

國立暨南國際大學資訊工程學系

碩士論文

應用於智慧電網之 ZigBee/TCPIP 轉換器備援協定
Fault-Tolerant ZigBee/TCPIP Translators in Smart Grid

指導教授：吳坤熹博士

研究生：陳麗雯

中華民國一〇〇年七月二十九日

國立暨南國際大學資訊工程學系

碩士論文

應用於智慧電網之 ZigBee/TCPIP 轉換器備援協定
Fault-Tolerant ZigBee/TCPIP Translators in Smart Grid

指導教授：吳坤熹博士

研究生：陳麗雯

中華民國一〇〇年七月二十九日

國立暨南國際大學碩士論文考試審定書

資訊工程學系

學系（研究所）

研究生 陳麗雯 所提之論文

應用於智慧電網之 ZigBee/TCPIP 轉換器備援協定

Fault-Tolerant ZigBee/TCPIP Translators in Smart Grid

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

溫志昂

委員兼召集人

王忍成

委員

吳坤熹

委員

中華民國 一〇〇 年 07 月 29 日

致謝

感謝我的指導教授吳坤熹老師，不僅讓我學到各方面的專業知識，更學會面對問題以及處理事情的方法，在我沒有自信的時候推著我走並且給我鼓勵，使得當初從沒想過的國外研討會報告，現在卻成了很棒的回憶。

感謝上天讓我進入了 409 實驗室，才能在研究遇到問題時有人一起討論、尋找解答，東西趕不出來時有人陪我熬夜；常常有人關心我的研究，總是有人記得幫我買飯甚至載我回家。無論是已經畢業的學長姊、一起畢業的好同學或即將畢業的學弟妹，研究的路上有你們真是太溫馨了！

感謝我的朋友們，在研究難產時聽我抱怨、替我打氣，尤其是共同在暨大努力的雁亭、小 B、勺菱、紫鈞、筱璇，煩躁的時候一起大笑消除壓力，需要鬥志的時候相互加油得到動力，你們是我在埔里生活的依靠。

最後，感謝我的家人，從小到大總是支持我做想做的事，有你們的辛苦付出，我才能無後顧之憂的完成碩士學位。

論文名稱：應用於智慧電網之 ZigBee/TCP/IP 轉換器備援協定

校院系：國立暨南國際大學科技學院資訊工程學系

頁數：28

畢業時間：一百年/七月

學位別：碩士

研究生：陳麗雯

指導教授：吳坤熹 博士

論文摘要

為了解決能源耗竭的問題，各國政府陸續推動智慧電網（Smart Grid）的基礎建設以有效利用能源，而智慧型電表基礎建設（Advanced Metering Infrastructure, AMI）則是實現智慧電網的第一步。它讓電力公司從電表收集到用電量資訊，再對電表下控制命令來做出因應的動作，以有效管理用電量。智慧型電表基礎建設的網路多半用到 ZigBee 和 TCP/IP 這兩種通訊協定，並且需要一個協定轉換器連接不同網路。因此，這個協定轉換器在系統中通常容易成為一個單點故障（Single Point of Failure）的所在。在本篇論文中，我們提出協定轉換器的冗餘方法來解決這個單點故障問題，使智慧電網的通訊更可靠。

關鍵詞：ZigBee、冗餘、智慧型電表基礎建設、智慧電網

Title of Thesis : Fault-Tolerant ZigBee/TCPIP Translators in Smart Grid

Name of Institute : Department of Computer Science and Information Engineering,

College of Science and Technology,

National Chi Nan University

Pages : 28

Graduation Time : 7/100

Degree Conferred : Master

Student Name : Li-Wen Chen

Advisor Name : Dr. Quincy Wu

Abstract

Smart Grid is an infrastructure promoted by governments in many countries for facilitating efficient management of energy in order to tackle the global warming and energy shortage problem. Advanced Metering Infrastructure (AMI) is an important part of Smart Grid to provide two-way communications for meters. This will allow the information and commands to be sent and received between meters and facility companies for energy management. AMI networks usually adopt ZigBee and TCP/IP as the transport protocols and a ZigBee/TCPIP translator must be installed between the two protocols. Therefore, the translator usually becomes a single point of failure in a communication system. In this thesis we propose redundant translators as a solution to this problem to make communication in Smart Grid more reliable.

Key words: AMI, Redundancy, Smart Grid, ZigBee

目錄

致謝	I
論文摘要	II
Abstract	III
目錄	IV
圖目錄	VI
表目錄	VII
1. 研究背景與動機.....	1
2. 與轉換器冗餘相關的研究.....	5
2.1. 無線感測網路中的冗餘及其應用	5
2.2. 虛擬路由器冗餘協定	6
2.3. 網路位址轉換的冗餘	7
2.4. 網路位址轉換冗餘的連線機制	9
3. 研究方法.....	11
3.1. 轉換器的共同位址：Multicast	11
3.2. 轉換器間的溝通：Heartbeat.....	15
4. 系統架構與實作.....	21
4.1. 系統架構	21
4.2. 封包遺失率實驗與結果	22
5. 結論.....	27

参考文献..... 28

圖目錄

圖 1	智慧電網概念圖 [1]	1
圖 2	智慧型電表基礎建設通訊	2
圖 3	ZeeBee 與 OSI 參考模型	2
圖 4	ZigBee 網路透過轉換器與 Internet 相連	3
圖 5	使用物理性冗餘的感測器[4]	5
圖 6	VRRP 工作原理	6
圖 7	VRRP 封包格式[5]	7
圖 8	Devarapalli 等人提出網路位址轉換的冗餘系統圖	8
圖 9	位於不同網域的 NAT 冗餘	9
圖 10	ZigBee Multicast 控制欄位	11
圖 11	ZigBee Multicast 封包傳遞	12
圖 12	ZigBee Multicast 封包接收端流程	13
圖 13	ZigBee Multicast 封包發送端流程	13
圖 14	ZigBee 節點送資料給伺服器	14
圖 15	主控轉換器送心跳給備援轉換器	15
圖 16	轉換器狀態切換流程圖	16
圖 17	實驗平台	21
圖 18	系統架構	22
圖 19	無備援機制，故障比率與故障時間對封包遺失率的影響	23
圖 20	備援機制下，故障比率與心跳間隔時間對封包遺失率的影響	24
圖 21	封包遺失的改善效能比率	25

表目錄

表 1	Timer 相關計算器.....	17
表 2	Heartbeat Timer 相關程式.....	18
表 3	Active Down Timer 相關程式.....	19
表 4	無備援機制，故障比率與故障時間對封包遺失率的影響.....	23
表 5	備援機制下，故障比率與心跳間隔時間對封包遺失率的影響.....	24
表 6	封包遺失的改善效能比率.....	25

1. 研究背景與動機

能源耗竭與全球暖化一直是近年來備受關注的重要議題，所以為了節約能源，出現了智慧電網（Smart Grid）[1]的概念如圖 1。智慧電網除了原有的電力傳送系統之外，更結合了通訊技術、智慧能源裝置、資訊科技及數位控制，因此能更有效率地管理能源，並且各國政府也開始陸續推動這項基礎建設，希望能藉此改善環境問題。

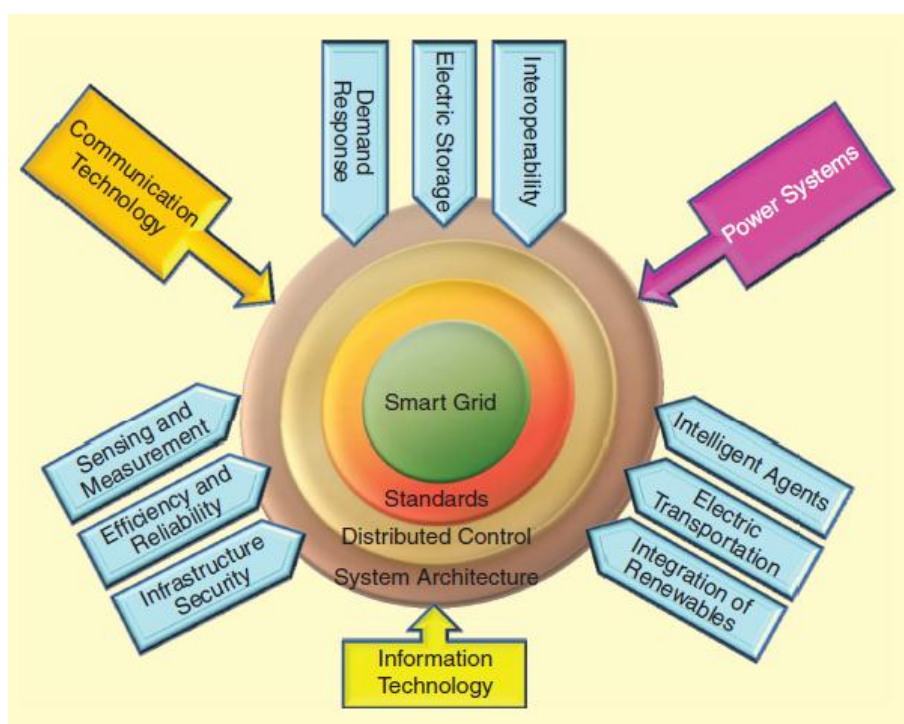


圖 1 智慧電網概念圖 [1]

要實現智慧電網，就必須先建立智慧型電表基礎建設（Advanced Metering Infrastructure，簡稱 AMI），才能讓智慧電網中各個子系統更有效率地溝通。智慧型電表基礎建設提供了雙向通訊，讓電力公司可以收集電表上的用電量資訊，再對電表下達控制命令，以控制用電量。如圖 2 所示，終端用戶和電力公司之間的通訊通常分成三個部分：(1) 電器用品到電表、(2) 電表到資料聚集點、(3) 資料聚集點到電力公司，依序分別是家庭區域網路（Home Area Network，簡稱 HAN）、社區區域網路（Neighborhood Area Network，簡稱 NAN）以及後置網路（Backhaul Network）。

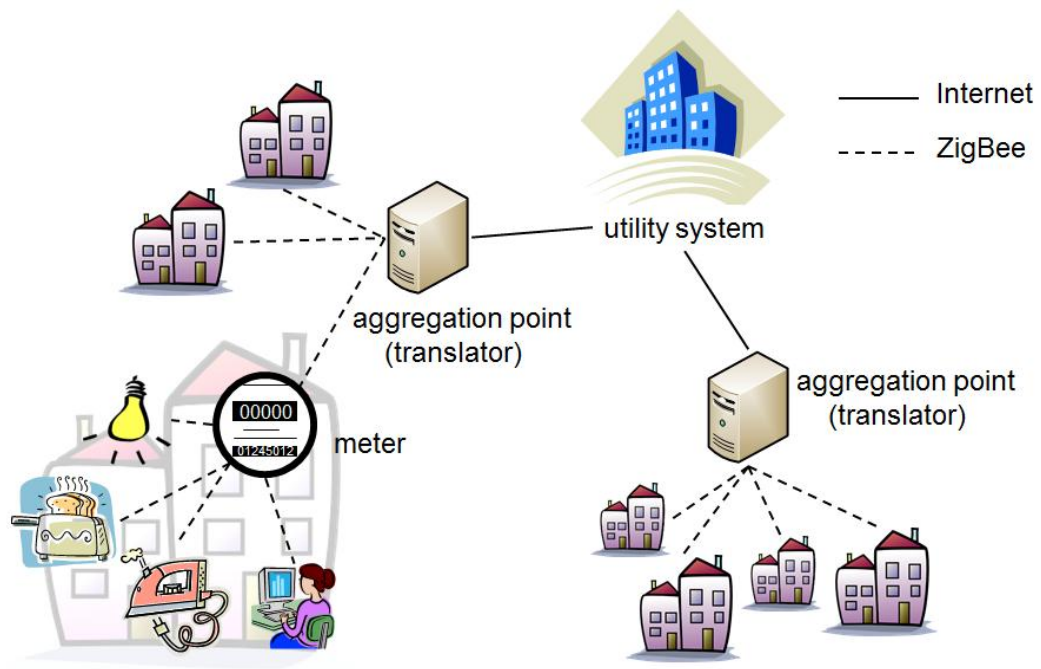


圖 2 智慧型電表基礎建設通訊

在各種通訊技術中，ZigBee[2]是目前正在崛起的無線感測網路（Wireless Sensor Network）協定，支援大量網路節點及多種網路拓撲，並且具有低成本、低功耗、低速率、可靠等特色，因此常被應用在家庭自動化、工業控制或個人醫療照護系統上，AMI 裡 HAN 和 NAN 的部分，也都使用 ZigBee。

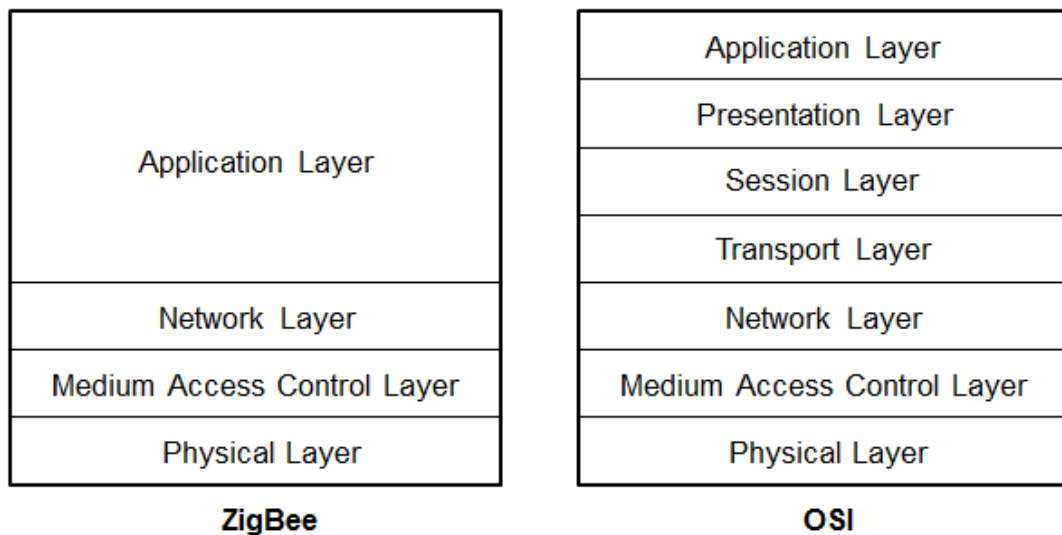


圖 3 ZeeBee 與 OSI 參考模型

ZigBee 堆疊中，第一層到第三層分別是實體層 (Physical Layer)、媒體存取層 (Media Access Control Layer) 和網路層 (Network Layer)，與開放式通訊系統互聯 (Open System Interconnection, 簡稱 OSI) 參考模型一致；而 ZigBee 的應用層 (Application Layer) 則包括了 OSI 的第四層到第七層，如圖 3 所示。其中 ZigBee 的實體層和網路層採用 IEEE 802.15.4 的規範，第三層以上則由 ZigBee 聯盟定義。

ZigBee 網路以其低功耗、短距離的特性，通常被稱為「個人區域網路(Personal Area Network, 簡稱 PAN)」，而 PAN 裡的節點類型可以分為三種：ZigBee 網路協調者 (ZigBee Coordinator, 簡稱 ZC)、ZigBee 路由器 (ZigBee Router, 簡稱 ZR) 和 ZigBee 終端設備 (ZigBee End-Device, 簡稱 ZED)。ZC 負責建立 PAN 讓 ZR 和 ZED 加入，ZED 感測到的資料會經 ZC 或 ZR 轉送，最後所有資料都被送到伺服器上進行處理。

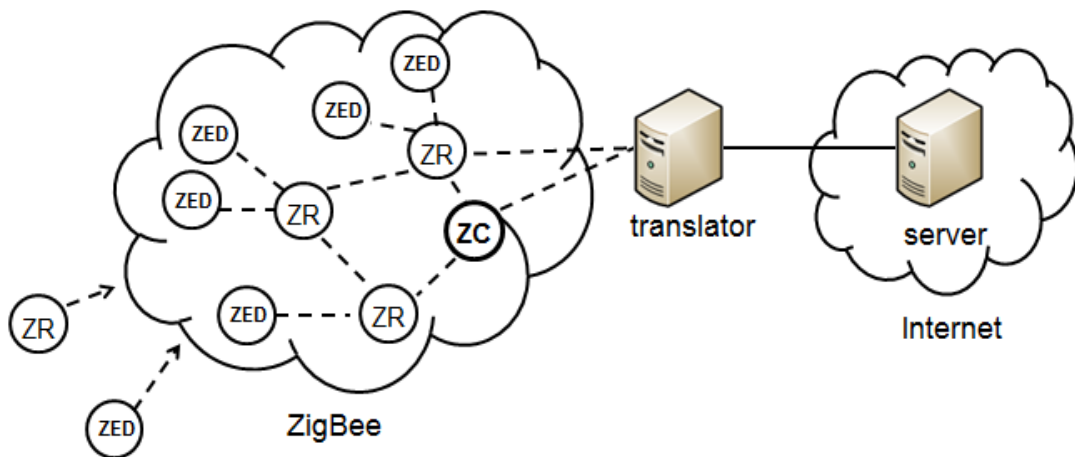


圖 4 ZigBee 網路透過轉換器與 Internet 相連

伺服器可以放置在 PAN 內，讓 ZigBee 節點直接將資料送到伺服器上。但若伺服器位於 ZigBee 最大傳輸距離 (100 公尺) 之外，甚至在一、兩公里的距離之外，這時若要沿路再行佈建許多 ZR 來轉送訊號，在效益上就不夠好。因此在實務上，許多廠商會改讓 ZigBee 的資料透過轉換器轉到 TCP/IP，利用既有的 Internet 送到伺服器，以延伸資料的傳輸範圍。如此一來，不僅訊號比透過多個 ZR 轉送要來得可靠，也省下 ZR 的佈建費用。而 AMI 中，電表到後置網路的通訊也多採用這種作法。在這種

情況下，每一個 PAN 裡就需要一個轉換器當作 ZigBee 網路對外的出口，負責轉換 ZigBee 和 TCP/IP 的封包，如圖 4 所示。

由於無線感測網路常佈署在無遮蔽物、日曬雨淋的環境中，非常容易失去傳輸的可靠性，因此容錯機制被認為是相當重要的議題。ZigBee 現有的研究多是著重在 PAN 內部路由錯誤的容錯機制。以[3]為例，作者探討在 PAN 內部的 Wireless Sensor Network 中，當某些節點故障，造成其他節點形成孤兒（Orphan）時的處理方式，此一議題目前已有許多相關研究進行探究。然而，以 Sensor Network 對外的轉換器而言，單點故障（Single Point of Failure）的問題亦不容忽視。本篇論文的目的則是解決轉換器單點故障的問題，利用冗餘的轉換器，來確保 PAN 對外的通訊亦可具備容錯的功能。

本論文接下來的章節安排如下：第二章簡單回顧在無線及有線網路中現有的冗餘相關應用。第三章提出冗餘的 ZigBee/TCP/IP 轉換器運作方式。第四章說明如何實作系統以驗證可行性，並測量使用此系統前後的封包遺失率是否有所改善。最後在第五章提出我們的結論。

2. 與轉換器冗餘相關的研究

2.1. 無線感測網路中的冗餘及其應用

Curiaic 等人將無線感測網路上的冗餘方法分類成空間冗餘 (Spatial Redundancy)、時間冗餘 (Temporal Redundancy 或 Time Redundancy) 以及資訊冗餘 (Information Redundancy) [4]。

空間冗餘指的是，從不同來源得到某一特定位置的資訊，又分為物理性冗餘 (Physical Redundancy) 和分析性冗餘 (Analytical Redundancy)，前者使用多於一個的感測器來量測某一特定位置的變量，後者則根據鄰居感測器量測到的實際數據，利用數學模型來估算某一特定位置的變量。

時間冗餘被定義為，某一個特定動作會因為時間而失去準確性，為了增加可靠性，在確認結果後會重覆此動作一次以上。又可分成多次測量的時間感測冗餘 (Temporal Sensing Redundancy)，和多次傳送相同封包的時間通訊冗餘 (Temporal communication Redundancy)。

資訊冗餘使用冗餘的資料來偵測錯誤並重建遺失的資訊。例如同位元碼 (Parity Code)，或是被視為概括了同位元碼的抹除碼 (Erasure Code)，他們都是資料冗餘的一種實現，當封包遺失時，就利用冗餘的資料來重製原始訊息，而不需要重傳。

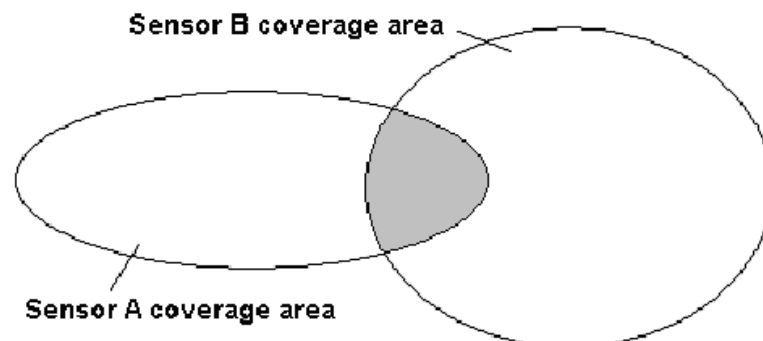


圖 5 使用物理性冗餘的感測器[4]

本篇論文在 ZigBee 網路裡多使用一個轉換器，根據 Curiaac 等人的分類，屬於空間冗餘裡的物理性冗餘。[4]提到的物理性冗餘是應用在密集分佈的感測器上，如圖 5，Sensor A 與 Sensor B 感測到的範圍重疊，因此灰色區域的資料感測便因冗餘增加了可靠度。而本論文是將物理性冗餘應用在收集感測器資料的轉換器上，相較於[4]必須對感測到的冗餘資料做處理，本論文的應用需解決的則是轉換器間如何溝通（將於 3.2 節詳述）。

2.2. 虛擬路由器冗餘協定

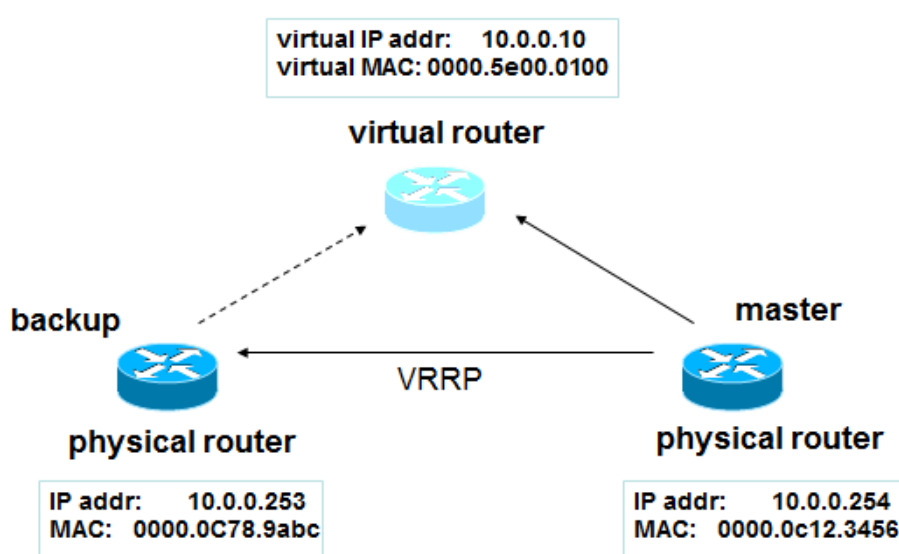


圖 6 VRRP 工作原理

在有線網路中，使用冗餘的方法讓系統具備容錯功能已經是十分常見的應用，虛擬路由器冗餘協定（Virtual Router Redundancy Protocol，簡稱 VRRP）[5]就是一個針對路由器單點故障所提出的容錯機制。運行 VRRP 的路由器稱為「VRRP 路由器」，終端設備可以設定一個虛擬的 IP 位址作為預設路由器的位址，而實際上送到這個虛擬 IP 位址的封包是被一個 VRRP 路由器接收並轉發。在同一區域網路內可以有一個以上的 VRRP 路由器，但同時時間內只會有一台主控路由器「擁有」這個虛擬 IP 位址，負責接收、轉發的工作，其他 VRRP 路由器則是待命的備援路由器。當主控路由器故障後，備援路由器就改變狀態為主控路由器，取得虛擬 IP 位址的所有權，繼續工作，

且終端設備不需改變預設路由器的 IP 位址。因此無論虛擬 IP 位址的所有權在 VRRP 路由器間如何輪替，對終端設備來說，整個系統的表現像是一個具有 IP 位址的「虛擬路由器」，如圖 6 所示。而用來識別不同虛擬路由器的 VRID (Virtual Router Identifier) 也形成虛擬路由器的 MAC (Media Access Control) 位址，以此 MAC 位址回應 ARP (Address Resolution Protocol[6]) 的要求。

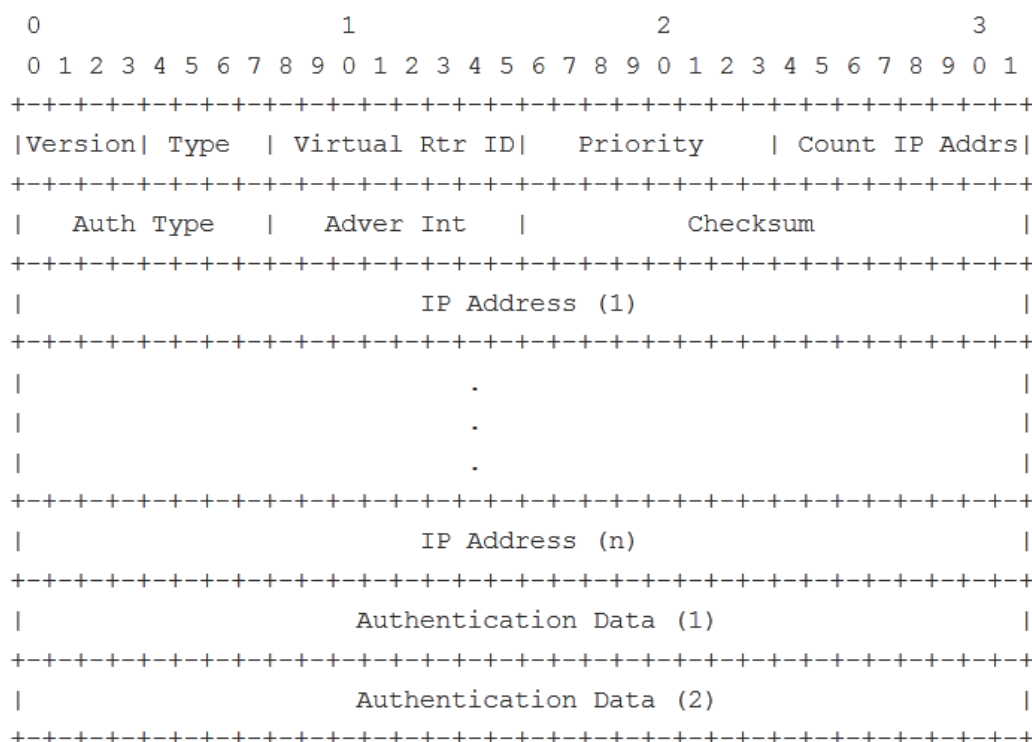


圖 7 VRRP 封包格式[5]

主控路由器會定期發出公告 (Advertisement) 的 VRRP 封包 (圖 7) 給備援路由器，當備援路由器經過一段等待的時間都沒有收到 Advertisement，就判定主控路由器已故障，改變自己的狀態為主控路由器。等待時間定義為三倍的 Advertisement 間隔時間加上偏移時間，Advertisement 間隔時間預設為 1 秒，偏移時間則根據不同備援路由器的優先權而定。優先權高的路由器會得到較小的偏移時間，而較快成為主控路由器。

2.3. 網路位址轉換的冗餘

除了路由器，類似的冗餘方法也應用在網路位址轉換 (Network Address

Translation，簡稱 NAT) 上。相較於路由器純粹轉發收到的封包，NAT 必須將公有 IP 位址及私有 IP 位址做轉換，並維護轉換前後的 IP 位址對應表，因此冗餘設備要照著原來設備上的 IP 位址對應表工作，才能迅速接替工作，於是 Devarapalli 提出了圖 8 的架構[7]。

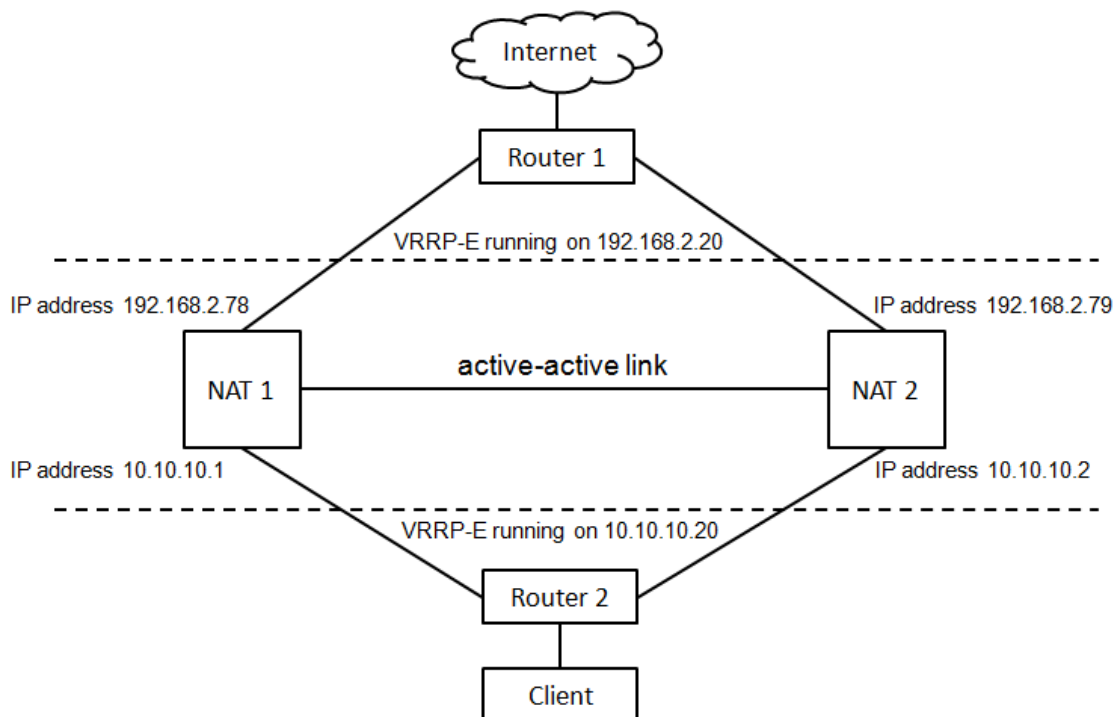


圖 8 Devarapalli 等人提出網路位址轉換的冗餘系統圖

網路中有一台主控 NAT，以及一台以上的備援 NAT 做為冗餘設備，兩台 NAT 共用同樣的 IP 位址，並且在主控與備援的 NAT 間會有專用的連線來同步目前的會議 (Session)。主控 NAT 定時送「活著 (Alive)」的訊息及同步 Session 的資訊給備援 NAT，當備援 NAT 不再收到 Alive，表示主控 NAT 無法正常作用。此時 IP 位址的所有權就會由主控 NAT 轉移給備援 NAT，備援 NAT 便接手原本主控 NAT 負責轉換的 IP 位址，繼續目前的 Session 並處理新的 Session。圖 8 中的 active-active link，表示備援 NAT 平時除了接收同步 Session 的資訊，也會處理自己的轉換工作。

2.4. 網路位址轉換冗餘的連線機制

雖然[7]提出了 NAT 的冗餘方法，但當系統中存在一個第三方（Third Party）設備作為區域網路與廣域網路兩端主機間通訊的橋樑時，這樣的冗餘方法便不能發揮作用。如圖 9 中的位址伺服器（Address Server），即一台會議初始協定（Session Initiation Protocol，簡稱 SIP[8]）註冊機，負責 Local Host 與 Remote Host 間的 SIP 通訊。當 Local Host 透過 NAT1 向 Address Server 註冊自己的位址後，即使 NAT1 一發生故障 NAT2 就馬上接替工作，Remote Host 仍然不能藉由原先保留在 Address Server 上的位址資訊來存取 Local Host。這是由於 NAT1 及 NAT2 對外的 IP 位址不同，因此 NAT2 接手工作後，並不知道故障發生前 Session 的公有 IP 位址與私有 IP 位址對應，而會重新分配一個對應的位址給 Local Host。這會導致 Remote Host 在兩台 NAT 交接後無法存取 Local Host。

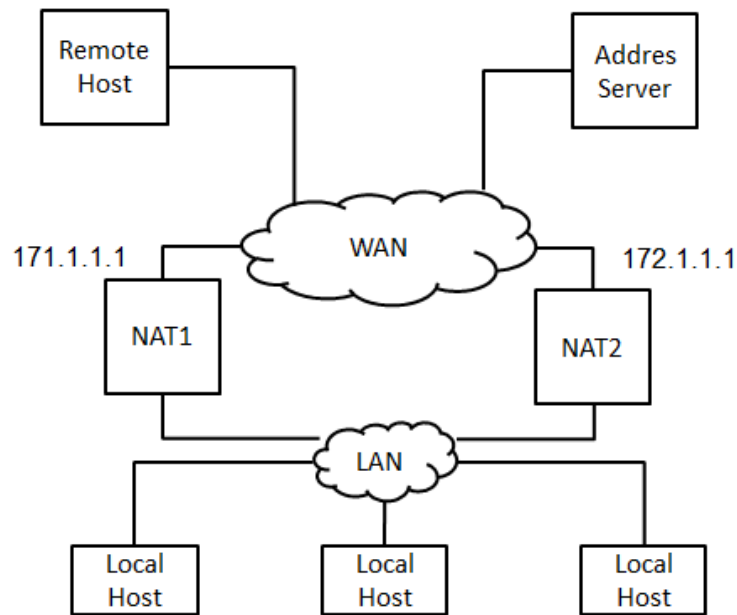


圖 9 位於不同網域的 NAT 冗餘

於是 Biswas 等人提出了克服以上問題的方法[9]，進一步確保兩台 NAT 也能達到有效的容錯。[9]說明主控 NAT 平時維護著一個會議表（Session Table），並且定期將

表格內 Session 的更新訊息送給備援 NAT；而備援 NAT 平時接收 Session 更新資訊並儲存起來，當接手主控 NAT 後，便會根據所儲存的 Session 資訊發送虛擬訊息(Dummy Message) 給 Address Server 和 Remote Host。Address Server 和 Remote Host 收到的虛擬訊息包含了重新轉換後的來源 IP 位址和通訊埠口 (Port)，即可藉此更新 Local Host 的公有 IP 位址。

3. 研究方法

為了使系統具有容錯功能，我們採用物理性冗餘的方式，在同一個 PAN 裡使用兩個轉換器，分別是負責所有封包轉換工作的主控轉換器 (Active Translator)，以及待命等著接手工作的備援轉換器 (Backup Translator)。主控轉換器會定時送心跳 (Heartbeat) 給備援轉換器，以告知備援轉換器自己仍正常運作；當備援轉換器經過一段時間都沒有收到來自主控轉換器的 Heartbeat，便判定主控轉換器已停止運作，並將自己的狀態改為主控，開始轉換封包。對於網路上其他節點來說，他們不會知道主控轉換器已經交換，因為他們永遠都只發送封包到同一個位址。在這一章我們提出一個冗餘轉換器機制，讓兩個轉換器共用同一個位址，以及隨時溝通確認彼此狀態。

3.1. 轉換器的共同位址：Multicast

當主控轉換器故障，並由備援轉換器接手工作後，為了讓網路上其他節點不做任何改變的繼續傳送封包，我們讓兩個轉換器擁有一個共同的位址。

ZigBee 像 TCP/IP 一樣，除了單點播送 (Unicast) 以外，也支援群組播送 (Multicast)。ZigBee Multicast 的作法是，將網路上需要收到封包的節點加入同一個群組 (Group)，它們便擁有一個共同的群組位址 (0x0000~0xFFFF)，而發送端本身無論是不是群組成員，只要把封包送到這個群組位址，則所有群組成員都會收到同樣的封包。使用 Multicast 時，ZigBee 封包在網路層會新增 Multicast 欄位如圖 10。其中 Multicast Mode 用來指明發送端是否為群組成員，分為群組成員的 Member Mode 和非群組成員的 Nonmember Mode，而 NonmemberRadius 和 MaxNonmemberRadius 分別代表非群組成員的傳送範圍和最大傳送範圍。

Bits: 0-1	2-4	5-7
Multicast Mode	NonmemberRadius	MaxNonmemberRadius

圖 10 ZigBee Multicast 控制欄位

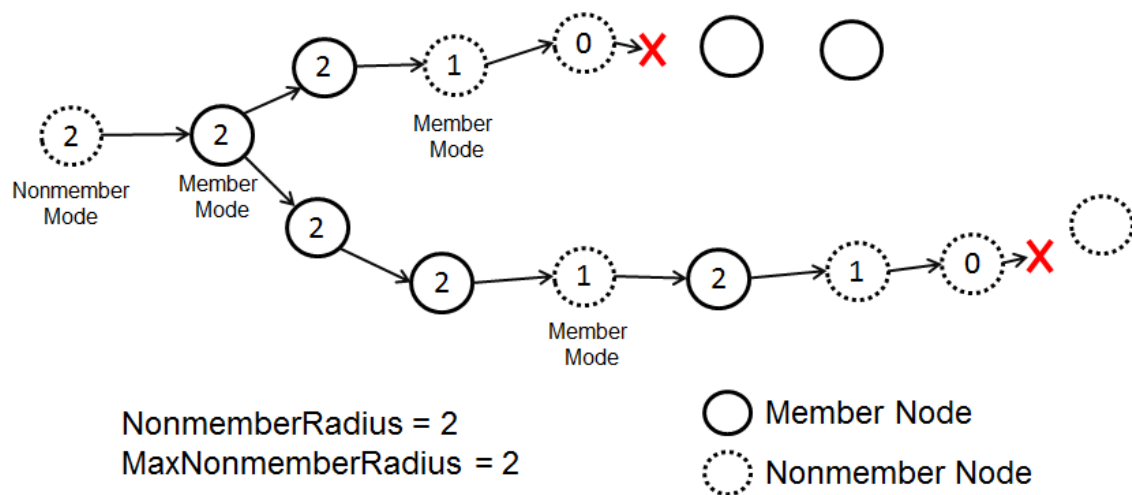


圖 11 ZigBee Multicast 封包傳遞

Multicast 封包的傳遞方式如圖 11 所示，來源節點為非成員節點時，Multicast 封包為 Nonmember Mode，當遇到成員節點時，即轉為 Member Mode，並將 NonmemberRadius 的值改為 MaxNonmemberRadius。然而，Member Mode 的封包遇到非成員節點時，並不會再轉為 Nonmember Mode，而是將 NonmemberRadius 減一後轉發，直到 NonmemberRadius 被減為 0，便丟棄封包。由圖 11 可看出，當 Radius 為 1 時，Multicast 封包只可到達和本身相鄰的成員節點；Radius 為 2 時，中間只允許隔一個非成員節點，以到達未直接相鄰的成員節點。

在 ZigBee 中接收 Multicast 封包的程式流程如圖 12。首先宣告一個裝置本身要加入的群組 Multicast_Group，以及一個用來接收封包的變數 MSGpkt。而 MULTICAST_ENDPOINT 表示 ZigBee 裝置中的 Endpoint，相當於 TCP/IP 裡使用的 Port，用以區分同一裝置上提供的不同服務，在收送封包時是必要的，範圍是 1~240，在此我們給定一個隨意的值為 20。接著設定所宣告的群組 ID (16 位元) 及群組名稱，並使用 aps_AddGroup(...)，將此裝置的 Endpoint 加入群組，等同於將此裝置加入群組。最後以 MSGpkt = osal_msg_receive(...) 接收封包，即可對收到的封包 MSGpkt 進行處理。

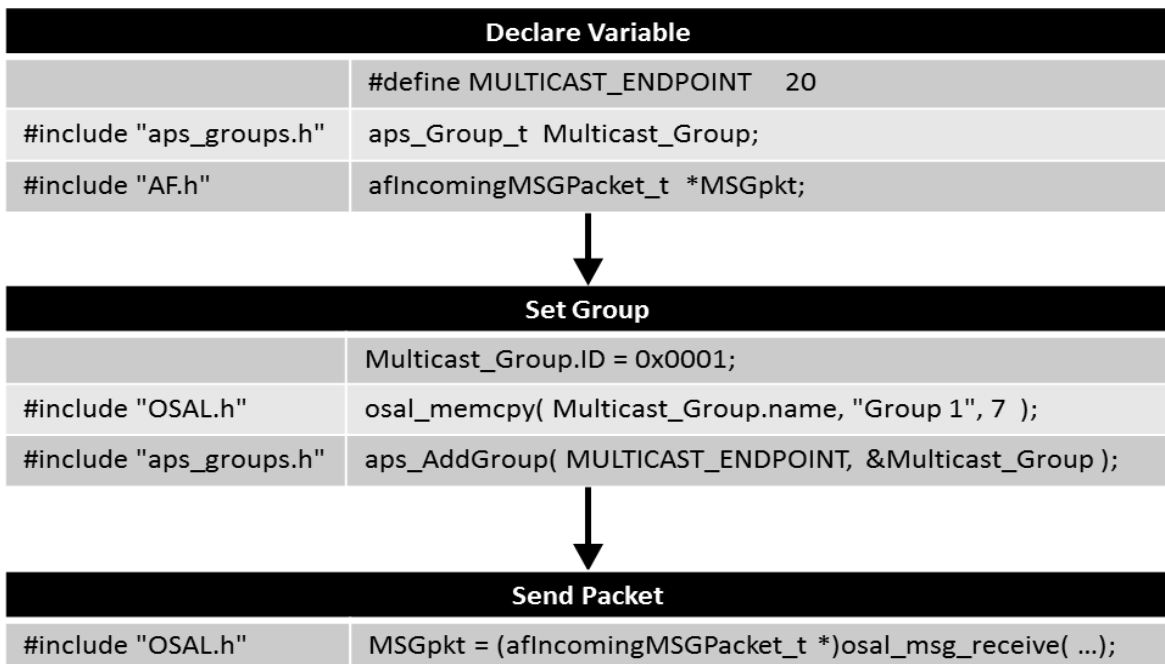


圖 12 ZigBee Multicast 封包接收端流程

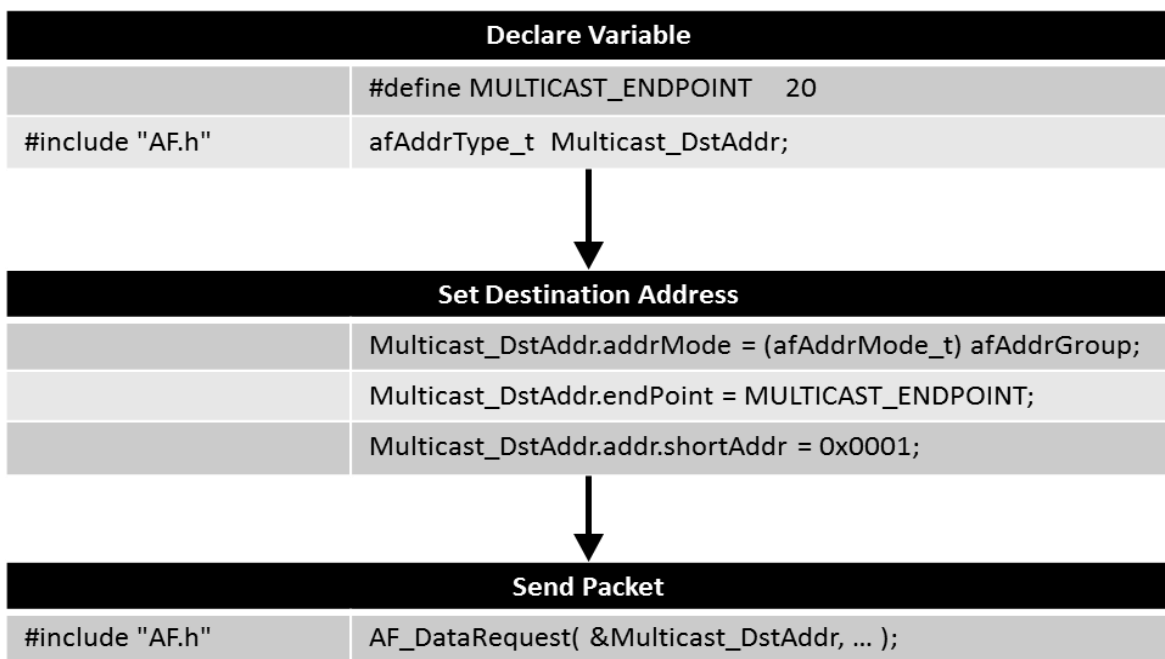


圖 13 ZigBee Multicast 封包發送端流程

在 ZigBee 中發送 Multicast 封包的程式流程如圖 13。首先宣告一個目的位址 Multicast_DstAddr，以及封包接收端使用的 Endpoint MULTICAST_ENDPOINT。接著設定所宣告的目的位址各項資訊，將 Multicast_DstAddr.addrMode 設成 afAddrGroup 表示此位址為群組位址，並在 Multicast_DstAddr.endPoint 及 Multicast_DstAddr.addr.shortAddr 填入接收端的 Endpoint 和群組 ID。最後使用 AF_DataRequest(...) 將封包發送至 Multicast_DstAddr，便可讓群組成員都收到此封包。

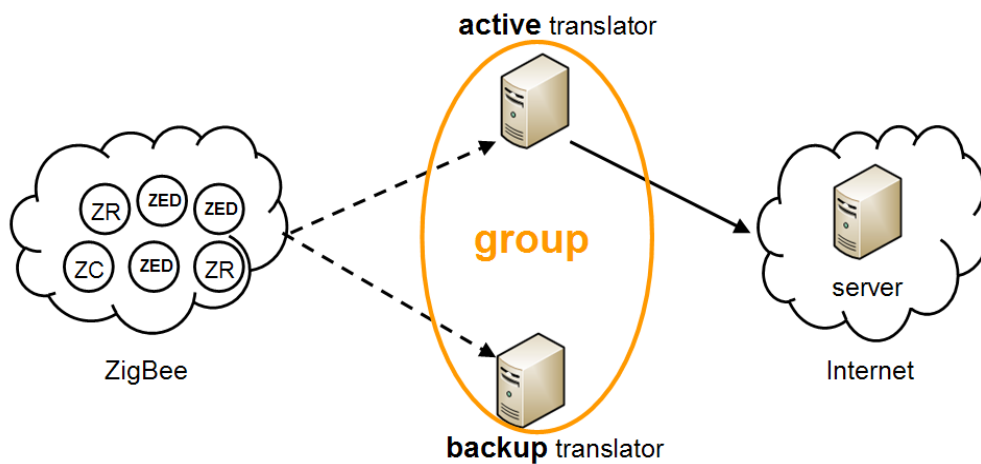


圖 14 ZigBee 節點送資料給伺服器

在 ZigBee 網路中，我們讓主控轉換器與備援轉換器都加入「轉換器群組」中。如此一來，ZigBee 網路中的各節點只要將資料送到這個群組位址，兩台轉換器都能收到這筆資料，但只有主控轉換器會繼續將資料轉送給伺服器，而備援轉換器則直接將資料丟棄，如圖 14。同理，主控轉換器與備援轉換器在 TCP/IP 網路中也加入一個轉換器群組，Internet 上的伺服器端把封包傳給這個群組位址時，主控轉換器與備援轉換器都會收到。再由主控轉換器送到 ZigBee 節點，而備援轉換器會將封包丟棄。

3.2. 轉換器間的溝通：Heartbeat

轉換器收到 Multicast 的封包後，只有主控轉換器需要對封包進行處理並轉發；備援轉換器則是在主控轉換器失效後，才會代替主控轉換器工作。如此一來，兩台轉換器便需要定期交換訊息來確認彼此的狀態為何，以達到有效率的備援。因此，我們仿效第 2.2 節中，虛擬路由器冗餘協定的 Advertisement，使轉換器能夠隨時溝通並即時切換狀態。

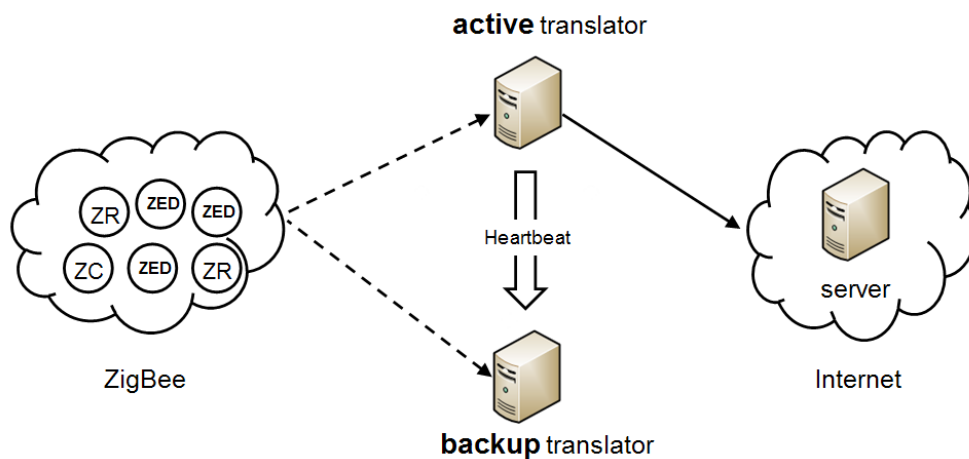


圖 15 主控轉換器送心跳給備援轉換器

轉換器有自己的優先序 (Priority)，優先序較大者成為主控。主控轉換器定期發送給備援轉換器的訊息，我們稱為心跳 (Heartbeat)，如圖 15。備援轉換器依據收到的 Heartbeat 來決定自己是否繼續維持在備援狀態。如果過了三次間隔時間都沒有收到心跳，表示主控轉換器已經停止運作，這時備援轉換器便將自己的狀態轉換成主控。Heartbeat 裡含有轉換器的優先序，如果主控轉換器收到了來自其他轉換器的 Heartbeat，而且顯示對方的優先序比自己高，則必須把主控權交出去，切換為備援狀態。

轉換器的狀態切換流程如圖 16。trs_priority 是轉換器本身的優先序，範圍是 1~255，若優先序是 255 必為主控轉換器，而小於主控轉換器的優先序則為備援轉換器。

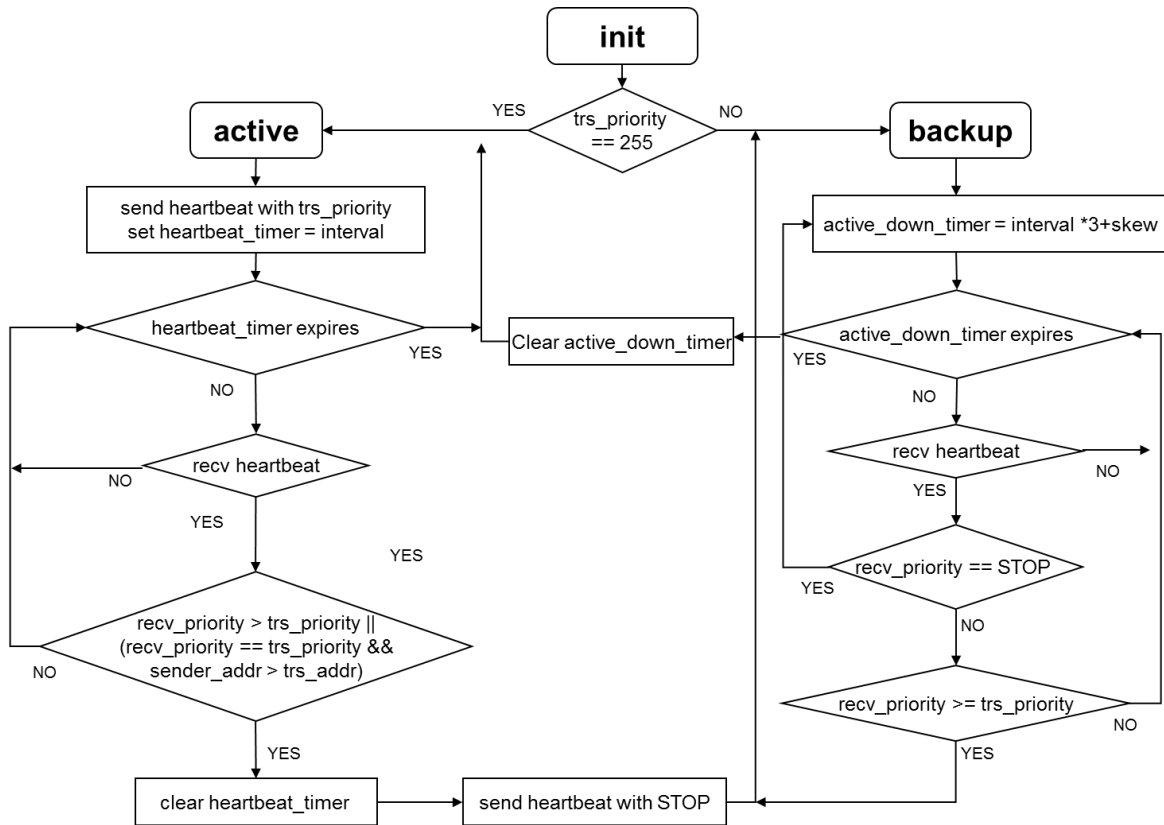


圖 16 轉換器狀態切換流程圖

當主控轉換器開始執行，首先判斷 `trs_priority` 與 255 相等後即進入 Active 狀態。Active 狀態主要做的事就是發送 Heartbeat，因此一進入 Active 的迴圈便發出 Heartbeat，並將 `heartbeat_timer` 設為每次要發送 Heartbeat 的間隔時間 (`interval`)，讓 `heartbeat_timer` 開始倒數。接著在倒數的過程中不斷檢查是否收到來自其他轉換器的 Heartbeat；如果收到了 Heartbeat，且其中包含的 `recv_priority` 資訊大於 `trs_priority`，也就是對方的優先序高於自己，便讓出主控權，進入 Backup 狀態。否則，繼續檢查 `heartbeat_timer` 倒數是否已經歸零，歸零後重複迴圈內的動作。

當備援轉換器執行後，判斷 `trs_priority` 不等於 255，即進入 Backup 狀態。Backup 狀態主要做的事就是確認是否收到來自主控轉換器的 Heartbeat，因此一進入 Backup 的迴圈便將 `active_down_timer` 設為一個等待主控轉換器失效的時間，並

開始倒數。倒數的時間為三倍的 `interval+skew`，`skew=(256-trs_priority)/256`。接著在倒數的過程中不斷檢查是否收到 Heartbeat；如果收到了，且其中包含的 `recv_priority` 資訊大於 `trs_priority`，也就是對方的優先序高於自己，表示主控轉換器仍正常運作，便重複迴圈內的動作，重新設定 `active_down_timer` 進行倒數。否則，繼續檢查 `active_down_timer` 是否已減為零；若減為零表示主控轉換器已失效，備援轉換器隨即取得主控權，進入 Active 狀態。另外，如果收到的 Heartbeat 中，`recv_priority == STOP(0)`，表示主控轉換器要停止運作，備援轉換器會重新設定 `active_down_timer` 進行倒數，最後便由 `trs_priority` 大的轉換器優先取得主控權，進入 Active 狀態。

本節最後附上 `heartbeat_timer` 和 `active_down_timer` 在 Heartbeat 程式中的相關部分，表 1 為計算 Timer 時所用到的定義，表 2 為 Active 狀態時 Heartbeat Timer 的相關程式，表 3 則為 Backup 狀態時 Active Down Timer 的部分。

表 1 Timer 相關計算器

```
#define TIMER_SET( val, delta ) (val) = TIMER_CLK() + (delta)
#define TIMER_SUB( t1, t2 )
    ((int32_t)((uint32_t)t1)-((uint32_t)t2))
#define TIMER_DELTA( val )    TIMER_SUB( val, TIMER_CLK() )
#define TIMER_EXPIRED( val ) ((val) && TIMER_DELTA(val)<=0)
#define TIMER_CLR( val )    (val) = 0
#define TIMER_IS_RUNNING( val ) (val)
#define TIMER_HZ            1000000
uint32_t TIMER_CLK( void )
{
    struct timeval tv;
```

```

    gettimeofday(&tv, NULL );

    return tv.tv_sec* TIMER_HZ+tv.tv_usec;
}

#define TIMER_SKEW( srv )

    ((256-(srv)->priority)*TIMER_HZ/256)

```

表 2 Heartbeat Timer 相關程式

```

TIMER_SET( trs->heartbeat_timer, trs->heartbeat_int );

trs->state = STATE_ACTIVE;

if( TIMER_EXPIRED(trs->heartbeat_timer) )
{
    send_heartbeat( trs, trs->priority );

    TIMER_SET(trs->heartbeat_timer, trs->heartbeat_int);

    return;
}

if( ! read_packet() ) return;

if( recv->priority == 0 )
{
    return;
}else if(recv->priority > trs->priority ||
    (recv->priority == trs->priority &&

```

```

    ntohl(iph->saddr) > trs->vif.ipaddr) )
{
    TIMER_CLR( trs->heartbeat_timer );

    send_heartbeat ( trs, STOP );

    trs->state     = STATE_BACK;

    delay = 3*trs->heartbeat_int + TIMER_SKEW(trs);

    TIMER_SET( trs->active_down_timer, delay );
}

```

表 3 Active Down Timer 相關程式

```

delay = 3*trs->heartbeat_int + TIMER_SKEW(trs);
TIMER_SET( trs->active_down_timer, delay );

trs->state     = STATE_BACK;

if(TIMER_EXPIRED(trs->active_down_timer) )
{
    TIMER_CLR( trs-> active_down_timer );

    trs->state = STATE_ACTIVE;

    send_heartbeat ( trs, trs->priority );

    TIMER_SET( trs->heartbeat_timer, trs->heartbeat_int );

    return;
}

if( !read_packet() )    return;

```

```
if (recv->priority == 0 )
{
    TIMER_SET( trs->active_down_timer, TIMER_SKEW(trs) );
} else if( recv->priority >= trs->priority )
{
    TIMER_SET( trs->active_down_timer, delay );
}
```

4. 系統架構與實作

4.1. 系統架構

本論文使用三台個人電腦及一組 ZigBee 開發套件進行實驗。其中兩台電腦作為轉換器，一台為伺服器，作業系統皆為 Linux CentOS 5.5，三台電腦透過 Ethernet 介面互相通訊。ZigBee 開發套件則選用德州儀器 (Texas Instruments, TI) 的 CC2530ZDK (ZigBee Development Kit)，將兩個擁有 RS232 介面的 ZigBee 開發板做為 ZigBee Router，並與轉換器連接，透過此方式使轉換器等同於擁有 802.15.4 介面，成為 ZigBee 網路的節點之一。另將兩個電池裝置的開發板做為 ZigBee Coordinator 與 ZigBee End-Device，便與 ZigBee Router 形成一個 ZigBee PAN。如圖 17。

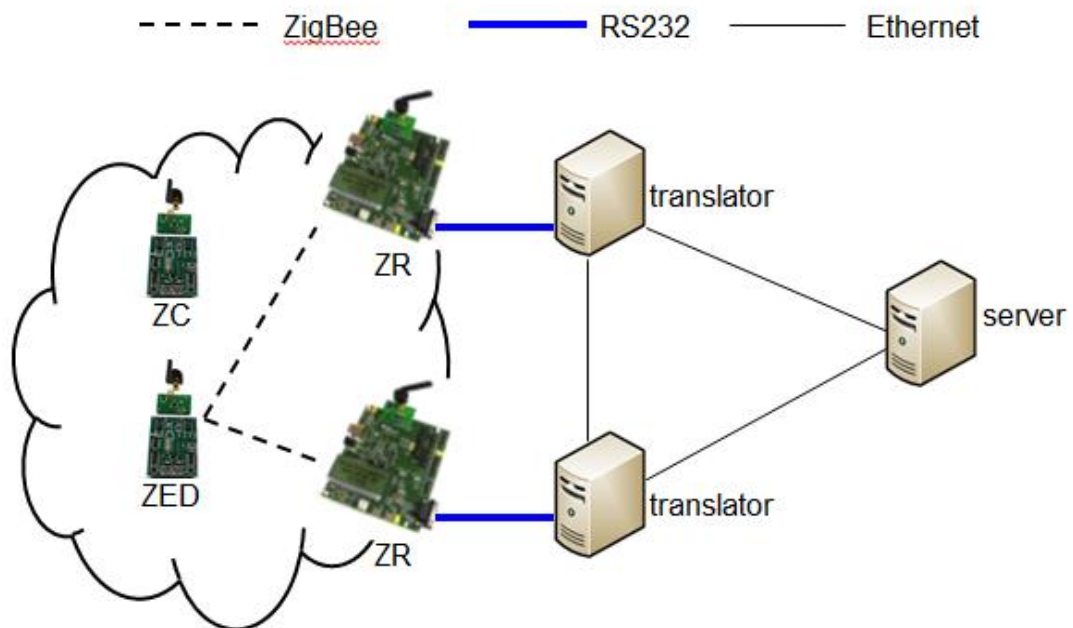


圖 17 實驗平台

系統使用到的模組分成四個部分：(1) ZigBee End-Device 與 ZigBee Router 的封包傳遞，使用到 ZigBee 堆疊中的應用層 Application Layer (APL)、(2) ZigBee Router 和轉換器的結合，則使用 Universal Asynchronous Receiver/Transmitter (UART)來傳輸資料、(3) 轉換器之間利用 Heartbeat 溝通彼此的狀態、(4) 轉換器透過 socket 將封包送到伺服器，架構如圖 18 所示。

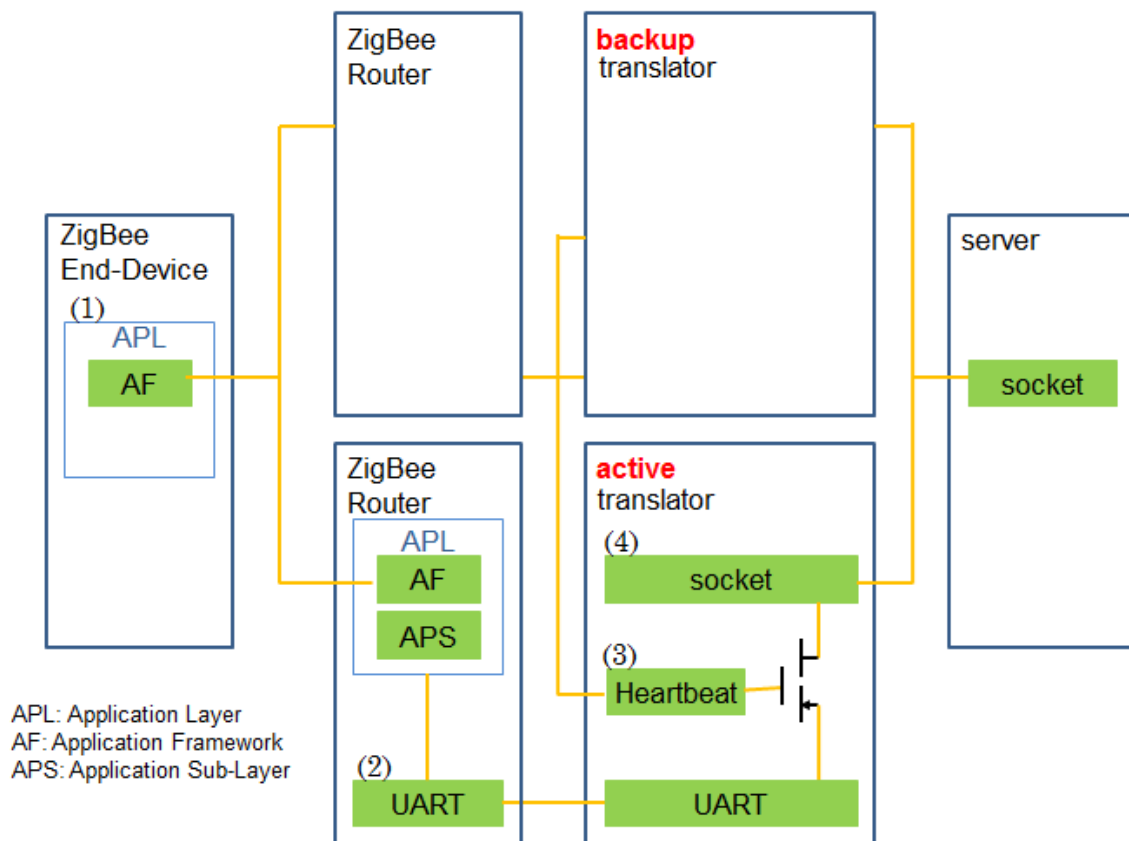


圖 18 系統架構

4.2. 封包遺失率實驗與結果

為了瞭解使用備援機制後的效益如何，我們首先實驗沒有備援轉換器的情形。對於每一個由 ZigBee End-Device 發送到伺服器的封包，我們給定一個故障的比率，以及每次故障的時間長度。每當故障發生時，便忽略接下來一定數量的封包，不進行轉

發，藉此模擬封包遺失。而 ZigBee End-Device 發送封包的頻率為一秒一個，因此每次故障發生時所忽略的封包數即為故障的時間秒數。由於不同的故障比率與故障時間，分別會有不同的封包遺失率，我們針對 1/10、1/100 的故障比率，以及 2~20 秒不等的故障時間，分別進行每回合 100 個封包、共十回合 1000 個封包的測試，得到的數據如表 4，所畫成的長條圖如圖 19。

表 4 無備援機制，故障比率與故障時間對封包遺失率的影響

packet loss (%)		fail time (sec)					
		2	3	5	10	15	20
fail rate	1/10	23.5	36.5	40.5	54	62.2	71.6
	1/100	3.3	3.6	6.6	10.1	12.7	23.5

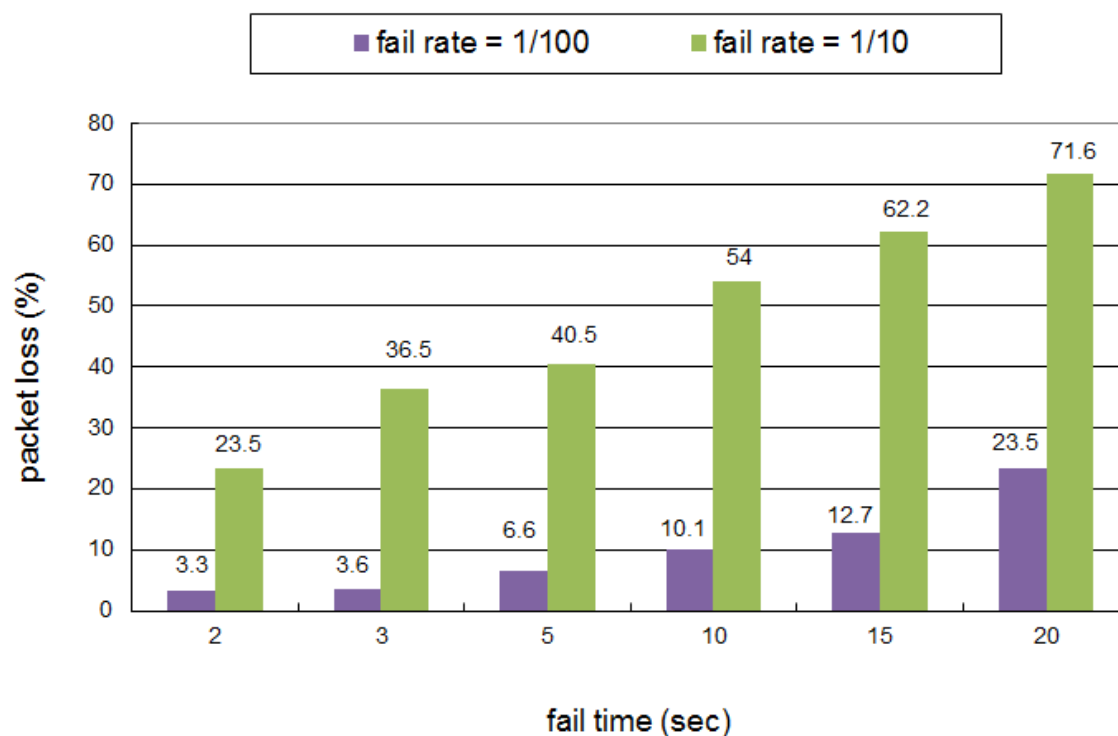


圖 19 無備援機制，故障比率與故障時間對封包遺失率的影響

接下來實驗加入備援機制後的封包遺失率改善情形。由於智慧型電表基礎建設的實際佈建中，轉換器是在嵌入式系統上運行，我們假設故障發生後，自動重新啟動系統的時間為 10 秒。因此我們在故障時間為 10 秒的情況下，測試不同 Heartbeat 間隔時間與不同故障比率對封包遺失率的影響，實驗長度為每回合 100 個封包、共十回合 1000 個封包，得到的數據如表 5，所畫成的長條圖如圖 20。

表 5 備援機制下，故障比率與心跳間隔時間對封包遺失率的影響

packet loss (%)		heartbeat interval (sec)							
		0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
fail rate	1/10	26.1	28.5	31.1	36.1	37.9	38.6	35.1	39.9
	1/100	0.1	1.0	1.2	2.8	3.3	3.0	3.1	3.3

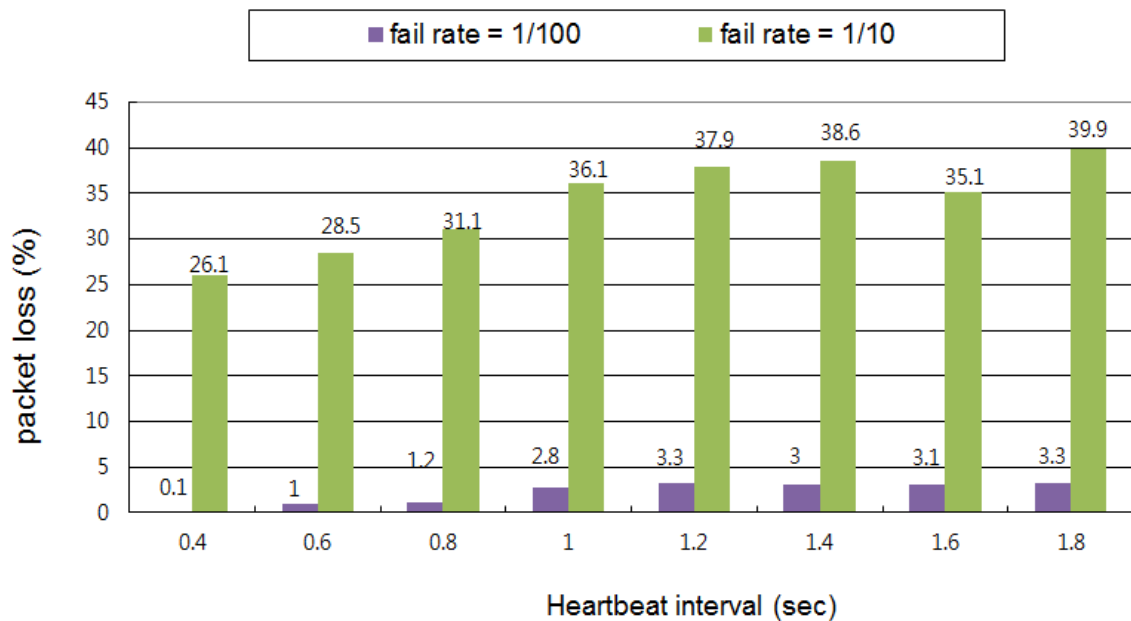


圖 20 備援機制下，故障比率與心跳間隔時間對封包遺失率的影響

根據表 4，沒有備援機制下，故障時間 10 秒時，故障比率 1/10、1/100 的封包遺失率分別是 54%、10.1%。而由表 5 可以看到，故障時間同樣是 10 秒，故障比率同樣也是 1/10、1/100 的情形，在加入備援機制後，使用 0.4~1.8 秒的 Heartbeat 間隔時間，封包遺失率都低於沒有備援機制的 54%、10.1%。

假設無備援的封包遺失率是 loss1%，有備援的封包遺失率是 loss2%，我們使用下列公式來計算改善的效能比率：

$$(loss1 - loss2) / loss1 \times 100\% = \text{改善比率}$$

得到的結果為表 6，所繪製成的曲線圖為圖 21。

表 6 封包遺失的改善效能比率

improvement (%)		heartbeat interval (sec)							
		0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8
fail rate	1/10	51.67	47.22	42.41	33.15	29.81	28.52	35.00	26.11
	1/100	99.01	90.10	88.12	72.28	67.33	70.30	69.31	67.33

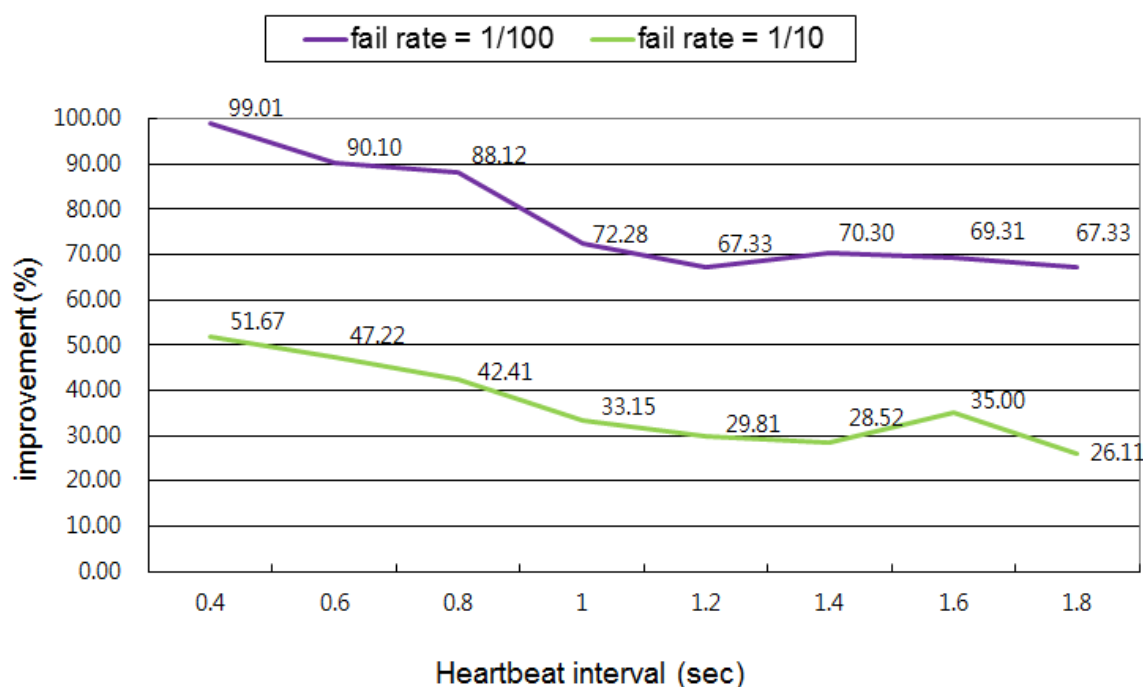


圖 21 封包遺失的改善效能比率

由表 6 可看出加入備援機制後，封包遺失的改善效能比都在 26.11% 以上。另外值得注意的是，表中的效能並不一定隨著 Heartbeat 時間增長而降低，但圖 21 顯示的趨勢仍大略是向下的曲線，因此估計是由於實驗次數不足所造成的誤差。

5. 結論

為了解決智慧電網的通訊系統中的單點故障問題，本論文提出了冗餘的 ZigBee/TCPIP 轉換器。在 ZigBee 與 Internet 網路中分別利用 Multicast 讓兩個轉換器擁有共同的群組位址，當主控轉換器失敗後，網路上其他節點也不需改變封包的目的位址，直接由收到同樣封包的備援轉換器接手封包轉發工作。轉換器之間的溝通則仿效 VRRP 的機制，備援轉換器會定時收到主控轉換器傳送的 Heartbeat，來確認系統仍正常運作；當超過一定的等待時間尚未收到 Heartbeat，即判定主控轉換器已故障，備援轉換器很快便轉換狀態為主控轉換器以接替工作。透過 Multicast 與 Heartbeat，使得主控轉換器故障後備援轉換器能快速交接，且不須對其他節點原先配置好的目的位址做任何更動。最後實作此系統並測量所改善的封包遺失率，驗證了此系統的可行性。

參考文獻

- [1] H Farhangi, “The path of the smart grid”, *IEEE Power and Energy Magazine*, January/February 2009, pp.18-28.
- [2] ZigBee Specification, “ZigBee Specification”, ZigBee document 053474r17, ZigBee Standard Organization, January 2008.
- [3] Skender Ben Attia, “Fault-Tolerance Mechanisms for Zigbee Wireless Sensor Networks”, MSc Thesis, SupCom, Tunis, Tunisia, HURRAY-TR-070710, July 2007. Work carried out at the CISTER-ISEP Research Unit, Polytechnic Institute of Porto (Portugal).
- [4] D. Curiac, C. Volosencu, D. Pescaru, L. Jurica and A. Doboli, “*Redundancy and Its Applications in Wireless Sensor Networks: A Survey*”, WSEAS Transactions on Computers, Volume 8, Issue 4, 2009, pp.705-714.
- [5] R. Hinden, “*Virtual Router Redundancy Protocol (VRRP)*”, IETF RFC 3768, April 2004.
- [6] David C. Plummer, “*An Ethernet Address Resolution Protocol*”, IETF RFC 826, November 1982.
- [7] Sridhar J. Devarapalli, “*Redundancy Support for Network Address Translation (NAT)*”, United States Patent, No. 7,647,427, 2010.
- [8] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, “SIP: Session Initiation Protocol,” IETF RFC 3261, June 2002.
- [9] Kaushik P. Biswas and Senthil M. Sivakumar, “*Mechanism for Providing Connectivity in NAT Redundant Fail-over Scenarios in Unshared Address-Space*”, United States Patent, No. 7,518,987 B2, 2009.