Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

ENERGY CONSUMPTION OF DATA DRIVEN TRAFFIC SIMULATIONS

SaBra Neal
Richard Fujimoto

Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Michael Hunter

Civil and Environmental Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

ABSTRACT

Dynamic Data-Driven Application Systems (DDDAS) implemented on mobile devices must conserve energy to maximize battery life. For example, applications for online traffic prediction require use of real-time data streams that drive distributed simulations. These systems involve embedding computations in mobile computing platforms that establish the state of the system being monitored and collectively predict future system states. Understanding where energy consumption takes place in such systems is vital to optimize its use. Results of an empirical investigation are described that measure energy consumption of aspects such as data streaming, data aggregation, and traffic simulation computations using different modeling approaches to assess their contribution to overall energy consumption.

1 INTRODUCTIONS

Energy consumption is an on-going concern in mobile and embedded computing systems powered by the device's battery. With the growing use of real-time data for traffic prediction applications one must understand tradeoffs between energy consumption for communications and computations under certain performance and accuracy constraints in order to ensure effective operation. For example, question might concern the approach used to model the system and the amount and frequency with which data should be collected to drive the simulation computations. This information is necessary to develop power and energy aware techniques to optimize energy use.

Dynamic Data Driven Application Systems (DDDAS) allow simulations to incorporate real-time or online data in order to drive the simulation system to produce predictions that can be used to aid measurements or optimize system operation (Darema 2004). DDDAS applications may be embedded within the physical system being monitored or optimized in order to utilize real-time data near the source of the data. For example, embedded traffic simulations may be part of a sensor network where real-time traffic data is used as input to drive transportation simulations. Monitoring ecological development, forest fires, and tracking multiple targets in an ad hoc sensor network are examples where a DDDAS system might be embedded within the physical system (Rodríguez et. al. 2009; Schizas and Maroulas 2015). In situations where battery-powered mobile devices are used as the DDDAS platform energy consumption by DDDAS computations and communications is an important issue.

Ad-hoc distributed simulation systems have been proposed for applications such as data-driven distributed traffic network simulations (Fujimoto et. al. 2007). These distributed simulations may be implemented in sensor networks where sensor data is communicated to distributed simulation processes within close physical proximity. In a transportation application each simulation is responsible for using the sensor data to make future state predictions about a portion of the traffic network and exchange current and predicted state information with other simulations. These simulations collectively predict the future state of the traffic network as a whole. Individual simulations may be in close proximity of the sensors, resulting in the distributed simulation being embedded within mobile devices. Alternatively, there one may execute a simulation in a smart phone to predict future delays. Utilizing distributed simulations in this way alleviates the cost of deploying and maintaining a centralized system. Our interest lies within the class of DDDAS that use distributed traffic simulations embedded in mobile devices.

This paper examines the energy consumed by data-driven simulations in predicting future states of a transportation network. The system utilizes sensor data specifying traffic flow on various road segments as input and makes future state predictions of an arterial traffic network. The future state predictions may then be distributed to other simulations, e.g., using the ad-hoc distributed simulation approach in order to enable state predictions of the entire network. A model characterizing energy utilized by the system is proposed that separates energy consumption in such a system for simulation computations and communications.

The organization of the paper is as follows: First we provide a brief overview of existing work in ad hoc distributed simulation systems, energy profilers, data driven traffic simulations, and energy consumption in cloud computing. In section 3 we present an architecture for an ad hoc distributed simulation system and describes a model that captures the different energy consuming components. Section 4 details the computational elements of the ad hoc distributed simulation, describes the embedded traffic simulations used here, and presents results of experiments measuring energy consumption for the simulations. Section 6 details the communication elements of the ad hoc distributed simulation and presents the results of energy consumption experiments.

2 RELATED WORK

Ad-hoc distributed simulations were proposed in (Fujimoto et. al. 2007) and their application to queuing networks and transportation systems are described in (Huang et. al., 2010; Henclewood et. al., 2010). There has not been work examining the energy consumption of ad hoc distributed simulations to date. Sudusinghe created a model that is intended for Data driven Embedded Signal Processing Systems where energy constraints are taken into account based on different application modes (Sudusinghe et. al. 2014). Recent work has been conducted in the area of power consumption of data distribution management services defined in High Level Architecture that is utilized by distributed systems (Neal et. al. 2014). Work evaluating the power consumption of disseminating state information in a distributed virtual environment that focuses on trade-offs between state consistency and power consumption has been conducted in the area of distributed computing (Shi et. al. 2003). Studies focusing particularly on energy of distributed systems include (Fujimoto & Biswas 2015).

Energy profilers are often utilized in order to measure energy consumption of mobile systems. Trepn is an example of a software tool that was developed by Qualcomm to measure power of Android systems. Recent work on profiling distributed simulations has been conducted (Biswas & Fujimoto 2016). Power Tutor is a software application that was developed to aid the design and selection of power efficient software for embedded systems (Zhang et. al 2010). The application informs users of power consumption to aid application design and use. WattsOn, like PowerTutor is a software application that allows developers to estimate the energy consumption of applications during development (Mittal 2012). Utlizing techniques such as the energy foot print of mobile hardware systems, fine grained system trace calling, and self constructive approaches where mobile systems automatically generate their energy model without any external assistance through a smart battery interface have been conducted to gain an understanding of how energy is disipatied in mobile devices (Dongarra et. al. 2012; Pathak et. al. 2011).

Data driven traffic simulators utlize acquiredacquire traffic data generally from a sensor network located within the environment of the area of study. There has been ample development in the area of data driven traffic simulations, Zhang surveys the different type of data driven intelligents systems for traffic networks that are currently implemented and used. The findings from Zhang's work include work in the area of vision intelligent systems, where vision based systems are used to detect, track, and recognize traffic related objects and vehicle detection. Multisource driven intelligents systems utilize loop detectors, lasar radar, and GPS data to drive there systems. Learning driven intellegent systems utlize online learning, data fusion, rule extraction, ADP (Adaptive Dynamic Programming) based learning control, and ITS (Intelligent Transportation System) oriented learning to drive their transportation systems (Zhang et. al. 2001).

The emerging use of cloud computing and mobile systems have made energy consumption in such systems an involving area of research. Cloud computing allows systems to offload work that would otherwise be performed locally on the device. Mobile systems often have a limited amount of resources such as data storage available to them. Utlizing the cloud instead allows data to be leveraged onto the cloud rather than consuming the limited storage on the mobile device. However, transfering data to the cloud requires energy, another scare resource in mobile devices. Researchers are exploring how to reduce and make best use of energy in such systems. Miettinen and Nurminen explore the critical factors that affect energy consumption of mobile clients in cloud computing. They analyzed the energy trade offs between local computation on the mobile device and wireless communication of mobile clients communicating information to the cloud. Their experiments revealed that bulk data transfers is a good technique to use in order to save energy when communicating with the cloud and computation off loading can improve performace and energy savings in some cases but not all. When designing applications to use computation off loading developers must be careful not to introduce long latencies into the system that could cause decreases in performance and increases in energy consumption **(**Mitten & Nurminen 2010). Shu created a system called eTime that leverages energy used in data transmission between the cloud and mobile device. The system can achieve between a 20 – 30% overall energy saving on trace – driven simulations and real world implementations (Shu 2013). Lee and Zomaya discovered that by energy consumption can be reduced in mobile systems by leverging under utilized resources involved in cloud computing (Lee & Zomaya 2012).

3 DATA-DRIVEN SIMULATION ARCHITECTURE

Ad hoc distributed simulation is defined as a set of simultaneously executing, autonomous simulations connected through a wireless network. Each simulation is responsible for modeling a portion of the overall system determined locally by the simulator itself. Each simulator communicates state predictions to other simulations to model the system as a whole.

Each simulation is a logical process (LP) in conventional distributed simulation terminology and is executed on a mobile device. The mobile devices are connected through a wireless network. Each device is responsible for connecting to a sensor or sensors within the environment in which it operates in order to obtain local traffic state information. Each sensor collects data information such as (speed, acceleration, direction) concerning vehicles that pass through its sensor range. Each sensor communicates this information to nearby mobile devices and the data is then used directly or is aggregated to be used as input for the embedded simulations within the mobile device. Predicted simulator states, e.g., future flow rates on various links may be then transmitted to other simulators. Here, we focus on one simulator of an ad hoc distributed simulation. We consider the energy used by the simulator and that of communications used to drive the simulation and to communicate results produced by the simulator that are distributed to other simulations in the ad hoc distributed simulation.

The proposed energy model represents the different components of a DDDAS ad-hoc distributed simulation. This model separates the energy consumption of the total system into three major components: data communication, data aggregation, and the embedded traffic simulations. The model illustrates each major component in such a system that effects the energy consumption . The system depicted in Figure 1 collects data from sensors spread across the area under study, sends data that was either unaggregated or aggregated at the sensor to the mobile device and uses the data sent in order to drive the embedded simulations on the device to simulate an updated state of the traffic network.

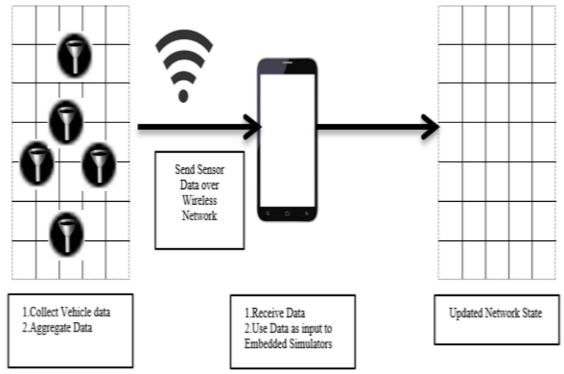

Figure 1: Aggregate at sensor architecture.

Energy for data communication in the mobile device includes receiving data from the sensor network and sending predicted state information to other simulations (LPs). There are several options to transmitting date from the sensor network to the LP. Assuming data is sampled at some given rate, one could simply send each data update directly to the LP. Alternatively, the data samples could be collected in the sensor and periodically a collected set of measurements could be sent as a single message. Still another option is to aggregate the data within the sensor, and transmit an aggregated value, e.g., an average flow rate or the parameters for a probability distribution to the LP. Each of these options will result in different amounts of energy consumption in the system and will impact the results computed by the distributed simulation. For example, aggregating data within the sensor and sending the aggregated results will likely reduce energy consumption to transmit the data, but at the cost of providing less detailed information to the simulation and introducing delays before the online data can be incorporated into the simulation predictions.

The embedded data-driven simulations are responsible for making future state predictions of the traffic network. The amount of energy required by simulations may be significant, and requires exploration. The energy consumed by the simulation includes energy required by the CPU as well as energy used in the memory system and transmitting instructions and data between the two. These depend on the specific modeling approach that is used. Here, we focus on the energy used by transportation models using two widely used abstractions. As discussed momentarily, a model based on cellular automata is evaluated as well as a second based on queuing network abstractions implemented as a discrete event simulation.

4 ENERGY CONSUMPTION: SIMULATION

4.1 Embedded Traffic Simulations

The cellular automata and queuing network models were configured to simulate the traffic of the arterial road network along Peachtree St. located in midtown Atlanta, GA (see Figure 2). This area was selected because of the availability of data. Specifically, traffic data from the NGSIM data set was utilized as the input to develop our simulation models (FHWA 2007). The data was collected on November 8, 2006 during a fifteen-minute time frame from 4:00 PM to 4:15PM. The area includes five intersections, four that are signalized and six road segments. The data set consists of data pertaining to individual vehicle trajectories with time and location stamps, from which the link travel times of individual vehicles could be calculated. Figure 3 reflects a visual representation of the NGSIM data set area. In this study, link travel time refers to the time from when a vehicle enters the arterial link to the time when the vehicle passes the stop-bar at the end of the link. Intersection travel time is excluded).

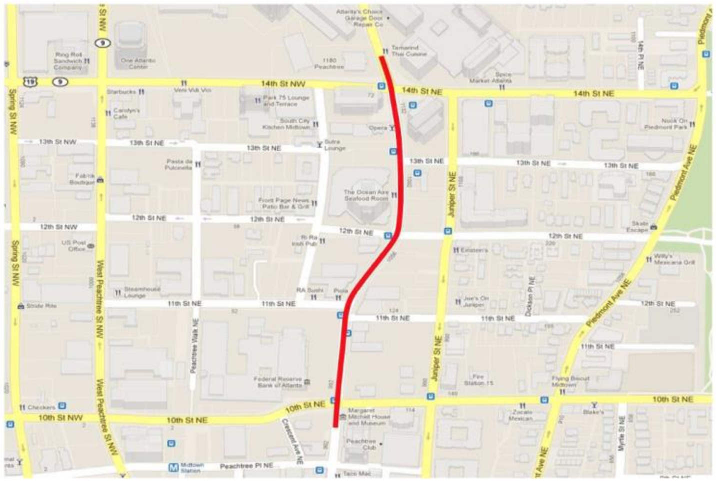

Figure 2: NGSIM study area.

Both simulation models use the same input parameters and assumptions that were used as the basis for all simulations used in this study. Both models were developed in C and implemented as an Android native application. The output of each model is the average travel time for vehicles that are traversing the section of Peachtree St. described earlier. The model was validated by comparing the average travel times produced by the model to those observed in the NGSIM data set.

It was assumed that there were no pedestrians or emergency vehicles. Further the simulation exclude u-turns, aggressive driver behavior, adverse weather conditions, road construction, and vehicle accidents. Due to the data limitations these aspects were not included in the models. The inter arrival time of vehicles entering the simulated area were assumed to be independent and identically distributed following an exponential distribution. We assume that the destination zones of our model have unlimited capacity so that once a vehicle reaches its destination it departs from the system instantaneously.

The input parameters of each model include the historical traffic data collected from the NGSIM data set. Signal timings for each traffic light and probability of vehicle turns for each origin and destination zone were derived from the dataset. The parameters that were varied outside of the given parameters include the traffic intensity and the simulation time.

To simplify our model we assumed that all vehicles were of the same length. We also assumed that all vehicles are identical, and travel with the same acceleration and maximum velocity parameters and had instantaneous deceleration. We assumed that the safe distance between vehicles is uniform for all vehicles.

4.2 Cellular Automata Model

The cellular automata simulation models the system on a relatively low level that allows the users to understand the interactive behavior of objects within the system. In the realm of vehicle traffic systems cellular automata model the micro level dynamics of traffic flow behavior. Individual vehicle behavior can be modeled in such a system. Nagel and Schreckenberg created a well-known cellular automata single-lane model that divides the road into cell segments. Each cell has a state that is considered occupied if a vehicle is there or empty if there is not. The state of the cells change on every iteration based on the neighboring cells surrounding it (Nagel and Schreckenberg 1992). Esser and Schreckenberg implemented an urban traffic network based on Nagal's original model (Esser and Schreckenberg 1997). Statistics such as throughput, travel time and individual vehicle speed and location are computed in the model. Cellular automata models derive macro level traffic flow behavior from micro level dynamics.

Our cellular automata model was implemented in C using the two-lane cellular automata modeling approach. The two lane model approach was proposed and implemented by Rickert and Nagel (Rickert et al. 1996). The model consists of the following modules: cells, vehicles, and road segments. The simulation environment includes a two-dimensional array of 69 X 89 cells, each cell was set to the size of 7.5 meters and can own up to one vehicle at a time. A cell can be in one of five states at any time: empty, normal, source, sink, or a traffic light. A normal cell is one that is a part of a road segment. A source cell represents a location where vehicles enter the system. A sink cell represents the location within the model where vehicles leave the system. A traffic light cell represents a cell where a traffic light is located. Vehicles are stopped based on the state of the traffic light and assigned a turn probability if applicable. An empty cell represents a cell that is currently not occupied by a vehicle. Each cell has a row and column location, street id for the street on which they are currently located, the direction in which they are traveling, and an array of turn probabilities for a vehicle that occupy that cell. Each vehicle has an id, vehicle arrival time, departure time, total time in the system, arrival street, and departure street. The vehicles velocity corresponds to the number of cells the vehicle can proceed forward.

The overall system executes in a time-stepped manner. The tick time pertains to the overall simulation time in seconds. For each time step vehicles are added to the simulation system and traffic lights are updated. Each road segment of cells pertaining to the vehicle lanes are evaluated in a s shaped pattern checking vehicles against the flow of traffic. Each road segment begins its evaluation at the end of the road allowing vehicles closest to exiting the road segment the ability to move first. This then allows the vehicles behind it to have the ability to move forward since once they evaluate the cell before them it is considered empty. Vehicles that have the possibility to move forward to the next traffic cell are moved to the next available cell. A vehicle has the ability to move if the next forwarding cell is empty, if the forwarding cell is a cell pertaining to an intersection the vehicle has the probability to turn. These probabilities are pre-computed and hard coded into the simulation from a input file based on the data from the NGSIM data set. A vehicle moves forward a set number of cells based on the vehicle's current velocity. As long as the vehicle's velocity is below the maximum velocity for all vehicles the vehicle accelerates v + acceleration steps ahead in the system as long as room permits for that number of cells for the vehicle to proceed, if the vehicle is not able to proceed v = v+ acceleration steps ahead it proceeds to move as many cells as it can towards the value v that is available. If a vehicle reaches a cell that contains a traffic light, a vehicle is now in an intersection and the vehicle is assigned a turn probability which is preset at initialization of the traffic network based on the data from the NGSIM data set. If a vehicle is assigned to turn their direction property is changed and they now proceed in that direction. Intersection traffic light states are updated each time step, the state of the light changes based on the length of the phase of each state as determined based on the information provided from the NGSIM dataset. This sequence continues until the simulation cycle is completed.

4.3 Queuing Model

In the queuing model simulation traffic lanes are represented using queues that hold vehicles occupying the lane. The model is event driven. Events with smaller timestamps are processed first and continued until all events have been processed or the simulation has completed.

The discrete event queuing model was also implemented in C. The model consists of the following modules: simulation engine, simulation application, event, vehicle, intersection, section, priority queue, and linked-list and implements a standard event-oriented execution paradigm. The model is driven by the simulation engine which holds the main loop that continuously executes until no events remain or the set simulation end time has been reached. The priority queue is implemented using a binary heap. The simulation application module is responsible for initialization of system variables that start the simulation and calculating output values such as the average travel time. The simulation application is responsible for processing the callback functions and event handlers implement event processing routines. The event handlers include global arrival and departure events and events for each intersection that handle vehicle events which include: arrival, entering, crossing, and departing. Traffic light events are also a part of intersection events. These events are responsible for switching the state of the traffic light when called. Events are created using the event module which creates an event object. Each object has the attributes of an event type including the timestamp and callback function. Vehicles are created using the vehicle module. Each vehicle created has a set origin, destination, id, lane id, and velocity. Section modules represent road lanes between traffic intersections. Each section module object maintains values to attribute to the number of vehicles occupied in each section and a flag indicating congestion. Intersection modules hold attributes corresponding to a road network intersection. Each lane of the intersection is represented using a queue into which vehicles are placed once they enter each intersection. The intersection module is also responsible for handling traffic light signal changes where signal lights states are based on phase lengths. If a vehicle is within an intersection during a green light phase, vehicle events are scheduled to proceed the vehicle forward to the next street section of the system.

4.4 Experiments

The embedded simulations represent the main computational portion of the DDDAS system. Each model is responsible for modeling the vehicle throughput of the arterial network. The cellular automata model must update the position of each vehicle every time step in the simulation. The queuing model is event driven and does not need to process state updates of each vehicle so frequently. However, a priority queue is needed to hold the set of pending events, and a significant amount of energy must be expended inserting and removing events.

Experiments were completed to measure the energy consumption as vehicle arrival rate (Figure 4) and simulation size (Figure 5) are varied. Energy was measured using the Trepn profiler app installed on an Android LG Nexus 5x cellular phone. It is seen that the cellular automata model consumes more energy than the queuing model in these experiments. The cellular automata model must access each vehicle within the road segments each loop iteration causing the need for more computation operations to be performed resulting in larger energy consumption.

.

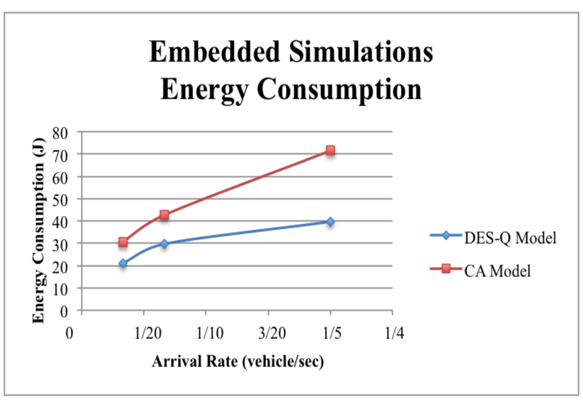

Figure 3: Energy as traffic load varies.

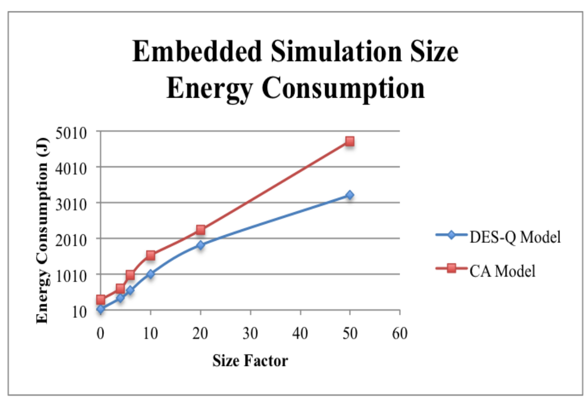

Figure 4: Energy as simulations size varies.

These results quantify the energy cost of using a more detailed model. Figure 3 indicates that increasing the inter arrival rate of vehicles in the system results in an increase in energy consumption due to the increased number of vehicles in the system. Energy consumption in the cellular automata model is impacted to a larger degree than the queueing model as the arrival rate increases. A larger arrival rate results in more vehicles residing in the system. As the cellular automata model must update the state of every vehicle in the system every time step and make updates according to its neighboring cells an increase in arrival rate should reflect an increase in energy. Whereas the queuing model must only process events for vehicles at the front of each queue that have been scheduled at each iteration this impact of additional vehicles on energy usage will be less.

Figure 4 shows the results of increasing simulation size. In the cellular automata simulation the number of cells increases in proportion to network size. Our results show the original network size based on the configuration of the area under study and simulations of areas a factor of four and six times as large. For these experiments the network was replicated by the set input parameters to increase the number of cells in the cellular automata model, and to increase the number of queues and events in the queuing discrete event model. All instances of this experiment use an arrival rate of 1 vehicle every 5 seconds.

5 ENERGY CONSUMPTION: COMMUNICATION

The data streaming and data aggregation models were written as a Java Android application. This application mimics communications of data between sensors and the distributed simulations through a wireless network.

5.1 Data Streaming

A data streaming application was created that is composed of a TCP server socket that communicates to TCP clients sockets over the wireless network. The server socket creates a thread that controls the execution of communication between the server socket and client socket. The server thread is responsible for establishing a connection with the client socket through a given port. Once the connection is established a thread is created to either send or receive data.

The receive thread is responsible for receiving data through the port in which a connection has been established. The received thread establishes an input stream in order to accept data streams sent from the client. The thread continuously accepts data until it is interrupted which occurs if a connection is lost.

The send thread is responsible for sending data through the port in which the connection has been established. Like the receive thread the sending thread establishes a connection from the client in order to begin sending data. The thread continuously sends data until the connection with the client has ended.

5.2 Data Aggregation

Data aggregation is the process of gathering data into summary form. For the purpose of this work data aggregation was used in order to aggregate traffic information in order to drive the embedded traffic simulations. The data aggregation process was set to aggregate information on the client side, which is the sensor in this case.

Aggregating data on the client side assumes that the client is a part of the sensor network. The client collects traffic information and as the information is collected aggregates the collected values. The aggregated values involve summarizing the number of vehicles and the arrival rates of vehicles. These summarized values are then sent to the DDDAS application over the wireless network. The DDDAS application has the option of receiving the values in a set interval fashion. This option alleviates the need to receive continuous data until the sensor has gained enough information to aggregate or the level of traffic is not heavy enough for the change in summary values to occur.

5.3 Experiments

A set of experiments were implemented to evaluate energy consumption. The experimental setup is intended to mirror a DDDAS embedded traffic simulation system where the execution process includes the system receiving real-time sensor data that is aggregated and used as input in the embedded traffic simulations. The simulations produce future state predictions that are sent to other simulations making up the ad-hoc distributed simulation. All energy consumption measurements were evaluated utilizing direct measurements from the Trepn profiler app installed on an Android LG Nexus 5x cellular phone. The profiler was utilized with the Delta settings enabled, which allows the application to collect power data of the entire system during a baseline interval. The average power value is then determined and subtracted from power values determined for the running application in order to give an accurate power measurement of the application.

Data streaming experiments were conducted utilizing the Android phone as the server with the DDDAS application installed on the mobile device. All communication occurred through a WLAN network. A laptop was utilized to represent a sensor in the sensor network and communicated collected sensor data to the mobile device. The experiments show how energy consumption varies with message sizes. The results show measurements when sending and receiving data continuously, and sending data at different payloads between the mobile device and the laptop (sensor).

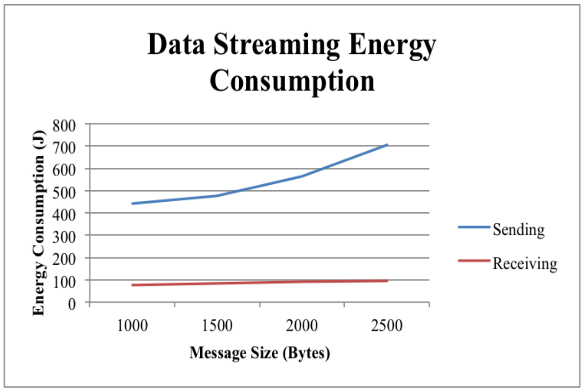

Figure 5: Data streaming energy consumption.

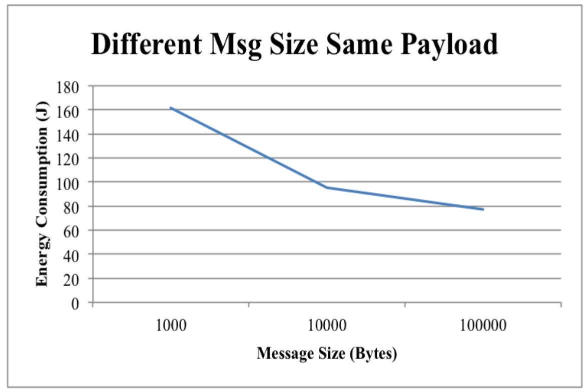

Figure 6: Payload energy consumption.

Figure 5 shows the results of receiving and sending data messages from the sensor and mobile device. The results show that receiving data streams on the mobile device requires significantly less energy than sending data from the mobile device. Both figures show that in the case of sending and receiving data the energy consumption increases with message size, as one would expect. When receiving data messages energy consumption similarly increase with message size. The sending energy consumption shows a steady but more significant increase.

Figure 6 shows the results from an experiment sending 100,000,000 bytes of data using messages of different sizes. As the message size increases the number of messages that need to be send decreases in proportion, and although the power need to transmit larger messages increases the overall transmission time is smaller resulting in less energy consumption. This illustrates that energy can be conserved by collecting data samples in the DDDAS system and sending them as a larger message rather then immediately sending each sample as it is collected. The drawback of this approach is an increased delay to transmit each individual sample. The same experiment was implemented and energy measured on the receiving side of the mobile device. Similar results were obtained.

6 DISCUSSION

DDDAS systems in mobile environments require energy for both communication of data and simulation computations. Figure 7 and Figure 8 show the average *power* consumption (energy per unit time) drawn from sending and receiving data continuously and the power drawn by both simulation models under different arrival rate inputs. Our experiments show that energy consumption for communication dominates energy consumption of the overall system for this traffic network configuration. Communication energy can be reduced by sending data with larger message payload sizes.

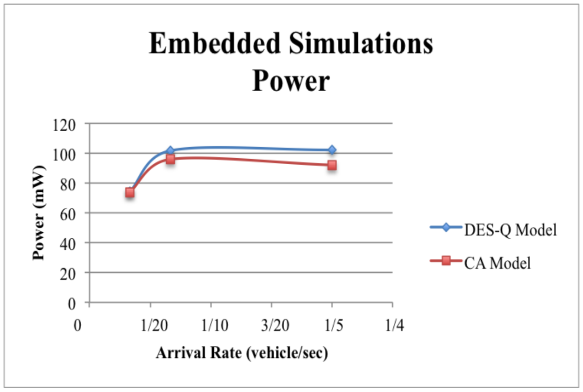

Figure 7: Data Streaming Power.

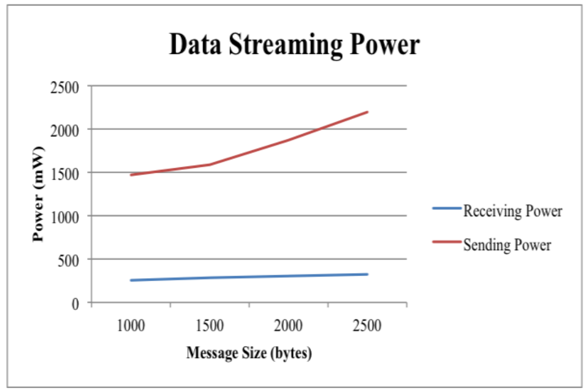

Figure 8: Embedded Simulations Power.

This data indicates that the power consumption of receiving data in the mobile device and executing the embedded simulations is modest relative to that for sending data. Of course, simulating a larger area will result in more energy consumption. Figure 8 shows the average power drawn by the simulation as the network size is increased by a specified factor.

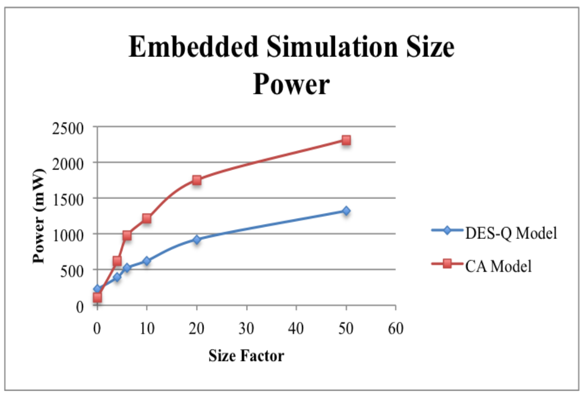

Figure 9: Embedded Simulation Power based on Size.

Figure 9 shows how power consumption increases with the size of the network. This increase is more significant in the cellular automata model compared to the queuing model, as discussed earlier. In relation to the overall DDDAS system Figure 9 shows that compared to original network size the power increase in comparison to receiving data is almost twice as large as the network reaches 6 times its original size.

7 CONCLUSION

The presented work reflects a proposed architecture for embedded distributed simulation systems that are embedded in mobile platforms that are energy constrained. A DDDAS application system for predicting vehicle traffic flow was created and implemented in order to understand the energy consumption components of such a system. The major components were divided into computation and communication.

The computation components of such a system were defined to be the embedded traffic simulations that are responsible for making future state predictions of the traffic network. We compared the widely used cellular automata simulation model and a queuing network simulation. Our experiments show that when energy is the focus the cellular automata model consumes more energy. The CA model's energy consumption increases more rapidly as model size and traffic density increase.

The communication components of the system involved communication between the distributed simulation systems embedded on the mobile device and a sensor or sensors within the traffic network. Our experiments show that communication consumes a significant amount of energy. Sending messages from the mobile device holding the distributed simulation consumes far more energy than receiving messages. Greater energy efficiency is obtained by packing multiple data samples into a single message rather than sending multiple messages, but at the cost of increased delay to receive sampled data. Further, we observed that for these experiments the energy to send data greatly exceeds that required for the simulation computation and receiving messages, though this result depends on the size of the modeled network and the traffic load.

Our work illustrates the energy trends one might encounter under different communication and embedded simulation models used in a DDDAS system designed for predicting traffic network states that are driven by real time data streams.

ACKNOWLEDGMENTS

This research was supported by NSF/AFOSR grant 1462503.

REFERENCES

Biswas, A. and R. Fujimoto. 2015. "Profiling Energy Consumption in Distributed Simulations." In *Proceedings of the 4th ACM SIGSIM/PADS Conference on Principles of Advanced Discrete Simulation*, Association for Computing Machinary.

Darema, F. 2004. "Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements." *Computational Science-ICCS 2004*, Springer: 662-669.

Dongarra, J., H. Ltaief, P. Luszczek and V. M. Weaver. 2012. "Energy Footprint of Advanced Dense Numerical Linear Algebra Using Tile Algorithms on Multicore Architectures." In *Proceedings of the 2012 Second International Conference on Cloud and Green Computing (CGC)*, IEEE.

Esser, J. and M. Schreckenberg. 1997. "Microscopic Simulation of Urban Traffic Based on Cellular Automata." *International Journal of Modern Physics C* 8(05): 1025-1036.

FHWA. (2007) "Next Generation Simulation" http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

Fujimoto, R. and A. Biswas. 2015. "An Empirical Study of Energy Consumption in Distributed Simulations." *2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications* (DS-RT), Institute of Electrical and Electronics Engineers, Inc.

Fujimoto, R., M. Hunter, J. Sirichoke, M. Palekar, H. Kim and W. Suh. 2007. "Ad Hoc Distributed Simulations." In *Proceedings of the 21st International Workshop on Principles of Advanced and Distributed Simulation,* Institute of Electrical and Electronics Engineers, Inc. Computer Society.

Henclewood, D., A. Guin, R. Guensler, M. Hunter and R. Fujimoto. 2010. "Real-time Data Driven Arterial Simulation for Performance Measures Estimation." In *Proceedings of the 2010 Winter Simulation Conference,* edited by B. Johannson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 2057-2069. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Huang, Y.-L., C. Alexopoulos, M. Hunter and R. M. Fujimoto. 2010. "Ad Hoc Distributed Simulation of Queueing Networks." In *Proceedings of the 2010 IEEE Workshop on Principles of Advanced and Distributed Simulation,* IEEE Computer Society: 57-64.

Lee, Y. C. and A. Y. Zomaya. 2012. "Energy Efficient Utilization of Resources in Cloud Computing Systems." *The Journal of Supercomputing* 60(2): 268-280.

Mittal, R., A. Kansal and R. Chandra. 2012. "Empowering Developers to Estimate App Energy Consumption." *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking,* Association for Computing Machinery.

Miettinen, A. P. and J. K. Nurminen. 2010. "Energy Efficiency of Mobile Clients in Cloud Computing." *HotCloud* 10: 4-4.
Nagel, K. and M. Schreckenberg. 1992. "A Cellular Automaton Model for Freeway Traffic." *Journal de physique I* 2(12): 2221-2229.
Neal, S., G. Kantikar and R. Fujimoto. 2014. "Power Consumption of Data Distribution Management for On-line Simulations." In *Proceedings of the 2nd ACM SIGSIM/PADS Conference on Principles of Advanced Discrete Simulation*, Association for Computing Machinary.
Pathak, A., Y. C. Hu, M. Zhang, P. Bahl and Y.-M. Wang. 2011. "Fine-grained Power Modeling for Smartphones Using System Call Tracing." In *Proceedings of the Sixth Conference on Computer Systems*, Association for Computing Machinary.
Rickert, M., K. Nagel, M. Schreckenberg and A. Latour. 1996. "Two Lane Traffic Simulations Using Cellular Automata." *Physica A: Statistical Mechanics and its Applications* 231(4): 534-550.
Rodríguez, R., A. Cortés and T. Margalef. 2009. "Injecting Dynamic Real-time Data into a DDDAS for Forest Fire Behavior Prediction." *Computational Science–ICCS 2009*, Springer: 489-499
Schizas, I. D. and V. Maroulas. 2015. "Dynamic Data Driven Sensor Network Selection and Tracking." *Procedia Computer Science* 51: 2583-2592.
Shi, W., K. Perumalla and R. Fujimoto. 2003. "Power-aware State Dissemination in Mobile Distributed Virtual Environments." In *Proceedings of the Seventeenth Workshop on Parallel and Distributed Simulation, 2003.(PADS 2003).* Institute of Electrical and Electronics Engineers, Inc.
Shu, P., F. Liu, H. Jin, M. Chen, F. Wen and Y. Qu. 2013. "eTime: Energy-Efficient Transmission Between Cloud and Mobile Devices." In *INFOCOM, 2013 Proceedings IEEE*, IEEE.
Sudusinghe, Kishan, Inkeun Cho, Mihaela Van Der Schaar, and Shuvra S. Bhattacharyya. 2014. "Model Based Design Environment for Data-driven Embedded Signal Processing Systems." *Procedia Computer Science* 29: 1193-1202.
Trepn profiler. 2016 Trepn Power Profiler. Accessed: April 1. https://developer.qualcomm.com/software/trepn-power-profiler
Zhang, L., B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao and L. Yang. 2010. "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones." In *Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, Association for Computing Machinary.
Zhang, J., F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu and C. Chen. 2011. "Data-driven Intelligent Transportation Systems: A Survey." *IEEE Transactions on Intelligent Transportation Systems,* 12(4): 1624-1639.

AUTHOR BIOGRAPHIES

SABRA NEAL is a PhD student in the School of Computational Science and Engineering at the Georgia Institute of Technology. She received her B.S. in Computer Science from North Carolina Agricultural and Technical State University. Her email address is sneal6@gatech.edu.

MICHAEL HUNTER is an Associate Professor at the School of Civil and Environmental Engineering at the Georgia Institute of Technology. He received his Ph.D. in Civil Engineering from The University of Texas at Austin. His email address is michael.hunter@ce.gatech.edu.

RICHARD FUJIMOTO is Regents' Professor in the School of Computational Science and Engineering at the Georgia Institute of Technology. He received a Ph.D. in Computer Science and Electrical Engineering from the University of California-Berkeley. His email address is fujimoto@cc.gatech.edu.