

MSA - 16.03.2021

Riprendiamo da dove ci siamo fermati la volta scorsa. Abbiamo illustrato in qualche dettaglio l'architettura statica del protocollo che stiamo considerando, 802.11. Quindi in particolare abbiamo visto l'organizzazione dei pacchetti che circolano sul canale wireless gestito in accordo da questo protocollo. Esistono tre categorie di pacchetti: di gestione, dati e di controllo. Abbiamo discusso come è organizzato il frame dati, rimaneva da fare una piccola coda prima di parlare alla parte dinamica del protocollo. Alcuni frame di controllo che possono circolare sul canale sono i frame di ack, rts o cts, e da questo schema vediamo come, se ricordate la struttura generale che può avere il frame che abbiamo discusso la volta scorsa, in questi frame di controllo si utilizza un sottoinsieme ristretto di tutti i campi opzionali. Sono sempre presenti i primi quattro byte, frame control che definisce la strutturazione di ciò che segue ed il campo duration, poi la parte di correzione di eventuali errori presenti nella parte che precede e poi i cambi indirizzi, in alcuni casi come ACK, dei quattro possibili nel caso di ACK e CTS se ne utilizza uno solo. Qui possiamo toccare con mano l'overhead causato dai frame RTS e CTS rispetto alla trasmissione del pacchetto vero e proprio. Se la trasmissione dei dati supera una certa soglia, l'overhead diventa trascurabile. Se invece non è abbastanza grande, l'overhead può essere significativo. Passerei direttamente alla parte dinamica, quella di gestione dell'accesso del canale. Come abbiamo detto, è un protocollo a contesa, ed il protocollo è basato sullo schema CSMA/CA. CD non è applicabile, questa è l'unica opzione. Dei due meccanismi di questa modalità di gestire l'accesso, il primo di Carrier Sensing, si tenta di trasmettere solo se il canale non è occupato. L'ascolto dello stato del canale avviene in due modi, che servono a tentare di limitare i problemi creati dalla caratteristica intrinseca di un canale wireless che il segnale si attenua con la distanza, dando vita ai problemi di cui abbiamo parlato. La condizione di canale occupato può avvenire in due modi: fisico, se io rilevo la presenza di un segnale sul canale wireless su cui sono sintonizzato, questo è indicativo di occupazione. Altrimenti c'è un ascolto virtuale. Anche se non ascolto nulla, se sono stato informato che c'è una trasmissione in corso, ho memorizzato questa informazione al mio interno e considero il canale occupato fin quando questa indicazione continua ad essere valida. Questa informazione è basata sul campo duration dei pacchetti. C'è scritto proprio per quanto tempo il canale risulterà occupato. Se un nodo viene a conoscenza, riceve un pacchetto che contiene questo campo duration, usa questa informazione per incrementare un suo contatore interno (**NAV**), e anche se sento il canale libero, fin quando questo contatore non arriva al valore 0 il canale viene considerato occupato. La stazione omette di trasmettere.

Poi c'è la parte di Collision avoidance, che entra in gioco quando il canale è percepito libero. Quest'oschema prevede che una stazione non tenta subito di occupare il canale ma aspetta ancora un po'. Se il canale risulta libero anche dopo l'attesa, la stazione inizia a trasmettere. L'attesa è basata da una parte fissa e da una parte variabile. La parte fissa (DIFS, PIFS e SIFS) ha una durata che può variare. A questa parte fissa si può sommare la parte variabile. Immaginiamo di essere in questo scenario: una stazione percepisce il canale libero, sia in senso fisico che virtuale. Una stazione rimane bloccata per un tempo pari all'intervallo DIFS. Trascorso questo tempo possono succedere due cose. O nessuno ha occupato

l'intervallo prima del termine del DIFS, se siamo in questa situazione la stazione inizia a trasmettere. Oppure può succedere che mentre la stazione è in attesa che trascorra l'intervallo, qualcun altro occupi il canale. Se questo succede la stazione si astiene dall'occuparlo, aspetta che il canale si liberi, e la stazione aspetta un intervallo DIFS, ed a questo intervallo sommerà un altro intervallo di durata variabile (back-off). Questo perché se nel tentativo di accesso precedente qualcuno ha occupato il canale prima che scorra DIFS in quanto altri potrebbero aver sperimentato questa situazione, e ci sarebbe una probabilità molto elevata di collidere al termine del singolo intervallo DIFS. Vincerà la competizione chi ha estratto il valore minore tra tutti quelli sorteggiati dai vari candidati. Chi risulta sfortunato, congelerà il valore del suo timer, che non scorre per il tempo di occupazione, e riprenderà il conteggio al ritorno del canale in condizioni di non occupazione.

Una volta ottenuto il diritto di trasmettere, lo scambio dati è affidabile. Si inviano dati e si aspetta un ACK. Il protocollo garantisce che lo scambio dati+ack sia atomico, non ci si possa interporre nessuno. Questo si garantisce con l'intervallo SIFS. Quando il mittente ha inviato i suoi dati, se il destinatario li ha ricevuti non aspetta un tempo DIFS ma un tempo SIFS, in modo che nessun altro possa intromettersi nella comunicazione. Il campo duration che è presente nel pacchetto contiene al suo interno l'informazione sul tempo necessario a trasmettere il pacchetto più l'ack successivo.

Giusto per rimarcare il concetto della fairness. Inizio dei tempi, durante il primo intervallo ci sono soltanto la stazione 1 e 3 che hanno qualcosa da trasmettere. All'istante 0 abbiamo solo la stazione 3, che percepisce canale libero ed inizia ad aspettare il tempo DIFS, continuando a rilevare la presenza di attività sul canale. Nel frattempo la stazione 1 si affaccia sul canale, anche la stazione 1 inizierà a far scorrere il suo timer. Stazione tre continua a sentire il canale libero e lo occupa. Altre stazioni hanno ascoltato il canale occupato quindi aspettano che si liberi. Avendo percepito il canale occupato, una volta che si è liberato, dovranno aspettare un tempo DIFS a cui si sommerà un tempo casuale che estraggono. In questa simulazione, la stazione fortunata è la 2 avendo estratto il valore più piccolo. Questo meccanismo non garantisce l'ordine di arrivo. La stazione due occupa il canale, e le altre due congelano il loro contatore. In verde c'è il valore residuo rimasto prima di arrivare a 0. Il valore residuo viene conservato perché quando termina la trasmissione della stazione 2, tutte quelle che desideravano trasmettere, una o più aggiunte nel frattempo, tutte quante avendo percepito canale occupato aspettano un tempo DIFS più un tempo casuale, che nel caso della stazione arrivata ultima sarà estratto casualmente, mentre per le altre sarà il valore congelato in precedenza. Questo è lo schema di fairness. Cosa succederebbe se chi era in attesa di trasmettere riestraesse di nuovo dall'urna un valore casuale? Si perderebbe completamente la memoria della situazione passata. Se una stazione continua ad essere sfortunata, la sua attesa rischia di diventare infinita. Situazione improbabile ma non impossibile è che l'ultimo arrivato estragga un numero pari al residuo precedente di una stazione. In questo caso scaduto questo tempo di attesa che è uguale, tutte e due iniziano a trasmettere. I pacchetti vanno in collisione, si corrompono, e nessuno dei due viene trasmesso. In casi fortunati può essere che grazie ai codici correttori almeno uno dei due riesce ad essere ricevuto correttamente. Al ciclo successivo, se hanno perso la

contesa tutti e due il discorso del timer residuo non si applica e devono estrarre un nuovo valore. La stazione 1 rimasta in attesa continua ad usare il suo valore residuo.

Nel caso RTS e CTS le cose vanno in questo modo. Di nuovo il preambolo in maniera analoga allo scambio dati più ack, anche il preambolo deve essere trasmesso in modalità atomica rispetto allo scambio successivo. Notate come queste parti verdi vogliono rappresentare l'informazione di durata veicolata dai pacchetti. Chi ascolta RTS riceve l'informazione sull'occupazione del canale per un tempo preciso, diverso da CTS e DATA. In momenti diversi, chi ascolta uno di questi pacchetti viene informato di quanto tempo ancora il canale sarà occupato, anche se non si percepisce l'occupazione. Menzionavamo la volta scorsa che è possibile che un singolo pacchetto dati se abbastanza grande venga frammentato se il canale risulta eccessivamente disturbato, quindi con un tasso di errore significativo. In questo caso la comunicazione si tenta di garantire l'atomicità nella trasmissione di tutti i frammenti. Se un frammento arriva a destinazione, il frammento successivo viene inviato in maniera atomica. Non appena qualcosa va storto, quindi un frammento non arriva a destinazione, chi stava trasmettendo perde il privilegio di occupazione del canale ed i frammenti successivi andranno trasmessi come il risultato di una competizione alla pari. Quello che la figura mostra è che l'informazione di durata contenuta nei pacchetti che circolano serve a garantire l'invio di un frammento, dell'ack successivo e del nuovo frammento. Se lo schema va avanti si riesce a trasmettere tutti i frammenti in maniera atomica. Man mano che si va avanti, i NAV delle altre stazioni vengono aggiornati in modo da continuare a considerare il canale occupato per il tempo necessario ad inviare la prossima coppia frammento più ack. Non appena questo non succede più, la cosa torna in una situazione di contesa alla pari fra tutti quanti.

Questo che vi ho presentato è il meccanismo base di condivisione del canale garantito dal protocollo. Quindi un accesso in cui tutte le stazioni, incluso l'AP se presente. Tutti gli interessati ad utilizzare questo canale, che fanno parte di un BSS, competono alla pari con questa modalità di accesso. Il protocollo garantisce anche una diversa modalità di accesso, che serve a dare delle garanzie forti sul ritardo massimo che la trasmissione può subire. E' vero che nello schema illustrato c'è il meccanismo di fairness che rende molto improbabile con il passare del tempo che l'attesa si prolunghi entro una certa durata, però non è possibile dare una certezza sul tempo massimo di attesa prima di trasmettere un pacchetto. Se questa certezza è necessaria, allora il protocollo mette a disposizione uno schema diverso. L'idea è quella di organizzare l'asse nei tempi in intervalli consecutivi, dove l'intervalli sono questi etichettati con CFP. La durata di questi intervalli è variabile nel senso che non è fissata una volta per tutte nel protocollo ma può essere decisa da ogni AP in maniera autonoma. Questo schema è applicabile soltanto in una situazione con infrastruttura in presenza di un AP. Nella durata di ogni intervallo, il tempo viene suddiviso in due parti. Una parte DCF, in cui l'accesso al canale avviene in modalità a contesa, ed una parte PCF dove invece l'accesso avviene con una modalità di tipo polling, in cui è l'access point che seguendo una sua politica raccoglie le richieste da parte dei nodi di utilizzare questa modalità di accesso al canale, durante questo intervallo interpella le varie stazioni una alla volta, e quando arriva il suo turno trasmetterà e passerà la palla nelle mani dell'AP che assegnerà il diritto di trasmettere alla successiva e così via, fino alla chiusura dell'intervallo. Sostanzialmente, il rapporto tra la durata della parte PCF e durata della

parte DCF all'interno di un repetition interval mi dirà quale frazione della banda disponibile verrà dedicata all'accesso a contesa e quale sarà la porzione dedicata all'accesso regolamentato, a prescindere dalla durata in termini assoluti. La lunghezza totale dell'intervallo dice quanto tempo chi vuole trasmettere in modalità PCF deve aspettare prima che gli venga assegnato il diritto di trasmettere. Se consideriamo che questa modalità viene utilizzata da chi ha necessità di trasmettere traffico a flusso continuo, a tasso regolare come segnali video, è un segnale che si accumula su chi deve trasmettere. Quanto più è lungo questo intervallo, quanto maggiore è la quantità di dati che si accumulano al mittente in attesa di essere trasmessi, che dunque dovranno essere memorizzati in un buffer. Quanto più lungo l'intervallo, quanto più capiente dovrà essere il buffer di cui il mittente dispone per conservare questi dati prodotti in attesa di essere trasmessi. La lunghezza dell'intervallo e la frazione di questo intervallo dedicata a questa modalità, viene negoziata tra l'AP e le stazioni che vogliono usare queste modalità. Come fa una stazione a sapere che si sta aprendo questa finestra? Le stazioni che non sono interessate devono sapere che c'è un intervallo di tempo in cui si devono astenere ad affacciarsi sul canale. Come fanno le stazioni a sapere quando inizia l'intervallo di tempo in cui chi vuole accedere a contesa deve astenersi dal farlo, in quanto l'uso del canale è regolamentato? L'avvio di ogni intervallo è segnalato dal segnale di **beacon**, come una luce che si accende periodicamente, e che segnala che sta iniziando questo periodo. Questo segnale viene emesso ad intervalli regolari dall'AP. Veicola con se una serie di informazioni. E' un segnale che si usa anche per gestire altri tipi di funzioni.

L'idea teorica sarebbe che l'asse dei tempi è stato suddiviso in intervalli regolari di uguale durata. All'inizio di ogni intervallo l'AP invia il suo segnale di beacon. Questo è lo scenario ideale. L'AP lavora alla pari con le altre. Quando arriva il tempo t_0 in cui teoricamente dovrebbe iniziare il ciclo di accesso riservato al canale, può succedere che il canale è occupato. L'AP deve comunque aspettare che termini la comunicazione in corso quando arriva all'istante t_0 . Terminata la comunicazione in corso, l'AP dovrebbe inviare il suo segnale di beacon. Ma notate che l'intervallo PCF non è ancora iniziato, in quanto il segnale di beacon segnala l'inizio del periodo di accesso non a contesa del canale. Succede che se il canale è con tendibile e lui va in contesa con gli altri, se l'AP è sfortunato perde la contesa e l'invio del beacon viene ritardato in modo non tollerabile. Per garantire all'AP di riuscire a trasmettere il suo beacon si dà un privilegio, quello di aspettare per trasmettere un tempo PIFS. Da un punto di vista della sua durata, il PIFS è maggiore dell'intervallo SIFS, quindi evita che l'AP si possa intromettere in uno scambio di dati ACK, ma è minore di DIFS, e questo garantisce che quando il canale si libera l'AP sarà il primo ad occuparlo ed a trasmettere il suo beacon. E' vero che rispetto all'intervallo teorico di inizio del periodo senza contesa ci può essere un ritardo, ma questo è limitato superiormente in maniera certa. Il limite superiore certo è il tempo necessario a trasmettere un pacchetto di lunghezza massima, trasmissione iniziata prima dell'arrivo del tempo t_0 , fissata a poco più di 2000 byte. L'AP dovrà aspettare che termini la trasmissione del pacchetto, caso peggiore, terminata la trasmissione aspetta un tempo PIFS e poi sicuramente riuscirà a trasmettere il suo beacon. Quello che segnala il beacon è un NAV che è pari alla durata di tutto il periodo senza contesa. All'interno di questo periodo l'AP interpella uno ad uno le stazioni. Dopo un tempo SIFS la stazione occupa il canale inviando i suoi dati, e questo ciclo si ripete ad intervalli tra l'invio dell'AP e delle stazioni distanziato di un tempo SIFS. Quello che può

succedere è che l'AP interpella una stazione, la stazione 3 e non riceve nessuna risposta. Che cosa fa l'AP? Aspetta un po, ma non può aspettare tanto perché ci sono gli altri interessati a trasmettere. E Non si può aspettare più di tanto anche perché bisogna evitare problemi che adesso vedremo. Questo tempo di attesa è pari a PIFS. L'AP aspetta questo intervallo di tempo, e trascorso passa al successivo. Il tempo è PIFS, inferiore a DIFS, in quanto se fosse DIFS, chi non ha aggiornato il NAV perché magari arrivato dopo, potrebbe trasmettere in modalità a contesa.

Funzioni di gestione. Ci sono varie funzioni di gestione necessarie per il buon funzionamento di questo meccanismo. Una di queste, basilare è il mantenimento in sincronia gli orologi, quanto più possibile, di tutti i nodi di un BSS. E' una funzione di base importante. Un'altra funzione di gestione è relativa alla gestione dell'energia. Come accennavo quando ho introdotto il protocollo, uno dei requisiti di progetto è di mettere in atto politiche di risparmio energetico. Poi altra funzione di gestione è lo schema di associazione, aggiunta di una stazione in un BSS. Funzione di base della sincronizzazione tra gli orologi delle varie stazioni. Questo è fondamentale. Tutte le stazioni devono essere in accordo sullo scorrere del tempo. Come si fa a mantenere gli orologi sincronizzati? La cosa è estremamente semplice in una situazione con infrastruttura, in cui l'orologio dell'AP farà da riferimento per tutti. Tutte le stazioni sono tenute ad aggiustare il loro orologio su quello dell'AP. Questa è una delle informazioni che vengono diffuse tramite il segnale di Beacon. Ad intervalli regolari di tempo, la durata tipica nelle implementazioni attuali è di 100 millisecondi, una delle informazioni veicolate dal beacon è il valore dell'orologio dell'AP. Il valore viene utilizzato dalle varie stazioni per aggiustare il valore dell'orologio interno. A cadenza regolare tutti quanti sono tenuti a riaggiustare l'orologio. Eventuali derive vengono corrette. L'invio del segnale di beacon può essere eventualmente ritardato. Il valore dell'orologio inglobato è quello esatto, non quello teorico. Non c'è altro da dire. La situazione è un po più complessa nello schema senza infrastruttura, in cui tutte le stazioni si ascoltano l'una con l'altra. In questo caso quello che succede è che ogni stazione, in base al suo orologio interno conta lo scorrere del tempo, ed in maniera regolare tenta di inviare un beacon. Quando scade il tempo teorico, ogni stazione cessa le sue attività di trasmissione e dà priorità all'invio del beacon. Tutte cercano di trasmettere in contesa, e chi la vince manda il valore del suo orologio. Non c'è un orologio privilegiato. L'avanzamento del tempo è un po meno uniforme rispetto al caso precedente perché non c'è un orologio di riferimento ma tanti orologi che si riallineano tra di loro ad intervalli regolari. Ma alla fine rimangono tutti quanti ragionevolmente riallineati.

L'allineamento è importante nel caso del risparmio energetico. Premessa fondamentale per capire il meccanismo fisico su cui si basa la politica di risparmio energetico che andiamo a descrivere. Questo meccanismo fisico si basa sul meccanismo fisico di operare delle schede di rete. E' una tabella relativa a schede di rete che implementano una delle prime versioni del protocollo, la b, ma le cose non sono cambiate. Una scheda di rete wireless si può trovare in vari stati. I primi tre sono nel momento in cui è accesa, l'ultimo quando è in stato sleep. Se guardate le informazioni relative al consumo energetico causato da questi tre stati, vediamo che l'attività che comporta il maggior consumo energetico è l'attività di invio dei dati, che consuma mediamente il doppio dell'energia necessaria per ricevere i dati, ma siamo comunque nello stesso ordine di grandezza. L'informazione rilevante è che anche se

una scheda di rete è inattiva, ma è accesa, il consumo di energia è sì minore rispetto alle situazioni precedenti, ma con lo stesso ordine di grandezza. Una scheda di rete accesa, anche se inattiva, consuma energia, anche tanta. Per questo motivo le schede di rete wireless sono tipicamente dotate della modalità sleep. E' una via di mezzo tra l'essere completamente acceso e l'essere completamente spento. In questa modalità ha quasi tutti i circuiti spenti, tranne un piccolo sottoinsieme relativo alla gestione del clock. La stazione in questo stato non è in grado né di ricevere né tanto meno di trasmettere. In questo stato il consumo energetico si abbassa di un ordine di grandezza rispetto alle situazioni precedenti. Questo avveniva sulla versione b del protocollo, ma in versioni avanzate la situazione si ripete. Date queste caratteristiche fisiche della scheda di rete wireless, l'idea del risparmio energetico qual'è?

E' quella di dire che se una stazione non ha nulla da trasmettere mette la sua scheda di rete in modalità sleep, non la spegne. Mettere in modalità sleep significa che non è in grado né di ricevere né di trasmettere. Se c'è qualcuno interessato a trasmettere dati ad una stazione in modalità sleep questo qualcuno non è in grado di portare a termine il suo lavoro. Se prova ad inviare un pacchetto sul canale wireless, nessuno si riconoscerà come destinatario. Quello che succede è che se una stazione si mette in modalità sleep, si deve periodicamente svegliare per capire se ci sono dati destinati a questa stazione. La politica di risparmio energetico, detta in due parole, è che le stazioni che vogliono risparmiare energia mettono in sleep l'interfaccia quando non vogliono trasmettere e ritornano in modalità attiva periodicamente. Il clock, la circuiteria continua ad essere attiva e conta lo scorrere del tempo. Quando scatta l'istante giusto le stazioni si accendono, verificano se ci sono dati destinati a loro e ritornano in sleep. Come commento generale, questo significa che chi ha dati da trasmettere verso una stazione che adotta questa politica, deve aspettare che il destinatario si svegli periodicamente. In attesa che si svegli, chi trasmette deve mantenere al suo interno i dati destinati ad una stazione in fase di sonno. Questo comporta un ritardo nell'invio dei dati. Una politica del genere fa risparmiare energia, ma comporta un allungamento nei tempi di completamento della trasmissione. C'è un compromesso da definire tra esigenza di risparmiare energia e tempestività della consegna dei dati. E' un compromesso che può essere sciolto in un modo o nell'altro a seconda del contesto in cui ci si trova.

Questa politica di risparmio energetico può essere applicata in situazioni con e senza infrastruttura. E' più semplice realizzarla in modalità con AP piuttosto che senza. La presenza di un'entità privilegiata semplifica le operazioni di gestione. Come si realizza il risparmio energetico? Qui entra in gioco il meccanismo del beacon. Tra le informazioni contenute nel segnale di beacon c'è l'informazione relativa alla mappa di quali sono le stazioni che hanno deciso di adottare una politica di risparmio energetico, e quali sono le stazioni per le quali esistono dati da essere. Questi dati sono stati memorizzati sull'AP in attesa di essere trasmessi. Quindi è una semplificazione notevole dovuta alla presenza dell'AP. Chi aveva dati da trasmettere li ha comunque trasmessi all'AP. L'AP sa chi sono le stazioni che adottano questo schema operativo e quindi immagazzina al suo interno i dati in attesa di trasmissione. Ogni volta che invia un segnale di beacon, invia la mappa delle stazioni per cui ci sono dati. Come fanno le stazioni destinatarie a ricevere questo segnale? Una stazione che mette in atto una politica di risparmio è comunque tenuta a svegliarsi con

l'istante di tempo in cui l'AP potrà trasmettere il suo beacon. Per questo è importante che gli orologi siano sincronizzati. Quando l'AP invierà il suo beacon, tutte quante devono essere sveglie. La mappa può essere di due tipi: TIM, e questa è una mappa che segnala la presenza di dati per singole stazioni. Oppure può essere una mappa di tipo DTIM, che indica la presenza di dati di tipo broadcast, destinati a tutte le stazioni del BSS. Mappe di tipo DTIM vengono inviate con una spaziatura maggiore rispetto alla mappa TIM. Quando arriva il tempo per l'invio di una nuova mappa la stazione si deve risvegliare. Può arrivare con un tempo di ritardo, ma fin quando non arriva la stazione deve rimanere accesa. Tramite il beacon riceve la mappa, ed in questo esempio la stazione non si riconosce tra quelle indicata nella mappa. Non ci sono dati per lei, la stazione si spegne, rimane spenta, il canale verrà utilizzato da altre stazioni e poi di nuovo allo scattare del nuovo istante di tempo, la stazione ha il suo orologio aggiustato con il valore contenuto del beacon, si risveglia, riceve la mappa, e nell'esempio in figura nella mappa la stazione viene nominata. Segnala all'AP di essere sveglia, l'AP invia i suoi dati, riceve altri dati e/o ACK dalla stazione e poi fatto questo la stazione si può spegnere. Altre stazioni faranno il loro invio e poi la situazione si ripete. Nel caso AP tutto liscio, abbastanza semplice. AP fa da buffer, regola l'uso del canale.

Nel caso senza AP, la cosa diventa leggermente più complessa non essendoci un'entità privilegiata. Di nuovo è importante la sincronizzazione degli orologi, tutti devono essere d'accordo su quando svegliarsi. Le stazioni si svegliano in parallelo e cercano di inviare il segnale di beacon che segnala l'inizio di una fase di scambio dati. Appena svegliati tutti quanti cercano di inviare il beacon ed una ci riuscirà a contesa. Il segnale di beacon viene ricevuto da tutte le stazioni del BSS, e quindi capiscono che è iniziata la così detta ATIM windows. In questa finestra tutte le stazioni sono tenute ad essere accese e sono autorizzate ad inviare la loro mappa di dati destinati ad altre stazioni. In questo intervallo ogni stazione cerca di informare le altre che ha dei dati destinati a loro. In questo primo scambio nessuna delle due stazioni invia mappe di questo tipo. Terminata l'ATIM windows le stazioni si spengono in quanto non ci sono dati da inviare nell'esempio. Arriva il momento di risvegliarsi, le due stazioni si risvegliano, la stazione 2 vince la contesa per l'invio del beacon, in figura nuovamente nessuno ha nulla da inviare alle altre. Trascorre il tempo, ritorna il tempo in cui si devono accedere, la stazione due vince la contesa, e questa volta all'interno della finestra, la stazione 1 ha da trasmettere qualcosa verso la stazione 2, quindi la stazione 1 segnala questo fatto, la stazione 2 invia un ack, fa capire di avere ascoltato la richiesta di invio. A questo punto allo scadere dell'ATIM window le due stazioni rimangono accese per tutto il tempo necessario a completare lo scambio. Poi la storia si ripete. Rispetto allo scenario con infrastruttura, lavorando tutti alla pari, la funzione di buffer deve essere svolta da tutti quanti. Gestire un buffer è un'attività che può comportare un consumo energetico più o meno importante. C'è anche una questione di scalabilità, in quanto all'aumentare del numero di stazioni che applicano questa politica di risparmio energetico, maggiore sarà la dimensione delle mappe ATIM inviate durante l'ATIM window. Se la dimensione del BSS è abbastanza grande ed il numero di stazioni che vogliono applicare questa politica anche, ci possono essere problemi di scalabilità.

C'è un altro problema un po' delicato relativo all'effettiva efficacia di questo meccanismo. Se guardiamo la situazione in maniera restrittiva a livello 2, interfaccia di rete, è una politica

che sembra fare risparmiare energia. Se nei periodi di inattività della stazione io cerco di mettere, per quanto più tempo possibile la sua scheda di rete in modalità sleep, sicuramente riesco a risparmiare in maniera notevole energia, o almeno mi aspetto questo. Però quello che succede nella realtà può essere diverso. Alcuni ricercatori hanno sperimentato che se uno va a guardare quello che effettivamente succede nella realtà, è che stazioni che adottano la politica di risparmio energetico descritto, si possono trovare a consumare più energia. La ragione è dovuta al fatto che se è vero che se ci limitiamo a guardare come vanno le cose a livello 2, sono politiche applicate al risparmio energetico. Ma se vediamo come quello che succede a questo livello succede a livelli diversi, si possono creare interazioni perverse che annullano i potenziali effetti positivi di una politica di risparmio energetico, e questo è lo scenario che vedete. Esperimento semplice: si cerca di trasmettere utilizzando un protocollo TCP un file di 00 KB su un link wireless gestito nel protocollo descritto con la politica di risparmio energetico. Commento alle figure. Premessa: quando io utilizzo un link che ha una certa banda teorica, quindi un certo numero di byte per unità di tempo, se devo trasferire un file, la quantità di byte che vengono trasmesse cresce in maniera lineare con lo scorrere del tempo. Se mi immagino il grafico teorico che ha sull'asse delle x lo scorrere del tempo ed in ordinata la quantità di byte trasmessi, mi aspetto di osservare in assenza di interferenze è una linea retta che parte dall'origine. Concretamente se il protocollo è di tipo TCP, l'idea teorica può essere parzialmente smentita. Come sappiamo, TCP è congegnato in maniera tale da non sfruttare da subito tutta la banda disponibile. Usa un approccio cauto. All'inizio usa solo una frazione della banda disponibile con una crescita esponenziale se le cose vanno bene, fino a saturare la capacità del canale, in cui si realizza la crescita lineare. Le due figure illustrano quello che succede a causa della presenza del protocollo TCP. Le due figure sono per certi versi simili. Per un certo tempo, la crescita con il tempo del numero di pacchetti inviato è abbastanza lenta. L'inclinazione della curva è minore. A partire da un certo istante di tempo in cui si satura la banda la crescita procede in modo lineare con velocità superiore. Quello di diverso tra le figure è la differenza di angolazione e la piattezza della parte iniziale a sinistra. A cosa è dovuta questa maggiore lentezza nel raggiungere la saturazione? E' dovuta al fatto che in questo esperimento, il RTT è di 25 ms, mentre nel secondo è di 75 ms. A fronte di un intervallo tra due beacon successivi che è di 100 ms, valore tipicamente usato dal protocollo 802.11. E' abbastanza vicino al RTT (il tempo necessario per inviare pacchetti e ricevere l'ack, quindi per fare il raddoppio nell'invio di pacchetti TCP), mentre nel primo caso siamo parecchio sotto la durata del beacon. Nel secondo caso c'è una quasi perfetta sintonia tra completamento delle fasi TCP e fasi di sveglia/addormentamento. Le due fasi vanno abbastanza in sintonia e non si disturbano l'una con l'altra. Nel primo caso invece dal punto di vista del protocollo TCP invio i pacchetti e sarei subito in grado di trasmettere l'ACK, però il fatto di essermi addormentato ed addormentato solo dopo 100 ms rallenta lo scambio dati ack. Rispetto alla velocità teoricamente raggiungibile per il raggiungimento della saturazione, questo viene rallentato per la mancanza di sintonia tra il protocollo TCP e l'alternanza delle fasi di sveglia/addormentamento della scheda di rete. Questo provoca in proporzione che il tempo totale per terminare l'invio diventa molto maggiore del tempo necessario per completare lo stesso invio in assenza di una politica di risparmio energetico. Se devo sommare tutti i tempi in cui una scheda di rete dovrà essere accesa per consentire l'invio e la ricezione dei dati, la somma di questi tempi essendo il tempo necessario a completare il trasferimento più lungo, la somma dei tempi sarà

maggiore del caso in cui la scheda di rete sarebbe rimasta accesa in assenza della politica di risparmio energetico. Essermi spento quando non c'era nulla da trasmettere mi ha fatto risparmiare energia solo nel secondo caso. Quando si vogliono ottimizzare le prestazioni di un sistema costituito da una stratificazione di livelli, ognuno dei quali sfrutta le funzioni offerte dai livelli sottostanti, cercare di raggiungere degli obiettivi di efficienza ed ottimizzazione lavorando solo ad un livello ignorando le possibili interazioni con altri livelli, rischia di essere una scelta miope e non efficace. Se guardo le cose in modo isolato, appena allargo l'orizzonte scopro che in realtà il discorso è più complesso. Non è una buona notizia il discorso dell'isolamento tra livelli, si è un potente strumento dal punto di vista della progettazione dei sistemi, permettendo ad ogni progettista di concentrarsi sulle problematiche del singolo livello, ma dal punto di vista dell'ottimizzazione complessiva bisogna cercare di allargare la visione. Altrimenti si rischia di ottenere il contrario di quello che si vorrebbe ottenere.