Constrained Application Protocol for Internet of Things

Xi Chen, chen857 (at) wustl.edu (A paper written under the guidance of Prof. Raj Jain)

Download 

Abstract:

Internet of things (IoT) is an important part of a new generation of technology that every object no matter things or human could be connected to Internet. There are many wireless protocols (like IEEE 802.11 Series, 802.15 Series, Zigbee, etc) for communication between devices. However, considering a lot of small devices are unable to communicate efficiently with constrained resources, Internet Engineering Task Force (IETF) has developed a lightweight protocol: Constrained Application Protocol (CoAP). Firstly, this paper summarizes some wireless protocols. Then it introduce CoAP and corresponding security protocol DTLS. Finally, an application is given.

Keywords

IoT, CoAP, DTLS, wireless protocols, smart home, energy control system

Table of Contents:

1. Introduction

Internet of Things (IoT) is represented as a global network which intelligently connects all the objects no matter devices, systems or human, it is with self-configuring capabilities based on standard and

interoperable protocols and formats [Petersburg 12]. Through smart sensing, identification technology and persuasive computing, IoT has been called the Third Wave in information industry following the computer and the Internet. There are hundreds of protocols supported by IoT. Of the many protocols, wireless protocols play an important role in IoT development.

In section 1, some wireless protocols in different layers of IoT are introduced. One latest protocol for application layer CoAP is given and its features and functions are summarized. By comparing it with Hypertext Transfer Protocol (HTTP), its advantages are presented. In section 2, some important CoAP models are explained in details, such as the message layer model, request/response layer model and message format. In sections 3, a security protocol Datagram Transport Layer Security (DTLS) for transmission protection is described. It achieves necessary elements for securing CoAP, like integrity, authentication and confidentiality. Then, an application of CoAP Smart Homes is described, it helps users to manage energy control systems, which reduce power consumption and prevent accidents.

2. Overview of Wireless Protocols for IoT

IoT needs to integrate various sensors, computer and communication equipment, which are using different communication protocols. Wireless Protocols are mainly used in three layers, which are PHY/MAC layer, Network/Communication layer and Application layer.

CoAP is one of the latest application layer protocol developed by IETF for smart devices to connect to Internet. As many devices exist as components in vehicles and buildings with constrained resources, it leads a lot of variation in power computing, communication bandwidth etc. Thus lightweight protocol CoAP is intended to be used and considered as a replacement of HTTP for being an IoT application layer protocol.

2.1 Protocols in Different Layers

IoT PHY/MAC Layers involve all the common wireless communication technology, such as IEEE 802.11 series, 802.15 series, HART (Highway Addressable Remote Transducer), etc. IEEE 802.15.4 standard specifies MAC/PHY part for long-range wireless personal area network (LR-WPAN). Zigbee, WirelessHART are based on IEEE802.15.4 by adding upper layers.

As TCP/IP lay the foundation for the Internet, thus IoT communication network mainly employ TCP and UDP protocols. Compared with UDP protocol, TCP protocol is more complex which makes it not easy to employ on resource-constrained devices. Now, most of IoT use UDP protocol. But UDP is not stable, which needs to combine with application layer to improve the stability.

Application layer usually employ HTTP to provide web service, but HTTP has high computation complexity, low data rate and high energy consumption. Therefore, IETF has developed several lightweight protocols, e.g., CoAP, Embedded Binary HTTP (EBHTTP), Lean Transport Protocol (LTP). The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks [Z.Shelby13]. EBHTTP is a binary-formatted, space-efficient, stateless encoding of the standard HTTP/1.1 protocol [G.Tolle13]. EBHTTP is primarily designed for transportation of small data between resource-constrained nodes, which is similar to CoAP. LTP is a lightweight Web Service transport protocol that allows the transparent exchange of Web Service messages between all kinds of resource constrained devices and server or PC class systems [Glombitza10].Table 1 summarizes protocols in different layers.

Table 1 protocols in different layers

Application layer	HTTP, CoAP, EBHTTP, LTP, SNMP, IPfix, DNS, NTP, SSH, DLMS, COSEM, DNP, MODBUS
Network/Communication layer	IPv6/IPv4, RPL, TCP/UDP, uIP, SLIP, 6LoWPAN,
PHY/MAC layer	IEEE 802.11 Series, 802.15 Series, 802.3, 802.16, WirelessHART, Z-WAVE, UWB, IrDA, PLC, LonWorks, KNX

2.2 CoAP Features

With the completion of the CoAP specification, it is expected that there will be million of devices deployed in various application domains in the future. These applications range from smart energy, smart grid, building control, intelligent lighting control, industrial control systems, asset tracking, to environment monitoring. CoAP would become the standard protocol to enable interaction between devices and to support IoT applications [S.Keoh13]. The Constrained RESTful Environments (CoRE) is the workgroup in IEFT that is designing the CoAP protocol.

CoAP needs to consider optimizing length of datagram and satisfying REST protocol to support URI (Uniform Resource Identifier). It also needs to provide dependable communication based on UDP protocol. Table 2 shows the CoAP features [Petersburg12]

Table 2 CoAP features

Constrained web protocol fulfilling M2M requirements
Security binding to Datagram Transport Layer Security (DTLS)
Asynchronous message exchanges
Low header overhead and parsing complexity.
URI and Content-type support.
Simple proxy and caching capabilities
UDP binding with optional reliability supporting unicast and multicast requests.
A stateless HTTP mapping, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way or for HTTP simple interfaces to be realized alternatively over CoAP.

2.3 CoAP vs. HTTP

CoAP is network-oriented protocol, using similar features to HTTP but also allows for low overhead, multicast, etc. As HTTP protocol is a long-term successful standard, it can use small script to integrate

various resources and services. Interoperation provided by HTTP is the key point of IoT, for this, HTTP is employed in application level. However, HTTP is based on TCP protocol using point to point (p2p) communication model that not suitable for notification push services. Also, for constrained devices, HTTP is too complex.

Unlike HTTP based protocols, CoAP operates over UDP instead of using complex congestion control as in TCP [Koojana11]. CoAP is based on REST architecture, which is a general design for accessing Internet resources. In order to overcome disadvantage in constrained resource, CoAP need to optimize the length of datagram and provide reliable communication. On one side, CoAP provides URI, REST method such as GET, POST, PUT, and DELETE. On the other side, based on lightweight UDP protocol, CoAP allows IP multicast, which satisfies group communication for IoT. To compensate for the unreliability of UDP protocol, CoAP defines a retransmission mechanism and provides resource discovery mechanism with resource description [Shelby11]. Fig 1 shows the HTTP and CoAP protocol stacks.

Fig 1: HTTP and CoAP protocol stacks

CoAP is not just a simply compression of HTTP protocol. Considering low processing capability and low power consuming demand of restrained resource, CoAP redesigned some features of HTTP to accommodate these limitations. HTTP used under unconstrained network and CoAP used under constrained network. Recently, HTTP-CoAP cross protocol proxy arouses scientific interest, it has and important role in solving congestion problem in the constrained environment [Castellani12].

3. CoAP Structure Model

CoAP interactive model is similar to HTTP's client/server model. Fig 2 shows that CoAP employs a two layers structure. The bottom layer is Message layer that has been designed to deal with UDP and asynchronous switching. The request/response layer concerns communication method and deal with request/response message.

Fig. 2: Abstract Layer of CoAP [Z.Shelby13]

3.1 Message Layer model

Message Layer supports 4 types message: CON (confirmable), NON (non-confirmable), ACK (Acknowledgement), RST (Reset) [Z.Shelby13].

a) Reliable message transport: Keep retransmission until get ACK with the same message ID (like 0x8c56 in fig. 3). Using default time out and decreasing counting time exponentially when transmitting CON. If recipient fail to process message, it responses by replacing ACK with RST. Fig. 3 shows a reliable message transport.

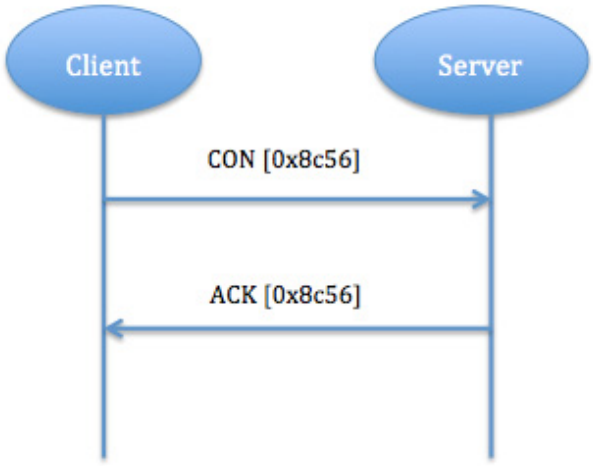


Fig. 3: Reliable Message Transport

b) Unreliable message transport: transporting with NON type message. It doesn't need to be ACKed, but has to contain message ID for supervising in case of retransmission. If recipient fail to process message, server replies RST. Fig. 4 shows unreliable message transport.

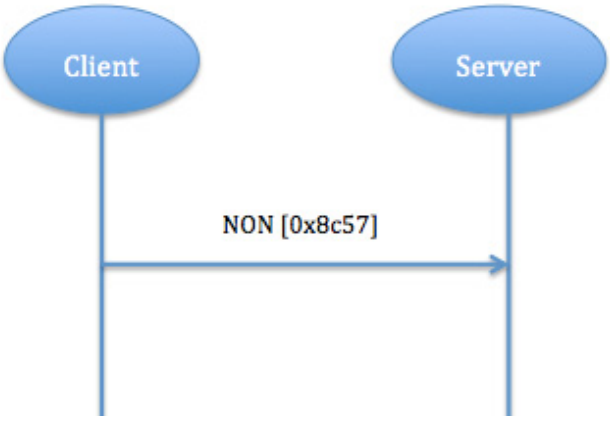


Fig. 4: Unreliable Message Transport

3.2 request/response Layer model

a) Piggy-backed: Client sends request using CON type or NON type message and receives response ACK with confirmable message immediately. In fig. 5, for successful response, ACK contain response message (identify by using token), for failure response, ACK contain failure response code.

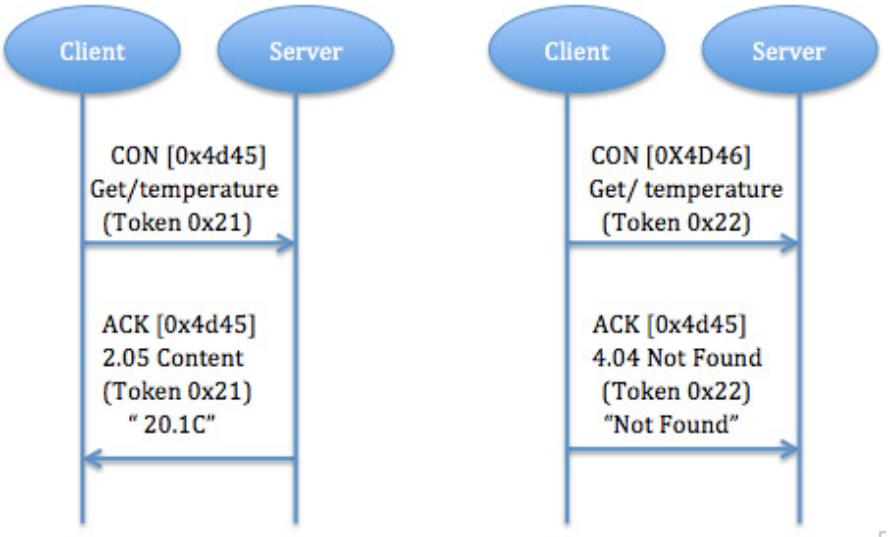


Fig. 5: The successful and failure response results of GET method

b) Separate response: If server receive a CON type message but not able to response this request immediately, it will send an empty ACK in case of client resend this message. When server ready to response this request, it will send a new CON to client and client reply a confirmable message with acknowledgment. ACK is just to confirm CON message, no matter CON message carry request or response (fig. 6).

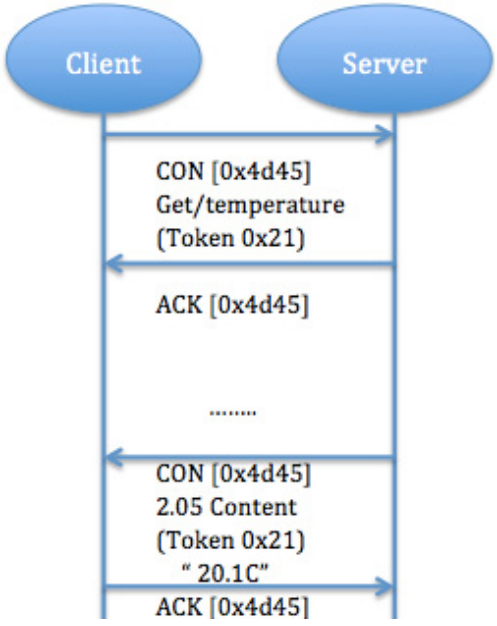


Fig. 6: A Get request with a separate response

c) Non confirmable request and response: unlike Piggy-backed response carry confirmable message, in Non confirmable request client send NON type message indicate that Server don't need to confirm. Server will resend a NON type message with response (fig. 7).

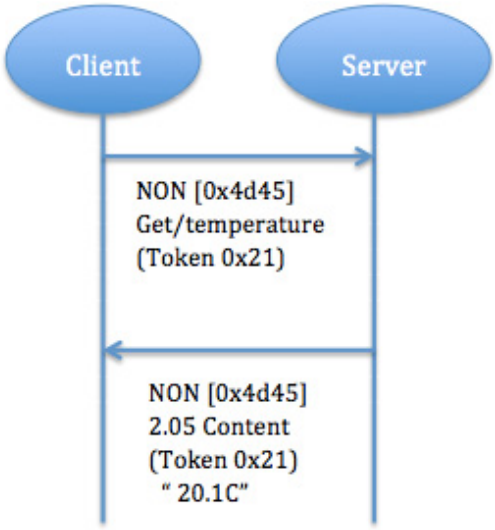


Fig. 7: Non confirmable request and response

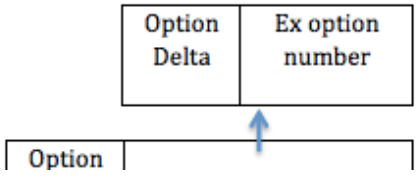
3.3 Message Format

CoAP is based on the exchange of compact messages that, by default, are transmitted over UDP (i.e. each CoAP message occupies the data section of one UDP datagram) [Petersburg12]. Message of CoAP uses simple binary format. Message= fixed-size 4-byte header plus a variable-length Token plus a sequence of CoAP options plus payload. The format is shown in Table 3.

Table 3 Message Format

0			1													2														3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver			T			OC			Code							MessageID																
Token (if any, TKL bytes)…																															
Options (if any)…																															
Payload (if any)…																															

Option number=option delta+ ex option number. The format is shown in Fig. 8



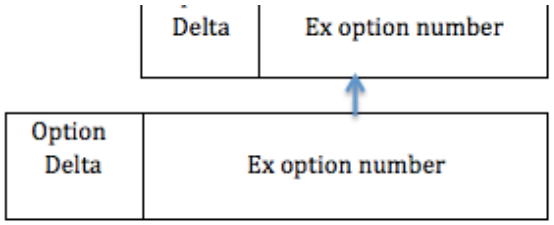


Fig. 8: CoAP option format

4. Security Protocol & Application for CoAP

CoAP is now becoming the standard protocol for IoT applications. Security is important to protect the communication between devices. In the following part, a security protocol DTLS is introduced. Also, one of CoAP application, Smart Homes, is described in this section.

There are three main elements when considering security, namely integrity, authentication and confidentiality. DTLS can achieve all of them [Kothmayr 12]. IETF modifies DTLS to develop another protocol DTLS. DTLS employ TCP, which is too complex. DTLS solves two problems: reordering and packet lost. It adds three implements: 1 packet retransmission. 2 assigning sequence number within the handshake. 3 replay detection.

Unlike network layer security protocols, DTLS in application layer (fig.9) protect end-to-end communication. No end-to-end communication protection will make it easy for attacker to access to all text data that passes through a compromised node. DTLS also avoids cryptographic overhead problems that occur in lower layer security protocols.

4.1 Why use DTLS for CoAP Security

There are three main elements when considering security, namely integrity, authentication and confidentiality. DTLS can achieve all of them[Kothmayr12]. IETF modifies DTLS to develop another protocol DTLS. DTLS employ TCP, which is too complex. DTLS solves two problems: reordering and packet lost. It adds three implements: 1 packet retransmission. 2 assigning sequence number within the handshake. 3 replay detection.

Unlike network layer security protocols, DTLS in application layer (fig. 9) protect end-to-end communication. No end-to-end communication protection will make it easy for attacker to access to all text data that passes through a compromised node. DTLS also avoids cryptographic overhead problems that occur in lower layer security protocols.

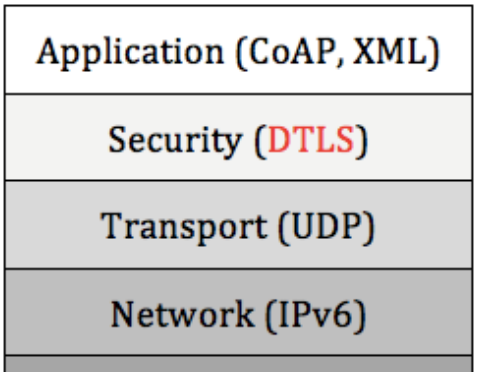


Fig. 9: DTLS in protocol stack

4.2 Structure of DTLS

There are two layers in DTLS. The bottom one contains Record protocol. The top one include three protocols which are Alert, Handshake and application data, in some condition Change Cipher Spec protocol may replace one of them. Change Cipher Spec message is used to notify Record protocol to protect subsequent records with just-negotiate cipher suite and keys. [Raza13]

Record protocol [E.Rescorla12]protects application data by using keys generated during Handshake. For outgoing message, protocol divide, compress, encrypt and apply Message Authentication Code (MAC) to them. For incoming message, protocol reassemble, decompress, decrypt and verify them. Record header consists of two parts, one is content type and another is fragment field. Content type decides what is contained in fragment field. It could be alert protocol, Handshake protocol or application data. Compare with DTLS Record, Handshake protocol is rather a complex one, which involves in a lot of exchange steps. Individual messages are grouped into message flights. Fig. 10 shows the process of Handshake.

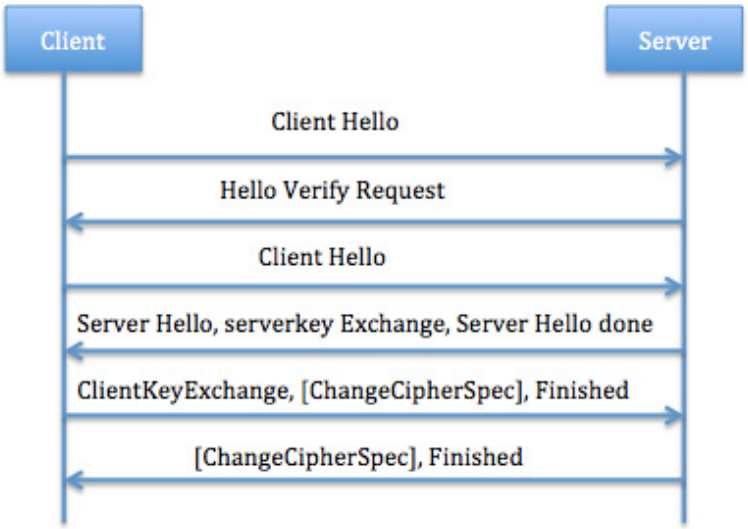
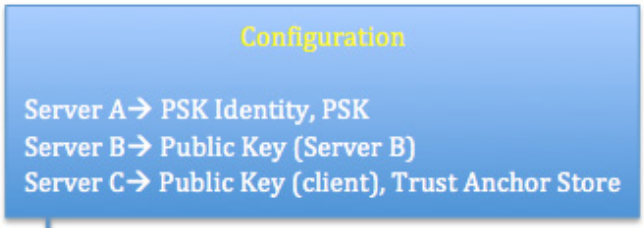


Fig. 10: Process of Handshake

The Architecture show below (Fig. 11) is for uni-cast communication (client interacts with one or more servers). The client also needs to know which certificate or raw public key it has to use with a specific server [Hartke13].



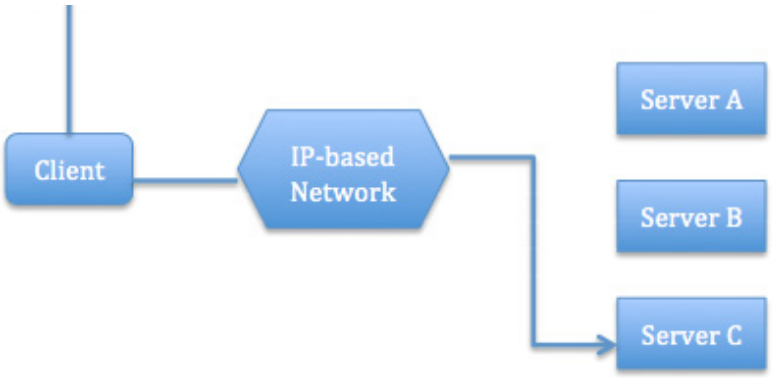


Fig. 11: Uni-cast communication model

4.3 CoAP Application for Smart Homes

Information appliance, control equipment and communication equipment in Smart home networks have the characters of low-cost and lightweight. Thus, CoAP could be seen as the best protocol choice for home communication networks.

Smart home network provide controlling and monitoring energy of home devices. Energy control systems employ smart socket management and monitor power consuming equipment to provide voltage, current and other energy information. It could realize accident warning, remote control and dynamic energy saving. The system structure is shown in Fig. 12. Every data collection node with CoAP client could exchange information with other nodes. CoAP could both be installed in LAN or Internet. Unlike Many wireless protocols for home automotive devices, CoAP is designed not constrained in a local network but provide the fundamental basis of the web [Bergmann12]. In this system, CoAP-HTTP proxies are employed to provide HTTP client connection to CoAP resources and vice versa.

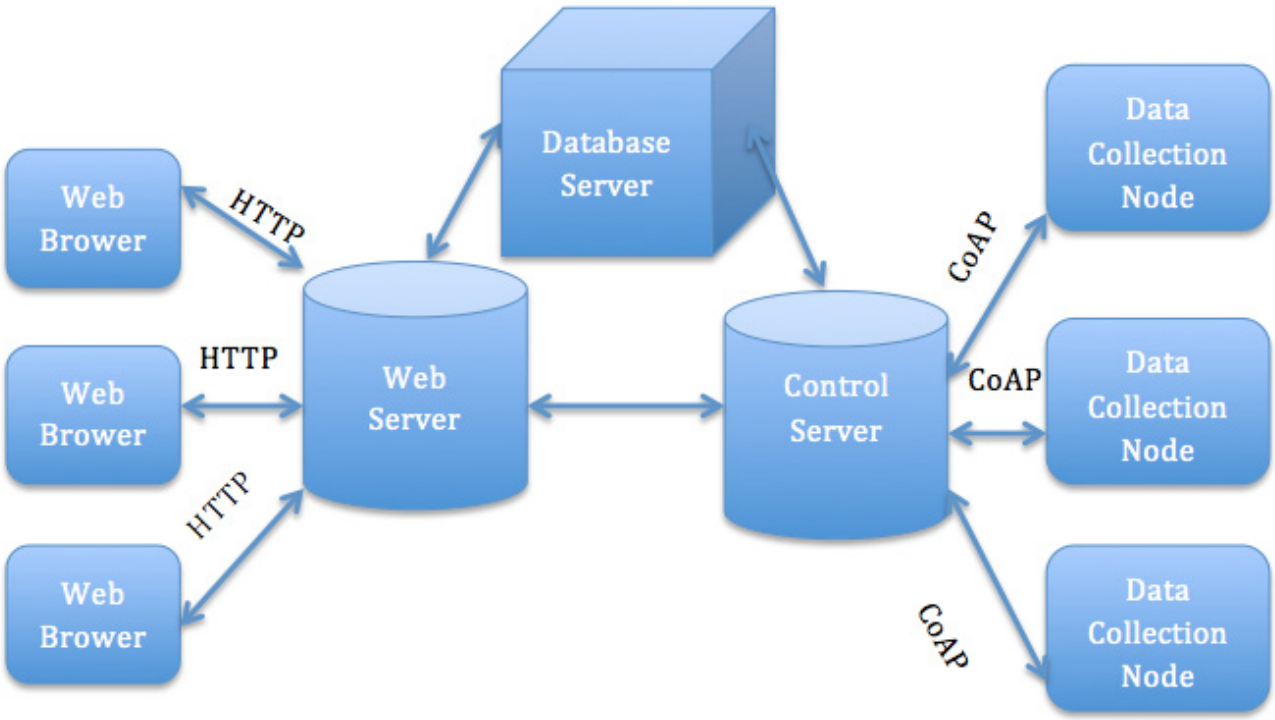


Fig. 12: energy control system

In system networking, date collection nodes consist of one proxy, smart socket and wireless data collection module. Energy information and environment information of equipment is collected by the smart socket and transported to data collection module through wireless channel, then send serial data to proxy to process and pack data. Control server analyzes all the data and stores them in database. The system integrate home network and Internet, users can access system webpage to remotely control switch, manage configuration, query energy consumption, etc. At the same timei1/4OEthis system will monitor environment situation. If any abnormal things happens, (like high temperature or voltage beyond the safety range) the system will analyze it and close relevant equipment.

5. Conclusion

This report summarizes some wireless protocols for IoT and introduces CoAP protocol in details with describing main features and modules. CoAP is based on HTTP protocol and is designed for constrained resource devices. Through comparing CoAP and HTTP together, the advantages of CoAP for IoT are analyzed. This report also provides a corresponding security protocol DTLS and one possible application of CoAP for IoT. As a number of protocols are being perfected, more and more smart homes will employ these wireless protocols for intelligent control.

6. References

- [Petersburg12] St. Petersburg, "Internet of Things, Smart Spaces, and Next Generation Networking," Russia, August 27-29, 2012 http://link.springer.com/book/10.1007%2F978-3-642-32686-8
- [Z. Shelby13] Z. Shelby, Sensinode, K. Hartke, "Constrained Application Protocol (CoAP)," draft-ietf-core-coap-18. [2013-06--28] http://tools.ietf.org/html/draft-ietf-core-coap-18
- [G.Tolle13] G.Tolle, Arch Rock Corporation. Embedded Binary HTTP (EBHTTP) draft-tolle-core-ebhttp-00.[2010-03-23] https://tools.ietf.org/html/draft-tolle-core-ebhttp-00
- [Glombitza10] Glombitza, N.; Pfisterer, D.; Fischer, S., "LTP: An Efficient Web Service Transport Protocol for Resource Constrained Devices," Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on , vol., no., pp.1,9, 21-25 June 2010 http://ieeexplore.ieee.org/xpl/login.jsptp=&arnumber=5508255&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5508255
- [S. Keoh13] S. Keoh, Philips Research, Z. Shelby,Sensinode. Profiling of DTLS for CoAP-based IoT Applications draft-keoh-dice-dtls-profile-iot-00 [2013-11-05] http://tools.ietf.org/html/draft-keoh-dice-dtls-profile-iot-00
- [Koojana11] Koojana Kuladinithi, Olaf Bergmann, Thomas Potsch Markus Becker, Carmelita Gorg, "Implementation of CoAP and its Application in Transport Logistics," In Proceedings of the Workshop on Extending the Internet to Low power and Lossy Networks (April 2011) http://hinrg.cs.jhu.edu/joomla/images/stories/coap-ipsn.pdf
- [Shelby11] Zach Shelby, "CoRE Link Format," draft-ietf-core-link- format-07 https://tools.ietf.org/html/draft-ietf-core-link-format-01
- [Castellani12] Castellani, A.; Fossati, T.; Loreto, S., "HTTP-CoAP cross protocol proxy: an implementation viewpoint," Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on , vol.Supplement, no., pp.1,6, 8-11 Oct. 2012 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6708523

- [Kothmayr12] A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-way Authentication Thomas Kothmayr, Corinna Schimitt, Wen Hu, Michael Bruning. 2012 http://kothmayr.net/wp-content/papercite-data/pdf/kothmayr2012dtls.pdf
- [Raza13] Raza, S.; Shafagh, H.; Hewage, K.; Hummen, R.; Voigt, T., "Lithe: Lightweight Secure CoAP for the Internet of Things," Sensors Journal, IEEE , vol.13, no.10, pp.3711,3720, Oct. 2013 http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6576185
- [E. Rescorla12] E. Rescorlai, N. Modadugu, "Datagram Transport Layer Security Version 1.2," RFC Standard 6347, Jan. 2012. http://tools.ietf.org/html/rfc6347
- [Hartke13] Klaus Hartke, Hannes Tschofenig, A DTLS Profile for the Internet of Things draft-hartke-dice-profile-00, [2013-11-04] http://tools.ietf.org/html/draft-hartke-dice-profile-00
- [Bergmann12] Bergmann, O.; Hillmann, K.T.; Gerdes, S., "A CoAP-gateway for smart homes," Computing, Networking and Communications (ICNC), 2012 International Conference on, pp.446,450, Jan. 30 2012-Feb. 2 2012 http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6167461&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6167461

7. Acronyms

IoT	Internet of Things
IEFT	Internet Engineering Task Force
CoAP	Constrained Application Protocol
HTTP	Hypertext Transfer Protocol
HART	Highway Addressable Remote Transducer
LR-WPAN	Low-Rate Wireless Personal Area Network
EBHTTP	Embedded Binary HTTP
LTP	Lean Transport Protocol
CoRE	Constrained RESTful Environments
REST	Representation State Transfer
URI	Uniform Resource Identifier
P2P	Point to Point
DTLS	Datagram Transport Layer Security

Last Modified: April 30, 2014
This and other papers on current issues in Wireless and Mobile Networking are available online at
http://www.cse.wustl.edu/~jain/cse574-14/index.html
Back to Raj Jain's Home Page