Managing heterogeneous wireless environments via Hotspot servers

T. Simunic[1,2], W. Qadeer[1], and G. De Micheli[1]
[1]Stanford University, [2]HP Labs

ABSTRACT

Wireless communication today supports heterogeneous wireless devices with a number of different wireless network interfaces (WNICs). A large fraction of communication is infrastructure based, so the wireless access points and hotspot servers have become more ubiquitous. Battery lifetime is still a critical issue, with WNICs typically consuming a large fraction of the overall power budget in a mobile device. In this work we present a new technique for managing power consumption and QoS in diverse wireless environments using Hotspot servers. We introduce a resource manager module at both Hotspot server and the client. Resource manager schedules communication bursts between it and each client. The schedulers decide what WNIC to employ for communication, when to communicate data and how to minimize power dissipation while maintaining an acceptable QoS based on the application needs. We present two new scheduling policies derived from well known earliest deadline first (EDF) and rate monotonic (RM) [26] algorithms. The resource manager and the schedulers have been implemented in the HP's Hotspot server [14]. Our measurement and simulation results show a significant improvement in power dissipation and QoS of Bluetooth and 802.11b for applications such as MP3, MPEG4, WWW, and email.

Keywords: wireless, low-power, scheduling, Hotspot, WLAN, Bluetooth, heterogeneous, access point

1. Introduction

Wireless communication today spans a wide range of devices, from cell phones to wireless sensors. The characteristics of wireless network interfaces (WNICs) are just as heterogeneous, ranging from sensor radios, to 802.11 family of standards and wide area networks (e.g. GPRS). Many portable devices have more than one type of WNIC. For example, IPAQ 3970 has Bluetooth and 802.11b WLAN built in, with the possibility of adding on WAN coverage through PCMCIA sleeve. Many of the issues related to usage of multiple WNICs in a mobile device remain. For example, seamless switching among the interfaces has not been widely implemented, so a mobile user is faced with a loss of a connection while migrating among different wireless interfaces. More importantly, power management of heterogeneous wireless devices still needs to be addressed at all levels, ranging from hardware all the way up to the application level.

Majority of wireless traffic today is infrastructure oriented. As a result, Hotspots [14] are becoming more ubiquitous, and with them the availability of fast wireless access. When a new client enters the Hotspot environment it registers via an application level proxy before it is able to access data in the infrastructure. Most of the infrastructure and applications assume that the wireless devices are always connected. This can cause a large battery drain, especially on the smaller devices. For example, work presented in [12] shows that even when 802.11b MAC layer power management is enabled, clients end up saving relatively little power in when a larger number of communicating clients is present. Higher-level management of power dissipation allow seamless integration between user applications and power management policy design thus allowing energy consumption to be reduced while maintaining a desired QoS. Even more benefits can be gained by migrating some power management to the Hotspot servers, as they have a better insight into what is going on with the overall wireless network under their span of control. As an increasing fraction of wireless traffic is multimedia, it is becoming even more important to be able to give some guarantees for the QoS (e.g. meet deadlines of streaming video). Thus there is a number of issues that a Hotspot server should be able to address: management of multiple wireless interfaces and seamless switching between them, power management of wireless communication and meeting QoS needs of diverse types of traffic One way

Multimedia Computing and Networking 2005, edited by Surendar Chandra,
Nalini Venkatasubramanian, Proc. of SPIE-IS&T Electronic Imaging,
SPIE Vol. 5680 © 2005 SPIE and IS&T · 0277-786X/05/$15

to address all these issues is to implement a resource management and scheduling algorithm at the level of the Hotspot server.

This work presents a new methodology for managing client-server based heterogeneous wireless environments through Hotspot servers. Once the client is registered via an application level proxy with the Hotspot server, it is handed off to a resource manager for scheduling of further communication. The goal of scheduling is to meet QoS requirements of clients while minimizing their power consumption. Our improved EDF and RM (IEDF,IRM) algorithms schedule the time and the size of data to be transferred between client and the server depending on the needs of the clients and the conditions in the wireless network. When needed, they switch communication between wireless interfaces at run time and enable low power states of the interfaces. We implemented the policies on HP's Hotspot server [14] using HP's IPAQ 3970 portable device supporting Bluetooth and 802.11b wireless network interfaces. The applications we tested range from MPEG video streaming to email. Both measurement and simulation results show that by utilizing our schedulers we are able to provide excellent quality of service, while saving up to an order of magnitude in energy on the client devices.

2. Related work

The techniques developed to date for the management of heterogeneous wireless environments primarily concentrate on improving their accessibility and QoS. These methods enable mobile devices to communicate with each other by introducing changes in the network protocol stack. For example, Mobile IP [1] assists the host's home network in forwarding packets to its network of current residence. However, with mobile IP data needs to traverse a multi-hop path even if communicating devices are in the same network. *Contact Networking* [2] resolves this issue by manipulating routing tables. Seamless switching between base stations has been proposed by using buffering of data on multiple base stations in close proximity to the mobile host [5]. Unfortunately, managing power consumption at the system level in heterogeneous wireless environments has been mainly overlooked.

Power reduction methodologies presented in the past largely focus on improving energy consumption of a single device, such as wireless LAN. Predictive techniques place a WNIC into low-power mode when longer idle period is anticipated [7]. However, incorrect estimates cause performance and power penalties. In contrast, stochastic models derive provably optimal power management policies [8-10] which have been shown to give large power savings in practice [10]. The 802.11b power management standard [11] has been in use for a number of years, but its savings are highly dependent on the amount of traffic present in the environment. The work presented in [12] uses a separate control channel with low-power radio to wake up a device whenever data is present. A number of scheduling algorithms aim to provide good QoS while saving power in the data access over high data rate (HDR) CDMA cell phones [31]. These schemes utilize information that is specific to CDMA-HDR design. There have been many scheduling algorithms proposed at IP and lower levels with the goal of addressing fairness and QoS in medium access [26-28]. Since these schedulers operate beneath application level, they can't utilize information available at higher layers.

Application level information is used for power management of streaming media in [13]. An excellent overview of energy efficient wireless network protocols is given in [18], and for ad-hoc networks in [34,35]. The power management techniques presented to date mostly focus on one WNIC [33]. This leads to inefficient power management for portables with multiple diverse communication interfaces and for systems where many client devices are present. Power-friendly transformations to media streams delivered to a client, and proxy-assisted prediction of packet delivery have been presented in [36]. Recent work [23] focused on managing power and QoS on the client side for multiple WNICs without considering other clients in the environment.

In contrast to previous work, in this paper we present an integrated resource management and scheduling policy for handling communication in the heterogeneous wireless environments for multiple clients. The HP's Hotspot server [14] uses the scheduling policies derived from EDF and RM [26] to dynamically select the appropriate wireless network interface on each client, to schedule data transfer in bursts of packets and to allocate appropriate bandwidth for communication. Each burst of communication with a client includes both uplink and downlink traffic between the Hotspot server and that particular client. Since all clients in Hotspot environment have to register with the Hotspot server in order to be able to access the infrastructure, the amount of contention from clients that have not registered is minimal. Details of the policies are discussed in Section 3. Section 4 presents measurement results that show large energy saving with good QoS when scheduling communication from Hotspot server to IPAQs using Bluetooth and 802.11b for applications ranging from MPEG video to email.

3. Resource management and scheduling

We enhance the Hotspot servers with a resource management and scheduling policy for communicating between heterogeneous clients and the Hotspot server with the goal of minimizing energy consumption in the clients while keeping good quality of service. Our system architecture is shown in Figure 1. All communication between clients and server is routed through client/server proxies. In today's implementations, the proxy is only used for authentication. We expand the proxies to include management of power and QoS of the clients. Note that applications can either be enhanced to provide information of their QoS needs to the resource manager residing in the proxy, or, if that is not feasible, the resource manager can predict application characteristics using a maximum likelihood estimator. A big advantage of scheduling at the server is that we can utilize the information provided to the server by all clients in the overall network. For example, if 802.11b gets saturated with the clients, the Hotspot server can seamlessly transition some of the clients, such as the ones reading email, onto the Bluetooth connection. In this way all clients have a better chance of having their QoS needs met while saving power. To highlight and contrast power savings and QoS improvements obtained by scheduling communication between multiple clients and the server, we compare six different scheduling algorithms: classical earliest deadline first (EDF) and rate monotonic (RM) algorithms, our improved versions of EDF (IEDF) and RM (IRM), round robin queuing (RRQ) and weighted fair queuing (WFQ). We use real-time scheduling algorithms, such as EDF and RM, since a growing fraction of wireless traffic is becoming deadline driven (e.g. video and audio).

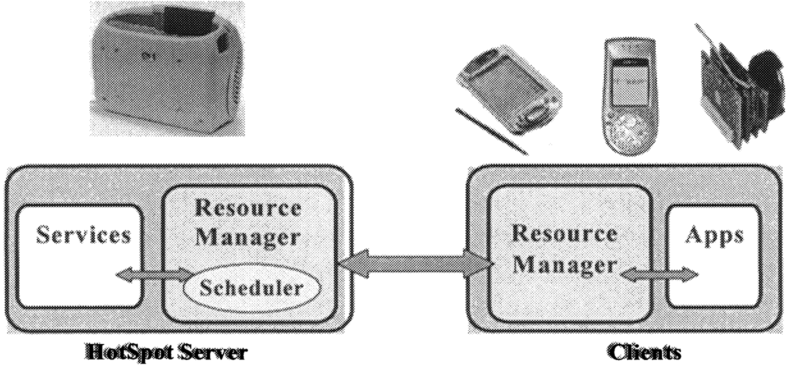

Figure 1: System architecture

The client's resource manager informs the server regarding the rate at which application consumes data already present in the client's buffer, λ_c, the average supported bandwidth through a particular WNIC, λ_t, the energy cost of communication, E_{comm}, the time and energy cost for switching between the interfaces, t_{switch}

and E_{switch}, the energy dissipation of data storage (e.g. RAM), $E_{storage}$, the energy dissipation due to computation, $E_{computation}$, the reception and transmission buffer sizes, B_{client}, and the application QoS requirements. The client's resource manager selects and wakes up the appropriate WNIC at the times decided by the Hotspot server, performs the data transfer and then transitions the interface into a low-power mode until the next rendezvous with the server.

The Hotspot server's resource manager schedules the communication time points with the clients. The server supports multiple types of network interfaces, so it can communicate with all clients in its environment, and as needed, it can request that a client switch from one interface to another. HP's Hotspot server currently supports 802.11b and Bluetooth [14]. We define separate data/scheduling queues at the proxy level for each network interface as they have significantly different characteristics in terms of power and performance. The scheduler partitions each queue into virtual time slots, where the minimum size of the time slot, t_{slot}, is determined from the maximum transmission unit, *MTU*, supported by the interface and the maximum throughput supported by the interface, λ_{tmax}, as shown in Equation 1. Although time slots are used to keep track of the schedule, the actual scheduling occurs only upon event occurrences (e.g. arrival of data to the client). The data is transferred in larger packet bursts in order to save power, not a packet at a time which is common to link-level schedulers.

$$t_{slot} = MTU / \lambda_{t\max} \tag{1}$$

When a connection request is received by the server, the scheduler selects a subset of available WNICs that meet the client's application bandwidth requirements and the server's schedule. Otherwise the client is not admitted. We select the WNIC that provides for the lowest overall energy usage while still meeting the QoS requirements, as defined by Equation 2. The communication can also be transferred from one WNIC to another using connection diversity implemented on the Hotspot server. Connection diversity assigns a locally fixed IP address to each client, so IP based applications can move seamlessly between different interfaces by buffering and manipulating routing tables. More details on how this is accomplished are in [2].

$$E_{total} = E_{comm} + E_{switch} + E_{computation} + E_{storage} \tag{2}$$

At the beginning of the communication, the client is informed about the amount of uplink and downlink data that is to be transferred between client and the server, B_{client}. It is a function of the energy tradeoff between communication and storage, as well as quality of service parameters [23]. If there is space in the schedule, the server transfers extra data to provide a cushion against variations in the system characteristics, B_{cush}. T_{norm} is the amount of time normally needed to transfer data, while T_{worst} is how long it takes to also transfer the data cushion, as shown in Equations 3 and 4.

$$T_{norm} = B_{client} / \lambda_t \tag{3}$$

$$T_{worst} = \left(B_{client} + B_{cush}\right) / \lambda_t \tag{4}$$

$$T_{next} = B_{client}\left(1/\lambda_c - 1/\lambda_t\right) \tag{5}$$

$$T_{deadline} = \left(B_{client} + B_{cush}\right)\left(1/\lambda_c - 1/\lambda_t\right) \tag{6}$$

The next communication point, T_{next}, is scheduled at the time when all of the normal buffer, and not the cushion, is emptied by the client. The absolute last chance to schedule is at $T_{deadline}$ when even the cushion is processed. Figure 2 and Equations 5 and 6 define these variables. Both values are a function of the client's buffer sizes, the application data consumption rate, λ_c, and the average supported bandwidth through a particular WNIC, λ_t.

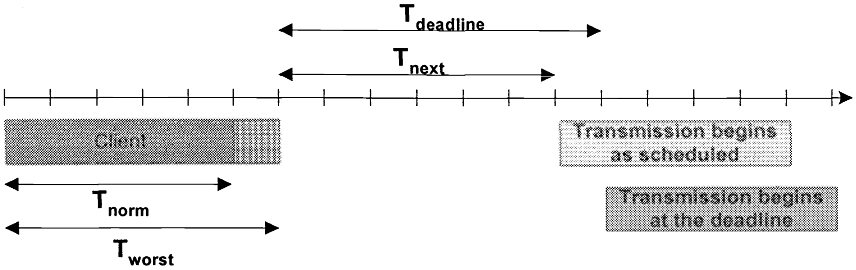

Figure 2: Scheduling deadlines

In order to ensure maximum power savings and acceptable QoS, scheduling of communication between the Hotspot server and its multiple heterogeneous clients needs to satisfy the time constraints described. Data transfer between Hotspot and its clients occurs at a time predetermined by the Hotspot server so the client's buffer doesn't overflow or empty out. The length of the client's sleep time is maximized to ensure largest possible battery lifetime. The client only wakes up when its point of communication arrives to lessen the effects of transition penalty between active and low-power states. The next subsections outline various scheduling algorithms we evaluated.

3.1. Classical EDF and RM

Classical earliest deadline first (EDF) and rate monotonic algorithms have been used in real-time applications. They are of interest to us as more and more of wireless data traffic is deadline driven (e.g. streaming media). Both algorithms provide a guaranteed limits on the utilization when used on a uniprocessor system with tasks whose characteristics are known at scheduling time (e.g. deadline, worst case execution time etc.) [26]. Multiprocessor systems and systems with highly dynamic task characteristics have been shown to be NP-complete from the scheduling standpoint [28]. Hotspot server communicating to multiple dynamically changing clients is thus also NP-complete. As a result, heuristics need to be used. RM assigns the highest priority to the client with the highest data rate. EDF gives the highest priority going to a client with the soonest deadline. In our implementation of EDF, the clients are prioritized based upon the amount of data that each client has, B_{rem} and the time it would take to empty out its buffer, T_{rem} using the most current system parameters as defined in Equation 7. The client with the shortest T_{rem} among the conflicting clients is assigned the highest priority and the possession of the WNIC. One performance issue of both algorithms is that entry of higher priority client interrupts the lower priority client's communication, thus incurring cost in power and delay. In our experiments we have found that a majority of Bluetooth scheduling conflicts leave 75% of disrupted clients with only a few seconds of data transfer left when their communication is preemptively interrupted.

$$T_{rem} = B_{rem} / \lambda_c \tag{7}$$

Another factor that limits the performance of dynamic EDF and RM is shown in Figure 3. *Client 1* gets scheduled at the end of its communication interval, $T_{tran,}$, to the point in time represented by T_{next}. Once scheduled, *Client 1* goes to low-power mode and thus is unreachable. Meanwhile, *Client 2* completes its communications and gets scheduled for its next point of communication at the same time as *client 1*. Since the priority of *client 2* is higher, *client 1*'s data transfer is delayed. *Client 1* wakes up at the predetermined point of communication only to find that its communication point has been delayed causing unnecessary transition between active and low-power state. This conflict can only be resolved if at the time of scheduling *client 1* some adjustment is taken for any other higher priority clients that might need to be scheduled at that same point in time later in the communication interval.

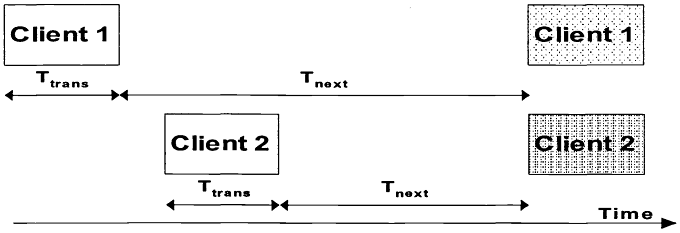

Figure 3: Illustration of scheduling conflicts

3.2. Improved EDF and RM

To solve the problem presented in Figure 3 we extrapolated the future points of conflicts, as shown in Figure 4. The priorities of the conflicting clients are estimated according to the current scheduling policy. Virtual schedule is developed for the clients whose future data transfer points fall between T_{next} and T_{dead} of the current client. The current client is scheduled when its priority is highest relative to the virtual schedule, thus avoiding the conflict.

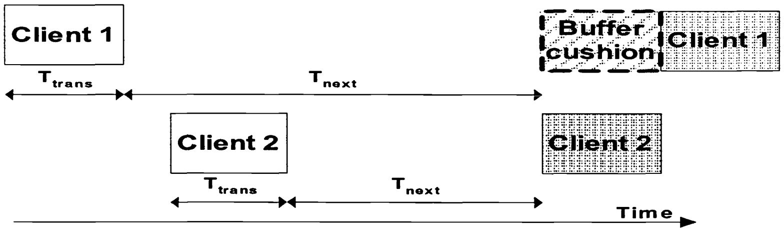

Figure 4: Removal of conflicts through extrapolation

We also made improved EDF and RM completely dynamic by enabling our scheduler to continually track and adapt to changes in the environment using the maximum likelihood estimator shown in Equation 8. A change in rate occurs at point *c* when computed likelihood over the last *w* data points is greater than a preset threshold. In our work we use 99.5% as a threshold. The change is observed between the old, λ_{old}, and the new rate, λ_{new}. Details of this algorithm are further discussed in [10]. Hotspot server immediately adapts its schedule to the newly detected change.

$$\ln(P_{\max}) = (w-c+1)\ln\frac{\lambda_{new}}{\lambda_{old}} - (\lambda_{new}-\lambda_{old})\sum_{j=k}^{m}\Delta t_j \qquad (8)$$

In order to solve the problem with frequent communication interruptions, we implemented non-preemptive versions of EDF/RM and at the same time increased the size of the buffer cushion. In this way we are able to minimize the negative effect on the delay due to non-preemptive nature, while keeping conflicts to a minimum. Note that delay is a big issue primarily in streaming media applications, while email is not so sensitive to it.

3.3. RRQ and WFQ

In this work we also compare two common packet level scheduling algorithms: round robin queuing (RRQ) and weighted fair queuing (WFQ) [32]. Note that in this work both of these algorithms are implemented proxy level and send bursts of data between the client and the Hotspot server. Both RRQ and WFQ are non-preemptive in nature with the same sizes of buffer cushions for Bluetooth and 802.11b as in IEDF and IRM. RRQ assigns the highest priority to the client that waited the longest. As a result, each client has its fair share of the bandwidth. WFQ estimates the time to finish each client's data transmission based upon the ratio of throughout assigned to each of the clients. The highest priority is assigned to a client whose data transfer time is shortest. In the next section we present the comparison between the scheduling policies discussed in this section in terms of power dissipation and QoS.

4. Results

We use HP's Hotspot server [14] to communicate with the clients according the scheduling policies considered: classical EDF and RM, improved variants of dynamic EDF (IEDF) and RM (IRM), round robin queuing (RRQ) and weighted fair queuing (WFQ). The clients are Linux IPAQ 3970 PDAs and laptops supporting both 802.11b (CISCO Aironet 350 PCMCIA LAN) and Bluetooth (CSR) interfaces. Tables 1 and 2 show the low-power states characteristics of the two WNICs. A DAQ card with a sampling rate of 10k samples/sec is used to obtain power measurements of WNICs. Tcpdump [24] and hcidump [25] utilities provided us with the throughput measures over the two WNICs. The resource manager and schedulers are a part of proxy, as discussed in previous section. The buffers used during scheduling of communication for each type of wireless interface are integrated with proxy as well, and as such are separate from link and network layer buffers.

Bluetooth	TRANSITION TIME (M SEC)	AVG. POWER (W)
ACTIVE MODE		0.18
PARK MODE		0.061
Park mode entry	2.16	0.077
Park mode exit	4.12	0.126
DEEP SLEEP		270μ
Deep sleep entry	250	0.061
Deep sleep exit	1	0.061

802.11b	TRANSITION TIME (M SEC)	AVG. POWER (W)
TRANSMITT		1.4
RECEIVE		0.93
Off state entry	1	1
Off state exit	300	1.4
DOZE STATE		0.045-0.93
Doze state entry	0.1	1
Doze state exit	1	1.4

Tables 1 & 2 : Bluetooth and 802.11b power characteristics

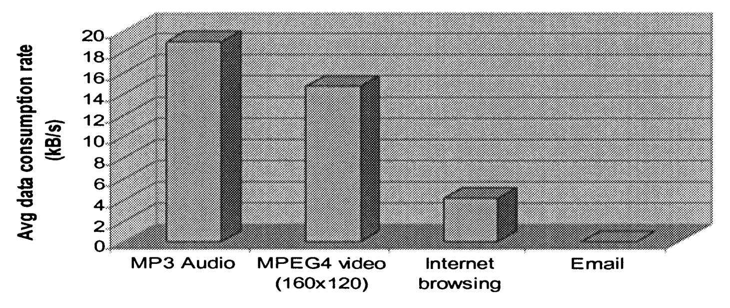

Figure 4: Application data consumption rates (kB/sec)

We experimented with a number of applications: MPEG2 video, MP3 audio, internet browsing and email. Diverse bandwidth and computation requirements of these applications provide an effective basis to compare system power savings and QoS tradeoffs. Figure 4 shows the measured data consumption rates of each application. Note that in all our scheduling algorithms we transfer data to client only when there is a need for data at the application level. Thus if the application is a web browser, and a request is to download a web page, we schedule the transfer of that web page. The client's WNIC is placed into low power mode once the transfer is complete, and if no other requests come from other applications on the client. Thus, when we show in our results collisions, idle time and delay, it reflects situations in which client's applications were expecting data, but did not get it on time.

4.1. Experimental validation

Meaningful analysis of power and performance of clients simultaneously communicating with the Hotspot requires a large number of portable devices. Network simulation, such as NS2 [30], is often used in such situations to enable fast and easy comparison of policies with a larger number of clients. In this section we compare the results of measurements on the hardware with simulations in order to validate the simulator implementation. We use Markov chains to model the behavior of each WNIC queue. State transitions are based upon events governed by the power state of the WNIC and data requests/transfers generated by the application and the server. These events have been simulated using exponential distributions with their appropriate data rates. Transitions between active and low-power states are controlled by the scheduling policy running on the server.

BLUETOOTH			*Experiment time:3000 sec*
MP3			
	Avg. power (W)	**Idle time (sec)**	**Data transferred (MB)**
Simulation	0.0583	0.19	56.4
Measurement	0.0556	0	55.7
MPEG4			
Simulation	0.0468	0.12	44.0
Measurement	0.0432	0	45.3

802.11b			*Experiment time:3000 sec*
MP3			
	Avg. power (W)	**Idle time (sec)**	**Data transferred (MB)**
Simulation	0.0437	0	53.0
Measurement	0.0447	0	55.7
MPEG4			
Simulation	0.0336	0	43.0
Measurement	0.0347	0	45.3

Tables 3 & 4: Comparison between measurements and simulations for Bluetooth and 802.11b

We used MPEG4 and an MP3 applications in our comparison as they have tighter deadlines as compared to web browsing and email. Table 3 depicts measurements for Bluetooth whereas Table 4 shows results for 802.11b with classical EDF implementation. Similar results can be obtained for other algorithms. In the tables we show *Average power* dissipated by the client during communication, *Idle time* spent by the client in the state when its buffer is empty and the total *Data transferred* by the client during the period of simulation. It is clear from the Tables that we have a very close match between power and quality of service measures (idle time and data transferred). The negligible discrepancy observed in the idle time values on Bluetooth is attributable to the use of exponential distributions in the simulator for modeling of data requests. Thus we verified that our simulation setup matches closely what we measured in the real implementation. Also note that this is an excellent example where Bluetooth consumes more power than 802.11b for the same application set. Bluetooth uses park mode for power savings, while WLAN is completely turned off. No performance degradation occurs on either due to careful selection of the size of data burst.

4.2. Scheduling policy comparison

We next highlight the effects of scheduling conflicts on power and QoS. As such conflicts occur in significant numbers only when the network is overloaded, we transfer on each WNIC 40% of its maximum bandwidth capacity. Note that in average conditions our algorithms are able to meet all QoS deadlines with minimum power consumption. The results obtained for 802.11b are in Tables 5- 8 whereas Tables 9- 12 show the results for Bluetooth. *Average power dissipation* is calculated over all the clients using the same application type averaged over the number of such clients. *Average idle time* is the average time spent in the

802.11b	*avg. throughput: 607.6 kB/sec* *Simulation time:9000 sec*					
Average power dissipation (mW)						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	35.0	46.5	34.7	41.8	40.2	40.6
Email	2.3	2.1	2.4	3.7	3.1	3.7
MP3	43.5	54.5	43.3	43.1	49.2	49.9
Internet browsing	14.7	17.7	14.4	21.0	17.9	20.6

Table 5. Average power dissipation for 802.11b

802.11b	*avg. throughput: 607.6 kB/sec* *Simulation time:9000 sec*					
Average number of collisions						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	21	190	18	119	64	92
Email	4	0	4	24	15	19
MP3	7	160	4	0	89	88
Internet browsing	15	63	13	104	62	90

Table 6. Average number of collisions for 802.11b

802.11b	*avg. throughput: 607.6 kB/sec* *Simulation time:9000 sec*					
Average time delayed (sec)						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	19	209	16	132	51	284
Email	4	0	9	27	13	33
MP3	6	190	2	0	83	272
Internet browsing	16	63	14	113	61	174

Table 7. Average time delayed for 802.11b

802.11b	*avg. throughput: 607.6 kB/sec* *Simulation time:9000 sec*					
Average idle time (m sec)						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	8.5	7.7	8.0	8.2	6.8	6.4
Email	0	0	0	0	0	0
MP3	11.7	11.9	11	11.3	12	12.3
Internet browsing	0.8	0.8	7.5	0.9	0.7	0.8

Table 8. Average idle time for 802.11b

Bluetooth	*avg. throughput: 62 kB/sec* *Simulation time:9000 sec*					
Average power dissipation (mW)						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	42.0	40.5	41.7	41.1	41.7	41.6
Email	0.4	0.4	0.4	0.5	0.5	0.5
MP3	53.2	51.4	52.8	53.3	54.0	53.8
Internet browsing	11.9	11.8	11.8	11.9	12.0	12.0

Table 9. Average power dissipation for Bluetooth

Bluetooth	*avg. throughput: 62 kB/sec* *Simulation time:9000 sec*					
Average number of collisions						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	10	143	9	71	80	74
Email	3	0	5	46	28	44
MP3	0	182	0	0	141	111
Internet browsing	16	78	12	175	81	121

Table 10. Average number of collisions for Bluetooth

Bluetooth	*avg. throughput: 62 kB/sec* *Simulation time:9000 sec*					
Average time delayed (sec)						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	24	1508	19	906	558	506
Email	22	0	70	489	246	489
MP3	0	2178	0	0	911	728
Internet browsing	60	735	52	2100	707	1330

Table 11. Average time delayed for Bluetooth

Bluetooth	*avg. throughput: 62 kB/sec* *Simulation time:9000 sec*					
Average idle time (m sec)						
Applications	**Scheduling algorithms**					
	IEDF	**EDF**	**IRM**	**RM**	**RRQ**	**WFQ**
MPEG4	181	393	181	187	193	164
Email	0	0	0	0	0	0
MP3	273	369	314	696	290	260
Internet browsing	31	43	28	25	31	17

Table 12. Average idle time for Bluetooth

state when the clients buffer is empty even though their applications are expecting data. *Average time delayed* quantifies the delay applications experience in getting data due to scheduling conflicts. *Average number of collisions* is defined as the average number of scheduling conflicts experienced by all the clients using the same application type. Bluetooth has five concurrently connected clients with two clients using MPEG4 video streaming and one client each of MP3 audio, email and internet browsing. WLAN has six MPEG4 video streaming clients, six MP3 clients, ten internet browsing clients and four email clients.

From Tables 5-12 we can see that our improved variants of dynamic EDF and RM outperform other scheduling policies. The improvement is large for some measurements (e.g. average delay time of 19 for IEDF with MPEG4 video streaming vs. 284 for WFQ), and for some it is slight (e.g. average idle time for WWW on 802.11b from 11 for IRM vs. 12.3 for WFQ). Note that although for a few individual cases our improved algorithms do not perform as well due to scheduling tradeoffs made between different applications (e.g. Table 9 for MPEG4 video streaming gives 42.0W vs. 40.5 for EDF), when averaged over all applications and clients, our algorithms always outperform. In addition, as compared to always-on policy used in today's WNICs, our algorithms save more than a factor of five in terms of power for Bluetooth, and a factor of 22x for WLAN. The maximum reductions in the average delayed time are 42x and 48x for Bluetooth and WLAN respectively whereas for the total number of conflicts these reductions become 14x and 17x.

As can be seen from results presented in the Tables, classical EDF and RM suffer from large average time delays (e.g. Table 11 shows 1508sec for EDF vs. 24sec for IEDF) as compared to the other scheduling algorithms due to their inability to extrapolate schedule in order to handle conflicts and due to their preemptive nature. RRQ gives priority to the client that has waited longest for its share of bandwidth, whereas WFQ gives the priority to the client whose data transfer takes the least time. Thus RRQ does not do as well at reducing average time delayed and the number of scheduling conflicts (e.g. in Table 6 RRQ has 64 collisions vs. 18 by IRM). IEDF and IRM, in contrast to other algorithms, schedule currently communicating client in a more sophisticated manner. As a result, IEDF and IRM have minimum average time delays as compared to the other algorithms (Table 7 shows 19sec for IEDF with MPEG4 video streaming vs. 284sec for WFQ). Average idle times are significantly reduced for Bluetooth (Table 12), but not affected much for 802.11b (Table 8) due to its faster response time. Email application in all cases has zero idle time, due to its small bandwidth needs and very flexible deadlines. IEDF and IRM also offer reductions in the average power dissipation of 1.2x relative to the worst performing scheduling algorithms and 22x relative to not performing any scheduling. In addition, the total number of scheduling conflicts of IEDF and IRM yield an order of magnitude reduction over the other scheduling algorithms.

4.3. Dynamic interface switching

In this section we highlight the advantages of dynamic tracking of system parameters and interface switching. In this experiment the total of 8 clients, each running one of the four applications (2 MP3, 2 MPEG4, 2 WWW, 2 email) are communicating with the Hotspot using Bluetooth while their 802.11b interfaces are off. During communication, the throughput of Bluetooth decreases which is detected by our maximum likelihood estimator. The system becomes overloaded with more clients than Bluetooth can support with its reduced throughput. Scheduling conflicts among clients increase, which in turn causes longer delay times (*Avg. time delayed column without switching* in Table 13). Switching is performed using Connection Diversity framework described in [2]. The amount of time it takes to switch interfaces and the energy consumed during switching are shown in [23]. Once a client switches from Bluetooth to 802.11b, the switched clients are able to transfer their scheduled amount of data without experiencing any conflicts since the throughput of 802.11b is much higher as compared to Bluetooth. The results of this experiment are presented in Table 13. *Avg. power switched* and *Avg. power no switch* indicate the average power if an interface switch has and has not taken place respectively. Similarly, *Avg. time idle switch* and *Avg. time idle no switch* indicates the average idle time if interface switch does(not) take place.

Scheduling Algorithm	Applications	Avg. power (mW)		Avg. time idle (s)		Avg. time delayed (s)	
		With switching	Without switching	With switching	Without switching	With switching	Without switching
IEDF	mp3	45.8	47.2	0.02	0.73	22.8	5766
	mpeg4	36.2	38.6	0.01	0.83	29.2	6035
	WWW	13.4	12.6	0.03	0.05	89.7	4240
	Email	0.4	6.7	0	0	23.3	3266
IRM	mp3	55.3	55.3	0.9	0.7	0	0
	mpeg4	34.8	42.0	0	0.29	0	2972
	WWW	2.7	2.1	0	0	6.6	8849
	Email	2.2	1.2	0	0	0	8650

Table 13: Interface switching between Bluetooth and 802.11b

Average idle and delayed times of the clients switching from Bluetooth to 802.11b are considerably lower in comparison to the non-switched case as shown in Table 13. For example, IEDF experience a minimum reduction of 47x in average time delayed when dynamic transition between interfaces is enabled at run-time. Similarly, idle times also decrease representing an enhancement in the QoS of the clients after switching as compared to the case when no switch is performed. As 802.11b's switching overhead is higher as compared to Bluetooth, it dissipates slightly more power for clients with lower data rates and smaller buffers such as email and internet browsing because the cost of switching is not amortized by the amount of data that is required to fill up the buffer. On the other hand, their QoS is excellent as indicated by the enormous reductions in the average time delayed. Clearly, the ability to schedule data communication and switch between WNICs has proven to significantly improve both client power consumption and QoS.

5. Conclusion

In this paper we have presented two new dynamic resource management and scheduling policies for managing client-server communication in the heterogeneous wireless environments for multiple clients. Improved EDF and RM policies select suitable WNIC for each client at connection setup, determine data transfer schedules and allow clients to enter low-power states during periods of inactivity to reduce power dissipation. Furthermore, they support seamless dynamic switching between interfaces during communication. Change in the system parameters is tracked using the maximum likelihood estimator. Our results show that by scheduling communication between Hotspot and its clients both the power consumption and the number of packet collisions is considerably reduced (e.g. 22x reduction in power for WLAN relative to always on policy).

6. References

1. D. B. Johnson, D. A. Maltz, "Protocols for adaptive wireless and mobile networking," IEEE Personal Communications, 1996.
2. C. Carter, R. Kravets, J. Tourrilhes, "Contact Networking: A Localized Mobility System," in Proc. of the First International Conference on Mobile Systems, Applications, and Services, 2003.

3. J. Flinn, M. Satyanarayanan, "Energy-aware adaptation for mobile applications," Proceedings of the 17th ACM Symposium on Operating Systems Principles, 1999.
4. B. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, K. Walker, "Agile Application-Aware Adaptation for Mobility," in Proc. of 16th ACM Symposium on Operating System Principles, 1997.
5. E.. Brewer, R. Katz, E. Amir, H. Balakrishnan, Y. Chawathe, A. Fox, S. Gribble, T. Hodes, G. Nguyen, V. Padmanabhan, M. Stemm, S. Seshan, T. Henderson, "A network architecture for heterogeneous mobile computing," IEEE Personal Communications Magazine, 5(5):8 -- 24, 1998..
6. H.Wang, B. Raman, C. Chuah, R. Biswas, R. Gummadi, B. Hohlt, X. Hong, E. Kiciman, Z. Mao, J. S. Shih, L. Subramanian, B. Zhao, A. Joseph, and R. Katz, " ICEBERG: An Internet-core Network Architecture for Integrated Communications," IEEE Personal Communications: Special Issue on IP-based Mobile Telecommunication Networks, 2000.
7. C. H. Hwang and A. Wu, "A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation", in International Conference on Computer Aided Design, pp. 28–32, 1997.
8. E. Chung, L. Benini and G. De Micheli, "Dynamic Power Management for non-stationary service requests", Design, Automation and Test in Europe, pp. 77–81, 1999.
9. Q. Qiu and M. Pedram, "Dynamic power management of Complex Systems Using Generalized Stochastic Petri Nets", Design Automation Conference, pp. 352–356, 2000.
10. T. Simunic, L. Benini, P. Glynn, G. De Micheli, "Event-Driven Power Management," IEEE Transactions on Computer-Aided Design, July 2001.
11. The Editors of IEEE 802.11, IEEE P802.11D5.0 Draft Standard for Wireless LAN, July, 1996.
12. E. Shih, P. Bahl, M. J. Sinclair, "Wake on Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices," in Proc. of ACM MobiCom, 2002.
13. A. Acquaviva, T. Simunic, V. Deolalikar, and S. Roy, "Remote power control of wireless network interfaces," in Proc PATMOS, 2003.
14. D. Das, G. Manjunath, V. Krishnan, P. Reddy, "Hotspot – a service delivery environment for Nomadic Users System Architecture," Mobile Systems and Storage Lab., HP Laboratories Palo Alto, CA Rep. HPL-2002-134, 2002.
15. CSR, "BlueCore Power Saving Modes," Jan, 2003.
16. Bluetooth Special Interest Group, "Specification of the Bluetooth System 1.1, Volume 1: Core," Feb 2001.
17. T. Simunic, L. Benini, A. Acquaviva, P. Glynn, and G. De Micheli, "Dynamic voltage scaling for portable systems," in Proceedings of the 38th Design Automation Conference, June 2001.
18. C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. "A survey of energy efficient network protocols for wireless networks", Wireless Networks, 7(4):343–358, July 2001.
19. M. Weiser, "The Computer for the Twenty-First Centry." Scientific American, Sept. 1991, pp. 94-104.
20. G. Pottie, W. Kaiser, "Wireless sensor networks," Communications of the ACM, pp 51-58, May 2000.
21. N. B. Priyantha, A. Miu, H. Balakrishnan, S.Teller. The Cricket Compass for Context-Aware Mobile Applications. Mobile Computing and Networking, pp. 32 - 43, 2001.
22. J. M. Kahn, R. H. Katz and K. S. J. Pister, "Mobile Networking for Smart Dust", ACM/IEEE Intl. Conf. on Mobile Computing and Networking (MobiCom 99), Seattle, WA, August 17-19, 1999
23. W. Qadeer, T. Simunic, J. Ankcorn, V. Krishnan, G. De Micheli, "Heterogeneous wireless network management," PACS 2003.
24. V. Jacobson, C. Leres, S. McCanne, The "tcpdump" Mannual Page, Lawrence Berkeley Laboratory.
25. M. Krasnyansky, hcidump, Qualcom.
26. C. L. Liu, J. W. Layland, "Scheduling algorithms for multi-programming in a hard real-time environment," JACM 1973.
27. Y. C. Tay, K. C. Chua, "A capacity analysis for the IEEE 802.11 MAC protocol," in Proc. of Wireless Networks 2001.
28. J. A. Stankovic, M. Spuri, M. D. Natale, G. Buttazzo, "Implications of classical scheduling results for real-time systems," in Proc. of IEEE Computer 1995.
29. D. Strassberg, "The glass is half full…", EDN, p. 53-60, May 27th, 2004.
30. K. Fall, K. Varadhan, " The ns Manual", 2003, http://www.isi.edu/nsnam/ns/ns-documentation.
31. S. Shakkottai, A. Stolyar, "Scheduling Algorithms for a Mixture of Real-Time and Non-Real-Time Data in HDR," Proceedings of 7th International Teletraffic Congress, 2001.
32. J. Bennett, H. Zhang, "WF2Q: Worst-case Fair Weighted Fair Queueing," Proc of the IEEE Infocom, pp. 120-128, 1996.
33. H. Zhu, G. Cao, "A Power-aware and QoS-aware service model on wireless networks," ," Proc of the IEEE Infocom,, 2004
34. S. Singh, C. Raghavendra, "PAMAS – power aware multi-access protocol with signaling for ad hoc networks," ACM SIGCOMM 1998.
35. S. Chandra, "Energy Conservation in Ad Hoc Multimedia Networks using Traffic-Shaping Mechanisms", MMCN 2004.
36. P. Shenoy, P. Radkov, "Proxy-assisted power-friendly streaming to mobile devices," MMCN 2003.