

**BUILDING HIGH-PERFORMANCE WIRELESS SYSTEMS THROUGH
DYNAMIC SPECTRUM ACCESS**

by

Tan Zhang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Computer Sciences)

at the

UNIVERSITY OF WISCONSIN-MADISON

2016

Date of final oral examination: 04/26/2016

The dissertation is approved by the following members of the Final Oral
Committee:

Suman Banerjee, Professor, Computer Sciences

Aditya Akella, Associate Professor, Computer Sciences

Paul Barford, Professor, Computer Sciences

Parmesh Ramanathan, Professor, Electrical and Computer Engineering

Xinyu Zhang, Assistant Professor, Electrical and Computer Engineering

© Copyright by Tan Zhang 2016
All Rights Reserved

*To my parents, Jisheng Zhang and Rong Zheng,
my grandparents, Zhi Zheng and Jin Fang,
and my beloved wife, Ning Leng.*

ACKNOWLEDGMENTS

The seven-year academic journey at UW-Madison have left me a wonderful experience and many cherished memories about the exciting research and informative training I have gone through, the nice and quiet living at Madison, and most importantly, the outstanding people I have collaborated with.

First of all, I would like to thank my Ph.D. advisor, Suman Banerjee. He has done an amazing job in guiding my graduate studies, transforming me from a college graduate without knowing what to do, to a confident Ph.D. with solid skills at hands and a clear career goal in mind. He has always been there, pointing out exciting and ambitious research directions, giving me abundant freedom to explore different ideas and career paths, and providing insightful comments and encouraging feedback to my research. I deeply appreciate the time, patience, care, and support Suman has devoted to guiding my professional development, along with the help on many personal matters.

I also had the great honor to work with many outstanding researchers through several fantastic internships. I would especially like to thank Matt Welsh and Michael Buettner for mentoring me at Google. They have not only provided me detailed guidance and rigorous training in software engineering, but offered tremendous support and valuable suggestions for my job search. I am also fortunate enough to work with Victor Bahl during my internship at Microsoft Research. His far-reaching vision, hard-headed intelligence, and immense passion about the system work have deeply impacted my later research. I would also like to thank Aakanksha Chowdhery, Ranveer Chandra, Sharad Agarwal, and Matthai Philipose at Microsoft Research for their valuable suggestions on my research and career choices.

More luckily, I got the chance to continue my collaboration with Victor and Aakanksha through a subsequent journey to London, UK, where we worked with Kyle Jamieson and Jie Xiong from University College London. I appreciate the significant effort Kyle and Jie have dedicated to enhance the Vigil system to its final form. I also cherish that short, but extremely productive working experience, and how we spend those days and nights with cheerful discussion,

tough experiments, and brutal paper writing. I am grateful to Milind Buddikot and Peter Wolniansky for their kind guidance in my first two internships at Bell Labs, where I have gained fundamental skills for building and debugging wireless systems.

At UW-Madison, I also want to thank Aditya Akella, Paul Barford, Xinyu Zhang, and Parmesh Ramanathan for providing insightful feedback that has greatly improved this thesis. I appreciate Sayandeep Sen for his detailed mentoring in the early stages of my Ph.D. studies to gear me up for later research adventure. I also thank other senior students, Shravan Rayanchu, Jongwon Yoon, and Yadi Ma for their valuable advice in my research and career development. I am grateful to my colleagues and fellow graduate students — Ashish Patro, Lance Hartung, Michael Griepentrog, Srinivas Govindan, Neil Klingensmith, Dale Willis, Peng Liu, Greyson Hensley, Alex Sherman, Peter Den Hartog, Hongil Yoon, and Raghu Moola. They are always willing to lend a helping hand, be it for discussing ideas, offering help for deployments and experiments, giving valuable feedback, or simply hanging in there to conquer hardcore system problems with me. I also appreciate the opportunity for mentoring some excellent graduate students — Yijing Zeng, Vishal Siva Subramanian and Brett Panosh. It is really rewarding to see how they pick up their skills and embark on exciting career tracks.

Apart from my colleagues, I have spent some great time with my friends at Madison — Yizheng Chen, Yinan Li, Di Wang, Qiuling He, Mo Zhao, and Deng Liu. I especially thank my childhood friend, Yuzhou Zhang, for his long-time support and always being a trusted companion to share my joy and sorrow.

Lastly but most importantly, I would like to express my sincere and heartfelt gratitude to my family members — my parents Rong Zheng and Jisheng Zhang, my grandparents Zhi Zheng and Jin Fang, my aunt and uncle Hong Zheng and Tong Wu, and my dear wife Ning Leng. Without their boundless love, unconditional support and consistent sacrifice, I would have never reached this far. This dissertation is dedicated to everyone of them!

CONTENTS

Contents	iv
List of Tables	vi
List of Figures	vii
1	Introduction 1
1.1	<i>Focus of This Thesis</i> 4
1.2	<i>A Vehicle-based Measurement Framework to Enhance Whitespace Spectrum Databases</i> 8
1.3	<i>A Heterogeneous Network Architecture to Extend Coverage</i> 10
1.4	<i>An Edge Computing Architecture to Improve Bandwidth Efficiency</i> 12
1.5	<i>Contributions</i> 14
1.6	<i>Outline</i> 17
2	A Vehicle-based Measurement System to Enhance Whitespace Spectrum Databases 18
2.1	<i>Motivation</i> 18
2.2	<i>Limitation of existing spectrum occupancy databases</i> 20
2.3	<i>V-Scope Design</i> 25
2.4	<i>Experimental Results</i> 38
2.5	<i>Issues and Discussion</i> 46
2.6	<i>Summary of V-Scope</i> 48
3	A Heterogeneous Network Architecture to Extend Coverage 50
3.1	<i>Motivation</i> 50
3.2	<i>Advantages of a Scouting Radio</i> 53
3.3	<i>Scout: A heterogeneous design</i> 58
3.4	<i>Implementation</i> 70
3.5	<i>Experimental Results</i> 74
3.6	<i>Issues and Discussion</i> 98

3.7	<i>Summary of Scout</i>	100
4	An Edge Computing Architecture to Improve Bandwidth Efficiency	102
4.1	<i>Motivation</i>	102
4.2	<i>Vigil's Use cases</i>	104
4.3	<i>Vigil Design</i>	105
4.4	<i>Implementation</i>	118
4.5	<i>Experimental Results</i>	121
4.6	<i>Issues and Discussion</i>	131
4.7	<i>Summary of Vigil</i>	132
5	Related Work	134
5.1	<i>Whitespace Network Designs</i>	134
5.2	<i>Whitespace Determination and Management Approaches</i>	135
5.3	<i>Providing Vehicular Internet Connectivity</i>	138
5.4	<i>Centralized Wireless Architecture</i>	139
5.5	<i>Vision Analytic Systems</i>	140
6	Conclusions and Future Work	143
6.1	<i>Contributions</i>	143
6.2	<i>Future Work</i>	146
A	Impact of This Dissertation	151
B	Pseudocode of Signal Processing Functions in Scout	153
	References	155

LIST OF TABLES

2.1	Summary of datasets.	21
2.2	Two types of errors in whitespace determination.	22
2.3	Accuracy of primary detection algorithm.	29
2.4	False positive rates of the database and V-Scope models fitted for different region sizes in predicting TV broadcasts.	40
2.5	Absolute error in predicting the power of different types of signals by a 100m V-Scope model.	41
3.1	Example of link capacity estimates.	68
3.2	Summary of results in a single base station setup.	75
3.3	Alternative systems to be compared against <i>Scout</i>	76
3.4	Summary of results in a multi-base station setup.	87
3.5	PHY-layer data rates used by base stations to send downlink packets to different clients.	94
4.1	Summary of video traces used to benchmark intra-cluster algorithms	123
4.2	Average processing delay of different vision analytic functions in Vigil using a laptop with a 2.4GHz dual-core CPU.	130

LIST OF FIGURES

1.1	A typical whitespace network architecture	2
1.2	Overview of our solutions.	6
1.3	A snapshot of measurements collected from a metro bus over 120 square kilometer area.	9
1.4	V-Scope approach overview.	9
1.5	Scout design	11
1.6	Vigil design to improve the bandwidth efficiency of whitespace networks for intelligent video surveillance applications.	13
2.1	System architecture of V-Scope.	20
2.2	Wastage of whitespace spectrum by existing databases for protecting primary incumbents	22
2.3	CDF of the maximum power differences of whitespace channels at different measured locations.	24
2.4	CDF of the available whitespace spectrum at different locations based on measurements and database prediction.	24
2.5	Flow of operations in V-Scope.	25
2.6	Different spectrum captures of a -114dBm digital TV signal	27
2.7	CDF of differences between the power of a digital TV signal and that of its pilot.	28
2.8	Illustration of region models in V-Scope.	31
2.9	Accuracy of a fitted region model and a global model in predicting the power of TV signals.	32
2.10	Region models fitted by linear regression and weighted regression in predicting a TV broadcast.	33
2.11	Measured signal strength vs. propagation range of a whitespace transmitter.	34
2.12	Measurement locations for a whitespace transmitter	35
2.13	Absolute correlation of all the sectors partitioned from the centroid location with different angles.	36

2.14	Power of a TV broadcast vs. power of its adjacent-channel leakage at different locations.	38
2.15	Fraction of locations under-utilized by V-Scope and the database for predicting different TV broadcasts.	40
2.16	CDF of number of whitespace channels mis-selected at different locations under various quality constraints.	40
2.17	Fraction of locations where all the suitable channels are correctly selected by different V-Scope models.	40
2.18	Accuracy in predicting the power of TV signals.	43
2.19	Accuracy in localizing a whitespace device and microphone device.	44
2.20	Performance of different partitioning angles used by our approach for localizing the whitespace device.	45
2.21	Accuracy in predicting the power of unlicensed devices.	45
2.22	Accuracy in predicting the leakage power from TV broadcasts.	46
3.1	<i>Scout</i> design diagram	51
3.2	Illustration of channel estimation in <i>Single</i> and <i>Scout</i>	52
3.3	Outdoor deployment at the campus area of Madison, WI, USA.	54
3.4	CDF of MAC layer losses of two state-of-the-art rate adaptation algorithms with mobility and under feedback delay in the cellular uplink.	56
3.5	Average of absolute loss differences at various lags for 12Mbps packets. A lag is the elapsed time between two measurements. <i>Single</i> denotes one radio and <i>Scout</i> is the two radio setup.	58
3.6	<i>Scout</i> network architecture.	59
3.7	Operation flow of downlink communications	60
3.8	Architecture of the central controller.	67
3.9	TV whitespace radios in an OpenWrt based router.	71
3.10	Software architecture of <i>Scout</i>	72
3.11	Routes for single base station experiments.	74
3.12	Average downlink TCP performance	77
3.13	TCP performance of <i>Scout</i> and MIMO systems	78

3.14	Time series plot of a 60s video played in different systems along road segment A	80
3.15	Downlink TCP performance of different systems along road segment B.	81
3.16	Performance improvement of rate adaptation (microbenchmark).	83
3.17	Performance improvement of inter-packet FEC (microbenchmark).	84
3.18	Performance improvement of intelligent traffic duplication over cellular path when needed	85
3.19	Downlink TCP performance of Scout and alternative systems using a symmetric, whitespace-only architecture.	86
3.20	Performance of <i>Scout</i> in static scenarios.	86
3.21	Routes used in multi-base station experiments.	88
3.22	Downlink performance under the equal throughput policy with base stations operating in different channels: CDF of aggregate downlink throughput.	89
3.23	Downlink performance under the equal throughput policy with base stations operating in different channels: CDF of maximal throughput differences among contending clients.	90
3.24	Packet distribution among base stations in a single drive under the equal-throughput policy.	90
3.25	CDF of base station switching duration in a single drive under the equal-throughput policy.	91
3.26	The id of base stations serving each client defined by <i>Scout</i> , in a single drive under the equal-throughput policy.	91
3.27	Downlink throughput under two other policies with base stations operating in different channels.	92
3.28	Downlink throughput under different channel settings at base stations	93
3.29	Average downlink throughput measured by individual clients under different channel contention policies.	94
3.30	Average downlink throughput of individual clients at different scheduling round intervals in <i>Scout</i>	95
3.31	Number of possible combinations in log10 scale	97

3.32	Latency for a varying number of combinations	97
3.33	Aggregate throughput achieved by 5 base stations and a varying number of clients.	98
3.34	Aggregate throughput achieved by 15 base stations under the equal-throughput policy.	98
4.1	The two Vigil use-cases targeted in this paper.	104
4.2	Vigil architecture	106
4.3	Different views on the same scene	107
4.4	Illustration of the intra-cluster frame selection algorithm.	109
4.5	Step 2 of the Vigil intra-cluster frame selection algorithm	110
4.6	Two cameras simultaneously capture a scene containing the same two faces	112
4.7	The advanced intra-cluster frame selection algorithm.	113
4.8	Step 2 of the advanced intra-cluster algorithm	114
4.9	Illustration of the per-cluster ops queues.	116
4.10	The Vigil-DRR algorithm operating over three camera clusters with a quantum of $\frac{1}{2}$	117
4.11	Vigil deployment at three sites.	119
4.12	Example of surveillance accuracy metric.	122
4.13	Accuracy of intra-cluster frame-selection	123
4.14	Vigil 's accuracy in a single-cluster surveillance network with different number of cameras.	124
4.15	Accuracy of a multi-cluster system as the wireless capacity varies	126
4.16	Accuracy of a multi-cluster system with 10 clusters	127
4.17	Bandwidth required versus accuracy of the face detection vision algorithm	128
4.18	CDF of the distance between two people's faces and the projection errors	130

BUILDING HIGH-PERFORMANCE WIRELESS SYSTEMS THROUGH DYNAMIC SPECTRUM ACCESS

Tan Zhang

Under the supervision of Professor Suman Banerjee
At the University of Wisconsin-Madison

There is a surging demand for providing ubiquitous and high-speed Internet connectivity with the proliferation of smartphones, tablets, and other mobile devices. Nevertheless, existing wireless spectrum for mobile access has become overcrowded to meet such demand. To alleviate the spectrum crunch, regulatory agencies worldwide has started embracing a dynamic spectrum access model, which allows opportunistic communications in the unused spectral blocks. A key requirement therein is to protect existing wireless services, leading to various inefficiencies in the unlicensed, secondary applications.

To address this problem, this dissertation makes contributions in building measurement infrastructures, network architectures, and communication protocols to enhance this dynamic spectrum access model in the specific context of TV whitespaces (unused television channels).

We start by deploying a vehicle-based measurement system called V-Scope to enhance existing approaches for whitespace determination. V-Scope leverages spectrum sensors on public vehicles to collect measurements from the road. These measurements are used to refine various propagation models that can better determine whitespace spectrum, estimate its channel quality, and localize primary and secondary devices. Given the improved spectrum knowledge, we then focus on extending the coverage of whitespace networks to provide vehicular Internet connectivity. We present a heterogeneous network called Scout, which leverages TV whitespaces for downlink communications and a traditional cellular path for uplink to address the power asymmetry issue. Scout further uses an additional radio to “foresee” the channel condition to improve the transmission decisions under delayed feedback. It can also

coordinate multiple base stations to enhance the network coverage and throughput.

In the last part of the thesis, we focus on improving the bandwidth efficiency of whitespace networks. We present an edge computing system Vigil to provide intelligent video surveillance in real-time. Vigil uses TV whitespaces to connect edge computing nodes co-located with camera sensors, which can filter video traffic locally to conserve wireless bandwidth.

We believe that our measurement infrastructure, network architectures, and wireless protocols are useful in enhancing the performance of TV whitespace networks. Furthermore, most of the concepts and techniques can have broader applications to dynamic spectrum access beyond TV whitespaces.

1 INTRODUCTION

Many users have the annoying experience of their browsers getting stuck, video streaming becoming slow and choppy, and navigation services no longer accessible, especially in big cities like Chicago and New York. Most of these common frustrations come from a spotty wireless connection. Despite more and more cellular towers being deployed today, often heard from wireless providers is the concern that the surging demand for ubiquitous connectivity, intrigued by smartphones, tablets and other mobile devices, is outpacing their network capacity. A key communication resource being exhausted is the *radio frequency spectrum* over which wireless signals propagate. To address the ongoing spectrum crunch, various regulatory agencies across the world intend to open up additional spectrum bands for wireless broadband access.

Since most of the wireless spectrum has been assigned to some existing communication services, a *dynamic spectrum access* model is widely embraced by regulatory agencies around the world, which allows opportunistic communications in those spectrum blocks that are not being used by the primary applications. As the first incarnation of this concept, the unused spectrum portions in the television band (512 – 698MHz), popularly referred to as *TV whitespaces*, has been released for unlicensed usage in many parts of the world (e.g., US, UK, Singapore). This new swathe of spectrum contains substantial spectrum resource (up to 180MHz) that can be used *for free*. Further, signals in the TV band can propagate at a much longer range (up to 30km) than those in the WiFi and cellular band. Hence, as the demand for mobile and wireless connectivity continues to grow, such spectrum is going to be particularly helpful to alleviate the ongoing spectrum crunch. Moreover, the regulations and technologies developed for TV whitespaces are valuable for the release of other spectral blocks in near future, such as the 3.5 GHz band [27] and 5 GHz UNII band [26] advocated by FCC, to sustain the prosperity of mobile broadband networks.

An unique and fundamental requirement of dynamic spectrum access is to avoid interrupting those *primary spectrum incumbents* – a set of wireless devices

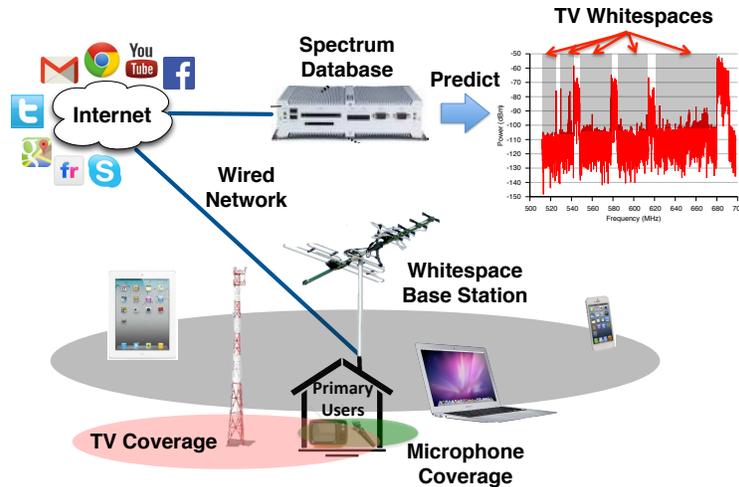


Figure 1.1: A typical whitespace network today, in which base stations query the vacant TV channels from a spectrum database on behalf of their clients. This thesis includes optimizations at the spectrum occupancy database, as well as the network architectures and communication protocols employed between the base stations and clients.

for which a spectral band is licensed. As a result, the unlicensed, secondary devices are required to only operate in the vacant channels free from primary communications. There are two types of primary devices in TV whitespaces, i.e., TV broadcast transmitters and wireless microphones. A present-day whitespace network (as depicted in Figure 1.1) relies on a few *spectrum occupancy database* to determine the unused channels as required by many regulation agencies [25, 94]. These databases are operated by some third-party companies according to some spectrum regulations. They receive queries from whitespace devices with their operating locations, and return all the vacant channels as response. Such queries are periodically sent by each whitespace device, and relayed through a subset of static base stations with wired connections to the database. Internally, each database leverages a *fixed*, widely-used propagation model [16] to predict the coverage contour of primary devices. A channel is concluded to be “whitespace” only for a secondary device operating outside the predicted contours of all the primary devices.

The fundamental requirement of protecting primary incumbents and the corresponding operation model of whitespace networks have brought various challenges as detailed below.

- **Spectrum wastage:** Existing spectrum occupancy databases are likely to have errors in predicting whitespace spectrum. Often observed are available whitespaces to be marked as occupied, leading to wasted opportunities for communications. This is because the underlying propagation model used by the databases is not able to capture the fine-grained environment variation, e.g., shadowing and multipath fading of specific contours, objects, and topologies. Since protecting primary users is at the top priority, they have to adopt a very conservative configuration for their models, which causes under-utilization of whitespace channels over a large area. Also missing from the spectrum database is the ability to validate the location of primary and secondary transmitters, which is a crucial parameter for determining the whitespace availability. Further, the databases do not attempt to distinguish the quality of vacant channels, which can differ significantly and thus have a large impact on the network performance.
- **Power asymmetry:** A mobile whitespace network today suffers from a limited communication range of its individual links due to the *asymmetric transmit power limits*. According to the ruling of many regulatory agencies [25, 94], the transmit power of the mobile whitespace devices is limited to 100 mW, whereas the power of static base stations can be up to 4 W. This 40× difference in power limit is to prevent mobile devices from causing harmful interference to the primary incumbents during roaming, as it is fairly hard for a database to determine the exact location of all the devices on the move. Hence, the stringent power limit for mobile devices seems a necessary precaution to limit their potential interference. Since most of the communication protocols need to be bi-directional, the asymmetric power limits have significantly reduced the operating range of a whitespace link to that of a “weaker” mobile client. The shorter communication range in turn necessitates much more base stations to be

deployed, causing a dramatic increase in deployment cost, along with frequent client handoff that degrades link performance.

- **Routing and handoff:** Apart from the issue of power asymmetry, the coverage of a single base station might not be sufficient for providing a satisfying wide-area coverage. In such case, a network comprising multiple base stations is needed. This can bring a variety of challenges in base station assignment, traffic scheduling, and client handoff, all of which are further complicated by the fast varying channel condition under mobility.
- **Bandwidth limitation:** While the amount of whitespace spectrum is abundant in many rural areas, it can be quite scarce in other metropolitan districts, e.g., the Times Square in which only a 5MHz TV channel is reported to be available [92]. The precious link capacity might be further cut down by various protocol overhead such as loss recovery and channel contention, especially for wide-area networks. When the link capacity is at a premium, it is far more challenging to support many popular, yet bandwidth-hungry applications like video streaming as would be done in wired networks.

These challenges motivate solutions from two broad spaces – i) building scalable measurement infrastructures for improving spectrum utilization, and ii) designing efficient network architecture and robust wireless protocols for optimizing network performance and bandwidth consumption.

1.1 FOCUS OF THIS THESIS

As discussed, the inefficiency in whitespace determination, asymmetry in transmit power limits, channel dynamics under mobility, and scarcity in link capacity, have collectively restricted the performance and applicability of present-day whitespace networks. Since these problems lie in the two constituent components of the dynamic spectrum access model – spectrum occupancy databases and wireless network designs, they naturally lead to two

orthogonal, yet complementary solution spaces that we have explored in this thesis.

To enhance the spectrum occupancy databases, we posit that the spectrum measurements can serve as an useful gauge for measuring and calibrating their underlying propagation models, as they can best reflect a given propagation environment and the wireless signal activity present therein. Nevertheless, collecting these measurements from arbitrary locations over a large area is a fairly challenging and laborious task. Hence, this dissertation explores scalable spectrum measurement infrastructures that is able to collect wide-area measurements at a relatively low cost, while efficiently utilizing this substantial measurements using statistical methods to enhance various functions of a spectrum database.

Accurately determining the available and high-quality whitespace spectrum is merely the first step toward building a high-performance dynamic spectrum access network. Challenges such as power asymmetry, link dynamics, and capacity limitation, still plague the operation of a whitespace network. We thus embark on a complementary research endeavor in this dissertation – to optimize various aspects of a wireless network to tackle the specific challenges deriving from the dynamic spectrum access.

Problem Statement and Solution Approaches

Problem statement: Given the fundamental requirement of protecting primary incumbents for a whitespace network, this thesis has attempted to address the following question —

How can we build scalable measurement infrastructures as well as robust network architectures and communication protocols, to enhance the performance of wireless networks through dynamic spectrum access?

More specifically, we explore the design and deployment of low-cost measurement systems to enhance existing spectrum databases for better predicting whitespace channels and their individual quality, while investigating stylized network architectures, wireless protocols and cross-layer techniques

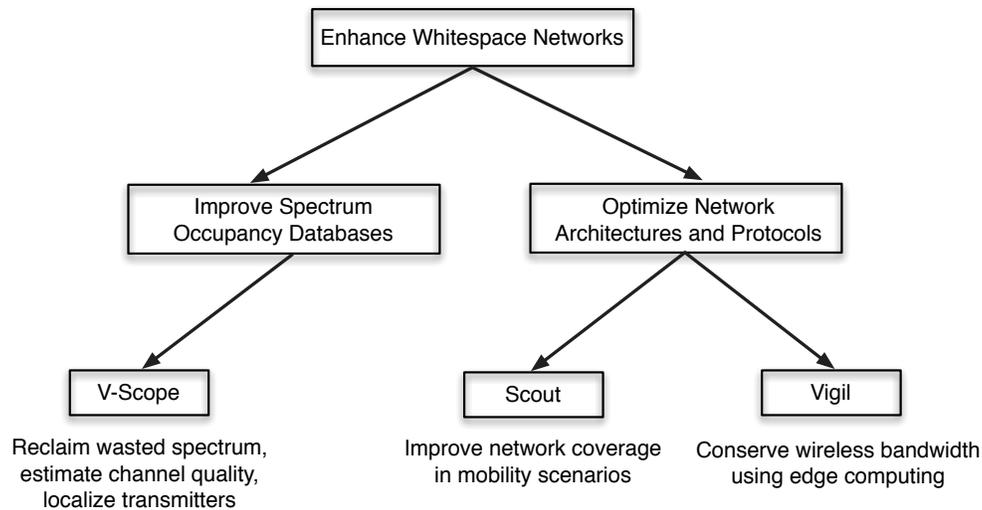


Figure 1.2: Our solutions in this thesis. *V-Scope* is implemented and deployed on public transit vehicles to quantify and enrich various management functions of existing spectrum databases, which include whitespace spectrum determination, channel quality estimation, and transmitter localization. *Scout* is a whitespace network that uses a heterogeneous architecture and a coordination framework to provide extensive coverage for vehicles and other mobile entities. *Vigil* is an intelligent video surveillance system built over TV whitespace networks, which leverages edge computing to conserve wireless bandwidth. As illustrated above, each system contributes toward enhancing whitespace networks from a different aspect.

to improve the coverage, efficiency, and robustness of a whitespace network. Figure 1.2 illustrates our approaches towards this goal, by augmenting the external spectrum databases and underlying wireless networks respectively. We next describe these solutions in more details.

Our solutions: We divide our solution space into two complementary components, each including a set of optimizations at spectrum occupancy databases or whitespace networks. To enhance spectrum databases, we build an opportunistic measurement system called V-Scope (Vehicular Spectrum Scope). V-Scope leverages spectrum sensors on public vehicles to collect wide-area measurements at a low cost. Through only a 6-week deployment on a single metro bus at Madison, WI, V-Scope is able to collect spectral data at over

one million distinct locations across a 120 sq. km. area. These measurements are used to refine various propagation models, which can enable a spectrum database to better predict the availability of whitespace spectrum, estimate the quality of individual channels (in noise power), and validate the location of primary and secondary devices.

We next switch our attention to extending network coverage in mobility scenarios, and explore a particularly challenging application of providing vehicular Internet connectivity. Our system *Scout* aims to address the dual challenges of power asymmetry and base station coordination. To deal with the power asymmetry issue, *Scout* leverages a heterogeneous architecture that uses TV whitespaces for downlink communications (from a base station to a vehicular client), while leveraging existing cellular connectivity for the uplink (from a client to the base station). The ubiquitous cellular coverage has helped circumvent the weak whitespace uplink, thereby maximizing the coverage of each whitespace base station. To compensate the high feedback delay in the cellular path, *Scout* leverages an additional radio to “foresees” the channel condition at each client, thereby improving the transmission decisions made by each base station. *Scout* can further coordinate multiple base stations through a central controller to enhance network coverage and throughput.

In the last part of the thesis, we focus on optimizing the bandwidth utilization in whitespace networks using an *edge computing* architecture. We build Vigil - a real-time, intelligent video surveillance system operating over a whitespace network. Vigil leverages edge computing units attached to distributed camera sensors to pre-process camera feed, while only selecting those relevant frames to upload to the cloud for deeper analysis. Such a *context-aware* uploading strategy can eliminate a large fraction of irrelevant video frames to conserve wireless bandwidth usage. To further improve bandwidth efficiency, Vigil leverages a cross-layer technique to *prioritize* video frames based on their utility value and the wireless capacity. It also coordinates multiple cameras monitoring a common area to boost the surveillance accuracy, while eliminating their redundant video frames to conserve bandwidth. We now describe each of these systems in more details in the following sections.

1.2 A VEHICLE-BASED MEASUREMENT FRAMEWORK TO ENHANCE WHITESPACE SPECTRUM DATABASES

We start by exploring scalable measurement infrastructures and efficient statistical approaches to augment existing approaches for whitespace determination. The present-day approach for determining the vacant whitespace channels is to have secondary devices to query a spectrum occupancy database [25, 94]. These databases are operated by some third-party companies as delegated by regulatory agencies like FCC. They use a fixed, widely-used propagation model [16] to predict the coverage contour of TV broadcasts. In addition, they establish a fixed (2km) protection contour around the location of each licensed microphone, while reserving two channels nationwide for their exclusive usage. A channel is concluded to be whitespace if a secondary device is operating outside the predicted contours of all the primary devices.

Being based solely on an empirical propagation model, such databases tend to be very conservative in protecting primary incumbents, causing unnecessary blocking of some whitespace channels over a large area (up to 71%). Further, the databases do not attempt to distinguish the quality of individual whitespace channels, which can differ significantly (40dB) at a given location, due to the *co-channel interference* from unlicensed devices (i.e., whitespace devices and unlicensed microphones), as well as *adjacent-channel* leakage from TV broadcasts. The large variation in channel quality can significantly impact the performance of a whitespace link. Also missing from the databases is the ability to validate the locations reported by primary and secondary devices. Such information can be inaccurate due to various reasons, such as incorrect entries to databases and transmitter reallocation, which causes additional errors in whitespace determination.

To quantify and address the limitations of existing databases, we build *V-Scope*, a low-cost measurement system that leverages public transit buses to carry spectrum sensors and collect measurements opportunistically as they travel. Such an approach has a unique advantage in that each mobile sensor can contribute a proportional volume of useful measurements over a relatively *large* area. Through only a 6-week deployment on a single metro bus at Madison,

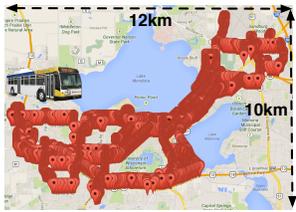


Figure 1.3: A snapshot of measurements collected from a metro bus over 120 square kilometer area. Each marker is a measured location.

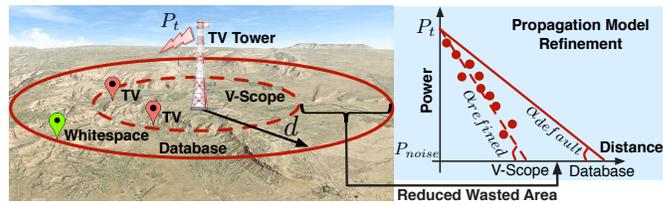


Figure 1.4: Databases use a default propagation model to predict the coverage of primary devices, blocking the use of whitespace spectrum in certain area. V-Scope reduces the wasted area by refining their propagation model with measurements collected on public vehicles.

WI, we have been able to collect measurements at more than one million distinct locations over a 120 sq. km. area (Figure 1.3). These measurements are used to augment existing databases to better predict the availability of whitespace spectrum, estimate the quality of individual vacant channels, and localize primary and secondary devices as follows.

V-Scope leverages two complementary approaches to improve the accuracy of whitespace determination. It first uses an enhanced version of feature detection technique to accurately detect primary signals based on the spectrum measurements, while being able to measure their signal strength up to the FCC-mandated sensing threshold (-114dBm [25]). It then utilizes the accurate signal strength estimates as “anchor points” to refine the parameters of any given propagation model (e.g., the slope α in Figure 1.4). The calibrated model can be used by existing databases to better predict the coverage of primary incumbents. Using a similar model fitting technique, V-Scope builds a signal strength model for each unlicensed device, while constructing a leakage model for each TV broadcast transmitter to empower the database to estimate the quality of whitespace channels. Apart from signal strength prediction, V-Scope can help pinpoint the location of any primary and secondary device. To do so, it uses a variant of the RSSI modeling localization technique [12, 75] that is tailored specifically to the outdoor scenario. Such a technique only uses a fraction of measurements that match well with an expected propagation trend

for localization, thereby reducing the impact of environment-induced variation on localization accuracy.

Based on our wide-area measurements and a combination of the proposed techniques, we find that commercial databases that are based solely on propagation models cause under-utilization of whitespace spectrum over a wide area (up to 71% measured locations for protecting TV and 70% locations for microphones). We further demonstrate that V-Scope can reclaim the spectrum wastage for protecting primary incumbents at up to 59% locations. Further, V-Scope can help identify all the suitable whitespace channels at 72 – 83% locations given different channel quality constraints. It can also pinpoint TV-band devices at various locations at a low error of 16 – 27m, outperforming the state-of-the-art localization techniques by 1.2 – 3.5 \times .

1.3 A HETEROGENEOUS NETWORK ARCHITECTURE TO EXTEND COVERAGE

In the next piece of work, we explore a heterogeneous network architecture to address the power asymmetry issue in whitespaces. We focus on a particularly challenging application of providing vehicular Internet access, which has become increasingly popular for diverse applications, e.g., improved traffic intelligence, transportation safety, and infotainment for passengers. As our long-term deployment goal, we intend to bring “on-board” Internet access to the commuters of the Madison Metro Transit operating hundreds of buses. To this end, we have mounted a few base stations along the roadside. Each base station uses a whitespace radio to communicate with a mobile gateway (client) mounted on buses. The gateway provides a WiFi hotspot inside the bus, allowing users to connect with their WiFi-capable devices.

A key challenge in this setup is the asymmetric power limits that significantly limits the coverage of each base station. Specifically, the mobile whitespace devices (gateway nodes) are required to use a much lower transmit power (up to 100 mW) than that of static base stations (4 W) [25, 94]. This 40 \times difference in the transmit power is to restrict the roaming-induced interference of mobile devices. Since most of the communication protocols need to be bi-directional, the asymmetric power limits have significantly limited the coverage radius

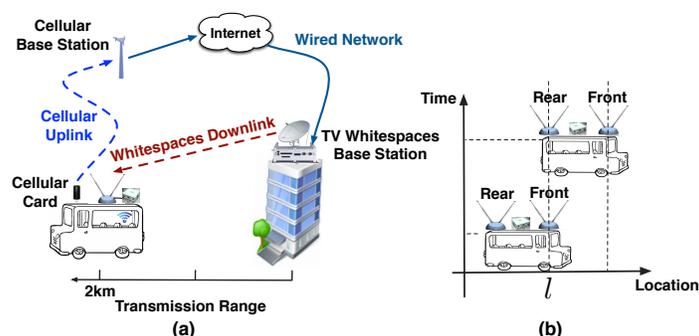


Figure 1.5: Scout design to improve base station coverage: (a) heterogeneous architecture; (b) illustration of channel “scouting”.

of a base station to that of a “weaker” mobile client (e.g., from 2km to 500m). To produce a similar coverage, much more base stations would be deployed, significantly increasing the infrastructure and management cost.

To circumvent the bottleneck of “weak” whitespace uplink, we build a heterogeneous network called *Scout*. *Scout* sends the downlink traffic primarily over the whitespace path, whereas communicating the uplink traffic over existing cellular paths (depicted in Figure 1.5(a)). The extensive cellular connectivity allow *Scout* to maximize the downlink coverage of each whitespace base station. Further, *Scout* is efficient in utilizing TV whitespaces for relaying network traffic, as many networking applications are downlink dominated ($10\times$ in WiRover [36]). While effectively extending the coverage, the *high latency* in the cellular uplink significantly degrades the performance of a heterogeneous network. Since most of communication systems rely on channel estimation for making *various* protocol decisions, e.g., rate adaptation, the stale feedback leads to inaccurate channel estimation, especially in the mobile environment, causing poor transmission decisions at the base stations.

To improve the channel estimation with a slow feedback path, we leverage an extra “scouting” radio to measure the channel condition at each reception location in advance (depicted in Figure 1.5(b)). Our core intuition is that the reception location of a radio largely determines its experienced channel condition [61, 63, 83]. We exploit this location-specific channel characteristics

by mounting the scouting radio at the head of the vehicle, while placing the receiving radio at the rear. The head radio can “scout” the likely channel condition at each future location where the rear radio will visit shortly. By the time the rear radio reaches the same location l , the base station can use this more relevant channel estimate previously reported by the front radio at l to improve transmission decisions. Built on this scouting-based channel estimation technique, we have developed multiple aggressive transmission techniques to improve individual link robustness, while coordinating multiple base stations to enhance network coverage and throughput.

1.4 AN EDGE COMPUTING ARCHITECTURE TO IMPROVE BANDWIDTH EFFICIENCY

We now focus on enhancing an orthogonal aspect of whitespace network – wireless bandwidth utilization. We explore the solutions under a bandwidth-hungry application of providing intelligent video surveillance. Automated surveillance systems continue to pervade our physical existence with many valuable applications. Examples include business analytics at retail stores, security monitoring in corporate and educational campuses, and traffic control in smart cities. Nevertheless, most of these camera surveillance systems today such as DropCam [21] rely on a wired network infrastructure to upload camera feeds to the cloud for offline analysis. This significantly limits the coverage of surveillance applications, while increasing the deployment cost.

To remove the constraints of wired connections, we explore a wireless video surveillance network that leverages a TV whitespace network to connect distributed camera sensors to the cloud. The key challenge in such a system is the limited capacity of the underlying whitespace network, which can be far from sufficient to accommodate simultaneously uploading of multiple high-definition video feeds (22 Mbps for each 1080p video). When the wireless capacity lags behind the bandwidth demand of video traffic, large video backlog and even dropping of video frames can occur, significantly degrading the accuracy and responsiveness of a surveillance application. While prior work in video compression [1, 2] and upload shaping techniques [17, 37] have explored reducing the bandwidth consumption of video traffic, most of these techniques

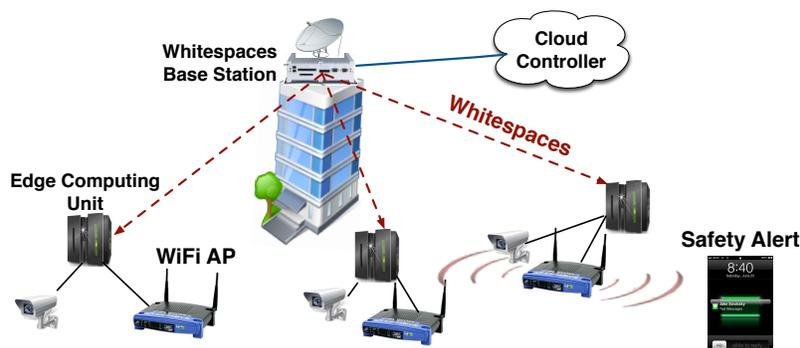


Figure 1.6: Vigil design to improve the bandwidth efficiency of whitespace networks for intelligent video surveillance applications.

are agnostic to the usefulness of video frames pertaining to the end surveillance application, hence leading to significant bandwidth wastage of uploading irrelevant data.

To address this bandwidth inefficiency, we build Vigil – a real-time wireless video surveillance system that uses an edge computing framework to filter surveillance traffic at the network edge (depicted in Figure 1.6). These edge computing units (ECNs) are collocated with camera nodes to pre-process video feed based on a specific user query, such as locating people or tracking objects with certain features. Each ECN node uses lightweight vision algorithms (e.g., motion and object detection) as specified in the query to process video feed, while generating a stream of analytic results for individual video frames. These analytics include the frame significance and object features, which are uploaded over TV whitespaces to a cloud based controller. Based on the vision analytics, the controller subsequently pulls those significant frames from the corresponding ECNs for deeper analysis, e.g., people recognition or trajectory synthesis. This context-aware uploading strategy can eliminate a large fraction of unrelated video frames from uploading to the cloud, thus conserving the bandwidth usage of the underlying whitespace network. The saved capacity is scavenged by each ECN to backhaul users' WiFi traffic from its attached access point, thereby offsetting the deployment cost.

We develop a cross-layer frame selection algorithm at the controller to further improve the bandwidth utilization of uploading video surveillance

traffic. Such an algorithm prioritizes video frames based on their significance and the underlying capacity of wireless links. To quantify a frame's significance, we design the *ops* metric (objects per second), which captures both the amount of useful information contained in a video frame (in number of objects in interest), and the bandwidth cost of uploading this frame to the cloud. Based on this metric, our algorithm uploads video frames in the descending order of their significance under the estimated bandwidth constraint. This greedy algorithm is shown to maximize the number of query-specific objects delivered to the cloud, while minimizing their bandwidth consumption.

To boost the accuracy of vision analytic functions, Vigil can leverage multiple cameras to monitor a common area from different locations. Such an approach can deal with the inherent inaccuracy of vision algorithms that are sensitive to various environmental factors such as lighting and capture angle. It can further detect and eliminate redundant frames captured by multiple cameras, based on a light-weight image stitching technique running at the controller.

We have deployed Vigil at three sites across two countries, and evaluated its efficacy in surveillance applications under vastly different network conditions. Given a wide range of activity levels in different monitored scenarios, Vigil is able to extend the video surveillance coverage by 5 – 200 times over a simpler approach that streams video feeds directly over the underlying whitespace network.

1.5 CONTRIBUTIONS

While dynamic spectrum access is a promising solution to address the continued spectrum crunch, building a mobile broadband network based on this opportunistic access model is particularly challenging. The hardness stems from the fundamental requirement of protecting primary incumbents, and is further exacerbated by the link dynamics caused by device mobility and channel contention. This leads to a range of challenges in whitespace determination and management, network coverage, and bandwidth utilization of present-day whitespace networks. To tackle these problems, this dissertation has explored two complementary solution spaces to optimize the constituent components of

dynamic spectrum access – the spectrum management module and wireless networks. These efforts involve developing low-cost measurement systems and model refinement techniques to augment existing spectrum occupancy databases, while building stylized network architectures, robust wireless protocols, and intelligent scheduling techniques to improve the coverage and bandwidth efficiency of whitespace networks. Specifically, the contributions of this dissertation are described as follows:

1. We designed and implemented V-Scope, an opportunistic measurement system that leverages spectrum sensors on public vehicles to collect wide-area measurements at low-cost. The measurements are used to quantify the performance of existing spectrum databases, while enhancing them to better predict whitespace spectrum, estimate the channel quality, and localize primary and secondary devices. While a few prior studies [56, 105] have reported the inefficiency of databases for protecting primary incumbents, they are based on measurements manually collected at a very few locations. In contrast, V-Scope has enabled the first of its kind study over wide-area, by leveraging a metro bus to collect data at about 1 million distinct locations over a 120 square-km area at Madison, WI. Using these measurements and an enhanced signal detection algorithm, we show that commercial databases cause under-utilization of several whitespace channels over a wide area (up to 71% measured locations). We also report that the quality of whitespace channels can differ significantly at a given location (e.g., up to 40dB in noise power), due to various primary and secondary interference. Motivated by these observations, we developed model fitting techniques in V-Scope, which can use measurements as “anchor points” to refine various propagation models for augmenting databases. We show that the calibrated models are able to reclaim the wasted area in whitespace determination by up to 59%. They can further enable the databases to identify suitable whitespace channels at 72 – 83% locations given different quality constraints. In addition, these models can be extended to localize TV-band devices in outdoor scenarios with a relatively low error (16 – 27m).

2. We designed and implemented *Scout*, a heterogeneous network that tackles the specific problem of power asymmetry in whitespaces for providing robust Internet connectivity to vehicles. *Scout* leverages a heterogeneous network architecture where the downlink communications is primarily conducted over TV whitespaces, whereas the uplink data is sent over the cellular path. The extensive cellular coverage enables *Scout* to circumvent the bottleneck of weak whitespace uplink, thereby maximizing the coverage of each whitespace base station. To address the problem of feedback delay over the cellular uplink, we introduce the notion and demonstrate the benefits of a scouting radio for channel estimation. 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. We have built a functional system around this scouting based channel estimation to adapt aggressive transmission techniques, i.e., rate adaptation, inter-packet FEC, and intelligent traffic duplication over cellular paths, for enhancing the downlink performance of vehicular connectivity (by 3 – 8× in TCP throughput). We further built a centralized controller to coordinate multiple whitespace base stations to enhance the aggregate network capacity (at 73 – 374% gain), while forcing flexible channel contention policies.
3. We designed and implemented *Vigil*, a real-time intelligent video surveillance system over TV whitespaces. *Vigil* leverages a TV whitespace network to connect distributed camera sensors wirelessly to a cloud-based controller to support pervasive video surveillance functions. To deal with the bandwidth limitation of whitespace links, *Vigil* intelligently partitions video processing between ECNs co-located with camera nodes and the cloud. Each ECN processes the camera feed locally with some light-weight vision analytic algorithms, while reporting the results to the controller. These analytics enables the controller to only request the *relevant* video frames from the ECNs to conduct sophisticated analysis for conserving wireless bandwidth. The saved wireless capacity is further scavenged by

each ECN node to provide WiFi access to recoup the operation cost. To further improve bandwidth efficiency, Vigil leverages a frame scheduling algorithm to prioritize video frames based on their utility and the available wireless capacity. It further leverages multiple cameras to monitor an overlapping area from different angles and locations to improve accuracy. Such a technique can also identify frames containing duplicate objects captured by these cameras based on their vision analytics, while only uploading a best copy to minimize bandwidth usage. Through three deployments across two countries, we demonstrate significant benefits of Vigil for improving the bandwidth efficiency of whitespace networks, by supporting a 5 – 200× larger surveillance area compared to a traditional approach streaming the entire video to the cloud.

1.6 OUTLINE

The rest of the thesis is organized as follows. In Chapter 2, we present our measurement infrastructure V-Scope, which leverages public vehicles for collecting wide-area measurements to augment spectrum occupancy databases. In Chapter 3, we focus on enhancing the coverage of whitespace networks under mobility scenarios. We present *Scout*, a TV whitespace network based on a heterogeneous architecture and a centralized coordination framework to provide robust Internet connectivity to vehicles. In Chapter 4, we aim to enhance the bandwidth efficiency of whitespace networks. We present Vigil, a real-time wireless video surveillance network that leverages edge computing to conserve wireless bandwidth. In Chapter 5, we compare our work with prior approaches and systems to manage and enhance wireless networks through dynamic spectrum access. We conclude and discuss the avenues for further research in Chapter 6.

2 A VEHICLE-BASED MEASUREMENT SYSTEM TO ENHANCE WHITESPACE SPECTRUM DATABASES

2.1 MOTIVATION

In this chapter, we describe our work to address the first and foremost issue in dynamic spectrum access — to accurately determine vacant and high-quality channels. We leave how to optimize wireless networks to utilize such spectrum in following chapters. The current mechanism for determining vacant TV whitespaces is to have secondary devices to query *spectrum occupancy databases* [25, 94]. These databases are operated by some third-party companies following the spectrum regulation. They rely on a *same*, widely-used propagation model (R6602 [16]) to predict the coverage contour of TV broadcasts. In addition, they set up a fixed (2km) protection contour around licensed microphones and reserve two channels nationwide for their exclusive usage. A channel is concluded to be whitespace if a secondary device is outside the predicted contours of all the primary devices.

Being based solely on a propagation model with conservative configurations, such databases are likely to have errors in predicting the available whitespace spectrum, causing under-utilization of some whitespace channels over large area (up to 71%). Further, the databases do not attempt to distinguish the quality of whitespace channels, nor are they responsible for validating the locations of primary and secondary transmitters. To address the limitations of existing databases, we explore the use of spectrum measurements combined with propagation models to better determine whitespace spectrum, estimate the quality of whitespace channels, and localize primary and secondary devices. In particular, we present *V-Scope* (Vehicular Spectrum Scope), a measurement system that leverages public transit buses to carry spectrum sensors and collect wide-area measurements opportunistically as they travel. These measurements are used to enhance various functions of databases as follows.

V-Scope uses two related techniques based on measurements to enhance databases for whitespace determination. It first uses an enhanced version of

feature detection that can detect primary signals and measure their power at up to the FCC-mandated detection threshold (-114dBm [25]). Based on the measured signal strength of primary devices, V-Scope refines the parameters of a propagation model, which can be used by databases to better predict the coverage of primary devices.

Apart from whitespace determination, V-Scope enables databases to estimate the quality of whitespace channels. Such a function is particularly valuable as the noise power in whitespace channels can differ significantly (up to 40dB) based on our measurements. Such variation in channel quality is contributed by the *co-channel interference* from unlicensed devices (whitespace devices and unlicensed microphones), and *adjacent-channel* leakage from TV broadcasts. Unfortunately, both types of interference cannot be captured by existing databases without knowing their transmission characteristics (i.e., location and transmit power). To predict co-channel interference, V-Scope uses a similar model fitting procedure mentioned above to construct signal strength models for each unlicensed device. For adjacent-channel leakage, V-Scope constructs a leakage model that can take the strength of a TV broadcast signal to accurately estimate its leakage power at a given location.

V-Scope can also empower databases to localize primary and secondary devices. Such location information is needed to predict the availability and quality of whitespace channels. But it is currently missing from existing databases for secondary devices, and can be inaccurate for primary transmitters. V-Scope utilizes the measured signal strength of these devices to pinpoint their location, thereby enabling database operators to obtain and validate such information. To achieve this, we have leveraged a RSSI modeling technique that is commonly used in indoor localization systems [12, 75], while enhancing it to account for environmental variation in outdoor scenarios.

While our opportunistic wardriving approach has some unique advantages in collecting wide area measurements, a similar goal can certainly be achieved by other approaches such as local spectrum sensing and crowd-sourcing mobile whitespace devices. Regardless of how measurements are collected, most of our proposed techniques remain useful for enhancing databases based on these measurements.

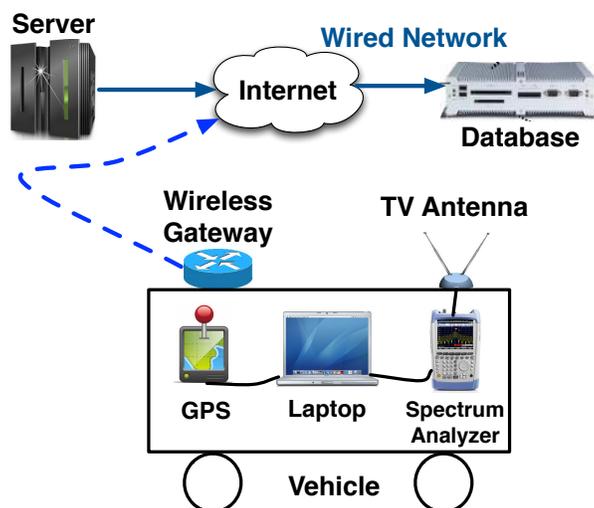


Figure 2.1: System architecture of V-Scope.

2.2 LIMITATION OF EXISTING SPECTRUM OCCUPANCY DATABASES

We now motivate the design of V-Scope by demonstrating the two major limitations in existing spectrum databases. First, to tolerate the inaccuracy of propagation models, the databases assign very conservative protection contours for primary devices, which cause non-negligible wastage in whitespace utilization. Second, they are agnostic to the quality of whitespace channels, which can differ significantly at a given location, and thus can have a large impact on the performance of a whitespace network. We start by describing our measurement setup and datasets.

Measurement setup

Our current deployment consists of a server and a client as shown in Figure 2.1. The client has been deployed on a metro bus traveling in and around a mid-sized US city for a 6-week period. It uses a laptop to configure a portable spectrum analyzer (WSA4000 [96]) for collecting spectrum samples (FFTs) in all the UHF channels. An omni-directional TV antenna with 0dB gain is used to capture signals from all the directions. Based on these spectrum samples of a channel,

Dataset	Measured Locations	Coverage (sq. km.)	Channels	Vehicle	Target Device
A	1 million	120	30 UHF channels	A metro bus	All TV-band devices
B	3.5K	3	Channel 42	A personal vehicle	A 3.8W whitespace transmitter atop a 8-floor building
C	2.5K	2	Channel 28	A personal vehicle	A 100mW microphone in 5 buildings

Table 2.1: Summary of datasets.

the laptop performs some real-time analysis (§ 2.3), i.e., primary detection and power estimation, while obtaining the measured location from a GPS module. It uploads the GPS reading and measurement results to our server over cellular networks. The server is situated in our laboratory, with an Ethernet to receive the measurement results. It queries a commercial database [92] at measured locations to evaluate the database’s accuracy.

Datasets: Using this measurement setup, we have collected three datasets on different vehicles. Table 2.1 summarizes their salient features. **Dataset A** is our main dataset collected from the metro bus, and covers about one million distinct locations over a 120 sq. km. area. The data consists of signal type, power and the database’s prediction for all (30) UHF channels at each measured location. We will use this dataset to evaluate the performance of the database and V-Scope as well as studying whitespace channel quality. **Dataset B** contains measurements in a whitespace channel where our whitespace radio was transmitting 12Mbps UDP packets using the 802.11 baseband technology. This whitespace device was mounted atop a 8-floor building with a transmit power of 3.8W. **Dataset C** contains measurements in a whitespace channel for a microphone device transmitting at 100mW in 5 different buildings at a downtown area. Dataset B and C were collected from a personal vehicle driving along multiple road stretches, and comprise the signal type and channel power at each measured location. We will use them to benchmark our measurement clustering algorithm (§ 2.3) and localization technique (§ 2.3).

	Ground truth	Prediction
False Positive	Occupied	Vacant
False Negative	Vacant	Occupied

Table 2.2: Two types of errors in whitespace determination.

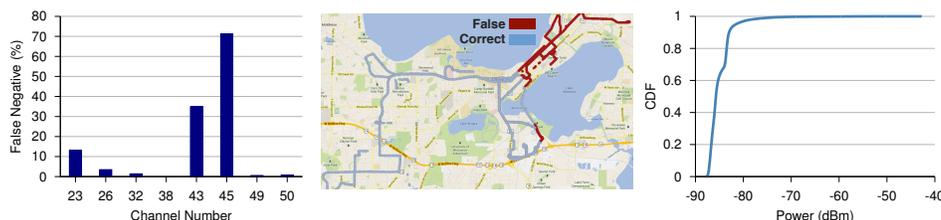


Figure 2.2: Wastage of whitespace spectrum by existing databases for protecting primary incumbents. (a) False negative rates in predicting TV broadcasts. (b) Spatial distribution of false negatives and true negatives in channel 43. (c) CDF of power measurements collected in reserved channels for licensed microphones.

Inefficiency in protecting primary users

We study the performance of a commercial database (SpectrumBridge [92]) in predicting TV and licensed microphones respectively. Note that other FCC-approved databases would produce the same prediction results. Our focus is to understand its performance tradeoff between the safety in primary protection and efficiency in whitespace utilization. All the evaluation in this section is based on dataset A unless otherwise mentioned.

Performance in protecting TV broadcasts: Borrowing the definition from prior work [56], we divide the errors in whitespace determination into two types, i.e., *false positive* and *false negative* as summarized in Table 2.2. A false positive is a location where the database mis-predicts an occupied channel (with measured power ≥ -114 dBm) as whitespaces, whereas the opposite being a false negative. We find negligible false positives ($< 0.29\%$) of database in predicting all the TV broadcasts, which is similar to a prior report [56]. Thus, existing databases can

reliably protect TV broadcasts. However, the first panel of Figure 2.2 shows a 13–71% false negative rate in half of these channels, indicating these channels being unnecessarily blocked for unlicensed usage over a wide area. The second panel of Figure 2.2 shows a typical spatial distribution of false negatives in one such channel, along with those locations with this channel correctly predicted to be whitespace (true negatives). We note most of these false negatives are at the north-east side, which is at the *closer* side to the TV tower. Thus, we believe these false negatives are caused by over-provisioning the protection contour of TV broadcasts.

Performance in protecting licensed microphones: To quantify false negatives in protecting licensed microphones, we calculate the power of about 200k measurements collected in their reserved channels. The last panel of Figure 2.2 shows that about 70% measurements have very low power, suggesting that microphone signals are unlikely to be present in many reserved locations. While these wardriving measurements might not capture every instance of microphone transmissions, they are statistically sufficient to suggest that licensed microphones are not making efficient use of their reserved channel, possibly due to coarse-grained reservation periods (24 hours) and over-sized protection contours (2km). On the other hand, we find no other type of signals in these channels, suggesting the database to be sufficient in protecting licensed microphones.

Variation in whitespace channel quality

We now study the variation in whitespace channel quality, which are caused by *co-channel interference* from secondary devices and *adjacent-channel leakage* from TV broadcasts.

Magnitude of quality variation: Figure 2.3 shows the CDF of absolute differences in noise power between the best channel and the worst channel at each measured location. *Overall*, *TV-adjacent*, and *Non-adjacent* choose the worst channel from all the whitespace channels, adjacent channels to TV broadcasts, and non-adjacent channels, respectively. All of them pick the best channel from all the whitespace channels. We observe that the worst channel in *Overall* has a

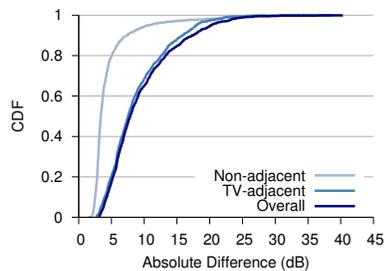


Figure 2.3: CDF of the maximum power differences of whitespace channels at different measured locations.

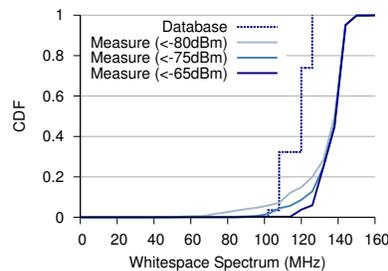


Figure 2.4: CDF of the available whitespace spectrum at different locations based on measurements and database prediction.

8dB higher power in median and 17dB higher power in 90-quartile compared to the best channel. Such variation is currently contributed by unlicensed microphone transmission and TV broadcast leakage. We also note the non-adjacent channels generally have better quality than adjacent channels due to the absence of broadcast leakage. However, its worst channel still has 8dB higher power than the best channel at 10% locations due to unlicensed microphones. We envision this quality variation to become more prominent with the future proliferation of whitespace devices.

Prevalence of quality variation: Figure 2.4 shows the CDF of the amount of whitespace spectrum below different channel power based on our measurements and the database. We observe that for about 80% locations, the amount of whitespace spectrum identified by measurements is similar across different power thresholds, which is about 18 – 24MHz more than that predicted by the database. For the rest 20% locations, this measured amount differs by 12 – 48MHz (2 – 8 channels) between the highest and the lowest power threshold (-65dBm and -80dBm). Thus, whitespace devices are likely to operate in a low-quality channel at these locations, if they randomly pick a channel suggested by the database.

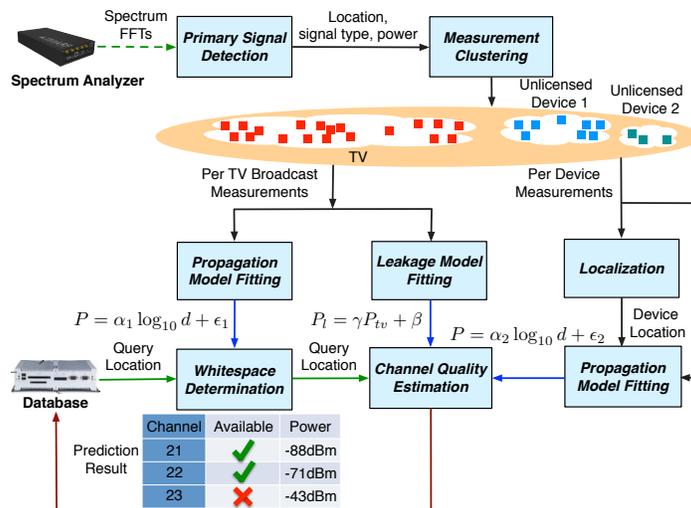


Figure 2.5: Flow of operations in V-Scope.

2.3 V-SCOPE DESIGN

V-Scope is based on our measurement architecture described in § 2.2, and consists of following components to augment databases with measurements — i) primary detection, ii) measurement clustering, iii) propagation model refinement, iv) localization, and v) leakage model construction. Figure 2.5 shows its operation flow.

Overview: Our proposed system leverages a few clients mounted on public vehicles to detect primary and secondary devices based on spectrum samples collected in different UHF channels (§ 2.3). The detection results that include the device type and power of each channel are forwarded to our server. The server groups these measurements according to different device types in each UHF channel, and segregates each group for individual devices (§ 2.3). It then localizes those (secondary) devices if their location is not available in databases (§ 2.3). Using each device’s location and its associated measurements, the server constructs a propagation model that is tailored to local environment to predict its signal strength (§ 2.3). It also builds a model to predict the power of adjacent-channel leakage from TV broadcasts (§ 2.3). To predict whitespace

channels in the vicinity of measurements, the databases can use our refined propagation models to better estimate the coverage of TV broadcasts. Similarly, the databases may use our models constructed for secondary users to predict their in-band interference, while leveraging the leakage model to estimate the adjacent-channel leakage from TV broadcasts. The sum of the predicted power of these interference signals is the quality of a whitespace channel. To predict for those locations far from measurements, the databases may switch back to use its default model (R6602). Finally, we envision the databases to use its current mechanism for protecting licensed microphones due to the limitations of our wardriving approach as will be discussed in § 2.3. We now explain each of these components in detail.

Zoom-in pilot tracking algorithm for primary detection

Our measurement module aims to utilize spectrum samples for detecting different types of primary transmitters and unlicensed devices (i.e., whitespace devices and unlicensed microphones). We accomplish this task by leveraging feature detection algorithms [45, 59] that identify different signals based on their spectral features. While such a technique performs well for detecting unlicensed signals, we find non-trivial challenges in satisfying the FCC's requirement for detecting primary signals at up to -114dBm in our measurement setup. Such a stringent detection threshold is to tolerate the sensing inaccuracy caused by fading and shadowing, and to take account the transmission range of a whitespace device. We use TV detection to illustrate this challenge and our specific solution.

Existing algorithms detect a TV signal by tracking its pilot in the spectral domain. A pilot is a set of preambles in a TV packet, which produces a predominant peak at a fixed frequency. Unfortunately, for a TV signal close to the detection threshold, even this peak can be overwhelmed by noise and thus unable to be captured. To illustrate this, we attenuated a TV signal to be about -114dBm , while collecting its spectrum over a 6MHz TV channel with the largest number of FFTs (32768) available in our high-end spectrum analyzer [96]. As shown in the first panel of Figure 2.6, the pilot of this TV spectrum is hard to

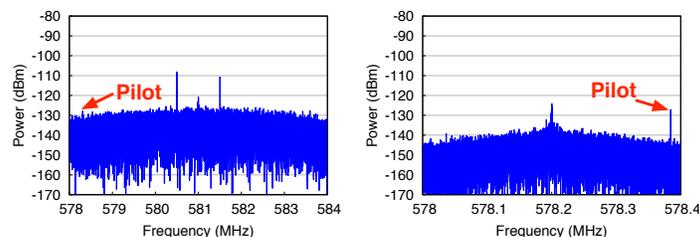


Figure 2.6: Different spectrum captures of a -114dBm digital TV signal. (a) Full-channel capture; (b) Zoom-in capture at the first 488KHz band. Both captures consist of 32768 FFTs.

be distinguished from the noise floor. This is because the noise spectral density is -159dBm/Hz in our spectrum analyzer. When using 32768 FFTs to represent a 6MHz band, the bandwidth per FFT bin is 183Hz ($6\text{MHz}/32768$) and thus the noise power per bin is about -135dBm. As will be discussed later, the pilot power is usually about 10 - 15dB lower than the total power of a TV signal, and thus can be as low as -129dBm. Comparing to the -135dBm noise floor, the 6dB stronger pilot is not robust enough for tolerating the signal strength fluctuation due to fading and shadowing. Prior systems [59] have overcome this challenge by leveraging low-noise amplifiers to amplify a weak signal by a fixed amount (e.g., 20dB). But such an approach is observed to cause *saturation* of a spectrum analyzer when capturing strong signals ($>-35\text{dBm}$), leading to distorted spectrum that can significantly degrade detection performance. The alternative solution of using tunable amplifiers would largely increase design complexity and hardware cost.

Zoom-in pilot tracking algorithm: Instead of resorting to signal amplification, V-Scope improves the sensitivity of feature detection by capturing the most prominent portion of spectrum with the available FFTs. To detect a TV signal, for example, we configure a spectrum analyzer to capture at a narrow band (488KHz) around its pilot frequency. This can effectively improve spectral resolution while reducing the noise floor, thereby producing a clear pilot peak as shown in the second panel of Figure 2.6. Since this peak is well distinguishable at the detection threshold after zoom-in, V-Scope uses it as an unique feature to detect TV signals.

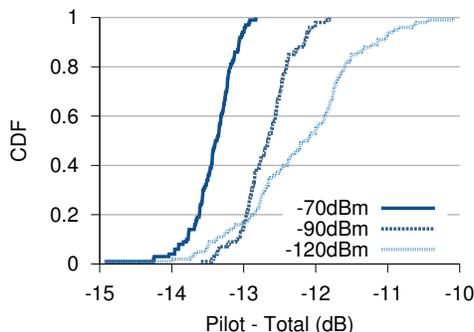


Figure 2.7: CDF of differences between the power of a digital TV signal and that of its pilot.

The detected pilot is, however, insufficient to determine whether the power of a TV signal to be above or below the -114dBm threshold. We also need the precise power information of a primary signal for constructing its signal strength models (§ 2.3). To achieve this goal, V-Scope leverages the power of a TV pilot to estimate its total power. According to the digital and analog TV standards [15], there is a fixed power offset between a TV signal and its pilot. For example, the pilot of a digital TV is required to be 11.3dB lower than its total power. Figure 2.7 shows this relationship indeed holds for a DTV signal at a wide range of power, albeit with some variation ($10 - 15\text{dB}$). Thus, V-Scope computes the total power of a TV signal by adding to its pilot power a constant offset η (20dB in our implementation). The estimated power is then compared with the -114dBm detection threshold to determine TV whitespaces.

Our proposed technique can be applied to microphone detection in a similar way. Briefly, we start by examining the 6MHz wide spectrum of each TV channel for detecting potential microphone tones that are narrow spikes carrying audio signals. Since these spikes can also be a result of noise fluctuation, we perform a narrow-band capture around each spike, extracting various features from its zoom-in spectrum for further validation. Once a microphone signal is confirmed, we use the power of its tones in the zoom-in spectrum as its total power because these tones contain most of the power ($> 95\%$) of a microphone signal. Despite the effectiveness of this detection technique, V-Scope might not

Detected Ground truth	Digital TV	Analog TV	Microphone
Digital TV	94.9%	0.7%	4.4%
Analog TV	0.5%	97.4%	2.1%
Microphone	1.2%	0.7%	98.1%

Table 2.3: Accuracy of primary detection algorithm.

be able to sense all the intermittent microphone transmissions because a client visits each location only for a short amount of time. To guarantee sufficient protection, we envision the databases to protect licensed microphones as is. As our future work, we intend to enhance our system with static sensors deployed in the proximity of microphone devices to reliably capture their transmission.

To benchmark the accuracy of our detection algorithms, we collected spectrum data from 30 UHF channels at multiple locations. We established the ground truth results by using a TV receiver and our microphone transmitters. The identified primary signals were further attenuated for constructing spectrum traces at a wide range of power (-40dBm to -114dBm). A standard cross validation was then performed by randomly choosing 90% spectrum traces to detect the rest 10%. Table 2.3 shows the accuracy of our detection algorithm. We observe reasonably low error rates (<5%) in detecting different types of signals.

Summarizing, our measurement procedure works as follows. (a) A client captures spectrum fragments around pilot frequencies in each UHF channel. (b) It extracts a potential pilot by searching for the maximum FFT bin and including all the surrounding bins with power above a threshold. (c) From the obtained FFT bins, several features (e.g., power, center frequency) are extracted and fed to a classifier for detecting TV pilots and broadcast type (analog or digital). (d) If a pilot is detected, the client estimates its total power by adding a specific power offset η to the pilot power. (e) The estimated power is then compared with -114dBm threshold to determine the presence of TV signals. (f) If no TV signal is detected in a given channel, the client further captures a full-channel spectrum to detect microphones and unlicensed devices while measuring their in-band power. g) If the full-channel spectrum contain potential microphone tones, a

narrow-band capture is performed around each spike for validation. The entire procedure takes less than 1 second for processing all 30 UHF channels at a location, and produces results of *signal type* and *power* in each channel.

Measurement clustering

Since a V-Scope client has classified measurements to different device types (i.e., TV, microphone, whitespace devices), our clustering module further segregates measurements among devices of a same type. We developed a preliminary algorithm that leverages k-medoids clustering based on the *power* and *inter-distance* of measurements. k-medoids clustering aims to minimize the sum of pairwise dissimilarities of measurements in each cluster. We define the dissimilarity metric between pair of measurements using gower dissimilarity score [31]. A unique feature of gower dissimilarity score is its capability of assigning different weights to observations. By assigning higher weights to stronger measurements, our algorithm is robust to noise and outliers in choosing the center of clusters. Specially, we take only measurements with power $P_n > -90$ dB to reduce computational overhead. With N selected measurements, we first calculate distance between each pair of measures. This gives us a N -by- N symmetric matrix, in which the (i, n) th entry d_{in} indicates physical distance between measure i and n . We call columns of this matrix as variables, gower dissimilarity score between the i th and j th measure (row) is defined as a weighted sum of dissimilarities for each variable $\frac{\sum_{n=1, \dots, N} w_n D_{ij}^n}{\sum_{n=1, \dots, N} w_n}$, in which D_{ij}^n is defined as $|d_{in} - d_{jn}|$ and the weight of variable n is defined as $w_n = \log(P_n + 90)$. We iteratively apply this algorithm with different number of clusters k between $1 - k_{max}$, and determine the optimal k that can maximize the *silhouette* distance [81]. k_{max} is set to be 100 in our implementation and a R package [49] is used to perform clustering. To evaluate this algorithm, we merged groups of measurements corresponding to different transmission locations of our microphone device in dataset C. We applied our algorithm on the aggregated measurements, and found only 6% measurements mis-classified to a different location. One limitation of our current approach is that mis-classification may happen when devices are close to each other. However, the

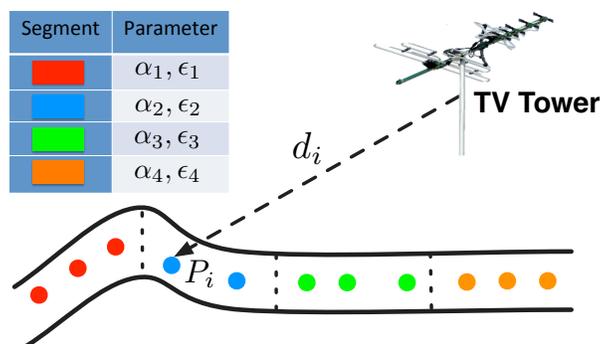


Figure 2.8: Illustration of region models in V-Scope.

mis-classified measurements are generally far from a device location and have relatively lower power, thus imposing limited impact on the overall prediction accuracy of V-Scope.

Region model for predicting signal strength of different devices

Using the clustered measurements of each device, V-Scope refines the parameters of a propagation model that can be used to better predict its signal strength in the vicinity of measurements. To choose an appropriate model, we note that most of the UHF-band models can be generalized in a form of $P = \alpha \log_{10}(d) + \epsilon$, where P is the power of a device at a reception location, d is the distance between this location to the device, α is the rate at which the signal power attenuates over an increasing distance, and ϵ captures both the transmission power of a device and the fixed attenuation of environmental shadowing. The difference of these models lies in how the parameters α, ϵ are determined, e.g., based on antenna height (in Egli, Hata) and various environmental factors (in Longley-Rice, R6602). Since the measured signal strength can best reflect a propagation environment, V-Scope uses this general model and calculates α, ϵ based on measurements.

A standard model fitting procedure [68] solves α, ϵ by plugging measurements P_i, d_i into a given propagation model. This forms a set of linear equations $P_i = \alpha \log_{10} d_i + \epsilon$. It uses least-squares linear regression to

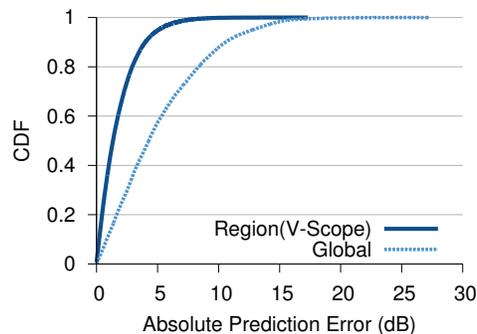


Figure 2.9: Accuracy of a fitted region model and a global model in predicting the power of TV signals.

calculate α, ϵ , with the objective of minimizing the squared sum of fitting errors $\sum_i (P_i - \alpha \log_{10} d_i - \epsilon)^2$. V-Scope improves this approach by (i) fitting an individual set of parameters for each local area to better model its propagation environment, and (ii) performing a weighted regression to avoid fitting bias caused by non-uniform distribution of vehicular measurements.

Region-specific model: V-Scope groups measurements into road segments and fits a different set of parameters for each segment as shown in Figure 2.8. The motivation is that these regions are likely to have different propagation characteristics, especially in an urban environment. This region-specific variation can hardly be captured by a global propagation model using a single set of parameters. To demonstrate this, we fit different α and ϵ based on measured signal strength of TV broadcasts in each 100m road segment. We compare this model with a global model fitted with all the measurements in predicting the strength of TV signals.

Figure 2.9 shows that our region model achieves a median error of 1.4dB and 75 quartile error of 2.6dB, which are $3\times$ and $2.9\times$ lower than a global model. Thus, V-Scope fits a different set of model parameters (α, ϵ) to better capture each local propagation environment.

Weighted regression fitting: In fitting a region model, we note the non-evenly spaced vehicular measurements can degrade the performance of linear regression. Since a public vehicle drives at a varying speed and stops quite

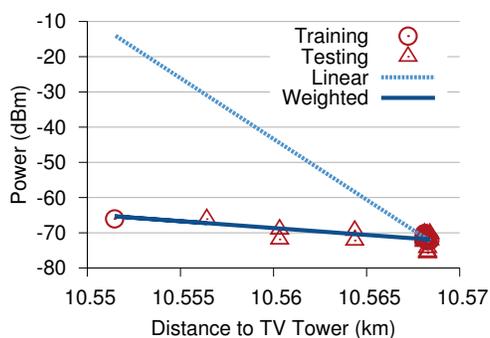


Figure 2.10: Region models fitted by linear regression and weighted regression in predicting a TV broadcast.

often, V-Scope collects measurements at non-uniform density. This causes linear regression to produce a biased model that favors *densely* measured area, and has large errors at *sparsely* measured area. The underlying reason is that linear regression aims to minimize the squared sum of fitting error; the area with sparse measurements contribute less to this squared sum, thus being under-fitted. Figure 2.10 shows an example where we fit a model for a 100m region with most of training measurements collocated at a bus top. We observe that the model fitted by linear regression has up to 36dB error in predicting TV signal strength at the testing locations.

V-Scope uses weighted regression to compensate the effect of non-uniform measurement density in model fitting. The algorithm assigns a weight W_i to each measurement i , with the objective of minimizing the weighted squared sum of fitting errors $\sum_i W_i (P_i - \alpha \log_{10} d_i - \epsilon)$. A higher weight can indicate greater importance in fitting a measurement. Therefore, we assign *higher* weights to *sparse* measurements to compensate for the difference in measurement density in model fitting. To accurately capture this measurement sparsity, we calculate the weight of a measurement based on its total distance to other measurements $W_i = \sum_j \text{dist}(i, j)$. Figure 2.10 shows our model fitted by weighted regression achieves high prediction accuracy at all the testing locations.

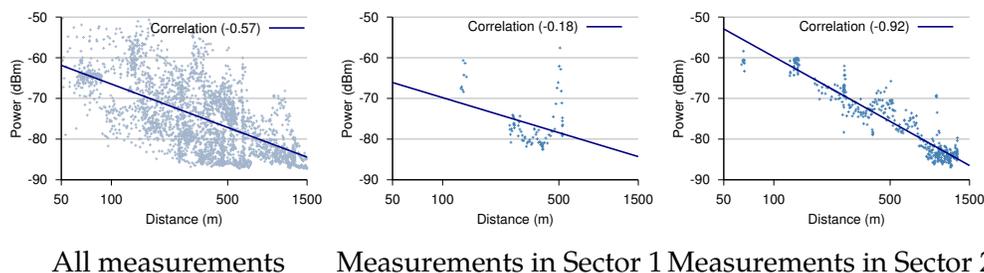


Figure 2.11: Measured signal strength vs. propagation range (in log scale) of a whitespace transmitter.

To recapitulate, our model fitting procedure first bins measurements of each device into road segments. For each segment, it computes a weight for each measurement based on its distance to other measurements. It then takes these measurements and their weights as the input to weighted least-squares regression for constructing region models.

The fitted model will then be used to predict for a m -by- m square-shape region centered around the measured road segment. The region size m is the length of the road segment. We will explore the performance tradeoff between the prediction accuracy and storage overhead of different region sizes in § 2.4. Finally, beyond the coverage of these regions, the database may use its default model (R6602) for predicting TV whitespaces.

Sector based localization

The above model fitting procedure requires the location of a transmitting device to calculate the distance d_i for each measurement. V-Scope leverages popular *RSSI modeling* techniques [12, 75] to localize a device if its location is unknown or awaits validation. Our motivation of using this technique comes from its flexibility in localizing a device from arbitrarily measured locations, and the simplicity of using signal strength (RSSI) information that is readily available from spectrum measurements. Such an approach works similarly to model fitting. It uses signal strength measurements and a propagation model to form a set of equations $P_i = \alpha \log_{10} d_i + \epsilon$. Here d_i is replaced with

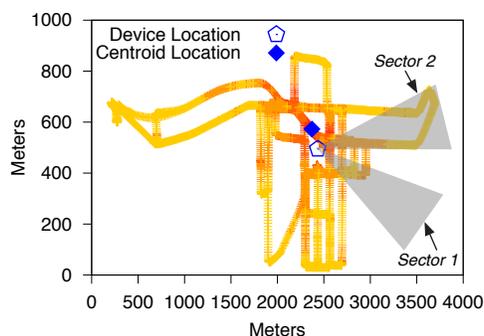


Figure 2.12: Measurement locations for a whitespace transmitter. Darker color indicates higher power. Radiation sectors in Figure 2.11 are marked.

$\sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}$, where (x_t, y_t) is the transmitter location and (x_i, y_i) is the location of each measurement. The algorithm solves $\alpha, \epsilon, x_t, y_t$ to estimate a device's location. V-Scope adapts this approach to outdoor scenario by (i) carefully selecting measurements in certain radiation sectors of a device and (ii) constructing a *sector-specific* model based on the chosen sectors.

While a RSSI modeling approach is reported to achieve a high accuracy within few meters in an indoor scenario, directly applying this technique to our outdoor area leads to large localization error (up to 100m in § 2.4). The reason is that its underlying propagation model matches poorly with the signal strength pattern of a transmitting device in an outdoor scenario. We use the wardriving measurements collected for our whitespace device in dataset B to demonstrate this. The first panel of Figure 2.11 shows the received signal strength over different transmission ranges for all the measurements. A propagation model expects a *linear* trend of the measured signal strength P_i over an increasing distance $\log_{10}d_i$, but we observe many measurements deviate from this trend (the fitted line). We use Pearson correlation coefficient to quantify the linearity of these measurements, with 1 and -1 being an exact positive and negative linear trend and 0 implying no correlation. We find the correlation value to be merely -0.57.

Such a poor linear trend is caused by surrounding environment, e.g., terrain elevation, obstacles, etc. To illustrate this, we decompose measurements

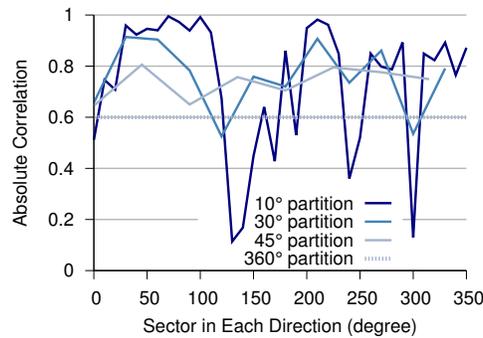


Figure 2.13: Absolute correlation of all the sectors partitioned from the centroid location with different angles.

according to different radiation sectors of the device with 10-degree angle as shown in Figure 2.12. The second panel of Figure 2.11 is a scatter plot for measurements from one such sector. We observe a sharp drop of signal strength at a distance between 350 – 500m due to a building blocking the transmission, but a drastic increase in signal strength beyond 500m due to the rising terrain elevation coupled with diminishing blocking effect of the building at faraway locations. This environmental-induced variation can largely *perturb* a propagation model in capturing the large-scale path loss, leading to large localization error. Fortunately, we find measurements in some other radiation sectors present a less noisy propagation trend as show in the third panel of Figure 2.11. This is owing to the environmental shadowing being unlikely to affect the signal along all the directions. Thus, V-Scope only uses measurements in a few sectors that present a good propagation trend for localization.

Localization procedure: Our proposed algorithm proceeds in five steps. (a) We use the *centroid* location of the *strongest* measurements (top 5dB) as the partitioning center, while grouping measurements into radiation sectors at a narrow angle (10 degree). The intuition behind this heuristic approach is that the centroid location is usually not far from a device’s actual location (e.g., 100m in Figure 2.12); and by using a narrow angle, some radiation sectors are likely to “fall through” the gaps of a blocking environment, thus having measurements following a good propagation trend. To validate this intuition, Figure 2.13

shows the *absolute* correlation of measurements in different sectors partitioned at various angles for the whitespace device. We observe that the narrowest angle used in our implementation indeed leads to best correlation (>0.9) for some sectors. (c) Post partitioning, we calculate the Pearson correlation value between the measured signal strength and the transmission range for those sectors that have at least some measurements present. We then select the sectors with a high correlation value (top 0.1 bin) for localization. (d) Given these candidate sectors, we use a *sector-specific* model with different α_j, ϵ_j for each sector j . This is because these chosen sectors can still have a different propagation trend in slope and intercept due to environmental variation. We omit this result for the sake of brevity. (e) We construct linear equations based on measurements from all the chosen sectors, and solve x_t, y_t and all α_j, ϵ_j to estimate a device's location using an optimization function from a statistics package[71].

Adjacent-channel model for predicting TV broadcast leakage

V-Scope builds a model to predict the leakage power of a TV broadcast into its adjacent channels based on its in-band power¹. It first identifies those locations with adjacent-channel measurements classified to be noise because the leakage is simply high-power noise. It then leverages a *linear* relationship between the measured power of this TV broadcast and that of its adjacent channel to construct the model. To demonstrate this power relationship, Figure 2.14 shows the power of a TV broadcast in channel 26 and that of its leakage in channel 27 at each chosen location. We observe that the leakage power increases approximately along a 45-degree line with TV power, and can be as high as -65dBm. The Pearson correlation between these datasets is 0.91. This prominent linear relationship comes from both a TV signal and its leakage being transmitted from a *same* location and at a power of *constant* difference. After traversing along similar paths to any location, their power still differ approximately by this amount leading to a linear trend between the in-band power and adjacent-channel power.

¹Different TV broadcasts are usually allocated in far apart channels, and their adjacent channels therefore do not overlap.

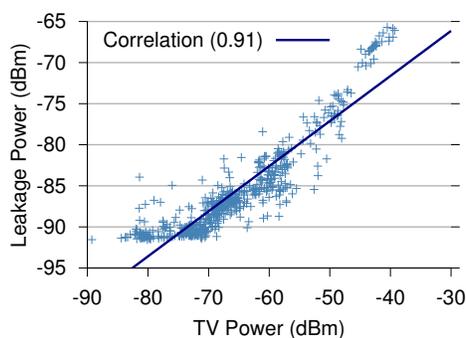


Figure 2.14: Power of a TV broadcast vs. power of its adjacent-channel leakage at different locations.

Since the specific leakage trend is different among TV transmitters, V-Scope constructs an individual model for each TV broadcast. The model takes the form $P_l = \gamma P_{tv} + \beta$, where P_{tv} and P_l are the power of a TV signal and its leakage at a given location, and γ, β are tunable parameters to capture the power relationship between these signals. V-Scope calculates γ, β by applying linear regression on the power measurements. To estimate the leakage of a TV broadcast at a given location, the databases can first use our region model to predict its in-band power P_{tv} , then leveraging the leakage model to predict P_l .

2.4 EXPERIMENTAL RESULTS

We evaluate the performance of V-Scope based on three datasets collected over a 120 square-km area in and around a US city as described in § 2.2. We start by evaluating the overall performance of V-Scope in § 2.4. We then benchmark the performance of individual components in § 2.4. Overall, we find V-Scope can reduce false negative rates for protecting TV broadcasts by up to 59%, identify all the suitable whitespace channels at 72 – 83% locations under different channel quality constraints, and localize unlicensed devices at various locations with an error of 16 – 27m.

Implementation: We have implemented the measurement collection module and all the data processing modules in 7500 lines of Python, and built a database query utility in 650 lines of C++.

Methodology: We apply a standard five-fold cross-validation by using 80% randomly selected measurements to construct V-Scope models. The fitted models are used to predict the power of TV broadcasts, unlicensed devices and adjacent-channel leakage at remaining measured (testing) locations. We compare the predicted power of TV broadcasts with the -114dBm threshold to determine TV whitespace, while combining the predicted power of unlicensed devices and adjacent-channel leakage to estimate the noise power of whitespace channels. We compared our predictions with ground truth results based on measurements at these testing locations. To evaluate localization, we use measurements for our whitespace device and microphones operating at various locations, with ground truth locations determined by a GPS device.

Evaluation metrics: We use a variety of metrics to evaluate the performance of V-Scope such as false positive and false negative rates in predicting the availability of TV whitespaces, number of inappropriate whitespace channels selected under different channel quality constraints, absolute error in predicted signal strength, localization error between the actual location and predicted location of different TV-band devices.

Overall performance of V-Scope

We use dataset A to evaluate overall performance of V-Scope in predicting whitespace channels and estimating their quality. We quantify the gain of V-Scope over a FCC-approved database [92] that predicts whitespace channels based solely on a propagation model without distinguishing their quality. We also evaluate the performance of V-Scope models fitted under different region sizes. This can provide database operators with insights about the tradeoff between prediction accuracy and storage overhead in choosing an appropriate region size for different spectrum management tasks.

Predicting TV whitespace spectrum: We start by quantifying the false negative rates of the database and V-Scope models fitted at different region

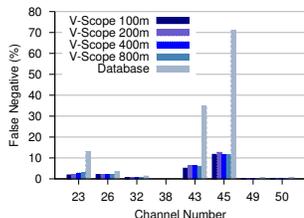


Figure 2.15: Fraction of locations under-utilized by V-Scope and the database for predicting different TV broadcasts.

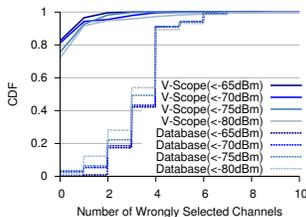


Figure 2.16: CDF of number of whitespace channels mis-selected at different locations under various quality constraints.

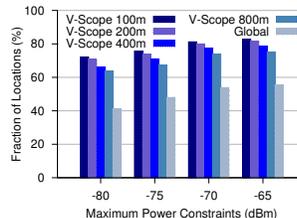


Figure 2.17: Fraction of locations where all the suitable channels are correctly selected by different V-Scope models.

Approach	Database	VS-100m	VS-200m	VS-400m	VS-800m
False Positive Rate	0.29%	0.29%	0.22%	0.17%	0.17%

Table 2.4: False positive rates of the database and V-Scope models fitted for different region sizes in predicting TV broadcasts.

sizes for predicting TV broadcast channels that are accessible at least in part of our measured area. Figure 2.15 shows that different V-Scope models can reclaim the spectrum wastage of the databases by up to 59% locations. We then present the false positive rates averaged across different broadcast channels for these approaches in Table 2.4. Compared to the conservative database, we find the V-Scope models are able to offer same or even better protection to primary users. One important observation is that the V-Scope models fitted under different region sizes present little difference in the accuracy of whitespace determination, e.g., $\leq 2\%$ in false negative rate and $\leq 0.1\%$ in false positive rate. This is because most of their differences in predicted signal strengths are masked when comparing them with the detection threshold. We may therefore use a large region size (e.g., 800m) to fit V-Scope models for TV broadcasts, thereby reducing the overhead of storing model parameters.

Signal Type	25th	Median	75th	95th	99th
TV broadcast	0.2dB	0.5dB	1.1dB	3.2dB	7.7dB
Unlicensed signal	0.2dB	0.6dB	1.3dB	3.4dB	6.9dB
TV leakage	0.3dB	0.7dB	1.4dB	3.6dB	6.9dB

Table 2.5: Absolute error in predicting the power of different types of signals by a 100m V-Scope model.

Selecting suitable whitespace channels: Using the predicted whitespace channels, we evaluate V-Scope and the database in selecting appropriate channels under different power constraints. A whitespace channel is deemed to be suitable if its interference power is below a given constraint. Such a quality constraint can be estimated by network operators based on parameters such as the distance of wireless links, transmission power, and the minimum signal-to-noise ratio (SNR) for decoding a received signal (under desired modulations).

Figure 2.16 shows the CDF of the number of wrongly selected channels at different locations for a 100m V-Scope model and the database under different quality constraints. Note that a 5dB increase in two consecutive power constraints can lead to a 15 – 30Mbps drop in the achievable PHY rates for the 802.11n technology [99]. Without attempting to distinguish channel quality, the database can select all the appropriate channels at less than 2% of the locations, and have 3 – 4 channels wrongly selected for 50% of the locations. In contrast, V-Scope correctly selects all the qualified whitespace channels at 72% – 83% locations, and mis-predicts at most 1 channel for 92% – 97% locations. The much higher accuracy suggests that V-Scope can help avoid most of the performance penalty on a whitespace network due to channel mis-selection. We break down the accuracy of V-Scope in predicting the power of different types of signals in Table 2.5. We note for all the signal types, the median error is below 0.7dB and a 95 quartile error below 3.6dB, thus explaining its high accuracy in channel selection.

We now study the impact of region sizes on the accuracy of V-Scope models for selecting suitable whitespace channels. Figure 2.17 shows the fraction of locations where all the appropriate channels are correctly identified by these

models. We observe that the accuracy degrades at a larger region size. For example, a 800m region model correctly selects all the suitable channels at 8% – 11% fewer locations than a 100m region model, and a global model has 27% – 31% fewer such locations than a 100m model. This is because a model fitted for a larger region is not fine-tuned enough to capture local environment. The database operators may use this information to choose an appropriate region size for channel quality estimation based on *available storage* and *desired accuracy*.

Microbenchmarks of V-Scope

We now benchmark the performance of individual components in V-Scope, which are region-specific model in predicting TV broadcasts, sector-based localization algorithm, region-specific model in predicting unlicensed signals, and adjacent-channel models in predicting the leakage of TV broadcasts.

Accuracy in predicting TV broadcasts: We compare the V-Scope model with those models fitted by two alternative approaches. *Global* is a single model fitted for the entire measured area. *Local* is a region model fitted by linear regression instead of weighted regression. We quantify the gain of V-Scope models based on two datasets — dataset A collected during a 6-week period and a subset of it collected in an initial week.

The first panel of Figure 2.18 shows the 99th quartile error of different models fitted based on the 1-week dataset. We first observe that *Global* has the highest prediction error since a global model can hardly be tailored to different local propagation environments. We then note a 19% – 40% reduction in prediction error achieved by *Local* over *Global* because *Local* tunes an individual model to each small region. V-Scope outperforms *Local* by 8% – 13% due to the use of weighted regression to compensate the non-uniform measurement density in model fitting.

As more measurements were subsequently collected, the measurement density tends to become uniform in each road segment. As a result, the second panel of Figure 2.18 shows *Local* achieves a similar accuracy to V-Scope based on the 6-week data. This suggests that both versions of regression apply well on uniformly distributed measurements, but our weighted regression has its

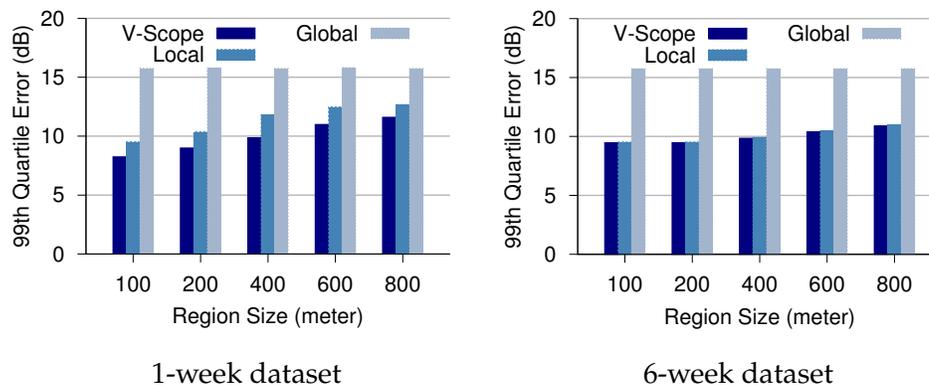


Figure 2.18: Accuracy in predicting the power of TV signals.

unique advantage in dealing with *non-uniform* measurements collected during a *short* wardriving period. Finally, we observe the accuracy in predicting the TV power improves at a smaller region size from both datasets.

Accuracy in localizing unlicensed devices: We use measurements collected for our whitespace transmitter and a microphone device in dataset B and C for evaluating our localization technique. The ground truth locations of these devices were obtained by a GPS device as mentioned before. We compare following localization techniques.

Single-deter and *Single-prob* are two popular RSSI modeling techniques used in EZ system[12] and WiFiNet system [75] respectively. *Single-deter* uses a deterministic propagation model as described in § 2.3, whereas *Single-prob* uses a probability model based on the same propagation trend. *Sector-deter* and *Sector-prob* are our sector based versions of these common techniques. *Centroid* is the center location used by our technique to partition sectors, which is the geometric center of the strongest measurements (top 5dB).

Figure 2.19(left) shows the error of different algorithms for localizing our whitespace device. The measurements were selected based on different maximum power thresholds to emulate a variety of localization environment. We first observe that *Single-deter* achieves a low error of 26.9m using all the measurements (-45dBm threshold). Under different power thresholds, our sector based techniques improve *Single-deter* and *Single-prob* by $1.2 - 3\times$ and 1.5

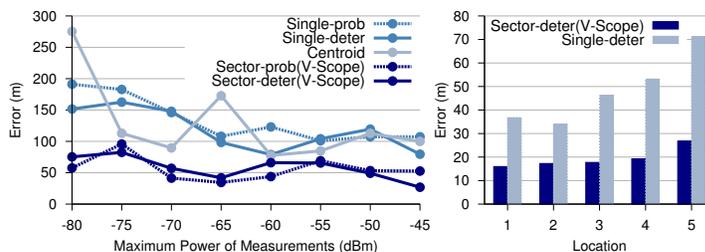


Figure 2.19: Accuracy in localizing a whitespace device (left) and microphone device (right).

– $3.5\times$, because they carefully choose a few sectors and develop an individual propagation model for them. *Sector-deter* also refines the partitioning center *Centroid* by $1.2 - 4.1\times$. The error of *Sector-deter* and *Sector-prob* increases by up to $2.8\times$ and $1.8\times$ when using measurements at a lower power (e.g., -80dBm threshold). This is because the weak measurements present a less distinct path loss trend and the partitioning center (*Centroid*) deviates more from the device’s actual location.

Figure 2.19(right) shows the accuracy of *Sector-deter* and *Single-deter* for localizing our microphone device in 5 different buildings based on clustered measurements as described in § 2.3. The maximum measured power ranges from -70dBm to -60dBm for different operating locations of the device. We observe that our technique *Sector-deter* achieves a low error between $16 - 27\text{m}$, which are $2 - 2.8\times$ lower than *Single-deter*.

To understand the effect of partitioning angles on the performance of sector based localization approach, The first panel of Figure 2.20 shows the accuracy of *Sector-deter* with different partitioning angles for localizing the whitespace device. We observe that a larger angle leads to worse performance, e.g., the error increases by $1.5\times$ from the 10-degree angle to the 30-degree angle, and by $2.06\times$ to a 360-degree angle that is the baseline algorithm – *Single-deter*. To explain this performance degradation, we present the absolute correlation ratio averaged over the selected sectors for localization. Note that these sectors have a correlation ratio within the top 0.1 bin of all the sectors (§ 2.3). The second panel of Figure 2.20 shows the absolute correlation decreases over an increasing angle, with a 360-degree sector having 0.36 lower correlation than a 10-degree . The

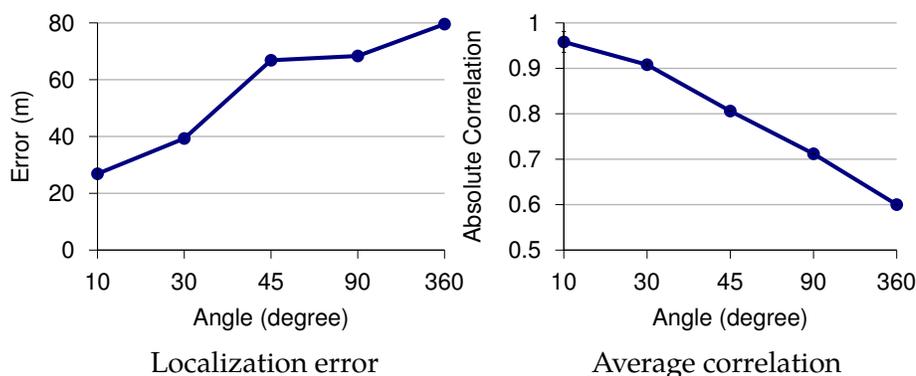


Figure 2.20: Performance of different partitioning angles used by our approach for localizing the whitespace device.

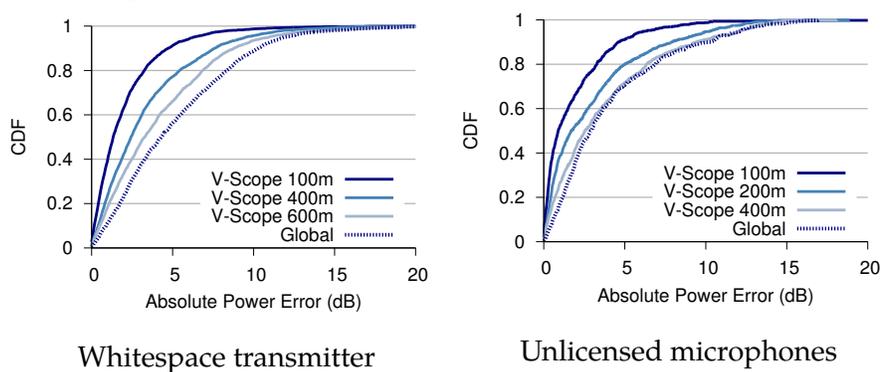


Figure 2.21: Accuracy in predicting the power of unlicensed devices.

highest correlation of 10-degree sectors indicates a best match of the selected measurements to the linear path loss trend, which in turn leads to highest accuracy. This justifies our use of the narrowest partitioning angle in the final algorithm.

Accuracy in predicting unlicensed signals: Using the device locations predicted by *Sector-deter*, we construct region-specific models for predicting the signal strength of the whitespace device and all (5) the microphone instances. Figure 2.21 shows the CDF of absolute errors in predicting the signal strength of different types of devices. We observe a 100m V-Scope model can achieve a median error of 1.3dB and 0.9dB, and a 75-quartile error of 2.8dB and 2.7dB for predicting the whitespace and microphone devices respectively. The prediction

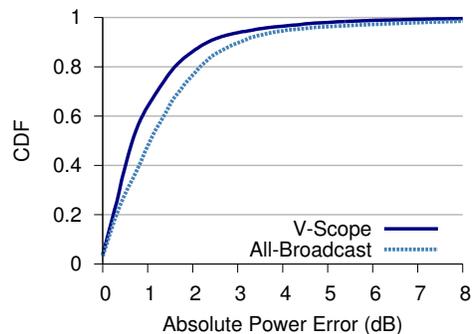


Figure 2.22: Accuracy in predicting the leakage power from TV broadcasts.

error increases at a larger region size, with a global model having about $3\times$ higher median error than a 100m region model for both types of devices.

Accuracy in predicting TV broadcast leakage: We compare our leakage model tuned to individual TV broadcasts with an alternative model (*All-Broadcast*) comprising a single set of parameters fitted for all the TV broadcasts. Figure 2.22 shows the CDF of absolute error for predicting the TV leakage power in all the adjacent channels based on dataset A. We observe that V-Scope achieves a 0.7dB median error and a 1.4dB 75-quartile error. *All-Broadcast*, albeit with slightly lower storage overhead, has a $1.6\times$ higher error in median and $1.4\times$ higher error at 75-quartile. The worse performance in *All-Broadcast* is because a single model fitted for all the TV broadcasts cannot accurately capture the specific leakage characteristics of each TV transmitter. Since the overhead of storing a separate set of model parameters for a few TV broadcasts is low, we choose this *broadcast-specific* model in our final design to improve accuracy.

2.5 ISSUES AND DISCUSSION

V-Scope has taken an initial but important step towards augmenting spectrum databases using wide-area measurements. We comment on its impact on spectrum regulations, its performance limitations and possible enhancements.

- **Impact on FCC ruling.** To harvest the benefits of V-Scope, we envision several changes to be made to the current FCC ruling. To ensure the validity of spectrum measurements, the measurement hardware needs to be rigorously tested to ensure the required detection accuracy, perhaps following the same testing procedure for the sensing-based whitespace devices. To utilize the opportunistic measurements, the database may choose an appropriate region size to fit the V-Scope models based on the desired accuracy. It may fall back to use its default model (R6602) when predicting those areas beyond the regions covered by measurements. Finally, while we find non-negligible temporal and spatial under-utilization of whitespace spectrum in microphone protection, our system has *inherent limitations* in reliably protecting licensed microphones with the use of vehicular measurements. To achieve this goal, perhaps some static sensors can be deployed in the proximity of microphone reserved locations for continuously monitoring their activity, or a proactive protocol can be adopted by microphones to alert interference from whitespace devices as suggested in prior work [64].
- **Coping with deployment cost.** Our opportunistic wardriving approach has costs and overheads in deploying and managing whitespace sensors. Perhaps this measurement infrastructure is most useful in urban areas where seizing additional spectrum of good quality can be particularly beneficial to users. In such scenarios, spectrum database providers may contract with public vehicle operators to deploy this infrastructure, and recoup their costs by charging additional fees to use the services enabled by them. In this paper, we do not explore the economic aspects of opportunistic wardriving, but focus on the technical aspects in collecting and utilizing spectrum measurements to augment databases.
- **Addressing temporal variations and storage overhead.** The collected measurements can become invalid with the change of environment and transmission behavior of (secondary) devices. While such temporal variation is observed to be small ($<5\text{dB}$) at most of our measured locations, it can increase with the future proliferation of whitespace devices. We

envision to use time-based filtering, backed up by statistical algorithms such as Dixon's test[18], to identify up-to-date measurements. To mitigate storage overhead, the databases may only store the fitted model parameters and device locations while discarding the measurements after using them.

- **Impact of measurement volume and coverage.** Since our measurements are collected on a public transit bus opportunistically, the number of measurements in each region vary to a large extent. For example, different 100m regions can have measurements ranging from two to hundreds. We do not observe any clear trend between the number of measurements and the accuracy of the fitted model. Instead, the location of these measurements can play a more significant role. For example, using densely distributed measurements to fit a model can have lower accuracy than using a few measurements collected on sparse locations, which is the very motivation for our weighted regression model. Finally, the evaluation of our region model is not applicable to those locations beyond road segments. However, for those neighborhoods surrounded by the roads, we expect similar fading and shadowing effects introduced by the same environment, hence rendering our models to be relatively useful in predicting these areas.

2.6 SUMMARY OF V-SCOPE

In this chapter, we presented V-Scope, a measurement system that leverages public vehicles to collect wide-area measurements to augment whitespace spectrum databases. V-Scope applies an advance feature detection algorithm on the vehicular measurements to detect primary signals at the -114dBm threshold, while using the measured signal strength to refine various propagation models. These models can enhance spectrum databases to better predict whitespace channels, estimate their channel quality, and pinpoint the location of primary and secondary devices.

We have deployed V-Scope on a metro bus traveling in Madison, WI, and strived to collect measurements at one million distinct locations over a 120 sq. km. area. Using the substantial measurements combined with our statistic approaches, we have shown that commercial databases cause under-utilization of whitespace spectrum over a wide area (up to 71% measured locations). We also reported large variation (up to 50dB) in the quality of whitespace channels, which the databases are not designed to capture. V-Scope can enable spectrum databases to reclaim the spectrum wastage by up to 59% locations. It further allows databases to identify all the suitable whitespace channels at 72 – 83% locations under different channel quality constraints. In addition, V-Scope can localize TV-band devices at a decent accuracy of 16 – 27m, improving state-of-the-art techniques by 1.2 – 3.5 \times in outdoor scenarios.

3 A HETEROGENEOUS NETWORK ARCHITECTURE TO EXTEND COVERAGE

3.1 MOTIVATION

We now focus our attention on how to efficiently use the vacant whitespace spectrum to provide wide-area network coverage. We investigate this question in a particularly challenging use case — providing vehicular Internet access. Such a service has become increasingly popular for diverse applications, e.g., navigation, infotainment, and driving assistance. Numerous research projects such as MAR [80], WiRover [36], ViFi [7], Wiffler [6], CaberNet [22], along with commercial endeavors, manage to provide this today through existing cellular technologies and opportunistic WiFi access. We have explored the use of TV whitespaces not *merely* because of the low spectrum cost, but for its excellent propagation characteristics to match the needs of vehicular connectivity. Our long term deployment goal is to provide an “on-board” Internet service for a city metro transit operating hundreds of buses at Madison, WI.

To efficiently utilize TV whitespaces, we propose a *heterogeneous* network design called *Scout*, which (i) communicates the downlink traffic primarily over TV whitespaces paths while sending the uplink traffic over existing cellular paths, and (ii) uses an additional “scouting” radio on a vehicle to probe the channel condition in advance, thereby compensating for the high feedback latency in making protocol decisions. Figure 3.1 (left side) shows our proposed 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 gateway serves as a WiFi hotspot inside the bus, allowing users to connect to the Internet through their WiFi-capable devices. It relays users’ WiFi traffic through an aggregation proxy situated behind the whitespace and the cellular networks to Internet.

Why a heterogeneous network design? Our use of the heterogeneous architecture comes from the asymmetric transmit power limits widely imposed by regulatory agencies [25, 94]. That is, the transmission power of mobile

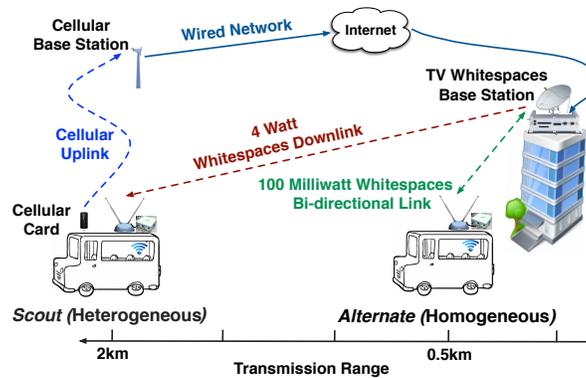


Figure 3.1: *Scout* uses TV whitespaces path primarily for downlink traffic and cellular path primarily for uplink, unlike a traditional, homogeneous design that uses the same path for both directions.

whitespace devices is limited to 100 mW, whereas the power of static base stations can be up to 4 W. This $40\times$ difference in the power limits is to prevent mobile devices from causing harmful interference during roaming. Since most of networking applications have bi-directional traffic, a conventional whitespace-only network, as depicted in Figure 3.1 (right), would limit the operating range of a base station to that of “weaker” mobile clients. The significant reduction in operating range (from 2km to 2m) requires much more whitespace base stations to be deployed, largely increasing the infrastructure cost.

To address the power asymmetry issue, *Scout* employs a heterogeneous network architecture where the downlink traffic is mainly communicated over TV whitespaces, while the uplink traffic is sent over a cellular link. The extensive cellular connectivity has helped to circumvent the weak whitespace uplink, thereby maximizing the downlink coverage of each whitespace base station. Further, since many applications are downlink dominated ($10\times$ in WiRover [36]), *Scout* is efficient for using unlicensed whitespaces to offload network traffic.

Why the use of a scouting radio? While effectively extending the coverage, a heterogeneous network based on traditional networking protocols yields poor downlink performance. This is because of the *high feedback latency* in the cellular uplink, which leads to inaccurate channel estimates, especially

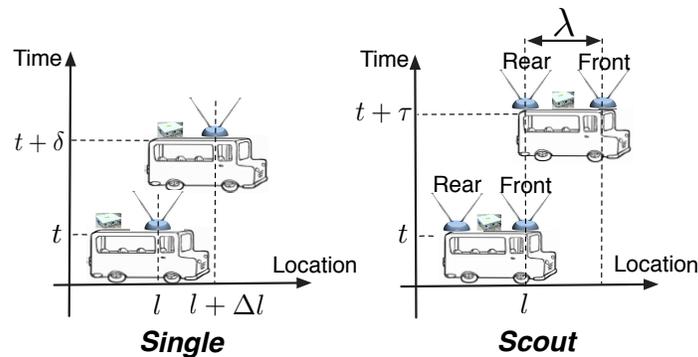


Figure 3.2: Diagram of *Single* and *Scout*. λ is the antenna separation, τ is the rear radio's travel time to front position, and δ is the cellular path delay ($\delta \leq \tau$).

in the mobile environment. Since most of communication systems rely on channel estimates for making *various* protocol decisions, e.g., rate adaptation and FEC, the stale feedback leads to higher packet losses. Further, the high uplink latency drastically inflates the bandwidth delay product of whitespace networks, rendering TCP based applications extremely susceptible to channel losses. It also slows down retransmissions for loss recovery, which further exacerbates this bandwidth inefficiency.

To deal with the slow feedback, we explore the use of an extra “scouting” radio to accurately measure the channel condition for any future reception location. Our core intuition is that the location of a radio largely determines its experienced channel condition [61, 63, 83]. For a single-radio system as shown in Figure 3.2 (*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 delayed feedback. In contrast, if we were to place two radios as shown in Figure 3.2 (*Scout*), the channel condition experienced by the front radio would be somewhat similar to that of the rear radio after it reaches the forward post. In essence, the front radio can “scout” the likely channel condition for future reception conditions of the rear radio. Hence, we can send the channel estimates made by the front radio over the cellular path. By the time the rear radio reaches the same location l , the base station can use this earlier feedback to choose better transmission parameters for the rear radio.

Note that our use of a scouting radio is *independent* to multi-antenna techniques [97] (MIMO) that combine signals at the physical layer for scaling throughput or enhancing reception robustness. Furthermore, our 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 § 3.5. Based on this scouting radio based channel estimation technique, we have developed multiple aggressive transmission adaptation techniques, i.e., rate adaptation, inter-packet FEC, and intelligent traffic duplication, to enhance the robustness of individual vehicular links.

How to coordinate multiple base stations? Built on this heterogeneous architecture, *Scout* further leverages multiple whitespace base stations to extend the network coverage and throughput. It achieves this through a central controller situated in the network backbone, which acts as an aggregation proxy to connect all the base stations to the Internet. It forwards downlink traffic through a best base station to each client, while forwarding its uplink traffic from the cellular path to the Internet. Such a centralized backplane allows efficient base station assignment based on the global view of network condition using an optimization technique (§ 3.3). It also uses a flexible packet forwarding framework to enforce any given channel contention policy, while supporting seamless client handoff — a feature often missing in existing systems [6, 7, 22, 36].

We have deployed *Scout* in a campus testbed with multiple base stations and vehicular clients spanning several square kilometers. Based on experiments conducted by driving multiple vehicles around various routes for hundreds of miles, we find that *Scout* can significantly enhance the robustness of vehicular connectivity, while maximizing the overall network throughput.

3.2 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

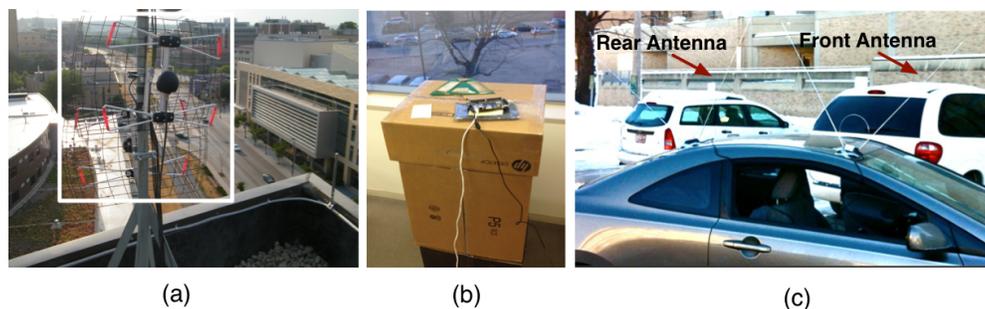


Figure 3.3: Outdoor deployment at the campus area of Madison, WI, USA: (a) High-gain antennas used by two base stations on top of a building; (b) The third base station deployed inside a building; (c) vehicular client with two TV antennas mounted atop.

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 to improve such decisions. We start by describing our experiment setup.

Testbed Setup

Our testbed is built around the UW campus area with a coverage radius of several kilometers. It currently consists of three base stations and several mobile clients. Figure 3.3(a) shows the antennas used by the two base stations mounted atop a 8th floor building. Each antenna has a 60 degree beam-width pointing in different directions along a bus route. Figure 3.3(b) shows the third base station deployed in a building, which uses a directional antenna focusing on another direction. Each base station uses a whitespace radio transmitting at a fixed frequency of 574MHz, 602MHz or 638MHz, and a 20MHz bandwidth spanning across 4 contiguous, unused TV channels according to a spectrum database [92]. The transmission power of these base stations are configured to be 4 watt as per FCC's ruling [25]. Each base station is equipped with an Ethernet connection to the central controller, which is running on a desktop in our lab. As the

mobile client, we use a personal vehicle carrying two whitespaces radios to receive downlink traffic as shown in Figure 3.3(c). Each radio is connected to an omni-directional antenna for capturing signal from all directions. A 3G cellular card is used for uplink communications.

Our radio platform is similar to the WhiteFi radio [4]. 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 § 3.4. 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.

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 (depicted in Figure 3.11) for 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 3.4 presents the MAC layer loss rates for two state-of-the-art rate adaptation algorithms (Minstrel [8] and RRAA [103]) 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.

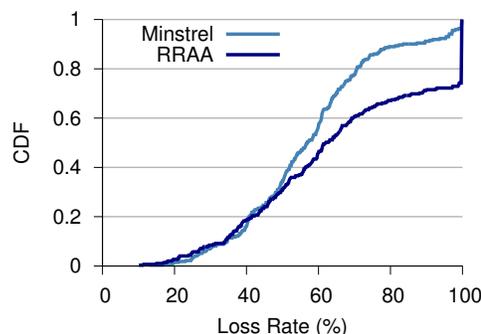


Figure 3.4: CDF of MAC layer losses of two state-of-the-art rate adaptation algorithms with mobility and under feedback delay in the cellular uplink.

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 § 3.5, for a single radio system using a combination of a state-of-the-art rate adaptation algorithm (RRAA), link 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*.

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 3.2. 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*.

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 3.5 (*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 3.5 (*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 3.5 (*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.

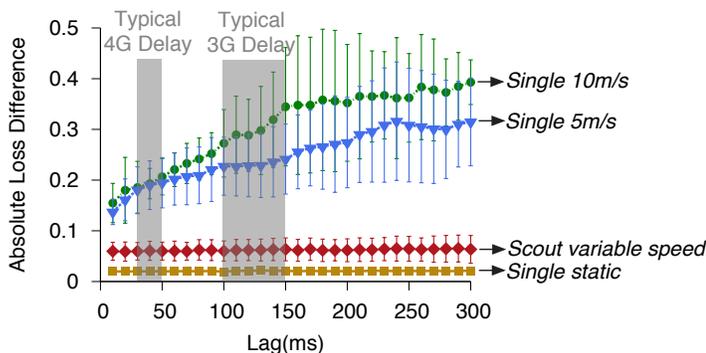


Figure 3.5: Average of absolute loss differences at various lags for 12Mbps packets. A lag is the elapsed time between two measurements. *Single* denotes one radio and *Scout* is the two radio setup.

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. We can therefore use this more accurate feedback to improve the robustness of vehicular connectivity by adapting multiple transmission adaptation techniques at the link layer, and assigning appropriate base stations at the network layer. We will describe these techniques in the next section.

3.3 SCOUT: A HETEROGENEOUS DESIGN

Scout leverages a heterogeneous network architecture to deal with the power asymmetry issue for maximizing the base station coverage. Figure 3.6 illustrates the two-tier architecture of *Scout*. At the top tier, a central controller connects all the base stations to the Internet. For downlink communication, it forwards

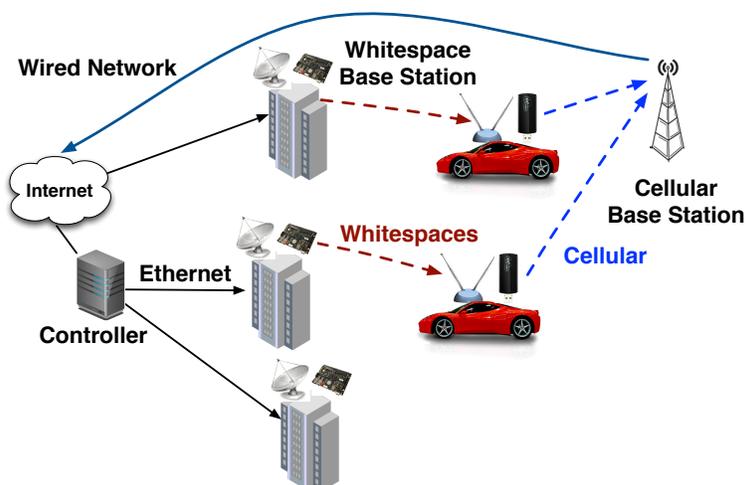


Figure 3.6: *Scout* network architecture. A central controller forwards downlink traffic from Internet to base stations, which use TV whitespaces to send traffic to vehicular clients. Each client sends uplink traffic over the cellular path directly to the controller, which forwards it to Internet.

a client's traffic to a selected base station for sending over whitespaces. For uplink traffic, the client sends it over a cellular path to the controller, which relays it to the Internet. At the bottom layer, each base station uses a whitespace radio to transmit at the maximum transmit power (4W), thereby extending the downlink coverage to its clients. It further duplicates a fraction of downlink traffic *intelligently* when the whitespace link is experiencing an outage. Each client is equipped with two radios to receive the downlink traffic for channel estimation and reception diversity. Based on the received downlink packets, the client sends acknowledgment packets (ACKs) along with its GPS information over the cellular path to each corresponding base station, which leverages this feedback for transmission adaptation and link capacity estimation (§ 3.3). We now explain each layer of *Scout*.

Heterogeneous architecture for extending coverage

Scout uses a heterogeneous architecture with multiple aggressive transmission techniques to extend the coverage of each base station. These techniques include

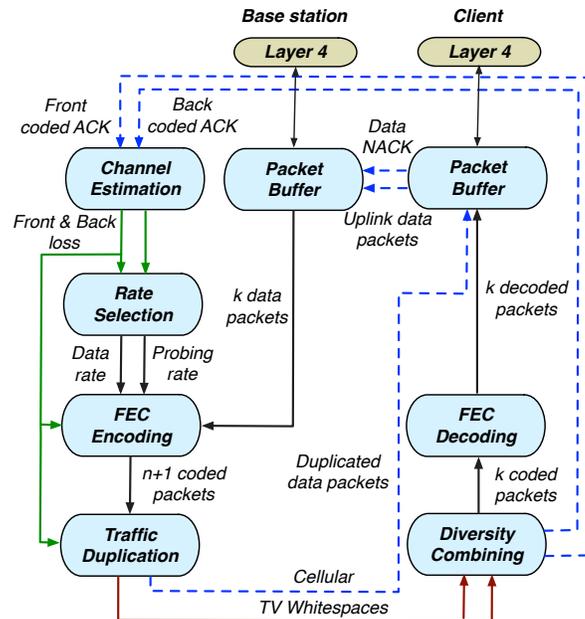


Figure 3.7: Operation flow of downlink communications

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 3.7 illustrates the operation flow of downlink communications.

To send downlink traffic, the base station first obtains channel quality estimates based on the feedback generated for both radios by the client (§ 3.3). It then chooses an appropriate link-layer data rate using the obtained channel estimates (§ 3.3). Based on the error performance of the chosen data rate, it constructs coded packets for each batch of data packets for forward error correction (§ 3.3), 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 (§ 3.3).

At the client (vehicular gateway), we leverage the dual-radio diversity by combining the coded packets from both radios (§ 3.3). 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 major signal processing functions is presented in Appendix B. We now explain each of these components in detail.

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 Appendix B.

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 3.5, 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 location coordinates reported by the GPS, since 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 3.2). 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 determined by the GPS reading that is updated by the client frequently (every second). We next describe how to use the obtained channel quality estimates from the scouting radio to adapt three major transmission decisions, i.e., rate adaptation, forward error correction, and intelligent traffic duplication.

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 (Appendix B). 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 § 3.3. 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.

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 Appendix B.

Inter-packet FEC: To circumvent the high retransmission inefficiency caused by a slow cellular uplink as mentioned in § 3.2, 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 § 3.3 that the above choice leads to significant performance gain.

We choose a type of erasure codes based on Vandermode encoding matrices [78] in our implementation, which incurs low computational latency (order of microseconds). The specific erasure codes provide an 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 § 3.3.

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 in Appendix B.

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 whitespaces 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.

Additional techniques to enhance link robustness

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 packet losses.

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 [52], 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.

Central controller for efficient packet forwarding

At the top tier, the central controller assigns a best base station to send downlink traffic to each client, while scheduling packet forwarding to enforce any given contention policy. It also relays uplink traffic from clients to the Internet. Figure 3.8 shows the controller's architecture, which is partitioned into a control plane and a data plane. The control plane performs base station assignment periodically (in every second), and outputs a routing table to the data plane. The data plane schedules downlink packets for simultaneous forwarding among all the contention domain. It relays packets to the corresponding base stations based on the routing table. Through the partition of the control and data planes and parallelization of the data plane across contention domains, the controller can achieve high efficiency and scalability in packet forwarding. We next describe the control plane and data plane respectively.

Control Plane

In the control plane, the router finder selects a base station for sending downlink packets to each client in real-time. It computes routes based on the conflict graph of base stations (possibly obtained through RF survey), the policy to coordinate transmission for each contention domain (optionally given by a network operator), and the link capacity reported by each base station for all

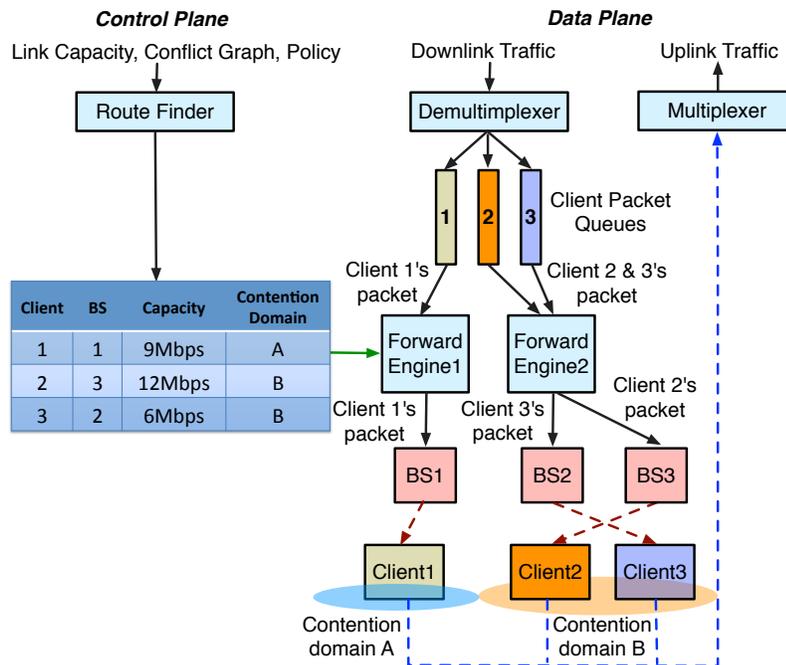


Figure 3.8: Architecture of the central controller.

its clients. The output is a routing table as illustrated in Figure 3.8. Each entry contains the selected base station, its link capacity, and contention domain for a client. The route finder leverages a real-time optimization algorithm for assigning base stations in every second. Such algorithm takes the link capacity estimate $\theta_{i,j}$ reported by each base station j for each client i , in which $i = 1, \dots, I$ and $j = 1, \dots, J$. It outputs a I -by- J matrix B for the assignment of base stations, such that $\beta_{i,j} = 1$ if a client i is assigned to a base station j and $\beta_{i,j} = 0$ otherwise.

The goal of the base station assignment task is to maximize the aggregate link capacity of all the clients. We note a popular greedy algorithm from existing systems [6, 36, 57, 58, 61] is inefficient for assigning base stations *from multiple contention domains*. To illustrate this, Table 3.1 shows the capacity estimates $\theta_{i,j}$ between 2 base stations and 2 clients. The greedy algorithm selects a base station with the maximum link capacity, and would thus assign both clients to

	Base station 1	Base station 2
Client 1	12Mbps	6Mbps
Client 2	12Mbps	11Mbps

Table 3.1: Example of link capacity estimates.

base station 1. Nevertheless, a better algorithm could assign the two clients to different base stations, almost doubling the aggregate network throughput.

To identify the optimal assignment, one may design a native approach to enumerate all the base station assignments, with the goal of maximizing $C = \sum_j \sum_i \beta_{i,j} \theta_{i,j} \rho_{i,j}$, under the constraint of $\sum_j \beta_{i,j} = 1$ for any i . The $\rho_{i,j}$ indicates the fraction of transmission time that base station j allocates to client i , which is set to 0 if i is not assigned to j . Thus, $\rho_{i,j}$ depends on the base station assignment $\beta_{i,j}$, and a given channel contention policy (as an input to the controller). Take the policies of equal-time, equal-throughput, and proportional throughput for example. The transmission time $\rho_{i,j}$ allocated by a base station j to each client i are $\frac{1}{\sum_i \beta_{i,j}}$, $\frac{\frac{1}{\theta_{i,j}}}{\sum_{i:\beta_{i,j}=1} \frac{1}{\theta_{i,j}}}$, and $\frac{\theta_{i,j}}{\sum_{i:\beta_{i,j}=1} \theta_{i,j}}$ respectively. The exhaustive search based algorithm would need to explore all the J^I base station assignments at an exponential complexity. Even for a small network of 6 base stations and 6 clients, such an algorithm would enumerate 46656 combinations, incurring a prohibitive delay for real-time implementation. Note that a conventional linear programming approach is not applicable to solving this optimization problem because the transmission time $\rho_{i,j}$ depends on the base station assignment $\beta_{i,j}$.

Base station assignment algorithm: To reduce the computational latency, we have developed an optimization technique that strategically reduces the search space through a L-Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [9]. We choose the L-BFGS-B extension of this algorithm that allows box constraints between 0 and 1 for solving the $(J - 1) * I$ assignment parameters $\{\beta_{i,j} : i = 1, \dots, I, j = 1, \dots, J - 1\}$. $\beta_{i,J}$ can then be calculated from $1 - \sum_{j:j < J} \beta_{i,j}$. Such an algorithm can significantly reduce the computational complexity from exponential J^I to polynomial $M^2 * (J - 1)I$ [23], while yielding a close-to-optimal solution as shown in § 3.5. Here M is the number of the past assignment

results used for the current iteration, which is set to be 5 in our implementation. After solving the assignment matrix B , the algorithm selects the base station as $b_i = \{j : \beta_{i,j} = 1\}$ for each client i , and its link capacity as $c_i = \{\theta_{i,j} : \beta_{i,j} = 1\}$. The selected base station b_i , its link capacity c_i , and contention domain are populated in a routing table, which will be used by the data plane for packet forwarding as described next.

Data Plane

As illustrated in Figure 3.8, the data plane consists of a few uplink multiplexers, downlink multiplexers, per-client packet queues, and forwarding engines (FEs) for individual contention domains. All of these modules can be distributed for scalability. The multiplexers forwards uplink traffic from clients to the Internet. In contrast, a few demultiplexers disentangle downlink packets from Internet, storing them into the corresponding packet queues based on the destined clients (Figure 3.8). These queues are then divided based on the contention domain of their clients as indicated in the routing table. For each contention domain, a forwarding engine (FE) schedules packets from all the associated queues, while relaying them to the assigned base stations. For example, the FE1 in Figure 3.8 forwards packets from client 1's queue in contention domain A, whereas FE2 forwards packets of client 2 and 3 in domain B. The packet forwarding is conducted in parallel among FEs, whose receiving base stations do not interfere with each other across contention domains. In contrast, each FE *serializes* forwarding packets from its queues, since only one base station can transmit within a contention domain. We next describe how a FE schedules packets for clients in a single contention domain.

Downlink traffic scheduling algorithm: The forwarding engine uses a variant of deficit round robin (DRR) scheduling [87] to forward packets for contending clients. To demonstrate its flexibility, we have implemented three contention policies, i.e., equal time, equal throughput, and proportional throughput. To enforce these policies, the FE maintains a *deficit counter* DC_i for each of the associated client i . Unlike the traditional DRR algorithm that uses this counter to track the maximum traffic size allowed to send [87], our

counter captures the maximum *transmission time* allocated to each client. This definition effectively captures both the traffic volume and link capacity. The FE iterates on a list of non-empty queues. It adds a fix mount of transmission time q_i , called *quantum*, to the deficit counter of the current client. It then estimates the transmission time t_i of the first packet in the client’s queue as $t_i = L/c_i$. L is the length of the packet and c_i is the estimated link capacity from the assigned base station. If t_i is less than the deficit counter DC_i , the FE forwards that packet to the base station and decrements DC_i by t_i . It then waits for t_i time to let the base station finish sending that packet, and continues to forward the next packet. This dequeue operation stops when either t_i is greater than DC_i or the queue becomes empty, which makes the FE to move on to the next active client.

The key of this algorithm lies in setting an appropriate quantum for each client. Specifically, q_i is calculated as $q_i = D \times f_i$, in which D is the time duration for sending all the clients’ packets in a scheduling round, called *round interval*, and f_i is the fraction of time allocated to each *active* client i . f_i is computed based on the link capacity estimate c_i and the given contention policy. Specifically, we compute f_i as $\frac{\frac{1}{c_i}}{\sum_{i=1}^N \frac{1}{c_i}}$ for the equal throughput policy, $\frac{c_i}{\sum_{i=1}^N c_i}$ for the proportional throughput policy, and finally, $\frac{1}{N}$ for the equal time policy. Here N indicates number of clients assigned to this base station. To determine the round interval D , we observe a clear tradeoff between the network throughput and accuracy of policy enforcement. A smaller D ensures a precise allocation of transmission time among clients, but causes larger overhead in synchronizing concurrent access to the packet queues among demultiplexers and FEs. The opposite holds for a larger D . In our implementation, we select D to be 10 microseconds, which strikes a good balance of this tradeoff as will be shown in § 3.5.

3.4 IMPLEMENTATION

We describe the hardware platform and the software architecture to implement *Scout*.



Figure 3.9: TV whitespace radios in an OpenWrt based router.

Hardware platform

We used broadband transceivers from the Doodle Labs Inc [20] for TV whitespace communications. Figure 3.9 shows these radios mounted in an OpenWrt based router [28] deployed at our base stations and clients. Each router can drive up to 4 radios through its Mini PCI interfaces. The radio is based on a frequency translator integrated with a 802.11 baseband chipset that is capable of transmitting at a data rate up to 54Mbps. Due to the fixed RF filtering, it can only operate at a pre-defined center frequency of 563MHz, 596MHz or 638MHz, and with a tunable bandwidth of 5, 10, 20MHz. The transmission power of these radios is up to 1W, which is configurable through the Ath5k driver [93].

Software architecture

Our software framework runs at the layer 3.5 based on encapsulation tunnels. It is implemented in an user-space program at the controller, base stations and clients respectively. All the programs are written in 15000 lines of C++ code. We also added about 50 lines of C code in the WiFi driver to control the transmission data rate of base stations. Figure 3.10 shows the software framework of *Scout* based on an example network setup of two base stations and a client.

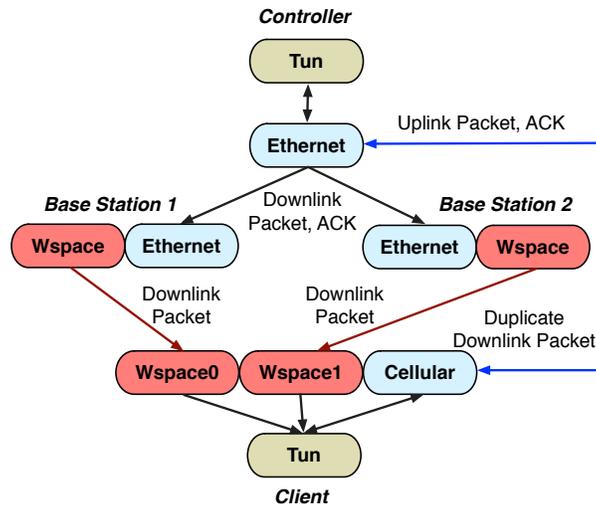


Figure 3.10: Software architecture of *Scout*.

Single-link abstraction: To let existing applications run over heterogeneous links, we expose a single-link abstraction using a Linux virtual network device called *TUN* [3]. All the application traffic is sent 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).

For downlink communications, the controller receives application traffic from the Tun interface, and forwards them through its Ethernet interface to the base stations. Each base station performs all the signal processing on the downlink packets as described in § 3.3, while sending the encoded packets through the whitespaces interface. It also sends the original data packets via the Ethernet interface upon detecting the outage of the whitespace link. The client receives the downlink packets from its whitespace interfaces, and occasionally from the cellular interface. It decapsulates these packets and performs various decoding steps as mentioned in § 3.3. After decoding, the client collates all the unique data packets into the packet buffer, passing those received in-order through the Tun interface to the application layer. It also sends the uplink traffic and ACK packets through its cellular interface to the controller. The controller relays uplink packets to the Internet, while forwarding ACKs to the

corresponding base stations¹. By leveraging the virtual networking interface, the end-user applications are agnostic to the underlying heterogeneous links and their dynamic changes during handoff.

Protocols for fast handoff: To minimize the overhead in switching routes during handoff, we carefully select protocols across the stack. At the MAC layer, we choose to use the ad-hoc mode for our network. It obviates various association handshakes that are otherwise needed in the infrastructure mode leading to tens of seconds of delay [6, 22]. At the network layer, we use UDP tunnels to make all the connections stateless, while performing both loss recovery and flow control at the layer 3.5.

Broadcast-based downlink communication: We configure a 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. 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 link². 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.

Multiple sequence number spaces: To support in-order and reliably delivery, *Scout* uses three sets of sequence numbers to index downlink packets destined to each client. The *end-to-end sequence numbers* are included by the controller for clients to reorder packets sent from multiple base stations. The *data sequence numbers* are added by each base station for link-layer retransmission.

¹The decision of forwarding cellular traffic through the controller is because only its Ethernet interface has a public IP address in our implementation, and traffic destined to a private IP can be sometimes blocked by cellular networks.

²If 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.



Figure 3.11: Two example road segments in experiments with a single base station.

The *raw sequence numbers* are used by the base station to index coded packets for estimating packet loss rates. A client sends NACK packets for each sequence number space to the controller in every 50 milliseconds.

3.5 EXPERIMENTAL RESULTS

In this section, we first evaluate the performance of *Scout* in a single base station setup to quantify the efficacy of its link layer techniques guided by the scouting radio. We then evaluate *Scout* with multiple base stations to understand its benefits for base station coordination.

Single base station setup

We evaluate the performance gain of *Scout* by exploiting the scouting feedback to optimize multiple protocol decisions in a single base station scenario. These techniques include rate adaptation, inter-packet FEC, and intelligent traffic duplication implemented in our prototype system as described in § 3.3. For the experiments, we have driven about 500 miles in the neighborhood of the whitespaces base station across various road segments. We present results primarily from two road segments as shown in Figure 3.11. Table 3.2 summarizes our main results.

Experiment Setup	Compared Algorithms	Scout Gains
Overall Performance		
Whitespace + cellular, road A	<i>Scout</i> v.s. A1 – A4	23 – 120× over single radio, 3 – 8× over dual radios
WiFi + cellular, road A	<i>Scout</i> v.s. MIMO	Scout(2-SISO-Radios): 4×, Scout(2-MIMO-Radios): 7×
Whitespace + cellular, road A (video streaming)	<i>Scout</i> v.s. A4	90% reduction in buffering delay
Whitespace + cellular, road B	<i>Scout</i> v.s. A3 – A4	9× at 75th quartile
Microbenchmarks		
Whitespace + cellular, road A	ScoutRate v.s. RRAA and Minstrel	3.6× and 1.3× (with diversity)
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
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
WiFi + cellular, indoor	<i>Scout</i> v.s. A3 – A4	0.8× (under-perform)

Table 3.2: Summary of results in a single base station setup.

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.

Compared systems: We use RRAA and Minstrel as representative algorithms that use packet error rates at the receiver to select PHY rates. Table 3.3 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 [52]) 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 road segments in all the experiments unless mentioned otherwise. The two routes in Figure 3.11 have very different coverage characteristics — *segment A* is part of one major 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).

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

Table 3.3: Alternative systems to be compared against *Scout*.

Overall performance

We evaluate the overall downlink 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.

TCP throughput (road segment A): 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.

The first panel of Figure 3.12 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.

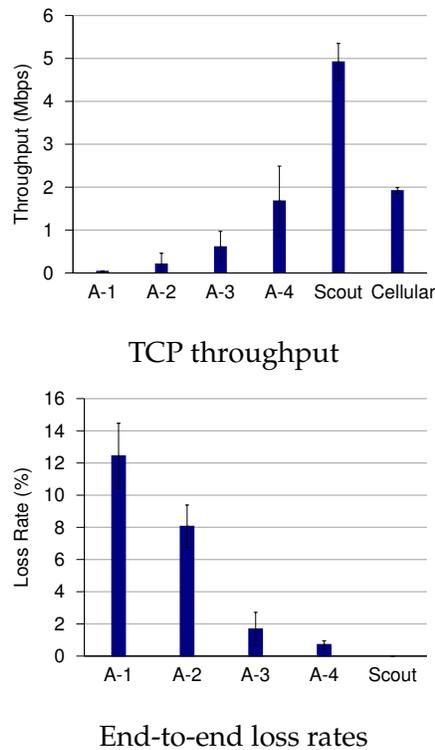


Figure 3.12: Average downlink TCP performance on road segment A. The error bar shows standard deviation.

The second panel of Figure 3.12 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. This in turn enables *Scout* to fully utilize the available link capacity leading to the dramatic improvement in TCP throughput.

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 [97] 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

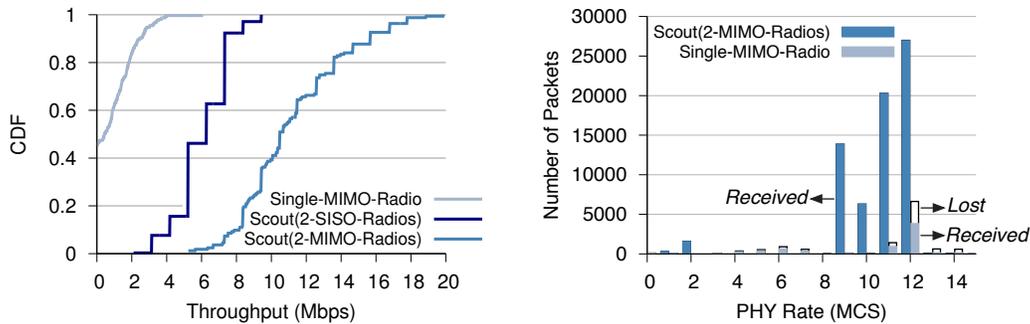


Figure 3.13: TCP performance of *Scout* and MIMO systems on road segment A. (a) CDF of downlink TCP throughput. (b) Distribution of end-to-end packets received and lost at different PHY rates.

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.

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 systems³, because the dual radio chains in WDR are not sufficiently synchronized for 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.

³This is the only outdoor experiment not done in TV whitespaces.

The first panel of Figure 3.13 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.

To understand the reason for the performance difference between the two MIMO systems, the second panel of Figure 3.13 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 $5\times$ larger than that of Single-MIMO-Radio.

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.

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. Figure 3.14 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,

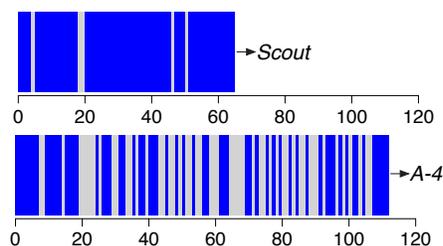
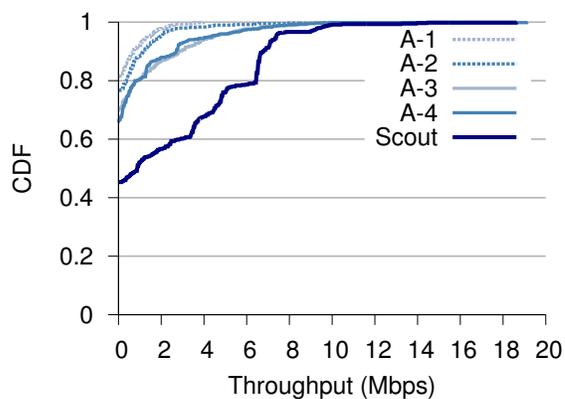


Figure 3.14: Time series plot of a 60s video played in different systems along road segment A. The dark color shows the playing time and the light color shows the buffering delay.

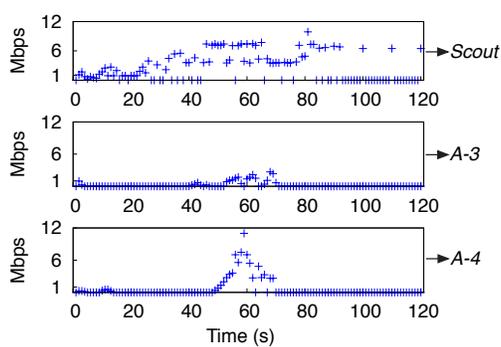
whereas the same video played in *Scout* only stopped for a few times (4), with each lasting for less than 3 seconds. This leads to an overall buffering delay to be 90% lower than that of A-4, indicating better user experience in *Scout*.

TCP throughput (road segment B): 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. Figure 3.15(a) presents the CDF of TCP throughputs averaged over 1 second bins from all the drives of each system. We observe that *Scout* achieved $9\times$ 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.

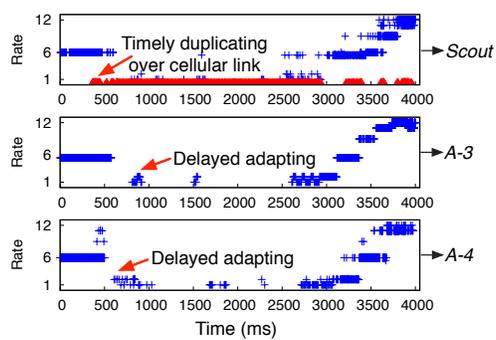
Figure 3.15(b) presents the throughput measurements over time in one of 5 drives for each system. We observe that *Scout* strived to maintain a vehicular connectivity for more than 75% of the time, whereas A-3 and A-4 were able to support connectivity for about 15% of the time. During the first 20 seconds 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



(a) CDF of TCP throughput



(b) Time series plot of throughputs



(c) Snapshot of received packets

Figure 3.15: Downlink TCP performance of different systems along road segment B.

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 whitespace 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 3.15(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.

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.

Rate adaptation: To understand the contribution of the scouting based channel estimation to rate selection, we evaluated the rate adaptation algorithm used

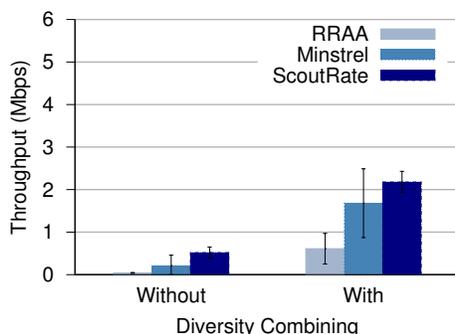


Figure 3.16: Performance improvement of rate adaptation (microbenchmark).

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. Figure 3.16 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 diversity combining can effectively improve the performance of RRAA, Minstrel and ScoutRate by $15\times$, $8\times$ and $4\times$.

Inter-packet Forward Error Correction: 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 § 3.3.

Figure 3.17(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.

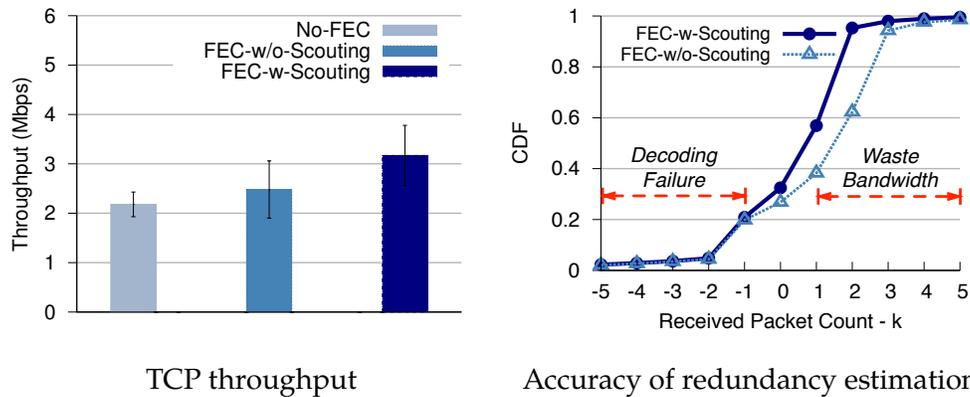


Figure 3.17: Performance improvement of inter-packet FEC (microbenchmark).

To understand the advantage of the scouting based channel estimation, we compare 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 (§ 3.3). 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 3.17(b) shows the CDF of this metric for different batches. We note the gain of FEC-w-Scouting mainly came from *fewer over-estimations* of redundancy, which allowed better utilization of channel capacity.

Intelligent traffic duplication over cellular links: 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.

Figure 3.18 shows that the intelligent traffic duplication in *Scout* improved the overall TCP throughput by 55% over No-DUP, because whitespace links (like

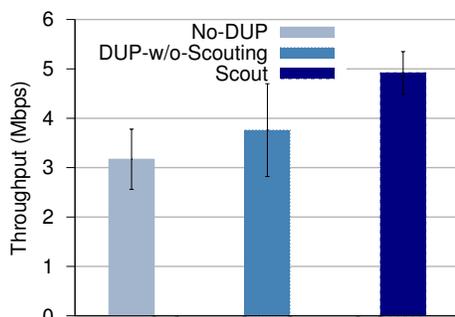


Figure 3.18: Performance improvement of intelligent traffic duplication over cellular path when needed (microbenchmark). No-DUP is the same as FEC-w-Scouting.

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.

Scout v.s. whitespace-only networks

We now evaluate the efficacy of scouting based channel estimation. We compare Scout based on a heterogeneous architecture with alternative systems (in Table 3.3) that use a conventional, whitespace-only architecture as discussed in § 3.1. To eliminate the effect of channel estimation, we have restricted the base station and client to use 100 mW for all the systems, which leads to about a 4× reduction in the transmission range compared to the 4 W power. We then measured the average downlink TCP throughput when driving in a fraction of road segment A for about 200m (due to the restricted base station coverage). Figure 3.19 shows the average downlink TCP throughput of each system measured from multiple drives.

We first observe that *Scout* achieves a similar throughput compared to A3 and A4 that use symmetric whitespace links with negligible feedback delay.

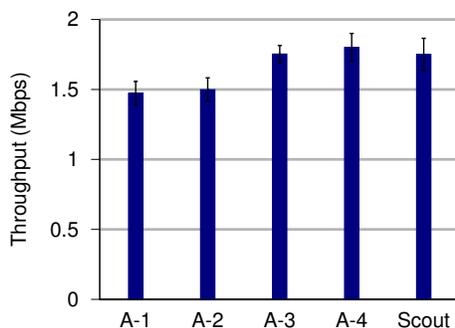


Figure 3.19: Downlink TCP performance of Scout and alternative systems using a symmetric, whitespace-only architecture.

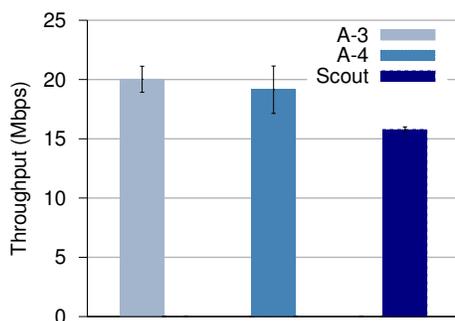


Figure 3.20: Performance of *Scout* in static scenarios.

This demonstrates that our scouting radio can mask most of the effect on channel estimation imposed by high feedback delay in the cellular path. *Scout* also achieves 19% and 17% higher throughput than A1 and A2, owing to the dual-radio combining technique. We also note the throughput of *Scout* based on a 100 mW transmission power is only a third of that with a 4 W power (Figure 3.12), which demonstrates the coverage advantage of our heterogeneous architecture.

Scout in static scenarios

As discussed in § 3.3, when a client is static, our base station loses its advantage in channel estimation by reverting to the delayed feedback of the rear radio. To study the performance of *Scout* in these static environments, we measured the

Setup	Algorithms	Scout Gains
Overall Performance		
Different channels; equal-throughput	<i>Scout</i> , Greedy, Round-robin, Duplication	73% over Greedy and Duplication, 1.9 \times over Round-robin
Different channels; equal-time & proportional-throughput	<i>Scout</i> , Greedy, Round-robin, Duplication	90 – 96% over Greedy and Duplication, 96 – 374% over Round-robin
Different, hybrid, same channels; equal-time	<i>Scout</i> , Greedy	90 – 104% over Greedy in hybrid and different channels
Microbenchmarks		
Different channels; all policies	<i>Scout</i> , Greedy	2 \times over Greedy in average throughput
Simulation		
Different channels; all policies; varying number of bs and clients	<i>Scout</i> , Greedy, Exhaustive search	400 \times lower than Exhaustive in latency
Different channels; all policies; 5 and 15 bs with varying number of clients	<i>Scout</i> , Greedy, Exhaustive search	11 – 77% gain over Greedy, 11% – 20% loss to Exhaustive

Table 3.4: Summary of results in a multi-base station setup. Performance gain is for median throughput unless mentioned otherwise.

downlink TCP throughput in a 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. Figure 3.20 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 detect such cases using GPS updates from clients.

Multiple base station setup

We now evaluate the performance of *Scout* using three base stations and three vehicular clients driving around our testbed area simultaneously for about 200 miles. We start by presenting the overall performance of *Scout*. We then benchmark its various scheduling functions. Finally, we conduct a trace-driven simulation of *Scout* at various network scales. Table 3.4 summarizes our main experiment results in the multi-base station setup.

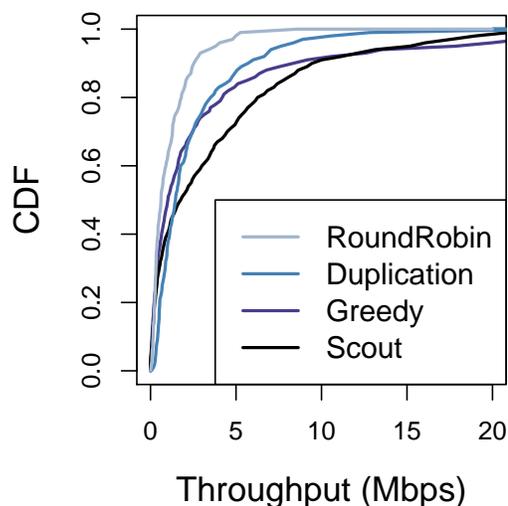


Figure 3.22: Downlink performance under the equal throughput policy with base stations operating in different channels: CDF of aggregate downlink throughput.

stations and three vehicles driving in different road segments simultaneously. Figure 3.21 shows the routes taken by these vehicular clients ranging from 0.5 to 1 mile, and the location of each base station. While driving, we start parallel UDP downloads from the controller to individual clients, and record their downlink throughput averaged in 1 second intervals. We then align their throughput measurements based on the time stamps to compute the aggregate network throughputs. All the results are collected from 3 drives for each algorithm.

Performance under equal-throughput policy: We first evaluate the downlink performance of all the systems under the equal-throughput policy. Each base station is configured to transmit in a *non-overlapping* channel for maximizing aggregate throughput. Figure 3.22 shows the CDF of the aggregate throughput for each system. We observe that *Scout* achieves 73% median gain and 78% at the 75th quartile over *Greedy*, owing to its optimization algorithm that efficiently leverages simultaneous transmission from multiple base stations⁴. *Greedy*

⁴The relatively lower end-to-end throughput is due to a ≤ 18 Mbps PHY rate often suitable for vehicular clients, coupled with various link-layer overhead such as carrier sensing and our own FEC.

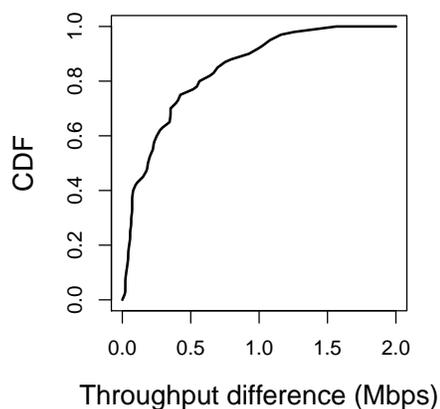


Figure 3.23: Downlink performance under the equal throughput policy with base stations operating in different channels: CDF of maximal throughput differences among contending clients.

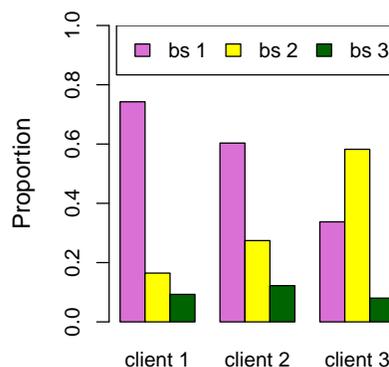


Figure 3.24: Packet distribution among base stations in a single drive under the equal-throughput policy.

outperforms Round-robin by 65% in the median throughput because it assigns base stations in a throughput-aware fashion. Interestingly, Duplication has a similar performance to Greedy especially in worse channel condition. This might be because Duplication utilizes all the base stations for redundancy, whereas Greedy leaves some of them idle.

We next quantify the accuracy of *Scout* in enforcing any given contention policy. Figure 3.23 shows the CDF of maximum throughput difference among contending clients for *Scout*. This result is based on 79% throughput measurements with at least 2 clients falling in a single contention domain. We observe a median difference of 0.2Mbps and a 90-quartile of 0.9Mbps, demonstrating its ability to faithfully execute the equal throughput policy. We have observed a similar performance of *Scout* in enforcing other policies, and omit the result for the sake of brevity.

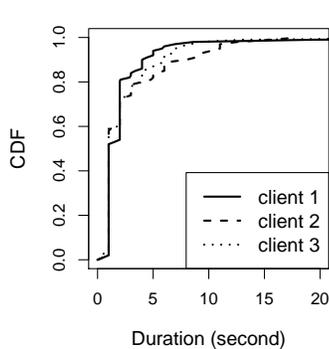


Figure 3.25: CDF of base station switching duration in a single drive under the equal-throughput policy.

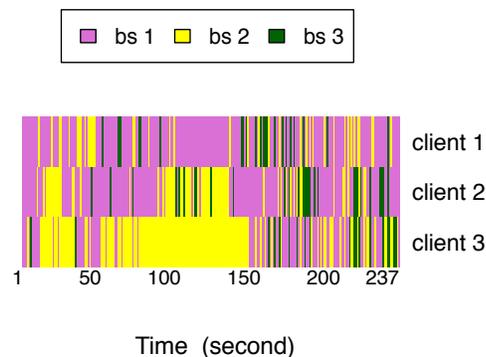


Figure 3.26: The id of base stations serving each client defined by *Scout*, in a single drive under the equal-throughput policy.

To better understand the performance of *Scout*, we compute various statistics about its base station assignments in a single drive. Figure 3.24 shows the distribution of packets sent by individual base stations. We can see the base station 3 has sent least number of packets because it is deployed in a building with a relatively small coverage. In addition, the client 1 and 2 have a similar distribution of packets sent from base station 1 and 2, which is opposite to that of client 3. This is because client 1 and 2 were driving in the same direction (albeit for different distances), which aligns well with the coverage of base station 1 based on our testbed measurements. In contrast, client 3 is driving in the opposite direction primarily covered by base station 2. These meaningful results demonstrate that *Scout* is accurate in assigning appropriate base stations for each client.

To validate the need for a fast assignment algorithm, Figure 3.25 shows the duration of switching base station assignments for each client. We observe that about 80% of such durations are quite short – within 5 seconds, thus necessitating a real-time assignment algorithm. Nevertheless, only about 5% of durations falling below 2 seconds, which suggest our assignment algorithm running at a 1 second interval to be adaptive enough to channel variation. Figure 3.26 shows the ids of base stations serving each client. We note frequent

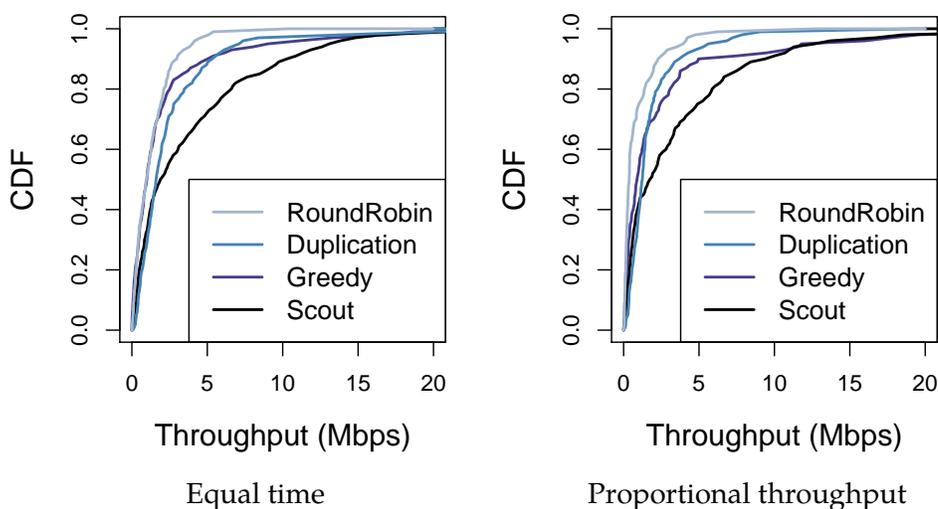


Figure 3.27: Downlink throughput under two other policies with base stations operating in different channels.

changes of base station assignments. Despite that, each client has maintained a continuous traffic flow owing to the seamless handoff feature of *Scout*.

Performance under two other policies: We now evaluate the downlink performance of these systems under two other policies, i.e., equal-time and proportional-throughput. Figure 3.27 shows the CDF of the aggregate network throughput for each policy. Similar to equal-throughput policy, *Scout* outperforms Greedy by 90% and 96% in median throughput under the equal-time and proportional-throughput policy, respectively. It further outperforms Round-robin by 96% and 374% for each policy.

Different channel configurations: We now evaluate the performance of *Scout* under a varying number of unique channels used by base stations as follows. The 3-channel setup configures all the base stations to use a different, non-overlapping channel, which is the same setup in the previous evaluation. The 2-channel setup lets two of the base stations to operate in a same channel (centered at 602MHz), and the third one in another channel (638MHz). The 1-channel case makes all three base stations to use a single channel (602MHz). In this setup, *Scout degenerates* to Greedy since there is no channel diversity

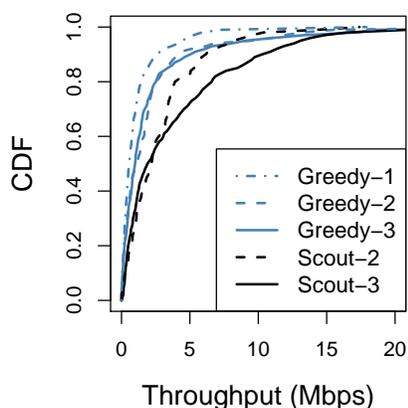


Figure 3.28: Downlink throughput under different channel settings at base stations. The number appended to each algorithm marks how many unique channels are used by the base stations.

to exploit. Figure 3.28 shows the CDF of aggregate throughput of *Scout* and Greedy under the equal-time policy, with the number of unique channels appended to each algorithm. We observe a similar performance achieved by Greedy across different channel settings. This is because Greedy selects base stations solely on their link capacity without considering their contention effect (§ 3.3). In contrast, *Scout* achieves a higher throughput with additional channel diversity, outperforming Greedy by 90% and 104% in the cases of two channels and three channels respectively. This demonstrates its advantage in scheduling concurrent transmissions across contention domains.

Controller microbenchmarks

We first quantify the dual-efficacy of our controller for the base station assignment and contention policy enforcement. We then validate our selection of the round interval parameter for packet scheduling. All the results were measured in a lab environment using the same heterogeneous architecture.

Performance of traffic scheduling: We compare *Scout* and Greedy for scheduling downlink traffic from 2 base stations to 3 clients under different channel contention policies. Each base station is configured to transmit in a different

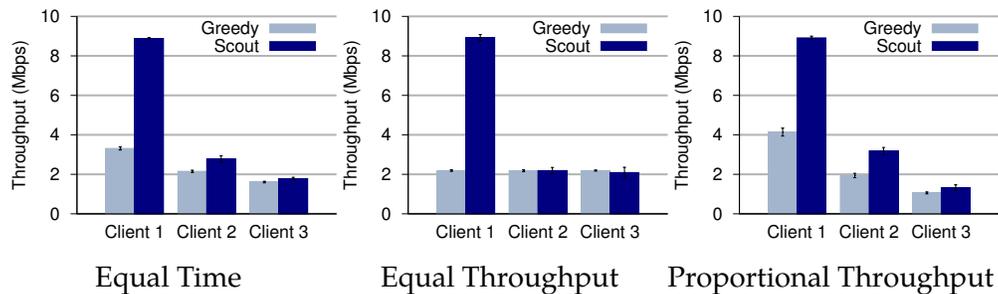


Figure 3.29: Average downlink throughput measured by individual clients under different channel contention policies.

	Client 1	Client 2	Client 3
Base station 1	1Mbps	9Mbps	6Mbps
Base station 2	18Mbps	12Mbps	9Mbps

Table 3.5: PHY-layer data rates used by base stations to send downlink packets to different clients.

channel, and at a fixed data rate for each client as shown in Table 3.5. The first panel of Figure 3.29 shows the downlink TCP throughput of each client under the equal-time policy. We can see that *Scout* achieves a $2\times$ aggregate throughput compared to Greedy. This is because Greedy assigns all the clients to base station 2 due to its higher link capacity, incurring wasted transmission opportunities at base station 1. In contrast, *Scout* fully exploits the concurrent transmission opportunity of the two base stations, by assigning client 1 to base station 2 and the other two clients to base station 1. We also observe that those clients in a single contention domain has a throughput *proportional* to their link capacity. This suggests that each contending client has received a similar amount of transmission time, demonstrating the efficacy of *Scout* for executing the equal-time policy. For the other two policies, The second and third panel of Figure 3.29 show that *Scout* regulates the throughput distribution of clients equally well to match each policy. We have experimented with various other setups and link configurations, and observed a similar traffic scheduling performance of *Scout*.

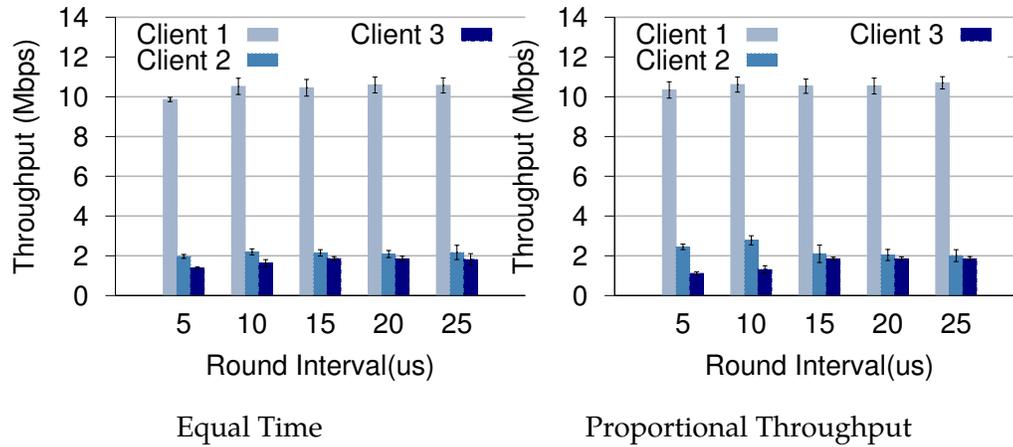


Figure 3.30: Average downlink throughput of individual clients at different scheduling round intervals in *Scout*.

Impact of round interval: We use the same setup in the previous benchmark experiment, but vary the round interval in our DRR algorithm to study its tradeoff between throughput efficiency and policy enforcement. The first panel of Figure 3.30 shows the average throughput of individual clients in *Scout* under the equal-time policy. Based on our previous setup, we expect a 3:2 throughput ratio between client 2 and client 3 as both of them are assigned to base station 1. Nevertheless, such a ratio can only be achieved accurately at a small round interval of ≤ 10 microseconds. Above this threshold, the throughput ratio tends to become 1 since our whitespace radios use a 802.11 MAC protocol that by default leads to a similar throughput among clients [100]. On the other hand, we note a slight throughput decrease at a lower interval, with a noticeable degradation of 7 – 17% at the 5 microsecond interval. This is because such a small interval leads to frequent concurrent access by the packet demultiplexers and forwarding engines, which brings significant synchronization overhead (locking) as discussed in § 3.3. The second panel of Figure 3.30 shows a similar trend under the proportional-throughput policy. Hence, we select the round interval to be 10 microseconds for accurate policy enforcement, yet low throughput overhead.

Trace-driven simulation

To evaluate the performance of *Scout* at various network scales, we conduct trace-driven simulation to quantify its base station assignment latency and aggregate network capacity. We compare *Scout* with Greedy and the exhaustive search algorithm as described in § 3.3. The traces consist of packet loss measurements that are collected from the vehicles in the outdoor experiments (§ 3.5). We segment the loss rate measurements into 50 meter road segments based on their GPS reading, and synthesize them in a random order. We then replay this synthetic trace to determine the current loss rate between a base station and a client. To emulate the feedback delay in cellular uplink, we use the loss rate collected at 100ms ago as the feedback to these systems, while computing the downlink throughput based on the current loss rate and their transmission decisions. Using this strategy, we generate traces for 15 candidate base stations and 20 candidate clients. To simulate different network sizes, we randomly pick I clients and J base stations for each simulation, with I varying from 2 to 20 and J from 2 to 15. Each base station is assumed to use a different channel given the abundant spectral resource in whitespaces. There are 266 simulations performed in total.

Latency of base station assignment: We start by analyzing the latency of base station assignment by different algorithms. Figure 3.31 shows the number of possible assignments in cases with a given number of base stations and clients, all of which need to be explored by the exhaustive search. The numbers are in a log10 scale, e.g., 4 indicates 10000. Figure 3.32 shows the latency of the three algorithms for choosing an assignment decision from all possible combinations, as a function of total number of combinations to choose from. All three contention policies are evaluated. For a given algorithm, contention policy and number of base station and client, the latency is averaged over 1 second intervals. The measurement is based on a desktop computer with a 8-core CPU and 32GB RAM, and only conducted for up to 10^6 combinations due to the prohibitive latency of exhaustive search. This corresponds to a moderate network size of 7 base stations and 7 clients as shown in the top x axis, and several other setups can be found in Figure 3.31. We observe that the

1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	2 bs
1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	10	3 bs
1	2	2	3	4	4	5	5	6	7	7	8	8	9	10	10	11	4 bs
1	2	3	3	4	5	6	6	7	8	8	9	10	10	11	12	13	5 bs
2	2	3	4	5	5	6	7	8	9	9	10	11	12	12	13	14	6 bs
2	3	3	4	5	6	7	8	8	9	10	10	11	12	13	14	15	7 bs
2	3	4	5	5	6	7	8	9	10	11	12	13	14	14	15	16	8 bs
2	3	4	5	6	7	8	9	10	10	11	12	13	14	15	16	17	9 bs
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	10 bs
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	11 bs
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	12 bs
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	13 bs
2	3	4	6	7	8	9	10	11	12	13	14	15	16	17	18	19	14 bs
2	3	5	6	7	8	9	10	11	13	14	15	16	17	18	19	21	15 bs
2	4	5	6	7	8	9	11	12	13	14	15	16	18	19	20	21	16 bs
2	4	5	6	7	8	9	11	12	13	14	15	16	18	19	20	21	17 bs
2	4	5	6	7	8	9	11	12	13	14	15	16	18	19	20	21	18 bs
2	4	5	6	7	8	9	11	12	13	14	15	16	18	19	20	21	19 bs
2	4	5	6	7	8	9	11	12	13	14	15	16	18	19	20	21	20 bs

Figure 3.31: Number of possible combinations in log10 scale. The cases measured in Figure 3.32 are marked in blue.

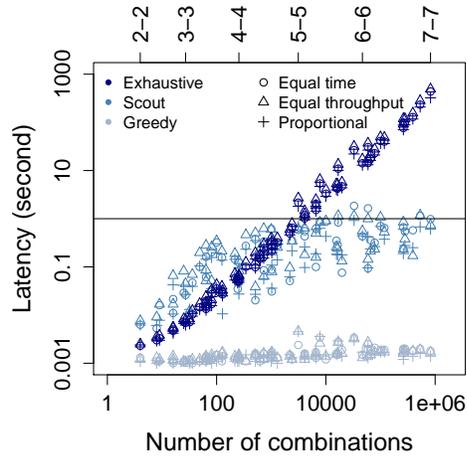


Figure 3.32: Latency for a varying number of combinations. The top x axis marks example network setups based on Figure 3.31.

latency of the exhaustive search increases exponentially. For 10^6 combinations, it reaches about 400 seconds rendering the system completely unresponsive. In contrast, *Scout* achieves a sub-second latency by filtering out a large number of unpromising assignments. Such a latency is low enough for the 1 second period used by *Scout* to assign base stations (§ 3.3). Greedy has the lowest latency of about 1 milliseconds due to its linear algorithm by selecting a base station with the maximum capacity. Nevertheless, it incurs a significant throughput penalty as will be shown next.

Aggregate network throughput: We simulate the aggregate network throughput achieved by different algorithms in a network of 5 base stations and 2 to 5 clients. Figure 3.33 shows the throughput results averaged over 1 second intervals. We observe that the throughput increases with more clients as expected. *Scout* achieves a 20% – 77% gain over Greedy by efficiently exploiting the concurrent transmissions among base stations. Furthermore, it only underperforms the exhaustive search by 11% - 20%, but achieving orders of magnitude lower latency. Figure 3.34 shows the throughput results at a larger network of

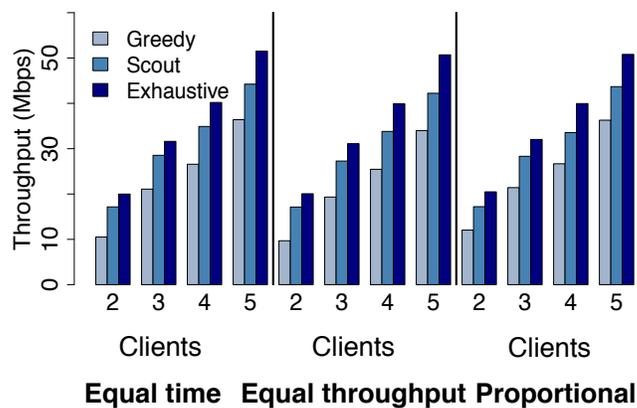


Figure 3.33: Aggregate throughput achieved by 5 base stations and a varying number of clients.

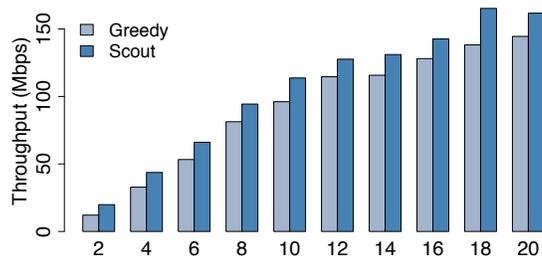


Figure 3.34: Aggregate throughput achieved by 15 base stations under the equal-throughput policy.

15 base stations and 2 to 20 clients. We omit the result of the exhaustive search due to its substantial latency. We observe that *Scout* achieves a similar gain of 11% – 64% over Greedy, demonstrating its advantage in base station assignment at various network scales.

3.6 ISSUES AND DISCUSSION

We outline several possible enhancements of *Scout* in efficiency, scalability and applicability.

- **Leveraging whitespace uplink for opportunistic communications.** *Scout* currently sends all the uplink traffic over the cellular path. One possible enhancement is to allow opportunistic communication in whitespaces uplink when a vehicle is within its transmission range to a whitespace base station. Such a design brings additional challenges to coordinate channel contention among base stations and clients, since conventional carrier sensing is hard to detect transmission from weak clients. Explicit synchronization protocols might be employed to coordinate such contention, perhaps using TDMA protocols similar to those in long-distance WiFi networks [67, 73]. In addition, dynamically switching between whitespace links and cellular technologies lead to complicate effects on TCP protocols. When switching from a fast whitespace uplink to a slow cellular path, spurious TCP timeout might occur due to the abrupt increase in link latency, leading to redundant retransmission and unnecessary congestion avoidance [10]. In contrast, when switching from the slow cellular link to fast whitespaces, the frequent arrival of Acks causes bursty transmission at the sender leading to packet losses [5]. To address these TCP anomalies, some Ack pacing techniques [5, 10, 43, 60] might be employed at the link layer to regulate the Ack arrival time at the base station to smooth out RTT variation. Additional TCP enhancements such as Ack congestion control [5] and Ack regulation [10] might be used to mitigate source burstiness by manipulating the frequency of Ack generation.
- **Estimating background traffic.** Our current design does not take account the impact of background interference generated by other whitespace networks. In future work, we intend to estimate its impact on the link capacity estimation, which can be achieved through a passive channel measurement technique based on an additional monitoring radio as proposed in WhiteFi [4], or an in-band probing technique based on the arriving interval of downlink packets similar to DenseAP [57]. Intelligent sensor placement strategies might be further used to reduce the hardware cost and energy consumption, by activating a subset of monitoring devices based on their location proximity and operation environment. Efficient

scheduling techniques are also needed to coordinate these sensors based on their channel condition information and the power budget. In addition, accurately translating the impact of the measured channel noise into link capacity awaits further exploration.

- **Collecting channel contention information.** *Scout* relies on the contention domain of base stations to perform various scheduling tasks. Collecting this channel conflict data is greatly simplified in our heterogeneous network, since only a few static base stations are transmitting at a fixed power. Hence, it is quite possible to use a standard RF survey employed in indoor WiFi networks for this. Of course, one may leverage a dynamic channel conflict measurement technique in prior work [44, 76, 77] to improve accuracy.

3.7 SUMMARY OF SCOUT

In this chapter, we have presented *Scout*, a heterogeneous network 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 network coverage compared to a symmetric whitespace-only architecture, thereby reducing the deployment cost.

We have also introduced the notion and demonstrate the benefits of a scouting radio to address the feedback delay 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×).

We further developed a central controller to coordinate multiple whitespace base stations to enhance the network coverage and throughput. The controller leverages an efficient base station selection algorithm and a flexible packet scheduling framework to improve the overall network capacity (73 – 374% gain),

enforce given channel contention policies (0.2Mbps median throughput error), and support seamless client handoff. We believe that the proposed network architecture and our specific techniques, can have a broader application to dynamic spectrum access networks beyond TV whitespaces.

4 AN EDGE COMPUTING ARCHITECTURE TO IMPROVE BANDWIDTH EFFICIENCY

4.1 MOTIVATION

In previous chapters, we presented network architectures and communication protocols to enhance the coverage of whitespace networks. We now focus on techniques to improve the bandwidth utilization of whitespace networks. Such a solution can be particularly valuable for long-distance networks suffering from sporadic losses and low data rates, or some urban deployments with scarce spectrum resource. We have explored the solution space to support a specific class of bandwidth-hungry and delay-sensitive applications — intelligent video surveillance. Video surveillance has become pervasive with many applications such as retail store analytics, corporate security, and traffic monitoring. While cities such as London and Beijing have nearly a million cameras deployed today, the vast majority of them are connected to wired network infrastructures, which leads to substantial deployment cost and limited coverage.

To overcome the limitations of wired connection, we intend to design a wireless video surveillance system that leverages TV whitespaces (and other unlicensed spectrum) to upload real-time video feeds to the cloud for intelligent vision analysis — a function commonly provided by wired surveillance systems today (e.g., DropCam [21]). A key challenge in realizing such a system comes from the limited wireless capacity, which is far from sufficient to support simultaneous uploading of high-definition video feeds from distributed cameras. Moreover, wireless networks need to overcome channel losses with error correction coding and retransmission techniques, both introducing protocol overhead that erodes the effective network capacity. When the wireless capacity lags behind the traffic demand of surveillance cameras, a significant backlog of video traffic coupled with frame losses can occur, largely degrading the responsiveness and accuracy of surveillance applications.

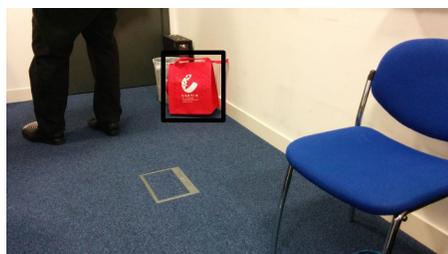
We present Vigil, a real-time wireless video surveillance system that leverages the edge computing technology [82] to support many cameras over a

large area. Vigil starts its processing flow with an user query, such as locating people or objects with certain features. Such a query is distributed among edge compute nodes (ECNs) that are collocated with camera sensors. Each ECN uses some light-weight vision algorithms such as face detection, to process the video feed locally. It produces a stream of utility values for individual video frames, which are uploaded to a cloud-based controller over whitespaces. Upon receiving these analytics, the controller only requests those relevant video frames from ECNs for more sophisticated analysis (e.g., face recognition). Hence, by pruning irrelevant video frames at the network edge, Vigil can conserve a substantial amount of wireless capacity, which is further scavenged to provide WiFi access to recoup the deployment cost.

Vigil leverages a novel frame scheduling technique to further maximize the bandwidth utilization for video surveillance applications. Such algorithm prioritizes the uploading of video frames based on their significance, thereby maximizing their utility for video surveillance under varying wireless capacity limits. To quantify the relative importance of video frames, we propose new metric called *ops* (objects per second), which can capture both the utility of a video frame (in number of objects), as well as its bandwidth cost for uploading such frame to the cloud. Our algorithm then greedily prioritizes the video frames with a higher *ops* to improve the wireless bandwidth efficiency.

We also introduce a camera collaboration technique to deal with the inherent inaccuracy of vision algorithms. These algorithms are often sensitive to various environmental factors such as capturing angle, lighting, and camera focus. Our proposed technique enhances their robustness by leveraging multiple cameras to monitor a common area simultaneously. It then uploads a minimum set of frames captured by these cameras to comprehensively capture the *unique* objects specific to an user's query.

We have deployed Vigil at three sites in two countries under vastly different conditions. Experimental results show that Vigil can enable a $5 - 200 \times$ larger surveillance coverage compared to a conventional approach that streams all the camera feed over the whitespace network. For a fixed region of coverage and bandwidth, Vigil outperforms the baseline systems by delivering up to 25% more query-specific objects.



Object of interest.



Line of people.

Figure 4.1: The two Vigil use-cases targeted in this paper.

4.2 VIGIL'S USE CASES

In this section, we motivate Vigil by describing various vision application domains it can possibly support in security, public safety, and business management.

- **Security and counter-terrorism.** Across our cities, CCTV cameras are installed in underground transport networks, on buses, and in airports and other public areas. We envisage online, real-time processing of the wireless video feed so that law enforcement and counter-terrorism can track public threats in real-time. For example, in the event of multiple co-ordinated attacks on public transport, the video surveillance network can pick out the face of one perpetrator, scan the database of cloud-stored video for other people the perpetrator was spotted with, and then search for those associated persons in real-time as the attack progresses, directing law-enforcement to the locations of the perpetrator's accomplices for intervention.
- **Locating people or objects of interest.** In many situations, people are interested in locating objects or people of interest, such as an "Amber Alert" in the United States, or an unattended bag of a certain color (the first panel of Figure 4.1). An airport might choose to continuously run a query on CCTV footage looking for bags that are not held by any person

nearby, flagging up unattended baggage to airport authorities. Traffic monitoring systems use vision-based algorithms to detect and count the number of cars on the highways [14, 70].

- **Customer queue analytics.** In places where customers line up for service, such as coffee shops, supermarkets, or amusement parks, shopkeepers might have an interest in knowing numbers of people waiting in line and the dynamics thereof over the course of a day (the second panel of Figure 4.1). Cameras are used to track line length, but face or body counting is challenging, as people strike different poses and turn at different angles to the camera. Consequently, a better design is to deploy an array of cameras surrounding the queuing area. The system then fuses their data together to form a more accurate count of the people in line.

4.3 VIGIL DESIGN

Vigil proposes a novel architecture that leverages the computing elements at the edge of the network to minimize bandwidth consumption of a wireless video surveillance network without sacrificing surveillance accuracy. Vigil consists of the two major components shown in 4.2. The *controller* is located in the cloud and receives users' queries, coordinating all the other parts of the system to answer the query. An *edge compute node* (ECN) is a small computing platform (*e.g.*, a laptop or embedded system) that is attached to a camera to bring cloud resources close to the edge of the network.^{1 2} Each ECN receives the video feed from its connected camera, and executes the first, stateless, stages of computation such as face detection or object recognition. It then periodically uploads analytic results to the controller. The ECNs also perform video compression, indexing and maintaining a short-term store of the video frames they capture. ECNs connect to the controller via wireless links operating over TV whitespace or WiFi bands. The controller runs a frame scheduling

¹Note that we use the terms ECN and camera interchangeably in the remainder of the paper.

²We discuss the case of connecting multiple cameras to a single ECN in § 4.6.

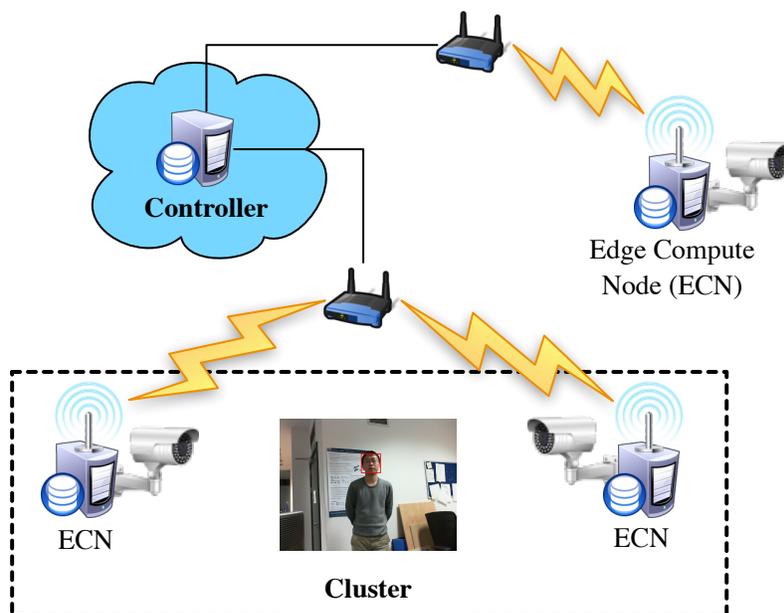


Figure 4.2: Vigil architecture, in which *Edge computing nodes (ECN)* are connected to camera devices to perform simple vision analytic functions, while uploading a relevant portion of video feed to a *Controller* in the Internet. Cameras monitoring the same areas form a *cluster*.

algorithm, requesting ECNs to only upload a fraction of relevant video frames to conserve wireless bandwidth.

Vigil further utilizes saved wireless bandwidth to provide public network access, by augmenting each ECN with a WiFi access point. The AP forwards users' traffic over the controller to the Internet, thus providing Internet hotspot functionality for nearby users.

To improve the accuracy of vision analytic functions, we introduce the notion of a **cluster**: a group of camera nodes monitoring a single geographical area with **substantially overlapping** views, as illustrated in Figure 4.2. Leveraging a cluster of cameras allows us to capture multiple views of objects from different angles and overcome the limitations of the state-of-the-art visions algorithms (Figure 4.3). These clusters are constructed during a calibration phase based on the covered area of each camera. Vigil effectively fuses the observations



Figure 4.3: Different views on the same scene can reveal more people at favorable angles to the camera, as captured by Camera 1's view of this scene with three people.

from cameras in a cluster to improve surveillance accuracy without significant wireless bandwidth overhead while existing surveillance systems upload video from each camera to the cloud before executing vision analytic functions.

Design goals and scope. The primary goal of our design is to maximize the number of query-specified objects the system returns while minimizing the bandwidth required to upload the images containing these objects. We also limit the scope of our design, noting the following non-goals:

1. Each vision algorithm has a certain accuracy and degree of confidence in the results it returns. Improving the accuracy of vision algorithms is outside this paper's scope.
2. Enough cameras are present and use a high enough resolution and frame rate so that with high probability, the resulting raw video streams capture objects of interest.
3. In a Vigil deployment, cameras are line-powered, so there are no battery-conservation issues.

The next section describes the ECN in detail, followed by a description of how Vigil's controller prioritizes frames to upload within a cluster of ECNs (§4.3, *shape Intra-cluster processing*), and arbitrates demand across multiple clusters (§4.3, *shape Inter-cluster traffic shaping*).

Edge compute node

We begin by describing the stateless image processing functions performed by each ECN. Each Vigil application implements a callback API *frameUtility*, which returns an integer value evaluating the importance of a video frame to that application. Referring to our two use-cases, queue counting application at a coffee shop defines *frameUtility* to be the number of people visible in the frame. The application that locates people or objects of interest defines *frameUtility* to be one if the object of interest is found in the frame, and zero otherwise. When Vigil receives a new query, the query contains a definition of the *frameUtility* function, which is disseminated to all ECNs. Each ECN calls *frameUtility* on every received frame to generate an array of *analytic data* we denote as *utils*. It then uploads these analytic data to the controller.³ While we focus on person-counting applications to define *frameUtility* in this paper, Vigil can process any queries that can process a vision analytic function at ECN and output a *frameUtility* (for example, an object or a license plate number is present or not).

We note here that camera placement, focus, environmental conditions, motion and many other factors can blur objects and faces of interest to the system. Indeed, based on our experience, vision algorithms such as face detection are more likely to fail on blurred images, and so *frameUtility* implicitly factors the image quality into the number of detected objects.⁴ We elaborate further on this design choice in Section 4.6.

Video storage. Each ECN is equipped with some persistent storage devices to retain all the video frames captured close to the time of detected events. This allows Vigil to support “drill-down” queries, which can be answered by uploading additional video data from ECNs to the cloud for more detailed analysis. Consequently, Vigil provides a time window, such as one or two weeks, within which the system retains important video information, as existing wired surveillance systems do.

³As discussed below in Section 4.3, an advanced intra-cluster scheduling algorithm also requires ECNs to upload the location of each detected object.

⁴Blurry frames will result in fewer objects being detected, thus the *frameUtility* metric will characterize blurry frames as less useful. Furthermore, the minimum size required in terms of face pixels is encoded in the face or person detector.

Frame index:	0	1	2	3	4	5	6	7	8	9
util[1]:	1	1	3	3	3	3	3	1	1	1
util[2]:	1	1	1	1	1	1	1	1	1	1

Figure 4.4: The intra-cluster frame selection algorithm processes on counts representing the number of objects from each ECN camera in a cluster. This example shows object counts over a certain time epoch from two ECNs.

Intra-cluster processing

The Vigil controller runs an intra-cluster algorithm to determine the most valuable frames from cameras within a cluster to upload. The key challenge is to eliminate redundant observations of multiple cameras within a cluster, capturing the same objects, to minimize communication bandwidth without actually exchanging the redundant frames. This section describes our intra-cluster scheduling algorithm in two iterations: a straw-man version described next, and its generalization in Section 4.3, which proposes a re-identification approach to check if the same objects are captured by cameras in a cluster.

Basic frame-selection algorithm

The basic algorithm selects frames to upload by examining the frame utility array $utils[c]$ that each ECN c reports to the controller. We show an example of the controller’s view of the frame utility arrays for a cluster containing two ECNs in Figure 4.4: each element of $util$ is an object count captured by a ECN during consecutive time slots. The maximum object count is based on the vision analytic function ECN is running (i.e., practically, the vision analytic algorithm will only detect a limited number of faces in a frame, for example). To reduce protocol overhead, our intra-cluster algorithm operates over a certain number of L_e time slots, referred to as an **epoch** in this section. The controller selects a single ECN in each epoch—the selected ECN then uploads a fraction of its

	0	1	2
sis:	(2,3)	(7,1)	(0,1)

Figure 4.5: Step 2 of the Vigil intra-cluster frame selection algorithm aggregates one camera's beginning frame index and utility into a *selected image sequence* (*sis*) array containing a sequence of (frame index, utility) pairs.

frames to the controller determined by inter-cluster traffic shaping (§4.3 on p. 5). The basic version of our scheduling algorithm proceeds in three steps:

shape Step 1: The controller sums the L_e object counts from each camera across the epoch, selecting the camera c^* with the highest average counts (most information about the scene) in the epoch. In the running example of Figure 4.4, the algorithm selects Camera 1 for further processing.

shape Step 2: The second step of the algorithm processes Camera c^* 's counts, finding the frames that begin changes to the scene, and collecting them into a *selected image sequence* array *sis*, as shown in Figure 4.5. The *sis* array contains pairs of (frame index, utility), sorted by utility, breaking ties in utility by favoring the *sis* element with the longer duration sequence of images.

The frames can only be uploaded at a rate lower than the capacity of the wireless link from the ECN to avoid network congestion and frames from being dropped. To avoid congestion, the controller estimates the link capacity (in bits per second) C available from each ECN by examining the near-term sending rates and the loss rates. The ECN measures a time-averaged packet loss rate L at its wireless link layer, and the physical-layer bit rate R . The ECN then estimates the link capacity as:

$$C = R \cdot (1 - L). \quad (4.1)$$

shape Step 3: The final step of the algorithm takes as input the estimated available wireless capacity C , and estimates the number of bits that ECN c^* can upload. We make a simplifying assumption that ECNs within a cluster have similar wireless links to controller because they cover same geographical area of interest.

Suppose the size (in bits) of the frames in the sis is N_{sis} and the length of the epoch in seconds is T_e . At the end of the epoch, the controller sends a control message to ECN c^* soliciting an upload. If $N_{sis} \leq C \cdot T_e$, the ECN uploads sis , along with all the changing frames as indicated in the sis after compression. Otherwise, the ECN uploads a fraction of the compressed images in sis in the decreasing order of utility. In the example of Figure 4.5, $sis[0]$ has utility of 3, so frame 2 is uploaded first, followed by frame 7 and 0.

We note that this algorithm is an approximation that will lose information when more than one camera in the cluster sees objects that other cameras in the cluster miss. So, we describe a sophisticated intra-cluster algorithm to select the most valuable frames within a cluster.

Sophisticated frame-selection algorithm

Vigil’s sophisticated intra-cluster frame-selection algorithm specifically targets cases where one camera in a cluster sees objects that other cameras miss to select more than one ECN to upload images during a time epoch.

This algorithm relies on geometry and known locations of the ECNs to detect redundant viewpoints without actually exchanging the redundant frames. The algorithm first identifies duplicates of the same objects from multiple ECNs in overlapping camera views using object re-identification. It then prioritizes video frames from the cluster, factoring the count of “re-identified” objects into the frame utility metric.

Object re-identification. Object re-identification determines redundant objects reported by multiple ECNs in a cluster with overlapping camera views. By selecting the smallest subset of camera views to cover the overlapping views, we identify the unique objects across cameras within a cluster. To achieve this, Vigil uses the following lightweight approach [116] called re-identification.

Figure 4.6 shows an example where two cameras simultaneously detect the faces of two people in their overlapping views. The re-identification algorithm identifies if the detected face instances belong to the same person or not. It first selects common reference points between the camera views, for e.g., the

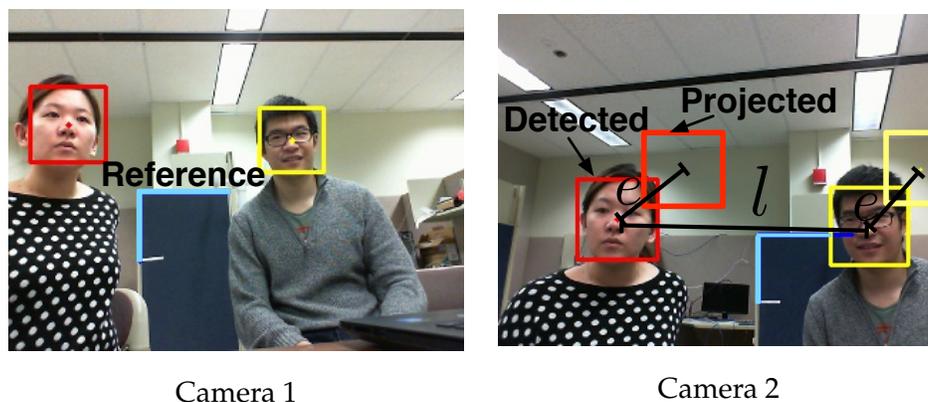


Figure 4.6: Two cameras simultaneously capture a scene containing the same two faces. To avoid redundant counting, Vigil projects faces from Camera 1 to Camera 2 based on common reference points denoted by the two blue lines labeled “Reference” in Camera 1’s view. If projection error does not exceed the distance between the detected faces, re-identification correctly identifies the two views of each face.

two blue lines in Figure 4.6.⁵ The reference points are then used to project **any** pixel in Camera 1’s view onto the plane of Camera 2’s view. The detected face instances are identified as the same person when the distance between projected face and the detected face is below a threshold. For e.g., the error e between the projection and the detected face in the second panel of Figure 4.6 is much smaller than the distance between the two different faces. We set an empirical value to determine the threshold for this projection error e that accounts for any inaccuracy in marking reference points as benchmarked in Section 4.5. The threshold value is set during a calibration stage and can remain useful for a long period of time for static cameras. This re-identification approach has linear complexity in number of cameras because it projects all of the captured scenes to a common plane to calculate inter-object distance.

To integrate the re-identification technique on top of the basic algorithm Section 4.3, each ECN reports the center coordinates of detected faces and the frame utility in the modified *utils* to the controller. The controller then

⁵The reference points are manually determined in a camera calibration phase or extracted using algorithms that extract SIFT image features [48].

Frame index:	0	1	2	3	4	5
objects[1]:	a, b, c	a, b, c	b, d	b, d	b, d, f, g	b, d, f, g
objects[2]:	a, b	a, b	d, e	e	d, h, i	f, h, i
Frame index:	0	1	2	3	4	5
util[1]:	3	3	2	2	4	4
util[2]:	0	0	1	1	2	2

Figure 4.7: The advanced intra-cluster frame selection algorithm operating on unique objects after applying the object re-identification algorithm. Each letter denotes an unique object identified by re-identification. The algorithm chooses ECN 1's camera to be the primary camera view, while debiting the object counts reported by ECN 2 by the number of duplicate objects re-identified (in bold).

performs re-identification using the analytic data in *utils*. Based on the results of object reidentification, The controller then executes the following sophisticated scheduling algorithm:

shape Step 1: The controller determines an ECN that has maximal average object counts (thus capturing the most information about the scene) to be the **primary** ECN. It then projects the detected objects from other (**complementary**) camera views onto the primary camera view. For each re-identified object, the controller debits the object count of all the complementary ECNs by one. This produces an updated utility array for each ECN. For example, Figure 4.7 shows unique object counts captured by two ECNs after applying object re-identification, along with their utility arrays. We choose ECN 1 as the primary camera view, while debiting object counts for ECN 2 by the number of duplicate objects re-identified (marked in bold).

shape Step 2: The controller determines a selected sequence of frames in the **sis** array for each ECN, which comprises tuples of (ECN identifier, frame index, utility). As illustrated in Figure 4.8, the modified **sis** array includes the frames captured by the primary ECN with changes in its object count (*i.e.*, frames four, zero, two). It also contains frames captured by the complimentary ECNs when a frame captured by the primary ECN fails to cover all the unique objects. For

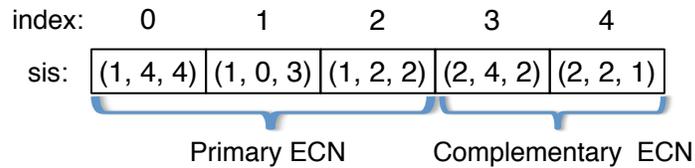


Figure 4.8: Step 2 of the advanced intra-cluster algorithm generates a selected image sequence (**sis**) array comprising camera identifier, frame index, utility.

example, frames four and two of the (complementary) ECN 2 are appended to the **sis** array because they include additional unique objects, *i.e.*, objects h and i in frame four and object e in frame two.

shape Step 3: The controller consults the estimated wireless capacity C to determine whether all the selected frames in **sis** can be uploaded. If not, it prioritizes the selected images from the primary ECN, in decreasing order of their utility value. It then polls the selected images from all the complementary ECNs, in the order of their debited utility value.

Vigil can handle a large number of objects in a single frame as long as the vision analytic function is capable of doing so. The intra-cluster frame selection falls back to uploading each frame when the number of objects per frame are very large for each camera in the scene.

Inter-cluster traffic shaping

After determining the priority of frames to be uploaded within a cluster, the Vigil controller needs to coordinate upload bandwidth demand across the clusters that are within a wireless contention domain (*i.e.* served by a single access point). To do so, Vigil uses a novel inter-cluster traffic scheduling algorithm that attempts to allocate upload rates to each cluster that maximizing the number of useful objects per second delivered to the application.

We describe our algorithm in the context of two application scenarios, *i.e.*, counting applications such as customer queue analytics, and security based applications such as tracking a person or finding a suspicious object. These scenarios are monitored by two clusters of ECNs, which can contend with each

other for uploading images. For the application scenarios related to analyzing a queue of customers or locating an object of interest, Vigil allocates rates proportional to the objects detected per second at each camera cluster. In these scenarios, the application can manually intervene when it detects large number of people or objects. For e.g., in a Starbucks coffee shop, the queues with higher person count could be served faster, and at the airport security checkpoints, the queue with highest person count would use more personnel to manage the queue. Following the notation of Section 4.3, we denote wireless capacity from cluster c as C_c . Since different video frames compress at different ratios, ECNs may also upload unequal frame sizes: we denote the size of the compressed frame in i th index of the selected image sequence from cluster c as L_i^c . Based on these quantities, we calculate the number of useful objects per second (*shape ops*) for i th index of the selected image sequence of cluster c :

$$\text{ops}[c][i] = \frac{\text{sis}[c][i].\text{utility}}{L_i^c/C_c}. \quad (4.2)$$

The numerator of the *ops* metric is a count of objects, while the denominator has units of seconds (bits divided by bits per second). The *ops* thus captures how many useful objects per second the frame at i th index of the selected image sequence from cluster c will deliver if it is scheduled for transmission.

shape Step 1: Every time epoch, each cluster c uploads its selected image sequence sis , to the controller, which the controller stores in array element $sis[c]$, an array of selected image sequences indexed by cluster number c .

shape Step 2: For each cluster c and frame group index i , the controller computes $\text{ops}[c][i]$ using Equation 4.2. The controller's state now appears as in Figure 4.9: a per-cluster queue of image sequences' *ops* values.

shape Step 3: The controller schedules service to different clusters using a variant of deficit round robin (DRR) scheduling [87] to approximate fair queuing [19]. To define terminology and provide context, we now briefly recall the DRR algorithm, in the context of serving Vigil clusters. Each cluster has a *deficit counter*, which represents the amount of information it is allowed to transmit when it is its turn. Vigil -DRR works by considering clusters in round-robin

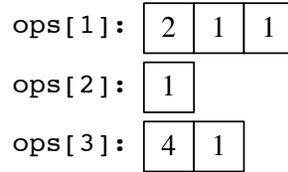


Figure 4.9: The per-cluster ops queues at the controller capture the utility of respective image sequences uploaded from each cluster in objects per second.

order. The controller adds a fixed amount of credit, called the *quantum*, to each cluster's deficit counter.⁶ If the cluster's deficit counter exceeds the size of the packet, then the cluster transmits the packet and the controller decrements the cluster's deficit counter by the size of the transmitted packet.

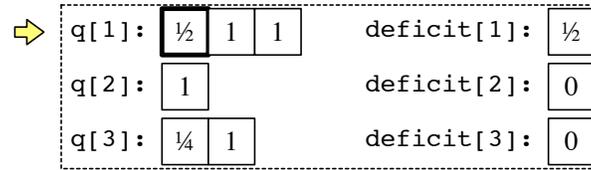
Our DRR variant uses the reciprocal of *ops* in place of the packet length, for queue weights

$$q[c][i] = \frac{1}{ops[c][i]}, \quad (4.3)$$

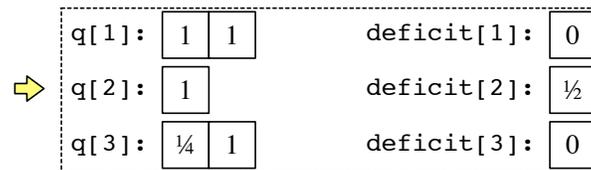
in the case that *ops* is non-zero, and drop the upload in the case that *ops* is zero. Vigil -DRR provides fair air-time to different clusters, each of which may have different wireless throughputs possible to the base station. Vigil -DRR also provides fairness between clusters in terms of the number of seconds per object change (utility). Thus a cluster that can upload a frame containing two object changes compared to a cluster that can upload a frame containing one object change.

Vigil -DRR also maximizes the number of objects per second delivered to the controller. To see this, consider the example of Figure 4.9 where there are three clusters, with frames at their respective queue heads having *ops* of two, one, and four respectively. If we service the clusters at rates r_1 , r_2 , and r_3 (bits per second), the number of objects per second uploaded to the controller will be the inner product of the preceding rate vector with the *ops* vector (2, 1, 4): $\langle (r_1, r_2, r_3), (2, 1, 4) \rangle = 2 \cdot r_1 + 1 \cdot r_2 + 4 \cdot r_3$. The Cauchy-Schwartz inequality states that we can maximize this inner product (*i.e.*, the number of objects per

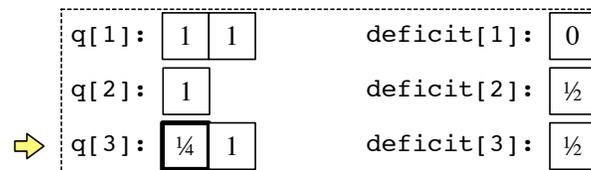
⁶We describe our setting of *quantum* in Section 4.4.



Transmit from q[1].



No transmit from q[2], update deficit.



Transmit from q[3].

Figure 4.10: The Vigil-DRR algorithm operating over three camera clusters with a quantum of $\frac{1}{2}$.

second uploaded) by choosing (r_1, r_2, r_3) proportional to the *ops* vector. Setting packet length inversely proportional to *ops* in the DRR algorithm accomplishes this, as DRR will schedule packets at rates inversely proportional to the packet length normalized by the wireless throughput.

Figure 4.10 shows an example of Vigil-DRR in operation over three clusters, with a quantum value of $\frac{1}{2}$. The first panel of Figure 4.10 shows the initial state of all queues. At the first time-step, the algorithm increments the deficit counter of the first queue by quantum ($\frac{1}{2}$ in this example) and then checks the $q[i]$ value of the change at the head of the queue. Since it is $\frac{1}{2}$, Vigil-DRR transmits the change and decrements the deficit counter by $\frac{1}{2}$, leaving zero in the deficit counter, as shown in the second panel of Figure 4.10. At the next

time-step, the algorithm increments the deficit counter of the second queue by $\frac{1}{2}$, but the $q[l]$ value at the head of the second queue is greater than $\frac{1}{2}$, and no transmission occurs, as shown in the third panel of Figure 4.10. At the final time-step, the algorithm increments $deficit[3]$ by $\frac{1}{2}$ and transmits, leaving $\frac{1}{4}$ in the deficit counter. The algorithm then proceeds similarly in a round-robin fashion.

For certain security-based applications such as intruder detection and tracking a person, the ECN nodes may assign high-priority to the captured frames. We modify the design of Vigil-DRR algorithm similar to MDRR algorithm to allow a high priority queue that allows the frames requesting priority access on the channel to be uploaded immediately overriding the *ops* metric.

4.4 IMPLEMENTATION

In this section, we describe our implementation of Vigil . With the goal of understanding system performance *in situ*, we have deployed Vigil at three sites. We describe these deployments in the next section, and hardware and software details in Section 4.4 and Section 4.4, respectively.

Testbed deployments

We deploy a single cluster of Vigil cameras at three sites under vastly different operational conditions.⁷ Firstly, we study an outdoor surveillance scenario by deploying a cluster of camera ECN nodes at a shuttle bus stop at Site #1's outdoor campus (the third panel of Figure 4.11), where we monitor real-time vision analytic functions such as counting passengers. The ECN nodes in this deployment connect to a controller by long-distance backhaul links over TV whitespaces. Secondly, we study an indoor surveillance scenario in a busy office hallway by deploying a cluster of camera ECN nodes at Site #1, where we monitor the frequency of passers-by. The ECN nodes in this deployment

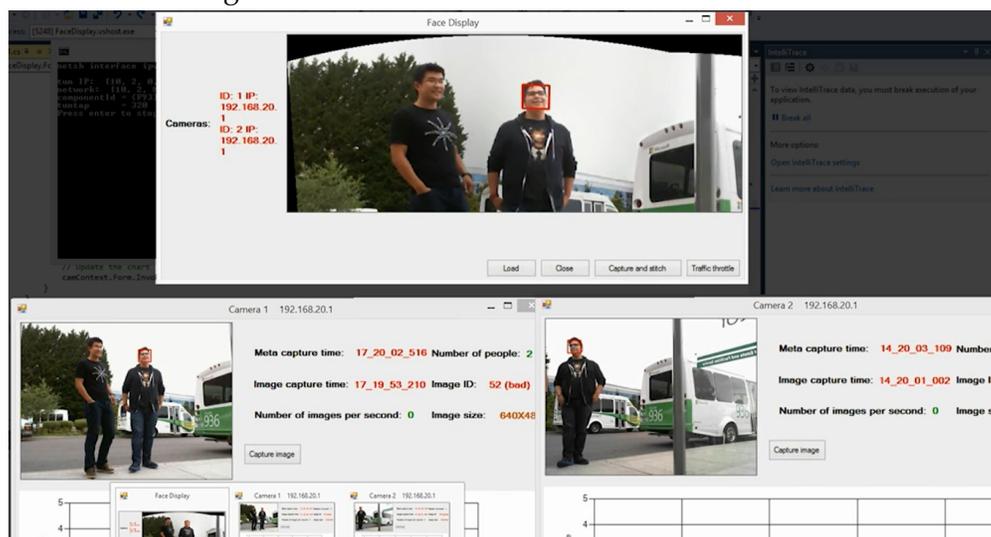
⁷Privacy of the monitored users is preserved by ensuring that the cameras only capture users who had given prior permission to take part in the study.



Site #1 lounge.

Site #2 office.

Site #3 lab.



Site #1 outdoor bus stop pilot.

Figure 4.11: Vigil deployment at three sites.

use unlicensed 2.4 GHz WiFi to connect to the controller. Finally, we monitor an open-plan office at Site #2 (the first panel of Figure 4.11) and an indoor lab environment at Site #3 (the second panel of Figure 4.11), where we monitor working hours and office occupancy. The ECN nodes in these deployments use unlicensed 2.4 GHz WiFi at Site #2 and TV whitespace radios at Site #3 to connect to the controller. The indoor deployments in the three sites have been operational for the last two months, giving us valuable information on traffic patterns.

Hardware

We describe the hardware platform used to implement the controller and the ECNs. For the ECN nodes, we use laptops running a user-space program to perform image analysis functions on video feeds from the connected cameras. We have used laptops and the Intel Next Unit of Computing (NUC) to prototype our system, but embedded devices (e.g. Gatework routers or NUC) with 500 MHz–1 GHz CPU have enough processing power to run image analysis functions for ECNs. Recent trends in vision are moving toward smart cameras, thereby enabling face recognition, motion detection, and other image analysis tools to be implemented to an increasing extent in hardware [32].

Each laptop is also attached to a WiFi based router to provide public Internet access. WiFi traffic along with vision analytic data are sent over the connected routers to a central controller. Linux-based routers running the OpenWRT operating system upload vision analytic traffic from the ECN nodes to the controller. These router boards control two different types of wireless interface cards for communication in TV whitespaces and the 2.4 GHz ISM band, respectively. In our outdoor deployment, we use TV-band transmitters from Doodle Labs [20] for TV whitespace communications. The radios are configured in a single vacant TV channel at a center frequency of 580 MHz, with a 5 MHz bandwidth.⁸ In our indoor deployment, we use off-the-shelf 802.11a/b/g radios operating in the 2.4 GHz band.

We implemented the central controller on a workstation hosting a user-space program that collects ECN traffic for further processing. Both ECN and controller run code implemented in Microsoft Visual Studio on Windows 8.1.

Software architecture

This section describes the software architecture of the controller and ECNs. The entire software codebase consists of 6,000 lines of C Sharpcode that implements

⁸Before each experiment, we query a commercial spectrum occupancy database [92] to ensure this channel is vacant.

vision analytic algorithms and end-to-end protocols, along with approximately 100 lines of C code for ath5k driver modifications.

Virtual networking device: We use the `tun` virtual networking device in OpenVPN [66] software at each camera node. All traffic from WiFi users is directed through this virtual device, which is subsequently captured by our application for traffic shaping before sending through the underlying whitespace interface.

MAC layer modifications: We disable the rate adaptation function in the ath5k driver, and allow the ECN nodes to control the physical-layer data rate used to send each packet by appending a special bit-rate in the header of each packet to be transmitted. This allows the controller to accurately estimate the wireless capacity C of the link from the ECN node to the controller (equation 4.1). In our evaluation, we fixed C to isolate the effect of our frame selection algorithms. However, any rate adaptation algorithm can be easily adopted in ECNs to further improve performance.

4.5 EXPERIMENTAL RESULTS

In this section we evaluate Vigil 's performance gains over conventional approaches. Sections 4.5 and 4.5 evaluate the accuracy improvement of Vigil 's intra-cluster frame selection (§4.3) and inter-cluster traffic shaping (§4.3) components. Section 4.5 presents microbenchmarks that stress-test Vigil 's vision and advanced scheduling algorithms (§4.3).

For object-counting applications, we measure the accuracy of Vigil as the accuracy of object counts relative to baseline data from running vision algorithms on all frames of the raw video streams of all available cameras. If the system uploads all the frames where the object count changed (*i.e.* the zeroth, second, sixth, and eighth frames, as shown in Figure 4.12) the system will make no errors relative to baseline and has 100% accuracy. Otherwise the accuracy is the percentage of frames that the system uploads where the object count changed.

Frame index:	0	1	2	3	4	5	6	7	8	9
utility:	1	1	2	2	2	2	1	1	3	3
Changes:	1		2				1		3	

Figure 4.12: Frame utilities and indices where utility changes. Uploading all the frames where utility changes leads to a 100% accuracy.

Intra-cluster frame selection

In this section, we evaluate the accuracy of Vigil in a cluster of cameras using intra-cluster frame selection (§4.3).

Methodology. In this experiment, we use video traces collected from a cluster of three cameras at Site #3 in a lab environment. The three cameras log video traces synchronously for a duration of 180 seconds, with about 2,000 image frames. The ECN connected to the camera uploads detected face counts and the controller selects which frames are uploaded from ECNs based on the intra-cluster frame selection algorithm.

We choose the ECN's slot time (§4.3, p. 3) to be 100 milliseconds: this strikes a good tradeoff between the detection errors of vision algorithm and responsiveness in detecting people. We configure the epoch time L_e to be five slots: this choice of epoch length reduces the protocol overhead of sending control messages, while enabling use of the best available camera in detecting people. We experimented with other choices of L_e and found end-to-end performance was not sensitive to this parameter.

We compare the accuracy of intra-cluster frame selection in Vigil to two approaches: a *Round-Robin* approach that cycles through all cameras within a cluster in a round-robin manner to upload frames and a *Single-Camera* approach that arbitrarily selects a single camera to upload frames. Note that all approaches only upload the frames where the count of detected faces changes. We constrain the capacity of wireless link from each ECN to the controller and repeat the experiment five times.

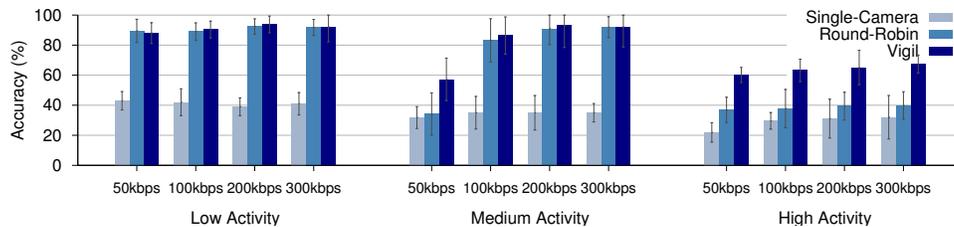


Figure 4.13: Accuracy of intra-cluster frame-selection in Vigil relative to a single-camera system and a multi-camera system with round-robin scheduling. Error bars show standard deviation of the experiment in varying wireless conditions.

Level	# people	Change interval(s)	Bandwidth(Kbps)
Low	1	5–15	4–16
Medium	4	1–2	32–80
High	7	0.2–1	100–400

Table 4.1: Summary of video traces used to benchmark intra-cluster algorithms in terms of the number of participants, the frequency of change in object counts, and required bandwidth to upload the frames where object count changes.

Results. Figure 4.13 shows the performance gains of Vigil as we increase the per-camera available wireless capacity for video traces collected at low, medium and high activity levels. The bandwidth required at each activity level is summarized in Table 4.1. In Figure 4.13, the bandwidth required at low activity level (at most 16 Kbit/s) is lower than the available per-camera wireless capacity and therefore, both Vigil and *Round-Robin* achieve more than 90% accuracy, while the single camera suffers because of lack of sufficient coverage. Similar results are observed for medium activity level, except Vigil outperforms other approaches when the available per-camera wireless capacity 50 kbps is lower than the bandwidth required for medium activity level (at most 80 kbps). Finally at high activity level, the bandwidth required is much higher than the available per-camera wireless capacity and we observe 23-30% gains for Vigil compared to *Round-Robin* because Vigil prioritizes those frames across cameras that maximize the accuracy.

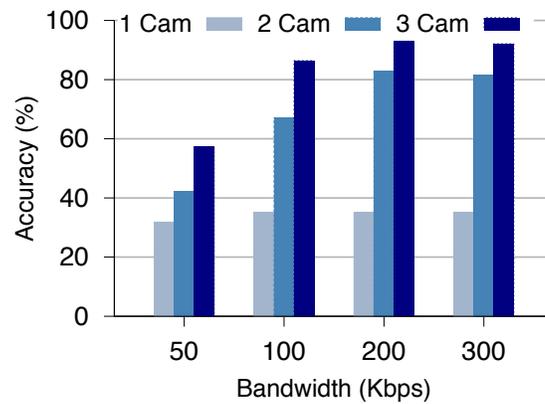


Figure 4.14: Vigil 's accuracy in a single-cluster surveillance network with different number of cameras.

We gather more insight in to why Vigil results in higher accuracy compared to a round-robin or a single camera approach. Figure 4.14 shows that Vigil 's accuracy increases with the number of cameras up to the point where no blind spots are left uncovered. In this example, two cameras provide a significant gain over a single camera approach, but subsequently adding more cameras does not improve performance, because Vigil already prioritizes those frames which maximize accuracy.

Inter-cluster scheduling

In this section, we evaluate the accuracy of Vigil across multiple clusters of cameras by using the inter-cluster traffic scheduling algorithm (§4.3), examining to what extent the system can maintain accuracy as wireless capacity becomes more and more scarce, and more camera clusters contend on the same wireless bandwidth. We compare the accuracy of Vigil 's inter-cluster traffic scheduling algorithm to two approaches: an equal throughput allocation which is a throughput-based fairness policy that gives equal throughput to all the camera clusters such as in the case of WiFi and an equal time based allocation which is

a time-based fairness policy. Note that all approaches only upload the frames from the selected image sequence *sis* of each ECN.

Methodology. We simulate a network of clusters of cameras that contend over a shared wireless channel. We vary the wireless capacity of this shared channel from 1 Mbps to 20 Mbps to quantify the ability of inter-cluster frame selection to alleviate congestion on the shared wireless medium. We emulate different activity levels by modeling the arrivals at each camera cluster by a Poisson arrival process to emulate the traffic patterns from our real-world deployments, where an increasing rate λ corresponds to higher activity levels. We assume that each arriving person departs after a constant dwell time. A single image in our experiments is 30 Kbytes, which takes approximately 240 ms to upload at one Mbit/s. We choose a Vigil -DRR *quantum* of 100 (seconds/object) so that Vigil -DRR would transmit a three-object frame without cycling round the clusters. We found that Vigil -DRR is not sensitive to our choice of *quantum*. We simulate the system over a time period of approximately one hour.

We evaluate the accuracy of Vigil -DRR algorithm across multiple clusters of cameras. In these experiments, we simulate a network of ten clusters of cameras that contend over a shared wireless bandwidth where each cluster has two cameras. The number of faces detected at each cluster is modeled by a Poisson arrival process, where the rate of the Poisson arrival process λ is set to 2.5 (objects/second) for low activity level, 5 (objects/second) for medium activity level, and 12.5 (objects/second) for high activity level. Note that while Vigil selects the most relevant frames from two cameras in each cluster based on intra-cluster frame selection, but the equal throughput and equal time approach assume one camera per cluster for fair comparison of traffic scheduling.

Equal wireless capacity

We first consider the scenario where the available wireless capacity is same from each ECN to the controller in ten camera cluster. Figure 4.15 shows the performance gains of Vigil -DRR as we increase the shared wireless capacity at low, medium and high activity levels. We observe that Vigil -DRR requires more wireless bandwidth to achieve 100% accuracy at higher activity levels. In

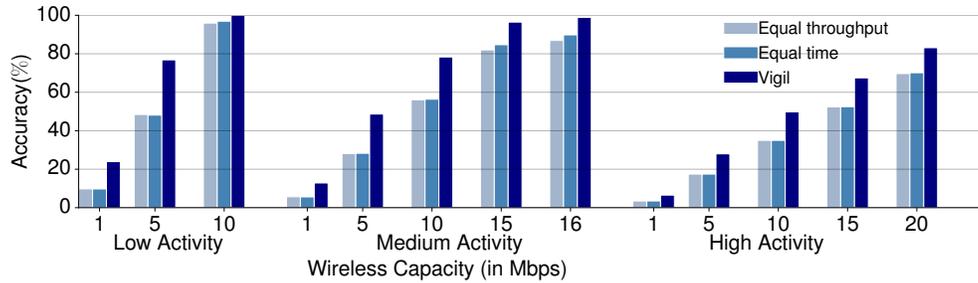


Figure 4.15: Accuracy of a multi-cluster system as the wireless capacity varies as shown on the x-axis. We compare Vigil with time-based fairness and equal throughput allocation for ten cluster of cameras. Vigil uses two cameras per cluster to select the most relevant frames, but equal throughput and equal time approach assume one camera per cluster for fair comparison.

this example, Vigil -DRR utilizes only 10 Mbit/s at low activity level to achieve 100% accuracy for ten camera clusters, but at high activity level, it achieves only 80% accuracy at bandwidths as high as 20 Mbps. Further, we note that Vigil -DRR significantly outperforms the equal throughput allocation and equal time allocation approach when the shared wireless capacity is not sufficient because it prioritizes the frames with maximum object count, using the *ops* metric. In this example, Vigil -DRR achieves gains of 20-25% over the other two approaches at 5 Mbit/s in low activity level and 10 Mbit/s in medium activity level. Finally, we note that the equal throughput and equal time allocation approaches achieve similar accuracy gains because the available wireless capacity is same from each ECN to the controller.

Unequal wireless capacity

Now we consider a scenario where the available wireless capacity from each ECN to the controller varies across ECNs. Figure 4.16 shows the performance gains of Vigil -DRR when the available wireless capacity from five clusters to ECN is C_1 and from the other five clusters to ECN is C_2 in a network with ten clusters of cameras. We first note that equal throughput allocation approach penalizes the clusters with high wireless capacity C_2 to sacrifice accuracy to ensure all clusters get equal throughputs. On the other hand, equal time

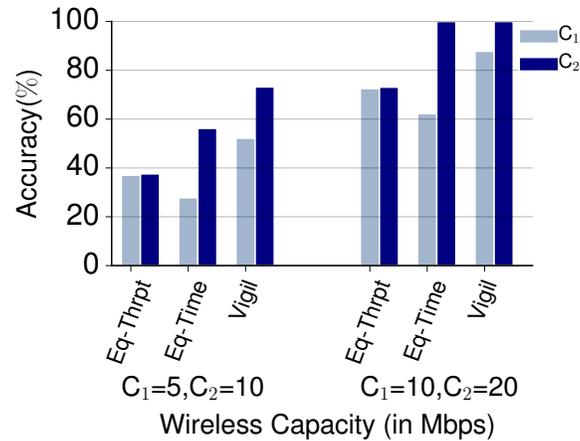


Figure 4.16: Accuracy of a multi-cluster system with 10 clusters where the wireless capacity of 5 clusters is C_1 and other 5 clusters C_2 . Vigil outperforms both equal time-based fairness (Eq-Time) and equal throughput allocation (Eq-Thrpt).

allocation approach ensures time-based fairness allowing the clusters with high wireless capacity C_2 to upload more frames than clusters with low capacity C_1 . But Vigil -DRR outperforms both these approaches in terms of accuracy for both the clusters with low wireless capacity C_1 and high wireless capacity C_2 . Further, the gap in accuracy between the high-capacity and low-capacity clusters is much smaller for Vigil -DRR compared to equal time allocation approach because of maximizing the *ops* metric.

Vision algorithm microbenchmarks

In this section, we evaluate the two vision algorithms Vigil uses: face detection and a re-identification algorithm to associate faces detected in overlapping camera views.

Face detection

This evaluation answers two questions: first, how accurate is the Haar-cascade-classifier-based face detection algorithm used in our system? Second, in terms

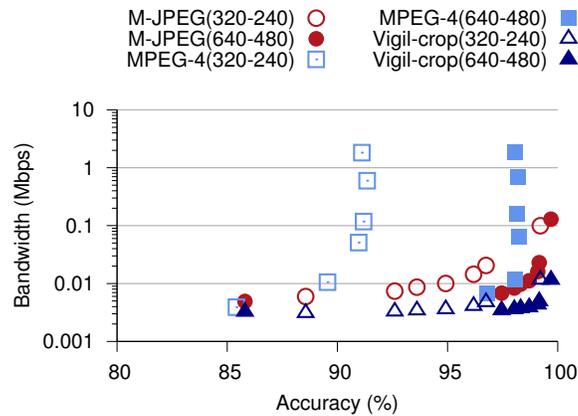


Figure 4.17: Bandwidth required versus accuracy of the face detection vision algorithm on videos compressed with different video compression algorithms, including our algorithm Vigil -crop that performs M-JPEG on the cropped objects. Only the selected frames in the `sis` array are compressed for this bandwidth calculation.

of that accuracy, what is the impact of various video compression schemes on system bandwidth savings? We compare two state-of-the-art video compression algorithms, M-JPEG and MPEG-4, to determine which fits the design of Vigil best.

Methodology. We use a single camera to record five-minute video traces at two different resolutions. Each trace contains about 7,000 images. Two people arrive and departed randomly in the scene, facing the camera. The distance of the subjects to the camera ranges from two to eight meters. The ground-truth person count is established by visual confirmation. We compress the frames with different state-of-the-art compression algorithms at different levels, and then apply a face detection algorithm on the compressed images to understand the tradeoff of accuracy and bandwidth required.

Results. Figure 4.17 shows a scatter plot of bandwidth required as the accuracy of the face detection vision algorithm increases. For each video compression algorithm, the accuracy of the face detection increases when it is compressed less because of low information loss. Here Vigil -crop applies M-JPEG compression only on the cropped faces in an image (by replacing the image background

with a single RGB color). We observe that Vigil -crop outperforms M-JPEG compression without object cropping by 2–5× in bandwidth savings. It even outperforms the state-of-the-art MPEG-4 algorithm by a factor of two in bandwidth savings for the same accuracy. We therefore choose M-JPEG algorithm with object cropping in Vigil . Further, we note that Vigil allows a wide accuracy and bandwidth tradeoff compared to MPEG-4. This is because MPEG-4 applies delta-based frame compression, which either removes all the details in intermediate frames, or have to keep most of the redundant information.

Object re-identification

We evaluate the accuracy of the re-identification algorithm described in Section 4.3. This section addresses the question of how well Vigil can tag faces in overlapping camera views as to the same person.

Methodology. In this experiment, two cameras synchronously log video traces at four indoor locations, where each trace consists of 300 images with the faces of same two people detected at different locations. The two cameras have a partially overlapping view as illustrated in Figure 4.6. The re-identification algorithm projects the faces detected at the first camera onto the corresponding images captured by the second camera. It then calculates the distance between the projection and the faces detected at the second camera, which is the projection error e in the second panel of Figure 4.6. This projection error is then compared to the distance l between the two people’s faces detected in the same camera view.

Results. Figure 4.18 shows the CDF of the projection error e and the distance l between different faces of all the images at each captured location. We observe that the maximum projection error e is 89 pixels at these locations, and is much lower than l for most of the cases. Thus, we use this maximum e as the threshold of associating projected faces in our implementation. (§ 4.3), which can lead to >98.1% re-identification accuracy at all the measured locations. The maximum projection error can be obtained by running this experiment as part of the camera calibration process. The threshold value, set during calibration

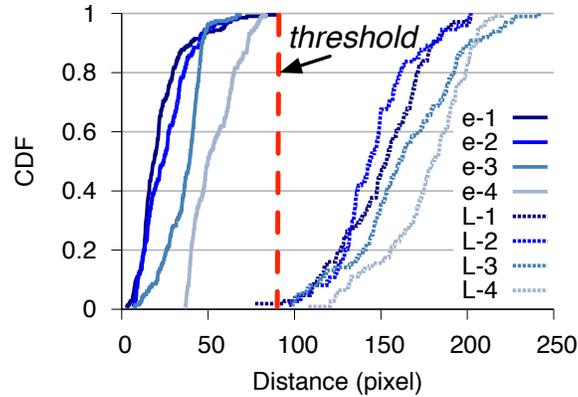


Figure 4.18: CDF of the distance l between two people's faces and the projection errors e when mapped from one camera view to another (illustrated in Figure 4.6).

Resolution	Face detection	Re-identification	Compression
320×240	32ms	2ms	13ms
640×480	80ms	2ms	15ms

Table 4.2: Average processing delay of different vision analytic functions in Vigil using a laptop with a 2.4GHz dual-core CPU.

process, is useful for a long period of time for static cameras. Finally, our current algorithm can fail to distinguish unique objects that are densely located (e.g., within <89 pixel). In such a scenario, we may revert back to the basic intra-cluster scheduling by picking the maximum object counts from all the ECNs. Enabling Vigil to dynamically switch between different scheduling algorithms remains as our future work.

Finally, Table 4.2 summarizes the processing delay of running different vision analytic functions of Vigil on a standard laptop. The latency is measured based on our video traces at two different resolutions. We observe a low latency of <80 ms for all the processing functions, which enables Vigil to promptly capture objects in highly dynamic scenes.

Area coverage

The following back-of-the-envelope calculation shows that Vigil can achieve significant area coverage gains over systems that stream MPEG-4 video. Assuming that we cluster cameras in groups of four covering a 100 sq. ft. area per cluster, each camera covers an amortized 25 sq. ft. area. Assuming an available capacity of 20 Mbit/s, the status quo approach of deploying a camera stream 1 Mbit/s video (a typical MPEG-4 rate) will support 20 cameras, for a total coverage area of 500 sq. ft. But referring to Table 4.1, we see that the bandwidth a low activity scene actually requires is only on the order of 10 Kbit/s, while a high activity scene requires approximately 200 Kbit/s. So Vigil can function at data rates ranging from 40–800 Kbit/s per cluster, resulting in 500 clusters for low activity and 25 clusters for high activity. Consequently, Vigil covers between 2,500 sq. ft. and 50,000 sq. ft, resulting in a coverage gain of between $5\times$ and $200\times$ over status quo video streaming.

4.6 ISSUES AND DISCUSSION

We comment on various design points of Vigil with the full hindsight of the previous sections.

- **Required compute resources at edge compute node.** Recent trends in vision are moving toward smart cameras, thereby enabling face recognition, motion detection, and other vision tools to be implemented to an increasing extent in hardware [32, 55, 95]. Consequently, simple embedded platforms with 500 MHz–1 GHz CPU are sufficient to implement the vision analytic functions at ECNs.
- **Amount of ECN-local storage.** The local storage at ECN retains video frames captured close in time to significant events in the video stream. MPEG-4 video at sufficient resolution for our application (320×200 or above) consumes a maximum bit-rate of between 384 and 8,000 Kbit/second (172 to 3,600 Mbytes/hour), depending on the encoding rate. The price of hard drive storage has been rapidly falling over the years, reaching

USD 0.03/gigabyte in recent months [53], pricing the local storage between $\frac{1}{2}$ and five US cents per hour for MPEG-4 video. Based on these figures, we expect the incremental cost for including storage for most queries requires less than USD 1.00 per ECN, adding a negligible cost to the overall bill.

- **Cost of distributed processing.** Deploying ECNs together with cameras can inevitably increase the infrastructure cost. However, when wireless capacity is limited, the saved bandwidth by Vigil can be used to forward users' traffic, thereby recouping the cost of ECNs. To further reduce the cost, we envision that multiple cameras can connect to a single ECN to upload vision analytic functions of each connected camera to the controller. But this leads higher contention on the wireless medium between camera nodes, and hence, a more complex design of scheduling algorithms that we plan to address in future work.
- **Hybrid hotspot functionality.** The hybrid camera-hotspot functionality can subsidize the cost of a droplet. With the cost of a camera in the range of USD 5–10, ECNs can become ubiquitous.
- **Choice of an object count-based metric.** In Section 4.3 we motivated the use of an object count-based metric (*frameUtility*) for Vigil. We choose an object count-based metric in our design because it is a good first order approximation to frame value in a vast number of surveillance applications such as object identification and tracking. Also, our design is general enough to support any utility function by letting the controller push the utility function to the edge compute node. Therefore, the definition of utility can be modified based on the specific surveillance function performed by the system.

4.7 SUMMARY OF VIGIL

In this chapter, we presented Vigil, a wireless video surveillance system that leverages distributed camera sensors connected over a TV whitespace network to support pervasive surveillance functions. Vigil leverages edge computing

nodes co-located with camera sensors to pre-process video feeds, while only uploading those relevant frames to the cloud for deeper analysis. Such a context-aware uploading technique can significantly conserve bandwidth of the underlying whitespace network. The saved capacity is dedicated to WiFi hotspots to recoup the deployment cost. Vigil further leverages a novel frame scheduling technique that accounts for both the significance of individual frames and the redundancy across multiple cameras to optimize bandwidth utilization. Through deployments at three sites across two countries, we demonstrated that Vigil can significantly improve the bandwidth efficiency of wireless networks by enabling a $5 - 200 \times$ larger coverage than a conventional approach that simply streams the entire video to the cloud. We believe the proposed edge computing architecture has the potential of driving many vision analytic domains such as smart-city, public safety, and augmented reality.

5 RELATED WORK

In this chapter, we discuss various prior research efforts and ongoing projects that are related to this thesis. We start by discussing research in TV whitespace network designs. Then, we discuss prior approaches for whitespace determination and proposals to enhance spectrum databases. Next, we focus on wireless networks to provide vehicular Internet access. Specifically, we discuss research prototypes and commercial solutions that provide vehicular connectivity using existing wireless technologies. We then discuss centralized architectures used in cellular networks and enterprise WLANs to improve wireless performance. We close this chapter by discussing ongoing research in vision analytic systems and applications.

5.1 WHITESPACE NETWORK DESIGNS

We compare our heterogeneous architecture with prior designs based on TV whitespaces alone, and those enhanced with other communication technologies.

Using only whitespaces. There are two standards initially proposed for wireless networks operating in TV whitespaces. IEEE 802.22 standard [39] describes a WiMax-like, OFDMA based communication technology for wireless regional area networks, whereas IEEE 802.11af standard [40] is an extension of WiFi based 802.11 protocols for local area networks. Both standards assume a symmetric network in which both uplink and downlink communications are conducted over TV whitespaces. Coupled with these standards are some early network prototypes demonstrating these concepts. WhiteFi [4] is perhaps the first whitespace network that uses a WiFi-like protocol similar to IEEE 802.11af. A campus-wide network based on such design for supporting vehicular connectivity has been subsequently reported in [11]. Following these research prototypes, various commercial whitespace deployments [65] are available for emerging applications, such as rural broadband access and smart-grid monitoring. All these systems use a symmetric design based solely on whitespace links, hence suffering from the problem of power asymmetry in TV

whitespaces. Our heterogeneous design can improve their coverage by using the ubiquitous cellular connectivity to circumvent the weak whitespace uplink.

Enhancing with cellular technologies. A recent system WhiteCell [84] has also explored the use of cellular technology to enhance whitespace connectivity. Nevertheless, it is based on a very different scenario of femto-cell communications, focusing on offloading cellular traffic to the most efficient part of unlicensed whitespaces. To this end, WhiteCell uses a symmetric network design that uses whitespaces to offload most of the *bi-directional* network traffic over TV whitespaces, while bridging its outages due to collaborative sensing with cellular connectivity to provide minimum performance guarantee. In contrast, *Scout* aims to improve the network coverage by using the cellular path to only send the uplink traffic (while duplicating a small fraction of downlink traffic). Further, the two designs can complement each other by leveraging the heterogeneous architecture to extend coverage, while using collaborative sensing to select a best whitespace channel for downlink communications.

5.2 WHITESPACE DETERMINATION AND MANAGEMENT APPROACHES

We now discuss prior work in the areas of designing spectrum occupancy databases, improving its underlying propagation models, localizing wireless transmitters, and detecting primary signals.

Whitespace spectrum databases. Spectrum occupancy databases play a key role in determining the available spectrum for whitespace networks. In their recent ruling, various regulatory bodies [25, 94] have mandated the use of empirical propagation models [16] for commercial databases to predict the coverage of primary incumbents. We have found non-negligible spectrum wastage caused by these databases, possibly due to their underlying models that are not able to capture the shadowing and fading effects in an urban environment. To enhance the database performance, Senseless [56] proposes the use of terrain data to augment the Longley-Rice propagation model. This terrain-enhanced model is, however, insufficient to capture diverse effects in a propagation environment such as the shadowing of buildings and other surrounding objects. In addition, Senseless is not designed to predict the

quality of whitespace channels, as it does not have the necessary transmission information (e.g., power) of secondary devices. To overcome Senseless's limitations, V-Scope tunes any given propagation model to local environment using wide-area measurements. Such a measurement-driven model is able to capture those environmental effects along with transmission impact of TV-band devices, allowing the databases to better predict whitespace channels and estimate their individual quality.

V-Scope is also motivated by a spectrum database (WISER [105]) to improve whitespace determination in the indoor environment. It is based solely on measurements collected by spectrum sensors at strategic indoor locations. Targeting at a different scenario, V-Scope combines vehicular measurements with propagation models to enhance the database's performance over the vast outdoor area.

Propagation model enhancement. Prior work has also explored using disparate measurements in signal strength to enhance an empirical propagation model. Caleb et.al [68] has proposed a least-square regression technique to construct an adaptive path loss model for estimating the coverage of a WiMax network. With a similar goal, a geostatistical approach has been later proposed in [69], which applies systematic interpolation on the measurements to build radio coverage maps. Recently, the authors in [79] have proposed a technique to capture a transmitter's distinct characteristics for different radiation sectors. The fitted model is used to predict the coverage of a WiFi mesh network at high accuracy. Regardless of the specific model fitting technique, most of these proposals have a stringent requirement on the measurement locations, which can hardly be guaranteed in vehicular sensing like V-Scope. The unstructured measurement distribution can in turn bias a propagation model fitted by these existing techniques to have large errors at certain locations. To address this challenge, V-Scope uses a robust weighted regression technique that can compensate the variation of measurement density to improve model accuracy.

Device localization. V-Scope leverages a RSSI modeling localization technique to enable existing databases to validate the location of primary and secondary devices when needed. Similar techniques have been used in existing WiFi based systems (e.g., EZ [12] and WiFiNet [75]) for indoor

scenarios, achieving an accuracy of a few meters for pinpointing the location of WiFi transmitters. When applied to localize TV band devices in the outdoor scenarios, such a technique tends to be biased by environment-induced variation (e.g., terrain elevation) that is present in the wide-area measurements. This leads to inaccurate modeling of the large-scale path loss trend, which results in large location errors (up to 100m). V-Scope enhanced the robustness of these approaches for the outdoor scenarios, by selecting measurements in certain radiation sectors that present a good propagation trend, while constructing a sector-specific model to account for both environmental variation and directional antennas used by wireless transmitters.

Recent AoA based localization systems [42, 85, 104] are reported to achieve an even higher accuracy ($< 1\text{m}$) for indoor scenarios. These systems leverage multiple wireless sniffers at different locations, and triangulate a wireless device by measuring the arrival angle of its signals using an antenna array at each sniffer. As a result, these techniques incur the cost for deploying additional wireless sniffers, the protocol overhead for coordinating them, and the performance penalty due to the location uncertainty when deploying these sensors on moving vehicles. In contrast, our RSSI based technique can be performed by individual vehicular sensors without any coordination, thus achieving a much lower protocol overhead and higher robustness. Further, the concept of our sector-specific model fitting technique is beneficial to the AoA approaches as well, by only using measurements at a decent signal strength for localization.

Spectrum sensing and primary signal detection. One alternative approach to determining TV whitespaces is to let whitespace devices to detect primary signals from local spectrum sensing [4], and optionally exchange the detection results with other devices to improve accuracy [45, 84]. Nevertheless, such a technique has rarely been adopted in practice, due to the significant hardware cost and protocol overhead to meet the stringent detection threshold (-114dBm) imposed by regulatory agencies [25]. Such a low detection threshold is to account for the hidden-terminal problem and fading-induced inaccuracy for detecting primary signals, which can hardly be satisfied without high-end spectrum analyzers [4, 56].

Despite its rare use in whitespace networks, spectrum sensing remains as an important function to evaluate the efficacy of spectrum occupancy databases. At its core is a signal detection algorithm to identify primary signals from spectrum measurements. Energy detection is perhaps the oldest and most straightforward algorithm. However, it fails to detect primary signals at the required detection threshold, due to the thermal noise of a spectrum analyzer that can overwhelm a weak primary signal [45]. Subsequent work [45, 59] proposes to use the spectral features of primary signals to improve accuracy, which are more resilient to noise. V-Scope further enhance feature detection with a zoom-in technique to meet the sensing threshold. Such a technique can effectively reduce the noise floor by capturing narrower spectrum snippets around the primary features. It can also estimate the power of these signals based on their zoom-in features, which is needed to refine propagation models.

5.3 PROVIDING VEHICULAR INTERNET CONNECTIVITY

We now discuss prior work in providing vehicular Internet access using cellular technologies, WiFi access, and directional antennas.

Using cellular technology. Existing systems such as MAR [80] and WiRover [36] have explored the use of existing cellular connectivity to relay Internet traffic to moving vehicles. These designs leverage an aggregation proxy, which routes traffic across one or multiple cellular paths to a gateway node on each vehicle. A recent work (PluriBus [50]) has further enhanced vehicular connectivity over WiMax links by performing erasure coding and delay-aware traffic striping at the proxy. Despite a similar use of the traffic aggregation proxy, this dissertation focuses on utilizing the new and additional TV whitespace spectrum for this application. Our proposed heterogeneous architecture can effectively combine TV whitespaces with cellular technologies to overcome the power asymmetry challenge in TV whitespaces. The coordination framework in *Scout* is also unique in providing adaptive base station assignment, seamless client handoff, and flexible channel contention policy.

Using WiFi access. To avoid the spectrum cost, numerous research efforts [6, 7, 22, 61] have explored the use of WiFi access points to provide

vehicular connectivity. CaberNet [22] is an early system leveraging public WiFi APs with streamlined protocols to connect a taxi fleet. A subsequent work (ViFi [7]) improves the vehicular connectivity by enabling each vehicular client to connect to multiple APs using a single radio. Wiffler [6] further leverages cellular connections to bridge the coverage holes in WiFi access, whereas MobiSteer [61] proposes the use of directional antennas to extend the AP's transmission range. Despite numerous optimizations on wireless protocols and transmission hardware, these systems suffer from an inherent bottleneck in the communication range of the ISM band ($< 200\text{m}$). The limited AP coverage causes frequent connectivity losses and client handoffs, degrading the link performance (e.g., averaged 86kbps in [22]).

To circumvent the limitation of WiFi access, this thesis attempts to harness the excellent propagation property in TV whitespaces to match the needs of vehicular connectivity. Further, instead of relying on clients to individually make protocol decisions (e.g., AP association), *Scout* uses a controller to perform centralized routing and contention resolution. This centralized design is particularly efficient for utilizing the scarce network resources for providing vehicular connectivity. But it can hardly be supported by these WiFi based systems, due to the lack of control on wired infrastructure.

5.4 CENTRALIZED WIRELESS ARCHITECTURE

We now discuss related work in centralized wireless architectures used in cellular networks and enterprise WLANs respectively.

Cellular networks. The cellular networks [86, 91] today employ a centralized design of the packet forwarding infrastructure (aka., packet core). In a wired backbone, a few gateways are deployed as aggregation proxies to the external network. They relay Internet traffic via encapsulation tunnels to base stations (called eNodeBs in 4G [86]), which in turn send them to each client (UE). The client handoff is achieved seamlessly by switching encapsulation tunnels between gateways and base stations without interrupting the end applications. While *Scout* uses a similar wired backplane proxied by a central controller, it needs to deal specifically with uplink and downlink traffic through

heterogeneous links, which are used to address the power asymmetry issue in TV whitespaces. Furthermore, unlike cellular networks performing resource scheduling at individual base stations, such a function is performed by our controller that regulates the traffic forwarding rate for each base station. Hence, it can leverage a global view of network condition to maximize network capacity, while supporting customized channel contention policies. Note that the latter feature is especially hard to implement by existing cellular networks for coordinating channel contention among distributed base stations.

Enterprise WLANs. The centralized architecture has been frequently explored in enterprise WLANs [57, 58, 88]. Most of prior research such as Dyson [58] and DenseAP [57], along with commercial endeavors from major WLAN vendors like Aruba and Meru, only focused on centralizing the control and management functions. Example functions include client-AP association, access control, channel allocation, transmit power management, etc. A recent system Centaur [88] further explored a centralized data plane to address the problem of hidden terminals and exposed terminals. Centaur achieves this with a central controller that imposes appropriate delay for packets forwarded to WiFi APs. Built on all the prior work, *Scout* centralizes both the control plane and data plane, with each performing base station assignment and traffic scheduling respectively. This centralized combo is able to achieve a broader set of performance goals — to maximize network capacity, enforce flexible contention policy, and support seamless handoff. Further, most of existing systems are complementary to *Scout* by enhancing other design aspects, such as enforcing access control and mitigating hidden terminals.

5.5 VISION ANALYTIC SYSTEMS

We compare Vigil with prior work in intelligent video surveillance systems, edge computing architectures for mobile vision analytics, and their underlying vision algorithms.

Intelligent video surveillance systems. Many of “smart” video surveillance products today such as Dropcam [21] and SimpliCam [89], rely on a wired network to upload camera feed to the cloud for diverse vision analysis and

persistent storage. A similar architecture has been explored in early research prototypes like IrisNet [30], Bolt [33], and S3 [98] to coordinate a large number of camera sensors. While these systems have addressed various scalability challenges in computation and storage, the wired backhauls ultimately limit their applicability to pervasive surveillance applications, while increasing the deployment cost. To address this limitation, Vigil uses a TV whitespace network to provide ubiquitous surveillance functions, and leverages an edge computing architecture combined with cross-layer scheduling techniques to conserve wireless bandwidth.

Edge computing based analytic frameworks. Edge computing architectures have been increasingly used in vision analytic applications for computation offloading and bandwidth efficiency. For example, Odessa [72] supports interactive perception applications by dynamically offloading certain computation tasks from mobile devices to the cloud. A recent system Gabriel [34] targets a similar class of augmented reality applications with a cloudlet architecture, which adaptively partitions the computation between the network edge and the cloud to achieve low latency and high availability. In contrast to these systems, Vigil focuses on an orthogonal problem of conserving wireless bandwidth, and addresses such challenge through scheduling algorithms based on vision analytics and link capacity estimates reported by edge computing nodes.

Several recent systems [17, 37] have explored the use of edge computing to filter video traffic uploaded to the cloud. Dao et al. [17] proposes a framework that can suppress redundant images uploaded from smartphones by running feature matching algorithms on thumbnails of captured images. Hu et al. [37] proposes a more general technique for traffic filtering, using various metrics such as camera focus, blur effect, and the similarity across successive frames. These systems differ from Vigil in their uploading decisions, which are based solely on the quality and similarity of images, instead of the objects pertaining to a user's query. Vigil further advances the edge computing technology by running object re-identification algorithms to eliminate redundant images captured by multiple cameras, while prioritizing video frames uploaded to the cloud for maximizing their utility under a varying bandwidth constraint.

Vision analytic algorithms. A large body of prior work has leveraged diverse vision algorithms for many applications. For example, Gabriel [34] and Glimpse [35] use a Harr Cascade classifier for face detection. CarSafe [106] and WalkSafe [102] leverage SIFT based object detection algorithms to detect cars and traffic lanes for road safety. In addition, InSight [101] recognizes people based on their clothing patterns, by performing Wavelets transformation on the distribution of color pixels. Vigil is agnostic to the specific vision algorithm, and can complement these systems with efficient frame scheduling decisions, when deployed over wireless networks.

Object re-identification is a popular vision research field for identifying unique objects from different camera views. Prior work [46, 54] has leveraged perceptual hashing techniques to quantify the similarity of images based on their hashed properties. A recent system (sTrack [13]) has leveraged an encryption algorithm to securely construct and match 3D facial models for people re-identification. Most of these techniques have significant computational complexity, resulting in a high processing delay at the cheap edge computing nodes (e.g., several seconds per frame). To reduce the computation latency, Vigil uses a light-weight algorithm [116] that projects all the objects detected from multiple camera views on a common plane, where distinct objects are identified based on a distance threshold. Such a technique has a linear complexity, which is simple enough to be executed by ECN nodes.

6 CONCLUSIONS AND FUTURE WORK

In this thesis, we have shown how to design and implement measurement infrastructures and wireless networks to improve the network connectivity through dynamic spectrum access. We have attempted to optimize three key aspects of this opportunistic access model, i.e., spectrum efficiency, network coverage, and bandwidth utilization. Specifically, we have developed an opportunistic measurement system based on public vehicles to augment the spectrum occupancy databases. We show how the vehicular measurements can be combined with propagation model to enable existing databases to better predict whitespace channels, estimate their individual quality, and validate the location of primary and secondary devices. We have also explored a heterogeneous network architecture and a centralized coordination framework to extend the coverage of whitespace networks for vehicles. Finally, we have developed an edge computing architecture along with cross-layer techniques, which can improve the bandwidth utilization of whitespace networks for pervasive video surveillance functions.

We highlight the main contributions of this thesis in the next section, and then outline future research endeavors.

6.1 CONTRIBUTIONS

We summarize the main contributions of this thesis as follows:

A measurement infrastructure to augment spectrum occupancy databases.

We have presented V-Scope, an opportunistic measurement system that leverages spectrum sensors mounted on public vehicles to collect wide-area measurements at low-cost. We have deployed V-Scope on a single metro bus traveling in and around Madison, WI for a 6-week period. Since the bus operator tends to rotate their buses through multiple routes in the course of each day, we have been able to collect spectrum measurements at more than one million distinct locations over a 120 sq. km. area. Using these measurements, we first show that existing spectrum databases based solely on empirical propagation

models can be quite inaccurate in whitespace determination, causing under-utilization of some whitespace channels over a large area (up to 71% measured locations). We also observe large differences in whitespace channel quality (up to 40dB) at many locations, which are not captured by existing databases, but can significantly impact the performance of whitespace networks. Motivated by these limitations of existing databases, we have developed various model fitting techniques based on an enhanced signal detection algorithm, which can refine various propagation models to accurately predict the signal strength of primary and secondary devices. These models can be used by databases to better determine whitespace channels, estimate their channel quality, and localize primary and secondary devices.

Using a cross-validation approach based on these substantial measurements, we show that V-Scope can help reduce the spectrum wastage for protecting primary incumbents by up to 59% locations. It can also identify all the suitable whitespace channels at 72 – 83% locations under different channel quality constraints. In addition, V-Scope achieves a low error of 16 – 27m in localizing various types of TV-band transmitters in vast outdoor areas (of several sq. km.), improving state-of-the-art localization techniques by 1.2 – 3.5 \times .

A heterogeneous architecture to extend network coverage. We have explored a heterogeneous network architecture to tackle the unique challenge of power asymmetry in whitespaces for providing robust vehicular connectivity. Our system - *Scout* uses the TV whitespace spectrum to send downlink traffic to vehicles, while leveraging the already pervasive cellular connectivity for the uplink communications. Such a design can circumvent the bottleneck in the weak whitespace uplink, thereby allowing each whitespace base station to maximize its downlink coverage. Based on our testbed measurements, *Scout* is able to achieve a 4 \times longer operating range of each base station compared to traditional symmetric networks. The significant improvement in operating range indicates much fewer base stations to be used by *Scout* for a similar network coverage, allowing a substantial reduction in the infrastructure and management cost.

We further developed a channel estimation technique based on a scouting radio to deal with the high feedback delay in the cellular uplink. The scouting

radio can look ahead and collect accurate channel parameters for the rear receiving radio when it eventually reaches the forward post. This technique is shown to significantly improve the accuracy of channel estimation based on the delayed feedback (e.g., $5 \times$ error reduction in estimating packet loss rates). Built on this channel scouting technique, we have developed a number of transmission mechanisms — rate adaptation, inter-packet FEC, and intelligent traffic duplication – to enhance the robustness of individual vehicular links. We further built a central controller for coordinating multiple base stations to collaboratively maximize the network capacity, enforce flexible channel contention policies, and support seamless client handoff.

Based on experiments conducted by driving multiple vehicles for hundreds of miles around our outdoor testbed, we find that *Scout* can enhance the TCP throughput of individual links by $3 - 8 \times$ over alternative SISO and MIMO systems. It can further achieve 73 – 374% gains in the median aggregate network throughput over alternative systems for coordinating multiple base stations. In addition, *Scout* can faithfully enforce various channel contention policies with a median throughput error of 0.2Mbps.

An edge computing architecture to improve bandwidth efficiency. We designed and implemented Vigil, a wireless video surveillance system connected by whitespace networks. Vigil intelligently partitions video processing between edge computing nodes (ECNs) attached to cameras and the cloud to conserve wireless bandwidth. Base on an user query, each ECN selects light-weight algorithms to process the video feed locally. It then reports the analytic results to a cloud-based controller over whitespace links, which pulls a fraction of relevant video frames for deeper analysis. By pre-processing video feed at the network edge, Vigil can avoid the bandwidth usage of uploading irrelevant video frames. The saved capacity is utilized to provide WiFi access to recoup the deployment cost.

To improve the bandwidth efficiency, we have designed the *ops* metric (objects per second) to quantify the significance of a video frame along with its associated bandwidth cost. We then designed a frame scheduling algorithm that leverages this metric to greedily prioritizes video frames, with an attempt to maximize the number of query-specific objects delivered to the cloud, while

minimizing their bandwidth consumption. We also developed a distributed camera collaboration technique to address the inherent inaccuracy of vision analytic algorithms. Such a technique leverages multiple cameras to monitor a common area from different locations, and selects a best subset of frames for uploading to the cloud.

Through our deployments at three sites across two countries, we find that Vigil can support a 5 – 200× larger surveillance area than a traditional approach that stream the entire video over the whitespace network. For a fixed region of coverage and bandwidth constraint, Vigil is able to deliver 25% more objects of interest to the cloud, compared to a baseline system that is agnostic to the relative significance of video frames.

6.2 FUTURE WORK

We believe that this dissertation is successful in developing some important building blocks of measurement systems and network architectures to enhance the wireless connectivity based on dynamic spectrum access. We now discuss potential future research directions in this space.

Leveraging mobile devices to crowdsource spectrum measurements. While V-Scope has some unique advantages to collect measurements in the vast outdoor area (Chapter 2), such an opportunistic wardriving technique has some inherent limitations in the indoor scenarios. To further improve the coverage of V-Scope, we envision an alternative measurement approach of crowdsourcing spectrum analytics from smartphones, tablets and other mobile devices. As the number of these devices exceeds the world population, a crowdsourcing service enabled by them can truly achieve a global reach. Such an approach seems indeed feasible, given that many mobile devices already have a spectral scan functionality built into their WiFi NICs, e.g., the Atheros 9280 family of chipsets. It can provide a unique lens of subcarrier energy samples towards the WiFi spectrum (2.4GHz), and has thus been leveraged in recent work [74, 75] to provide various interesting WLAN management functions, such as detecting non-WiFi interference. By attaching some frequency translation hardware in

front, this WiFi spectrum view can be flexibly shifted to other frequency bands, thereby realizing mobile wide-band spectrum sensing.

In fact, we have made an initial attempt by building a crude spectrum sensing platform [112]. It leverages a frequency translator attached to a commercial WiFi radio, and is able to provide a reasonable spectral view across different frequency bands. The major hurdle of this approach is the coarse-grained spectrum data provided by the WiFi cards, which has orders of magnitude lower resolution than commercial spectrum analyzers, largely degrading the accuracy of signal detection results that V-Scope models hinge upon. Developing robust signal detection algorithms tailored to this coarse-grained spectrum is a major topic of our future work. Additional research also needs to be done in the crowdsourcing architecture and measurement protocols to account for both resource constraints at mobile devices and the usefulness of their measurements.

Regulating dynamic spectrum access in other frequency bands. The measurement architecture and model fitting techniques in V-Scope (Chapter 2) can be extended to predict whitespaces in other frequency bands. Examples include the 5 GHz UNII band [26] and the 3.5 GHz radar band [27]. Unlike the stable TV broadcasts, primary devices in these bands such as radar systems and satellite broadcasts, are highly dynamic in transmission patterns and usually use a substantial power. For example, radar systems can send narrow beams at megawatts power, while constantly rotating their transmission direction to detect movement at a wide angle. Many of such systems are conducting critical military functions such as weapon control and missile guidance, which impose a much stricter protection requirement compared to TV whitespaces. This presents a host of new challenges for managing dynamic spectrum access in these bands. First of all, even detecting these primary signals is very hard due to the large dynamic power range (200 dB) produced by these high-power devices. Adaptive attenuation techniques need to be used to enhance the signal detection technique for preventing strong signals from damaging spectrum analyzers, while ensuring sufficient accuracy for detecting weak signals. The model fitting techniques need to use statistically significant measurements aggregated over sufficient time to tolerate the *temporal variation* in primary signal strength. Additional protection techniques and registration protocols

are required to detect and tolerate the potential roaming of these devices like mobile radar units. In addition, static measurement nodes need to be deployed at strategical locations, and intelligently combined with opportunistic sensors in V-Scope to collect temporal and spatial characteristics of these highly dynamic devices. Building such a scalable measurement infrastructure to efficiently coordinate static and mobile sensors, is a topic for future research.

Building a hybrid network architecture to enhance throughput and coverage.

The heterogeneous architecture in *Scout* (Chapter 3) is effective in extending network coverage using a limited number of base stations. Nevertheless, its symmetric counterpart has the inherent advantage of boosting network throughput with better spatial reuse and streamlined protocols, albeit with many more base stations. Hence, the tradeoff between the two architectures lies in the deployment cost and network performance. To exploit this tradeoff, we can envision a hybrid network that marries both architectures to optimize the coverage and throughput — a homogeneous architecture is used to cover crowded places with many base stations to enhance throughput, whereas a heterogeneous architecture used to provide wide-area coverage with fewer base stations, in a way similar to the concept of macrocell and femtocell used in today’s cellular networks. Such a hybrid network can further adapt the amount of “heterogeneity” on the fly, based on various network factors such as real-time link condition, degree of network contention, device mobility, and the characteristics of end applications. Exploring such a hybrid, reconfigurable whitespace network to support a wide range of applications and usage scenarios, remain as a main topic of our future work.

Building a scalable network coordination framework. *Scout* leverages a central controller to coordinate all the base stations and clients in a whitespace network, which presents a potential challenge for scalability and reliability. To address this challenge, we may distribute certain coordination functions among network nodes. For example, base stations might determine their associated clients by exchanging information with nearby peers. The client handoff might be achieved seamlessly by tunneling network traffic across base stations, instead of forwarding through the controller. In addition, a channel contention policy can be possibly enforced in a distributed fashion by adapting the amount of

random backoff at individual network nodes. All the distributed functions require robust protocols to tolerate network dynamics such as connectivity losses, device failures, and wired congestions. One possible solution might be to use a controller as both a backup for network management and an authority to reconcile inconsistent decisions made by individual base stations.

Apart from reducing the degree of network centralization, we may scale up the central controller using state-of-the-art techniques in distributed systems. For example, many management functions such as base station assignment involves iterative matrix computations, which might be suited for those streaming based big data systems, such as Spark [107] and Heron [47]. It is also desirable to dynamically select appropriate algorithms based on the computing infrastructure, available computing resources, network scales and topologies. One may further scale up the packet forwarding plane by deploying multiple controller instances and performing dynamic load balancing. Such a decision depends on many factors, such as the traffic intensity, the available wired capacity and queuing delay of each controller instance, the priority of end-user applications. Hence, an efficient multivariate load balancing technique is desired to prevent chokepoints at these controller instances. To alleviate the bottleneck of wired connections, a scalable packet forwarding infrastructure might be used, perhaps similar to the software-defined architecture in large data center networks [41, 90].

Extending edge computing to other application domains. Apart from vision analytics, it is possible to extend the edge computing architecture in Vigil (Chapter 4) to many other application domains such as health monitoring, transportation management, speech analysis, and diverse analytic applications of Internet of Things (IoTs). For example, in heart rate monitoring, perhaps the average heart rate is more useful than the instant readings to derive health-related information such as daily calorie consumption and abnormal health condition. Similarly, the average gas consumption and driving speed is more interesting to auto manufacturers than the raw reading reported from vehicular sensors. In many such applications, there exist a capacity gap between the raw data generated at the network edge, and analytics consumed by end applications, which can thus be benefited by pre-processing data at

the network edge. A key challenge in realizing a general analytic architecture is to select appropriate analytic functions for edge computing nodes. Such a decision depends on various factors such as the data characteristics, processing capabilities of edge computing nodes, available uploading bandwidth, and application requirements. These constraints might also change during the course of analysis. Hence, it would be valuable to automatically generate these analytic functions based on application semantics and available computing and network resources. Machine learning techniques can be further used to *learn* the efficacy of multiple candidate analytic functions by sampling real data, while adaptively choosing the best algorithm in real-time. Care also needs to be taken to balance the tradeoff between the analytic accuracy and bandwidth consumption in making such decision. In addition, many wireless devices such as home routers and mobile devices today have general processing capabilities, and can thus serve as edge computing nodes to support many exciting applications, e.g., virtual reality for home entertainment, rich interaction and content sharing among wireless devices, and continuous sensing from wearables. Extending our edge computing architecture to incorporate these heterogeneous computing devices for diverse applications is an interesting area for future research.

A IMPACT OF THIS DISSERTATION

We briefly summarize the publications and the broader impact of this thesis below.

- **V-Scope:** We have deployed V-Scope on a public bus operated by Madison Metro Transit for 6 weeks. Such a deployment has strived to collect measurements at more than one million distinct locations, over a 120 sq. km. area at Madison, WI, USA. Using this large data set comprising both spectrum measurements and the corresponding predictions of a commercial-grade database [92], we have conducted a first-of-its-kind academic study on the efficacy of existing databases for protecting primary incumbents over wide-area. We have also identified significant variation in whitespace channel quality due to various interference sources, which have neither been reported by prior studies, nor been taken account by existing databases. We further used these measurements to refine various signal strength models, which can enable the databases to better predict whitespace channels, estimate their channel quality, and localize primary and secondary devices.

V-Scope was initially published at the HotNets workshop 2013 [108], and the full paper was published at MobiCom 2014 [111], with a best presentation award. The work has also won the first place in the ACM Student Research Competition at MobiCom 2013 [109]. In addition, a variant of this system based on crowd-sourcing mobile spectrum analytics was published at the HotMobile workshop 2014 [112]. After the final publication, we have released our data set to the research community. Our measurement infrastructure has been further leveraged in the Spectrum Observatory Project [51] led by Microsoft Inc to study spectrum utilization in a much broader band (4GHz) beyond TV whitespaces.

- **Scout:** We have proposed a heterogeneous network architecture in *Scout* to address the specific challenge in power asymmetry. Our system leverages the use of additional spectrum available in TV whitespaces

and combines it with the already pervasive cellular networks to provide extensive network coverage for vehicles. It further leverages a scouting radio that is suited specifically for moving vehicles to improve channel estimation under the high feedback delay in the cellular up-link. The ideas have been instantiated in a functional system, and extensively evaluated on a vehicular testbed with more than 500 miles of driving experiments. *Scout* was initially reported at HotMobile workshop 2013 [113], and the full paper was published at MobiSys 2014 [115]. A version of this system was also showcased in a demo session at MobiSys 2013 [114], and the video of this demo is available at http://www.youtube.com/watch?v=_rnzH7owtBw.

- **Vigil:** Vigil has demonstrated the efficacy of an edge computing architecture to improve the bandwidth efficiency of whitespace networks. It strived to provide pervasive video surveillance functions in real-time, using distributed camera sensors connected by whitespace links with limited capacity. To conserve wireless bandwidth, it leverages the edge computing nodes co-located with camera sensors to pre-process their video feeds, while using various algorithms to select and prioritize video frames for uploading to the cloud based on their significance. It also employs a hybrid access network architecture to recoup the deployment cost, by scavenging the saved wireless capacity to provide WiFi access.

Vigil has been deployed in three states across two countries (i.e., WA and WI in USA, and London in UK), and demonstrated significant improvement in bandwidth efficiency under vastly different operational conditions. In particular, one deployment at the Microsoft campus has been up and running over the past 2 years. It is based on a long-distance TV whitespace network [11] with wireless links spanning over 2km, connecting camera nodes at bus stops to assist shuttle scheduling services with various vision analytic functions, such as counting passengers. The work was published at MobiCom 2015 [110], and later covered by several media reports (e.g., New Scientist [62], GeekWire [29]). A video demo of Vigil is available at https://youtu.be/2e6UHeW_xmE.

B PSEUDOCODE OF SIGNAL PROCESSING FUNCTIONS IN SCOUT

We present the pseudocode of individual transmission functions in *Scout* (Chapter 3), which include scouting based channel estimation (ScoutEstimate), rate adaptation (ScoutRate), cross-packet FEC (ScoutFEC), and intelligent traffic duplication over the cellular path (ScoutDUP).

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

$$\mathcal{L}_f, \mathcal{L}_r = \text{ScoutEstimate}(\mathcal{R}, \mathcal{A}_f, \mathcal{A}_r, t, \lambda, v)$$

$$\mathcal{R}_{\text{data}}, \mathcal{R}_{\text{probe}} = \text{ScoutRate}(\mathcal{R}, \mathcal{L}_f, \mathcal{L}_r)$$

$$\mathcal{P}_{\text{wspace}} = \text{ScoutFEC}(\mathcal{D}, \mathcal{L}_f, \mathcal{L}_r, \mathcal{R}_{\text{data}}, \mathcal{R}_{\text{probe}})$$

$$\mathcal{P}_{\text{cell}} = \text{ScoutDUP}(\mathcal{P}_{\text{wspace}}, \mathcal{L}_r)$$
return $(\mathcal{P}_{\text{wspace}}, \mathcal{P}_{\text{cell}})$

```

1: procedure SCOUTESTIMATE( $\mathcal{R}, \mathcal{A}_f, \mathcal{A}_r, t, \lambda, v$ )
2:    $\tau \leftarrow \lambda/v$ 
3:    $\mathcal{E}_f \leftarrow \{P_i.\text{rate} : P_i \in \mathcal{A}_f, t - i \leq W\}$  ▷  $W$  is a time window
4:    $\mathcal{E}_r \leftarrow \{P_i.\text{rate} : P_i \in \mathcal{A}_r, |(t - \tau) - i| \leq W/2\}$ 
5:   if  $\mathcal{E}_r = \phi$  then
6:      $\mathcal{E}_r \leftarrow \{P_i.\text{rate} : P_i \in \mathcal{A}_r, t - i \leq W\}$ 
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  $\mathcal{L}_f(R_i) \leq L_{\text{thresh}}$  and  $\mathcal{L}_r(R_i) \leq L_{\text{thresh}}$  then
15:       $\text{throughput}(R_i) \leftarrow R_i \times (1 - L_r(R_i))$ 
16:    end if
17:  end for
18:   $R_{\text{data}} \leftarrow R_i$  with maximum throughput
19:   $\text{probing\_set} \leftarrow \{R_i : R_i \in \mathcal{R}, R_i > \text{throughput}(R_{\text{data}})\}$ 
20:   $R_{\text{probe}} \leftarrow$  a random  $R \in \text{probing\_set}$ 
21:  return  $(R_{\text{data}}, R_{\text{probe}})$ 
22: end procedure

```

```

1: procedure SCOUTFEC( $\mathcal{D}, \mathcal{L}_f, \mathcal{L}_r, R_{data}, R_{probe}$ )
2:    $L \leftarrow \mathcal{L}_f(R_{data}) \times \mathcal{L}_r(R_{data})$ 
3:    $rr \leftarrow L \times (1 + \text{extra\_redundancy})$ 
4:    $N \leftarrow \text{coherence\_time} \times R_{data} / \text{packet\_size}$ 
5:    $K \leftarrow N \times (1 - rr)$ 
6:    $\mathcal{P}_{wspace} \leftarrow \text{encode}(\mathcal{D}, K, N + 1)$ 
7:   for  $P_i \in \mathcal{P}_{wspace}$  do
8:     if  $i = N + 1$  then
9:        $P_i.\text{rate} \leftarrow R_{probe}$ 
10:    else
11:       $P_i.\text{rate} \leftarrow R_{data}$ 
12:    end if
13:  end for
14:  return  $\mathcal{P}_{wspace}$ 
15: end procedure

```

```

16: procedure SCOUTDUP( $\mathcal{P}_{wspace}, \mathcal{L}_r$ )
17:    $\mathcal{P}_{cell} \leftarrow \emptyset$ 
18:   for  $P_i \in \mathcal{P}_{wspace}$  and  $i \leq K$  do
19:     if  $P_i.\text{rate} = \text{lowest}$  and  $\mathcal{L}_r(P_i.\text{rate}) > L_{dup\_thresh}$  then
20:        $\mathcal{P}_{cell} \leftarrow \mathcal{P}_{cell} \cup P_i$ 
21:     end if
22:     if  $P_i.\text{retransmission} = \text{true}$  then
23:        $\mathcal{P}_{cell} \leftarrow \mathcal{P}_{cell} \cup P_i$ 
24:     end if
25:   end for
26:   return  $\mathcal{P}_{cell}$ 
27: end procedure

```

REFERENCES

- [1] Motion JPEG Video Codec. <http://www.digitalpreservation.gov/formats/fdd/fdd000063.shtml>.
- [2] MPEG4/H264 Video Coding Standard. <http://mpeg.chiariglione.org/standards/mpeg-4>.
- [3] Tun/tap interface tutorial. <http://backreference.org/2010/03/26/tuntap-interface-tutorial/>.
- [4] Bahl, Paramvir, Ranveer Chandra, Thomas Moscibroda, Rohan Murty, and Matt Welsh. 2009. White space networking with WiFi like connectivity. In *Sigcomm*.
- [5] Balakrishnan, Hari, Venkata N Padmanabhan, and Randy H Katz. 1999. The effects of asymmetry on tcp performance. *Mobile Networks and applications*.
- [6] Balasubramanian, Aruna, Ratul Mahajan, and Arun Venkataramani. 2010. Augmenting mobile 3G using WiFi. In *Mobisys*.
- [7] Balasubramanian, Aruna, Ratul Mahajan, Arun Venkataramani, Brian Neil Levine, and John Zahorjan. 2008. Interactive WiFi connectivity for moving vehicles. In *Sigcomm*.
- [8] Bicket, John C. 2005. Bit-rate selection in wireless networks. In *Master's thesis, mit*.
- [9] Byrd, Richard H, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16(5):1190–1208.
- [10] Chan, Mun Choon, and Ramachandran Ramjee. 2005. Tcp/ip performance over 3g wireless links with rate and delay variation. *Wireless Networks*.

- [11] Chandra, Ranveer, Thomas Moscibroda, Paramvir Bahl, Rohan Murty, George Nychis, and Xiaohui Wang. 2011. A campus-wide testbed over the TV white spaces. *MC2R* 15.
- [12] Chintalapudi, Krishna, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. 2010. Indoor localization without the pain. In *Mobicom*.
- [13] Chu, Chun-Te, Jaeyeon Jung, Zicheng Liu, and Ratul Mahajan. 2014. strack: Secure tracking in community surveillance. In *Proceedings of the acm international conference on multimedia*.
- [14] Coifman, Benjamin, David Beymer, Philip McLauchlan, and Jitendra Malik. 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. *Elsevier Transportation Research Part C: Emerging Technologies* 6(4):271 – 288.
- [15] Committee, Advanced Television Systems. 2011. Atsc digital television standard - part 2: Rf transmission system characteristics.
- [16] Damelin, Jack, William Daniel, Harry Fine, and George Waldo. 1966. Development of VHF and UHF propagation curves for TV and FM broadcasting.
- [17] Dao, Tuan, Amit K. Roy-Chowdhury, Harsha V. Madhyastha, Srikanth V. Krishnamurthy, and Tom La Porta. 2014. Managing redundant content in bandwidth constrained wireless networks. In *Acm conext*.
- [18] Dean, RB, and WJ Dixon. 1951. Simplified statistics for small numbers of observations. *Analytical Chemistry*.
- [19] Demers, A, S Keshav, and S Shenker. 1989. Analysis and simulation of a fair queueing algorithm. In *ACM SIGCOMM*.
- [20] Doodle Labs. Broadband tv whitespace transceivers. <http://www.doodlelabs.com/products/radio-transceivers/sub-ghz-range/174-790-mhz-tv-band-devices/>.
- [21] Dropcam. Dropcam website. <https://www.dropcam.com>.

- [22] Eriksson, Jakob, Hari Balakrishnan, and Samuel Madden. 2008. Cabernet: vehicular content delivery using WiFi. In *Mobicom*.
- [23] Erway, JB, and RF Marcia. 2012. Solving limited-memory bfgs systems with generalized diagonal updates. In *Proceedings of the world congress on engineering 2012*, vol. 1, 1–5.
- [24] Federal Communications Commission. 2009. IEEE standard for local and metropolitan area networks.
- [25] Federal Communications Commission. 2010. Unlicensed operation in the TV broadcast bands, second memorandum opinion and order.
- [26] Federal Communications Commission. 2013. Revision of part 15 of the commission’s rules to permit unlicensed national information infrastructure (u-nii) devices in the 5 ghz band.
- [27] Federal Communications Commission. 2015. Amendment of the commission’s rules with regard to commercial operations in the 3550-3650 mhz band.
- [28] Gateworks. Laguna gw2388 router board. <http://www.gateworks.com/product/item/laguna-gw2388-4-network-processor>.
- [29] GeekWire. Microsoft looks to stop bike crashes before they happen, testing minority report-style predictive intelligence. <http://www.geekwire.com/2015/microsoft-looks-to-stop-bike-crashes-before-they-happen-testing-minority-report-style-predictive-intelligence/>.
- [30] Gibbons, Phillip B, Brad Karp, Yan Ke, Suman Nath, and Srinivasan Seshan. 2003. Irisnet: An architecture for a worldwide sensor web. *Pervasive Computing, IEEE* 2(4):22–33.
- [31] Gower, John C. 1971. A general coefficient of similarity and some of its properties. *Biometrics*.

- [32] Gudis, E., L. Pullan, D. Berends, K. Kaighn, G. Van der Wal, G. Buchanan, Sek Chai, and M. Piacentino. 2013. An embedded vision services framework for heterogeneous accelerators. In *Computer vision and pattern recognition workshops (cvprw), 2013 ieee conference on*, 598–603.
- [33] Gupta, Trinabh, Rayman Preet Singh, Amar Phanishayee, Jaeyeon Jung, and Ratul Mahajan. 2014. Bolt: Data management for connected homes. In *Nsdi*.
- [34] Ha, K, Z Chen, W Hu, W Richter, P Pillai, and M Satyanarayanan. 2014. Towards Wearable Cognitive Assistance. In *ACM MobiSys*.
- [35] Han, S, R Nandakumar, M Philipose, A Krishnamurthy, and D Wetherall. 2014. GlimpseData: Towards continuous vision-based personal analytics. In *ACM MobiSys Workshop on Physical Analytics*.
- [36] Hare, Joshua, Lance Hartung, and Suman Banerjee. 2012. Beyond deployments and testbeds: experiences with public usage on vehicular WiFi hotspots. In *Mobisys*.
- [37] Hu, Wenlu, Brandon Amos, Zhuo Chen, Kiryong Ha, Wolfgang Richter, Padmanabhan Pillai, Benjamin Gilbert, Jan Harkes, and Mahadev Satyanarayanan. 2015. The case for offload shaping. In *Hotmobile*.
- [38] IEEE. 1999. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.
- [39] IEEE. 2011. Cognitive Wireless RAN Medium Access Control and Physical Layer Specifications: Policies and Procedures for Operation in the TV Bands.
- [40] IEEE. 2014. Wireless LAN in the TV White Space.
- [41] Jain, Sushant, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al. 2013. B4: Experience with a globally-deployed software defined wan. In *Acm sigcomm*.

- [42] Joshi, Kiran, Steven Hong, and Sachin Katti. 2013. Pinpoint: Localizing interfering radios. In *Nsdi*.
- [43] Karandikar, Shrikrishna, Shivkumar Kalyanaraman, Prasad Bagal, and Bob Packer. 2000. Tcp rate control. *ACM SIGCOMM Computer Communication Review*.
- [44] Kashyap, Anand, Samrat Ganguly, and Samir R Das. 2007. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *Mobicom*.
- [45] Kim, Hyoil, and Kang G. Shin. 2008. In-band spectrum sensing in cognitive radio networks: energy detection or feature detection? In *Mobicom*.
- [46] Kozat, S, R Venkatesan, and M Mihcak. 2004. Robust perceptual image hashing via matrix invariants. In *Icip*.
- [47] Kulkarni, Sanjeev, Nikunj Bhagat, Maosong Fu, Vikas Kedigehalli, Christopher Kellogg, Sailesh Mittal, Jignesh M. Patel, Karthik Ramasamy, and Siddarth Taneja. 2015. Twitter heron: Stream processing at scale. In *Sigmod*.
- [48] Lowe, D. 1999. Object recognition from local scale-invariant features.
- [49] Maechler, Martin, Peter Rousseeuw, Anja Struyf, Mia Hubert, Hornik Kurt, Matthias Studer, and Pierre Roudier. 2014. cluster: Cluster analysis extended rousseeuw et al.
- [50] Mahajan, Ratul, Jitendra Padhye, Sharad Agarwal, and Brian Zill. 2012. High performance vehicular connectivity with opportunistic erasure coding. In *Usenix atc*.
- [51] Microsoft Inc. Spectrum Observatory. <http://observatory.microsoftspectrum.com/>.

- [52] Miu, Allen, Hari Balakrishnan, and Can Emre Koksal. 2005. Improving loss resilience with multi-radio diversity in wireless networks. In *Mobicom*.
- [53] mkomo.com. A History of Storage Cost. <http://www.mkomo.com/cost-per-gigabyte-update>.
- [54] Monga, V, and B Evans. 2006. Perceptual image hashing via feature points: Performance evaluation and tradeoffs. *IEEE Trans. on Image Processing* 15(11):3452–3465.
- [55] Moshnyaga, V, K Hashimoto, and T Suetsugu. 2008. A Hardware Design of Camera-based User’s Presence Detector. In *Ieee international conference on systems, man and cybernetics*.
- [56] Murty, Rohan, Ranveer Chandra, Thomas Moscibroda, and Paramvir (Victor) Bahl. 2012. Senseless: A database-driven white spaces network. *IEEE Transactions on Mobile Computing*.
- [57] Murty, Rohan, Jitendra Padhye, Ranveer Chandra, Alec Wolman, and Brian Zill. 2008. Designing high performance enterprise wi-fi networks. In *Nsdi*.
- [58] Murty, Rohan, Jitendra Padhye, Alec Wolman, and Matt Welsh. 2010. Dyson: An architecture for extensible wireless lans. In *Usenix annual technical conference*.
- [59] Narlanka, S., R. Chandra, P. Bahl, and J.I. Ferrell. 2007. A hardware platform for utilizing tv bands with a wi-fi radio. In *Lanman*.
- [60] Narváez, Paolo, and Kai-Yeung Siu. 1998. New techniques for regulating tcp flow over heterogeneous networks. In *Local computer networks*. IEEE.
- [61] Navda, Vishnu, Anand Prabhu Subramanian, Kannan Dhanasekaran, Andreas Timm-Giel, and Samir Das. 2007. Mobisteer: using steerable beam directional antenna for vehicular network access. In *Mobisys*.

- [62] New Scientist. Intelligent cameras can put an end to always-on surveillance. <https://www.newscientist.com/article/mg22730371-800-intelligent-cameras-can-put-an-end-to-always-on-surveillance/>.
- [63] Nicholson, Anthony J., and Brian D. Noble. 2008. Breadcrumbs: forecasting mobile connectivity. In *Mobicom*.
- [64] Nychis, George, Ranveer Chandra, Thomas Moscibroda, Ivan Tashev, and Peter Steenkiste. 2011. Reclaiming the white spaces: Spectrum efficient coexistence with primary users. In *Conext*.
- [65] Open Gardens. An analysis of early white space network success stories and their future implications. <http://www.opengardensblog.futuretext.com/archives/2012/02/an-analysis-of-early-white-space-network-success-stories-and-their-future-implications.html>.
- [66] OpenVPN Technologies. OpenVPN Virtual Networking Device. <https://openvpn.net>.
- [67] Patra, Rabin K, Sergiu Nedevschi, Sonesh Surana, Anmol Sheth, Lakshminarayanan Subramanian, and Eric A Brewer. 2007. Wildnet: Design and implementation of high performance wifi based long distance networks. In *Nsdi*.
- [68] Phillips, C., D. Sicker, and D. Grunwald. 2011. Bounding the error of path loss models. In *Dyspan*.
- [69] Phillips, C., M. Ton, D. Sicker, and D. Grunwald. 2012. Practical radio environment mapping with geostatistics. In *Dyspan*.
- [70] Płaczek, B. 2010. A real time vehicle detection algorithm for vision-based sensors. In *Computer vision and graphics*, ed. Leonard Bolc, Ryszard Tadeusiewicz, L Chmielewski, and Konrad Wojciechowski, vol. 6375 of *Lecture Notes in Computer Science*, 211–218. Springer Berlin Heidelberg.

- [71] R Core Team. The R Stats Package. <http://stat.ethz.ch/R-manual/R-devel/library/stats/html/optim.html>.
- [72] Ra, M, A Sheth, L Mummert, P Pillai, D Wetherall, and R Govindan. 2011. Odessa: Enabling Interactive Perception Applications on Mobile Devices. In *ACM MobiSys*.
- [73] Raman, Bhaskaran, and Kameswari Chebrolu. 2005. Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks. In *Mobicom*. ACM.
- [74] Rayanchu, Shravan, Ashish Patro, and Suman Banerjee. 2011. Airshark: detecting non-wifi rf devices using commodity wifi hardware. In *ACM IMC*.
- [75] Rayanchu, Shravan, Ashish Patro, and Suman Banerjee. 2012. Catching whales and minnows using wifinet: Deconstructing non-wifi interference using wifi hardware. In *Nsdi*.
- [76] Rayanchu, Shravan, Vivek Shrivastava, Suman Banerjee, and Ranveer Chandra. 2011. Fluid: Improving throughputs in enterprise wireless lans through flexible channelization.
- [77] Reis, Charles, Ratul Mahajan, Maya Rodrig, David Wetherall, and John Zahorjan. 2006. Measurement-based models of delivery and interference in static wireless networks. In *Sigcomm*.
- [78] Rizzo, Luigi. 1997. Effective erasure codes for reliable computer communication protocols. *ACM CCR*.
- [79] Robinson, Joshua, Ram Swaminathan, and Edward W. Knightly. 2008. Assessment of urban-scale wireless networks with a small number of measurements. In *Mobicom*.
- [80] Rodriguez, Pablo, Rajiv Chakravorty, Jon Crowcroft, Julian Chesterfield, and Suman Banerjee. 2004. MAR: A commuter router infrastructure for the mobile Internet. In *Mobisys*.

- [81] Rousseeuw, Peter J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*.
- [82] Satyanarayanan, M, P Bahl, R Caceres, and N Davies. 2009. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing* 8(4):14–23.
- [83] Schulman, Aaron, Vishnu Navda, Ramachandran Ramjee, Neil Spring, Prahlad Deshpande, Calvin Grunewald, Kamal Jain, and N. Padmanabhan, Venkata. 2010. Bartendr: a practical approach to energy-aware cellular data scheduling. In *Mobicom*.
- [84] Sen, Sayandeep, Tan Zhang, Milind Buddhikot, Suman Banerjee, Dragan Samardzija, and Susan Walker. 2012. A dual technology femto cell architecture for robust communication using whitespaces. In *Dyspan*.
- [85] Sen, Souvik, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. 2013. Avoiding multipath to revive inbuilding wifi localization. In *Mobisys*.
- [86] Sesia, Stefania, Issam Toufik, and Matthew Baker. 2009. *Lte: the umts long term evolution*. Wiley Online Library.
- [87] Shreedhar, M, and G Varghese. 1995. Efficient fair queuing using deficit round robin. In *ACM SIGCOMM*.
- [88] Shrivastava, Vivek, Nabeel Ahmed, Shravan Rayanchu, Suman Banerjee, Srinivasan Keshav, Konstantina Papagiannaki, and Arunesh Mishra. 2009. Centaur: Realizing the full potential of centralized wlans through a hybrid data path. In *Mobicom*.
- [89] Simplicam. Simplicam website. <https://www.simplicam.com/>.
- [90] Singh, Arjun, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, et al. 2015. Jupiter rising: A decade of clos topologies and centralized control in google’s datacenter network. In *Acm sigcomm*.

- [91] Smith, Clint, and Daniel Collins. 2001. *3g wireless networks*. McGraw-Hill.
- [92] Spectrum Bridge Inc. Spectrum bridge database. <http://www.spectrumbridge.com/Home.aspx>.
- [93] The Madwifi Project. Ath5k driver. <http://madwifi-project.org/wiki/About/ath5k>.
- [94] The Office of Communications. 2015. Decision to make the wireless telegraphy (white space devices).
- [95] Theodoridis, T, G Link, N Vijakrishnan, M Irwin, and W Wolf. 2004. Embedded hardware face detection. In *IEEE conf. on vlsi design*.
- [96] ThinkRF. The thinkrf wireless signals intelligence platform. <http://www.thinkrf.com/>.
- [97] Thornycroft, Peter. 2007. Designed for speed: Network infrastructure in an 802.11n world. In *Aruba networks white paper*.
- [98] Tian, Y.-L., L. Brown, A. Hampapur, M. Lu, A. Senior, and C. Shu. 2008. IBM Smart Surveillance System (S3): Event Based Video Surveillance System with an Open and Extensible Framework. *Mach. Vision Appl.* 19(5-6):315–327.
- [99] Ubiquiti Inc. Datasheet of ubiquiti sr71-a mimo cards. http://dl.ubnt.com/sr71a_datasheet.pdf.
- [100] Vaidya, Nitin, Anurag Dugar, Seema Gupta, and Paramvir Bahl. 2000. Distributed fair scheduling in a wireless lan.
- [101] Wang, He, Xuan Bao, Romit Roy Choudhury, and Srihari Nelakuditi. 2013. Insight: Recognizing humans without face recognition. In *Acm hotmobile*.
- [102] Wang, Tianyu, Giuseppe Cardone, Antonio Corradi, Lorenzo Torresani, and Andrew T. Campbell. 2012. WalkSafe: A Pedestrian Safety App for Mobile Phone Users Who Walk and Talk While Crossing Roads. In *Acm hotmobile*.

- [103] Wong, Starsky H. Y., Hao Yang, Songwu Lu, and Vaduvur Bharghavan. 2006. Robust rate adaptation for 802.11 wireless networks. In *Mobicom*.
- [104] Xiong, Jie, and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *Nsdi*.
- [105] Ying, Xuhang, Jincheng Zhang, Lichao Yan, Guanglin Zhang, Minghua Chen, and Ranveer Chandra. 2013. Exploring indoor white spaces in metropolises. In *Mobicom*.
- [106] You, Chuang-Wen, Nicholas D. Lane, Fanglin Chen, Rui Wang, Zhenyu Chen, Thomas J. Bao, Martha Montes-de Oca, Yuting Cheng, Mu Lin, Lorenzo Torresani, and Andrew T. Campbell. 2013. CarSafe App: Alerting Drowsy and Distracted Drivers Using Dual Cameras on Smartphones. In *Mobisys*.
- [107] Zaharia, Matei, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, and Ion Stoica. 2013. Discretized streams: Fault-tolerant streaming computation at scale. *SOSP*.
- [108] Zhang, Tan, and Suman Banerjee. 2013. Inaccurate spectrum databases?: Public transit to its rescue! In *Hotnets*.
- [109] Zhang, Tan, and Suman Banerjee. 2013. V-scope: an opportunistic wardriving approach to augmenting tv whitespace databases. In *Mobicom*.
- [110] Zhang, Tan, Aakanksha Chowdhery, Paramvir Victor Bahl, Kyle Jamieson, and Suman Banerjee. 2015. The design and implementation of a wireless video surveillance system. In *ACM MobiCom*.
- [111] Zhang, Tan, Ning Leng, and Suman Banerjee. 2014. A Vehicle-based Measurement Framework for Enhancing Whitespace Spectrum Databases. In *Mobicom*.
- [112] Zhang, Tan, Ashish Patro, Ning Leng, and Suman Banerjee. 2015. A wireless spectrum analyzer in your pocket. In *ACM HotMobile*.

- [113] Zhang, Tan, Sayandeep Sen, and Suman Banerjee. 2013. Scout: An asymmetric vehicular network design over TV whitespaces. In *Acm hotmobile*.
- [114] Zhang, Tan, Sayandeep Sen, and Suman Banerjee. 2013. Video streaming using whitespace spectrum for vehicular applications. In *Mobysis video session*.
- [115] Zhang, Tan, Sayandeep Sen, and Suman Banerjee. 2014. Enhancing vehicular internet connectivity using whitespaces, heterogeneity, and a scouting radio. In *Mobisys*.
- [116] Zhang, Zhong, Andrew Scanlon, Weihong Yin, Li Yu, and Pe'ter L. Venetianer. 2012. Video surveillance using a multi-camera tracking and fusion system. In *M2sf2*.