Leveraging CoAP towards monitoring agriculture sensors network

A. Paventhan, Sai Krishna Allu, V. Gayathri, Sameer Barve and N. Mohan Ram
ERNET India R&D Centre, (An autonomous scientific society under the
Ministry of Communications & Information Technology, Government of India)
397, 13th Cross, Sadhashiva Nagar, Bangalore - 560080, INDIA
Email: mohan@eis.ernet.in, paventhan@eis.ernet.in

***Abstract*—Wireless Sensor Networks are typically made of resource constrained devices that are low-cost, low-power, low bit-rate supporting short-range communications. Towards realizing the broader vision of Internet of Things (IoT), various standards & protocols are emerging providing end-to-end internet connectivity between the resource constrained devices. IEEE 802.15.4 LoWPAN defines physical & MAC layer standards while the IETF 6LoWPAN specifies network layer adaptation of IPv6 protocol to the LoWPAN networks. The Constrained Application Protocol (CoAP) is emerging as the open application layer protocol designed to support resource constrained machine-to-machine (M2M) application environment. CoAP supports basic HTTP methods enabling easy integration with the existing Web. In this paper, we present an approach to using CoAP for agriculture monitoring. We discuss the application requirements, soil sensor integration to Wireless Sensor Networks and how CoAP can be utilized for realtime soil property monitoring over the Web.**

I. Introduction

In recent years, Wireless Sensor Networks has captivated the attention of researchers considering its potential application in wide variety of areas to observing the real world phenomena. Sensor networks comprise of collection of devices individually observing a phenomena in close range and collectively, they can observe properties of a larger area by employing appropriate communication topology. The potential gains of the collective effort of numerous sensor devices relies in how the event features are reliably communicated to the monitoring node. There are many challenges in designing application, transport, network and physical layer communication protocols for providing end-to-end internet connectivity between these resource constrained devices.

The idea of *Internet of Things (IoT)* has the potential to enable seamless integration of devices (*things*) into the Internet infrastructure. US National Intelligence Council has identified “Internet of Things” as one of the six most disruptive technologies with the potential to impact on US national interests leading to 2025 [1]. European Union has identified specific action plans for the promotion of Internet of Things [2] that includes innovation through pilot projects, research & development and standardization activities. The industry estimate predicts that there would be 50 billion devices connected to the Internet by 2020 [3] with number of devices online will be more than that of people [4]. Current networking technologies are optimized for human-to-human interactions rather than machine-to-machine (M2M) communications. There is an important need to support end-to-end communications between the resource constrained devices by integrating them to the existing Internet backbone infrastructure.

The widely used IEEE 802.15.4 LoWPAN [5] for the PHY/MAC layer and IETF 6LoWPAN [6], [7] for the Network layer adaptation enables resource constrained devices to talk IPv6 at the network layer & UDP at the transport. But, the right application layer protocol suitable for the resource constrained environment that take cognizance of the underlying stack is equally important. IETF Constrained Application Protocol (CoAP) [8] designed specifically for machine-to-machine application scenario, is emerging as an open application layer specification for constrained nodes. CoAP complies to REST (Representational State Transfer) architecture supporting HTTP mapping and Web integration. The major features of CoAP include constrained web protocol for M2M applications, asynchronous communications, resource identification using URIs, resource discovery, HTTP to CoAP & CoAP to HTTP conversions.

In this paper, we show how CoAP can be utilized for the following:

- Model agriculture sensor properties as a resource.
- Sensor resource management using RESTful approach using HTTP methods like GET, PUT, POST and DELETE
- Monitor / manage agriculture sensor network using traditional Web browser.

The agriculture sensor network utilized in our project comprises of wireless sensor nodes interfaced to soil sensors supporting measurement of electrical conductivity (EC), soil temperature and volumetric water content of soil.

The remainder of this paper is organized as follows: Section II presents the protocols relevant to IP-based WSN, and detailed discussion on the CoAP protocol and its relevance to WSN monitoring. Section III provides the details of agricultural application requirements and how the CoAP protocol is being used for soil sensor network monitoring. CoAP-based monitoring approach is compared with SNMP-based approach in section IV. Finally, section V presents the conclusion and future work.

	802.15.4 LoWPAN	IPv6 network
Minimum MTU	127 bytes	1280 bytes
Topology	Mesh or Star	Broadcast
Node address & Autoconf	16 bit short or EUI-64 long	128 bits DHCPv6 and stateless address autoconfiguration
Throughput	Max 250 kbps	10/100/1000 Mbps and above (802.3 LANs) 11/54/600 Mbps for 802.11b/g/n WLAN 384 Kbps (UMTS-WCDMA) / 14.4 Mbps (UMTS-HSDPA)
Power	Battery operated, Low power requirements (10 to 30 mA during Tx/Rx, negligible in sleep mode)	High-end CPU, more power requirements
Communication distance	10 meters; 100 meters in line of sight	$>$ 100 meters
Routing	Link layer multi-hop mesh routing (No direct reachability assumed)	Complex, intra-/inter-domain routing, supporting multiple paths
Memory	Few Kbytes	O(MBytes) for Tx/Rx buffers, routing table etc.

TABLE I
COMPARISON OF 802.15.4 NETWORK WITH IPV6

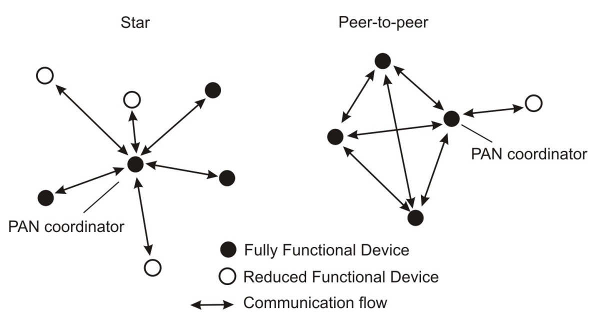


Fig. 1. IEEE 802.15.4 LoWPAN topologies [5]

II. BACKGROUND: PROTOCOLS FOR IP-BASED WIRELESS SENSOR NETWORK

The potential of Wireless Sensor Networks monitoring can enable important new class of applications such as structural monitoring, office/home automation, automatic weather observation systems, patient health monitoring, agricultural field measurements and automated early warnings to an impending disaster. Sensor networks comprise of collection of nodes individually observing a phenomena in close range and collectively, they can observe properties of a larger area by employing appropriate communication topology. There are many challenges in designing application, transport, network and physical layer communication protocols for the monitoring of wireless sensor networks. We will discuss in sections below some of the standards and protocols relevant to our work.

A. IEEE 802.15.4 LoWPAN

IEEE 802.15.4 standard [5] defines Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (WPANs). The standard uses carrier sense multiple access with collision avoidance (CSMA-CA) medium access mechanism and supports star as well as peer-to-peer topologies as shown in Figure 1.

The PHY layer is responsible for

- Activation and deactivation of transceiver
- Energy detection
- Link quality indicator for measuring the quality of received packet
- Channel frequency selection
- Data transmission and reception

IEEE 802.15.4 uses two frequency bands: 868-915 MHz with 40 kbps data rate using Binary Phase Shift Keying (BPSK) modulation technique and 2.4 GHz with 250 kbps data rate using Offset Quadrature Phase Shift Keying (O-QPSK).

The MAC sub layer is responsible for

- Generating network beacons if the node is a coordinator and synchronizing
- PAN association and disassociation
- Device security
- CSMA-CA for channel access
- Link reliability between peer MACs

B. IPv6 over 802.15.4 (6LoWPAN)

The IETF 6LoWPAN working group has defined specifications [6], [7] to efficiently transport IPv6 datagrams over IEEE 802.15.4 links. The Internet Protocol (IP) is predominantly used over Ethernet links that offer increasingly high throughput. The transmission of IPv6 packet over LoWPAN links are faced with several challenges due to the resource constraints as shown by the comparison in Table I.

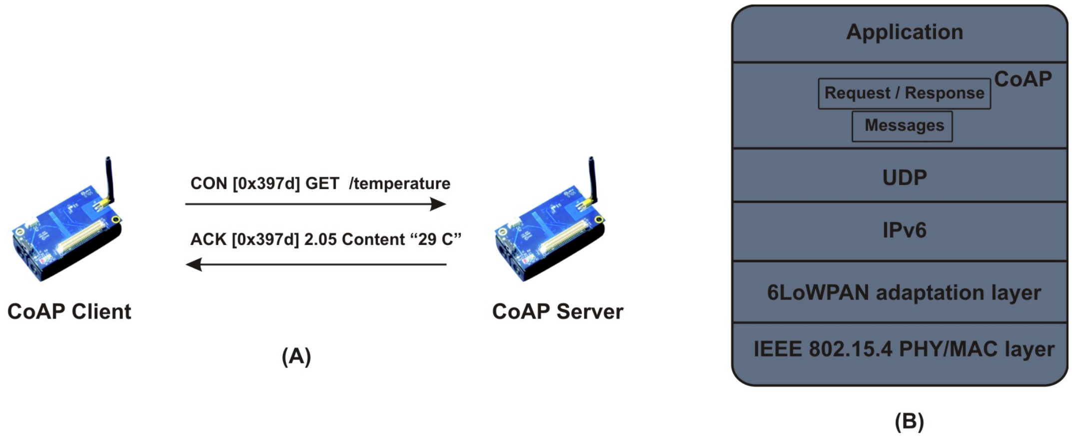


Fig. 2. (A) Reliable CoAP message exchanges (B) Typical CoAP application protocol stack

But, the benefits in enabling IPv6 over 802.15.4 links include:

1) Large IPv6 address space and stateless autoconfiguration
2) Easy to monitor/manage the network
3) Reusability of application layer protocols
4) Seamless and end-to-end integration with internet
5) Programmability using of socket APIs

IETF RFC 6282 [7] defines an adaptation layer considering that the minimum IPv6 MTU (1280 bytes) is much larger than the largest 802.15.4 frame size (127 bytes). The major functions of the 6LoWPAN adaptation layer that works between IPv6 and 802.15.4 MAC layer are listed below.

Fragmentation: Considering IPv6 MTU size (1280 bytes) and 802.15.4 frame size (127 bytes), the fragmentation and reassembly is essential at the 6LoWPAN adaption layer. The fragmentation header includes IPv6 datagram size, datagram offset and a datagram tag to help in reassembly.

Header compression: The header compression is important to increase the effective pay load of the upper layers. For example, 40 bytes of IPv6 header is compressed to 3 bytes as the source-, destination- address and datagram size can be inferred from layer 2 header.

Layer 2 forwarding: In mesh routing, the source and the final destination need not be directly connected. The mesh header carries the source and final destination link layer addresses and hop count. The intermediate nodes forwards the packet to the next hop after deducting the hop count. The Reduced Function Device (RFD) rely on the Full Function Device (FFD) for datagram forwarding.

C. Constrained Application Protocol (CoAP)

IETF Constrained RESTful Environment (core) [9] working group is defining framework for application development in resource constrained networks such as 6LoWPAN. The resource constrained nodes are characterized by low data rate, low power consumption, low cost and short-range operations.

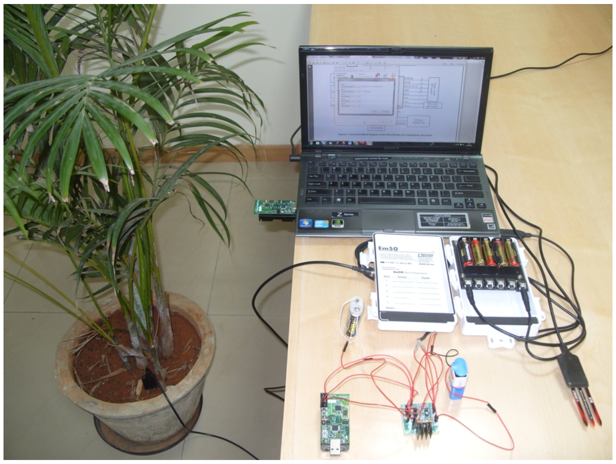
Figure 2 shows the reliable message exchanges between CoAP client & server, and typical CoAP protocol stack used for application development. The CoAP layer handles two major functions: (1) Asynchronous UDP message interactions (2) Request/response semantics. CoAP messages are short and they are carried over UDP. They are made of 4-byte fixed size header, followed by message options if any, and payload. The CoAP header fields include: a) ‘Type’ that takes one of the 4 values: i) 0 for [CON] Confirmable reliable transmissions ii) 1 for [NON] Non-confirmable / unacknowledged requests iii) 2 for [ACK] Acknowledgements and iv) 3 for [RST] Reset. b) ‘Code’ indicating client *request* method (`GET`, `PUT`, `POST` or `DELETE`) or server *response* code such as `2.05 Content`, `4.04 Not found`, `5.03 Service unavailable` etc. similar to HTTP. c) ‘Message ID’ for identifying duplicate messages and matching messages to acknowledgments.

CoAP resources are organized hierarchically and they are identified by Uniform Resource Identifiers (URIs) using a scheme similar to HTTP. A coap URI is of the form

`coap://coap-server[:port]/resource-path/`

The `coap-server` can be domain name or a valid IPv4/IPv6 address. Analogous to `'https'`, `'coaps'` can be used for secured service. The UDP `port` for the CoAP service is optional with the default port for CoAP being 5683. CoAP supports `/.well-known/core` resource path to help in resource discovery or to obtain any other metadata about the CoAP server. The CoAP request methods that act on the resources are `GET`, `POST`, `PUT` and `DELETE`. `GET` fetches the current state of the resource identified by the requested URI. When the `Observe` option is carried with the `GET` request, any change in the resource state is asynchronously notified to the client. `POST` creates a new resource or updates the existing one, depending on the target resource and the requesting server. `PUT` will update, or create if it does not exist, a resource identified by the URI. `DELETE` will remove the identified resource on the server.

CoAP adheres to the REST architecture [11] style and supports limited HTTP functionality. HTTP clients’ access to resources managed by CoAP server are supported through intermediary proxy operations. Similarly, CoAP clients’ access to resources managed by HTTP servers are possible.



(a) Soil Sensor integration with TelosB motes

(b) Soil Science Division - IIHR

Fig. 3. Agriculture Sensor Network Monitoring

Major goals of the CoAP protocol are discussed in detail in [10].

III. Agriculture Usecase

A. Application Requirements

ERNET India is collaborating with agriculture scientists from Indian Institute of Horticulture Research (IIHR) (under Indian Council of Agricultural Research). Based our interactions with domain scientists, it was understood that the following parameters will be helpful to farmers: (1) Soil properties (moisture, electrical conductivity, temperature) at different depths (2) Soil nutrients (Nitrogen-N, Phosphorous-P, Potassium-K) (3) Spectral reflectance measurements for plant nutrients (4) Measurement of leaf wetness etc. Our initial tests are being carried out at IIHR that is already enabled with connectivity ERNET India's IPv6 backbone and subsequently, in consultation with IIHR, the agriculture sensors network will be deployed at a suitable agriculture farm land for remote monitoring.

B. Agriculture Sensor Network Setup

1) Hardware: ERNET India's agriculture sensor network for the development of management/monitoring framework consists of heterogeneous hardware platforms: TelosB and IRIS motes from Memsic and AVR RAVEN development kit from Atmel Corporation.

Considering the project requirements and the availability of various agriculture sensors, currently we are using the 5TE soil sensor [13] from Decagon device. This soil sensor is interfaced to work with TelosB for our application demonstration. 5TE is an integrated sensor (Figure 3a) that can measure 3 different soil parameters - electrical conductivity (EC), soil temperature and volumetric water content. Like all ECH2O sensors, 5TE determines volumetric water content by measuring the dielectric constant of the media using capacitance/frequency domain technology. The sensor uses a 70 MHz frequency, which minimizes salinity and textural effects, making the 5TE accurate in most soils. The 5TE measures temperature with an onboard thermistor, and electrical conductivity using a stainless steel electrode array. The sensor connects through a 3 wire cable with a stereo connector or bare wire interface. The three connections are Excitation, Ground and Serial Out (data).The excitation is from 3.6 to 15 volts. Current drain during the water content measurement (approx. 10ms duration) can be as high as 45mA. The sensor outputs data at 1200 baud rate in asynchronous mode with 8 data bits, 1 stop bit and no parity bit. The output from the sensor is in the form of raw data in TTL format as given below:

```
56 432 645<0D>zG<0D><0A>
```

The raw data requires to be converted with suitable formula to arrive at the actual soil parameters. Figure 3a shows the TelosB mote connected to the Soil Sensor in our lab environment for development & testing. As shown in the figure, the soil sensor can also be connected to Em50 data logger for outdoor operation. Em50 has 5 sensor port and it can support storing an hourly measurement of data up to 4 years.

2) Software: Contiki μIPv6 & Erbium CoAP Stack: Contiki 2.6 embedded OS release is used with TelosB motes in our project. Contiki [14] is an open source, highly portable, multi-tasking operating system for the resource constrained wireless sensor networks. The 6LoWPAN μIPv6 [15] stack in Contiki provides highly memory efficient implementation of IP, UDP, TCP and ICMP protocols. The Contiki OS kernel is event driven and it is completely written in C programming language, and supports dynamic runtime linking of application programs. Contiki also implements ContikiRPL, an RPL implementation of IPv6 Routing Protocol for Low-Power and Lossy Networks as specified in IETF RFC 6550 [16]. RPL support in OS, can ensure IP-stack interoperability of two heterogeneous sensor network deployments. For example, the IPv6 layers, BLIP in TinyOS and μIPv6 in Contiki, can interoperate using their, respective, TinyRPL and ContikiRPL implementations.

Contiki Erbium CoAP implementation [17] that is bundled with Contiki 2.6 is used in our development. The Erbium REST engine includes framework for developing both CoAP

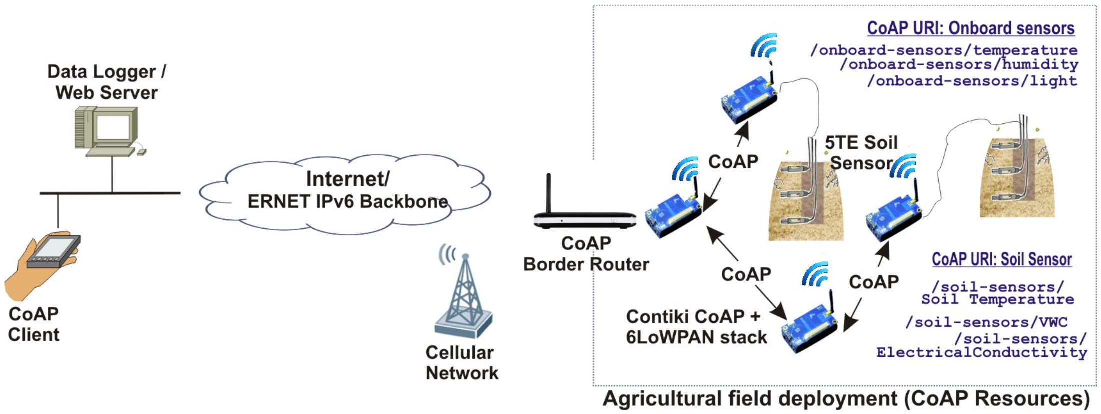


Fig. 4. Deployment Architecture for Agriculture Usecase

server and CoAP client applications. The release also includes a Firefox browser plugin Copper (Cu), a CoAP user agent implementation for monitoring resources using Web browser.

CoAP resources in Erbium are defined using the `RESOURCE` macro. For example, the following macro defines a temperature sensor resource. The parameters include name of the resource, request methods supported, URI path string, title and the resource type.

```
RESOURCE(temperature, METHOD_GET,
"onboard-sensors/temperature",
"title="Sensirion Temperature Sensor
(supports JSON)";rt="TemperatureSensor"");
```

Similarly, the *volumetric water content (VWC)* resource of Soil Sensor can be defined using the following macro.

```
RESOURCE(vwc, METHOD_GET,
"soil-sensors/VWC", "title="5TE Soil
Sensor (supports JSON)";rt="SoilSensor"");
```

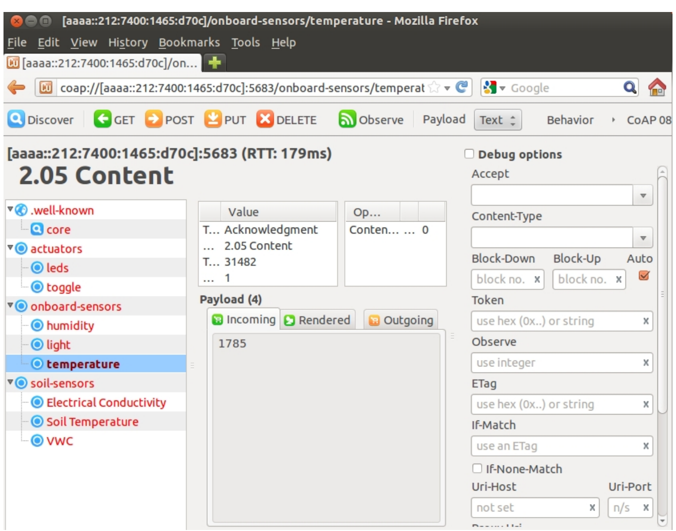


Fig. 5. Monitoring agriculture sensor: Web browser as CoAP Client

All the resources are statically defined and the associated functions are registered when the REST engine in CoAP server is started. Each resource has to implement a handler function with the name `[resource_name]_handler`. For example, when the CoAP client sends a `GET` request to the Erbium server for the CoAP URI `/soil-sensor/VWC`, `vwc_handler` function will be called by the Erbium REST engine. The handler implements actions for sending an excitation signal to 5TE soil sensor that results in reading of the VWC value. The CoAP response message is formatted according to the client requested format, which can be plain text, xml or JavaScript Object Notation (JSON). Figure 5 shows CoAP client plugin in Mozilla web browser displaying the `2.05 Content` response received for a `GET` request. The browser displays individual icons for each request methods like `GET`, `POST`, `PUT` and `DELETE`. The panel on the left showing the expanded list of resources for the `DISCOVER` request.

3) Field Deployment: Our agriculture sensor network monitoring using CoAP has been tested in a lab environment. The Proposed CoAP monitoring architecture for agricultural application is shown in Figure 4. The architecture shows the monitoring of remote agricultural sensor resources using CoAP. The architecture has two network segments, namely Agricultural field network and Monitoring Network. Field network consists of motes that are interfaced to soil sensors. The motes run CoAP server managing the monitorable CoAP resources. The soil sensor resources and onboard sensor resources can be discovered and CoAP methods can be acted on them using the identified CoAP URI. The CoAP application network will be connected to the ERNET IPv6 backbone/internet using the CoAP Proxy/border router. The CoAP proxy is the gateway for the field network to connect internet using cellular network. Monitoring network consists of CoAP client node(s) and web server. CoAP clients can be utilized for real-time access to sensor data, while web server can give access archived data to the farmer and the agricultural scientist.

IV. Discussion: Approaches to WSN Management & Monitoring

The 6LoWPAN adaption layer enables the reusability of application layer protocols of the TCP/IP stack for the LoWPAN networks. Simple Network Management Protocol (SNMP) [18] has been widely used for managing network resources

FEATURES	CoAP	SNMP
Mesg./Data format	Binary	MIBs / ASN.1
Identification	URI	Object ID (OID)
Query	GET	Get, GetNext, GetBulk
Create/Update	PUT, POST	Set
Remove	DELETE	*No equivalent operation*
Async. Notifications	OBSERVE	Trap, Inform
Resource Discovery	Multicast, CORE Link format	*Limited support -* SNMP Walk
Memory Footprint	Erbium CoAP: *Flash ROM: 8.5K, RAM: 1.2K*	Contiki-SNMP: *Flash ROM: 31.2K RAM: 0.2K*

TABLE II
COMPARING PROTOCOL FEATURES OF SNMP & COAP

for long. The features and protocol operations of CoAP are comparable to SNMP. But, SNMP is primarily designed for high-end network equipments like routers, while, CoAP has been designed keeping the resource constraints of sensor nodes in mind. But, there are also many adaptations & applications of SNMP [19], [20], [21], [22] for the low power wireless sensor networks. Table II lists the feature comparison of CoAP & SNMP. Some of the features essential to monitoring such as network-wide Resource Discovery is built-in to the CoAP protocol whereas SNMP protocol by itself does not support discovery, but using `SNMPwalk`, monitorable objects can be discovered within a device. The Resource Discovery in CoAP is achieved by sending a multicast `GET` request for the resources hosted in servers at the default entry point `"/.well-known/core"` and the response would be list of link descriptions as defined in CoRE Link Format [23].

In terms of the memory footprint, if we compare the Erbium CoAP [17] and Contiki-SNMP [21] implementation for the Contiki platform, the CoAP memory requirements are relatively low (see Table II) as compared to SNMP. The memory requirements for storing the resources in CoAP is also minimal. For example, in order to store 5 resources under `"/.well-known/core"`, Erbium requires about 1.5K of additional ROM memory.

V. CONCLUSION & FUTURE WORK

The number of devices that will connect to the Future Internet is increasing exponentially. The 6LoWPAN adaptation layer enables assignment of IPv6 addresses to low-power wireless devices making them reachable from any other node on the internet. In this paper, we have shown how CoAP open application layer protocol can be utilized for real-time monitoring of IP-enabled agriculture sensors network.

Our immediate future work includes the field deployment of our CoAP-based agriculture sensors network and its connectivity to ERNET IPv6 backbone for real-time monitoring over the internet. Further, we plan to integrate more agriculture sensors such as soil NPK, soil pH and leaf wetness. In addition to monitoring, motes can be connected to actuators in order to effect irrigation and fertigation in response to a particular condition. ERNET India is also planning to add heterogeneous devices and offer as an experimental facility to test various ideas relating to heterogeneity, interoperability, scalability, standardization, and so forth in the Internet of Things domain.

ACKNOWLEDGMENT

This project was funded under the R&D Grants-in-Aid by the Convergence, Communication & Broadband Technologies (CC&BT) Group, Department of Electronics & Information Technology (DeitY), Ministry of Communications and Information Technology, Government of India.

REFERENCES

[1] *Disruptive Civil Technologies: Six Technologies with Potential Impacts on US Interests out to 2025*. National Intelligence Council, April 2008.
[2] *Internet of Things An action plan for Europe*. http://ec.europa.eu/information_society/policy/rfid/documents/commiot2009.pdf.
[3] *Towards 50 Billion Devices: Personal & M2M*. http://www.ericsson.com/campaign/opportunitysupportsystems/newsfeed/posts/15-heading-towards-50-billion-connections/.
[4] D. Evans, *Top 25 Technology Predictions*. Cisco IBSG Innovations Practice, 2009.
[5] *IEEE Std 802.15.4 Specifications for Low-Rate Wireless Personal Area Networks WPANs*. IEEE Computer Society, September 2006.
[6] G. Montenegro, N. Kushalnagar, J. Hui and D. Culler, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*. IETF RFC 4944, September 2007.
[7] J. Hui and P. Thubert, *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*. IETF RFC 6282, September 2011.
[8] Z. Shelby, K. Hartke, C. Bormann and B. Frank, *Constrained Application Protocol (CoAP)*. draft-ietf-core-coap-11, July, 2012.
[9] *Constrained RESTful Environments (core) Working Group*. http://datatracker.ietf.org/wg/core/.
[10] C. Bormann, A. Castellani and Z. Shelby, *CoAP: An Application Protocol for Billions of Tiny Internet Nodes*. IEEE Internet Computing, 16(2):62-67, 2012
[11] R. Fielding, *"Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. Dissertation, University of California, Irvine, 2000.
[12] *TNAU agri portal - precision farming*. http://agritech.tnau.ac.in/pres_farm_agri.html.
[13] *5TE Soil Sensor*. http://www.decagon.com/products/sensors/soil-moisture-sensors/, 2012
[14] *Contiki: The Operating System for Connecting the Next Billion Devices - the Internet of Things*. http://www.sics.se/node/108, 2012.
[15] A. Dunkels, J. Alonso and T. Voigt, *Making TCP/IP Viable for Wireless Sensor Networks*. In First European Workshop on Wireless Sensor Networks, 2004.
[16] T. Winter and P. Thubert (Eds), *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. IETF RFC 6550, March 2012.
[17] M. Kovatsch, S. Duquennoy and A.Dunkels, *A Low-Power COAP for Contiki*. In IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, 2011.
[18] D. Harrington, R. Presuhn and B. Wijnen, *An Architecture for Describing SNMP Management Frameworks*. IETF RFC 3411, December 2002.
[19] *6PANview: A network monitoring system for the "internet of things"* http://ece.iisc.ernet.in/6panview, 2012.
[20] A. Paventhan, S.K. Allu, S. Barve, V. Gayathri and N. Mohan Ram, *Soil Property Monitoring using 6LoWPAN-enabled Wireless Sensor Networks*. In Third National Conference on Agro-Informatics and Precision Agriculture (AIPA), August 2012.
[21] S. Kuryla and J. Schönwälder, *Evaluation of the Resource Requirements of SNMP Agents on Constrained Devices*, In 5th International Conference on Autonomous Infrastructure, Management (AIMS), June 2011.
[22] H. Choi, N. Kim and H. Cha, *6LoWPAN-SNMP: Simple Network Management Protocol for 6LoWPAN*, In 11th IEEE International Conference on High Performance Computing and Communications (HPCC), June 2009.
[23] Z. Shelby, *Constrained RESTful Environments (CoRE) Link Format* IETF RFC 6690, August 2012.