CoolSpots: Reducing the Power Consumption of Wireless Mobile Devices with Multiple Radio Interfaces

Trevor Pering
Intel Research
2200 Mission College Blvd.
Santa Clara, CA 95054
trevor.pering@intel.com

Yuvraj Agarwal
CSE, UC San Diego
9500 Gilman Drive
La Jolla, CA 92093
yuvraj@cs.ucsd.edu

Rajesh Gupta
CSE, UC San Diego
9500 Gilman Drive
La Jolla, CA 92093
gupta@cs.ucsd.edu

Roy Want
Intel Research
2200 Mission College Blvd.
Santa Clara, CA 95054
roy.want@intel.com

Abstract

CoolSpots enable a wireless mobile device to automatically switch between multiple radio interfaces, such as WiFi and Bluetooth, in order to increase battery lifetime. The main contribution of this work is an exploration of the policies that enable a system to switch among these interfaces, each with diverse radio characteristics and different ranges, in order to save power – supported by detailed quantitative measurements. The system and policies do not require any changes to the mobile applications themselves, and changes required to existing infrastructure are minimal. Results are reported for a suite of commonly used applications, such as file transfer, web browsing, and streaming media, across a range of operating conditions. Experimental validation of the CoolSpot system on a mobile research platform shows substantial energy savings: more than a 50% reduction in energy consumption of the wireless subsystem is possible, with an associated increase in the effective battery lifetime.

Categories and Subject Descriptors

C.2.2 [COMPUTER-COMMUNICATION NETWORKS]: Network Operations – *network management, network monitoring.*

General Terms

Algorithms, Measurement, Design

Keywords

Multiple-radio systems, low-power systems, handheld devices, wireless communication, pervasive computing.
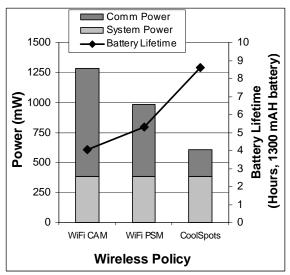
Figure 1: Estimated impact of optimizing the communication subsystem power on total system power consumption and battery lifetime.

1 Introduction

The utility of mobile devices is directly impacted by their operating lifetime before on-board batteries need to be replaced or recharged. In advanced mobile computing platforms such as PDAs and smart-phones, the wireless communication subsystem accounts for a major component of the total power consumption [1][10] due to the communication centric usage of these devices. Furthermore, these platforms are increasingly being equipped with multiple radio interfaces to handle a variety of connections, ranging from Bluetooth for personal-area links, WiFi for local-area connectivity, and GPRS for wide-area data access.

Previous research has explored the idea of switching among multiple radio interfaces in an attempt to reduce overall power consumption: By using the appropriate wireless interface for the current application workload, and keeping the others effectively turned off, the system can save energy. For example, for applications with low network-utilization the low-power/low-bandwidth interface can be used, and the system can dynamically switch to the high-power/high-bandwidth interface when necessary. CoolSpots explores the *policies* necessary to enable switch-

ing among radio interfaces, in order to decrease overall power consumption while still maintaining enough bandwidth to support active applications.

Figure 1 shows an estimate, based on the experimental setup described below, of the effects of reducing the wireless power consumption for a mobile device, both in terms of total power consumption and resulting increase in battery lifetime. Measurements of the CoolSpot system and the associated policies show that more than a 50% reduction in energy consumption by the wireless subsystem is possible, which can effectively double the system battery lifetime (depending on overall system behavior).

The chief contributions of this paper are the basic CoolSpots model, which demonstrates the advantages of using multiple radio interfaces considering their different transmission ranges, and a suite of policies that guide the automatic switching decision based on a variety of metrics. The CoolSpots system does not require application modification or extensive infrastructure to implement, paving the way to easy incremental deployment and integration into existing wireless systems.

2   Mobile Computing Platforms

CoolSpots are motivated by the evolving PDA and smart phone cellular platforms, which are able to provide access to an ever expanding list of applications beyond basic communication and networking needs. These devices are now able to wirelessly download games, music and video, and play this multimedia content either locally, or remotely by wirelessly connecting to proximate environments such as the Digital Home [21][20]. This usage model is likely to play a more significant role in daily activities in future years, as both Microsoft and Intel roll-out the next generation of Windows Media Center Edition and Media Center PCs supporting multiple radio-interfaces such as WiFi and Bluetooth.

For the spectrum of applications that need to be considered, latency and bandwidth requirements vary significantly depending on the applications in use, e.g., from infrequent low-bandwidth control messages to high-bandwidth video streaming. Taking into account these varying bandwidth needs we explore techniques that dynamically reduce power consumption for mobile devices without compromising network connectivity to the local infrastructure, communication range or limit the peak bandwidth needs of applications.

Currently, there are two dominant short range wireless standards frequently incorporated into mobile devices: WiFi and Bluetooth. WiFi, or IEEE 802.11a/b/g offers high-bandwidth local-area coverage up to 100 meters. Bluetooth, intended for cell-phone class devices, is primarily a cable-replacement technology up to 10m and focuses on low-power consumption for handheld battery constrained devices. Emerging devices, such as HTC's Universal palmtop, the Nokia N80, and the Motorola CN620 al-
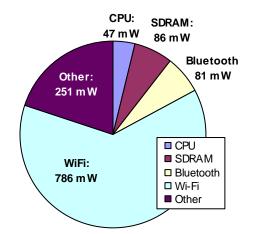

Figure 2: Power breakdown for a *connected* mobile device in idle mode. The wireless interfaces consume approximately 70% of the total power. Since the device is idle, the LCD and backlight are turned off – consuming zero power. Other includes power regulation and other smaller subsystems (such as LEDs).

ready possess *both* of these technologies, allowing them to interface with a wide variety of wireless networks and peer devices. The CoolSpots project coordinates the use of both radios to provide a single logical wireless channel with improved power, bandwidth, and range characteristics.

Wireless bandwidth is often the primary differentiator among wireless technologies, understandably because the bandwidth of wireless networks is much less than their wired counterparts. For example, the basic 802.11b WiFi standard offers a nominal 11 Mb/s symbol rate, compared with the 100 Mb/s or even 1 Gb/s rates offered by modern wired Ethernet technologies, while Bluetooth offers a bandwidth of only 1 Mb/s. The basic 802.11b data-rate has been sufficient to create a wave of popular interest that has, resulted in the deployment of over 100,000 WiFi access points worldwide.

Unfortunately, the downside of the WiFi design is its relatively high power-consumption, which in an active data transfer state is of the order of 890 mW, compared to only 120 mW for Bluetooth due to a limited range and a simpler radio architecture. For smaller devices such as cell-phones and PDAs, with limited power budgets, the power consumption of a WiFi radio represents a significant proportion of the overall system power [1][10][12], even for an idle system (Figure 2).

Even more significant than the active transmit power is the power consumed for a radio network in its idle state. Given typical use models, most wireless devices are only communicating for a small percentage of the time the device is actually on. Bluetooth is optimized to be in an extremely low-power state operating at only a 2% power duty-cycle, typically consuming on the order of 1 mW while still remaining available for device discovery and connection setup. By comparison WiFi is based on CSMA
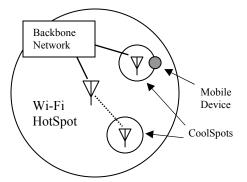
Figure 3: Multiple Bluetooth-enabled CoolSpots, inside of a traditional WiFi HotSpot, allow mobile devices to connect other devices through the backbone network. CoolSpots are connected to the backbone network either directly (wired) or through the WiFi network (wireless).
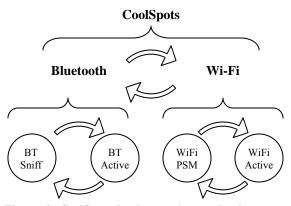

Figure 4: CoolSpots implement *inter*-technology power management on top of *intra*-technology techniques such as Bluetooth sniff mode and WiFi PSM.

and, although recent implementations support a power saving mode (PSM), the underlying design means that typical power consumption even though reduced is still close to 250 mW in this state.

Another significant difference between WiFi and Bluetooth mentioned earlier, is the expected communication ranges - being 100m for WiFi and 10m for Bluetooth. The two radios have different transmit powers to support communication at these distances. Effective ranges of these radios determine how and where a user can operate their wireless devices. Distance between communicating endpoints will affect a given radio technology's performance, both in terms of realized throughput, due to contention for bandwidth, and higher latencies due to increased channel noise, resulting in more retransmissions for reliable communication. Both of these effects contribute to the higher power consumption of WiFi networks, compared to Bluetooth networks which serve relatively smaller areas.

The CoolSpots model takes into account all three of these factors (bandwidth requirement, power, and distance), to determine the optimal radio configuration to use. For example, users (within the CoolSpot) can be provided with only the bandwidth necessary to support a particular appli-

Table 1: Power consumption for various wireless interfaces. Values marked with a * are measured values, while others are taken from datasheets.

Interface	Low-Power Idle	Active Tx
WiFi Cards		
Cisco PCM-350*	*390 mW*	*1600 mW*
Netgear MA701	264 mW	990 mW
Linksys WCF12*	*256 mW*	*890 mW*
Bluetooth		
BlueCore3	5.8 mW	81 mW
BlueCore3*	*25 mW*	*120 mW*

cation. The CoolSpots communication sub-system can then support low-bandwidth activity through Bluetooth alone and when required, switch to WiFi for high-bandwidth communication. In order to save idle power the WiFi radio can be turned off when not in use.

3   CoolSpots

CoolSpots uses a hybrid WiFi/BT system to provide improved communication capabilities when a mobile device is within a CoolSpot-enabled region (Figure 3). For example, if a home user is in their living room near the entertainment center, appropriately enabled as a CoolSpot, the mobile device would exhibit lower-power consumption while still supporting the desired application bandwidth. From a technology perspective, this means that the CoolSpot (in this case the entertainment center) is equipped with a Bluetooth radio, while the entire house is assumed to be covered by WiFi.

CoolSpots casts the generic concept of using two different classes of radios into a concrete implementation that seamlessly integrates with existing applications, systems, and devices. In essence, CoolSpots implement multi-radio power management that directly takes advantage of the diversity between different radio technologies (Figure 4). Based on the idle power consumption of typical WiFi and Bluetooth radios (Table 1), this concept has the potential to realize 10x reduction in power consumption for an idle system (from 256 mW down to 25 mW); however, the actual power savings depend on the implementation details, as explored by the CoolSpots implementation.

From a networking perspective, the CoolSpots model is a simple addition to common networking infrastructure and does not require any extensive hardware changes. A CoolSpots enabled basestation providing Bluetooth capability can simply be added into an existing WiFi network, allowing nearby mobile devices reduced-power operation: by employing network routing changes, this base station can claim and route traffic to the mobile device, a technique similar to [6]. Bluetooth was designed as a low-cost addition to handheld devices, and so would be relatively inexpensive to add into the environment. Although the CoolSpot prototype co-locates the Bluetooth and WiFi radios in a single access point for experimental evaluation, this is not a requirement and the system could also be con-

Table 2: CoolSpot policies with switch up/down criteria.

Policy	Switch Up	Switch Down	Comments
wifi-fixed	*N/A*	*N/A*	Only uses WiFi
bandwidth	Static Bandwidth Threshold	Static bandwidth	Can fail in bad network conditions
cap-static	Capacity Detection	Static bandwidth	Same as Bandwidth policy
cap-dynamic	Capacity Detection	Use Switch-*Up* Bandwidth	Handles all network conditions
bluetooth-fixed	*N/A*	*N/A*	Only uses Bluetooth

figured such that the Bluetooth and WiFi radios are physically separated. If a mobile device is not sufficiently near a CoolSpot, and wants to save power, it could fall back to using WiFi in PSM mode.

The Bluetooth radio standard defines several power saving states, the most applicable of which is the *sniff* mode. In this mode a connected BT radio sleeps with a specified duty cycle and then is automatically activated when there is any wireless data activity. Using this mode, it is possible to drop the power consumption of Bluetooth to very low-levels with an idle connection, while still being able to quickly transition to fully-active communication. Relying on this mode is quite effective at optimizing the lower-end of the power/bandwidth spectrum, but it is fundamentally limited by the maximum bandwidth offered by Bluetooth. Unfortunately, the Bluetooth setup (BlueCore3) used in our experiments does not enter the lowest-possible power-save mode (deep sleep) due to technical implementation details in the radio (the discrepancy can be seen in Table 1); however, utilizing the sniff mode provides the bulk of the power savings, and using the additional sleep-mode would only further enhance our results and the energy performance of CoolSpots.

WiFi uses a similar Power Save Mode (PSM) wherein the radio duty cycles between sleep and active states to reduce its power consumption. It is important to note that from a pure energy/bit standpoint, WiFi is more energy efficient than Bluetooth, which consume 81 nJ/bit and 120 nJ/bit, respectively, for the basic symbol rates of 11 Mb/s and 1 Mb/s. In short, the higher-bandwidth transfer rate of WiFi more than compensates for the increased power consumption. Ultimately, the power-savings of WiFi is limited by the power consumed in the PSM state (256mW), which is still significant compared to the Bluetooth radio in sniff mode (25mW). This is where transitioning to the Bluetooth Sniff state can provide substantial energy savings.

The core of the CoolSpots model, outside of the basic switching architecture, are the policies that determine when to switch between the various radio technologies. There are two decisions that need to be made: when to "*switch-up*" to WiFi to increase available bandwidth, and when to "*switch-down*" to Bluetooth and power-off WiFi to conserve energy. The main penalty for switching is the latency and energy overhead of (de)activating the WiFi network interface – resulting in energy expended that is not doing any useful work. In fact, a poorly implemented policy would increase overall energy consumption by switching back and forth between wireless technologies without ever staying in any one state long enough to lower the overall energy consumption. The primary contributions of this paper are empirical measurements and evaluation of various policies that effectively manage multi-radio switching, thus resulting in an overall reduction in communication energy.

4 Related Work

Several previous projects have investigated techniques to reduce the power consumption of the wireless interface in portable battery powered devices. These techniques range from protocol optimizations at the various layers of the networking stack for single-radio systems to techniques balancing the capabilities of multiple radios to optimize overall system behavior.

Systems based on WiFi with optimizations for reducing energy consumption using just that radio include work done at the application layer [7][9], transport layer [4][2] and MAC layer [24][8]. The optimizations at the MAC layer adjust some of the parameters of the 802.11 Power Save Mode (PSM): for example, a bounded-delay addition to PSM can drastically reduce incurred delay while maintaining power savings [8]. CoolSpots, in contrast, augments WiFi with the lower power Bluetooth radio which has an order of magnitude less idle power than 802.11 PSM.

There have been a variety of proposals to extend Mobile-IP to support more efficient localized networking [6][25], but they have not considered utilizing multiple wireless interfaces from an energy saving perspective. Similarly, the handoff between local-area and wide-area networks can be used to provide ubiquitous coverage [18][19], but again not considering power consumption.

On-Demand Paging [1] and Wake-on-Wireless [14] investigate the use of a second lower-power radio as a wake-up channel for the main radio in order to save power, but incur large start-up latencies to activate a channel; these systems do not take advantage of the low-bandwidth, low-power wake-up channel to actually send data, an extension briefly mentioned in [4], which does not, however, address the switching policies or provide a detailed analysis. CoolSpots, in contrast, actively uses WiFi and Bluetooth radios for data communication and switches between them based on various criteria. Allowing both radio channels to transmit data, instead of just using one radio as a wake-up channel, has the benefit of allowing applications with low-bandwidth communications to operate with only the lower-power radio active. Similarly, multiple radios can be lever-

aged to minimize energy consumption in the device discovery and connection establishment phase for mobile devices [10], but this has not been considered in the context of active data transfer.

Multi-radio systems can also be controlled using application-level hints to determine which wireless channel would be most efficient [11][16], unlike CoolSpots, which requires no application hints; these other systems, however, do not provide much detail about their experimental set-up. In [16] the authors claim a 10% *increase* in power when switching between multiple interfaces for a www benchmark, but do not explain where the increase in power comes from. Furthermore, neither work considers the impact of range, or location on the system.

Turducken [17] looks at hierarchical power management from a system-wide perspective, considering the entire system – not just the wireless interface – as a hierarchy of sub-components that can be turned off to save power. Their approach however requires applications to be written specifically for the optimized platform, and transitions between states are triggered by explicit application events.

Using multiple radio subsystems is a popular technique in the field of sensor networking, which involves a number of small highly energy-constrained devices. A second duty-cycled radio can be used to provide wake-up signals for sensor networks [13], and overlay networks [3] use a short-range low-power radio for connecting small nodes with intermediate base-stations, and then a larger-hop radio, such as WiFi, for communication over larger distances – reducing the total energy consumption of the aggregate system. Conceptually, CoolSpots could also be used to optimize links in an overlay network, but the range disparity between the different wireless technologies would have to be addressed.

5 Switching Policies

As mentioned earlier, managing power used by multiple network interfaces requires the system to make two decisions: when to activate the higher-power WiFi interface, and when to shut off the WiFi interface and switch down to Bluetooth. Framed in terms of bandwidth, the question becomes when is there too little bandwidth available on the Bluetooth channel (switch up), or when is there too much unused bandwidth available (switch down). The simplest approach, using statically coded thresholds, does not work well given the variation in channel characteristics: the optimal bandwidth threshold changes as the distance between devices changes.

The various switching policies and the different criteria that guide the switch up/down decisions are enumerated in Table 2. These policies are executed by the mobile device and they are used to decide when to switch interfaces. Some switching decisions are made by measuring the active bandwidth on a given channel, and then activating a radio switch when a specified threshold is crossed. How-

ever, as mentioned earlier, this technique of employing bandwidth as the only criteria has problems as the available channel capacity changes with device range; this issue is addressed by two policies that use dynamic channel capacity detection for switch-up, and two different switch-down techniques.

A number of techniques to indirectly determine available channel capacity, such as indexing off of the measured channel RSSI, transmit power, or link quality, were investigated as a means to provide a dynamic switching threshold. None of these techniques, however, proved successful because the underlying metrics were either too unstable or not sufficiently correlated to actual channel capacity.

5.1 Switching Framework

The basic switching decision is made between a mobile device and a CoolSpot enabled base station, both of which possess Bluetooth and WiFi capabilities. The Bluetooth PAN profile is used to provide a standard IP channel. The mobile device is responsible for making the primary policy decision: it monitors the appropriate channel characteristics and (de)activates the WiFi radio when necessary. Also, it is responsible for communicating the switch to the base station in order to alter traffic routing (i.e., route packets across either the Bluetooth or WiFi link). The Bluetooth radio link is always kept active except when the device is out of range, but it is not used for communication while the WiFi radio is active. In effect, the switching is done at the networking layer (IP) of the networking protocol stack.

The CoolSpots switching framework is completely application agnostic for IP workloads: no application modifications are required to communicate bandwidth decisions to the underlying infrastructure. The CoolSpots framework can, therefore, work with any application, although it has to infer the optimal switching characteristics across a wide variety of application behaviors.

The switching setup assumes that a viable Bluetooth/PAN connection exists between the two devices, and that the mobile device knows the ESSID of the appropriate WiFi network: although pre-configured in the experiments, it would be easy to communicate this information across the Bluetooth link. Alternatively, the device could start with a valid WiFi connection, and then communicate the necessary Bluetooth connection information over the WiFi link.

Each policy has a number of corresponding parameters that can be tuned to affect its sensitivity and responsiveness, such as the sampling interval, switch threshold, etc. Changes in these parameters will affect the sensitivity and reliability of the system: causing it to be more or less aggressive, which will ultimately affect system energy consumption.

Table 3: Measured benchmark suite, with summary statistics (for a WiFi-only channel). Data transmitted is measured through the network interface, and so includes any protocol overhead.

Benchmark	Time over WiFi	Data Transmitted	Average Bandwidth (Data Size / Time)	Data Pattern
idle	60s	0.0 MB	0 kbps	None
transfer-1	13s	6.6 MB	4482 kbps	Bulk transfer
transfer-2	27s	13.3 MB	4519 kbps	Bulk transfer
www-intel	176s	21.6 MB	1022 kbps	Intermittent data
www-gallery	150s	2.9 MB	158 kbps	Intermittent data
video150k	150s	3.1 MB	172 kbps	Real time streaming video
video250k	150s	7.3 MB	402 kbps	Real time streaming video
video384k	150s	8.5 MB	464 kbps	Real time streaming video

5.2 Baselines

The *wifi-CAM*, *wifi-fixed,* and *bluetooth-fixed* policies serve as baseline cases for measuring the basic system capability and performance. *wifi-CAM*, used as a baseline, operates the WiFi radio in always-on mode, while all other policies operate WiFi in power-save mode (PSM). For some of the benchmarks, one of either the *wifi-fixed* or *bluetooth-fixed* policies will often behave quite well; specifically, the WiFi benchmark works well for bandwidth-intensive applications, while Bluetooth works better in low-bandwidth situations. The true strength of the switching policies is their ability to work well across the entire range of benchmarks, as well as to handle applications with dynamic workloads.

5.3 Bandwidth

The *bandwidth-X* policies (Figure 8) monitor the bandwidth of traffic going across the active wireless link, and trigger a switch when the measured bandwidth goes above or below the specified threshold (X). The same threshold is used for switching up and switching down. In an attempt to remove spurious transitions, the algorithm has some hysteresis: it periodically monitors the bandwidth and triggers the switch when it exceeds the threshold for a specified number of consecutive intervals. The evaluation section details a suite of static bandwidth tests that cover the range of switching thresholds.

Overall, a static bandwidth policy will perform quite well assuming it is properly tuned to the available channel. The real world is less ideal as the underlying channel capacity can change due to distance from the base station, interference, obstacles or other circumstances, and so it is hard to a-priori pick the optimal static switching bandwidth. If the switching threshold is too high for a given channel, then the policy will never switch to the higher-capacity interface, limiting the system throughput to that of the current, less capable, radio. If the switching threshold is too low, then it will unnecessarily switch to the higher capacity interface, leading to wasted energy.

The bandwidth policies are primarily parameterized by the switching threshold, specified in terms of kB/s. If the measured bandwidth is above/below this value, the policy will trigger a switch to the other network interface. Implicit in this parameterization is a choice of the bandwidth measurement interval and a hysteresis component, in order to avoid switching on temporary or short spikes in measured bandwidth. For all evaluated cases, a 250 ms interval and a hysteresis constant of 6 subsequent intervals is used. These constants behave relatively well for the given workload, although an exhaustive search was not performed on the parameterization space.

5.4 Cap-Static

The *cap-static-X* policies (Figure 8) use an active channel-capacity measuring technique to switch up, and then a static bandwidth threshold to switch down. A simple network ping round-trip time measurement is used to determine when the channel is "saturated" – a small round trip time (RTT) indicates that there is still available channel capacity, while a larger round trip times means that all the transmission slots are full, thereby delaying the ping packet. The ping RTT metric also works very well to detect other channel issues such as interference and obstacles. Switching down is accomplished just as with the static bandwidth benchmark, based on observed bandwidth across the higher capacity channel.

The basic asymmetrical nature of the algorithm is due to the asymmetry of the underlying network channels. A similar ping channel capacity detection technique can not be used for the WiFi channel, in order to detect when to switch down, because the channel is likely to be under loaded even at bandwidths that are still too high for the Bluetooth channel. The primary weakness of the cap-static policy is similar to that of the bandwidth policies: the fixed switch-down threshold may not be optimal for the given channel.

The cap- policies' (including cap-dynamic, below) switch-up is parameterized by the check interval, ping latency threshold, and number of intervals checked. For all evaluated cases, the check interval is set to 250 ms, and two consecutive latencies greater than 750 ms will trigger a switch. The switch-down parameterization of cap-static is identical to the bandwidth policies.

5.5 Cap-Dynamic

The *cap-dynamic* policy uses the same switch-up technique as cap-static, but instead uses a dynamically calculated threshold to affect the switch-down behavior. Specifically, it uses the measured bandwidth at the time of switch-up as the switch-down threshold. This technique dynamically captures the available channel capacity, implicitly taking into account the actual channel characteristics, such as range or interference. This dynamic capability prevents the system from either unnecessarily keeping WiFi active when the Bluetooth channel would be sufficient, or erroneously switching back to Bluetooth only to find the channel is still congested. The policy does assume that the channel characteristics do not change significantly during the switch-up state: a shortcoming that may potentially place the system in sub-optimal configurations.

As an optimization, both the cap-static and cap-dynamic algorithms only actively measure the channel capacity when the measured network flow is above a small minimum threshold, thus avoiding the ping packets causing unnecessary network activity and energy wastage. This minimum threshold is small enough that it does not cause a switch.

Other than the ping RTT measurement frequency and threshold (described under cap-static), there is no additional parameterization of the cap-dynamic policy.

6 Benchmarks

A suite of representative benchmarks (Table 3), ranging from simple file transfer to web-browsing emulations and media streaming, provides the basis for the evaluation of the various switching policies. Based on the nature of the CoolSpot models, different benchmarks will have wildly different impacts on the dynamic switching policies: For example, the "Idle" benchmark literally does nothing for an extended period of time and therefore will rely solely on the system's Bluetooth capability, while an intensive file-copy benchmark should immediately trigger WiFi for the duration of the transfer. Intermediate benchmarks with varied workloads, such as streaming media or web browsing, will provide a more accurate demonstration of the benefit for the various policies and their capabilities: these policies benefit applications that people are more likely to use on mobile devices.

Primarily, the "Communications Energy" metric is used to report the systems effectiveness. This metric, which is the product of completion time and average communication power consumed, succinctly summarizes overall system characteristics: it directly represents changes in the system's behavior (power consumption) and performance (completion time). Evaluating communication energy does not make any measure of a subjective "user appreciation" time, i.e., how impatient they might become waiting for their file to download – but this analysis is beyond the scope of this paper.

Similarly, the calculations only consider the communications power of the system, that is the combined Bluetooth and WiFi subsystem power, but not the power consumed by the entire device. The CoolSpots system is designed to reduce the power of the wireless subsystem, and as such is independent of the rest of the system. However, there are some indirect benefits. Streaming video to a PC/Digital Home TV to watch it there means that it is not being watched on the local LCD screen and the display can be shut down by the application (waiting for a key press to enable it), further extending battery life.

6.1 Baseline

Two basic benchmarks, *idle* and *transfer*, provide a baseline for evaluating the performance of the various algorithms, although they, in themselves, will not be indicative of real workloads. *Idle* causes no network traffic to be sent across the wireless link, while *transfer* represents a fast-as-you-can file transfer, which will consume all available bandwidth.

These two benchmarks are not indicative of a real system because they do not capture the system behavior surrounding the action itself: the process of starting the benchmark, or processing the results afterwards. Instead, they represent the asymptotic behavior of the communication channel, and assess how it would behave under extreme, and constant, workloads. Not surprisingly, *idle* and *transfer* correspond directly to the two basic wireless technologies, Bluetooth and WiFi, respectively: Bluetooth was designed as a low-power always-on technology, while WiFi was a high-bandwidth network replacement in which minimizing always-on power consumption was not a primary design constraint. Other benchmarks present a more realistic balance between these two extremes, something that will be more indicative of real workloads.

6.2 Streaming

The *streaming* benchmarks are a series of the same MPEG-4 video file transcoded to stream at various bit rates and transported using the Real Time Streaming Protocol (RTSP). The videos, coded at 128 kbps, 250 kbps, and 384 kbps (although the actual realized bitrates may vary), represent bit-rates suitable for mobile devices using a Bluetooth/PAN communication link. Higher bit-rates would be possible, of course, but would restrict the system to only using WiFi – and would not be very feasible for a battery-constrained mobile device. Furthermore, higher bit-rates would not be necessary for watching a movie on a small-screen mobile device such as a cell-phone. The streaming benchmark is implemented using the Darwin streaming media server and VLC video player – both of which are open-source standard components.

The basic data pattern of a RTSP stream is an initial flurry of activity as the player buffers video data to smooth out jitter in the delivery time, and then is followed by a

steady stream of data at the representative bit rate. The end of the movie, therefore, continues to play after data transfer has stopped, emptying out the buffer. Ideally, from an energy perspective, the system would use WiFi up-front to initially fill the buffer, and then fall-back to Bluetooth to handle the trickle of data and closing idle period. The video player program VLC will exit if it drops too many consecutive frames, indicating an underlying failure in the transport channel that represents an undesirable user viewing experience.

6.3 Web Traffic

The two *www* benchmarks represent a standard web browsing session, including downloading html pages, associated images, idle "think" time, and downloading large content files (such as a data sheet or other large document). The www benchmark was created by monitoring a typical web session, downloading the content locally, and then creating a script which mimics the traffic pattern for the content. Two versions of the www benchmark are derived from two different web sessions and have different traffic patterns, although the overall benchmark flows are similar.
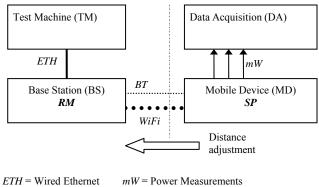
The www benchmark comprises a variety of traffic patterns, which presents a good opportunity for a dynamic-switching algorithm to optimize overall energy consumption: The WiFi-only policy will behave poorly because it will consume a lot of power in the active state, and the Bluetooth-only policy will be very slow for downloading large images or data files. Furthermore, many individual web-pages are actually fairly small, making it more worthwhile to use Bluetooth to transfer them, instead of a higher-power WiFi transfer. Overall, it is difficult to evaluate the end-user effectiveness of the www benchmark because changes in the download speed can have subjective effects on the user experience; therefore, the evaluation suite focuses only on the energy/power/latency evaluation of the system.

7 Experimental Setup

The experimental setup depicted in Figure 5 is designed to evaluate the behavior of the different policies across a variety of environmental conditions (i.e., distance between components). A Base Station (BS) device effectively acts as a wireless hub which supports both Bluetooth and WiFi capabilities, and also allows dynamic switching between the interfaces. Likewise, the Mobile Device (MD) possesses both wireless interfaces and runs the switching policies. A Test Machine (TM) is responsible for running the test suite and comprises the data-transfer end-point (through to the MD). The Data Acquisition (DA) machine uses specialized hardware to capture detailed power traces for the MD. The effect of range on the CoolSpot system is measured by physically moving the test apparatus around on an equipment cart, placed at specific well-defined locations.

Table 4: Different location configurations. The bandwidth and power numbers represent the measured channel characteristics at the given range for full data transfer.

Locations	Measured Bluetooth Bandwidth	Description
Location-1	564 kbps	2 meters (line of sight)
Location-2	544 kbps	7 meters (line of sight)
Location-3	256 kbps	8 meters (through wall)

ETH = Wired Ethernet   *mW* = Power Measurements
BT = Bluetooth Link   *WiFi* = WiFi Link
RM = Route Management   *SP* = Switching Policy

Figure 5: Experimental Setup

To exercise the test, the TM is given a file with a list of benchmarks (size N) and a file with the list of policy (size M), generating an MxN series of results. All the relevant data (including benchmark completion time) is captured by the DA. From the DA, the data can either be viewed graphically or exported to a file for processing. A post-processing script then processes the data to produce the duration of each benchmark and average power consumption for the various subcomponents for the MD: WiFi, Bluetooth and total power. It does this for each benchmark/policy pair, generating an MxN table of results of time, Bluetooth power, and WiFi power.

7.1 Hardware Specifications

The BS and MD are virtually identical hardware components based on the Stargate [23] research platform. The basic platform is based on the Intel® XScale™ PXA255 processor, and runs a standard version of the Linux operating system. The TM is an IBM ThinkPad T42 laptop, also running Linux. The TM is connected to the BS through wired 10 Mbps Ethernet. Details of the DA and collection hardware are provided below in Section 7.3.

The MD uses a Linksys WCF12 CF WiFi card, that has been updated to run a more recent version of card firmware that supports PSM. The card is supported using the HostAP wireless drivers, a standard component of Linux. Unless otherwise noted, the WiFi card is placed in PSM mode; the card firmware automatically switches to

fully-active mode when there is significant data traffic present.

Bluetooth is provided on the MD by the BlueCore3 module from CSR, supported by the Linux BlueZ Bluetooth stack. A Bluetooth/PAN profile connection provides standard TCP/IP link between the two devices. The BlueCore3 module supports multiple low-power modes, and the system is operating in a sniff mode with sleep enabled.

7.2 Network Management

Network traffic from the test machine (or any other machine on the local wired network) to the mobile device is managed by the BS using ARP and modifications to its local routing table. A network address on the local wired network is assigned to the mobile device, and its packets are routed either across the Bluetooth or WiFi network, as appropriate. Switching on the BS, therefore, merely entails a modification to the local routing table, and a similar adjustment is necessary for the MD. To effect a switch, the mobile device simply sends a "change route" message to the BS, after setting up its own network interfaces.

Although this implementation assumes that the BS is acting as both the WiFi and Bluetooth base station, it would be feasible to separate the technologies into separate base stations and use similar networking techniques to manage routing [6].

7.3 Energy Measurement

The energy measurement setup consists of a Fluke NetDAQ 2645A data acquisition (DA) device connected to a standard WindowsXP desktop system. The individual power rails for WiFi and Bluetooth are monitored by placing separate 1%-tolerance 20 mΩ resistors in series with each subsystems' power supply. The voltage drop across this resistor is measured, enabling the current flowing into the device to be calculated. The absolute voltage of the supply line at the device is also measured (nominally 3.3 V), and when multiplied by the current measurement provides the instantaneous power dissipation for the respective subsystem, which is then logged by the DA. Samples are measured at 10 ms intervals.

Energy, and not power consumption, is used to show the majority of the results because it captures both the power and time aspects of a particular benchmark. For example, if two benchmarks run and one consumes half as much power as the other, but takes twice as long, it will consume the same amount of energy.

The energy numbers reported here only measure the communication components of the system (WiFi and Bluetooth). Other components, such as processor, power regulators, memory, display, etc. are not included because, although they can be significantly impacted by the behavior of the wireless subsystem, they are not central to the Cool-
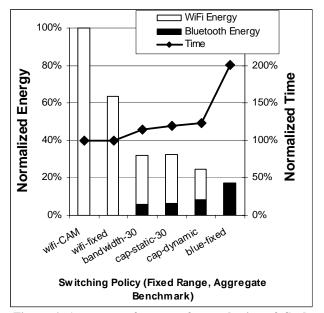

Figure 6: Average performance for a selection of CoolSpots policies at Location-2, across all benchmarks. Each bar summarizes the WiFi and Bluetooth energy consumed, while the line represents the execution time – both normalized to WiFi in fully active mode (without PSM).

Spots concept and their contribution can vary widely between platforms.

7.4 Location Configurations

Several different locations, summarized in Table 4, are used to measure the impact of range on the CoolSpots system. Bluetooth only has a nominal range of only 10m, and although it is operational at this range its effective bandwidth is considerably reduced. The effect of distance is the primary motivator behind the cap-based policies, which can dynamically reveal channel quality. Although not directly measured, increasing the distance simulates the effect of other kinds of channel interference that reduces overall channel capacity. The range of WiFi, which is on the order of 100m, is large enough that it does not factor into the measurements.

To make measurements at the various distances, the infrastructure side (TM and BS) of the test environment is located on a moveable cart and manually positioned at the specified location. This measurement technique does not take into account the results of dynamic mobility, that is, movement during operation, but rather just migration of the device from spot to spot; however, this case is representative of typical usage models in a home or office environment, where people don't actually use computing very much while moving, but more commonly access computing statically in a few well-defined locations.
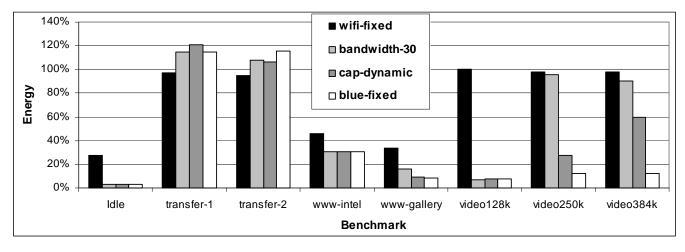
Figure 7: Breakdown across benchmarks for a selection of policies at Location-2, showing how the properties of the different benchmarks impact the various policies. Energy is percentage of WiFi-CAM.

8 Results

Figure 6 shows an overview of the benefits provided by the CoolSpots system, using a geometric mean across the entire benchmark suite normalized to WiFi-CAM, showing both energy and time for a variety of policies. The geometric mean is a standard technique used by the SPEC [22] benchmark suite because it has desirable properties when combining results across a range of disparate benchmarks. These results clearly show how dynamic switching policies can reduce energy consumption of the wireless interface by as much as 75% (using cap-dynamic), without significantly increasing the overall delay. The Bluetooth-fixed policy has the lowest energy consumption, with a significant increase in time.

Parameterizations of the bandwidth and cap-static policies are shown as bandwidth-X, where X indicates the switching threshold, measured in kB/s, so bandwidth-30 would correspond to a 30 kB/s switching threshold. No parameterizations for the cap-dynamic, wifi-fixed, and blue-fixed policies are shown. A 30 kB/s threshold translates to a 240 kbps data stream, which is less than all the measured streaming benchmarks.

8.1 Benchmarks

The effectiveness of the dynamic switching policies directly relates to the underlying benchmarks. Figure 7 shows a comparison of the performance of selected policies, in terms of energy consumption, across the range of benchmarks. Some, such as file transfer, offer little room for improvement since the channel is completely saturated; in fact, the dynamic policies slightly increase the overall energy consumption for file transfer as they incur the switching overhead without added benefit. In contrast, the idle benchmark is handled very well by the dynamic policies which can identify idle periods effectively and switch to Bluetooth.

The difference between the Transfer-1 and Transfer-2 benchmark shows the impact of the switching overhead for the dynamic policies. Transfer-2 handles exactly twice as much data, and so the overhead is roughly cut in half; the WiFi-fixed and Bluetooth-fixed policies are unaffected, since they have no switching overhead. The www benchmarks provide an intermediate point between Idle and Transfer, with intermittent periods of bulk transfer nestled between idle segments.

The streaming video benchmarks are interesting to examine in the context of CoolSpots because of their constant, but not saturated, workload. Basically, they trigger a failure point for WiFi-PSM because there is just enough data to keep the active mode of PSM activated – but there isn't really enough data to warrant the WiFi being used at all. All the dynamic policies strike a balance between the ideal case (Bluetooth-only) and worst-case (WiFi): although not ideal, they successfully handle the majority of cases. The streaming bandwidth required determines how well the channel behaves: the high-bandwidth streams are more apt to trick the dynamic policies into using WiFi when unnecessary.

8.2 Range

Range is a significant difference between WiFi and Bluetooth, and results indicate that the far-end of the Bluetooth range does in fact cause significant problems for some policies. Figure 8 details the bandwidth benchmarks when applied to the suite of locations, which shows either an increase in energy consumption or failure at the furthest measured location. The real problem with increased distances is unreliability: at distances further than Location-3, Bluetooth cuts out completely and is not a viable transport channel. This disparity highlights the benefit of the basic CoolSpots switching model, where the best available channel is used as conditions permit.
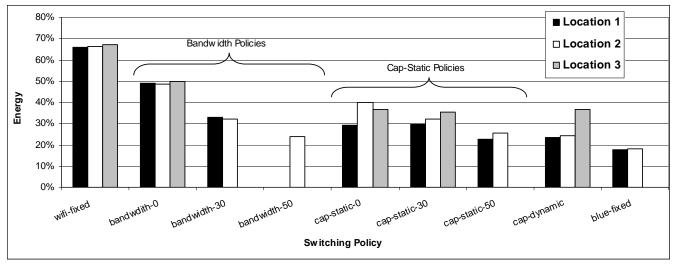
Figure 8: Location effect on benchmarks. Missing columns indicate that the given policy was not able to successfully handle every benchmark in the suite (at least one benchmark failed). Bandwidth-only benchmarks do very well for some locations, but do not handle a variety of locations well.

The hypothesis behind the cap-static and cap-dynamic benchmarks was that they would be able to better handle a variety of channel characteristics. As can be seen from the results, although the bandwidth policies behave very well at the short ranges, they are unable to usefully handle the longer ranges, and outright fail because they can not detect a necessary switch to WiFi. The cap-static policies, although they can detect the necessary switch-up, have an inappropriate switch-down threshold and either consume excess energy, or eventually fail. The cap-dynamic policy does not succumb to this problem because it dynamically senses both the switch- up and down points based on *measured* channel capacity.

8.3 Switching

The basic switching process encompasses a start-up process for the higher-level radio that incurs some delay. The initial switching process can roughly be divided into four parts: pre-transfer, detection, power-up, and switched. Figure 9 shows the timeline for both transfer-1 and video250k benchmarks using a simple bandwidth-only algorithm, showing the achieved data-rate a function of time. Since the Bluetooth channel is always available for communication, data transfer continues until the switch to use the WiFi interface is complete.

The bandwidth traces shown are measured from the respective network interfaces, and so include all TCP/IP overhead (resulting in about a 7% increase over just the basic data traffic). Right before the switch-on decision there is a spike for the Bluetooth channel of up to 1000 kb/s, which is greater than the maximum Bluetooth channel capacity, which is only about 500 kb/s. This spike is caused by buffering in the transmit path, which temporarily gives the illusion that more bandwidth is available on the Bluetooth channel.

The video250k graph shows the variable data rate requirements, which are theoretically supported by Bluetooth, but occasionally trigger the WiFi radio. Overall, the bandwidth-50 algorithm consumes 77% less energy than WiFi-fixed for this benchmark.

9 Discussion

Overall, the Idle benchmark highlights the necessity of incorporating a 2nd radio channel into the system: An effective automatic switching policy will identify the idle state and power down the WiFi radio, drastically lowering the overall power consumption. Alternatively, the user would need to manually activate the interface each time they wished to use it for data communication, something which does not yield a very compelling user experience. On the other hand, if the system is purely used for transferring bulk data then dynamic switching would not be necessary – but, it is unlikely that any general-purpose mobile device would be used in this fashion.

The bandwidth-0 policy is conceptually very similar to the use of the second low-power radio purely as a paging-channel as explored in [1][14]: as soon as any data transfer is necessary, the higher-power radio is activated. From this, it is easy to see how the more generic CoolSpots model realizes a large energy savings because it can sometimes utilize just the low-power radio for data transfer, without unnecessarily activating the higher power one. Technically, the bandwidth-0 algorithm still uses the low power radio for communication, until the higher power radio has been activated, a detail which only strengthens the argument.

Note that the measured results shown here focus on the communication subsystems of a device only, and do not include energy consumed by the rest of the system. One additional side effect of a slower policy like Bluetooth-fixed is that although it may consume less energy for com-

munication, it will keep the entire device active for a longer duration, reducing overall battery life because of the associated *system*-level power drain.

All the policies are subject to parameterization; however, the more they are specifically tuned for a specific situation the less likely they are to work well in others. The bandwidth and cap-static policies can either be tuned more towards energy savings, or towards performance – but the problem with optimizing for performance is that they become unreliable under some circumstances. The strength of the cap-dynamic policy is that it is relatively free of tuning – there is only one *ping threshold* parameters – and therefore is better able to handle a wide variety of channel characteristics.

10   Conclusion

The CoolSpots model provides a seamless way for mobile devices to automatically reduce their power consumption during wireless communication. Without requiring any application modification, the system utilizes multiple wireless channels to realize a greater than 50% energy savings across a representative suite of benchmarks when compared against standard WiFi-only power-saving techniques.

Several policies form the basis for switching between the wireless interfaces. The simplest policies, based on bandwidth monitoring, do very well under constrained channel conditions but have a difficult time adapting to greater communication ranges. A more adaptive algorithm (cap-dynamic), based on active channel measurements, is very effective at recognizing the appropriate instant to switch interfaces across a variety of channel conditions, yielding a robust and energy-efficient solution.

For applications with real-time traffic patterns, like the streaming media benchmarks, the standard WiFi power save mode (PSM) performs poorly as it ends up keeping the radio active. The cap-dynamic policy in CoolSpots, however, proves very beneficial: saving between 40% and 92% power (over WiFi CAM) for the various streaming benchmarks. This ability to adapt to steady-state low-bandwidth applications is a strength of a multiple-radio system.

There are several additional techniques that were not explored that could also yield effective switching policies. One example would be to monitor the amount of data waiting in the operating system's network buffering queue – providing information on both the channel capacity (data out), and application requirements (data in), which could be used to control switching.

Currently, the CoolSpots system only focuses on using Bluetooth and WiFi radios to provide lower-power local area network access, but the basic concept can be easily extended to other radio technologies. Wide-area technologies, such as GPRS or EDGE, can provide connectivity when a user wanders outside of WiFi coverage. Conversely, an even lower-power technology such as Zigbee could pro-
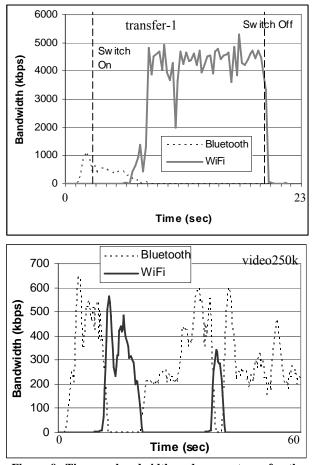

Figure 9: Time vs. bandwidth and power trace for the transfer-1 (top) and part of video250k (bottom) benchmarks using the bandwidth-50 policy. For transfer-1, data transfer starts at 1s, and the switching decision is made just before 3s, and then WiFi transfer starts around 7s. The video 250k graph shows multiple instances of switching, adapting to the dynamic load. The switch on/off events are not shown in the video 250k graph due to the compressed time scale.

vide an even lower-power alternative to Bluetooth (albeit at an even lower bandwidth); however, currently only WiFi and Bluetooth (and GPRS or equivalent for cell-phone devices) are commonly found in mobile devices, and so the applicability of Zigbee remains to be seen.

The CoolSpots concept easily integrates with existing network setups: it does not require extensive set-up or instrumentation of the infrastructure. A simple Bluetooth-enabled access point, operating within a home or office environment with WiFi coverage, is all that is needed. Alternatively, a conglomeration of devices could form an ad-hoc peer-to-peer CoolSpot, using Bluetooth for local communication and letting devices take turns maintaining a WiFi link with the local hotspot. This technique, although requiring individual devices to cooperate, would allow deployment of the Coolspots model with absolutely no modifications to the infrastructure.

11 References

[1] Agarwal, Y., Schurgers C., and Gupta R., "Dynamic Power Management using On Demand Paging for Networked Embedded Systems", In *Proc. of Asia-South Pacific Design Automation Conference (ASPDAC),* 2005.

[2] Akella, S. A., Balan R. K., and Bansal N., "Protocols for Low-Power," tech. rep., Carnegie Mellon University, 2001.

[3] Andersen, D. G., Balakrishnan H., Kaashoek M. F., Morris R., "Resilient Overlay Networks". In *Proc. 18th ACM SOSP, Banff, 2001.*

[4] Bahl, P., Adya, A., Padhye, J., and Wolman A., "Reconsidering Wireless Systems with Multiple Radios". *In ACM SIGCOMM Computer Communications Review (CCR),* 2004

[5] Bertozzi, D., Raghunathan, A., Benini, L., and Ravi, S.. Transport protocol optimizations for energy efficient wireless systems. In *Design Automation and Test in Europe*, 2003.

[6] Carter, C., Kravets R., Tourrilhes J., "Contact Networking: A Localized Mobility System," In *Proc. of the First International Conference on Mobile Systems, Applications, and Services(MobiSys),* 2003

[7] Flinn, J., and Satyanarayanan M., "Managing battery lifetimewith energy-aware adaptation". In *ACM Transactions on Compute Systems (TOCS) 22*, 2 (May 2004).

[8] Krashinsky, R., , and Balakrishnan H., "Minimizing energy for wireless web access with bounded slowdown", In *Proc. ACM Int. Conf. Mobile Computingand Networking, 2002*

[9] Kravets, R. and Krishnan P., "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, 2000

[10] Pering, T., Raghunathan, V., and Want R., "Exploiting radio hierarchies for power-efficient wireless device discovery and connection setup". In *Proceedings of the IEEE International Conference on VLSI Design ,*(January 2005).

[11] Qadeer., W., Simunic T., Ankcorn J., Krishnan V., De Micheli G., "Heterogeneous wireless network management," in *Proc. Of Workshop on Power Aware Computer Systems,* 2003.

[12] Raghunathan, V, Pering T., Want R., Nguyen A., Jensen P., "Experience with a low power wireless mobile computing platform". In *ISLPED*, 2004

[13] Schurgers, C., Tsiatsis V., Ganeriwal S., and Srivastava M.. "Optimizing Sensor Networks in the Energy-Latency-Density Space". In *IEEE Transactions on Mobile Computing*, pages 70–80, 2002.

[14] Shih, E., Bahl P., and Sinclair, M.J., "Wake on Wireless: an event driven energy saving strategy for battery operated devices". In *Proc. ACM Int. Conf. Mobile Computing and Networking*, 2002.

[15] Simunic, T., Benini L., Glynn P. W., and Micheli G. D., "Dynamic power management for portable systems," in *Proc. ACM Int. Conf. Mobile Computing and Networking*, pp. 11–19, 2000.

[16] Simunic, T., Qadeer W., and De Micheli G.. "Managing heterogeneous wireless environments via Hotspot servers", In *Proc. Of Multimedia Communication and Networking(MMCN)*, 2005

[17] Sorber, J., Banerjee N., Corner M. D., and Rollins S.. "Turducken: Hierarchical Power Management for Mobile Devices.", in *Proc. of The Third International Conference on Mobile Systems, Applications, and Services(MobiSys), 2005*

[18] Stem, M. and Katz R. H.. "Vertical Handoffs in Wireless Overlay Networks". In *ACM Mobile Networking (MONET), Special Issue on Mobile Networking in the Internet*, 1997.

[19] Wang, H.J., Katz R.H., Giese J., "Policy-Enabled Handoffs Across Heterogeneous Wireless Networks", in *Proc. Of Workshops on Mobile Computing and Applications*, 1999

[20] Want, R., Pering T., Danneels G., Kumar M., Sundar M., Light J.: "The Personal Server: Changing the Way We Think about Ubiquitous Computing.", in *Proc. of Ubiquitous Computing (Ubicomp), 2002*: 194-209

[21] Web: The Digital Home http://www.intel.com/personal/digital_home

[22] Web: http://www.spec.org/

[23] Web: Stargate Platform, http://platformx.sourceforge.net

[24] Woesner, H., Ebert J.-P., Schlager M., and Wolisz A., "Power saving mechanisms in emerging standards for wireless LANs: The MAC level perspecitve". In *IEEE Personal Communications*, vol. 5, pp. 40–48, June 1998.

[25] Zhao, X., Castelluccia, C., Baker, M., "Flexible network support for mobile hosts", in *Proc. Mobile Networking and Applications*, 2001