

在 BSN 網路上建立與重組可靠的反向路由樹的機制

蕭智仁 許蒼嶺
 電腦通訊網路實驗室
 國立中山大學電機工程學系
 sheu@ee.nsysu.edu.tw

摘要

本論文在人體感測網路(Body Sensor Networks, BSN)上建立與重組可靠的反向路由樹(Reversed Routing Tree, RRT)的機制，此一反向路由樹有兩個優點，一為使用 multi-hop 的樹狀架構去取代傳統的 one-hop 星狀架構以減少封包的碰撞機率，二為在選擇路徑時考慮到連線品質(Link Quality Indicator, LQI)可以讓封包遺失的比率(Packet Loss Ratio, PLR)降低。另外，我們建立的 RRT 可以針對人體姿勢的突然改變進行重組，使得移動位置的 sensor node 得以重新加入新的父節點，以顯著的減少 PLR。

關鍵詞：LQI 、BSN 、PLR 、碰撞、反向路由樹。

Abstract

This paper constructs a reconfigurable and reliable reversed-routing tree (RRT). There are two purposes of building the RRT: (i) to avoid packet collisions, we use multi-hop tree structure to replace traditional single-hop star structure; (ii) to reduce packet loss ratio (PLR), the RRT selects the routes based on the link quality indicator (LQI). Additionally, once there is a sudden human movement, the RRT can reconfigure itself such that a moving sensor node will re-connect to another parent node, which reduces PLR significantly.

Keywords: LQI 、BSN 、PLR 、collision 、RRT.

1. 前言

近年來人體感測網路(Body Sensor Network, BSN)已經開始蓬勃發展，而 BSN 網路的應用遍及了運動員的訓練、病人的看護、老年人的照料等等，因此藉由在人體上裝置 sensors 來收集各種資訊，例如心電圖(ECG)、肌電圖(EEG)、血壓等，由於這些資料都是與老人、病人的身體健康息息相關，所以降低 sensor nodes 收集資料的出錯機率也成為了一個重要課題。

2. 人體感測網路

2.1 IEEE 802.15.6

IEEE 802.15.6 [1]於 2007 年時便被提出討論，

直到 2012 年才終於制定這項通訊標準，此標準是針對於人體周邊及人體內部的通訊協定，它的應用包括醫療、個人娛樂、消費電子等，而 IEEE 802.15.6 定義最遠為 3 公尺的工作範圍，最高可以達到 10 Mbps 的傳輸速率，它也提供 QoS 的功能。

2.2 BSN 拓樸

BSN 網路的拓樸約有 3 種：星狀(star)、網狀(mesh)、樹狀(tree)，而本論文針對樹狀拓樸來做研究，其中樹狀拓樸可分為兩種：正向路由樹(Forwarded Routing Tree, FRT)與反向路由樹(Reversed Routing Tree, RRT)，在樹狀拓樸中一般是由 root 開始往其他節點一圈一圈的建立一棵樹，每一圈在樹狀拓樸中稱為深度(depth)，正向路由樹是資料流傳送的方向與建立樹的方向相同；而反向路由樹是資料流傳送的方向與建立樹的方向相反，因為在 BSN 網路中有個特性，就是 sensors 收集資料後會將這些資料傳送給 hub，所以必須採用反向路由樹，在 BSN 中我們考慮資料傳送路徑的連線品質，其目的是傳送資料時，能減少封包的遺失或是錯誤，因此建立反向路由樹時，我們必須收集周圍 sensors 的連線品質，然後依據連線品質建立反向路由樹，所以會增加建樹的 overhead 與 complexity。

2.3 相關研究

BSN 建立樹狀拓樸的研究主要都集中在如何建立路由協定，在建立路由協定時必須考慮網路的可靠度、減少封包的碰撞、省電、身體姿勢的改變等，與網路可靠度相關的協定，例如 H. Chebbo 等人[2]提出 Restricted Tree Topology (RTT)，RTT 修改了 MAC frame 格式，在 frame control 加入 relaying flag 表示拓樸圖為星狀或樹狀，建立樹狀拓樸是根據 node 連線期間 ACK 訊息遺失的情況選擇出 relaying node。在減少碰撞相關的協定有如 R. Annur 等人[3]提出應用樹演算法，此演算法將 BSN 網路的 sensor nodes 依據優先權先建立一棵選擇樹，sensor nodes 在樹中的位置都代表著競爭通道的優先權，而在樹中同一深度的 sensor nodes 其優先權高低是由左往右遞減，因此在最左邊的 sensor nodes 可以優先傳送資料，作者利用此優先權的方法減少碰撞的發生。

最後在 BSN 網路上，因為身體姿勢改變而提出的通訊協定有 2 篇，第一篇是 M. Quwaider 等人

[4] 提出的 On-body Store and Flood Routing (OBSFR)，它會在 sensor node 傳資料時，檢查目前的身體姿勢下有哪些鄰近 sensor nodes，並且選擇一個離 sink 最近的 sensor node，若沒有便先將資料暫存於 sensor node，當身體姿勢改變後，sensor node 有找到距離 sink 較近的 sensor node 時再傳送資料，最後作者使用嵌入式系統做實驗，從實驗結果發現，OBSFR 與現有的 probabilistic 路由協定比較，OBSFR 有比較小的 routing delay；第二篇是由 C. H. Wu 等人[5]所提出，他們的方法是去建立一個最小成本的路由樹，此方法是讓一個 sensor node 去 overhear 鄰近 sensor nodes 的資料傳輸，接著此 sensor node 會計算自己收集的資料與鄰近 sensor nodes 收集的資料的相似度，並將資料的相似度當作建立路由樹的成本，而資料的相似度越高則代表成本越小，然後為了傳輸每個 sensor node 所收集的資料，作者使用 Kruskal algorithm 建立一棵最小成本的路由樹。

3. 可靠的反向路由樹的建立與重組

在本章節中我們提出一個可靠的反向路由樹 (reversed-routing tree)協定，因為我們建立樹的方向與資料傳送的方向相反，所以會增加建立樹的複雜度，此反向路由樹協定以 Link quality indicator (LQI) 為基礎建立一棵樹，並且樹中的父節點會周期性地檢查自己與子節點的 LQI，然後適時地動態重組反向路由樹的結構，其目的是為了在 Body Sensor Network (BSN) 網路傳送資料時，能夠減少封包的遺失以及降低封包的錯誤率。在本章節中我們會先介紹反向路由樹的系統架構，然後介紹我們如何設計反向路由樹協定。

3.1 反向路由樹系統架構

圖 1 BSN 的架構

如圖 1 所示，BSN 網路是由一個 hub、數個 sensors、外部的 sensor 以及個人電腦所組成，而 BSN 網路是以 IEEE 802.15.6 為通訊協定，如果醫生想要收集病患的健康資訊，例如 ECG、EEG、體溫、血壓...等，醫生會藉由個人電腦發出蒐集身體資訊的要求，當 hub 收到這個要求後，它會通知人體上的 sensors 開始收集此資料，並且 sensors 會將收集到

的資料傳回給 hub，最後 hub 再將這些資料傳送給醫生的個人電腦，如此一來醫生便可以由這些資料判斷病患的身體情況。

3.2 反向路由樹協定

於 BSN 網路中，sensors 收集資料後，它們都是朝 hub 傳送，所以我們希望傳送的路徑都有好的 LQI，以確保封包不會遺失或出錯。我們建立一棵以 hub 為 root 的樹，然後樹中的父節點會周期性檢查子節點的 LQI，如果當子節點的 LQI 小於預設的 LQI threshold (LQI_TH) 時，父節點會通知子節點要另尋新的父節點，如此可以動態重組反向路由樹 (RRT)，以確保傳送路徑的連線品質。

於 IEEE 802.15.6 中我們新增了 8 個反向路由樹協定的控制封包，如圖 2，我們在 frame control 欄位中的 frame type 新增一個 RRT 的類別，然後在 RRT 類別下，我們在 frame subtype 欄位新加入 8 個 RRT 的控制封包，分別是：Build_Tree_Request (BTREQ)、Build_Tree_Reply (BTREP)、Parent_Selection (PS)、Member_Notification (MN)、Periodical_LQI_Measurement (PLM)、Bad_LQI (BL)、Joint_Request (JREQ)、Joint_Reply (JREP)，而它們對應的 frame subtype value 為 0000 到 0111，其中 MN 與 JREP 的 frame payload 中會紀錄 8-bit ΔLQI。

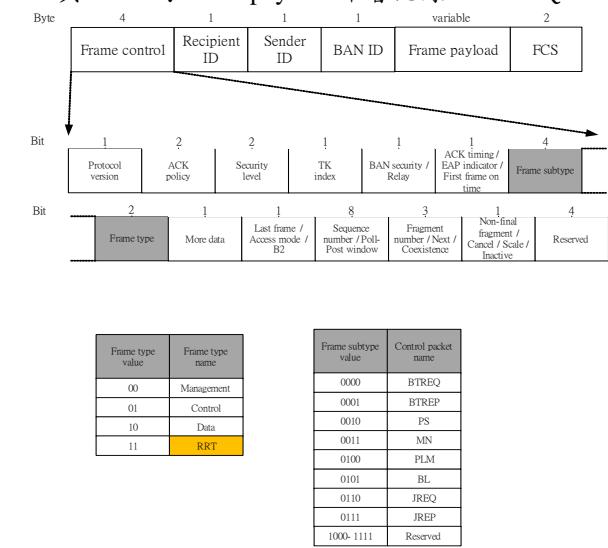


圖 2 MAC frame 中新增的控制封包

3.3 建立反向路由樹演算法

圖 3 為建立反向路由樹的演算法，首先 hub 會廣播 BTREQ 封包給鄰近 sensors，當鄰近 sensors 收到 BTREQ 時，如果它們已經是反向路由樹的節點，那那麼就會將封包丟棄，但如果它們不是反向路由樹的節點，則會回應 BTREP 封包。

如圖 4，當 S_i 收到 S_j 發送的 BTREP 封包時， S_i 會量測 LQI 值，並且判斷此 LQI 值是否大於 LQI_TH，如果大於，則 S_i 會送出 MN 封包給 S_j 。接著如果 S_i 收到 MN 封包時，若 MN 是由 hub 發送的，則 S_i 會將

hub 設為父節點，然後送回 PS 封包給 hub，接著廣播 BTREQ 封包，最後它會周期性地傳送 PLM 封包給 hub；若當 S_i 收到非 hub 傳送的 MN 封包， S_i 會在第一次收到 MN 封包時，紀錄下 MN 封包中的 ΔLQI ，並暫時將 MN 封包的發送端設為父節點，如果 S_i 非第一次收到 MN 封包，它會比較前一次的 ΔLQI 是否小於該封包的 ΔLQI ，若小於，那麼 S_i 便會更新父節點，但是如果前一次的 ΔLQI 與該封包的 ΔLQI 相等時， S_i 便會從具有相同 ΔLQI 的 sensors 中任意挑選一個 sensor 作為父節點，當確定了父節點後， S_i 會送出 PS 封包通知父節點，接著 S_i 會周期性地傳送 PLM 封包給父節點，最後 S_i 繼續廣播 BTREQ 封包尋找下一個深度的子節點。

```

//At first, the hub broadcasts BTREQ.
If  $S_i$  receives BTREQ from  $S_j$ 
    If  $S_i$  is a tree member, it drops BTREQ.
    Else  $S_i$  responses BTREP.

If  $S_i$  receives BTREP from  $S_j$ , measures LQI.
    If  $S_j.LQI > LQI\_TH$ 
         $S_i$  sends MN to  $S_j$ 
    Else return

If  $S_i$  receives MN
    If MN is from the hub
         $S_i$  set the hub as parent
        Send PS to the coordinator
        Broadcast BTREQ
        Send PLM message to the hub
        periodically
        Return
    Else // MN is not from the hub
        If the first reception of MN
            PREV_ $\Delta LQI$  = MN_ $\Delta LQI$ 
            Set MN_SOURCE as parent
        Else
            If MN_ $\Delta LQI$  > PREV_ $\Delta LQI$ 
                PREV_ $\Delta LQI$  = MN_ $\Delta LQI$ 
                Set MN_SOURCE as parent
            If MN_ $\Delta LQI$  == PREV_ $\Delta LQI$ 
                Randomly choose a sensor as
                parent
            //Parent has been chosen
            Send PS to  $S_i$ _parent
            Send PLM to  $S_i$ _parent periodically
        Broadcast BTREQ
    
```

圖 3 建立反向路由樹演算法

3.4 動態重組反向路由樹的演算法

圖 4 是動態重組反向路由樹的演算法，在演算法中，我們分成兩種情況來處理，分別為 sensor 是反向路由樹的成員以及非反向路由樹的成員，當 S_i 為反向路由樹的成員且為父節點時， S_i 會收到由它的子節點 S_j 發送 PLM 封包，並且 S_i 會量測 LQI 值，

如果此 LQI 值小於 LQI_TH ，那麼 S_i 會送出 BL 封包通知子節點 S_j ，當子節點 S_j 收到 BL 封包後，它會廣播 JREQ 封包，並且進入重組反向路由樹 (RECONSTRUCTION) 的步驟；另一個情況是 S_i 為非反向路由樹的成員，當 S_i 已經收集好資料欲傳送時，它卻發現自己沒有父節點，於是 S_i 會廣播 JREQ 封包，然後進入重組反向路由樹的步驟。

最後介紹如何重組反向路由樹，當 S_i 接收到來自 S_j 的 JREQ 封包，而 S_i 會回覆 JREP 給 S_j ，當 S_j 第一次接收到 JREP 封包時，它會記錄下 JREP 封包中 ΔLQI 欄位的數值，並且 S_j 會先將 S_i 選為父節點，如果 S_j 非初次收到 JREP 封包時，它會比較前次記錄的 ΔLQI 與目前 JREP 封包的 ΔLQI ，如果目前的 JREP 封包的 ΔLQI 比較大時， S_j 會記錄下目前 JREP 封包的 ΔLQI ，然後更新父節點為 S_i ，但是如果前次記錄的 ΔLQI 與目前 JREP 封包的 ΔLQI 相等時， S_j 便會隨機挑選具有相同 ΔLQI 值的 sensors 為父節點，最後確定了新的父節點後， S_j 就會送出 PS 封包給新的父節點，其用意是跟新的父節點講 S_j 為它的子節點，然後 S_j 會開始周期性地傳送 PLM 封包給新的父節點。

Tree member:

```

//  $S_i$  is  $S_j$ 's parent.
If  $S_i$  receives PLM message from  $S_j$ 
    If  $S_j.LQI < LQI\_TH$ 
         $S_i$  sends BL message to  $S_j$ 
    If  $S_j$  receives BL message
        Broadcast JREQ //Find a new parent.
        Go to RECONSTRUCTION()
    
```

Non Tree member:

```

//When  $S_i$  collects data and is ready to
transmit.
 $S_i$  found itself has no parent
//After the construction of
//reversed-routing tree, some sensors not
//in the tree.
Broadcast JREQ //Find a parent
Go to RECONSTRUCTION()
    
```

RECONSTRUCTION:

```

If  $S_i$  receives JREQ from  $S_j$ 
     $S_i$  responses JREP to  $S_j$ 
If  $S_j$  receives JREP from  $S_i$ 
    If the first reception of JREP
        PREV_ $\Delta LQI$  = JREP_ $\Delta LQI$ 
        Set  $S_i$  as parent
    Else
        If PREV_ $\Delta LQI$  < JREP_ $\Delta LQI$ 
            PREV_ $\Delta LQI$  = JREP_ $\Delta LQI$ 
            Set  $S_i$  as parent
        If PREV_ $\Delta LQI$  == JREP_ $\Delta LQI$ 
            Randomly choose a sensor as parent
        Send PS //Send PS to the new
        //parent.
        Send PLM periodically
    
```

圖 4 動態重組反向路由樹演算法

4. 模擬與分析

為了要分析使用反向路由樹建立 BSN 網路的效能，我們使用網路模擬器 NS-2，因為 NS-2 目前並沒有提供 IEEE 802.15.6 的功能，所以我們選擇 NS-2 有提供的 IEEE 802.15.4 當作 sensor nodes 之間的通訊協定。在本章節中，我們會先介紹 NS-2 的模擬環境，然後對模擬的結果進行分析討論。

4.1 模擬環境

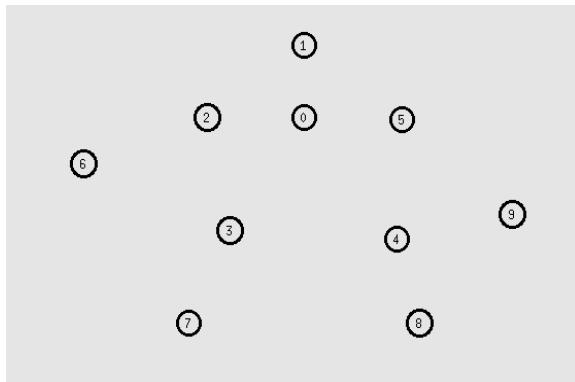


圖 5 使用 NS-2 建立的反向路由樹

圖 5 為我們使用 NS-2 所建立的人體感測網路 (BSN)，我們建立的 BSN 有 10 個 sensor nodes，其中編號 0 的 sensor node 為 hub，其他的 sensor nodes 會以不同的速率(從 0.8 Kbps 增加到 4.8 Kbps)收集資料，sensor nodes 將收集的資料依照反向路由樹 (RRT)的拓樸傳送給 hub。在我們建立的 RRT 拓樸中，sensor node ID=1, 2, 3, 4, 5 的只需要 1 個 hop 就可以送到 hub，然而 sensor node ID=6, 7, 8, 9 的卻需要 2 個 hops 才可以傳送至 hub。

為了簡單起見，我們在模擬 BSN 中人體各部位可能會移動的情境，只考慮到手的抬高，例如圖 6 中的 sensor node ID=9 會從 sensor node ID=5 的下方位置移動到 sensor node ID=5 的上方位置。表 1 為我們在 NS-2 中所設定的模擬參數。

4.2 模擬程式的架構

圖 6 是 NS-2 的物件架構，白色區塊是 NS-2 原來就有的模組，灰色區塊 RRT 則是我們新加入 NS-2 的模組。TclObject 是用來連結 C++ 與 Tcl 的程式；NsObject 繼承 TclObject，它為所有基本網路組件的父類別，而繼承 NsObject 的子類別能做參數傳遞給 Scheduler 來排程處理；Biconnector 類別具有承上啟下的作用，可以連結網路層與 MAC 層，它擁有連結網路層和 MAC 層的指標：uptarget 與 downtarget，藉由這兩個指標可以快速的找到網路層和 MAC 層的物件；MAC 類別繼承 Biconnector 類別，MAC 類別中描述基本的 MAC 功能，然後再衍生出不同的 MAC 協定，例如圖 6 中的 Mac802_15_4；另外 PHY 類別也是繼承 Biconnector

類別，在 PHY 類別中會模擬比較簡單的實體層，然後它會衍生 wirelessPhy 類別，在 wirelessPhy 類別中模擬無線的實體層功能，最後衍生出 Phy802_15_4。Connector 類別是從 NsObject 類別所衍生出來，它的作用是連接不同的 NsObjects 物件；Agent 類別為 Connector 類別的子類別，它代表了網路層的起點與終點，並用於實現各層級的協定，例如 TCP、UDP、AODV 等；最後 RRT 的模組是我們繼承 Agent 類別所新增的反向路由樹協定。

表 1 模擬參數的設定

參數	設定值
Sensor node	10
Packet size (Bytes)	50
Number of packets	1000
Bit rate (Kbps)	0.8 - 4.8
Moving speed of human body (m/s)	0.1 - 0.5

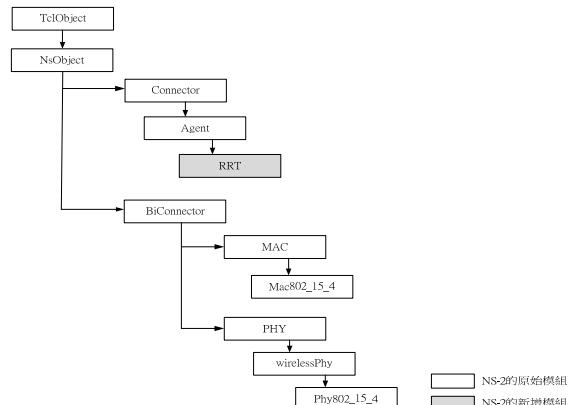


圖 6 NS-2 的架構

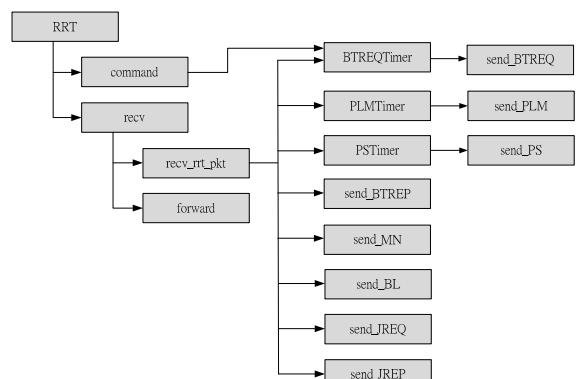


圖 7 RRT 的函式架構

新增的 RRT 模組其最大的貢獻是將原來 802.15.4 的 1 個 hop 的 STAR 拓樸擴充成 2 個 hops 的 TREE 拓樸，在 NS-2 的 802.15.4 環境上，從 1 個 hop 增加成 2 個 hops 的最大困難點在於建立封包轉傳的機制，為了詳細描述此一困難點我們以圖 7 的函式架構來說明，在 RRT 執行的時候會啟動 TCL_command 與 recv 的執行，TCL_command 函式

的目的是讓我們建立的 RRT 與 NS-2 的 TCL 設定建立連結，並在模擬程式開始時，啟動 BTREQTimer，然後 timer 倒數結束後，hub 會呼叫 send_BTREQ 函式，開始建立以 hub 為 root 的 RRT 拓樸，另外 recv 函式負責接收傳輸層和 MAC 層的封包，它會先判斷收到封包的類型，如果收到的是 RRT 的控制封包，就會呼叫 recv_rrt_pkt 函式處理，若是非 RRT 的控制封包則呼叫 forward 函式。當 sensor node 呼叫了 recv_rrt_pkt 函式後，此函式會先判斷目前收到的封包是哪一種 RRT 的控制封包，如果收到 BTREQ 封包，則我們會判斷此 sensor node 是否已在 RRT 拓樸中，若是，我們會將收到的 BTREQ 封包丟棄，若否，我們會呼叫 send_BTREP 函式發送 BTREP 封包；如果 recv_rrt_pkt 函式收到 BTREP 封包，我們會檢查 LQI 是否大於 LQI_TH，若大於，我們會呼叫 send_MN 函式；如果 recv_rrt_pkt 函式收到 MN 封包時，若此 sensor node 已經在 RRT 拓樸中，那麼就會丟棄 MN 封包，若此 sensor node 不在 RRT 拓樸中，又收到來自 hub 的 MN 封包時，會呼叫 send_PS 函式和 send_BTREQ 函式，若它收到來自非 hub 的 MN 封包，會啟動 PLMTimer 與 BTREQTimer，在 timer 倒數結束前，這個 sensor node 會挑選出具有最佳 LQI 的 parent，然後 timer 倒數結束後，PLMTimer 會呼叫 send_PLM 函式，並開始周期性的量測 LQI，另外 BTREQTimer 會呼叫 send_BTREQ 函式，再去尋找下一個深度的 sensor node；如果此 sensor node 收到 PLM 封包，它會檢查 LQI 是否小於 LQI_TH，若小於，則它會取消 PLM 封包的周期性傳送，並且呼叫 send_BL 函式，送出 BL 封包給它的 child；如果 recv_rrt_pkt 函式收到 BL 封包，我們會將此 sensor node 標記為不在 RRT 拓樸中，並且呼叫 send_JREQ 函式為此 sensor node 尋找新的 parent；若 recv_rrt_pkt 函式收到 JREQ 時，此 sensor node 會呼叫 send_JREP 函式，回覆 JREP 封包；當 recv_rrt_pkt 函式收到 JREP 時，此 sensor node 便會啟動 PSTimer 和 PLMTimer，在 timer 倒數期間，此 sensor node 會挑選具最大 LQI 的 sensor node 為新的 parent，若是有多個 sensor node 具最大 LQI，那麼便從中隨機挑選一個 sensor node，最後 timer 倒數結束後，此 sensor node 便會呼叫 send_PS 函式及 send_PLM 函式。最後在 forward 函式中，此 sensor node 會判斷是否已經抵達目的地，若是，它會將封包往傳輸層傳遞，若否，則它會依據我們建立的 RRT 拓樸轉傳封包給它的父節點。

4.3 模擬結果

圖 8 是我們固定 bit rate (BR)為 4.8 Kbps 的情況下，比較 STAR 與 RRT 在每個 sensor node 的 packet loss ratio (PLR)，其中 PLR 是我們先統計每個 sensor node 遺失的封包個數，然後將遺失的封包個數除以這個 sensor node 傳出的封包總數。從圖 8 中，我們發現在 STAR 時，sensor node ID = 6, 7, 8, 9

的 PLR 都比較高，因為這些 sensor nodes 距離 hub 比較遠，較差的 LQI 會導致較高的 PLR；另外 RRT 的 PLR 都比 STAR 的 PLR 低，這是因為在 RRT 的拓樸中所有 sensor nodes 在選擇路徑時都有考慮到 LQI，因此可以導致比較低的 PLR。

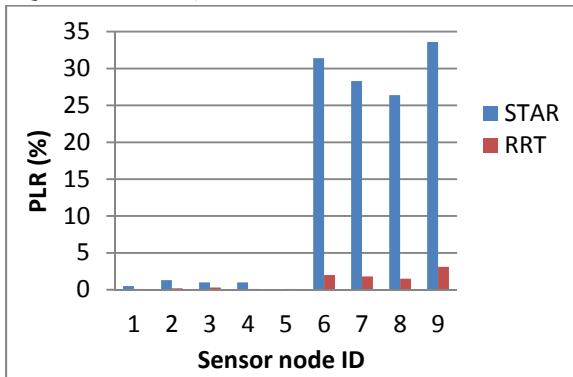


圖 8 個別的 PLR 在 STAR 和 RRT 比較

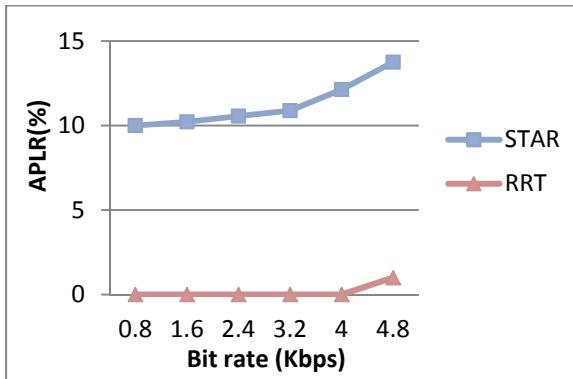


圖 9 所有的 APLR 在 STAR 與 RRT 的比較

圖 9 是我們將 BR 從 0.8 Kbps 增加到 4.8 Kbps 時，STAR 與 RRT 在 average packet loss ratio (APLR) 的比較，APLR 是將所有 sensor node 的 PLR 作平均。從圖 9 中我們發現 STAR 的 APLR 大約都在 10% 以上，而且 STAR 的 APLR 總是高過 RRT 的 APLR，這是因為在 STAR 中的每個 sensor node 都是與 hub 直接相連，所以距離 hub 較遠的 sensor nodes 就會因 LQI 較差而導致資料的出錯而被丟棄；此外隨著 data rate 的增加，封包碰撞變多而導致封包遺失，因此 STAR 的 APLR 會逐漸上升，而 RRT 的 APLR 只有在 BR 上升到 4 Kbps 時，封包碰撞的次數才會突然上升，進而導致 APLR 上升。

圖 10 為我們固定 BR 為 4.8 Kbps 的情況下，比較 STAR 與 RRT 的每個 sensor node 在 average packet delay (APD)的變化，各別 sensor node 的 APD 是我們統計每個 sensor node 傳送給 hub 的 1000 個封包的平均延遲。從圖 10 中我們發現每個 sensor node 在 STAR 的拓樸中其 APD 都高於在 RRT 拓樸中的 APD，這是因為從圖 8 中我們發現每個 sensor node 的 PLR 在 STAR 拓樸中都比較高，封包遺失需要重傳，所以封包成功抵達 hub 的時間會增加；而在 RRT 的拓樸中，sensor node ID=6, 7, 8, 9 的 APD 都比較高，這是因為它們在 RRT 的拓樸中都

需要經過 2 個 hops 才能將資料傳到 hub, 因此 APD 會比較高。

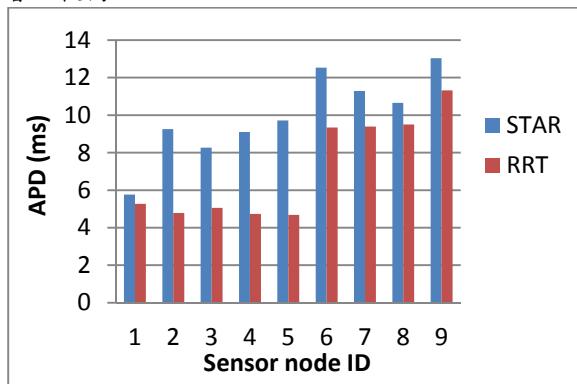


圖 10 個別的 APD 在 STAR 和 RRT 的比較

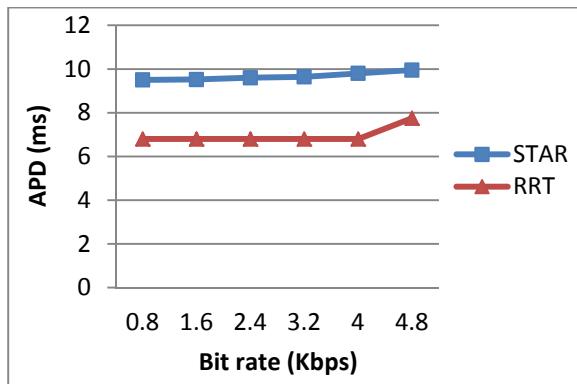


圖 11 所有的 APD 在 STAR 與 RRT 的比較

圖 11 是我們將 BR 從 0.8 Kbps 增加到 4.8 Kbps 時, STAR 與 RRT 在所有 sensor nodes 的 APD 比較, 所有 sensor nodes 的 APD 是我們先統計每個 sensor node 傳送給 hub 的 1000 個封包的平均延遲, 然後將每個 sensor node 的平均封包延遲再一起作平均。在圖 11 中我們發現在 STAR 拓樸中, 所有 sensor nodes 的 APD 都高過在 RRT 拓樸中的 APD, 因為在 STAR 拓樸中距離 hub 較遠的 sensor nodes 傳送的封包會因為 LQI 比較差而出錯, 此時 sensor nodes 會重新傳送封包, 所以使得 hub 成功接收到這些封包的時間變長; 另外隨著 sensor node 的 BR 增加, 封包產生碰撞的次數增加, 碰撞的封包必須重新競爭傳送的機會, 使得在 RRT 拓樸中的 APD 在 BR 上升到 4 Kbps 時會突然上升。

圖 12 是我們將人體移動速度從 0.1 m/s 增加到 1.1 m/s 時, RRT 重組時間(RT)的變化, 我們觀察到 0.1 m/s 至 0.5 m/s 時, 所有的曲線都呈現下降趨勢, 這是因為藉由週期性的發送 PLM 封包給 parent, parent 可以較快的知道 child 的 LQI 不好, 並通知 child 尋找新的 parent; 當我們固定 BR 改變 PLM interval (PI) 時, 我們發現 PI=0.1 s 時 RRT 重組時間比較短, 因為 PI 比較小時 parent 可以較快發現 child 的 LQI 變差, 進而較快啟動重組; 又當我們固定 PI 改變 BR 時, 發現 BR=4.8 Kbps 時 RRT 重組時間比較長, 因為 BR 較大的關係導致 PLM 封包會容易產生碰撞, 所以延長了 RRT 的重組時間。在圖 12

中, 我們觀察到在人體移動速度為 0.5 的時候, RRT 可以達到最短的重組時間。另外隨著人體移動速度繼續增加, 我們發現 0.5 m/s 之後, 重組的時間都開始變長, 並且 PI=1 s 的 RRT 比 PI=0.1 s 的 RRT 較快上升, 這是當 PI 較大時, child 移動太快, 使得 child 收不到來自 parent 訊息, 導致重組時間延長。

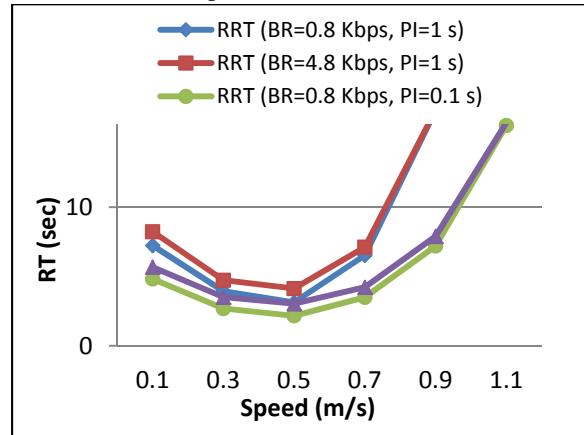


圖 12 RRT 的重組時間

5. 結論

我們在 BSN 網路上建立 RRT 拓樸, 使得每個 sensor node 的資料傳送路徑都具有良好的 LQI, 所以能夠有效減少 PLR, 又因為 PLR 降低, 減少封包重傳, 所以 RRT 的 APD 比 STAR 的 APD 來的小, 在人體動作改變時, 我們能夠局部的進行 RRT 拓樸的重組以減少封包的遺失。但是當人體移動速度太快時, RRT 拓樸的重組時間會延長甚至失敗。

致謝

作者要感謝 NHRI, Taiwan. This work is supported under the grant number NHRI-EX102-10142EI.

參考文獻

- [1] IEEE Standard for Local and Metropolitan Area Networks—Part 15.6: Wireless Body Area Networks, IEEE 802 LAN/MAN Standards Committee, Feb. 6, 2012.
- [2] H. Chebbo, S. Abedi, T.A. Lamahewa, D.B. Smith, D. Minutti, and L. Hanlen, “Reliable Body Area Networks Using Relays: Restricted Tree Topology,” 2012 International Conference on Computing, Networking and Communications (ICNC), Maui, Jan. 30, 2012.
- [3] R. Annur, N. Wattanamongkhhol, S. Nakpeelerayuth, and L. Wuttisittikul, “Applying the Tree Algorithm with Prioritization for Body Area Networks,” 2011 10th International Symposium on Autonomous Decentralized Systems (ISADS), Tokyo, Mar. 23-27, 2011.
- [4] M. Quwaider and S. Biswas, “On-body Packet Routing Algorithms for Body Sensor Networks,” 2009 First International Conference on Networks and Communications (NETCOM 2009), Chennai, Dec. 27-29, 2009.
- [5] C. H. Wu and Y. C. Tseng, “Data Compression by Temporal and Spatial Correlations in a Body-Area Sensor Network: A Case Study in Pilates Motion Recognition,” 2011 IEEE Transactions on Mobile Computing, vol. 10, no. 10, pp. 1459-1472, Oct. 2011.