

Implementation Cloud Platform with CoAP to Enhance IOT Services Performance

Ta-Te Lu*, Hao-Hsiang Hung, Yie-Wan Wang

Department of Computer Science and Information Engineering, Chien Hsin University of Science and Technology, Taoyuan City, Taiwan

Email address:

ttl@uch.edu.tw (Ta-Te Lu), m10313006@gapps.uch.edu.tw (Hao-Hsiang Hung), m10213012@gapps.uch.edu.tw (Yie-Wan Wang)

*Corresponding author

To cite this article:

Ta-Te Lu, Hao-Hsiang Hung, Yie-Wan Wang. Implementation Cloud Platform with CoAP to Enhance IOT Services Performance. *Science Discovery*. Vol. 4, No. 4, 2016, pp. 207-212. doi: 10.11648/j.sd.20160404.11

Received: April 10, 2016; **Accepted:** April 12, 2016; **Published:** July 20, 2016

Abstract: Recently, the “Internet of Things” (IOT) has become a popular technical term for machine-to-machine (MM) communication. The IOT is an interaction communication infrastructure for consumer electronics (CE) to connect smart objects. We developed the cloud platform with the Constrained Application Protocol (CoAP) Communication technology to enhance IOT services performance. The proposed technology allows smart objects to access and push cloud resources over the cloud platform. The IETF Constrained RESTful Environments (CORE) working group has defined the CoAP to integrate with web services. The main features of CoAP are UDP transport, asynchronous message exchanges, and low complexity. The objective is to implement the Java Script Object Notation (JSON) to transmit messages between the CoAP controller and the cloud, providing a stable cloud environment and sending CoAP messages from clients to the CoAP controller in each group.

Keywords: Internet of Things, The Constrained Application Protocol, Smart Sensors

實作 CoAP 通訊技術以提升物聯網雲端服務傳遞效能

魯大德*, 洪浩翔, 王亦文

電資學院資訊工程系, 健行科技大學, 桃園市中壢區, 台灣

邮箱

ttl@uch.edu.tw (魯大德), m10313006@gapps.uch.edu.tw (洪浩翔), m10213012@gapps.uch.edu.tw (王亦文)

摘要: 國際電信聯盟於2005年所發布的報告中提出物聯網The Internet of Things (IOT), 邁向工業4.0, 物聯網是一種互動式傳輸平台可用於連結智慧型感應裝置, 具有廣闊的市場和應用前景。IETF(Internet Engineering Task Force)的CoRE工作組提出了一種基於REST架構的the Constrained Application Protocol (CoAP)協定可以減少IOT 協定傳輸的複雜性。本研究實作一種 CoAP Controller以提升物聯網雲端服務傳遞效能, 減少CoAP與HTTP傳輸協定轉換的複雜性。以樹梅派實作出CoAP Controller 接收來自雲端命令及CoAP clients, CoAP clients 和 CoAP controller 間採用 CoAP 協定傳輸訊息, 雲端及CoAP controller 間則採用 the Java Script Object Notation (JSON) 作為資料交換, 可提高雲端與物聯網產品市場競爭力。

关键词: 物聯網, The Constrained Application Protocol, 智慧型感應裝置

1. 引言

國際電信聯盟於2005年提出物聯網The Internet of Things (IOT)概念，物聯網 (IOT)是一組協定和技術，人們可以藉由物聯網(IOT)使通訊設備相互連接[1]–[2]。在現今邁向工業4.0的世代除了人與人可以透過網際網路聯繫，人也可以透過網路取得物件的資訊外，物件與物件之間亦可以透過smart sensors 和網路環境來進行聯繫，物聯網的組成包括：(a) Mobile Control：將透過行動裝置，達到無論何處都能利用網路來隨時隨地的監控管理或控制。(b) Devices：透過雲端化技術將物件之間連結起來，使得實體物件或系統均可透過網路來進行溝通。(c) Notification Event：將控制管理的物件或系統資訊能即時顯示在各種螢幕上如：手機、平板、電視螢幕等。IBM公司在2008年提出“智慧地球”[1]與現行物聯網概念較為相近，並提出透過 Instrumented，Interconnected和 Intelligent三個層面來實現智慧地球。依據國際數據資訊 (IDC) 預估物聯網在製造業應用中的市場產值將從2013年的422億美元到2018年時增加到989億美元，而車用物聯網的商機，根據Transparency Market Research(TMR) 預估，2019年車用物聯網的產值將達1,319億美元，80%出廠的車輛將導入車用智慧物聯[2]。

隨著網際網路的服務應用快速發展，人們透過網路來傳遞訊息及達到資源分享的目的。而主機的運作模式從單機運算、平行計算、分散式運算、網格運算和SaaS(software as a service)等階段，以雲端運算應用服務的模式成為未來資訊技術發展的一個重要方向[19]–[21]，雲端運算具備許多優點，其特點有：1. 虛擬化技術、2. 使用者可靈活選擇服務項目、3. 軟體、資料服務將不在使用者端而在雲端、4. 單機設備尚無法達到的快速運算和儲存能力等。

雲端基本概念如圖1所示，其目標是使用一般操作介面如：電腦、行動裝置等，透過網路，各項服務在雲端如：計算、服務、儲存等，實現人們希望的一切。例如使用者端只需要一些基本的硬體設施，使用內建雲端作業系統，的瀏覽器連上網路，使用者需要的應用服務及軟體都是雲端可以達成的，本身的電腦無須下載及安裝額外的服務應用軟體，因為服務都是在雲端。

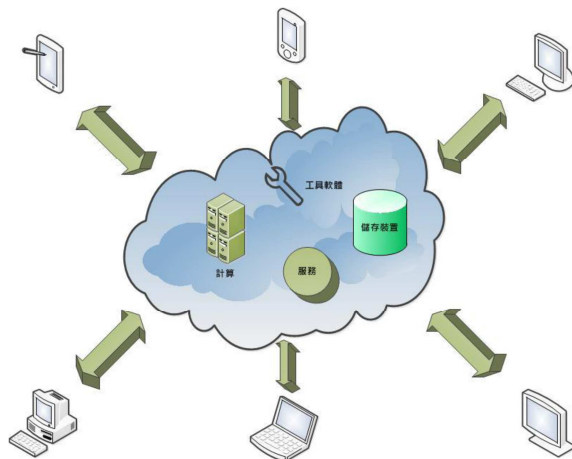


圖1 雲端示意圖。

IETF(Internet Engineering Task Force)的CoRE工作組提出了一種基於REST架構的CoAP協定可以減少HTTP的複雜性。REST的優點為：1. 可更高效利用緩存來提高響應速度。2. 通訊本身的無狀態性可以讓不同的伺服器的處理一系列請求中的不同請求，提高伺服器的擴展性。3. 瀏覽器即可作為客戶端，簡化軟體需求。4. 相對於其他疊加在HTTP協議之上的機制，REST的軟體依賴性更小。5. 不需要額外的資源發現機制。6. 在軟體技術演進中的長期的相容性更好。下一節將針對CoAP 格式及協定進行說明。

2. CoAP 格式及協定

2.1. CoAP 格式

CoAP (Constrained Application Protocol) 是一種物聯網之應用協定，表1為CoAP封包的格式，格式說明如下

表1 CoAP封包格式。

Ver	T	OC	Code	Message ID
Options (Optional) –TLV Formation				
Payload				

- 1. Ver為CoAP的版本。
- 2. T (Type) 表示訊號的類型。
- 3. OC (Option Count) 表示後面的可選擇的選項數量。
- 4. Code表示訊號的類型：請求訊號、回應消息或者是空消息。
- 5. Message ID表示消息編號，用於重複訊號檢測、匹配訊號類型等。

2.2. CoAP protocol

CoAP建立在UDP 之上，而HTTP建立在 TCP之上。圖2顯示CoAP和 HTTP 兩者運用不同的傳輸層。TCP跟UDP的最大差別 [3][15] 在於TCP提供的是一個連線導向(Connection Oriented)的可靠傳輸，相對而言UDP 則是一個非連線型(Connectionless)的非可靠傳輸協定，它並不會運用確認機制來保證資料是否正確的被接收、不需要重傳遺失的資料、資料的接收可不必按順序進行、也不提供回傳機制來控制資料流的速度。因此UDP 信息可能會在網路傳送過程中丟失、重複、或不依順序，而且抵達速度也可能比接收端的處理速度還快，在傳送的封包UDP的封包也比TCP來的小。為了解決UDP非可靠傳輸根據[4]說明CoAP是由兩層架構而來的，包含訊息層面 (Message)以及請求/回應層面 (Request/Response)。首先，訊息層面是負責訊息的傳輸，也提供對多重播送和壅塞控制的支持，此層面包含四種類型，Confirmable(可確認)、Non-Confirmable(無需確認的)，Acknowledgment (確認)、Reset (重新設定)。CoAP的雙層處理方式，使得CoAP沒有使用TCP協議，也可以提供可靠的傳輸機制。

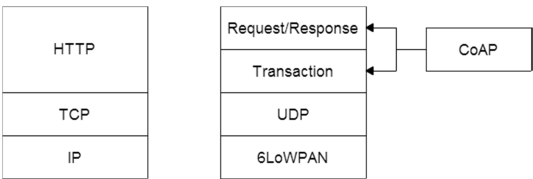


圖2 HTTP跟CoAP協定層。

2.3. CoAP相關研究及應用

Bormann[4] 實作在區域環境及Internet環境下透過HTTP和CoAP傳輸物聯網。CoAP使用Datagram Transport Layer Security (DTLS)協定用於保護資料傳輸的安全，但DTLS 檔頭太長不適用於IPV6環境下低功率無線設備傳輸。Raza[5]提出了在IPV6環境下，壓縮DTLS檔頭可以降低額外的security bits，以適用於低功率無線物聯網設備傳輸。Becker [6]以CoAP使用於無線感測物聯網，達到傳遞溫度及濕度值的目的。IETF RFC7252 [7]訂定CoAP的標準。Colitti[8]提出CoAP與HTTP傳輸協定的比較，由於CoAP對於檔頭進行壓縮可以提供較小的封包，更適合用於低耗能之無線網路端點設備傳輸使用。Moritz [9]結合SOAP與CoAP協定，可以節省43% TCP的雙向交握往返之時間。然而以上這些研究都偏重在IPv6低耗能之無線網路環境下端點對端點的物聯網通訊，並未與雲端系統加以整合或相關之應用。Angelo [10]提出的HTTP-CoAP轉換，以Proxy (代理伺服器)來實現UDP轉HTTP或HTTP轉UDP如圖3所示，此架構是當CoAP Endpoint傳資料給HTTP User Agent時會先把資料丟給HTTP-CoAP Cross proxy從UDP轉成HTTP後再傳給HTTP User Agent，同理當HTTP User Agent傳給CoAP Endpoint時一樣先丟給HTTP-CoAP Cross proxy從HTTP轉成UDP後再傳給CoAP Endpoint。Bergmann[11]提出的智慧家庭應用架構，使用的核心設備為IAD(integrated access device)，它是一個內嵌式系統，本身包含多個模組，其中LAN模組主要是用來與辦公室或家中的網路設備連接，目前絕大部分IAD是以10/100M Ethernet介面與 LAN 相連，在WAN的部分，IAD在與外界連接傳輸的模組是以ADSL/SDSL為首選，傳輸模組在傳輸訊號方面主要是由實體層來處理傳輸訊號，將資料解封裝後交由CPU作進一步處理。CPU模組除了執行程式以管理整套系統外，還要負責通訊協定的轉換處理，資料型式的轉換。可以使用在家庭電氣設備，如電燈，電視機，立體聲音響等系統利用無線WiFi或有線連接到IAD，所有的電器設備的資訊都會透過這台IAD來處理，當使用手機藉由網際網路連接到這台IAD時，IAD會把電器設備的資訊傳送到手機，手機會藉由軟體把IAD所傳送的電器資訊顯示在螢幕上例如電燈的狀態、溫度計的溫度、目前所使用電量等等的資訊。Jin [12] 提出一種 CoAP Service Gateway 在感測網路及Internet之間建立 CoAP service。Chun [13]在無線環境下sensor nodes 跨網段移動時透過所提出行動 CoAP 機制達到管理的目的。Bhalerao [14] 分析及改良在頻寬受限下CoAP Congestion Control的能力。

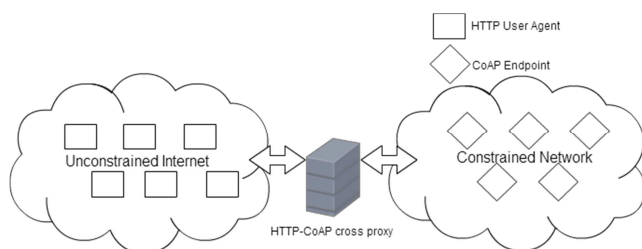


圖3 HTTP-CoAP cross proxy架構。

3. 物聯網與雲端架構實作方法

本研究系統架構透過QRCode將新的設備註冊在CoAP Controller上使用，當註冊完成後會在手機或電腦上顯示該電器設備的資訊，使用者可以選擇CoAP和Cloud 兩種模式來控制電器設備。

3.1. 系統運作架構

系統運作架構如圖4所示，註冊流程1. 手機拍攝QRcode 2. 拍攝完成時設備資訊(設備的資訊包含設備IP位置Port及設備的型號)通過CoAP協定註冊在CoAP Controller 3. 註冊時CoAP Controller會把註冊的設備資訊儲存在資料庫並編譯一組UID(UID使用2Byte單位)，UID會在資料庫對應IP及Port和設備型號然後轉發給Cloud，4. Cloud會把設備資訊儲存在系統上並顯示在使用者所操作的介面上，因此只要傳送一組UID CoAP Controller就會跟據資料庫裡儲存的UID對應表查看設備的位置及型號來控制該設備。

使用者以操作介面傳送訊息到雲端，由於目前雲端僅支援TCP傳輸協定，雲端接收到訊息後以TCP/IP協定傳送訊息給CoAP Controller之後，CoAP Controller轉成CoAP協定後根據傳送的控制資料控制電器，當控制程序完成後會把目前設備的狀態顯示在設備介面，除此之外，在區域內的使用者也可以藉由手機或電腦直接發送CoAP協定給CoAP Controller 後控制電器設備，使用者不必透過雲端來控制，控制完成後CoAP Controller也會把CoAP Client所控制的設備狀態傳送給Cloud後顯示在設備介面。

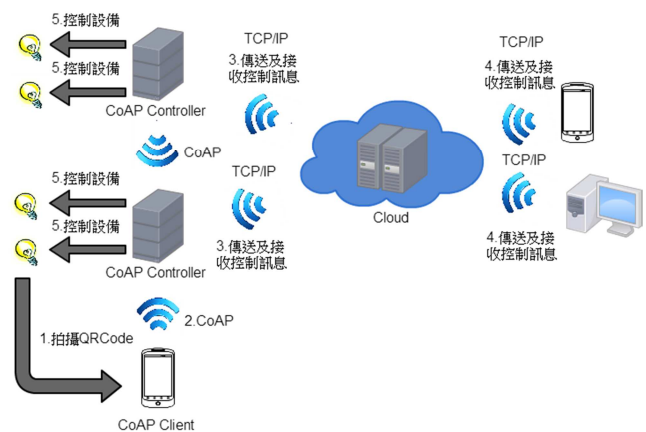


圖4 系統運作架構。

3.2. 雲端架構

在此架構裡Cloud功能有1. 使用智慧型手機或者電腦透過網頁傳送控制訊號給Cloud 2. Cloud接收到控制訊號後會透過TCP/IP協定傳送給CoAP Controller 3. CoAP Controller 接收到控制訊號後會把控制結果回傳給使用者，如果CoAP Client也在控制設備會傳送結果給Cloud 4. 把控制結果傳給設備介面並顯示如圖5所示。



圖5 Cloud架構。

3.3. CoAP Controller架構

如圖6 CoAP Client會先拍攝QRCode來註冊電器設備 (Translation)到手機端，手機端會顯示該設備的狀態及控制該設備的選項，當控制設備選項後會傳送控制資料給CoAP Controller資料裡面有包含UID作為該電器設備的識別碼，CoAP Controller收到CoAP Client發送控制資料及根據UID來選擇UID1、UID2、UID3、UID4的電器設備然後進行控制。

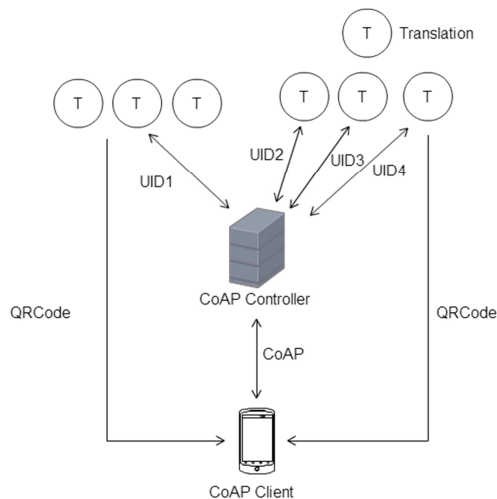


圖6 CoAP Controller架構。

3.4. 雲端、CoAP controller、CoAP clients 訊息傳遞模式

雲端及CoAP controller之間訊息傳遞採用JSON (Java Script Object Notation) [17]格式，JSON格式是一個是一種輕量級的資料交換語言，以純文字為基底去儲存和傳送簡單結構資料，使用者可以透過特定的格式去儲存任何資料(字串，數字，陣列，物件)，也可以透過物件或陣列來傳送較複雜的資料，一般Web應用程式中，通常不需要用到XML的複雜階層性，此時採用JSON作為資料交換為現行常用的方式。

本計畫雲端、CoAP controller、CoAP clients 訊息傳遞模式如圖7所示，CoAP clients 和 CoAP controller 間採用CoAP協定傳輸訊息，雲端及CoAP controller 間則採用JSON作為資料交換。

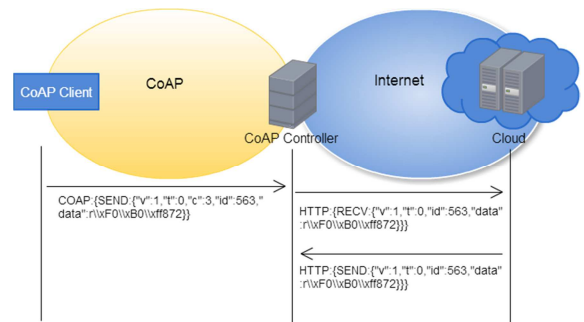


圖7 雲端、CoAP controller、CoAP clients 訊息傳遞模式。

4. 實驗環境與測試結果

4.1. 實驗環境

實作方面我們使用樹梅派(Raspberry Pi) [18]做為CoAP controller開發平台，樹梅派是基於Linux系統所開發的單機板電腦，是由英國的樹莓派基金會所開發的電腦，作業系統支援GNU/Linux (Debian, Fedora, 和Arch Linux ARM), RISC OS, FreeBSD, Plan 9, 記憶體分別為A型256MByte、B型516MByte儲存空間使用SD卡當作儲存媒體，CPU使用700MHz博通出產的ARM架構BCM2835處理器，圖8是樹梅派的實體機器。



圖8 CoAP Controller。

4.2. 實驗環境

圖9顯示由CoAP Client 發送封包到CoAP Controller的執行狀況，-m put是指推送封包-e 872是指發送的資料(data)後面CoAP是指定IP及61616的port。

```

root@localhost:~/桌面/libcoap-4.0.3/examples
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(I) 求助(H)
-N send NON-confirmable message
-o file output received data to this file (use '-' for STDOUT)
-p port listen on specified port
-s duration subscribe for given duration [s]
-v num verbosity level (default: 3)
-O num,text add option num with contents text to request
-P addr[:port] use proxy (automatically adds Proxy-Uri option to request)
-T token include specified token

examples:
coap-client -m get coap://[::1]/
coap-client -m get coap://[::1]/well-known/core
coap-client -m get -T cafe coap://[::1]/time
echo 1000 | coap-client -m put -T cafe coap://[::1]/time -f -
[root@localhost examples]# ./coap-client -m put -e 872 coap://[192.168.108.86]:61616
v:1 t:t0 tk:l:0 c:3 id:1808
5.04
[root@localhost examples]# ./coap-client -m put -e 872 coap://[192.168.108.86]:61616
v:1 t:t0 tk:l:0 c:3 id:2820
5.04
[root@localhost examples]#

```

圖9 CoAP Client 發送到CoAP Controller封包。

表2 CoAP和TCP/IP傳輸平均時間。

協定	CoAP	TCP/IP
傳輸時間	0.24	0.28
傳輸時間	0.26	0.32
傳輸時間	0.27	0.23
傳輸時間	0.3	0.26
傳輸時間	0.29	0.24
傳輸時間	0.24	0.22
傳輸時間	0.27	0.28
傳輸時間	0.25	0.27
傳輸時間	0.29	0.27
平均時間	0.267	0.26

表2的測試環境是在同一網段傳送及接收9次的平均時間，其中CoAP協定是指由CoAP Client傳送到CoAP Controller的平均時間，TCP/IP是由雲端傳送到CoAP Controller的平均時間，雖然2個協定的傳送的平均時間幾乎相同，但在文獻探討中有提到Walter Colitti[16]這位學者測試CoAP、TCP/IP的消耗功率比較，在CoAP協定是比TCP/IP所來的低。

以樹梅派做為CoAP Controller來控制LED燈，當CoAP Client或雲端傳送UID:872給CoAP Controller，CoAP Controller依據接收到的UID來控制LED燈的開關或使LED燈閃爍如圖10。

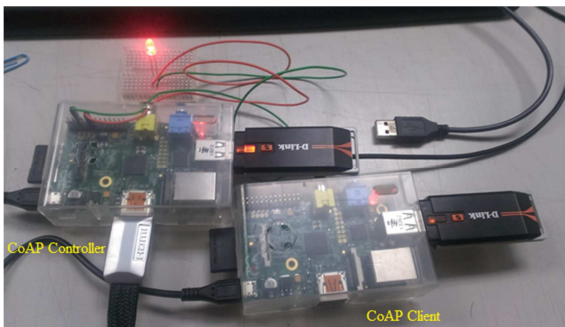


圖10 CoAP Controller控制LED燈閃爍。

5. 結束語

本論文以樹梅派實作出CoAP Controller 接收來自雲端命令及CoAP clients，CoAP clients 和 CoAP controller 間採用CoAP協定傳輸訊息，雲端及CoAP controller 間則採用JSON作為資料交換，物聯網裝置QRCode將新的設備註冊在CoAP Controller上使用，當註冊完成後會在手機或電腦上顯示該電器設備的資訊，使用者可以選擇CoAP和Cloud 兩種模式來控制電器設備。

参考文献

- [1] 物 联 网 ： 开 拓 未 来 的 蓝 海, http://www-31.ibm.com/cn/services/bcs/iibv/pdf/internet_of_things_0117_cn.pdf.
- [2] 台灣物聯網聯盟, <http://www.tiota.org.tw/>.

- [3] TCP and UDP, http://www.pcnet.idv.tw/pcnet/network/network_ip_tcp.htm.
- [4] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," IEEE Internet Computing, Vol. 16, pp. 62-67, April 2012.
- [5] S. Raza, D. Tralbalza, and T. Voigt, "6LoWPAN Compressed DTLS for CoAP," in IEEE Conference on Distributed Computing in Sensor Systems, pp. 287-289, May 2012.
- [6] M. Becker, T. Potsch, K. Kuladinithi, and C. Goerg, "The Constrained Application Protocol for Pervasive Machine-to-Machine Communications," in IEEE Workshop on Pervasive Computing and Communications, pp. 483-485, March 2012.
- [7] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF RFC 7252, June 2014.
- [8] W. Colitti, K. Steenhaut, N. D. Care, B. Buta, and V. Dobrota, "Evaluation of Constrained Application Protocol for Wireless Sensor Networks," in IEEE Workshop on Local & Metropolitan Area Networks, pp. 1-6, Oct. 2011.
- [9] G. Moritz, F. Golatowski, and D. Timmermann, "A Lightweight SOAP over CoAP Transport Binding for Resource Constraint Networks," in IEEE Inter. Conference on Mobile Ad-Hoc and Sensor Systems, pp. 861-866, Oct. 2011.
- [10] A. P. Castellani, T. Fossatiz, and S. Loreto, "HTTP-CoAP Cross Protocol Proxy: An Implementation Viewpoint" in IEEE 9th International Conference on Mobile Adhoc and Sensor Systems (MASS), pp. 1-6, Oct. 2012.
- [11] O. Bergmann, K. T. Hillmann, and S. Gerdes, "A CoAP-Gateway for Smart Homes" in International Conference on Computing, Networking and Communications (ICNC), pp. 446-450, Feb. 2012.
- [12] X. Jin, K. Hur, S. Chun, M. Kim "Automated Mashup of CoAP Services on the Internet of Things" in IEEE 2nd World Forum on Internet of Things (WF-IOT), pp. 262-267, Dec. 2015.
- [13] S. M. Chun, J. T. Park "Mobile CoAP for IoT Mobility Management" in IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 283-289, Jan. 2015.
- [14] R. Bhalerao, S. S. Subramanian, J. Pasquale "An analysis and improvement of congestion control in the CoAP Internet-of-Things protocol" in IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 889-894, Jan. 2016.

- [15] M. Ruta, F. Scioscia, A. Pinto, E. Di Sciascio, F. Gramegna, S. Ieva, G. Loseto “Resource annotation, dissemination and discovery in the Semantic Web of Things: a CoAP-based framework” in Green Computing and Communications (GreenCom), IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, pp.527-534, Oct.2013.
- [16] W. Colitti, K. Steenhaut, N. D. Care, B. Buta, and V. Dobrota, “Evaluation of Constrained Application Protocol for Wireless Sensor Networks,” in IEEE Workshop on Local & Metropolitan Area Networks, pp.1-6, Oct. 2011.
- [17] JSON, <http://zh.wikipedia.org/wiki/JSON>.
- [18] 树莓派, <http://zh.wikipedia.org/wiki/树莓派>.
- [19] Cloud Computing, <http://eblog.cisanet.org.tw/post/Cloud-Computing.aspx>.
- [20] Windows Azure 平台開發技術, http://www.runpc.com.tw/content/cloud_content.aspx?id=105317.
- [21] Vmware, <http://www.vmware.com>.
- [22] 分散式運算把電腦變大、拉近科學距離, http://www.richyli.com/report/20050220_DistributedComputing.htm.
- [23] Cloud Computing, http://www.cc.ntu.edu.tw/chinese/epaper/0008/20090320_8008.htm.
- [24] 虛擬化, <http://www.microsoft.com/taiwan/virtualization/cloud-computing/default.msp>.