriate for sending voice. Path adaptation enabled by label based forwarding is improving the QoS, while channel diversity, path diversity and aggregation improve capacity.

2.5 Conclusions

We experimentally investigated several methods to improve the quality of VoIP over a WLAN mesh. These are the use of multiple interfaces, label based forwarding architecture, and packet aggregation. Each of these methods produces considerable improvement in the operation of the mesh - with respect to capacity, QoS, or both. After evaluating several design options we believe that a label based solution is the most appropriate for carrying realtime traffic in a wireless mesh operating in the unlicensed spectrum. Our architecture combines routing and label based forwarding, and addresses all aspects required to support VoIP over the WLAN mesh: call admission, mobility, QoS. We implemented a distributed packet aggregation strategy that is work conserving by using MAC waiting to perform aggregation, without introducing unbounded packet delays. These performance optimizations are implemented in a 15 node wireless mesh network, and the experimental results show an increase of 13 times for a 6 hop string when all optimizations are used.

Chapter 3

TCP and VoIP in Wireless Meshes

3.1 Problem Statement

Most traffic that flows over the Internet makes use of the Transmission Control Protocol (TCP) and wireless multihop networks are one way to provide access extension. TCP is one of the protocols designed for wired networks and exhibits severe degradation in multihop networks. It was designed to provide reliable end-to-end delivery of data over unreliable networks and has been carefully optimized in the context of wired networks. For example, large TCP default window sizes that are appropriate for a wired network are too large for wireless links in multihop networks.

Another type of traffic that becomes more prevalent in homes and institutions is VoIP. This capability becomes available in most new cell-phones as well, due to convenience and cost savings. VoIP, however, is different from most other traffic in that it has quite stringent delivery requirements. While mechanisms to provide for this QoS exist in the wired networks, in the popular 802.11 based networks they were only an afterthought.

In this paper we show that coexistence between these two popular traffic is a difficult one in multihop networks, and investigate the different methods that can be used to facilitate it. Even if QoS enhancements such as 802.11e were added, it doesn't really address the central problem of multihop networks, which is interference, the main factor affecting the coexistence.

The interaction between TCP and VoIP over a multihop network is very complex. We summarize the most important points:

- *TCP is an end to end protocol.* There are no explicit signaling mechanisms in the network to tell the TCP peers how fast to send, how much to send, or when to slow down a transmission. A peer is responsible for controlling these parameters from implicit knowledge it obtains from the network or from explicit knowledge it receives from the other peer. TCP needs to be aggressive in discovering link bandwidth because this way it can achieve high utilization. This is achieved using large windows, which aggravates channel contention on wireless links.

- *TCP produces bursty traffic, while VoIP is uniform.* In the so called 'slow' start phase, TCP doubles its window for each ACK received - in reality an exponential increases in bandwidth consumption. This creates trains of packets that hog the medium for prolonged times. VoIP on the other hand needs regularity in the network delay and a low loss rate. When the network is congested by interference or too much TCP data, VoIP traffic suffers from increased network losses and delays. However, TCP just goes into the recovery stage, reducing its sending rate until the network is recovered from congestion, and then sends all postponed packets. This cycle of burstiness leads to both low utilization for TCP and unacceptable quality for voice.

- *TCP assumes that losses come from congestion.* This observation has been the basis of many studies and proposed modifications focusing on preventing TCP congestion control mechanism to react to link layer errors. Many performance studies of the TCP protocol over 802.11-based multihop show standard TCP behavior may lead to poor performance because of packet drops due to hidden terminal induced problems such as channel interference and TCP data/ACK contention.

- *VoIP packets are small, while TCP packets are large.* For a given bit error rate, TCP packets will have less success, so many of them would be retransmitted across multihop links, thus generating even more load that in turn generates more interference.

Accurate and timely estimation of the available bandwidth is very important. Although there are many tools

to estimate this parameter over multiple hops, most prior work has largely focused on improving TCP performance over multihop networks, and was not concerned with the coexistence of TCP with real-time applications such as VoIP. VoIP is mostly constant bit rate, has very tight delay and loss requirements, and should always be served prior to TCP traffic. Classical solutions such as priority queues, bandwidth limitation, and traffic shaping do not provide satisfactory solutions for the coexistence problem. Even if voice traffic has priority locally within a node, bursty TCP traffic affects voice packets on other nodes within the interference range.

This paper investigates the behavior of TCP and VoIP flows in a shared network and proposes a novel bandwidth control technique to enable this coexistence in interference-ridden conditions such as multihop networks and WLANs. We examine ways in which TCP and VoIP can coexist while **satisfying two contradicting goals**: maintenance of VoIP quality, but without sacrificing TCP performance and network utilization. We found that merely limiting bandwidth of TCP, while necessary, is not sufficient, as the bursty behavior of TCP results in poor utilization. Another finding is that it is beneficial to make TCP look more CBR like for two reasons: it is more predictable and therefore VoIP friendly, but also has hidden benefits for TCP itself.

The problem is not simply a matter of bandwidth estimation, even if VoIP takes a predictable share of the resources, because it matters not only how much TCP to allow in the network, but also what shape as well. We propose *Voice Adaptive Gateway Pacer (VAGP)* - a method that uses existing VoIP traffic that shapes incoming TCP to protect both VoIP and utilization. This estimation is a continuous process that has to adapt to a changing environment: wireless channel conditions, VoIP load, TCP flow arrivals, internet delays - all may change on time scales of seconds. *VAGP* limits TCP share within milliseconds of noticing reduced capacity, but still allows aggressive discovery when new bandwidth becomes suddenly available, all while protecting voice traffic.

3.2 Existing work

A large amount of research has focused on the optimization of the TCP performance over wireless networks - single and multihop.

Studies have been conducted to understand the capacity of multihop network and to improve VoIP capacity over the wireless mesh networks, as presented in [49], [50]. Several performance optimization schemes have been proposed to improve the VoIP quality over a WLAN: [51] proposed the use of a dedicated queue to provide higher priority to VoIP traffic over data traffic, while in [52], packet aggregation is used to increase capacity.

Many bandwidth estimation techniques have been proposed for wired networks. In [53], the authors survey different methods to evaluate the capacity, the available bandwidth and the bulk-transfer capacity.

Ultimately, the question of knowing how much TCP to allow in the network is one of bandwidth estimation, as VoIP takes a predictable share of the resources. In 802.11-based ad hoc networks, few works deal with solutions for bandwidth estimation. in [54], each mobile estimates the available bandwidth by computing the channel utilization ratio and using a smoothing constant. The channel utilization ratio is deduced from a permanent monitoring of the channel states. The main idea presented in [55] is that a probe packet delay higher than the maximum theoretical delay can characterize the channel utilization. The authors propose a method to compute the medium utilization from the delays and then derive the available bandwidth from the channel utilization.

In wired environments, TCP is considered to be too slow in capturing the available bandwidth of high-performance networks, especially over long-distance paths. Two issues commonly identified as the underlying reasons are:

1. Limited buffers at the TCP sender or receiver impose a conservative upper bound on the effective window of the transfer, and thus on the maximum achievable throughput.

2. Packet losses cause a large and multiplicative window reduction, and a subsequent slow (linear) window increase rate, causing an overall low average throughput.

Some TCP-related issues that often impede performance are: multiple packet losses at the end of slow-start (commonly resulting in timeouts), the inability to distinguish between congestion and medium packet losses, the use of small segments, the coarse granularity of the retransmission timeout, or the initial value of the *ssthresh* parameter. Networking research has focused on these problems, pursuing mostly modified congestion control algorithms. From the wireless perspective, the problem that arises in the usage of TCP over wireless networks comes from the

fact that wireless links have different characteristics with respect to wired ones, namely lower reliability and time-variant behavior, node mobility, hand-offs, limited available bandwidth and large RTTs. Since the only reaction provided by TCP in the event of unsuccessful packet delivery is the congestion control mechanism, TCP implementations perform poorly in wireless environments. The majority of the solutions proposed by the research community fall in three main categories:

Connection splitting solutions: The key problem for TCP over hybrid wireless/wired networks lies in the different characteristics of wireless networks and wired Internet. While most of packet losses experienced in wireless networks are due to hidden terminals and channel contention at the intermediate nodes, drops in the Internet almost always are due to buffer overflows at the routers. A solution to this network convergence problem [56] lies in splitting the TCP connection at the node interfacing the wired and wireless part of the network, denoted as the Internet gateway. Connection splitting can hide the wireless link entirely by terminating the TCP connection prior to the wireless link at the base station or access point. With this approach, the communication in wireless network can be optimized independently of the TCP applications. However it requires extra overhead to maintain two connections for one TCP communication. It also violates end-to-end semantics and complicates the handover process.

Link layer solutions: These try to make the wireless link layer look similar to a wired layer from the perspective of TCP. The most relevant and interesting proposal is the snoop protocol [57]. A snoop agent is introduced at the base station to perform local retransmissions using information sniffed from the TCP traffic passing through the base station. Another link layer solution proposes QoS scheduling with priority queues in the access point (AP) [51] to improve VoIP quality by placing TCP data in a lower QoS level.

Gateway solutions: One way to address TCP performance problems within wireless networks is to evenly space, or pace data sent into the multihop over an entire round-trip time, so that data is not sent in a burst. Pacing [58] [59] can be implemented using a data and/or ACK pacing mechanism. TCP-GAP [58] suggested congestion control scheme to reduce burst of TCP packets based on estimating 4-hop propagation delay and variance of recent RTTs at the Internet gateway for wired-wireless hybrid networks. TCP-GAP scheme is relatively responsive, provides fairness among multiple TCP flows and better goodput than TCP-NewReno. However, it still depends on network topology, and fails to estimate TCP bandwidth correctly in the presence of real-time traffic, such as VoIP. Congestion control is operated by the general TCP scheme, which is too aggressive for wireless multihops.

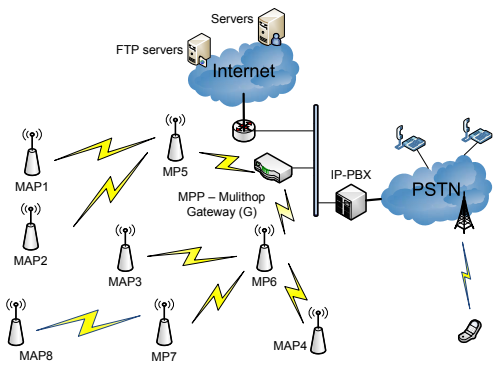


Figure 3.1: Wireless multihop topology: a multihop gateway connects to the wired Internet to deliver TCP traffic, or to PSTN via IP-PBX for VoIP calls. A Multihop Point (MP) just forwards traffic, whereas a Multihop Access Point (MAP) also allows stations (STA) to associate with it. In this paper we consider downlink TCP traffic originating on the Internet, destined to a client associated with a MAP.

3.3 TCP and VoIP: Difficult coexistence

It is well understood from queueing theory that bursty traffic produces higher queueing delays, more packet losses, and lower goodput. It has been observed that TCP's congestion control mechanisms and self-clocking create extremely bursty traffic in networks with large bandwidth-delay products and causes long queues and the likelihood of massive losses. In addition, wireless multihop network traffic tends to have a self-similar behavior [60] making it difficult to provide stable rates necessary for VoIP.

Fig. 3.1 shows the wired/wireless hybrid networks we consider in this paper. A multihop extension carries traffic from wired Internet, or PSTN (through IP-PBX). The multihop leg is where VoIP needs to be protected from TCP.

3.3.1 Bursty traffic

To understand the difficulties in supporting VoIP, we start with a short primer on VoIP quality requirements. VoIP traffic is CBR, and for certain vocoders (G711, G729), its quality can be estimated using packet loss and mouth-

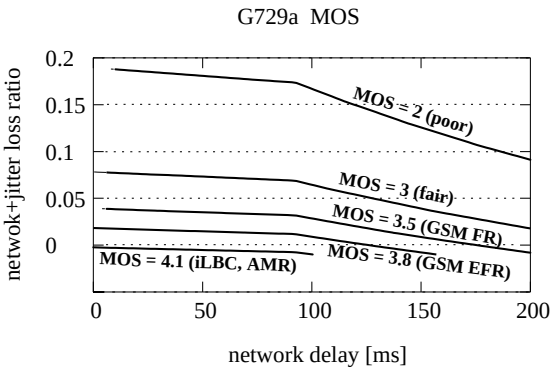


Figure 3.2: For certain vocoders, such as G729a, VoIP quality (*MOS-score*) can be computed as a function of loss and one way delay. Loss include packets lost in the network, and packets which miss their deadline because of jitter.

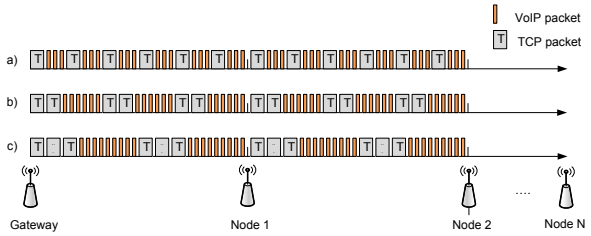


Figure 3.3: VoIP statistics and data goodput as the burstiness increases; 5 VoIP calls and 550Kbps of data offered; 4-hops string topology, 12Mbps, 802.11a.

to-ear (one way) delay. Fig. 3.2 shows the values of *MOS-score* with respect to network delay and total loss for 60ms playout-buffer and 25ms vocoder delay. For example, in order to obtain *MOS-score* of 3.6 (comparable to GSM quality), the network has to deliver all packets in less than 160ms, or deliver 98% in less than 104ms (for more details, see [61]). For G.729a used in the rest of the paper, 3.9 is the maximum quality achievable, but we consider 3.6 to be "acceptable quality".

First, we show the burstiness is the main cause of reduced VoIP quality. To this end, we set up a string with four wireless nodes, equivalent with the sequence MPP - MP6 - MP7 - MAP8 - Client in Figure 3.1. All these hops operate on the same channel. In this setup we experiment with various packet patterns as shown in Fig. 3.3. In these scenarios we have the same mean offered rate for data (550Kbps), but with different burst lengths. The rest of the capacity is filled by VoIP packets.

The results corresponding to different burst lengths are shown in table in Table 3.1. Virtually all quality indicators for both VoIP (loss, one-way delay) and data (goodput, one-way delay) suffer because of the increased burst length. In fact, we can support 5 voice calls with 1 data packet bursts, but only 3 with 5 data packet bursts. In Internet scenarios, when long delays can be present on the Internet portion, even one TCP flow is expected to require windows much larger than 5 packets, and therefore produce even more degradation for itself and for VoIP.

burst length	VoIP calls	VoIP Loss (%)	VoIP delay (ms)	Data Goodput (Kbps)	Data delay (ms)
1	5	0.92	13	509	15
2	5	1.26	16	501	19
3	4	1.44	17	495	23
4	4	1.64	19	488	26
5	3	2.06	21	476	31

Table 3.1: VoIP statistics and data goodput as the burstiness increases; 5 VoIP calls and 550Kbps of data offered; 4-hops string topology, 12Mbps, 802.11a.

Table 3.2: Max number of VoIP calls and TCP throughput (Mbps) as the hop count changes, string topology, one channel, 12Mbps, 802.11a

Hops	No. of VoIP	TCP [Mbps]	Reason for degradation
1	45	9.5	Contention
2	23	4.9	Contention
3	16	3.2	Contention, Interference
4	12	2.5	Contention, Interference
5	9	2.0	Contention, Interference

In the same topology of four hops we try to establish what kind of performance we can expect from each type of traffic, and in combination. Running each of the four hops at 12Mbps, we can either support 11 VoIP calls, or 1.35Mbps of TCP, as illustrated in Table 3.2.

However, if we mix 5 VoIP calls and 3 TCP flows, we found that voice quality is below the minimum acceptable ($MOS < 2$) while TCP flows get a cumulative goodput[1] of 615Kbps using a throughput[2] of 903Kbps. **This shows that simply sharing the network fails to protect the VoIP traffic, and also yields lower utilization.** TCP uses a sliding window-based protocol which determines the number of packets that can be sent, and uses the receipt of acknowledgments to trigger the sending of packets. The window used by a TCP sender is chosen based on its view of the congestion in the network and based on the receiver's acceptable number of bytes. If the window size is too large, then the sender is allowed to inject more traffic than the network can handle. Given a wireless multihop network, there exists a TCP window size W^* at which TCP's bandwidth consumption is appropriate. This

[1] goodput = total bytes delivered by TCP receiver to the application layer.

[2] throughput = total bytes sent by the TCP sender.

number depends on many conditions, including presence of real time traffic, but the main point is that default TCP algorithms are not able to discover this W^*. The current TCP protocols do not operate around W^* but instead typically grow their average window much larger. This results in VoIP degradation, or in low TCP performance if VoIP traffic is not present [62].

3.3.2 Behavior of mixed traffic

Continuing with the four hop scenario, we then mix these two types of traffic: five voice calls are started at the beginning of the experiment, and after 10 seconds, three TCP flows are started in parallel with the voice. Fig. 3.4 shows how VoIP quality is degraded as TCP window size increases. VoIP quality is shown in the low-right of Fig 3.4, which starts at *MOS-score* of around 3.9 and decreases to *MOS-score* of around 2.2 as soon as the TCP flows with default window size 32 are introduced. TCP flows initially go through the "slow-start" phase increasing the number of packets in transit exponentially, which means every ACK packet triggers twice as much data as it acknowledges. After reaching the congestion threshold, TCP switches to linear rate increase to maintain high goodput without causing congestion in the congestion avoidance phase. These two alternating phases produce the well known saw-tooth pattern shown in the upper left of Fig. 3.4. In the lower-left of Fig. 3.4, we see an additional disadvantage that is wireless specific: the self-interfering nature of the wireless multihop, coupled with well known 802.11 unfairness issues [63] causes severe unfairness between the TCP flows. In the top right of Fig. 3.4, we trace the delays experienced by VoIP packets of one of the flows during the same experiment: due to TCP packet bursts and large packet size, network delay jumps from 10ms to about 250ms mostly due to extra wait in the queues. This is the main driver of reduced voice quality, as large queues and 802.11 retries largely cover most network losses in this example.

3.4 Candidate Solutions

It is clear from the previous section that VoIP and TCP cannot simply share a multihop network without experiencing severe reduction in TCP capacity and voice quality degradation. We first consider the various enhancements to TCP proposed by the research community in the recent years, namely: Reno, Vegas, Westwood, CUBIC, and Compound TCP (C-TCP). Then, we look at external con-

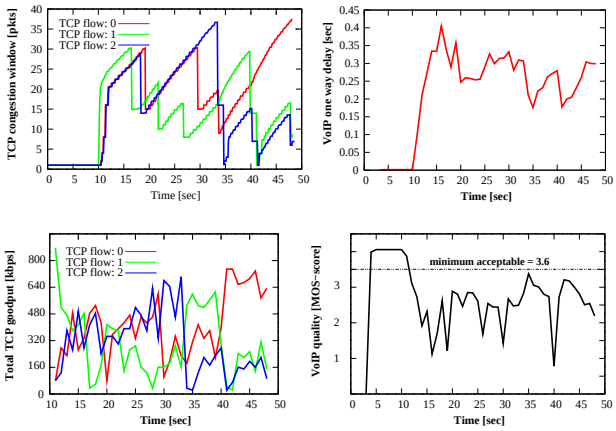


Figure 3.4: Uncontrolled TCP has many drawbacks: built-in backoff mechanism of TCP reacts too late to protect VoIP; Increased one-way network delay for VoIP; unfairness between TCP flows; low total utilization.

trol methods, that would leave TCP unchanged, but would police or instrument traffic at the multihop gateway.

3.4.0.1 TCP Reno

This is the traditional algorithm in most operating systems currently deployed, and we consider it as a base case. TCP Reno defines four key mechanisms: slow start, congestion avoidance, fast retransmission and fast recovery. In the 'slow'-start phase, the congestion window grows exponentially increasing *cwnd* by 1 with every acknowledgement, until a timeout occurs or a duplicate ACK is received. The latter implies that a packet has been lost and this signals that the sender transmits packets faster than the network can handle. Congestion control algorithm is used to slow down the transmission rate.

In the congestion avoidance phase, the sender grows its window linearly assuming that the sending rate is close to the bottleneck capacity, until it detects a packet loss or a timeout. Reno also includes a fast retransmit and recovery mechanisms which make it possible to quickly recover lost packets.

3.4.0.2 TCP Vegas

TCP Vegas was introduced with the idea that it is more efficient to prevent congestion than to fix it. One of the core features of Vegas is that **all changes are confined to the sender side**, including loss detection, estimation of the available bandwidth, and the new slow start behavior. These modified mechanisms use observed delay to detect an incipient stage of congestion and try to adjust the congestion window size before packets are lost. Thus, Vegas

attempts to determine the correct window size without relying on packet losses.

3.4.0.3 TCP Westwood

TCP Westwood enhances the window control and backoff process. Westwood relies on end-to-end rate estimation. The key innovative idea is to continuously measure at the TCP sender the packet rate of the connection by monitoring the rate of returning ACKs while trying to find the bandwidth estimate which is defined as the share of bottleneck bandwidth available to the connection. The estimate is then used to compute congestion window *cwnd* and slow start threshold *ssthresh* after a congestion episode, that is, after three duplicate acknowledgments or a timeout. Westwood is a sender side modification of the congestion window algorithm aiming to improve the performance of Reno in wired as well as wireless networks. However, the available bandwidth estimation algorithm is complex and may not be able to follow the rapid changes in a hybrid wireless network.

It fully complies with end-to-end TCP design principle. Whenever a Westwood sender detects a packet loss that indicates a timeout has occurred or that three duplicate ACKs have been received, the sender estimates the bandwidth to properly set the congestion window and the slow start threshold. Westwood avoids overly conservative reductions of *cwnd* and *ssthresh*. The available bandwidth estimation algorithm is complex and cannot follow the rapid changes in a hybrid mobile network.

3.4.0.4 TCP CUBIC

TCP CUBIC [64] was proposed to address the underutilization problem due to the slow growth of TCP congestion window in high-speed networks. The window growth function is updated with the elapsed time since the last loss event, so that its growth is independent on network delay. This means the sender is allowed to put more packets without long waiting for the acknowledgements in a network with large bandwidth delay products, probing the bottleneck bandwidth quickly. CUBIC has been used by default in Linux kernels since version 2.6.19.

Its window growth function is based on the elapsed time since the last loss event independently of RTT. The congestion control algorithm of CUBIC increasing sender's window aggressively. and the fast responses in fast long distance networks The window growth function of CUBIC is updated in a real-time so that its growth is independent of RTT. by a cubic function in terms of the elapsed time siThis is the preferred algorithm in the newest Linux kernels (after 2.6.13).

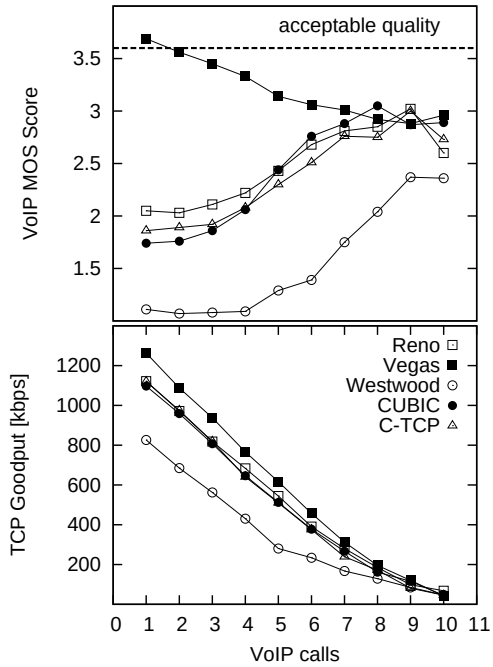


Figure 3.5: Upper: VoIP quality with different variants of TCP; Lower: TCP goodput. TCP variants are too aggressive and use all available bandwidth, reducing voice quality. However, with more voice calls TCP experiences more packet loss, which leads to retransmissions and frequent slow start phases.

3.4.0.5 Compound TCP (C-TCP)

With the idea that pure loss-based or delay-based congestion control approaches that improve TCP throughput in high-speed networks may not work well, this algorithm is designed to combine two approaches. C-TCP [65] can rapidly increase sending rate when network path is under utilized, but gracefully retreat in a busy network when bottleneck queue grows. C-TCP is the algorithm included with Windows Vista and Windows Server 2008. However, due to the loss-based component, CUBIC and C-TCP are not designed for high-loss wireless paths.

What is true for all TCP variants is that data packets arrive at the receiving host at the rate that the bottleneck link will support. A TCP sender's self-clocking depends on the time spacing of ACKs being preserved end to end. Jitter introduced in network queues misleads the sender into pushing more data than the network can accept. Cumulative acknowledgement or ACK compression may cancel the spacing of the ACKs and result in bursty traffic with a high risk of high peak rate beyond network capacity. A

single ACK can acknowledge several thousands of packets, opening up the window in a large burst.

We compared these five TCP variants with respect to their capacity to coexist with VoIP and utilization of the multihop. Fig. 3.5 shows that all TCP variants fail to protect VoIP in a simple shared environment. Most of the time, they simply increase TCP goodput with large window size. Vegas exhibits both a better VoIP protection and utilization of the multihop links, due to its balanced congestion control in low number of VoIP flows. Surprisingly, Westwood, which is designed specifically for lossy links performs worst on both measures, wasting half of the capacity on retransmissions $\frac{goodput}{total\_sent} \approx 0.5$ (not shown in the figure).

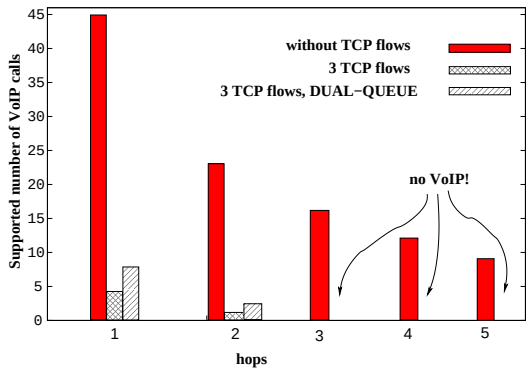


Figure 3.6: VoIP quality degradation with 3 TCP flows, with and without service differentiation. Maximum number of calls are values from Table 3.2 repeated for reference.

3.4.1 Policing TCP traffic

In fact, an even more likely situation is that none of the TCP endpoints can be controlled because upgrading TCP is unfeasible or undesirable for other reasons. Even enhanced TCP endpoints cannot possibly protect wireless multihop networks in the path. We therefore explore other methods to enable the coexistence at the gateway into the wireless multihop. Policing of TCP traffic can be performed using classical methods such as priority queues and traffic shaping, or by instrumenting TCP packets to manipulate receiver's advertised window (*awnd*). Any of these methods has the goal of reducing the amount or the shape of the TCP data pushed into the multihop.

3.4.1.1 Priority queues

One solution to harmonize VoIP and TCP traffic is the use of priority queues. We simulated priority queues in *ns*-2 allocating the highest priority for VoIP traffic in all nodes. We found that only 20% of the voice capacity can be used, and only for one or two hops. For cases of three or more hops, priority queues are not able to support any amount of VoIP traffic. The reason is that priority queues, or even 802.11e[3] cannot protect from interference generated two or three hops away. On the contrary, this approach even increases packet burstiness while building up TCP packets in the queues. These localized approaches cannot provide solution to a global problem of hidden terminals interfering across several hops.

Figure 3.6 shows the number of voice calls that can be supported in the presence of TCP traffic, and compares it with maximum possible values listed in Table 3.2. For 1 hop, the maximum number of VoIP calls is around 45, however with TCP traffic, it is reduced to 4 calls. In case of more than 3 hops, it becomes even worse – no voice calls are supported. The throughput of lower priority TCP is 6.3Mbps in the presence of higher priority VoIP calls (1 hop, 3 TCP flows, 7 VoIP calls supported).

3.4.1.2 Window resizing

TCP bandwidth discovery operates from the sender, and cannot be easily manipulated. The advertised window of the receiver however, can be instrumented in the network to reflect the actual bandwidth available in the wireless network. In concordance with previous studies, we found that limiting TCP sending behavior has beneficial effects even in the case when only TCP traffic is present in the network. In order to control TCP sending rate without modification of TCP endpoints and maintain end-to-end semantics, we modify the advertisement window in each ACK packet at the gateway (G in figure 3.1). This method limits the total number of TCP data packets in transit between the end points. If the gateway changes TCP advertisement window based on the network status, TCP throughput can be limited close to its entry point. By keeping the window size small to protect VoIP, retransmission and fairness problems among TCP flows are also relieved.

As shown in Figure 3.4, default TCP behavior does not discover the proper W* and generally pushes too much data. If the knowledge of optimal W* is available, the edge routers in the wireless domain could modify the advertisement window size of TCP ACK packets, an operation transparent to TCP applications. This window-controlled TCP traffic does not produce more data pack-

[3] 802.11e uses multiple queues for downlink traffic, and preferential contention parameters for uplink traffic in order to offer priority to QoS traffic.

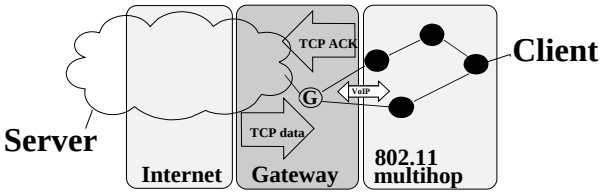


Figure 3.7: Window resizing: gateway modifies TCP advertisement window size in the TCP ACK packets according to the traffic in network. TCP data packets pass through unchanged, but the server across the internet pushes less data as for a slower client.

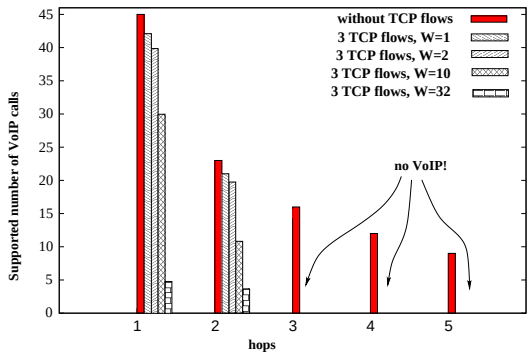


Figure 3.8: Maximum number of VoIP call supported when TCP advertisement window size is instrumented to reduce TCP share. The actual window needed is dependent on configuration: number of hops, number of TCP flows, RTT.

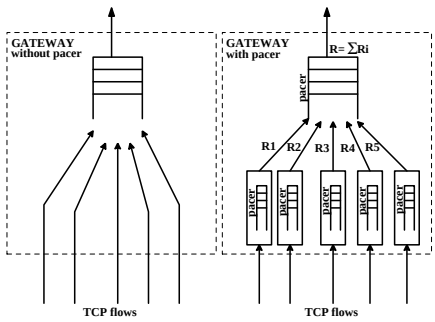


Figure 3.9: TCP data pacer: each flow may be shaped individually for fairness, but the total TCP traffic should also be shaped to make it VoIP friendly.

ets than the path capacity which the network is able to support, allowing VoIP packets to be transferred within the delay and jitter budget. Figure 3.7 shows the architecture of window resizing mechanism. Gateway node G peeks into all TCP ACK packets to modify TCP advertisement window field such that VoIP traffic can at least have a proper share of the bandwidth. TCP data passes through G untouched. *How to harmonize the window size with VoIP?* In Figure 3.8 the maximum VoIP capacity is shown with solid bars. The patterned bars indicate the VoIP capacity with the window limitation feature. While the actual result is dependent on the characteristics of the network simulated, the trend shows that smaller window sizes bring more benefit for VoIP support and even for TCP traffic itself in case of no VoIP traffic. With 3 TCP flows, optimum window size (number of unacknowledged packets) is one or two. As the window size increases, the number of TCP packets in transit gets larger, and the timely delivery of VoIP packets is affected. TCP clients with large default window size (64KB in Linux 2.4, 47KB in *ns*-2) consumes almost the whole bandwidth allowing only 5 VoIP calls to be supported even in a single hop network. It is clear that smaller windows are more beneficial than larger windows, and this is a consequence of the fact that TCP's share of the wireless medium needs to be reduced [62]. Unfortunately, this solution does not seem scalable with the number of TCP flows.

3.4.1.3 TCP data/ACK Pacing

One problem that is not solved by window resizing is that of packet bursts. TCP pacing promises to reduce burstiness of TCP traffic and alleviate the impact of packet loss, network delay, and delay jitter of VoIP traffic. TCP pacing evens out the transmission of a window of packets based on a shaper parameter R. After a packet of size pkt_size goes out in the air, the next packet is scheduled no earlier than $\frac{pkt\_size}{R}$. The gateway chooses a rate R based on the network status to determine how much to send, and when to send. One way to understand the impact of pacing is to consider burstiness from network delay, jitter and packet loss perspective. With bursty traffic, packets arrive all at once at the gateway. As a result, queueing delay and delay jitter of VoIP packets grows linearly with TCP load due to large packet size, even when the load is below capacity. From the viewpoint of TCP, 802.11 links which support VoIP traffic seem to have large capacity left for TCP data. This ignores the interference side effects that are felt several hops away from the link in question.

TCP data shaper shown in Figure 3.9 evenly spaces data packets sent into the network for each TCP flow as well as for multiple TCP flows. TCP shaper first calculates the fair share R_i for each TCP flow from the total share for TCP. Data from each TCP shaper might still create burst packets, and to handle this, there is one more pacer with rate R for all TCP flows. This policy enforces fairness between TCP flows, but other policies or priorities can be used.

In Figure 3.10 the performance of the TCP data shaper is compared with the basic case. Although it supports a lower number of voice calls, the token bucket offers

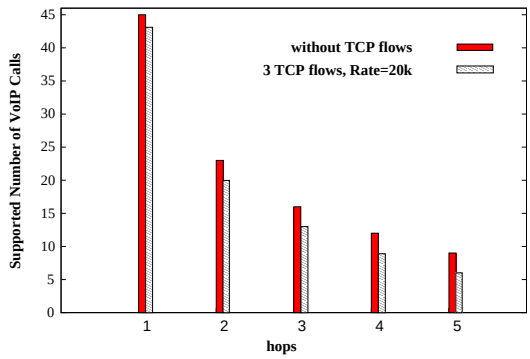


Figure 3.10: Maximum number of voice calls supported when TCP data is policed with a shaper: it scales better to higher number of hops, and it provides reasonable utilization.

some protection to VoIP traffic. The disadvantage of this method is that it may allow some unfairness when several TCP flows are shaped together along the same path. In addition, buffer overflows at the gateway due to capacity fluctuation will cause drops, unlike the window resizing solution. However, the main advantage is that it works with higher number of hops, and does not require instrumentation of TCP packets.

In our measurements, the pacer offered protection to VoIP at the cost of sacrificing available bandwidth for retransmissions. While providing benefits such as small buffer size at the pacer, ACK pacing may fail to prevent bursty data packets which results in low TCP performance and degradation of VoIP quality. The disadvantage of pacing is that buffer overflows at the gateway due to capacity fluctuation will cause packet drops and increase queueing delay. The increased queueing delay easily causes TCP retransmission timer to expire, which results in retransmitting the packets already transferred to the receiver, unlike the window resizing solution. However, the main advantage is that it works with higher number of hops, and does not require instrumentation of TCP packets.

3.4.2 Window resizing vs. pacing

In this section, we examine in detail which of the two candidates is more appropriate to protect voice traffic and provide better utilization of the multihop. For voice emulation, we generated and analyzed flows of 50 packets per second, 20 bytes per packet in each direction that emulate G.729a traffic. The reason for using this type of traffic is that most VoIP SIP phones (Zyxel, Utstarcom, Netvox) support it, and can be evaluated using the loss and the delay measured in the network (Figure 3.2), without employing waveform analysis such as PESQ. In our experiments, G729a like traffic is considered supported if it achieves a quality better than MOS=3.6.

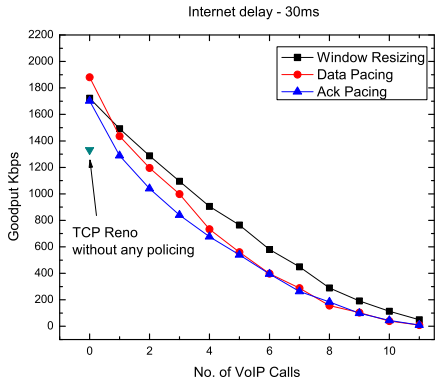


Figure 3.11: Shared capacity with VoIP and TCP using data pacing, ACK pacing, and window control. All methods are bounded by the nominal capacity of the network. Window resizing reduces total number of retransmissions, which leads to a higher TCP goodput compared to data/ACK pacing

Testbed setup: We implemented TCP data pacing and window control in an 802.11a test-bed using Atheros based cards using the madwifi driver under Linux. The test-bed consists of 5 nodes spread over the floor of a building such that a string with appropriate combination of contention and interference is created. In particular, nodes that are two hops apart are out of communication range, but within carrier sense range. Nodes three hops apart are out of contention range but within interference range. First and last nodes are outside contention and interference range of each other. This particular setup captures a variety of situations that is likely to be encountered in indoor 802.11 networks - both WLAN and multihop. An additional reason for aiming at a interference/contention ridden setup is that without interference, wireless networks would behave much like wired networks, in which more traditional methods of separation of traffic are available.

Communication range was verified using icmp traffic. Carrier sense range of two nodes was verified using broadcast traffic at maximum capacity from both nodes. If the amount they put on the air sums up to 100%, they are in carrier sense range. If they output a total of 200%, they are not deferring to each other. Interference relations are verified by running triplets of source, destination and interferer, with the interferer outside the carrier sense range of the source. An actual interferer would reduce the throughput between source and destination. For more details on how to measure carrier sense and interference conditions, see [66].

Network utilization with TCP and VoIP: In Figure 3.11,

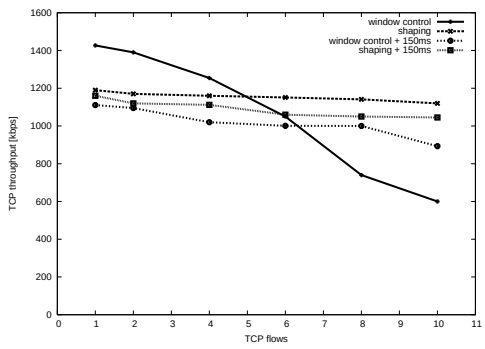


Figure 3.12: TCP data pacing scales better with number of TCP flows, and with TCP internet delays.

we look at how TCP and VoIP can share the available bandwidth using window control and data/ACK pacing. On the horizontal axis, we increased the number of calls from 1 to 11, and attempted to maximize the TCP throughput while still maintaining *MOS-score* of 3.6 for the VoIP traffic. While all three control methods achieve some amount of sharing between the two types of traffic, the window control attains better utilization. The benefit of TCP policing is visible even without VoIP traffic when plain TCP wastes capacity on retransmissions achieving a lower goodput.

Scalability with respect to number of TCP flows and amount of internet delay: From previous experiments we can conclude that while all methods can be used to control TCP rate, with the window control having a slight advantage by providing higher utilization. We then experimented with increasing number of TCP flows and found that window control cannot support more than 11 TCP flows when 3 voice calls are present as it requires a window size less than 1 packet. When using internet delay of 150ms(RTT), the required window sizes are larger, but only 15 TCP flows can be supported due to the same reasons.

We now look in more detail at the effect of larger number of TCP flows and of internet delay (Figure 3.12). First, we inject between 1 and 10 TCP flows over a fixed voice load of 3 voice calls. As before, the goal is to maintain an *MOS-score* of 3.6 for these three voice calls, while maximizing the throughput of TCP. The curve labeled 'window control' in Figure 3.12 shows the degradation caused by the excessively small TCP window required. The shaping method is more immune to the amount of TCP flows, but has a lower utilization when there are few TCP flows. Another interesting aspect is when TCP flows are terminated across the internet, incurring additional delay. We added 150ms one way delay to the TCP traffic in our testbed, while maintaining the VoIP traffic at the speed dictated by the load. We see a general reduction in utilization, but the difference between the two methods is smaller, as well as the dependence on the number of TCP flows.

If TCP traffic terminates across the Internet, connections with high bandwidth-delay product might still require a large window in order to achieve the desired TCP throughput. Consider the example when the optimal window size is $W = 2$ on a 4 hop topology: 6 calls are being supported, and a remaining bandwidth of 600kbps can be used by TCP when $RTT = 2 \times 8 \times 1500/600000 = 40ms$ across the multihop, according to $RTT = \frac{W}{Bandwidth}$. When an internet delay of 100ms is included and TCP faces an $RTT = 140ms$ end to end, in order to achieve the 600kbps available, a window $W = 7$ is needed, which is larger than the optimal window size. While achieving the job of limiting the data in the wireless string, $W = 7$ also allows bursts of 7 packets of 1500 bytes, thus disturbing VoIP flows. The window control is not able to prevent packet burstiness with many TCP flows, but even for even one flow, when the delay is large. We conclude that **shaping is more robust against varying conditions such as number of TCP connections and increased internet delay, although it comes at the cost of slightly reduced utilization**, especially in the case when there are few TCP flows. TCP window control has the potential to achieve better utilization, but is more sensitive to external conditions, so it requires a highly dynamical behavior that depends on load, delay, and number of connections. In addition, it requires instrumentation of TCP traffic, which can be undesirable for high speed implementation, or when transporting encrypted traffic.

Both methods examined have a straightforward application only when the wireless capacity is fixed. In reality, the capacity is highly variable depending on a multitude of factors: number of hops and their configuration, the amount and type of interference, the actual capacity of each hop, the amount of voice to be served. To support VoIP under varying conditions, the basic policing tools examined here should be used in conjunction with methods to dynamically estimate available bandwidth in real-time.

3.5 *Voice Adaptive Gateway Pacer*

Having established that pacing is an appropriate method to control TCP in a shared network in Section 3.4.1.3, the question is what data rate should TCP get? We can di-

vide bandwidth statically between the voice calls and the amount of TCP, but the static bandwidth sharing causes poor network utilization when usage is low for TCP and high for VoIP or even fails to protect voice in case of poor wireless link. In reality, the capacity is highly variable depending on a multitude of factors: network topology and its configuration, the amount and type of traffic, the actual capacity of each node, the amount of voice and other traffic to be served for each node. The method we propose –*Voice Adaptive Gateway Pacer* uses the existing voice traffic as probe traffic to estimate the rate that can be given to TCP:

$$R_{TCP}(t) = \sum_{i=1}^{N} R_{TCPi}(t) = \big(R_{TCP}(t-1) + R_{adj}(t)\big) \quad (3.1)$$

The bandwidth for TCP flows can be estimated by measuring the average of number of bytes transmitted over the interface, $R_{TCP}(t-1)$, adjusted with R_{adj}, the variation of VoIP quality. The adaptive transmission rate R_{TCP} for TCP bandwidth computed by the multihop gateway is allocated to each TCP flow following a bandwidth sharing policy. The period of adjustment and the time scale used to measure the rate depend on the estimated round trip time and the changing ratio of the voice quality. A basic approach to allocate the bandwidth would be to measure the maximum achievable bandwidth, and to calculate the target bit-rate of the different flows according to the weights assigned to each flow by the bandwidth sharing policy. This policy is outside the scope of this paper, and we use fair sharing for the rest of the experiments/simulations.

In order to smooth out TCP burstiness either for flows with a large BDP[4] which have intra-flow burstiness, or for other flows which may show inter-flow burstiness, rate based traffic control is necessary. *VAGP* is designed to reduce TCP packet bursts by releasing packets smoothly into the network rather than in bursts. With long end-to-end delays, TCP application tends to inject enough packets to fill high BDP paths [67]. With wireless multihops this creates congestion, and drops in the multihop portion require retransmissions, which reduce overall utilization. The method we propose controls the data transmission rate and dynamically adjusts the parameters for the actual TCP sending rate to consume residual bandwidth while keeping good VoIP quality. *VAGP* has two

[4]Bandwidth×delay product (BDP) refers to the product of a data link's capacity and its end-to-end delay (sometimes the data link's capacity times its round trip time).

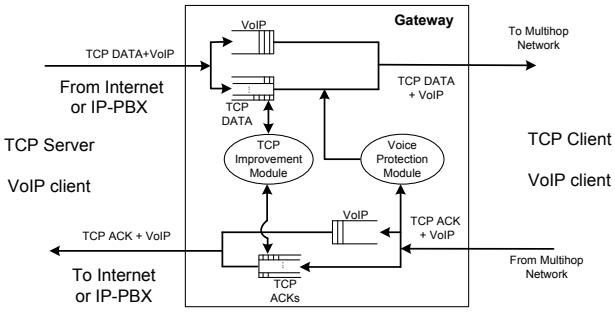


Figure 3.13: *VAGP* functionality installed in the gateway monitors both VoIP and TCP traffic. It controls TCP sending rate based on the current estimation of VoIP quality. Voice Protection Module adjusts TCP data sending rates based on measured voice quality. TCP Improvement Module eliminates redundant packets which were reached to the destination and retransmits the dropped packets indicated by the duplication ACKs

main functions: to tame the bandwidth discovery nature of the TCP at least for the wireless multihop portion making the TCP friendly to VoIP traffic, and to maintain high utilization. These functionalities are implemented in two modules that monitor/control both the voice and the TCP queues.

3.5.1 Voice Protection

As shown in Figure 3.13, Voice Protection Module (VPM) monitors the network parameters of VoIP traffic, including network delay, network loss, and jitter loss. VPM calculates *MOS-score* of the monitored VoIP and estimates the bandwidth portion of TCP traffic. It then translates the available bandwidth into a TCP sending rate for each TCP flow using a bandwidth sharing method (i.e. fair share). Algorithm 3.1 briefly shows how *VAGP* works. It estimates the bandwidth portion for TCP traffic with the measured *MOS-score* which is classified using three thresholds for triggering TCP rate adjustment: good, fair and poor. TCP rate is adjusted either by a modest amount A, or aggressive A_{agg}, depending on how much current voice quality is different from Q_{fair}.

3.5.2 TCP Improvement

In Algorithm 3.1, VPM limits TCP share of the bandwidth in favor of VoIP traffic. However, it may cause TCP performance degradation due to queue overflow at the gateway or frequent TCP timer expiration at the sender while waiting too long for ACK delivery. TCP retransmission timer expiration may trigger unnecessary retransmission for packets in one of the following situations: 1) waiting

Algorithm 3.1 Voice Protection Module

```
Voice quality parameters: Q_good, Q_fair, Q_poor;
TCP rate adjustment: A, A_agg;
for all VoIP flow i do
    Measure MOS-score;
end for
for all TCP flow i do
    if MOS-score > Q_good then
        R_i ⇐ R_{i-1}(1 + A_agg);
    else if MOS-score ∈ [Q_fair, Q_good) then
        R_i ⇐ R_{i-1}(1 + A);
    else if MOS-score ∈ [Q_poor, Q_fair) then
        R_i ⇐ R_{i-1}(1 − A);
    else if MOS-score ≤ Q_poor then
        R_i ⇐ R_{i-1}(1 − A_agg);
    end if
end for
wait(period);
```

Algorithm 3.2 TCP Improvement Module

```
for all received TCP packets do
    if packet type == TCP Data then
        read seq from TCP header
        if seq <= highest_seq then
            Drop it (Redundant Retransmission Drop);
        else
            send it to TCP Queue
        end if
    else if packet type == TCP ACK then
        read seq from TCP ACK header;
        if duplicated ACK then
            retransmit TCP data with seq + 1 (LR);
        else
            update highest_seq
        end if
    end if
end for
```

in the queue at the gateway for the multihop access, 2) being delivered over the multihop network, and 3) already arrived at the destination. These retransmissions waste the multihop resource and decrease both VoIP and TCP performance. Because of larger packets, TCP also suffers higher packet drop ratio, which decreases the number of packets in the path, which finally results in poor performance in high BDP networks.

To fully utilize the space left by voice traffic, TCP Improvement Module (TIM) uses two mechanisms – local recovery (LR) and redundant retransmission drop (RRD). Algorithm 3.2 outlines the functionality provided by these two procedures. When a duplicate ACK is received, LR retransmits the lost packet to the destination eliminating the delay of the packet transmission all the way from the sender. Packets with lower sequence number lower than *highest_seq* are dropped by RRD to prevent unnecessary transmission of packets already at the destination.

3.5.3 Queue Management

Instead of transmitting packets immediately upon receipt of TCP data, TIM delays transmitting packets to spread them out at the rate controlled by VPM, causing a increased queueing delay, which results in long RTT at the TCP sender. Queueing delay increases linearly until the number of packets reaches to min(*awnd*, *cwnd*). The local recovery mechanism of *VAGP* requires more delay while the lost packets in multihop are recovered by local retransmission. With no packet drops and little few timer expirations, TCP sender may generate more packets perceiving the multihop as a large BDP network, which increases queuing even delay further. To address these problems, we use an adaptive queue-aware window control algorithm to minimize the queuing delay and buffer requirement at the gateway. When the multihop is bottlenecked, the queueing and network delay are much larger than the ones in the wired part:

$$q_i(t) + M_i(t) < min(awnd(t), cwnd(t)) \tag{3.2}$$

At an arbitrary time t, let us denote the flow queue length of the gateway as $q_i(t)$ and the number of packets in delivery over the multihop (*max_sent_seq* - *highest_ack*) as M_i. We use subscript i to index the connections. Inequation 3.2 represents the maximum number of packets in transit which are clearly smaller than TCP sender's window size. One can also see the queue length can be controlled by modifying the advertisement window *awnd* in the ACK packets. In the queue-aware window control management, we use the downstream queue length $q_i(t)$ to represent the queueing delay. The basic control strategy for the queue management is as follows: 1. TIM reduces *awnd* to $q_{min} + M_i$ as soon as the queue length exceeds a threshold, q_{max_i}; 2. TIM increases *awnd* to $q_{max\_i} + M_i$ when the queue length falls below a threshold q_{min_i}, here $q_{min\_i} < q_{max\_i}$; 3. if TCP bandwidth becomes 0, TIM reduces *awnd* to 0 to freeze all timers at the TCP sender. Otherwise, *awnd* will be set to $\frac{q_{max\_i}+q_{min\_i}}{2} + M_i(t)$. We use q_{min_i} = 2 and q_{max_i} = 6 which is a reasonable choice to obtain low queueing delay while preventing buffer underflow.

This strategy is shown to reduce the queueing delay while keeping minimum number of packets - low queueing delay, as well as preventing the underflow of the

Algorithm 3.3 Queue-aware window control operation

if $q_i(t) < q_{min\_i}$ **then**
  $awnd \leftarrow q_{max\_i} + M_i(t)$
else if $q_i(t) > q_{max\_i}$ and $R_i(t) > 0$ **then**
  $awnd \leftarrow q_{min\_i} + M_i(t)$
else if $q_i(t) > q_{max\_i}$ and $R_i(t) == 0$ **then**
  $awnd \leftarrow 0$
else
  $awnd \leftarrow \frac{q_{max\_i}+q_{min\_i}}{2} + M_i(t)$
end if

queue. Our in-depth simulation results have shown reduction of RTT from 3 seconds to 310 msec in case of 4 hops in Table 3.3 while the same TCP goodput and voice quality are achieved as the one without queue management strategy.

3.5.4 VAGP evaluation

In this section we evaluate *VAGP* for the voice quality achieved, TCP goodput (data rate seen by applications), and TCP throughput (data rate used by TCP). The difference between the latter two is due to losses in the wireless multihop which lead to TCP retransmissions.

String Topology: we first consider the simple string topology with both TCP and VoIP being transported across the same four wireless hops through the gateway *G* in Fig. 3.1. The TCP data originates across the Internet, on a node labeled *Server*, and is sent a wireless client attached to the access point labeled *MAP*8. VoIP traffic may originate in the PSTN network, but travels across the same four wireless hops through the enterprise IP-PBX to/from another client attached to *MAP*8. *VAGP* functionality is added at the gateway node *G*. All the wireless links operate with 802.11a at 12Mbps. This is a good example of multihop connectivity in enterprise networks which use PSTN/IP-PBX for VoIP traffic and Internet for TCP traffic. G.729 codec produces 50 packets per second of 20 bytes each in each direction. We measure all the VoIP characteristics (delay, jitter and packet loss) contributing to *MOS-score*, and TCP goodput achieved using *Voice Adaptive Gateway Pacer*.

VPM monitors VoIP quality of the voice traffic from *G*, which is the entry point of the TCP flows and evaluates the *MOS-score* periodically. For these experiments, the parameter values which are used are Q_{good} = 3.9, Q_{fair} = 3.7, Q_{poor} = 3.6, Q_{choke} = 3.3 for VoIP quality and R_{good} = 5%, R_{fair} = 3%, R_{poor} = 10%, R_{choke} = 50% for TCP rate adjustment. These values are somewhat conservative to preserve voice quality, as drops are inevitable at sudden changes in channel conditions or traffic patterns. But, as we will show in the next experiments these values work

hop	Reno			TCP-GAP			VAGP		
cnt	VoIP	TCP		VoIP	TCP		VoIP	TCP	
1	0.9	2596	2654	3.9	2587	2645	3.9	2587	2634
2	1.3	1070	1312	3.1	997	1189	3.6	750	791
3	1.9	552	869	2.3	545	853	3.7	664	705
4	2.5	309	558	3.3	287	509	3.7	325	374

Table 3.3: Average VoIP quality (*MOS*), TCP goodput and throughput(Kbps) with the 70% VoIP traffic and 30% TCP; Internet delay = 30ms.

across widely different patterns and conditions. They are in fact tuned to maintain the target of *MOS-score*=3.6, rather than specific to the topology and conditions.

In Table 3.3, we see that *VAGP* significantly outperforms TCP-GAP with respect to the basic premise of isolating the voice traffic. Using Reno, TCP traffic occupies the entire available bandwidth at cost of VoIP quality, while both flows in TCP-GAP share the available bandwidth giving a larger portion of the bandwidth to TCP traffic, which results in poor voice quality. In case of 4 hops running 7 voice calls and 3 TCP flows, *VAGP* achieves the fair share - *MOS-score* = 3.7 and 325Kbps TCP goodput, while TCP-GAP allocates more bandwidth to TCP, which results in poor voice quality, *MOS-score* = 3.3. In fact, *VAGP* also achieves a higher aggregate goodput than TCP Reno and TCP-GAP. Thus, more VoIP calls can be supported using *VAGP*, when TCP gets an optimum share of the bandwidth.

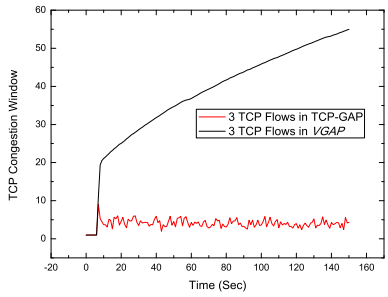


Figure 3.14: Comparison of TCP Congestion window between TCP-GAP and *VAGP*, in case of 4hop in Table 3.3

Figure 3.14 shows that *VAGP*'s congestion window keeps increasing constantly, while TCP-GAP's fluctuates over time, although around a low value. The path is never perceived as congested with *VGAP*, and TCP's share is reduced due to pacing and possibly increased RTT. Being designed to operate over 4-hops, TCP-GAP shows poor performance within less than 4-hop topology. At 4 hops, *VAGP* produces a slight increase in RTT, but with a lower standard deviation as shown in Table 3.4.

In summary, for the string topology *VAGP* im-

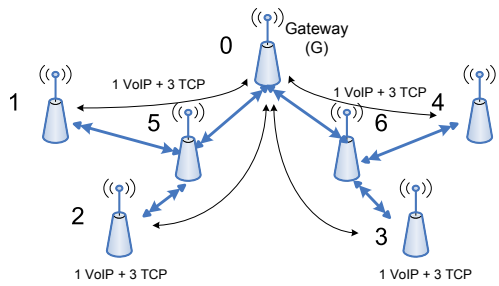


Figure 3.15: Tree topology: downstream TCP flows are set up between wired servers (not shown) and leafs. VoIP sessions are set up between IP-PBX (not shown) and each leaf. Performance shown in Table 3.5

.

RTT(ms)	Reno	TCP-GAP	*VAGP*
mean	242	288	312
std	271	209	100

Table 3.4: *VAGP* reduces variation in roundtrip time. This corresponds to 4 hops topology in Table 3.3

proves **both** VoIP performance and network utilization ($\frac{goodput}{throughput}$).

Mixed flow sources/destinations: We consider the case where three VoIP flows and one TCP flow originate in the wired domain and end at MP6, MAP3, MP7, and MAP8 respectively. Results are summarized in the first row of Table 3.5.

Tree Topology: as a more complex topology, we consider a tree which consists of seven nodes placed in a tree two hops deep, with the gateway G is positioned at the root of the tree as shown in Fig. 3.15. When run separately, this topology can support either 2 VoIP calls per leaf, or 3199 Kbps of aggregate TCP goodput in case of 3 TCP flows per leaf node. If we mix 1 VoIP call and 3 TCP flows for each leaf, *VAGP* gets less goodput than TCP-GAP, but VoIP quality is maintained above the *MOS-score*=3.6 required (second row in Table 3.5).

***VAGP* with TCP variants:** we consider the same

Topology	TCP-GAP		VAGP	
	VoIP	TCP	VoIP	TCP
String - 4 hops	2.6	437 732	3.7	309 346
Tree - 2 depth	1.1	985 1125	3.8	416 463

Table 3.5: *VAGP* performance compared with TCP-GAP with mixed flow destinations: string and tree. Columns show MOS score for voice, goodput and throughput for TCP. *VAGP* achieves VoIP protection as well as TCP performance with little sacrifice of the aggregate goodput. Standard TCP is not included because it doesn't perform in the simple string case.

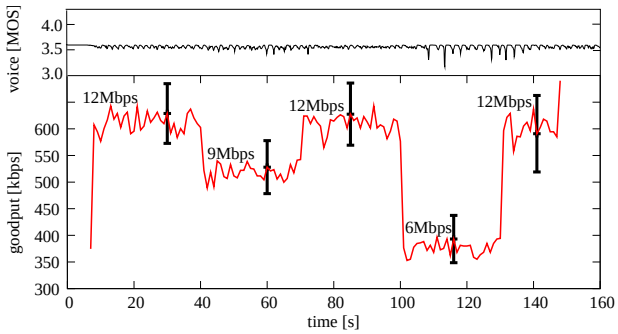


Figure 3.16: TCP uses the residual bandwidth while VoIP quality is kept high, 5 calls 1 TCP over 4 hop string topology, G729a, 12Mbps, link capacity between MP6 and MP7 varies from 12 Mbps to 9 Mbps during 40s - 70s, 12 Mbps during 70s - 100s, 6 Mbps during 100s - 130s, 12 Mbps during 130s - 150s. The vertical bars represents the standard deviation of TCP goodput

string topology as depicted in Fig. 3.1 to show fairness between TCP variants under the control of *VAGP*. Running 5 TCP flows consisting of five TCP variants between wired server and MAP8 and 5 VoIP calls between PSTN and MAP8, we see that *VAGP* works well with any type of TCP. A fair share of around 20% is maintained between TCP flows while 5 VoIP calls are supported with an acceptable quality of *MOS-score* = 3.8 (Table 3.6).

Dynamic Bandwidth Estimation: in this experiment using the same string topology as depicted in Fig. 3.1, we verify the capability of *VAGP* to support VoIP together with TCP flows when the actual capacity of the multi-hop link is varying. The timeline in Fig. 3.16 shows how *VAGP* adjusts TCP bandwidth consumption for one TCP flow while keeping good VoIP quality for 5 VoIP calls when the capacity of a 802.11a link fluctuates. On the horizontal axis, we have time in seconds, the top graph shows the voice quality, while the bottom graph shows the TCP end to end goodput. The bitrate of the link between nodes MP6 and MP7 is changed in the following sequence: 12Mbps-9Mbps-12Mbps-6Mbps-12Mbps at times 40s, 70s, 100s, 130s. This emulates the behavior of a rate adaptation algorithm, or simply the variation in capacity of an indoor wireless channel. VoIP maintains average *MOS-score* above 3.6 for all calls during this period.

Quick Responsiveness: looking at the detail of the timeline around 100s, we can see how *VAGP* swiftly reduces the TCP share R in order to protect VoIP. After an initial drop of R_{choke}=50%, R is adjusted up using R_{good}=5% to the feasible rate of around 0.38Mbps. This event is completed in 4s, ensuring that VoIP quality is maintained.

Hop count	VoIP (MOS)	Total TCP 627 (Kbps)				
		RENO	C-TCP	CUBIC	VEGAS	WW
4	3.8	125 (19%)	126 (20%)	126 (20%)	124 (19%)	126 (20%)

Table 3.6: *VAGP* ensures fairness between flows, regardless of the TCP flavor they use

3.6 Summary

TCP and VoIP can coexist in interference-ridden multi-hop networks, but only with some policing help. None of the existing solutions (TCP variants, priority queues, MAC support) can protect VoIP, and produce low utilization as TCP wastes wireless capacity on retransmissions. We propose *VAGP*, a method that addresses **both goals** of VoIP protection and network utilization. It uses existing VoIP traffic to continuously estimate the amount of bandwidth that can be dedicated to TCP. Making TCP more CBR like is beneficial in two respects: it becomes more VoIP friendly, and minimizes self interference. The latter is also beneficial for TCP itself, which suffers disproportionately because of its large packets. Our approach, called as *VAGP*, has the following advantages:

- can be placed in the wireless gateway to monitor downlink TCP traffic;
- maintains end to end semantics and compatibility;
- doesn't require setting of parameters. An initial setting regarding the voice quality desired works across different conditions and situations.
- provides complete separation, good utilization, and swift responsiveness to changing conditions;
- performs well in a variety of conditions - various network topologies, several TCP flows, diverse network delays, different interference patterns, varying wireless capacity.

There are several other possible continuations of this work. Since interference is a phenomenon that acts across several nodes, centralized solutions would provide a global view that could protect realtime traffic. Alternately, edge solutions, such as the one presented in this article are less intrusive and are easier to implement and deploy. Currently we are developing a multihop solution for TCP traffic originating from the multihop. As opposed to the gateway approach in which all TCP traffic originating from Internet is distributed at MPs, the traffic originating from the multihop aggregates to MPs which results in bursty traffic even if the traffic from the client is regulated by MAPs.

Chapter 4

Interference in Dense Wifi Populations

4.1 Introduction

For large populations of wireless devices, such as sensor networks with single RF nodes, or meshes with multiple RF nodes with small number of independent channels, interference is the major performance factor that is part of a circular feedback of interdependence. Interference determines how channels are assigned, how flows are routed, and how calls are admitted, but all these may in turn cause changes in the interference. For these dense networks, it is therefore important to get an understanding of interference that can help heuristics for all the mentioned problems, and provide prediction for the quality of service obtainable.

When enough orthogonal channels are available to cover with channel reuse an entire 802.11 network, interference between base stations is not of concern. However, for the current crop of available channels in the unlicensed spectrum 802.11b/g the number of channels is limited such that channels need to be reused in an indoor environment that requires higher deployment density. The availability of 802.11a, with its shorter range, higher bit rates, and more orthogonal channels provides one way of scaling up the wireless service: by increasing the bandwidth per area ratio. But this is done by increasing the density of access points and the number of wireless cards per access point. In such a dense wireless network, be it multiple hop (ad-hoc, mesh), or single hop, several base stations and clients operating on the same channel are bound to interfere each other. It is an accepted fact that indoors the nature of interference is generally unpredictable for these carriers (2.4GHz and 5GHz) due to variability in building construction, people movement, and other uncontrolled sources, such as microwave ovens. Adding that channel usage is unregulated by most institutions, it is generally hard to predict what the quality of service can be achieved even in a one hop setup. For multiple hops, the problem becomes harder because backhaul traffic interferes with itself when carried on the same channel. The nature of the traffic also makes the amount of interference hard to predict - TCP traffic depends on the congestion, which means that the interference it produces depends on conditions on other orthogonal channels. This tends to link together the problems of interference, routing, and channel allocation.

The problem that we address in this paper is to predict the effects of interference under some restrictive conditions:

1. we assume that traffic is completely controlled for the channels of interest. This can be achieved through administrative ways, and through call admission policies, and has the role of making all traffic at all nodes on all channels accounted for.

2. the traffic on all nodes is constant bit rate (CBR), which avoids problems caused by time variable behavior of congestion aware protocols, like TCP. These two requirements are actually met by VoIP networks, both in WLAN and mesh networks.

3. measurements of interference can be made in the absence of real traffic. They can be performed periodically at low traffic times, for example midnight/weekend. The purpose of this last requirement is the building of the interference map – a collection of measurements in which the relation between source, destination, and the interferers is unaffected by exogenous unpredictable factors such as live traffic.

One hypothesis that we validate in this work is that interference measurements for simple configurations can be used to predict interference effects for more complex scenarios. The simple configurations have the advantage of taking a reduced time to test, and of requiring collaboration of fewer nodes (in our case three). Scenarios with many interfering nodes are harder to test mainly because of complexity - the number of tests can grow ex-

ponentially with the size of the group. If interactions of triplets of nodes can be used to predict interaction for larger groups, the short measurements (that require network downtime) can be performed more often, improving the accuracy of the prediction.

While there is research that needs interference information for the mentioned problems (channel allocation, routing, call admission), interest in the actual measurement and modeling of interference is quite recent. In [68], Padhye et al. proposed a pairwise interference measurement method, which we also use in this work. The broadcast based interference estimation is shown to be an adequate estimation for interference produced between unicast flows. They also found that carrier sense is the major cause for interference. Das et. al [69] show that remote nodes which have no interference effects in isolation may combine to produce interference when acting together, but the occurrence is rare. Our proposed model closely predicts their results for close interferers. Another study of carrier sense [70] shows that it is not always a good predictor of transmission success, and also suffers of the exposed terminal problem (close by senders cannot send simultaneously even if their destinations can actually receive packets), and is overly conservative with respect to the capture effect. The question of interference is acknowledged to be central for the problems of channel assignment, bandwidth allocation and routing in [40]. The authors find that there is a circular dependency between these problems and interference, and propose a centralized algorithm. Other researchers have identified interference as being a cause for unfairness [71]. The most studied problem is that of capacity being affected by interference [30, 41, 72, 73], but complex interference scenarios are considered an input for the optimization process, without addressing the problem of obtaining them. More recent works [74, 75] try to reduce the complexity of measuring the interference by only considering pairwise communication, but this ignores the remote interferers (outside carrier sense range), which as we will show are the main source of uncontrolled loss.

One contribution of this work is a model of interference that allows complex traffic scenarios be predictable using simple, low complexity measurements. In an 802.11a testbed, the the model is shown to be very accurate (correlation of 0.97 between analysis and measurement). This is a positive result in that it greatly reduces the complexity of obtaining the interference map for one channel, and one card per node. The second contribution is showing that the complexity of the complete interference map is much higher than previously thought: each card and each channel have in fact to be measured independently. This latter result is rather pessimistic, as it implies that in a real world setup when nodes have multiple cards, and are supposed to use orthogonal channels, the complexity of producing the complete map is prohibitive for dense networks of even moderate scale.

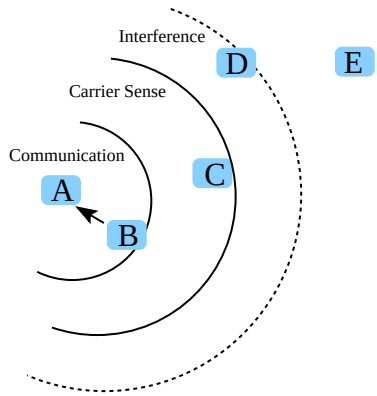


Figure 4.1: Possible interference relationships.

4.2 Interference map

If we consider two nodes, there are several relative positions of interest with respect to each other (Figure 4.1). If they are close enough, like A and B, they are in communication range, meaning that packets sent by B can be received at A. The actual distance depends on conditions, carrier frequency, and output power. For example a 5004 MP Atheros a/b/g card claims 300m outside and 30m inside when operating in 802.11b at 11Mbps. 'Distance or range' in this description are just convenient terms because in reality delivery ratio decreases from 1 to 0 in a progression that describes a donut around A, rather than a circle. The circular shape is of course true only in void, whereas indoors the shape of coverage is highly irregular. The next range of interest is that of carrier sensing (CS). The carrier sense range includes the communication range, regardless of their actual shape. When A and C are this far apart, even if C cannot send packets to A, its carrier can be sensed at A which backs off when C has a transmission of its own. A will then defer transmission so that it doesn't destroy packets originated by C (if it is the case, see next paragraph). Both communication and CS can be asymmetrical: B can send to A, but not vice-versa, or C can sense A, but not vice-versa.

The next range of interest is the interference range. This range is not in a fixed relationship with the other two ranges as indicated in the figure, but rather depends on the source as well: the interference range is defined for the ordered tuple (B, A), in which B is the source and A the destination. Assuming that B and D are outside CS range of each other and therefore send packets at the same time, the question is if A is able to receive B's packets. The interference range is then defined by all positions of D that destroy some of B's packets at A. If B's power at A does not exceed D's power at A by some capture threshold, then B's packet is lost. In literature, when D is outside CS range of B, but in communication range of A, it is called a hidden terminal. In this paper however, we refer to all D nodes that destroy packets at A as hidden terminals or interferers: close hidden terminals are ones inside CS range of the destination, whereas remote hidden terminals are outside.

Finally, if far enough apart, station E is completely out of interference range of A, meaning none of its packets can interfere with packets arrived at A, regardless of B's position. As shown recently [69], such nodes may together generate enough power to have non negligible effect over B→A, but the occurrence of this situation is fairly rare. Interference for the link B→A is defined as the cumulative effect other nodes in the network have on throughput achievable in that link. Based on the previous definitions, some nodes may reduce B's sending capacity by being in its CS range, or may destroy packets after they arrive at A. We therefore want to differentiate between sending and receiving interference because they are qualitatively different: while sending interference caused by CS is nondestructive, the receiving interference destroys packets that require retransmission. When we say that sending interference is nondestructive, we mean not that it is beneficial, but that it grabs the resource (the carrier) to support some useful transfers, whereas receiving interference is much more wasteful in that it corrupts packets at the destination, after the air resource has been used already.

Having a large population of wireless devices in an area operated on a small number of channels brings the question of how devices on the same channel interact. Any two devices are in one of the four situations described above, but the amount of interference they create depends on the amount of traffic they carry, and is therefore linked to problems of routing, load, call admission and channel allocation. The interference map is a data structure that characterizes this interaction, so that one can answer questions like: given a channel allocation and routing, is a particular traffic matrix supported? Can an additional call be accommodated? When a link goes down, can the current service be maintained? What is a channel coloring that favors particular patterns of traffic (tree, mesh)?

The information in the interference map has three disjoint, but dependent parts: delivery ratio matrix, the carrier sense matrix, and the hidden terminal relationships. The delivery ratio matrix describes the capacity for each pair of nodes in the network. This includes effects of SNR degradation because of local geography, fading, multipath, as well as external external interference sources. The carrier sense aspect governs **what can be sent** into the air, which is the first step in getting the data across in the wireless network. The hidden terminal aspect is the actual interference information and describes **what can be received** at a destination under interference from other nodes.

The interference map of n nodes on the same channel consists of:

- d_i^k delivery ratio from node i to node k without any interference, $1 \leq i,k \leq n$
- cs_{ij} what fraction of the maximum capacity node i can put on air, when j is active at maximum capacity as well, $1 \leq i,k \leq n$
- d_{ij}^k delivery ratio from i to k with interferer j sending at maximum capacity

In addition, we introduce the following notations:

- s_i^k traffic sent from i to k
- $s_i = \Sigma s_i^k$ all traffic flowing out of node i,$\forall k$ neighbor of i

All these values are normalized to the interval [0,1], and they can be easily obtained from throughput measurements as we detail in the next section. Traffic sent out is divided by the nominal capacity, while delivery ratios are directly measured as throughput of broadcast traffic, divided by nominal capacity. For example, $s_i^k = 0.3$ would mean that node i sends at 30% of maximum capacity. $d_{ij}^k = 0.7$ means that in the presence of interferer j the throughput $i \rightarrow j$ is 70% of the maximum supported by the channel.

4.2.1 Carrier sense (sending) interference

When two nodes sense each other they share the medium completely, and the sum of their maximum output rates on to the air is 1. When they are out of CS range of each other, each of them can send at full throttle, yielding a total output rate of 2. Any value between 1 and 2 is possible, because CS is not a symmetric or discrete phenomenon - one node may sense the carrier from a source only a fraction of the time, or the CS may behave asymmetrically. cs_{ij} is an $n \times n$ mostly symmetric matrix describing the capacity that two nodes can put on air when sending at the same time. $cs_{ij} + cs_{ji}$ represents the total capacity placed on the air ranging from 1 to 2. To gather cs_{ij}, we send broadcast traffic at full throttle from nodes i and j. We then use the packet rate reported as sent by each node - producing cs_{ij} and cs_{ji}. The complexity is $O(n^2)$, where n is the number of nodes involved.

Matrix cs_{ij} can also be seen as a directional CS graph: $cs_{ij} = 1$ indicates no link from i to j in the the CS graph, because i is sending at full throttle even when $j$is active. When $cs_{ij} + cs_{ji} = 1$, there are two bidirectional links - each node pointing to the other. $cs_{ij} + cs_{ji} = 1.5$ usually indicates a one direction CS link: one node has an output of 1, and the second of 0.5 because the first node doesn't hear the second. The direction of the link in the CS graph indicates the direction of sensing. To compute how much of that traffic can actually be received is the role of receiving interference mapping, which is described in the next section.

4.2.2 Hidden terminal (receiving) interference

This component is sometimes described in literature by a model called *conflict graph* [41]. The conflict graph indicates which groups of links mutually interfere and hence cannot be active simultaneously. In this paper, we quantify receiving interference by emphasizing the interferer in isolation, without considering him as part of a link. This model is appropriate when considering multiple interferers, regardless of which their destinations are.

The purpose of the receiving interference map d_{ij}^k is to have an estimate of the effect a remote source j has over traffic sent from i to k. When i and j are in CS range, they share the medium, and j does not destroy i's packets at k. However, when j does not sense i's transmission, packets received at k may be garbled - this is known as the hidden terminal problem. Example: when j is silent and i sends, throughput $i \to k$ $d_i^k = 0.8$. When i and j send, throughput $i \to k$ becomes $d_{i,j}^k = 0.4$. Conclusion: traffic leaving j produces a degradation of 50% for traffic $i \to k$. We collect the measurements d_i^k for all pairs (i,k) and $d_{i,j}^k$ for all triplets (i,j,k) in the network. The complexity of collecting the entire data set is $O(n^2)$.The measurement process has these steps (this procedure was first proposed in [68]):

1. node i alone broadcasts and all other nodes k record d_i^k = throughput of $i \to k$. All nodes take their turn in broadcasting - complexity $O(n)$

2. nodes i and j broadcast simultaneously and all other nodes k record: $d_{i,j}^k$ = throughput of $i \to k$ jammed by j and $d_{j,i}^k$ = throughput of $j \to k$ jammed by i. All possible pairs (unordered) take their turn - time complexity is $O(n^2)$.

The storage requirement for the second step is $O(n^3)$ as there is one interference measurement stored for each ordered (i,j,k) triplet in the network. For the first step, the required storage is $O(n^2)$ - throughput measurement for each directed link in the network. Note that these are in fact upper bounds for when the communication ranges (for step 1) and interference ranges (for step 2) of all nodes extend over a constant fraction of the entire network. In reality, the number of nodes that can produce interference at a destination are limited to a donut shaped region around the destination, the radius of the region being dependent on the hardware, bitrate, antenna, etc. In this case, the storage complexity could be reduced to $O(dn^2)$ where d is the degree of the node, or some other spatial density measure. Another factor in reduction of the complexity of measurements is the fact that not all links of the network are interesting, as will be seen in the experiments section.

This measurement procedure can produce a reasonably accurate image of what happens when a triplet (i,j,k) is involved in a sending/jamming process. But in reality, there are several nodes sending at the same time on the same channel, and one node's traffic is another node's interference. Having analyzed the complexities of measurement and storage for one interferer scenario only, it becomes clear that a measurement based approach is not scalable to the entire network: arbitrarily large groups of nodes can send at the same time, effectively jamming each other on the same channel. What is needed is a method that can predict the effect of several interferers acting simultaneously from single interferer measurements.

4.2.3 Analytical model

The model we propose comprises of the following two relations, the first expressing the limit for sending, and the second one the limit for receiving:

$$s_i + \sum_{j \in CS(i)} s_j \quad < \quad 1 \tag{4.1}$$

$$d^k_{i,all} \quad = \quad d^k_i \prod_{j \in I(i \to k)} [1 - (1 - d^k_{i,j}) s_j] \tag{4.2}$$

In the first condition (4.1), sending capacity of node i is limited by contention with all nodes it has to defer to. These are the nodes towards which i has a directed link in the CS graph.The second equation (4.2) models the delivery ratio of link $i \to k$ when all its interferers are active. Each interferer j contributes with an amount of interference that is measured separately as $d^k_{i,j}$. This property of independence between different interferers makes the procedure scalable as $d^k_{i,j}$ can be measured in $O(n^2)$ time for the entire network. If this independence wouldn't hold, a complete interference map would have to measure each possible group of interferers which at run time might affect the capacity of link $i \to k$. In the worst case, this is the power set of all nodes, of exponential size. The s_i factors in these equations represent the sending rates at the current node i and its neighbors j. These rates are considered known for the entire network, as stated by conditions 1 and 2 in the introduction. To understand the rationale of this second equation assume that $s_j = 1$ meaning that all the interferers send at full capacity. In this case $d^k_{i,all}$ becomes $d^k_i \prod_{j \in I(i \to k)} d^k_{i,j}$, showing that the final delivery ratio is merely a product of delivery ratios achieved when each interferer acts in isolation.

These relations can be used in any heuristics for solving problems that implicitly depend on interference: routing, call admission and channel assignment. For example, a call admission decision should first use the first relation to assess whether the proposed new traffic can be sent onto the air. There is no point sending voice traffic that is dropped even before it leaves the access point, so a call should not be admitted unless sending capacity for all nodes remains valid under condition (4.1). The second relation would then estimate the delivery ratio achievable across various links, using the measured values of $d^k_{i,j}$, and the s_i values accepted by the first step. Alternately, for a route or flow optimization procedure, these relations would participate as constraints in the optimization process (albeit nonlinear).

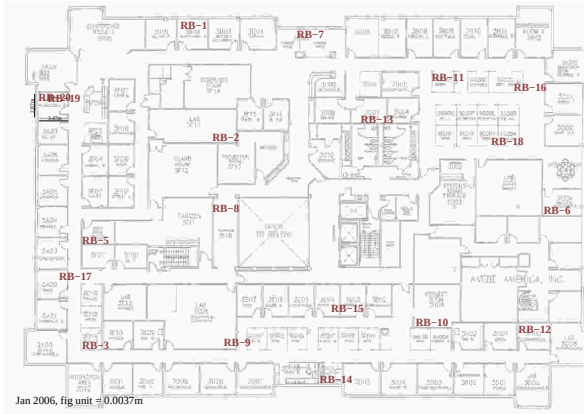


Figure 4.2: 802.11a testbed: 20 nodes in a 45m x 60m building.

The rest of the paper is devoted to validate equation (4.2), namely confirming the fact that effect of different interferers can be measured independently and used to predict complex scenarios with several interferers. In the next section, we present several experimental results that explore the dependence of packet delivery ratio on several variables: sending rate, jamming rate, and number of interferers, distance, actual card used, actual channel used.

4.3 Experimental results

4.3.1 Testbed setup

We use a 20 node testbed deployed in a 45m x 60m building (Figure 4.2). Each node is equipped with two 802.11a/b/g cards tuned to 802.11a, running Linux with `madwifi-old` driver for Atheros chipsets. In order to cover the entire building, we use the lowest bit rate setting (6Mbps) which allows the longest range indoors. We employ Click modular router [44] to generate broadcast traffic for all the measurements: delivery ratio, carrier sense and interference. One particular feature that is needed for the measurement of carrier sense is the tx feedback: the driver gives a report for each packet submitted to the card - whether it was ACK-ed successfully, retried to the maximum and dropped. For broadcast packets the feedback only says that the packet made it on the air, as there is no ACK or retry. This allows for each node to directly measure its access to the medium, without the need of other receivers. For all measurements, we collect rates in packets per second and divide them by the nominal capacity

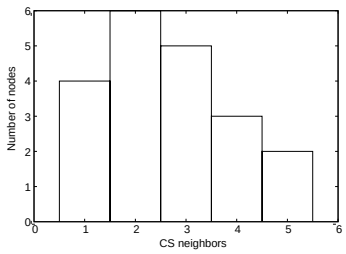


Figure 4.3: Histogram of carrier sense (CS) degree of nodes - on average, 2.6 nodes are within CS range in a mesh of 20 nodes.

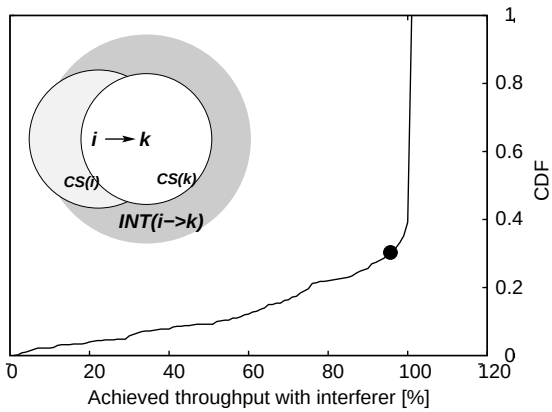


Figure 4.4: Inset: relative positions of CS and interference areas. Graph: CDF of throughput achieved in the presence of all possible remote interferers. These are the ones outside the CS range of both the source *i* and the destination *k*, therefore cannot be detected by ether the source or the destination. 70% of the potential interferers allow the link to function at 95% or more of its capacity, but this includes nodes outside the interference range of the link $i \to j$. The other 30% of jamming situation produce sizable damage on the capacity of the link.

of the channel, so that all the values handled are between 0 and 1: delivery ratio, traffic sent out, damage produced by an interferer, etc. Broadcast at all bitrates is possible with `madwifi`, so this measurement method is not limited to the basic 6Mbps rate.

We run the procedure outlined in section 4.2.2, accumulating the structures cs_{ij}, d_i^k and $d_{i,j}^k$ for all ordered triplets (i, j, k). For *n* nodes, bw_i^k is an $n \times n$ matrix, usually asymmetric, containing the throughput from node *i* to node *k*. Knowing the maximum capacity $lcap$ of a link, and assuming that bidirectional communication is usually necessary, only links with good delivery ratio are considered for the rest of the experiments.

- $d_i^k = \frac{bw_i^k}{lcap}$ is the delivery ratio for each directional link, normalized to the interval [0,1].
- $ETX_{ik} = \frac{1}{d_i^k d_k^i}$ metric describing quality of a bidirectional link - the expected number of transmissions required to send a packet from *i* to *k* or from *k* to *i*.

Assuming an ETX [48] value of 4 as a cutoff point, we are left with only 19 bidirectional 'good' links (a particular example of a bidirectional link with ETX=4 is one for which delivery ratio in both directions is 0.5). $d_{i,j}^k$ is then a matrix $p \times n$, where p is the number of interesting unidirectional pairs - 38 in our setup. Each value in this matrix represents the capacity of the pair in the presence of the interferer.

The *cs* matrix contains a directional CS graph as described in section 4.2.2. In Figure 4.3, we see how the number of CS neighbors is distributed among the 20 nodes - the average CS degree is 2.6 (compared to the average node degree with the 'good' links of 1.9). These sources of interference are not the most dangerous, since nodes in the carrier sense range take turns in sending packets, as opposed to nodes in interference range (hidden terminals).

The most critical question for any link is how many interferers are out there, and how bad are they? In Figure 4.4, we look at potential interferers for all 'good' links. The inset picture shows a source *i* and a destination *k* with their respective CS ranges. The large gray circle around the destination *i* labeled $INT(i \to k)$ represents the area of potential interferers that can affect the transmission $i \to k$. We plot the CDF of all $d_{i,j}^k$ for the selected 'good' links, and all their potential interferers . The CDF does not include interferers which are in CS range with the sender or the receiver $CS(i) \cup CS(k)$, but does include nodes which are far away from both sender and receiver, outside of $INT(i \to k)$. This last category is the largest, as we see that more than 70% of the potential interferers allow links to operate at more than 95% capacity. The rest are real remote hidden terminals producing real damage on the links - packets which are garbled at the destination.

Figure 4.5 shows the number of potential interferers and what effect they have on the link. Again, a large number of nodes (13 out of the total of 18 possible interferers) leave the capacity almost intact, meaning that the interference range covers about a third of our 20 node network. The real hidden terminals however are quite present as well: there are on average 2 hidden terminals which reduce the link capacity to 60% or less. From the statistics we eliminated non interferers outside $INT(i \to k)$. From the remaining, we also eliminated nodes which are in CS range with the sender $CS(i)$, but allowed the ones which are in $CS(k)$: these are all the effective hidden terminals (close and remote), contained in the region $INT(i \to k) - CS(i)$. In Figure 4.6 we also eliminated

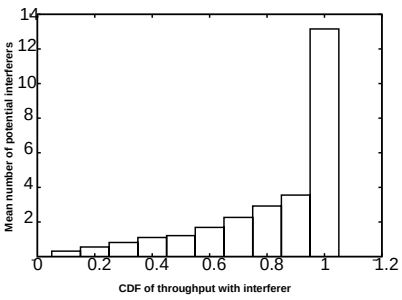


Figure 4.5: Cumulative distribution of the number of possible interferers and the amount of damage they produce. Nodes outside interference range are not included. On average, there are 2 interferers which reduce the capacity of the link to 60% or less.

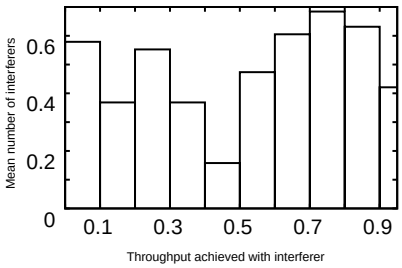


Figure 4.6: Histogram of the number of interferers for each interval of achieved throughput. interferers allowing more than 95% of the throughput are omitted. The sum of the first two bins shows that there is on average one interferer reducing the capacity to 20% or less.

interferers which produce less than 5% damage, to have a closer look at the worst offenders. The sum of the first two bins in this histogram shows that there is on average one interferer which reduces the throughput to 20% or less.

These statistics confirm that remote hidden terminals are a significant presence even in a sparse wireless network like ours, with an average degree of 1.9. These numbers are likely to be much worse in a better connected network, as all the regions surveyed here would be more populated. These remote hidden terminals will adversely affect routing, channel allocation, and call admission, as all these issues directly influence the amount of traffic on each link. Previous work [68] found that most interference is in the form of carrier sense, and we attribute this to hardware / software differences: examining interference map (cs_{ij} and $d^k_{i,j}$ structures) we found that the carrier sense range is almost perfectly overlapped over the communication range for our hardware/software configuration. This means that there are almost no cases when nodes are in CS range, but no packet can fly across (node C in Figure 4.1). The more important aspect however, is that interference range starts immediately beyond communication range, which makes this testbed appropriate

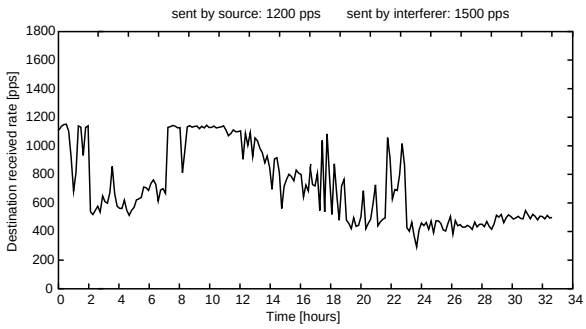


Figure 4.7: Time line: due to external factors, interference measurements taken at different times cannot be compared. We use a round robin scheme to alternate between measurements and identify stable periods, which allow for meaningful comparisons.

for the study of remote interference - area designated by node D in Figure 4.1 is quite large. Two recent contributions [74, 75] proposed modeling of the interference only on the basis of packets which are successfully received between stations, easily achieving $O(n)$ complexity for measuring the interference for the entire network. But this clearly ignores remote hidden terminals which are out of the CS range of both sender and receiver, which have a considerable effect even in sparse networks as ours.

4.3.2 Interference properties

After having established the extensive presence of both close and remote hidden terminals, we set to explore in more depth equation (4.2). Some of its more useful features are the linearity with respect to interferer rate, shown by the presence of s_j inside the product, and the linearity with respect to source rate, which is implied by the absence of s_i. For these measurements we want to compare the achieved throughput for different source/interferer rate, but how relevant is this comparison if the measurements are not taken at the same time? In most situations we want to compare configurations that cannot possibly be ran at the same time because they inherently affect each other - and this is always the case with interference measurements. In addition, the wireless medium is highly variable indoors, depending on the level of human activity. Both these factors make the measurement of the interference difficult to setup, reproduce, and interpret.

4.3.2.1 Measurement methodology

The methodology we use is to have a round robin of short runs for each experiment (E_1, E_2, E_3, E_1, E_2, ...) over longer periods of time in order to identify stable periods

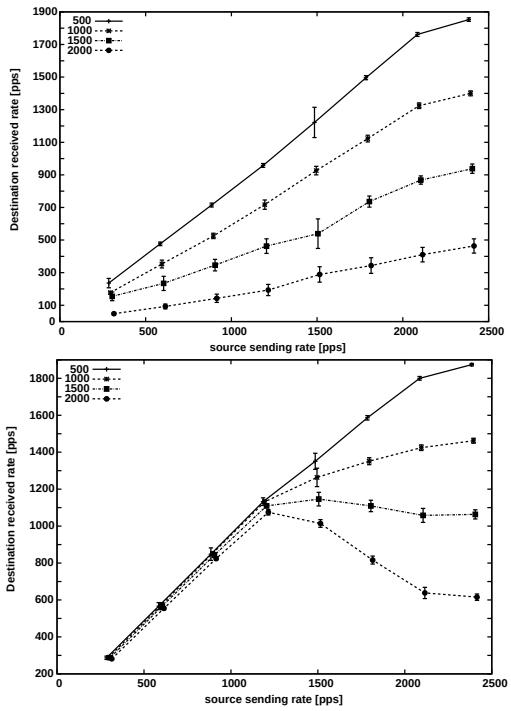


Figure 4.8: Delivery rate with one interferer. Top: in most cases, the achieved throughput in packets per second is linear with the sending rate for the entire range of sending rates. Bottom: low rate flows from the source are not affected by the interferer, but for higher rates, the behavior is still linear.

during which external conditions do not vary much and measured values show some stability. During those periods, we may compare the results of experiment E_1 with the results of experiment E_2, even if they are not run exactly at the same moment, assuming that conditions were comparable since each measured value shows stability. This method of comparison is important especially for the experiments which would conflict over resources. One example is using the same channel by two cards over the same period of time, as in section 4.3.2.5. Another one is testing the same source-destination pair of cards over different channels as in section 4.3.2.6. In all the experiments, there is a time sharing between the experiments so that each experiment has exclusive use of the resource, and yet its result can be compared with a virtually parallel experiment using the same resource.

For example, we setup three nodes - source, destination, and interferer to operate on the same channel in 802.11a, using the 6Mbps rate to send 200 byte packets in broadcast mode. The channel capacity for this setup (packet size, bit rate, SNR) is about 2300 pps (packets per second) for broadcast packets - no ACK, and no retry. In addition to other measurements mentioned below, we record the packet rate received by the destination under the conditions that the source sends 1200 pps, and the interferer sends 1500 pps. In Figure 4.7, we follow the delivery ratio at the destination over a period of 33 hours and attribute the high variation to external factors (moving people, doors). There is no institutional use of 802.11a in our building, and to the best of our knowledge there is no unaccounted traffic on the channels used. While there are large variations in the delivery ratio, we used times 8-12 and 24-32 as relatively stable periods to investigate for our purposes. In fact, all the following measurements were performed during the above time line, virtually time-sharing with the experiment in Figure 4.7.

4.3.2.2 Linearity with source and interferer rates

In the Figure 4.8(top) we vary on the horizontal axis the sending rate of the source from 300pps to 2400pps and measure the delivery rate for four packet rates of the interferer: 500,1000,1500, and 2000, represented by separate lines in the graph. Each point also shows the standard deviation over all the samples used. Because curves are mostly straight, we infer that delivery ratio is stable for different sending rates of the source. For example a delivery ratio of about 83% is maintained for the top curve when the source sends between 300 and 2100pps, and the interferer sends at 500 pps. The source and the interferer are confirmed to be outside each other CS range by periodical verification of sustained simultaneous output of 2300pps. Figure 4.8 bottom corresponds to period 24-32, and Figure 4.8 top to period 8-12, which we deemed as stable for the purposes of comparing results. For period 8-12, we can see that the linearity with respect to sending rate is not followed anymore, especially for low packet rates. Specifically, when the source sending rate is below 1200pps, the transmission is not affected by the interferer. The anomaly we believe is caused by a behavior of the Atheros chipset which sends weaker packets when the packet rate is high. A separate 16 hours experiment measuring delivery ratio between another source and a destination (Figure 4.9) shows that for longer than 10 hours, the higher packet sending rate (2100pps) consistently gets a lower delivery ratio than the lower sending rate (1800pps). We found that there were 5dB more for the signal strength of the slower rate, thus justifying the results shown at the bottom of Figure 4.8. This and other nonstandard features of Atheros based chipsets are confirmed by other researchers [76], and mostly explained by aggressive power saving implementations.

Fortunately, this behavior is sporadic, and in most cases

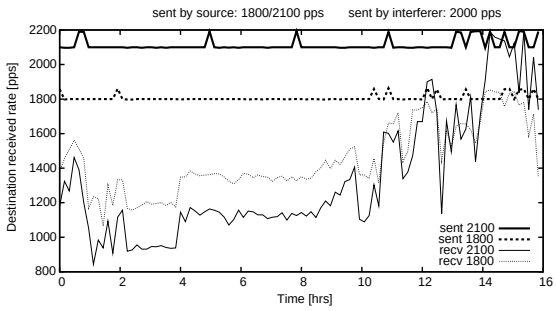


Figure 4.9: Anomaly: packets sent at higher rate use a lower signal strength, yielding in a lower delivery ratio. This behavior is persistent for many hours.

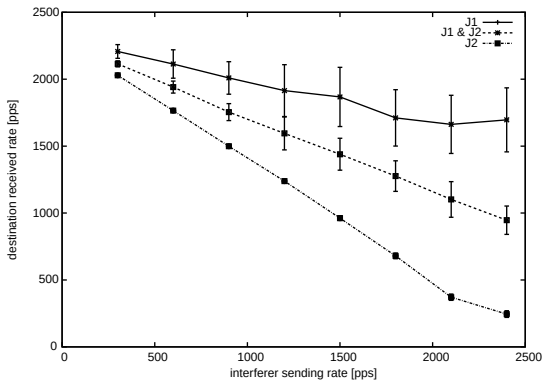


Figure 4.10: The amount of interference is linear with respect to interferer rate. Separate interferers acting simultaneously also create interference that is linear with their combined rate.

we can observe the linearity of the interference with respect to the sending rate of the sender. Although somewhat visible in Figure 4.8, the linearity with respect to interferer rate is plotted in Figure 4.10 for two different interferers. The source sent 2300pps for all experiments, whereas the cumulative interferer rate is shown horizontal axis. For the middle curve both interferers sent simultaneously each with half the rate, showing that independent measurements for each interferer can be used to derive the effect of several interferers sending concurrently. This linear combination of effects of different interferers is also maintained even in the case of anomalous delivery cases mentioned in the previous paragraph, as shown in Figure 4.11. One of the interferers produces almost negligible damage, while the second one is worse for high rates at the source, but their combined effect is piecewise linear.

4.3.2.3 Independence of different interferers

The other crucial aspect of the model we propose (equations (4.1) and (4.2)) is the fact that interferers have effects that are independent of each other. Basically, the probability of a packet being delivered in presence of several in-

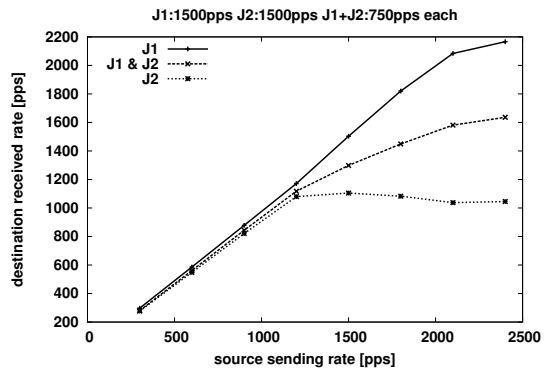


Figure 4.11: Even with anomalies in power of emitted packets, effect of interferer is linear with respect to sending rate for single interferers, and for combinations of two.

terferers is the product of delivery probabilities when the interferers act in isolation. This is the major reason why the network wide interference can be characterized with a small number of measurements: the ability to combine simple measurements for isolated interferers to predict the effect of possibly every node sending, as it happens in a network under normal use.

In a separate 34 hour experiment we verified that the delivery ratios with two different interferers can be treated as independent variables across various delivery ratios in time. We send data from a source S and two interferers J1 and J2 at the maximum capacity for several situations:

- S→D: verifies the nominal capacity of the link
- S, J1 → D: measures the capacity with J1 alone
- S, J2 → D: measures the capacity with J2 alone
- S, J1, J2 → D: measures the capacity with both interferers, monitors the CS between S,J1, and J2
- J1, J2, D: monitors the CS relation between the J1, J2, and D

The placement of the source, destination and the two interferers is set such that the source cannot sense the carrier of any of the two interferers, so it is always sending at full throttle. On the destination side, interferer J1 is far enough not to defer to any node (Figure 4.12 top). The second interferer has a clear deferral period, and an independent period. The destination (shown with dots and also with a smoothed graph, because of high variation) experiences a more ambiguous situation with respect to the interferers in which it senses either none, one, or both of the interferers throughout the course of the experiment. The middle of Figure 4.12 shows the delivery ratio achieved

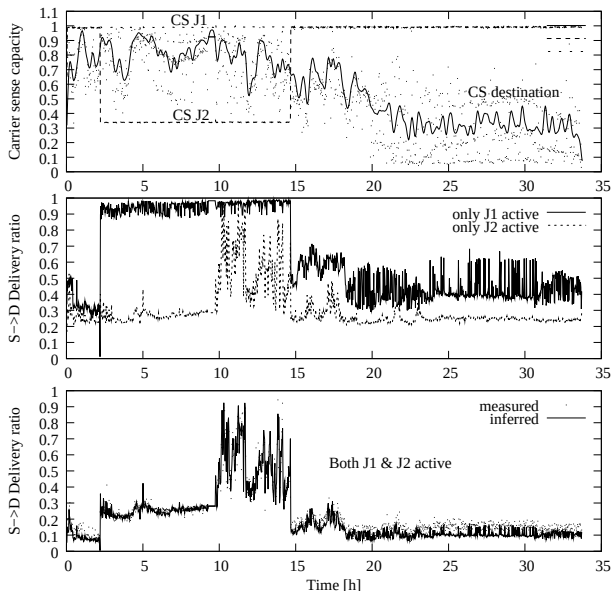


Figure 4.12: Top: History of access to the medium for the interferers and the destination. Middle: delivery ratios sampled independently for each interferer. Bottom: When both interferers are active, measured delivery ratio confirms the independence of the two interferers .

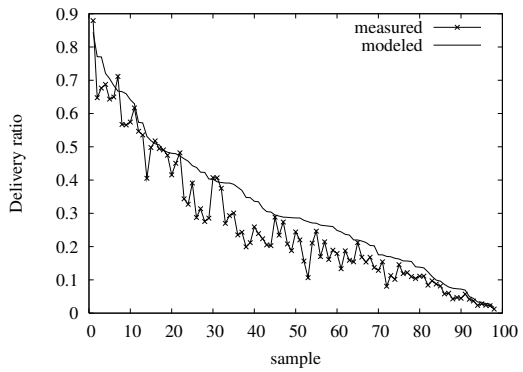


Figure 4.13: A source and two interferers send data at random rates - we sort the experiments based on delivery ratio achieved. Using equation (4.2), we predict the delivery ratio based on separate measurements for each interferer. The correlation is quite high (0.97) but the model overvalues the delivery ratio by about 4.5%.

with each interferer independently, ranging from 20% to 100% depending on the time, and position of the interferer. We chose this scenario from a larger set of experiments with similar placement of nodes because its diversity makes it appropriate for verifying the independence assumption between the two interferers.

In the bottom figure, we plot the measured delivery ratio with both interferers active, together with the product of the delivery ratios measured separately. We see that the modeled value is highly correlated (0.96) with the measurement, indicating real independence. More than actual correlation, the value inferred as the product of the separate delivery probabilities closely tracks the measured delivery probability with J1 and J2 active.

Finally, we verify the accuracy of all aspects of equation (4.2) in a 3.5 hour long experiment that is similar to the one in the previous paragraph, except that both the rate of the source and the rates of the interferers are selected randomly and independently in the interval $[0..1]$, from a uniform distribution. The delivery ratio for the non interfered link is around 95% for the entire length of the experiment, but with the presence of both interferers, it can drop below 5% as seen in Figure 4.13. The analytical model using delivery ratios measured for independent interferers $d^k_{i,j}$ closely follows the values measured for two simultaneous interferers. Although it tends to overestimate the delivery ratio for higher values, it has a good correlation (0.97) with the measured data. For the entire duration of the experiment the interferers are out of the carrier sense range of each other and of the receiver. After each interference measurement, a separate broadcast of both interferers and the receiver was run to confirm that the interferers are able to send at maximum throughput, therefore do not defer to the receiver or each other. The receiver however is carrier sensing one of the interferers for the duration of the experiment. This shows that a combination of interferers can be modeled with around 5% error in delivery ratio when only measurements for individual interferers are available.

After the validation of equation (4.2), which is the main way of reducing the measurement complexity of the interference map for one channel, we turn to other aspects of interference: first, we show that an approximation of the cs_{ij} structure can be inferred if positions of the nodes are known. Second, we look at the complexity of the interference map for the multiple card case.

4.3.2.4 Correlation with distance

Since producing the interference map has high cost ($O(n^2)$ network down time), it would be desirable to produce at least a good approximation of it by some cheaper method. Knowing that in theory interference range is linked to the communication range, we want to see how predictable the interference is for our particular indoor setup with respect to distance. In most static networks when access points are deployed in a building, a map association is usually available so that distances can be estimated with reasonable precision from a map drawn at scale. We used a map of our building as the one shown in Figure 4.2, and assigned coordinates to each node based

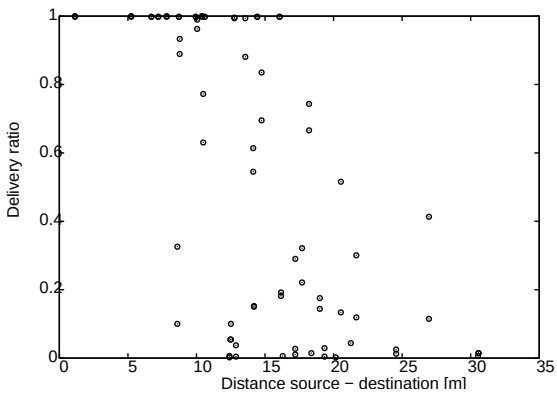


Figure 4.14: Delivery ratio is weakly correlated with distance.

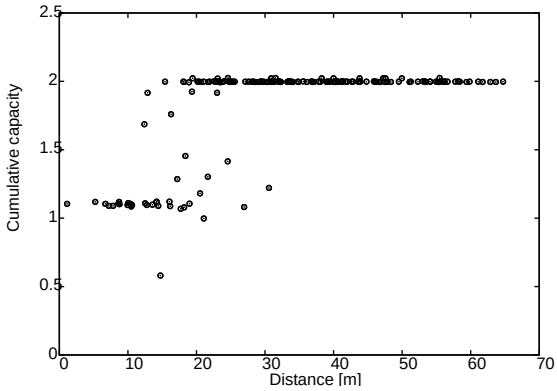


Figure 4.15: Carrier sense depends strongly on distance.

on its relative position, and using the known dimensions of the building. We then associated each measurement d_i^k with the distance between i and k. In Figure 4.14, we see that delivery ratio is very weakly correlated with distance, and this corroborates well with other findings in literature showing that delivery ratio (and also signal strength) and distance do not correlate well [77]. For carrier sense however, distance is a much better indicator - Figure 4.15 plots $cs_{ij} + cs_{ji}$, so that a value of 1 indicates carrier sense, while a value of 2 indicates independence. CS on 802.11a 6Mbps shows a well defined threshold at 18m for indoors 802.11a. This means that at least part of the $O(n^2)$ complexity can be avoided by getting an estimate of the CS graph based on the distances between access points or mesh nodes.

In Figure 4.16 we look at the relation between the damage produced by hidden terminals and distance. The damage produced to communication $i \to k$ by a hidden terminal j is computed as $1 - \frac{d_{i,j}^k}{d_i^k}$. Interferers in CS range of the source or destination are excluded. Although the amount of interference and distance are correlated (cor=-0.61), the correlation is not strong enough to produce a prediction based on distance. For example, for a hidden terminal at

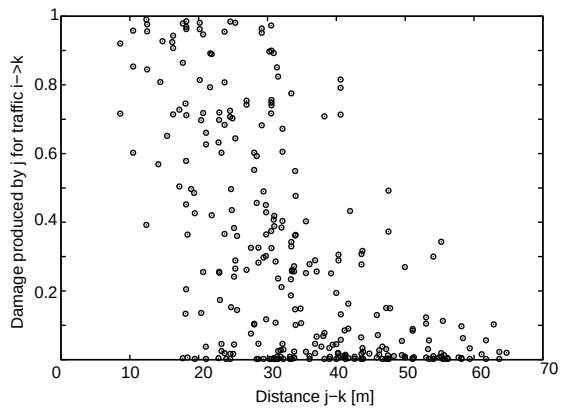


Figure 4.16: Effect of the hidden terminals is correlated with distance, but not strong enough for a prediction.

30m, we cannot really say what amount of damage it will cause. We also examined correlation of the damage with ratio of distances $(\frac{jk}{ik})^2$ as classical communication theory would indicate, but the obtained correlation is weaker (cor=-0.29).

As mentioned in section 4.2.2, the $O(n^2)$ complexity is driven by both measurements of carrier sense and of hidden terminals. The conclusion of these distance based statistics is that while we can infer a non-directional CS graph just by using coarse node positions, the hidden terminals effect is not sufficiently correlated with distance. Therefore, pairwise measurements would still be necessary to measure receiving interference, so in the process, they might collect the CS directional graph as well.

4.3.2.5 Consistency across interfaces

In all the experiments so far we considered that all nodes have one available wireless card, tuned to the same one channel for the entire mesh. For scalability reasons however, it is desirable that each node use several cards tuned to different channels. The allocation of channels and channel reuse will clearly affect the amount of interference, and this is one of the main targets of producing an interference map. A solution to channel allocation assigns a channel to each card so that connectivity is preserved and higher throughput becomes available. Several solutions have been proposed in the literature [40, 78, 79, 80, 81], but all implicitly assume that links may use any card in the same machine, implicitly assuming they are equivalent in terms of their interference patterns.

To verify this hypothesis, we measured delivery ratios for each pair of nodes in a group of 4 nodes for a total duration of 110 hours spread over a period of two weeks. Two sets of measurements were taken for two 'parallel'

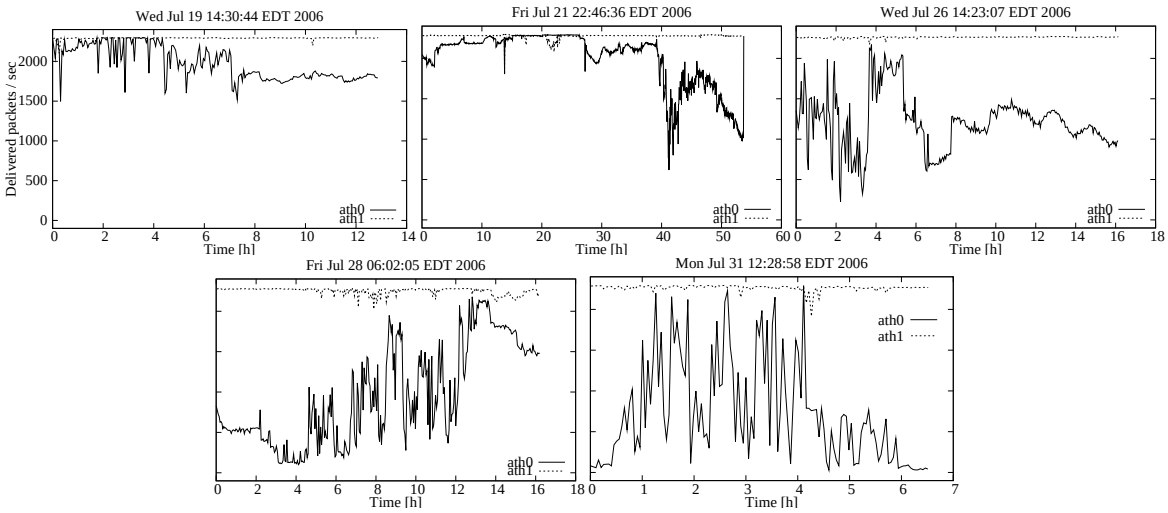


Figure 4.17: Delivery ratio sampled for a total of 110 hours spread over a period of two weeks. Delivery ratio differs widely across interfaces. Channel allocation algorithms may not assume equivalence of link performance based on sampling of links from a single interface.

networks - one created over *ath0* interfaces, and the other over *ath1*. In more than half of the links measured, the delivery ratio across one interface is completely different from the other one in both quantity and quality, even if the channel used is the same. In figure 4.17 we follow one particular pair of parallel links with series of measurements spanning the entire period, comprising of a mix of busy weekday mornings and quiet weekend nights. We can see that while the link across *ath1* interfaces shows solid performance across the entire period, the link across *ath0* interfaces ranges from acceptable to less than 10%, and from steady to highly variable in the samples taken. The differences in the other 11 directional links (obtained among 4 nodes) ranged from high variation to steady delivery and from maximum capacity to no link - in fact half of these links showed differences like those in Figure 4.17 or worse. Given that the wavelength of 802.11a is about 6cm, it is very likely that shadowing and multipath would create variation between points that are that close. Our antennas are spaced 40cm apart, and as the experiments confirm, there is very little correlation between the performances of cards in the same node.

These measurements show that between two communicating nodes, we cannot consider a logical link that can use any two pair of interfaces. In fact, each of the four physical links between two nodes has to be treated as a different link, and we conclude that f**or the purpose of the interference map, each physical link has to be measured separately**.

4.3.2.6 Consistency across channels

Another assumption made by channel assignment solutions that employ variations of edge or vertex coloring schemes is that carrier channels are basically equivalent - so they can be assigned any color (channel). This assumption also turns out to be too optimistic, at least for the case of 802.11a. In Figure 4.18, we examine a 28 hour trace of delivery ratios across channels 36, 44, 52, 64, 149, 157 and 165 under the same conditions used in previous experiments. The last three channels (recommended for outdoors) performed worse, all having an almost negligible delivery ratio, barely visible next to 0; channel 64 had 100% performance and is not visible at the top. Channel 44 also shows a strong and consistent performance maintaining 80%-90% delivery ratio throughout the day. 36 and 52 showed periods of stability mixed with periods of high variation, but even during the stable periods, the performance across channels differs greatly. The high variation experienced by channels 36 and 52 is all happening when other channels show steady (high or low) performance. In order to validate these results, we ran additional measurements for different sets of nodes, different power settings, and with longer settling time after the channel switch, but are not including them for the sake of brevity. As in the card comparison, the consistency across channels is rather the exception than the rule, with very few cases in which a link performs the same across all frequencies. We conclude that **for the purposes of interference map, possible channels of any link have to be measured separately.**

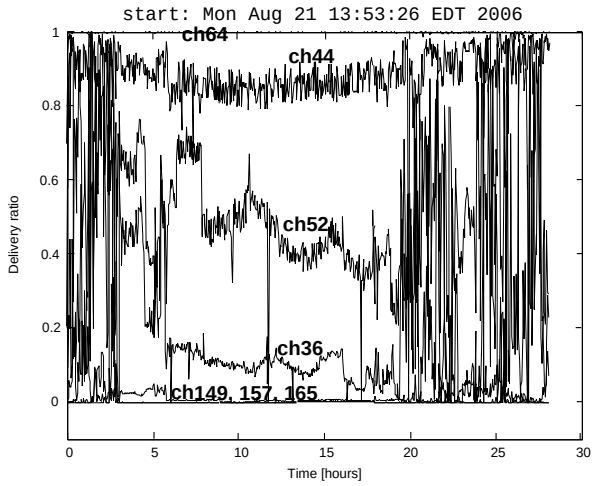


Figure 4.18: Delivery ratio differs widely across channels. Channel allocation algorithms may not assume that channels are interchangeable in terms of performance.

4.4 Discussion and summary

The most relevant works in this field are [68], [69] and [74, 75]. We extend the work in [68] by clearly defining the interference map as a collection of: delivery ratios, carrier sense matrix, and hidden terminal matrix and by modeling the effect of multiple interferers. We characterize the complexity of gathering the map, and propose an analytical model to allow the use of pairwise measurements. The model reduces measurement complexity by using certain properties of interference: linearity with respect to source rate, interferer rate, and independence of multiple interferers. [69] looks at remote interferers (out of CS, that produce damage), and 'no impact' nodes which produce no damage. Their numerical results are properly captured by our model: remote interferers – can be combined in a linear fashion using relation (4.2); 'no impact' nodes – although they may become interferers when acting together, the occurrence is very rare. In addition, we show that the occurrence of remote interference is quite severe, phenomenon which is prone to rise with increase in deployment density. The remote interferers are ignored in [74, 75] by considering only the ones from which signal strength can be read. Signal strength however can only be used when packets are received properly, which means inside communication range. We model interferers outside communication range, which can be inside CS for sending interference, or inside interference range for remote hidden terminals (these are the causes of the $O(n^2)$complexity).

However, our model also has a few drawbacks:

- requires a global view of the network. This stems from the fact that interference has non local effects which we believe are best tackled in a global, centralized manner. However, many measurements can be performed in a distributed, asynchronous fashion. For example, delivery ratios d_i^k between nodes can be monitored passively on live traffic. Carrier sense and hidden terminal measurement require network down time to obtain d_{ij}^k, but they can be performed one triplet at a time, during periods of relative silence in an area between i and j.

- only models CBR traffic. This limitation is an attempt to eliminate the time factor from the model. Because each flow of traffic is a potential creator of interference somewhere else in the network, non constant flows such as TCP would create highly variable interference patterns for otherwise steady conditions. Voice networks handle only CBR traffic, and also have stringent loss requirements, so are good candidates for the controlled interference environment proposed here.

- is verified extensively only for 802.11a networks. If density of WiFi devices increases at the current pace, one way to increase bandwidth per area is by using more independent channels, and more cards. While 802.11b allows for only three orthogonal channels, most chipsets also suffer from electrical interference so that two cards must be at least 60cm apart, regardless of the channel. Depending on regulations, in 802.11a there are 12 orthogonal channels available and we found the electrical interference to be almost negligible.

- requires network down time. The very core of measuring interference is to quantify the effect other nodes have over a particular communication. Therefore any uncontrolled traffic has the potential of skewing the effects produced by an interferer: it can either increase the packet drop at the destination, or it can have the opposite effect, by contending for the medium with the interferer, and therefore produce a better delivery ratio at the destination.

The main factors driving the complexity of measurement of the interference map are the pairwise style measurements, and the asymmetries of the cards and channels. Pairwise measurements are the direct way of determining $d_{i,j}^k$ (the delivery ratio from i to k when j is interfering) when j is not in contact with either k or i. Since no packets from j can be received at k or i, no delivery ratio, or

signal strength can be employed to determine the potential damage j can produce. In this case, generating traffic from j is a reliable, albeit costly way of gauging its effect over $i \rightarrow k$ communication. Obviously, for networks spreading over a wide area, only nodes in a circular region around the receiver are candidates for being remote interferers, so the complexity in these cases instead of $O(n^2)$, becomes just $O(n)$ - with a constant depending on the size of interference range. However, for setups that are small in area, but large in the population of wireless nodes, the potential interferers can be in a large fraction of the network. In our testbed, about one third of the nodes can interfere, so effectively, the complexity is still $O(n^2)$ – for the single card case.

The second cause of complexity is the asymmetry in card / channel behavior: we conclude that a more accurate estimation of the time complexity to obtain the interference map for all channels and across all possible links should be adjusted to $O(fcn^2)$ where c is the number of cards, and f the number of available orthogonal channels (the total number of experiments would be $O(fc^2n^2)$ but a node can run c at a time when $f \geq c$). This is a considerable difference from the original $O(n^2)$, given the desirability of large number of cards to make use of the channel parallelism and decreased range/ increased density of 802.11a. To put this into perspective, a 20 node network, one card, one channel, 20 second measurements and associated overheads required for our experiments about 2.5 hours of network downtime - corresponding to $O(n^2)$. If we consider for example the case of a dual card 802.11a node ($f = 12$ and $c = 2$), the required downtime becomes unacceptable.

4.4.1 Summary

We proposed a model for interference in dense wireless networks that enables a low complexity procedure to collect the interference map in one card networks. We confirmed experimentally that interference measurements for isolated triplets of nodes (source/destination/interferer) can be used to predict the damage from several simultaneous interferers. The interference from distant interferers behaves linearly with respect to rate of the source and rate of the interferer, and shows independence between interferers. The most important result is that behavior of complex interference scenarios can be estimated based on measurements that have relatively low complexity $O(n^2)$, which could otherwise depend exponentially on the group size. On the negative side, measurement of the interference map faces asymmetries in the card and channel behavior, which make the complexity still prohibitive for dense multiple card networks.

Chapter 5

Mobile antennas for WiFi links

5.1 Introduction

It is an established fact that radio signal reception varies in both space and time. Spatial variation is due either to user mobility or to scattering effects. As they move, mobile users experience signal and coverage changes due to fading and multipath effects. Reception varies with position and speed of the antennas, and with the quality of the environment. But even for static users, there is spatial variation due to multipath, and temporal variation due to changing patterns of human activity (indoors), or atmospheric conditions (outdoors). In all these cases, wireless links experience degradation which translate in lower QoS for the users. Long term outdoor point to point links require manual tuning and maintenance, while indoor links require site surveys or over-provisioning to account for this variability in link quality.

Use of multiple antennas on the senders and receivers (MIMO) has been a way to increase capacity and resilience. However, when operating on the same carrier wave, these antennas interfere with each other, and the achieved channel does not always have a high rank. The rank of the achieved channel depends on the actual deployment geometry of the sender and receiver, and on the environment in between, all of which determine the correlation of the multipath signal. One class of approaches to improve performance is to use adaptive coding and modulation techniques that are tailored for the given channel [82]. Another direction is to increase the diversity of the achieved channels [83].

One method is to increase the spatial diversity, and MIMO itself implies spatial diversity by use of multiple elements. Spacing of elements is a subject of research and values between 0.1λ and 10λ are deemed appropriate for different scattering and SNR conditions. A second way of improving diversity is the use of different polarization. This decreases mutual coupling between close by elements, and increases the likelihood of uncorrelated paths. Pattern diversity is yet another way of minimizing the correlation of achieved channels. Usage of different patterns, or directional antennas is a technique that works for beamforming, and for SISO systems as well.

In this article we explore yet another way to increase the diversity of the channels, by mechanically changing the position of the antenna elements. This adds another degree of freedom to the joint optimization of coding, modulation, and diversity techniques that have been used so far. In highly scattered indoor environments the quality of the signal from a source may change on a scale less than the carrier wavelength [84]. This is generally seen as a degrading factor for indoor wireless, as it induces coverage dead zones and unexpected variability. We exploit this existing diversity in propagation to find antenna element spots that produce 'better' channels.

For the SISO case, these are simply channels with better delivery ratios. The advantages of a reconfigurable antenna are that:

- it requires no manual intervention or technical expertise.
- adapts to changes automatically.
- can be retrofitted to existing antenna technology.
- is low cost - only a servomotor, and a controlling algorithm.

For the MIMO case, 'better' channels mean less correlated for the purpose of increased capacity. The advantages are:

- mobile antenna elements are a complementary technology, and it can be coupled with most other diversity techniques mentioned.
- the channels that can be obtained are simply not available to traditional MIMO techniques that only optimize using weights, phases, or gains.

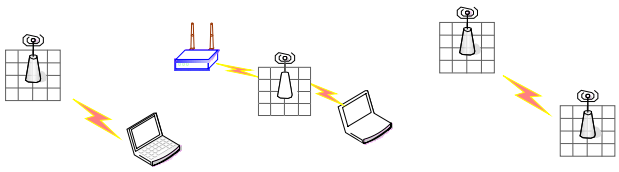


Figure 5.1: Applications for mobile antenna/mobile element technique: at access point, for mostly static clients; at relay points to optimize both links; at long term point to point outdoor links.

- antenna element position and coding/modulation are both optimization techniques. They can both make use of feedback to each other to improve the end to end channel.

There are a number of applications where a mobile antenna can be easily implemented (Figure 5.1):

- ceiling mounted access points that have motors to allow changing of the antenna positions on a centimeter scale. They can automatically tune to provide better service to user populations that are mostly static.
- in relays: adaptation of uplink and downlink could enhance relay performance.
- long term point to point links that require periodic adaptation and optimization.
- mobile antennas could be implemented on larger mobiles (laptops) as well, as they are usually embedded in the screen, therefore providing a large search space.

For both SISO and MIMO we experimentally show that due to the diversity of the indoor signal, there is ample opportunity for optimization if antenna elements are mobile, even on a small scale.

5.2 Related work

The quest for improving MIMO performance through antenna enhancement is mainly based on improving diversity along three main directions: spatial, polarization and pattern. For a short review on various diversity techniques for MIMO antennas, see [83]. Reference [85] shows the advantages a reconfigurable antenna that can change its frequency and polarization. They claim a performance gain of up to 30dB over conventional fixed antenna MIMO. Reference [86] shows that adapting the antenna element spacing to the level of sparsity in the physical multipath environment has a profound impact on capacity. They claim that three canonical array configurations are enough for near optimum performance over the entire SNR range. Zangi [87] analytically investigates effect of antenna element geometry to the capacity of MIMO channels. For the 2×2 case, they find that spacing of antenna elements between $\lambda/2$ and 10λ are beneficial for low, respectively high SNR situations. For pattern diversity, Liang [88] explores ways to use antennas with dissimilar radiation patterns to induce decorrelation that could favor MIMO systems. They show that the achievable decorrelation is limited by the scattering environment. In her master thesis [89], Cotanda finds that using parasitic elements with small displacements ($< 0.4\lambda$) can have significant decorrelating effects. She found that the optimal element spacing was 0.1λ at the receiver, and 7.5λ at the transmitter for $SNR = 20dB$. [90] shows that an antenna consisting of two microstrip dipoles with variable electrical length, at a fixed $\lambda/4$ spacing can be used to increase capacity in an indoor environment, mainly for low SNR situations.

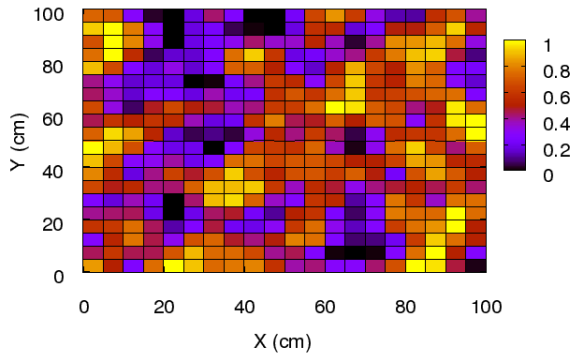


Figure 5.2: Packet delivery ratio is measured from a robot mounted access point that changes positions across a $1m^2$grid. The access point and the receiver are fixed for each measurement.

While most of existing work is either analytical, or through simulation, we aim at quantifying experimentally the gains obtainable through antenna mobility, and the scale of mobility that is necessary. In high scattering environments, particularly indoors, we have shown in previous work that because packet delivery ratios can vary wildly within distances as small as the carrier wavelength, wireless multihop paths can be optimized for increased capacity [84].

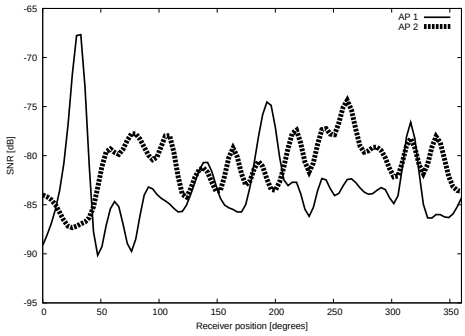


Figure 5.3: A client rotating around its own axis can find a signal up to $20dB$ stronger. Both access points and the client are fixed for each measurement.

5.3 Indoor signal variation across larger scales

Variation in signal quality across indoor spaces is experienced by most users of the popular unlicensed frequencies at 2.4GHz. To get a sense of the amount of variation, we used a robot mounted 802.11g access point and recorded the packet delivery ratio (PDR) to a fixed client. Measurements were performed with the access point assuming different position at grid points across a patch of $1m^2$ at carpet level. For 2.4GHz the corresponding wavelength is approximately $12.5cm$, and as shown in Figure 5.2, PDR can vary from 0% to 100% in distances within a few multiples of the carrier wavelength.

In a second experiment, we used the robot as a client, but restricted its movement to merely rotating around its own axis. The antenna is placed $20cm$ out of the axis of rotation. In Figure 5.3, we plot the RSSI of the received packets from two different fixed access points. At different angles of rotation the power of the signal received routinely varies with 10dB, but can vary as high as 20dB. The pattern of variation depends on the particular features of the space between the AP and the receiver, and different APs are likely to produce different patterns.

These examples show that even with small antenna displacements it is possible to find a better channel. The size of the searchable space depends on where the functionality is implemented - on the access point, or on the receiver. While the access point may be larger and offer more potential for optimization, a client has usually a lesser dimension in the degrees of freedom. In laptops, which exhibit lower mobility patterns, antenna is mounted in the screen, so a degree of freedom can be achieved with a simple translation of antenna elements.

5.4 MIMO Implementation

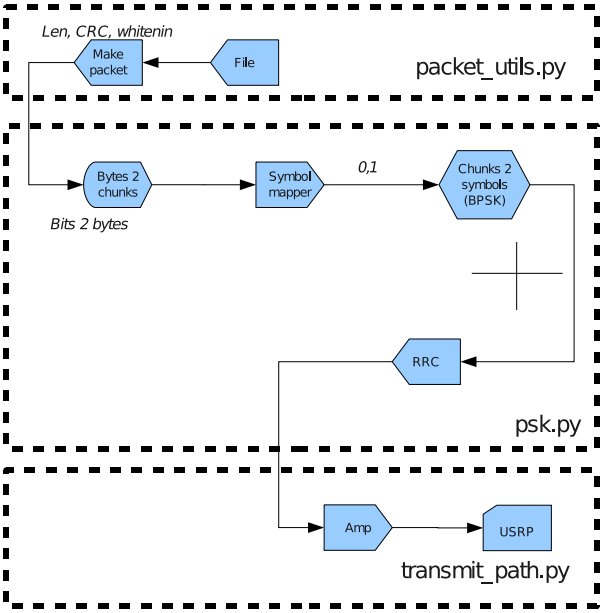
Having established that high signal variation can be found on a scale of the carrier wavelengh, we now look at how a MIMO system can take advantage of creating uncorrelated channels. We aim at measuring performance with full diversity sending of two independent streams on a 2×2 system, $0 \rightarrow 0$ and $1 \rightarrow 1$.

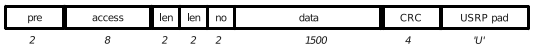
We used GNUradio [91] libraries, driving an USRP 1 board equipped with two 2.4GHz daughter-boards. The MIMO system described here is assembled from SISO examples that came with the library. A block diagram of the sender is shown in Figure 5.4. For the purposes of measuring the channel in a 2×2 MIMO system, we haven't built a full fledged MAC, but a more basic system system that is lacking any multiuser capability such as carrier sense, acknowledgments, etc. The system could accept data from a file, or generate data on the fly, that is then encoded using BPSK, filtered through a root-raised cosine, and then sent to the antenna, with the desired amplitude. To facilitate computing of BER at the destination, we use a simple framing scheme in which the first 10 bytes are used to detect the beginning of the frame. Out of these 80 bits, we only consider frames that match in at least 77 positions for the computation of BER at the receiver. The next fields contain the length of the data portion of the frame, and the antenna used at the sender (0x00 or 0xFF). These are only used for reference since we know that our independent streams are sent $0 \rightarrow 0$ and $1 \rightarrow 1$. Data of up to 1500 bytes is then trailed with a CRC code, and padding required for transferring across the USB to the USRP board. This picture is repeated for each antenna at the sender, and the streams are then interleaved before being sent to USRP. There they are separated and sent to the respective antennas.

For the receiver (Figure 5.5), the complementary blocks are present: BPSK demodulator, preceded by the RRC filter, and automatic gain control. Again, we have two such chains, one for each antenna. Due to oscillator and phase shifts between the sender and the receiver, we had to employ a Costas loop before the demodulation.

5.4.1 BER Computation

BER computation is done off-line, using a combination of C and GNU Octave code. The 80bits of the preamble are used to detect the beginning of the frame, and the body of the frame is then used to compute BER when CRC fails. Since stream 0 is always sent from antenna 0 for the des-


Len, CRC, whitenin
Make packet
File
packet_utils.py
Bytes 2 chunks
Bits 2 bytes
Symbol mapper
0,1
Chunks 2 symbols (BPSK)
RRC
psk.py
Amp
USRP
transmit_path.py


pre
access
len
len
no
data
CRC
USRP pad
2
8
2
2
2
1500
4
'U'

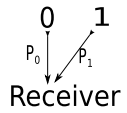


Figure 5.7: Each of the sender antennas emits a different tone - and their power is recorded at the destination antenna.

is employed before decoding. To capture the gain of the channel, we read the power P_{00} at the entry of AGC. In the second run, we send a stream on antenna 1 and keep antenna 0 inactive in order to measure h_{11} and h_{10} with a similar procedure.

In a third run two independent streams are sent from each antenna and the channel estimation is used at the receiver to combine the symbols from the two antennas. The MIMO channel is then assembled in the following manner:

$$
\begin{aligned}
y_0 &= x_0 h_{00} + x_1 h_{10} + n_0 \\
y_1 &= x_0 h_{01} + x_1 h_{11} + n_1 \\
y &= Hx + n \\
x &= (H^T H)^{-1} H^T y
\end{aligned}
$$

The matrix $W = (H^T H)^{-1} H^T$ gets y symbols from each antenna and feeds the x values obtained further to the slicer that performs the hard decision. After each of the three runs, BER is computed as described in the previous section.

5.4.3 Indoor signal variation across small scales

In this section we quantify the performance obtained by exploring diversity of antenna positions over small spaces. For these experiments, spacing of the sender antenna is fixed at 1.5λ, whereas for the receiver we move one or both elements, but they are separated by 4λ on average. Sender and receiver are within LOS, at 2.5m of each other, in a typical office environment.

In order to identify good orthogonal channels, we send two different tones of equal power from each of the antennas and measure the power at each receiver antenna (Figure 5.7). For antenna 0, the recorded powers would correspond to P_{00} and P_{10}. An FFT of the spectrum visible at each receiver antenna is shown in Figure 5.8. The origin of each tone of interest is labeled using a '0' or a '1' in the figure. In this example, the power difference at receiver 0 between the signals from the sender is of 9dB

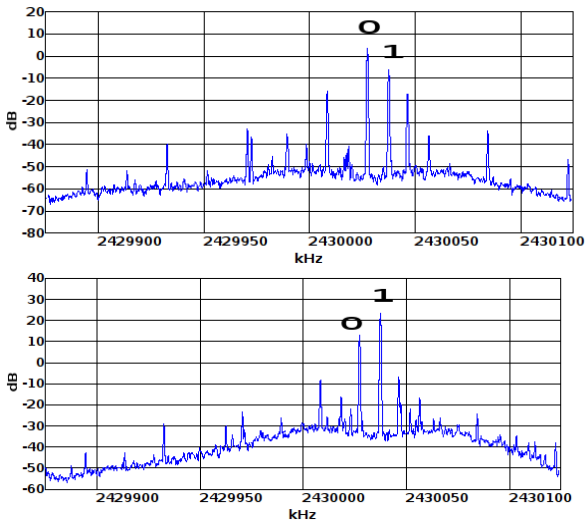


Figure 5.8: FFT of two tones as seen at antenna 0 (top) and antenna 1(bottom). The tones are sent from the antenna 0 and 1 respectively at the sender. This is an uncorrelated, high capacity MIMO channel.

in the favor of 0→0 signal over 1→0 (upper Figure 5.8). In the lower figure, we see the reading of the same two tones at receiver antenna 1 - now the 1→1 signal is 10dB stronger than 0→1. This provides a good isolation between 0→0 and 1→1, but in most situations, channels are not as orthogonal as in this example.

We measured the ratio of the power of the two tones P_0/P_1 for different positions of one receiver antenna, and summarized the results in Figure 5.9. It is only necessary to explore this space for one receiver antenna, as the gain difference would be the same if we measure it with another (second) receiver antenna. The area explored is about $500cm^2$. The top figure shows a 3D view of the power difference for each point, with a range of -17dB to +24dB. The lower part of the figure shows a histogram of all the differences measured in the explored space. Out of a total of 144 positions, 16 exhibit an absolute difference of at least 10dB between P_0 and P_1.

Using positions identified using the above procedure, we compute the BER for different amplitudes of the signal at the sender. In Figure 5.10 we look at the performance of two such realizations. The top one is for an uncorrelated channel, like the one identified in Figure 5.8, which achieves 200% capacity compared with the SISO case, for the same power used per sender antenna. The lower figure one corresponds to a channel which has some correlation, the power difference at the two receiver antennas being only about 5dB.

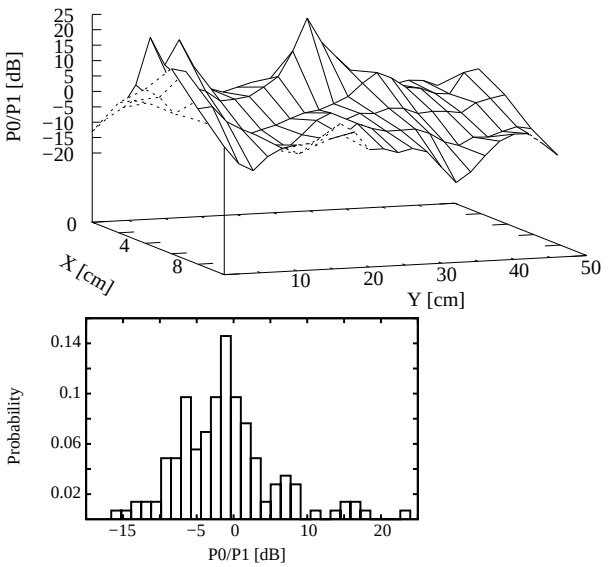


Figure 5.9: Upper: gain difference at the receiver antenna between the two sending antennas. The receiver antenna takes different positions one grid spanning a $500cm^2$area. Lower: Power difference distribution histogram. 11% of the points exhibit more than 10dB absolute difference in the power received from the two sender antennas.

5.5 Summary and future work

Antennas with mobile elements can achieve better performance for both SISO and MIMO . This is especially true indoors for the popular 2.4GHz carrier, where a difference of a few centimeters reaches a completely different channel. On a large scale of movement (above 1λ) the method is applicable to ceiling mounted access points, long term links, and the gains can be made available to any existing antennas. On a small scale, the idea can be used to achieve higher capacity in MIMO systems by finding channels with high rank correlation matrices. This work can be continued in quite a few directions:

- more extensive evaluations for different LOS/NLOS conditions, carrier frequencies, and bandwidths. Since the high variation per unit of distance is produced by indoor fading and multipath, it is likely to be available for other carriers, and NLOS conditions. The use of spreading however, might complicate the search procedure under frequency selective fading conditions.

- apply the same principle of “finding” good channels for the problems of beamforming, multiple users, and canceling interference. Any problem that involves a form of channel estimation and then optimization can benefit from an extra parameter to drive the optimization process.

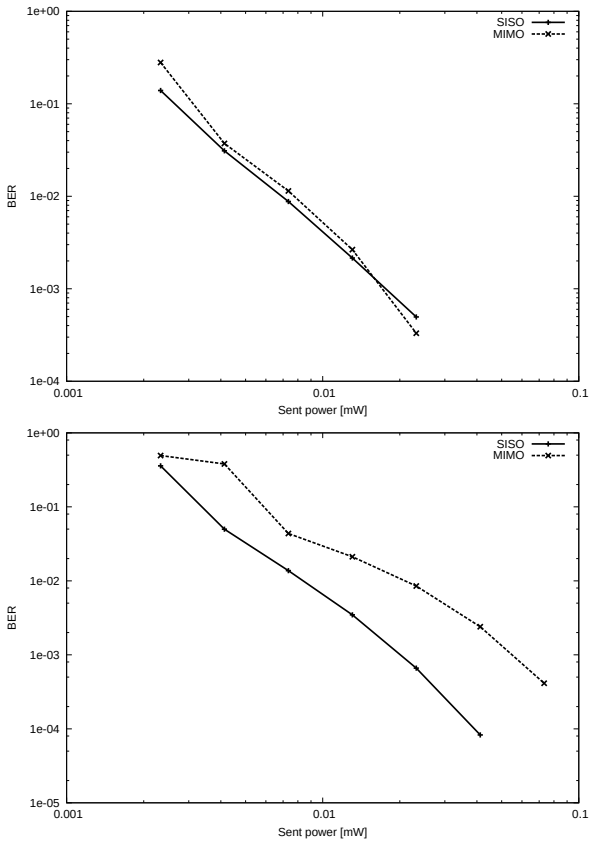


Figure 5.10: BER plot for a decorrelated channel (upper), and for a channel with some correlation (lower).

- actual algorithms to explore the space of antenna positions, depending on the capabilities. They would have to consider the scalability with number of antennas, the degree of mobility, which brings a tradeoff between the time to optimize and the time for the channel to remain stable.

- antenna position adaptation and coding/modulation adaptation can feed back to each other to achieve the optimum for a long term link. Essentially different element positions offer (almost) different channels, each of them with a different optimal coding and modulation strategy. Reciprocal feedback between these optimization processes offers the possibility of even higher gains.

Chapter 6

Evolution and Development of Academic Career

Based on my past experience and on the research and teaching directions detailed below, my short term goal is to build a research team that can accumulate expertise in most areas related to mobile computing. All my research and teaching activities are closely intertwined, and this could be a catalyst for the team to become competitive in applying for national and European funding.

The long term goal is to lead a group that could produce high quality research - published in competitive venues and highly cited, but could also produce prototypes and experimental systems to bridge collaboration with the industry. I am presently contributing to two FP7 projects collaborating closely with foreign industrial partners (France Telecom, Thomson), and in parralel, I am applying for funding in parnership with R&D oriented domestic companies (Intel).

Research Directions

The mobile computing area is highly active in both research and industry, and is mainly fueled by two factors. The first one is that mobile communications see a lot of innovation compared with other areas of communications. Recent interest surges in sensor networks, Zigbee, Bluetooth, RFID, UWB, WiMAX and LTE, 60GHz and WirelessHD, all point to an area that is continuously changing. The lower part of the OSI stack sees many changes, and this usually comes with a lot of research challenges as well. Users expect mobile networks to behave as an extension of the better understood wired networks, but they usually don't. The problems involved are sometimes wireless specific (interference, density), sometimes technology specific (VoIP overhead, competing flows, channel allocation), and other times just due to mobility which only now becomes pervasive. The second factor is the trend of higher powered, lower cost, pervasive mobile devices. These are bound to produce large amounts of diverse traffic. In fact, 3G operators claim that the increasing population of devices produces large amounts of signalling traffic that is not controlled directly by the users. At the upper levels there is also a lot of innovation in the application area, supported by multiple application distribution venues (app stores). These two regions of high innovation, at the top, and at the bottom of the OSI stack, put a high pressure on the classic and somewhat ossified layers that provide the core of the hierarchy - network and transport (namely IP and TCP).

Based on these trends, and on accumulated expertise in the area of mobile networking, my research plans presently include the following directions: multipath TCP, cognitive WiFi, and location based services. Each of these themes are large areas in their own right, and each of them have enough open problems to accommodate several PhD theses.

Multipath TCP

Host mobility has traditionally been solved at the network layer, but even though Mobile IP has been standardized for 15 years, it hasn't been supported by operators. IP's double role as a location identifier and communication endpoint identifier brings a number of functional and performance problems. Several researchers argue that the best place to handle mobility is at the transport layer. While this is not a new argument, I believe that the emerging standard of Multipath TCP (MPTCP) can be used to solve many issues related to mobility. MPTCP naturally implements **make-before-break**, can be incrementally deployed, is backwards compatible with TCP, and could even ease incremental adoption of IPv6.

On the surface, MPTCP is an update to TCP that al-

lows for a TCP flow to use multiple interfaces. This allows a machine to use all available interfaces for faster network access, but there are several new functionalities enabled by a dynamic list of interfaces. An interesting use for the datacenter is to expose several alternate paths in the topology through virtual interfaces. MPTCP could then naturally use several available paths just based on its end-to-end implementation. For the mobility, the communication endpoint could now be identified by the MPTCP socket. The IP of the interfaces used will be left as an identifier for the location, therefore solving the problems that were circumvented by Mobile IP. A MPTCP connection could start life on some interface, in some network, and continue to live through other interfaces, but, most importantly across different networks.

This research has two important implications. First, there has been a lot of work about vertical handoff for various technologies. Many of these problems are now naturally solved with MPTCP by migrating a connection across interfaces - a technology specific solution is not necessary (except perhaps use of MIH 802.21). Second, even at layer 2 there are ways to enhance handoff using MPTCP, through simultaneous association with several access points so that the old connection could be used till the new one becomes active. Third, in heterogeneous mobile networks, there is the possibility of power saving by transparently toggling interfaces based on their perceived cost per bit.

Cognitive WiFi

The 1997 WiFi standard allocated 3 independent channels in the 2.4GHz band and 12 channels in the 5GHz band (depending on country). In retrospect, this was obviously not sufficient, but the success of the WiFi technology, based on its high spectral efficiency, and its reduced cost would naturally lead to its implementation in other areas of the spectrum. Dynamic Spectrum Access (DSA) is therefore expected to play a key role in increasing the capacity of consumer WLANs by improving spectrum-usage efficiency. WLANs equipped with DSA-capable nodes or secondary users (SUs) can opportunistically utilize licensed channels when they are unused by their primary users (PUs). During such unlicensed usage, it is critical to ensure safe (low-interference) communication of PUs while ensuring high channel-use efficiency of SUs for achieving good coexistence. A simple approach to guaranteeing PU-safety is that the time for PUs to be interfered with by SUs is bounded by a prespecified limit. However, this simple approach limits the application of DSA to spectrum regions that exhibit slow-varying ON/OFF PU behavior, leaving a large fraction of licensed spectrum difficult to accommodate to DSA.

This research direction aims at identifying avenues for coexistence of the popular WiFi standard in spectrum areas that were not previously allocated to it. There are several open problems stemming from the availability of extra spectrum. One challenge is how to create hard limits in time for a CSMA based technology, like 802.11. Another is how to adapt known standard procedures to the large number newly available channels. For example, in 802.11b/g/n sweeping 11 channels accounts for most of the time of the handoff. In a multi card multi hop topology coloring procedures usually take into account each channels radio conditions (traffic, interference, channel quality). All these procedures need to be adapted with a high increase of number of channels, many of which may be vacated at a short notice.

Location based services

As mobiles become more pervasive and more important to daily life, their interface with the user and with the network becomes more critical. When it comes to data input, IO to and from the network, today's mobiles most of the time behave as regular computers, that is, require extensive attention from the user. One avenue to explore here is that of accelerometer usage to be used for data input, or for changing the state of the phone. Gesture based input translates movements sensed by the accelerometers and gyroscopes in today's smart phones into data that can be input for the phone: drawings, SMSs, or shortcuts through the interface.

A second direction is that of using wireless for providing a context dependent medium that can be used for reading and fetching documents "from the air". Using new techniques used for indoor positioning (radio fingerprints), it is possible to associate the state of a user with a certain location in signal space. This allows some users to post documents at certain location, where others can retrieve them. Other application of radio fingerprints include access control, and authentication with reduced interaction.

In a larger context, current problems in mobile networks are just prefigurations of the problems that will be faced in the near future in a world pervaded by mo-

bile computers (mostly small, heterogenous and unorganized). I am also interested in how to manage so many entities with which individual interaction and administration is not humanly possible. My recent work addresses some of the issues, such as positioning, routing, and interference management, but many remain unsolved, e.g. mobility, diversity of the devices, diversity of the spectrum, replenishing their energy supply, reconfiguring for different purposes, management costs, and robustness in autonomous operation.

Teaching Directions

My teaching experience, both at Rutgers University and at Politehnica University of Bucharest, has an extensive breadth, ranging from discrete mathematics, computer networks, databases, and operating systems. As a teaching assistant at Rutgers for systems oriented courses,like "Databases", "Internet Technology", and "Operating Systems", I was responsible, besides teaching recitations and labs, for designing projects, midterms, and occasionally giving full length lectures.

At Politehnica, I am teaching the "Architectures and Protocols for Communications" course for undergraduates, "Transport and Access Technologies" and "Metropolitan and Rural Networks" for masters students, all at ETTI (Electronică Telecomunicații și Tehnologia Informației). Currently, both graduate courses I am teaching involve study of recent high quality research papers. This gives students a chance to see how today's technologies are born, and how research results are produced. Many of these students are involved in my current research and have the chance of producing publications of their own.

For the future, I see the **development of a "Mobile Computing" curricula** as a high priority for the school. The skill set necessary for this discipline spans across networking, databases, and electrical engineering (electronics), and by its nature draws from several areas of technology. Based on current trends of increasing importance of mobile devices, the marketplace is likely to continue to require graduates highly specialized in this area.

Bibliography

[1] Dragoș Niculescu and Badri Nath. VOR base stations for indoor 802.11 positioning. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 58–69, New York, NY, USA, 2004. ACM. 15, 17

[2] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov. Performance of VoIP in a 802.11 wireless mesh network. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –11, april 2006. 15, 17

[3] Dragoș Niculescu. Interference map for 802.11 networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 339–350, New York, NY, USA, 2007. ACM. 15, 17

[4] Dragoș Niculescu. Finding MIMO. In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, pages 267 –272, may 2010. 15, 17

[5] Kyungtae Kim, Dragoș Niculescu, and Sangjin Hong. Gateway strategies for VoIP traffic over wireless multihop networks. *TIIS*, 5(1):24–51, 2011. 15, 17

[6] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM*, Tel Aviv, Israel, March 2000. 21, 31

[7] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, Guillaume Marceau, Lydia E. Kavraki, and Dan S. Wallach. Robotics-based location sensing using wireless ethernet. In *ACM MOBICOM*, Atlanta, GA, September 23–26 2002. 21, 31

[8] Moustafa Youssef, Ashok Agrawala, and Udaya Shankar. WLAN location determination via clustering and probability distributions. Technical report, University of Maryland, College Park, March 2003. 21

[9] P. Krishnan, A. S. Krishnakumar, Wen-Hua Ju, Colin Mallows, and Sachin Ganu. A system for LEASE: System for location estimation assisted by stationary emitters for indoor rf wireless networks. In *IEEE Infocom*, Hong Kong, March 7–11 2004. 21, 29, 31

[10] Kamin Whitehouse. The design of calamari: an ad-hoc localization system for sensor networks. Technical report, University of California at Berkeley, 2002. Master's Thesis. 21

[11] Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, March 1964. 26

[12] http://www.octave.org. 26

[13] Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek Patil. Landmarc: Indoor location sensing using active RFID. In *IEEE International Conference in Pervasive Computing and Communications 2003 (Percom 2003)*, Dallas, TX, March 2003. 29, 31

[14] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. In *ACM Transactions on Information Systems*, volume 10, pages 91–102, January 1992. 30

[15] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *ACM MOBICOM*, pages 59–68. ACM Press, 1999. 30

[16] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *ACM MOBICOM*, Boston, MA, August 2000. 31

[17] N.B Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The cricket compass for context-aware mobile applications. In *ACM MOBICOM*, Rome, Italy, July 2001. 31

[18] Kay Römer. The lighthouse location system for smart dust. In *ACM/USENIX Conference on Mobile Systems, Applications, and Services (MobiSys*

2003), pages 15–30, San Francisco, CA, USA, May 2003. 31

[19] J. Hightower, G. Boriello, and R. Want. SpotON: An indoor 3D location sensing technology based on RF signal strength. Technical Report CSE Technical Report 2000-02-02, University of Washington, February 2000. 31

[20] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low cost outdoor localization for very small devices. In *IEEE Personal Communications Magazine*, Special Issue on Smart Spaces and Environments. October 2000. 31

[21] A. Savvides, C.-C. Han, and M. Srivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM MOBICOM*, Rome, Italy, 2001. 31

[22] S. Čapkun, M. Hamdi, and J.-P. Hubaux. GPS-free positioning in mobile ad-hoc networks. *Cluster Computing*, 5(2), April 2002. 31

[23] Dragoş Niculescu and Badri Nath. Ad hoc positioning system (APS). In *GLOBECOM*, San Antonio, November 2001. 31

[24] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *International Conference on Intelligent Robots and Systems*, Maui, Hawaii, October 2001. 31

[25] L. Doherty, L. E. Ghaoui, and K. S. J. Pister. Convex position estimation in wireless sensor networks. In *IEEE INFOCOM*, Anchorage, AK, April 2001. 31

[26] Chris Savarese, Jan Rabaey, and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. Technical report, Delft University of Technology, 2001. 31

[27] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus Fromherz. Localization from mere connectivity. In *ACM MOBIHOC*, Annapolis, MD, June 1-3 2003. 31

[28] R. Yarlagadda, I. Ali, N. Al-Dhahir, and J. Hershey. Geometric dilution of precision (GDOP) bounds and properties. Technical Report 97CRD119, GE Research & Development Center, 1997. 31

[29] *Skype*. http://www.skype.net. 35

[30] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *ACM MobiCom*, pages 61–69, Rome, Italy, July 2001. 35, 36, 64

[31] S. Tao, K. Xu, A. Estepa, T. Fei, L. Gao, R. Guerin, J. Kurose, and D. Towsley. Improving VoIP quality through path switching. In *In Proc. of IEEE INFOCOM*, March 2004. 36

[32] Y. Amir, C. Danilov, S. Goose, D. Hedqvist, and A. Terzis. 1-800-overlays: Using overlay networks to improve VoIP quality. In *In Proc. of NOSSDAV*, 2005. 36

[33] J. L. Sobrinho and A.S. Krishnakumar. Real-time traffic over the IEEE 802.11 MAC protocol. In *In Bell Labs Technical Journal*, Autumn 1996. 36

[34] J.-Y. Yeh and C. Chen. Support of multimedia services with the IEEE 802.11 MAC protocol. In *In Proceedings of IEEE ICC*, 2002. 36

[35] M. Veeraraghavan, N. Cocker, and T. Moors. Support of voice services in IEEE 802.11 wireless LANs. In *In Proceedings of IEEE INFOCOM*, 2001. 36

[36] D. Hole and F. Tobagi. Capacity of an IEEE 802.11b wireless LAN supporting VoIP. In *In Proceedings of IEEE ICC*, 2004. 36

[37] S. Garg and M. Kappes. Can I add a VoIP call? In *IEEE ICC*, Anchorage, Alaska, 2003. 36

[38] J. Yu, S. Choi, and J. Lee. Enhancement of VoIP over IEEE 802.11 WLAN via dual queue strategy. In *In Proceedings of IEEE ICC*, 2004. 36

[39] W. Wang, S. C. Liew, and V. O. K. Li. Solutions to performance problems in VoIP over 802.11 wireless LAN. In *IEEE Transactions on Vehicular Technology*, 2005. 36

[40] Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In *ACM Mobile Computing and Communications Review(MC2R)*, volume 8, April 2004. 36, 41, 64, 73

[41] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *ACM MobiCom*, San Diego, CA, September 2003. 36, 64, 66

[42] J. Padhye R. Draves and B. Zill. Comparison of routing metrics for static multi-hop wireless networks. In *In Proceedings of ACM SIGCOMM*, 2004. 36

[43] R. G. Cole and J. H. Rosenbluth. Voice over IP performance monitoring. *ACM Computer Communication Review*, 31(2), April 2001. 36

[44] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, August 2000. 38, 67

[45] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, pages 234–244, New York, NY, USA, 1994. ACM Press. 39

[46] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the ieee 802.11 mac layer handoff process. *SIGCOMM Comput. Commun. Rev.*, 33(2):93–102, 2003. 39

[47] *Rude UDP packet generator*. http://rude.sourceforge.net/. 39

[48] Douglas De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *9th ACM MobiCom*, San Diego, CA, September 2003. 41, 68

[49] J.Li, C. Blake, D. S. J. De Couto, H. K. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *7th ACM Int. Conf. Mobile Comput. Netw.*, pages 61–69, Rome, Italy, July 2001. 48

[50] Kyungtae Kim and Sangjin Hong. Vomesh: Voice over wireless mesh networks. In *Wireless Communications and Networking Conferenc*, volume 1, pages 193–198, 2006. 48

[51] J. Yu, S. Choi, , and J. Lee. Enhancement of VoIP over ieee 802.11 WLAN via dual queue strategy. In *ICC*, pages 3706–3711, 2004. 48, 49

[52] W. Wang, S.C. Liew, and V. O. K. Li. Solutions to performance problems in voip over 802.11 wireless lan. In *IEEE Trans. Veh. Technol.*, pages 366–384. 48

[53] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy. Bandwidth estimation: metrics, measurement techniques, and tools. In *IEEE Networks*, volume 17, pages 27–35, Nov.-Dec. 2003. 48

[54] Xu K., Tang K., Rajive Bagrodia, Gerla M., and Bereschinsky M. Adaptive bandwidth management and qos provisioning in large scale ad hoc networks. In *IEEE Military Communications Conference*, volume 2, pages 13–16, Oct. 2003. 48

[55] Frank Y. Li, Mariann Haugea, Andreas Hafslund, Livind Kure, and Pal Spilling. Estimating residual bandwidth in 802.11-based ad hoc netowrks: An empirical approach. In *The Seventh International Symposium on Wireless personal Multimedia Communications*, Abano Terme, Italy, September 2004. 48

[56] A. Bakre and B.R. Badrinath. I-TCP: indirect TCP for mobile hosts. *Distributed Computing Systems, 1995., Proceedings of the 15th International Conference on*, pages 136–143, May-2 Jun 1995. 49

[57] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving TCP/IP performance over wireless networks. In *MobiCom '95: Proceedings of the 1st annual international conference on Mobile computing and networking*, pages 2–11, New York, NY, USA, 1995. ACM. 49

[58] Sherif M. ElRakabawy, Alexander Klemm, and Christoph Lindemann. Gateway adaptive pacing for TCP across multihop wireless networks and the internet. In *ACM MSWiM*, Torremolinos, Spain, 2006. 49

[59] Huan-Yun Wei, Shih-Chiang Tsao, and Ying-Dar Lin. On shaping TCP traffic at edge gateways. *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 2:833–839 Vol.2, Nov.-3 Dec. 2004. 49

[60] Hong Fei and Bai Yu. Performance Evaluation of Wireless Mesh Networks with Self-Similar Traffic. *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 1697–1700, Sept. 2007. 49

[61] R. G. Cole and J. Rosenbluth. Voice over IP performance monitoring. In *Computer Communication Review*, volume 31, pages 9–24, April 2001. 50

[62] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP throughput and loss. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 3:1744–1753 vol.3, March-3 April 2003. 51, 54

[63] S. Xu and T. Saadawi. Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks? *Communications Magazine, IEEE*, 39(6):130–137, Jun 2001. 51

[64] L. Xu and I. Rhee. CUBIC: A new TCP-Friendly high-speed TCP variant. In *Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDNet06)*, Feb. 2005. 52

[65] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A Compound TCP Approach for High-Speed and Long Distance Networks. *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006. 52

[66] Anand Kashyap, Samrat Ganguly, and Samir Das. Characterizing interference in 802.11-based wireless mesh networks. In *submitted for publication*, 2006. 55

[67] Kai Chen, Yuan Xue, Samarth H. Shah, , and Klara Nahrstedt. Understanding bandwidth-delay product in mobile ad hoc networks. In *Special issue on protocol engineering for wired and wireless networks, Elsevier Computer Communications*, volume 27, pages 923–934, 2004. 57

[68] Jitendra Padhye, Sharad Agarwal, Venkata N. Padmanabhan, and Lili Qiu. Estimation of link interference in static multi-hop wireless networks. In *Internet Measurement Conference*, 2005. 64, 66, 69, 75

[69] Saumitra M. Das, Dimitrios Koutsonikolas, Y. Charlie Hu, and Dimitrios Peroulis. Characterizing multi-way interference in wireless mesh networks. In *WiNTECH '06: Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, pages 57–64, 2006. 64, 65, 75

[70] K. Jamieson, B. Hull, A. Miu, and H. Balakrishnan. Understanding the real-world performance of carrier sense. In *ACM SIGCOMM E-WIND Workshop*, 2005. 64

[71] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan. Achieving mac layer fairness in wireless packet networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 87–98, New York, NY, USA, 2000. ACM Press. 64

[72] Murali Kodialam and Thyaga Nandagopal. Characterizing achievable rates in multi-hop wireless networks: the joint routing and scheduling problem. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 42–54, New York, NY, USA, 2003. ACM Press. 64

[73] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87, New York, NY, USA, 2005. ACM Press. 64

[74] Charles Reis, Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. Measurement-based models of delivery and interference in static wireless networks. In *ACM SIGCOMM*, Pisa, Italy, September 2006. 64, 69, 75

[75] Anand Kashyap, Samrat Ganguly, and Samir Das. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *ACM MOBICOM*, 2007. 64, 69, 75

[76] Giuseppe Bianchi. Surprises in experimental assessment of wireless LAN networks. In *EXPONWIRELESS*, Helsinki, Finland, June 2007. keynote speech. 70

[77] John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *ACM MobiCom*, pages 31–42, Cologne, Germany, 2005. 73

[78] Anand Prabhu Subramaniam, Himanshu Gupta, and Samir Das. Minimum-interference channel assignment in multi-radio wireless mesh networks. Technical report, Computer Science Dept, SUNY Stony Brook, 2006. 73

[79] Bhaskaran Raman. Channel allocation in 802.11-based mesh networks. In *IEEE Infocom*, 2006. 73

[80] A. Sen, S. Murthy, S. Bhatnagar, and S. Ganguly. Topology formation in multi-channel multradio mesh networks. Technical report, NEC Laboratories America, 2006. 73

[81] Mansoor Alicherry, Randeep Bhatia, and Li (Erran) Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 58–72, New York, NY, USA, 2005. ACM Press. 73

[82] S. Catreux, V. Erceg, D. Gesbert, and Jr. Heath, R.W. Adaptive modulation and MIMO coding for broadband wireless data networks. *Communications Magazine, IEEE*, 40(6):108–115, Jun 2002. 77

[83] A. Ali and S. Thiagarajah. A review on MIMO antennas employing diversity techniques. In *International Conference on Electrical Engineering and Informatics*, Bandung, Indonesia, 2007. 77, 78

[84] K. H. Kim, D. Niculescu, and K. Shin. Mars: A mobile autonomous router system for multi-hop wireless relays. Technical report, NEC Laboratories America, 2008. 77, 78

[85] B.A. Cetiner, E. Akay, E. Sengul, and E. Ayanoglu. A mimo system with multifunctional reconfigurable antennas. *Antennas and Wireless Propagation Letters, IEEE*, 5(1):463–466, Dec. 2006. 78

[86] Akbar M. Sayeed and Vasanthan Raghavan. Maximizing MIMO capacity in sparse multipath with reconfigurable antenna arrays. *IEEE Journal on Special Topics in Signal Processing*, 2007. 78

[87] C. Zangi and L. Krasny. Impact of transmit antenna array geometry on downlink data rates in MIMO systems. In *European Wireless*, 2007. 78

[88] Liang Dong, Hao Ling, and R.W. Heath. Multiple-input multiple-output wireless communication systems using antenna pattern diversity. volume 1, pages 997–1001 vol.1, Nov. 2002. 78

[89] E. C. Cotanda. Analysis of a reconfigurable antenna array for MIMO systems using parasitic coupling. Master's thesis, KTH Electrical Engineering, Stockholm, 2006. 78

[90] D. Piazza and K.R. Dandekar. Reconfigurable antenna solution for MIMO-OFDM systems. *Electronics Letters*, 42(8):446–447, 2006. 78

[91] GNUradio. http://www.gnu.org/software/gnuradio/. 79

Index