

Multi-channel Scheduling (MCS) for IEEE 802.11 CSMA-based Wireless Networks

Wei-Hsu Su, Ting-Yu Lin, and Guang-Chuen Yin
Department of Communication Engineering
National Chiao-Tung University

摘要

IEEE 802.11 DCF 是採用 CSMA/CA 機制透過競爭的方式來取得媒介的使用權，當競爭者越來越多，DCF 的效能會因為碰撞的機會變高而逐漸的下降；DCF 主要用於單一頻道，多通道的環境可以將兩個發送源分散在不同的正交頻道上而不會干擾到彼此，在現有的技術下，過度的切換頻道反而會使得整體的效能降低；MCS (Multi-channel Scheduling) 是一個建立在多頻道環境下的鏈結層管理方式，目的在於建立一個無斷裂的連結網路以及根據網路流量的分佈來挑選一個適合傳輸的頻道來減少節點間的干擾以及減少頻道切換的次數，MCS 能夠有效的利用多頻道的特性來增加網路整體的吞吐量，並且在延伸到更多數的收發器時，我們加入了收發器排程的關連性來加深網路的連結性，以及加速各個偵測頻道的速度來讓頻道狀態保持在可使用的狀態。

關鍵詞: ad-hoc wireless networks, multi-channel, channel diversity

1. 前言

無線區域網路 (WLAN: Wireless Local Area Network) 的規範中以 IEEE 802.11 最為普遍，802.11 中制定 physical layer 和 MAC layer (Medium Access Control) 協定，從 1999 年提出的 802.11b 和 802.11a 到近年的 802.11y，802.11 系列中定義了多個傳輸頻道，如 802.11b 定義在 2.4GHz ISM 頻帶，總共有 11 個頻道，每個頻道有 22MHz 的頻寬，802.11a 則使用 5GHz 的頻帶並有 13 個頻道 (有些國家則是 12 或是 14 個頻道)，若是頻道可以獨自的傳送資料而不影響彼此的傳送，這些頻道被稱做為 "orthogonal channel" 或者是 "independent channel"，封包即可以傳輸在不同的 orthogonal channel 上並且不受彼此的干擾；大多數的無線網路標準包含了多個 orthogonal channel，在

802.11b 中包含了 11 個頻道，其中有 3 個為 orthogonal channel，分別為 ch1、ch6、ch11，802.11a 則全為 orthogonal channel；802.11 DCF (Distributed Coordination Function) 為一個常用的分散式協調功能，主要探討單一頻道時節點如何取得頻道的使用權，因此 802.11 定義許多的頻道，卻沒辦法有效發揮 multi-channel 的優點，圖(1-1a)為典型的 multi-channel 範例，節點 A 和節點 C 分別有資料要傳送給節點 B 和節點 D，在單一頻道環境節點 A 和節點 C 會干擾對方的傳輸，為了成功傳送封包，則必須要將節點 A 節點 C 有先後順序的傳送；相對的，圖(1-1b)在 multi-channel 環境下，節點 C 和節點 D 可以切換到另一個頻道傳輸而節點 A 和節點 B 則可以留在原本的頻道。我們主要研究一個分散式的媒介存取方式 (MAC) 來運用 multi-channel 的優點，以增加網路整體的吞吐量 (network aggregate throughput)。在許多探討 multi-channel MAC 的文獻中，常見的議題有兩個：

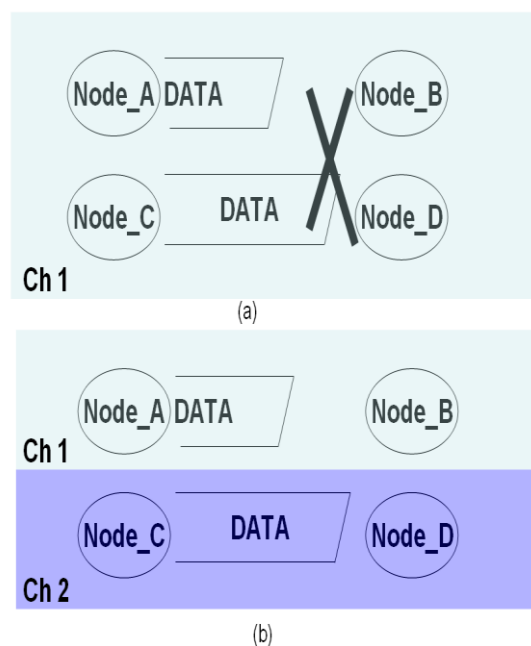


圖 1-1 Multi-channel property

1.) 連結性 (connectivity):

一對通訊節點只能在同一個頻道上作傳送和接收，若是兩個節點沒辦法切換到相同的頻道上，即使在彼此的傳送距離之內，也無法成功傳送封包；確保節點和周遭的節點可以切換到相同的頻道，能夠維持網路的連結性，而不會因為 multi-channel 的特性造成節點間的連結斷裂或是網路分裂成多個無法互通的小型網路 (network partitions)，是這類多通道研究需要特別注意的議題。

2.) 選擇傳送頻道 (transmission channel selection): 當一個節點成功傳送 control packet 給接收端時，下個動作就是要決定傳送封包的頻道；兩個以上的傳送端同時傳送在同一個頻道，則有可能發生碰撞 (collision)，大部分的無線網路介面卡 (NIC: Network Interface Card) 為半雙工 (half-duplex)，在傳送封包時沒辦法偵測封包是否遭受到碰撞，傳送端必須等待來自接收端的回應才能判斷封包是否傳送成功；在 multi-channel 環境下，則可以允許多個傳輸同時進行在不同的頻道中，因此在多頻道環境中，傳送頻道的選擇方式會直接影響到整體網路的效能。

另一個多通道環境常見的問題為: multi-channel hidden terminal problem，DCF 利用 virtual carrier sensing 來防止 hidden terminal problem，而在 multi-channel 的環境下，因為節點可能因切換頻道而無法持續監控原頻道，有可能 virtual carrier sensing 並沒辦法成功通知該節點，結果 hidden terminal problem 依舊會發生；一個簡單的做法是當節點切換使用頻道時，必須先等待一個最大封包長度傳輸時間 (transmission time of maximum-size packet) 才能進行媒介競爭，如此一來 hidden terminal problem 將可被適當防止。

不同於以往的文獻中假設收發器切換接收頻道的時間 (switching delay) 可以被忽略，在文獻 [7] 中提到，隨著頻道切換系統所需要花費的時間在數個毫秒 (millisecond)，相對於封包傳輸時間 (亦可能數毫秒)，switching delay 影響效能顯著；在本篇論文中，我們提出了一個 partition-proof multi-channel scheduling (MCS) 演算法，可以保證網路的連結性；在 MCS 考慮的環境中，switching latency 是個不可被忽略的要素，其中 switching latency 包含了切換頻道後所等待的最大封包長度的傳輸時間，以及系統切換頻道的時間 (processing time)，並且，MCS 針對各個頻道的使用程度

(channel load) 來分配傳輸對 (transmission pair) 所在的頻道，讓 CSMA/CA 可以保持較好的效能。本篇論文剩餘的部份則為: 第二章節介紹已被提出的 multi-channel protocols，並簡單敘述運作過程和特性；第三章節說明有關於 random sequence 的代數特性，並且於第四章節詳細說明有關於 MCS 的運作過程和原理；第五章節為模擬驗證結果，最後為第六章的未來展望和第七章的總結。

2. 相關文獻

許多文獻提出使用 multi-channel 可以提高網路的效能，根據控制封包的交換方式，我們可以將這些文獻分類成兩類: (1) single rendezvous: 所有的節點固定地會集中在相同的頻道上來交換訊息，一次只能有一組控制封包能夠被交換，但是所有的節點皆可以聆聽 (overhear) 控制封包；(2) multiple rendezvous: 控制封包的交換是分散各個頻道上，多組的控制封包可以被交換在不同的頻道中，而每個控制封包的交換只有部分的節點可以接收到。

2.1 Single Rendezvous

這類型的 protocol 有 MMAC、DCA，這類型特色在於控制封包的交換時，所有的節點都會聚集在同一個頻道中聆聽，但是在 802.11 控制封包中並沒有包含傳輸頻道的資訊，因此 single rendezvous 類型的方法皆需要修改封包格式或是協商方式來達到 multi-channel 的控制；各個節點藉著聆聽控制封包來掌握所有頻道的使用情況，當一個傳送端嘗試要傳送時，則可以挑選一個沒被使用的頻道避免產生碰撞；MMAC [2] 設定固定時間內，節點會切換到固定的頻道，DCA [5] 則採用一個固定的頻道來交換並且使用另外一個收發器來接收或傳送控制封包。

2.1.1 Multi-channel MAC (MMAC)

MMAC 的節點配有一個半雙工收發器，節點知道可使用頻道數目，並且節點間要求有時間同步 (time synchronization)；MMAC 針對 IEEE 802.11 省電模式 (PSM: Power Saving Mechanism) 來設計，在 PSM 時間被分為眾多的 beacon interval，一個 beacon interval 分為前段的 ATIM window (Ad hoc Traffic Indication Messages) 和剩餘的後段時間，所有的節點在 ATIM window 時會切換到一個 common channel 來進行控制封包的交換，並且 ATIM window

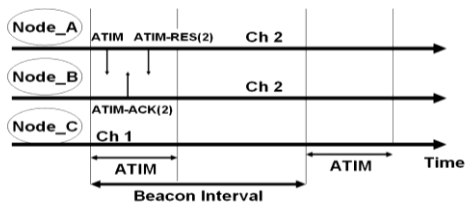


圖 2-1 MMAC

結束後切換到選定的頻道來進來資料的傳輸。圖(2-1)為一個 MMAC 運作的例子，在 ATIM 開始的時候，節點 A、節點 B 和節點 C 會切換到頻道一 (ch 1)，節點 A 想要傳送封包給節點 B，因此節點 A 在 ATIM window 時候傳送 ATIM 封包給節點 B，其中 ATIM 封包包含了節點 A 的 PCL (Preferable Channel List)；節點 B 透過節點 A 的 PCL 和自身的 PCL 判斷並且使用頻道二 (ch2) 接著回覆一個 ATIM-ACK(2)給節點 A，節點 A 則在送出一個 ATIM-RES(2)通知周圍的節點，等到 ATIM window 結束，節點 A 和節點 B 則切換到頻道二進行資料傳輸。

2.1.2 Dynamic Channel Assignment (DCA)

相對於 MMAC 用時間來分配資料封包和控制封包的傳送，DCA 的節點額外裝備一個收發器，並且從可以用頻道中取出一個頻道當作控制頻道 (control channel)，其餘的頻道則當作資料頻道，DCA 的節點上有兩個收發器，一個節點固定在控制頻道上用來傳輸和接收控制封包，而另外一個收發器能夠自由的切換在資料頻道來傳送資料圖 2-2；DCA 的節點會維持 CUL (Channel Usage List)來記錄節點的鄰居使用頻道的情形，並且動態的計算出 FCL (Free Channel List)；節點 A 在控制頻道上傳送 RTS 給節點 B，在 RTS 封包中包含了節點 A 的 FCL，節點 B 則尋找一個與節點 A 共通的 free channel 來當作傳送頻道並將選定的頻道放在 CTS 中傳回給節點 A，節點 A 收到 CTS 後，一方面將傳送資料的收發器切換到選定的頻道，另一方面在控制頻道上透過 RES 封包來通知週遭的節點使得節點 A 的鄰居節點可以更新它們的 CUL。

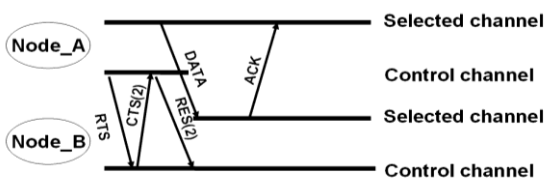


圖 2-2 DCA

2.2 Multiple Rendezvous

multiple rendezvous 沒有一個特定的控制頻道用來傳送控制封包，每個頻道都可以被用來傳送控制封包和資料封包。這類別的 protocol [1, 3, 6]會將所有的節點分散在各個頻道上，節點則可以針對所在的頻道進行競爭，針對單一頻道的媒介存取可以沿用 802.11 協定。

2.2.1 Hybrid Multi-channel Protocol (HMCP)

HMCP [1, 8]的節點至少需要兩個收發器，如同 DCA，HMCP 將所有的收發器分成兩個群組，其中一個群組為”fixed interfaces”，另一個群組為”switchable interfaces”；屬於 fixed interfaces 的收發器會固定在由節點選定的頻道，這些頻道被稱作”fixed channel”，頻道的挑選會因節點的不同而不一樣；節點廣播”Hello messages”交換 fixed channel，每個節點則會維持一個”Neighbor Table”紀錄其他節點的 fixed channel，傳送端和接收端要是沒有相同的 fixed channel，則傳送端會切換 switchable interface 到接收端的其中一個 fixed channel 上，之後傳送端和接收端即可在該 fixed channel 通訊。

2.2.2 Slotted Seeded Channel Hopping (SSCH)

SSCH [3]節點配有一個收發器，節點間要求時間同步；SSCH 使用頻道排程 (channel schedule)來安排收發器所在的位置，節點想要傳送給某個節點時，會使用接收端的頻道行程來取代原本的頻道行程，如此一來傳送端和接收端可以保持在相同的頻道上以便於傳輸。SSCH 的收發器會在頻道上停留一個固定的時間，每個節點使用多組的 (channel, seed)配對來構成頻道行程表，收發器跟隨著頻道排程來切換頻道，每組 (channel, seed)可以展開成一個 random sequence，任意兩個使用不同 seed 的 random sequence 會週期性的出現一個相同的項 (element)，若是兩個節點對應的 random sequence 均使用相同的 seed，將會使得這兩個節點無法聯繫，造成網路的斷裂；為了防止網路分裂，SSCH 另外增設一個 parity slot 加入排程中；圖(2-3)用來說明 SSCH 的運作，節點 A

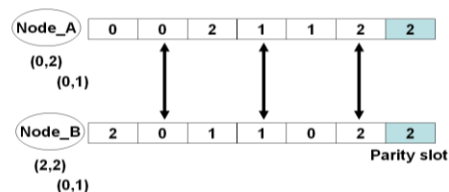


圖 2-3 SSCH

擁有兩個 (channel, seed)對，第 1、3、5 個 slot 由第一組 (channel, seed)來決定，2、4、6 slot 由第二組決定，圖中節點 A 和節點 B 的第二組配對皆為 (0,1)，因此節點 A 和節點 B 在第 2、4、6 slot 收發器所在的頻道式相同的，節點 A、B 可以利用這些 overlapping slot 來進行協商。

single rendezvous 可以簡單的讓週遭的節點聆聽到控制封包，使得所有的節點容易挑選狀況好的頻道進行傳輸，只是單一的控制頻道會成為網路效能的瓶頸 (bottleneck)；multiple rendezvous 方式可以令許多的協商在不同的頻道上進行。在 [4] 文章中分析，multiple rendezvous 擁有較好的效能；HMCP 有可能形成 traffic load 分配不均，而 SSCH 經常的切換傳輸頻道能將 traffic load 分散在各個頻道上，但是過度頻繁的切換頻道會受到 switching latency 的影響。

3. Random Sequence Properties

MCS 依據 random sequence 來當作頻道的排程，在這個章節我們將介紹有關 random sequence 的特性和特色。在上個章節提到每組 (channel, seed)可以展開成為一個 random sequence，每一個 random sequence 包含了 p 個元素，其中 p 為一個質數，random sequence 的特性則可以由 Galois field 來推導，接著我們稍微的介紹 Galois field，在第二小節則會介紹 random sequence 的性質。

3.1 Galois Field

對一個質數 p ，考慮集合 $S = \{0, 1, 2, \dots, p-1\}$ 和兩個運算子 (operator)加法“ \oplus ”和乘法“ \odot ”，分別稱作模數 P 加法和模數 P 乘法；在這樣的定義下， $\{S, \oplus, \odot\}$ 可以形成一個有限元素場 (field)，稱之為有限場 (finite field)或是 Galois field，以 $GF(p)$ 表示。

$$\forall a, b \in S, c = a \oplus b \triangleq a + b \text{ mod } p$$

$$\forall d, e \in S - \{0\}, f = d \odot e \triangleq d * e \text{ mod } p$$

Galois field 內含有加法恆等元素、乘法恆等元素 (Identity)，並且符合結合律 (associative law)，分配律 (distributive law)等性質。

性質一：

模數 P 加法和模數 P 乘法具有結合律。

$$\forall \alpha, \beta, \gamma \in S$$

$$\text{addition : } \alpha \oplus (\beta \oplus \gamma) = (\alpha \oplus \beta) \oplus \gamma$$

$$\text{multiplication : } \alpha \odot (\beta \odot \gamma) = (\alpha \odot \beta) \odot \gamma$$

性質二：

$GF(p)$ 存在兩個恆等元素，分別為加法恆等元素以及乘法恆等元素；“0”表示加法恆等元素，“1”表示乘法恆等元素。

$$\forall \alpha \in S, \alpha \oplus 0 = 0 \oplus \alpha = \alpha$$

$$\forall \beta \in S - \{0\}, \beta \odot 1 = 1 \odot \beta = \beta$$

$$\text{and } \forall \gamma \in S, \gamma \odot 0 = 0 \odot \gamma = 0$$

性質三：

S 中的所有元素對於模數 P 加法皆具有加法反元素，使得兩者相加會等於加法恆等元素；所有元素除了加法恆等元素“0”外，對於模數 P 乘法均有一個乘法反元素，使得兩者相乘會等於乘法恆等元素；以 $-\alpha$ 來表示 α 的加法反元素，以 β^{-1} 表示 β 的乘法反元素，反元素具有唯一性。

$$\forall \alpha \in S, \exists -\alpha \alpha \oplus (-\alpha) = (-\alpha) \oplus \alpha = 0$$

$$\forall \beta \in S - \{0\}, \exists \beta^{-1} \beta \odot \beta^{-1} = \beta^{-1} \odot \beta = 1$$

性質四 (分配性)：

$$\forall \alpha, \beta, \gamma \in S \quad \alpha \odot (\beta \oplus \gamma) = (\alpha \odot \beta) \oplus (\alpha \odot \gamma)$$

性質五 (封閉性)：

在 S 中的元素，任意對模數 P 加法和模數 P 乘法做運算所得到的結果依然為一個屬於 S 的元素。

性質六 (除法和減法)：

在 Galois field 內僅有定義加法和乘法，減法和除法則分別為加上加法反元素和乘上乘法反元素，因為加法恆等元素“0”對於模數 P 乘法沒有乘法反元素，所以除法不能對“0”做運算。

$$\forall \alpha, \beta \in S$$

$$\alpha - \beta = \alpha \oplus (-\beta)$$

$$\alpha \div \beta = \alpha \odot (\beta)^{-1} \quad \beta \neq 0$$

圖(3-1)為 $GF(3)$ 中各個元素對加法和乘法的對照表，可以由圖得知元素“2”在 $GF(3)$ 的加法反元素為元素“1”，乘法反元素為元素“2”，而且可以驗證任何運算的結果依然為一個小於三的非負整數。

⊕	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Addition of GF(3)

⊙	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Multiplication of GF(3)

圖 3-1 Illustration of addition and multiplication of GF(3)

3.2 Random Sequence

介紹完 GF(p)的基本代數性質，我們將用來解釋 random sequence 的特性；如同前面所述 random sequence 可以使用一組 (channel, seed) 來完全的描述，其中 channel 和 seed 均為小於 p 的非負整數，也就是說 channel 和 seed 都是集合 $S = \{0, 1, 2, \dots, p-1\}$ 裡面的一個元素，channel 代表現在的狀態 (current state)，seed 則用來決定下一個狀態 (next state)。

在 SSCH 中使用：

$$\text{channel} \leftarrow (\text{channel} + \text{seed}) \bmod p \quad \text{---(1)}$$

來決定一個 random sequence，而公式(1)可以改寫成：

$$\text{channel} \leftarrow (\text{channel} \oplus \text{seed}) \quad \text{---(2)}$$

例如在 GF(7)中，一組 (channel, seed) 為 (0,3)，則 (0,3) 所代表的 sequence 為 $\{0, 3, 6, 2, 5, 1, 4, 0, 3, 6, \dots\}$ ；(channel, seed) 會隨著狀態的改變而不斷的更新，為了說明方便，我們將改變 SSCH 對於 (channel, seed) 中 channel 的定義方式為 random sequence 的第一個項，於是 random sequence 的第 N 個項可以表示為：

$$\text{channel} \leftarrow (\text{channel} \oplus (\text{seed} \odot (N-1))) \quad \text{---(3)}$$

任意 random sequence 具有以下的性質：

性質一 (週期性)：

任意 random sequence 具有週期性，其週期為 p 。channel = (channel \oplus (seed \odot p))；因為 random sequence 以 p 為週期，因此 random sequence 可以表示成為一個 p 次元向量 ($\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{p-1}$)，後面文章提到向量表示時則將以 random sequence vector 來表示。

性質二 (循環性)：

將任一 random sequence 作循環位移後仍然為一個 random sequence。

性質三：

任意兩個擁有不同 seed 的 random sequence (χ_A, α_A)、(χ_B, α_B) 在一個週期內，會有一個相同的項存在；假設 (χ_A, α_A) 對 (χ_B, α_B) 會在第 $K+1$ 項有相同的元素，則：

$$\forall \chi_A, \chi_B, \alpha_A, \alpha_B \in S$$

$$\chi_A \oplus \alpha_A \odot K = \chi_B \oplus \alpha_B \odot K$$

$$K = (\chi_A \oplus (-\chi_B)) \odot (\alpha_B \oplus (-\alpha_A))^{-1}$$

因為 α_A 不等於 α_B ，所以 K 存在且唯一，根據封閉性 K 必為屬於 S 的元素，因此 K 為一個小於 p 的非負整數；如果 α_A 等於 α_B ，則 K 為無限多解或是無解；要是 χ_A 等於 χ_B 則兩個 random sequence 表示的是同一個 sequence，因此 K 有無限多解，否則 K 為無解，這兩個 random sequence 不會有相同的項存在。

4. Multi-channel Scheduling

MCS 為一個建立在 CSMA/CA 機制設計的分散式 link-layer 頻道管理方式 (channel management)；經由頻道排程來將網路上的節點分散在不同的頻道上，各頻道的媒介存取則是透過 802.11 DCF 的方式來競爭取得，利用傳送政策 (transmission policy) 減少頻道切換的次數以及頻道負荷偵測 (channel loading detection) 來避免各頻道間的負荷相差太大，並且擴充不同規模的網路延伸到任意質數的頻道和任意數量的收發器。

環境參數的假設為 1) 有 p 個頻道可以用來傳送資料，其中 p 是一個質數，並且各個頻道的條件是相同的；2) 每個節點備有收發器數量是相同的，每個節點備有 i 個收發器；3) 每個收發器皆為半雙工 (half-duplex) 並且可以自由的切換頻道，一個半雙工的收發器一次只能在一個頻道上傳送或是接收；4) 收發器從任一頻道切換到另一個頻道所需要的花費的時間 (switching latency: T_{sd}) 不會因為兩頻道相差的頻率而有所不同。

4.1 802.11 DCF (Distributed Coordination Function)

802.11 無線網路是共用廣播的媒介，一旦有個節點在進行傳送，則以此傳送端 (假設為節點 A) 為球心半徑 d 所形成的球體內節點皆會收到訊號， d 稱作干擾範圍 (interference range)；此時若是另一個傳送端 (假設為節點 C) 也要傳送封包給另外一個接收端 (假設為節點 D)，並且節點 B (節點 A 對應的接收端) 在節點 C 的干擾範圍內，則在節點 B 收到的訊號則有可能受到碰撞 (collision) 而無法成功的解讀

封包；因此 DCF 使用了兩個機制來防止 collision 和 hidden terminal problem 的發生，一個是硬體的 PCS (Physical Carrier Sensing)，一個是使用軟體的 VCS (Virtual Carrier Sensing)；PCS 運作為當一個節點嘗試傳送之前，會先偵測頻道上的訊號強度，要是訊號強度大於一個臨界值，則視為有另一個傳送端正在傳送中而不進行傳送，直到頻道空閒 (idle: 訊號強度小於臨界值) 一段 IFS (Inter-frame Space) 的長度時，節點會從 contention window 內 $[0, cw-1]$ 隨機挑選一個 backoff timer 倒數，一旦倒數到零即將封包傳送出去；VCS 的運作為在傳送資料封包 (DATA packet) 之前，先送出一個較短的控制封包 RTS (Request-To-Send) 給接收端，接收端則回應一個 CTS (Clear-To-Send)；在 RTS、CTS 和 DATA 封包中有嵌入一個時間參數 NAV (Network Allocation Vector)，當其他的節點聆聽到 RTS、CTS 或是 DATA 封包時會靜止 NAV 所記載的長度而等到 NAV 過去後才會嘗試著做傳送。

表 3-1 Parameter definition

Parameter	Description
p	Number of available orthogonal channels.
i	Number of interfaces equipped at a node.
T_{sd}	Process time of interface switching from a channel to another.
T_{slot}	Time interval of a interface stay in a channel including switching latency.
(χ_A, α_A)	χ_A is the starting channel of node A. (first element of random sequence of node A). α_A is the hopping offset of node A.
$\{\beta, (\chi_A, \alpha_A)\}$	β : Seed-dependent channel element. A representation form of modified random sequence.
T_p	Duration of a sequence cycle.

4.2 Channel Scheduling

我們先將討論的範圍限定在 $i = 1$ 的情形，這表示說網路中的每個節點僅僅配有一張半雙工的網路卡；頻道標示為 0 到 $p-1$ ，每個節點會選定一個行程表 (schedule) 給這個收發

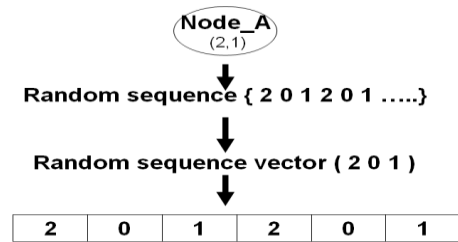


圖 4-1 Simple random sequence

器去使用，這個行程表必須要透過廣播給所有的鄰居知道；若是這個行程表為無限長度，那麼大部分的網路資源將會消耗在交換行程表，而導致網路效能的下降，所以這個行程表將會是一個有限長度並且重複使用的行程表；當一個節點 (假設為節點 A) 知道其他鄰居 (假設為節點 B) 的行程表，節點 A 則可以根據節點 B 的行程表得知何時能夠與節點 B 處於相同頻道，進而和節點 B 做溝通。在 MCS 中，我們挑選 random sequence 來當作我們的行程表，因為一個 random sequence 可以壓縮成一組 (channel, seed) 配對來完全的描述，這對於行程表的交換有非常大的幫助。如圖(4-1)中節點 A 挑選了 (2,1) 來當作行程表，其餘的節點僅需要接收到 (2,1) 訊息即可以完全的掌握到節點 A 的頻道變化 (以 $p = 3$ 當例子)。

我們以 (χ_A, α_A) 來表示節點 A 所挑選的 random sequence，其中 χ_A 和 α_A 都不大於 p 的非負整數；若是 α_A 挑選到 "0" 則代表節點 A 不會切換頻道 (使用在不具有頻道切換功能的收發器)，節點 A 的收發器會一直固定在頻道 χ_A 上，雖然 seed 為 0 的節點不會跳動，但它依然擁有 random sequence 的性質：兩個擁有不同 seed 的 random sequence 會有一個相同的項；但是兩個擁有相同 seed 的 random sequence 卻不一定會有交集，因此我們必須添增幾個只和 seed 有關 (seed-dependent) 的元素來達到我們想要的連結性，在 $i = 1$ 的情況，我們只需要新增一個元素 β 在 random sequence vector 的最前端，random sequence vector 則改寫成 $(\beta, \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{p-1})$ ，以 modified random sequence vector 來稱呼； β 的產生方式必須要每個節點都一樣才能確保兩個相同 seed 的節點產生出來的 β 是相同的，簡單的方法則是讓 β 和 seed 相同或者是 seed 加上一個 offset，modified random sequence 則以 $\{\beta, (\chi_A, \alpha_A)\}$ 來表示，而週期也將變成為 $p+1$ ，因為每個節點均使用同一個 β 產生方式，所以這個修改並不會增加所要廣播的資訊量；一旦行程表決定後，將 $p+1$

個 time slots 依序的對應 modified random sequence vector 中各個元素， β 擺放在 slot 1， v_0 擺放在 slot 2以此類推，每個元素所代表的數字 $[0, p-1]$ 對應要一個實體層 (physical) 的頻道；在 802.11b 有三個 orthogonal channel 分別為 ch1、ch6、ch 11，則集合 $S = \{0, 1, 2\}$ 的各個元素可以一對一的對應，用 element 0 代表 ch 1、element 1 和 element 2 分別對應到 ch6 和 ch 11。我們假設各節點間是有時間同步 (time synchronized) 的，並且有一個共通的 slot 來擺放 β ；每 $p+1$ 個 time slot 為一個 cycle，根據 random sequence 特性，兩個節點在一個 cycle 內至少會有一個 overlapping slot (所在的時間和頻道皆是相同的)。

每個節點會定期的廣播 beacon 封包在各個頻道上，封包中包含它的行程表給它的鄰居節點；雖然 modified random sequence 可以簡化成為一組 $\{\beta, (\text{channel}, \text{seed})\}$ 配對，但是我們不需要送出完整的一組 $\{\beta, (\text{channel}, \text{seed})\}$ 配對就可以將完整的行程表傳送出去，MCS 中限制只有節點遵守它的行程表時才會送出 beacon 封包，當節點所在的 slot 不是 seed-dependent slot (β 所對應的 slot)，beacon 封包內夾帶的資訊為 seed；反之，beacon 封包內夾帶的資訊則為 channel；在 seed-dependent slot 每個節點會切換到 β 所對應的頻道，相同的 seed 會計算出相同的 β 值，因此擁有相同 seed 的兩個節點會切換到同一個頻道上，seed 在此時不是個未知數，所以 beacon 中只要傳送 channel 的資訊即可；同樣的，在兩個節點交會在非 seed-dependent slot 表示這兩個節點擁有不同的 seed，所以將 beacon 封包內放入 seed 的資訊，根據接收到的 seed 和現在所在的頻道就能推出一個完整的行程表；每個節點會維持一個行程資料表 (RSTable)，資料表內包含了節點所使用的行程表，接收到鄰近節點傳送的 beacon 封包就會更新資料表。每隔一段時間節點就會檢查資料表，並選定一個較少使用的 $\{\beta, (\text{channel}, \text{seed})\}$ 配對來當作新的行程表。

圖(4-2)是一個小型的無線網路，總共有七

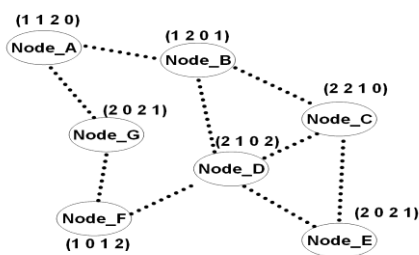


圖 4-2 Example network

個節點 A、B、C、D、E、F、G。並且有三個 orthogonal channel (ch 0, ch 1, ch 2)，每個節點所使用的 modified random sequence 標記在節點的旁邊；假設節點 D 是一個新加入的節點，並且已經選定 (2 1 0 2) 當作它的行程表，現在節點 D 要廣播它的行程表給週遭的節點 B、C、E、F；在 slot 1 時，節點 D、C 和節點 E 會切換到 ch 2 (令 $\beta = \alpha$)，節點 D 將 channel(D) = 1 包含在 beacon 封包中傳送出去，節點 C 和節點 E 收到節點 D 使用的 channel 值為 1，則可以推斷節點 D 的行程表為 $\{2, (1, 2)\} = (2 1 0 2)$ ；在 slot 3 時，節點 D 與節點 B 會交會在 ch 0，此時節點 D 僅需要傳送 seed(D) = 2 給節點 B，節點 B 利用 seed(D) = 2、slot 3 所在的頻道為 ch 0 則可以反推算節點 D 使用的起始頻道為 1， $0 = \alpha_D \oplus 2 \odot 1 \rightarrow \alpha_D = 0 - 2 \odot 1 \rightarrow \alpha_D = 1$ ；同樣的，節點 F 會根據 slot 4 ch 2 上收到節點 D 傳來的訊息 seed(D) = 2 來推斷節點 D 的起始頻道為 1，其它的節點也是遵循一樣的模式將行程表廣播出去。

4.3 Transmission Policy

使用 modified random sequence 可以防止網路分裂，並且形成一個 multiple rendezvous 的網路，但是一個節點可以會有一個以上的目的地 (destination)，若是使用單一 FIFO (First In First Out) 佇列 (queue) 將會產生 head-of-line 阻塞的問題；比方說：節點 A 有兩個目的地，分別稱做節點 B 和節點 C，節點 A 和節點 B 交會在 slot 2，而節點 A 和節點 C 交會在 slot 4，給節點 B 和節點 C 的封包是交錯的進入佇列中，那麼節點 A 在 slot 2 傳送一個封包給節點 B 之後必須要等到下一個週期才能夠傳送下一個封包給節點 B，如此一來將會浪費大部分的時間在等待。MCS 採用 burst negotiation 的方式減緩 head-of-line 阻塞的發生，並且減少切換頻道造成的損失 (switching latency)；我們準備了 $N+1$ 個 FIFO queue， N 為鄰居節點數， N 個 queue 按照不同的鄰居來存放封包，剩下的一個 queue 則是用來存放 broadcast 封包，節點依照目的地佇列的封包數來決定傳送順序的先後；每個封包對於媒介的存取是使用 802.11 DCF 的方式來競爭 (MCS 中使用 4-way handshaking，即使用 RTS/CTS)，考慮到網路擁塞的程度和 slot size，一個佇列的封包不一定有辦法在一個 slot 內傳送完畢，此時傳送端和接收端則會暫時的脫離它們的行程表保持在傳輸的通道上並且持續的做傳送直到佇列淨空為止；我們並設定一個傳輸高標 (Tx_high)

和一個最大允許傳輸長(max_slot)，一旦傳送成功的封包數大於 Tx_high 或者是總傳送時間超過 max_slot 則會將這個 slot 結束時傳送端和接收端也結束這次的傳送；來不及傳送出去的封包則必須下一次的協商才能傳送出去；要是佇列內的封包可以在這兩個限制內傳送完畢，傳送端會傳送一個我們自行定義的 RET(Hopping return)封包給接收端，目的在告知接收端這次的傳輸只到這個 slot 的結尾，RET 也可以由接收端主動來發送，使用 burst negotiation 主要目的在於減少傳輸時頻道切換的次數降低切換頻道時引發的 switching latency 和 head-of-line 阻塞。由上面的行為可以得知，MCS 是一個接收端為主(receiver-based)的方式。

4.4 Channel Load Detection

modified random sequence 可以將各個節點間的連結分散在各個 slot 和頻道中間，這個特性適合用在流量均勻分布的網路，但是不是所有的網路都能夠利用這個好特性，在圖(4-3)的網路拓撲，節點 D 和節點 E 在 slot 1 時開始傳輸，節點 B 和節點 C 的傳輸開始於 slot 2，節點 A 和節點 G 則是開始於 slot 3，雖然這三個傳輸開始的時間皆不一樣，但是單單透過行程表交會的頻道來傳輸，三條流量皆被分配到使用 ch 2 來傳送；即使網路中包含 3 個可使用的頻道，這三個傳輸卻只使用到一個，而沒有辦法獲得任何的頻道多樣性 (channel diversity)。

為了防止這類的頻道負荷 (channel loading)分配不均勻的問題發生，我們加入偵測頻道負荷的機制，並且允許節點依據頻道狀態來暫時性的改變行程表，在各個 slot 內各個收發器會收集該停留頻道上所接收到的 NAV 訊息，802.11 DCF 會將封包傳送估計的時間 (NAV)夾帶在 RTS、CTS 和 DATA 封包中；頻道狀態則以加總後的 NAV 來當作指標；各個

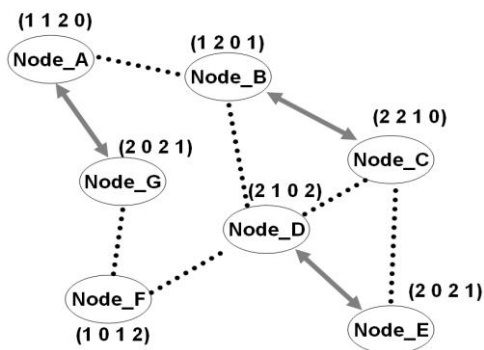


圖 4-3 Unbalanced channel utilization

節點會維持一個頻道狀況的資料表，其中包含了各個頻道的 NAV 總和以及每個頻道的記數器，每個 slot 的開始會把所有的記數器加一，並且開始收集停留頻道的封包，在 slot 結束時則將 NAV 總和更新到對應的頻道欄位並將該頻道的記數器歸零；當一個傳送端想要傳送封包之前，會計算出與接收端的 overlapping channel 並且檢查頻道狀態，要是該頻道狀態大於設定的參數 Nav_Threshold 或是記數器大於兩倍 cycle 長度加一(2p + 3)時，傳送端則會放棄使用這個頻道並且根據接收端的行程表來找一個最接近的頻道，這頻道的狀態必須要在 Nav_Threshold 以下，一旦頻道的記數器大於 (2p + 3)，我們將這個頻道狀態視為過時的；Nav_Threshold 則是所有非過時頻道狀態的平均值，當一個節點 (假設為節點 A)想要傳送封包給另一個節點 (假設為節點 B)時，難免會發生節點 B 暫時脫離行程表，這個時候節點 A 因為不知節點 B 行程表的短暫變更，所以會在一個節點 B 接收不到的頻道嘗試做傳送而導致傳送失敗；節點 A 發現傳送失敗會依照 802.11 重新傳送的機制將封包丟掉，並在重新挑選一個新的頻道來嘗試做傳送在，切換頻道的過程中節點 A 會尋找節點 B 發出來的封包，只要節點 A 的封包存在在某個頻道上，節點 A 則會加入該頻道的競爭對節點 B 發送封包。

圖(4-4)為一個 MCS 例子，節點 A{1, (2, 1)} 想要傳送封包給節點 B{2, (1, 2)}。根據節點 A 和節點 B 的行程表來計算，兩節點將會有一個交會在 slot 3 也就是 ch 0 上面，但是節點 A 去查看它自身的頻道狀態資料表發現 ch 0 的頻道狀態大於平均值，因此節點 A 決定找一個新的頻道來傳送；節點 A 根據節點 B 的行程表來逐一的檢驗頻道是否可以被使用，ch 1 則是一個被新挑選出來的傳送頻道，因此在 slot 2 時，節

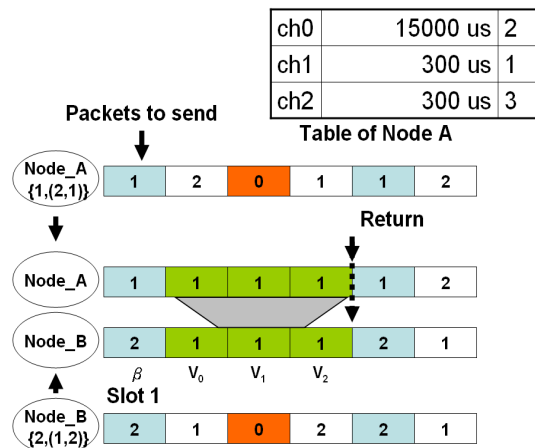


圖 4-4 MCS

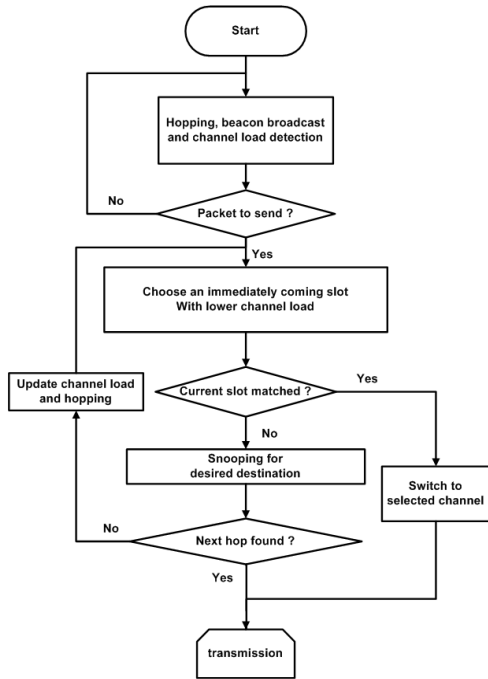


圖 4-5 System flow chart

點 A 改變了自己的行程表由原本的 ch 2 改為 ch 1 去符合節點 B，在 ch 1 上，節點 A 發送 RTS 給節點 B 並且開始做協商，當節點 A 和節點 B 成功傳送封包後則會留在 ch 1 繼續的做傳送直到 RET 封包出現或是到達兩個限制：Tx_high 和 max_slot。

流程圖 4-5 可以用來說明 MCS 在單一收發器時的運作；當節點於靜止狀態時會依照它的行程表來做頻道的切換，目的是要收集各個頻道的 NAV 以及廣播自身的程表表給週遭的節點；一旦有封包要傳送，則會根據接收端的行程表來計算出交會的頻道，並且檢查該頻道狀態是否有超過 Nav_Threshold，若是超過則會依照接收端的行程表依序去尋找一個低於 Nav_Threshold 的頻道來當傳輸通道，若否則使用該交會的頻道；在達到傳送 slot 之前，節點會依照原本的行程表來掃描頻道來紀錄各個頻道的狀態和聆聽是否有從節收端傳送出來的封包，一旦發現接收端存在就開始對傳送端發送控制封包進行傳輸，每個 slot 結束後，節點會更新頻道狀態表並且尋找一個更靠近現在 slot 且頻道狀態低於 Nav_Threshold 的頻道來當傳輸通道，直到到達選定的 slot，節點會切換到選定的頻道上（因為根據接收端的行程表依序找出，不一定會跟傳送端的行程表符合）並開始傳送控制封包。

4.5 Multi-channel Extension

當 orthogonal channel 數量增加時，各個節點之間的聯繫則會降低並且頻道狀態的更新時間也會拉長；MCS 也支援要多張收發器的情形，主要目的在於縮短兩節點會面所需要等待的時間，假設一個節點擁有 i 張網路卡，分別稱作 NetIF 1、NetIF 2、NetIF i ；每個節點會針對每個收發器給予一個行程表，但是單單如此增加網路卡數也不一定能加強網路的連結性，節點 A 擁有兩個 random sequence vector $(2, 1) = (2\ 3\ 4\ 5\ 6\ 7\ 1)$ 和 $(4, 3) = (4\ 0\ 3\ 6\ 2\ 5\ 1)$ ，節點 B 擁有 $(0, 6) = (0\ 6\ 5\ 4\ 3\ 2\ 1)$ 和 $(5, 4) = (5\ 2\ 6\ 3\ 0\ 4\ 1)$ （以 $p=7$ 來舉例），從節點 A 和節點 B 中各任意挑選一個收發器所得到的 overlapping slot 皆是一樣的，這樣子增加收發器數目並不能帶來任何好處；我們針對同一節點的 random sequence 找出挑選的準則，能保持固定時間內就會出現一個 overlapping 的性質。

首先要避免所有的 overlapping 同時發生，我們挑選同一節點內的 random sequence 使用相同的 seed 但是不同的起始頻道，節點 A 的第 i 個收發器使用地 random sequence 則標示成 (χ^i_A, α_A) ，根據 random sequence 的性質三，一個節點內的收發器將不會有交集；任意兩個節點在一個週期內則可以擁有 i^2 個 overlapping，只要兩節點有任一個收發器切換到相同的頻道上則兩節點可以做溝通，這些 random sequence 一樣也是需要修改才能防止網路的斷裂，修改方法則在後面的小節會說明。

每個節點會以 NetIF 1 的起始頻道 (χ^1_A) 為基準，其它的收發器的起時頻道則分別加上 D_2, D_3, \dots, D_i 倍的 seed 值， $\chi^2_A = \chi^1_A \oplus \alpha_A \odot D_2$ 、 $\chi^3_A = \chi^1_A \oplus \alpha_A \odot D_3$ 。依照這樣的選擇方式，假設節點間第一個收發器會交會在第 K 項，其它的收發器則交會在 $K \oplus (-D_2)$ 、 $K \oplus (-D_3)$ 、 $K \oplus (-D_4)$ 、..... 等項上。

overlapping slot 的證明如下：

利用性質四：節點 A 和節點 B 擁有 i 個 random sequence， (χ^1_A, α_A) 、 (χ^2_A, α_A) 、.....、 (χ^i_A, α_A) ； (χ^1_B, α_B) 、 (χ^2_B, α_B) 、.....、 (χ^i_B, α_B) 。假設兩節點的 NetIF 1 交會在 slot K_1 、NetIF 2 交會在 slot K_2 、.....、NetIF i 交會在 slot K_i 。若是設定 $\chi^2_A = \chi^1_A \oplus \alpha_A \odot D_2$ 、 $\chi^3_A = \chi^1_A \oplus \alpha_A \odot D_3$ $\chi^i_A = \chi^1_A \oplus \alpha_A \odot D_i$ ； $\chi^2_B = \chi^1_B \oplus \alpha_B \odot D_2$ 、 $\chi^3_B = \chi^1_B \oplus \alpha_B \odot D_3$ $\chi^i_B = \chi^1_B \oplus \alpha_B \odot D_i$ 。則 $K_2 = K_1 \oplus (-D_2)$ 、 $K_3 = K_1 \oplus (-D_3)$ 、 $K_4 = K_1 \oplus (-D_4)$ 、.....、 $K_i = K_1 \oplus (-D_i)$ 。

Proof :

Node_A with random sequence

$$(\chi_A^1, \alpha_A), (\chi_A^2, \alpha_A), (\chi_A^3, \alpha_A), \dots, (\chi_A^i, \alpha_A)$$

Node_B with random sequence

$$(\chi_B^1, \alpha_B), (\chi_B^2, \alpha_B), (\chi_B^3, \alpha_B), \dots, (\chi_B^i, \alpha_B), \forall \alpha_A \neq \alpha_B$$

netIF1 overlapping slot K_1

$$\chi_A^1 \oplus \alpha_A \odot K_1 = \chi_B^1 \oplus \alpha_B \odot K_1$$

$$K_1 = (\chi_B^1 - \chi_A^1) / (\alpha_A - \alpha_B)$$

netIF2 overlapping slot K_2

$$\chi_A^2 \oplus \alpha_A \odot K_2 = \chi_B^2 \oplus \alpha_B \odot K_2$$

$$\chi_A^1 \oplus \alpha_A \odot D_2 \oplus \alpha_A \odot K_2 = \chi_B^1 \oplus \alpha_B \odot D_2 \oplus \alpha_B \odot K_2$$

$$K_2 = (\chi_B^1 - \chi_A^1) / (\alpha_A - \alpha_B) \oplus (D_2 \odot \alpha_B - D_2 \odot \alpha_A) / (\alpha_A - \alpha_B)$$

$$K_2 = K_1 \oplus (-D_2)$$

similar,

$$K_3 = K_1 \oplus (-D_3)$$

.....

$$K_i = K_1 \oplus (-D_i)$$

如此一來，我們可以藉由設定 D_2 到 D_i 等元素來達到均勻分布 overlapping 的目的；例如在 $p = 11, i = 3$ 的條件下，設定 $(-D_2) = 4, (-D_3) = 7$ 再加上 random sequence 的週期性，所有的交會點會呈現 $K_1, K_1+4, K_1+7, K_1+11, K_1+15, \dots$ 。

接著我們必須要再修改 random sequence (以 (χ_A^1, α_A) 當範例) 來滿足剛剛推導出來的性質以及防止網路斷裂，這時候我們需要 i 個 seed-dependent elements $(\beta_1, \beta_2, \dots, \beta_i)$ ，modified random sequence vector 所包含的元素則有 $p + i$ 項；分別將這些 seed depend elements 放在 vector 的第 1 項，第 $1+D_2+1$ 項，第 $1+D_3+2$ 項.....再將 random sequence element 逐項的放入來填滿其他空缺的項，以 $\{\beta_1, (\chi_A^1, \alpha_A)\}$ 來表示；NetIF 2 使用的 modified random sequence 則是將 $(\beta_1, \beta_2, \dots, \beta_i)$ 作一個元素的循環變成 $(\beta_2, \beta_3, \dots, \beta_1)$ 和 (χ_A^2, α_A) 來組合而成 $\{\beta_2, (\chi_A^2, \alpha_A)\}$ ；NetIF 3 則是用 $(\beta_3, \beta_4, \dots, \beta_2)$ 以及 (χ_A^3, α_A) ，以此類推。我們挑選每個 seed-dependent element 有不同的產生方式，這樣在 seed-dependent slot 時才不會有同一節點的收發器切換到相同的頻道上，擁有相同 seed 的節點則可以用 seed-dependent slot 來做溝通。圖(4-6)為一個 multi-interface MCS 的範例； $p = 13, i = 3, D_2 = 4, D_3 = 8, \beta_1 = \alpha, \beta_2 = 2 \odot \alpha, \beta_3 = 3 \odot \alpha$ ，節點 A

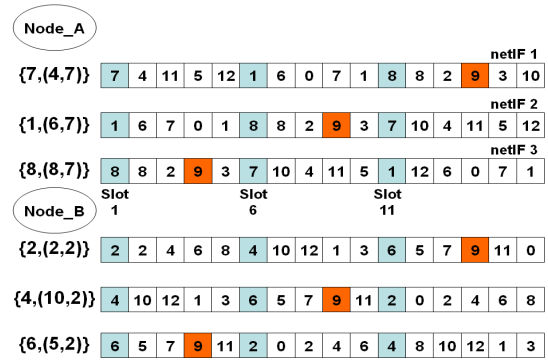


圖 4-6 Multi-interface extension

配有的 modified random sequence 為: $\{7, (4, 7)\}, \{1, (6, 7)\}, \{8, (8, 7)\}$ ；節點 B 的為 $\{2, (2, 2)\}, \{4, (10, 2)\}, \{6, (5, 2)\}$ 。 K_1 出現在 slot 14，往前算四格 (不包含 slot-dependent slot) 則是 K_2 產生的 slot (slot9)，由 K_1 往前算八格則是 K_3 ，如此一來任意連續的六個 slot (包含 seed-dependent slot) 內，兩節點至少會有一個 overlapping slot。

5.Simulation

我們使用網路模擬平台 NS-2 extension 來模擬 MCS 的運作，以及模擬 HMCP 和 SSCH 兩個方法當做比較。HMCP 每個節點配有兩張收發器，一張為 fixed，另外一張為 switchable，SSCH 每個節點則挑有四組的 (channel, seed) 配對以及一個 parity slot，每個 slot size 10 ms。可使用的頻道有三個 ($p = 3$)，每個通道的傳輸速率為 Data Rate 11 Mbps、Basic Rate 1 Mbps，其餘的參數則是使用 ns2 初始值。MCS 使用的參數為 slot size 30 ms，Tx_High 25 packets，max_slot 6 slots，並且使用單一收發器。

5.1 Non-disjoint Single-hop Transmission

在這次的模擬中，我們設立了 24 個節點，分別標記從 0 ~ 23，節點排成一個 6×4 的長方形，每個節點和相鄰的節點間隔 25 公尺，節點的干擾半徑為 500 公尺，也就是說長方形的對角線兩個頂點的傳輸會互相干擾。我們設定了 12 個 UDP 的流量，偶數節點會往正右方的單數節點作傳送如圖(5-1)所示。Flow 1 由節點 0 往節點 1 傳送、Flow 2 由節點 2 往節點 3 傳送、.....。每個 flow 的 sending rate 為 800 Kbps，封包大小為 500 bytes。

圖(5-2)為模擬跑出來的結果圖，802.11 DCF 從 1 到 12 個 flow 皆傳輸在一個頻道上，因此在流量數越大時，整體網路的效能反而會

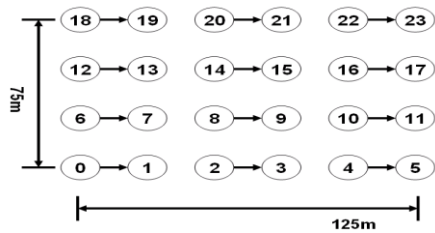


圖 5-1 Scenario 1 topology

下降，而在流量數為 3 的地方，可以看到有個很明顯的轉折，這個轉折則是單一頻道的飽和區；在低流量時，HMCP 頻道的增加的速度較快，因為每個頻道尚未到飽和區，約到 6 條流量之後，HMCP 的增加速率則開始變慢，甚至在流量 9 稍微低於 SSCH，只是最後所有頻道皆達到飽和時，SSCH 每個 slot 中有 3 ms 的時間不能傳送，而 HMCP 則是因為每個頻道上傳輸的流量不均勻，使得整體效能略低於 MCS，由 MCS 的曲線可以看到這兩個問題有被改善。

5.2 Impact of Unbalanced Channel Load

使用和 5.1 節相同的網路拓撲，我們變更每個 UDP 的流量為 1200 Kbps，如此一來每個頻道僅需兩個流量即會進入飽和區；在圖(5-3)中可以看到 MCS 整體網路吞吐量的上升速度比 SSCH 和 HMCP 快速，因為在這樣的流量下單一頻道容易進入飽和狀態，HMCP 隨機的分配接收通道造成的頻道分佈不均勻的缺點會讓整體的效能下降的更嚴重；SSCH 雖然使用多組的 random sequence 來分配頻道，但是一個 slot 內各個頻道的使用情況不一定是均勻的，並且在考慮 3 ms switching latency 的情況下，SSCH 最後的吞吐量低於 MCS 和 HMCP。

5.3 Impact of Switching Latency

我們將使用一個鏈型的網路拓撲，這個鏈型有六個節點彼此相隔 250 公尺，且節點的傳輸距離為 250 公尺干擾半徑為 500 公尺，可使用頻道數為 3；我們放置一個 800 Kbps 的 UDP 流量從鏈型網路的最左端節點送往最右端的節點，結果如圖(5-4)，MCS 在效能上的表現比 SSCH 好，並且 SSCH 在 3 ms switching latency 的吞吐量和 5 ms 時的吞吐量則相差了 270 Kbps，而 MCS 則是在這兩個條件下相差 60 Kbps 的吞吐量；MCS-2 節點內裝備兩張收發器，可以看見的是 MCS-2 比 MCS 有更好的效能，因為 MCS 只配有一張半雙工的收發器，而 MCS-2 可以使用兩張收發器同時接收和傳送，並且收發器會各自挑選不同的頻道運作減少 interference。

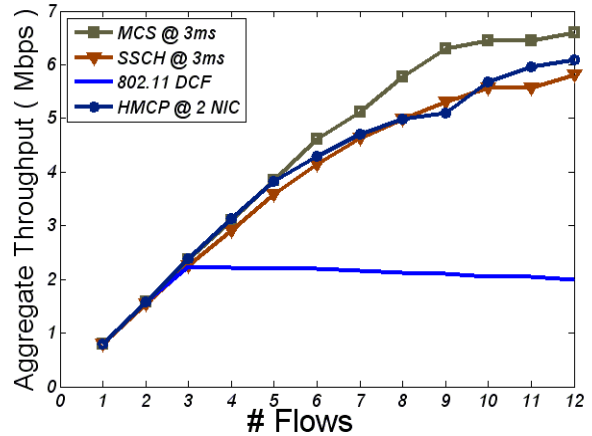


圖 5-2 Performance comparison v.s. number of UDP flows

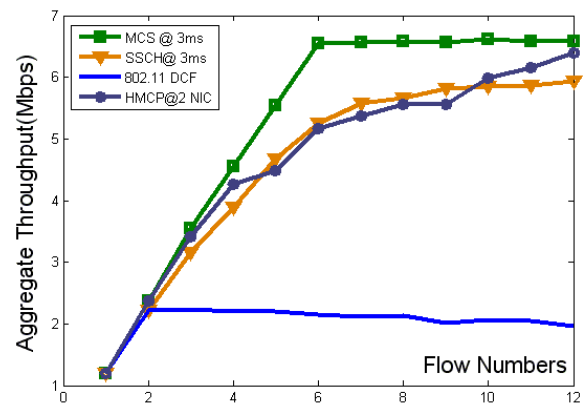


圖 5-3 Impact of unbalanced channel loading

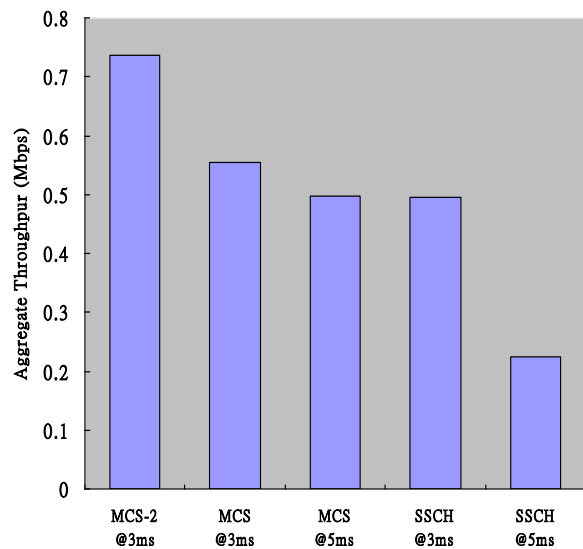


圖 5-4 Impact of switching latency

6. Future Work

我們提出了 MCS 的方式來增加網路的 throughput，但是還有很多衍生的問題可以解決。例如只使用傳送端所獲得的 NAV 訊息來判斷各個頻道的狀態有些誤差，我們計劃使用 busy tone 來當作新的指標，給予每個頻道一個對應的 busy tone 並在接收或是傳送時將 busy tone 打開，各個頻道的使用狀況則可以傳輸給其他節點得知，且每個節點所觀測的環境均不一樣，若是能夠彼此的交換訊息再來判斷各個頻道狀況來選擇傳輸頻道，將能夠更準確的判斷整體網路的情形，挑選出一個條件最好的頻道來進行傳輸；當節點進行頻道負荷偵測時，或許可以依照接收到的封包來做一個統計並且建立一個網路節點間流量的分析方式，來強化接收端行程的估計。現階段 MCS 只能用在質數的頻道數上，我們將探討如何將其推廣到任意正整數的頻道數，並可以保有透過交換少數的資訊量來表達整個頻道排程的內容。在行程表方面，我們也可以添加一些廣播的 slot 專門用來處理廣播封包。在多收發器的時候，我們企圖去新增多的控制封包，使得接收端可以分散不同傳送端到各個收發器上，平均各個傳送端所遇到的競爭。

7. 結論

MCS 使用了 modified random sequence 來當作通道的排程來建立一個節點間溝通的鏈結，並且在多張收發器的時候使用 modified random sequence 的相關性來加速各節點間的聯繫；利用 802.11 DCF 中夾帶的 NAV 資訊來當作頻道負荷的指標，使得傳送端可以選擇一個頻道狀況較好的頻道來減少節點間的競爭，並且安排多個佇列存放相同目的地的封包，使得一次節點交會可以傳送更多的封包降低切換頻道的頻率，透過模擬可以顯現 MCS 減緩了兩個會嚴重影響網路吞吐量的因素：channel load unbalance、switching latency，因此能夠得到好的總吞吐量。

參考文獻

- [1] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya. Multi-channel Mesh Networks: Challenges and Protocols. **IEEE Wireless Communications, 2006.**
- [2] J. So, and N. H. Vaidya. Multi-channel MAC

for Ad Hoc Networks: Handling Multi-channel Hidden Terminals using a Single Transceiver. **In Proc. ACM MobiHoc, 2004.**

- [3] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-hoc Wireless Networks. **In Proc. ACM MobiCom, 2004.**

- [4] J. Mo, H. W. So, and J. Walrand. Comparison of Multichannel MAC Protocols. **In proc ACM MSWiM, 2005.**

- [5] S. -L. Wu, C. -Y. Lin, Y. -C. Tseng, and J. -P. Sheu. A New Multi-channel MAC Protocol with on Demand Channel Assignment for Mobile Ad-hoc Networks. **In Proc. ISPAN, December 2000.**

- [6] H. W. So, J. Walrand, and J. Mo. McMAC: A Multi-channel MAC Proposal for Ad Hoc Wireless Networks. **In Proc. IEEE WCNC, 2007.**

- [7] C. Chereddi, P. Kyasanur, and N. H. Vaidya. Net-X: A Multichannel Multi-interface Wireless Mesh Implementation. **ACM MC2R, 2007.**

- [8] P. Kyasanur and N. H. Vaidya. Routing and Link-layer Protocols for Multi-channel Multi-interface Ad-hoc Wireless Networks. **ACM MC2R, 2006.**