A device ecosystem that conserves energy by learning to put devices to sleep based on their usage pattern

Research Proposal and Progress Report *

Austin Derbique†
Utah State University
4205 Old Main Hill
Logan, Ut 84322
aderbique@aggiemail.usu.edu

Maneesh Mohanavilasam‡
Utah State University
4205 Old Main Hill
Logan, Ut 84322
maneesh.m@aggiemail.usu.edu

1. KEYWORDS

Sleep Server, Java, Subscriber, Publisher, Energy Efficiency, Networking, MAC Address, Wake-on-LAN, Machine Learning

2. INTRODUCTION

In today's day in age, most people own several devices whether it be a smart phone, tablet, or computer. Generally, people tend to only use one or two devices at a time, leaving the others to remain powered on and idle. This means that there is a great waste of energy in the devices not being used by the user.When a large number of our devices rely on batteries to operate, every bit of energy saved on a device is crucial. Additionally, when the user goes to sleep, it is unlikely that all devices are put into sleep mode, once again wasting valuable energy that could have otherwise been saved. One of the most common reasons that devices are left on is because users want to use their machines at will. Users typically want to avoid the time it takes for their device to boot [3].

As society becomes more energy conscious, there is a growing demand to reduce energy consumption while minimizing noticeable effects on the user. This paper's goal is to target that specific issue. How can we save battery life on your devices without affecting the user? The answer is a device ecosystem controlled by a lightweight server that can put any and all devices to sleep based on usage patterns. This means that your devices will go to sleep without the user initiating a sleep action. By simply having your devices sleep when they're not being used, it has been estimated there is a potential for 60-85 percent in energy savings [8].

The most innovative part of our sleep system is embedded in the usage pattern recognition. If for instance, our algorithms detect that the user goes to sleep at 11pm every night and wakes up at 7:30am, a message will automatically be broadcasted out to all listening devices and put them to sleep. Shortly before the user wakes up in the morning, another message will be broadcasted to wake up the devices. The patterns are endless and the more this sleep system is used, the more efficient it will become.

3. RELATED WORKS

Although the concept of sleep proxying has been around for quite a while [reference], there has been a lot of recent attention [references].In [1], the authors describe solution using the Wale-on-LAN (WOL) technology to allow computers in a corporate environment to go to sleep and be waken up. While this sleep server technology is powerful in the corporate environment, it is limited to Microsoft Windows running machines and is not suitable for all devices the user may use on a regular basis. Recently, Apple has released a sleep proxy that is catered to home networks that allows the devices to sleep [4].Unfortunately, this system only works with select Apple hardware and cannot be used with outside products. As for enterprise networks, there are systems like Verdiem [5] and Adaptiva [6] that focus on estimating power usage and waking up sleeping machines to perform tasks such as patching.

Apart from the approaches of saving power by sleeping, there are several methods that include power-proportional computing [7], the TickLess kernel [9], OS level power management [10], and dynamic voltage and frequency scaling [11]. Using these methods, CPU utilization can be fit into a model of computer energy use which could provide detailed accuracy into making our sleep server more efficient.

4. BACKGROUND

One of the most important papers used in this research consisted of a sleep server built by researchers at University of California, San Diego. In their study, the researches recorded power data for sleeping and idle machines as well as built an implementation of a sleep server for an enterprise environment. This made for a very similar foundation to base our research on.

5. TECHNIQUES

People today own multiple devices like cell phones, tablets, laptops etc. We would like to think of these devices owned by a person as an ecosystem of devices. All of these devices

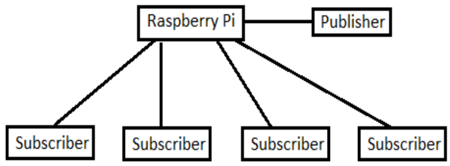

Figure 1: A sample black and white graphic (.eps format) that has been resized with the `epsfig` command.

in the ecosystem are dependent on a battery and are mostly running day and night. We want to try and reduce the energy consumption of this device ecosystem as a whole. Our hypothesis is that since different devices have different usage patterns we can put one or more devices in the ecosystem to sleep based on their usage patterns or at the users will. Thus, we will leverage both a publish-subscribe model and machine learning techniques to make this happen. Also, we would like the devices that are asleep in the ecosystem to still be available when it is needed by a user or service. For this we use the concept of a sleep server which is explained in detail below.

5.1 SLEEP SERVER

The concept of sleep server is not new and has proven to conserve energy as shown in the following research[1]. We would like to use the concept of sleep server in our project to make the devices that were put to sleep still available when needed by a user or a service. Before a device is put to sleep information like its MAC address and all the open ports the device was listening to is sent to the sleep server so that the sleep server can listen to the open ports of devices that are asleep. The respective devices are awakened by the sleep server whenever there is an activity on the ports that are being listened by the sleep sever for that specific device.

5.2 SLEEP PREDICTION MODEL

As described in the previous section, the sleep server is responsible for waking the device up. As it monitors open ports of all the devices in the ecosystem for activity there is scope for usage pattern detection. We plan to log the port activity of all devices at the sleep server and use machine learning techniques to identify usage patterns that will help us build a dynamic sleep prediction model. The role of this sleep prediction model is put the devices to sleep automatically based on the usage patter by the user and activity on open port.

5.3 PUBLISH SUBSCIBE MODEL

Since there are multiple devices in the ecosystem it is cumbersome to put individual devices to sleep manually. We propose the use of the publish subscribe model to control the devices in the ecosystem and put them to sleep and wake them up again. Any devices in the ecosystem are subscribers to the ecosystem it self and any device can step up and become the publisher. The publisher can put one, many or all devices in the ecosystem to sleep and transfer all controls to the sleep server.

6. SYSTEM DESCRIPTION

This section explains the basic working of our system.As shown in Figure 1, our system consists of multiple devices acting as a subscriber and one device acting as a publisher. All devices are connected to a sleep sever(Raspberry Pi) and to each other in a peer to peer network. The following steps give an understanding of the workings of the system:

1. All devices register to the ecosystem as a subscriber
2. Any one device in the ecosystem can act as a publisher
3. The publisher can instruct one or all devices in the ecosystem to sleep
4. All sleeping devices are partly awake on the sleep server as the The sleep server monitors open ports of all sleeping devices for activity
5. The sleep server also logs the activity on open ports for all devices that are asleep for sleep pattern detection
6. The sleep server also logs the users activity as a publisher and records all manual sleep instructions to devices for sleep pattern detection
7. On collection of sufficient data a Sleep Prediction Model is generated based on the sleep pattern for each devices and the sleep server adheres to this model unless instructed otherwise by the user
8. The Sleep Prediction Model decides when a device in the ecosystem can be put to sleep without user the sleep instruction coming from the user itself.

7. PROJECT PLAN

The following table describes our project plans and deadlines

8. SLEEP SERVER ARCHITECTURE

Sr No	Task	Deadline
1	Implement publish subscribe Model	03/24/2017
2	Integrate sleep server functionality	04/07/2017
3	Identify sleep pattern	04/14/2017
4	Build sleep prediction model	04/31/2017
5	Final Submission	05/05/2017

9. DELIVERABLES

A working model of the device ecosystem that conserves energy by putting particular devices to sleep based on their usage pattern.

10. PROGRESS REPORT

Our current progress lies in calculating energy usage of a desktop computer to better accurately give us a model for our implementation. The tool used is ptop and the data samples are recorded in 1000ms increments. The energy computation model was recorded using an ALPHA value of 0.6. The max CPU power consumption topped out at 24.5 watts and the minimum CPU power consumption was 3.8 watts. System idle was recorded at 45.0 watts for the entire machine with a current power weight of 0.8. This was used to to compute the weighted power of the system. The following experiment was conducted for a duration of an hour and the average consumption was computed for the total number of simultaneous listening ports.

Ports Open	Average Power Consumed
1 Port	8.5 Watts
2 Ports	9.8 Watts
3 Ports	11.3 Watts
4 Ports	11.7 Watts
5 Ports	12.0 Watts

The average power consumption seems to settle down and does not increase considerably as the ports increase. Maybe they stagnate at a specific number of ports. More evaluations must be conducted for the same. Future research must be made before an implementation is feasible. Future work will continue in identifying sleep patterns, building a sleep prediction model, and then implementing this model into a sleep server for "smart" functionality.

11. EVALUATION

Because the focus of our research was on a sleep model rather than a sleep server, our goal was not to implement the system as this could be done easily in future research. Using such a sleep model showed significant energy reduction based on power measurements of a computer idle and a computer asleep. It is estimated that up to 70 percent of energy can be saved if a sleep server using our sleep model is implemented.

12. POTENTIAL USAGES

Using existing sleep servers combined with a sleep prediction model such as the one we designed would have the possibility to extend into residential and commercial markets. The research has shows that even with only a couple of devices entering sleep mode, the cost of running a raspberry pi and start up energy consumption is less than the energy saved by putting those devices to sleep. This application could be integrated into a smart home that automatically turns all devices off. This could include your TV, air conditioner, really anything that supports TCP/IP protocol.

13. CONCLUSION

Overall, the idea of a sleep server is nothing new. But if we can harness the power of machine learning to produce a sleep server that learns on its own and has automated functionalities, the practicality and usability of such a system would greatly increase. With over 70 percent of energy savings can lead to longer battery life for the user and a greener future for us all.

14. REFERENCES

[1] J. Reich, M. Goraczko, A. Kansal, J. Padhye, Sleepless in Seattle No Longer,USENIX Annual Technical Conference. 2010

[2] K. J. Christensen and F. B. Gulledge. Enabling power management for network-attached computers. Int. J. Netw. Manag., 8(2):120âĂŞ130, 1998

[3] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta. Somniloquy: augmenting network interfaces to reduce pc energy usage. In NSDIâĂŹ09, Berkeley, CA, USA, 2009

[4] Apple Wake On Lan, http://www.macworld.com/article/142468/2009/08/ wake on demand.html

[5] Adaptive Technologies, http://www.adaptiva.com/

[6] Verdiem Technologies, http://www.verdiem.com/

[7] Wake-on-LAN, http://en.wikipedia.org/wiki/Wake-on-LAN

[8] Agarwal, Yuvraj, et al. "Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage." NSDI. Vol. 9. 2009

[9] Tickless kernel. http://www.lesswatts.org/projects/tickless/

[10] SleepServer: A Software-Only Approach for Reducing the Energy Consumption of PC's within Enterprise Environments

[11] D. C. Snowdon, E. Le Sueur, S. M. Petters, and G. Heiser. Koala: a platform for os-level power management. In EuroSys âĂŹ09: Proceedings of the 4th ACM European conference on Computer systems, pages 289âĂŞ302, 2009

[12] A. Sinha and A. P. Chandrakasan. Energy efficient real-time scheduling. Computer-Aided Design, International Conference on, 0:458, 2001.