●**謝仁偉教授**

1. Lifetime定義，單位?

Lifetime定義為在整體網路中系統開始到最早耗電結束的感測器時間

各感測器lifetime計算會以電池電量(mAh)除與平均電流(A)，計算出單位為hours

1. 對於碰撞問題以往是用CSMA/CD做處理，那此系統是否可行?

以往的網路各個感測器可互相分享頻寬進而讓主控台做監控，但在BLE是會以點對點互相分享以除去監聽其他的感測器來達到省能，因此以CSMA/CD方式無法實現在此架構當中。 CSMA所耗的電流為27mA相比於BLE的peak current為17.5mA，其所耗的電流相對的大。

|  |  |
| --- | --- |
| Classic bluetooth | Bluetooth Low Energy |
| F:\文件夾\M10207432\Report\paper\BT_topology1.jpg | F:\文件夾\M10207432\Report\paper\BLE_topology1.jpg |

(<http://www.summitdata.com/blog/ble-overview/>)

|  |  |
| --- | --- |
| Zigbee (CSMA phase3-4) | BLE |
|  |  |

1. Earliset Interval可能會誤導意思，因可改為the shortest interval

有把命名取為The Shortest Interval First

1. waiting time是?

指的為master端需要等待感測器回應的時間，此時間就為各個感測器的connection interval。

1. 會有其他的拓鋪方式嗎? 為何選用star?

拓鋪有含星狀(star)和mesh兩種，選用星狀理由為在我們的real case議題底下是針對body sensor network，因此會以星狀方式較為適合。**(MBStar : A Real-time Communication Protocol for Wireless Body Area Networks)**

1. Lazy是指哪方面lazy? 為何load會在某時間點變多?

🡪Lazy的調整方式類似於Linux ondemand governor調頻方式，lazy指的是會在一定時間內做connection interval校正來達到省能，倘若有application load增加導致現在的interval無法服務就會採用最高速來服務。**(Adaptive Online Power-Management for Bluetooth**

**Low Energy)**

🡪因為會變化connection interval，所以在sleep duration時load會逐漸累增。

Buffer

Time

Time

Send

Application Arrival

Send

Send

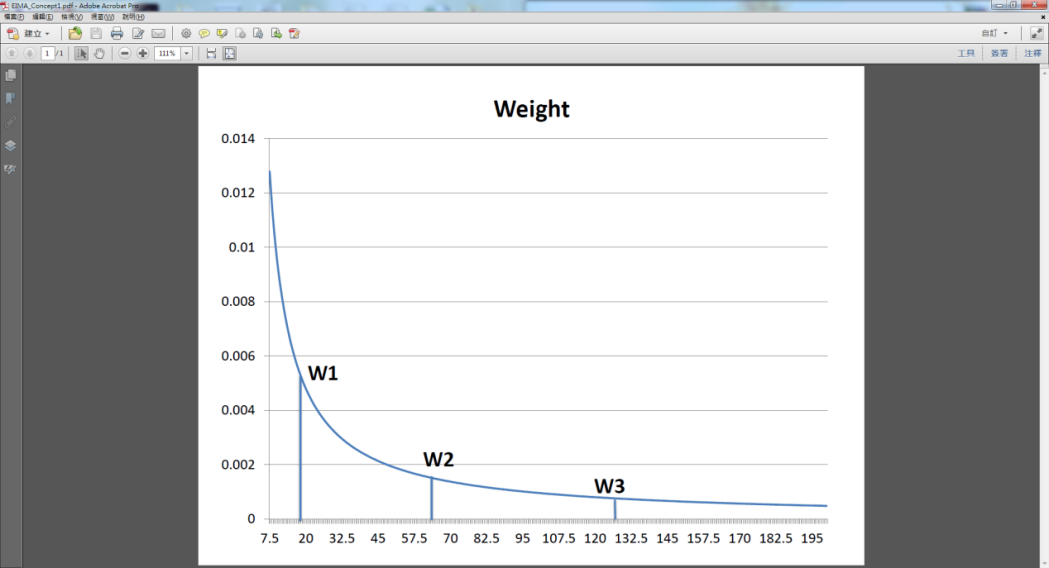
Connection event

1. Meet ratio定義?

參考文獻當中所提出的表現方法，實質在規定時間(deadline)下接到的封包數量比上所有接到的封包數量比例做為meet ratio表現。**(MBStar : A Real-time Communication Protocol for Wireless Body Area Networks)**

1. 提出的方法EIMA因該要100% meet ratio?

在low data rate情況下，各個service interval差距過大，導致最大的interval還是會block最小的service interval，因為整體的write-request schedule是屬於non-preemption且目前並沒有最佳的schedulability test在此議題下，所以EIMA只含有所計算出的slave中connection interval比service interval總和不超過1



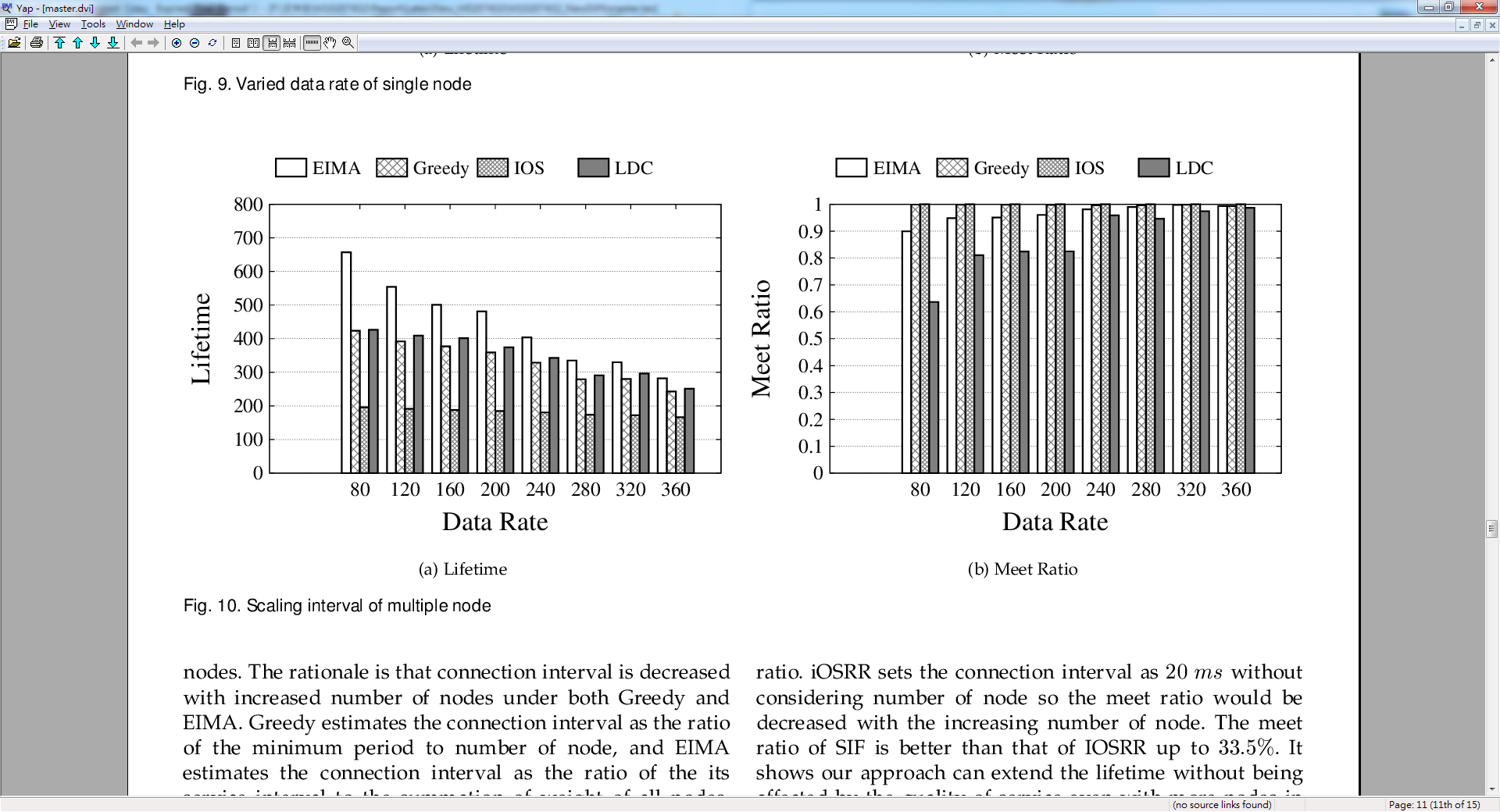
|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
|  |  | Connection event current |
|  |  | Connection event duration |
|  |  | Sleep current |
|  |  | Service interval of i th node |
|  |  | Connection interval of i th node |

1. 參數的定義要清楚，不同load耗能?

不同load下，量測出的電流都差不多，有加入說明於論文中。

1. 實驗結果可以做複合呈現，可定義多少時間點下的meet ratio能達到的quality，因該要比出自己的優勢

現在的實驗會將data rate的組數變小只取中間較有差異性的比較，使得觀看上不會過於狹窄，會以lifetime和meet ratio做互相比擬。



●**修丕承博士**

1. Lifetime只看slave還是master?

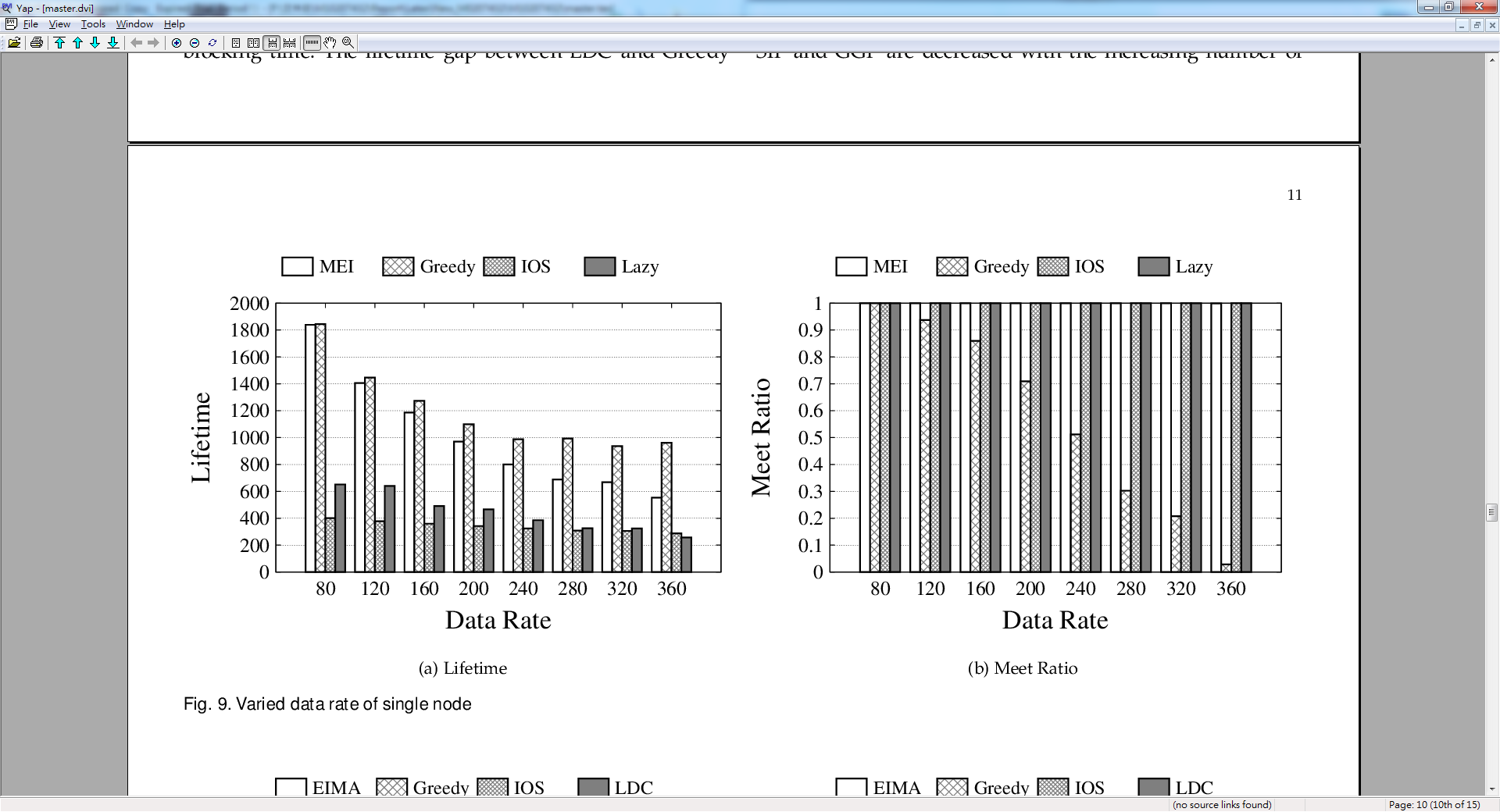
只看salve端，因為大多slave端是用電池做供電以利於放於生活周遭，而master端監控端會用一般插座供電。

1. Related work中相似於Adaptive online power-management for Bluetooth Low Energy，差異為?

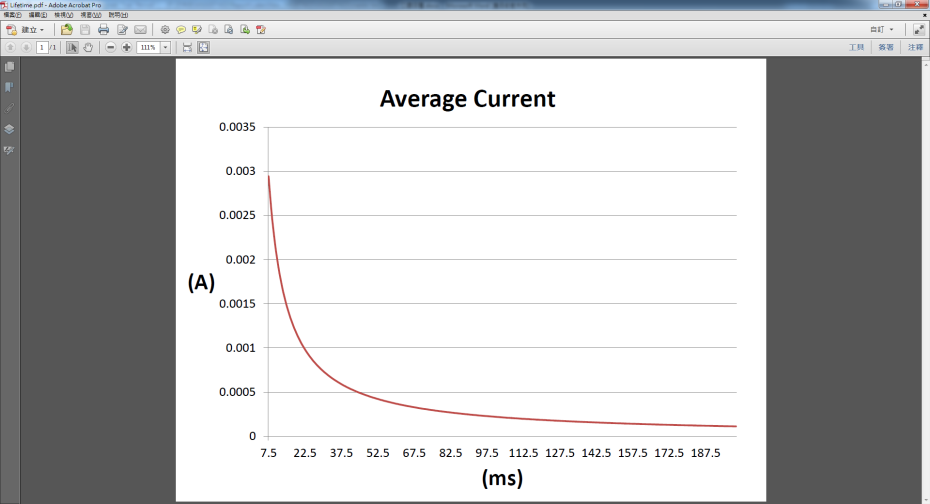
因在動態調整當中會有調整時間上的overhead，使得real-time 議題底下會有嚴重的延遲性問題，除此之外動態調整在multiple slaves下議題會更加複雜，動態調整會需要考量master需要等待的slave的時間和彼此間block的時間，使得動態議題上會更為複雜。

1. 一對一中，對於我們方法和dynamic方法不同處為? 必須有一理由說明為何用Static取代Dynamic，因Dynamic有彈性可以解決不同情況(在論文內容)

Dynamic在real-time議題下會需要考量調整時候的overhead來達到省能，我們Static方法會根據已知的input data去除overhead來達到real-time效用，若要用dynamic我們採用論文中的Lazy方法來作比較，但實質效益並不會比我們static方法好，因為Lazy的方法會在判斷是否要調整為最短connection interval來服務所有的event。



1. Weight的curve是如何計算出來的? 不同device下weight曲線是否會改變?

權限是利用與lifetime有相關的average current來做計算，我們將計算出average weight來做調整，那weight曲線會因為不同硬體上的active current和active time參數會有所不同致使曲線會有高低落差，但決定變化的還是為connection interval。

1. Packet來的時間是已知? 在BLE中是否為periodic?

Application來的時間會以periodic方式做，而在BLE當中的sensor device會以固定的sample rate來做擷取資料，我們將設定各個application 的quality period來做為各個application period，在BLE當中除了application會週期性擷取資料之外在link layer當中的connection interval也是會週期性的服務現有的sensor data。

**(WIRELESS COMMUNICATION PROTOCOL BASED ON EDF FOR WIRELESS BODY SENSOR NETWORKS)**

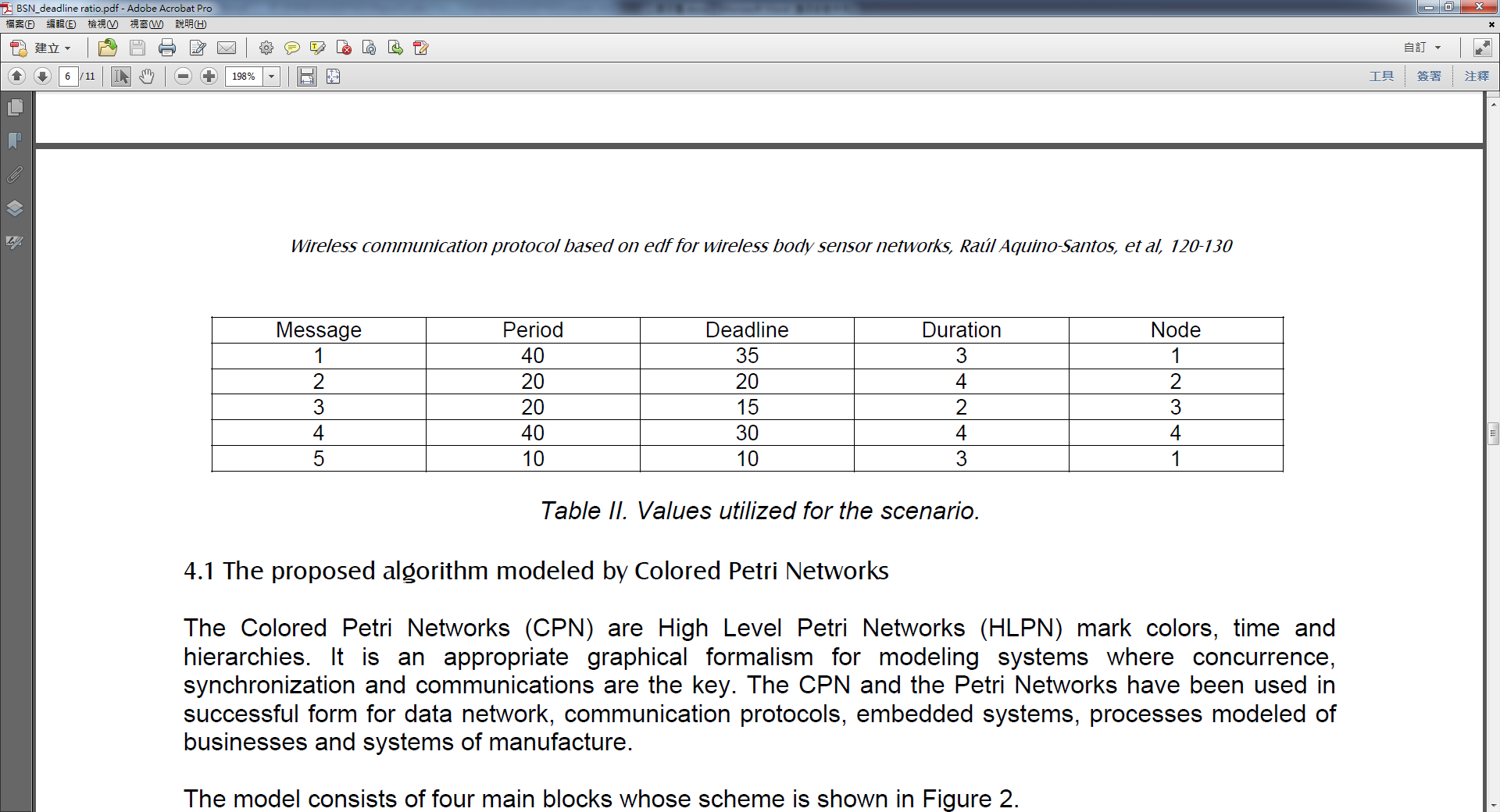
**(A Deadline-Aware and Distance-Aware Packet Scheduling Algorithm for Wireless Multimedia Sensor Networks)**

1. 若application period比service interval還短該如何處理?

在我們的MEI演算法下，一開始就會以最小的period來做計算，因此不會發生application period比service interval還短的問題。

1. Deadline的定義會在?

我們依據文獻1當中提出針對各個node上period的倍率設定方式來做，並且數值是依照文獻2當中的ECG變化時間來設定，大約455ms~457ms。



文獻1: **WIRELESS COMMUNICATION PROTOCOL BASED ON EDF FOR WIRELESS BODY SENSOR NETWORKS**

文獻2: **Adaptive Online Power-Management for Bluetooth Low Energy**

1. Native作法 (Service Interval或Connection Interval)?

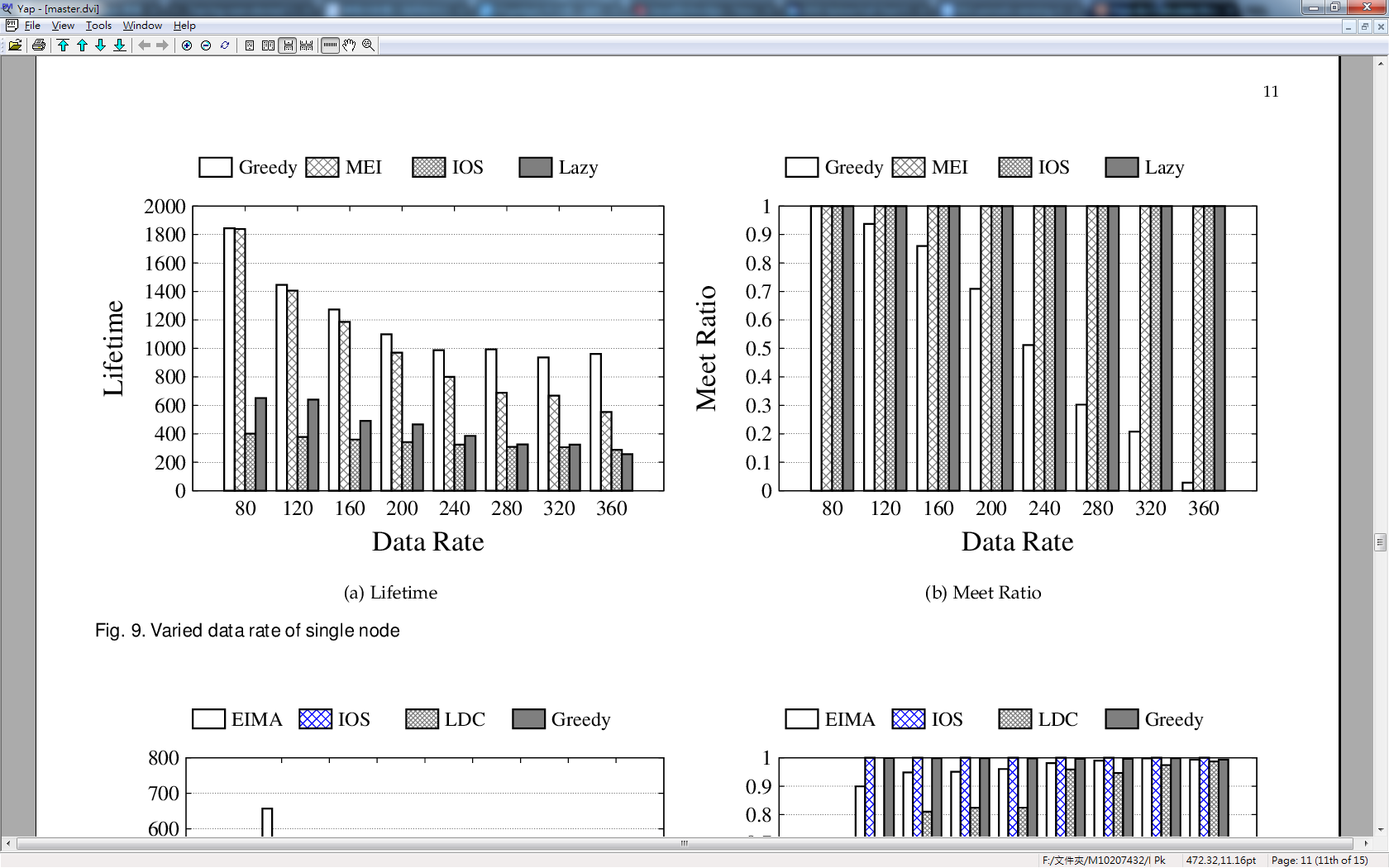
在Native作法上，是按照IOS的設定方法來比較，在單一對一議題底下IOS會調整為40ms[文獻1]，在multiple slave下我們依照文獻2提出的方式做connection interval assign (20ms)。

文獻1: **(http://lists.apple.com/archives/bluetooth-dev/2012/Dec/msg00023.html)**

文獻2: **Bluetooth Low Energy for Data Streaming: Application-level Analysis and Recommendation**

1. Native方式若會有miss問題發生那廠商該如何解決?

在我們的議題下會有real time方式，以往是沒有，但就IOS系統下master會將slave端直接調整為40ms，因此在這如此短的interval就能夠服務完我們所有的application，底下是對於一對一的實驗數據。



1. EIF和NPEDF方法在meet ratio上差異性不多，彼此間的差異? Data Rate往小延伸?

在我們的EIF會是以NPEDF概念方式做，但差異性是在request時會有加入block上額外的判斷，讓meet ratio可以提升。若Data Rate往小的部分做其結果差異性也不高，因為一個connection event是能傳輸80bytes，若往80bytes/s以下調整則數據差異不大。

1. 實作中高速情況下EIF還是有些miss?

那是因為高速情況下EIF還是會有些IDLE time的發生而導致miss，且實作上會有application 的offset (其他process造成delay)造成miss，但meet ratio結果也是可以達到95%左右。

1. 實作上lifetime是如何取得?

我們會取500ms的實測出來的資料(電流值)，將電流值乘上sample period計算出電量，用電量除上500ms即為平均電流，最後用230mAH除上平均電流為lifetime。

1. 若Buffer增加，就不會有collision問題，因該是master同一時間只能和一個slave做溝通，並非buffer上的議題。

這是會與packet對於同一個master schedule有相關，所有slave中的packet放於buffer中master要取決於先後接收順序。在collision 因素上有一點是關於一個master只對單一slave做同步，在此時間中master無法接聽到其他slave資訊，因此若有其他slave傳輸則會有封包遺失現象。

**●吳晉賢教授**

1. 是決定一個interval就不會再調整，或是再加入一node再調?

我們的方式是會決定完後就會以此interval來做服務，相對於動態調整我們的方法可以減低overhead上造成delay的問題。

1. 現今sensor network是以BLE還是Zigbee?

工業上會以Zigbee較為主流，在穿戴是裝置上會以BLE來運作，在我們的情境是放在healthy care上因此較適合的radio protocol為BLE。

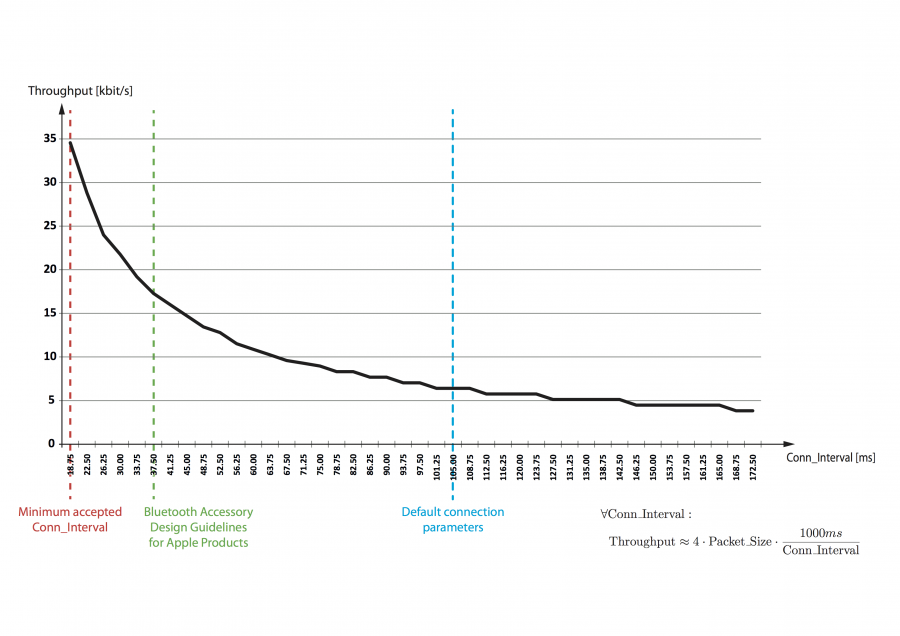
1. 一個master可接幾個slave?

要看各家廠商設計出可支援的數量，目前最多8個(Nordic)。

**(https://developer.mbed.org/questions/6130/Can-I-create-a-master-and-slave-in-one-n/)**

1. Service interval是有決定要送多少封包? 那Interval越長是能送越多嗎?

Service interval是傳輸間格時間即為兩個connection event彼此的間距，若越長表示封包回應速度越慢，interval越短才能傳輸越多。下圖顯示出若connection interval越短則throughput越高反之則越底。 **(https://atmosphere.anaren.com/wiki/Data\_rates\_using\_BLE)**

****

1. Slave包含的packet數量，可以分析為一個slave含一個packet，所以Slave數量就為packet數量，那這樣和包含在一起的數量差異是? 為何不看成是單一slave中load加總看最小period即可? 為何不可以將packet合在一起?

🡺若區分一個slave含一個packet，則每一個slave的data rate就不會太高使得計算單一node的interval就無必要性，變為在多slaves的議題下會需要考量，我們有加入多slave情況下我們的演算法和比較組情況。

🡺若看成load加總做計算，會變為沒有packet概念，使得不同application下的data需要做封裝進入一個packet當中，才能被服務，倘若不封裝為一個packet就直接加總load計算，會使得interval計算過大導致application無法在規定時間下被服務。

🡺所有application被封裝一起做發送，需要在application就需規劃完成，我們取而代之的為在application就做好packet封裝於buffer當中。

1. Lifetime是指slave還是master?

Lifetime是取決於slave端，因為我們的master是取決於插座供電。

1. 可以考量master中的耗電量?

在master當中是會以輪循方式對slave做request，因此slave的interval即為master上的interval，在multiple node情況下，master的connection event將會增加為node數量，但在不同演算法上master的interval也會隨之變動進而達到省能效果。

1. 考量現在的slave電量進一步做延伸?

加入此部分的話將會是在EIMA演算法中做延伸，其weight就需要改為各自node lifetime的倒數，在式子中是和我們原先方法一樣(battery一樣)，因目前須考量battery狀態所以須將battery狀態放入。