Achieving Energy Efficiency while Synchronizing Personal Data

Nalini Belaramani and Mike Dahlin
The University of Texas at Austin
{nalini, dahlin}@cs.utexas.edu

Abstract:

As users increasingly rely on various portable devices and network connectivity to carry out computing, battery life will always be a major concern for users. This paper argues that energy-efficiency should be a first-class criteria for systems targeting personal environments.

We examine how energy-efficiency can be achieved in data synchronization systems targeting this environment. We present general techniques that can be adopted by any synchronization system to save energy. In addition, we develop a novel cost model, based on *weighted-energy-per-bit (E_b)* and E_b forecasts, to help devices pick the right technology, the right path and the right time to carry out synchronization. Our initial estimations indicate that up to two orders of magnitude of energy savings are possible depending on workloads.

1 Introduction

Ancient wisdom emphasizes achieving harmony in personal life without exerting too much energy. This paper aims to do just that, but in a modern context. This paper strives to encourage data synchronization systems targeting personal environments to attain the ideal goal of energy efficiency as they try to harmonize, i.e. synchronize, different copies of data spread across various user devices.

On a daily basis, users access a wide range of devices including mobile phones, laptops, desktops, netbooks, ebooks, portable music players. It is widely accepted that portable devices will be the primary computing device in the future [6]. In addition, with cloud computing gaining momentum [4] and with devices having multiple networking capabilities, such as WIFI, Bluetooth, EDGE, and 3G, usage patterns will increasingly rely on network communication. Unfortunately, network communication does not come cheap. It is a well known that network communication can greatly reduce battery life: from 300 hours to 6 for an iPhone 3G [1]. Even if battery capacity increases, network usage will always put a drain on battery life.

Also, users are increasingly using these devices to create, store, and access data. With data spread across multiple devices, servers, or cloud storage, manual data management is becoming a major chore. Users will start relying heavily on systems that automatically synchronize data across the devices and that automatically provide access to latest versions.

Unfortunately, most current systems targeting this environment [10, 11, 16, 17] overlook energy. Instead, they focus on the synchronization protocol and how to best carry synchronize the devices in order to achieve their consistency, availability, and performance goals. The few that consider energy, only look at very specific scenarios [13].

This paper argues that energy-efficiency should be a primary criterion for data synchronization systems targeting personal and mobile environments. To that end, it explores techniques that can be applied to any synchronization framework to enable it make energy-smart synchronization decisions. It applies these techniques to derive a novel cost model based on *weighted energy-per-bit(E_b)* for each networking technology and *E_b forecasts* to aid a device to make synchronization decisions. Our preliminary calculations indicate that two orders of magnitude of savings are achievable.

There is a significant body of work that looks into energy saving techniques at the routing layer. However, the routing level does not have sufficient information to make decisions that affect the guarantees of the synchronization system such as delaying synchronization. Hence, better savings are possible if the synchronization framework works together with the underlying layer.

We are currently applying these techniques and the cost model to build a new personal file system that takes energy into account from the ground up. We hope that our work will prove that significant energy savings are possible and encourage other systems to attain the ideal state of harmony with energy efficiency.

2 How to Save Energy

In every synchronization system a device needs to make difficult decisions about when, with what technology, and with which other device to synchronize. The fact that a mobile device often has several networking technologies, the presence of nearby peers, and the possibility of delaying synchronization give the device ample opportunities to save energy. We present three techniques that can be used to take advantage of these opportunities and save energy.

Picking the right path. Mobile devices often have multiple networking technologies. Some technologies, such as WiFi, EDGE, and 3G, can be used to connect to the Internet, whereas others, such as Bluetooth and ultra-wide-band (UWB), can be used to connect to nearby peers. Every technology has different energy requirements. Figure 1 compares the bandwidth and power consumption of several wireless technologies. In some scenarios, it may be more energy-efficient for device to use another device as an intermediary for synchronization. For example, a phone will save power if, in order to synchronize with an online repository, it uses a laptop as an intermediary, connecting via WIFI or even Bluetooth rather than the more energy-intensive 3G connection.

Picking the right time. Message batching has been commonly used to save communication energy because it reduces the energy spent in setting up and tearing down connections. By synchronizing updates as a batch rather than as soon as they occur, energy efficiency can be improved. Since the energy savings comes at a cost of increased latency, how long or how many the updates are batched depends on the level of consistency the system wants to guarantee.

In addition, since personal devices are mobile, they move in and out of areas of better connectivity. For example, when a user is on a bus to work, his phone may only have 3G connectivity. However, when he reaches work, the phone may have WiFi connectivity as well. By leveraging bandwidth forecasts [12] and holding off on synchronization when appropriate, energy-efficiency can be improved.

Keeping priorities. All data is not equal. It is not necessary to use the same synchronization policy for all data a device stores. It is possible to synchronize different subsets of data via different paths and at different times without sacrificing the availability and consistency guarantees of the system. Energy is saved because not all synchronization occurs via an energy-intensive path.

Data can be prioritized in several ways. One approach is to prioritize meta-data over data synchronization. Transmitting meta-data via more immediate means allows other devices to be quickly informed about the location of the latest version without expending the energy to send the whole update. The system can still provide the same consistency guarantees as sending the whole update [2].

Another approach is to prioritize data that is shared among multiple devices and is more frequently accessed, such as calendar entries and contacts. There is a higher chance for a user to access data that is shared among a lot of devices and, hence, should be transmitted immediately. Updates to not-so-well-shared data can be transmitted when a more energy-efficient connection is available.

3 Realizing Energy Savings

How can these techniques be applied in a real system? In this section, we derive a cost model based on the techniques and demonstrate that it can lead to energy savings in a simple synchronization system. Even though we look at a specific system, the cost model can be applied in any other synchronization system.

We consider a simple system, MAYA, in which every device stores a subset of data, called its local set, and synchronizes with a central repository accessible via the Internet. MAYA guarantees that users are able to access the latest versions of their data at most times. Hence, its synchronization policy aims to ensure that the repository either stores the latest versions of the data or knows the location of them. The key to achieving energy efficiency lies in is picking the right technology, the right path and the right time to synchronize.

We assume that power consumption for available network technologies on a device is known or can be determined by running an energy profiler [15].

	Bluetooth	UWB	Zigbee	WiFi	EDGE	3G
Max Bandwidth	3Mb/s	110Mb/s	250Kb/s	54 Mb/s	386Kb/s	3.6Mb/s
Power Consumption	103 mW	750 mW	74 mW	723 mW	1160 mW	1450 mW

Fig. 1: Comparison of different wireless connection protocols. UWB stands for ultra-wide-band. The information has been collected from various sources [5, 8, 9, 14]. Note that theoretically 3G can provide bandwidths up to 14.4Mb/s. However, most network providers currently only support 3.6Mb/s speeds.

3.1 Deriving the cost model.

Focus on energy not power. Picking the technology to synchronize with is not as simple as picking the one with the lowest power consumption. Bluetooth consumes less power than WiFi. However, because of the difference in bandwidth, it takes longer time to transmit the same amount of data via Bluetooth than via WiFi, and hence, requiring more energy [3]

Instead of using the power consumption as means of comparison, we use the *energy-per-bit* (E_b) value. Energy-per-bit calculated by dividing the power consumption of the technology by bandwidth giving us an indication of the energy used to transmit a bit via that technology. The lower the energy-per-bit value, the more energy-efficient the technology and the better its *energy-connectivity*.

Get real. Deriving energy-per-bit values from stated ideal bandwidths can inaccurate. In theory, 3G promises much faster than EDGE with a cost of 25 percent more power consumption. However, in practice, 3G may not live up to its promise. bandwidths due to the quality of the network coverage(705Kb/s for 3G vs 146Kb/s for EDGE) [7]. Similarly, bandwidths achieved via WiFi networks vary greatly [12]. In order to have a more accurate picture, a device determines the energy-connectivity, E_b, for each of its available technologies periodically by measuring the bandwidth achieved to connect to the repository via that technology. At any one time, the minimum of all E_b values is denoted as *rep-connectivity* $RepE_b$. Rep-connectivity is an indication of the minimum energy required to send a bit to the repository directly.

Since measuring real rep-connectivity expends energy, there exists a fine balance between accuracy and energy spent on measurement. In some scenarios, the stated values may be close enough to the real values eliminating the need for real-time measurement. On the other hand, if real-time measurement is a must, then it is possible to save energy by piggy-backing connectivity measurement with synchronization.

Count on peers. In some scenarios, it may be more energy-efficient to leverage peer connectivity. For example, in the absence of WiFi, energy can be saved by using Bluetooth to connect to a nearby laptop with a wired connection instead of transferring data to the repository via 3G. Therefore, in addition to keeping track of local E_b values for connecting to the repository, a device keeps track of E_b values to connect to nearby peers and their $RepE_b$. If the peer has better rep-connectivity and it takes less energy to connect to the peer (i.e. a peer's $RepE_b$ and the E_b to connect to it are smaller than local $RepE_b$), then the peer should be used as an intermediary for synchronization.

Don't be taken advantage of. For a device with excellent rep-connectivity, being used as an intermediary has two repercussions. On one hand, it takes its toll on battery life, and on the other hand, it reduces the latency to receive latest update if the data being routed is part of the device's local set. Hence, when a peer inquires about a device's rep-connectivity, the device reports three values: its local set, the actual rep-connectivity, $RepE_b$, for data belonging to its local set, and third, a weighted rep-connectivity, $wRepE_b$, for data not belonging to its local set. The weighted rep-connectivity is calculated by multiplying the actual rep-connectivity with a factor inversely proportional to the remaining battery life. Hence, as the battery life of a device decreases, the less likely it is for the device to be used as an intermediary for data that is not in its local set.

Not everything needs to be done now. Not all updates need to be immediately transferred to the repository. By assigning priorities to data and holding off transmission of some data until better connectivity is available, energy can be saved.

In order to ensure that the system achieves its guarantees, meta-data is given the highest priority. Since the meta-data is smaller, it quickly reaches the repository and other devices, and hence reduces the time window for conflicts. Data that is not widely shared is assigned lower priority because there is less chance that it will be accessed from another device within a short period of time. Even if that happens, the meta-data informs the user of the possibility of conflict.

The key idea is that data with lower priority can be delayed for a longer time for better connectivity. We leverage the techniques use for bandwidth forecasts [12] to get energy-connectivity forecasts. Every priority level is associated with a hold-off time that is inversely proportional to its priority. If the predicted energy-connectivity after the hold-off time is better than the current energy-connectivity by a threshold, transmission is delayed. At the end of the hold-off time, all the updates that were delayed at that priority level are transmitted. Another option is to vary the threshold for each priority level so that a higher priority update is delayed only if the predicted energy savings are larger.

Putting it all together. A device maintains the following information:

- Energy-connectivity for every wide-area connection that is re-evaluated periodically or when there is a network change.
- Rep-connectivity, which is equal to the minimum of all energy-connectivities and the weighted rep-connectivity, which is equal to the rep-connectivity divided by the remaining battery life.
- For every peer, energy-connectivity to the peer, the peer's local set, rep-connectivity, and weighted rep-connectivity
- The priorities for different subsets of data and the maximum hold-off time for each priority level.

The device uses the following protocol for sending new updates to the repository: It first determines whether the updates can be held off by evaluating the priority, the hold off time, and connectivity forecasts. When it is time to send the updates, if there are no peers available, the device uses the technology with the best energy-connectivity. But in the presence of peers, the device compares its local set with that of a peer and its rep-connectivity with the peer's rep-connectivity or weighted rep-connectivity accordingly. It uses the peer as an intermediary if it is more energy-efficient to do so.

	Energy-per-Kilobit
Bluetooth	0.034 mJ
UWB	0.007 mJ
Zigbee	0.296 mJ
WiFi	0.013 mJ
EDGE	3.005 mJ
3G	0.402 mJ

Fig. 2: Energy-per-kilobit values of different technologies. Energy-per-kilobit is equal to bandwidth divided by power. It is an indication of the energy required to send 1Kb of data.

When a device wants to retrieve new updates from the repository, it determines whether the data it requires is in the local set of a peer and whether it is more energy-efficient to connect to the peer. If so, it tries to retrieve the updates from the peer. Otherwise, it uses the technology with the best energy-connectivity to retrieve updates from the repository.

3.2 Example

One evening, I go on my daily stroll and I notice a very pretty flower. Since I don't have my camera with me, I use my phone to take a picture of it. Because I have installed MAYA, I expect that the picture will automatically be sent to the central repository and my laptop at home. MAYA has several options: It can either transfer the picture immediately to the repository via EDGE or 3G, or wait till I get home and use the WiFi or Bluetooth connection and use my laptop as an intermediary. Figure 2 shows the energy-per-kilobit value for each of these technologies. If MAYA waits and uses WiFi to transfer the photo, it uses 30 times less energy than 3G and 230 times less energy than EDGE. In fact, in the future, if my phone has UWB capabilities, it would use 420 times less energy than EDGE.

4 So What Next?

Even though we have presented techniques any data synchronization system can employ to make energy saving decisions, there are several questions that need further investigation:

- How significant are the energy savings? Is the battery life lengthened by seconds, minutes or hours?

- How easy it is to come up with a simple algorithm that works well? Is the extra complexity worth it?
- How much consistency needs to be given up in order to gain decent energy savings?
- Can these techniques be extended to achieve energy efficiency in other applications?

These questions can be only be answered with real data. We are currently implementing a personal file system that uses the *weighted energy-per-bit* model to make synchronization decisions. The system is being built with PADS [2] because it simplifies implementation and experimentation of different synchronization policies. We believe our work will shed light to these important questions.

We also intend to extend the synchronization model take into account the latency of different technologies and the monetary cost involved in synchronizing with cloud storage.

5 Conclusion

The paper argues that energy efficiency should be considered as a first-class criterion when designing systems for personal and mobile environments. It focuses on energy efficiency in data synchronization system. It explores several techniques that can be used to save energy in data synchronization system. It proposes a novel *weighted energy-per-bit scheme* that can be used to make energy-smart synchronization decisions. We are currently employing the cost model in a personal file system in order to determine whether the extra complexity leads to significant energy savings.

We believe that the techniques presented and the lessons learnt from this work are valuable for other systems targeting this environment.

References

[1] Apple - iphone - technical specifications. http://www.apple.com/iphone/specs.html.

[2] N. Belaramani, M. Dahlin, A. Nayate, and J. Zheng. PADS: A Policy Architecture for building Distributed Storage systems. In *Proc NSDI*, Apr. 2009.

[3] How bluetooth, uwb, and 802.11 stack up on power consumption. http://www.wirelessnetdesignline.com/howto/207200448.

[4] Portable devices give cloud computing more clout. http://www.reuters.com, Dec. 2008.

[5] Eric's corner: Which uses more power? wifi or 3g? http://ericscorner.blogspot.com/2008/11/which-uses-more-power-wifi-or-3g.html.

[6] Future of the Internet III. http://www.pewinternet.org, Dec. 2008.

[7] How to iphone speedtest 3g vs edge vs wifi. http://www.iphoneincanada.ca/tips-tricks/how-to-iphone-speedtest-3g-vs-edge-vs-wifi/.

[8] J. S. Lee, Y. W. Su, and C. C. Shen. A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wifi. In *IEEE IECON*, Nov. 2007.

[9] List of device bandwidths. http://en.wikipedia.org/wiki/List_of_device_bandwidths.

[10] Live mesh beta. http://www.mesh.com.

[11] J. Mazzola, P. David, S.Tom, and Y. K. Chen. Footloose: A case for physical eventual consistency and selective conflict resolution. In *IEE WMCSA*, 2003.

[12] A. Nicholson and B. Noble. Breadcrumbs: Forecasting mobile connectivity. In *MobiCom*, 2008.

[13] E. Nightingale and J. Flinn. Energy-efficiency and storage flexibility in the blue file system. In *Proc. OSDI*, Dec. 2004.

[14] No 3g on the iphone, but why? a battery life analysis. http://www.anandtech.com/gadgets/-showdoc.aspx&i=3036&p=3.

[15] Nokia engery profiler. http://www.nokia.com.

[16] V. Ramasubramanian, T. Rodeheffer, D. B. Terry, M. Walraed-Sullivan, T. Wobber, C. Marshall, and A. Vahdat. Cimbiosys: A platform for content-based partial replication. Technical report, Microsoft Research, 2008.

[17] S. Sobti, N. Garg, F. Zheng, J. Lai, E. Ziskind, A. Krishnamurthy, and R. Y. Wang. Segank: a distributed mobile storage system. In *Proc. FAST*, pages 239–252. USENIX Association, 2004.