

國立交通大學

網路工程研究所

碩士論文

降低在 LTE 網路中對於 M2M 服務所產生的 MME 負載

Reducing MME Overload for M2M Services
in LTE Networks

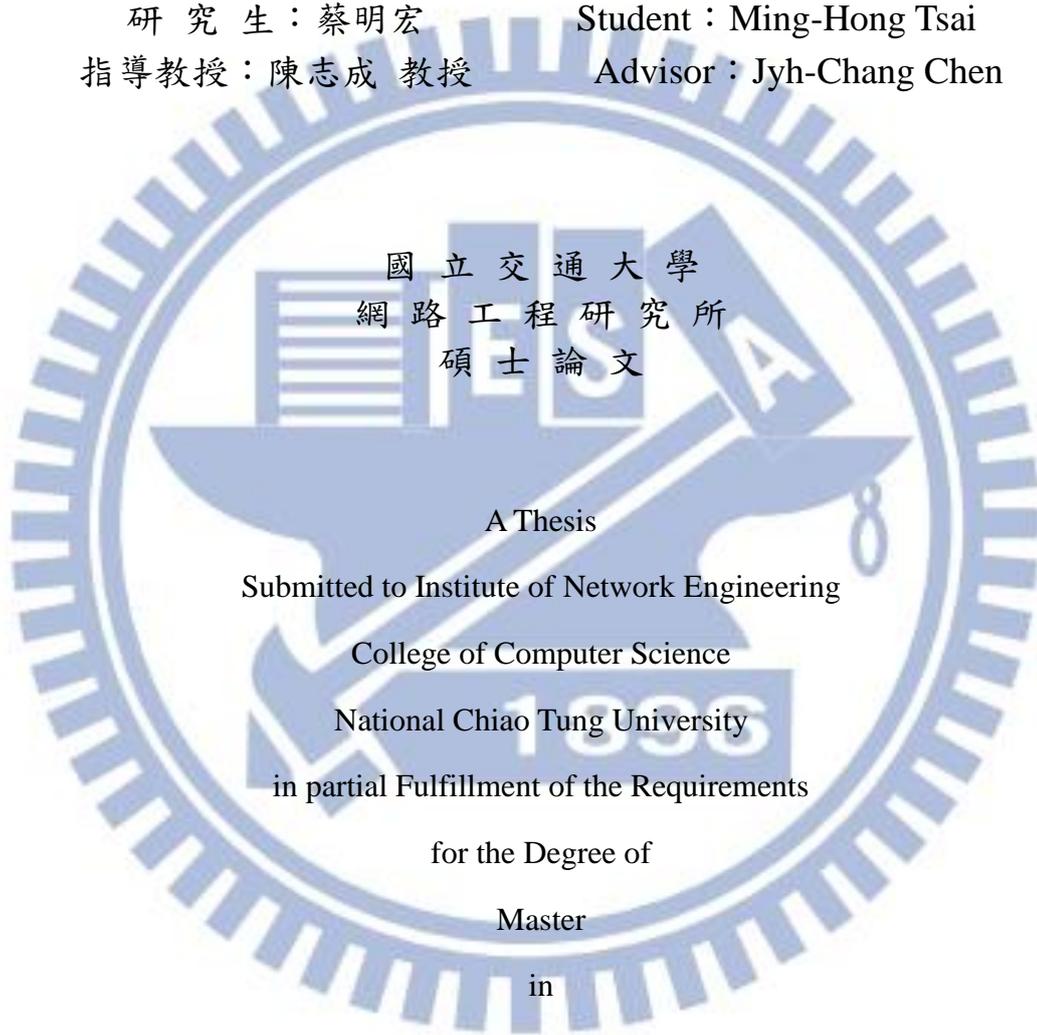
研究生：蔡明宏

指導教授：陳志成 教授

中華民國一〇一年七月

降低在 LTE 網路中對於 M2M 服務所產生的 MME 負載
Reducing MME Overload for M2M Services in LTE Networks

研究生：蔡明宏 Student：Ming-Hong Tsai
指導教授：陳志成 教授 Advisor：Jyh-Chang Chen



國立交通大學
網路工程研究所
碩士論文

A Thesis
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Network Engineering

July 2012

Hsinchu, Taiwan, Republic of China

中華民國一〇一年七月

降低在 LTE 網路中對於 M2M 服務所產生的 MME 負載

學生：蔡明宏

指導教授：陳志成 教授

國立交通大學網路工程所碩士班

摘要

由於通訊技術在這近 20 年來的突飛猛進，從家用的寬頻網路到手機無線網路的普及，使得無線網路逐漸趨於成熟，在 Human-to-Human (H2H) 的應用更加多元化，同時也漸漸飽和，促使另一種新興的應用開始發展，也就是 Machine-to-Machine (M2M)。M2M networks 是由一群 Machine-Type Communication (MTC) devices 相互連接而所形成的網路，彼此之間進行資料的交換、處理、維護或是做一些簡單的修復動作，其過程中不需要人為的介入操作也可以運行正常，而 3rd Generation Partnership Project (3GPP) LTE 也針對這類的應用制定標準。

本篇論文針對標準中所提及的高負載議題來研究，在一段允許時間 (Grant Time Interval) 中，如何讓大量的 MTC devices 連接到 3GPP 網路時降低 MME 的負載，我們提出了三種方式來處理所連接上來的 MTC devices，盡量地降低 MME 負載，分別是 Heuristic-based、Binary-based 和 Fibonacci-based 三種方法。最後在模擬結果中，我們分析了三種方式，實驗結果顯示，我們所提出的方法可以改善 MME 的 overload 問題。

關鍵字：LTE、M2M、overload

Reducing MME overload for M2M services in LTE networks

Student : Ming-Hong Tsai

Advisor : Jyh-Cheng Chen

Institute of Network Engineering
National Chiao-Tung University

ABSTRACT

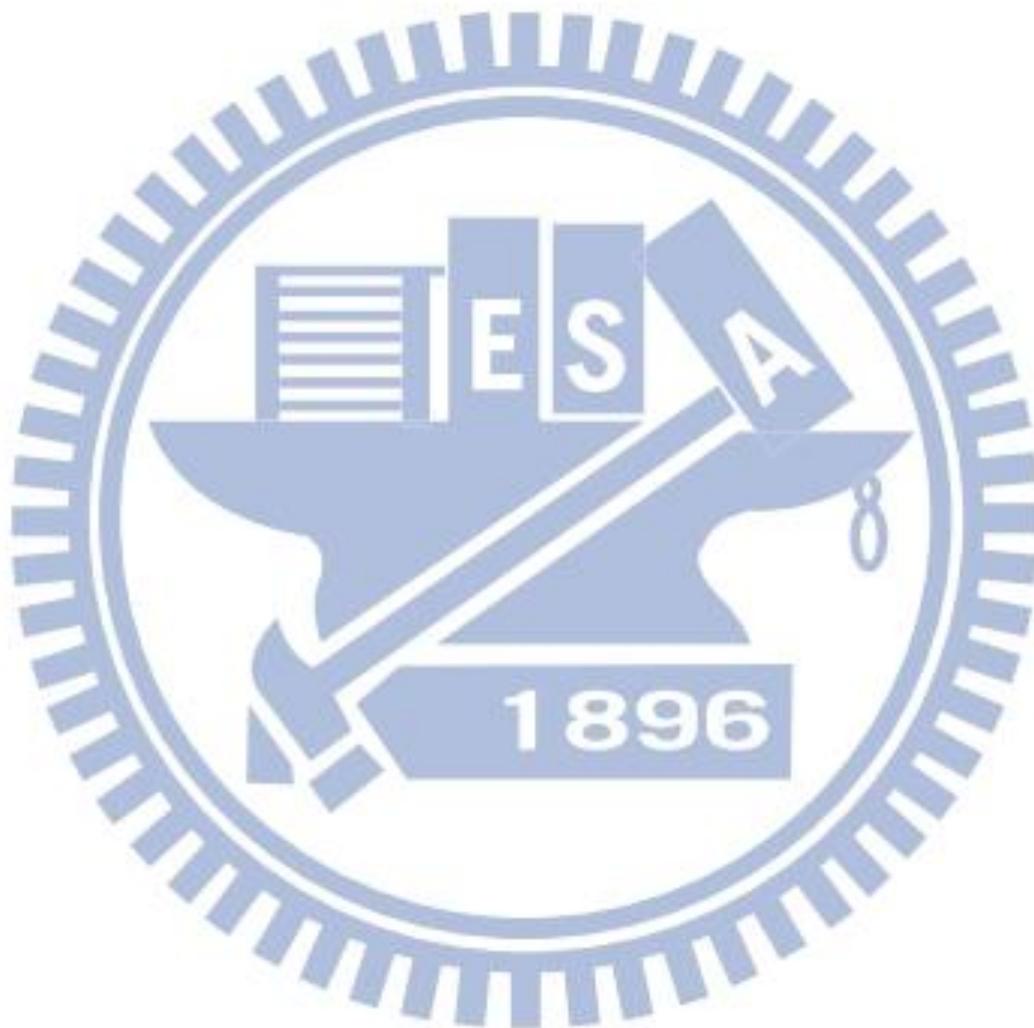
In the past decades, communication technologies have been advanced from broadband networks to the popularity of mobile wireless networks. Wireless networks have become more and more mature. Because applications in Human-to-Human (H2H) are gradually saturated, prompting another emerging applications such as Machine-to-Machine (M2M) communications has begun to develop. M2M networks are networks formed by a group of Machine-Type Communication (MTC) devices which are interconnected with each other. The process does not require human intervention. The operation can also be operated normally. The 3rd Generation Partnership Project (3GPP) LTE is also developing standards for such applications.

In this thesis, we investigate how to make many MTC devices connected to a 3GPP network in the Grant Time Interval (GTI). To reduce the MME load, we propose four techniques to handle the connection of MTC devices, namely Heuristic, Binary exponential backoff, Fibonacci backoff, and random backoff. Finally, we analyze the proposed four techniques by using simulation. The experimental results show that our proposed techniques can reduce the overload of the MME effectively.

Keyword : LTE 、 M2M 、 overload

誌謝

我首先最想感謝的是我的指導教授，陳志成教授。隨著教授的細心指導以及提供非常好的研究環境，發覺到研究的樂趣以及意義。與教授的互相討論，使得我在很多方面受益良多。同時我也要感謝 WIRE 實驗室裡面的所有成員，隨時隨地給予我幫助以及提供我一些非常好的建議。



目 錄

| | 頁次 |
|---|-----|
| 中文摘要 | i |
| 英文摘要 | ii |
| 誌謝 | iii |
| 目錄 | iv |
| 表目錄 | vi |
| 圖目錄 | vii |
| | |
| 一、 簡介 | |
| 1.1 M2M 網路介紹 | 1 |
| 1.2 議題介紹 – Time Control | 2 |
| 1.3 議題介紹 – Signaling Congestion Control | 3 |
| | |
| 二、 議題解法 | |
| 2.1 議題描述 | 4 |
| 2.2 議題解法 | 5 |
| 2.3 提出方法 | 5 |
| (A) Heuristic | 6 |
| (B) Binary exponential backoff | 6 |
| (C) Fibonacci backoff | 7 |
| (D) Bounded Random backoff | 8 |
| | |
| 三、 模擬結果 | |
| 3.1 參數設定 | 9 |
| 3.2 模擬分析 | 10 |
| (A) Average Total Load | 10 |
| (B) Average MTC Load | 11 |
| (C) Average Grant Rate | 12 |
| (D) Request Arrival Rate | 14 |
| (E) Average Grant Delay | 15 |
| (F) Average Back off Count | 16 |

四、 結論

| | |
|----------------|----|
| 4.1 分析總結 | 18 |
| 4.2 未來改進 | 18 |
| 參考文獻 | 19 |



表 目 錄

| | 頁次 |
|-----------------------------------|----|
| 表 1 模擬參數設定 | 8 |
| 表 2 模擬結果 Average grant rate | 13 |

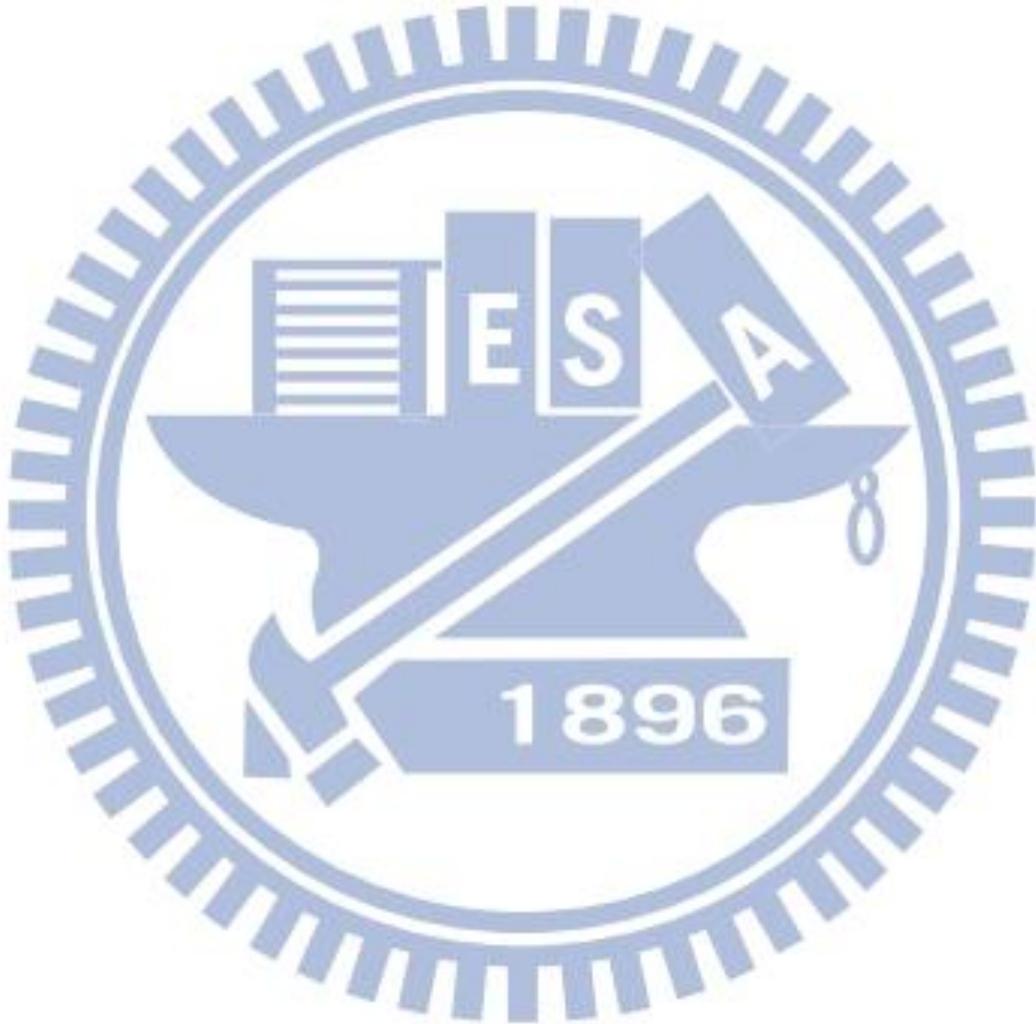
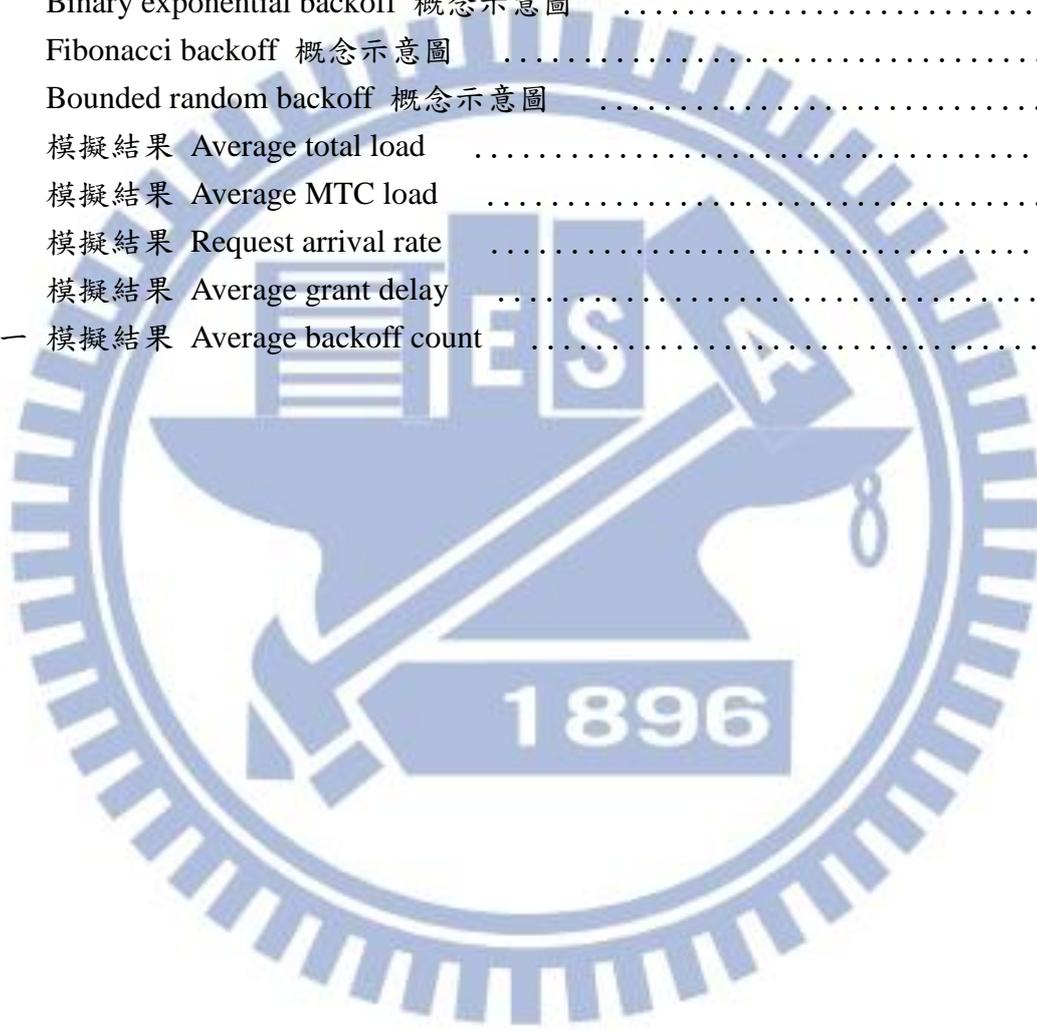


圖 目 錄

| | 頁次 |
|---|----|
| 圖一 3GPP MTC 網路架構 | 2 |
| 圖二 Time Control 時序圖 | 3 |
| 圖三 Heuristic 概念示意圖 | 6 |
| 圖四 Binary exponential backoff 概念示意圖 | 6 |
| 圖五 Fibonacci backoff 概念示意圖 | 7 |
| 圖六 Bounded random backoff 概念示意圖 | 8 |
| 圖七 模擬結果 Average total load | 10 |
| 圖八 模擬結果 Average MTC load | 11 |
| 圖九 模擬結果 Request arrival rate | 14 |
| 圖十 模擬結果 Average grant delay | 16 |
| 圖十一 模擬結果 Average backoff count | 17 |



一、簡介

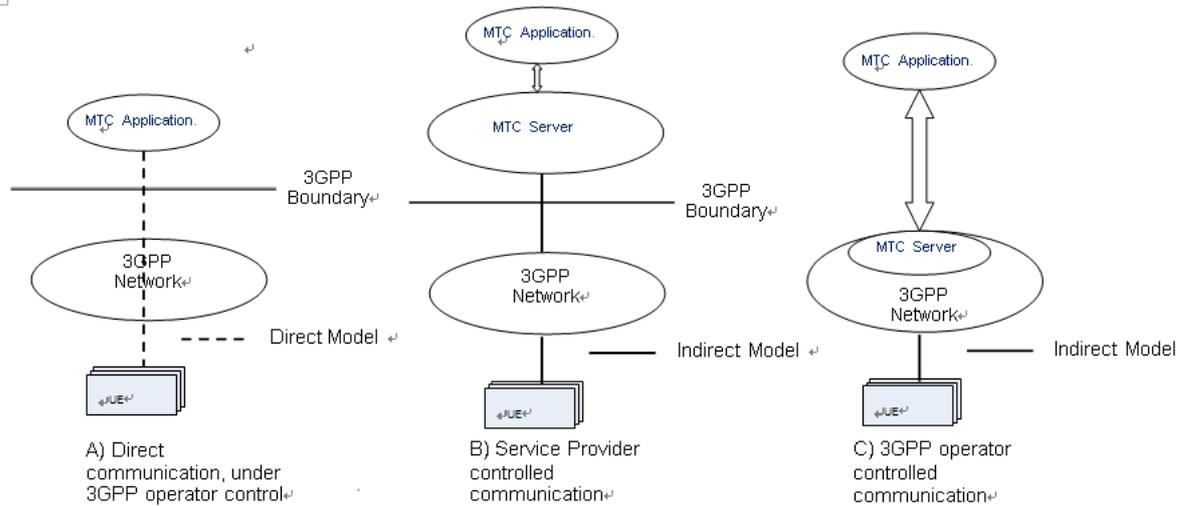
1.1 M2M 網路介紹

在 M2M 網路中，主要是由一群 MTC devices 所連接而成，其應用相當廣泛，可以由同一類型的 MTC devices 連接，也可以是混和類型的 MTC devices 互相連接，例如：智慧電表就可以不需要人工抄表的方式，每戶家庭的電表將連接到 3GPP 網路，可以進行計算度數、計費、軟體更新等功能；一般家電、車輛甚至工業用的 MTC devices 都可以做相對應的應用。但是 M2M 網路與一般網際網路 (Internet) 有某些不一樣的特性，例如：在 M2M 網路中，其封包大小相對於 Internet 較小，而且 MTC devices 久久才傳輸一次，由於這些特性的不同，所以原本 Internet 所使用的協定無法直接套用在 M2M 網路上。

不同的 machine 類型流量在 MTC Application 與 3GPP 網路之間，可預見有幾種不同模型的存在。在一種叫做 Direct Model 之下，MTC Application 可以透過 3GPP 網路直接的與 UE 來進行溝通，如下圖(A)所示。在互補方式下，可預見 Indirect Model 有幾個子模型，在那些子模型中，MTC Application 主要都是使用 3GPP 網路提供的一些額外服務來與 UE 之間做溝通。如下圖(B)所示，MTC Application 使用 MTC Server，而 MTC Server 主要是由第三方供應商來提供一些額外價值與增值服務。

也就是說，MTC Server 是不屬於 3GPP 負責，而 MTC Server 與 MTC Application 之間的介面也非 3GPP 所提供，但在 MTC Server 與 3GPP 網路之間的溝通介面是 3GPP 所制定的。如下圖(C)所示，MTC Application 同樣使用 MTC Server，但是其額外價值與增值服務是由 3GPP 所制訂與提供，MTC Server 與 MTC Application 之間的溝通介面一樣不屬於 3GPP 所制定，不一樣的是 MTC Server 是在 3GPP 網路的範圍內與 MTC Application 溝通。

本篇論文中主要探討的負載議題主要是參考 TR 23.888-1.6.1 [1]，此外為了將模擬器修改至接近標準所定義，還參考了其他文獻[2][3][12][13]，針對標準中所提及的高負載議題來研究，在一段允許時間 (Grant Time Interval) 中，如何讓大量的 MTC devices 連接到 3GPP 網路時降低 MME 的負載，我們提出了三種方式來處理所連接上來的 MTC devices，盡量地降低 MME 負載，分別是 Heuristic-based、Binary-based 和 Fibonacci-based 三種方法。最後在模擬結果中，我們分析了三種方式，實驗結果顯示，我們所提出的方法可以改善 MME 的 overload 問題。



圖一 3GPP MTC 網路架構

1.2 議題介紹 – Time Control

在 LTE 網路的架構中，主要負責處理 device 連接的為 MME 設備，所以這篇論文是以 MME 的角度來處理大量 MTC device 連接的問題。此外，高負載還牽涉到另外兩個議題，Time controlled 與 Signaling Congestion Control。在 Time Controlled 中提到 MTC device 必須要有 Time Controlled 的功能，只能在預先定義的時間區間中 (GTI) 進行資料的傳輸與接收，營運商可以依照某些準則來調整 GTI 的大小，如：每天的流量大小，之後 MTC device 便可以在 GTI 的時間內進行動作。

營運商另外還定義了 Forbidden Time Interval (FTI)，FTI 主要的目的是為了限制 MTC device 在 GTI 外對網路的存取，以免增加負載，另外在 FTI 中還可以進行 MTC Server 的維護，並且不能與 GTI 互相重疊，所以在 FTI 中，MME 必須拒絕所有的 MTC device 請求。通常營運商會將 MTC device 進行分類，並且為每類 MTC device 定義所屬的 GTI，營運商必須將 GTI 通知 MTC device。

當有其他情況時還必須通知 MTC Server 和 MTC User。在 GTI 裡，每個 MTC device 對網路的存取必須要分散在這個區間中，以免造成 signaling 和 data traffic 的 peak load。在眾多的應用中，由於 M2M 的特性，MTC device 傳輸資料的時間並不長，所以營運商在接受 MTC device 的請求後，會定義一個 Communication Window (CW) 來讓 MTC device 來傳輸資料。

為了避免 signaling 和 data traffic 的 CW 產生 overload，MTC device 必須要分散開來，例如隨機開始 CW 的運作。對於營運商而言，越多 MTC device 連接網路越不好，所以當 MTC device 在 CW 結束時，營運商必須將完成工作的 MTC device 進行 detach 的動作，

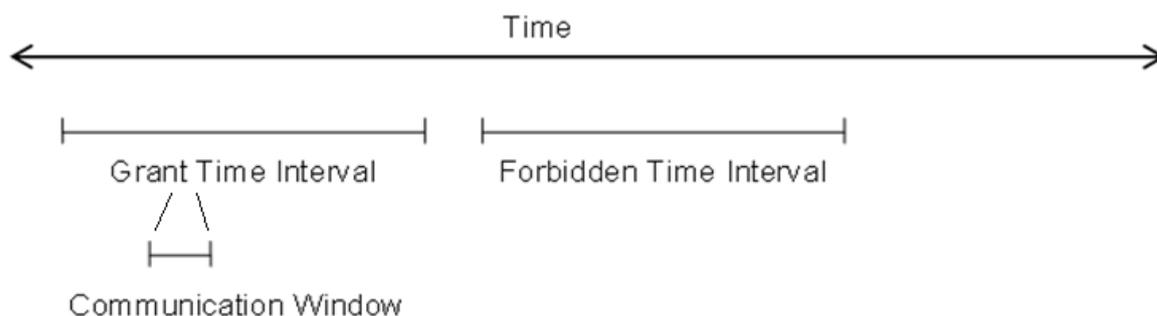
好讓之後更多的 MTC device 可以連接上來。此外，某些應用可能會需要 MTC device 在 GTI 之外的時間連接網路，這時營運商便可以針對此種類型的應用來做不同的收費動作。

1.3 議題介紹 – Signaling Congestion Control

另一個有關的議題為 Signaling Congestion Control，這是營運商對於 signaling 的壅塞和負載所急需面對的問題，有關於 MTC 的 signaling congestion 和 overload 主要有三個因素所導致：

- (一) 在 MTC 應用或 MTC Server 發生故障
營運商在不影響其他 MTC device 的情況下，為了保護整個網路運作而造成的 signaling congestion。
- (二) 由外部事件觸發造成大量 MTC device 一次連接到網路
在整個網路癱瘓前，營運商為了保護網路而造成的 overload 情況，營運商的保護機制將會影響眾多的 MTC 應用。
- (三) 某些 MTC 應用週期性的定期同步到網路
這種類型的應用會導致 peak load 的發生。

為了對抗 signaling congestion，MME 必須要可以拒絕或預防 MTC device 的要求，而且要可以封鎖造成 congestion 的特定 MTC 應用，並且不影響其他的應用。應該要考慮可以拒絕連線要求，並且不能讓被拒絕的要求馬上再重送一次，所以營運商必須要命令 MTC device 要在一個 back off 時間之後才能再次傳送，讓 back off 時間用來改變 MTC device 的要求時間點。接下來在第二章會介紹關於問題的描述與解決方法，在第三章則是模擬與結果分析，最後第四章為結論。



圖二 Time Control 時序圖

二、 議題解法

2.1 議題描述

由於 MTC device 在 GTI 的初期可能會同時進行連接動作，使得營運商可能會遭遇負載的劇烈變動，在標準中提到解決方法主要的概念是讓 SGSN/MME 來限制可連接上網路的 UE 數量，UE 數量中包含 MTC device 與 non-MTC device 的裝置，並且在 GTI 期間中預先定義一個可連接裝置的上限值，使得任一時間所連接上的 UE 數量必定會小於或接近上限值，由於只有限制 MTC device 的存取，所以這個方法建議每個 MTC device 攜帶一個 MTC 指標，用來區分 MTC device 與一般的 Human-to-Human (H2H) 的裝置，同時也建議 MTC device 如果沒有預期要進行任何的通訊，那麼便與網路維持連線的狀態，以減輕營運商的負擔。

若要實作這個解決方法，營運商必須設定一個 UE 連接上限值，若要正常的運作，連接上網路的 UE 數量便不能超過上限值，上限值可以依據營運商的設備能力和運作經驗來決定。在這方法中，主要是在 GTI 的期間選擇一連串的時間點（例如：每隔一秒），並在每個時間點判斷 MTC device 是否可以連上網路，在這些時間點以外的時間若有 MTC device 的連線請求，必須拒絕所有的連線請求，並且告知這些被拒絕的 MTC device 下一次發送請求的時間點為何時。

在每個時間點，SGSN/MME 會檢查 non-MTC device 連接上來的數量，而且與上限值之間的差值，被用來當作 MTC device 可以連接上網路的最大值，若目前已連接的 MTC device 數量小於上限值，那 SGSN/MME 便直接接受新的連線請求直到最大值，否則的話連線請求皆會被拒絕。而當 SGSN/MME 拒絕一個請求時，SGSN/MME 必須要選擇一個之後的時間點，並且告知 MTC device 必須要在那特定的時間點重新傳送一次連線請求。

在這個演算法中，處理的先後順序是先來先處理 (FCFS)，若 MTC device 在 GTI 內無法連線，那 MTC device 便會嘗試在下個 GTI 一開始傳送請求。此外，MTC device 有可能會發生一直被拒絕的情形，導致發生飢餓的狀況。在 GTI 期間內完成工作的 MTC device 應該由營運商或自行切斷連線，以便其他還未完成工作的 MTC device 可以有機會連上網路。

2.2 議題解法

為了解決 MME overload 的問題，關鍵在於 MME 如何幫 MTC device 選擇之後的請求時間點，在我們所提出的方法中，主要的概念是盡可能地減少所額外攜帶的資訊，來達到最大的效益，在前面有提到，營運商會幫每種類型的 MTC device 分類，在分類後會針對每種類型的 MTC device 給予不同的 GTI 時間。

首先將 GTI 時間切割成多個 slot，每個 slot 稱為一個 back off time，back off time 是以 MTC device 完成工作的時間來設定，所以在下一個時間點到來時，理論上前一個時間點所允許的 MTC device 皆完成工作，使得在每個時間點 MME 只需要考慮當下已經連接的 non-MTC device 數量，不須考慮上個時間點還未完成工作的 MTC device 數量。

在這裡總共的 UE 數量上限值為 B ，在每個時間點所連接的 non-MTC device 數量為 N_U ，每當時間點一到 MME 便會計算有多少 MTC device 可以連接上來，可連接上來的 MTC device 數量為 N_M ，而 $N_M = B - N_U$ ，在時間點以外的連線請求，MME 一律拒絕連線，並且告知 MTC device 何時再重送連線請求，而每個時間點請求連線的 MTC device 數量為 N_R ，當 N_R 大於 N_M 時，根據 FCFS，多餘的 MTC device 也是全部拒絕，並且告知下次連線的時間點。

2.3 提出方法

在我們所提出的方法中，每個 MTC device 需要儲存一些額外資訊，也就是 back off 相關的資訊，儘管我們提出的方法計算量非常的小，為了盡可能地降低額外資訊的傳輸與 MME 負載，MTC device 不須傳輸 back off 相關的資訊，並且在 MTC device 上做計算下次要傳輸的時間點，當進入新的 GTI 時間時，每個 MTC device 在第一次發送連線請求時，若第一次就允許，則完成工作後進行 detach 的動作，若是第一次就被 MME 拒絕，MME 必須要在回應封包中加入這次 GTI 的 back off time 時間，MTC device 收到回應封包後記錄下 back off time 的時間，這時 MTC device 就知道在 GTI 中全部可以請求連線的時間點，以便 MTC device 之後的計算，在下面我們列出三種 MTC device 處理 back off time 的方法：

(A) Heuristic

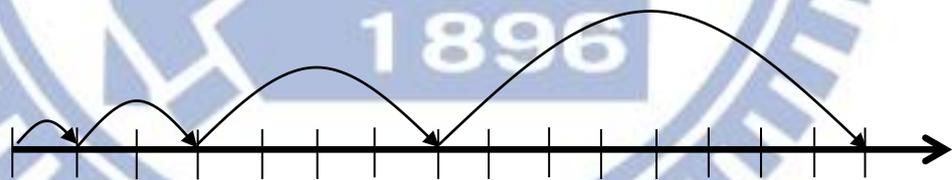
這個方法是最直覺的想法，在每次 MTC device 被拒絕時，當連線請求在非時間點上，MME 就會告知 MTC device 下個離目前最近的時間點，若是連線請求在時間點上，MME 就會告知 MTC device 一個 back off time 之後的時間點，這個直覺的想法，當然會產生很直覺的結果，降低 MME overload 的效果很低，但其主要的優點為不需攜帶任何額外的資訊，如下圖所示。



圖三 Heuristic 概念示意圖

(B) Binary exponential backoff

這個方法的概念是將問題對應到 WiFi 的 back off window [8]，也就是說，每次 MME overload 時拒絕 MTC device 的情況相當於 WiFi 的碰撞，每當等了一段時間之後，若再次被拒絕，則下次的連線請求時間點，為上次等待時間的兩倍加上目前的时间點。在這裡 MTC device 需要額外儲存一項資訊，就是 back off length，back off length 就是以 back off time 為單位，為 MTC device 目前等了多少個 back off time，例如：某 MTC device 的 back off length 為 8，表示這個 MTC device 已經等了 8 個 back off time 的時間。概念如下圖所示。



圖四 Binary exponential backoff 概念示意圖

若是連線請求在時間點之外被拒絕，假設 MTC device 目前的 back off length 為 16，並且將 16 乘上兩倍變成 32，隨後計算離目前最近的時間點與目前的时间差 d_t ， d_t 會小於一個 back off time，在這裡 d_t 也當作一個 back off time，接著計算下個連線請求的時間點，也就是 31 乘上 back off time 加上 d_t 再加上目前時間；時間點如下列式子：

$$TP_{next} = d_t + (BOL_{next} - 1) \times BOT + TP_{now} \quad (1)$$

若是連線請求在時間點之上被拒絕，MTC device 以相同方法計算，差別在於不需要計算 d_t ，也就是下個請求時間點為 back off length 乘上 back off time 加上目前時間點。時間點如下列式子：

$$TP_{next} = BOL_{next} \times BOT + TP_{now} \quad (2)$$

這個方法會產生一個問題，如前面所提到，MTC device 有可能會一直被拒絕，由於 back off 的等待時間是以 exponential 的速度在成長，所以被拒絕多次之後，back off 的等待時間變長，造成完成工作的延遲也會變長。

此外，可能會發生下次可以發送連線請求的時間點已經超出 GTI 的時間之外，以至於 MTC device 必須要等到下個 GTI 的到來。在這裡我們將 back off length 加以限制，除了 binary-based 之外，另一個方法為 bounded binary-based，bounded binary-based 就是在 back off length 超過限制時，將 back off length 重設為 1 開始，除了利用這個方法來縮短完成工作的延遲外，也減少下個連線請求的時間點超出 GTI 的時間次數。

(C) Fibonacci backoff

Fibonacci-based 主要是 Binary-based 的概念延伸，由於 Binary-based 的 back off 是以 exponential 的速度在成長，完成工作的延遲時間有可能會變很長，所以主要概念是提出以 Fibonacci 數列的速度來計算 back off，來減少完成工作的延遲時間。概念如下圖所示



圖五 Fibonacci backoff 概念示意圖

在這裡 MTC device 需要紀錄上次與目前的 back off length，當被 MME 拒絕的情況下，如果在時間點之外，將目前的 back off length 與上次的相加後，計算出所需的 back off length， d_t 一樣當作一個 back off time，則下次的連線請求時間點如下列式子：

$$TP_{next} = d_t + (BOL_{next} - 1) \times BOT + TP_{now} \quad (3)$$

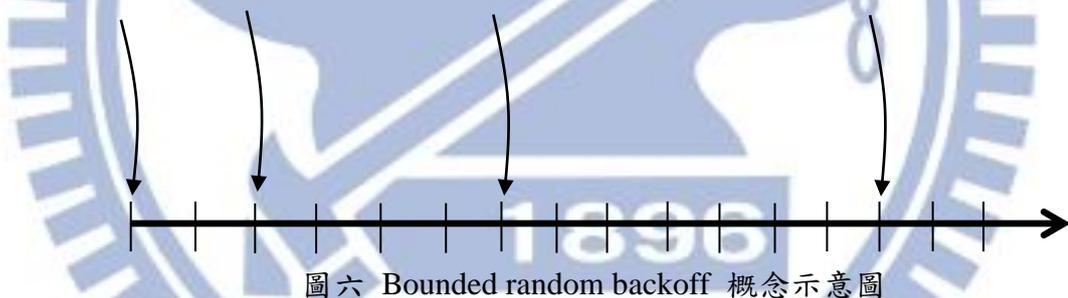
若是在時間點上被拒絕，計算方法一樣，不同在於不需要計算 d_t ，則下個時間點式子如下：

$$TP_{next} = BOL_{next} \times BOT + TP_{now} \quad (4)$$

如同 Binary-based 所提到，若是一直無限的增加 back off length 會出現完成工作延遲過長的問題，所以這裡也是採取同樣的方法，限制 back off length 的長度來減少延遲時間以及降低超過 GTI 時間的次數。

(D) Bounded random backoff

這個方法主要是在一個有限的時間範圍內，隨機一個時間點做連線請求，可預期的是，若限制的時間範圍越長，則負載會越低，延遲也會隨著增加，若是時間範圍越短的話，則負載會越高，延遲會縮短。所以這個方法可以依不同的 MTC 類型需求，來調整相對適合的 Back off 方法。概念如下圖所示。



在這裡並不需要紀錄任何的 back off length，只須設定多少的時間範圍，例如：在第 10 秒時 MTC device 被拒絕，若設定的時間範圍是 32 秒，就是在第 11 秒到第 42 秒之間隨機一個時間點，並且通知 MTC device 在隨機點連上來。下一個時間點式子如下。

$$TP_{next} = \text{Random Time Point} \quad (5)$$

三、 模擬結果

3.1 參數設定

在這裡使用的模擬軟體為網路上個人開發的開放軟體 LTE-Sim [9]，由於 LTE-Sim 在某些部份沒有實作完全，這裡我們參考一些文獻，並且實作和修改了某些底層部分的程式，如 control plane [4] [6] [7][12]、MAC scheduler [5][8][10][11]，並且參考其他文獻 [14][15][16][17] 來彌補其他不足的地方，我們使用了 100 個節點來模擬大量的 MTC device 連線請求，每個節點的發送速率依照下列公式：

$$\lambda = E[N_{GM}] \times T_S \times \frac{1}{L_T} \times load \times \frac{1}{T_S} \times \frac{1}{MTC} = \frac{E[N_{GM}]}{L_T \times MTC} \times load \quad (6)$$

在上列式子中， $E[N_{GM}]$ 代表的是平均接受 MTC device 連線請求個數，我們設定 MME 可以處理 UE 數量的上限為 500，而這裡的 $E[N_{GM}]$ 為 250， T_S 代表的是總共的模擬時間，我們模擬時間是 50000 秒， L_T 代表的是一個 frame 的長度，這裡我們設定為 1 秒，前面 $E[N_{GM}] \times T_S \times \frac{1}{L_T}$ 代表在 50000 秒內總共可以允許的 MTC device 數量，也就是說

MME 在 50000 秒內處理 MTC device 的能力，而 $E[N_{GM}] \times T_S \times \frac{1}{L_T}$ 乘上 *load* 就是我們要模擬的負載情況，例如：MME 在模擬時間內可以處理的 MTC device 的數量為 100000，乘上 *load* 90% 為 90000，所以我們所需要送出的連線請求為 90000，以 MME 的角度來看，90000 除以 T_S 可以得到每秒有多少連線請求，再除以 *MTC* 之後便是每個節點所要發送請求的速率。此外，MME 在每次時間點處理 MTC device 時，會依據優先權來處理，這裡所設定的優先權是以 back off count 為準則，也就是被 back off 次數越多的優先處理，以下為詳細參數表：

| Parameter | value |
|---------------------------|-----------------|
| Upper bound of number UEs | 500 |
| Simulation time | 50000 s |
| Frame time | 1 s |
| Average number of MTC | 250 |
| Average number of non-MTC | 250 |
| Number of node | 100 |
| Maximum packet delay | 10m s |
| priority | Back off length |

表一 模擬參數設定

3.2 模擬結果

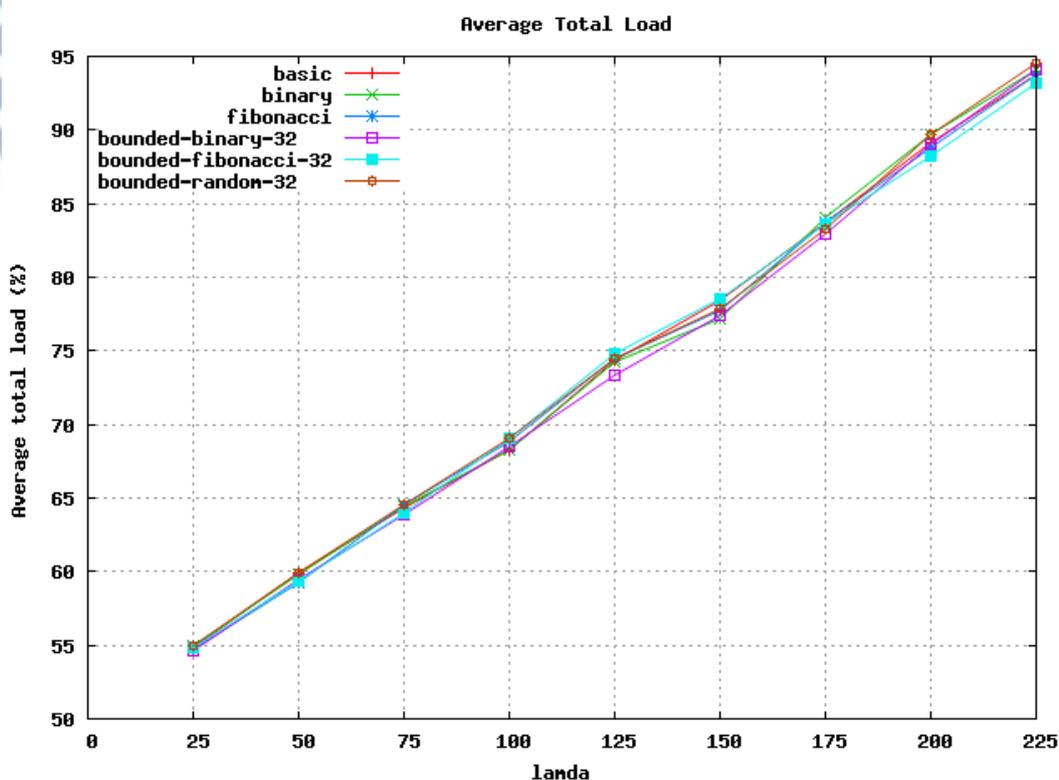
在這個小節中將分析六種不同的衡量標準，並且一一在每個衡量標準中做仔細的分析。

(A) Average total Load

首先是 Average total load，所謂 Average total load 就是在每次 time point 計算總負載，也就是 MTC device 的數量加上 non-MTC device 的數量之後除以上限值，並且將每次 time point 的總負載累加起來，在模擬結束時除以 time point 次數，以便得到最後平均的總負載值，其公式如下。

$$U = \sum_{N_G} \frac{N_{GM} + N_U}{N_M + N_U} \quad (7)$$

在上述的式子中， N_G 表示在模擬時間內總共的 time point 次數， N_U 表示在每次 time point 時，MME 檢查目前有多少的 non-MTC device 在使用， N_M 表示 UE 上限值減掉 N_U 後所能最大允許的 MTC device 數量， N_{GM} 表示目前所允許的數量，其中 $N_{GM} < N_M$ 。下圖中顯示出在每個 λ 時，各種方法所測量到的 Average total load。



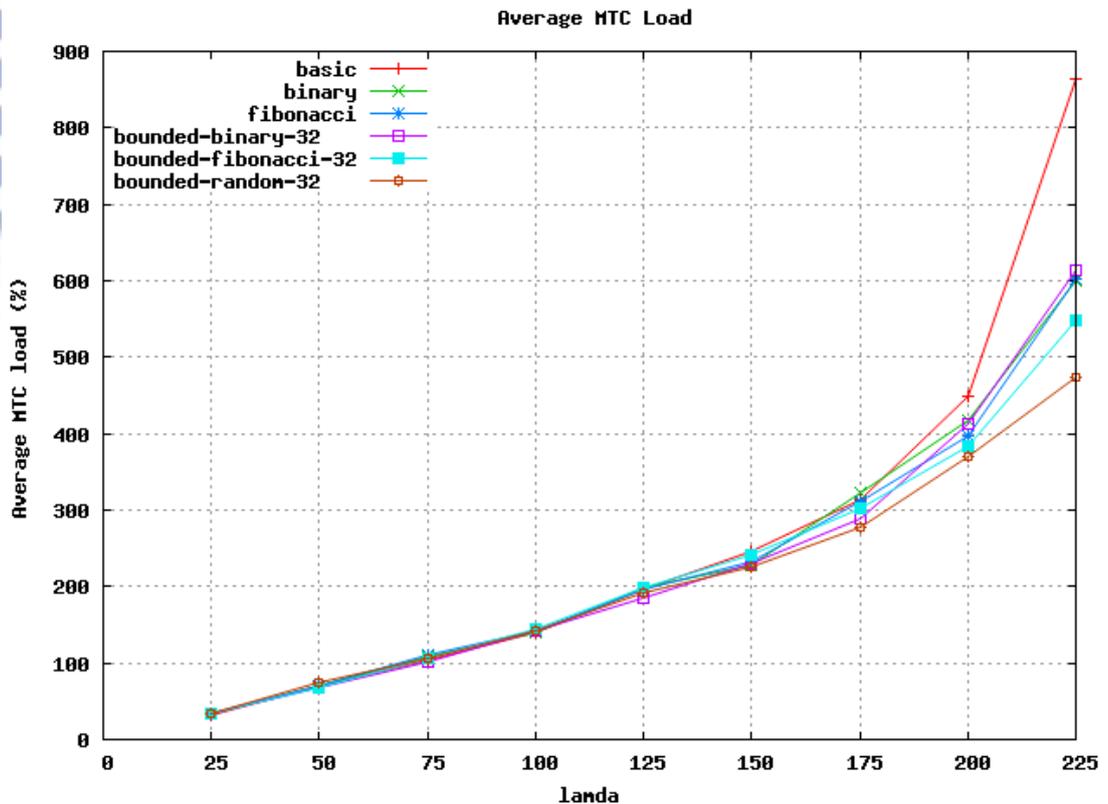
圖七 模擬結果 Average total load

(B) Average MTC Load

在這些分析中，Average MTC load 是最重要的一項，其主要的意義為各個方法之間辨識的衡量標準之一，在不同的 λ 下所得到的 Average MTC load。其計算方法為每次 time point 時，將當時所有要求連線的 MTC device 數量除以當時最大可允許 MTC device 的數量，並且將每次的 MTC 負載累加起來，在最後除以總共的 time point 次數，其公式如下。

$$L = \frac{1}{N_G} \sum \frac{N_R}{N_M} \quad (8)$$

在上述式子中， N_R 表示每次 time point 總共要求連線的 MTC device 數量，在下圖中可以發現，在高負載之前 (55% ~ 80%) 各種方法的平均 MTC 負載幾乎是一樣的，也就是在 MTC 數量不多時，不管任何方法 MME 都可以負擔並且不會造成對 non-MTC device 的影響，但是在高負載之後 (85% ~ 95%) 有著明顯的差異。



圖八 模擬結果 Average MTC load

以 Basic-based 為例，當 MTC device 數量越多，總負載越高時，MTC 負載會迅速的飆升，因為假如當某個 time point 的 N_M 很低時，大多數的 N_R 會被拒絕並且通知 MTC device 下一個 time point 再次請求連線，當下一個 time point 的到來，由於這次新的連線請求加上上一個 time point 被拒絕的連線請求，造成 N_R 的數量更大，若不幸的這次的 N_M 的值又很小，負載又會被拉更高而且會有更大量的連線請求被拒絕，造成類似雪球效應越滾越大，導致在總負載很高時，MTC 負載會被拉高。

在圖中顯示出 Fibonacci-based 和 Binary-based 比 Basic-based 的 MTC 負載降低非常多，主要是因為 Fibonacci-based 和 Binary-based 兩種方法都是採取跳躍式，分散下一次要求連線的時間點，與 WiFi 的 back off window 相似，當每次遇到碰撞(拒絕連線)，都將 back off window 乘以兩倍，來避免再次碰撞的發生。

Fibonacci-based 顯示出比 Binary-based 來的差，但是差別的幅度不太，Fibonacci-based 較差的原因是因為，其 frame 的成長幅度較 Binary-based 小，因為在 Binary-based 被拒絕跟 Fibonacci-based 一樣多次數時，Binary-based 的 frame 的長度比 Fibonacci-based 的長度大非常多，所以 Binary-based 可以更有效分散其連線請求的 time point，雖然 Fibonacci-based 稍微密集，負載也較 Binary-based 高一點，但在後面的分析中可以發現，Fibonacci-based 的延遲可以比 Binary-based 降低非常多。

Bounded Random-based 的負載比上述的相對更低，主要是因為能夠在時間範圍內將連線請求均勻的分散在時間點上，比跳躍式的 Binary-based 與 Fibonacci-based 更好的利用度，也就是不會發生某些時間點沒有分配到連線請求，或很少的連線請求。

(C) Average Grant Rate

平均允許請求率是重要的衡量標準之一，這裡的計算方式是將每次 time point 所同意的 MTC device 連線要求數量紀錄並累加起來，在模擬的最後再除以總共發送的連線請求次數，所得到的結果即為平均允許請求率，其公式如下。

$$\frac{\text{Total Grant Request}}{\text{Total Request}} \quad (9)$$

如下表所示，可以看到 Bounded Random-based 的方法有較好的平均允許請求率，雖然前面提到 Basic-based 有很高的 MTC 負載，而且在總負載高的時候如滾雪球般的越來越大，也就因為如雪球般的運作，使得 Basic-based 的平均允許請求率

稍微低一些，主要是因為在運作的過程中，中間會有 MTC 負載低於 100% 的 time point，也就因為如雪球般地滾動，以至於每個 MTC 負載較低的 time point 都沒有錯過，而且可以盡可能地在最大請求延遲的時間內被允許，到模擬結束時只剩極少數的連線請求還未被允許，導致超過 GTI 的時間，必須要等到下次 GTI 的到來。

| | 25 | 50 | 75 | 100 | 125 | 150 | 175 | 200 | 225 |
|-------------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Basic | 100 | 100 | 99.999048 | 100 | 99.999967 | 99.999958 | 99.998516 | 99.996239 | 99.999991 |
| Binary | 100 | 99.999421 | 99.998798 | 100 | 100 | 100 | 99.997935 | 99.997448 | 99.99829 |
| Bounded-Binary | 100 | 100 | 100 | 100 | 100 | 99.999681 | 100 | 99.994698 | 99.996277 |
| Fibonacci | 100 | 99.999915 | 99.998857 | 99.998842 | 100 | 100 | 99.999988 | 99.999433 | 99.999991 |
| Bounded-Fibonacci | 100 | 100 | 100 | 99.999069 | 100 | 99.99796 | 100 | 99.999486 | 99.997187 |
| Bounded-Random | 100 | 100 | 100 | 100 | 100 | 100 | 99.998386 | 99.995078 | 100 |

表二 模擬結果 Average grant rate

而另外兩個方法 Fibonacci-based 與 Binary-based 的平均允許請求率會比 Basic-based 稍微高，是因為兩種皆是採取跳躍式的方法，但是這樣運作會有兩個缺點，第一點是有可能會時常錯過 MTC 負載較低的 time point，而第二點是因為被拒絕多次時，frame 的長度過長，導致被拒絕沒幾次就超過最大請求延遲的時間，使得 MTC device 丟棄連線請求並且等待下次 GTI 的到來。

另外在 Bounded Fibonacci-based 與 Bounded Binary-based 明顯會有比沒限制的方法好，原因很簡單，當 frame 增加到限制的長度時，便會重新從長度 1 開始，一方面不像無限制的方式那樣，frame 成長太快導致一下子就超過最大請求延遲，另一方面在最大請求延遲時間內，有限制的方法比無限制的方法所發送的連線請求次數多很多，當然會被允許的機率也就增加了。

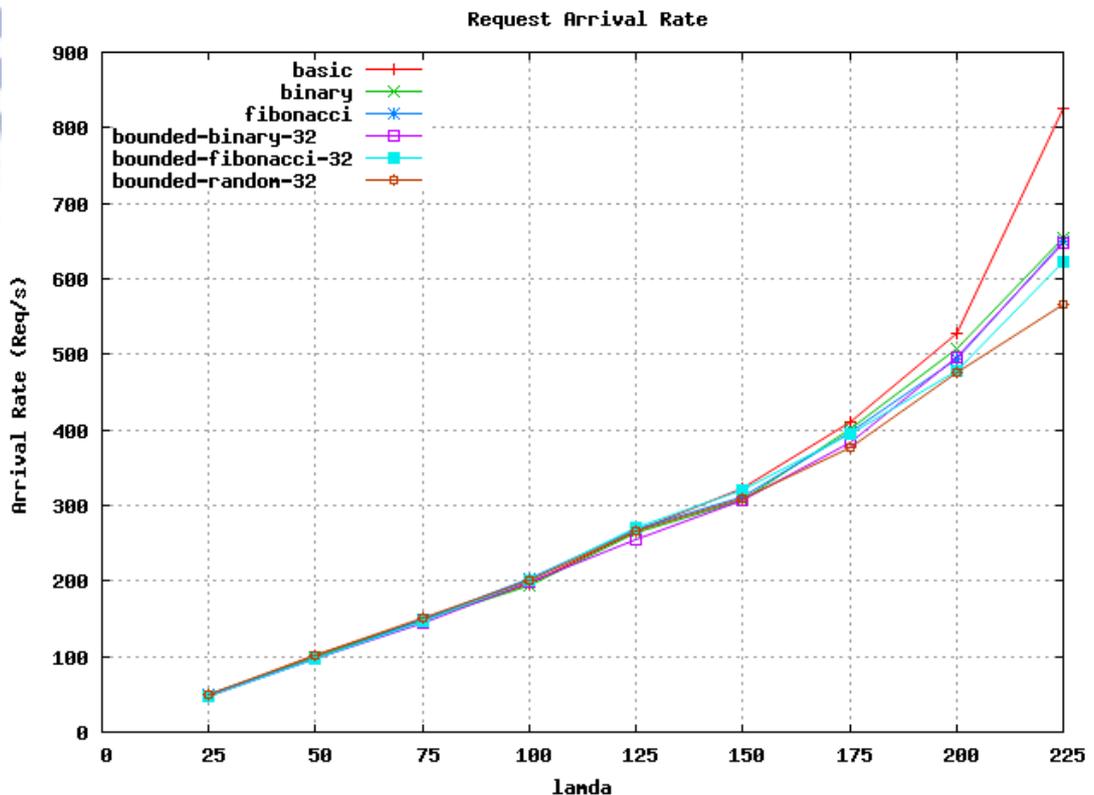
Bounded Random-based 由於有最好的分散效果，所以在有限的時間範圍內，這個方法的允許的成功機率就增加了，由上表可知幾乎都是百分之百，是所有方法中最高成功率。

(D) Request Arrival Rate

連線請求速率主要是以 MME 的角度來看，每秒所進來總共的連線請求數量，這也是重要的衡量標準之一，在這個分析中可以觀察出，每個方法間是否可以有效地在每次 time point 時降低連線請求的數量，其計算方法為每次紀錄總共的連線請求數量，並且在模擬結束時除以模擬時間，便是 MME 所看到平均每秒有多少的連線請求，其公式如下。

$$\lambda = \frac{1}{T_s} \times \sum^{N_G} N_R \quad (10)$$

如下圖所示，Basic-based 的方法可預見的是連線請求速率非常的高，這也代表 MME 的負載相對的提高，其中連線請求的數量由兩個部分所構成，一個部分是第一次發送連線請求的 MTC devices，另一部分為被拒絕的 MTC devices 重新發送的連線請求，由此可見，若方法有效的話，重新發送連線的數量亦會減少，在圖中可以看到高負載 (85% ~ 95%) 也就是 MTC device 數量相當大時，其連線請求速率開始明顯的高於其他五種方法，當 MTC device 數量越多時，其雪球效應越強，每次的連線請求數量會越滾越大。



圖九 模擬結果 Request arrival rate

另外在圖中也可以發現到其他四種方法的連線請求速率，在 MTC device 數量越大時，其連線請求速率是比 Basic-based 的方法小非常多，而且彼此之間差距不大，由此可見四種方法的分散效果是非常明顯的。而第五種方法 Bounded Random-based 在同樣限制最大 32 的 back off length 之下，效果是所有裡面最好的，正因為有效的分散連線請求，所以相對的被拒絕的請求也變少了。

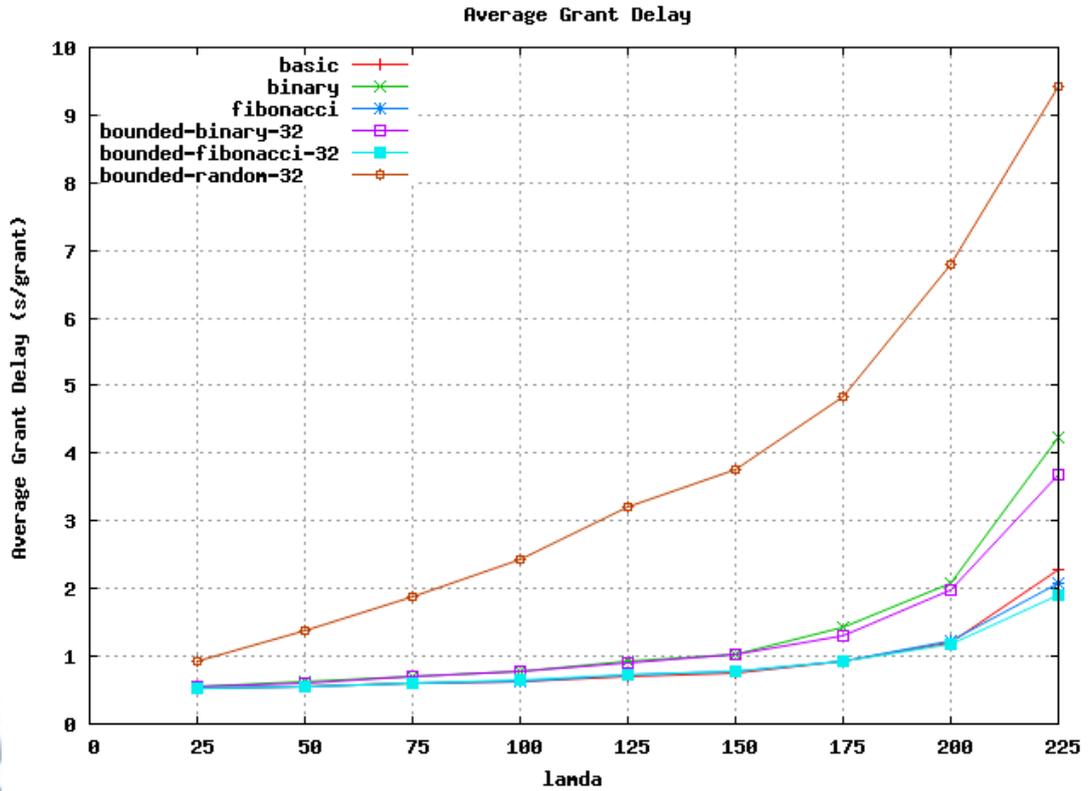
(E) Average Grant Delay

平均允許延遲是非常重要的衡量標準，我們計算從每次發送要求到被允許之間的延遲時間，它代表了越低的延遲時間可以越快的完成傳輸工作，越快完成工作就可以有越多的 MTC device 連線上來，所以方法不僅要降低 MME 的高負載問題，也要降低每次的連線延遲問題，在這裡計算方法為計算每次連線的發送時間，每當連線的接受時，計算出延遲時間並累加起來，在最後模擬結束時除以總共已允許的連線數量，其公式如下。

$$D = \frac{1}{N_{GM}} \times \sum^{N_{GM}} (\text{accept time} - \text{send time}) \quad (11)$$

在圖中可以看到 Bounded Random-based 的延遲相當的高，因為相對於其他方法來說，可能常常發生在時間區間內隨機出的時間點可能會比目前時間點較遠，由於是使用 Uniform distribution，而其他跳躍式的方法大多還未到 32 的 back off length 就已經被接受請求，所以 Bounded Random-based 的延遲會比其他的方法高非常多。

而 Bounded Binary-based 與 Binary-based 兩個方法的延遲比其他三個高出許多，主要是因為以 Binary-based 為基礎的兩個方法，其 frame 都是以指數在成長，若是五種方法都被拒絕相同次數，則 Binary-based 為基礎的兩個方法的 frame 都會比其他大很多，相對的延遲也就提高了很多。另外，有限制的 Binary-based，由於超過門檻後會重新設定為一開始，所以由圖中可見，延遲會比無限制的 Binary-based 的方法要稍微低一點。



圖十 模擬結果 Average grant delay

此外，Fibonacci-based、Bounded Fibonacci-based 與 Basic-based 三種方法是比 Binary-based 的方法好許多，Fibonacci-based 主要是因為其 frame 的成長幅度遠低於二的指數成長，所以相對的延遲也就沒有這個高，而有限制的 Bounded Fibonacci-based 跟前面提到的一樣，有限制的 frame 會比無限制的要減少許多。由圖中可知，全部最低的延遲為 Bounded Fibonacci-based，也可以看到 Basic-based 的延遲與 Fibonacci-based 差不多，

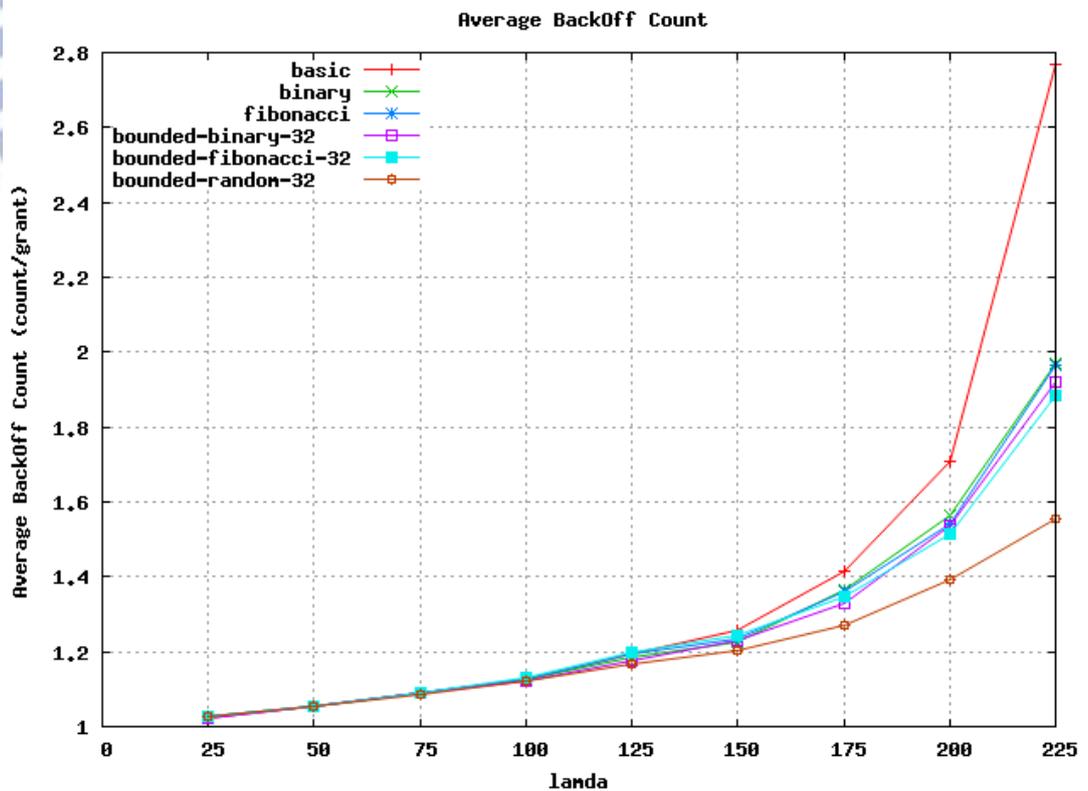
(F) Average Back Off Count

平均退回次數主要是跟平均允許延遲來互相對照，在這個分析中，主要是觀察每個 MTC device 每次發送連線請求時被拒絕的次數，在這裡次數越低越好，越低代表越快完成工作，同樣也代表有越多的 MTC device 可以連上來，而越少的退回次數也代表連線請求速率也越低，同時也代表有越低的 MME 負載。計算方法是每個 MTC device 在每次被退回時累加一次，直到被允許為止，在模擬結束時在除以總共發送被允許的連線請求次數，其公式如下。

$$BOC = \frac{1}{N_{GM}} \times \sum^{N_{GM}} \text{backoff count of each request} \quad (12)$$

在圖中可以看到，Basic-based 的方法在高負載時，被退回的次數比其他的方法高非常多，若與上個分析互相對照可以發現，雖然 Basic-based 的延遲與 Fibonacci-based 相差不多，但退回次數卻是非常高，主要是因為在同樣 frame 的長度下，Fibonacci-based 的方法會產生較少的退回次數，但 Basic-based 的方法卻是需要與更多的退回次數，因為並不是跳躍式的成長，所以其 frame 才能與 Fibonacci-based 的方法相當。

另外其他幾種方法中，其退回次數其實也是差不多的，可以看到 Bounded Random-based 依然是最低的，原因是因為 frame 是以指數成長的方式在跳躍，再加上限制，使得其退回次數是較低的。但是由於 Bounded Random-based 的分散效果又更好，所以被拒絕的次數也變少了，相對的退回次數也是最低。在一系列的 analysis 下來，雖然看起來 Bounded Random-based 的方法在大多数的分析中是較好的，但是在最重要的一樣，平均允許延遲中，Bounded Random-based 的方法相當的高，所以必須依據不同的 MTC device 應用來採取不一樣的方法。



圖十一 模擬結果 Average backoff count

四、結論

4.1 分析總結

在這篇論文中，主要是解決 3GPP 所提出來的實作議題，如何降低在大量的 MTC device 連線請求時所產生的高負載，儘管 3GPP 提出在每個 time point (例如相隔一秒) 時，判斷每個 MTC device 是否可以連上網路，若是被拒絕要告知 MTC device 何時再發送連線請求，我們主要是模擬解決告知 MTC device 何時發送連線請求可以降低 MME 的負載。在我們所提出的方法架構中，我們改變原本的議題中，MME 每次拒絕 MTC device 時都要告知下次發送連線請求的時間，而是變成在每個 MTC device 自行計算下次所要發送連線請求的時間，而且 MME 只需在第一次拒絕 MTC device 時告知 frame 的大小，其主要的目的是減少網路中訊息的交換，儘管我們提出的方法計算量不大，並將計算移至 MTC device 來運作，但在 MME 處理大量的情況下也是不無小補。

我們所提出的六個方法中，Bounded Random-based 在多項分析中，效果都好於其他的五種方法，但是在平均延遲分析中，遠遠高於其他的方法。此外，雖然 Bounded Fibonacci-based 在前面的分析中都稍微高於 Bounded Binary-based，但在平均允許延遲的分析中，Bounded Fibonacci-based 卻是遠遠低於 Bound Binary-based，因此衡量之下還是建議用 Bounded Random-based 來處理大量連線的 MTC device，不僅可以有效降低 MME 負載，而且可以針對每類 MTC device 的最大允許延遲來相對的調整，這也代表每個 MTC device 可以盡快地完成相關的工作。

4.2 未來改進

在未來改進部分，我們所提出的方法中，除了 Basic-based 之外，其他的平均允許率都較低，這在 MTC Application 中也是相當重要的一項指標，因為我們在 MTC 發送要求時，有最大請求時間，若是超過最大請求時間，便放棄在這次 GTI 做連線的動作，在未來我們希望可以導進以接近最大請求時間為優先權的判斷方式，在有限制的跳躍式成長的 frame 之下，以越接近最大請求時間的 MTC device 為優先連上網路，使得在跳躍式的方法下，也可以增進平均允許率。

參考文獻

- [1] 3GPP specifications –System improvements for Machine-Type Communications (MTC) TR 23.888-1.6.1, February (2012).
- [2] 3GPP specifications –Architecture enhancements to facilitate communications with packet data networks and applications (MTC) TS 23.682-11.1.0, June (2012)
- [3] 3GPP specifications –Study on enhancements for MTC, TR 22.888-0.76.0, May (2012)
- [4] 3GPP specifications –Study on facilitating M2M communication in GSM and UMTS, TR 22.868-8.0.0, March (2007)
- [5] 3GPP specifications –Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification, TR 36.321-10.4.0, December (2011)
- [6] 3GPP specifications –Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS), TS 24.301-11.3.0, June (2012)
- [7] 3GPP specifications –Service requirements for Machine-Type Communications (MTC), TS 22.368-11.2.0, June (2011)
- [8] IEEE Std 802.11g Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June (2003).
- [9] Giuseppe Piro, Luigi Alfredo Grieco, Gennaro Boggia, Francesco Capozzi, and Pietro Camarda, “Simulating LTE Cellular Systems: an Open Source Framework,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, Feb, 2011
- [10] Zukang Shen, A. Papasakellariou, J. Montojo, D. Gerstenberger, Fanli Xu, “Overview of 3GPP LTE-advanced carrier aggregation for 4G wireless communications,” *IEEE Communication Magazine*, vol. 50,no. 2, pp. 122-130, February, 2012

- [11] F. Capozzi, G. Piro, L. Grieco, G. Boggia, P. Camarda, "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey," *IEEE Communications Surveys & Tutorials*, no. 99, pp. 1-23, 2012
- [12] M. Salem, A. Adinoyi, M. Rahman, H. Yanikomeroğlu, D. Falconer, Young-Doo Kim, Eungsun Kim, yoon-Chae Cheong, "An Overview of Radio Resource Management in Relay-Enhanced OFDMA-Based Networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 422-438, 2010
- [13] E. Pateromichelakis, M. Shariat, A. Quddus, R. Tafazolli, "On the Evolution of Multi-Cell Scheduling in 3GPP LTE / LTE-A," *IEEE Communications Surveys & Tutorials*, no. 99, pp. 1-17, 2012
- [14] Ming-Yuan Cheng, Guan-Yu Lin, Hung-Yu Wei, A.C.-C. Hsu, "Overload control for Machine-Type-Communications in LTE-Advanced system," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 38-45, February, 2012
- [15] K.I. Pedersen, T.E. Kolding, F. Frederiksen, I.Z. Kovacs, D. Laselva, P.E. Mogensen, "An overview of downlink radio resource management for UTRAN long-term evolution," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 86-93, July, 2009
- [16] Shao-Yu Lien, Kwang-Cheng Chen, Yonghua Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66-74, April, 2011
- [17] Martin Sauter, "An Introduction to mobile networks and mobile broadband," A John Wiley and Sons, 2011