Università degli Studi dell'Aquila

Facoltà di Ingegneria

Corso di Laurea in Ingegneria delle Telecomunicazioni

Studio e sviluppo di un sistema embedded per la trasmissione dati con tecnologia HSPA

Relatore	**Studente**
Prof. Fabio Graziosi	Diego Tuzi
Correlatore	**Matricola**
Ing. Andrea Colarieti	172629

A.A. 2009-2010

Premessa

Questa tesi è intitolata "Studio e sviluppo di un sistema embedded per la trasmissione di dati con tecnologia HSPA". L'organizzazione del documento discende direttamente dal titolo. Dopo il primo capitolo di introduzione, i capitoli 2, 3 e 4 si occupano della parte di "studio" del sistema dati.

Il capitolo 2 fornisce informazioni teoriche sulla tecnologia HSPA, partendo dal WCDMA per poi passare alle tecnologie HSDPA e HSUPA.

Il capitolo 3 è interamente dedicato al modulo che permette di accedere alla tecnologia HSPA, con particolare attenzione ai comandi AT che sono necessari per controllare il modulo stesso.

Il capitolo 4 si occupa dell'altro componente del sistema di trasmissione dati, ovvero il sistema di elaborazione. Il capitolo fornisce gli strumenti necessari per operare le configurazioni necessarie affinché il sistema di elaborazione possa controllare il modulo che fornisce la connessione. Particolare attenzione viene riservata al ppp (point to point protocol) e alla sua implementazione nei sistemi operativi basati su kernel linux, ovvero pppd (ppp daemon).

Arrivati a questo punto si hanno tutte le conoscenze necessarie per iniziare la fase di "sviluppo".

Il capitolo 5 descrive il procedimento di realizzazione del sistema di trasmissione dati. Nella parte iniziale del capitolo si configurano i moduli singolarmente. Nella seconda parte si configura l'intero sistema. Per concludere si fornisce un esempio concreto del sistema dati, utilizzandolo in una applicazione denominata "trappola fotografica".

Indice

1

Introduzione

Il mondo delle Telecomunicazioni (TLC), e più precisamente quello delle reti dei servizi di comunicazione, sta vivendo una fase di grande trasformazione dai significativi impatti non solo sugli aspetti industriali ed economici del settore, ma anche sull'intero sistema socio-economico del mondo globalizzato. Tutto ciò come conseguenza dello sviluppo dell'ICT (Information & Communication Technology) che rende sempre più possibile memorizzare, elaborare e trasmettere grosse quantità di informazioni tra due punti del globo terrestre senza particolari vincoli di tempo, ed a costi contenuti. Una delle parole chiavi di questa trasformazione è la *"Larga Banda"*.

La *Larga Banda* esprime la crescente capacità della delle reti digitali di nuova generazione di trasportare fino all'utente grossi volumi di dati, garantendo così la fornitura in rete di qualunque servizio multimediale interattivo. La necessità della larga banda è scaturita dalla sempre maggiore richiesta di utilizzo della rete internet, da parte degli utenti, insieme all'evoluzione della tipologia dei servizi richiesti.

La crescente necessità di utilizzare servizi con flusso informativo maggiore e la necessità di ricevere questi servizi in mobilità, caratterizzano la grande evoluzione delle reti mobili a cui si sta assistendo in questi anni.

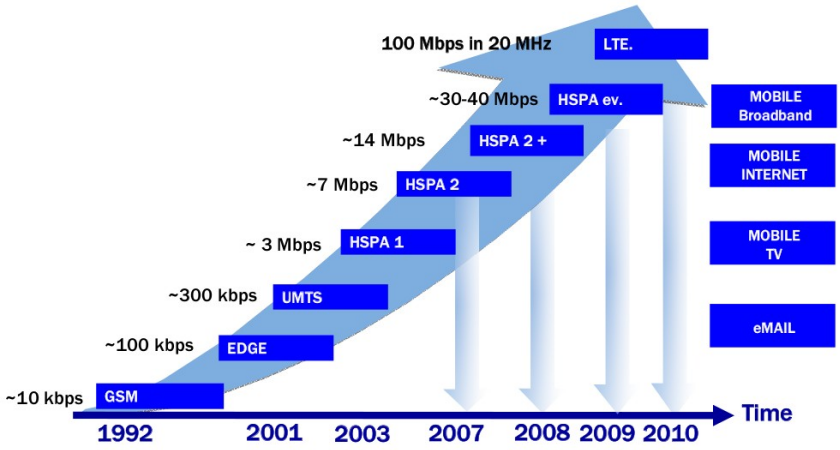

Figura 1: Evoluzione delle tecnologie di accesso alle reti mobili

Nel corso degli anni l'utilizzo della rete mobile, da parte degli utenti, si è

intensificato da più punti di vista. Innanzitutto sono cresciuti il numero degli utilizzatori e la quantità dei servizi disponibili.

Le evoluzioni delle tecnologie di accesso alla rete mobile, inizialmente sono state mirate principalmente ad aumentare il data rate del canale di downlink. Con le tecnologie più recenti ha assunto sempre maggiore importanza anche lo sviluppo del canale di uplink, ciò discende dal fatto che il comportamento dell'utente si sta trasformando. Nato come fruitore di contenuti, grazie al continuo sviluppo dei terminali di accesso, l'utente assume anche il ruolo di fornitore di contenuti. Con il diffondersi dei social network, l'integrazione di essi su ormai tutti i terminali utente, la facilità di utilizzo e il costo per bit notevolmente ridotto, hanno come risultato che l'asimmetria tipica tra il canale di downlink e quello di uplink diviene meno marcata. Ciò spiega lo sviluppo di tecnologie come l'HSUPA e superiori, ove il canale di uplink assume un ruolo sempre maggiore.

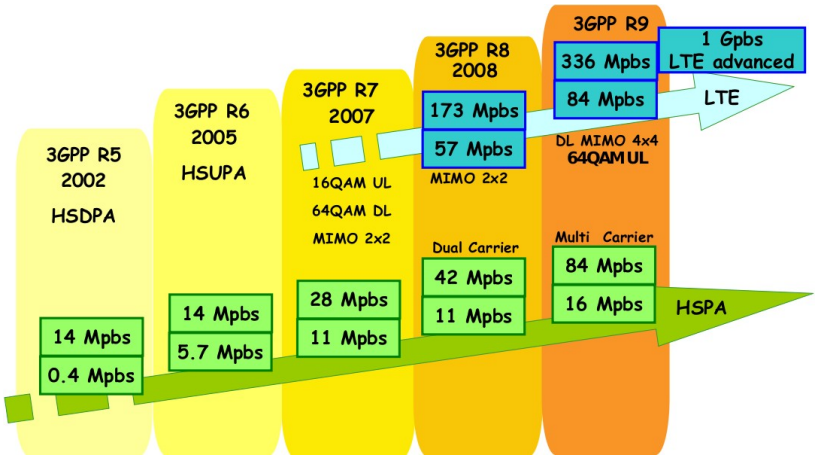

Figura 2: Evoluzione delle tecnologie di accesso alle reti mobili con velocità e tecnologie caratterizzanti

Una ulteriore motivazione dello sviluppo del canale di uplink, è collegata ad uno dei grandi cambiamenti che la rete sta per subire, denominato *"The internet of things"*, ovvero l'internet delle cose.

L'Internet delle cose genererà un grande cambiamento nella Rete, in quanto la connessione avverrà non più solo tramite computer, ma anche attraverso gli *Smart Object* (letteralmente 'oggetti intelligenti'), piccoli dispositivi, di solito delle dimensioni di un paio di centimetri, dotati di processore, memoria, capacità di archiviazione, modulo di comunicazione e una carica di energia. Uno smart object non differisce poi molto da un computer in miniatura, dotato di un sensore o attivatore. Tali oggetti possono essere utilizzati in una miriade di applicazioni, ma la

loro maggiore funzione è quella di raccogliere dati e inviarli utilizzando la connessione di rete disponibile.

Il crescente numero di utilizzatori, costituito da utenti e oggetti, spiegano il grande interesse nello sviluppo di tecnologie di accesso alla rete internet.

Lo sviluppo delle reti mobili rientra anche in uno degli obiettivi generali dello sviluppo delle reti di telecomunicazioni che è quello di realizzare la *Convergenza*, ovvero permettere all'utente di usufruire, in modo trasparente, delle prestazioni di mobilità e della larga banda, praticamente ignorando la rete a cui è agganciato.

2

High-Speed Packet Access (HSPA)

High Speed Packet Access (HSPA) è una famiglia di protocolli per la telefonia mobile che estendono e migliorano le prestazioni dell'UMTS. Con il termine HSPA si fa riferimento congiuntamente allo standard HSDPA, che migliora sensibilmente le prestazioni riguardanti il collegamento di downlink rispetto alle precedenti tecnologie, e anche allo standard successivo, ovvero l'HSUPA che completa l'evoluzione iniziata con l'HSDPA, attraverso il miglioramento del collegamento in uplink.

L'introduzione del HSDPA rappresenta il primo passo nella evoluzione del WCDMA (Wideband Code Division Multiple Access). Sebbene la comunicazione a pacchetto era supportata già nella prima release dello standard WCDMA, HSDPA porta dei miglioramenti in termini di sistema e in termini di performance sull'utente finale. L'evoluzione della parte downlink avvenuta con l'HSDPA è stata completata

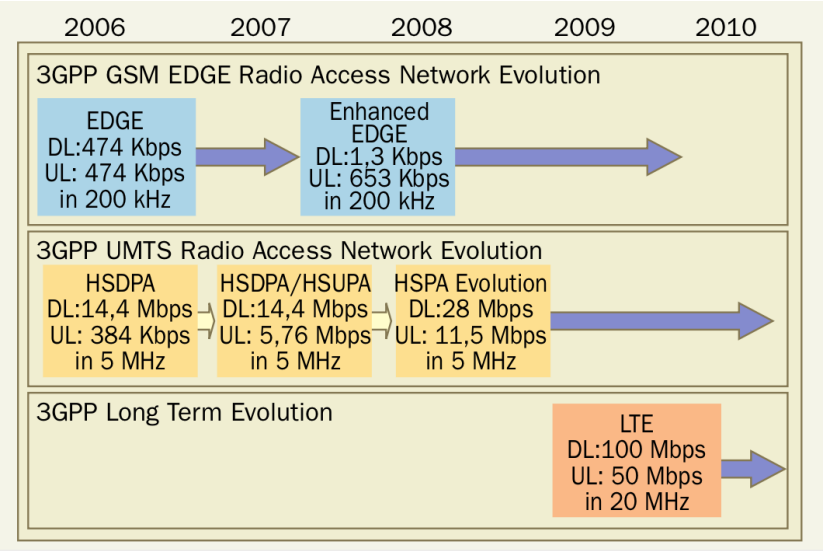

Figura 3: Recente evoluzione dei principali sistemi mobili

con l'evoluzione della parte uplink avvenuta con l'introduzione dell'Enhanced Uplink, spesso conosciuto con il nome di HSUPA. In genere ci si riferisce congiuntamente a HSDPA e HSUPA parlando di HSPA (High-Speed Packet Access). Per completare il panorama delle attuali tecnologie mobili si devono menzionare l'HSPA evolution e l'LTE (long term evolution). L'HSPA evolution consiste in un upgrade dello standard

HSUPA, mentre l'LTE rappresenta una nuova generazione per i sistemi di accesso mobile a larga banda e si colloca in una posizione intermedia fra gli attuali standard 3G come l'UMTS e quelli di quarta generazione (4G) ancora in fase di sviluppo.
A differenza dell'HSPA e dell'HSPA Evolution, che utilizzano la stessa copertura radio della rete UMTS, nel caso dell'LTE è necessario predisporre una copertura radio dedicata, realizzando di fatto una nuova rete aggiuntiva a quella dell'UMTS, o di qualsiasi altro sistema di accesso cellulare.
L'ente che si occupa di definire le specifiche tecniche per i sistemi mobili è *3GPP (The 3rd Generation Partnership Project)*. Esso è nato da un accordo di collaborazione fra enti che si occupano di standardizzare sistemi di telecomunicazione in diverse parti del mondo. Una volta definito uno standard, tutte le specifiche tecniche vengono raccolte in una Release.
Il WCDMA è definito nella Release 99, l'Hsdpa nella release 5, l'Hsupa nella release 6, l'HSPA evolution nella release 7 e l'LTE nella release 8.

2.1 – Wideband Code Division Multiple Access (WCDMA)

Per capire meglio le tecnologie HSPA è utile partire dallo standard WCDMA dal quale le tecnologie HSPA discendono. Lo scopo di questo paragrafo permette di evidenziare i miglioramenti apportati al WCDMA con l'introduzione delle tecnologie HSPA.
Il WCDMA è una interfaccia radio altamente flessibile e versatile che può essere configurata per incontrare le specifiche di una vasta gamma di servizi. Nel seguito faremo riferimento alle funzionalità comunemente usate per la trasmissione di dati a pacchetto.

2.1.1 – Architettura generale

WCDMA si basa su una architettura gerarchica con differenti nodi e interfacce come mostrato nella Figura 4. Un terminale comunica con uno o più Nodi B. Nella architettura WCDMA, il termine Nodo B si riferisce al nodo logico responsabile per i processi a livello fisico come la correzione degli errori di codifica e la modulazione, ovvero del processo di conversione del segnale in banda base in segnale a radio frequenza trasmesso dall'antenna. Un Nodo B controlla trasmissione e ricezione attraverso una o più celle. Una stazione base è una possibile implementazione di

Nodi B.

L'RNC (Radio Network Controller) controlla più Nodi B. Il numero di Nodi B connessi ad un RNC dipende dall'implementazione e dal contesto, ma in generale il numero si aggira intorno ad un centinaio di Nodi B. L'RNC si occupa, tra i tanti compiti, di gestire il call setup, la qualità del servizio offerto e la gestione delle risorse radio delle celle di cui è il responsabile. Nell'RNC si trova anche il protocollo ARQ , che si occupa delle ritrasmissioni dei dati errati. Quindi nella Release 99, l'intelligenza nella rete di accesso risiede negli RNC, mentre i Nodi B principalmente agiscono come modem. Infine gli RNC sono connessi ad Internet e alla tradizionale rete telefonica (PSTN) attraverso la core network.

La Core network (CN) è responsabile dello switching e routing delle chiamate e delle connessioni dati verso le reti esterne: reti a commutazione di circuito (con tecnologia ISDN-derived) oppure reti a commutazione di pacchetto (con tecnologia IP-derived).

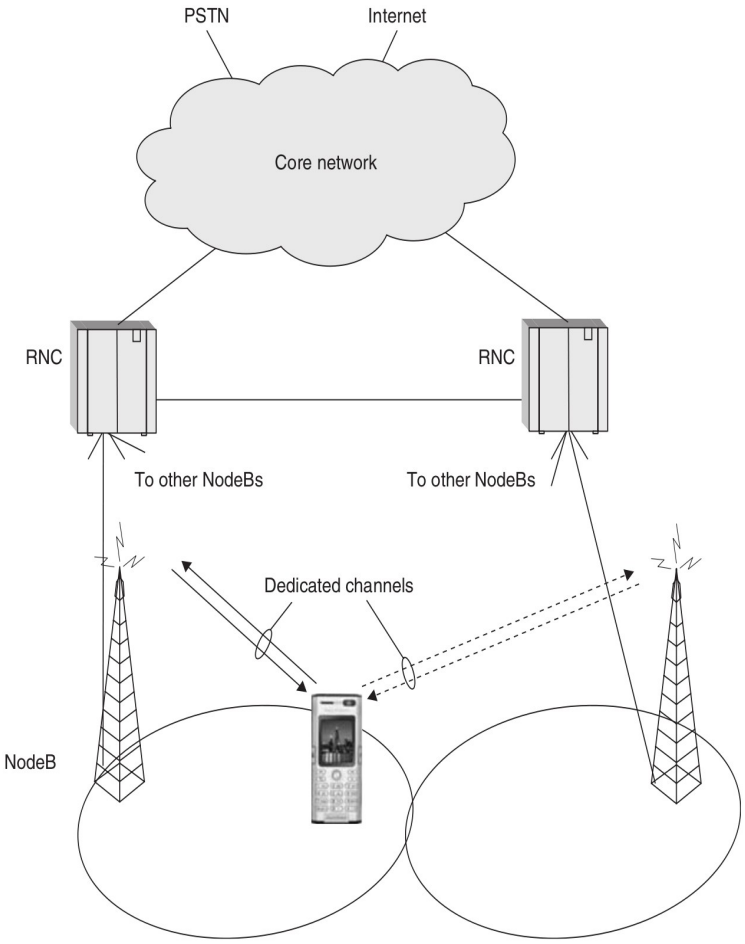

Figura 4: Architettura WCDMA

Un numero variabile di Nodi B controllati da uno o più RNC, in base a criteri di tipo

geografico costituiscono gli RNS (Radio Network Subsystem). L'insieme di tutti gli RNS costituisce la rete di accesso denominata UTRAN (UMTS Terrestrial Radio Access Network).

Architettura Protocollare

Molti dei moderni sistemi di comunicazione strutturano i processi in differenti livelli e il WCDMA non fa eccezioni. Questo tipo di approccio apporta dei benefici in quanto esso fornisce una determinata struttura all'intero processo dove ogni layer è responsabile per una specifica parte delle funzionalità dell'accesso radio. I differenti protocolli utilizzati dal WCDMA si possono osservare nella Figura 5.

I dati utenti provenienti dalla core network, per esempio in forma di pacchetti IP, sono inizialmente processati dal *Packet Data Convergence Protocol (PDCP)* che effettua una compressione dell'header. Pacchetti IP hanno un header relativamente pesante, 40 byte per IPv4 e 60 byte per IPv6, quindi per preservare risorse sull'interfaccia radio, la compressione dell'header è importante.

Il *Radio Link Protocol (RLC)* è, tra i tanti compiti, responsabile per la segmentazione dei pacchetti IP in unità più piccole denominate RLC Protocol Data Unit (RLC PDUs). Alla fine della ricezione l'RLC si occupa del riassemblaggio dei segmenti ricevuti. Il protocollo RLC implementa anche il protocollo ARQ. Per servizi di dati a pacchetto, la consegna priva di errori dei dati è spesso essenziale e per questo motivo l'RLC può essere configurato per richiedere la ritrasmissione delle RLC PDUs errate. Per ogni PDU erroneamente ricevuta, l'RLC richiede la ritrasmissione.

Il layer *Medium Access Control (MAC)* offre servizi all'RLC nella forma dei così detti Canali Logici (Logical channels). Il layer MAC può multiplare dati da più canali logici. Il MAC è anche responsabile della determinazione del formato di trasporto (Transport Format) dei dati inviati al successivo layer, il *livello fisico (Physical Layer)*. In pratica, il formato di trasporto è il data rate che in quell'istante viene usato sul collegamento radio. L'interfaccia tra livello MAC e livello Fisico è specificata attraverso i così detti Canali di Trasporto (Transport Channels) sui quali i dati, nella forma di traspor blocks, sono trasferiti.

In ogni Transmission Time Interval (TTI) uno o più transport blocks sono inviati dal layer MAC al layer fisico, che effettua codifica, interleaving, multiplexing e spreading, etc., prima di trasmettere i dati. Un TTI più grande implica un migliore time diversity, ma anche un ritardo più grande. Nella release 99 che specifica il

WCDMA, il TTI ha una lunghezza di 10, 20, 40 e 80 ms. Per supportare differenti data rates, il MAC può variare il transport format tra due consecutive TTIs. IL formato di trasporto consiste in alcuni parametri che descrivono come i dati dovrebbero essere trasmessi in un TTI. Variando la dimensione e il numero dei transport blocks, possono essere realizzati differenti data rates.

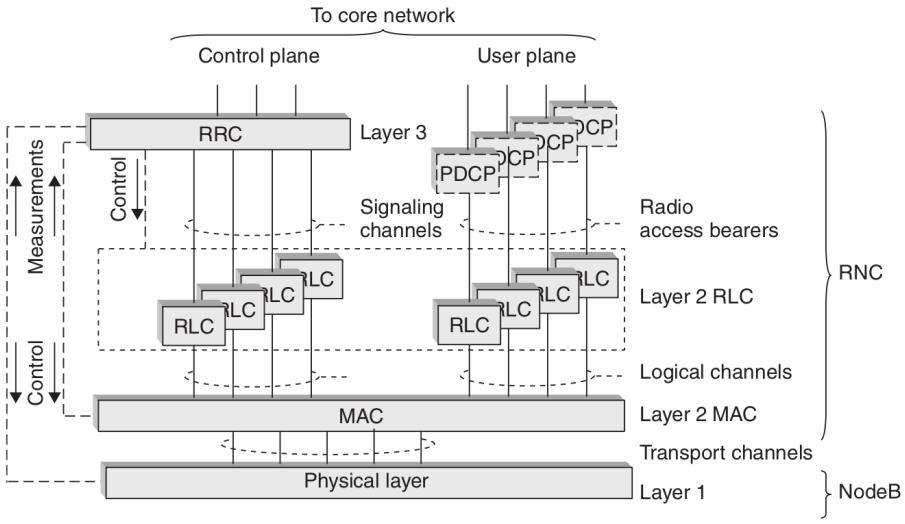

Figura 5: Architettura protocollare del WCDMA

PDPC, RLC, MAC e livello fisico, sono configurati dal protocollo Radio Resource Control (RRC). Settando i parametri di RLC, MAC e livello fisico in modo appropriato, l'RRC può fornire la qualità del servizio richiesta dalla core network per un determinato servizio.

Dal lato rete, le entità MAC, RLC, e RRC sono tutte collocate nel RNC mentre il livello fisico risiede principalmente nei Nodi . Le stesse entità esistono anche per ciascun terminale. Per esempio il MAC nei terminali è responsabile della selezione del formato di trasporto, effettuando la scelta tra un set definito dalla rete, per le trasmissioni in uplink. Comunque, la gestione delle risorse radio di una cella sono controllate dalle entità RRC nella rete e i terminali seguono le decisioni prese dall'RRC nella rete.

2.1.2 – Livello fisico

È particolarmente interessante approfondire il livello fisico perché la maggior parte delle modifiche che portano a definire L'HSPA dal WCDMA risiedono in questo livello.

La funzione base del livello fisico consiste nella operazione di *spreading*, tuttavia tale livello svolge numerose operazioni come la codifica, la multiplazione dei canali di trasporto, e la modulazione della portante a radio frequenza. Un semplice riepilogo di ciò che accade nel livello fisico è riportato in Figura 6.

Per ogni transport block da trasmettere, il livello fisico aggiunge una *Cyclic Redundancy Check (CRC)* allo scopo della rilevazione degli errori. Successivamente i dati vengono codificati utilizzando un *Turbo Coder* con rate 1/3. Il *rate matching* è un processo che provvede a rendere la dimensione dei transport block uguale a quella del frame radio. L'*Interleaving* è il processo con il quale un flusso di bit o simboli da una PDU o una SDU può essere ripartito in uno specifico ordine utilizzando bit e simboli di altre PDU o SDU. Questa tecnica è spesso usata per ridurre l'effetto degli errori in una trasmissione su un collegamento radio. Successivamente i canali di trasporto vengono multiplati insieme, formando un singolo flusso di bit da essere convertito in chip rate e poi modulato. Per il downlink, viene utilizzata la modulazione QPSK, mentre per l'uplink si usa la BPSK.

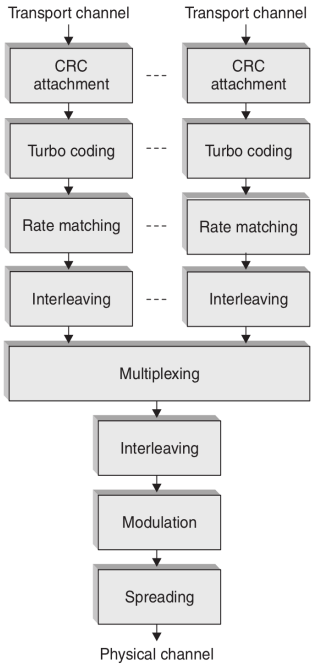

Figura 6: Riepilogo processi del livello fisico nello standard WCDMA

Il risultante flusso di simboli modulati viene mappato su un canale fisico e successivamente fatto passare in una conversione digitale – analogico e modulata su una portante a radio frequenza.

In linea di principio, per separare le trasmissioni di utenti diversi, si associa ad ogni canale fisico un codice di spreading diverso.

Spreading e Scrambling

La funzione più interessante del livello fisico è quella di *Spreading*. Ogni comunicazione è distinta dalle altre tramite l'assegnamento di un codice. I codici devono permettere di distinguere tra loro:

1. Le differenti sorgenti trasmissive:
   (a) Stazioni base nella tratta di downlink
   (b) Terminali mobili nella tratta di uplink
2. Le differenti trasmissioni di una singola sorgente:
   (a) Differenti canali di una stessa stazione base nella tratta di downlink
   (b) Differenti canali di uno stesso terminale mobile nella tratta di uplink.

Tutto questo è ottenibile tramite la moltiplicazione del segnale di sorgente non con un solo codice ma con 2 codici: un codice di *Scrambling* che permette di distinguere le differenti sorgenti trasmissive); codici di *Spreading* o *Canalizzazione* che separano i segnali di una singola sorgente trasmissiva e allargano lo spettro del segnale. Quali che siano i codici utilizzati il chip rate finale del segnale che deve essere trasmesso è fissato a 3.84 Mchip/s.

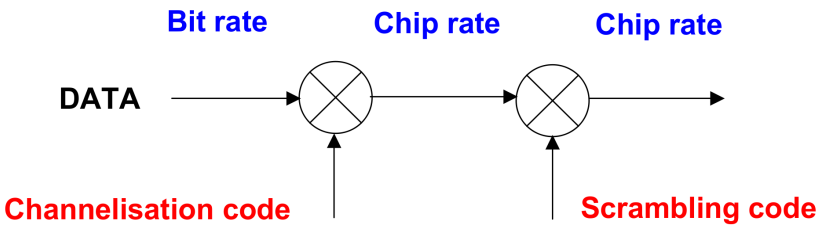

Figura 7: Spreading e scrambling

Nella tratta di downlink, i dati modulati QPSK di un determinato utente, inclusa la necessaria informazione di controllo, sono trasportati su un canale fisico dedicato (DPCH – Dedicate Physical Channel) codificati con corrispondente codice OVSF. Variando il fattore di spreading, possono essere forniti differenti symbol rate sui DPCH. Il fattore di spreading è daterminato dal formato di trasporto dal livello MAC. C'è anche la possibilità di usare più canali fisici per ogni singolo utente in modo da elevare il data rates.

Alcuni channelization codes nel downlink sono pre allocati per specifici scopi. Uno dei più importanti è il codice usato per il segnale di riferimento in trasmissione, che nel WCDMA esso è denominato Common Pilot Channel (CPICH). Il CPICH contiene informazioni note ed è usato come riferimento per la stima del canale di

downlink da tutti i terminali della cella.

I canali fisici sono separati dai codici OVSF, le trasmissioni su differrenti canali fisici sono ortogonali e non interferiranno con gli altri. Il WCDMA è perciò detto *ortogonale*. Comunque, dal lato ricezione, l'ortogonalità viene in parte persa in caso di canale selettivo in frequenza.

Nella tratta di uplink, i dati modulati BPSK sono trasportati dal Dedicated Physical Data Channel (DPDCH). In modo simile al caso di downlink, sono realizzati differenti data rates usando differenti fattori di spreading per il DPDCH. Anche nel caso di uplink si utilizza la demodulazione coerente, che richiede una stima del canale. A differenza del downlink dove si usa un segnale pilota comune, le trasmissioni di uplink sono originate da differenti sorgenti. Perciò, non può essere usato un segnale pilota comune ma ogni utente deve avere un segnale pilota separato. Questo segnale è trasportato su un Dedicated Physical Control Channel (DPCCH). Il DPCCH trasporta anche informazioni circa il formato di trasporto dei dati trasmessi sul DPCCH. Questa informazione è richiesta dal livello fisico nei Nodi B per una corretta demodulazione.
Nella tratta di uplink del WCDMA sono utilizzati codici di scrambling specifici per ogni utente, mentre I channelization codes sono utilizzati solo per separare differenti canali fisici utilizzati dallo stesso utente. Lo stesso set di channelization codes può essere usato da più terminali. Siccome le trasmissioni da differenti terminali non sono sincronizzate nel tempo, la separazione dei differenti terminali usando i codici OSVF non è possibile. Quindi, l'uplink è *non ortogonale* e le trasmissioni di utenti diversi interferiscono tra loro.

Fast closed-loop power control

Il fatto che l'uplink sia non ortogonale rende il controllo di potenza ad anello chiuso (fast closed-loop power control) una caratteristica essenziale del WCDMA. Con il fast closed-loop power control, il Nodo B riceve una misura del rapporto segnale-interferenza (SIR) sul DPCCH da ogni terminale e 1500 volte ogni secondo comanda ai terminali di regolare il livello di potenza in base alle rilevazioni effettuate. Lo scopo del controllo di potenza è quello di assicurare che il SIR ricevuto sul DPCCH sia ad un appropriato livello per ogni utente. Da notare che il livello di

SIR richiesto dipende dal data rate e può essere differente per ogni utente. Se il SIR è sotto la soglia richiesta, ciò significa che è troppo basso per una corretta demodulazione, in tal caso il Nodo B comanda al terminale di alzare la potenza di trasmissione. In modo analogo, se il SIR ricevuto è superiore alla soglia, al terminale viene comandato di ridurre la potenza in trasmissione. Se il controllo di potenza non fosse presente comporterebbe una elevata interferenza segnali utenti, che causerebbe la possibile non decodifica di alcuni di essi. Questo problema è comunemente noto con il termine *near-far problem*: trasmissioni da utenti vicini alla stazione base saranno ricevute dalla stessa con con un significativo livello di potenza superiore rispetto ad utenti lontani dalla stazione base, rendendo la demodulazione impossibile per tali utenti a meno che non venga utilizzato il controllo di potenza.
Il fast closed-loop power control è utilizzato anche in downlink, sebbene non motivato principalmente dal near-far problem grazie al fatto che il downlink è ortogonale. Il controllo di potenza si rivela utile anche nel downlink in quanto combatte gli affievolimenti rapidi. Quando le condizioni del canale sono favorevoli, c'è un minore utilizzo di potenza in trasmissione e viceversa. Tale comportamento porta ad una minore potenza media trasmessa, che si traduce in una minore interferenza media tra celle e in un miglioramento della capacità del sistema.

Soft handover o macro diversità

Un altra caratteristica chiave dello standard WCDMA è il soft handover. Esso permette ai terminali di comunicare contemporaneamente con più celle, generalmente appartenenti a più Nodi B, al fine di migliorare le performance. L'insieme di celle con cui il terminale sta comunicando viene definito *active set*. L'RNC determina, basandosi su misure ricevute dai terminali, l'insieme delle celle che formano l'active set.
Nel downlink, il soft handover implica che i dati per un terminale siano trasmessi simultaneamente da più celle. Ciò fornisce diversità contro il fading rapido.
Nell'uplink il soft handover implica che la trasmissione da un terminale è ricevuta da più celle. Le celle possono spesso appartenere a differenti Nodi B. Anche in questo caso la ricezione dei dati in più locazioni apporta benefici e fornisce diversità contro il fading rapido. Nel caso in cui le celle che ricevono la trasmissione di uplink facciano parte dello stesso NodoB, la combinazione dei segnali ricevuti viene effettuata nel livello fisico del ricevitore. Nel caso in cui le celle appartengano a Nodi

B differenti non è possibile combinare i segnali ricevuti a livello fisico, ogni NodoB prova a decodificare il segnale e inoltra le unità dati ricevute correttamente all'RNC. Il terminale considera la trasmissione avvenuta con successo quando almeno un NodoB riceve i dati in modo corretto, nel caso in cui più Nodi B ricevano correttamente il segnale l'RNC può scartare gli eventuali duplicati.

Il soft handover entra anche nel merito del power control. Per richiedere un abbassamento del livello di potenza in trasmissione da parte del terminale è sufficiente che una sola delle celle, a cui il terminale è collegato, lo richieda. Mentre per avere un innalzamento della potenza in trasmissione da parte del terminale tutte le celle dell'active set devono richiederlo. Questo meccanismo conosciuto con il termine di *or-of-the-downs*, assicura che il valore della potenza media trasmessa sia mantenuto il più basso possibile. Come conseguenza della proprietà di non ortogonalità dell'uplink, qualsiasi riduzione della interferenza media si traduce direttamente in un incremento di capacità.

2.1.3 – Gestione delle risorse in una sessione dati a pacchetto

Nel momento del Call setup, quando si stabilisce la connessione tra terminale è la RAN (radio-access network), l'RNC verifica che sia presente la quantità di risorse di cui il terminale ha bisogno durante la sessione.

Nel downlink, le risorse necessarie sono i channelization codes e la potenza in trasmissione. L'RNC è responsabile dell'allocazione delle risorse, esso conosce la frazione dell'albero del channelization code non riservata per ogni utente. Le misure nei NodiB forniscono all'RNC le informazioni circa la disponibilità media della potenza trasmissiva.

Nell'uplink, grazie alla natura non ortogonale, non c'è un limite per i channelization codes. Le risorse consistono nella quantità di interferenza addizionale che la cella può tollerare. Per quantificare tale risorsa sono spesso usati i termini *noise rise* o *rise-over-thermal*.

Accertatosi che ci siano sufficienti risorse sia in uplink che in downlink, l'RNC ammette il terminale alla cella e configura i canali fisici dedicati in entrambe le direzioni. L'RNC riserva le risorse corrispondenti al massimo data rate che il terminale può raggiungere durante la chiamata dati. Nel downlink, una frazione del dell'albero di codice deve essere riservata, quella corrispondente al più piccolo

fattore di spreading che può essere richiesto durante una sessione. In modo analogo, nell'uplink, l'RNC deve assicurare che la massima interferenza nella cella non sia superata quando il terminale trasmette al suo più elevato data rate.
Durante la chiamata dati a pacchetto, il data rate della trasmissione può variare in relazione al traffico. Come conseguenza, la potenza in trasmissione varia in funzione del data rate istantaneo. Comunque, la parte di albero di codice allocata per un determinato utente per il downlink rimane costante durante la chiamata a pacchetto.
La quantità di risorse riservate per ogni utente è abbastanza statica durante il periodo della chiamata dati a pacchetto. Ciò è adatto per servizi con un relativo data rate costante, per esempio servizi voce o trasmissioni video interattive, ma per servizi di dati a pacchetto che hanno una richiesta di risorse rapidamente variabile sono possibili notevoli miglioramenti (alcuni dei quali implementati nelle tecnologie HSPA).

2.2 – High-Speed Downlink Packet Access (HSDPA)

L'introduzione dell'High-Speed Downlink Packet Access, (HSDPA), implica una maggiore estensione della interfaccia radio del WCDMA, migliorandone le performance dal punto di vista del data rate di picco e della capacità. Questi risultati sono raggiunti utilizzando differenti tecniche, tra cui, modulazioni di ordine superiore, controllo del data rate, channel dependent scheduling, e Hybrid ARQ con soft combining.

2.2.1 – Trasmissione su canale condiviso

Una caratteristica fondamentale dell'HSDPA è l'utilizzo dello Shared-Channel Transmission. Lo Shared-channel transmission implica che una determinata frazione delle risorse totali disponibili in una cella per il downlink, ovvero i channelization codes e la potenza trasmessa nel caso di WCDMA, è vista come una risorsa comune che è dinamicamente condivisa tra più utenti, principalmente nel dominio del tempo. L'utilizzo dello shared-channel transmission, implementato attraverso l'High-Speed Downlink Shared Channel (HS-DSCH), permette di allocare rapidamente una grande frazione delle risorse di downlink per la trasmissione di dati di uno specifico utente. Questo procedimento è adatto per applicazioni che utilizzano dati a pacchetto che tipicamente hanno caratteristiche di intermittenza e quindi hanno una richiesta di

risorse che varia rapidamente.
La struttura base di un HS-DSCH in funzione del tempo e del codice sono riportati in Figura 8.

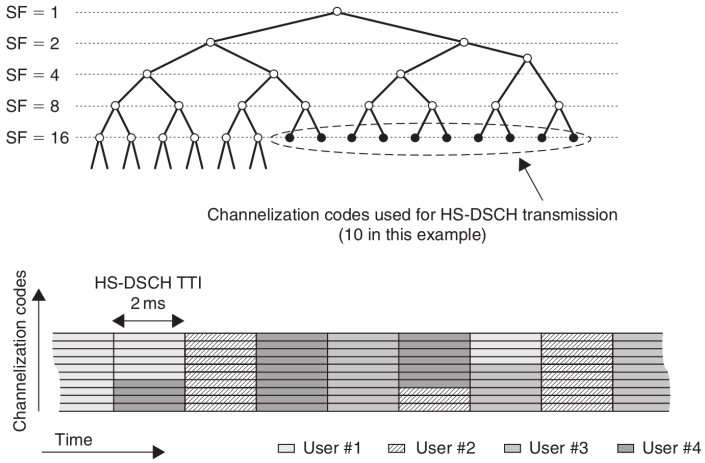

Figura 8: Struttura di un HS-DSCH nel tempo e in codice

Le risorse in termini di codici di un HS-DSCH consistono in un insieme di channelization codes con fattore di spreading pari a 16, dove il numero di codici disponibili per un HS-DSCH è configurabile da 1 a 15. I codici non riservati per la trasmissione su HS-DSCH sono utilizzati per altri scopi, per esempio sono legati ai segnali di controllo.
L'allocazione dinamica delle risorse di codice di un HS-DSCH per la trasmissione di uno specifico utente viene effettuata ogni 2ms che corrisponde al TTI base. L'uso di un così breve TTI per l'HSDPA riduce il ritardo e migliora il monitoraggio delle rapide variazioni del canale che sono sfruttate dal rate control e dal channel-dependent scheduling.

2.2.2 – Channel-dependent scheduling

Il processo di scheduling decide a quale utente assegnare la trasmissione sul canale condiviso in un determinato istante di tempo. La scheduler è un aspetto fondamentale e per larghe estensioni determina le prestazioni generali del sistema, specialmente quando la rete ha un elevato carico. In ogni TTI, lo scheduler decide quale utente o utenti dovrebbero trasmettere, e in stretta cooperazione con il rate control, a quale data rate.

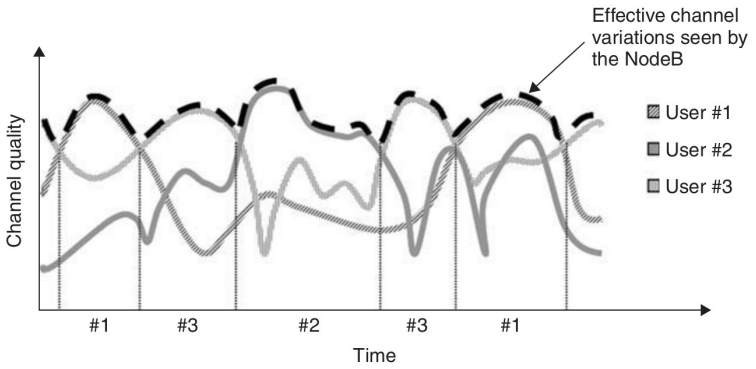

Figura 9: Channel-dependent scheduling per l'HSDPA

Siccome le condizioni radio per collegamenti radio di differenti terminali con una cella variano tipicamente in modo indipendente, ad ogni istante di tempo c'è sempre almeno un collegamento radio la cui qualità del canale è vicina al picco. Per un collegamento avere una buona qualità del canale significa che per quel collegamento può essere utilizzato un elevato data rate. Il miglioramento ottenuto nel trasmettere gli utenti che hanno una favorevole condizione del collegamento radio è conosciuto come multi-user diversity. I miglioramenti apportati dal Channel-dependent scheduling sono maggiori quanto più variano le condizioni del canale e quanto più cresce il numero degli utenti di una cella. Così, in contrasto con la visione tradizionale per cui il fading rapido è un effetto indesiderabile da combattere, con il channel-dependent scheduling il fading è potenzialmente favorevole e viene sfruttato.

In aggiunta alle condizioni del canale, lo scheduler tiene in considerazione anche le condizioni di traffico. Per esempio, non c'è motivo di assegnare risorse ad un utente che non ha dati in attesa di essere trasmessi, anche se le condizioni del canale sono favorevoli. Inoltre, alcuni servizi sono preferiti in quanto a priorità più alta. Ad esempio, per i servizi di streaming dovrebbero essere assicurati un data rate costante relativamente costante per un periodo di tempo abbastanza lungo, mentre per i servizi di background come ad esempio il download dei file, le richieste di risorse sono meno stringenti in termini di costanza del data rate a lungo termine.

2.2.3 – Rate control e higher-order modulation

Per l'HSDPA, il rate control è implementato variando dinamicamente il channel coding rate e selezionando dinamicamente il formato di modulazione tra QPSK e 16QAM. Le modulazioni di ordine superiore come la 16QAM permettono un

migliore utilizzo della banda rispetto alla modulazione QPSK, ma richiedono di ricevere una maggiore livello di energia media per bit. Di conseguenza, la 16QAM è principalmente utilizzata in condizioni vantaggiose del canale. Il data rate è selezionato indipendentemente per ogni 2ms (TTI) dal Nodo B.

2.2.4 – Hybrid ARQ con soft combining

La tecnologia Hybrid ARQ con soft combining permette al terminale di richiedere la ritrasmissione di transport blocks errati. Il terminale tenta di decodificare ogni transport block che riceve e rendiconta al Nodo B il successo oppure il fallimento dopo 5ms dalla ricezione del transport block. Ciò permette di effettuare rapide ritrasmissioni dei dati non ricevuti correttamente, riducendo significativamente il ritardo associato alle ritrasmissioni rispetto al caso dello standard WCDMA.

Il soft combining implica che i terminali che non riescono a decodificare un transport block non scartano l'informazione ricevuta (anche se non decodificabile), come avviene nei tradizionali protocolli ARQ. Tale informazione (definita “soft information”) viene utilizzata insieme con quella ritrasmessa per aumentare le probabilità di decodifica del blocco di informazione. Il soft combining utilizza anche l'IR (Incremental Redundancy), ovvero il transport block ritrasmesso contiene dei bit di parità che nella trasmissione originale non c'erano, per aumentare le probabilità di una corretta decodifica.

2.2.5 – Architettura

Le tecniche di base dell'HSDPA, che lo rendono più performante rispetto al WCDMA, mirano ad un veloce adattamento alle rapide variazioni delle condizioni del collegamento radio. Queste tecniche devono essere localizzate vicino alla interfaccia radio del lato rete che è il Nodo B.

Altro aspetto fondamentale è la minimizzazione dei cambiamenti della architettura utilizzata dagli standard precedenti, in modo da semplificare l'introduzione dell'HSDPA in reti già esistenti. L'HSDPA introduce un nuovo sotto livello MAC nel Nodo B, il MAC-hs che è responsabile delle operazioni di scheduling, del rate control e dell' Hybrid-ARQ. Quindi, fatta eccezione per i miglioramenti dell'RNC come quello per il controllo dell'accesso per gli utenti HSDPA, l'introduzione dell'HSDPA principalmente comporta modifiche solo sul Nodo B.

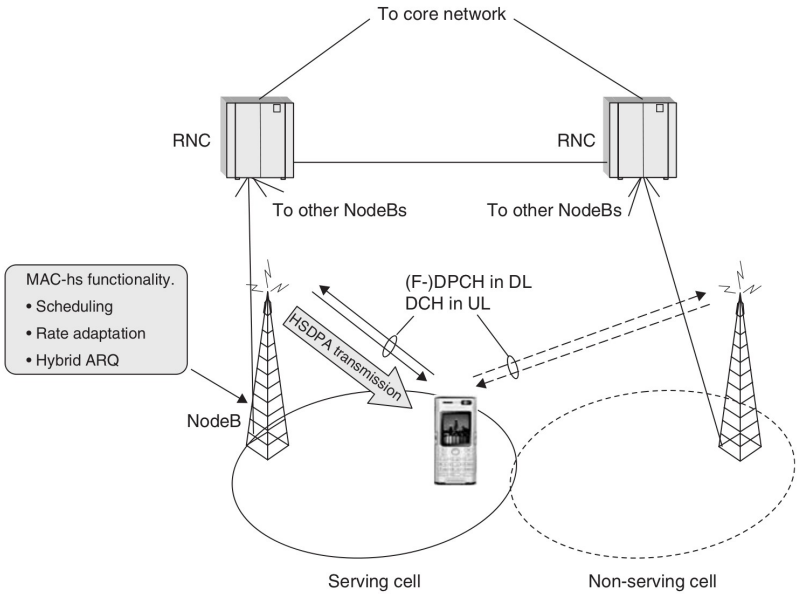

Figura 10: Architettura generale dell'HSDPA

L'HSDPA gestisce facilmente la mobilità da un cella che supporta HSDPA e una che non lo supporta e viceversa. Viene garantita la continuità del servizio, ovviamente il data rate cambia, l'RNC sposta il traffico utente da un canale HS-DSCH (canale condiviso) ad un canale dedicato collegato ad una cella non HSDPA.

2.3 – High-Speed Uplink Packet Access (HSUPA)

High-Speed Uplink Packet Access (HSUPA) anche conosciuto con il nome di Enhanced Uplink è stato introdotto nella release 6 di 3GPP. Esso fornisce miglioramenti nella capacità e nelle performance di uplink dello standard WCDMA. L'HSUPA è il complemento naturale dell'HSDPA, parlando di tutti e due gli standard si usa il termine HSPA.

2.3.1 – Differenze con l'HSDPA

Al centro dell'HSUPA ci sono due tecnologie di base utilizzate anche per l'HSDPA, il fast scheduling e il fast hybrid ARQ con soft combining. Per ragioni simili a quelle dell'HSDPA il TTI è fissato a 2ms. Questi miglioramenti sono implementati nel WCDMA attraverso un nuovo canale di trasporto, l'Enhanced Dedicated Channel (E-DCH).

Sebbene siano utilizzate le stesse tecnologie sia per l'HSDPA che per l'HSUPA ci

sono fondamentali differenze tra di loro.

- Nel downlink, le risorse condivise sono la potenza trasmessa e lo spazio dei codici, le quali sono entrambe collocate in un nodo centrale, il Nodo B. Nell'uplink, la risorsa condivisa è la quantità di interferenza permessa, che dipende dalla potenza trasmessa da più terminali distribuiti.
- In downlink lo scheduler e i buffer per le trasmissioni sono collocati nello stesso nodo, mentre in uplink lo scheduler è collocato nel Nodo B mentre i buffer dei dati sono distribuiti nei terminali. Quindi, i terminali hanno bisogno di un canale di controllo che segnali lo stato dei buffer allo scheduler.
- L'uplink del WCDMA e anche dell'HSUPA, è intrinsecamente non-ortogonale, e perciò soggetto ad interferenza tra trasmissioni in uplink effettuare da diversi terminali con la stessa cella. Questo è in contrasto con il downlink, dove i differenti canali trasmessi sono ortogonali. Il rapido controllo di potenza è essenziale in uplink per gestire il near-far problem.
- Il soft handover è supportato attraverso l'E-DCH. Ricevere dati da un terminale in più celle è di notevole beneficio, esso fornisce diversità. Il soft handover implica anche il controllo di potenza da più celle, e se necessario limita la quantità di interferenza generata dalle celle vicine, inoltre mantiene la retro compatibilità e la coesistenza con terminali che non usano l'E-DCH per la trasmissione dei dati.
- A differenza del downlink, l'uplink non ha la necessità di utilizzare modulazioni di ordine superiore. Nell'uplink non è necessario condividere i channelization codes tra gli utenti e i channel coding rates sono tipicamente più bassi del downlink.

2.3.2 – Scheduling

Per l'Hsupa, lo scheduler è un elemento chiave, esso controlla quando e a quale data rate un terminale è abilitato a trasmettere. Più alto è il data rate con cui il terminale sta trasmettendo e più alta deve essere la potenza ricevuta dal Nodo B per mantenere il livello di energia medio per bit richiesto per una corretta demodulazione. Incrementando la potenza trasmessa, il terminale può trasmettere ad un data rate maggiore. Comunque, causa la non ortogonalità dell'uplink, la potenza ricevuta da

una cella relativa ad un terminale, rappresenta interferenza per gli altri terminali. Quindi, la risorsa condivisa in uplink è la quantità di interferenza tollerabile da una cella. Se il livello di interferenza è troppo alto, alcune trasmissioni nella cella potrebbero non essere ricevute correttamente. Se il livello di interferenza è troppo basso ciò potrebbe indicare che le capacità del sistema non sono completamente sfruttate. Per quanto detto, lo scheduler ha il compito di fornire agli utenti il massimo data rate possibile senza che si superi il massimo livello di interferenza tollerabile nella cella.

A differenza dell'HSDPA, dove lo scheduler e i buffers di trasmissione sono entrambe collocati nel Nodo B, nel caso dell'uplink i dati da trasmettere risiedono sui terminali. Allo stesso tempo, lo scheduler è sempre nel Nodo B ed ha il compito di coordinare le trasmissioni dei terminali nella cella. Risulta necessario un meccanismo per comunicare ai terminali le decisioni dello scheduler e fornire informazioni sui buffer dei terminali allo scheduler. Tali informazioni nell'HSUPA sono basate su *scheduling grants*, inviate dallo scheduler nel Nodo B per controllare le trasmissioni dei terminali e *scheduling requests*, inviate dai terminali per richiedere risorse. Il data rate concesso dallo scheduler al terminale dipende dalla larghezza della scheduling grant. Lo scheduler controlla lo scheduling grant in ogni terminale per mantenere il livello di interferenza nella cella sotto la soglia massima tollerabile, tutto ciò è possibile mediante misure istantanee del livello di interferenza.

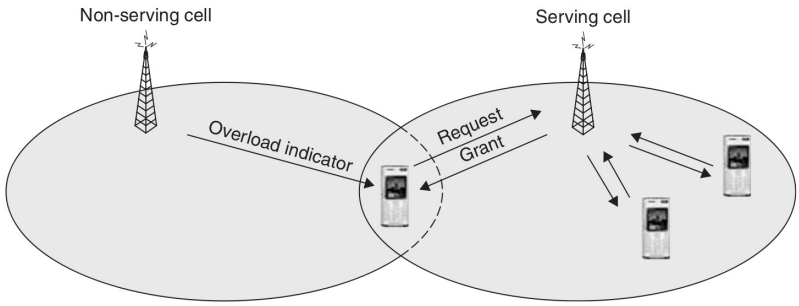

Figura 11: Struttura dello scheduling nell'HSUPA

Nell'HSDPA, tipicamente un singolo utente viene indirizzato ogni TTI. Nell'HSUPA la specifica implementazione dello scheduling fa sì che nella maggior parte dei casi vengano serviti più utenti in parallelo. La ragione sta nel fatto che la potenza trasmissiva di un terminale è molto più piccola di quella di un Nodo B, quindi per sfruttare a pieno la capacità del canale sono permesse trasmissioni a più utenti contemporaneamente. Tale comportamento ha come conseguenza che deve essere

valutata anche l'interferenza dentro la cella. Quindi, se lo scheduler ha permesso ad un terminale di trasmettere al massimo data rate basandosi su un accettabile valore di interferenza dentro la cella, questo potrebbe causare una interferenza non accettabile per le celle vicine. Per risolvere questo problema, nel soft handover, la cella servente ha le principali responsabilità delle operazioni di scheduling, ma i terminali monitorano anche le informazioni di scheduling provenienti da tutte le celle con cui il terminale è in soft handover (le celle che appartengono all'active set). Le celle non serventi possono richiedere a tutti gli utenti non serventi di abbassare il loro data rate nell' E-DCH trasmettendo un indicatore di overload in downlink. Questo meccanismo assicura la stabilità delle operazioni di rete.

2.3.3 – Hybrid ARQ con soft combining

Le metodologie Hybrid ARQ e soft combining sono usate dall'HSUPA generalmente per le stesse ragioni per cui sono usate dall'HSDPA, cioè per fornire robustezza contro occasionali errori di trasmissione. Per ogni transport block ricevuto in uplink, il Nodo B trasmette un singolo bit ai terminali per indicare il successo della decodifica (ACK), oppure la richiesta di ritrasmissione di un transport block erroneamente ricevuto (NACK).

Una differenza con ciò che accade nell'HSDPA scaturisce quando in uplink si sta utilizzando il soft handover. Quando un terminale è in soft handover, ciò implica che il protocollo hybrid ARQ è terminato in più celle. Di conseguenza, in molti casi, i dati trasmessi possono essere ricevuti con successo da alcuni Nodi B ma non da altri. Dal punto di vista del terminale è sufficiente che solo un Nodo B abbia ricevuto con successo i dati. Quindi, nel soft handover, tutti i Nodi B coinvolti tentano di decodificare i dati ricevuti e trasmettono un ack o un nack in relazione all'esito della decodifica. Se il terminale riceve almeno un ack considera i dati correttamente ricevuti dal nodo B e quindi provvede ad eliminare il transport block dal buffer.

Hybrid ARQ con soft combining può essere sfruttato non solo fornendo robustezza contro l'interferenza, ma anche per migliorare l'efficienza del collegamento incrementando la capacità e/o la copertura. Una possibilità per fornire un data rate di x Mbit/s è quella di trasmettere a x Mbit/s e impostare la potenza in trasmissione allo scopo di avere una bassa probabilità di errore nel primo tentativo di trasmissione.

Alternativamente, lo stesso data rate risultante può essere fornito trasmettendo n volte ad un data rate più alto senza aumentare la potenza in trasmissione e utilizzando il protocollo hybrid ARQ. Il secondo approccio porta ad un minore costo per bit, perché in genere i dati sono ricevuti correttamente prima di effettuare n ritrasmissioni. Questa tecnica è conosciuta con il nome di *early termination gain* e può essere vista come un implicito adattamento di data rate.

2.3.4 – Architettura

Per le stesse motivazioni dell'HSDPA, le funzionalità di scheduling e hybrid ARQ dell'HSUPA sono collocate nel Nodo B. È preferibile lasciare intatti i livelli dell'interfaccia radio sopra il MAC.

Viene inserita una nuova entità, il MAC-e, sia nel Nodo B che nel terminale. Nel Nodo B il MAC-e è responsabile per il supporto dell'hybrid ARQ e dello scheduling, mentre nel terminale, il MAC-e è responsabile per la selezione del data rate entro i limiti imposti dallo scheduler nel MAC-e del Nodo B.

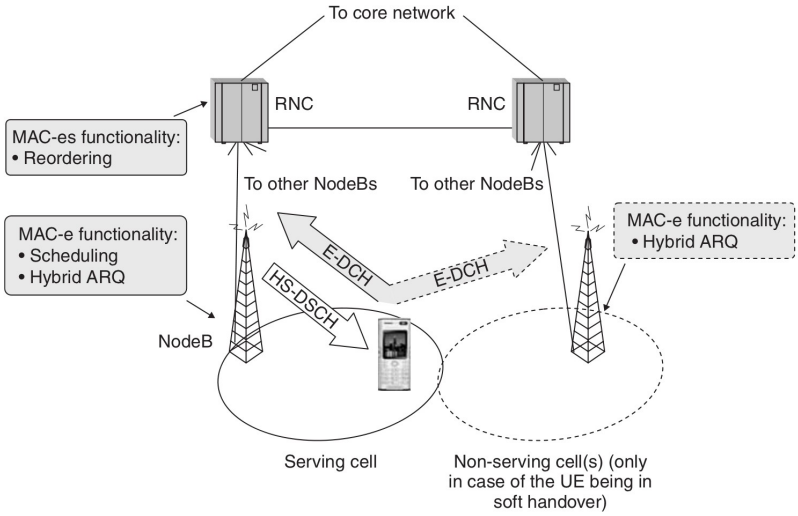

Figura 12: Architettura Hsupa

Quando il terminale è in soft handover con più Nodi B, differenti transport blocks possono essere decodificati in differenti Nodi B. Di conseguenza, un transport block potrebbe essere ricevuto con successo in un Nodo B mentre un altro NodoB è ancora impegnato nella ritrasmissione di un precedente transport block. Per tale motivo, per assicurare la consegna nel giusto ordine dei blocchi di dati al protocollo RLC, è richiesta una funzionalità di riordino nell'RNC sotto la forma di una nuova entità MAC, la MAC-es. Nel soft handover, sono usate più entità MAC-e per il terminale

quando i dati sono ricevuti da più celle. Comunque, le maggiori responsabilità di scheduling sono demandate al livello MAC-e della cella servente. I MAC-e delle celle non serventi hanno principalmente solo il compito della gestione del protocollo hybrid ARQ.

3

Modulo Wavecom

Per lo sviluppo del sistema di trasmissione dati con tecnologia HSPA si sono utilizzati due moduli Wavecom, il Q2687 e il Q26 extreme.

Figura 13: Q2687 e Q26 extreme

I moduli Wavecom sono essenzialmente la combinazione di un processore e un modem, da qui il nome di Wireless CPU.

La famiglia di Wireless CPU Q26 è una gamma di processori programmabili con capacità di comunicazione Wireless, designata per comunicazioni Wireless M2M (machine-to-machine). Si può scegliere tra connessioni GSM/GPRS, EDGE, WCDMA, HSxPA e CDMA 2000 versioni 1x - tutte con fattori di forma comuni.

Il Q26 Extreme opera con modalità duale 2G/3G che supporta fino a 7.2 Mbps HSDPA e 2 Mbps HSUPA con il passaggio automatico tra 2G/3G mentre il Q2687 opera fino alla modalità EDGE.

Dalla fine del 2008 la Wavecom è stata acquisita dalla Sierra Wireless, e i moduli Wavecom Q26 si trovano attualmente in commercio con la denominazione Sierra Wireless Air Prime Q26 series.

3.1 – Sierra Wireless Development Kit Q26

Per controllare i moduli Wavecom in ambito di sviluppo si utilizza il Sierra Wireless development kit Q26.

La board del kit di sviluppo è caratterizzata dalle seguenti interfacce: connettore esterno board to board e TP per accedere a tutti i segnali della Wireless CPU; bus dati parallelo; collegamento seriale principale RS 232, UART1 con tutti i segnali; indicatore di telefonata; collegamento seriale ausiliario RS 232, UART2 con 4 segnali; porta USB; testiera con 6 bottoni; connettori audio (AUDIO1, AUDIO2); leds per numerose indicazioni; pulsante di ESET; connettori di alimentazione;

commutatore ON/OFF; commutatore SWITCH; buzzer led; flash led; charger led. Nella Figura 14 è mostrata la board di sviluppo con le indicazioni dei componenti.

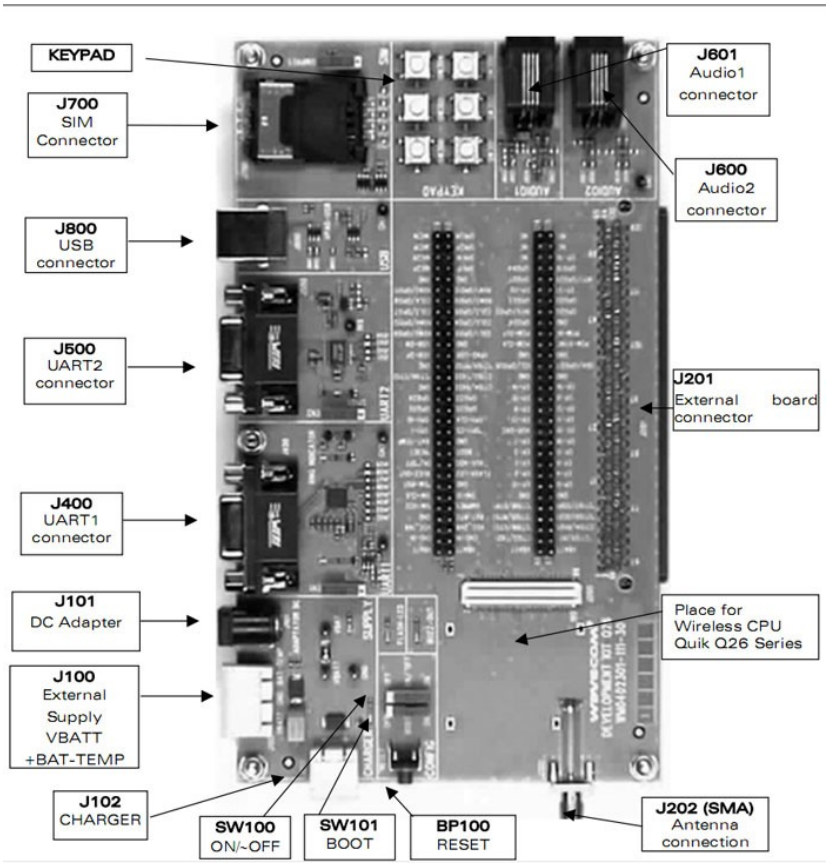

Figura 14: Board del Sierra Wireless Development Kit Q26

- J201 (external board connector): tutti i segnali della Wireless CPU sono connessi al connettore esterno della board (J201) e sono disponibili via TP al centro del kit di sviluppo Q26.
- Sono disponibili due sorgenti di alimentazione: (J100) alimentazione esterna DC; (J101) adattatore AC/DC.
- J102 è utilizzato per simulare un caricabatterie. Quando è utilizzato le batterie devono essere collegate.
- Flash LED lavora in due stati: se la Wireless CPU è OFF lampeggia; se la Wireless CPU è ON indica gli stati della rete.
- Buzzer LED è un indicatore costituito da un led di colore giallo che è controllato dal segnale. E' pilotato dal segnale PWM.
- Funzioni di controllo: ON/OFF; RESET: tale pulsante una volta premuto fa un reset generale. In genere è preferibile un reset del sistema operativo rispetto a quello hardware; BOOT: il commutatore è utilizzato solo per il

download di un nuovo software sulla Wireless CPU tramite UART1, con lo specifico software di download fornito da Wavecom.

- UART1 è la connessione seriale principale RS232 con la Wireless CPU (3.0V) sul development kit Q26. Di default, la UART1 è disponibile sul suo connettore dedicato J400.
- UART2 è la connessione seriale ausiliaria RS232 con la Wireless CPU (1.8V) sul development kit Q26. Di default, la UART2 è disponibile sul suo connettore dedicato J500.
- J700 SIM: socket sim standard a 1.8V e 3V.
- J800 connettore USB.
- J601 e J602: 2 interfacce audio costituite da connettori RJ9 a 4 pin.
- P200+J202: connettore RF. L'antenna è collegata alla board con un connettore SMA. Il cavo dell'antenna è dotato di un connettore FME. In dotazione al kit è fornito un adattatore SMA/FME che permette la connessione alla board.

3.2 – Open AT Software Suite

I moduli Wavecom hanno a disposizione dei tools di sviluppo raccolti in una suite, l'Open AT Software Suite.

L'Open AT Software Suite è una piattaforma per applicazioni cellular wireless e ottimizzata per comunicazioni M2M (machine to machine). La suite permette di usare la Wireless CPU sia come modem che come processore che esegue applicazioni che rispettano lo standard ANSI C.

La Open AT Software Suite è una suite proprietaria, sviluppata e mantenuta dalla Sierra Wireless. Essa include il software Wavecom che gira sulla Wireless CPU, drivers e tools di sviluppo. Componenti dell'Open AT Software Suite:

- **M2M Studio**: è l'ambiente di sviluppo integrato basato su Eclipse, permette di sviluppare, compilare, trasferire sulla wireless CPU, testare ed effettuare il debug della applicazione Open AT creata. L'applicazione Open AT è realizzata utilizzando differenti librerie (Open AT OS e Open AT plug-in libraries).
- **Open AT® OS**: Libreria che fornisce una ricca disposizione di API basate su linguaggio C, che permettono di sfruttare tutte le funzionalità della Wireless

CPU.

- **Open AT® Plug-Ins**: Set estensivo di librerie addizionali che offrono servizi supplementari, di solito servono per controllare moduli aggiuntivi.
- **Open AT® Firmware**: Software Wavecom che gestisce le comunicazioni wireless (GSM/GPRS/EDGE/HSPA) e l'interfaccia a comandi AT della Wireless CPU.
- **Expresso**: è un tool semplice ed intuitivo per Windows che permette una veloce configurazione delle principali caratteristiche del modem della Wireless CPU.
- Driver per il controllo da PC con Windows e altri sistemi operativi.

Per lo sviluppo del sistema di trasmissione dati a cui è mirato questo studio, il modulo Wavecom è utilizzato soltanto come modem. L'applicazione della suite Open AT che maggiormente ci interessa è la Open AT Firmware. Per gestire tale applicazione è necessario utilizzare i Comandi AT.

3.3 – Comandi AT

La maggior parte dei modem fonici utilizza i **comandi AT Hayes**, uno specifico insieme di comandi originalmente sviluppato per il modem *Hayes Smartmodem*.
Ogni funzione del modem è governata dal relativo comando AT (che sta per *ATtention*, attenzione). Per inviare un comando occorre trasmettere sulla porta seriale del modem una stringa ASCII formata da AT seguito da uno o più comandi. La trasmissione di dati su porta seriale può avvenire con un comune terminale di comunicazione seriale, ne esistono varie versioni per differenti sistemi operativi.
I comandi AT riconosciuti dal modulo Wavecom sono molti. Una parte di essi sono comandi AT generici utilizzati da qualsiasi modem, mentre la restante parte sono comandi AT proprietari che sono validi esclusivamente per le Wireless CPU.
L'insieme dei comandi AT possibili per la Wireless CPU sono contenuti in un documento denominato “AT Commands Interface Guide for Firmware x.xx” dove al posto delle x vi è il numero di versione (attualmente la più recente è la 7.43).
I comando AT riconosciuti dal modulo Wavecom sono di tre tipologie:

- Action Command: utilizzato per assegnare una configurazione;

- Read Command: utilizzato per verificare quale configurazione si sta utilizzando;
- Test Command: utilizzato per verificare le configurazioni che sono possibili per il comando AT scelto.

Non tutti i comandi AT hanno tutte e tre le tipologie di formato.
In tabella 1 è riportato un esempio effettuato su una selezione di comandi AT interessanti, la maggior parte dei quali mirati allo sviluppo del sistema di trasmissione dati in esame.

COMANDO	**RISPOSTA**
AT	OK
AT+CSQ	+CSQ: 9,0 OK
AT+COPS?	+COPS: 0,2,22299,2 OK
AT+COPN	... +COPN: 22201, "I TIM" +COPN: 22210, "vodafone IT" +COPN: 22288, "I WIND" +COPN: 22299, "3 ITA" ... OK
AT+WOPN=0,22299	+WOPN: 0,"3 ITA" OK
AT+CREG=2	+CREG: 1,"C390","00411B8D",2 OK
AT+WWSM?	+WWSM: 2,0 OK
AT+WWSM=1	OK
AT+WGPRS=9,2	+WGPRSIND: 3 OK

Tabella 1: Esempio di comunicazione con comandi AT

Comando AT+CSQ

Questo comando è utilizzato per leggere l'indicazione della potenza del segnale ricevuto e il bit error rate del canale.

Sintassi comando:	AT+CSQ
Sintassi risposta:	+CSQ: <rssi>,<ber> OK
Valori predefiniti:	<rssi>: received signal strength 0 -113 dBm or less 1 -111 dBm 2 to 30 -109 to –53 dBm 31 -51dBm or greater 99 not known or not detectable <ber>: channel bit error rate 0...7 99 not known or not detectable

Tabella 2: Il comando AT+CSQ

Con le chiamate dati l'intensità del segnale è particolarmente importante. Il comando AT+CSQ ci restituisce il valore della qualità del segnale misurato (SQM). Non c'è uno standard riguardante tale misurazione, comunque la maggior parte dei produttori riportano il CSQ in una scala compresa tra 0 e 31.

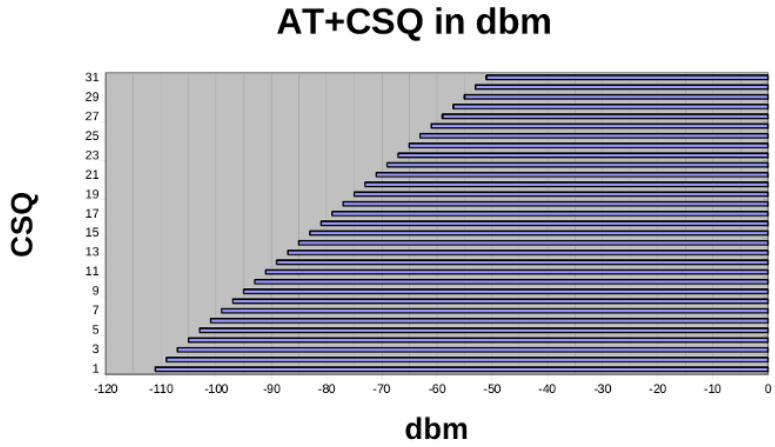

Figura 15: AT+CSQ conversione in dBm

Per convertire il CSQ in dBm, si deve assumere la seguente relazione: *dBm = -113 +n x 2, dove n è il CSQ.*

Riguardo l'esempio in tabella 1, la stringa ”+CSQ: 9,0” ha il seguente significato:

- rssi pari a 9 che corrisponde a-95 dBm;
- ber compreso nel più piccolo intervallo di valori riscontrabile.

Comando AT+COPS

Il comando AT+COPS gestisce la registrazione da parte dell'utente alla rete dell'operatore mobile. Tale registrazione può avvenire in tre modi: manuale, automatica oppure manuale la prima volta e se non ha esito positivo tenta in modo automatico.

Sintassi comando:	AT+COPS?
Sintassi risposta:	+COPS: <mode>[,<format>,<oper>[,<AcT>]] OK
Valori predefiniti:	<mode>: network registration mode 0 automatic (default value) 1 manual 4 manual/automatic <format>: format of <oper> field 0 long alphanumeric format <oper> 1 short alphanumeric format <oper> 2 numeric <oper> (default value) <oper>: operator identifier (MCC/MNC) <AcT>: access technology selected 0 GSM 2 UTRAN

Tabella 3: Il comando AT+COPS?

Riguardo l'esempio in tabella 1, la stringa di output "+COPS: 0,2,22299,2" corrisponde a:

- modalità di registrazione alla rete automatica;
- formato del campo operatore numerico;
- campo operatore 22299;
- tecnologia di accesso selezionata UTRAN.

Il valore del campo <AcT> è molto importante, in quanto ci dice se la rete a cui si è registrati permette di effettuare connessioni GSM, EDGE (tramite GERAN) oppure UMTS, HSPA (tramite UTRAN).

Comando AT+COPN

Il comando AT+COPN fornisce il output la lista di tutti gli identificatori degli operatori di rete e le corrispondenti denominazioni. Utile per vedere il nome dell'operatore di rete a cui si è registrati.

Comando AT+WOPN

Il comando AT+WOPN fornisce una stringa di informazioni sull'operatore di rete sulla base dell'identificatore numerico in input.

Comando AT+CREG

Questo comando è usato dalla applicazione per verificare lo stato di registrazione del modulo Wavecom.

Sintassi comando:	AT+CREG=<mode>
Sintassi risposta:	+CREG: <mode>,<stat>[,<lac>,<cid>[,<AcT>]] OK
Valori predefiniti:	<mode>: request operation 0 disable network registration unsolicited result code (default value) 1 enable network registration code result code +CREG: <stat> 2 enable network registration and location information unsolicited result code +CREG: <stat>,<lac>,<ci>,<AcT> if there is a change of network cell <stat>: network registration state 0 not registered, ME is not currently searching for a new operator 1 registered, home network 2 not registered, ME currently searching for a new operator 3 registration denied 4 unknown 5 registered, roaming <lac>: location area code string type; two byte location area code in hexadecimal format (e.g. "00C3" equals 195 in decimal) <ci>: cell ID string type ; four byte UTRAN/GERAN cell ID in hexadecimal format <AcT>: access technology of the registered network 0 GSM 2 UTRAN

Tabella 4: Il comando AT+CREG

Riguardo l'esempio in tabella 1, il comando “AT+CREG=2” e l'output "+CREG: 1,"C390","00411B8D",2" corrispondono a:

- il comando azione attiva la risposta non sollecitata che fornisce numerose

informazioni sulla registrazione alla rete;

- si è registrati alla rete home;
- l'area code è la stringa C390;
- il cell ID è il numero esadecimale 00411B8D;
- la tecnologia di accesso alla rete registrata è UTRAN.

Con il precedente comando il modulo invia periodicamente la risposta non sollecitata e quindi è possibile tenere sotto controllo i cambiamenti dei dati relativi alla registrazione della rete.

Comando AT+WWSM

Il comando AT+WWSM è proprietario Sierra Wireless ed è usato dalla applicazione per selezionare la tipologia di rete con cui deve operare. Tale comando può essere anche utilizzato per verificare con che tipo di rete si sta operando. Per applicare le modifiche apportate da questo comando è necessario un reset del sistema, utilizzando il pulsante sulla board di sviluppo oppure togliendo e ripristinando l'alimentazione.

Sintassi comando di read:	AT+WWSM?
Sintassi risposta di read:	+WWSM: <mode>[,<prefer_nwk>] OK
Sintassi comando di action:	AT+WWSM=<mode>[,<prefer_nwk>]
Risposta comando di action:	OK
Valori predefiniti:	<mode>: cellular network to operate 0 GSM Digital Cellular Systems (GERAN only) 1 UTRAN only 2 3GPP Systems (both GERAN and UTRAN) (default) <prefer_nwk>: preferred network (when <mode>=2) If omitted, the default value '0' will be used. 0 automatic (default) 1 GERAN preferred 2 UTRAN preferred

Tabella 5: Il comando AT+WWSM

Riguardo l'esempio in tabella 1, il primo comando (di read) ha ottenuto come risposta l'output “+WWSM: 2,0”, ovvero:

- sono utilizzabili i sistemi 3GPP (sia GERAN che UTRAN);
- il tipo di rete preferito è automatico.

Il secondo comando digitato, “AT+WWSM=1”, imposta la registrazione a sole reti di accesso UTRAN.

Comando AT+WGPRS

Il comando AT+WGPRS permette di modificare alcune impostazioni relative ai parametri GPRS della Wireless CPU. E' un comando proprietario. Inoltre tale comando ci fornisce informazioni riguardanti le tecnologie disponibili della rete a cui si è registrati.

Sintassi comando:	AT+WGPRS=9,2
Sintassi risposta:	+WGPRSIND: <techno>
Valori predefiniti:	<techno>: technologies available on the used network 0 neither GPRS nor EGPRS features supported 1 only GPRS feature supported 2 EGPRS feature supported 3 3G Release 99 supported 4 HSDPA supported 5 HSUPA supported 6 HSUPA and HSDPA supported

Tabella 6: Il comando AT+WGPRS

Riguardo l'esempio in tabella 1, la stringa di output "+WGPRSIND: 3 " corrisponde a:

- la rete a cui si è registrati supporta le connessione dati fino allo standard definito dalla Release 99 dell'ente 3GPP, ovvero il WCDMA con velocità massima effettiva in download pari a 384 Kbps.

4

Fox Board LX832

Per realizzare il sistema di comunicazione dati in esame è necessario un sistema di elaborazione che controlli il modulo Wavecom e ne stabilisca la connessione ad internet sfruttando la tecnologia HSPA. Per gli scopi sopra descritti si utilizza la Fox Board lx 832.

La FOX Board lx832 è un sistema Linux completo in soli 66x72mm. La FOX Board esegue un vero sistema operativo linux su un microprocessore ETRAX 100LX, una CPU 100MIPS RISC realizzata dalla Axis Communications.

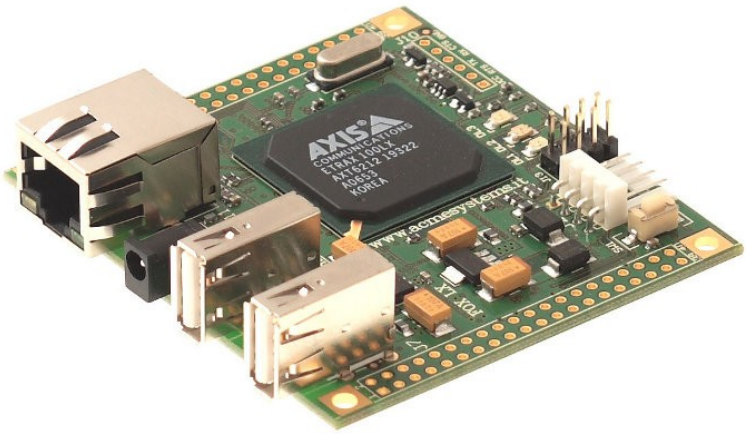

Figura 16: Fox Board LX832

La FOX Board ha due principali campi di applicazione:

- periferica stand alone per realizzare micro web server o altri dispositivi di rete come proxy, router, firewall, etc;
- core da posizione su circuiti stampati di board per applicazioni utente invece di un semplice microcontroller.

La fox board è dotata di:

- due interfacce USB 1.1 che possono essere connesse a USB memory stick, hard disk, webcam, modem, Wi-Fi o Bluetooth dongle, ADSL adapter, Serial converter, etc;
- un'interfaccia ethernet 10/100 con la quale è possibile comunicare con la board, attraverso il completo stack protocollare TCP/IP, tramite i servizi di:

Web server, FTP server, SSH e Telnetd;

- due 40 pins sockets sono disponibili per il collegamento ad altre board. Su questi pin sono disponibili molti segnali.

Specifiche software e hardware

Specifiche software	
Kernel	Up to version 2.6.19
Server	HTTP (WEB), FTP, SSH, TELNET
Driver	USB Pen driver, FTDI and PROLIFIC USB to Serial Converter
SDK	Fully Open Source and freely downloadable Software Development Kit
Language	C, C++, PHP, PYTHON, etc
Specifiche hardware	
Size	66 x 72 mm (2.6 x 2.8 inches)
Cpu	100MIPS Axis ETRAX 100LX 32 bit, RISC, 100MHz
Memory	8MB FLASH 32MB RAM
Power	Single power supply 5 Volt 280mA (1 watt)
Ports	1 Ethernet (10/100 Mb/s) 2 USB 1.1 1 serial console port
Extension	2 extension sockets with IDE, SCSI, serial lines, parallel ports, I/O lines, I2C bus interface
Weight	37 gr
Temperature range	0-70 °C

Descrizione componenti

J6, J7: socket di estensione per altre board o device. Forniscono segnali per I/O, I2C, SPI, porte seriali e parallele. Non tutte le interfacce possono essere usate contemporaneamente.

J2, J14: alimentazione 5V DC. Ci sono tre metodi per alimentare la board, attraverso J2 con un connettore uguale a quello per i floppy, attraverso J14 con un alimentatore oppure da J6, J7 ricevendo l'alimentazione da altri moduli collegati.

SW1: tenendo premuto questo pulsante durante l'accensione si ripristinano le impostazioni iniziali della cartella /etc.

SW2: è composto, partendo dall'alto dagli switch 2.1, 2.2, 2.3, 2.4. Lo switch 2.1 se chiuso spegne la Fox. Lo switch 2.2 se chiuso resetta la cpu. Lo switch 2.3 se chiuso durante l'accensione permette il flashing tramite porta seriale. Lo switch 2.4 se chiuso durante l'accensione abilita il flashing tramite LAN.
J10: ttys0 porta seriale usata come porta di console di sistema (115200 baud, 8N1, 3.3V)

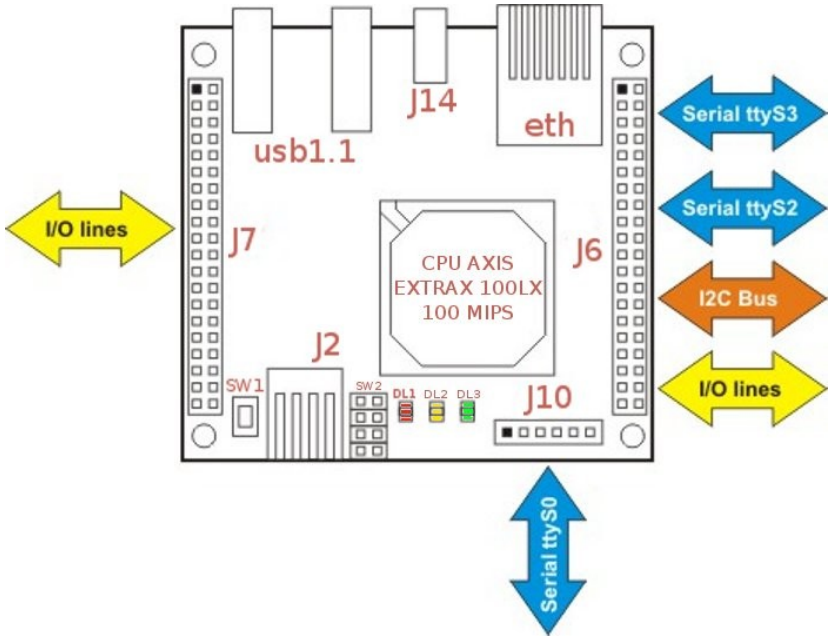

Figura 17: Componenti Fox Board LX832

DL1 RED: il suo stato normale è in off. Durante l'accensione è utilizzato dal kernel per indicare uno stato iniziale di errore della board. Lampeggia se non è stato configurato l'indirizzo MAC dell'adattatore ethernet.
DL2 YELLOW: mostra il traffico LAN attraverso l'adattatore ethernet.
DL3 GREEN: è connesso direttamente all'alimentazione. Mostra se la board è accesa.
Alla FOX Board possono essere connesse molte periferiche:

- fino a 48 general purpose I/O;
- fino a 3 porte seriali asincrone;
- I2C bus;
- fino a 2 porte parallele e 1 porta parallela wide;
- fino a 4 porte IDE ovvero 8 periferiche.

In aggiunta si deve osservare che non tutte le periferiche e le porte possono essere

collegate contemporaneamente.
Inoltre porte seriali disponibili in realtà sono 4. La /dev/ttys0 fornita da J10, le /dev/ttys2 e /dev/ttys3 fornite da J6 e la /dev/ttys1 che è in comune con la porta USB1 (utilizzabile per convertitori seriali).

4.1 – Installazione e uso del Phrozen Sdk

Il Phrozen SDK è una collezione di programmi Open Source utile per sviluppare applicazioni, installare driver di periferiche e ricompilare l'intero kernel linux che girano sulla Fox Board lx832. Questo sdk è fortemente basato sullo sdk standard fornito dall'Axis. ma include un impressionante numero di patches, applicazioni ed utility create da John Crispin e da altri sviluppatori della fox board.
La procedura indicata in questo paragrafo permette di installare il Phrozen SDK per la FOX Board LX832 su un sistema operativo linux, necessario per cross-compilare applicazioni per la Fox Board.

Requisiti di sistema

I requisiti base cui necessita il sistema per utilizzare il Phrozen SDK sono:

- Linux, qualsiasi nuova distro dovrebbe funzionare;
- una interfaccia di rete ethernet;
- una connessione ad internet attiva;
- privilegi di root;
- GCC C compiler, CRIS cross-compiler, GNU make, GNU wget, Subversion, awk (o gawk), bc, byacc (o yacc se byacc è un collegamento ad esso), lex o flex, perl, sed, tar, zlib, md5sum, pmake, curses or ncurses, bison, which.

Installazione su Debian Lenny

Per raggiungere i requisiti di sistema devono essere installati alcuni pacchetti. Digitare (bisogna autenticarsi come root):

```
# apt-get install subversion
# apt-get install pmake
```

A questo punto è necessario scaricare il pacchetto Cris compiler. Dopo aver scaricato il pacchetto, posizionarsi nella cartella in cui è stato scaricato e per installarlo

digitare da un terminale:

```
# dpkg –i cris-dist_1.63-1_i386.deb"
```

Creare una directory di lavoro, ad esempio */home/fox*. Scaricare il file *install_svn_sdk.sh* nella directory creata. Renderlo eseguibile e avviare l'installazione. Ciò avviene digitando:

```
# chmod +x install_svn_sdk.sh
# ./install_svn_sdk.sh
```

Una volta che il processo è terminato, posizionarsi nella cartella di installazione, nell'esempio */home/fox*, accedere alla cartella *devboard-R2_01* e avviare il menu di configurazione. Il menu si avvia digitando:

```
# cd devboard-R2_01
# make menuconfig
```

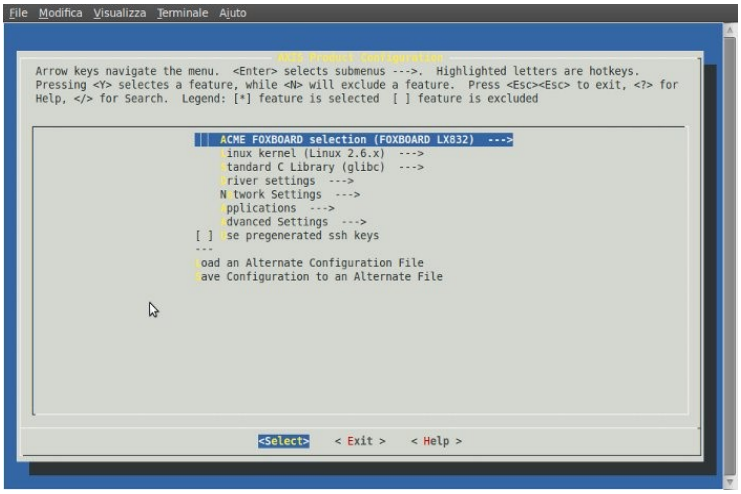

Figura 18: Menu di configurazione del Phrozen SDK

Mediante questo menu di configurazione si possono personalizzare una grande varietà di opzioni per la fox board, quali la versione del kernel, i driver, le applicazioni e molto altro.

Una volta creata la propria configurazione, in base all'utilizzo che si deve fare della fox, si genera una fimage (fox image) che successivamente deve essere trasferita sulla board.

Per generare la propria fimage si deve solo salvare quando si esce dal menu di configurazione e successivamente digitare:

```
# ./configure
```

```
# make
```

Successivamente la nuova fimage generata deve essere trasferita sulla Memoria Flash della Fox Board.

4.2 – Reflash della Fox Board

La fox board ha una memoria flash dove si possono memorizzare tutti i file dei programmi necessari per il suo corretto funzionamento. Su questa memoria sono memorizzati il sistema operativo, tutte le applicazioni e i dati di cui la fox ha bisogno, come un hard disk per un PC.
Per creare una nuova fimage dai sorgenti è necessario utilizzare il Phrozen sdk. Ogni Fox boards è rilasciata con una immagine linux preinstallata e pronta all'uso. Si rende necessario talvolta, in relazione all'uso della fox, cambiare tale immagine. Se si ha bisogno di cambiare l'immagine di default, con una personalizzata generata con l'sdk, sono possibili diversi metodi.
Una cosa importante da conoscere è che *è sempre possibile ripristinare una fox board dopo aver caricato una immagine non funzionante*, caratteristica non banale in fase di sviluppo.

Reflash mediante LAN

Questo metodo richiede che il PC e la Fox Board siano sulla stessa LAN. Posizionarsi nella cartella di installazione del SDK, (nell'esempio /home/fox/devboard-R2_01/) e digitare:

```
# . init_env
# boot_linux –F –i image_filename
```

Una volta lanciato boot_linux si otterrà un messaggio come questo:

```
Using internal boot loader: INTERNAL_NW - Network boot (default).
Starting boot...
We're doing a flash write, this may take up to a few minutes...
```

A questo punto si deve spegnere la fox, chiudere il jumper sw2.4 (abilita il flashing tramite ethernet), riaccendere la fox.
Se tutto va bene, si dovrebbe vedere il processo di trasferimento della immagine nella Flash memory della fox board, con dei messaggi come i seguenti:

```
...
0x80360000: Writing 0x00010000 bytes
0x80370000: Writing 0x00010000 bytes
0x80380000: No need to write
0x80390000: No need to write
0x803a0000: Writing 0x00010000 bytes
0x80000000: Verifying...OK
JUMP
0x00000000
END
Exiting with code 0
```

Durante il procedimento il led rosso (DL1) rimane acceso per tutto il tempo. Alla fine della programmazione la fox board si riavvia.
Terminato il reflash è necessario rimuovere il jumper sw2.4.

Reflash mediante FTP

La fox board è dotata di un server FTP abilitato a trasferire file in ingresso e in uscita dal proprio file system. Ci sono due speciali file che, se correttamente usati, abilitano la completa riprogrammazione della memoria Flash. Questo permette, ovviamente di aggiornare il firmware del prodotto anche da remoto tramite una semplice connessione TCP/IP. Questa procedura può avvenire in due differenti modalità, attraverso l'utilizzo di due diversi file.

- *flash*: permette l'aggiornamento della partizione di sola lettura della memoria flash che contiene il kernel, lasciando intatta la partizione di lettura/scrittura dove sono contenuti i file personali.
- *flash_all*: permette di riscrivere completamente la memoria flash della fox. Tutti i dati trasferiti prima del flash_all saranno cancellati in modo definitivo.

E' possibile utilizzare un qualsiasi ftp client e connettersi alla fox con le credenziali: username *root*,e password *pass*. Durante il procedimento il led rosso (DL1) rimane acceso per tutto il tempo.

Reflash mediante WEB

Nella web page di default disponibile nella fox board ci sono due link che permettono di effettuare l'upload della nuova immagine nella memoria flash della Fox Board:

- Update FOX firmware - system area: questo link permette l'aggiornamento della partizione di sola lettura della memoria FLASH che contiene il kernel, lasciando intatta la partizione di lettura/scrittura dove sono contenuti i file

personali.

- Update FOX firmware - entire system : questo link permette di riscrivere completamente la memoria flash della fox. Tutti i dati trasferiti prima del flash_all saranno cancellati in modo definitivo.

Bisogna soltanto cliccare sul link preferito e selezionare il file che contiene l'immagine da memorizzare nella flash memory. Durante il procedimento il led rosso (DL1) rimane acceso per tutto il tempo.

4.3 – Compilare applicazioni in C per la Fox Board

Andare nella directory *devboard-R_01*, poi digitare:

```
/home/fox/devboard-R_01# . init_env
Prepending "/..../devboard-R2_01/tools/build/bin" to PATH.
Prepending "/usr/local/cris/bin" to PATH.
/home/fox/devboard-R_01# cd apps
```

Creare una nuova directory con lo stesso nome della applicazione:

```
/home/fox/devboard-R_01/apps# mkdir prova
/home/fox/devboard-R_01/apps/prova# cd prova
```

Scrivere il codice sorgente e salvarlo con il nome *prova.c:*

```
#include "stdio.h"
int main(void) {
  printf("Prova !\n");
  return 0;
}
```

Creare il seguente file e salvarlo con il nome *Makefile:*

```
AXIS_USABLE_LIBS = UCLIBC GLIBC
include $(AXIS_TOP_DIR)/tools/build/Rules.axis
PROGS = prova
all: $(PROGS)
$(PROGS): $(PROGS).o
        $(CC) $(LDFLAGS) $^ $(LDLIBS) -o $@
clean:
        rm -f $(PROGS) *.o core
```

Per scegliere gnuuclibc come compilatore digitare:

```
/home/fox/devboard-R_01/apps/prova# make cris-axis-linux-gnuuclibc
make[1]: Entering directory `/home/devboard-R2_01/apps/prova'
make[1]: Leaving directory `/home/devboard-R2_01/apps/prova'
```

Con il comando precedente viene creato nella directory *prova* un file nascosto con

estensione *.target-makefrag* che contiene le istruzioni per il comando *make.*

Per compilare digitare:

```
/home/fox/devboard-R_01/apps/prova# make
 gcc-cris  -isystem /home/devboard-R2_01/target/cris-axis-linux-gnu/
include
   -mlinux -mno-mul-bug-workaround -Wall -Wshadow -O2 -g    -c -o
prova.o prova.c
 gcc-cris  -isystem /home/devboard-R2_01/target/cris-axis-linux-gnu/
include
   -mlinux -mno-mul-bug-workaround  -L/home/devboard-
R2_01/target/cris-axis-linux-gnu/lib
   -Wl,-rpath-link,/home/devboard-R2_01/target/cris-axis-linux-gnu/
lib prova.o  -o prova
```

In caso di corretta esecuzione, nella cartella prova si trova il file eseguibile pronto per girare sulla FOX board:

```
/home/fox/devboard-R_01/apps/prova# ls -l
 total 40
 -rwxr-xr-x  1 root root 17684 Oct 26 17:45 prova
 -rw-r--r--  1 root root    80 Mar  3  2005 prova.c
 -rw-r--r--  1 root root 11868 Oct 26 17:45 prova.o
 -rw-r--r--  1 root root   478 Oct 26 16:54 Makefile
```

A questo punto per far girare il programma, esso deve essere trasferito sulla Fox Board. Il modo più semplice è quello di utilizzare SCP:

```
/home/fox/devboard-R_01/apps/prova# scp prova
root@192.168.0.90:/mnt/flash
The authenticity of host '192.168.0.90 (192.168.0.90)' can't be
established.
RSA key fingerprint is
89:2f:6e:f7:5b:d0:02:07:d3:9d:18:10:76:6e:99:0c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.90' (RSA) to the list of known
hosts.
root@192.168.0.90's password: pass
```

Digitare:

```
# telnet 192.168.0.90
Entering character mode
Escape character is '^]'.
axis-00408c012220 login: root
Password: pass

[root@axis-00408c012220 /root]182# cd /mnt/flash
[root@axis-00408c012220 /mnt/flash]182#
```

Per rendere eseguibile il programma, si deve digitare (va fatto solo per la prima esecuzione):

```
[root@axis-00408c012220 /mnt/flash]182# chmod +x prova
```

Per lanciare il programma digitare:

```
[root@axis-00408c012220 /mnt/flash] # ./prova
Prova !
```

4.4 – Collegamento 2G/3G con PPP

Per fare una connessione TCP/IP sfruttando la rete 2G/3G (la tecnologia di accesso dipende dalla disponibilità della rete), è richiesto l'utilizzo di PPP (point to point protocol) che viene gestito dai sistemi linux con PPPd (poin to point protocol daemon).

4.4.1 – PPP e PPPD

Il protocollo ***PPP (Point to Point Protocol)*** fornisce un metodo standard per il trasporto di diversi tipi di protocollo su una connessione punto punto.
Le caratteristiche principali di PPP sono:

- supporto multi protocollo;
- supporto all'autenticazione;
- riconoscimento errori;
- supporto all'indirizzamento IP dinamico.

I principali componenti del protocollo PPP sono tre:

- Un metodo di *incapsulazione* per i datagrammi di diversi protocolli;
- *LCP (Link Control Protocol)* per stabilire, configurare e mantenere sotto controllo la connessione;
- *NCP (Network Control Protocol)* per configurare i diversi protocolli network a livello rete che vengono trasportati.

L'*incapsulazione* permette di trasportare simultaneamente più protocolli sullo stesso collegamento, questo processo è stato progettato in modo da garantire la compatibilità tra hardware di vendor differenti. Il formato dei frame PPP è simile a quello utilizzato da HDLC (High-level Data Link Control).
Il *link control protocol* permette di avere a disposizione un metodo per tenere sotto controllo e quindi gestire la connessione PPP. LCP oltre a stabilire e terminare la comunicazione si occupa anche dell'autenticazione dei peer e del monitoraggio del suo corretto funzionamento.

Con il *network control protocol* è possibile avere un protocollo di controllo per ogni livello rete supportato. Ncp si occupa quindi di negoziare le opzioni del livello rete, come per esempio l'attribuzione dell'indirizzo IP utilizzato poi anche da altri protocolli TCP/IP trasportati.

Le fasi di una connessione PPP sono tipicamente quattro:

- *Definizione della connessione (Link Establishment Phase)*: la connessione viene stabilita dal Link Control Protocol il quale provvede a scambiare dei pacchetti di configurazione tra i due host che si mettono in contatto. La negoziazione avviene per i valori differenti da quelli predefiniti. Terminata questa fase la connessione è stabilita.
- *Autenticazione (Authentication Phase):* in alcuni casi prima di scambiare pacchetti a livello rete, l'host che si connette deve essere autenticato. Per default l'autenticazione non è obbligatoria su una connessione PPP, ma se si decide di utilizzarla essa deve avvenire al più presto. Nessuno scambio a livello rete può avvenire prima che l'autenticazione sia completata. I protocolli comunemente utilizzati per l'autenticazione sono *PAP (Password Authentication Protocol)* e *CHAP (Challange Handshake Autentication Protocol).*
- *Configurazione Protocollo di rete (Network-Layer Protocol Phase):* in questa fase ogni protocollo di rete viene separatamente configurato tramite il proprio Network Control Protocol.
- *Terminazione della connessione (Link Termination Phase):* la connessione PPP può terminare in qualsiasi momento per differenti motivi: caduta della portante, fallimento dell'autenticazione, decadimento della qualità della linea, scadenza del tempo di inattività (idle-time) o chiusura da parte di un amministratore.

Classico esempio di utilizzo del protocollo PPP è la connessione ad Internet tramite modem da parte di un PC. Questo protocollo è altre sì utilizzato per connessioni router to router su linee dedicate.

PPPD (Point to Point Protocol Daemon) è il demone che si occupa di stabilire una connessione tramite il protocollo PPP, viene usato sia come client, per collegarsi ad

un server remoto, che come server, che autentica i client che si collegano.
La connessioni PPP su un sistema Linux vengono stabilite grazie a pppd che si trova in */usr/sbin/pppd*. Questo server fornisce il supporto base per LCP, per l'autenticazione e per NCP riguardo alla configurazione IP. L'incapsulazione è invece fornita da routine presenti nel Kernel del sistema operativo.

4.4.2 – Configurazione PPPD sulla Fox Board

Dapprima è necessario verificare che pppd sia presente sulla fox board. A tale scopo digitare:

```
# ls /usr/sbin/pppd
```

A tal punto si possono verificare due situazioni:

`/usr/sbin/pppd`	pppd è installato
`ls: /usr/sbin/pppd: No such file or directory`	pppd non è installato

Se pppd non è sulla fox board deve essere installato mediante l'sdk, deve essere generata la nuova fimage e successivamente trasferita sulla fox.

Creazione file di configurazione e script

Per realizzare una connessione utilizzando ppp sono necessari alcuni files di configurazione.
Creare la directory /etc/ppp:

```
# cd /etc
# mkdir ppp
# cd ppp
```

All'interno della nuova directory creare i seguenti file:

- ppp-start;
- gprs-options;
- gprs-connect;
- gprs-disconnect;
- pap-secrets.

Esempio di contenuto	Descrizione
`/usr/sbin/pppd file /etc/ppp/gprs-options`	*ppp-start* è un file di comando che lancia il demone pppd passandogli come parametro il file /etc/ppp/gprs-options che contiene le opzioni di lavoro di PPP.

`/dev/ttyS2`
`115220`
`defaultroute`
`user "guest"`
 	
`connect "/usr/sbin/chat -v -f /etc/ppp/gprs-connect"`
 	
`disconnect "/usr/sbin/chat -v -f /etc/ppp/gprs-disconnect"`
	In *gprs-options* sono specificati la porta seriale a cui il modem è connesso, la velocità della porta e l'username da usare durante la fase di connessione dati. Inoltre all'interno di questo file sono presenti i comandi chat con cui vengono lanciati i file contenenti i comandi AT per lo start e lo stop della connessione.

`TIMEOUT 60`
`ABORT 'BUSY'`
`ABORT'ERROR'`
`ABORT 'NO CARRIER'`
`'' 'AT'`
 	
`OK`
`AT+CGDCONT=1,"IP","tre.it","0.0.0.0",0,0`
 	
`OK  ATD*99***1#`
`CONNECT ''`
	gprs-connect è utilizzato dal comando chat nella fase iniziale, allo start di ppp.

`ABORT 'BUSY'`
`ABORT'ERROR'`
`ABORT'NO DIALTONE'`
`TIMEOUT30`
`'''+++\c'`
`SAY " + sending break"`
`'NO CARRIER' 'ATH'`
 	
`SAY "\n + dropping data connection"`
 	
`OK'AT+CGATT=0'`
`SAY"\n + disconnecting from GPRS"`
`OK '\c'`
`SAY "\n + disconnected."`
	gprs-connect è utilizzato dal comando chat nella fase finale, allo stop di ppp.

`Guest Guest`	*pap-secrets* contiene username e password per autenticarsi sulla rete. Per gli operatori di telefonia italiani password e username possono essere qualsiasi.

Tabella 7: Files di configurazione per pppd

5

Sviluppo del sistema di trasmissione dati

Nei precedenti capitoli si è discusso degli aspetti teorici e della preparazione degli strumenti necessari per sviluppare un sistema di trasmissione dati che sfrutta la tecnologia HSPA.

In questo capitolo si espone il procedimento svolto per la realizzazione del sistema (collegamenti, configurazioni, etc.) e infine si fornisce un esempio di utilizzo concreto del sistema realizzato, *la trappola fotografica*.

5.1 – Collegamento PC con modulo Wavecom

Il collegamento illustrato in figura 19 permette di testare le notevoli funzionalità del modulo Wavecom. In realtà il collegamento è stato realizzato tra PC e board di sviluppo sierra wireless. Ciò facilità molto il lavoro, che altrimenti si sarebbe arricchito della realizzazione fisica dei connettori dati e di alimentazione, cosa non semplicissima visto le dimensioni ridotte dei componenti. Tale situazione che sembrerebbe essere un punto negativo in ambiente di sviluppo, si rivela fondamentalmente positiva quando, terminata la fase di sviluppo, si devono realizzare soluzioni integrate in cui spesso dimensioni ridotte sono preferite.

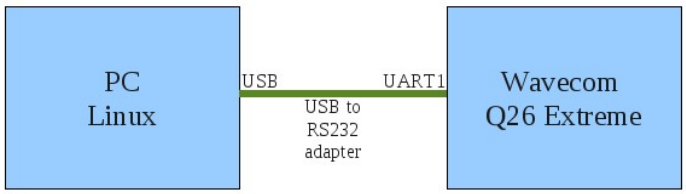

Figura 19: Schema di collegamento tra PC e Q26 extreme

Per collegare il cavo seriale al PC, qualora il PC fosse sprovvisto di tale porta, si deve utilizzare un adattatore da usb a seriale, facilmente reperibile sul mercato.

Per comunicare con il modulo Wavecom sul PC si deve installare un terminale di comunicazione seriale. Un semplice programma dotato di interfaccia grafica per sistemi operativi linux è *moserial terminal*.

Prima di iniziare la comunicazione si devono impostare dei parametri di connessione sul terminale seriale:

- device: /dev/ttyUSB0;
- baud rate: 115200;
- data bits: 8;
- stop bits: 1;
- parity: none.

In particolare la porta device assegnata può essere ottenuta utilizzando, in un terminale, il comando:

```
# dmesg
```

Una volta stabilita la connessione il modulo Wavecom non richiede alcuna stringa di inizializzazione. Tuttavia prima di poter rispondere ai comandi AT, il modulo da in output delle risposte non sollecitate che forniscono delle informazioni sullo stato del modulo.

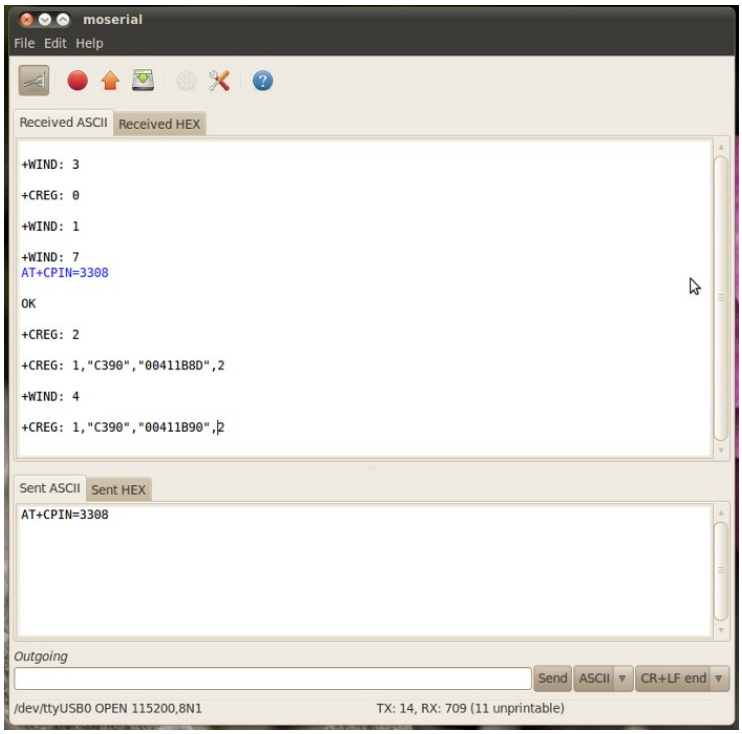

Figura 20: Inizio comunicazione seriale tra PC e modulo Wavecom

In figura 20 è possibile vedere un esempio. In questo specifico caso la scheda sim utilizzata ha la richiesta del pin abilitata, in tal caso si deve usare il comando AT+CPIN=xxxx, ove al posto delle x va inserito il pin.

In tabella 8 si può osservare una spiegazione delle singole risposte inviate dal modulo.

+WIND: 3	Reset della periferica allo scopo di far partire l'applicazione interna (nella sessione di esempio solo Open AT Firmware)
+WIND: 1	Il pin di presenza SIM è stato rilevato su “SIM inserita”
+WIND: 7	Il servizio di rete è disponibile per chiamate di emergenza
+WIND: 4	L'inizializzazione è stata completata

Tabella 8: Risposte non sollecitate visualizzate all'inizio della connessione seriale

Dopo questa fase il modulo Wavecom è pronto per accettare ogni comando AT supportato.
Con la configurazione illustrata in figura 19, utilizzando un PC Windows, è possibile utilizzare i tools di sviluppo Sierra Wireless (menzionati nel capitolo 3). In particolare è di notevole aiuto il programma Espresso. Esso permette di utilizzare le principali funzioni del modulo tramite interfaccia grafica e visualizzare in una finestra i comandi AT necessari per fare l'operazione prescelta.

5.2 – Collegamento PC con FOX Board LX832

Il collegamento tra PC e FOX Board viene realizzato attraverso un cavo ethernet incrociato con connettori RJ45, come in figura 21.

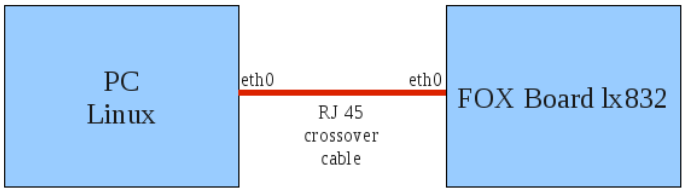

Figura 21: Schema di collegamento tra PC e Fox Board

Affinché i due dispositivi possano comunicare, devono essere impostati degli indirizzi IP della stessa classe.

PC linux	192.168.10.10/24
Fox Board	192.168.10.91/24

Tabella 9: Configurazione interfacce ethernet di PC e fox

Per assegnare gli indirizzi sulle interfacce, digitare dai rispettivi terminali:

```
# ifconfig eth0 indirizzo_ip
```

Per accedere alla fox board si possono utilizzare diversi servizi, il telnet, l'ftp, l'ssh o il server web. Il collegamento come da figura 21 serve per verificare il corretto funzionamento della fox board ed è la configurazione ottimale per effettuare il reflash della board. Per gestire la connessione con la tecnologia HSPA la fox board deve includere pppd. Per inserire pppd sulla fox si deve utilizzare l'sdk, generare la nuova fimage e poi installarla sulla fox board.
Come già spiegato nel paragrafo 1 del capitolo 4 per avviare l'sdk, in un terminale sul PC linux digitare:

```
# cd /home/fox/devboard-R2_01
# make menuconfig
```

Una volta aperto l'sdk, andare in *Application → Networking* e selezionare *Enable point to point protocol (PPP) support*, come mostrato in figura 22.

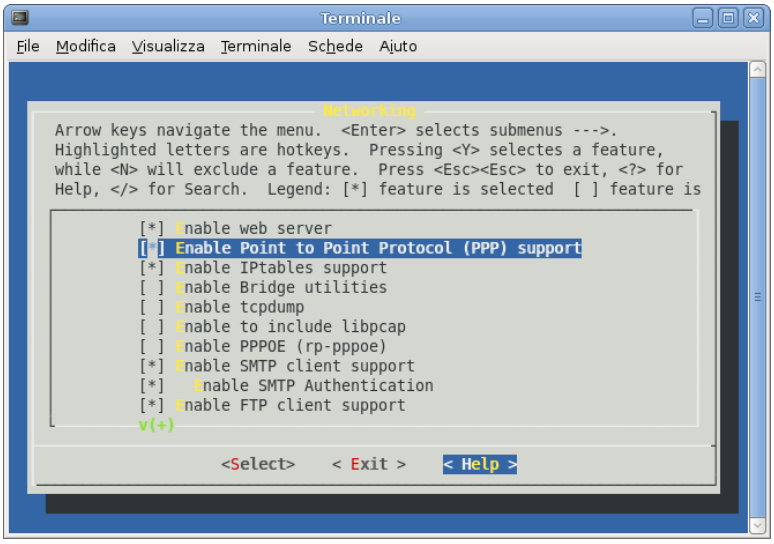

Figura 22: Utilizzo dell'sdk per abilitare il supporto a ppp

Per generare la nuova fimage digitare:

```
# ./configure
# make
```

Dopo alcuni minuti nella stessa directory in cui si è digitato il comando, si trova il file *fimage*.

Scegliendo il reflash tramite ftp, digitare:

```
diego-notebook:/home/diego# ftp 192.168.10.91
Connected to 192.168.10.91.
220 Acmesystems.it - FOXBOARD LX832 release 2.20 (lug 25 2010)
ready.
Name (192.168.0.90:root): root
331 User name okay, need password.
Password: pass
230 User logged in, proceed.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put fimage flash
```

Durante il procedimento vengono mostrate le percentuali di avanzamento, fino a quando la fox board non si riavvia. Arrivati a questo punto la fox board è pronta per essere collegata per il modulo Wavecom e gestire la connessione.

5.3 – Collegamento modulo Wavecom e FOX Board lx832

Il collegamento tra la Fox board e il modulo Wavecom avviene tramite porte seriali. Per il modulo Wavecom si utilizza la UART1 della board di sviluppo, mentre per la Fox si utilizza la porta usb1 utilizzando un adattatore da usb a seriale.

Per controllare il sistema si utilizza un PC con sistema operativo linux (debian lenny), si collegano le porte ethernet del PC e della Fox tramite un cavo ethernet incrociato.

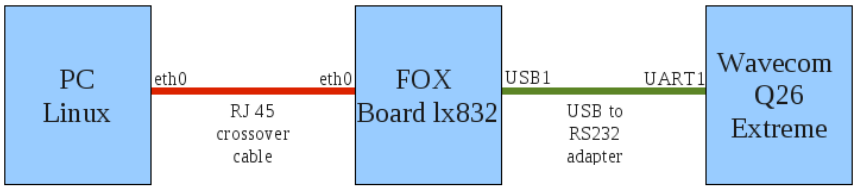

Figura 23: Schema di collegamento tra PC, Fox Board e Q26 extreme

Configurazione ppp

Per permettere alla Fox di connettersi tramite il modulo Wavecom, si devono configurare i file necessari al demone ppp.

Come già riportato nel paragrafo 4.2.2, è necessario creare alcuni files all'interno del percorso */etc/ppp/*.

ppp-start

```
/usr/sbin/pppd file /etc/ppp/hspa-options
```

hspa-option

```
/dev/ttyUSB0
115200
defaultroute
user "diego"
connect "/usr/sbin/chat -v -f /etc/ppp/hspa-connect"
disconnect "/usr/sbin/chat -v -f /etc/ppp/hspa-disconnect"
```

hspa-connect

```
'TIMEOUT'       '60'
'ABORT'         'BUSY'
'ABORT'         'ERROR'
'ABORT'         'NO CARRIER'
''              'AT'
'OK'            'AT+CGDCONT=1,"IP","tre.it","0.0.0.0",0,0'
'OK'            'AT+WWSM=1'
'OK'            'AT+CSQ'
'OK'            'AT+COPS?'
'OK'            'AT+WGPRS=9,2'
'OK'            'ATD*99***1#'
'CONNECT'       ''
```

hspa-disconnect

```
ABORT           'BUSY'
ABORT           'ERROR'
ABORT           'NO DIALTONE'
TIMEOUT         30
''              '+++\c'
SAY             " + sending break"
'NO CARRIER'      'ATH'
SAY             "\n + dropping data connection"
OK              'AT+CGATT=0'
SAY             "\n + disconnecting from HSPA"
OK              '\c'
SAY             "\n + disconnected."
```

ppp-stop

```
/usr/sbin/chat -v -f /etc/ppp/hspa-disconnect
```

pap-secrets

```
diego       diego
```

Tabella 10: Files di configurazione per demone ppp su fox board

Inoltre per abilitare il routing HSPA è necessario disabilitare il gateway di default utilizzando il comando:

```
# route del default gw 192.168.10.1 eth0
```

Per memorizzare questa impostazione bisogna editare il file */etc/conf.d/net.eth0* e cancellare o commentare la riga che riguarda il gateway.

Un ulteriore passo da fare prima di passare alla connessione è l'impostazione dei DNS. I corretti DNS, che variano in base all'operatore telefonico, devo essere inseriti nel file */etc/resolv.conf.*

```
nameserver 62.13.171.1
nameserver 62.13.171.2
```

Tabella 11: File resolv.conf con i dns della Tre

Connessione

Per controllare la connessione si rendono necessarie due finestre terminali sul PC, ognuna collegata tramite telnet (o ssh) alla Fox board.

Il terminale 1 serve per far dare i comandi di start e stop alla connessione. Il terminale 2 serve per monitore lo stato della connessione.

Sul terminale 2 digitare il comando:

```
 # tail –f /var/log/messages
```

Sul terminale 1 digitare il comando:

```
# . /etc/ppp/ppp-start
```

Sul terminale 2 si osserva la fase di start della connessione:

```
Jan  1 00:05:46 axis-00408c161024 pppd[127]: pppd 2.4.2b3 started by
root, uid 0

... [cut]

Jan  1 00:05:48 axis-00408c161024 chat[129]: send (ATD*99***1#^M)
Jan  1 00:05:49 axis-00408c161024 chat[129]: expect (CONNECT)
Jan  1 00:05:49 axis-00408c161024 chat[129]: ^M
Jan  1 00:05:49 axis-00408c161024 chat[129]: ATD*99***1#^M^M
Jan  1 00:05:49 axis-00408c161024 chat[129]: CONNECT

Jan  1 00:05:49 axis-00408c161024 pppd[127]: Serial connection
established.
Jan  1 00:05:49 axis-00408c161024 pppd[127]: Using interface ppp0
Jan  1 00:05:49 axis-00408c161024 pppd[127]: Connect: ppp0 <-->
/dev/ttyUSB0
Jan  1 00:05:55 axis-00408c161024 pppd[127]: Remote message:
Welcome!
Jan  1 00:05:55 axis-00408c161024 pppd[127]: PAP authentication
succeeded
Jan  1 00:05:56 axis-00408c161024 pppd[127]: local  IP address
1.103.130.83
Jan  1 00:05:56 axis-00408c161024 pppd[127]: remote IP address
192.168.111.111
```

Tabella 12: Risultati terminale 2 dopo il ppp-start nel terminale 1

Per verificare l'attivazione del protocollo ppp, si può utilizzare il comando "ifconfig" nel terminale 01.

```
[root@axis-00408c161024 /root]104# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:40:8C:16:10:24
inet addr:192.168.0.90  Bcast:192.168.0.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:912 errors:0 dropped:0 overruns:0 frame:0
TX packets:694 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:60123 (58.7 KiB)  TX bytes:58901 (57.5 KiB)
Interrupt:17 DMA chan:1

lo        Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
UP LOOPBACK RUNNING  MTU:16436  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ppp0      Link encap:Point-Point Protocol
inet addr:1.103.130.83  P-t-P:192.168.111.111  Mask:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
RX packets:4 errors:0 dropped:0 overruns:0 frame:0
TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:3
RX bytes:40 (40.0 B)  TX bytes:58 (58.0 B)
```

Tabella 13: Ifconfig nel terminale 1

Per accertarsi che la connessione funzioni e i dns siano correttamenet configurati si testa la connessione facendo un ping.

```
[root@axis-00408c161024 /etc]138# ping www.diegotuzi.netsons.org
www.diegotuzi.netsons.org is alive!
[root@axis-00408c161024 /etc]138# ping www.google.it
www.l.google.com is alive
```

Tabella 14: Ping per controllare la connettività e la risoluzione degli host

Disconnessione

Per disconnettersi è necessario richiamare lo script di stop digitando nel terminale 1:

```
# cd /etc/ppp/
# ./ppp-stop
```

Nel terminale 2 si osserva l'interruzione della connessione:

```
...
Jan  1 00:31:43 axis-00408c161024 pppd[127]: Terminating on signal
15.
Jan  1 00:31:43 axis-00408c161024 pppd[127]: Connection terminated.
Jan  1 00:31:43 axis-00408c161024 pppd[127]: Connect time 25.9
minutes.
Jan  1 00:31:43 axis-00408c161024 pppd[127]: Sent 58 bytes, received
40 bytes.
```

A questo punto il sistema di elaborazione fox board può controllare la connessione utilizzando gli script di start e stop.

5.4 – Trappola fotografica

Realizzato il sistema con connessione HSPA, questo paragrafo si propone di presentare un esempio di applicazione a cui il sistema può dedicarsi. L'esempio riportato è denominato *trappola fotografica*.

Il principio di funzionamento della trappola fotografica, consiste nel catturare l'immagine mediante una fotocamera, nel momento in cui si attraversa un varco su cui sono collocate delle fotocellule. L'immagine viene acquisita dal sistema di elaborazione che, attraverso il modulo che fornisce la connessione, carica la foto in un sito web.

I collegamenti da effettuare per testare questa applicazione sono riportati in figura 24.

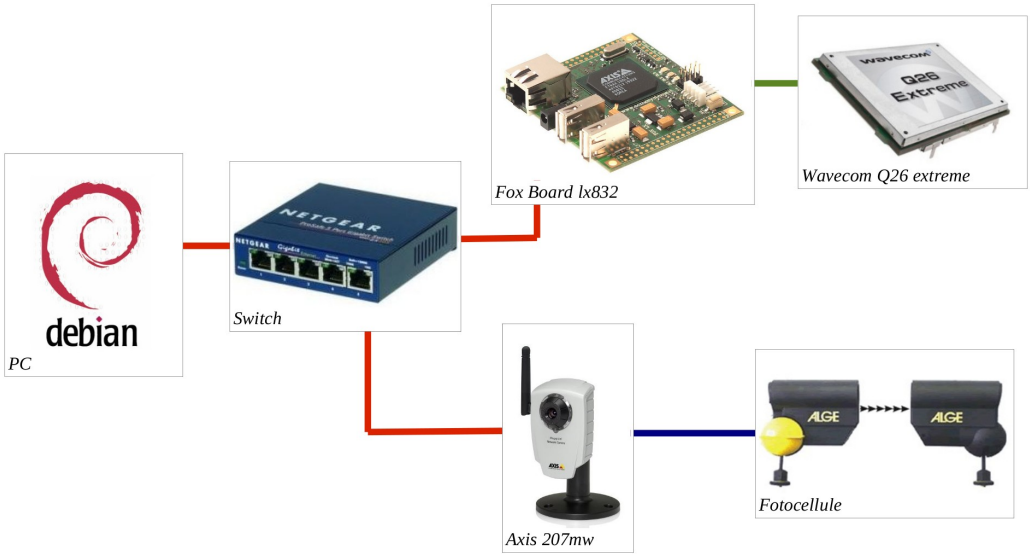

Figura 24: Schema collegamento per trappola fotografica

5.4.1 – Configurazioni necessarie

Le interfacce ethernet dei rispettivi componenti utilizzano gli indirizzi IP riportati in tabella 15.

PC	192.168.10.10
Fox Board	192.168.10.91
Axis 207 mw	192.168.10.253

Tabella 15: IP dei componenti della trappola fotografica

La realizzazione della trappola fotografica implica la modifica del file di configurazione ppp-start. Inoltre si utilizzano il programma *web_axis_server* e gli script *ip-up*, *ip-down* e *firewall* realizzati dall'*Ing. Stefano Marchesani*.

Modifica del file di configurazione ppp-start

ppp-start

`insmod /lib/modules/2.6.26/kernel/net/ipv4/netfilter/nf_conntrack_ipv4.ko`
`insmod /lib/modules/2.6.26/kernel/net/ipv4/netfilter/nf_nat.ko`
`insmod /lib/modules/2.6.26/kernel/net/ipv4/netfilter/iptable_nat.ko`
`/etc/init.d/httpd stop`
`/usr/sbin/pppd file /etc/ppp/hspa-options`

Tabella 16: ppp-start per la trappola fotografica

Il nuovo file ppp-start è riportato in tabella 16. Le prime tre righe consentono il caricamento dei moduli del kernel necessari allo script *firewall*.

Tali moduli devono essere presenti sulla immagine di sistema della fox board, per inserirli si deve utilizzare l'sdk. Inoltre deve anche essere presente il supporto al comando iptables.

La riga */etc/init.d/httpd stop* ferma il server web della foxboard.

Il programma *"web_server_axis"*

Il programma *web_axis_server*, una volta lanciato si pone in ascolto sulla porta passatagli come parametro. La fotocamera considera il passaggio attraverso le fotocellule come un evento allarme, ed è impostata per mandare l'immagine scattata sulla porta 8080 dell'indirizzo ip della fox board utilizzando il protocollo HTTP.

La fox board che ha in esecuzione il programma web_axis_server raccoglie l'immagine, la salva nella cartella */mnt/flash/root* con il nome *image.jpg* e, se presente una connessione, invia l'immagine sul sito *http://www.foxwsn.altervista.org/gransasso/*.

Web_server_axis deve essere collocato nella directory */mnt/flash/root.* Il programma viene lanciato automaticamente dallo script ip-up, per lanciarlo manualmente è necessario digitare:

```
# ./web_server_axis 8080
```

Gli script *"ip-up", "ip-down"* e *"firewall"*

Lo script ip-up è avviato automaticamente da pppd all'inizio della connessione. Lo script ip-up avvia lo script firewall e avvia il programma web_axis_server sulla porta 8080. Lo script ip-up va collocato nella directory */etc/ppp/*.

Lo script firewall, ha il compito di inoltrare tutte le richieste provenienti dalla rete verso la porta 80, all'indirizzo ip della axis 207mw. Tale operazione ha lo scopo di rendere accessibile il server web della fotocamera anche da remoto attraverso internet, consentendo di modificare tutti i parametri della fotocamera, come ad esempio la dimensione dell'immagine. Lo script firewall va collocato nella directory */etc/init.d/*.

Lo script ip-down, viene chiamato automaticamente da pppd al termine della connessione ed ha il compito di fermare lo script firewall. Lo script ip-down va collocato nella directory */etc/ppp/*.

5.4.2 – Avvio Trappola Fotografica

Per avviare la trappola fotografica collegarsi alla foxboard e digitare:

```
# telnet 192.168.10.91
Trying 192.168.10.91...
Connected to 192.168.10.91.
Escape character is '^]'.
axis-00408c161024 login: root
Password: pass
[root@axis-00408c161024 /etc/ppp]384# ./ppp-start
 * Stopping web server...
[ ok ]
[root@axis-00408c161024 /etc/ppp]384#
```

Tabella 17: Avvio della trappola fotografica

Una volta avviato il file ppp-start, oltre ad essere avviata la connessione ppp, vengono avviati il programma web_server_axis e gli script.

```
[root@axis-00408c161024 /etc/ppp]384# ps
  PID USER       VSZ STAT COMMAND
    1 root      1096 S    init
    2 root         0 SW<  [kthreadd]
    3 root         0 SW<  [ksoftirqd/0]
    4 root         0 SW<  [events/0]
    5 root         0 SW<  [khelper]
   31 root         0 SW<  [kblockd/0]
   41 root         0 SW<  [khubd]
   58 root         0 SW   [pdflush]
   59 root         0 SW   [pdflush]
   60 root         0 SW<  [kswapd0]
   61 root         0 SW<  [aio/0]
   62 root         0 SW<  [nfsiod]
  105 root         0 SW<  [mtdblockd]
  149 root         0 SW<  [rpciod/0]
  159 root         0 SWN  [jffs2_gcd_mtd3]
  168 root      1152 S <  /sbin/udevd --daemon
  306 root      2296 S    /bin/sh
  310 root      1096 S    /sbin/respawnd
  313 root      1272 S    /usr/sbin/syslogd -m 0 -o 40000
  317 root      1104 S    /usr/sbin/klogd -x
  359 root      1104 S    /bin/utelnetd -d
  364 root      1328 S    /bin/vftpd -r
  376 root      1440 S    /usr/sbin/dropbear
  384 root      2312 S    -sh
  389 root      2312 S    -sh
  394 root      2304 S    tail -f /var/log/messages
  422 root      1928 S    /usr/sbin/pppd file /etc/ppp/gprs-options
  435 root      1088 S    /mnt/flash/root/web_axis_server 8080
  438 root      2336 R    ps
```

Tabella 18: Visualizzazione dei processi attivi con il comando ps

In tabella 18 sono riportati i processi attivi sulla fox board. Si può verificare che sia pppd che web_axis_server sono in esecuzione.

In tabella 19 è riportato l'avanzamento dello script di connessione fino a che la connessione non viene stabilita.

```
# tail -f /var/log/messages
Jan  1 00:03:17 axis-00408c161024 pppd[422]: pppd 2.4.2b3 started by
root, uid 0
Jan  1 00:03:18 axis-00408c161024 chat[424]: timeout set to 60
seconds
Jan  1 00:03:18 axis-00408c161024 chat[424]: abort on (BUSY)
Jan  1 00:03:18 axis-00408c161024 chat[424]: abort on (ERROR)
Jan  1 00:03:18 axis-00408c161024 chat[424]: abort on (NO CARRIER)
Jan  1 00:03:19 axis-00408c161024 chat[424]: send (AT^M)
Jan  1 00:03:19 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:19 axis-00408c161024 chat[424]: AT^M^M
Jan  1 00:03:19 axis-00408c161024 chat[424]: OK
Jan  1 00:03:19 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:19 axis-00408c161024 chat[424]: send
(AT+CGDCONT=1,"IP","tre.it","0.0.0.0",0,0^M)
Jan  1 00:03:20 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:20 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:20 axis-00408c161024 chat[424]:
AT+CGDCONT=1,"IP","tre.it","0.0.0.0",0,0^M^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: OK
Jan  1 00:03:20 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:20 axis-00408c161024 chat[424]: send (AT+WWSM=1^M)
Jan  1 00:03:20 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:20 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: AT+WWSM=1^M^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: OK
Jan  1 00:03:20 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:20 axis-00408c161024 chat[424]: send (AT+ICF?^M)
Jan  1 00:03:20 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:20 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: AT+ICF?^M^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: +ICF: 3,4^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:20 axis-00408c161024 chat[424]: OK
Jan  1 00:03:20 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:20 axis-00408c161024 chat[424]: send (AT+CSQ^M)
Jan  1 00:03:21 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:21 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: AT+CSQ^M^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: +CSQ: 13,0^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: OK
Jan  1 00:03:21 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:21 axis-00408c161024 chat[424]: send (AT+WGPRS=9,2^M)
Jan  1 00:03:21 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:21 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: AT+WGPRS=9,2^M^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: +WGPRSIND: 3^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: OK
Jan  1 00:03:21 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:21 axis-00408c161024 chat[424]: send (AT+COPS?^M)
Jan  1 00:03:21 axis-00408c161024 chat[424]: expect (OK)
Jan  1 00:03:21 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: AT+COPS?^M^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: +COPS: 0,2,22299,2^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:21 axis-00408c161024 chat[424]: OK
Jan  1 00:03:21 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:21 axis-00408c161024 chat[424]: send (ATD*99***1#^M)
Jan  1 00:03:22 axis-00408c161024 chat[424]: expect (CONNECT)
Jan  1 00:03:22 axis-00408c161024 chat[424]: ^M
Jan  1 00:03:22 axis-00408c161024 chat[424]: ATD*99***1#^M^M
Jan  1 00:03:22 axis-00408c161024 chat[424]: CONNECT
```

```
Jan  1 00:03:22 axis-00408c161024 chat[424]:  -- got it
Jan  1 00:03:22 axis-00408c161024 chat[424]: send (^M)
Jan  1 00:03:22 axis-00408c161024 pppd[422]: Serial connection
established.
Jan  1 00:03:22 axis-00408c161024 pppd[422]: Using interface ppp0
Jan  1 00:03:22 axis-00408c161024 pppd[422]: Connect: ppp0 <-->
/dev/ttyUSB0
Jan  1 00:03:29 axis-00408c161024 pppd[422]: Remote message:
Welcome!
Jan  1 00:03:29 axis-00408c161024 pppd[422]: PAP authentication
succeeded
Jan  1 00:03:29 axis-00408c161024 pppd[422]: local  IP address
1.99.20.233
Jan  1 00:03:29 axis-00408c161024 pppd[422]: remote IP address
192.168.111.111
```

Tabella 19: Verifica connessione ppp

Verificata la connessione e l'esecuzione di web_axis_server, si può azionare la trappola fotografica. Passando attraverso la fotocellula la fotocamera scatta la foto, la fox board la memorizza e, tramite la connessione con tecnologia HSPA, la invia al sito internet predefinito.

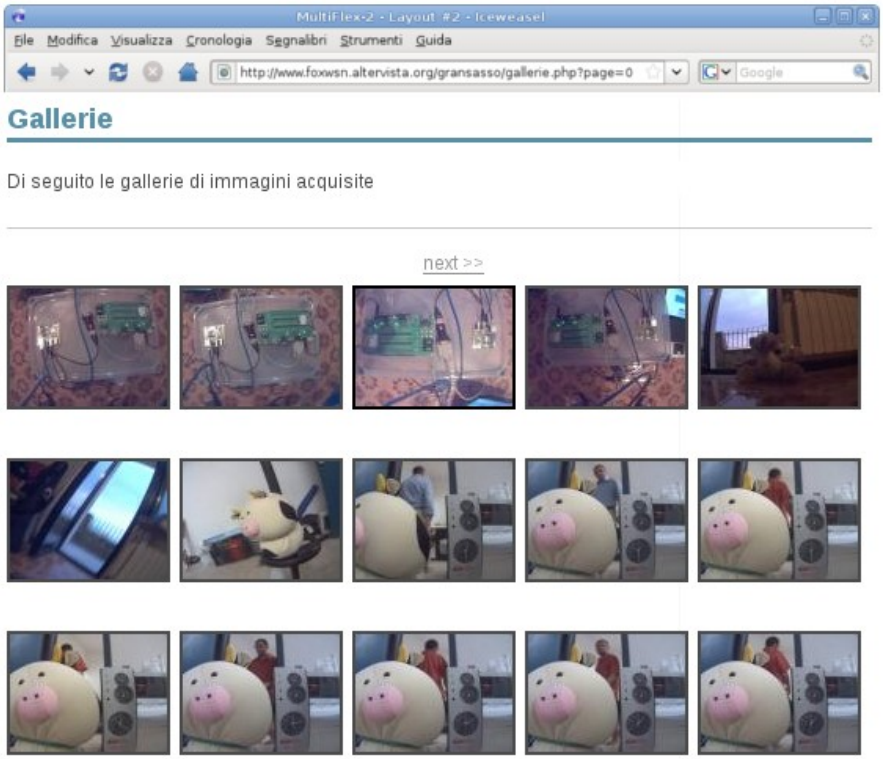

Figura 25: Sequenza di immagini acquisite e inviate sul sito

Per terminare la connessione eseguire lo script di disconnessione.

```
# ./ppp-stop
```

6

Conclusioni e possibili sviluppi futuri

L'obiettivo di questo lavoro di tesi è stato quello di studiare e sviluppare un sistema di trasmissione dati che sia in grado di utilizzare le tecnologie HSPA. L'aspetto più interessante del sistema realizzato è la sua generalità di utilizzo. In questo documento è stato trattato un solo esempio di applicazione, ma ce ne sono molte altre che possono essere sviluppate su tale sistema. Praticamente questo sistema potrebbe essere utilizzato per rendere mobile una qualsiasi applicazione.

Un altro aspetto importante è la facilità con cui è possibile sviluppare tali sistemi. Facilità intesa nel senso che si possono realizzare applicazioni di notevole valore anche con hardware particolarmente economici e senza strumentazioni rilevanti.

Attraverso lo sviluppo di tali sistemi, si evince l'importanza di avere competenze nel campo dei sistemi operativi linux e competenze nei linguaggi di programmazione. La maggior parte degli hardware utilizzati hanno tutti la stessa struttura di base, ovvero un sistema di elaborazione commisurato al carico di lavoro, su cui gira un sistema dedicato allo specifico hardware, spesso basato su kernel linux.

Per quanto riguarda i possibili sviluppi futuri, si potrebbe fare molto per il piccolo sistema sviluppato in questo documento. Una operazione interessante potrebbe essere quella di rendere il sistema totalmente integrato.

La configurazione sviluppata ha un consistente spreco di risorse. Il modulo Wavecom è una wireless CPU e, come già visto nel capitolo 3, significa che è un modem e un sistema di elaborazione. Un possibile miglioramento sarebbe quello di realizzare applicazioni direttamente sul modulo Wavecom, sfruttando la sua CPU. Quindi un primo passo possibile sarebbe quello di eliminare la Fox board.

In realtà anche la videocamera ha un proprio sistema di elaborazione. Un ulteriore sviluppo potrebbe riguardare l'utilizzo del solo sensore ottico controllato dalla stessa CPU che controlla il modulo della connessione. Quindi il passo successivo potrebbe essere quello di eliminare la fotocamera e utilizzare il solo sensore ottico.

Ulteriore passo potrebbe essere quello di integrare la fotocellula nel sistema.

Facendo quanto detto sopra, si arriverebbe al punto di creare un unico hardware con uno sfruttamento delle risorse disponibili sensibilmente maggiore al caso precedente e porterebbe ad avere minori dimensioni e minori costi. Ovviamente tale cambiamenti, che apportano notevoli benefici, vanno a togliere un po' di facilità al processo di sviluppo.

Le possibilità di sviluppo sono davvero molte e continueranno a crescere, grazie al continuo progresso di sviluppo tecnologico, che fornisce hardware più potenti a prezzi sempre minori, e alla crescente importanza della mobilità.

Bibliografia

- Bohagen, Binningsbo, “HSPA and LTE–Future-proof Mobile Broadband Solutions”, Telektronikk, gennaio 2010
- D'Antonio, Gianola, Romano, “Evoluzioni delle reti mobili verso la larga banda”, Notiziario tecnico telecom italia, luglio 2007
- Dahlman, Parkvall, Sköld, Beming, 3g Evolution: Hspa and Lte for mobile broadband,Oxford, Academic Press, 2008
- Franceschini , “La copertura dei servizi HSDPA e le ulteriori possibilità di sviluppo delle connessioni mobili per Telecom Italia”, Telecom Italia , marzo 2009
- Manco, “L'evoluzione delle reti e dei servizi di TLC”, Ordine degli ingegneri di napoli, ottobre 2006
- Piccardi, “GaPiL Guida alla Programmazione in Linux ”, 2008
- Sierra Wireless, “AT Commands Interface Guide for Firmware 7.43”
- Vasseur, “L'invasione degli Smart Object”, L'espresso, febbraio 2010
- Zaniboni, “UMTS”, Wireless Future, giugno 2006

- Acme Systems, “FOX Board documentation”, http://foxlx.acmesystems.it/?id=14
- Axis, “Axis developer wiki”, http://developer.axis.com/wiki/doku.php
- Open skill, “ppp e pppd – Introduzione al protocollo ppp e utilizzo del server pppd su Linux”, http://openskill.info/topic.php?ID=127
- Samba.org, “ppp e pppd”, http://ppp.samba.org/ppp/pppd.html
- Wikipedia.it
- 3GPP, Release 99, 5, 6, 7 e 8, http://www.3gpp.org/.

Ringraziamenti

Desidero innanzitutto ringraziare con grande affetto i miei genitori, Antonino e Filomena. I miei genitori sono per me persone speciali, sono persone che ammiro e da cui prendo esempio tutti i giorni. Un ringraziamento ai miei cari Claudio, Luisa, Elio, Sandra e un grande bacio ai miei nipoti Giulia, Beatrice e Domenico che riescono a rendermi felice anche nei momenti più difficili. Ringrazio tutti i miei amici e in particolare la mia "amica" Francesca.

Desidero ringraziare il Professor Fabio Graziosi, e l'Ing. Andrea Colarieti che mi hanno assistito durante tutto il lavoro di tesi. Un ringraziamento particolare all'Ing. Stefano Marchesani per la sua disponibilità in momenti critici. Inoltre, ringrazio sentitamente l'Università degli Studi dell'Aquila e il centro di eccellenza DEWS per avermi permesso di utilizzare il materiale hardware necessario allo sviluppo della tesi.

Infine voglio ringraziare tutti coloro che mi vogliono bene e che mi fanno sentire la loro vicinanza.