Source Code:

<https://github.com/ntujoseph/homework>

1. **CSMA/CA的版本**

在相同L下, 當N愈大時，throughput愈低，因為發生碰撞的機率愈高。在相同N下，L愈大throughput愈高，L愈小throughput愈小，基於一個簡單概念，全部傳送資料量和整體花在Backoff的時間成本。

圖1為Y軸throughput為相同N={1~40}, L={1~10} 的條件下, 執行50回所得到平均throughput。當N>30時, 整體throughput幾乎接近0, 因為節點執行Backoff程序的次數將會很快達最大BE值(=7)而放棄傳送。

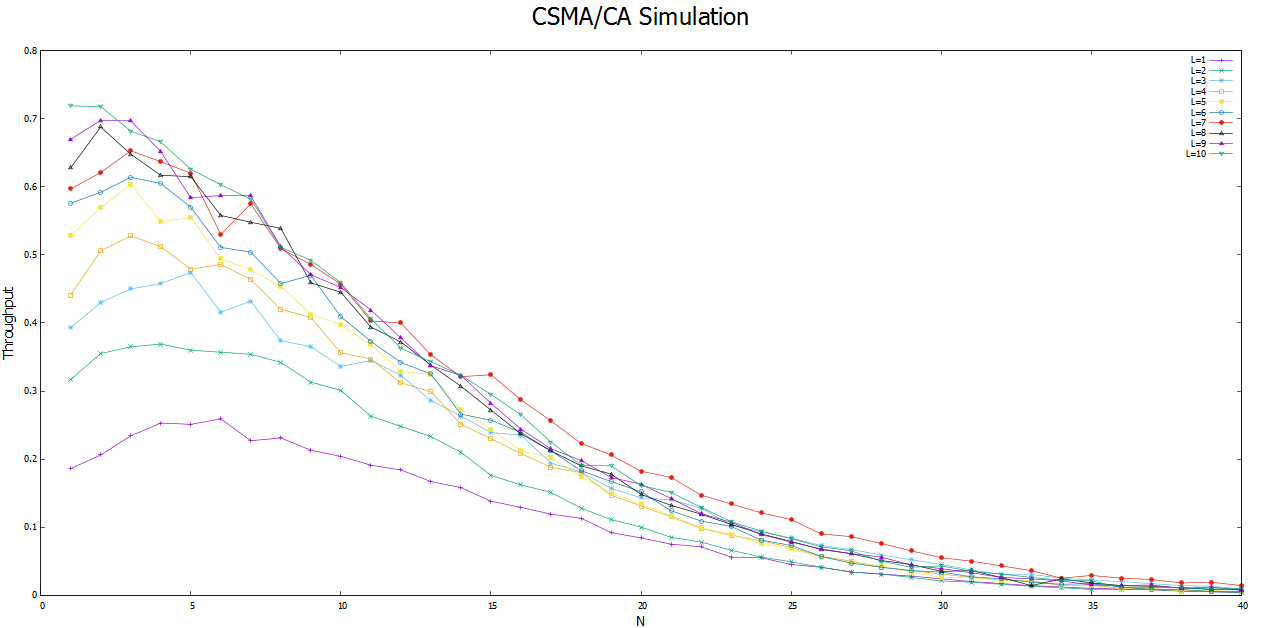


圖 Y軸為Throughput , X軸為N={1~40}, 顯示L={1~10}的結果

1. **CSMA/CA with Freeze的版本**

在Backoff期間,不減少BE值, 此對於當N很大時, 每一個節點執行Backoff程序的次數就不會那麼快到達最大BE值而放棄傳送。因此，和圖1相比，當N>30時，許多節點還不會放棄傳送，故throughput不會是0或接近0。

圖2為Y軸throughput為相同N={1,40}, L={1~10} 的條件下, 執行50回所得到平均throughput



圖: Y軸為Throughput , X軸為N={1~40}, 顯示L={1~10}的結果

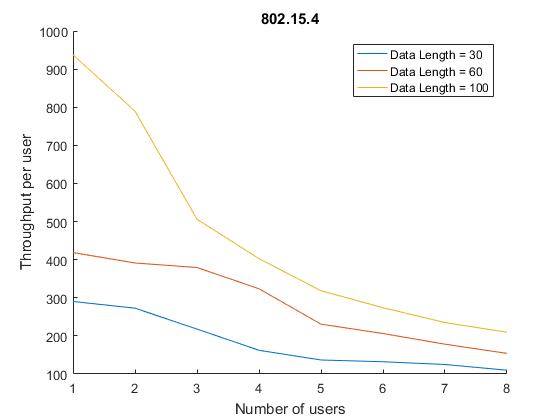
1. 圖3為FCM2401 Zigbee模組實機測試的結果，符合CSMA/CA模擬的狀態, 即在相同L下, 當N愈大時，throughput愈低。而在相同N下，L愈大throughput愈高，反之L愈小throughput愈小。

圖 3: Y軸為Throughput , X軸為N={1~8}, 顯示L={30,60,90}的結果 :

**Source Code:**

<https://github.com/ntujoseph/project_zigbee/tree/master/sample_code/throughput_test/RF_RX>

**main.c 主要調整參數**

#define ROLE 0 // this code is for 1: transmitter, 0: receiver

#define MY\_DEVICE\_ADDR 0x0007 //mainly for transmitter device address (0~7)

#define MY\_DEVICE\_ID MY\_DEVICE\_ADDR //should be (0~7)

#define TARGET\_ADDR 0x00F1 //receiver address

#define MAX\_PACKET\_COUNT 1000  //transmitter 會送1000個後就停止傳送

#define MAX\_RECV\_COUNT 400 //收到多少個封包後,就結算throughput

**1個收, 8個送**

* Receiver address: 0x00F1
* ID 1: Transmitter address: 0x0001
* ID 2: Transmitter address: 0x0002
* ID 3: Transmitter address: 0x0003
* ID 4: Transmitter address: 0x0004
* ID 5: Transmitter address: 0x0005
* ID 6: Transmitter address: 0x0006
* ID 7: Transmitter address: 0x0007
* ID 8: Transmitter address: 0x0008

**封包格式**



typedef struct \_packet

{

uint16\_t dev\_id \_\_attribute\_\_((packed)); //id=0,1,2,3,...,8,

uint16\_t seq\_no \_\_attribute\_\_((packed)); //sequence number

uint8\_t data[60] \_\_attribute\_\_((packed));

}Packet;

**記錄每一個人的throughput統計表**

typedef struct \_record {

uint16\_t dev\_id;

uint16\_t last\_seq\_no;

uint16\_t packet\_count;

uint16\_t t1; //the time of the first packet

uint16\_t t2; //the timestamp of being received packet

uint8\_t finish;

} Record;

當接收端收到MAX\_RECV\_COUNT的量時, 會輸出以下報表內容(以N=8為例):

# N=8, MAX\_RECV\_COUNT=100

ID COUNT SEQ T1 T2

8 100 141 65 85

ID COUNT SEQ T1 T2

1 100 183 65 93

ID COUNT SEQ T1 T2

2 100 264 65 100

ID COUNT SEQ T1 T2

3 100 215 70 100

ID COUNT SEQ T1 T2

7 100 250 70 105

ID COUNT SEQ T1 T2

6 100 285 69 113

ID COUNT SEQ T1 T2

5 100 335 67 115

ID COUNT SEQ T1 T2

4 100 325 67 117

**計算每個人的throughput**

throughput=(packet\_count\* sizof(Packet)/((T2-T1)\*100ms)  
其中packet\_count=MAX\_RECV\_COUNT(we set to 400)

**開發畫面**

