

大專學生研究計畫研究成果報告

* 計畫 *

* : 探討 LoRa 和 LoRaWan 的通訊傳輸模式

*

* 名稱 *

執行計畫學生：李彥儒

研究期間： 110 年 9 月 1 日至 111 年 6 月底止，計 9 個月

指導教授：王超教授

處理方式(請勾選)：☒立即公開查詢

☐涉及專利或其他智慧財產權，

☐一年☐二年後可公開查詢

執行單位：師範大學資訊工程學系

中華民國 1 1 1 年 6 月 21 日

目錄

目錄	I
摘要	II
Abstract	III
研究動機與問題	1
文獻回顧與探討	2
研究方法與步驟	3
參考文獻	19

摘要

現今，信息技術高度發達，繼網際網路之後，物聯網以迅雷不及掩耳之勢引發社會廣泛的關注。WiFi、Bluetooth、行動網路等等的無線通信傳輸協議，使得構建物聯網時有了豐富的選擇，但是對於長距離，低功耗，只有少量資料需要傳輸的應用場景，過往還沒有能更滿足上述需求的無線通訊技術，這也就是 LoRa 技術誕生的原因。LoRa 為英文 Long Rang 的縮寫，為低功耗廣域網路(Low Power Wide Area Network, LPWAN)通信技術的一種，Semtech 公司於 2013 年發布的超長距離低功耗數據傳輸技術。LoRa 無線技術的出現，改變了關於傳輸距離與功耗的折衷考慮方式，不僅可以實現遠距離傳輸，並且同時兼具低功耗、低成本的優點。本文將藉由了解 LoRa 傳輸技術的兩種傳輸模式：(1)加入 LoRaWan 進行雲端傳輸和(2)LoRa 終端節點之間的資料傳輸(Peer-to-Peer)，研究兩者傳輸模式的架構和限制，並探討將兩者傳輸模式結合的可行性。設計目標為延續 LoRa 傳輸技術低功耗的優勢，並且讓 LoRa 資料間的傳輸效益最大化。

關鍵詞：LoRa、LoRaWan、節點間傳輸、長距離、低功耗、物聯網、時間同步、

無線通信傳輸、廣域網路

Abstract

Nowadays, information technology is highly developed. After the Internet Network, the Internet of Things technology has attracted widespread attention in society at a rapid pace. Wireless communication transmission protocols such as WiFi, Bluetooth, mobile network, etc., provide a wealth of choices when building the Internet of Things. However, the application scenario for long-distance, low-power, and only a small amount of data to be transmitted, has not been able to be satisfied by the above-mentioned wireless communication technology. This is the reason why LoRa technology was born. LoRa is the abbreviation of Long Range. It is a kind of Low Power Wide Area Network (LPWAN) communication technology. Semtech released the ultra-long-distance low-power data transmission technology in 2013. The emergence of LoRa wireless technology has changed the compromise between transmission distance and power consumption. It can not only achieve long-distance transmission, but also have the advantages of low power consumption and low cost. In this paper, by understanding the two transmission modes of LoRa transmission technology: (1) Adding LoRaWan for cloud transmission and (2) Data transmission between LoRa terminal nodes, we will study the architecture and limitations of the two transmission modes, and discuss how to combine the two transmission modes. The design goal is to continue the advantages of low power consumption of LoRa transmission technology and maximize the transmission efficiency between LoRa data communication.

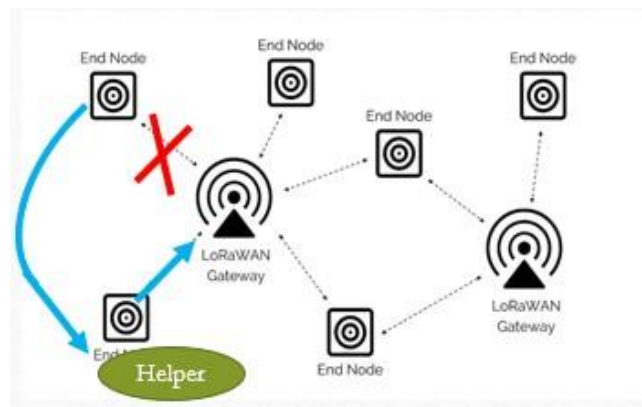
Keywords: LoRa 、 LoRaWan 、 Peer-to-Peer Communication 、 long-distance 、
low power consumption 、 Internet of Things 、 Time synchronization 、

Wireless communication transmission

研究動機與問題

動機：

在 LoRa 的通訊網中，LoRa 終端與 Gateway 間的通訊常常受周遭環境的影響而發生資料傳輸失敗的情況，以致 LoRa 終端有時會與 LoRaWan 失去連線而無法及時將資料上傳到雲端伺服器。本研究探討 LoRa 終端裝置無法有效與 LoRaWan 連接導致資料丟失過多的情況時，如何藉由終端與終端之間的傳輸方式(P2P)，將失去聯繫的 LoRa 終端資料傳給其他仍與 LoRaWan 連線的裝置後，讓有即時性的資料能在時限內傳給 LoRaWan 的雲端伺服器。



問題：

1. 如何結合 LoRa 終端節點之間的 Peer-to-Peer 傳輸方式，將失去聯繫的 LoRa 終端資料傳給尚未與 LoRaWAN 連接的 LoRa 終端節點(Helper)，並藉由 Helper 加入 LoRaWan 將資料傳給雲端伺服器
2. 當有兩個以上的 LoRa 終端裝置時，如何選擇最適合的 LoRa 終端裝置當做 Helper 做資料傳輸，並提出一套選擇流程。

文獻回顧與探討

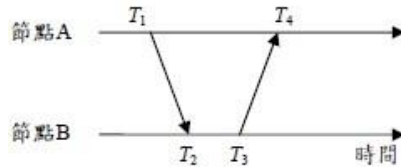
在LoRa終端節點之間的傳輸模式中，時間同步對於增加有效訊息傳送的次數以及減少能量耗損是一個很重要的課題，而在目前在無線感測器網路中，有許多已發表之時間同步方法。這邊將介紹由Ganeriwal, Kumar, and Srivastava所提出一個用於無線感測網路中的時間同步的方法。

Timing-sync Protocol for Sensor Networks (TPSN)

此方法假設網路中擁有一個參考點，而所有節點對照此節點的時間。他提出的演算法包含了兩個步驟，第一步驟為Level discovery phase，第二步驟為Synchronization phase。

(1) Level discovery phase：一開始網路中節點進入此階段，會建立一個階層式樹狀結構。其中，參考點指定為level 0，每經過一個節點，會將自己的level數增加1單位。level值代表此節點到達參考點之最小步數(hops)，同一個level數的節點代表距離參考點的步數相同，由此進入下一個階段。

(2) Synchronization phase：此階段將開始進入時間同步的步驟。由於Level將網路連結成樹狀架構，樹狀架構中的父子節點開始兩兩進行同步，由Root（參考點）往下延伸。屬於level i的節點將與level i-1的節點進行時間同步，每個節點都與自己的父節點交換兩個同步訊息，此動作延伸直到Leaf節點。最終，網路中所有的節點時間將可與參考點的時間一致。



同步兩節點時間的方法，將利用兩個節點在很短時間內，互相交換兩個封包。如上圖所示，節點A、B欲進行時間同步。圖中T1、T4分別代表節點A的時間，T2、T3則分別為節點B的時間。假設在同步封包傳送過程中，兩個節點之時間偏差值(offset)為固定值 Δ ，兩節點封包傳遞延遲時間均為d。

首先，節點A開始同步的動作，於時間T1傳送一個Synchronization_pulse封包給節點B，並且將T1放入此Synchronization_pulse封包中，節點B於T2時間收到此封包，則 $T2 = T1 + \Delta + d$ ；完成接收封包後，節點B回覆一個封包至節點A，且於封包中加入封包送出時間T3，節點A於T4接收此封包，則 $T4 = T3 - \Delta + d$ 。節點A可以將所收集之時戳資料，代入以下公式，計算出 Δ 與d值。

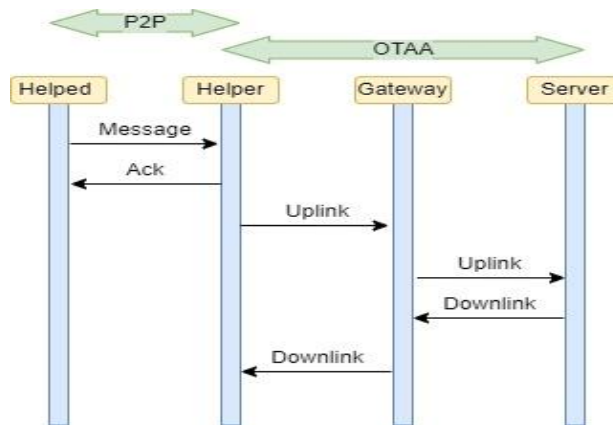
$$\Delta = [(T2 - T1) + (T4 - T3)] / 2$$

$$d = [(T2 - T1) + (T4 - T3)] / 2$$

若節點A要與節點B同步，只要將本身的时间加上算出與節點B的時間偏差值 Δ 並減去兩節點封包傳遞延遲時間d，即可完成時間同步。此方式不只可以將兩節點的時間同步，也可以有效的縮小封包傳送延遲所造成的時間誤差值。

研究方法與步驟

此研究步驟將先探討 LoRa 終端節點之間的傳輸模式(P2P)，深入了解如何減少節點間傳輸時的能量損耗，再研究 LoRa 終端加入 LoRaWan 的傳輸模式(OTAA)和架構，並提出有效結合 LoRa 兩種傳輸模式的方案，最後討論 LoRa 終端有多個 Helper 時，如何選擇更符合效益的 Helper 作傳輸，並比較其差異。

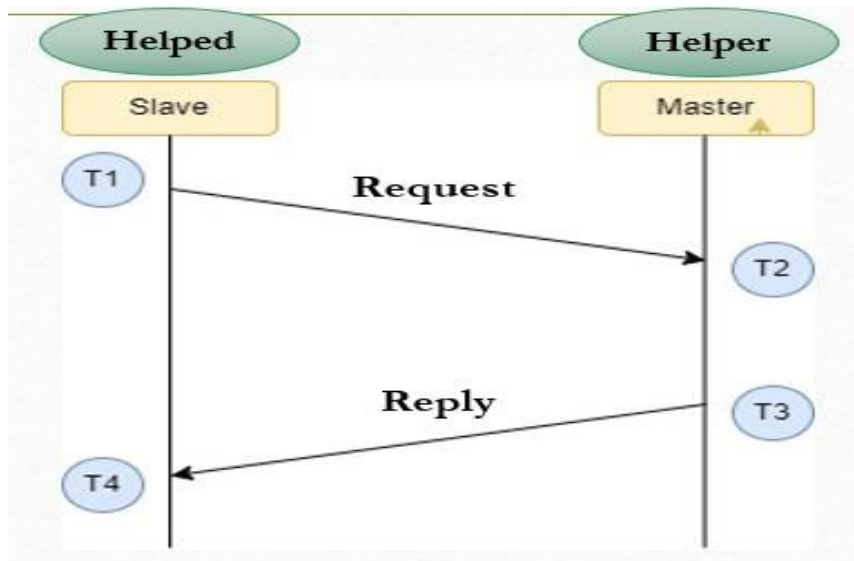


一、 LoRa 終端節點之間的傳輸模式

(Peer-to-Peer Communication Mode)

LoRa 是一種基於物理 (PHY) 層的射頻載波信號，可將其接收到的數據轉換為信號。其發送窗口與接收窗口無法同時開啟，故常使用傳送窗口與接收窗口交替開啟的傳輸模式。

而LoRa終端節點與節點之間的傳輸有時會遇到一方發送時間與另一方接收時間不一致的情況，導致傳送封包的訊息遺失。為了減少傳送訊息的遺失，可能會選擇增加相同訊息的發送頻率，或是延長接收訊息窗口的時間。以上方法皆會導致更多的能量耗損。這邊我們採用時間同步**Timing-sync Protocol for Sensor Networks (TPSN)**的方法，將兩個終端節點開發版中的內部時鐘時間同步，達到控制兩方發送時間與接收時間同步一致的目標。



如上圖的流程圖中，與LoRaWan失聯的終端節點(Helped)將作為Slave clock，將和LoRaWan中還有聯繫的終端節點(Helper) Master clock作時間同步。

假設節點之時間偏差值(offset)為固定值 Δ ，兩節點封包傳遞延遲時間均為 d 。

- (1) 首先由Helped終端節點發出一個封包 (Request) 給Helper終端節點，封包中將紀錄Helped終端節點發送的時間T1。
- (2) Helper終端節點接收到封包後，會記錄收到的時間T2，並將T2減去T1的時間作儲存。Helper終端節點再回復一個封包(Reply)給Helped終端節點，封包中將記錄發送時間T3及T2-T1的時間。
- (3) 最後Helped終端節點紀錄收到回覆封包的時間T4，藉由T2-T1, T3, T4計算出時間偏差值 Δ 和延遲時間 d 。
- (4) Helped終端節點將內部時鐘設為自身當前時間Slave clock加上 Δ 減去 d ，其時間將與Helper終端節點的內部時間Master clock相同，達成時間同步。

實作結果：

下圖中，左方視窗為Helped終端節點的輸出結果，右方視窗為Helper終端節點得輸出結果。其中Real time為電腦標準時間，clock time為內部時間，以 millisecond(ms)為單位顯示。可看到標準時間04:53:19.8時，Helped終端節點內部時鐘為58914 ms，Helper終端節點內部時鐘為24505 ms。

在標準時間 04:53:26 時，Helped 終端節點與 Helper 終端節點完成時間同步。Helper 終端節點內部時鐘變為 30115 ms，與 Helper 終端節點的內部時鐘 30658ms 只相差了約 0.5 秒。



分析時間同步誤差的可能原因：

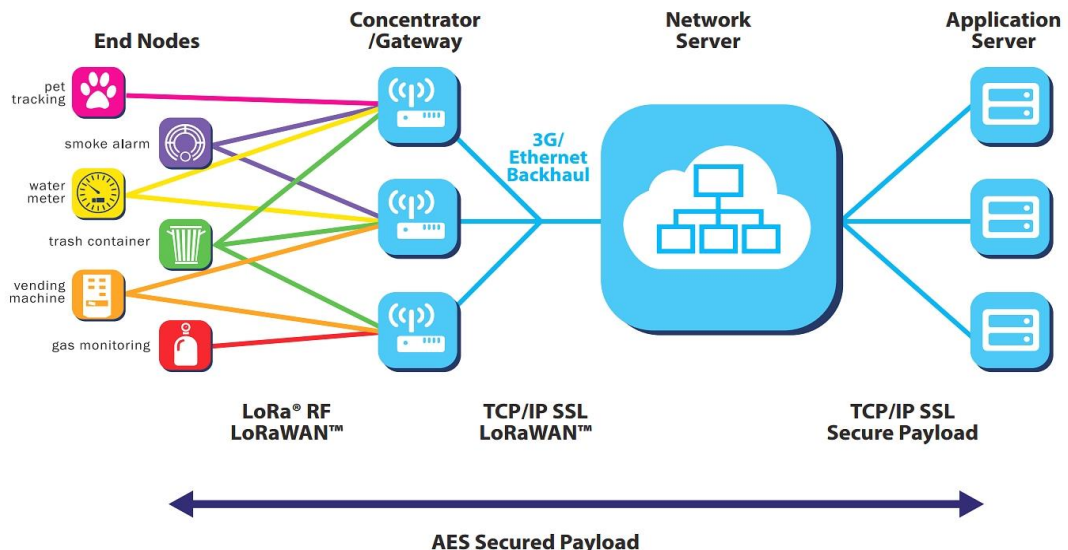
(1) 兩節點來回封包傳遞延遲時間d不固定

在TPSN時間同步方法中，我們假設來回封包的傳輸延遲時間固定為d，但是實際上兩個節點間收送封包的延遲時間並不會固定不變，此誤差可能造成計算上的錯誤。

(2) 無線裝置內計算時間的振盪器，震盪頻率有差異

振盪器為開發版中計算內部時間的重要儀器。但是相同型號的無線裝置內的振盪器，震盪頻率也可能各不相同，在不同的自然環境影響，頻率可能會有所改變，計算內部時鐘的數值也會隨之受到影響。

二、 LoRa 終端與 LoRaWan 的傳輸模式(Join LoRaWan)



1.架構介紹：

1. End Nodes –代表終端裝置或傳感器
2. Gateway – 從多個端節點收集或整合數據
3. Network Server – 整合來自網關的數據並上傳到應用服務器
4. Application Server – 處理或顯示 Network Server 傳來的數據

LoRa 終端節點包含收集有價值的信息，並通過將 LoRa 資料數據藉由廣播傳送到附近的 LoRaWAN 網關(gateway)，LoRaWan 網關再將數據資料傳輸到雲端。

由於 LoRa 終端節點發送端廣播範圍內的所有網關都能接收到數據，這種傳輸創建了一個星型拓撲，並促進了 LoRaWAN 網絡中的密集互連和可靠性。

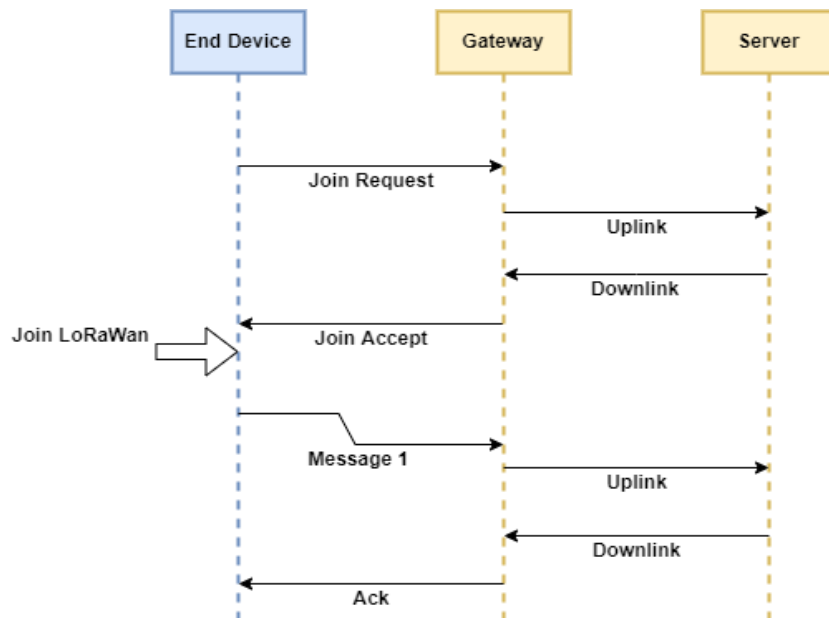
2. 加入 LoRaWan 的方式：

LoRa 終端裝置要加入 LoRaWan 需要先獲取三個加密參數：

DevAddr–Device Address

NwkSkey–Network Session Key
AppSkey–Application Session Key

1. OTAA(Over The Air Activation)



如上圖所示，此加入方式 LoRa 終端裝置需要先發送加入請求(Join Request)，並藉由網關將請求上傳到雲端伺服器。雲端伺服器會下傳加密參數資訊，並由網關將資訊(Join Accept)發送回 LoRa 終端裝置。最後 LoRa 終端裝置再藉由 Join Accept 計算出三個加密參數，並加入 LoRaWan，開始傳輸資料給雲端伺服器。

2. ABP(Activate By Personalization)

LoRa 終端裝置連線到 LoRaWan 時將直接被配置三個加密參數，省略發送請求加入 LoRaWan 的入網流程。

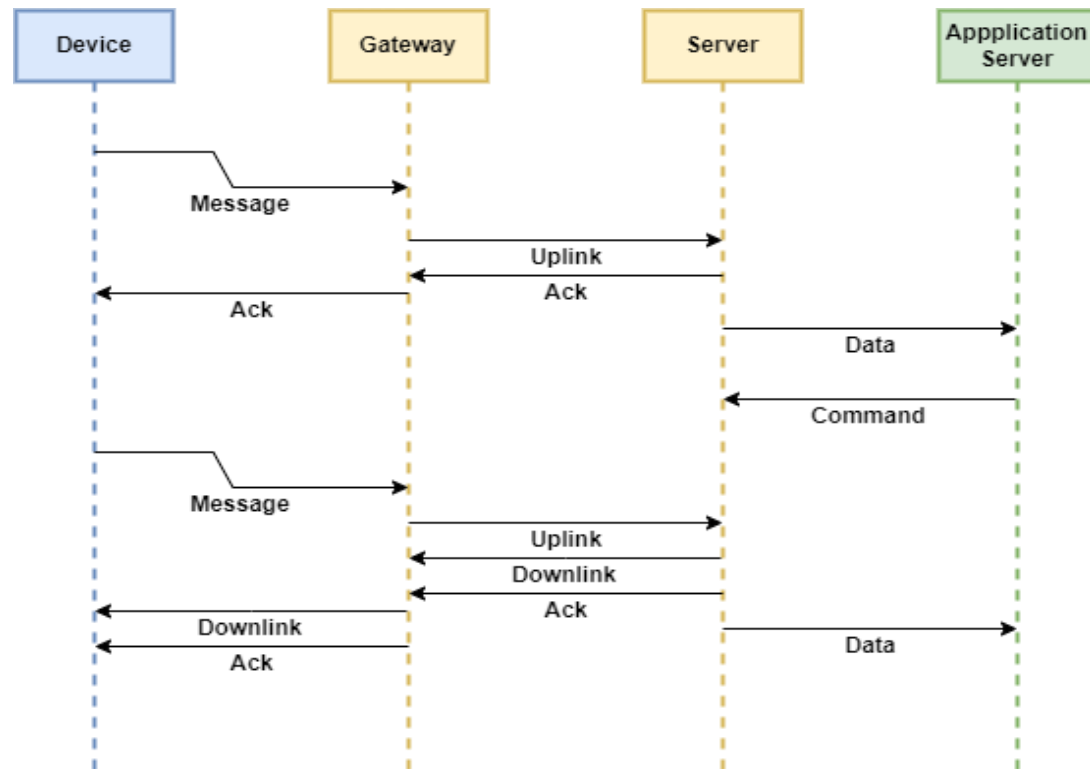
優點：減少每次重新入網所需要花費的時間。並且由於不需要終端裝置收到 Join Accept 才能加入 LoRaWan 的流程，能避免終端裝置在弱網覆該區域時，因終端裝置接收網關訊息能力較差而發生無法加入 LoRaWan 的情況。

缺點：因為直接配置三個加密參數，當這些加密參數洩漏出去時會因為無法更新加密參數而有嚴重的資安風險。並且由於此加入方式的終端裝置加入 LoRaWan 時有傳輸次數限制(Frame count)，當到達次數限制時會無法再與 LoRaWan 傳輸資料。

有鑑於入網時間較少的優勢，本次研究將使用 ABP 做為加入

LoRaWan 的方式。

3.傳輸流程



這邊將介紹加入 LoRaWan 後 LoRa 終端裝置與雲端伺服器以及應用程式伺服器的詳細傳輸流程，其機制與之後設計的方案將有密切關係。

上傳資料：

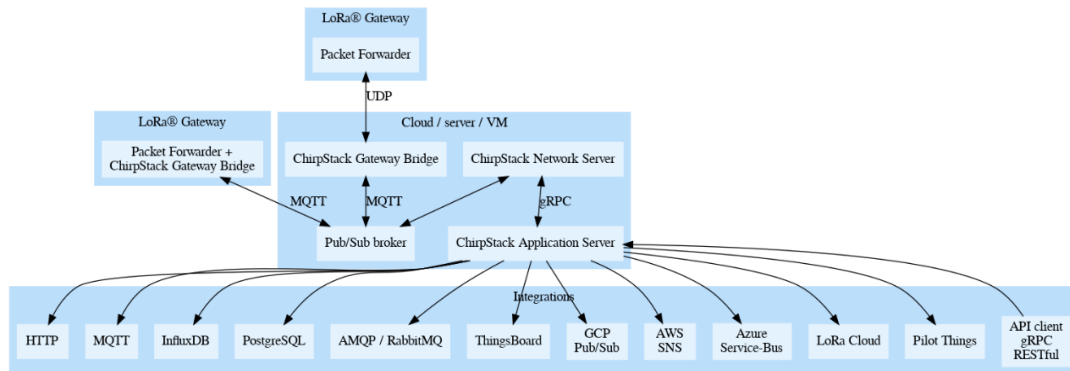
首先，LoRa 終端裝置(Device)將資料(Message)傳給網關(Gateway)後，Gateway 會將 Message 上傳(Uplink)給 LoRa 雲端伺服器(Server)。當 Server 確實接收到 Uplink 的動作時，會立即回傳一個回覆(Ack)給 Gateway，並由 Gateway 將 ACK 傳給 Device。而在此同時，應用程式伺服器(Application Server)可從 Server 提取剛剛 Uplink 的數據(Data)，並分析 Data 的資料後，由使用者決定是否要下傳一筆資料(Command)給 Device。

下傳資料：

當有下傳 Command 時，此 Command 會傳給 Server 並等待傳給 Device。由於此研究我們使用 Class A 的機制，及事件觸發式。此 Command 將暫存在 Server 的佇列中，直到下一次的 Uplink 發生時才會觸發下傳動作(Downlink)，並將 Command 傳給 Device。值得注意的

是，此時 Ack 仍會跟著經由 Gateway 傳給 Device。

4.取得 LoRaWan Server 資料的方式



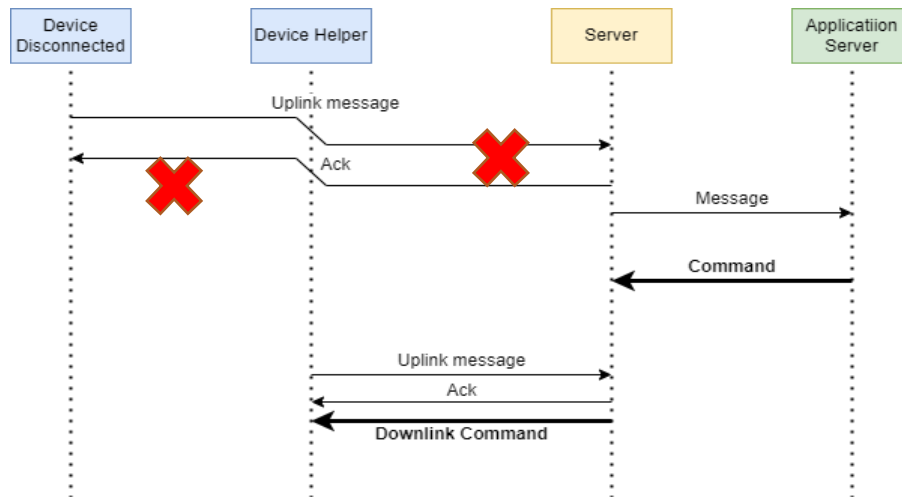
LoRaWan 的 Server(ChipStack Network Server)可與許多應用程式作對接，包括使用 gRPC 與 LoRaWan 本身的應用程式 ChipStack Application Server，此包含網頁介面和 APIs 方便操作。其也可使用 MQTT 訂閱其終端裝置上傳的資料和發布訊息到終端裝置。

本次研究使用 MQTT 的方式訂閱 LoRa 終端裝置上傳到 Server 的訊息並解析其內容，並使用 ChipStack Network Server 的 api 下傳內容給 LoRa 終端裝置。

三、 結合兩種傳輸方式的轉換機制

此次研究將以兩個 LoRa 終端裝置作為實驗對象，其中 Seeeduino1(Device 1)將作為正常終端裝置，Seeeduino2(Device 2)將作為中途斷線終端裝置。

1. 判斷 LoRa 裝置失去連線



如上圖所示，已知每個 LoRa 終端裝置在成功上傳訊息給 Server 時，將同時收到一個 Server 回傳的 Ack。而當訊息上傳給 Server 的過程失敗時，LoRa 終端裝置也將無法收到預期的 Ack。

此次實驗將判斷 LoRa 終端裝置失去連線分為兩端：

(1) LoRa 終端裝置：

LoRa 終端裝置(Device 2)與 Server 失去連線時，無法自行得知訊息是否成功上傳，故以 Server 回傳的 Ack 作為判斷依據。當連續沒收到的 Ack 到達一定的數目時，Device 2 會判斷自己與 Server 失去連線，並轉換模式。

(2) Server 端：

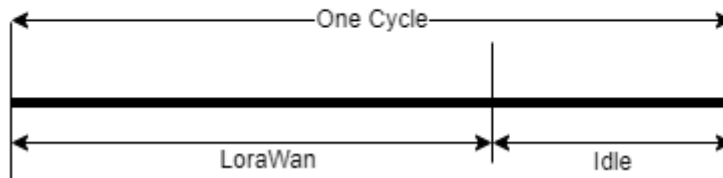
Server 端將依據是否收到來自 LoRa 終端裝置(Device 2)的訊息判斷 Device 2 是否與 Server 端失去連線。當連續沒收到的訊息到達一定的數目時，Server 端將判斷 Device 2 與 Server 失去連線，並從 Application Server 下傳一道命令訊息給另一臺 LoRa 終端裝置 (Device 1)，使其轉換模式。

為了保留資料的完整性，與 Server 失去連線的 LoRa 終端裝置將保存失去連線時的傳送失敗的訊息，並逐一刪除超過時限(4 個週期)的資料，直到成功將訊息傳送出去。

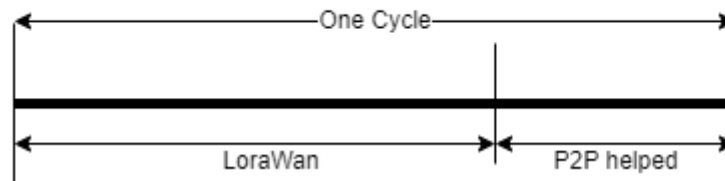
2. LoRa 終端裝置轉換模式

此實驗的轉換方案提出三種 LoRa 終端裝置的傳輸模式結合方式，分別為一般模式、斷線模式以及幫助模式。

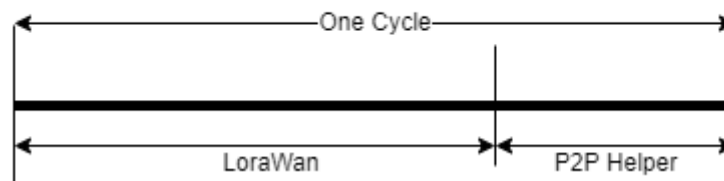
Normal Mode



Disconnected Mode



Helper Mode



(1) 一般模式：

此模式為終端裝置正常運行時的模式。

在一般模式中，裝置會在前半段使用 Join LoRaWan 的傳輸方式與 Server 連線並上傳資料。後半段則進入休息模式，等待下一輪周期開始。

(2) 斷線模式：

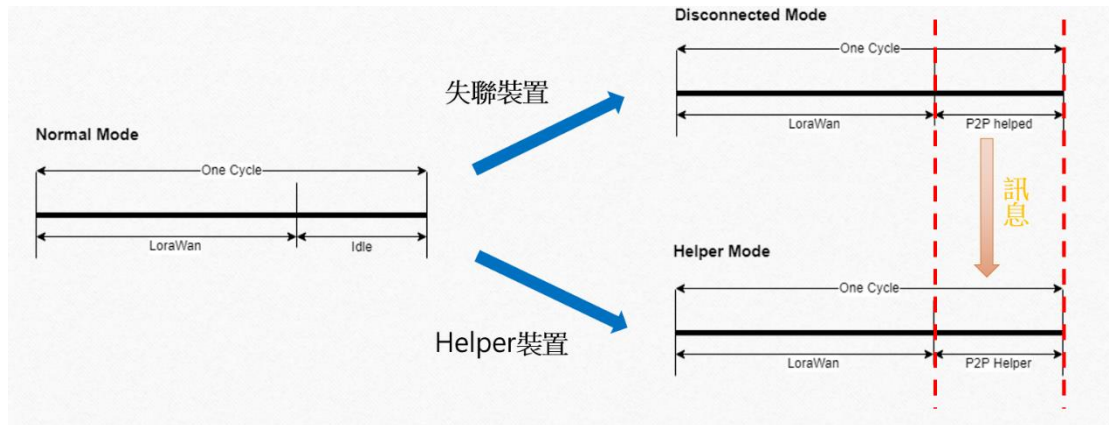
此模式為終端裝置與 Server 失去連線時的運行模式。當終端裝置判斷自己與 Server 失去連線時，會自行進入此模式尋找周圍的其他終端裝置做資料傳送。

此斷線模式中，前半段終端裝置會進入使用 Join LoRaWan 的傳輸方式嘗試重新與 Server 連線，後半段則使用 P2P 的傳輸方式，將資料傳給其他終端裝置

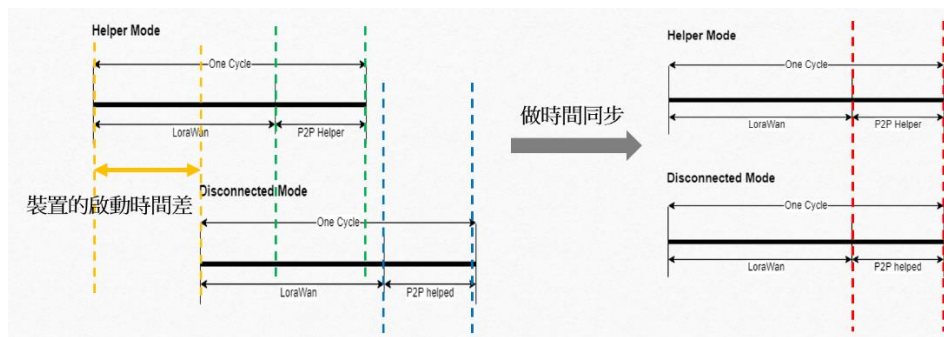
(3) 幫助模式：

此模式為終端裝置幫助其他失去連線的終端裝置時的運行模式。當終端裝置收到 Server 端的下傳的命令訊息時，會進入該運行模式接收其他裝置藉由 P2P 傳輸方式廣播的訊息。

此斷線模式中，前半段終端裝置會進入使用 Join LoRaWan 的傳輸方式與 Server 連線，並傳送自己以及幫助的終端裝置訊息，後半段則使用 P2P 的傳輸方式，接收其他終端裝置的訊息。



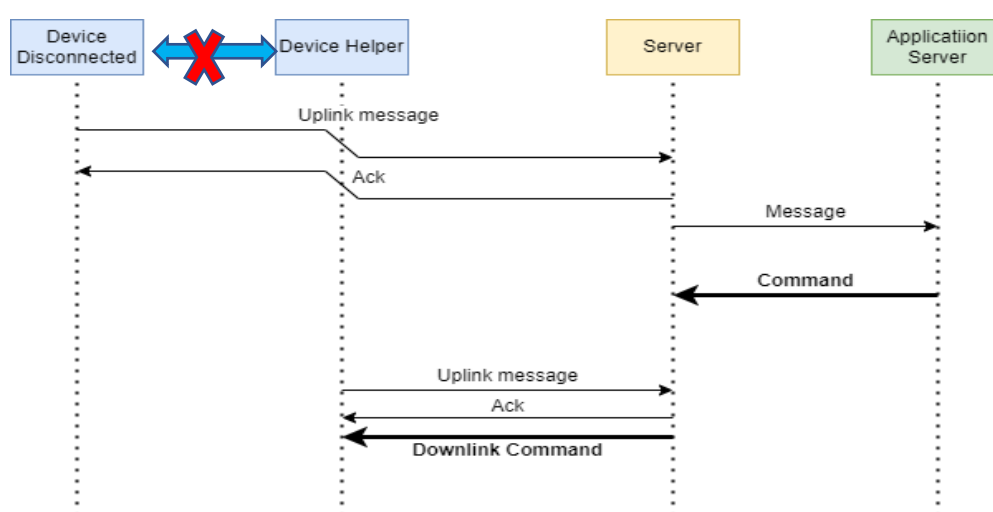
斷線模式與幫助模式之間的關係：



由於裝置啟動時間的差異，在相同週期的循環下，兩種模式之間的 P2P 傳輸方式時間有可能會完全錯開，導致正常連線的終端裝置無法接收斷線的終端裝置所傳送的訊息，以致無法協助傳遞資訊。

故在各自的終端裝置轉換模式後，須先進行時間同步的動作，才能減少資料傳遞失敗的機會。時間同步的詳細內容已在前面提到，這邊將不再贅述。

3. LoRa 終端裝置重新連線



當失去連線的終端裝置重新與 Server 連線時，將會有進行三個步驟回復正常運行模式。

(1) 重新連線的終端裝置(Device 2)：

終端裝置與 Server 連線並成功上傳訊息後，將再次收到 Server 回傳的 Ack。終端裝置將依此判斷自己已與 Server 成功連線，並自行從斷線模式轉換回一般模式，停止 P2P 的傳輸方式傳送訊息給其他終端裝置。

(2) Server 端：

當失去連線的終端裝置(Device 2)重新與 Server 連線後，Server 將再次收到 Device 2 上傳的訊息，並依此判斷終端裝置已重新連線。

Server 端將藉由 Application Server 下達一道命令訊息給另一臺正常連線的裝置(Device 1)，使其從幫助模式轉換回一般模式

(3) 幫助傳遞訊息的終端裝置(Device 1)：

幫助傳遞訊息的終端裝置(Device 1)將接收到 Server 端下傳的命令訊息，並從幫助模式轉換回一般模式，停止使用 P2P 的傳輸方式接收其他終端裝置的訊息。

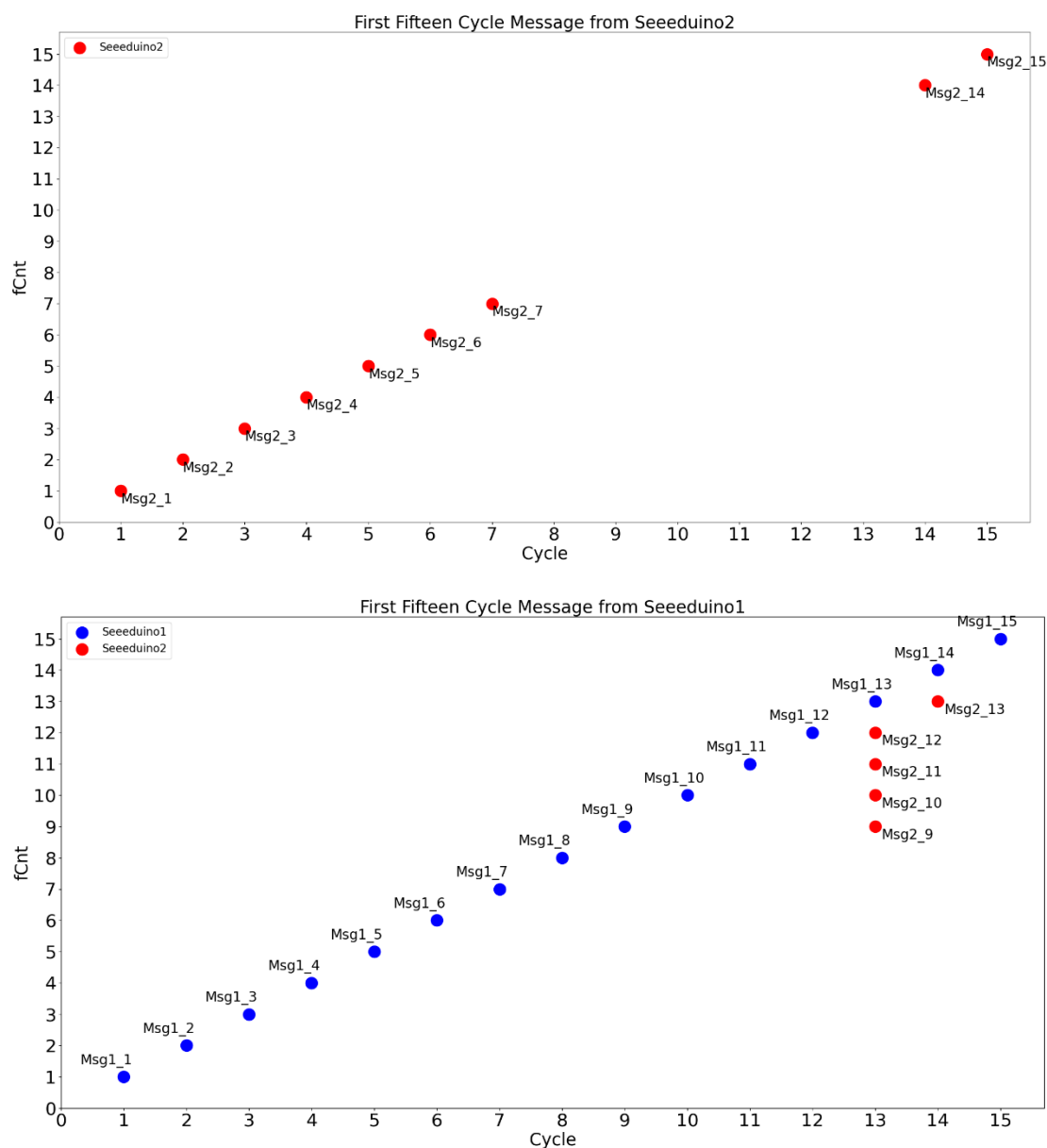
實作成果：

Cycle	Seeeduino1	Seeeduino2
Helper:	(Seeeduino2)	(Seeeduino1)
Cycle 1	1:msg1_1#	2:msg2_1#
Cycle 2	1:msg1_2#	2:msg2_2#
Cycle 3	1:msg1_3#	2:msg2_3#
Cycle 4	1:msg1_4#	2:msg2_4#
Cycle 5	1:msg1_5#	2:msg2_5#
Cycle 6	1:msg1_6#	2:msg2_6#
Cycle 7	1:msg1_7#	2:msg2_7#
Cycle 8	1:msg1_8#	No message
Cycle 9	1:msg1_9#	No message
Cycle 10	1:msg1_10#	No message ! #Find Helper...
Cycle 11	1:msg1_11#	No message ! #Find Helper...
Cycle 12	1:msg1_12#	No message ! #Find Helper...
Cycle 13	1:msg1_13#2:msg2_9,msg2_10,msg2_11,msg2_12	No message ! #Find Helper...
Cycle 14	1:msg1_14#2:msg2_13	2:msg2_14#

此實驗使用兩個 LoRa 終端裝置 Seeeduino1(Device 1)及 Seeeduino2(Device 2)，彼此互為對方的幫助模式對象。

此實驗設計在第 10 次週期時使 Device 2 與 LoRaWan Server 斷線。在 Device 2 連續 3 筆資料都上傳失敗後，Device 2 開始轉換成斷線模式並尋找周圍的其他終端裝置。而 Device 1 在收到 Server 端的命令訊息後轉換成幫助模式。兩部裝置在第 11 週期完成時間同步，並從第 12 週期開始，Device 1 開始接收 Device 2 斷線時未成功上傳的訊息，並在第 13 週期時成功上傳給 Server 端。

分析：



上面兩張圖分別代表 Server 端從 Seedeuino1 以及 Seedeuino2 所接收到的訊息與時間的關係圖。橫軸代表經過的週期時間，縱軸代表資料所屬的周期數。其中藍色圓點代表此訊息屬於 Seedeuino1 的資料，紅色圓點則是代表此訊息屬於 Seedeuino1 的資料。

由圖可看出，Seedeuino1 在 Seedeuino2 斷線後需要 5 次週期才成功幫助傳遞訊息，包括三次連續上傳失敗的偵測週期，以及一次時間

同步周期和傳遞訊息週期。並且每一次幫忙傳遞的訊息皆比正常時序晚一個週期。

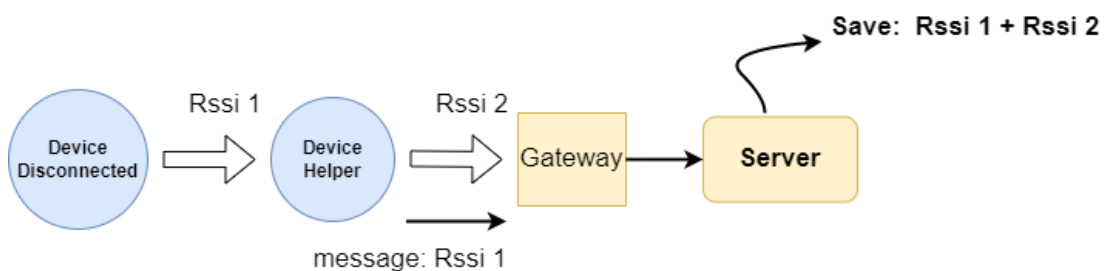
由於第一次傳遞成功時 Seeeduino2 的第八份資料已經超過時限(4週期)，在傳遞時 Seeeduino1 將只會收到第八份週期後的資料。

四、多部終端裝置的選擇

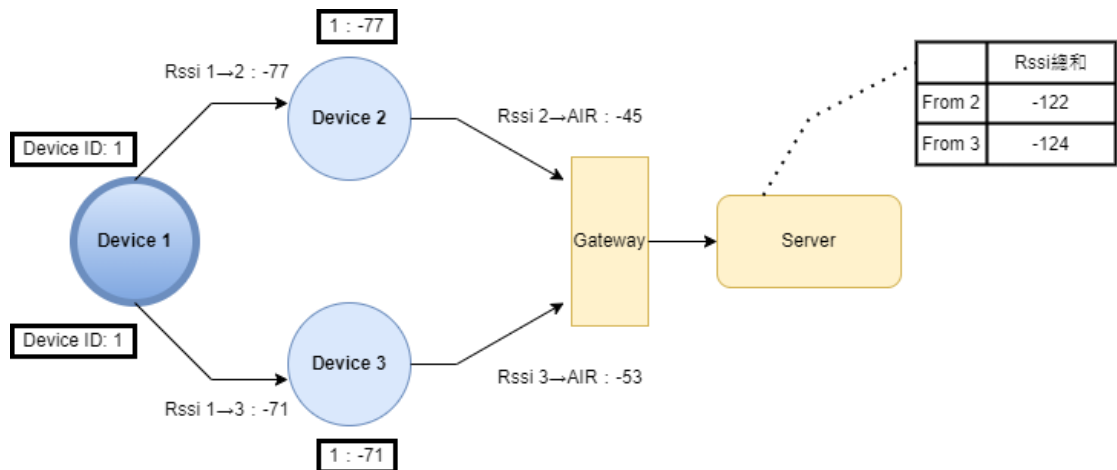
此為延伸上述只有兩部終端裝置的實驗情境，探討當有多部終端裝置時，斷線的終端裝置如何挑選最好的終端裝置作為幫忙傳遞訊息的對象。

此實驗提出以接收訊號強度 RSSI(Received Signal Strength Indicator)作為挑選傳遞對象的依據。RSSI 為接收無線信號的強度指示，可用來判定通訊質量。它和無線模塊的發送功率,射頻前端的設計以及天線的增益有關，單位是功率的單位，一般用 dBm 來表示。RSSI 的值越大，代表其通訊品質越好，表示方法為負值，以 0 為最好的通訊品質。

在 LoRa 的傳輸過程中有兩個通訊品質可用 RSSI 值量化。一個為終端裝置間的傳輸(P2P 傳輸方式)，另一個為終端裝置與網關(Gateway)之間的傳輸。



本實驗會以斷線的終端裝置(Device Disconnected)與幫忙傳遞訊息的終端裝置(Device Helper)間的 RSSI 1，加上 Device Helper 與網關(Gateway)間的 RSSI 2 來作為判斷叫好傳遞對象的依據。

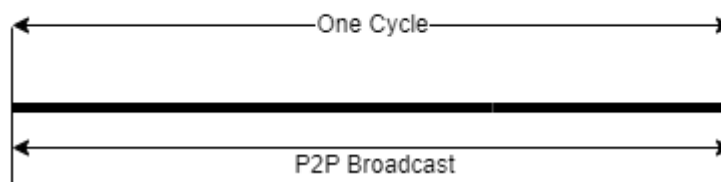


此實驗提出的設計為每個終端裝置輪流向周圍廣播訊息並且傳遞自己的裝置編號，藉由周圍的終端裝置接收訊息並紀錄裝置編號與 RSSI 值的關聯後，將此訊息傳送給 Server。最後當 Server 在接收這些訊息的同時，加上此終端裝置與網關的 RSSI 值並與其他終端裝置路徑所上傳的數值做比較後，選擇較大的總和即為該斷線終端裝置的最佳傳遞訊息對象。

以上圖為例，先由 Device 1 廣播自己的 ID(1)訊息給周圍的終端裝置。Device 2 及 Device 3 以幫助模式(Helper Mode)的 P2P 傳輸方式階段接收到訊息後記錄各自接收時的 RSSI 值(RSSI 1->2 及 RSSI 1->3)，並合併接收到的 ID(1:-77 及 1:-71)後傳給 Server。Server 端在接收訊息時會紀錄 Device 2 及 Device 3 傳送訊息給 Gateway 時的 RSSI 值(RSSI 2->AIR 及 RSSI 3->AIR)，並進行加總。在 Application Server 獲取這些數值後對兩種由 Device 1 傳遞過來的 RSSI 值總和進行排序後，選擇最大的值，其傳遞的終端裝置(Device 2)便是廣播的終端裝置(Device 1)最好的傳遞對象。其他終端裝置的挑選以此類推。

為了實現上述設計，我們加入新的終端裝置傳輸模式：廣播模式

Broadcast Mode



此模式將週期內將全部使用 P2P 模式發送自己的終端裝置編號。此模式設計原因為多部終端裝置的開機時間有所差異，可能導致每個終端裝置在週期內進入 P2P 傳輸方式的時間點不同。為了確保周為每個終端裝置都能接收到廣播訊息以提高挑選時的完整度，此模式將可保證周圍範圍內可接收到廣播訊息的終端裝置都能在各自的 P2P 傳輸方式時段內收到訊息。

但是此模式將會犧牲一次與 Server 連線的機會，必須將此次週期的資料訊息儲存後，在下次的週期才能一併傳給 Server 端。

實作成果：

成果

Select Helper

Cycle	Seeeduino1	Seeeduino2	Seeeduino3
Cycle 1	1:msg1_1#	2:msg2_1#	3:msg3_1#
Cycle 2	1:msg1_2#	2:msg2_2#	3:msg3_2#
Cycle 3	No message	2:msg2_3#1:-77	3:msg3_3#1:-71
Cycle 4	1:msg1_3,msg1_4#	No message	3:msg3_4#2:-70
Cycle 5	1:msg1_5#2:-77	2:msg2_4,msg2_5#	No message
Cycle 6	1:msg1_6#3:-72	2:msg2_6#3:-70	3:msg3_5,msg3_6#
Helper:	(Seeeduino2)	(Seeeduino3)	(Seeeduino2)
Cycle 7	1:msg1_7#	2:msg2_7#	3:msg3_7#
Cycle 8	1:msg1_8#	2:msg2_8#	No message
Cycle 9	1:msg1_9#	2:msg2_9#	No message
Cycle 10	1:msg1_10#	2:msg2_10#	No message #Find Helper...
Cycle 11	1:msg1_11#	2:msg2_11#	No message ! #Find Helper...
Cycle 12	1:msg1_12#	2:msg2_12#	No message ! #Find Helper...
Cycle 13	1:msg1_13#	2:msg2_13#3:msg3_9,msg3_10,msg3_11,msg3_12	No message ! #Find Helper...
Cycle 14	1:msg1_14#	2:msg2_14#3:msg3_13	3:msg3_14#
Cycle 15	1:msg1_15#	2:msg2_15#	3:msg3_15#

此實驗可看到從週期 3 開始進入挑選傳遞對象的流程，Seeeduino1, Seeeduino2, Seeeduino3 輪流進入廣播模式，並在該次週期沒有上傳資料給 Server，並在下次週期一併將資料上傳上來。於此同時其他兩個終端裝置也將接收到的廣播終端裝置編號與 P2P 之間的 RSSI 值上傳給 Server。

在挑選流程結束後篩選出各自的最佳傳遞對象，並且在週期 10 開始後使 Seeeduino3 與 LoRaWan 斷線。可看到 Seeeduino3 依照挑選結果將斷線後的訊息傳遞給 Seeeduino2。

挑選流程的詳細數值與排序：

```
*****
All device finish collecting rssi!

Rssi map:
  None   (-77, -45) (-71, -53)
(-77, -67)   None   (-70, -55)
(-72, -65) (-70, -44)   None

Start sorting...
Sseeduino1's pre-help_list: [(2, -122), (3, -124)]
Sseeduino2's pre-help_list: [(3, -125), (1, -144)]
Sseeduino3's pre-help_list: [(1, -137), (2, -114)]

Sseeduino1's final helper_list: [(2, -122), (3, -124)]
Sseeduino2's final helper_list: [(3, -125), (1, -144)]
Sseeduino3's final helper_list: [(2, -114), (1, -137)]

HELPER LIST ==>
  Device   Sseeduino1 Sseeduino2 Sseeduino3
Helper_1   Sseeduino2 Sseeduino3 Sseeduino2
Helper_2   Sseeduino3 Sseeduino1 Sseeduino1

*****
Finish selecting helper list process !!!
[New time: 175.7.1]
```

在 Rssi Map 中，縱軸為廣播終端裝置，橫軸為傳遞對象，排列依序為 Sseeduino1, Sseeduino2, Sseeduino3

文獻探討

Increasing LoRaWAN Application-Layer Message Delivery Success Rates

(Shie-Yuan Wang and Tzu-Yang Chen Department of Computer Science National Chiao Tung University, Taiwan)

Long-Term Performance Studies of a LoRaWAN-based PM2.5 Application on Campus

(Shie-Yuan Wang, Ji-Jhe Zou, Yo-Ru Chen, Chun-Chia Hsu, Yu-Hsiang Cheng, and Chia-Hung Chang Department of Computer Science National Chiao Tung University, Taiwan)

在精密時間協定下應用於無線感測網路之分散式量測系統研究成果報告

(楊中平,張保榮, 國立成功大學資訊工程學系 (所))