Modeling and Analysis of Routing in IoT Networks

Sriram Sankaran
Department of Computer Science and Engineering
Amrita University
Amritapuri, Kollam 690525
Email: ss346@buffalo.edu

Ramalingam Sridhar
Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY 14214
Email: rsridhar@buffalo.edu

***Abstract*—Internet of Things (IoTs) is gaining increasing significance due to real-time communication and decision making capabilties of sensors integrated into everyday objects. IoTs are power and bandwidth-constrained with applications in smarthome, healthcare, transportation and industrial domains. Routing bears significant importance in IoTs where sensors acting as hosts deliver data to the gateways which in turn impacts power consumption. Thus there exists a need for modeling and analysis of routing in IoT networks towards predicting power consumption. In this work, we develop an analytical model of a naive flooding based routing protocol using Markov chains. In particular, we derive steady state transition probabilities of transmit and receive states using protocol execution traces and further utilize them towards predicting power consumption. Our approach to modeling is generic in that it can be applied to routing protocols across domains. Evaluation of the model shows that the predicted values for power consumption lie closer to the actual observations obtained using ns-2 simulation thus resulting in minimal mean square errors.**

I. Introduction

Advances in sensing, computing and communication have changed the internet for people to internet of things (IoTs). IoTs are composed of sensors and actuators embedded into everyday objects that are capable of real-time communication and decision making. These sensors are power and bandwidth-constrained with applications in healthcare, automative, transportation, smart-home and industrial domains. While limited improvements in battery technology have failed to keep up with the advancements in computing devices, the need for managing power consumption becomes necessary. Further power management requires models for estimating the power consumption of devices in the IoTs.

Routing bears significant importance since nodes in a IoT network act as hosts and routers delivering data to the gateways. Many routing protocols have been proposed for sensor networks and are applicable within the IoTs. The routing of data from source to destination impacts the power consumption of forwarding nodes. Due to the random behavior of the network, stochastic methods are a natural fit to studying the power consumption of the individual nodes and the overall network. These methods profile past history of events towards predicting future behavior.

In addition, routing typically involves nodes to discover routes to destinations through beaconing which incurs significant amount of overhead. During beaconing, source nodes flood route requests/ping messages to their neighbors which in turn is rebroadcasted until packets arrive at the destination. The destination responds to the requests and the route is constructed. Further, factors such as beacon interval impact the rate at which beacons are transmitted. Thus there exists a need for quantifying the energy and performance impact of routing protocol and analyzing the associated control and data packet overheads.

Existing approaches for power estimation can be classified into Measurement and Modeling. Measurement based approaches involve measuring the power consumption using external power monitors such as Monsoon Power Monitor. These tools provide a coarse grained estimation of power consumption but do not account for energy consumed by individual components. On the other hand, modeling based approaches estimate power consumption through analytical, empirical, statistical or simulation based methods. Simulation tools such as ns-2 [1], PowerTOSSIM [2], Wattch [3] provide models for power estimation.

In this work, we develop an analytical model for routing in IoT networks using Markov chains. Our contributions include:

- Simulating a naive flooding based routing protocol using ns-2 and analyzing the protocol execution traces
- Modeling the routing protocol using Markov chains from protocol execution traces and deriving steady-state transition probabilities of transmit and receive states
- Analyzing the impact of factors such as beaconing and number of hops on overall performance of the routing protocol and estimating the overheads
- Predicting power consumption using the transition probabilities and comparing the results with the actual observations obtained using ns-2 network simulation

II. Related Work

Numerous works have profiled the power consumption of different components in a smartphone [4]. Recently, developing user adaptive solutions for energy efficiency in mobile devices was explored. Falaki *et al.* [5] analyzed the application patterns of users for power optimization. Fei *et al.* [6] traded off display quality for power savings. Shye *et al.* [7] studied the usage patterns of different users for power optimization. Banerjee *et al.* [8] modeled the interaction of users with batteries for power management. Zhu *et al.* [9] developed

978-1-4673-7309-8/15/$31.00 ©2015 IEEE

an energy-aware scheduling scheme which considers available energy and energy consumed by applications for optimal scheduling.

To estimate the energy consumed by PCs and mobile embedded devices, energy models have been developed at all levels ranging from hardware, system [10], applications, network protocols [11] as well as individual components such as displays [12] [13] and disks [14]. Measurement based methods [15] [16] utilize an external measurement device while modeling relies on cycle-level simulation tools [2] [3] to model the power consumption of individual components. Tiwari *et al.* [17] modeled the power consumption at an instruction-level. McCullough *et al.* [18] utilized performance counters to model the power consumption of full systems. Rivoire *et al.* [19] provided a comparative evaluation of different power models.

Markov Modeling [20] has been utilized in the past for prefetching [21], caching [22], predicting I/O access patterns [23] [24] and dynamic power management [25] [26]. Mini *et al.* [27] proposed a probabilistic approach to predict energy consumption in wireless sensor networks. Recently, markov modeling has been utilized for disk power management where access patterns for a disk are analyzed and transition probabilities are computed. These transition probabilities are further used to predict future disk idleness for power management.

III. IoTs and Routing

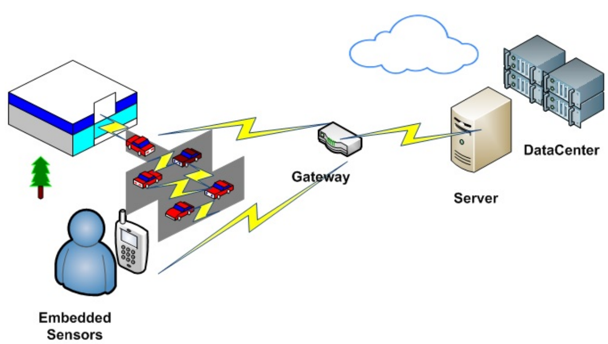

Fig. 1: Internet of Things

The functionality of IoTs is illustrated in Figure 1. IoTs are typically organized into three tiers. Tier I contains multitude of embedded devices monitoring objects and their surrounding areas. Tier II represents gateway nodes which receive data from the embedded devices. Smartphones are typically considered as gateway nodes which are computationally more powerful than the embedded sensors. Tier III contains servers or datacenters which store the data received from gateway nodes for processing. Servers or datacenters perform complex analytics by developing models using the data received from the gateway nodes.

There exists numerous categories of routing protocols each satisfying different operating requirements. Below, we discuss each category and the protocols involved.

Naive Routing:

Naive routing relies on flooding [28] to discover routes to the destination. Assuming nodes can hear their neighbors, source nodes flood route request packets until it reaches the destination. The destination nodes reply back to the source with a route reply message. The source on receiving the route reply message unicasts data packets to the destination along the constructed route. Many of the popular ad hoc routing protocols such as DSR, AODV and DSDV fall under this category.

Hierarchical Routing:

In hierarchical routing, nodes form clusters and a cluster-head is chosen in each cluster to forward data to the sink on-behalf of the cluster nodes. To facilitate load balancing, cluster-heads are rotated among the nodes in the network. Hierarchical routing is suitable for nodes distributed across groups. LEACH [29] is an example of a hierarchical routing protocol within the context of Internet of Things.

Query-based Routing:

Query-based Routing is a radical shift from the notion of naive and hierarchical routing. Many popular paradigms such as Publish-Subscribe emerged from Query-based Routing. The idea behind query-based routing is that nodes disseminate data among themselves such that the querying node retrieves the data from any node in the network. Many of the popular routing protocols such as SPIN [30] and Directed Diffusion [31] fall under this category. Similarly, publish-subscribe based routing protocols work by subscriber nodes (gateways) subscribing to the data published by the publisher nodes (sensors).

Protocols for Internet of Things:

There exists routing protocols developed for sensor networks and further gradually adapted to the IoTs due to the limitations in power and bandwidth. One of the routing protocols, RPL [32], facilitates bi-directional communication between source and sink nodes. Further, multiple modes of operation such as Multipoint-to-Point, Multipoint-to-Multipoint and Point-to-Multipoint communication coexist in RPL. RPL is widely been recognized among the IoT working groups since it interoperates with the IPv6 stack at the network layer and Constrained Application Protocol (CoAP) [33] at the application layer.

IV. Flooding based Routing Protocol

Although numerous categories of routing protocols exist that can be applied to the IoTs, we consider a naive flooding based routing protocol and model its behavior using Markov chains. This is due to the simplicity of the protocol and that more functionalities exist in contrast to the specialized versions such as Directed Diffusion. Below, we describe the flooding based routing protocol.

Figure 2 illustrates the flooding based routing protocol. In a flooding based routing protocol, sink nodes periodically send beacons and other nodes construct route towards the sink. Once routes to the sink are constructed, nodes periodically report to sink node using the UDP protocol. Our protocol is generic in that, numerous functionalities exist and will be explored in the modeling process.

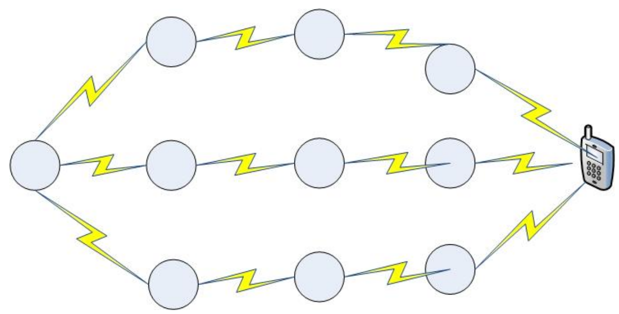

Fig. 2: Flooding based Routing Protocol

We implement the flooding based routing protocol using the ns-2 network simulator and generate execution traces. These traces are further used to model routing using Markov chains as a function of individual node behavior.

A. Markov Chains

A Markov model is represented by total number of states n and the transition probability matrix P. These states are used to reflect the context of the system being modeled. For instance, CPU can choose to be in Active mode performing computations or transition to the idle mode after a certain time interval. There can be any number of states and transitions for a system. The probability with which the system transitions from one state to another reflects the actual behavior of the system. A system is said to possess the Markov property if the future state depends only on the current state and the transition probability. The transition probability is an NxN matrix, where N denotes the number of states. Each of the entries in the matrix corresponds to the probability of transitioning from state i to state j.

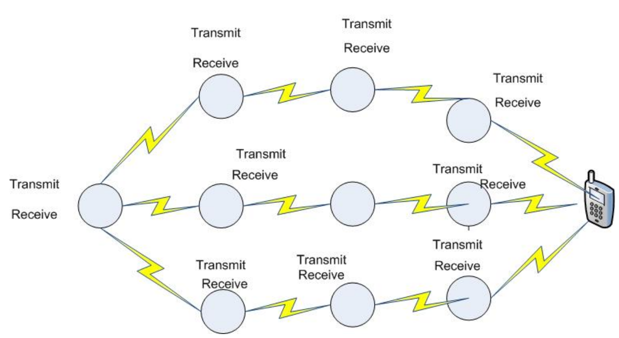

Fig. 3: Markov model

Transition probability matrix is used to predict next set of observations given the current value. We can evaluate the accuracy by comparing the actual and predicted states. We compute the next state using the following equation.

$$S_1 = S_0 * P$$

State	Transmit	Receive
Transmit	0	1
Receive	0.1004	0.8996

State	Transmit	Receive
Transmit	0	1
Receive	0.2513	0.7487

State	Transmit	Receive
Transmit	0.0034	0.9966
Receive	0.4958	0.5042

State	Transmit	Receive
Transmit	0	1
Receive	0.2347	0.7653

TABLE I: Radio State Transition Probabilities for Node 0, Node 1, Node 10 and Node 50

where S_0, S_1 and P refer to the initial state, next state and transition probability matrix respectively.

In a regular Markov chain, successive state matrices always approach a unique stationary matrix called the equilibrium or steady state. The steady state can be computed using the following equation.

$$S * P = S$$

where S, and P refer to the stationary matrix and transition probability respectively.

B. Model Construction

In a routing protocol, each node can choose to transmit or receive depending on node characteristics. For instance, a node closer to sink can receive as well as transmit packets more often than those farther from the sink. Based on these insights, we construct the Markov chain as a function of number of nodes in the network. Each node in the Markov chain can be in 2 different states, "Transmit" and "Receive". It can choose to be in Transmit mode performing transmissions or transition to the Receive mode after a certain time instant. Figure 3 pictorially describes the Markov chain.

We construct the model using execution traces from the ns-2 network simulator. In particular, we observe the time series sequence of routing events in the trace files and wrote Perl scripts to extract the events for each node. The events extracted for each node contain a combination of T and R messages to indicate transmission and reception respectively. Table I contains the results for the transition probabilities for different nodes in the network. Further these transition probabilities used to generate the stationary matrix for each node.

The transition probabilities indicate that when a node starts with the "Transmit" state, it is most likely to transition to the "Receive" state since the probability of transitioning from the former to the latter is close to one. Similarly, when a node starts with the "Receive" state, the probability of transitioning to the same state is more likely than transitioning to the "Transmit" state.

V. Model Analysis

In this section, we quantify the energy and performance impact of flooding based routing protocol for IoT networks. In particular, we investigate the impact of factors such as beaconing and number of hops on the overall performance and estimate control and data packet overhead of the routing protocol. The insights obtained from the analysis would enable developers to design optimized versions of the protocol considering context and usage behavior. Below, we present our analysis of the factors.

A. Impact of Beaconing

Beaconing is typically utilized for location tracking, discovering routes to destinations and tracking neighbors through keep-alive requests. One of the most important factors that affect performance is the beacon interval in the route discovery process. If the beacon interval is too small, the number of beacons generated becomes huge. On the other hand, a higher beacon interval incurs a lesser number of generated beacons.

In our simulation, we measured the amount of remaining energy when beacons were transmitted at varying intervals. Figure 4 contains the results for the simulation. We assign an initial energy of 100 J and beacon interval of 10s, 30s, 50s, 70s and 90s. The results indicate that the increase in beacon interval does not incur a significant difference in remaining energy consumption. To alleviate the overhead caused by flooding, key-based routing [28] is used to discover routes. Key-based routing protocols utilize a distributed hash table (DHT) to store routes for each of the nodes in the network. It would be worthwhile to model key-based routing and analyze energy-performance trade-offs.

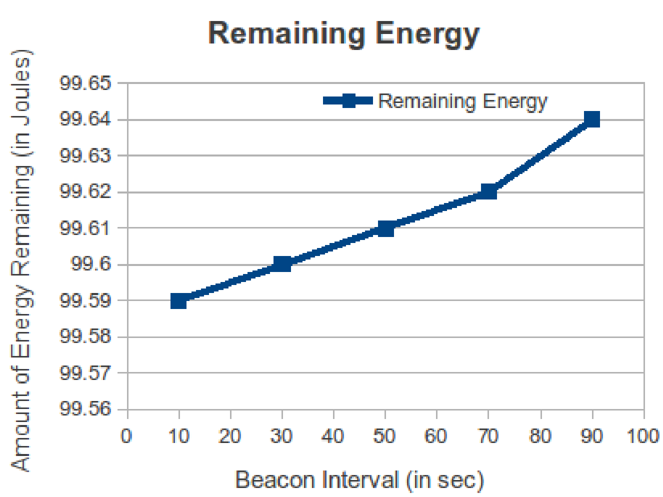

Fig. 4: Impact of Beaconing

B. Control Packet Overhead

We define control packet overhead as the average number of beacons generated per second. Beaconing incurs significant amount of overhead since nodes receive duplicate packets from neighbors. In our simulation, we estimate control packet overhead by computing the sum of number of beacons generated for each beacon interval divided by the simulation time. Control packet overhead $Overhead_{Control}$ can be computed using the following equation.

$$Overhead_{Control} = (\sum_{i=1}^{n} N_i)/T_{sim}$$

where n, N_i and T_{sim} refer to total number of beacon intervals, number of packets generated during beacon interval i and total simulation time respectively.

In our simulation, we estimate control packet overhead for varying beacon intervals. Figure 5 contains the results for the simulation. The results indicate that as beacon interval increases, total number of beacons generated decreases due to the reduction in the number of beacon transmission periods.

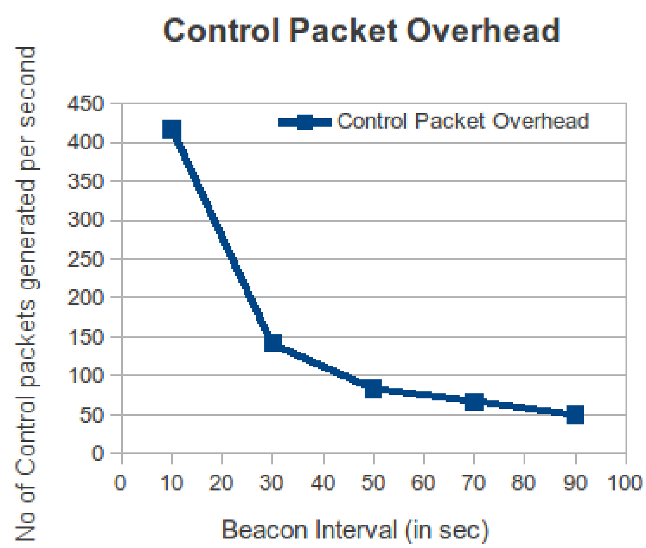

Fig. 5: Control Packet Overhead

C. Data Packet Overhead

We define data packet overhead as the average number of data packets generated per second. Data packets are routed along the constructed path. Performance is proportional to the number of hops to the destination. In our simulation, we estimate data packet overhead by computing the sum of number of data packets generated for each hop divided by the simulation time. Data packet overhead $Overhead_{Data}$ can be computed using the following equation.

$$Overhead_{Data} = (\sum_{i=1}^{h} N_i)/T_{sim}$$

where h, N_i and T_{sim} refer to total number of hops to the destination, number of data packets for each hop i and total simulation time respectively.

In our simulation, we set the total simulation time to be 500 seconds and estimated data packet overhead for varying number of hops. Figure 6 contains the results for the simulation. The results indicate that as number of hops increases, total number of data packets generated increase proportionally.

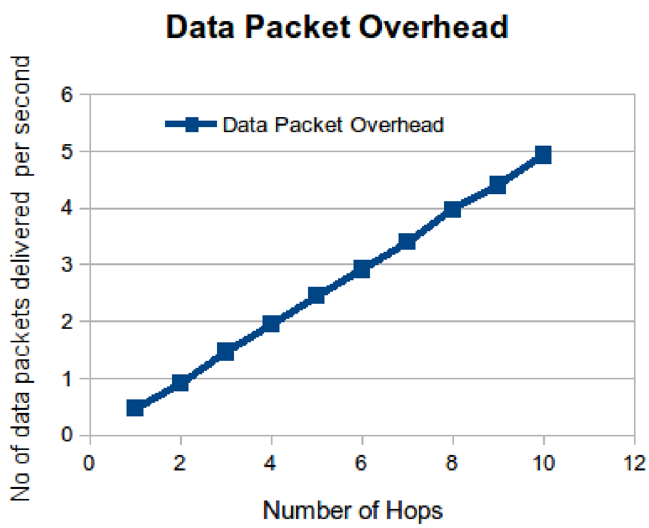

Fig. 6: Data Packet Overhead

D. End-to-End Delay

End-to-end delay refers to the time taken by the packet to reach the destination. It is directly proportional to the number of hops between source and destination. This means that increase in the number of hops incurs a corresponding increase in end-to-end delay.

In our simulation, awk scripts were written to estimate end-to-end delay for varying number of hops. Figure 7 contains the results for the simulation. The results indicate a linear relationship between number of hops and end-to-end delay. This can further be used to develop linear models for delay prediction towards optimizing the routing protocol.

VI. Model Evaluation

We evaluate the model by running the simulation for varying seeds. Since nodes are distributed randomly, we ran the simulation for one hundred iterations and computed the transition probabilities for each iteration. After deriving the steady state, transition probabilities from each of the iterations were averaged by node and the resultant transition probability for each node was computed. We believe that running the simulation for varying number of seeds is necessary to incorporate the randomness in each iteration.

A. Predicting Power Consumption

In this subsection, we use the steady state transition probabilities derived for each node towards predicting power consumption. These probabilities give an accurate picture of the individual node behavior and the overall network. Power

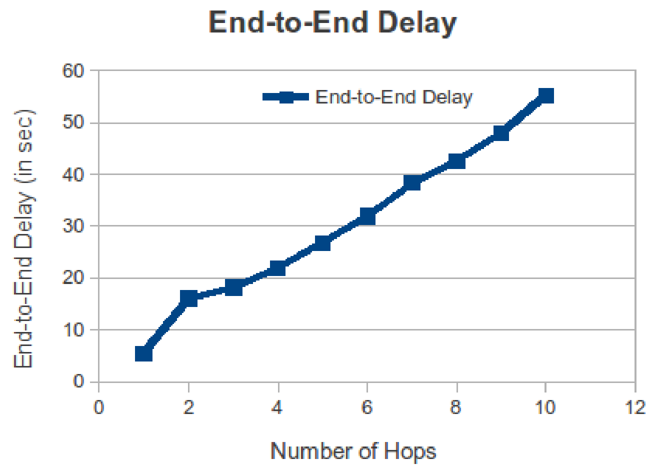

Fig. 7: End-to-End Delay

TABLE II: Simulation Parameters

Parameter	Value
Number of nodes	100
Seed	1-100
Topology	120*120
MAC/PHY	802.15.4
Channel	Wireless Channel
Radio Propagation	Two Ray Ground
Beacon Interval	10,30,50,70,90 s
Constant Bit Rate Interval	2s
Bandwidth	0.1Mb

consumed by the radio P_{radio} can be computed using the following equation

$$P_{radio} = P_{Transmit} + P_{Receive} + P_{Sleep} + P_{Idle}$$

where $P_{Transmit}$, $P_{Receive}$, P_{Idle} and P_{Sleep} refer to the power consumed during transmission, reception, idle and sleep respectively.

In the above equation, we do not consider idle and sleep modes since transmission and reception contribute primarily towards power consumption in routing in contrast to idle and sleep modes. Further, sleep mode consumes negligible amount of power in the routing process. Thus, power consumed by the radio becomes

$$P_{radio} = P_{Transmit} + P_{Receive}$$

We incorporate the transition probabilities for transmission and reception in the following manner. In the above equation, P_{radio} becomes

$$P_{radio} = p_{transmit} * P_{Transmit} + p_{receive} * P_{Receive}$$

where $p_{transmit}$ and $p_{receive}$ refer to the steady state transmission and reception probabilities derived for each node. We obtain the values for $P_{Transmit}$ and $P_{Receive}$ to be 1.5W and 1W respectively from the Lucent WaveLAN datasheet. Substituting the values in the above equation, we compute the radio power consumption. To validate the accuracy of the prediction, we obtain the ground truth using ns-2 network simulations. Mean Square Error (MSE) can be computed using the following equation

$$MSE = P_{Actual} - P_{Predicted}$$

where P_{Actual} and $P_{Predicted}$ refer to the actual and predicted power consumption respectively.

Figure 8 contains the results for the Cumulative Distribution Function (CDF) of the mean square error computed for each of the nodes in the network. The results show that predicted observations lie closer to the actual ones thus resulting in minimal mean square errors.

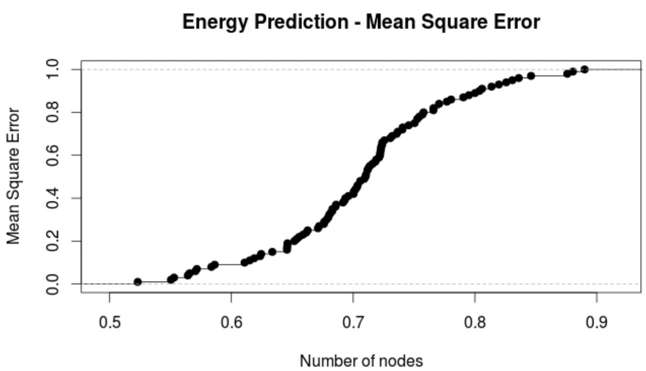

Fig. 8: Mean Square Error

B. Overhead

In addition to the errors associated with predicting power consumption, we evaluate the overhead involved in constructing the model and predicting power consumption for the nodes in the network. The factors contributing to the overhead are processing protocol execution traces and extracting communication patterns, deriving steady state transition probabilities and predicting power consumption for each of the nodes in the network.

Figure 9 contains the results for CDF of the model overhead for the nodes in the network. It is evident from figure 9 that there exists minimal overhead incurred towards model construction and prediction for each of the nodes in the network.

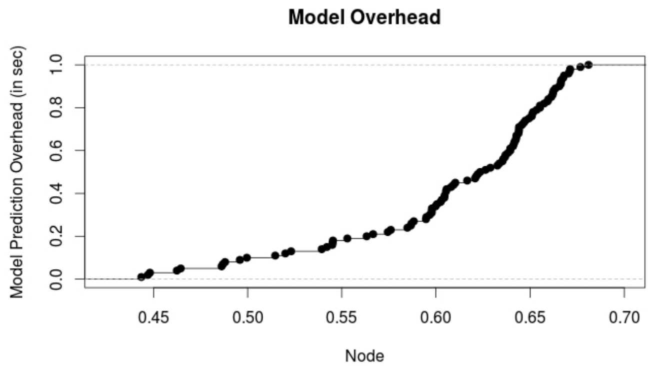

Fig. 9: Model Overhead

VII. Conclusion

In this paper, we modeled and predicted power consumed by routing for Internet of Things. In particular, routing protocols proposed for ad hoc and sensor networks were surveyed and a naive flooding based routing protocol was modeled using Markov chains. Our modeling is generic in that it can be applied to any routing protocol. We derive steady-state transition probabilities from protocol execution traces for each node and further utilize them towards predicting power consumption. Further the impact of factors such as beaconing and number of hops on overall performance of the routing protocol were analyzed and the resulting overheads were estimated. Evaluation of the model showed that the predicted values for power consumption lie closer to the actual observations obtained using ns-2 simulations thus resulting in minimal mean square errors.

References

[1] K. Fall and K. Varadhan, "The network simulator–ns-2," *URL: http://www. isi. edu/nsnam/ns*, 2007.

[2] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. ACM, 2004, pp. 188–200.

[3] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proceedings of the 27th Annual International Symposium on Computer Architecture*, ser. ISCA '00. New York, NY, USA: ACM, 2000, pp. 83–94.

[4] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone." in *USENIX annual technical conference*, 2010, pp. 271–285.

[5] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, ser. MobiSys '10. ACM, 2010, pp. 179–194.

[6] L. Z. Y. Fei and N. K. Jha, "An energy-aware framework for coordinated dynamic software management in mobile computers," in *Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, ser. MASCOTS '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 306–317.

[7] A. Shye, B. Scholbrock, and G. Memik, "Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, ser. MICRO 42. ACM, 2009, pp. 168–178.

[8] N. Banerjee, A. Rahmati, M. D. Corner, S. Rollins, and L. Zhong, "Users and batteries: interactions and adaptive energy management in mobile systems," in *Proceedings of the 9th international conference on Ubiquitous computing*, ser. UbiComp '07. Springer-Verlag, 2007, pp. 217–234.

[9] T. Zhu, A. Mohaisen, Y. Ping, and D. Towsley, "Deos: Dynamic energy-oriented scheduling for sustainable wireless sensor networks," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2363–2371.

[10] A. Roy, S. M. Rumble, R. Stutsman, P. Levis, D. Mazières, and N. Zeldovich, "Energy management in mobile devices with the cinder operating system," in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys '11. ACM, 2011, pp. 139–152.

[11] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '09. ACM, 2009, pp. 280–293.

[12] M. Dong, Y.-S. K. Choi, and L. Zhong, "Power modeling of graphical user interfaces on oled displays," in *Proceedings of the 46th Annual Design Automation Conference*, ser. DAC '09. ACM, 2009, pp. 652–657.

[13] M. Dong and L. Zhang, "Chameleon: a color-adaptive web browser for mobile oled displays," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ser. MobiSys '11. ACM, 2011, pp. 85–98.

[14] N. G. F. Z. A. K. J. Zedlewski, S. Sobti and R. Wang, "Modeling hard-disk power consumption," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, ser. FAST '03. Berkeley, CA, USA: USENIX Association, 2003, pp. 217–230.

[15] J. Flinn and M. Satyanarayanan, "Powerscope: a tool for profiling the energy usage of mobile applications," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, Feb 1999, pp. 2–10.

[16] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "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*, ser. CODES/ISSS '10. ACM, 2010, pp. 105–114.

[17] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: A first step towards software power minimization," *IEEE Transactions on VLSI Systems*, vol. 2, pp. 437–445, 1994.

[18] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, "Evaluating the effectiveness of model-based power characterization," in *USENIX Annual Technical Conf*, 2011.

[19] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A comparison of high-level full-system power models," in *Proceedings of the 2008 Conference on Power Aware Computing and Systems*, ser. HotPower'08. USENIX Association, 2008, pp. 3–3.

[20] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[21] R. Joseph and M. Martonosi, "Run-time power estimation in high performance microprocessors," in *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*, ser. ISLPED '01. ACM, 2001, pp. 135–140.

[22] K. Li, W. Qu, H. Shen, D. Wu, and T. Nanya, "Two cache replacement algorithms based on association rules and markov models," in *Semantics, Knowledge and Grid, 2005. SKG '05. First International Conference on*, Nov 2005, pp. 28–28.

[23] J. Oly and D. A. Reed, "Markov model prediction of i/o requests for scientific applications," in *Proceedings of the 16th International Conference on Supercomputing*, ser. ICS '02. New York, NY, USA: ACM, 2002, pp. 147–155.

[24] T. M. Madhyastha and D. A. Reed, "Input/output access pattern classification using hidden markov models," in *Proceedings of the Fifth Workshop on I/O in Parallel and Distributed Systems*, ser. IOPADS '97. New York, NY, USA: ACM, 1997, pp. 57–67.

[25] Q. Qiu and M. Pedram, "Dynamic power management based on continuous-time markov decision processes," in *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference*, ser. DAC '99. New York, NY, USA: ACM, 1999, pp. 555–561.

[26] T. Simunic, L. Benini, P. Glynn, and G. De Micheli, "Dynamic power management for portable systems," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 11–19.

[27] R. A. Mini, B. Nath, and A. A. Loureiro, "A probabilistic approach to predict the energy consumption in wireless sensor networks," in *IV Workshop de Comunicao sem Fio e Computao Mvel*, 2002, pp. 23–25.

[28] T. Zahn, G. O'Shea, and A. Rowstron, "An empirical study of flooding in mesh networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 2, pp. 57–58, Oct. 2009.

[29] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, ser. HICSS '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 8020–.

[30] W. R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999, pp. 174–185.

[31] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 56–67.

[32] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," 2012.

[33] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.