International Conference on Wireless and Ubiquitous Systems
8-10 October 2010, Sousse, TUNISIA

EMP: A Network Management Protocol for IP-Based Wireless Sensor Networks

Shafique Ahmad Chaudhry
COINS Research Group, CCIS/IMAM
Al-Imam Muhammad bin Saud University
Riyadh, Saudi Arabia

George Boyle, Weiping Song, Cormac Sreenan
Mobile and Internet Systems Laboratory,
Department of Computer Science, University College Cork
Cork, Ireland

Abstract— **Wireless Sensor Networks (WSNs) have attracted significant research interest in recent years because of their suitability to a wide range of real world applications. The envisioned Internet Protocol (IP) support for WSNs requires interoperability with existing management solutions, like Simple Network Management Protocol (SNMP), in order to provide remote management functionality and assure the correct operation of the WSN. It is essential to provide a network management system that is interoperable with standard network management solutions, customizable, and extensible to various WSN applications. In this paper we present work in progress on our EmNetS Network Management Protocol (EMP), a lightweight and SNMP-compliant IP-based WSN (IP-WSN) management solution. We present detailed operational architecture and a Management Information Base (MIB) which is extensible to IP-WSN applications. We also present implementation details and preliminary results from our laboratory testbed.**

Index Terms: Sensor network management, Sensor network management protocol, SNMP, EMP, 6LoWPAN

I. INTRODUCTION

Wireless sensor networks (WSNs) have emerged as a catalyst technology, poised to change the traditional methods of data collection, monitoring and control. WSNs are known for their suitability for various environmental and industrial applications but the true potential of WSNs can truly be utilized by connecting them to IP-based networks where most of the existing information resources reside.

The integration of WSNs with IP networks has been stimulated by various factors. First, IP networks allow the use of existing infrastructure and available information resources. Secondly, IP based technologies, along with their diagnostics, management and commissioning tools, already exist, and are proven. Thirdly, IP based devices can more easily be connected to other IP networks, without the need for translation gateways etc. Internet Engineering Task Force (IETF)[1] is standardizing the transmission of IPv6 over IEEE 802.15.4[2] through a working group known as 6LoWPAN [3]. These IP-WSNs are considered a major technology for the realization of ubiquitous and pervasive environments.

To keep these networks always operational, robust and efficient network management architecture is needed. IP-based technologies, along with their diagnostics and management tools like Simple Network Management Protocol (SNMP) are available but such tools cannot be deployed directly on WSNs because of the resource limitations. It is, therefore, essential to have a network management system that is lightweight enough to run on WSNs and is yet interoperable with SNMP.

Most of the existing solutions cover either WSNs or IP network but does not consider the IP-based WSN like 6LoWPANs. The management of IP-WSN is different from just WSNs as well as just from IP networks. For example on one hand, 6LoWPANs are IPv6 networks; while on the other hand, these are low power sensor networks with extremely limited resources, which means we want IP-like solutions but lighter weight which can be deployed on IP-WSNs. Additionally, the traditional networks run a diversity of applications as compared to WSNs where the network is generally executing a single application in a cooperative fashion although certain efforts are being made to support multiple applications on WSNs. On the contrary, because of IP support, there is a possibility that LoWPANs support a variety of services making it further complicated for network management operations.

Considering a diversity of objectives and challenges, we present EmNetS Management Protocol (EMP) for IP-WSNs, which we have designed and developed. Not only EMP is lightweight, it also provides interoperability with SNMP making it feasible to monitor and manage the WSNs remotely through the Internet from anywhere. We have also designed a Management Information Base (MIB) which helps extensibility and adaptability for various applications.

The remainder of the paper is organized as follows. In section 2, we discuss the state-of-the art, followed by the system model in section 3. We describe detailed architecture of EMP and its implementation details in sections 4 and 5 respectively. We present the preliminary evaluation results in Section 6 and conclude the paper with a summary in section 7.

II. RELATED WORK

Simple Network Management Protocol (SNMP) is the standard network management framework for traditional IP networks however it cannot be deployed directly on the sensor networks because of various WSN characteristics: a) putting SNMP message overhead, over bandwidth resource constrained WSNs, is not practical, b) SNMP does not specifically address the problem of node-failure as common

phenomenon, which is common in WSNs, and c) SNMP requires huge Management Information Base (MIB) and sensor nodes generally cannot support such storage requirement.

Traditional network management protocols for ad-hoc networks, e.g. Ad-hoc network management protocol (ANMP) [7] and Guerilla [8] are also not without limitations for WSNs. ANMP is the extension of SNMP, therefore, inherits the limitations associated with SNMP. The Guerilla architecture provides an adaptive management for ad hoc networks with heterogeneous node capabilities with the assumption of the presence of some nodes with processing power more than the sensor nodes, which is not always true in the case of WSNs.

MANNA [4] architecture is the most pertinent work proposed for sensor networks which presents the technical basis to how management can be performed in WSNs. MANNA is a policy-based management framework which collects network management information from the MIB and then maps it into sensor network model. However, it presents the architecture for management of WSNs highlighting its inherent dependency on application for which it is being developed and does not consider the possibility of multiple applications running on the WSN.

Other architectures like BOSS [5] also focus on application specific scenarios. BOSS is a service discovery management architecture that serves as a mediator between Universal Plug-n-Play (UPnP) networks and sensor nodes. The scope of BOSS is very limited and the WSN management requirements demand more than just mediation between UPnp and WSN.

Sensor Network Management System SNMS [6] is an interactive network management system for WSNs. SNMS provides query based network health data collection and event logging but this approach requires a large key space and therefore high memory usage. Other main drawback of SNMS is that its network management functions are limited to passive monitoring only.

An implementation of SNMP over 6LoWPAN is presented in [9] where authors have used header compression and proxy forwarder to support SNMP over 6LoWPAN. This work does not consider the possibility of using the existing cache information to reduce management information collection request.

III. SYSTEM MODEL AND ARCHITECTURE

The 6LoWPAN entities can support star and mesh topologies and support both 16-bit short and IEEE-EUI64 bit extended address. The network entities can be classified as Gateway, Full Functional Devices (FFDs) and Reduced functional devices (RFDs). The main 6LoWPAN entities are shown in fig 1.

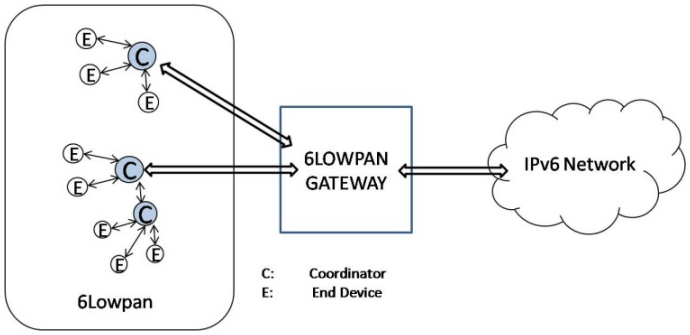

Fig. 1. 6LoWPAN entities

The Gateway is the implementation of 6LoWPAN adaptation layer functionality as specified in [10] and acts as an interface between an IPv6 network and IEEE802.15.4 based network. Multiple Personal Area Networks (PANs) can coexist under the gateway.

6LoWPAN devices are assumed to host and execute IP-stack, on top of the 14 PHY and 35 MAC primitives making them highly energy starved and are classified based on their resource set and constraints. An FFD which can be used as a PAN coordinator, coordinator or an end device can communicate with other FFDs as well as other RFDs. While acting as a coordinator FFD supports all defined primitives of IEEE 802.15.4. One of the coordinator acts as PAN coordinator for the specific PAN. An RFD implements only a partial set of primitive and acts only as the end device. RFD can only communicate with an FFD.

A. Network Discovery

In our approach management operations are carried out by the entities within the 6LoWPAN after the network discovery phase has been completed. We have distributed the tasks of network discovery, monitoring and management, across the 6LoWPAN, and the device discovery is performed by the coordinators. The main objective of task delegation is to reduce the communication cost. The bandwidth is the most valuable resource of WSNs, and it is known that cost associated with communication is usually more than that of sensing and processing. Here, we give an outline of our network discovery mechanism; the granular details are, however, out of scope of this paper.

In the network initialization phase, the coordinator populates the list of attached devices as well as the state table showing their status information. A coordinator is responsible for maintaining the state information of its all its subordinate devices down the hierarchy and reporting the status updates of their subordinates to their parent devices.

The coordinator filters and aggregates the received subordinate state information and sends this information to the upper level coordinator, in addition to its own state during the network initialization phase. The coordinator's address is added to the list of reporting devices on the parent coordinator and the reported state data is filled in the state table of the parent coordinator. Subsequently, the information travels up the hierarchy and reaches the gateway. During the normal operational phase, only changes in the subordinate states

instead of the whole subordinate state information are reported by the subordinate coordinators which results in reduction of communication overhead and increased lifetime of the network. This technique provides the network-wide snapshot of resources in the architecture. The management system collects state variables from network devices and processes them to perform control actions on the network in accordance with the management objectives.

Based on the collected management information various network models can be created and maintained including Network Topology map, Residual Energy map, Audit map, Network / Link throughput, Link Quality, etc. Based on these maps various other statistics can easily be generated / inferred using the collected information e.g. amount of sensed data, estimated network lifetime, number of transmissions, delivery latency, packet loss probability, data redundancy factor, etc.

EMP is independent of the specific type of routing protocol or operating system, and should be easily deployable on various platforms with minimal changes.

IV. EMNETS MANAGEMENT PROTOCOL (EMP)

Network discovery phase provides us with the network wide snapshot which is used to manage the resources. The network state can be obtained from the nodes using active probing or through periodic reports from the nodes. Active probing means that each device can responds to the coordinator's query when the coordinator is checking the state of the device. In this scheme nodes can be queried for management information anytime using the EMP *GetRequest* message. In case of periodic reporting each device sends its status to the coordinator periodically. The interval between the reports can be changed and tuned dynamically, providing a trade-off between network life and management information 'freshness'. The accuracy of the state data depends upon the reporting interval and the network latency. Therefore, the reporting interval should be adjusted to a value which achieves an acceptable level of accuracy. It is important to understand that different applications have diverse requirements, e.g, a network for an oil refinery plant monitoring needs more accuracy as compared to a building monitoring application.

All the 6LoWPAN coordinators report the status of all the subordinates to gateway up through their ancestor coordinators in the hierarchy.

To prevent excessive load on the sensor network, the gateway keeps a cache of network state. The information in the cache is valid only for *EmpObjExpiry* time, after which the information is considered as stale. As part of the protocol, the expiry times for each individual object can be set. This is achieved using two tables:

- empObjExpiryTable – a configurable table that matches each object type with its expiry time (in hundredths of a second); and
- empObjUpdateTable – a read-only table that, for each object instance, gives the amount of time (in hundredths of a second) since its cache value was last updated.

To check if the data for an object instance has expired, both the times are compared. If the last update time is greater than the expiry time, then the cache value has expired and it must be obtained again. An object that never expires will have no row in the empObjExpiryTable, whereas an object that should not be cached has an expiry time of 0 seconds.

A. SNMP Compliance

It is highly desirable that the queries, needed to monitor the states of devices within the WSNs, support a standard management protocol such as SNMP. But it is impractical to transport SNMP over WSNs because of the inherent bandwidth limitations in WSNs technologies like IEEE 802.15.4. In our proposal, as shown in fig. 2, SNMP is supported on the IP network side only whereas the EMP implementation on the WSN side provides interoperability with SNMP.

The management packets are translated to and from SNMP to the simplified EMP format on the EMP manager hosted on the gateway. Whenever an SNMP request arrives from a remote SNMP agent, the SNMP request is parsed and is translated to EMP query that contains object identifiers (OIDs) to be retrieved from the destination device's agent.

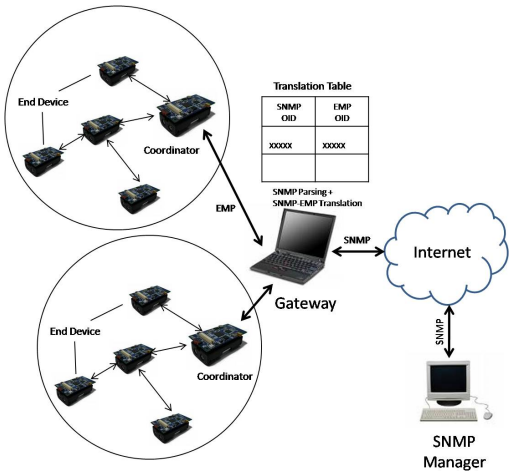

Fig. 2. SNMP-EMP interoperability

If the OID being requested for is a constant for the network, then the response is translated to SNMP response which is sent to the requesting SNMP agent. If the identifiers being requested are not constant but the information in the MIB cache is still valid for this OID, then the information is fetched from the cache, translated into SNMP format and response is sent back. On the other hand if the information in the cache is not fresh, an EMP query is sent to the device and the SNMP response is sent back to the requester after the EMP-SNMP translation. The process flow is shown in fig. 3.

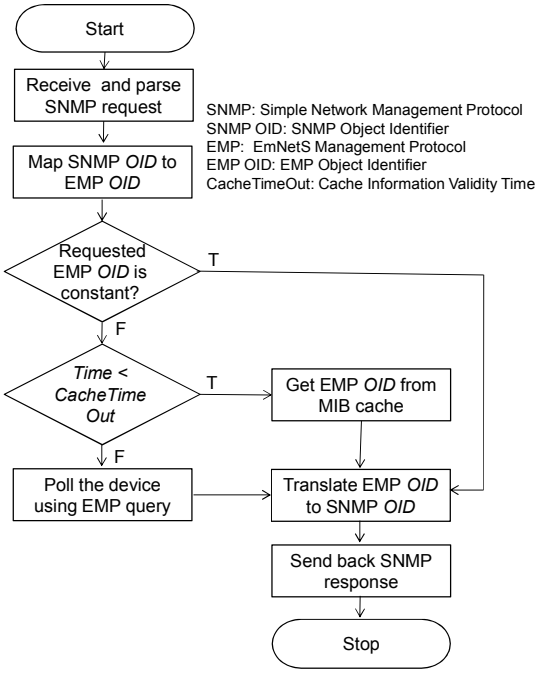

Fig. 3. Device monitoring procedure from SNMP Manager

B. Management Information Base

The provision of a simple yet customizable MIB for WSN is essential because resource limitations at WSNs make it impractical to support the complete SNMP Management Information Base (MIB)[11] on the sensor networks. We have designed a light-weight and simpler MIB for the WSNs which interoperates with SNMP.

Our MIB module for WSNs is divided into Network, Application and Mote groups. The Network group contains general and constant information about the sensor network. For Example in case of a 6LoWPAN network, the whole network is a single hop from the point of view of IPv6 routing, which means that many objects in the WSN are constant, obviating the need to poll the individual end nodes for these values.

The Application group provides statistics for, and allows configuration of the desired application; in our case it mainly deals with EMP statistics. However, when multiple applications run on the network, these parameters can be used and extended to provide application specific information.

The Mote group deals with information specific to WSN nodes, such as available radio frequencies and those in use by the Mote.

Additionally, the information bases for IEEE 802.15.4 PHY and MAC layers are already defined in the PAN Information Base (PIB) [11] and can be accessed locally. In order to access this information from outside of the WSN, OIDs need be assigned to these parameters. The provisioning of such OIDs means that this information can be shared with the SNMP manager outside the WSN.

V. IMPLEMENTATION

The EMP framework implementation follows the manager-agent model and is written in Java. The agent component has been implemented on blip [12] stack running on Tmote Sky [13] sensor nodes with MSP430 microcontroller, 10KB RAM, and 48KB flash memory. The nodes support IEEE 802.15.4 compliant CC2420 RF transceiver. The Base Station (BS) is connected to the gateway station through USB interface. The monitoring agent on the device supports access to EMP MIB and can respond to the EMP queries.

The management station runs on the gateway node, which also runs Net-SNMP [14] agent. The Net-SNMP agent (snmpd) communicates with EMP manager through its standard input and output. For the EMP manager, the java class EMPManager handles communication between IP and WSN networks with SNMP on one hand and EMPMessenger object on the WSN side. EMPMessenger class implements send GetRequest and SetRequest operations on the WSN (6LoWPAN) side.

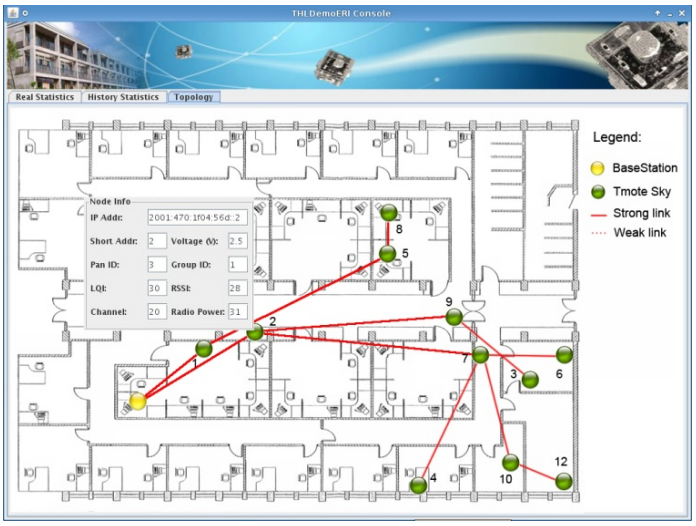

Fig. 4. Snapshot of WSN topology using EMP GUI

We have also developed an interactive graphical user interface (GUI) which provides user-friendly environment to view live network topology, monitor network statistics, and run management actions on the network. Fig. 4. is a snapshot of the deployed sensor network topology.

Users, by just navigating the topology map, can get the basic node information e.g. node ID, node's IP address, and sensed variables. We have demonstrated the EMP functionality, and remote management through its GUI with a WSN test bed for building management in [15]. Fig.5. shows the detailed information related to the sensor nodes under various categories.

Message logs are also maintained to provide historical data which can be used in analyzing the performed management operations. Run-time network probing is also supported to collect network management information and network statistics, e.g. nodes' energy levels, link maps, RSSI

information and so on.

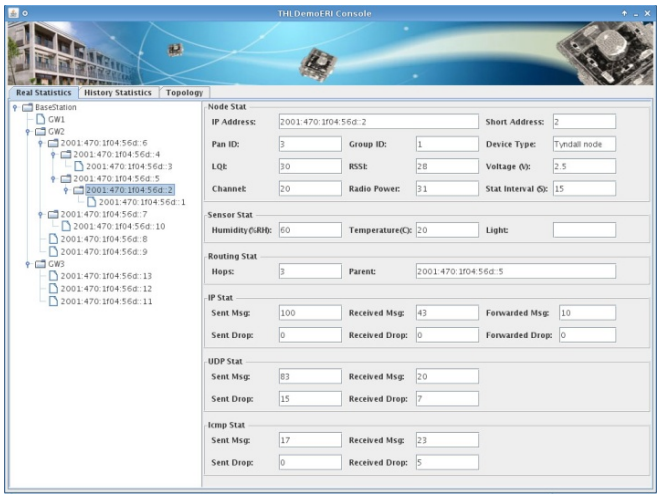

Fig.5. Snapshot of WSN device info using EMP

VI. PRILIMINARY RESULTS

We have deployed and evaluated EMP on our test bed of 11 nodes as shown in fig. 4, on the 3rd floor of Kane Building, University College Cork. We chose query-response latency, management traffic overhead, and reliability against different query-intervals to evaluate the performance of EMP. We evaluated EMP for 5 runs of 10 minutes each varying inter-query interval time and with running more than one applications.

A. Query-response Latency

Query-response latency is the time of getting the data from a WSN device against an EMP *GetRequest* and it is a good measure to check the response time from the node against a management request.

In our experiments we observed the query-response latency against different number of hops and in the presence of different applications running on the nodes. Fig. 6 shows the round-trip latency of receiving response data from the WSN nodes in the presence of other applications running on the WSN nodes. The number of hops on the X-Axis represents the number of hops from the gateway. This delay includes all kind of other associated delays i.e. the delay between the Gateway and the Coordinator and the delay between the Coordinator and the node. Similarly hop value 1 means that the device is one hop away from the Gateway. We observed that query response time is bounded and the addition of each hop adds about 38 ms as an additional delay.

In reality the network depth may or may not have an unbounded effect as the number of nodes becomes extremely large. But as a fact the deployment topology and communication model determine the management system performance. For example, a sensor network with a large number of nodes but smaller average path lengths (in hops) could experience less query delay as compared to a smaller network with longer paths (in hops). It means that the framework is scalable for different topologies. There may not be a proportionate effect on the performance metrics when seen in the context of an increase in network size.

However, in the presence of other applications running on the WSN, the query-response delay depends on how 'busy' remains the node because of the other application. As expected if the non-management application is sending data to the BS at a higher packet rate then the query-response time is very high. If the non-management application is sending more than 10 packets per second over the network then there was almost no reply for the queries from the nodes which are 3 or more hops away from the BS.

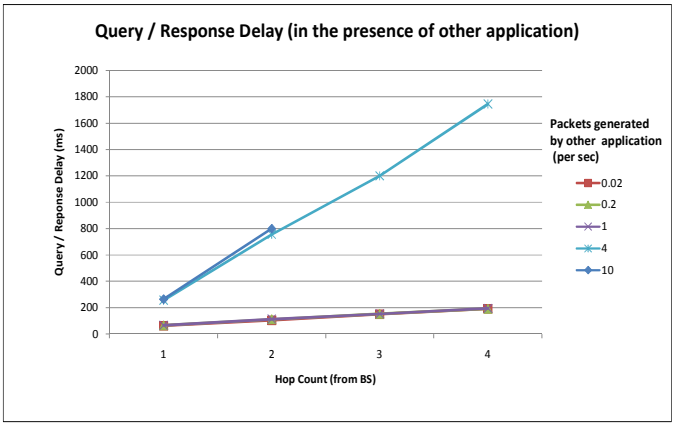

Fig. 6. Query Response Time with various applications

B. Management Traffic Overhead

The overhead generated by the EMP depends on the MIB-cache validity time. Fig. 7 shows the traffic overhead against various values of MIB-cache validity time. Longer cache validity time means less network polling, resulting into less network overhead messages and therefore longer network life. However, the cache information at the end of cache validity time may not be 100% accurate. The smaller cache-validity time assures that the information in the cache has most recently been updated, but this approach generates more traffic, adversely affecting the network life.

The accuracy of the information depends on the network operation and network dynamics. For example, in case of a static network where nodes are sensing and sending the data periodically at a slow rate, even the longer cache validity could give fairly high accuracy level. On the contrary, in the highly dynamic environments, even a smaller cache validity time may not be able to provide 100% accuracy.

Based on its information needs, a network manager can adjust the cache-validity to obtain an optimized combination of accuracy and network life. Devising an automatic strategy to adjust the cache validity remains a task for future research.

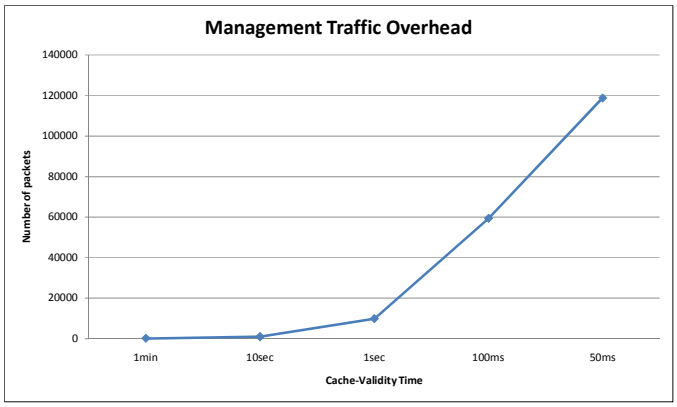

Fig.7. Management traffic overhead

C. Reliability

We define the reliability as the query-response ratio for the management queries made by the manager. Fig. 8 shows the query-response ratio with different value of inter-query delay while querying the data from the end device. We observed that inter-query delay of 1 sec or more gives around 97% success rate. However, the success rate drops considerably when multiple queries are sent every second over the network. This can be attributed to the IEEE 802.15.4 performance which drops to about 70% even in single hop networks if the transmission rate is raised to about 10 packets per second.

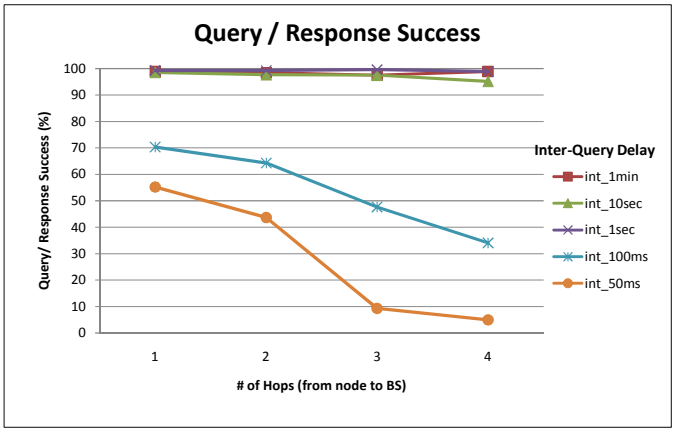

Fig. 8. Query-Response Success rate with different Inter-Query Delay and Number of Hops

VII. CONCLUSION

In this paper we present EMP, a light-weight and SNMP-interoperable network management framework for IP-WSNs. The operational architecture emphasizes reduction of communication cost in order to increase the network lifetime. The MIB defines the information that is needed to be managed on the devices for management purposes. The solution has been implemented on real test bed running blip and gives good results. Our future work focuses on the optimization of network life and aggregation mechanics of EMP traffic for IP-WSN networks. Another important aspect is to use inference mechanisms to reduce network queries.

ACKNOWLEDGEMENT

This work was partially supported by lownetQoS project grant # 300901 by the Deanship of Research, Al-Imam Muhammad bin Saud University, Saudi Arabia, and Embedded Networked Sensing (EmNetS) project funded by Enterprise Ireland's industry lead Wisen Program.

REFERENCES

[1] Internet Engineering Task Force. http://www.ietf.org/

[2] 802.15.4-2003, IEEE Standard., "Wireless medium access control and physical layer specifications for low-rate wireless personal area networks," May 2003.

[3] IPv6 over Low Power WPAN Working Group http://www.ietf.org/html-.charters/6LoWPAN-charter.html

[4] L.B. Ruiz, J.M.S. Norgueira, A.A.F. Loureiro, "MANNA: Management Architecture for Wireless Sensor Networks," IEEE Communications Magazine, Vol. 41, No.2, Feb. 2003.

[5] H. Song, D. Kim, K. Lee, and J. Sung, "UPnP-Based Sensor Network Management Architecture," ICMU '05, April 2005.

[6] Gilman Tolle and David Culler, "Design of an application cooperative management system for Wireless Sensor Network," EWSN'05, 2005

[7] W. Chen, N. Jain and S. Singh, ANMP: Ad hoc Network Management protocol, IEEE Journal on Selected Areas in Communications 17(8), August 1999, 1506-1531.

[8] C. Shen, C. Jaikaeo , C. Srisathapornphat , Z. Huang, "The Guerrilla Management Architecture for Ad hoc Networks," Milcom 2002.

[9] EmNetS Project Website http://www.cs.ucc.ie/emnets/

[10] Transmission of IPv6 Packets over IEEE 802.15.4 Networks RFC 4944, http://www.ietf.org/rfc/rfc4944.txt

[11] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1902, IETF, Jan. 1996.

[12] Berkley Implementation of 6LoWPAN http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip

[13] Tmote Sky datasheet http://www.moteiv.com/products/ docs/tmote-sky-datasheet.pdf

[14] SNMP suite implementation http://www.net-snmp.org/

[15] R. Spinar, P. Muthukumaran, R. de Paz., D. Pesch, W. Song, S.A. Chaudhry, C.J. Sreenan, E. Jafer, B. O'Flynn, J. O'Donnell, A. Costa, M. Keane, "Efficient building management with IP-based wireless sensor network. In: EWSN 2009, 6th European Conference on Wireless Sensor Networks. Cork, Ireland 11-13 February 2009.