Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 73-78 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

A "LIGHTER" JSON FOR MESSAGE EXCHANGE IN HIGHLY RESOURCE CONSTRAINED WIRELESS SENSOR NETWORK APPLICATIONS

César Ortega-Corral[2,1], Luis E. Palafox[1], J. Antonio García-Macías[3], Leocundo Aguilar[1], Jaime Sánchez-García[4], Ana Cristina Valenzuela-León[1], Israel Chon-Aguiar[1]

[1]Facultad de Ciencias Químicas e Ingeniería.
Universidad Autónoma de Baja California.
Calzada Tecnológico 14418, Mesa de Otay, Tijuana, B.C., CP22390
+52 (664) 6821 033 – 5800
Correo-e: cesar.ortega@uabc.edu.mx, lepalafox@uabc.edu.mx

[2]Tecnologías de la Información y Comunicación
Universidad Tecnológica de Tijuana
Km. 10 Carretera Libre Tijuana-Tecate,
Fracc. El Refugio. Quintas Campestre. Tijuana, B.C., C.P. 22650
Tel +52 (664) 969 9700
e-mail: cesar.ortega@uttjiuana.edu.mx

[3]Depto. De Ciencias de la Computación. [4]Depto. de Electrónica y Telecomunicaciones
Centro de Investigación Científica y de Educación Superior de Ensenada,
Carr. Ensenada–Tijuana No. 3918, Zona Playitas, Ensenada, B.C. 22860. México
+52 (646) 175 0500
e-mail: jagm@cicese.mx , jasan@cicese.mx

Keywords: Wireless Sensor Networks, Application Protocols, Network Communications.

ABSTRACT

This is an 8-bit implementation of the well known JSON (Java Script Object Notation) application protocol which has been modified to achieve very short messages while maintaining data integrity in a wireless sensor network (WSN). Most WSN have very low energy constraints, in this sense, the emphasis of this work is on saving transmission power during WSN communications by using structured data messages with very low syntax overhead. An added benefit is that at the gateway an encoder can be fashioned to translate Lighter JSON (L-JSON) messages to standard Unicode JSON, which can be sent through TCP sockets to a web server capable of doing data extraction and orderly non-volatile sensor data storage on a remote database system.

1. INTRODUCTION

With the arrival of the Word Wide Web (WWW), application protocols such as Hyper Text Transfer Protocol (HTTP) were developed to enclose information within tags so as to structure page layout visual designs which a remote client browser could decode and present to the user. Soon after, the standard called eXtensible Markup Language (XML) appeared offering structured data representations using tags as well [1]. Although XML has been a very successful information format, several other formats have been proposed. A notable proposal that became a standard is called JSON [2] which was originally created to be used by web browsers to receive complex information and script function calls in a simple yet highly structured manner. JSON is said to be a "lightweight" application protocol because its syntax is much simpler than that of XML. JSON is a data interchange format intended to be easy for humans to read/write and for software to encode and decode.

On the other hand, in portable device deployments an issue has always been extending their operational lifetime applying power saving strategies. This implies several approaches proposed for WSN technology that work in combination [3, 4]. One of which tries to reduce as much transmission power as possible by keeping the information payload as short as possible (cutting down on message tag overhead) reducing power consumption while maintaining reliable communications. This means a difficult compromise between saving as much power against conveying structured and thorough sensor data. The first issue is not apparent to a WSN system user, that may not grasp the whole remote telemetry problem. While the second part of the trade-off, complex structured data is what is important to the user. If it is done right the result will be reflected by receiving precise and complete information, with a sense of what and where things are happening in the wireless sensor field.

2. WSN STRUCTURE AND DATA ISSUES

There are several types of WSN topologies, the most basic is the star topology and the most complex is a full mesh network [5]. What they all have in common is that sensor data usually flows from endpoints towards a sink node, as shown in figure 1.

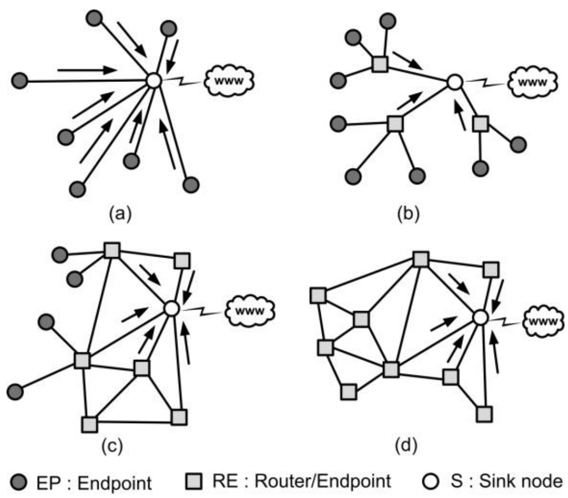

Figure 1. (a) Star topology. (b) Hierarchical tree network. (c) Partial mesh. (d) Full mesh topology.

This is a “many to one” model where the sink is the overall destination node located at the WSNs edge. From there, data can be sent to a remote non-volatile database storage system. Data flowing through the network may not be just acquired sensor data; it could also be configuration or operational status reports answering to base station queries. This means that messages sent through the WSN may hold varying amounts and types of data, and in need of a structured representation. In a few words, with WSN growth in size and functionality it will become necessary to deploy complex messaging between pending processes running on the networks distributed wireless embedded systems.

The WSN endpoints (some times called reduced function devices, RFD) acquire sensor data and transmit it to an associated node. Depending on the topology, the connections may be done directly to the sink node, like in a star network. Or maybe, the connections to the sink are done indirectly, like in a multi-hop mesh network. This is possible with the use of intermediate router nodes (sometimes called full functioning devices, FFD) through which the sensor data is forwarded towards the sink node [6]. With the potential of having a heterogeneous network (having both RFD and FFD in the field) a hierarchy of sorts is established and taken advantage of, needing increased messaging structure and communications activity.

In most WSN deployments, the sink node is attached to a more powerful computerized base station (BS), whose host is a PC or a standalone embedded system, where data buffering is done before it is sent out the gateway towards the Internet [7, 8]. Other BS tasks may have to do with endpoint (EP) configurations that can be issued remotely. This means transmitting messages with the appropriate syntax that the EP can understand, and from which it can be able to extract incoming commands and set new behaviors correctly.

After establishing the WSN messaging problem dimensions, the simple sensor network is not so simple anymore. Messaging becomes a central design issue if scaling and remote configuration capabilities are part of the system requirements [9, 10]. And as previously mentioned, a well known Internet application protocol called JSON has been widely used for object and data representations between application processes [11]. At first glance, JSON may be a good candidate to be used in WSN base station communications with the outside world, and then to a web server for manipulation and storage.

3. JSON: SYNTAX AND STRUCTURE

JSON messages represent variables and processes, they are built on two paradigms: (1) As an object organized as a collection of

Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 73-78 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

name/value pairs separated by a colon symbol: enclosed between braces (fig. 2), or (2) as an array made with a name string followed by a colon and the list of its values separated by comas enclosed within brackets, (fig. 3). All JSON names of objects, arrays or variables are strings and are invariably enclosed between quote symbols " ". Another important note is that JSON does not admit hexadecimal notation in any of its representations, which in embedded systems is very practical during code development and debugging.

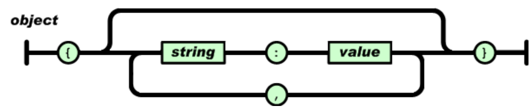

Figure 2. JSON object syntax.

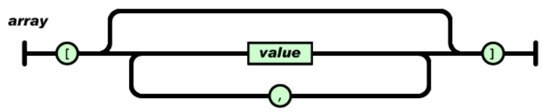

Figure 3. JSON array syntax.

Using JSON as the BS application protocol to communicate with an off-site web server is an efficient way of communicating structured data to the outside world. Nevertheless, if communication between small embedded systems would be done, it would be impractical because JSON uses Unicode characters which are 16 bits long. While usually, embedded systems uses 8 bits data units to establish serial communications.

4. LJSON MEANS "LIGHTER"JSON

From the start of any WSN deployment, the need for a practical way of sending aggregated and structured information from source nodes, through possible middle router nodes, and all the way to their sink nodes, is apparent and is resolved in many different forms. One such way has been with the use of custom API (Application Program Interface) framing and such [12], which in many respects is rigid and at a human level not easily understood. Also, there is always a need for more "intelligence" while coping with very low processing and energy resources. At an application level message exchange between wireless node processors needs to take place.

Here we propose using an existing application protocol for standard networks that can be humanly interpreted and yet be compact enough for WSN application data messaging. For this, we propose an 8-bit casted version of JSON with reduced syntax. In few words, two simple modifications to the standard are done: (1) the use of "quotes" for name definitions is suppressed and (2) 8-bit ASCII characters are used instead of 16 bit Unicode. The overall aim is to reduce the wireless message length and reduce transmission power overhead. For example, a structured JSON message that conveys EP sensor data is shown next:

```
{"e":2,"T":"1334545815","An":4,"Av"
:[235,9E8,81C,430]}
```

This string describes information of a single member EP object whose ID number is "e":2, with a timestamp, an ADC counter number "An":4 and the sensor data array "Av" in hexadecimal notation enclosed in brackets. This message is 54 characters in length. If 16-bit Unicode were to be used it would actually be 108 bytes long. Using 8-bit characters and eliminating the quote symbols that enclose the name strings, the L-JSON coding would then be:

```
{e:2,T:"1334545815",An:4,Av:[235,9E
8,81C,430]}
```

The resulting string is 44 ASCII characters long, equal to 44 bytes. Meaning that by using L-JSON, the result is only 40.7% that of the standard JSON byte length. As stated before, these LJSON messages are exchanged between different endpoints, routers and the overall sink, within the WSN communication structure.

Another important difference is that ordinary JSON does not admit hexadecimal numbers. But in this L-JSON implementation, hexadecimal notation is included in the coding and decoding process. It can be justified in saying that in embedded systems the use of HEX is very practical and efficient when doing logical or network processing.

5. USE CASES: TIME SYNC AND SAMPLING IMPLEMENTATIONS

In figure 4, a common use case is represented as the Time Synchronization sequence diagram. Different systems update time in varying ways [13]. In this case, when a wireless node starts up, an initial action is to send a Time Synchronization Request through a corresponding LJSON message. A common

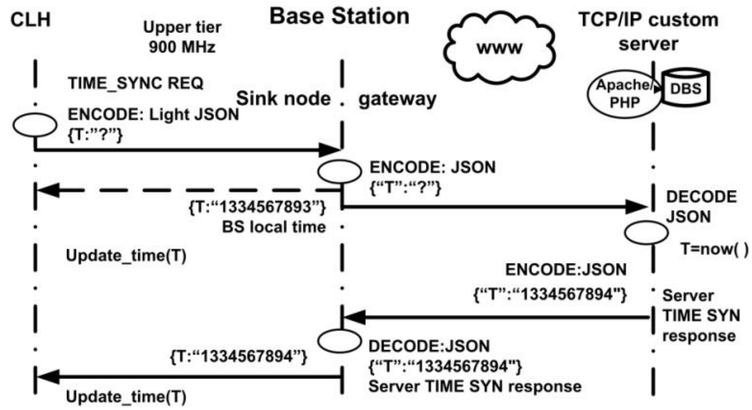

Figure 4. Endpoint time synchronization use case.

sense message could be {T:?}. The BS responds through its sink with its local time value and sends a Time Request JSON string {"T": "?"} to a server with more resources to answer with a precise Time Synchronization response. A subsequent **now( )** value is sent from the server to the BS, which broadcasts a delay adjusted value to the WSN, so not just the requesting node but all the nodes in the field can perform a **time_update( )** procedure.

The next common use case is automatic EP sampling and data transfer all the way to the WWW. The related sequence diagram is shown in figure 5.

6. HIERARCHICAL WSN FOR MARINE HABITAT MONITORING AND L-JSON

This modified messaging scheme is being developed for a long range Hierarchical WSN (HWSN) prototype (with a topology similar to figure 1.b) that will convey environmental sensor data coming from groups of endpoints organized as "clusters". Each of these groups has a main node, called a "cluster-head" (CLH), through which all outside interaction is done. This hierarchical WSN will be applied to long range marine habitat monitoring. The general two tier architecture is shown in fig. 6.

The EPs automatically acquire sensor data and transmit it to their associated CLHs using a 2.4GHz channel. Every CLH has two radios onboard, a 2.4 GHz low tier transceiver and another operating at 900MHz for higher tier communications. When the CLH receives EP data it constructs an application layer message and sends it to the base station. In this case, we propose that the messages be in a L-JSON format.

The BS microprocessor extracts sensor data from the CLH L-JSON message and other important EP information. And then the BS converts the received light JSON messages to standard Unicode JSON, then their sent to a web server where message decoding is done using a server scripting language, which finally stores relevant information in flat files or on a more powerful database system.

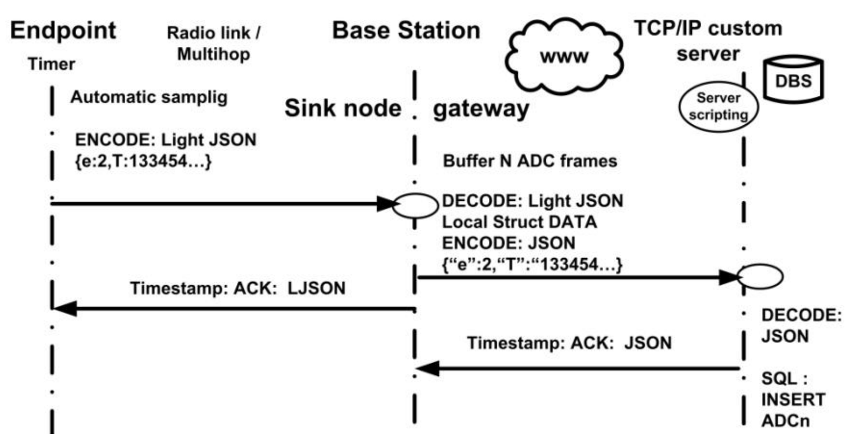

Figure 5. Automatic EP sampling sequence diagram

Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 73-78 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

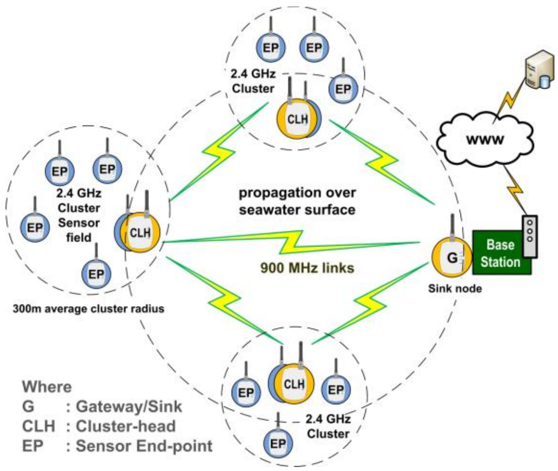

Figure 6. The long range dual frequency two tier HWSN.

In this implementation, other use cases were deployed with different actions on behalf of CLH, the BS controller, and at the server side. Web server scripting was used to develop a custom TCP server, which finally queries the DBS to store particular sensor data, which was originally transmitted through LJSON/JSON messaging. For example, a structured L-JSON message that conveys EP sensor data is shown next:

```
{C:1,En:1,e:[{i:2,T:"1334545815",An
:4,Av:[235,9E8,81C,430]}]}
```

This is an ASCII string that describes CLH object information of cluster number 1, it holds a single member EP object of an array whose ID number is 2 (e:[{i=2,…). It is 61 bytes long. After JSON recoding the resulting string is

```
{"C":1,"En":1,"e":[{"i":2,"T":"1334
545815","An":4,"Av":[235,9E8,81C,43
0]}]}
```

With 16-bit Unicode the message is actually 152 bytes long. This means that the L-JSON is only 39.6% that of the JSON byte length, similarly to the previous L-JSON generic examples

In figure 7, a time-stamped sensor data table is shown after tests were done. The sampled data was sent by endpoints with EP_IDs 1 and 2. Both sent a four sample vector stored as columns ADC0, ADC1, ADC2 and ADC3. In our application these 10-bit sensor values represent air temperature, seawater temperature, the surrounding air relative humidity and the remaining battery level. Application specific sensor information interpretation is another matter, and it is not covered in this document.

ID	TIME_STAMP	CLH_ID	EP_ID	n	ADC0	ADC1	ADC2	ADC3
173	1334548483	1	1	4	1023	1023	1023	1023
172	1334548481	1	1	4	1023	1023	1023	1023
171	1334548480	1	2	4	133	228	271	321
170	1334548478	1	1	4	1023	1023	1023	1023
169	1334548478	1	2	4	142	224	275	334
168	1334548476	1	2	4	134	223	276	314

Figure 7. Image of the SQL table that holds the endpoint sample information successfully transferred.

The next step is to deploy other use cases, which will make the HWSN remotely configurable. Also, worth mentioning is that up to now we have not taken into account the wireless message maximum length. And if a LJSON messages exceed the wireless frame payload size then message fragmentation will take place. A classic solution would be to include in the LJSON message a sequence number to indicate the fragment order for message reassembly purposes on the receiving ends software.

7. CONCLUSIONS

The need to conserve battery power for remote portable applications in WSN is always present. In this sense, message length impacts the overall network lifetime. LJSON with its simple modifications takes advantage of JSON syntax simplicity with even lighter notation and byte length characters with the purpose of saving transmition power while maintaining highly structured information.

Further tests will yield actual transmition time results and power savings by using LJSON in larger structured message transmitions. These initial results are promising considering that even with a short sampling message LJSON achieves at least a 60% shorter byte length string compared to using standard JSON encoding between wireless sensor devices.

8. REFERENCES.

[1] Official web site. ***Extensible Markup Language, XML***. http://www.w3.org/TR/REC-xml/. Visited 12/04/2012.

[2] RFC4627. ***JSON: JavaScript Object Notation Std. Spec***. http://www.ietf.org/rfc/rfc4627.txt. Last visited 12/04/2012.

Congr. Int. Ing. Electrón. Mem. Electro 2012, vol.34, pp. 73-78 Chihuahua, Chih. México
http://depi.itchihuahua.edu.mx/display/memorias_electro/MemoriaElectro2012.zip
ISSN 1405-2172

[3] R. C. Shah and J. Rabaey, ***Energy Aware Routing for Low Energy Ad Hoc Sensor Networks***. IEEE Wireless Communications and Networking Conference (WCNC), March 17-21, 2002.
[4] Shio K. Singh, M.P. Singh, D.K. Singh. ***A survey of energy efficient hierarchical cluster-based routing in wireless sensor networks.*** J. of Avanced Netowrking and Applications. Vol. 2. Issue 2. P. 570-580. 2010.
[5] IEEE Standards Committee 802.15.4. ***Part 15.4: Wireless MAC and PHY Layer Specifications for Low Rate WPAN.*** Institute of Electrical and Electronics Engineers. New York: IEEE Press. 2006.
[6] Official web site. ***Zigbee Alliance***. 2006 Std. http://www.zigbee.org/ Visted april/2012.
[7] Chris Townsend, Steven Arms. ***Wireless Sensor Networks: Principles and Applications***. Sensor technology handbook (2005) . Volume: 11, Issue: 6, Publisher: Newnes, Pages: 575-589.
[8] Ethan Culler-Mayeno. ***A Technical Report: Wireless Sensor Networks and How They Work.*** University of California Santa Barbara. 2006.
[9] Barjinder Singh Kaler, Er. Manpreet Kaur Kaler. ***Challenges in Wireless Sensor Networks***. ACM SIGBED Review (2004). Volume: 1, Issue: 2, Publisher: ACM Press, Pages: 142-142.
[10] Azzedine Boukerche. ***Algorithms and Protocols for Wireless Sensor Network***. Book Ed. John Wiley & Sons Inc. 2009.
[11] Official web site. ***JavaScript Object Notation***. http://www.json.org/. Last visited 15/04/2012.
[12] [12] Digi Intl. Inc ***XBee-PRO® 900/DigiMesh™ 900 RF Modules.*** User's manual, 2009.
[13] Cox, D.; Jovanov, E.; Milenkovic, A.. ***Time synchronization for ZigBee networks***. System Theory, 2005. SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on system Theory. March 2005. page(s): 135 - 138
[14]Erzsebet Ravasz and Albert-Laszlo Barabasi. ***Hierarchical organization in complex networks***. The American Physical Society Journal. 2003.