行政院國家科學委員會專題研究計畫 成果報告

電視頻帶白空間之寬頻行動雲端感知無線網路
研究成果報告(完整版)

計 畫 類 別 ： 個別型
計 畫 編 號 ： NSC 99-3113-P-009-004-
執 行 期 間 ： 99 年 11 月 01 日至 101 年 01 月 31 日
執 行 單 位 ： 國立交通大學資訊工程學系（所）

計 畫 主 持 人 ： 王協源

計畫參與人員： 學士級-專任助理人員：王淳渝
碩士班研究生-兼任助理人員：陳弼揚
碩士班研究生-兼任助理人員：王柏凡
碩士班研究生-兼任助理人員：張文哲
碩士班研究生-兼任助理人員：陳建都
碩士班研究生-兼任助理人員：李韻立
碩士班研究生-兼任助理人員：蔡佳玟
碩士班研究生-兼任助理人員：莫尚儒
碩士班研究生-兼任助理人員：周煜軒
碩士班研究生-兼任助理人員：梁喬峰
碩士班研究生-兼任助理人員：李宗霖
碩士班研究生-兼任助理人員：林佳賢
碩士班研究生-兼任助理人員：張嘉軒
碩士班研究生-兼任助理人員：劉人僖
碩士班研究生-兼任助理人員：周詩梵
碩士班研究生-兼任助理人員：郭俊義
碩士班研究生-兼任助理人員：殷裕雄
碩士班研究生-兼任助理人員：江培立
碩士班研究生-兼任助理人員：林巧桐
碩士班研究生-兼任助理人員：鄭仲傑
碩士班研究生-兼任助理人員：陳人維
碩士班研究生-兼任助理人員：蕭永中
大專生-兼任助理人員：蔡東倫

博士班研究生-兼任助理人員：郭子綺
博士班研究生-兼任助理人員：江長廷
博士班研究生-兼任助理人員：曾俊凱
博士班研究生-兼任助理人員：邱麟凱
博士班研究生-兼任助理人員：邱新栗
博士班研究生-兼任助理人員：邱榮東
博士班研究生-兼任助理人員：黃汀華

報 告 附 件 ：出席國際會議研究心得報告及發表論文

公 開 資 訊 ：本計畫可公開查詢

中 華 民 國 101 年 02 月 23 日

中文摘要： 近年來，FCC 公告了第一個使用電視空白頻帶的商業應用規範，為感知無線電的發展走出了突破性的一步。另一方面，雲端運算的快速發展，使得利用現有的無線通訊裝置建構都會區網路在經濟上變成可行。在面對這兩種技術帶來的機會與挑戰，我們提出一種創新的 CRCN(CR Cloud Network)模型，用以實現感知無線電在電視頻帶上的通訊。在運用雲端強大而彈性的運算資源下，我們發展了合作式頻譜估測(cooperative spectrum sensing)演算法估測授權使用者(PU)的無線功率圖(radio power map)，並建立資料庫儲存其成果。此感知無線雲(CR Cloud，CRC)系統已實作在微軟 Azure 雲端平台，並能夠支援合作式頻譜估測、感知無線電通道存取(CR channel access)以及行動管理(mobility management)。同時，我們也發展一個媒體層通訊協定(medium access control protocol)用以蒐集回報的資訊並且提供存取感知無線雲的服務以及電視空白頻帶的通訊資源。藉由所實作的 CRCN 原型，我們可以量測許多重要的網路參數，如合作式頻譜估測的誤差、通道閒置時間的延遲(channel vacating delay)以及以雲端為基礎的換手(handover)時間。藉由這樣的實作，讓我們能夠評估 CRCN 模型的設計以及概念。除此之外，為了進一步改善合作式頻譜估測演算法的效能，我們探討把它執行在 Amazon EC2 公有雲上的效能瓶頸。我們發現 Amazon EC2 公有雲上所提供的 Hadoop 計算平台並不適合此合作式頻譜估測演算法的執行，因此我們仔細研究 Hadoop 平台的設計及實作。經過我們改良後，目前我們已經能大幅增進此合作式頻譜估測演算法在 Amazon EC2 公有雲上的執行效能。

中文關鍵詞： 雲端運算 感知無線電 Channel access control； resource allocation； Bayesian Sparse Learning

英文摘要：

英文關鍵詞：

行政院國家科學委員會補助專題研究計畫 ■成果報告 □期中進度報告

電視頻帶白空間之寬頻行動雲端感知無線網路

計畫類別：■個別型計畫　□整合型計畫
計畫編號：NSC 99 －3113 －P －009－004－
執行期間：99 年 11 月 1 日至 101 年 1 月 31 日(一年期計畫)

執行機構及系所：國立交通大學 資訊工程學系

計畫主持人：王協源 教授

計畫參與人員：伍紹勳教授、趙禧綠教授

成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本計畫除繳交成果報告外，另須繳交以下出國心得報告：
□赴國外出差或研習心得報告
□赴大陸地區出差或研習心得報告
■出席國際學術會議心得報告
□國際合作研究計畫國外研究報告

處理方式：除列管計畫及下列情形者外，得立即公開查詢
■涉及專利或其他智慧財產權，□一年■二年後可公開查詢

中　華　民　國　101 年 1 月 31 日

中英文摘要及關鍵字

The FCC's approval for the first commercial operation in TV white space (TVWS) gives a new momentum to the development of cognitive radio (CR) in TVWS. On the other hand, the rapid growth of Cloud computing makes it possible and more economical to build a CR metropolitan area network with commodity hardware. In view of the opportunity and challenges brought about by these two technologies, we propose a CR Cloud Networking (CRCN) model that is able to support CR access in TVWS. Making use of the flexible and vast computing capacity of the Cloud, a database and a cooperative spectrum sensing (CSS) algorithm that estimates the radio power map of licensed users are realized on a CR Cloud (CRC) implemented with Microsoft's Windows Azure Cloud platform. The CRC can support CSS, CR channel access and mobility management. A medium access control protocol is also developed for this CRCN model to collect sensing reports and provide access to the TVWS and CRC services. Through this CRCN prototype, important network parameters such as the mean squared errors in CSS, the CR channel vacating delay and the Cloud-based handover time are evaluated for the design and deployment of the CRCN concept. In addition, to further improve the performance of the CSS algorithm, we investigate running it on the popular Amazon EC2 public cloud using the Hadoop computing platform provided by Amazon. We found that the design and implementation of Hadoop do not suit the CSS algorithm well. We have successfully improved Hadoop to achieve great performance speedup over Amazon EC2 public cloud.

近年來，FCC公告了第一個使用電視空白頻帶的商業應用規範，為感知無線電的發展走出了突破性的一步。另一方面，雲端運算的快速發展，使得利用現有的無線通訊裝置建構都會區網路在經濟上變成可行。在面對這兩種技術帶來的機會與挑戰，我們提出一種創新的CRCN(CR Cloud Network)模型，用以實現感知無線電在電視頻帶上的通訊。在運用雲端強大而彈性的運算資源下，我們發展了合作式頻譜估測(cooperative spectrum sensing)演算法估測授權使用者(PU)的無線功率圖(radio power map)，並建立資料庫儲存其成果。此感知無線雲(CR Cloud，CRC)系統已實作在微軟Azure雲端平台，並能夠支援合作式頻譜估測、感知無線電通道存取(CR channel access)以及行動管理(mobility management)。同時，我們也發展一個媒體層通訊協定(medium access control protocol)用以蒐集回報的資訊並且提供存取感知無線雲的服務以及電視空白頻帶的通訊資源。藉由所實作的CRCN原型，我們可以量測許多重要的網路參數，如合作式頻譜估測的誤差、通道閒置時間的延遲(channel vacating delay)以及以雲端為基礎的換手(handover)時間。藉由這樣的實作，讓我們能夠評估CRCN模型的設計以及概念。除此之外，為了進一步改善合作式頻譜估測演算法的效能，我們探討把它執行在Amazon EC2公有雲上的效能瓶頸。我們發現Amazon EC2公有雲上所提供的 Hadoop 計算平台並不適合此合作式頻譜估測演算法的執行，因此我們仔細研究 Hadoop 平台的設計及實作。經過我們改良後，目前我們已經能大幅增進此合作式頻譜估測演算法在Amazon EC2公有雲上的執行效能。

Keywords—cloud computing; Cognitive Radio; Channel access control; resource allocation; Bayesian Sparse Learning

目錄

圖目錄

表目錄

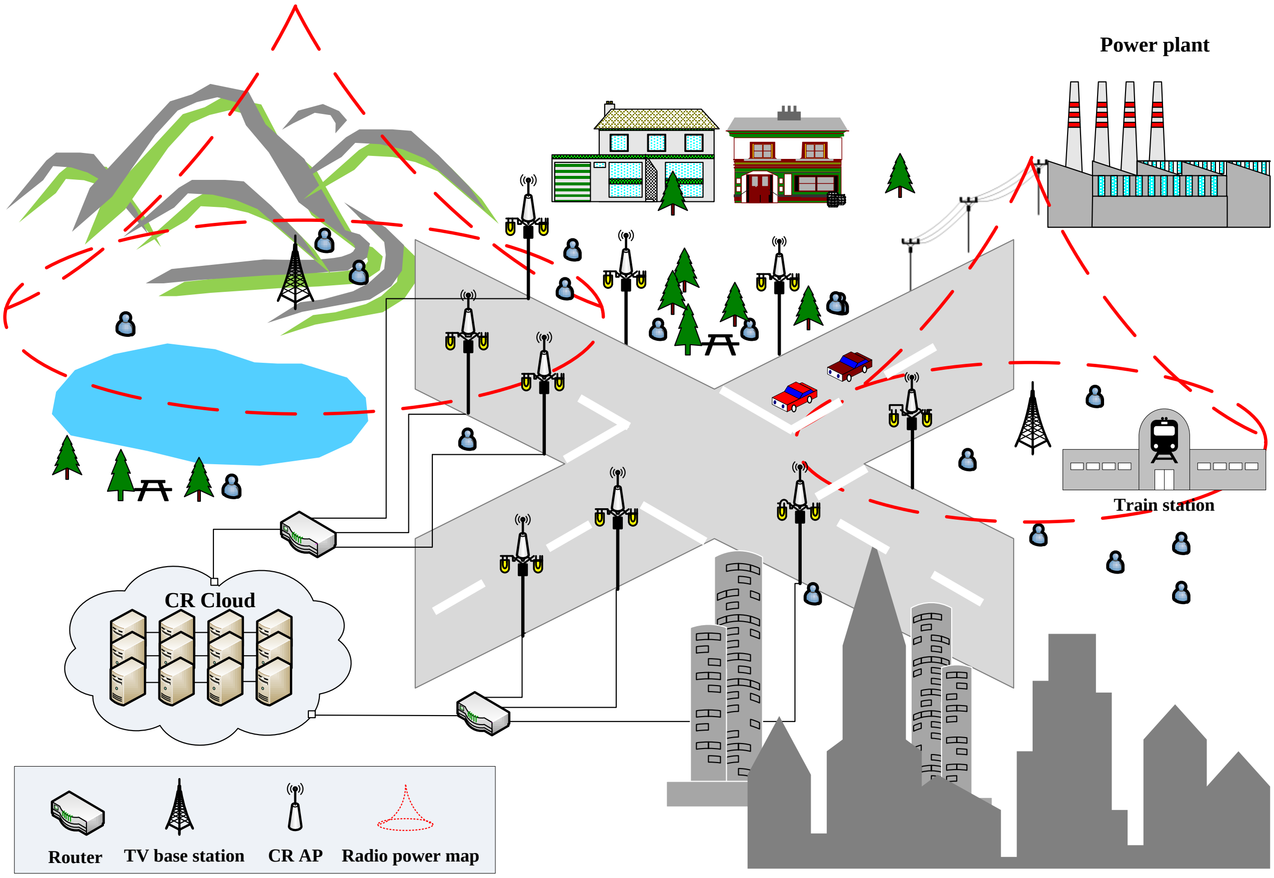
一、 前言

感知無線電技術(Cognitive Radio，CR)[1]為 Joseph Mitola III 所提出，比起傳統的無線通訊，感知無線技術藉由對於優先權使用者(Primary User，PU)的保護，提供無執照使用者(Secondary User，SU)對頻帶的存取，以改善頻譜利用的效率。在目前的無線通訊環境下，根據統計，頻譜的使用率約略只有 25%[2]。因此，CR 技術被視為下一世代通訊的選項。由於上述因素，各國的電信管理當局也開始注重 CR 的發展。美國聯邦電信委員會(Federal Communications Commission，FCC)便利用收回類比電視頻譜的契機，開放 CR 的測試並制定在電視白頻帶(TV White Space，TV WS)上 CR 的使用規範，此一舉動，促使各大國際組織開始定義 CR 的通訊協定，較為知名的有：IEEE 802.22、ECMA 392 以及 IEEE 802.16m 等。

在電視白頻帶上，PU 為數位電視基地台，為了保護數位電視的收訊品質，FCC 要求 CR 的營運者必須準確偵測出 PU 的功率，設立 SU 可用門檻為-116dBm，並且建立 PU 的資料庫，記錄下 PU 的名稱、天線場型、所在位置、功率、使用頻帶與高度，也在報告中定義了不同種類 SU 的功率限制。相較於 FCC 對於 CR AP(Access Point)的規範，數位電視基地台的功率覆蓋範圍遠超於一個 CR AP 的功率範圍，為了達成 FCC 的規範，我們使用合作式頻譜估測，藉由分散式的功率感測，克服環境中的訊號遮蔽效應，重建出 PU 的功率分布圖。

為達上述預期目標，我們提出了感知雲端網路(Cognitive Radio Cloud Network，CRCN)在電視空白頻譜上的應用。在這樣的架構下，我們不但可使用雲端平台在骨幹網路節點的特性，利用各 CR AP 收集 SUs 的功率量測資料，以雲端平台作為頻譜偵測的資料彙整中心以及合作式演算法的運算中心，並作為 CR 頻譜資源分配與排程的控制中心，實現一個高度彈性的中央控制式通訊網路架構。在 CR 的環境裡，SUs 的數量將隨著時間和地區而改變，這樣的性質使得選擇雲端平台作為資料匯流中心能有更大的助益，雲端平台將根據偵測回報的數量，動態調整使用的運算資源，估測 PUs 的資訊，一直到分配通訊資源，提供完整的通訊服務，並針對在空間以及時間上量測數量的變化，提供有效率的平行化處理，使得 CRCN 能適用於跨區域網路(Regional Area Network, RAN)的通訊服務。

二、 研究目的



圖一、CRCN 的應用環境示意圖。

隨著無線頻寬需求的上升，無線區域網路的存取點(Access Point)將如同路燈般的公共設施，遍布在城市中的以供大眾使用。然而，建立一個如此龐大的無線網路系統，需求更多頻譜資源以及一個共同的控管系統。感知無線電在保護優先權使用者的權限下，提供了更為廣闊的頻譜使用，雲端運算的技術則提供了建立此控管系統的可能，圖一就是我們所設想的 CRCN 應用環境示意圖。

為了測試 CR 與雲端結合後的效能表現，我們提出了 CRCN 的架構。在 CRCN 的架構中，我們設計了 MAC 層的通訊協定、雲端的平行化架構、合作式頻譜估測以及排程演算法，讓 CR AP 及其服務的 SUs 在 CRCN 的架構下進行頻譜的偵測以及資料的傳輸。在 MAC 層協定的設計上，我們實作一個簡單但仍能呈現 CR 概念的通訊協定。該通訊協定初步以 FD-TDMA 為基礎，並使用兩個不同的頻道，分別用來傳送資料與控制訊息。為了完成合作式頻譜的偵測，我們要求使用者定期回傳資料頻道的功率偵測數據，該數據包含偵測時間、偵測到的功率值以及其所在位置，雲端平台再藉由骨幹網路上的 CR APs 蒐集這些資訊來做頻譜估測。

關於頻譜估測演算法，我們提出一個以貝氏稀疏演算法 (Sparse Bayesian Learning algorithm) 為基底的合作式頻譜感測演算法（Cooperative spectrum sensing），這個演算法會利用來自於 SUs 所搜集的資訊，計算出 PUs 所在位置以及其功率分布圖。根據 PUs 的功率分布以及 SUs 所在位置，我們可

以計算 SUs 的排程資訊，在不影響 PUs 的前提下，安排 SUs 在適當的資料頻道進行資料傳送或接收。

當考慮到範圍較大的區域時、可能包括數個 CR APs 時，基於以下三點考量，由雲端來做管理是比較可行的。

(1) 相對於一個 AP 所能夠提供的計算能力，雲端平台具較強大的計算能力，也可以隨著需求而有較具彈性的擴充。

(2) FCC 要求 CR 營運商對 PUs 所用的頻帶以及所在位置建置一資料庫，此資料庫的容量需隨著網路範圍增大以及使用者數目的增加適度地擴充及發展；此外，雲端平台具有適應不同運算負載的特性，所以此工作也適合由雲端來執行。

(3) CR APs 之間的資料交換，若 PU 的覆蓋範圍包含了數個 CR APs，此時 PU 的發送功率和覆蓋範圍無法經由單一 CR AP 所收到的感測數據估測，一定得經由不同 CR APs 的資料交換後才有可能實現。而想要在大量 CR APs 之間進行資料交換時，一可行的方式就是所有的 CR APs 都連上雲端，由雲端來做資料整合和交換，並且有權限來管理所有的 CR APs。

為達成前述研究目的，此計畫執行時分三個分項進行：

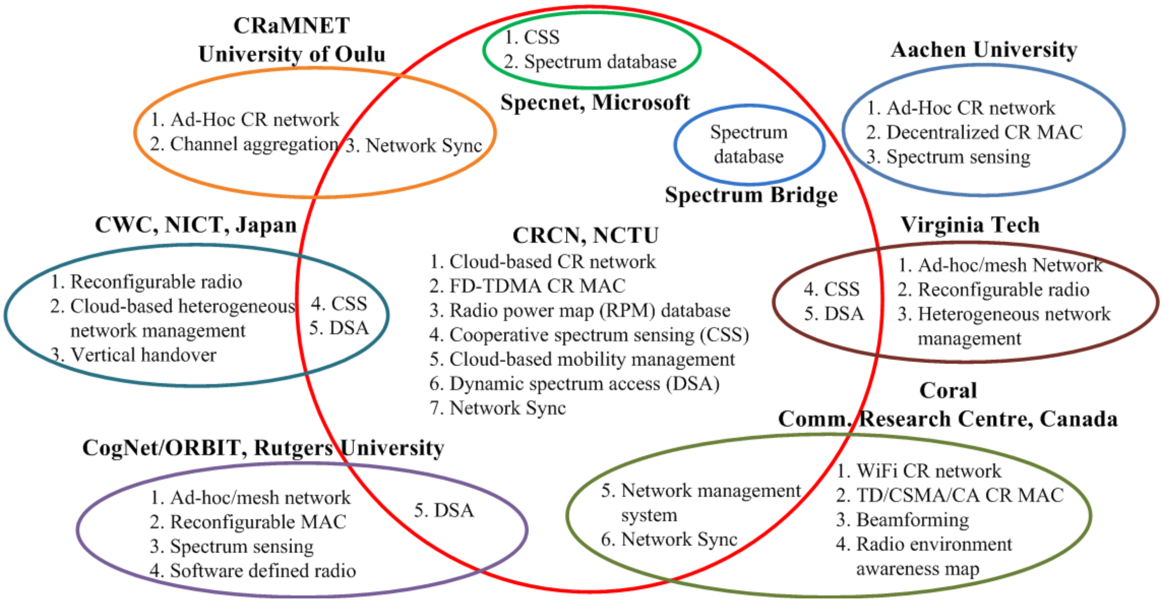
分項一：CRCN 實體層之感測與實作

分項二：CRCN 媒體層協定提案及實作

分項三：雲端運算平台之改良

三、 文獻探討

1. 世界各研究單位感知無線電平台原型之調查



圖二、世界各研究單位感知無線電平台原型之功能比較

我們調查並比較世界各研究團隊之感知無線電平台，並且依功能整理如圖二所示。其中專注在資料庫建立的有 SpecNet 以及 Spectrum Bridge。SpecNet 為微軟印度研究院的提出的研究平台，在其所提出研究中，敘述了建立估測 PU 的必要性，利用簡單的動態估測機制偵測 TV 基地台並建立資料庫。Spectrum Bridge 為美國的頻譜資料庫管理公司，該公司擁有全美國的靜態頻譜，並符合 FCC 對於頻譜資料庫的規定。目前 Spectrum Bridge 公司已經和 Google、MicroSoft 等公司合作，在美國建立並提供小範圍之電視空白頻帶網路。

CRaMNet (Cognitive Radio Assisted Mobile Ad Hoc Network)為 Oulu 大學所提出的平台原型[3]，其平台利用多台 WARP 實作 Ad-Hoc 感知無線電網路，並實作了網路同步與通道聚合(channel aggregation)，使得 channel 中有部分頻寬被佔住時 SU 可切換中心頻率與調變，使用剩下的頻寬。在頻譜感測演算法上，CRaMNet 使用的機制是在每一台 WARP 上，各自感測該頻譜是否有其他使用者在使用，而沒有建立全域的 PU 資訊。CRaMNet 並利用其所實作的網路原型，實作了第一個經由感知無線電網路傳送網路語音服務(Voice over IP，VoIP)。

CWC (Cognitive Wireless Cloud)為 NICT (National Institution of Information and Communication Technology) 所提出的感知無線電網路原型 [4][5]，也是目前所有原型中有利用到雲端架構的網路原型，然而和我們所提出的網路原型不同，CWC 的雲端是用來最佳化在 CWC 中的無線路徑。換言之，CWC 提供了 Reconfigurable 的網路模型，終端使用者可以藉由重新編譯無線電路，使用不同的無線網路，而 CWC 中的雲端就是用來處理無線路徑的使用問題。在 CWC 的實作中，感知無線電也是 CWC 眾多的無線介面之一，而 PU 資訊是藉由各個 CR 基地台合作感測所形成，SU 若是要存取此網路，

則必須透過 CR 基地台。

Rutgers 大學所提出的 CogNet [6]和 Virginia 理工學院所出的感知無線網路原型[7]都是基於 Ad-Hoc 的模式，建立網狀網路。他們共同的特點是對 Reconfigurable 功能的支援，亦即在他們提出的感知無線電原型下，使用者能藉由重新編譯硬體而使用不同的無線通道資源。兩者主要的差異在於 CogNet 團隊所發展的平台藉由各節點感測的結果決定該結點在此無線通道是否能夠使用，而 Virginia 理工學院所提出的感知無線電原型則是藉由合作式的頻譜估測，估測 PU 是否存在，並且保護 PU 的使用權力。Aachen 大學所提出的原型也是建立在 Ad-Hoc 模式上，他們實作了頻譜感測機制，並且發展了一個非集中式的 MAC 層演算法。

Coral (COgnitive RAdio Learning Platform)為加拿大通訊研究中心 (Communication Research Centre, Canada)所提出的原型[8][9]。此原型藉由改變 wifi 網卡的驅動程式，使得終端裝置能夠在 wifi 的頻帶上進行感知無線電的運作。在感知 PU 方面，該平台利用合作式頻譜估測的技術蒐集各個節點的感測資訊，估測出 PU 的資訊，並且實作了 beam forming 的功能增加頻譜的使用效能。

2. CRCN 實體層之感測與實作

為了保護 PUs 的優先使用權，FCC 規定當 PUs 的訊號功率大於或等於-116dBm 這個臨界值的情況下，CR APs 要避免使用這個通道與左右相臨的子頻道。換言之，SUs 至少要能分辨出功率值等於偵測臨界值的 PUs 信號。但是偵測此臨界值對電路來說是一項嚴格的考驗；以數位電視為例，通道 6MHz 的雜訊值經過計算約為-106.2dBm，然而所需偵測的臨界值為-116dBm，訊雜比為-9.8dB；若是再考量電路的雜訊(一般約 10dB)，基頻的訊號處理必須要能在-20dBm 的訊雜比下分辨出是否有 PUs 的信號存在，這對於單一電路來說有實作上的困難。因此，我們必須藉由合作式的頻譜估測演算法來偵測 PUs 的訊號。

一般的分散式頻譜估測技術演算法，並無考慮 SUs 身處的地理位置；忽略此項位置資訊的演算法相當於用一平均的估測值來代表整體頻譜在不同區域的強度表現，無法呈現不同區域之強度差異。然而 FCC 在 2008 年 11 月針對 CR 發布的五大方針裡[10]，亦認為 SUs 應回報自己的地理資訊來幫助 CR APs 作頻譜估測。故我們希望能發展一合作式頻譜估測演算法，結合所有 SUs 回傳的訊息的演算法，以重建不同區域之頻譜功率強度。[11][12]是目前已考慮結合地理資訊與估測結果的方法，然而當中估測 PUs 位置的方式是採用傳統的 Angle-of-arrival (AOA) 或 Time-of-arrival (DOA)，必須依賴 SUs 與 PUs 之間訊息的交換。此訊息的交換在 CR 系統的設計當中是不容許的；根據標準定義，SUs 理應在完全不打擾 PUs 的情況下，精準的估測 PUs 是否正在使用。故發展一個能單純依據 SUs 訊號感測功率及地理位置做估測的演算法，為我們的重點研究項目之一。

3. CRCN 媒體層協定提案及實作

為了保障 PUs 對頻道的使用權，FCC 規定當 PUs 要使用資料頻道時，而此頻道被 SUs 借用時，SUs 須在兩秒鐘內將資料頻道歸還於 PUs；此外，FCC 亦規範 SUs 需偵測資料頻道 2ms，以確認是否

已有 PUs 或其它 SUs 使用此資料頻道。為達 FCC 所定義之要求，頻道資源排程與存取控制演算法為實現 CR 技術不可或缺的要項。

[13-17]均是針對 CR 網路所設計的頻譜資源排程與控制演算法。相關文獻可依有無使用 SUs 專用的控制頻道而分為兩大類。有使用控制頻道之頻道資源排程演算法均讓 SUs 在控制頻道上溝通所要借用的資料頻道，藉由同步偵測資料頻道的使用狀況，找到一條對傳送端與接收端均為閒置的資料頻道。另一類不使用控制頻道的演算法則是讓傳送端與接收端各自有一固定跳頻的頻道序列，透過設計頻道序列讓傳送端與接收端可以在一特定的資料頻道上碰面，進行資料傳送。而為避免 PUs 在需要使用資料頻道時等待時間超過 FCC 規範，大多數演算法皆保守地讓 SUs 在資料頻道上傳送一個資料訊框後釋出頻道使用權。因此，[13-17]所提的頻道資源排程與存取控制演算法大抵有可用頻道搜尋時間過長以及頻道利用率過低的共通缺點。再者，[13-17]的演算法僅適用於小範圍之 CR 隨意網路。SUs 的資料傳送為 single hop，沒有 CR AP 可以協助傳送 multihop 的資料，故能真正實作的可能性低。藉由合作式頻譜估測演算法所獲得的 PUs 資訊、雲端平台的運算能力、再加上 SUs 提供的自身地理位置，設計一能適用於雲端平台、具彈性擴充計算模組功能、能支援大範圍 multihop 的通訊模式的中央控制式頻道資源排程演算法是我們的重點研究項目之一。

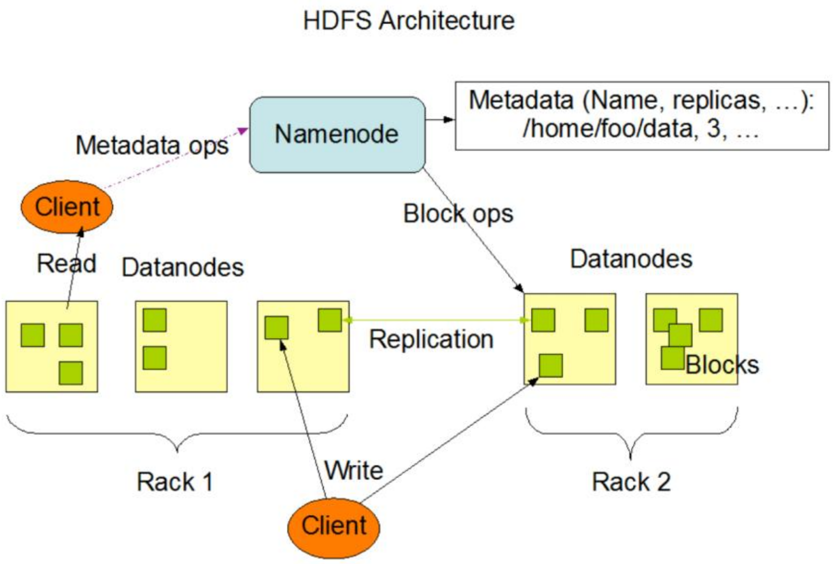
4. 雲端運算平台之改良

(1) Hadoop

Hadoop [18] 是由開源的 Apache 計劃發展出的雲端運算框架，並由 Yahoo 資助、開發與使用。Hadoop 計劃是由許多子計劃組成，如 HDFS (Hadoop Distributed File System)、MapReduce、HBase 等。

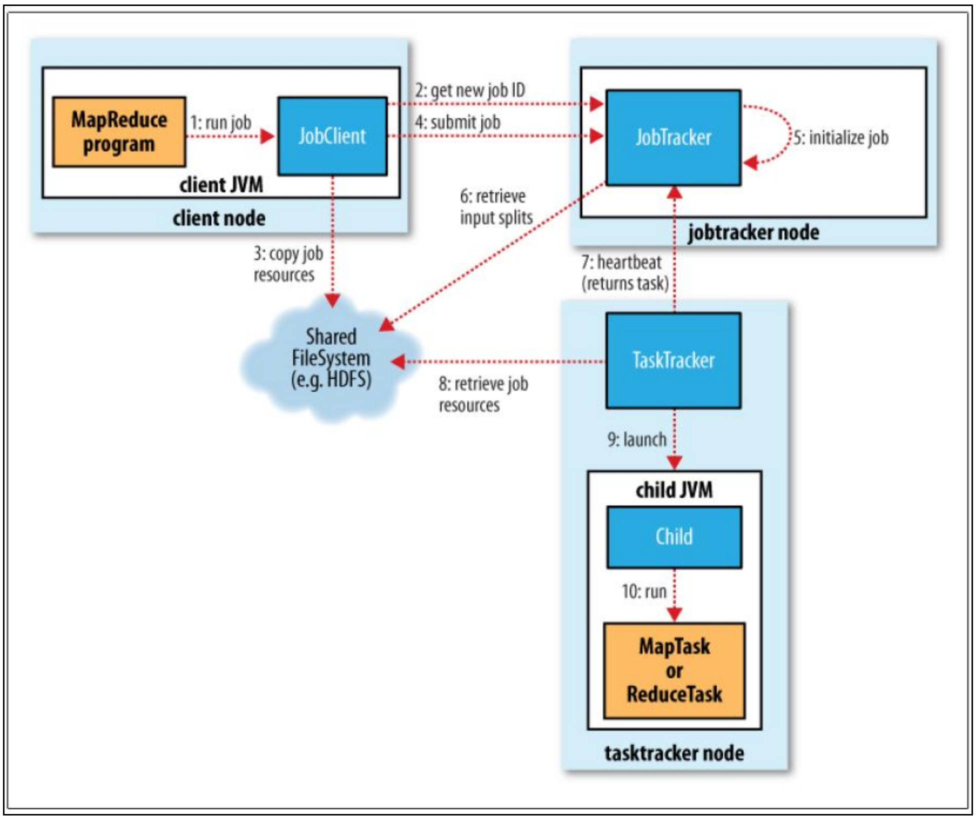
Hadoop 採用主從式架構一個 Hadoop 文件系統，由一個目錄節點和數個資料節點組成，其分散式檔案系統稱為 HDFS，如圖三所示。HDFS 的特色是有高度的容錯性以適用於規模量產的硬體運算組件，並為應用程式提供了高速的資料存取服務，適合需要處理大量資料的應用程式。一個 HDFS 叢集包含了一個 NameNode，責檔案名的維護管理，也是用戶端訪問文件的入口。檔名的維護包括文件和目錄的創建、刪除、重命名等。同時管理資料塊和資料節點的映射關係，用戶端需要訪問目錄節點才能知道一個文件的所有資料塊都保存在哪些資料節點上。

另外還有一定數量的 DataNode 負責管理與之相連節點上的儲存系統，DataNode 一般就是叢集裡面的一台機器，負責資料的儲存和讀取。在寫入時，由目錄節點分配資料塊儲存，然後用戶端直接寫到對應的資料節點。在讀取時，當用戶端從目錄節點獲得資料塊的映射關係後，就會直接到對應的資料節點讀取資料。資料節點也要根據目錄節點的命令創建、刪除資料塊，和冗餘複製。



圖三、HDFS 架構示意圖(節錄自[18])。

Hadoop 程式開發框架為 MapReduce，讓開發者能夠開發平行處理大量資料的應用程式。一個 MapReduce 的工作在執行的時候分成兩個部分，Mapper 先將輸入的資料切割成不同的小塊進行個別的處理，處理後經過 Reducer 透過重新排程(shuffle)這個動作，將 Mapper 處理過後的資料集結過來，整理成最後輸出結果。



圖四、MapReduce 架構圖[19]

Job-Tracker：控制與分配整個 job 的執行，將 job 切割成許多小的 task，並要求 Task-tracker 來執行，在 MapReduce 中擔任 Master 的角色。

Task-Tracker：在 MapReduce 運算中擔任 Slave 的角色，利用機器上的資源執行 Job-Tracker 分配下來的 task，而 task 是一個 job 的基本執行單位。

Hadoop 工作執行流程：當一個工作被 Job client 提交至 Job-Tracker 執行 ，一直到執行完畢，共被分為四個階段：

- 初始設定階段：當Job-Tracker收到新的job時會發送一個setup task給還有空位(free slot)的Task-Tracker來做初始化環境的工作，等到setup-task結束後，job會切換到RUNNING狀態，之後才會開始執行此job真正的工作內容。
- Map階段：等到setup task執行完畢之後，Job-Tracker開始將切割好的map tasks分配給Task-Tracker執行，Task-Tracker會持續傳送一個"Heartbeat"訊息給Job-Tracker，訊息中包含了該Task-tracker仍然還存活著的訊息以及該task-tracker目前是否有空位可以用來執行更多的task，如果回報仍然有空位來執行 task，Job-Tracer會使用hearbeat回傳值來指派新的task給Task-Tracker來執行。
- Reduce階段：當有Map task執行完後，Reduce task也會開始由Job-tracker啟動。
- 清除階段：當工作完成後，Job-tracker會啟動一個cleanup task把之前用到的環境清除，例如工作資料夾、環境變數等等，等到cleanup task完成後，此工作的執行結果就會確定，SUCCESS、FAIL或是KILLED。

(2) Windows Azure

Windows Azure 平台是一種「Platform as a Service (PaaS)」產品。提供一種平台來執行 Windows 應用程式並將資料儲存至雲端。這些應用程式可以是既有應用程式，或者是為了要在 Windows Azure 上執行而專門設計的全新應用程式。開發人員可以使用熟悉的工具，例如 Visual Studio 2010，為 Windows Azure 平台開發各種應用程式。Windows Azure 雲端平台提供包含 Windows Azure、Live Services、SQL Azure、Microsoft .NET Services、Microsoft SharePoint Services 以及 Dynamics CRM Services，如圖五所示。

圖五、Windows Azure 的應用與平台架構 (節錄自[20])。

Windows Azure 的作業系統是專為在資料中心所開發的一種特殊 Windows Server 作業系統版本，並且定時和 Windows Azure Fabric Controller 進行溝通，接收指令以及回傳執行狀態資料等等，作為 Azure 服務平台的開發、服務代管及服務管理環境。Windows Azure 提供開發人員載入、管理與執行網際網路上的雲端運算應用程式，並提出了兩種不同的程式設計模型，提供不同的服務：Worker Role 提供背景運算服務，Web Role 用以連結使用者以及背景程式。Windows Azure 提供了不同的虛擬主機等級，對應到不同的 CPU 核心，記憶體以及硬碟空間並提供多核心的虛擬主機，至於其他 Windows Azure 所提供的服務則建立在其架構上，提供使用者雲端上應用程式開發的支援。

四、 研究方法

1. CRCN 實體層之感測與實作

CR 和傳統的無線通訊相比較，有一個重要的變化即是區分出 PUs 以及 SUs 的存取。SUs 為了確保 PUs 的權利，必須經由偵測等方式，估測出 PUs 的訊號源位置及訊號涵蓋範圍。由於保護 PUs 的權利是如此重要，FCC 定義了相當嚴謹的標準，希望控制對 PUs 的影響。然而，在無線通訊環境中，遮蔽效應以及雜訊影響了對於 PUs 訊號的估測。因此，我們提出了合作式頻譜估測的概念，藉由蒐集分散的各點偵測結果，還原出更準確的估測結果，有效的對抗環境中的遮蔽效應及雜訊以嘗試達到 FCC 的要求。

特別在電視空白頻譜的環境中，由於對於 CR AP 功率上的限制以及數位電視基地台的功率範圍過於龐大，我們必須集合跨區域的感測結果才能夠估計出 PUs 的位置以及功率覆蓋範圍，這樣的特性使我們考慮運用雲端平台做為演算法之運算執行中心，雲端平台必須能夠彙整來自於 SUs 的感測資料，藉由一個統一的資料匯流中心建立一個中央控制式的通訊系統，確保 PUs 的使用通訊並隨著時間以及地區調整運算量。

我們所提出的合作式頻譜估測演算法是基於貝氏稀疏演算法來還原 PUs 的功率傳播圖。比起傳統壓縮式偵測(Compressive sensing)演算法，此方法能夠利用更少的取樣點還原出 PUs 的功率傳播圖，並且同時估測出 PUs 的個數、位置、功率衰減係數以及環境中雜訊的變異量。

然而，此種頻譜估測演算法的複雜度將隨著資料量成三次方的倍率升高，因此在本計畫中，我們將此合作式頻譜估測演算法實作於雲端平台，並且嘗試利用不同的平行化技術來改善其計算速度。在 Windows Azure 的架構下，我們參考 MapReduce 的平行化方式，藉由地理位置以及頻譜上感測功率相互獨立的特性，平行將資訊分散到不同的 VMs 來進行計算，並利用 Windows Azure 對於多執行緒平行化架構的支援，分層級的減少頻譜估測所需要的時間。在通訊系統中，即時(Real Time)一直是一個重要的議題，尤其是在 CR 環境下，越即時的頻譜估測能夠提供 PUs 的使用權越好的保護，因此我們希望能在不影響估測誤差的前提下，利用雲端平台的架構，增進整體的效能，並利用雲端平台的彈性，隨著感測數據的多寡，動態的調整使用的 VMs 數量，以有效率的支援 CRCN 的需求。相關的成果，請參考附錄一。

在 CRCN 系統實作上，我們利用 WARP 開發平台作為 CR AP。WARP 是一個以場域可編程閘陣列(FPGA)與處理器(PowerPC)為開發核心之開放式軟體無線電(Software define Radio)發展平台。在此平台上，一方面我們將 ISM band 之頻率鍵移調變(FSK)收發模組 HMTR 擴充至 WARP 平台上，另一方面我們也利用 GP IO 連接 CC1111，使得 WARP 平台能與 SUs 進行簡單的通訊。

為了將 SUs 的資料透過骨幹網路傳送到雲端，在 CR AP 上必須要有符合網路的通訊協定如 TCP/IP，才有辦法與網際網路(Internet)溝通。因此我們在 WARP 開發平台的 PowerPC 上移植了一個簡單的 Linux 作業系統。Linux 作業系統是一個開放式平台的作業系統，許多的網路協定都已經包含

在其中，因此我們不需要在額外客製化一些通訊協定在 WARP 上，就能夠利用標準的網路程式與雲端平台溝通，將 SU 感測到的資訊回報透過骨幹網路傳到雲端中心進行頻譜估測，或者將 SU 的資料藉由網際網路傳送給另一個 SU。此外，在 WARP 平台上建立 Linux 環境也有助於 CR AP 控制程式的發展，由於所有的無線通訊模組 (HM-TR 與 CC1111) 都將變成 Linux 系統下的通訊接口，因此我們可以在 Linux 環境下利用現有的 C 函式庫進行程式開發。

2. CRCN 媒體層協定提案及實作

在CR MAC的設計中，我們考慮到PU的特性，對每條不同的頻帶套入機率模型做成功傳送機率的評估，AP_{CR}會根據每個非執照使用者所提出之頻寬要求，針對每個非執照使用者對於每條頻帶會計算三種數值，分別為成功在該頻道上通訊的機率(***P***)、感測頻道時間 (***W***)和transmission quota (***Q***)。因此，感知無線電的存取點(AP)會告知非執照使用者其一個N×3矩陣，記錄在不同頻道上計算出的上述三數值。若非執照使用者C (傳送端) 和非執照使用者B (接收端) 欲進行通訊，非執照使用者C會在控制頻道上採用CSMA/CA機制通知感知無線電的存取點 (AP) 跳至成功傳送機率最大之資料頻道i，接著傳送與資料頻道i相對應的$\boldsymbol{Q_i}$欄位之最多可傳的封包個數。若傳送$\boldsymbol{Q_i}$個封包後之後非執照使用者C仍有資料要傳，又或者在感測頻道時間結束前偵測出資料頻道i已被使用，非執照使用者C與感知無線電的存取點(AP)會跳至第二高的機率值之資料頻道，繼續上述之過程，若所有N個資料頻道皆被使用，非執照使用者C會回到控制頻道重新進行CSMA/CA機制，直到傳送完所有資料，在不影響PU的情況下，重覆利用頻道資源。相關的成果，請參考附錄二。

透過SUs週期性地回報他們的地理位置以及在每一條資料頻道上所量測到的RSSI值，雲端運算平台在執行合作式頻譜估測演算法後，可以重建出PUs的地理位置以及功率分布圖。當SUs需要使用頻道傳送資料時，透過控制頻道告知CR APs它們的需求。CR APs 進一步將收集到的要求傳送到雲端平台，由雲端平台執行頻道資源排程演算法，在不影響PUs情況下，以達到系統最大效能為前提，分配可用頻道資源。

為了能於雲端平台實作合作式頻譜估測與頻道資源排程演算法且實測頻道排程演算法的效能，我們實作一套以FD-TDMA為主的頻道存取控制機制。我們以CR AP為媒體層協定管理主軸，將整體運行分為四個部分：(詳細的流程請參考附錄三)

(1) Initialization：CR APs、SUs、SDs 的註冊與時間同步控制。

(2) Sensing report collection：SDs 在 quiet period (QP) 感測頻帶並透過 CR AP polling 來回報。

(3) Data channel coordination，由 CR AP 以演算法排程結果來協調 SUs 使用頻帶的情況。

(4) Mobility management，透過雲端與 CR APs 來管理 SUs 的 mobility。

並依提出的媒體層協定對於 SU 提供了四個主要的功能：

(1) 同一個 CR AP 網域內的資料傳輸，

(2) 不同 CR AP 網域間的資料傳輸，

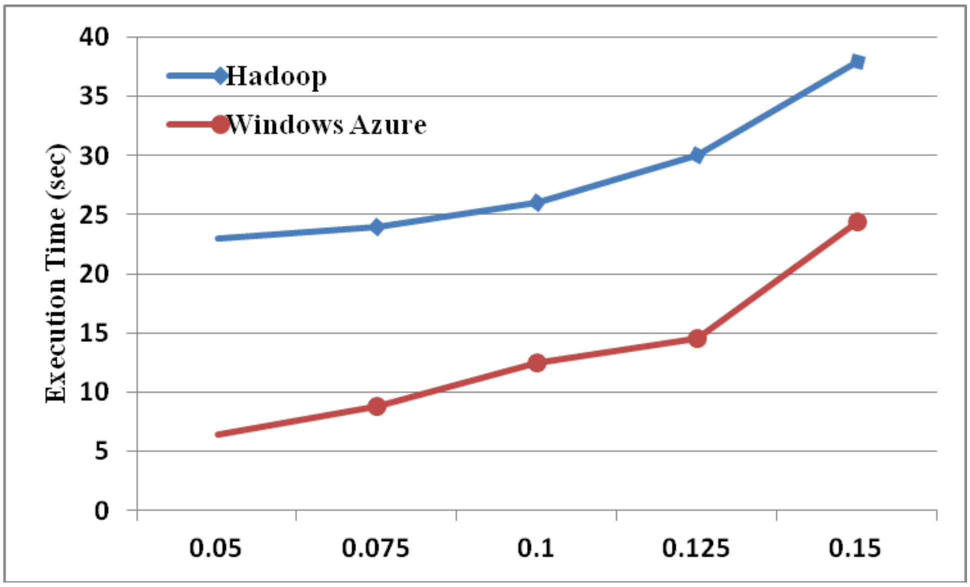
(3) SUs 從一個 CR AP 移動到另一個 CR AP 的換手機制，

(4) 當 SU 在移動中或因其他因素失去與 CR AP 的通訊時，將來自其他使用者的資料暫存在雲端保持資料不遺失的機制。

透過上述的四個功能，SU 可以自由移動到任意 AP 的範圍內來傳送使用者資料給任意網域的 SU，並且能在 PU 出現時，透過資料暫存來避免資料的遺失。而根據提出的媒體層協定實作，我們探討機制中對於 PU 出現時，雲端的頻譜估測演算法在所實作出媒體層協定中的反應時間，其中透過量測 PU 實際出現時間與 AP 透過雲端得知頻譜估測與排程演算法結果的時間為差距來計算出退出使用的頻帶所需花費時間。並且量測透過資料暫存機制，對於使用者傳送資料的延遲所需的時間。透過上述兩個數據，我們可以分析所提出的媒體層協定提案對於 PU 使用頻帶的干擾，與對於頻譜的使用效率。

3. 雲端運算平台之改良

期中報告時我們發現了 Hadoop 平台執行頻譜估測演算法所需要的執行時間大於 Windows Azure 平台所需要的執行時間的問題（如圖六），並推測這是因為 Hadoop 平台原本的設計導致花了許多時間在工作的分配管理，在期中報告之後，為了解決 Hadoop 執行時間的問題，我們針對 Hadoop 平台的架構做了更深入的研究與了解，終於發現問題的原因，並加以修改，最後成功的降低了 Hadoop 平台所需要的執行時間，提高了使用 Hadoop 平台做為 CRCN 後端計算的可行性。以下會介紹造成執行時間過長的主要原因，以及我們做修改的地方：



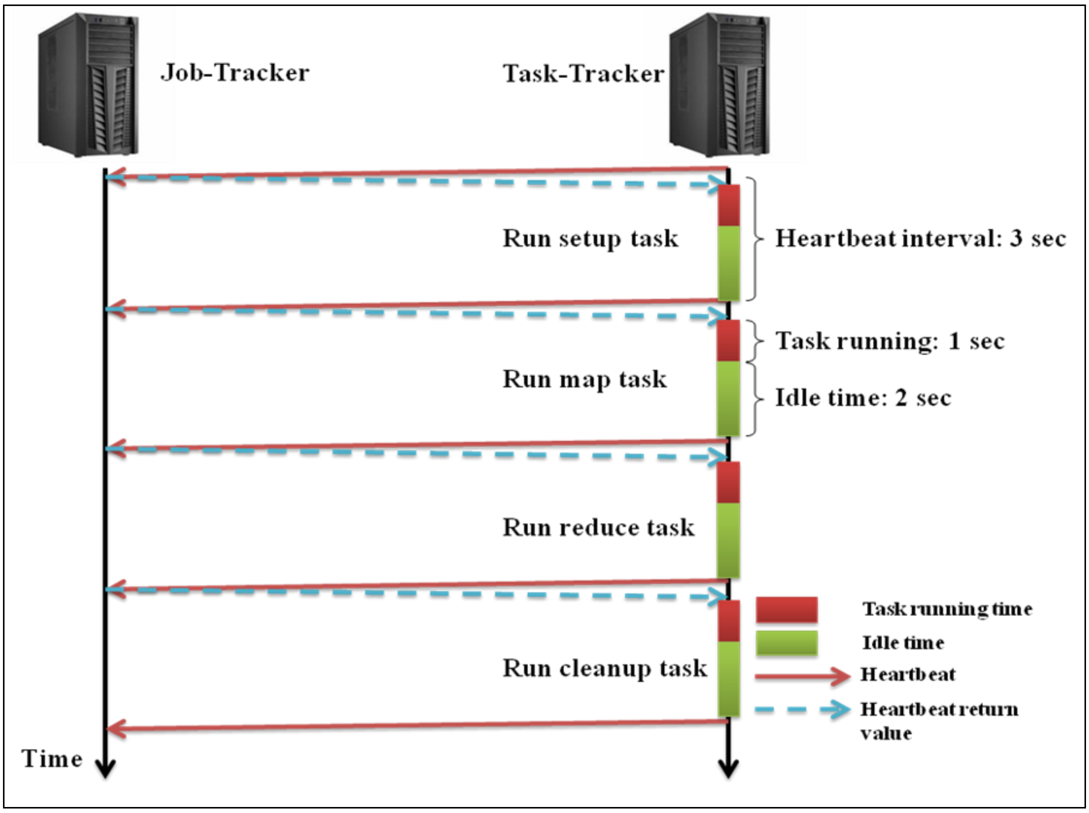
圖六、在不同雲端平台所量測的執行時間

造成執行時間過長的主要原因有三點：Heartbeat 傳送的時間間隔，Reducer 的睡眠時間，以及 Job

Commit 的睡眠時間。以下將一一說明。

(1) Heartbeat 傳送的時間間隔

讓我們以一個只包含一個 map task 與一個 reduce task 的 job 的執行流程來說明 heartbeat 對整個 job 執行時間的影響(如下圖七)：



圖七、Hadoop 工作執行流程圖

(a) 當 Job-Tracker 收到 job 的執行要求後，會產生一個 setup task 並由 heartbeat return value 來要求一個 Task-Tracker 來執行。

(b) 當 Task-Tracker 執行，並且完畢 setup task 之後，會利用執行完後下一個 heartbeat 訊息來回報給 Job-Tracker。

(c) Job-Tracker 在收到 setup task 完成的訊息之後，會馬上透過 heartbeat return value 要求 Task-Tracker 來執行 map task。

(d) 同樣的，當 Task-Tracker 完成 map task 之後，會用完成後的下一個 heartbeat 來回報給 Job-Tracker。

(e) Job-Tracker 在收到 map task 完成的訊息之後，會馬上透過 heartbeat return value 要求 Task-Tracker 來執行 reduce task。

(f) 當 Task-Tracker 完成 reduce task 之後，會用完成後的下一個 heartbeat 來回報給 Job-Tracker。

(g) 等收到 reduce task 完成的訊息後，Job-Tracker 最後會再透過 heartbeat return value 發出一個 clean up task 的要求給 Task-Tracker。

(h) Task-Tracker 完成 clean up task 之後，會用完成後的下一個 heartbeat 來回報給 Job-Tracker。

(i) 最後，Job-Tracker 收到 clean up task 完成後回報狀態來確定整個 job 已經執行完畢。

可以從以上的工作執行流程發現到，Job-Tracker 只有在收到 Task-Tracker 的 heartbeat 訊息並確定 Task-Tracker 有可以執行 task 的空間之後，才能用 heartbeat return value 來要求 Task-Tracker 來執行 task，同樣的，Task-Tracker 只能透過定時發送的 heartbeat 訊息來向 Job-tracker 報告 task 的執行進度或是 task 的執行結果，如此會發生一個問題，當一個 task 的執行時間 (假設為 1 秒鐘)小於預設的 heartbeat 傳送的時間間隔 (3 秒鐘)，當 task 迅速的執行完畢後，Job-tracker 必須要等到下一個 heartbeat 才能夠知道，在 task 執行完畢與回報 task 執行結果的中間，Task-Tracker 是處在一種閒置的狀態，如圖所示，紅色方塊代表 task 的執行時間，綠色的方塊代表 Task-Tracker 閒置的時間，同時，整個 job 的執行流程也被停了下來，如圖中的工作流程，總共浪費了 8 秒鐘在等待上，故當有人想要執行須要及時性的工作，如本計畫中的 SS 演算法，這八秒鐘是非常大的延遲。

為了更清楚的看到 heartbeat 時間間隔對整個 job 執行時間的影響，我們利用在 Hadoop 平台上執行 PiEstimator (一個 Hadoop 內部的範例)的執行結果來呈現。PiEstimator 是利用蒙地卡羅演算法 [22] 來估計 Pi 值，當 PiEstimator 執行時，它會發起多個 map task 來執行蒙地卡羅演算法，並用一個 reduce task 來收集得到的結果。PiEstimator 有兩個參數，第一個參數是使用者想要執行幾個 map task，第二個參數是每個 map task 的計算次數，當我們將計算次數設定的很小的時候，每個 map task 的執行時間也會變得很小，這讓我們可以直接的看到 heartbeat 時間間隔的影響。下表呈現了預設 heartbeat 時間間隔(3 秒)與修改過的 heartbeat 時間間隔(0.05 秒)分別執行 PiEstimator 所需要的時間(表一)，我們可以看到修改過的 heartbeat 時間間隔可以降低約平均 11 秒的執行時間，另外執行工作的機器規格請見表二。

表一、PiEstimator 在不同 heartbeat 時間間隔的執行時間

	預設 heartbeat 時間間隔 3 sec	修改過的 heartbeat 時間間隔 0.05 sec
Pi 1 100	22.362 sec	10.365 sec
Pi 4 100	22.430 sec	10.393 sec
Pi 8 100	22.433 sec	10.721 sec
Pi 16 100	22.738 sec	11.396 sec
Pi 32 100	26.841 sec	11.398 sec

表二、機器規格

Worker number	CPU	Memory	Disk space	Mapper Reducer max number
2	I7 2600	16 GB	1TB	8

(2) Reducer 的睡眠時間

我們也發現當 Reduce task 被執行起來的時候，Reduce task 會開始向各個 Task-Tracker 要求 map 已經 task 先處理過的中間資料，但在工作執行的過程中，可能會有某些時刻是沒有中間資料可以給 reduce task 的，此時，reduce task 就會進入睡眠 5 秒鐘，之後再起來檢查是否有新的資料可以取得，這樣的設計是因為 Hadoop 當初是設計給非常大型的 job 來使用，每個 map task 的執行時間可能達到數分鐘甚至數十分鐘，因此這個設計是非常合理的，可以將執行 reduce task 的資源釋放，讓 map task 能夠更快的完成。但是當一個 job 能夠被很完善的平行分配，讓 map task 的執行時間能夠降低到非常小，相較起來預設的 5 秒鐘睡眠時間則是一個相對大的浪費，特別是針對須要及時反應的應用。同樣的，我們用 PiEstimator 的執行時間來觀察 Reduce task 的睡眠時間對執行時間造成的影響，如表三所示，

表三、不同 Reduce task 睡眠時間對 PiEstimator 執行時間的影響

	預設 Reduce 睡眠時間 5 秒鐘	修改過的 Reduce 睡眠時間 0.05 秒鐘
Pi 1 100	22.362 sec	16.716 sec
Pi 4 100	22.430 sec	19.731 sec
Pi 8 100	22.433 sec	20.084 sec
Pi 16 100	22.738 sec	19.409 sec
Pi 32 100	26.841 secs	21.428 sec

(3) Job Commit 的睡眠時間

最後我們也發現，在 hadoop job 執行到最後會呼叫一個 done function，呼叫該 function 之後，Task-Tracker 就會與 Job-Tracker 確認工作是否能夠結束，這時候也會進入一個 1 秒鐘的短暫睡眠，等待 Task-Tracker 透過 heartbeat 訊息來跟 Job-Tracker 確認工作的結束，由於我們已經縮小了 heartbeat 的時間間隔，在這裡我們也不需要再多等待這 1 秒鐘，從表四我們可以看到降低 Job commit 等待時間後的效果，減少了大約一秒鐘的時間。

表四、Job commit 的睡眠時間對 PiEstimator 執行時間的影響

	預設 Reduce 睡眠時間 5 秒鐘	修改過的 Reduce 睡眠時間 0.05 秒鐘
Pi 1 100	10.365 sec	9.355 sec
Pi 4 100	10.393 sec	9.362 sec
Pi 8 100	10.721 sec	9.412 sec
Pi 16 100	11.396 sec	10.395 sec
Pi 32 100	11.398 sec	10.401 sec

(4) 同時修改三種設定的影響

最後，我們將以 PiEstimator 測試同時修改三種參數的結果，如下表五，可以看到我們大幅降低了 hadoop 在計算較少的工作上的執行時間，這個結果對本計畫有正面的幫助，能夠讓計算頻譜資源的速度更快，達到更即時的頻譜資料庫更新反應。

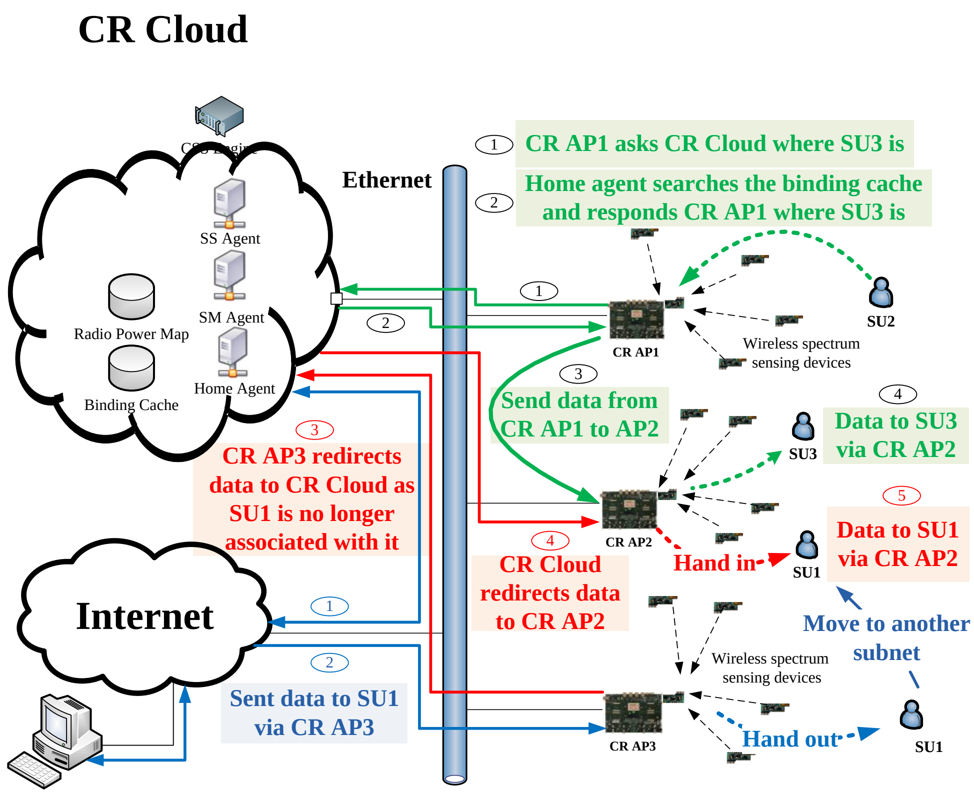
表五、同時修改 Hadoop 三個參數後的執行時間

	Hadoop 原始設定	修改過的 Hadoop
Pi 1 100	22.362 sec	4.361 sec
Pi 4 100	22.430 sec	4.354 sec
Pi 8 100	22.433 sec	4.346 sec
Pi 16 100	22.738 sec	5.382 sec
Pi 32 100	26.841 secs	6.393 sec

五、 實驗成果

1. 以 Windows Azure 為雲端平台的通訊系統實測

(1) 通訊系統硬體建置、網路架構、與雲端平台



圖八、系統架構示意圖。

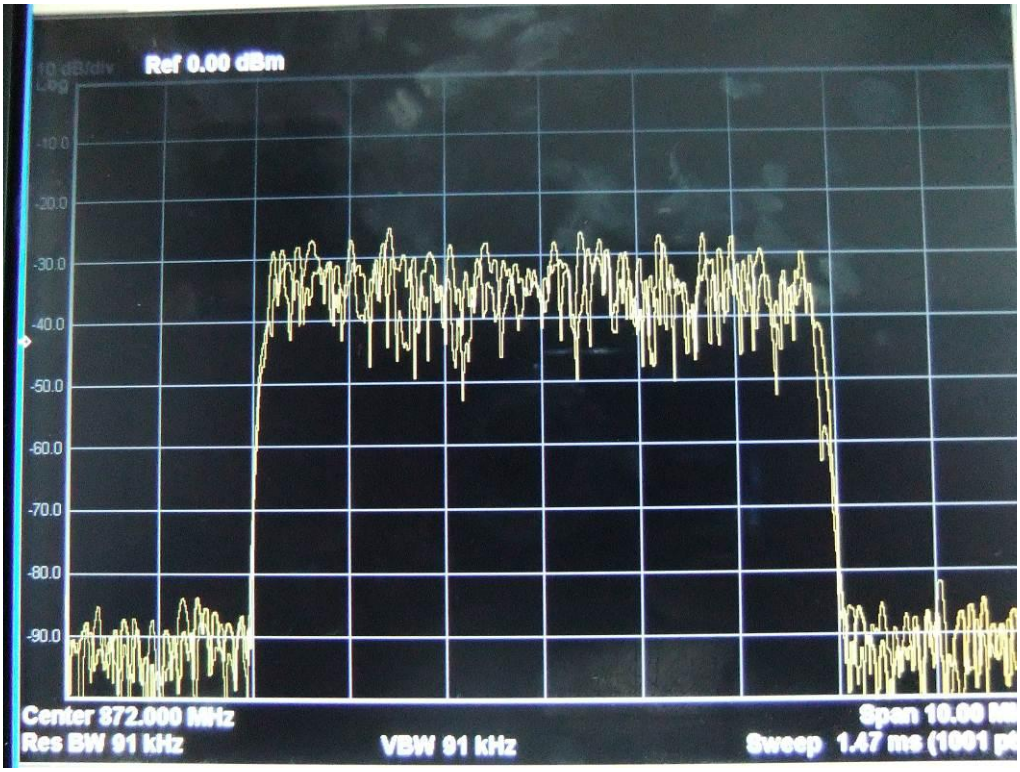
在我們建立的測試環境中（系統架構如圖八），我們使用 CC1111 與 HMTR 兩種通訊模組 (如圖九所示)。CC1111 為 SUs 作為無線通訊使用的元件；HM-TR 則作為 SDs 使用，負責感測資料頻帶上的功率，並提供資料給雲端去估計 PUs 的功率分布。另外，我們使用訊號產生器在資料頻帶上模擬 PUs 的行為，並以 WARP(Wireless Open Access Research Platform) [21]作為我們的 CR APs (如圖十所示)，CC1111 與 HM-TR 都受 WARP 管理，在我們制定的 MAC 層協定下運作。我們使用筆記型電腦接上一 CC1111 通訊模組作為我們的 SUs (如圖十一所示)。

圖九、通訊模組實照。

圖十、WARP 開發平台接上 CC1111s 及 HMTR 作為我們的 CR AP

圖十一、筆電接上 CC1111 通訊模組作為我們的 SUs

為了實際模擬在 TV WS 上的環境，在這一次實作中，我們使用基頻訊號產生器(Baseband Signal Generator)產生 DVB-T 6MHz 的數位電視基頻訊號，並利用射頻訊號產生器(Vector Signal Generator)將原本的基頻數位電視訊號移至 872MHz 也就是 Data Channel 所在的頻帶上模擬 TV WS 上的 PU (如圖十二所示)，也就是數位電視基地台的訊號。



圖十二、在頻譜分析儀上的 PU 訊號，中心頻率為 872MHz，頻寬為 6MHz

CR APs 以 WARP 作為運算的裝置，同時連接上 CC1111s 與 HMTR，HMTR 負責管理其他作為 SDs 的 HMTR；而一個 CC1111 分別控制一條頻帶上的通訊行為（目前資料頻帶與控制頻帶各由一個 CC1111 管理）。SUs 上的 CC1111 需要在控制頻帶與資料頻帶間來回切換，先在控制頻帶上加入 CR AP 的網域，並取得資料傳輸的權限後，才能切換至資料頻帶進行資料傳輸。為了讓 SDs 能收集到正確的功率資料，在資料頻帶上每個週期都有一段禁止任何資料傳輸的時間，SDs 必須在這段時間內收集 PUs 的功率資料，然後回到控制頻帶將結果送給 CR AP 轉交至雲端，進行頻譜估測。另一方面，CR AP 會定期接收雲端平台計算出的排程資訊，藉由 MAC 層協定回報給 SUs，控制 SUs 在資料頻道上的傳輸，確保 PUs 的使用權利。

我們實作一基於 FD-TDMA 之 MAC 層協定，並利用 CC1111、HMTR 兩種通訊模組、WARP 以及 Windows Azure 雲端平台，建構出一個小型的 CRCN 測試模型。在我們的 CRCN 測試模型中，我們實作了四種主要功能：(1) 同一個 CR AP 網域內的資料傳輸，(2) 不同 CR AP 網域間的資料傳輸，(3) SUs 從一個 CR AP 移動到另一個 CR AP 的換手機制，(4) 當 SU 在移動中或因其他因素失去與 CR AP 的通訊時，將來自其他使用者的資料暫存在雲端保持資料不遺失的機制。詳細功能介紹請參考我們的 demo 影片。除此之外，我們也嘗試量測在我們設計的 MAC 層通訊協定下，量測實際需要的頻譜估測時間，以及評估 SUs 對 PUs 的影響，藉以評估 CRCN 的可行性以及分析之後如何繼續改善 CRCN 的架構。

在我們設計的通訊協定中，包含了兩種不同的通訊頻帶，資料頻帶為 PUs 所擁有，SUs 只能在 PUs 沒有使用時傳送資料；控制頻帶(可為 ISM 頻帶或是其他沒有 PUs 的頻帶)，用以傳送控制訊息，

包括 SUs 感測結果的回傳以及頻道資源分配的結果。排程演算法則建立在合作式頻譜估測之上，合作式頻譜估測建立 PUs 的功率分布圖，排程演算法將判斷 PUs 所在地理位置上可能接收到的 SUs 干擾功率，在不影響 PUs 的前提之下，分配頻譜資源與 SUs。

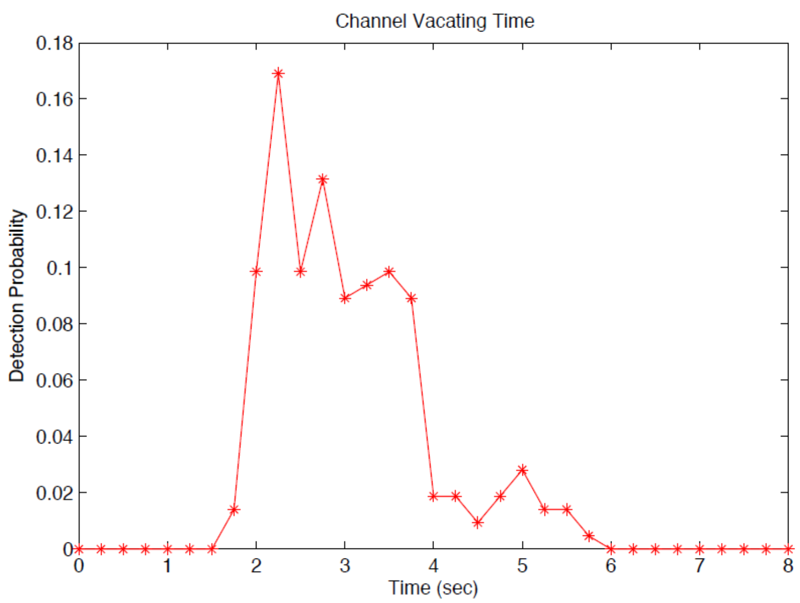
從 CR APs 蒐集的量測數據藉由網際網路更新到 SQL Azure 的 SUs 資料庫，在判斷回報量足夠後，Windows Azure 上的運算單元便執行我們所設計的合作式頻譜估測演算法，估計 PU 的功率分布圖，並將結果儲存於 PU 的資料庫。在雲端上實作的排程演算法則會利用頻譜估測的功率分布，以及從感測報告中記錄下的 SUs 的位置，計算出 PUs 的功率分布，並且根據此資訊決定頻譜資源的分配。

為了實現不同網域間的傳送，當 CR AP 收到來自使用者的資料時，會根據雲端上的使用者資訊，找到目的端使用者所在的網域，再透過 Internet 將資料傳送至目的端 CR AP 進一步轉交給目的端使用者。我們也在 CR AP 中加入定期將 SUs 剔除的機制，SUs 必須在此時間內重新發送 JOIN 訊息以保持跟 CR AP 的關聯；這個機制有利於換手機制的實制，當 SU 離開某個 CR AP 的管理範圍時，CR AP 會因為沒收到 SU 的 JOIN 訊息而將 SUs 踢除，讓 SU 能成功加入另一個 CR AP 而不產生任何衝突。

另外，由於在換手過程中，SU 被踢除到 SU 加入另一個 CR AP 會間隔一小段時間，若在這段時間內其他使用者發送訊息給該 SU，該 SU 會因為不在原 CR AP 的範圍內收不到資料導致資料流失，為了避免此狀況，當 CR AP 發現收到目的端 SU 不在自己網域內的資料時，會將資料傳送到雲端暫存，當 SU 加入另一個 CR AP 後，雲端會把暫存的資料送到該 CR AP 轉交給 SU，保護資料不因上述的狀況遺失。

(2) 實驗數據與討論

實驗分兩部分，第一部分測試頻帶的釋出時間，也就是當 PU 回到了網路中時，SU 需要多少時間才能將頻帶釋放，在這個實驗中，不同的 SDs 會附屬於兩台不同的 CR AP 上，並且在不同的時間點回傳所收集的資訊，所偵測的數據如圖十三。

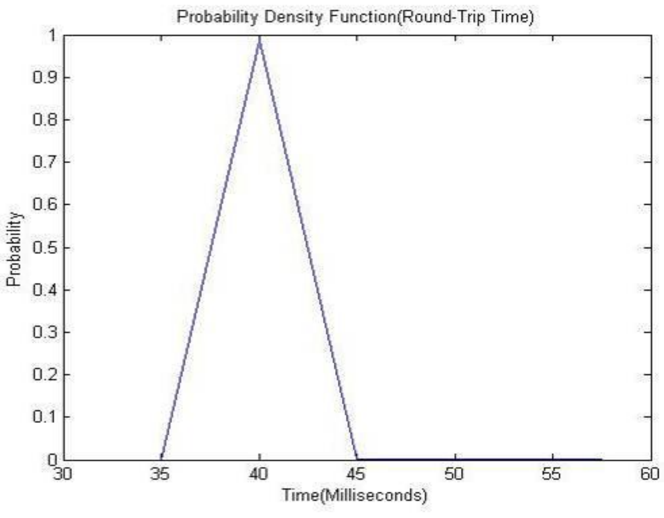


圖十三、頻帶釋出時間

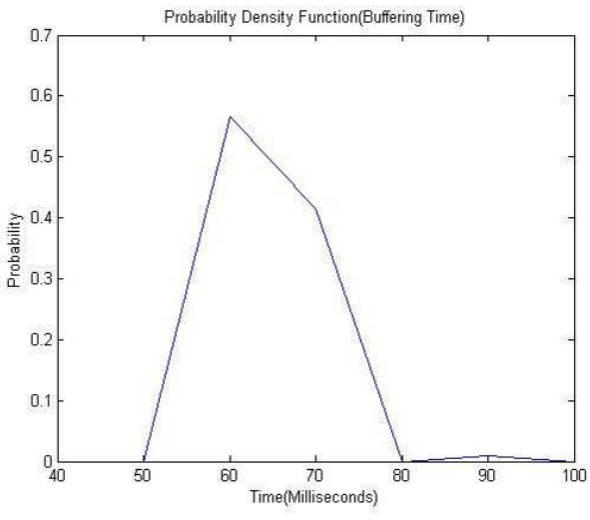
我們觀察到最小、平均以及最大的頻帶釋出時間分別為 1.880、3.1623、和 5.844 秒，這段時間包含了許多過程如存取資料庫、執行 RPM 的建置相關頻譜估測演算法和網路通訊時間，在這邊我們逐步分析這些時間是如何出現，首先我們看到最佳的情況，這個狀況是出現在 PU 一出現後，SDs 馬上緊接的進行感測，並且雲端即時的在 CR AP 詢問雲端頻帶狀況前，執行完 RPM 的建置，並且在詢問過後得到正確的 PU 相關資訊，在我們 CR MAC 中，AP 像雲端詢問的時間會在最後一個 frame 約在 1.6 sec 的時間詢問，網路通訊時間估測為 0.2 秒，因此最佳的時間會落在 1.6 + 0.2 = 1.8 秒，是十分接近我們的實驗結果。

換個情境觀察，若 PU 出現的時間點坐落在 SDs 執行完感測動作之後，正確的頻帶資訊將會在第二個 Superframe 被 SDs 感測，而感測的資訊將會在第二個 Superframe 開始後的 650ms 上傳至雲端 (感測時間 490ms 和回報時間 160ms)，因此雲端若要能及時回傳，約略只有 0.95 秒去執行 RPM 的建置，若雲端來不及在第二個 Superframe 回報前解出，那正確的頻帶資訊則會在第三個 Superframe 時回傳，所以最差的頻帶釋出時間會坐落在 4 + 1.6 + 0.2 = 5.8 秒左右，也十分符合我們的實驗結果。

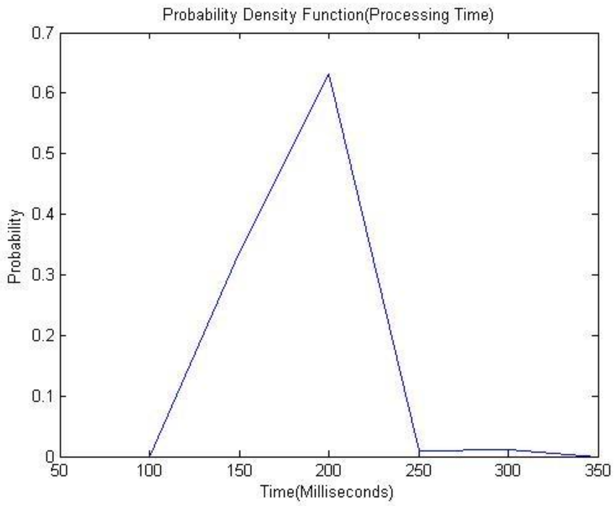
第二部分，我們將實驗在我們的架構下換手(Handover)所花費的時間，我們在每段時間的測量上均取 1000 次抓取其平均值，整體換手的時間主要包含了封包來回時間(Round-Trip Time) (42 ms)，儲存暫存資料至暫存器和更新離線資訊的時間 (70 ms)，以及抓取暫存資訊和查詢目的端所在位置的時間(204 ms)，若忽略了 CR AP 處理資料的時間，我們換手的時間會在 42 + 70 + 204 = 316 ms 內完成，這數據說明了我們能在 1 個 frame 之內將雲端的資訊更新，並且將舊有暫存資料重新傳送給新的所屬 AP，完成完整的換手，並保證資訊不遺漏。



(a) 封包來回時間



(b) 暫存資料及更新狀態時間

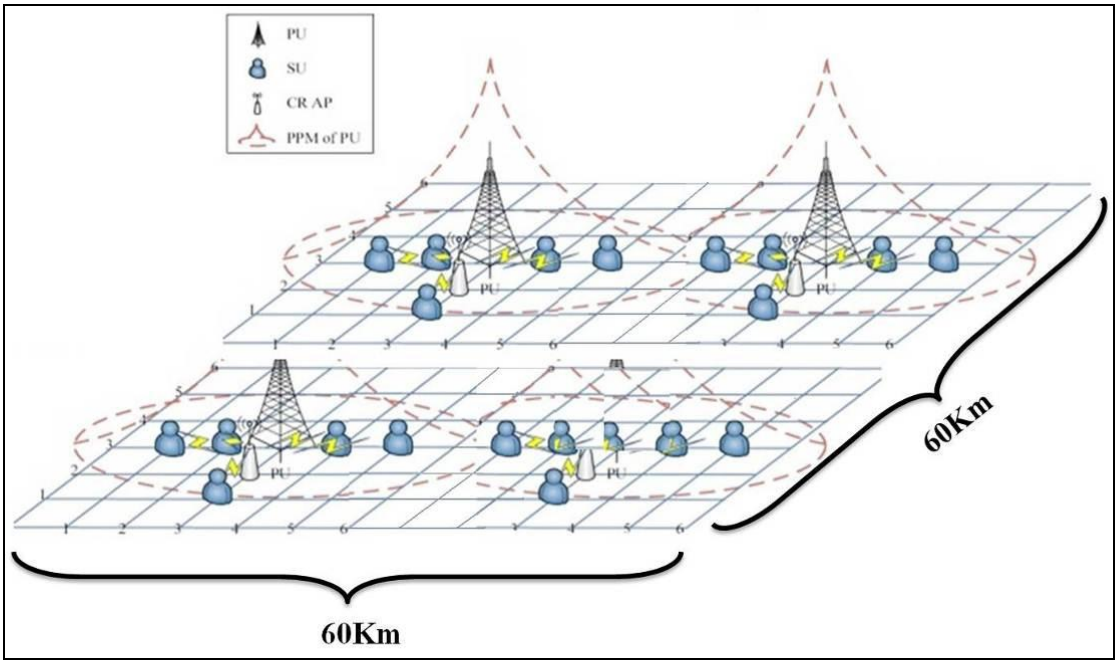


(c) 暫存資料及查詢時間

圖十四、換手執行中內部所花時間

2. 改進 Hadoop 雲端平台模擬頻譜估測演算法的執行時間

在研究方法中，我們以 Hadoop 內含的一個簡單的 PiEstimator[15]來測試我們的發現與改進，並得到了不錯的結果，接下來我們要正式來量測我們的修改對頻譜估測演算法的效果，為此，我們設計了一個有許多 PUs 與 SUs 的環境，如下圖十五，模擬環境包含了個 60km*60km 的區域，有三個 PU 分別在(15, 15)、(45, 15)和(15, 45)這三個地方，SU 則隨機分布在區域內，並有一個參數 Measurement Rates 代表著 SU 的密度，SU 的數量 = 面積(km^2)* Measurement Rates ，以下圖而言，若 Measurement Rates 取 0.1，則共會有 3600*0.1=360 個 SU。每個 SU 所量測到的 RSSI 是先直接透過 PU 的參數計算該位置的 RSSI 在隨機加上 Gaussian noise 來代表現實世界中對訊號的影響。



圖十五、模擬環境示意圖

執行頻譜估測演算法時，CRCN 會先蒐集所有的 SUs 量測到的 RSSI 以及他們的位置，之後在 map 階段依位置來切割計算的區域，每一個切割的區域再各自以一個 reduce task 同時做運算，以達到加速計算的目的，本模擬環境將會切為四個 30*30 的區域來同時計算以增加效率。實驗將會執行 measurement rate 為 0.05、0.075、0.1、0.125、與 0.15 的情況，每個 measurement rate 將執行三次以得到較可靠的結果。實驗將在三種不同的平台上執行分別是我們實驗室自己的機器，以及真實世界雲端供應商 EC2[23]上不同的機器等級組成的環境，詳細資訊請參考下表六。

表六、各種平台資訊比較表

	實驗室自組平台	EC2 Large Instance	EC2 Extra Large Instance
Worker number	2	2	2
CPU	I7 2600	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)
Memory	16 GB	7.5 GB	15 GB
Disk space	1 TB	850 GB	1,690 GB
Mapper/Reducer max num.	8	8	8

從表七中我們可以看到經過我們的修改，在不同 measurement rate 之下執行頻譜估測演算法的執行時間大福的減少 (約 23 秒) ，執行時間減少至只有數秒鐘，讓大範圍的感知無線網路更有效率。

表七、實驗室自組平台執行頻譜估測演算法的時間比較

	Hadoop 原始設定	修改過的 Hadoop
0.05	26.777 sec	3.35 sec
0.075	26.918 sec	3.25 sec
0.1	26.924 sec	3.82 sec
0.125	27.986 sec	4.29 sec
0.15	28.619 sec	5.89 sec

為了比較目前市面上可用的 public cloud 的實際效能，我們選擇了 EC2 平台來與我們實驗室的自組平台來做比較，表八是我們用三台 EC2 Large Instance 執行頻譜估測演算法的執行時間，可以看到整體的時間花費是比實驗室的自組平台來的慢，不過經過修改仍然有降低整體的執行時間。

表八、利用 EC2 Large Instance 執行頻譜估測演算法的執行時間

	Hadoop 原始設定	修改過的 Hadoop
0.05	25.689 sec	9.75 sec
0.075	25.684 sec	9.99 sec
0.1	27.280 sec	10.89 sec
0.125	27.447 sec	11.64 sec
0.15	32.554 sec	17.05 sec

表九是我們使用 EC2 等級更高的 VM 來量測執行時間，可以看到比起 Large Instance，Extra Large Instance 的效能更接近實驗室的自組平台。

表九、利用 EC2 Extra Large Instance 執行頻譜估測演算法的執行時間

	Hadoop 原始設定	修改過的 Hadoop
0.05	24.436 sec	5.52 sec
0.075	24.327 sec	5.56 sec
0.1	26.463 sec	7.11 sec
0.125	27.780 sec	7.66 sec
0.15	30.342 sec	11.30 sec

另外，為了測試整個演算法的可擴充性，我們將之前 60*60 大小的模擬環境擴充至 300*300 大小，其中包含了 80 個 PU，並以 14 個 EC2 Extra Large instance 來同時執行，整個 300*300 的區域會被切割為 100 個 30*30 大小的區域，並同時使用 100 個 reduce task 來做計算，我們依照之 前表六的設定，每個 Task-Trackers 能夠同時執行的 reduce task 為 8，故 100 個 reduce task 須要 14 台 Virtual machine 來同時計算(其中一台執行 Job-Tracker，另外 13 台 Task-Trackers 最多能夠同時執行 104 個 reduce task)，從表十中可以看到執行的時間增加了許多，就我們目前的觀察，時間增加的關鍵在於所有 reduce task 同時向唯一的 map task 要求處理好的中間資料時，map task 無法同時處理太多要求而導致執行的時間增加，這個問題的解決方法仍有待進一步的研究。

表十、區域大小 60*60 的模擬與區域大小 300*300 的執行時間

	區域大小 60*60 的模擬時間	區域大小 300*300 的模擬時間
0.05	5.52 sec	11.21 sec
0.075	5.62 sec	12.54 sec
0.1	7.11 sec	13.84 sec
0.125	7.66 sec	16.93 sec
0.15	11.30 sec	21.55 sec

六、 結論

在這份期末報告，各子計畫的研究團隊分別在排程演算法、雲端平台建置以及合作式頻譜估測三個議題上努力，並且整合個子計畫的成果，實作一個小型的 CRCN 通訊模型，實測頻譜量測所需時間以及 SUs 在該模型下對於 PUs 產生的影響。討論在 CRCN 的架構下，對於 PUs 使用權的保護以及探討 CRCN 作為跨區域網路的可能性，並且比較不同雲端平台 (Windows Azure 以及 Hadoop) 上演算法的效能，分析雲端運算中延遲產生的原因，為未來在雲端平台上的改進提供了想法。表十一彙整了計畫完成之研究細項。

在頻譜感測演算法部分，我們將原本頻譜估測演算法對於不同的地區以及頻譜的特性在雲端平台上進行平行運算，並利用雲端平台運算的特性，減少由於運算所產生的延遲。在雲端系統上，除了進行頻譜運算的運算單元，我們也建立了許多控管的運算單元，控制資料在不同 CR AP 之間的流動，以及 SU 註冊的資訊，用以實作 CRCN 通訊系統。

我們已完成將即時處理的 MAC 層協定實作在 CC1111 及 HMTR 的韌體上。配合 CC1111 和 HM-TR 上 MAC 通訊協定，我們所發展的 MAC 層協定上，可以同時控制協調 HMTR 及 CC1111 兩種通訊模組，讓兩種不同的通訊模組能在同樣的頻段上操作且不互相干擾。

在 CRCN 網路中，整體的管理則交由 CR AP 實現。為了完成這樣的目標，我們在 WARP 發展平台上架設了 Linux 作業系統，讓 WARP 能作為我們系統中的 CR AP 使用，並能與雲端進行通訊。除此之外，我們也實現了網域內及網域間的資料傳輸，並提供簡單的換手機制和保護資料不遺失的暫存機制。

在雲端平台部分，我們在減少 Hadoop 執行頻譜估測演算法的時間上有了重大的突破，能夠搭配對 SUs 做區域的切割來達到平行化計算以減少計算時間的目的，同時，我們也針對 Hadoop 做了修改，讓計算量較小的 job 能夠迅速的執行完畢。結合以上兩點，我們成功的讓頻譜估測演算法的執行時間降低至數秒鐘，因此能夠更快速的更新 PU 的頻譜資源資料庫讓整個 CRCN 的運作更即時。

表十一、計劃完成之研究細項

分項	研究項目
分項一	電視空白頻段(TV White Space)上傳送與接收模組購買與建置
	設計適用於雲端平台上之合作式頻譜圖重建演算法
	在 Microsoft Azure 平台實作並驗證頻譜圖重建演算法
	以分項二訂定之 FD-TDMA based MAC 為基礎，運用電視空白頻段上的傳送與接收模組以及 Microsoft Azure 平台，並結合合作式頻譜估測演算法共同架構一 CRCN 之實驗雛形系統
	藉由 CRCN 之實驗雛形系統實作與驗證頻譜感測資料之擷取、傳收，並即時於 Microsoft Azure 上進行合作式之頻譜圖重建
	藉由 CRCN 之實驗雛形系統驗證與測試不同 Microsoft Azure 雲端架構對合作式頻譜圖重建演算法性能之影響
	搭配分項三之 Hadoop 平台實作並驗證頻譜圖重建演算法
	分析與設計合作式頻譜圖重建演算法於雲端上之 scale out 架構
	進行電視空白頻段合作式頻譜圖重建實驗
	搭配分項二與三，在 Azure 平台共同架構一 CRCN 之實驗雛形系統
	藉由 Azure 平台之 CRCN 實驗雛形系統實作與驗證頻譜感測資料之擷取、傳收與即時之合作式頻譜圖重建
	以 WARP 平台建構 CRCN 實驗雛形系統之 CR AP
分項二	設計分散式頻譜資源排程與頻道存取演算法
	電腦模擬分散式頻譜資源排程與頻道存取演算法
	訂定 FD-TDMA based MAC 訊框架構以及狀態轉換圖
	以應用程式方式實作 FD-TDMA based MAC 協定並在筆記型電腦上進行測試
	以 FD-TDMA based MAC 為基礎，運用電視空白頻段上的傳送與接收模組以及 Windows Azure 平台，並搭配分項一之合作式頻譜估測演算法共同架構一 CRCN 之實驗雛形系統
	在 Microsoft Azure 平台上與分項一之合作式頻譜估測演算法整合，進行實測
	藉由 CRCN 之實驗雛形系統實測頻譜感測資料之擷取，並搭配 Microsoft Azure 與分項一之頻譜估，實作與驗證電視空白頻段上頻道之擷取控制與資料傳收
	在 WARP 上測試並實作 FD-TDMA based MAC
	以 firmware 方式實作 FD-TDMA based MAC 協定於通訊模組 (CC1111 & HMTR) 進行測試
分	建立 Hadoop 運算平台

項三	安裝與 Hadoop 平台搭配的 HBase 資料庫
	搭配分項一於 Hadoop 平台實作並驗證頻譜圖重建演算法
	經由模擬測試 Hadoop 平台的效能瓶頸
	探討利用 Hadoop 平台平行化頻譜圖重建演算法的效能提升程度
	探討 Hadoop 平台效能瓶頸以及改善方法
	改善 Hadoop 平台效能
	與分項一配合提升頻譜圖重建演算法的效能
	分析合作式頻譜圖於 Hadoop 平台以及 Microsoft Azure 平台之性能差異，以及適用合作式頻譜圖之雲端架構

七、參考文獻

[1] J. Mitola III, J. Maguire, and Q. Gerald, Jr. “Cognitive radio: making software radios more personal,” IEEE Personal Communications Magazine, pp. 13 – 18, Aug. 2009.

[2] FCC, Spectrum Policy Task Force Report, ET Docket No. 02-155, Nov. 02, 2002.

[3] Final framework description, preliminary cognitive manager structure and first mechanisms for QoS support. FP7-ICT-2009-4/248454 QoSMOS project, 2011.

[4] H Harada, “Cognitive Wireless Cloud: A Network Concept to Handle Heterogeneous and Spectrum Sharing Type Radio Access Networks” IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Sept 2010.

[5] H. Harada, "A Software Defined Cognitive Radio Prototype," IEEE PIMRC 2007, Sept. 2007.

[6] F. Akyildiz, W. Y. Lee, M.C. Vuran and S. Mohanty, “A Survey on Spectrum Management in Cognitive Radio Networks,” *IEEE Communications Magazine,* April 2008.

[7] F.Ge, R. Rangnekar, A. Radhakrishnan, S. Nair, A. Fayez, Q. Chen, A. Young, Y. Wang, M. D. Silvius, T. Brisebois, G. Marballie, X. Cheng, N. He, B. Li, C. W. Bostian, and M. Hsiao, “A Heterogeneous Cognitive Radio Network Enabling Dissimilar Cooperative Spectrum Sensing, Dynamic Spectrum Access, Interoperability”, White paper for demonstration in IEEE Proc. DySPAN, Chicago, October, 2008.

[8] J.Sydor, D.Roberts, B.Doray and A.Ghasemi, “Cognitive, Radio-Aware, Low-Cost (CORAL)Research Platform” , IEEE DySPAN , May 2010.

[9] J.Sydor, D.Roberts, B.Doray and A.Ghasemi, ”A generic cognitive radio based on commodity hardware” , IEEE INFCOMW , June 2011.

[10] FCC 08-260, November 14.

[11] B. L. Mark and A. O. Nasif, “Estimation of Interference-Free Transmit Power for Opportunistic Spectrum Access,” in Proceedings of IEEE Wireless Communications and networking Conference (WCNC), Las Vegas, NV, pp. 1679-1684, March 2008.

[12] T. Do, B. L. Mark, “Joint Spatial-Temporal Spectrum Sensing for Cognitive Radio Networks,” IEEE Transactions on Vehicular Technology, Vol. 59, No.7, pp. 3480 – 3490, Sept. 2009.

[13] J. C. Jia and Q. Zhang, “HC-MAC: A Hardware-Constrained Cognitive MAC for Efficient Spectrum Management,” IEEE Journal on Selected Areas in Communication, Vol. 26, No.1, pp. 106-117, Jan. 2008.

[14] B. Hamdaoui and K. G. Shin, “OS-MAC: An Efficient MAC Protocol for Spectrum-Agile Wireless Networks,” IEEE Transactions on Mobile Computing, Vol. 7, No. 8, pp. 915-930, Aug. 2008.

[15] H. Su and X. Zhang, “Cross-Layer Based Opportunistic MAC Protocols for QoS Provisioning Over

Cognitive Radio Mobile Wireless Networks," IEEE Journal on Selected Areas in Communication, Vol. 26, No. 1, pp. 118-129, Jan. 2008.

[16] C. Cordeiro and K. Challapali, "C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks," in Proceedings of IEEE Dynamic Spectrum Access Networks (DySPAN), Dublin, Ireland, pp. 147-157, Apr. 2007.

[17] T.-K. Lou, H.-L. Chao, and T.-J. Tsai, "On Synchronized Channel Sensing and Accessing for Cognitive Radio Users in IEEE 802.11 Wireless Networks, " in Proceedings of IEEE Personal, Indoor and Mobile Radio Communications (PIMRC), Tokyo, Japan, pp. 757-761, Sep. 2009.

[18] Available at http://Hadoop.apache.org/hdfs/

[19] Hadoop: The Definitive Guide MapReduce for the Cloud By Tom White

[20] Windows Azure Platform, available at http://www.microsoft.com/azure/default.mspx.

[21] Available at http://warp.rice.edu

[22] http://hadoop.apache.org/common/docs/current/api/org/apache/hadoop/examples/PiEstimator.html

[23] http://aws.amazon.com/ec2/instance-types/

八、附錄

附錄 1

Cooperative Spectrum Sensing in TV White Spaces: When Cognitive Radio Meets Cloud

Chun-Hsien Ko, Din Hwa Huang and Sau-Hsuan Wu,
Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan 300
E-mail:{dke.cm99g@nctu.edu.tw, dinhwa.cm94g@g2.nctu.edu.tw, sauhsuan@cm.nctu.edu.tw}

Abstract—A Cognitive Radio Cloud Network (CRCN) in TV White Spaces (TVWS) is proposed in this paper. Under the infrastructure of CRCN, cooperative spectrum sensing (SS) and resource scheduling in TVWS can be efficiently implemented making use of the scalability and the vast storage and computing capacity of the Cloud. Based on the sensing reports collected on the Cognitive Radio Cloud (CRC) from distributed secondary users (SUs), we study and implement a sparse Bayesian learning (SBL) algorithm for cooperative SS in TVWS using Microsoft's Windows Azure Cloud platform. A database for the estimated locations and spectrum power profiles of the primary users are established on CRC with Microsoft's SQL Azure. Moreover to enhance the performance of the SBL-based SS on CRC, a hierarchical parallelization method is also implemented with Microsoft's dotNet 4.0 in a MapReduce-like programming model. Based on our simulation studies, a proper programming model and partitioning of the sensing data play crucial roles to the performance of the SBL-based SS on the Cloud.

Index Terms—Cognitive Radio, Cloud Computing, MapReduce, Windows Azure, Sparse Bayesian Learning.

I. Introduction

The concept of cognitive radio (CR) is first introduced by Joseph Mitola III in [1]. Since then CR has attracted significant research attentions either in telecommunications technologies or their related regulations. In view of the inefficient usages of the licensed spectrum (less than 25% overall [2]) and the opportunity of the termination of analog TV broadcasting, the Federal Communications Commission (FCC) of the U.S has established the regulation for CR accesses in its TV White Spaces (TVWS) in 2007 [3] and has recently granted field tests of the CR network in part of the TVWS. Encouraged by the acts of FCC, many international organizations have also started to define CR standards on TVWS, *e.g.* IEEE 802.22, 1900, 802.16m and ECMA 392, etc.

To ensure the received signal quality of TV sets, the FCC requires CR operators in TVWS being able to detect the TV signal even if its strength is 0.8dBm below the noise level (-106.2dBm). In addition, the CR operators should also provide databases that maintain the geographical locations of TV base stations (BS) and their radiation powers, antenna heights and numbers of channels, *etc*. To help achieve these goals of spectrum sensing (SS) in TVWS, the secondary users (SUs) of the CR network in TVWS are suggested to provide their sensing data and geographical locations for CR operators to perform cooperative SS.

Compared to the BSs of regular cellular networks, the radiation power of a TV base station (BS) typically covers a much larger area than that covered by the transmit power of a mobile device. To reconstruct the radiation power profile of even a TV BS, it requires sensing reports from SUs located in different positions inside the coverage area of the TV signal. However, the distribution and population density (sparsity) of SUs are not uniform in different areas, and vary in different time of a day. Moreover, the received signal strength of a SU is likely to be attenuated by the shadowing effects of wireless channels. Considering these characteristics of sensing measurements and the strict requirements of SS in TVWS, we study a cooperative SS algorithm for TVWS in this paper based on the concept of sparse Bayesian learning (SBL) and the relevance vector machine (RVM) [4].

As the number of SUs vary with time, the computational demands to reconstruct the power propagation map (PPM) of a large area may change significantly over time if sensing reports inside the area are all used in cooperative SS. To control the algorithm complexity and in the meantime to maintain the quality of SS, not only should the number of sensing measurements be limited inside an area, but the area from which measurements are collected for the computation of a RVM should also be adjusted with time. Consequently, the overall computational quantity to reconstruct the PPM of a nation or region will scale up and down significantly over time. This makes the SS in TVWS an ideal application for Cloud computing.

A similar concept of cognitive wireless Cloud (CWC) has been introduced by H. Harada *et al* in [5] where they consider a heterogeneous network that consists of various types of wireless networks and propose a Cloud-based algorithm to optimize the spectrum resource scheduling among the heterogeneous networks in CWC. In contrast to their ideas in heterogeneous networks, we propose herein a more complete concept of Cognitive Radio Cloud Network (CRCN) that enables and integrates cooperative SS and resource scheduling in TVWS. Making use of the scalability and the vast storage and computing capacity of the Cloud, the database of PPM can be established, updated and accessed by a large amount of SUs in an efficient manner. Under this infrastructure of CRCN, we study and implement a SBL-based cooperative SS algorithm on Microsoft's Windows Azure Cloud platform, and propose a scalable mapping method under a MapReduce-like programming model to dynamically partition the geographical area according to the SU density. Utilizing the scalable mapping method and the dynamic computing resource allocation of the Cloud, the CRCN can provide a PPM in different precisions according to the density of SUs. Based on our simulation studies, a proper programming model and partitioning of the sensing data play crucial roles to the performance of the SBL-based cooperative SS on the Cloud.

This paper is organized as follows. Section II specifies the system model for CRCN and provides some background knowledges on Windows Azure. Section III reviews the basic concept of SBL and the MapReduce [6] programming model, and introduces a SBL-based cooperative SS and a scalable mapping method on Window's Cloud platform. Simulation results are presented in Section IV followed by some conclusions and discussions made in Section V.

II. The Infrastructure of CRCN

The purpose of CR is to utilize the precious radio resources more intelligently. Fig. 1 illustrates the infrastructure of the CRCN proposed in this paper. SUs in the CRCN are allowed to use a spectrum in time and space as long as not seriously deteriorating the signal qualities of primary users (PUs) in the same spectrum. To make the most out of the available spectra, a command and control center (also referred to the CR Cloud (CRC)) is used to coordinate and manage the entire radio resources in TVWS. In the CRCN, there are various CR BSs to collect sensing measurements from distributed SUs. The sensing results are fed back through CR BSs to the CRC to estimate the PPM with a SBL algorithm implemented on Microsoft's Windows Azure. The resultant PPM of SS contains the number of PUs and their locations and corresponding radio power profiles, and are stored in Microsoft's SQL Azure.

A. The Windows Azure CRC Platform

The CRC is in fact implemented on Microsoft's Windows Azure Cloud platform which can support program developments in JAVA or in C] and Visual Basic on Visual Studio. The operating system for Windows Azure is Windows Azure Guest OS 1.8 which is a virtual machine (VM) version of Windows Server 2008. Windows Azure supports three types of data storages which are BLOB for general binary data, Table for systematic data and Queue for data passing between webs and programs.

For the programming model in Windows Azure, there are two different roles which are:

- Web Role: The task of web role is to communicate between users and background processes. It can be implemented by dynamic web language, for example, ASP.NET and PHP, etc.
- Worker Role: It is a background process in Windows Azure. Worker Roles grab and execute jobs, and then export the results periodically.

On the other hand, SQL Azure is the Cloud version of SQL server and is built on Windows Azure. Designed for Cloud, SQL Azure only supports part of the functions of SQL server.

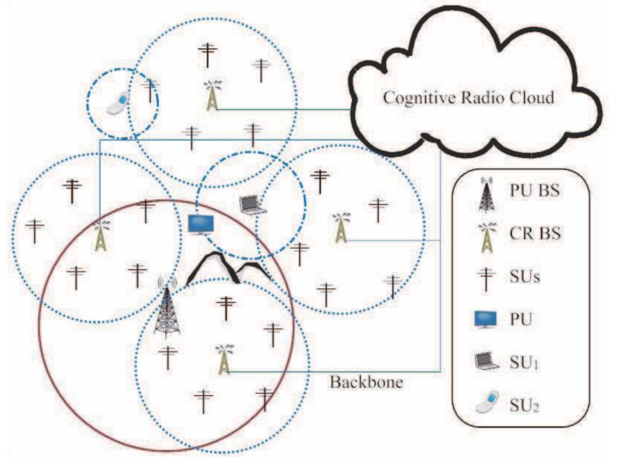


Fig. 1. A conceptual map of the *Cognitive Radio Cloud Network*, which illustrates the geographical relationship among the PUs, PU BSs, CR SUs and CR BSs of the network. As shown in the figure, the received signal of SU_1 is blocked by a mountain which makes the SU_1 difficult to detect the PU BS by itself. Thus, SU_1 will interfere with the PU when it uses the same frequency band to connect to the CR BS. With the CRCN, this shadowing effect can be easily resolved and the frequency band will be allocated to SU_2 to enhance the spectra efficiency.

To reconstruct the PPM on Windows Azure, each SU inside the CRCN is assumed equipped with a global positioning system (GPS) device and is able to feed back its location, time and value of the received signal strength indicator (RSSI) to the CRC through the Web Role of Windows Azure as shown in Fig. 2. Each Web Role takes the inputs sent from a CR BS and stores the sensing reports in the input database of SQL Azure. The CRC then partitions the sensing measurements in the input database into blocks according to their associated positions, and maps the data of each block to a Worker Role of Windows Azure to parallelizes the SBL algorithm. A Worker Roles performs the SBL-based SS algorithm with the sensing measurements of each block and stores the reconstructed PPM of PUs of each time slot in the output database of SQL Azure. If a SU wants to access the CRCN, it first sends a request together with its location to the CRC to ask for permission. The CRC will allocate the radio resource to the SU according to the PPM of PUs and the locations of all users both stored in the input and output databases of SQL Azure.

III. The Implementation of the SBL-Based Cooperative SS on CRC

More details about the SBL-based cooperative SS algorithm and how we implement and parallelize the algorithm on the CRC are provided in this Section. A scalable mapping function is also proposed to adjust the block scale of each Worker Role.

A. The SBL-Based Cooperative SS Algorithm

Assume that there are N PUs in an area of $N_p \times N_p$, and M_p CR BSs to collect these PUs' sens- ing results $\mathbf{t} = (t_1, t_2, \cdots, t_N)^T$ and locations $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$, with $\mathbf{x}_j = [x_j, y_j]^T$, and feed back them to the CRC. We select a basis function $\varphi_j(\mathbf{x}_i) =$

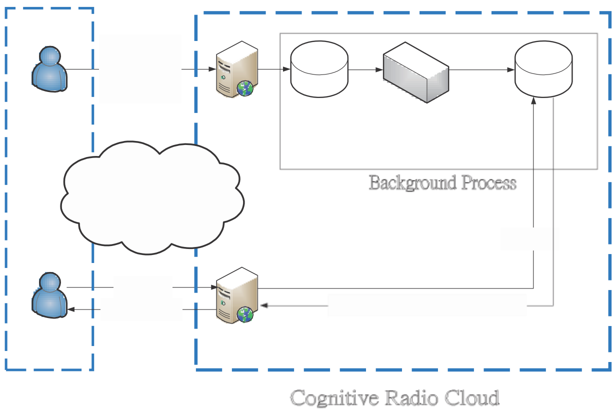


Fig. 2. *The block diagram of the Cognitive Radio Cloud platform*. The platform includes two dynamic web pages and one database of SQL Azure for SUs and one for PUs. The SBL-based cooperative SS algorithm is operated in Worker Roles and triggered periodically.

$\frac{1}{2s_j}\exp\left\{-\sum \left((x_{i,x}-\mu_{j,x})^2+(x_{i,y}-\mu_{j,y})^2\right)/s_j\right\}$ and use the SBL to solve the regression problem of [7]

$$t = \Phi w + n \tag{1}$$

where $\Phi_{N\times M} = [\psi_1(X), \psi_2(X), \cdots, \psi_M(X)]$, with $\psi_j(X) \triangleq (\varphi_j(x_1), \varphi_j(x_2), \cdots, \varphi_j(x_N))^T$, is the basis matrix determined by the number M, the locations $\mu = (\mu_1, \mu_2, \cdots, \mu_M)^T$, with $\mu_j \triangleq [\mu_{j,x}, \mu_{j,y}]$, and the power decaying rates $s = (s_1, s_2, \cdots, s_M)^T$ of the bases, and also by the number N and the location vector X of the SUs. The vector $w = (w_1, w_2, \cdots, w_M)^T$ denotes the weighting coefficients of the basis functions, and each of its entries w_j is endowed a *prior* probability $N(0, \alpha_j^{-1})$. The n denotes the shadowing effect, with each entry being a zero-mean Gaussian random variable (RV) with variance β^{-1}. The SBL iteratively modifies the RVM and estimates the parameters M, α_j^{-1}, β^{-1}, μ_j, and s_j to maximize the marginal likelihood function $p(t|X, \alpha, \beta, \mu, s, M)$ from sparse measurements.

The SBL can also be viewed as an alternative EM algorithm. In the E-step, the covariance Σ and mean m of the posterior distribution of weighting coefficients w are evaluated by

$$\Sigma = (\beta\Phi^T\Phi + A)^{-1}, \text{ and } m = \beta\Sigma\Phi^T t \tag{2}$$

where $A \triangleq \mathrm{diag}(\alpha_j)$ is a $M \times M$ diagonal matrix. Consequently, we can obtain the estimated weighting coefficients $\hat{w} = m$ and then delete those low-weighted bases according to $\hat{w}$. Here we select the bases whose weighting coefficients are larger than a threshold η and count their number to renew the M. For the k-th iteration of SBL, we have

$$M_{(k)} = \sum_{j=1}^{M_{(k-1)}} I(\hat{w}_j \geq \eta) \tag{3}$$

where I is the indicating function. Since there are two M-steps in SBL and m is mainly adjusted in the first M-step. The bases deleting criterion is only applied in the first EM in which the basis parameters μ, s and M are assumed known. Therefore, the variance parameters α_j^{-1} and β^{-1} are estimated as

$$\alpha_j^{-1} = \frac{m_j^2}{\gamma_j}, \text{ and } \beta^{-1} = \frac{\|t-\Phi m\|^2}{N-\sum_{j=1}^{M}\gamma_j} \tag{4}$$

where $\gamma_j \equiv 1 - \alpha_j\Sigma_{jj}$, and Σ_{jj} are the diagonal terms of Σ. The smaller is the α_j^{-1}, the more likely is φ_j a redundant basis. Besides, the deleting threshold η in (3) should be set based on the noise variance β^{-1}. In the second M-step, w, β^{-1}, and M are assumed known. We infer the basis parameters μ and s that maximize the likelihood function $p(t|\mu, s; X, M, \beta, \alpha)$ by the *gradient descent method*

$$\begin{bmatrix}\mu_{j,x}(k)\\ \mu_{j,y}(k)\\ s_j(k)\end{bmatrix} = \begin{bmatrix}\mu_{j,x}(k-1)\\ \mu_{j,y}(k-1)\\ s_j(k-1)\end{bmatrix} - \delta\begin{bmatrix}\frac{\partial Q}{\partial \mu_{j,x}}\Big|_{\mu_{j,x}(k-1)}\\ \frac{\partial Q}{\partial \mu_{j,y}}\Big|_{\mu_{j,y}(k-1)}\\ \frac{\partial Q}{\partial s_j}\Big|_{s_j(k-1)}\end{bmatrix} \tag{5}$$

where k is the iteration index, $Q \triangleq -\ln p(t|\mu, s; X, M, \beta, \alpha)$ and $\delta > 0$ is the step size or referred to as the *learning rate*. The details of the iteration process is shown in Table I.

According to the compressive sensing (CS) theorem [8], a signal can be exactly reconstructed when the measurement rate (N/N_p^2) is larger than 0.16. Even though the SBL algorithm does not adopt orthogonal bases and as such does not abide by the CS theorem, the CS theorem still provides for the SBL algorithm a useful reference figure on the measurement rate. In the sequel, we only simulate the cases whose measurement rates are less than 0.15. The estimation errors do not improve significantly when the measurement rates are larger than 0.15, while the complexity will increase dramatically.

TABLE I

The Sparse Bayesian Learning Algrithm
1) Uniformly spread M bases φ_j in the area of interest.
2) Initiate the iterations with $\alpha_j = 1$, $\beta = 1$ and $k = 0$, and evaluate the corresponding mean m and covariance Σ.
3) Let $k = k + 1$. Update α^{-1} and β^{-1} and then evaluate m, Σ and $Q(k)$.
4) Delete the bases whose corresponding weights $\hat{w}_j < \eta$ and then renew the M equal to the number of the surviving bases. Renew the matrix Φ and A.
5) Let $k = k + 1$. Update $\mu_j = [\mu_{j,x}, \mu_{j,y}]$ and s_j and and then evaluate m, Σ and $Q(k)$. Go to step 6) if $(Q(k) - Q(k-1))/Q(k-1) < 0.0001$. Otherwise, repeat this step for L times, then go back to step 3).
6) Output the $\mu_j = [\mu_{j,x}, \mu_{j,y}]$ and s_j. Let $\hat{M}_p = M$ and $\hat{w} = m$.

B. Area Parallelization with a MapReduce-like method

Because the algorithm complexity of the SBL-based SS scheme grows in the third order of the number of sensing measurements, we partition the sensing data of the distributed

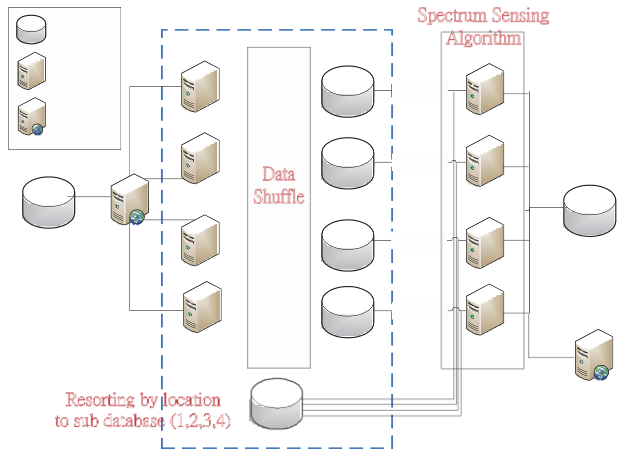


Fig. 3. The flowchart of the parallelized SBL-based cooperative SS algorithm in the background process of Windows Azure. In this example, there are one web role and four worker roles, and each role operates on an individual VM. The web role distributes the sensing data; in contrast, the worker roles execute programs. The detailed execution steps of the algorithm are listed in Table II.

SUs by their locations into blocks to reduce the processing time of the SS algorithm. The data of each block are processed independently by a Worker Role of a VM to execute the SBL-based SS scheme for the block. When the number, locations and the RSSI levels of PUs are obtained, each VM reports the results to a common PUs database.

Under the MapReduce programming model [6], VMs exchange data in a format of (*key*, *value*) pair. Applying this concept to our SS problem, we define the time and the location measurements as the Mapper's *input data key* and *output data key*, respectively. For the Reducer, both the *input data key* and the *output data key* are location information. The flow chart is shown in Fig. 3. We note that VMs and SQLs are not guaranteed to be implemented in the same server, thus exchanging data between VMs might become the bottleneck of our implementation. It is a tradeoff between the degree of parallelism and data exchange. The detailed description of this MapReduce Programming Model is shown in Table II.

TABLE II

MapReduce Programming Model
1) Job tracker distributes SUs' data to different Worker Roles according to the chronical order.
2) Worker Role distributes the SUs' data to different sub-databases according to the SUs' locations. Each worker Role renews its state = 1 in Check SQL when the distribution is done.
3) Worker Roles check state value in the Check SQL. If all state values are equal to 1, then start to run the spectrum sensing algorithm. Otherwise, check state value periodically.
4) Export the estimation results into PUs' database.

C. Hierarchical Parallelization

Although area parallelization can reduce the processing time significantly, the speed improvement is still restricted by the power coverage areas of the PUs, in particular, for a PU like a TV broadcasting station. This is because data processed by a VM should come from an area larger than that covered by the power of a PU to ensure the correctness of the reconstructed PPM of a PU. To lift this fundamental limit on the computational speed of the SS algorithm, one can consider a traditional parallelization method of multi-threading.

Specifically, we consider a hierarchical parallelization structure for the computation of the SBL-based SS algorithm. Measurement data are first partitioned by area into blocks for the algorithm complexity is of the third order of the number of measurements. Each block are handled by one VM with multiple CPU cores. Signal processing within each VM is further parallelized with multi-threading over multiple cores.

In Microsoft's dotNet 4.0, a simple multi-thread instruction of P arallel.F or can be used to parallelize computations This is an advantage of Windows Azure. Unlike Hadoop, Windows Azure allows users to define some system-level properties for the different VMs of Web Roles and Worker Roles. Therefore, using multi-threading in VMs with multiple cores on Windows Azure platform can also reduce the communication cost between VMs when only single-thread instructions are allowed in each VM as in typical MapReduce programming model.

IV. Simulation Results

Before we introduce the simulation results, we first give some figures about the Windows Azure Platform. Windows Azure offers different options of VMs whose system parameters are listed in Table III. These options allow us to do a fairer comparison between the speeds and accuracies of different measurement rates.

We consider herein an area of 60×60 with 3 PUs located at (15, 45), (45, 45) and (15, 45), respectively. A baseline SBL-based SS algorithm is performed for this area on Windows Azure using the large instance in Table III. To study the performance of parallelization on the Cloud, we test three types of parallelization methods for the SBL-based SS algorithm. The Type I performs parallelization for the SS algorithm by simple multi-threading using four CPU cores of small instance in one VM of the Worker Role. In comparison, the Type II (in Host) partitions the entire area into four blocks. Each area includes at most one PU located at the same position relative to the baseline example. Data from each block are processed by one CPU core of a VM with 4 cores. In contrast, the Type II (on Cloud) processes data from each block on a VM of a single CPU, *i.e.*, each Worker Role processes the measurement data from an area of 30×30. Finally, the Type III processes data from each block on a VM with 4 CPU cores. As a results, the total number of CPU cores for the Worker Role becomes 16.

The simulation results for different measurement rates (sparsities) are listed in Table IV to VIII and are also shown in Fig. 4 to Fig. 8. Fig. 4 shows that parallelization by partitioning the area is most crucial to the computation of the SS algorithm.

TABLE III
THE COMPUTE INSTANCE SIZE OF WINDOWS AZURE

Computer Instance Size	CPU	RAM	Storage	I/O efficiency
Extra Small	1.0GHz	768 MB	20 GB	Low
Small	1.6GHz	1.75 GB	225 GB	Moderate
Medium	2 × 1.6GHz	3.5 GB	490 GB	High
Large	4 × 1.6GHz	7 GB	1, 000 GB	High
Extra large	8 × 1.6GHz	14 GB	2, 040 GB	High

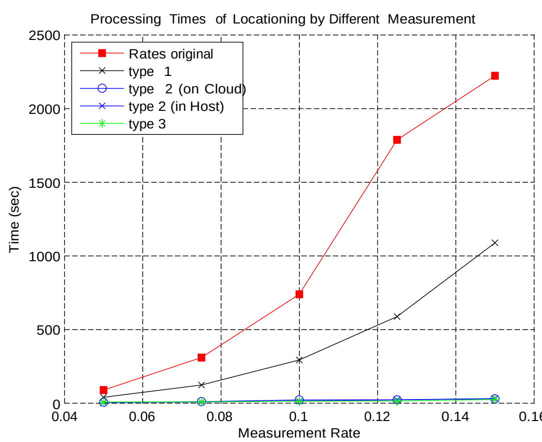


Fig. 4. The average processing times in the CRC platform versus the measurement rates. Only 10 runs are executed for each point.

At the measurement rate of 0.1, Type II (in Host) can improve the processing speed of Type I by 20 times, while Type II (on Cloud) improves the processing speed of Type I around 13 times due to the communication time between the VMs. The computational advantage of Type II results from the matrix inversion involved in the SBL-based SS algorithm since, for the Gaussian-Jordan method, the time complexity grows with the third order of the amount of sensing reports.

TABLE IV
THE SPENDING TIMES (SEC) UNDER DIFFERENT MEASUREMENT RATES

Measurement rates	0.05	0.075	0.1	0.125	0.15
original	89.4	309.9	739.4	1788.4	2223.5
Type I	40.0	124.1	293.2	588.1	1089.0
Type II (on Cloud)	6.3	10.2	22.1	22.9	29.9
Type II (in Host)	3.8	7.8	14.2	22.1	31.2
Type III	6.5	8.9	12.5	14.6	24.5

The hierarchial parallelization algorithm will not effect the complexity, it only reduces the processing time for each area. Nevertheless, this feature is particular useful for TVWS due to the large scale of the power coverage areas of PUs. For the SBL-based SS algorithm, a VM should process sensing data at least from a PU, which prevents from partitioning the area into very small processing blocks.

Table V and VI show the mean squared errors (MSE) of

TABLE V
THE MSES OF LOCATION UNDER DIFFERENT MEASUREMENT RATES

Measurement rates	0.05	0.075	0.1	0.125	0.15
original	0.138	0.091	0.075	0.060	0.050
Type I	0.180	0.09	0.068	0.057	0.048
Type II	0.414	0.276	0.212	0.162	0.167
Type III	0.510	0.306	0.217	0.190	0.158

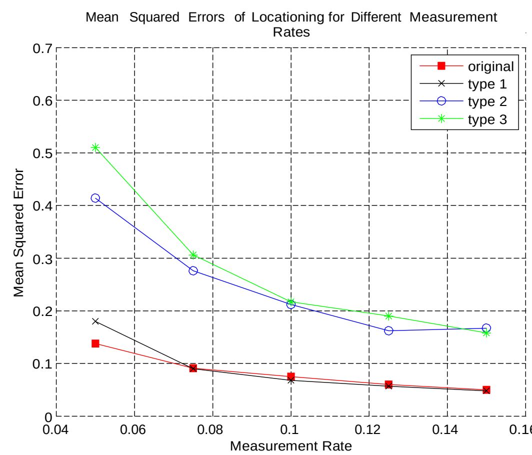


Fig. 5. The average mean squared errors of the SBL-based SS algorithm versus the measurement rates. The result doesn't include the false alarms and missing detection cases.

estimation. Table V shows the MSEs in the locationing of the PUs, and Table VI presents the MSEs of the reconstructed PPM. If the radiation power of a PU is 5KW, one scale in our simulations corresponds to 15km. As a result, the MSE in locationing is around 1.8 km when measurement rate is at 0.1. More results on MSEs are presented in Fig.5 and 6.

Table VII and VIII show the missing ratios and the false alarm ratios of the different types of the implementation

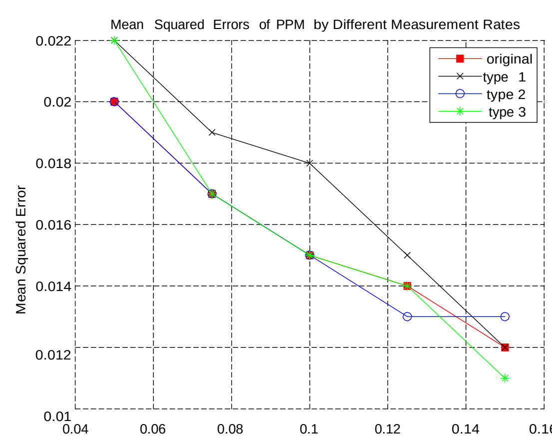


Fig. 6. The average mean squared errors of the SBL-based SS algorithm versus the measurement rates. The result doesn't include the false alarms and missing detection cases.

TABLE VI
THE MSES OF PPM UNDER DIFFERENT MEASUREMENT RATES

Measurement rates	0.05	0.075	0.1	0.125	0.15
original	0.020	0.017	0.015	0.014	0.012
Type I	0.022	0.019	0.018	0.015	0.012
Type II	0.020	0.017	0.015	0.013	0.013
Type III	0.022	0.017	0.015	0.014	0.011

TABLE VII
THE MISSING RATIOS UNDER DIFFERENT MEASUREMENT RATES

Measurement rates	0.05	0.075	0.1	0.125	0.15
original	0.09	0.04	0.04	0.04	0.04
Type I	0.108	0.02	0.011	0	0
Type II	0.11	0.04	0.02	0	0
Type III	0.13	0.03	0.01	0	0

methods for the SBL-based SS algorithm. For the Type II, it appears to have a higher false alarm ratio in an area without PU. To resolve this problem, we set a threshold for the estimated power. With this mechanism, we can find for all of the proposed algorithms that they exhibit consistent results either in the false alarm ratios or the missing ratios.

V. DISCUSSIONS AND FUTURE WORKS

A CRCN was proposed for cooperative SS in TVWS. Based on the SBL algorithm, a cooperative SS algorithm was tested on Microsoft's Windows Azure Cloud platform. Making use of the multi-threading features of the Windows Azure platform, a hierarchial parallelization method was proposed to improve the processing speed of the SBL-based SS algorithm on the Cloud. According to our simulation studies, the performance of the SS algorithm can be greatly improved with the parallel computing capacity and the MapReduce-like programming model of the Cloud. Under the framework of CRCN, more

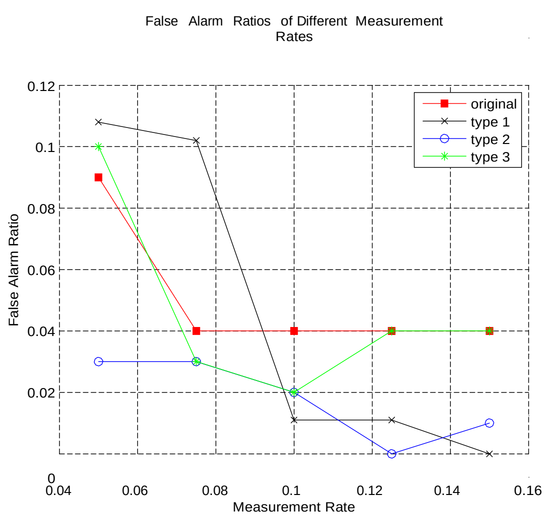


Fig. 7. The false alarm ratios of the SBL-based SS algorithm versus the measurement rates. The power of PU is 30 and the noise is zero mean with variance equal to 2 in this simulation case. Each point runs for 100 times.

TABLE VIII
THE FALSE ALARM RATIOS UNDER DIFFERENT MEASUREMENT RATES

Measurement rates	0.05	0.075	0.1	0.125	0.15
original	0.12	0.04	0.01	0	0
Type I	0.108	0.102	0.011	0.011	0
Type II	0.03	0.03	0.02	0	0.01
Type III	0.10	0.03	0.02	0.04	0.04

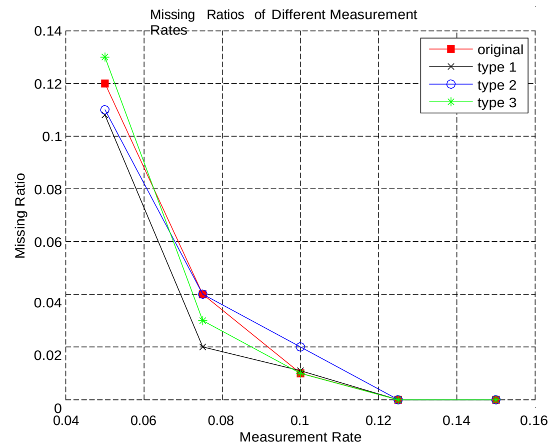


Fig. 8. The missing ratios of the SBL-based SS algorithm versus the measurement rates.The power of PU is 30 and the noise is zero mean with variance equal to 2 in this simulation case. Each point runs for 100 times.

advanced algorithms or ideas on cooperative SS or spectrum resource scheduling can be tested for CR in TVWS.

REFERENCES

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[2] M. A. McHenry, *NSF spectrum occupancy measurements project sum- mary*, shared spectrum co. report, Aug. 2005.

[3] FCC, *Second report and order and memorandum opinion and order*, FCC 08-260, Nov. 2008.

[4] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1.

[5] H. Harada, H. Murakami, K. Ishizu, S. Filin, Y. Saito, H. N. Tran, G. Miyamoto, M. Hasegawa, Y. Murata, and S. Kato, "A Software Defined Cognitive Radio System Cognitive Wireless Clouds ," in *IEEE proceeding of IEEE Global Communications Conference (GLOBECOM)*. Washington DC, USA.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[7] D.-H. Tina Huang, S.-H. Wu, and P.-H. Wang, "Cooperative Spectrum Sensing and Locationing: A Sparse Bayesian Learning Approach," in *Proc. IEEE GLOBECOM*. Miami, USA, Dec. 2010.

[8] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles : Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

附錄二

A Decentralized MAC Protocol for Cognitive Radio Networks

Shuhua Jiang, Li-Hua Chao, and Hsi-Lu Chao
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan
Email: hlchao@cs.nctu.edu.tw

Abstract－One of the most challenging issues in cognitive radio networks is efficient channel sensing and channel accessing. In this paper, an analytical queueing model is used to derive the probability of successful transmission, channel sensing time, and transmission quota, for each data channel. Each CR node records the derived statistics in a channel preference matrix. A CR pair selects a data channel for sensing and accessing based on the successful transmission probability. According to the derivations, we design a media access control protocol, which utilizes the powerful computation capability of cloud servers to estimate the behavior of PUs, for infrastructure-based cognitive radio networks. We validate the analytical model with simulation results. Besides, the proposed MAC protocol is compared with other approaches via simulation. The simulation results showed that our protocol performs well in both utilization of channel idle time and the average tries of channel search.

Keywords － cognitive radio network; channel sensing; channel access

I. Introduction

With rapid increase of the wireless applications and products, unlicensed bands such as Industrial, Scientific and Medical (ISM) has become over-crowded. Cognitive Radio (CR) [1], as a promising solution to efficiently utilize the unused spectrum, has become an attractive research topic nowadays. The concept of the CR technique is that cognitive radio nodes (CR nodes) can temporarily borrow unoccupied channels from primary users (PUs) without interfering with PUs.

To utilize available spectrum efficiently, a media access control (MAC) protocol is of great importance to CR nodes. Existing CR MAC protocols can be classified into two categories: single rendezvous [2-3] or parallel rendezvous [4-5]. The former utilizes a common channel for CR nodes to exchange control messages; in the latter, contrarily, control messages are delivered on data channels.

The major advantage of single rendezvous protocols is the avoidance of collision and meaningless channel hops. This control channel, however, does become a bottleneck. Therefore, how to design a MAC protocol with a control channel efficiently is big challenge.

In [2], CR nodes perform negotiations on a common control channel. Besides, a CR pair can only transmit one packet on the temporarily occupied data channel (or on the control channel). In this mechanism, all CR nodes need to achieve global synchronization. However, this global synchronization significantly decreases the utilization of channel idle time. In [3], a CR pair can exchange at most "*TXQ*" frames once they discover an idle data channel. "*TXQ*" parameter efficiently reduces the average channel sensing time. However, how to properly set "*TXQ*" parameter is not addressed.

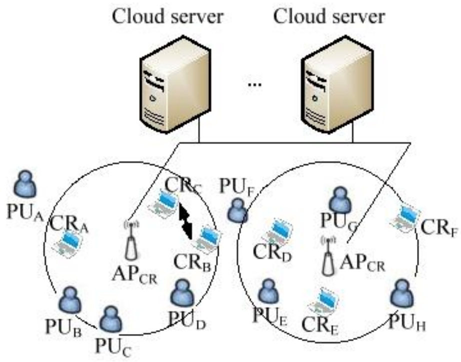


Figure 1. A simple network topology of infrastructure-based cognitive radio networks

On the other hand, the basic idea of parallel rendezvous protocols [4-5] is that nodes hop among different data channels according to their own sequences, and control messages are exchanged when a CR pair meets each other on a data channel. However, synchronization problem and hopping sequence generating function are still opened problems.

In this paper, we propose a cloud server-assisted MAC protocol for infrastructure-based cognitive radio networks (CRN), as in [6]. CR nodes cooperatively and periodically report channel qualities and positions to CR access points (denoted as AP_{CR}). AP_{CR}s further deliver collected information to cloud servers. One characteristic of cloud computing is the provided powerful computation capability. Cloud servers derive the distribution of PUs' arrival rate and channel idle time for each CR node and this information is forwarded by AP_{CR}s. This information helps on a CR node to estimate how much time it should spend on sensing a specific data channel, how many data frames it can deliver, and what the success probability is. CR nodes sense channels in decreasing order of successful probability.

The rest of this paper is organized as follows. Network model and problem description are presented in Section II. The designed cloud server-assisted MAC protocol is described in Section III. Section IV presents and discusses the simulation results, while Section V concludes the paper.

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

II. Network Model and Problem Description

A. Network Model

We consider an infrastructure-based CRN which consists of AP_{CR}s, and CR nodes, as shown in Fig. 1. Besides, cloud servers are used to support the computation overhead of PUs' locations and the distribution of arrival rate and channel idle time. We assume there are *N* orthogonal data channels and one control channel. CR nodes register to an AP_{CR} for joining the CRN. In this paper, we consider single-hop CR flows. That is, two CR nodes can exchange frames when both are within each other's transmission range.

Each CR node equips a GPS and has only one transceiver. CR nodes periodically report their positions and measured channel qualities to AP_{CR}. AP_{CR} further forwards the collected information to cloud servers. Accordingly, for each data channel, cloud servers can identify PUs' locations [7]. Upon knowing positions of PUs' and CR nodes', and taking hidden terminal problem into consideration, cloud servers provide each CR node the distribution of PU traffic arrival rate and idle time for each channel.

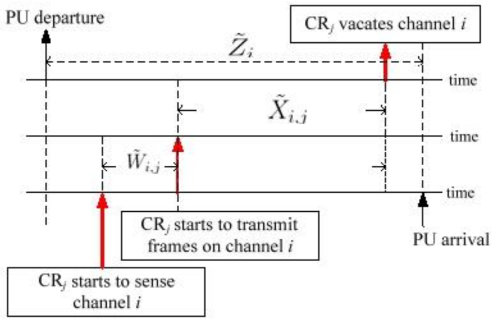
Communications between CR nodes and the AP_{CR} are on the control channel; while the AP_{CR} and the cloud server communicate through a backbone network. The control channel could be either a dedicated channel or an ISM-band channel.

B. Problem Description

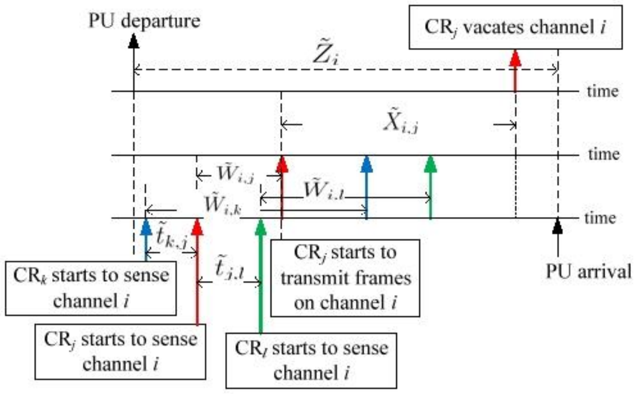
In this paper, we aim at increasing the successful transmission probability of a CR pair while avoiding interfering on PUs. Due to the support of cloud server, each CR node obtains channel and PUs' statistics without performing complex computation [7]. Our design concept is, for a CR node, to use the obtained statistics to estimate the successful transmission probability of each channel. Among all data channels, a CR pair exchanges frames on the data channel which has the highest successful probability. As a result, the major challenge is how to calculate this probability.

Specifically, frame transmission of a CR node is affected by both PUs and other CR nodes. Fig. 2(a) is an example to illustrate how PUs impact on CR transmission. For data channel *i* and CR node *j* (denoted as CR_j), let $\tilde{Z}_i$, $\tilde{W}_{i,j}$, and $\tilde{X}_{i,j}$ represent the channel idle time, channel sensing time, transmission time, respectively. To guarantee CR_j's successful transmission, $\tilde{W}_{i,j}+\tilde{X}_{i,j}\le\tilde{Z}_i$. Therefore the probability that CR_j will successfully deliver frames on channel *i* without interfering PUs is $p_1=P(\tilde{W}_{i,j}+\tilde{X}_{i,j}\le\tilde{Z}_i)$.

On the other hand, the impact from other CR nodes is shown in Fig. 2 (b). In Fig. 2 (b), CR_k starts to sense channel *i* before CR_j. Let $\tilde{t}_{k,j}$ be the difference of start-sensing time of CR_j and CR_k. If $\tilde{W}_{i,j}+\tilde{t}_{k,j}>\tilde{W}_{i,k}$, CR_j fails to transmits frames on channel *i*. We then consider another case that CR_l starts to sense channel *i* after CR_j. Similarly, if $\tilde{W}_{i,j}>\tilde{W}_{i,l}+\tilde{t}_{j,l}$, CR_j also fails to deliver frames on channel *i*. Therefore, the probability that CR_j transmits on channel *i* without forestalling by other CR nodes is $p_2=1-P(\tilde{W}_{i,j}+\tilde{t}_{k,j}>\tilde{W}_{i,k})-P(\tilde{W}_{i,j}>\tilde{W}_{i,l}+\tilde{t}_{j,l})$.



(a). An example to illustrate the impact of PUs on CR_j



(b) An example to illustrate the impact of other CR nodes on CR_j

Figure 2. An illustration of CR_j accessing channel *i* successfully

The objective of this paper is, for each CR node, to derive $p_1\times p_2$ value for all channels.

III. Cloud Server-Assisted MAC (CSA-MAC) Protocol for Cognitive Radio Networks

In this section, we describe the designed cloud server-assisted medium access control protocol, named CSA-MAC, in detail.

In CSA-MAC, each CR node, say CR_j, maintains/updates a channel preference matrix $\boldsymbol{H}_j=[\boldsymbol{P}_j\ \boldsymbol{W}_j\ \boldsymbol{Q}_j]$ when periodically receiving channel statistics from the AP_{CR}. $\boldsymbol{H_j}$ is an *N*×3 matrix, as shown in (1). Each row is for a specific data channel; the three elements of a row are successful and data transmission probability, sensing time, and transmission quota. Here $p_{i,j}$, $\tilde{W}_{i,j}$, and $q_{i,j}$ represent the successful transmission probability, channel sensing time, and transmission quota, of CR_j on channel *i*, respectively.

$$\boldsymbol{H_j}=[\boldsymbol{P_j}\quad \boldsymbol{W_j}\quad \boldsymbol{Q_j}]=\begin{bmatrix} p_{1,j} & \tilde{W}_{1,j} & q_{1,j} \\ p_{2,j} & \tilde{W}_{2,j} & q_{2,j} \\ \vdots & \ddots & \\ p_{N,j} & \tilde{W}_{N,j} & q_{N,j} \end{bmatrix} \tag{1}$$

In the following, we explain how to derive column vectors $\boldsymbol{W_j}$, $\boldsymbol{Q_j}$, and $\boldsymbol{P_j}$.

A. Derivations of $\boldsymbol{W_j}$ and $\boldsymbol{Q_j}$

Based on [8], the sensing time of CR_j on channel *i*, denoted as $\tilde{W}_{i,j}$, is

$$\tilde{W}_{i,j} = \frac{1}{B_i \gamma_{i,j}^2}\left[1.2 + 1.2(\gamma_{i,j}+1)\right]^2, \qquad (2)$$

where B_i and $\gamma_{i,j}$ are channel capacity (in Hz) and measured signal-to-noise ratio (SNR) (in dB), respectively. Thus the sensing time of CR_j on each channel is $\boldsymbol{W_j} = \left[\tilde{W}_{1,j}, \tilde{W}_{2,j}, \ldots, \tilde{W}_{N,j}\right]^T$.

Next, let $q_{i,j}$, t_x and t_{ctl} indicate the transmission quota of CR_j on channel *i*, frame transmission time, and control frame transmission time, accordingly. Given the idle time of channel *i* being *r*, i.e., $\tilde{Z}_i = r$, the time duration that CR_j can utilize to transmit frames is $(\tilde{Z}_i - \tilde{W}_{i,j})$, which can accommodate $\left\lfloor \frac{\tilde{Z}_i - \tilde{W}_{i,j} - t_{ctl}}{t_x} \right\rfloor$ data frames. Thus the column matrix $\boldsymbol{Q_j}$ is

$$\boldsymbol{Q_j} = [q_{1,j}, \ldots, q_{N,j}]^T$$

$$= \left[\left\lfloor \frac{\tilde{Z}_1 - \tilde{W}_{1,j} - t_{ctl}}{t_x} \right\rfloor, \ldots, \left\lfloor \frac{\tilde{Z}_N - \tilde{W}_{N,j} - t_{ctl}}{t_x} \right\rfloor\right]^T \qquad (3)$$

B. Derivation of $\boldsymbol{P_j}$

Both PUs and other CR nodes affect data transmission of CR_j on channel *i*. Therefore our derivations consist of two parts: impact from PUs and impact from CR nodes.

(1) Impact from PUs

We assume the idle time of channel *i* is a random distribution $\tilde{Z}_i$ and for a specific period *k* its distribution is

$$f_{\tilde{Z}_{i_k}} = \frac{r f_{\tilde{Z}}(r)}{E[\tilde{Z}]} \qquad (4)$$

Let $\tilde{S}_{i,j} = \tilde{W}_{i,j} + \tilde{X}_{i,j}$. According to the imbedded Markov chain [9], we can find the occupancy distribution of a CR node by applying z-transform on (4),

$$\begin{aligned} F^*_{\tilde{S}_{i,j}}(s) &= \frac{(1-z)(1-\rho)F^*_{\tilde{x}_{i,j}}(\lambda_c - \lambda_c z)}{F^*_{\tilde{x}_{i,j}}(\lambda_c - \lambda_c z) - z}\Big|_{z=\frac{\lambda_c - s}{\lambda_c}} \\ &= \frac{(1-\rho)F^*_{\tilde{x}_{i,j}}(s)}{1-\rho[1-F^*_{\tilde{x}_{i,j}}(s)]/(sE[\tilde{x}_{i,j}])}, \end{aligned} \qquad (5)$$

where λ_c is the arrival rate of CR nodes, μ is channel service rate, and $\rho = \lambda_c/\mu$.

We use *M*/*M*/1 as an example to further explain how to derive $F^*_{\tilde{S}_{i,j}}(s)$. Assume $F^*_{\tilde{x}_{i,j}}(s) = \frac{\mu}{\mu+s}$, then

$$\begin{aligned} F^*_{\tilde{S}_{i,j}}(s) &= \frac{s(1-\rho)\frac{\mu}{\mu+s}}{s - \lambda_c + \lambda_c\frac{\mu}{\mu+s}} \\ &= \frac{\mu(1-\rho)}{s+\mu(1-\rho)} \end{aligned} \qquad (6)$$

The probability density function of $\tilde{S}_{i,j}$ is $f_{\tilde{S}_{i,j}}(t) = \mu(1-\rho)e^{-\mu(1-\rho)t},\ t \geq 0$. Let λ_{p_i} be the PUs' arrival rate on channel *i*. The successful transmission

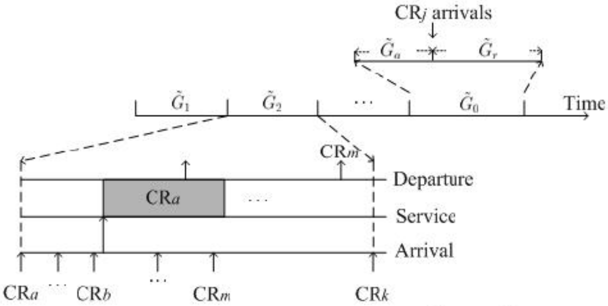


Figure 3. An example to illustrate the meanings of $\{\tilde{G}_\eta, \eta \geq 1\}$, aging time $\tilde{G}_a$, and residual time $\tilde{G}_r$

probability of CR_j on channel *i* without PU interruption is $P(\tilde{Z}_{i_k} \geq \tilde{S}_{i,j})$ and

$$\begin{aligned} P(\tilde{Z}_{i_k} \geq \tilde{S}_{i,j}) &= \int_r^\infty P(\tilde{Z}_{i_k} \geq r|\tilde{S}_{i,j} = r) f_{\tilde{Z}_{i_k}(r)} dr \\ &= \int_r^\infty \int_0^r \mu(1-\rho)e^{-\mu(1-\rho)t} f_{\tilde{Z}_{i_k}}(r) dt dr \\ &= \int_r^\infty f_{\tilde{Z}_{i_k}}(r)dr - \int_r^\infty e^{-\mu(1-\rho)r} f_{\tilde{Z}_{i_k}}(r)dr \\ &= e^{-\lambda_{p_i} r} - \frac{\lambda_{p_i}}{\lambda_{p_i} + \mu - \lambda_c}(e^{-(\lambda_{p_i}+\mu-\lambda_c)r}) \end{aligned} \qquad (7)$$

From (7), if $\tilde{Z}_{i_k} = r \geq \tilde{S}_{i,j}$, channel *i* has available capacity to serve CR_j without interfering PUs. Further, if the channel idle time is exactly the sum of sensing time and frame transmission time, i.e., $r = \tilde{W}_{i,j} + \tilde{X}_{i,j},\ i = 1, 2, \ldots, N$, CR nodes maximally utilize the channel idle time.

(2) Impact from other CR nodes

Let $\{\tilde{G}_\eta, \eta \geq 1\}$ denote a sequence of i.i.d. non-negative random variables with $P(\tilde{G}_\eta \geq 0), \forall \eta$. Here $\tilde{G}_\eta$ is the time interval which a CR user enters this channel which is unoccupied until the time which channel becomes unoccupied. An example is shown in Fig. 3. CR_a starts to sense an idle channel during $\tilde{G}_2$, and there are other CR nodes hop to that channel to perform sensing. While all CR users finish their transmission or hop to other data channels pending, we call the period $\tilde{G}_2$ is finished.

We assume that CR_j starts to sense channel during $\tilde{G}_0$. Since the start-sensing time is randomly distributed within $\tilde{G}_0$, we divide $\tilde{G}_0$ into two parts: before CR_j's start-sensing time (named aged time $\tilde{G}_a$), and after CR_j's start-sensing time (named residual time $\tilde{G}_r$). We know $f_{\tilde{G}_0} = \frac{g f_{\tilde{G}}(g)}{E[\tilde{G}]}$. Therefore,

$$\begin{aligned} f_{\tilde{G}_r}(p) &= \int_{g=p}^\infty f_{\tilde{G}_r|\tilde{G}_0}(p|\tilde{G}_0 = g) dF_{\tilde{G}_0}(g) \\ &= \frac{1}{E[\tilde{G}]}\int_p^\infty f_{\tilde{G}}(g)dg \\ &= \frac{1 - F_{\tilde{G}}(p)}{E[\tilde{G}]} \end{aligned} \qquad (8)$$

Again, in Fig. 2 (b), two criteria that CR_j can successfully transmit frames on that channel without forestalling other CR nodes' transmission are (1) $\tilde{t}_{k,j} \leq \tilde{W}_{i,k} - \tilde{W}_{i,j}$, and (2) $\tilde{t}_{j,l} \geq \tilde{W}_{i,j} - \tilde{W}_{i,l}$. Let $x = \tilde{t}_{j,l} = \tilde{t}_{k,j}$, $\zeta_i = \tilde{W}_{i,j} - \tilde{W}_{i,l}$ and $\psi_i = \tilde{W}_{i,k} - \tilde{W}_{i,j}$, respectively, then the successful probability is

$$\begin{aligned} f_{\tilde{G}_a}(\zeta_i \leq x \leq \psi_i) &= \int_0^{\psi_i} f_{\tilde{G}_a}(x)dx - \int_0^{\zeta_i} f_{\tilde{G}_a}(x)dx \\ &= F_{\tilde{G}_a}(\psi_i) - F_{\tilde{G}_a}(\zeta_i) \end{aligned} \tag{9}$$

The column vector $\boldsymbol{P_j}$ is

$$\boldsymbol{P_j} = [p_{1,j}, p_{2,j}, \ldots p_{N,j}]^T$$

$$= \begin{bmatrix} (F_{\tilde{G}_a}(\psi_1) - F_{\tilde{G}_a}(\zeta_1))(e^{-\lambda_{p_1} r} - \frac{\lambda_{p_1}}{\lambda_{p_1}+\mu-\lambda_c}(e^{-(\lambda_{p_1}+\mu-\lambda_c)r}) \\ (F_{\tilde{G}_a}(\psi_2) - F_{\tilde{G}_a}(\zeta_2))(e^{-\lambda_{p_2} r} - \frac{\lambda_{p_2}}{\lambda_{p_2}+\mu-\lambda_c}(e^{-(\lambda_{p_2}+\mu-\lambda_c)r}) \\ \vdots \\ (F_{\tilde{G}_a}(\psi_i) - F_{\tilde{G}_a}(\zeta_i))(e^{-\lambda_{p_i} r} - \frac{\lambda_{p_i}}{\lambda_{p_i}+\mu-\lambda_c}(e^{-(\lambda_{p_i}+\mu-\lambda_c)r}) \\ \vdots \\ (F_{\tilde{G}_a}(\psi_N) - F_{\tilde{G}_a}(\zeta_N))(e^{-\lambda_{p_N} r} - \frac{\lambda_{p_N}}{\lambda_{p_N}+\mu-\lambda_c}(e^{-(\lambda_{p_N}+\mu-\lambda_c)r}) \end{bmatrix} \tag{10}$$

C. CSA-MAC operations

In this paper, we propose two MAC protocols: CSA-MAC with handshaking and CSA-MAC without handshaking. For a CR pair, the sender (say CR_j) transmits an invitation to its intended receiver (say CR_k) on the control channel. If CR_k is idle and within CR_j's transmission range, it replies its channel preference matrix $\boldsymbol{H_k}$ to CR_j. CR_j is responsible to determine the channel preferences. How to determine the channel preferences is described below.

Upon receiving $\boldsymbol{H_k}$, for this CR flow, CR_j calculates the successful transmission probability of each channel, sensing time, and transmission quota, as in (11).

$$\begin{cases} p_i = p_{i,j} \times p_{i,k}, \\ \tilde{W}_i = max(\tilde{W}_{i,j}\ ,\ \tilde{W}_{i,k}), \quad i = 1, 2, ..., M \\ q_i = min(q_{i,j}\ ,\ q_{i,k}), \end{cases} \tag{11}$$

CR_j then sorts all data channels in decreasing order of p_i. This sorted channel sequence is exact the hopping sequence. CR_j informs CR_k the hopping sequence and the corresponding sensing time and transmission quota. Followed, both CR_j and CR_k hop to data channel(s) for channel sensing.

The major difference between CSA-MAC with handshaking and without handshaking is the exchanges of RTS_{CR} and CTS_{CR} on data channels. For CSA-MAC with handshaking, CR_j and CR_k will further exchange RTS_{CR} and CTS_{CR} when either side senses a data channel being idle; while CSA-MAC without handshaking does not perform RTS_{CR} and CTS_{CR} exchanges.

In the following, we use an example to illustrate CSA-MAC protocol. We consider two data channels (denoted as Ch1 and Ch2), and each is with 2MHz capacity with BPSK modulation scheme. CR_B wants to transmit frames to CR_C, as shown in Fig. 1. We assume 2048-byte frame size, and the frame transmission time is 8.4ms. The channel utilization of PUs of Ch1 and Ch2 are 0.4 and 0.5 individually.

Table I. The successful probability of derivation and simulation results in CRN.

N=1	CR user = 1	Derivation result	0.8625
		Simulation result	0.8000
	CR user = 2	Derivation result	0.9472
		Simulation result	0.8791
N=2	CR user = 1	Derivation result	0.7350
		Simulation result	0.7062
	CR user = 2	Derivation result	0.9058
		Simulation result	0.8136

The SNR values of Ch1 and Ch2 measured by CR_B are 0.0246 dB and 0.0231 dB individually. Thus the sensing times of Ch1 and Ch2 are $\tilde{W}_{1,B} = 1.3\text{ms}$, and $\tilde{W}_{2,B} = 1.4\text{ms}$. Assume that the observation time of Ch1 and Ch2 is 90ms and 86ms. According to (3), $q_{1,B} = \lfloor 0.6(90-1.3)/8.44 \rfloor = 6$, and $q_{2,B} = \lfloor 0.5(86-1.4)/8.44 \rfloor = 5$. Upon obtaining both transmission quota and sensing time, we can further calculate $\boldsymbol{r} = [1.3 + 6 \times 8.44, 1.4 + 5 \times 8.44]^T = [51.94, 48.6]^T$, and $q = [0.4707, 0.3316]^T$. Let $\tilde{G}_n$ be in lognormal distribution. By substituting all results into (9), the successful transmission probabilities that CR_B does not forestall other CR nodes' transmission on Ch1 and Ch2 are 0.1524 and 0.1241, respectively. Considering both impacts from PUs and other CR nodes, the successful transmission probability of CR_B is $\boldsymbol{p}_B = [0.07173, 0.07173]^T$. Finally, $\boldsymbol{H}_B = \begin{bmatrix} 0.07173 & 1.3 & 6 \\ 0.04115 & 1.4 & 5 \end{bmatrix}$.

CR_C performs similar operations, while the viewed PU traffic loads and measured channel qualities on Ch1 and Ch2 are (0.3, 0.0231 dB) and (0.4, 0.0246 dB) individually. Thus its $\boldsymbol{H}_C = \begin{bmatrix} 0.07612 & 1.4 & 8 \\ 0.07173 & 1.3 & 6 \end{bmatrix}$. Furthermore, the successful transmission probabilities of Ch1 and Ch2 are 0.00546 and 0.00295, respectively. Note that in this example, both channels have the same sensing time (which is 1.4 ms) and transmission quota (which is 6). As a result, the hopping sequence of this CR pair is (Ch1, Ch2).

D. Model validation

We validate the derivation of successful transmission probability with simulation results. we assume CR nodes are always backlogged. In the simulation experiment, the mean and standard deviation of PUs' traffic load are 0.5 and 0.1, respectively. The comparison is summarized in Table I. There exists discrepancy between the derivation and simulation results, which is due to the setting of standard deviation. In our derivation, a CR node uses the mean traffic load value of PUs to estimate the corresponding successful transmission probability. However, in simulation experiment, channel idle time maybe cannot accommodate q_i frames, i=1, 2. In such a situation, PUs should wait for transmission completion. Those events are not counted in the calculation of the successful transmission probability. Thus the successful probability of simulation result is smaller than that of derivation. One significant achievement of our mechanism is that CR nodes utilize at least 70% of the channel idle time.

IV. Performance Evaluation

In this section, we develop a simulation program to compare the performance of the designed CACS mechanism with OSA-MAC [2], SSA-MAC [3], CH-MAC [4], and DRA-MAC [5].

In this experiment, there are one control channel, and five data channel. The PU traffic load on data channel i, $i = 1, 2, \ldots 5$, is poisson distribution with rate λ_{p_i}. Moreover, we set $\lambda_{p_1} = \lambda_{p_2} = 0.4$; $\lambda_{p_3} = 0.5$, and $\lambda_{p_4} = \lambda_{p_5} = 0.6$. CR nodes are always backlogged. The bandwidth of a data channel is 2 Mbps. Frame size is 2048 bytes. The transmission ranges of PUs, CR nodes, and CR APs are 150 meters, 100 meters, and 100 meters, respectively. The duration of DIFS and SIFS is 0.05 and 0.01 ms, accordingly. For SSA-MAC, the settings of TXQ and RTV are 4 and 1, respectively. The simulation time is 100 seconds. The observed performance metrics include "utilization of channel idle time", and "average tries of channel search".

We first investigate the utilization of channel idle time of various mechanisms, and the results are shown in Fig. 4. We found that CSA-MAC (with handshaking) performs better than other MAC protocols. The reasons have twofold: setting transmission quota according to PUs' traffic load; and adapting channel sensing time based on measured channel quality. As a result, CR nodes utilize channel idle time as much as possible. The performance gap between CSA-MAC with handshaking and without handshaking is caused by different dwell time when sensing a busy channel. Indeed, the dwell time for CSA-MAC with handshaking is $\tilde{W}_i + t_{ctl}$, while it's $\tilde{W}_i + \tilde{X}_i$ for CSA-MAC without handshaking. The reason of low utilization for OSA-MAC is that a CR pair only exchanges one data frame when occupying a data channel. Moreover, the common drawback of DRA-MAC and CH-MAC is that if being aware of PU presence on the sensed data channel, CR nodes will stay at that channel for five slots, thus resulting in low utilization. SSA-MAC has a mechanism for PUs to interrupt CR transmission. Thus, SSA-MAC performs worse than CSA-MAC (with handshaking).

Next, the performance of the average tries of channel search for various mechanisms is in Fig. 5. It is common for all mechanisms that, when the number of CR pairs increases, the average tries of channel search also increases. Besides, CSA-MAC (with handshaking) outperforms CH-MAC and DRA-MAC. The reason is, in CH-MAC and DRA-MAC, a CR sender does not select channels according to PUs' traffic loads, and thus may frequently sense busy channels. Besides, comparing with random hopping sequence performed in SSA-MAC, our estimation of successful transmission probability makes a great impact when there are more than five CR pairs. In OSA-MAC, a CR sender only sense once during a fixed period. Thus, OSA-MAC has the least tries of channel search among all mechanisms, while its drawback is low utilization of channel idle time as previously discussed. Note that CSA (without handshaking) still performs better than most compared protocols. The reason is that a CR pair has to wait for $\tilde{W}_i + \tilde{X}_i$ when sensing a busy channel, which implies that CSA-MAC (without handshaking) has relative long sensing time.

V. Conclusions

In this paper, we proposed a cloud server-assisted MAC protocol, named CSA-MAC, for infrastructure-based cognitive radio networks. In CSA-MAC, each CR nodes maintains a channel preference matrix, which records the successful

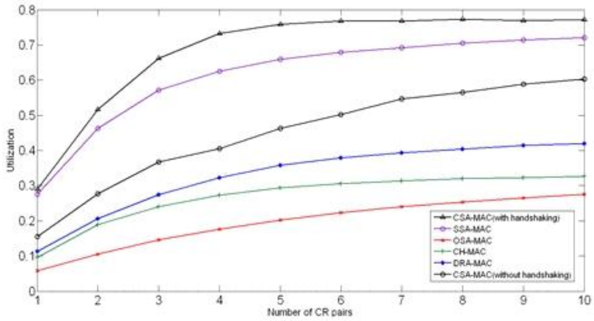


Figure 4. The utilization of channel idle time v.s. the number of CR pairs.

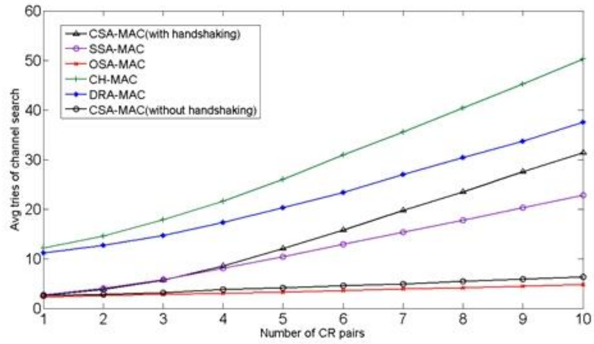


Figure 5. The average tries of channel search v.s. the number of CR pairs

transmission probability, sensing time, and transmission quota, of each data channel. The three parameters are derived through an analytical queueing model, and the support of powerful cloud servers. Two versions of CSA-MAC are presented and compared in this paper, with handshaking and without handshaking. The simulation results showed that CSA-MAC with handshaking performs better in the utilization of channel idle time, while CSA-MAC without handshaking diminishes the average tries of channel search. In the future, we will investigate the impact of different arrival rate of CR users and extend this work to multi-hop CR flows.

Acknowledgement

This work was supported in part by National Science Council under grants NSC 99-3113-P-009-004 and NSC 100-2219-E-009-005, and in part by the Information and Communications Research Laboratories (ICL), Industrial Technology Research Institute (ITRI), Taiwan, under grant A352BW2100.

References

[1] FCC Spectrum Policy Task Force, "Report of the Spectrum Efficiency Working Group," FCC, Technical Report 02-155, 2002.

[2] L. Long and E. Hossain, "OSA-MAC: A MAC Protocol for Opportunistic Spectrum Access in Cognitive Radio Networks," in Proceedings of IEEE WCNC, 2008.

[3] T.C. Lou, H.L Chao, and T.J Tsai, "On Synchronized Channel Sensing and Accessing for Cognitive Radio Users in IEEE 802.11 Wireless Networks," in Proc of IEEE PIMRC, 2009.

[4] H. Su, and X. Zhang, "Channel-hopping Based Single Transceiver MAC for Cognitive Radio Networks," in Proc. of IEEE CISS, 2008.

[5] L. Jiao, and F.-Y. Li, "A Single Radio Based Channel Datarate-aware Parallel Rendezvous MAC Protocol for Cognitive Radio Networks," in Proc. of IEEE LCN, 2009.

[6] IEEE 802 LAN/MAN Standards Committee 802.22 WG on WRANs. http://www.ieee802.org/22.

[7] D-H. Huang, S.-H. Wu, and P.-H. Wang, "Cooperative Spectrum Sensing and Locationing: A Sparse Bayesian Learning Approach," in Proc. of IEEE GLOBECOM, 2010.

[8] W-Y Lee, and I.F. Akyildiz, "Optimal Spectrum Sensing Framework for Cognitive Radio Network," IEEE Wireless Communications, vol. 7, pp. 3845-3857, Oct. 2008.

[9] Kleinrock, L., "Queueing systems. Vol. II: Computer applications," John. Wiley & Sons, New York, 1976.

附錄三

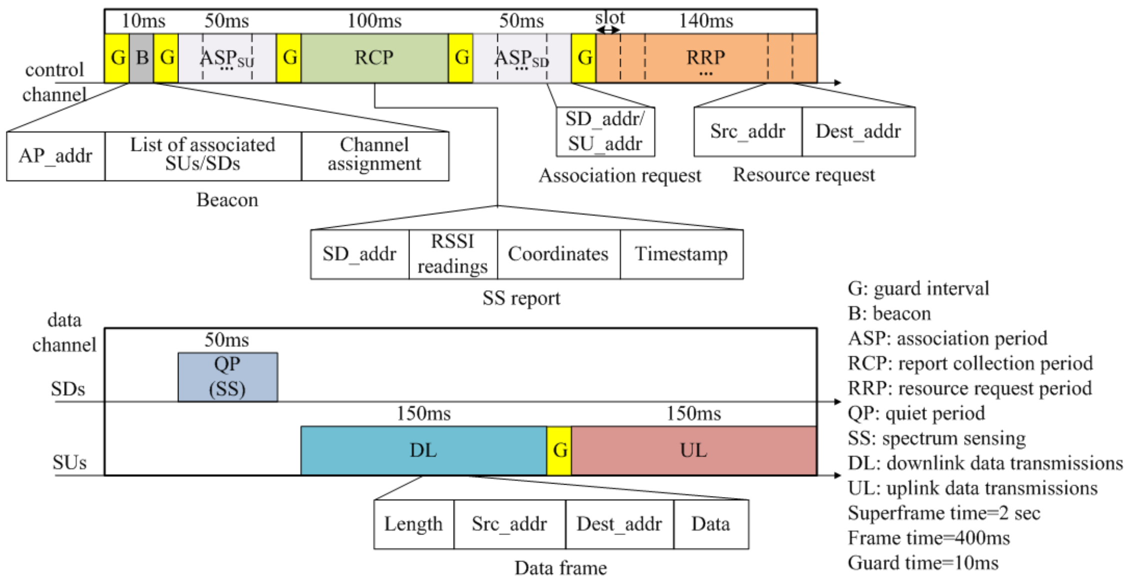


圖 Framing and five defined frame formats

我們在 infrastructure-based CR network 中設計並且實作一個 FD-TDMA based CR-MAC protocol。該 CR-MAC protocol 能夠在一個建置於 TVWS 的 CRCN prototype 內運行，並支援 CSS 與 channel access 的能力。

Framing and Frame Formats

在我們所提出的 CR-MAC protocol 中，我們將時間分為多個 frame，並由數個 frame 構成一個 superframe。在控制信號頻帶上每個 frame 包含以下幾個區段：Beacon (B), Association Period for SUs (ASP_{SU}), Report Collection Period (RCP), Association Period for SDs (ASP_{SD}), and Resource Request Period (RRP). 在我們的實作中，考量硬體的限制與同步的準確性，我們插入一個 10ms 的 guard time 在連續的兩個區段之中。而在 data channel 上每個 frame 包含以下幾個區段： Quiet Period (QP), DownLink data transmissions (DL), and UpLink data transmissions (UL). 同樣的我們在 UL 與 DL 之中插入一個 10ms guard time。

在我們的設計中，CR AP 會定期的在控制信號頻帶上 broadcast beacon 去通知使用者 CR AP 的存在並且管理頻帶的使用，而 SU 與 SD 都必須在存取頻帶使用前與感測頻帶前加入 CR AP，並且與 CR AP 時間同步。而 beacon 包含三部分：CR AP address ($Addr_{AP}$), the list of associated SUs and SDs, and data channel assignments。而 ASP_{SU} 與 ASP_{SD} 分為多個 slots，能夠分別讓 SU 與 SD 在對應時間內以 contention-based 的機制來傳送註冊請求給 CR AP。

在 QP 時，已經加入網域的所有 SDs 會在 QP 同時且依序地對所有的 data channels 進行感測，此時已加入的 SUs 不能使用 data channels 傳送資料。之後 CR AP 使用 RCP 來 poll 加入的 SDs 回報頻譜感測資料，而回報的資料包含：RSSI readings of data channels, coordinates, and timestamp。而在頻帶可以使用的情況下，SU 可以在 UL 發送 data packet 給 CR AP，其中包含：packet length in byte (Length), source SU address (Src_addr), destination SU address (Dest_addr), and data payload (Data)。而 CR AP 也可以在 DL 發送 data packet 給對應的目的端。

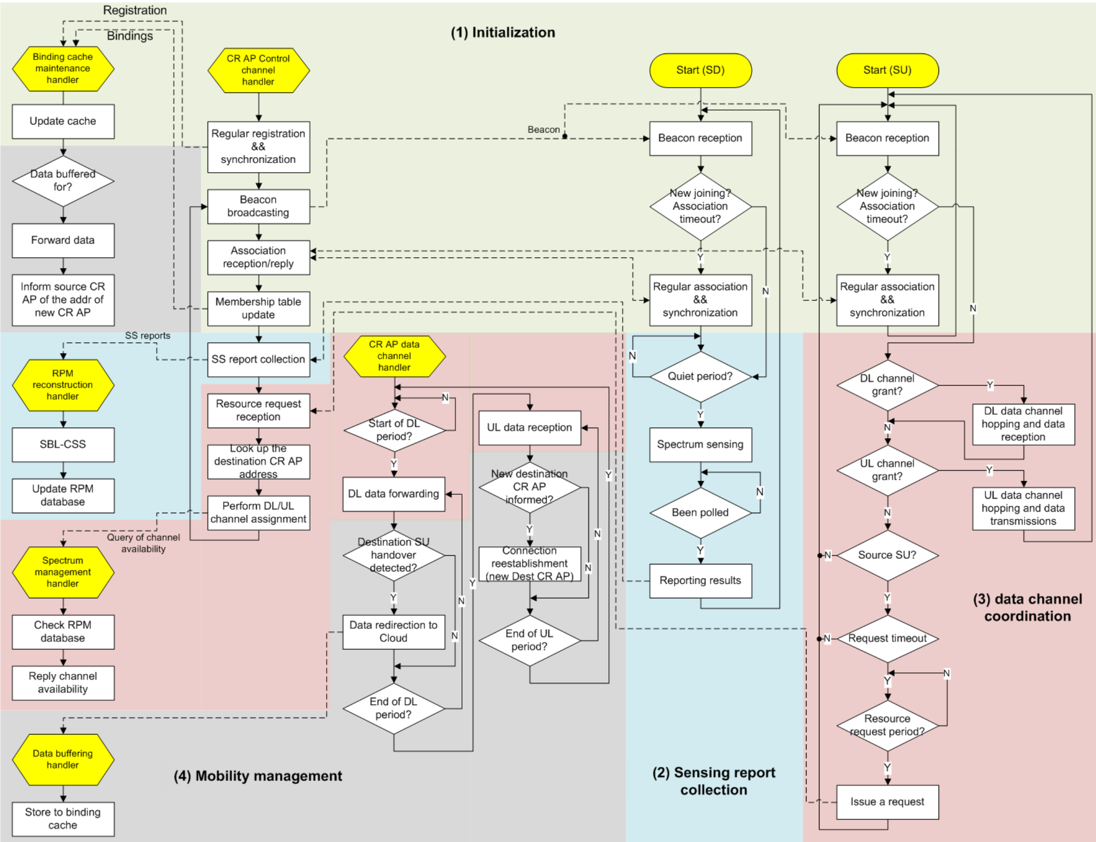


圖 The flowchart of the designed Cloud-based CR-MAC protocol

Protocol Operation

在所提出的網路層協定提案中，我們的流程包含了四個部分：(1) Initialization，AP、SU、SD 的註冊與時間同步控制 (2) Sensing report collection，SD 在 QP 感 測頻帶並透過 AP polling 來回報 (3) Data channel coordination，由 AP 以演算法排 程結果來協調 SU 使用頻帶的情況 (4) Mobility management，透過雲端與 AP 來 管理 SU 的 mobility。在接下來將對各部份進行詳述的介紹。

1. Initialization

在此階段，CR APs、SUs、SDs 會先啟動註冊與時間同步的機制。CR AP 會先送出註冊請求到雲端的 home agent。當 home agent 收到註冊請求時，會更新其對應的 binging cache 並且回應 CR AP 註冊成功與時間標籤。而 CR AP 收到註冊成功訊息並根據時間標籤進行系統時間的校正之後，就能夠開始廣播 beacon 並

且服務 SUs。

對於 SU 與 SD 來說，會先在控制信號頻帶上聆聽 CR AP 的 beacon 後，在對應的時間 ASP $_{SU}$ 與 ASP $_{SD}$ 送出註冊請求給 CR AP。同樣的，CR AP 會維護目前網域內的 SUs 與 SDs 的 membership table，並且將 membership table 中的成員註冊到雲端的 home agent。因此在 home agent 中，會記錄 CR AP 與其所管理的使用者資訊在 binding cache 中，以便於未來 home agent 在於使用者 Communication 與 Mobility management 的維護上。SU、SD 在發出註冊請求後，透過聆聽下一次 beacon 中 associated list of SUs and SDs 來判斷是否有順利地註冊成功，否則就會持續的發送註冊請求給 CR AP。在 SU 順利註冊成功後，即可以發出請求頻帶使用的請求，之後透過 CR AP 的管理跳至對應的頻帶傳送資料。而 SD 順利註冊成功後，即在對應的頻帶進行感測並且透過 AP polling 來回報。

對於 home agent 的 binding cache 與 CR AP 的 membership table 採取 ”soft - state ” 的方式，會因應網路拓樸的改變而動態改變 membership，因此對於 CR APs、SUs、SDs 都需定期的發送註冊請求以刷新使用者狀態，而 CR AP 也會定期的確認使用者目前的狀態來判斷是否使用者正在使用頻帶或是已經離開該網域。

2. Sensing report collection

在此階段，已經加入網域的所有 SDs 會在 QP 同時且依序地對所有的 data channels 進行感測，同時已加入的 SUs 不能使用 data channels 傳送資料，而新加入的 SUs 則可在控制信號頻帶傳送 join 的請求。在 SDs 感測之後，在 RCP 所有 SDs 會跳回控制信號頻帶來等待 CR AP 的 polling 以便回報感測資料。透過 CR AP polling 來讓感測資料能夠準確回報，以期讓之後的頻譜估測能有更高的準確性。在所有 SDs 回報之後，CR AP 透過 Internet 來將所有感測資料回報雲端，雲端啟動 CSS Engine 來進行頻譜估測並且更新頻譜估測結果。

3. Data channel coordination

SU 在加入網域之後，並不能夠馬上得到頻帶的使用權。當 SU 需要資料傳送時，會先在控制信號頻道聆聽 beacon，beacon 中的 channel assignment 數值會依據雲端與 CR AP 的演算法排程結果來給定，而 SU 透過 channel assignment 來跳至對應的頻帶上傳送資料。而當 PU 出現，channel assignment 會填入沒有頻帶可以使用的訊息，因此當 SU 接收到該 beacon 後，會先將資料 buffer 在使用者端直

到 PU 離開之後，才會重新跳至對應的 data channel 傳送資料。

4. Mobility management

隨著使用者的移動，使用者會加入到另個 CR AP(新 CR AP)，但對於使用者原先所在的 CR AP(舊 CR AP)而言，並不知道該個使用者已經離開其所管理的區域，並且在此時所有傳送給該使用者的資料都會遺失。

為了因應上述情形，我們採取 ” soft - state ” 的 membership 維護，CR AP 可以透過定期的檢查 membership table 來確認使用者是否離開。若使用者已經離開但此時仍有資料欲透過 CR AP 傳送給該名使用者，此時 CR AP 會將該筆資料傳送給 home agent，而 home agent 會幫使用者 buffer 資料直到 timeout。當使用者重新加入另個 CR AP 時，home agent 會在 CR AP 註冊使用者的請求的時候，同時判斷是否有該使用者的資料，若有則進行資料傳送。之後若有 CR AP 向 home agent 詢問該使用者所在位置，home agent 會通知使用者新註冊的 CR AP 位址。

To appear in IEEE WCNC 2012, 1-4 April 2012, Paris, France

附錄四

Optimizing the Cloud Platform Performance for Supporting Large-Scale Cognitive Radio Networks

Shie-Yuan Wang, Po-Fan Wang, and Pi-Yang Chen
Department of Computer Science
National Chiao Tung University, Taiwan
Email: {shieyuan, bfwang, bychen}@cs.nctu.edu.tw

***Abstract*—In this paper, we optimize the performance of a cloud platform to effectively support cooperative spectrum sensing in a cognitive radio (CR) cloud network. This cloud uses the Apache Hadoop platform to run a cooperative spectrum sensing algorithm in parallel over multiple servers in the cloud. A cooperative spectrum sensing algorithm needs to process a very large number of spectrum sensing reports per second to quickly update its database that stores the current activities of all primary users of the CR network. Because the updates of the database must be finished as soon as possible to make the CR approach effective, the cloud platform must be able to run the algorithm in real time with as little overhead as possible. In this work, we first measured the execution time of such an algorithm over our own cloud and the Amazon EC2 public cloud, using the original Hadoop platform design and implementation. We found that the original Hadoop platform has too much fixed overhead and incurs too much delay to the cooperative spectrum sensing algorithm, which makes it unable to update the primary user database in just a few seconds. Therefore, we studied the source code and the design and implementation of the Hadoop platform to improve its performance. Our experimental results show that our improvement of the Hadoop platform can significantly reduce the required time of the cooperative spectrum sensing algorithm and make it more suitable for large-scale CR networks.**

I. Introduction

Recently, the concept of cognitive radio (CR), which was first introduced by Joseph Mitola III [1], has become more and more popular and important due to the limitation of wireless bandwidth. In [2], the authors show that the usage of licensed spectrum is less than 25%, which is very inefficient. In a CR network, the secondary/unlicensed users (SUs) are allowed to use the empty spectra in frequency, time and space under the constraints of not interfering with the primary/licensed users (PUs). The CR approach has a great potential to improve the utilization of licensed spectrum.

The Federal Communication Commission (FCC) of the U.S. has approved unlicensed radio transmitters to operate in the broadcast television spectrum at locations where that spectrum is not being used by licensed services (this unused TV spectrum is often termed as ”white spaces”) [3] and has granted field trials of CR networks. With this opportunity, many international organizations have defined CR standards on TV white spaces (TVWS), such as IEEE 802.22, IEEE 1900, IEEE 802.16m and ECMA 392.

To avoid a SU from interfering with the signal quality of TV sets, the FCC requires that CR operators in TVWS be able to detect the TV signal. In addition, the CR operators should also provide a database that maintains the geographical locations of TV base stations (BS) and their radiation powers, antenna heights, and numbers of channels, etc. To help achieve these goals of spectrum sensing in TVWS, the SUs of a CR network in TVWS are suggested to provide their sensing data and geographical locations for the CR operators to perform cooperative spectrum sensing. Cooperative spectrum sensing will reconstruct the power propagation map (PPM) and periodically update the PPM in a database.

A SU may operate either as a Mode I device (which operates only on the channels identified by either a fixed device or a Mode II personal/portable device) or as a Mode II device (which relies on geo-location and database access to determine available channels at its location) [3]. Due to the variation of time and space, TV sets and wireless microphones may occasionally be turned on to use the TV spectrum that was previously unused. If the SUs and database cannot quickly discover that PUs have become active and are using their allocated spectrum, a severe interference between PUs and SUs might occur. FCC requires that a Mode II personal/portable device check the database at least one time in 24 hours to ensure the availability of spectrum. If it cannot reach the database in a day, it can only operate after the end of the day. On the other hand, a Mode II device has to check its own location every 60 seconds. If it moves across a distance more than 100 meters from the location where it lastly checked into the database, a reconfirmation to the database is required. A Mode I device has to check the availability of spectrum every 60 seconds through Mode II devices. The above rules are set up for avoiding the interference between PUs and SUs. The information of PUs database plays a very important role in the TVWS CR network. The reconstruction of PPM and the update of PUs information in the database must be done as fast as possible. Real-time updates of the PPM database based on periodic sensing data from SUs can greatly reduce the interference between PUs and SUs.

In [4], the authors proposed a Cognitive Radio Cloud Network (CRCN) architecture to address the needs of a CR network. Their cloud used the public Windows Azure cloud as the computing platform to execute the spectrum sensing (SS) algorithm. They implemented the sparse Bayesian learning (SBL) [5] algorithm for cooperative SS using a MapReduce-like method over the SQL Azure and Windows Azure. Since the computation complexity of the SBL algorithm grows in the order of 3 by the number of SUs, they implemented a hierarchical parallelization method with Microsofts dotNet 4.0 using a MapReduce-like programming model to reduce the execution time.

Despite their efforts, their experimental results show that when the measurement rate is 0.15 (their definition of measurement rate is the number of SUs divided by the region in units of K m^2), the SBL algorithm still needs 24.5 seconds to finish over Window Azure, which is far from meeting the real-time requirement of a CR network. Therefore, in this paper we focus on reducing the execution time of the SBL algorithm to only a few seconds to make the CR approach more effective. To do so, we improved the design and implementation of the Apache Hadoop platform [6] and had successfully reduced the execution time of the SBL algorithm to only 5.89 seconds under the 0.15 measurement rate.

The rest of the paper is organized as follows. In section II, we briefly introduce Hadoop, the reason why we chose to use it as the CRCN computing platform, and the problem we encountered with it. In section III, we present our study results about Hadoop and propose solutions to these encountered problems. Experimental settings are presented in section IV and various experimental results are then presented in section V. Finally, we conclude the paper in section VI.

II. Using Hadoop for CRCN

A. Background

Hadoop is an open source project composed of Hadoop MapReduce, an implementation of MapReduce designed for large clusters, and Hadoop Distribution File System (HDFS), a file system that provides high-throughput access to application data. It allows users to process a large data set with distributed processing without fully knowing the knowledge of distributed computing and without using expensive computing servers. Hadoop is compatible with Hadoop database (HBase) [7], which can perform random, real-time read/write accesses to Big Data. HBase is suitable for the CRCN database, which needs to store a very large number of sensing data from SUs.

A Hadoop system consists of a single master node and many worker nodes. The master, called the Job-Tracker, is responsible for accepting jobs from clients, dividing a job into tasks, and assigning these tasks to worker nodes to execute them. Each worker runs a Task-Tracker process that manages the execution of the tasks currently assigned to that worker node. Each Task-Tracker has a fixed number of slots for executing tasks (there are two map slots and two reduce slots by default).

A Hadoop job consists of two major phases, the map phase and the reduce phase. Each phase has key-value pairs as input and output and the types of key-value pairs may be chosen by the user. The user also specifies the map function and the reduce function that perform the data processing work defined by the user.

1) Map: Each map task (mapper) is assigned a portion of the input file called a split. By default, a split contains a single HDFS block (64 MB by default). A mapper will read the task's split from HDFS, parse it into records (key/value pairs), process the records by the user-defined map function, and then generate intermediate data as the input of the reduce tasks. After all of the input records have been processed by the user-defined map function, the mapper generates its final output. The mapper then registers the final output with the Task-Tracker. Finally, the TaskTracker informs the Job-Tracker that the map task has been finished.

2) Reduce: The execution of a reduce task (reducer) is divided into three phases:

I) The shuffle phase: In this phase, a reducer fetches its input data from the output of all mappers by issuing HTTP requests to all Task-Trackers. Each reducer is assigned a partition of the key range produced by the map step, so the reducer must fetch the content of this partition from every mapper's output. When all required data have been received, the reducer enters into the next phase.

II) The sort phase: In this phase, a reducer groups together the records from each mapper's output with the same key to form a list of values headed by the same key.

III) The reduce phase: In this phase, a reducer applies the user-defined reduce function to each key and its corresponding list of values. The output of the reduce function is written to a temporary location on HDFS. After the reduce function has been applied to each key in the reducer's partition, the reducer's HDFS output file is atomically renamed and moved from its temporary location to its final location.

B. Why Using Hadoop for CRCN

In a CRCN, the execution time of the SS algorithm on the cloud determines the delay of updates of the PPM database of PUs. To make the CR approach effective, the execution time of the SS algorithm must be as small as possible to reflect the activities of PUs in real time. Table I shows the execution time of the SS algorithm reported in [4]. The machine that they used was Windows Azure large instance (See the machine specification information in Table II). One can see that the execution time of the SS algorithm under higher measurement rates is still very large. The measurement rate is defined as the number of SUs divided by the region in units of km^2 . As the measurement rate grows, the correctness of the PPM database increases. However, one can see that the execution time grows very fast as the measurement rate grows due to the $O(n^3)$ complexity of the SS algorithm. These execution time results are only for a small 60 (Km) by 60 (Km) region. For a real-world large region, the execution time of the SS algorithm will grow up further and needs a parallel computing platform to reduce it. To achieve this goal, we decided to use Apache Hadoop to build our own cognitive radio cloud to reduce the execution time of the SS algorithm.

We chose Hadoop as the computing platform of CRCN for the following reasons:

1) Hadoop is a mature and reliable platform. It is widely used and supported. For example, Amazon EC2 cloud

TABLE I
EXECUTION TIME (SEC) OF THE SS ALGORITHM UNDER DIFFERENT MEASUREMENT RATES [4]

Measurement Rates	0.05	0.075	0.1	0.125	0.15
Execution time	6.5	8.9	12.5	14.6	24.5

TABLE II
THE COMPUTER INSTANCE SIZE OF WINDOWS AZURE

Computer instance size	CPU	RAM	Storage	IO
Extra small	1.0 GHz	768 MB	20 GB	LOW
Small	1.6 GHz	1.75 GB	225 GB	Moderate
Medium	2 x 1.6 GHz	3.5 GB	490 GB	High
Large	4 x 1.6 GHz	7 GB	1,000 GB	High
Extra Large	8 x 1.6 GHz	14 GB	2,040 GB	High

platform provides the Hadoop platform on which a user can write a Hadoop program to process a very large amount of data.

2) Hadoop is designed based on the MapReduce method, which is very suitable for executing the SS algorithm in parallel.

3) Hadoop is an open-source project. Thus, one can study its source code and change its internal design and implementation to meet one's special requirements.

With Hadoop MapReduce, one can easily use the region-division method to run the SS algorithm in parallel. However, we found that the fixed overhead of Hadoop is always greater than 20 seconds, which means that no matter how small the data set is, any program running on Hadoop always needs 20 seconds or more to finish. This is a very serious problem when one wants to use Hadoop to run a SS algorithm in real time for CRCN. To overcome this problem, we studied the Hadoop source code to realize how a job is processed over Hadoop and successfully found methods to reduce its fixed overhead. In the following, we describe our discoveries and solutions.

III. IMPROVEMENTS MADE TO HADOOP

A. Hadoop Job Execution Flow

To minimize the overhead of Hadoop, one should first realize the Hadoop Job execution flow. A job can be broken into four steps after the job client submits the job to the Job-Tracker.

1) Setup Step: After receiving a new job, the Job-Tracker will issue a setup-task request to a Task-Tracker that has a free slot for execution. A setup task will be created to initialize the environment for the job, which includes creating a temporary output directory for the job. Once the setup task is completed, the state of the job is switched to the RUNNING state.

2) Map and Reduce Step: After the setup task is finished, the Job-Tracker starts assigning tasks to a Task-Tracker. The Task-Tracker sends a heartbeat message periodically to the Job-Tracker informing the Job-Tracker that the Task-Tracker is still alive. A heartbeat message also contains the information that indicates whether the Task-Tracker is ready to run a new task or not. If it is ready, the Job-Tracker will use the heartbeat

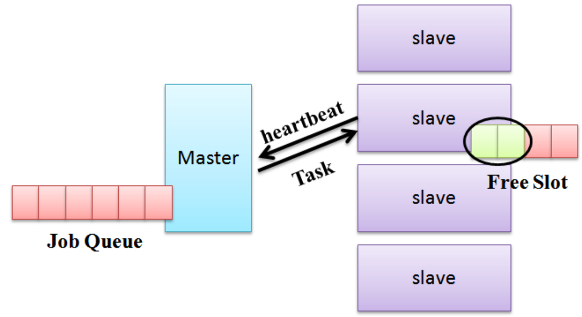


Fig. 1. The Heartbeat Design in Hadoop

return message to assign it a new task for execution. Fig. 1 shows the heartbeat design in Hadoop.

3) Cleanup Step: This step is used to clean up the job environment after a job has completed. For example, the temporary output directory created during the job execution should be removed after the job is completed. Job cleanup is done by a separate task at the end of the job. A job will be declared SUCCEEDED, FAILED, or KILLED after the cleanup task completes.

B. Main Sources of Hadoop Fixed Overhead

We found that the major sources of the Hadoop fixed overhead come from 1) Heartbeat interval, 2) Reduce sleep time, and 3) Commit sleep time. In the following, we explain these sources in details.

1) Heartbeat Interval: If we consider a small job that is processed by only 1 map task and 1 reduce task, the job execution flow in Hadoop is as follows:

1. The Job-Tracker receives a job submission and issues a setup-task request within a heartbeat return message to a Task-Tracker.
2. The Task-Tracker executes and completes a setup task and then reports to the Job-Tracker in its next heartbeat.
3. The Job-Tracker then asks the Task-Tracker to start a map task right after the completion of the setup task via a heartbeat return message.
4. The Task-Tracker completes the map task and reports to the Job-Tracker in its next heartbeat.
5. The Job-Tracker then asks the Task-Tracker to start a reduce task right after the completion of the map task via a heartbeat return message.
6. The Task-Tracker completes the reduce task and reports to the Job-Tracker in its next heartbeat.
7. The Job-Tracker then asks the the Task-Tracker to start a cleanup task right after the completion of the reduce task.
8. The Task-Tracker completes the cleanup task and reports to the Job-Tracker in its next heartbeat.
9. The Job-Tracker receives a completion report from the cleanup task, which indicates that the job is successfully done.

As one can see in the above execution flow, the Job-Tracker can only issues a task request after the Task-Tracker sends it a heartbeat to inform it that the Task-Tracker has a free slot for execution. Also, the Task-Tracker reports the completion of a task to the Job-Tracker only through the periodic heartbeats, which means that if a task's execution time (assuming it is 1 second) is smaller than the default heartbeat interval (which is 3 seconds), the Task-Tracker will sit idle in the remaining time of the current heartbeat period (i.e., 2 seconds) and the job procedure will be blocked until the heartbeat reports. Worse yet, there are four heartbeat messages in the job flow. Therefore, in this example case, one will waste 8 seconds (i.e., 2 seconds $*$ 4) doing nothing in the heartbeat periods. If one wants to run a real-time job such as the SS algorithm in CRCN, the 8-second latency is a very large fixed overhead.

To clearly see the effect of the heartbeat interval on the fixed overhead of Hadoop, we ran the PiEstimator [8] on the Hadoop platform. The PiEstimator (Pi) is a Hadoop built-in example. It uses the Quasi-Monte Carlo method to estimate the Pi value. A Pi job needs many map tasks to perform the Quasi-Monte Carlo method and a reduce task to calculate the estimated result. There are two parameters of a Pi job. The first one specifies the number of map tasks while the second one specifies how many sample points a map task should generate. Because the Pi job is a very small job needing very little computing time, we ran it to measure the fixed overhead of Hadoop. Table III and Fig. 2(a) show that, after we reduce the heartbeat interval from the default 3 seconds to 0.05 second, the execution time of a small job can be reduced by 11 seconds on average. The specification of the machine used for this Pi experiment is listed in Table VII under the "Our own machine" column.

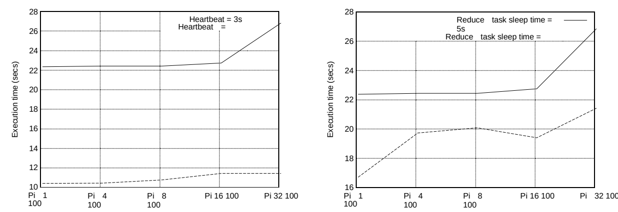
TABLE III
EXECUTION TIME OF PI UNDER DIFFERENT HEARTBEAT INTERVALS

	Default Interval = 3 sec	Modified Interval = 0.05 sec
Pi 1 100	22.362	10.365
Pi 4 100	22.430	10.393
Pi 8 100	22.433	10.721
Pi 16 100	22.738	11.396
Pi 32 100	26.841	11.398

2) The Sleep Time of A Reduce Task: In addition to the heartbeat interval, we also found that when a reduce task starts up, it polls the intermediate results generated by the map tasks that have completed. If a reduce task finds that there is no result to collect, it will sleep 5 seconds and then try the polling again. Using 5 seconds as the default sleep time is for saving the number of polling in a large job. This is because in such a job a map task may take tens of minutes or even hours to finish and it is reasonable that a reduce task uses a large sleep time between polling the output of map tasks. However, when a job can be effectively parallelized to make the computing time of a map task small, the default 5 seconds sleep time becomes a large fixed overhead for real-time applications. To see the effectiveness of the sleep time of a reduce task, we changed the default 5 seconds to 0.05 second. Table IV and Fig. 2(b) show that our modification of this parameter value can reduce the fixed overhead by almost five seconds.

TABLE IV
EXECUTION TIME OF PI UNDER DIFFERENT REDUCE TASK SLEEP TIME

	Default sleep time = 5 sec	Modified sleep time = 0.05 sec
Pi 1 100	22.362	16.716
Pi 4 100	22.430	19.731
Pi 8 100	22.433	20.084
Pi 16 100	22.738	19.409
Pi 32 100	26.841	21.428



(a) Effect of Heartbeat Interval (b) Effect of Reduce Task Sleep Time

Fig. 2. Effects of Heartbeat Interval and Reduce Task Sleep Time

3) The Sleep Time of Task Commit Function: We also discovered that when a job calls a done function, it will enter into a commit step to wait for the commitment from the Job-Tracker. In the done function, there is a 1-second sleep time between polling the arrival of the commitment. If we set it to a small value such as 0.05 second, the fixed overhead of Hadoop can be further reduced by about 1 second. Table V and Fig.3(a) show the execution time of the Pi job under different settings of this parameter.

TABLE V
EXECUTION TIME OF PI UNDER DIFFERENT COMMIT SLEEP TIME

	Default commit sleep time = 1 sec	Modified commit sleep time = 0.05 sec
Pi 1 100	10.365	9.355
Pi 4 100	10.393	9.362
Pi 8 100	10.721	9.412
Pi 16 100	11.396	10.395
Pi 32 100	11.398	10.401

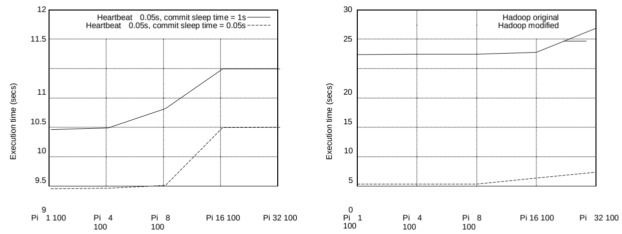
Table VI and FIg.3(b) show the execution time of Pi under the default parameter settings and under all of the three modified settings. One can see that the fixed overhead of Hadoop is reduced by about 18 seconds, which is important to help CRCN achieve high spectrum utilization.

TABLE VI
EXECUTION TIME OF PI UNDER THE DEFAULT AND ALL MODIFIED SETTINGS (SEC)

	Default settings	All modified settings
Pi 1 100	22.362	4.361
Pi 4 100	22.430	4.354
Pi 8 100	22.433	4.346
Pi 16 100	22.738	5.382
Pi 32 100	26.841	6.393

IV. EXPERIMENT CONFIGURATIONS

The Hadoop fixed overhead results presented in the previous section were measured when the simple Pi job was executed. To estimate the fixed overhead of the Hadoop platform when



(a) Effect of Commit Sleep Time (b) Effect of All Changes

Fig. 3. Effects of Commit Sleep Time and All Changes

the SS algorithm is executed to support CRCN, we used a small region with three PUs and some SUs. The size of the region is 60 (Km) by 60 (Km) with 3 PUs located at (15, 45), (45, 15) and (45, 45), respectively. We randomly selected some coordinate points and denoted them as the locations of SUs. The number of SUs used in an experiment is determined by the used measurement rate, which has been defined before in the paper.

We computed the RSSI (received signal strength indication) of each coordinate point where a SU resides and randomly added a Gaussian noise to it to represent the effect of signal noise in the real world. These sensing data represent the signal power sensed and reported by SUs in CRCN. The sensing data are saved in a file in the format of (x-position, y-position, RSSI) and the file is stored on the HDFS. We wrote a Hadoop MapReduce program to estimate the positions of PUs and their transmit powers using the SS algorithm. The map step of this SS algorithm first separated the SUs sensing data into 4 groups by their locations in order to run the SS algorithm in parallel. After the map step is finished, a total of 4 reduce tasks were then launched to process these data.

We ran the SS algorithm case under different measurement rates, which are 0.05, 0.075, 0.1, 0.125, and 0.15, respectively. As defined before in the paper, the measurement rate is defined as the number of SUs divided by the region in units of K m^2. Since in the experiment the region is 60*60 = 3,600 K m^2, the corresponding numbers of SUs in these experiments are 180, 270, 360, 450, and 540, respectively. Under a specific measurement rate, we ran the experiment three times and reported the average execution time of the three runs. The execution time of a job is determined by the job's start time and finish time logged in the Job Tracker's log file.

We used three different machine platforms to show the effectiveness of our modifications on the execution time of the SS algorithm. The first one is our own machine platform, composing of one i7 machine acting as the Hadoop master and two i7 machines acting as the Hadoop workers. The second one is composed of 3 Amazon EC2 [9] "large instances," with one playing the role as the Hadoop master while the others playing the role as the Hadoop workers. The third one is composed of 3 Amazon EC2 "extra large instances," with one being the Hadoop master while the others being the Hadoop workers. Table VII shows the detailed information about these three hardware platforms.

TABLE VII
HARDWARE PLATFORM INFORMATION

	Our Own Machine	EC2 Large Instance	EC2 Extra Large Instance
Worker number	2	2	2
CPU	I7 2600	4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each)	8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each)
Memory	16 GB	7.5 GB	15 GB
Disk Space	1 TB	850 GB	1,690 GB
Mapper Reducer max num.	8	8	8

V. EXPERIMENTAL RESULTS

Table VIII and Fig.4(a) show that our modifications to the original Hadoop platform can successfully reduce the execution time of the SS algorithm by 23 seconds on our own machine platform. This improvement is very important to a large-scale cognitive radio network as now the SS algorithm can be finished in only a few seconds, which makes the cognitive radio approach much more effective.

TABLE VIII
EXECUTION TIME (SEC) OF THE SS ALGORITHM ON OUR OWN MACHINE

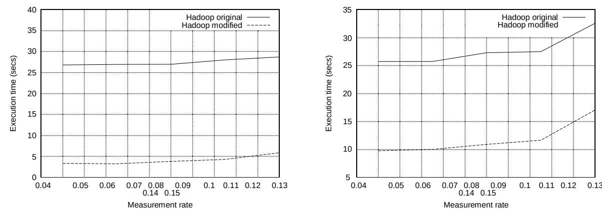
Measurement Rates	Hadoop Original	Hadoop Modified
0.05	26.777	3.35
0.075	26.918	3.25
0.1	26.924	3.82
0.125	27.986	4.29
0.15	28.619	5.89

Table IX and Fig.4(b) show the execution time of the SS algorithm on the EC2 public cloud using its large instances. The EC2 cloud already provides the original Hadoop platform for its users to run their Map/Reduce programs on it without any modification. To see the effectiveness of our modifications to the original Hadoop platform, we installed and used our modified Hadoop platform on the EC2 instances that we used for doing experiments. The results show that on average our modifications to the Hadoop platform can reduce the execution time of the SS algorithm by about 16 seconds over EC2 large instances. In contrast, Table X and Fig.5(a) show the execution time of the SS algorithm on the EC2 public cloud using its extra large instances. The results show that on average our modifications to the Hadoop platform can reduce the execution time of the SS algorithm by about 19 seconds over EC2 extra large instances.

Comparing Table IX and Table X with Table VIII, one can see that the execution time of the SS algorithm over the EC2 public cloud platform, whether its large or extra large instances are used, are still much larger than the execution of the SS algorithm over our own machine platform. These results may indicate that the instances (virtual machines) provided by the EC2 public cloud are equipped with slower CPUs than the PCs used in our own machine platform.

TABLE IX
EXECUTION TIME (SEC) OF THE SS ALGORITHM ON EC2 LARGE INSTANCES

Measurement Rates	Hadoop Original	Hadoop Modified
0.05	25.689	9.75
0.075	25.684	9.99
0.1	27.280	10.89
0.125	27.447	11.64
0.15	32.554	17.05

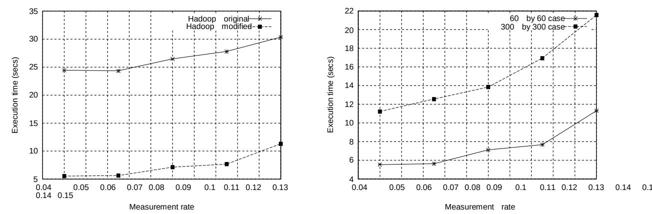


(a) Execution time of the SS algorithm on our own machine (b) Execution time of the SS algorithm on EC2 large instances

Fig. 4. Execution time of the SS algorithm on our own machine and on EC2 large instances

TABLE XI
EXECUTION TIME (SEC) OF THE SS ALGORITHM ON EC2 EXTRA LARGE INSTANCES USING OUR MODIFIED HADOOP FOR THE 60 X 60 AND 300 X 300 REGION CASES

Measurement Rates	60 x 60 region	300 x 300 region
0.05	5.52	11.21
0.075	5.62	12.54
0.1	7.11	13.84
0.125	7.66	16.93
0.15	11.30	21.55



(a) Execution time of the SS algorithm on EC2 extra large instances (b) Execution time of the 60 x 60 and 300 x 300 region cases

Fig. 5. Execution time of the SS algorithm on EC2 extra large instances under the 60 x 60 and 300 x 300 region settings

To test the scalability of our design and implementation, we built and ran a larger test case with a 300 (Km) by 300 (Km) region and 80 PUs. The map function first separated the SUs sensing data into 100 30 (Km) by 30 (Km) regions. The data associated with a region are assigned to a reduce task to calculate the SS result in that region. The machine platform that we used for running this case is composed of 14 EC2 extra large instances, among which one instance acts as the master and the other instances act as the 13 workers. The machine information and configuration is the same as those listed in Table VII. Table XI and Fig.5(b) show that for a specific measurement rate, the execution time of the 300 x 300 region case is about 6 seconds to 10 seconds larger than that of the 60 x 60 region case, even though in both cases an instance is responsible for the same 30 x 30 region. Our preliminary study showed that this execution time increase is caused by the bottleneck in the reduce shuffle phase of Hadoop and we will explore this issue further in our future work.

VI. CONCLUSION

In this paper, we optimize the cloud platform performance for supporting large-scale cognitive radio networks. In such a network, a cloud platform is used as the computing platform to run the SS algorithm in real time. The goal is to make the PU database as accurate as possible at any given time.

TABLE X
EXECUTION TIME (SEC) OF THE SS ALGORITHM ON EC2 EXTRA LARGE INSTANCES

Measurement Rates	Hadoop Original	Hadoop Modified
0.05	24.436	5.52
0.075	24.327	5.56
0.1	26.463	7.11
0.125	27.780	7.66
0.15	30.342	11.30

Due to its maturity and popular supports over public clouds such as the Amazon EC2 public cloud, the Hadoop platform is very suitable for running the SS algorithm in parallel on the cloud. However, we found that the original design and implementation of the Hadoop platform cause a significant fixed overhead for any job running on it, including the SS algorithm. To overcome this problem, we studied the source code of the Hadoop platform to understand how it processes a job in a distributed manner.

Our detailed study identified three main sources of the fixed overhead in the original Hadoop platform. Our modifications to the Hadoop platform can successfully reduce the fixed overhead by about 20 seconds on our own machine platform and make the resulting execution time less than only a few seconds. In summary, the improvements that we made to the original Hadoop platform make the PU database more accurate at any given time, which in turn makes the cognitive radio approach much more effective.

REFERENCES

[1] J. Mitola and G. Q. Maguire, Cognitive radio: Making software radios-more personal, IEEE Personal Communications, vol. 6, no. 4, pp. 1318, Aug. 1999.
[2] M. A. McHenry, NSF spectrum occupancy measurements project sum- mary, shared spectrum co. report, Aug. 2005.
[3] FCC, Second report and order and memorandum opinion and order, FCC 08-260, Nov. 2008.
[4] Chun-Hsien Ko, Din Hwa Huang and Sau-Hsuan Wu, Cooperative Spectrum Sensing in TV White Spaces: When Cognitive Radio Meets Cloud, IEEE INFOCOM 2011 Workshop on Cloud Computing
[5] D.-H. Tina Huang, S.-H. Wu, and P.-H. Wang, Cooperative Spectrum Sensing and Locationing: A Sparse Bayesian Learning Approach, in Proc. IEEE GLOBECOM. Miami, USA, Dec. 2010.
[6] Apache Hadoop, http://hadoop.apache.org/
[7] Apache Hbase, http://hbase.apache.org/
[8] http://hadoop.apache.org/common/docs/current/api/org/apache/hadoop/examples/PiEstimator.html
[9] http://aws.amazon.com/ec2/instance-types/

A Conceptual Model and Prototype of Cognitive Radio Cloud Networks in TV White Spaces

Sau-Hsuan Wu†, Hsi-Lu Chao‡, Chung-Ting Jiang†, Shang-Ru Mo‡, Chun-Hsien Ko†,
Tzung-Lin Li‡, Chiau-Feng Liang‡, and Chung-Chieh Cheng†
†Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan 300
E-mail:{sauhsuan@cm.nctu.edu.tw, {tim750811.cm98g, dke.cm99g, james770329.cm99g}@nctu.edu.tw}
‡Institute of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 300
E-mail:{hlchao@cs.nctu.edu.tw, {maxmo77715.cs99g, ivljyf.cs99g, streamaway.cs99g}@nctu.edu.tw

***Abstract*—A Cognitive Radio Cloud Network (CRCN) model is proposed for wireless communications in TV White Spaces (TVWS). Making use of the flexible and vast computing capacity of the Cloud, a database and a sparse Bayesian learning (SBL) algorithm are developed for cooperative spectrum sensing (CSS) and implemented on Microsoft's Windows Azure Cloud platform. A medium access control (MAC) scheme is also prototyped for this CRCN model to collect sensing reports and access channels with Rice University's wireless access research platform (WARP). Through this CRCN prototype, important network parameters such as the mean squared errors in CSS, the time to detect the presence and/or the absence of primary users, and the channel vacating delay are measured and analyzed for the design and deployment of the future CRCN.**

***Index Terms*—Cognitive Radio, Cloud Computing, CR-MAC, Cooperative Spectrum Sensing and Sparse Bayesian Learning.**

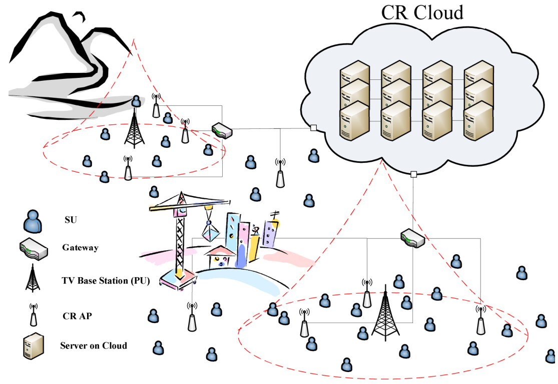


Fig. 1. A conceptual *Cognitive Radio Cloud* network (CRCN) model in TV White Space. It illustrates the infrastructure of CRCN and the power coverage ranges of the TV BS (PU BS), CR SUs and CR APs inside the CRCN.

I. INTRODUCTION

The concept of cognitive radio (CR) is first introduced by Joseph Mitola III in [1]. Under the framework of CR, unlicensed Secondary Users (SUs) can access the spectrum for licensed Primary Users (PUs) under the condition that the service quality of PUs can be effectively preserved or guaranteed. In view of the inefficient usages of some legacy spectrum holders (less than 25% overall according to [2]), CR is expected to resolve part of the spectrum shortage issues in wireless metropolitan and local area networks (MAN/LAN).

The termination of analog TV broadcasting allows the Federal Communications Commission (FCC) of the U.S to take an initiative to test the CR concept [3]. CR networks are later granted for field trials in part of it's TV White Spaces (TVWS). Encouraged by FCC's policy on CR, international organizations have also started to define CR standards in TVWS, e.g. IEEE 802.22, 802.11af and ECMA 392, etc.

To ensure the received signal quality of TV sets, the FCC requires CR operators able to access databases that can provide the power propagation map (PPM) of TV signals with a sensitivity of 0.8dBm below the noise level (-106.2dBm). In addition, a database also needs to provide the geographical locations of TV base stations (BS) and their radiation powers, antenna heights and channel numbers, etc. SUs use the databases to determine their access rights in TVWS. To achieve these multiple purposes of CR databases in TVWS, some research units have started to develop databases that make use of location-based cooperative spectrum sensing (CSS).

Compared to the BSs of regular cellular networks, the radiation power of a TV BS typically covers a much larger area which is beyond the reach of the transmit power of a mobile device. This makes the operation of a CR network in TVWS different from that of a LAN or a MAN. The reasons are at least two-fold: First, to reconstruct the PPM of even a TV BS, it requires sensing reports from devices located in different positions inside the TV signal's coverage range. Considering the much weaker transmit power of a sensing device (SD), it requires many CR access points (APs) in the same area to collect the sensing reports from distributed SDs. The CR APs further send the sensing reports through backhaul to a fusion center to reconstruct the PPM of the TV signal. Second, even if the channel accesses of SUs are established through their associated CR APs, the wireless links' activities and power strengthes should still be coordinated by a center in order to facilitate resource sharing among the multiple CR APs or operators in the huge spectrum of TVWS. These two important features of CR in TVWS have attracted significant research attentions either in CSS or spectrum resource sharing (SRS). To exploit the vast spectrum resources in TVWS, we propose a CR network model powered by Cloud (CRC) computing. The CRC not only can harmonize the functions of CSS, channel access (CA) and SRS in TVWS, but also facilitates and materializes the establishment of a CRC network (CRCN) whose operation can be more delicate and complex than a LAN, while its infrastructure is much simpler than a MAN. A conceptual operation model of CRCN is illustrated in Fig. 1.

A similar concept of cognitive wireless Cloud (CWC) has been introduced by H. Harada *et al* in [4] where a Cloud-based algorithm is proposed to coordinate SRS among heterogeneous networks. In contrast to their ideas in heterogeneous networks, th CRCN model proposed herein integrates the functions of CR spectrum sensing (SS), management, CA and SRS under a uniform Cloud framework. Making use of the readings of positions and radio signal strength indicators (RSSI) collected from SDs or SUs, a CSS algorithm is proposed to estimate the PPM, positions and the number of PUs based on the concept of sparse Bayesian learning (SBL) [5]. Considering that the positions and the population densities of CR SUs are supposed to be random and may vary in different times and areas, the SBL-based CSS algorithm is developed on Microsoft's Windows Azure Cloud platform. Exploiting the scalability of Cloud computing, the proposed CSS architecture is in principle able to do SS over a huge area with sparse sensing reports.

In addition to CSS, we also design a multiple access control (MAC) protocol for cognitive channel accesses in TVWS. Based on Rice University's wireless access research platform (WARP) and Texas Instruments (TI) radio transceivers, the CR-MAC protocol is implemented for SS and frequency-division and time-division multiple accesses (FD-TDMA) among SDs and SUs. Together with the spectrum database of PUs built with the SQL Azure of Microsoft's Cloud platform, the proposed CRCN model is prototyped in an architecture illustrated in Fig. 2. Through this prototype, important CRCN parameters such as the time for and the mean squared error (MSE) of PPM reconstructions, the channel vacating delay of SUs in the presence of PUs can be assessed for future designs of CRCN in TVWS.

The paper is organized as follows. Section II introduces the infrastructure of CRCN. The CSS algorithm is specified in Section III followed by the CR-MAC protocol in Section IV. The experimental results are presented and discussed in Section V. Concluding remarks are provided in Section VI.

II. The Infrastructure of the CRCN

We first introduce Microsoft's Windows Azure Cloud platform on which the CRCN is prototyped. The Cloud platform supports program developments in JAVA or in C$^\sharp$ and Visual Basic. The programming models in Windows Azure include:

- Web Role: to communicate between users and background processes. It can be implemented by dynamic web languages, for example, ASP.NET and PHP, etc.
- Worker Role: to execute tasks with background processes in Windows Azure and export the results periodically.

The operating system (OS) for Windows Azure is Windows Azure Guest OS 1.9 which is a virtual machine (VM) version of Windows Server 2008 SP2. Windows Azure also supports three types of data storages which are BLOB for general binary

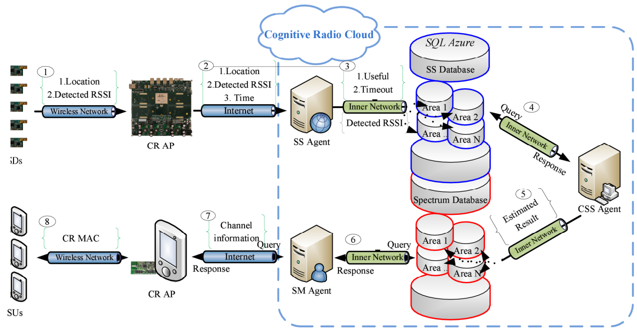


Fig. 2. The block diagram of the proposed CRCN model. The CRC consists of SS agents and SM agents through which SS data are stored and retrieved, respectively, from the databases of SQL Azure. Each agent is implemented on a VM in the Worker Role of Windows Azure. The SBL-based CSS agent partitions the CSS task into parallel procedures running on different VMs.

data, Table for systematic data and Queue for data passing between webs and programs. Moreover, built on Windows Azure, SQL Azure is the Cloud version of the SQL server, though, it only supports part of the functions of the SQL server.

To reconstruct the PPM on Windows Azure, CR APs, SDs or SUs inside the CRCN are assumed equipped with the global positioning system (GPS) devices. The CR APs collect and report the locations, times and RSSI readings of their associated SDs or SUs to an SS agent (VM) in the CRC of Fig. 2, through which the data are stored in an SS database built with SQL Azure. The database are processed by the CSS engines to do the SBL-based CSS whose details will be introduced in Section III. Because the CSS algorithm complexity grows in the third order of the number of sensing measurements, the sensing data are partitioned by a CSS agent into blocks according to their geographical locations and population densities. Data of each block are processed independently by a CSS engine to parallelize the CSS task. The reconstructed PPM of the PUs of each time slot are stored into a spectrum database of SQL Azure, which contains the estimated number and positions of the PUs as well.

When an SU wants to access PUs' licensed channels, it first sends a CA request to its associated CR AP through which the aggregated spectrum requests are issued to an spectrum management (SM) agent (VM) in the CRC. The agent checks the spectrum database and informs the CR-AP of its granted wireless channels and corresponding transmit power. The SUs then access the channel according to the CR-MAC which is going to be introduced in Section III. An illustration that depicts the infrastructure of CRCN is provided in Fig. 2.

III. The SBL-Based Cooperative SS Algorithm

Assume that there are N SDs or SUs in an area of $N_p \times N_p$ in which exist M_p active PUs. The RSSI readings of $\boldsymbol{t} = [t_1, t_2, \cdots, t_N]^T$ collected from SDs and/or SUs at locations of $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N]$ in the area, with $\boldsymbol{x}_i \triangleq [x_i, y_i]^T$, are fed back by their associated CR APs to the CRC. The RSSI $\boldsymbol{t}$ in decibel (dB) are modeled in a linear regression form of [6]

$$\boldsymbol{t} = \boldsymbol{\Phi} \boldsymbol{w} + \boldsymbol{n} \tag{1}$$

where $\mathbf{\Phi}_{N\times M} \triangleq [\boldsymbol{\psi}_1(\boldsymbol{X}), \boldsymbol{\psi}_2(\boldsymbol{X}), \cdots, \boldsymbol{\psi}_M(\boldsymbol{X})]$, with $\boldsymbol{\psi}_j(\boldsymbol{X}) \triangleq [\phi_j(\boldsymbol{x}_1), \phi_j(\boldsymbol{x}_2), \cdots, \phi_j(\boldsymbol{x}_N)]^T$ further defined by $\phi_j(\boldsymbol{x}_i) \triangleq \frac{1}{2s_j}\exp\left\{-\sqrt{(x_{i,x}-\mu_{j,x})^2+(x_{i,y}-\mu_{j,y})^2}/s_j\right\}$ for $j = 1 \dots M$. Supposed there are M basis functions $\phi_j(\boldsymbol{x}_i)$ that model the transmit power of PUs located at $\boldsymbol{\mu} \triangleq [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_M]^T$, with $\boldsymbol{\mu}_j \triangleq [\mu_{j,x}, \mu_{j,y}]$. The power decaying rates of $\phi_j(\boldsymbol{x}_i)$ are $\boldsymbol{s} \triangleq [s_1, s_2, \cdots, s_M]^T$. The RSSI vector $\boldsymbol{t}$ is a linear combination of $\phi_j(\boldsymbol{x}_i)$ whose weighting coefficients are $\boldsymbol{w} \triangleq [w_1, w_2, \cdots, w_M]^T$. Each entry w_j is endowed a *prior* probability $\mathcal{N}(0, \alpha_j^{-1})$. The logarithms of the shadowing effects, $\boldsymbol{n}$, in $\boldsymbol{t}$ are modeled as zero-mean Gaussian random variables whose variances are β^{-1}. The SBL-based CSS algorithm iteratively estimates the parameters $\boldsymbol{\alpha} \triangleq [\alpha_1, \dots, \alpha_M]^T$, β, $\boldsymbol{\mu}$, $\boldsymbol{s}$ and M to maximize $p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{\alpha}, \beta, \boldsymbol{\mu}, \boldsymbol{s}, M)$ from sparse measurements of $\boldsymbol{t}$ and $\mathbf{X}$.

The CSS algorithm can be viewed as an alternative EM algorithm. In the E-step, the covariance $\mathbf{\Sigma}$ and the mean $\boldsymbol{m}$ of the posterior distribution of $\boldsymbol{w}$ are evaluated to be

$$\mathbf{\Sigma} = (\beta\mathbf{\Phi}^T\mathbf{\Phi} + \boldsymbol{A})^{-1}, \text{ and } \boldsymbol{m} = \beta\mathbf{\Sigma}\mathbf{\Phi}^T\boldsymbol{t} \tag{2}$$

where $\boldsymbol{A} \triangleq \mathrm{diag}\{\alpha_j\}$ is an $M \times M$ diagonal matrix. We can thus assign the estimated weighting coefficients to be $\widetilde{\boldsymbol{w}} = \boldsymbol{m}$ and delete those $\phi_j(\boldsymbol{x}_i)$ whose weighting $\widetilde{\boldsymbol{w}}$ are less than a threshold η. The number of survival $\phi_j(\boldsymbol{x}_i)$ then becomes

$$M_{(k)} = \sum_{j=1}^{M_{(k-1)}} \mathbf{I}(\widetilde{w}_j \geq \eta) \tag{3}$$

where $\mathbf{I}$ is an indicating function. Becasue there are two M-steps in the alternative EM algorithm, and $\boldsymbol{m}$ is mainly adjusted in the first M-step, the bases deleting criterion is only applied in the first EM in which $\boldsymbol{\mu}$, $\boldsymbol{s}$ and M are given. The remaining two parameters α_j^{-1} and β^{-1} are estimated to be

$$\alpha_j^{-1} = \frac{m_j^2}{\gamma_j}, \text{ and } \beta^{-1} = \frac{||\boldsymbol{t} - \mathbf{\Phi}\boldsymbol{m}||^2}{N - \sum_{j=1}^{M}\gamma_j} \tag{4}$$

where $\gamma_j \equiv 1 - \alpha_j\Sigma_{jj}$, and Σ_{jj} are the diagonal terms of $\mathbf{\Sigma}$. The smaller is the α_j^{-1}, the more likely is ϕ_j a redundant basis. Besides, the deleting threshold η in (3) should be set according to the noise variance β^{-1} of $\boldsymbol{n}$.

In the second M-step, α_j^{-1}, β^{-1}, and M are given. We thus infer the parameters $\boldsymbol{\mu}$ and $\boldsymbol{s}$ that maximize the likelihood function $p(\boldsymbol{t}|\boldsymbol{\mu}, \boldsymbol{s}; \boldsymbol{X}, M, \beta, \boldsymbol{\alpha})$ by the *gradient descent method*

$$\begin{bmatrix} \mu_{j,x}(k) \\ \mu_{j,y}(k) \\ s_j(k) \end{bmatrix} = \begin{bmatrix} \mu_{j,x}(k-1) \\ \mu_{j,y}(k-1) \\ s_j(k-1) \end{bmatrix} - \delta \begin{bmatrix} \left.\frac{\partial Q}{\partial \mu_{j,x}}\right|_{\mu_{j,x}(k-1)} \\ \left.\frac{\partial Q}{\partial \mu_{j,y}}\right|_{\mu_{j,y}(k-1)} \\ \left.\frac{\partial Q}{\partial s_j}\right|_{s_j(k-1)} \end{bmatrix} \tag{5}$$

where k is the iteration index, $Q \triangleq -\ln p(\boldsymbol{t}|\boldsymbol{\mu}, \boldsymbol{s}; \boldsymbol{X}, M, \beta, \boldsymbol{\alpha})$ and $\delta > 0$ is the step size or referred to as the *learning rate*. The details of the iteration process is shown in Table I.

Under the infrastructure of CRCN in Fig. 2, the SBL-based CSS algorithm is implemented on Windows Azure following a MapReduce-like programming model [7]. The input SS data from CR APs are first processed by SS agents and stored in different tables of the SS database according to their locations. The CSS engines are managed by a CSS agent and are activated periodically to estimate the PPM, namely the parameters $\boldsymbol{\alpha}$, β, $\boldsymbol{\mu}$, $\boldsymbol{s}$ and M from the sparse measurements $\boldsymbol{t}$ and $\mathbf{X}$ in the tables. The reconstructed PPM, $\mathbf{\Phi}\tilde{\boldsymbol{w}} = \mathbf{\Phi}\boldsymbol{m}$, and the estimated number $\tilde{M}_p$ and locations $\tilde{\boldsymbol{\mu}}$ of the PUs are stored in the spectrum database to be used by the SM agents. A block diagram that depicts this processing flow is in Fig. 3.

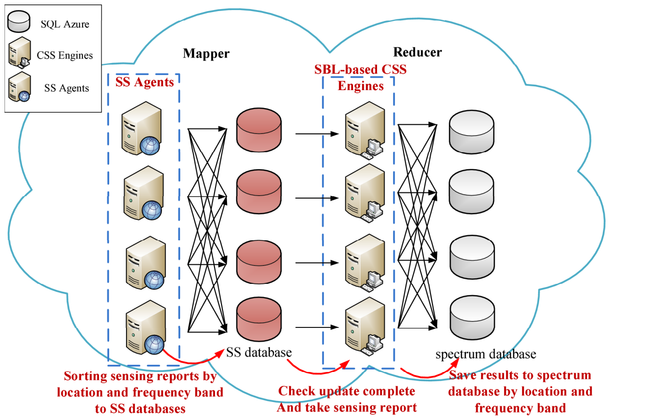


Fig. 3. The block diagram of the SBL-based CSS algorithm implemented on Microsoft's Windows Azure. In this example, there are 4 SS agents to process the input SS data. Data in each table of the SS database is processed by a CSS engine. The results are output to the spectrum database.

TABLE I

The Sparse Bayesian Learning Algrithm
1) Uniformly spread M bases ϕ_j in the area of interest.
2) Initiate the iterations with $\alpha_j = 1$, $\beta = 1$ and $k = 0$, and evaluate the corresponding mean $\boldsymbol{m}$ and covariance $\mathbf{\Sigma}$.
3) Let $k = k + 1$. Update α^{-1} and β^{-1} and then evaluate $\boldsymbol{m}$, $\mathbf{\Sigma}$ and $Q(k)$.
4) Delete the bases whose corresponding weighting $\widetilde{w}_j < \eta$ and then renew the M equal to the number of the survival bases. Renew the matrix $\mathbf{\Phi}$ and $\boldsymbol{A}$.
5) Let $k = k + 1$. Update $\boldsymbol{\mu}_j = [\mu_{j,x}, \mu_{j,y}]$ and s_j and and then evaluate $\boldsymbol{m}$, $\mathbf{\Sigma}$ and $Q(k)$. Go to step 6) if $(Q(k) - Q(k-1))/Q(k-1) < 0.0001$. Otherwise, repeat this step for L times, then go back to step 3).
6) Output the $\boldsymbol{\mu}_j = [\mu_{j,x}, \mu_{j,y}]$ and s_j. Let $\widetilde{M}_p = M$ and $\widetilde{\boldsymbol{w}} = \boldsymbol{m}$.

IV. CR MAC Protocol for Channel Access in TVWS

We design and implement a FD-TDMA based CR MAC protocol in this CRCN prototype. This CR MAC protocol is specifically designed for an infrastructure-based CR network, which consists of CR APs, SDs, and SUs. Channel sensing and reporting are mandatory to SDs and APs, while they are optional to SUs. Besides, a SU must be associated with a CR AP for data channel accesses. TVWS spectrums are channelized and partitioned into control channels and data channels. A CR AP uses one control channel and manages a number of data channels. The control channels are for CR

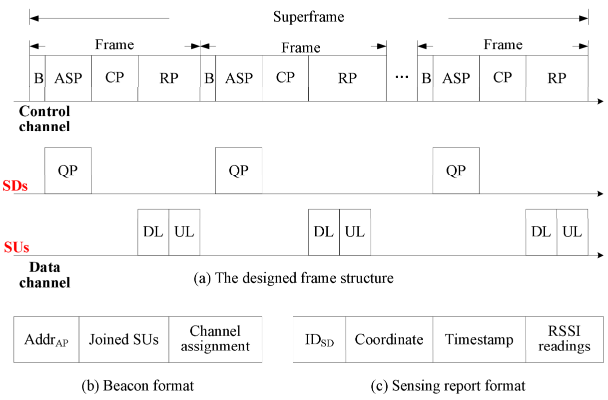


Fig. 4. The frame format of the CR-MAC for CRCN.

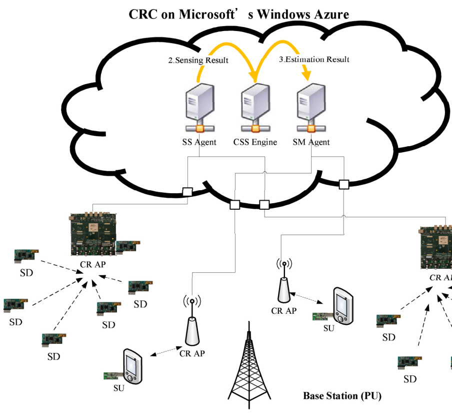


Fig. 5. The experimental setting for a CRCN prototype.

APs to coordinate data channel accesses and collect sensing reports; data channels are assigned to SUs for data deliveries. Multiple SUs can be assigned to different data channels or share the same channel in a TDMA manner. SDs and SUs are synchronized with their CR APs, and always listen to the control channel unless they are requested to perform channel sensing or permitted to deliver data on data channels.

In the CR-MAC, time is framed, and several frames are grouped into a superframe, as shown in Fig. 4 (a). For simplicity, we only show one data channel. Each frame on control channels starts with a beacon, followed by an association period (ASP), a collection period (CP), and a reservation period (RP). ASP and RP are for SUs to send joining/leaving messages and channel requests to their CR APs, respectively, while CP is for CR APs to collect channel sensing reports. Both ASP and RP are divided into time slots, and adopt a contention-based multiple access scheme. CR APs periodically broadcast beacon messages on the control channels to announce their existence, and management channel activities. To achieve this, a beacon consists of three fields: CR AP addresses ($Addr_{AP}$), associated SUs' IDs, and data channel assignments, as shown in Fig. 4(b). Data transmissions are classified into downlink (DL) and uplink (UL) deliveries. We elaborate the operations of channel access and report collection in details in the following.

A. Channel Access

When receiving a beacon, a new joining SU sends a joining request to the CR AP in a randomly selected time slot of the ASP. The association process is successful if the SU's ID is listed in the associated SUs' IDs of the next frame. Otherwise, the SU keeps sending joining request in the following frames. Similar operations are performed for SUs' leaving.

When an associated SU wants to initiate a data delivery, it first sends a channel request in a randomly selected time slot of the RP to the CR AP. Each channel request is with a timer. The CR AP further forwards the received requests in a specific frame to the SM agent through the backhaul. Based on these requests, SUs' locations, and PUs' information, the SM agent performs channel assignment and informs the CR AP of the results. The CR AP then broadcasts the channel assignments to managed SUs in the following beacon. If a SU does not get channel grant after timeout, it will re-issue a channel request. Upon getting a channel access grant, the SU hops to the designated data channel to do data deliveries. It vacates the channel when either finishing data transmissions or PUs are detected. For the latter, the CR AP will assign another available data channel to the SU for the remaining data transmissions.

B. Sensing Report Collection

To guarantee the sensing quality and increase the precision of PPM estimation, CR APs and SDs synchronously sense the data channels for a time duration, which is named Quiet Period (QP). In this QP, associated SUs are forbidden to transmit data on data channels, while new SUs are allowed to send joining requests on the control channel. Therefore, in our design, the time duration of QP equals to that of the ASP.

Following the QP, the SDs hop back to the control channel to report their sensing results in the CP. Instead of using contention-based channel accesses, the CR AP actively polls the managed SDs for sensing reports. Since SDs are designated for SS and reporting only, a polling mechanism is more efficient for SDs to avoid collisions in transmissions.

The format of a RSSI sensing report is shown in Fig. 4(c). The CR AP and SDs report not only the measured RSSI readings of data channels, but also their IDs, coordinates, and the timestamps.

V. Experimental Results

We present some experimental results on a CRCN prototype developed in this project. In the CRCN, the SUs use TI's transceiver modules numbered CC1111, and the SDs use the HM-TR transceiver modules of HOPERF [8]. There are 10 SDs, 2 SUs, 4 CR-APs, 2 control channels and 2 data channels used in the CRCN. The CR APs that collect sensing reports from SDs are implemented with WARPs [9], while the SUs and the CR APs that manage cognitive channel accesses are implemented with laptop computers and CC1111s. The experimental setting is illustrated in Fig. 5. A CR-AP for SS

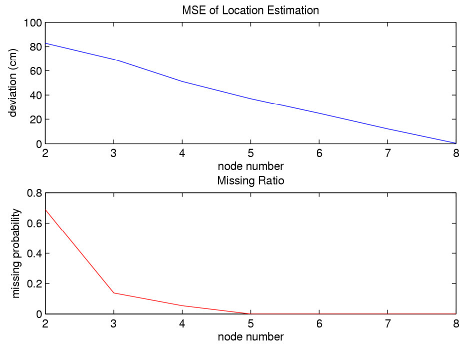


Fig. 6. The MSE of the PU position estimation and the missing ratio in the PU detection versus the number of SDs in the CRCN prototype.

is associated with 5 SDs which share the same control channel. To avoid interfering with TV signals, the frequency channels used in experiments are set in the ISM bands right above the UHF TV band. The 2 control channels are set at the 866 and 868 MHz bands, and the data channels are set at the 870 and 872 MHz bands whose bandwidth is 128kHz for each channel.

TABLE II
THE RSSI LEVELS OF THE HM-TR TRANSCEIVER MODULES

RSSI Readings	1	2	3	4	5
dBm Values	−96.2	−87.1	−78.6	-71.2	-61

A. SBL-Based Cooperative Spectrum Sensing in CRCN

In the experiments, the subnetworks formed with the four CR APs overlap with each other in an area that exists one PU BS only. The PU's frequency is set at 870 MHz, and SS are done by the SDs only, whose RSSI readings are sent back to the CRC through their associated CR APs. The RSSI readings of the HM-TRs consist of 5 strength levels whose values are shown in Table II. To conduct experiments in a constrained lab space, the transmission power of the PU BS is set to the same order of the CR AP such that the PPM of a PU BS can be reconstructed with the sensing reports of the 10 SDs. This setting not only simplifies the complexity of experiments, but also helps assess the robustness of the proposed SBL-based CSS algorithm as there are now only 10 SDs whose RSSI readings are partitioned into only 5 levels in a wide range of signal strengthes from -96 to -61 dBm. The experiments thus help examine the effects of RSSI's precision on the quality of the reconstructed PPM.

The MSE of the estimated PU location versus the number of SDs in PPM reconstruction is shown in the upper subplot of Fig. 6. The missing ratio in PU detection is shown in the lower subplot. The missing ratio is less than 0.1 when the number of SDs is greater than 4. Both results show that the proposed SBL-based CSS algorithm is quite robust even with limited number of imprecise sensing measurements.

B. Response Time of Channel Vacating in CRCN

In this experiment, a superframe is set consisting of 5 frames according to the CR-MAC format in Fig. 4. Each frame duration is set to 400 ms. SDs perform channel sensing once per superframe and only in the QP of the first frame. A QP (ASP) lasts for 50 ms, and is preceded by a BP of 10ms and followed by a CP of 160ms for sensing reports. To compensate the synchronization errors, fields in and between frames are separated by a guard interval of 10ms as well. Further, each SD takes 30 ms to complete its reporting. Therefore, a CR AP can manage reports of 25 SDs at most. In the experiments, the SDs associated with the two different APs start reporting at different frames in a supperframe, one group at the first CP (90ms), the other at the second CP (490ms) of a superframe.

On the other hand, to meet the FCC requirement, upon maintaining a database of PU information, of a maximum query interval of 60 sec, CR APs query the SM agent about channel availability once per superframe (i.e., 2 sec query interval). Specifically, the query time is at the beginning of the last frame in each superframe, i.e., at time 1.6 second, 3.6 second, etc. The performance metric we investigate here is channel vacating time, which is the time from the presence of a PU to the time the PU channel is released. The testing results of 360 runs are presented in Fig. 7.

We observed that the minimum, average, and maximum channel vacating times are 1.723 sec, 3.7757 sec, and 5.985 sec, accordingly. Several factors affect the measured channel vacating time, such as the time to read/write databases, the execution time of the PPM estimation algorithm, and network communication time. Based on the implemented FD/TDMA-based MAC protocol, we then analyze the reasonableness of those testing results. First, the best-case performance would be the situation that the PU appears before SDs start their sensing, processing is finished and PU information is available before the CR AP querys the SM agent in the same frame. In such a case, the estimated channel vacating time is the timestamp of CR AP issuing a query plus communication time from the SM agent to the CR AP and from the CR AP to the SU. Through observing those testing results, the communication time is roughly 0.2 sec, and thus the minimum channel vacating time is approximately 1.6 sec + 0.2 sec = 1.8 sec.

On the other hand, when the PU appears after the first QP of a superframe, it will be detected in the next superframe. As a result, the 10 SDs will finish reporting at the end of the second CP, which is 490ms + 160ms = 650ms after the start of the second superframe. Plus the typical 1.36 sec for PPM estimation in the CSS engine, the PU's presence cannot be made available when CR APs query the SM agent at 1.6 sec in the same superframe. Therefore, CR APs can only know the PU's presence in the third superframe, which ends up with the worst-case response time of approximately 4 + 1.6 + 0.2 = 5.8 sec. The testing results match the estimated values.

VI. CONCLUSIONS AND DISCUSSIONS

In this paper, we introduced the concept of a CRCN, and pointed out the importance and the necessity of CSS in a

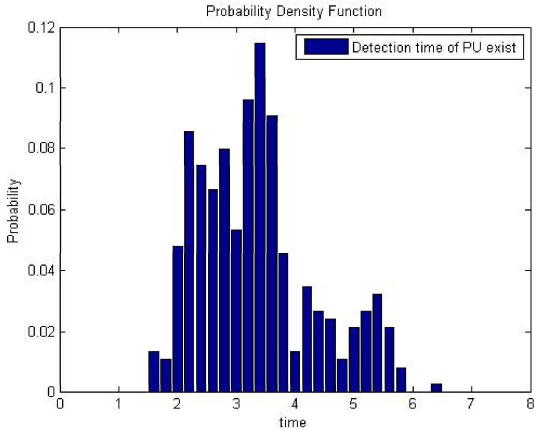


Fig. 7. The probability density function versus the channel vacating time of SUs in the CRCN prototype.

CRCN. In addition, we prototyped a CRCN with Microsofts Windows Azure Cloud platform. Through implementing the SBL-based CSS algorithm and the FD/TDMA-based MAC protocol on the CRCN platform, we evaluated the MSE and the channel vacating time in the presence of PUs. The results showed that the proposed SBL-based CSS algorithm performs well even with limited number of imprecise sensing measurements. Moreover, we analyzed and discussed the reasonableness of the CRCN response time in PU detection. More advanced design issues such as asynchronous CSS, VM scaling up and out, and real-time MAC programming will be examined and verified in our future work.

VII. Acknowledgement

This research has been funded by the National Science Council, Taiwan, under Grant NSC 100-2219-E-009-011.

References

[1] J. Mitola and G. Q. Maguire, “Cognitive radio: Making software radios more personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
[2] M. A. McHenry, *NSF spectrum occupancy measurements project summary*, shared spectrum co. report, Aug. 2005.
[3] FCC, *Second report and order and memorandum opinion and order*, FCC 08-260, Nov. 2008.
[4] H. Harada, H. Murakami, K. Ishizu, S. Filin, Y. Saito, H. N. Tran, G. Miyamoto, M. Hasegawa, Y. Murata, and S. Kato, “A Software Defined Cognitive Radio System: Cognitive Wireless Clouds,” in *Proc. IEEE Global Communications Conference (GLOBECOM)*. Washington DC, USA, Nov. 2007.
[5] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *Journal of Machine Learning Research*, vol. 1.
[6] D.-H. Tina Huang, S.-H. Wu, and P.-H. Wang, “Cooperative Spectrum Sensing and Locationing: A Sparse Bayesian Learning Approach,” in *Proc. IEEE GLOBECOM*. Miami, USA, Dec. 2010.
[7] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
[8] HOPERF, *HM-TR Series UHF Wireless Transparent Data Transceiver*, HOPERF co., Jun. 2008.
[9] *WARP Repository*, available at: http://warp.rice.edu/trac.

國立交通大學出國報告書

100 年 4 月 10 日

<table>
<tr><td rowspan="2">報告人姓名</td><td rowspan="2">王協源</td><td rowspan="2">申請單位
（學生請加註系級）</td><td rowspan="2">資訊工程系</td><td>職稱</td><td>教授</td></tr>
<tr><td>電話</td><td>5131550</td></tr>
<tr><td>出國類別</td><td colspan="5">□考察 □訪問 □進修 □研究 ☑國際會議 □其他：________________</td></tr>
<tr><td>會議/出國計畫名稱</td><td colspan="5">IEEE WCNC 2011 會議</td></tr>
<tr><td>出國期間</td><td colspan="2">自 100 年 3 月 26 日至 100 年 4 月 3 日</td><td>出國地點</td><td colspan="2">Cancun, Mexico</td></tr>
<tr><td>出國目的</td><td colspan="5">口頭發表論文</td></tr>
<tr><td>補助金額</td><td colspan="2">100,000</td><td>經費來源
（校內會計編號）</td><td colspan="2"></td></tr>
</table>

報告內容應包括下列各項：

一、出國經過

IEEE WCNC 2011 會議於 2011 年 3 月 28 日至 3 月 31 日於 Mexico Cancun 城市召開。會議有出論文集，會中並有多個 panel 及 session 召開。會議涵蓋之主要研究方向為 Communication, Internet protocol 及 computer network 之研究。 此會議是個大型的國際會議，約有 700 人參加。會議其間，本人與當地研究人員交談討論電腦通訊網路研究的最新趨勢。

二、心得（可含照片）

本 WCNC 會議可說是是 IEEE 電腦通訊網路界中數一數二相當具權威性的的電腦通訊網路國際會議。在會議中所發表的論文均為精選的佳作。除此之外，還有專家及 CEO 發表精闢的演講，令本人瞭解世界網路研究的最新趨勢，真是獲益良多。

三、考察參觀活動(無是項活動者，或前已敘述者可省略此項)

四、建議

本人很感謝此次的補助。出席國際性學術會議對研究學者是一種很大的鼓勵。能藉此增廣見聞、吸收新知、且與同一研究領域的學者互相切磋討論。另外補助學者出席國際性學術會議也能增加台灣在國際學術界的能見度及知名度，是一件重要的措施。

五、攜回資料名稱及內容

- 大會論文集 （CD）
- 一些展示系統的軟體 CD 片

六、其他

出席國際學術會議心得報告

計畫編號	99-3113-P-009-004
出國人員姓名 服務機關及職稱	趙禧綠 交通大學資工系助理教授
會議時間地點	2011/9/5~2011/9/8, San Francisco, USA
會議名稱	2011 IEEE 74th Vehicular Technology Conference: VTC2011-Fall
發表論文題目	Resource Allocation with Sum Throughput Improvement for LTE Uplink Transmission

一、參加會議經過

此次國際研討會共計四天，報告人的 session 在九月六日下午。該篇論文的主題是針對 LTE 上傳頻寬，設計提升總體吞吐量的資源配置演算法。由於我的指導學生申請到國科會補助，故學生一同前往參加會議，並且由學生進行口頭報告。雖然這是指導的學生第一次參與大型國際會議，但他事前多次的 rehearsal，讓他當天的報告口條很清楚。由於他在交大讀書時曾參與接待加州大學柏克萊分校的國際學生，所以他能聽懂當場與會者的發問。針對我們的論文，提出的問題有二，一是我們最佳化模型提出後，模擬在跑數據時的時間複雜度；二是我們的模擬程式是否在應用層真正執行如 ftp、http 等應用。針對這兩問題，由於最佳化模型並沒有找到一個有效的方法去 reduce 它，所以我們的程式採暴力法，需要的執行時間頗高。而上層的模擬我們是假設 UE 的資料都是一直持續在進來，所以並沒有跑一個真正的應用程式。儘管如此，與會者所提出的建議，我們會在後續的研究，試著改善並實現。

二、與會心得

與前幾個報告人今年參與的國際會議相比較，VTC2011-Fall 有很多議題與前幾個一致，例如 cognitive radio、vehicular networks 以及 LTE。然近年來被熱門討論的雲端議題，在 VTC2011-Fall 尚未看到。利用雲端強大的運算能力與儲存空間，是很多研究的趨勢。或許 VTC2012-Spring 我們就會看到這樣的 sessions 或 tutorial 出現了。

藉由在國際間分享研究與國內外學者交流，並聽取世界各地的研究報告以獲取新知，可以說是非常有收穫的一次行程。這次的與會，對我的學生影響甚大。他有機會與來自不同地方的學者/學生交換研究心得，增廣見聞。回來台灣後，他開心地跟實驗室學弟妹分享經驗，並鼓勵大家努力在自己的研究議題上，做出好的 work，爭取出國開會的機會，提升自身的能力。

國立交通大學
National Chiao Tung University

出國報告（出國類別： □A類、考察訪問
□B類、出國短期研究
■ C類國際會議）

會議名稱：IEEE Wireless Communications and Networking Conference (WCNC 2011)

服務機關：電信工程所
姓名職稱：邱榮東　博士生
前往國家：國家 城市 場所
出國期間：2011/03/24~04/06
報告日期：

撰　寫　人	審核人	初　閱	複　閱
邱榮東			

備註：出國報告書審核程序如下

一、 各學院教師 A、B、C 類及其他行政單位 A 類由單位主管，研究生由指導教授；中心計畫及學群 A、B、C 類由各中心計畫主持人。

二、 複閱：經費所屬之一級單位；中心計畫及學群 A、B、C 類由頂尖計畫執行長。

國立交通大學「發展國際一流大學及頂尖研究中心計畫」

C 類出國報告書

年　　月　　日

<table>
<tr><td rowspan="2">報告人姓名</td><td rowspan="2">邱榮東</td><td rowspan="2">申請單位
（學生請加註系級）</td><td rowspan="2">電信工程所
博士班一年級</td><td>職稱</td><td>博士生</td></tr>
<tr><td>電話</td><td>54557</td></tr>
<tr><td>出國目的/發表論文題目</td><td colspan="5">A BAN System for Realtime ECG Monitoring : from Wired to Wireless Measurements</td></tr>
<tr><td>補助金額</td><td colspan="2">28,000</td><td>經費來源
（校內會計編號）</td><td colspan="2"></td></tr>
</table>

報告內容應包括下列各項：

一、參加經過

IEEE WCNC(Wireless Communications and Networking Conference)是一個由Communication Society所舉辦的conference，對於通訊以及網路領域來說屬於國際一流之研討會。於去年十以將論文完成投稿後，今年三月底參加位於墨西哥坎昆所舉辦的會議。由於台灣直達墨西哥的班機甚少，旅行社亦無法代訂抵達坎昆的機票(大部份包含轉機只有到墨西哥市的)；因此機票部份需自行至國外的航空公司訂購。在考量價錢方面後，機票分成兩段訂購，分別是台灣至洛杉磯以及洛杉磯至坎昆兩段，同時再為了能夠省下週末航班的加價(五、六、日出發的班機票價會貴上一千至數千元台幣不等)，再配合會議時間，因此全部的行程時間拉的較長，從3/24出發而4/6抵達台灣。

在美國的期間，除了短暫在洛杉磯的停留外，主要是由位於Folsom的友人招待。期間參觀了Intel的Folsom分公司，以及距離不遠的大學UC Davis。而抵達坎昆後，第一天主要是tutorial和Welcom reception，第二～四天則是各自不同的Session。此次WCNC包含議題極廣，主要的議題可分為Physical Layer(PHY)、Medium Access Control(MAC)、Networking、和Services and Applications四大類，而每一大類的議題又會再細區分為好幾個小類，如PHY又包含了interference、Cooperative Communications、Coding、MIMO、OFDM等議題，MAC則有Scheduling、Multiple Access、Resource Management、Protocols等議題;Networking則包含了Ad-Hoc and Sensor Networks、Wireless Networking、Vehicular Networks、Routing等議題。至於Service and Applications則可說是無所不包，其他不屬於上列等的主流通訊問題幾乎都被分到這個項目來。

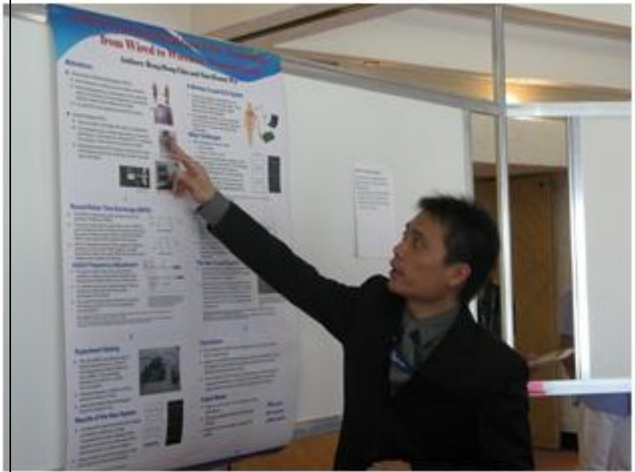
這次我論文的題目算是通訊和訊號處理在生醫領域的應用，因此被分到的是 Service 和 Application 這個主題當中；在這次的會議中，關於生醫領域的應用算是比較少人討論的金題，比如說身體區域網路(Body Area Network)相關的論文包含我的作品一共只有四篇，除了我的論文之外，一篇是韓國的 Inwa 大學的老師所做的，另外兩篇則是加拿大的沈書明老師所指導完成的論文，分別討論合作式通訊和安全性相關的議題在身體區域網路上的處理；遺憾的是完成兩篇論文的學生似乎都己畢業另謀高就了，因此報告是委託同一實驗室的其他同學代為報告，也無法有進一步的討論。

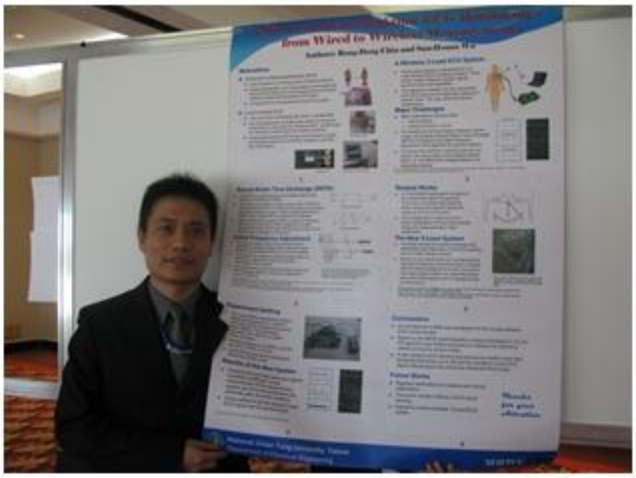
二、心得（可含照片）

WCNC 雖己屬一流的國際會議，被接受的論文品質都有一定程度的保證；然由於報告人的來自世界各地，報告的技巧、方式，以及英文的口音都不見得讓人容易了解。有趣的是通常對台灣人來說最容易聽懂的英文是美式的口音，然而英、美人似乎不管是什麼樣口音的英文如印度、日本等對我們來說很難聽懂的口音，仍能夠溝通自如。加上報告時間有限，一個人報告時間含問題討論時間限制在十五分鐘，因此聽報告時是否能夠抓到作者研究的精華，就真的是運氣了。通常歐美國家的報告者，講述的方式比較能夠讓人接受，其餘的通常得從投影片來猜出其關鍵的研究成果，或是事後再讀其論文了。

反而 Panel Discussion 的討論極為精采，討論的質量俱佳，且主題皆可說是未來通訊會面臨的挑戰與關鍵議題。而主要的報告者也不限學術界人士，更多的是相關的業界人士參與。如第二天的 Wireless Green Networking 就邀請了美國阿爾卡特朗遜貝爾實驗室、以及中國華為的高階主管參加，除了比較偏理論、模擬的研究成果外，也提到各大公司在各地所實際佈建綠能網路的計畫。有了實際建置的成果，對其論述才能夠產生更為有力的實證，這是一般在學界的人所無法做到的。

而 Poster 也是相對有趣的一個 session。每個論文作者站在各自的海報前，就像是業務員一樣的等待顧客上門講解。從中也可看出各研究團隊對於研究成果展現重視的程度。比如說有的人會將論文以 A4 格式印出，有的則是將投影片拼成海報；也有的是委託同學代為張貼；但若是實際作者有到場的，往往都能夠進行較為深入的討論，不管是就這個研究的出發點、應用、挑戰及困難等，都能夠有較為有趣且深入的了解。

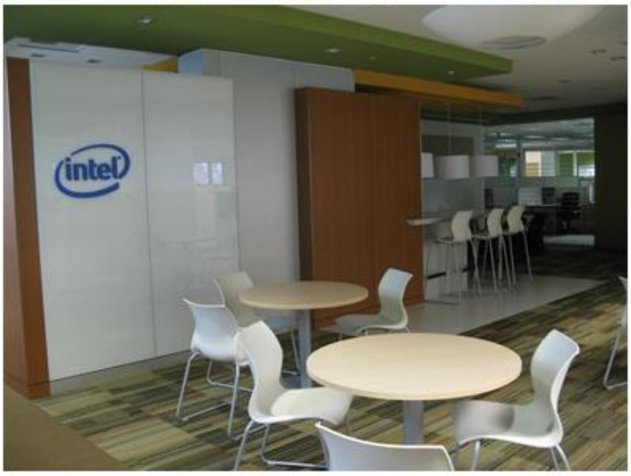




左上圖是在報告的過程,右上圖則是我和聽眾的合照。左下是和巴西友人在會議 Coffee Break 時的合照，右下則是我與海報的合照。

三、考察參觀活動(無是項活動者，或前已敘述者可省略此項)

在 WCNC 期間大會並無安排特別的參訪活動；必竟該處為渡假勝地，最熱門的活動是參訪馬雅文化的金字塔遺跡。倒是在美國時拜託友人安排下，有幸參訪了 Intel 位於 Folsom 的分公司。據友人轉述在 Intel 的待遇比起矽谷的科技公司算是相對較低的，但工作上的負擔也相對較輕鬆，同時公司內亦相當注重員工休閒活動；算是很能夠在生活、工作上取得平衡的一家公司。實際參訪完後，則對其工作空間、人性化的管理感到印象深刻，同時也反思以 Intel 這樣一家引領世界科技潮流一二十年的公司(近年來行動裝置當道的風潮似乎對其領導地位造成了極大挑戰，但在自九 0 年代以來至 2008 年左右其和 Windows 合稱的 Wintel 聯盟，不管是就營業額或是影響力而言，在科技業界都是站在絕對主導者的地位)，為何還能夠以人性化的工作方式來引領潮流，反觀台灣科技業則是以責任制、爆肝來換取低成本，甚至最近過勞死頻傳；而在中國大陸人力素質、成本以逐漸追上我國的同時，是否還能以這種壓低成本的方式繼續和世界各大公司競爭？我想這是值得我們好好深思的。



左上圖是入口處的照片，右上則是位於公司內的遊戲間。左下是一張 6-DRAK 的告示，一般而言雨司過了晚了六點就會關燈，要加班需要開燈的話則需打電話請管理員開啟。右下則是公司辦公室內部一角，有開放式的討論空間以及類及 BAR 台的設計，這類的開放式空間在 Intel 辦公室內部隨處可見，可見其貼心為員工著想之處。

四、建議

海報的論文其實在討論、互動方面往往能夠進行的較為深入。此外論文作者在海報 session 中往往需要不斷的覆述，以我個人為例在一個半小時 session 時間內的大概持續講解了一個小時左右，中間休息時間其實不長。而對此方面有興趣的人士，或者是研究有相關的人生，往往能夠直接針對一些關鍵議題多作討論，比如說合作的可能性以及相關技術最新發展等。而在教育部的補助中規定補助上限是一般口頭報告論文的 60%，個人是覺得此乃不合時宜之作法。

五、攜回資料名稱及內容

論文光碟及大會秩序冊

六、其他

國立交通大學博士班研究生

出席國際會議報告

報告人姓名	邱麟凱	報告日期	１００年０７月 １２ 日
系所及年級	電信工程所 博士班 96 級	核定文號	11D095
連絡電話	0917567798	電子信箱	sazabi.cm96g@nctu.edu.tw
會議期間	11/06/05~11/06/09	會議地點	日本　京都 國際會議中心
會議名稱	（中文）國際電機電子工程學會 國際通訊會議 2011 （英文） IEEE ICC 2011		
發表論文題目	（中文）以貝氏科拉姆·勞下界在通道追跡的觀點下分析基於訓練的多輸入輸出系統在時變衰減通道下可達到的傳輸率 （英文）The Achievable Rate of the Training-Based MIMO Systems over Time-Varying Fading Channels: A BCRB Perspective on Channel Tracking		

報告內容包括下列各項：

一、參加經過

這次是我第二次參與 IEEE International communication conference，也是我個人第三次參與國際大型的會議。不過在國科會的申請補助上並為獲得接收，也使得我在開會前不到六個星期才向教育部申請補助，直到會議結束後三星期才知道獲得教育部的肯定，提供完整的補助，很感謝教育部對我這次發表論文的支持。

這次會議在日本京都舉行，因三月的東北大地震以及所伴隨而來的核子發電廠的事故，會議官方單位在會議舉行前多次寄信以回應論文發表者和與會的人對於這些事件的疑慮，並表示京都方面並沒有受到核發事件的影響，所以會議將如期進行，但也考量到發表者的疑慮，官方也同意開放錄音的方式給那些不克參加的發表者。這也造成這次有很多的論文是由他人代為發表和使用錄音發表。

開會地點京都需由位於大阪的關西空港轉進入再藉由其它的交通工具轉達，京都方面並沒有受到東北大地震以及核發的影響，相較於東京日常生活上並沒有不便或有所限制，大會於會議的第一天舉辦

Tutorial 和 Workshop，Tutorial 包含 cooperative wireless communications、visible light communications、cognitive radio、passive optical networks、LTE and EPC、wireless mesh networks 以及 vehicular networking，而 Workshop 包含 heterogeneous network、smart grid communications、physical layer security 以及 game theory and resource allocation 等研究。當天的歡迎晚會大會也準備了京都當地有特色的食物為與會者接風表示歡迎。

會議的第二天到第四天這為期三天的議程主要為各投稿論文的發表、keynote 演講、贊助場商的展示以及 Business Forums，keynote 演講分別為 NTT Docomo 對於通訊的成長所計劃的應變動作、討論未來網路和服務，以及未來通訊技術的發展趨勢，此外會議第三天的 business forum 請了中華電信的執行長針對電信的轉變和挑戰來做演講。而在投稿論文發表上，這次特別多出來以往少見的 e-health 和 satellite space communication 的議題針對通訊結合醫療方面和衛星通訊的論文的發表。此外日本皇太子也於會議的第三天中午到會場致辭。

會議的最後一天則是只有 Tutorial 和 Workshop，不同於第一天的議題的有 MIMO detection、lower-powered and energy-harvesting integrated circuits 和 ad-hoc network。

二、心得

這次 ICC 的會議場地為京都國際會議中心，是京都主要用來舉辦大型會議的地點，也是聯合國於 1997 年 12 月 11 日簽定京都議定書的地方，座落於京都寶池旁，會場內設備先進且空間廣大，內部和外部的構造都十分華麗且氣派，由其是庭院和外部的設計上很有日本傳統的建築的風格。

ICC 屬通訊界國際型的大型重要會議，這次 ICC 會議中有不少台灣的學者、教授以及學生出席發表論文，甚至有請到中華電信的執行長到場演講，這對於台灣通訊領域在產業和學術界的發展有極度正面的意義。此外在會議期間還可以不時和他們討論會議所發表的論文和演講來交換意見及想法。

相較於去年所參加同樣為 IEEE 所舉辦的各人室內移動無線電通訊(personal indoor mobile radio communication, PIMRC)會議，在 ICC 上所發表的論文大多偏理論的結果，並沒有很多實作上或通訊應用上的論文，以下簡單的介紹幾篇為我覺得特別且有趣的論文：

1. K. Yang, D. Calin, C.-B. Chae and S. Yiu, “Distributed beam scheduling in muli-cell networks via auction over competitive markets”

   這篇論文討論一個在多細胞(multi-cell)的網路架構下的問題，當每個細胞的基地台備有許多指向性天線時，且不同基地台的指向天線會有互相干擾時，各基地台要如何去選擇各自所使用的天線才能不影響其它細胞基地台的通訊且這些細胞整體的通訊品質可以最好。這篇論文首先將所有基地台的所有天線分成好幾個集合，這個集合中的天線在同一個時間中只能有一個被使用，否則各細胞彼此的干擾就會太大，使整個系統的傳輸失敗。論文中證明當一根天線出現在不同的集合中三次以上，那這個問題的最佳解是一個無法在多項式時間內解出來的問題。所以，這一篇論提出了一個次佳的演算法來解答這個問題。

2. Y.-C. Chen, S.-H. Tsai and G. C.H. Chuang, “A joint codebook design for beamforming systems with transmit antenna selection”

這一篇論文討論一個在多輸入單輸出(Multiple input single output, MISO)系統架構下，對於傳送端做波束成形時碼書(codebook)的設計，有別於之前的文獻，這一篇論文的碼書有同時考慮到天線選擇的問題，也就是碼書在設計會根據通道的特性，將不需要使用的天線反應在碼字(codeword)上，如果有一個天線是不需被使用的話，那碼字上對應到此天線的數值就會是 0。此論文提出的碼書設計是屬於向量量化(vector quantization)的方式，並利用一般化的洛依德演算法(Generalized Lloyd Algorithm, GLA)的一種遞迴式的方式，在每一個遞迴中會先將上一次的結果中每個碼字裡數值很小的部分設為零，也就代表在這個碼字中不會使用到所對應到的天線，才開始這次的碼書設計，直到演算法收斂碼書不再變化。此外這一篇論文還導出使用所設計的碼書在符元錯誤率的下界。

除了投稿論文的發表外，我還去參觀了這次會議贊助商在通訊領域上的成果展示，其中 Panasonic 對於 60GHz 無線電展示了他們所設計的原型晶片，並利用此晶片在 60GHz 的頻帶下來無線傳輸 3D 電視影像。此晶片是針對 60GHz 無線電在無線區域網路中，業界所主導的規格 WiGig 所設計的，主要應用在行動裝置上，因為為行動裝置，所以可以有人為的方示使傳送端和接收端之間形成直視路徑(line-of-sight)，也因此在天線設計上只使用有四個平面天線所形成的相位陣列天線來提供大約 3 公尺的無線通訊，目前這個相位陣列天線在設計上是嵌在晶片的背片，並非在制做在晶片裡面，而 Panasonic 對於這一個產品的開發預計在二年內能將相位陣列天線制做在晶片裡頭。此外因為此晶片的設計只適用於直視路徑的無線傳輸，所以展示人員也有展示出當傳接收機之間有障礙物時，通訊品質就會變的不好現象。而 WiGig 本身主打用來克服直視路徑被阻擋時的適應性波束成形的技術並沒有在這次的展示中展出。

我這次的論文發表是在會議的第四天的最後一個時段，在發表過程中一切都很順利，沒有太大的問題，發表完後對於主持人有針對論文所提出問題，也能順利的回答。總結，我在這次參加 ICC 這個會議的經歷中收獲不少。

三、建議

藉由出國開會，對於學生來說是一個不錯的學習機會與場合，可以試著用英文來和不同國家的學者交流，一來可以訓練學生的語文能力或是給予想要把英文運用的更好的動力，再來可以了解目前在個自擅長的領域或是在通訊這一個大領域上有什麼新的議題發展和趨勢，教育部這個給予學生的輔助的方案可以促使學生願意出國開會增廣見聞的機會和意願，對於學術上的長久發展是一個不錯的政策，所以希望這個政策可以持續的執行，使更多的學生可以受益。

四、攜回資料名稱及內容

1. 這次 ICC 會議中所有發表的論文的隨身碟一個。
2. 這次 ICC 會議中含有 Tutorial：Cognitive radio：a practical solution for software-defined radio and dynamic spectrum access complexity 的隨身碟一個。
3. Panasonic 於 ICC 會場中對於 60GHz 室內無線電所展示之原型晶片的介紹文件。

六、其他

這次去日本，打破了我對於日本人英文不太好的長久迷思，這次所到之地，所對話到的日本人其英文的能力都非常的好，而且都能講出一口流利且發音不錯的英文，在日常生活上是用英文溝通也不太會有問題，實在是出乎我的意料。

出席國際學術會議心得報告

計畫編號	99-3113-P-009-004
出國人員姓名 服務機關及職稱	殷裕雄　　交大電信所 碩士生
會議時間地點	100/9/11~100/9/14
會議名稱	(中文)國際電子電機工程學會個人化室內行動通訊會議 (外文)IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2011)
發表論文題目	A Space-Time Precoded Hybrid Beamforming Architecture for Broadband Transmissions in 60GHz Radio

一、參加會議經過

此行在 PIMRC 2011 參加了幾場 keynote speech、plenary talk、panel discussion，並於會議第三天進行 oral presentation。會議行程中有許多來自包括美國、歐洲、亞洲的知名研究機構及頂尖大學的學者專家給的 talk、論讀、oral presentation、post 等，主題涵蓋了通訊系統、網路、多媒體、資訊理論、電腦通訊領域等，提供了一個很好的交流平台。Department of Homeland Security 的 Jalal Mapar 演講了 Emergency responder tracking monitoring technology 的主題，發展救難人員的通訊設備並結合 Honeywell 跟 Argon 系統來達到目標通訊品質。AT&T 的代表講的是有關 cloud computing 的 data visualization，使用 gamification 來做 data processing。而 Microsoft 的講者說明有關 cloud computing 的 offloading Cloud in the palm of your hands，在什麼時候決定要 offload、什麼東西需要 offload，programming effort issue 等，另外也提到 Microsoft 的 Project Hawaii。接著是 Nokia 的講者，主題是 Making sense of a zettabyte world，他提到 data processing 需要考慮 robustness、elasticity、還有 energy efficiency 的問題。綜合以上的 talk 可以看出未來通訊發展將走向 unlimited storage、unlimited computing、limited bandwidth、photonic speed、和 follow Moore’s law。

二、與會心得

此次參加 PIMRC 2011 研討會，在到達會議所在地城市加拿大多倫多的港邊堡壘後，依預定的議程進行報告所發表之論文題目“A Space-Time Precoded Hybrid Beamforming Architecture for Broadband Transmissions in 60GHz Radio”之 oral presentation。報告過程及結束後和與會相關領域的學者及產業界人士討論及交換意見擴大我的世界觀，並增加了對於 oral presentation 的掌握度。對於我來說是一個難得的經驗。

國立交通大學出國報告書

100 年 4 月 10 日

報告人姓名	王協源	申請單位 （學生請加註系級）	資訊工程系	職稱 電話	教授 5131550
出國類別	□考察 □訪問 □進修 □研究 ☑國際會議 □其他：__________				
會議/出國計畫名稱	IEEE WCNC 2011 會議				
出國期間	自 100 年 3 月 26 日至 100 年 4 月 3 日		出國地點	Cancun, Mexico	
出國目的	口頭發表論文				
補助金額	100,000		經費來源 （校內會計編號）		

報告內容應包括下列各項：

一、出國經過

IEEE WCNC 2011 會議於 2011 年 3 月 28 日至 3 月 31 日於 Mexico Cancun 城市召開。會議有出論文集，會中並有多個 panel 及 session 召開。會議涵蓋之主要研究方向為 Communication, Internet protocol 及 computer network 之研究。 此會議是個大型的國際會議，約有 700 人參加。會議其間，本人與當地研究人員交談討論電腦通訊網路研究的最新趨勢。

二、心得（可含照片）

本 WCNC 會議可說是是 IEEE 電腦通訊網路界中數一數二相當具權威性的的電腦通訊網路國際會議。在會議中所發表的論文均為精選的佳作。除此之外，還有專家及 CEO 發表精闢的演講，令本人瞭解世界網路研究的最新趨勢，真是獲益良多。

三、考察參觀活動(無是項活動者，或前已敘述者可省略此項)

四、建議

本人很感謝此次的補助。出席國際性學術會議對研究學者是一種很大的鼓勵。能藉此增廣見聞、吸收新知、且與同一研究領域的學者互相切磋討論。另外補助學者出席國際性學術會議也能增加台灣在國際學術界的能見度及知名度，是一件重要的措施。

五、攜回資料名稱及內容

- 大會論文集 （CD）
- 一些展示系統的軟體 CD 片

六、其他

出席國際學術會議心得報告

計畫編號	99-3113-P-009-004
出國人員姓名 服務機關及職稱	趙禧綠 交通大學資工系助理教授
會議時間地點	2011/9/5~2011/9/8, San Francisco, USA
會議名稱	2011 IEEE 74th Vehicular Technology Conference: VTC2011-Fall
發表論文題目	Resource Allocation with Sum Throughput Improvement for LTE Uplink Transmission

一、參加會議經過

此次國際研討會共計四天，報告人的 session 在九月六日下午。該篇論文的主題是針對 LTE 上傳頻寬，設計提升總體吞吐量的資源配置演算法。由於我的指導學生申請到國科會補助，故學生一同前往參加會議，並且由學生進行口頭報告。雖然這是指導的學生第一次參與大型國際會議，但他事前多次的 rehearsal，讓他當天的報告口條很清楚。由於他在交大讀書時曾參與接待加州大學柏克萊分校的國際學生，所以他能聽懂當場與會者的發問。針對我們的論文，提出的問題有二，一是我們最佳化模型提出後，模擬在跑數據時的時間複雜度；二是我們的模擬程式是否在應用層真正執行如 ftp、http 等應用。針對這兩問題，由於最佳化模型並沒有找到一個有效的方法去 reduce 它，所以我們的程式採暴力法，需要的執行時間頗高。而上層的模擬我們是假設 UE 的資料都是一直持續在進來，所以並沒有跑一個真正的應用程式。儘管如此，與會者所提出的建議，我們會在後續的研究，試著改善並實現。

二、與會心得

與前幾個報告人今年參與的國際會議相比較，VTC2011-Fall 有很多議題與前幾個一致，例如 cognitive radio、vehicular networks 以及 LTE。然近年來被熱門討論的雲端議題，在 VTC2011-Fall 尚未看到。利用雲端強大的運算能力與儲存空間，是很多研究的趨勢。或許 VTC2012-Spring 我們就會看到這樣的 sessions 或 tutorial 出現了。

藉由在國際間分享研究與國內外學者交流，並聽取世界各地的研究報告以獲取新知，可以說是非常有收穫的一次行程。這次的與會，對我的學生影響甚大。他有機會與來自不同地方的學者/學生交換研究心得，增廣見聞。回來台灣後，他開心地跟實驗室學弟妹分享經驗，並鼓勵大家努力在自己的研究議題上，做出好的 work，爭取出國開會的機會，提升自身的能力。

國立交通大學
National Chiao Tung University

出國報告（出國類別： ☐ **A 類、考察訪問**
☐ **B 類、出國短期研究**
■ **C 類國際會議）**

會議名稱：IEEE Wireless Communications and Networking Conference (WCNC 2011)

服務機關：電信工程所
姓名職稱：邱榮東　博士生
前往國家：國家 城市 場所
出國期間：2011/03/24~04/06
報告日期：

撰　寫　人	審核人	初　　閱	複　　閱
邱榮東			

備註：出國報告書審核程序如下

一、 各學院教師 A、B、C 類及其他行政單位 A 類由單位主管，研究生由指導教授；中心計畫及學群 A、B、C 類由各中心計畫主持人。

二、 複閱：經費所屬之一級單位；中心計畫及學群 A、B、C 類由頂尖計畫執行長。

國立交通大學「發展國際一流大學及頂尖研究中心計畫」

C 類出國報告書

年　　月　　日

報告人姓名	邱榮東	申請單位（學生請加註系級）	電信工程所博士班一年級	職稱	博士生
				電話	54557
出國目的/發表論文題目	A BAN System for Realtime ECG Monitoring : from Wired to Wireless Measurements				
補助金額	28,000		經費來源（校內會計編號）		

報告內容應包括下列各項：

一、參加經過

IEEE WCNC(Wireless Communications and Networking Conference)是一個由Communication Society所舉辦的conference，對於通訊以及網路領域來說屬於國際一流之研討會。於去年十以將論文完成投稿後，今年三月底參加位於墨西哥坎昆所舉辦的會議。由於台灣直達墨西哥的班機甚少，旅行社亦無法代訂抵達坎昆的機票(大部份包含轉機只有到墨西哥市的)；因此機票部份需自行至國外的航空公司訂購。在考量價錢方面後，機票分成兩段訂購，分別是台灣至洛杉磯以及洛杉磯至坎昆兩段，同時再為了能夠省下週末航班的加價(五、六、日出發的班機票價會貴上一千至數千元台幣不等)，再配合會議時間，因此全部的行程時間拉的較長，從3/24出發而4/6抵達台灣。

在美國的期間，除了短暫在洛杉磯的停留外，主要是由位於Folsom的友人招待。期間參觀了Intel的Folsom分公司，以及距離不遠的大學UC Davis。而抵達坎昆後，第一天主要是tutorial和Welcom reception，第二～四天則是各自不同的Session。此次WCNC包含議題極廣，主要的議題可分為Physical Layer(PHY)、Medium Access Control(MAC)、Networking、和Services and Applications四大類，而每一大類的議題又會再細區分為好幾個小類，如PHY又包含了interference、Cooperative Communications、Coding、MIMO、OFDM等議題，MAC則有Scheduling、Multiple Access、Resource Management、Protocols等議題;Networking則包含了Ad-Hoc and Sensor Networks、Wireless Networking、Vehicular Networks、Routing等議題。至於Service and Applications則可說是無所不包，其他不屬於上列等的主流通訊問題幾乎都被分到這個項目來。

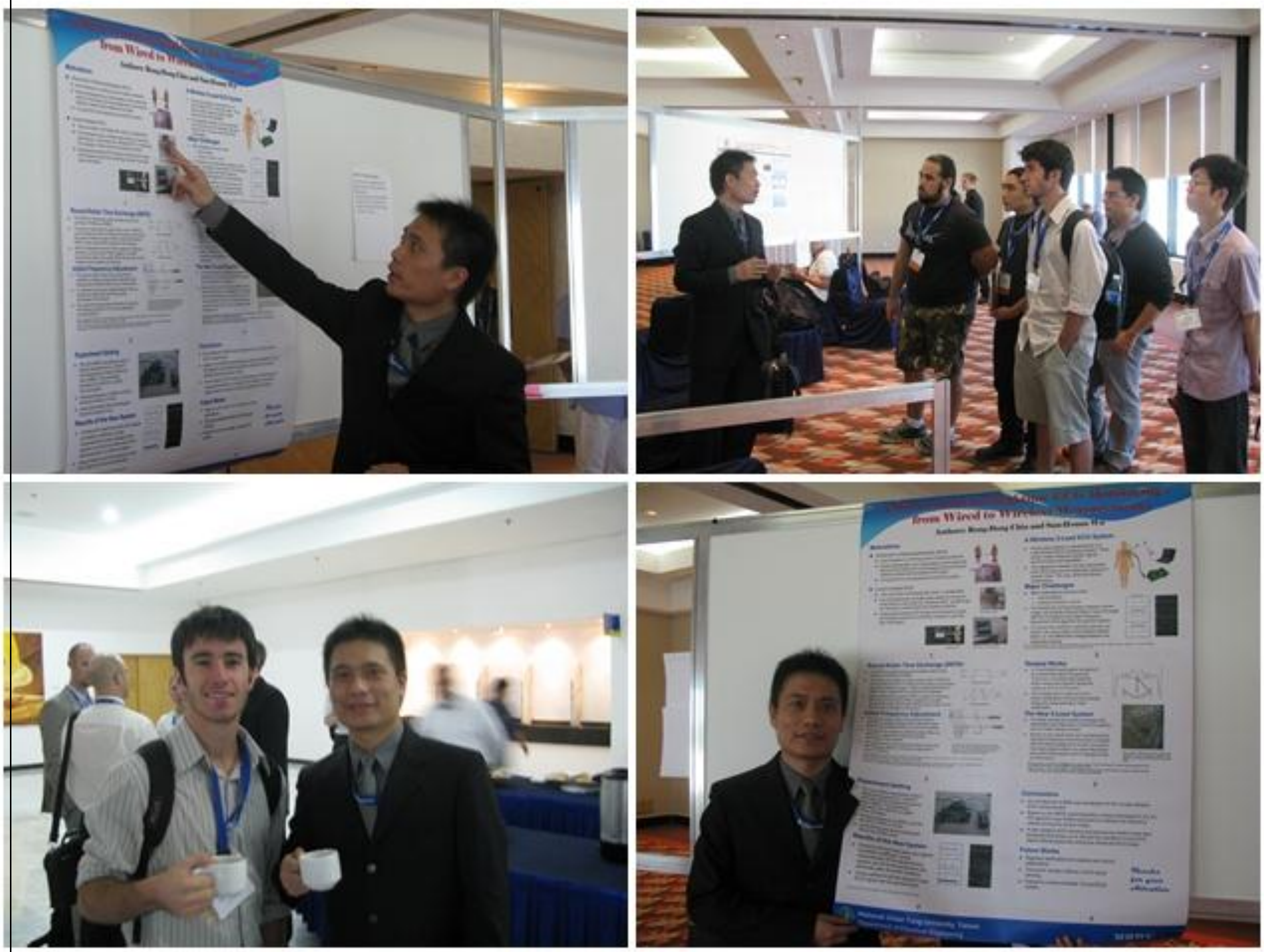
這次我論文的題目算是通訊和訊號處理在生醫領域的應用，因此被分到的是 Service 和 Application 這個主題當中；在這次的會議中，關於生醫領域的應用算是比較少人討論的金題，比如說身體區域網路(Body Area Network)相關的論文包含我的作品一共只有四篇，除了我的論文之外，一篇是韓國的 Inwa 大學的老師所做的，另外兩篇則是加拿大的沈書明老師所指導完成的論文，分別討論合作式通訊和安全性相關的議題在身體區域網路上的處理；遺憾的是完成兩篇論文的學生似乎都己畢業另謀高就了，因此報告是委託同一實驗室的其他同學代為報告，也無法有進一步的討論。

二、心得（可含照片）

WCNC 雖己屬一流的國際會議，被接受的論文品質都有一定程度的保證；然由於報告人的來自世界各地，報告的技巧、方式，以及英文的口音都不見得讓人容易了解。有趣的是通常對台灣人來說最容易聽懂的英文是美式的口音，然而英、美人似乎不管是什麼樣口音的英文如印度、日本等對我們來說很難聽懂的口音，仍能夠溝通自如。加上報告時間有限，一個人報告時間含問題討論時間限制在十五分鐘，因此聽報告時是否能夠抓到作者研究的精華，就真的是運氣了。通常歐美國家的報告者，講述的方式比較能夠讓人接受，其餘的通常得從投影片來猜出其關鍵的研究成果，或是事後再讀其論文了。

反而 Panel Discussion 的討論極為精采，討論的質量俱佳，且主題皆可說是未來通訊會面臨的挑戰與關鍵議題。而主要的報告者也不限學術界人士，更多的是相關的業界人士參與。如第二天的 Wireless Green Networking 就邀請了美國阿爾卡特朗遜貝爾實驗室、以及中國華為的高階主管參加，除了比較偏理論、模擬的研究成果外，也提到各大公司在各地所實際佈建綠能網路的計畫。有了實際建置的成果，對其論述才能夠產生更為有力的實證，這是一般在學界的人所無法做到的。

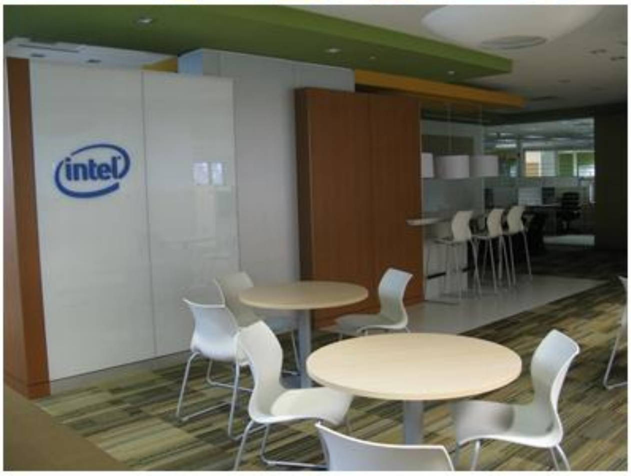
而 Poster 也是相對有趣的一個 session。每個論文作者站在各自的海報前，就像是業務員一樣的等待顧客上門講解。從中也可看出各研究團隊對於研究成果展現重視的程度。比如說有的人會將論文以 A4 格式印出，有的則是將投影片拼成海報；也有的是委託同學代為張貼；但若是實際作者有到場的，往往都能夠進行較為深入的討論，不管是就這個研究的出發點、應用、挑戰及困難等，都能夠有較為有趣且深入的了解。



左上圖是在報告的過程,右上圖則是我和聽眾的合照。左下是和巴西友人在會議 Coffee Break 時的合照，右下則是我與海報的合照。

三、考察參觀活動(無是項活動者，或前已敘述者可省略此項)

在 WCNC 期間大會並無安排特別的參訪活動；必竟該處為渡假勝地，最熱門的活動是參訪馬雅文化的金字塔遺跡。倒是在美國時拜託友人安排下，有幸參訪了 Intel 位於 Folsom 的分公司。據友人轉述在 Intel 的待遇比起矽谷的科技公司算是相對較低的，但工作上的負擔也相對較輕鬆，同時公司內亦相當注重員工休閒活動；算是很能夠在生活、工作上取得平衡的一家公司。實際參訪完後，則對其工作空間、人性化的管理感到印象深刻，同時也反思以 Intel 這樣一家引領世界科技潮流一二十年的公司(近年來行動裝置當道的風潮似乎對其領導地位造成了極大挑戰，但在自九 0 年代以來至 2008 年左右其和 Windows 合稱的 Wintel 聯盟，不管是就營業額或是影響力而言，在科技業界都是站在絕對主導者的地位)，為何還能夠以人性化的工作方式來引領潮流，反觀台灣科技業則是以責任制、爆肝來換取低成本，甚至最近過勞死頻傳；而在中國大陸人力素質、成本以逐漸追上我國的同時，是否還能以這種壓低成本的方式繼續和世界各大公司競爭？我想這是值得我們好好深思的。



左上圖是入口處的照片，右上則是位於公司內的遊戲間。左下是一張 6-DRAK 的告示，一般而言雨司過了晚了六點就會關燈，要加班需要開燈的話則需打電話請管理員開啟。右下則是公司辦公室內部一角，有開放式的討論空間以及類及 BAR 台的設計，這類的開放式空間在 Intel 辦公室內部隨處可見，可見其貼心為員工著想之處。

四、建議

海報的論文其實在討論、互動方面往往能夠進行的較為深入。此外論文作者在海報 session 中往往需要不斷的覆述，以我個人為例在一個半小時 session 時間內的大概持續講解了一個小時左右，中間休息時間其實不長。而對此方面有興趣的人士，或者是研究有相關的人生，往往能夠直接針對一些關鍵議題多作討論，比如說合作的可能性以及相關技術最新發展等。而在教育部的補助中規定補助上限是一般口頭報告論文的 60%，個人是覺得此乃不合時宜之作法。

五、攜回資料名稱及內容

論文光碟及大會秩序冊

六、其他

國立交通大學博士班研究生

出席國際會議報告

報告人姓名	邱麟凱	報告日期	１００年０７月 １２ 日
系所及年級	電信工程所 博士班 96 級	核定文號	11D095
連絡電話	0917567798	電子信箱	sazabi.cm96g@nctu.edu.tw
會議期間	11/06/05~11/06/09	會議地點	日本　京都 國際會議中心
會議名稱	（中文）國際電機電子工程學會 國際通訊會議 2011 （英文） IEEE ICC 2011		
發表論文題目	（中文）以貝氏科拉姆‧勞下界在通道追跡的觀點下分析基於訓練的多輸入輸出系統在時變衰減通道下可達到的傳輸率 （英文）The Achievable Rate of the Training-Based MIMO Systems over Time-Varying Fading Channels: A BCRB Perspective on Channel Tracking		

報告內容包括下列各項：

一、參加經過

這次是我第二次參與 IEEE International communication conference，也是我個人第三次參與國際大型的會議。不過在國科會的申請補助上並為獲得接收，也使得我在開會前不到六個星期才向教育部申請補助，直到會議結束後三星期才知道獲得教育部的肯定，提供完整的補助，很感謝教育部對我這次發表論文的支持。

這次會議在日本京都舉行，因三月的東北大地震以及所伴隨而來的核子發電廠的事故，會議官方單位在會議舉行前多次寄信以回應論文發表者和與會的人對於這些事件的疑慮，並表示京都方面並沒有受到核發事件的影響，所以會議將如期進行，但也考量到發表者的疑慮，官方也同意開放錄音的方式給那些不克參加的發表者。這也造成這次有很多的論文是由他人代為發表和使用錄音發表。

開會地點京都需由位於大阪的關西空港轉進入再藉由其它的交通工具轉達，京都方面並沒有受到東北大地震以及核發的影響，相較於東京日常生活上並沒有不便或有所限制，大會於會議的第一天舉辦

Tutorial 和 Workshop，Tutorial 包含 cooperative wireless communications、visible light communications、cognitive radio、passive optical networks、LTE and EPC、wireless mesh networks 以及 vehicular networking，而 Workshop 包含 heterogeneous network、smart grid communications、physical layer security 以及 game theory and resource allocation 等研究。當天的歡迎晚會大會也準備了京都當地有特色的食物為與會者接風表示歡迎。

會議的第二天到第四天這為期三天的議程主要為各投稿論文的發表、keynote 演講、贊助場商的展示以及 Business Forums，keynote 演講分別為 NTT Docomo 對於通訊的成長所計劃的應變動作、討論未來網路和服務，以及未來通訊技術的發展趨勢，此外會議第三天的 business forum 請了中華電信的執行長針對電信的轉變和挑戰來做演講。而在投稿論文發表上，這次特別多出來以往少見的 e-health 和 satellite space communication 的議題針對通訊結合醫療方面和衛星通訊的論文的發表。此外日本皇太子也於會議的第三天中午到會場致辭。

會議的最後一天則是只有 Tutorial 和 Workshop，不同於第一天的議題的有 MIMO detection、lower-powered and energy-harvesting integrated circuits 和 ad-hoc network。

二、心得

這次 ICC 的會議場地為京都國際會議中心，是京都主要用來舉辦大型會議的地點，也是聯合國於 1997 年 12 月 11 日簽定京都議定書的地方，座落於京都寶池旁，會場內設備先進且空間廣大，內部和外部的構造都十分華麗且氣派，由其是庭院和外部的設計上很有日本傳統的建築的風格。

　　ICC 屬通訊界國際型的大型重要會議，這次 ICC 會議中有不少台灣的學者、教授以及學生出席發表論文，甚至有請到中華電信的執行長到場演講，這對於台灣通訊領域在產業和學術界的發展有極度正面的意義。此外在會議期間還可以不時和他們討論會議所發表的論文和演講來交換意見及想法。

　　相較於去年所參加同樣為 IEEE 所舉辦的各人室內移動無線電通訊(personal indoor mobile radio communication, PIMRC)會議，在 ICC 上所發表的論文大多偏理論的結果，並沒有很多實作上或通訊應用上的論文，以下簡單的介紹幾篇為我覺得特別且有趣的論文：

1. K. Yang, D. Calin, C.-B. Chae and S. Yiu, “Distributed beam scheduling in muli-cell networks via auction over competitive markets”

　　這篇論文討論一個在多細胞(multi-cell)的網路架構下的問題，當每個細胞的基地台備有許多指向性天線時，且不同基地台的指向天線會有互相干擾時，各基地台要如何去選擇各自所使用的天線才能不影響其它細胞基地台的通訊且這些細胞整體的通訊品質可以最好。這篇論文首先將所有基地台的所有天線分成好幾個集合，這個集合中的天線在同一個時間中只能有一個被使用，否則各細胞彼此的干擾就會太大，使整個系統的傳輸失敗。論文中證明當一根天線出現在不同的集合中三次以上，那這個問題的最佳解是一個無法在多項式時間內解出來的問題。所以，這一篇論提出了一個次佳的演算法來解答這個問題。

2. Y.-C. Chen, S.-H. Tsai and G. C.H. Chuang, “A joint codebook design for beamforming systems with transmit antenna selection”

這一篇論文討論一個在多輸入單輸出(Multiple input single output, MISO)系統架構下，對於傳送端做波束成形時碼書(codebook)的設計，有別於之前的文獻，這一篇論文的碼書有同時考慮到天線選擇的問題，也就是碼書在設計會根據通道的特性，將不需要使用的天線反應在碼字(codeword)上，如果有一個天線是不需被使用的話，那碼字上對應到此天線的數值就會是 0。此論文提出的碼書設計是屬於向量量化(vector quantization)的方式，並利用一般化的洛依德演算法(Generalized Lloyd Algorithm, GLA)的一種遞迴式的方式，在每一個遞廻中會先將上一次的結果中每個碼字裡數值很小的部分設為零，也就代表在這個碼字中不會使用到所對應到的天線，才開始這次的碼書設計，直到演算法收斂碼書不再變化。此外這一篇論文還導出使用所設計的碼書在符元錯誤率的下界。

除了投稿論文的發表外，我還去參觀了這次會議贊助商在通訊領域上的成果展示，其中 Panasonic 對於 60GHz 無線電展示了他們所設計的原型晶片，並利用此晶片在 60GHz 的頻帶下來無線傳輸 3D 電視影像。此晶片是針對 60GHz 無線電在無線區域網路中，業界所主導的規格 WiGig 所設計的，主要應用在行動裝置上，因為為行動裝置，所以可以有人為的方示使傳送端和接收端之間形成直視路徑(line-of-sight)，也因此在天線設計上只使用有四個平面天線所形成的相位陣列天線來提供大約 3 公尺的無線通訊，目前這個相位陣列天線在設計上是嵌在晶片的背片，並非在制做在晶片裡面，而 Panasonic 對於這一個產品的開發預計在二年內能將相位陣列天線制做在晶片裡頭。此外因為此晶片的設計只適用於直視路徑的無線傳輸，所以展示人員也有展示出當傳接收機之間有障礙物時，通訊品質就會變的不好現象。而 WiGig 本身主打用來克服直視路徑被阻擋時的適應性波束成形的技術並沒有在這次的展示中展出。

我這次的論文發表是在會議的第四天的最後一個時段，在發表過程中一切都很順利，沒有太大的問題，發表完後對於主持人有針對論文所提出問題，也能順利的回答。總結，我在這次參加 ICC 這個會議的經歷中收獲不少。

三、建議

藉由出國開會，對於學生來說是一個不錯的學習機會與場合，可以試著用英文來和不同國家的學者交流，一來可以訓練學生的語文能力或是給予想要把英文運用的更好的動力，再來可以了解目前在個自擅長的領域或是在通訊這一個大領域上有什麼新的議題發展和趨勢，教育部這個給予學生的輔助的方案可以促使學生願意出國開會增廣見聞的機會和意願，對於學術上的長久發展是一個不錯的政策，所以希望這個政策可以持續的執行，使更多的學生可以受益。

四、攜回資料名稱及內容

1. 這次 ICC 會議中所有發表的論文的隨身碟一個。
2. 這次 ICC 會議中含有 Tutorial：Cognitive radio：a practical solution for software-defined radio and dynamic spectrum access complexity 的隨身碟一個。
3. Panasonic 於 ICC 會場中對於 60GHz 室內無線電所展示之原型晶片的介紹文件。

六、其他

這次去日本，打破了我對於日本人英文不太好的長久迷思，這次所到之地，所對話到的日本人其英文的能力都非常的好，而且都能講出一口流利且發音不錯的英文，在日常生活上是用英文溝通也不太會有問題，實在是出乎我的意料。

出席國際學術會議心得報告

計畫編號	99-3113-P-009-004
出國人員姓名 服務機關及職稱	殷裕雄　　交大電信所 碩士生
會議時間地點	100/9/11~100/9/14
會議名稱	(中文)國際電子電機工程學會個人化室內行動通訊會議 (外文)IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2011)
發表論文題目	A Space-Time Precoded Hybrid Beamforming Architecture for Broadband Transmissions in 60GHz Radio

一、參加會議經過

此行在 PIMRC 2011 參加了幾場 keynote speech、plenary talk、panel discussion，並於會議第三天進行 oral presentation。會議行程中有許多來自包括美國、歐洲、亞洲的知名研究機構及頂尖大學的學者專家給的 talk、論譠、oral presentation、post 等，主題涵蓋了通訊系統、網路、多媒體、資訊理論、電腦通訊領域等，提供了一個很好的交流平台。Department of Homeland Security 的 Jalal Mapar 演講了 Emergency responder tracking monitoring technology 的主題，發展救難人員的通訊設備並結合 Honeywell 跟 Argon 系統來達到目標通訊品質。AT&T 的代表講的是有關 cloud computing 的 data visualization，使用 gamification 來做 data processing。而 Microsoft 的講者說明有關 cloud computing 的 offloading Cloud in the palm of your hands，在什麼時候決定要 offload、什麼東西需要 offload，programming effort issue 等，另外也提到 Microsoft 的 Project Hawaii。接著是 Nokia 的講者，主題是 Making sense of a zettabyte world，他提到 data processing 需要考慮 robustness、elasticity、還有 energy efficiency 的問題。綜合以上的 talk 可以看出未來通訊發展將走向 unlimited storage、unlimited computing、limited bandwidth、photonic speed、和 follow Moore’ s law。

二、與會心得

此次參加 PIMRC 2011 研討會，在到達會議所在地城市加拿大多倫多的港邊堡壘後，依預定的議程進行報告所發表之論文題目“A Space-Time Precoded Hybrid Beamforming Architecture for Broadband Transmissions in 60GHz Radio”之 oral presentation。報告過程及結束後和與會相關領域的學者及產業界人士討論及交換意見擴大我的世界觀，並增加了對於 oral presentation 的掌握度。對於我來說是一個難得的經驗。

國科會補助計畫衍生研發成果推廣資料表

日期:2012/02/23

國科會補助計畫	計畫名稱: 電視頻帶白空間之寬頻行動雲端感知無線網路
	計畫主持人: 王協源
	計畫編號: 99-3113-P-009-004- 學門領域: 其他

無研發成果推廣資料

99 年度專題研究計畫研究成果彙整表

<table>
<tr><td colspan="3">計畫主持人：王協源</td><td colspan="5">計畫編號：99-3113-P-009-004-</td></tr>
<tr><td colspan="8">計畫名稱：電視頻帶白空間之寬頻行動雲端感知無線網路</td></tr>
<tr><td colspan="3" rowspan="2">成果項目</td><td colspan="3">量化</td><td rowspan="2">單位</td><td rowspan="2">備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）</td></tr>
<tr><td>實際已達成數（被接受或已發表）</td><td>預期總達成數(含實際已達成數)</td><td>本計畫實際貢獻百分比</td></tr>
<tr><td rowspan="14">國內</td><td rowspan="4">論文著作</td><td>期刊論文</td><td>0</td><td>0</td><td>100%</td><td rowspan="3">篇</td><td></td></tr>
<tr><td>研究報告/技術報告</td><td>2</td><td>2</td><td>100%</td><td></td></tr>
<tr><td>研討會論文</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
<tr><td>專書</td><td>0</td><td>0</td><td>100%</td><td></td><td></td></tr>
<tr><td rowspan="2">專利</td><td>申請中件數</td><td>2</td><td>2</td><td>100%</td><td rowspan="2">件</td><td></td></tr>
<tr><td>已獲得件數</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
<tr><td rowspan="2">技術移轉</td><td>件數</td><td>0</td><td>0</td><td>100%</td><td>件</td><td></td></tr>
<tr><td>權利金</td><td>0</td><td>0</td><td>100%</td><td>千元</td><td></td></tr>
<tr><td rowspan="4">參與計畫人力（本國籍）</td><td>碩士生</td><td>21</td><td>0</td><td>100%</td><td rowspan="4">人次</td><td></td></tr>
<tr><td>博士生</td><td>7</td><td>0</td><td>100%</td><td></td></tr>
<tr><td>博士後研究員</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
<tr><td>專任助理</td><td>1</td><td>0</td><td>100%</td><td></td></tr>
<tr><td rowspan="14">國外</td><td rowspan="4">論文著作</td><td>期刊論文</td><td>0</td><td>0</td><td>100%</td><td rowspan="3">篇</td><td></td></tr>
<tr><td>研究報告/技術報告</td><td>3</td><td>3</td><td>100%</td><td></td></tr>
<tr><td>研討會論文</td><td>4</td><td>4</td><td>100%</td><td></td></tr>
<tr><td>專書</td><td>0</td><td>0</td><td>100%</td><td>章/本</td><td></td></tr>
<tr><td rowspan="2">專利</td><td>申請中件數</td><td>2</td><td>2</td><td>100%</td><td rowspan="2">件</td><td></td></tr>
<tr><td>已獲得件數</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
<tr><td rowspan="2">技術移轉</td><td>件數</td><td>0</td><td>0</td><td>100%</td><td>件</td><td></td></tr>
<tr><td>權利金</td><td>0</td><td>0</td><td>100%</td><td>千元</td><td></td></tr>
<tr><td rowspan="4">參與計畫人力（外國籍）</td><td>碩士生</td><td>0</td><td>0</td><td>100%</td><td rowspan="4">人次</td><td></td></tr>
<tr><td>博士生</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
<tr><td>博士後研究員</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
<tr><td>專任助理</td><td>0</td><td>0</td><td>100%</td><td></td></tr>
</table>

其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)	無

	成果項目	量化	名稱或內容性質簡述
科教處計畫加填項目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與（閱聽）人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

   ■達成目標

   □未達成目標（請說明，以 100 字為限）

   　　□實驗失敗

   　　□因故實驗中斷

   　　□其他原因

   說明：

2. 研究成果在學術期刊發表或申請專利等情形：

   論文：■已發表 □未發表之文稿 □撰寫中 □無

   專利：□已獲得 ■申請中 □無

   技轉：□已技轉 □洽談中 ■無

   其他：（以 100 字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）