The Design and Implementation of an Over-the-top Cloud-based Vertical Handover Decision Service for Heterogeneous Wireless Networks

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Science

---

by
Adam Jordan Hodges
August 2014

Accepted by:
Dr. Jim Martin, Committee Chair
Dr. Jason Hallstrom
Dr. Brian Malloy

Abstract

The widespread availability of heterogeneous wireless networks (hetnets) presents a resource allocation challenge to network operators and administrators. Overlapping network coverage should be utilized to its fullest extent, providing users with a fair share of bandwidth while maximizing the efficient use of the operator's resources. Currently, network selection occurs locally at the mobile device and does not take into account factors such as the state of other networks that might be available in the device's location. The local decision made by the device can often result in underutilization of network resources and a degraded user experience. This type of selfish network selection might not result in optimal bandwidth allocation when compared to approaches that make use of a centralized resource controller [5]. The decision making process behind the selection of these networks continues to be an open area of research, and a variety of algorithms have been proposed to solve this problem.

An over-the-top handover decision service treats each wireless access network in a hetnet as a black box, assuming detailed network topology and state information is unavailable to the handover decision algorithm. The algorithm then uses network data gathered empirically from users to provide them with a network selection service that considers the current conditions of available networks in a given location. This is a departure from past designs of vertical handover decision algorithms, which tend to approach the problem from the perspective of individual network operators. The wide range of radio access technologies operated by different network operators that are available to a device within a hetnet, coupled with the mobile data offload effort, is the primary motivator behind our novel choice in direction. This thesis documents the design and implementation of such an over-the-top vertical handover decision service.

Dedication

This paper is dedicated to all the friends and family who have helped me throughout my academic journey. I could not have accomplished any of this on my own without your love and support.

Acknowledgments

I must thank, first and foremost, Dr. Jim Martin for his unwavering patience and support throughout my undergraduate and graduate research projects. He has given me many unique opportunities to perform exciting research that has completely transformed my college experience.

I would like to thank Ryan Izard and Ke Xu for their friendship and assistance with WiMAX-related issues throughout the life of my thesis project.

Much thanks to Dr. Jason Hallstrom and Dr. Brian Malloy for being two of the most effective teachers that I have had the pleasure of learning from, and for serving on my committee.

I would also like to thank the various members of the open source community that contributed to the projects that were used in this thesis. FOSS is an important piece of the computing world that will continue to facilitate the rapid growth of the field.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The number of devices connected to the internet is growing at an enormous rate, as exemplified by the recent exhaustion of public IPv4 addresses. There is an immense variety of devices contributing to this growth, consisting of anything from refrigerators to smartphones. As this medley of connected devices surpassed the number of human beings on earth in 2009, the internet community has begun to use the more accurate moniker, the "Internet of Things" [10]. The rise in global availability of smartphones has been one of the major contributing factors to the growth of the Internet of Things. The June 2014 Ericsson Mobility report predicts that growth will continue to rapidly increase, with smartphone subscriptions expected to triple and mobile data traffic expected to increase tenfold by 2019 [27].

To keep up with the increasing demand from mobile users, wireless network deployments have also seen explosive growth. The Ericsson Mobility Report states that 85% of the world's population is currently within coverage of 2G GSM/EDGE networks, 60% is covered by 3G WCDMA/HSPA networks, and 20% is covered by 4G LTE networks [27]. The rapid growth in mobile data traffic is pushing network operators to find innovative ways to provide adequate network capacity to their customers. Cellular network technologies have continually evolved to provide higher rates of spectral efficiency, but are quickly approaching their theoretical limits [8]. The ownership and usage of spectrum in the US is heavily regulated by the FCC and the growth of wireless traffic will likely outpace any reallocation and private acquisition of spectrum [30].

Network operators, unable to increase capacity by further technological advancements or acquisition of spectrum, are forced to find other ways to add capacity to their networks. One method of increasing capacity is to increase the macro cell density in network deployments. The high cost of deploying new macro cells and the interference that dense cell networks experience make this approach infeasible. Network operators have instead turned to smaller cheap low-power cells, called pico and femto cells, to increase the cellular density of their deployments [8]. While this technique adds capacity, it is not a silver bullet as these cells still have relatively high deployment costs and interference issues.

Unable to solve the problem of spectrum scarcity solely through the increase of cell density, network operators have begun to focus on the use of unlicensed spectrum to offload traffic to other networks. Wireless local area network (WLAN) technologies such as 802.11 Wi-Fi and wireless personal area network (WPAN) technologies such as Bluetooth and ZigBee can be used as supplementary mobile data access technologies to a cellular mobile device. This offloading technique is growing in popularity with network operators as public Wi-Fi coverage is becoming increasingly ubiquitous. The Wireless Broadband Alliance predicts that by 2018, 20% of additional mobile data capacity will be provided by Wi-Fi offload [12].

1.1 Terminology

The advent of cloud-based services has brought about changes to the computing paradigm for many industries. Cloud providers typically utilize bulk data processing or collection to provide services that would otherwise be prohibitively expensive to users. These cloud services typically are manifest as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS). These services are made available through use of the public internet, with the goal of offloading workload from the clients. The concept of cloud services lends itself very well to the field of mobile computing, where devices have limited resources to perform complex tasks.

The combination of the overlapping availability of different radio access technologies (RATs) to mobile devices creates what are referred to as heterogeneous wireless networks (hetnets). As illustrated in figure 1.1, a heterogeneous wireless network can be comprised of various network technologies including legacy 2G/3G networks, Bluetooth, LTE, WiMAX, and WiFi. A modern mobile device makes use of multiple radios, switching its active network connection between these RATs as it roams in what is known as a vertical handoff.

An issue present in modern mobile devices with access to hetnets is the passive management

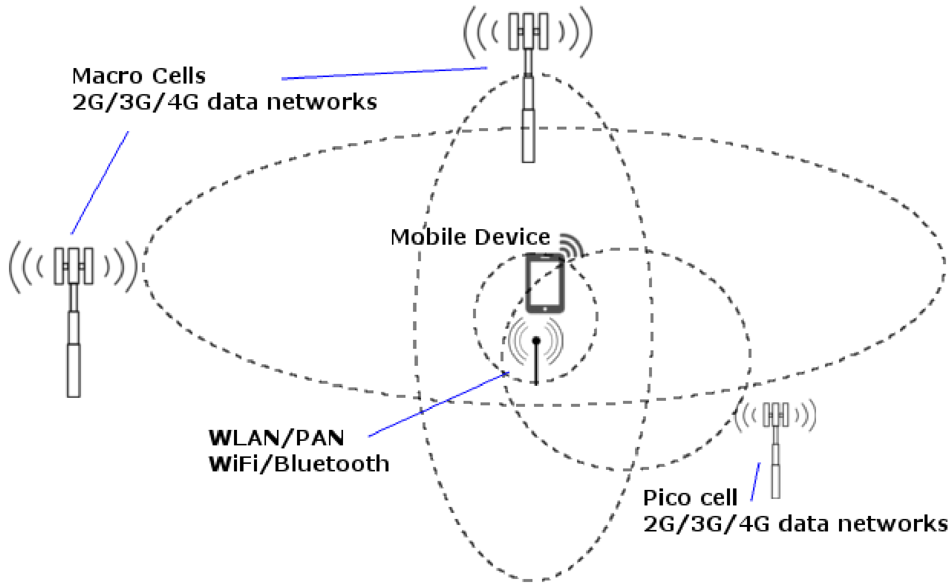

Figure 1.1: An example of a heterogeneous wireless network

of network interfaces. An Android device will only change active interfaces if a RAT with a higher priority becomes available. If the signal quality or performance of the highest-priority RAT begins to degrade, an Android device will not switch RATs until all connectivity is lost [3]. In other words, the vertical handover decision (or interface management) scheme in modern mobile devices tends to be reactive rather than proactive.

From a user's perspective, mobility between RATs in a heterogeneous wireless network without loss of connectivity is highly desirable. Mobile devices should be able to maintain network layer connectivity in spite of vertical handovers within a hetnet. Achieving seamless IP mobility requires the network to intelligently route packets to a mobile device, regardless of location. This problem can be solved using different methods, using different technologies such as IPv6 [26] or SDN [20]. These solutions also have varying levels of assumptions; some assume that the mobile device will be roaming between RATs in a single network, others provide global IP mobility. Robust global IP mobility with low handover delays is crucial to the feasibility of active network management from a user experience (UX) perspective. IP mobility is not a focus of this thesis, but is essential to the viability of active network interface management within a hetnet.

1.2 Research Motivations and Direction

Network selection in modern mobile devices relies on some key assumptions about the nature of each available RAT. Each RAT is typically assigned a priority within the OS. In Android, 802.11 WiFi networks are assigned the highest priority. Cellular RATs are then given priorities based on the network generation associated with them. This scheme makes the assumption that the networks with higher priorities are always more spectrally efficient than networks with lower priorities [3]. The dynamic nature of network traffic and coverage in real-world hetnets frequently breaks this assumption. A congested RAT can result in a higher cost (both monetary and power) and lower achievable throughput than a lower priority RAT with low utilization. The result of these 'selfish' priority-based local network selection schemes is non Pareto-optimal bandwidth allocation among mobile devices; there exists another feasible allocation in which at least one user gets more bandwidth, and all others get at least the same bandwidth [2].

This suboptimal usage of hetnet resources by mobile devices presents network operators and administrators with a resource allocation problem. Finding a solution to this problem not only benefits end users, who see the benefits of fair resource allocation and higher average achieved throughput, but also the network operators who want to maximize network utilization due to spectrum scarcity. The infrastructure-centric nature of this problem has led to most real-world research on the topic taking place within the cellular network industry.

Frameworks to support network-initiated handover decisions have been studied as ways to globally manage network resources [15] [23]. Network selection algorithms have been studied typically from an optimization perspective [2] [35] [5] [36]. Most of these studies, however, are conducted from the perspective of network operators. A cloud-based over-the-top service running on the public internet independent of all network operators can collect empirical data about any deployed wireless network, public or private, and use that data to provide mobile devices with vertical handover decisions that consider every available RAT. This thesis documents the design and implementation of such a service and demonstrates how a cloud-based vertical handover decision be made using crowd-sourced geospatial network performance data.

1.3 Research Objectives

The main objectives and contributions of this thesis include:

- The design and implementation of a mobile network performance testing tool
- The design and implementation of a publicly available over-the-top vertical handover decision service
- The design and implementation of a basic vertical handover decision algorithm to be used as a proof of concept by the service
- A demonstration of the functionality of the service through the use of test scenarios
- The creation of groundwork for future research efforts focusing on the real-world analysis of cloud-based vertical handover decisions

1.4 Thesis Outline

The remainder of this thesis is organized as follows. In chapter 2, background discussion is provided that covers all relevant previous work that contributed to the foundation of this thesis. Chapter 3 details the conceptual design of both the network testing tool and the handover decision service. The implementation details of these systems are further expanded upon in chapter 4. Chapter 5 outlines the design of the test scenarios as well as the results gathered. The conclusions of this thesis as well as future work to be carried out on the systems are identified in chapter 6.

Chapter 2

Background

One of the major driving motivators for the work presented in this thesis is the concept of being Always Best Connected (ABC). Gustafsson and Johnson introduced this term in a discussion about the advent of cellular data networks [17]. ABC means a person should not only always be connected, they should be connected through the best available device and access technology at all times. 'Best' is a subjective term that can be defined in many ways, but the core idea behind ABC is applicable to all persons acting within heterogeneous wireless networks. User experience is a major motivator behind the ABC concept. It is noted that for an ABC service to provide a strong UX, seamless information delivery and mobility support should be present. Applications should be able to adapt to network changes in such a way that is transparent to the user, and a mobile device should ideally be able to roam seamlessly within a hetnet without loss of data.

The functional components of ABC services were identified by Gustaffson and Johnson as follows:

- Access discovery
- Access selection
- Authentication/authorization/accounting (AAA) support
- Mobility management
- Profile handling
- Content adaption

AAA support, profile handling, and content adaption are important components of ABC services, but are not a focus of this thesis. Access discovery is the term used to describe the discovery of access networks available to a mobile device. This is a constant process that occurs locally on a mobile device even after network selection occurs. There are two key issues related to access discovery that must be solved by an ABC solution:

- Each access network must be able to be described in generic terms such that they can be easily compared. Terms of interest include network type, network operator, QoS, network cost, and type of connectivity.

- The collection of network statistics by mobile devices must take place. These statistics should be used to inform a network selection decision.

These issues are vital to the practicality of access selection techniques. Generic statistics and network characteristics can be used to facilitate access selection.

Access selection refers to the access network selection process. There are three main ways in which this selection can be made:

- Terminal-based (local)

- Network-based

- Through user intervention

An access selection solution must consider a number of different aspects. Firstly, access selection should consider user preferences, which can affect desirability for access networks based on cost or power consumption. Likewise, there may be preferences from the perspective of the ABC service provider based on cost or current network conditions. Network characteristics (bandwidth, latency, operator, etc.), device capabilities, and application requirements should also be a factor in access selection. The nature of heterogeneous wireless networks necessitates the mobile device to be capable of "offline" network selection. This can be accomplished through use of a stored profile, priority list, or default setting. This provides the device with a network selection during initial boot, or when experiencing connection loss.

Network-based selection methods have many advantages over local selections. Network-based selection allows ABC providers to make use of network-specific information that may not be readily available to individual mobile devices. ABC providers can also use network-based selection to act as a form of load balancing; providers can perform radio-resource-efficient selection, maximizing total system throughput.

Mobility management refers to the processes behind seamless information delivery and mobility support. There are three pieces of functionality that mobility management adds to an ABC service:

- User reachability
- Session transfer
- Session continuity

User reachability is the ability to reach an ABC user at his or her current access network and device. Session transfer is the maintaining of session as a user moves between different devices. Session continuity refers to the maintaining of a session throughout a mobile device's migration between different access networks and technologies [17]. Achieving session continuity is one of the biggest obstacles that exists in heterogeneous wireless networks.

2.1 Mobility Management

Mobility management is important to a good user experience within a heterogeneous wireless network. Session continuity can be achieved through different techniques with varying level of assumptions. If a user is migrating between basestations within a single access technology (horizontal handovers), session continuity can be maintained through use of link layer signaling protocols. If the hetnet also includes other access technologies that are still part of a single core network, network-based IP mobility can be used without participation from the mobile device. However, if a mobile device wishes to maintain session continuity in a truly heterogenous network, the mobile device must be an active participant in the mobility management protocol.

2.1.1 Mobile IP

The IEEE standardized technology for maintaining network-layer session continuity and user reachability is known as Mobile IP. The latest revision of the Mobile IP protocol has been

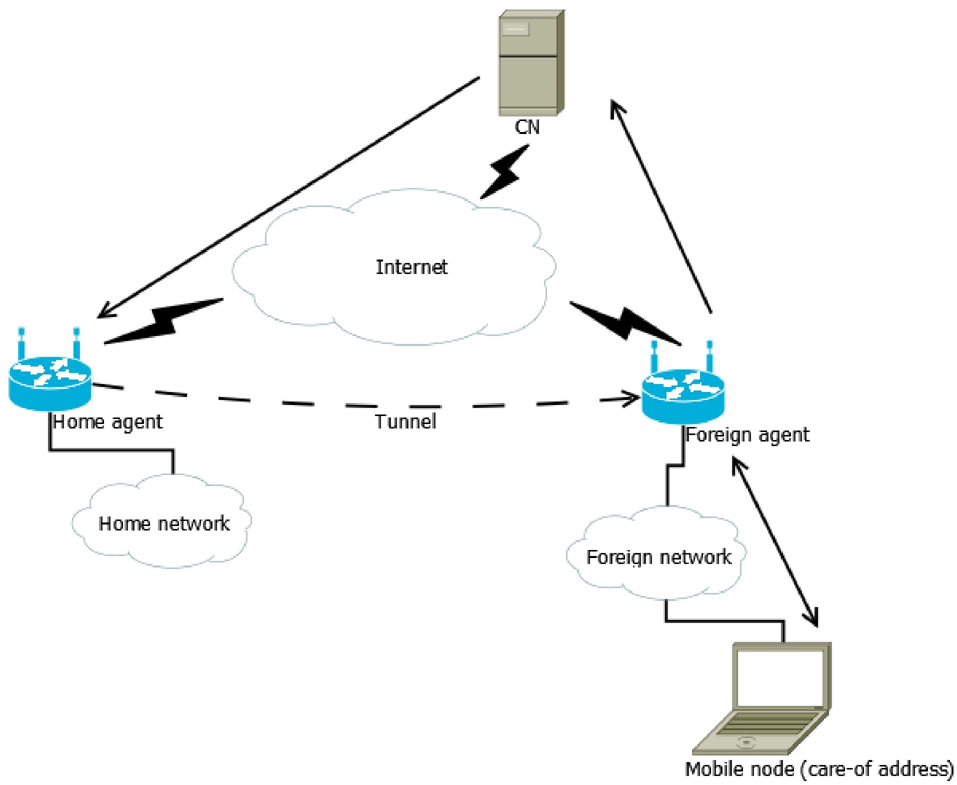

Figure 2.1: A Mobile IP scenario without route optimization

implemented as an extension to the IPv6 protocol (MIPv6), taking advantage of some of the new features of its updated addressing scheme. As illustrated in figure 2.1.1, a mobile device has two IP addresses: a home address and a care-of address. The protocol is designed such that packets sent to its home address are transparently delivered to its care-of address. Through the use of MIPv6, a client retains the ability to use a single IP address regardless of its active access network. Early revisions of Mobile IP accomplished this through use of a tunnel, requiring additional hops and processing overhead for each packet [25]. The use of IPv6 allows for route optimization to occur through use of binding updates; a mobile device can directly give its correspondent node (CN) its care-of address, eliminating the need for a tunnel [26].

While MIPv6 is standardized, it is not yet widely deployed. Open-source implementations of MIPv6 exist, UMIP [1] being the most prevalent. A performance study investigating the performance of UMIP shows that mobile devices still experience handover latency ranging from 1-5 seconds in length [6]. This level of service disruption is unacceptable for ABC services, as it violates the principle of session continuity. Extensions to MIPv6 exist to speed up the handover process [31][21], but are not implemented in UMIP. Another issue with MIPv6 is the reliance on IPv6 features. While IPv6 adoption is growing, Google reports that as of June 2014, only around 3.5% of their users have IPv6 availability [13]. Even if all mobile devices devices currently used IPv6, the advantage of MIPv6 route optimization would not be realized unless all CNs also followed the MIPv6 protocol. In other words, MIPv6 is a long way away from becoming a reality.

There are other IP mobility schemes that attempt to solve some of these issues. Most of these protocols are network-based, requiring minimal participation from mobile devices. Proxy Mobile IPv6 (PMIPv6) is one such protocol that has been standardized in RFC5213 [16]. PMIPv6 enables the network to track a mobile device's movements within a network and update routing accordingly. A Local Mobility Anchor (LMA) resides at the root gateway of the network, and communicates with Mobility Access Gateways along the edge of the network. A similar approach is taken in [20], leveraging software designed networking (SDN) features to create a network-based IPv4-compatible IP mobility solution. An understanding of IP mobility issues is crucial to this thesis, but is not a focus of the presented work.

2.1.2 Active Interface Management

The other aspect of mobility management is providing applications with mechanisms to handle changes in connectivity and react to handovers. Ideally, these handovers should be completely transparent to applications. The nature of hetnets usually requires mobile devices to make use of multiple radios abstracted to the OS as network interfaces. Current Linux kernel functionality is lacking in support for vertical handovers. Switching active network interfaces results in a socket error, and there is no standardized way to retrieve network state information from cellular radios. More complex interface management schemes have been proposed to facilitate intelligent and seamless vertical handovers. 802.21 [15] is an IEEE standard that was established in 2008 to support seamless vertical handovers, but has not seen widespread adoption by cellular operators. There is an open source implementation of 802.21, ODTONE [32], but it was not used for the purposes of this thesis.

The 3rd Generation Partnership Project (3GPP) and Open Mobile Aliance (OMA) have developed mobility management standards independent of 802.21. Their solution, part of the 3GPP standard and new evolved packet core (EPC), is called OMA Device Management (OMA-DM). OMA-DM provides network operators with the ability to remotely manage a wide variety of device settings, including radios. The specific function that is used to trigger vertical handovers is called access network discovery and selection function (ANDSF), which allows network operators to have fine grain control over the network selection process in their participating mobile devices [23].

The focus of this thesis is providing a vertical handover decision mechanism, not low-delay or seamless handovers. While mobility management is key to UX in ABC services, the simple Python-based interface management scheme outlined in this thesis was sufficient for demonstrating the effectiveness of this work.

2.2 Network Selection

Access network selection within hetnets occurs in what is commonly referred to as a vertical handover decision. Vertical handover decision (VHD) algorithms are designed to provide mobile devices with ABC connectivity. VHD algorithms can consider a wide range of criteria depending on the assumptions made by the system and the available data. Some examples of network decision criteria for VHD algorithms include:

- Received signal strength (RSSI)
- Network connection time
- Available bandwidth
- Power consumption
- Monetary cost
- Security
- User preferences

RSSI is one of the most commonly used criteria in VHD algorithms due to the relative ease by which RSSI data can be sampled. RSSI is also a good indication of signal quality. Horizontal handovers often use RSSI as a main decision criterion, but when considering heterogeneous networks RSSI isn't always a strong indicator of network quality. Acceptable RSSI levels vary between network technologies, and even if RSSI is normalized by technology there still exists the problem of differing network characteristics. Hetnets necessitate the consideration of other decision criteria.

Network connection time is the amount of time that a mobile device maintains a connection. This measurement is important for the proper timing of handover decisions, especially in mobile devices. Also, setting a minimum network connection time reduces superfluous handovers and network ”thrashing” that a mobile device may experience while roaming.

Available bandwidth can be interpreted as either link capacity or current achievable throughput. Channel capacities of access networks are generally static for a mobile device in a given location, but achievable throughput varies based on network load. Channel capacity is easier to obtain and can be used as a criteria for VHD algorithms, but will not always provide accurate results. Achievable throughput is more difficult to obtain and usually requires active bandwidth measurements to be taken by the mobile device. Network operators may have passive throughput data for some networks, but will not always have access to such data for every RAT in a hetnet.

Power consumption, monetary cost, security, and user preferences are all criteria that place value in the resources available to a given device. A mobile device may have radios with different levels of power consumption. If a device is low on battery, the VHD algorithm should react by switching the mobile device to a more power efficient radio. Achievable throughput also plays a role in power consumption; a radio with high relative achievable throughput could conceivably be more power efficient even with a higher level of power consumption. Monetary cost refers to the cost incurred by the use of each radio as per access network service plan. Security can have an impact on the VHD; if the user is making use of an application that transmits confidential data, the VHD should avoid switching the mobile device to an unsecure access network. Aside from these user preferences, users may have other reasoning behind preferring one type of network to others, which should be accommodated in VHD algorithms [36].

VHD algorithms analyze these criteria through many different methodologies to arrive at a decision. Some examples of techniques that have been applied to this problem include utility theory, multiple attribute decision making (MADM), fuzzy logic, game theory, combinatorial optimization, and Markov chains [33]. In [35], a VHD algorithm featuring coalition formation games is used in conjunction with a cloud-based distributed database containing network performance data. Xu reasons that game theory is more suitable for scenarios in which network resources are a constraint. The use of a cloud-based network performance data repository allows this algorithm to use historical network usage patterns to improve the accuracy of the VHD.

2.3 Network Scheduling

Resource allocation among users in any shared network medium is challenging to manage. Network schedulers handle the distribution of network resources among hosts. 3G CDMA networks make use of a proportional fair (PF) scheduling at each basestation to schedule downlink traffic among users [4]. PF allocates resources to users based on channel condition and asserts fairness by weighting the achievable throughput by a user by the average throughput received by that user. This approach works well for allocating resources within a single basestation, but does not consider the resources made available by all other basestations in a deployment. Network selection in mobile devices is based on local RAT priority and availability, creating load imbalances throughout the network. Network schedulers in wireless network deployments should consider the network as a whole to avoid non-Pareto optimal bandwidth allocation [5].

This problem is compounded when dealing with heterogeneous wireless networks. Not only do heterogeneous networks typically consist of networks with a wide range of characteristics, they can also operated by different organizations entirely. This necessitates that at a macro level in a hetnet, resource allocation must be performed by an OTT handover decision service. This service must have knowledge of the state and characteristics of each network within the hetnet. Mobile devices roaming within this hetnet defer their network selection decision to this service, which makes an informed decision based on criteria such as network load. Such a service, if deployed universally, could impose a coarse resource scheduling policy among all users within a hetnet.

2.4 Mobile Data Offloading

Hotspot 2.0 is an effort that has arisen to help offload traffic from cellular data networks. This effort has been organized by the Wireless Broadband Alliance, a collaboration between device manufacturers, network equipment vendors, and network operators. Hotspot 2.0 is designed with the goal of providing widespread 802.11 Wi-Fi access to mobile devices while maintaining the security and accessibility that users experience in their network operator's own network. Hotspot 2.0 enabled mobile devices will authenticate with Hotspot 2.0 enabled routers deployed on any network and the device will be able to act as it would on its home network [34]. The proliferation of Hotspot 2.0 availability will only further compound the necessity for an over-the-top vertical handover decision service by adding new networks to a mobile device's available hetnet.

2.5 Network Coverage Mapping

It is a common for an internet user to want to assess their current achievable throughput. There are many free online tools available for benchmarking network performance. Examples of such tools are Speedtest.net [24] and DSLReports [9]. These tools aim to provide users with a neutral evaluation of their current network connection, allowing them to investigate any discrepancies with their service plan's advertised rate. As mobile data networks become increasingly popular, similar apps have been developed to assess mobile network performance. Mobile network operators often use claims about network speeds and coverage in their advertisements, so it is natural for users to want to independently confirm those claims. Mobile devices have the added benefit of having integrated GPS, allowing network test results to be associated with a precise location. Examples of apps that provide users with this assessment include Speedtest.net, FCC Speed Test, and CoverageMap. The Speedtest.net app is a simple network testing app that collects latency, and upstream/downstream throughput data.

As part of the FCC Measuring Broadband America program, the FCC released their own open-source speed test app in late 2013. This app was designed to be an effort in crowd-sourcing, or collecting data from a large amount of individual users, a public database containing mobile broadband performance across the US. The FCC Measuring Broadband America program has a stated purpose of "improving the availability of information for consumers about their broadband service" [7]. While the FCC does release condensed data reports, no reports have been released with mobile broadband findings. Open access to the collected data is not yet provided by the FCC.

Rootmetrics is a company with the goal of providing comprehensive, unbiased information about real-world network performance. The Rootmetrics testing app, CoverageMap, is installed on a user's device. Similar to the FCC app, the CoverageMap app is used to crowdsource network performance data. Rootmetrics then processes this data and presents it to users on a public web-based visualization map as shown in 2.5. Consumers can view this map to see how the coverage and performance of their network operators stacks up against competitors [29]. Some network operators such as T-Mobile embrace the openness and availability of public data and have integrated Rootmetrics coverage maps in their own websites. While Rootmetrics provides valuable coverage data in an easily digestible visualization map, their raw test result data is not made publicly available.

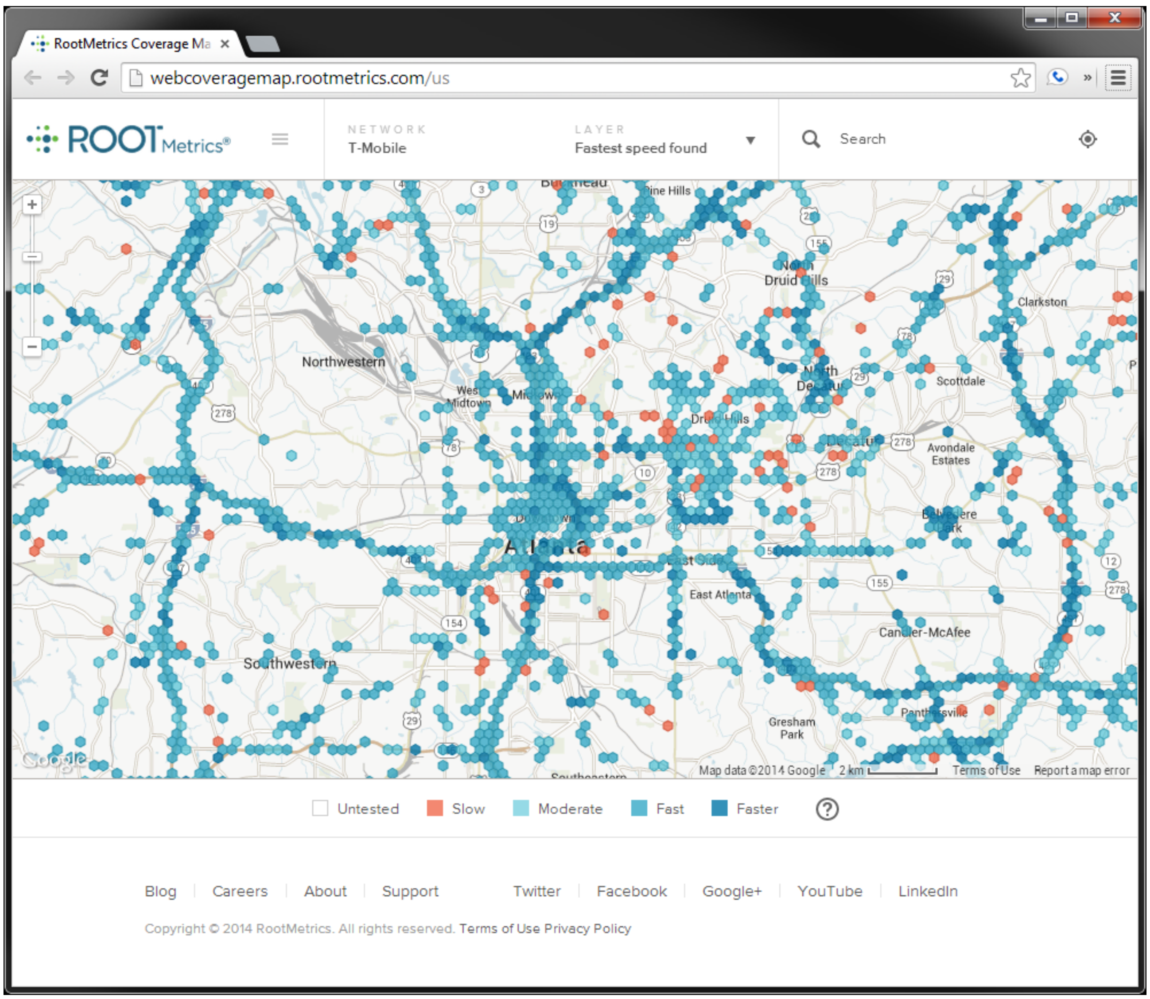

Figure 2.2: The Rootmetrics visualization map

2.6 Open Problems

Much of the research presented on vertical handover decisions has been conducted through use of network simulators, partly due to the high cost of implementation of mobile data network infrastructure. The academic community is lagging behind the industry in terms of prototypes and testbeds evaluating the network selection problem. An over-the-top handover decision service would allow researchers to evaluate their handover decision strategies in a real-world environment using existing network infrastructure.

The current deployment direction of mobility management in hetnets is highly cellular oriented, as it is driven by the industry. Providing mobility management services from the perspective of an 'internet' model can be more widely adopted by the general public, hence the focus on an over-the-top solution.

Chapter 3

System Design

The vertical handover decision service has been designed in a way that maintains a clear separation of duties between each component of the system. This simplifies the codebase, improving maintainability. The result is multiple concurrent processes running on both the client and server. To meet the goal of making informed network selection decisions, data collection must take place. This necessitates the design of a network testing component that continuously gathers data from mobile hosts. This data can then be used by a separate handover decision process, facilitating clients with the selection of the optimal wireless network. Separating the roles of each component in this manner also allows devices not making use of the handover decision service to contribute network data to the database via the independent network testing tool. Figure 3 shows an example of a heterogeneous network taking advantage of such a service. Devices are equipped with a network test client, local resource controller, GPS, and multiple network interfaces. The local resource controller requests handover decisions from a global resource controller, and uses that decision to select the device's active interface. Meanwhile the network test client seeds the performance testing data used by the GRC.

3.1 Network Test Tool

The CyberTiger Network Test Tool has been developed as a cross-platform network performance analysis framework. It consists of a client-server architecture in which tests are negotiated via TCP and carried out over TCP/UDP. The framework has been designed to be extensible in

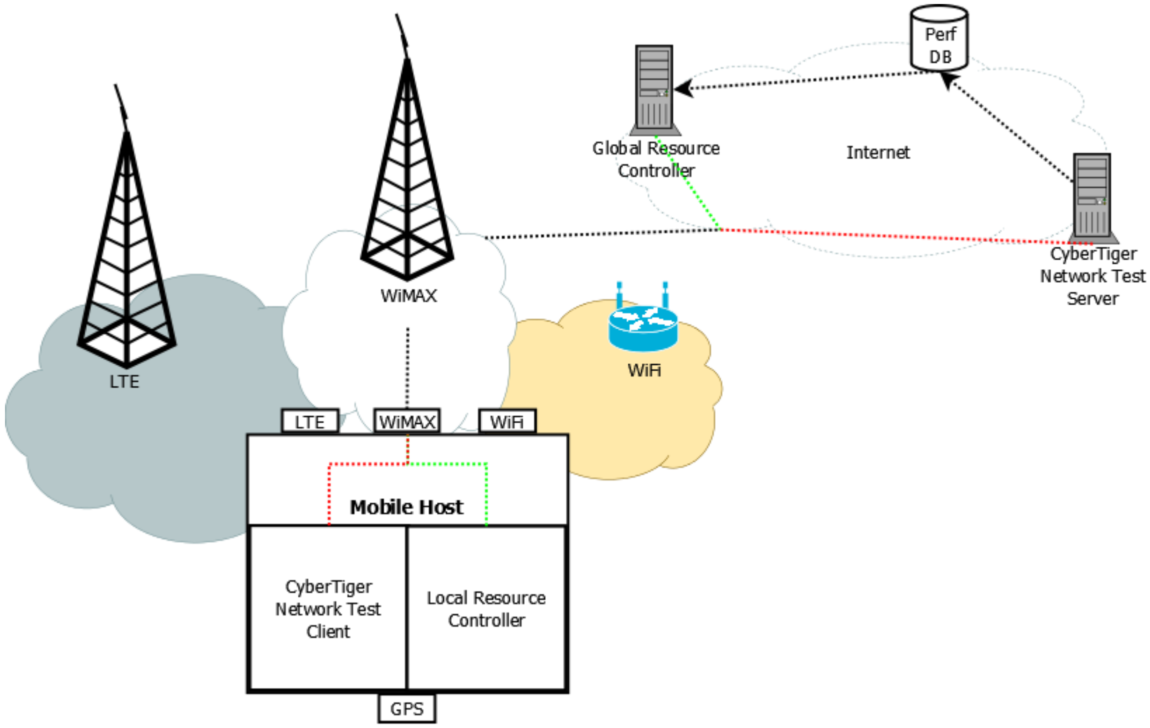

Figure 3.1: High-level system design

that new TCP/UDP-based network evaluation metrics can be added while maintaining backwards compatibility and without the need for a redesign. Clients can be anything that can communicate using TCP/UDP sockets, from a Linux laptop to an Android smartphone. The only other requirements of the client is that it has geolocation capabilities and access to radio state/RSSI data. Test results are stored in a database and then made publicly available on a web-based visualization map. This framework facilitates the geospatial mapping of wireless network performance using a variety of metrics such as throughput, RSSI levels, and latency. In this system, the CyberTiger Test Tool is run in the background, executing a latency test approximately every 5-10 seconds. This ensures that the database has fresh data for the vertical handover decision algorithm to work with.

3.2 Network Visualization

The visualization of the collected data by the CyberTiger Test Tool is designed to provide a publicly-accessible live map of network performance measurements. Each data point is visualized on the map and color-coded based on one of the metrics gathered by the network test tool. The

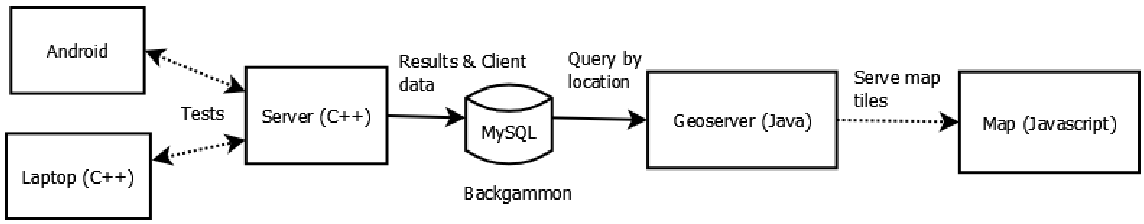

Figure 3.2: CyberTiger components

map also provides filters that can be used to specify the data to be visualized. This map can be used to assess network coverage and make comparisons between different providers. A visual-spatial representation of various network qualities can prove to be invaluable when trying to choose an optimal location from which to utilize wireless resources. In this project, the visualization map was used to identify a location in which all access networks had strong coverage, ensuring continuous network availability throughout the duration of the test scenarios.

3.3 Handover Service

The vertical handover decision service is designed to be over-the-top, in that it exists on the internet independent of network providers and the active RAT. From the perspective of network operators, it is a 3rd party service. This service will utilize the data collected by the CyberTiger Test Tool to make an intelligent decision on which RAT a client should use in a given area. The over-the-top nature of the service allows it to act as a central Global Resource Controller (GRC) for all participating access networks.

The handover decision algorithm in this thesis will consider the relative load (congestion) of each network as well as network availability (coverage). The relative load of the network can be estimated by analyzing the latency observed by a client at a location. A mobile host connected to a network experiencing congestion or load will report abnormally high latency values. The measurement of load can be maintained at the server for each network as a moving average of z-score normalized latency. Mobile hosts can be assigned to networks that are experiencing lower than average load. This serves a dual-purpose of both providing clients with connectivity to a minimally congested network and providing a simple load-balancing scheme for hetnets.

Availability will be calculated using a z-score normalization of RSSI for each network. The algorithm will make use of recent RSSI measurements gathered in an area to determine the current signal strength for each network available to a mobile host at a location. Once the service has an estimate of network coverage and load for all the RATs of a mobile host, it can make an intelligent decision on the optimal network for the client to use. This decision is a function of the estimated load and estimated availability. The calculation will be a simple score computed by taking the difference of availability and load metrics. The resulting decision based on this score will provide mobile hosts with a network decision that maintains high relative availability and low relative load.

```
input  : latlng: a location in latitude/longitude
         networkTypes: a list of available RATs
         t: width of time window in seconds
         b: width of latlng bounding box in degrees
output: bestNetwork: name of the best RAT for the mobile device's location

bestScore = -999;
bestNetwork = "";
foreach RAT in networkTypes do
    meanRTT = database_query(global mean RTT for RAT);
    stddevRTT = database_query(global stddev of RTT for RAT);
    localRTT = database_query(mean RTT for RAT in the last t minutes within b
    degrees of latlng);
    zRTT = (localRTT - meanRTT) / stddevRTT;
    meanRSSI = database_query(global mean RSSI for RAT);
    stddevRSSI = database_query(global stddev of RSSI for RAT);
    localRSSI = database_query(mean RSSI for RAT in the last t minutes within
    b degrees of latlng);
    zRSSI = (localRSSI - meanRSSI) / stddevRSSI;
    networkScore = zRSSI - zRTT;
    if networkScore > bestScore then
        bestScore = networkScore;
        bestNetwork = RAT;
    end
end
```

Algorithm 1: VHD algorithm pseudocode

An assumption that is made with this methodology is that the database is densely seeded with latency values for the location that the mobile device is requesting decisions from. This is necessary to establish baseline latency measurements. The z-score normalization of latency will allow us to calculate the deviation of the current network load from the average. Comparing RSSI between different RATs is not always straightforward. This approach assumes a linear relationship between

RSSI and perceived connection quality. Another assumption made is that the impact of RSSI on latency is negligible, though it is possible that low RSSI could result in link-layer retransmissions, increasing latency.

A design requirement of the handover service is that it lay a solid foundation for future work. The framework should be able to swap decision algorithms dynamically, enabling comparisons between different decision algorithms. The decision algorithm outlined above acts merely as a proof of concept for future implementations of cloud-based handover decision algorithms based on techniques such as game theory [35] or proportional fair bandwidth allocation. This is further discussed in section 6.1.

3.4 Client

The client component that queries the GRC and reacts to the handover decision is known as the Local Resource Controller (LRC). Each client will run both the CyberTiger Test Tool and LRC as a daemon in a background process. An assumption that is made with this solution is that the client has internet connectivity, either through at least one of its active RATs or via a control channel. This approach could be further improved by implementing a local offline handover decision algorithm as a failover. The LRC will query the GRC periodically and handle the active management of network interfaces on the devices.

3.5 Access Networks

The deployment of wireless infrastructure is an inherent obstacle for all researchers in the field. Oftentimes, it is necessary to interact with the network at a more intimate level than that provided by consumer-grade cellular data networks. These consume-grade cellular data networks do not make the details of their deployments publicly known, making it difficult for researchers to fully understand the topology of the infrastructure that they are using. Furthermore, the network utilization of these consumer networks fluctuates unpredictably over time, skewing results and making it difficult to design repeatable experiments. Consumer-grade cellular networks also generally enforce a per-device flat monthly rate. This pricing model is not suitable for a wireless researcher who may be using a variety of devices and may not be utilizing all of their monthly allotted bandwidth.

SciWiNet is a project that aims to address some of the issues that researchers face when studying wireless networks. SciWiNet is an Arterra-powered Mobile Virtual Network Operator (MVNO) that offers the widespread coverage of the Sprint access network with a researcher-oriented pricing model. Instead of a per-device monthly rate, SciWiNet allows researchers to purchase 'buckets' of data that can be consumed at any rate. This allows researchers to purchase only the data they will use, and have any number of devices drawing from that single bucket of 'data'. The SciWiNet data model is geared towards researchers whose average monthly bandwidth consumption is less predictable than a typical smartphone user's.

The test scenarios in this project make use SciWiNet 3G and private WiFi 802.11n networks. These networks have been extensively mapped using the CyberTiger coverage tool for the locations chosen for testing.

Chapter 4

Implementation

The discussion of the implementation of this service is divided logically in to two sections; the test data collection and visualization component and the decision engine component.

4.1 Test Framework

The CyberTiger Test framework was designed with the goal of cross-platform compatibility. For this reason, Google's Protocol Buffers (protobuf) library [?] was chosen for its message passing and serialization format. Protocol Buffers provides compilers to generate data access classes that make it trivial to parse and serialize messages in almost any programming language. The format also provides many other benefits over other message passing formats such as XML, such as speed and size improvements. This library greatly simplified the code for both the client and the server.

Network measurements are always initiated by the client. This addresses both privacy concerns and ensures proper NAT traversal. Before initiating a test, the client first must gather geolocation and RSSI information from an accurate source. Once that information is obtained, the client sends a ClientDetails protobuf message to the server via an established TCP control channel. The server parses that message, allocates a UDP port for use with tests by that client, and starts a thread that then follows the protocol for the requested test type. The server sends a ServerAck protobuf message to the client letting it know its assigned port and that the server is ready to begin the testing. From that point on, the client-server interaction is defined completely by the specific type of test that has been requested by the client. The issue of NAT traversal is further addressed by making use of UDP hole punching; the client always sends the first UDP packet.

The latency test protocol consists of three pings between the client and server. These are single byte TCP messages use TCP with the TCP_NODELAY flag enabled, which disables the use of Nagle's algorithm. While TCP behaves differently than a typical ICMP ping and introduces some additional overhead, this method provides the actual latency that would be observed by a TCP application over a network connection. Once the round trip time (RTT) is calculated by averaging the three pings, the results are stored in a TestResults protobuf message and sent from the server to the client. The server then also inserts the results in the MySQL test results database.

4.1.1 Database

The database that is used to store test results has been designed using the third normal form database normalization principle. This helps to reduce the amount of redundant data stored and improves data organization. The DBMS currently in use is MySQL. The latitude and longitude of test results are stored using both the 'double' MySQL data type as well as the MySQL spatial extensions point geometry data type. Storing the location as a point allows for the use of MySQL spatial extensions queries, providing the efficient querying of test results by location. Bounding box queries are used by both the visualization as well as the GRC to query for data within a region. MySQL views are used to efficiently denormalize the data into a flat table format for easy processing and analysis.

4.1.2 Visualization

Data is visualized by using a combination of web-based technologies to render a coverage map in a browser. GeoServer is an open-source program that has been built to process and share geospatial data. Part of the capabilities of GeoServer includes the ability to render and serve transparent map tiles, which can be used as an overlay on other tile-based maps. A GeoServer layer is configured to query the CyberTiger database, making use of the MySQL spatial extension points. These points are rendered as dots on a transparent tile, color coded by the relative values of a performance metric. The tiles are then served over HTTP to a Javascript-based map library called Leaflet. Leaflet handles the asynchronous loading and rendering of map tiles, and has the ability to load tiles in different layers. The GeoServer tiles have a transparent background and are

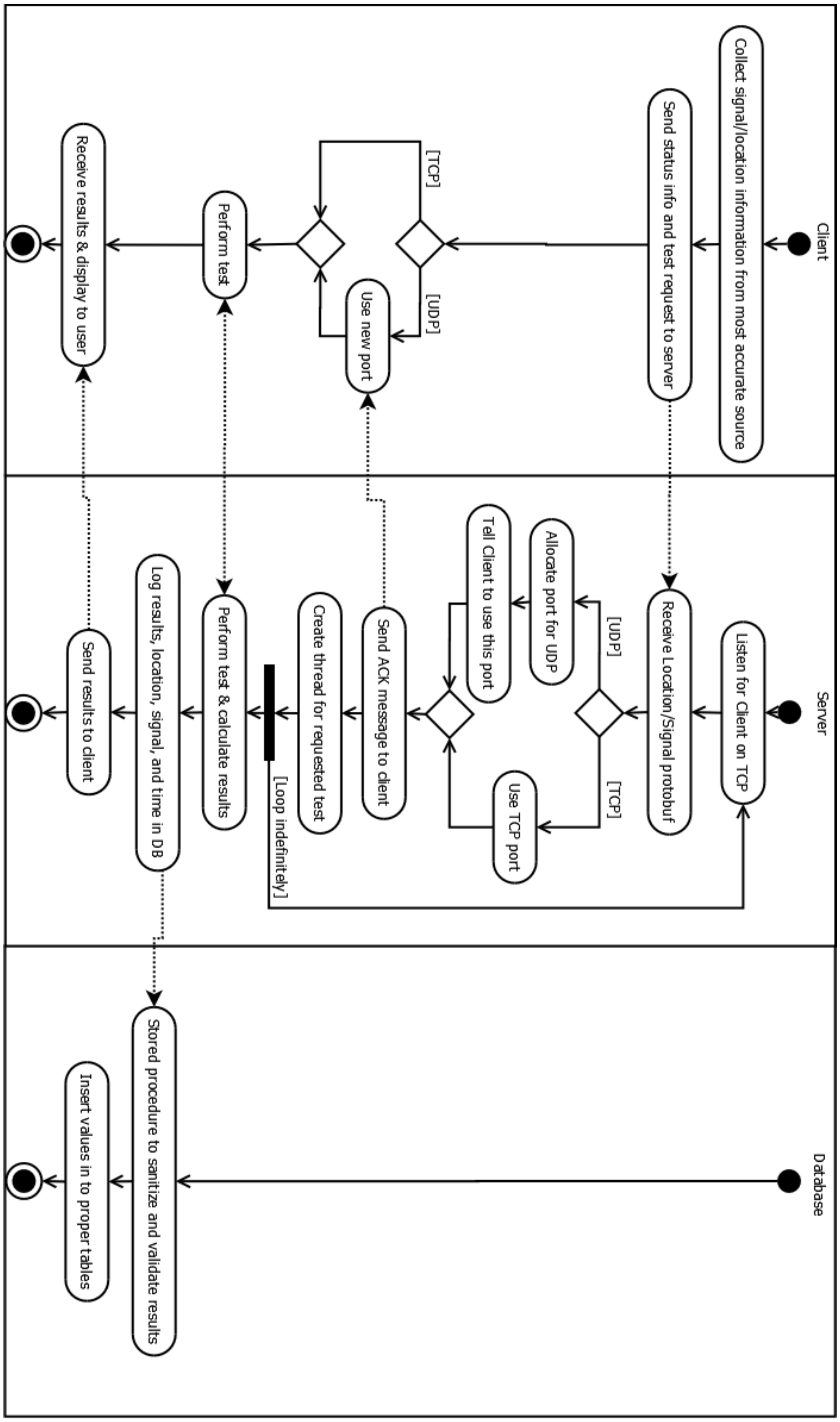

Figure 4.1: CyberTiger test protocol

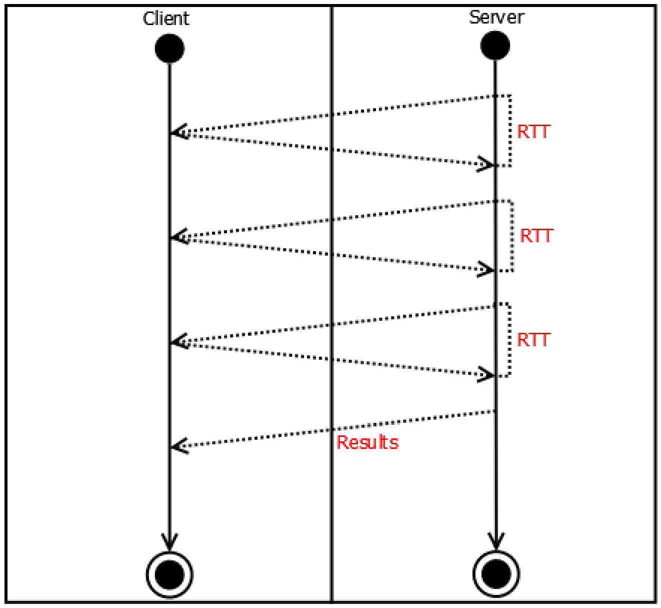

Figure 4.2: Ping test

rendered on top of street map tiles provided by MapQuest. The user is presented controls that, when interacted with, send different query parameters to GeoServer. GeoServer then renders and serves the corresponding tiles back to Leaflet. The user can also interact with the map itself by clicking on the dots, which sends an asynchronous command to GeoServer to return the raw data of all points associated with that location. This data is formatted in a table that is displayed to the user. The jQuery Javascript library is used throughout the client component of this visualization tool to reduce code complexity and ensure compatibility with all major web browsers.

4.1.3 Server

The CyberTiger Test Server is written in C++ and makes use of POSIX Threads. It runs on the server in the background as a system service, and uses a Cron job to ensure rapid recovery in the event of a crash. The server code itself is fairly standard; the server listens on an open port for TCP connections. When a client connects, the server knows to wait for a ClientDetails message to be sent. The server then parses that message to determine which test type has been requested by the client.

The server uses a class called PortManager to maintain a list of open ports. This class also inherently controls the maximum number of concurrent threads allowed in our program. The default

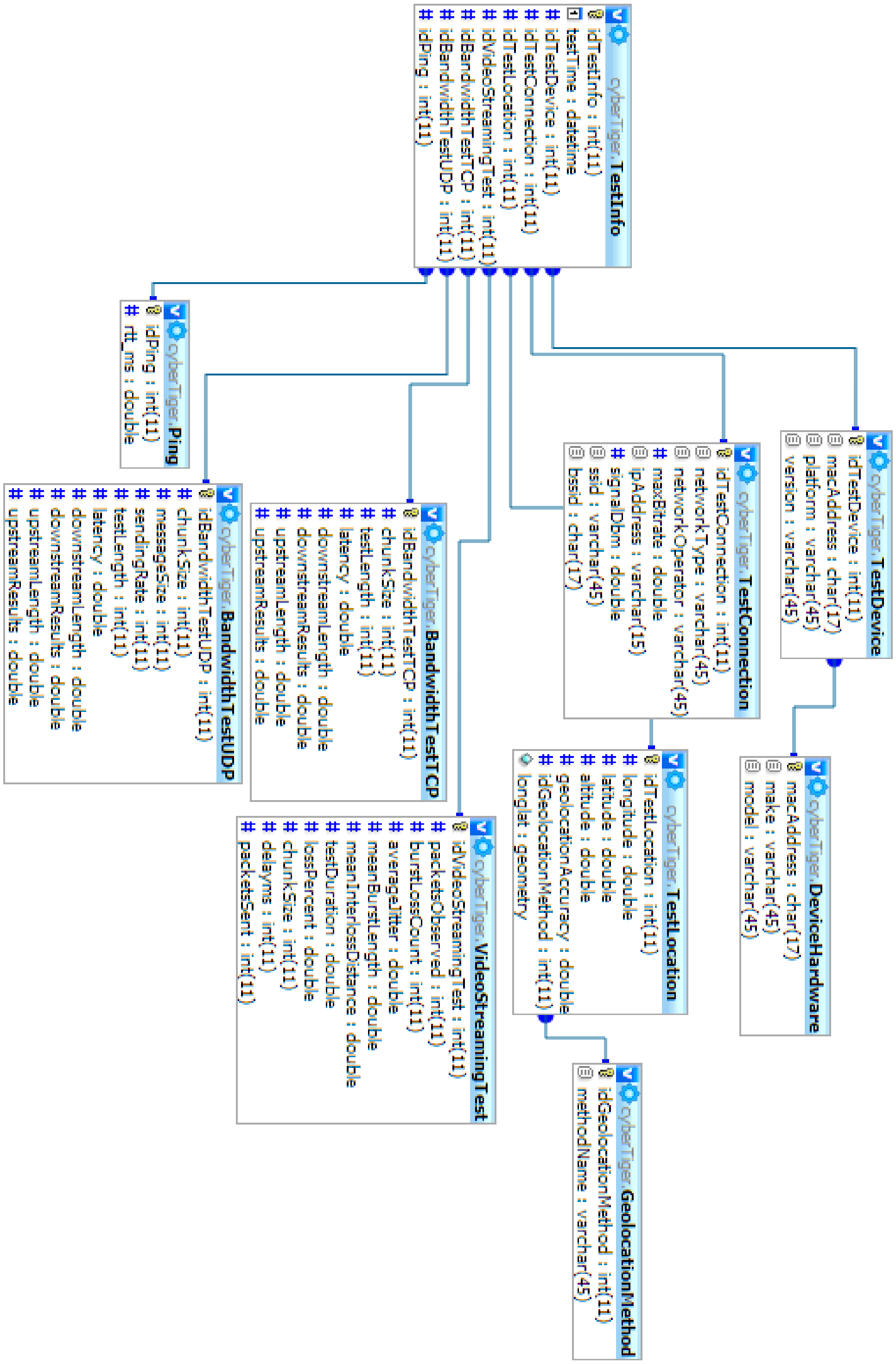

Figure 4.3: Database design

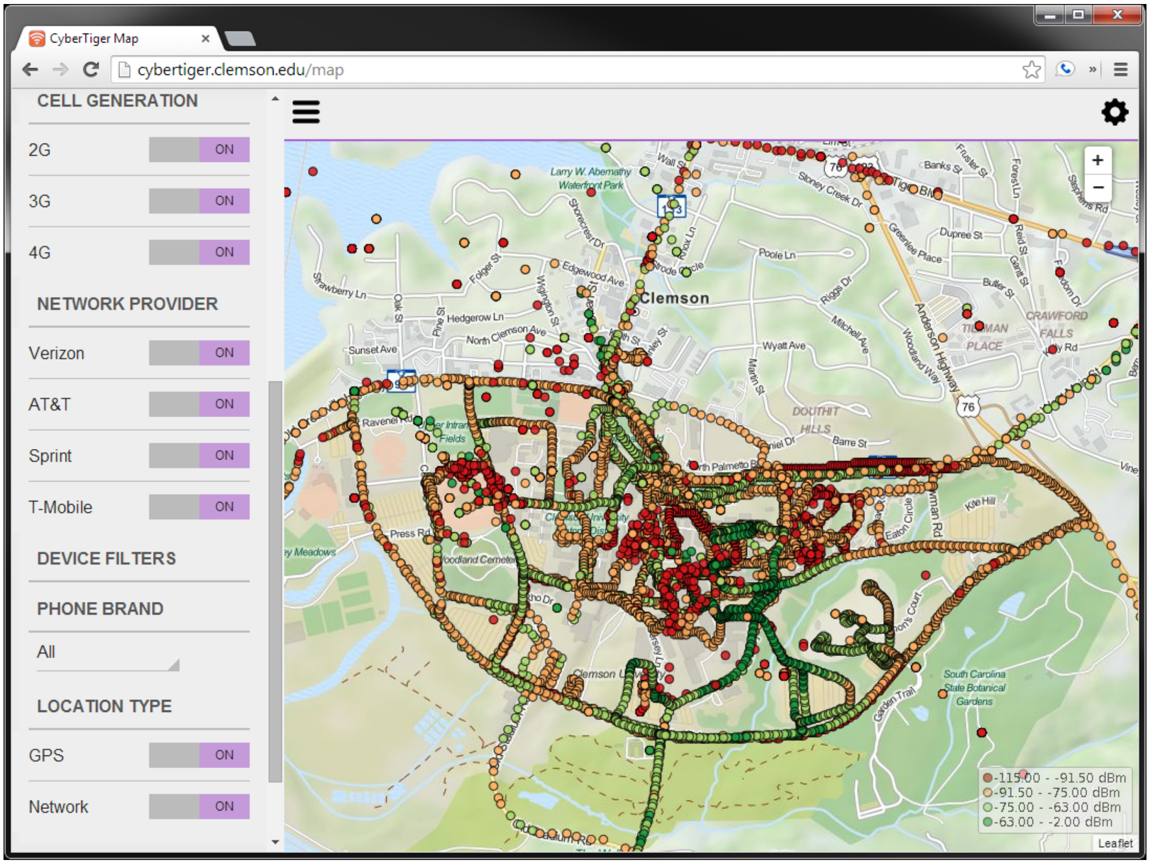

Figure 4.4: Visualization map UI

number of ports (and therefore also maximum concurrent threads) allocated is 10. The PortManager class keeps a list of port numbers and tracks their use throughout the life of the program. When a client is allocated a port from PortManager, that port is marked as used. When the client's test has been completed, that port is then released for use by future clients. Note that the allocated port may not actually be used by the client depending on the requesting test. The unique port is typically only useful for UDP-based tests, as the client can also communicate on the TCP socket that is spawned by the TCP accept() call.

Once the client has been allocated a port by the server, the client is notified by the server via a ServerAck message. Alternatively, if all ports are currently in use, the server responds with a "error: server busy" message, which the client must handle appropriately. After the ServerAck message is sent, the server code hits a switch() statement that spawns a thread using a routine that corresponds with the requested test type. Each of these routines are maintained in separate files for the sake of extensibility; to add an additional test type all it takes is a new routine and a case in the switch() statement. These routines are passed a testArgs struct that contains client details and test information, and are expected to use this information to log the test results in the database. These routines also typically should send a TestResults protobuf message to the client containing the results of the test.

4.1.4 Android Client

The design of the CyberTiger Android Test Client has iterated a few times, the latest being geared towards use by the SciWiNet MVNO project. One feature that SciWiNet would like to provide for its users is the ability to map out their own network coverage for their devices and regions. With that goal in mind, the SciWiNet Android app leverages Google's Fused Location Provider API and also maintains a local sqlite database of test results. The application itself consists of four main components that can be logically separated by virtue of the Model View Controller design pattern. The user interface code (View) is contained in a main Activity class, which consists of many Fragments. The Model is represented by the ContentProvider class that handles the local database interaction. The controller is the Service that runs in the background of the app, reacting to location updates and executing the actual CyberTiger tests. The result is a client that can execute CyberTiger tests in both the foreground and background contexts, recording both results and failed tests. The failed tests can later be inserted in to the database to record areas with no coverage.

4.1.5 Linux Client

The CyberTiger Linux Test Client is a relatively simple piece of software. A CLI is provided with options such as test type, iterations desired, and interface name. The client utilizes the GPSd C++ library to get a GPS fix, and then gathers WiFi state information via ioctl(). The gathering of connection state information from other devices such as WiMAX and LTE dongles is more difficult and relies on the manufacturer of the dongle to provide an API to access that information. For example, the Teltonika UM6225 WiMAX adapter provides a REST API that can be used to collect current RSSI. Modem AT commands can also be issued to the NETGEAR 341U LTE dongles to retrieve connection state information. Once that data is collected, the client uses a switch() similar to the one used in the server, but does not make use of multithreading. The code for each test is maintained in separate files, and each test should generally print a results message via stdout.

4.2 Decision Framework

The handover service was implemented as a web service, taking advantage of the robustness and ease of use of HTTP as a transport protocol. The two main components consist of a Python server and a Python client. The client handles the gathering of location data and interface switching, while the server makes the vertical handover decision. The handover service is deployed on a server at a public IP so that it can be made available over-the-top to users.

4.2.1 Web Service

The Python server makes use of the Flask web microframework. Flask handles all of the necessary HTTP server duties while remaining lightweight and simple to use [28]. The Flask server binds to a public port and listens for HTTP POST requests. Upon receiving an HTTP POST, the handler function parses the parameters and executes the VHD algorithm.

In this case, the VHD algorithm queries the database using MySQL spatial extensions to retrieve all points of interest within a bounding box. These data points are also filtered by time, as the decision algorithm only considers recent data within a certain window of time. After the relevant data is collected, the algorithm computes the z-scores for both RTT and RSSI. These two parameters are combined in equal weights to produce a network score. The name of the RAT with the highest network score is returned to users in the HTTP response.

4.2.2 Linux Client

The Python client polls the web service continually for handover decisions at a predefined interval. Before contacting the handover service, the client waits for fresh data from GPSd, the Linux GPS driver. Once it determines the current location, it issues the HTTP POST request to the handover service with the following parameters:

- Device ID (WiFi MAC)
- Latitude/longitude
- A list of currently available RATs
- The active RAT

Upon receiving a decision from the service, the mobile device updates its routing table such that the default route utilizes the newly active RAT. This implementation does not manage layer 2 connectivity, which is assumed to always exist or be managed by an external entity. A python module named dnet is also used to help manage the network state.

Chapter 5

Validation and Analysis

The laptop to act as a mobile device in the validation of the handover service is equipped with an Intel WiFi Link 5100 802.11a/b/g/n WiFi radio and a NETGEAR 341U Sprint LTE USB modem [22]. A GlobalSat BU-353 USB GPS mouse [18] is used in conjunction with GPSd to provide location data. To aid in data collection, two smartphones are used: a Samsung Galaxy Epic Touch phone, and a Google Nexus 5. To avoid horizontal handoffs during testing, both the 341U LTE dongle and the Google Nexus 5 are set to "3G only" mode. The laptop is running Ubuntu 3.10 with the network manager disabled. Links are established with the networks prior to testing, and are assumed to be present throughout the duration of testing.

There are parameters within the handover service as well as the VHD algorithm itself that have a direct impact on the system performance. The chosen values of these parameters are shown in table 5. Time window size represents the period of time prior to the present that the VHD considers. This window governs both the RTT and RSSI samples used in the decision making process. Bounding box size is the width and height of the bounding box used to query data for a given longitude and latitude. Handover decision frequency is the frequency at which a mobile device attempts to query the handover service for a decision.

5.1 Validation

Two scenarios are outlined to demonstrate the functionality of the handover decision service. These scenarios have been chosen to showcase two of the features provided by the currently imple-

Parameter	*Value*
Time window	1 minute
Bounding box	.0008 degrees
Decision frequency	10 seconds

Table 5.1: System parameters used in validation

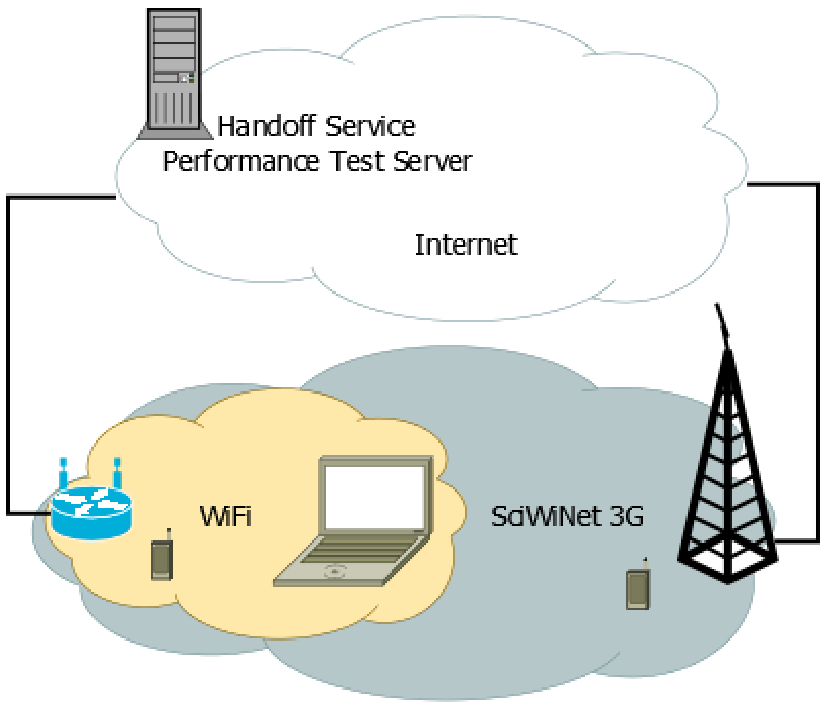

Figure 5.1: Network diagram for test scenarios

mented VHD algorithm: network selection based on availability, and network selection based on current network load. Throughput measurements are taken throughout the duration of these tests to provide a picture of how vertical handovers can affect achievable throughput. These measurements are taken using iperf, a popular TCP/UDP bandwidth measurement tool [19]. The achieved throughput on all interfaces can be monitored passively through the use of the bwm-ng, which also supports the collection of data in a CSV format [14]. RTT and signal strength data is gathered throughout the duration of the tests and stored in the CyberTiger database. Three trials are to be conducted in each test scenario. A diagram of the scenario design is shown in figure 5.1.

5.1.1 Test Scenario 1

In this scenario, a mobile device roams within a hetnet, receiving handover decisions made using local availability data. Two access networks will be used in this demonstration: WiFi and Sprint 3G. The WiFi access point to be used is mounted to a light pole next to a parking lot Clemson's campus. The mobile device will be driven in a vehicle circling the parking lot at an average speed of 5 mph. As the device roams to areas with lower levels of WiFi availability (signal strength), it will be switched over to make use of the SciWiNet 3G data connection provided by the NETGEAR dongle. Likewise, as the device roams back in to coverage it will be switched back to WiFi. A control trial will also be conducted, measuring WiFi and 3G throughput during the mobile scenario to further illustrate the throughput improvements over default network selection behavior. The proactive handover approach will demonstrate a departure from the default network selection behavior in the control.

The starting point for this test will be directly adjacent to the light pole. The network performance tool is used for all RATs at least one full minute prior to a test. This seeds the decision algorithm's time window with recent RTT and RSSI data. Once a minute has elapsed, the mobile device's processes (bwm-ng, iperf test, and VHD client) are started and the test begins. The mobile device is moved at a constant pace, circling the parking lot. Once the mobile device has completed one circle around the parking lot, it should have ideally undergone two vertical handovers between WiFi and 3G; one from WiFi to 3G as the device leaves WiFi coverage and one from 3G to WiFi as it re-enters WiFi coverage.

5.1.2 Test Scenario 2

This scenario involves a stationary mobile device that is connected to a highly congested WiFi network. The two networks to be used in this scenario are WiFi and Sprint 3G. The WiFi network is broadcast from a secured private access point. As in scenario 1, WiFi and 3G data will be collected for a full minute prior to each trial to seed the database with performance data. The mobile device will begin the test connected to the WiFi network, initiating its three processes (bwm-ng, iperf test, and VHD client). After 60 seconds have elapsed, the WiFi network is placed under an increased load. The load is created through use of netem [11]. Netem is configured to induce random delay in a normal distribution with a mean of 250ms and standard deviation of 50ms.

Netem will also induce a random packet loss rate of 5%. These conditions were chosen to simulate a highly congested WiFi environment. Once netem is enabled, the VHD algorithm is expected to react to this increase in network congestion by switching the mobile device to 3G.

This demonstrates a scenario in which the VHD algorithm is used as a load balancing mechanism. The user is moved to a less congested network, alleviating congestion on the old network while providing the user with a better experience. This technique could conceivably be used to help solve the resource allocation problem prompted by the recent trends towards mobile data offloading schemes.

5.2 Analysis

One of the issues encountered throughout the testing process was the resiliency of iperf in a mobile environment. During interface switching, the iperf client gave no indication of failure. The experimenter has to keep an eye on the handover decisions and manually restart iperf after the laptop switches its active interface. This could be avoided with the use of a seamless handover solution, or more intelligent application logic.

The tables in this section provide measurements collected by the VHD algorithm during the trials. The included measurements represent samples taken at the beginning of the trial, and just prior to the handoff occurring on the mobile device. RTT is the mean RTT that is both spatially and temporally local to the mobile device. This local RTT is then used to compute the RTT z-score (network load), which is a z-score normalization of the local RTT in respect to the historical RTT of that network. RSSI is the mean RSSI in dBm measured by the client that is both spatially and temporally local to the mobile device. Availability is a z-score of RSSI for the active network type. Network score is simply the combination of availability and RTT score, which is used to select the "best" network.

5.2.1 Scenario 1

The scenario 1 testing area can be seen in the coverage maps provided in figure 5.2. The position of the WiFi AP is indicated on the WiFi coverage and latency maps. Throughout the duration of the tests, the mobile device is within coverage of both WiFi and Sprint 3G. The driving speed was an average of 5mph throughout the test, and the vehicle traveled in a counter-clockwise direction around the perimeter of the parking lot. As a control, the scenario was first carried out without the use of the handover service. Ubuntu's default network manager was used. The device never used the WiFi connection due to the network manager's priority-based system of preferring ethernet links over wireless ones. The NETGEAR 341U that was used in this test is a hostless device, meaning the radio link state management is abstracted from the host OS behind a virtual ethernet link. This behavior illustrates one of the advantages of the use of a handover decision service.

In the testing for scenario 1, three laps were taken around the parking lot. This mobility lead to the handovers illustrated in figures 5.5, 5.7, and 5.9. As can be seen from these handover score graphs, Sprint 3G "won" the handover decision the majority of the time. The decision algorithm consistently switched the mobile device to WiFi when the vehicle was close to the AP. As the car went further away from the AP, the WiFi RTT increased and RSSI decreased. This was reflected by the results in trials 1, 2, and 3 and can be seen in tables 5.2.1, 5.2.1, and 5.2.1. In reference to the control throughput graph, figure 5.3, the handover service did not always switch over to 3G in areas with degraded WiFi connectivity, but did consistently switch over to 3G in the area with the longest stretch of intermittent WiFi connectivity, from roughly 50s-200s. This behavior resulted in a higher average throughput than what could have been achieved through the use of either WiFi or 3G alone. These results are exemplified by figures 5.6 and 5.8. The selection algorithm in Trial 1 switched 3G back to WiFi prematurely - this was a result of fluctuations in RSSI and RTT over 3G, suggesting that the time window of the decision algorithm should be increased to better accommodate these erratic network conditions. Overall, the results of this test scenario show that a mobile devices throughput within a hetnet can be improved with an over-the-top handover strategy. This test specifically illustrates how such a service lends itself very well to the concept of WiFi offloading.

5.2.2 Scenario 2

Trials 1, 2 and 3 of scenario 2 all exhibit signs of the VHD algorithm clearly responding to network congestion. As shown in table 5.2.2, trial 1 saw the handoff occur at approximately 112 seconds, reacting to congestion in 52 seconds. The handoff in trial 2 occurred at 88 seconds, a mere 28 seconds after the network became congested. The conditions of these handoffs are shown in table 5.2.1. As can be seen in the table, the handoff decision was accelerated by changes in 3G network quality; the 3G RTT dropped while RSSI rose. The conditions of the 3G network

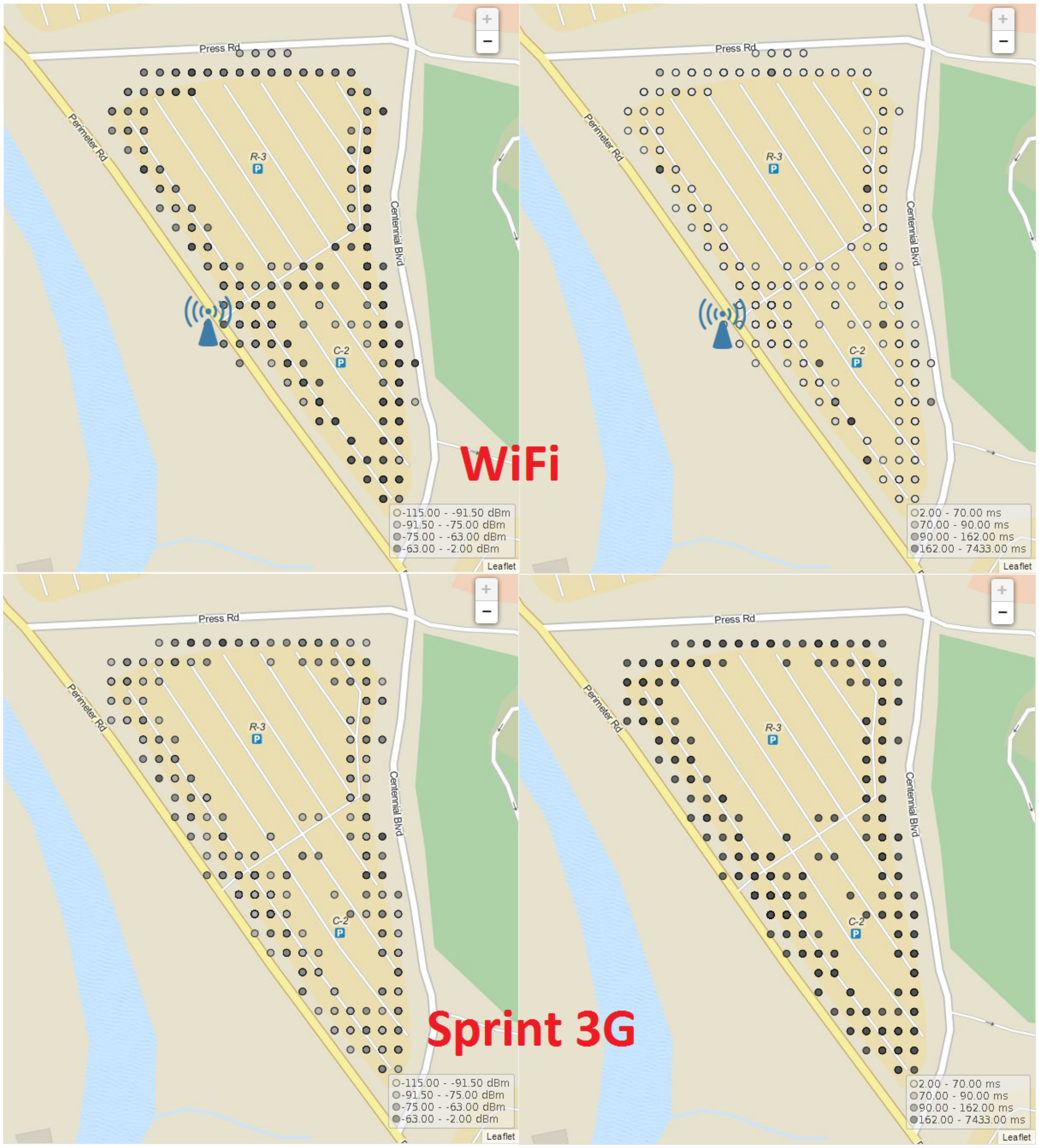

Figure 5.2: Performance data for WiFi and 3G - Left: coverage, right: latency

Handoff #	***Time (s)***	***RTT (ms)***	***RTT z-score***	***RSSI (dBm)***	***RSSI z-score***	***Network score***
1 (WiFi)	72.31	339.1	0.831218	-56.5	0.414473	-0.416745
1 (3G)	72.31	301.6667	0.361537	-75.0476	0.662368	0.300832
2 (WiFi)	173.52	11.32	-0.277351	-62.090909	-0.096465	0.180886
2 (3G)	173.52	314.3182	0.426316	-75.6818	0.597923	0.171607
3 (WiFi)	203.74	13.5	-0.269809	-64	-0.270967	-0.001158
3 (3G)	203.74	318.1111	0.4457	-74.5556	0.707089	0.261389
4 (WiFi)	216.79	31.11	-0.208575	-61.786145	-0.068843	0.139732
4 (3G)	216.79	307.0396	0.388914	-77.4455	0.423904	0.03499

Table 5.2: Scenario 1 trial 1 handoff measurements

Handoff #	***Time (s)***	***RTT (ms)***	***RTT z-score***	***RSSI (dBm)***	***RSSI z-score***	***Network score***
1 (WiFi)	47.24	326.62	0.784422	-58.923077	0.187823	-0.596599
1 (3G)	47.24	323.6719	0.46379	-78.8047	0.285186	-0.178604
2 (WiFi)	212.47	32.44	-0.199362	-60.457014	0.054225	0.253588
2 (3G)	212.47	315.5848	0.41891	-77.4777	0.413509	-0.005401

Table 5.3: Scenario 1 trial 2 handoff measurements

Handoff #	***Time (s)***	***RTT (ms)***	***RTT z-score***	***RSSI (dBm)***	***RSSI z-score***	***Network score***
1 (WiFi)	47.69	217.07	0.42847	-61.130435	0.009612	-0.418857
1 (3G)	47.69	345.7143	0.568191	-76.1964	0.537962	-0.030229
2 (WiFi)	208.63	26.56	-0.213143	-63.840206	-0.228976	-0.015834
2 (3G)	208.63	335.757	0.513468	-77.4472	0.414232	-0.099236

Table 5.4: Scenario 1 trial 3 handoff measurements

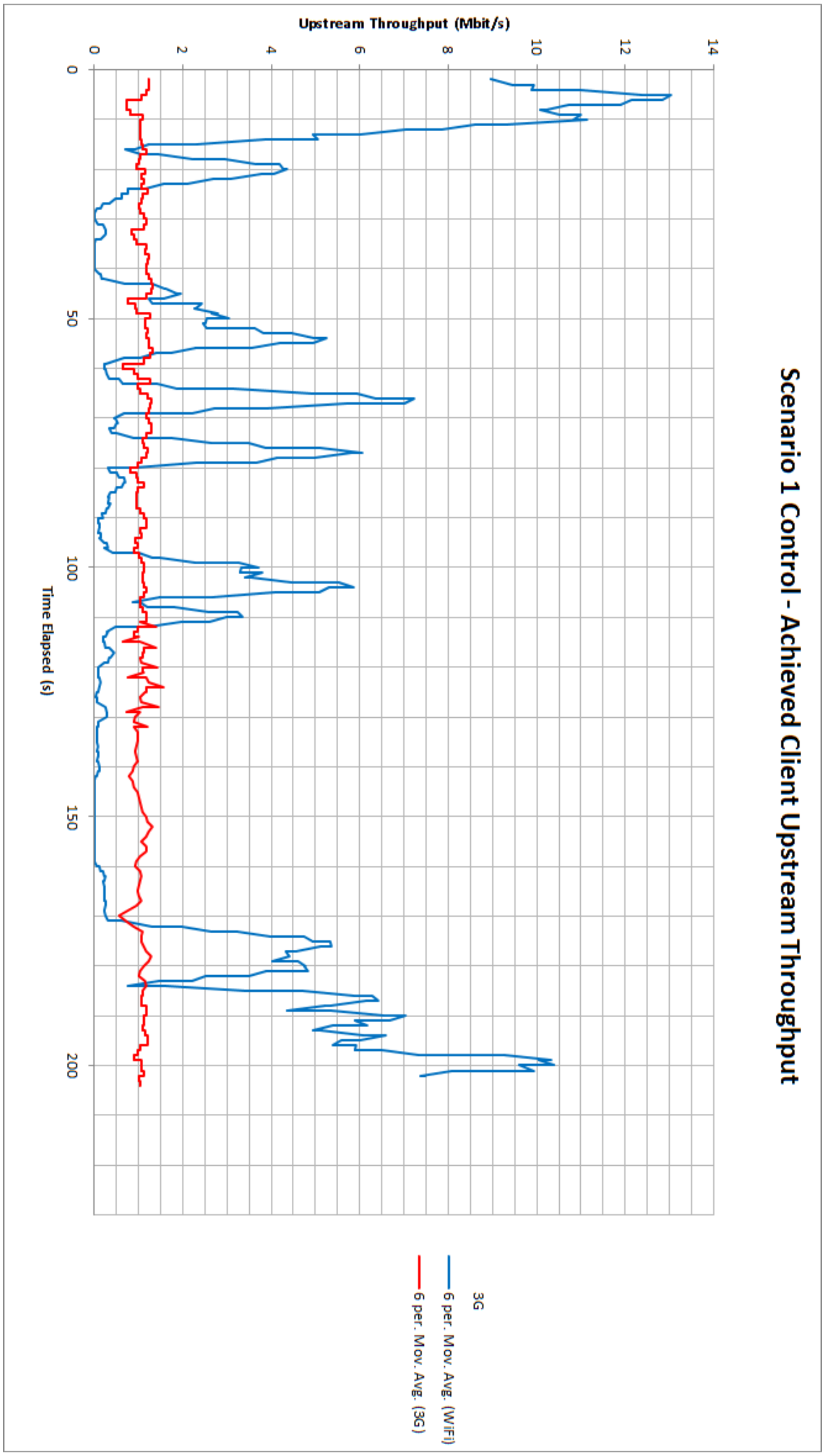

Figure 5.3: Throughput observed in control

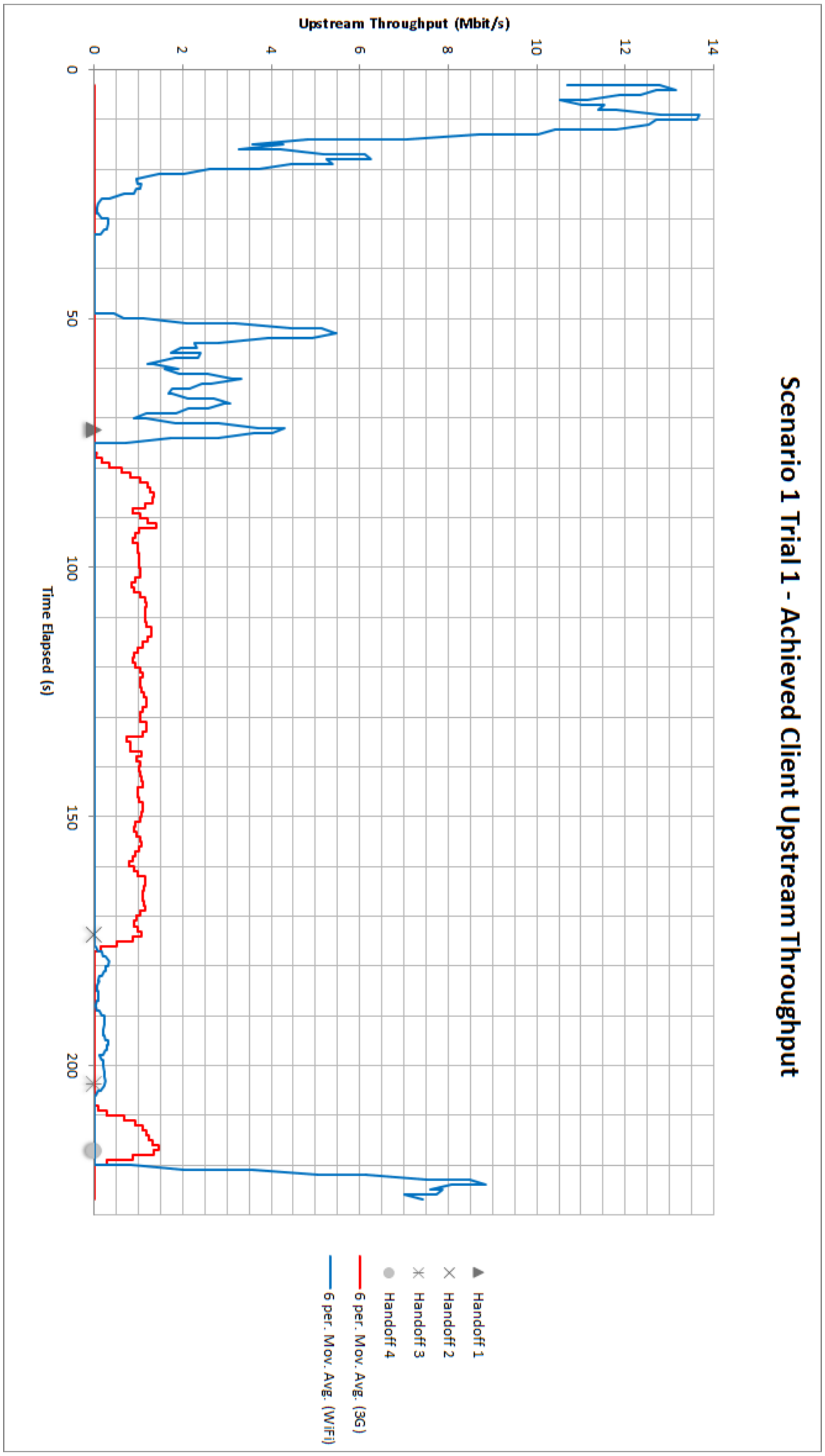

Figure 5.4: Throughput and handovers observed in scenario 1 trial 1

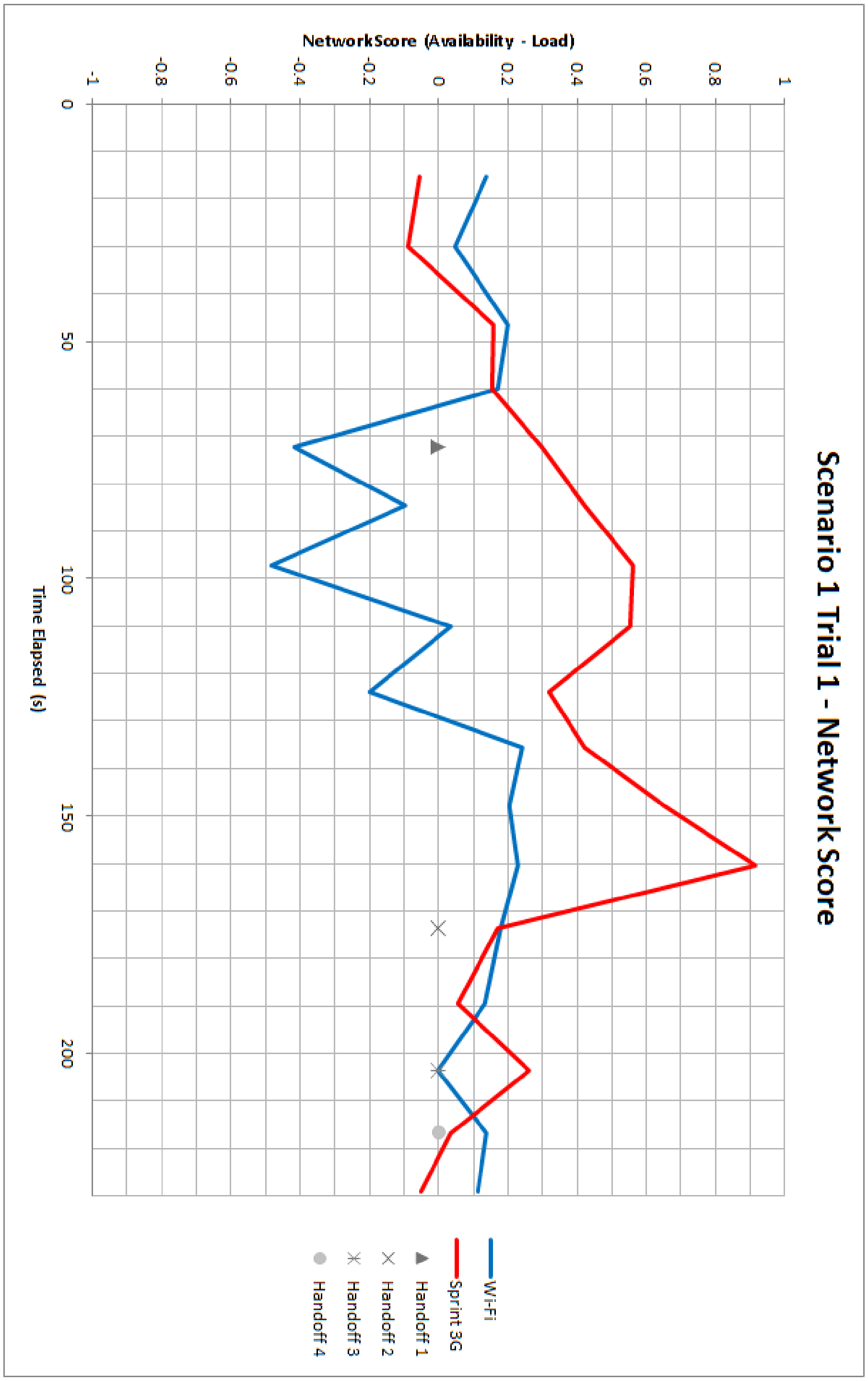

Figure 5.5: Network score and handovers observed in scenario 1 trial 1

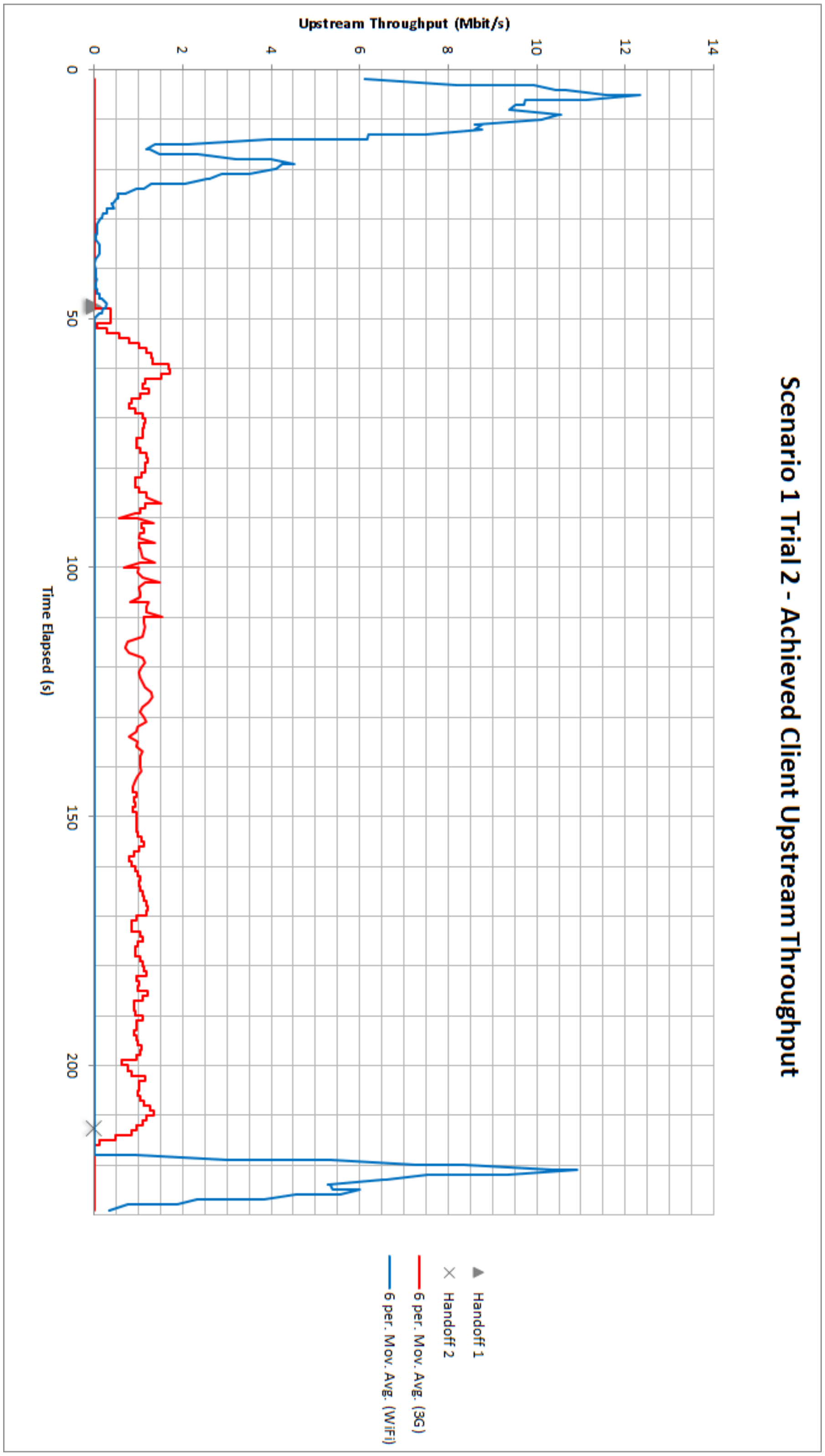

Figure 5.6: Throughput and handovers observed in scenario 1 trial 2

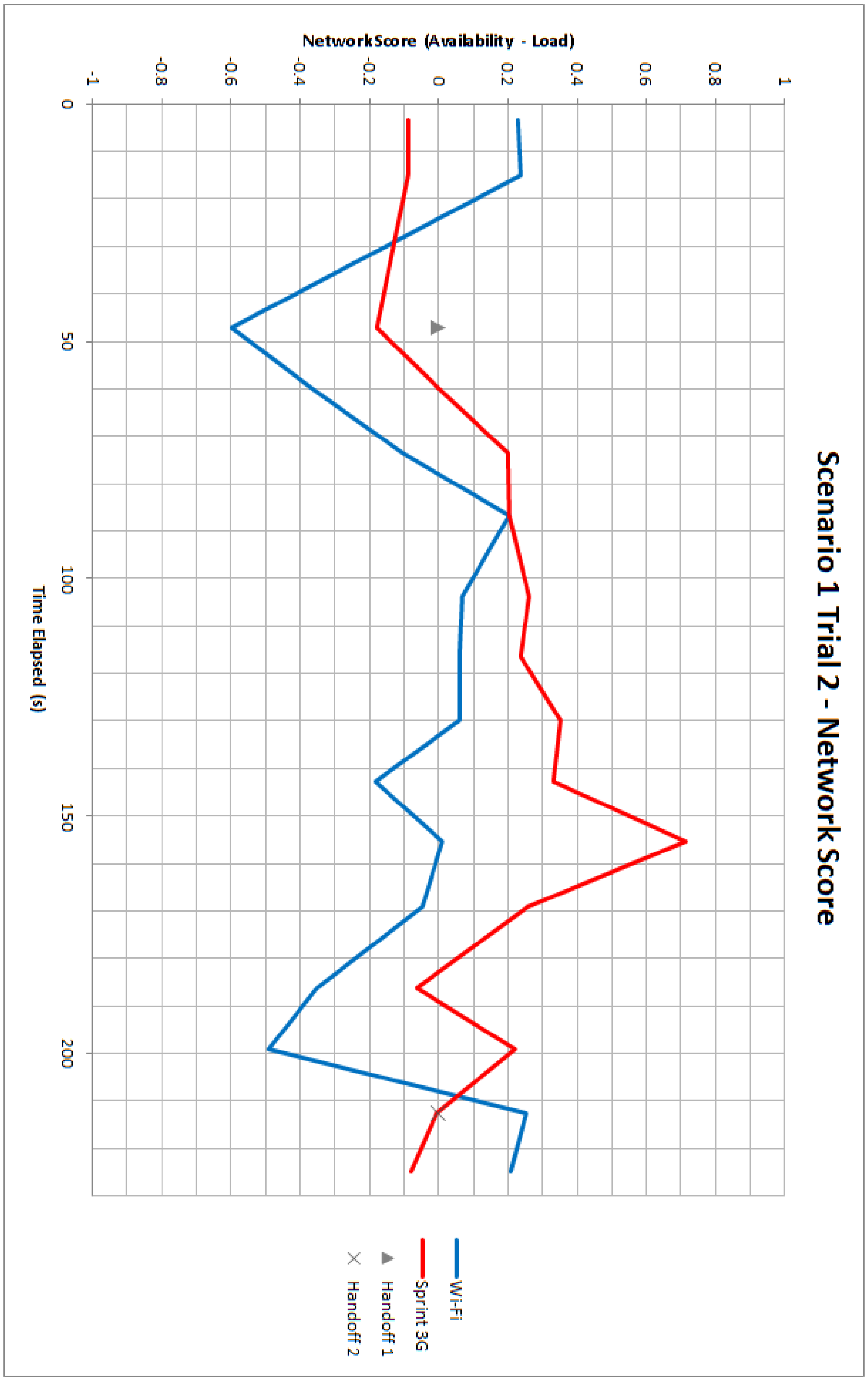

Figure 5.7: Network score and handovers observed in scenario 1 trial 2

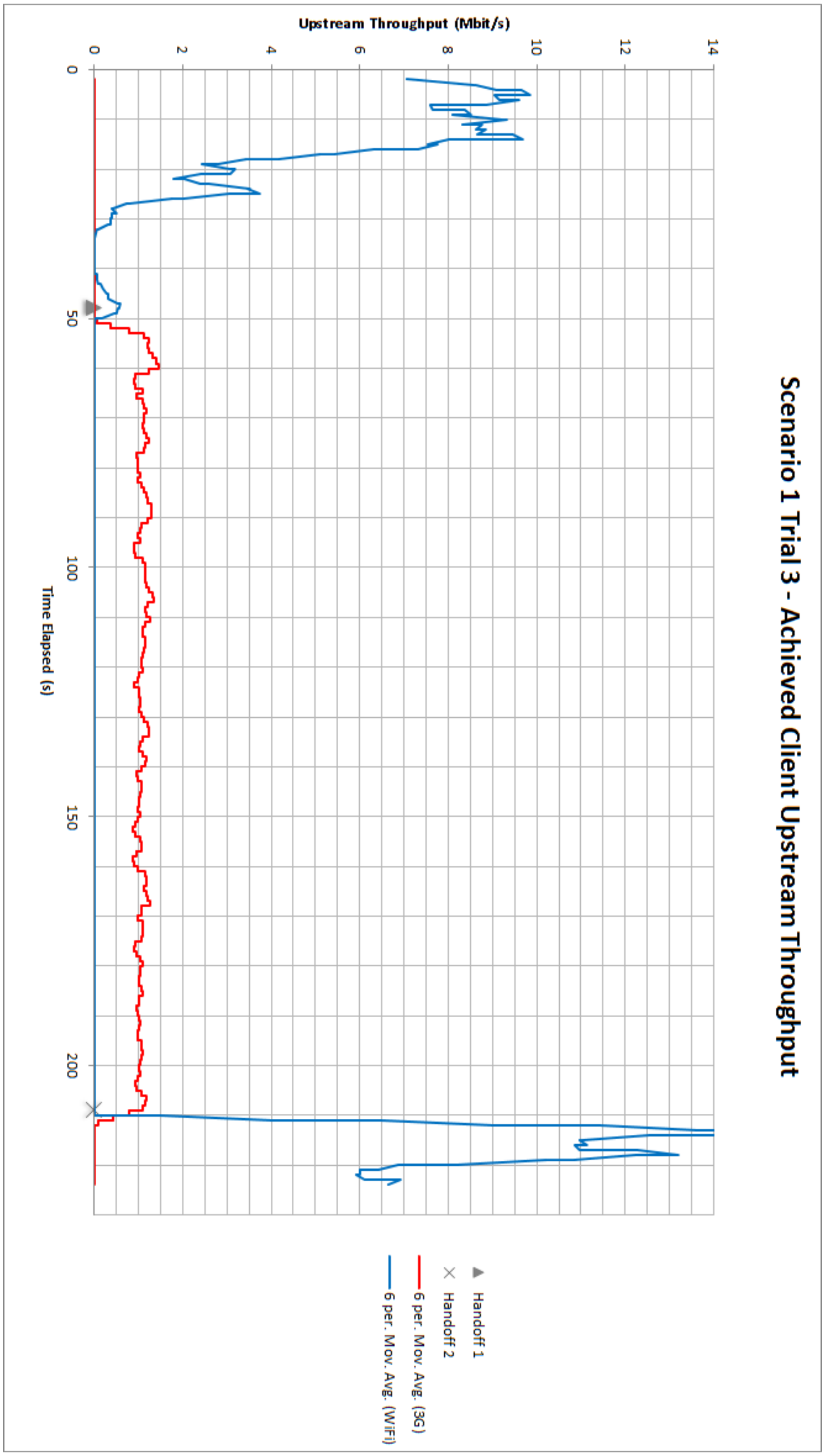

Figure 5.8: Throughput and handovers observed in scenario 1 trial 3

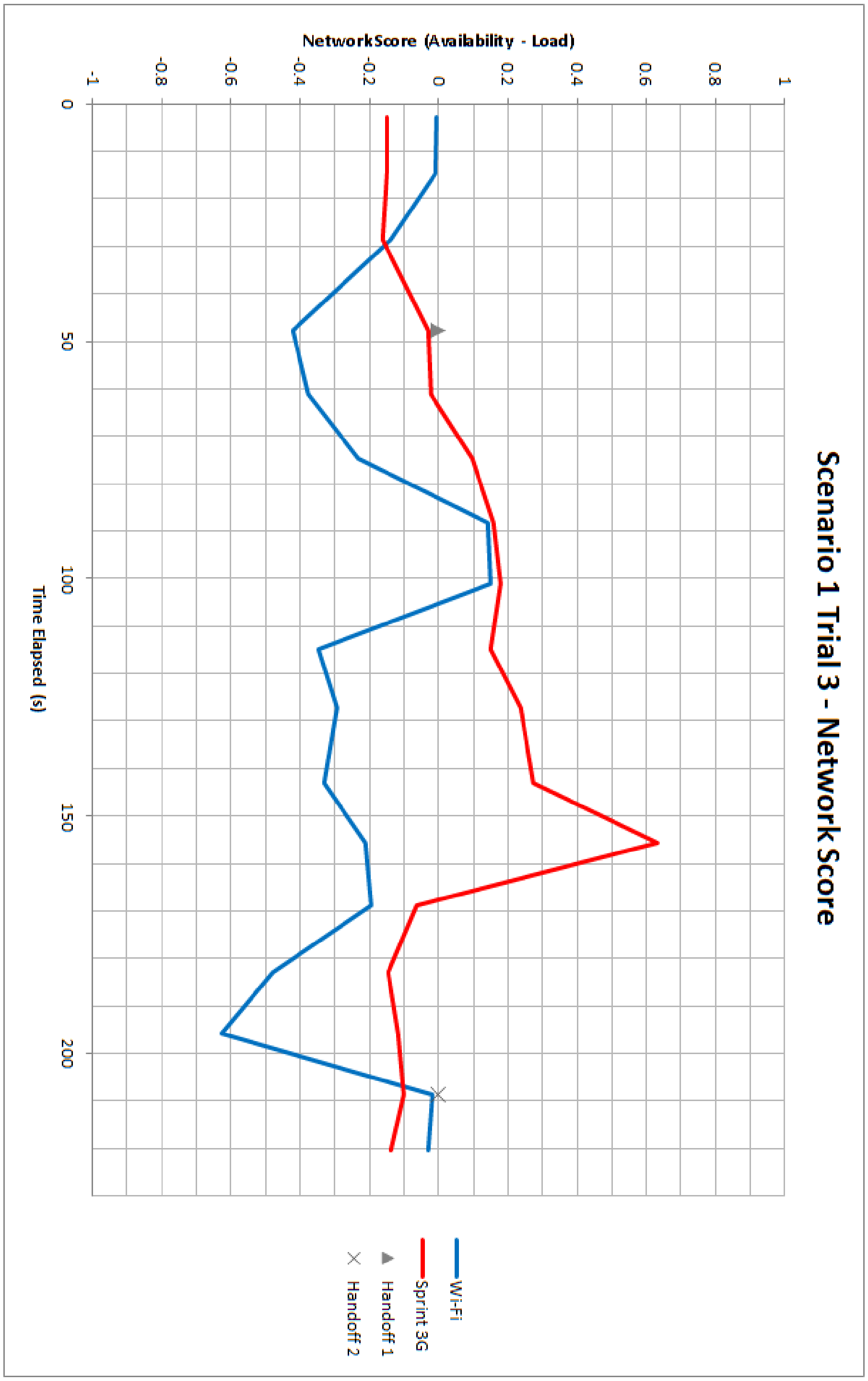

Figure 5.9: Network score and handovers observed in scenario 1 trial 3

	Time (s)	***RTT (ms)***	***RTT z-score***	***RSSI (dBm)***	***RSSI z-score***	***Network score***
WiFi	0.88	2.48	-0.315532	-59.869565	0.141106	0.456638
3G	0.88	238.6538	0.012662	-80.3846	0.114606	0.101944
WiFi	112.63	162.62	0.291095	-56.333333	0.460802	0.169707
3G	112.63	222.1379	-0.073422	-80.4483	0.108076	0.181498

Table 5.5: Scenario 2 trial 1 handoff measurements

	Time (s)	***RTT (ms)***	***RTT z-score***	***RSSI (dBm)***	***RSSI z-score***	***Network score***
WiFi	0.72	1.61	-0.320166	-63.826087	-0.218054	0.102112
3G	0.72	299.5625	0.330815	-80.5	0.102632	-0.228183
WiFi	88.93	63.32	-0.085068	-61.590909	-0.015223	0.069844
3G	88.93	241.8095	0.028847	-80.2857	0.124258	0.095411

Table 5.6: Scenario 2 trial 2 handoff measurements

during the third trial were uncharacteristically poor. The network test tool measured RTT values that were higher than normal. This increased latency observed on the 3G network resulted in the VHD algorithm showing a reluctance to switch the mobile device from WiFi to 3G. As the WiFi conditions worsened, however, the handoff did occur at around 138 seconds, 78 seconds after the network became congested.

This test scenario was designed to demonstrate the VHD reacting to a congested network, which the results successfully demonstrate. Figures 5.10, 5.12, and 5.14 show that the average RTT is lower on 3G after the load is imposed on the WiFi network. Even though the latency benefits are clear from our results, metrics such as packet loss or jitter can be used to further illustrate the experience improvement from a mobile device's perspective. In all trials, the achieved throughput was actually decreased after a handover to the 3G network. This is due to the high relative throughput that a WiFi network can provide. These throughput measurements suggest that network bandwidth should be a consideration in a VHD algorithm. In this scenario, the 3G network was under average load and yielded a lower average throughput than the congested WiFi network.

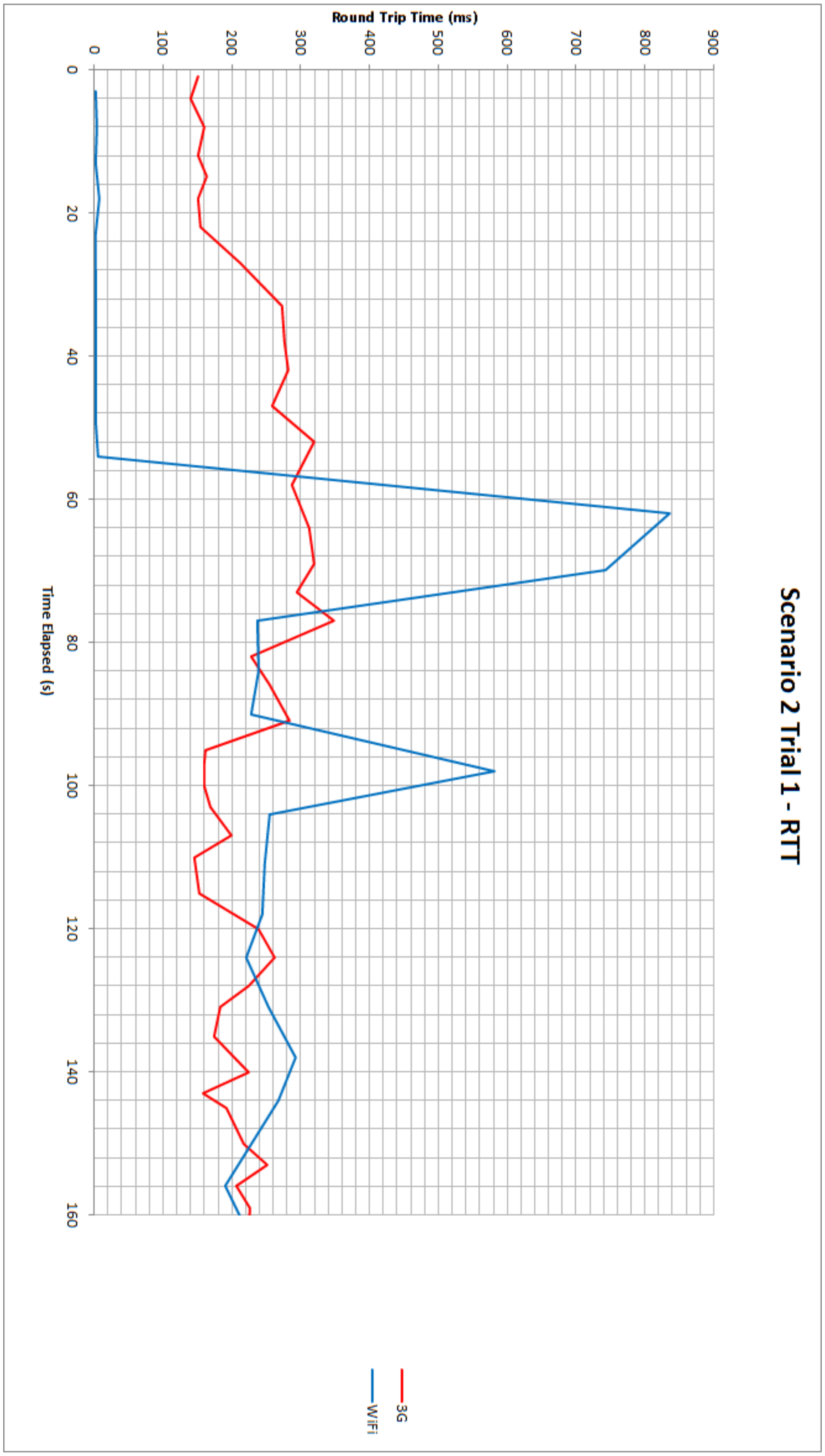

Figure 5.10: RTT observed by client in trial 1 of scenario 2

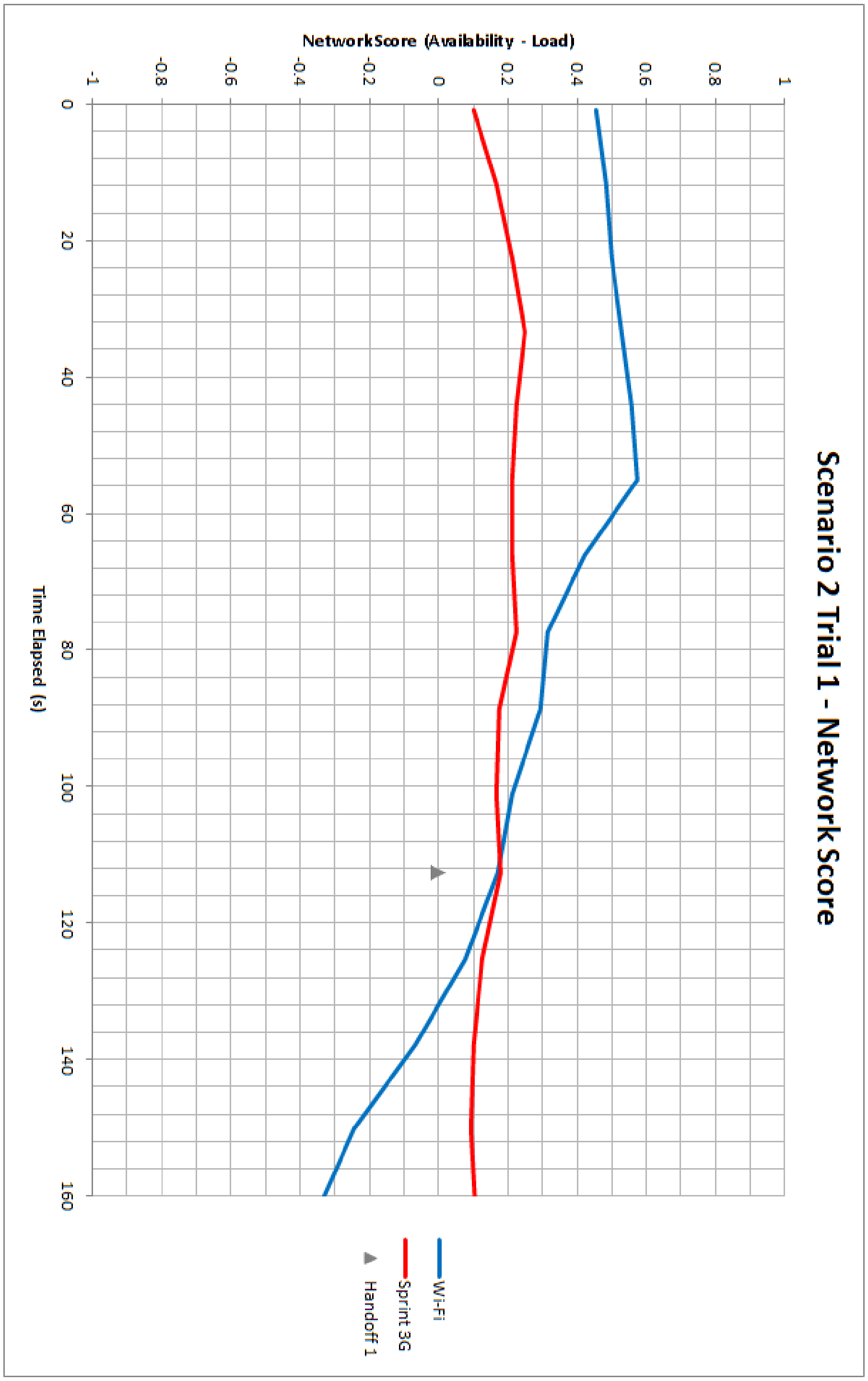

Figure 5.11: Network score and handoff observed in trial 1 of scenario 2

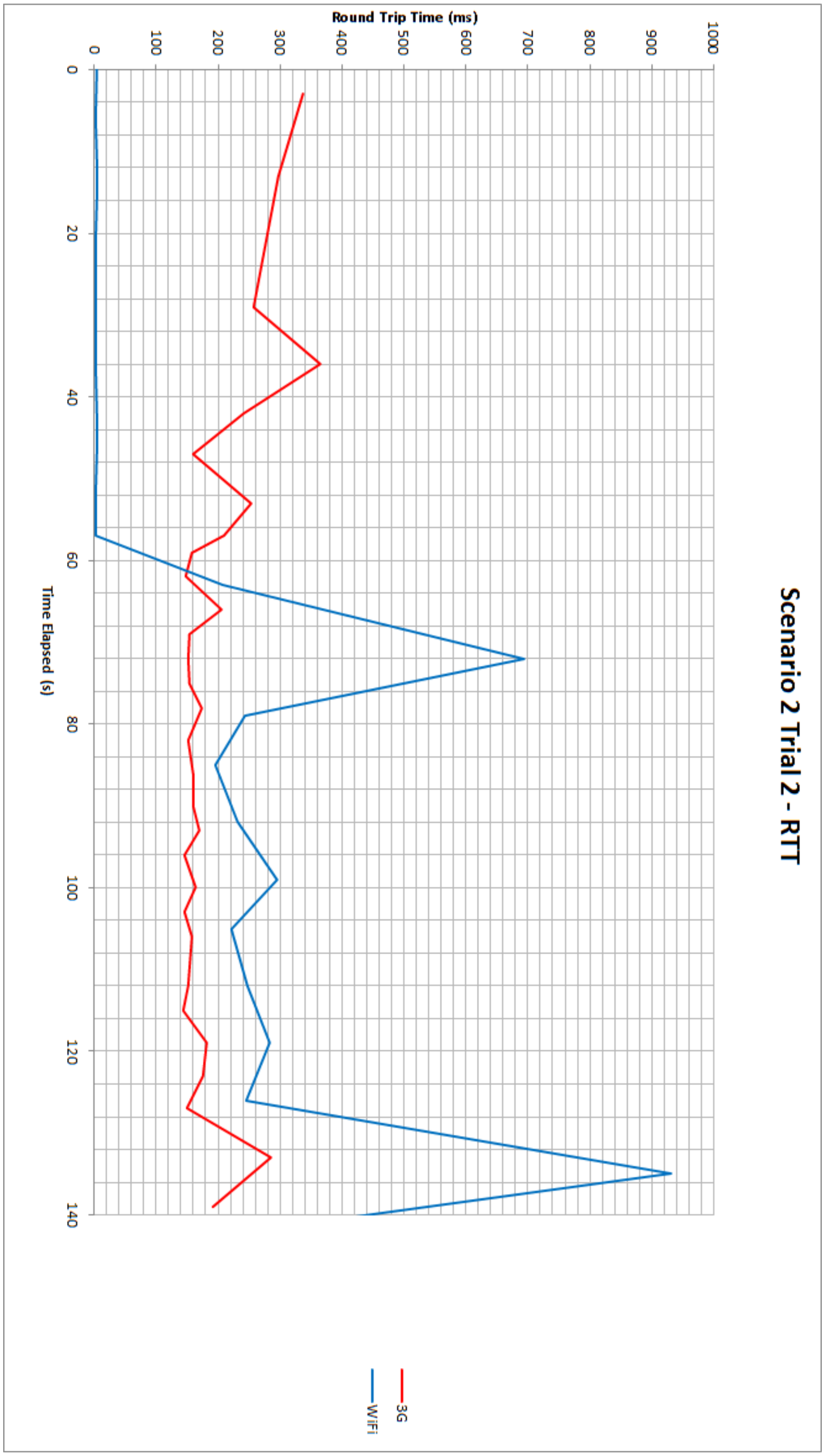

Figure 5.12: RTT observed by client in trial 2 of scenario 2

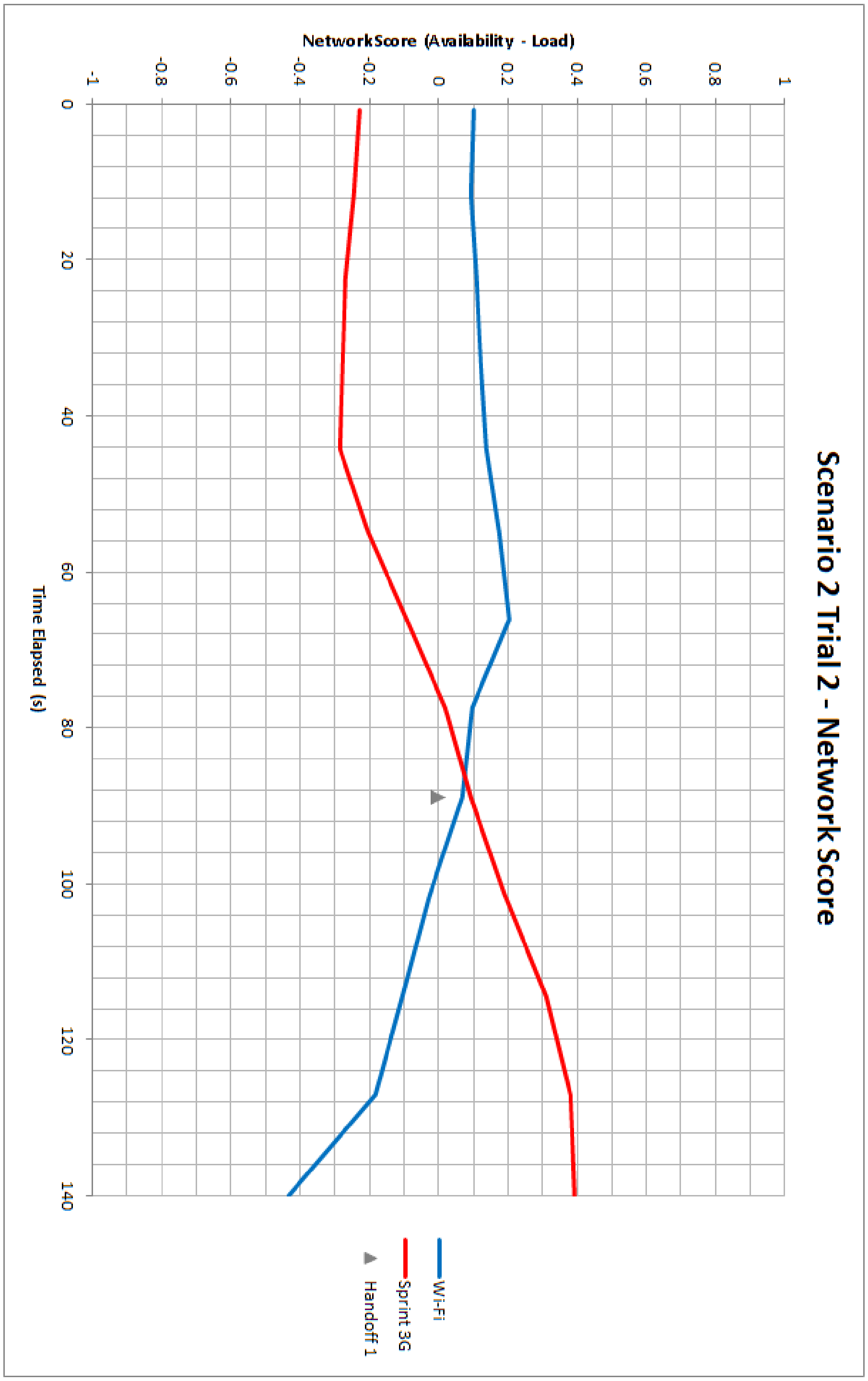

Figure 5.13: Network score and handoff observed in trial 2 of scenario 2

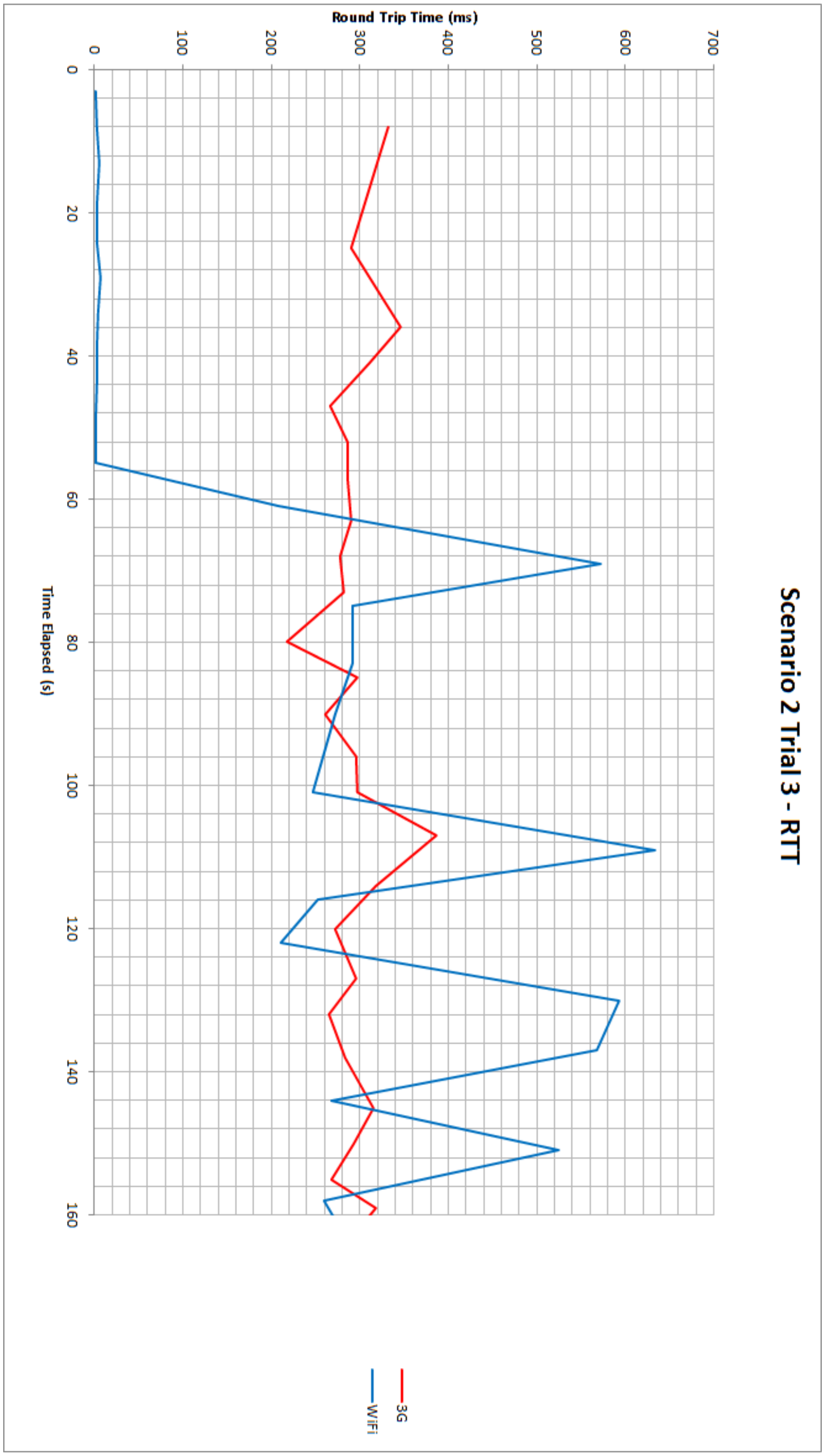

Figure 5.14: RTT observed by client in trial 3 of scenario 2

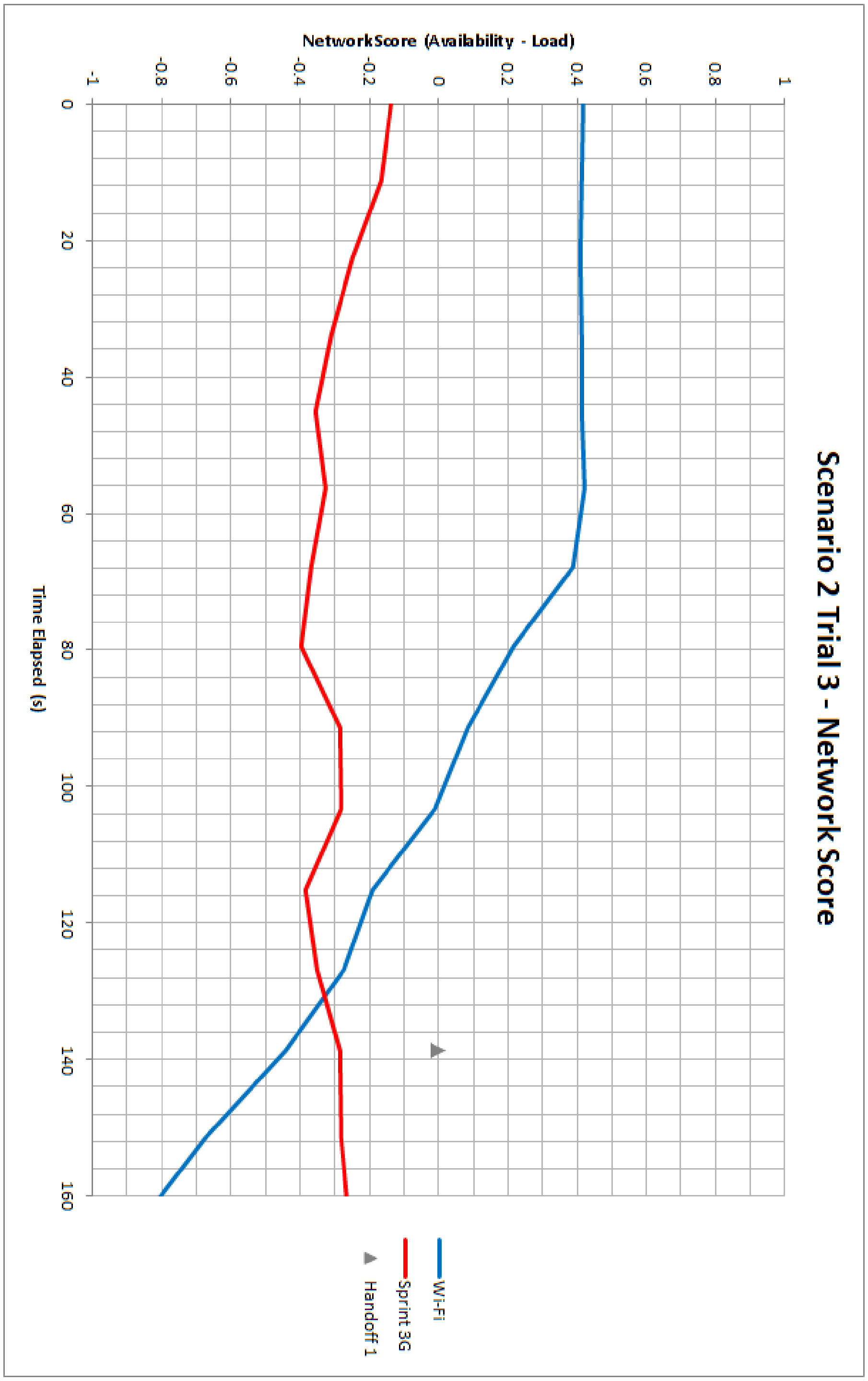

Figure 5.15: Network score and handoff observed in trial 3 of scenario 2

	Time (s)	***RTT (ms)***	***RTT z-score***	***RSSI (dBm)***	***RSSI z-score***	***Network score***
WiFi	0	1.25	-0.304357	-60.208333	0.112478	0.416835
3G	0	276.8571	0.211381	-80.7857	0.073491	-0.137891
WiFi	138.68	200.28	0.503857	-60.722222	0.06443	-0.439427
3G	138.68	302.7368	0.347607	-80.8947	0.062307	-0.2853

Table 5.7: Scenario 2 trial 3 handoff measurements

Chapter 6

Conclusions and Discussion

This thesis details the implementation of a service to enable research on a topic that is traditionally only studied through network simulation. The academic community has lagged behind the industry in the study of real-world implementations of ABC mechanisms due to lack of access to wireless deployments. The over-the-top constraint of this system is partially imposed out of necessity in an academic context, but also has the added benefit of allowing the service to be used globally within any heterogeneous wireless networks. The use of an over-the-top service allows vertical handover decisions to be approached from a researcher's perspective rather than a network operator's. In this way, real-world hetnets can be studied and used in experiments.

We saw, through the use of two test scenarios, how an over-the-top handover decision service can be used to provide improvements in network selection that otherwise would be infeasible. The adoption of an 'internet model' as opposed to traditional cellular-oriented approaches allows an over-the-top service to perform resource allocation among a diverse group of access networks. The results of the scenario trials are positive in that a fully functioning system was implemented and evaluated, but there is still work to be done in the way of maturing this system.

6.1 Future Work

The work described in this thesis is set to be expanded in future research efforts. More sophisticated handover decision algorithms are to be developed using this service. Algorithms must consider more than just RTT and RSSI; network capacity and user bandwidth consumption plays a large role in the achievable throughput in each network. Some other efforts [35] are already looking towards the real-world study of cloud-based handover decision services, so this thesis should serve as a strong foundation for those projects.

Improvements can also be made outside of the network selection algorithm. Further experimentation should be carried out to find the optimal values for the parameters listed in table 5. State can also be maintained in memory by the GRC - tracking "user requirements", or how much throughput each user is currently demanding from a network. An accurate picture of network traffic and consumption can allow for resource allocation solutions such as proportional fairness to be approximated via network selection. The LRC could incorporate a SDN-powered interface switching and IP mobility solution similar to the one presented in [20], allowing sockets to be maintained throughout the handoff process.

6.2 Answering the Research Questions

The viability of an over-the-top cloud-based handover decision service has been demonstrated through our work. The handover decision service, as well as the mobile network testing tool, is a service operating on the public internet that can be used by mobile devices operating on any mobile network. An algorithm based on the concept of relative network load and availability has been included in this service, which serves its purpose as a proof of concept for real-world over-the-top network selection. The functionality of this algorithm, as well as all of the other components has been demonstrated through use of test scenarios, covering two major use cases of such a service. The proof-of-concept work in this thesis has shown measurable benefits over current local network selection methods, but more importantly, provides the groundwork for further real-world research on over-the-top vertical handover decision algorithms.

Appendices

Appendix A Protocol Buffers Message Examples

```
message ClientDetails {
    optional string mac = 1;
    optional string make = 2;
    optional string model = 3;
    optional string platform = 4;
    optional string version = 5;

    optional double longitude = 6;
    optional double latitude = 7;
    optional double altitude = 8;
    optional double gps_accuracy = 9;
    optional sint32 geolocation_method= 10;

    optional string network_type = 11;
    optional string network_operator=12;
    optional double max_bitrate = 13;
    optional double signal_dbm = 14;
    optional string ssid = 15;
    optional string bssid = 17;

    optional sint32 test_requested = 16;
}

message ServerAck {
    optional sint32 port = 1;
    optional string message = 2;
}

message TestResults {
```

```
    optional sint32 test_requested = 1;
    optional string message = 2;

    optional double bandwidth_up = 3;
    optional double bandwidth_down = 4;
    optional double latency = 5;

    optional sint32 packet_obs = 6;
    optional double loss_percent = 7;
    optional sint32 blc = 8;
    optional double average_jitter = 9;
    optional double mbl = 10;
    optional double mild = 11;

    optional double duration = 12;
}
```

Bibliography

[1] Umip - mobile ipv6 and nemo for linux. http://umip.org/.

[2] Rahul Amin. *Towards Viable Large Scale Heterogeneous Wireless Networks*. PhD thesis, Clemson University, August 2013.

[3] Yutaka Arakawa, Y Yamaguchi, S Tagashira, and A Fukuda. Application layer active wireless network switching on a smartphone. In *International Workshop on Smart Mobile Applications*, 2012.

[4] Paul Bender, Peter Black, Matthew Grob, Roberto Padovani, Nagabhushana Sindhushyana, and Andrew Viterbi. Cdma/hdr: a bandwidth efficient high speed wireless data service for nomadic users. *Communications Magazine, IEEE*, 38(7):70–77, 2000.

[5] T. Bu, Li Li, and R. Ramjee. Generalized proportional fair scheduling in third generation wireless data networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[6] Jesús Calle-Cancho, David Cortés-Polo, Javier Carmona-Murillo, and Jose-Luis González-Sánchez. Experimental evaluation of mobile ipv6 for linux. *International Journal of Computer Science and Telecommunications*, Volume 4(Issue 12), dec 2013.

[7] Federal Communications Commission. Measuring broadband america - mobile broadband services. http://www.fcc.gov/measuring-broadband-america/mobile, 2013.

[8] A. Damnjanovic, J. Montojo, Yongbin Wei, Tingfang Ji, Tao Luo, M. Vajapeyam, Taesang Yoo, Osok Song, and D. Malladi. A survey on 3gpp heterogeneous networks. *Wireless Communications, IEEE*, 18(3):10–21, June 2011.

[9] DSLReports.com. Broadband dslreports.com. http://www.dslreports.com/, 2014.

[10] Dave Evans. The internet of things - how the next evolution of the internet is changing everything. http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, April 2011.

[11] Linux Foundation. netem. http://www.linuxfoundation.org/collaborate/workgroups/networking/netem, 2014.

[12] Caroline Gabriel. Wireless broadband alliance industry report 2013: Global trends in public wifi. http://www.wballiance.com/wba/wp-content/uploads/downloads/2013/11/WBA-Industry-Report-2013.pdf, November 2013.

[13] Google. Ipv6 adoption statistics. https://www.google.com/intl/en/ipv6/statistics.html, June 2014.

[14] Volker Gropp. bwm-ng (bandwidth monitor ng). http://www.gropp.org/?id=projects&sub=bwm-ng, 2014.

[15] WG802.21 Media Independent Handoff Working Group. Ieee 802.21. http://www.ieee802.org/21/, 2009.

[16] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6. RFC 5213 (Proposed Standard), August 2008. Updated by RFC 6543.

[17] E. Gustafsson and A. Jonsson. Always best connected. *Wireless Communications, IEEE*, 10(1):49–55, Feb 2003.

[18] USGlobalSat Inc. About the bu-353 usb gps receiver. http://www.usglobalsat.com/p-62-bu-353-w.aspx, 2014.

[19] Iperf.fr. Iperf - a modern alternative for measuring tcp and udp bandwidth performance. https://iperf.fr/, 2014.

[20] R. Izard, A. Hodges, J. Liu, J. Martin, K.-C. Wang, and K. Xu. An openflow testbed for the evaluation of vertical handover decision algorithms in heterogeneous wireless networks. In *TRIDENTCOM 2014*, 2014.

[21] R. Koodli. Mobile IPv6 Fast Handovers. RFC 5568 (Proposed Standard), July 2009.

[22] NETGEAR. Aircard 341u usb modem. http://www.netgear.com/service-providers/products/mobile/usb-modems/341u.aspx, 2014.

[23] SEVEN Networks. Andsf - essential enabling technology for mobile data traffic offloading. http://www.seven.com/technology-andsf-wifi-offloading.php, 2014.

[24] Ookla. Ookla speedtest. http://www.speedtest.net/, 2014.

[25] C. Perkins. IP Mobility Support for IPv4, Revised. RFC 5944 (Proposed Standard), November 2010.

[26] C. Perkins, D. Johnson, and J. Arkko. Mobility Support in IPv6. RFC 6275 (Proposed Standard), July 2011.

[27] Rima Qureshi. Ericsson mobility report. http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf, June 2014.

[28] Armin Ronacher. Flask - web development, one drop at a time. http://flask.pocoo.org/, 2014.

[29] Rootmetrics. The rootmetrics testing methodology. http://www.rootmetrics.com/us/methodology, 2014.

[30] Brent Skorup. Reclaiming federal spectrum: Proposals and recommendations. *Columbia Science and Technology Law Review*, Vol. XV, 2013.

[31] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier. Hierarchical Mobile IPv6 (HMIPv6) Mobility Management. RFC 5380 (Proposed Standard), October 2008.

[32] Advanced Telecommunications and Networks Group. Odtone - an open source multi-platform implementation of the ieee 802.21 protocol. http://atnog.github.io/ODTONE/, 2013.

[33] Lusheng Wang and G-SGS Kuo. Mathematical modeling for network selection in heterogeneous wireless networksa tutorial. *Communications Surveys & Tutorials, IEEE*, 15(1):271–292, 2013.

[34] Ruckus Wireless. Hotspot 2.0 - making the wi-fi roaming experience as secure and easy to use as with cellular data. http://www.ruckuswireless.com/technology/hotspot2, 2014.

[35] K. Xu, K.-C. Wang, R. Amin, J. Martin, and R. Izard. A fast cloud-based network selection scheme using coalition formation games in vehicular networks. In *IEEE Transactions on Vehicular Technology*, 2014.

[36] Xiaohuan Yan, Y. Ahmet ekerciolu, and Sathya Narayanan. A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks. *Computer Networks*, 54(11):1848 – 1863, 2010.