

國立清華大學 電機工程學系

實作專題研究成果摘要

地球同步軌道衛星之波束管理

Beam Management in Geostationary Earth
Orbit Satellite System

專題領域：通訊領域

組別：B482

指導教授：洪樂文 教授

組員姓名：黃鼎翔、李政緯

研究期間：2024 年 1 月 29 日至 2024 年 11 月 29 日止，共 10 個月

摘要

低軌道衛星星座網路因其全球覆蓋、低成本、高持續性等優勢，使衛星通訊在 6G 通訊系統中扮演重要的角色。其中多波束衛星(MBS)結合跳頻(Beam hopping)技術成為新一代衛星通訊系統的趨勢 [1][2]，多波束衛星可以靈活分配衛星資源，以解決地面需求量分布不均勻的問題，而結合跳頻技術則可以實現類似 TDMA 的效果，透過錯開相鄰地區的服務時間來降低互相的干擾，以提升衛星資源的使用效率。因此，本專題研究的目標是設計合適的 Beam pattern，將衛星波束有效分配到不同的地區與時間，來最大化服務的傳輸量。

此外，考慮到系統穩定性問題，本專題為了避免衛星服務量無法滿足用戶需求量而導致的需求量積壓(queue backlog)問題，我們利用 Lyapunov theorem 來設計演算法[3]，能將問題簡化成單一時隙來操作同時找出系統性能與穩定性間的最佳權重，並結合 greedy algorithm [4]來設計 Beam hopping pattern，以實現穩定且高效的衛星通訊服務。

一、背景與動機：

隨著 OneWeb、Starlink 和 Telesat 等公司大規模部署低軌道衛星星座，解決了傳統衛星網路因為距離遠而存在的高延遲、高成本的缺點。覆蓋全球的衛星通訊系統變得能夠實現，覆蓋範圍廣、持續性和建設成本低的優勢使衛星通訊系統在第六代通訊系統中扮演重要的角色。而如何最有效率地將衛星資源分配給地面不均勻的服務需求，成為衛星通訊研究的目標。多波束衛星(MBS)可以同時產生數個波束，對於不同地區的分配不同波束資源能夠更加有效地利用頻寬與功率，而 Beam hopping 技術被廣泛運用在多波束衛星中，允許衛星通訊系統切割時隙，選擇性地照射波束在選定的區域，在每個 time slot 中設計不同的照射區域(BH pattern)，以避免鄰近的波束互相干擾的同時滿足各地區的服務需求，並達到最佳化的服務效率[1]。

由於低軌道衛星需要考慮衛星間的干擾以及衛星內部不同波束的干擾與頻寬分配[1]，還須考慮低軌道衛星移動造成的服務區變換問題[2]，而地球同步軌道衛星的服務區域相較於低軌道衛星不會有改變，不需要考慮接替服務(handover)的問題，也不需考慮因位置改變而變換的仰角與照射功率。因此本專題先選用地球同步軌道衛星來做為研究對象，研究如何能在維持系統穩定性的限制下[3]，實現衛星的最高服務量，實現高效的頻寬利用。

二、研究目的：

本專題以地球同步衛星(GEO)作為研究系統，採用 beam hopping 技術希望能設計演算法來最大化衛星的服務量，同時維持系統穩定性。並觀察衛星功率、地面服務需求量等變因會如何影響衛星系統的運作。

三、研究方法：

本專題採用 beam hopping 的方式，利用波束的空間自由度來設計 BH pattern 以避免相鄰的波束干擾，同時最大化服務量。我們還考慮到需求堆積的問題，當波束在某時隙中沒有服務或是服務的傳輸率不足以應付需求量時，該地區的需求量就會堆積到下一個時隙，如果一個系統在長時間下需求量無限堆積，則稱此系統為不穩定的系統。因此我們使用 Lyapunov theorem 來將問題從長時間的平均交通量轉換成每個時隙的最佳化問題設計演算法，同時能透過調整穩定性與表現的權重來調整我們的演算法，避免系統不穩定的情形產生。

系統模型：

我們架設一顆地球同步衛星(GEO)，並將服務區域切割成 7*7 的地面站，每個地面站標記為 n ，總共有 $N=49$ 個地面站。

Beam hopping：將時間分成 T 各 time slot 並在每個 time slot 各別設計 BH pattern(如 Fig.1 中所示的不同顏色即為不同 time slot 的 BH pattern)，也就是每個 time slot 選定當下最佳的照明方案，選定照明的地面站記作 $\beta=1$ ，未被選定的則記作 $\beta=0$ ，且同時被照明的地面站總量不能超過衛星波束上限 K 。

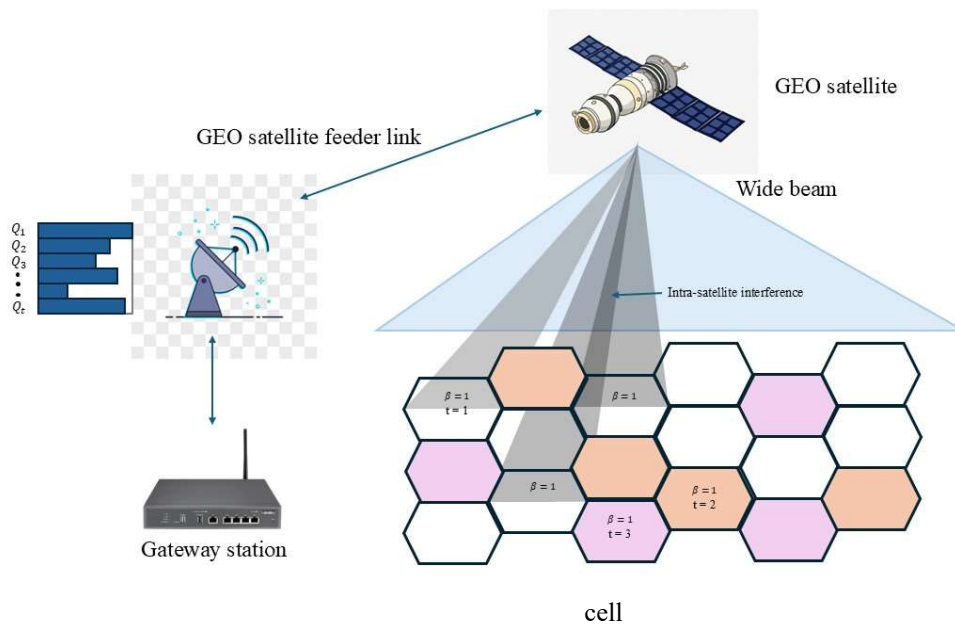


Fig.1 Beam hopping model

通道模型 [1][5]：

G_t ：當波束照在 n' 與地面站 n 呈夾角 θ 時的天線增益

$$G_t(\theta) = \begin{cases} G_m, & \theta = 0^\circ \\ G_m \left| \frac{2J_1(2\pi a \sin \theta / \lambda)}{2\pi a \sin \theta / \lambda} \right|^2, & 0^\circ < |\theta| \leq 90^\circ \end{cases}$$

$h_{n'}^n$ ：當波束照在 n' ，時，地面站 n 與衛星間的通道係數

$$h_{n'}^n = \frac{\sqrt{G_t(\theta_{n'}^n) G_r}}{4\pi d_{n'}^n / \lambda}$$

$SINR_n^t$: 在 time slot t 選定的 BH pattern 下，地面站 n 的 SINR

$$SINR_n^t = \frac{\beta_n(t)P|h_{n'}^n|^2}{\sum_{n' \neq n} \beta_{n'}(t)P|h_{n'}^n|^2 + \sigma^2}$$

R_n : 在 time slot t，地面站 n 的交通量

$$R_n(\beta_n(t)) = B \cdot \log_2(1 + SINR_n^t)$$

Notations	Definitions
a	天線的半徑
J_1	Bessel function of the first kind and first order
Gm	最大天線增益
B	衛星提供的總頻寬
P	衛星提供的總功率
K	衛星最大波束數量
d_n	地面站到天線的距離
θ	兩地面站與衛星的夾角
β	地面站是否被波束照到
σ	噪聲

Table.1 Notation of the parameter

需求量模型：

根據[4]中排隊理論的推導，下一秒的 Queue backlog 會是上一秒 Queue backlog 減去服務掉的 data rate 再加上下一秒新的 data rate。如果服務的 data rate 大於原本的累積，則原本累積的全部被服務完，下一秒的累積會是 $0 + D_n(t+1)$ 。

$$Q_n(t+1) = \max[Q_n(t) - R_n(\beta_n(t)), 0] + D_n(t+1)$$

$A_n(t)$: Arrival data rate 理想的需求 data rate，本專題中設為在每個 timeslot 隨機平均分布在 $[0, A_{\max}]$

$D_n(t)$: Actual demand data rate 系統實際決定的需求 data rate, $D_n(t) < A_n(t)$

$Q_n(t)$: Queue backlog 系統累積的需求量積壓。

問題定義：

參考[3]我們設計演算法目標在能滿足越多需求量越好，即長時間每個地面站的需求量總和，同時需要符合一些限制：

1. 選擇照的地面站總數不能大於衛星的波束數量上限。
2. β 為二元變數，代表地面站選擇亮或不亮。
3. 實際決定的需求量 $D_n(t)$ 在 0 到理想需求量 $A_n(t)$ 之間。
4. Queue backlog 長時間的平均期望值必須有限，代表系統穩定。

$$P_0: \max_{D_n(t), \beta_n(t), \forall n, t} \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N D_n(t)$$

$$C1: \sum_{n=1}^N \beta_n(t) \leq K$$

$$C2: \beta_n(t) \in \{0, 1\}$$

$$C3: 0 \leq D_n(t) \leq A_n(t)$$

$$C4: \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \sum_{n=1}^N E[Q_n(t)]$$

接著我們利用[3][4]提到的 Lyapunov theorem 處理 C4

$$\text{Lyapunov function: } L(Q(t)) = \frac{1}{2} \sum_{n=1}^N Q_n^2(t)$$

$$\text{Lyapunov drift: } \Delta Q(t) = E\{L(Q(t+1)) - L(Q(t)) \mid Q(t)\}$$

Lyapunov theorem 利用 Lyapunov drift 來表示系統的穩定性，當 ΔQ 越小，代表累積的 Q_n 越少，長期下來系統便會趨於穩定。

同時利用 Lyapunov，我們能將問題從長時間平均轉換成單一時隙的最佳化問題，以便我們在單一時隙設計演算法。

因此我們希望在最小化 ΔQ 的同時可以最大化 performance $E \left[\sum_{n=1}^N D_n(t+1) \mid Q(t) \right]$
 於是將問題定成最小化 $\Delta Q(t) - VE \left[\sum_{n=1}^N D_n(t+1) \mid Q(t) \right]$ 並透過調整 V 權重來達成最理想的結果。

問題推導：

我們首先進行 Lyapunov drift 的推導[3][4]，目的是要簡化上界：

$$\begin{aligned}
 L(Q(t+1)) - L(Q(t)) &= \frac{1}{2} \sum_{n=1}^N Q_n^2(t+1) - Q_n^2(t) \\
 &= \frac{1}{2} \sum_{n=1}^N \{ \max[Q_n(t) - R_n(\beta_n(t)), 0 + D_n(t+1)] - Q_n^2(t) \} \\
 &\leq \frac{1}{2} \sum_{n=1}^N \frac{D_n^2(t) + R_n^2(t)}{2} + \sum_{n=1}^N Q_n(t)[D_n(t+1) - R_n(t)]
 \end{aligned}$$

Let $E \{L(Q(t+1)) - L(Q(t)) \mid Q(t)\} = B$

$$\begin{aligned}
 \Delta Q(t) &= E\{L(Q(t+1)) - L(Q(t)) \mid Q(t)\} \\
 &\leq B + E \left[\sum_{n=1}^N Q_n(t) D_n(t+1) \mid Q(t) \right] - E \left[\sum_{n=1}^N Q_n(t) R_n(t) \mid Q(t) \right]
 \end{aligned}$$

接著將 ΔQ 的不等式帶回問題：

$$\begin{aligned}
 \Delta Q(t) - VE \left[\sum_{n=1}^N D_n(t+1) \mid Q(t) \right] \\
 \leq B + E \left[\sum_{n=1}^N (Q_n(t) - V) D_n(t+1) \mid Q(t) \right] - E \left[\sum_{n=1}^N Q_n(t) R_n(t) \mid Q(t) \right]
 \end{aligned}$$

於是我們可以將問題變成兩個子問題，由於 D_n 的需求量是我們設計的，因此可以很直觀地解出 P_1 的答案：

$$\mathbf{P}_1 : \min_{D_n(t), v_n, t} \sum_{n=1}^N D_n(t+1)(Q_n(t) - V), \quad C_3 : 0 \leq D_n(t) \leq A_n(t)$$

The optimal solution of \mathbf{P}_1 :

$$D_n^*(t+1) = \begin{cases} 0, & Q_n(t) \geq V \\ A_n(t+1), & Q_n(t) < V \end{cases}$$

$$\mathbf{P}_2 : \max_{\beta_n(t), \forall n, t} \sum_{n=1}^N Q_n(t) R_n(\beta_n(t))$$

$$C_1 : \sum_{n=1}^N \beta_n(t) < K$$

$$C_2 : \beta_n(t) \in \{0, 1\}$$

再來我們經由[3]的發想設計 greedy algorithm 解決 P_2 。

波束管理:

本專題採用貪婪演算法進行波束管理，整體的演算法架構圖如下:

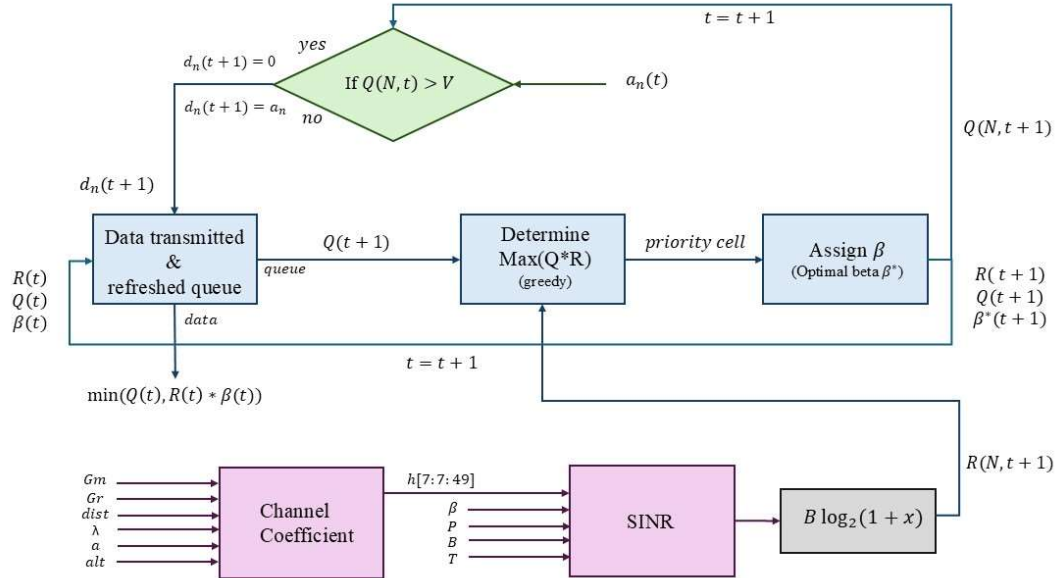


Fig.2 Block Diagram of Algorithmmr

Notations	value	Notations	value
Gm	30(dB)	Alt	35786000(m)
Gr (receiver gain)	0(dB)	P	10^6 (J/s)
Dist.	10000(m)	B	10^6 (Hz)
λ	0.015	T	300(k)
a	0.15(m)	A_max	10^8
V	$3 \cdot 10^7$	K	20

Table.2 Value of the parameter (initial)

為簡化模擬進行的過程，我們忽略了 uplink，downlink 以及與 gateway 間的 propagation delay，而直接分析每個時段中的數據，假設根據上一個 timeslot 得出的 queue length 進行下一個 timeslot input 資料量的運算。為了讓衛星在收發訊息的時候能維持系統的穩定性，系統使用了資料接收量上界藉此令資料 queue 長度不大於我們所設定的上界。

其中綠色的部分，當 queue length 大於 Lyapunov Control Parameter 時，不允許 cell 增加傳輸資料的需求量，因此 data demand 為 0，反之則直接允許 cell 端的

需求量。這樣一來可以確認在 $t \rightarrow \infty$ 時 Q 為一定值，也就是說，確保 ΔQ ，也就是 *queue length* 的變化量在長時間下趨近於 0，符合對於 *stable system* 的定義，證明此設計對於系統穩定性的良好影響。

紫色的部分為計算通道模型與 *data rate* 模型所需的模組，通道模型在忽略大氣與天氣所造成的 *attenuation* 影響後，仍舊必須考慮環境中一些因素的影響。例如，不同位置的 *cell* 與 *satellite* 的距離相異，對於 *satellite* 的仰角也不同，因此造成 *channel coefficient* 上巨大的差異。

藍色的部分則為 *Greedy Algorithm* 的架構[3]。示意圖所沒提及的部分為，每個 *timeslot* 中所計算得出的 β 值皆為 K (beam 的數量(最大 $k_max = 20$))由小到大迭代而成。在 *Determine Max(Q*R)* 模組中。首先在 $K=1$ 時，計算出 QR 值最大的 *cell*，得出在該 *timeslot* 時，該 *cell* 的 β 值為 1，並將 QR summation 設為 QR_max 以利之後的比較。接著推進至 $K=2$ ，如果出現比 QR_max 還要大的 summation 的情況，則將該 *cell* 的 *priority* 設為 1，並在所有 *cell* 都分析完後，將 *priority* 為 1 的 *cell*，其 β 值設為 1，並以此類推，分析完 $K=20$ ，或是在一定的 K 值下，並沒有任何組合可以超越 $K-1$ 得到的 QR_max 後，即完成一個 *timeslot* 的計算。以下為 **Greedy Algorithm** [3]:

Algorithm:

Input: k, rowsize(7), colsize(7), y Output: beta

```
for count = 1:K                                % K 為衛星所能提供最多波束的量
    for i = 1:rowsize
        for j = 1:colsize
            將第 (i,j) 個 cell 的 beta 值設為 1，求 performance
            QR_current = QR_current + currentdataRate*queueLength;
            % 計算當這個 cell beta 值為 1 時的 summation
            if QR_current > QR_max                % 判斷是否為最佳解
                priority = [i,j];                % 是則更新最佳解
                QR_max = QR_current;
            end
        end
    end
```

```

        QR_current = 0;           % 重置並進行下一個
        currentbeta(i,j) = 0;    % cell 的計算
    end
end
if any(priority)
    currentbeta(priority(1),priority(2)) = 1;
end
end
end

```

四、研究結果

1. Demand & Queue Length 對時間之比較

我們使用賦予 β 總共 K ($K = 20$)個 random 值為 1 的情況，比較 Greedy algorithm 優化的部分：

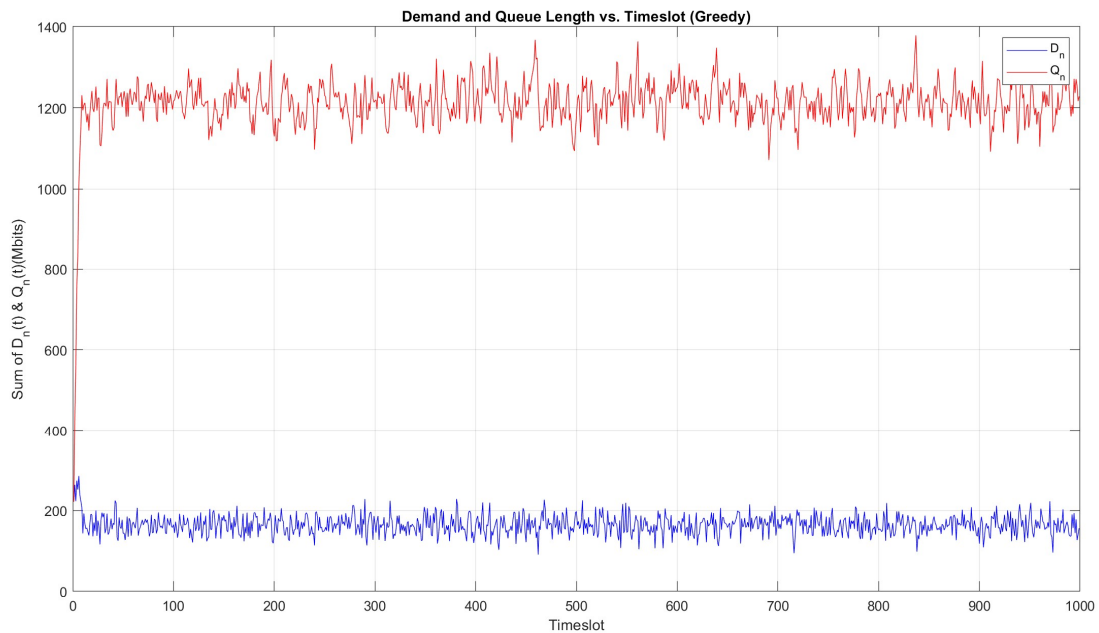


Fig. 3 performance due to changing timeslot (Greedy)

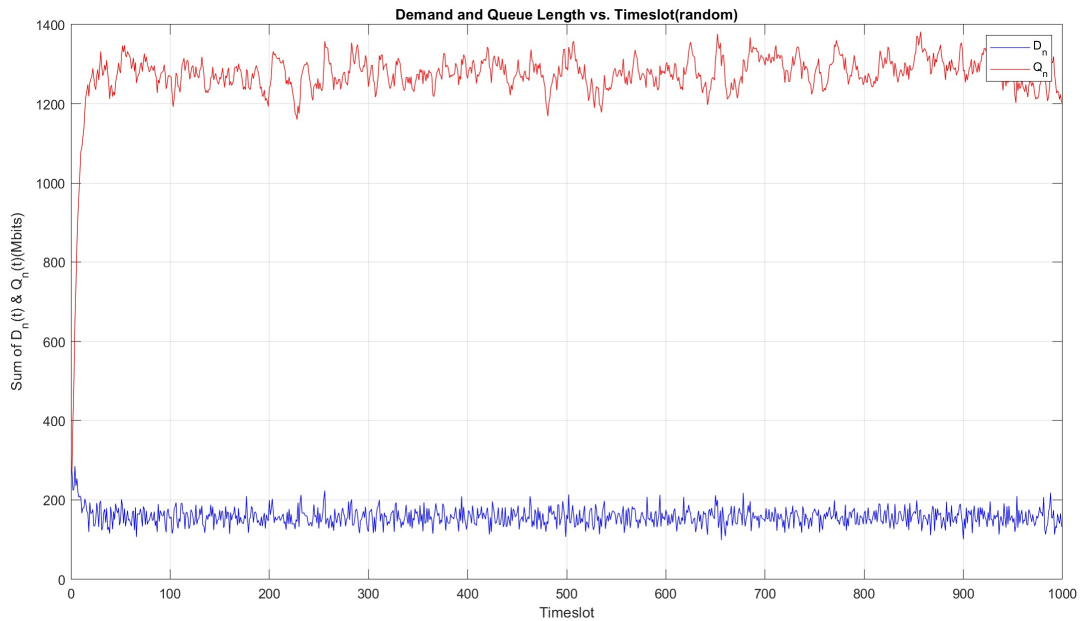


Fig. 5 performance due to changing timeslot (random)

在參數相同的狀況下，使用 Greedy 演算法的 data demand 相比 random beta 來的更高，queue length 也相對較低。而兩張圖也有相似的地方，兩者皆於前幾個 timeslot 時皆位於高 data demand 與低 queue length 的狀態，隨著時間的增長，queue length 越趨近於一定值，說明了系統的穩定性。

2. Demand 對 V(Lyapunov Control Parameter)之比較

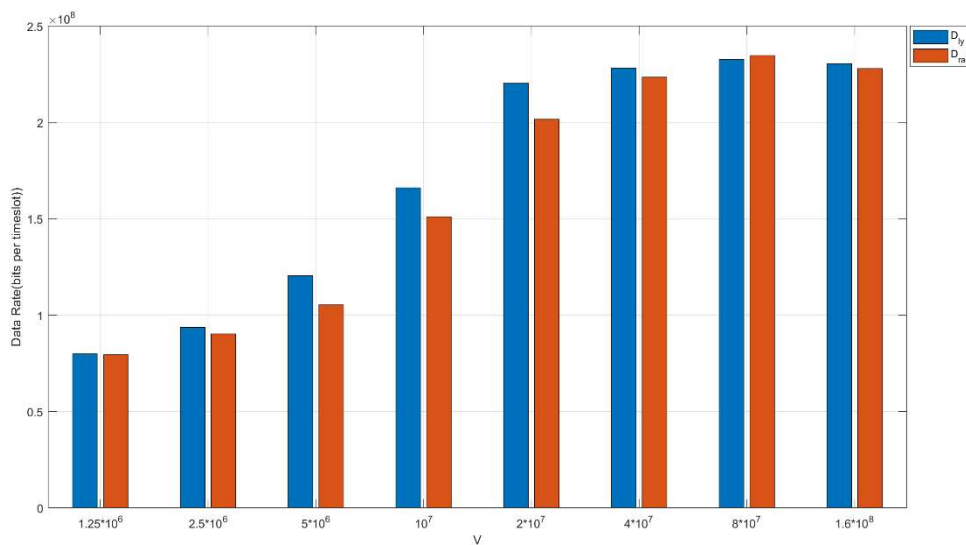


Fig. 6 Demand due to changing Lyapunov Control Parameter

上圖中 D_{ly} & D_{ra} 分別為 Greedy Algorithm 與 random β 在不同 V 的情況下所得出的 demand，所得出的 demand 為在 20 個 timeslots 中平均需求為多少。

在 $V = 1.25 \times 10^6$ 時，兩者的表現十分接近，推測有兩個原因：一是因為 queue length 的長度迅速推升至大於 V ，使得 demand 除了上一個 timeslot 有 beam 的連結的 cell，其餘多數為 0；二是因為 V 的數量級太小，而上圖是以兩倍作為指數增長的，因此兩者的差異友但並不明顯。

隨著 V 的增加，從趨勢上來看，兩者的 demand 都有所增加，且 Greedy 的 average demand 比 random beta 來的更高，代表在有限的 power，有限的 beam 內，使用 Greedy Algorithm 可以在相同的時間內消化更多的 data，從而產生比較好的效率。

在 V 很大時，demand 的成長趨緩，推測有一個原因，是因為本次模擬 20 個 timeslot，在這 20 個 timeslot 中，如果 queue length 持續地穩定成長且都沒有碰到上界，則最後得出的 demand 就會完完全全是我們的 input a_n ，為 random variable 的陣列，因此繼續提升 V 的值並不會影響 demand 的大小。

3. Queue Length 對 V (Lyapunov Control Parameter)之比較

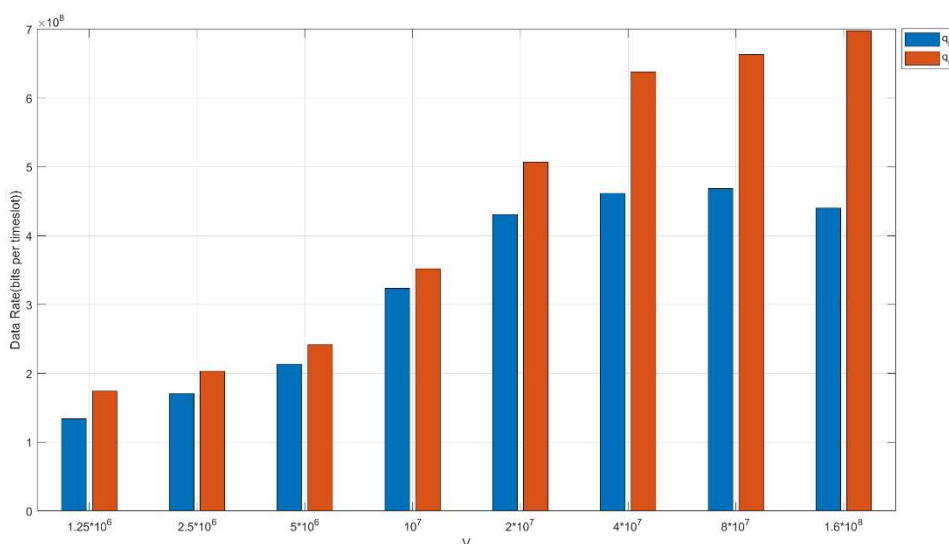


Fig. 7 queue length due to changing Lyapunov Control Parameter

上圖中 q_{ly} & q_{ra} 分別為 Lyapunov Optimization 與 random β 在不同 V 的情況下所

得出的 queue length，所得出的 queue length 為在 20 個 timeslots 中平均需求量為多少。

先從 Fig. 6，demand 對於 V 的圖進行推測，Greedy β 的 queue length 會比 random beta 來的更小，且到 V 大的時候，queue length 的成長會趨緩，因為提升 V 的值並不會影響 demand 的大小，然而 random beta 的 queue length 值卻一直增加。其中有一個重大的原因在於：雖然 random beta 的方式在 V 大的時候並沒辦法讓 queue length 超越 V 來做穩定系統的平衡，但以 random beta 的方式並沒有辦法有效讓 queue length 高的 cell 優先消化排隊的資料量，因此在 queue length 上的表現才會大大輸給 Greedy algorithm。

此外，從 $V = 4 \times 10^7$ 開始增加 V 值基本上不會影響 demand 的量與 queue length 的量，可以發現已經達到一個穩定的狀態，體現了 Lyapunov Optimization 穩定的特點。

四、結論

從結果可以看出，本專題利用貪婪演算法設計出的 beam hopping pattern 效能優於隨機分佈的模型，同時也能維持系統穩定性。我們也透過改變 Lyapunov Control Parameter 找出系統的最佳解，能夠最大量的滿足地面站需求。

以此模型為基礎，可以套用相同演算法到低軌道衛星的系統模型，能夠在此專題之上延伸至更深入的衛星通訊系統研究。

五、參考資料

[1] Z. Lin, Z. Ni, L. Kuang, C. Jiang, and Z. Huang, "Multi-satellite beam hopping based on load balancing and interference avoidance for NGSO satellite communication systems," IEEE Transactions on Communications, vol. 71, no. 1, pp. 282–295, 2023.

[2] J. Zhu, Y. Sun, and M. Peng, "Beam management in low earth orbit satellite networks with random traffic arrival and time-varying topology," IEEE Transactions on Vehicular Technology, pp. 1–16, 2024.

[3] Weichun Chen, Zongjin Yang, working paper.

[4] Michael J. Neely, "Stochastic Network Optimization with Application to

Communication and Queueing systems”, 2010.

[5] Gérard Maral, Michel Bousquet, and Zhili Sun, “Satellite Communications Systems”, Sixth Edition.

六、心得感想

黃鼎翔

從開始專題研究至今也已經有十個月的時間，這次專題學到的東西是扎實的，從衛星通訊的相關書籍開始閱讀，得知衛星通訊並不像地面通訊，在不同的時間中，對於軌道、link 的連接十分的要求，有各種的參數，各式各樣的公式需要去做調整，甚至要確定地面站要在何時連結哪顆衛星都值得討論。這也是後來做專題時碰到的難題之一，至今無法克服 LEO 的難題。在教科書讀一個段落後，選定了 Lyapunov Optimization 的題目做研究，感謝有同組李政緯在我演算法架構時出問題時積極處理並一同克服，感謝專題的學長不辭辛勞每個禮拜都與我們討論專題的架構與方向，並提供了一些十分有建設性的建議，使我不至於成為無頭蒼蠅一般，也感謝專題教授與我們各組進行多次的 meeting 為我們的報告提出一點建議，並指點迷津，使我更了解研究一個領域時所需注意的重點。

李政緯：

這次專題我們從頭開始學起衛星通訊，不僅學到了衛星基本運動定律，也學會了通訊的模式與限制，隨著深入的研究，更是透過論文中了解新一代衛星通訊系統的研究趨勢，不只有此篇的 Beam hopping 技術，還有 cluster beam hopping 及 pre-coding 等技術，以及更複雜的演算法，讓我了解到這個領域的廣度及深度，透過實際設計演算法也讓我在實務上有更加深入的了解，我獲得的不僅是衛星系統與通訊系統上的知識，更是習得問題研究與論文閱讀的方法，這項技能相信在各未來的各種領域上都会有很多幫助。

感謝老師都會在百忙之中抽空來聽學生的專題研究進度，每次都能給出相當精闢的建議，也謝謝助教學長用心地手把手指導，只要有疑問都能馬上獲得學長的幫忙。還有專題夥伴們互相學習、砥礪讓研究的過程中不至於那麼枯燥。

