

MANET 中以 TDMA 為基礎的頻寬預留機制

趙峰亭

中華大學資訊工程系
m9002017@chu.edu.tw

嚴力行

中華大學資訊工程系
lhyen@chu.edu.tw

摘要

目前在行動式隨意網路中(Mobile Ad Hoc Network)中有相當多的繞徑通訊協定,如 AODV、DSR、DSDV 等等,不過這些繞徑協定大都並未考慮到服務品質保證的問題,只提供一個能讓資料從來源端到目的端的路徑。而本文中,我們提出了在行動式隨意網路中藉由頻寬預留以提供服務品質(Quality-of-Service)保證的機制,在媒介存取控制層中,我們採用了 TDMA 頻道模組。我們的方法是由目的端來選擇頻寬預留所要保留的時槽,藉由優先選擇擁有最少頻寬的 link 中的時槽,以提高頻寬預留的成功率,並由收集鄰近主機的時槽使用狀況來解決 hidden-terminal 和 expose-terminal 的問題。另外針對[5]中所提的方法做改進,以提高其頻寬預留的成功率。實驗結果顯示我們所提出的方法在頻寬預留的成功率上的確高於其他方法。

關鍵詞:行動式隨意網路(Mobile Ad Hoc Network), 服務品質(Quality-of-Service), TDMA, 頻寬預留

一、簡介

行動式 Ad hoc 無線網路 (MANET) 是由一群配備無線設備而可移動的主機所構成的網路。在此網路架構中並無集中式的網路管理設備,如:基地台等,因為每一個行動式主機電波範圍的限制,當一個封包 (packet) 由來源端主機 (source) 要傳送到目的端主機 (destination) 時,必須藉由鄰近的行動式主機負責幫忙轉送。然而在 MANET 網路中由於主機可以隨意移動,導致網路拓樸 (topology) 隨時都可能改變,為了確保資料傳輸的順利,MANET 的繞徑協定 (Routing Protocol) 必須考量到網路拓樸隨時間改變的情形。到目前為止,已有大量的 MANET 繞徑協定被發表[9][4]。這些研究成果大多單純探討找尋來源端到目的端的路徑,及如何維持來源端到目的端的路徑不因網路拓樸的改變而中斷,提供最有效的資料傳輸路徑,然而對於資料傳輸的延遲、所需頻寬等相關服務品質的問題較少做探討。

目前在 MANET 上較常考量的服務品質有傳輸延遲、頻寬、封包遺失率、產出量等,其中以頻寬方面的研究是較多的。在 MANET 上藉由預留頻寬來提供服務品質保證大多是架構在特定的媒介存取控制 (Media Access Control) 頻道模式下,如 multi-antenna、TDMA、CDMA over TDMA 等。multi-antenna 媒介存取控制頻道模式是非常昂

貴且理想化的,每一個主機都具有 multi-antenna,使得一個 link 的傳輸都擁有各自的頻率而不干擾到其相鄰的 link。目前以此頻道模式的研究有[1]。TDMA 媒介存取控制頻道模式是網路上每一個主機共用一個頻道,然後以 TDMA 的機制來運作,目前以此頻道模式的研究有[5]。CDMA-over-TDMA 媒介存取控制頻道模式是網路上每一個主機共用一個頻道然後加上 TDMA 的機制,而且每一台主機都分配一個正交碼,為了避免傳輸的干擾,每個主机的正交碼與其相鄰兩個 hop 之內的主機都不相同。以此頻道模式的研究有[6][7][8]。

在本篇論文中,我們單純採用了 TDMA 的頻道模組,提出一個由目的端主機決定整個路徑頻寬預留的機制,在我們的方法中也將 hidden-terminal 和 expose-terminal 列入考量,此外我們將[5]所提的方法做一點改進,以提高其頻寬預留的成功率,我們將提出的方法和其他方法做模擬比較,結果顯示我們所提出的方法有較高的成功率,此外我們比較每個方法中收集鄰近主機時槽使用狀態所送出的訊息數量。

本文主要的架構如下,第二節描述相關的背景知識和研究;第三節則詳細介紹本篇所提出的頻寬保留機制。第四節以模擬方式分析我們所提出的演算法效能。最後是本文的結論。

二、相關知識與研究

2.1 相關研究

到目前為止,我們可以發現在整條路徑中,時槽的決定方式可以分為兩種:

逐段決定

在尋找由來源端到目的端路徑的過程中,即預留主機與主機之間傳送封包所需要使用的時槽。屬於這一類的有[6][7][5]。在[6][7]中是由目前的主機來決定與前一個主機之間所要使用的時槽,而[5]中則是由目前的主機來決定與下一個主機之間所要使用的時槽。這兩種的差別在於前者作法為主機將本身時槽的使用資訊附加於路徑找尋的封包 (routing discovery packet) 中一同傳送給下一個主機,由下一主機依據其本身時槽的使用狀況和接收到的前一個主機的時槽使用資訊來做一個比對,以決定所要使用的時槽;而後者必須平時就要收集鄰居各主機的時槽使用狀況,才能在收到路徑找尋封包時能迅速地預留與下一個主機做傳輸時

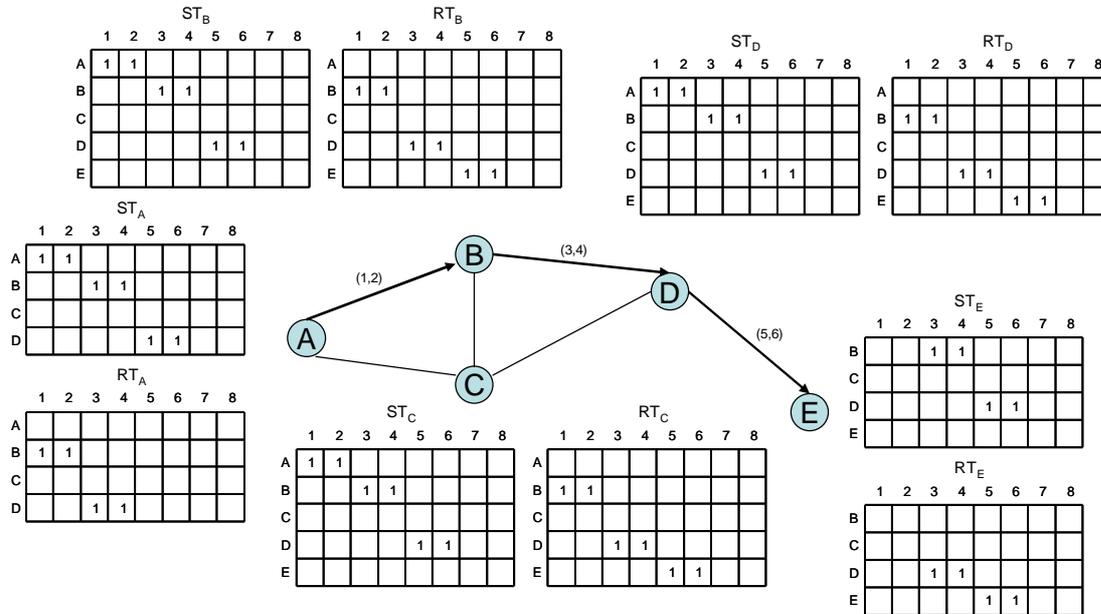


圖 2.1：各主機收集鄰近兩個 hop 主機的時槽使用狀況

所要使用的時槽。

Tseng and Shih[5]中提出了收集鄰近兩個 hop 主機的時槽使用資訊和記錄兩個 hop 之內的網路連接狀態以便選擇用來頻寬預留的時槽。在他們的方法中，收集來的時槽使用資訊被分成兩個部分：Send 和 Receive，分別存放在 Send Table (ST) 和 Receive Table (RT)，table 以矩陣的方式構成，鄰近兩個 hop 之內的主機其時槽若為使用中，則依照 send 或 receive，在 Send Table 或 Receive Table 中的相關欄位標記，而 H 矩陣表示鄰近兩個 hop 內主機的連接狀況，若主機之間是相鄰一個 hop，在矩陣中則以 1 表示，否則為 ∞ 。如圖 2.1，目前 A 主機正在傳送封包到 E 主機，A 主機使用時槽 {1,2} 傳送給 B 主機，B 主機使用時槽 {1,2} 接收從主機 A 傳來的封包，並用時槽 {3,4} 在將封包傳送給 D 主機，D 主機使用時槽 {3,4} 接收封包，再將封包利用時槽 {5,6} 傳給 E 主機，因此 A 主機的 ST 中 [A,1], [A,2], [B,3], [B,4], [D,1], [D,2] 標記為 1，表示正在使用，A 主機記錄兩個 hop 內主機連接狀況的 H 矩陣如圖 2.2 所示，其他以此類推。假若有新的連線要建立，路徑上的主機便會由這些 table 中的資訊找出可以使用的時槽。

$$H_A = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} \infty & 1 & 1 & \infty \\ 1 & \infty & 1 & 1 \\ 1 & 1 & \infty & 1 \\ \infty & 1 & 1 & \infty \end{bmatrix} \end{matrix}$$

圖 2.2：A 主機所記錄兩個 hop 內的網路連接狀態

目的端決定

在此方法中，當一個連線要求被提出時，由來

源端到目的端之間各個主機必須將本身時槽的使用狀況附加在路徑找尋的封包之中，最後封包到達目的端主機，由目的端主機依據所收集到整個路徑上各主機的時槽使用狀況決定主機與主機之間所要使用的時槽。屬於這一類的有[8]。

Lin[8]中使用的頻道模組為 CDMA-over-TDMA，一個 link 的時槽保留只會影響到相鄰 link 的時槽，所以在 Lin[8]的方法中會優先選擇對相鄰 link 沒有影響的時槽，之後再隨機選取那些尚未保留時槽的 link，從中選出對相鄰 link 影響最小的時槽，直到每個 link 都有保留時槽為止，其方法稱之為 bottleneck-link-first。

2.2 hidden-terminal 和 expose-terminal 問題

在 TDMA 的環境下，一個主機無法同時傳送和接收，必須將其分散在不同的時槽，因此我們將面對兩個主要的問題：hidden-terminal 和 expose-terminal。舉例而言，在圖 2.4 的狀況中，假設 D 主機目前正和 E 主機利用時槽 {6} 做傳輸，此時 C 主機想傳送封包給 B 主機，會有兩種狀況產生。若 D 為接收端 (receiver) 的話，C 使用時槽 {6} 傳送封包給 B，將導致在 D 主機這邊產生碰撞的情形，此為 hidden-terminal 的問題；若 D 主機為傳送端 (sender)，C 主機將不會使用時槽 {6} 傳送封包給 B 主機，但 C 主機若使用時槽 {6} 來傳送封包給 B 主機並不會干擾 D 主機傳 E 主機的封包，就實際上而言時槽 {6} 是可以使用的，此為 expose-terminal 的問題。

在 Tseng and Shih[5]中收集鄰近兩個 hop 的主

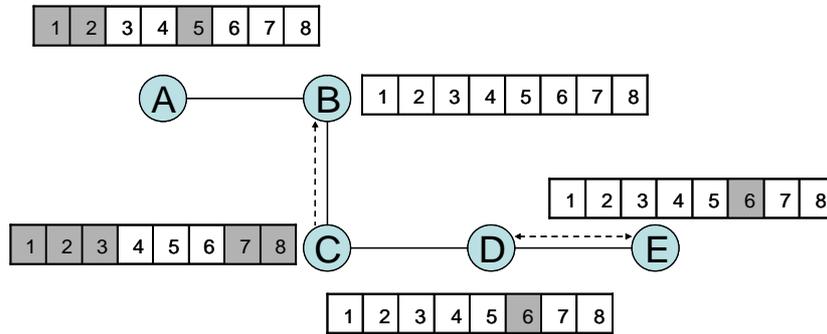


圖 2.4：hidden-terminal and expose-terminal

機的時槽使用資訊和網路連接狀態除了提供時槽的選擇外，還可以用來避免 hidden-terminal 和 expose-terminal 問題的發生。接下來我們以圖 2.1 的例子來說明[5]的方法如何應付 hidden-terminal 和 expose-terminal 問題。我們假設 E 主機有封包要送給 D 主機，此時 D 主機正在使用時槽{3,4} 接收由 B 主機送來的封包，在 E 主機中除了時槽{5,6} 是用來接收從 D 主機傳來的封包，其餘時槽{1,2,3,4,7,8}都是未使用的狀態，而且 E 主機可以藉由查表得知 D 主機未使用的時槽有{1,2,7,8}，另外可以由 H 和 RT 表格中得知 D 主機的鄰居中並未有任何主機在時槽{1,2,7,8}是 Send 狀態，所以 E 主機可以用時槽{1,2,7,8}傳送封包給 D 主機，如此一來 hidden-terminal 的問題便不會發生。至於如何解決 expose-terminal 的問題，假設 C 主機要傳送封包給 A 主機，透過 ST 和 RT 所記錄的資訊，C 主機可以利用時槽{5,6,7,8}傳送封包，但是 D 主機也正在使用時槽{5,6}傳送封包，一般情況下 C 主機無法確認 A 主機是否可以用時槽{5,6}來接收封包，因此 C 主機會放棄使用時槽{5,6}而選擇時槽{7,8}，不過現在可以透過 H 表格中得知 A 主機和 D 主機並不相鄰，所以 D 主機可以使用時槽{5,6}傳送封包給 A 主機。

在本文中，我們也將以收集鄰近主機時槽的使用狀況來解決 hidden-terminal 和 expose-terminal 的問題。

三、我們的方法

在本文中我們提出了兩種以 TDMA 為基礎的頻寬預留機制，以下將針對此兩種方法做一個詳細的說明。

3.1 方法一：Destination Determined Slot Reservation(DDSR)

在此方法中，我們將路徑上所有主機的時槽使用狀況集中到目的端，由目的端來決定哪些時槽用於頻寬預留。

3.1.1 收集整條路徑上主機的時槽使用狀況

在 Tseng and Shih[5]的方法中採用收集鄰近

兩個 hop 內主機的時槽使用資訊，時槽的保留方式為 hop by hop，而且是由上游主機選擇與其下游主機傳輸時所要使用的時槽。之所以要收集到兩個 hop 內主機的時槽使用資訊的原因是因為上游主機必須知道下游主機鄰近主機的時槽使用資訊，避免在選擇時槽時產生 hidden-terminal 和 expose-terminal 的問題，由於收集到鄰近兩個 hop 主機的時槽使用資訊所須送出的訊息數太多，因此我們希望把兩個 hop 降為一個 hop。

Tseng and Shih[5]的方法中，收集的資訊主要在於得知鄰近主機的時槽為使用中或未使用，針對這點我們做一點改變，我們把[5]中上游主機決定時槽的方法改成由下游主機來決定與其上游主機傳輸時要用的時槽，每一個主機只收集鄰近一個 hop 內主機的時槽使用資訊，這些資訊則用 ST 和 RT 來存放，雖然下游主機有上游主機的時槽使用資訊，但並沒有上游主機的鄰近主機的時槽使用資訊，這些資訊只有上游主機才有，因而下游主機在決定要保留時槽時，可能導致 hidden-terminal 和 expose-terminal 的問題，因此我們的重點在於上游主機在收集鄰近主機的時槽使用資訊後，依據這些資訊藉由路徑找尋封包告知下游主機每個時槽所能扮演的角色為何，這裡的扮演角色是指主機在時槽 t 可以為 Sender, Receiver 或兩者都可以或兩者都不行，如此一來下游主機便可以依照所獲得的資訊來決定時槽而且也可以避免 hidden-terminal 和 expose-terminal 問題的發生。而我們的方法是由目的端主機來決定時槽，可以視為由下游主機來決定與其上游主機之間傳輸所要使用的時槽，只是時槽的決定權都交給目的端主機，因此我們將採用上述方法來收集整個路徑上主機的時槽使用資訊。

在我們的方法中也如[5]中使用了 Send Table 和 Receive Table 來存放收集到的鄰近主機的時槽資訊，如圖 3.1(a)中所表示目前網路中部分主機的時槽使用狀況，大寫 "S" 表示目前此主機在這個時槽為 Sender，大寫 "R" 則為 Receiver。我們以 B 主機來說明，在圖中，B 主機一個 hop 的鄰居有 A、C、D，因此 B 主機會收集到 A、C、D 三台主機的時槽使用狀況，然後以 Send Table 和 Receive Table 分別存放鄰近主機的時槽使用狀況，如圖 3.1(b)。

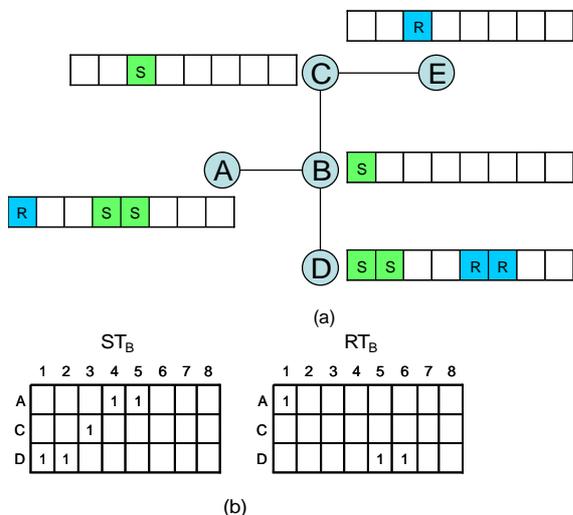


圖 3.1：(a)網路目前各主機的時槽使用狀態，(b)B 中 ST 和 RT 所記錄兩個 hop 主機的實槽使用狀況

當收到路徑找尋封包時，主機必須將其每個時槽所能扮演的角色資訊附加在封包之中，我們將角色資訊分成 6 種，分別為：

- n S：此時槽正在用來傳送封包中，扮演 Sender 的角色。
- n R：此時槽正在用來接收封包中，扮演 Receiver 的角色。
- n s：此時槽未使用，可以用來傳送封包，可以扮演 Sender 的角色。
- n r：此時槽未使用，可以用來接收封包，可以扮演 Receiver 的角色。
- n _：此時槽未使用，可以用來傳送或接收封包，可以扮演 Sender 或 Receiver 的角色。
- n X：此時槽雖然是未使用，但無法用來做任何事。

我們以圖 3.1(a)的 B 主機為例子，B 主機的時槽 1 為“S”表示目前 B 主機正在使用此時槽且扮演的角色為 Sender；時槽 2 為未使用，但在 ST 中發現 D 主機正利用此時槽在做封包的傳送，而 RT 中並無任何主機用此時槽做接收的動作，因此 B 主機在時槽 2 可以為 Sender；時槽 3,4 也因 C,A 兩主機在傳送封包，因此 B 主機在時槽 3,4 也是可以用來為 Sender；時槽 5 雖然為未使用，但是可以發現在 ST 中記錄著 A 主機在做傳送，RT 中記錄著 D 主機在做接收，B 主機在此時槽為 Sender 或 Receiver 都會干擾 D 主機或被 A 主機干擾，所以 B 主機的時槽 5 是不能使用的；時槽 6 的情況則與時槽 2 相反，因此 B 主機在時槽 6 可以為 Receiver；時槽 7 和 8 則是兩者都行。最後 B 主機所要附加在路徑找尋封包的時槽使用資訊為 {B；S, s, s, s, s, X, r, _ , _ }。

3.1.2 時槽的選擇

對於整條路徑來說，其頻寬是由主機與主機的 link 所串連而成，因此整條路徑的最大頻寬將取決於擁有頻寬最小的 link，如圖 3.2 中整個路徑的最大頻寬為 3。另外在 TDMA 頻道模式中，一台主機的時槽保留會影響到其相鄰兩個 hop 以內主機的時槽，如圖 3.3(a)中，A 主機和 B 主機的時槽 1 為未使用，這時 C 主機 C 和 D 主機之間若有資料要傳輸，不論是 C 傳送給 D 或 D 傳送給 C 都能使用時槽 1 來達成，但若如圖 3.3(b)中主機 B 使用時槽 1 傳送資料給 A 主機，此時 C 主機 C 和 D 主機之間有資料要傳輸，時槽 1 只能夠用於 C 傳送給 D，D 傳送給 C 只能使用其他的時槽。



圖 3.2：路徑 A 到 D 的最大頻寬為 3

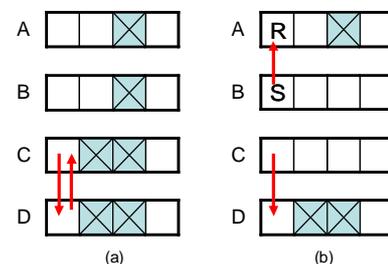


圖 3.3：時槽的選擇影響兩個 hop 主機的時槽

為了避免最小頻寬的 link 因其他 link 保留時槽後使得頻寬變少，而導致整條路徑頻寬預留的失敗，因此在我們方法中優先保留擁有最小頻寬的 link 中的時槽，藉此提高頻寬預留的成功率。

當路徑找尋封包到達目的端主機時，目的端主機可由封包中獲得路徑上所有主機的時槽資訊，此時目的端主機將依照服務品質的頻寬需求來決定預留的時槽。在此階段中，分成以下四個步驟：

- (1) 計算每一個 link 的中所有時槽的狀況，並統計每一條 link 的空時槽數，在此的空時槽必須主機和主機之間可以成為 Send-Receive 的配對，然後對所有 link 將編號相同的空時槽作統計。
- (2) 選出擁有最少共通空時槽的 link，由其共通的空時槽中，依(1)中對編號相同的空時槽的統計結果，從最少的開始配置 Send-Receive 或 Receive-Send。
- (3) 配置完後，有可能影響到鄰近主機的時槽使用狀況，因此對(1)的結果做必要的更新。
- (4) 重覆(2)到(3)的步驟，已經選過的 link 不選擇，直到每個 link 都選過為止。

在步驟(1)中計算每一個 link 的中所有時槽的狀況，其結果有四種情況，我們將這些時槽用不同

的符號來表示，分別為：

- n 上游主機可以為 Sender，其下游主機可以為 Receiver，為 Send-Receive 配對，時槽以“1”表示。
- n 上游主機可以為 Receiver，其下游主機可以為 Sender，為 Receive-Send 配對，時槽以“2”表示。
- n 上述兩者都可以以“0”。
- n 若兩者都不行以“X”表示。

接下來我們以例子來說明。假設 A 主機和 E 主機要做雙向通訊，而 A 到 E 的路徑為 (A,B,C,D,E)，依照我們的方法，路徑上各主機的時槽使用資訊會附加在路徑找尋封包之中，如圖 3.4.1(a)中所顯示，E 主機由路徑找尋封包所獲得整條路徑各主機的時槽使用狀況，在收到這些資訊後，E 主機依上述步驟(1)做進一步的處理，結果如圖 3.4.1(b)。

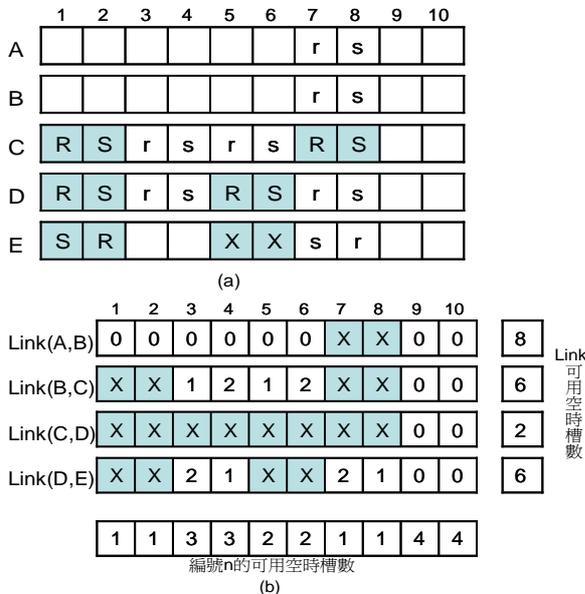


圖 3.4.1：主機 E 依收集的資訊計算 link 的空時槽數

再來先選出擁有最少共通空時槽的 link 做進一步的時槽選擇，由步驟(1)的結果，最少的為 Link(C,D)，其共通空時槽數只有 2，分別為時槽 9 和時槽 10。由於是雙向通訊，一個 link 需要使用 2 個時槽，因此時槽 9 和 10 會被選擇來做頻寬預留。由於時槽 9 和 10 都可以為 Send-Receive 或 Receive-Send 配對，在此我們將 Link(C,D)的時槽 9 配置為 Send-Receive，而時槽 10 為 Receive-Send，C 主機的時槽 9 為 Send，時槽 10 為 Receive，D 主機的時槽 9 為 Receive，時槽 10 為 send。在配置完後，必須對影響到的時槽做更新，影響到的時槽有 B 和 E 主機的時槽 9,10，更新完後如圖 3.4.2 所示。

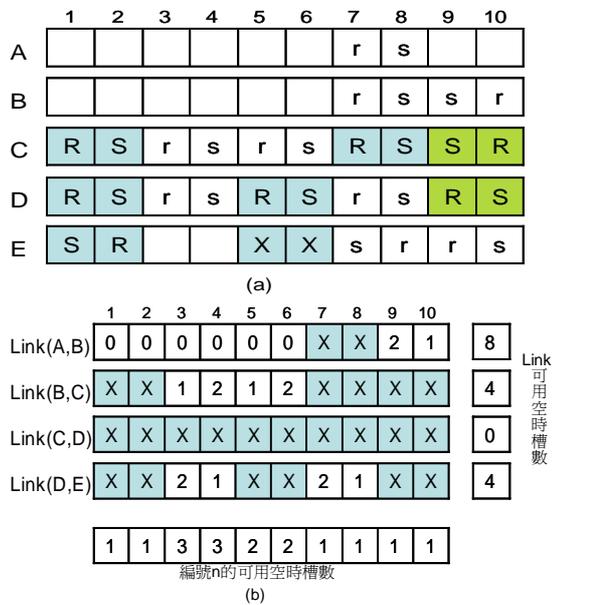


圖 3.4.2：選取 Link(C,D)

接下來繼續從尚未被選擇過的 link 中選出其共通空時槽為最少的 link，從圖 3.4.2(b)中可得知有 Link(B,C)和 Link(D,E)，因為兩個 link 其共通空時槽都為 4 個，因此我們任意選擇其中一個，在此我們選擇到的是 Link(D,E)。Link(D,E)中共通空時槽有 3,4,7,8，依照步驟(2)編號 7,8 的時槽在整條路徑上各只有 1 個共通空時槽，因此我們會選擇這兩個共通空時槽來做為頻寬預留之用，之後再依步驟(3)後，其結果則為圖 3.4.3。

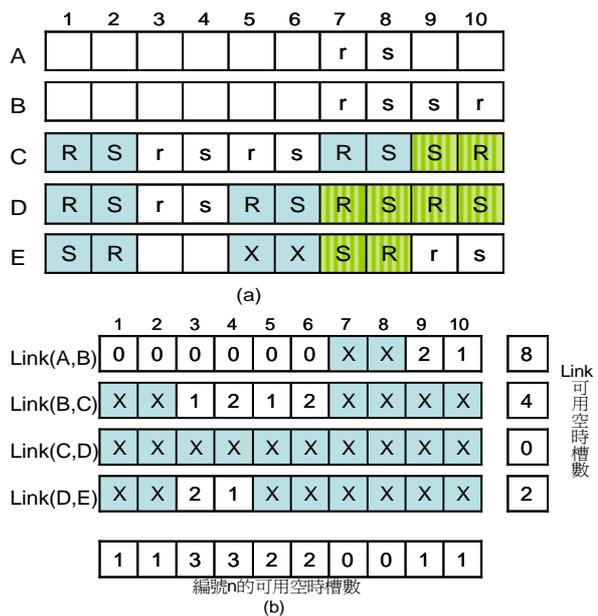


圖 3.4.3：選取 Link(D,E)

接著我們選擇 Link(B,C)，在 Link(B,C)中共通空時槽有 3,4,5,6，依照步驟(2)所選擇的會是時槽

5,6, 如圖 3.4.4。最後只剩 Link(A,B)尚未被選擇，因此從 Link(A,B)中選擇要保留的共通空時槽，所選擇的會是時槽 9,10, 如圖 3.4.5, 至此，整個路徑的頻寬預留完成。

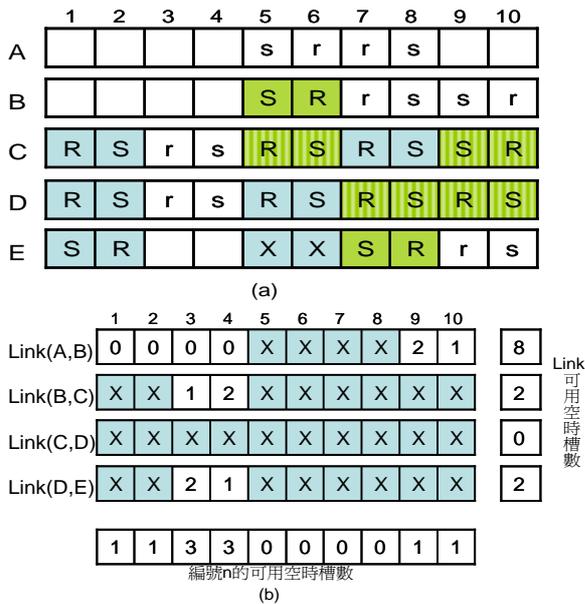


圖 3.4.4：選取 Link(B,C)

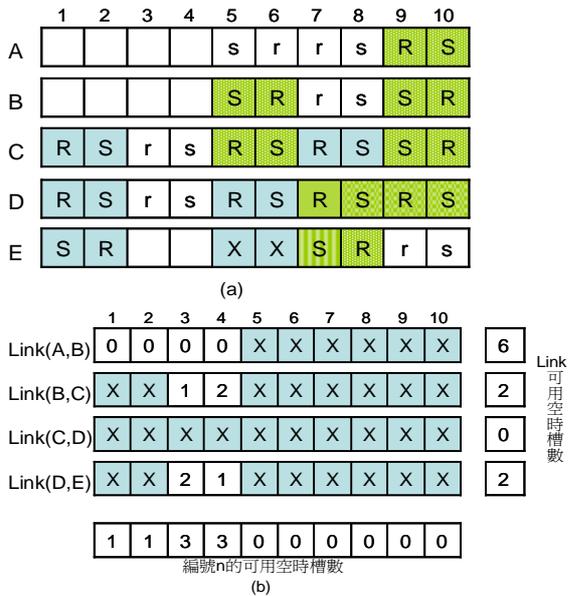


圖 3.4.5：選取 Link(A,B)

在目的端將時槽選擇完後必須告知路徑上所有主機所要使用的時槽，此一訊息我們將其附加在路徑回覆封包 (Routing Replay Packet) 中，使得路徑上的主機能夠從此封包中知道所要使用的時槽。若目的端無法找出滿足頻寬需求所要使用的時槽，目的端要告知來源端連線失敗，而路徑上的主機也不做任何時槽保留的動作。

3.2 方法二：Hop-By-Hop 4 (HBH4)

此方法是針對[5]中所提出的方法做改進，以提高時槽的使用率和頻寬預留的成功率。

3.2.1 問題描述

假設 A 主機要與 E 主機做雙向通訊，其路徑上所經過的主機為 B,C,D, 而 E 主機正在使用時槽 (1) 做接收時槽 (2) 做傳送的動作。由 [5] 的方法 A 主機可得知 B,C 主機的時槽使用狀況，由於此三個主機的時槽皆未使用，所以 A 主機可以選擇任意一個時槽用來傳送封包給 B 主機，一個時槽用來接收 B 主機送來的封包。我們假設 A 主機選擇時槽 (1) 傳送，時槽 (2) 為接收，如圖 3.5(a) 所示，如此一來，C,D 主機之間就無法使用時槽 (1,2) 來傳送封包。假若 A 主機知道 E 主機的時槽使用狀況，則能夠避免此問題，而使用時槽 (1) 接收，時槽 (2) 傳送，C,D 主機彼此之間也就能夠使用時槽 (1,2) 來傳送封包，如圖 3.5(b)。

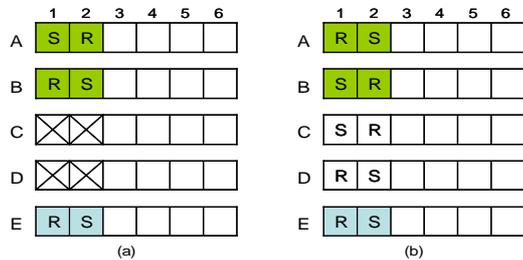


圖 3.5：two hop 和 four hop

3.2.2 演算法

在此方法中，鄰近主機的時槽使用狀況收集方式跟 [5] 是一樣的，只是將原來的兩個 hop 增加至四個 hop，而在時槽的選擇，若有上述狀況則優先選之，主要目的是增加網路可用頻寬。

四、模擬評估

本節我們以模擬實驗方式，評估我們方法的頻寬預留成功率，並與其他方法做比較。其他方法有 Tseng and Shih [5] 的方法和將 Lin [8] 中由 CDMA-over-TDMA 改成純 TDMA 的方式。

我們的模擬環境如下：

- n 模擬區域：1000m x 1000m
- n 主機個數：30 台，隨機分部於模擬區域內
- n 傳輸範圍：300m
- n TDMA frame：16 slots
- n 連線需求產生率 (traffic load)：以 poisson 方式產生，mean rate 為 1/12000 ~ 1/500 per ms，每一個成功的連線將持續 180 秒後中斷
- n 頻寬需求：單向通訊頻寬需要 1,2,4 個 slots，雙向通訊頻寬需求 2,4 個 slots
- n 模擬時間：1000 秒

4.1 頻寬預留成功率

在此實驗中，我們設定頻寬需求分為單向通訊 1, 2, 和 4 個時槽以及雙向通訊 2, 和 4 個時槽。圖 4.1.1~4.1.3 為單向通訊的結果，圖 4.1.4~4.1.5 為雙向通訊的結果，從圖中我們可以發現方法一不論是在單向還是雙向傳輸中其成功率明顯地比 Tseng and Shih[5]和 bottleneck-link-first[8]的方法要來得高，而方法二在單向傳輸需求 2 和 4 個時槽的時後，其成功率大至上跟方法一差不多，而在其他連線需求的時候其成功率明顯的比方法一要來的低。

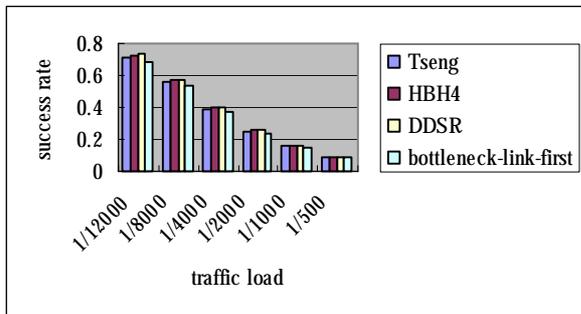


圖 4.1.1：單向通訊時槽數需求 1 個時槽

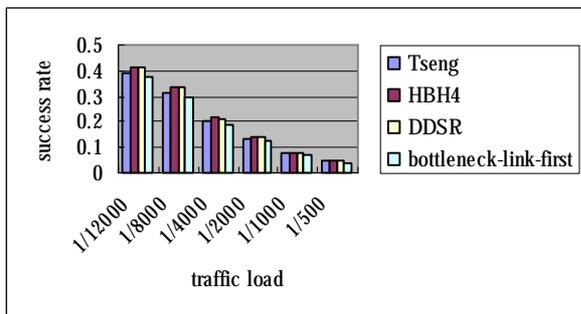


圖 4.1.2：單向通訊時槽數需求 2 個時槽

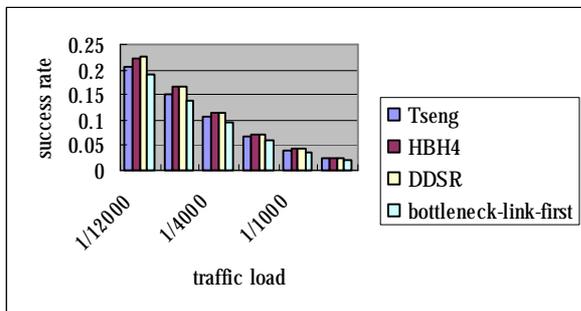


圖 4.1.3：單向通訊時槽數需求 4 個時槽

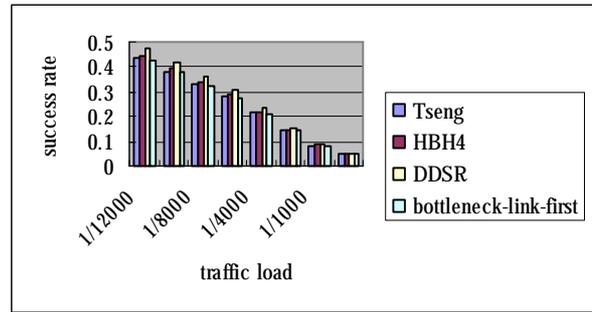


圖 4.1.4：雙向通訊時槽數需求 2 個時槽

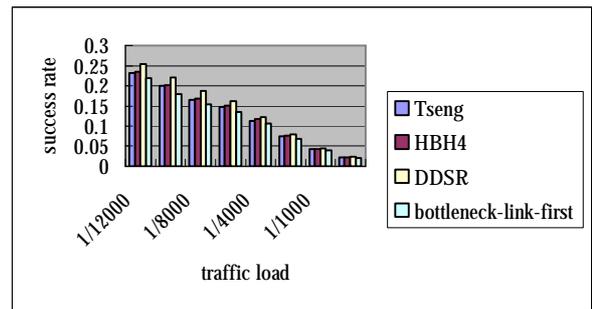


圖 4.1.5：雙向通訊時槽數需求 4 個時槽

4.2 時槽資訊訊息

另外，為了解決 hidden-terminal 和 expose-terminal 的問題，必須收集鄰近主機的時槽使用狀況，對此我們計算這些方法在整個網路所送出的時槽資訊訊息數，結果如圖 4.2.1 所顯示，方法一只需收集鄰近一個 hop 主機的時槽使用狀況，所送出的訊息數與網路主機數一樣，與其他方法比較起來，所需送出的訊息數為最少。而方法二則收集鄰近四個 hop 內主機的時槽使用狀況，所以數量是最多的。

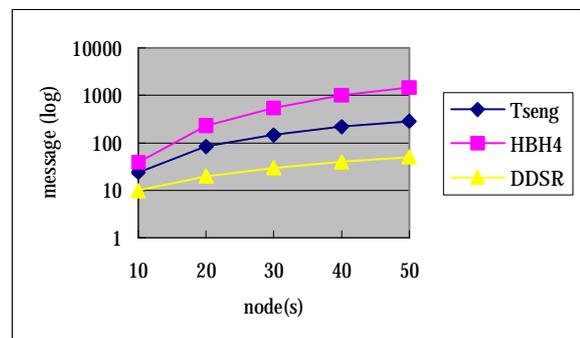


圖 4.2：收集鄰居資訊所需訊息數

4.3 路徑找尋封包的大小

在我們和 Tseng and Shih[5]的方法中，為了提供時槽選擇所需的資訊，每個主機都必需將其時槽資訊附加在路徑找尋封包之中，方法不同所附加進去的資訊大小也不同，因此我們比較方法一和 Tseng and Shih[5]所附加的資料量大小。

Tseng and Shih[5]的方法為 hop by hop，由上游主機選擇與下游主機傳輸要用的時槽，而且在 TDMA 頻道模組中時槽的選擇會影響到鄰近兩個 hop 之內主機的時槽，所以一個主機所附加進去的資訊必須讓兩個 hop 內的主機獲得，而在他們的方法中會將每個主機所選擇的時槽附加在封包之中，因此封包大小會隨著路徑增加。我們的方法一是由目的端來決定要保留時槽，路徑找尋封包內所附加進去的主機時槽資訊是隨著路徑而增加，而每次增加的大小都是固定的。

我們計算路徑上所有路徑找尋封包中附加的時槽資訊量，而且分別以頻寬需求 1 個時槽和 4 個時槽來做比較，Tseng and Shih[5]的方法因為附加上去的資訊為選擇的時槽，所以當頻寬需求越大，附加的資料也相對增加，而我們的方法是附加主機在每個時槽所能扮演的角色資訊，因此所附加的時槽資訊跟頻寬需求並無任何關連，只跟路徑的長短有關。結果如圖 4.3 所示。

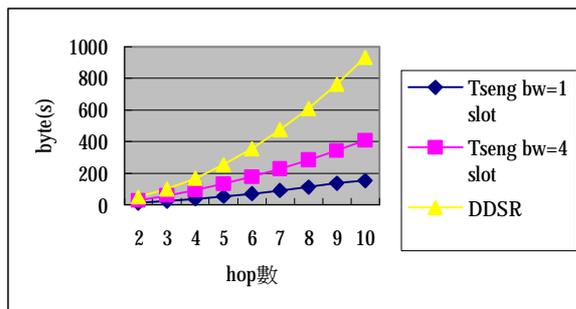


圖 4.3：附加在路徑找尋封包中的時槽資訊大小

五、結論

本文中，我們提出了兩個以 TDMA 為基礎的頻寬預留機制。方法一中我們提出了一個由目的端決定頻寬預留所要使用的時槽機制，透過優先選擇擁有最少頻寬的 link 中的時槽，提高了頻寬預留的成功率，並且收集鄰近一個 hop 主機的時槽使用狀態和得知各個主機在每個時槽所能扮演的角

色，解決了 hidden-terminal 和 expose-terminal 的問題，也降低收集資訊所需發出的訊息數。方法二中我們將 Tseng and Shih[5]中的方法由收集兩個 hop 增加為收集四個 hop 之內主機的時槽使用資訊，雖然增加了收集鄰近主機時槽使用狀況所需發的訊息數，不過在也增加了頻寬預留的成功率。

參考文獻

- [1] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks", *IEEE Journal on Selected Area in Communications*, 17(8):1488-1505, August, 1999.
- [2] S.Chen and K.Nahrstedt, "Distributed Quality-of-service Routing in Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, vol.17, no.8, pp.1488-1505, Augst 1999
- [3] J.J.Garcia-Luna-Aceves and J.Raju, "Distributed Assignment of Codes for Multihop Packet-Radio Networks,"*Proc. of IEEE MILCOM '97*,1997.
- [4] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pp. 81-153, 1996.
- [5] W.H. Liao and Y.C. Tseng and K.P. Shih, "A TDMA-based Bandwidth Reservation Protocol for QoS Routing in a Wireless Mobile Ad Hoc Networks", in *Proc. of IEEE International Conference on Communications*, pp. 3186-3190, 2002
- [6] C.R. Lin, "On-Demand QoS Routing in Multihop Mobile Networks", in *Proc. of IEEE INFOCOM 2001*, pp. 1735-1744, April, 2001.
- [7] C.R. Lin, "QoS Routing in Ad Hoc Wireless networks", *IEEE Journal on Selected Areas in Communication*, 17(8):1426-1438, August, 1999.
- [8] H.C. Lin and P.C. Fung, "Finding Available Bandwidth in Multihop Mobile Wireless Networks", in *Proc. of the IEEE Vehicular Technology Conference*, pp. 15-18, May, 2000.
- [9] E.M. Royer and C.E. Perkins, "Ad-hoc On-Demand Distance Vector Routing", in *Proc. of Second IEEE Workshop on Mobile Computer System and Application*, pp. 1488-1505, August, 1999.