|  |
| --- |
|  |
| Formule utilizzate nel modello in excel |
| Spiegazione dei vari campi e delle formule usate |

|  |
| --- |
| Lorenzo Pagliari  10/03/2015 |

Sommario

[1. HP. 2.1 3](#_Toc414371030)

[1.1. Advertising Event 3](#_Toc414371031)

[1.1.1. ToT. A.E. Time [ms] 3](#_Toc414371032)

[1.2. Connection Event 4](#_Toc414371033)

[1.2.1. N° Data PDU needed 4](#_Toc414371034)

[1.2.2. Interval 1 4](#_Toc414371035)

[1.2.3. Interval 2 5](#_Toc414371036)

[1.2.4. Time dpdu + ack 6](#_Toc414371037)

[1.2.5. N. dpdu per Interval 6](#_Toc414371038)

[1.2.6. Connection Windows needed 6](#_Toc414371039)

[1.2.7. Total C.E.s Time 8](#_Toc414371040)

[1.3. Bound of Total Tx Time 9](#_Toc414371041)

[1.4. Energy Consumption 10](#_Toc414371042)

[2. HP.2.2 11](#_Toc414371043)

# HP. 2.1

Consideriamo una situazione in cui si ha una serie di utenti disposti a "catena", quindi ognuno è in "range" di altre due persone. In media ogni dispositivo esegue (2n-2)/n transazioni attive e passive. In questa ipotesi non viene usato nessun tipo di controllo. Di seguito vi è uno schema che rappresenta la struttura della rete di dispositivi per questa ipotesi.

≈

≈

Trasmissione attiva

Trasmissione passiva

## Advertising Event

### ToT. A.E. Time [ms]

Il totale del tempo impiegato per completare un Advertising Event , in cui vi è subito un Initializer pronto a connettersi, è la somma del tempo di Advertising, più il tempo di interframe tra pacchetti più il tempo di trasmissione del pacchetto di connection request. Il tempo di advertising cambia a seconda se si usa il metodo indiretto o il metodo diretto. Nel foglio di calcolo Il ToT A.E. Time viene calcolato usando un sistema di maschere in modo da utilizzare il tempo corretto, coerente con la maschera scelta. Le maschere possibili sono solo due: 10 (per utilizzare il metodo diretto) e 01 (per utilizzare il metodo indiretto).

## Connection Event

### N° Data PDU needed

Il numero di data PDU needed viene semplicemente calcolato dividendo la grandezza del messaggio (in Byte) per la grandezza massima del payload di un singolo Data PDU, che da specifiche è di 27B:

“*The Length field of the Header indicates the length of the Payload and MIC if included. The length field has the range of 0 to 31 octets. The Payload field shall be less than or equal to 27 octets in length. The MIC is 4 octets in length.*” *(pag. 2209).*

### Interval 1

Con ***Interval 1*** ci si riferisce alla finestra di Connection Window in cui viene fatta anche la chiusura dello scambio dati (*fig.1*). La durata di Interval1 è minore o uguale a connInterval.

*Figura 1.1: Interval 1*

S -> M Tx

pool

≈

ack

T\_IFS

T\_IFS

ConnInterval

LL\_TERMINATE

Left of Time Interval 1

M -> S Tx

#### Setup on Interval 1

Con ***Setup on Interval 2*** ci si riferisce a quell’intervallo di tempo, comprensivo di più intervalli di tempo separati, in cui si scambiano dati necessari per inizializzare la finestra (stabilire il nuovo anchor point) e alla fine per terminare la connessione (*fig.2*).

*Figura 1.2: Setup on Interval 1*

pool

≈

ack

T\_IFS

T\_IFS

ConnInterval

LL\_TERMINATE

S -> M Tx

M -> S Tx

#### Left On Interval 1

Con ***Left on Interval 1*** ci si riferisce all’intervallo di tempo rimanente dall’Intervallo1 tolto il tempo di setup; ovvero è il tempo rimanente per ogni Intervallo1 in cui è possibile inviare dati.

### Interval 2

Con ***Interval 1*** ci si riferisce alla finestra di Connection Window in cui viene fatto solo setup di apertura connessione e scambio dati (*fig.1*). La durata di Interval1 è uguale a connInterval in quanto i dati da trasmettere richiedono tutta la finestra; se non fosse così allora sarebbe un Interval1. In *fig.3* si può vedere la struttura della finestra.

*Figura 1.3: Interval 2*

pool

T\_IFS

T\_IFS

ConnInterval

Left of Time Interval 2

M -> S Tx

#### Setup on Interval 2

Con ***Setup on Interval 2*** ci si riferisce a quell’intervallo di tempo, comprensivo di più intervalli di tempo separati, in cui si scambiano dati necessari per inizializzare la finestra (stabilire il nuovo anchor point) e ci si assicura di chiudere il Connection Event almeo T\_IFS prima delsuccessivo (*fig.4*).

*Figura 1.4: Setup on Interval 2*

pool

T\_IFS

T\_IFS

ConnInterval

M -> S Tx

#### Left On Interval 2

Con ***Left on Interval 2*** ci si riferisce all’intervallo di tempo rimanente dall’Intervallo2 tolto il tempo di setup; ovvero è il tempo rimanente per ogni Intervallo2 in cui è possibile inviare dati.

### Time dpdu + ack

Questo valore rappresenta il tempo necessario per effettuare un completo invio di un pacchetto dati alla massima grandezza, più il suo ack di ritorno, più i tempi T\_IFS tra i pacchetti (*fig.5)*. Ci si riferirà a questo lasso di tempo come “*Data pattern”*.

*Figura 1.5: time data pdu + ack*

ack

S -> M Tx

M -> S Tx

T\_IFS

full data packet

Time dpdu + ack

### N. dpdu per Interval

Rappresenta il numero di “*Data pattern”* (Time dpdu + ack) che è intermante possibile contenere all’interno di intervallo specificato. Serve per calcolare quanti DATA PDU di massima grandezza è possibile inserire in un determinato intervallo, compresi i loro acknowledge. Includiamo il tempo di ack nel calcolo perché ipotizziamo chel’ack di ogni pacchetto arrivi nella stessa finestra in cui il rispettivo data pdu è stato inviato; se così non fosse possibile il data pdu verrà rimandato alla finestra successiva.

### Connection Windows needed

Ora che sappiamo quanto tempo necessita un DATA PDU completo per essere inviato e confermato, quello che ci serve sapere è quante connection windows ci servono per inviare tutto il messaggio. La formula excel che ho implementato cerca di capire quante connection windows di tipo Interval 2 sono necessarie più un’ultima connection window di tipo Interval 1 per completare e chiudere la trasmissione. Il problema principale è che una finestra di tipo Interval 1 al suo massimo non può contenere tanti *Data pattern* quanto una tipo 2, quindi quello che bisogna fare è dividere il messaggio per la massima capienza di Interval 2 e poi il resto capire se basta un Interval 1 a chiudere oppure è necessario un Interval 2 supplementare.

La formula è stata rappresentata in fig.6 tramite pseudo-codice, per una migliore comprensione.

*Figura 1.6: pseudo-codice per algoritmo che esegue il calcolo "connection windows needed"*

***If*** *( n°data pdu needed ≤ n°dpdu per Interval1){*

*/\*basta solo la connection window di tipo 1 \*/*

*num\_conn\_wind\_needed = 1;*

*}****else if*** *((n°data pdu needed* ***MOD*** *n°dpdu per Interval2) > 0 &&  
 (n°data pdu needed* ***MOD*** *n°dpdu per Interval2) ≤ n°dpdu per Interval1  
 ){*

*/\*quello che resta dalla divisione dpdu needed/dpduInterval 2  
\* è maggiore di zero e minore o uguale al numero di dpdu che Interval1  
\* può contenere significa che il restante entra in una finestra di tipo 1 \*/*

*num\_conn\_wind\_needed = ;*

*}****else****{*

*/\* Il rimanente eccede la capienza di una finestra di tipo 1, quindi  
 \* occorre un’ulteriore finestra di tipo 2 e poi una finestra di tipo 1 \*/*

*num\_conn\_wind\_needed = + 1;*

*}*

***If*** *( n°data pdu needed ≤ n°dpdu per Interval1){*

*/\*basta solo la connection window di tipo 1 \*/*

*num\_conn\_wind\_needed = 1;*

*}****else if*** *((n°data pdu needed* ***MOD*** *n°dpdu per Interval2) > 0 &&  
 (n°data pdu needed* ***MOD*** *n°dpdu per Interval2) ≤ n°dpdu per Interval1  
 ){*

*/\*quello che resta dalla divisione dpdu needed/dpduInterval 2  
\* è maggiore di zero e minore o uguale al numero di dpdu che Interval1  
\* può contenere significa che il restante entra in una finestra di tipo 1 \*/*

*num\_conn\_wind\_needed = ;*

*}****else****{*

*/\* Il remanente eccede la capienza di una finestra di tipo 1, quindi  
 \* occore un’ulteriore finestra di tipo 2 e poi una finestra di tipo 1 \*/*

*num\_conn\_wind\_needed = + 1;*

*}*

### Total C.E.s Time

Quello che resta da calcolare è il tempo totale di tutti i Connection Events richiesti per trasmettere il messaggio di grandezza specificata in “***msg. size [B]***”. Per far ciò dobbiamo sommare due contributi:

1. Il temo necessario per completare tutte i Connection Events eseguiti con connection window di tipo 1, quindi tutte quelle finestre che hanno inviato DATA PDU di grandezza massima, alla capienza massima della finestra.
2. Il tempo per trasmettere “il residuo” del messaggio che deve essere inviato con la finestra di tipo 2.

Purtroppo quello che non sappiamo ancora è quanto realmente è grande questa parte restante di messaggio e quindi quanto tempo realmente richiede; inoltre della restante parte del messaggio si dovrà capire quanti full data pdu occorrano e quanto grande, ma non full, sarà l’ultimo pacchetto. Successivamente verranno esposte le operazioni per calcolare ciò.

Questa formula calcola quanti sono i byte presenti null’ultima finestra di trasmissione.

Questa formula calcola quanti full data pdu ci sono nei byte restanti nell’ultima finestra.

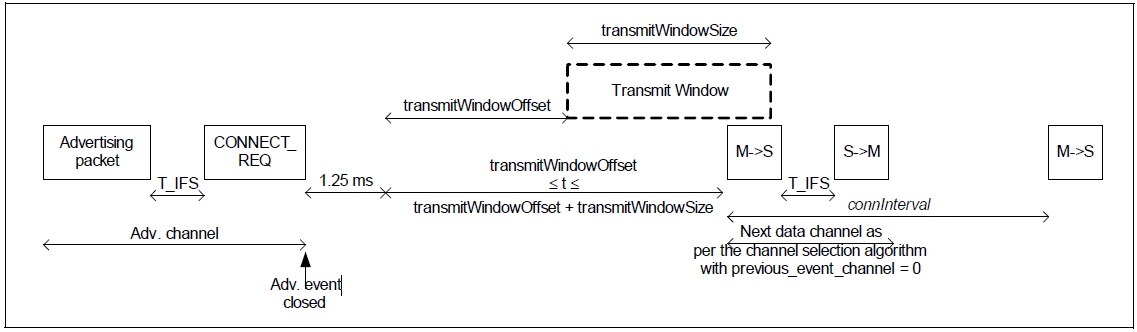
Questa formula calcola quanti byte sono rimasti nell’ultimo pdu.

Ora che abbiamo tutti i dati possiamo calcolare il tempo totale.

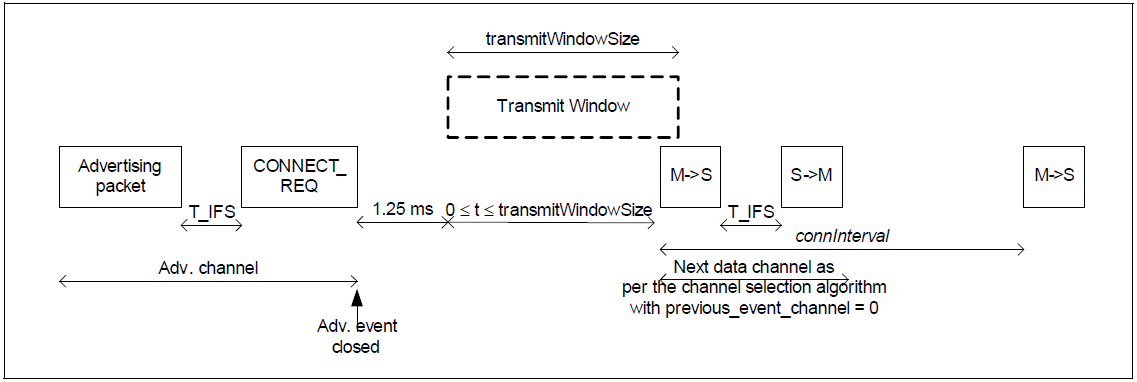
Nel *lastWindowTxTime* sono stati tenuti in considerazione i T\_IFS e i T\_ack per la corretta trasmissione di ogni data pdu, full o no, similarmente al calcolo del *Data pattern.*

## Bound of Total Tx Time

Nella Buond Analysis fatta per quest’ipotesi, non sono stati tenuti in considerazione controlli di alcun tipo e errori di trasmissione; una volta che i dispositivi si sono connessi, lo scambio dati è avvenuto regolarmente. I buond sono caratterizzati dal fatto che il master può iniziare la trasmissione in una finestra. In fig.7 e in fig.8 si possono osservare le due casistiche in questione.



*Figura 1.7: master trasmission con windowOffset non nullo*



*Figura 1.8: master transmission con windowOffset nullo*

## Energy Consumption

Il consumo di energia medio viene stimato calcolando un min e un max, rispettivamente utilizzando il LowerBound e l’UpperBound dei tempi di trasmissione e poi viene fatta la media tra i due.

Il ***TxEnConAVG*** è l’energia media richiesta per trasmettere o ricevere ed è mediamente ≈15mW.

Quindi il ***avgEC*** cresce in maniera lineare rispetto al numero di Transazioni, intese come somma di Transazioni attive più Transazioni passive ed esse sono rispettivamente linearmente dipendenti dal numero di utenti /dispositivi mobile che costituiscono la popolazioni in analisi.

Di conseguenza possiamo dire che il consumo medio di energia cresce con legge **O(n)** rispetto al numero di utenti/dispositivi mobile.

# HP.2.2

Consideriamo una situazione in cui si ha una serie di utenti disposti a "catena", quindi ognuno è in "range" di altre due persone. In media ogni dispositivo esegue (2n-2)/n transazioni attive e passive. In questa ipotesi non viene usato nessun tipo di controllo. Di seguito vi è uno schema che rappresenta la struttura della rete di dispositivi per questa ipotesi.

≈

≈

Trasmissione attiva

Trasmissione passiva

Ogni dispositivo mobile subisce una trasmissione passiva quando riceve il messaggio la prima volta, e poi esegue trasmissioni attive verso quei dispositivi che richiederanno poi di ricevere il messaggio. Quindi in una situazione a catena come quella sopra nello schema, si avrà che le “gli archi di ritorno” saranno solo comunicazioni di advertising alle quali non vi sarà richiesta di connessione, quindi ai fini del calcolo del consumo energetico, si possono trascurare.