SwitchR: Reducing System Power Consumption in a Multi-Client, Multi-Radio Environment

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

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

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

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

1 Abstract

Multiple wireless network interfaces in a single mobile device exist in order to support their diverse communications and networking needs. These heterogeneous networks can be used to reduce power consumption, and thus extend the battery life of a mobile device, by enabling dynamic switching of radio interfaces depending on application requirements. This paper proposes a general switching architecture, *SwitchR*, for managing radio communications for multiple (client) devices utilizing multiple heterogeneous radios per device. To be effective and useful, the radio switching must be transparent to applications, and should be deployable incrementally within existing wireless infrastructures. The *SwitchR* framework considers the load imposed on the wireless channel by other communicating clients in order to make optimal switching decisions. We show that the resulting switching policy successfully handles multiple clients and reduces the energy consumption of all participating devices. We demonstrate reduction in energy consumption of a mobile device by 47% - 72%, depending upon the application, over the Power Save Mode in WiFi. *SwitchR* also leads to 13% - 60% reduction in energy consumption over previous multi-radio architectures that do not consider the interactions between multiple clients. Furthermore, we present a detailed analysis of how radio-switching affects applications, such as VoIP, that are commonly targeted for mobile devices.

2 Introduction and Motivation

Mobile networked devices increasingly feature multiple radio technologies such as cellular, wireless LAN and personal area networks in response to the increasing and converging capabilities on the mobile computing platform. Mobile applications include support for media streaming, web access, and downloading content from the Internet in addition to voice telephony. Radios commonly integrated with these mobile devices range from short range (local-area) radios such as Bluetooth and 802.11 (WiFi) to GPRS/EDGE or 1xEVDO for wide-area network access.

These *heterogeneous* radios present diverse capabilities, in terms of range of operation, nominal bandwidth, latency and power consumption characteristics. In most mobile platforms, the radio subsystems – whether the RF electronics or transmitted power – constitute up to 50% (786 mW WiFi, 81mW BT out of a total of 1.3W for the mobile device with the LCD turned off [14]**)** of the total mobile platform power [2][14][17].

Based on their origins, each of these radios have been architected for a specific purpose. As a consequence, these radios and their network interfaces are optimized to provide different forms of energy efficiency, depending on their primary design target. For short distances and low bandwidth connections, Bluetooth is highly efficient consuming on the order of 70 mW for active transfers, compared to almost 800 mW for active WiFi radios. Yet, for high-throughput applications, WiFi provides a lower energy/bit interface at 0.14mW/kbps compared to >0.22 mW/kbps for Bluetooth. Therefore for high throughput applications WiFi is more energy efficient than Bluetooth, which is more suited for lower data-rate or long idle conditions (Figure 1).

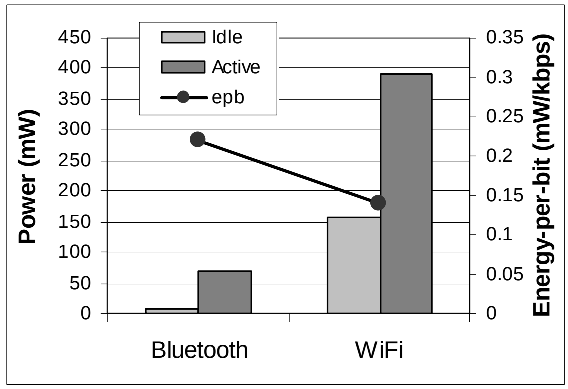


Figure 1: Base Radio Comparison – idle and active power of each radio technology, along with the active-transfer energy-per-bit (epb). Data is measured values from the experimental setup described later.

As the diversity and capabilities of mobile applications increase, so do the diversity and variation in the device communication needs. For instance, use of WiFi radios on mobile devices can lead to severely reduced battery lifetimes [2][14][17][22] despite their ubiquitous availability. Prior work has shown that this reduction in battery lifetime for a mobile device can be up to 1/5th of the rated lifetime, as opposed to keeping the WiFi radio always turned off [2]. A reasonable approach when dealing with devices with widely varying active/idle times and dynamically changing bandwidth requirements is to use Bluetooth and WiFi in a combined framework. Such observations have been used earlier in defining radio-switching techniques that, for instance, use a low power radio purely as a 'pager' to the higher power radio [2][17] or to do active data transfer [14]. These techniques range from the use of custom designed radios [17], to making substantial changes to the handset and/or the existing base stations [1][14].

This paper presents a major generalization of earlier work where the participating devices exploit knowledge of network-level parameters and the application needs at individual nodes when making switching decisions. In doing so, the proposed *SwitchR* system must overcome the following challenges:

(a) devise switch mechanism(s) that take into account not only the local knowledge of the wireless channel as seen by a communicating client, but also the traffic patterns of other simultaneously communicating clients;

(b) given an underlying wireless network infrastructure (such as WiFi), devise a switching architecture that allows incremental insertion of low-power access points that enables the clients to transparently switch networking technologies without any application-level modifications;

(c) ensure that dynamic switching among radios is not only energy efficient when including the overheads due to the switching decision, but that it also meets the quality of service requirements of diverse applications running on different clients.

There are practical aspects to these challenges: we must find ways to re-engineer the communications infrastructure while ensuring its easy adoption within existing wireless networks and using existing applications. This paper is primarily a demonstration of the fact that indeed the envisioned 'multi-client switching policy' using both local and global channel information can be implemented and leads to much more energy efficient switching decisions than can be taken by clients independently.

WiFi access points (APs), are already pervasive – commonly deployed in most workspaces, homes, and many urban locales. Several major metropolitan areas are considering plans to deploy universal WiFi coverage within their boundaries. The continued momentum for a widespread deployment presents both an opportunity and a challenge since any installed base of technologies is hard to replace or modify.

Our work is particularly relevant in the context of streaming media and Voice-over-IP (VoIP) applications being increasingly used on emerging mobile devices. While its ubiquitous availability and inexpensive cost of access of WiFi make it possible to use such applications, it also places an excessive power-drain on these devices. The WiFi radio interfaces cannot simply be turned off during idle time to save power, since the ensuing high latency of discovery and setup would make it impossible to meet QoS constraints such as latency and jitter associated with media applications.

This paper makes three primary contributions towards an effective multi-client multi-radio switching system:

1. An energy-saving switching architecture, SwitchR, based on *independent* low-power Bluetooth enabled APs that, unlike previous work, are incrementally deployable within an *existing* WiFi infrastructure.
2. A multi-client switching policy – and its detailed characterization and analysis – that enables energy efficient communication and networking among multiple simultaneously communicating clients within a multi-radio environment.
3. Analysis of how multi-client switching affects streaming media applications, including the standard Voice-over-IP (VoIP) protocol.

3 Related Work

Several techniques have been proposed to wireless power consumption beyond simple *idle* power-save modes. For instance, in the case of systems based on a single available radio, usually a WiFi interface, the ideas explored range from protocol optimizations at the application layer [8][13] to those at the transport layer [4]. Optimizations at the MAC layer [12][25] usually adjust tunable parameters of the 802.11 Power Save Mode (PSM) [10][11]; for example a bounded delay addition to PSM can drastically reduce the incurred delay while maintaining power savings [12]. However, optimizations for a single WiFi radio are limited to the energy savings enabled by the low power mode of the WiFi radio, which still has a power consumption that is an order of magnitude higher than that of other types of radios, such as Bluetooth, designed to be highly power efficient.

Given the limitations of single radio based systems, recent approaches have aimed at leveraging the availability of multiple radio interfaces on the same device. On-Demand-Paging [1], Cell2Notify [2] and Wake-on-Wireless [17] investigate the use of a second low power radio purely for wake-up purposes. However, these approaches incur large wake-up latencies and do not take advantage of the fact that the lower power radio can be used for active data transfer as well. Multiple radios can also be used to minimize the energy consumption during the initial phase of a device discovery and connection setup [15].

The idea of using multiple radios for various purposes, including a scheme called *data-on-lpr,* which suggests using the low power radio for active data transfer has been proposed earlier [3]. However, the authors do not present any evaluation of the benefits of such a scheme. Another technique is to use application level hints to decide which wireless interface would be most energy efficient for active data transfer [16][19]. The authors unfortunately do not provide much detail in terms of their evaluation and experimental setup; [19] in fact reports a 10% increase in power consumption for a benchmark emulating web traffic.

Our earlier work on the CoolSpots project [14] uses both Bluetooth and WiFi to optimize the power consumption of a *single* client, using a unified Bluetooth and WiFi access point, The SwitchR system differs from CoolSpots in two main respects: First, the switching architecture does not require unified APs and thus can be easily deployed within *existing* WiFi infrastructure. Second, the SwitchR switching policies consider the interactions between multiple wireless clients. Furthermore, our evaluation of SwitchR provides detailed jitter and performance characterization of the wireless channel during switching, and includes a detailed analysis of the VoIP application. We highlight the impact of the policy changes in our results section, as appropriate.

Seamless "vertical handoff" between heterogeneous local-area and wide-area networks [7], has been explored extensively within the perspective of overlay networks to provide ubiquitous coverage [20][21], and to provide bandwidth aggregation across multiple links [9]. However, neither of these schemes considers energy reduction as a goal to perform handoff. Extensions to Mobile-IP have also been proposed to support more efficient localized networking [5][26], but again they have not considered utilizing multiple radio interfaces for energy savings.

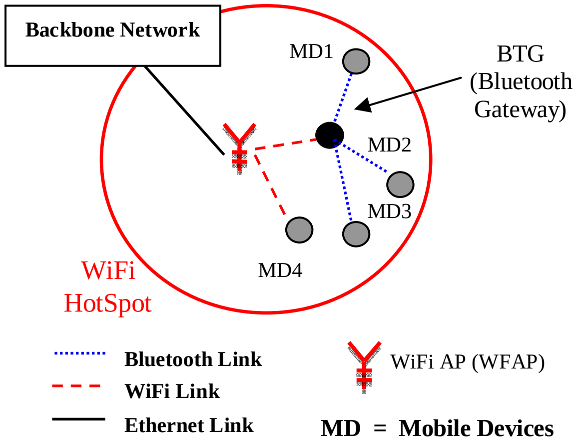


Figure 2: System Architecture – Scenario with multiple MDs, with the BTG node serving as the Bluetooth Gateway. MD1 – MD4 can connect through the BTG using Bluetooth or directly to the WiFi-AP using WiFi. For example, MD4 is directly communicating with the WiFi AP

4 System Overview

The SwitchR architecture, shown in Figure 2, introduces a low-power Bluetooth Gateway (BTG) device into already existing WiFi infrastructure networks. The BTG utilizes the Bluetooth PAN profile [5] to provide network layer (IP) connectivity to other Bluetooth devices. The WiFi AP is connected to the backbone network over an Ethernet link, while the BTG can be connected to the backbone network either over Ethernet or over WiFi. Individual mobile devices (MDs), initially connect to a WiFi AP (WFAP), just as they would when accessing a WiFi hot-spot, but then can optionally transition their connection to a BTG: enabling them to switch off their WiFi radio as desired.

A key contribution of our system is the mechanism for switching between the two network access points/gateways. Switching in the SwitchR architecture is accomplished transparently for MDs with active network connections, minimizing both switching time and connectivity disruption. Since the WFAP and BTG are separate access points, additional care is needed to facilitate this transition and have packets efficiently routed to the appropriate MD. Details of the switching mechanism are discussed further in Section 5.

4.1 Gateway/AP Separation

As mentioned earlier, one of the primary design goals of our switching architecture was to be easily deployable within *existing* WiFi infrastructure. Thus, the two main components of our SwitchR architecture are regular WiFi APs, which can be *any* 802.11 access point that is part of an existing infrastructure, and a Bluetooth Gateway (BTG) which is a device that functions as a Bluetooth AP. While a mobile device (MD) is communicating using its Bluetooth interface, its network traffic is routed through the BTG, which serves as a gateway to the infrastructure network, allowing the MDs WiFi interface to be switched off. Subsequently, when an application executing on the mobile device requires a higher-bandwidth connection, the MD can turn on its 802.11 interface and access the infrastructure's WiFi APs directly.

4.2 Multiple Clients

CoolSpots [14] presented an evaluation of switching policies for a *single* client scenario with a co-located AP setup; i.e. both WiFi and BT interfaces were attached to the same access point. When considering the scenario of multiple MDs communicating simultaneously in a multi-radio environment, making optimal switching policy decisions becomes much harder. For the single MD case, the MD had to determine whether the "quality" of the Bluetooth channel was satisfactory for its application requirements. In the case of multiple communicating MDs, the policies for switching between various interfaces must now take into account the dynamic nature of the Bluetooth channel as the presence of other MDs affects the total bandwidth available, in addition to the link quality of the Bluetooth channel between one particular MD and the BTG.

In Coolspots, an MD was able to estimate the Bluetooth channel condition as it was the only communicating client, with no other cross traffic. However, in the general case of multiple communicating clients there is only a limited amount of information that an MD can independently gather about channel utilization. Another alternative is for the BTG to control the switching decisions for the various MDs, since it has a global view of the Bluetooth network. The BTG however, has no knowledge of the communication needs of individual applications running on an MD. A hybrid approach that takes into account both the MDs application requirements and the effective capacity of the wireless channel is thus needed to design effective switching policies. The details of the multi-client policy that we have implemented will be described further in Section 8.3.

5 Switching Mechanism

Switching between the WFAP and BTG in the SwitchR architecture is accomplished by network level reconfiguration using Address Resolution Protocol (ARP) adjustments in the network and route-table updates on the MD as well as the BTG. In doing so we can ensure that the source and the destination IP addresses (Layer-3) of traffic to or from the MD remain the same irrespective of whether the MD is communicating over WiFi or Bluetooth. As a result, the switch between interfaces is completely transparent to any application executing on the MD as well as any remote clients wanting to communicate with the MD. The setup assumes that the MDs, the BTG, and the WFAP are all on the same IP subnet.

The switching mechanism is similar to that used for handoff in managed WiFi deployments with multiple physical WiFi APs as part of the same logical wireless network. In these WiFi deployments, seamless handoff is achieved using functionality provided by the Address Resolution Protocol (ARP) protocol, which provides a means to map an IP address (Layer-3) to the associated MAC address (Layer-2). In the case of WiFi deployments when a mobile client performs a handoff and associates with a different WiFi AP, the new AP sends out a *"gratuitous ARP"* to update all nodes on the local network (subnet).

5.1 Switching from WiFi to BT

Switching from WiFi to Bluetooth is a relatively quick operation that essentially relies on the BTG receiving packets for the MD over WiFi and routing them through BT – which

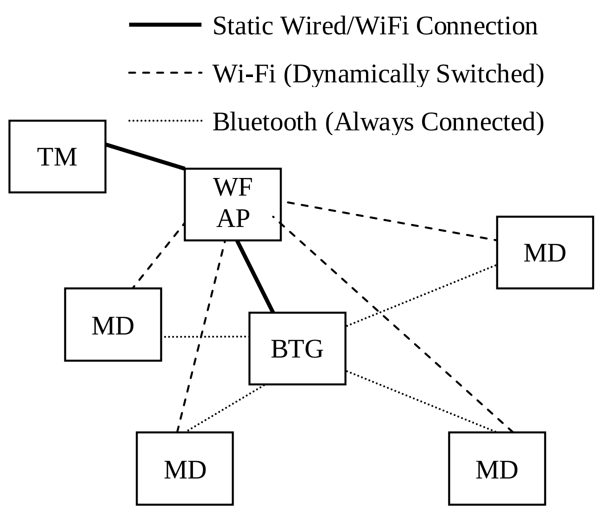


Figure 3: Experimental Setup, showing the four mobile devices (MD), Bluetooth Gateway (BTG), WiFi Access Point (WFAP), and Test Machine (TM).

can be accomplished by the BTG handling ARP requests in the case of traffic destined for the MD (called a *proxy ARP*). For traffic originating from the MD, as an optimization the ARP cache entries before the switch can be sent to the BTG. This "warming" of the ARP cache prevents unnecessary delays. The important steps for the switch to BT are:

1. Adjust MD routing table to outgoing traffic over BT
2. MD sends its ARP cache to BTG; set up proxy ARP on BTG and send out gratuitous ARPs
3. Delay to let WiFi buffers on the MD drain
4. Power off WiFi radio interface on the MD

5.2 Switching from BT to WiFi

Switching from BT to WiFi is dominated by the latency incurred by powering up the WiFi interface and the subsequent association with the WFAP. After that point, it is a fairly quick process to switch traffic over to WiFi. Similar to the switch to BT, it is necessary to warm the local ARP cache for the new WiFi interface to prevent unnecessary delays. The important steps for the switch to WiFi are:

1. Power on the WiFi radio interface on the MD
2. Wait until MD can contact the WFAP over WiFi. Then adjust MD routing table to send outgoing traffic over WiFi
3. Fetch ARP table from BTG to warm local cache
4. Send gratuitous ARP to redirect MD-bound traffic through WiFi
5. Release proxy ARP on BTG

5.3 Mobile Device Migration

In this paper we assume that the MDs are typically *nomadic*, i.e. they are mobile however they remain in several well defined areas (where a BTG is available for example). When an MD moves out of coverage of the BTG, there is an implicit disconnection of the Bluetooth connection: The MD will switch its connection to WiFi automatically to maintain connectivity. Effectively, a device moving out of range is handled using the same mechanism employed for handling a highly congested Bluetooth radio channel. In case the MD subsequently comes back in range of the BTG it can re-establish the Bluetooth connection and resumes its use of the SwitchR architecture to save energy.

5.4 Baseline Switching Analysis

Figure 4 shows a basic characterization of the switching mechanism of the SwitchR architecture under various operating conditions. For streaming applications we measure jitter and packets loss, to quantify the effect of a mid-stream interface switch. Figure 4a illustrates a single TCP transfer session that starts off on Bluetooth with a switch to WiFi is triggered, with an associated rise in observed TCP throughput once the switch to WiFi is complete. Figure 4b illustrates a switch from Bluetooth to WiFi and subsequently back to Bluetooth for a single 128Kbps UDP stream in an unloaded wireless environment (no cross traffic). As can be seen from Figures 4a and 4b, throughout the switch data continues to be transferred through at least one interface without interruption, highlighting the seamlessness of the switching mechanism.

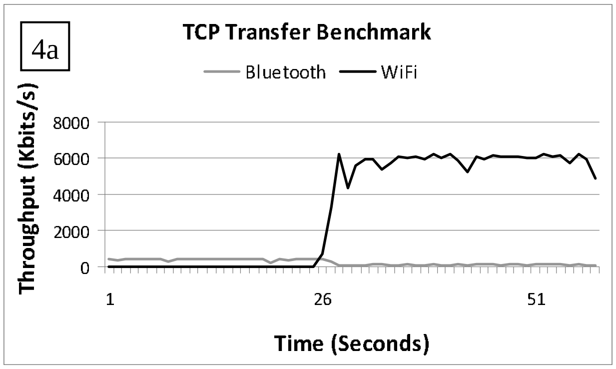
Figure 4c shows switching of a 128kpbs UDP stream when the wireless channels are loaded with other cross traffic: a 156Kbps UDP stream over Bluetooth, and a TCP transfer over WiFi. This graph illustrates that in the case of a loaded channel the jitter of the UDP stream rises above 20ms for a short period, but stays below 50ms (jitter requirement for VoIP). The two spikes in the jitter curves appearing in Figures 4b and 4c are a result of the MD communicating with the BTG as part of the interface switch protocol. Additionally, the time taken for the loaded switch to complete is slightly longer because some of the phases of switching require sending a message to the BTG or WFAP, which takes more time when the network is loaded.

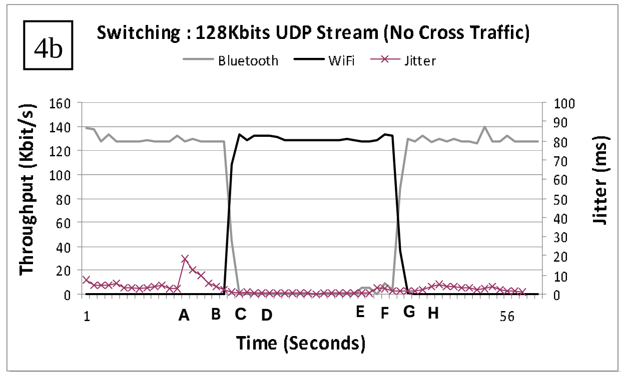
6 Experimental Setup

An experimental test bed consisting of multiple mobile nodes placed at various fixed locations in a moderately sized laboratory (8m by 12m) is used to test the SwitchR framework. Each mobile node is instrumented with an integrated power measurement capability and also monitors its own network traffic to log the amount of data transferred. Using this distributed power measurement and data logging capability, we can simultaneously measure the energy consumption for all of the mobile devices to get a detailed characterization of the overall system power consumption.

6.1 Device Configuration

The test setup we use for our evaluation, depicted in Figure 3, consists of four mobile devices (MD), a Bluetooth Gateway (BTG), a WiFi Access Point (WFAP), and a Test Machine (TM). The MDs and the BTG are based on the Stargate2 [23] research platform, an updated revision of the original Stargate platform. The SG2 platform has an on-board Bluetooth radio (Bluecore3) and supports a compact flash slot for inserting a wireless card (Netgear MA701). The WFAP that we have used is an off the shelf wireless router from Linksys (BEFW11S4), operated in AP mode. The BTG, Test Machine (TM) and the WiFi AP are all connected to a separate local subnet for the sake of performing controlled experiments and to prevent any spurious cross traffic effects.





UDP Switching Legend

A = Start of switch WiFi: begin powering on WiFi
B = WiFi interface ready, adjust local routes
C = After sending switch command to BTG
D = Send gratuitous ARPs to reroute traffic

E = Start of switch Bluetooth, fetch ARP table from BTG
F = After adjusting local routes
G = After sending switch command to BTG
H = WiFi card has been shut off

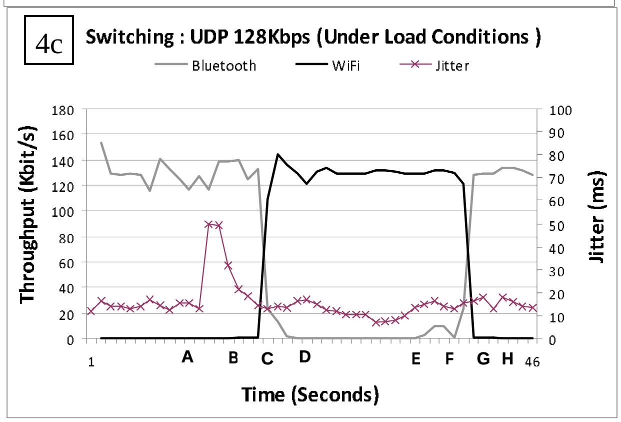


Figure 4: Switching Analysis, showing the switching time profile for three key situations. The TCP transfer switch (4a) shows the bandwidth difference between the two radios. The UDP switch shows the difference in bandwidth, jitter, and switching time between unloaded (4b) and loaded (4c) wireless channels.

A local DHCP server leases out dynamic IP addresses to the mobile devices on this subnet. The WiFi AP was set to use one of the non-overlapping frequency channels available in our building. All the other WiFi APs in our part of the building are on orthogonal channels, thus minimizing any interference effects from other WiFi clients.

The various MDs are spread across a laboratory room (8m by 12m), with the WFAP placed in one corner of the room while the BTG is placed near the center. The devices remain in their respective fixed locations during the experiment to ensure that the conditions are similar across multiple runs of different policies. In a mobile (pedestrian) environment the channel conditions will vary, of course. Our switching mechanism handles MDs moving in and out of range of the BTG as described earlier in Section 5.3.

6.2 Energy Measurement

To evaluate the effectiveness of our switching architecture under various switching policies and load conditions, we measure the energy consumed by the communication subsystem of each mobile device in our setup, which essentially means the Bluetooth and the WiFi radios. We do not include the power consumed by other components of the SG2 platform, such as memory, CPU, etc. as our primary goal is to reduce the energy consumed for communication. The power consumed by the other components is considered the "base" power consumed by the platform, under the observation that it can vary significantly across platforms. Although this base power is important to consider from an overall system power minimization perspective, it is not central to the concept of utilizing multiple-radios for reducing communication energy which as shown in previous work constitutes a major portion of the total energy budget of a mobile device [1][2][14][17].

We measure the energy consumption for all the devices *simultaneously* so that we can correlate the effects of energy consumption on each device with the traffic imposed by other MDs. The SG2 devices in our testbed are instrumented to have an on-board power measurement subsystem with an integrated Analog to Digital (A-to-D) converter. We have placed sense resistors in series with all the power rails supplying its operational subsystems, including the WiFi and the BT radios. To measure the energy consumption at any particular instant each device measures and logs the average power consumption of both BT and WiFi at regular intervals. At the start of a test the power logs are annotated with the test parameters, and when the tests have completed the logs are collected from all the mobile devices. The energy consumption for a particular test run is then calculated in a non-time critical fashion using our laboratory PCs. Using this capability we are able to measure the energy consumption of all the MDs simultaneously, giving us an accurate energy profile for all the mobile devices in our testbed.

6.3 Experimental Design

Our experimental design consists of four benchmark tests running on the four mobile devices; where in any run each mobile device executes a different benchmark. We ensure that each benchmark executes at least once on each device, factoring out any hardware variance between individual devices. In any run, all devices use the same policy; each benchmark suite is replicated for each of the four policies, resulting in 4 (benchmarks) x 4 (devices) x 4 (policies) = 64 benchmark runs for a set of results. The benchmark themselves execute in a continuous loop (since they are not necessarily the same length), and an individual result consists of a fixed-length sample of different statistics (e.g. power consumed) consisting of at least two complete benchmark executions. A detailed description about the individual benchmarks is presented in Section 7.

Statistics are collected independently on each device, consisting of power measurements, benchmark results (e.g., data transferred, packet jitter), and switching events. Results are post-processed by a script, which collects data from similar runs (same policy/benchmark) across the various mobile devices and aggregates results. The energy-per-bit values are calculated from the base power consumption (shown as individual Bluetooth and WiFi components), and total data transferred (not shown).

7 Benchmarks

The benchmark set used to evaluate SwitchR includes media streams at various bit rates, VoIP sessions, and web browsing traces. Since our evaluation focuses on a multi-client scenario, we use a set of *n* benchmarks to constitute an *application suite*, where *n* corresponds to the number of MDs in our test setup. There is considerable interplay between the various benchmarks due to the shared nature of the wireless network channel, which is representative of use in real world situations. Thus it is important to consider the effects of a given benchmark in the context of other benchmarks. The important characteristic of each benchmark is the bandwidth of data transfer in each time-slice, depicted in Figure 5.

7.1 Idle and Transfer

The two baseline benchmarks we use are the *idle* and the *transfer* benchmarks. The *idle* benchmark is the state of the system in which there is no data transfer taking place, while the *transfer* benchmark represents a TCP stream that tries to send data as fast as it can over the wireless link.

7.2 Streaming

The streaming benchmark models viewing live video content on a handheld, or streaming audio MP3s. Another increasingly popular application is Voice-over-IP (VoIP), which uses either SIP or the H323 protocols. All of these streaming type applications have real-time requirements and need QoS guarantees, which if not met can cause severe degradation in quality and result in bad user experience. Most media streams are sent over UDP and the two QoS metrics that are often used are *jitter* and *packet loss*. Media streaming applications can handle some packet loss by data buffering and interpolation, however large packet loss causes degradation in audio or video playback quality.

The standard *Iperf* tool is used to generate various sets of traffic patterns such as media streaming and VoIP. Iperf has various configurable parameters that allow customization of the UDP stream, such as a fixed data payload and a particular bandwidth, and is thus able to emulate a number of VoIP codecs. For our evaluation we emulate a commonly used VoIP codec, g711[24]. Furthermore, we use three streaming

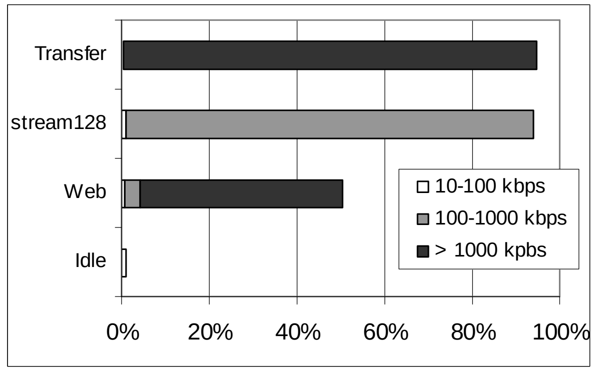


Figure 5: Benchmark Profile, showing the percentage of time each benchmark spends at a given bandwidth. This characteristic determines hints as to which radio technology should be used at any given time.

benchmarks: stream128, stream156, and stream200 with data rates of 128Kbps, 156Kbps and 200Kbps respectively. We have chosen these sample bit rates because these streams can be handled by our current BT v1.2 hardware (1Mbps). Recent Bluetooth v2.1 EDR+ hardware can provide even higher data rates (3Mbps) without any increase in power consumption.

7.3 Web Traffic

The *web* benchmark emulates the traffic pattern of a web browsing session. We monitored the web browsing traffic of a typical user and then downloaded the content that they visited locally. In addition, we measure the inter-arrival time between subsequent page requests capturing the user "think" time. To be consistent with our overall experimental setup, we used Iperf, which allows transferring data over a TCP connection to a remote device by reading data from a representative file. Our goal in creating this benchmark was to emulate a session with sporadic data transfer characteristics, i.e. periods of small transfers and a large transfer, interspersed with various idle intervals. This benchmark demonstrates the opportunity for energy saving, especially during the "think" time, when the low power Bluetooth radio is most efficient. Another purpose of this benchmark is to evaluate the effects that the bursty and sporadic nature of web requests have on the other benchmarks, such as media streaming and VoIP.

8 Switching Policies

There are two main decisions to be made when managing the power consumption in a mobile device with multiple network interfaces (a) When to switch on the high power, high throughout WiFi radio, and (b) when to switch back down to the low power, low throughput Bluetooth radio. Excessive switching can potentially increase power consumption and adversely affect applications, on the other hand inadequate switching will lead to inefficient operation. Furthermore, since the wireless is a shared medium, the switching decisions indirectly affect other nodes in the system, calling for policies that are aware of other nodes in the network.

Several simple policies are included to create a baseline comparison, while a client-focused policy (cap-dynamic),

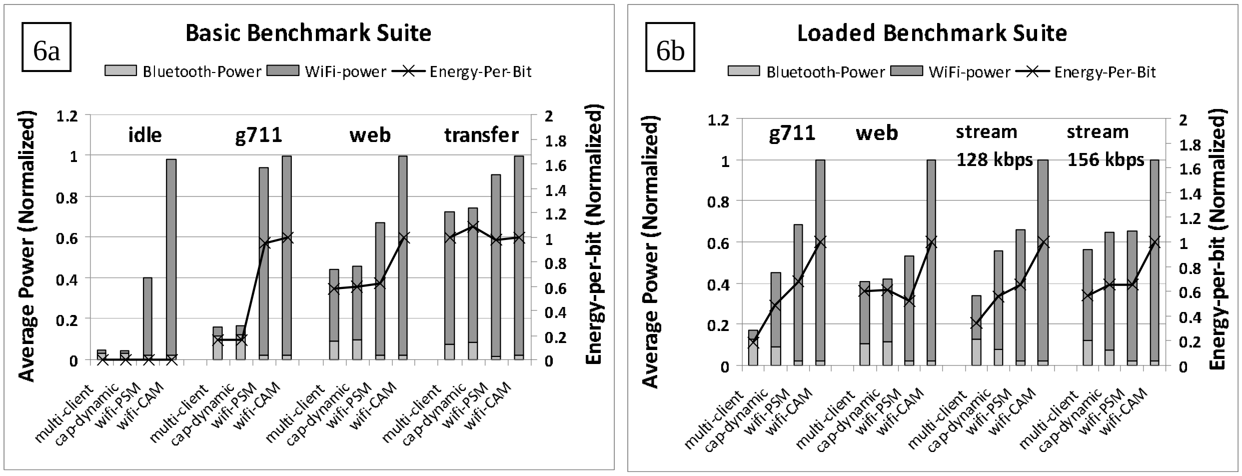


Figure 6: Switching results for the switching policies for two representative benchmark suites. These graphs show how, in some circumstances, the multi-client policy dynamically adapts to the changing conditions of the wireless channel. Details of the experimental design are described in Section 6.3. Figure 6a shows a benchmark suite consisting of the basic benchmark classes. Figure 6b shows a loaded benchmark suite that stresses the capacity of the underlying Bluetooth channel. (Note: Each bar represents an average of 4 runs for each benchmark)

represents the benefits of interface switching when a particular client only considers its own requirements. Finally, a multi-client policy, which considers all the nodes in the network, represents the added-value of the SwitchR architecture.

8.1 Baseline Policies

The *wifi-CAM*, *wifi-PSM* policies serve as baseline cases for evaluating the energy and performance behavior of the system. *wifi-CAM*, used as a baseline, operates the WiFi radio in always-on mode. *wifi-PSM* and all the other policies use the Power Save Mode (PSM) of WiFi [10][11], which essentially duty cycles the WiFi radio. We do not show results for a *Bluetooth-only* policy as Bluetooth bandwidth by itself is not enough to support multiple communicating clients and does not make for an interesting comparison.

8.2 Cap-Dynamic Policy

The *cap-dynamic* policy was the most energy-efficient switching policy from CoolSpots [14], which looked at the current capacity of the Bluetooth channel in order to make its switching decision. It uses ping echo-responses as an active channel capacity measurement technique for switching up, and uses a dynamically calculated bandwidth threshold to effect the switch-down behavior. Basically, a perceptible increase in the ping echo-response times, averaged over several intervals signaled a congested channel. For switching down it specifically uses the measured bandwidth at the time of switch-up as the switch-down threshold.

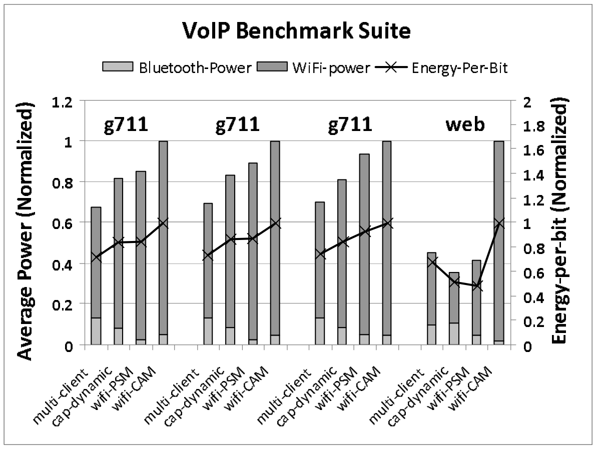
The *cap-dynamic* policy works reasonably well for single client situations; however, in multi-client situations it has significant problems correctly predicting the available bandwidth since it assumes that Bluetooth channel conditions measured on switch up remain constant. If the channel subsequently becomes free, for example by some device switching to WiFi or finishing its communication altogether, the other devices on the WiFi channel have no way of knowing this fact. These devices will thus continue to use WiFi believing the BT channel to still be congested and lose an opportunity to save energy by switching to Bluetooth.

8.3 Multi-Client Policy

A naive policy may cause the multiple MDs that are communicating at the same time, to independently conclude that the BT channel is busy and switch up to WiFi. Unlike the *cap-dynamic* policy described earlier, an effective multi-client policy needs to take two metrics into consideration when switching down to BT from WiFi. First, the policy needs to measure the *quality* of the BT channel as this places an upper bound on the total throughput the MD can possibly achieve given its location and range characteristics. The BT channel quality measurement does not capture the bandwidth that a client can get at a *particular* instant. Thus, the policy needs to determine whether there are other MDs actively using the BT channel at that time, and whether there is enough spare *capacity* on the BT to handle the MDs current application requirements. However, to estimate the spare capacity on the BT channel by the MD independently is difficult as the MD only has limited knowledge.

Taking into account these issues, the *multi-client* policy takes a different approach to determine the appropriate switching points. For the switch-up case to WiFi the *multi-client* policy uses the active channel quality measurement metric of multiple echo-response packets and the Received Signal Strength Indication (RSSI) of the BT link to estimate channel quality. If the average RSSI of the BT link degrades, and/or the echo-responses time increases substantially it signals a drop in channel quality. As soon as an application starts to transfer a large amount of data measured by an increase in echo-response time a switch-up to WiFi is triggered.

The switch-down case to Bluetooth is a combined decision that involves the MD as well as the Bluetooth Gateway. At the BTG the maximum bandwidth $MAXBW_{bt}$ that the BT interface can support is estimated empirically and set up statically at the start of experimentation. For switching-down the policy (executing on the MD) periodically measures the average bandwidth on the WiFi channel If the average bandwidth observed on the WiFi interface is greater than $MAXBW_{bt}$ then the policy reverts back to measuring the



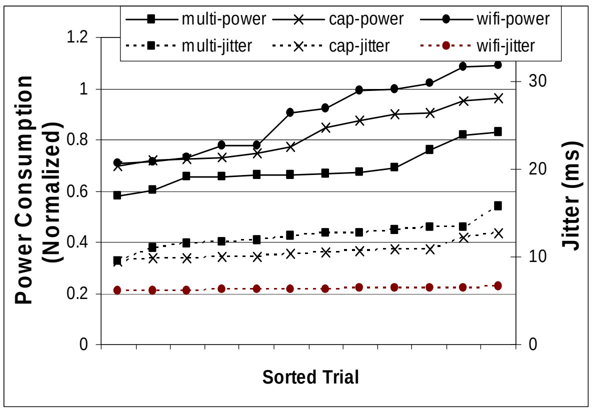
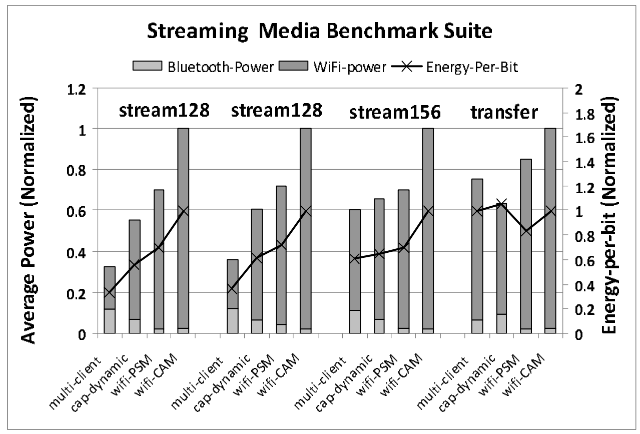


Figure 7a (left): VoIP streaming benchmark suite results, comprising of three standard VoIP benchmarks and one web benchmark. g711 is a high bandwidth (64kbps) uncompressed VoIP codec.

Figure 7b (left bottom): VoIP streaming power and jitter distribution. Each line consists of the 12 runs of the VoIP g711 benchmark (three instances, four runs in the experimental design). These items are not averaged over multiple runs; some of the variation is due to the variation of the different MDs. (wifi-power and wifi-jitter are for wifi in PSM)

Figure 8 (bottom): Loaded media streaming channel benchmark suite, with normalized averaged power consumption and energy-per-bit.



WiFi channel as there is no point in switching down to BT given the current application requirements. However if the bandwidth measured on WiFi is less than $MAXBW_{bt}$, then the *multi-client* policy performs multiple checks to determine whether it is optimal to switch down to Bluetooth.

First the policy checks the quality of the Bluetooth link by measuring the RSSI and the time for multiple echo-response packets. If these parameters measured do not reflect a good enough channel the policy does not trigger a switch-down to BT. In case the BT channel characteristics to the BTG are measured to be good, the *multi-client* policy queries the BTG and sends as a parameter the average application bandwidth requirements as measured on the WiFi channel. The BTG continuously measures the total bandwidth it observes through its BT interface and in case there is some spare capacity (bandwidth < $MAXBW_{bt}$) it sends back the spare capacity to the particular MD that sent the query. Once the *multi-client* policy running on the MD gets this message from the BTG it can switch down to BT if the BTG reports spare capacity on the BT channel. If the BTG does not report spare capacity the policy reverts back to the state of measuring the bandwidth on the WiFi interface and follows this decision process again.

Since it takes some time for the interface switch to happen, there may be a period during which the BTG has replied to a query by a particular mobile device (MD1), for available spare capacity, and meanwhile it receives another query from another device (MD2). If the BTG measures its spare capacity before the MD1 has actually switched to Bluetooth it will send an incorrect value to MD2 making both MDs switch to BT. This is only an issue if the BTG does not have enough spare Bluetooth capacity to handle both MD1 and MD2, in which case the MDs will immediately measure the channel to be congested and decide to switch back up to WiFi, causing a thrashing effect. In order to prevent this, as part of the policy the BTG delays replying to any more queries from other MDs after sending a spare capacity message to an MD.

8.4 Policy Analysis

Figure 6 summarizes the impact of each policy for two separate benchmark suites. Figure 6a considers the four basic benchmark types, and highlights the overall effectiveness of the multi-radio switching concept, while Figure 6b considers a more loaded scenario that highlights changes introduced in the multi-client policy. For all graphs, the impact of using the 802.11 Power Save Mode (*wifi-PSM*) as compared to using WiFi in the Awake Mode (*wifi-CAM*) shows how a single-radio optimization technique can impact power consumption by entering a low-power state when there is no data to transfer. The overall results are not surprising: Idle shows great savings, transfer shows very little savings, and the streaming media and web benchmarks show varied savings depending on context.

Note that measuring only power consumption can be misleading for some instances (such as base data transfer) because it ignores the *amount* of data transferred, which is captured in the calculated energy-per-bit value. So, although the dynamic and PSM policies consume less *power* for a straight transfer operation, they also decrease system throughput, resulting in a near-constant energy-per-bit value. Therefore, in most cases, a successful power-saving policy will show a reduction in the energy-per-bit along with overall power consumption.

As illustrated in Figure 6b, the multi-client policy saves up to 62% over the cap-dynamic policy and up to 72% energy over the wifi-PSM, depending on the application. The normalized energy-per-bit for the multi-client policy for the web benchmark in Figure 6b is slightly higher than that for wifi-PSM. The reason for this increase is that the web benchmark is active a lot of time and does not exhibit a lot of contiguous idle-time; therefore, there are not enough opportunities for the dynamic switching policies to switch down to BT to save power. Other web-browsing sessions that might contain more “idle-think” time, will lead to the switching policies performing much better than the wifi-PSM policy, which keeps the WiFi radio turned on.

The multi-client policy shows its main improvement for the VoIP and streaming media benchmarks, as shown in Figure 6b. These workloads are relatively constant, and the corresponding switching decision is dictated primarily by the behavior of the *other* nodes in the system (e.g., a change in workload by the web benchmark). The primary drawback with the cap-dynamic policy is that it only considers the data traffic through the respective device itself, and ignores other traffic on the wireless channel: when the web benchmark stops transferring data, the cap-dynamic policy does not adjust to make use of the now free Bluetooth channel.

9 Media Streaming Applications

As discussed earlier, streaming media applications such as audio, video and VoIP are important for emerging mobile devices. In this section we evaluate the effect that the SwitchR architecture has on streaming media, specifically with regards to the effect that multi-radio switching has on the QoS parameters associated with such real-time traffic.

9.1 Application Characteristics

The actual bandwidth required by a VoIP session is usually quite low and depends on the codec used (8Kbps for g729 and 64Kbps for g711). The inter-packet arrival time is usually around 20-30ms, with each packet data payload being between 20 – 60 bytes. A VoIP session thus sends and receives a large number of packets per second, although each packet is relatively small, resulting in an overall low bit-rate. Unlike normal media traffic which can be buffered a priori to reduce the effect of jitter, voice traffic has a strict jitter requirement, which is set to be less than 50ms for continuous speech. (All VoIP characteristics are taken from [24].)

We use Iperf to emulate several parallel VoIP streams to the mobile devices, for two commonly used voice codecs: g711 and g729 respectively [24]. g711 is an uncompressed-codec with a bandwidth of 64 kbps x 2 (bi-directional) while g729 is compressed and uses 8kbps x 2 (bi-directional).

9.2 Streaming Application Analysis

Figures 7, 8 and 9 outline results that focus on the behavior of media streaming and VoIP benchmarks. Figure 7a shows the aggregate results in a similar format as the previous section. It is important to note that any switching policy that utilizes the low-power channel for low-bandwidth traffic tends to benefit greatly in a VoIP scenario, given its low-bandwidth requirements. The Power Save Mode of WiFi [10][11], which is based on duty cycling the radio during periods of inactivity, is thus not efficient due to the short inter-packet arrival time between subsequent VoIP packets. The switching policies however perform much better in terms of power consumption as compared to the baseline wifi-CAM and wifi-PSM policies since they are able to switch to the lower power radio. From the graphs (Figure 7a and Figure 9), it is easy to see that the multi-client policy benefits by allowing some of the VoIP streams to drop down to the lower bandwidth radio, while the cap-dynamic policy does not effectively enable this switch. However the channel capacity of the current Bluetooth hardware (~400 kbps) in our testbed is less than the combined requirements of three bi-direction g711 VoIP streams (Figure 7a), thus requiring some of the streams to transition up to the WiFi radio.

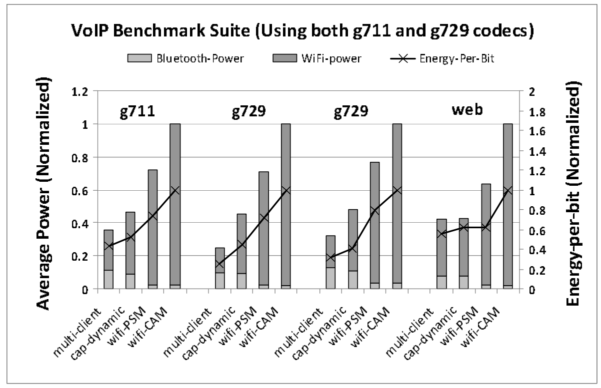


Figure 9: VoIP benchmark suite results, for 3 simultaneous (1 X g711 and 2 X g729) VoIP streams, in combination with a web benchmark. The multi-client policy saves between 18% and 45% energy-per-bit over the cap-dynamic policy for the VoIP streams. Jitter values for all the VoIP streams are within VoIP QoS requirements.

Figure 9 considers a suite of two low bandwidth g729 VoIP streams, a high bandwidth g711 stream and a web benchmark. Although the Bluetooth channel capacity should be able to handle these three VoIP streams, the occasional additional traffic induced by the web benchmark causes some of the VoIP streams to switch up to WiFi. The advantages of the multi-client policy can be clearly seen as it allows some of the streams to switch down to the low power radio. The multi-client policy thus results in substantial energy saving ranging from 18% to 45% as compared to cap-dynamic policy, and from 41% to 65% compared to wifi-PSM for the various VoIP streams (for the g711 and g729, respectively).

Figure 7b shows a power and jitter distribution curve for the various policies applied to g711 VoIP. Each data set is sorted from low to high, showing the resulting power and jitter distribution. This data represents the individual runs of three identical g711 VoIP benchmarks (with one web benchmark also running, not shown in the figure). The 12 data samples represent four executions of three simultaneous g711 VoIP streams. As can be seen from the graph, the jitter values for all the VoIP streams are less than 20ms, well within the QoS requirements of a standard VoIP session (50ms jitter tolerance). The jitter values for the lower bandwidth g729 codec benchmark (Figure 9) are not shown: they were all measured to be less than 20ms.

Figure 8 similarly illustrates a suite of three simultaneous media streams and a transfer benchmark. In the case of the multi-client policy both 128kbps media streams switch back *down* to Bluetooth after an initial period, thus reducing energy-per-bit by almost 40% compared to cap-dynamic and by over 52% compared to wifi-PSM.

10 Conclusions and Future Work

In this paper we have presented *SwitchR*, a novel multiple-radio based switching architecture, enabling mobile devices to use standard wireless applications yet significantly increase their battery operating time. A major advantage of our SwitchR architecture is that it is incrementally deployable within existing WiFi infrastructure, and that it can be used without modifying client applications. Furthermore, SwitchR performs well even with *multiple* simultaneous communicating clients, and reduces the energy requirements of all participating devices substantially. For our suite of representative benchmark applications, the multi-client policy enables energy savings up to 72% over the WiFi Power Save Mode (PSM), and up to 60% compared to previous multi-radio architectures.

We have also characterized the effect that switching between multiple radios has on media streaming applications and real-time VoIP traffic. We show that these applications can benefit substantially by using the SwitchR architecture in terms of energy savings, while maintaining the stringent QoS requirements placed on VoIP traffic.

WiFi and Bluetooth radios are currently-available technologies that are commonly found in existing platforms. Going forward, it will be important to investigate newer technologies such as 802.11n, which is a higher-bandwidth version of the earlier WiFi a/b/g standards, and Ultra Wide-Band (UWB) which is a very high-bandwidth, short-range technology. Although these technologies will present different power and performance profiles than current technologies, their different design targets (computer networking and consumer electronics, respectively), will most likely result in similar opportunities for power savings.

In our current implementation of SwitchR we employ our Bluetooth Gateway (BTG) devices to provide access to the low power Bluetooth channel, primarily because they are independent and can therefore be connected anywhere in the backbone network. This can also be done using commodity PCs which are already pervasive in enterprises and homes. These PCs can be appropriately augmented with a USB based BT dongle and can thus support the SwitchR infrastructure without requiring a separate BTG device.

11 References

[1] Agarwal, Y., Schurgers C., and Gupta R., "Dynamic Power Management using On Demand Paging for Networked Embedded Systems", In *Proc. of ASPDAC,*, 2005.

[2] Agarwal, Y., Chandra R., Wolman A., Bahl P., Chin K., Gupta R., "Wireless Wakeups Revisited : Energy Management for VOIP over Wi-Fi Smartphones", In *Proc. ACM MobiSys* ,2007

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

[4] 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.

[5] Bluetooth Special Interest Group, "Bluetooth Core" . Specification of the Bluetooth System, v 1.1. 2001

[6] Carter, C., Kravets R., Tourrilhes J., "Contact Networking: A Localized Mobility System," In In *Proc. ACM MobiSys* ,2003

[7] Chen, L., Sun, T., Gerla M., "USHA : A Practical Vertical Handoff Solution". *In Proc. of Multimedia Services Access Networks (MSAN), 2005*

[8] Flinn, J., and Satyanarayanan M., "Managing Battery Lifetime with Energy-aware Adaptation". In *ACM Transactions on Compute Systems (TOCS) 22*, 2 (May 2004).

[9] Hsieh, H., and Sivakumar R., "A Transport Layer Approach for Achieving Aggregate Bandwidths on Multi-Homed Mobile Hosts". in *Proc. of ACM Mobicom*, 2002

[10] IEEE 802.11b/D3.0. Wireless LAN Medium Access Control (MAC) and Physical (PHY) Layer Specification. High Speed Physical Layer extensions in the 2.4Ghz band, 1999

[11] IEEE 802.1x-2001. IEEE Standards for Local and Metropolitan Area Networks, 1999

[12] Krashinsky, R., and Balakrishnan H., "Minimizing Energy for Wireless Web Access with Bounded Slowdown", In *Proc. ACM Mobicom*, 2002

[13] Kravets, R. and Krishnan P., "Application-driven Power Management for Mobile Communication," *Wireless Networks*, vol. 6, no. 4, 2000

[14] Pering, T., Agarwal, Y., Gupta R., Want R., "*Coolspots:* Reducing the Power Consumption of Wireless Mobile Devices with Multiple Radio Interfaces". In In *Proc. ACM MobiSys* ,2006

[15] Pering, T., Raghunathan, V., and Want R., "Exploiting Radio Hierarchies for Power-Efficient Wireless Device Discovery and Connection Setup". In *Proc. IEEE - VLSI Design* ,(January 2005).

[16] 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.

[17] Shih, E., Bahl P., and Sinclair, M.J., "Wake on Wireless: an Event Driven Energy Saving Strategy for Battery Operated Devices". In *Proc. ACM Mobicom*, 2002.

[18] Simunic, T., Benini L., Glynn P. W., and Micheli G. D., "Dynamic Power Management for Portable Systems," in *Proc. ACM Mobicom*, 2000.

[19] Simunic, T., Qadeer W., and De Micheli G.. "Managing Heterogeneous Wireless Environments via Hotspot Servers", In *MMCN* , 2005

[20] Stemm, 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.

[21] 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

[22] Want R., Pering T., "System Challenges for Ubiquitous and Pervasive Computing". In *Proceedings of the International Conference on Software Engineering (ICSE)*, 2005.

[23] http://embedded.seattle.intel-research.net/wiki/

[24] VoIP Codecs, http://www.voip-info.org/wiki-Codecs

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

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