Technical Report TR-2006-018, Mathematics and Computer Science Department, University of Mannheim, November 2006
by Thomas King, Thomas Haenselmann, Stephan Kopf, Wolfgang Effelsberg

REIHE INFORMATIK

TR-2006-018

Overhearing the Wireless Interface for 802.11-based Positioning Systems

Thomas King, Thomas Haenselmann, Stephan Kopf, Wolfgang Effelsberg

University of Mannheim

– Fakultät für Mathematik und Informatik –

Praktische Informatik IV

A5, 6

D-68159 Mannheim, Germany

Overhearing the Wireless Interface for 802.11-based Positioning Systems

Thomas King, Thomas Haenselmann, Stephan Kopf, Wolfgang Effelsberg
{king,haenselmann,kopf,effelsberg}@informatik.uni-mannheim.de
Department of Computer Science
University of Mannheim

Abstract

Not only the proliferation of 802.11, but also the capability to determine the position of mobile devices make 802.11 highly appealing for many application areas. Typically, a mobile device that wants to know its position regularly performs active or passive scans to obtain the signal strength measurements of neighboring access points. Active and passive scanning are survey techniques originally intended to be performed once in a while to learn about the presence and signal reception quality of access points within communication range. Based on this survey the best suitable access point is selected as the gateway to the wired network. However, so far, no investigations are known to have been launched into how regular scanning affects concurrent data transmissions from an end-user point of view. In this paper, we explore how common data communication is affected while actively or passively scanning at the same time. We found that with an active scanning interval of less than 2 seconds the network conditions such as throughput and round trip delay are insufficient for interactive applications. The same is true for passive scanning if a scanning interval of less than 7 seconds is chosen. Furthermore, we present a novel scan scheme called *Monitor Sniffing* to reduce client service disruptions. Monitor Sniffing exploits the fact that 802.11 operates on overlapping channels by overhearing the wireless interface. We have implemented our Monitor Sniffing algorithm using commodity 802.11g hardware, and we demonstrate that it is faster than active and passive scanning and does not disturb concurrent data communication. Finally, our approach only requires software modifications on the client side, making the adoption process quite easy.

1 Introduction

During recent years we have seen considerable improvements in downsizing computer hardware and in increasing the capacity of rechargeable batteries, as well as the advent of wireless networks for the mass markets. These technologies allowed the manufacturers to build mobile devices that can be carried around and have a similar performance as desktop computers had several years ago. The benefit of mobile devices can be leveraged by so-called *location-based services*: Applications that act differently depending on the location of the user or, even better, proactively offer location-dependent information to the user, are currently a hot topic in research, and are considered to be a promising market.

Nowadays, the *Global Positioning System* [12] is the predominant outdoor positioning system. Whereas GPS works well in many outdoor scenarios, it suffers from obstacles such as skyscrapers creating shielded street canyons or walls blocking

the radio signal. To this end, a large number of research projects conceived novel positioning techniques (e.g. [22], [23], [18], [26], and [5]). However, all these systems either require specialized hardware or show poor positioning accuracy.

Many recent research activities focus on *IEEE* 802.11-based positioning because almost everywhere, especially in occupied areas of developed countries, 802.11 network infrastructure is available for data communication [2]. Universities, offices and many private homes utilize 802.11 networks to get rid of wires. As a reaction to the proliferation of 802.11 almost all modern mobile devices, ranging from smartphones to laptops, are shipped with build-in 802.11 network interfaces. 802.11 networks are not only used in indoor scenarios; even outdoors, many universities and coffee stop owners support nomadic users. Furthermore, 802.11 radio waves tend to travel outside the intended area covering adjacent regions as well. For instance, an access point deployed at a private home is often detectable while passing by [6]. Finally, 802.11-based positioning systems show sufficient positioning accuracy to be useful for a wide range of applications.

Another key argument for 802.11-based positioning systems is that 802.11 hardware can be used in dual mode: For data communication and for measuring the signal strength of neighboring access points as a prerequisite for positioning systems. For instance, the *GUIDE* tourist guide [7] and *PlaceLab* [16] are two representatives for outdoor positioning systems that utilize 802.11 for data transmissions as well as position determination. Many indoor positioning systems such as *RADAR* [3], *HORUS* [28] and *COMPASS* [15] require a high rate of signal strength readings especially if they are running in tracking mode.

In 802.11 the typical way of measuring the signal strength of access points within communication range is to perform active or passive scans. So far, it was unknown what happens to the data communication capabilities of mobile devices if signal strength measurements in form of active or passive scans are preformed concurrently. This work fills the gap by investigating how throughput and round trip delay suffer from different scan techniques and intervals.

Based on these results we conclude that active and passive scanning are inappropriate for positioning systems because they require too much time to gather information about neighboring access points and hence produce large communication dropouts. To overcome this problem, we propose a novel scanning technique called Monitor Sniffing. Monitor Sniffing exploits the fact that 802.11 uses overlapping channels by overhearing the wireless interface. Overhearing allows the mobile device to listen to frames from adjacent channels while concurrently staying on the channel used for data communication with the access point it is associated with. This is usefull because from [19] and [6] we know that in many populated areas on average 2.4 access points are in communication range, and very often the access points are configured to channel 6 or 10 as predefined in the factory defaults. We have implemented Monitor Sniffing using commodity 802.11g hardware and show that this scan scheme does not disrupt concurrent data transmissions and produces faster scanning results. Finally, our approach is compliant with the existing 802.11 standard and requires only a software update on the client side to get ready for use.

The rest of the paper is organized as follows. Section 2 introduces active and passive scanning and discusses how these approaches affect the communication capabilities of mobile devices. In Section 3, we propose Monitor Sniffing, a scan scheme especially designed for 802.11-based positioning systems. Section 4 presents the relevant related work. Finally, we conclude the paper and give directions for future work in Section 5.

2 Active vs. Passive Scanning

In this section, we discuss two techniques to discover neighboring access points, namely *active scanning* and *passive scanning*, as described by the IEEE 802.11 standard [11]. After introducing these two techniques, we investigate the effects of concurrent scanning on common data transmissions.

2.1 Functional Principles

IEEE 802.11 subdivides the radio spectrum into a set of channels. The number of available channels depends on where 802.11 is used and which sub-specification of 802.11 physical layer is selected. For instance, in the United States, only 11 channels are allowed for 802.11b and 802.11g, whereas 13 channels can be used in Europe. In contrast, the commercially less successful 802.11a defines 12 channels, however, in some countries the radio spectrum of 802.11a is still assigned to other purposes today.

A wireless network interface usually listens to one channel at a given time. So, if a mobile station wants to know all the access points in communication range, it has to tune its wireless network interface to each channel, one after another, and perform a scan.

IEEE 802.11 defines two scanning techniques: Active scanning and passive scanning. The former approach requires a bi-directional communication initiated by the mobile station. For the latter approach, the mobile station passively listens for management frames send out by access points. The details of these two techniques are discussed in the following two sections.

For the remainder of this paper, we focus on the infrastructure mode of 802.11 because this is the typical scenario in the field of positioning systems. We mainly focus on 802.11g because this is the one most frequently used and it is the latest substandard of the 802.11 standard family. However, our results are also applicable to 802.11b. Furthermore, we assume a scenario where we are located in Europe and 802.11 operates on 13 channels.

2.1.1 Active Scanning

A mobile device follows the subsequent procedure for each channel to perform an active scan: It tunes the wireless interface to the particular channel. Depending on the network card used, switching the channel requires 5 to 19 milliseconds [24]. The mobile device waits for either incoming frames generates by other devices due to data transmission or for the so-called *ProbeDelay* timer to expire. The timer makes sure that the mobile device is only waiting a certain period of time for incoming frames. After that, it uses the 802.11 medium access procedure to gain access to the channel and sends a Probe Request frame. It waits for the *MinChannelTime* to elapse, and if no frame is received it proceeds to the next channel. If a frame is received, the mobile device processes any Probe Response frame until the *MaxChannelTime* elapses.

The IEEE 802.11 standard does not define default values for these timers, however, [1] and [25] empirically studied the values used by wireless network card manufactures. In total, the exact time required to perform an active scan can vary significantly based on the number of available access points and hardware capabilities. In our measurements, we found

that at most 20 milliseconds are required to scan one channel. In total, an active scan over all channels takes less than 260 milliseconds to complete.

Active scanning relies on the support of access points. If an access point runs in the so-called "cloaked mode" it replies only to Probe Request frames that are looking for access points of the network it belongs to. A mobile station that is unaware of this particular network and uses active scanning is not able to learn anything about the presence of this particular access point and network. This behavior might be preferable in some circumstances (e.g., in order not to attract attention to someone who does not know the name of a privately owned wireless network), but for positioning systems the knowledge of every access point in communication range is desirable. This problem can be solved by performing passive scans.

2.1.2 Passive Scanning

Passive scanning has been introduced to reduce the workload of mobile devices and hence save battery power. As indicated by the name, passive scanning does not require any active communication of the mobile device. While scanning passively, a device listens to each channel and waits for a given period of time. If an access point is assigned to a particular channel, the mobile station should receive a so-called *Beacon* frame. Every access point broadcasts Beacon frames on a regular basis to maintain the network. Beacons usually contain the same information as Probe Response frames, such as supported data rates, supported extended data rates, and the name of the network. By examining the received Beacon frames a mobile device is able to recognize neighboring access points and their capabilities.

Once in a while, an access point sends out a Beacon frame that carries additional management information: A *Delivery Traffic Information Map* (DTIM). The idea is that battery powered mobile devices may sleep in low power mode while the access point buffers frames for theses stations. The DTIM indicates which stations have buffered traffic waiting to be picked up and hence this map should be received by every mobile device associated to a given access point. This means, that a mobile device should never miss such a Beacon regardless if it is sleeping or scanning. In case a scan is performed while a DTIM is scheduled by the access point, the mobile device is associated with, it is forced to cancel the scan process and wait for the map. For this, Beacons contain the beacon interval as well as how many regular Beacons will be transmitted before the next DTIM. This information allows a mobile device to determine the point in time when a DTIM will be transmitted by the access point it is associated with. The rate of DTIMs is often configurable, and commonly every 10^{th} Beacon is used for this purpose.

Access points usually broadcast a Beacon packet every 100 milliseconds which means that a mobile station should stay on a particular channel at least for the same period to make sure not to loose a Beacon from an unknown access point. Once again, note that this configurable value is not defined in the IEEE 802.11 standard. In total, a passive scan requires at least 1.3 seconds to be completed. This is nearly five times the length required for an active scan.

Another interesting fact arises from a privacy point of view: The infrastructure is not able to recognize a (mute) mobile station even if it utilizes the infrastructure for positioning. This has many implications on privacy and has already been discussed in [16]. Furthermore, passive scanning can even be used if a mobile station communicates with the infrastructure but does not want to reveal that it is computing its position based on 802.11.

2.2 Effects on Communication Performance

Among other tracking systems, 802.11-based positioning systems rely on a steady stream of signal strength measurements to determine the position of the user. This means, that active or passive scans are executed at a high rate and hence the network card is quite busy with scanning. In this subsection, we investigate how active and passive scanning affect regular data transmission of a mobile device in terms of throughput and round trip time.

2.2.1 Experimental Environment

To achieve interpretable results we simplified our environment: Only one mobile device communicates with one access point. This simplification allows us to investigate how scanning affects concurrent data transmission and to draw general conclusions. In a more complex scenario with additional mobile devices, throughput and round trip time may even be worse and more volatile.

We used a *Fujitsu Siemens* Lifebook T4010 laptop as a mobile device running *Linux* kernel 2.6.16 and *Wireless Tools* 28pre13. We implemented passive scanning support into the ipw2200 1.1.3 network interface driver [13], so that we were able to use the build-in *Intel* PRO/Wireless 802.11b/g network card of the laptop.

A *Linksys / Cisco* WRT54GS access points assigned to channel 8 has been used to gain access to the local network of the *University of Mannheim* whilst two *enterasys* RBT-4102-EUR access points set to channel 1 and 6, respectively, are used as landmarks. The WRT54GS access point was running the *Alchemy* firmware version 1.0 and was configured for 802.11b/g mode with a beacon interval of 100 milliseconds and a Delivery Traffic Indication Map every 10^{th} Beacon. The distance between the laptop and the access point was approximately 3 meters, and during the measurements a 54 MBit/s link between the laptop and the access point was established.

We conducted data transmission measurements with *iperf* 2.0.2 [9] to gauge throughput for both *UDP* [20] and *TCP* [21] and measured the round trip time with a UDP application. For this, we used an iperf server within the local network and an iperf client running on the laptop. The iperf server was connected to the access point via a 100 MBit/s switched Ethernet, so that the wireless link was the only bottleneck. Iperf was configured to measure the throughput every 0.5 seconds and to transmit data for 60 seconds. Our simple UDP application running on the laptop transmitted an UDP packet every 0.5 seconds and measured the time until it was echoed back from the same machine running the iperf server. For all the graphs presented in this paper, we carried out the experiments at least three times and selected the result showing the highest throughput.

2.2.2 Experimental Results

In this section, we investigate how scanning affects concurrent data transmission. For this, we first focus on UDP traffic and then analyze TCP data transmissions. Throughput and round trip delay are the main objectives and are first measured without any scanning at all to get a reference. Based on this reference, throughput and round trip time are quantified for various scan intervals and different scan schemes. The relation between the maximum throughput and the throughput achievable for a particular scan interval gives a well-balanced estimate on how common data communication is affected

by scanning. Additionally, the round tip time measurements indicate how interactive data communication is strained by concurrent scans.

UDP First of all, we measure the maximum throughput that is achievable with UDP over an 802.11g link. For this, we stepwise increase the bandwidth acquired by iperf from 15 MBit/s to 30 MBit/s. Our results show that the throughput reaches the peak at a sending data rate of 27 MBit/s. Figure 1 shows the achieved throughput and round trip time. As we see from this figure, the throughput varies between 22.25 and 26.75 Mbit/s while the round trip time is between 3 and 323 milliseconds. On average, the throughput is 25.155 MBit/s and the round trip delay is 82.30 milliseconds. Variations in throughput and delay are due to retransmissions on the MAC layer and changing radio channel characteristics.

In our next experiment, we perform active scans at a high rate to meet the requirements of tracking systems. Our wireless network card requires at least 260 milliseconds to perform an active scan, so we set the scan interval to 0.3 seconds. This means that only 40 milliseconds can be used for data transmissions between two consecutive scans. As we see from Figure 2, the throughput does not exceed 7 MBit/s and is very often lower than 1 MBit/s, resulting in an average throughput of 797 KBit/s with a standard deviation of 1.571 MBit/s. In other words, only 3 percent of the throughput is available in comparison to performing no scanning at all. The throughput peaks around seconds 6, 13, 22, 45, and 49 are due to canceled scans. The reason for this is that a Delivery Traffic Indication Map is scheduled by the access point as mentioned in Section 2.1.2. Furthermore, if we look at the round trip time we see that it varies between 23.5 and 6656 milliseconds resulting in an average delay of 2400 milliseconds with a standard deviation of 1331 milliseconds. With such a high round trip time in conjunction with low throughput, meaningful communication is often not feasible, especially if we consider typical applications in the field of mobile devices such as video streaming or Voice over IP.

Our next experiment shows the trade-off between active scan intervals and throughput. For this, we carried out measurements with active scan intervals ranging from 0.3 seconds to 4 seconds. Figure 3 depicts the results. As we see from the figure, an active scan interval of 0.5 seconds still produces a round trip delay of nearly 500 milliseconds and limits the throughput to 3.5 MBit/s. However, an active scan interval equal to or greater than 2 seconds produces acceptable results. On average, a throughput of more than 20 MBit/s (this corresponds to nearly 80 percent of the throughput that is available if no scanning is performed) and a round trip delay of less than 150 milliseconds is achievable, rendering network conditions well enough to allow meaningful communication. Dependent on the positioning system used and network applications running, an active scan interval can be selected to fit the needs of both applications.

In addition, we also investigate how passive scanning affects concurrent data transmission. While an active scan can be completed in less than 300 milliseconds, a passive scan requires at least 1300 milliseconds to terminate. For instance, the wireless network card we use stays 120 milliseconds on each channel while performing a passive scan, resulting in a overall passive scan time of 1560 milliseconds. Being busy with scanning such a long time may dramatically limit the achievable throughput. To investigate this phenomenon, we carried out experiments with a passive scan interval between 2 and 9 seconds. As shown by Figure 4, with a passive scan interval of 2 seconds only an average throughput of slightly more than 5 MBit/s is achievable while the round trip delay is approximately 700 milliseconds. Such a network condition is far from being usable. Especially, the high round trip delay makes interactive data communication useless. It takes at least

a scan interval of 7 seconds to obtain an average throughput and round trip time that is sufficient to allow common data transmissions. Even with a scan interval of 7 seconds the standard deviation is so large that the network condition cannot be considered stable. Compared to the active scanning approach, we see that passive scanning disturbs throughput and round trip delay even more.

TCP In this paragraph, we analyze how scanning affects TCP data transmissions. We distinguish between UDP and TCP traffic because UDP does not have any flow control mechanisms and hence in our simple scenario it shows the maximum achievable throughput. On the other side, TCP is the most frequently used transport protocol, and its flow control algorithms might be confused about communication dropouts caused by frequent scanning operations.

To get a reference value of how much data can be transferred over a 802.11g link, we invoked iperf in TCP mode and sampled the throughput and round trip delay for 60 seconds. Our measurements show, on average, a throughput of 17.1 MBit/s and a round trip time of 39.3 milliseconds; the standard deviation of the measurements is 768 KBit/s for throughput and 11.82 milliseconds for delay. We see a large difference between TCP and UDP throughput; this difference has already been studied by George Xylomenos et al. in [27] and by other researchers.

To make the results of the TCP measurements comparable to the ones obtained for UDP, we used the same scan intervals. Figure 5 shows average throughput and average round trip time as well as standard deviations for both measures in the active scanning scenario. An active scan interval of 0.3 seconds results in 547 KBit/s and a round trip time of 1.7 seconds. 547 KBit/s correspond to nearly 3 percent which is comparable to the UDP results. However, with an active scan interval of 1 second and TCP as transport protocol 10.4 MBit/s are achievable. In other words, more than 60 percent of the throughput can be obtained if no scanning is performed. The round trip delay also shows relatively stable values around 90 milliseconds with a small standard deviation of nearly 60 milliseconds. These are interesting results, especially in comparison to the UDP results, where stable network conditions can be observed only with an active scan interval larger or equal to 2 seconds.

For the sake of completeness, we compare TCP and UDP data transmission while performing passive scans. With a passive scan interval of 2 seconds TCP nearly achieves 2.2 MBit/s; that is less than 35 percent of what is achievable with UDP (see Figure 6). If we further compare the throughput of TCP and UDP, we see that the ratio nearly converges against equality if the scan interval is increased. For instance, for a scan interval of 5 seconds the ratio is 1:2 and a scan interval of 9 seconds shows a ratio of 1:1.27. To sum up, we see that small passive scan intervals impose a larger burden on TCP than on UDP and that with larger scan intervals UDP and TCP produce nearly the same throughput and delay.

3 Overhearing the Wireless Interface

An important and desirable feature of a good scan scheme for positioning systems is that it should result in minimal client service disruption as well as deliver a high rate of signal strength measurements of access points within communication range. Note that service disruption incorporates communication dropouts, declined throughput and increased delays from an end-user point of view. Unfortunately, these two requirements are mutually exclusive. Consequently, any scanning approach needs to balance the trade-off between these two requirements.

In the previous section, we already presented how active and passive scanning balance these features. Both approaches mainly focus on a high rate of signal strength measurements and care less about client service disruptions. To address a scan scheme that handles the client service disruption with care, we present a novel scan approach called Monitor Sniffing. Monitor Sniffing exploits overlapping channels of 802.11 to cut down client service disruptions while at the same time delivering a high rate of signal strength measurements of at least a subset of neighboring access points.

3.1 Monitor Sniffing

Our Monitor Sniffing scan scheme works as follows:

- We configure the wireless network card to work in monitor mode while associate it at the same time with a given access point for data communications. Monitor mode means that the network card driver does not block management frames such as Beacons, Acknowledgments, Probe Request and Probe Response. Instead, the network card driver forwards management frames to the network interface socket, so that these frames can be captured by a user space program.

  In the past, a wireless network interface could not be used to monitor the wireless channel while at the same time transmit data. For instance, the *Lucent* Orinoco Silver network card works only in monitor mode or in data communication mode, but not in both modes at the same time. Since recently, wireless network cards are available that support both modes concurrently (e.g., cards based on the *Intel* Pro/Wireless 2200BG or *Atheros* AR5xxx chipsets).

- We wiretap the wireless network interface by switching it into promiscuous mode to receive all frames and not only the ones addressed to the network card's MAC address.

- We add a filter to the network interface socket, so that only Beacons get through.

- We examine any Beacon we receive and offer the signal strength as well as the MAC address of the access point to the application layer.

The following sections discuss the concept of overlapping channels used by 802.11 and present an experiment that investigates how this concept can be exploited by overhearing the wireless interface to meet our needs.

3.2 Overlapping Channels

The IEEE 802.11b/g standards utilize the frequency spectrum between 2.4 and 2.5 Ghz. This spectrum is subdivided into 13 overlapping channels whose center frequencies are 5 Mhz apart while each channel has a spread of 22 Mhz around the center frequency. For instance, channel 6 spreads from 2.426 to 2.448 Mhz while channel 4, 5, 7, and 8 also utilize parts of this spectrum by having their center at 2.427 Mhz, 2.432 Mhz, 2.442 Mhz, and 2.447 Mhz, respectively. As a result, a transmission on one channel becomes detectable on an adjacent one. While overlapping signals from neighboring channels are undesirable for undisturbed data transmission it is preferable for positioning systems if the signals can be decoded.

Broadcasts such as Beacons or Probe Requests are usually sent with a data rate of 2 MBit/s in 802.11b/g networks. For this data rate, a *Direct Sequence Spread Spectrum* (DSSS) encoding is used. In general, DSSS spreads a signal over a wider frequency band which means in our case that the signal is spread over a complete channel. Therefore, transmissions encoded with DSSS are well protected against noise. Noise tends to take the form of relatively narrow pulses and hence destroys only a part of the spread signal. Even if only parts of a DSSS encoded signal can be heard, it is often still feasible to decode the signal.

3.3 Experimentation in a Real Environment

We carried out an experiment to investigate how well frames from adjacent channels can be decoded. The experimental environment as well as the results are described in the subsequent sections.

3.3.1 Experimental Environment

In addition to the experimental environment already described in Section 2.2.1, we set up four *Netgear* WG102, three *D-Link* 700AP, and three *Cisco / Linksys* WRT54G access points to cover all 802.11 channels. The access points were placed on a table in our lab, and the distance between the laptop and the access points was approximately 5 meters. Furthermore, we configured the Intel PRO/Wireless 802.11b/g network card of the laptop to work in monitor mode.

We developed a user space program to wiretap the wireless network card and collect Beacon frames from the network card socket. Additionally, the application also associates the network interface with a given access point for a given amount of time and logs every Beacon (e.g., signal strength and MAC address of the originating access point).

3.3.2 Experimental Results

For each channel, we collected the Beacons that were received during 60 seconds. Given a beacon interval of 100 milliseconds, an access point is supposed to emit 600 Beacons during a measurement cycle. Table 1 shows how many Beacons were received from which access point while the laptop switched through the channels. Note that we name the access points *A, B, . . ., M* corresponding to the channels they are assigned to (e.g., access point *A* is assigned to channel 1, the access point with name *M* is assigned to channel 13).

From the table we see that only a few Beacons are lost from the access point the network interface is associated with. The number of received Beacons ranges from 394 (see channel 6) to 582 (see channel 11). The reasons for this are a high interference rate due to the large number of access points located in close proximity as well as a high number of Beacon collisions. On the MAC layer, Beacons are broadcasted and hence no acknowledgments are used causing a colliding Beacon not to be retransmitted, and therefore to be missed.

Furthermore, we see that at least the Beacons from two neighboring channels in ascending and descending order can be heard even if a larger number of Beacons get lost. For instance, if the wireless network interface is assigned to channel 8 it only receives about 35 percent of the Beacons the access point of channel 9 is broadcasting. In contrast, if the wireless interface is tuned to channel 9 it is able to decode more than 87 percent of the Beacons access point *H* is emitting.

The variations are even larger if we look at the channels two steps away from the channel the wireless network interface is assigned to. For example, if the network card listens on channel 5 it is only capable to decode nearly 7 percent of the Beacons access point *G* is broadcasting. This is in contrast to channel 11 where nearly 85 percent of all Beacons broadcasted on channel 13 can be decoded.

Sometimes it is still feasible to decode Beacons from channels three steps away. As listed in Table 1, we see that it works for ten channels and only in three cases it is not possible to decode frames from channels three steps away (see channel 6, channel 9, and channel 10). If we calculate the percentage of received Beacons we see that there is a large variation. For example, only 1 percent of all broadcasted Beacons from access point *L* can be decoded if the network interface is tuned to channel 9. In contrast, nearly 64 percent of all Beacons from channel 1 can be decoded if the interface listens to channel 4. The reason for this is that channels three steps away do not directly overlap, instead, only side lobes as radiation of the main signal lobe are detectable. However, it is pretty difficult for the wireless interface to detect these low-powered signals and correctly decode them.

Additionally, we measured the signal strength of Beacons to explore if the reception power of Beacons transmitted on adjacent channels can be properly measured (see Table 2). An accurate signal strength measurement is a crucial feature for positioning systems because most 802.11-based positioning systems rely on signal strength measurements to determine positions. Due to the arrangement of the access points and the nearly equal distances between the access points and the laptop, we expected a low variation of the signal strength between channels during a measurement cycle. This is exactly what we see if we look at the columns of Table 2. The average signal strength measurements differ between 2 and 0 dBm. Signal strength measurements with less than three readings are omitted because less than three samples cannot be considered as a stable representative of the signal strength.

3.4 Discussion

On the basis of the experience we gained from the data collections, we learned that even in a different environment or if different hardware is used, the number of Beacons receivable from adjacent channels may vary but the general tendency remains valid. So, the experiment presented here shows that it is feasible to collect Beacons from adjacent channels by overhearing the wireless network card without switching channels.

Our most striking departure from previous work is that we wiretap the wireless network interface running in monitor mode to receive Beacons from adjacent channels as well as the channel the network card is assigned to. We chose this course for several reasons: First, based on the analysis presented in Section 3.2 we see that it is possible to receive Beacons from more than half of all 802.11 channels while staying on the channel the network card uses for communication. Depending on which channel is used for communication, an additional number of 3 to 6 adjacent channels can be received. For instance, for the channels between 4 and 10, 6 additional channels are accessible, and for all other channels at least 3 channels are receivable. Secondly, the rate of signal strength measurements is increased as compared to active and passive scanning. With Monitor Sniffing, a signal strength measurement is available every 100 milliseconds. For active and passive scanning

nearly 260 and 1300 milliseconds are required. Eventually, Monitor Sniffing is 2.6 times faster than active scanning and 13 times faster than passive scanning, at the expense of scanning all channels completely.

To analyze the performance of Monitor Sniffing with data communication at full speed, we repeated the experiment described in the previous section in such a way that we applied the iperf setup discussed in Section 2.2.1 to create the data communication. The average TCP throughput dropped around 4 MBit/s as compared to the experiment in Section 2.2.2 to nearly 12.5 MBit/s. The reason for this are frame collisions due to regular traffic on adjacent channels. For the same reason, the average round trip time slightly increased to a still acceptable value of 58 milliseconds. If we compare the number of adjacent channels from which Beacons can still be received, we notice that the numbers are comparable to the values presented in the previous section. We see that Monitor Sniffing produces scan results at high rate even if data is transferred at full speed.

Nowadays, mobile users are often covered by many access points at the same time. In such a scenario, the performance of positioning systems is affected slightly if only a subset of access points is recognized. For instance, a study of 802.11-based positioning algorithms [6] states that if a mobile device is covered by more than three access points the positioning accuracy only increases marginally. In case only one or two additional access points can be recognized by our approach, the mobile device is still able to perform active or passive scans.

4 Related Work

Several previously published studies investigate throughput and delay on 802.11. Xylomenos and Polyzos [27] explore the throughput of UDP and TCP achievable with several early 802.11 hardware devices. Their research focuses on throughput limitations caused by software implementation issues. The researchers recommend changes in the implementations of network protocols as well as in drivers. Duchamp and Reynolds measure throughput while varying the distance between a mobile device and an access point [8]. In [4], Bing measures delay and throughput for two early 802.11 network interfaces in a lab environment. A performance degradation is observed by Heusse et al. [10] if some mobile devices use a lower bit rate than the other devices. The authors analyzed the problem theoretically as well as empirically and derived a simple expressions for the useful throughput. Compared to our work, all these approaches do not consider scanning at all and use rather out-dated 802.11 or 802.11b hardware.

Scanning has been investigated mainly in the field of handover and roaming optimizations. Ishwar Ramani et al. [24] proposed *SyncScan*, a technique that tries to reduce the time required to perform an active scan by synchronizing the time when access points transmit Beacons. In their solution, access points are synchronized and broadcast Beacons in a pre-decided order. A mobile device is able to predict the time when a neighboring access point will broadcast a Beacon by listing to a Beacon of the access point it is associated with.

The IEEE 802.11k task force specification proposes that access points should not only advertise their own presence but should also provide a neighboring access points report upon request. Such a report contains a list of all access points as well as their assigned channels that are available in the neighborhood of a given access point. By knowing on how many access points can be expected at each channel, the time required to perform an active scan can be highly reduced.

These two techniques require support from the infrastructure in some kind which raises adoption hindrances. In contrast to this, a client side solution is proposed in [17]. The authors state that in typical urban and enterprise environments the access point density is fairly high, so that a mobile device often faces the opportunity of handover within the currently used channel. This approach is comparable to our Monitor Sniffing technique, however, we additionally exploit the fact that 802.11 defines overlapping channels and hence a mobile device is able to receive frames from access points assigned to neighboring channels.

5 Conclusions and Future Work

The primary contributions of this paper are an empirical analysis of how active and passive scanning affect concurrent data transmission, and we propose a novel scanning technique for 802.11-based positioning systems called Monitor Sniffing. We found out that with an active scanning interval of 2 seconds the resulting network conditions can be considered stable enough and suitable for common data transmission. The same is true for a passive scanning if a scan interval of 7 seconds is applied. For smaller scan intervals, the resulting network conditions cannot be considered as stable.

Our novel scanning technique allows 802.11-based positioning systems to stay on a certain channel while concurrently receiving Beacons from access points assigned to adjacent channels. We achieved this by overhearing the wireless interface running in monitor mode. This allows undisturbed data transmissions and high rate signal strength measurements to support precise position estimates.

In our ongoing work, we are trying to conceive an algorithm that automatically switches between active and monitor sniffing scans to maintain undisturbed data transmission and to deliver a minimum number signal of strength measurements from different access points. In the future, we also plan to work on improved active and passive scanning approaches that utilize overhearing to receive additional information and hence provide a higher measurement rate for these scan schemes.

Acknowledgments

The authors acknowledge the financial support granted by the *Deutschen Forschungsgemeinschaft* (DFG).

Availability

The passive scan and Monitor Sniffing implementations are available for download at [13] and [14], respectively.

References

[1] W. Arbaugh, M. Shin, and A. Mishra. An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process. *ACM SIGCOMM Computer Communications Review*, 33(2):93–102, April 2003.

[2] arkasha and bobzilla. Wireless Geographic Logging Engine - Plotting WiFi on Maps. Website: http://www.wigle.net, 2001-2005.

[3] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings of the 19th International Conference on Computer Communications (InfoCom)*, volume 2, pages 775–784, Tel Aviv, Israel, March 2000. IEEE.

[4] B. Bing. Measured Performance of the IEEE 802.11 Wireless LAN. In *Proceedings of the 26th Conference on Local Computer Networks (LCN)*, pages 34–42, Lowell, MA, USA, October 1999. IEEE.

[5] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. EasyLiving: Technologies for Intelligent Environments. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing*, pages 12–27, Bristol, UK, September 2000. Springer.

[6] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 233–245, Seattle, WA, USA, June 2005. ACM Press.

[7] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of Developing and Deploying a Context Aware Tourist Guide: The GUIDE Project. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 20–31, Boston, MA, USA, August 2000. ACM Press.

[8] D. Duchamp and N. F. Reynolds. Measured Performance of a Wireless LAN. In *Proceedings of the 17th Conference on Local Computer Networks (LCN)*, pages 494–499, Minneapolis, MN, USA, September 1992. IEEE.

[9] M. Gates, A. Warshavsky, J. Pietsch, B. Cerveny, M. Lambert, D. Finkelson, M. Zekauskas, M. Zekauskas, and K. Oliver. Iperf. Website: http://dast.nlanr.net/Projects/Iperf, August 2006.

[10] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance Anomaly of 802.11b. In *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, volume 2, pages 836–843, San Francisco, CA, USA, March-April 2003. IEEE.

[11] Institute for Electrical and Electronics Engineers, Inc. ANSI/IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Website: http://standards.ieee.org/getieee802/, 1999.

[12] E. Kaplan and C. Hegarty, editors. *Understanding GPS: Principles and Applications*. Artech House Incorporated, second edition, December 2005.

[13] T. King. Passive Scanning not supported? ipw2100 developer mailinglist: http://sourceforge.net/mailarchive/forum.php?thread%5Fid=27023011\&forum%5Fid=38938, July 2006.

[14] T. King and S. Kopf. Loclib - A Location Library. Website: `http://www.informatik.uni-mannheim.de/pi4/lib/projects/loclib/`, November 2005.

[15] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses. In *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, Los Angeles, CA, USA, September 2006. ACM Press.

[16] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. In *Proceedings of the Third International Conference on Pervasive Computing (PerCom)*, pages 116–133, Munich, Germany, May 2005. IEEE.

[17] V. Mhatre and K. Papagiannaki. Using Smart Triggers for Improved User Performance in 802.11 Wireless Networks. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 246–259, Uppsala, Sweden, June 2006. ACM Press.

[18] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. LANDMARC: indoor location sensing using active RFID. *Wireless Networks*, 10:701–710, November 2004.

[19] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall. Improved access point selection. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 233–245, Uppsala, Sweden, 2006. ACM Press.

[20] J. Postel. RFC 768: User Datagram Protocol. Website: `http://www.ietf.org/rfc/rfc0768.txt`, August 1980.

[21] J. Postel. RFC 793: Transmission Control Protocol. Website: `http://www.ietf.org/rfc/rfc0793`, September 1981.

[22] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43, Boston, MA, USA, August 2000. ACM Press.

[23] N. B. Priyantha, A. Miu, H. Balakrishnan, and S. Teller. The Cricket Compass for Context-Aware Mobile Applications. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–14, Rome, Italy, July 2001. ACM Press.

[24] I. Ramani and S. Savage. SyncScan: practical fast handoff for 802.11 infrastructure networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, volume 1, pages 675–684, Miami, FL, USA, March 2005. IEEE.

[25] H. Velayos and G. Karlsson. Techniques to Reduce the IEEE 802.11b Handoff Time. In *Proceedings of the IEEE International Conference on Communications (ICC)*, volume 7, pages 3844–3848, Paris, France, June 2004. IEEE.

[26] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[27] G. Xylomenos and G. C. Polyzos. TCP and UDP Performance over a Wireless LAN. In *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, volume 2, pages 439–446, New York, NY, USA, March 1999. IEEE.

[28] M. Youssef. *Horus: A WLAN-Based Indoor Location Determination System*. PhD thesis, University of Maryland at College Park, 2004.

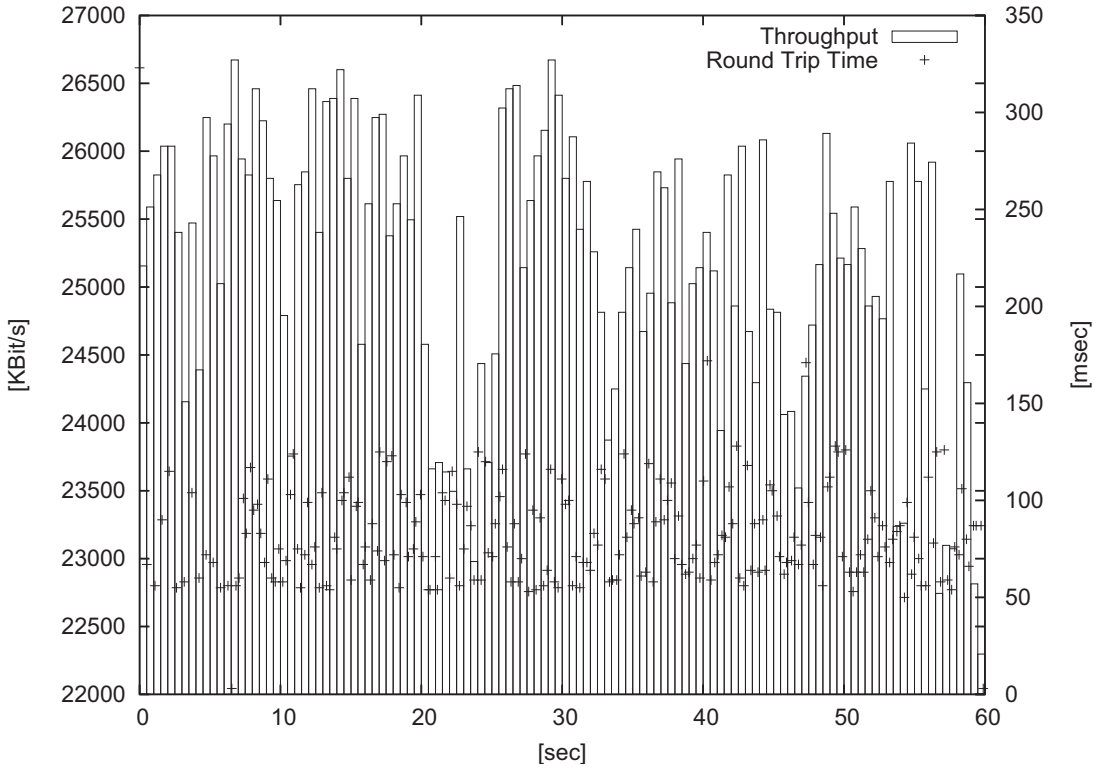

Figure 1: Throughput and round trip time of UDP data transmission over an 802.11g link.

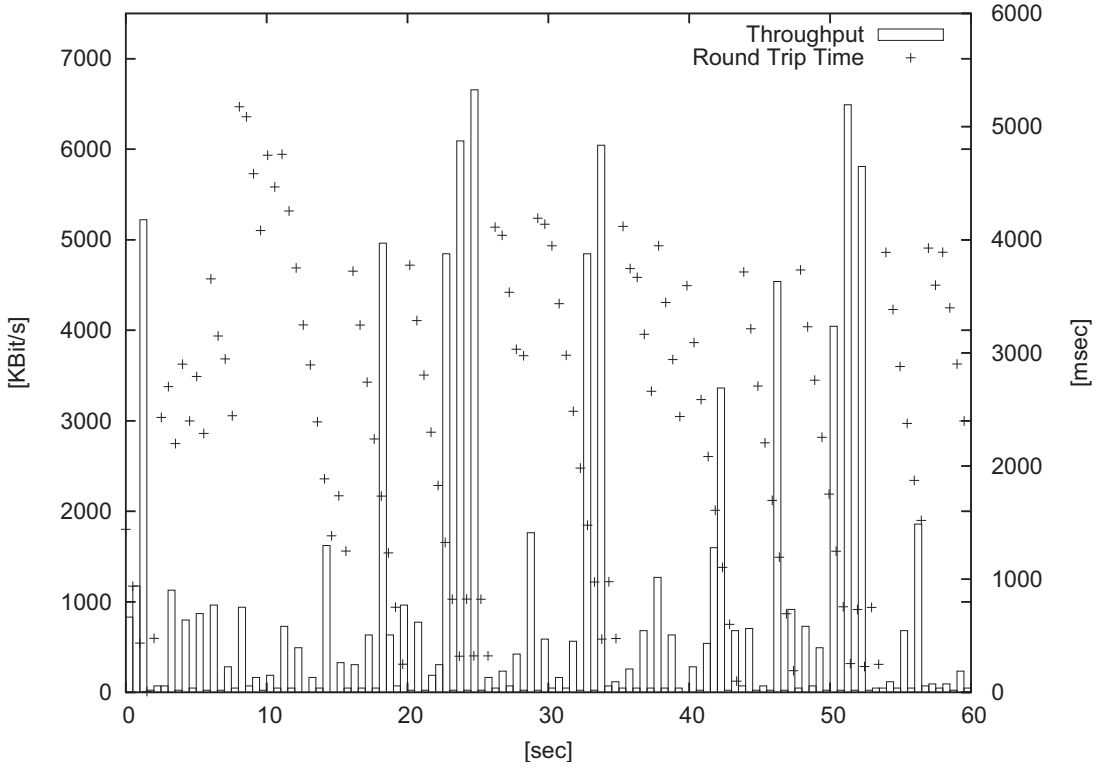

Figure 2: Throughput and round trip time of UDP traffic while concurrently performing active scans every 0.3 seconds.

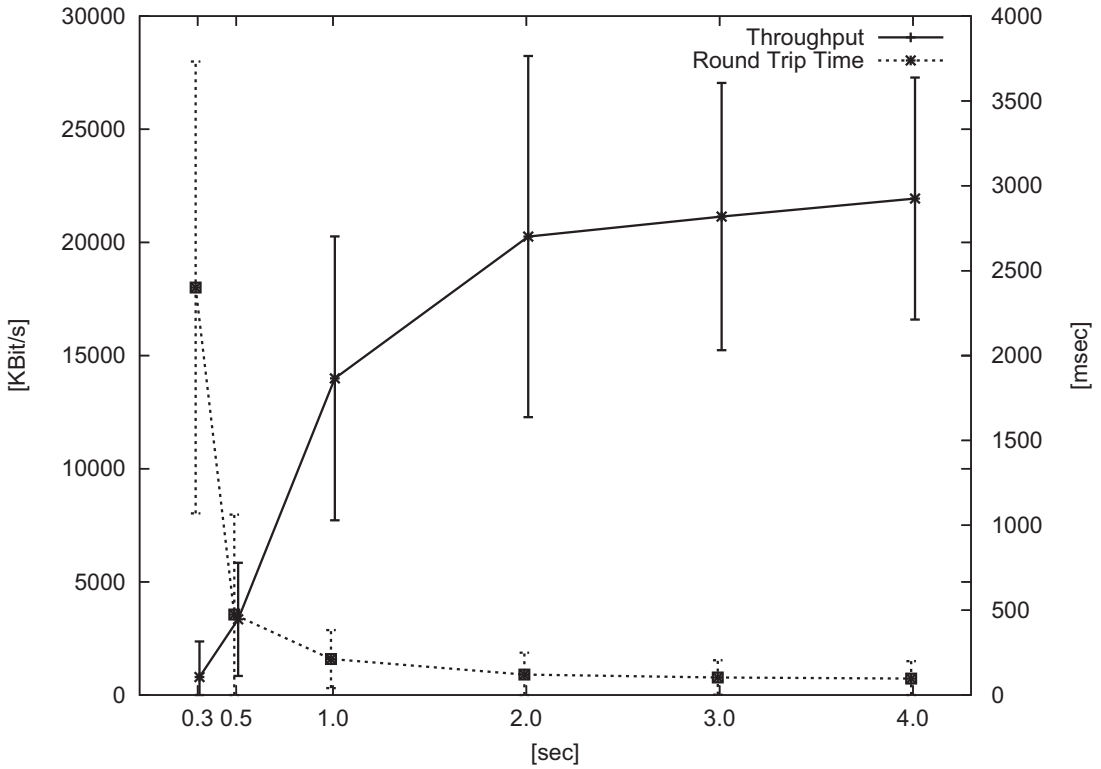

Figure 3: Average and standard deviation of UDP throughput and round trip time while concurrently performing active scans every 0.3, 0.5, 1.0, . . ., 4.0 seconds.

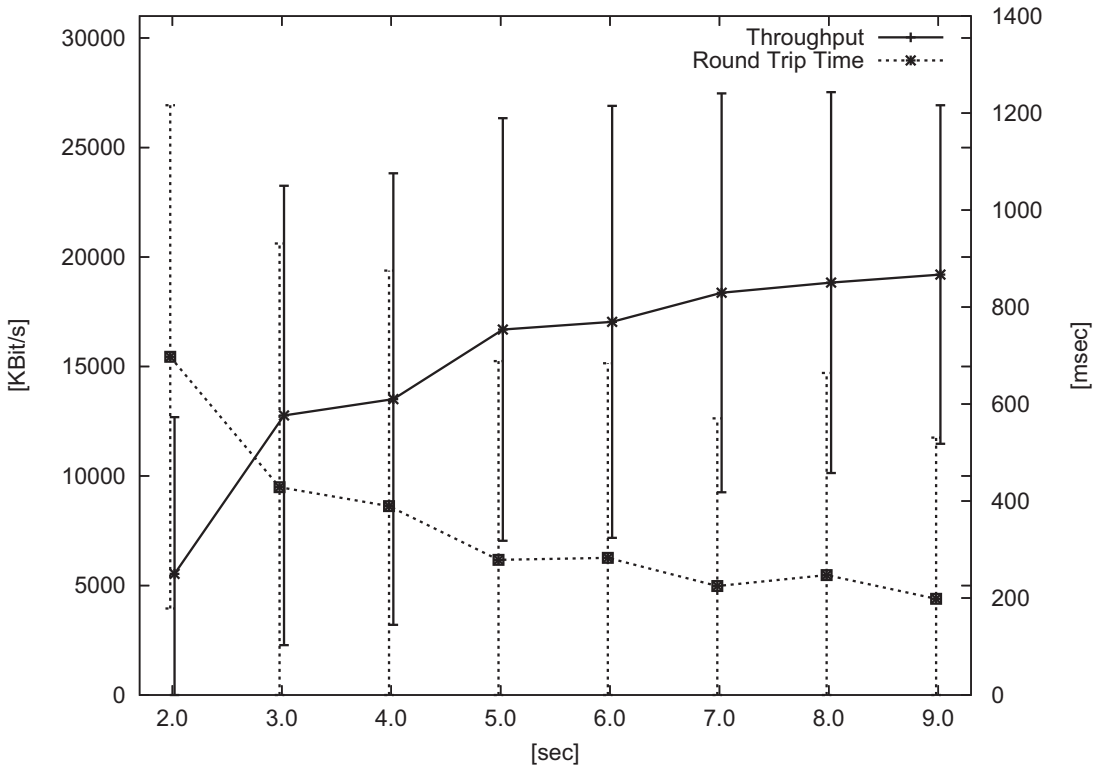

Figure 4: Average and standard deviation of UDP throughput and round trip time while concurrently performing passive scans every 2.0, . . ., 9.0 seconds.

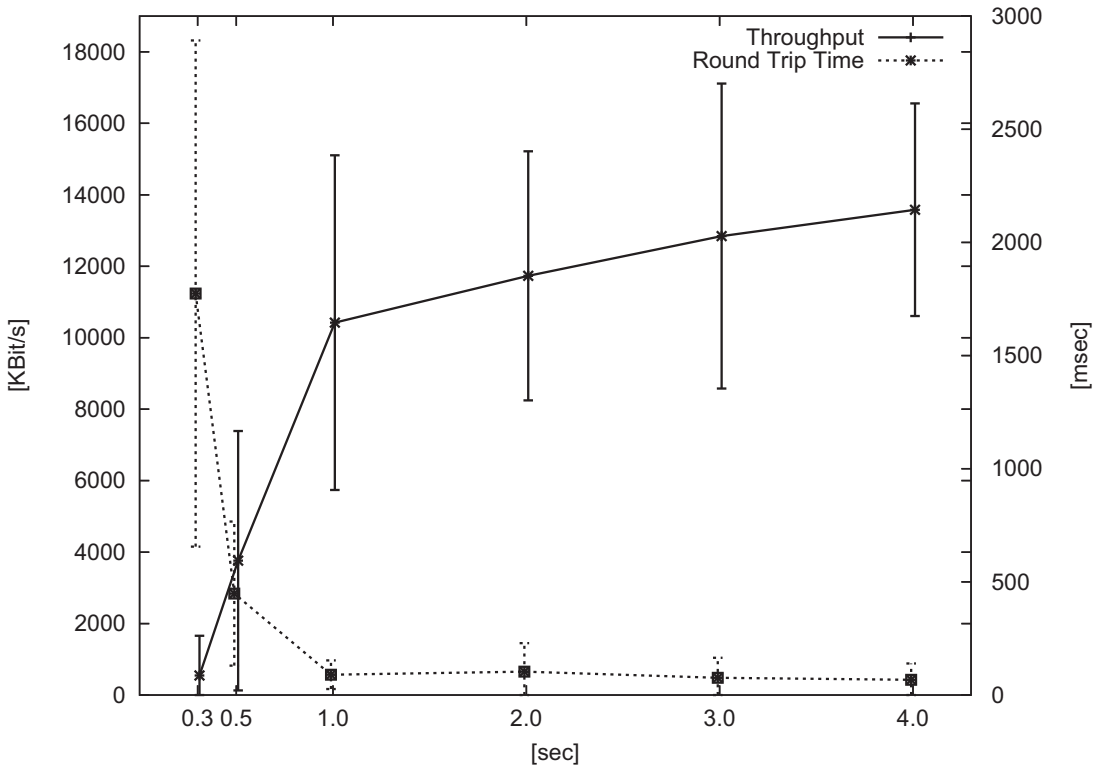

Figure 5: Average and standard deviation of TCP throughput and round trip time while concurrently performing active scans every 0.3, 0.5, 1.0, . . ., 4.0 seconds.

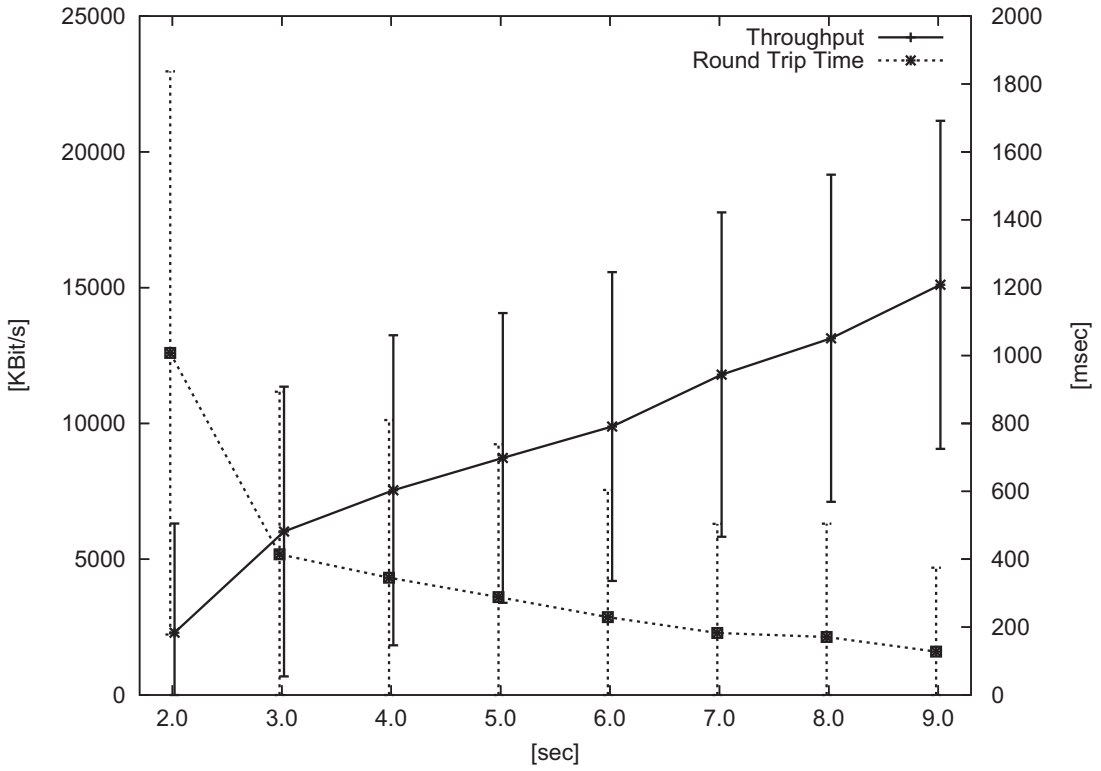

Figure 6: Average and standard deviation of TCP throughput and round trip time while concurrently performing passive scans every 2.0, . . ., 9.0 seconds.

Access	Channels												
Points	1	2	3	4	5	6	7	8	9	10	11	12	13
A	517	542	207	386	0	0	0	0	0	0	0	0	0
B	405	554	409	300	21	0	0	0	0	0	0	0	0
C	182	282	449	367	179	38	0	0	0	0	0	0	0
D	256	379	437	415	424	106	102	0	0	0	0	0	0
E	0	197	94	329	415	354	95	74	0	0	0	0	0
F	0	0	6	86	228	394	331	86	0	0	0	0	0
G	0	0	0	3	44	273	518	149	53	0	0	0	0
H	0	0	1	0	118	283	446	545	523	411	38	0	0
I	0	0	1	0	0	0	78	208	535	354	214	1	0
J	0	0	1	0	0	0	34	140	502	533	458	57	1
K	0	0	1	0	0	0	0	62	482	503	582	572	307
L	0	0	1	0	0	0	0	0	8	331	551	571	500
M	0	0	0	0	0	0	0	0	0	10	508	484	579

Table 1: Number of Beacons received from each channel while associated with an access point assigned to one of the 802.11 channels.

Access Points	Channels												
	1	2	3	4	5	6	7	8	9	10	11	12	13
A	-55.55	-57.14	-57.83	-58.01	0	0	0	0	0	0	0	0	0
B	-56.58	-58.03	-59.06	-58.61	-61.38	0	0	0	0	0	0	0	0
C	-56.23	-57.69	-58.63	-58.45	-61.20	-63.61	0	0	0	0	0	0	0
D	-56.36	-57.57	-58.49	-58.21	-61.09	-63.10	-64.31	0	0	0	0	0	0
E	0	-58.26	-59.29	-58.53	-61.47	-63.30	-64.16	-63.30	0	0	0	0	0
F	0	0	-61.00	-59.37	-61.83	-63.51	-64.38	-62.76	0	0	0	0	0
G	0	0	0	0	-62.00	-63.54	-64.45	-62.91	-63.49	0	0	0	0
H	0	0	0	0	-61.25	-63.10	-64.02	-62.43	-62.44	-62.69	-63.63	0	0
I	0	0	0	0	0	0	-64.88	-63.20	-63.03	-63.22	-63.78	0	0
J	0	0	0	0	0	0	-64.53	-62.54	-62.97	-63.03	-63.22	-60.86	0
K	0	0	0	0	0	0	0	-63.39	-62.86	-62.82	-63.20	-60.43	-60.01
L	0	0	0	0	0	0	0	0	-62.50	-62.76	-63.00	-60.45	-59.82
M	0	0	0	0	0	0	0	0	0	-62.60	-62.48	-60.24	-59.53

Table 2: Average signal strength of Beacons received from each channel while associated with an access point assigned to one of the 802.11 channels.