Journal of Ubiquitous Systems and Pervasive Networks
Volume 2, No. 1 (2011) pp. 15-22

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

Shafique Ahmad Chaudhry [a*], Weiping Song [b], Muhammad Habeeb Vulla [a], Cormac Sreenan [b]

[a] *COINS Research Group, CCIS / IMAM, Al-Imam Muhammad bin Saud University, Riyadh, Saudi Arabia*
[b] *MISL, Computer Science Department, University College Cork, Cork, Ireland*

Abstract

Wireless Sensor Networks (WSNs) have attracted significant research interest in recent years because of their suitability to a vast 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 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 evaluation results from our laboratory testbed.

Keywords: *Sensor network management, Sensor network management protocol, SNMP, EMP, 6LoWPAN*

1. Introduction

Wireless Sensor Networks consist of large number of small, low power, and intelligent sensors, and are envisioned to change the way in which data will be collected from environment; giving a new paradigm to monitor and control the ambient environments. Sensor nodes are generally projected with small dimensions (cm^3 or mm^3) and this size limitation results into severe resource constraints such as limited battery power, low computational and memory resources, scarce wireless bandwidth, and limited communication capability. To keep these networks always operational, robust and efficient network management architecture is needed.

The uniqueness of the purported challenges in these networks makes management techniques of traditional networks readily impractical. For example, first the occurrence of a fault is a problem in networks but a 'feature' of WSNs. For large scale WSNs, faults occur frequently and components maintenance or energy recharge is not an option. In a few cases as reported in [15], configuration errors and even the environment interference can cause the loss of an entire WSN even before it starts to operate.

Second, unlike for traditional networks, wherein the primary goals are to minimize response time and provide detailed management information, sensor networks are designed with the primary goal of minimizing the energy usage [16]. A workable method to achieve this goal may be to carry out the management activity through minimum communication between the network elements for monitoring purposes.

Third, traditional networks are designed to run a large number of user applications. Therefore, the network components are installed, and configured with an objective to support a large number of different kinds of services. The WSNs are generally application-oriented. A network management system (NMS) designed for WSNs should provide a set of management functions that addresses such unprecedented network and behavioral features of WSNs.

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 exiting information resources reside.

The integration of WSNs with IP networks has been stimulated by various factors. First, IP networks allow the use of existing

* Corresponding Author: Phone: +966 12586728
Email: hazrat.shafique@gmail.com

DOI: 10.5383/JUSPN.02.01.002

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.

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. Table 1 shows the extent of limitations associated with sensor nodes available in the market, and justifies our assertion.

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 do 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. In this paper, considering a diversity of objectives and challenges, we present EmNetS Management Protocol (EMP), which we initially designed and developed under Enterprise Ireland funded EmNetS project [9] for managing 6LoWPANs. Not only EMP is lightweight, it also provides interoperability with SNMP making it feasible to monitor and manage the WSNs remotely through Internet from anywhere.

Table 1: Resource limitations with LoWPAN devices

Product	CPU	Radio Bandwidth	Memory	
			Flash	RAM
Chipcon2430	8-bit	250 Kbps	128 KB	8 KB
ATmega256RZBV	8-bit	250 Kbps	256 KB	8 KB
MicaZ Motes	8-bit	250 Kbps	128 KB	4 KB
Helicomm IP-link 1221-2034	8-bit	250 Kbps	64 KB	8 KB
TelosB motes (TPR2420)	16-bit	250 Kbps	48 KB	10 KB
Crossbow Technologies Imote2 (IPR 2400)	32-bit	250 Kbps	32MB	256MB

The main contributions of our work are: a) review of management goals, requirements, and design for 6LoWPAN, b) design of a network management protocol (EMP), c) design of management information base for 6LoWPAN networks and, d) provisioning of interoperability between EMP and SNMP.

The remainder of the paper is organized as follows. In section 2, we discuss network management of 6LoWPAN, followed by the related work section 3. The system model and EMP architecture are described in 4 and 5 respectively. Section 5 outlines the implementation details while evaluation details are given in 6. We conclude the paper in 7 by providing a summary and future works.

2. Network Management of 6LoWPAN

Network management refers to the process of managing, monitoring, and controlling the behavior of a network. A network management system (NMS) generally provides a set of management functions that integrate configuration, operation, administration, security, and maintenance of all elements and services of a network. A large number of management solutions exist for traditional networks but these solutions are not directly applicable to WSNs because of the unique operational and functional attributes of WSNs.

A sensor NMS should be able to perform a variety of management operations on the network elements based on the monitored data, e.g., controlling sampling frequency, switching node on/off (power management), controlling wireless bandwidth usage (traffic management), and performing network reconfiguration in order to recover from node and communication faults (fault management).

The general goals of network management for sensor networks need to establish a clear and direct relationship with the mission-oriented design of sensor networks. These goals are defined as follows:

Scalability: Sensor nodes are assumed to be deployed in large numbers, therefore, the management system should be able to handle large volumes of sensory data as well as high density of sensor nodes.

Limited Power consumption: Due to the limited energy resource with WSNs, the management operations should be lightweight on node-local resources in order to prolong its lifetime.

Memory and Processing Limitations: The sensor nodes are supposed to have limited memory and processing power. The management applications need to be aware of such constraints and may only impose minimal overhead on the low-powered nodes for the storage of management information and processing.

Limited Bandwidth consumption: The energy cost associated with communication is usually more than that of sensing and processing. Therefore the management applications should be designed with this consideration in mind.

Adaptability: The management system should be able to adjust to network dynamics and rapid changes in the network topology. The system should be able to gather the reported state of the network and the topology changes. It should also be able to handle node mobility, the addition of new nodes, and the failure of existing nodes.

Fault tolerance: Sensor nodes may run out of energy causing a fault in the network. Moreover the node may go to sleep mode to conserve energy or may be disconnected from the sink node because of network partitioning. The management system should be aware of such dynamics and it should adjust accordingly.

2.1. Design Goals for 6LoWPANs

The integration of IP with LoWPANs brings in several other objectives that need to be addressed while designing a management system for 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. It is, therefore, essential to have a network management system that is lightweight enough to run on WSNs and is yet interoperable with SNMP.

Table 2: 6LoWPAN considerations and requirements

6LoWPAN Considerations	*Requirements for 6LoWPAN Management System*
6LoWPANs exhibit user heterogeneity	Queries should be supported across user domains
Communication is across networks	Network management framework must cognize and act to network and channel behaviors
Network elements are many and heterogeneous	Elements of 6LoWPAN NMS must be distributed across networks, optimally
Syntax and semantics vary across networks	Translators and proxies should be embedded in 6LoWPAN NMS, wherever necessary
Querying types and scopes vary across networks	Consistent query types and specific Management Information Base (MIB) must be defined

These distinctive characteristics of 6LoWPAN form considerations for its management system, which pose specific requirements as shown in Table 2. Each of these requirements has a direct impact on the management system design.
Based on the requirements in table 2, we have identified the following goals that need to be accomplished while designing a management framework for 6LoWPAN.

Interoperability: The management system for 6LoWPANs must provide backward compatibility with legacy management protocol in IP, such as SNMP and its variants. The following actions must be taken into consideration to achieve this goal.

- Network elements that can implement SNMP readily must be identified.
- Network elements that cannot implement SNMP should connect to the management system through an SNMP parser and a proxy.

Minimized Communication Cost: The management system for 6LoWPANs must pose least on the communication of the 6LoWPAN. In order to meet this goal:

- A resource discovery mechanism must exist to circumvent futile communication with the unavailable nodes.
- A mechanism must exist that classifies network elements as available, alive, sleeping or expired.
- A fragmentation mechanism must exist to allow a large sized management packet to be split into the least number of fragments.

Placement of Managers and Agents: The management system must place managers and agents onto the network elements optimally. Therefore, a mechanism should exist that assigns manager and managed roles to appropriate network elements.

MIB Utilization: The management system must distribute and utilize MIB to ensure information availability. To accomplish this goal the followings must be provided:

- A mechanism that determines the constituents of MIB for all the network elements.
- A mechanism that ensures correctness and availability of MIB.
- A mechanism that provides resilience to network element failures,
- A mechanism that distributes new MIB definitions or incremental information with relevant network elements.

3. 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 usual 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 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.
There a few innovative patents proposals [18],[19] for WSN management. The patent in [18] describes the management equipment in general but does not give technical details especially for 6LoWPAN which we have established are different than WSNs in terms of management tasks. The work in [19] talks about managing NON-IP WSNs using SNMP and a gateway. The main problem of this work is lack of end-to-end reliability which is a main feature of IP-Based WSNs.
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.

4. Network Model and Architecture

In this section we present the basic network architecture of 6LoWPAN on which EMP is running. The 6LoWPAN entities can support star and mesh topologies and support both 16-bit short and IEE-EUI64 bit extended address. Followings are the entities that comprise a 6LoWPAN network.

Gateway: The detailed implementation of adaptation layer functionality [10] is implemented through a 6LoWPAN gateway that sits between IPv6 and LoWPAN networks. It is an unconstrained device under which multiple PANs can coexist.

6LoWPAN Devices: 6LoWPAN devices are in great contrast to their wired counterparts in size, computation, and energy resources. These devices host and execute IP-stack, on top of the 14 PHY and 35 MAC primitives making them highly energy starved. 6LoWPAN devices can be categorized into Full Functional devices (FFDs) and reduced functional devices (RFDs).

- A FFD can communicate with reduced function devices (RFDs) and other FFDs, and operate in three modes serving either as a PAN coordinator, coordinator, or an end-device. The coordinator is a device with capability of 6LoWPAN adaptation layer (layer 2.5) routing i.e. routing packets to the next hop device in 6LoWPAN. One of the coordinator serves as the PAN coordinator and is the primary controller of the PAN. The coordinators may also co-exist in a hierarchical manner. While working as a coordinator, a FFD supports all 49 primitives as defined in 802.15.4.

- RFD is intended to run extremely simple applications and supports only 38 primitives at the minimum level of configuration. It can communicate only to an FFD, therefore, its role is limited to only as an end device.

The 6LoWPAN entities are shown in figure. 1.

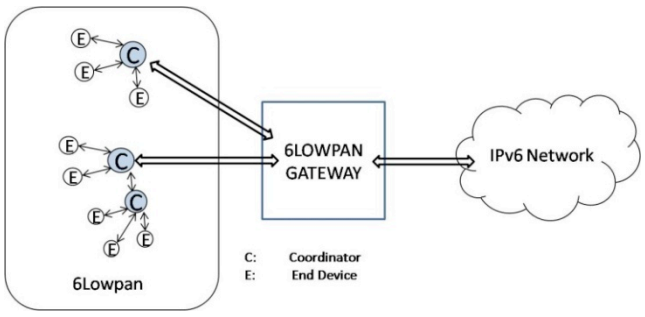

Figure. 1. 6LoWPAN entities

4.1 Network Discovery

The network management operations can be executed only after network devices have been discovered, and same is the case in our proposal. 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 and provide scalability. This task distribution helps saving the bandwidth, which is the most valuable resource in WSNs. It is known that cost associated with communication is usually more than that of sensing and processing, therefore, we opt for a tradeoff between communication cost and processing cost. In this subsection we provide 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 with 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. The drawback, however, is that if a network has a deep hierarchy then the coordinators may use a huge portion of their available resources for network management activities. However, in indoor environments we can afford to have some 'resource-rich' nodes to act as coordinators. In the outdoor environments it shall be important to distribute the network depth uniformly across all the coordinators to increase the network life.

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

5. EmNets Management Protocol

Network discovery phase provides us with the network wide snapshot which is used to manage the network 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.

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 user-defined *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.

5.1. 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 solution framework, as shown in figure 2, SNMP is supported on the IP network side only whereas the EMP implementation on the WSN side provides interoperability with SNMP. SNMP support means that the WSN can be accessed and monitoring from anywhere using a standard SNMP manger. Our web-based interactive interface enables the user to view and monitor the network.

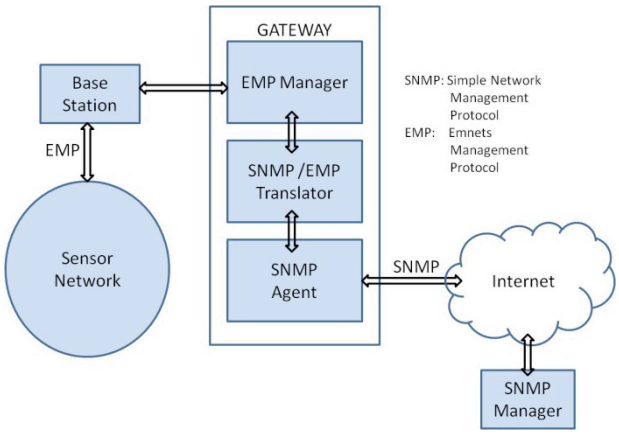

Figure 2. EMP operational architecture

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.

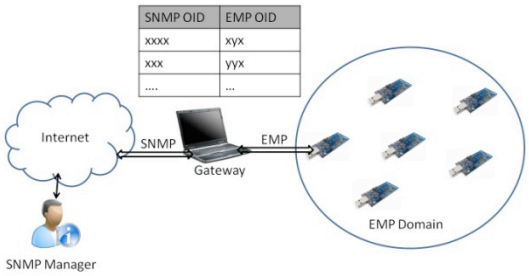

Figure 3. SNMP-EMP interoperability

When an SNMP query arrives at the gateway, following scenarios are possible:

- If the OID being requested for is a constant for the network, e.g, network subnet mask, then the reply 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 to SNMP manager.
- If the information in the cache is not fresh, which can be checked through the *empObjExpiryTable,* an EMP query is sent to the device and the SNMP response is sent back to the requester after the EMP-SNMP translation. At the same time the MIB entry for this object is updated in the *empObjUpdateTable*. The detailed process flow is shown in figure 4.

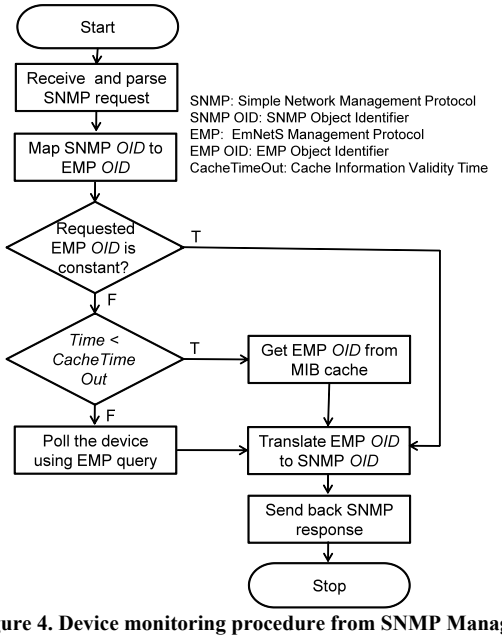

Figure 4. Device monitoring procedure from SNMP Manager

5.2. 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. Not only this MIB provides basic information, it is also extendable because it has been designed using standard MIB definition SMIv2[17].

Our MIB module for WSNs is divided into Network, EMP (the management 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 6LoWPAN 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 EMP group provides statistics for, and allows configuration of the management application, in our case it mainly deals with EMP statistics. However, in case 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 the radio frequencies available and those that are in use by the mote or the software capabilities of a node. A section of our MIB attributes is shown in figure 5.

The MIB is extendable and new sensor new modules can be added to the MIB as follows.

- Any new modules that contain variables which may potentially be stored on a node are to be attached as a sub tree of the empMIB tree.
  - Network-wide variables can be stored as scalar sub identifiers of the module. Node-level variables are stored in a MIB table with the node identifier as one of the indices.

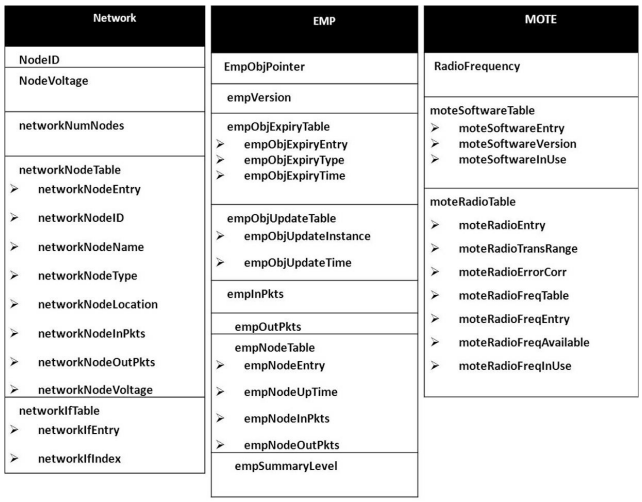

Figure 5. A section of EMP MIB

Additionally, the information bases for IEEE 802.15.4 PHY and MAC layers are already defined in 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.

For implementation purposes, within the MIB, each node in the sensor network is required to have a unique and persistent numerical identifier. Information applying to the network as a whole is represented in the MIB as scalar objects. Variables for individual nodes are stored in MIB tables, with the node identifier as an index. Data applying to a node that is more complex than simple scalar values can be represented with the node identifier acting as one index on a table with multiple indices. An example of a two-index table is shown in Table 3.

Table 3:A table with multiple indices

NodeID (Idx)	InterfaceID (Idx)	Type	...
1	0	Serial(USB)	...
1	1	Radio	...
2	0	Radio	...
⋮	⋮	⋮	⋮
35	1	Radio	...

6. 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, 10k RAM, and 48k Flash. The nodes support IEEE 802.15.4 compliant CC2420 RF transceiver. The 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.

Figure 6. Snapshot of WSN topology using EMP GUI

We have also developed an interactive graphical user interface which provides user-friendly environment to view live network topology, monitor network statistics, and run management actions on the network. Figure 6 is a snapshot of the deployed sensor network topology.

7. Evaluation

We have deployed and evaluated EMP on our test bed of 11 nodes as shown in figure 7, in our laboratory facility. There are various parameter which can be considers as performance metrics. We chose query-response latency, management traffic overhead, and reliability against different query-intervals to

evaluate the performance of EMP and to represent the accomplishment of goals described in table 2.

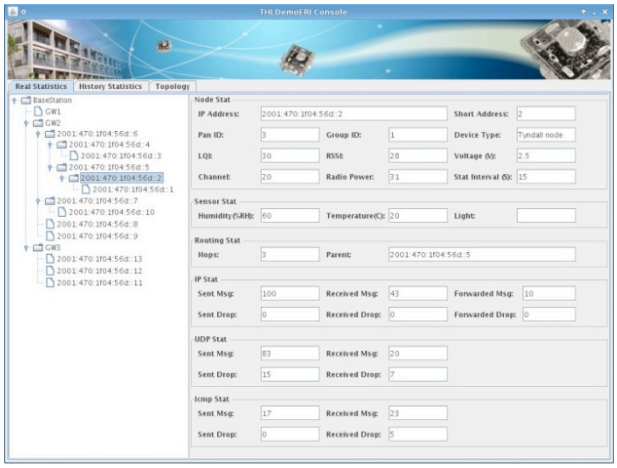

Figure7. Snapshot of WSN device info using EMP

7.1 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. This is the time between the request is sent to the node and the reception of response from the node.

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. Figure 8 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. This delay would also depend on the task scheduling between applications and node's sleeping schedule and duty-cycles. The 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. However, if a sleeping schedule is applied to all the nodes in the network than this delay may vary but eventually it shall still be bounded reflecting the data relay schedule of the network nodes.

In reality the network depth may or may not have a linear delay effect especially of the network is extremely large. We assert that the deployment topology and communication model play an integral role in determining 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 applications. 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.

7.2. Management Traffic Overhead

The overhead generated by the EMP depends on the MIB-cache validity time. Figure 9 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.

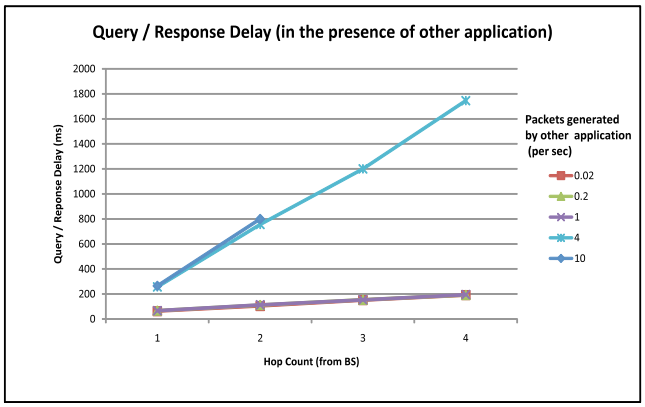

Figure 8. Query Response Time with various applications

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.

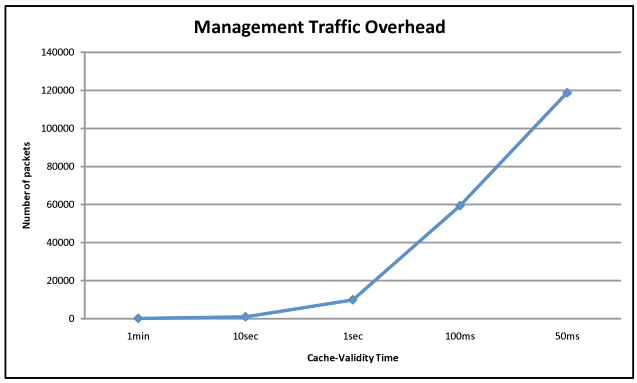

Figure9. Management traffic overhead

7.3. Reliability

We define the reliability as the query-response ratio for the management queries made by the manager. Figure 10 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 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 if transmission rate is raised to about 10 packets per second.

Based on our results we can easily assert that any network management system working over IEEE802.15.4 shall be working 'reliably' if the query rate is lower than 1 query per second. This is because of the inherent characteristics of IEEE 802.15.4 and there is not much a management solution could do to improve this.

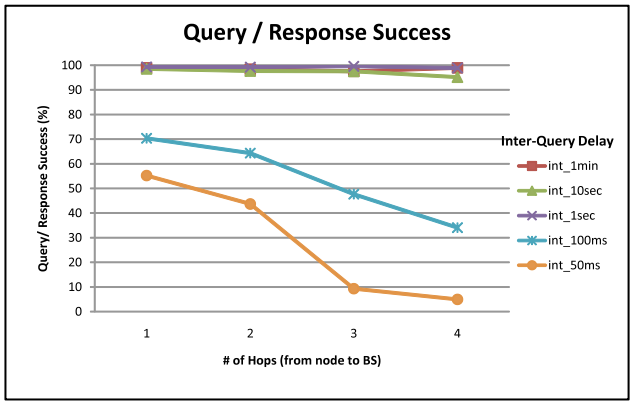

Figure 10. Query-Response Success rate with different Inter-Query Delay and Number of Hops

3. Conclusion and future work

In this paper we present EMP, a light-weight and SNMP-interoperable network management framework for 6LoWPAN. 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. Though EMP provides a basic management tool, there are various open challenges that need to be addressed and remain the focus of our future works. For example, it is important to investigate the deployment of autonomic management model over the 6LoWPAN. The granular details for such solution would enable robust 6LoWPAN operations with self-management capabilities. Additionally, the aggregation for management data and specifically sharing this data between various applications is an area that has not been really explored for 6LoWPAN.

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. http.dx.doi.org/10.1109/49.780355
[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] L.B. Ruiz, F.B. Silva, T.R.M. Braga, J.M.S. Nogueira, and A.A.F. Loureiro, "On Impact of Management in Wireless Sensor Networks," in Proc. IEEE/IFIP NOMS, Apr. 2004.
[16] M. Welsh and G. Mainland, "Programming Sensor Networks Using Abstract Regions," in Proc. USENIX NSDI Conf., Mar. 2004.
[17] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)" RFC 2578, April 1999.
[18] G. Bullard, D. Brake, C. Pruetting, R.Stitis, J. Ryberg, J. Thomas, D. Winchil, "Sensor Network Management," US Patent, 2010. http://www.freepatentsonline.com/y2010/0315207.html
[19] E. Kim, J.Lee, Y. Kim, H. Kim, "Method of Managing Non-IP Based Sensor Network Using Simple Network Management Protocol," US patent, US20100153551, 2010. http://www.freepatentsonline.com/y2010/0153551.html