Performance Evaluation of A Power Management Scheme for Disruption Tolerant Networks

Yong Xi, M. Chuah
CSE Department
Lehigh University
Bethlehem, PA 18015

Abstract— **Disruption Tolerant Network (DTN) is characterized by frequent partitions and intermittent connectivity. Power management issue in such networks is challenging. Some existing power management schemes for wireless networks cannot be directly applied to DTNs because they assume the networks are well-connected. Since the network connectivity opportunities are rare, any power management scheme deployed in DTNs should not worsen the existing network connectivity. In this paper, we design a power management scheme called context-aware power management scheme (CAPM) for DTNs. Our CAPM scheme has an adaptive on period feature that allows it to achieve high delivery ratio and low delivery latency when used with Prophet, a recently proposed DTN routing scheme. Via simulations, we evaluate the performance of the CAPM scheme when used with the Prophet routing scheme in different scenarios e.g. different traffic load, node speeds and sleep patterns. Our evaluation results indicate that the CAMP scheme is very promising in providing energy saving (as high as 86%) without degrading much the data delivery performance.**

Keywords: power management; disruption tolerant networks; sleep pattern; optimization

I. INTRODUCTION

A Disruption Tolerant Network (DTN) is a network of nodes with frequent partition and intermittent connectivity. Recent research interests in this area include network architecture design and different routing algorithms for DTNs [1][2][3][4][5][6]. The routing schemes provide intra and inter-region data delivery services.

Another research effort is related to efficient power management in DTNs. Most wireless devices can operate in different power consumption modes: idle listening, sleeping, transmitting, and receiving. Studies show that idle listening takes high energy consumption, which hence should be reduced and avoided when it is not necessary. The objective of any power management scheme is to minimize nodes' idle listening durations while maintaining network connectivity so that data delivery performance does not degrade.

Wake-up based power management schemes in multi-hop wireless networks fall into three categories [8][9][10][16],[21],[22],[23],[24]: scheduled rendezvous, asynchronous, and on-demand. Scheduled rendezvous schemes [8],[21],[22] require sleeping nodes to wake up at the same time. Asynchronous wake-up mechanisms [16],[20], however, do not require time synchronization among different nodes. The sleep and wake up schedules in asynchronous wake-up schemes are designed such that any two neighboring nodes will have overlapped active periods within a specified number of sleep cycles. On-demand schemes [23],[24] usually uses a second low power wireless interface which remains on while the high power wireless interface is turned on only when necessary. However, on-demand schemes are not suitable for DTNs since the lower power radio often does not have enough transmission and reception ranges to allow nodes that are sparsely distributed to discover one another.

In [10][11], Jun etc did some studies on power management issues in DTNs. Power management design in DTNs faces more challenges. For example, nodes in sparse network suffer more network partitions. Thus, any deployed power management scheme should not worsen the existing connectivity opportunities. In this paper, we are interested in the following questions:

1. Can one design a power management scheme for DTNs that is effective when used with different routing schemes?

2. What is the impact of node movement speed or traffic load on the data delivery performance and energy saving capability of a power management scheme when used with any DTN routing scheme?

In this paper, we propose an asynchronous power management scheme called the context-aware power management (CAPM) scheme for DTNs. We first describe how the CAPM scheme works. Via simulations, we study how the data delivery performance changes when the CAPM scheme is used with two DTN routing schemes, namely (a) the MaxProp scheme [12], and (b) the PROPHET scheme [13]. We also perform sensitivity analysis of the data delivery performance with the CAMP scheme in different scenarios. For example, we investigate how the data delivery performance changes with different traffic models, different node speeds etc. The remainder of this paper is organized as follows. We provide a brief review of related work in Section 2. In Section 3 we present our power management scheme in detail. In Section 4, we describe our simulation setup and present our simulation results. We conclude in Section 5 with some discussions on future work .

II. RELATED WORK

A. *Wakeup scheduling in multi-hop wireless networks*

Many wakeup scheduling schemes were proposed for efficient power management in multi-hop wireless networks. In [8][9][10][16], they were summarized into three categories: scheduled rendezvous, asynchronous, and on-demand. In scheduled rendezvous mechanisms, nodes operate in predetermined wakeup patterns. Clock synchronization is generally assumed in these mechanisms. Asynchronous mechanisms do not require time synchronization, but the sleep pattern must be carefully designed to ensure that there are necessary wakeup overlaps for communications. In on-demand wake-up schemes, nodes are supposed to be awakened in an on-demand basis. This is usually accomplished by employing a second wireless interface which has extremely low power consumption.

B. *Routing in Intermittently Connceted Networks*

Several routing schemes have been proposed for DTNs [3],[12],[13].Here, we give an overview of Prophet [13], the DTN routing protocol that we used in this paper.

PROPHET. In [13], Lindgren etc designed a probabilistic routing protocol called PROPHET that uses history of encounters and transitivity information for intermittently connected networks. This probabilistic routing scheme establishes a probabilistic metric called delivery predictability at every node *a* for each known destination *b*. When two nodes meet, they exchange the delivery predictability information they stored. This information is used to update the estimated delivery predictability to the destination. A message is transferred to the other node if the delivery predictability of the destination of the message is higher at the other node.

C. *Power consumption model*

Our power consumption model is the same as the one used in [10],[11]. This power consumption model is based on nodes activity states. The source of power consumption of a node could come from four activity states: idle, sleep, transmit, and receive. We define each state in Table 1.

Table 1: Power consumption model

State		Idle	Sleep	Transmit	Receive
Processor	Active	√		√	√
	Sleep		√		
Radio Board	Transmit			√	
	Receive				√
	Sleep	√	√		

More studies on the power consumption model can be found in [7][17] and some specific data is listed in [18]. In our paper, we use the same energy parameters in [10].The values of the energy parameters are replicated in Table 2.

Table 2: Power Usage

State	Idle	Sleep	Transmit	Receive
Power (W)	0.1791	0.0141	0.2818	0.2053

III. OVERVIEW OF THE CAPM SCHEME

In this section, we describe our context-aware power management (CAPM) scheme for Disruption Tolerant Networks. Our scheme includes neighbor discovery and data delivery procedures.

A. *Preliminary*

We assume that each node independently operates its own wakeup schedule. The sleeping pattern of the CAPM scheme is shown in Figure 1. Each node runs a fixed length duty cycle. In each duty cycle, the node wakes up for a fixed or adaptive period and then sleeps for the remaining time. We refer to the tuple *(wakeup, sleep)* as the sleep pattern in this paper. Each duty cycle consists of one wakeup and one sleep period.

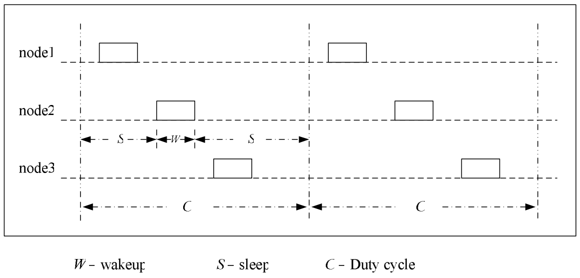


Fig. 1. Random wakeup with fixed duty cycle

B. *Neighbor Discovery*

Neighbor discovery is used to find neighbors that can be used as potential next-hop nodes for data forwarding. The neighboring nodes that are discovered by a node are referred to as its contacts. To discover neighbors, we assume each node periodically broadcasts a beacon when it wakes up. The beacon message provides information about its node identifier, the time it will remain active. A node that wakes up and have data for delivery piggybacks a delivery notification extension to its beacon message. The delivery notification contains the information about the node identifiers of the destinations of stored messages. An active node that receives a delivery notification with its identifier included will send a delivery accept message to the node that sends the delivery notification.

Via some examples, we illustrate how our neighbor discovery scheme works.

In Figure 2-a, all nodes have no data for delivery, so each of them broadcasts a regular beacon message upon waking up. In Figure 2-b, node 1 has data for delivery. When it wakes up, it broadcasts a beacon with a piggybacked delivery notification extension. Since node 2 and node 3 are asleep, there is no reply to node 1's beacon message. But when node 2 wakes up and broadcasts its regular beacon, node 1 will receive this beacon. Then, node 1 can deliver its stored data to node 2. Similar actions are taken when node 3 wakes up later. In Figure 2-c, node 2 has data for delivery. Thus, it issues a regular beacon with piggybacked delivery notification extension upon waking up. Let us assume that node 2 has data for node 1. Since node 1 is active, it will send a delivery acceptance message to node 2 when it hears node 2's delivery notification. Thus, node 2 can discover node 1 and deliver messages to node 1. In Figure 2-d, node 3 has stored data for delivery to nodes 1 & 2. Again, after node 3 sends a regular beacon with piggybacked delivery notification, nodes 1 & 2 can each reply with a delivery acceptance message. Thus, node 3 can discover both nodes and deliver stored messages to them.

In Figure 3, we illustrate how each node maintains a neighbor list after receiving the beacon messages from its neighbors. Let us assume that each node has a fixed on period (W). Upon receiving a beacon from Node 2, Node 1 computes its remaining on time as (t_1 +W – t_4) and sends a delivery accept message to Node 2. Node 2 will insert node 1's information into its neighbor list upon receiving this delivery accept message. Similarly, when node 2 receives node 3's regular beacon at time t8, node 2 again updates its neighbor list as shown in Figure 3. We also show in Figure 3 how node 1 updates its neighbor list upon receiving a beacon with piggybacked delivery notification from node 2 and a regular beacon message from node 3.

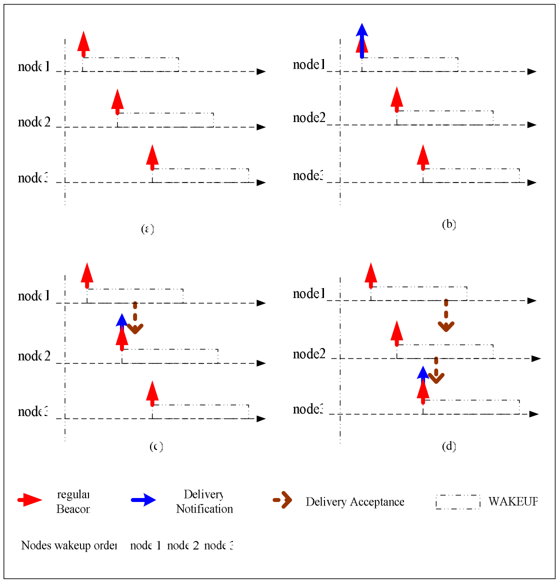
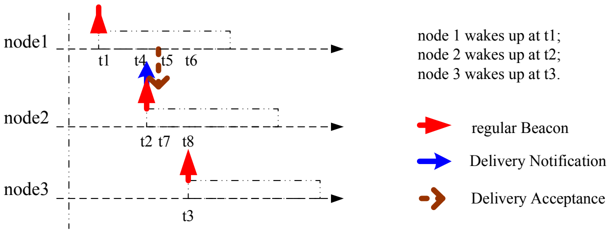


Fig. 2. Neighbor Discovery



Neighbor knowledge (*W* – wakeup length):

node1

Node Id	Remaining Active Time	Time Stamp
2	*W*	t4
3	*W*	t6

node2

Node Id	Remaining Active Time	Time Stamp
1	*W* – (t5 – t1)	t7
3	*W*	t8

Fig. 3. Construction of Neighbor List

C. Data Delivery Scheme

To ensure high delivery ratio and low message delivery latency, we need to make sure that (a) the sending node knows when the receiving node is active, (b) the receiving node must have sufficient remaining on time to receive all the messages that a node has for it, (c) a packet scheduling scheme that transmits old data first, and (d) a node proactively discovers neighbors at a rate that can balance between energy consumptions and the expected data delivery performance.

There are four scenarios to consider for data delivery. Scenario 1: Both the sending and receiving nodes are active. When a packet arrives, the sender can immediately deliver the packet to the receiving node (as shown in Figure 4(a). Scenario 2: The sending node is active and has data for the receiving node which is not active (as shown in Figure 4(b)). Upon receiving the regular beacon sent out by the receiving node when it wakes up, the sending node can check if it has data for this newly wakeup node. If the sending node finds stored data for this newly wakeup node, the sending node will immediately deliver the data packets after a random backoff delay. Scenario 3: the sending node wakes up near the end of the active period of the receiving node. Thus, upon receiving the beacon piggybacked with delivery notification from the sending node, the receiving node extends its active period long enough to receive all the packets that the sending node intends to send. This is illustrated in Figure 4(c). The last scenario to consider is when the sending node and the receiving node do not have any overlap in their sleep pattern (shown in Figure 4(d)). In this case, we allow the sending node to wake up for one whole duty cycle if it has any packets that cannot be delivered to any next hop node after waiting for K duty cycles (K is configurable and set to 1 in Figure 4(d)) so that it can find the next-hop node.

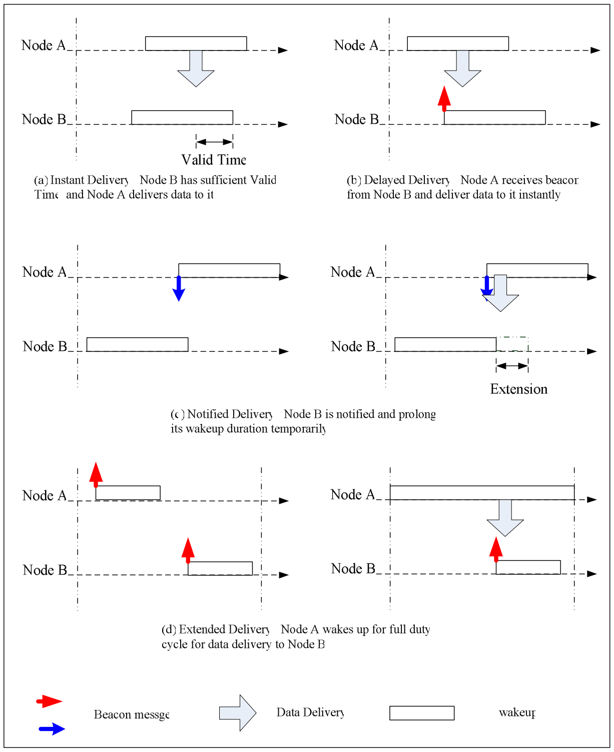
Since the two nodes randomly select on period, their on periods may not coincide and hence they may not discover one another. To allow them opportunities to discover each other, we allow a node to turn on for a full duty cycle when it has queued packets that have been waiting for K duty cycles.

IV. PERFORMANCE EVALUATION

A. *Simulation Setup*

In order to evaluate our power management scheme, we implemented our CAPM scheme and the two DTN routing schemes in ns-2 [19]. The performance metrics we used in our evaluation are: (1) *Delivery Ratio*, which is the successfully received number of data divided by the number of total delivered data. (2) *Normalized Energy Consumption*, which is the ratio between the energy consumption with power management and the energy consumption in the absence of power management. This metric is a measure of energy efficiency of our power management scheme. (3) *End-end delay*, (4) *Average number of hops it takes for message delivery*, and (5) the effective power-on ratio. Since the on period is adaptively adjusted based on traffic conditions, we are interested in knowing what the effective power-on ratio is.

In our simulation, we use a network scenario with 40 nodes distributed over 1000x1000m^2. The nodes move according to the random waypoint model. We set the pause time to be 10 seconds, and the maximum node speed to be 5 m/s. All the nodes communicate using a normal transmission range of 250 m and a bandwidth of 2 Mbps. As we mentioned before, we use the same energy model as in [10] (refer to Table 2). We use constant bit rate traffic as our traffic model. We assume that there are 10 CBR flows. Unless otherwise stated, the sources and destinations of these flows are randomly selected among the 40 nodes. The packet size is 512 bytes and the traffic generation rate for each flow is varied from 0.25 pkts/sec to 3 pkts/sec. We run each simulation for 600 seconds with a warming up period of 1000 seconds and the reported simulation results are based on the average of 5 runs.



(a) Instant Delivery Node B has sufficient Valid Time and Node A delivers data to it

(b) Delayed Delivery Node A receives beacon from Node B and deliver data to it instantly

(c) Notified Delivery Node B is notified and prolong its wakeup duration temporarily

(d) Extended Delivery Node A wakes up for full duty cycle for data delivery to Node B

Fig. 4. Data Delivery for our CAPM scheme

We first evaluate the performance of the PROPHET routing scheme in the absence of power management. Without power management, we assume that the beacons are sent every 5 seconds.Then, we investigate the impact of our power management scheme with PROPHET as the DTN routing protocol under different traffic loads, sleep patterns, node moving speeds.

B. *Power management with PROPHET*

In this section, we evaluate the impact of having a power management scheme on the data delivery performance of a DTN running the Prophet routing scheme. For the traffic model, 10 CBR flows with randomly selected sources and destinations are used. We vary the traffic generation rate of each flow from 0.25 pkts/sec to 3 pkts/sec and evaluate the packet delivery ratio, the delay and normalized energy consumption as we vary the duty cycle duration with a fixed *r* (r=W/S) ratio.

Figures 5 to 9 show the delivery ratio, the average delay, the normalized energy consumption, the effective wakeup ratio and the average number of delivery hops respectively obtained from our simulation results. Figure 5 shows the delivery ratio with a fixed r ratio but different sleep cycle duration. The results indicate that with increasing T_{cycle} , the delivery performance drops. With increasing S, the nodes may not be able to discover contacts to efficiently deliver the messages. Figure 6 plots the average delay with different sleep cycle durations. Compared to the average delay achieved using a sleep cycle of 1.05 second, the average delay increases by 100% and 300% when the sleep cycle is 4.2 seconds and 8.4 seconds respectively. Figure 7 plots the normalized energy consumption with the CAMP scheme turned on. We see that one can achieve a 70-85% energy saving while still maintaining comparable delivery performance as in the case without power management. Figure 8 plots the effective wakeup ratio. Since the CAMP scheme allows the nodes to turn on a full cycle sometimes to discover next-hop nodes, it is interesting to know the effective wakeup ratio. Figure 9 shows the average hop used by the routing schemes with different sleep cycles. The plots show that with the power management scheme, packets may take longer routes to be delivered.

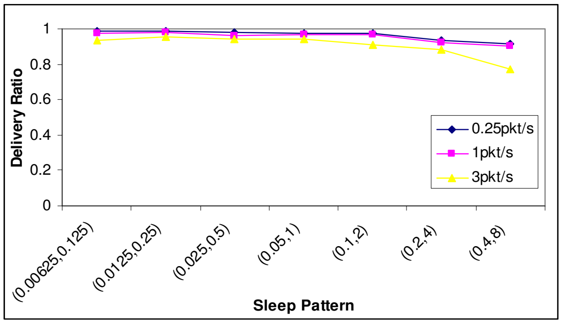


Fig. 5. Delivery Ratio with/without Power Management

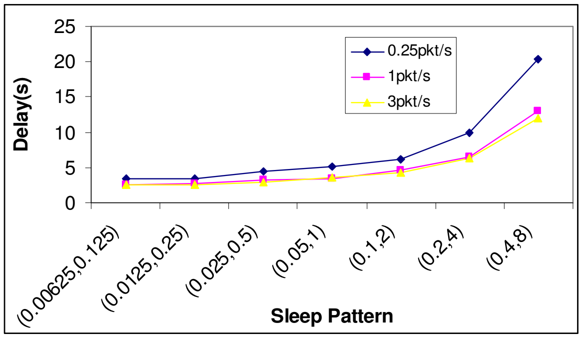


Fig. 6. Average Delay with Power Management

C. *Impact of different node speeds*

In order to evaluate the mobility impact on our power management scheme, we let the nodes move with different maximum speeds and repeat some of the earlier experiments by using the sleep pattern (0.025,0.5). As before, 10 CBR flows with random sources and destinations are used. The traffic generation rate of each flow is varied from 0.25 pkt/sec to 3 pkts/sec.

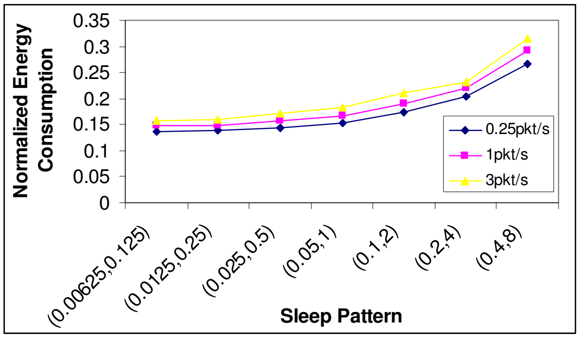


Fig 7: Normalized Energy Consumption with the CAPM scheme

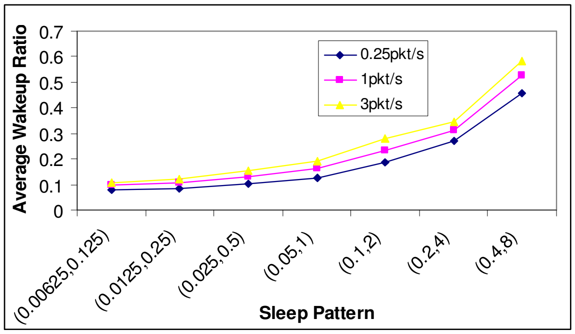


Fig 8: Effect Wakeup Ratio with the CAPM scheme

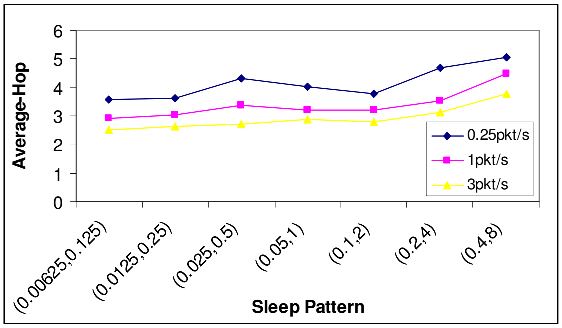
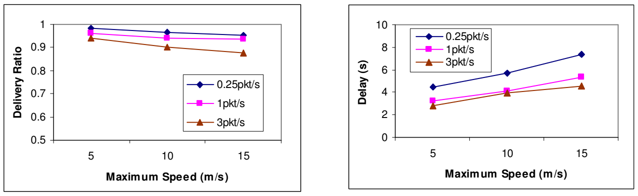
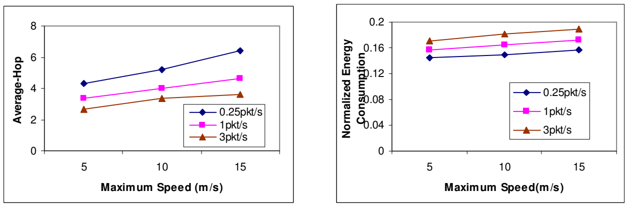


Figure 9: Average Delivery Hops with Power Management.

From Figure 10, we can see that as nodes move faster, the delivery ratio drops. The drop rate is higher with higher node speeds. In addition, we see that our power management scheme allows the system to maintain high delivery ratio with about 84% to 86% energy saving at low load (0.25 pkts/sec) and 81.5% to 84% energy saving at high load (3 pkts/sec). The average number of hops it takes to deliver messages increases slightly with increasing node speeds.



(a) Delivery Ratio (b) Delay



(c) Average-Hop (d) Energy Efficiency

Fig. 10. Impact of different node speeds.

V. CONCLUSION

In this paper, we present a context-aware power management (CAPM) scheme for DTNs. Our power management scheme allows us to maintain network connectivity while saving energy. Then, via simulations, we study the data delivery performance and energy saving that one can get when CAPM is used together with Prophet. Our results indicate that the CAPM scheme allows us to save energy while maintaining reasonably close data delivery performance to the case without power management. Then, we investigate how the performance changes with changing sleep pattern and traffic load. We also investigate how the data delivery performance changes with different node speeds. Our simulation results demonstrate that our CAPM scheme is adaptive to different network environments and can provide energy saving of up to 85%.

In this work, we only evaluate the impact of power management on the data delivery performance using only one DTN routing scheme. We intend to study the performance of the CAMP scheme with other DTN routing schemes. In addition, we intend to study the impact of different mobility models on the performance of the CAMP scheme with any DTN routing schemes. We also intend to investigate if the same power management scheme can be effective in DTNs with both static and mobile nodes.

Acknowledgements: This work has been supported by DARPA under Contract W15P7T-06-C-P430. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsor of this work.

REFERENCES

[1] Forrest Warthman, "Delay-Tolerant Networks (DTNs)", DTN Research Group Internet Draft, March 2003.

[2] Zhensheng Zhang, "Routing in Intermittently Connceted Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges", IEEE Communications Surveys & Tutorials, 1st Quarter 2006.

[3] S.Jain, K.Fall, and R. Patra, "Routing in a Delay Tolerant Network", SIGCOMM'04, Aug. 30-Sept. 3, 2004.

[4] D.K.Goldenberg, J.Lin, and A.S.Morse, "Towards mobility as a network control primitive", in Mobihoc, 2004.

[5] M.M.B.Tariq, M. Ammar, and E. Zegura, "Message Ferry Route Design for Sparse Ad hoc Networks with Mobile Nodes", ACM MobiHoc , May22-25, 2006.

[6] K. Fall, "A delay-tolerant network architecture for chanllenged internets", in SIGCOMM, 2003.

[7] V.Shnayder, M. Hempstead, B.chen, G.Werner, and M.Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", 2004 ACM.

[8] T.Armstrong, "Wake-up Based Power Management in Multi-hop Wireless Networks", Term Survey Paper, University of Toronto.

[9] A.Keshavarzian, H.Lee, and L.Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks", MobiHoc'06.

[10] Hyewon Jun, Mostafa H. Ammar, and Ellen W. Zegura, Power Management in Delay Tolerant Networks: A Framework and Knowledge-Based Mechanisms, In *IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, September, 2005.

[11] Hyewon Jun, Mostafa Ammar, Mark Corner, and Ellen Zegura, Hierarchical Power Management in Disruption Tolerant Networks with Traffic-Aware Optimization, In *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)* , September, 2006.

[12] J. Burgess, B. Gallagher, D. Jensen, and B.L.Levine, Maxprop: Routing for vehicle-based disruption-tolerant networks. In INFOCOM, 2006.

[13] A. Lindgren, A.Doria, and O.Scheln, Probabilistic Routing in Intermittently Connected Networks. In Proc. Workshop on Service Assurance with Partial and Intermittent Resources, August 2004.

[14] P.Juang et al. Engery-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. SIGOPS Oper. Syst. Rev., 36(5): 96-107, 2002.

[15] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research", Wireless Communications & Mobile Computing (WCMC), Vol .2, no. 5, pp. 483-502, 2002.

[16] R.Zheng, J.C.Hou, and L.Sha, Asynchronous Wakeup for Ad Hoc Networks, MobiHoc'03.

[17] L.M.Feeney and M.Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc network", Proceedings of IEEE INFOCOM, 2001.

[18] http://www.xbow.com/Products/productsdetails.aspx?sid=62

[19] "The network simulator ns-2", [Online] at http://www.isi.edu/nsnam/ns/.

[20] Yu. Tseng, C. Hsu, and T. Hsieh, Power-Saving Protocols for IEEE 802.11 –Based Multi-Hop Ad Hoc Networks. 2002 IEEE.

[21] B. Chen et al, "Span: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks", Proceedings of 7th ACM, Mobicom ,2001, pp 85-96.

[22] B. Awerbuch, et al, "The pulse protocol: energy efficient infrastructure access",Proceedings of IEEE Infocom, 2004.

[23] E. Shih, P. Bahl, M. J. Sincalir, „Wake on wireless: An event driven energy saving strategy for battery operated devices", Proceedings of 8th Mobicom, 2002.

[24] M. J. Miller, N. H. Vaidya, "Power save mechanisms for multihop wireless networks, Proceedings of 1st International Conference on Broadband Networks", 2004 pp 518-526.