

# 以 CoAP 為基礎的輕量級智慧病房即時資訊系統

## Design and Implementation of a CoAP-based Light-weight Smart Ward Information System

陳竣彥<sup>1,2</sup> 廖峻鋒<sup>2</sup>

財團法人長庚紀念醫院醫療資訊管理部<sup>1</sup> 國立政治大學資訊科學系<sup>2</sup>

Chun-Yen Chen and Chun-Feng Liao

Department of Information Management Chang Gung Memorial Hospital<sup>1</sup>

Department of Computer Science, National Chengchi University<sup>2</sup>

{104971025,cfliao}@nccu.edu.tw

### 摘要

物聯網應用服務一般來說是建構於一個分散式且包含許多異質裝置的環境下。其中，服務的提供是透過機器與機器的資訊交流技術來達成，任何裝置都可以當成感測器節點，並透過無線傳輸技術來傳送資料並進行監測。本研究將以藍芽低功耗傳輸技術與 CoAP 物聯網通訊協定為基礎，設計一套輕量級智慧病房即時資訊系統，並據以提出一個具擴充性的智慧病房即時資訊系統架構，透過此架構，感測節點可建立連動機制，例如蒐集病房點滴或血袋重量來主動通知護理人員目前點滴狀態。此外，使用資料可視覺化網頁，提供護理人員能主動更換點滴、建立更安全的醫療環境，達到智慧病房的功效。

**關鍵字：**物聯網、智慧病房、普適醫療保健、CoAP。

## 1 前言

由於現今的生活品質大幅提升，加上目前趨近於少子化和人口老化的社會。導致越來越多人罹患文明病，進出醫院的人也日與俱增，所以醫院佔床率一直居高不下。為了有更好的醫療品質，站在第一線的護理人員就顯得格外重要。傳統護理人員值班方式是頻繁到各個病床進行巡查，並逐一檢查病人狀況且利用填寫紀錄表來做紀錄，事實上因為護理人員人手短缺，這種傳統方式不僅讓護理人員增添許多工作量，也間接增加體力和精神力的消耗，加上工作時數長、休假時間少、責任制度等原因，長時間累積的疲勞之下，導致越來越多護理人員不堪負荷，這樣的惡性循環下，台灣的醫療品質只會更差，病人也沒辦法接受較好的醫療照顧。

隨著硬體的成本降低、效能的提升、以及體積的縮小，越來越多的感測器被開發出來，因此，近年來物聯網技術各領域都出現許多研究與應用。在本研究中，為了減輕護理人員的工作負荷，我們透過物聯網的應用來設計開發一套以 CoAP 為基礎的輕量級智慧病房即時資訊系統，使護理人員能隨時在護理站監測目前各病房點滴容量，而不用耗費人力去巡查病房點滴。便可在點滴用完之前，可馬上至病房更換點滴。提升工作效率，減輕工作負荷及提升醫療服務品質及病人對醫院的科技信任感。醫院病房的設計必須符合許多安全規定，其中包含了各病房的總用電量與電器佈署的限制。因此，目前物聯網在智慧病房應用最需要解決的問題便是省電與資料收集的問題，其中，由於藍芽低功耗傳輸技術的推出，使得利用藍芽低功耗傳輸的裝置使用時間大幅增加。CoAP 協定是物聯網中資源受限設備制定的應用層協定，透過 CoAP 協定在管理感測器和節省電力和傳輸資源是最有效的協定之一。因此，本研究的目標即為減少護理人員的工作量，我們以藍芽低功耗傳輸技術與物聯網 CoAP 協定為主軸，設計出一套「以 CoAP 為基礎的輕量級智慧病房即時資訊系統」，希望能藉由這套系統來減少護理人員的工作壓力和負荷，用系統來監測點滴並主動通知，使護理人員在護理站就能輕易掌握各病房點滴狀態，以降低護理人員工作負荷並提升工作效率，進而提升整體醫療品質為目標。

## 2 相關研究

由於在資訊和通信技術的最新發展，有許多種物聯網基於感測器的系統和設備被部署在城市規模，甚至國家。它代表了一個很好的機會來支持日常生活活動。這樣的智慧環境實際上是促進智慧城市，可以支持其居民的活動，以提高生活質量，並確保在許多領域，如醫療，運輸，體育和娛樂，專業和社會生活的可持

\* 本論文之部份研究受科技部計畫經費補助，編號：104-2221-E-004-001，104-3115-E-004 -001 與 104-2627-E-002-006

續性 [1]。醫療保健系統是非常重要的，因為它注重人文關懷與人類生活相關。近年來，我們關注電子醫療技術：如電子健康記錄和緊急檢測和回應的重要性迅速上升 [2]。無線自身區域感測器網路 (WBASNs) 技術的出現，讓慢性病患者的健康狀況和生理數據可自動和遠端收集 [3]。

無線和普通醫療保健應用程序通常從功能的正確性，可靠性，可用性和安全性的角度來看目前的關鍵要求。在對比的情況下，無線和普通醫療保健的行為是受使用者和資源及位置的影響 [4]。使用醫療保健監測感測器和無線感測器網路是醫療監控應用迅速增長的領域。為了成功地監測病人而有許多討論議題，如複雜的分佈式數據處理，數據融合，不可靠的數據通信，以及數據分析的不確定性問題 [5]。大量的環境輔助技術的共存，他們之間沒有共同語言。他們有不同的用戶界面，通信基礎設施和維護計劃。此外，應用程序和設備部署過於複雜，並不能由醫務人員或公民本身進行處理，所以經常需要技術人員的支援。進而增加成本，降低了採用新技術的可行性 [6]。

一個物聯網系統通常在資源約束的環境，其中感測環境屬性和傳輸感測設備所檢測到的數據傳輸到後端基礎設施。使用在物聯網上的感測器和感測閘道器通常是受限制的設備。因此，它仍然是進行通信在頻寬和效能上的挑戰且安全的感測器數據方式 [7]。約束應用協定 (CoAP) 是由 IETF 標準化的一個新的 Web 協定。這不是由 HTTP 的一個單純的壓縮，是由從上而下的 REST 架構重新設計。因此，它的功能是專為與高度資源受限設備 (IoT) 的應用程序和機器對機器 (M2M) 方案的網路。它仍然是由瀏覽器和直觀的用戶互動的網路世界脫節 [8]。在網路壅塞可能導致顯著性能下降和給定的資源受限設備及低功率無線電的限制和限制硬碟容量是一個經常性現象。設計的 IoT 通信約束應用協定 (CoAP) 定義為具有兩個端點之間的端對端可靠的訊息交換壅塞控制 [9]。在醫療系統中的一個主要方面是病人的生命體徵，如溫度監測，血壓和心臟跳動速率。不少監控設備即顯示患者的生命體徵是通常存在於該重症監護病房手術室 [10]。

根據上述的相關研究，我們可以得知使用無線網路傳輸技術搭配感測器來做普通醫療保健監測系統的基礎設施是一種趨勢。在本研究中，選擇低功耗藍芽傳輸技術在普通醫療保健監測系統是最適合的無線傳輸協定，因為低功耗可以延長感測器的能源使用量，加上病房通常會有許多醫療監測儀器，而藍芽低功耗傳輸技術相對其他無線傳輸技術來說是最安全且最節省資源的技術。而在物聯網受資源限制的環境中，使用 CoAP 協定來做資料交換是在有限頻寬的環境中和本身的效能是相當不錯的，可解決在資源受限環境網路壅塞的問題。最後，在物聯網許多物物相聯的環境中，要有一套可通用且跨平台的架構是相當困難且複

雜的，通常不同感測器及不同微型電腦上會使用不同的程式語言來開發，因此本研究提出的架構只使用一種程式語言和開發平台可使用在不同種類的感測器和微型電腦及任何無線傳輸技術上，可節省許多人力和開發成本，進而達到本研究的最大效益。

### 3 系統架構與設計

這一節我們介紹本文所提出的智慧病房即時資訊系統的設計架構，藉由整合感測器與微型電腦，並透過資料庫來儲存感測器所接收的資料，整體系統架構如圖 1 所示。Sensor Module 會透過藍芽低功耗傳輸技術將感測器量測資料傳輸至 CoAP Server，CoAP Server 接收資料時會將 JSON 格式資料儲存至 NoSQL 的 MongoDB 當作 Log 資料庫，接著護理站伺服器會透過 CoAP Client 送出 Request 向 CoAP Server 要求感測器接收到的資料，並儲存至護理站伺服器的 MySQL 關聯式資料庫，最後，透過 D3.js 資料可視化技術將感測器蒐集到資料，用網頁方式呈現給護理人員監測各病房的感測器狀態。以點滴感測應用為例，首先須將 TAL220 負重 (直桿) 結合 HX711 重量感測放大器模組當作主要的點滴量測感測器，接上本系統後，感測資料就會透過 BLE 送達 CoAP Server，透過 CoAP Client 向 CoAP Server 取得資料後，存在資料庫中再透過 Web 介面供護理人員監測。

整體系統架構如圖 1，其主要包含三個部分：

- Sensor Module：以上述 Sensor Module 中的點滴感測應用為例，為了可以量測病房內所有輸液類 (例如：食鹽水、血袋)。故感測器採用 TAL220 荷重元元件和 HX711 訊號放大器並透過藍芽低功耗技術來傳輸資料，降低高功耗會影響病房內其他醫療設備，及大幅增加感測器能源使用時間。且大幅提升點滴感測器對病患的安全。本架構的設計並不限於點滴感測器，基於本架構下，可隨時增加其他感測器模組。
- CoAP Server (Raspberry Pi 微型電腦)：CoAP Server 主要接收感測器透過藍芽低功耗傳輸的資料，並儲存於 NoSQL 資料庫的 MongoDB，當作 Log 資料庫，並透過物聯網最廣為使用的 CoAP 協定將 JSON (JavaScript Object Notation) 格式資料回應給 CoAP Client，使傳輸資料為最佳輕量化。經由 CoAP 協定讓 CoAP Server 能對每個感測器定義唯一 URI，讓護理站伺服器能透過 CoAP Request 向 CoAP Server 要每個病房的點滴量測資料。在 Raspberry Pi 微型電腦上也是使用 Node.js V8 引擎平台搭配 JavaScript 語言開發，可隨時彈性增加並接收其他種類感測器模組所傳輸的資料。

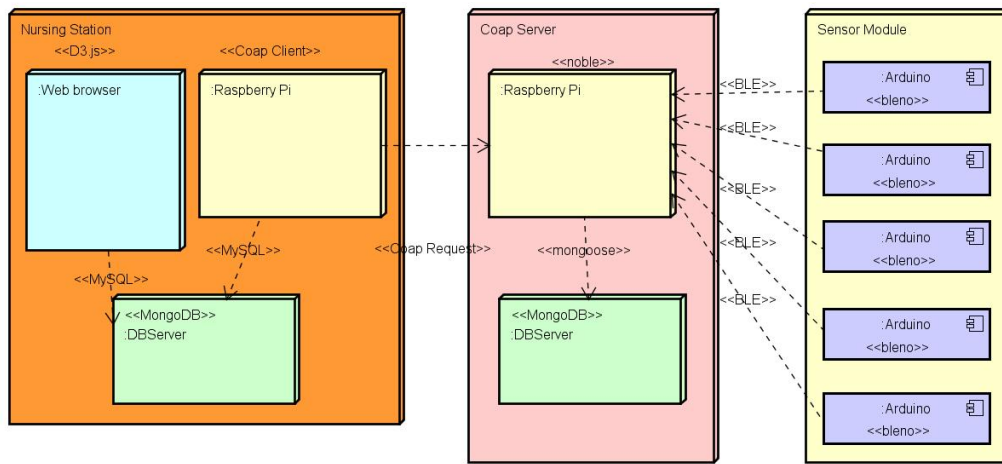


圖 1: 系統架構圖

- Nursing Station: 從護理站伺服器向 CoAP Server 發出 CoAP Request 要求感測器資料後，護理站伺服器會接收到 CoAP Server 所發送出的 JSON 格式資料，每接收到一筆資料就將資料儲存至 MySQL 關聯式資料庫以利後續資料可視覺化及研究分析使用。使用者介面則是透過 D3.js 可視覺化資料網頁，將接收到的資料呈現給護理站監測每樓層所有病房的目前點滴量測狀態。使用者介面是使用 Node.js V8 引擎平台搭配 JavaScript 語言及 D3.js 開發。

以上三個部分是用開放原始碼且跨平台的 Node.js 平台搭配 JavaScript 語言開發，使得程式可以作為獨立伺服器執行。達到彈性佳且容易擴充的智慧病房物聯網應用架構。

### 3.1 感測器模組

系統之藍芽低功耗傳輸技術是使用 BLE Shield 2.1 藍芽低功耗模組，其製造商 RedBearLab 是國際知名 Maker 開發套件大廠，擁有完整軟體 SDK 與 Arduino 韌體。RedBearLab BLE Shield 目前除了 BLE Arduino 之外，其餘的 iOS SDK 與 Arduino Library 都是開源的，其 nRF8001 藍芽晶片 SPI 控制介面 3.3V or 5V，SPI 通訊可選 ICSP 或 Pins 11-13，使 Arduino Uno 使用藍芽低功耗能有長時間待機。本研究主要將 BLE Shield 2.1 藍芽低功耗模組結合 Arduino Uno R3 開發版，透過 JavaScript 語言在 Node.js 平台上開發藍芽低功耗傳輸程式，便可使用藍芽低功耗技術來傳輸感測器所讀取到的數值。此部分藍芽低功耗傳輸技術發送端是使用 GitHub 上的開放原始碼專案 bleno [12]，而接收端則是採用開放原始碼專案 noble [13] 來實作本系統之藍芽低功耗傳輸技術。

### 3.2 CoAP 閘道器 (CoAP Server)

系統之 CoAP Server 建構在 Raspberry Pi 的 Raspbian 作業系統之上，Raspbian 是基於 Debian 的計算機操作系統，屬於 Linux 作業系統，也是開放原始碼的作業系統。可看出它是一套擁有廣大社群維護的作業系統，並且擁有許多免費應用軟體支援，對連結於網路的系統而言，安全性與網路應用能力較嵌入式系統來的高。Raspbian 作業系統搭配 Node.js 開發平台可以開發出具彈性的物聯網應用架構且達到跨平台的目標。由圖??亦可發現，CoAP Server 與 Nursing Station 上均配置了不同種類的資料庫管理系統。在 CoAP Server 下，採用的是市佔率最高的 NoSQL 資料庫 MongoDB 和最受 Node.js 喜愛的關聯式資料庫 MySQL 資料庫，兩者都屬於開放原始碼免費應用的資料庫，非常適合本系統架構所使用。MongoDB 是一種文件導向資料庫管理系統，由 C++ 撰寫而成，以此來解決應用程式開發社群中的大量現實問題。

本系統之 Nursing Station 所使用的 MySQL 關聯式資料庫在過去由於效能高、成本低、可靠性好，已經成為最流行的開源資料庫，因此被廣泛地應用在 Internet 上的中小型網站中。隨著 MySQL 的不斷成熟，它也逐漸用於更多大規模網站和應用，比如維基百科、Google 和 Facebook 等網站。非常流行的開源軟體組合 LAMP 中的「M」指的就是 MySQL。資料庫強大的儲存與查詢能力，可以儲存感測器相關資料 (例如：感測器所屬院區、所屬病房、感測器蒐集到的點滴量測資料)，連結護理站伺服器可即時呈現給護理人員監測點滴狀態。

### 3.3 使用者介面

我們以圖 2 來表達 UI 介面架構規劃，使用者介面主要依地區劃分，可分為四個層次。第一層為院區，第二層為大樓，第三層為樓層的病房號碼，最後一層是感

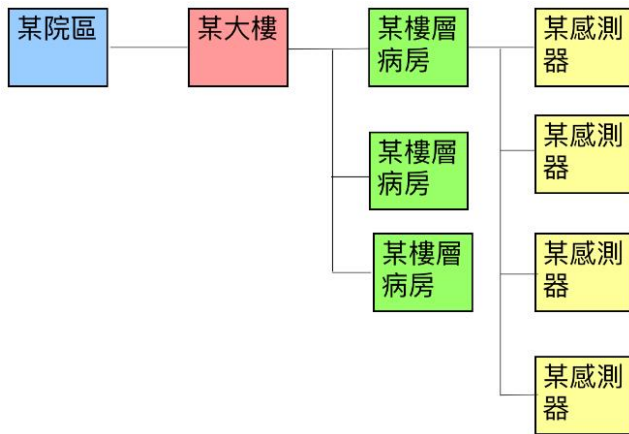


圖 2: 使用者介面結構圖

測器的名稱。系統會自動蒐集透過藍芽技術傳輸的感測器傳出的資料，護理人員可透過護理站伺服器的網頁看到目前樓層的所有病房內所有感測器目前狀態。

### 3.4 系統運作

圖 3顯示了系統的運作架構。首先，系統先啟動病房內的點滴感測器發送端程式，此程式會開始等待藍芽接收端程式啟動並回應連線成功或連線失敗訊息。如果連線失敗代表接收端找不到發送端感測器的 uuid，則必須重新啟動接收端程式來尋找正確相對應感測器的 uuid。若建立連線成功，則每 200 毫秒時，點滴感測器從序列埠讀出來的值透過發送端程式並使用 Notify 功能，將值透過藍芽低功耗傳輸給接收端程式，如果前一次和當次的傳的值相同時就不傳輸，所以當讀取到的數值有改變時，才將感測器接收的值傳出。這個設計是要節省不必要的傳輸浪費，以節省感測器電力和 Log 資料庫的儲存空間。接著，CoAP Server 會將接收的值先儲存到 Log 資料庫，再將值透過 CoAP 協定發出 Response。而護理站有 CoAP Client 程式每秒向 CoAP Server 來請求 JSON 格式資料，若 CoAP Server 有發出 Response 時，則 CoAP Client 才會接收到 JSON 資料，並儲存至 MySQL 資料庫，最後提供可視覺化網頁給護理人員監測。

## 4 系統評估

### 4.1 系統實作與功能驗證

圖 4表達了系統的資料流，根據傳統原本護理人員去巡病房且利用填寫紀錄表來做紀錄的流程針對改善，利用系統活動圖來改善原本的作業流程，讓原本被動的巡病房作業流程，改善為主動通知護理站，讓護理人員可以減少工作負荷。本研究從感測器到藍芽傳輸技術和 CoAP 協定的實作及可視覺化點滴監測網頁皆是在 Node.js V8 引擎開發平台並使用 JavaScript

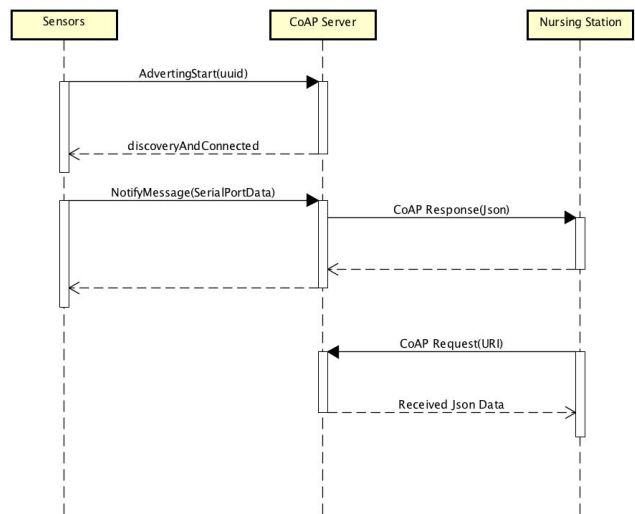


圖 3: 系統運作架構

語言來實踐數據即時性與環境應用功能，後端則使用 NoSQL 資料庫 MongoDB 和 MySQL 關聯式資料庫。而 Arduino 程式的撰寫是使用 Open-Source Arduino Software(IDE)，瀏覽器則是用 Chrome 具有最佳的效果。

#### 4.1.1 智慧病房即時資訊系統

我們以圖 5的 Sensor Module 來介紹所有用到的感測元件，Sensor Module 是由 Arduino Uno R3 開發板加上 BLE Shield 2.1 藍芽低功耗模組來連接各式感測器。以智慧點滴系統為例，感測器採用了 TAL220 負重感測模組（直桿），又稱應變儀，主要是在測量負重的變化轉換為電子訊號。最大承受力為 10Kg，由於 TAL220 負重感測模組需要搭配 HX711 訊號放大器模組並透過撰寫 Arduino 程式將目前感測到重量的數值轉換成毫升單位。將 TAL220 負重感測模組（直桿）加上 HX711 重量感測放大器模組，是使用 HX711 晶片將你讀取的重量透過放大器傳送到控制器，讓你可以讀取到很精確的電阻值變化，讓後續程式邏輯的編寫較為容易來達到目的。所以本感測器採用 TAL220 負重感測模組（直桿）結合 HX711 重量感測放大器模組，將目前感測到重量的數值轉換成公克單位。Sensor Module 從序列埠將讀取值傳出是使用 GitHub 上的開放原始碼專案 node-serialport 來實作 [11]。TAL220 負重感測模組（直桿）結合 HX711 重量感測放大器模組，將所感測到的值由序列埠傳出至微型電腦，因撰寫 Arduino 韌體從序列埠將資料由藍芽低功耗技術傳輸給 Raspberry Pi 3 微型電腦，在此我們皆使用 Node.js 平台搭配 JavaScript 語言來完成撰寫軟體，更容易進行操作及後續資料處理且 JavaScript 語言支援 CoAP 協定。

我們以圖 6的 CoAP Server 來介紹所使用的微型電



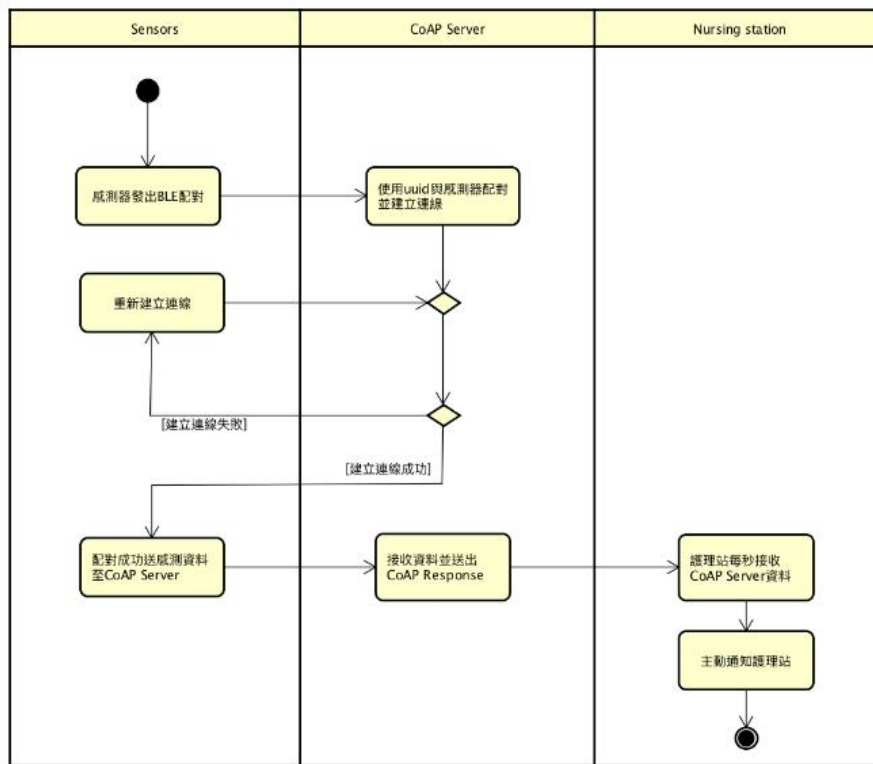


圖 4: 資料流

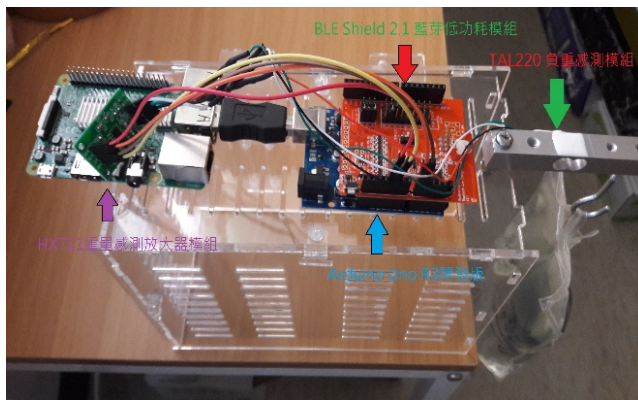


圖 5: Sensor Module



圖 6: CoAP Server 以 Raspberry Pi 結合 BLE 實作

腦及其架構，在 Raspberry Pi 3 微型電腦上，使用 NoSQL 資料庫 MongoDB 記錄每個感測器透過藍芽低功耗技術傳輸過來的資料並儲存在 NoSQL 資料庫當作傳輸過來的 Log 資料庫，因 Raspberry Pi 3 微型電腦本身的儲存空間不大，加上一個護理站只會有一個 Raspberry Pi 3 微型電腦來接收此樓層所有的點滴感測器。所以我們將會安排每天做 NoSQL 資料庫的批次轉檔備份，讓微型電腦上的空間可供每天所有感測器的所有資料儲存，也確保資料庫不會因空間不足而無法運作。接著，Raspberry Pi 3 微型電腦是透過 CoAP 協定將資料回應給 CoAP Client，而 CoAP Client 會對 CoAP Server 發出 Request 請求，所以護理站伺服器會對此樓層的 CoAP Server 發出 CoAP Request 請求，當 CoAP Server 一接收到點滴感測器所傳輸過來的資料時，會先寫入資料至 NoSQL 資料庫並發出 CoAP Response 給 CoAP Client。

最後，護理站上的 CoAP Client 向 Raspberry Pi 3 微型電腦上的 CoAP Server 發出 Request 請求並接收到資料後，馬上將接收到的 JSON 格式資料整理並寫入 MySQL 關聯式資料庫，提供護理站的可視化點滴監測網頁系統做最即時的查詢，讓護理人員可即時對病患更換點滴，不只能提升醫療品質且減輕護理人員工作負擔，更能達成智慧醫院結合智慧醫療的物聯網科技。

我們以圖 7 的可視化點滴監測網頁系統來提供給護理人員在護理站的監測活動，系統主要將自動蒐集

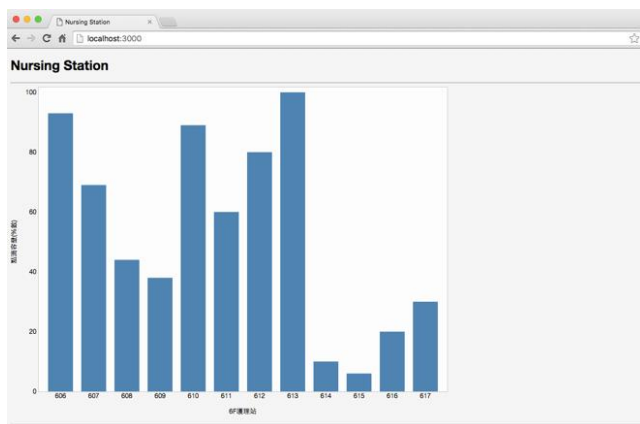


圖 7: 可視覺化監測系統

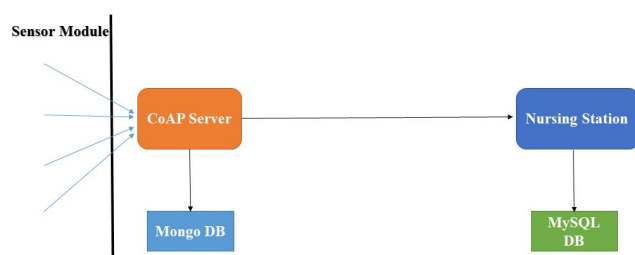


圖 8: 實驗設置

感測器的所有資料，將最新蒐集到的資料即時在護理站伺服器電腦上呈現，提供給護理人員判斷資訊，讓護理人員工作更有效率進而達到提高病患醫療品質。

## 4.2 效能實驗

我們用實際傳輸的方法使用每秒傳輸 5、10、15、20 個訊息來記錄實際傳輸的總時間和總傳輸量 (Bytes)，並透過在 CoAP Server 和 Nursing Station 上兩個資料庫所儲存的時間差來做計算。圖 8 的資料傳輸流程圖顯示了從 Sensor Module 傳資料到 CoAP Server，再從 CoAP Server 傳輸至 Nursing Station 的流程。而我們可以從圖 9 的資料傳輸總時間圖，可看出呈現線性

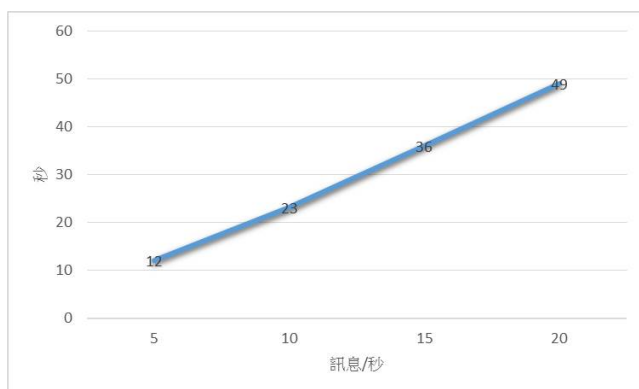


圖 9: 資料傳輸總時間圖

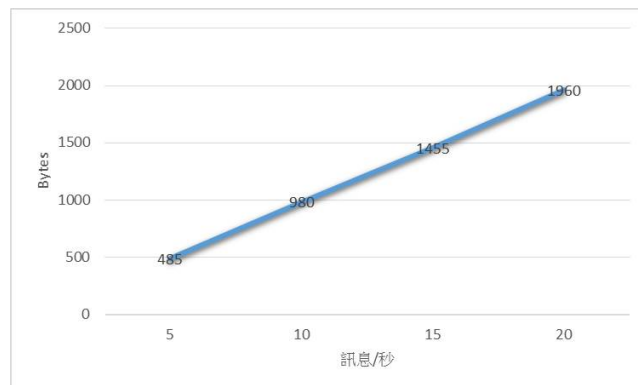


圖 10: 資料傳輸總量圖

成長的趨勢。接著，我們將透過藍芽低功耗傳輸技術和 CoAP 協定傳輸的封包，用 Wireshark 工具來看兩個使用不同的傳輸技術和協定各使用多少封包數來傳輸。圖 11 為藍芽低功耗傳輸封包，是使用 Wireshark 工具來觀察透過藍芽低功耗傳輸技術的封包紀錄。因傳輸的值只有感測器從序列埠偵測到的重量值，所以傳輸封包為 8 bytes。圖 12 為 CoAP 協定傳輸封包，透過 CoAP 協定將 JSON 格式資料傳輸給 Nursing Station 所使用的封包紀錄。因傳輸的 JSON 格式字串有 6 個欄位資料，所以傳輸封包為 97 bytes。我們可以由圖 10 的資料傳輸總量表來看出以 CoAP 協定的輕量級傳輸資料量相對傳統的系統架構上的傳輸是節省傳輸資源的，非常適合使用在物聯網。

## 5 結論

在本篇論文中，我們提出了一個具有彈性且智慧的物聯網應用架構，可以實際應用蒐集病房的點滴資料，提供護理站可以即時監測。本研究提供跨平台的物聯網系統，不管是任何作業系統，只要有支援 Node.js 的開發平台都可以在上面運行。且除了本篇論文所使用的藍芽低功耗技術，其他像 Wi-Fi 傳輸技術也可以在本篇論文的物聯網應用架構上實作。因醫院病房使用高功耗電子產品，可能會影響其他各種醫療儀器的運作，所以本研究所採用的藍芽低功耗傳輸技術是醫院病房最適合使用的傳輸技術。加上 CoAP 協定可提供每個院區的每棟樓層的每個病房的每個感測器都提供唯一的 URI 識別，讓每層樓的護理站伺服器可對 CoAP Server 請求每個感測器所傳輸的資料。讓 CoAP Server 和護理站伺服器間可以方便管理每個感測器。達到萬物皆可聯網的目標，提供方便，快速，好管理的物聯網應用架構。在實際的應用中，我們可以預期護理人員不用再被動的巡視病房，而是透過在護理站的可視覺化網頁來主動監測病房點滴的狀態，讓病患享受到更好的醫療服務品質。最後，本篇論文的物聯網應用架構，在不久的將來，將會提供更多不

同種類的感測器來增加病房的其他應用服務。例如：智慧尿布。提供給病患更安全，更有效率的就醫環境與品質。

## 參考文獻

- [1] M. M. Hassan, H. S. Albakr, and H. Al-Dossari, “A cloud-assisted internet of things framework for pervasive healthcare in smart city environment,” in *Proceedings of the 1st International Workshop on Emerging Multimedia Applications and Services for Smart Cities*. ACM, 2014, pp. 9–13.
- [2] S. Poorejbari and H. Vahdat-Nejad, “An introduction to cloud-based pervasive healthcare systems,” in *Proceedings of the 3rd International Conference on Context-Aware Systems and Applications*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 173–178.
- [3] S. Nikolidakis, V. Giotsas, D. Vergados, and C. Douligeris, “A mobile healthcare system using ims and the hl7 framework,” in *Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*. ACM, 2009, pp. 187–188.
- [4] A. Coronato and G. D. Pietro, “Formal specification of wireless and pervasive healthcare applications,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 10, no. 1, p. 12, 2010.
- [5] H. Lee, K. Park, B. Lee, J. Choi, and R. Elmasri, “Issues in data fusion for healthcare monitoring,” in *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2008, p. 3.
- [6] S. Wagner, F. O. Hansen, C. F. Pedersen, M. Mathissen, and C. Nielsen, “Carestore platform for seamless deployment of ambient assisted living applications and devices,” in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on*. IEEE, 2013, pp. 240–243.
- [7] A. Ukil, S. Bandyopadhyay, A. Bhattacharyya, and A. Pal, “Lightweight security scheme for vehicle tracking system using coap,” in *Proceedings of the International Workshop on Adaptive Security*. ACM, 2013, p. 3.
- [8] M. Kovatsch, “Coap for the web of things: from tiny resource-constrained devices to the web browser,” in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 1495–1504.
- [9] A. Betzler, C. Gomez, and I. Demirkol, “Evaluation of advanced congestion control mechanisms for unreliable coap communications,” in *Proceedings of the 12th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. ACM, 2015, pp. 63–70.
- [10] M. Gupta, V. Patchava, and V. Menezes, “Healthcare based on iot using raspberry pi,” in *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on*. IEEE, 2015, pp. 796–799.
- [11] “node-serialport:a node.js package to access serial ports for reading and writing or welcome your robotic javascript overlords.” (Date last accessed 16-May-2016). [Online]. Available: <https://github.com/voodootikigod/node-serialport>
- [12] “A node.js module for implementing ble (bluetooth low energy) peripherals,” (Date last accessed 16-May-2016). [Online]. Available: <https://github.com/sandeepmistry/ble>
- [13] “A node.js ble (bluetooth low energy) central module,” (Date last accessed 16-May-2016). [Online]. Available: <https://github.com/sandeepmistry/noble>

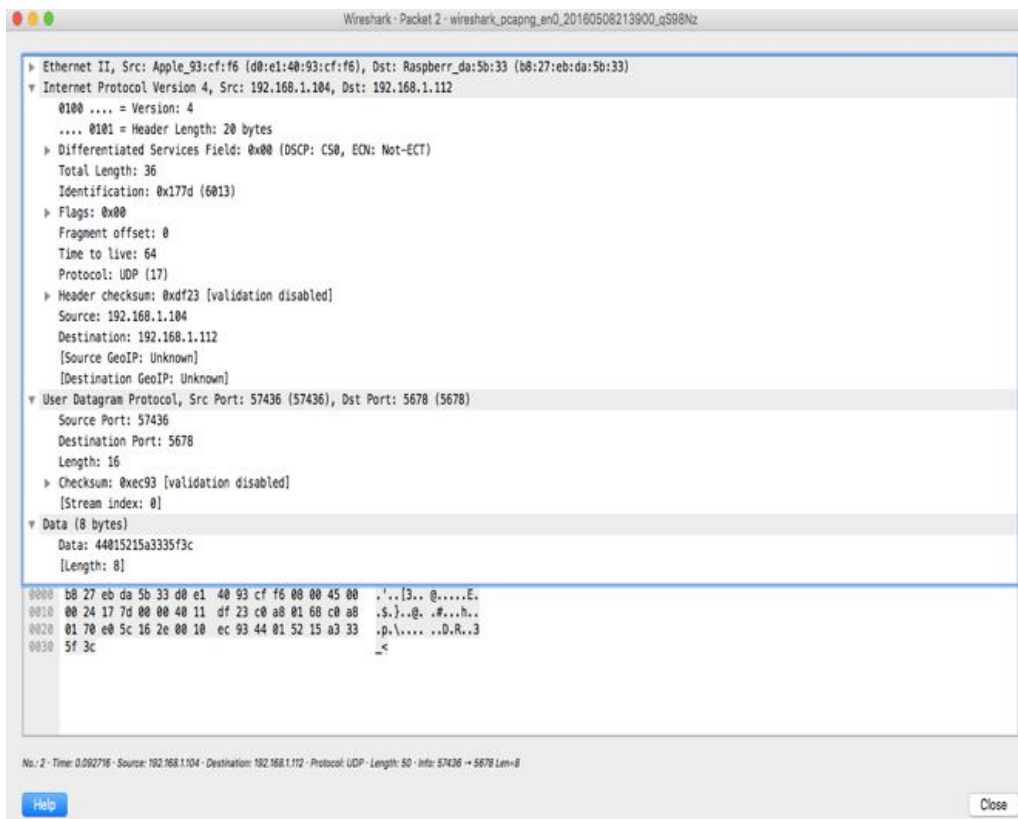


圖 11: BLE 封包

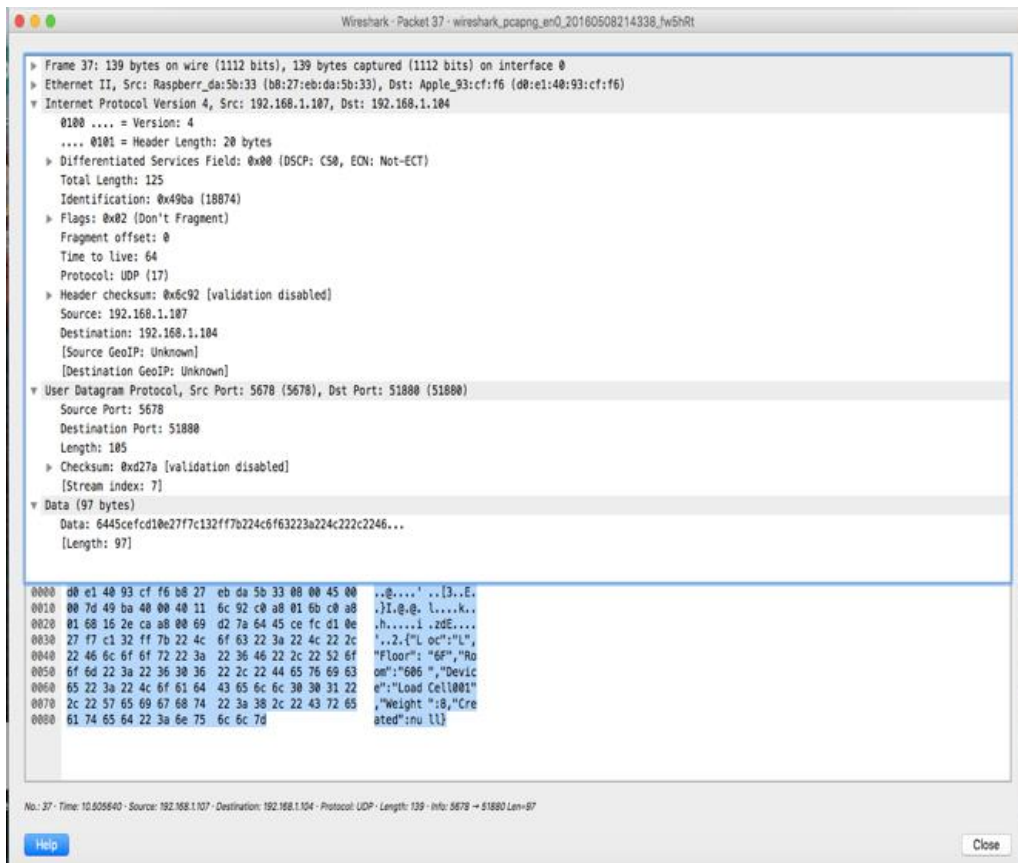


圖 12: CoAP 封包