

科技部自由軟體專案研究計劃『系統測試報告』

System Testing Document Of MOST Open Source Project

雲端物聯技術與平台設計：以智慧農業為驗證場域
**Design of Cloud of IoT Technology and Platform: Intelligent
Agricultural Testbed**
MOST 104-2221-E-020-016

龔旭陽

國立屏東科技大學 資管系

Department of Engineering and Applied Science

National Science Council, Taiwan

2016/05/24

開放式物聯網中介平台之設計

Design of An Open IoT Middleware System (OIMS)

執行時間：2015.08 至 2016.07

OIMS System Testing Document

版本 1.0

(Version 1.0)



開放式物聯網中介平台之設計

Design of An Open IoT Middleware System (OIMS)

執行時間：2015.08 至 2016.07

OIMS System Testing Document

Prepared by

黃思淵

楊翌倩

林宛億

張宇鈞

張筑鈞

Approved by

龔旭陽 教授



目錄

版本變更記錄.....	6
第 1 章 測試目的與接受準則 (Objectives and Acceptance Criteria)	7
1.1 系統範圍(System Scope).....	7
1.2 測試文件目的(Purpose of this Document).....	8
1.3 測試接受準則(Test Acceptance Criteria).....	9
第 2 章 測試環境(Testing Environment)	10
2.1 操作環境(Operational Environment).....	錯誤! 尚未定義書籤。
2.2 硬體規格(Hardware Specification)	11
2.3 軟體規格(Software Specification).....	12
2.4 測試資料來源(Test Data Sources)	12
第 3 章 測試行程、流程與職責 (Testing Schedule, Procedure, and Responsibility)	13
3.1 測試行程(Testing Schedule)	13
3.1.1 時程.....	13
3.1.2 里程碑.....	13
3.2 測試流程(Testing Procedure)	13
3.2.1 Subsystems Validation	14
3.2.2 Integration Testing	16
3.2.3 Acceptance Testing	16
3.3 個人職責 (Testing Responsibility)	20
第 4 章 Test Case	21
4.1 Integration Testing Cases	21
4.1.1 IT1 Test Case	21
4.1.2 IT2 Test Case	22
4.1.3 IT3 Test Case	23
4.1.4 IT4 Test Case	24
4.2 Acceptance Testing Case	26
4.2.1 AT1 Test Case	26
4.2.2 AT2 Test Case	27
4.2.3 AT3 Test Case	28
4.2.3 AT4 Test Case	28
5. 測試效能分析	30
5.1 車路通訊模擬效能分析	30
5.2 可行性分析	錯誤! 尚未定義書籤。
6. Test Result and Analysis	45

	Integration Testing Cases	45
7.	Appendix A: Glossary	46
8.	Appendix B: Traceability	48

版本變更記錄

版本	變更項目	變更日期
V1.0	第一版	2016.05.24

第1章 測試目的與接受準則

(Objectives and Acceptance Criteria)

1.1 系統範圍(System Scope)

子計畫一設計與實作「開放式物聯網中介平台之設計(Design of An Open IoT Middleware System)」，本計畫著重探討各種使用者(包括農業生產者、一般民眾)透過網路服務在農業資訊系統上存取各式應用服務時，如何享有即時傳輸與穩定的服務品質，而各式應用服務的資料來源為多型態資料流(文字、圖片、影像、聲音)，且同時在存取各式應用服務的使用者眾多，這種情況下該如何調節配置傳輸資源與依照不同使用者需求給予資料串流傳輸優先權，為我們重要的研究議題。在農業生產的環境中會佈建多種感測設備(溫度、濕度、二氧化碳、土壤水份等)來監測控制環境狀態，這些資料透過網路系統可上傳至雲端環境儲存記錄，以提供相關資訊給各式應用系統或使用者運用。對於民眾來說，他們對食品安全有重視，故可利用農業資訊系統建置的生產履歷管理來追溯產地資訊與整個運送過程；農業休閒觀光逐漸盛行，民眾會希望透過資訊導覽解說服務來獲得豐富的農業相關資訊或知識，甚至與之互動；對於農民生產者，他們可透過遠距監控系統查看場域環境的各種因素狀態。當使用者在存取服務時，往往應用系統所需的資料來源會同時涵蓋文字資料、影像資料、圖片資料等多型態資料流，而使用者數量眾多，物聯中介伺服器必須依照各使用者需求給予即時且穩定的傳輸服務。而當使用者想立即查看場域中所感興趣的監測資料，例如作物影像，可先從物聯網資料庫中辨別最新一筆的相關資料是否足以呈現服務，若判定其無法表現最新狀態，則可以由物聯網伺服器向相關聯的感測設備要求資料以提供給應用服務。

物聯網即是透過物與物相聯的概念，免去人力操控的介入而達到智慧化控制的服務，而過去物聯網應用平台皆屬垂直式整合，不同的應用情境即發展不同

的物聯網平台，根據其特殊的需要來整合所需之感測器、網路、應用服務層，要移植到不同應用情境實屬不易，為了達成水平式整合之效，需要建置一整合之開放物聯網中介平台。因此本計畫將建置並整合以下子元件，如表 1.1 所示。

表 1.1 各子元件功能及輸出結果

子元件名稱	功能	輸出結果
高效能物聯網資訊傳輸平台	測試感測器與系統之連線是否穩定，且 SCTP 的多重串流功能(multi-streaming functionality)能否降低網路 overhead。	1. 可收集完整感測資料 2. 提升流量控制之效率
溫室監控伺服器與資料庫系統	測試溫室監控伺服器能否即時監測環境狀況及現場影像，且資料庫是否能回饋經運算整理過後的溫室統計報表等資料。	1. 即時監測環境狀況及現場影像。 2. 提供溫室統計報表等資料。
多異質感測設備資料整合中介平台	能否根據異質感測服務資料來源，針對農業環境變化與植栽變化資訊進行整合性的變化監測，且進行資料格式的轉換處理。	整合各異質感測器之感測資料。
oneM2M 服務處理伺服器	所導入之 oneM2M 架構，能否處理大量感測資源，且將資料根據使用者之興趣主動推撥。	系統可主動推撥使用者感興趣之資料，也可要求其他資料。

1.2 測試文件目的(Purpose of this Document)

本文件主要建立「開放式物聯網中介平台之設計」的測試和整合計劃、程序與報告，在系統進行整合之前，我們必須先確定系統元件之單元測試是否已完成，並且著重於從需求文件和設計文件來進行整合系統測試(Integration Test)及接受度測試(Acceptance Test)，典型之系統整合程序如圖 1.1 所示。本文件內容將依據系統需求規格書與系統設計文件，描述相關整合測試的相關計畫內容，並希望透過此文件之描述與實踐，達到順利進行測試工作之目的。

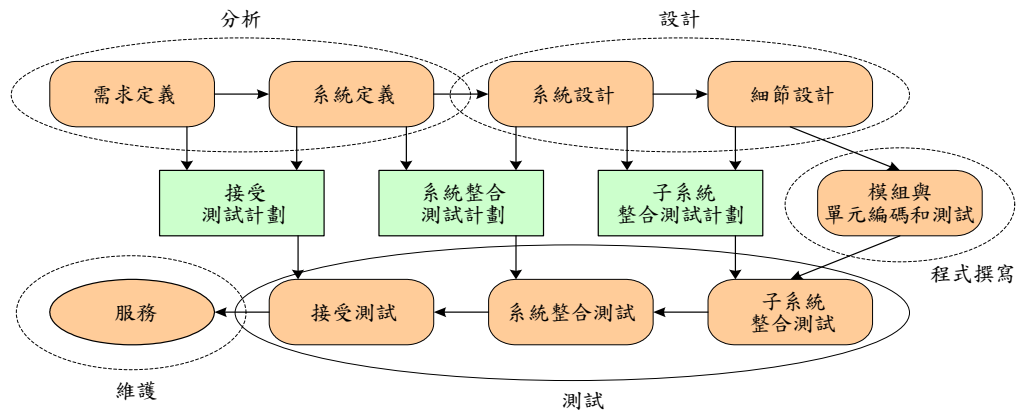


圖 1.1 系統整合程序

1.3 測試接受準則(Test Acceptance Criteria)

本測試計畫需要滿足下面的測試接受準則：

- 針對所有優先次序列為 Critical、Important 與 Desirable 之需求做完整測試。
- 測試程序需要依照本測試計畫所訂定的程序進行，所有測試結果需要能符合預期測試結果方能接受。
- 以測試案例為單位，當測試未通過時，需要進行該單元的測試，其接受的準則如第一項中所規定的相同。
- 解決系統中相互抵觸的元件與相關功能。

第2章 測試環境(Testing Environment)

2.1 操作環境(Operational Environment)為了讓測試環境中的任何設備(包含行動裝置和感測器等)皆能藉由統一的標準及傳輸介面相互溝通與控制，本計畫導入 oneM2M 標準來完成中介平台之建置，以期望達到無須人力即可達到智慧化控制與異質感測資料整合之目的，為了實現上述，本計畫規劃了 4 大元件，分別為(1) 高效能物聯網資訊傳輸平台(High Performance IoT Platform for Information Transmission)、(2) 溫室監控伺服器與資料庫系統(Greenhouse Monitoring Server and Database System)、(3) 多異質感測設備資料整合中介平台(Multi Heterogeneous Sensing Device Integration Middleware)、(4) oneM2M 服務處理伺服器(oneM2M Service Processing Server)。本系統之系統架構圖，如圖 2-1 所示。

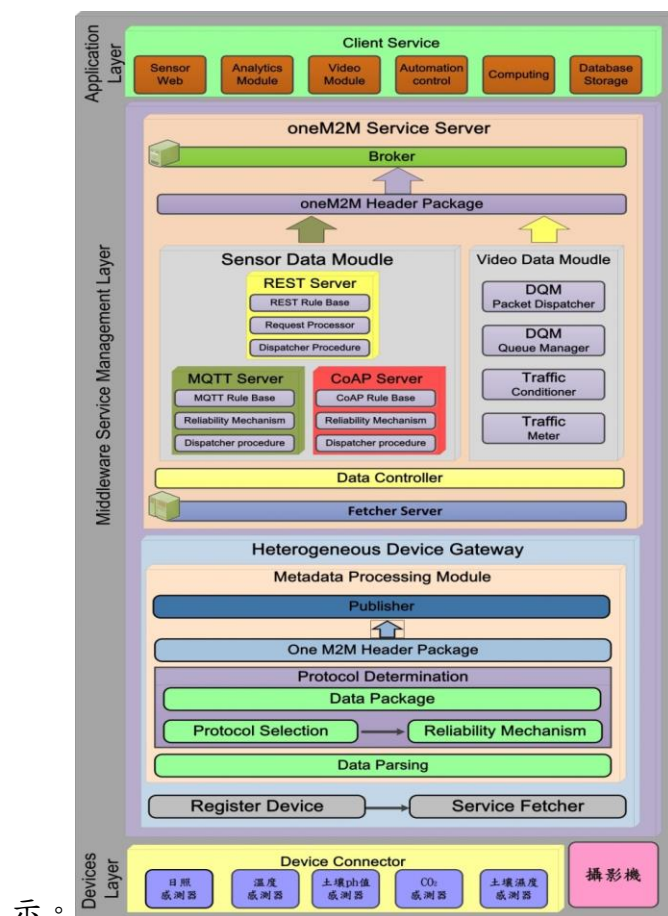


圖 2.1 開放式物聯網平台之設計與實作

2.2 硬體規格(Hardware Specification)

本系統主要硬體設備包括運算伺服器、行動式設備、環境感測器、以及資料傳輸設備等構成，提供感測器之間資訊的運算、分析及無線傳輸通訊能力，無線傳輸設備及行動式運算設備具備資訊轉送及接收能力。相關的設備需求如表 2.1 所示。

表 2.1 各設備裝置資源

硬體名稱	規格說明
Mobile Devices (AE)	廠牌：SAMSUNG GALAXY Note10.1 CPU：Quad-Core 1.4GHz Memory：2GB OS：Android 4.1.2 Development Tools：Eclipse Android SDK
oneM2M Server (IN)	CPU：Intel(R)Core(TM)i7-4770 CPU@3.40GHz Memory：8GB Development Tools：Node.js Development Language：JavaScript Database：MySQL OS：Windows 7 64 bit
oneM2M Gateway (CSE)	CPU：Intel(R)Core(TM)i5-3570 CPU@3.40GHz Memory：3.5GB Development Tools：Eclipse Development Language：Java Database：MySQL OS：Windows XP 32 bit
Sensor Devices	瑞帝-Zigbee無線通訊繼電器控制模組(Zigbee協定)、自動

	控制模組(燈具+繼電器)、溫溼度感測器，二氧化碳感測器、光照感測器、土壤濕度感測器、網路攝影機。
--	--

2.3 軟體規格(Software Specification)

本系統所需求的軟體有(1)作業系統：伺服器端為 Microsoft Windows 7 專業版 SP1 作業系統與使用者端為 Android4.0 作業系統。(2)開發工具：Eclipse SDK、Java 語言、ASP、PHP 網頁程式語言、及 SQL 資料庫語言。(3)使用套件：node js v4.4.4-x86、python-2.7.11。

2.4 測試資料來源(Test Data Sources)

關於本測試計畫測試期間所需之測試資料來源，說明如下：

- 農業環境下微氣候感測設備收集之感測資料。
- 網路攝影機即時監控之影像。
- 透過 JenNet-IP 監看各感測器之網路連線及運作狀況，包含網路組成報告。
- 藉由 JenNet-IP 回傳的設備識別資料提供溫室監控伺服器資料整合與資料表設計之參考。
- 利用各串流資料之 life time、data rate 及更新頻率定義多重串流間傳輸速率的協調方式，以達效能最大化。
- 利用 Zigbee、WSN、GPS、Camera 等感測設備作為異質感測服務的資料來源，針對農業環境變化與土壤資訊及影像資訊進行整合性的變化監測。
- 根據感測器資料接收來源來進行標準化資料格式編碼與資料轉換處理。

第3章 測試行程、流程與職責

(Testing Schedule, Procedure, and Responsibility)

3.1 測試行程(Testing Schedule)

3.1.1 時程

- OIMS 子系統之高效能物聯網資訊傳輸平台單元測試—2016/1/15 ~ 2016/3/2
- OIMS 子系統之溫室監控伺服器與資料庫系統單元測試—2016/3/5 ~ 2016/4/10
- OIMS 子系統之多異質感測設備資料整合中介平台單元測試—2016/4/11~2016/5/2
- OIMS 子系統之 oneM2M 服務處理伺服器單元測試—2016/5/3~ 2016/5/14
- OIMS 與各子機制整合測試—2016/5/14~ 2016/5/19

3.1.2 里程碑

- OIMS 子系統之高效能物聯網資訊傳輸平台單元測試—2016/1/15 ~ 2016/3/2
- OIMS 子系統之溫室監控伺服器與資料庫系統單元測試—2016/3/5 ~ 2016/4/10
- OIMS 子系統之多異質感測設備資料整合中介平台單元測試—2016/4/11~2016/5/2
- OIMS 子系統之 oneM2M 服務處理伺服器單元測試—2016/5/3~ 2016/5/14
- OIMS 與各子機制整合測試—2016/5/14~ 2016/5/19

3.2 測試流程(Testing Procedure)

3.2.1 Subsystems Validation

1. 高效能物聯網資訊傳輸平台(HPIT)：由於溫室中可能存在溫度、溼度、照度、二氧化碳濃度等各式各樣的感測器，並配備多支攝影裝置以利監控溫室各區域的生長狀態，當農夫利用智慧型手機來開啟溫室監控軟體時，可能會嘗試同時擷取溫室中各項裝置所能提供的資訊，此時 Data originators 與 user 之間將如典型的物聯網一樣，產生大量的連線。為避免大量連線造成網路橋接設備的負擔，甚至不勘負荷而當機，因此將針對 Data originators 與 Data collector 間的連結進行研究，透過佈建 NXP Semiconductors 所發佈的 JenNet-IP 來解決無線感測器等裝置與無線網路存取點間的連線問題。
2. 溫室監控伺服器與資料庫系統(GMSD)：溫室監控伺服器於本計畫中扮演重要角色，是實現自動化溫室監控系統的樞紐。本計畫針對伺服器對溫室內設備以及伺服器與使用者間之傳輸協調進行設計，以期減少伺服器網路、運算以及儲存資源的浪費。本計畫中所佈建之溫室監控伺服器對下收集來自溫室感測器等設備所量測的資訊，對上則為使用者提供兩種資訊(1)來自溫室現場最即時的環境條件參數以及現場影音畫面等『資料』；(2)來自資料庫，經運算整理後的溫室統計報表等『資訊』。因此在實作溫室監控伺服器時，將設計一 M2M API 來與溫室內 gateway 溝通，並遵循 oneM2M 標準對溫室內各式設備進行溝通與控制，另一方面則設計專屬的 User API 來與使用者設備進行互動。伺服器本身將做為使用者與溫室內設備間的溝通橋樑，除定期收取來自溫室各種設備的資料，將分類好的資料儲存於資料庫系統，另一方面也接受使用者的請求，提供溫室現場的資料或資料庫經整理後的資訊。為了提升溫室至伺服器與伺服器至使用者設備間的傳輸品質，本計畫將採用串流控制傳輸協議 (Stream Control Transmission Protocol, SCTP) 來做為傳輸層的通訊協定。因此，將在溫室監控伺服器上開發 SCTP 多重串流控制機制，針對 M2M 的傳輸特性進行流量控制的優化，提供此計畫高效能的網路功能性支援。

3. 多異質感測設備資料整合中介平台(HDIM)：此部分可分為 2 層進行設計與實作，分別為(1)感知設備層(Sensor Device Layer)：主要為將各式感測服務與設備訊息傳至異質感測服務閘道器進行資料的轉換，我們依據農業應用情境，利用 Zigbee、WSN、GPS、Camera 等感測設備作為異質感測服務的資料來源，針對農業環境變化與土壤資訊及影像資訊進行整合性的變化監測，以提升精準農業種植與環境監控之效率；(2)中介服務平台管理層(Middleware Service Management Layer)：主要功能為進行感測資料的標準化動態中介處理與提升伺服器的服務效能，為了使其架構能更符合物聯網環境與增加系統平台擴增性，本階層又分別為閘道器(Gateway)與 M2M 服務伺服器(M2M Service Server)兩個部份進行整合性設計與開發，其依序為異質感測服務閘道器(Heterogeneous Sensor Service Gateway)與 M2M 服務伺服器架構。於異質感測服務閘道器方面，針對資料接收來源來進行標準化資料格式編碼與資料轉換處理，再經由 oneM2M 標準所規範之通訊協定(REST、MQTT、CoAP)針對感測資料進行合適之協定挑選，以達輕量化資料交換格式進行傳輸；而 M2M 服務伺服器(M2M Service Server)，主要接收異質感測服務閘道器所傳來之感測資料，透過資料分配器(Data Controller)將資料分為感測資料或影像資料，進行資料服務品質之設定，再以 IBM 物聯網中的 Broker 機制，直接將資料分享給訂閱者，例：資料分析器、資料庫、感測服務觀測，藉此減少伺服器的處理負擔與有效利用感測資源。
4. oneM2M 服務處理伺服器(OSPS)：本計畫為提供統一應用服務與來源格式標準化，所設計之異質感測服務閘道器進行來源格式的前置處理，但在物聯網感測環境下之閘道器(Gateway)主要負責將大量感測資源藉由網路傳送至後端進行資源處理，而後端伺服器如何承受大量的感測資源並有效進行存取控制與快速地將所需服務提供給使用者，將為此設計開發重點。有鑑於大量感測資料所產生的服務存取、實體機器負載、來源格式等問題，本研究導入

oneM2M 架構，設計一伺服器處理架構用於處理物聯網環境下的大量感測資源(Resource)，並且讓各異質化感測器都能藉由此服務伺服器進而將資料做更有效率的應用，以提升物聯網環境之處理效率與擴增性，並改變以往資料需由使用者向伺服器提出請求，使用訊息推播機制，讓訊息可以主動的被推播至有興趣的客戶端(Client)。

3.2.2Integration Testing

子計畫一整合性測試，針對開放式物聯網中介平台之設計各元件之整合測試，並測試各子機制間相關的整合。如圖 3.1 所示，為 OIMS 之整合測試圖。

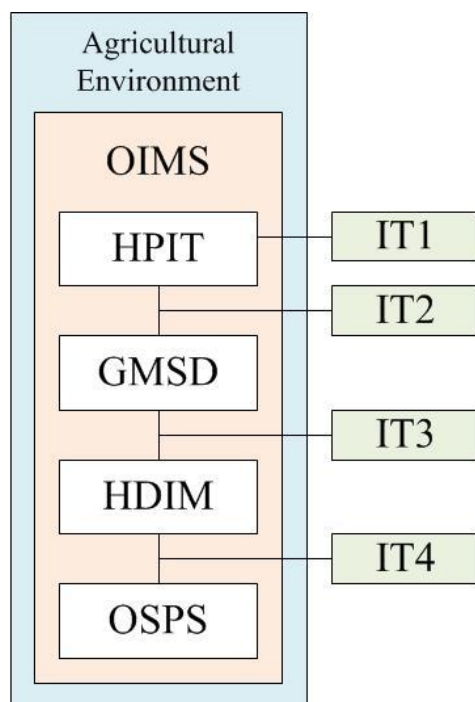


圖 3.1 OIMS 之整合測試

3.2.3Acceptance Testing

本計畫將提出一標準介面之「開放式物聯網平台」，其中導入 oneM2M 標準來設計物聯網中介平台，其包含了(1)異質感測網路服務閘道器(Gateway)，主要參考 oneM2M 標準中所規範之三種傳輸協定(REST、MQTT、CoAP)，藉此進行異質感測設備之封包封裝與資料的中介轉換處理；(2)oneM2M 服務伺服器處

理架構，主要參考 oneM2M 標準設計而成，透過資料控制器將異質感測網路服務閘道器所傳來的資料進行分配，將其分配給不同通訊協定的處理伺服器，再導入 IBM 規範之物聯網標準進行資料推播，包含以下元件。

- (1). JenNet-IP 模組：採用 IETF 之增強型 6LoWPAN 網路層，並針對 IEEE 802.15.4 MAC layer 之連線而設計，可令各種智慧型裝置採用 IPv6 連結至網路，並可同時容納 500 個以上的網路設備。
- (2). SCTP 傳輸協定模組 (Stream Control Transmission Protocol, SCTP)：為了提升溫室至伺服器與伺服器至使用者設備間的傳輸品質，將採用 SCTP 來做為傳輸層的通訊協定。
- (3). 改良式串流控制傳輸協議(Improved Stream Control Transport Protocol)：在單一 SCTP 連線中搭載多個串流，以減少連線的 overhead。且這些屬於同一 SCTP 連線的串流擁有各自獨立的流量控制機制(rate control)，不互相干擾，因此能有效減少 TCP 常見的 Head of line blocking(HLB)問題。
- (4). 異質感測服務閘道器(Heterogeneous Service Gateway)：環境感測設備需先向閘道器進行註冊為發送者(Publisher)，閘道器將註冊成功的感測設備進行設備的連結及開始感測器收值，並依照感測器類型進行對應的感測資訊解析，解析完後針對感測器資料類型進行合適通訊協定之挑選，再結合各通訊協定之可靠性機制進行感測資料可靠度格式設定，最後進行 oneM2M 格式封裝，以完成感測資料上網前置作業。

如圖 3.2 所示，為 OIMS 使用個案圖。如表 3.1 所示，為 OIMS 各元件的 Acceptance Testing Table。

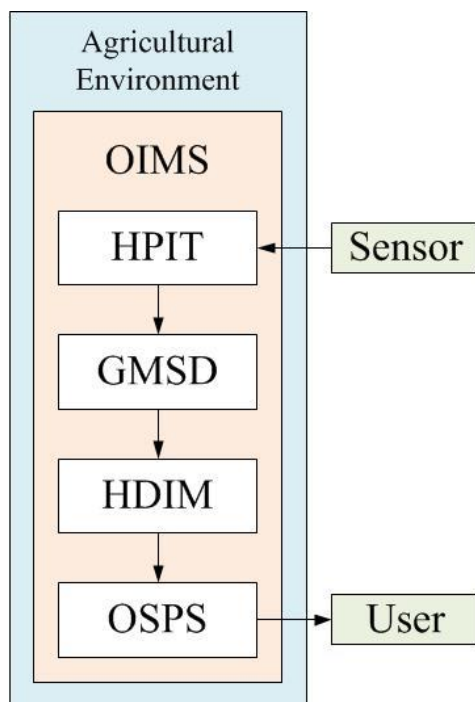


圖 3.2 OIMS 之使用個案圖

表 3.1 OIMS 各元件的 Acceptance Testing Table

編號	元件	測試項目
AT1	高效能物聯網資訊傳輸平台	測試感測器與系統之連線是否穩定，且 SCTP 的多重串流功能(multi-streaming functionality)能否降低網路 overhead。
AT2	溫室監控伺服器與資料庫系統	測試溫室監控伺服器能否即時監測環境狀況及現場影像，且資料庫是否能回饋經運算整理過後的溫室統計報表等資料。
AT3	多異質感測設備資料整合中介平台	能否根據異質感測服務資料來源，針對農業環境變化與植栽變化資訊進行整合性的變化監測，且進行資料格式的轉換處理。

AT4	oneM2M 服務處理伺服器	所導入之 oneM2M 架構，能否處理大量感測資源，且將資料根據使用者之興趣主動推撥。
IT1	於有限資源下，進行感測資料傳送	驗證感測器於有限資源下，能否穩定且完整的將感測資料傳送至高效能物聯網資訊傳輸平台。
IT2	透過 gateway 導入 oneM2M 標準對各式設備進行溝通與控制	測試 gateway 能否進行溝通與控制，以作為使用者與設備間的溝通橋樑。
IT3	異質感測服務閘道器 (Heterogeneous Sensor Service Gateway) 負責收集感測資料，再透過 M2M 服務伺服器進行資料分配	測試異質感測服務閘道器能否將資料完整傳送至 M2M 服務伺服器進行資料分配。
IT4	設計一動態服務品質佇列對應 (Dynamic QoS Queue Mapping, DQ ² M) 演算法，以改變進入差異化服務佇列的流量型態。	驗證 DQ ² M 演算法能否解決某一佇列封包過多而其他佇列使用率低，進而造成壅塞佇列承受高延遲時間與高遺失率，無法整體有效利用資源之問題。

3.3 個人職責 (Testing Responsibility)

OIMS 子系統預計完成的細部功能分別由下列人員進行單元性測試與整合性之測試，人員測試分派如表 3.2 所示。

表 3.2 人員與職責

Testing Activities	Personnel
AT1	黃思淵、張宇鈞
AT2	楊翌倩、林宛億、張筑鈞
AT3	黃思淵、張宇鈞
AT4	黃思淵、張宇鈞、張筑鈞
IT1	黃思淵、張宇鈞
IT2	黃思淵、張宇鈞
IT3	楊翌倩、林宛億
IT4	黃思淵、張宇鈞、張筑鈞

第 4 章 Test Case

4.1 Integration Testing Cases

4.1.1 IT1 Test Case

本測試主要驗證高效能物聯網資訊傳輸平台，依據[HPIT -N-009]與[HPIT -N-011]需求，交由[HPIT -N-001]、[HPIT -N-003]、[HPIT -N-004]將收集的資訊進行傳送，透過[HPIT -N-012]將得到 JSON 格式之封包，並運用[HPIT -N-013]提升傳輸之效率。

Identification	IT1
Name	於有限資源下，進行感測資料傳送
Test Target	驗證感測器於有限資源下，能否穩定且完整的將感測資料傳送至高效能物聯網資訊傳輸平台。
Reference	HPIT -N-001 HPIT -N-003 HPIT -N-004 HPIT -N-009 HPIT -N-011 HPIT -N-012 HPIT -N-013
Severity	1
Instruction	1. 收集環境感測資料 2. 封裝 JSON 格式之封包 3. 傳送高效能物聯網資訊傳輸平台
Expected Result	穩定且完整的將感測資料傳送至高效能物聯網資訊傳輸平台
Cleanup	無。

4.1.2 IT2 Test Case

主要驗證[GMSD-N-015]與[GMSD -N-016]需求，依據[GMSD -N-001]與[GMSD -N-002] 建立一具有 oneM2M 標準之 gateway，透過[GMSD -N-003]進行溝通訊息之傳送，以確認設備狀況，根據[GMSD -N-007]與[GMSD -N-008]測試本系統所建立之 gateway 能否控制感測設備。

Identification	IT2
Name	透過 gateway 導入 oneM2M 標準對各式設備進行溝通與控制
Test Target	測試 gateway 能否進行溝通與控制，以作為使用者與設備間的溝通橋樑。
Reference	GMSD-N-001 GMSD-N-002 GMSD-N-003 GMSD-N-007 GMSD-N-008 GMSD-N-015 GMSD-N-016
Severity	1
Instruction	1. 建立一具有 oneM2M 標準之 gateway 2. 傳送溝通訊息，以確認設備狀況 3. 傳送控制指令，以利操控設備
Expected Result	1. 可與感測器進行訊息溝通 2. 可透過指令控制感測設備
Cleanup	無。

4.1.3IT3 Test Case

主要驗證 [HDIM -N-010]與[HDIM -N-011]需求，交由[HDIM -N-001]進行異質資料收集，再經由[HDIM -N-002]設定分配的門檻值，透過[HDIM -N-005]與[HDIM -N-008]判斷資料所需分配之位置。

Identification	IT3
Name	異質感測服務閘道器(Heterogeneous Sensor Service Gateway)負責收集感測資料，再透過 M2M 服務伺服器進行資料分配
Test Target	測試異質感測服務閘道器能否將資料完整傳送至 M2M 服務伺服器進行資料分配。
Reference	HDIM -N-001 HDIM -N-002 HDIM -N-005 HDIM -N-008 HDIM -N-010 HDIM -N-011
Severity	1
Instruction	1. 藉由異質感測服務閘道器進行資料收集 2. 設定判斷之標準 3. 進行資料合適位置的分配
Expected Result	1. 完整收集異質感測資料 2. 可判斷感測資料之類型 3. 可根據判斷結果分配合適位置
Cleanup	無。

4.1.4IT4 Test Case

主要驗證[OSPS -N-010]與[OSPS -N-009]需求，透過[OSPS -N-002] 與[OSPS -N-003] 依不同服務需求設計一動態服務品質佇列對應演算法，再交由[OSPS -N-005] 與[OSPS -N-008]根據封包大小與資料優先順序進行週期性的評估，由[OSPS -N-09]可動態地依照網路狀況進行佇列調配，以解決壅塞佇列承受高延遲時間與高遺失率，無法整體有效利用資源之問題。

Identification	IT4
Name	設計一動態服務品質佇列對應(Dynamic QoS Queue Mapping, DQ ² M) 演算法，以改變進入差異化服務佇列的流量型態。
Test Target	驗證 DQ ² M 演算法能否解決某一佇列封包過多而其他佇列使用率低，進而造成壅塞佇列承受高延遲時間與高遺失率，無法整體有效利用資源之問題。
Reference	OSPS -N-002 OSPS -N-003 OSPS -N-005 OSPS -N-008 OSPS -N-009 OSPS -N-010
Severity	1
Instruction	1. 設計一動態服務品質佇列對應演算法 2. 接收感測資訊及監控影像 3. 進行週期性評估 4. 動態的進行佇列調配
Expected Result	解決壅塞佇列承受高延遲時間與高遺失率，無法整體有

	效利用資源之問題
Cleanup	無。

4.2 Acceptance Testing Case

4.2.1 AT1 Test Case

目的：驗證[HPIT -N-013]需求，測試感測器與系統之連線是否穩定，且 SCTP 的多重串流功能(multi-streaming functionality)能否降低網路 overhead。

Identification	AT1	
Name	高效能物聯網資訊傳輸平台	
Tested Target	HPIT -N-013	
Reference	[HPIT -N-001] [HPIT -N-002] [HPIT -N-003] [HPIT -N-004] [HPIT -N-005] [HPIT -N-006]	
Severity	1	
Instruction	<i>Actor actions</i>	<i>System responses</i>
	系統請求感測資料	
		與感測器進行連線
		回傳感測資料
		SCTP 進行流量控制的優化
	獲得感測資訊	
Expected Result	1. 可收集完整感測資料 2. 提升流量控制之效率	
Cleanup	無。	

4.2.2 AT2 Test Case

目的：驗證[GMSD -N-015]需求，測試溫室監控伺服器能否即時監測環境狀況及現場影像，且資料庫是否能回饋經運算整理過後的溫室統計報表等資料。

Identification	AT2	
Name	溫室監控伺服器與資料庫系統	
Tested Target	GMSD -N-015	
Reference	[GMSD -N-001] [GMSD -N-002] [GMSD -N-003] [GMSD -N-004] [GMSD -N-005] [GMSD -N-006] [GMSD -N-007]	
Severity	1	
Instruction	<i>Actor actions</i>	<i>System responses</i>
	使用者請求服務	
		溫室監控伺服器與感測器建立連線
		感測器進行資訊收集與管理控制
		感測器與伺服器之間的通溝與訊息交換
		儲存感測資料及相關紀錄
	獲得服務資訊	
Expected Result	1. 即時監測環境狀況及現場影像。 2. 提供溫室統計報表等資料。	
Cleanup	無。	

4.2.3 AT3 Test Case

目的：驗證[HDIM -N-011]需求，測試確實選擇品質最好的傳輸通道進行傳輸。

Identification	AT3	
Name	多異質感測設備資料整合中介平台	
Tested Target	HDIM -N-011	
Reference	[HDIM -N-001] [HDIM -N-002] [HDIM -N-003] [HDIM -N-007] [HDIM -N-008]	
Severity	1	
Instruction	<i>Actor actions</i>	<i>System responses</i>
	使用者請求服務	
		分析資料來源
		整合性的變化監測
		資料格式的轉換處理
	獲得服務資訊	
Expected Result	1. 整合各異質感測器之感測資料。	
Cleanup	無。	

4.2.4 AT4 Test Case

目的：驗證[OSPS -N -010]需求，測試所導入之 oneM2M 架構，能否處理大量感測資源，且將資料根據使用者之興趣主動推撥。

Identification	AT4
Name	oneM2M 服務處理伺服器
Tested Target	OSPS -N -010

Reference	[OSPS -N-001] [OSPS -N-002] [OSPS -N-003] [OSPS -N-004] [OSPS -N-006] [OSPS -N-009]	
Severity	1	
Instruction	<i>Actor actions</i>	<i>System responses</i>
	使用者請求服務	
		將資料根據類型配送至合適的伺服器
		封裝 oneM2M 標頭格式
		主動推撥使用者感興趣之資訊
	獲得服務資訊	
Expected Result	系統可主動推撥使用者感興趣之資料，也可要求其他資料。	
Cleanup	無。	

5. 測試效能分析

本計畫『開放式物聯網中介平台之設計』所實現之功能，主要用於物聯網感測環境中，利用開放式標準進行來源感測資訊格式的標準化，並藉由 oneM2M 來設計資料處理架構，藉此來有效處理環境中大量異質感測資訊的問題。在效能分析部份主要針對本研究所應用之應用層通訊協定進行效能分析及比較探討。為了能有效顯示本計畫所設計之架構優點，本章節將針對服務時間效率、資料量服務品質等方向進行整體效能的比較與評估，其中將參考 Thangavel, D.等學者[18]的效能比較對象進行 MQTT、CoAP、REST 三種規範於 oneM2M 標準中的應用層通訊協定來進行傳輸效率之評估。

5.1 分析之情境與比較項目

在評估項目方面本章節將分為三部分來進行效能整體評估(1)字串事件(String Events)服務往返(Round-Trip)時間效能比較；(2)影像事件(Image Events)服務往返(Round-Trip)時間效能比較；(3)影片事件(Video Events)服務往返時間效能比較，並由各種不同環境來進行分析，情境如表 5-1 分析情境說明所示，而分析類型設定如表 5-2 系統測試參數說明表所示，以下為各評估項目詳細之說明。

表 5-1 分析情境說明

情境	情境說明
情境 1	Local(Server 與 Client 同一台電腦)，如圖 5-1 情境 1 示意圖所示。
情境 2	Intranet(Server 與 Client 不同台電腦，但屬於對內網路)，如圖 5-2 情境 2 示意圖所示。
情境 3	Internet(Server 與 Client 不同台電腦，但屬於對外網路)，如圖 5-3 情境 3 示意圖所示

表 5-2 系統測試參數說明表

比較項目	實驗樣本數
建立物聯網環境感測事件時間效能分析比較(Local、Intranet、Internet) 感測器個數：5 個 感測事件內容大小：966 Bytes	各 100 次
建立物聯網影像事件時間效能分析比較(Local、Intranet、Internet) 影像大小：180k (184,411 Bytes)	各 100 次
建立影片事件時間效能分析比較(Local、Intranet、Internet) 影片大小：888k (909,312 Bytes)	各 100 次
物聯網環境封包掉落率分析比較(Intranet、Internet) 感測器個數：5 個 感測事件內容大小：966 Bytes 傳送速率：10 次/S	10000 次

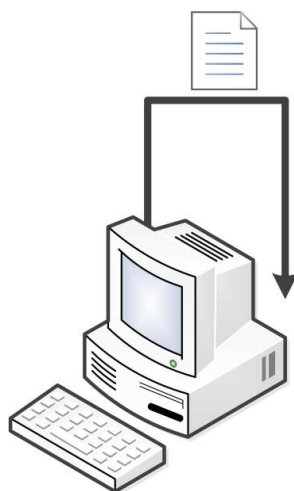


圖 5-1 情境 1 示意圖

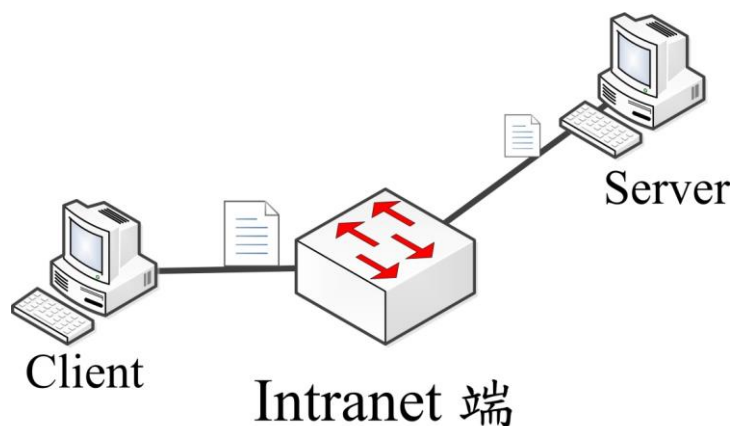


圖 5-2 情境 2 示意圖

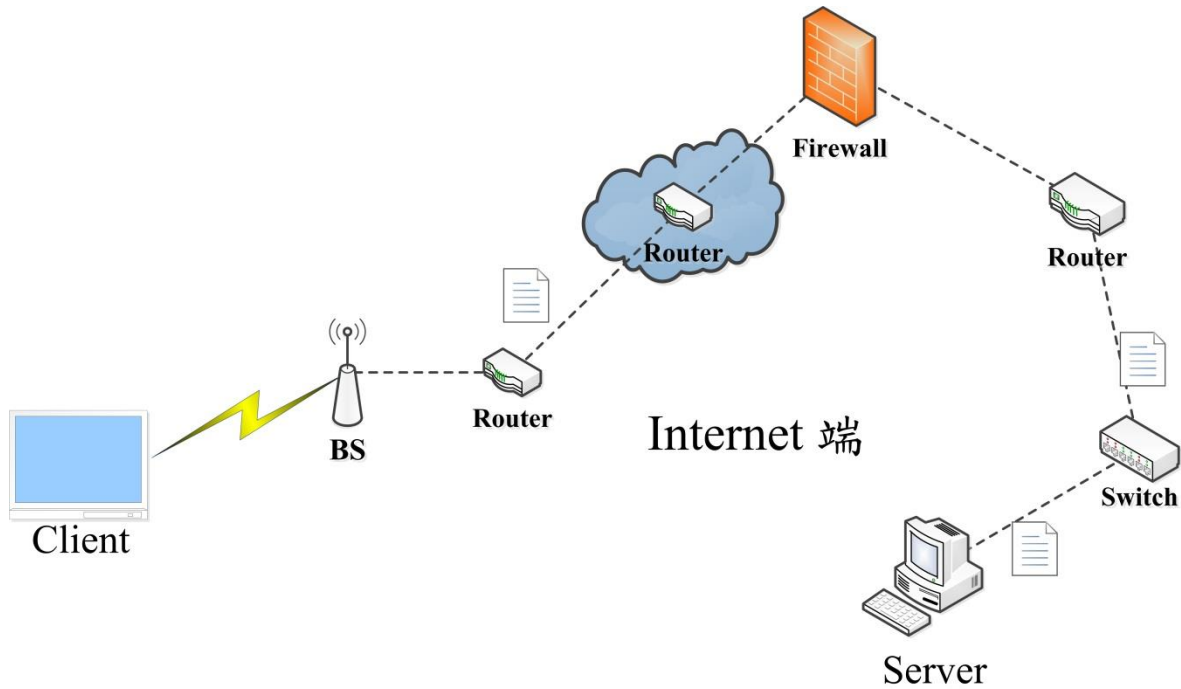


圖 5-3 情境 3 示意圖

5.1.1. 物聯網環境感測事件服務往返(Round-Trip)時間效能比較

在測量次數部分，本實驗進行 100 次完整服務的測量，藉此觀察整體服務的平均服務時間與服務建立穩定度。於情境 1 藉由分析結果如表 5-3 MQTT、REST、CoAP 物聯網環境感測事件時間效能分析比較表(Local)與

圖 5-4 MQTT、REST、CoAP 物聯網環境感測事件服務往返時間之分析(Local)所示，MQTT 最大服務時間與最小服務時間分別為 85 ms 與 73 ms，差距為 12 ms，REST 最大服務時間與最小服務時間分別為 275 ms 與 251 ms，差距為 24 ms，CoAP 大服務時間與最小服務時間分別為 96 ms 與 87 ms，差距為 9 ms，分析結果指出，三者相比較可以發現 MQTT 服務時間反應較快，且所需之平均服務處理時間少於 REST 整體服務時間約 3.4 倍，其平均服務處理時間也少於 CoAP 整體服務時間約 1.2 倍，藉由公式(1)、(2)進行性能的運算評估 MQTT 在建立事件的平均服務處理效能上快於 REST 及 CoAP 約 70 %及 17 %的效能處理速度，於 Local 情境中物聯網環境感測事件上 MQTT 能較快速地進行服務的建立。由於在 Local 情境中，封包傳輸於網路卡中就

將資料轉回，故網路效能 TCP 及 UDP 差異不大，由標頭封包較小的 MQTT 傳輸速率最為快速。

$$Performance\ Evaluation = \frac{REST_{Average\ RTT} - MQTT_{Average\ RTT}}{REST_{Average\ RTT}} \times 100\% \quad \text{公式(1)}$$

$$Performance\ Evaluation = \frac{CoAP_{Average\ RTT} - MQTT_{Average\ RTT}}{CoAP_{Average\ RTT}} \times 100\% \quad \text{公式(2)}$$

表 5-3 MQTT、REST、CoAP 物聯網環境感測事件時間效能分析比較表(Local)

Type	MQTT	REST	CoAP
Average	77 ms	263 ms	93 ms
Max	85 ms	275 ms	96 ms
Min	73 ms	251 ms	87 ms

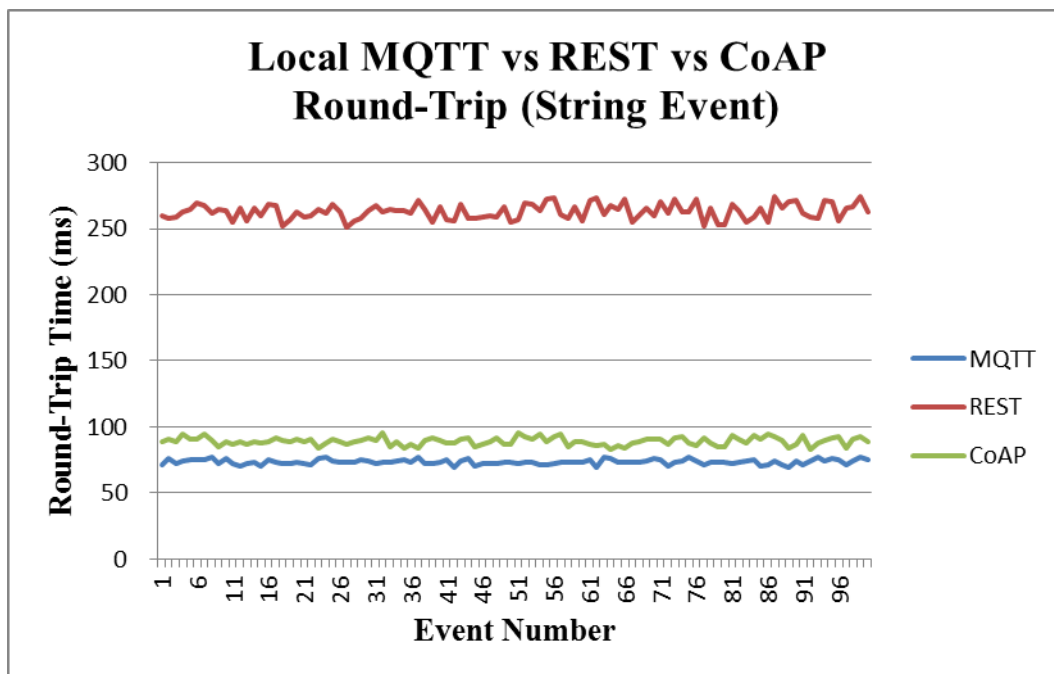


圖 5-4 MQTT、REST、CoAP 物聯網環境感測事件服務往返時間之分析(Local)

於情境 2 藉由分析結果如表 5-4 MQTT、REST、CoAP 物聯網環境感測事件時間效能分析比較表(Intranet)與

圖 5-4 MQTT、REST、CoAP 物聯網環境感測事件服務往返時間之分析(Intranet)所示，分析結果指出，MQTT 所需之平均服務處理時間少於 REST 整體服務時間約 3.1 倍，其平均服務處理時間也少於 CoAP 整體服務時間約 1.4 倍，藉由公式(1)、(2)進行性能的運算評估 MQTT 在建立事件的平均服務處理效能上快於 REST 及 CoAP

約 68 % 及 26 % 的效能處理速度，於 Intranet 情境中物聯網環境感測事件上 MQTT 能較快速地進行服務的建立。由於在 Intranet 情境中，封包傳輸於內部網路，而實驗環境未跨越 Switch，故網路效能 TCP 及 UDP 差異不大，由標頭封包較小的 MQTT 傳輸速率最為快速。

表 5- 4 MQTT、REST、CoAP 物聯網環境感測事件時間效能分析比較表(Intranet)

Type	MQTT	REST	CoAP
Average	89 ms	280 ms	121 ms
Max	98 ms	298 ms	124 ms
Min	83.5 ms	261 ms	115 ms

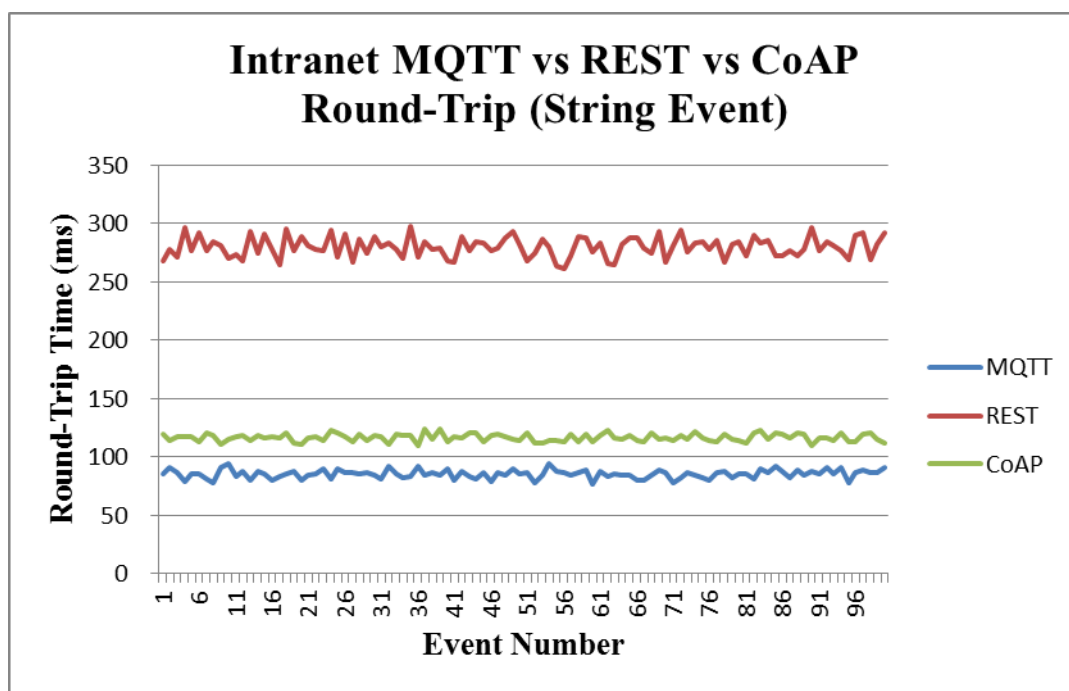


圖 5- 5 MQTT、REST、CoAP 物聯網環境感測事件服務往返時間之分析(Intranet)

於情境 3 藉由分析結果如表 5-5 MQTT、REST、CoAP 物聯網環境感測事件時間效能分析比較表(Internet)與圖 5-6 MQTT、REST、CoAP 物聯網環境感測事件服務往返時間之分析(Internet)所示，分析結果指出 CoAP 所需之平均服務處理時間少於 REST 整體服務時間約 3.3 倍，其平均服務處理時間也少於 MQTT 整體服務時間約 1.8 倍，由圖可見，於情境 3 中，CoAP 服務處理時間較佳，藉由公式(3)、(4)進行性能的運算

評估 CoAP 在 Internet 情境下建立事件的平均服務處理效能上快於 REST 及 MQTT 約 70 % 及 46 % 的效能處理速度，於 Internet 情境中物聯網環境感測事件 CoAP 能較快速地進行服務的建立。由於在 Internet 情境中經過的路由器路徑很多，各應用層傳輸協定的底層效能將在 Internet 情境中顯現出來，由 UDP 設計而成的 CoAP 傳輸協定服務往返時間優於其他兩項。

$$Performance\ Evaluation = \frac{REST\ Average\ RTT - CoAP\ Average\ RTT}{REST\ Average\ RTT} \times 100\% \quad \text{公式(3)}$$

$$Performance\ Evaluation = \frac{MQTT\ Average\ RTT - CoAP\ Average\ RTT}{MQTT\ Average\ RTT} \times 100\% \quad \text{公式(4)}$$

表 5- 5 MQTT、REST、CoAP 物聯網環境感測事件時間效能分析比較表(Internet)

Type	MQTT	REST	CoAP
Average	349 ms	631 ms	190 ms
Max	391 ms	680 ms	204.5 ms
Min	308ms	585 ms	181.5 ms

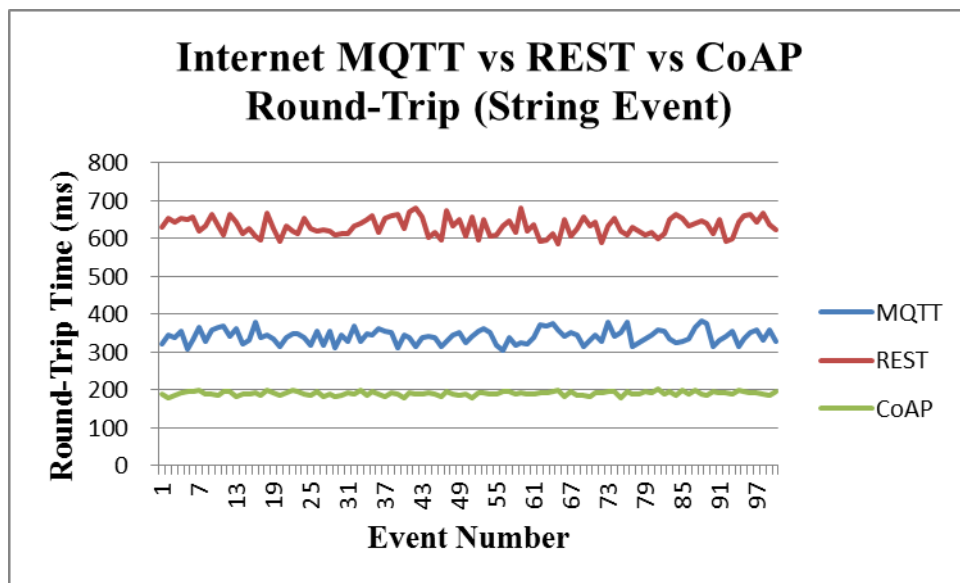


圖 5- 6 MQTT、REST、CoAP 物聯網環境感測事件服務往返時間之分析(Internet)

5.1.2. 建立物聯網影像事件服務往返(Round-Trip)時間效能比較

本實驗分析主要為分析比較 MQTT、REST、CoAP，在建立物聯網影像事件上服

務往返(Round-Trip)時間的效能，影像事件如表 5-6 影像事件說明所示。而為了提高效能比較的正确率與數據的可用性，在測量次數部分，本實驗進行 100 次完整服務的測量，藉此觀察整體服務的平均服務時間與服務建立穩定度。

表 5-6 影像事件說明

影像事件	物聯網環境中，網路攝影機拍攝實境。 大小：180k (184,411 Byte) 傳送方法：利用 Base64 轉置法將影像轉為字串進行傳輸。
------	---

於情境 1 藉由分析結果如表 5-7 MQTT、REST、CoAP 建立物聯網影像事件時間效能分析比較表(Local)與圖 5-7 MQTT、REST、CoAP 建立物聯網影像事件服務往返時間之分析(Local)所示，MQTT 最大服務時間與最小服務時間分別為 250 ms 與 245 ms，差距為 5 ms，REST 大服務時間與最小服務時間分別為 458 ms 與 425 ms，差距為 33 ms，CoAP 大服務時間與最小服務時間分別為 330 ms 與 295 ms，差距為 35 ms。分析結果指出 MQTT 所需之平均服務處理時間少於 REST 整體服務時間約 1.8 倍，其平均服務處理時間也少於 CoAP 整體服務時間約 1.3 倍，藉由公式(1)、(2)進行性能的運算評估 MQTT 在建立事件的平均服務處理效能上快於 REST 及 CoAP 約 44 % 及 21 % 的效能處理速度，於 Local 情境中物聯網影像事件上，由於影像經由 Base64 資料轉置方法，將影像資料轉置成訊息格式，並加以傳送，由 MQTT 較快速地進行服務的建立。由於在 Local 情境中，封包傳輸於網路卡中就将資料轉回，故網路效能 TCP 及 UDP 差異不大，由標頭封包較小的 MQTT 傳輸速率最為快速。

表 5-7 MQTT、REST、CoAP 建立物聯網影像事件時間效能分析比較表(Local)

Type	MQTT	REST	CoAP
Average	247 ms	442 ms	313 ms
Max	250 ms	458 ms	330 ms
Min	245 ms	425 ms	295 ms

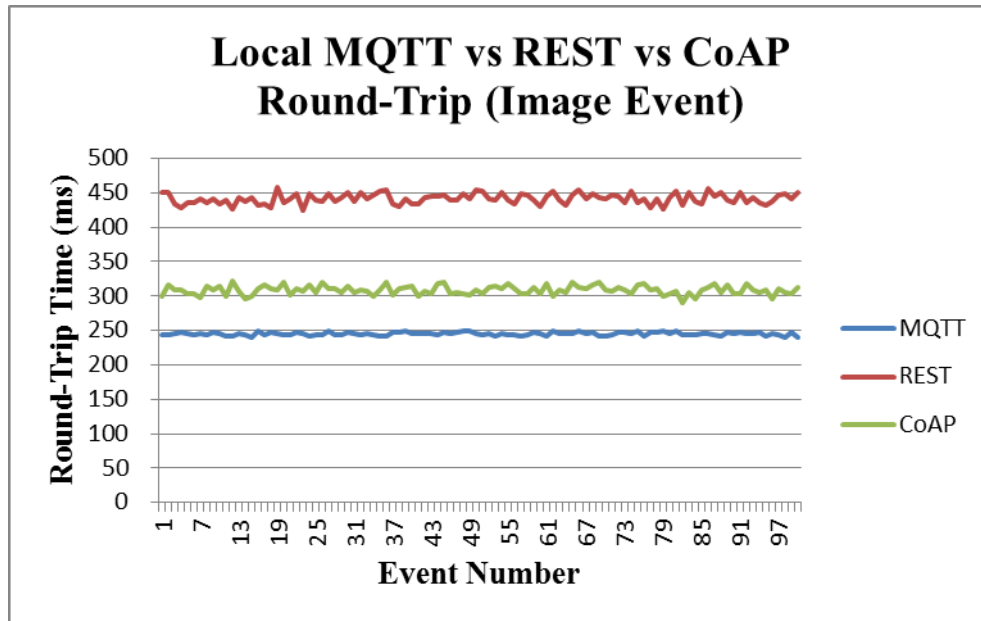


圖 5- 7 MQTT、REST、CoAP 建立物聯網影像事件服務往返時間之分析(Local)

於情境 2 藉由分析結果如表 5-8 MQTT、REST、CoAP 建立物聯網影像事件時間效能分析比較表(Intranet)與圖 5-8 MQTT、REST、CoAP 建立物聯網影像事件服務往返時間之分析(Intranet)所示，分析結果指出，MQTT 所需之平均服務處理時間少於 REST 整體服務時間約 1.7 倍，其平均服務處理時間也少於 CoAP 整體服務時間約 1.1 倍，藉由公式(1)、(2)進行性能的運算評估 MQTT 在建立事件的平均服務處理效能上快於 REST 及 CoAP 約 43 % 及 8.7 % 的效能處理速度，於 Intranet 情境中物聯網影像事件上，由於影像經由 Base64 資料轉置方法，將影像資料轉置成訊息格式，並加以傳送，由 MQTT 較快速地進行服務的建立。由於在 Intranet 情境中，封包傳輸於內部網路，而實驗環境未跨越 Switch，故網路效能 TCP 及 UDP 差異不大，由標頭封包較小的 MQTT 傳輸速率最為快速。

表 5- 8 MQTT、REST、CoAP 建立物聯網影像事件時間效能分析比較表(Intranet)

Type	MQTT	REST	CoAP
Average	264 ms	461 ms	289 ms
Max	280 ms	487 ms	296 ms
Min	250 ms	440 ms	284 ms

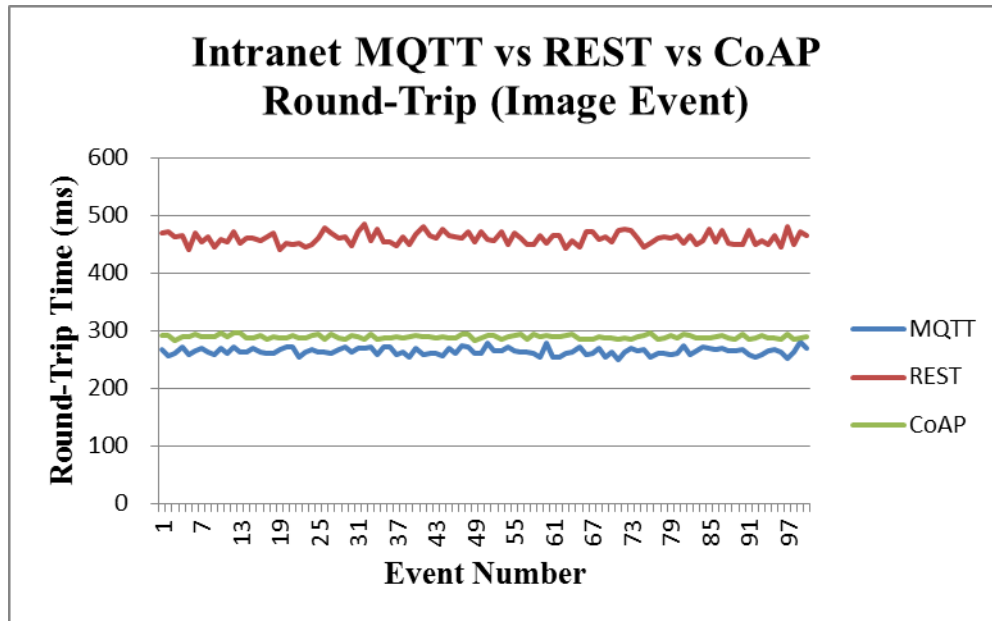


圖 5- 8 MQTT、REST、CoAP 建立物聯網影像事件服務往返時間之分析(Intranet)

於情境 3 藉由分析結果如表 5-9 MQTT、REST、CoAP 建立物聯網影像事件時間效能分析比較表(Internet)與圖 5-9 MQTT、REST、CoAP 建立物聯網影像事件服務往返時間之分析(Internet)所示，分析結果指出，CoAP 所需之平均服務處理時間少於 REST 整體服務時間約 1.9 倍，其平均服務處理時間也少於 MQTT 整體服務時間約 1.3 倍，藉由公式(3)、(4)進行性能的運算評估 CoAP 在建立事件的平均服務處理效能上快於 REST 及 CoAP 約 47 %及 21 %的效能處理速度，於 Internet 情境中建立物聯網影像事件 CoAP 能較快速地進行服務的建立。由於在 Internet 情境中經過的路由器路徑很多，各應用層傳輸協定的底層效能將在 Internet 情境中顯現出來，由 UDP 設計而成的 CoAP 傳輸協定服務往返時間優於其他兩項。

表 5-9 MQTT、REST、CoAP 建立物聯網影像事件時間效能分析比較表(Internet)

Type	MQTT	REST	CoAP
Average	847 ms	1266 ms	665 ms
Max	889 ms	1374 ms	737 ms
Min	793 ms	1172ms	573 ms

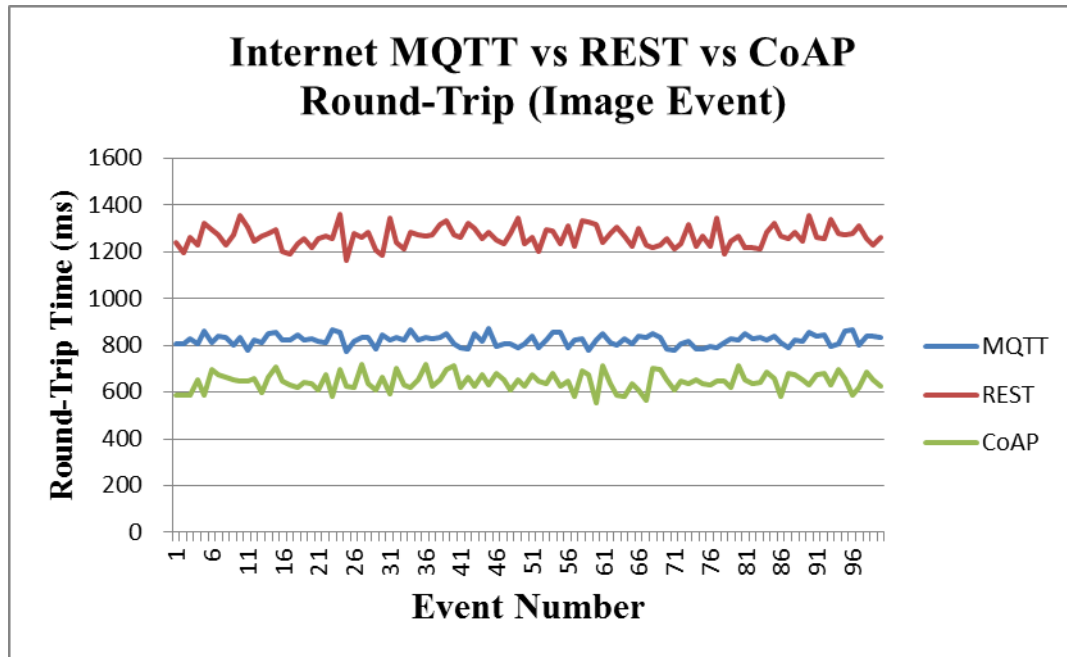


圖 5-9 MQTT、REST、CoAP 建立物聯網影像事件服務往返時間之分析(Internet)

5.1.3. 影片事件(Video Events)服務往返(Round-Trip)時間效能比較

本實驗分析主要為分析比較 MQTT、REST、CoAP，在建立影片事件(Video Events)上服務往返(Round-Trip)時間的效能，由於本實驗環境中，無影像事件之相關設備，故採用一部影片進行實驗分析，影像事件如表 5-10 影像事件說明所示。而為了提高效能比較的正確率與數據的可用性，在測量次數部分，本實驗進行 100 次完整服務的測量，藉此觀察整體服務的平均服務時間與服務建立穩定度。

表 5-1 影片事件說明

影片事件	大小：888k (909,312 Byte) 傳送方法：利用 Base64 轉置法將影像轉為字串進行傳輸。
------	--

於情境 1 藉由分析結果如表 5-11 MQTT、REST、CoAP 建立影片事件時間效能分析比較表(Local)與圖 5-10 MQTT、REST、CoAP 建立影片事件服務往返時間之分析(Local)所示，MQTT 最大服務時間與最小服務時間分別為 441 ms 與 324 ms，差距為 117 ms，REST 大服務時間與最小服務時間分別為 545 ms 與 526 ms，差距為 19 ms，CoAP 大服務時間與最小服務時間分別為 475 ms 與 354 ms，差距為 121 ms，分析結果指出 MQTT 所需之平均服務處理時間少於 REST 整體服務時間約 1.4 倍，其平均服務處理時間也少於 CoAP 整體服務時間約 1.1 倍，藉由公式(1)、(2)進行性能的運算評估 MQTT 在建立事件的平均服務處理效能上快於 REST 及 CoAP 約 29 % 及 9 % 的效能處理速度，在 Local 傳送影片上 MQTT 也能較快速地進行服務的建立，但優異的幅度較前面事件減少。

表 5-2 MQTT、REST、CoAP 建立影片事件時間效能分析比較表(Local)

Type	MQTT	REST	CoAP
Average	378 ms	535 ms	415 ms
Max	441 ms	545 ms	475 ms
Min	342 ms	526 ms	354 ms

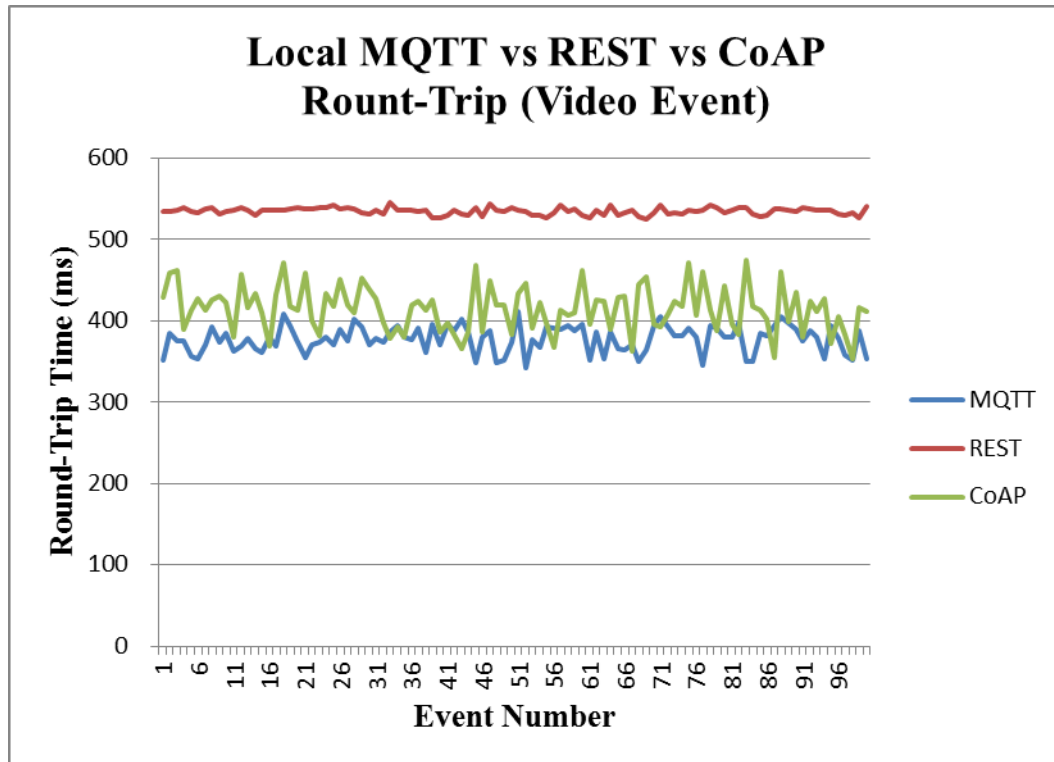


圖 5- 1 MQTT、REST、CoAP 建立影片事件服務往返時間之分析(Local)

於情境 2 藉由分析結果如表 5-12 MQTT、REST、CoAP 建立影片事件時間效能分析比較表(Intranet)與圖 5-11 MQTT、REST、CoAP 建立影片事件服務往返時間之分析(Intranet)所示，分析結果指出，MQTT 所需之平均服務處理時間少於 REST 整體服務時間約 1.4 倍，其平均服務處理時間也少於 CoAP 整體服務時間約 1.1 倍，藉由公式(1)、(2)進行性能的運算評估 MQTT 在建立事件的平均服務處理效能上快於 REST 及 CoAP 約 27 % 及 9 % 的效能處理速度，於 Intranet 的情境下傳送影片上 MQTT 能較快速地進行服務的建立。

表 5- 3 MQTT、REST、CoAP 建立影片事件時間效能分析比較表(Intranet)

Type	MQTT	REST	CoAP
Average	501 ms	685 ms	550 ms
Max	512 ms	692 ms	575 ms
Min	490 ms	679 ms	527 ms

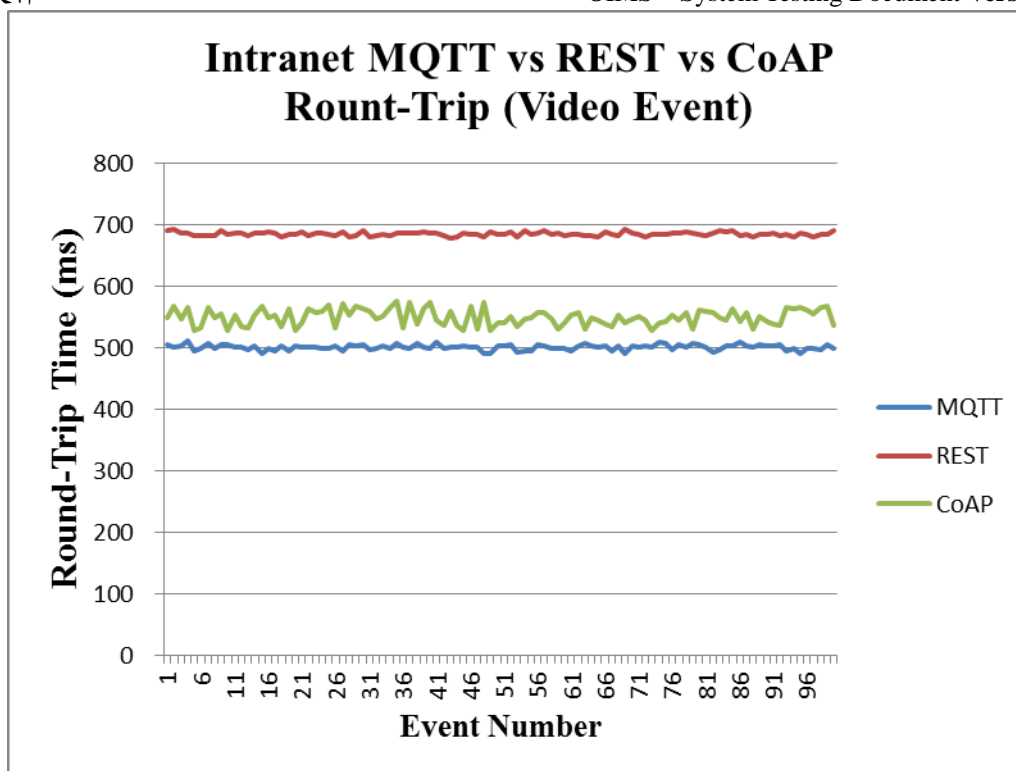


圖 5- 2 MQTT、REST、CoAP 建立影片事件服務往返時間之分析(Intranet)

於情境 3 藉由分析結果如表 5-13 MQTT、REST、CoAP 建立影片事件時間效能分析比較表(Internet)與圖 5-12 MQTT、REST、CoAP 建立影片事件服務往返時間之分析(Internet)所示，分析結果指出，CoAP 所需之平均服務處理時間少於 REST 整體服務時間約 1.4 倍，其平均服務處理時間也少於 MQTT 整體服務時間約 1.1 倍，由，藉由公式(3)、(4)進行性能的運算評估 CoAP 在 Internet 環境中建立影片事件的平均服務處理效能上快於 REST 及 MQTT 約 29%及 10 %的效能處理速度，於 Internet 的情境下傳送影片上 CoAP 能較快速地進行服務的建立。

表 5- 4 MQTT、REST、CoAP 建立影片事件時間效能分析比較表(Internet)

Type	MQTT	REST	CoAP
Average	7252 ms	9282 ms	6545 ms
Max	7888 ms	9905 ms	7021 ms
Min	6603 ms	8608ms	6029 ms

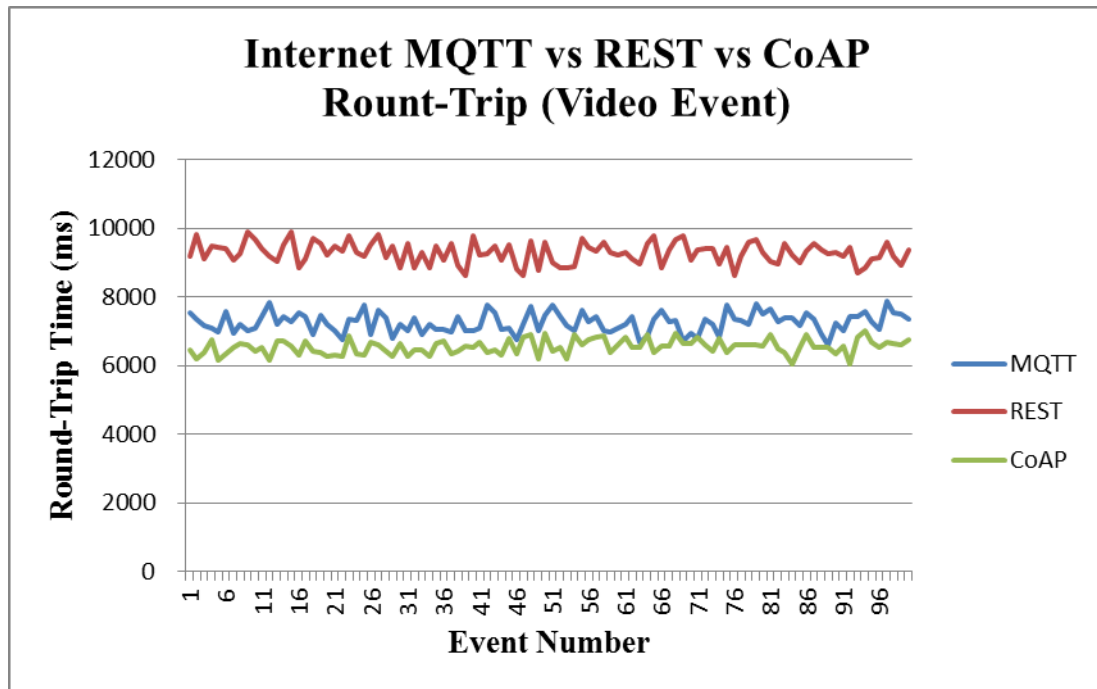


圖 5- 3 MQTT、REST、CoAP 建立影片事件服務往返時間之分析(Internet)

5.1.4. 封包傳輸掉落率(Packet Lost)比較

本實驗分析主要為分析比較 MQTT、REST、CoAP，在在傳輸事件上的封包到達率之效能，測試事件如表 5-14 封包掉落率測試說明所示。而為了提高效能比較的正确率與數據的可用性，本實驗事先確認三種傳輸協定皆分應用層傳輸協定，由於 CoAP 設計上有資料重傳之實作，於本實驗中，取消 CoAP 重傳之機制，透過最原始之傳輸協定進行實驗測試，在測量次數部分，本實驗進行 10000 次服務的測量，藉此觀察整體傳輸封包的掉落率。

表 5- 5 封包掉落率測試說明

測試事件	測試環境：Intranet、Internet。 感測器數量：5 個。 測試數量：10000 次。 發送速率：10 次/s。
------	--

實驗結果，藉由公式(5)計算掉落封包率，計算數據如表 5-15 封包掉落率實驗數據，由測試分析結果，如圖 5-13 各通訊協定之封包掉落率分析圖所示，於資料可靠

傳輸上，由 REST 表現最為優異，其封包掉落率於 Intranet 僅 0.07 %，而在 Internet 情境，掉落率僅 0.86 %，而採用 UDP 進行傳輸的 CoAP，於資料可靠傳輸上較為不利，於 Intranet 情境中，CoAP 掉落率為 29.17 %，於 Internet 情境中，掉落率為 38.66 %。

$$Packet\ Lost_{(Protocol)} = \frac{Testing\ Times - Arrive\ Times}{Testing\ Times} \times 100\% \quad \text{公式(5)}$$

表 5- 6 封包掉落率實驗數據

	MQTT	REST	CoAP
Intranet 封包到達次數	9925	9993	7083
Internet 封包到達次數	9653	9914	6134

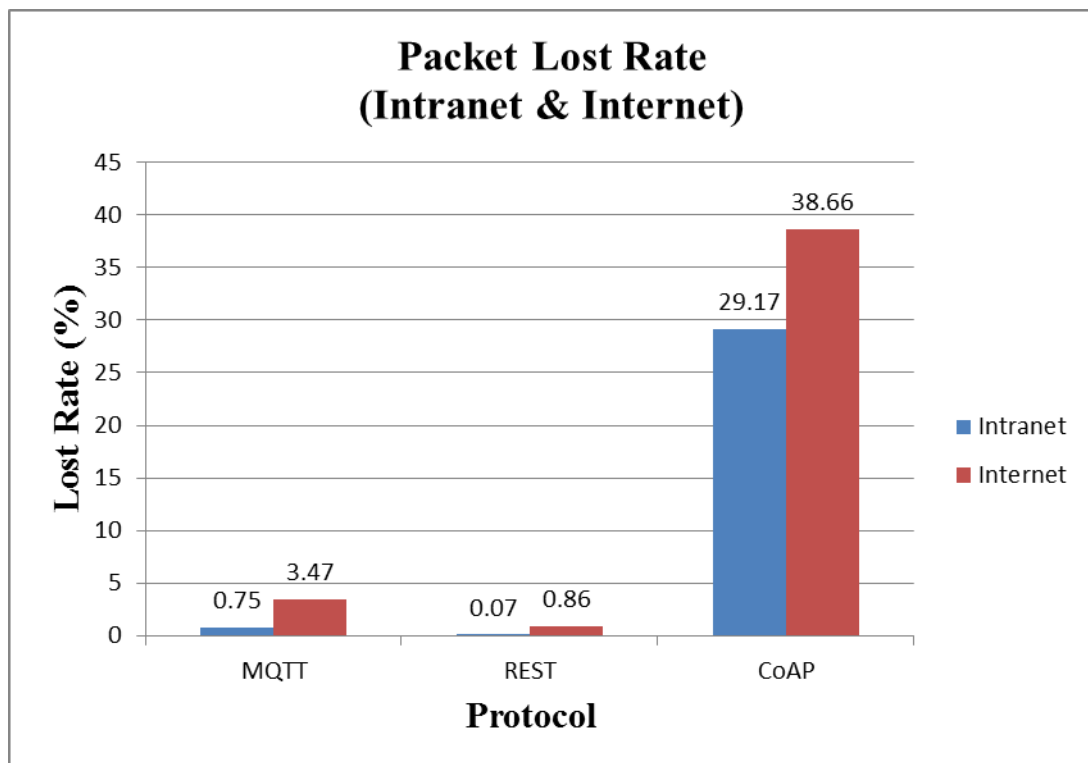


圖 5- 4 各通訊協定之封包掉落率分析圖

6. Test Result and Analysis

Integration Testing Cases

如表 6.1 所示，依據上述的 test case 並進行實際系統測試，其結果如下所示。

表 6.1 計畫測試結果

Test Case	Result(Pass / Fail)	Comment
AT1	Pass	完全符合測試狀況。
AT2	Pass	完全符合測試狀況。
AT3	Pass	完全符合測試狀況。
AT4	Pass	完全符合測試狀況。
IT1	Pass	完全符合測試狀況。
IT2	Pass	完全符合測試狀況。
IT3	Pass	完全符合測試狀況。
IT4	Pass	完全符合測試狀況。
RATE	100 %	完全符合測試狀況。

7. Appendix A: Glossary

Test Case

Test Case is a commonly used term for a specific test. This is usually the smallest unit of testing. A Test Case will consist of information such as requirements testing, test steps, verification steps, prerequisites, outputs, test environment, etc.

A set of inputs, execution preconditions, and expected outcomes developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test Environment

The hardware and software environment in which tests will be run, and any other software with which the software under test interacts when under test including stubs and test drivers.

Testing

The process of exercising software to verify that it satisfies specified requirements and to detect errors. The process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs), and to evaluate the features of the software item (Ref. IEEE Std 829).

The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component.

Test Procedure

A document providing detailed instructions for the execution of one or more test cases.

Traceability Matrix

A document showing the relationship between Test Requirements and Test Cases.

Validation The process of evaluating software at the end of the software development process to ensure compliance with software requirements. The techniques for validation is testing, inspection and reviewing.

Verification

The process of determining whether or not the products of a given phase of the software development cycle meet the implementation steps and can be traced to the incoming objectives established during the previous phase. The techniques for verification are testing, inspection and reviewing.

Equivalence Class

A portion of a component's input or output domains for which the component's behavior is assumed to be the same from the component's specification.

Acceptance Testing

Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

8. Appendix B: Traceability

水平追蹤矩陣（Horizontal Traceability Matrix）

（●：表示有關聯）

	AT1	AT2	AT3	AT4	IT1	IT2	IT3	IT4
OIMS -N-001	●			●	●		●	
OIMS -N-002	●	●		●			●	
OIMS -N-003		●				●		
OIMS -N-004	●				●		●	
OIMS -N-005		●	●	●				●
OIMS -N-006		●		●				●
OIMS -N-007	●		●		●			
OIMS -N-008	●				●	●		
OIMS -N-009	●		●		●			
OIMS -N-010		●		●				●
OIMS -N-011				●				●
OIMS -N-012			●		●		●	
OIMS -N-013	●				●			
OIMS -N-014	●		●		●			●
OIMS -N-015				●			●	
OIMS -N-016	●		●		●			
OIMS -N-017	●	●				●	●	
OIMS -N-018		●	●	●				●
OIMS -N-019		●		●				●

OIMS -N-020	●		●			●		
OIMS -N-021	●		●			●		
OIMS -N-022		●		●				●
OIMS -N-023		●	●			●		
OIMS -N-024	●		●			●		
OIMS -N-025		●		●	●			
OIMS -N-026	●		●		●			●
OIMS -N-027	●			●	●			
OIMS -N-028	●		●			●		
OIMS -N-029		●		●				
OIMS -N-030	●	●				●		
OIMS -N-031	●					●		●
OIMS -N-032			●	●			●	
OIMS -N-033	●	●		●				

	AT1	AT2	AT3	AT4	IT1	IT2	IT3	IT4
HPIT-N-001	●			●				
HPIT-N-002	●						●	
HPIT-N-003			●					●
HPIT-N-004		●			●			
HPIT-N-005			●			●		
HPIT-N-006			●			●		
HPIT-N-007				●				●

HPIT-N-008	●				●			
HPIT-N-009		●					●	
HPIT-N-010		●						
HPIT-N-011		●					●	
HPIT-N-012				●		●		
HPIT-N-013			●	●		●		
HPIT-N-014	●							
HPIT-N-015	●						●	
HPIT-N-016	●			●	●			
HPIT-N-017	●		●			●		
HPIT-N-018		●		●				
HPIT-N-019	●	●				●		
HPIT-N-020	●					●		●
HPIT-N-021			●	●			●	
HPIT-N-022	●	●		●				
HPIT-N-023			●			●		
HPIT-N-024		●			●			
HPIT-N-025		●					●	

	AT1	AT2	AT3	AT4	IT1	IT2	IT3	IT4
GMSD -N-001		●		●		●		
GMSD -N-002		●				●	●	

GMSD -N-003	●				●			
GMSD -N-004			●					●
GMSD -N-005		●						
GMSD -N-006	●				●			●
GMSD -N-007			●					
GMSD -N-008		●				●		
GMSD -N-009	●				●			
GMSD -N-010		●	●				●	
GMSD -N-011					●			●
GMSD -N-012			●			●		
GMSD -N-013	●				●	●		
GMSD -N-014		●		●				
GMSD -N-015		●			●			
GMSD -N-016		●	●				●	
GMSD -N-017						●		
GMSD -N-018		●				●		
GMSD -N-019	●			●	●			
GMSD -N-020	●		●			●		
GMSD -N-021		●		●				
GMSD -N-022	●	●				●		
GMSD -N-023	●					●		●
GMSD -N-024			●	●			●	
GMSD -N-025			●	●			●	
GMSD -N-026			●			●		
GMSD -N-027		●		●		●		

GMSD -N-028		●				●		
-------------	--	---	--	--	--	---	--	--

	AT1	AT2	AT3	AT4	IT1	IT2	IT3	IT4
HDIM -N-001	●		●					●
HDIM -N-002			●			●		
HDIM -N-003	●				●			
HDIM -N-004			●			●		
HDIM -N-005		●					●	
HDIM -N-006						●		
HDIM -N-007		●			●			
HDIM -N-008			●			●	●	
HDIM -N-009				●			●	
HDIM -N-010		●	●					●
HDIM -N-011					●		●	
HDIM -N-012		●						
HDIM -N-013	●		●			●		
HDIM -N-014		●		●				
HDIM -N-015	●	●				●		
HDIM -N-016	●					●		●
HDIM -N-017			●	●			●	
HDIM -N-018			●					
HDIM -N-019		●					●	
HDIM -N-020			●	●			●	
HDIM -N-021			●			●		
HDIM -N-022					●		●	

HDIM -N-023		●						
HDIM -N-024	●		●			●		
HDIM -N-025		●		●				

	AT1	AT2	AT3	AT4	IT1	IT2	IT3	IT4
OSPS -N-001		●		●			●	●
OSPS -N-002							●	
OSPS -N-003							●	
OSPS -N-004		●		●				●
OSPS -N-005		●		●				
OSPS -N-006	●			●				
OSPS -N-007				●				●
OSPS -N-008				●				●
OSPS -N-009	●					●		
OSPS -N-010			●					
OSPS -N-011		●			●			
OSPS -N-012	●			●	●			
OSPS -N-013	●		●			●		
OSPS -N-014		●		●				
OSPS -N-015	●	●				●		
OSPS -N-016	●					●		●
OSPS -N-017			●	●			●	
OSPS -N-018	●	●		●				
OSPS -N-019				●		●	●	
OSPS -N-020		●			●			

OSPS -N-021			●			●		
-------------	--	--	---	--	--	---	--	--

9. Reference

- [1] Alexandros Kaloxylos, Aggelos Groumas, Vassilis Sarris, Lampros Katsikas, Panagis Magdalinos, Eleni Antoniou, Zoi Politopoulou, Sjaak Wolfert, Christopher Brewster, Robert Eigenmann, Carlos Maestre Terol, “A cloud-based Farm Management System: Architecture and implementation” , Computers and Electronics in Agriculture, 100, 168-179,2014.
- [2] Denis H, “Middleware,” Free On-line Dictionary of Computing, Retrieved on Jan 2009.
- [3] Duan Yan-e, “Design of Intelligent Agriculture Management Information System Based on IoT”, IEEE Intelligent Computation Technology and Automation, Vol.1, pp. 1045-1049, 2011.
- [4] Ji-chun Zhao, Jun-feng Zhang, Yu Feng, Jian-xin Guo, “The Study and Application of the IOT Technology in Agriculture”, IEEE Computer Science and Information Technology, Vol. 2, pp. 462-465, 2010.
- [5] He, M., Ren, C., Wang, Q., Shao, B. and Dong, J., “The Internet of Things as an Enabler to Supply Chain Innovation,” Proceedings of International Conference on e-Business Engineering.
- [6] Lin, M. and Zhang, J., “The Application and Development of Internet of Things with its Solutions of Restrictive Factors,” Proceedings of International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pp.282-285, 2011.
- [7] Ren Duan, Xiaojiang Chen, Tianzhang Xing, “A QoS Architecture for IOT,” IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing, 2011.
- [8] Soma Bandyopadhyay, Munmun Sengupta, Souvik Maiti and Subhajit Dutta, “ROLE OF MIDDLEWARE FOR INTERNET OF THINGS: A STUDY,” International

Journal of Computer Science & Engineering Survey (IJCSSES) Vol.2, No.3, August 2011.

- [9] Tomasz Kobialka, Rajkumar Buyya, Christopher Leckie, Ramamohanarao Kotagiri, “A Sensor Web Middleware with Stateful Services for Heterogeneous Sensor Networks, ” IEEE ISSNIP, 2013, pp. 491 –496.
- [10] Yu, C., “Research and Design of Logistics Management System based on Internet of Things,” Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), pp. 6314-6317, 2011.
- [11] Krakowiak, and Sacha, “What’s middleware”, ObjectWeb.org, 2003, Retrieved on jan 2009.
- [12] Yelin HONG, “A Resource-Oriented Middleware Framework for Heterogeneous Internet of Things, ” International Conference on Cloud Computing and Service Computing, 2012, pp. 12 –16.
- [13] Mauro Caporuscio, Marco Funaro, Carlo Ghezzi, “Resource-oriented Middleware Abstractions For Pervasive Computing, ” IEEE International Conference on Software Science, Technology and Engineering, 2012.
- [14] JÖRG SWETINA, GUANG LU, PHILIP JACOBS, FRANCOIS ENNESSER, JAESEUNG SONG, “TOWARD A STANDARDIZED COMMON M2M SERVICELAYER PLATFORM: INTRODUCTION TO ONEM2M”, IEEE Wireless Communications, Vol. 21, Issue. 3, pp.20-26, 2014.
- [15] Elmangoush, A. , Steinke, R. , Al-Hezmi, A. , Magedanz, T. , “On The Usage of Standardised M2M Platforms for Smart Energy Management”, IEEE Information Networking (ICOIN), pp.79-84, Feb. 2014.
- [16] SungHyup Lee, KyoungKeun Kim, WonGyu Jang, Anh Ngoc Le, “Service-adaptive functional architecture in M2M communications”, IEEE Ubiquitous and Future

Networks, pp. 735-737, July. 2013.

[17] oneM2M, “oneM2M Functional Architecture Baseline Draft”, Aug. 2014.

[18] Thangavel, D., Xiaoping Ma, Valera, A., Hwee-Xian Tan, Tan, C.K.-Y.,
“Performance evaluation of MQTT and CoAP via a common middleware”, pp. 1-6,
2014.

[19] IBM, “MQ Telemetry Transport (MQTT) V3.1 Protocol Specification”, 2010.

[20] Giang, N.K., “SCoAP: An integration of CoAP protocol with web-based application”,
IEEE Global Communications Conference, pp.2648-2653, 2013.

[21] Yohanes Baptista Dafferianto Trinugroho, Martin Gerdes, Mohammad Mahdi ahdavi
Amjad, Frank Reichert and Rune Fensli, “A REST-Based Publish/Subscribe Platform
to Support Things-to-Services Communications, ” Asia-Pacific Conference on
Communications (APCC), Bali –Indonesia, 2013, pp. 321 –326.

[22] Stefano Turchi, Lorenzo Bianchi, Federica Paganelli, Franco Pirri, Dino Giuli,
“Towards a Web of Sensors built with Linked Data and REST, ” IEEE, 2013.