Section A: Project Summary

This proposal addresses the broad area of operating system support for quality-of-service (QOS) resource management. It specifically targets systems that exploit this resource management strategy to achieve adaptive behavior in *mobile computing.* Building on techniques for adaptation recently identified by ourselves and other researchers, our proposed work will develop the critical understanding and the experimental tools and techniques needed to compare the merits of alternative approaches to adaptation. It is likely that the insights offered by such evaluation will lead to the development of new techniques for adaptation. Further, the resource that has dominated attention so far in the research community is *network bandwidth.* We propose to broaden the scope of adaptation to include *battery power.*

We will focus on three areas:

- Developing the techniques and tools to *accurately map energy usage to software components*, and then using them to develop applications that adapt to battery level.
- Designing the experimental infrastructure and techniques that enable *accurate re-creation of wireless networking conditions* encountered by mobile systems, and then using these to investigate the behavior of a prototype adaptive system.
- Exploring approaches to *quantifying the behavior of adaptive systems* involving multiple resources and concurrent applications, and using these to develop benchmarks for adaptive mobile systems.

This work will be done in the context of *Odyssey,* a prototype system that supports *application-aware adaptation.* In this model of adaptation, the operating system performs monitoring and allocation of resources such as network bandwidth and battery power. Individual applications decide how best to translate the levels of resources available into appropriate adaptations. Odyssey will be the crucible in which we generate, test and refine our ideas in the three areas mentioned above. Our research may suggest important improvements and extensions to Odyssey, and we will pursue these as appropriate. However, such refinements are incidental to the primary goals of this research.

Our approach will be experimental in nature, resulting in the development of tools, techniques and benchmarks that embody our answers. The work will involve software implementation, controlled experiments, and empirical data collection. Validation and impact of our research will come from their successful application and use in the next generation of adaptive mobile systems.

We foresee the work proposed here contributing many important results toward the goal of developing a well-validated and well-accepted set of design principles for mobile computing systems. Making energy a first-class resource is critical for the next generation of mobile systems. Enabling researchers to conduct experiments involving physical motion will facilitate scientific studies of the effects of adaptive strategies on the usability of mobile computing sytems. Finally, the ability to quantify the multiple facets of adaptation without oversimplification will make it possible to develop benchmarks that allow well-grounded comparisons between alternative system designs. The ensuing ability to quantitatively demonstrate improvements will be a catalyst for progress in the field.

Section B: Table of Contents

Section C: Evaluating Adaptive Mobile Systems

C.1 Problem Statement

Adaptation is the key to mobility. Only through effective adaptation can applications on a mobile client overcome the many challenges they face. These include unpredictable variation in network quality, wide disparity in the availability of remote services, limitations on local resources imposed by weight and size constraints, and concern for battery power consumption.

Responding to this challenge, a number of experimental systems have been built over the last few years to explore concepts relevant to adaptation for mobility. Examples include our own work on Odyssey [12] and Coda [10], as well as work by other researchers such as the Rover toolkit [6], dynamic distillation of Web data [5], disconnectable Web browsing [9], variable consistency data management in Bayou [20] and Rumor [15], adaptive transport protocols [3, 2] and adaptive location management in Mobile IP [14].

Where do we go from here? As a research community, we have successfully identified certain techniques that can serve as building blocks in the design of adaptive mobile systems. We now need to develop the critical understanding and the experimental tools and techniques needed to compare the merits of alternative approaches to adaptation, and to quantitatively assess the results of their application in specific situations. It is likely that the insights offered by such evaluation will lead to the development of new techniques for adaptation. Further, the resource that has dominated attention so far has been *network bandwidth.* We now need to broaden the scope of adaptation to include other resources, most importantly *battery power.*

The research that lies ahead can be characterized by the following set of questions:

- Given two adaptive systems *A* and *B*, how does one compare their adaptative capabilities? What benchmarks does one use? What are the appropriate figures of merit? How does one characterize the range of circumstances in which one system is better than the other? How does one quantify the extent of this superiority?
- How does one strike the right balance between the need for realism (so that an evaluation is meaningful), and the need for good experimental control (so that the evaluation is reproducible and consitutes good science)? Specifically, how does one conduct such a balanced evaluation involving two key variables, *physical motion* and *wireless network quality*, that are notoriously difficult to characterize and control?
- How does one characterize and analyze the effects of adaptation on energy consumption? How does one attribute observed effects to specific software components? Are the adaptive techniques for energy conservation consistent with, or contradictory to, those used for conserving network bandwidth?
- When adaptation involves multiple resources, how does one characterize the tradeoffs involving these resources? How does one perform controlled experiments involving multiple resources? What figures of merit and benchmarks can we use in such evaluations?

C.2 Research Strategy

Our research aims to make substantial progress toward answering these questions. We will focus on three areas:

- Developing the techniques and tools to *accurately map energy usage to software components,* and then using them to develop applications that adapt to battery level.
- Designing the experimental infrastructure and techniques that enable *accurate re-creation of wireless networking conditions* encountered by mobile systems, and then using these to investigate the behavior of a prototype adaptive system.
- Exploring approaches to *quantifying the behavior of adaptive systems* involving multiple resources and concurrent applications, and using these to develop benchmarks for adaptive mobile systems.

This work will be done in the context of *Odyssey,* a prototype system that supports *application-aware adaptation.* In this model of adaptation, the operating system performs monitoring and allocation of resources such as network bandwidth and battery power. Individual applications decide how best to translate the levels of resources available into appropriate adaptations. Odyssey will be the crucible in which we generate, test and refine our ideas in the three areas mentioned above. Our research may suggest important improvements and extensions to Odyssey, and we will pursue these as appropriate. However, such refinements are incidental to the primary goals of this research.

Our approach will be experimental in nature, resulting in the development of tools, techniques and benchmarks that embody our answers. The work will involve software implementation, controlled experiments, and empirical data collection. Validation and impact of our research will come from their successful application and use in the next generation of adaptive mobile systems.

The next four sections describe our research in more detail. We begin in Section C.3 with an overview of Odyssey. This description is necessarily brief; further details can be found in a recent paper [12]. Following this overview, we elaborate on the three focus areas. In each of the corresponding sections, C.4 to C.6, we first provide conceptual background, then present the current status of our work, and finally describe our future plans. We complete the proposal with a discussion of expected impact in Section C.7.

C.3 Overview of Odyssey

Adaptation in Odyssey involves the trading of data quality for resource consumption. For example, when the bandwidth available to a video application drops, it could switch to a video stream in black and white or coarser resolution to avoid suffering dropped frames. Similarly, when the bandwidth available to a Web browser drops, it might fetch images that have been aggressively compressed rather than suffering long delays in fetching the full-quality versions. Odyssey captures this notion of data degradation through an attribute called *fidelity*, that defines the degree to which data presented at a client matches the reference copy at a server. This is necessarily a type-specific notion since different types of data can be degraded differently.

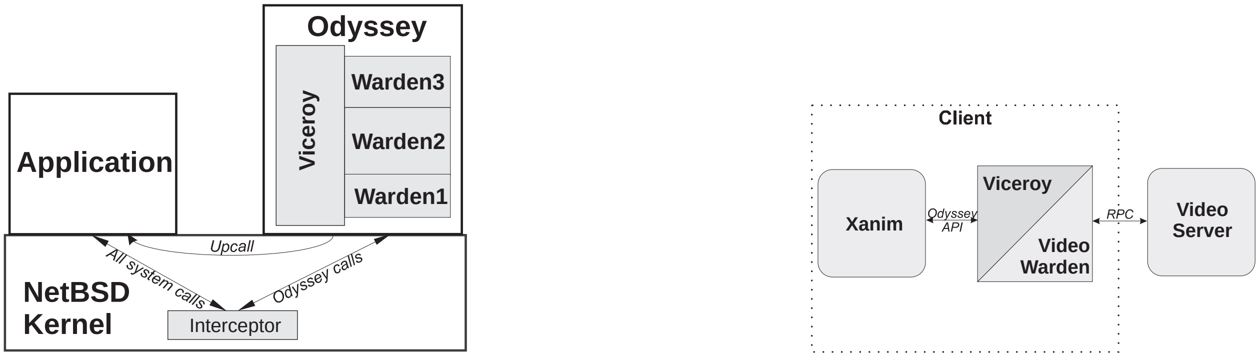
The architecture of an Odyssey client is shown in Figure 1(a). Odyssey is conceptually part of the operating system, even though it is implemented in user space for simplicity. The *viceroy* is the Odyssey component responsible for monitoring the availability of resources and managing their use. Code components called *wardens* encapsulate type-specific functionality. There is one warden for each data type in the system.

An application communicates resource expectations to Odyssey through a set of extensions to the Unix API. These extensions provide the ability for applications to indicate a *window of tolerance* for each resource. If the available level of a resource drops below the lower limit of this window, or rises above its upper limit, Odyssey notifies the application through an *upcall*. The application then adjusts its fidelity to match the new resource level and informs Odyssey of its new window of tolerance. The Odyssey API is general enough to allow expectations to be expressed about a broad range of resources such as network bandwidth and latency, disk space, computing cycles, remaining battery power, and money. However, the current Odyssey prototype only pays attention to network bandwidth.

We have built three adaptive applications on top of Odyssey. A major design goal of Odyssey is the ability to effectively support adaptation by diverse, concurrent applications such as these. The first application, shown in Figure 1(b), is an adaptive video player that stores multiple tracks of each video segment on the server. The client switches to a higher or lower fidelity track as bandwidth varies during replay.

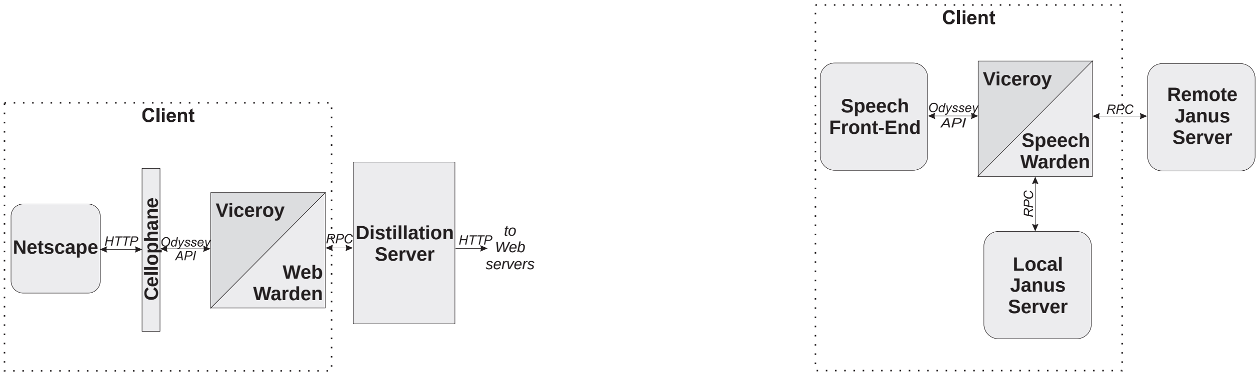
The second application, shown in Figure 1(c), is an adaptive Web browser based on Netscape. In this application, Odyssey and a distillation server located on either side of a variable-quality network mediate access to Web servers. Depending on the bandwidth available between Odyssey and the distillation server, Web images are degraded through lossy JPEG compression before transmission to the client.

The third application, shown in Figure 1(d), is an adaptive speech recognizer based on the Janus speech recognition system [21]. When bandwidth is high, speech is shipped to a high-performance remote server for recognition. When disconnected, speech recognition is performed locally; to reduce memory and CPU requirements, a smaller vocabulary and acoustic model are used and the user is informed of this fact through a synthesized voice. At intermediate bandwidths, the first phase of recognition is performed locally, resulting in a compact intermediate representation; this is then shipped to the remote server to complete the recognition.



(a) Client Architecture

(b) Video Player



(c) Web Browser

(d) Speech Recognizer

Figure 1: Odyssey and Adaptive Applications

C.4 Mapping Energy Usage to Program Structure

C.4.1 Conceptual Background

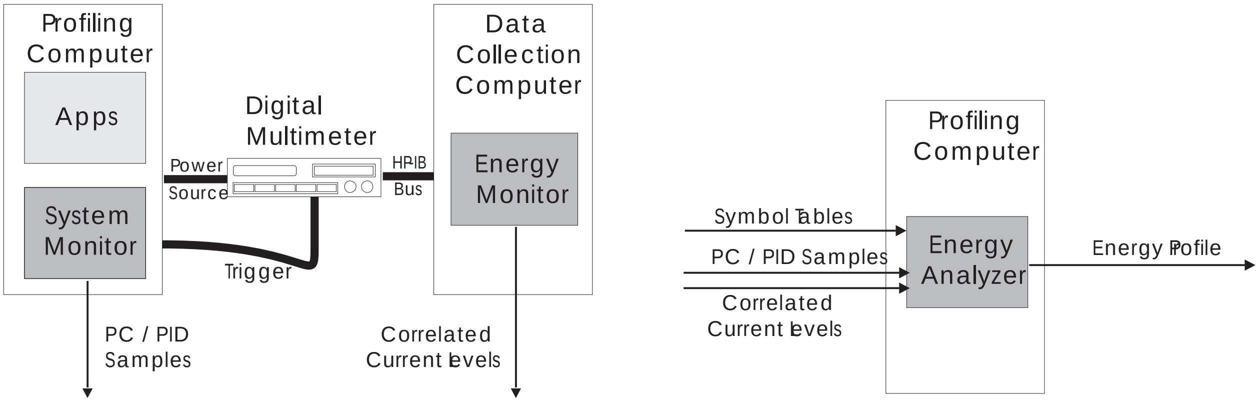
Energy is a critical resource for mobile computers [8, 23]. In spite of many improvements in low-power hardware design and battery life, there is now growing awareness that a strategically viable approach to energy management must include higher levels of the system [13]. For example, a network application that offers acceptable service while temporarily disconnected can save a considerable amount of energy by suppressing non-essential wireless communication. The resulting savings add to those offered by energy-efficient hardware. In contrast, efforts aimed solely at the hardware cannot benefit from application-specific knowledge.

Progress in energy-efficient software design requires the ability to attribute energy consumption to specific software components, in much the same way that CPU profilers such as *prof* and `gprof` help expose code components that are wasteful of processor cycles. To address this need, we have built an initial version of a tool called *PowerScope* to profile energy usage.

PowerScope maps energy consumption to program structure. Using PowerScope, one can determine what fraction of the total energy consumed during a certain time period is due to specific processes in the system. Further, one can drill down and determine the energy consumption of different procedures within a process. By providing such fine-grained feedback, PowerScope allows attention to be focused on those system components responsible for the bulk of energy consumption. As improvements are made to these components, PowerScope quantifies their benefits and helps expose the next target for optimization. Through successive refinement, a system can be improved to the point where its energy consumption meets design goals.

C.4.2 Research Status

The prototype version of PowerScope, shown in Figure 2, uses statistical sampling to profile the energy usage of a computer system. To reduce overhead, profiles are generated by a two-stage process. During the data collection stage, the tool samples both the power consumption and the system activity of the profiling computer. PowerScope then generates an energy profile from this data during a later analysis stage. Because the analysis is performed off-line, it creates no profiling overhead.



(a) Data Collection (b) Off-Line Analysis

Figure 2: PowerScope Architecture

During data collection, we use a digital multimeter to sample the current drawn by the profiling computer through its external power input. We require that this multimeter have an external trigger input and the ability to sample DC current at high frequency (at least every 10 ms.). Our current implementation uses a Hewlett Packard 3458a digital multimeter, which satisfies both these requirements. A separate data collection computer controls the multimeter and stores samples from it.

The functionality of PowerScope is divided among three software components. As shown in Figure 2(a), the System Monitor and Energy Monitor components share responsibility for data collection. The System Monitor samples system activity on the profiling computer by periodically recording information which includes the program counter (PC) and process identifier (PID) of the currently executing process. The Energy Monitor runs on the data-collection computer, and is responsible for collecting and storing current samples.

The Energy Analyzer, shown in Figure 2(b), uses the raw sample data collected by the monitors to generate the energy profile. The analyzer runs on the profiling computer since it uses the symbol tables of the executables on disk to map samples to specific procedures. There is an implicit assumption in this method that the executables being profiled are not modified between the start of profile collection and the running of the off-line analysis tool.

Using PowerScope, we have conducted preliminary experiments to understand the energy usage of the adaptive video application shown in Figure 1(b). Figure 3 illustrates a subset of the profile information currently produced by PowerScope for this application.

Guided by the energy profile information provided by PowerScope, we have been able to obtain a 43% reduction in the total energy consumption of the adaptive video application on Odyssey. Some of the steps in the path to achieving this reduction were counterintuitive — for example, lossy compression resulted in hardly any energy savings, while reducing the size of the displayed image proved much more effective. Thus, although this research is still in its early stages, we are convinced that PowerScope is a valuable tool for exploring adaptive techniques for conserving energy.

```
                                   Elapsed       Total       Average
Process                           Time (s)    Energy (J)    Power (W)
------------------------------   ----------   ----------   ----------
/usr/odyssey/bin/xanim                66.57       643.17         9.66
/usr/X11R6/bin/X                      35.72       331.58         9.28
/netbsd (kernel)                      50.89       328.71         6.46
Interrupts-WaveLAN                    18.62       165.88         8.91
/usr/odyssey/bin/odyssey              12.19       123.40        10.12
------------------------------   ----------   ----------   ----------
Total                                183.99      1592.75         8.66
```

(a) Summary of Energy Usage by Process

```
          Energy Usage Detail for process Interrupts-WaveLAN

                       Kernel-level procedures:
                                   Elapsed       Total       Average
Procedure                         Time (s)    Energy (J)    Power (W)
------------------------------   ----------   ----------   ----------
_xferDMAbuffer                        16.66       147.38         8.85
_pwlread                               0.30         2.90         9.65
_pwlget                                0.30         2.68         8.93
_pwlintr                               0.24         2.31         9.62
```

(b) Energy Usage of WaveLAN Interrupt Handler

Figure 3: Example Output from PowerScope

C.4.3 Future Plans

The current PowerScope prototype positions us well to conduct a detailed and long-term investigation of energy conservation in mobile systems. Our first task will be to study and understand how the adaptive strategies of the three applications currently supported by Odyssey influence energy consumption. A small step in this direction is the preliminary investigation of the video application mentioned in the previous section. Initially, our investigations will be on individual applications in isolation. Later, we will investigate the effects of concurrently executing these applications.

This study of adaptive applications is likely to lead to a number of ideas for adaptation strategies specifically tuned to energy conservation. We will implement these ideas and investigate their merits using PowerScope. As our work proceeds, the richness of the problem space defined by our three current applications may prove inadequate. We may therefore have to identify other applications, implement them on Odyssey, and investigate the impact of their adaptive strategies on energy consumption.

Another area of investigation will be to understand the interplay between low-level hardware techniques (such as slowing processor speed) and higher level software techniques (such as lowering fidelity). There are situations in which uncoordinated use of these techniques can be counter-productive. We would like to understand how these occur, and how to avoid them. This will involve conducting experiments with adaptive Odyssey applications on a wide range of diverse mobile hardware.

As our research proceeds, we are likely to expose inadequacies with the current PowerScope prototype that limit further progress. For example, our sampling rate may be too coarse to discriminate between alternative hypotheses to explain an observed phenomenon. In such cases, we will refine and improve our instrumentation. This may sometimes require more sophisticated hardware, but will more often involve modifications to the PowerScope software. For example, we may decouple sampling from the clock interrupt of the profiling computer and instead use hardware performance counters (such as those on the Intel Pentium Pro) to generate sampling interrupts. As another example, the output of PowerScope may aggregate data too much; in that case, we will modify PowerScope to generate output at finer granularity (such as detailed histograms and time series representations).

A related area of investigation will be to measure the loading of PowerScope on the system being studied. Although small, there is inevitably some energy consumed by PowerScope components themselves during the data collection phase. This may distort measurements involving highly energy-efficient applications. Reducing this overhead to a minimum, and quantifying it so that measurements can be corrected, will be important.

Continuous energy profiling, along the lines used for CPU profiling by Anderson et al [1], is another potential area of investigation. By monitoring the energy usage of systems in real use, one may gain valuable insights. This will require careful attention to the data collection component of PowerScope, to ensure that it is indeed capable of transparent, low-overhead monitoring over extended periods of time.

C.5 Enabling Controlled Experiments with Motion

C.5.1 Conceptual Background

Studying and interpreting the behavior of a mobile computing system under conditions of motion is difficult. We therefore propose an experimental infrastructure called a *wireless obstacle course* that facilitates such evaluations. This infrastructure consists of a physically compact, limited-access region over which the quality of wireless network performance can be precisely controlled. As a test system moves through the obstacle course, it perceives wireless network quality that varies spatially according to a pattern set up by the experimenter. One can thus re-create in a microcosm, conditions that a mobile system would encounter in the real world. For good experimental control, the obstacle course typically consists of a closed indoor region from which extraneous wireless transmissions and moving objects can be excluded during experiments.

We envision the role of this infrastructure in mobile computing as analogous to that of wind tunnels in aviation research. Like a wind tunnel, a wireless obstacle course offers realism, reproducibility and flexibility. Within this synthetic environment, a mobile system can be subjected to a wide range of wireless network conditions as it moves. The path and speed of motion may sometimes be unconstrained, as when a human user in an augmented-reality experiment wanders at will through the obstacle course. In other cases, such as a robot traversing the course, path and speed may be carefully controlled. The frequency and intensity of conditions that occur sporadically in normal operation can be exaggerated in the obstacle course to enable debugging, stress testing and benchmarking. Of course, as with a wind tunnel, any given obstacle course will be limited in the range of conditions that it can faithfully reproduce; explorations beyond these limits will require a different obstacle course.

C.5.2 Research Status

We have recently implemented an early prototype of a wireless obstacle course. Building this prototype required us to solve three key problems:

- controlling wireless network quality predictably and accurately.
- knowing one’s position within the obstacle course.
- flexibly mapping position to network quality, reflecting any desired spatial distribution.

Since our research focuses on the transport protocol and higher system layers, we have used the recently-developed technique of *trace modulation* [17, 11] to control network quality. This technique for varying end-to-end network quality based on delay and loss parameters consists of three phases. The first phase captures bandwidth, latency and loss characteristics from a traversal of a real wireless network under a known workload. Using a simple network model, the second phase transforms these measurements into a time-varying sequence of parameter values called a *replay trace.* A replay trace can also be constructed synthetically, by direct generation of parameter values to reflect the properties of a hypothetical network. Once a replay trace is obtained, it can be used in the third phase to modulate the loss and delay behavior of a wired or wireless LAN. During modulation, unmodified application and system software at the transport and higher levels perceive time-varying network quality faithful to that seen by the moving host during the first phase. In our prototype, we use the AT&T WaveLan wireless network with a nominal bandwidth of 2 Mb/s as the underlying network. Modulation based on synthetic traces is performed on this network to yield the degraded wireless network quality used in experiments.

For location sensing, we initially considered using GPS because of its popularity and widespread availability of hardware. But it soon became clear that GPS was a poor choice for use in obstacle courses. Its primary drawback is that it tends to perform poorly indoors, where we expect most obstacle courses to be built for reasons of experimental control and weather protection.

We have therefore adapted a mechanism originally developed for a different purpose. The concept of an *active badge* was originally invented by Olivetti Research to easily track the movement of entities such as people or packages in a building [22, 18]. An active badge consists of a lightweight, compact infra-red transmitter that is affixed to the object being tracked. Each badge periodically emits a signal encoding its unique ID. Sensors situated at known locations in a building detect badges that come within range, and decode their identities. Tracking software on a central machine collects information from these sensors via a wired network, integrates the information, and displays it on a GUI or logs it in an audit trail.

Our adaptation consists of reversing the roles of badge and sensor: we place active badges at different points of the obstacle course, and fit the mobile system under test with a sensor. As the mobile system moves, its sensor detects active badges as they come in range. The IDs of the active badges that are currently visible allow the mobile system to infer its location.

This simple adaptation of the active badge concept is well suited to our purpose. Since active badges were designed for indoor use, they perform well in that context. Reconfiguration of an obstacle course is simple because it merely involves changing the locations of easy-to-move active badges. In our current prototype, the badges are attached with Velcro to the ceilings of corridors and offices. Setup and teardown of a system involving 12 badges can be done in just a few minutes.

The third component in our design is a Java-based GUI tool for configuring the obstacle course. Using this tool, an experimenter paints a map of an obstacle course with different colors, each representing a different bandwidth. The output produced by the tool correlates location information with network quality, and is used as input to trace modulation. Figure 4 shows a screenshot from the tool.

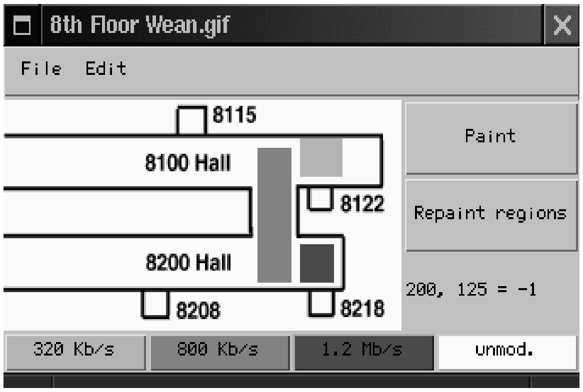


Figure 4: Typical Display of GUI Tool for Wireless Obstacle Course

We have used this prototype to conduct an initial exploration of the behavior of Odyssey in actual motion. The conditions re-created for this exploration anticipate those likely to be experienced by a pedestrian using a wearable computer in an urban setting with wireless overlay network coverage [7]. Our initial experience confirms the value of a wireless obstacle course for experimentation, but also exposes important limitations of our prototype. Specifically, our measurements indicate that the hardware we use for indoor location sensing is only adequate for very slow motion. Substantial improvement is therefore needed to extend our prototype's range of usefulness.

C.5.3 Future Plans

Our first task in this area is to improve the ability of our prototype obstacle course to operate at significantly higher speeds. Initial experiments indicate that the factor limiting speed currently is the beaconing frequency of the active badges used in our prototype. Increasing this frequency is nontrivial, because the badges have no external controls on them. An approach we plan to explore is to use multiple badges with identical but offset periods to emulate a single badge of higher frequency. An alternative approach is to collaborate with Olivetti, the badge manufacturer, to implement frequency control on the badges. A third approach is to explore use of a different indoor location sensing mechanism such as the recently-announced RF-based PinPoint *3D-iD* system [24].

Once we are able to operate the obstacle course at typical human walking speeds, we will be able to conduct Odyssey experiments that have enhanced credibility because they provide a realistic user experience. The focus of these experiments will be on understanding how the adaptation provided by Odyssey applications is perceived by the user. For example, does a particular adaptation strategy provide a perceived performance improvement, not just a measured one? Is the adaptation too jarring, or could it be more aggressive? What kinds of control does the user want in practice? How smoothly does the system support truly mobile usage?

A related set of experiments will pertain to adaptation policy in the presence of multiple applications. The current resource arbitration policy in Odyssey is hardwired into the system. While acceptable in some situations, we expect there will be many cases where a user will want control over which application gets the lion's share of currently available resources. For example, if a user is viewing two videos concurrently he may have a strong preference regarding which one is more important, and hence should be degraded less when bandwidth drops. What kinds of controls should Odyssey provide to enable such preferences to be expressed? Should they be application-specific or system-wide? These are the kinds of questions we hope to answer. Once again, the ability of a wireless obstacle course to provide a realistic user experience will guide our exploration in fruitful directions and help ensure the credibility of our results.

In the current prototype, we use synthetic traces as input to the trace modulation layer. We would like to obtain more realistic wireless traces both through empirical data collection and through the use of detailed RF propagation models. Greater realism in the traces will enable us to conduct experiments where the conditions re-created in the obstacle course are closer to those observed in real mobile computing environments.

Although trace modulation is a valuable tool for controlling network quality, it is not of universal applicability. Because it operates at the IP layer of the protocol stack, trace modulation only influences the network characteristics seen by higher layers. It is therefore not an appropriate tool if a wireless obstacle course is to be used for studying the impact of mobility on the data link or physical layers of a system. To build an obstacle course suitable for such studies, we plan to explore the conditioning of wireless performance by influencing physical propagation of RF signals through careful placement of transmitters, reflectors and absorbers to model a desired environment.

C.6 Quantifying the Behavior of Adaptative Systems

C.6.1 Conceptual Background

Supporting multiple levels of fidelity complicates the task of evaluation. Since adaptive applications trade fidelity of data for performance, focusing solely on the latter can result in a misleading evaluation. For example, by forcing applications to operate at their lowest fidelity levels at all times, a system could ensure better performance than a competing system that strives to support higher fidelity levels when possible. Yet, this degenerate approach violates our intuitive notion of what constitutes a good system. Hence, the evaluation of an adaptive system must take into account both fidelity and performance.

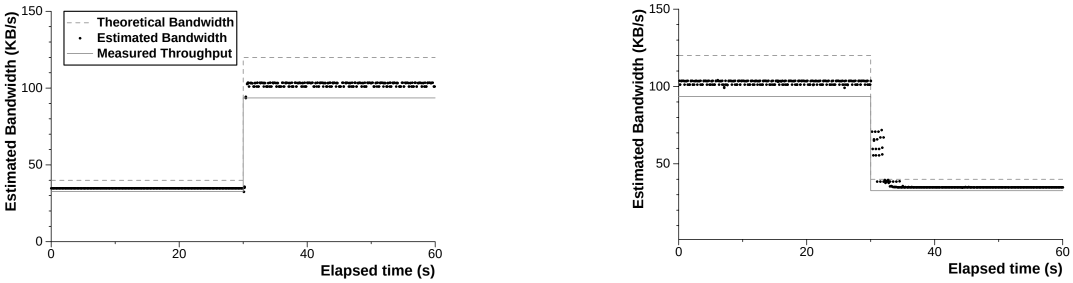
Sound adaptation decisions require accurate and timely knowledge of resource availability. Ideally, a mobile client should have perfect knowledge of current resource levels and should be able respond instantaneously to relevant changes. Of course, no physical system can meet this ideal — at best, we can hope for good approximations. A key attribute of an adaptive system, called its *agility,* is the speed and accuracy with which it detects and responds to changes in resource availability.

Agility is a complex property with many components. One source of complexity is differing sensitivity to different resources. For example, a system may be much more sensitive to changes in network bandwidth than to changes in battery power level. Another source of complexity is differing origins of changes in resource availability. The change may be caused by variation in the *supply* of a resource due to mobility, or by changed *demand* for it by concurrent applications. Since different mechanisms may be involved in detecting these two different types of changes, it may be necessary to distinguish these components of agility.

One approach to quantifying adaptive ability is to draw upon well-established principles for measuring dynamic response from the field of *control systems* [4, 16]. The accepted practice there is to subject a test system to resource level changes corresponding a set of input *reference waveforms*, and to observe how closely the system is able to track them. Each reference waveform is conceptually simple, yet greatly stresses the adaptive ability of the system by varying the input in some sharp and substantial manner. Although reference waveforms are idealized, approximations to them can occur in practice. *Step* waveforms can arise in overlay networks, where a mobile client may seamlessly switch between different network interfaces. Further, *virtual radios* such as SpectrumWare [19] may allow sharp bandwidth degradation. *Impulse* waveforms can arise as a result of frequent transitions in either of these situations, or in the presence of bursty background traffic.

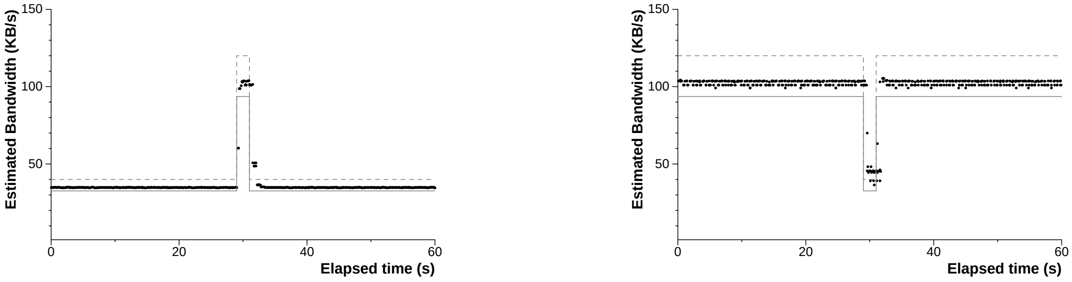
C.6.2 Research Status

We have conducted an initial set of experiments to evaluate Odyssey's agility with respect to variation in network bandwidth. This evaluation is based on trace modulation using synthetic traces corresponding to reference waveforms. Figure 5 illustrates the reference waveforms and the corresponding contour of bandwidth estimates by Odyssey. The high and low bandwidths in these experiments were 120 KB/s (kilobytes per second) and 40 KB/s.



(a) Step Up

(b) Step Down



(c) Impulse Up

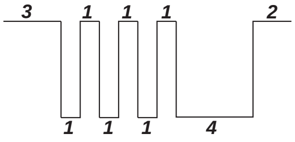
(d) Impulse Down

This figure shows the agility of Odyssey's bandwidth estimation in the face of varying supply. Each graph merges the results from five trials, and each bandwidth observation is represented by a single dot on the graph. The dashed lines represent the theoretical bandwidth of the emulated network, as specified by the synthetic traces used for emulation. The dotted lines are the measured, instantaneous throughputs obtained using a large bulk transfer between client and server. Ideally, all samples would lie between the dashed and dotted lines. To ensure that the system was in steady state, we primed it for thirty seconds prior to observation.

Figure 5: Agility of Odyssey on Step and Impulse Trace Modulation Waveforms

As Figure 5(a) shows, Odyssey has excellent agility on the Step-Up waveform. The second graph, Figure 5(b), shows that agility on the Step-Down waveform is not quite as good as on Step-Up. The *settling time* for this waveform, which is the time required to reach and stay within the nominal bandwidth range, is 2.0 seconds. The slower downward transition is caused by the fact that the current implementation of Odyssey generates a throughput estimate only at the end of a window of data transmission. If bandwidth falls abruptly during a large window, the drop is not noted until the last packet of the window. Figures 5(c) and 5(d) show agility for the Impulse-Up and Impulse-Down waveforms. The leading edge of the upward impulse is accurately traced, but the trailing edge has a noticeable settling time. The downward impulse is too short for estimation to settle accurately, and there is again a noticeable settling time on the trailing edge of the impulse.

To investigate the adaptive behavior of Odyssey under more complex conditions, we have conducted an experiment in which the three adaptive applications described in Section C.3 are concurrently executed on an Odyssey client while bandwidth is varied using the trace modulation waveform shown in Figure 6(a). This 15-minute synthetic trace is intended to model the bandwidth variation seen by a user walking through a hypothetical urban setting. The user begins well-connected but soon enters a region of intermittent quality. He then enters the radio shadow caused by a large building, and finally returns to good connectivity.



	Video		Web		Speech
	Drops	Fidelity	Seconds	Fidelity	Seconds
Odyssey	1018 (48.6)	0.25 (0.00)	0.54 (0.02)	0.47 (0.01)	1.00 (0.01)
Laissez-Faire	2249 (80.2)	0.39 (0.01)	0.95 (0.03)	0.93 (0.01)	1.21 (0.01)
Blind-Optimism	5320 (23.3)	0.80 (0.00)	1.20 (0.00)	1.00 (0.00)	1.26 (0.02)

(a) Trace Modulation Waveform

(b) Measured Behavior of Concurrent Applications

Part (a) of this figure shows the synthetic trace modulation waveform used for bandwidth variation. Each number indicates the time duration of the corresponding segment in minutes. The high and low bandwidths are 120 KB/s and 40 KB/s respectively. Part (b) of this figure shows the average fidelity and performance of each application, when all three are concurrently executed on Odyssey, using coordinated resource management by the viceroy. For comparison, the figure also shows the fidelity and performance for two uncoordinated resource management strategies, labelled *laissez-faire* and *Blind Optimism.* Each observation in this table is the mean of five trials, with standard deviations given in parentheses.

Figure 6: Example Fidelity versus Performance Tradeoff Experiment in Odyssey

The results of this experiment, shown in Figure 6(b), demonstrate the importance of centralized resource management in Odyssey by comparing it to two uncoordinated resource management strategies called *laissez-faire* and *Blind Optimism.* Details of these alternative strategies, as well as the workloads used for each application, can be found elsewhere [12]. The message of Figure 6 is that Odyssey's centralized resource estimation and management provides significant benefits over the two uncoordinated adaptation strategies examined. By correctly accounting for bandwidth competition, the Odyssey Web browser and video player fetch data at lower fidelity, thus enabling all applications to come much closer to their performance goals. Odyssey drops a factor of 2 to 5 fewer frames than the other strategies, and Web pages are loaded and displayed roughly twice as fast. The resulting decrease in network utilization improves speech recognition time as well.

C.6.3 Future Plans

The experiments described in the previous section are only the starting point of an in-depth investigation of the adaptive properties of Odyssey. Our first priority will be to conduct a sensitivity analysis of the experiments shown in Figure 6. These results are based on a single trace modulation waveform and a single set of workloads for each of of the three Odyssey applications. Varying these parameters will help us understand the robustness of the results. This will involve design of new waveforms, design of new workloads, and conducting a series of experiments similar to that shown in Figure 6. We also plan to explore the sensitivity of these results to alternative centralized resource allocation strategies in Odyssey.

The assignment of a fidelity value between 0 and 1.0 to a particular level of data degradation is currently ad hoc — the only requirement is that fidelity be monotonically increasing as the quality of presented data increases. Although this requirement is sufficient for making simple comparisons, it is inadequate for expressing magnitudes of differences. We would like to eliminate this limitation by investigating more rigorous approaches to assigning fidelity values. In many cases, this will require us to map fidelity values to perceptions of data quality by users. The wireless obstacle course experiments described in Section C.5 can be helpful in obtaining this information. Those experiments are also likely to help us characterize and quantify the property of *stability,* which is the complement of agility. Striking the right balance between agility and stability requires the appropriate level of hysteresis to be provided in Odyssey as well in the adaptative strategies of each of its applications.

Trace modulation has proved invaluable in evaluations of Odyssey with respect to bandwidth adaptation. We plan to explore the possibility of extending this idea to evaluations involving energy. The goal is to provide Odyssey and its applications with information about available energy that is controllable by an experimenter. This will, for example, allow an experimenter to study the behavior of Odyssey when battery level drops low, without having to physically drain the battery to that level. This has the dual merits of better experimental control and faster running of experiments. Once this capability is available, and Odyssey has been extended to provide energy adaptation, we can conduct experiments involving simultaneous variation of both bandwidth and energy. Understanding how to characterize and report such multivariate adaptation will be an important part of our future work.

From a conceptual point of view, we plan to further explore what can be borrowed from the field of control systems. We have been able to use some simple concepts so far, but suspect that many more tools, concepts and analytical techniques can be adopted. Of course, these may need to be modified before they are applicable to our domain. Understanding the nature of those modifications and working toward a deeper understanding of the applicability of control systems concepts to evaluation of adaptive mobile systems are important long-term goals.

C.7 Expected Impact

The design of adaptive mobile systems is a seat-of-the-pants activity today. We do not yet have a well-validated and widely-accepted set of design principles for designing such systems. Getting to the point where we are able to enunciate such principles with confidence requires us to first critically evaluate alternative designs. The research described in this proposal directly addresses this issue. By making it possible to conduct controlled and reproducible experiments, consistent with sound scientific practice, we will be able to critically examine the merits of alternative design and implementation strategies for adaptive systems. That, in turn, will eventually lead to a body of knowledge on which the designers of adaptive mobile systems can rely.

We foresee the work proposed here contributing many important results toward this overarching goal. Making energy a first-class resource is critical for the next generation of mobile systems. Being able to profile and analyze the energy usage of applications will make it possible to develop suitable adaptation strategies for coping with finite battery life. Enabling researchers to conduct experiments involving physical motion will facilitate scientific studies of the effects of adaptive strategies on the usability of mobile computing sytems. Finally, the ability to quantify the multiple facets of adaptation without oversimplification will make it possible to develop benchmarks that allow well-grounded comparisons between alternative system designs. The ensuing ability to quantitatively demonstrate improvements will be an important catalyst for progress in the field.

C.8 Results from Prior NSF Support

The NSF award most relevant to this proposal is the Presidential Young Investigator Award entitled "The Sharing of Data in Distributed Environments". The NSF Award Number was CCR-8657907, the amount was $312,500 and its period was 7/1/1987 to 12/31/1992.

Note: a more recent award on which M. Satyanarayanan was a co-PI is DMI-9527190 entitled "HPCC: A Distributed Architecture for Rapidly Reconfigurable Assembly Systems", with Dr. Ralph Hollis as PI. This award is for research in manufacturing, a topic far removed from that of the current proposal. We have therefore provided details on the earlier award, which is more relevant to the current proposal.

C.8.1 Summary of Results

The goal of this research was to investigate fundamental issues in the sharing of data in distributed environments. The work was done in the context of distributed file systems. There were two specific focus areas: *scalability* and *resiliency to failures.*

The first issue, scalability, was addressed in the context of the Andrew File System. We compared the elapsed time as well as server CPU and disk usage of AFS and the Sun Network File System (NFS) using controlled experiments. Our results showed that the *callback* based cache coherence mechanism of AFS dramatically improves scalability relative to the *check on open* cache coherence scheme of NFS.

The second issue, failure resiliency, was addressed in the context of the Coda File System which is a descendant of AFS. Coda uses two distinct but complementary mechanisms to achieve resiliency: *server replication* and *disconnected operation.* Our work demonstrated that the performance cost of server replication could be kept within acceptable bounds by the use of a parallel RPC and file tranfer protocol. We examined the importance of hardware multicast support and established that its primary value was in reducing network load rather than latency. We also established that a log-based strategy for resolving directory conflicts imposed acceptable space and time costs. Finally, we demonstrated that disconnected operation based on an optimistic replication strategy and implemented using a tri-state model of *hoarding, emulation,* and *reintegration* is viable and effective.

C.8.2 Resulting Publications and Research Products

The research conducted under this award has resulted in many publications as well as in software that has been distributed outside Carnegie Mellon. The publications are listed below. The software corresponds to the source code for the Coda File System. This code has been widely distributed to numerous universities and research groups. Source code distribution is being provided entirely free: there are neither licensing or royalty costs, nor is there any distribution fee.

Howard, J.H., Kazar, M.L., Menees, S.G., Nichols, D.A., Satyanarayanan, M., Sidebotham, R.N., West, M.J.
Scale and Performance in a Distributed File System.
ACM Transactions on Computer Systems 6(1), February, 1988.

Kistler, J.J., Satyanarayanan, M.
Disconnected Operation in the Coda File System.
ACM Transactions on Computer Systems 10(1), February, 1992.

Kistler, J.J.
Disconnected Operation in a Distributed File System.
PhD thesis, Department of Computer Science, Carnegie Mellon University, May, 1993.

Kumar, P., Satyanarayanan, M.
Log-based Directory Resolution in the Coda File System.
In Proc. of the Second International Conference on Parallel and Distributed Information Systems.
San Diego, CA, Jan., 1993.

Lorence, M., Satyanarayanan, M.
IPWatch: A Tool for Monitoring Network Locality.
In Proc. of the 4th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, Mallorca. September, 1988.

Satyanarayanan, M.
Integrating Security in a Large Distributed System.
ACM Transactions on Computer Systems 7(3), August, 1989.

Satyanarayanan, M.
A Survey of Distributed File Systems.
In Traub, J.F., Grosz, B., Lampson, B., Nilsson, N.J. (editors), Annual Review of Computer Science., 1989.

Satyanarayanan, M., Kistler, J.J., Kumar, P., Okasaki, M.E., Siegel, E.H., Steere, D.C.
Coda: A Highly Available File System for a Distributed Workstation Environment.
IEEE Transactions on Computers 39(4), April, 1990.

Satyanarayanan, M.
Scalable, Secure, and Highly Available Distributed File Access.
IEEE Computer 23(5), May, 1990.

Satyanarayanan, M., Siegel, E.H.
Parallel Communication in a Large Distributed Environment.
IEEE Transactions on Computers 39(3), March, 1990.

Satyanarayanan, M.
An Agenda for Research in Large-Scale Distributed Data Repositories.
In Karshmer, A., Nehmer, J. (editors), Operating Systems of the 90s and Beyond. Springer-Verlag, 1991.

Satyanarayanan, M.
The Influence of Scale on Distributed File System Design.
IEEE Transactions on Software Engineering 18(1), January, 1992.

Section D: References Cited

[1] Anderson, J.M., Berc, L.M., Dean, J., Ghemawat, S., Henzinger, M.R., Leung, Shun-Tak A., Sites, R. L., Vandevoorde, M. T., Waldspurger, C.A., Weihl, W. E.
Continuous Profiling: Where Have All the Cycles Gone?
In *Proceedings of the 16th ACM Symposium on Operating Systems and Principles*. Saint-Malo, France, October, 1997.

[2] Balakrishnan, H., Seshan, S., Amir, E., Katz, R.H.
Improving TCP/IP Performance Over Wireless Networks.
In *Proceedings of the First Annual International Conference on Mobile Computing and Networking*. Berkeley, CA, November, 1995.

[3] Caceres, R., Iftode, L.
Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments.
IEEE Journal of Selected Areas in Communications 13(5), June, 1995.

[4] Dorf, R.C. (editor).
The Electrical Engineering Handbook.
CRC Press, 1993.

[5] Fox, A., Gribble, S.D., Brewer, E.A., Amir, E.
Adapting to Network and Client Variability via On-Demand Dynamic Distillation.
In *Proceedings of the Seventh International ACM Conference on Architectural Support for Programming Languages and Operating Systems*. Cambridge, MA, October, 1996.

[6] Joseph, A.D., deLespinasse, A.F., Tauber, J.A., Gifford, D.K., Kaashoek, M.F.
Rover: A Toolkit for Mobile Information Access.
In *Proceedings of the 15th ACM Symposium on Operating System Principles*. Copper Mountain, CO, December, 1995.

[7] Katz, R. H., Brewer, E. A.
The Case for Wireless Overlay Networks.
In *SPIE Multimedia and Networking Conference*. January, 1996.

[8] Lorch, J.R., Smith, A.J.
Software Strategies for Portable Computer Energy Management.
IEEE Personal Communications 5(3), June, 1998.

[9] LoVerso, J.R., Mazer, M.S.
Caubweb: Detaching the Web with Tcl.
In *Proceedings of the Fifth Annual Tcl/Tk Workshop*. Boston, MA, July, 1997.

[10] Mummert, L.B., Ebling, M.R., Satyanarayanan, M.
Exploiting Weak Connectivity for Mobile File Access.
In *Proceedings of the 15th ACM Symposium on Operating System Principles*. Copper Mountain, CO, December, 1995.

[11] Noble, B. D., Satyanarayanan, M., Nguyen, G. T., Katz, R. H.
Trace-Based Mobile Network Emulation.
In *Proceedings of ACM SIGCOMM'97 Conference*. Cannes, France, September, 1997.

[12] Noble, B.D., Satyanarayanan, M., Narayanan, D., Tilton, J.E., Flinn, J., Walker, K.R.
Agile, Application-Aware Adaptation for Mobility.
In *Proceedings of the 16th ACM Symposium on Operating Systems Principles*. St. Malo, France, October, 1997.

[13] *Energy-Efficient Technologies for the Dismounted Soldier*
Board on Army Science and Technology, National Research Council, 1997.

[14] Rajagopalan, S., Badrinath, B.R.
An Adaptive Location Management Strategy for Mobile IP.
In *Proceedings of the First Annual International Conference on Mobile Computing and Networking*. Berkeley, CA, November, 1995.

[15] Reiher, P., Popek, J., Gunter, M., Salomone, J., Ratner, D.
Peer-to-peer Reconciliation based Replication for Mobile Computers.
In *Proceedings of the ECOOP Workshop on Mobility and Replication*. July, 1996.

[16] Rubin, O.
The Design of Automatic Control Systems.
Artech House, Norwood, MA, 1986.

[17] Satyanarayanan, M., Noble, B.
The Role of Trace Modulation in Building Mobile Computing Systems.
In *Proceedings of the Sixth IEEE Workshop on Hot Topics in Operating Systems*. Chatham, MA, May, 1997.

[18] Spreitzer, M., Theimer, M.
Providing Location Information in a Ubiquitous Computing Enviornment.
In *Proceedings of the 14th ACM Symposium on Operating Systems Principles*. Asheville, NC, December, 1993.

[19] Tennenhouse, D.L., Turletti, T., Bose, V.G. .
The SpectrumWare Testbed for ATM-based Software Radios.
In *IEEE International Conference on Universal Personal Communications*. September, 1996.

[20] Terry, D.B.,Theimer, M.M., Petersen, K., Demers, A.J.
Managing Update Conflicts in Bayou, a Weakly Connected Replicated Storage System.
In *Proceedings of the 15th ACM Symposium on Operating System Principles*. Copper Mountain, CO, December, 1995.

[21] Waibel, A.
Interactive Translation of Conversational Speech.
IEEE Computer 29(7), July, 1996.

[22] Want, R., Hopper, A., Falcao, V., Gibbons, J.
The Active Badge Location System.
ACM Transactions on Information Systems 10(1), January, 1992.

[23] Weiser, M., Welch, B., Demers, A., Shenker, S.
Scheduling for Reduced CPU Energy.
In *Proceedings of the First USENIX Symposium on Operating System Design and Implementation (OSDI)*. Monterey, CA, November, 1994.

[24] Werb, J., Lanzl, C.
Designing a Positioning System for Finding Things and People Indoors.
IEEE Spectrum 35(9), September, 1998.

Section E: Biographical Sketch

E.1 Vita

Address

Professor Mahadev Satyanarayanan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Phone: (412)-268-3743
Fax: (412)-268-4136
Email: satya@cs.cmu.edu

Education

PhD in Computer Science, March 1983,
Carnegie Mellon University, Pittsburgh, PA.

Master of Technology in Computer Science, First Class with Distinction, July 1977,
Indian Institute of Technology, Madras, India.

Bachelor of Technology in Electrical Engineering (Electronics), First Class with Distinction, July 1975,
Indian Institute of Technology, Madras, India.

Employment

Carnegie Mellon University, School of Computer Science, Carnegie Group Professor, since May 1997.

Carnegie Mellon University, School of Computer Science, Professor, July 1995 to May 1997.

Carnegie Mellon University, School of Computer Science, Associate Professor, July 1989 to July 1995.

Carnegie Mellon University, Department of Computer Science, Assistant Professor, Sep. 1986 to Sep. 1989.

Carnegie Mellon University, Information Technology Center, System Designer, April 1983 to August 1986.

Carnegie Mellon University, Department of Computer Science, Research Assistant, Sep. 1977 to March 1983.

Awards and Honors

Herbert A. Simon Award for Teaching Excellence, Carnegie Mellon University, April 1998.

Allen Newell Medal for Research Excellence, Carnegie Mellon University, March 1997.

Outstanding Paper, 14th ACM Symposium on Operating Systems Principles, 1993.

Outstanding Paper, 13th ACM Symposium on Operating Systems Principles, 1991.

Outstanding Paper, 11th ACM Symposium on Operating Systems Principles, 1987.

Presidential Young Investigator Award, National Science Foundation, January 1987.

Faculty Development Award, IBM Corporation, 1987.

Outstanding Paper, 10th ACM Symposium on Operating Systems Principles, 1985.

Certificate of Academic Distinction, Indian Institute of Technology, Madras, September 1977.

E.2 Relevant Publications

Noble, B.D., Satyanarayanan, M., Narayanan, D., Tilton, J.E., Flinn, J., Walker, K.R.
Agile, Application-Aware Adaptation for Mobility.
In Proc. of the 16th ACM Symposium on Operating Systems Principles, St. Malo, France, October, 1997.

Noble, B. D., Satyanarayanan, M., Nguyen, G. T., Katz, R. H.
Trace-Based Mobile Network Emulation.
In Proceedings of ACM SIGCOMM'97 Conference, Cannes, France, September, 1997.

Satyanarayanan, M.
Mobile Information Access.
IEEE Personal Communications 3(1), February, 1996.

Lu, Q., Satyanarayanan, M.
Resource Conservation in a Mobile Transaction System.
IEEE Transactions on Computers 46(3), March, 1997.

Satyanarayanan, M.
Fundamental Challenges in Mobile Computing.
In Proc. of the Fifteenth ACM Symposium on Principles of Distributed Computing, Philadelphia, PA, May, 1996.

Mummert, L.B., Ebling, M.R., Satyanarayanan, M.
Exploiting Weak Connectivity for Mobile File Access.
In Proc. of the 15th ACM Symposium on Operating System Principles, Copper Mountain, CO, Dec., 1995.

Kistler, J.J., Satyanarayanan, M.
Disconnected Operation in the Coda File System.
ACM Transactions on Computer Systems 10(1), February, 1992.

E.3 Collaborators within Last 48 Months

Baseil, R. (Bellcore)
Ebling, Maria (IBM Research)
Hollis, Ralph (Carnegie Mellon University)
Katz, Randy (University of California, Berkeley)
Kistler, Jay (Fore Systems, Inc.)
Kumar, Puneet (Compaq Systems Research Center)
Lewin, B.R. (Bellcore)
Lu, Qi (Yahoo, Inc.)
Mummert, Lily (IBM Research)
Nguyen, G. (University of California, Berkeley)
Noble, Brian (University of Michigan)
Spasojevic, Mirjana (HP Labs, Palo Alto)
Steere, David (Oregon Graduate Institute)
Tripathi, Satish (University of California, Riverside)

E.4 Advisor/Mentor Relationships

Total number of PhD students graduated: 7

Ebling, Maria (IBM Research)
Huizinga, Dorota (California State Univesity, Fullerton)
Inamura, Hiroshi (NTT, Inc.)
Kistler, Jay (Fore Systems, Inc.)
Kumar, Puneet (Compaq Systems Research Center)
Lu, Qi (Yahoo, Inc.)
Masashi, Kudo (NEC, Inc.)
Muranaga, Tetsuro (Toshiba, Inc.)
Mummert, Lily (IBM Research)
Noble, Brian (University of Michigan)
Steere, David (Oregon Graduate Institute)

E.5 PI’s PhD Co-Advisors

Professor William A. Wulf (University of Virginia)
Dr. George G. Robertson (Microsoft Research)