Title
Low-cost Appliance State Sensing for Energy Disaggregation

Author
Wu, Tianji

Publication Date
2012

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Low-cost Appliance State Sensing for Energy Disaggregation

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical Engineering

by

Tianji Wu

2012

Abstract of the Thesis

Low-cost Appliance State Sensing for Energy Disaggregation

by

Tianji Wu

Master of Science in Electrical Engineering

University of California, Los Angeles, 2012

Professor Mani B. Srivastava, Chair

Fine-grained per appliance electrical energy consumption data is crucial to electrical energy conservation. However, energy meters are installed at few central points in buildings, providing only aggregated energy consumption data. Therefore, people are seeking ways to get disaggregated energy information. A key issue and most challenging problem in energy disaggregation is to know the power state of each appliance. We design a sensing system that can reliably keep track of the binary (on-off) states of appliances.

In our system, the sensors are designed to be deployed at each appliance. However, we are able to minimize the hardware requirements so that the sensors are inherently low-cost. The whole system is totally affordable for large scale deployment. The evaluation shows that, despite the simplicity of hardware, the system can keep track of the power state of tens of appliances at 99.5% precision and recall with a single base station. We also propose an energy disaggregation approach, based on the information our sensor network provides combined with central power meter readings.

The thesis of Tianji Wu is approved.

Mario Gerla

William J. Kaiser

Mani B. Srivastava, Committee Chair

University of California, Los Angeles

2012

Table of Contents

List of Figures

List of Tables

Acknowledgments

In my two years of study and research, I am always feeling grateful to all members of NESL. NESL is a place where people are passionate, freely discuss about the latest technologies, creative ideas, etc. Instead of keeping as words, people do realize those ideas, no matter big or small, passionately. If I had not come to join NESL, I would never have learned so much cutting edge science and technology knowledge and trends and views. If there was only one thing I would miss when I return to my home country, I would miss the geeky environment of NESL.

Among all, I would especially thank Professor Mani Srivastava, my advisor. Each time when I talked with him, either about academic problem or random ideas, I could feel the inspiration from his broad and deep knowledge and incisive and sharp view, both at architectural level and implementation level. Besides a great academic advisor, he is also a great mentor, helping us to set high expectations in life.

I would also thank my collaborator Kanthi. Although we have worked together only for couple of weeks, her inspiring ideas and great work have been important to this work. Also thank all NESL members whom I bothered with deploying the sensors, and Fe, without her hard work the whole lab can not run smoothly.

Finally, I would express my appreciation to my parents, who, despite thousands of miles away, never stop thinking of me; to Boss Wangs, who took care of everything back in China so that I could focus on this work; and to my beloved Catherine, who brings warmness and tenderness to the strict and busy world.

CHAPTER 1

Introduction

Electricity has become one of the most important form of energy in the modern age. In 2010, 41% of the total energy consumption in the United States is consumed to generate electricity. Among the major sectors, residential and commercial use accounts for more than 70% of the total electricity end use, and it has been continuously growing for the past 40 years, even when consumptions of other major energy sources, such as natural gas and coal, have become stable over time. [US11]

Traditionally, electricity utility companies report only the aggregated energy use to each household or commercial entity, and electricity power meters are usually installed at few central points. However, in order to conserve energy, accurate and fine-grained energy consumption information is crucial. Hopefully, people can get energy use reports telling them how much energy each appliance consumes, and then they can be able to manage the usage of the appliances accordingly in order to conserve energy. Studies have shown that simply providing instantaneous and disaggregated energy consumption feedback to people can help them conserving energy [Dar06, PHM06, Fis08].

There are several ways to disaggregate energy use into finer granularity. One approach is to install power meters for each and every appliance that we are interested. While it can provide the most accurate results we can expect, this method is too expensive in term of the cost of metering devices and the difficulty of deployment. There are several commercial plug load power meters, such as *Watts up?PRO* and *Kill-A-Watt*. Low-end products with LCD display are usually tens of dollars each, while high-end products with networking capabilities are hundreds. In a typical household where there are tens of appliances, the total cost could be easily reach thousands of dollars. Moreover, these meters usually have a big physical

dimension compared to AC outlets and plugs, making it inconvenient to deploy house- or building-wide.

Due to these limitations of the direct metering method, a kind of new approach called nonintrusive appliance load monitoring (NILM or NIALM) has been developed, and it has attracted great attentions in recent years. The main idea is to use signatures observed solely at the central power meter to infer the state and energy usage of each appliance. We leave detailed discussion about NILM to Chapter 2. This kind of method is promising because it requires minimal deployment efforts. Only a smart central meter is needed and all inferences are based on the signals observed at the central meter. It works well especially with high-consumption appliances such as washing machines, dish washers, microwaves. However, its accuracy degrades significantly when the number of appliances increases. In fact, most of these methods can not accommodate more than 6-7 appliances. They also have trouble differentiating appliances with similar signatures, or appliances that are of the same model.

A hybrid solution can help remedy the problem. In this paper, we will show that networked sensors deployed at each appliance can assist an intelligent central meter to infer the states as well as energy use of the appliances. We designed a very-low-cost sensor node that can sense and transmit the binary on-off state of an appliance attached onto it. We will show that with such under-designed sensors, the cost can be lowered into an acceptable range for large-scale deployment, and yet they can still provide much crucial information for energy disaggregation. In our experiments, we deploy 20 such sensors, and are able to identify and track the state of all appliances attached to them. With the power readings from the central meter, we are able to infer energy consumption of each individual appliance. The process is fully autonomous, without the need for explicit training.

The paper will be structured as following. In Chapter 2, we will discuss related work, including various energy disaggregation approaches. The hardware of the sensor we developed will be discussed in Chapter 3. Because we use an unidirectional radio in our system, a specially designed network protocol is presented in Chapter 4, as well as the evaluations. In Chapter 5, we will discuss our experiments done in a real lab setting. An energy disaggregation algorithm that incorporated with our sensing system is presented as well. Chapter 6

concludes the work.

CHAPTER 2

Related work

Direct sensing, i.e. metering at every appliance, is the most accurate approach we can expect towards energy disaggregation. Yet it is too costly. Commercial products range from low-end LCD display meters (e.g. Kill-A-Watt) that cost around 20 dollars to high-end networked meters (e.g. Watts up? PRO) that cost around 200 dollars. There are also great research platforms such as ACme [JDD09] which has metering and control capability and wireless connectivity. However, the 60-dollar cost still makes it unaffordable for large-scale deployment other than research purposes.

Another kind of approaches attract much attentions in recent years, namely non-intrusive appliance load monitoring (NILM), which is the opposite extreme to direct sensing, i.e. having no sub-meters, and putting all the intelligence in a central meter.

NILM, or most of energy disaggregation methods without direct metering, have generally three phases: 1) power state change detection, i.e. detecting that one or more appliances' power consumption has changed. Later, we would use *event* and *power state change* interchangeably. Most appliances have few discrete power states. For example, a microwave oven goes from off to on, or an electric fan goes from high speed to low speed. There are also some appliances that do not have discrete power states, such as a battery charger whose consumption varies continuously during the charging cycle, or a computer whose consumption varies with different workload. 2) Appliance identification, identifying the appliance that has made the events. This needs some prior knowledge about the signatures of all appliances interested. In some NILM algorithms, this phase is done together with the first phase. 3) Estimation of energy consumption for each appliance, i.e. keeping track of disaggregated power of each appliance. This is based on the previous two phases.

The methods of NILM can be generally classified into two categories. Some methods analysis steady states, such as apparent power, real power (P), reactive power (Q), power factor (PF), etc. These values are usually obtained at a low sampling frequency, typically less than 1Hz. However, some steady states also need high frequency sampling, for example higher harmonic of current trace or power trace.

The original NILM method [Har92] falls in this category. Step changes in aggregated real and reactive power (P and Q) are detected and clustered, so that similar changes are identified to be associated with the same appliance. This method faces the limitation that some appliances have very similar signature on the P-Q plane. Moreover, for non-linear appliances, the reactive power is not linear, meaning that the change in reactive power measured at the central meter is not the change of reactive power of the appliance. Other algorithms such as genetic algorithm and dynamic programming has also been used [BV04a, BV04b]. They use P and Q or just P as their signatures. Some work also exploits higher harmonics in the steady state power measurements [LCS03]. This adds some new signatures to differentiate appliances. In order to reduce the number of appliances that need to be distinguished, researchers also tried pushing the meters down to circuit branch level [MHH11]. This helps detecting and identifying events because some of the large appliances are placed on separate circuits.

On the other hand, some NILM methods identify appliances by their transient signatures during power states change [LSK95, CLS06]. For example, appliances with motors (e.g. refrigerators) usually produce a large spike when turned on, due to the inductive coils. The drawback is that transient signatures are hard to detect when multiple events happen together [LSK95]. Recent research uses the Hidden Markov Model to model the power consumption of a power cycle of appliances [KJ12]. Therefore, the model captures both transient signatures and the expected length of a power cycle.

Although current NILM methods work well in certain circumstances, they still have some limitations. The most challenging problem with NILM is that when the number of appliances increases, 1) it is more likely to have appliances with similar signatures; 2) it is more likely to have overlapping events. In both cases, detection and identification of events become

significantly hard. Due to these limitations, current NILM systems can hardly work with more than 6-7 appliances. To push up the number of appliances, researchers developed a type of hybrid methods. These methods try to augment the central inferencing algorithm with some additional data collected from distributed sensors. These sensor data can hopefully dramatically improve the accuracy or scalability of the system, without raising the cost too much.

ViridiScope [KSC09] uses several kinds of sensors to tell the on-off states of appliances (assuming all appliances have binary states). These sensors include light sensors, sound sensors and magnetometers. With the knowledge of the states of all appliances and the aggregated power readings over time, we can calibrate how much power each appliance use by linear programming. In another related work [JS10], the authors are able to automatically identify where to install sub-meters in order to achieve satisfiable disaggregation accuracy. It assumes availability of reliable on-off state information of all appliances, which is actually the purpose of our work.

In this paper, we presents a low cost sensor network that can deliver reliable on-off state information of appliances. By reducing the hardware capabilities, our sensors are inherently low-cost and suitable for large scale deployment. Yet the reliability and accuracy is not compromised because of the constraints of low-cost hardwares.

CHAPTER 3

Hardware design of a low-cost sensor node

We design our own sensors for event detection. The sensor plugs in an AC outlet, and provides one AC outlet for the appliance, and communicates with a base station. Each appliance in question should connect to a separate sensor directly. In order to make it affordable for large scale deployment, we need to make low-cost a primary design goal. We achieve the goal not by choosing cheap components, but by simplifying the requirements for sensor hardware. The extremely low price is inherent in the simplicity.

The functions and design goals of the sensor nodes are

- Able to detect binary binary on-off state changes (events).
- Able to reliably deliver detected events to a central node.
- Affordable for large-scale deployment.

The hardware block diagram of the sensor is displayed in Fig.3.1. A photo of our prototype sensor is displayed in Fig.3.2.

3.1 Binary power state sensing

As discussed before, one of the function of the sensor is to detect binary (on-off) state changes. An appliance is off when it is in the lowest possible standby mode, and it is on when it is in any power mode higher than the off mode. We found that a simple threshold on power can achieve the job. Typically, in order to get power, both voltage and current need to be sampled, multiplied and accumulated. The power calculation is usually done in a

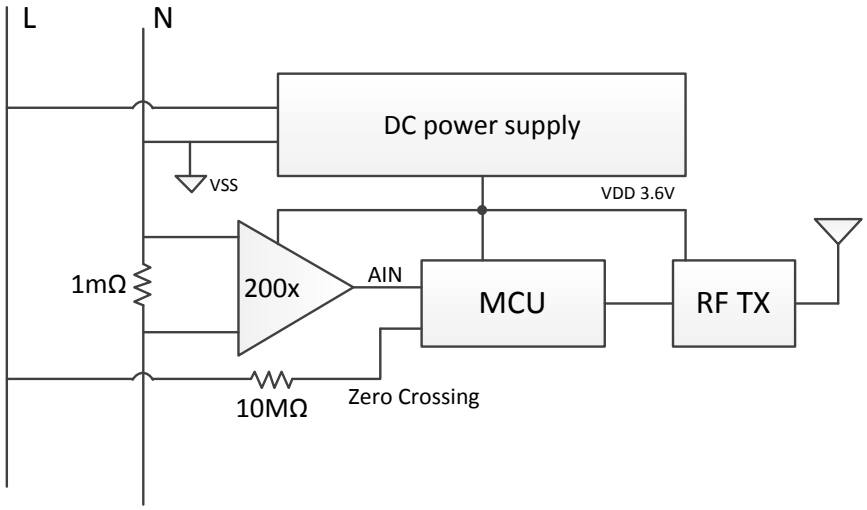


Figure 3.1: Block diagram of the on-off state sensor hardware

separate metering chip, or as a dedicated hardware module integrated in a microcontroller. In our design, we decide to sample the current only, and detect state changes solely based on thresholding current. Although there is not enough information to obtain the actual power consumption from the electrical current, our experiments show that it is good enough to determine on-off events. In this way, the circuitry can be simpler without the voltage sampling channel and the dedicated metering chip or module.

A $1m\Omega$ resistor in series with the appliance captures the instantaneous current flow. The voltage drop across the resistor is amplified 100 times by two stages of amplifier. An ATTiny10 microcontroller then samples the signal at approximately 1.6kHz, and run the event detection algorithm.

The event detection algorithm run on the microcontroller is illustrated in Fig.3.3. First, the range (i.e. max-min) of instantaneous current samples is taken for each 200-sample window. Note that the range does not necessarily represent the peak-peak amplitude of the electrical current due to clippings (discussed later). Nevertheless, the range does capture enough information for on-off state detection. We take a threshold and get an instantaneous

Figure 3.2: Photo of the sensor prototype

binary state for each window. The threshold is adjusted slightly based on the state of the appliance, so that the binarization has some hysteresis.

The threshold is hardcoded in the microcontroller firmware, and it is set to around 5W (not precise because of the uncalibrated pre-amplifier). Appliances with dry-contact switch, such as lamps and electric fans, draw absolutely zero power when turned off. Other appliances with soft-controlled switch, such as microwave and computer monitors, draw a very low standby power when turned off. Our experiments show that 5W is a reasonable threshold for most appliances.

The 200-sample window in which the range of samples is taken and binarized is about 12ms, or about 7 AC cycles long. The length is short, to ensure quick response to state changes. However, the instantaneous state is too sensitive and bouncing. Therefore, we employ a debouncing counter that counts up to 16. Then appliance state is changed only if 16 successive windows all show the opposite instantaneous state.

As mentioned before, the range of samples in each window does not necessarily represent the peak-peak amplitude of the electrical current. The reason is as following. Both the amplifier and the ADC are powered with a single positive DC power. Ideally, the negative half of the AC signals will be clipped in all stages of amplification and AD conversion. In

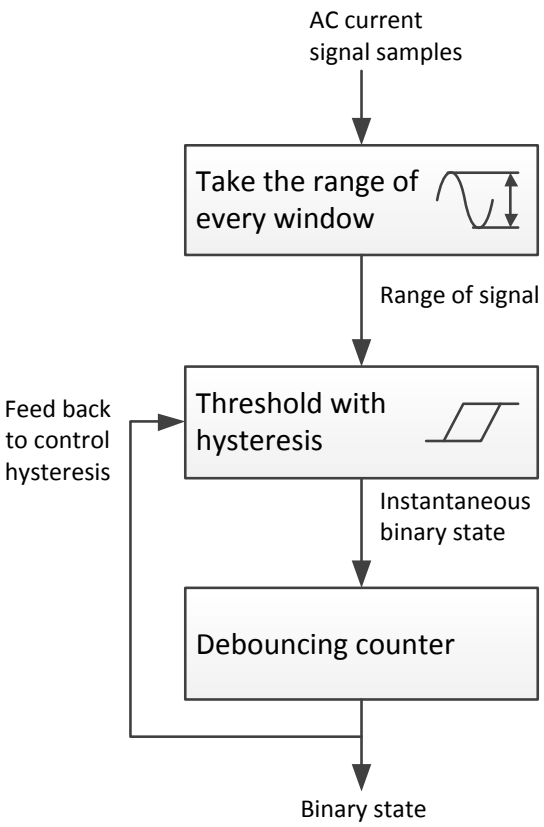


Figure 3.3: Event detection algorithm flowchart

the real circuit, some Op-Amps we use may have negative input bias voltage up to -4mV. To compensate it, we use a $2.4M\Omega$ pull-up resistor to bias the input signal.

Although the analog front-end is not designed to reduce noise and best preserve the signal, it performs well enough in detecting binary states.

3.2 Simplex radio

The sensor should be able to transmit detected events to a central base station with no additional wiring. There are several technologies for networking sensor nodes in short ranges. For instance, IEEE 802.15.4 is a standard for short range radio communication, and is widely used in sensor networking. Proprietary wireless sensor network technologies are also available such as ANT radio and TI sub-GHz radio. On the other hand, communication is also possible

through the AC power lines, namely Power-line Communication (PLC). The most popular PLC technology for home automation is X10, despite its low bitrate.

We do not use the most popular wireless sensor network radio technologies such as 802.15.4. These radio technologies usually provide several hundred kBps or even several MBps of duplex link, with sophisticated packeting and MAC, etc. These radio ICs are a few dollars each. In order to control the radio chip, microcontroller needs to have SPI or other serial link, which further pushes up the requirement of the microcontroller, as well as the price. However, in our case, appliances usually generate events once per several minutes, or even hours. And a state changing event is literally one bit of data, plus the ID of sensor and few control bits.

Low rate PLC technology is suitable for our application. However, it is not low-cost, and has other problems inherently. To couple signals onto the power lines, PLC transmitter needs coils which are costly and big in physical size. Non-linear appliances can interfere the transmission on the power lines, making the channel even worth than wireless channels. Moreover, devices on different circuit branches can not communicate without a bridge device.

Considering our requirements, we decide to use a bare-bone 315MHz ASK transmitter. The transmitter module is basically an oscillator, with one digital gate pin. To reliably utilize the radio module, we design a coding scheme (details in Appendix A). And because it is transmit-only, to improve reliability, we designed a retransmission scheme, which will be discussed in detail in Chapter 4.

3.3 DC Power supply

The sensor draws power from AC power lines with a voltage-drop-capacitor DC power supply. This is a DC supply topology with very compact design and low cost, suitable for low current DC requirements. The supply voltage of the circuit is 3.6V DC with the neutral line as reference voltage. The current is 3mA when standby, 15mA when transmitting.

We also have zero-crossing detection capability on our circuit, which is used for synchro-

nization and timing purposes.

3.4 Cost and scalability

There are several design decisions that ensures the sensor to be inexpensive inherently. By sensing the current only, instead of power, we eliminate the need for a meter IC or a microcontroller with meter module. Because only binary output is needed, distortion and clippings are tolerable. Thus, we do not need to calibrate the analog frontend. By employing a simple RF module, we eliminate the need for an RF IC or a microcontroller with SPI or other interfaces.

Without the need for any peripherals other one analog input and two digital GPIOs, we are able to use the simplest microcontrollers. The microcontroller we use in our design is ATtiny10, which has only 6 pins, 1024 bytes of program memory. Moreover, the functions of the microcontroller are 1) taking threshold and smooth it, and 2) toggle a pin for RF transmission. The first can be done with pure analog circuitry, while the latter can be replaced by a sequence generator circuitry. Hence, it is even possible to replace the microcontroller with a very simple customized IC, which is more cost-effective at large quantity.

We reuse the plastic packaging of a commercial plug-load timer to make our prototype sensors, as shown in Fig.3.2. Therefore, the physical size is limited by the packaging. In fact, the circuit board can be made much smaller. It is even possible to embed the circuitry in every plug or every outlet. The cost of all electronics components of our prototype board is 2.3 dollars each at a quantity of 1000. Including PCB, assembly and packaging, the overall cost should be as low as 4.3 US dollars. It is totally affordable to have one sensor for each appliance, considering that appliances themselves are usually tens or hundreds of dollars.

CHAPTER 4

Radio packet format and retransmission scheme

As previously discussed, we use a simplex radio in our sensor nodes. There is no receiver module in the sensors. They can not receive any kind of acknowledgement of their transmission, nor can they do carrier detection. The sensors has no means to verify whether a packet has been delivered successfully. Therefore, we need much redundancy to improve the probability of successful event delivery. Our solution is multiple retransmissions with random backup.

4.1 Radio packet format and retransmission scheme

In our design, each packet has 16-bit of payload plus one parity bit. Each packet is 64 milliseconds long, which is slightly less than 4 AC cycles (assuming 60Hz). Hence, we choose 4 AC cycles to be one timeslot for transmission.

When a sensor detects an event, it transmits 5 packets in a timespan of 2.2 seconds. These packets are fully redundant. The event is successfully delivered if at least one packet is successfully received by the base station. In the rare case that another event is detected with the same sensor before it finishes with the previous event, any further retransmission of the previous event is cancelled. In this case, we call the previous event a *flash event*. These flash events are usually caused by the transient power changes, and are not important in our case.

The sensors synchronize their transmissions to AC cycles, thanks to the zero-crossing detection circuitry.

When an event is detected, the sensor initiates a transmission at the following AC cycle,

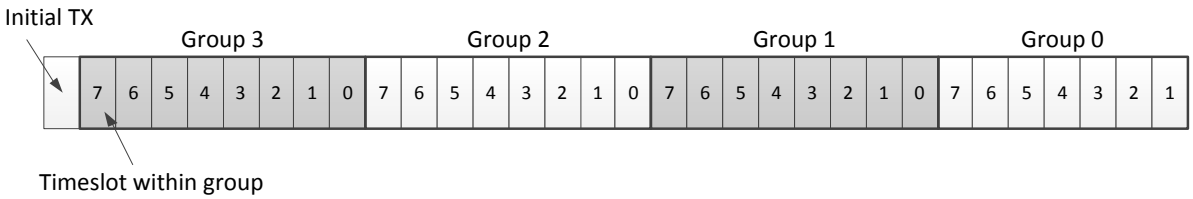


Figure 4.1: Transmission timeslots: There is an initial transmission, followed by 4 groups of timeslots. One slot is randomly picked from each group for a retransmission.

which is the start of the first timeslot. After the first timeslot, there are 31 more slots, arranged in 4 groups. And each group has 8, 8, 8 and 7 slots respectively. Fig.4.1 shows the arrangement of transmission timeslots. The sensor will randomly pick one slot from each group. Altogether there are 32 time slots, or 128 AC cycles, which is roughly 2.2 seconds.

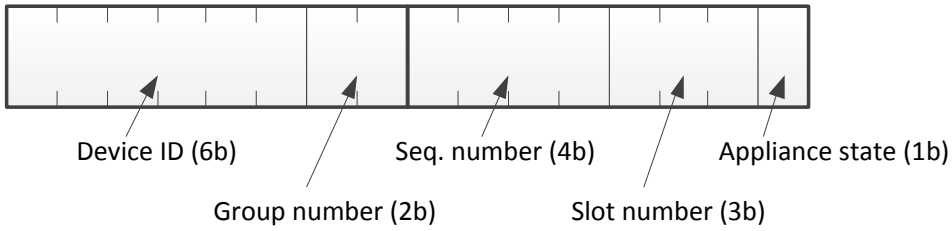


Figure 4.2: 16-bit packet format

The format of packet is shown in Fig.4.2. The initial transmission of an event will have group number 0 and slot number 0. The group and slot number are included in the packet so that the receiver can estimate the delay from the event to the transmission. The 4-bit sequence number is advanced in every packet. It helps the receiver to detect packet losses and event losses. The 6-bit device ID is hardcoded in every sensor node, allowing up to 64 sensors working in a same region with a single receiver.

4.2 Evaluation of the radio network

We evaluate the network performance in simulation and a controlled testbed. We also present the statistics of the network in the real setup.

We have two abbreviations used throughout this section: PDR and EDR. Packet delivery ratio (PDR) is the ratio of successfully received packets among all transmitted packets. Event delivery ratio (EDR) is the ratio of successfully delivered events among all detected events. Because of the redundancy of packets, an event loss happens only if all packets of the event are lost.

4.2.1 Simulation: Poisson distribution events

We develop a simulator according to the timing of our packet and the retransmission scheme. The simulator is capable to simulate transmitting and receiving at packet level in the continuous time domain. It detects packet collisions which guarantee packet losses, and mimics a constant PDR_0 due to reasons other than collisions. PDR_0 represents the power delivery ratio inherent to the deployment of the sensor. Reasons that may affect PDR_0 include transmitter power, distance to the receiver, walls and obstacles, etc. Note that because of collision, the overall PDR is less than or (in case of no collision at all) equal to PDR_0.

The testing datasets are synthesized according to the following assumptions. 1) Appliances are independent with each other; 2) Events on appliance A_k conform to a poisson distribution with parameter λ_k; 3) All appliances have the same poisson distribution parameter, i.e. $\lambda_k = \lambda$ for all k. Here λ is the average number of events in 1 second on a single appliance. And $\mu = 1/\lambda$ is the expected interval between two events of an appliance.

According to these assumptions, the overall set of events of all appliances conform to a poisson distribution with parameter $N\lambda$, where N is the number of appliances. In another word, the density of events can be adjusted by adjusting either N or λ, ignoring flash events. Taking flash events into account, the result of high N and high λ are not exactly the same. In the case of high λ, there are more flash events, and they are more vulnerable to losses

because they do not benefit from the full redundancy of having all 5 transmission. In the case of high N, there is more chance of collisions.

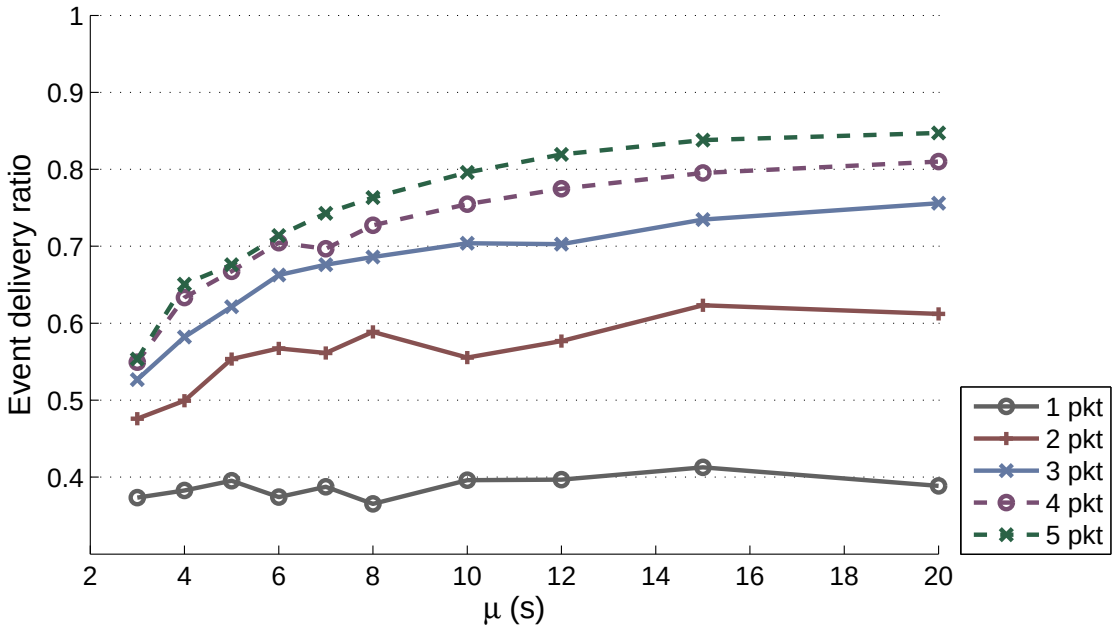
In this simulation, we synthesize the datasets with a fixed $N = 6$, and let $\mu = 1/\lambda$ vary from 3 to 20, which means the expected interval between events of an appliance is 3 to 20 seconds. Note that here μ is much shorter than what we would expect in real settings. In real settings, appliances do not frequently change state in tens of seconds. We choose shorter μ to compensate a small N, and to stress our network. Each dataset consists of 1500 events in total, 250 from each sensor.

Firstly, we evaluate the effect of having different number of retransmissions for each event. We assume all sensors have the same PDR_0. And sensors are independent with each other. Fig.4.3-4.6 shows the effect on EDR and PDR with different number of retransmissions under different μ, and with several fixed values of PDR_0.

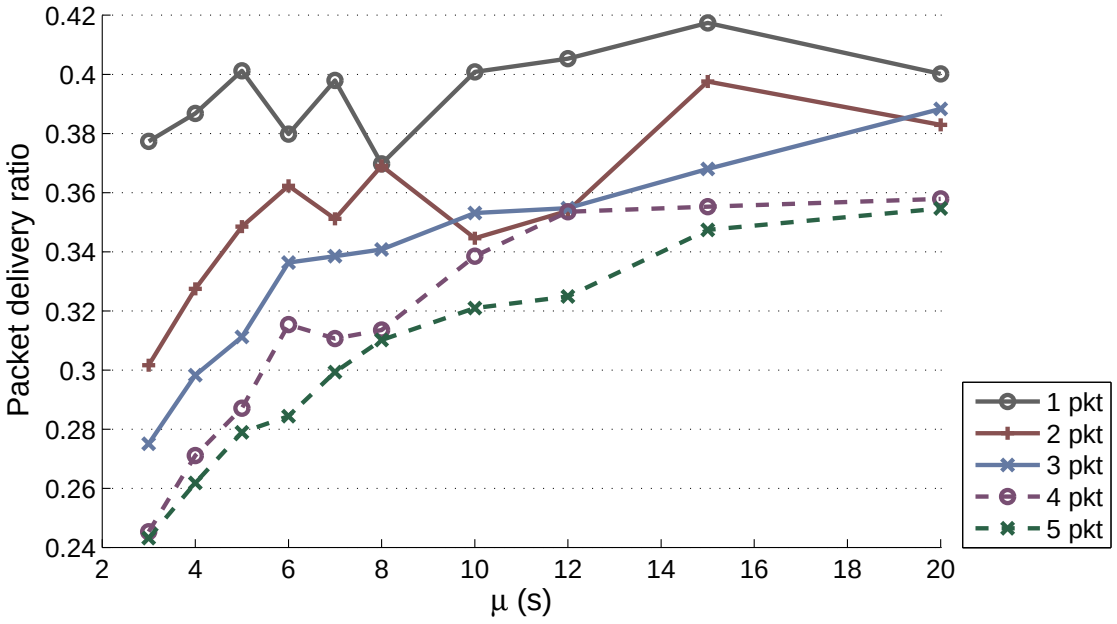
When the events are dense, i.e. shorter μ. Having multiple transmissions do not improve EDR, because the number of packets is multiplied, so does the chance of collision. When the events are sparse, the advantage of having multiple redundant transmissions becomes significant, especially with low PDR_0. This is more likely to be the case in real settings: sparse events, and some sensors have low PDR_0 because of long range or obstacles.

We know that in computer networks, having packets fitted into timeslots which are fully aligned among senders can reduce collision and improve throughput, as in slotted ALOHA [Rob75]. In our design, packets are synchronized to AC cycles. However, they are not fully aligned because each timeslot takes 4 AC cycles. Here, through simulation, we evaluate the difference in EDR and PDR with fully asynchronous, 1-cycle-synchronous and 4-cycle-synchronous (fully aligned) systems.

The result in Fig.4.7 shows that synchronizing packets to AC cycles do have slight improvement on EDR and PDR compared to the case of fully asynchronous. On the other hand, having the packet slots fully aligned improves the delivery ratios significantly, compared to the other two cases. Currently, we can synchronize to one cycle by the zero-crossing circuit. We can not employ the fully-aligned scheme unless we add other hardware capabil-

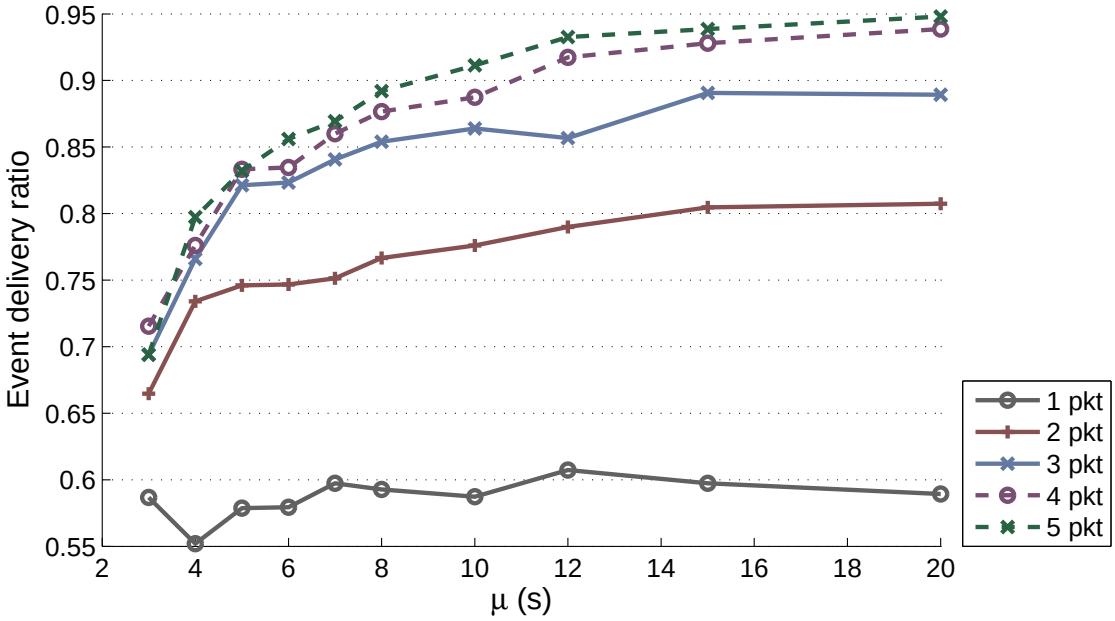


(a)

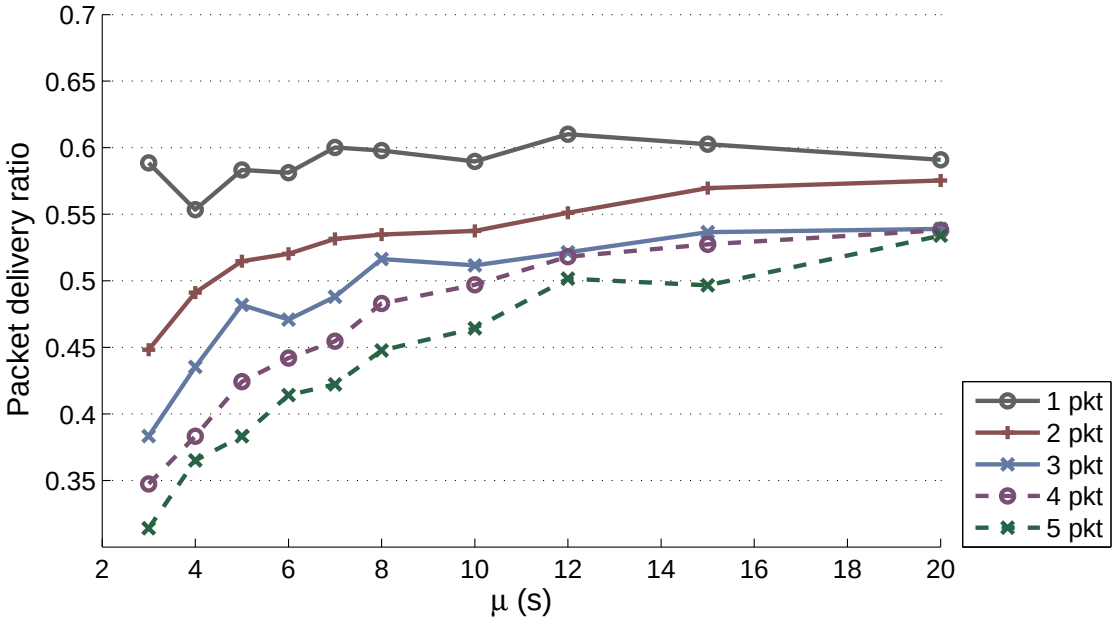


(b)

Figure 4.3: Simulation results showing EDR(a) and PDR(b) with different number of retransmissions per event, at $PDR_0 = 0.4$

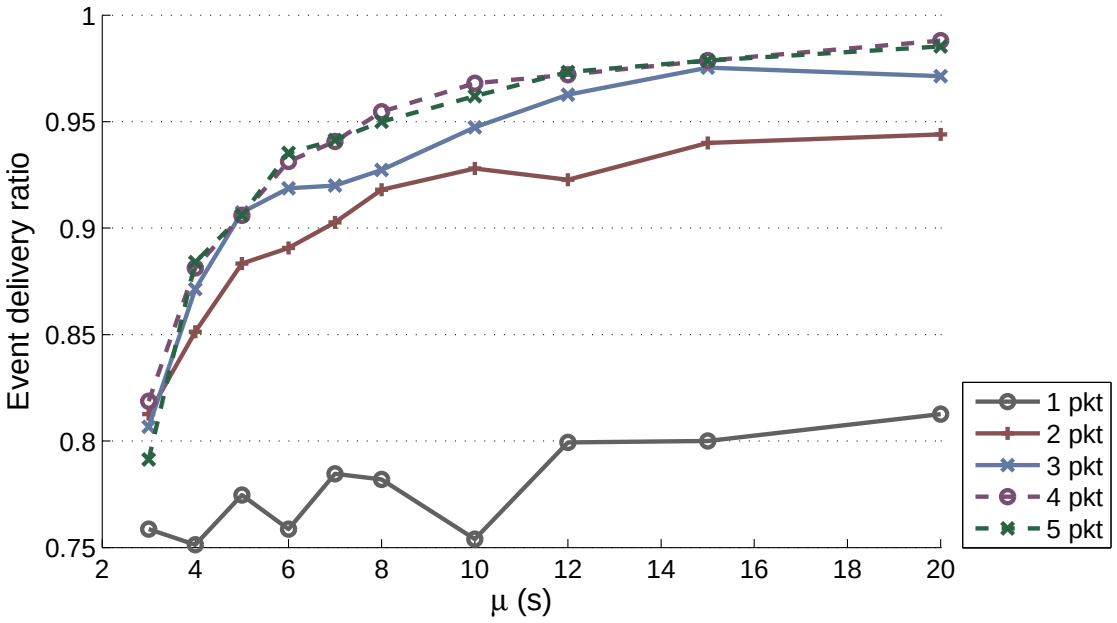


(a)

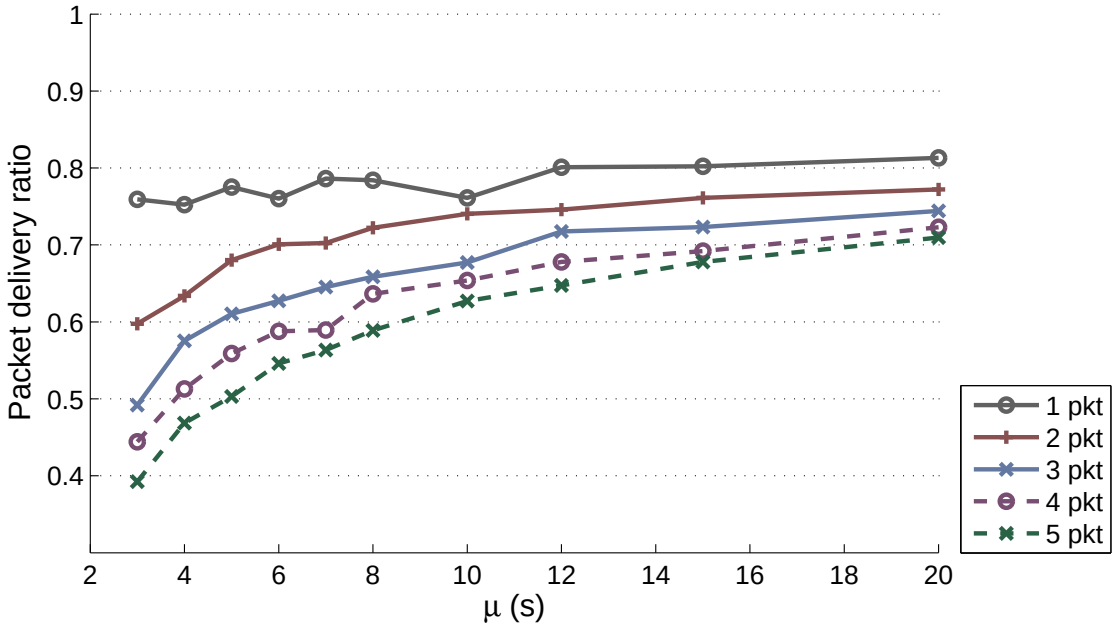


(b)

Figure 4.4: Simulation results showing EDR(a) and PDR(b) with different number of retransmissions per event, at $PDR_0 = 0.6$

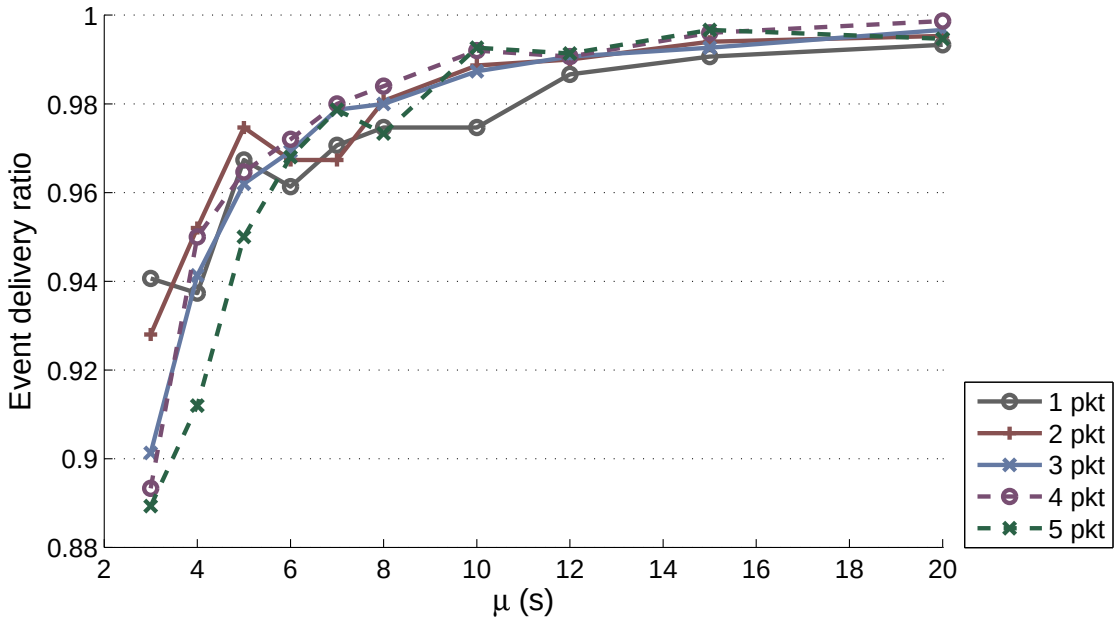


(a)

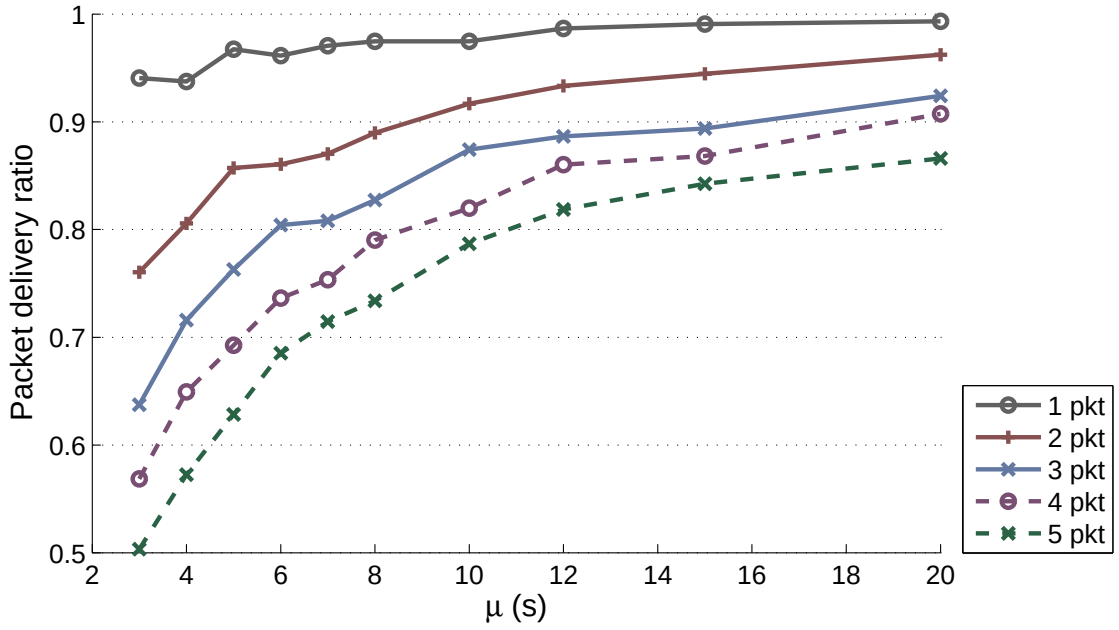


(b)

Figure 4.5: Simulation results showing EDR(a) and PDR(b) with different number of retransmissions per event, at $PDR_0 = 0.8$

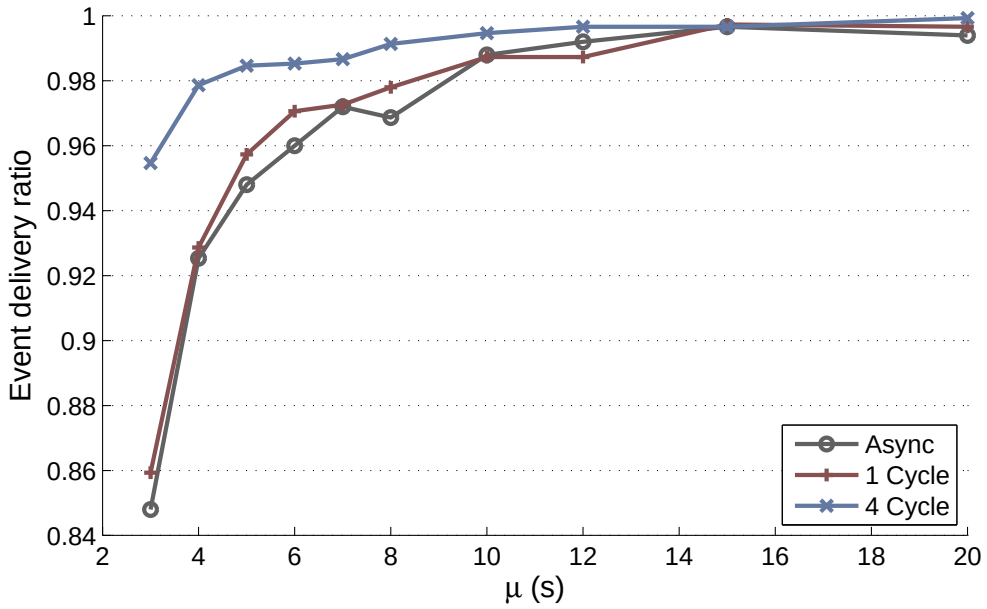


(a)

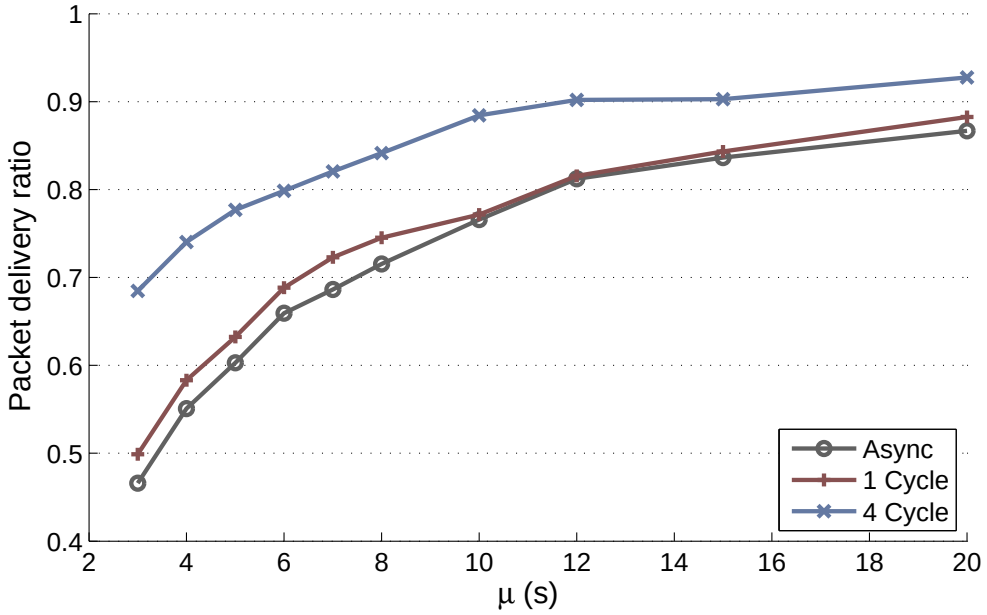


(b)

Figure 4.6: Simulation results showing EDR(a) and PDR(b) with different number of retransmissions per event, at $PDR_0 = 1$



(a)



(b)

Figure 4.7: Simulation results showing EDR(a) and PDR(b) with different timeslot alignment

ities. Another way is to reduce packet length to less than one AC cycle, which means also raising transmitter power in order to get the same PDR_0.

4.2.2 Two-event collision test and Poisson distribution events

We setup a small scale testbed so that we can test real hardwares with programmatically controllable events. We control the events through an extra microcontroller which is connected to a computer. The extra microcontroller, which is an mbed[1], is connected to each sensor with a wire. It toggles the wire to indicate events. The mbed is also connected to the computer with a USB emulated serial port. A perl script running on the computer can issue events to the sensors with the help of the mbed.

The sensors run a slightly modified version of firmware, which enables them to receive events from the digital input pin, instead of using the AC current step detection. Nevertheless, the AC current step detection algorithms are still running, in order to emulate the timing of event detection as accurate as possible. The RF transmission code is unmodified. The sensors are plugged into AC sockets and powered by AC, so that the zero-crossing function is also working in the same way as it would in real deployment.

We conduct a two-event collision test. Since all 5 transmissions of a single event can take up to 2.2s, any two events that happen within 2.2s on two different sensors are considered collided events. Note that two events that happen within 2.2 on a single sensor are not collided, because any pending retransmissions of the first event (flash event) are cancelled.

In this test, we generate collided events two at a time for 100 times. In each testing set, the interval between each pair of collided events is fixed. Fig.4.8 shows the resulting EDR and PDR. The plots show that EDR becomes perfect when the interval between collided events are greater than 0.2s. In real cases, it is unlikely that two events happen at exactly the same time. Even if some appliances are highly correlated in term of their on-off states, such as a DVD player and a TV set, their turn-on time are not strictly aligned. Even in the extreme case of two strictly synchronous events, the system can still delivery them at a high

[1] http://mbed.org/

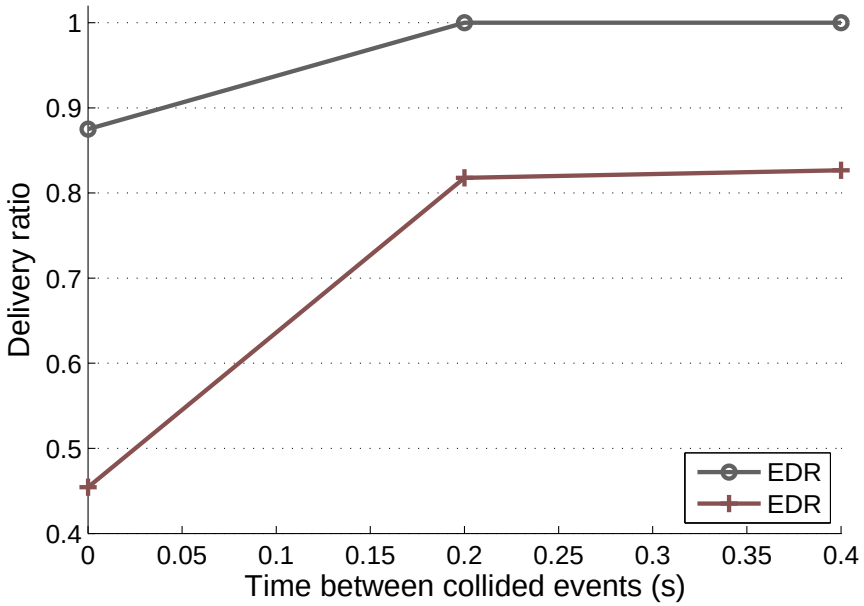


Figure 4.8: Two-event collision test result

probability of 88%.

Again, we use poisson distribution datasets on the testbed. The datasets are synthesized in the same way as described in the previous section, with $N = 6$, μ varying from 3 to 20. We limit the length of each dataset to 300 seconds long. Fig.4.9 shows the average EDR and PDR, as well as the EDR of each sensor separately. We can see that as the event density increases, some sensors are much worse than the others. This is because the transmission power differs among the sensors. The receiver has an automatic gain control (AGC) to tune the receiving signal strength. However, the AGC needs time to settle. When the transmission are too busy, the AGC can not accommodate all sensors fast enough. If the events are sparse, in the case of longer μ, all sensor nodes have near perfect EDR. Again, even with $\mu = 20$, the events are much more denser than a real setting. Appliances are very unlikely to be switched off and on within 20 seconds.

4.2.3 Evaluation in a real setting

In this section, we provide the statistics of the sensor network with a dataset collected in a real setting. We have 21 sensors deployed in our lab, numbered #1 through #22. Details

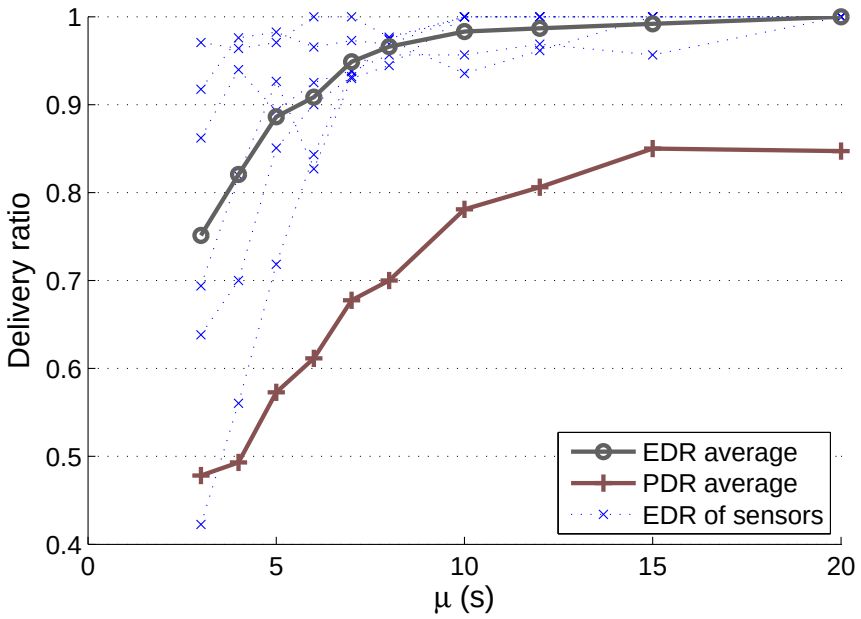


Figure 4.9: Poisson distribution events test

about the deployment are discussed in Chapter 5. #10 is not deployed. #16 is deployed, but there is no event during the time window of this dataset. #20 is later removed.

The dataset we used for the statistics contains data collected in 18 days. Note that in the real setting, lost packets and lost events are detected by purely analyzing the sequence number. This is accurate unless there are more than 16 successive packet loss, which is very rare. Fig.4.10 shows the EDR and PDR of each sensor. We can see that all sensors achieve almost perfect EDR, except for #1. Detailed data are shown in Table 4.1.

ID	RX Packets	TX packets	RX Events	TX Events
1	308	415	83	89
2	247	275	64	64
3	628	634	137	137
4	208	230	44	45
5	331	333	67	67
6	37	38	8	8
7	20364	21818	4534	4569
8	290	304	69	69
9	307	312	81	81
11	354	373	81	81
12	216	217	54	54
13	438	440	88	88
14	134	138	30	30
15	55	60	12	12
16	0	0	0	0
17	255	265	54	54
18	1125	1135	227	227
19	146	150	30	30
20	452	483	97	97
21	586	596	120	120
22	270	290	58	58
Total	26751	28506	5938	5980

Table 4.1: Sensor network delivery statistics in a real setting

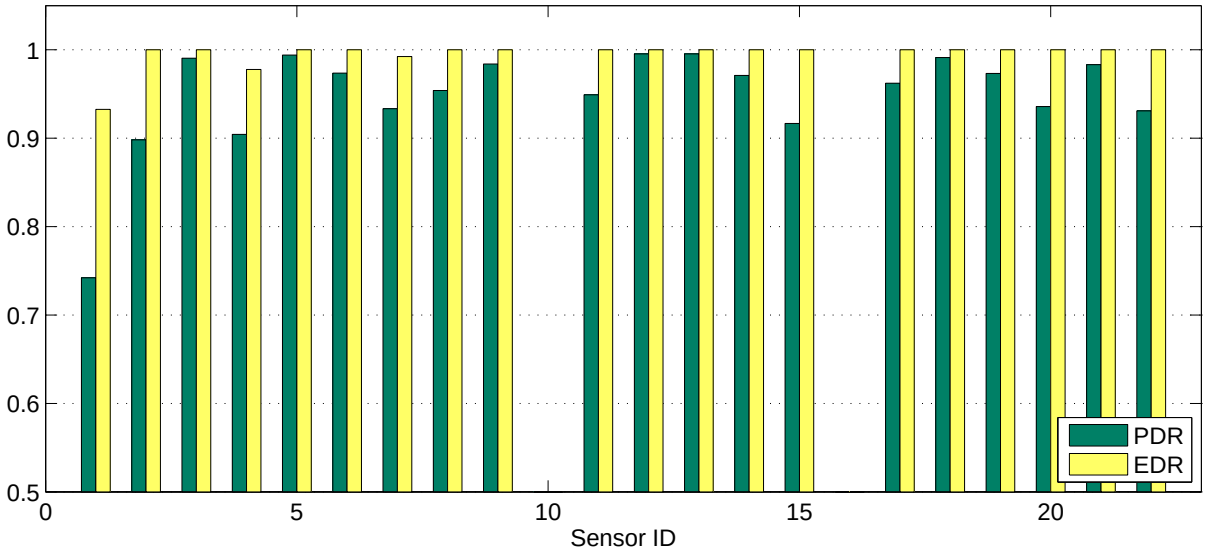


Figure 4.10: EDR and PDR in a real setting

CHAPTER 5

Energy disaggregation assisted by appliance state sensing

In this chapter, we propose an energy disaggregation algorithm. It uses our appliance state sensing system to determine states of appliances, and uses a central energy meter to provide power numbers in Watts.

5.1 Energy disaggregation algorithm

The sensors can tell when and which appliance power state has changed. However, it has no capability to provide power consumption numbers. With the knowledge of the aggregated power consumption, we are able to associate events with the changes in total consumption, so that we are able to infer the consumption of that particular appliance, assuming its consumption is stable when it is powered on.

There are some challenges to do so. Firstly, there is usually slight variation in the power that an appliance takes over time. Some appliances have stable power traces, such as lights, others may have large variations in their power traces. We see 5W-10W variation in the power traces of laptops when simply browsing web pages. When there are a number of appliances, the variation adds up at the central meter, so that we usually see a very noisy power trace. Most of the appliances in our experiments consume tens of watts, and the variation is sometimes at the same order of magnitude. Therefore, even knowing there is an event happening, it is hard to tell the accurate power step change. We can not use a lowpass filter, because the high frequency noises are in the same frequency range as the step changes.

Secondly, power changes observed at the central meter may not be synchronous with the events. The time difference can be as long as several seconds. In the case when multiple events happens close in time, the power trace goes as a slope, instead of having sharp steps.

To overcome these problems, we run piecewise constant (PWC) denoising algorithm on the central power trace first. It estimates the original signal with a piecewise constant signal which only have a small number of step changes, removing the noise. The algorithm is discussed in the next section in detail. After PWC denoising, we match the step changes with the events, according to their timestamp. If an event is clear without overlapping with other events, we will have its associated power number. However, if multiple events happens close to each other, i.e they are overlapping, then further disaggregation is needed to get the power number for each individual event. The block diagram of the energy disaggregation algorithm is shown in Fig.5.1.

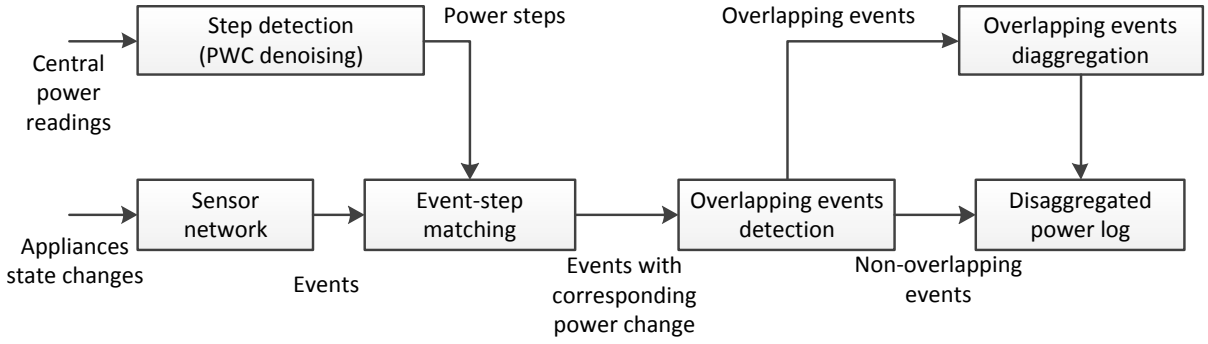


Figure 5.1: Block diagram of an energy disaggregation algorithm

5.1.1 PWC denoising and step detection with the central power trace

In our experiments, we found that it is important to have a robust PWC denoising algorithm, especially when the number of appliances scales up and the interested appliances consume only tens or a hundred watts. Step changes in the aggregated power are swamped in noise. A low pass filter is not suitable in this case because the noises have the same frequency as the step changes that we want to preserve. The problem is not particularly significant in previous research, because previously, people mostly focus on few appliances with very large

power consumption, such as dish washers, heaters or ovens.

Several methods have been used in previous work to remove noise or detect step-change in the power trace. In [Har92], the author developed an edge detection algorithm. It looks for stable periods where the variation is less than a threshold (15W or VAR). The method works only when the step changes are much larger than the variation, which is not true in our case. In [NL96], the authors used a median filter to remove spikes in the raw signal. Total variation denoising is used in [KJ12]. The method can effectively remove noises in the signal and preserve steps. However, its purpose is not step detection. Hence, its output is not guaranteed to be a piecewise constant signal with sparse steps.

A method called jump penalization is introduced in [LJ11]. The original method is an offline one. The algorithm begins with a constant signal as the estimation of the input signal. Usually the mean or median value is used for the initial estimation. In each iteration, a greedy search tries to insert the best new step-change in the current estimation. The algorithm ends when the improvement of estimation accuracy by inserting new step-changes is too small. The algorithm guarantees piecewise constant output, and the step-changes are explicitly found in each iteration.

In this work, we adopt the algorithm to an online variant, which can process the data in real time. Instead of inserting step-changes iteratively, we insert step-changes one by one as new samples come in. When new samples come in, we consider adding at most one new step in the period of time from last step-change to the current sample. The flow chart of our step-detection algorithm is shown in Fig.5.2. The error function is defined as following:

$$E(x, y) = \frac{1}{2} \sum_n (x_n - y_n)^2$$

The parameter γ acts as a threshold in the algorithm, or 'penalty' to insert a new step-change. With a lower γ, the output contains more steps. Instead of a constant γ, We use a time-variant γ_t based on our knowledge about events. When there are events detected by the sensor network, we use the lower γ_l so that it is more likely to detect small step changes. When no events are detected, we use the higher γ_h to reduce false alarms. In practice, when the power signal is in kW, we choose $\gamma_l = 0.0001$ and $\gamma_h = 0.0025$.

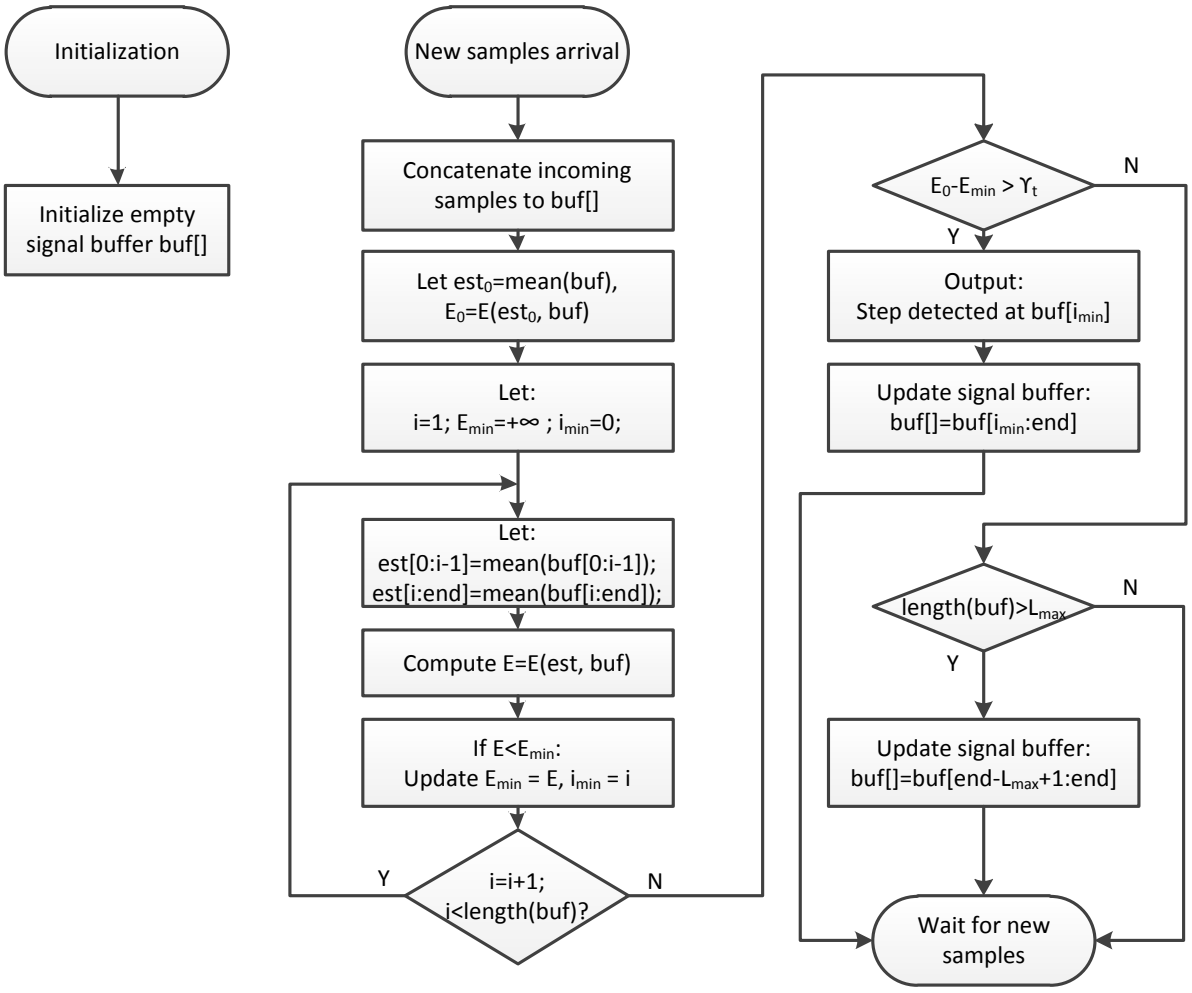


Figure 5.2: Modified jump penalization algorithm for step detection

We also limit the length of signal buffer to L_{max}, so that the greedy search domain is limited to the last L_{max} samples. Otherwise, the buffer can grow indefinitely when there is no step-change for a long time.

5.1.2 Overlapping events disaggregation

When multiple events happen close enough, only the aggregated effect will shown on the power readings at the central meter. In this case, we use the historical data to estimate the power changes. Specifically, we use the median value of all power changes that we observed before on non-overlapping events of a particular appliance. This works under the assumption that all appliances have binary on-off states.

5.2 Experimental results

We deployed an experimental setup in our lab. It is a mixed office and lab setting. All the interested appliances are arranged on 4 branch circuits, in order to minimize the interference from other appliances. The lab has 20 branch circuits in total, which is monitored by a Veris E30A panel energy meter[1]. We collect data from the meter at an interval of 2 seconds, and use the sum of power readings on the 4 branch circuits as the aggregated power reading.

We have 20 sensor nodes deployed, including 9 LCD monitors, 8 laptop computers, a solder station, a workbench light and a water dispenser. Most appliances are binary-state, except the water dispenser, which has both cooler and heater, and they work independently like two separate appliances.

There are 11 appliances that we have instrumented with *Watts up? PRO* meters. These meters capture the power consumption of individual appliances and log the data on a server through Gumstix nodes. These data serve as the ground truth. They are not used in any way in the energy disaggregation and inferencing process.

The dataset used in this section is a 7-day long dataset collected from June 1st-7th, 2012.

5.2.1 Step detection results

Fig.5.3 shows some signal snippets with PWC denoising and step detection results. The big steps showing in the figure is the water dispenser heater. Except that, other power steps are mostly tens of Watts. And there is a oscillating noise in the power signal due to some laptop computers. The PWC denoising algorithm can detect steps in the same order of magnitude as the noise.

5.2.2 Appliance state tracking accuracy

We use the state change events that the sensors detected to reconstruct the state of each individual appliances over time. By comparing appliance states collected by our sensors with

[1] http://www.veris.com/

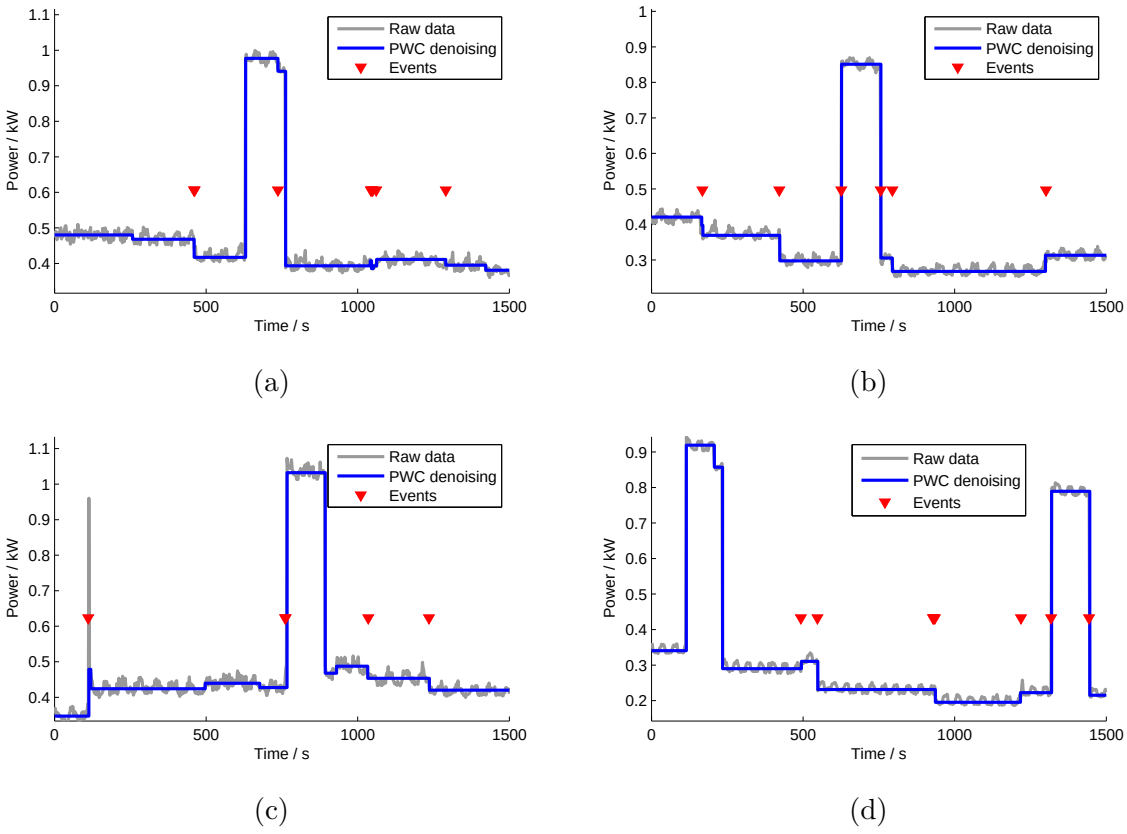


Figure 5.3: Signal snippets showing PWC denoising results

ID	On (s)	Reported on (s)	On&reported on (s)	Precision (%)	Recall (%)
1	24659	24586	24577	99.965	99.667
2	24657	24586	24571	99.937	99.651
3	19125	19131	19117	99.925	99.958
5	136328	135248	135214	99.975	99.183
8	19477	19477	19461	99.920	99.918
12	128745	128198	128173	99.981	99.556
13	106761	106200	106151	99.954	99.429
14	764	763.7	762	99.773	99.738
17	75308	75471	75296	99.768	99.984
18	218431	217935	217771	99.925	99.698
21	34720	34708	34642	99.810	99.775

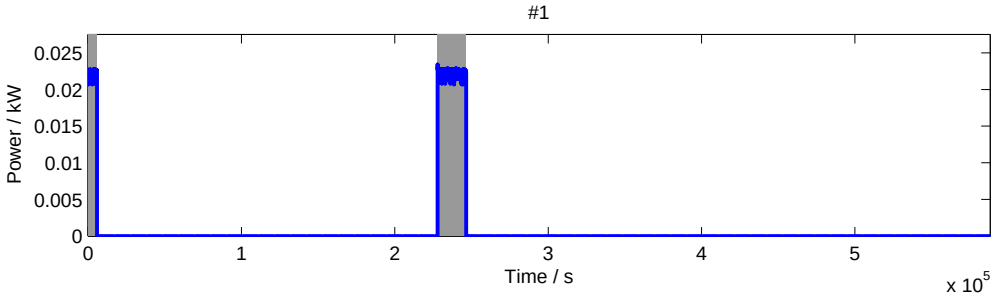
Table 5.1: Appliance state detection accuracy

the ground truth, we show the accuracy of our system. We measure accuracy by precision and recall, defined as following:

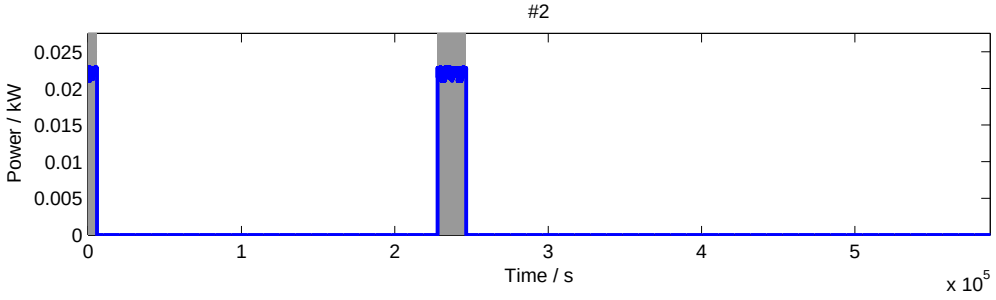
$$\text{Precision} = \frac{tp}{tp + fp} = \frac{\text{Time appliance is on and reported on}}{\text{Time appliance is reported on}}$$

$$\text{Recall} = \frac{tp}{tp + fn} = \frac{\text{Time appliance is on and reported on}}{\text{Time appliance is on}}$$

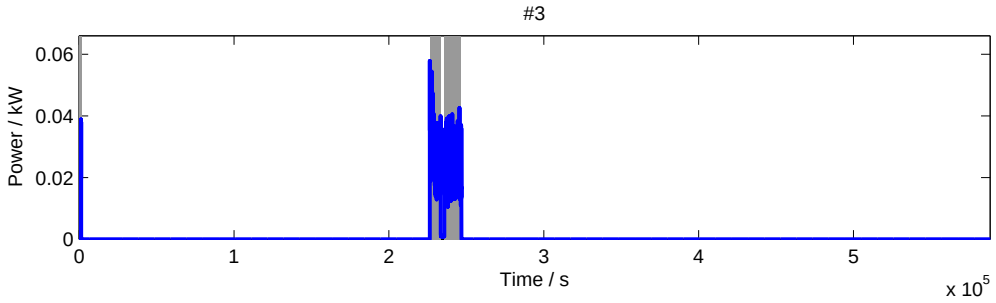
Table 5.1 shows the statistics of state tracking with ground truth. Here, an appliance is considered in on state when its power consumption is greater than 5W. The table shows power-on time, reported-on time, reported-on-and-indeed-on time, along with precision and recall. Fig.5.4-5.6 shows the power consumption over time of the appliances. The solid traces are power ground truth of each individual appliance. Shaded area shows the period when our sensors report power-on. We can tell from the data that our system can provide very reliable and accurate appliance state data.



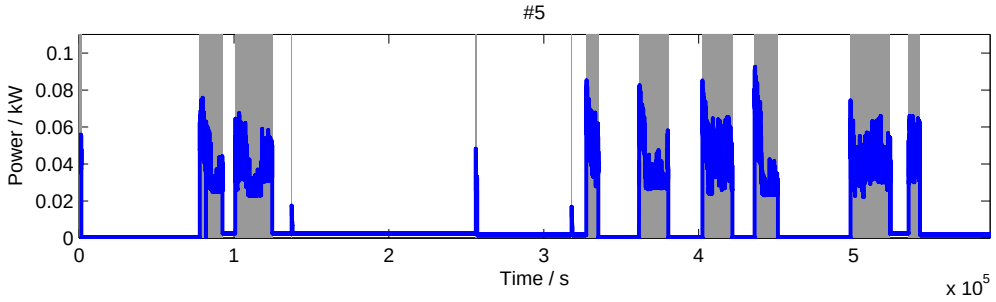
(a) #1 Monitor



(b) #2 Monitor

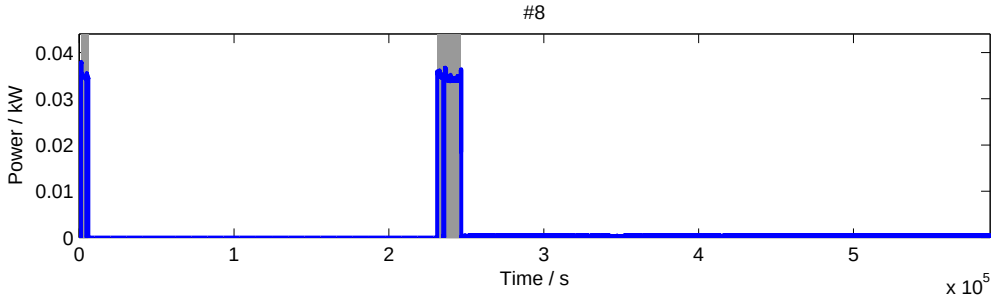


(c) #3 Laptop

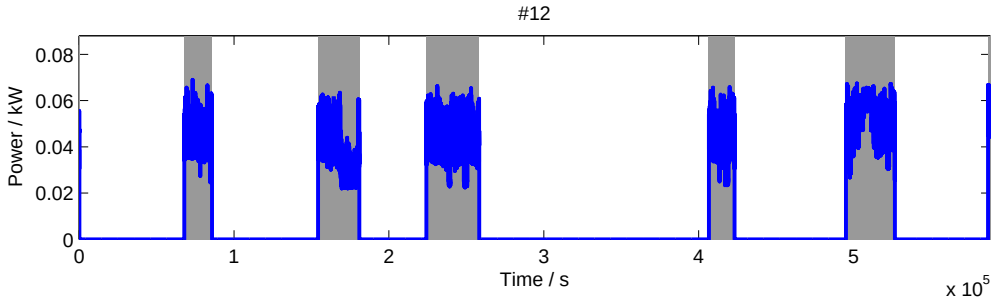


(d) #5 Laptop

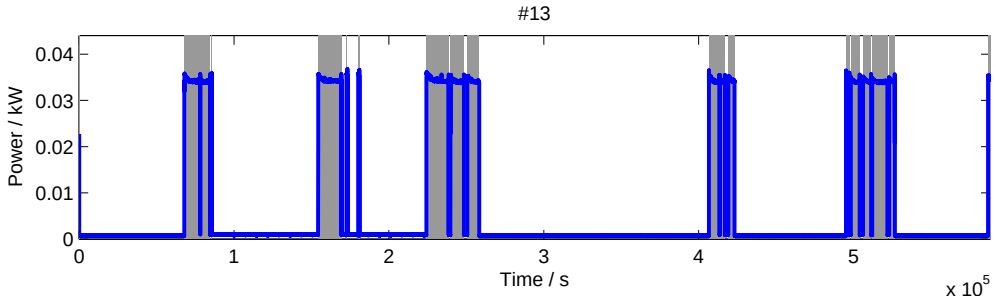
Figure 5.4: Appliance state reported vs. truth (#1,2,3,5)



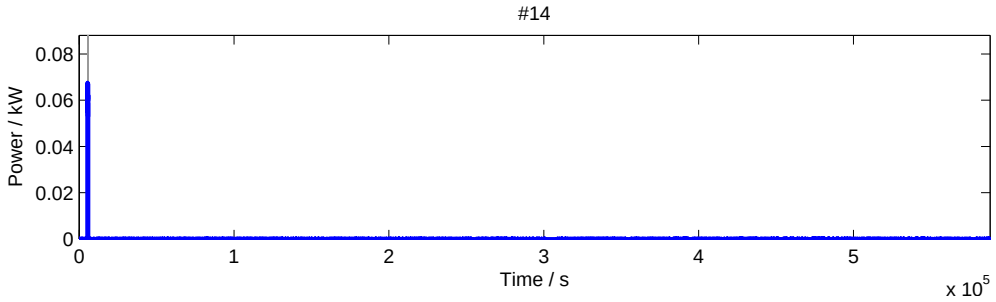
(a) #8 Monitor



(b) #12 Laptop

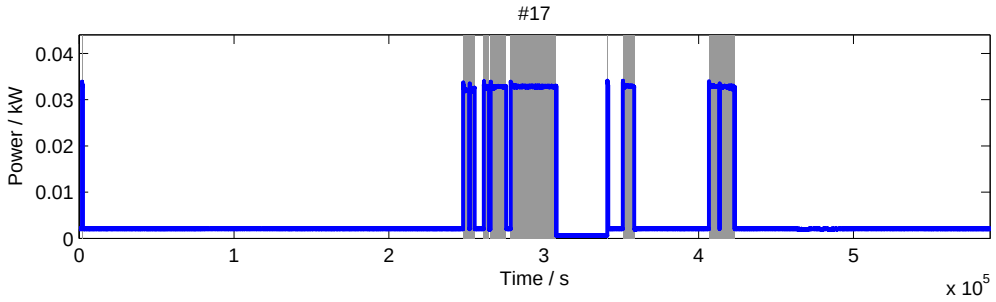


(c) #13 Monitor

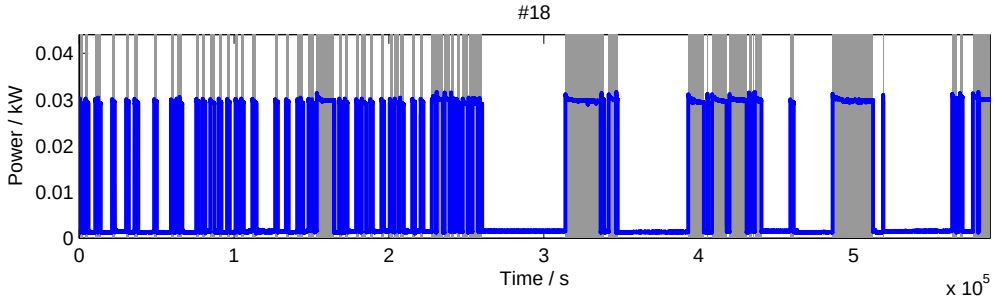


(d) #14 Laptop

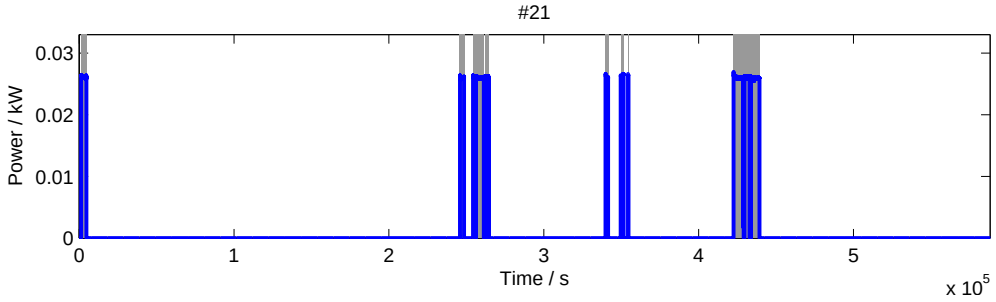
Figure 5.5: Appliance state reported vs. truth (#8,12,13,14)



(a) #17 Monitor



(b) #18 Monitor



(c) #21 Monitor

Figure 5.6: Appliance state reported vs. truth (#17,18,21)

5.2.3 Energy consumption of individual appliances

Once we have reliable appliance power states and the step changes in the aggregated power, we can estimate the power traces of individual appliances, according to the algorithm discussed previously.

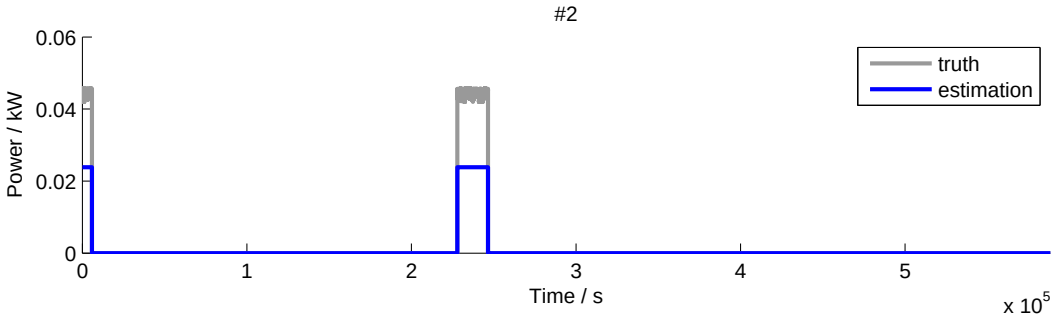
Table 5.2 shows the energy consumption estimation of the appliances. For those appliances instrumented with ground truth power readings, we show both the true energy consumption over the 7-day period of the dataset, and the estimated energy consumption. For those appliances without ground truth power readings, only the estimation is shown. In the experiments, we found that appliance #1 and #2, which are two monitors, are working in complete synchrony. Theoretically, there is no way to distinguish the power consumption of them other than metering them separately. Hence, we regard them as a single combined appliance.

The estimation error differs very much among different appliances. For most appliances, the error is less than 30%. However, for monitors #1+2, laptop #3 and laptop #5, the error is large. Generally, LCD monitors, which have stable on-state power, have more accurate results, while energy estimation for laptops are less accurate. This is because the power of an on-state laptop can vary as much as one third over time, due to different load, battery charging status, etc. Other causes of inaccuracy include lack of state changes. Some appliances are kept on or off for a long time, and there are only few state changes. In this case, the errors in the power step detection will keep accumulating. Monitors #1+2 and laptop #3 all fall in this case. They have just 3-5 state changes during the period of the dataset.

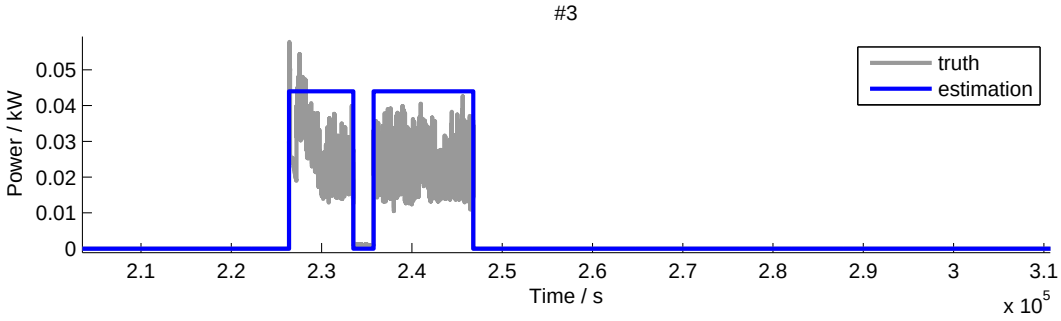
Fig.5.7-5.9 displays some snippets of estimated power vs ground truth for several appliances.

ID	Power-on time (h)	Avg. power (W)	Energy (Wh)	Est. energy (Wh)	Error (%)
1+2	6.85	47.9	327.9	163.0	-50.3
3	5.31	29.9	158.9	233.8	+47.1
5	37.87	38.6	1461.4	2279.1	+56.0
8	5.41	44.9	242.9	199.0	-18.1
12	35.76	36.9	1319.1	1639.1	+24.3
13	29.66	36.2	1072.1	1241.7	+15.8
14	0.21	63.1	13.4	9.60	-28.3
17	20.92	45.9	961.0	667.8	-30.8
18	60.68	31.1	1885.2	1718.3	-8.9
21	9.64	26.9	259.8	290.2	+11.7
4	64730	-	-	537.7	-
7	259984	-	-	17216	-
9	521012	-	-	5697.8	-
11	61032	-	-	829.0	-
15	3852	-	-	47.9	-
19	55160	-	-	324.7	-
22	59216	-	-	640.6	-

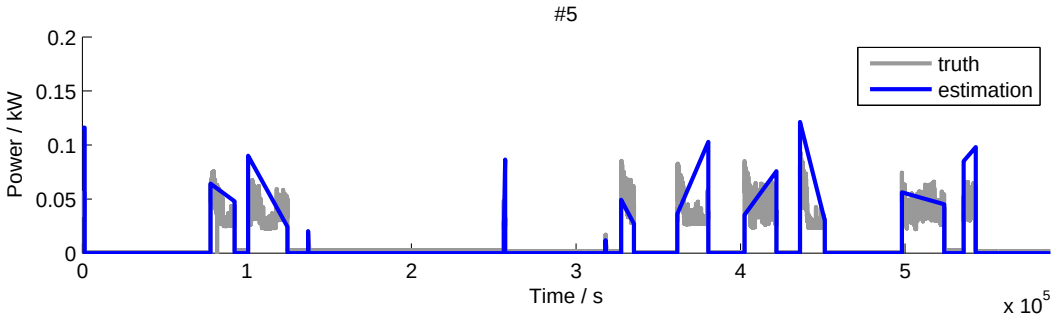
Table 5.2: Energy consumption estimation of individual appliances



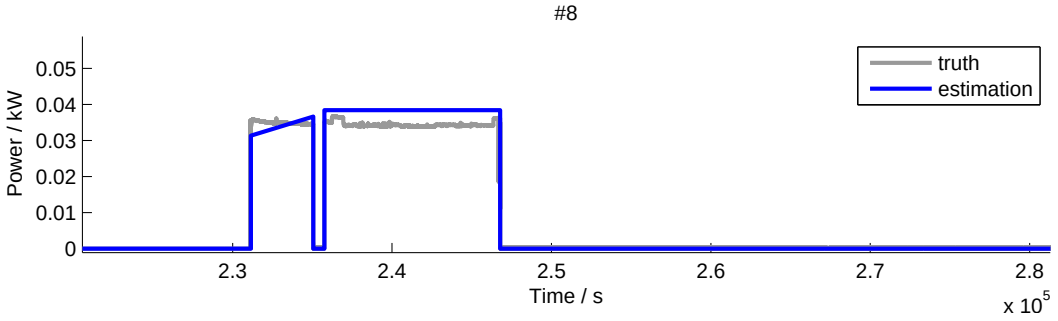
(a) #1+2 Monitors



(b) #3 Laptop

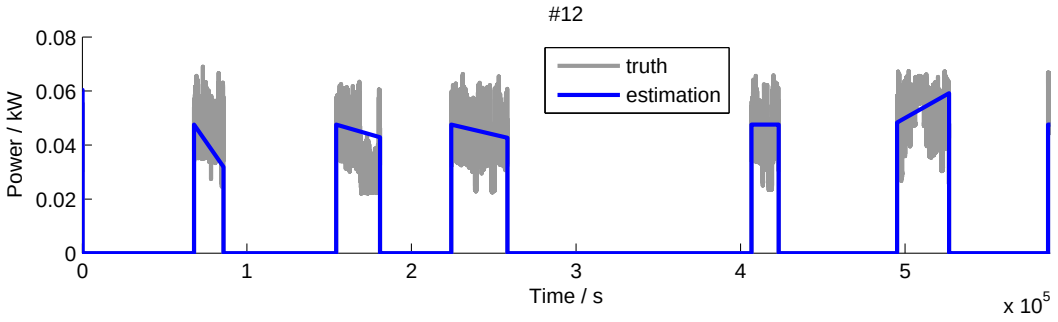


(c) #5 Laptop

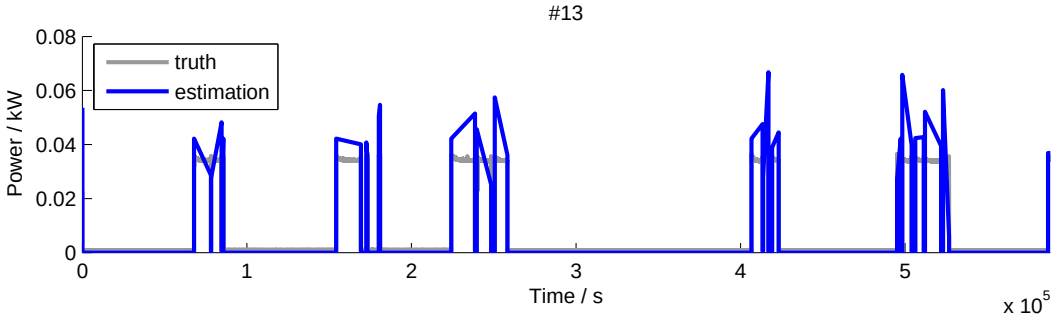


(d) #8 Monitor

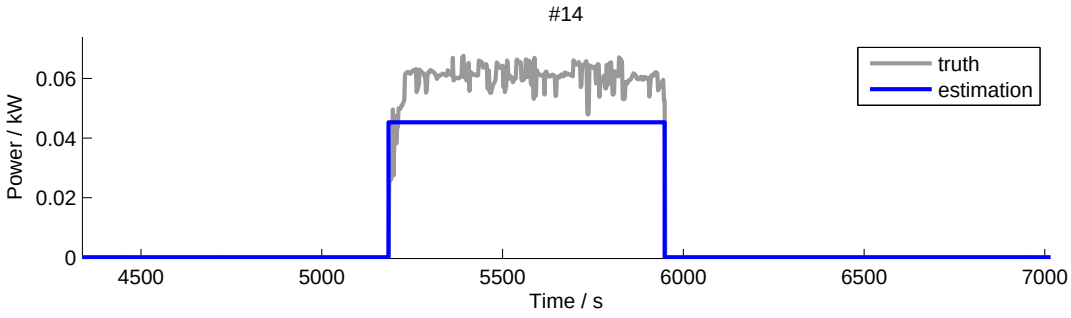
Figure 5.7: Appliance power consumption estimation vs. truth (#1+2,3,5,8)



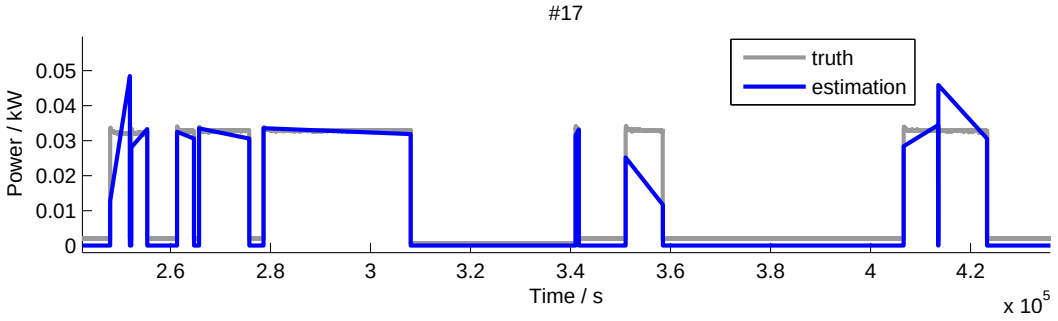
(a) #12 Laptop



(b) #13 Monitor

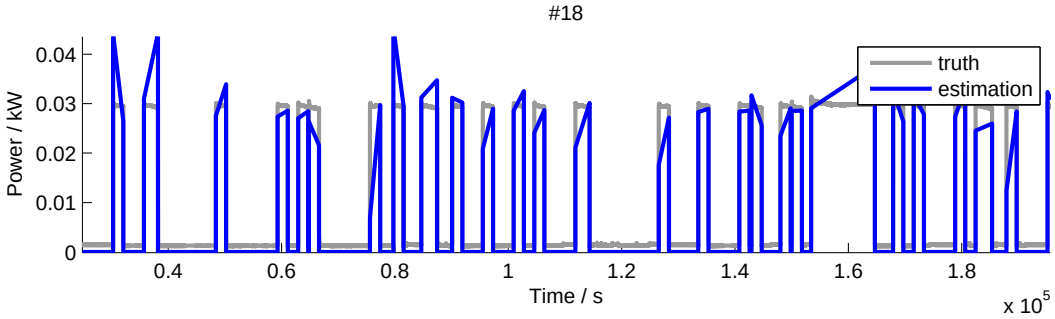


(c) #14 Laptop

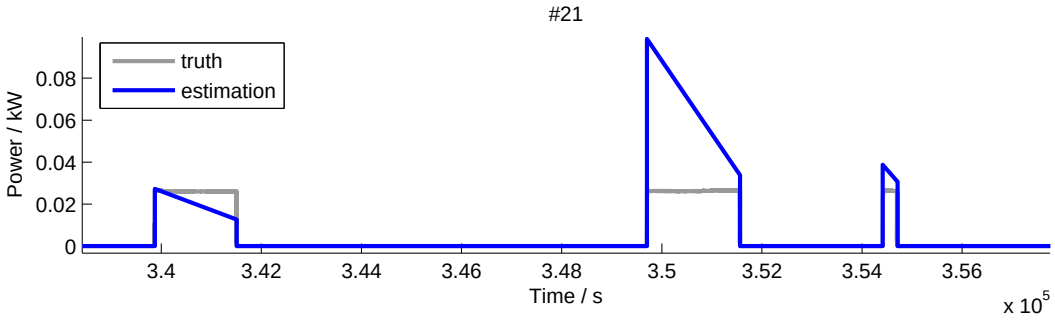


(d) #17 Monitor

Figure 5.8: Appliance power consumption estimation vs. truth (#12,13,14,17)



(a) #18 Monitor



(b) #21 Monitor

Figure 5.9: Appliance power consumption estimation vs. truth (#18,21)

CHAPTER 6

Conclusion

In this paper, we propose a new sensing system that can detect and report power state changes of individual appliances. In our system, the sensors are only responsible for on-off state change detection, which makes their hardware requirements minimal. The sensors have a extremely simple microcontroller onboard, along with an uncalibrated sensing frontend, and a transmit-only radio transmitter. We show that the simplicity in requirements allows the sensors to be made at an overall price as low as 4.3 dollars each. This makes it affordable for large scale deployment, where sensors are deployed at each appliance, or even embedded in power outlets or plugs.

Despite low-cost underpowered, the sensor nodes are sufficient to detect on-off state of appliances accurately. Combined with the robust radio scheme we developed, the system can keep track of the power state of tens of appliances with a single base station. Our evaluation both in simulation and in real settings shows that the accuracy (precision and recall) of power state tracking is more than 99.5%. The system solves two of the main issues in energy disaggregation: event detection and appliance identification.

Combining the power state data from our sensing system with a central power meter, we are able to associate power state changes with real Watts numbers. Then, we can keep track of the power consumption of individual appliances, and estimate their disaggregated energy consumption. The accuracy is more than 70% for most appliances in our challenging 20-appliance disaggregation setup, where most appliances have low and similar power consumption.

Since knowing the state of appliances is the most challenging problem of a number of energy disaggregation methods, our sensing system can surely assist these methods to achieve

higher accuracy or scalability.

APPENDIX A

315MHz radio coding scheme

We use a very simple ASK transmitter and receiver. The transmitter module only provides a digital input pin which controls whether it transmits (1) or not (0). And the receiver has one digital output pin.

There are two issues about the receiver we need to take care of when we design the encoding and decoding software. 1) When no transmitter is transmitting, or when a transmitter is on for several milliseconds, either way the receiver will output a random stream of bits instead of steady 0 or 1. This is due to the auto gain control (AGC) on the receiver trying to adapt to the current background signal strength. Moreover, when a transmitter begins transmitting, the AGC takes time to adapt, hence the output of the beginning period is more noisy than afterward. 2) When the signal strength is not good enough, the output stream will contain lots of spikes. Thus, we can not reliably decode the message by looking at transition edges or by sampling the signal only once per bit.

Each packet is 16-bit long in our design. The packet begins with 7 milliseconds of 1, which serves to stabilize the receiver AGC. Then follows a preamble, for synchronization purpose. The preamble consists of two positive edges that are precisely 4 bits apart in time. The receiver uses these edges to determine the length of each bit. Following that, 16 data bits are transmitted using Manchester code, where 1 is represented with 1-0 transition, and 0 is represented with 0-1 transition. Note that the second edge of the preamble is actually the beginning of a dummy bit 1 transmitted with Manchester code. Finally, a parity bit is transmitted, also in Manchester code.

The length of each bit is 2.5 milliseconds in practice. Each packet is approximately 62 milliseconds. The length is less than 4 AC cycles, which is 66.7 milliseconds. Hence, we

choose 4 AC cycles as a transmission timeslot.

References

[BV04a] Michael Baranski and Jürgen Voss. "Detecting Patterns of Appliances from Total Load Data Using a Dynamic Programming Approach." In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pp. 327–330. IEEE, 2004.

[BV04b] Michael Baranski and Jürgen Voss. "Genetic algorithm for pattern detection in NIALM systems." In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, number 2, pp. 3462–3468. IEEE, 2004.

[CLS06] Robert Cox, Steven B. Leeb, Steven R. Shaw, and Leslie K. Norford. "Transient Event Detection for Nonintrusive Load Monitoring and Demand-Side Management Using Voltage Distortion." In *Twenty-First Annual IEEE Applied Power Electronics Conference and Exposition, 2006. APEC '06.*, pp. 1751–1757. Ieee, 2006.

[NILM.]

[Dar06] Sarah Darby. "The effectiveness of Feedback on energy Consumption." *A Review for DEFRA of the Literature on Metering, Billing and direct Displays, April*, 2006.

[Fis08] Corinna Fischer. "Feedback on household electricity consumption: a tool for saving energy?" *Energy Efficiency*, **1**(1):79–104, May 2008.

[Har92] G.W. Hart. "Nonintrusive appliance load monitoring." *Proceedings of the IEEE*, **80**(12):1870–1891, 1992.

[Auto training. 15W/Var resolution. 1s sampling rate. Both real and reactive domain. Auto clustering algorithm. Too old to implement.]

[JDD09] Xiaofan Jiang, Stephen Dawson-Haggerty, Prabal Dutta, and David Culler. "Design and implementation of a high-fidelity ac metering network." In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pp. 253–264, 2009.

[JS10] Deokwoo Jung and Andreas Savvides. "Estimating building consumption breakdowns using ON/OFF state sensing and incremental sub-meter deployment." In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*, p. 225, New York, New York, USA, 2010. ACM Press.

[KJ12] J Zico Kolter and Tommi Jaakkola. "Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation." In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume XX, 2012.

[HMM.]

[KSC09] Younghun Kim, Thomas Schmid, Zainul M Charbiwala, and Mani B. Srivastava. "ViridiScope." In *Proceedings of the 11th international conference on Ubiquitous computing - Ubicomp '09*, p. 245, New York, New York, USA, 2009. ACM Press.

[LCS03] Christopher Laughman, Robert Cox, Steven Shaw, Steven Leeb, Les Norford, and Peter Armstrong. "Power signature analysis." *IEEE Power and Energy Magazine*, **1**(2):56–63, March 2003.

[3rd, 5rd or 7rd harmonic commerial building.]

[LJ11] Max a Little and Nick S Jones. "Generalized methods and solvers for noise removal from piecewise constant signals. II. New methods." *Proceedings. Mathematical, physical, and engineering sciences / the Royal Society*, **467**(2135):3115–3140, November 2011.

[LSK95] S.B. Leeb, S.R. Shaw, and J.L. Kirtley. "Transient event detection in spectral envelope estimates for nonintrusive load monitoring." *IEEE Transactions on Power Delivery*, **10**(3):1200–1210, July 1995.

[Transient power.]

[MHH11] Alan Marchiori, Douglas Hakkarinen, Qi Han, and Lieko Earle. "Circuit-Level Load Monitoring for Household Energy Management." *IEEE Pervasive Computing*, **10**(1):40–48, January 2011.

[need training for all devices, probability model. not applicable.]

[NL96] Leslie K. Norford and Steven B. Leeb. "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms." *Energy and Buildings*, **24**(1):51–64, January 1996.

[PHM06] Danny Parker, David Hoak, Alan Meier, and Richard Brown. "How Much Energy Are We Using? Potential of Residential Energy Demand Feedback Devices." In *ACEEE 2006 Summer Study on Energy Efficiency in Buildings*, 2006.

[Rob75] Lawrence G Roberts. "ALOHA packet system with and without slots and capture." *ACM SIGCOMM Computer Communication Review*, **5**(2):28–42, April 1975.

[US11] U.S. Energy Information Administration. "Annual Energy Review 2010." Technical report, 2011.