

Enhancing Vehicular Internet Connectivity using Whitespaces, Heterogeneity, and a Scouting Radio

Tan Zhang
Univ. of Wisconsin-Madison
Madison, WI 53706, USA
tzhang@cs.wisc.edu

Sayandeep Sen^{*}
IBM Research India
Bangalore, KA 560045, India
sayandes@in.ibm.com

Suman Banerjee
Univ. of Wisconsin-Madison
Madison, WI 53706, USA
suman@cs.wisc.edu

ABSTRACT

We explore the use of TV whitespace communication systems for providing robust connectivity to vehicles. A key challenge in this setup is the *asymmetry in transmit power limits* – the fixed base station is allowed to communicate at up to 4 W, while the mobile gateways in vehicles are limited to 100 mW. This paper presents a specific solution to deal with this asymmetry in which whitespace transceivers are used in the downlink direction while a more traditional cellular path is used in the uplink one. While heterogeneous communication systems have been considered before (e.g., in some satellite networks), our solution explores some unique opportunities that arise in vehicular systems. In particular, we describe a system called *Scout* that uses a front radio at the head of a vehicle to look ahead and identify the best channel parameters to be used when the rear radio eventually reaches the forward post. We use these channel estimates to adapt a number of transmission mechanisms for improving the performance of flows through this heterogeneous network. We have implemented and deployed this system on moving vehicles in an urban environment, and demonstrated 3 – 8 \times performance improvement over simpler alternatives that do not use the scouting technique.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Wireless communication*

Keywords

TV Whitespaces; Vehicular Internet Access; Power Asymmetry; Heterogeneous Network; Scouting Radio

1. INTRODUCTION

There is a growing need to provide high bandwidth and robust Internet connectivity to vehicles for supporting diverse applications, e.g., improved traffic intelligence, transportation safety, and infotainment for passengers. Numerous research projects such as

^{*}The author was a graduate student at University of Wisconsin-Madison during the course of this work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiSys'14, June 16–19, 2014, Bretton Woods, NH, USA.
Copyright 2014 ACM 978-1-4503-2793-0/14/06 ...\$15.00.

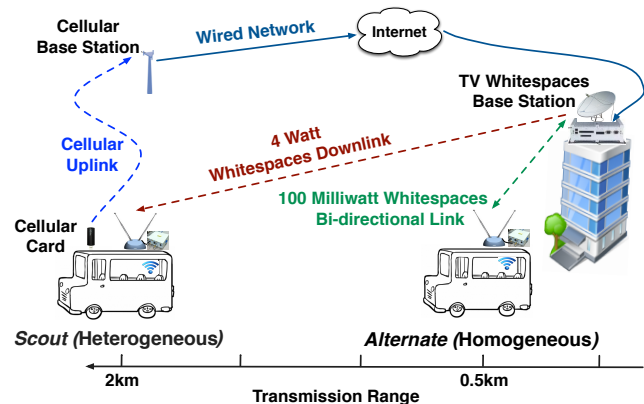


Figure 1: *Scout* uses TV whitespaces path primarily for downlink traffic and cellular path primarily for uplink, unlike a traditional, homogeneous design that uses the same path for both directions.

MAR [19], WiRover [10], ViFi [3], Wiffler [2], CaberNet [7], along with commercial endeavors strive to provide this today primarily using existing cellular technologies, and sometimes enhanced through opportunistic WiFi access. In this paper, we explore the feasibility and design challenges of using the emerging TV whitespaces spectrum for this application.

TV whitespaces (512 – 698MHz), recently released for unlicensed usage in some parts of the world (e.g., U.S. [9]), offer a new communication medium to wireless systems. The substantial spectrum resource (up to 180MHz) and good propagation characteristics (up to 30km) make this additional spectrum especially attractive for wide-area communications. In this work, we have explored the use of TV whitespaces spectrum not *merely* because of the low spectrum cost, but for reclaiming additional spectrum resources that is especially suited for long-range communications to match the needs of vehicular connectivity. Our near term goal of this project is to deploy an “on-board” Internet access service over a TV whitespaces network for commuters of a city metro transit operating hundreds of buses at Madison, WI, USA.

To efficiently utilize TV whitespaces, we propose a *heterogeneous* network design called *Scout* that (i) communicates the downlink traffic (usually the dominant fraction of traffic) primarily over TV whitespaces paths while sending the uplink traffic primarily over existing cellular paths, and (ii) uses an additional “scouting” radio in the vehicle to probe the channel condition in advance, thereby compensating for the high feedback latency in making protocol decisions.

Power	Range (km)
100 mW	0.4 – 0.5
4 W	1.9 – 2

Table 1: Transmission range of our whitespace radio at different transmit powers when operating at 1 Mbps PHY rate.

Figure 1 (left side) shows the *Scout* architecture. Gateway nodes (clients) on buses are each equipped with two TV whitespaces radios and one cellular radio. Each of these radios communicates with its corresponding base station — the whitespaces radios with our own whitespace base stations and the cellular radio with the usual commercial cellular base stations. The gateway serves as a WiFi hotspot inside the bus, allowing users to connect to the Internet through their WiFi-capable devices. This path diversity, however, is hidden from end applications through an encapsulation tunnel between the gateway and an aggregation proxy situated behind the TV whitespaces and the cellular networks. We contrast this design of *Scout* to a simpler symmetric network where the TV whitespaces path alone is used for both uplink and downlink communications, such as in the WhiteFi system [1] (Figure 1 right side).

Why a heterogeneous network design? Most of communication systems typically use a single network path for transferring both uplink and downlink traffic. Unfortunately, applying this design convention to our vehicular context significantly limits the coverage of the base stations over whitespaces. As per FCC’s ruling [9], the transmission power of the mobile whitespace devices (gateway nodes on the bus) is limited to 100 mW, whereas the power of static base stations can be up to 4 W¹. This 40× difference in the transmit power limit is to prevent mobile devices from causing harmful interference to the primary incumbents during roaming. Since most of the communication protocols need to be bi-directional, a whitespace-only network design as shown in Figure 1 (right side) would limit the operating range of a whitespace path to that of the “weaker” mobile clients.

Experiments with our whitespaces radios show that the operating range of a symmetric, homogeneous system using 100 mW transmit power would be 4× less than our proposed dual network design that uses 4 W power for downlink alone. This is shown in Table 1². While the operating range can be improved somewhat by choosing a lower rate on the uplink or using more sophisticated signal processing techniques, a significant link asymmetry is expected to remain. To achieve the same coverage, the lower bi-directional range means much more whitespace base stations to be deployed, leading to higher infrastructure and management cost.

Scout circumvents the problem of “weak” whitespaces uplink by adopting a heterogeneous network architecture in which the downlink traffic (from the base station to the vehicle) is mainly communicated over TV whitespaces, while the uplink traffic (from the vehicle to the base station) is sent over a cellular link. This can maximize the downlink coverage of each whitespace base station by leveraging the extensive cellular connectivity in the uplink. Furthermore, since many networking applications are downlink dominated (10× in WiRover [10]), *Scout* is efficient in utilizing TV whitespaces for relaying traffic.

¹This power limit includes the gain of antennas and regardless of number of channels used for transmission.

²The measured transmission ranges are similar to a prior report [6] based on a different hardware platform.

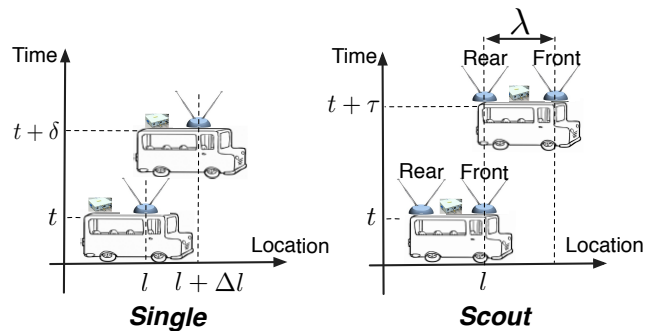


Figure 2: Diagram of *Single* and *Scout*. λ is the antenna separation, τ is the rear radio’s travel time to front position, and δ is the cellular path delay ($\delta \leq \tau$).

Why the use of a scouting radio? While effectively extending the coverage, a heterogeneous network based on traditional networking protocols yields poor downlink performance due to the *high feedback latency* in the cellular uplink. First, the stale feedback leads to low accuracy of channel estimation, especially in the mobile environment. Since most of communication systems rely on channel estimation for making *various* protocol decisions, e.g., rate adaptation, FEC, etc., the poor estimation lead to higher packet losses. Second, the feedback delay drastically inflates the bandwidth delay product rendering the performance of TCP based applications extremely susceptible to channel losses. Third, the delayed feedback severely slows down retransmissions that are primarily used in traditional wireless systems for loss recovery. This in turn leads to high bandwidth inefficiency. To tackle these problems caused by the slow feedback, we leverage an extra “scouting” radio to accurately measure channel condition for a location in advance. This information is then used by several aggressive transmission techniques to enhance link performance by minimizing end-to-end loss.

The core intuition of “channel scouting” comes from the simple observation that the reception location of a radio largely determines its experienced channel characteristics [13, 16, 20]. Thus, in a single-radio system as shown in Figure 2 (*Single*), the radio’s actual channel experience in a new location would be different from the feedback conveyed at the old location by the time our base station acts on this feedback. This is especially true with a cellular uplink with at least tens of milliseconds delay. In contrast, if we were to place two radios as shown in Figure 2 (*Scout*), the channel condition experienced by the front radio would be somewhat similar to that of the rear radio after it moves forward a short time later. In essence, the front radio can “scout” the likely channel condition for future receptions by the rear radio. Thus, we can send back the observed channel condition by the front radio at its current location l . By the time the rear radio reaches the same location l , the base station can use this feedback made earlier by the front radio to choose better transmission parameters for the rear radio. While the underlying intuition is simple, to successfully realize the benefits of the channel scouting in real settings, we had to address challenges such as identifying relevant feedback to a given location and handling varying vehicle speed and driving patterns, which will be discussed in § 3.1.

Built upon this scouting based channel estimation framework, we design multiple transmission adaptation techniques, i.e., rate adaptation, inter-packet FEC, and intelligent traffic duplication, to enhance the reliability of vehicular connectivity. Since TCP is the dominant transport protocol used by most applications, we demonstrate the efficacy of the combination of our proposed mechanisms by showing how they can improve TCP performance.

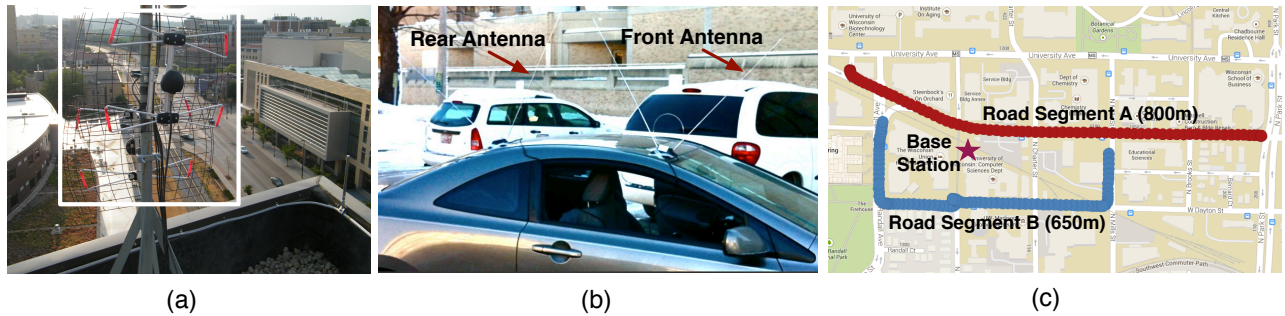


Figure 3: Outdoor deployment at the campus area of Madison, WI: (a) whitespace base station deployed on top of a 8-floor building; (b) vehicular client with two TV antennas mounted atop; (c) two example road segments used in experiments.

We note our dual radio approach is quite successful for a moving vehicle with adequate length ($\geq 1.5\text{m}$). However, this approach as designed, is not relevant to static settings (§ 5.3), or for a platform with limited physical size such as a smart phone. In addition, our use of a scouting radio is *independent* to multi-antenna techniques (MIMO) that combine signals at the physical layer for scaling throughput or enhancing reception robustness. Our scouting radio operates at the MAC and higher layers, and is primarily used to determine channel properties at a reception location where the rear radio visits later. Furthermore, our scouting technique is *complementary* to MIMO, and a significant performance gain can be achieved when a scouting radio is leveraged by MIMO for compensating feedback delay as will be shown in § 5.1.2.

We also note that existing cellular technologies (3G, 4G, etc.) have addressed the problem of power asymmetry in the *licensed* band. The key technique used is *power equalization*, which enables the base station to manage interference among multiple mobile clients by tightly controlling their transmission power [8, 23]. This ensures that the base station receives uplink signals at similar power from different clients, thus being able to decode each one through an amplification process. Unfortunately, imposing centralized power control over the *uncoordinated and opportunistic* TV whitespaces appears to be difficult. Since the whitespaces spectrum is intended for uncoordinated usage, a base station, by design, would be incapable of controlling other unlicensed devices potentially operating with a different communication protocol. The unrestricted interference is likely to overwhelm the weak uplink signal, causing decoding at the base station to fail.

Finally, heterogeneous networks have been widely used in satellite communications in which the downlink connection is established over the satellite link and the uplink is over dial-up lines. While similar in approach, *Scout* is a stylized design to meet the asymmetric power constraints in TV whitespaces. More importantly, *Scout* can *compensate* the disadvantage in the slow cellular uplink through the use of a scouting radio.

Contributions: The key contributions of our work are:

- We present a stylized design of a heterogeneous network structure, which effectively uses TV whitespaces for providing wide-area network connectivity to vehicles. Our system avoids a denser deployment of whitespaces base stations that are otherwise needed in a whitespaces-only network.
- To address the problem of feedback delay over the cellular uplink, we propose a channel estimation technique enabled by an additional scouting radio suited specifically for moving vehicles. We further design and implement a functional system that leverages this channel estimation for three transmis-

sion techniques, i.e., rate adaptation, inter-packet FEC and intelligent traffic duplication over cellular links. Based on our experiments (conducted by driving about 500 miles on different routes around our whitespace base station), *Scout* can enhance TCP performance by $3 - 8\times$ over alternative SISO and MIMO systems.

In the rest of the paper, we first validate the feasibility and advantage of our scouting radio approach in § 2. In § 3, we present the design of *Scout* in detail. We then outline some implementation details in § 4, followed by our evaluation of *Scout* in § 5.

2. ADVANTAGES OF A SCOUTING RADIO

Most wireless communication systems use rate adaptation and link-layer retransmission (or their variants) for adapting to changes in channel conditions. Rate adaptation enables the sender to choose a combination of modulation and channel coding suitable for the channel characteristics experienced by the receiver. Link-layer retransmission allows the sender to retransmit lost frames to avoid their negative impact at higher layers, e.g., TCP. In this section, we will show the limitations of a single radio system under mobility in making these protocol decisions. We then demonstrate the advantage of leveraging a scouting radio in obtaining better channel estimates for improving such decisions. We start by describing our experiment setup.

2.1 Experiment setup

Our testbed is built around the UW campus area with a coverage radius of several kilometers. It currently consists of one base station and several mobile clients. The base station is mounted atop a 8-floor building as shown in Figure 3(a). It is equipped with an Ethernet connection for receiving the uplink traffic and relaying it to the Internet on behalf of the client. It performs downlink transmission using a whitespaces radio, a high-gain power amplifier, and a directional TV antenna. The total transmission power is set to be 3.8 W for extensive downlink coverage. To avoid interfering to primary incumbents, we configure the base station to operate across 4 contiguous, unused TV channels (at a center frequency of 662MHz) as indicated by a spectrum occupancy database [26]. As the mobile client for experiments, we use a personal vehicle carrying two whitespaces radios to receive downlink traffic as shown in Figure 3(b). Each radio is connected to an omni-directional antenna for capturing signal from all directions. A 3G cellular card is used for uplink communications.

Figure 3(c) shows the two of many road segments where our experiments were conducted. These two road segments have very different coverage characteristics — *segment A* is part of one major

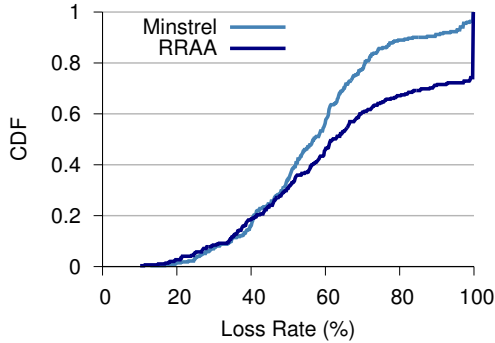


Figure 4: CDF of MAC layer losses of two state-of-the-art rate adaptation algorithms with mobility and under feedback delay in the cellular uplink.

bus route we intend to cover. It has good connectivity to the whitespaces base station with more than 70% of the path in line-of-sight to the base station’s antenna. In contrast, the *segment B* represents a worse coverage scenario, with more than 60% of the path blocked by several tall buildings. The mobile client drove around this area at speeds between 18 and 35 km/hr (governed by city speed limits).

Our radio platform is similar to the WhiteFi radio [1]. It performs a frequency translation function by converting the signal from commercial WiFi cards to the UHF band. We will describe this platform for more details in § 4.1. Note that most of the techniques in *Scout* are not limited to the 802.11 baseband technology, but applicable to other technologies (WiMax, LTE, etc) as well.

2.2 Limitations of a single-radio system under vehicular mobility

We now discuss how vehicular mobility in a single radio setup impacts the accuracy of rate adaptation and the efficiency of link layer retransmission. In particular, this impact is *exacerbated by the feedback delays* in the cellular uplink.

Rate selection mismatch with mobility: We conducted experiments where we drove a vehicle along road segment A multiple times at speeds of about 35 km/hr. The vehicular gateway sent the feedback over cellular links to the TV whitespace base station for selecting downlink data rates. As the vehicle kept moving, the usable PHY rates relevant to a given location changed quickly. Thus, by the time the base station could act on a received feedback, the location of the receiving radio on the vehicle had changed and the feedback become stale. As a result, any rate selection algorithm that depends on such feedback experienced poor performance. To illustrate this, Figure 4 presents the MAC layer loss rates for two state-of-the-art rate adaptation algorithms (Minstrel [4] and RRAA [29]) calculated over 100 millisecond intervals. We observe a median loss rate of 57% and 62% for each algorithm based on the delayed feedback.

Retransmission overhead: While MAC layer retransmissions can hide some of the packet losses, they introduce bandwidth inefficiency and additional delay that can also limit the performance of a system. Based on our testbed measurements, the current cellular paths (3G and 4G) have a minimum delay of 25 milliseconds and a typical delay between 50 and 150 milliseconds. Each retransmission will therefore inflate the end-to-end path latency by a corresponding amount, significantly cutting down TCP throughput. As will be shown in § 5.1, for a single radio system using a combination of a state-of-the-art rate adaptation algorithm (RRAA), link

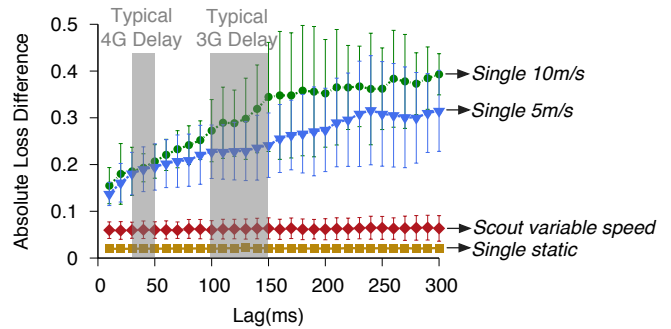


Figure 5: Average of absolute loss differences at various lags for 12Mbps packets. A lag is the elapsed time between two measurements. *Single* denotes one radio and *Scout* is the two radio setup.

layer retransmissions, and a cellular uplink, the achieved downlink TCP throughput can be quite low (about 40 Kbps). In contrast, our alternative design is able to provide a performance more than two orders of magnitude better in many cases. We next validate the intuition of the scouting based channel estimation, which is the *core* technique used in *Scout*.

2.3 Using a scouting radio to improve channel estimation

The advantage of the scouting based channel estimation can be ascertained by comparing the following two schemes as shown in Figure 2. In *Scout*, suppose the front radio measures the loss property at location l , time t . How accurate is this measurement in predicting the channel condition for the rear radio when it reaches the same location l at time $t + \tau$? We contrast this with the alternative design of a single radio (*Single*). In this single radio setup, the only radio will measure the loss property at location l , time t , and use this estimate to predict the channel condition at location $l + \Delta l$, time $t + \delta$.

Metric: We used the packet loss rate as an indicator of channel quality for a given location and at a given time. Each loss rate was calculated for 10 contiguous packets at a fixed PHY rate. We then measured the magnitude of difference in loss rates under different time separations to classify whether channel condition has changed with varying location or time (or both). We denote this time separation as a *lag* in the following discussion.

Evaluation: To understand the stability of channel loss properties as only a function of time, we present the variation of loss rates at same locations with different lags. We measured this by placing a single radio mounted atop a car at 12 equally spaced locations on the road segment A. We then averaged the absolute differences for pairs of loss rates separated by each time lag at all these locations. Figure 5 (*Single static*) shows that the variation of loss rates remains small with a lag below 300 milliseconds for all the measured locations.

We next determine the stability of loss measurements done by the same single radio as a function of both time and location. The speed of the vehicle in these experiments was between 5m/s (18 km/hr) and 10m/s (35 km/hr), which is typical for urban area due to the 40km/hr speed limit. As can be seen from Figure 5 (*Single 10m/s* and *5m/s*), the difference in loss rates increased drastically with increasing lags. The degree of variation is expected as the single radio system was measuring the loss rates at different locations and different time. When using the stale channel observation

to predict the loss rate, *Single* would make an estimation error of over 30% under the typical delay of a 3G uplink (100 to 150 milliseconds), and over 20% under the delay of a 4G link (25 to 50 milliseconds). Note that these delay values were obtained from our testbed measurement as mentioned before.

We finally benchmark the mismatch in the loss rates under *Scout* setup with two radios (front and rear) aligned at the same locations under various lags introduced by different vehicle speeds. The result is again shown in Figure 5 (*Scout variable speed*). For a lag of 300 milliseconds, we note that the difference in loss rates between two radios at the same location remained a fifth of a single radio traveling at 10 m/s speed for a 3G uplink, and within a third for a 4G uplink. This demonstrates that *Scout* can indeed improve the channel estimation for the rear radio.

While our vehicle was driving at a city speed of 18 – 50 km/hr for the experiments reported in this paper, we note much higher speed (up to 120 km/hr) can be supported by *Scout*. In particular, the average length of a car in the US is 5 meter, which sets the maximum limit for the separation of two radios at the client. Assuming this 5 meter separation and a highest vehicular speed of 120 km/hr (33 m/s), it takes at least 150 milliseconds for the rear radio to reach the front radio’s location. This delay is typically higher than the observed latency in the cellular links (3G and 4G), thus allowing the base station to receive the front radio’s feedback usually before the rear radio reaches the front position.

3. SCOUT: A HETEROGENEOUS DESIGN

Scout adopts a heterogeneous network architecture to take most advantage of power asymmetry between the whitespace base stations and vehicular clients. In this network, the downlink communication is mainly conducted over TV whitespaces with the maximum allowed power, thereby achieving extensive coverage by our whitespace base stations. For uplink communications, we choose to use existing cellular path to circumvent the limited transmission range in TV whitespaces. We further allow downlink traffic to be intelligently duplicated over the cellular path when the whitespace link is experiencing an outage³.

Specifically, we have developed following related mechanisms to enhance whitespace connectivity — an efficient way of estimating channel properties (of the rear radio, using a forward radio) and several transmission techniques based on the obtained channel estimates, i.e., rate adaptation, forward error correction (FEC), and intelligent traffic duplication over cellular links. For uplink communications, we choose to provide best-effort delivery because the loss rates and packet reordering are observed to be very low in the cellular path. Figure 6 shows the overall architecture of *Scout*.

To send downlink traffic, the base station first obtains channel quality estimates based on the feedback generated for both radios by the client (§ 3.1). It then chooses an appropriate link-layer data rate using the obtained channel estimates (§ 3.2.1). Based on the error performance of the chosen data rate, it constructs coded packets for each batch of data packets for forward error correction (§ 3.2.2), then transmitting them over TV whitespaces at the selected rate. Upon detecting connection blackouts in the whitespace link, the base station also sends the data packets in duplicate over the cellular path (§ 3.2.3).

At the client (vehicular gateway), we leverage the dual-radio diversity by combining the coded packets from both radios (§ 3.3).

³In fact, our system is designed to be opportunistic in both communication directions, e.g., if the vehicle is within the uplink range to a whitespace base station, we also allow uplink traffic to be sent over TV whitespaces.

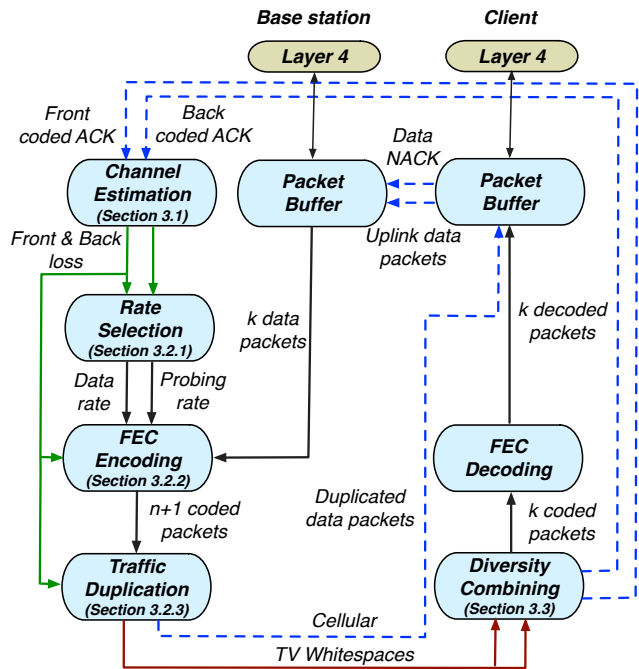


Figure 6: System architecture of Scout.

The client then decodes these packets to obtain data packets. If duplicated packets are received on the cellular path, the client also merges them with the data packets, then passing all the packets received in order to the application. To provide the base station with channel estimates and information about lost packets for retransmission, the client sends feedback for both coded packets and data packets over the cellular link. In addition, the client reports its GPS information periodically to the base station for obtaining relevant feedback to the rear radio’s reception location. The pseudocode of *Scout* is shown in Algorithm 1 (appendix). We now explain each of these components in detail.

3.1 Scouting radio based channel estimation

The first and the most important step in the transmission process is for the base station to obtain accurate channel estimates for making various transmission decisions. This is performed by the *ScoutEstimate* procedure in Algorithm 1.

Channel estimation: We use the *loss rates of coded packets* as the estimator of channel quality because such information is readily accessible from the commercial WiFi cards used in our implementation. Note that our proposed technique can be used to collect other types of channel statistics such as channel state information (CSI) and bit error rates in a similar way. To estimate the packet loss rates, the base station first collects feedback generated for the *coded packets* received by both radios at the client. For each PHY rate r , the base station calculates the loss rate $L_f(r)$ for the front radio based on the packets most recently ACKed for the front radio. It also calculates the loss rate $L_r(r)$ for the rear radio based on the packets previously received by the front radio at the current location l of the rear radio. If no coded packets are sent when the front radio was at l , the base station will use the delayed ACKs for the rear radio to calculate $L_r(r)$. To obtain sufficient, yet relevant channel estimates, we choose a time window of 25ms for calculating both loss rates. As a safeguard against random channel fluctuations, we apply exponential averaging with a heavy weight (0.85) on the current estimates to compute the final channel estimates.

Feedback alignment: To accurately estimate the channel quality at a given location l , it is critical to identify a relevant set of coded packets that are sent to the front radio at l . As shown in Figure 5, the loss rate estimation can be off by about 40% due to a 3 meter location error (300 milliseconds lag at 10 m/s speed). Unfortunately, our low priced GPS modules used at the clients have a positioning inaccuracy up to 10 meter.

To circumvent this problem, we use the speed reading v instead of the coordinate reading reported by the GPS because it has much lower error (0.1 m/s). Using this speed reading, the base station obtains the radio alignment period τ , which is the time elapsed for the rear radio to reach the location l since the front radio was previously at l (Figure 2). This period is calculated by $\tau = \lambda/v$ where λ is the antenna separation of two radios, which is set to be 1.5 m in our implementation. Note that v remains constant because the GPS reading is updated at short intervals (1 second). The base station then obtains coded packets that are received by the front radio τ time ago for estimating $L_r(r)$ at location l . Since τ is usually below 300 milliseconds (1.5m / 5m/s), the feedback alignment error in *Scout* is less than 0.03 meters (0.1 m/s \times 300 milliseconds).

Finally, for the occasional cases when the vehicle stops ($v = 0$), the base station uses the delayed ACKs for the rear radio to calculate $L_r(r)$ because they remain relevant to this radio's reception location. The base station will switch back to use the scouting feedback upon detecting the vehicle to move again. The change of vehicle mobility can be quickly determined by the GPS reading that is updated by the client frequently (every second).

3.2 Utilizing Scout channel estimates

In *Scout*, we utilize the obtained channel quality estimates for improving rate adaptation, forward error correction and intelligent traffic duplication over cellular links. We present the details of these mechanisms next.

3.2.1 PHY rate selection

We start by describing how to use the channel estimates from the scouting radio to choose the first transmission parameter – PHY rates at the base station. Our algorithm – *ScoutRate* proceeds in three steps as shown in Algorithm 1.

The first step is to identify a candidate set of reliable rates C for the rear radio. To this end, we examine the loss rate at both radios – $L_f(r)$ and $L_r(r)$ for each PHY rate r . We only include those PHY rates with both loss rates below a threshold (0.65) into C . This is because our testbed measurements show that higher loss rates usually have large variation, which makes accurately estimating loss properties difficult.

In the second step, we calculate the achievable throughput at the rear radio $T = r \times (1 - L_r(r))$ for each data rate in the candidate set C . We select the rate r_{data} with the highest throughput, and assign it to a batch of coded packets as will be described in § 3.2.2.

Finally, we append an additional *coded* packet at each batch of packets to randomly probe a different rate r_{probe} . The probed rate should be higher than the throughput of the current data rate, but not necessarily in C . Thus, *Scout* can quickly adapt to the improved channel condition through random probing, while promptly adapting to channel degradation via either probing or identifying the change of the candidate rate set.

3.2.2 Inter-packet forward error correction

We now explain the next step of the transmission process that inserts redundant packets into each batch of data packets for forward error correction. All the coded packets are transmitted using

the *same* selected rate r_{data} over TV whitespaces. This process is shown by *ScoutFEC* in Algorithm 1.

Inter-packet FEC: To circumvent the high retransmission inefficiency caused by a slow cellular uplink as mentioned in § 2.2, we leverage *inter-packet* forward error correction as the primary mean of error recovery. The base station first groups data packets to be transmitted into batches. It then encodes each batch of k data packets into n coded packets for transmission. Upon receiving *any* k coded packets, the client can decode the original k data packets.

Our motivation of performing encoding *across* packets comes from the empirical analysis of loss characteristics in the vehicular testbed. In particular, we find that most of packet losses (>80%) result from the *failure of detecting packets* at the receiver, rather than *bits corruption*. Two factors might contribute to this fact. First, our vehicular network operates in an urban area with many tall buildings that completely blocks off the whitespaces transmission in certain road segments. Second, the intra-packet FEC used in the 802.11 baseband technology is effective in correcting bit errors within a packet. Thus, we use the inter-packet FEC to complement the intra-packet FEC in the 802.11 technology. We will show in § 5.2.2 that the above choice leads to significant performance gain.

We choose a type of erasure codes based on Vandermonde encoding matrices [18] in our implementation, which incurs low computational latency (order of microseconds). The specific erasure codes provide a unique advantage by having data packets as their first k coded packets, denoted as D . Upon successfully receiving all the packets in D , the receiver can directly pass them to the higher layer without incurring any decoding delay. More importantly, when less than k coded packets are received, the receiver can scavenge whatever packets received in D to reduce the penalty of decoding failure.

Redundancy estimation: To determine the number of data packets k and that of coded packets n to be included in a given batch, we calculate the effective loss rates L after the diversity combining at the client. Assuming independent channel losses at both radios, L should be calculated by $L = L_f(r_{data}) \times L_r(r_{data})$, where $L_f(r_{data})$ and $L_r(r_{data})$ are the loss rates of the selected PHY rate r_{data} estimated for both radios. We then determine the *redundancy ratio* rr , which are defined as $rr = (n - k)/n$. rr should be no less than the loss rate L for successful decoding the entire batch. In our implementation, we insert 5% extra redundancy to be conservative, thus $rr = L \times (1 + 0.05)$.

Using the calculated rr , we now determine k and n . n is chosen such that the transmission duration of all the coded packets are within the channel coherence time c (25ms based on our measurements). This is due to our design assumption that all the packets within a batch should experience similar degree of channel losses, thus sharing the same data rate and redundancy. We calculate n by $n = c \times r_{data}/s$, where s is the size of each encoded packet. After determining n , we calculate k by $k = n \times (1 - rr)$. Finally, we append one additional coded packet at the end of each batch for probing a different data rate as mentioned in § 3.2.1.

3.2.3 Intelligent traffic duplication over cellular path

We now present the third transmission technique in *Scout* that intelligently duplicates some downlink packets over the cellular link when necessary to bridge coverage holes. This is shown in the *ScoutDUP* procedure of Algorithm 1.

Our decision of traffic duplication is motivated by non-negligible frequency of connection blackouts (11% for road segment A) observed at the mobile client. Some of the connectivity losses can last for long periods of time (up to 5 seconds), causing a TCP connection to timeout before all the recovery mechanisms over whites-

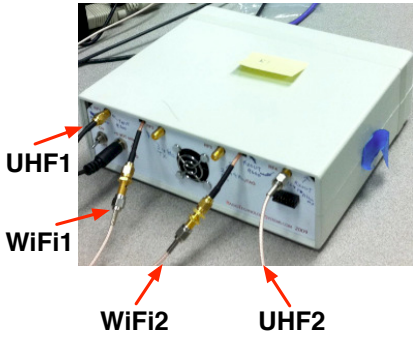


Figure 7: Scout wide band digital radio.

paces to take effect. To effectively handle connection blackouts, the base station duplicates traffic on both the whitespace and cellular paths upon detecting intermittent connectivity over TV whitespaces. The reason we duplicate the downlink traffic, instead of sending it solely on the cellular path, is to accurately determine when the connectivity is recovered over whitespace, and stop sending on the cellular path from that point. Furthermore, we only duplicate the k data packets in each batch to reduce the cellular usage, since the loss rates in the cellular path are observed to be very low as mentioned before.

The intelligent duplication is triggered when either of the following criterion is met. *Criteria I*: the current data rate r_{data} selected by the base station is the lowest rate (1Mbps) and the corresponding loss reported by the scouting radio is greater than a duplication threshold (0.75). This is to minimize the cellular usage by having the rate adaptation algorithm to decrease the PHY rate first. *Criteria II*: the downlink packet is a retransmission packet. This attempts to minimize further retransmissions, thereby alleviating the effect of head-of-line blocking at the client.

3.3 Additional techniques

We describe two additional techniques used in *Scout*, i.e., dual-radio diversity combining and NACK based retransmissions. These two techniques, albeit simple, are useful in combating end-to-end losses as shown in § 5.2.

Dual-radio diversity combining: Apart from channel estimation, we leverage the additional scouting radio to enhance reception diversity. To this end, we combine the *coded* packets received at both radios, passing those *unique* packets for FEC decoding. Since two radios are likely to capture a different set of packets, a phenomenon widely known as reception diversity [12], combining packets between them can effectively enhance the reception performance.

NACK based retransmission: In *Scout*, the base station and the client follow a sliding-window protocol to provide in-order and optionally reliable delivery in downlink communication. For uplink, *Scout* provides best-effort delivery due to the trivial loss rates observed in the cellular links as discussed. To avoid saturating the cellular path, the client sends a NACK packet for a block of downlink data packets periodically (every 50ms). The base station retransmits each lost packet for a maximum retry count before discarding it. The retry limit is tunable to accommodate various application requirements.

4. IMPLEMENTATION

We describe the hardware platform and the software architecture of *Scout*, along with the rate adaptation algorithms implemented for comparison.

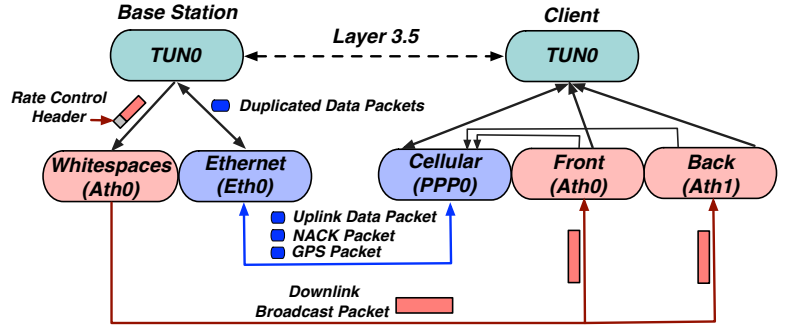


Figure 8: Software architecture of Scout.

4.1 Hardware platform

Our radio platform, called the *Scout Wide Band Digital Radio* (WDR), is shown in Figure 7. It enables the whitespaces communication by translating the WiFi signals to the UHF band. To support two WiFi radios at the client, we used an enhanced version of WDR equipped with two independent signal processing paths, each of which can translate a stream of the WiFi signal. This allows the use of one WDR for two radios at each client to reduce the hardware cost. Due to some limitations in the RF front-end, the current version of WDR can support 802.11 b/g data rates up to 18Mbps.

4.2 Software architecture

We implement *Scout* with a userspace program running at the 3.5 layer of the networking stack on top of the cellular and whitespaces links. Figure 8 shows our software architecture. We have written this program in about 7000 lines of C++ and added 50 lines of C code to modify the Ath5k and Ath9k driver. The salient characteristics of our implementation are listed below.

Single-link abstraction: To allow traditional applications to run over two heterogeneous links without modifications, we expose the abstraction of a single link using a virtual network device in Linux called *TUN*. All the applications send traffic through this fake device, which is then intercepted by our user-space program, and redirected through UDP tunneling over the underlying physical links (i.e., whitespaces and cellular).

As shown in Figure 8, for downlink communication, the server program at the base station first reads the application traffic from *TUN* into a packet buffer. It performs all the signal processing as described in § 3, sending the encoded packets through the whitespaces interface (Ath0). It also sends the original data packets via the Ethernet interface (Eth0) upon detecting the outage of the whitespace link. The client program receives the downlink packets from both whitespace interfaces (Ath0, Ath1), and occasionally from the cellular interface (PPP0). It decapsulates these packets and performs various decoding steps as mentioned in § 3. After decoding, the client collates all the unique data packets into the packet buffer, passing those received in-order through *TUN* to the application layer. The uplink communication is conducted in a similar fashion by having the client program send the uplink traffic over the cellular interface, which will be received at the Ethernet interface of the base station, and passed through the *TUN* interface by the server program.

Broadcast-based downlink communication: We configure the base station to broadcast downlink traffic to all the clients. The client program is responsible for discarding those packets destined to a different client based on their IP address after decapsulation.

Section	Experiment Setup	Compared Algorithms	Scout Gains
Overall Performance			
5.1.1	Whitespace + cellular, road A	<i>Scout</i> v.s. A1 – A4	23 – 120× over single radio, 3 – 8× over dual radios
5.1.2	WiFi + cellular, road A	<i>Scout</i> v.s. MIMO	Scout(2-SISO-Radios): 4×, Scout(2-MIMO-Radios): 7×
5.1.3	Whitespace + cellular, road A (video streaming)	<i>Scout</i> v.s. A4	90% reduction in buffering delay
5.1.4	Whitespace + cellular, road B	<i>Scout</i> v.s. A3 – A4	9× at 75th quartile
Microbenchmarks			
5.2.1	Whitespace + cellular, road A	ScoutRate v.s. RRAA and Minstrel	3.6× and 1.3× (with diversity)
5.2.2	Whitespace + cellular, road A	FEC-w-Scouting v.s. No-FEC, FEC-w/o-Scouting	1.5× over No-FEC, 1.3× over FEC-w/o-Scouting
5.2.3	Whitespace + cellular, road A	<i>Scout</i> v.s. No-DUP, DUP-w/o-Scouting	1.6× over No-DUP, 1.3× over DUP-w/o-Scouting
5.3	WiFi + cellular, indoor	<i>Scout</i> v.s. A3 – A4	0.8× (under-perform)

Table 2: Summary of main evaluation results. Performance gain is reported for TCP throughput unless mentioned otherwise.

There are three reasons for this decision. First, the 802.11 protocol disables the link-layer retransmissions in the broadcast mode. Thus, *Scout* can make its own retransmission decisions based on the NACK packets sent over the cellular link⁴. Second, the broadcast packets can be received by both radios at a client for channel estimation and multi-radio diversity combining. Third, each client can even overhear other clients’ downlink traffic for channel estimation. Here we assume that the security is assured by higher layer protocols, e.g., https. We modify the WiFi driver so that the server program can use a specific header field to control the PHY rate of each broadcast packet.

Dual sequence number spaces: *Scout* uses two different spaces of sequence numbers for data packets and coded packets respectively. The *raw sequence numbers* are used to index coded packets for channel estimation. In contrast, the *data sequence numbers* are used to identify data packets for link-layer retransmission. The client program detects packet loss based on the gaps in the sequence numbers of received packets. It generates different types of NACKs, which contain the sequence numbers of the lost data packets and those of missing coded packets respectively.

4.3 Rate adaptation algorithm

We select RRAA [29] and Minstrel [25] as representative rate adaptation algorithms implemented in the alternative systems for comparison with *Scout*. Minstrel is an enhanced version of SampleRate [4], which can respond faster to the channel variation and is the default algorithm used by the Ath5k driver. We further implement a Minstrel variant – Minstrel-HT [24] used by the Ath9k driver for MIMO rate adaptation.

While we experiment with only two rate adaptation algorithms, we expect *any* algorithm that requires accurate channel estimates at the client to be benefited from the scouting feedback in a heterogeneous network. Examples of such algorithms include SNR-triggered protocols [5, 11], which depend on accurate SNR measurements at the receiver to perform well. Similarly, for recently proposed schemes such as SoftRate [28] and AccuRate [21] that require PHY layer feedback from the receiver, and SensorHints [17] that uses mobile sensor inputs to detect the receiver mobility, adoption of the scouting feedback over stale feedback should lead to better performance. Exploring the complexity of adapting all the above algorithms for *Scout*, albeit challenging, is beyond the scope of our current work.

⁴If the unicast communication is used otherwise, the WiFi-based radio at the base station would perform unnecessary link-layer retransmission up to the maximum retry limit, since it cannot receive any link-layer ACKs over the weak whitespace uplink.

Alternative Systems	Rate Adaptation	Dual-radio Combining
A-1	RRAA	No
A-2	Minstrel	No
A-3	RRAA	Yes
A-4	Minstrel	Yes

Table 3: Alternative systems to be compared against *Scout*.

5. EVALUATION

In this section, we evaluate the performance gains by exploiting the scouting feedback to optimize multiple protocol decisions, which include rate adaptation, inter-packet FEC, and intelligent traffic duplication implemented in our prototype system. For the experiments, we have driven about 500 miles in the neighborhood of our whitespaces base station across various road segments. We primarily use two road segments shown in Figure 3(c) in this evaluation. Table 2 summarizes our main results.

Evaluation metrics: We use a variety of metrics to evaluate the link performance such as TCP throughput, end-to-end packet loss rates, buffering delay of video streaming, etc.

Compared systems: As discussed in § 4.3, we use RRAA and Minstrel as representative algorithms that use packet error rates at the receiver to select PHY rates. Table 3 describes four alternative systems implemented upon these algorithms and using SISO radios for downlink communication. A-1 and A-2 use RRAA and Minstrel respectively. A-3 and A-4 are dual-radio variants of A-1 and A-2, which also instantiate the prior concept of multi-radio diversity (MRD [12]) as done in *Scout*. We further implement a MIMO version of A-2 based on Minstrel-HT to select MIMO PHY rates. All the above systems use link-layer retransmission with a retry limit of 4.

Data collection: We collected measurements from 5 drives for each algorithm along these routes in all the experiments unless mentioned otherwise. The testbed is the same as described in § 2.1.

5.1 Overall performance of Scout

We evaluate the overall performance of *Scout* in supporting TCP downloading and video streaming on the two representative road segments. We quantify the performance gain of *Scout* over the alternative systems using SISO and MIMO radios, respectively.

5.1.1 TCP throughput (road segment A)

Method: We measured the downlink TCP performance of experiments conducted on road segment A, which has a better coverage from our whitespaces base station. The duration of each drive was 60 seconds approximately, and the results were averaged over 1 second bins for each system.

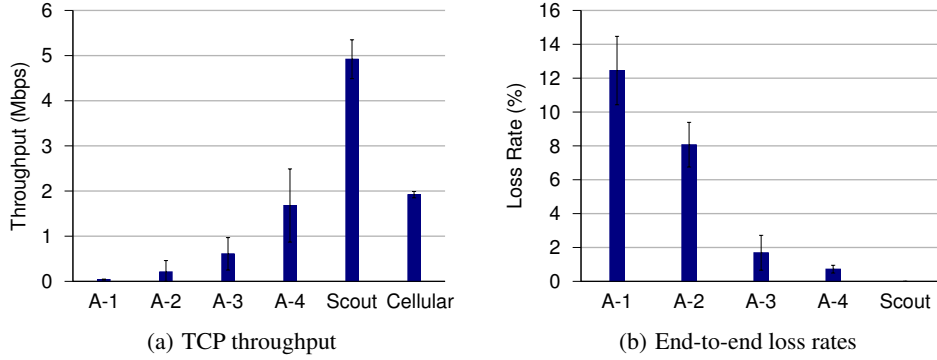


Figure 9: Average downlink TCP performance on road segment A. The error bar shows standard deviation.

Results: Figure 9(a) shows that *Scout* achieved $8\times$ and $3\times$ improvement in TCP throughput over A-3 and A-4, both of which used dual-radio diversity combining in the downlink path. These dual-radio systems outperformed their single-radio counterparts by $15\times$ and $8\times$, and hence it is sufficient to focus on the dual-radio versions alone. We also provide the TCP throughput of an existing 3G cellular link just for reference. Note that a direct comparison between *Scout* and the cellular link is not useful since the quality of hardware platforms and the load at the base station are different. We envision much higher throughput to be possible in a commercially designed whitespace system with improved antennas, sophisticated placement, and a radio incorporating higher PHY rates and various signal processing optimizations.

To gain some insights behind the throughput performance, we analyzed the end-to-end loss rate and the congestion window size of each TCP connection. Figure 9(b) shows that *Scout* can completely eliminate the end-to-end loss, whereas A-3 and A-4 suffered from 1.69% and 0.72% loss. Thus, we find that the average congestion window in *Scout* was $6\times$ and $3\times$ larger than that in A-3 and A-4.

5.1.2 *Scout* or MIMO? (road segment A)

One natural question we need to address is how does *Scout* compare to a 2×2 MIMO system, considering both using *two* radio chains at the client. A commercial 2×2 MIMO system based on the 802.11n standard [27] leverages path diversity by simultaneously sending different data streams from the two radios at the transmitter (multiplexing mode) to boost throughput under good channel condition, or by duplicating a single data stream across these radios (diversity mode) to enhance reception under a harsh channel. In order to be effective, a MIMO transmitter needs to select an appropriate operation mode and modulation based on a given channel condition, which is fast varying under vehicular context and can hardly be captured with the delayed feedback. Hence, we find that the use of a scouting radio can be particularly beneficial to a MIMO system in a heterogeneous network.

Method: We compare three multi-radio systems using the cellular uplink to send feedback. (i) Single-MIMO-Radio uses one 2×2 MIMO transceiver at the client for receiving downlink traffic. (ii) Scout (2-SISO-Radios) is the *Scout* design using two SISO radios at the client, thus having a comparable number of radio chains at the client. (iii) Scout (2-MIMO-Radios) is the *Scout* version of MIMO that is enhanced by one extra scouting MIMO radio at the client. We used commercial WiFi radios operating in the 2.4GHz band for downlink communications in all the systems⁵, because the dual radio chains in WDR are not sufficiently synchronized for

⁵This is the only outdoor experiment not done in TV whitespaces.

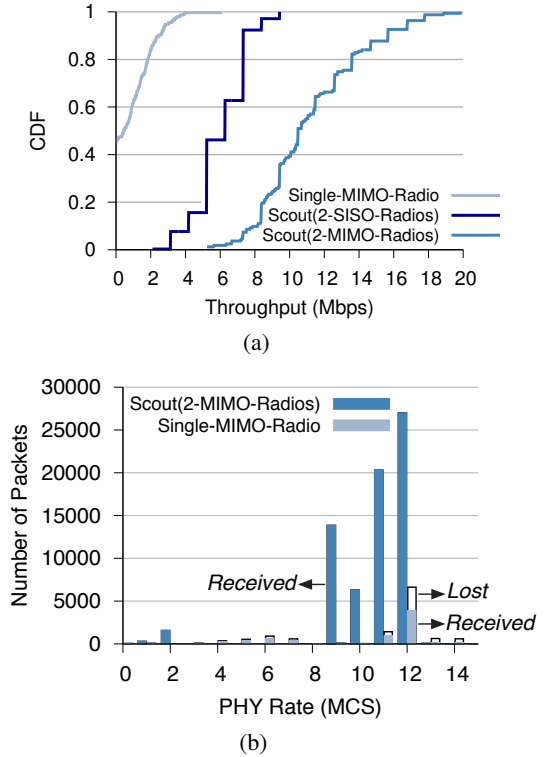


Figure 10: TCP performance of *Scout* and MIMO systems on road segment A. (a) CDF of downlink TCP throughput. (b) Distribution of end-to-end packets received and lost at different PHY rates.

MIMO signal processing. The base station was transmitting at a power of 1.4 W. We measured the downlink TCP performance of each system on road segment A, but with shorter drives due to the limited WiFi range. The experiment was conducted at late nights to avoid external interference.

Results: Figure 10(a) shows the CDF of TCP throughputs averaged over 1 second bins of each system. We observe that Single-MIMO-Radio achieved a TCP throughput of less than 2Mbps at 75th quartile due to inappropriate rate decisions caused by delayed feedback. Scout (2-SISO-Radios) had $4\times$ higher throughput compared to Single-MIMO-Radio owing to better channel estimation and aggressive error recovery. Finally, Scout (2-MIMO-Radios) can outperform Scout (2-SISO-Radios) by $1.8\times$ at the 75th quartile by fulfilling the PHY layer diversity gain in MIMO with improved protocol decisions enabled by the scouting radio.

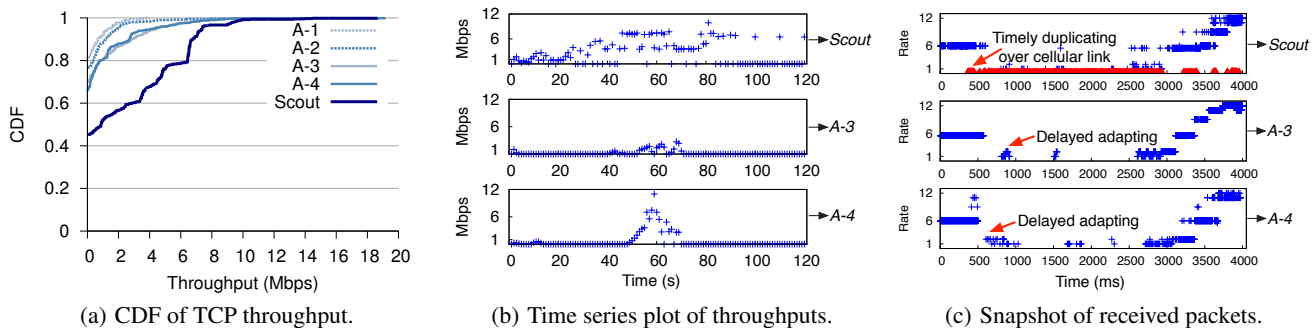


Figure 12: Downlink TCP performance of different systems along road segment B.

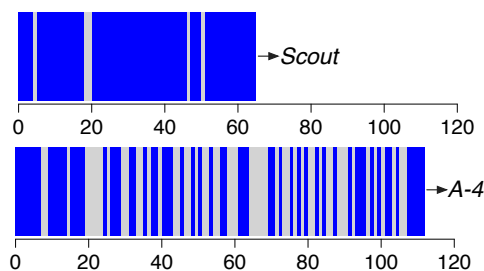


Figure 11: Time series plot of a 60s video played in different systems along road segment A. The dark color shows the playing time and the light color shows the buffering delay.

To understand the reason for the performance difference between the two MIMO systems, Figure 10(b) shows the distribution of end-to-end packets that were received and lost at each modulation and coding index (MCS). A MCS indicates a combination of the operation mode and modulation to be selected by a rate adaptation algorithm; the upper 8 indices select different modulations in the multiplexing mode, whereas the lower 8 choose those in diversity mode. We find that Single-MIMO-Radio based on delayed feedback over-selected MCS, incurring up to 42% loss rates at different MCS. This caused frequent TCP timeout and a small TCP window size (inferred by total packet counts). After enhanced by the scouting MIMO receiver, Scout (2-MIMO-Radios) has reduced loss rates to be less than 0.1%, thus being able to maintain a TCP window 5× larger than that of Single-MIMO-Radio.

5.1.3 Video streaming over TCP (road segment A)

We next evaluate the quality of video streaming in different systems. Video streaming is an application of growing popularity. A lot of emerging streaming services (Netflix, Hulu, YouTube, etc.) are implemented on top of TCP. For these applications, one performance metric is the playing outages due to buffering delays, which have a large impact on the user experience.

Method: We streamed a high definition (1280 × 720) video clip from YouTube when the client was driving on road segment A. The duration of this video was 60s. We compared the video performance between Scout and A-4. Note that A-4 performed the best among all the alternative systems.

Results: Figure 11 shows the time series plot of the video played in both systems based on our captured traces. We observe that the video played in A-4 stopped frequently for buffering, whereas the same video played in Scout only stopped for a few times (4), with

each lasting for less than 3 seconds. This leads to an overall buffering delay to be 90% lower than that of A-4, indicating better user experience in Scout.

5.1.4 TCP throughput (road segment B)

Method: We measured the downlink TCP throughput along road segment B to study the performance of different systems on a road with poor coverage from the TV whitespaces base station. Each drive lasted about 120 second, corresponding to the time for the vehicle to finish the entire route.

Results: Figure 12(a) presents the CDF of TCP throughputs averaged over 1 second bins from all the drives of each system. We observe that Scout achieved 9× gain over A-3 and A-4 at the 75th percentile. Apart from higher throughput, Scout reduced the “black-out” period (0 Mbps throughput) in both alternative systems by 25% and 20% due to the intelligent traffic duplication.

Figure 12(b) presents the throughput measurements over time in one of 5 drives for each system. We observe that Scout strived to maintain a vehicular connectivity for more than 75% of the time, whereas A-3 and A-4 were able to support connectivity for about 15% of the time. During the first 20 seconds of driving, Scout accurately detected the connection outage with the scouting radio, activating the cellular duplication to ensure a minimum connectivity. In contrast, A-3 and A-4 suffered from the link outage with zero throughput. During the period between 20 to 80 seconds, Scout detected the recovery of the whitespaces connectivity, thus stopped using the cellular link for traffic duplication. It boosted the throughput performance by selecting aggressive PHY rates while using FEC to guard against channel losses in whitespaces. Finally, all systems failed to provide connectivity at the end of the drive (approximately after 90 second) because both the cellular and whitespaces links were severely blocked by the surrounding buildings.

To illustrate how the scouting radio can improve performance, we present a 4 second packet trace for each system along the same road segment in Figure 12(c). We observe that all three systems experienced the loss of connectivity over whitespaces during a time period between 500 and 2500 milliseconds. Scout immediately detected this channel degradation by observing the packet failures at the front radio. It then enabled the base station to duplicate packets over the cellular link until the connectivity was fully recovered at 3000 milliseconds. In contrast, A-3 and A-4 incurred the delay of 225 milliseconds and 97 milliseconds in adapting to the deteriorating channel. Subsequently, both A-3 and A-4 suffered from a connection “black-out” for about 2 seconds where all packets were lost over the whitespaces. This leads to significant throughput degradation due to TCP timeout, which was avoided by Scout with traffic duplication.

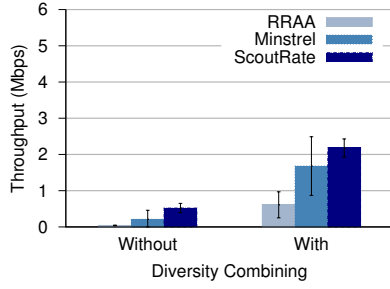
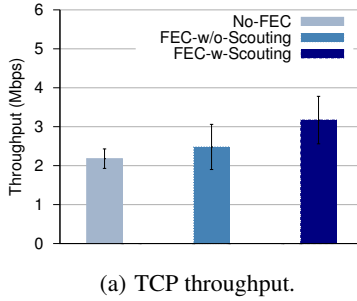
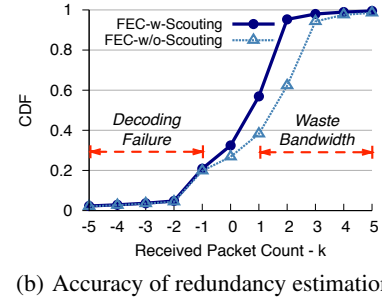


Figure 13: Performance improvement of rate adaptation (microbenchmark).



(a) TCP throughput.



(b) Accuracy of redundancy estimation.

Figure 14: Performance improvement of inter-packet FEC (microbenchmark).

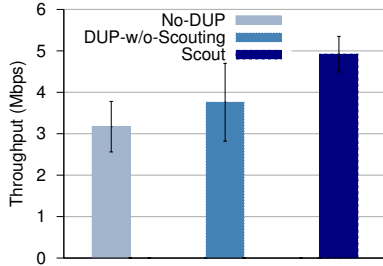


Figure 15: Performance improvement of intelligent traffic duplication over cellular path when needed (microbenchmark). No-DUP is the same as FEC-w-Scouting.

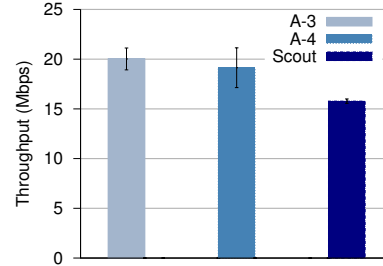


Figure 16: Performance of Scout in static scenarios.

5.2 Microbenchmarks of Scout

We benchmark the performance improvement due to individual components in *Scout*, which are rate adaptation, dual-radio diversity, inter-packet FEC, and intelligent traffic duplication. We applied these techniques incrementally to our system while carefully measuring the change in performance before and after adopting each scheme. We further quantified the benefits of the scouting radio in enhancing each transmission decision. The link-layer retransmission was enabled with a retry limit of 4 along with these techniques. All the results were downlink TCP throughputs averaged over 1 second bins from 5 drives on road segment A.

5.2.1 Rate adaptation

Method: To understand the contribution of the scouting radio to rate selection, we evaluated the rate adaptation algorithm used in *Scout* alone, denoted as ScoutRate. We compared ScoutRate against RRAA and Minstrel under the two cases where the dual-radio diversity combining was used or not.

Results: Figure 13 shows that ScoutRate has achieved 13 \times and 2.5 \times throughput improvement over RRAA and Minstrel without reception diversity, as well as 3.6 \times and 1.3 \times over the two algorithms with reception diversity. We also note that the reception diversity effectively improved the performance of RRAA, Minstrel and ScoutRate by 15 \times , 8 \times and 4 \times .

5.2.2 Inter-packet Forward Error Correction

Method: To study the efficacy of inter-packet FEC, we compared three systems, which are No-FEC, FEC-w/o-Scouting, and FEC-w-Scouting. No-FEC uses ScoutRate to select PHY rates. FEC-w/o-Scouting uses ScoutRate and FEC, but estimates redundancy based on the delayed feedback for both radios. FEC-w-Scouting uses ScoutRate and FEC based on the channel estimates provided

by the scouting radio. All the systems leverage diversity combining and have taken account its multiplicative effect in loss reduction as described in § 3.2.2.

Results: Figure 14(a) shows that FEC-w-Scouting outperformed No-FEC by 45% due to the use of inter-packet FEC. Our further analysis shows that the inter-packet FEC reduced the number of retransmissions by 62%, thus significantly reducing the bandwidth inefficiency caused by slow retransmission. While both using FEC, FEC-w-Scouting outperformed FEC-w/o-Scouting by 28% due to the improved channel estimates enabled by the scouting radio.

To understand the advantage of the scouting radio based channel estimation, we analyzed the accuracy of redundancy estimation in FEC-w-Scouting and FEC-w/o-Scouting. To this end, we calculated the difference between the number of coded packets received for each batch and the minimum number of packets needed to decode the entire batch. This minimum decodable count is equal to the number of data packets included in the batch – k as described in § 3.2.2. A positive result indicates over-estimation of redundancy, leading to wastage in bandwidth. A negative result means under-estimation of redundancy, resulting in failure of decoding. Figure 14(b) shows the CDF of this metric for all the batches. We observe that the gain of FEC-w-Scouting mainly came from fewer over-estimations of redundancy, which allowed better utilization of limited channel capacity.

5.2.3 Intelligent traffic duplication over cellular links

Method: We benchmark the efficacy of the intelligent traffic duplication in *Scout*. We compared the following three systems: (i) No-DUP that is FEC-w-Scouting without the use of duplication, (ii) DUP-w/o-Scouting that additionally performs necessary traffic duplication but based on the delayed feedback, and (iii) *Scout* – the complete version of our system. Again, all the systems use the loss rates at both radios for making transmission decisions.

Results: Figure 15 shows that the intelligent traffic duplication in *Scout* improved the overall TCP throughput by 55% over No-DUP, because whitespace links (like all other links) can experience occasional outages and the ability to divert traffic to alternate paths should always improve performance. *Scout* outperformed DUP-w/o-Scouting by 31% due to the timely duplication based on the scouting radio. Finally, since the duplication in *Scout* was done on demand, i.e., when the whitespace link is considered unresponsive, we find only 7% duplicated data to be unnecessary.

5.3 Scout in static scenarios

As discussed in § 3.1, when a client is static, our base station loses its advantage in channel estimation by reverting to the delayed feedback of the rear radio. Below we evaluate the performance of *Scout* in these static environments.

Method: We measured the downlink TCP throughput in our indoor lab environment where the loss rate was quite low. All the systems used SISO based WiFi radios for downlink communication and a cellular path for uplink. Each experiment was run for 60 seconds.

Results: Figure 16 shows the average TCP throughput of different systems. We observe that *Scout* had 21% and 17% lower throughput compared to A-3 and A-4 since the scouting radio no longer provided any useful feedback to the system. Further, some of the error recovery mechanisms such as FEC were not necessary in this setup, but introducing overheads. Hence, the scouting related functions, while particularly suited for highly mobile scenarios, should be disabled in static situations by the base station, which can quickly detect such cases using the GPS updates from the client.

6. RELATED WORK

We delineate *Scout* from prior work in the following six categories, whitespaces networking, vehicular networking, heterogeneous networking, context-based channel estimation, multi-antenna systems, and rate adaptation.

WhiteFi [1] is a system that operates over TV whitespaces using a WiFi-like protocol. A campus-wide network based on such design for vehicular connectivity has been subsequently reported in [6]. These systems use whitespaces for both uplink and downlink transmissions, thus suffering from the problem of power asymmetry avoided by *Scout*.

A large body of prior work has explored providing Internet connectivity to moving vehicles. Example systems include MAR [19], ViFi [3], CaberNet [7], MobiSteer [13], Wiffler [2], WiRover [10]. These systems use different combinations of cellular and WiFi connectivity, and do not explore the unique challenges related to the use of TV whitespaces for supporting this application.

The heterogeneous architecture has been widely adopted in satellite networks, e.g., direct broadcast satellite. These systems use the satellite link to send the downlink traffic while leveraging the dial-up modem to exchange the uplink traffic. *Scout* is different from them in two major ways. First, *Scout* is a stylized design to mitigate the effect of slow feedback in making transmission decisions over the fast whitespaces downlink. In contrast, the prior designs focus on how to efficiently utilize the slow satellite downlink with a relatively fast feedback path [14, 15]. Second, *Scout* uses link-layer only solutions to establish a robust connection, over which a variety of higher layer protocols can be efficiently supported. On the other hand, some of the solutions [14, 15] proposed in satellite networks are transport-layer centric and focus on TCP enhancements.

To address the feedback delay in a heterogeneous network, *Scout* uses the location specific channel information collected by an extra scouting radio. Predicting network connectivity based on the receiver’s location has been explored before (e.g., Bartendr [20],

BreadCrumbs [16], MobiSteer [13]). However, most of these solutions rely on some training information that is updated on a coarse timescale, e.g., order of days [20]. In contrast, *Scout* allows collecting such channel information in real-time, while obviating any training overhead.

In addition to obtaining channel estimates, the extra radio in *Scout* is also used for packet-level diversity combining. Diversity combining techniques have been studied in many multi-radio, multi-antenna systems [12, 22, 27]. These MIMO systems harness the path diversity in wireless channel for robust reception [27], scaling throughput [27], or simultaneous communications with multiple users [22]. However, most of the diversity techniques work at the physical layer of the networking stack whereas *Scout* leverages diversity at the MAC layer. Furthermore, *Scout* is shown to be complementary to these techniques by providing better channel estimates to enhance their transmission decision.

We compare *Scout* with two widely used rate adaptation algorithms, i.e., RRAA [29], Minstrel [25]. RRAA tracks packet loss rates in a short period, and uses some predetermined loss thresholds to adapt data rates. Minstrel is an enhanced version of SampleRate [4], which are used in the Ath5k driver. It periodically probes a random rate, and selects the one with the highest throughput. Recent rate adaptation techniques [17, 21, 28] attempt to collect diversified information such as PHY layer hints and device mobility, to better infer the channel condition experienced at a receiver. To be effective, all these techniques require accurate feedback, and thus can be enhanced by the scouting radio in a heterogeneous network.

Finally, the concept of a scouting radio was first reported in a position paper [30] that explored the feasibility of the idea. In this paper, we build upon this prior concept by evaluating various challenges around building a practical and working system, designing algorithmic techniques to meet such challenges, and performing detailed evaluations on the prototypes developed.

7. CONCLUSIONS

In this work, we explore a heterogeneous network design called *Scout* to provide wide-area Internet connectivity to vehicles. Our system leverages the use of new and additional spectrum available in TV whitespaces and combines it with the already pervasive cellular networks. The proposed architecture significantly enhances the coverage of vehicular whitespace deployments compared to a symmetric whitespace-only architecture.

We also introduce the notion and demonstrate the benefits of a scouting radio in the proposed architecture. In a vehicular setting, since it is natural for a rear radio to follow a front radio along a given path, the scouting radio placed at the head of a vehicle is able to provide accurate channel estimates for the main receiving radio at the rear. This accurate feedback in turn leads to better protocol decisions that ultimately translate to substantial performance improvement (3 – 8×). A preliminary version of our system has been showcased in a recent conference demo [31] and the video of this demo is at http://www.youtube.com/watch?v=_rnzH7owtBw.

Acknowledgments

We thank Ning Leng for providing various suggestions to improve the techniques developed in this work. We also thank Ren Chao and Lei Kang for their help in some of outdoor experimentation. We are grateful to our shepherd, Romit Roy Choudhury, and the anonymous reviewers whose comments helped bring the paper to its final form. All the authors are supported in part by the following awards of the US National Science Foundation: CNS-1040648, CNS-0916955, CNS-0855201, CNS-0747177, CNS-1064944, CNS-1059306, CNS-1343363, and CNS-1230751.

Algorithm 1: Scout

Input: t : Current time, λ : Antenna separation, v : Vehicle speed,
 \mathcal{D} : Batch of data packets to be transmitted, \mathcal{R} : PHY rates for selection,
 \mathcal{A}_f : Set of packets ACKed for the front radio,
 \mathcal{A}_r : Set of packets ACKed for the rear radio.

Output: \mathcal{P}_{wspace} : Coded packets sent over TV whitespaces,
 \mathcal{P}_{cell} : Data packets duplicated over cellular links.

```
 $\mathcal{L}_f, \mathcal{L}_r = \text{ScoutEstimate}(\mathcal{R}, \mathcal{A}_f, \mathcal{A}_r, t, \lambda, v)$   
 $R_{data}, R_{probe} = \text{ScoutRate}(\mathcal{R}, \mathcal{L}_f, \mathcal{L}_r)$   
 $\mathcal{P}_{wspace} = \text{ScoutFEC}(\mathcal{D}, \mathcal{L}_f, \mathcal{L}_r, R_{data}, R_{probe})$   
 $\mathcal{P}_{cell} = \text{ScoutDUP}(\mathcal{P}_{wspace}, \mathcal{L}_r)$   
return ( $\mathcal{P}_{wspace}, \mathcal{P}_{cell}$ )
```

```
1: procedure SCOUTESTIMATE( $\mathcal{R}, \mathcal{A}_f, \mathcal{A}_r, t, \lambda, v$ )  
2:    $\tau \leftarrow \lambda/v$   
3:    $\mathcal{E}_f \leftarrow \{P_i.rate : P_i \in \mathcal{A}_f, t-i \leq W\} \triangleright W$  is a time window  
4:    $\mathcal{E}_r \leftarrow \{P_i.rate : P_i \in \mathcal{A}_r, |(t-\tau)-i| \leq W/2\}$   
5:   if  $\mathcal{E}_r = \phi$  then  
6:      $\mathcal{E}_r \leftarrow \{P_i.rate : P_i \in \mathcal{A}_r, t-i \leq W\}$   
7:   end if  
8:    $\mathcal{L}_f \leftarrow \text{calculate\_loss\_rates}(\mathcal{E}_f, \mathcal{R})$   
9:    $\mathcal{L}_r \leftarrow \text{calculate\_loss\_rates}(\mathcal{E}_r, \mathcal{R})$   
10:  return ( $\mathcal{L}_f, \mathcal{L}_r$ )  
11: end procedure
```

```
12: procedure SCOUTRATE( $\mathcal{R}, \mathcal{L}_f, \mathcal{L}_r$ )  
13:  for  $R_i \in \mathcal{R}$  do  
14:    if  $\mathcal{L}_f(R_i) \leq L_{thresh}$  and  $\mathcal{L}_r(R_i) \leq L_{thresh}$  then  
15:       $throughput(R_i) \leftarrow R_i \times (1 - L_r(R_i))$   
16:    end if  
17:  end for  
18:   $R_{data} \leftarrow R_i$  with maximum throughput  
19:   $probing\_set \leftarrow \{R_i : R_i \in \mathcal{R}, R_i > throughput(R_{data})\}$   
20:   $R_{probe} \leftarrow$  a random  $R \in probing\_set$   
21:  return ( $R_{data}, R_{probe}$ )  
22: end procedure
```

```
23: procedure SCOUTFEC( $\mathcal{D}, \mathcal{L}_f, \mathcal{L}_r, R_{data}, R_{probe}$ )  
24:   $L \leftarrow \mathcal{L}_f(R_{data}) \times \mathcal{L}_r(R_{data})$   
25:   $rr \leftarrow L \times (1 + extra\_redundancy)$   
26:   $N \leftarrow coherence\_time \times R_{data}/packet\_size$   
27:   $K \leftarrow N \times (1 - rr)$   
28:   $\mathcal{P}_{wspace} \leftarrow encode(\mathcal{D}, K, N + 1)$   
29:  for  $P_i \in \mathcal{P}_{wspace}$  do  
30:    if  $i = N + 1$  then  
31:       $P_i.rate \leftarrow R_{probe}$   
32:    else  
33:       $P_i.rate \leftarrow R_{data}$   
34:    end if  
35:  end for  
36:  return  $\mathcal{P}_{wspace}$   
37: end procedure
```

```
38: procedure SCOUTDUP( $\mathcal{P}_{wspace}, \mathcal{L}_r$ )  
39:   $\mathcal{P}_{cell} \leftarrow \phi$   
40:  for  $P_i \in \mathcal{P}_{wspace}$  and  $i \leq K$  do  
41:    if  $P_i.rate = lowest$  and  $\mathcal{L}_r(P_i.rate) > L_{dup\_thresh}$  then  
42:       $\mathcal{P}_{cell} \leftarrow \mathcal{P}_{cell} \cup P_i$   
43:    end if  
44:    if  $P_i.retransmission = true$  then  
45:       $\mathcal{P}_{cell} \leftarrow \mathcal{P}_{cell} \cup P_i$   
46:    end if  
47:  end for  
48:  return  $\mathcal{P}_{cell}$   
49: end procedure
```

8. REFERENCES

- [1] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh. White space networking with WiFi like connectivity. In *SIGCOMM*, 2009.
- [2] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3G using WiFi. In *MobiSys*, 2010.
- [3] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan. Interactive WiFi connectivity for moving vehicles. In *SIGCOMM*, 2008.
- [4] J. C. Bicket. Bit-rate selection in wireless networks. In *Master's thesis, MIT*, 2005.
- [5] J. Camp and E. Knightly. Modulation rate adaptation in urban and vehicular environments: cross-layer implementation and experimental evaluation. In *MobiCom*, 2008.
- [6] R. Chandra, T. Moscibroda, P. Bahl, R. Murty, G. Nychis, and X. Wang. A campus-wide testbed over the TV white spaces. *MC2R*, 15, July 2011.
- [7] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: vehicular content delivery using WiFi. In *MobiCom*, 2008.
- [8] Federal Communications Commission. IEEE standard for local and metropolitan area networks, 2009.
- [9] Federal Communications Commission. Unlicensed operation in the TV broadcast bands, second memorandum opinion and order, 2010.
- [10] J. Hare, L. Hartung, and S. Banerjee. Beyond deployments and testbeds: experiences with public usage on vehicular WiFi hotspots. In *MobiSys*, 2012.
- [11] G. Judd, X. Wang, and P. Steenkiste. Efficient channel-aware rate adaptation in dynamic environments. In *MobiSys*, 2008.
- [12] A. Miu, H. Balakrishnan, and C. E. Koksal. Improving loss resilience with multi-radio diversity in wireless networks. In *MobiCom*, 2005.
- [13] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das. Mobisteer: using steerable beam directional antenna for vehicular network access. In *MobiSys*, 2007.
- [14] Network Working Group. Enhancing TCP over satellite channels using standard mechanisms, 1999.
- [15] Network Working Group. Ongoing TCP research related to satellites, 2000.
- [16] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom*, 2008.
- [17] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden. Improving wireless network performance using sensor hints. In *NSDI*, 2011.
- [18] L. Rizzo. Effective erasure codes for reliable computer communication protocols. *ACM CCR*, 1997.
- [19] P. Rodriguez, R. Chakravorty, J. Crowcroft, J. Chesterfield, and S. Banerjee. MAR: A commuter router infrastructure for the mobile Internet. In *Mobisys*, 2004.
- [20] A. Schulman, V. Navda, R. Ramjee, N. Spring, P. Deshpande, C. Grunewald, K. Jain, and N. Padmanabhan, Venkata. Bartendr: a practical approach to energy-aware cellular data scheduling. In *MobiCom*, 2010.
- [21] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Accurate: constellation based rate estimation in wireless networks. In *NSDI*, 2010.
- [22] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong. Argos: practical many-antenna base stations. In *Mobicom*, 2012.

- [23] C. Smith and D. Collins. *3G Wireless Networks*. McGraw-Hill, 2001.
- [24] D. Smithies and F. Fietkau. Linux wireless minstrel high throughput. <http://wireless.kernel.org>.
- [25] D. Smithies and F. Fietkau. minstrel: Madwifi and linux kernel rate selection algorithm, 2005.
- [26] Spectrum Bridge Inc. Spectrum bridge database. <http://www.spectrumbridge.com/Home.aspx>.
- [27] P. Thornycroft. Designed for speed: Network infrastructure in an 802.11n world. In *Aruba Networks white paper*, 2007.
- [28] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *SIGCOMM*, 2009.
- [29] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *MobiCom*, 2006.
- [30] T. Zhang, S. Sen, and S. Banerjee. Scout: An asymmetric vehicular network design over TV whitespaces. In *ACM HotMobile*, 2013.
- [31] T. Zhang, S. Sen, and S. Banerjee. Video streaming using whitespace spectrum for vehicular applications. In *Mobysis video session*, 2013.