MobiSteer: Using Steerable Beam Directional Antenna for Vehicular Network Access

Vishnu Navda[1], Anand Prabhu Subramanian[1], Kannan Dhanasekaran[1],
Andreas Timm-Giel[2], Samir R. Das[1]
[1] Computer Science Department, Stony Brook University, Stony Brook, NY 11794, U.S.A.
[2] University of Bremen, TZI, Communication Networks, 28334 Bremen, Germany

ABSTRACT

In this work, we investigate the use of directional antennas and beam steering techniques to improve performance of 802.11 links in the context of communication between a moving vehicle and roadside APs. To this end, we develop a framework called *MobiSteer* that provides practical approaches to perform beam steering. *MobiSteer* can operate in two modes – cached mode – where it uses prior radio survey data collected during "idle" drives, and online mode, where it uses probing. The goal is to select the best AP and beam combination at each point along the drive given the available information, so that the throughput can be maximized. For the cached mode, an optimal algorithm for AP and beam selection is developed that factors in all overheads.

We provide extensive experimental results using a commercially available eight element phased-array antenna. In the experiments, we use controlled scenarios with our own APs, in two different multipath environments, as well as *in situ* scenarios, where we use APs already deployed in an urban region – to demonstrate the performance advantage of using *MobiSteer* over using an equivalent omni-directional antenna. We show that *MobiSteer* improves the connectivity duration as well as PHY-layer data rate due to better SNR provisioning. In particular, *MobiSteer* improves the throughput in the controlled experiments by a factor of 2 – 4. In *in situ* experiments, it improves the connectivity duration by more than a factor of 2 and average SNR by about 15 dB.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques; C.2.1 [**Network Architecture and Design**]: Wireless Communications—*Vehicular Communications*

General Terms

Performance, Design, Measurement, Experimentation

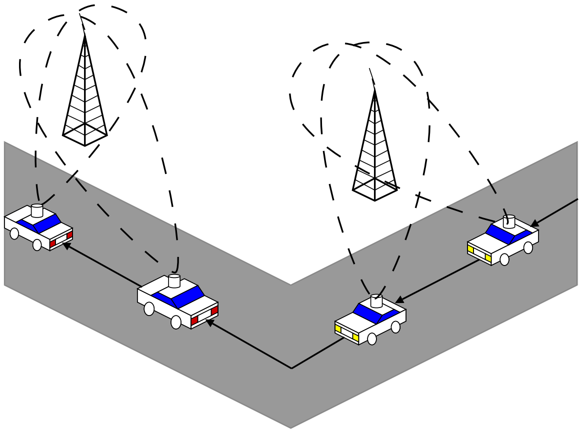


Figure 1: Beam steering and AP selection to improve connectivity.

Keywords

Steerable Beam, Phased-array Antenna, Vehicular Internet Access, Fast Handoff

1. INTRODUCTION

Recently there is a strong interest in developing networking techniques for moving vehicles, to enable wireless communication between vehicles, or between vehicles and fixed infrastructures near the road. Generally, three types of applications are emerging. First, communications between vehicles can enable various traffic safety and traffic information applications, developing the need for ad hoc communication between vehicles [26, 30, 23, 43]. Second, cars can serve as mobile sensors, providing a range of sensed information involving information related to the car, driving condition, road condition, traffic and environment – that can be uploaded into some form of "infrastructured" database to be queried by other cars or by some monitoring application [28, 25]. Third, the ability to do a general-purpose Internet access from cars can keep the car occupants entertained and informed, and can potentially generate services and applications specifically for in-car use [16, 31, 39].

To succeed, these applications need good connectivity to other vehicles or fixed access points (AP). While delay-tolerant techniques [22, 36] can be designed to smooth out periods of disconnections, end-to-end delay and possibility of lost messages do increase with long periods of disconnections

MobiSys'07, June 11–14, 2007, San Juan, Puerto Rico, USA.

reducing the effectiveness of many applications. In experiments reported in [16] regarding vehicular access of existing WiFi networks in urban areas, the authors have observed that the median connectivity duration to an AP is about 13 seconds, but the average inter-arrival times for “associable” APs is about 75 seconds, indicating possible long periods of disconnections. In addition the link-layer delivery rate is about 80% even when connected, which would be considered quite poor for TCP.

In this work, we will study a physical layer enhancement – directional communication — to improve network connectivity in vehicular context. By focusing energy in one direction, a directional antenna can get a better transmit or receive gain for a targeted direction compared to an omni-directional antenna [42, 41, 37]. In addition, directional antenna has the potential to provide a better immunity from co-channel interference [42, 27, 37] and multipath fading [14]. However, having a directional antenna alone is not sufficient for a moving vehicle. The direction must be steered appropriately for the best link quality. The steering must be done on a continuous basis as the car moves so that good connectivity can be maintained to the appropriate network node. The beam steering should be done in such a way so as to increase the duration of connectivity and improve the link quality. See Figure 1.

1.1 MobiSteer Design Goals

Our goal in this work is to develop practical beam steering techniques so as to maximize the duration and quality of connectivity between the moving vehicle and fixed access points (AP). We address this goal by developing *MobiSteer*, a 802.11-based mobile network node that uses steerable-beam directional antenna to be specifically used in moving vehicles with the appropriate beam steering technology. The current design of *MobiSteer* addresses the scenarios driven by the second and third applications mentioned above, where the vehicles access a fixed network (Internet) using one-hop 802.11 links. We use 802.11b/g as the link layer because of its wide availability, though much of the techniques developed in this work are not link layer specific and would apply as well to 802.11p based DSRC [3, 10], for example.

In the model we consider, the roadside APs use regular (omni-directional) antennas and the vehicle uses a steerable-beam directional antenna. There are two reasons for choosing such a model. First, omni-directional antennas are appropriate on the APs, as they may be associated to multiple vehicular nodes in different directions; thus to use directional antennas, they have to perform certain coordinations with their clients so that the beam from the AP steer to the right client at the right time. This is hard to do in a random access environment. Second, use of this architecture opens up the possibility of using existing 802.11 networks [16] that are now deployed widely. This enables cars to directly use existing 802.11 public hotspots, mesh networks, or even home networks that now blanket many urban regions. Accessing home 802.11 networks from moving cars in the neighborhood streets is attractive as most such 802.11 APs carry a broadband backhaul that is often idle [16].[1]

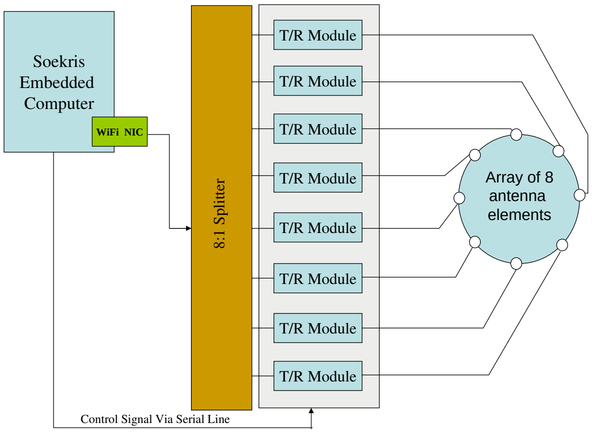


Figure 2: Block diagram of directional antenna components.

Given this model, we address the question of how the directional antenna beam steering should be done for maximizing duration and quality of connectivity to fixed APs. The quality of the link is determined by the PHY-layer data rate that is possible in the link between the vehicle and AP.

When the moving vehicle hears multiple access points, the decision of handoff arises. Essentially, the mobile node has to decide which access point to associate and which beam to use to communicate with the access point to get better connectivity. This boils down to the problem of designing beam steering techniques along with proper handoff decisions.

1.2 Contributions

The contributions in this paper are three fold.

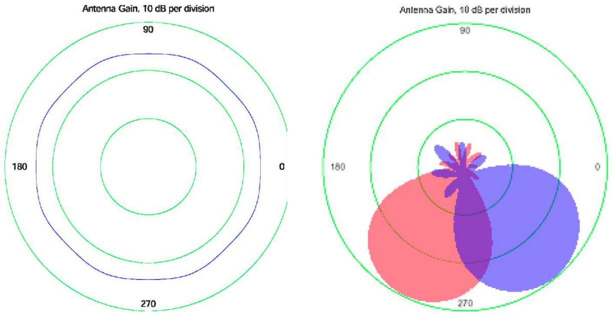
1. We show that use of steerable beam directional antenna with an appropriate beam steering technique can provide a significant performance advantage over using a fixed beam. An equivalent omni-directional beam pattern is used as a comparison point.

2. We develop practical beam steering approaches for two modes of operation – *cached* and *online*. In the cached mode, radio survey data is collected during the idle periods when the vehicle is not communicating with the fixed infrastructure and a geocoded RF signature database is created and maintained for frequently driven routes. This database is used to drive an algorithm that generates a trace of how beams should be steered and handoffs initiated as the car moves along the known route. In the online mode, on the other hand, no such database exists. We develop simple heuristics for beam steering and handoff based on lessons learnt from our experimental study.

3. We perform extensive measurements in a *controlled environment*, where we deploy our own road-side APs to aid performance data collection. We also provide

---

[1] The issue of security and open access are important in this context. However, they are a matter of developing appropriate application and protocol support, and is not our direct concern in this paper. Some of these discussions can be found in [16]. Community building efforts such as FON [4] are also relevant in this context.



(a) (b)

Figure 3: The beam patterns for Phocus Array antenna: (a) omni-directional; (b) two directional beams. The concentric circles are 10db per division. Taken from [1].

measurement data for accessing *in situ* networks [16], where we drive around in urban areas with reasonably dense population of 802.11 networks. The measurements show the power of our designed beam steering approaches.

The rest of the paper is organized as follows. In Section 2, we provide a detailed description of the architecture of *MobiSteer*. Section 3 describes the beam steering algorithm for the cached mode of operation. Section 4 presents detailed performance results for this mode. In Section 5 we discuss the online beam steering algorithm and also present the performance results. Section 6 discusses the related work. We conclude our paper and present the future work in Section 7.

2. MOBISTEER ARCHITECTURE

In this section, we describe the architecture of *MobiSteer*. First, we describe the hardware and software setup used in our experiments. Next, we describe the two different modes of operation, and then discuss methods of collecting data that drive the techniques.

2.1 Hardware Setup

Our directional antenna set up uses electronically steerable Phocus Array antennas from Fidelity Comtech [1] for the 2.4 GHz band used in IEEE 802.11b/g. The Phocus Array antenna system consists of eight element *phased arrays* driven by eight individual T/R (transmit-receive) boards that receive radio signals from the wireless card via an eight way RF splitter (See Figure 2). The phased arrays combine radio waves by introducing different phase differences and gains in the eight arrays [42, 1]. A T/R board is essentially a vector modulator with bi-directional amplifier controlled by software. Various beam patterns are possible by setting the phases and gains in different boards. The antenna is set to behave identically for transmit and receive, i.e., *the antenna gains for transmit and receive are the same.*

The software control on the antenna to produce different beam patterns is achieved via serial-line commands from an embedded computer (a Soekris net4511 board [9]). The Soekris net4511 embedded computer has a 100/133 Mhz AMD processor, 64MB SDRAM and a compact flash card interface used for storage. It also has a miniPCI and a PCMCIA interface. We use a 802.11 a/b/g miniPCI card based on Atheros [2] chipset with an external antenna interface. The embedded computer runs pebble Linux [8] with the Linux 2.4.26 kernel and the widely used `madwifi` [7] device driver for the 802.11 interface.

While many beam patterns are possible using the phased array, the manufacturer ships the antenna with 17 precomputed patterns – one omnidirectional beam and 16 directional beams, each with an approximately 45° half-power beam-width and low sidelobes. Each directional beam is overlapping with the next beam and is rotated by 22.5° with respect to the next, thus covering the 360° circle with 16 beam patterns. The directional gain is about 15dBi. Figure 3 shows the manufacturer provided beam patterns. In the experiments reported in this paper, we have used only the 8 non-overlapping beams out of the 16 beams to limit the number of experiments. Because of the overlapped nature of the beams, we found in our early experiments that the advantage gained from use of all 16 beams is marginal. We refer to the omni-directional beam with beam index 0 and the 8 directional beams we use with beam indices 1 to 8. Adjacent beams are numbered successively.

In order to get the location of the vehicle along the route it travels, we use a USB-based Garmin [5] GPS receiver inside the car that is connected to the embedded computer via a PCMCIA-to-USB converter. This GPS receiver provides a position accuracy of less than 3 meters 95% of the time. The embedded computer is powered via a PoE interface. We used the car battery and a PoE injector for power. For convenience, we will refer to the entire vehicular setup, including the embedded computer with 802.11 and GPS interfaces and the directional antenna as the *MobiSteer node*, in the rest of this paper. The setup is shown in Figure 4(a) and (b).

In Figure 4(c), we show the embedded router platform for the APs we use for our controlled experiments described in Section 4. The APs are Soekris [9] net4826 router boards with similar Atheros based 802.11 a/b/g miniPCI cards connected to regular rubber duck omnidirectional antennas. The APs also run the same base software (pebble Linux and `madwifi` driver) as the *MobiSteer* node.

2.2 Software Setup

The directional antenna beam pattern is changed via sending a serial line command from the embedded computer. The antenna vendor, Fidelity Comtech, supplied us with a patched `madwifi` driver, which implements an interface for user level programs to control the antenna beam patterns through the `/proc` virtual file system in Linux. On receiving a command via the `/proc` interface, the madwifi driver sends the command to the antenna controller over the serial interface and initiates a busy loop in the driver for 150 μs in order for the beams to stabilize. This is the nominal beam switching latency. We found that the `/proc` interface method added a huge delay in user to kernel communication(75 ms). So, we implemented a Linux `ioctl()` interface for controlling the antenna that incurred only an additional delay of 100 μs. Including this overhead, the total beam switching latency is 250 μs.

The `madwifi` driver allows creation of additional raw *virtual* interface (`ath0raw`) for a physical wireless interface. The virtual interface allows reception of all 802.11 frames (control, management, data) as if in the monitor mode, while the main interface can still operate in the ad hoc or in-


(a) (b) (c)

Figure 4: Hardware setup: (a) *MobiSteer* node – open; (b) *MobiSteer* node – in cover and mounted on a car; (c) embedded router platform for APs in controlled experiments.

frastructure mode. We modified Kismet [6], a popular wireless packet sniffer software to *optionally* capture all packets received on the raw virtual interface. Kismet communicates with the GPS server, running as a daemon (`gpsd`), and stamps the current time and GPS coordinates with each received frame from any AP. Each received frame (from APs) is also annotated with an index for the current beam pattern on the antenna, the current channel, the PHY-layer data rate and the SNR (Signal to Noise Ratio). The SNR, data rate, channel information for each received frame is obtained from the radio-tap header appended by the `madwifi` driver for each received frame. The entire frame with this annotated information is logged during the drive. We refer to this logged data as the *RF signature database*. This is built *during the idle periods* of the *MobiSteer* node when it is not communicating. This is used for later analysis.

2.3 Operational Modes

MobiSteer operates in two modes – *cached* and *online*. The cached mode is applicable when the car is in a familiar driving route (e.g., home to work). As noted before, the RF signature database is populated and maintained whenever the *MobiSteer* node is not in communication. This database is used later to drive an optimal beam steering and AP selection algorithm discussed in Section 3, when the *MobiSteer* node wishes to communicate and is on a "familiar" route. Thus, any "idle" drive provides *MobiSteer* with "samples" for the multidimensional <*location, timestamp, AP, channel, datarate, beam, SNR*> dataset. More drives on the same route provides more samples and thus better statistical confidence. We use an averaging technique (described in Section 3.1) for use in our algorithm. We have ignored a few practicalities such as AP churn in our current work; however, this can be easily accommodated in our technique by providing more weight to recent samples, ignoring APs not heard recently, etc.

The applicability of the cached mode technique is not limited to the routes driven by the same user. It is possible to share RF signature databases by multiple *MobiSteer* nodes. For example, databases could be uploaded to a central server and all nodes can benefit from such shared database. Whenever a user wants to travel a particular route, she can download the RF signature database, run our algorithm on it to get an optimal beam steering and AP selection pattern for the route. The CarTel [25] architecture is a perfect framework of how this can be done. The issue of sharing, however, is orthogonal to our work here. For the cached mode of operation, *MobiSteer* must know the projected route and an estimated speed of the car along every point on the route. We assume that such information is available from the navigation system and/or prior driving history.

In the online mode of operation, no such database is available. This mode of operation is used when the user travels in a previously untraveled route and wants to communicate with APs in the route. Here, the *MobiSteer* node scans the environment in all the beams and channels using active probing (discussed in the next section) and chooses the best beam and AP combination depending on the SNR values of the probe response frames received. This mode of operation is discussed in detail in Section 5. Figure 5 presents an overview of the modes of operation of *MobiSteer*. Note the possibility of switching between the modes of operation during the drive. A hybrid mode of operation is also possible, though we do not explore this here. *MobiSteer* switches to data collection whenever it is idle.

2.4 Data Collection

Two methods of data collection are used to build the RF signature database – *passive scanning* and *active probing*. Active probing is also used for the online mode of operation. In both these methods, the frames received at the *MobiSteer* node are used to infer the quality of the link between the AP and the *MobiSteer* node in both uplink and downlink directions. It is acceptable to assume that uplink and downlink qualities are similar. This is because the radio propagation characteristics are symmetric and the antenna transmit and receive processing are identical by default.

In passive scanning, the *MobiSteer* node scans for *any frame from APs* using the monitor interface (`ath0raw` in our case) on all antenna beams staying on each beam for about 200ms. This is done for every channel in sequence. The interval 200ms is selected due to the fact most APs broadcast beacons at the default interval of 100ms. Thus, 200 ms provides enough opportunity for the car to receive at least one frame from any neighboring AP when the surrounding medium is idle. For a fast moving vehicle, however, 200ms

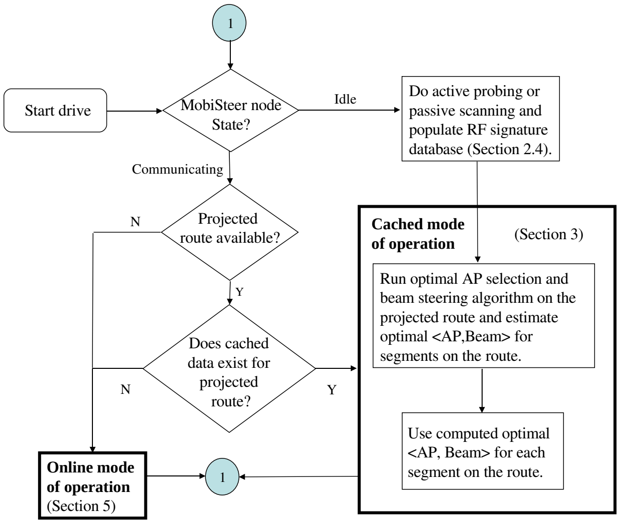


Figure 5: *MobiSteer* operation.

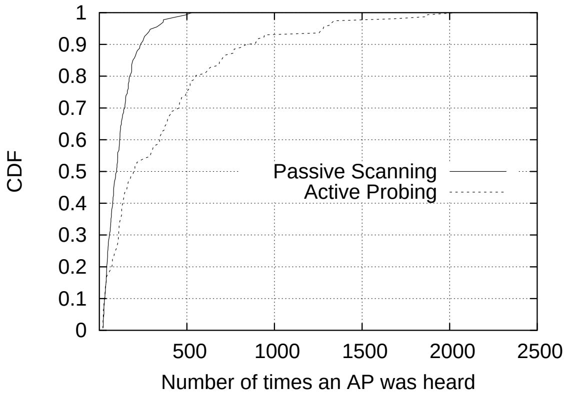


Figure 6: Performance of active probing vs. passive scanning in building RF signature database.

holding time for each beam can be too long. For example, with 9 beams (8 directional and 1 omni) the complete scan on 11 channels takes $9 \times 11 \times 200$ ms = 19.8 sec. At 100 km/hr the car can move 550m during this time. Compare this with the typical range of an AP, which is 150m according to [13]. This distance can be enough for the car to miss beacons from some APs. From our experience in data collection using passive scanning, we feel that this rarely would present a problem, as the data collection is likely to be repeated many times along the same route. Given the randomness in beacon generation and driving speeds, we expect that all APs would be heard over time on the beams/channels they are supposed to be heard. Also, several optimizations can be used. For example, from our own wardriving experience in connection with this work, we found over 90% of the APs are in channels 1,6, and 11. Here, the scanning time can be reduced significantly by scanning only in these three channels.

The second method of data collection is to use active probing to build the RF signature database faster. In active probing, we do not wait for beacons; instead the *MobiSteer* node sends out periodic *probe request* frames and record *probe responses* from the APs. Our software generates probe request frames that can be sent at customizable intervals. These probe request frames have the same format as those used in 802.11 probe request frames. Similar to passive scanning, in each channel, the *MobiSteer* node cycles through all 9 beams. Whenever a node sends a probe request frame, it takes about 30ms for the AP to send back the probe response frames. So with each beam it first sends out a probe request frame and wait for 30ms to gather probe responses, before switching over to the next beam.

Evidently, active probing allows quicker sampling, and thus gets more samples per drive. Figure 6 shows this. This data is based on 8 drives on the same 5 km route with active and passive probing near Stony Brook university campus with a fairly dense population of APs. It is thus possible that a stable RF signature database can be built with lesser number of drives on the same route when active probing is used. Either passive scanning or active probing can be used to support the cached mode of operation. However, active probing is exclusively used in the online mode to reduce the scan time. It is also possible to update the RF signature database when the *MobiSteer* node is actually communicating. Data packets then simply act as probes.

3. CACHED MODE OPERATION

The cached mode of operation rely on an existing RF signature database collected during “idle” drives. The idea here is to utilize this existing data to choose (i) APs along the route to connect to and (ii) appropriate directional beams to use. This is to be done at all points on the route. The goal is to maximize performance in terms of data transfer rate.

3.1 Optimal AP and Beam Selection

The data collection phase (either passive scanning or active probing) builds the RF signature database with tuples like <*location, timestamp, AP, channel, datarate, beam, SNR*> for a trajectory of interest, where *AP* denotes the BSSID and MAC address of the AP, *channel* is its channel, *beam* is an indicator for the beam used to receive this frame,[2] *SNR* is the Signal to Noise ratio value of the received frame, *location* is the GPS coordinates where the frame is received, *timestamp* is the time at which the frame was received and *datarate* is the PHY-layer data rate of the frame.The cached mode of operation uses an optimal AP and beam selection algorithm that computes the best AP to associate with at every point in the trajectory and the best beam to use to communicate with this AP.

To develop this algorithm, we first discretize the problem from practical consideration. We break the trajectory into segments of length Δ and assume that any AP or beam selection decision is taken only at the start of a segment. In each segment, we scan all beams on all channels to gather data samples. The value of Δ depends on the average speed of the car along the route and the time it takes to completely

---

[2] We assume that the antenna orientation with respect to the car is fixed in order to have a common representation for a *beam*.

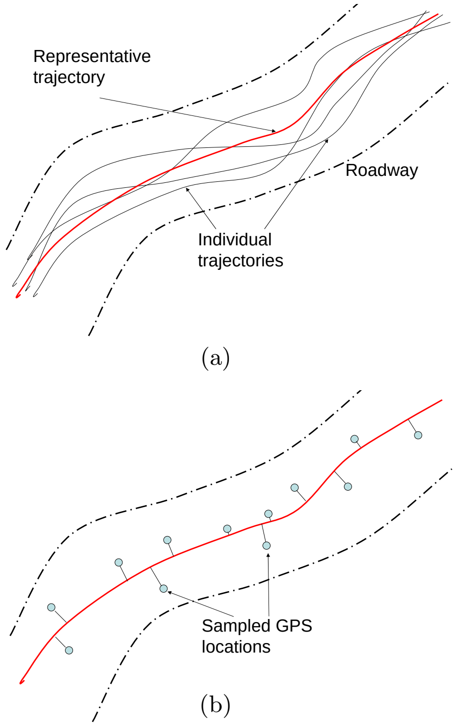


Figure 7: (a) Notion of representative trajectory. (b) Mapping of GPS samples.

scan all beams on all channels. Two values of Δ are used in the experiments we report, 5m and 40m, depending on the average speed of the car in the environments the studies were done.

In the data collection phase for the same route, the vehicle is likely to drive on slightly different trajectories on different runs. This is true even when the car is always driven on the same lane. Even if it is on the exact same trajectory, there is no guarantee that the locations provided by the GPS is exactly the same because of GPS accuracy issues. Thus, there is a need to compute some notion of a *representative trajectory* for a driving route. See Figure 7. There could be many ways to do this – for example, (i) by merging all location coordinates in the RF signature database for the same route and then interpolating a smooth path through them, or (ii) computing an average trajectory from the individual trajectories for each run, or (iii) by simply picking any one trajectory as the representative. We used the third method to save computational effort. The trajectories used in our study were quite close, and the actual method will have little influence on the results, if any.[3]

Every RF database tuple, collected during different runs of the same route, is mapped onto a point on the representative trajectory that is closest to the location of the tuple. See Figure 7. For every segment of length Δ on the representative trajectory, all such tuples mapped onto this segment are analyzed to compute the best AP and beam for this segment. This is done by computing the average SNR for each AP and beam combination and selecting the *best beam* for each AP for every segment. This will be used in the algorithm in Section 3.1.1.

Recall that the beam steering latency is about 250μs. This is negligible compared to the travel time in a segment. For example, for a segment of length (Δ) 5m, even at a very high speed of 120km/hour, the travel time is about 150ms. Even at this small segment size and high speed, the beam steering latency is around 0.1% of the travel time. We have used 5m as the lower bound of the segment length Δ. A smaller length is meaningless as it becomes comparable to GPS accuracy limits. Sometimes we have used higher lengths (upto 40m) because of some practical limitations we had. A small segment length implicitly requires a rich RF signature database with a dense set of samples. When driving speed is higher, samples collected become sparser – thus, for the same sample density, more runs are required.

While beam steering latency can be ignored, handoff latency may not be negligible. This may vary from a few ms to several 10s of ms, depending on whether channel is changed and/or authentication is used [32].[4] Since handoffs could be expensive at high speeds, the latency must be factored in the optimal AP and beam selection algorithm. This can be done in the following fashion.

3.1.1 *Optimal Handoff Algorithm*

Assume that the vehicle speed is known for each segment. This can be estimated from the RF signature database tuples at the time of computing the representative trajectory. At start of a segment i, we need to make a decision about the AP to use for this segment. Assume that the speed of the car is s_i at segment i and that the handoff latency when handing off from AP k to AP j is $h(k, j)$. Then, with this handoff latency, the useful time in segment i is given by

$$t(i, k, j) = \frac{\Delta}{s_i} - h(k, j).$$

Note that $h(k, k) = 0$. If the estimated average PHY-layer bit rate when associated with AP j in segment i *with the best beam* is $r(i, j)$, then the maximum number of bits that could be transferred in segment i is $r(i, j)t(i, k, j)$ with this handoff. Here the quantity $r(i, j)$ represents the link quality between AP j and the car within segment i.

Based on this, a simple dynamic programming algorithm can compute the best AP for each segment of the trajectory such that the aggregate number of bits transferred can be maximized. The algorithm is as follows:

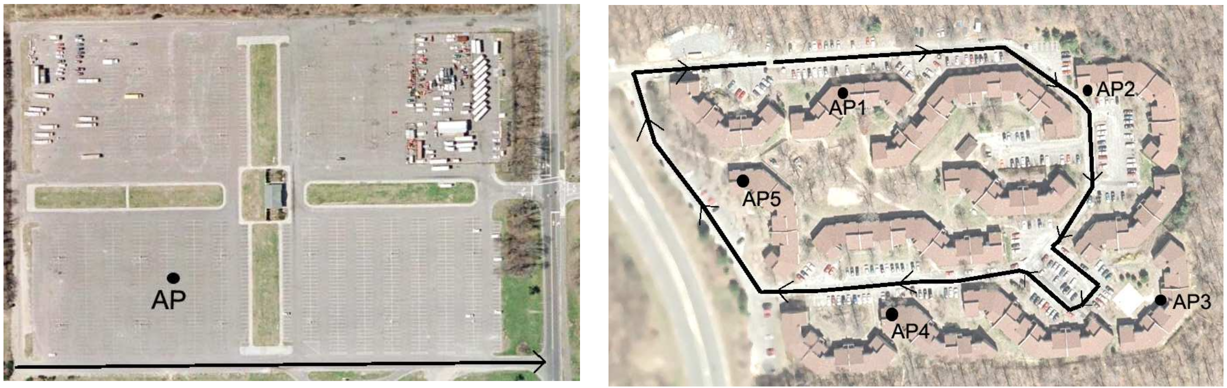
```
for $i = 2$ to Number of segments do
  for $j = 1$ to Number of APs do
    $best(i, j) = \max_{\forall k}(r(i-1, k)t(i-1, k, k) + r(i, j)t(i, k, j))$
    $prev(i, j) = k$ value for which the previous quantity
    is maximized.
  end for
end for
```

After this algorithm is run, the maximum $best(N, j)$ is picked for the last segment (N). Assume that this is $best(N, m)$. Then, $prev(i, j)$ is traced backwards as $prev(N, m)$, $prev(N-$

---

[3] In our experiments, we always stuck to the same lane. We have anecdotal evidence that two lane roadways can still be approximated quite well using a single trajectory. We do not have experience with wider roadways, where it may be possible that we need more than one trajectory to represent a driving route.

[4] Note that in WLAN deployments handoff latency is much higher because of probing. But probing is not needed here.



(a) (b)

Figure 8: Two environments for the controlled experiments: (a) a large empty parking lot in the Stony Brook university campus, (b) a student apartment complex in the same campus. The black dots show the AP locations and the black arrow shows the driving route.

$1, prev(N,m))$, etc., thus enumerating the best AP for each segment i. If no AP is visible for some segment i on the trajectory, a designated null AP is assumed (with bit rate $r(i, null) = 0$) so that the algorithm can run correctly.

This computed AP and beam combination for each segment is used to drive the beam steering and handoff of the *MobiSteer* node. Note that the algorithm is quite general. We have justifiably ignored beam steering latency and preselected the best beam for each AP in each segment. However, if the beam steering latency is high in a different hardware set up, it can be accounted for in a similar fashion as handoff latency.

In certain scenarios we can only measure the average SNR values within a segment and cannot estimate the average PHY-layer data rate. For these cases we can run the above optimal algorithm using average SNR values for $r(i, j)$. The algorithm determines the optimal AP and beam combination for each segment that maximizes the average SNR for the trajectory. Note also that the speed of the car is an input the algorithm and thus must be estimated. The algorithm must also know the route. We assume that this information is available from the navigation system or from prior driving history that can be cached.

4. CACHED MODE: EXPERIMENTAL RESULTS

In this section, we provide a detailed performance evaluation of the optimal AP and beam selection procedure described in the previous section.

4.1 Scenarios

We will use two scenarios for our experiments to evaluate our beam steering algorithm. The first is a "controlled scenario" where we deploy our own APs. See Figure 4(c) and associated description for the AP architecture. We use two specific controlled scenarios - (a) a large empty parking lot in Stony Brook University campus without any neighboring buildings and large trees — offering a virtually multipath-free environment with little, if any, external interference, (b) the graduate students' apartment complex in the same campus — offering diametrically opposite environment, rich in both multipath and external 802.11 traffic. See Figure 8 for the satellite image to get some understanding of the environments.

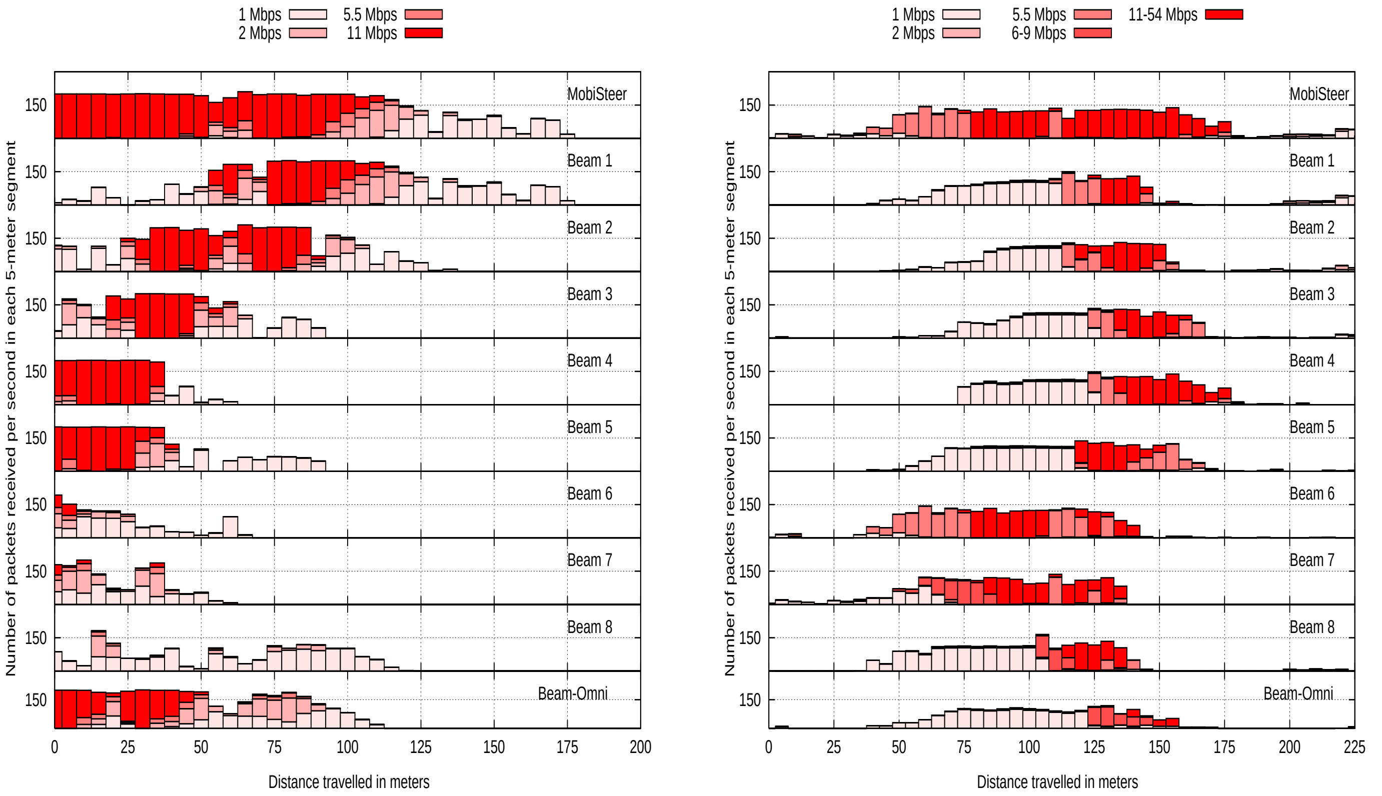
We use only one AP in the parking lot. It has been hard to use more than one AP gainfully in such a"clean" environment! However, we use five APs in the apartment complex. Here, the APs are carefully located so that at each point on our driving route, typically two APs are always heard and all points on the driving route are covered by at least one AP. This controlled set of experiments demonstrates the beam steering advantage by doing actual measurements of link-layer data transfer rate between the *MobiSteer* node and the APs. The APs are run on the same channel. Using just one channel in the experiments removes the channel variable from our experiments and lets us concentrate on only the beam steering aspect – the main focus of our work.

The second scenario uses *"In Situ"* wireless networks [16] in various urban roadways near Stony Brook University campus. This set of experiments demonstrate the beam steering advantage when using the APs that are deployed in an uncoordinated fashion in urban areas. Here also one single channel is used for the experiments. The most popular channel (channel 6, which is configured as the default channel in most commercial wireless routers) is used so that we can have most APs visible to the *MobiSteer* node.

In the *In Situ* experiments, we did not perform actual data transfers. There are a couple of reasons for this. First, this would restrict us using only open APs. We found that only 22% were open in our drive! Such a low fraction is atypical. We conjecture that in university and high-tech industrial areas (as ours) more wireless users know and care about security. Second, this would also require us to do statistically significant load tests that may interfere with the activities of the owners of the concerned networks. Thus, we have restricted load tests to only controlled scenarios. In *In Situ* scenarios SNR measurements coupled with coverage results are used to estimate performance advantages.

4.2 Collecting and Analyzing Data

In this section we first describe how we have collected data in the controlled scenario. Then we analyze the collected data for interesting properties.



(a) Parking lot (802.11b is used). (b) Apartment complex (802.11g is used; only one AP is shown).

Figure 9: Performance for individual beams for one specific AP along the driving route.

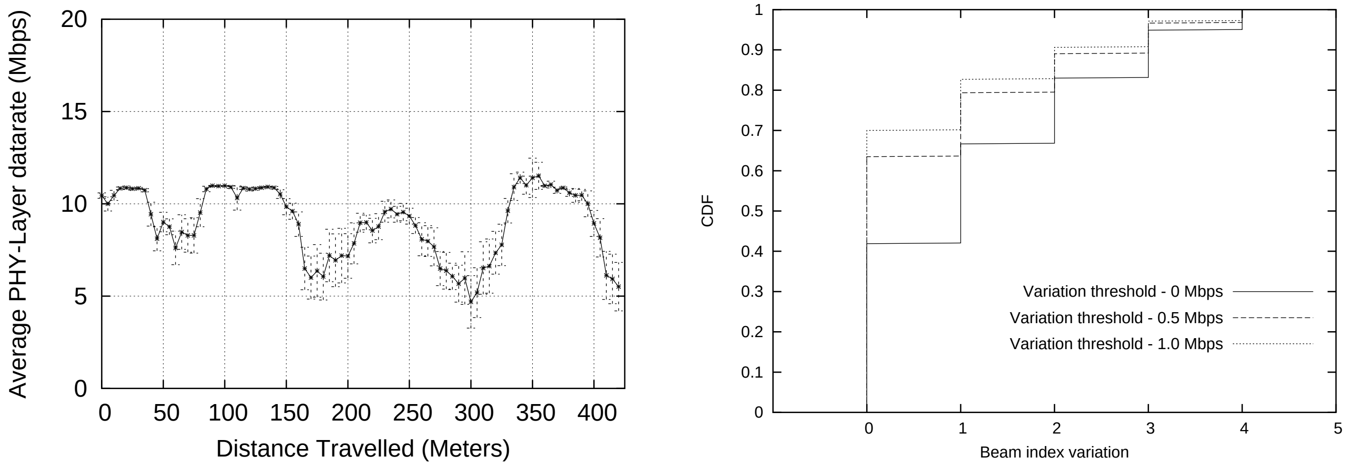


Figure 10: Variations in the collected data over multiple drives: (a) average PHY-layer data rate at every segment with 90% confidence interval; (b) variations in the best beam selection.

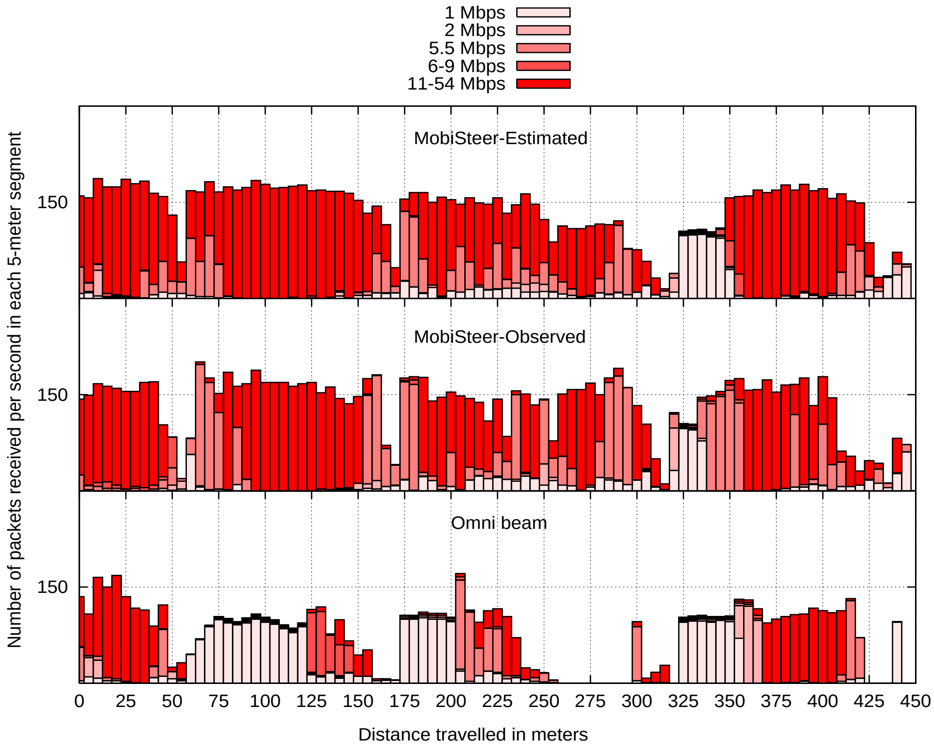


Figure 11: Estimated and observed performance on the best <*AP,beam*> along with the omni-directional beam performance for the entire path.

The APs operate in pseudo-ad hoc mode and continuously unicast 1000 byte UDP packets to the *MobiSteer* node at a constant rate of 300 packets/sec. The ad hoc mode is chosen instead of infrastructure mode so that the *MobiSteer* node can receive packets from any AP rather than only the specific AP it is associated to. This enables us to exclusively study the beam steering part of our algorithm. If and when the *MobiSteer* node receives any packet it records the tuple <*location,timestamp, AP, channel, datarate, beam, SNR*> as indicated before. The default auto-rate algorithm [15] in the card driver is used for rate adaptation.

The data collection in the controlled experiments is done fairly conservatively to eliminate any source of error. In order to eliminate any possibility of missing packets due to beam steering delays (which is already negligible), only fixed beams are used for each drive and beams are switched only between drives. So a set of 9 drives on the same path gives us data on each of the 8 directional beam plus the omni-directional beam. Each drive is done in a very slow speed (about 10 miles/hour). Segment length Δ is assumed to be 5m. Because of the slow speed, each segment receives enough packets to populate the RF signature database. We have done 8 such sets of drives on different days and times in order to analyze the variability of the data. Recall that we are using one channel as all our deployed APs are in the same channel.

Figure 9 shows the number of packets received per second on different beams in every segment of the drive based on the average statistics from the entire RF signature database collected over all the 8 drives. We also show the PHY-layer data rates with which these packets were received. When we did the parking lot experiments (Figure 9(a)) we used 802.11b. When we moved over to the apartment complex experiments (Figure 9(b)) we realized that we could take advantage of the higher data rate in 802.11g. So the plots have different data rate ranges. In any case, we will only qualitatively study them and not compare across them. In the apartment complex experiments (Figure 9(b)) we show the results for only one AP out of the five available for brevity. Plots for the other APs are qualitatively similar.

In the parking lot experiments (Figure 9(a)), we can clearly see how the performance of different beams vary as the *MobiSteer* node moves along the path. The performance improves as the car approaches the AP and then slowly fades as the car moves away from the AP. Also, the location of the peak performance shifts gradually as beams are changed from 1 to 2, 2 to 3 etc. Beams 6, 7 and 8 do not perform well as they are pointing to the opposite direction. The omni beam has only modest performance. The best beam (plot labeled *MobiSteer*) performance at every segment clearly outperforms any fixed beam and is significantly better than the omni beam. The aggregate throughput improvement over omni is more than twice.

In the apartment complex experiments (Figure 9(b)), the results are similar. However, given the rich multipath environment the shift in the peak performance point from beam to beam is not as clear any more. Also, no beam is clearly very poor. Unlike the parking lot, all beams achieve good data rate and offer similar connectivity durations. However, the best beam (labeled *MobiSteer*) clearly outperforms any individual beam and the omni beam. Here, the aggregate throughput improvement over omni is more than four times.

This set of plots brings out the potential of using beam steering as it exploits the beam diversity. There is also an improvement in the duration of connectivity by using beam steering compared to using omni-directional beam. This improvement is about 75% and 50% respectively in the two experiments. In addition, we notice that the unicast transmissions occur more often at higher PHY-layer data rates when using the best beam as compared to using the omni-directional beam. This is because the auto-rate algorithm switches to higher PHY-layer data rate when the number of packet retransmissions reduce. Since unicast data transmissions from an AP to the *MobiSteer* node includes synchronous ACK transmissions on the reverse direction, lower packet losses for unicast transmissions also imply that both uplink and downlink quality have improved, Going forward, for brevity we restrict our analysis to the apartment complex experiments only. This evidently offers a more challenging environment. Also, we have more APs in this environment to study AP selection.

Next, we analyze the variability of the collected data in the RF signature database over different drives in the same route. This is important, as more variability will require collection of a large number of samples and will make the entire process less reliable. Figure 10(a) shows the average PHY-layer bit rate for the best $<AP, beam>$ combination at each segment. The average is made over the 8 runs. The 90% confidence interval is also shown. While variations are indeed present, they are not significant. On careful analysis we also noted that the variations are a bit higher in the parts of the route which was surround by buildings *on all the sides*. We conjecture that this variation is due to the severity of multipath fading problem due to too many reflections.

Figure 10(b) shows the variability from another perspective. It shows how much the best $<AP, beam>$ selection would vary if the algorithm is run separately on the data set of each individual drive. It uses one of the drives as a reference and plots the difference in beam numbers for the best beam (recall that adjacent beams have successive beam numbers) over segments for each drive. The AP selections are not shown as the runs rarely differed on the selection of APs. The results show that both AP and beam selections are quite stable, and whenever the beam selection does vary, often an adjacent beam is selected. The plots use a notion of variation threshold where the beam selection process ignores variations in the bit rate values that are below certain threshold. For example, the plot shows that if a difference of 1 Mbps is acceptable, then 70% of times there is no difference in the best beam selections for different drives, and 83% of the times the beam selections remain within adjacent beams.

So far, we have only analyzed the collected data. To evaluate the performance of the optimal beam steering and handoff algorithm as presented in the previous section, we run the algorithm on the aggregate RF signature database. Since the PHY-layer data rate is available, the data rate – rather than SNR – is used for the best beam selection. The algorithm provides the best $<AP, beam>$ for each segment that would maximize the overall throughput.

4.3 Optimal Steering Results

We run the APs in pseudo-ad hoc mode as before with the same traffic. The *MobiSteer* node now steers the beam and selects APs following the algorithm output. No real handoff is done, as the *MobiSteer* node receives packets from all APs in the ad hoc mode. The *MobiSteer* node simply ignores packets from APs other than the one selected for the current segment. The beam steering is done as follows. The GPS receiver on the *MobiSteer* node continuously samples locations, maps the sampled location to the nearest point on the representative trajectory and determines the segment the vehicle is on. At the time of a segment change, the *MobiSteer* node steers the beam indicated by the best beam pattern from the algorithm.

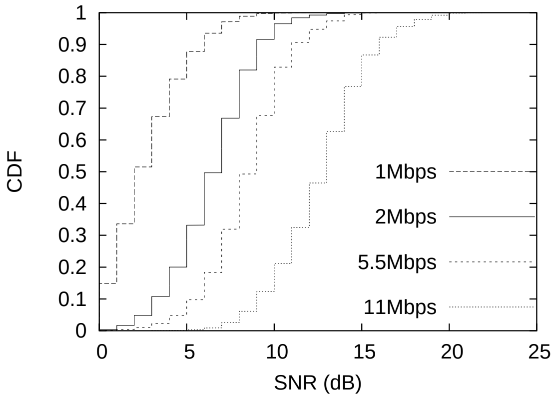


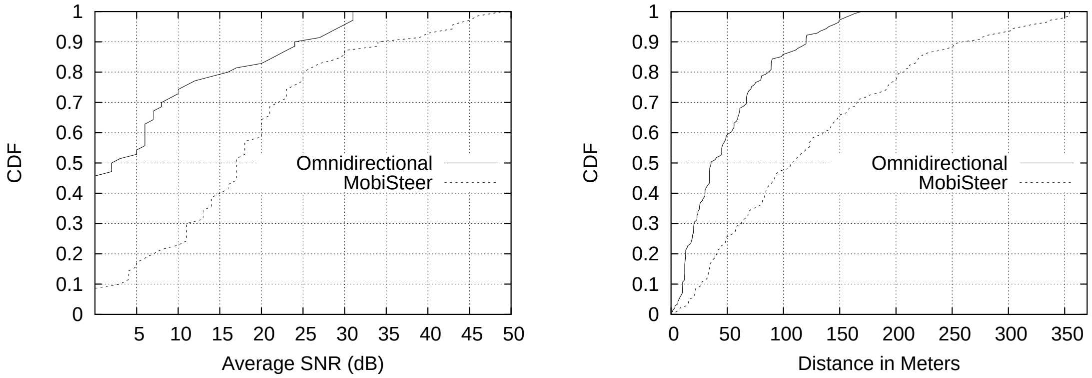
Figure 12: Measurement data demonstrating the relationship between SNR and PHY-layer datarate in our setup.

Figure 11 shows the observed and estimated performances on the best $<AP, beam>$ for every segment on the drive. The estimated performance is derived from the dataset that generated Figure 10(a). The observed performance is an average of 4 runs. The observed performance is close to the estimated performance with some variations due to the temporal changes in the propagation environment. A careful visual inspection also reveals that the difference is greater at the portions where large variability has been observed in Figure 10(a). The observed performance is significantly higher than the performance when using the omni-directional beam alone.

4.4 Experiments with *In Situ* Networks

The previous experiments in a controlled setting have established the power of directional beam steering and the viability of cached mode of data collection and computation of optimal steering and handoff. However, the experiments have been done in a controlled setting with carefully planned deployment of the 5 APs with good visibility everywhere on the driving route. We will now study the power of our cached mode technique using 802.11 APs normally present in urban environment.

For these experiments, we drove around the urban areas in the surroundings of Stony Brook university campus on different days and collected AP information using active probing. The route through which we drove (about 5 km) have both offices and homes on both sides of the road. The average speed along this route is around 30–40 miles/hour. Initial scouting runs through these routes revealed that a very high percentage of the APs were tuned to channel 6. So, for the experimental data collection, we used channel 6 alone. Our dataset consists of 307 unique



(a) (b)

Figure 13: (a) Average SNR from *In Situ* APs in each 40 meter segment along the route with beam steering and with omni-directional beam. (b) Distance along the route each AP was heard with beam steering and with omni-directional beam.

APs. Since the probe response frames are all sent using the same datarate (1Mbps in our case), we use the SNR of the received frames to determine the quality of the link. A set of independently performed micro-benchmarks demonstrates that SNR indeed impacts on PHY-layer data rates in our setup. See Figure 12. Clearly, with higher SNR the probability of the link using a higher data rate is higher.

We employ our algorithm on this data set to study the performance of *MobiSteer* and using omni-directional beam. A segment length (Δ) of 40m is used to factor in the higher driving speed. Figure 13(a) shows the CDF of the average SNR in each 40 meter segment along the route driven during the data collection. For each segment, we compute the average SNR for the frames received using omni-directional beam and the maximum of the average SNR for the frames received using each directional beam. This clearly shows the advantage of using directional beams as better SNR helps in achieving better PHY-layer data rate improving the quality of connection between the vehicle and the fixed access point. The median SNR on omni beam is around 2dB and using directional beams is around 17dB. Referring to Figure 12 this presents a significant improvement in data rate.

Figure 13(b) shows the distance for which each AP along the route is heard using omni-directional beams and beam steering. This plot ignores some outliers, where some APs are heard only for a very brief interval (1m or less). This plot shows the usefulness of *MobiSteer* to improve the connectivity duration. The median connectivity duration with an AP (note that this is different from AP range) improves from 40m with omni beam to more than 100m with *MobiSteer*.

5. ONLINE MODE

The cached mode of operation described in Section 3 is applicable only when the RF signature database for the route is available. When the *MobiSteer* node ventures into a completely new environment, AP selection and beam steering must be done in an online fashion. In this section we present a simple online heuristic to scan the environment on-the-fly and choose the best $<AP,beam>$ combination. We will study only the beam steering aspect as handoff and AP selection techniques are well investigated in literature [29, 34].

For online beam steering, active probing is performed over all beams and channels and the SNR values on all probe responses are recorded as described in Section 2.4. This is called a *probing* phase. After the probing phase, the $<AP,beam>$ combinations, where any probe response is received, are ranked in a table T according to the average SNR values of the probe responses. The *MobiSteer* node then associates with the AP AP_i, if $< AP_i, beam_j >$ has the highest SNR in table T. This $< AP_i, beam_j >$ combination is continued until d consecutive packet drops, when the next best beam (say, $beam_k$) for the same AP_i is selected, if $< AP_i, beam_k >$ exists in table T. This continues until no other beam for AP_i exists in T, or the last selected beam for AP_i failed to transmit a single packet successfully. At this time, the next best AP is selected from table T to hand off to and the corresponding beam is used for this AP. This strategy continues. If such an AP does not exist in T or if the handoff fails to associate, another probing phase is started to refresh table T. The size of the table T and the threshold of d drops that determines when to switch to the next $< AP, beam >$ combination are parameters for the heuristic.

The main penalty incurred when using the online mode compared to the cached mode of operation is the probing time. In order to probe in all the beams and channels, the probing time is around 3080 ms when using active probing. It takes 270ms to scan in all the 9 beams and there are 11 channels (assuming 802.11 b/g). The channel switching delay is around 10ms for our wireless cards. For the online mode of operation using only the omni-directional beam, the probing time is 440 ms as only channel scan and no beam scan is needed. Also, in the online mode, at every instant *MobiSteer* may not be able to use the optimal $< AP, beam >$ combination as these combinations are only discovered during probing and not during communication. However, online operation may have one advantage. Since the probing is done right before communication, it is relatively immune to temporal variabilities due to changes in vehicle locations, propagation environment, etc.

Figure 14 shows the median total number of bytes received by the *MobiSteer* node using various modes of operation in the controlled experiments in the apartment complex scenario from the 5 APs over 8 runs. It also shows the break-

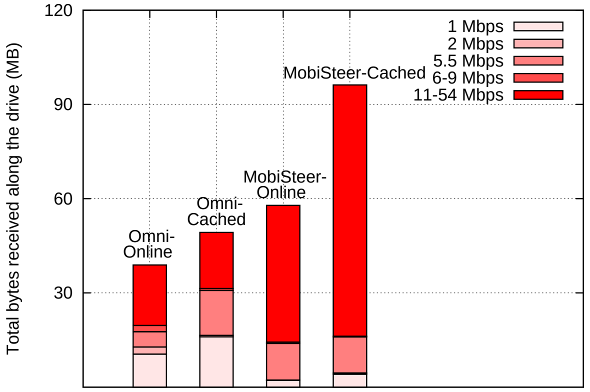


Figure 14: Total number of bytes received along the drive in "controlled" experiments (apartment complex scenario).

up of bytes received at different PHY-layer data rates. The figure clearly shows that beam steering outperforms omnidirectional communication significantly in both online and cached modes. It also shows that cached mode is superior to online. About 39% less packets are received when using online mode as opposed to cached mode. This is due to the time spent in the probing phase and possible use of suboptimal <*AP,beam*> combination. Also there is over 50% improvement in using *MobiSteer* in online mode compared to using only omni-directional beam in online mode.

In Figure 15 we show the average SNR in each 40 meter segment along the route for the in situ experiments. For the online modes, the zero SNR value indicates a probing phase or periods of no probe response along the route. This figure shows the SNR improvement in using beam-steering compared to omni-directional communication and also the performance benefits of using cached mode compared to online mode. In this set of experiments, the average SNR using *MobiSteer* in online mode is around 11 dB compared to 4 dB when using omnidirectional in online mode. The average SNR using *MobiSteer* in cached mode is 18 dB with a 7 dB gain over *MobiSteer* in online mode. The online mode sometimes performs better than cached mode due to availability of fresher channel estimates, but, in general, scanning all beams and channels in online mode incurs a large penalty.

The main insight from these experiment is that using *MobiSteer* cached mode of operation is far superior to online mode, and should be used whenever RF signature database is available. The *MobiSteer* online mode is still superior compared to using omni-directional antenna.

6. RELATED WORK

Several works have demonstrated the feasibility of IEEE 802.11 based communication from moving vehicles. In [35] the authors show how long a connection between a moving car and a roadside AP can be maintained while driving at different speeds between 80–180 km/hour. They show that approximately a third of the connections can be reasonably used and communication is possible for about 4–9 seconds in these speeds. Up to 9 MB of data could be transmitted at speeds around 80 km/hour. These experiments have been conducted in a well-planned environment, similar to our controlled experiments. More recently in [16], experiments are performed to connect to unprotected access points in metro areas in normal driving environments and use them for Internet connectivity. The experiments – performed over 290 drive hours – show that the average duration of connectivity to an AP is about 13 seconds at prevalent driving speeds, and several 100s of KBytes data on average can be transferred during this period using TCP to a server on the Internet. While this seems to be a good news, the data also show that that there could be significant periods of disconnections, necessitating disconnection tolerant protocols at upper layers. In particular, while the median connectivity duration is about 13 seconds, the average inter-arrival times for "associable" APs is about 75 seconds, indicating possible long periods of lack of connectivity. In addition, the link-layer delivery rate is about 80%, which would be considered quite poor for TCP. These observations originally provided motivation for our work.

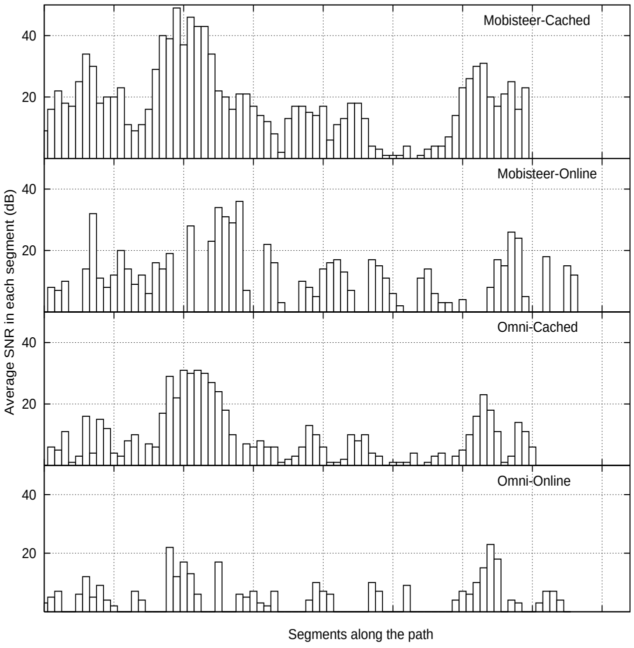


Figure 15: Average SNR in each segment along the drive in *In Situ* experiments.

Several studies investigate the characteristics of unplanned deployment of 802.11 APs in dense urban areas. Prominent of them are several "war driving" studies [11, 12] and extensive measurement studies in [13]. However, these studies do not directly consider feasibility of vehicular access. Other studies have considered vehicular access, but are interested in developing disconnection tolerant protocols [36]. Several architectural and application-layer works have also appeared in literature that considers many uses of urban vehicular grid, including vehicular Internet access [24].

None of the above works consider use of directional antenna to improve connectivity or any other performance measures. However, in other contexts – such as ad hoc or mesh networks – several papers have considered use of directional antenna to improve spatial reuse and immunity from interference. See, for example [27, 20, 37, 19]. These papers mainly considers modifications and extensions of 802.11 for use with directional antennas. They, however, do not directly consider mobility. All of these works are simulation studies on using directional communication.

Experimental studies combining 802.11 and directional antennas are quite limited. Some researchers appear to do experimental work [18, 38, 17, 33] using various steerable and switched beam antennas. However, actual experimental data has been reported only in [18]. In [17, 33], the authors use the same directional antenna we use in our work. In [17], they study how to enhance security using directional communication and in [33] present ways of exploiting directional communication for better spatial reuse. But no real experimental results are reported. Another paper [38] demonstrates significant throughput improvements in mobile environments using directional antennas with respect to omni-directional. A design is presented, but the results use only high-fidelity simulations.

In wireless communications, signal propagation and antenna literatures, the concept of adaptive beam steering from vehicles or from fixed base stations have indeed appeared [21, 40, 44]. However, experimental work in real environments has been limited. Also, this set of work is not related to either 802.11 or Internet access. To the best of our knowledge, ours is the first experimental study that considers steerable beam directional antennas for 802.11 networks for accessing roadside APs for the purpose of using the Internet.

7. CONCLUSIONS AND FUTURE WORK

In this work, we have investigated the use of directional beam steering to improve performance of 802.11 links in the context of communication between a moving vehicle and roadside APs. To do this, we have used a framework called *MobiSteer*. *MobiSteer* can operate in the cached mode – using prior radio survey collected during “idle” drives – or, it can operate in an online mode, using probing. The goal is to select the best AP and beam combination at each point along the drive given the information available so that throughput can be maximized. We have used extensive experiments – controlled scenarios with our own APs, in two different multipath environments, as well as *in situ* scenarios, where we use APs already deployed in an urban region – to demonstrate the performance advantage of using *MobiSteer* over using an equivalent omni-directional antenna. *MobiSteer* improves the connectivity duration as well as PHY-layer data rate due to better SNR provisioning. Summarizing the results, *MobiSteer* has improved the throughput in our controlled experiments by a factor of 2 – 4. In *in situ* experiments, it has improved the connectivity duration by more than a factor of 2 and average SNR by about 15 dB.

We have also demonstrated that cached mode of operation is superior to online mode giving more than 50% improvement in throughput. Thus, improved techniques to collect, maintain, organize and share radio survey data (RF signature database in our terminology) need to be developed. In our ongoing work, we are researching this aspect and collecting significant volumes of radio survey data in furtherance of our *in situ* performance results.

The concept of *MobiSteer* can be used in several related vehicular applications. For example, it can be used for ad hoc communications among vehicles. This, of course, adds to the complexity as both communicating nodes have to steer their beams, needing coordination techniques. We also plan to augment *MobiSteer* node with cellular modem service and further extend the cached mode operation to include cellular link quality information. This will enable us efficiently multiplex between the cellular network and the WiFi network based on availability and link quality of these networks at different locations. The other application of *MobiSteer* is localization of roadside APs. We believe that reasonably accurate localization is possible, as the moving vehicle is able to take many SNR samples at different directions and locations, thus providing diversity. Though multipath propagation can complicate the measurements, our initial work indicates that statistical estimation techniques can be used to improve accuracy in such cases. Accurate localization of roadside APs can provide useful datasets for the wireless networking research community to understand better the nature of chaotic WiFi network deployments in urban areas. This will also be useful to create realistic topologies for wireless mesh networking research.

8. ACKNOWLEDGMENT

We would like to thank Fidelity Comtech [1] for providing excellent technical support for the Phocus Array antenna devices and answering many questions. Special thanks to our shepherd, Lin Zhong and all the reviewers for their useful comments. This work is supported in part by the NSF grants CNS-0519734, OISE-0423460 and CNS-0308631 and by the German Research Foundation in the framework of the Collaborative Research Center 637: Autonomous Cooperating Logistic Processes: A Paradigm Shift and its Limitations.

9. REFERENCES

[1] 802.11 Phocus Array Antenna System by Fidelity Comtech. http://www.fidelity-comtech.com/.

[2] Atheros Communications. http://www.atheros.com.

[3] Dedicated Short Range Communications. http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[4] FON. http://www.fon.com/.

[5] Garmin GPS System. http://www.garmin.com/.

[6] Kismet Wireless. http://www.kismetwireless.net/.

[7] MADWIFI Project. http://sourceforge.net/projects/madwifi/.

[8] NYCwireless Pebble Linux. http://www.nycwireless.net/pebble.

[9] Soekris Engineering. http://www.soekris.com/.

[10] Telecommunications and Information Exchange between Roadside and Vehicle Systems – 5.9 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical layer (PHY) Specifications. ASTM International, www.astm.org.

[11] War Driving. http://www.wardriving.com/.

[12] Wireless Geographic Logging Engine. http://www.wigle.net/.

[13] A. Akella, G. Judd, P. Steenkiste, and S. Seshan. Self Management in Chaotic Wireless Deployments. In *11th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2005.

[14] T. Aubrey and P. White. A Comparison of Switched Pattern Diversity Antennas. In *IEEE Vehicular Technology Conference (VTC)*, 1993.

[15] J. Bicket. Bit-rate selection in wireless networks. *MIT Masters Thesis*, 2005.

[16] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In *12th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2006.

[17] J. M. Carey and D. Grunwald. Enhancing WLAN Security with Smart Antennas: A Physical Layer Response for Information Assurance. In *IEEE Vehicular Technology Conference (VTC)*, 2004.

[18] R. R. Choudhury, T. Ueda, J. Bordim, and N. Vaidya. BeamNet: Ad Hoc Networking Testbed Using Beamforming Antennas. In *IEEE Vehicular Technology Conference (VTC)*, 2005.

[19] R. R. Choudhury and N. Vaidya. Deafness: A MAC Problem in Ad Hoc Networks when using Directional Antennas. In *10th IEEE International Conference on Network Protocols (ICNP)*, 2004.

[20] R. R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya. Using Directional Antennas for Medium Access Control in Ad Hoc networks. In *8th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[21] F. Dobias and W. Grabow. Adaptive array antennas for 5.8 GHz vehicle to roadside communication. In *IEEE Vehicular Technology Conference (VTC)*, 1994.

[22] K. Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *ACM SIGCOMM*, 2003.

[23] W. Franz, R. Eberhardt, and T. Luckenbach. FleetNet - Internet on the Road. In *8th World Congress on Intelligent Transportation Systems (ITS)*, 2001.

[24] M. Gerla, B. Zhou, Y.-Z. Lee, F. Soldo, U. Lee, and G. Marfia. Vehicular Grid Communications: The Role of the Internet Infrastructure. In *Wireless Internet Conference (WICON)*, 2006.

[25] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *4th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2006.

[26] W. Kellerer. (Auto)Mobile Communication in a Heterogeneous and Converged World. *IEEE Personal Communications*, 8(6):41–47, 2001.

[27] Y. Ko, V. Shankarkumar, and N. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In *19th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2000.

[28] U. Lee, E. Magistretti, B. Zhou, M. Gerla, P. Bellavista, and A. Corradi. Efficient Data Harvesting in Mobile Sensor Platforms. In *Second IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS)*, 2006.

[29] A. Mishra, M. ho Shin, and W. A. Arbaugh. Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network. In *23rd Annual IEEE Conference on Computer Communications (INFOCOM)*, 2004.

[30] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto. CarNet: A Scalable Ad HocWireless Network System. In *9th ACM SIGOPS European Workshop*, 2000.

[31] A. Nandan, S. Das, G. Pau, M. Sanadidi, and M. Gerla. Cooperative downloading in Vehicular Ad Hoc Networks. In *2nd Annual Conference on Wireless On demand Network Systems and Services (WONS)*, 2005.

[32] V. Navda, A. Kashyap, and S. R. Das. Design and evaluation of iMesh: An infrastructure-mode wireless mesh network. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2005.

[33] M. Neufeld and D. Grunwald. Using phase array antennas with the 802.11 MAC protocol. In *1st International Conference on Broadband Networks (BroadNets)*, 2004.

[34] A. Nicholson, Y. Chawathe, M. Chen, B. Noble, and D. Wetherall. Improved Access Point Selection. In *4th International Conference on Mobile Systems, Applications, and Services (Mobisys)*, June 2006.

[35] J. Ott and D. Kutscher. Drive-thru Internet: IEEE 802.11b for Automobile Users. In *23rd Annual IEEE Conference on Computer Communications (INFOCOM)*, 2004.

[36] J. Ott and D. Kutscher. A Disconnection-Tolerant Transport for Drive-thru Internet Environments. In *24th Annual IEEE Conference on Computer Communications (INFOCOM)*, 2005.

[37] R. Ramanathan. On the Performance of Ad Hoc Networks with Beamforming Antennas. In *2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2001.

[38] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit. Ad Hoc Networking with Directional Antennas: A Complete System Solution. *IEEE Journal on Selected Areas in Communication (JSAC)*, 23(3):496–506, 2005.

[39] R. Sengupta and Q. Xu. DSRC for Safety Systems. In *Intellimotion–Research Updates in Intelligent Transportation Systems*, 2004.

[40] J. H. Winters. Signal Acquisition and Tracking with Adaptive Arrays in the Digital Mobile Radio System IS-54 with Flat Fading. *IEEE Transactions on Vehicular Technology*, 42(4):377–384, 1993.

[41] J. H. Winters. Smart antennas for Wireless Systems. *IEEE Personal Communications*, 5(1):23–27, 1998.

[42] J. H. Winters and M. J. Gans. Versus Phased Arrays in Mobile Radio Systems. *IEEE Transactions on Vehicular Technology*, 48(2):353–362, 1999.

[43] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-vehicle safety messaging in DSRC. In *1st ACM international workshop on Vehicular ad hoc networks (VANET)*, 2004.

[44] Y. Yamasaki, M. Yasunaga, Y. Murakami, and H. Moribe. A Beam Controlled Array Antenna for Road to Vehicle Communications. In *IEEE Intelligent Transportation Systems*, 2001.