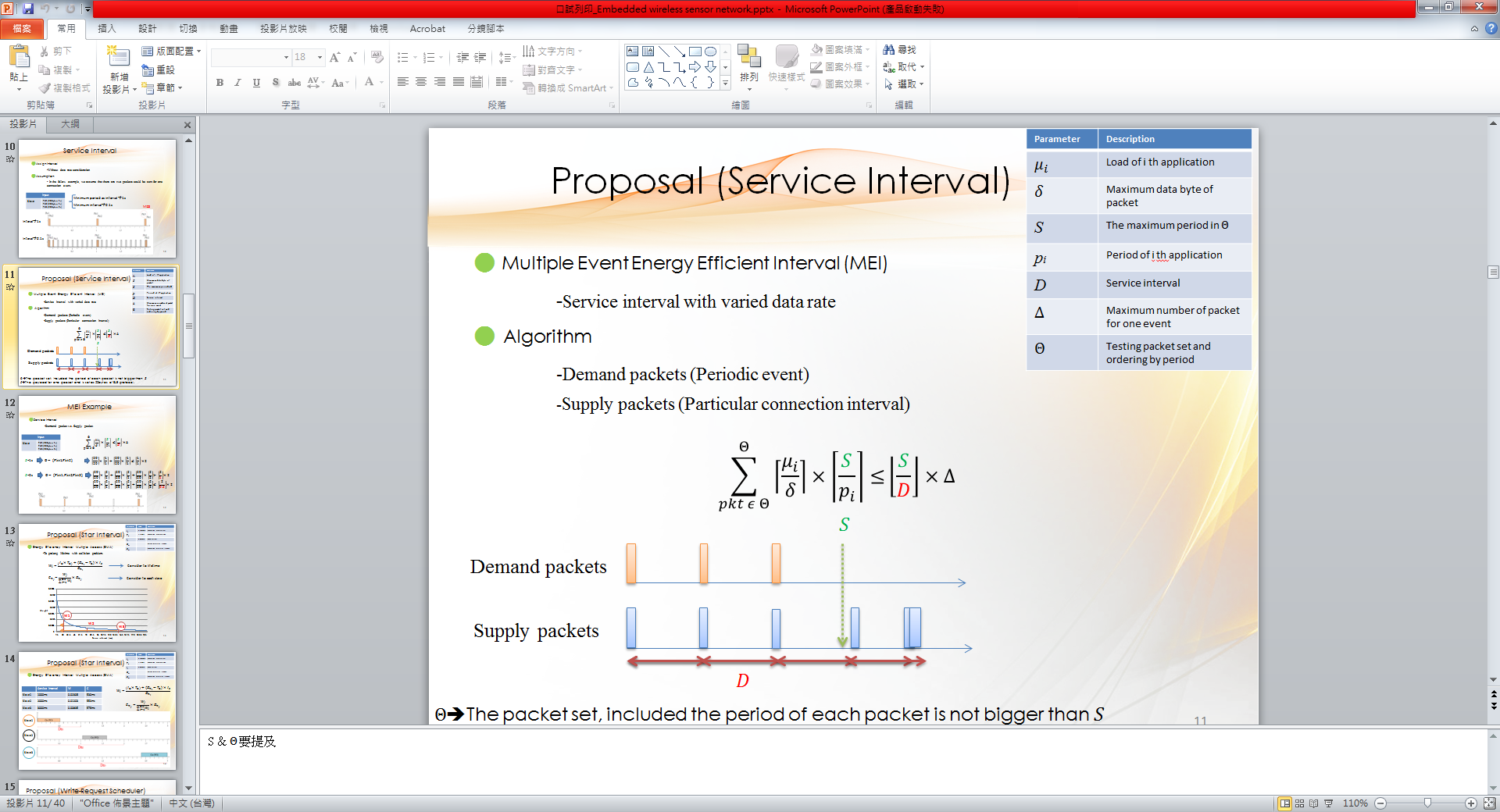
**論文內容**

●系統架構:

|  |  |
| --- | --- |
| **Parameter** | **Description** |
|  | Load of i th application |
|  | Maximum data byte of packet |
| *S* | The maximum period in |
| *pi* | Period of i th application |
| *D* | Service interval |
|  | Maximum number of packet for one event |
|  | Testing packet set and ordering by period |

●**MEI** : 處理不同data rate下對單一node上interval計算，此interval為服務所有application最緊的時間區間 (Service Interval)



●**EIMA** :

概念:利用average current曲線作為各個slave的weight assignment曲線，主要是用此weight表達會造成的耗能多寡。用佔據總體weight的比例去做connection interval計算

若比例高表示一開始一開始耗能高，所以decrease量不能多

若比例低表示一開始一開始耗能少，decrease量要多以免造成太多blocking time

●**EIF** :

●實驗設定:

在不同的data rate下做不同的load比較

(各個slave的period分別為200ms, 400ms, and 800m左右)

**模擬內容**

●檔案區分

●Read Input File

●Service Interval (Single node)

**🡪Multiple Event Energy Efficient Interval (MEI)**

校正Service Interval

Supply packet

在S時間點之前

系統提供的pkt數量

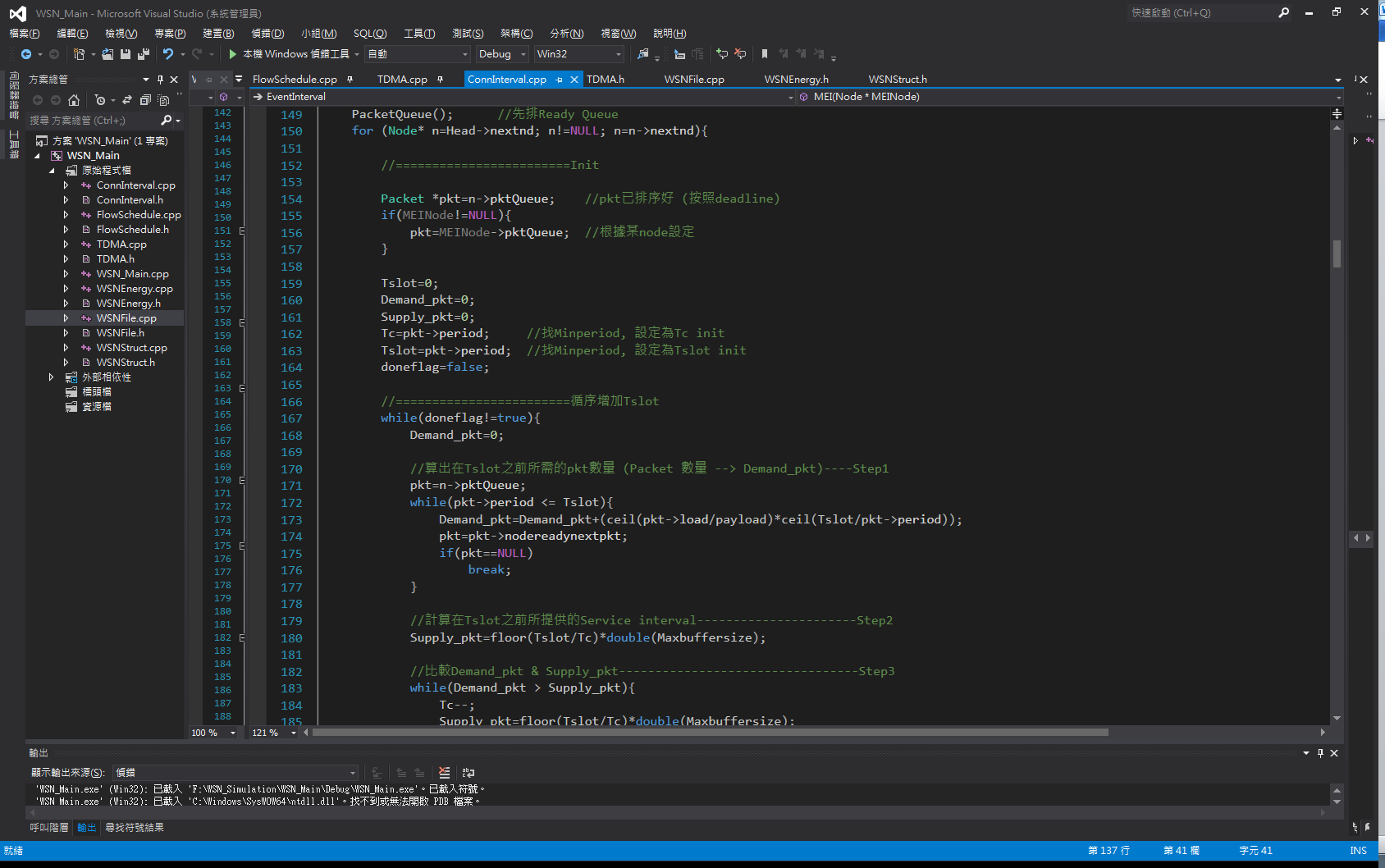
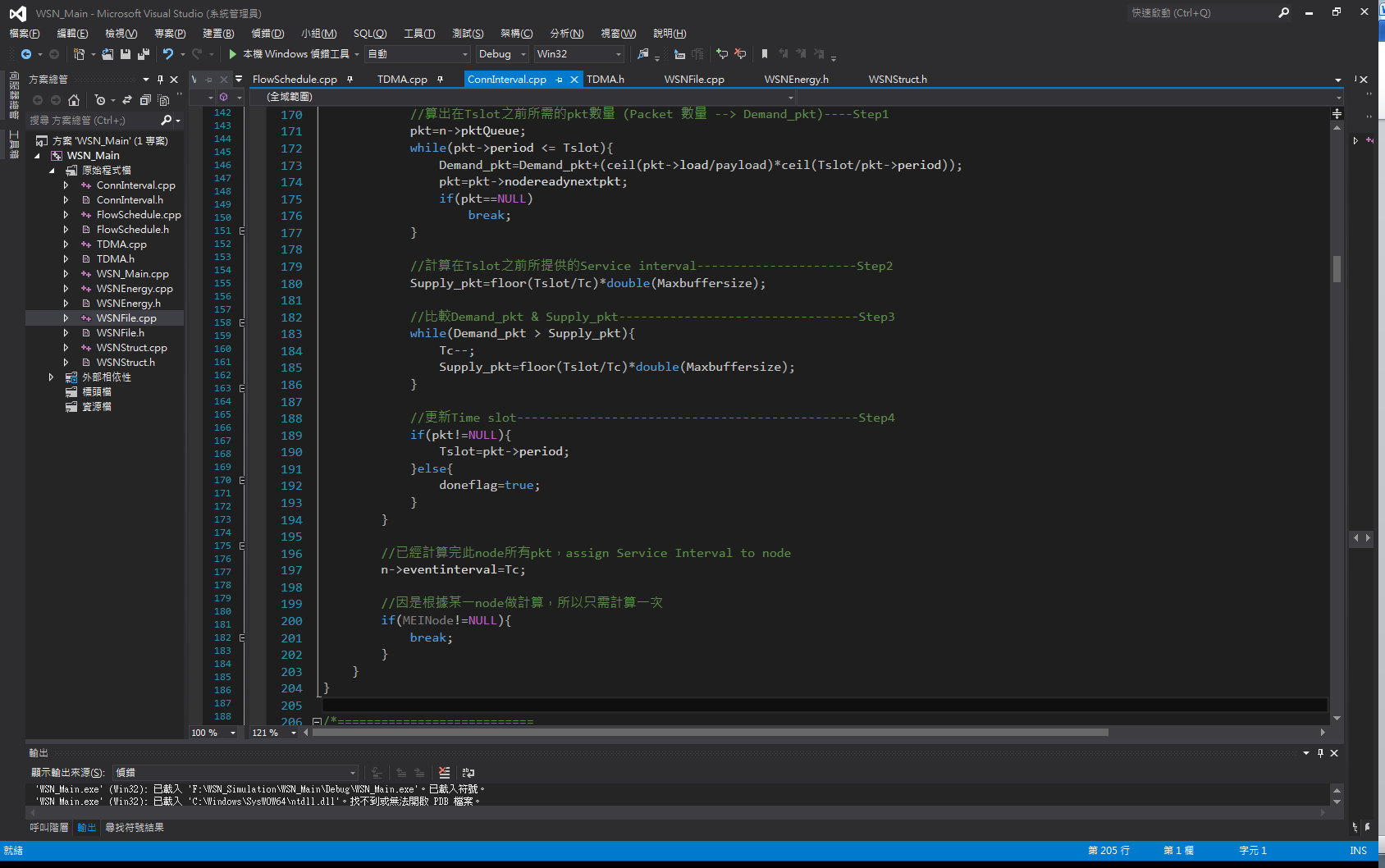
Demand packet

在S時間點之前會有的pkt數量

要先排好優先權

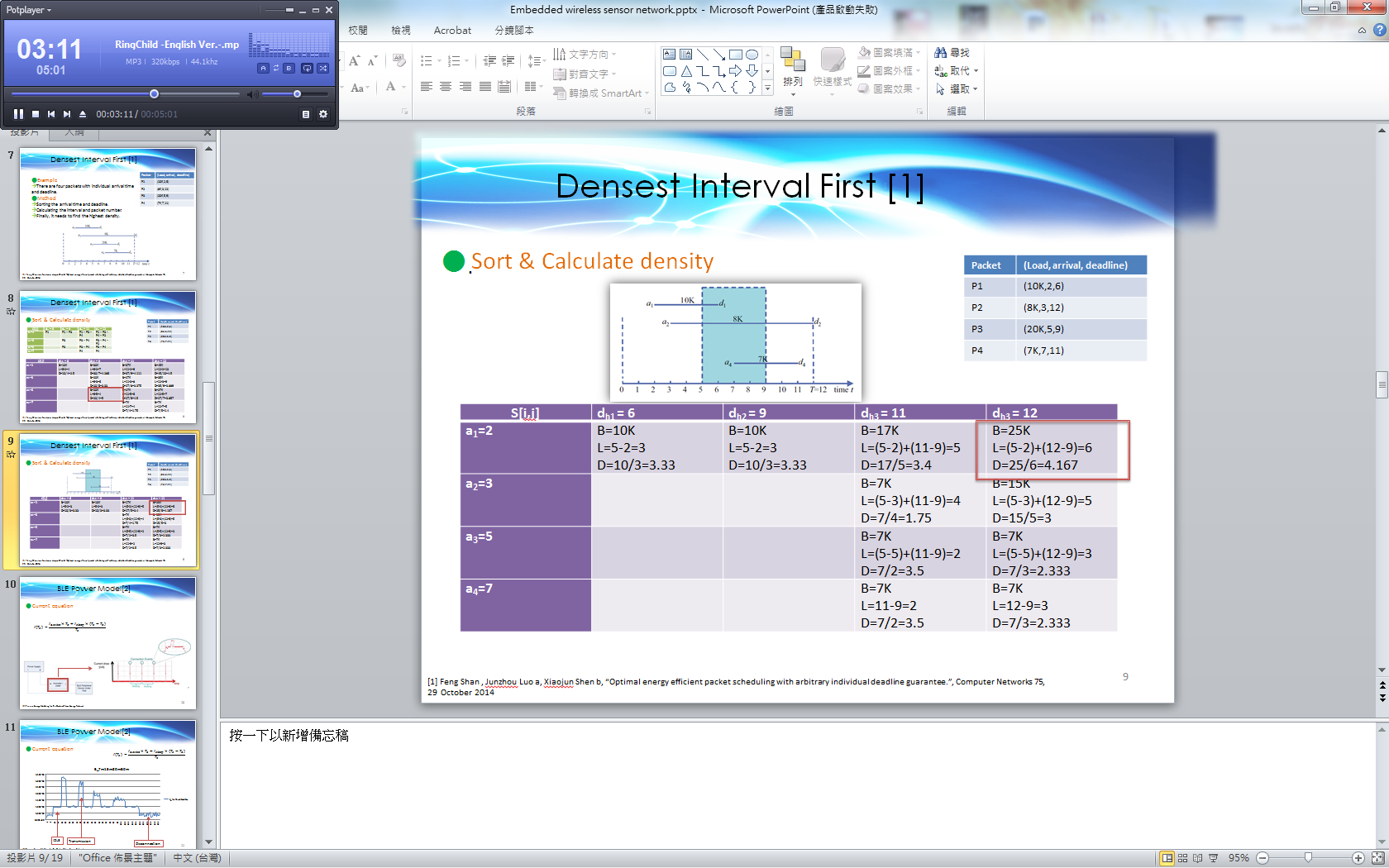
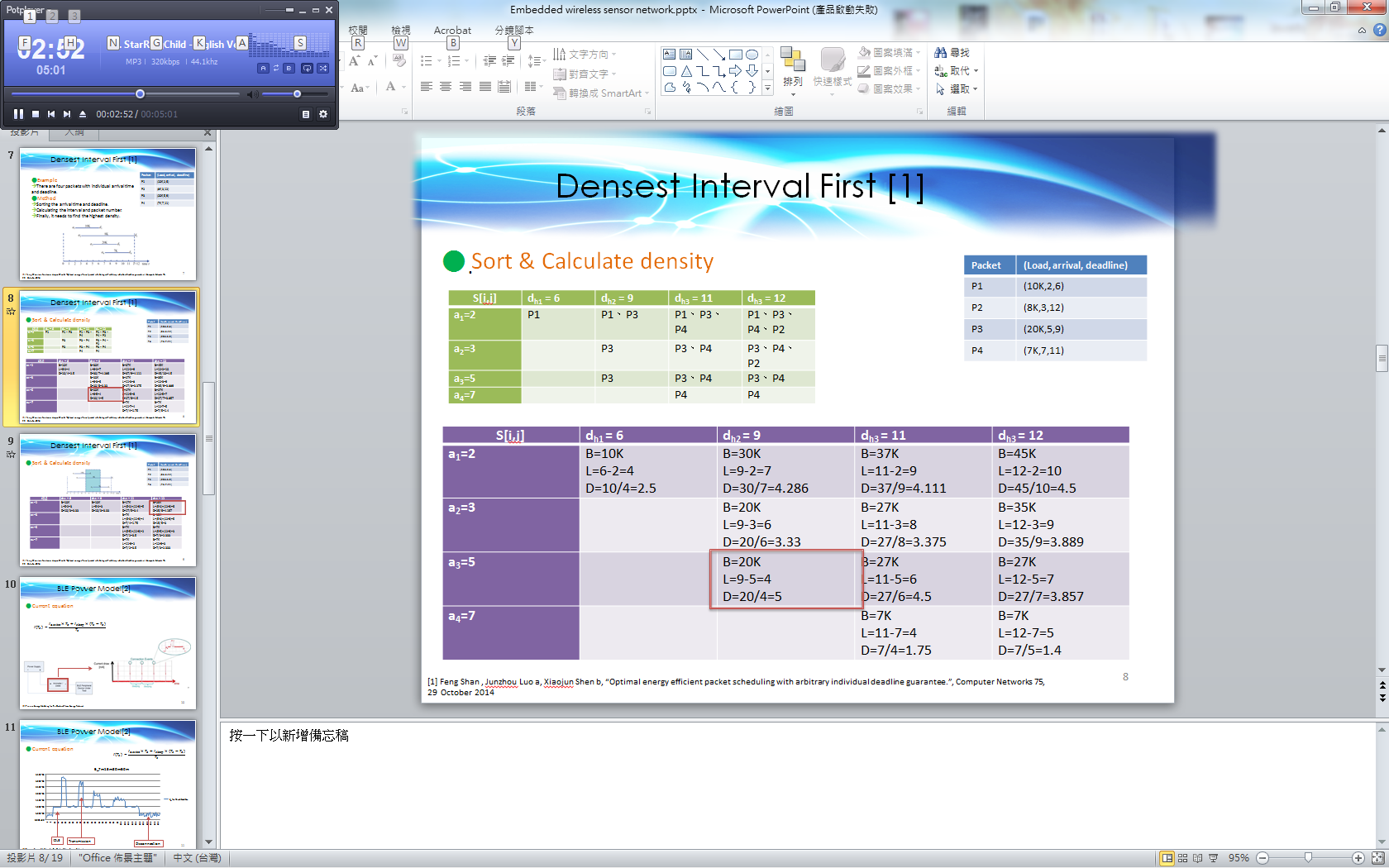
設定好起始Service Interval

S (Tslot)基準點



**🡪Densest Interval First (DIF)**

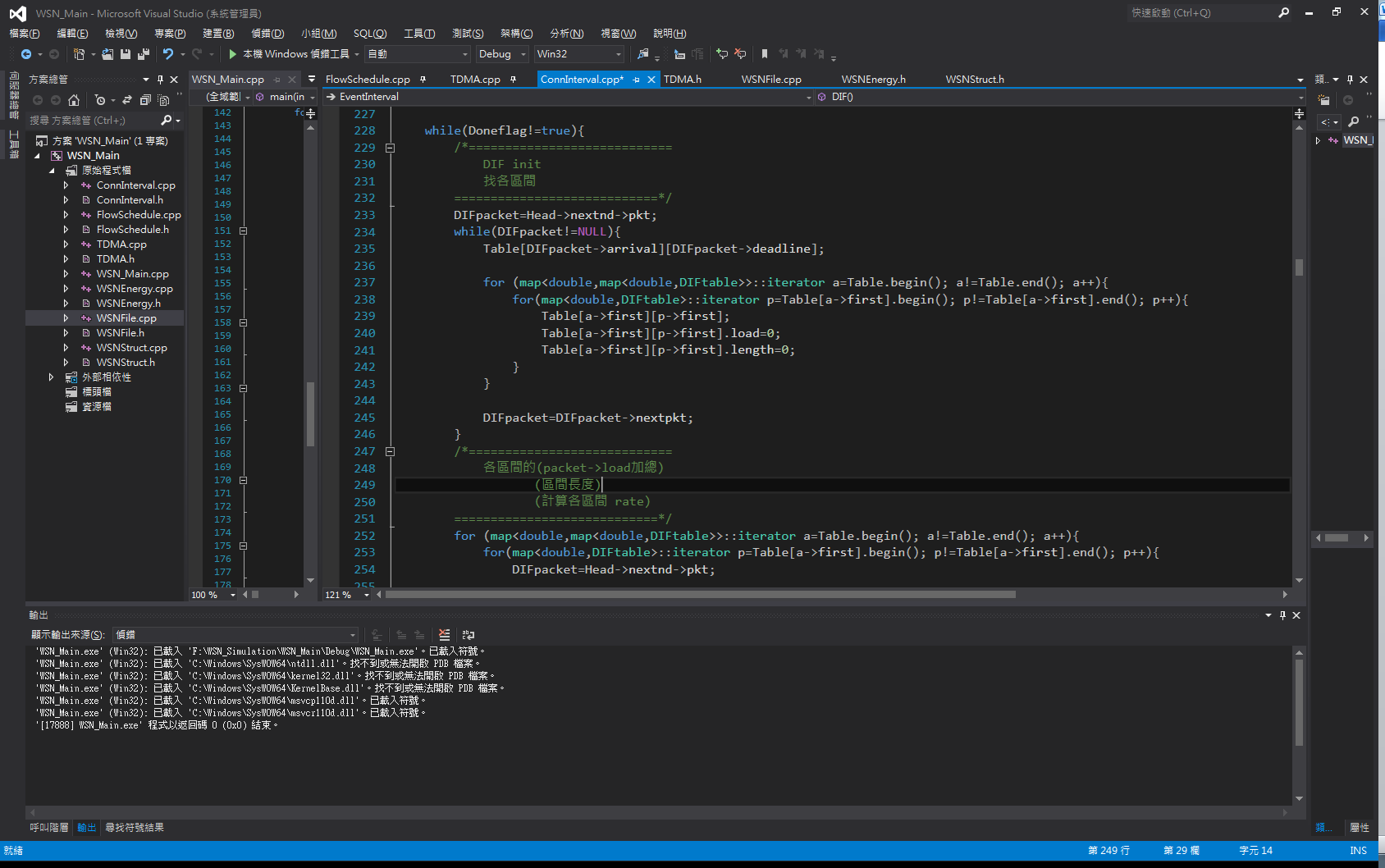
概念: 區間內找最大rate做assign，在assign 長度時要排除已經assign過的區間，區間內的封包定義為封包arrival大於區間起始點，deadline小於區間終點



Map🡺為key對應value或struct的架構 (map<key, value>)

Map->first為key

Map->second為value或struct

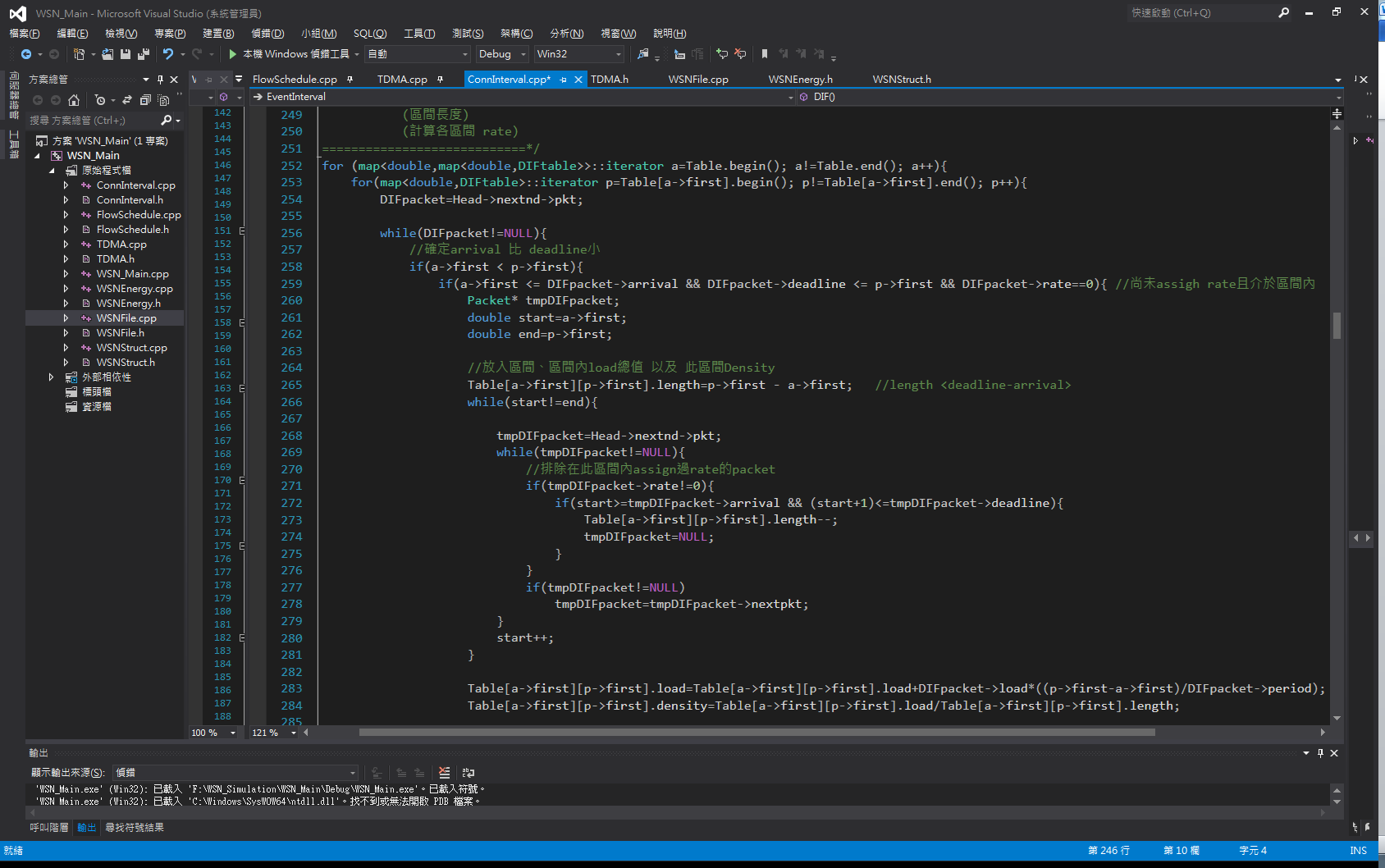


Init

二維map

(**first表示key**, second表示struct)

(Table[arrival][deadline])

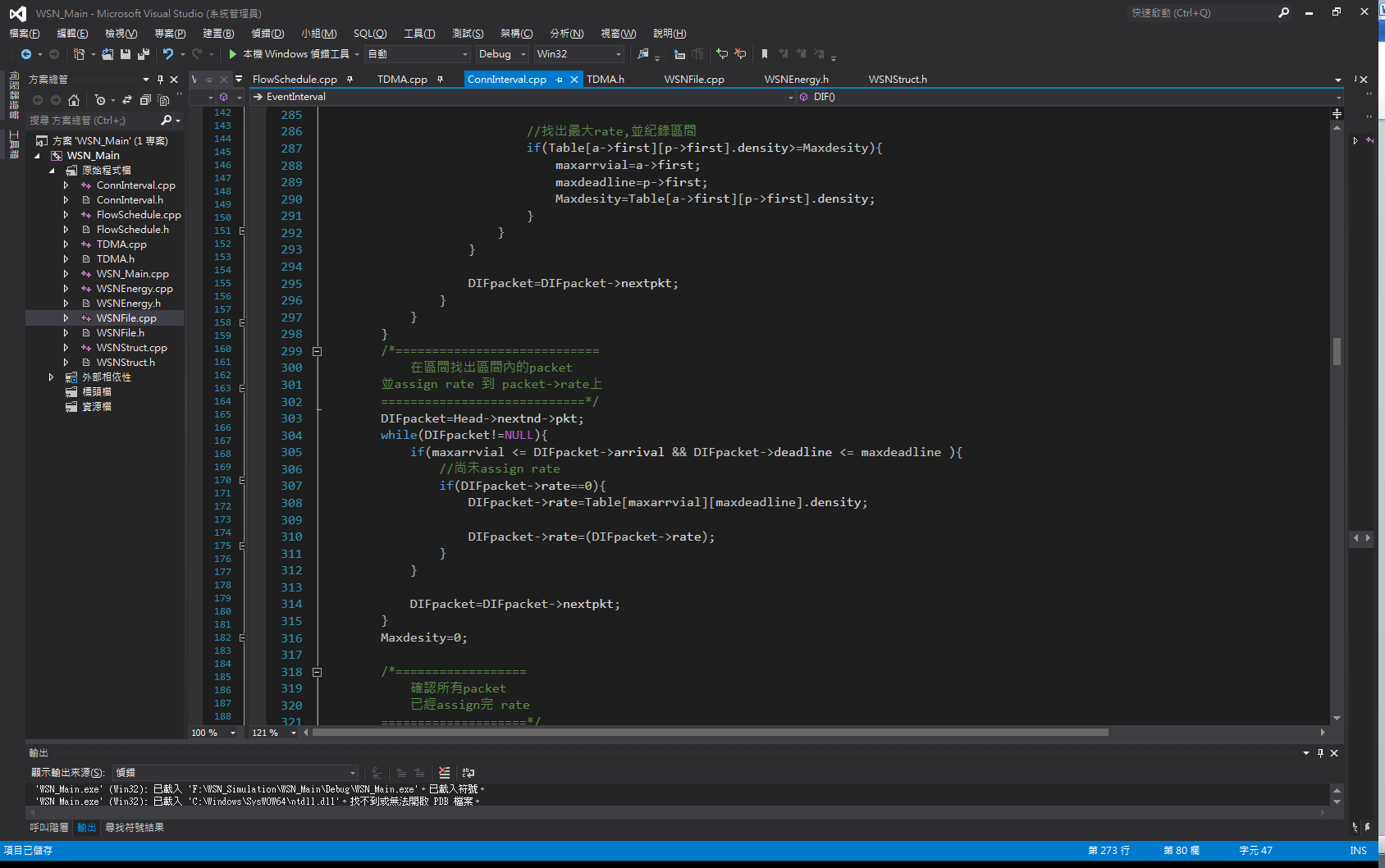


刪除已assign過rate的區間

a->first為arrival

p->second為deadline

(確認DIFpacket在區間內)



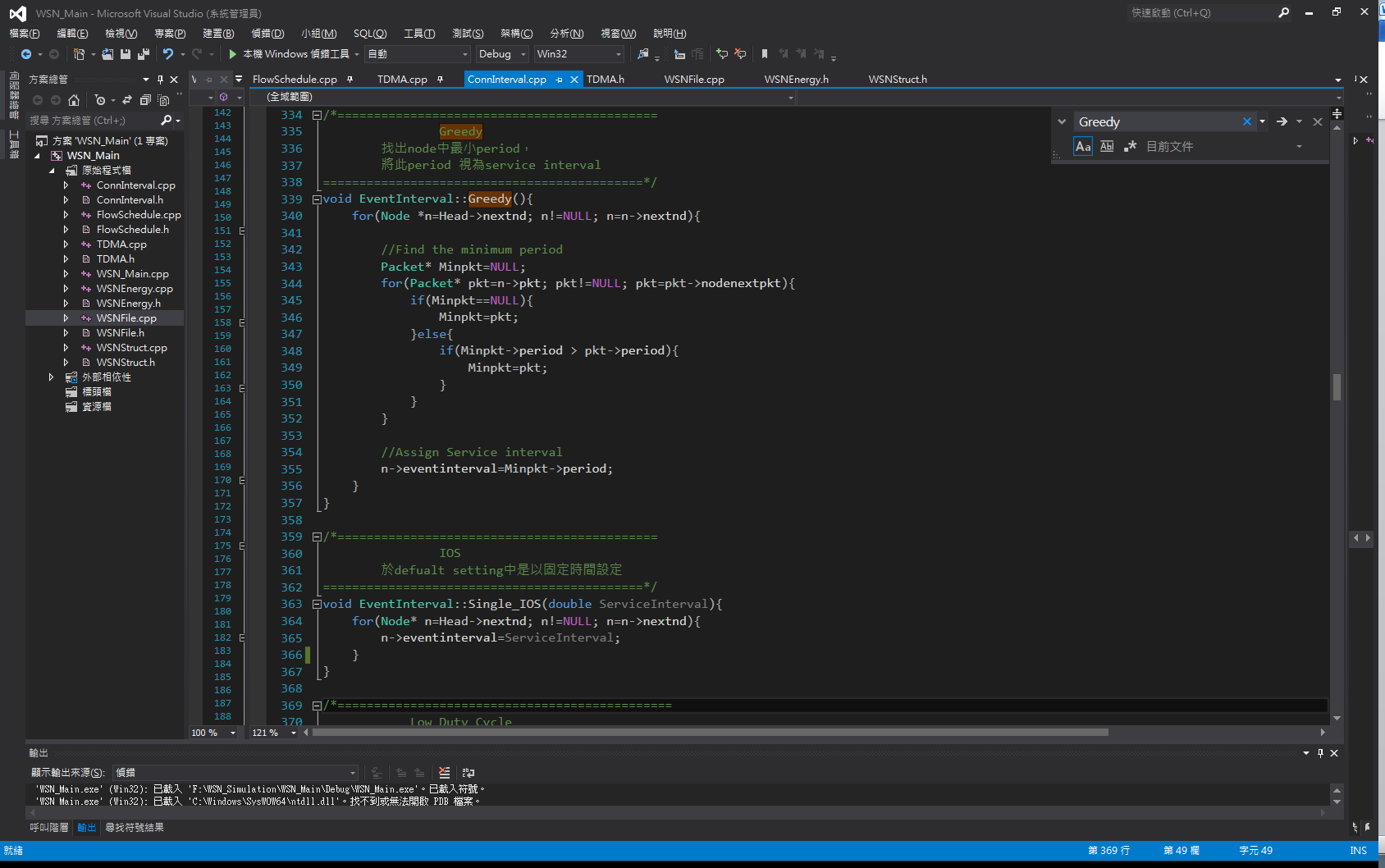
最大rate作各個

packet rate assignment

**🡪Greedy & IOS**

概念: Greedy為採用最小period作為service interval

概念: IOS為採用30ms



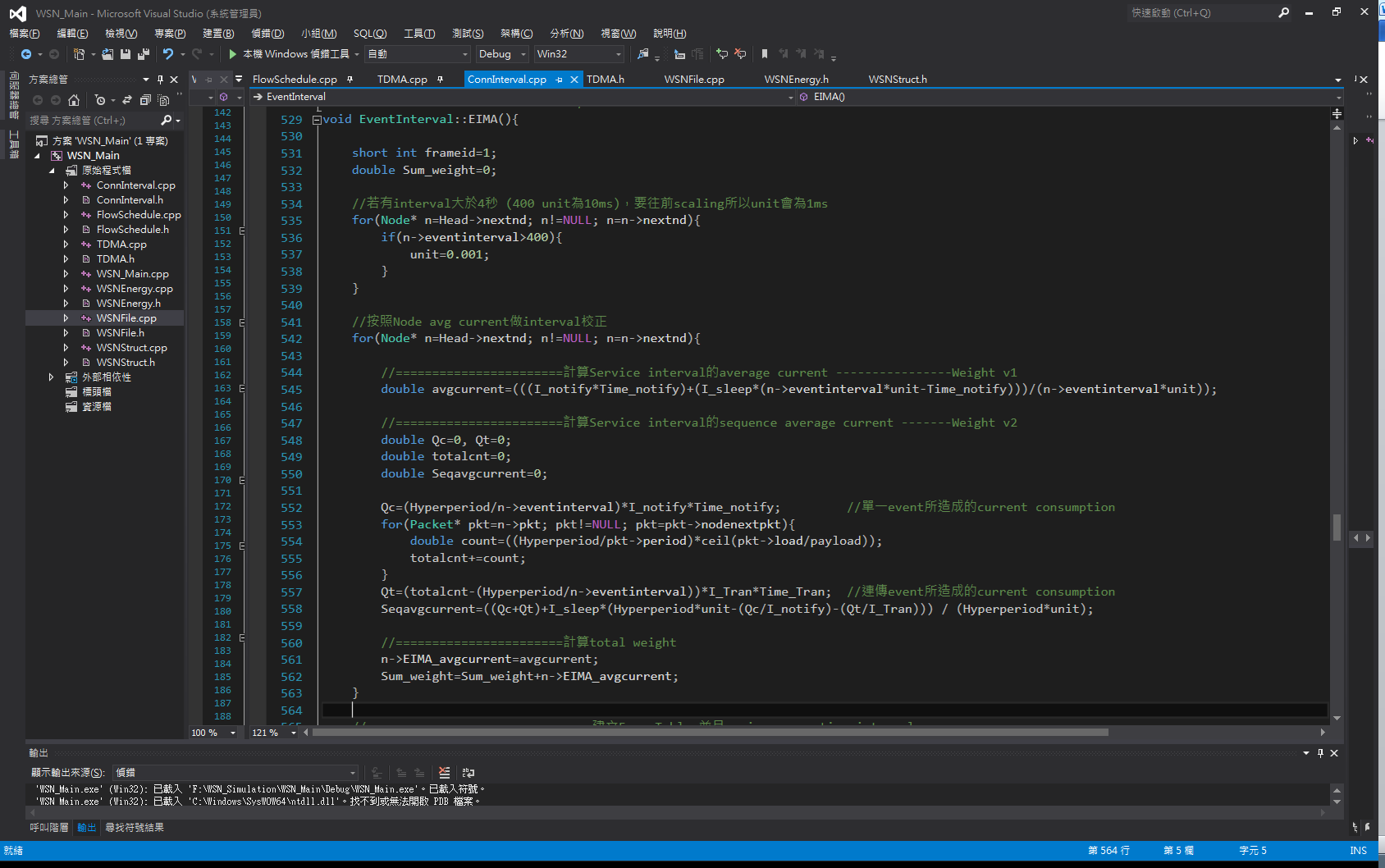
找最小period

Service Interval為30ms

●TDMA

●Connection Interval (Multiple node)

**🡺Energy Efficiency Interval Multiple Access (EIMA)**

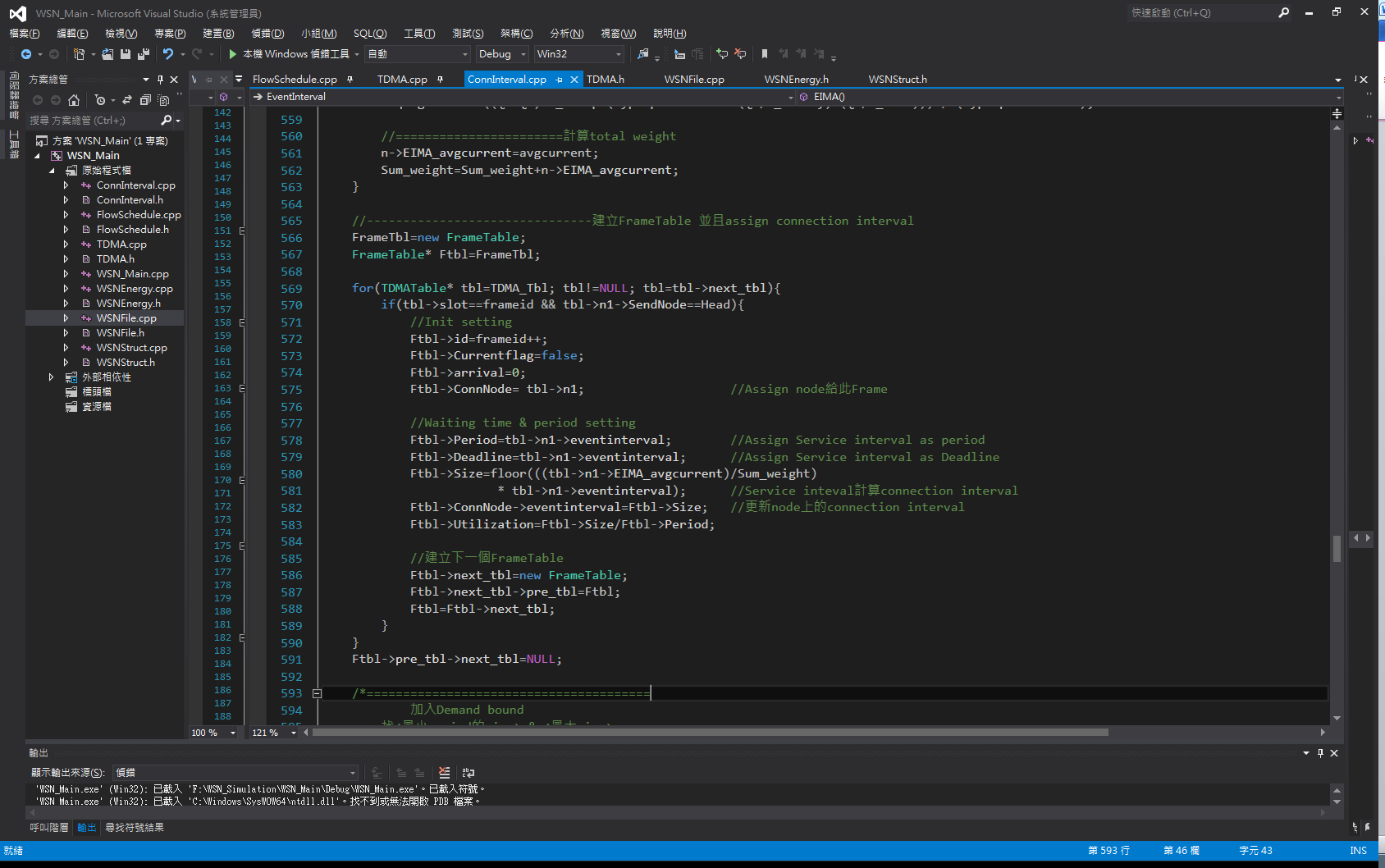


Assign connection interval

計算各個salve weight

Weight v1指的是不考量連傳性質(avgcurrent)

Weight v2指的是考量連傳性質(Seqavgcurrent)



FrameTable Setting

Period🡺Service Interval

Deadline🡺Service Interval

Size🡺Connection Interval

依照weight比例分配

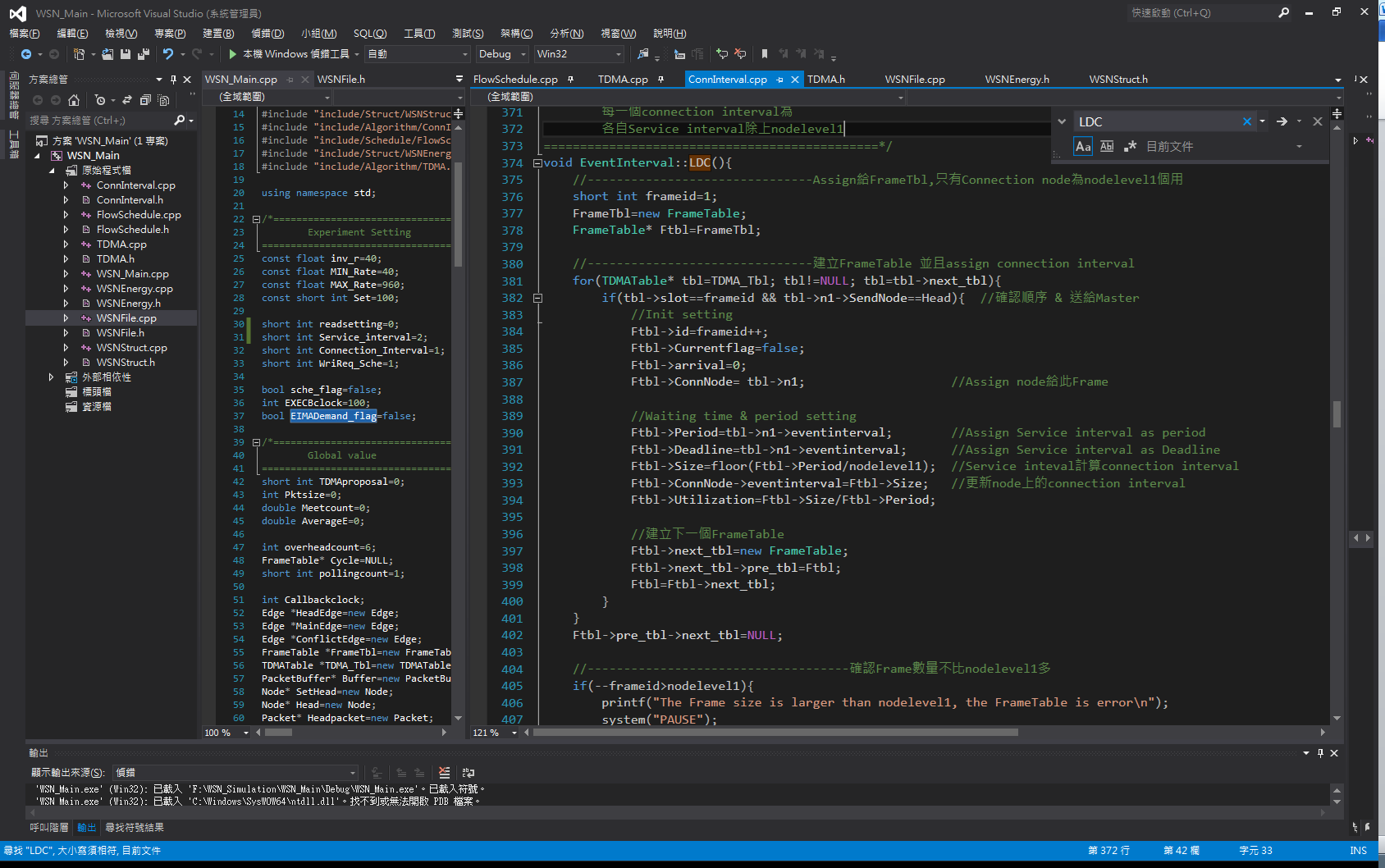
**(記得一定要在做ConnNode的connection interval修正)**

建立schedule的FrameTable

會依據TDMA slot的順序做assign

**🡺Low Duty Cycle (LDC)**

概念: 各個slave的connection interval依照原本service interval除上同level node數量，目的是要節省整體network耗能



FrameTable Setting

Period🡺Service Interval

Deadline🡺Service Interval

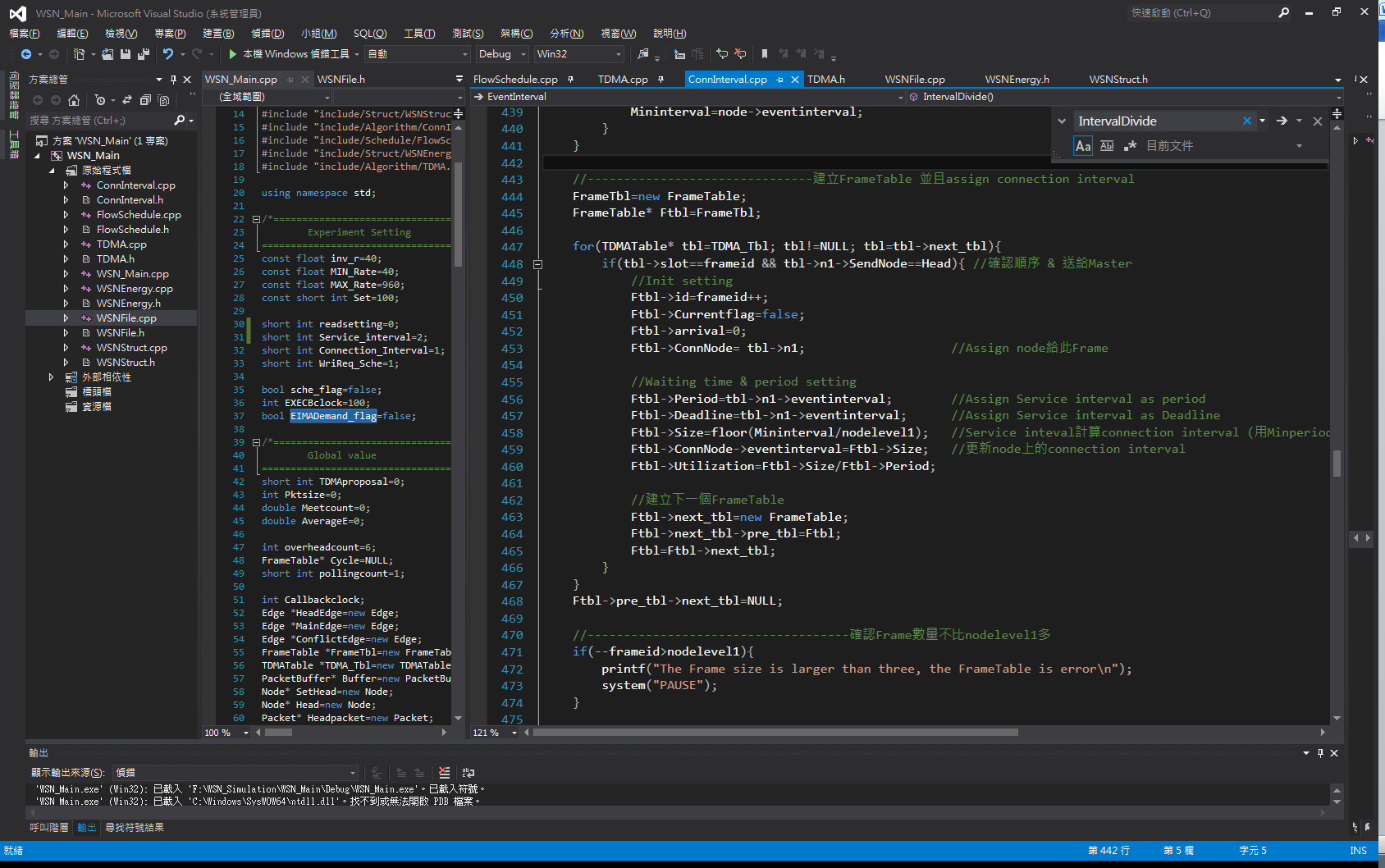
Size🡺Connection Interval

各個service interval除與同level node數量

**(記得一定要在做ConnNode的connection interval修正)**

**🡺IntervalDivide (Greedy)**

概念: 各個slave的connection interval依照最小service interval除上同level node數量，目的是要防止彼此間的block



FrameTable Setting

Period🡺Service Interval

Deadline🡺Service Interval

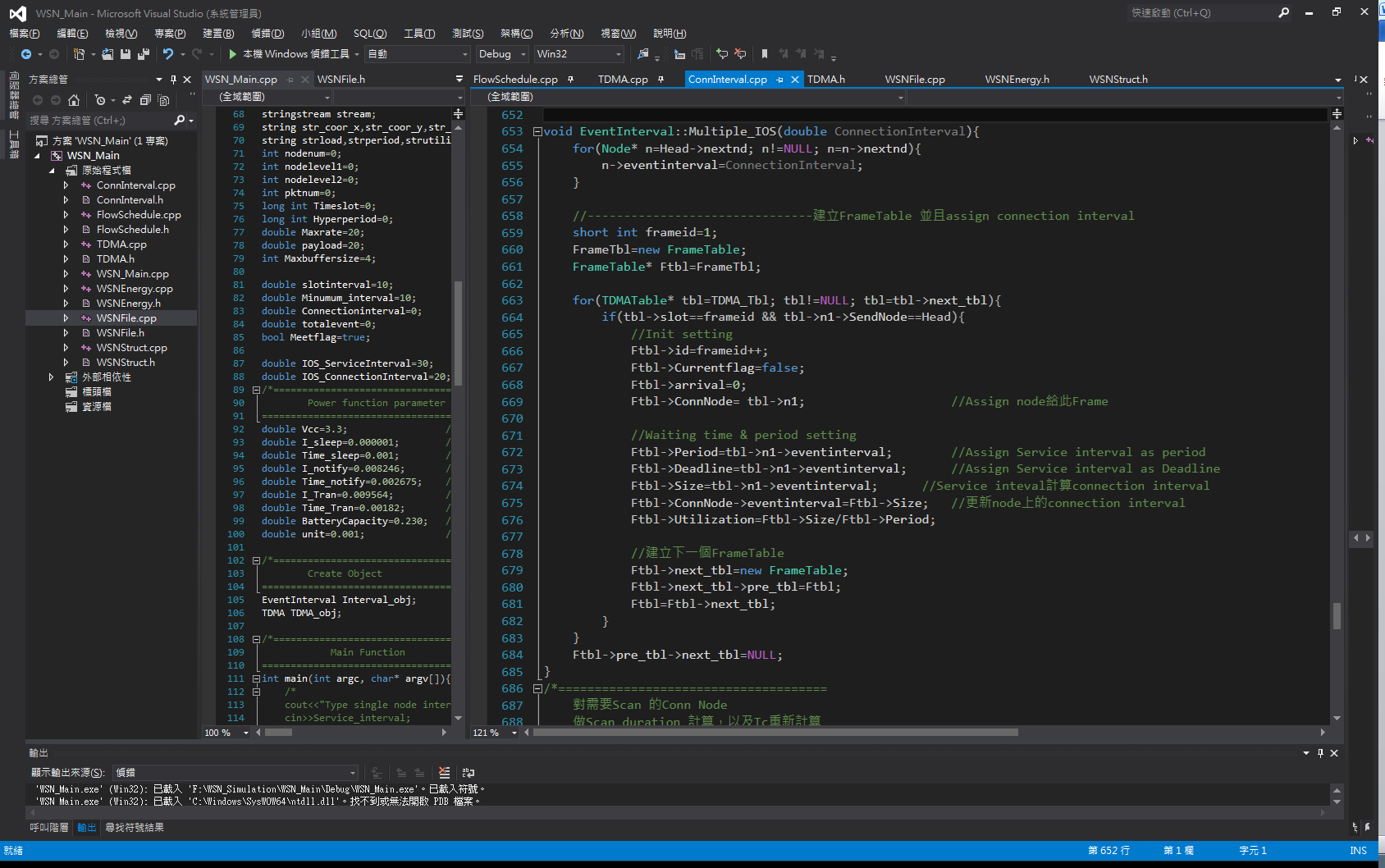
Size🡺Connection Interval

最小service interval除與同level node數量

**(記得一定要在做ConnNode的connection interval修正)**

**🡺Multiple\_IOS (IOS)**

概念: 各個slave的connection interval依照原本service interval除上同level node數量，目的是要節省整體network耗能



FrameTable Setting

Period🡺 Connection Interval

Deadline🡺 Connection Interval

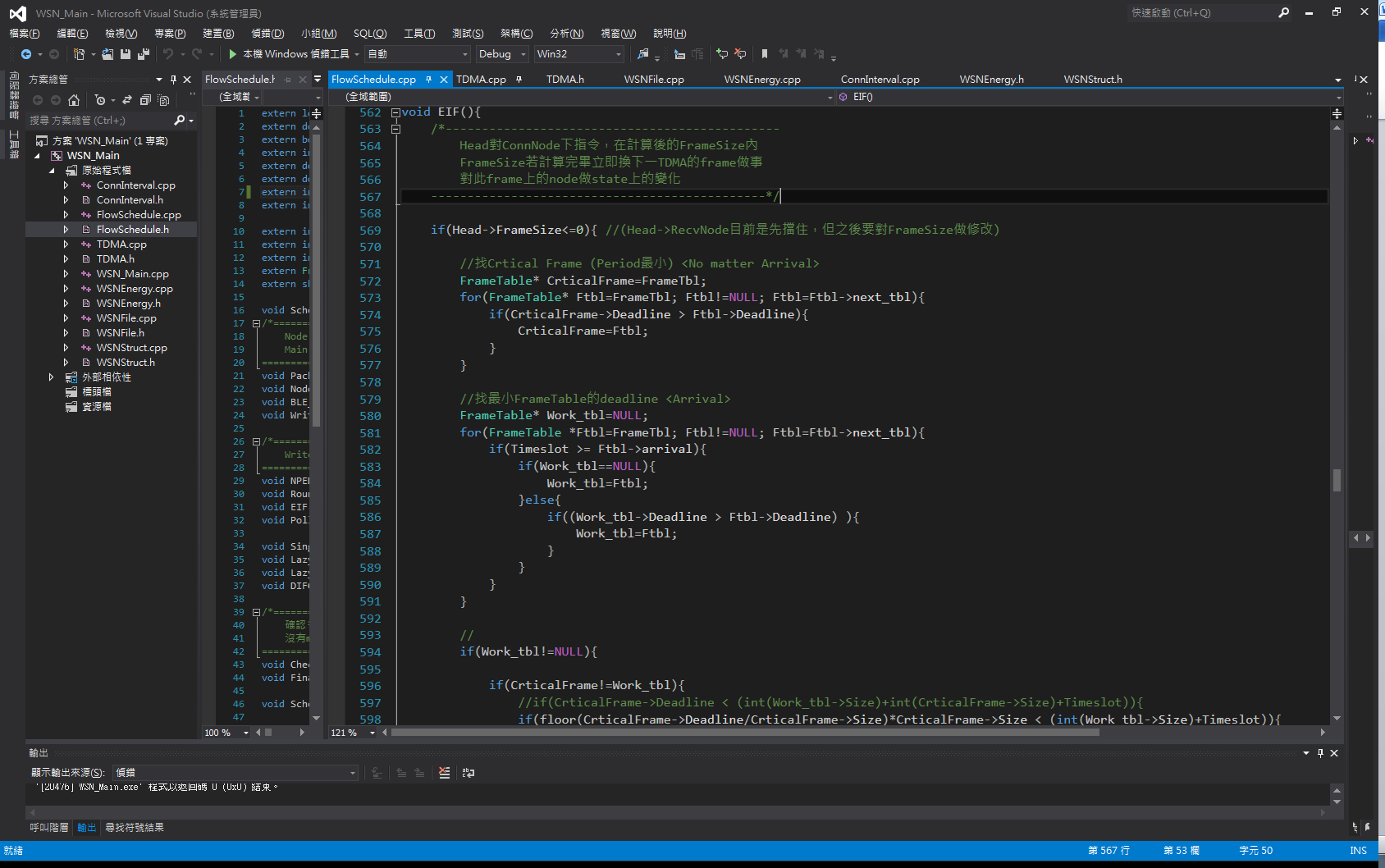
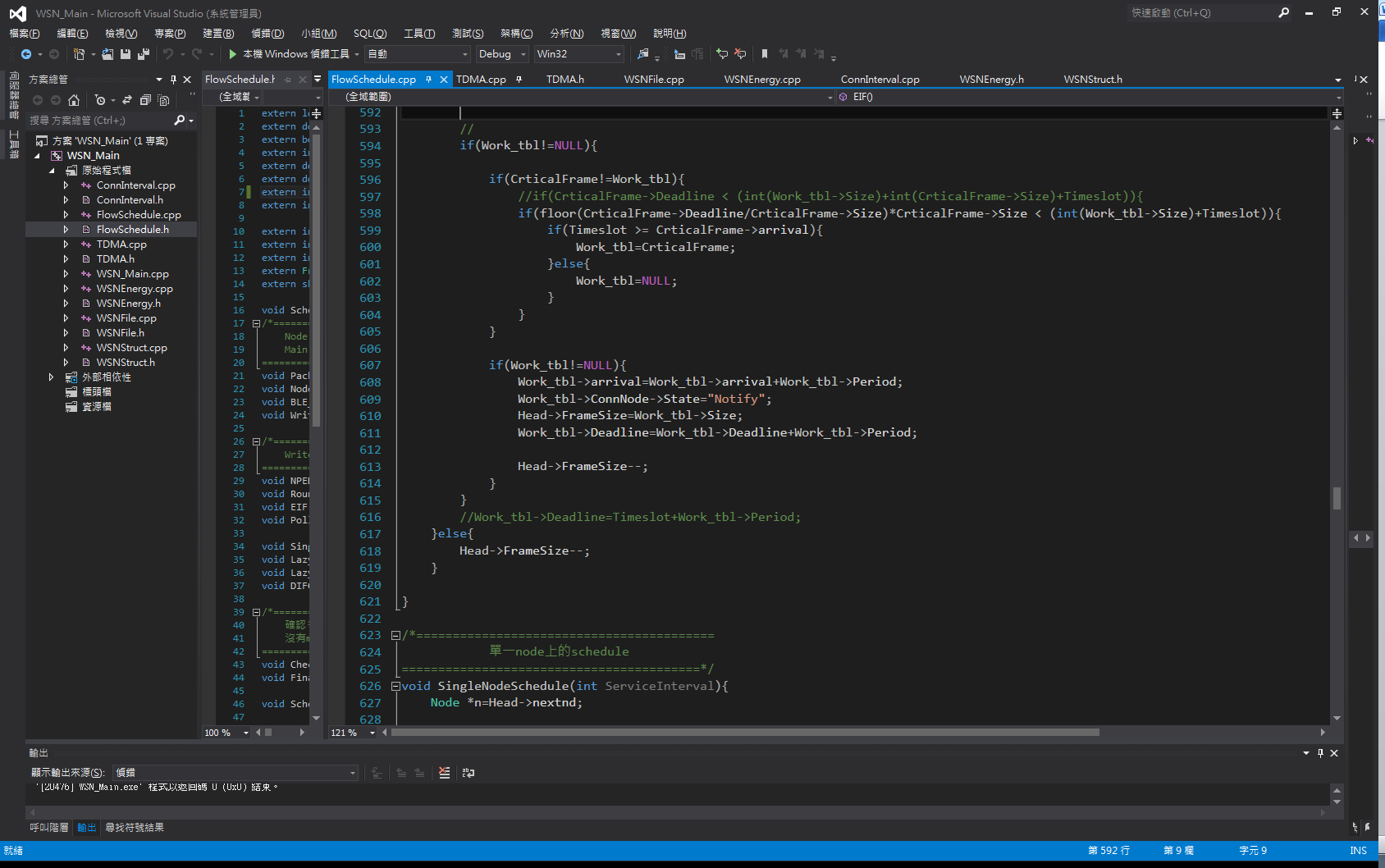
Size🡺Connection Interval

<IOS\_ConnectionInterval>

**(記得一定要在做ConnNode的connection interval修正)**

●Write-Request Scheduler

EIF



找最小deadline的Frame，不管是否arrival

CrticalFrame

做request

Node state會為Notify

做CrticalFrame和WorkTbl比較

找最小deadline的Frame

WorkTbl

Round Robin

Write\_Request

🡪依據前面的幾個schedule方法做request完之後，對node做傳輸判定

Energy計算

進行傳輸

Notify State

>將State換成是Transmission

>RecvNode切換node

(因event時間為2.67ms所以Timeslot要加)

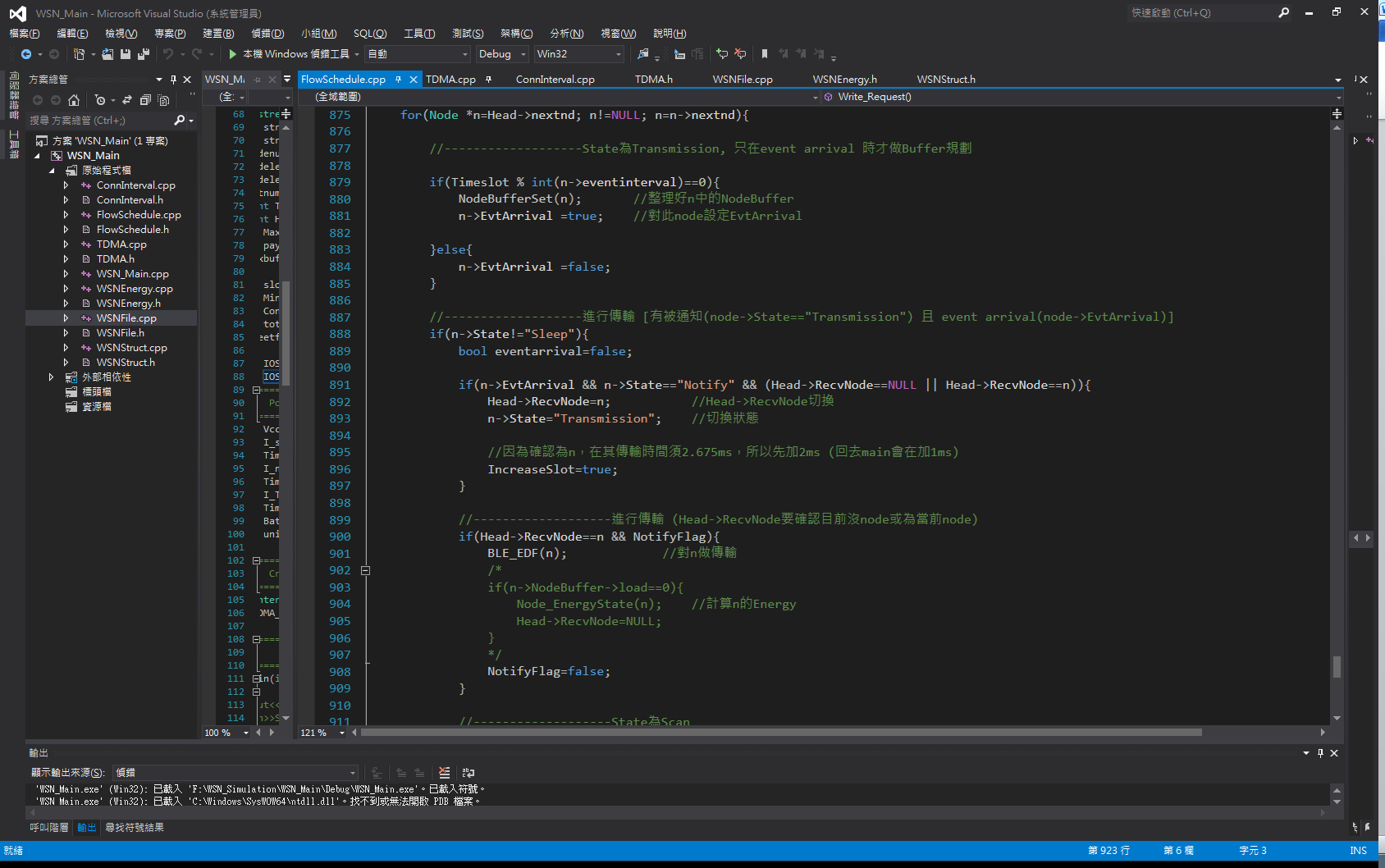
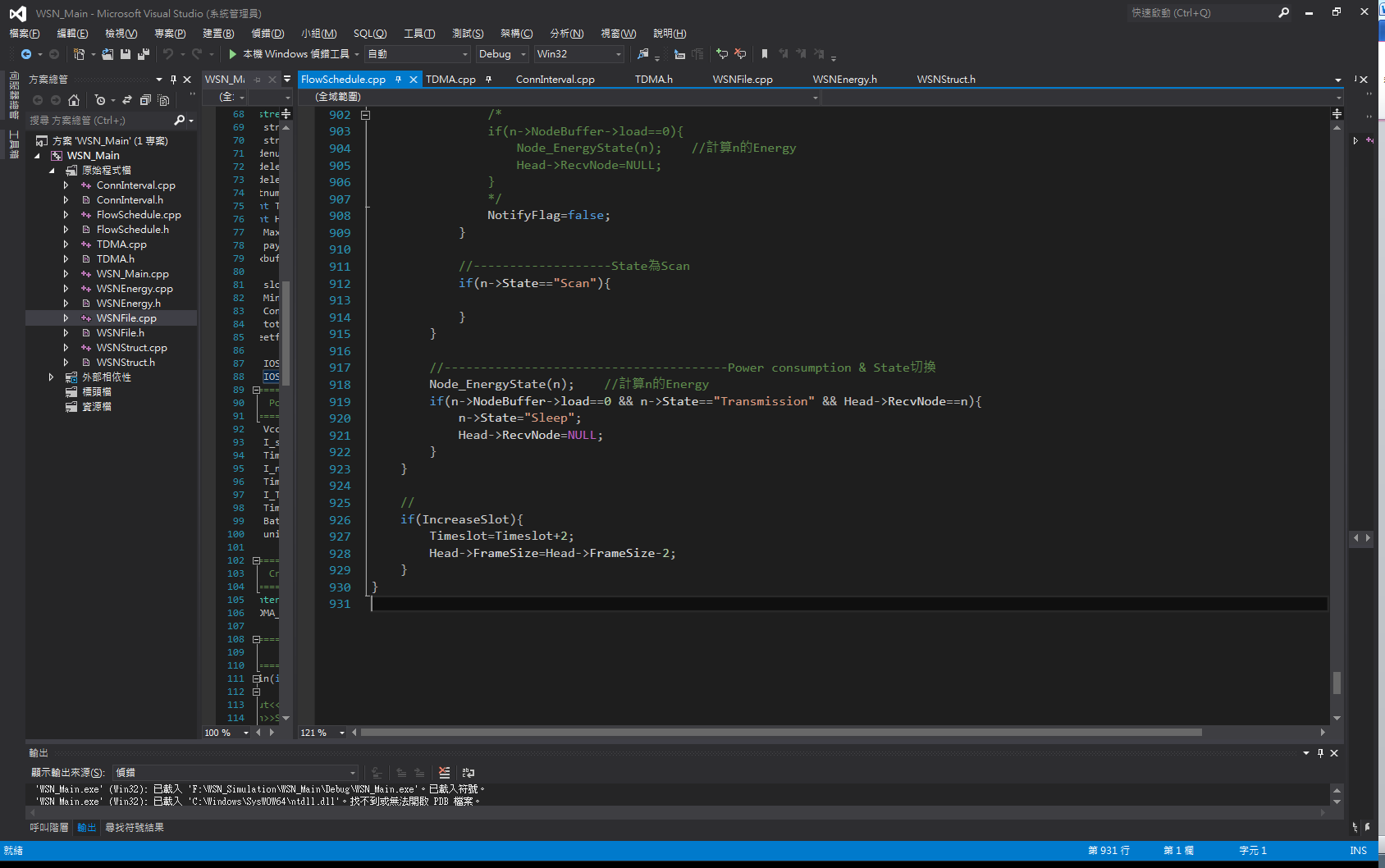
被通知或是連傳才會進入

(State為”Notify” or “Transmission”)

確認每一時間點下

Node是否會有connection event發生

(node的EvtArrival會trigger)



Write-Request中有三個function

NodeBufferSet(Node) 🡺放入現在時間點下，priority最高的packet於Buffer中 (**pktsize**, **load**, and **pkt**)

BLE\_EDF(Node) 🡺對buffer做傳輸

Node\_EnergyState(Node)🡺計算event耗電量

●Write Output File

●Draw plt

**實作**

●檔案區分

●硬體設定

●量測方法