On Selected Problems in Backscatter Networks and Quality of Experience in Networked Applications

A Dissertation Presented

by

Jihoon Ryoo

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2017

Stony Brook University
The Graduate School

Jihoon Ryoo

We, the dissertation committee for the above candidate for the
Doctor of Philosophy degree, hereby recommend
acceptance of this dissertation.

Samir R. Das, Dissertation Advisor
Professor, Department of Computer Science

Himanshu Gupta, Chairperson of Defense
Associate Professor, Department of Computer Science

Aruna Balasubramanian, Committee Member
Assistant Professor, Department of Computer Science

Petar M. Djurić, External Committee Member
Professor, Department of Electrical and Computer Engineering

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

On Selected Problems in Backscatter Networks and Quality of Experience in Networked Applications

by

Jihoon Ryoo

Doctor of Philosophy

in

Computer Science

Stony Brook University
2017

Wireless communication uses valuable energy especially because it is primarily done through battery-driven mobile devices. How we use energy revenue is a vital issue because people depend on this mode of communication for a variety of reasons, including personal, social, economic, and educational. When devices use energy inefficiently, they drain resources too quickly, resulting in network disconnection. In conventional wireless networks that continue to be the norm, devices transmit information through active radio, costing the major portion of energy in the transmission. To remove energy hungry active radio, the industry introduced RFID (radio frequency identification) systems which built on the backscatter communication mechanism. Through the backscatter mechanism, a power greedy active transmission is simplified to a passive reflection – ride on the existing carrier signal.

Building on passive backscattering approach, the first half of the dissertation makes two main contributions. First, it presents the first multi-hop backscatter tags that can successfully communicate in the presence of structural obstacles, and this backscattered signal utilized in the form of RF sensor. Therefore, the new system resolves scalability issue in RFID and ultimately provide activity signatures for human motion analytic. Second, we introduce an adequate localization technique for the backscatter based devices. The new system presented a phase-based ranging technique and demonstrated on application to shopping cart localization.

In the latter half of the dissertation, human's perception on web applications is investigated toward identifying the best quality of service and experience (QoS/QoE), the best satisfaction under the given network revenue. We humans see only a tiny region at the center of their visual field with the highest visual acuity, a behavior known as foveation. Visual acuity reduces drastically towards the visual periphery. Even the contents are served on its highest quality; humans cannot perceive as it served. Humans only perceive and recognize the small portion of contents. Because of human's characteristics, resources are oversupplied or misused.

We prioritize the contents for the best use of the same given network resource, yet serve better service and experience. Essentially, we prioritize the web contents based on the user's gaze information, in terms of the real-time gaze feedback and previously learned scan path patterns. Through our prioritized system, we can achieve to the quality level that contemporary service unable to reach. Our system demonstrated higher resolution where contemporary service can only serve medium or low resolution for the Internet streaming service, and validate the faster web page perception than contemporary web page service.

Thus, the latter part of the dissertation concludes by highlighting two major achievements. We first present layered Internet video streaming service which serves the best quality in terms of user perception yet save Internet traffic. Next, we introduce a reprioritized web page service in learned user's visual scan paths. The user's perception of page load time reduced 17 percent on average.

In this dissertation, we propose technical solutions to realize communication in lean resource environment in the first two chapters, then propose the best use of tightly given resources in the latter two chapters.

Dedicated to my Love

Dissertation Acknowledgements

This dissertation is the result of collaboration. I have a deep appreciation for the coworkers who have helped me produce this dissertation over the last six years.

In the most direct sense, *Akshay Athalye* contributed to backscatter works and gave me detailed feedback on my broader research agenda. Both *Jinghui Jian* and *Yasha Karimi* are collaborators on backscatter communication papers. Both deserve credit for delivering the product with me. I am looking forward to collaborating with each of you for coming years.

I would not have accomplished a Ph.D. if *Samir R. Das* had not provided me the opportunity, the guidance, and the support to succeed at every stage of my Ph.D. He challenged me to solidify my ideas and taught me the importance of depth and strategy. I wish I can accomplish a small piece of what he has achieved. I cannot express my thankfulness. Also thanks to advisors of my work that set the stage and provided the tools for this dissertation. This includes *Milutin Stanacevic, Petar Djurić* and *Himanshu Gupta*.

For the gaze-assisted work, I must thank *Conor Kelton* and *Aruna Balasubramanian* for their dedication and countless contributions to our masterpiece on NSDI achievement. Concerning this dissertation, Conor deserves particular recognition for the work he did to write the code, organize the document and Webpage. I must also acknowledge the computer vision group (*Kiwon Yun, Dimitris Samaras* and *Greg Zelinsky*). Their valuable feedback and contributions turn my idea into the product.

Thanks to *Hongjin Lee* and *Chanki Jung* for volunteering to write the software that collected RFID data. If they had not stepped forward, the RFID work could not have published. I also need to thank my friends in computer network group who helped me enormously and always encouraged me to push one more step. You *Ayon Chakraborty, Arani Bhattacharya, Vasudevan Nagendra, Mallesham Dasari* and *Sohee Kim Park* are my most precious asset that I earned in Stony Brook.

Previously Published Material

Chapter 3 revises a previous publication [131]: Jihoon Ryoo, and Samir R. Das. Phase-based Ranging of RFID Tags with Applications to Shopping Cart Localization. *In Proceedings of ACM MSWiM*, 2015.

Chapter 4 revises a previous publication [132]: Jihoon Ryoo, Kiwon Yun, Dimitris Samaras, Samir R. Das, and Greg Zelinsky. Design and Evaluation of a Foveated Video Streaming Service for Commodity Client Devices. *In Proceedings of ACM MMSys*, 2016.

Chapter 5 revises a previous publication [84]: Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R. Das. Improving User Perceived Page Load Times Using Gaze. *In Proceedings of Usenix NSDI*, 2017.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The “Internet of Things” (IoT) envisions an environment where any physical object could be sensor-enabled, networked and connected to the cyberspace for smart control and decision making. However, the traditional sensor’s mode of connection is inadequate to deliver on this vision. The primary reason is that a standard motes-class device (e.g., enabled by the low-power microcontroller and RF transceiver combination) is too power hungry – mainly due to the use of an active radio transceiver on board. Most of the commodity radio transceivers catering to the embedded RF devices market consume in the order of mW even when idle (listening) and in the order of 10s of mW when actively transmitting or receiving. This is at least two orders of magnitude higher than the power consumed by low-power microcontrollers in equivalent states (CPU idle and CPU active states). The radio power is one of the main reasons why – in spite of a decade and half of research in sensor networks – the IoT vision is yet to make inroads in practical systems.

In this dissertation, we adopt the backscattering mechanism to reduce energy consumption for network devices. Therefore, network devices able to operate in solely depend on harvestable energy. In the latter part of the dissertation, we take the approach of objects prioritization in order to optimize the user experience on the Web page and the video streaming.

Adopting backscattering for radio equipped devices

One approach to eliminating the power hungry active radio from the system that will be studied in this dissertation is to use a communication paradigm where devices communicate via backscattering using harvested power from an external RF source. This brings down the size, power and cost to an unprecedented level presenting a unique opportunity in terms of both usability and scalability. Backscattering works by modulating a continuous wave RF signal (transmitted from an external source) incident on the antenna of the device. The modulation is achieved simply by changing impedance levels seen by the antenna, thereby changing its reflection coefficient. The minimal intelligence needed to achieve this requires very little power and this power can be supplied by the RF signal itself. If designed right, this power is even sufficient to power the lowest power microcontrollers available today. The most widely used embodiment of this technology is in RFID (Radio Frequency Identification) [53], where this simple device is known as the 'tag,' to which external RF signal is provided via a relatively higher power embedded computer, possessing an active radio, called the reader. The reader is also able to receive and interpret the modulated signal backscattered by the tag and provides specific instructions to the tag about how to respond. But, as studied in this project, since this reader-based system, while widely used, still suffers from scalability issues as tags can only talk to the readers, there must be enough readers to 'cover' all tags in an environment and readers are indeed relatively high-power, significantly intelligent and expensive devices.

Enabling the IoT vision for energy limited devices

The general goal of this dissertation is to extend backscattering to its logical extreme so that we can eliminate the reader from the system. The idea is to enable the tags themselves to read and interpret the backscattered communications from other neighboring tags and possibly backscattering the information back, thus producing a form of multihop network of tags. While the general feasibility of this form of tag-to-tag communication has been demonstrated [95, 96, 120], two other advancements can add to the capabilities of these tags. First, low-power sensors can be added to the tags to go beyond mere identification [191]. Second, current generation low power microcontrollers can impart certain amount of 'computational' ability on the part of the tags ('Computational RFID' or CRFID) [152]. These advancements when integrated together with appropriate platform-level optimizations can turn the backscattering tags into a multihop network of tiny, battery-less, sensor-enabled compute platforms. To bring the vision of IoT closer to reality, we posit that this is the best way forward for significant advancements, as the use of active radios has largely failed so far.

The tags indeed require external RF sources (we call them 'exciters') for powering and backscattering. However, these sources simply feed RF power with zero intelligence devices in our system. Unlike standard RFID readers, these external sources could be much fewer in number as there is no reverse link under consideration. With no intelligence, they are of low cost and can be simply integrated into the infrastructure of the future as a standard practice. Also, it is certainly feasible to use ambient RF signals as a recent work has shown [95], e.g., TV or cellular signals. In this case, these TV towers or cellular transmitters take up the role of the exciters. Because using ambient signals presents a tradeoff, such as limiting the communication ability of devices. It is known that harvested power from these signals could be small depending on the location thus limiting the usable range of the backscattering links. As such, our design is flexible enough to embody both possibilities and exploring the tradeoffs between excitation power and deployment density.

Enabling IoT devices to RF-sensor

We present *BARNET* (Backscattering Activity Recognition NEtwork of Tags), a network of passive RF tags that use RF backscatter for tag-to-tag communication. *BARNET* not only provides identification of tagged objects but also can serve as a 'device-free' activity recognition system. *BARNET*'s key innovation is the concept of backscatter channel state information (BCSI) which can be measured via systematic multiphase probing of the backscatter link entirely by passive techniques. Changes in BCSI provide signatures for different activities in the environment that can be learned using suitable machine learning tools. We develop prototype *BARNET* hardware and firmware using COTS components and also provide evaluations for future ASIC implementations that can run entirely using harvested power. We use the COTS prototype to evaluate the tag-to-tag communication and activity recognition performance. We show that *BARNET* can recognize human daily activities with average error around 10%. Overall, *BARNET* uses passive tags to achieve the same level of performance as systems that use powered, active radios.

In the second half of the dissertation, the use of gaze feedback to improve user experience for Web access and streaming video is studied. Web page access and video streaming are two of the most popular applications on the global Internet. Web page access and video streaming dominate today's Internet experience. For example, over 80% of Internet traffic is predicted to be video traffic by 2019 [42]. Similarly, close to a billion Web pages exist on the Internet and the number of Web page passing through Google search alone is over 3.5 billion searches per day and 1.2 trillion searches per year [77]. The dissertation addresses the quality of user experience as a key issue because Web and video applications impact almost every facet of our life including education and research, media and entertainment, commerce, social networks, and even healthcare.

There are many techniques to improve the quality of service; however, user experience driven studies are deficient in the networking community, a user experience must be considered on top of the design. Because a user is the ultimate service recipient. Web contents service addresses another major problem in the field of Web browsing and streaming content space with respect to user experience. Although the network and server technologies that support these applications are mature, basics of these technologies have been developed without any regard to quality of experience. When the application is demanding (e.g., high resolution video or complex Web pages) or network is challenged (e.g., congested cellular links), the quality of experience suffers even in cases where much better quality could be delivered. Overall, the networking community lacks a good scientific grasp on the issue of quality of experience even though there is a broad expectation that this understanding will lead to network architectures and protocols that serve the user better. The central issue is that the quality of experience is subjective and requires understanding of cognitive aspects. Characterizing it also requires constant user feedback that is hard to gather automatically and unobtrusively.

Thus, in this dissertation, we make the case for using gaze tracking as a central tool to improve quality of experience of video and Web applications, bringing together research focusing on technological improvements and studies focusing on user experience. One of the most effective methods of bringing the two domains in context of each other is the research using gaze tracking. It is well known that user's gaze in terms of fixation, dwell time, blink rates, and search patterns, correlate well with user attention [31, 161]. Google has recently patented technology to use Google Glass to monitor user engagement [115]. Both Web and video being visual media, gaze provides a strong mechanism to characterize user's focus of attention, and gaze information can be exploited gainfully to both characterize and deliver better quality of experience. With advances in imaging, gaze tracking now can be performed using commodity technologies that can be integrated into end users' devices (§ 4). This makes gaze tracking an attractive solution to characterize and improve user experience. In the following we introduce specific quality of experience issues related to video and Web access.

Improving perceptual quality for Internet video streaming

When addressing the quality of perception in video streaming, it is quite different from the service quality. This dissertation focuses on how the user perceive on the quality of video through various user studies. Various industry analysts [124, 125] report that over 70%

of the Internet traffic during evening peak hours is from streaming video services, such as Netflix and YouTube. The average quality/resolutions of available videos are constantly improving. However, there is hardly enough bandwidth available. Roughly speaking, average bandwidth available per household in many developed countries is barely enough to stream only two HD quality videos concurrently [113, 155]. While content providers are intent on making available higher resolution 4K videos for streaming and display prices are falling fast, no ISP currently can sustain the bandwidth needed for such videos at scale [128]. This is even at only a slow frame rate (30 fps) and with the most aggressively compression. Similar bandwidth woes pervade mobile platforms albeit at a different scale. More and more media are now consumed on mobile devices and often outside the home/work networks. Both cellular providers and ISPs employ different rate plans and data caps making significant media consumption very expensive for the end user. They are also widely understood to employ various forms of traffic shaping and differentiations to reduce stress on their networks (e.g., [49]) that in turn affects end users' quality of experience. The situation is worse in emerging markets.

We propose to exploit gaze to improve the perceptual quality of the video without having to add more capacity in the network. The essential idea is to develop a variable resolution video streaming techniques that compresses the video based on where the user is looking with their central or foveal vision. This exploits a significant property of human visual system (HVS) where only a very small (about 2 degree) part at the center (fovea) of the visual field is perceived with the best possible acuity [44]. The acuity falls off rapidly away from the center. Thus, high-resolution images falling on the periphery of the visual field are wasted. This capacity is better utilized to provide even better resolution at the central vision of the user. While a number of techniques for achieving a foveated video compression by exploiting the aforementioned property of HVS does exist (see, e.g., [49]), existing methods of video streaming have failed to exploit these advances. The key reason is that it is usually unknown where a persons gaze will be pointing while watching a video.

This generally requires a continuous feedback of the gaze information to the video server so that the server can deliver an appropriately compressed video taking into account the gaze point and available bandwidth estimate into consideration. Determining gaze points requires an ‘eye tracker.’ While typically not a commodity device, recent advances in imaging makes it possible to use ordinary webcams or similar sensors to perform eye tracking with a reasonable accuracy [87, 117]. Our goal in this dissertation is to develop scalable solutions for foveated video streaming on the Internet and address related challenges.

Improving Perceptual Quality of Web Page Loads

Another major goal of this study is to identify how the user perceives the quality of Web browsing. Web page load performance is becoming critical for everyone in the Web ecosystem: the content providers, the service providers, and the users. For instance, for content providers such as Walmart, reducing page load time (PLT) by 1 second causes an up to 2% increase in conversations [30], Google, increasing PLT from 0.4 seconds to 0.9 seconds decreased traffic and ad revenues by 20% [151], Shopzilla, reducing the page load time (PLT) from 6 seconds to 1.2 seconds increased revenue by 12% [29]. Similarly, some studies show that when Web pages take longer than 10 seconds to load, the user will likely abandon the page [179]. Despite the importance of Web page load times, their performance continues to be generally poor [38, 180]. One critical problem is that the performance metrics used to measure the page load time (PLT) do not correlate well with user experience [38, 122, 147]. This is symptomatic of a flawed community-wide design approach, where the community optimizes a network metric that is divorced from user experience. All existing Web optimizations target the traditional PLT metrics; but the optimizations may not improve users quality of experience. We characterize the latter using a new perceptual metric that we call user perceived page load time or uPLT.

This dissertation provides scientific basis through large scale user studies to characterize the relationship between user's gaze and uPLT. Such a study is challenging because of the wide variety of Web pages. The gaze patterns and uPLT may depend on a large number of factors, such as Web page type, the intent of the user, the Web page familiarity, network and the client device characteristics. We use the findings from the above characterization to design user-driven Web optimizations that will be driven by gaze feedback. Our study (§ 5.2) shows the gaze patterns are unique to the user. While the gaze pattern of a user for a given page is consistent, they different across users and Web pages. This result motivates us to design user-driven Web optimizations. Our goal in this dissertation is to design and implement such Web optimizations and address related challenges.

Given the slow speeds experienced by most people due to the explosion of demand of network budget, our research operates within available resource in order to optimize the speed and user experience by prioritizing the loading objects on the video streaming and the Web page.

Chapter 2

Activity Recognition using Passive Backscattering Tag-to-Tag Network

We imagine a future where all living beings and physical objects are recognized inside a cyber environment. This enables automated systems to query and reason about the environment and perform analytics. Fundamental technologies to enable this vision have so far progressed in two rather different directions: i) development of batteryless, RF-powered tag-like devices with innovative platforms and novel communication techniques [82, 95, 120] and ii) 'device-free' activity recognition [67, 126, 174, 190] for inferring activities via analysis of RF signals reflected from objects and living beings in the surrounding environment. The term 'device-free' signifies that the techniques do not require these objects and beings carry any device. The goal of this work is to marry these two disparate technologies on a single platform thus enabling activity recognition using a network of passive tags. We call this *BARNET* ('B'ackscattering 'A'ctivity 'R'ecognition 'Ne'twork of 'T'ags). *This integration is challenging as RF-powered battery-less tags must use 'passive' techniques*

Acknowledge *Jinghui Jian and Yasha Karimi* who provide prototype PCB board and analyze its physical characteristics.

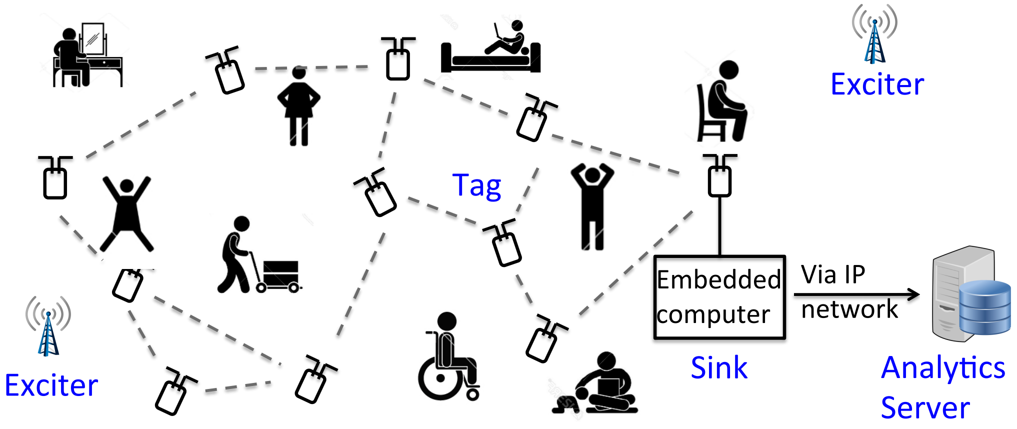


Figure 2.1: Overview of the *BARNET* tag network performing RF-based activity sensing. The dotted lines are tag-to-tag backscatter links that are both used for data communication as well as channel measurement.

to perform a level of RF processing that has so far been achieved only by using high-power active radios.

While activity recognition using RF techniques has been gaining increasing research interest lately (see Section 2.1), most existing techniques are centered around high-power active radios which limits the scalability of these approaches. The uniqueness of our approach *brings this technique to an extreme low-power regime*, where batteryless RF tags themselves are able to capture some fundamental characteristics of the wireless channel without the need for readers or access points. The proposed RF tags are also capable of multi-hop tag-to-tag communication [95, 110] allowing for scalable deployments.

Passive RF Sensing of Backscatter Channel

BARNET is a network of passive batteryless tags with a limited computational ability that i) directly communicate among themselves via *backscatter modulation* of an external RF signal and ii) can measure and record variations in the backscatter wireless links. The envisioned system is presented in Figure 2.1. The tags are attached to everyday objects identifying them, much like in RFID systems except that conventional RFID reader devices are not needed. Tags could also be part of the building infrastructure such as wall or ceiling panels. The tags form a multi-hop network using tag-to-tag backscatter links (dotted lines). Exciters supply RF power and provide the signal used for backscattering, but otherwise have no intelligence. If strong enough ambient RF signals are available (e.g., TV signals or WiFi [28, 82, 95]) they can proxy for exciters. Thus, intentionally deployed exciters are not critical for the fundamental technique described here.[1] Some tags (sink nodes) are attached to embedded platforms that are connected via IP networks to an analytics server that executes necessary machine learning processes.

The multi-path wireless channel (*backscatter channel*) between two passive tags undergoes changes related to dynamic alterations in their vicinity. We exploit a fundamental characteristic of this tag-to-tag backscatter link: amplitude of the received backscatter at an Rx tag depends upon the propagation delay (or phase) of the channel between the two tags. Then, by systematically varying the phase of the Tx signal and quantifying the Rx signal amplitude for the various phase values, we are able to detect patterns related to specific activities in the environment. We demonstrate this concept via mathematical analysis (Section 2.2) and experiments (Sections 2.4 and 2.4.3). Our proposed technique for passive channel measurement relies on quantized probing of the channel using a range of different Tx backscatter phases and quantized measurements of the signal amplitude on the Rx tag using very low-power techniques. This innovation is central to the passive tags that form the building blocks in *BARNET*. Together with the communication and measurement protocols, innovative hardware design and backend analysis *BARNET* forms a truly ubiquitous and scalable passive tag network that can simultaneously provide identification and activity recognition ability.

The contributions of this work are summarized as follows:

[1] We must note, however, that while recent literature [28, 82, 95] has promoted use of ambient signals, the actual power levels used in these papers are unusually high.

- **Passive techniques for channel measurement (Section 2.2):** We develop the concept of backscatter channel state information (BCSI). We show how BCSI can be measured by systematic, multiphase probing of the backscatter channel using a Tx backscatter pilot (BP). We develop the necessary protocol support for transmitting the BP and recording the BCSI.

- **RF-powered tag hardware (Section 2.3):** We develop a tag design capable of i) robust tag-to-tag backscatter communication and ii) BCSI measurement of the backscatter signal based on above concept. We describe power harvesting/management issues for a completely RF-powered operation of the tags. We also describe the COTS prototype that contains all necessary components except the power harvester/-manager.

- **Experimental demonstration (Sections 2.4 and 2.4.3):** We demonstrate robust tag-to-tag communication and relaying at long ranges. We also demonstrate *BARNET*'s ability to recognize human activities via a user study in our lab.

2.1 Background and Related Work

The BARNET vision draws on two fundamental advances in recent years. We describe these advances below to prove a context for our work.

2.1.1 RF-Powered Tags

RF-powered tags use RF power harvesting and backscatter communication to deliver enough functionality to communicate small amounts of information. Backscattering works by modulating an external RF signal incident on the antenna of the device. The modulation is achieved simply by changing impedance levels seen by the antenna, thereby changing its reflection coefficient. This requires very little power which, with appropriate design, can be supplied by the external RF signal itself. Today, the most widely used embodiment of this technology is in RFID (Radio Frequency IDentification) [50,57]. In RFID, the external RF signal is provided via a relatively higher power embedded computer, possessing an active radio, called the 'reader.' The reader also receives and interprets the modulated signal backscattered by the tag and issues specific instructions to the tag about how to respond.

RFID has long been standardized and is now widely deployed in logistics and inventory applications to perform identification and tracking [53]. Variations of RFID exists as research platforms, including tags with sensors [46, 189], tags with computational ability (Computational RFID) [127, 189], etc. Two recent innovations have improved the possibility of ubiquitous deployment of such tags. They include 1) use of ambient RF signals (e.g., TV or WiFi) to provide the external RF signal and/or to power the tag [82, 95] and 2) tag-to-tag backscatter communications [26, 95, 110]. Use of ambient RF signals mean that tags can be deployed anywhere with no other infrastructure support necessary. Tag-to-tag backscatter implies that there is no need to have high-powered reader devices in the neighborhood to read the tag signals; the tags themselves can read and in turn relay the information using multi-hop routing. This enables highly scalable deployment. Our work benefits from existing work on tag-to-tag backscatter, but enhances the basic techniques.

2.1.2 'Device-free' Activity Recognition

There is a growing body of work that measures or characterizes the RF wireless channel impacted by human activities around it. The channel changes closely reflect the activity and thus provide the necessary signature for activity recognition. Techniques vary depending on what exact RF technology is used or what property of the channel is measured and how. Early work has focused on 802.5.4 links with sensor mote class devices [141]. Recent work has started focused on WiFi due to its popularity. Within WiFi, several techniques focus on RSS of the entire channel (e.g., [19, 142]) and many others focus on RSS of individual sub-carriers of the OFDM WiFi channel (CSI or channel state information). This class of techniques has been shown to be successful in a range of applications such as counting people [182], lip-reading [164], recognizing various gestures [126, 164, 190], various human daily activities [67, 174], etc. Recently, channel modeling techniques have been used to bolster the performance of such techniques [170]. Various other RF technologies (e.g., 60 GHz) have been used as well using radar principles for similar applications [20, 169]. A central theme in all these works is the dependency on high-powered active radios for complex signal processing needs, a limitation *BARNET* removes.

Activity recognition has been successful in RFID domain as well [73, 92, 118]. However, this body of work relies on processing on the reader side (again an active device). The only work in our knowledge that exploits passive processing on tags is AllSee [83]. However, AllSee is limited in capability as it characterizes the channel in an elementary level, needs another smart device for each tag, and does not benefit from multiplicity of links in a tag network such as *BARNET*.

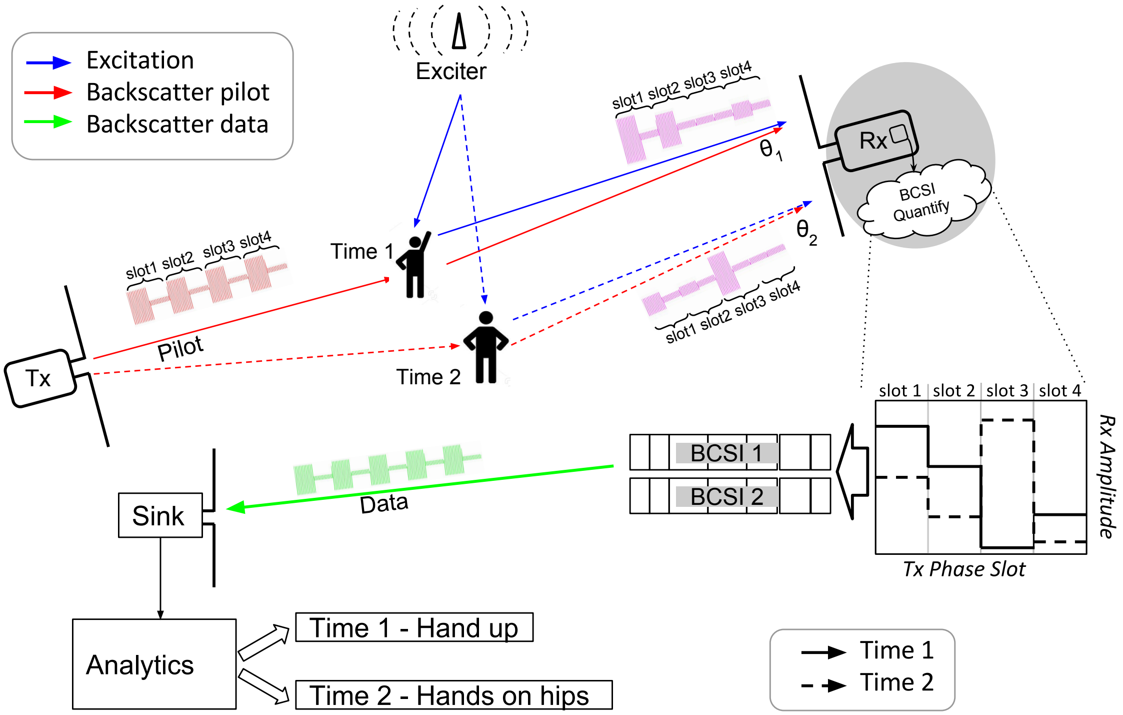


Figure 2.2: BARNET network showing two tags and one sink node

2.2 A Passive Backscattering Network for Activity Recognition

The main idea in *BARNET* is a novel technique enabling passive tags to quantify the dynamic variations in the wireless channel in response to specific activities in the vicinity. To this end, we develop a measure that we refer to as Backscatter Channel State Information (BCSI) which embodies the channel response incorporating both the amplitude (attenuation) and phase (propagation delay) between a pair of backscattering tags. We further design a novel method based on a Backscatter Pilot (BP) signal which enables passive tags to quantify and record BCSI while using only envelope detection. This ability, in conjunction with multi-hop tag-to-tag backscattering, allows tags to (1) measure wireless channel dynamics at various locations and (2) aggregate these distributed measures centrally for processing. As highlighted in Section 3, this concept amounts to a paradigm shift by enabling use of passive tag networks for activity recognition and physical analytics, applications which have, so far, been almost exclusively restricted to the realm of active radio systems.

Figure 2.2 demonstrates the proposed concept with a basic passive tag-to-tag network consisting of two tags and a sink node connected to a computer platform. The Tx tag backscatters a signal which is received by the Rx tag, in the presence of external excitation. Any activity in the vicinity of the link affects the multi-path channel between the two tags which in turn affects the amplitude and phase of the received signal. We first analyze the signal interactions to determine the impact of channel phase variation on the received signal. Based on this analysis we construct the BP signal which when backscattered by the Tx tag, enables the Rx tag to estimate and record the BCSI. *The BP signal essentially consists of short backscatter messages sent repeatedly with the same amplitude but a different, deterministic phase offset.* For implementation convenience, these repeated transmissions are done in time slots, called *phase slots*. Each phase slot is characterized by its fixed Tx phase offset which is unique from other slots. By comparing the received amplitudes in these phase slots, the Rx tag is able to quantify the instantaneous BCSI between the two tags.

As shown in Figure 2.2, we consider a situation with some human activity (movements) happening in the vicinity of a tag-to-tag link. The Rx tag is receiving two signals, the backscatter from the Tx tag and the excitation signal. We depict two time instances in-between which a specific movement is performed, thus altering the wireless channel between the two tags. We denote the resultant channel phase of the two instances as θ_1 and θ_2 (solid lines for instant 1 and dotted line for instant 2). At both instances, the Tx tag sends out the BP signal consisting of consecutive transmissions of a short message with the same amplitude, but in four different phase slots. The quantized amplitude of the baseband received signal in each phase slot constitutes our BCSI measurement, which is recorded at the Rx tag. The set of recorded BCSI measurements over time is then sent to the sink node utilizing multi-hop tag-to-tag backscattering. At the sink node, the aggregated BCSI measurements from the tags in the network are then sent to the analytics processing unit. As seen from the figure, the baseband Rx amplitude reaches peaks and nulls in different Tx phase slots. We will show with the analysis that follows that the Tx phase slot (or phase offset) where the peaks and nulls occur depends upon the instantaneous phase of the channel which in turn is determined by the multi-path environment surrounding the tag-to-tag link. Thus, the BCSI measurement provides an instantaneous fingerprint of the channel. Dynamic BCSI variation patterns can then be used for activity recognition and physical analytics using learning tools, which is implemented by the backend analytics server.

Our proposed technique is agnostic to deployment environment. This is because our technique utilizes phase diversity by introducing a deterministic phase offset in each Tx phase slot. As a result, the phase difference between the received signals in successive slots is always fixed, irrespective of the environmental clutter. This is demonstrated in our user study (Section 2.4.3) by ensuring that training and testing are done at different locations.

Figure 2.2 depicts a basic building block of *BARNET*. Use of multiple tags enables simultaneous BCSI measurements all over the deployment area. Multiple tags naturally provide redundancy and diversity that improve the accuracy and robustness. This is again demonstrated in our evaluations.

2.2.1 Signal Interactions in a Backscatter Link

The Tx tag backscatters the RF excitation signal by varying the reflection coefficient of its antenna between two states 1 and 2. This alters the signal reflected from the antenna thus achieving the well known backscatter modulation [50]. Being passive, the Rx tag receives this signal using envelope detection. We assume that the tag communicates using ASK backscattering wherein the amplitude of the reflected signal is varied between the two states.

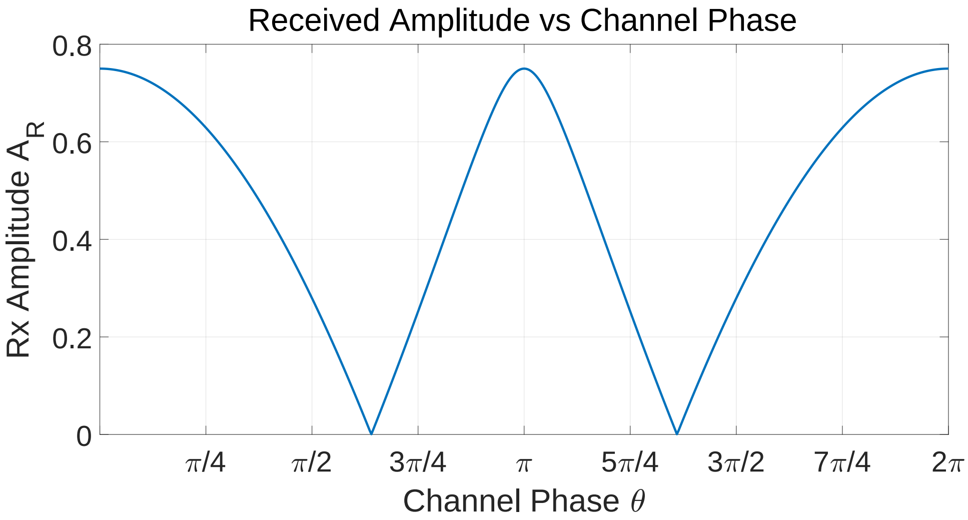


Figure 2.3: Variation in amplitude of received backscatter with phase of the channel

Then the signal received at the Rx tag in the two states denoted by S_R^i $(i = 1, 2)$ is as follows:

$$S_R^i = A_E \cos(\omega t) + A_T^i \cos(\omega t + \theta), \qquad i = 1, 2, \tag{2.1}$$

where, A_E is the amplitude of the received exciter signal, A_T^i is the received amplitude of the backscatter signal from the Tx tag in the two states, $\omega = 2\pi f$ with f being the carrier frequency of the excitation signal and θ is the resultant phase of the backscatter channel between the two tags. Expanding the above, we get

$$S_R^i = (A_E + A_T^i \cos\theta)\cos(\omega t) - (A_T^i \sin\theta)\sin(\omega t), \qquad i = 1, 2. \tag{2.2}$$

Then the amplitude of the total received signal at the Rx tag in the two states, A_R^i is:

$$A_R^i = \sqrt{(A_E)^2 + 2A_E A_T^i \cos\theta + (A_T^i)^2}, \qquad i = 1, 2 \tag{2.3}$$

Since the Rx tag uses envelope detection, the amplitude of the resultant received backscatter signal, A_R is given by

$$A_R = |A_R^1 - A_R^2| = \Bigg| \sqrt{(A_E)^2 + 2A_E A_T^1 \cos\theta + (A_T^1)^2} \\ - \sqrt{(A_E)^2 + 2A_E A_T^2 \cos\theta + (A_T^2)^2} \Bigg| \tag{2.4}$$

The plot of this received amplitude vs. the instantaneous backscatter channel phase θ is shown in Figure 2.3. The plot uses normalized amplitudes with A_E = 1, A_T^1 = 0.75 and A_T^2 = 0.05.[2] As we can see, the phase of the backscatter channel, θ has a significant impact on the received amplitude. Detailed analysis, along with simulations and experimental verification, are well established in this work [140]. As we can see, the Rx tag amplitude varies significantly between peaks and nulls depending upon phase of the backscatter channel. We exploit this phenomenon by introducing *known and fixed* phase offsets at the Tx tag.

2.2.2 Use of Multiple *Phase Slots* in Backscattering

We now repeat the above analysis for the case when the Tx signal is sent successively over multiple fixed phase slots. The phase slots are implemented using an extended backscatter modulator design which will be explained in Section 2.3. As mentioned earlier, the concept of multiple phase slots implies that tag will backscatter an ASK signal with the same amplitude, but a deterministic phase offset. The phase offset is the characterizing feature of each Tx phase slot. In order to better understand this concept, let us imagine a backscatter modulator that incorporates K phase slots. The ASK backscatter will be generated in each slot by switching between two states as follows:

$$\underbrace{A_T^{1,1} \longleftrightarrow A_T^{2,1}}_{\text{Slot 1}} \quad \underbrace{A_T^{1,2} \longleftrightarrow A_T^{2,2}}_{\text{Slot 2}} \quad \cdots \quad \underbrace{A_T^{1,K} \longleftrightarrow A_T^{2,K}}_{\text{Slot K}} \tag{2.5}$$

The characteristic phase offset of each slot is denoted by ϕ_k. This fixed phase offset is introduced into the backscattered signal at the Tx tag (prior to propagation to the Rx tag

[2] Backscattering works by reflecting a *fraction* of the incident power in both states. Moreover, in general, the excitation signal received at the Rx tag will be much larger than the signal received from the Tx tag in the two states. So this assumption for amplitude values simplifies the simulations without altering the accuracy of the characterization of received amplitude vs phase

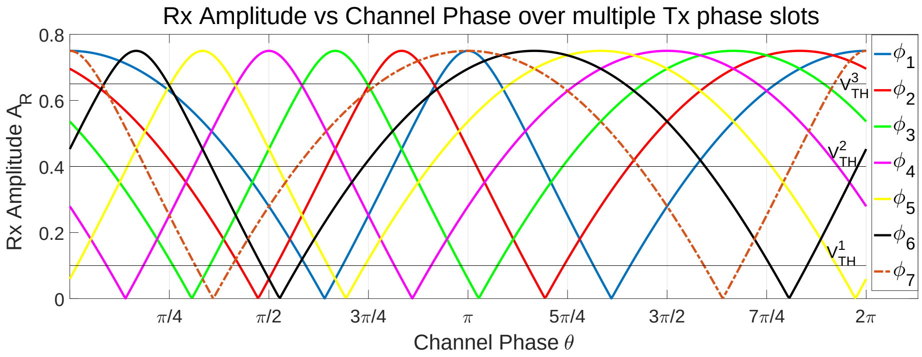


Figure 2.4: Received signal over multiple phase slots

over the backscatter channel). This value is *fixed* for each slot during the implementation of the tag hardware. Then, using equation 2.4, the amplitude of the received signal in each slot is given by

$$\begin{aligned} A_R^k &= |A_R^{1,k} - A_R^{2,k}| \\ &= \left| \sqrt{(A_E)^2 + 2A_E A_T^{1,k}\cos(\theta+\phi_k) + (A_T^{1,k})^2} \right. \\ &\quad \left. - \sqrt{(A_E)^2 + 2A_E A_T^{2,k}\cos(\theta+\phi_k) + (A_T^{2,k})^2} \right| \\ &\qquad\qquad k = 1, 2, ...K. \end{aligned} \tag{2.6}$$

Figure 2.4 shows the amplitude of the received backscatter signal vs. the instantaneous phase of the backscatter channel θ for $K = 7$ different phase slots each with phase ϕ_k = $\frac{(k-1)\pi}{6}$, $k = 1, 2, ...7$. As we can see, the Rx amplitude depends upon both the phase of the channel θ and the slot phase ϕ_k. At any given instant t the phase of the channel θ_t will depend upon the dynamic environment at that instant. However the phase difference between the received signals in successive phase slots, $\Delta\phi$ = $\phi_k - \phi_{k-1}$ *remains constant* $\forall t$. This key property along amplitude vs. channel phase behavior enables our BCSI quantification and phase estimation technique.

2.2.3 Quantifying BCSI

In the simplest embodiment, the receiver in the tag consists of an envelope detector followed by a single comparator. Consider a comparator threshold V_{TH}^{1} as shown in Figure 2.4. Depending upon the value of the instantaneous phase of the channel θ_t, the Rx signal in some specific slots will be above this threshold. For example, if the resultant channel phase θ is $\pi/2$, the detection vector for the 7 slots considered will be $[1, 0, 1, 1, 1, 0, 1]$. If the channel phase were to change to π, then the vector would change to $[1, 1, 0, 1, 1, 1, 1]$. This vector represents our passive BCSI measure which is recorded at the Rx tag. Our analytics processor then maps these BCSI measures to the channel characteristic in Figure 2.4 to obtain phase estimates of the channel which in turn are used for activity recognition.

Use of multiple thresholds to improve precision Clearly, using a single threshold to quantize the Rx amplitude will limit the precision of the phase estimation. To improve precision, we propose a BCSI estimator with multiple threshold levels. The hardware implementation of this circuit is described in Section 2.3. In this case, the amplitude in each phase slot will be represented by multiple (M) bits instead of 1. For instance, using three thresholds V_{TH}^{1}, V_{TH}^{2}, and V_{TH}^{3} as shown in Figure 2.4, the BCSI vector for channel phase $\pi/2$ and π respectively is $[100, 000, 110, 111, 110, 000, 100]$ and $[111, 110, 000, 100, 110, 111, 111]$. It is simplified here for illustration purposes, the quantized amplitude is represented using thermometer code. In the actual implementation, we will use an ADC (Section 2.3) whereby the quantized amplitude will be represented using binary code. Obviously, higher number of thresholds leads to more precise BCSI measurement while increasing the hardware resources and power consumption.

Selection of phase offsets for each phase slot In Figure 2.4, phase slot offsets ϕ_k are chosen uniformly between 0 and π. However, this may not always be the most optimal way to select the offsets. The accuracy of the BCSI measure is affected by the number of slots, the phase offset of each slot and the number of Rx quantization levels. Increasing the number of phase slots and quantization levels increases network latency, resource requirement and power consumption. The selection of ϕ_k values for our implementation will be guided by a detailed tradeoff analysis involving the above mentioned parameters.

Amplitude changes in the baseline (excitation) signal The amplitude of the baseline excitation signal at the Rx tag, A_E also changes dynamically in response to activities in the vicinity of the tag. These dynamic variations are rich in analytic information and provide additional supplementary information that can be used for activity recognition. Our design for BCSI measurement also allows us to record the baseline signal A_E. This information is appended to the BCSI measure and is used in the analytics processing.

Backscatter Pilot (BP) signal To enable the measurement of the instantaneous BCSI at the Rx tag, the Tx tag will transmit a *Backscatter Pilot* (BP) signal. This is conceptually similar to pilot signals used in traditional wireless communication for channel estimation. Within the BP signal, the Tx tag backscatters a self-identifier, a short random number used for synchronization (more on this later), followed by short *slot identifiers* sent successively over each phase slot. The Rx tag correspondingly records the Tx identifier, the synchronization random number (SRN), and then serially records the BCSI measure, consisting of M-bit received amplitude in each slot, and baseline signal A_E. This process is then repeated by the same Tx tag N times, each time incrementing the SRN by 1. The number repeated cycles for which a single Tx tag backscatters the BP signal is a parameter of the deployment environment and the frequency/speed of the activities that we wish to tune the network to. This N-cycle BCSI measurement , along with the identifier of the Rx tag, is then conveyed to the sink node for analytics processing using multi-hop tag-to-tag backscattering. Figure 2.5 shows the format of the BP signal and the information recorded by the Rx tag.

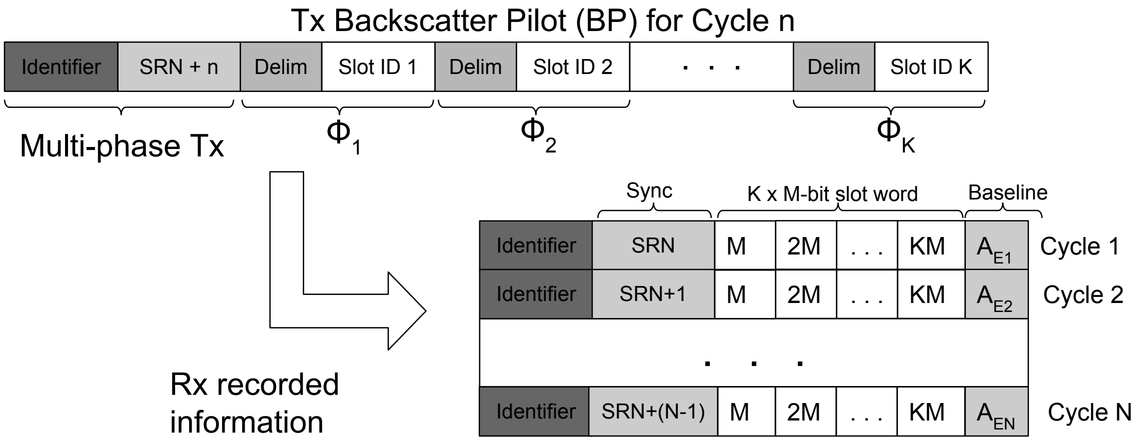


Figure 2.5: Transmission and reception of Backscatter (BP) signal

2.2.4 Activity Recognition Using Aggregated BCSI

The BCSI measures from various tags in the *BARNET* are aggregated at the sink node where they are passed on to the central analytics engine for processing. First, the incoming data is parsed for tag identifiers and then the BCSI measures from an individual tag are lined up in a time sequence according to the value of the SRN prepends to the BCSI measurements. The concept of the SRN helps to synchronize simultaneous BCSI readings from multiple tags. When one tag is transmitting a sequence of BP signals, multiple tags in the vicinity could receive it. The measurements recorded by each receiving tag correspond to the BCSI reading for separate channels. However if these are happening simultaneously, they recorded BCSI will have identical SRN values. The use of these SRNs is immensely valuable at the analytics engine because it gives the network the ability to detect synchronous BCSI measurements. Our approach has a simultaneous or parallel sensing advantage compared to other approaches which make use of a central active receiver. Multiple activities occurring throughout the network can be simultaneously and independently monitored by pairs of tags in different locations before being sent to the sink for processing. Here the relatively close communication range between passive tags works to our advantage by preventing the influence of far away activities on BCSI of a tag-to-tag link. At the same time, the low cost of the tags allows for dense deployments whereby a single activity can be sensed and recorded by multiple tags.

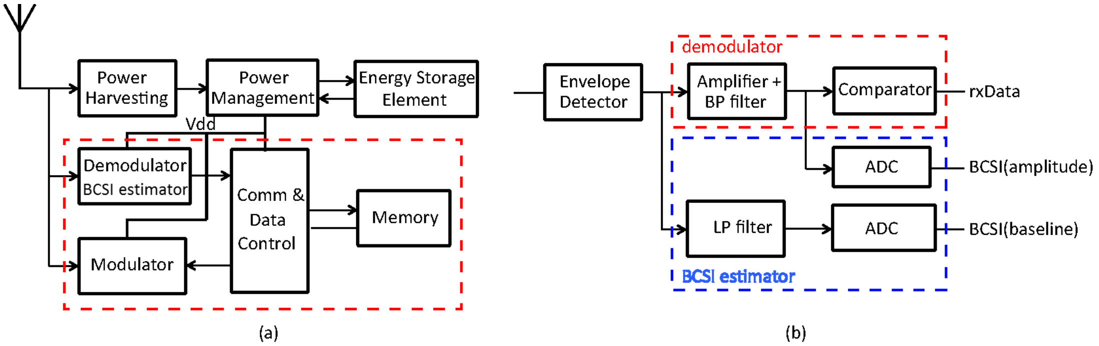


Figure 2.6: Block diagrams: (a) *BARNET* tag architecture, (b) demodulator/BCSI estimator shown in further detail. The COTS prototype used for evaluations in this work (Section 2.4) does not include power harvesting, power management and energy storage parts and communication/control and memory parts are implemented using a microcontroller unit (MCU).

2.3 *BARNET* RF Tag Implementation

In this section, we describe the hardware implementation of the tag that is the principal building block of *BARNET*. A high level block diagram of tag architecture is shown in Figure 2.6.

The primary hardware modules in *BARNET* are (1) the backscatter modulator incorporating the capability to transmit the BP signal for multiphase probing of the channel, (2) the demodulator incorporating the critical BCSI estimator, (3) the power harvesting/managing unit which harvests power from the excitation signal and manages power at the various blocks in tag's architecture. (The current COTS implementation used in the evaluation does not have the power harvesting/managing unit).

There are two specific challenges in the tag design we want to highlight here. First, in conventional RFID systems, the active readers transmit a signal which has a very large modulation depth (also called modulation index). This is a measure of how much the peak modulated level of the signal varies with respect to its unmodulated level. This large modulation depth along with high SNR of the reader signal enables tags to demodulate this signal using straightforward implementation of passive envelope detection. In contrast, *BARNET* tags have to detect backscatter signals from other tags that inherently have orders

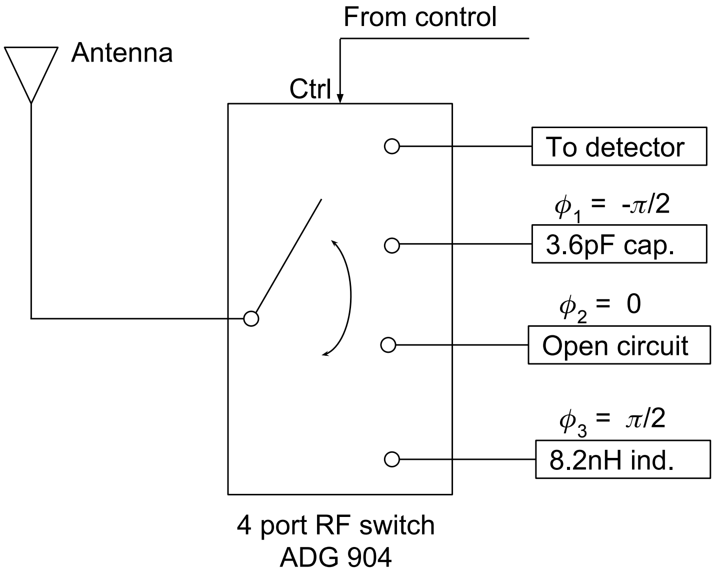


Figure 2.7: Implementation of multiphase backscatter modulator using an RF switch controlled by the MCU in the comm/control section.

of magnitude lower modulation depth and SNR. Second, *BARNET* tags also need to passively quantify the BCSI. These challenges makes design of the Tx (backscatter modulator) and Rx (envelope detector and BCSI quantifier) modules of the *BARNET* tag significantly more complex than conventional tags. We now describe the design of the principal building blocks of the *BARNET* tag.

2.3.1 Multi-Phase Backscatter Modulator Design

The traditional backscatter modulator transmits data by switching between two different impedances connected to the antenna. The *BARNET* modulator builds on this basic architecture and introduces novel schemes for two operating modes, viz., (i) channel probing between tags using the BP signal as described in Section 2.2.3, and (ii) regular data communication between tags. The modulator incorporates multiple terminating impedances each implementing the so-called phase slot. Figure 2.7 shows the conceptual architecture of the backscatter modulator. It is built using a multi-throw RF switch in a discrete PCB implementation of the *BARNET* tag in our current prototype and can be substituted with a transistor implementation on chip.

During channel probing using the BP signal, the modulator transmits systematically over different phase-slots. The terminating impedance and hence the resulting phase offset of each slot are deterministic and fixed by design. This enables the Rx tag to measure the amplitude of the backscatter signal as a function of the transmitter phase, as in Figure 2.4, and robustly quantify the BCSI of the link. The reflecting phases span the range from $-\pi$ to π. Clearly, looking at Figure 2.4, a larger number of phase slots will increase the accuracy of BCSI measurement. Conversely, higher number of phase slots will hamper the network throughput by increasing size of the BP signal. Higher number of phase slots also increases the hardware resources and cost. Thus there is a tradeoff involved in selection of the number of phase slots. The goal of the first prototype *BARNET* tag is to demonstrate and evaluate the proposed concept while keeping the implementation as simple as possible. To that end, we have implemented a three phase-slot modulator using a 4 port RF switch with one of the ports is connected to Rx detector and the other 3 throws are terminated with impedance to achieve the phase slot values of $\phi_1 = -\pi/2$ ($3.6pF$ capacitance), $\phi_2 = 0$ (open circuit); and $\phi_3 = \pi/2$ ($8.2nH$ inductance).

For regular data communication, the modulator must overcome the phase cancellation problem that was independently investigated in [140]. This problem can be understood by looking at Figure 2.4. The Rx amplitude in a given phase slot can go down to very small value or even zero depending upon the instantaneous phase of the channel. This happens because the exciter signal and the backscatter signal essentially combine out-of-phase at the receiving tag. *BARNET* already provides a built-in mechanism for protection against this problem by the use of multiple phase slots. *BARNET* tags can learn what phase is best to use for regular data communication between neighboring tags by first 1) redundantly transmitting over multiple phase slots (this is similar to channel probing described in the previous para) and then recording the received signal amplitudes and then 2) using that phase slot for subsequent communication that provides the highest amplitude. More on this is in Section 2.4.2.

2.3.2 Demodulator and BCSI Estimator Design

The architecture of the demodulator and the BCSI estimator is shown in Figure 2.6(b). Both blocks process the baseband signal obtained after the envelope detection. The demodulator converts the amplitude of the baseband signal into a binary signal for data communication, while BCSI estimator quantizes the same signal with a higher resolution. BCSI estimator also quantizes the baseline signal at the output of the envelope detector.

Architecture

Although the concepts of signal reception using passive envelope detection have been widely explored in the context of backscatter systems (see, for example, [39, 81]), their application to tag-to-tag communication system gives rise to unique challenges. As mentioned earlier for *BARNET* tags, the *modulation index of the incoming signal is at least an order of magnitude lower* than for standard RFID tags. This greatly limits the operating distance of passive tag-to-tag links in current literature except when CDMA encoding was used resulting in a very low data rate and higher power requirement [120].

We use the envelope detector as the input circuit at the interface with antenna followed by the passive filter to improve signal-to-noise ratio (SNR), as in the conventional RFID tags [39, 45, 81, 158]. To overcome the low modulation index, we implement demodulator with the architecture comprising an amplifier with integrated band-pass filter followed by comparator. The integrated filtering removes the baseline excitation signal from the modulated signal. With the proposed architecture, a small amplitude of the baseband signal (on the order of few mV) can be resolved with a straightforward low-power comparator implementation.

Our BCSI estimation involves measurement of the amplitude of the baseband signal over different Tx phase slots. The amplifier used in the demodulator is shared with BCSI estimator, but the output of the amplifier, in this case, has to be quantized with a finer resolution. The power budget of the tag limits the resolution of analog-to-digital conversion (ADC) and the optimal resolution is determined through experiments.

The baseline signal after the envelope detection is continuously measured while BCSI is collected. The baseline signal is extracted by low-pass filtering of the envelope detector signal with a cut-off frequency lower than the data rate. The same ADC architecture is used for this quantization as well, although the optimal resolution could be different than in the case of measurement of the backscatter signal amplitude.

COTS Implementation

As an initial approach to evaluation of *BARNET*, the proposed architecture of the demodulator and BCSI estimator has been implemented using commercial-off-the-shelf (COTS) components. The envelope detection comprises of two stage voltage multiplier implemented using zero bias Schottky diode HSMS-285x series from Avago Technologies. The envelope detector is followed by a high-resolution 16-bit (full range 5 V) 80 kbps analog-to-digital conversion. The recorded signal is captured and stored in memory. The on-board processing and filtering, along with comparison and limited resolution analog-to-digital conversion for demodulation and BCSI estimation, mimicking the architecture shown in Figure 2.6(b), is then implemented in Matlab using the captured data.

ASIC Implementation

To demonstrate the feasibility of the proposed architecture implementation in a passive tag with a limited power budget, we have designed a demodulator in 45nm CMOS technology and estimated the power consumption of the two ADCs used in BCSI estimator.

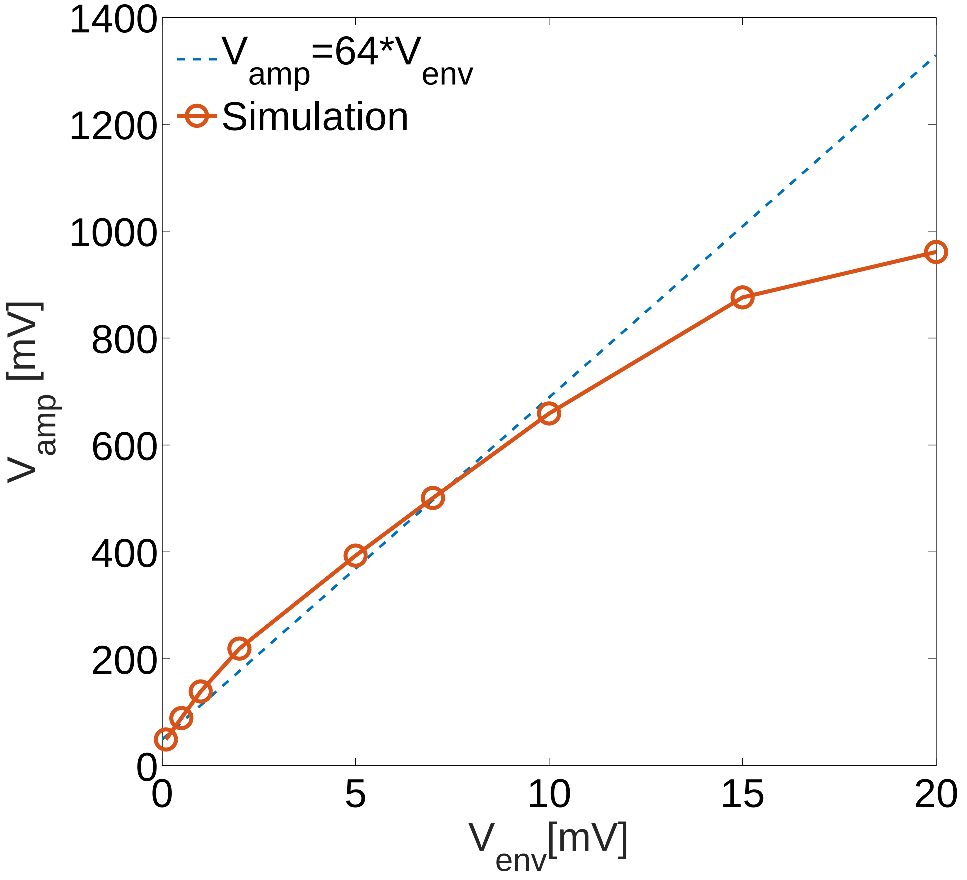


Figure 2.8: Output of the amplifier in the demodulator for different amplitudes of the received signal after envelope detection.

The demodulator implementation comprises voltage doubler for envelope extraction and an amplifier with integrated filtering followed by a comparator. The voltage doubler rectifies and at the same time increases the amplitude of the input signal. At the input power of -28 dBm, the optimized design of voltage doubler leads to the output voltage of 146 mV and ripple voltage of 228 μV [80]. The large output voltage and low ripple thereby provide a smooth, detectable baseband signal input to the amplifier. After the envelope detector, the ASK modulation in the baseband signal, due to low modulation index, cannot be distinguished by the comparator. Instead, the baseband signal is first amplified with integrated high-pass filter. The high-gain amplifier is implemented as the low-noise, low-power folded-cascode amplifier. The simulated gain of the baseband amplifier in the passband is 40.1 dB, with the corner frequencies at 2.9 kHz and 50 kHz. With the amplitude of the received baseband signal swept, the output of the amplifier is shown in Figure 2.8. Averaged gain of amplifier is 64 which is enough to detect very weak signal. The power consumption of the demodulator is 1.2 μW [80].

ADCs for BCSI estimation can be implemented using the successive approximation ADC architecture. Based on the similar implementations in the literature [71, 135], we estimate a power consumption of each ADC to be less than 1 μW at 10 ksamples/s sampling rate.

2.3.3 Power Harvesting

The power for the operation of the tag is harvested from a dedicated external RF exciter. Since the tag should operate on the distances on the order of 10 m from the exciter, the power consumption is the most stringent constraint in the design of tag's circuitry. The power is mostly harvested when *BARNET* tag is in the listening mode (no tag-to-tag communication and no channel sensing), since only a small fraction of the received power is consumed by the wake-up circuit. When the tag is transmitting data, only the modulator is turned on and the loading impedance of the antenna changes according to the modulation scheme. This means that the amount of the absorbed power will vary and that in some states the modulator has to be powered with the stored energy. However, the power consumption of the modulator is very low since the only active component is a switch. When the tag is receiving data or estimating BCSI, i.e., when either the demodulator or BCSI estimator block is turned on, the total received power is split between the power harvesting block and one of these blocks. In this case, extra energy for the operation is provided by the energy storage element implemented through a supercapacitor.

The COTS implementation of *BARNET* tag does not integrate the power harvesting and the tag is powered by a CR 1620 coin cell battery. The proposed ASIC design for the demodulator and BCSI estimation has the power consumption on the order of 4 μW. At this power level, it has been demonstrated that the demodulator can resolve a modulation index of 0.6% [80]. Considering that the state-of-the-art efficiency of the power harvesting circuitry is 30% [149], the tag could operate from the harvested RF energy in the environment in which the input power is on the order of $-20\,\mathrm{dBm}$. The tag-to-tag link could operate at distance of 2 m if the Tx tag is also receiving at the least the same input power of $-20\,\mathrm{dBm}$ [80].

2.3.4 Tag Antenna and Comm/Control Section

To build a fully functional *BARNET* tag, the above modules are augmented with an antenna and a digital processor. In current COTS implementation used for evaluations here (Section 2.4) we use a dipole based antenna for the tag printed on an FR4 circuit board with a single SMA connector to connect to the rest of the tag circuitry. The comm/control section of the tag also runs the physical and link layer protocols. This section is now implemented on a TI MSP430 microcontroller unit (MCU).

2.3.5 Physical and Link Layer Design

In this section we describe the PHY layer protocol parameters used in the BARNET tag implementation. For simplicity and robustness of implementation which we have chosen standard encoding techniques from backscatter communication literature. Use of well established encoding techniques allows to focus mainly on evaluation of the proposed BCSI measurement techniques.

Encoding Because of the passive modulation and demodulation techniques employed in *BARNET* tags, delay (or transition) encoding mechanisms like Miller and FM0 are a good fit. In such schemes the data to be transmitted is represented by the presence or absence of high-low transitions at the boundaries or the center of a symbol. These encoding schemes have been widely researched in the context of standard RFID systems [53]. We use Miller modulated subcarrier encoding with multiplication factor M-2 (Miller-2 encoding). This provides a good data rate vs noise immunity tradeoff. Also this encoding can be easily scaled up to Miller-4 or Miller-8 to further increase noise protection at the cost of data rate. Our *BARNET* tags have used an on-board clock of 10 KHz and a link data rate of 5 Kbps.

Pilot-tone	Preamble	Header	Data	CRC

Figure 2.9: Packet format in *BARNET*

Pilot Tone The pilot tone is simply a sequence of identical symbols which helps the to identify the start of a tag packet. In the *BARNET* tag the pilot tone also allows the RC circuit producing the comparator reference voltage to charge up to the required threshold value well before the actual preamble bits arrive. This ensures that when the actual data bits arrive, the comparator threshold is set to the proper value for correct digitization. When not communicating, the BARNET tag is in an idle mode and working on harvesting all received power. The pilot tone also serves to issue an interrupt to the MCU that forces the tag to wake up from the idle mode. This significantly reduces the idle state power consumption of the tag We use a sequence of ten 0's encoded using the Miller-2 scheme as our pilot tone.

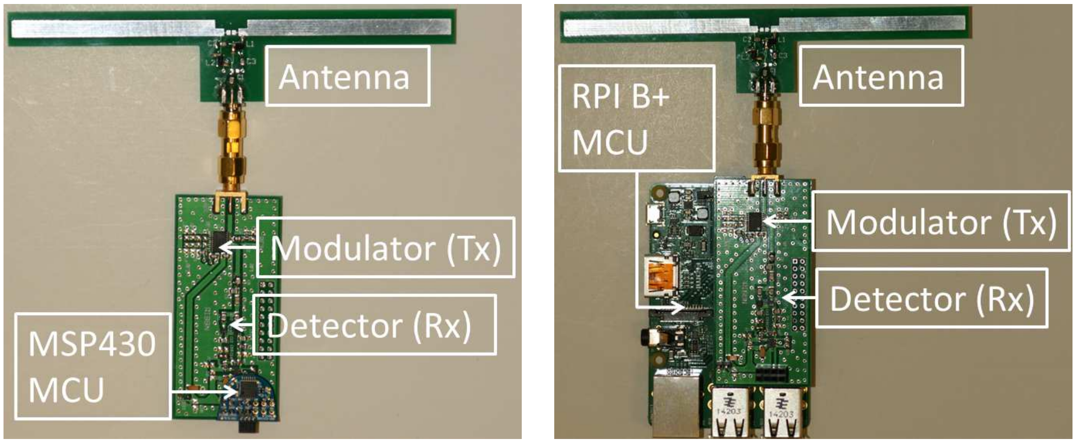
Decoding Decoding is implemented on the *BARNET* MCU by sampling the output of the comparator and measuring the time between edge transitions. This time between successive transitions is mapped to the decoded data bits. For the decoder the pilot tone can provide calibration to compensate for clock drifts or other timing drifts in the MCU. Timing of the edge transitions in the pilot tone at the start of each packet is used as a reference to decode bits in the remainder of the packet. Clearly a higher input sampling rate with add robustness to the decoder. But this would require a higher clock speed which can add to the power consumption. Our chosen data rate of 5 kbps provides a good balance between noise immunity and power consumption.

Packet Design In the current *BARNET* prototype the packet format is simple to keep it short. As shown in Figure 2.9, the packet consists of pilot-tone, preamble, header, payload, and CRC. The header contains the id of the transmitting tag for backscatter pilot (BP) transmission and also includes the id of the destination tag in case of point to point data communication, Finally, the data field consists of either the communication payload packet or the multiphase BP signal as explained in Section 2.2

Link Layer Design The *BARNET* link layer implements a basic listen-before-send mechanism via carrier sensing to avoid interference between neighboring *BARNET* links. Since the tags have receive ability implementation of carrier sensing is straightforward. The implementation follows the basic ideas outlined in [95]. Before transmission the MCU samples the comparator output to detect presence of pilot tone indicating presence of modulated backscatter. If present, the comparator output flips signifying presence of a neighboring transmission. We also implemented a random backoff strategy mimicking conventional CSMA/CA protocols (e.g., 802.11). The tag backs off for a randomly chosen duration if the carrier is sensed busy. The expected value of this interval doubles for each transmission failure. Each (unicast) transmission is followed by an ack to detect failure.

For communication *BARNET* tag operates in three distinct modes: *transmit*, *receive* and *listen* modes.

By default the tag is always in the listen mode waiting to detect a preamble. Once preamble is detected it switches to the receive mode. In this mode, the tag decodes the packet and after all bits are received checks for correctness. If the packet is correctly received, then it switches to the transmit mode to transmit an acknowledgment. Then, it evaluates the packet header and determines the action to be taken on the packet. For example, it may need to further relay the packet to a neighbor for eventual transmission to the sink node.



(a) *BARNET* tag (front)

(b) Sink node (RPI with tag)

Figure 2.10: Prototypes of (a) tag and (b) sink

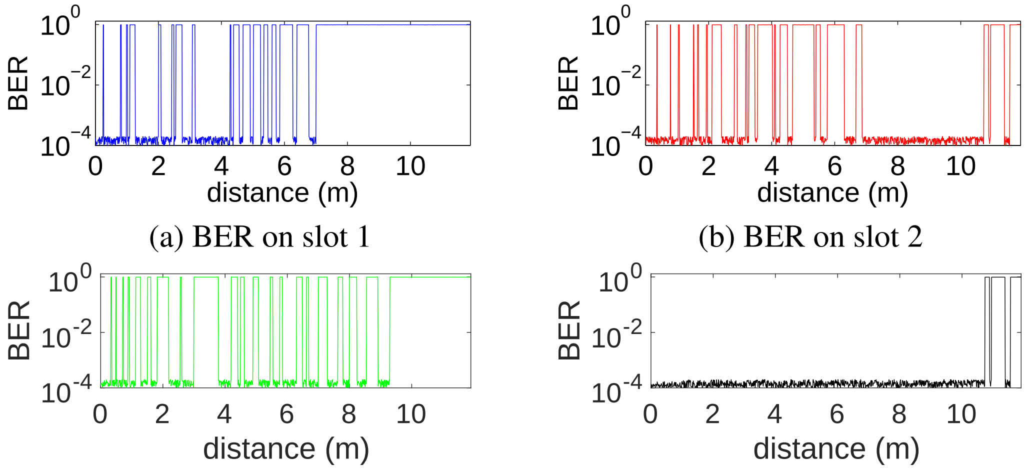
2.4 BARNET Prototyping and Evaluation of Links

2.4.1 COTS Prototype Implementation

The prototype *BARNET* platform used for evaluation in terms of communication and activity recognition performance is shown in Figure 2.10. It is made of 2-layer FR4 PCB in thickness of 31 mils with components on both sides. The PCB is designed using Altium designer software and is manufactured by Goldphoenix Printed Circuit Board Company in Wuhan, China. The antenna is implemented directly on a separate piece of PCB that is attached to the main PCB using SMA connector. This modular design helps future experiments with different types of antennas.

The ADG 904-SP4T CMOS FET from Analog Devices deployed as RF the switch [17]; the impedances mentioned in the previous section are connected to the input terminals of it. The MCP6561 Low-Power Push-Pull Output Comparator by the Microchip Technologies deployed as the comparator [13]; it links MCU's digital GPIO and envelope detector's output.

The gateway or sink is implemented using a Raspberry PI Model B board [18] taking up the role of the MCU. Here, the comparator output is fed to a GPIO pin of the Raspberry PI that is sampled through Wiring Pi's native C library [70] at the speed of 22MHz [123].



(a) BER on slot 1

(b) BER on slot 2

(c) BER on slot 3

(d) BER with the best slot

Figure 2.11: BER vs distance for a single *BARNET* link with $-15\,$dBm power at the Tx (backscaterring) tag

The Raspberry PI communicates with a host computer using the Ethernet. The computer runs all activity analysis software.

The exciter is implemented using a software radio platform (BladeRF [5]) and open source software [6]. The BladeRF is connected to a host computer using USB3.0. In between BladeRF and Laird's 902-928MHz 9dBi circularly polarized antenna [2], we plug RF Bay's 915-LNA series [12] to amplify exciter signal to supply the requisite excitation power.

2.4.2 Single Link Performance

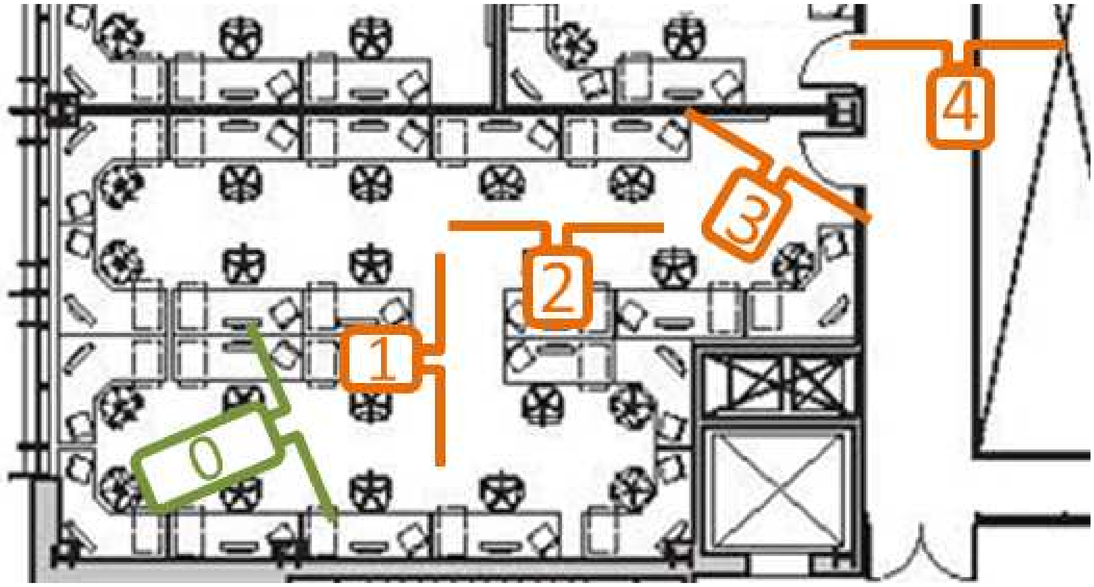
We examine the performance over different phase slots and measure the maximum achievable link range for different exciter power levels. All evaluations are done using a pair of communicating tags with a single bit rate (5 Kbps) with the exciter generating a CW signals at 915 MHz. The Tx tag is configured to continuously backscatter a known packet at regular intervals. Each transmission involves a repetition of the same packet over three different phase slots to overcome the phase cancellation problem (see Section 2.2.3 and [140]). The bit error rate (BER) is evaluated by comparing the bits decoded at the Rx tag with the known packet being continuously transmitted by the Tx tag. Figure 2.11 shows the communication distances that were achieved at a fairly low excitation power level. Importantly, Figure 2.11 shows how three phase slots exploit the phase diversity of a backscattering link. Specifically, each phase slot demonstrates signal nulls at certain distances where the exciter signal and the backscattered signal cancel each other. These nulls occur at different points for different phase slots. Choosing the best possible slot achieves very low BER providing a link range up to 10 m at -15dBm power at the Tx tag. This range reduces with lower power (not shown), e.g., about 3m at -20dBm power. A simple learning mechanism is implemented that probes each neighboring tag with different phase slots determines the right slot to use for a given neighbor.[3] The learning is to be repeated periodically to counter for any movement and changes in the environment.

2.4.3 Multihop Tag-to-Tag Network

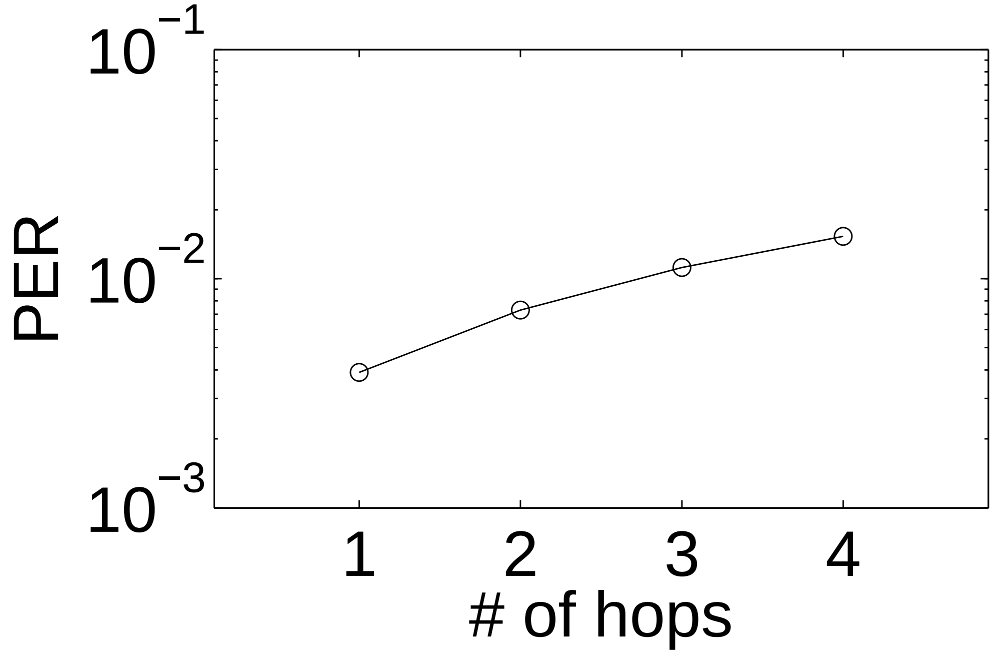
We have evaluated the performance of multihop relaying using a single *BARNET* tag configured as a transmitter, multiple *BARNET* tags configured as relays and one *BARNET* tag configured as the sink. The transmitter continuously transmits a pre-determined data packet at regular intervals. The relays listen for a transmission and then backscatter the received data packet. The sink listens and records all received packets. The goal of this experiment is to demonstrate and evaluate the potential of long distance communications via relaying over multiple backscatterring tags. Once a routing protocol has been implemented, all tags in the *BARNET* (except the sink) will have the same state machine and will switch between transmit and listen states as specified by the protocol.

The multihop set up extends the scenario we have shown earlier in Section 2.4.2 where the excitation power in the environment is set to around -25dBm to -20dBm through the length of the multihop link. We place 5 tags roughly in a line with inter-tag distance

[3] Redundant transmissions over multiple slots may be needed for broadcast communications so that all neighbors are covered.



(a) Multihop *BARNET* set up in lab



(b) end-to-end multihop PER

Figure 2.12: Multihop tag-to-tag communication in *BARNET* over 4 hops (-25 to -20 dBm excitation power at tags)

roughly about 3 m. One of the tags at the end is also the sink. See Figure 2.12(a). The set up provides end-to-end distance of about 12 m over 4 hops at the stated power level. See Figure 2.12(b). The packet error rate over 4 hops (total of ≈12 m) is about 1%.

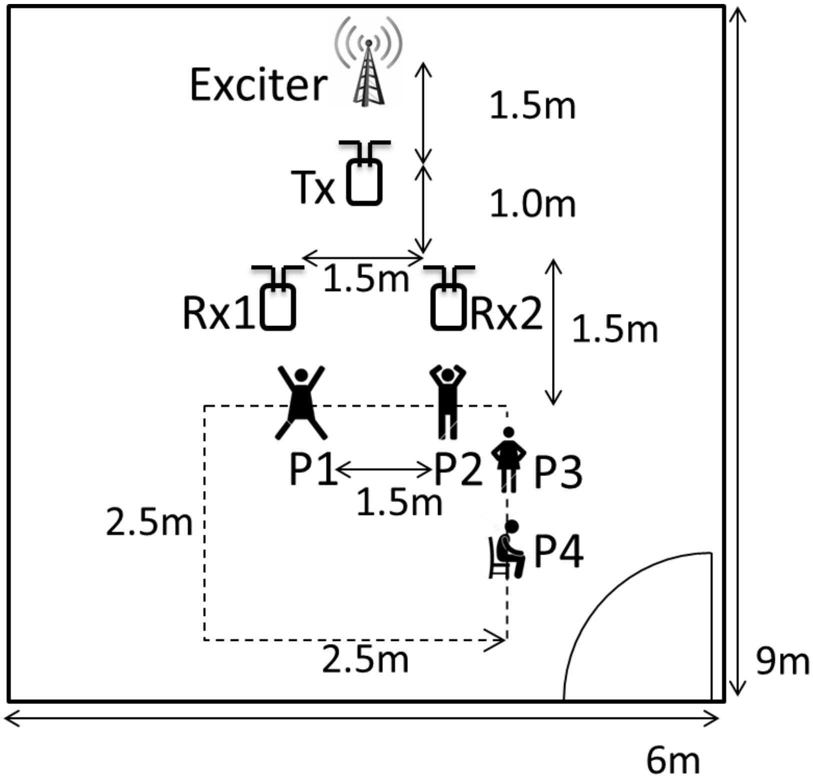


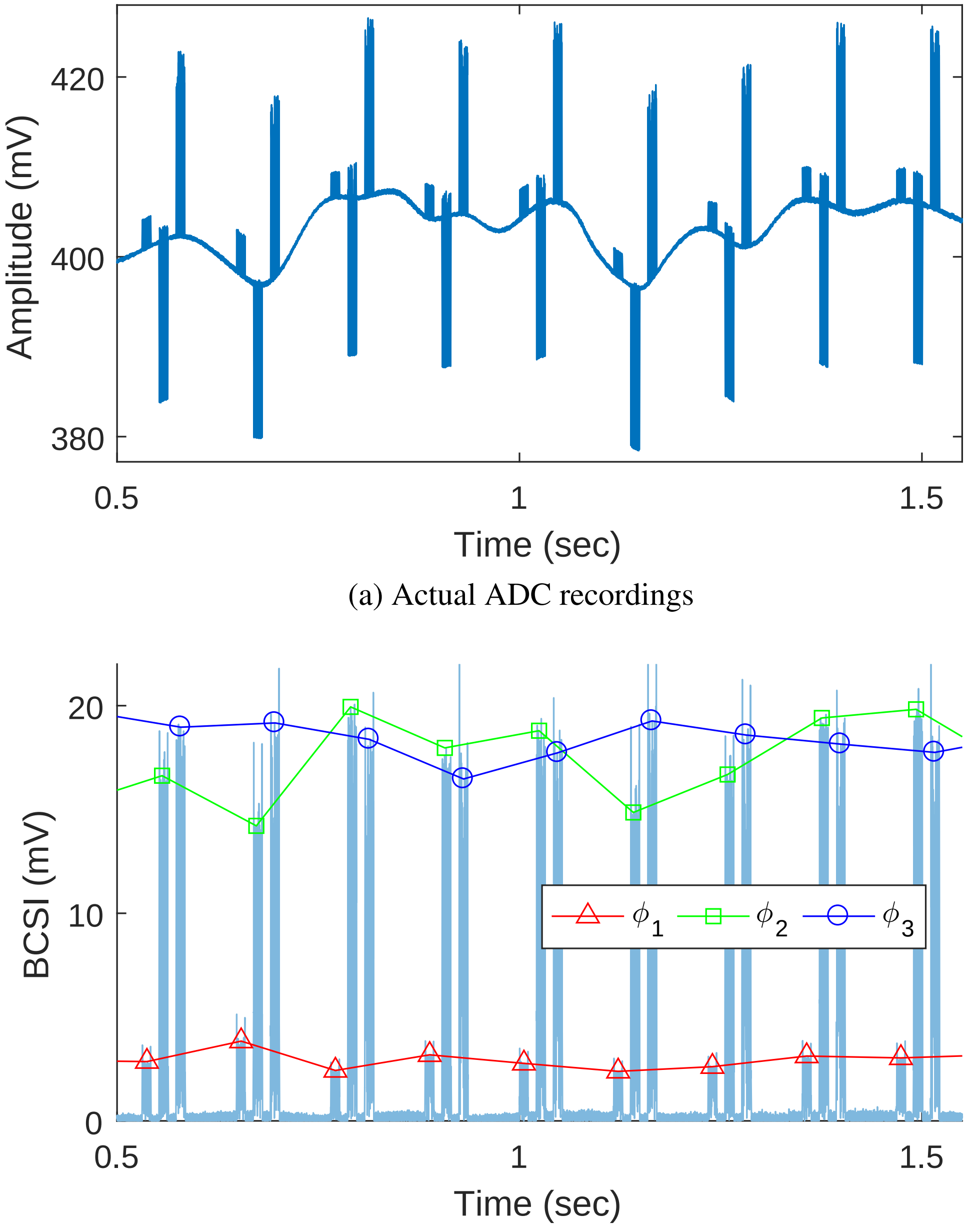
Figure 2.13: Setup for activity recognition experiments

2.5 Activity Recognition via BCSI Measurement

In this section we demonstrate *BARNET*'s capability of recognizing human activities in a 'device-free' fashion via measuring and analyzing BSCI as described in Section 2.2.

2.5.1 Evaluation Setup

As mentioned in Section 2.3.2 in the COTS implementation used here we have used an externally deployed high resolution ADC (Figure 2.6). This allows us to understand exactly how many bits are needed for the optimal design that makes the best choice between performance, complexity and power consumption. The analysis uses standard machine learning tools hosted on a computer (Intel i5-5250 CPU and 8GB RAM). It takes only about 50 ms to process 2.5 s worth of BCSI values sampled at about 100 samples/s. This sample rate is



(a) Actual ADC recordings

(b) Feature extraction

Figure 2.14: Feature creation from ADC recordings of the falling activity

chosen as it is roughly consistent with the data carrying capacity of the tag network. Thus *BARNET*'s activity recognition can be done roughly in real time.

For the study, 9 participants (8 male, 1 female, ages 25-35, all healthy, physically fit and of average built) conduct 10 different activities of daily living. The activities are 1) falling, 2) running on a path, 3) running in place, 4) sitting, 5) sitting down from standing

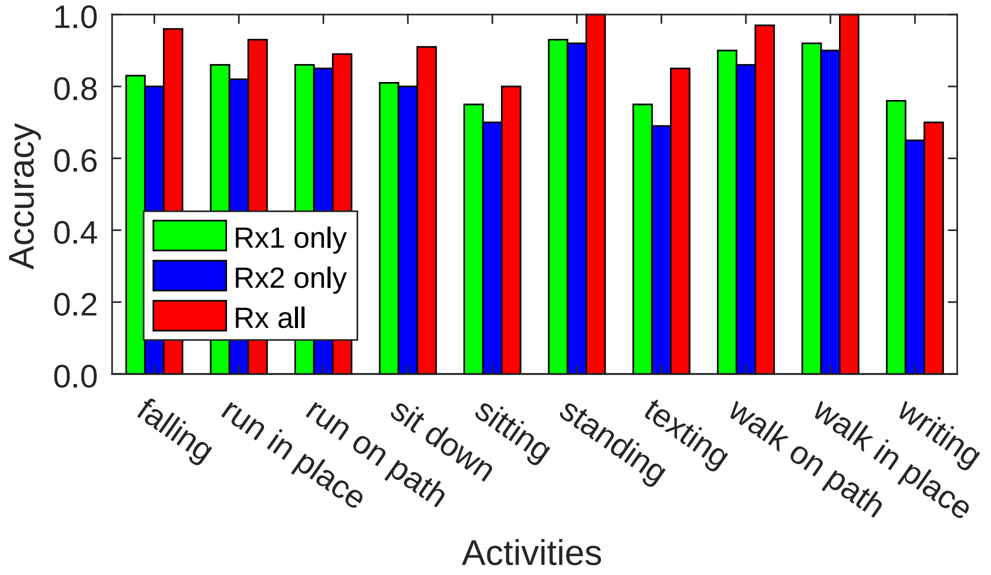


Figure 2.15: Activity recognition accuracy for different activities with one tag or both tags

position, 6) standing, 7) texting in a standing position, 8) walking on a path, 9) walking in place, and 10) writing in a standing position. The activities are repeated 5 times each in 4 different locations ($P1$ through $P4$) in our lab (9m × 6m). See Figure 2.13. Overall 60+ minutes worth of activities are recorded for analysis.

Figure 2.13 shows a map of the exciter and tag locations. The exciter power is set at 15dBm. The Tx tag keeps transmitting packets with BP header alternating between 3 phase slots. Characteristic phase offsets ϕ_k for the three slots are $-\pi/2$, 0 and $\pi/2$. The two Rx tags are connected to external data logger ADCs that record the voltage at the output of the envelop detectors on the host computer. The study participants conduct the suite of activities in 4 locations indicated by human figures in Figure 2.13. The dashed line indicates walking and running paths for activities #2 and #8.

2.5.2 Activity Recognition

After collecting the BCSI data using the data logger, we have applied Convolution Neural Networks on Multichannel Time Series for Human Activity Recognition (CNN for HAR)

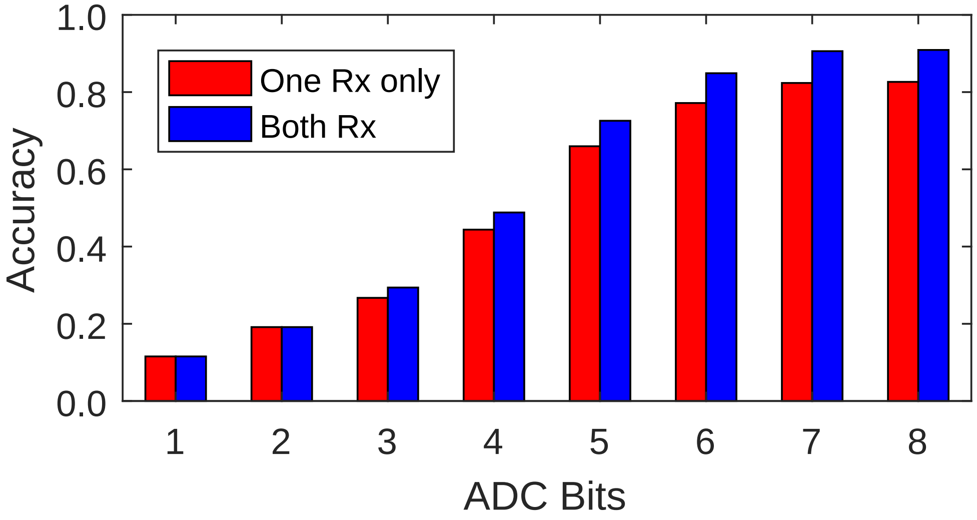


Figure 2.16: Impact of ADC resolution and number of tags on average activity recognition accuracy

to the aggregated data using available open source code [183,184]. We adopt the commonly used architecture of the CNN used for hand gesture recognition [35]. Since the input and output dimensions are simpler in BCSI based activity recognition, we have simplified the number of feature maps and sizes of convolution kernels. We have chosen the parameters in CNN $\kappa = 1, \alpha = 2 \times 10^{-4}, \beta = 0.75$ and followed the rules of thumb in [90] to choose other parameters.

To obtain activity features from BCSI for use as the input to the CNN, we construct the features as a time series of amplitude values measured in each phase slot. See the explanation earlier in connection with Figure 2.5. Figure 2.14 shows an example of how the features are created based on recorded ADC values. First the DC component is removed from the actual recording (subfigure (a)). Then, the sequence of values for each phase slot ϕ_1 through ϕ_3 are used as features (subfigure (b)). Not shown in the figure, the DC component values are also used as the fourth feature.

The location $P1$ in Figure 2.13 is selected as training point, and $P2$ to $P4$ are used for testing. We choose one of the participants as the training subject and rest as testing. Figure 2.15 shows the average recognition accuracy for each of the activities. Note that

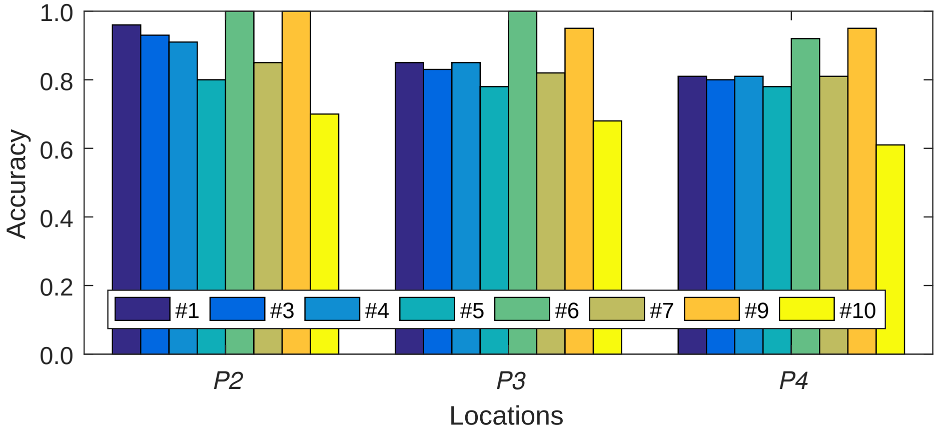


Figure 2.17: Accuracy when different training locations are used

for the running and walking activities (#2 and #8) there are no separate training and testing locations as the activities are based on paths. The overall accuracy is quite high, on average 90% across all cases. Of course, when a single receiving tag is used (Rx1 or Rx2), the accuracy is somewhat lower (on average 84% with only Rx1, and on average 80% with only Rx2). When they are combined, the accuracy improves. Clearly, more the number of tags more the accuracy (Section 2.2). Some activities such as falling, standing, walking in place, and walking on the path achieve close to 100% accuracy. We expect significantly improved overall performance when more tags are used that we expect to be a common case in a typical *BARNET* deployment.

It is important to evaluate how many ADC bits are sufficient. The ADC size influences the complexity and power requirements of the tag and also imposes burden on the communication capacity. Figure 2.16 plots average accuracy over all tasks for different number of bits for two cases (one tag and both tags). Note that the accuracy saturates beyond 6 bits.

BARNET maintains a good accuracy (>80%) even testing points are further away (close to 3 m) from the training point. Figure 2.17 shows the average accuracy when location $P2$ through $P4$ are used for training.

2.6 Conclusion

BARNET extends capabilities of passive RF tags to a different regime. They not only are able to perform tag-to-tag communications under conditions of very low modulation index, but also they are capable of channel measurements of the backscatter channel that correlates very well with environmental changes around the tags. This latter ability translates to human activity recognition. *BARNET*'s recognition accuracy is competitive with active radio-based techniques proposed in recent literature, while *BARNET* is able to operate using harvested power from the externally provided RF signal and using only backscatter-based communication.

Chapter 3

Phase-based Ranging of RFID Tags

RFID (Radio Frequency Identification) technology has matured over the years as a means for automatic, low-cost identification of objects at short ranges. RFID has been playing an increasingly bigger role in inventory management, logistics and access control. Typically, the RFID system consists of one or more readers and numerous tags attached to objects to be identified. Tags communicate with the reader using backscatter communications. Backscattering works by modulating a continuous wave (CW) RF signal emitted by the reader and incident on the antenna of tag. The modulation is achieved simply by changing impedance levels seen by the antenna, thereby changing its reflection coefficient. The minimal intelligence needed to achieve this requires very little power and this power is supplied by the RF signal itself (in the case of so-called ‘passive’ tags). The reader is also able to interpret the modulated signal backscattered by the tag and can issue specific instructions to the tag about how to respond via standards-compliant protocols such as Class 1 Gen2.

Most applications of RFID can benefit tremendously from an ability to accurately localize the RFID tags attached to objects. This can enable a whole new paradigm of applications related to the “Internet of Things.” However, limited capability on the part of the RFID tags themselves limit the range of techniques that can be used and their accuracy.

Much of the work so far has targeted using received signal strength (RSS) for localization either directly or indirectly (e.g., [108, 136]). However, use of RSS is notoriously unreliable as the tag orientation, antenna gain or multipath can influence the RSS significantly. Use of 'reference' tags [195] in the vicinity alleviates this issue to some extent, but this increases deployment effort and may not always be practical. Similar issues arise in using parameters that are indirectly related to RSS, such as read rate (fraction of tag read attempts that are successful). Recent efforts have thus focused on using other radio features such as angle-of-arrival (AoA) [27] or signal phase [69, 94, 109, 166, 167] for localization. Though these techniques tackle the orientation and gain problem for the most part and achieve a high degree of accuracy, multipath may still be an issue. The most serious limitation of these techniques is that they require complex set up, such as dense and carefully positioned antennas and/or profiling studies; but still provide limited read ranges.

In this work, we develop a localization technique that addresses the above limitations using a ranging method based on basic radar principles. The idea is to use *phase difference of signals in the frequency domain* to determine the distance between the reader antenna and the tag. Once ranging is done, the actual localization is straightforward and can use well-known trilateration principles using multiple antennas/readers. The ranging uses principles of frequency-modulated radar. Reflected CW signals from the tag at different frequencies will produce a phase differences depending on the difference between the frequencies (Δf, known) and the intervening distance (d, unknown). The distance d can be calculated by knowing the phase difference ($\Delta\phi$, measured). This technique is invariant to the orientation of the tags and immune to multipath for the most part. The basic principles do not depend on the orientation or RSS so long as the RSS is strong enough for the reader to successfully receive the signal from the tag. So long as a line-of-sight (LOS) is present, multipath is mitigated as frequency diversity is used.

While the above ranging principle (sometimes refered to as *frequency-domain phase-difference of arrival or FD-PDOA*) has been known for some time [109], thus far it was not practical to apply this technique to localize RFID tags using commercially available readers. But this bottleneck is removed now due to two key enablers. First, popularly used standards-compliant Class 1 Gen 2 tags use frequency diversity by default. Thus, many randomly chosen frequency channels (out of 50 possible) are enabled during the interrogation phase making application of FD-PDOA realistic in common deployments. By the FCC, regulation each channel must be used equally and channel occupancy is limited to 400 ms in any 10 second window. Second, current generation readers (e.g., ImpinJ's Speedway series, Motorola's FX series [76, 105]) allow reading of low level signal details (e.g., phase of the backscattered response) using standard API support [75] such as LLRP and customized protocols. Thus, measuring of phase no longer requires specialized equipments.

We specifically use the above ranging technique for a targeted problem – localizing shopping carts in a supermarket. Shopping cart localization is deemed important to track customers to assess their interests and also to deliver targeted ads.

In contrast with related approaches of shopping cart localization where active devices are used on the cart (such as RFID reader [100, 196] or Zigbee devices [23, 58]), the proposed method approach is significantly cost-effective. This is because the cart carries one or more passive tags only and thus does not need to use any battery. We believe that the proposed approach is useful in large supermarkets such as Walmart [112, 129] where significant RFID deployments are planned.

In the rest of this chapter is organized as follows. In § 3.2, we describe the preliminaries. In § 3.3 we describe the experimental details and evaluation. § 3.4 discusses the chapter.

3.1 Related works

Early RFID tag localization techniques were based on the RSS [69, 136] or read count [97, 196]. In majority of RSS based techniques use reference tags to overcome vulnerability of RFID's RSS, those were achieved high accuracy but installing dense reference tags for localization RFID tags are not realistic. In read count based techniques, they were very successful in detecting existence in reader's vicinity and robust in performance but accuracy cannot reached high enough [97,196].

In increasing popularity of phase study [27,69,94,109,166,167], the RFID's localization accuracy greatly improved. Literatures those exploit multi antenna's response called "angle of arrival (AoA)", they differentiate phase response for each antenna to locate the tag. In AoA approaches, multi path responses were challenged due to multiple antenna cannot distinguish multi path effects from LOS response [27,69,109].

Literatures those exploit RADAR technique, "FD-PDOA" achieved high accuracy. Wang et al's robot assisted phase based localization technique overcome multi path issue by building multi path profile with pre-installed reference tags [166,167], dense population of reference tags required to be installed, but they were able to overcome multi path effects.

Liu et al's proposed phase based locationing without reference tags [94], they achieved centimeter accuracy in their lab environment, however, multi path effect was not considered and range was very limited, they hypothesis RFID tags orientation was irrelevant to phase response, and showed 360 degree results, it is true that RFID tag in the position of LOS is guaranteed and multi path effect are limited, however, it is obvious as they change the height of tag they will observe phase response is different to the orientation of the RFID tag, it is called "bore sight effect" in the field of RFID industry, when the tag and antenna are perfectly aligned, they always performs beautifully, however their align is over 5 degree they start to show outliers to the theory.

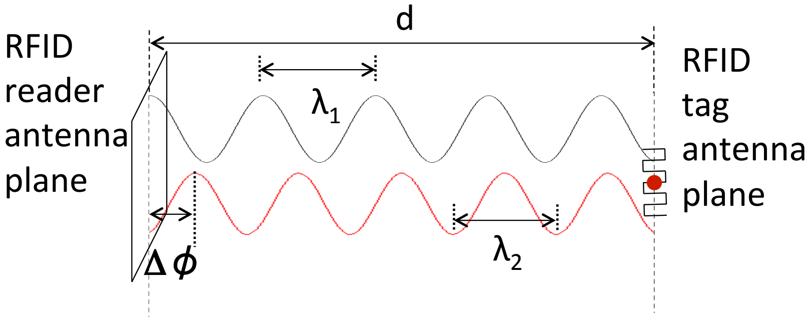


Figure 3.1: Bakscatter communication illustrating the phase difference-based ranging method.

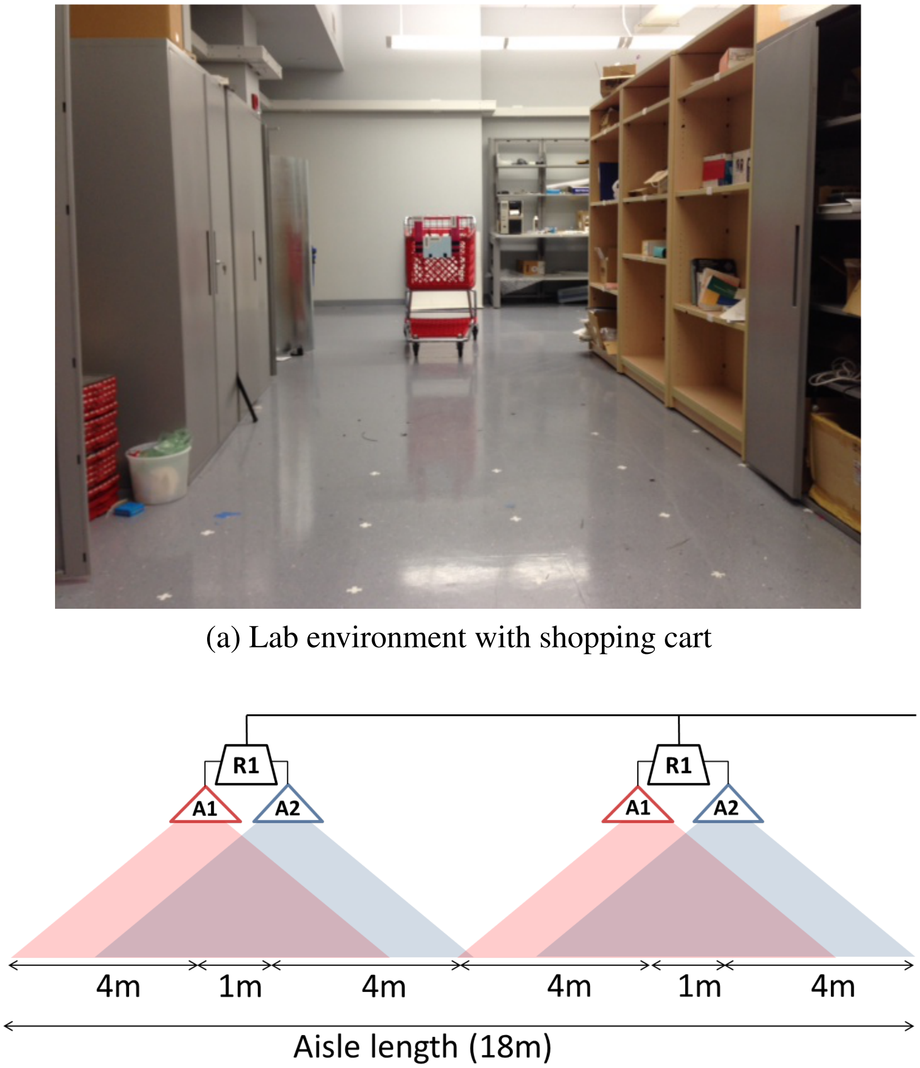
3.2 Preliminaries

In this section, we describe the preliminaries of the FD-PDOA technique. See Figure 3.1. Assume, the transmitted CW from the reader has frequency f ($= \frac{c}{\lambda}$, where c is speed of light and λ is the wavelength). The distance between the reader and the tag is d. Then the phase of the received signal ϕ with respected to the transmitted signal is given by

$$\phi = \frac{2\pi}{\lambda} \times 2d = \frac{2\pi f}{c} \times 2d. \tag{3.1}$$

Sampling the phase at two different frequencies (f_1 and f_2) provides

$$\phi_1 - \phi_2 = \frac{4\pi}{c} d(f_1 - f_2). \tag{3.2}$$



(a) Lab environment with shopping cart

(b) Reader deployment

Figure 3.2: System setup for evaluation.

Finally, solving for d gives

$$d = \frac{\Delta\phi}{\Delta f}\frac{c}{4\pi}. \tag{3.3}$$

Note that d is proportional to $\frac{\Delta\phi}{\Delta f}$. In commercial readers and standards compliant Class 1 Gen 2 tag environment, the *phase-frequency slope* $\frac{\Delta\phi}{\Delta f}$ can be constructed during the tag interrogation phase as the reader frequency hops in a random fashion, staying at each frequency for 400ms.

We have observed that due to experimental errors and phase uncertainties introduced by non-negligible antenna cable lengths a calibration factor α is needed for Equation 3.3. This calibration is done experimentally using the specific reader set up to be used:

$$d = \alpha \frac{\Delta\phi}{\Delta f} \frac{c}{4\pi}. \tag{3.4}$$

3.3 Experimental Details and Evaluation

3.3.1 System Details

one or two linearly polarized directional antennas with beam-width approximately $120°$ for all experiments. The tags used are typical off-the-shelf Class 1 Gen 2. The reader works in the frequency band $902 - 928$ MHz, that is split into 50 channels of width 500 kHz, hopping between them randomly with a dwell time of shorter than 400 ms at each channel. Per FCC requirement each channel must be used in equal proportion.

Some experiments use a multi-reader environment. A tag can independently be interrogated by and respond to two different readers using standard anti-collision mechanisms to avoid conflict (e.g., use of orthogonal channels). Even if it is just being interrogated by only one reader, another in the vicinity can simply be in 'listen' mode and collect measurement samples. This provides more samples and improves localization efficiency noticeably. *Software:* The software runs on a Windows PC using C#. It connects to the RFID reader using the LLRP (Low Level Reader Protocol) protocol. API is also opened to Java/C/C++, no additional software needed for ImpinJ devices due to ImpinJ's open SDK policy.

System Setup: The reader is mounted at the ceiling at the height of 3.75m with the antenna directed vertically downwards. A shopping aisle is simulated in the lab with wooden and metal shelves on the side. The tags are placed at an horizontal orientation (so that they receive the maximum possible signal) mounted on a shopping cart (height about 1m). See Figure 3.2(a). Note that this orientation is enough for testing as shopping carts can only rotate along a vertical axis in a normal use.

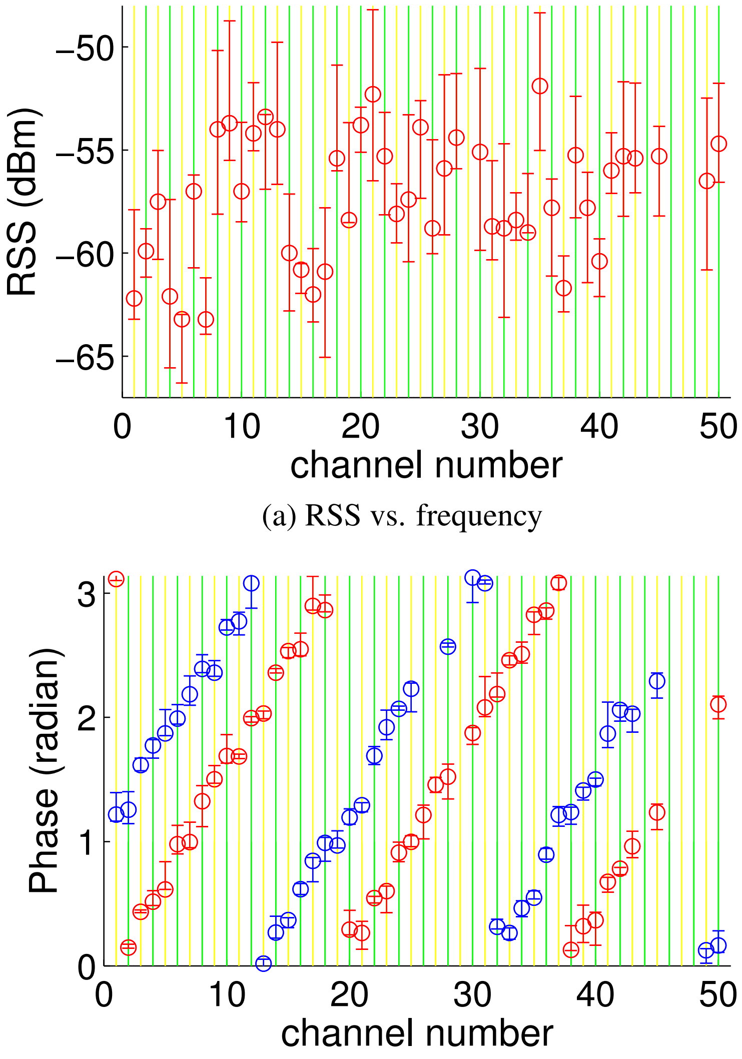
We have independently verified that the experimental results are invariant to the actual orientation of the tag so long as it remains horizontal facing the antenna. Otherwise, while the FD-PDOA technique does not fundamentally depend on the actual tag orientation, the tags may fail to respond in many channels when the incident RF signal is received at a slanted angle on the tag antenna. This is simply because the received signal at the tag may not be sufficient to 'wake up' the tag or the reflected signal is not strong enough for the reader to receive correctly. If many channels fail to respond, it introduces significant measurement noise and makes the slope calculation prone to error.

For the ranging experiments reported momentarily one reader with one antenna is sufficient. For the localization experiments reported at the end, we have used two readers each with two antennas mounted in a linear fashion along the center of the aisle. See Figure 3.2(b).

Note that we deal with two different but related distances. One is the actual reader-to-tag distance (d), measured from the center of the tag to the center of the reader antenna. The other is the 'floor-level distance,' which is the same distance, but projected horizontally on the floor. The floor distance is of practical use in our application, though the distance d is what is directly estimated.

3.3.2 Ranging by Estimating Phase-Frequency Slope

As a demonstration of the power of the technique we first show how the phase-frequency slope $\frac{\Delta\phi}{\Delta f}$ behaves at a specific distance d. For this demonstration, d is fixed such that the floor level tag-reader distance is 2 m. The tag is interrogated at each channel 1,000 times. The statistic of RSS and phase values recorded at the reader at different frequencies are shown in Figure 3.3 with error bars showing the min-max range. (The two colors in the phase plot actually corresponds to the values seen at two different antennas that are separated by 1m.) Note the significant variations of RSS at different channels as well as variations within the same channel due to fading, for example. On the other hand, the phase is reasonably stable. Further note that the linearity of the $\frac{\Delta\phi}{\Delta f}$ relationship evident from the plot (Figure 3.3(b)). The slope of this line has a straightforward relationship to distance (Equation 3.3).



(a) RSS vs. frequency

(b) Phase vs. frequency

Figure 3.3: RSS and phase (in radian) for the 50 channels for a specific (2 m) floor-level distance between the reader and the tag.

Note also that the $\frac{\Delta\phi}{\Delta f}$ line is in fact segmented. The segments repeat as half cycles in interval $[0, \pi]$ are completed as the frequency is increased.[1] Ideally, these line segments should be perfectly parallel. Any difference in slopes in the experimental data is solely due to measurement errors or impact of stray multipath, where a NLOS (Non-line-of-sight) path may dominate for certain frequencies. However, so long as a majority of the frequencies

[1] The phase ϕ is assumed to vary between 0 and π only, as this is the way the reader reports the phase information. Phase values larger than π is reported as $\phi - \pi$.

report the LOS path, the influence of the NLOS paths in the final estimate can be overridden (see below).

To determine the $\frac{\Delta\phi}{\Delta f}$ slope from the experimental data we use a sequence of simple data processing steps:

1. The median phase value of each frequency (channel) is chosen among those reported as representative.[2]

2. The phase-frequency values are clustered to identify the segments. Segments are then numbered 0, 1, 2, etc. with increasing frequency values.

3. All frequency values are 'translated' so that a *single* line is formed (as if aligning all the segments in a single line). This is done simply by adding $k\pi$ from the chosen phase value at step 1 for each frequency belonging to the k-th segment.

4. Finally, linear regression is used to determine the slope of this line. This slope determines the distance d per Equation 3.4.

There are a few practical issues to consider. While the plots in Figure 3.3 use all channels, this may not be realistic always. Also, sometimes responses from specific channels may not be available (due to fading etc) even when these channels are scanned. From experience we have found that responses from at least 5 different channels are needed to establish a reasonable level of confidence on the value of the slope, though clearly more channels provide a higher degree of confidence in the slope estimation. Since channel dwell time is 400 ms, the tag (shopping cart) must be stationary for at least ($\approx$ 2s) for 5 different channels to be sampled. The need for accurate localization is the highest when the cart is stationary for a longer periods time, e.g., several seconds or longer. We believe that the 2s bound is acceptable for the application at hand. Furthermore, the dwell time can be shorten to increase the frequency of channel hopping, which results in faster decision on localization.

[2] We experimented with other statistical measures, but median works quite well.

The above experiment is repeated at different floor-level distances. For each distance, different position of the cart is used. The phase-frequency slope (as determined above) is used to estimate first the reader-tag distance which in turn determines the floor-level distance. The CDF of the estimation error is shown in Figure 3.4. Note the median error is limited to about 5cm with 90-percentile error to about 15cm. This is competitive or better than the errors using other phase-based mechanisms recently reported in literature (e.g., recent work in [94]), while they require more complex set up and also provide limited range (up to only about 1m).

3.3.3 Motion Filtering

It is clear that when the shopping cart is in continuous motion, the phase responses will not form a straight line due to continuously changing location. The responses could be somewhat random. We have developed a simple heuristic to ignore these periods of motion so that all localizations concentrate on brief stationary periods when linear phase-frequency plots are possible. This 'motion filtering' continuously tracks the standard deviation of recently reported phase values for each frequency. When the average standard deviation falls within a threshold for at least 5 frequencies, the distance is estimated and is continuously refined as more samples are collected.

3.3.4 Localization Performance

Finally, we employ the ranging technique described above to localize a shopping cart using the set up described in Figure 3.2. The two reader antennas for each reader makes independent range measurements and these measurements are combined using trilateration to localize the tag (shopping cart). Since the current set up is limited to only two readers, trilateration is possible only if the cart moved on a straight line along the center of the aisle. This is a limitation of our current setup and is not a limitation of the technique.

The results are shown in Figure 3.5, where five sets of tests are done at $0-4$m floor-level distances at 1m intervals (the actual reader-tag distance is higher as noted in the figure). The estimated distance d is plotted across actual distance. The error bars show the range (min-max) of errors for 100 tests performed for each distance. The median error is separately shown which is only about 10cm again demonstrating excellent localization performance.

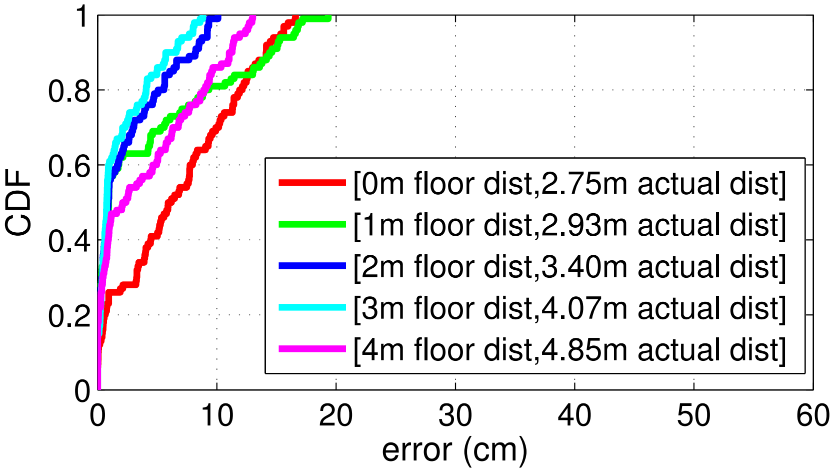


Figure 3.4: CDF of distance estimation error for different distances.

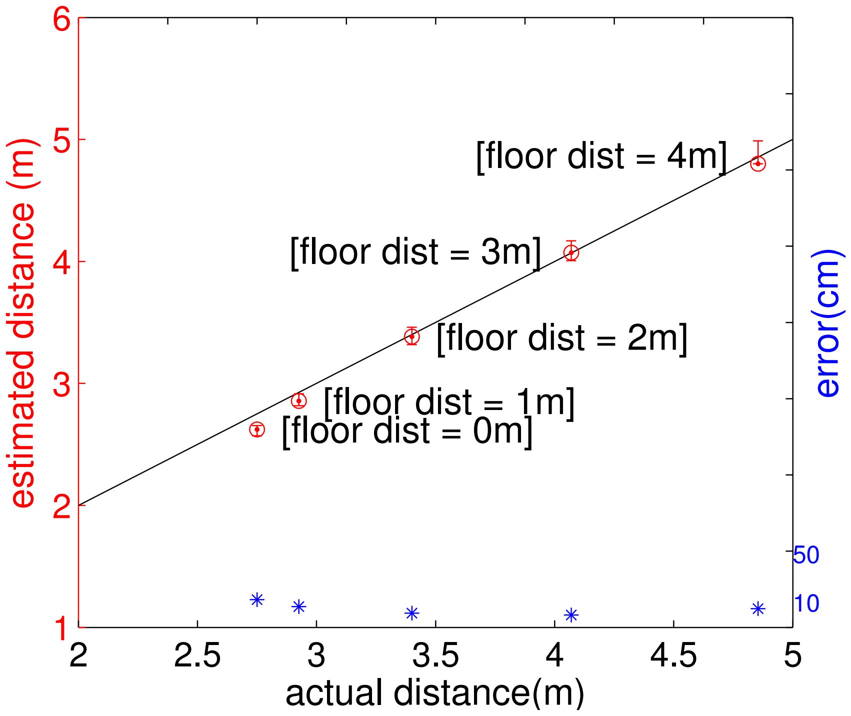


Figure 3.5: Localization performance at 5 different floor distances as shown.

3.4 Conclusion

While phase-based methods of localizing RFID tags have recently gained popularity, they all require careful and dense deployment of antennas or additional complex set up. We have demonstrated that a standard RFID set up can exploit phase to perform very accurate ranging just by using a single antenna by exploiting FM radar principles. The accuracy is competitive or better with better operating ranges seen in literature while requiring a straightforward set up. We have experimentally demonstrated median ranging accuracy of about 5 cm and localization accuracy of about 10 cm at distances over 4 m. Our ongoing work is focusing on deployment and refinement of the proposed method in realistic environments beyond the lab set up presented here.

Chapter 4

A Foveated Video Streaming Service

Various industry analysts [116, 133] report that over half (and projected to be over 80 percent in near future) of the Internet traffic during evening peak hours is from streaming video services, such as Netflix and Youtube. Specifically, the so-called ‘cord-cutters’ contribute significantly to this consuming over 100 hours of video per month per household, on average. The cord cutters’ proportion is increasing fast. The average quality/resolutions of available videos are also improving fast. However, there is hardly enough available bandwidth. Roughly, the average bandwidth available per household in many developed countries is barely enough to stream only two HD quality videos concurrently [113, 134]. While content providers are intent on making available higher-resolution 4K videos for streaming, and display prices are falling fast, no ISP currently can sustain the bandwidth needed for such videos at scale [128]. This is even at only a slow frame rate (30 fps) and with the most aggressive compression.

Similar bandwidth concerns apply to mobile platforms, albeit at a different scale. More and more media is now consumed on mobile devices and often outside the home/work networks. Both cellular providers and ISPs employ different rate plans and data caps, making significant media consumption very expensive for the end user. They are also

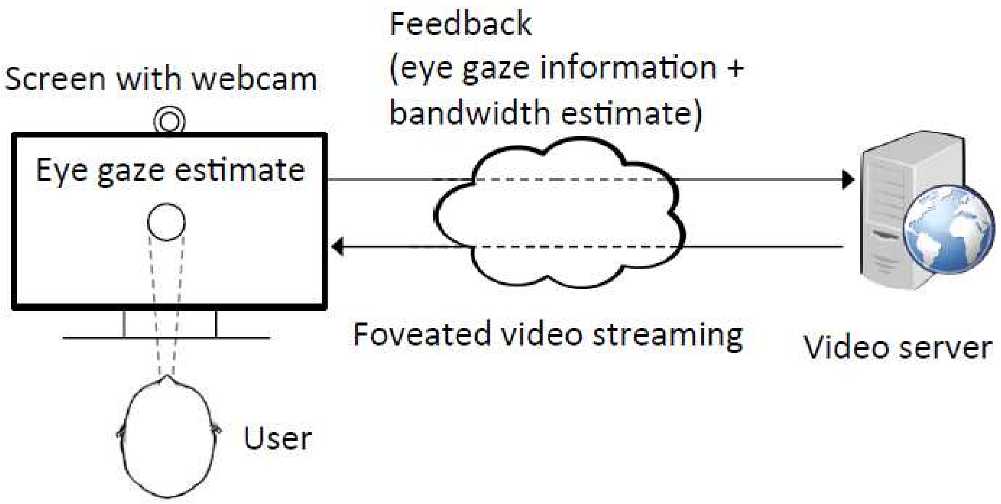


Figure 4.1: Overview of the proposed foveated video streaming system.

widely understood to employ various forms of traffic shaping and differentiations to reduce stress on their networks (e.g., [49]) that in turn affects end users' quality of experience.

We propose to alleviate this bandwidth crunch by developing a new, variable resolution video streaming service that compresses the video based on where a person is looking with their central or *foveal* vision – a behavior known as *foveation*. The human visual system (HVS) samples information very non-uniformly; sampling is very dense at the center of our visual field (fovea) but drops of roughly quadratically with distance from the center. This decrease in sampling rate explains why human vision is blurred in the visual periphery – we notice this by keeping our gaze fixed straight ahead and trying to read something out of the corner of the eye.

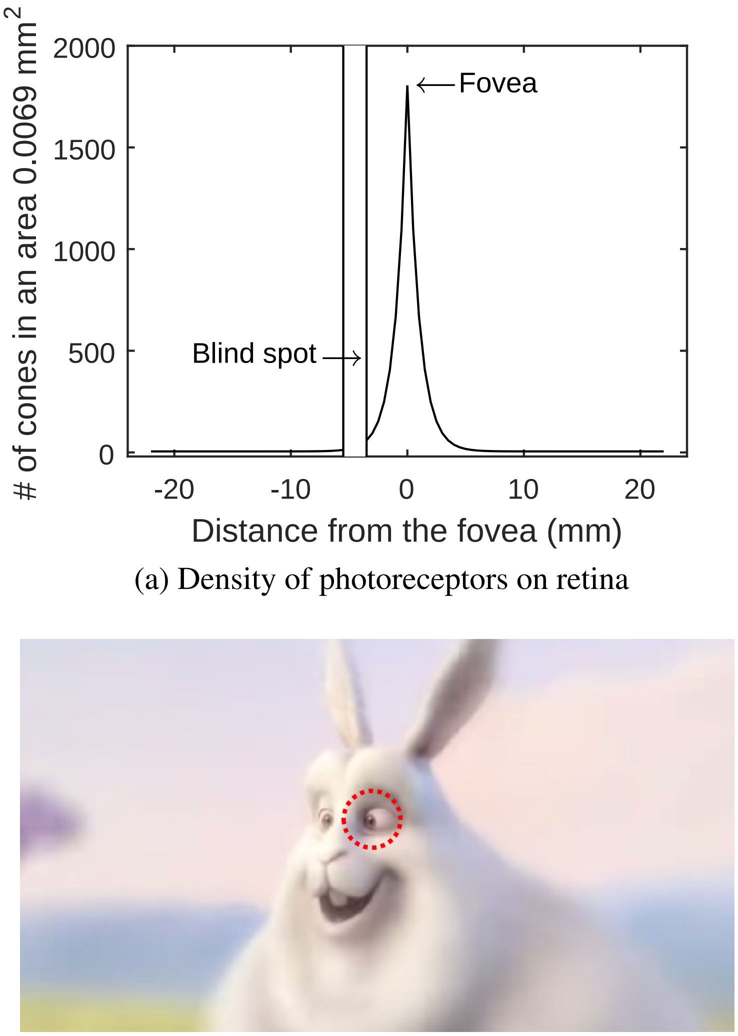
Our idea is to compress video so as to roughly match this sampling limitation of human vision, under the assumption that high-resolution video falling on the low-resolution peripheral retinas of viewers will be wasted. While a number of techniques for achieving a similar compression by exploiting the aforementioned property of the human visual system do exist (see, e.g., [32, 61, 121, 176]), existing methods of video streaming have failed to exploit these advances. The key reason is that it is usually unknown where a person's gaze will be 'pointing' at any given time while watching a video. This generally requires a continuous feedback of the gaze information to the video streaming server so that the server can deliver an appropriately compressed video taking into account the gaze information and available bandwidth estimate. See Figure 4.1. Determining gaze position typically requires a separate 'eye tracker' that is not a commodity device. A more realistic and scalable solution is needed for wide-spread adoption of foveated video compression for video streaming over the Internet.

With this backdrop we make three contributions in this paper:

1. We develop a multi-resolution video coding approach. The approach is 'scalable' in that only a limited number of copies of the video at different resolutions need to be stored at the server (Section 3.1). The coding approach is designed to match the error performance of an eye tracker built using a commodity webcam (Sections 3.2, 3.3).

2. We demonstrate that the technique is energy efficient and thus usable in mobile devices (Section 3.4).

3. We develop a methodology for performance evaluation of such a system. We use this methodology to perform a comprehensive user study (§ 4.3). The user study shows that significant bandwidth savings are possible by adopting such foveated video streaming without degrading perceptual video quality.



(a) Density of photoreceptors on retina

(b) Example of foveated compression
(gaze focused on eye of the rabbit)

Figure 4.2: Human visual system (HVS)

4.1 Background and Related Work

4.1.1 Concept of Foveation

The light passing through the optics of the human eye projects on the retina and is sampled by the photoreceptor cells – rods and cones. These photoreceptors are non-uniformly distributed over the surface of the retina (See Figure 4.2(a)). The concentration of cones (the specific type of photoreceptors responsible for chromatic vision in good lighting conditions) is the highest at the center of retina (zero eccentricity) in a very small area– called the *fovea*. The fovea occupies only about 2° of the visual field and is roughly the width of your index fingernail at arm's length. The concentration of cones declines almost quadratically with increasing eccentricity (see Figure 4.2(a)). This non-uniform sampling gives rise to a very sharp central vision (also called *foveal* vision) and rapid loss of sharpness as one moves away from the fovea.

When a human observer gazes at a point in a real-world image, a variable resolution image is thus transmitted to the brain. The region around the point of fixation (or foveation point) is imaged onto the fovea, sampled with the highest density, and perceived by the observer with the highest visual acuity. The sampling density, and thus the visual acuity, decrease dramatically with increasing eccentricity. Despite this non-uniform sampling in the human visual system, traditional imaging techniques use uniformly sampled images in rectangular lattices. This is clearly inefficient. More effective would be to roughly match the level of video compression to the non-uniform sampling introduced by the human visual system.

This observation gave rise to a significant body of work on foveated image processing (see, e.g. [48, 78, 119, 175]) targeted around various applications. This also included image/video compression and efficient communication. Clearly, compression losses are well tolerated in peripheral regions as opposed to near the fixation point. This can be utilized to produce variable rate compression and reduce the bandwidth required to transmit the same image for a similar level of user satisfaction. Similarly, in noisy communication environments, foveation provides a natural way for unequal error protection for different spatial regions of the image. Despite these advances, use of fovated compression techniques for Internet video streaming is far from a reality. This is largely due to scalability reasons. First, the compression must be computed in real time on the server side using the gaze feedback. Second, precise gaze feedback typically needs expensive eye trackers that are not commodity and also inconvenient to set up and use. In this work we address the former issue by designing a multi-resolution precoding approach compatible with modern day video servers. Thus, the need for real time computation is eliminated. We address the latter issue by using commodity webcam-based eye tracking. We discuss eye tracking next.

4.1.2 Eye Tracking

Eye tracking techniques are capable of providing very good estimates of the point of fixation needed for foveated video processing. Most common eye trackers today use a combination of infra-red (IR) illumination and an IR camera [68, 104]. The basic concept is to use the light source to illuminate the eye causing highly visible reflections, and the camera to capture an image of the eye showing these reflections [156]. The image captured by the camera is then used to identify the reflection of the light source on the cornea (glint) and in the pupil. Geometric calculations based on the relative positions of these images are then used to calculate the gaze direction. Most commercial eye trackers, such as Tobii [156], The Eye Tribe [154], and SMI [145], estimate eye gaze with high accuracy using the above approach. However, these devices are not commodity and are often large and expensive.

Advances in computer vision have now enabled estimation of human gaze direction using images of the eye captured on a webcam-class camera. While the accuracy of these techniques is generally poorer than IR-based trackers, these techniques could be very useful in practice as webcams are commodity and many end-user devices are already equipped with webcams.

Several related webcam-based eye-monitoring techniques have been recently extended to mobile smartphones and tablets [103, 178]. This trend is expected to continue. This generally establishes the potential of using eye tracking on commodity video platforms.

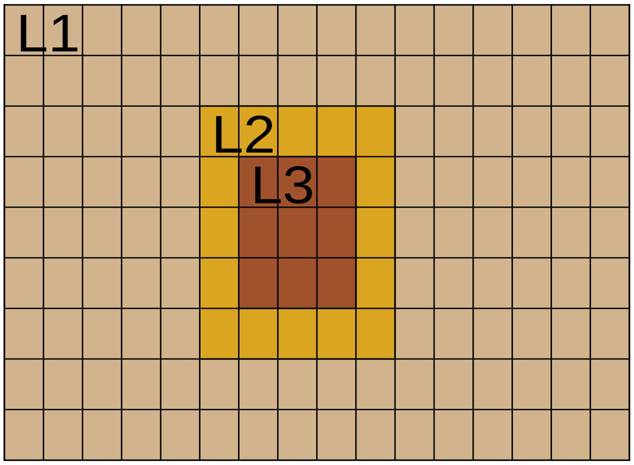
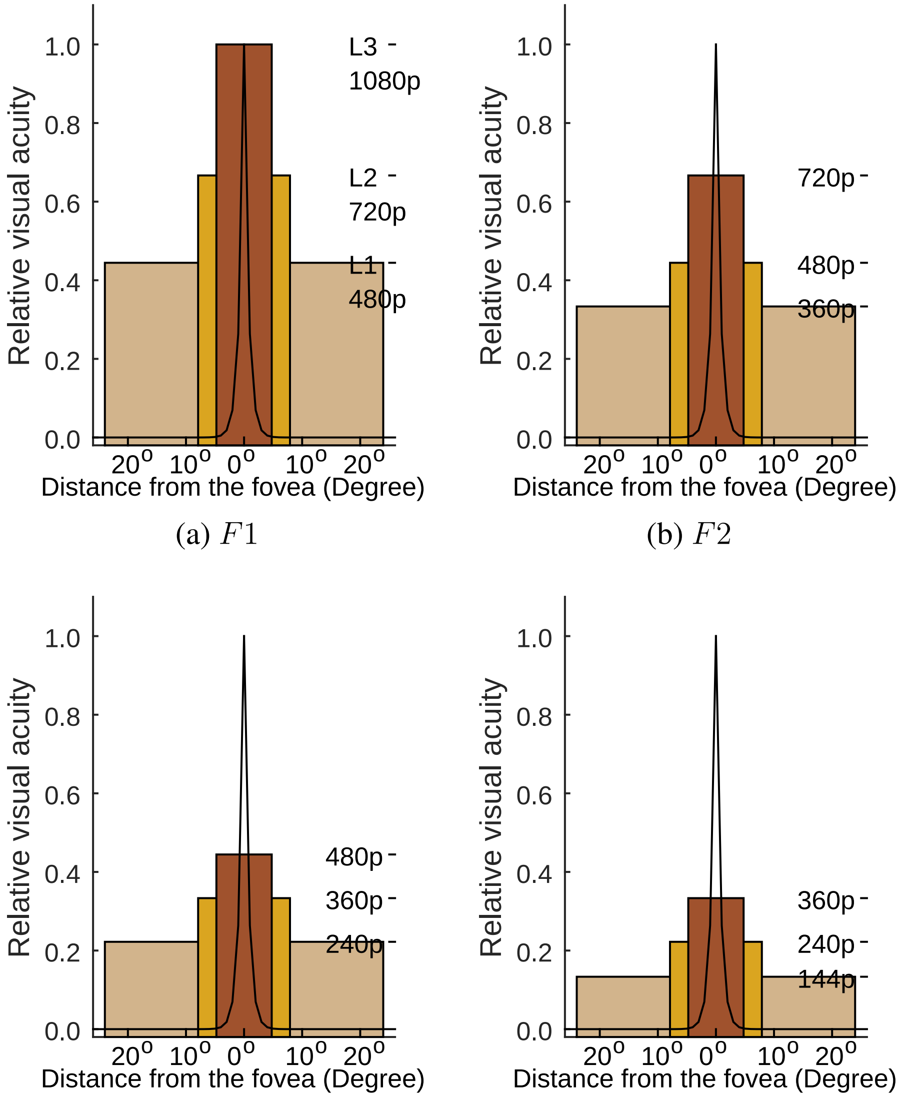


Figure 4.3: Multi-resolution coding in a 16×9 grid used in the experiments. The regions L_1, L_2 and L_3 are coded in progressively higher resolutions.

4.2 Design of The Foveated Streaming Video System

The basic design of the foveated video streaming system is similar to a conventional adaptive streaming video system [52, 72]. See Figure 4.1. The client player continuously estimates the available network bandwidth and possibly also available compute capacity on the user device and requests the next chunk of video in the appropriate resolution. The requested resolution is commensurate with the available network bandwidth and compute capacity available for decoding such that the required display frame rate can be maintained. Typically, the video is pre-coded at the server at a set of standard resolutions, as real time encoding could be difficult. The available resolutions are already known to the client at the initial negotiation time and thus the client only asks for one of the available resolutions for each chunk. In the foveated streaming system the basic framework is the same except that the client now feeds back the gaze information periodically to the server.



(a) $F1$

(b) $F2$

(c) $F3$

(d) $F4$

Figure 4.4: Four resolution levels (F_1 through F_4) chosen for the foveated system are shown along with the visual acuity in human visual system. The four levels consume progressively lesser network budget. In each level F_i, the choices of the actual resolution for the sections L_1 through L_3 are made so that they map, in relative terms, as closely as possible to the visual acuity.

4.2.1 Multi-resolution Coding

A design decision had to be made about how the video is to be coded on the server side. Foveated image/video coding is a mature topic (see a review article in [176]). However, the existing techniques are dependent on precise gaze feedback (point of fixation) and the coding is dependent on this point. This makes it hard to pre-code videos, as only a limited number of encodings are possible for a given video for scalability reasons.

Instead, we take a more practical approach. Due to the computational and network delays, the feedback to the server about the gaze information is expected to have a lag. Thus, precise real-time gaze information is not likely available. Additionally, the gaze feedback can only be periodic and not continuous. Thus, in between feedback events the gaze estimate is only approximate. (We study this aspect in more detail in Section 4.2.4.) Given these sources of error, the coding approach we use must be tolerant to errors in the gaze estimate. Based on this observation we take the following approach in order to pre-code videos.

Assume that the screen is split into a rectangular grid – a 16×9 grid is used in all experiments. See Figure 4.3. The choice of this grid size is somewhat ad hoc. 16×9 is similar to several standard screen sizes. Splitting the grid further (e.g., 32×18 etc.) would result in too many grid cells and affect scalability. Each of the 144 grid cell thus produced is now pre-coded in a standard set of resolutions that are commonly offered on Internet streaming servers. These cells are divided in three sections – L_1, L_2 and L_3, coded in progressively higher resolutions. The sections are organized as shown in Figure 4.3 with L_3 in the middle, surrounded by L_2 and then L_1. The general idea is to place L_3 such that the estimated fixation point is at the center.

Other important design choices include: (i) the respective sizes of the sections L_i and (ii) the actual resolutions that each of the L_i's should be coded in. These choices are made such that the offered screen resolution follows the central and peripheral visual acuity in the human visual system (Figure 4.2(a)) as closely as possible. For (i) we choose L_3 as a 3×3 region with L_2 as a unit width annular region around it (Figure 4.3). The rest of the frame is L_1. For (ii) we use 6 different resolutions for individual grid cells (144p, 240p, 360p, 480p, 720p, 1080p) and choose selected combinations of these in different sections

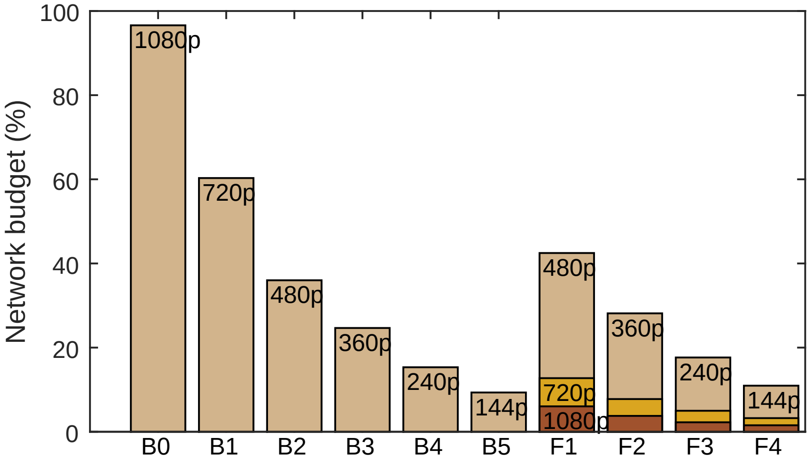


Figure 4.5: Network capacity used by different resolution levels in baseline and foveated players when normalized against uniform resolution 2K video. For the foveated player, the portions of capacity budget consumed by the different layers L_i are also shown.

(L_i). Overall, 4 different coding levels are used - $F1$ through $F4$ - with different choices of resolutions for the sections L_i. The actual choices made are shown in Figure 4.4.

The baseline system chosen for performance bench marking uses 5 different resolutions, from 240p to 1080p. These resolutions are presented uniformly to the entire screen. They are referred to as $B0$ through $B4$. All different choices are again summarized in Table 4.1 for easy reference.

We now compare the network capacity (bits/sec) needed to transmit the aforementioned resolution levels Bi (Fi). To do this we use the capacity needed for 2K video (i.e., 2048×1080p) as the reference, as if consuming 100% of capacity budget. We then determine the bits/sec capacity needed for all different resolution levels and normalize them against the above reference. The results are summarized in Figure 4.5. The capacity numbers are determined via actual measurement of bits/sec captured from real network traces. Thus, they capture the actual load on the network including all packet header and protocol related control packet overheads.

This figure demonstrates the network efficiency of the foveated system. For example, with $\approx 50\%$ network budget, the baseline player can play at only 480p, but the foveated

Baseline Player				Foveated Player				
Res. level	Res.	Net.(%)		Res. level	Net.(%)	L_3	L_2	L_1
B0	1080p	96.6						
B1	720p	60.35	$\longleftrightarrow$	F1	42.48	1080p	720p	480p
B2	480p	36.00	$\longleftrightarrow$	F2	28.14	720p	480p	360p
B3	360p	24.66	$\longleftrightarrow$	F3	17.65	480p	360p	240p
B4	240p	15.33	$\longleftrightarrow$	F4	10.95	360p	240p	144p

Table 4.1: Resolution levels used in the baseline and foveated players.

player can play 1080p for L_3, 720p for L_2 and 480p for L_1. As evaluations will show later, the latter player provides a much improved user satisfaction without consuming any additional resources.

Summary: In the proposed foveated system, the video is pre-coded on the server side with each of the 144 grid cells pre-coded and stored in 6 base resolutions. Then they are 'stiched' appropriately to form the three sections L_1 through L_3 (Figure 4.3) at different chosen resolutions, given the resolution level to be played - $F1$ through $F4$ (Figure 4.4). The positions of sections L_i on the frame is determined from the gaze feedback received from the client. The resolution level is chosen based on the network capacity budget (Table 4.1). The stitching is actually done on the client side in our experimental system for ease of implementation, but server-side mechanisms are possible.

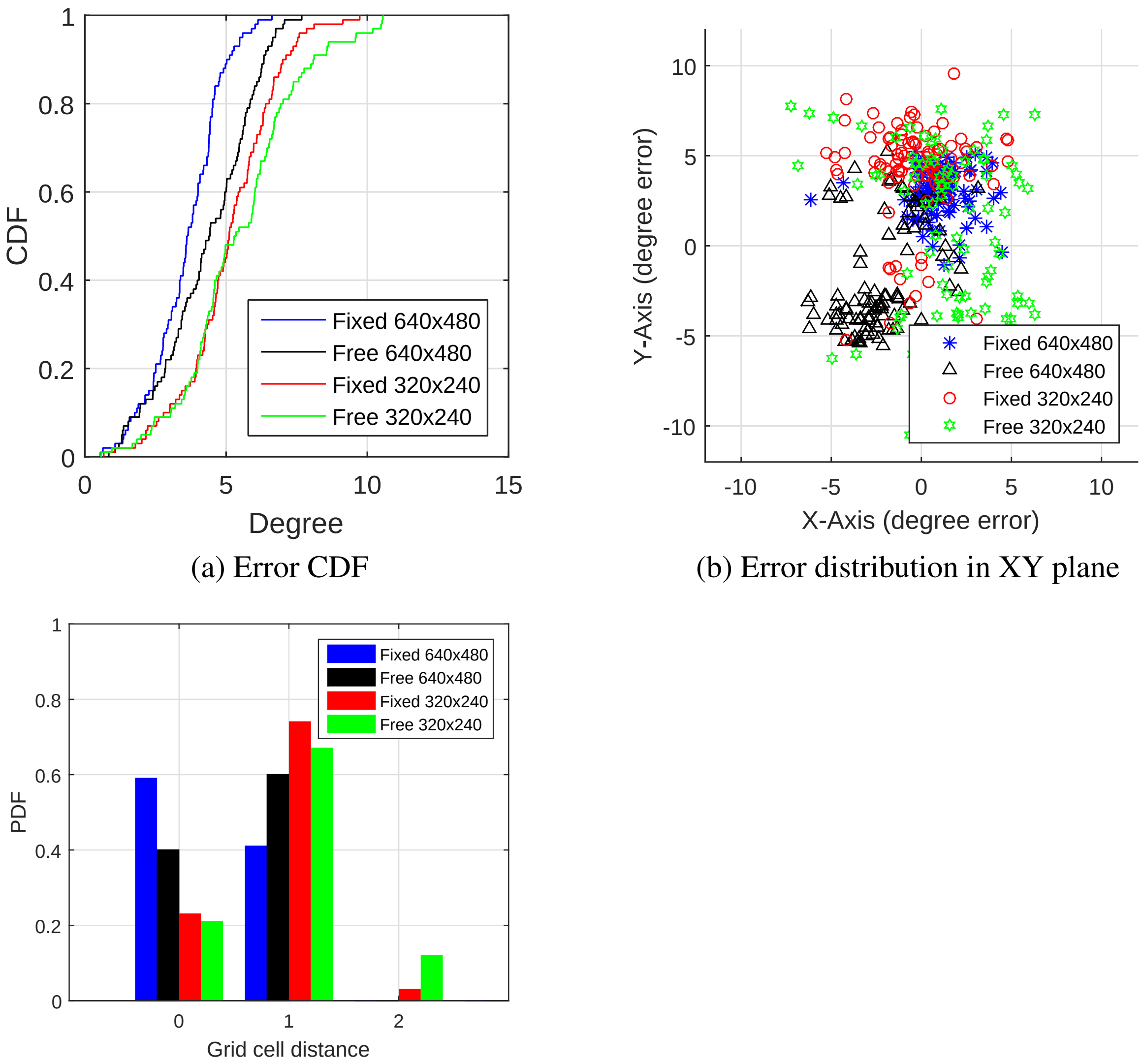
We will now describe the design of the gaze tracker on the client side.

4.2.2 Gaze Tracker Design

Low-cost real-time gaze tracking is at the core of foveated video streaming, as the perceived quality of the video depends on real-time knowledge of the viewer's visual attention. In a real-world service (i) the system should have adequate accuracy while using only inexpensive, commodity hardware; (ii) it should have low computational cost while providing close to real-time service; and (iii) it should be energy efficient overall so that the same general technique can apply to mobile devices. Any off-the-shelf method using a regular webcam allowing for natural head movement (Section 4.1.2) can be used for our service [99,178,192]. High accuracy is not required given the tolerance of our basic design – the highest resolution section L_3 is $\approx 8^\circ$ degrees wide. This is a significant margin as we will see in our evaluations later. We use an open source implementation for our webcam-based gaze tracker [21,60,114,117,146] and show that it performs adequately to the system needs.

The specific off-the-shelf webcam-based gaze tracker we used is called *Gaze-Pointer* [60]. Gazepointer performs in real time (30 frames/sec) and works even with a low-resolution webcam – this makes it widely useful in commodity platforms and also saves energy. It also does not require significant computational load (more on this later).

For the benefit of the reader, we briefly describe how Gazepointer works. More details are in [47]. Gazepointer follows the conventional model-based approach. First, it detects the position of the face at the first frame [162]. Then, when the face is detected, the characteristic points of the face are identified and a parameterized face mask is automatically mapped on the face. Detection of the characteristic points of the face and their temporal tracking are based on an Active Appearance Model [43] with regard to the 3D position of the camera. The eye corners are detected from the 3D face model, then the precise boundary of the pupil is extracted by a technique similar to [91]. The estimated eye position is computed based on the vector difference between the pupil center and the eye corners.



(a) Error CDF

(b) Error distribution in XY plane

(c) Error distribution in terms of grid cell distance

Figure 4.6: Performance of the webcam-based gaze tracker used in evaluation.

In order to calculate the fixation point on the screen, a mapping between locations on the screen and an estimated point must be established through an initial calibration procedure. The calibration works by looking at 9 different points on the screen. Re-calibration may be necessary if the head moves significantly. In our experience the system tolerates normal head movements well, when the user is within normal viewing distance and is engaged in the video on screen. In our user experiments re-calibrations were rarely needed.

4.2.3 Gaze Tracker Evaluation

We evaluate the gaze tracker above for accuracy. For this set of experiments, we use a 1920×1080 resolution 24 inch monitor with a webcam mounted on top. The screen is placed at a reading distance ($\approx 50\,\text{cm}$) from the subject. The subjects are asked to look at 100 points flashed at random locations on the screen and the degree of error is calculated from the point location and the estimated gaze point. We perform two different sets of experiments: (i) with *head fixed* using a chin-support to provide a set of baseline measurements, and (ii) with *head free* that would be the normal operating condition allowing free head motion. Two different webcam resolutions are used – 320×240 and 640×480.

Figure 4.6(a) shows the accuracy of the gaze tracker in terms of the error CDF (cumulative distribution function). As expected, if the resolution is lower or the head moves freely the performance is somewhat poorer. But overall the performance is quite good – with the 90-th percentile error between $6.3° - 8°$ depending on resolution used when the head moves freely. Figure 4.6(b) shows the scatterplot of the degree error for all collected data points separated into the X and Y direction. No systematic bias in any specific direction is observed.

It is important to consider what implications the degree of eye tracker error presented here has for the design choices we made in Section 4.2.1. To do this, we map each point to the corresponding grid cell (out of 144) and determine the error in terms of the grid cell distance (distance 1 is a neighbor cell when cells are considered 8-connected, etc). The distribution is presented in Figure 4.6(c), showing that the 90-th percentile error is within distance 1. Thus, given our conservative choice of the size of L_3 (3×3 grid) there is over a 0.9 probability that the user will perceive his/her central vision with the highest offered resolution.

We note here that with improvements in webcam-based gaze tracking and improved camera resolutions, it would be possible to make a much 'tighter' choice for L_3 for even better network performance.

4.2.4 Technical choices and scientific background for decision

Saliency

Instead of using eye gaze directly to determine visual attention, it may be sufficient to use automatic *saliency detection* as a proxy for gaze position [33, 78, 101, 188]. The idea is to preprocess the video to create a *saliency map* that predicts regions of interest (ROI) that have been shown to correlate with viewer fixations. Various types of saliency have been studied in the computer vision literature, such as saliency maps using low-level image features (e.g., color or intensity) or semantic saliency maps that use higher level features (e.g., face or other meaningful objects) [194]. However, saliency-based techniques have only limited potential in our context. First, the ROI determined via automated techniques may still be quite large in proportion to the screen size. Second, predictions of attention from saliency maps are far from perfect. It is well-known that human attention and fixations reflect the goals of the viewer [186], and these top-down goals are not captured by saliency models. Third, there is often poor agreement in the eye movements of people viewing the same image or video [51, 187], and this variability is also not captured by saliency models. However, saliency detection can still be useful in predicting a few frames ahead, when gaze feedback has substantial lag. It may be possible to learn more personalized, and consequently more accurate, saliency maps from the history of eye movements for a specific user viewing a specific video [130]. If so, perhaps automated saliency detection can be integrated into our foveated video streaming system to further improve performance.

Handling lag

The gaze tracker presented in Section 4.2.2 works at 30 Hz providing an update every 33 ms. On top of this, there is a network delay that could be in the order of 10-1000 ms on the Internet. But Internet latencies are improving fast with the introduction of CDNs by content providers. As a reference, a 2009 study on Google CDNs shows that the 90-th percentile one way delay is about 100ms [88]. To understand how much the eye gaze can move within this time, we analyzed a dynamic eye movement dataset for subjects watching video [101]. From this dataset, we computed the pixel distance that the subject's eye gaze travels within a given interval of time and mapped this pixel distance to the size of the center section L_3. The analysis showed that it took up to an average of $\approx 660\,\text{ms}$ for the gaze point to move outside of L_3 provided the initial fixation was right at the center of L_3. This bolsters our conviction that lag in the gaze feedback is not likely to be a significant problem except in a network with poor infrastructure. Further, various gaze estimation methods (e.g., [55, 86]) can be used to supplement and overcome possible lags.

Buffering

Most streaming systems pre-fetch frames in advance of playing and buffer them at the client. The general goal is to 'ride out' transient network bottlenecks. The amount of buffering is very design specific. Buffering introduces a problem with foveated videos as frames pre-fetched significantly in advance lack gaze feedback and may not be compressed correctly. To alleviate this, we propose to prefetch the entire frame only in the lowest resolution (i.e., L_1 resolution) consistent with the level being played. For example, when the video is being played at level F_2, the entire frame is pre-fetched at resolution 360p. See Table 4.1. The higher resolution parts of the image for L_3 and L_2 are fetched at approximately real-time based on gaze feedback. Further optimizations can be done by using a person's history of eye movements to pre-fetch parts of $L3$ and $L2$ in advance as well. Given that $L3$ and $L2$ present a small part of the network budget (Figure 4.5), we expect this to work well in practice.

Screen size

While the user experiments reported in this paper are done on relatively large screens (desktop monitor), our work is not fundamentally limited by screen size. This is because changes in viewer fixation still occur even for smaller screens. The fovea occupies only about 2° in the visual field (Sec 2.1), an area smaller than a fingernail at typical smartphone viewing distance. Thus, large regions of a phone screen can still be delivered at a much lower resolution without affecting the viewing experience.

Broadly, the only practical concerns are: (1) the angle that a screen subtends at the viewer's eye at a typical viewing distance and (2) degree-wise accuracy of gaze tracking with respect to this angle. Both the 24 inch and 12 inch screens (studied in the paper) subtend approximately $45^\circ - 50^\circ$ degree angle at the eye. For smaller screens, the viewing angle reduces somewhat falling to about 38° for a 6 inch screen, for example. This is due to the shortening of the viewing distance, which we expect will also improve the gaze tracking accuracy proportionately. Thus, the overall improvement is still expected to be quite similar for small screen devices.

Number of viewers

So far we have assumed that there is only a single viewer. Solo video watching is quite common, especially on personal devices. However, in principle our approach can be extended to multiple viewers, although tracking multiple eyes simultaneously does become computationally expensive. Also, if the viewers look at different locations on the screen, the efficiency of the system can diminish as multiple regions would have to be fetched at higher resolution. But several intermediate solutions are possible. For example, our foveated system can be used only when viewer fixations overlap.

This work

Our work here does not attempt to design a complete end-to-end solution including handling of real network lag or buffering issues, which are beyond the scope of this paper. Instead, our goal is to highlight the potential for a commodity-based, scalable foveated approach to reduce the network capacity budget and, in so doing, improve user satisfaction for the same budget. We do this via a comprehensive user study described in the following section.

4.3 User study

We describe a comprehensive user study in this section to determine the perceptual quality of the foveated video streaming. The goal is to showcase the potential for significant network budget reduction for the same perceptual quality, or significant improvement of perceptual quality for the same network budget. We first describe the user study set up and then present the results.

4.3.1 Study Setup

The study was conducted on a lab table top using a 24-inch monitor (1920×1080 resolution, 53.3×29.6 cm physical screen) driven by a general purpose PC (Intel i5, 4GB RAM, Windows 8.1). The subjects were positioned at a comfortable viewing distance (approx 50-60 cm). Thus, the viewing angle was about 50° with each grid cell subtending about 3° at the eye.

Choice of videos and recruitment of subjects

We picked 48 videos of 1080p (1920×1080) resolution from 16 channels of YouTube.[1] From each channel, the 3 most popular 1080p videos were chosen, for a total of 48 videos. Each video was then encoded in multiple resolutions: 1080p (original), 720p, 480p, 360p, 240p, and 144p. Only a 30 sec clip from each video was picked for the actual study.

We recruited 16 subjects, half male and half female, with ages ranging from 22-45 years and with varying ethnic backgrounds. None of the subjects were aware of the nature of experiments to be conducted. The subjects were instructed to watch video clips, and to press a single button when they perceived the video to be of poor quality and to release the button when they perceived the video to be of good quality. Button press times were

[1] Youtube has 19 channels but 3 of them are derivatives. The 16 chosen channels considered are 1) Music, 2) Comedy, 3) Film and Entertainment, 4) Gaming, 5) Beauty and Fashion, 6) TV, 7) Automotive, 8) Animation, 9) Sports, 10) Tech, 11) Science and Education, 12) Cooking and health, 13) Causes and non-profit, 14) News and Politics, 15) Lifestyle, 16) How-to and DIY.

logged for later evaluation. This task was intended to provide a relatively non-intrusive measure of video quality, with longer total button press times indicating a poorer quality viewing experience. Subjects were familarized with the task before the study by having them view a sample video (not part of the data set) at different levels of quality during a practice session. Nevertheless, 3 out of 19 tested subjects were excluded from the study for not following the instructions, leaving the data from only 16 subjects for analysis.

Structuring subjects and videos

The experiment was counterbalanced such that each subject was randomly paired with another subject for the purpose of evaluation. One subject of the pair watched a random half of the 48 clips in baseline mode and the other half in foveated mode The other subject watched the same sequence of clips - but in the opposite modes. In other words if one watched the 48 clips in the following fashion:

```
CLIP1 - CLIP2 - CLIP3 - CLIP4 - --- - CLIP48
Foveat - Base - Foveat - Base - --- - Base
```

the other would watch the clips in the sequence:

```
CLIP1 - CLIP2 - CLIP3 - CLIP4 - --- - CLIP48
Base - Foveat - Base - Foveat - --- - Foveat
```

We also made sure that the same content category was shown equally in the baseline and foveated modes. This was done because a users' sensitivity to poor video quality may depend on the video content.

Sequencing video resolutions

To measure perceptual video quality we first needed a reference. The reference would ideally be an uncompressed version of the video. We used the 1080p baseline video ($B0$ in Table 4.1) as a proxy for this ideal, which we denote as U. The idea is to show a segment of the U video either once at the start of the clip, or to periodically show the U video to subjects so as to enable them to recalibrate their expectation of good video quality and to obtain perhaps meaningful feedback about drops in quality when compressed versions are shown. More specifically, for each 30 sec video clip the two sequencing conditions were:

1. *Uncompressed after every transition:* Alternate between uncompressed and compressed several times for each video clip. For example, the 30 sec clip might be played in 8 segments (shown below), with a different resolution used for each 3.75 sec segment (stitched together to make a continuous video stream).

   $U \rightarrow B2 \rightarrow U \rightarrow B1 \rightarrow U \rightarrow B4 \rightarrow U \rightarrow B3$

   This condition models varying resolutions in the compressed versions ($B2, B1$ etc) due to a varying network capacity budget, but U is always shown before the compressed version to recalibrate the subject.

2. *Uncompressed once at start:* Show uncompressed segment only once at the start of a clip, followed by compressed segments varying in resolution. For example, the 30 sec clip might be played in 5 segments (shown below), with each 6 sec long segment having a different resolution.

   $U \rightarrow B3 \rightarrow B1 \rightarrow B4 \rightarrow B2$

   Note that in this condition U is shown only once at the start of the clip for subject calibration, followed by segments varying in resolution that might reflect varying network capacity budget.

The two sequencing methods enable different evaluations of the perceptual impact of the varying video resolutions. This is because a subject may perceive the same quality on the same clip differently depending on what was shown immediately prior.

For each subject, the first 24 video clips are shown under condition (1) and the second 24 are shown under condition (2). See Figure 4.7 for a graphic depiction. Note that each subject watched half of the videos with baseline compression ($B1, B2$, etc.) and other videos with foveated compression ($F1, F2$, etc.), and we ensured that the same sequences were used for both. For example, if a subject watched a video clip as $U \rightarrow B3 \rightarrow B1 \rightarrow B4 \rightarrow B2$, his/her paired viewer watched the same clip as $U \rightarrow F3 \rightarrow F1 \rightarrow F4 \rightarrow F2$. The actual sequence of resolutions, e.g., $F3 \rightarrow F1 \rightarrow F4 \rightarrow F2$, was randomly selected.

Figure 4.7 also shows button presses from example traces. Note that the user sometimes pressed the button after a slight delay following a drop in quality, and sometimes delayed slightly their release of the button after the quality improved. Our analysis later attempts to correct for this lag in reaction time.

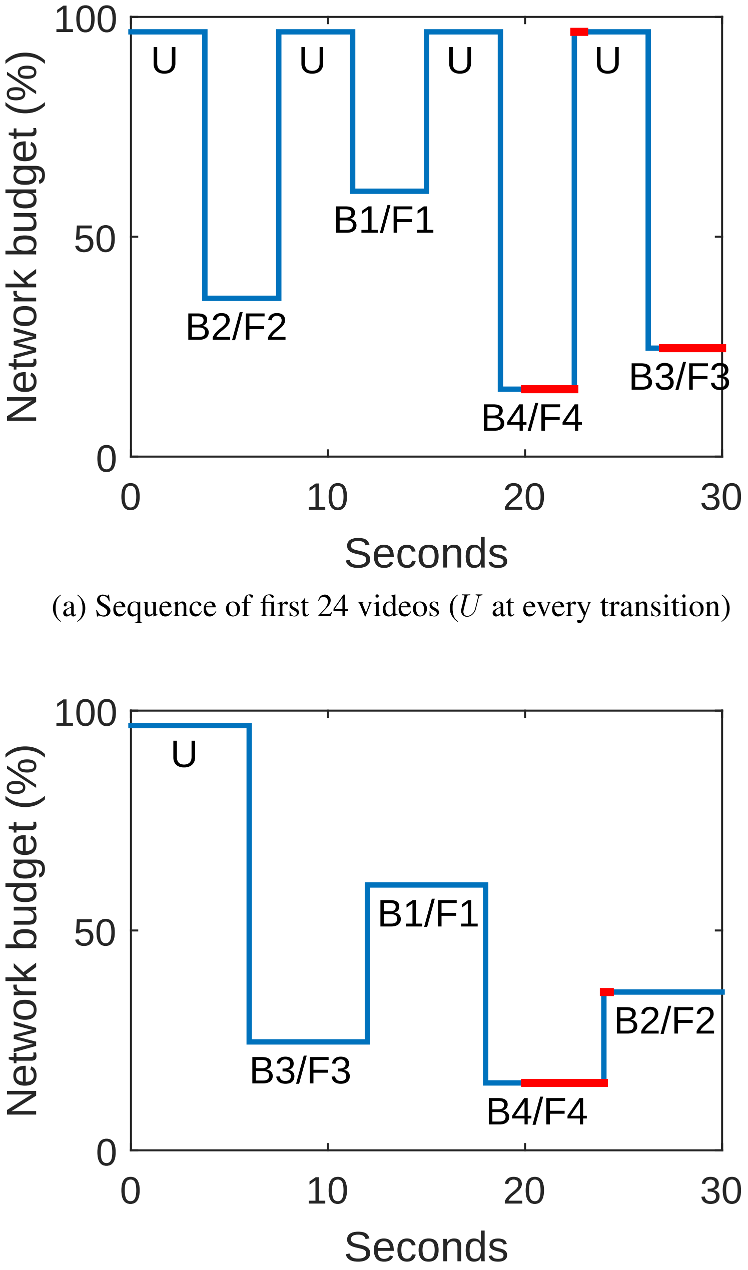
4.3.2 User Study Results

The results are summarized in Figure 4.8, which show the *fraction of time the subjects perceived poor video quality* for baseline and foveated compression conditions. The error bars indicate standard error. In the first two plots (Figures 4.8(a) and (b)), the total duration of button press times are used to calculate the fraction of time poor video quality was reported. The second two plots (Figures 4.8(c) and (d)) compensate for user reaction time lag (discussed in the previous subsection) as follows. If a button press is initiated in response to a given level of resolution, that entire clip segment shown at that resolution is counted as a poor viewing experience. Also, if the button is pressed for the entire duration of a segment (i.e., the button is pressed during an earlier segment and is released during a later segment), such segments are also counted as a poor viewing experience. No other time intervals are counted as indicating a poor viewing experience (e.g., when a button press ends but does not start in a segment) in this lag-corrected estimate.[2] Note that after this compensation, indications of a poor perception of U almost disappear, suggesting that the compensation method worked well.

[2] Thus, if the subject presses the button sometime after the start of a ‘bad’ interval and releases it after the start of a ‘good’ interval, the entire bad interval is counted as a poor viewing experience and no part of the good interval is counted as poor.

Compression levels with similar network budgets (see Table 4.1) are grouped together in the plots for ease of reading. For example, $B4$ and $F4$ are grouped together. Recall that in each of these cases they have similar (but not identical) network budgets, and the budget for the foveated version (e.g., $F4$) is always slighty lower (Table 4.1). Still, the foveated versions almost always result in a significantly better user experience. Looking at Figure 4.8(d), which portrays the most realistic evaluation with compensation applied, note that the fraction of poor quality perceptions roughly halves with foveated compression.

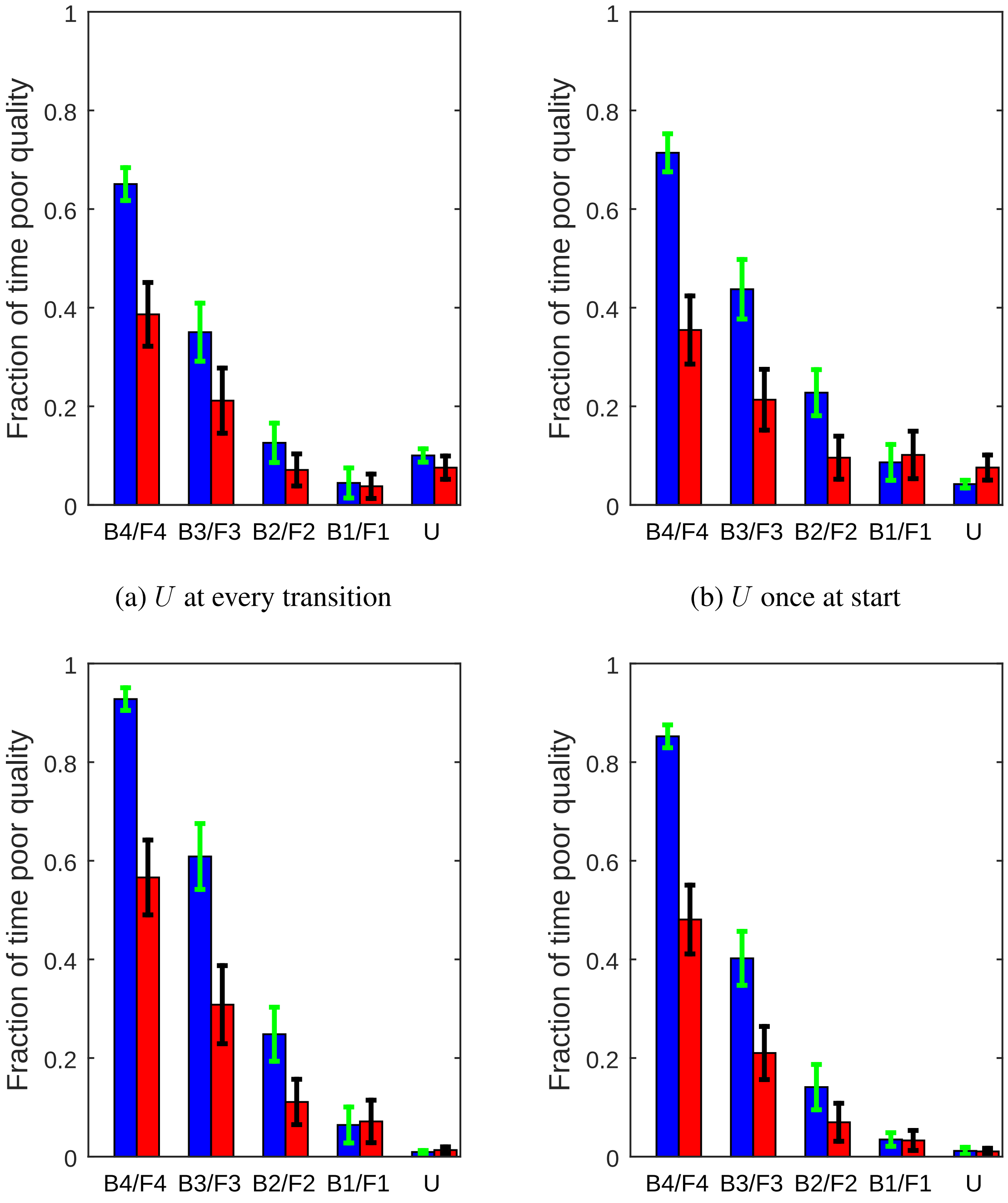
In Figure 4.9 the same data are plotted differently after normalizing for the quality of perception. The plots show *satisfaction level*, defined as '1 $-$ fraction of time the subject perceives poor video quality,' as a function of network budget. Note that for the same satisfaction level the network budget is roughly halved for the foveated version. As expected, the difference between the two gets smaller with increasing budget, indicating that constrained networks will benefit more from foveated video streaming. Qualitatively, there is no significant difference between the two cases 'U at every transition' or 'U once at start.'



(a) Sequence of first 24 videos (U at every transition)

(b) Sequence of last 24 videos (U once at start)

Figure 4.7: Sequencing of video resolutions for the user study. The network budget is from Table 4.1 and is shown only as a guide. The red segments correspond to a button press indicating poor quality perception from an example trace.



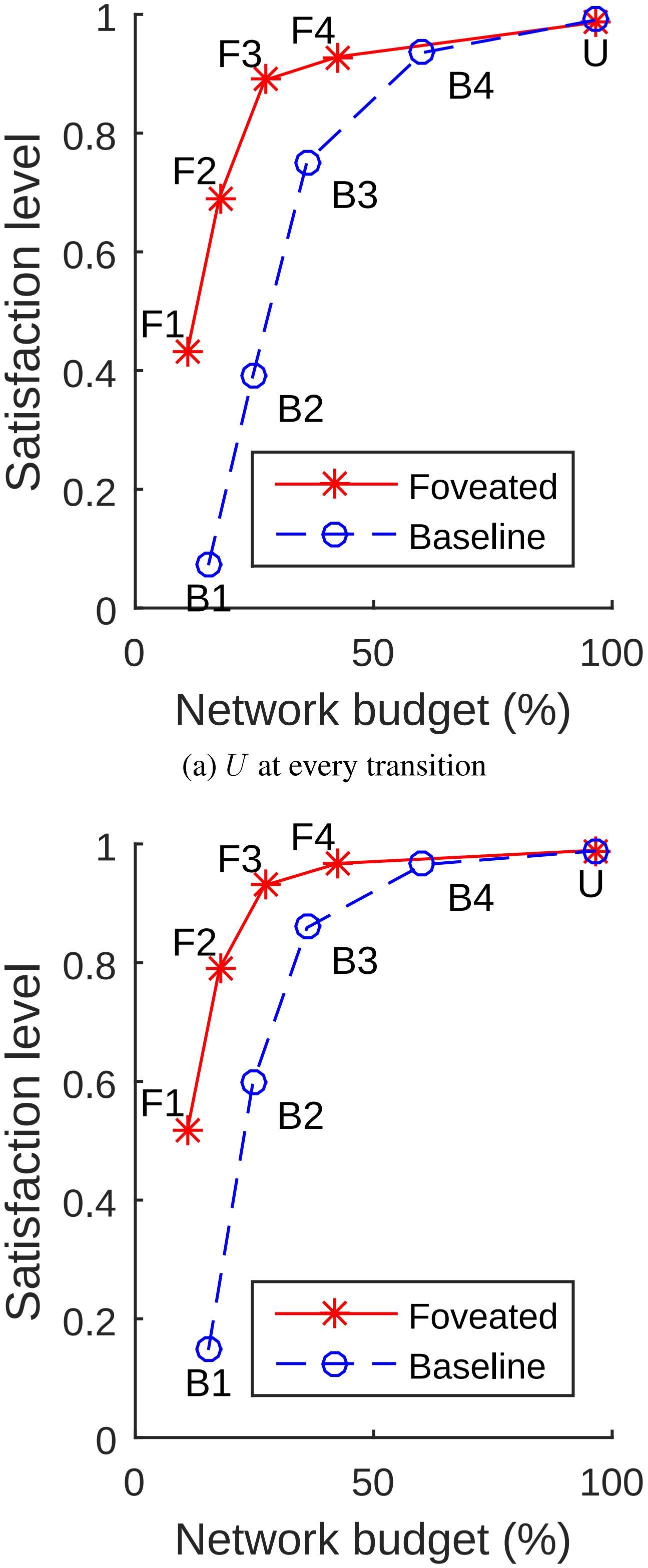
(a) U at every transition

(b) U once at start

(c) U at every transition
(compensated)

d) U once at start
(compensated)

Figure 4.8: User study results: perceptual video quality vs. various compression levels in the baseline and foveated players. Error bars indicate standard errors. Note that foveated streaming results in up to a 50% reduction in the time that poor video quality is perceived.



(a) U at every transition

(b) U once at start

Figure 4.9: Satisfaction level vs. network budget for baseline and foveated compressions.

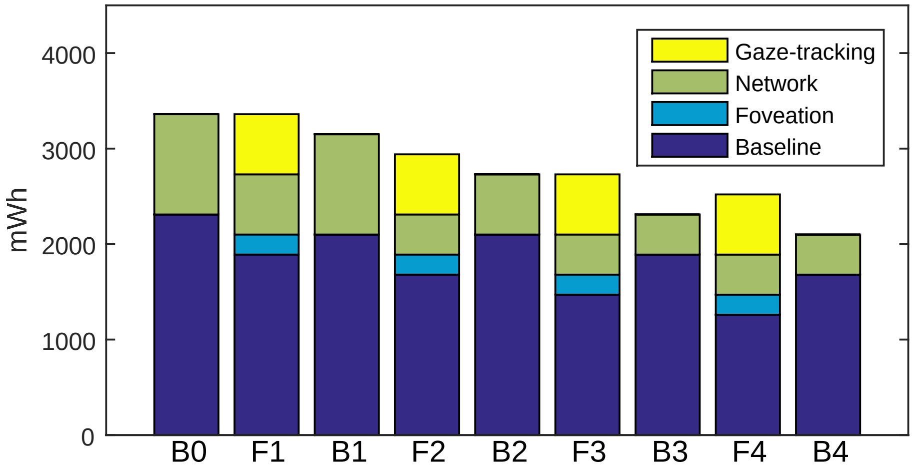


Figure 4.10: Energy expended in running the foveated streaming client at different resolution levels on a tablet over WiFi.

4.4 Energy Measurements for Mobile Use

Our goal is to make foveated video streaming widely available across a range of devices. Since a large and growing amount of video is consumed using untethered mobile devices, we want to ensure that our design is energy efficient when used on a mobile device. The concern is that our system requires that the webcam be continuously on, and that there is some additional computational load imposed by the gaze tracking.

We used a Microsoft Surface Pro3 tablet with 4GB RAM running Windows 8.1 to perform the energy measurements. Software-based measurements of the remaining battery level were obtained via the system-provided API. Four major components of the client player software were systematically assessed by turning on each component incrementally and measuring the total energy expenditure for running a 24 min long video (sequence of all the clips used in the user study). An average of 5 runs are reported. The components studied were:

1. *Baseline:* Basic functionality such as display and computations for the streaming service, excluding the contribution of the network components; the network is disabled and video is stored on local flash.

2. *Foveation:* Additional video decoding computation needed for foveated compression, such as stitching the sectioned multi-resolution images.

3. *Network:* All network components while using 802.11n to connect to an external server.

4. *Gaze tracking:* Webcam and gaze tracking software.

The results are shown in Figure 4.10. Note that the baseline energy costs are slightly higher for baseline (Bi) playing, as a high resolution video must be decoded for the entire screen. The network energy costs are roughly similar to the network budget. The energy cost for gaze tracking is non-negligible – about 20-25$\%$ of the total. We believe that the use of lower power cameras, such as those being proposed for continuous vision applications on mobile devices [93], or low power IR cameras, such as the ones used in Amazon Fire phone [25], would reduce this component of the energy cost. Computational optimizations internal to the gaze tracker are also possible. Similar optimization is also possible in the foveation component, but this was not attempted in the current implementation. Nevertheless, even with the current, unoptimized system, the energy results are encouraging: compression levels producing similar user satisfaction levels (e.g., $B2$ and $F1$, $B3$ and $F2$, $B4$ and $F3$, etc.) incur very similar energy costs. See the user study in Section 4.3 for the actual satisfaction levels.

4.5 Conclusion

This work introduces a conceptual framework for a foveated video streaming service for Internet video servers. The goal is to exploit commodity webcams commonly available in many devices to develop a scalable system that takes eye gaze information from the user and uses it to transfer different parts of the video frame from the server at varying resolutions. The hope is that such a service will alleviate the bandwidth crunch in today's Internet – much of it arising from streaming videos. We see the contribution of this chapter being the description of this conceptual framework, with its focus on a commodity-based scalable solution, and not the delivery of a system designed and optimized for specific hardware settings. For much of our study the basic tools that we used were off-the-shelf, such as the webcam-based eye gaze tracking.

This chapter also develops a robust methodology for conducting user studies aimed at comparing video quality when network budgets vary and video resolutions fluctuate. Using this methodology, we conducted a comprehensive user study that showed a factor of 2 bandwidth reduction while keeping the same user satisfaction. This is promising.

Although outside the scope of the current study, future work will move closer to a completely designed end-to-end system enabling the study of varying network conditions and prefetching and buffering techniques in the context of our foveated video streaming system. We will also attempt to combine automated saliency detection with eye gaze [33, 101, 130] so as to further improve system performance.

Chapter 5

A Case for Using Gaze Feedback for the Web page loading

Web performance has long been crucial to the Internet ecosystem since a significant fraction of Internet content is consumed as Web pages. As a result, there has been a tremendous effort towards optimizing Web performance [38, 107, 171]. In fact, studies show that even a modest improvement in Web performance can have significant impact in terms of revenue and customer base [29, 30, 85].

The goal of our work is to improve page load performance, also called the Page Load Time (PLT), from the perspective of the user. PLT is typically measured using objective metrics such as *OnLoad* [8], and more recently *Speed Index* [74]. However, there is a growing concern that these objective metrics do not adequately capture the user experience [3, 9, 122, 147].

As a first step, we define a *perceptual* variation of page load time that we call *user-perceived PLT*, or *uPLT*. We conduct a systematic user study to show what was anecdotally known, i.e., uPLT does not correlates well with the OnLoad or Speed Index metrics. How-

Acknowledge *Conor Kelton* who brought the idea of "collective fixation" and wrote the code for "HTTP/2 server push." His significant work on this chapter turn my idea into publication.

ever, almost all current Web optimization techniques attempt to optimize for the OnLoad metric [16, 107, 143, 153, 173] rendering their impact on user experience uncertain. The problem is that improving uPLT is non-trivial since it requires information about user's attention and interest.

Our key intuition is to leverage recent advances in eye gaze tracking. It is well known that user *eye gaze* – in terms of fixation, dwell time, and search patterns – correlate well with user attention [31, 161]. In the human visual system only a tiny portion (about 2°) at the center of the visual field is perceived with the highest visual acuity and the acuity sharply falls off as we go away from the center [163]. Thus the eye must move when a user is viewing different parts of the screen. This makes eye gaze a good proxy for user's attention. Further, the commoditization of gaze trackers allow accurate tracking using low cost trackers [87, 98, 117, 150], without the need for custom imaging hardware.

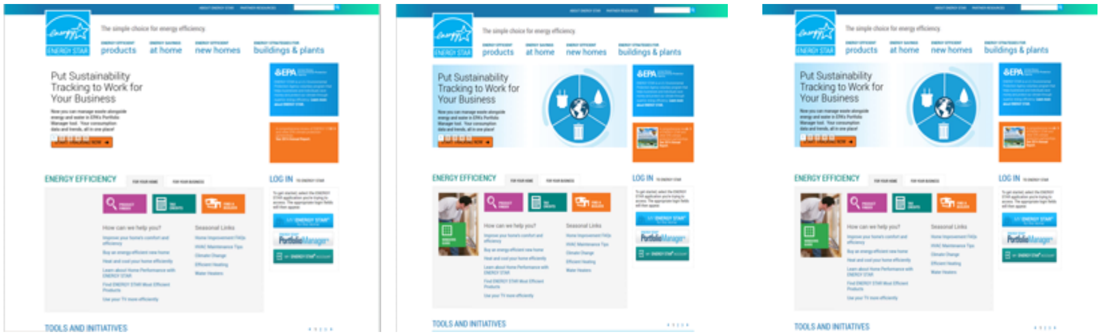
We design *WebGaze*, a system that uses gaze tracking to significantly improve uPLT. WebGaze prioritizes objects on the Web page that are more visually interesting to the user as indicated by the user's gaze. In effect, WebGaze encodes the intuition that loading "important" objects sooner improves user experience. The design of WebGaze has two main challenges: *(i)* Scalability: personalizing the page load for each user according to their gaze does not scale to a large number of users, and *(ii)* Deployability: performing on-the-fly optimizations based on eye gaze is infeasible since page loads are short-lived and the gaze tracker hardware may not be available with every user.

WebGaze addresses these challenges by first distilling gaze similarities across users. Our gaze user study shows that most users are drawn to similar objects on a page. We divide the page into visually distinctive areas that we call regions and define the *collective fixation* of a region as the fraction of users who fixate their gaze on the region. Our study with 50 users across 45 Web pages shows that a small fraction of the Web page has extremely high collective fixation. For example, of the Web pages in our study, at least 20% of the regions were viewed by 90% of the users. Whereas, at least a quarter of the regions of the page are looked at by less than 30% of the users.

WebGaze then uses the HTTP/2 Server Push [65, 102] mechanism to prioritize loading objects on the page that exhibit high degree of collective fixation. In fact, WebGaze provides a content-aware means of using the HTTP/2 Server Push mechanism. WebGaze does not require gaze tracking on-the-fly or require that every user participates in gaze tracking, as long as enough users participate to estimate the collective fixation. WebGaze's algorithm not only pushes the objects of interest, but also all dependent objects as obtained using the WProf tool [171].

The goal of WebGaze is to improve uPLT, a subjective metric that depends on real users. Therefore, to evaluate WebGaze, we conduct an extensive crowd-sourced user study to compare the performance of WebGaze's optimization with three alternatives: *Default*, *Push-All*, and *Klotski* [38]. Default refers to no prioritization. The Push-All strategy indiscriminately prioritizes all objects. Klotski is the state-of-the-art system whose goal is to improve Web user experience: Klotski works by prioritizing objects that can be delivered within the user's tolerance limit (5 seconds). We conduct user studies across 100 users each to compare WebGaze with each alternative.

The results show that WebGaze improves the median uPLT over the three alternatives for 73% of the 45 Web pages. In some cases, the improvement of WebGaze over the default is 64%. While the gains over the default case come from prioritizing objects in general, the gains over Push-All and Klotski come from prioritizing the right set of objects. *All user study data and videos of Web page loads under WebGaze and each alternative strategy can be found at* `http://gaze.cs.stonybrook.edu`.



(a) Speed Index = 3.7 seconds (b) Median uPLT = 8.2 seconds (c) OnLoad = 12 seconds

Figure 5.1: Snapshots of the page load of `energystar.gov` shown at the Speed Index, the median uPLT across 100 users, and OnLoad values.

5.1 Page Load Metrics

To study the perceptual performance of Web page loads, we define a perceptual variation of the PLT metric, that we call uPLT or user-perceived Page Load Time. uPLT is the time between the page request until the time the user ‘perceives’ that the page is loaded. In this section, we provide a background on traditional PLT metrics and qualitatively describe why they are different from uPLT. In the next section, we use a well posed user study to quantitatively compare traditional PLT metrics and uPLT.

OnLoad: PLT is typically estimated as the time between when the page is requested and when the OnLoad event is fired by the browser. The `OnLoad` event is fired when *all* objects on the page are loaded [14]. There is a growing understanding that measuring PLT using the OnLoad event is insufficient to capture user experience [3, 9, 147]. One reason is that users are often only interested in *Above-the-Fold (AFT)* content, but the OnLoad event is fired only when the entire page is loaded, even when parts of the page are not visible to the user. This leads to the OnLoad measure over-estimating the user-perceived latency. But in some cases, OnLoad can underestimate uPLT. For example, several Web pages load additional objects *after* the OnLoad event is fired. If the additional loads are critical to user experience, the PLT estimated based on the OnLoad event will under-estimate uPLT.

Variants of the OnLoad metric such as `DOMContentLoaded` [14, 171], are similarly disjoint from user experience.

Speed Index: Recently, `Speed Index` [74] was proposed as an alternate PLT measure to better capture user experience. Speed Index is defined as the average time for all AFT content to appear on the screen. It is estimated by first calculating the visual completeness of a page, defined as the pixel distance between the current frame and the "last" frame of the Web page. The last frame is when the Web page content no longer changes. Speed Index is the weighted average of visual completeness over time. The Speed Index value is lower (and better) if the browser shows more visual content earlier.

The problem with the visual completeness measure (and therefore Speed Index) is that it does not take into account the relative importance of the content. This leads to over- or under-estimation of user-perceived latency. If during the page load, a majority of visual components are loaded quickly, Speed Index estimates the page load time to be a small value. However, if the component critical to the user has not yet been loaded, the user will not perceive the page to be loaded. In other cases, Speed Index overestimates. For example, if a large portion of the page has visual content that is not interesting to the user, Speed Index will take into account the time for loading all the visual content, even though the user may perceive the page to be loaded much earlier.

Motivating Example: Figure 5.1 shows the `energystar.gov` page, and the three snapshots taken when the page was considered to be loaded according to the Speed Index, uPLT, and OnLoad metrics. In the case of uPLT, we choose the median uPLT value across 100 users who gave feedback on their perceived page load time (§5.4).

Speed Index considers the page to be loaded much earlier, at 3.2 seconds, even though the banner image is not loaded. For the users, the page is not perceived to be completely loaded unless the banner is loaded, leading to Speed Index under-estimating uPLT. On the other hand, the OnLoad metric estimates the page to be loaded 4 seconds *after* the user perceives the page to be loaded, even though their snapshots are the same visually. This is because the OnLoad event fires only when the entire page, including the non-visible parts, are loaded. This illustrative example shows one case when the traditional PLT metrics do not accurately capture user experience.

5.2 Gaze Tracking

Existing Web page optimizations focus on improving traditional PLT metrics. However, our analysis shows that traditional PLT metrics do not correlate well with uPLT, rendering the effect of existing optimizations on user experience unclear. Instead, we propose to leverage users' eye gaze to explicitly improve uPLT.

5.2.1 Inferring User Interest Using Gaze

Gaze tracking has been widely used in many disciplines such as cognitive science and computer vision to understand visual attention [41, 111]. Recently, advances in computer vision and machine learning have also enabled low cost gaze tracking [87, 98, 117, 150]. The low cost trackers do not require custom hardware and take into account facial features, user movements, a user's distance from the screen, and other user differences.

WebGaze leverages the low cost gaze trackers to capture visual attention of users. As a first step, we conduct a user study to collect eye gaze from a large number of users across Web pages. Using gaze data collected using both a low cost gaze tracker and an expensive custom gaze tracker, we show that the tracking accuracy of the low cost tracker is sufficient for our task.

Next, we analyze the collected gaze data to infer user patterns when viewing the same Web page. Specifically, we identify the *collective fixation* of a region on the Web page, which presents a measure to represent how much a broad group of users attention is fixated on the specific region. WebGaze uses collective fixation as a proxy for user interest, and leverages it to improve uPLT.

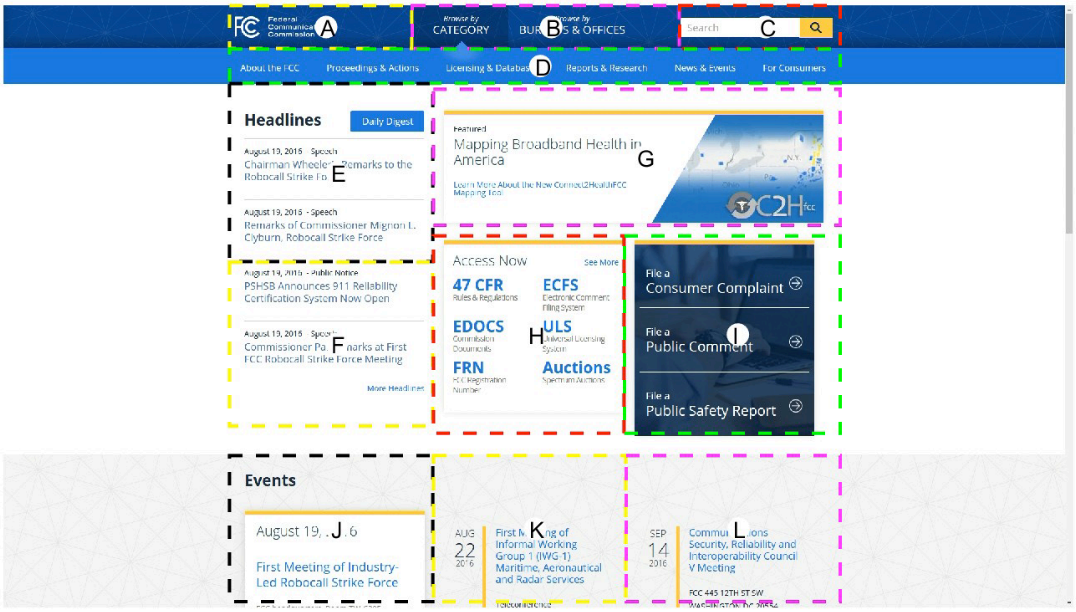


Figure 5.2: Segmentation of the Web page of fcc.gov into visual regions. The visual regions are named ”A”, ”B”, ”C”, etc.

5.2.2 Gaze User Study Set Up

The gaze user study set up is similar to the lab user study described in §5.4.1. Recall that in our lab user study, we collect uPLT feedback from 50 users as they browse 45 Web pages. In addition to obtaining the uPLT feedback, we also also capture the user’s eye gaze.

The gaze tracking is done using an off-the-shelf webcam-based software gaze tracker called GazePointer [60]. GazePointer tracks gaze at 30 frames/sec and does not require significant computational load because it uses simple linear regression and filtering techniques [98, 162] unlike gaze trackers that require more complicated machine learning [56]. We use a 1920 x 1080 resolution 24 inch monitor with a webcam mounted on top. The screen is placed at a reading distance (≈ 50cm) from the participant. We perform a random point test, where we ask the users to look at 100 pre-determined points on the screen. We find the error of tracker to be less than 5° at the 95th percentile.

The user study requires gaze calibration for each user; we perform this calibration multiple times during the study to account for users shifting positions, shifting directions, and

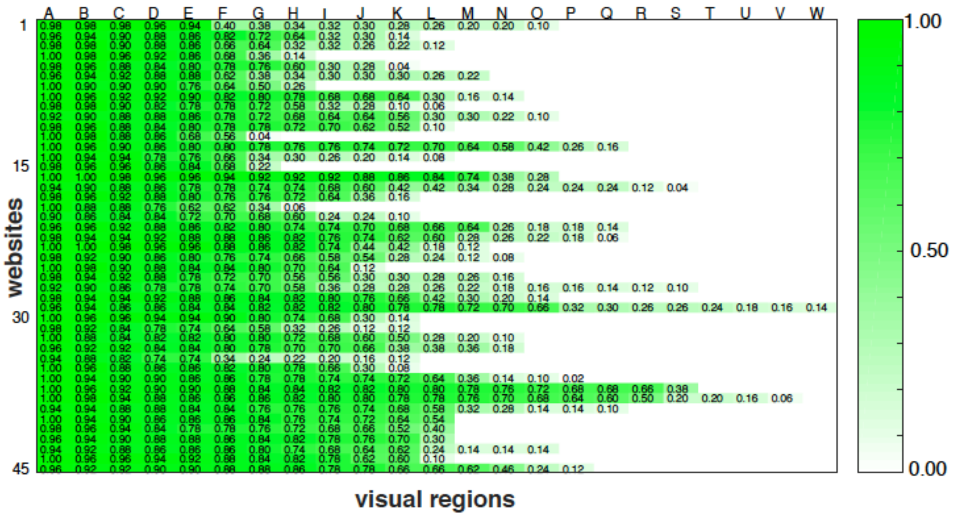


Figure 5.3: A heatmap of the collective fixation of Web page visual regions. Rows correspond to Web pages and the columns correspond to visual regions. For example, for Web site 1, visual region "A" has a collective fixation of 0.98 which means 98% of the users fixated on region "A" during gaze tracking.

other changes. We can potentially replace this calibration requirement using recent advances in gaze tracking that utilize mouse clicks for calibration [117]. These recent calibration techniques are based on the assumption that the user's gaze will follow their mouse clicks which can then be used as ground truth for calibration.

We augment the gaze study with an auxiliary study using a custom gaze tracker with 23 users. The study set up is similar to above, except we use a state-of-the-art *Eye Tracking Glasses 2 Wireless* gaze tracker manufactured by SMI [144]. The gaze tracker uses a custom eyeglass, tracks gaze at 120 frames/sec, and has a very high accuracy (≈ $0.5°$ is typical).

5.2.3 Gaze Tracking Methodology

When a human views a visual medium, his/her eyes exhibit quick jerky movements known as *saccades* interspersed with relatively long (≈ .5 second) stops known as *fixations* that define the his/her interest [79].

Web pages are designed for visual interaction, and thus contain many visually distinct elements, or *visual regions* [22], such as headers, footers, and main content sections, that help guide a user when viewing the page. Rather than track each fixation point, we segment a Web pages into its set of visual regions and track only the regions associated with the user's fixation points [54]. Figure 5.2 shows an example segmentation of `fcc.gov` into its visual regions. It is from this representation that we estimate the collective fixation of a visual region as the fraction of all users' gaze tracks that contain a fixation on the visual region. As part of future work, we will explore other signals of a user's gaze, including fixation duration and fixation order.

5.2.4 Collective Fixation Results

Figure 5.3 shows the collective fixation across each visual region of each Web page. The rows correspond to the Web page and the columns correspond to the visual regions in the Web page labeled 'A', 'B', etc (see example in Figure 5.2). Note that different Web pages may have different visual regions, since region creation depends on the overall page structure.

Figure 5.3 shows that for the first Web page, 5 regions have a collective fixation of over 0.9. In other words, 90% of the users fixated on these 5 regions in gaze tracking. But the remaining 75% of the regions have a collective fixation of less than 0.5.

In general, we find that across the Web pages, at least 20% of the regions have a collective fixation of 0.9 or more. We also find that on an average, 25% of the regions have a collective fixation of less than 0.3; i.e., 25% of the regions are viewed by less than 30% of the users.

Figure 5.4 shows the data in Figure 5.3 from a different perspective. Figure 5.4 is the median of the CCDF's of collective fixations for each site. Each point in the graph shows the percentage of regions with at least a certain collective fixation value. For example, the graph shows that 55% of the regions have a collective fixation of at least 0.7 in the median case. Our key takeaways are: *(i)* several regions have high collective fixation, and *(ii)* there is a significant number of regions that are relatively unimportant to the users. These points suggest that a subset of regions are relevant to the users' interests, an observation that can be exploited to improve uPLT (§5.5).

Figure 5.5 shows a visualization of the gaze tracks on `fcc.gov` across all users. The combined gaze fixations show a high degree of gaze overlap. The thicker lines show the regions on the Web page where the users' gaze exhibit a high degree of collective fixation. The thinner lines show the regions that only a few users look at.

5.2.5 Auxiliary Studies

In our auxiliary studies, we track gaze using a state-of-the-art gaze tracker as users viewed Web page loads under slow 3G and fast WiFi-like network conditions (network set up discussed in §5.4.1). The collective fixation results using the custom gaze tracker are quantitatively similar to the results when tracking gaze using the low cost tracker. For instance, 30% of the regions have a collective fixation of more than 0.8, and 30% of regions have a collective fixation of less than 0.1 under slow network conditions. The results under fast network conditions are similar.

We also conducted an additional set of experiments to study the effects personalized Web pages have on the user's gaze. Web pages such as `Facebook` customize their page to a given user, even though the overall structure of the page remains the same. This customization may result in different users focusing on different parts of the page. We choose five personalized Web pages where the users login to the site: Facebook, Amazon, YouTube, NYTimes, CNN. We conduct a user study with 20 users who gave us permission to track their gaze while they were browsing the logged-in Web pages. Despite customized content, we see similar patterns in collective fixation. All sites see a collective fixation of 0.8 or above for 30% of regions while still having at least 30% of regions with collective fixations below 0.1. In addition, on average these sites have 20% of their regions with

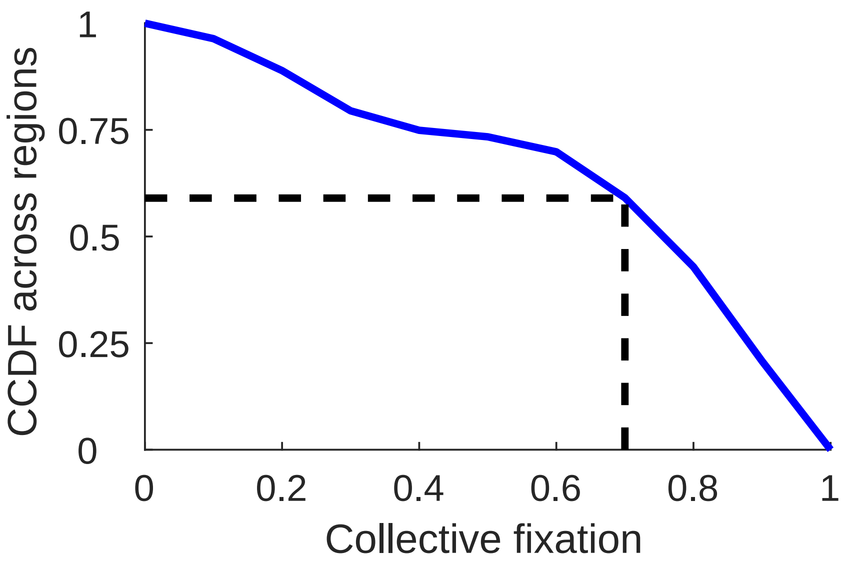


Figure 5.4: The median of the CCDF's of collective fixations across regions. Each point in the graph shows the fraction of regions with at least a certain collective fixation value in the median case.

a collective fixation above 0.9 and 33% below 0.3. Thus, even for pages where specific contents of the page vary across users, we observe there exist regions of high and low collective fixation.

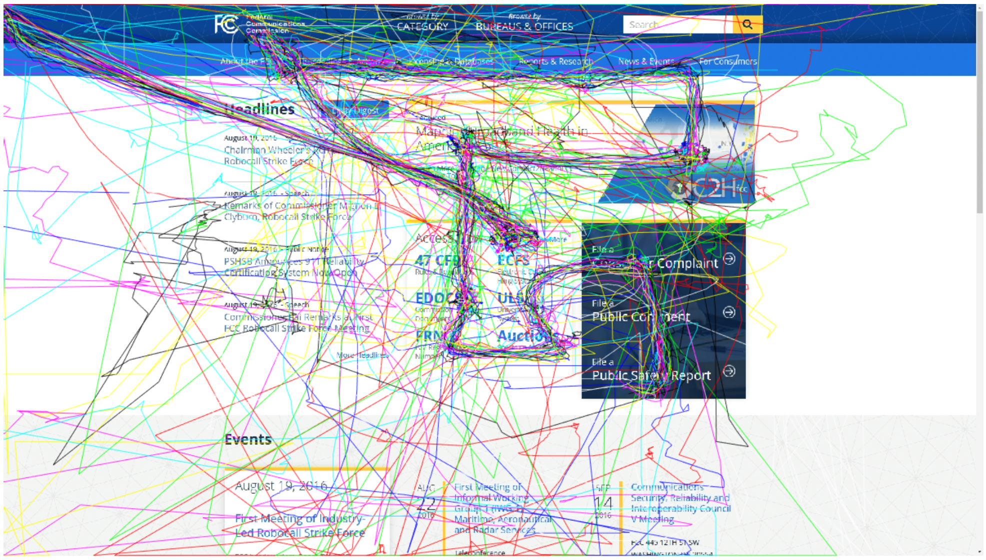


Figure 5.5: A visualization of the gaze of all users when viewing `fcc.gov`. Certain regions on the page have more gaze fixations than others (as evidenced by the thicker lines).

5.3 WebGaze Design and Architecture

The previous section establishes that for each Web page, there exists several regions with high collective fixation. WebGaze is based on the intuition that prioritizing the loading of these regions can improve uPLT. This intuition is derived from existing systems and metrics, including Klotski [38] and the Speed Index. The goal of the Klotski system is to maximize the number of objected rendered within 3–5 seconds, with the intuition that loading more objects earlier improves user experience. Similarly, Speed Index uses the visual loading progress of a page as a proxy for the user's perception. The Speed Index value improves when more objects are rendered earlier on the screen. Similar to these works, our goal is also to render more objects earlier, but WebGaze chooses objects that are more important to the users as determined by their gaze feedback.

5.3.1 Architecture

Figure 5.6 shows the architecture of WebGaze. WebGaze is designed: *(i)* to have no expectations that all users will provide gaze feedback, *(ii)* to not require that pages be optimized

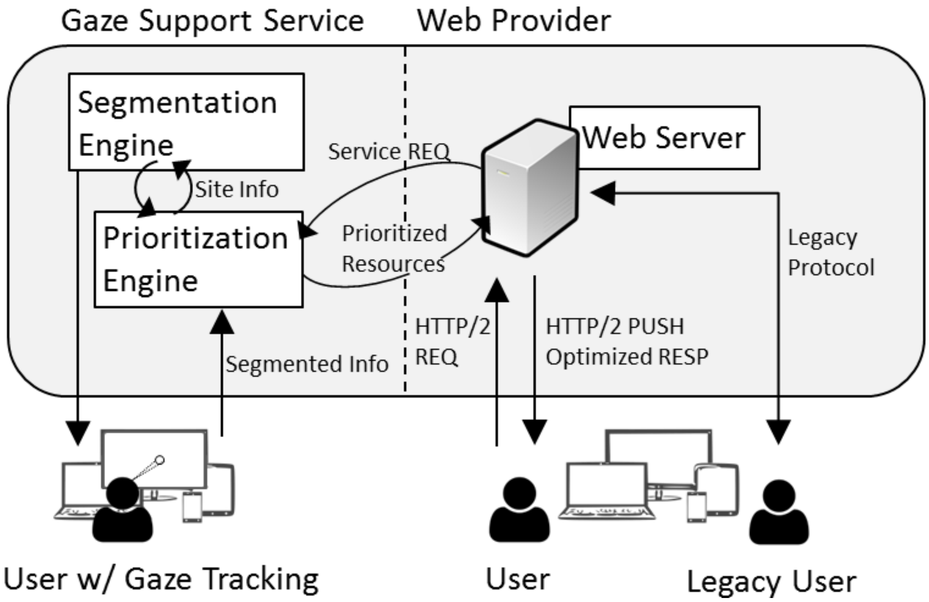


Figure 5.6: WebGaze architecture.

based on real time gaze feedback. We note that existing gaze approaches for video optimization do rely on real time gaze feedback for prioritization [132]. However, Web page loads are transient; the short time scales makes it infeasible to optimize the Web page based on real time gaze feedback.

The WebGaze architecture collects gaze feedback from a subset of users as they perform the browsing task. WebGaze collects the gaze feedback at the granularity of visual regions. When sufficient gaze feedback is collected, the WebGaze server estimates the collective fixation across regions. The server periodically collects more gaze information and updates its fixation estimations as the nature of the Web page and users' interests change.

Based on the collective fixation values, WebGaze, (1) identifies the objects in regions of high collective fixation, (2) extracts the dependencies for the identified objects, (3) uses HTTP/2 Server Push to prioritize the identified objects along with the objects that depend on them. Below, we describe these steps in detail.

5.3.2 Identifying Objects to Prioritize

To identify which Web objects to prioritize, we use a simple heuristic: if a region has a collective fixation of over a *prioritization threshold*, then the objects in the region will be prioritized. In our evaluation, we set the prioritization threshold to be 0.7, thus any objects within a visual region that has a collective fixation of 0.7 or higher are prioritized. Recall from Figure 5.4 that this value identifies 55% of regions as candidates for prioritization in the median case.

Moving this threshold in either direction incurs different trade-offs. When the prioritization threshold is increased (moving right in Figure 5.4) we become more conservative in identifying objects to prioritize. However, in being more conservative we may miss prioritizing regions of which are important to some significant minority of users, which can in-turn negatively affect the aggregate uPLT. When the prioritization threshold is decreased, more regions are prioritized. The problem is that prioritizing too many objects leads to data contention for bandwidth that in turn affects uPLT [157] (in §5.5 we show the effect of prioritizing too many objects.) Empirically, we find that the prioritization threshold we chose works well in most cases (§5.5), through it can be further tuned.

Since each region may have multiple objects, WebGaze extracts the objects that are embedded within a region. To do this, we query the Document Object Model (DOM) [7], which is an intermediate representation of the Web page created by the browser. From the DOM we obtain the CSS bounding rectangles for all objects visible in the 1920x1080 viewport. An object is said to be in a given region if its bounding rectangle is within the region. If an object is said to belong to multiple regions, we assign the maximum of the collective fixation of the regions to the object.

5.3.3 Extracting Dependent Objects

Web page objects are often dependent on each other and these dependencies dictate the order in which the objects are processed. Figure 5.7 shows an example dependency graph.

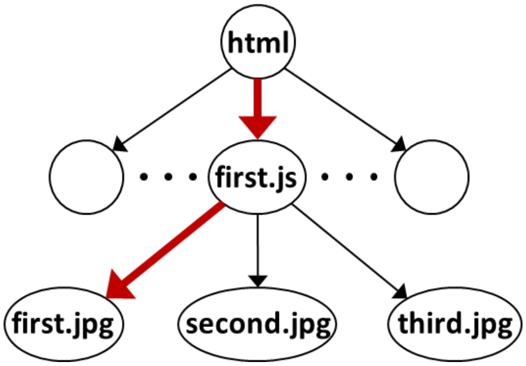


Figure 5.7: A dependency graph for an example page. If the `first.jpg` needs to be prioritized based on the collective fixation measure, then `first.js` also needs to be prioritized since `first.jpg` depends on it.

If `first.jpg` belongs to a region with high collective fixation and is considered for prioritization, then `first.js` also needs to be prioritized, because `first.jpg` depends on `first.js`. If not, then prioritizing `first.jpg` is not likely to be useful since the browser needs to fetch and process `first.js` before processing the image.

Our system identifies dependencies of each object to be prioritized, and considers these dependent objects for prioritization as well. Our current implementation uses WProf [171] to extract dependencies, but other dependency tools [38, 107] can also be used. While the contents of sites are dynamic, the dependency information has shown to be temporally stable [38, 107]. Thus, dependencies can be gathered offline.

5.3.4 Server Push and Object Sizes

WebGaze, like other prioritization strategies [38], uses HTTP/2's Server Push functionality to implement the prioritization. Server Push decouples the traditional browser architecture in which Web objects are fetched in the order in which the browser parses the page. Instead, Server Push allows the server to preemptively push objects to the browser, even when the browser did not explicitly request these objects. Server Push helps *(i)* by avoiding a round trip required to fetch an object, *(ii)* by breaking dependencies between client side parsing and network fetching [107], and *(iii)* by better leveraging HTTP/2's multiplexing [157].

Of course, Server Push is still an experimental technique and is not without problems. Google's studies find that using Server Push can, in some cases, result in a reordering of critical resources that leads to pathologically long delays [157]. To avoid such pathological cases, we check for a simple condition: if the FirstPaint of the page loaded with WebGaze takes longer than the LastVisualChange in the default case, we revert back to the default case without optimization (recall the definitions of FirstPaint and LastVisualChange from §5.4.1). In our evaluation, we found that for 2 out of the 45 pages, WebGaze's use of Server Push resulted in such delays.

Another problem is that Server Push can incur performance penalties when used without restraint. Pushing too many objects splits the bandwidth among the objects, potentially delaying critical content, and in-turn, worsening performance. To address this, Klotski avoids prioritizing large objects or objects with large dependency chains [38]. Although we do not consider object sizes in our current implementation, we plan to do so as part of future work.

Finally, Server Push can use *exclusive* priorities [102] to further specify the order in which the prioritized objects are pushed as to respect page dependencies. However, existing HTTP/2 implementations do not support fully this feature. With a full implementation of HTTP/2's exclusive priorities, WebGaze's mechanism can potentially be tuned even further.

5.4 uPLT User Study

We conduct a user study to systematically compare uPLT with traditional PLT metrics, with the goal of verifying our observations presented in §5.1.

5.4.1 Set Up

Our user study was conducted (1) in the lab, and (2) online using crowd-sourcing. For the lab-based study we recruit subjects from our university. The user subjects belong to the age group of 25 to 40, both male and female. The online study is conducted on the Microworkers [15] platform. We present results from 100 users, 50 from each study. *All user studies presented in this paper were approved by the Institutional Review Board of our institution.*

5.4.2 User Study Set Up and Task

A key challenge of conducting Web page user studies *in-the-wild* is that the Web page load timings experience high variance [172]. The uPLT feedback from two users for a given page may not be comparable under such high variance. To conduct repeatable experiments we capture *videos* of the page load process. The videos are captured via `ffmpeg` at 10 fps with 1920x1080 resolution as the page loads. The users see the video instead of experiencing an actual page load on their computers. This way, each user views exactly the same page load process.

The primary task of the user is to report their perceived page load time when they are browsing the page. We ask the user to view the Web page loading process and give feedback (by pressing a key on the keyboard) when they perceive that the page is loaded. There is an inevitable reaction time between when a user perceives the page to be loaded and when they enter the key. For all measurements, we correct for the user's reaction time using calibration techniques commonly used in user-interaction studies [10]. To ensure high quality data from the user study, we remove abnormally early or late responses. To do so we utilize the `First Paint` and `Last Visual Change` PLT metrics [66]. The First Paint is the time between when the URL begins to load and the first pixel is rendered, and the Last Visual Change is the time when the final pixel changes on the user's screen. Any responses before the First Paint and after the Last Visual Change events are rejected.

Web Pages

In the default case, we choose 45 Web pages from 15 of the 17 categories of Alexa [4], ignoring Adult pages and pages in a language other than English. From each category, we *randomly* choose three Web pages; one from Alexa ranking 1–1000, another from Alexa ranking 10k–20k, and the other from Alexa ranking 30k+. This selection provides wide diversity in the Web pages. The network is standardized to the accepted DSL conditions [177], 50ms RTT, 1.3Mbps downlink and 384Kbps uplink, using the Linux traffic controller '`tc`' [34].

We conduct additional user studies by varying network conditions using the `tc` tool [34] to emulate: i) WiFi-like conditions: a 12 ms RTT link with 20 Mbps download bandwidth and ii) 3G-like conditions: a 150 ms RTT link with a 1.6 Mbps download bandwidth. We conduct these additional user studies across 30 users and 23 Web pages, half from the top 100 and remaining from between 10k–20k Web pages from Alexa's list [4].

Measurement Methodology

We load the Web page using Google Chrome version 52.0.2743.116 for all loads. We do not change the Web load process, and all the objects, including dynamic objects and ads, are loaded without any changes.

When the Web page load finishes, we query Chrome's Navigation Timeline API remotely through its Remote Debugging Protocol [40]. Similar interfaces exist on most other modern browsers [106]. From the API we are able to obtain timing information including

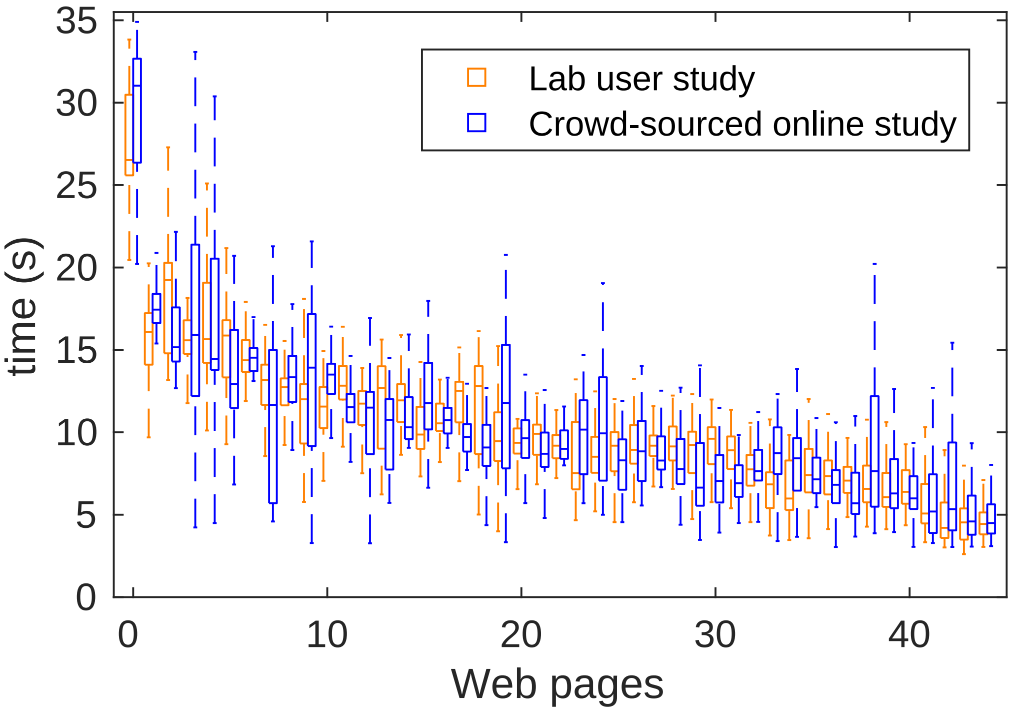


Figure 5.8: Comparing the uPLT box plot between the 50 lab and 50 crowd-sourced users. Although the uPLT values vary across users, the distributions are similar for the two data sets. This data is collected under the desktop environment.

the OnLoad measure. To estimate Speed Index, we first record the videos of the pages loading, recorded at 10 frames-per-second. The videos are fed through the WebPageTest Tool [177] that calculates the Speed Index.

5.4.3 Comparing uPLT with OnLoad and Speed Index

First, we compare the uPLT variations across lab-based and crowd-sourced studies for the same set of Web pages. Figure 5.8 shows the uPLT box plots for each Web page across the two different studies. Visually from the plot, we find that the lab and crowd-sourced users report similar distributions of uPLT. The standard deviation of the median uPLT difference between the lab and the crowd-sourced study for the same Web page is small, about 1.1 seconds. This same measure across Web pages is much larger, at about 4.5 seconds.

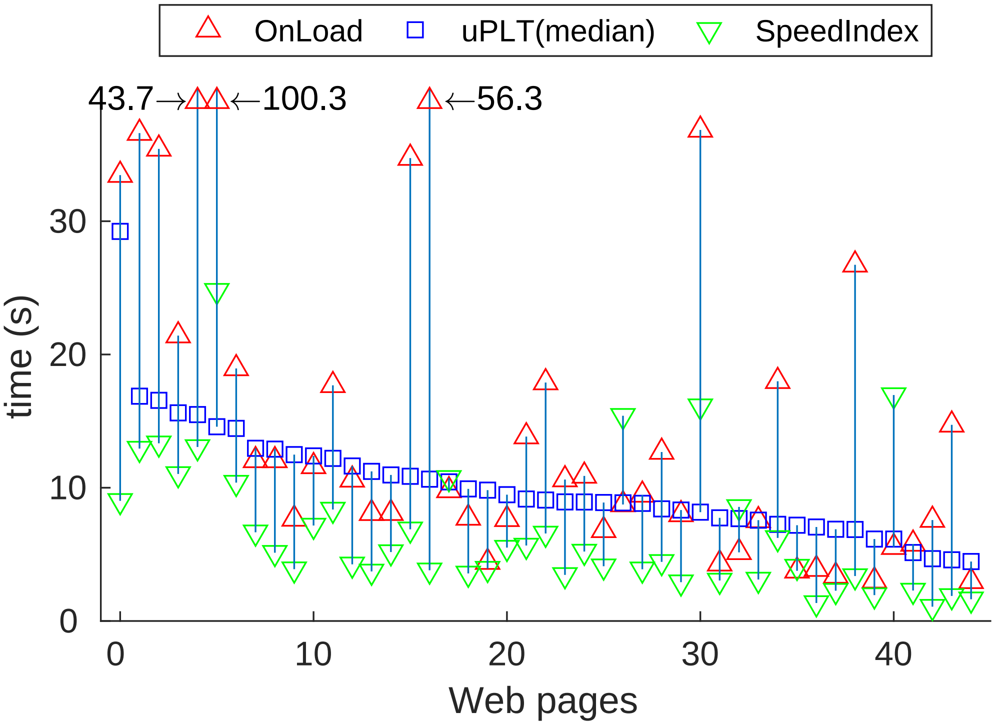


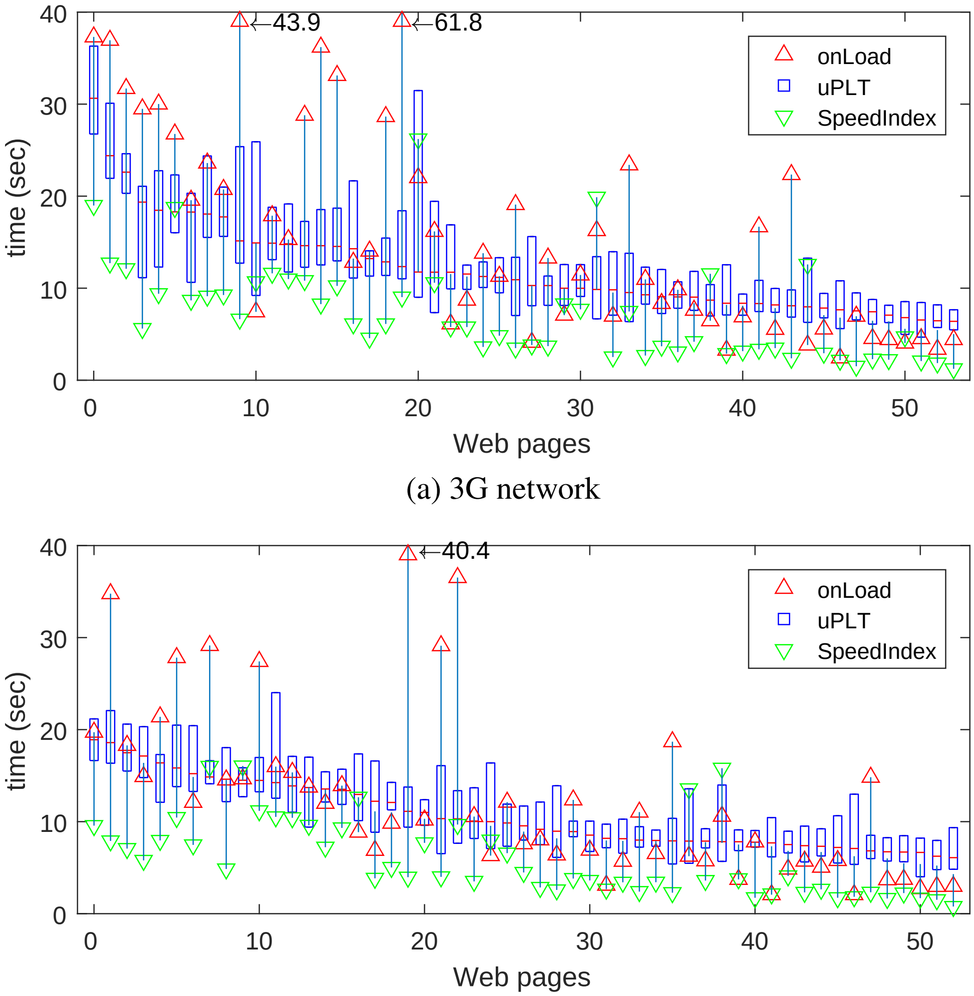
Figure 5.9: Comparing median uPLT with OnLoad and Speed Index across 45 Web pages and 100 users. The median uPLT is lower than OnLoad for 50% of the Webpages, and higher than Speed Index for 87% of Webpages.

This increases our confidence in the results from the crowd-sourced user study; we leverage a similar crowd-sourced study to evaluate WebGaze.

Figure 5.9 shows median uPLT compared to the OnLoad and Speed Index metrics across the 45 pages and 100 users, combining the crowd-sourced online study and the lab study. The Speed Index and OnLoad values are calculated from the same Web page load in which was recorded and shown to the users.

We observe that uPLT is not well correlated with the Onload and Speed Index metrics: the Correlation Coefficient between median uPLT and the OnLoad metric is ≈ 0.46 while the correlation between median uPLT and the Speed Index is ≈ 0.44. We also find the correlation between uPLT and the DomContentLoaded to be ≈ 0.21.

The OnLoad metric is about 6.4 seconds higher than the median uPLT on an average, for close to 50% of the pages. For 50% of Web pages, the OnLoad is lower than the median uPLT by an average of about 2.1 seconds. On the other hand, Speed Index, estimated over visible AFT content, is about 3.5 seconds lower than uPLT for over 87% of the Web pages. In Section 5.1 we discussed the cases in which the OnLoad and Speed Index can over and underestimate the user perceived page loads. From our results we see that while cases of uPLT over and underestimation occur in equal proportion for the OnLoad, the case of uPLT underestimation, as shown in Figure 5.1, occurs more for the Speed Index.



(a) 3G network

(b) 4G network

Figure 5.10: Comparing median uPLT with OnLoad and Speed Index across 54 representative Web pages and 100 users under (a) 3G and (b) 4G network conditions. This data is collected under the mobile phone environment.

5.4.4 uPLT Across Network Environments

We extend our user study to 3G and 4G network environments. We load 54 mobile pages on a Nexus4 smartphone running unmodified Google Chrome version 58.0. We choose 6 Websites from each of the 9 categories as suggested by Alexa [4] that were confirmed to have Mobile variants. The websites vary in terms of popularity and complexity within each category. We emulate two network conditions using built in application level network emulation as provided by Google Chrome [64]. The conditions were set to the presets for 3G (0.75 Mbps upload link, 1.5 Mbps download link, 100ms RTT) and 4G (3.0 Mbps upload link, 4.0 Mbps download link, 20ms RTT) network conditions.

We use the Chrome Remote Debugging APIs [40] to automate page navigation and to estimate the OnLoad metric. OnLoad is the time from when the URL was requested and when all objects are loaded. It is implemented within all modern browsers. For the Speed Index computation, we take a video of the page as it is being loaded, and input the video to a tool provided by Web Page Test [177]. The Speed Index is defined as the average time for all content in the client's visible window, or above-the-fold content, to appear on the screen [74]. Figure 5.10 shows the median uPLT values for 54 sites in (a) 3G and (b) 4G network environment, and the 25th and 75th percentile across 100 users (blue box with a short red line for the median), and the OnLoad and Speed Index values. The vertical blue lines in the figure do only serve to match the OnLoad, uPLT and Speed Index values for the same Web page.

These results are for (a) 3G network and (b) 4G network conditions. Note that there are significant differences between uPLT vs. OnLoad or uPLT vs. Speed Index. The correlations are also poor. The correlation between the OnLoad metric and median uPLT is 0.609 and between Speed Index and median uPLT is 0.638. In other words, the PLT metrics do not trend well with when the users perceive the page to be loaded, which indicates that these values should act as poor predictors in estimating the uPLT. Further, the median uPLT value is lower than OnLoad for 64% of pages, higher than the Speed Index for 90% of pages.

5.4.5 uPLT Across Categories

The 45 Webpages used in the study have diverse characteristics. In Figure 5.11, we study how uPLT differs from traditional PLT metrics for different categories. Each point in the plot is the median uPLT across 100 users.

We divide the Web pages across four (4) categories: *(i)* Light html: landing pages such as `google.com`, *(ii)* CSS-heavy; *(iii)* Javascript-heavy; and *(iv)* Image-heavy. To categorize the page into the latter three categories, we look at the types of objects downloaded for each page and count the number of CSS, Javascript, and images. The categories are based on the type of object that is fetched most when the page is loaded.

Light html and *CSS-heavy* pages are simple and see little difference between the uPLT and the OnLoad and Speed Index metrics. However, for pages with a lot of dynamic Javascript, the median difference between uPLT and OnLoad is 9.3 seconds. Similarly, for image-heavy pages, the difference between uPLT and OnLoad is high. This is largely because, as the number of images and dynamic content increases, the order in which the objects are rendered becomes important. As we show in the next section, users typically only focus on certain regions of the page and find other regions unimportant, making it critical that the important objects are loaded first.

5.4.6 Varying Network Conditions

Finally, to verify the robustness of our results, we analyze the differences between uPLT OnLoad, and Speed Index under varying network conditions.

Under the slower 3G-like network conditions across 30 lab users and 23 Web pages, median uPLT poorly correlates with OnLoad and Speed Index with a correlation coefficient of 0.55 and 0.51 respectively. The median uPLT was greater than Onload 46% of times, with a median difference of 4.7 seconds. The uPLT was less than Speed Index 72% of the time with the median difference of 1.86 seconds. When we evaluate under WiFi-Like conditions we find the correlation between between OnLoad and uPLT is much higher at

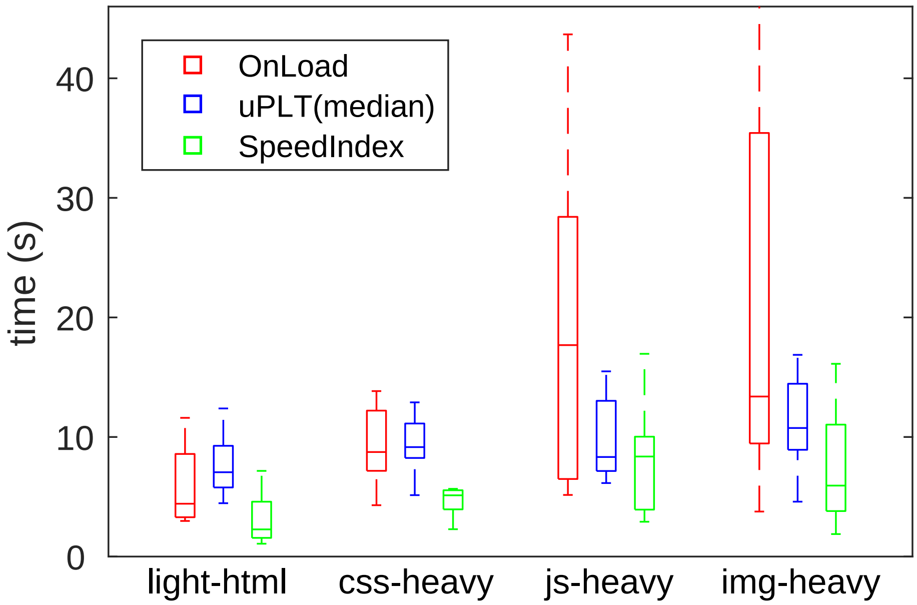


Figure 5.11: OnLoad, SpeedIndex and uPLT for different categories of Web pages

0.74. This result is likely because in faster networks, more pages load instantaneously causing the user perceived latency to not differ much from the OnLoad.

5.5 WebGaze Evaluation

We conduct user studies to evaluate WebGaze and compare its performance with three alternative techniques which are:

- *Default*: The page loads *as-is*, without prioritization
- *Push-All*: Push all the objects on the Web page using Server Push. This strategy helps us study the effect of Server Push at its extreme.
- *Klotski*: Uses Klotski's [38] algorithm to push objects. The algorithm pushes objects and dependencies with the objective of maximizing the amount of ATF content that can delivered within 5 seconds.

As before (§5.4.1), we record videos of the Web page as it is loaded using WebGaze and each alternate technique. The users provide feedback on when they perceive the page to be loaded as they watch the video. We conduct the user study across 100 users to compare WebGaze and each alternative technique. Videos of the Web page loads, under each technique, are available on our project Web page, `http://gaze.cs.stonybrook.edu`.

5.5.1 Methodology

Web pages: We evaluate over the same set of 45 Webpages as our uPLT and gaze studies (§5.4.1). Recall, from the WebGaze architecture, that the Web server corresponding to each Web page prioritizes content based on input from WebGaze. For evaluation purposes, we run our own Web server instead and download the contents of each site locally. We assume that all cross-origin content is available in one server. We note that HTTP/2 best practices suggest that sites should be designed such that as much content as possible is delivered from one server [157]. Nondeterministic dynamic content, such as ads, are still loaded from the remote servers.

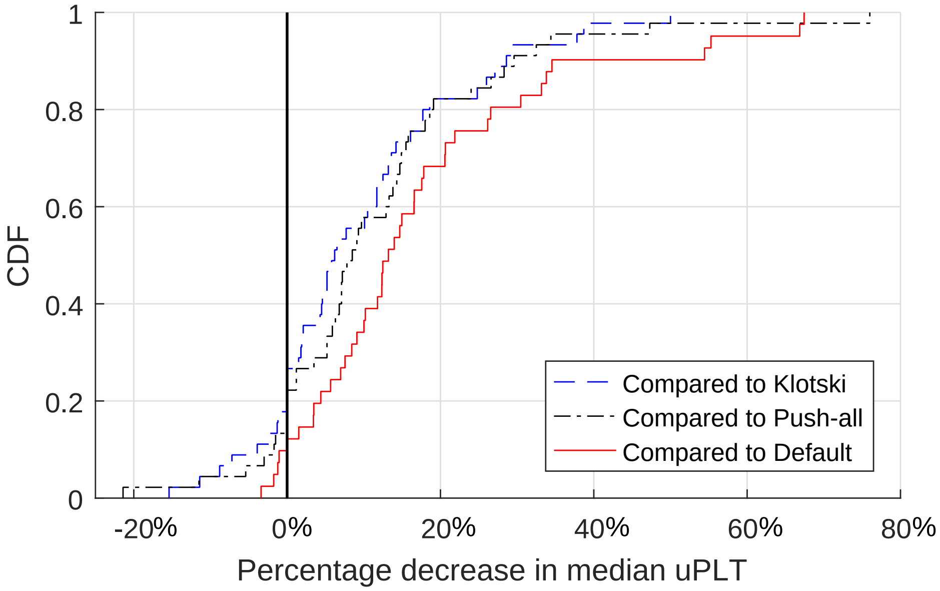


Figure 5.12: CDF of improvement in uPLT over Default, Push-All, and Klotski across the 100 users and 45 Web pages.

Server and client: The Web pages are hosted on an Ubunbu 14.04 server running version 2.4.23 of Apache `httpd` which supports HTTP/2 protocol and Server Push functionality. The Web client is Chrome version 52.0.2743.116, which supports both the HTTP/2 protocol and Server Push, that is also run on an Ubuntu 14.04 machine. Traffic on the client machine is controlled using `tc` [34] to replicate standard DSL conditions (§5.4.1). When using push, we use default HTTP/2 priorities. Due to the standardized conditions of our network, the average variance in OnLoad is less than 1%. So we are able to compare the uPLT values across different page loads.

User study: We conduct pairwise comparisons of uPLT. To this end, we show the users randomized Web page loads that are loaded using WebGaze and using one of the three alternatives. The users provide feedback on uPLT. For each set of 45 comparisons, we recruit 100 users, for a total of 300 users. An alternative design would be to conduct a user study where a single user provides feedback for Web page loads under all four alternatives; but this requires users to give feedback on 180 Web pages which becomes tedious.

Alternative	# WebGaze better	# WebGaze same	# WebGaze worse
Default	37	4	4
Push-All	35	4	6
Klotski	33	4	8

Table 5.1: Number of Web pages for which WebGaze performs better, same, and worse, in terms of uPLT in the median case compared to the alternatives.

5.5.2 Comparing WebGaze with Alternatives

Figure 5.12 shows the CDF of the percentage improvement in uPLT compared to alternatives. On an average, WebGaze improves uPLT 17%, 12% and 9% over Default, Push-All, and Klotski respectively. At the 95th percentile, WebGaze improves uPLT by 64%, 44%, and 39% compared to Default, Push-All, and Klotski respectively. In terms of absolute improvement, when WebGaze improves uPLT the improvement is by an average of 2 seconds over Default and Push-All, and by an average of 1.5 seconds over Klotski. At the 90th percentile, WebGaze improves uPLT by over 4 seconds.

In about 10% of the cases WebGaze does worse than Default and Push All in terms of uPLT and in about 17% of the cases, WebGaze performs worse than Klotski. Of these cases where the competing strategies outperform WebGaze, the average reduction in performance is 13%.

Table 5.1 shows the number of Web pages for which WebGaze performs better, the same, and worse in terms of uPLT for the median case, as compared to the alternatives. Next we analyze the reasons for the observed performance differences.

5.5.3 Case Study: When WebGaze Performs Better

It is not surprising that WebGaze improves uPLT over the default case. Recall our intuition based on prior work [38, 74] that prioritizing regions with high collective fixation can improve uPLT. In addition, pushing objects with adherence to their dependencies has been shown to improve page load performance [107, 171].

Push-All is an extreme strategy, but it lets us study the possible negative effects of pushing too many objects. We find that Push-All delays critical object loads and users see long delays for even the First Paint [66]. In our study, Push-All increases First Paint by an average of 14% compared to WebGaze. Push-All, in-turn, tends to increase uPLT. The problem with pushing too many objects is that each object only gets a fraction of the available bandwidth, in spite of techniques such as HTTP/2 priorities [157].

Different from uPLT, for OnLoad, it is more critical that all objects are loaded even if objects critical to the user are delayed. We see this tendency in our results: the Push-All strategy in fact improves OnLoad for 11 of the 45 pages, whilst hurting uPLT. This example shows that optimizations can help OnLoad, but hurt uPLT.

The uPLT improvement compared to Klotski comes from content-aware prioritization. In the case of Klotski, ATF objects are pushed based on whether they will be delivered within 5 seconds. This may not correlate with the objects that the user is interested in. For example, the Webpage `www.nysparks.com`, Klotski prioritizes the logo.jpg image which is in a region of low collective fixation. This essentially delays other more critical resources that are observed by a large number of users.

5.5.4 Case Study: When WebGaze Performs Worse

WebGaze performs worse than Klotski in 17% of the cases with a median performance difference of 5.5% and a maximum difference of 15.4%. In each of these cases, we find that Klotski sends less data compared to WebGaze and is more conservative. Figure 5.13 shows the relative size of objects pushed by WebGaze and Klotski across the Web pages. This

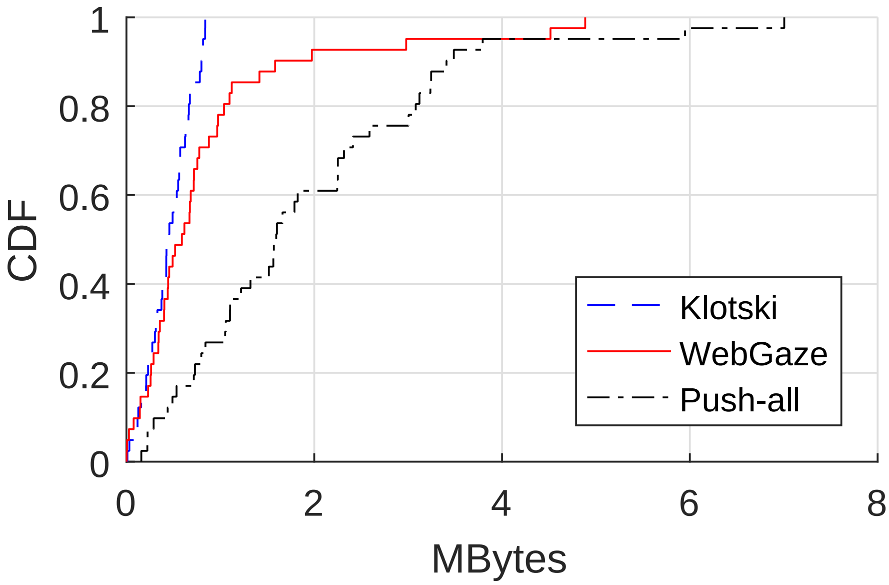


Figure 5.13: The total size of pushed objects under WebGaze, Klotski, and Push-All.

suggests that we need to perform more analysis on determining the right amount of data that can be pushed without affecting performance. Similarly, when compared to Default, WebGaze performs worse for 4 of the 45 Webpages. In each of these cases, WebGaze pushed too much data causing bandwidth contention.

In all cases when WebGaze performs worse compared to Push-All, we find that the Web pages were smaller, less than 1.2 MB. We speculate that pushing all objects for pages of small sizes does not force as much contention for bandwidth.

5.5.5 Case Study: When WebGaze Performs the Same

For a fraction of less than 10% of the pages we find that WebGaze performs similar to the other alternatives. For two of the Web pages, the uPLT values are similar across the four alternatives. In other words, Server Push did not change performance. This could be because the default page itself is well optimized. For the other two pages, WebGaze's Server Push resulted in pathologically delays, and therefore the pages were not optimized (§5.3.4).

Although they are not the metrics WebGaze intends to optimize, for completeness we briefly discuss the performance of WebGaze in terms of the OnLoad, Speed Index, and First Paint. In terms of all three metrics, WebGaze and Klotski show comparable performance. In comparison to Default and Push-All, WebGaze shows only 1–3% improvement in the OnLoad. WebGaze improves the Speed Index metric by an average of 18% compared to the Push-All strategy. However, there is no difference in the average Speed Index measure between WebGaze and Default. Lastly, as discussed earlier, WebGaze improves the average First Paint metric by 14% compared to Push-All. However WebGaze does increase the time to First Paint by 19% on average compared to Default, thus improving uPLT despite increasing the First Paint overall. This result loops back to our intuition (§5.3) that loading more objects important to the user sooner is critical to uPLT.

5.6 Related Work

We discuss three related lines of research that are relevant to our work: Web performance, page load measurements, and modeling user attention for Web pages.

5.6.1 Improving Web Performance

Given the importance of Web performance, significant research effort has gone into improving Web page loads. These efforts include designing new network protocols [16, 153], new Web architectures [107, 143, 173], best practices [1], and tools to help developers write better Web pages [63]. However, most of these efforts target the traditional *OnLoad* metric.

More recently, systems such as Klotski [38] are targeting the user quality of experience rather than optimizing traditional PLT metrics. As discussed earlier, Klotski uses HTTP/2's Server Push functionality to push *high utility and visible* objects to the browser.

WebGaze uses a similar prioritization technique, but prioritizes objects based on user interest. Our evaluations show that WebGaze improves uPLT compared to Klotski across 100 users (§5.5).

5.6.2 Page Load Measurements

The research community has reported on a broad spectrum of Web page user studies. On the one end, there are controlled user study experiments [159], where the researchers create specific tasks for the subjects to complete. However, to create repeatable user studies and to control the Web page load times, the authors create *fake* Web pages. On the other end, there are large scale, less controlled studies [37] that measure performance of hundreds of real Web pages. But these studies only measure objective metrics such as the OnLoad metric.

Around the same time as the design and development of WebGaze, researchers have developed a similar testbed called `eyeorg` to crowd-source Web user experience [160].

The eyeorg study also uses a user-perceived PLT metric to measure user experience, and records the Web pages to obtain standardized feedback from the users as to when they feel the page is loaded. Their methodology in obtaining feedback is slightly different from our study in that they allow the users to transition frame by frame before providing their uPLT. The eyeorg study finds high correlation between the OnLoad and uPLT metrics, similar to our findings in the WiFi-like environment. Different from the eyeorg study, we vary the network conditions when loading the page and show that the correlation results depend on the underlying network (§5.4). On slow networks, OnLoad and uPLT are poorly correlated, while in faster networks, OnLoad and uPLT are better correlated; the later corroborating more with the results of eyeorg. Going beyond crowd-sourcing uPLT feedback, our work also shows how uPLT can be improved by leveraging eye gaze.

5.6.3 Web Saliency

The computer vision community has widely studied how eye gaze data can be used as ground truth to build saliency models [62, 168]. Saliency defines the features on the screen that attract more visual attention than others. Saliency models predict the user's fixation on different regions of the screen and can be used to capture user attention without requiring gaze data (beyond building the model). While most of the research in this space focuses on images [137, 185], researchers have also built saliency models for Web pages.

Buscher et al. [36] map the user's visual attention to the DOM elements on the page. Still and Masciocchi [148] build a saliency model and evaluate for the first ten (10) fixations by the user. Shen et al. [139] build a computational model to predict Web saliency using a multi-scale feature map, facial maps, and positional bias. Ersalan *et al.* [54] study the scan path when the user is browsing the Web. Others have looked at saliency models for Web search [138] and text [59, 165].

However, existing Web saliency techniques have relatively poor accuracy [36, 139]. This is because predicting fixations on Web pages is inherently different and more challenging compared to images: Web pages, unlike images, are a mix of text and figures. Web page loading is an iterative process where all objects are not rendered on the screen at the same time, and there is a strong prior when viewing familiar Web pages.

Our work is orthogonal to the research on Web saliency. WebGaze can leverage better Web saliency models to predict user interest. This will considerably reduce the amount of gaze data that needs to be collected, since it will only be used to provide ground truth. We believe that our findings on how gaze data can improve user-perceived page load times can potentially spur research on Web saliency.

5.7 Discussions

There are several technical issues that will need a close look before a gaze feedback-based Web optimization can be widely useful.

Mobile Devices: It is expected that more and more Web content will be consumed from mobiles. Mobile devices bring in two concerns. First, errors in gaze tracking may be exaggerated in mobiles as the screen could be too small, or the performance of gaze tracking on mobile could be too poor. Significant advances are being made on camera-based gaze tracking for mobile smartphone class devices [11]. But, accuracy is also as not critical to our approach as we require the gaze to be tracked at the granularity of large visual regions.

A second concern is that gaze tracking on mobile devices may consume additional amounts of energy [132]. This is due to the energy consumed in the imaging system and on image analysis in the CPU/GPU. While this can be a concern, a number of new developments are pushing for continuous vision applications on mobiles and very low power imaging sensors are emerging (see, e.g., [93]). Also, lower resolution tracking may still provide sufficient accuracy for our application, while reducing energy burden. Therefore, we expect that gaze tracking can be leveraged to improve uPLT in mobile devices.

Exploiting Saliency Models: Saliency models have been discussed in the previous section. A powerful approach could be to decrease reliance on actual gaze tracking, but rely instead on saliency models. In other words, inspecting Web pages via suitable modeling techniques could discover potential regions of user attention that could be a good proxy for gaze tracks. This approach is more scalable and would even apply to pages where gaze tracking data is not available. The challenge is that research on saliency models for Web pages is not yet mature. Our initial results show promise in leveraging gaze for improving uPLT; exploiting Web saliency models can significantly increase the deployability of our approach.

Systems Issues: There are a number of systems issues that need to be addressed to build a useful Web optimization based on gaze feedback. For example, a standardized gaze interface needs to be developed that integrates with the browser. The gaze support service (Figure 5.6) needs to adapt to changing nature of the Web contents and user interests. For example, a major event may suddenly change users' gaze behaviors on a news Web site even when the structure of the page remains the same.

Security and Privacy: There are additional security and privacy related concerns if gaze feedback is collected by Web sites or third party services. For example, it is certainly possible that gaze tracking could provide a significant piece of information needed to uniquely identify the user, even across devices. The use of eye tracking on the end-user's device exposes the user to hacks that could misuse the tracking data. Note that course-grained tracking information is sufficient for our task, but guaranteeing that only course-grain information is collected requires a hardened system design.

Gaze Tracking Methodology: Web page loads are a dynamic process. Therefore, collecting gaze data when the user looks only at the loaded Web page is not representative of the Web viewing experience. Instead, in this work, we collect gaze data *as* the page is being loaded. However, one problem is that, the gaze fixation is influenced by the Web object ordering. For instance, if objects that are important to the user are rendered later, a user may direct her gaze towards unimportant, but rendered objects. Our methodology partially alleviates the problem by capturing gaze only after the First Paint (§5.5) and even after OnLoad. As part of future work, we propose to track user gaze when the Web objects are loaded in different orders. By analyzing gaze under different object orderings, we hope to alleviate the problem of the Web page loading order influencing gaze tracks.

There has been a recent interest in making user experience the central issue in Web optimizations. Currently, user experience is divorced from Web page performance metrics. We systematically study the user-perceived page load time metric, or uPLT, to characterize user experience with respect to traditional metrics. We then make a case for using users' eye gaze as feedback to improve the uPLT. The core idea revolves around the hypothesis that Web pages exhibit high and low regions of collective interest, where a user may be interested in certain parts of the page and not interested in certain other parts. We design a system called WebGaze that exploits the regions of collective interest to improve uPLT. Our user study across 100 users and 45 Web pages shows that WebGaze improves uPLT compared to three alternate strategies for 73% of the Web pages.

5.8 Conclusions

There has been a recent interest in making user experience the central issue in Web optimizations. Currently, user experience is divorced from Web page performance metrics. We systematically study the user-perceived page load time metric, or uPLT, to characterize user experience with respect to traditional metrics. We then make a case for using users' eye gaze as feedback to improve the uPLT. The core idea revolves around the hypothesis that Web pages exhibit high and low regions of collective interest, where a user may be interested in certain parts of the page and not interested in certain other parts. We design a system called WebGaze that exploits the regions of collective interest to improve uPLT. Our user study across 100 users and 45 Web pages shows that WebGaze improves uPLT compared to three alternate strategies for 73% of the Web pages.

Chapter 6

Dissertation Conclusion

Traditional systems have regarded backscatter communication as a limited ability communication method that bound the RFID to short range inventory system. In this dissertation, we took a fundamental approach and show that it is better than most communication devices that have limited energy resource. We incorporated this understanding into the design of protocols and systems. By doing so, we were able to design and build a practical system that transforms backscatter communication from limited ability inventory system to dexterous communication platform that spend smaller amounts of energy. Specifically, we make following contributions:

- **Studies in backscatter communications for Internet of things:** Our implemented passive backscatter-based tag-to-tag communication platform holds a tremendous potential in realizing ubiquitous IoT platforms. Our platform built on a learning technique enabling tags to determine the right channel to use for the communication with other tags. We have demonstrated tag-to-tag links operating reliably at various environment settings. We extended capabilities of passive RF tags to a different regime. They are capable of channel measurements of the backscatter channel that correlates very well with environmental changes around the tags. This latter ability translates to human activity recognition. We also demonstrated that a standard RFID setup can exploit phase in order to perform very accurate ranging just by using a single antenna that exploits FM radar principles. The accuracy is competitive or better with better operating ranges seen in literature while requiring a straightforward set up. We have experimentally demonstrated median ranging accuracy of about 5 cm and localization accuracy of about 10cm at distances over 4m.

- **Improving user experience for Web access and streaming video based on the gaze feedback:** Our work could pave the way to automate mechanisms for improving user experience for video and Web. This can level the playing field. In addition to improving user experience, our work on foveated video streaming significantly reduces data usage without adversely affecting user experience. Reducing data usage especially has an impact in economically developing regions of the world where data costs are much higher as a percentage of earnings. Also, since DSL or 2G/3G networks are often unable to provide high-quality user experience. Our research focus on prioritizing the objects in order to optimize user experience in given resources, therefore, the end user can experience the best quality out of given resources.

This dissertation took two different approaches, we first minimized the use of energy resources; therefore, we were able to realize batteryless communication. Then we exploit the best use of given network resources by prioritizing the objects; hence, we were able to design user experience based system.

Wireless networking has witnessed a paradigm shift over past three to five years. The field has been transformed from pursuing the communication platform toward faster at the cost of energy efficiency, into designing networked systems that tightly incorporate an understanding of energy aware communication platform, despite at slower speeds. This has allowed us to revisit and address contemporary problems on battery equipped devices and power harvesting devices.

The next few years are going to be exciting for wireless research because of its ability to change peoples lives through diverse applications from smartphones and RFIDs to critical fields such as healthcare. However, as wireless connectivity gets incorporated into diverse devices and applications, the density of wireless deployments increases. As a result, there is a need to design systems that can address battery-less communication platform at a very large scale (100s of devices in a regular sized room). While this dissertation takes the first few steps in this direction, addressing this problem at such a large scale in practice remains a longer term challenge.

6.1 Looking Forward

I would like to continue to explore research problems in the field of the batteryless communication system and human behavior based multimedia optimization. Following I describe two research directions, I will pursue soon for my research career.

- **Pervasive sensing through backscatter:** Backscatter has been proved its potential to enable gesture recognition and human activity recognition in the dissertation. The CSI based passive sensing technology has achieved high accuracy in gesture and activity recognition [170, 181]. For example, WiKey demonstrates the keyboard stroke classification in 96% accuracy for 26 alphabets through machine learning technology [24]; EQ-Radio shows the emotion recognition in 87% accuracy for four fundamental human emotions – excited, happy, angry and sad – through heartbeat analysis [193]. These approaches have proved the possibility of passive sensing gesture and activity classification even further. Since we have BCSI platform and knowledge of signal processing, we are planning to extend our *BARNET* for more diverse activity recognition. Instead of using CSI, we have multi-channel transmitter based on phase shift schemes; we can exploit this approach and collect high precision data in the analog domain, and make the connection on digital domain. It is necessary to avoid high definition ADC, because our *BARNET* platform designed for batteryless communication, and its key component is removing ADC. To migrate to a digital domain, we set three different threshold settings – high, medium, and low – of the reference value on the receiver side. Therefore, we can build three-dimensional digital phase response map. Multidimensional digital domain representation without any special tools, we could use as pseudo analog domain, without analog component, it makes our tool powerful. The intuition is that IoT device will pervasive our daily lives and able to monitor human's fundamental movements and health status without any particular devices. To realize this capability, we need efforts from multiple perspectives, including a vast amount of data collection from the very tightly controlled environment, and classification of the collected data through machine learning technology, and its extension to realistic conditions.

- **Saliency-based 360 live streaming:** A saliency has established its functional mechanism in image based perceptual and cognitive research. This saliency in the video system serves as the index for looking ahead of spotlighted regions. In 360 video, the only partial amount of visual contents is viewed at the client side, while entire contents are fetched to be ready for the possible head movement. In state-of-the-art 360 dynamic streaming technique differentiate serving resolutions, where the client is looking at serve in its highest resolution, and the rest of regions are fetched based on network environment to avoid discontinuity of service for any possible movements [89]. This technique can serve the higher quality of service. However, it always starves its baseline quality at the early stage of head movement; it always requires time to fill the buffer with high-quality contents. Since we can prefetch visual regions in an extra buffer based on saliency map, we can avoid the poor quality experience. The study shows that people view a pictorial medium his/her eyes exhibit quick jerky movements interspersed with relatively long stops known as fixations that define the viewer's attention [79]. The saliency technique can apply to estimate those fixations. To realize this capability and bring it to the research topic, we have to conduct the investigation in multiple perspectives. The research intuition is that not every saliency regions have to be buffered to stand by all possible movements, only the smallest set of saliency regions are needed to be buffered based on the probability calculation. Through user studies, you can verify best possible 360 live streaming. Furthermore, we can investigate foveate technique those studied in our previous literature, for this 360 live streaming and virtual reality environment.

Bibliography

[1] 14 Rules for Faster-Loading Web Sites. http://stevesouders.com/hpws/rules.php.

[2] 902-928 MHZ 9 DBIC CIRCULAR POLARITY PANEL. http://www.lairdtech.com/products/s9028pcl-s9028pcr.

[3] Above-the-fold time (AFT): Useful, but not yet a substitute for user-centric analysis. http://bit.ly/29MBmip.

[4] Alexa: a commercial web traffic data and analytics provider. http://www.alexa.com/.

[5] bladeRF - the USB 3.0 Superspeed Software Defined Radio. http://nuand.com/.

[6] bladeRF USB 3.0 Superspeed Software Defined Radio Source Code. https://nuand.com/forums/.

[7] Document Object Model(DOM). https://www.w3.org/DOM/.

[8] Globaleventhandlers.onload. https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/onload.

[9] Going Beyond OnLoad: Measuring Performance that matters. http://oreil.ly/2cpaUhV.

[10] Hick's law: On the rate of gain of information. https://en.wikipedia.org/wiki/Hick%27s_law.

[11] How to use eye tracking on samsung galaxy s7 and galaxy s7 edge. http://bit.ly/29LDiqj.

[12] LNA Series 902-928MHz Low Noise Amplifier. http://rfbayinc.com/products_pdf/product_75.pdf.

[13] MCP6561, 1.8V Low-Power Push-Pull Output Comparator. http://www.microchip.com/downloads/en/DeviceDoc/22139C.pdf.

[14] Measuring the critical rendering path with Navigation Timing. https://developers.google.com/web/fundamentals/performance/critical-rendering-path/measure-crp?hl=en.

[15] Microworkers: Crowdsourcing platform. https://microworkers.com/.

[16] SPDY: An experimental protocol for a faster web. https://www.chromium.org/spdy/spdy-whitepaper.

[17] Wideband 2.5 GHz, 37 dB Isolation at 1 GHz, CMOS 1.65 V to 2.75 V, 4:1 Mux/SP4T. http://www.analog.com/media/en/technical-documentation/data-sheets/ADG904_904R.pdf.

[18] RASPBERRY PI 1 MODEL B+. https://www.raspberrypi.org/products/model-b-plus/, 2014.

[19] Heba Abdelnasser, Moustafa Youssef, and Khaled A. Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *IEEE Conference on Computer Communications*, INFOCOM '15, pages 1472–1480, April 2015.

[20] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via rf body reflections. In *12th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '15, pages 279–292, 2015.

[21] AEGIS project. Opengazer: open-source gaze tracker for ordinary webcams. http://www.inference.phy.cam.ac.uk/opengazer/.

[22] M. Elgin Akpınar and Yeliz Yesilada. *Vision Based Page Segmentation Algorithm: Extended and Perceived Success*, pages 238–252. Springer International Publishing, 2013.

[23] Tareq Alhmiedat, Ghassan Samara, and Amer O. Abu Salem. An indoor fingerprinting localization approach for zigbee wireless sensor networks. *European Journal of Scientific Research*, abs/1308.1809(2):190–202, July 2013.

[24] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. Keystroke recognition using wifi signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 90–102. ACM, 2015.

[25] Amazon Fire Phone with Dynamic Perspective. Understanding the dynamic perspective ui. https://developer.amazon.com/public/solutions/devices/fire-phone/docs/understanding-the-dynamic-perspective-ui.

[26] Akshay Athalye, Vladimir Savic, Miodrag Bolic, and Petar M Djuric. Novel semi-passive rfid system for indoor localization. *Sensors Journal, IEEE*, 13(2):528–537, 2013.

[27] Salah Azzouzi, Markus Cremer, Uwe Dettmar, Rainer Kronberger, and Thomas Knie.

[28] Dinesh Bharadia, Kiran Raj Joshi, Manikanta Kotaru, and Sachin Katti. Backfi: High throughput wifi backscatter. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 283–296, 2015.

[29] Joshua Bixby. Case study: The impact of HTML delay on mobile business metrics. `http://www.webperformancetoday.com/2011/11/23/case-study-slow-page-load-mobile-business-metrics/`, November 2011.

[30] Joshua Bixby. 4 awesome slides showing how page speed correlates to business metrics at walmart.com. `http://bit.ly/1jfACl2`, February 2012.

[31] Agnieszka Bojko. Using Eye Tracking to Compare Web Page Designs: A Case Study. In *Journal of Usability Studies*, volume 1, pages 12–120, May 2006.

[32] Ayub Bokani. Empirical evaluation of real-time video foveation. In *Proceedings of the Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, VideoNext '14, pages 45–46. ACM, 2014.

[33] Ali Borji, Dicky N. Sihite, and Laurent Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, 2013.

[34] Martin A. Brown. Traffic control how to, overview of the capabilities and implementation of traffic control under linux. `http://www.tldp.org/HOWTO/html_single/Traffic-Control-HOWTO/`, 2006.

[35] Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46(3):33:1–33:33, 2014.

[36] Georg Buscher, Edward Cutrell, and Meredith Ringel Morris. What do you see when you're surfing?: Using eye tracking to predict salient regions of web pages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 21–30. ACM, 2009.

[37] Michael Butkiewicz, Harsha V. Madhyastha, and Vyas Sekar. Understanding website complexity: Measurements, metrics, and implications. In *Proceedings on Internet Measurement Conference*, IMC '11, pages 313–328, 2011.

[38] Michael Butkiewicz, Daimeng Wang, Zhe Wu, Harsha V. Madhyastha, and Vyas Sekar. Klotski: Reprioritizing web content to improve user experience on mobile devices. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, NSDI '15, pages 439–453. USENIX Association, 2015.

[39] Wenyi Che, Yuqing Yang, Conghui Xu, Na Yan, Xi Tan, Qiang Li, Hao Min, and Jie Tan. Analysis, design and implementation of semi-passive gen2 tag. In *IEEE International Conference on RFID*, RFID '09, pages 15–19, 2009.

[40] Chrome Debug Team. Chrome remote debugging. `https://developer.chrome.com/devtools/docs/debugger-protocol`.

[41] Gregory Ciotti. 7 marketing lessons from eye-tracking studies. `https://blog.kissmetrics.com/eye-tracking-studies/`.

[42] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2015–2020 White Paper. `http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf`.

[43] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, June 2001.

[44] Christine A. CuRcIo, Kenneth R. Sloan, Orin Packer, Anita E. Hendricson, and Robert E. Kalina. Distribution of cones in human and monkey retina: individual variability and radial asymmetry. In *Science*, number 4801 in SCIENCE '87, pages 579–582. American Association for the Advancement of Science, 1987.

[45] Hadar Dagan, Aviv Shapira, Adam Teman, Anatoli Mordakhay, Samuel Jameson, Evgeny Pikhay, Vladislav Dayan, Yakov Roizin, Eran Socher, and Alexander Fish. A low-power low-cost 24 ghz rfid tag with a c-flash based embedded memory. *IEEE Journal of Solid-State Circuits*, 49(9):1942–1957, 2014.

[46] Alanson P. Sample Daniel J. Yeager and Joshua R. Smith. *RFID Handbook: Applications, Technology, Security, and Privacy*, chapter WISP: A Passively Powered UHF RFID Tag with Sensing and Computation. CRC Press, 2008.

[47] Szymon Deja. Real-time system for eye detection and tracking of computer user using webcam. Master's thesis, A G H University of Science and Technology, 2010.

[48] LF Dell'Osso and RB Daroff. Congenital nystagmus waveforms and foveation strategy. *Documenta Ophthalmologica*, 39(1):155–182, 1975.

[49] Marcel Dischinger, Massimiliano Marcon, Saikat Guha, Krishna P. Gummadi, Ratul Mahajan, and Stefan Saroiu. Glasnost: Enabling end users to detect traffic differentiation. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI '10, pages 27–27. USENIX Association, 2010.

[50] Daniel M. Dobkin. *The RF in RFID: Passive UHF RFID in Practice*. Newnes, 2007.

[51] Michael Dorr, Thomas Martinetz, Karl R. Gegenfurtner, and Erhardt Barth. Variability of eye movements when viewing dynamic natural scenes. *Journal of Vision*, 10(10), 2010.

[52] Dynamic Adaptive Streaming over HTTP. http://en.wikipedia.org/wiki/dynamic-adaptive-streaming-over-http.

[53] EPC Global. EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID. `http://www.gs1.org/sites/default/files/docs/epc/Gen2_Protocol_Standard.pdf`.

[54] Sukru Eraslan, Yeliz Yesilada, and Simon Harper. Eye tracking scanpath analysis techniques on web pages: A survey, evaluation and comparison. *Journal of Eye Movement Research*, 9(1), 2015.

[55] Yunlong Feng, Gene Cheung, Wai tian Tan, and Yusheng Ji. Hidden markov model for eye gaze prediction in networked video streaming. In *IEEE International Conference on Multimedia and Expo*, ICME '11, pages 1–6, July 2011.

[56] Onur Ferhat and Fernando Vilariño. Low cost eye tracking: The current panorama. *Computational Intelligence and Neuroscience*, vol. 3(No. 2):1–14, 2016.

[57] Klaus Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. Wiley Publishing, 2nd edition, 2003.

[58] Shengnan Gai, Eui-Jung Jung, and Byung-Ju Yi. Localization algorithm based on zigbee wireless sensor network with application to an active shopping cart. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4571–4576, Sep 2014.

[59] Renwu Gao, Seiichi Uchida, Asif Shahab, Faisal Shafait, and Volkmar Frinken. Visual saliency models for text detection in real world. *PloS ONE*, vol. 9(No. 12):1–20, 2014.

[60] GazePointer. Control mouse cursor position with your eyes via webcam. `http://gazepointer.sourceforge.net/`.

[61] Wilson S Geisler and Jeffrey S Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In *Electronic Imaging*, pages 294–305. International Society for Optics and Photonics, 1998.

[62] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(10):1915–1926, Oct 2012.

[63] Google Developers. PageSpeed Tools: The PageSpeed tools analyze and optimize your site. `https://developers.google.com/speed/pagespeed/`.

[64] Google Developers. Simulate Mobile Devices with Device Mode. http://bit.ly/2eoizLk.

[65] Ramesh Govindan. Modeling HTTP/2 speed from HTTP/1 traces. In *Proceedings of 17th International Conference on Passive and Active Measurement*, volume 9631 of *PAM '16*, page 233. Springer, 2016.

[66] Ilya Grigorik. Analyzing critical rendering path performance. http://bit.ly/1ORhrNj.

[67] Chunmei Han, Kaishun Wu, Yuxi Wang, and Lionel M. Ni. Wifall: Device-free fall detection by wireless networks. In *IEEE Conference on Computer Communications*, INFOCOM '14, pages 271–279, April 2014.

[68] D.W. Hansen and Qiang Ji. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):478–500, March 2010.

[69] Cory Hekimian-Williams, Brandon Grant, Xiuwen Liu, Zhenghao Zhang, and Piyush Kumar. Accurate localization of RFID tags using phase difference. In *IEEE International Conference on RFID*, RFID '10, pages 89–96. IEEE, April 2010.

[70] Gordon Henderson. Wiring Pi, GPIO Interface library for the Raspberry Pi. http://wiringpi.com/, 2015.

[71] Hao-Chiao Hong and Guo-Ming Lee. A 65-fj/conversion-step 0.9-v 200-ks/s rail-to-rail 8-bit successive approximation adc. *IEEE Journal of Solid-State Circuits*, 42(10):2161–2168, 2007.

[72] HTTP Live Streaming. http://en.wikipedia.org/wiki/HLS.

[73] Jordy Huiting, Hubert Flisijn, Andre B.J. Kokkeler, and Gerard J.M. Smit. Exploiting phase measurements of EPC Gen2 RFID tags. In *RFID-Technologies and Applications (RFID-TA), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[74] Daniel Imms. Speed index: Measuring page load time a different way. https://www.sitepoint.com/speed-index-measuring-page-load-time-different-way/, September 2014.

[75] Impinj Corporation. SDK for RFID. http://www.impinj.com/.

[76] Impinj Corporation. Speedway RFID reader series. http://www.impinj.com/Speedway_Revolution_UHF_RFID_Reader.aspx.

[77] Internet Live Stats. Google Search Statistics. http://www.internetlivestats.com/google-search-statistics/.

[78] Laurent Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.

[79] John Findlay and Robin Walker. Human saccadic eye movements. `http://www.scholarpedia.org/article/Human_saccadic_eye_movements`.

[80] Yasha Karimi, Akshay Athalye, Samir R. Das, Petar Djurić, and Milutin Stanaćević. Design of backscatter-based tag-to-tag system. In *IEEE International Conference on RFID*, RFID '17, 2017. to appear.

[81] Udo Karthaus and Martin Fischer. Fully integrated passive uhf rfid transponder ic with 16.7-μw minimum rf input power. *IEEE Journal of Solid-State Circuits*, 38(10):1602–1608, 2003.

[82] Bryce Kellogg, Aaron Parks, Shyamnath Gollakota, Joshua R Smith, and David Wetherall. Wi-fi backscatter: internet connectivity for RF-powered devices. In *Proceedings of the 2014 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '14, pages 607–618. ACM, 2014.

[83] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. Bringing gesture recognition to all devices. In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI '14, pages 303–316, 2014.

[84] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R. Das. Improving user perceived page load times using gaze. In *14th USENIX Symposium on Networked Systems Design and Implementation*, NSDI '17, pages 545–559, Boston, MA, 2017. USENIX Association.

[85] Kit Eaton. How one second could cost amazon 1.6 billion in sales. `http://bit.ly/1Beu9Ah`.

[86] Oleg Komogortsev. Predictive perceptual compression for real time video communication. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, pages 220–227, 2004.

[87] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, 2016.

[88] Rupa Krishnan, Harsha V Madhyastha, Sridhar Srinivasan, Sushant Jain, Arvind Krishnamurthy, Thomas Anderson, and Jie Gao. Moving beyond end-to-end path information to optimize CDN performance. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC'09, pages 190–201. ACM, 2009.

[89] Evgeny Kuzyakov and David Pio. Next-generation video encoding techniques for 360 video and VR. `https:`

//code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/.

[90] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Effiicient backprop. In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 9–50. Springer-Verlag, 1998.

[91] Dongheng Li, D. Winfield, and D.J. Parkhurst. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '05, pages 79–79, June 2005.

[92] Hanchuan Li, Can Ye, and Alanson P. Sample. IDSense: A human object interaction detection system based on passive UHF RFID. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 2555–2564, 2015.

[93] Robert LiKamWa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 69–82. ACM, 2013.

[94] Tianci Liu, Lei Yang, Qiongzheng Lin, Yi Guo, and Yunhao Liu. Anchor-free backscatter positioning for rfid tags with high accuracy. In *IEEE International Conference on Computer Communications*, INFOCOM '14, pages 379–387. IEEE, 2014.

[95] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. Ambient backscatter: Wireless communication out of thin air. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '13, pages 39–50. ACM, 2013.

[96] Vincent Liu, Vamsi Talla, and Shyamnath Gollakota. Enabling instantaneous feedback with full-duplex backscatter. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 67–78. ACM, 2014.

[97] Yunhao Liu, Yiyang Zhao, Lei Chen, Jian Pei, and Jinsong Han. Mining frequent trajectory patterns for activity monitoring using radio frequency tag arrays. *IEEE Transactions on Parallel and Distributed Systems*, 23(11):2138–2149, 2012.

[98] Feng Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV '11, pages 153 –160. IEEE, 2011.

[99] Feng Lu, Y. Sugano, T. Okabe, and Y. Sato. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):2033–2046, Oct 2014.

[100] Xiang Lu, Lei Xie, Yafeng Yin, Wei Wang, Baoliu Ye, and Sanglu Lu. Efficient localization based on imprecise anchors in rfid system. In *IEEE International Conference on Communications*, ICC '14, pages 142–147. IEEE, June 2014.

[101] Stefan Mathe and Cristian Sminchisescu. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *Proceedings of the 12th European Conference on Computer Vision*, ECCV '12, pages 842–856. Springer-Verlag, 2012.

[102] Mike Belshe and Roberto Peon and Martin Thomson Ed. Hypertext Transfer Protocol Version 2 (HTTP/2). `https://tools.ietf.org/html/rfc7540`.

[103] Emiliano Miluzzo, Tianyu Wang, and Andrew T. Campbell. Eyephone: Activating mobile phones with your eyes. In *Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds*, MobiHeld '10, pages 15–20. ACM, 2010.

[104] Carlos H Morimoto and Marcio RM Mimica. Eye gaze tracking techniques for interactive applications. *Journals on Computer Vision and Image Understanding*, 98(1):4–24, 2005.

[105] Motorola Solutions. FX RFID reader series. `http://www.motorolasolutions.com/US-EN/Business+Product+and+Services/RFID/RFID+Readers`.

[106] Mozilla Developer Network. Remote debugging. `https://developer.mozilla.org/en-US/docs/Tools/Remote_Debugging`.

[107] Ravi Netravali, Ameesh Goyal, James Mickens, and Hari Balakrishnan. Polaris: Faster page loads using fine-grained dependency tracking. In *Proceedings of the 13th USENIX Conference on Networked Systems Design and Implementation*, NSDI '16, pages 123–136. USENIX Association, 2016.

[108] Lionel M Ni, Yunhao Liu, Yiu Cho Lau, and Abhishek P Patil. LANDMARC: indoor location sensing using active RFID. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, volume 10 of *PerCom '03*, pages 701–710. IEEE, 2003.

[109] Pavel V. Nikitin, Rene Martinez, Shashi Ramamurthy, Hunter Leland, Gary Spiess, and K.V.S. Rao. Phase based spatial identification of UHF RFID tags. In *IEEE International Conference on RFID*, RFID '10, pages 102–109. IEEE, April 2010.

[110] Pavel V. Nikitin, Shashi Ramamurthy, Rene Martinez, and K.V.S. Rao. Passive tag-to-tag communication. In *IEEE International Conference on RFID*, RFID '12, pages 177 –184. IEEE, april 2012.

[111] Christi O'Connell. Eyetracking and web site design. `https://www.usability.gov/get-involved/blog/2010/03/eyetracking.html`, March 2010.

[112] Mary Catherine O'Connor. Can RFID save brick-and-mortar retailers after all? Fortune magazine, April 2014.

[113] OECD. Average and median advertised download speeds, fixed broadband. `http://bit.ly/2afJtY3`, July 2015.

[114] OpenEyes. A open-source open-hardware toolkit for low-cost real-time eye tracking. `http://thirtysixthspan.com/openEyes/software.html`.

[115] Charlie Osborne. Google patent hints at monetizing glass, tracking user engagement. `http://www.zdnet.com/article/google/-patent-hints-at-monetizing-glass-tracking-/user-engagement/`.

[116] Chris Leo Palermino. Online video will account for 80 percent of the world's internet traffic by 2019. Technical report, Digital Trends, 2015.

[117] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, IJCAI '16, pages 3839–3845. AAAI, 2016.

[118] Raúl Parada, Joan Melià-Seguí, Anna Carreras, Marc Morenza-Cinos, and Rafael Pous. Measuring user-object interactions in IoT spaces. In *IEEE International Conference on RFID Technology and Applications*, RFID-TA '15, pages 52–58, 2015.

[119] Jincheol Park, Sanghoon Lee, A Bovik, Jincheol Park, Sanghoon Lee, and Alan C Bovik. 3D visual discomfort prediction: vergence, foveation, and the physiological optics of accommodation. *IEEE Journal of Selected Topics in Signal Processing*, 8(3):415–427, 2014.

[120] Aaron N. Parks, Angli Liu, Shyamnath Gollakota, and Joshua R. Smith. Turbocharging ambient backscatter communication. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '14, pages 619–630. ACM, 2014.

[121] Jeffrey S Perry and Wilson S Geisler. Gaze-contingent real-time simulation of arbitrary visual fields. In *Electronic Imaging*, pages 57–69. International Society for Optics and Photonics, 2002.

[122] Mike Petrovich. Going beyond onload: Measuring performance that matters. `http://radar.oreilly.com/2013/10/going-beyond-onload-measuring-performance-that-matters.html`, October 2013.

[123] Joonas Pihlajamaa. Benchmarking Raspberry Pi GPIO Speed. `http://bit.ly/1trFzt9`, 2015.

[124] Michael Poole. Streaming Entertainment Surges to 70 Percent of Internet Traffic During Peak Periods. `http://bit.ly/2akpgxZ`, December 2015.

[125] Emil Protalinski. Streaming services now account for over 70% of peak traffic in North America, Netflix dominates with 37%. `http://bit.ly/1m6HhlI`, December 2015.

[126] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th Annual International Conference on Mobile Computing Networking*, MobiCom '13, pages 27–38, 2013.

[127] Benjamin Ransford, Shane Clark, Mastooreh Salajegheh, and Kevin Fu. Getting things done on computational rfids with energy-aware checkpointing and voltage-aware scheduling. In *Proceedings of USENIX Workshop on Power Aware Computing and Systems*, HotPower '08, pages 5–5, 2008.

[128] Dan Rayburn. The Adoption Of 4K Streaming Will Be Stalled By Bandwidth, Not Hardware & Devices. `http://blog.streamingmedia.com/2015/01/4k-streaming-bandwidth-problem.html`, january 2015.

[129] Paula Rosenblum. How walmart could solve its inventory problem and improve earnings. Forbes magazine, May 2014.

[130] Dmitry Rudoy, Dan B. Goldman, Eli Shechtman, and Lihi Zelnik-Manor. Learning video saliency from human gaze using candidate selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 1147–1154, June 2013.

[131] Jihoon Ryoo and Samir R. Das. Phase-based ranging of rfid tags with applications to shopping cart localization. In *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '15, pages 245–249. ACM, 2015.

[132] Jihoon Ryoo, Kiwon Yun, Dimitris Samaras, Samir R. Das, and Gregory J. Zelinsky. Design and evaluation of a foveated video streaming service for commodity client devices. In *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, pages 6:1–6:11. ACM, 2016.

[133] Sandvine Incorporated ULC. GLOBAL INTERNET PHENOMENA REPORT. Technical report, Sandvine Intelligent Broadband Networks, 2014.

[134] Brahima Sanou. The World in 2013 ICT Facts and Figures, http://www.itu.int/en/itu-d/statistics/documents/facts/ictfactsfigures2013-e.pdf. `http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2013-e.pdf`, 2013.

[135] Jens Sauerbrey, Doris Schmitt-Landsiedel, and Roland Thewes. A 0.5-v 1-/spl mu/w successive approximation adc. *IEEE Journal of Solid-State Circuits*, 38(7):1261–1265, 2003.

[136] Longfei Shangguan, Zhenjiang Li, Zheng Yang, Mo Li, and Yunhao Liu. Otrack: Order tracking for luggage in mobile rfid systems. In *IEEE International Conference on Computer Communications*, INFOCOM '13, pages 3066–3074. IEEE, 2013.

[137] Gaurav Sharma, Frédéric Jurie, and Cordelia Schmid. Discriminative spatial saliency for image classification. In *Conference on Computer Vision and Pattern Recognition*, CVPR '12, pages 3506–3513. IEEE, 2012.

[138] Chengyao Shen, Xun Huang, and Qi Zhao. Predicting eye fixations on webpage with an ensemble of early features and high-level representations from deep network. *IEEE Transactions on Multimedia*, vol. 17(No. 11):2084–2093, 2015.

[139] Chengyao Shen and Qi Zhao. *Webpage Saliency*, pages 33–46. Springer International Publishing, 2014.

[140] Z. Shen, A. Athalye, and P. M. Djuri. Phase cancellation in backscatter-based tag-to-tag communication systems. *IEEE Internet of Things Journal*, 3(6):959–970, 2016.

[141] Gang Zhou Shuangquan Wang. A review on radio based activity recognition. *Digital Communications and Networks*, 1(1):20 – 29, 2015.

[142] Stephan Sigg, Ulf Blanke, and Gerhard Troster. The telepathic phone: Frictionless activity recognition from wifi-rssi. In *IEEE International Conference on Pervasive Computing and Communications*, PerCom '14, pages 148–155, 2014.

[143] Ashiwan Sivakumar, Shankaranarayanan Puzhavakath Narayanan, Vijay Gopalakrishnan, Seungjoon Lee, Sanjay Rao, and Subhabrata Sen. Parcel: Proxy assisted browsing in cellular networks for energy and latency reduction. In *Proceedings of the 10th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '14, pages 325–336. ACM, 2014.

[144] SMI Eye Tracking Glasses 2 Wireless. `http://bit.ly/29YWaDa`.

[145] SMI Sensomotoric Instruments. Eye & gaze tracking systems. `http://www.smivision.com/`.

[146] Robert Solso. *Cognition and the Visual Arts*. MIT Press, 1996.

[147] Steve Souders. Moving beyond window.onload(). `https://www.stevesouders.com/blog/2013/05/13/moving-beyond-window-onload/`, May 20.

[148] Jeremiah D. Still and Christopher M. Masciocchi. A saliency model predicts fixations in web interfaces. In *5th International Workshop on Model Driven Development of Advanced User Interfaces*, MDDAUI '10, pages 25–28, 2010.

[149] Mark Stoopman, Shady Keyrouz, Hubregt J Visser, Kathleen Philips, and Wouter A Serdijn. Co-design of a cmos rectifier and small loop antenna for highly sensitive rf energy harvesters. *IEEE Journal of Solid-State Circuits*, 49(3):622–634, 2014.

[150] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. Learning-by-synthesis for appearance-based 3d gaze estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1821–1828, 2014.

[151] Gabriel Svennerberg. Page Load Times vs Conversion Rates. http://www.svennerberg.com/2008/12/page-load-times-vs-conversion-rates/, December 2008.

[152] Texas Instruments. MSP430 ultra-low-power Microcontrollers. http://www.ti.com/lsds/ti/microcontrollers_16-bit_32-bit/msp/overview.page.

[153] The Chromium Projects. QUIC, a multiplexed stream transport over UDP. http://bit.ly/2cDBKig.

[154] The Eye Tribe. affordable eye tracking technology provider. https://theeyetribe.com/.

[155] The International Telecommunication Union. ICT Facts and Figures 2016. http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2016.pdf, 2016.

[156] Tobii. Provider of eye control and eye tracking products. http://www.tobii.com/en/.

[157] Michael Buettner Tom Bergan, Simon Pelchat. Rules of thumb for HTTP/2 server push. http://bit.ly/2d4O1RN.

[158] Nhan Tran, Bomson Lee, and Jong-Wook Lee. Development of long-range uhf-band rfid tag chip using schottky diodes in standard cmos technology. In *IEEE Radio Frequency Integrated Circuits (RFIC) Symposium*, pages 281–284, 2007.

[159] Martín Varela, Lea Skorin-Kapov, Toni Mäki, and Tobias Hoßfeld. QoE in the Web: A dance of design and performance. In *7th International Workshop on Quality of Multimedia Experience*, QoMEX '15, pages 1–7, 2015.

[160] Matteo Varvello, Jeremy Blackburn, David Naylor, and Konstantina Papagiannaki. Eyeorg: A platform for crowdsourcing web quality of experience measurements. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 399–412. ACM, 2016.

[161] Rita M. Vick and Curtis S. Ikehara. Methodological issues of real time data acquisition from multiple sources of physiological data. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, HICSS '03, page 7, Jan 2003.

[162] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[163] Peripheral vision. https://en.wikipedia.org/wiki/Peripheral_vision.

[164] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M. Ni. We can hear you with wi-fi! In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 593–604, 2014.

[165] Hsueh-Cheng Wang and Marc Pomplun. The attraction of visual attention to texts in real-world scenes. *Journal of Vision*, vol. 12(issue 6):26–26, 2012.

[166] Jue Wang, Fadel Adib, Ross Knepper, Dina Katabi, and Daniela Rus. Rf-compass: Robot object manipulation using rfids. In *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*, MobiCom '13, pages 3–14. ACM, 2013.

[167] Jue Wang and Dina Katabi. Dude, Where's My Card?: RFID Positioning That Works with Multipath and Non-line of Sight. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '13, pages 51–62. ACM, 2013.

[168] Peng Wang, Jingdong Wang, Gang Zeng, Jie Feng, Hongbin Zha, and Shipeng Li. Salient object detection for searched web images via global saliency. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '12, pages 3194–3201. IEEE, 2012.

[169] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 851–860, 2016.

[170] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 65–76. ACM, 2015.

[171] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. Demystifying page load performance with wprof. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, NSDI '13, pages 473–486. USENIX Association, 2013.

[172] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. How speedy is spdy? In *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI '14, pages 387–399. USENIX Association, 2014.

[173] Xiao Sophia Wang, Arvind Krishnamurthy, and David Wetherall. Speeding up web page loads with shandian. In *Proceedings of the 13th USENIX Conference on Networked Systems Design and Implementation*, NSDI '16, pages 109–122. USENIX Association, 2016.

[174] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 617–628, 2014.

[175] Zhou Wang and Alan C Bovik. Embedded foveation image coding. *IEEE Transactions on Image Processing*, 10(10):1397–1410, 2001.

[176] Zhou Wang and Alan C. Bovik. Foveated image and video coding. *Digital Video, Image Quality and Perceptual Coding*, pages 431–457, 2006.

[177] WebPagetest: website performance testing service. `http://www.webpagetest.org/`.

[178] Erroll Wood and Andreas Bulling. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 207–210. ACM, 2014.

[179] Sean Work. How loading time affects your bottom line. `https://blog.kissmetrics.com/loading-time/`.

[180] Hongren Wu and Kamisetty-Ramamohan Rao. *Digital Video Image Quality and Perceptual Coding (Signal Processing and Communications)*. CRC Press, Inc., 2005.

[181] Wei Xi, Dong Huang, Kun Zhao, Yubo Yan, Yuanhang Cai, Rong Ma, and Deng Chen. Device-free human activity recognition using csi. In *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition*, CSAR '15, pages 31–36. ACM, 2015.

[182] Wei Xi, Jizhong Zhao, Xiang-Yang Li, Kun Zhao, Shaojie Tang, Xue Liu, and Zhiping Jiang. Electronic frog eye: Counting crowd using wifi. In *IEEE Conference on Computer Communications*, INFOCOM '14, pages 361–369, 2014.

[183] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI '15, pages 3995–4001, 2015.

[184] Jianbo Yang. cnn-timeseries: Use cnn to classify time series data for activity recognition. `https://github.com/sibosutd/cnn-timeseries`, 2017.

[185] Jimei Yang and Ming-Hsuan Yang. Top-down visual saliency via joint CRF and dictionary learning. In *Conference on Computer Vision and Pattern Recognition*, CVPR '12, pages 2296–2303. IEEE, 2012.

[186] Alfred L. Yarbus. *Eye Movements and Vision*. Vision Science: Photons to Phenomenology, 1967.

[187] Kiwon Yun, Yifan Peng, Dimitris Samaras, Gregory J. Zelinsky, and Tamara L. Berg. Exploring the role of gaze behavior and object detection in scene understanding. *Frontiers in Psychology*, 4(917), 2013.

[188] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, MULTIMEDIA '06, pages 815–824. ACM, 2006.

[189] Hong Zhang, Jeremy Gummeson, Benjamin Ransford, and Kevin Fu. Moo: A batteryless computational RFID and sensing platform. Technical Report UM-CS-2011-020, Department of Computer Science, University of Massachusetts Amherst, Amherst, MA, June 2011.

[190] Ouyang Zhang and Kannan Srinivasan. Mudra: User-friendly fine-grained gesture recognition using wifi signals. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 83–96, 2016.

[191] Pengyu Zhang, Pan Hu, Vijay Pasikanti, and Deepak Ganesan. Ekhonet: High speed ultra low-power backscatter for next generation sensors. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, pages 557–568. ACM, 2014.

[192] Yanxia Zhang, Andreas Bulling, and Hans Gellersen. Pupil-canthi-ratio: A calibration-free method for tracking horizontal gaze direction. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 129–132. ACM, 2014.

[193] Mingmin Zhao, Fadel Adib, and Dina Katabi. Emotion recognition using wireless signals. In *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking*, MobiCom '16, pages 95–108. ACM, 2016.

[194] Qi Zhao and Christof Koch. Advances in learning visual saliency: From image primitives to semantic contents. In *Neural Computation, Neural Devices, and Neural Prosthesis*, chapter 14, pages 335–360. Springer, 2014.

[195] Yiyang Zhao, Yunhao Liu, and Lionel M Ni. Vire: Active rfid-based localization using virtual reference elimination. In *International Conference on Parallel Processing*, ICPP '07, pages 56–56. IEEE, 2007.

[196] Weiping Zhu, Jiannong Cao, Yi Xu, Lei Yang, and Junjun Kong. Fault-tolerant rfid reader localization based on passive rfid tags. *IEEE Transactions on Parallel and Distributed Systems*, 25(8):2065–2076, 2014.