Simulating Power Usage in a IOT-based Smart Home Lighting

Ashton Knight, Carter Casey, Chase Conley

Seth Drew, Thomas Rind, and Walton Lee

Tufts University, Comp 50/150-Internet Of Things

***Abstract*— As microcontroller technology becomes increasingly cheap, power efficient and mass-producible, the field of Internet of Things (IOT) has taken off. "Things" are more easily configurable than ever, and can be used a wide variety of complete Internet of Things (IOT) applications that at one time were infeasible because of startup and maintenance costs. This paper analyzes the power requirements for maintaining a home or work space lit exclusively by smart lights that turn on and off according to lighting needs in each room.**

***Keywords*— Power, Internet of Things, Smart Lighting, Smart Homes, Microcontrollers**

Introduction

Many new commercial buildings nowadays have some sort of automatic lighting, controlled by sensors in each room that time out when they do not receive activity for a certain amount of time. In homes on the other hand, mostly all lighting is still done manually. This means that household power usage for lights is determined by a person's habits and remembering to turn on and off their lights.

Furthermore, when a person walks into or out of a room in their house, they need to think about turning on or off the lights. This kind of manual lighting control has been such an integral part of our lives that we may not even notice anymore that our interaction with lights in our house could actually be much more convenient and energy efficient.

Applying the Internet of Things to the problem of lighting is a useful application because automatic lighting streamlines an otherwise manual process. The purpose of this paper is to determine the power requirements of a fully automatic home lighting system. There are two main questions: compared to a manual system where lights are often left on, could using a smart lighting system save power in the long run? How much extra power does it take to run the micro controllers, sensors, and required network components, and would the value of convenience make up for these extra energy costs?

Simulation

Design

In order to model this system, we designed a simulation engine to handle all of the required components. The simulation centers around an event queue that each component uses to communicate with other components. This eliminates direct communication between components, allowing for easier event-wise simulation.

The simulation is initialized with a component configuration file, detailing each component and the relationships between them. An event file is also loaded that details the user-driven events, such as turning on devices from the cloud or entering a room. As the simulation runs, components create additional events to interact with the other components in the system.

Each event, whether generated by components or the event file, is time coded. As events are added to the queue, they are processed by the simulation in time order in order to ensure that every string of events occurs in the correct sequence. This structure allows for the simulation to be run quickly (i.e. not in real time), but also preserve the correct order and duration of events for calculating power usage.

When processing an event, the simulation calls the destination component's "onEvent" function. The component then processes the event and pushes additional events to the simulation to be processed by other components. This creates a modular system of events that can be tailored to the needs of each component.

As the simulation runs, each component keeps track of its power or, in the case of the cloud, dollar usage. The simulation terminates with an end event which causes each component to report its total usage back to the simulation. This data is written to an output file which can be used for reporting and visualization.

Scenarios

We first simulated simple scenarios, such as the cloud repeatedly turning a single device on and off to verify the accuracy of our models and simulation engine. Then we created a control scenario, which models a person using turning on their lights in the morning when they wake up, turning them off when they leave, turning them on when they arrive home, and turning them off again when they go to sleep. This well approximates the way a household typically uses their lights, without much thought to when they are on or off, except when leaving and going to sleep.

We compared this with a typical usage scenario for one person using the system. They turn their bedroom light on with their smartphone when they wake up, the all of the other lights automatically turn on and off depending on what room they are in. A similar process occurs when they return home for the day, ending with them turning their bedroom light off with their smartphone.

Device

Design

For our devices in our simulation, we designed our device class to model the behavior of a single connected light bulb, while keeping track of both active and passive power consumed. The class is designed to use a variable fraction of the device's maximum power consumption. In the context of an IOT lightbulb, variable power consumption would mimic the behavior of a dimmable light. Having variable maximum power consumption allows the device to represent not only different kinds of bulbs but different appliances within a connected household. The construction variable for a light bulb object include max lumens, lumens per watt, and standby power. This way, a wide variety of lights can be modeled, ranging in overall brightness, efficiency, and power consumption of the network connected electronics. This even allows standard incandescent bulb to be modeled by setting max lumens to 800, lumens per watt to 14, and no standby consumption.

In order to emulate a connected device within our simulation, our device will both send and respond to different events for varying functions. For example, upon registering a motion event the device class will trigger a brightness control event, changing the amount of power consumed. The device is also contains built in delay values to simulate processing time in a real-life environment. All events also trigger messages to the cloud logging changes in the device's behavior.

Accuracy of the device simulation is extremely important as variations in power consumption of the devices should account for the majority of energy and personal cost saved. As such, we ensured our devices were highly customizable to more accurately model a connected household.

Network Interface

The device models network interaction by packaging any event which must be sent to the cloud inside a network send event that only specifies the network protocol. This way, the details of the protocol are abstracted away send from the device so the network model can handle the delay and power, and the details of what kind of event is being sent to the cloud is abstracted away from the network, because it is contained in the parameters of the network send event.

Power

The power considerations of the device are very important for this model. Using some typical values[5] of smart lightbulbs, taking into account their max lumens, lumens per watt, and standby power, as mentioned before, the power consumed by the device can be calculated by

$$P = \frac{B * L_{max}}{LPW}$$

Where B is the brightness coefficient, L_{max} is the maximum lumens, and LPW is lumens per Watt. The model includes a method for extracting the total power usage (kilowatt-hours) of the device, which it calculated by creating a running left Riemann sum of its power usage, which is recalculated every time the brightness level changes.

Network

Design

In our simulation, the simplest interpretation of the network is a unit that passes events from one device to another, consuming power and adding delay. The amount of power consumed - and delay added - largely depends on the protocol used: a protocol like UDP will consume less power than a protocol like TCP, which requires an active connection with acknowledgments to transmit messages.

To this end, our implementation utilizes a system of network nodes, one per network connected device, that handle the transmission of messages. Each NetworkNode knows how to route to any other node in the network (including the cloud). Additionally, there is no concept of network lossiness. A message, once sent, will be delivered.

Device Interface

In order to simplify the interface between the network and the device, a network-enabled device need only register a single NETWORK_SEND event with the simulator. Similarly, a NetworkNode will register an event received over the network with a NETWORK_RECEIVE event, which will contain the new event to be run on the receiving device.

Protocols

There are currently two implemented protocols in our simulation: UDP and TCP. Using UDP we can approximate both typical UDP use-cases, as well as local transmissions, which in a smart home we could implement with negligible loss. TCP is the standard choice for remote connections, and on the whole should be used for communications with the cloud, or any other service that uses a REST or similar API.

UDP is implemented as one would expect: a single UDP "packet" is transferred to each successive device in the network path. TCP is an approximation of the real protocol - a handshake of SYN SYN-ACK ACK is conducted, then a message is sent. As each event is sent individually (which isn't too far off from the limited communications we'd see between simple IoT devices) this should suffice in approximating TCP overhead.

Power

Power consumption by the network can be roughly divided into two states: active high consumption, and idle low consumption. Actively sending and receiving data requires considerably more power than simply listening for new data to come in, but the latter still requires energy. Our simulation estimates network power consumption by computing the amount of time spent in active and idle states.

Referring to an overview of power consumption in a smartphone [3] we can get a rough estimate of the power consumption used by a network connection. The peak consumption during a network-active test was measured at approximately 700 mW. Idle consumption was considerably lower, but still present: we'll estimate it to be about 50 mW.

Cloud

Design

The Cloud is accessed in two instances. The first instance is if a motion event occurs, and the second is if a user updates the brightness. The motion event will have the cloud turn on whichever light detected motion to a default brightness. In the other case the user tells the cloud that they want a certain light to be a certain brightness. In this case the user could communicate through a simple app to reach the cloud.

In the instance that there is a motion event the cloud will receive a message from the network. The cloud will add this event to its log and create a Brightness control Event with a default brightness that it sends to itself. It will also do a write cycle, since it is logging the event. It is important in this simulation to keep track of read and write cycles because they are involved in calculating the cost one user has on the cloud. Sending a Brightness control event to itself will have the cloud create a payload for the light that detected motion. That payload will be sent through the network through a node to the light. This will count as a read cycle, which is important for the user cost of the cloud.

In the instance of a user updating the light brightness, this will trigger a Brightness control event, except instead of a default value like in the motion event, the brightness value is determined by the user. This will create a payload that is sent through the network node to the specific light. This will count as a read cycle.

The cloud simulation also adds a delay to the execution time, to represent the calculations and communication with the network. For this simulation the delay was 200ms, as the response time of the cloud should be no greater than 200ms [4], and simulation is trying to not just represent the best case scenario of power costs.

Power and Cost

In this simulation, the amount of read and write cycles are kept track of. This can be used to calculate the cost of the cloud per person. By checking amazon web services, the cost of the Write Throughput is $0.0065/hour for every 10 units of Write Capacity. The Read $0.0065/hour for every 50 units of Read Capacity [2]. Since the packages needed to be sent are small, just containing brightness values, sources and destinations, and light ids, it can be assumed that a unit of capacity can handle each package. So, by keeping track of how many times a read and write occur, the total cost for one person can be calculated. This cost could be used as a rough indicator of how much power one person on the cloud uses. The cost should be able to cover the affect one person has on the cloud.

A cloud does add some additional power constraints to an IOT light control system. This system could be limited to just the local network. One benefit to this would be that the system would be more secure. In addition, there would be less delays when communicating from device to the network and no energy will be used to communicate with the cloud. A disadvantage would that a person needs to be one the local network to use the system. The cloud allows a user to control the lights from anywhere they can send a signal. So, if a user is away from their home and realizes they forgot if they left the lights on, they can check and if the lights are on, turn them off. While there are delays and additional power usages, the cloud does add value to the system. The cloud can also reduce the amount of power going to the lights.

If a user is constantly changing the brightness, the power usage of the cloud will be higher as a result of the numerous read cycles that would occur. Power usage would also be higher if the motion detection was easily triggered. If something moved back and forth through the motion detector as the light was turning off, then the cloud would receive constant write and read cycles.

To limit the amount of write and read cycles on the cloud measures can be taken. Having motion detection that accurately identifies a person would stop unnecessary Motion Events. This can lead to an increase in a delay for the lights to turn on, but allows for lower power usage.

The cloud allows for power saving opportunities, but the user must be partly responsible for those. They should not make numerous changes to the brightness of each light for no reason. They also can turn lights on and off remotely. This can limit the light usage and the amount of read cycles. The accuracy of the device will also limit the cloud power usage. By having accurate motion events the cloud will only need to perform write cycles when necessary.

Results

See results summary at the end of this paper

Future Work

We have described a framework for a rather in-depth power analysis of an IOT home lighting system, given parameters detailing the efficiency and brightness of each light, the network protocol and route, and a cloud interface, and using data describing movement patterns of inhabitants in the home. Where this experiment has the most opportunity to expand is in this data, and how it is constructed. Using data collected by observing a real home could give extremely robust and interesting results, as the simulation could calculate exactly how different lighting strategies could affect power consumption in the scenario for which the data was collected.
Furthermore, the simulation could take into account a dynamic lighting system, where the user specifies the desired overall brightness of a room, and a home computing system takes into account ambient light entering from outside to run indoor lighting more efficiently.

Distribution of Work

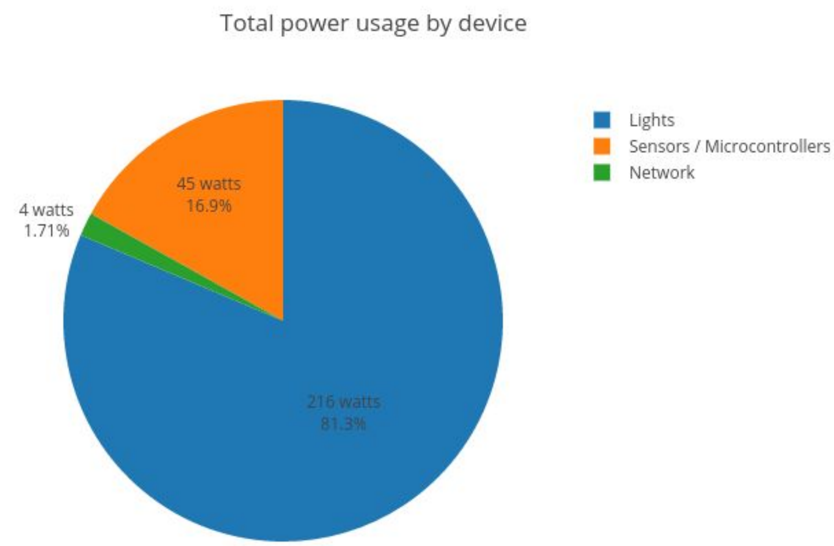
Ashton:	Device module
Carter:	Network module
Chase:	Simulation module
Seth:	Simulation module
Tom:	Cloud module
Walton:	Device module

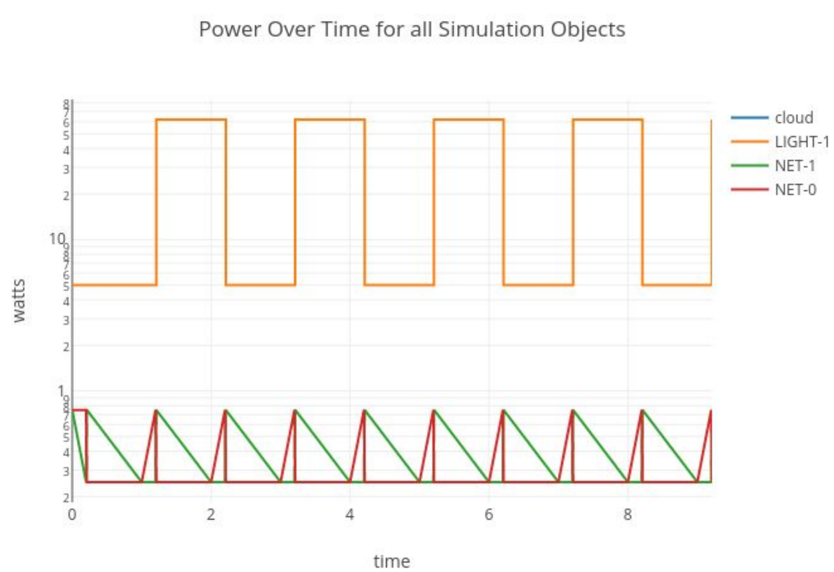
References

1. The National Appliance and Equipment Energy Efficiency Committee, *Australia's Standby Power Strategy 2002-2012*. Canberra, Australian. Web. http://www.energyrating.gov.au/sites/new.energyrating/files/documents/sb200411-sensors_0.pdf.
2. "Amazon DynamoDB Pricing." Amazon Web Services, Inc. N.p., n.d. Web. 19 Dec. 2016.
3. Carroll Aaron, Gernot Heiser, *An Analysis of Power Consumption in a Smartphone*. NICTA and University of New South Wales. Web. https://www.usenix.org/legacy/event/atc10/tech/full_papers/Carroll.pdf.

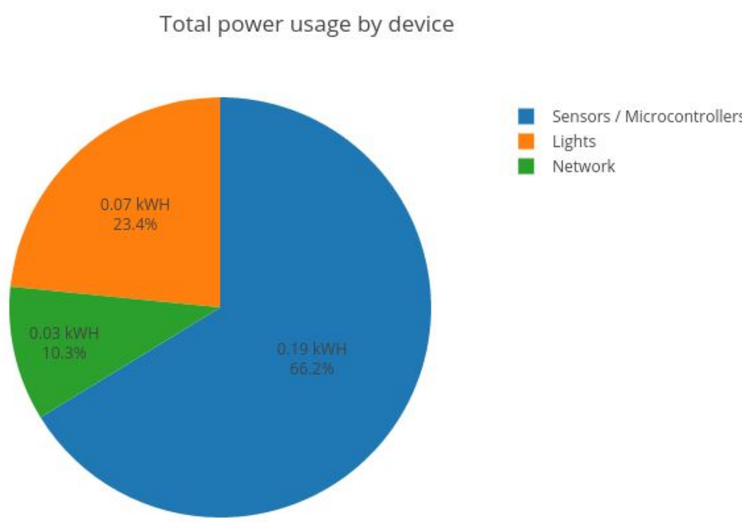
4. Kent, Phillip. "How to Improve Cloud Latency and Throughput: How Good Is Your Cloud Provider's Network?" *Interoute CloudStore*. N.p., 25 Nov. 2015. Web. 19 Dec. 2016.
5. Smart Lamp Testing - Initial Results (n.d.): n. pag. Electronics Devices & Networks Annex. International Energy Agency, Nov. 2014. Web. 19 Dec. 2016. http://edna.iea-4e.org/files/otherfiles/0000/0100/Smart_Lights_Paper_for_EDNA_Website_v3.pdf

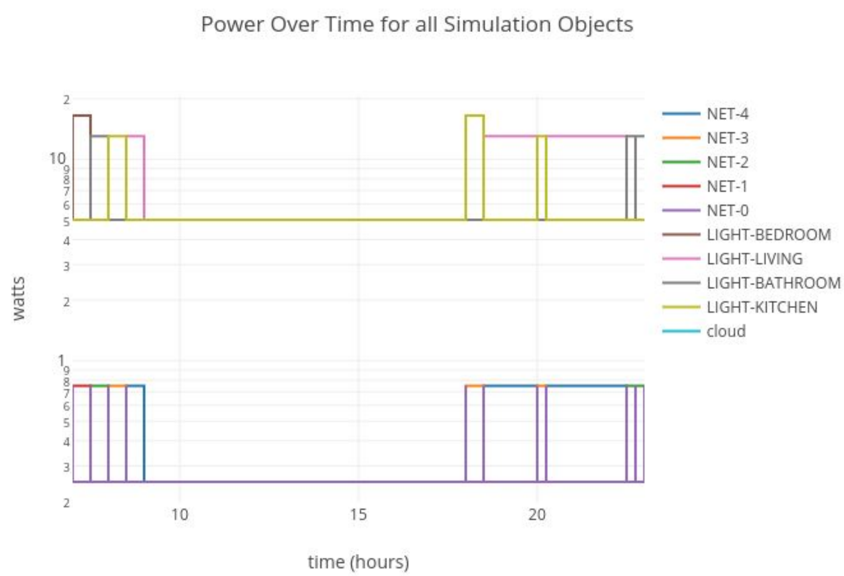
Results



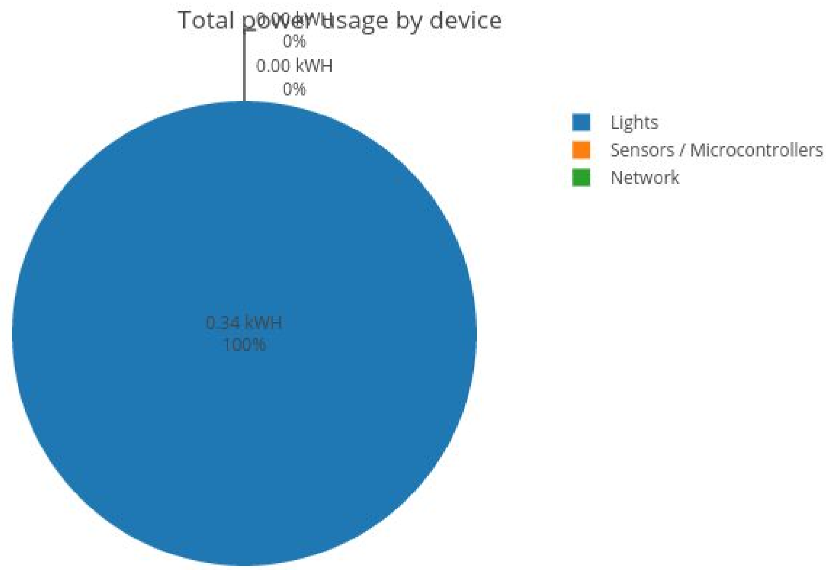


This was one experiment in which we turned on and off one light repeatedly using the network, to get a sense of scale. This chart shows watts on a logarithmic scale vs time in seconds. The network is at least an order of magnitude less power intensive than running the lights and the microcontrollers on the IOT lights. The cloud's power consumption is not taken into consideration.





This experiment was run mapping one person's movement throughout the house, and how our IOT lighting system would react to that. We can see in the wattage over time chart, that different lights throughout the house turn on and off as the person walks around their house. The large inactive period is in the middle of the day, when the person has left for work, and the tall section at the end is when the person returns. Notice that there are different types of lights installed throughout the house, and that shows up in that some of the power consumption bars are higher for some lights than the others. For example, the LIGHT-KITCHEN and LIGHT-BEDROOM often consume more power (or are turned on brighter) than the LIGHT-BATHROOM or the LIGHT-LIVING.



Our control experiment is run how we thought a person might use their lights. One person walks into their house, and turns on all the lights in the house, until they go to bed and turn them all off. Although this is likely a generous estimate for how much the lights are on when a person is manually controlling them, we felt that it was reasonable that, especially at night, a person would turn on their lights upon arriving at home, and not change them until leaving for bed. In total, the kWH in this experiment for the lights alone was enough to make up for the energy of running the sensors/microcontrollers all day.

Conclusion

In general, we see that the power required for a smart home and for a regular lighting system are quite similar. When people are out of the house during the day, the sensor systems required tend to dominate the power consumption, but with smart hibernation of sensors and microcontrollers when no activity is detected, this energy requirement could be significantly decreased. If the initial cost for transition to smart lighting is not overwhelming, then smart technology could save power and money in the long run.