



UNIVERSITA DEGLI STUDI DI CATANIA
FACOLTA DI SCIENZE MATEMATICHE, FISICHE E NATURALI
CORSO DI LAUREA IN INFORMATICA

Graziano Larocca

**STUDIO E REALIZZAZIONE DI UN
PROTOCOLLO DI COMPRESSIONE DATI
PER RETI DI SENSORI SISMICI**

Tesi di Laurea

Relatore:
Chiar.mo Prof. Ing. Salvatore Riccobene

Correlatore:
Dott. Ing. Danilo Reitano

Anno Accademico 2005 – 2006

INDICE

PREFAZIONE.....	1
1. La RSP digitale BB gestita da INGV CT.....	10
1.1 La trasmissione VSAT.....	10
1.2 Trasmissione VSAT: Networking.....	14
1.3 La trasmissione radio UHF.....	15
1.4 Descrizione della Rete Sismica Permanente.....	17
2. Digitalizzazione.....	24
2.1 Caratteristiche principali.....	28
2.2 Filtraggio e Sincronizzazione.....	28
3. L’algoritmo di compressione utilizzato.....	31
4. Calcolo della banda utilizzata (Satellitare e radio)	33
5. Il server di acquisizione.....	37
5.1 Acquisizione e storage dei dati: il <i>NaqsServer</i> ...	38
5.2 Accesso ai dati: Il <i>DataServer</i>	42
5.3 Controllo e gestione del sistema: <i>Playback Utilities</i>	42
5.4 Monitoraggio della Rete: Il <i>NaqsView</i> e il <i>Waveform Viewer</i>	43
5.5 Controllo dei processi server: <i>NmxWatchDog</i>	46
6. Protocollo per l’accesso ai dati di tipo Nanometrics®	47

6.1 Descrizione del formato dei messaggi del protocollo di sottoscrizione.....	49
6.2 Descrizione del formato dei messaggi del protocollo di accesso ai dati.....	51
7. Il Formato dati NMX.....	52
7.1 Inbound Packets e “bundles” per packet.....	53
7.2 Descrizione dei pacchetti del tipo inbound.....	53
7.3 Inbound packets: Pacchetti compressi di dati...	56
7.4 Inbound packets: Bundles di dati.....	58
7.5 Inbound packets: Transparent serial packet.....	60
8. Il tremore vulcanico.....	62
8.1 Analisi del tremore nell'eruzione dell'Etna del 2001.....	62
8.2 Calcolo del tremore in funzione della distanza..	64
8.3 Applicazione del calcolo del tremore nella Rete Sismica Permanente Odierna.....	65
9. Studio ed implementazione di un nuovo algoritmo di compressione.....	69
9.1 Distribuzioni Gaussiane – RMS	69
9.2 Deviazione standard nella situazione di quiete	73
9.3 Teoria di Huffman e First Difference.....	75
9.4 Pseudocodifica algoritmo predittivo proposto	76
9.5 Ottimizzazione del data bundle Nanometrics® ..	79

9.6 Calcolo della differenza media tra campioni	86
10. Confronto tra i protocolli di trasmissione:	
statistiche di efficienza.....	89
10.1 Test in laboratorio su consumo di banda.....	89
10.2 Statistiche di efficienza per la trasmissione radio UHF.....	89
10.3 Statistiche di efficienza per la trasmissione radio Satellitare	114
Appendice 1 . Utilizzo del protocollo Nanometrics®:	
Implementazione di un Client per lo streaming dei dati meteo.....	123
Bibliografia.....	132

Prefazione.

Nella seguente tesi si descrive l'attività svolta presso L'Istituto Nazionale di Geofisica e Vulcanologia (INGV) Sezione di Catania, Ente presso il quale sono anche dipendente, nel periodo che va da Settembre 2005 a Luglio 2006. Scopo della tesi è stato realizzare un protocollo che permettesse una ottimizzazione relativamente alla compressione dei dati trasmessi dalle stazioni sismiche digitali, facenti parte della Rete Sismica Permanente gestita dall'INGV di Catania.

La mansione da me svolta all'interno dell'Istituto è quella di gestire le stazioni sismiche a tecnologia digitale prodotte dall'azienda Nanometrics[®], sita nei pressi di Ottawa (Canada), e coordinare il gruppo di tecnici che vi operano.

L'efficienza, l'ottimo rapporto segnale/rumore e la robustezza di tali stazioni hanno permesso il loro utilizzo in sostituzione delle stazioni analogiche già presenti all'interno della Rete Sismica Permanente (RSP). L'utilizzo di link satellitari e radio operanti nella banda UHF hanno introdotto alcune incognite proprio per la limitatezza della banda utilizzata.

Dopo numerosi calcoli sulla banda impiegata, per le varie stazioni installate, ci si è accorti della sua variabilità, che risultava essere funzione della posizione delle stazioni stesse all'interno dell'areale etneo. Il problema rilevante deriva dal fatto che queste tipologie di stazioni non sono mai state utilizzate in prossimità di vulcani attivi, soprattutto nelle fasi eruttive. Alcune fenomenologie caratteristiche del vulcano Etna in fase eruttiva possono provocare un grande

innalzamento di alcuni parametri del segnale sismico acquisito, come ad esempio la sua ampiezza calcolata in valore RMS, determinando un conseguente incremento del consumo di banda.

In seguito a questa analisi si è voluto proporre uno studio del rapporto ampiezza RMS / consumo di banda, del protocollo di trasmissione e dell'algoritmo di compressione utilizzati dalla Nanometrics® .

Da ciò ne è conseguita una ottimizzazione di tale algoritmo di compressione utilizzando la Teoria di codifica di Huffman e ne è scaturita una stima del consumo di banda prevedibile in caso di eruzione del vulcano Etna.

Nella fase di studio del protocollo di trasmissione Nanometrics® ho inoltre realizzato un software per l'acquisizione dei dati di stazioni meteo, di ultima generazione, installate di recente nella RSP.

I capitoli presenti nella tesi sono:

Capitolo 1

Si descrive la RSP digitale Broad Band (BB) gestita dall' INGV, evidenziando le varie tipologie di trasmissioni e networking basate sulla tecnologia VSAT e link radio UHF.

Il termine VSAT, acronimo di Very Small Aperture Terminal, indica il riferimento alla comunicazione satellitare ed il fatto che queste reti sono costituite da piccole stazioni (VSAT per l'appunto) con un'antenna di diametro poco meno di 2,4m (da qui la definizione di *small aperture*). Le reti satellitari con il loro schema a stella,

utilizzando una stazione centrale (hub) più grande, permettono il trasferimento di dati tra stazioni molto distanti tra loro.

Viene quindi esaminato il protocollo utilizzato nella RSP, il **TDMA**; questi alloca tutta la banda disponibile ad ogni portante ciclicamente per un tempo limitato (*time slot*).

Le **onde radio** o **radio-onde** sono le radiazioni elettromagnetiche di frequenza compresa tra zero e 1 GHz, ovvero lunghezza d'onda da 30 cm in su. La quantità di informazione che può essere trasportata da un segnale radio è proporzionale alla sua frequenza. Le onde radio vengono suddivise in molteplici **bande**, come la UHF (Ultra High Frequency). Le frequenze utilizzate nella Rete Sismica Permanente sono incluse in un range compreso tra 435 e 446 MHz.

Una coppia radio UHF sulla stessa frequenza crea un link di dati che è condiviso dalle due radio con tecnica TDMA.

Capitolo 2

Viene descritta la prima fase dell'acquisizione di un segnale sismico, cioè quella della digitalizzazione del segnale analogico creato da un sensore.

Il movimento della terra viene in una prima fase trasformato in un segnale elettrico. Lo strumento che effettua questa operazione è il trasduttore o sensore. Il sensore utilizzato nelle stazioni sismiche è definito sismometro.

Il compito di trasformare un segnale analogico in uno digitale viene svolto da un *convertitore analogico – digitale* o più semplicemente da un *digitalizzatore*.

Nella conversione analogico-digitale il segnale analogico, che assume infiniti valori fra due limiti prefissati, viene convertito in un segnale digitale caratterizzato da un numero finito di valori. Il segnale analogico pertanto subisce un'operazione di *quantizzazione*, che consiste nel sostituire una grandezza analogica, che può variare con continuità, con una grandezza quantizzata che può assumere solo una serie discreta di valori.

Si affrontano poi i discorsi relativi al filtraggio di disturbi esterni e alle possibili ottimizzazioni del processo di digitalizzazione.

Capitolo 3

Viene esaminato il tipo di compressione dati utilizzato in fase di digitalizzazione nelle strumentazioni dell'azienda Nanometrics®.

Esistono due forme di compressione dati: *Lossless* e *Lossy*.

La compressione dati utilizzata nel nostro caso è di tipo *Lossless*. I dati devono mantenere la loro integrità, ovvero il segnale di origine può essere ricostruito fedelmente. Tuttavia ci sono dei limiti. Infatti il *Compression ratio* non è molto significativo, variando in genere in un range di $2:1 - 8:1$.

Capitolo 4

Viene descritto il metodo per calcolare quanta banda impiega una trasmissione dati e quanta bisogna assegnarne in previsione del tipo di dato sismico acquisito.

I link satellitari e radio costituiscono le connessioni tra le stazioni remote ed il centro di acquisizione. Conoscere qual è la capienza dei

link ed il traffico da gestire è fondamentale nella progettazione della Rete. Inoltre la banda utilizzata non è costante, poiché dipende dall'algoritmo di compressione utilizzato, la cui efficienza deriva dal tipo di segnale acquisito.

Capitolo 5

Vengono descritti i software installati nel server centrale di acquisizione dati. Ogni stazione sismica della rete permanente trasmette direttamente o indirettamente (utilizzando dei centri VSAT), al centro di acquisizione *CUAD* (Centro Unificato Acquisizione Dati) che si trova a Catania. I pacchetti trasmessi dalle stazioni remote vengono ricevuti dal **server di acquisizione** che gestisce tutti i dati della rete.

Il **NaqsServer** è il sistema di acquisizione dati progettato per *ricevere, processare e immagazzinare* dati seriali, dati sismici e gli *State of Health* (SOH). Il sistema NaqsServer è suddiviso in moduli che svolgono diverse funzioni.

Il **DataServer** è un programma di utility che permette l'accesso in locale e in remoto ai dati sismici seriali e agli SOH Nanometrics[®], attraverso l'utilizzo del protocollo TCP/IP.

Le **Playback utilities** costituiscono un insieme di file eseguibili per analizzare i dati una volta archiviati in hard disk.

Il **NaqwView** fornisce un'interfaccia intuitiva per vedere, in tempo reale, gli SOH estraendoli dal NaqsServer, e per spedire comandi di calibrazione e di centramento della massa dei sensori sismici.

Il **Waveform View** fornisce una semplice ed intuitiva interfaccia per vedere in tempo reale i segnali sismici estratti dal NaqsServer.

Capitolo 6

Si descrive come gestire i messaggi di una rete Nanometrics, analizzando il protocollo ed il formato dei dati per poter creare programmi client che ricevono ed interpretano i dati on line. Ogni programma, implementato per la comunicazione con il server di acquisizione, deve seguire le regole stabilite da un protocollo di *sottoscrizione* definito dall'azienda.

Capitolo 7

I dati ricevuti vengono pacchettizzati in formato proprietario **NMX** e quindi inglobati in pacchetti UDP per essere trasmessi con una certa priorità.

In questo capitolo si descrivono in modo dettagliato alcuni dei più importanti formati di pacchetti: i pacchetti di dati seriali e di dati sismici compressi.

Il formato richiede un'accurata sincronizzazione per permettere l'etichettatura del dato per la priorità di spedizione.

Il tempo alla stazione viene fornito da un ricevitore GPS munito di antenna. Il ricevitore è integrato all'interno dei tranceiver.

Ogni pacchetto è costituito da "bundle". Un **bundle** è una collezione di dati indipendente dalle altre. Le differenze maggiori tra un tipo e l'altro risiedono tutte nei byte riservati (*Packet specific*) degli header dei pacchetti.

Capitolo 8

Viene descritta in modo generico, l'eruzione dell'Etna del 2001 focalizzando tutta l'attenzione sulle variazioni del tremore vulcanico registrate. Alcuni ricercatori dell'INGV di Catania, hanno analizzato ampiezza e frequenza dei segnali sismici, riscontrando cambiamenti considerevoli nel tremore vulcanico.

Inoltre viene presa in considerazione l'eruzione relativa all'anno 2004, durante la quale, un gruppo di ricercatori dell'Istituto di Vulcanologia di Catania ha analizzato i dati del tremore vulcanico di otto stazioni sismiche digitali, installate sulla sommità del vulcano Etna. Lo scopo dell'analisi è stato quello di creare un modello matematico per localizzare la sorgente del tremore vulcanico. Il modello risulta fondamentale per poter affrontare un discorso sulla previsione di ampiezza del tremore e consumo di banda in eventuali future eruzioni dell'Etna.

Capitolo 9

Vengono innanzitutto introdotti degli indici di dispersione come l'**RMS** o **scarto quadratico medio** derivato direttamente dalla varianza. Un segnale sismico digitalizzato può essere visto come un insieme di punti che varia attorno ad un valore centrale, uguale a zero in condizioni ottimali della stazione sismica. Possiamo rappresentare tali punti tramite una distribuzione normale.

Sono stati esaminati i valori di *RMS-amplitude* in condizione di quiete di alcune stazioni sismiche dell'Etna, allo scopo di confrontare tali valori con quelli calcolati in fase di previsione di eruzione vulcanica.

Viene proposto un'ottimizzazione dello schema di compressione utilizzato dai digitalizzatori Nanometrics, introducendo un algoritmo predittivo. La procedura presentata è basata su un metodo denominato **Truncated Huffman First-Difference**, un algoritmo che impiega tabelle statiche di compressione. Inoltre si presentano dei protocolli alternativi per cercare una eventuale ottimizzazione dell'utilizzo di banda.

Viene descritto un metodo per la valutazione di un protocollo introducendo il calcolo delle differenze medie previste tra un campione ed il suo precedente.

Capitolo 10

Si descrivono i test effettuati per misurare il consumo di banda di una stazione tipo, utilizzando il tool software *extractp* facente parte del pacchetto **Playback utilities** della Nanometrics®.

Vengono quindi descritti alcuni calcoli eseguiti per permettere il confronto dei protocolli introdotti, verificandone l'efficienza nelle due tipologie di trasmissione radio VSAT e radio UHF.

Appendice 1

Viene illustrato un software, da me realizzato, per l'acquisizione da strumentazione meteo allo scopo di presentare un'applicazione degli studi svolti sul protocollo Nanometrics.

Il software *StreamingClient* è stato realizzato in linguaggio C++ e consiste in un client che si connette ad un qualsiasi server in cui è installato il software **NaqsServer**, per lo streaming dati.

Capitolo 1.

La RSP digitale BB gestita da INGV CT

1.1 La trasmissione VSAT

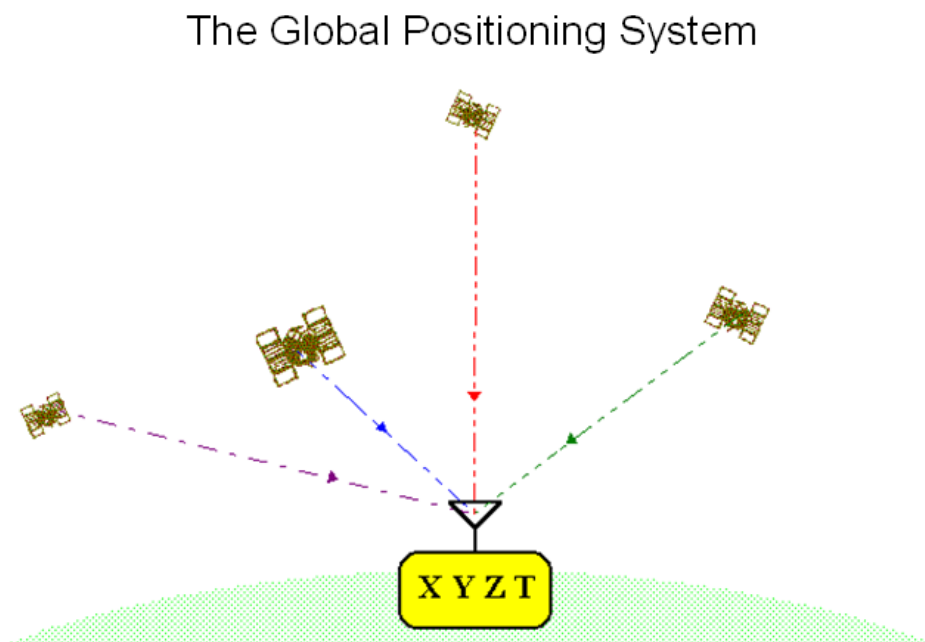
Il termine VSAT, acronimo di Very Small Aperture Terminal, era inizialmente un marchio di fabbrica di una piccola stazione terrestre battezzata nel 1980 dalla *Telcom General* negli Stati Uniti. In effetti il nome, considerando la parte *SAT*, indica il riferimento alla comunicazione satellitare [8].

Da allora l'utilizzo di reti VSAT nelle comunicazioni satellitari è aumentato in modo esponenziale e questo poiché esse sono costituite da piccole stazioni (VSAT per l'appunto) con un'antenna di diametro poco meno di 2,4m (da qui la definizione di *small aperture*); queste ultime non supportano link satellitari con banda elevata ma richiedono dei costi di acquisto e manutenzione alquanto modesti. Inoltre non necessitano di una installazione molto complessa, che può essere effettuata in maniera rapida da tecnici con una ragionevole esperienza nel settore.

La capacità dei link satellitari con una dimensione maggiore si identifica nel range di centinaia di canali telefonici o, in modo equivalente, in una banda di circa 100 Mb/s. Nel caso dei VSAT la capacità scende nell'ordine dei Kb/s, tipicamente da 32 a 119 Kb/s (in modo direttamente dipendente dalla modulazione utilizzata).

I VSAT fanno uso di satelliti orbitanti nel piano equatoriale della Terra ad una altitudine di 35786 Km dalla superficie terrestre. La

posizione di tali satelliti, rispetto alla superficie terrestre, rimane costante nel tempo, poiché si muovono nella stessa direzione e velocità della Terra: da qui il termine *satelliti geostazionari*.



Posizione in tre dimensioni (x,y,z) e GPS Time (T)

Figura 1.1: ricezione del segnale GPS per il posizionamento 3D e il tempo

Le stazioni VSAT sono connesse a tali satelliti utilizzando due radio link (RF) chiamati rispettivamente:

- *uplink*: comunicazione dalla stazione al satellite
- *downlink*: comunicazione dal satellite alla stazione.

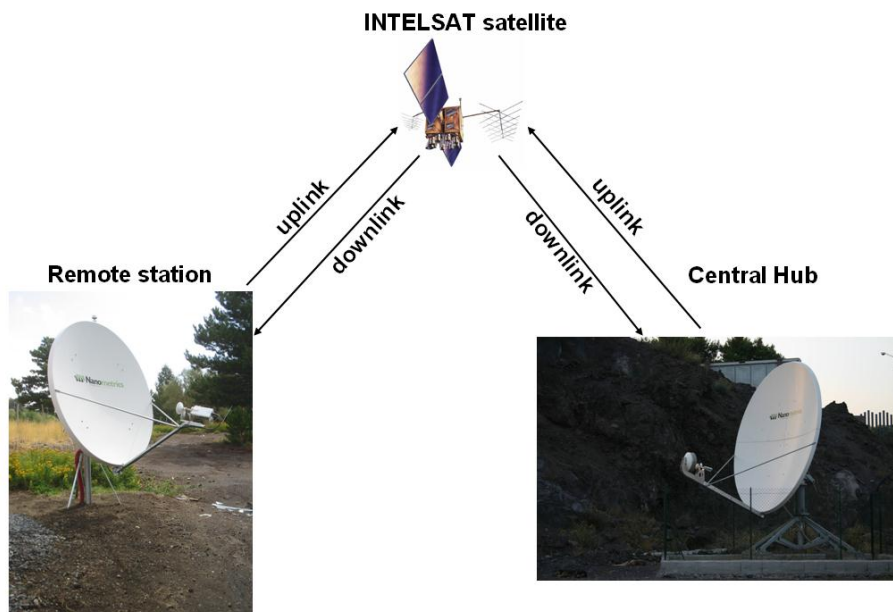


Figura 1.2: link radio up e down

Essendo tutti i VSAT visibili dal satellite geostazionario, ogni VSAT può trasmettere una portante ad un'altra VSAT qualsiasi nella rete utilizzando il satellite.

Questa modalità di trasmissione è però limitata da alcuni fattori che caratterizzano le reti VSAT.

L'elevata distanza da una stazione terrestre ad un satellite geostazionario comporta un'attenuazione della potenza della portante trasmessa di circa 200dB sia sull' *uplink* che sul *downlink*, e un ritardo (*delay*) di propagazione della portante che si aggira intorno agli 0.25 s per singola tratta radio (*uplink o downlink*).

La dimensione ridotta delle antenne utilizzate, limita la potenza di trasmissione e la sensibilità della ricezione del segnale. I **transponder** (canali satellitari) VSAT utilizzano frequenze radio con potenza limitata a poche decine di watts.

Come risultato di quanto detto è immaginabile pensare che il segnale demodulato e ricevuto dai VSAT non corrisponda alla qualità richiesta in acquisizione (elevata presenza di errori nel segnale). Di conseguenza i link diretti da VSAT a VSAT potrebbero non essere accettabili per sistemi di radiocomunicazione.

La soluzione è quella di installare nella rete una stazione molto più grande di una VSAT, chiamata *Hub Station*. La dimensione dell'antenna in questo caso varia da un diametro di 3,5 a 11 m.

Questo incrementa di molto la potenza trasmittente e la sensibilità in ricezione dell'hub, che sarà in grado di ricevere tutte le portanti trasmesse dalle stazioni VSAT e tutte le informazioni che devono convergere ad esse.

L'architettura della rete viene, dunque, organizzata come una *stella* (*star shaped net*) dove il centro stella è rappresentato dall'hub, mentre attorno troviamo tutti i VSAT presenti nella rete.

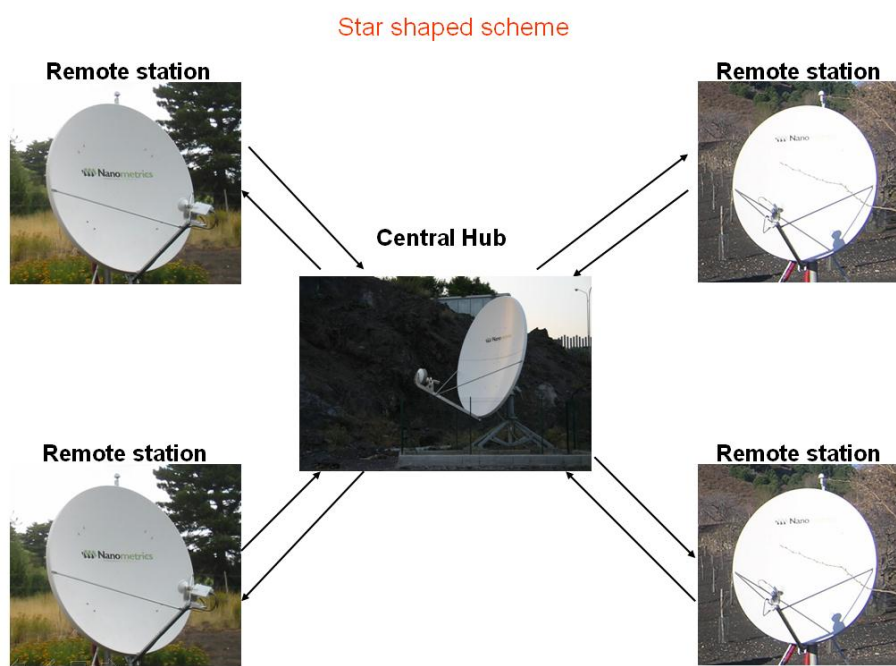


Figura 1.3: lo schema a stella della rete VSAT

1.2 Trasmissione VSAT: Networking

Le stazioni VSAT comunicano utilizzando un unico transponder e quindi un unico canale trasmissivo. In questo tipo di comunicazione è necessario utilizzare un protocollo di *accesso multiplo*.

I più diffusi protocolli sono l'**FDMA** (*Frequency Division Multiple Access*), il **TDMA** (*Time Division Multiple Access*) e il **CDMA** (*Code Division Multiple Access*).

Nel caso specifico il protocollo utilizzato nella Rete Sismica Permanente è il **TDMA** che alloca tutta la banda disponibile ad ogni portante sequenzialmente per un tempo limitato (*time slot*).

La sequenza è sincronizzata in modo tale che i *burst* occupino dei time slot assegnati e mai sovrapposti.

Di conseguenza i time slot sono organizzati attraverso delle strutture periodiche chiamate *TDMA frame* che contengono sequenze di time slot corrispondenti alle stazioni attive. Posizione e durata dei burst sono fissati in base alla domanda di banda di ogni singolo VSAT.

La sincronizzazione delle varie stazioni, distanti fra loro, avviene attraverso l'uso del GPS di cui ogni stazione è fornita.

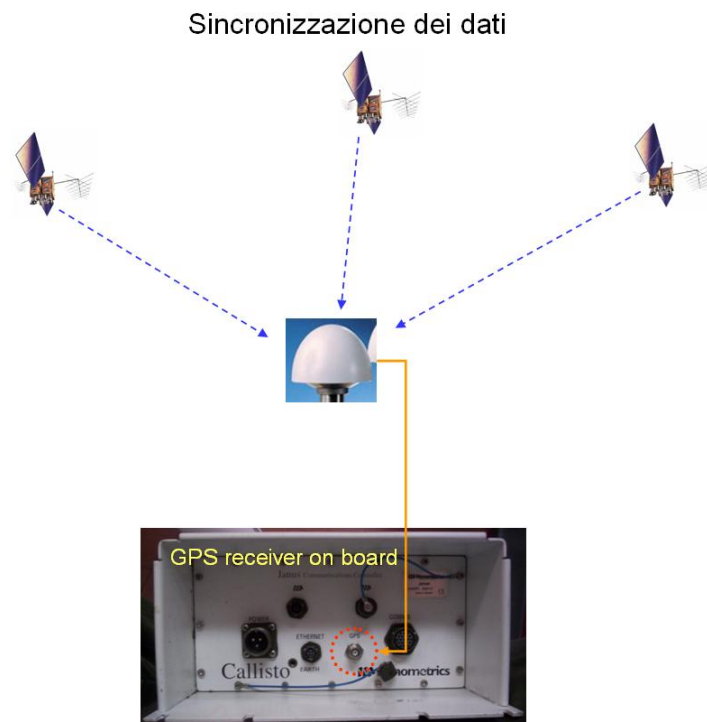


Figura 1.4: Il sistema di sincronizzazione delle strumentazioni Nanometrics®

1.3 La trasmissione radio UHF

Le **onde radio** o **radio-onde** sono le radiazioni elettromagnetiche di frequenza compresa tra zero e 1 GHz, ovvero lunghezza d'onda da 30 cm in su. La quantità di informazione che può essere trasportata da un segnale radio è proporzionale alla sua frequenza.

Le onde radio vengono suddivise in molteplici **bande**, come ad esempio la VHF o la UHF (Ultra High Frequency). Quest'ultima è contenuta in un range di frequenze che va dai 30 MHz ai 3 GHz, ovvero lunghezze d'onda da 10 m ai 10 cm. L'uso di queste frequenze è stato regolamentato e allocato per vari servizi: alle trasmissioni radio FM, ai canali della televisione VHF e UHF. La maggior parte della banda UHF è usata per i collegamenti a microonde e per la telefonia

cellulare. Parte della banda è anche usata per applicazioni di radio-navigazione (strumenti automatici di atterraggio), comunicazioni militari, e controlli radio per il traffico aereo. Infine le onde di questa banda sono usate per trasmissioni terrestri oltre l'orizzonte, utilizzando le irregolarità della troposfera (la parte bassa dell'atmosfera). Queste irregolarità riflettono le onde in tutte le direzioni: i segnali UHF si propagano in molte direzioni e quindi si disperdono su una vasta area, cosicché solo una frazione del segnale trasmesso viene ricevuto dal ricevitore. Inoltre, le condizioni meteorologiche variabili fanno sì che il segnale sia parecchio attenuato. L'assorbimento varia anche con la frequenza del segnale e dipende anche dal particolare percorso del segnale radio. Per questo motivo le trasmissioni possono fare uso di potenti trasmettitori, e di tecniche o combinazioni di tecniche atte a massimizzare la qualità del segnale ricevuto a seconda del posto, del tempo e delle frequenze usate.

Le frequenze utilizzate nella Rete Sismica Permanente sono incluse in un range 435 ÷ 446 MHz con l'eccezione della stazione denominata EBEL che utilizza una frequenza posta sugli 800 MHz.

I dispositivi radio utilizzati sono i Nanometrics[®] Telesto che inglobano, all'interno del case, radio modem a basso consumo e con potenza di trasmissione variabile da 200mW ad 1W.

Le radio possono essere programmate via seriale, per variare la frequenza di trasmissione/ricezione, la potenza di trasmissione e molti altri parametri.

Una coppia radio sulla stessa frequenza crea un link di dati che è condiviso dalle due radio con tecnica TDMA. In pratica il modem

della stazione remota prende possesso ciclicamente del link per 80% del tempo disponibile, mentre il restante 20% è utilizzato dal modem in ricezione, per effettuare le richieste di ritrasmissione. Il tempo ciclico è fissato in 4 secondi come default, ma può essere variato per eseguire una ottimizzazione.



Figura 1.5: il radio modem Nanometrics® Telesto

1.4 Descrizione della Rete Sismica Permanente BB digitale

La RSP BB digitale Nanometrics® gestita da INGV Catania è oggi costituita da:

- 25 stazioni equipaggiate con sensore BB Nanometrics® Trillium 40s, ad eccezione di due di esse che sono equipaggiate da sensore a corto periodo Lennartz® 3D/1s
- 3 centri VSAT senza sensore per concentrare segnali sismici provenienti da altri siti
- 1 centro VSAT acquisizione CUAD con parabola da 3,8m Ø

Geograficamente i siti sono dislocati in modo non omogeneo in tutta la Sicilia orientale e le isole Eolie. In particolare si hanno 15 stazioni

nell'area etnea, 5 nell'altopiano Ibleo, 3 nelle Isole Eolie, 2 nel Peloritano.

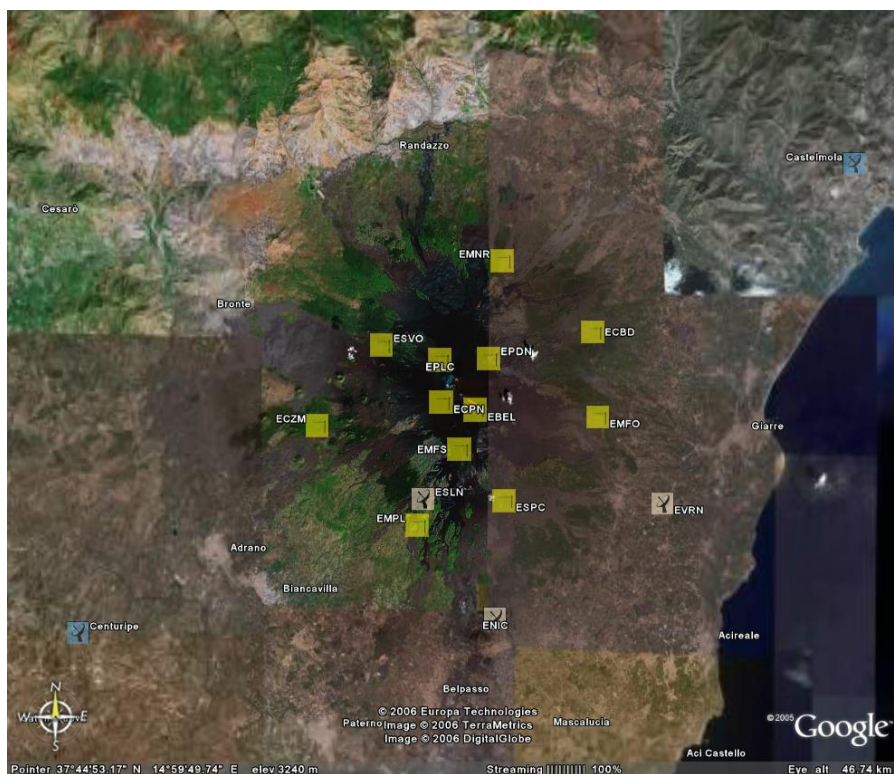


Figura 1.6: Le stazioni della RSP BB digitale Nanometrics® Etna

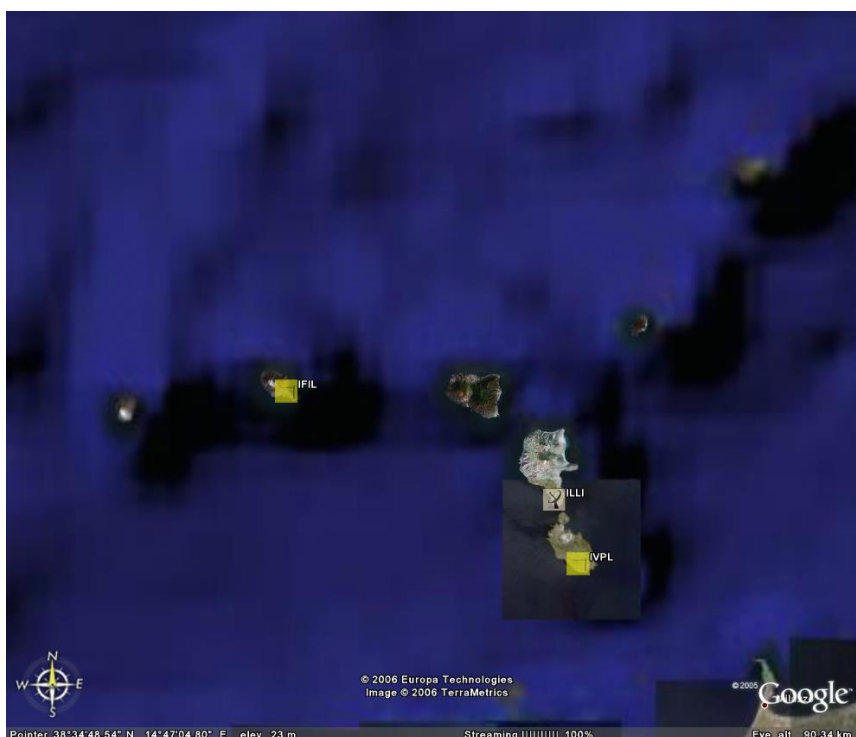


Figura 1.7:L'area Eoliana



Figura 1.8: Le stazioni della RSP BB digitale Nanometrics® nell'altopiano Ibleo

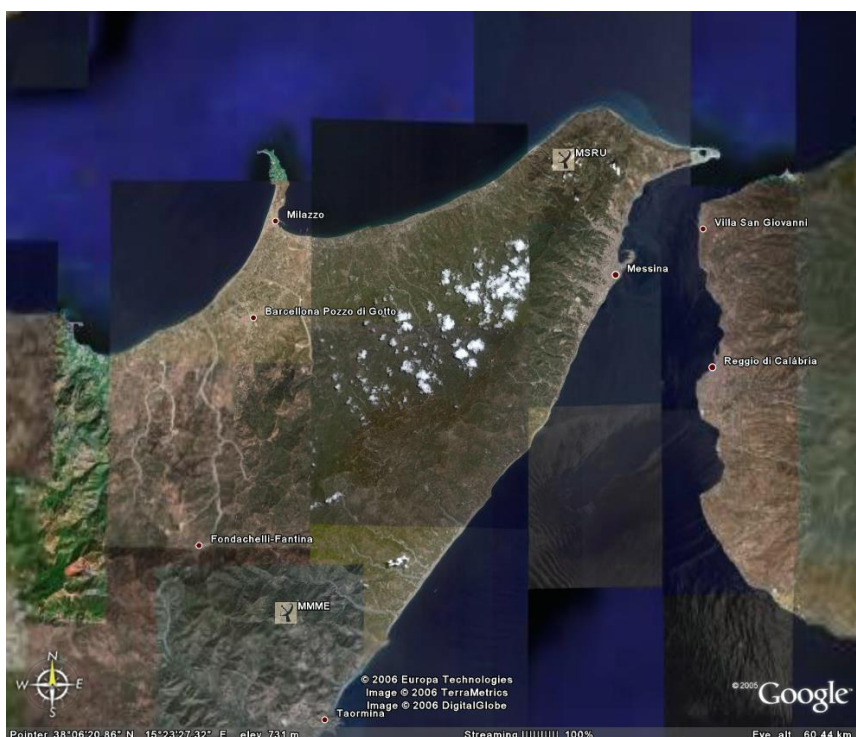


Figura 1.9: Le stazioni della RSP BB digitale Nanometrics® nell'area dei Peloritani

La RSP BB digitale Nanometrics® fa uso di tre tipologie di stazioni:

- Stazioni a trasmissione satellitare
- Stazioni a trasmissione radio UHF
- Ripetitori VSAT

Più precisamente è costituita da:

- 17 stazioni a trasmissione radio UHF
- 8 stazioni a trasmissione satellitare
- 5 ripetitori VSAT (2 con sensore)

Una stazione a trasmissione satellitare è equipaggiata con un sensore che in uscita fornisce una tensione analogica digitalizzata da un Nanometrics® Trident; questi è collegato, attraverso l'*NMX bus*, ad un Nanometrics® Cygnus, un dispositivo capace di bufferizzare il dato

sismico per eventuali richieste di ritrasmissione e che contiene al suo interno un trasmettitore modem satellitare. Il dispositivo è quindi connesso ad un'antenna parabolica da 1,8m Ø o da 1,2m Ø, in funzione del tipo di alimentazione adottata.

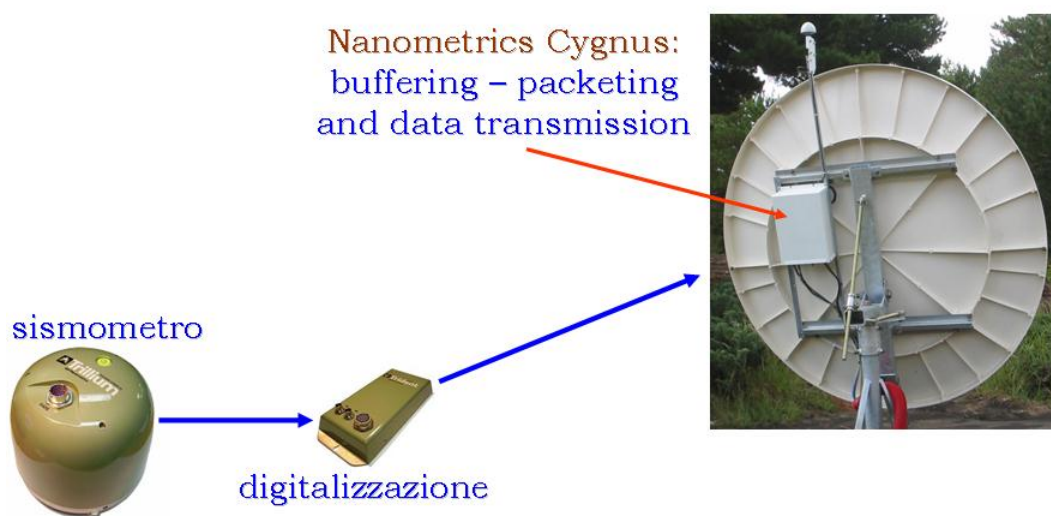


Figura 1.10: Una stazione tipo a trasmissione radio satellitare

Una stazione a trasmissione radio a banda UHF è equipaggiata con un sensore che in uscita fornisce una tensione analogica digitalizzata da un Nanometrics[®] Trident; questi è collegato, attraverso l'*NMX bus*, ad un Nanometrics[®] Janus, un dispositivo capace di impacchettare e bufferizzare il dato sismico per eventuali richieste di ritrasmissione. Come il Cygnus, quest'ultimo è corredato di due porte seriali, di cui una viene impiegata per il collegamento ad un modem radio (Nanometrics[®] Telesto) nella banda UHF. Il dispositivo è quindi connesso ad un'antenna Yagi, accordata alle frequenze di utilizzo.

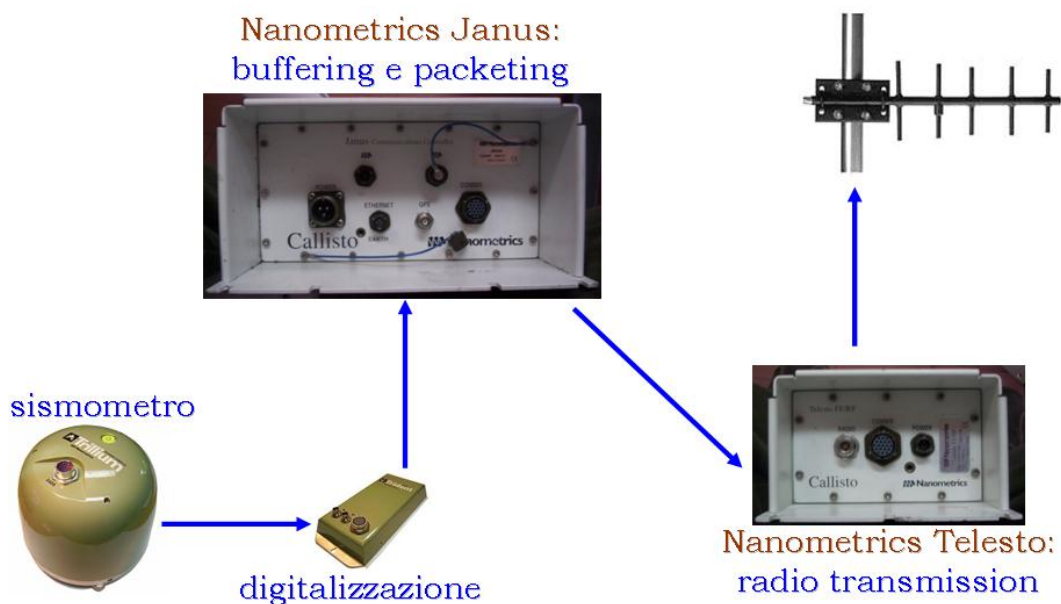


Figura 1.11: Una stazione tipo a trasmissione radio nella banda UHF

Un ripetitore VSAT è equipaggiato con un Nanometrics[®] Cygnus, corredato di due porte seriali a cui vengono collegati due modem radio (Nanometrics[®] Telesto) nella banda UHF, connessi a delle antenne Yagi, accordate alle frequenze di utilizzo.

Il segnale sismico ricevuto da una stazione remota viene quindi bufferizzato e pacchettizzato all'interno del Cygnus che provvederà alla trasmissione al CUAD, utilizzando il link satellitare. Si possono, inoltre, ricevere altre due stazioni remote collegando, con un cavo ethernet cross, il Cygnus ad un dispositivo Janus.

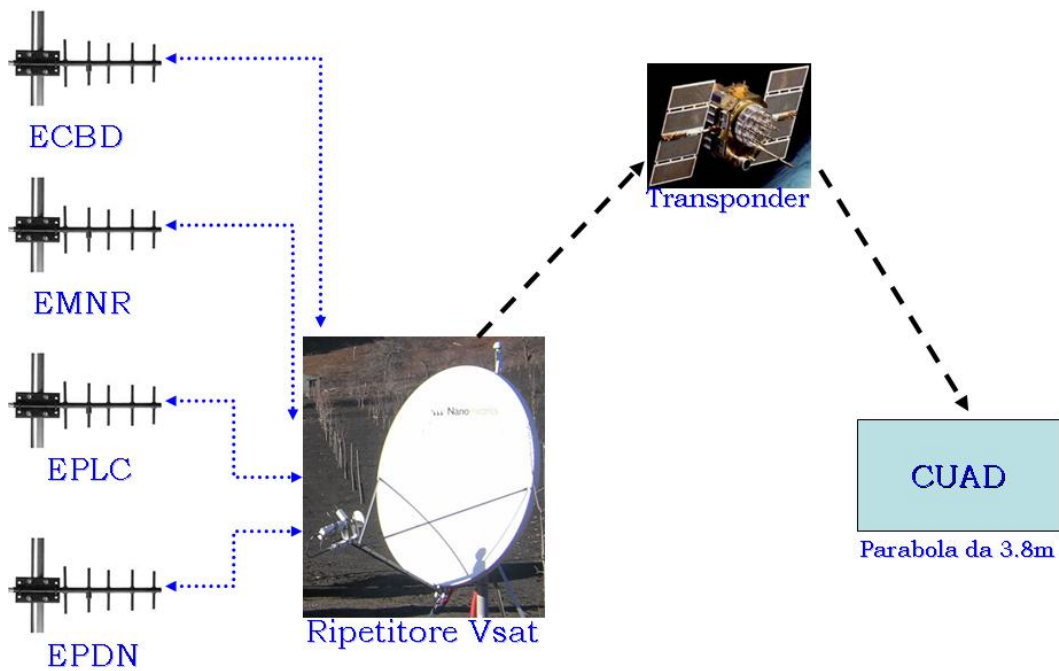


Figura 1.12: Il flusso dei dati di un ripetitore Vsat

L'alimentazione per le stazioni sismiche viene fornita utilizzando un alimentatore con output a 12V continui, usufruendo, dove è possibile, della rete 220V~ oppure, nella maggior parte dei casi, il sito è equipaggiato con pannelli fotovoltaici (4 o 6) associati a 2 o 4 batterie per l'accumulo della tensione, da sfruttare nelle ore notturne o in assenza di insolazione.

Capitolo 2.

Digitalizzazione

Nello studio dei fenomeni terrestri legati al movimento del suolo, ed in particolare riguardo lo studio dei terremoti, da molti anni sono stati utilizzati sistemi di traduzione capaci di legare il movimento della terra, a segnali elettrici. Ciò secondo i principi fondamentali delle leggi fisiche che regolano le forze elettromotrici indotte in magneti permanenti (Legge di Lenz). Lo strumento che effettua questa operazione è il trasduttore o sensore. Il sensore utilizzato nelle stazioni sismiche è un sismometro.

Il sismometro si occupa comunque solo di trasformare il movimento in un segnale *analogico* ovvero in cui i parametri elettrici sono direttamente legati alle variazioni meccaniche delle masse interne. Nella seconda fase il segnale analogico deve essere convertito in un segnale *digitale*. Nella nostra rete vengono utilizzati i sensori della Nanometrics[®] modello Trillium con periodo proprio di oscillazione di 40 s.

Il compito di trasformare un segnale analogico in uno digitale viene svolto da un *convertitore analogico – digitale* o più semplicemente da un *digitalizzatore*.

Nella conversione analogico-digitale il segnale analogico, che assume infiniti valori fra due limiti prefissati, viene convertito in un segnale digitale caratterizzato da un numero finito di valori. Il segnale analogico pertanto subisce un'operazione di *quantizzazione*, che consiste nel sostituire una grandezza analogica, che può variare con

continuità, con una grandezza quantizzata che può assumere solo una serie discreta di valori.

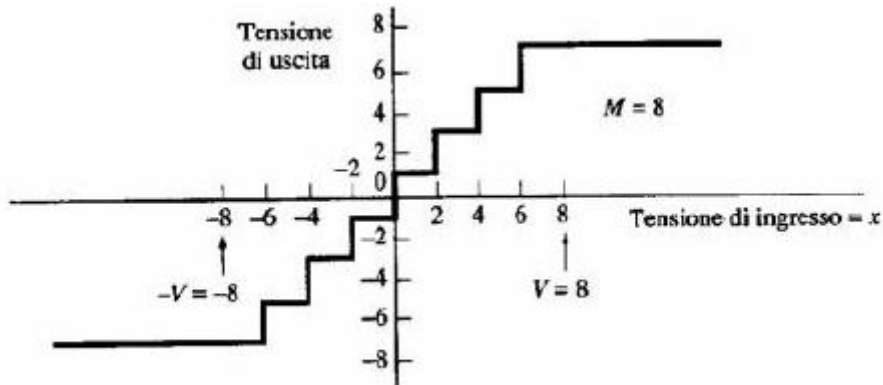


Figura 2.1: caratteristica ingresso-uscita del digitalizzatore

L'ingresso analogico ha un intervallo di variabilità da 0 a FS (valore di fondoscala) e viene convertito in una parola digitale con un numero di bit uguale alla risoluzione del digitalizzatore.

Nella Rete Sismica vengono utilizzati dei digitalizzatori **Nanometrics® Trident** che sono dotati di una risoluzione di 24 bit. Questo significa che l'ingresso analogico viene convertito in una parola digitale a 24 bit, sono quindi possibili $2^{24} = 16.777.200$ valori discreti. L'intervallo da 0 – FS viene pertanto suddiviso in 2^{24} parti, ciascun di ampiezza:

$$Q = FS/2^{24}$$

dove Q è detto *quanto*.

In generale, considerando un convertitore a n bit, il numero possibile di valori discreti sarà 2^n e l'intervallo di variabilità della grandezza analogica 0 – FS verrà suddiviso in 2^n parti [12]; il quanto varrà pertanto:

$$Q = FS/2^n$$

Il quanto corrisponde anche alla cosiddetta **risoluzione** del convertitore A/D intesa come la minima variazione dell'ingresso analogico che comporta, per qualsiasi valore dell'ingresso stesso, una variazione dell'uscita digitale.

Nel processo di quantizzazione è implicita l'introduzione di un errore, detto *errore di quantizzazione* (o anche *rumore di quantizzazione*) ; infatti gli infiniti valori della grandezza analogica, compresi in un intervallo di quantizzazione Q , sono sostituiti dal valore centrale dell'intervallo stesso.

Questo fa sì che l'errore massimo sarà pari a:

$$e = \pm Q/2$$

Si noti come l'errore rimanga costante per tutti gli intervalli in cui è suddiviso il campo di variazione della grandezza analogica; l'errore relativo, cioè il rapporto fra l'errore di quantizzazione e il valore analogico aumenta perciò al diminuire del segnale di ingresso (considerando l'errore di quantizzazione come un rumore si può dire che il rapporto segnale / rumore diminuisce).

Quando si vuole superare questo inconveniente, si ricorre alla quantizzazione non uniforme, la quale prevede che il quanto non viene più preso di valore costante ma venga progressivamente ridotto al diminuire del segnale analogico.

Il digitalizzatore usato nella Rete Sismica permette di scegliere il cosiddetto *range input* che corrisponde a fissare i valori limite del segnale analogico d'ingresso. Il range input in questo modo ci permette di variare il valore di fondoscala e di conseguenza la risoluzione ottenibile.

Il valore range input di default è fissato a **40 Vpp**. I sensori *Nanometrics® Trillium* permettono però una dinamica di **16Vpp** con un conseguente “spreco” di risoluzione. Un'ottimizzazione della funzionalità del digitalizzatore sta quindi nel fissare il valore di range input a 16Vpp in modo da sfruttare tutta la dinamica del sensore e la massima risoluzione ottenibile dal digitalizzatore.



Figura 2.2: il flusso dell'informazione

2.1 Caratteristiche principali

Il digitalizzatore utilizzato nella rete sismica permanente è un *Nanometrics® Trident digitizer*. Il *Trident* riceve come visto sopra, un segnale analogico da sensori con segnale *differenziale* e rilascia dati digitali ad un tranciever come un *Janus o Cygnus VSAT* per la trasmissione al centro acquisizione.

Il *Trident* garantisce una conversione analogico/digitale a 24 bit con un range dinamico di 142 dB. Inoltre esso è dotato di un amplificatore di segnale programmabile (*front –end gain*) per supportare una ampia gamma di sensori. Esso fornisce da uno a sei canali analogici che è possibile digitalizzare, utilizzando un altro *Trident* collegato in cascata è quindi possibile ottenere altri sei canali. È possibile inoltre programmare la frequenza di campionamento del segnale analogico da 10 a 10000 campionamenti al secondo ottenendo uno strumento che riesce a gestire segnali analogici a diverse frequenze di base.

2.2 Filtraggio e Sincronizzazione

Il digitalizzatore include un sistema di filtraggio software del segnale analogico. Il sistema include due filtri: il **FIR** (*Finite Impulse Response*) richiesto per il filtraggio passa basso, ed il **IIR** (*Infinite Impulse Response*) un filtro passa alto per rimuovere componenti continue che creano offset nel dato [4].

Tempo e spazio nelle analisi sismiche sono perfettamente correlati, e proprio per questo il dato deve essere sincronizzato con un'ottima qualità.

Il tempo viene acquisito dal trident per mezzo del cavo di comunicazione *NMX bus* che connette ad un *Time Server* incluso all'interno di ogni tranceiver Cygnus o Janus.

Il *Time Server* è un clock interno disciplinato da un ricevitore GPS. Esso è capace di mantenere la sincronizzazione anche per brevi interruzioni del segnale satellitare. L'utente può comunque decidere se accettare il dato non disciplinato da GPS o decidere di farne a meno e scegliere solo il dato che presenti una sincronizzazione di qualità.

Il **Global Positioning System** (abbreviato in **GPS**, a sua volta abbreviazione di **NAVSTAR GPS**, acronimo di **NAVigation System with Time And Ranging Global Positioning System**), è un sistema satellitare a copertura globale e continua gestito dal dipartimento della difesa (*Departement of Defence, DoD*) statunitense. Il GPS è stato creato a sostituzione del precedente sistema, il *Transit*, quando gli USA hanno rinunciato alla *Selective Availability* ed hanno reso il primo sistema accurato quanto il secondo, ed è supportato da un sistema di 24 satelliti artificiali. Il principio di funzionamento si basa su un metodo di posizionamento sferico, che consiste nel misurare il tempo impiegato dal segnale a percorrere la distanza satellite-ricevitore. Conoscendo l'esatta posizione di almeno 3 satelliti si può avere una esatta posizione 2D (bidimensionale) e occorrono 4 satelliti per avere una posizione 3D (tridimensionale) più il tempo impiegato dal segnale per giungere al ricevitore. E' possibile determinare la

posizione nello spazio del ricevitore stesso. Il GPS è stato inizialmente progettato per scopi militari, ma fin da subito sono apparse evidenti le potenzialità per uso anche civile. Fino al maggio 2000, il segnale per uso civile veniva degradato per ridurre la precisione attraverso la Selective Available (SA), consentendo precisioni nell'ordine di 100-150 m. Da quella data, invece, per decreto del Presidente degli Stati Uniti Bill Clinton, è stata disabilitata la degradazione del segnale, consentendo la precisione attuale di circa 10-20 m. La precisione può essere ulteriormente incrementata grazie all'uso di sistemi come il WAAS (statunitense) o l'EGNOS (europeo), perfettamente compatibili tra di loro. Consistono in uno o due satelliti geostazionari che inviano dei segnali di correzione. L'UE ha in progetto il completamento di una propria rete di satelliti (Galileo) per scopi civili, fra i quali il GPS.

Capitolo 3.

L'algoritmo di compressione utilizzato

I dati digitalizzati nel loro formato di origine sono rappresentati da interi a 3 bytes. Purtroppo questa dimensione non garantisce un efficace utilizzo della banda che è comunque limitata.

Per risolvere tale problematica, si cerca di risparmiare spazio occupato comprimendo i dati. In fase di digitalizzazione un processore si occupa di applicare un algoritmo di compressione al dato *grezzo*.

Esistono due forme di compressione dati: *Lossless* e *Lossy*.

La compressione dati *Lossless* è usata quando i dati devono mantenere la loro integrità, ovvero il segnale di origine può essere ricostruito fedelmente. Tuttavia ci sono dei limiti. Infatti il *Compression ratio* non è molto significativo, variando in genere in un range di $2:1 - 8:1$.

La compressione *Lossy* parte dal presupposto che i dati non devono essere archiviati fedelmente all'originale. Questo tipo di compressione viene usata soprattutto negli archivi video o audio e permette di raggiungere un *compression ratio* più elevato rispetto agli algoritmi *Lossless* (variano in un range del 5 - 20% della dimensione originale).

In entrambe le forme di compressione, le ridondanze presenti nel dato originale sono ridotte usando metodi come *coding*, *pattern recognition* e *linear prediction*. La forma *Lossy* implica uno studio di frequenza sul segnale di origine [13].

Per esempio nei file audio, secondo uno studio psicoacustico, molte frequenze non vengono percepite dall'orecchio umano, o le frequenze con una maggiore energia coprono quelle meno energiche

impedendone la percezione. Riducendo la complessità del segnale di origine si può poi comprimere con un rapporto di compressione di gran lunga maggiore.

L'algoritmo utilizzato dai digitalizzatori *Trident* è di tipo *Lossless compression* e viene implementato eseguendo un **first difference** del segnale di origine.

Applicare un first difference equivale ad eseguire una convoluzione del segnale con il filtro $[-1,1]$ che tende ad appiattire lievemente lo spettro [3].

Capitolo 4.

Calcolo della banda utilizzata(Satellitare e radio)

I link satellitari e radio costituiscono le connessioni tra le stazioni remote e il centro di acquisizione. Conoscere qual è la capienza dei link e il traffico da gestire è fondamentale nella progettazione della Rete. Inoltre la banda utilizzata non è costante, poiché dipende dall'algoritmo di compressione utilizzato, la cui efficienza deriva dal tipo di segnale acquisito.

Il numero di bit trasferiti per unità di tempo su un dato link prende il nome di *bit rate*. È necessaria una distinzione tra **information bit rate (Rb)**, il **channel bit rate (Rc)** e l'**average bit rate (⟨Rx⟩)**.

L'information bit rate è calcolato in base alla quantità di informazioni *utili*, cioè solo quelle di interesse per l'utente finale, contenute nel messaggio.

Il channel bit rate corrisponde all' effettivo bit rate del link sul quale la connessione è attiva. Oltre ai dati (informazioni utili) sono comprese anche informazioni di controllo (segnalazione e correzione degli errori. Di conseguenza il channel bit rate sul link è sempre più grande dell'information bit rate.

Infine l'average bit rate è usato quando il link non è sfruttato in modo costante, ma vi possono essere pause di trasmissione durante le quali il canale rimane inutilizzato. In alcune situazioni può essere utile indicarlo, assieme al channel bit rate o all'information bit rate.

Il **throughput** è l'average rate dell'information rate al quale una connessione nella rete spedisce le informazioni al ricevente.

THROU = <Rb> (bits/s)

Una volta che una connessione è stabilita, il terminale trasmittente è abilitato a trasferire messaggi. Quando il messaggio trasferito corrisponde ad un flusso continuo di dati durante la chiamata, si parla di traffico di tipo **stream**.

Nel caso in cui il messaggio trasferito è caratterizzato da sequenze di piccoli pacchetti, chiamati **burst**, si dice che il traffico è di tipo **bursty** [8].

La Rete Sismica Permanente utilizza un protocollo con traffico di tipo **bursty**.

Quando si vuole calcolare il consumo di banda satellitare o radio di un stazione Nanometrics[®], bisogna utilizzare il tool software **extractp**, richiamandolo da una finestra dos. Un esempio è riportato in figura 4.1.

```
EXTRACTP V4.18
(C) Copyright 1992-2001 Nanometrics Inc.
164813(extract.:0638)I0000 summary mode
164813(extract.:0640)I0000 Input file spec = rlab.bhn
164813(extract.:0641)I0000 Start time = 2006-05-12-17-23-00
164813(extract.:0642)I0000 Start time (longsec) = 1147454580
164813(extract.:0643)I0000 Duration (longsec) = 300
file: C:\ringbuff\RLAB.BHN

C:\ringbuff\RLAB.BHN
Oldest packet index = 0
Newest packet index = 20008
Maximum # of packets = 290691
Packet Size in Bytes = 340
Bundles Per Packet = 19
Data 2006-05-12_17:22:59.6800 0d 0h 5m 2.2400s @ 100 sps. 401: 401-607
Total packets: 207 (0 empty)
Data sections: 1 ( 0d 0h 5m 2.2400s)
Number of gaps: 0 ( 0d 0h 0m 0.0000s)
```

Figura 4.1: L'output del tool extractp

L'output è costituito da una serie di statistiche che, tra l'altro, inglobano il numero di pacchetti inviati nel range temporale richiesto e il numero di bundles per pacchetto.

A questo punto calcolare la banda richiesta per la trasmissione dati è relativamente facile.

Introduciamo ora il concetto di **consumo a campione**:

Esso rappresenta il numero di bytes richiesti per trasmettere un singolo campione di segnale. Si calcola con la seguente formula:

$$((n \cdot \text{pacchetti} - n \cdot \text{empty}) \times \text{width}) / (\text{tempo} \times \text{cps})$$

- **n°pacchetti**: il numero di pacchetti trasmessi o ritrasmessi correttamente nella finestra temporale scelta.
- **n°empty**: quantità di buchi nell'acquisizione all'interno della finestra temporale scelta.
- **width**: nel nostro esempio è il numero di byte di ogni pacchetto formato da 19 bundles.
- **tempo**: durata in secondi della finestra temporale.
- **cps**: numeri di campionamento al secondo.

Per ottenere il **data rate di base** si può moltiplicare per 8 la somma dei consumi a campione delle tre componenti:

$$(\text{byte/sample comp. 1} + \text{byte/sample comp. 2} + \text{byte/sample comp. 3}) \times 8 \times \text{cps}$$

ottenendo quindi un valore in bps. Come si può vedere, si utilizza una formula che esegue un calcolo ridondante; infatti prima si divide e poi, nella seconda formula, si moltiplica per lo stesso valore cps. Se si vuole conoscere la banda per una stazione sismica si può utilizzare invece la seguente espressione:

data rate singola componente (Bytes/s):

$$((n \cdot \text{pacchetti} - n \cdot \text{empty}) \times \text{width}) / \text{tempo}$$

da cui segue il data rate di base (bps):

$$(\sum \text{datarate componenti}) \times 8$$

Per calcolare la banda necessaria, la Nanometrics® raccomanda di aggiungere a questo risultato un 25%, per tener conto degli overheads dei pacchetti (indirizzamento, CRC ed altre informazioni).

In base a questi risultati si possono eseguire delle stime per assegnare la banda potenzialmente occupata dalle singole stazioni [3].

Capitolo 5.

Il server di acquisizione

Ogni stazione sismica della rete permanente trasmette direttamente o indirettamente (utilizzando dei centri VSAT), al centro di acquisizione *CUAD* (Centro Unificato Acquisizione Dati) che si trova a Catania. Una parabola da 3,8m di diametro provvede a ricevere i segnali di tutta la rete e a convogliarli nella strumentazione *Carina*, che di fatto rappresenta l'hub station della rete. Questi ultimi come abbiamo visto, svolgono il lavoro di un vero e proprio gateway, gestendo la connessione virtuale tra la rete satellitare *Libra* e la rete locale dell'Istituto. Il *Carina* inoltre si occupa di instradare i pacchetti ricevuti dalle stazioni remote ad un **server di acquisizione** che gestisce tutti i dati della rete.

“L'interfaccia” fisica tra utente del sistema e la rete Nanometrics® è fondata proprio su questo **server**. Al suo interno sono installati diversi moduli software che “guidano” il funzionamento della Rete. Le sue funzioni principali sono:

- **Acquisizione** dei dati provenienti dalla Rete utilizzando la rete locale e i *Carina*.
- Eventuale **richiesta** dei pacchetti dati da ritrasmettere di qualunque stazione sismica remota.
- **Storage** dei dati acquisiti utilizzando una struttura a coda circolare detta *Ringbuffer*.

- Gestione del controllo dei processi server attivi per aumentare la robustezza del sistema e prevenire l'interruzione dell'acquisizione in real time.
- Architettura **Server** per gestire le richieste di programmi client che desiderano acquisire e processare i dati della Rete.
- Un insieme di **Utilities** per gestire e monitorare il corretto funzionamento della Rete.
- Un insieme di **interfacce utente (GUI)** che velocizzano e semplificano il monitoraggio della Rete.

5.1 Acquisizione e Storage dei dati: Il *NaqsServer*

Il **NaqsServer** è il sistema di acquisizione dati disegnato per *ricevere, processare e immagazzinare* dati seriali, dati sismici e gli *State of Health* del sistema.

I *Tranceiver* (Cygnus o Janus) che si occupano della pacchettizzazione e la bufferizzazione dei dati acquisiscono le informazioni dall'esterno utilizzando diverse "fonti". Nel caso dell'acquisizione dei **dati sismici**, un sismometro o un accelerometro si occupano di trasdurre i movimenti della terra in segnale elettrico analogico, che viene poi codificato in segnale digitalizzato da un digitalizzatore che rappresenta la *fonte* per il tranceiver.

Per quanto riguarda i **dati seriali**, essi vengono introdotti nei tranceiver utilizzando la porta seriale Comms. La *fonte* in questo caso è uno strumento che esegue già la digitalizzazione e utilizza una porta seriale del RS232 per l'exporting dei dati. Nella nostra Rete i

ricevitori GPS della *Leica* funzionano proprio in questo modo. I dati verranno poi inviati al centro acquisizione nello stesso formato in cui sono stati acquisiti.

Oltre ai due precedenti tipi di dati, il tranciever produce ed invia al centro acquisizione il tipo di dati **State of Health**, o più brevemente **SOH**. Traducendo letteralmente si ha lo *Stato di Salute*. Infatti questi ultimi consistono di una serie di informazioni riguardanti il controllo della strumentazione presente nella stazione remota. L'utente potrà usufruire di informazioni che riguardano, per esempio, la temperatura della scheda controller integrata nel tranciever, oppure la tensione delle batterie, il numero di satelliti *locked* dal ricevitore gps interno e molto altro ancora. Gli SOH sono molto utili nel monitoraggio del corretto funzionamento della Rete.

Il sistema NaqsServer è suddiviso in moduli che svolgono diverse funzioni. Nella figura successiva possiamo apprezzarne uno schema descrittivo con il flusso dei dati.

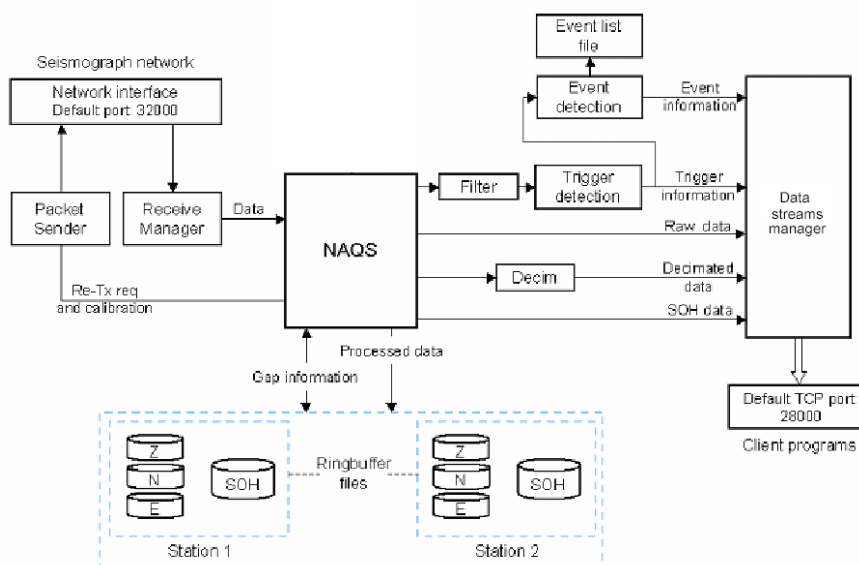


Figura 5.1: lo schema di funzionamento del NaqsServer

Il coordinamento dei vari moduli viene effettuato dal modulo software *Naqs*. Questi si occupa di ricevere i pacchetti dall'interfaccia di rete, individuare il *tag*, che indica la sorgente del canale dati, e lo ridireziona all'archivio corretto. Il *Naqs* individua se nei dati esistono dei gap e in conseguenza a questo si occupa di inviare un messaggio di richiesta di ritrasmissione del dato perso, direttamente alla stazione coinvolta. Inoltre l'utente ha la possibilità di inviare un segnale di calibrazione ad un bene determinato sensore o gruppo di sensori: Il *Naqs* si occupa quindi di inoltrare la richiesta alla stazione o alle stazioni coinvolte.

I dati vengono acquisiti attraverso l'interfaccia network che intercetta tutti i dati provenienti dalla porta 32000 che è la porta di default (quella utilizzata nel server presente a Catania).

Il processo **Receive Manager** si occupa di prelevare i pacchetti ed inviarli al *Naqs* per il loro successivo storage, mentre il **Packet sender** si occupa di trasmettere utilizzando al rete le richieste di trasmissione e calibrazione che gli vengono fornite dal *Naqs*.

I pacchetti acquisiti vengono poi rediretti nel loro archivio corretto. Gli archivi sono organizzati nella memoria hard disk come code circolari chiamati **ringbuffers** (cioè *buffer ad anello*). Esiste un unico ringbuffer per ogni singolo canale acquisito. Ogni ringbuffer viene identificato dal nome della stazione remota, preceduta da una *r* ed una estensione indicante il tipo di dato memorizzato (sismico – seriale – SOH o altro).

Ad esempio vogliamo identificare il canale sismico della stazione sismica *ESPC* ed in particolare la sua componente *Z*. Il sistema,

seguendo uno schema definito in un file di configurazione, creerà un ringbuffer chiamato **ESPC.BHZ** dove **BHZ** identifica la componente Z della stazione.

I ringbuffer essendo delle code circolari allocano memoria in uno spazio costante con tecnica *LIFO*, in questo modo i dati più vecchi vengono per primi sovrascritti da quelli nuovi.

Lo spazio riservato ai ringbuffer è programmabile dall'utente modificando opportunamente un file di configurazione.

Un altro modulo importante è il **DataStream Manager**. Esso provvede a fornire dati in *near real time* a processi client utilizzando il protocollo TCP/IP ed utilizza la porta 28000 come default per mettersi in "ascolto" dei clients. Per ottenere i dati comunica con il modulo *Naqs* richiedendo il canale corretto.

Questi appena enunciati sono moduli fondamentali, che assicurano la funzionalità minima del server. Oltre a questi ne esistono altri che offrono altri servizi meno fondamentali ma importanti. Tra essi troviamo il **Trigger detection** connesso all' **Event Detection** che si occupano, il primo di individuare nel dato sismico acquisito dei *Triggers* con caratteristiche impostate in fase di configurazione, mentre il secondo di elaborare questi triggers per individuare degli eventi sismici in modo automatico. Un altro modulo è costituito dal **Clock Manager** che fornisce l'UTC Time dello stamping dei ringbuffers. Esso apporta delle sincronizzazioni periodiche al tempo fornito, utilizzando un Time server esterno che si trova nella rete locale.

Il NaqsServer è inoltre capace di gestire, in stretta relazione al software **Alert Mailer**, dei *messaggi di allerta* ai gestori del sistema per semplificarne il monitoraggio.

5.2 Accesso ai dati: Il *DataServer*

Il DataServer è un programma di utility che permette l'accesso in locale e remoto ai dati sismici seriali e degli SOH Nanometrics[®], attraverso l'utilizzo del protocollo TCP/IP. In generale quest'ultimo accetta connessioni da programmi client e rispedisce i dati in risposta alla specifica richiesta: accede ai dati immagazzinati nel ringbuffer o su altri file e rappresenta un complemento del NaqsServer online per quanto riguarda la fruizione dei dati in tempo reale.

Il DataServer utilizza due file per la propria configurazione: *Naqs.stn*, per determinare il nome e la locazione dei file di dati a cui bisogna accedere, e il *dataserver.ini* che contiene sezioni relative alla descrizione del socket TCP server che il client usa ed altre info riguardanti modalità di accesso ai dati, descrizione delle directory e nome dei file relativi agli eventi e triggers che devono essere letti.

5.3 Controllo e gestione del sistema: *Playback Utilities*

Le Playback utilities è un insieme di file eseguibili che possiamo distinguere per le loro funzionalità. Alcuni di essi permettono al gestore della Rete o al Server di controllare nel migliore dei modi l'efficienza della Rete Sismica, l'individuazione di eventuali problemi

esistenti e prospettare una soluzione direttamente da remoto. Altri file eseguibili, invece, forniscono un utile strumento per l'elaborazione dei dati acquisiti.

5.4 Monitoraggio della Rete: Il *NaqsView* e il *Waveform Viewer*

Il NaqView fornisce un'interfaccia intuitiva per visualizzare in tempo reale gli SOH estraendoli dal NaqsServer, e per spedire comandi di calibrazione e di centramento della massa dei sensori sismici. NaqsView, inoltre, è utilissimo per graficare lo storico di ogni dato e per osservarne l'andamento temporale. NaqsView comunica con NaqsServer e DataServer utilizzando sockets TCP/IP. In questo modo può essere mandato in esecuzione sia sulla macchina server che su una qualsiasi altra macchina che possieda una connessione al server.

Station	Instrument	Last Update	Voltage	Temperature
HUB.JAN82	JAN082	2006-05-31 13:45:03	14.98	33.3
EPDN.JAN	JAN131	2006-05-31 13:45:00	14.34	12.3
EPLC.JAN	JAN117	2006-05-31 13:45:00	14.09	13.7
ESVO.JAN	JAN107	2006-05-31 13:45:00	13.94	24.9
HMDC.JAN	JAN120	2006-05-31 13:45:00	13.91	42.8
HMLR.CYG	CYG533	2006-05-31 13:45:00	13.91	30.3
HVZN.CYG	CYG468	2006-05-31 13:45:00	13.91	28.3
ECPN.JAN	JAN089	2006-05-31 13:45:00	13.88	16.1
ILLI.JAN	JAN088	2006-05-31 13:45:00	13.84	26.8
ESLN.CYG	CYG566	2006-05-31 13:45:00	13.77	25.4
EMPL.JAN	JAN096	2006-05-31 13:45:00	13.75	25.3
EMFS.JAN	JAN105	2006-05-31 13:44:00	13.72	21.3
ECML.JAN	JAN084	2006-05-31 13:45:00	13.70	24.2
ILLI.CYG	CYG488	2006-05-31 13:45:00	13.70	30.5
HAGA.JAN	JAN127	2006-05-31 13:45:00	13.66	36.4
SSY.CYG	CYG472	2006-05-31 13:45:00	13.66	32.3
ESPC.JAN	JAN113	2006-05-31 13:45:00	13.63	27.9
IFIL.JAN	JAN118	2006-05-31 13:45:00	13.61	34.2
ECML.CYG	CYG512	2006-05-31 13:45:00	13.56	28.2
EMFO.JAN	JAN068	2006-05-31 13:45:00	13.56	26.2
EMNR.JAN	JAN073	2006-05-31 13:45:00	13.52	26.1
EBEL.JAN	JAN129	2006-05-31 13:45:00	13.48	15.7
ECZM.JAN	JAN083	2006-05-31 13:45:00	13.41	28.9
IVPL.JAN	JAN130	2006-05-31 13:45:00	13.31	34.1
ENIC.CYG	CYG565	2006-05-31 13:45:00	13.23	30.9
EVRN.CYG	CYG367	2006-05-31 13:45:00	13.23	26.8
HAVL.JAN	JAN132	2006-05-31 13:45:00	13.17	37.1
MSRU.CYG	CYG492	2006-05-31 13:45:00	13.09	33.3
MMME.CYG	CYG489	2006-05-31 13:45:00	13.08	28.9
ECBD.JAN	JAN076	2006-05-31 13:45:00	13.06	24.3
CNT.JAN	JAN075	2006-05-31 13:45:00	12.88	30.1
CNT.CYG	CYG444	2006-05-31 13:45:00	12.77	35.1
HUB.JAN95	JAN095	2006-05-31 13:45:03	12.48	34.6
HUB2.JAN119	JAN119	2006-05-31 13:45:00	12.42	35.6
HUB2.CAR	CAR091	2006-05-31 13:45:00	11.86	35.3

Figura 5.2: Una schermata delle informazioni visualizzate dal NaqsView

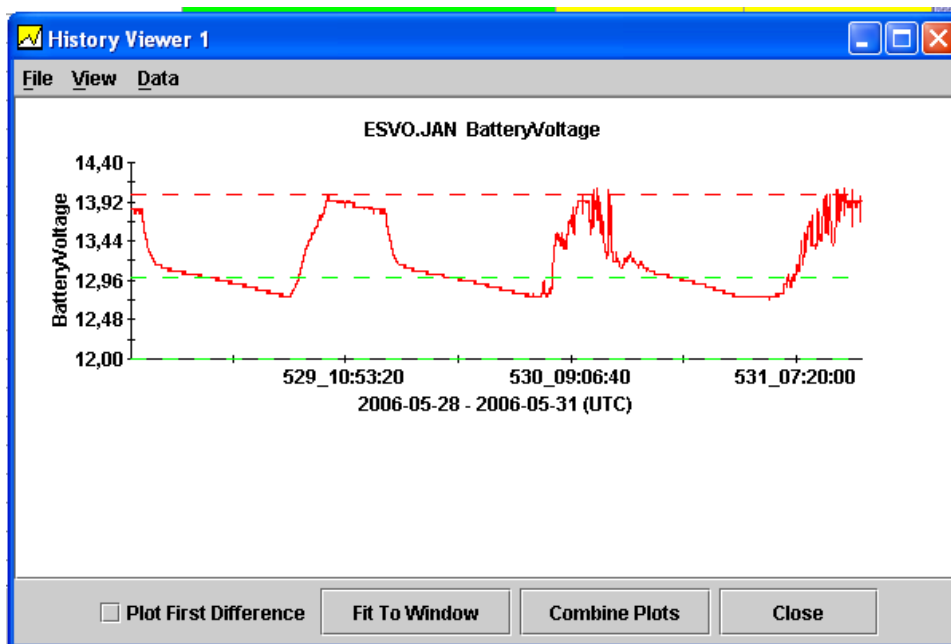


Figura 5.3: Un esempio di storico di una particolare informazione

Il Waveform View fornisce una semplice e intuitiva interfaccia per vedere in tempo reale i segnali sismici estratti dal NaqsServer . Si possono osservare un numero illimitato (solo teorico) di segnali sismici contemporaneamente in un'unica finestra. Tutte le tracce sono plottate usando la stessa scala dei tempi ma ogni traccia può essere personalizzata nella scala delle ampiezze. Anche il Waveform comunica con il NaqsServer attraverso socket TCP/IP usufruendo degli stessi vantaggi sopra descritti per il NaqsView.

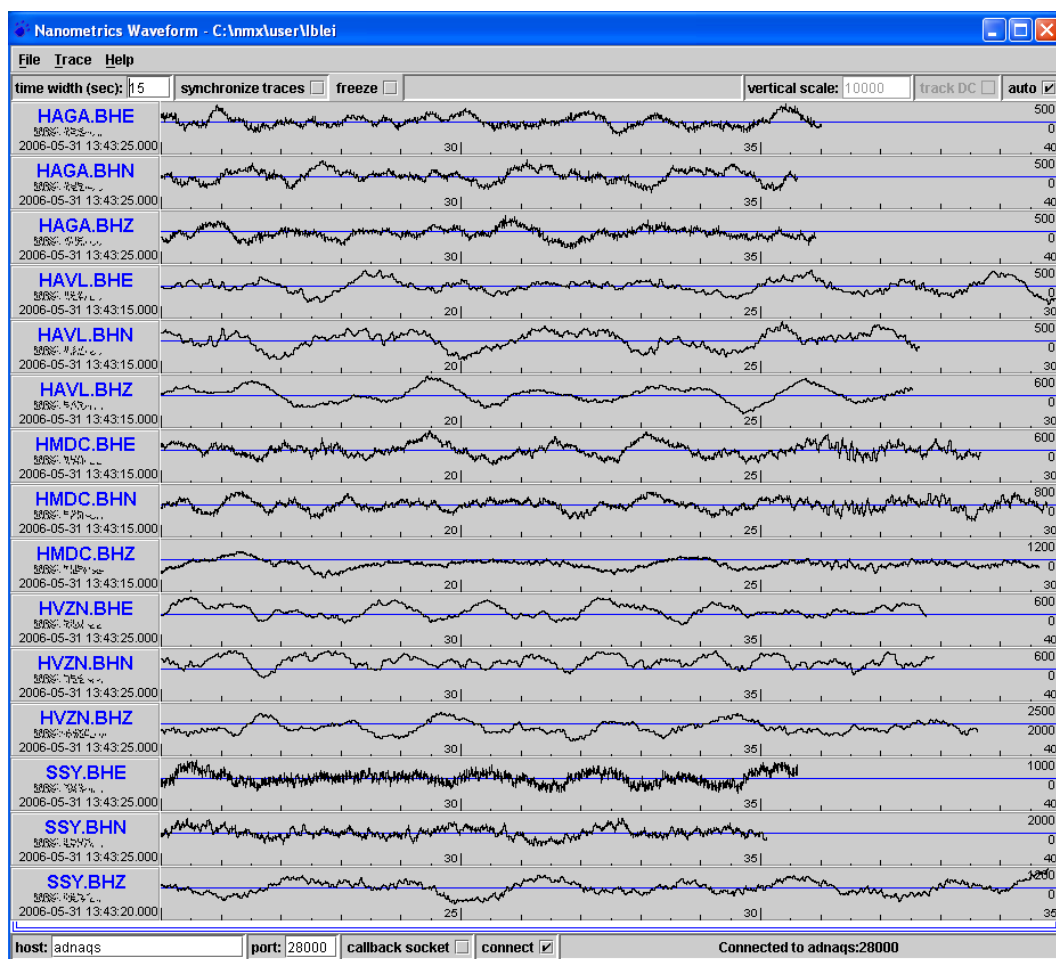


Figura 5.4: I segnali sismici visualizzati con Waveform

5.5 Controllo dei processi server: *NmxWatchDog*

L'*NmxWatchdog* è un servizio Windows che si occupa del monitoraggio continuo dei task e del loro avvio; tali task vengono definiti in un file di configurazione. Esso viene principalmente utilizzato per monitorare il *NaqsServer* e il *DataServer*. Infatti se uno di questi ultimi si blocca per un qualsiasi problema, l'*NmxWatchdog* lo fa ripartire riducendo così i tempi di mancata acquisizione.

Capitolo 6.

Protocollo per l'accesso ai dati di tipo Nanometrics®

Il software *NaqsServer* (vedi cap. 5) è costituito da diversi moduli che provvedono a gestire connessioni client per l'accesso ai dati in near real time.

Per poter accedere a questi dati, è necessario analizzare il protocollo e il formato dei dati per poter creare programmi client che ricevono ed interpretano i dati on line.

I tipi di streaming di dati supportati sono:

- SOH (State of Health)
- Dati sismici (*Time-series data*)
- Dati seriali (*Transparent serial data*)
- Triggers
- Eventi

I dati sismici possono essere richiesti in formato compresso o non compresso. I dati in formato compresso, sono trasmessi nel loro formato di pacchettizzazione originale ricevuto dal digitalizzatore Trident. I dati non compressi, invece, vengono trasmessi come interi a 32 bit. I dati seriali e gli SOH vengono sempre trasmessi in formato compresso.

Inoltre, i dati sismici possono essere gestiti in due modalità:

Raw stream: tutti i pacchetti (compresi quelli ritrasmessi) vengono inoltrati nello stesso ordine in cui essi vengono ricevuti dal NaqsServer. Ci possono essere dei duplicati, dei pacchetti persi, fuori ordine, ma i dati vengono ricevuti con il minimo delay.

Buffered stream: in questo caso i pacchetti vengono ricevuti in ordine cronologico, con dei gap molto corti che verranno colmati con delle richieste di ritrasmissione.

Ogni programma client, implementato per la comunicazione con il server di acquisizione, deve seguire le regole stabilite da un protocollo di *sottoscrizione* definito dall'azienda.

Il protocollo può essere riassunto in 8 passi:

1. Aprire un socket con lo Stream Manager, usando la porta specifica della configurazione del NaqsServer. La porta di default è la 28000.
2. Inviare un messaggio di tipo *Connect* allo Stream Manager.
3. Ricevere un messaggio del tipo *Channel List* che contiene una lista di canali dati a disposizione nel server.
4. Questo step è opzionale. Inviare un messaggio *Request Pending* con pause di pochi secondi (30 secondi). Se questo messaggio non viene ricevuto dallo *Stream Manager* la

connessione termina in *time out*; ne consegue l'interruzione della trasmissione dati da parte dello *Stream Manager*.

5. Inviare un messaggio del tipo *AddChannels* . Lo *Stream Manager* esegue una *sottoscrizione* al canale richiesto che verrà quindi trasmesso.

6. Ricevere dati e spedire un pacchetto handle allo *Stream Manager*. Come optional inviare un messaggio *AddChannels* o *RemoveChannels* per eventualmente cambiare la sottoscrizione. Ripetere questo passo fino a quando si vuole acquisire i dati.

7. Quando si finisce inviare un messaggio di tipo *Terminate Subscription* per cancellare la *sottoscrizione*.

8. Chiudere il socket.

6.1 Descrizione del formato dei messaggi del protocollo di sottoscrizione

Ogni messaggio consiste di un header di 12 byte ed un campo dati di lunghezza variabile. L'applicazione client, legge prima l'header per capire ed interpretare nel modo corretto il messaggio appena ricevuto.

Message Formats

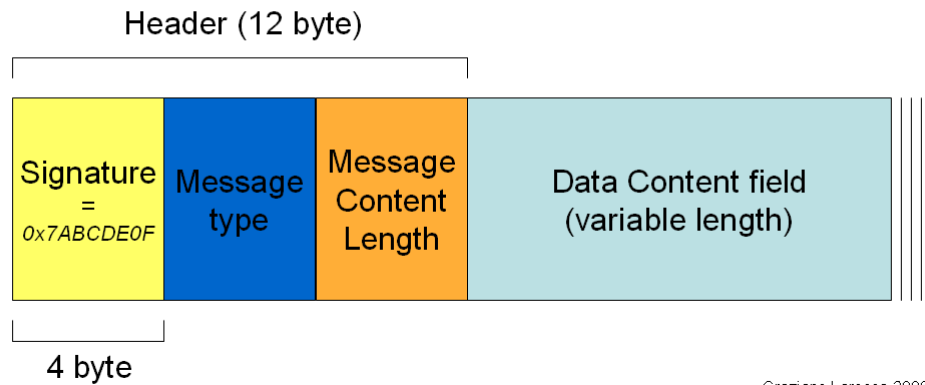


Figura 6.1: lo schema del formato dei messaggi

Come si vede in figura l'**header** dei messaggi è sempre costituito da tre campi da 4 byte ciascuno:

- **Signature:** il codice (*0x7ABCDE0F*) per individuare il messaggio proveniente dallo *Stream Manager*.
- **Message Type:** un intero che indica il tipo di messaggio.
- **Message Content length:** un intero che indica la lunghezza di payload.

Dopo l'header troviamo un campo, il **Data Content Field**, contenente una serie di dati di lunghezza variabile.

6.2 Descrizione del formato dei messaggi del protocollo di accesso ai dati

Esiste un altro protocollo per l'accesso ai dati (diverso dallo streaming in real time), utilizzabile per richiedere dati sismici, seriali State of Health utilizzando la comunicazione TCP/IP.

I programmi client fanno richiesta al software *DataServer* utilizzando un socket con porta **28002** che è quella di default.

Possono essere richiesti dati di tipo Time – series, SOH, Transparent Serial Data, Triggers e eventi.

Ogni tipo può essere richiesto per *canale*, *tempo di inizio* e *tempo di fine*. I tipi Time series, SOH e Transparent serial data vengono spediti nel loro formato originale compresso, ricevuto dagli strumenti di acquisizione che si trovano in esterno. Ogni tipo di dato viene fornito in ordine cronologico.

Il *DataServer* si occupa anche di procurare la lista dei canali dato disponibili. In particolare si può ottenere una *Channel list* che include solo la lista dei canali, o una *Precis List* che, oltre alla lista dei canali disponibili include il tempo di inizio e di fine disponibile nel server per ogni canale.

Ogni canale viene identificato da una chiave a 4 byte. La Channel List e la Precis List forniscono una tabella di referenza per passare da codice chiave a identificativo in formato ASCII dei canali.

Capitolo 7.

Il Formato dati NMX

I dati ricevuti nelle porte seriali della strumentazione Nanometrics® delle stazioni sismiche, vengono pacchettizzati in formato NMX e quindi inglobati in pacchetti UDP per essere trasmessi con una certa priorità.

Il formato richiede un'accurata sincronizzazione per permettere l'etichettatura del dato per la priorità di spedizione.

Il tempo viene fornito da un ricevitore GPS connesso ad una antenna GPS. Il ricevitore è integrato all'interno dei tranceiver.

Ogni pacchetto nel tranceiver viene identificato con una sequenza numerica che specifica il canale dati trasmesso e il numero sequenza più vecchio disponibile per quel canale.

Siccome ogni pacchetto viene numerato, la strumentazione di ricezione del segnale può richiedere i pacchetti non ricevuti correttamente. Il tranceiver si occupa di marcare i pacchetti da ritrasmettere che poi invierà quando la banda disponibile lo permetterà.

In genere la priorità più alta la posseggono i pacchetti più vecchi.

Il numero che identifica la sequenza è un intero a 4 byte che varia da 0 a 2^{32} .

7.1 Inbound Packets e “bundles” per packet

Un pacchetto INBOUND è un dato che viene trasmesso dalla stazione remota al centro di raccolta dati.

Ogni pacchetto è costituito da “bundle” di 17 byte ciascuno.

Un **bundle** è una collezione di dati indipendente per esempio informazione time stamp, dato vero e proprio o informazioni di stato. Il numero di bundles per pacchetto è un parametro programmabile e può variare da 1 a 255. Questo permette di variare la grandezza dei pacchetti in modo da customizzare il protocollo ed adattarlo alle proprie esigenze.

Ad esempio per pacchetti corti dovrebbero essere usati link non efficienti e pieni di errori. La Nanometrics[®] consiglia di utilizzare lunghezze omogenee per la rete sismica o comunque suddivisa in sequenze omogenee.

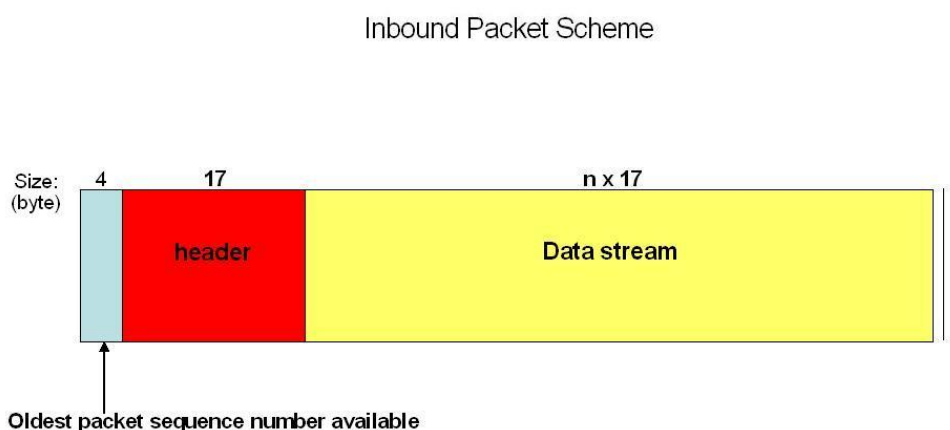
Il dato viene rappresentato nel formato little endian.

7.2 Descrizione dei pacchetti del tipo inbound

L'organizzazione dei pacchetti inbound è generica e rappresenta un gran numero di tipi di pacchetti, che il sistema è portato a gestire. Le differenze maggiori tra un tipo e l'altro risiedono tutte nei byte riservati (*Packet specific*) degli header dei pacchetti.

La loro dimensione è di **21 byte + 17 byte x n**, dove **n** indica il numero di bundles scelto.

Dei ventuno byte iniziali 4 bytes sono dedicati al numero di sequenza del pacchetto più vecchio disponibile per il data stream, mentre i successivi 17 byte costituiscono l'header vero e proprio.



Graziano Larocca 2006

Figura 7.1: Lo schema dei pacchetti di tipo inbound

L'header è, dunque, così costituito:

- *1 byte:* **Tipo di pacchetto**
- *4 byte :* **Tempo in secondi calcolato dal 1 gennaio 1970**
- *2 bytes:* **packet specific**
- *2 bytes:* **ID dello strumento**
- *4 bytes:* **Numero di sequenza**
- *4 bytes:* **packet specific**

Packet Header Scheme

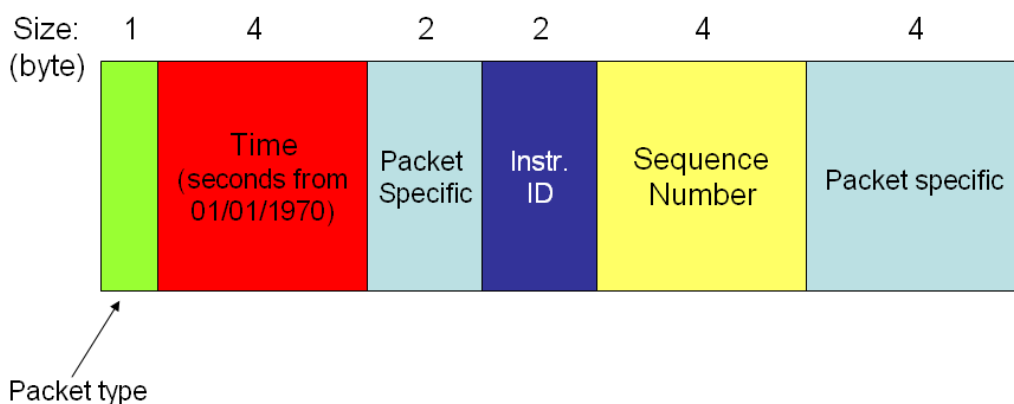


Figura 7.2: Lo schema degli headers dei pacchetti di tipo inbound

Il *tipo di pacchetto* stabilisce il tipo di comunicazione che si intende stabilire e fornisce informazioni per decifrare il pacchetto. Ci sono 59 tipi di messaggio diversi.

Time definito in uno spazio di 4 byte, contiene il tempo assegnato al pacchetto dall'acquisitore nella stazione remota. Infatti si ricorderà che i dati devono essere comunque sincronizzati utilizzando una fonte di tempo che è data da un ricevitore gps. Il tempo viene memorizzato come un unico intero long che rappresenta l'istante presente, considerando zero il primo gennaio 1970.

L'instrument ID (2 byte) indica lo strumento sorgente che trasmette il canale dati. In particolare i bit **0-10** definiscono il serial number dello strumento, i bit da **11-15** invece il modello:

ID	Model
0	HRD
1	ORION
2	RM-3
3	RM-4
4	LYNX
5	CYGNUS
6	EUROPA
7	CARINA
8	TimeServer
9	TRIDENT
10	JANUS
11-31	Reserved for future use

Tabella 7.3: La tabella che contiene gli identificatori della strumentazione

4 byte sono riservati per indicare il numero di sequenza. Facendo un semplice calcolo si possono avere numeri di sequenza che vanno da 0 a 2^{32} cioè da **0** a **4.294.967.296**.

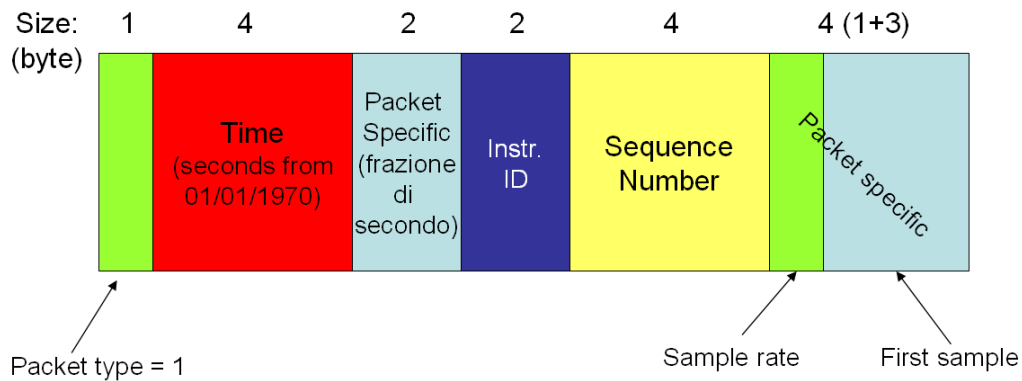
7.3 Inbound packets: Pacchetti compressi di dati

Fra i più importanti tipi di inbound packets, il primo che andiamo a descrivere è il pacchetto che contiene i dati sismici in formato compresso:

in generale il pacchetto è di un header con timestamp seguito da n bundles di dati.

Di seguito è riportato l'header dei pacchetti compressi di dati.

Compressed data packet header



Graziano Larocca 2006

Figura 7.4: L'header di tutti i pacchetti dati compressi

Il *packet type* definisce il tipo **dato compresso**, in questo caso è uguale a 1. Il bit posizione 5 di questo byte indica se si tratta di una prima trasmissione o di una ritrasmissione. In particolare se esso è settato a 1 indica una ritrasmissione.

Time come descritto prima. Il primo *packet specific* da 2 byte contiene una frazione di secondo del timestamp. In particolare le unità sono nell'ordine di 10^{-4} di secondo.

Instrument ID e Sequence number come descritto sopra.

Il secondo *Packet specific* è qui suddiviso in due parti: la prima parte, da un byte, contiene il sample rate utilizzato e l'indicativo del canale trasmesso (in particolare i bit 0-2 indicano il numero di canale, i restanti da 3-7 invece il sample rate). I 3 byte rimanenti contengono X_0 , cioè il primo campione di dati. Il campione viene rappresentato da un *intero con segno a 24 bit LSB first*.

Nella seguente tabella vengono descritti i codici per l'identificazione del sample rate:

codice	Sample rate
0	reserved
1	1 s/s
2	2 s/s
3	5 s/s
4	10 s/s
5	20 s/s
6	40 s/s
7	50 s/s
8	80 s/s
9	100 s/s
10	125 s/s
11	200 s/s
12	250 s/s
13	500 s/s
14	1000 s/s
15	25 s/s
16	120 s/s
17	240 s/s
18	480 s/s
19-31	Reserved for future use

Tabella 7.5: La tabella contenente gli identificativi per il sample rate

7.4 Inbound packets: Bundles di dati

Un bundle di dati contiene da 4 a 16 campioni di dato. I campioni vengono compressi utilizzando un algoritmo di tipo *prima differenza* (*first different algorithm*).

I bundles sono a loro volta suddivisi in quattro set di dati. Ogni insieme contiene il dato compresso rappresentato come una differenza in byte, word o long.

Si possono avere differenze di 4 byte, di 2 word o 1 long ciascuno. Inoltre, ogni insieme contiene 2 *bit di compressione* che indicano le modalità in cui viene pacchettizzata la differenza.

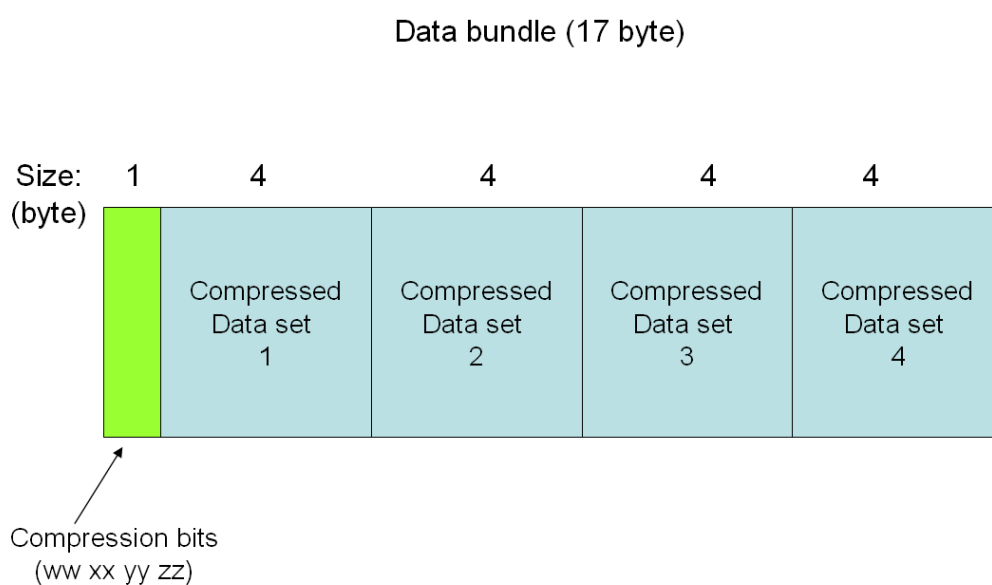


Figura 7.6: Lo schema del bundle Nanometrics

I bit di compressione vengono sistemati nel byte iniziale del bundle seguendo uno schema:

ww xx yy zz

ww: compressione per data set 1

xx: compressione per data set 2

yy: compressione per data set 3

zz: compressione per data set 4

I bit di compressione indicano quanto segue:

00: non usato

01: differenza in byte

10: differenza in word

11: differenza in long

7.5 Inbound packets: Transparent serial packet

I pacchetti di dati seriali hanno un formato leggermente diverso dai dati sismici. La differenza più significativa sta nel fatto che al posto degli n bundles che seguono l'header, qui abbiamo n bytes di dato seriale in formato binario.

Di seguito lo schema del pacchetto

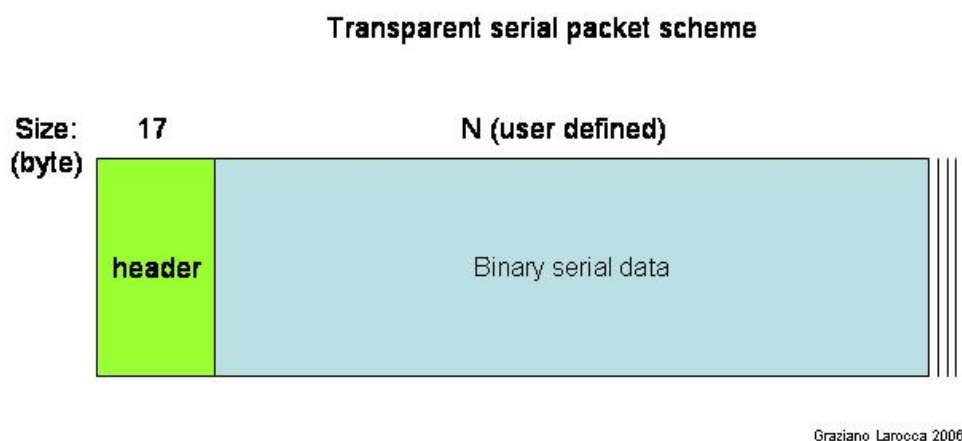
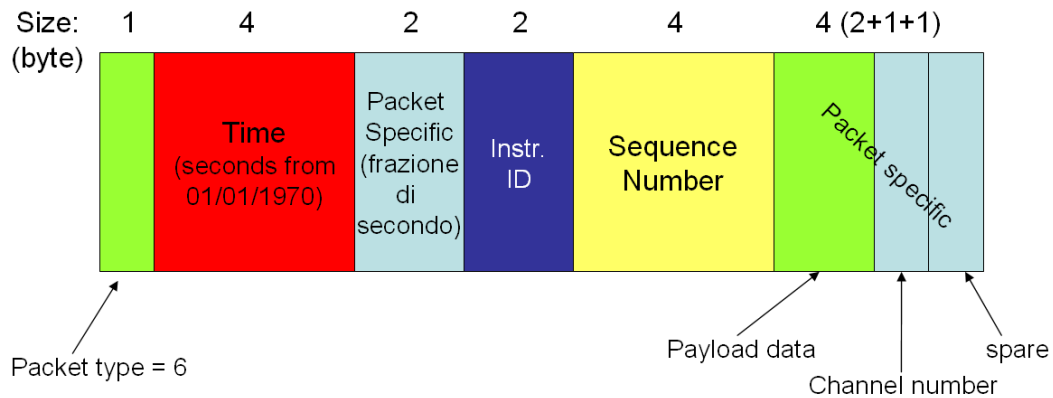


Figura 7.7: Il formato dei pacchetti dei dati seriali

e lo schema dell'header del pacchetto.

Transparent serial packet header scheme



Graziano Larocca 2006

Figura 7.8: L'header di tutti i pacchetti dati seriali

Si nota che il codice *packet type*, in questo tipo di pacchetto, è uguale a 6, mentre il bit posizione 5 indica se il dato è una trasmissione od una ritrasmissione (bit =1 in questo ultimo caso).

Proseguendo abbiamo, come nei formati visti in precedenza, 4 byte per il *time* 2 byte destinati al tempo ad unità di 10^{-4} secondi, 2 byte per *ID dello strumento* e 4 byte per il *sequence number*.

Gli ultimi 4 byte sono specifici di questo tipo di pacchetto e sono così costituiti:

- 2 byte per il *payload data*: indica il numero di byte di dati trasferiti.
- 1 byte per il *numero del canale* che stiamo trasferendo.
- 1 byte *spare*.

Capitolo 8. Il Tremore vulcanico

8.1 Analisi del tremore nell'eruzione dell'Etna del 2001

Tra il 17 Luglio e il 9 Agosto 2001 l'eruzione dell'Etna fu preceduta e accompagnata da sensibili variazioni nel tremore vulcanico. Basandosi sull'acquisizione delle stazioni della rete permanente impiegata nel vulcano. Alcuni ricercatori dell'INGV sezione di Catania, hanno analizzato ampiezza e frequenza dei segnali sismici, riscontrando cambiamenti considerevoli nel tremore vulcanico che indica la transizione a differenti stili di attività eruttiva (fontane di lava, esplosioni stromboliane, attività freatomagmatiche).

Il contenuto in frequenza del segnale decrementava da 5 Hz a 3 Hz nella stazione di riferimento ETF (non più esistente in quanto distrutta dall'eruzione del 2002) durante gli episodi di fontane laviche e continuava a decrementare fino a 2 Hz nella fase di emissione lavica intensa.

L'analisi è stata effettuata usando i dati di sei stazioni, anche queste non più esistenti, posizionate ad una distanza minore di 10 Km dai crateri sommitali. Le ragioni della scelta di queste stazioni sono state due:

1. Sono state selezionate le stazioni con il più lungo periodo continuo di trasmissione di dati nel periodo di luglio – agosto 2001.
2. Inoltre le stazioni erano quelle con il miglior rapporto segnale – rumore.

Esse erano equipaggiate da un sensore con periodo proprio di 1Hz (corto periodo). I segnali venivano trasmessi via radio al centro di acquisizione di Catania dove venivano digitalizzati a 12 bit con una frequenza di campionamento di 160 Hz [1].

Attualmente tali stazioni sono state soppiantate da altre a tecnologia digitale Nanometrics®.

La stazione di riferimento usata, cioè ETF, era posizionata in vicinanza delle spaccature da cui la lava veniva emessa.

Le analisi verificarono che l'ampiezza RMS del segnale sismico fluttuava fino ad uno sciame sismico iniziato il 12 di Luglio. In particolare, l'incremento riscontrato dal 4 al 7 luglio fu associato a due episodi di fontana di lava.

L'inizio dello sciame sismico e delle fontane di lava, durante il 12 luglio, furono associate alla transizione degli alti valori di ampiezza RMS. L'energia sprigionata rimase alta fino al 25 luglio, dal 31 luglio l'ampiezza RMS diminuì fino a ritornare a valori iniziali.

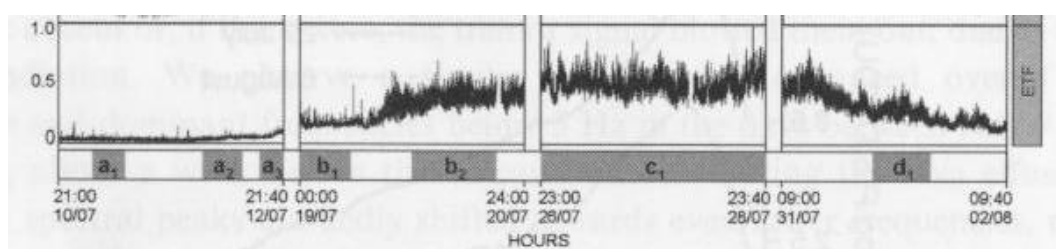


Figura 8.1: Rapporto normalizzato dell'ampiezza RMS del segnale sismico di ETF nelle quattro finestre temporali scelte per l'analisi [1].

8.2 Calcolo del tremore in funzione della distanza

Durante l'eruzione silente¹ del 2004, un gruppo di ricercatori dell'Istituto di Vulcanologia di Catania ha analizzato i dati del tremore vulcanico di otto stazioni sismiche digitali, installate nella sommità del vulcano Etna. Lo scopo dell'analisi è stata quella di creare un modello matematico per localizzare la sorgente del tremore vulcanico. Le otto stazioni prese in considerazione sono state scelte poiché fanno parte della nuova Rete Sismica Permanente a tecnologia digitale Nanometrics[®] con alta dinamica, quindi con sensore Broadband 40s Trillium e digitalizzatore a 24 bit. Inoltre, tali stazioni si trovano tutte in prossimità dei crateri e formano un ring attorno alla zona affetta da fratture eruttive.

Secondo una convenzione, l'attenuazione delle onde sismiche è definita da due fattori, lo spreading geometrico e l'assorbimento. Lo spreading geometrico è proporzionale a $1/s$ (e $1/s^{1/2}$ per le onde di superficie), dove s è la distanza tra la sorgente e la ricezione dell'onda. Sia

$$\alpha = \pi f / Qc$$

dove Q è un fattore di qualità, f è la frequenza dell'onda sismica e c è la velocità. La localizzazione della sorgente sismica viene stimata assumendo una caduta dell'ampiezza sismica in accordo con una legge generale:

¹ Questa eruzione è stata caratterizzata dalla quasi totale assenza di cambiamenti nell'attività sismica, da qui il nome di eruzione silente [2].

$$A(f, s) = A_0(f) s^{-b} e^{(-\alpha s)}$$

dove A è l'ampiezza dell'onda a distanza s e frequenza f , A_0 è l'ampiezza del segnale di riferimento generato alla sorgente, b una costante. Questa può essere riscritta inserendo il logaritmo in entrambe le parti:

$$\ln A_i + \alpha s_i = a - b \ln s_i \quad (8.1)$$

dove A_i è il valore di **ampiezza RMS** misurata all' i -esima stazione, a e b sono delle costanti e s_i è la distanza corrispondente tra sorgente e destinazione.

8.2 Applicazione del calcolo del tremore nella Rete Sismica Permanente Odierna

La legge (8.1) del paragrafo 8.1 risulta fondamentale per poter affrontare un discorso sulla previsione di ampiezza del tremore e consumo di banda in eventuali future eruzioni dell'Etna.

Lo scopo è quello di stabilire una stima approssimata del livello RMS delle stazioni sommitali, oggi presenti nella Rete Sismica Permanente, in un ipotetico caso di eruzione vulcanica con caratteristiche simili all'eruzione del 2001.

Si è voluto considerare proprio l'eruzione del 2001 poiché il livello RMS del tremore vulcanico in quella occasione era da considerarsi straordinario per le documentazioni fino ad oggi in nostro possesso.

In ogni caso bisogna focalizzare l'attenzione a valori di RMS, e quindi di consumo di banda, **massimali** in modo da non avere problemi di misura nelle future eruzioni.

La previsione è stata definita basandosi su tre ipotesi:

- a. come riferimento dell'eruzione 2001, ho considerato i valori RMS-amplitude della stazione ETF, ormai non più presente nella Rete Sismica Permanente;
- b. come riferimento odierno ho valutato la stazione ECPN, poiché la più vicina come coordinate a ETF;
- c. ho esaminato una sorgente del tremore vulcanico localizzata ad una quota di 2500 mslm, nel cuore del cratere Centrale dell'Etna.

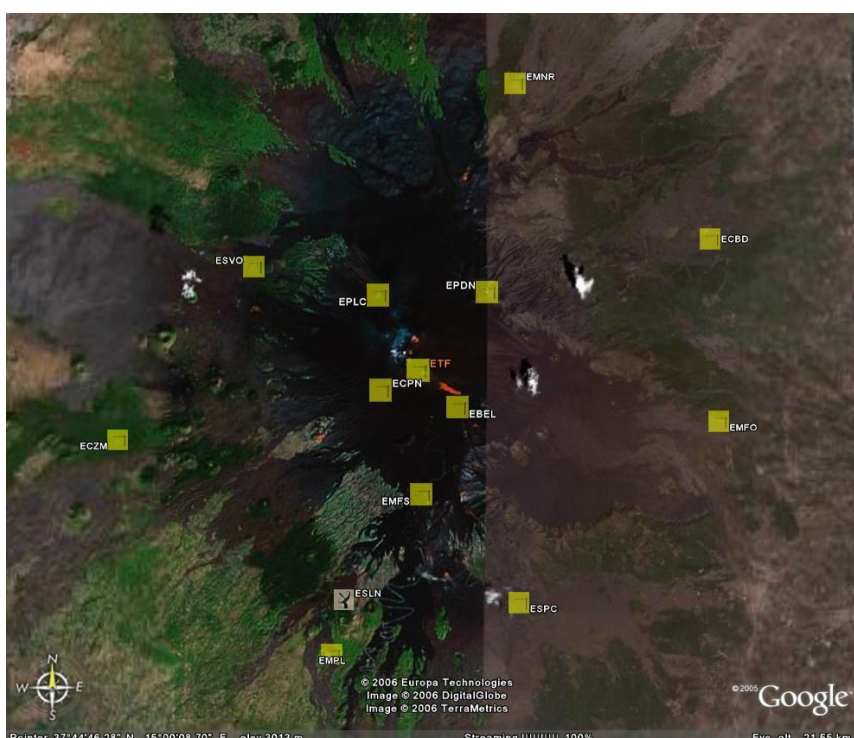


Figura 8.2: Immagine della Rete Sismica Permanente Etna

Sono state considerate 13 stazioni presenti attorno alla sommità dell'Etna, con distanza massima di 8,7 km (ECBD) dall'ipotetica sorgente del tremore.

Oltre quella distanza, le stazioni si ritengono poco influenzate da presenza di tremore vulcanico, così da non causare variazioni sul consumo di banda.

Dopo aver scelto la stazione di riferimento, si sono esaminate tutte le distanze in 3D tra questa e le altre stazioni. Si è applicata la formula e, dopo una conversione, si sono ottenuti i valori di ampiezza RMS previsti.

Stazioni	Lat_N	Lon_E	Quota ⁺	Distanza	d1/d2	RMS(cts)
ECPNZ	4177387	498811	2996	1,3051585		72000
EPDNZ	4179844	501480	2823	2,4540727	0,53183	38292,0254
EPLCZ	4179753	498740	2917	1,7390834	0,75049	54035,0261
EBELZ	4176944	500731	2851	1,800255	0,72499	52198,948
EMFSZ	4174705	499820	2507	3,6019128	0,36235	26089,3088
ESLNZ	4171798	497743	1735	6,8048089	0,1918	13809,5598
ECBDZ	4181434	507617	1419	8,6754697	0,15044	10831,8532
ECZMZ	4176010	491550	1346	8,4487382	0,15448	11122,5384
EMFOZ	4176494	507948	1163	8,6447829	0,15098	10870,3037
EMNRZ	4185489	502289	1797	7,7079519	0,16933	12191,4896
EMPLZ	4170245	497381	1438	8,4220902	0,15497	11157,7308
ESPCZ	4171691	502413	1610	7,2381271	0,18032	12982,8358
ESVOZ	4180646	495330	1691	4,9384841	0,26428	19028,3928

Tabella 8.3: tabella con i risultati dell'RMS-amplitude ottenuti

Dalla tabella possiamo notare il valore impostato come RMS-amplitude nella stazione di riferimento ECPN è equivalente a 72000 counts. Tale valore era quello riscontrato nel 2001 alla stazione di

ETF. La seconda, terza e quarta colonna indicano la posizione della stazione in base alle coordinate e la quota. In base a questi dati si ricava la distanza (quinta colonna) da un punto fissato al centro dei crateri sommitali Etnai ad una quota di 2500 mslm. Nella sesta colonna vengono calcolati i rapporti $d1/d2$, ottenuti dividendo la distanza di ECPN ($d1$) con la distanza di ciascuna stazione ($d2$). Si applica a questo punto la legge (8.1) e si ottengono, nell'ultima colonna, i valori di RMS.

Capitolo 9.

Studio ed implementazione di un nuovo algoritmo di compressione

9.1 Distribuzioni Gaussiane - RMS

Un **indice di dispersione** (o **indicatore di dispersione** o **indice di variabilità** o **indice di variazione**) serve per descrivere sinteticamente una distribuzione statistica quantitativa e, in particolare, il modo con il quale i valori sono distanti dal valore centrale.

In statistica la **varianza** è un indice di dispersione. Viene solitamente indicata con σ^2 (dove σ è la deviazione standard).

L'espressione per la varianza, nell'ambito della statistica descrittiva, è:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$$

dove μ rappresenta la media aritmetica dei valori x_i .

Nel caso si tratti di valori ponderati, allora la definizione diventa:

$$\sigma^2 = \sum_{j=1}^k f_j (x_j - \mu)^2$$

(in questo caso μ è la media aritmetica ponderata).

La varianza è un indicatore di dispersione in quanto è nulla solo nei casi in cui tutti i valori sono uguali tra di loro (e pertanto uguali alla loro media) e cresce al crescere delle differenze reciproche dei valori. Trattandosi di una sommatoria di quadrati, è evidente che la varianza non sarà mai negativa.

Più in generale, se \mathbf{X} è una variabile casuale si definisce la sua varianza come:

$$\text{var}[\mathbf{X}] = E[(\mathbf{X} - E[\mathbf{X}])^2] = E[\mathbf{X}^2] - E[\mathbf{X}]^2$$

essendo $E[\mathbf{X}]$ il valore atteso della variabile casuale \mathbf{X} . Poiché la variabile casuale $(\mathbf{X} - E[\mathbf{X}])^2$ è sempre positiva, il suo valore atteso, ovvero la varianza di \mathbf{X} , sarà anch'esso positivo.

Se μ_1 e μ_2 sono i momenti semplici di ordine 1 e 2 di \mathbf{X} la formula della varianza si riduce a:

$$\text{var}[\mathbf{X}] = \mu_2 - \mu_1^2$$

La Disuguaglianza di Tchebicheff garantisce che almeno il 75% dei valori assunti da X sono compresi tra $\mu-2\sigma$ e $\mu+2\sigma$ e almeno l'88% tra $\mu-3\sigma$ e $\mu+3\sigma$.

In pratica, la varianza risulta essere una misura della dispersione dei valori ottenuti in n prove, ovvero maggiormente differenti sono i valori ottenuti in n prove (ad esempio voti differenti in una prova

d'esame) e maggiore sarà il valore della varianza. Infatti è chiamata anche indice di dispersione, dato che offre una indicazione sull'addensamento dei valori della variabile attorno al valor medio. Se dunque abbiamo una varianza alta in una segnale sismico, questo vorrà dire che ci saranno valori molto differenti fra di loro; ne consegue che il rapporto di compressione sarà basso. Viceversa, se la varianza ha un valore basso, i valori acquisiti sono pressoché equivalenti, e quindi la compressione sarà molto più efficiente.

L' **RMS** o **scarto quadratico medio** è un indice di dispersione (vale a dire una misura di variabilità di una popolazione o di una variabile casuale) derivato direttamente dalla varianza, che ha la stessa unità di misura dei valori osservati (mentre la varianza ha come unità di misura il quadrato dell'unità di misura dei valori di riferimento). L'RMS (o anche deviazione standard) misura la dispersione dei dati intorno al valore atteso. Se non indicato diversamente, è semplicemente la radice quadrata della varianza, la quale viene coerentemente rappresentata con il quadrato di sigma (σ^2).

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

dove $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ è la media aritmetica.

Un segnale sismico digitalizzato può essere visto come un insieme di punti che varia attorno ad un valore centrale, uguale a zero in condizioni ottimali della stazione sismica. Possiamo rappresentare tali punti tramite una distribuzione normale.

La legge di distribuzione normale è caratterizzata da una densità di probabilità della forma:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

dove μ = media σ = scarto quadratico medio.

La curva descritta da tale funzione ha una forma caratteristica "a campana": tale curva è centrata sul punto di ascissa $x=m$ e in corrispondenza di esso ha il suo massimo in ordinata pari a $\frac{1}{\sigma\sqrt{2\pi}}$. Il parametro σ è correlato alla larghezza della "campana" e, in particolare rappresenta la distanza tra l'asse di simmetria e i punti di flesso della distribuzione. Se σ è piccolo la curva è stretta, se σ è grande la curva è larga e più "dispersa" rispetto al valor medio m . In particolare il parametro σ non è altro che la deviazione standard della distribuzione, così come m rappresenta il valor medio della funzione.

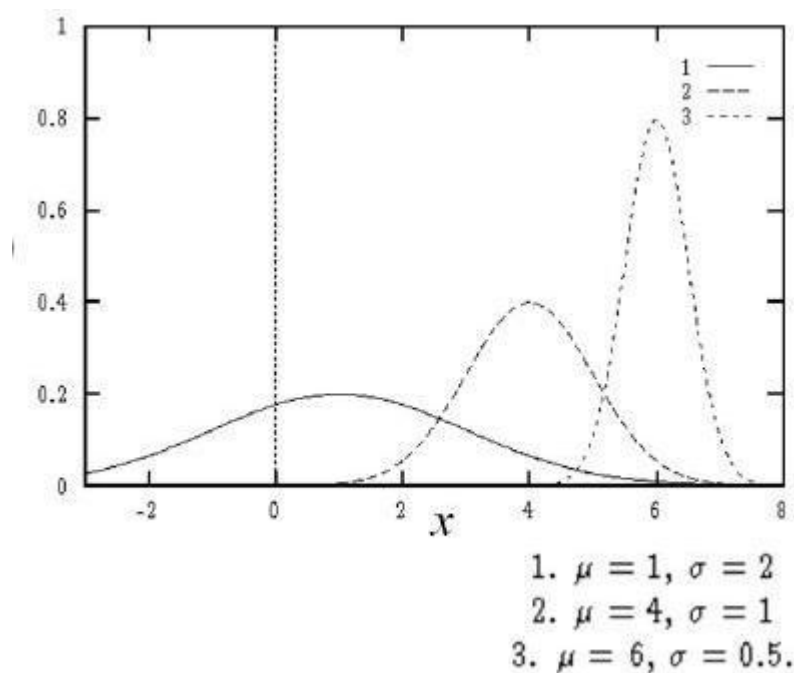


Figura 9.1: Esempi di Gaussiana con diversi valori di σ

Solitamente la distribuzione normale viene indicata con $N(m, \sigma)$, dove m e σ sono i due parametri che caratterizzano la distribuzione. Molto spesso si usa anche la distribuzione cosiddetta "standardizzata" $N(0,1)$ avente media 0 (centrata sull'origine degli assi) e deviazione standard pari ad 1.

9.2 Deviazione standard nella situazione di quiete

Sono stati esaminati i valori di RMS-amplitude in condizione di quiete di alcune stazioni sismiche dell'Etna, allo scopo di confrontare tali valori con quelli calcolati in fase di previsione di eruzione vulcanica. Nanometrics[®] Waveform mi ha dato la possibilità di ottenere in real time questi valori. A tal fine, si è osservata una finestra temporale di

300 secondi (5 minuti) di 13 stazioni in data 06/06/06 intorno alle 10.35 locali e in data 14/06/06 (inizio alle 18.40).

I risultati RMS-amplitude calcolati in *counts* sono stati i seguenti:

Stazione	Componente E/W	Componente N/S	Componente Z
ECPN	8605	7687	3440
EPLC	4427	5850	4270
EBEL	4080	3857	2015
EPDN	3837	3805	1985
EMPL	2100	1872	1122
EMFS	1712	1672	887
ESVO*	1505	1847	940
ECBD	1260	1100	687
ESLN	950	922	655
ILLI	945	985	647
EMFO	805	802	457
EMNR*	801	858	430
ESPC	587	850	885
ECZM	540	722	495
HAGA	517	557	427
EVRN	457	655	217
HAVL	377	362	322
HMDC	345	400	355
IVPL	332	282	230
HVZN	330	290	225
SSY*	255	423	228
MSRU	230	192	172
MMME	215	247	207

Tabella 9.2: I valori RMS-Amplitude misurati nella Rete Sismica Permanente

NOTA: I valori in counts delle stazioni sono state normalizzate a 1count/ μ V eccetto per le stazioni contrassegnate da un asterisco, che già sono predisposte con questo fattore di conversione.

9.3 Teoria di Huffman e First Difference

Utilizzare un algoritmo di compressione che si occupa solo di memorizzare differenze ha i suoi vantaggi ma soprattutto molti limiti.

Infatti questo tipo di compressione cambia il suo rendimento in base all'ampiezza e alla frequenza del segnale di ingresso.

Un miglioramento nella media del segnale acquisito sarebbe quello di poter "predire" il segnale di ingresso.

Molti algoritmi vengono detti appunto *predittivi* proprio perché, dopo un'analisi del segnale da trasmettere, si può ottenere una distribuzione di probabilità dei punti che possono essere acquisiti.

Questi algoritmi derivano dalla teoria di codifica di **D. A. Huffman** che nel 1952 descrisse un metodo di costruzione di un codice a ridondanza minima [16].

I codici di Huffman costituiscono una tecnica diffusa ed efficiente per la compressione dei dati. Generalmente, a seconda delle caratteristiche del segnale che deve essere compresso, questi codici permettono di risparmiare una quantità di spazio che va dal 20% al 90%. L'algoritmo proposto da Huffman per definire la strategia ottimale di rappresentazione di ogni carattere come stringa binaria, fa uso di una tabella che contiene la frequenza e le occorrenze di ciascun carattere.

L'idea di base è che i campioni del segnale di partenza vengono codificati con stringhe di bit di lunghezza variabile, in funzione

inversamente proporzionale alla loro probabilità di emissione: più precisamente si assegnano meno bit ai simboli più probabili e più bit ai simboli meno probabili rispetto alla frequenza media.

Una ulteriore ottimizzazione consiste nell'utilizzo dell'algoritmo *first difference* prima di applicare la codifica di Huffman. La *first difference* trasmette, al posto del singolo campione, la differenza tra il campione stesso ed il suo predecessore:

$$b'_i = b_{i+1} - b_i$$

In tal modo i valori genereranno una distribuzione gaussiana con valore medio uguale a zero e con larghezza pari a σ .

9.4 Pseudocodifica algoritmo predittivo proposto

La procedura da me proposta è basata su un metodo denominato **Truncated Huffman First-Difference**, un algoritmo che impiega tabelle statiche di compressione. Queste ultime, in genere, sono di grandi dimensioni e quindi richiedono una grossa quantità di memoria. Per evitare ciò, si propone di utilizzare tabelle di dimensione più piccole. In particolare userò una tabella da 4096 entries, con indici a 12 bits.

L'idea di base è che ogni campione input b_i viene sottratto dal campione precedente ed il risultato è usato come indice nella tabella di codifica opportuna. Il primo campione b_0 è presente nell'header del pacchetto dati.

Di seguito viene presentata la pseudocodifica proposta.

Truncated Huffman (b_0)

$i \leftarrow 1$

$j \leftarrow 0$

truncated $\leftarrow -1$

while (sizepack > 0)

Passo 1:

if ($i = 1$)

differenza $\leftarrow b_i - b_0$

else

differenza $\leftarrow b_i - b_j$

Passo 2:

if (differenza) ≤ 2048 e ≥ -2048

inserire nella tabella di Huffman

sizepack \leftarrow sizepack - 2

else

output \leftarrow prefissoHuffman + binarycode

sizepack \leftarrow sizepack - 4

truncated $\leftarrow i$

passo 3:

$i \leftarrow i + 1$

if (truncated < 0) $j \leftarrow j + 1$

else

if ($j < i - 1$)

$j \leftarrow j + 2$

truncated $\leftarrow -1$

Effettuato il first difference al passo 1, si procede al passo 2 al controllo del range in cui ricade ogni singola differenza ottenuta.

Se ricade nel range $[-2048 \div 2047]$, la differenza viene utilizzata come indice della tabella a 4096 elementi; in caso contrario il campione viene escluso dai successivi calcoli della differenza ed il suo valore viene trasmesso come valore binario puro con un prefisso prescelto di Huffman.

Di seguito una procedura per la costruzione di una codifica di Huffman.

Nella procedura seguente si suppone che C sia un alfabeto di n caratteri e che ogni carattere $c \in C$ sia un oggetto con una frequenza indicata da $f[c]$. Una coda con priorità, con chiavi basate su f , viene utilizzata per identificare i due oggetti con la frequenza più piccola da fondere insieme. Il risultato dell'operazione di fusione è un nuovo oggetto la cui frequenza è data dalla somma delle frequenze degli oggetti che sono stati fusi insieme.

Huffman(C)

```
1.  $n \leftarrow |C|$ 
2.  $Q \leftarrow C$ 
3. for  $i \leftarrow 1$  to  $n-1$ 
4.     do  $z \leftarrow \text{Allocate-Node}()$ 
5.          $x \leftarrow \text{left}[z] \leftarrow \text{Extract-Min}(Q)$ 
6.          $y \leftarrow \text{right}[z] \leftarrow \text{Extract-Min}(Q)$ 
7.          $f[z] \leftarrow f[x] + f[y]$ 
8.          $\text{Insert}(Q, z)$ 
9. return  $\text{Extract-Min}(Q)$ 
```

La linea 2 assegna alla coda con priorità Q i caratteri in C .

Il ciclo *for* (linee 3 – 8) rimuove dalla coda i due nodi x e y con la frequenza più bassa ed inserisce nella coda un nuovo nodo z che è il risultato della fusione di x e y (linea 7).

Il nodo z ha x come figlio sinistro e y come figlio destro (questo ordine è arbitrario, scambiando fra loro i figli di sinistra e di destra di ogni nodo si ottiene un codice differente ma con lo stesso costo).

Dopo n-1 iterazioni di fusione la procedura restituisce come risultato (linea 9) l'unico nodo che rimane della coda: la radice dell'albero del codice.

Dalla teoria della complessità si ricava che l'algoritmo di Huffman presenta un costo di esecuzione pari a

$$O(n \log n) [9]$$

9.5 Ottimizzazione del data bundle Nanometrics

Da un'attenta osservazione del protocollo Nanometrics, ho cercato di trovare una soluzione diversa per ottimizzare il consumo di banda per installazioni prossime ai vulcani attivi, esaminando i dati delle eruzioni Etnee.

Di seguito riporto lo schema dei data bundles Nanometrics allo scopo di confrontarli con le soluzioni alternative da me proposte.

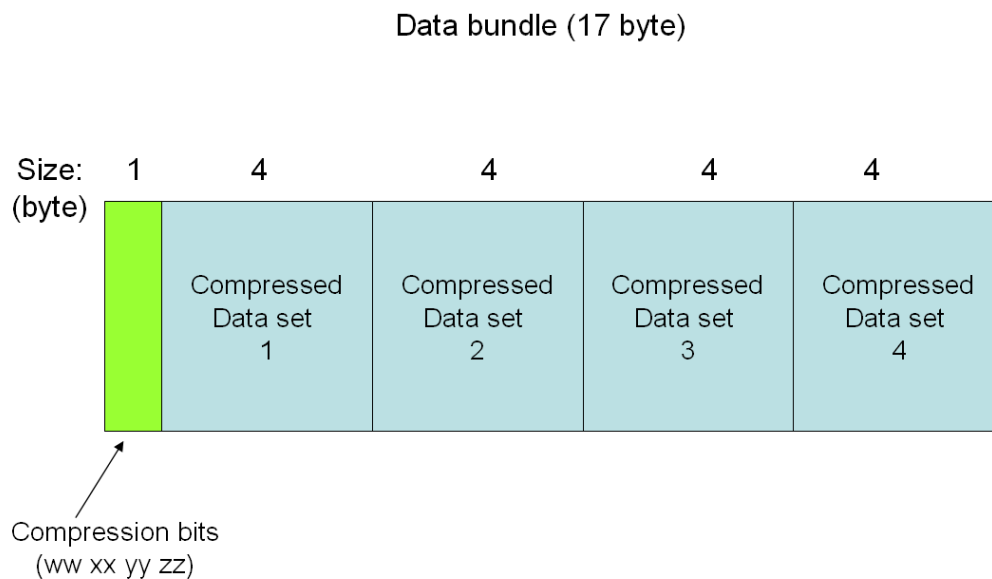


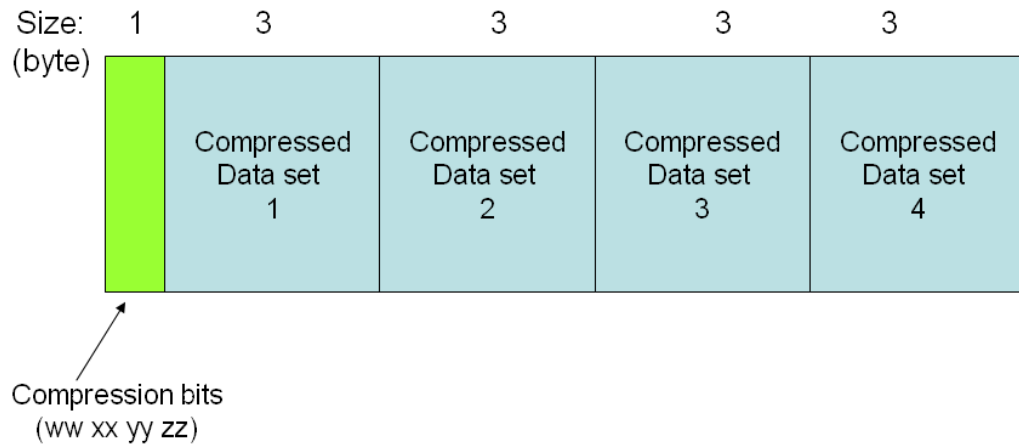
Figura 9.3: Nanometrics data bundles

Si nota che il data bundle è di 17 byte e per ciascun *compressed dataset* si può avere una tra le tre seguenti soluzioni:

- 4 campioni da 1 byte
- 2 campioni da 2byte
- 1 campione da 24bit

La prima soluzione, che chiameremo **soluzione321**, da me proposta prevede che il data bundle sia così composto:

Soluzione321 (13 byte)



Graziano Larocca 2006

Figura 9.4: Lo schema del protocollo soluzione321

Si nota che il data bundle è di 13 byte e per ciascun *compressed dataset* si può avere una tra le tre seguenti soluzioni:

- 3 campioni da 1 byte
- 2 campioni da 12 bit
- 1 campione da 24bit

I bit di compressione indicano quanto segue:

00: non usato

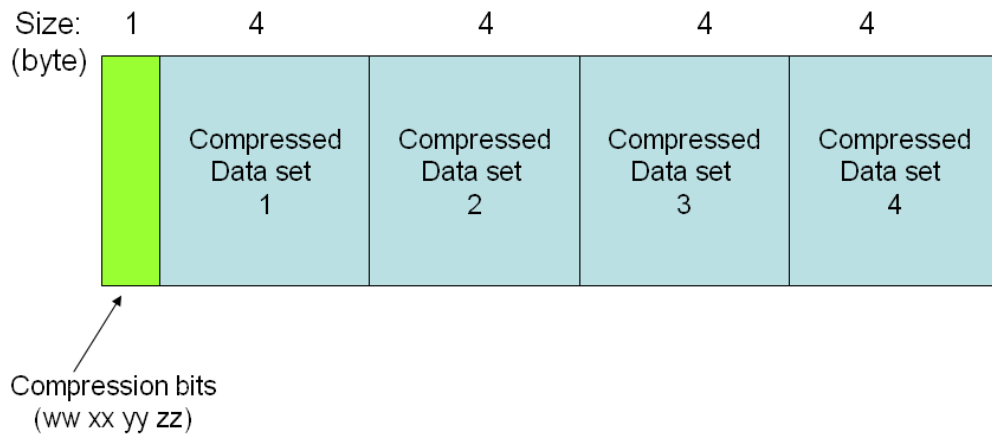
01: differenza in byte

10: differenza a 12 bit

11: differenza a 24 bit

La seconda soluzione, che chiameremo **soluzione431**, da me proposta prevede che il data bundle sia così composto:

Soluzione431 (17 byte)



Graziano Larocca 2006

Figura 9.5: Lo schema del protocollo soluzione431

In questo caso il data bundle è di 17 byte e, per ciascun *compressed dataset*, si può avere una tra le tre seguenti soluzioni:

- 4 campioni da 1 byte
- 3 campioni da 12 bit
- 1 campione da 24bit

I bit di compressione indicano, analogamente alla soluzione precedente, quanto segue:

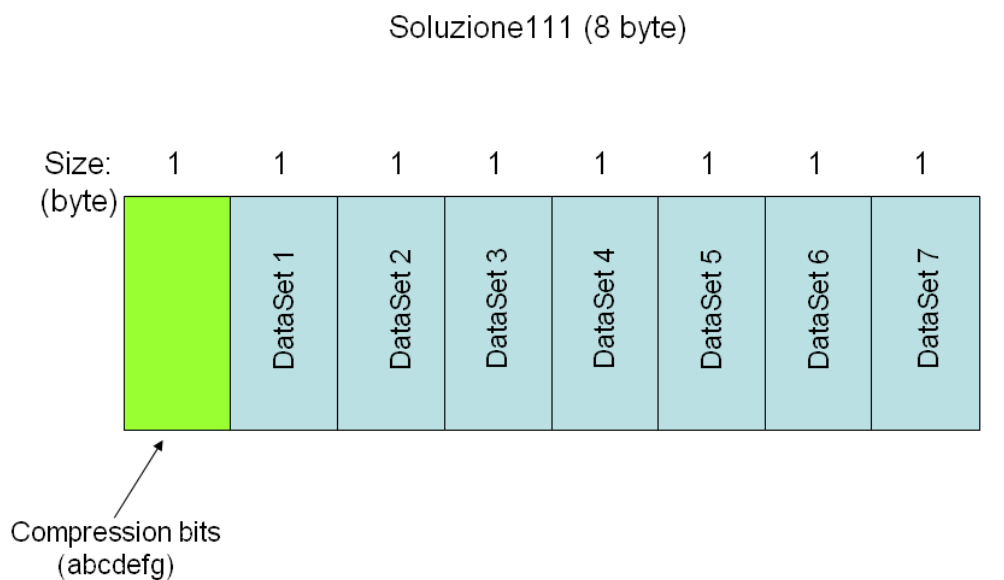
00: non usato

01: differenza in byte

10: differenza a 12 bit

11: differenza a 24 bit

L'ultima soluzione, che chiameremo **soluzione111**, da me proposta prevede che il data bundle sia così composto:



Graziano Larocca 2006

Figura 9.6: Lo schema del protocollo soluzione111

In questo caso il data bundle è di 8 byte e può avere una tra le tre seguenti soluzioni:

- 7 campioni da 1 byte
- 4 campioni da 14 bit
- 2 campioni da 24bit

I bit di compressione indicano quanto segue:

- il bit zero non è usato
- i successivi 7 bit indicano se si tratta di un segmento a 8 bit, 14 bit o 24 bit. Se il bit del dataset corrispondente è a zero si tratta di un nuovo

segmento. Se il bit è posto ad uno, il dataset deve essere letto unito al dataset precedente.

Esaminando il significato di **campionamento**, possiamo affermare che un campionamento a 100Hz equivale a dire che *sono necessari 100 campioni per formare un secondo di dati sismici*.

Da questa realtà possiamo dedurre un metodo matematico per verificare l'efficienza dei precedenti protocolli.

Supponendo di avere una configurazione con 19 bundles, si ottengono:

$$19 \times 4 \text{ (dataset per bundle)} = 76 \text{ dataset}$$

Di seguito i casi esaminati in situazione di quiete assoluta, media sismicità e *noise* massimo (non viene esaminata la soluzione111 poiché il suo funzionamento è anomalo con soli 19 bundles).

quiete assoluta: Si suppone di avere tutte le differenze dal campione precedente a 8 bit.

1. protocollo Nanometrics:

4 campioni per dataset

304 campioni per pacchetto

0.33 pacchetti per trasferire un secondo di dati

2. soluzione321:

3 campioni per dataset

228 campioni per pacchetto

0.43 pacchetti per trasferire un secondo di dati

3. soluzione431:

4 campioni per dataset

304 campioni per pacchetto

0.33 pacchetti per trasferire un secondo di dati

media sismicità (RMS al di sotto di 20000 counts): Si suppone che le differenze siano rappresentate sempre con 12 bit.

1. protocollo Nanometrics:

2 campioni per dataset

152 campioni per pacchetto

0.66 pacchetti per trasferire un secondo di dati

2. soluzione321:

2 campioni per dataset

152 campioni per pacchetto

0.66 pacchetti per trasferire un secondo di dati

3. soluzione431:

3 campioni per dataset

228 campioni per pacchetto

0.43 pacchetti per trasferire un secondo di dati

noise massimo: Si suppone che le differenze siano tutte rappresentabili con un numero binario a 32 bit.

In tutte e tre le soluzioni otteniamo:

1 campione per dataset

76 campioni per pacchetto

1,36 pacchetti per trasferire 1 secondo di dati.

9.6 Calcolo della differenza media tra campioni

Esaminando una onda sinusoidale senza offset, i punti che la compongono formano una distribuzione gaussiana avente media pari a 0 e σ uguale all'ampiezza della sinusoide.

Le differenze medie tra un campione ed il suo precedente dipendono in modo inversamente proporzionale al campionamento scelto.

Infatti una sinusoide a frequenza f avrà un periodo uguale a $1/f$, inoltre il fronte di salita o discesa della sinusoide sarà pari a $1/(4f)$ (ovvero 1/4 di periodo).

Ne consegue che il numero di campioni per fronte (NCF) sarà uguale a:

$$\text{NCF} = 1/(4f) \times f_c \quad (9.1)$$

dove f_c è la frequenza di campionamento.

Quindi la differenza media sarà:

$$A_{\text{RMS}} / \text{NCF} \quad (9.2)$$

Dove A_{RMS} è l'ampiezza RMS del segnale.

Un segnale sismico reale può essere pensato come una modulazione di tante armoniche a diversa frequenza e diversa ampiezza.

Considerando le formule (9.1) e (9.2), per il calcolo della banda possiamo approssimare il segnale sismico ad una onda sinusoidale che ha come ampiezza, di ogni semiperiodo, il livello RMS del segnale reale e come frequenza quella predominante.

ESEMPIO:

Consideriamo un segnale sismico con frequenza predominante a 5Hz , frequenza di campionamento pari a 100Hz e ampiezza RMS uguale a 1400 counts. Applichiamo le formule e otteniamo:

$$1/f = 1/5 = 0.2s \rightarrow 1/(4f) = 1/20 = 0.05s$$

$$NCF = 1/(4f) \times f_c = 0.05 \times 100 = 5 \text{ campioni}$$

$$\text{differenza media} = A_{RMS} / NCF = 1400 / 5 = 280 \text{ counts}$$

Di seguito, con l'utilizzo di un foglio di calcolo, alcune differenze medie calcolate a partire da segnali sismici a diverse frequenze e a diversi valori di RMS-amplitude:

RMS	frequenza	NCF	diff. Media
8000	1	25	320
8000	2	12,5	640
8000	3	8,333333333	960
8000	4	6,25	1280
8000	5	5	1600
8000	6	4,166666667	1920
8000	7	3,571428571	2240
8000	8	3,125	2560
8000	9	2,777777778	2880
8000	10	2,5	3200

Tabella 9.7: differenze medie ottenute variando la frequenza di ingresso

La prima colonna riporta il valore di ampiezza RMS fissato a 8000 counts un valore registrato a ECPN in assenza di attività eruttiva; la seconda colonna riporta le frequenze test; la terza colonna indica l’NCF o il numero di campioni per fronte di salita o discesa; l’ultima colonna riporta invece i valori di differenza media previsti. Si può notare che con valori RMS di 8000 counts già a frequenze di 1 Hz non basta 1 byte (max 256 valori differenza).

RMS	frequenza	NCF	diff. Media
500	5	5	100
1000	5	5	200
2000	5	5	400
4000	5	5	800
8000	5	5	1600
16000	5	5	3200
32000	5	5	6400
64000	5	5	12800
128000	5	5	25600
72000	5	5	14400

Tabella 9.8: differenze medie ottenute variando l’ampiezza RMS

In questo test vengono stimate le differenze medie tra campione e il suo precedente variando l’ampiezza RMS del ipotetico segnale d’ingresso. La riga evidenziata rappresenta il caso dell’ampiezza RMS registrata a ETF nell’eruzione del 2001.

Capitolo 10.

Confronto tra i protocolli di trasmissione: statistiche di efficienza.

10.1 Test in laboratorio su consumo di banda

Grazie all'utilizzo della strumentazione e il laboratorio di Sismologia ho eseguito dei test per misurare il consumo di banda di una stazione tipo.

La stazione tipo è stata simulata nel seguente modo:

- ✓ un Nanometrics[®] Cygnus (s/n 452), per l'impacchettamento dei dati e la successiva trasmissione via ethernet;
- ✓ un generatore di funzioni HP[®] modello 33120A, per creare dei segnali noti da digitalizzare e acquisire;
- ✓ un oscilloscopio Tektronix[®] modello TDS220, per controllare i segnali d'onda creati dal generatore di funzioni;
- ✓ un digitalizzatore Nanometrics[®] Trident (s/n 613), in cui venivano immessi i segnali noti;
- ✓ un notebook collegato con un cavetto cross al Cygnus, per l'acquisizione dei dati.



Figura 10.1: il generatore di funzioni utilizzato

Per poter immettere i segnali noti negli ingressi del digitalizzatore, ho costruito un cavetto opportuno avente da un lato tre ingressi separati utilizzando tre connettori BNC (per i tre canali a disposizione) e dall'altro lato un connettore militare a 26 pin adattato per il Trident.



Figura 10.2: il cavetto appositamente creato per il test

Per il calcolo della banda si sono usate le formule illustrate precedentemente². Per ottenere le informazioni necessarie per il conteggio dei pacchetti trasmessi, si è utilizzato il tool software *extractp* facente parte del pacchetto **Playback utilities** della Nanometrics®.

L'utility permette di ottenere una statistica sulla quantità di pacchetti ricevuti in un intervallo di tempo desiderato, oltre ad informazioni riguardanti la grandezza di un singolo pacchetto, il numero di bundles per pacchetto, l'eventuale numero di pacchetti persi (figura 1).

```
EXTRACTP V4.18
(C) Copyright 1992-2001 Nanometrics Inc.
165145(extract.:0638)I0000 Summary mode
165145(extract.:0640)I0000 Input file spec = rlab.bhn
165145(extract.:0641)I0000 Start time = 2006-05-12-17-43-00
165145(extract.:0642)I0000 Start time (longsec) = 1147455780
165145(extract.:0643)I0000 Duration (longsec) = 300
file: C:\ringbuff\RLAB.BHN

C:\ringbuff\RLAB.BHN
Oldest packet index = 0
Newest packet index = 20008
Maximum # of packets = 290691
Packet size in Bytes = 340
Bundles Per Packet = 19
Data 2006-05-12_17:42:59.8800 0d 0h 5m 2.4000s @ 100 sps. 1231: 1231-1440
Total packets: 210 (0 empty)
Data sections: 1 ( 0d 0h 5m 2.4000s)
Number of gaps: 0 ( 0d 0h 0m 0.0000s)
```

Figura 10.3: Le statistiche ottenute con *extractp* durante i vari tests

Oltre al pacchetto Playback utilities, nel notebook sono stati installati diversi software Nanometrics® per l'acquisizione e la visualizzazione dei dati come NaqsServer, DataServer e Waveform Viewer.

Di seguito nel dettaglio i test e i relativi risultati ottenuti.

² Vedi Capitolo 4.

TEST 1. (Test in frequenza) – 12 Maggio 2006

Configurazione adottata:

- ✓ input range: **40 Vpp**
- ✓ software gain: **1**
- ✓ sensitivity: **0,4 cnt/μV** (in conseguenza dei due precedenti parametri adottati)

I risultati ottenuti sono presentati qui di seguito:

ora inizio	ampiezza (Vpp)	Frequenza (Hz)	% Retx	ora fine	consumo del canale (BHN)
19.23	500mV (±100,000 counts)	0,3	10,8 (599,720s gap)	19.28	1863 bps
19.30	500mV (±100,000 counts)	0,5	10,8 (599,720s gap)	19.34	1889 bps
19.36	500mV (±100,000 counts)	1	10,8 (599,720s gap)	19.41	1889 bps
19.43	500mV (±100,000 counts)	2	10,8 (599,720s gap)	19.48	1889 bps
19.50	500mV (±100,000 counts)	4	10,8 (599,720s gap)	19.55	1889 bps
19.57	500mV (±100,000 counts)	8	10,8 (599,720s gap)	20.02	3173 bps
20.04	500mV (±100,000 counts)	16	0,0 (0s gap)	20.09	3778 bps
20.11	500mV (±100,000 counts)	32	0,0 (0s gap)	20.16	3778 bps
20.18	500mV (±40,830 counts)	45	0,0 (0s gap)	20.23	3400 bps
20.25	500mV (±100,000 counts)	0,3	0,0 (0s gap)	20.30	1863 bps

NB: I tempi sono da considerarsi in ora locale

Il test è stato eseguito per riscontrare gli eventuali legami tra frequenza del segnale di origine e consumo di banda.

Ho tenuto costante il valore dell'ampiezza del segnale in input a 500mV, mentre variavo di volta in volta il valore della frequenza. Il range d'ampiezza che ho testato era compreso tra 0,3Hz e 45Hz.

Si è osservato che il consumo di banda è costante per un range di frequenza che va dallo 0,3 ÷ 4 Hz. Da 4 Hz si ha un incremento esponenziale fino alla frequenza di 45 Hz. Quest'ultima frequenza

analizzata si trova vicino alla frequenza di taglio superiore del sensore Nanometrics® Trillium.

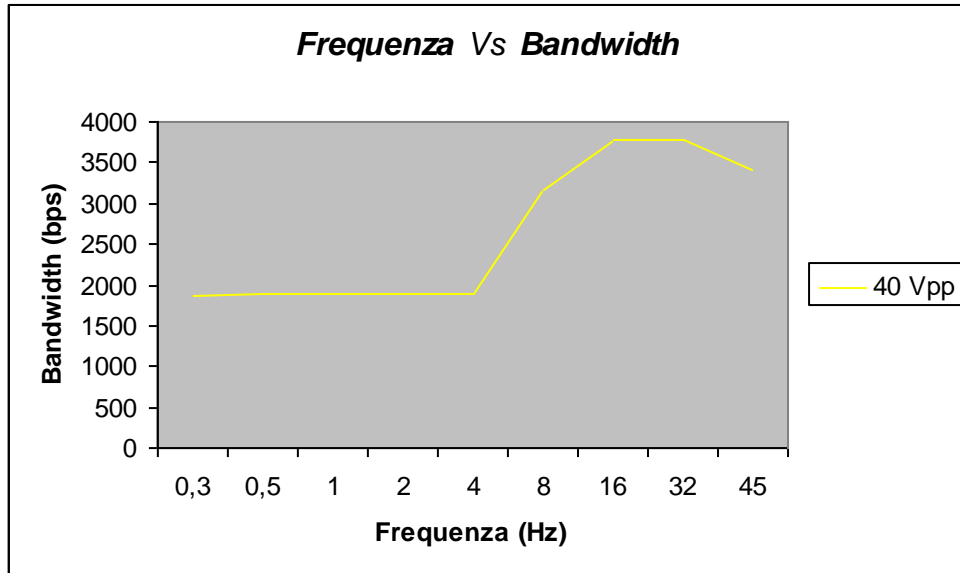


Figura 10.4: I risultati del TEST1 graficati

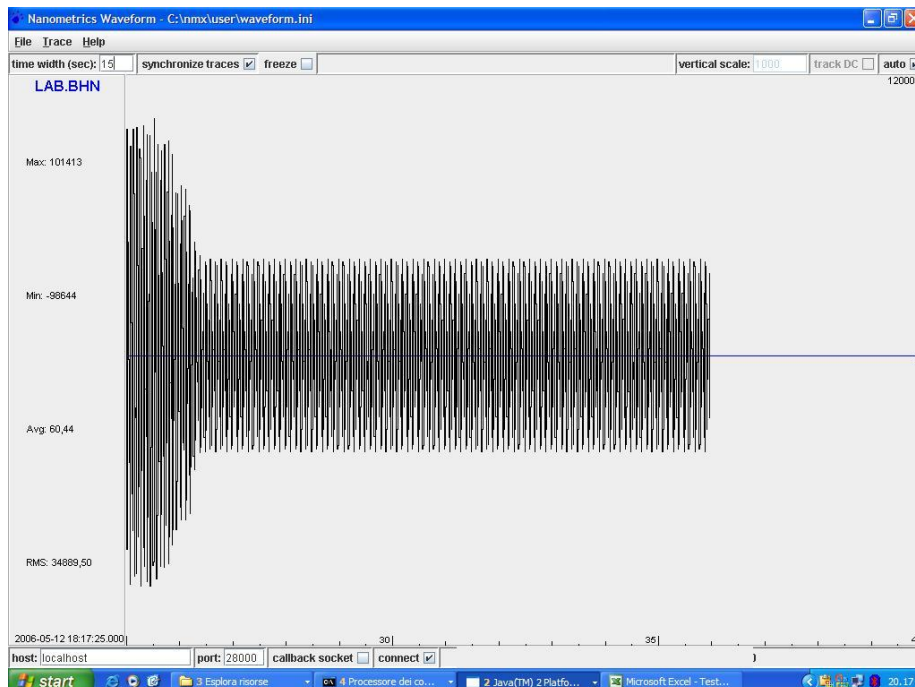


Figura 10.5: il passaggio ai 45Hz del segnale sorgente causa un attenuazione del segnale stesso.

TEST 2. (Test in ampiezza) – 15 Maggio 2006

Configurazione adottata:

- ✓ input range: **40 Vpp**
- ✓ software gain: **1**
- ✓ sensitivity: **0,4 cnt/ μ V** (in conseguenza dei due precedenti parametri adottati).

È stato eseguito per riscontrare gli eventuali legami tra ampiezza del segnale di origine e consumo di banda.

In questo secondo test, ho tenuto costante il valore della frequenza del segnale in input a 1 Hz, mentre variavo di volta in volta il valore dell'ampiezza. Il range d'ampiezza che ho testato era da 100 mV a 20 Vpp che sono i valori minimo e massimo ottenibili con il generatore di funzioni utilizzato.

I risultati ottenuti sono presentati qui di seguito:

ora inizio	ampiezza (Vpp)*	Frequenza (Hz)	% Retx	ora fine	consumo del canale (BHN)
19.23	500mV ($\pm 99,600$ counts)	1	1,3 (0s gap)	19.28	1889 bps
19.30	300mV ($\pm 59,760$ counts)	1	0,0 (0s gap)	19.35	1889 bps
19.37	100mV ($\pm 19,900$ counts)	1	0,0 (0s gap)	19.42	1886 bps
19.46	700 mV ($\pm 139,300$ counts)	1	0,0 (0s gap)	19.51	1889 bps
19.53	1 ($\pm 199,250$ counts)	1	0,0 (0s gap)	19.58	1889 bps
20.00	4 ($\pm 796,680$ counts)	1	0,0 (0s gap)	20.05	2946 bps
20.07	8 ($\pm 1,593,200$ counts)	1	0,0 (0s gap)	20.12	3400 bps
20.14	16 ($\pm 3,185,180$ counts)	1	0,0 (0s gap)	20.19	3627 bps
20.21	20 ($\pm 3,983,200$ counts)	1	0,0 (0s gap)	20.26	3627 bps

Come si può osservare dalla tabella, in condizioni di “quiete” il consumo di banda si attesta intorno ai 1889bps per la singola componente trasmessa. L'ampiezza non influisce sul consumo fino a circa 1 Vpp. Dai 4 Vpp in su si ha un incremento esponenziale fino ad arrivare a 3627bps con 20 Vpp di segnale input.

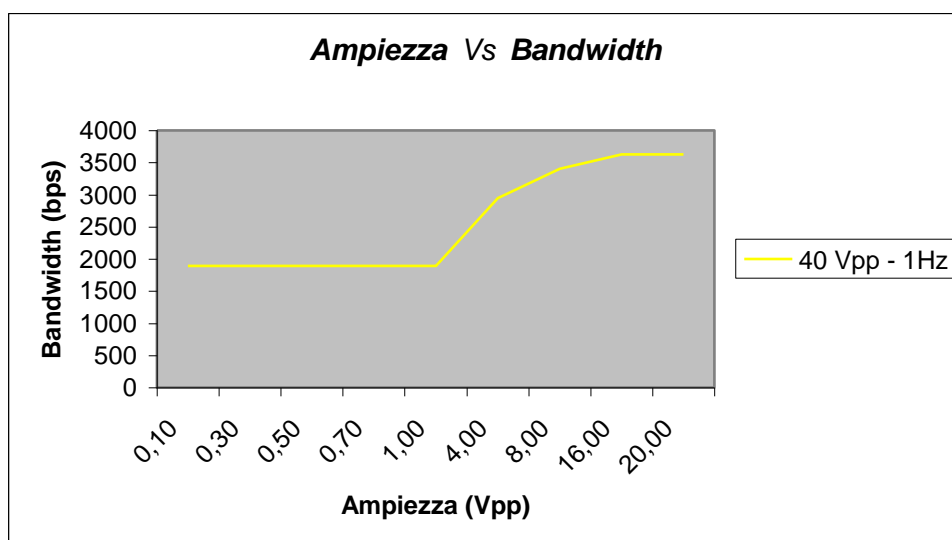


Figura 10.6: il grafico che relaziona ampiezza del segnale sorgente e consumo di banda.

TEST 3. (Test in frequenza) – 15 Maggio 2006

Configurazione adottata:

- ✓ input range: **16 Vpp**
- ✓ software gain: **1**
- ✓ sensitivity: **1 cnt/ μ V** (in conseguenza dei due precedenti parametri adottati).

È stato eseguito in maniera del tutto simile al primo test effettuato, l'unico parametro modificato è l'input range che qui è stato posto a 16 Vpp piuttosto che al valore di default di 40Vpp. È stato eseguito per poter analizzare eventuali legami tra consumo di banda e input range.

I risultati ottenuti sono presentati qui di seguito:

ora inizio	ampiezza (Vpp)	Frequenza (Hz)	% Retx	ora fine	consumo del canale (BHN)
18.56	500mV ($\pm 249,387$ counts)	0,3	0,0 (0s gap)	19.01	1889 bps
19.03	500mV ($\pm 249,387$ counts)	0,5	0,0 (0s gap)	19.08	1889 bps
19.10	500mV ($\pm 249,387$ counts)	1	0,0 (0s gap)	19.15	1889 bps
19.17	500mV ($\pm 249,387$ counts)	2	0,0 (0s gap)	19.22	1889 bps
19.24	500mV ($\pm 249,387$ counts)	4	0,0 (0s gap)	19.29	3173 bps
19.31	500mV ($\pm 249,387$ counts)	8	0,0 (0s gap)	19.36	3671 bps
19.38	500mV ($\pm 248,000$ counts)	16	0,0 (0s gap)	19.43	3778 bps
19.45	500mV ($\pm 248,000$ counts)	32	0,0 (0s gap)	19.50	3778 bps
19.52	500mV ($\pm 102,750$ counts)	45	0,0 (0s gap)	19.57	3763 bps

Come era prevedibile, i risultati sono del tutto simili, se non identici, al primo test effettuato.

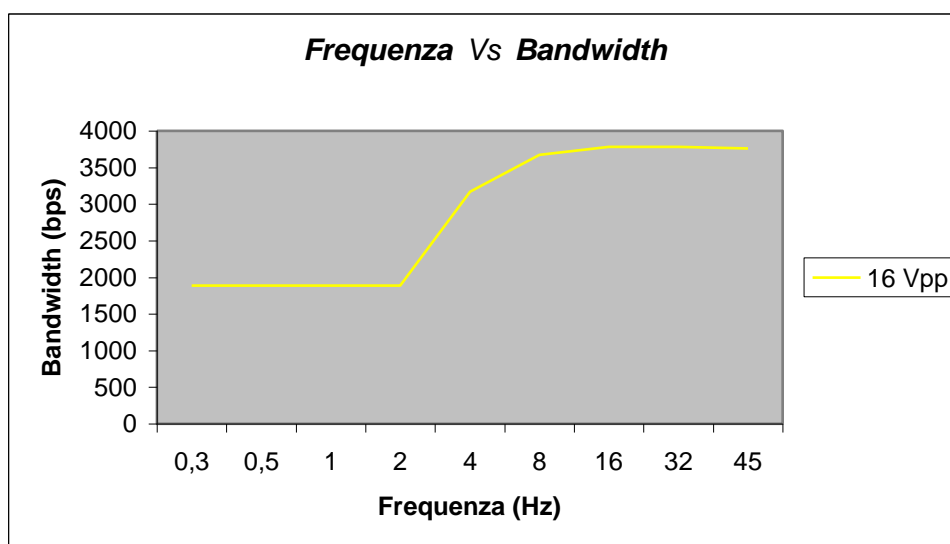


Figura 10.7: il grafico che relaziona frequenza del segnale sorgente e consumo di banda.

TEST 4. (Test in ampiezza) – 15 Maggio 2006

Configurazione adottata:

- ✓ input range: **16 Vpp**
- ✓ software gain: **1**
- ✓ sensitivity: **1 cnt/ μ V** (in conseguenza dei due precedenti parametri adottati).

È stato eseguito in maniera del tutto simile al secondo test eseguito, l'unico parametro modificato è l'input range che qui è stato posto a 16 Vpp piuttosto che al valore di default di 40Vpp. È stato eseguito per poter analizzare eventuali legami tra consumo di banda e input range.

I risultati ottenuti sono presentati qui di seguito:

ora inizio	ampiezza (Vpp)	Frequenza (Hz)	% Retx	ora fine	consumo del canale (BHN)
19.59	100mV ($\pm 49,950$ counts)	1	0,0 (0s gap)	20.04	1889 bps
20.06	300mV ($\pm 149,480$ counts)	1	0,0 (0s gap)	20.11	1889 bps
20.13	500mV ($\pm 249,280$ counts)	1	0,0 (0s gap)	20.18	1889 bps
20.20	700 mV ($\pm 348,880$ counts)	1	0,0 (0s gap)	20.25	1889 bps
20.27	1 ($\pm 498,490$ counts)	1	0,0 (0s gap)	20.32	1889 bps
20.34	4 ($\pm 1,991,940$ counts)	1	0,0 (0s gap)	20.39	3476 bps
20.41	8 ($\pm 3,984,687$ counts)	1	0,0 (0s gap)	20.46	3627 bps
20.49	16 ($\pm 7,962,125$ counts)	1	0,0 (0s gap)	20.54	3710 bps
20.56	20 ($\pm 11,450,000$ counts)**	1	0,0 (7,2s gap)	21.01	3777 bps

Come in precedenza i risultati ottenuti sono del tutto simili, se non identici, al secondo test effettuato.

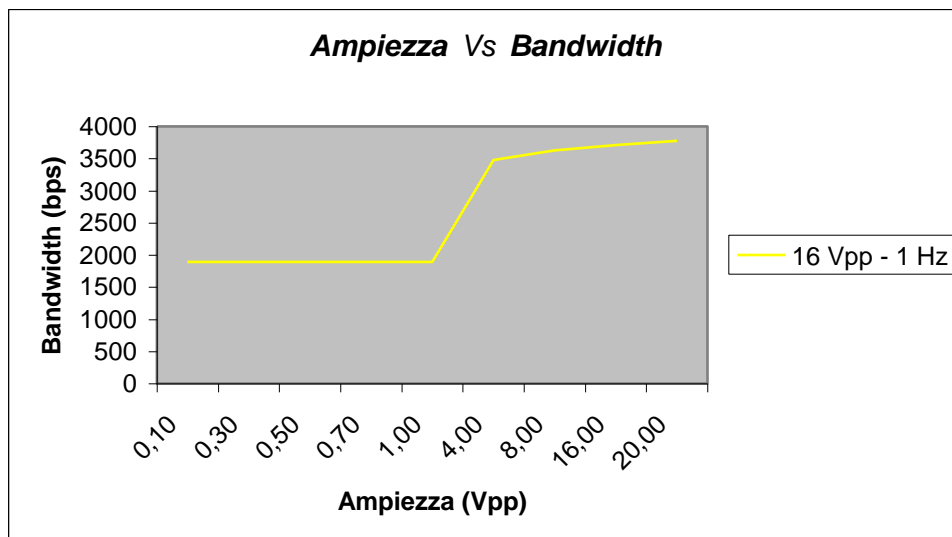


Figura 10.8: il grafico che relaziona ampiezza del segnale sorgente e consumo di banda.

Da notare il test eseguito con 20Vpp: l'input range è stato settato a 16Vpp, quindi si è andato, di proposito, fuori scala massima. In tale

situazione si nota una sensibile degradazione del segnale digitale. Di seguito una visualizzazione del segnale digitalizzato.

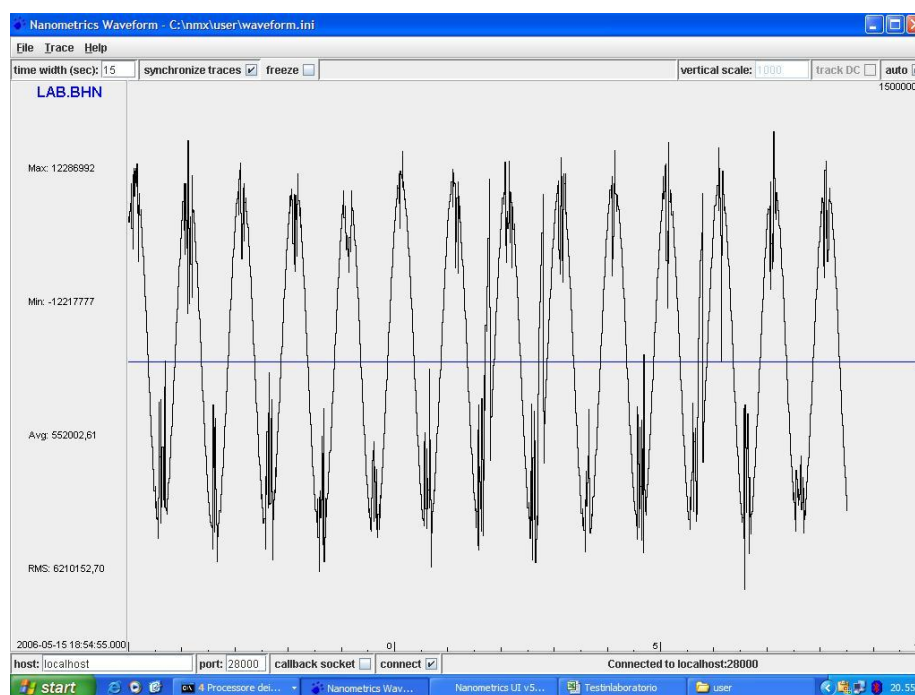


Figura 10.9: la degradazione che si è ottenuta inserendo un segnale d'ingresso con ampiezza maggiore dell'input range settato (16Vpp).

TEST 5. (Test in gain) – 18 Maggio 2006

Configurazione adottata:

- ✓ input range: **16Vpp**
- ✓ software gain: **variabile**
- ✓ sensitivity: **variabile** (in conseguenza del precedente parametro).
- ✓ frequenza: **2Hz**
- ✓ ampiezza: **100mVpp**

Il test è stato eseguito per osservare i legami tra il parametro *software gain* del digitalizzatore e il consumo di banda. Inoltre è servito per

osservare i valori di sensitività al variare del gain. L'ampiezza del segnale adottato è di 100mV mentre la frequenza a 2Hz. Il range del gain adottato è stato da 0,001 ÷ 8.

I risultati ottenuti sono presentati qui di seguito:

ora inizio	ampiezza (Vpp)	gain	Sensitivity (Counts/ μ V)	% Retx	ora fine	consumo del canale (BHN)
17.23	100mV (+ 43 -57 counts)	0,001	0,001 (res. 13,98 b)	4,5 (0s gap)	17.28	944 bps
17.31	100mV (+ 248 - 249 counts)	0,005	0,005 (res. 16,29bits)	3,8 (0s gap)	17.36	944 bps
17.38	100mV (+ 497 -497 counts)	0,010	0,01 (res. 17,29bits)	3,1 (0s gap)	17.43	944 bps
17.45	100mV (+ 996 -994 counts)	0,020	0,02 (res. 18,29bits)	2,6 (0s gap)	17.50	944 bps
17.52	100mV (+ 2490 -2497 counts)	0,050	0,05(res. 19,61bits)	2,3 (0s gap)	17.57	1738 bps
17.59	100mV (+ 4983 -4975 counts)	0,100	0,1 (res. 20,61bits)	2,2 (368.600s gap)	18.04	1813 bps
18.06	100mV (+ 9969 -9952 counts)	0,200	0,2 (res. 21,61bits)	0,0 (0s gap)	18.11	1889 bps
18.13	100mV (+ 24927 -24874 counts)	0,500	0,5 (res. 22,93bits)	0,0 (0s gap)	18.18	1889 bps
18.21	100mV (+ 49819 -49752 counts)	1,000	1 (res. 23,93bits)	0,0 (0s gap)	18.26	1889 bps
18.28	100mV (+ 99660 -99440 counts)	2,000	2 (res. 24,93bits)	0,0 (0s gap)	18.31	1889 bps
18.33	100mV (+ 199204 -198774 counts)	4,000	4 (res. 25,93bits)	0,0 (0s gap)	18.38	1889 bps
18.40	100mV (+ 398032 -397428 counts)	8,000	8 (res. 26,93bits)	0,0 (0s gap)	18.45	3022 bps

Si può osservare dalla tabella che il consumo si è mantenuto relativamente basso (944 bps) fino a valori di gain uguali o inferiori a 0.05. Per valori più alti di gain il consumo rimane costante fino a 4 dopo di che si ha un improvviso innalzamento a 8 Vpp con 3022bps di consumo per canale.

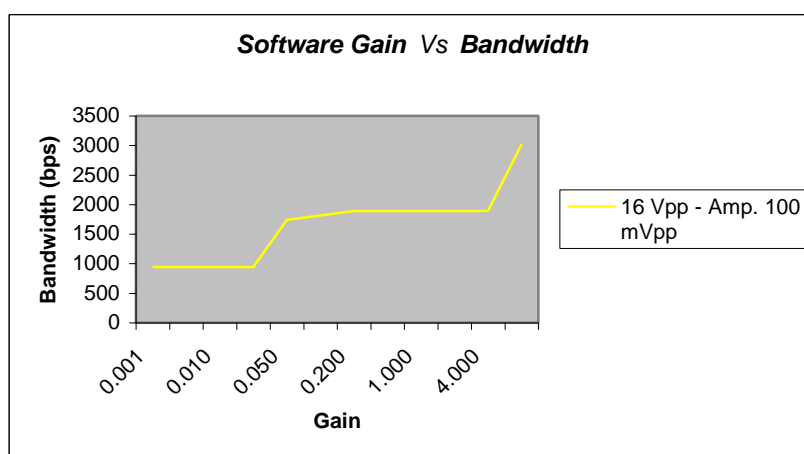


Figura 10.10: la curva a scalino che mette in risalto il legame tra il parametro software gain e il consumo di banda

I valori di sensitività sono uguali ai valori del parametro gain, cioè se si settava un valore, ad esempio, di 0.05 come gain con la configurazione adottata si aveva 0.05 cnt/ μ V come sensitività.

La risoluzione teorica in bit è passata da 13.96 a 26.93 bits, con un valore di *software gain* pari a 8.

10.2 Statistiche di efficienza per la trasmissione radio UHF.

Grazie all'utilizzo di un foglio di calcolo, sono state applicate le leggi indicate in precedenza e sono state applicate ipotizzando di utilizzare i protocolli già introdotti. La differenza tra trasmissione UHF e VSAT in queste applicazioni sta nella durata del *frame* e nella percentuale di tempo di trasmissione. In questo documento utilizzerò la parola **frame** per indicare la durata di un ciclo tdma.

Nella trasmissione radio UHF della Rete Sismica Permanente il frame ha una durata di 4 secondi e la strumentazione è settata per ottenere l'80% della durata frame per la trasmissione ed il restante 20% per richiedere eventuali ritrasmissioni.

Di conseguenza riusciamo ad avere:

- **900 frame ad ora**
- **15 frame al minuto**
- **3,2 secondi per frame per la trasmissione**
- **0,8 secondi per frame dedicati alle richieste di ritrasmissione**

Analizzando queste caratteristiche, di seguito vengono presentate delle statistiche di efficienza.

Calcolo del tempo di digitalizzazione per pacchetto.

Si analizza un segnale sismico input con una frequenza di 5Hz, mentre si ipotizzano ampiezze RMS differenti. Lo scopo del calcolo è quello di confrontare i tempi di digitalizzazione dei pacchetti nelle varie soluzioni a cui applichiamo il calcolo.

Possiamo così descrivere le tabelle seguenti:

La prima colonna ho inserito diversi valori di RMS in counts (in arancione il valore di ETF nell'eruzione 2001), nella seconda colonna la frequenza del segnale di input fissata a 5Hz e nella terza e quarta colonna il valore conseguente di NCF e differenza media. Nella quinta e nella sesta colonna riporto la quantità di campioni per singolo bundle e il conseguente numero di campioni per pacchetto. Nelle ultime due colonne viene riportato la quantità di pacchetti, necessaria al trasferimento di un secondo di dato in input, e il tempo necessario per digitalizzarlo.

- Protocollo: **Nanometrics**
- Bundles/packet: **19**

I risultati ottenuti sono presentati qui di seguito:

RMS	frequenza	NCF	diff. Media	NumCampBundle	NumCampPacc	NumPacc1sec	sec. per packet
500	5	5	100	4	304	0.328947368	3.04
1000	5	5	200	4	304	0.328947368	3.04
2000	5	5	400	2	152	0.657894737	1.52
4000	5	5	800	2	152	0.657894737	1.52
8000	5	5	1600	2	152	0.657894737	1.52
16000	5	5	3200	2	152	0.657894737	1.52
32000	5	5	6400	2	152	0.657894737	1.52
64000	5	5	12800	2	152	0.657894737	1.52
128000	5	5	25600	2	152	0.657894737	1.52
72000	5	5	14400	2	152	0.657894737	1.52

➤ Protocollo: **Soluzione321**

➤ Bundles/packet: **19**

I risultati ottenuti sono presentati qui di seguito:

RMS	frequenza	NCF	diff. Media	NumCampBundle	NumCampPacc	NumPacc1sec	sec. per packet
500	5	5	100	3	228	0.438596491	2.28
1000	5	5	200	3	228	0.438596491	2.28
2000	5	5	400	2	152	0.657894737	1.52
4000	5	5	800	2	152	0.657894737	1.52
8000	5	5	1600	2	152	0.657894737	1.52
16000	5	5	3200	2	152	0.657894737	1.52
32000	5	5	6400	1	76	1.315789474	0.76
64000	5	5	12800	1	76	1.315789474	0.76
128000	5	5	25600	1	76	1.315789474	0.76
72000	5	5	14400	1	76	1.315789474	0.76

➤ Protocollo: **Soluzione321**

➤ Bundles/packet: **27**

I risultati ottenuti sono presentati qui di seguito:

RMS	frequenza	NCF	diff. Media	NumCampBundle	NumCampPacc	NumPacc1sec	sec. per packet
500	5	5	100	3	324	0.308641975	3.24
1000	5	5	200	3	324	0.308641975	3.24
2000	5	5	400	2	216	0.462962963	2.16
4000	5	5	800	2	216	0.462962963	2.16
8000	5	5	1600	2	216	0.462962963	2.16
16000	5	5	3200	2	216	0.462962963	2.16
32000	5	5	6400	1	108	0.925925926	1.08
64000	5	5	12800	1	108	0.925925926	1.08
128000	5	5	25600	1	108	0.925925926	1.08
72000	5	5	14400	1	108	0.925925926	1.08

➤ Protocollo: **Soluzione431**

➤ Bundles/packet: **19**

I risultati ottenuti sono presentati qui di seguito:

RMS	frequenza	NCF	diff. Media	NumCampBundle	NumCampPacc	NumPacc1sec	sec. per packet
500	5	5	100	4	304	0.328947368	3.04
1000	5	5	200	4	304	0.328947368	3.04
2000	5	5	400	3	228	0.438596491	2.28
4000	5	5	800	3	228	0.438596491	2.28
8000	5	5	1600	3	228	0.438596491	2.28
16000	5	5	3200	3	228	0.438596491	2.28
32000	5	5	6400	1	76	1.315789474	0.76
64000	5	5	12800	1	76	1.315789474	0.76
128000	5	5	25600	1	76	1.315789474	0.76
72000	5	5	14400	1	76	1.315789474	0.76

➤ Protocollo: **Soluzione111**

➤ Bundles/packet: **40**

I risultati ottenuti sono presentati qui di seguito:

RMS	frequenza	NCF	diff. Media	NumCampBundle	NumCampPacc	NumPacc1sec	sec. per packet
500	5	5	100	7	280	0.357142857	2.8
1000	5	5	200	7	280	0.357142857	2.8
2000	5	5	400	4	160	0.625	1.6
4000	5	5	800	4	160	0.625	1.6
8000	5	5	1600	4	160	0.625	1.6
16000	5	5	3200	4	160	0.625	1.6
32000	5	5	6400	4	160	0.625	1.6
64000	5	5	12800	4	160	0.625	1.6
128000	5	5	25600	2	80	1.25	0.8
72000	5	5	14400	4	160	0.625	1.6

Calcolo del tempo di trasmissione per pacchetto.

Si analizza un segnale sismico input con una frequenza di 5Hz, mentre si ipotizzano ampiezze RMS differenti. Lo scopo del calcolo è quello di confrontare i tempi di trasmissione dei pacchetti nelle varie soluzioni a cui applichiamo il calcolo.

Possiamo così descrivere le tabelle seguenti:

La prima colonna riporta diversi valori di RMS in counts(in arancione il valore di ETF nell'eruzione 2001), nella seconda colonna la il numero di campioni per pacchetto in funzione dell'RMS, nella terza colonna viene calcolata la dimensione di ogni pacchetto utilizzando la seguente formula:

$$25 + (\text{dimensione bundle(bytes)} * \text{bundles/packet})*10$$

dove 25 bit sono l'insieme degli overhead. Nell'ultima colonna viene riportato il tempo di trasmissione calcolato dividendo la dimensione dei pacchetti con il baud rate utilizzato nella trasmissione.

Protocollo: **Nanometrics**

- Bundles/packet: **19**
- Baud rate: **9600bps**

Ne segue che ogni pacchetto ha una dimensione di 3480bit e quindi il tempo di trasmissione è **0,3625s**.

Protocollo: **Nanometrics**

- Bundles/packet: **19**
- Baud rate: **19200bps**

Ne segue che ogni pacchetto ha una dimensione di 3480bit e quindi il tempo di trasmissione è **0,18125s**.

Protocollo: **Soluzione321**

- Bundles/packet: **19**
- Baud rate: **9600bps**

Ne segue che ogni pacchetto ha una dimensione di 2720bit e quindi il tempo di trasmissione è **0,2833s**.

Protocollo: **Soluzione321**

- Bundles/packet: **27**
- Baud rate: **9600bps**

Ne segue che ogni pacchetto ha una dimensione di 3760bit e quindi il tempo di trasmissione è **0,3916s**.

Protocollo: **Soluzione431**

- Bundles/packet: **19**
- Baud rate: **9600bps**

Ne segue che ogni pacchetto ha una dimensione di 3480bit e quindi il tempo di trasmissione è **0,3625s**.

Protocollo: **Nanometrics**

- Bundles/packet: **15**
- Baud rate: **9600bps**

Ne segue che ogni pacchetto ha una dimensione di 2800bit e quindi il tempo di trasmissione è **0,2916s**.

Protocollo: **Soluzione111**

- Bundles/packet: **40**
- Baud rate: **9600bps**

Ne segue che ogni pacchetto ha una dimensione di 3450bit e quindi il tempo di trasmissione è **0,3594s**.

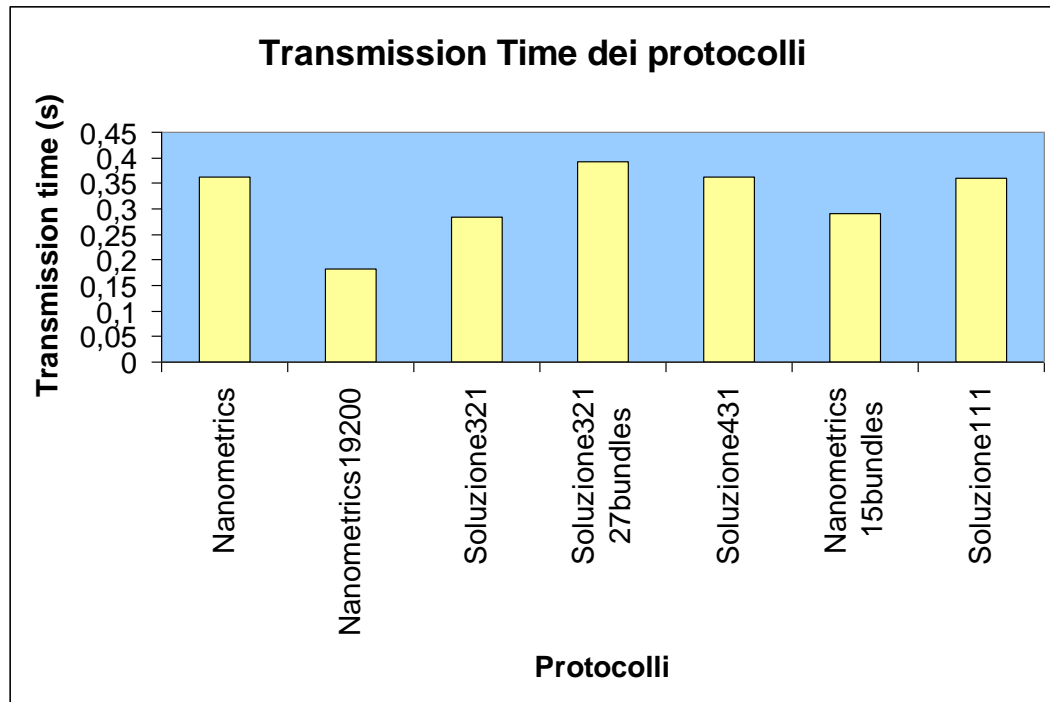


Figura 10.11: grafico che riporta il tempo di trasmissione per singolo pacchetto

Calcolo della quantità di dati per frame e previsione in funzione della percentuale di ritrasmissione.

Si analizza un segnale sismico input con una frequenza di 5Hz, mentre si ipotizzano ampiezze RMS differenti. Lo scopo del calcolo è quello di confrontare la quantità di dati trasmessa per frame e quella in funzione della percentuale di richieste di ritrasmissione presente. La prima tabella riporta nella prima colonna i diversi valori di RMS in counts (in arancione il valore di ETF nell'eruzione 2001), nella seconda colonna il numero di campioni per pacchetto in funzione dell'RMS, nella terza colonna viene calcolata la quantità massima di dati in secondo che possono essere trasmessi in un singolo frame, utilizzando la seguente formula:

$$(num_campioni_per_pacchetto \times num_pacchetti_per_frame)/100$$

Nella seconda tabella la prima colonna riporta i valori ipotetici di percentuale di ritrasmissione, la seconda i secondi di dato ritrasnesso utilizzando la seguente legge:

$$(((num_totale_di_pacc_trasm_in1minuto \times percentuale_retx)/100) \times num_campioni_pacc)/100$$

Le rimanenti colonne rappresentano le quantità massime di dato che si può trasmettere in base alla compressione raggiunta. Il calcolo si basa sulla seguente legge:

$$(quantità_dati_per_frame \times 15)/3$$

Il risultato è riferito alla singola componente (se ci sono tre componenti) e ad un minuto di trasmissione.

- Protocollo: **Nanometrics**
- Bundles/packet: **19**
- Baud rate: **9600bps**
- Transmission time: **0,3625s/packet**
- N° pacchetti per frame: **$3,2/0,3625 = 8,8276$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	304	26,835904
1000	304	26,835904
2000	152	13,417952
4000	152	13,417952
8000	152	13,417952
16000	152	13,417952
32000	152	13,417952
64000	152	13,417952
128000	152	13,417952
72000	152	13,417952

	con 304	con 304	con 152
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto
0	0	134,17952	67,08976
0,5	2,01096	133,5092	66,7546
1	4,02192	132,83888	66,41944
2	8,04384	131,49824	65,74912
4	16,08768	128,81696	64,40848
8	32,17536	123,4544	61,7272
16	64,35072	112,72928	56,36464
32	128,70144	91,27904	45,63952
64	257,40288	48,37856	24,18928
80	321,7536	26,92832	13,46416
100	402,192	0,11552	0,05776

- Protocollo: **Soluzione321**
- Bundles/packet: **19**
- Baud rate: **9600bps**
- Transmission time: **0,2833s/packet**
- N° pacchetti per frame: **$3,2/0,2833 = 11,2954$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	228	25,753512
1000	228	25,753512
2000	152	17,169008
4000	152	17,169008
8000	152	17,169008
16000	152	17,169008
32000	76	8,584504
64000	76	8,584504
128000	76	8,584504
72000	76	8,584504

	con 228	con 228	con 152	con 76
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	128,76756	85,84504	42,92252
0,5	1,931502	128,123726	85,41581733	42,70790867
1	3,863004	127,479892	84,98659467	42,49329733
2	7,726008	126,192224	84,12814933	42,06407467
4	15,452016	123,616888	82,41125867	41,20562933
8	30,904032	118,466216	78,97747733	39,48873867
16	61,808064	108,164872	72,10991467	36,05495733
32	123,616128	87,562184	58,37478933	29,18739467
64	247,232256	46,356808	30,90453867	15,45226933
80	309,04032	25,75412	17,16941333	8,584706667
100	386,3004	0,00076	0,000506667	0,000253333

- Protocollo: **Soluzione321**
- Bundles/packet: **27**
- Baud rate: **9600bps**
- Transmission time: **0,3916s/packet**
- N° pacchetti per frame: **$3,2/0,3916 = 8,1716$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	324	26,475984
1000	324	26,475984
2000	216	17,650656
4000	216	17,650656
8000	216	17,650656
16000	216	17,650656
32000	108	8,825328
64000	108	8,825328
128000	108	8,825328
72000	108	8,825328

	con 324	con 324	con 216	con 108
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	132,37992	88,25328	44,12664
0,5	2,744766	131,464998	87,643332	43,821666
1	5,489532	130,550076	87,033384	43,516692
2	10,979064	128,720232	85,813488	42,906744
4	21,958128	125,060544	83,373696	41,686848
8	43,916256	117,741168	78,494112	39,247056
16	87,832512	103,102416	68,734944	34,367472
32	175,665024	73,824912	49,216608	24,608304
64	351,330048	15,269904	10,179936	5,089968
80	439,16256	-14,0076	-9,3384	-4,6692
100	548,9532	-50,60448	-33,73632	-16,86816

- Protocollo: **Soluzione431**
- Bundles/packet: **19**
- Baud rate: **9600bps**
- Transmission time: **0,3625s/packet**
- N° pacchetti per frame: **$3,2/0,3625 = 8,8275$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	304	26,8356
1000	304	26,8356
2000	228	20,1267
4000	228	20,1267
8000	228	20,1267
16000	228	20,1267
32000	76	6,7089
64000	76	6,7089
128000	76	6,7089
72000	76	6,7089

	con 304	con 304	con 228	con 76
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	134,178	100,6335	33,5445
0,5	2,575336	133,3195547	99,989666	33,32988667
1	5,150672	132,4611093	99,345832	33,11527733
2	10,301344	130,7442187	98,058164	32,68605467
4	20,602688	127,3104373	95,482828	31,82760933
8	41,205376	120,4428747	90,332156	30,11071867
16	82,410752	106,7077493	80,030812	26,67693733
32	164,821504	79,23749867	59,428124	19,80937467
64	329,643008	24,29699733	18,222748	6,074249333
80	412,05376	-3,173253333	-2,37994	-0,793313333
100	515,0672	-37,51106667	-28,1333	-9,37766667

- Protocollo: **Soluzione111**
- Bundles/packet: **19**
- Baud rate: **9600bps**
- Transmission time: **0,3625s/packet**
- N° pacchetti per frame: $3,2/0,3625 = 8,8275$ **pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	280	24,93204
1000	280	24,93204
2000	160	14,24688
4000	160	14,24688
8000	160	14,24688
16000	160	14,24688
32000	160	14,24688
64000	160	14,24688
128000	80	7,12344
72000	160	14,24688

	con 280	con 280	con 160	con 80
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	124,6602	71,2344	62,3301
0,5	1,86984	124,03692	70,87824	62,01846
1	3,73968	123,41364	70,52208	61,70682
2	7,47936	122,16708	69,80976	61,08354
4	14,95872	119,67396	68,38512	59,83698
8	29,91744	114,68772	65,53584	57,34386
16	59,83488	104,71524	59,83728	52,35762
32	119,66976	84,77028	48,44016	42,38514
64	239,33952	44,88036	25,64592	22,44018
80	299,1744	24,9354	14,2488	12,4677
100	373,968	0,0042	0,0024	0,0021

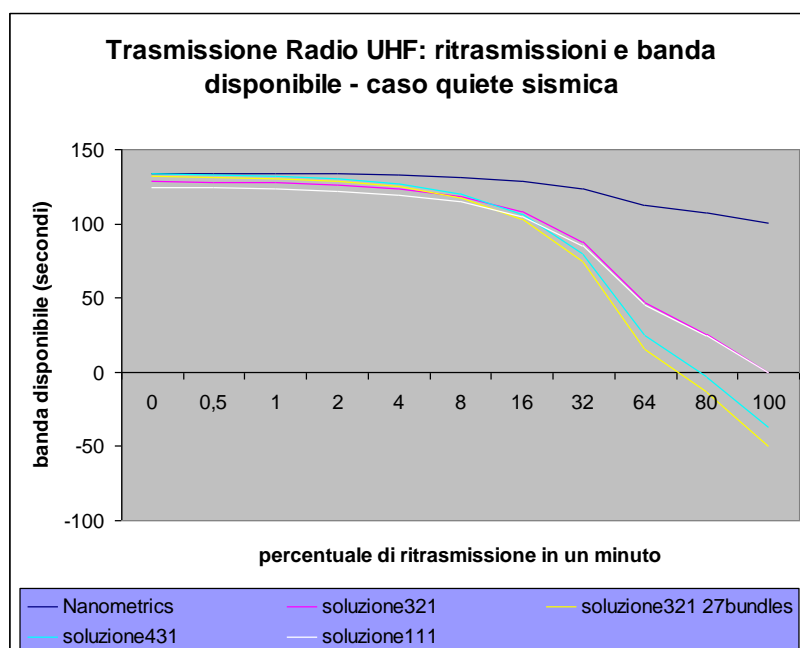


Figura 10.12: grafico che riporta la banda disponibile in base alla percentuale di ritrasmissione presente in un minuto, nel caso di sismicità bassa

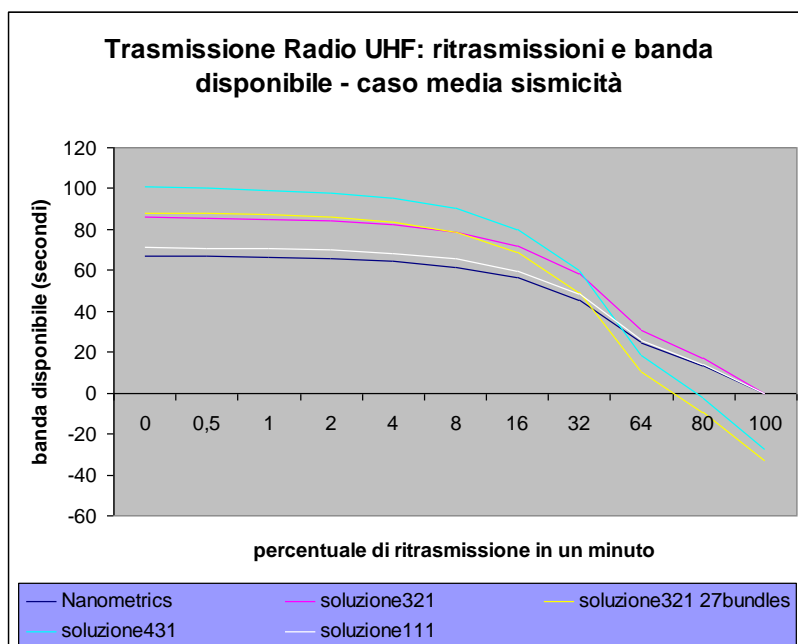


Figura 10.13: grafico che riporta la banda disponibile in base alla percentuale di ritrasmissione presente in un minuto, nel caso di livello di sismicità medio

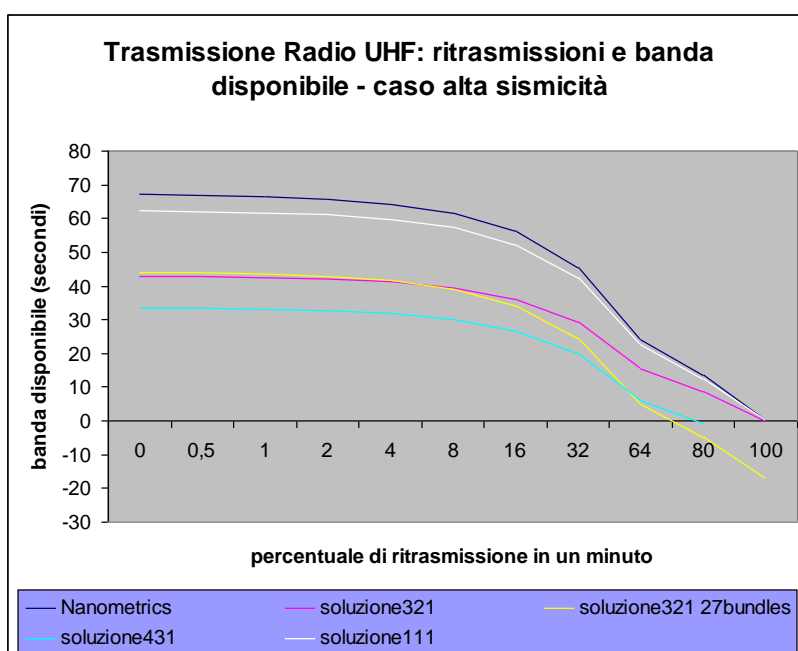


Figura 10.14: grafico che riporta la banda disponibile in base alla percentuale di ritrasmissione presente in un minuto, nel caso di livello di sismicità alto

10.3 Statistiche di efficienza per la Trasmissione radio satellitare.

Nella trasmissione radio satellitare della Rete Sismica Permanente il frame ha una durata di 10 secondi e la strumentazione è settata per ottenere una percentuale variabile della durata frame per la trasmissione. Nei calcoli che presento ho ipotizzato una percentuale dello 0,8% , i risultati sono:

- **360 frame ad ora**
- **6 frame al minuto**
- **0,8 secondi per frame per la trasmissione**

Analizzando queste caratteristiche, di seguito vengono presentate delle statistiche di efficienza.

Si attenziona che i calcoli per i tempi di digitalizzazione sono quelli riportati per la trasmissione UHF.

Calcolo del tempo di trasmissione per pacchetto.

Protocollo: **Nanometrics**

- Bundles/packet: **19**
- Baud rate: **114688bps**

Ne segue che ogni pacchetto ha una dimensione di 3480bit e quindi il tempo di trasmissione è **0,03034s**.

Protocollo: **Soluzione321**

- Bundles/packet: **19**
- Baud rate: **114688bps**

Ne segue che ogni pacchetto ha una dimensione di 2720bit e quindi il tempo di trasmissione è **0,02372s**.

Protocollo: **Soluzione321**

- Bundles/packet: **27**
- Baud rate: **114688bps**

Ne segue che ogni pacchetto ha una dimensione di 3760bit e quindi il tempo di trasmissione è **0,03278s**.

Protocollo: **Soluzione431**

- Bundles/packet: **19**
- Baud rate: **114688bps**

Ne segue che ogni pacchetto ha una dimensione di 3480bit e quindi il tempo di trasmissione è **0,03034s**.

Protocollo: **Nanometrics**

- Bundles/packet: **15**
- Baud rate: **114688bps**

Ne segue che ogni pacchetto ha una dimensione di 2800bit e quindi il tempo di trasmissione è **0,02441s**.

Protocollo: **Soluzione111**

- Bundles/packet: **40**
- Baud rate: **114688bps**

Ne segue che ogni pacchetto ha una dimensione di 3450bit e quindi il tempo di trasmissione è **0,03008s**.

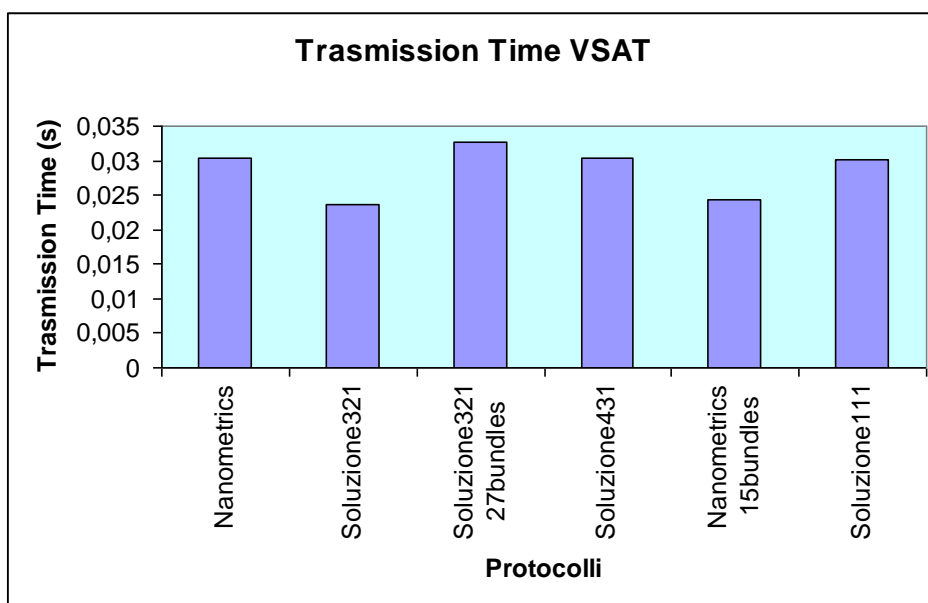


Figura 10.15: grafico che riporta il tempo di trasmissione del singolo pacchetto per ogni tipo di protocollo esaminato.

Calcolo della quantità di dati per frame e previsione in funzione della percentuale di ritrasmissione.

- Protocollo: **Nanometrics**
- Bundles/packet: **19**
- Baud rate: **114688bps**
- Transmission time: **0,03034s/packet**
- N° pacchetti per frame: **$0,8/0,03034 = 26,37$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	304	80,1648
1000	304	80,1648
2000	152	40,0824
4000	152	40,0824
8000	152	40,0824
16000	152	40,0824
32000	152	40,0824
64000	152	40,0824
128000	152	40,0824
72000	152	40,0824

	con 304	con 304	con 152
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto
0	0	160,3296	80,1648
0,5	2,404944	159,527952	79,763976
1	4,809888	158,726304	79,363152
2	9,619776	157,123008	78,561504
4	19,239552	153,916416	76,958208
8	38,479104	147,503232	73,751616
16	76,958208	134,676864	67,338432
32	153,916416	109,024128	54,512064
64	307,832832	57,718656	28,859328
80	384,79104	32,06592	16,03296
100	480,9888	0	0

- Protocollo: **Soluzione321**
- Bundles/packet: **19**
- Baud rate: **114688bps**
- Transmission time: **0,02372s/packet**
- N° pacchetti per frame: **$0,8/0,02372 = 33,73$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	228	76,9044
1000	228	76,9044
2000	152	51,2696
4000	152	51,2696
8000	152	51,2696
16000	152	51,2696
32000	76	25,6348
64000	76	25,6348
128000	76	25,6348
72000	76	25,6348

	con 228	con 228	con 152	con 76
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	153,8088	102,5392	51,2696
0,5	2,306904	153,039832	102,0265547	51,01327733
1	4,613808	152,270864	101,5139093	50,75695467
2	9,227616	150,732928	100,4886187	50,24430933
4	18,455232	147,657056	98,43803733	49,21901867
8	36,910464	141,505312	94,33687467	47,16843733
16	73,820928	129,201824	86,13454933	43,06727467
32	147,641856	104,594848	69,72989867	34,86494933
64	295,283712	55,380896	36,92059733	18,46029867
80	369,10464	30,77392	20,51594667	10,25797333
100	461,3808	0,0152	0,010133333	0,005066667

- Protocollo: **Soluzione321**
- Bundles/packet: **27**
- Baud rate: **114688bps**
- Transmission time: **0,03279s/packet**
- N° pacchetti per frame: **$0,8/0,03279 = 24,397$ pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	324	79,04628
1000	324	79,04628
2000	216	52,69752
4000	216	52,69752
8000	216	52,69752
16000	216	52,69752
32000	108	26,34876
64000	108	26,34876
128000	108	26,34876
72000	108	26,34876

	con 324	con 324	con 216	con 108
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	158,09256	105,39504	52,69752
0,5	2,371356	157,302108	104,868072	52,434036
1	4,742712	156,511656	104,341104	52,170552
2	9,485424	154,930752	103,287168	51,643584
4	18,970848	151,768944	101,179296	50,589648
8	37,941696	145,445328	96,963552	48,481776
16	75,883392	132,798096	88,532064	44,266032
32	151,766784	107,503632	71,669088	35,834544
64	303,533568	56,914704	37,943136	18,971568
80	379,41696	31,62024	21,08016	10,54008
100	474,2712	0,00216	0,00144	0,00072

- Protocollo: **Soluzione431**
- Bundles/packet: **19**
- Baud rate: **114688bps**
- Transmission time: **0,03034s/packet**
- N° pacchetti per frame **0,8/0,03034 = 26,37 pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	304	80,1648
1000	304	80,1648
2000	228	60,1236
4000	228	60,1236
8000	228	60,1236
16000	228	60,1236
32000	76	20,0412
64000	76	20,0412
128000	76	20,0412
72000	76	20,0412

	con 304	con 304	con 228	con 76
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	160,3296	120,2472	40,0824
0,5	2,404944	159,527952	119,645964	39,881988
1	4,809888	158,726304	119,044728	39,681576
2	9,619776	157,123008	117,842256	39,280752
4	19,239552	153,916416	115,437312	38,479104
8	38,479104	147,503232	110,627424	36,875808
16	76,958208	134,676864	101,007648	33,669216
32	153,916416	109,024128	81,768096	27,256032
64	307,832832	57,718656	43,288992	14,429664
80	384,79104	32,06592	24,04944	8,01648
100	480,9888	0	0	0

- Protocollo: **Soluzione111**
- Bundles/packet: **40**
- Baud rate: **114688bps**
- Transmission time: **0,0301s/packet**
- N° pacchetti per frame: **0,8/0,0301 = 26,57 pacchetti**

I risultati ottenuti sono presentati qui di seguito:

RMS	NumCampPacc	DatoSecondo/frame
500	280	74,396
1000	280	74,396
2000	160	42,512
4000	160	42,512
8000	160	42,512
16000	160	42,512
32000	160	42,512
64000	160	42,512
128000	80	21,256
72000	160	42,512

	con 280	con 280	con 160	con 80
% di ritrasmissione	DatoSecondo Retx	DatoSecondo/minuto	DatoSecondo/minuto	DatoSecondo/minuto
0	0	148,792	85,024	74,396
0,5	2,23244	148,0478533	84,59877333	74,02392667
1	4,46488	147,3037067	84,17354667	73,65185333
2	8,92976	145,8154133	83,32309333	72,90770667
4	17,85952	142,8388267	81,62218667	71,41941333
8	35,71904	136,8856533	78,22037333	68,44282667
16	71,43808	124,9793067	71,41674667	62,48965333
32	142,87616	101,1666133	57,80949333	50,58330667
64	285,75232	53,54122667	30,59498667	26,77061333
80	357,1904	29,72853333	16,98773333	14,86426667
100	446,488	-0,037333333	-0,021333333	-0,018666667

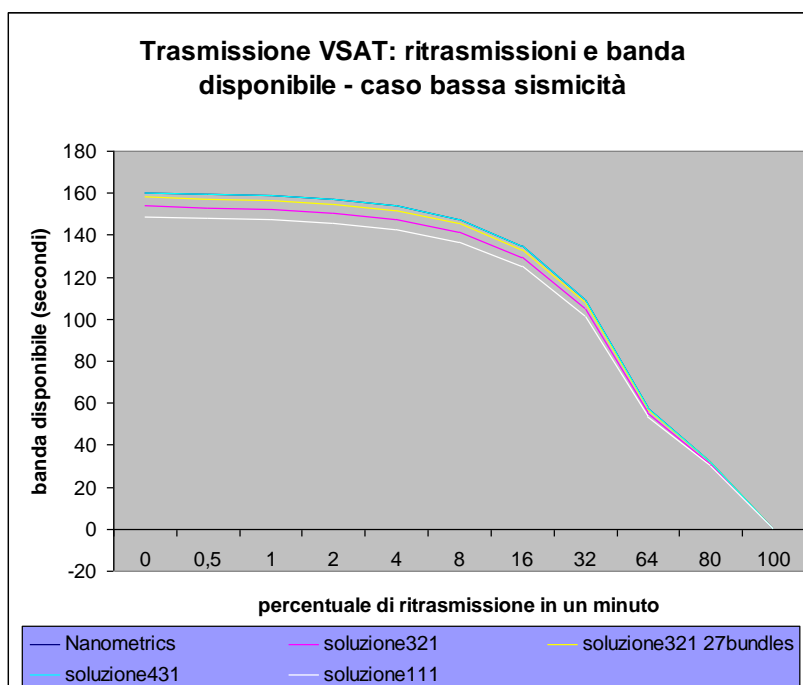


Figura 10.16: grafico che riporta la banda disponibile in base alla percentuale di ritrasmissione presente in un minuto, nel caso di livello di sismicità basso

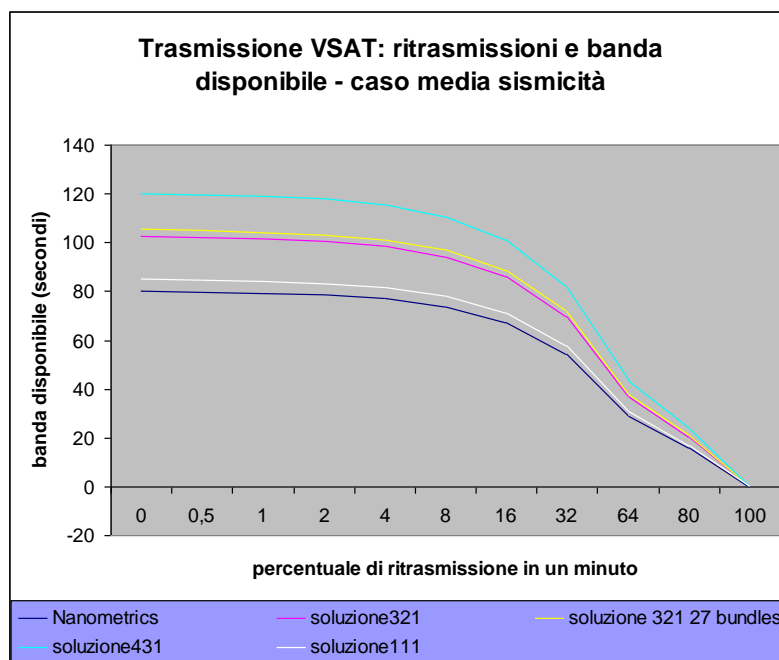


Figura 10.17: grafico che riporta la banda disponibile in base alla percentuale di ritrasmissione presente in un minuto, nel caso di livello di sismicità medio

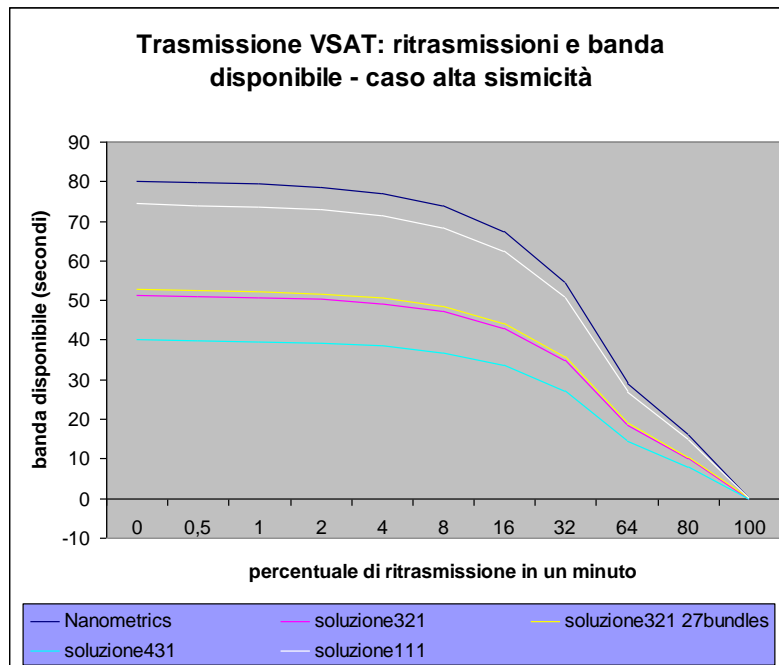


Figura 10.18: grafico che riporta la banda disponibile in base alla percentuale di ritrasmissione presente in un minuto, nel caso di livello di sismicità alto

Appendice 1.

Utilizzo del protocollo Nanometrics[®]:

Implementazione di un Client per lo streaming dei dati meteo

Nel corso dei mesi in cui mi sono occupato dello studio del protocollo di comunicazione utilizzato da Nanometrics[®], in Istituto si è testata della strumentazione nuova. Si è trattato di una piccola stazione meteo da integrare con le stazioni sismiche, per creare un nuovo tipo di stazione multisensore. Lo scopo di queste future installazioni sarà quello di analizzare e scoprire le sorgenti di *noise* esterno causato per esempio dal vento che, soprattutto ad alta quota, batte violentemente sui pannelli fotovoltaici provocando delle vibrazioni nel terreno circostante, vibrazioni che vengono sistematicamente registrate dal sensore sismico. Inoltre si vuole analizzare eventuali legami tra condizioni meteo e segnale sismico acquisito.

La stazione meteo è costituita da 5 sensori che comprendono:

- ❖ anemometro;
- ❖ barometro;
- ❖ termometro;
- ❖ pluviometro;
- ❖ igrometro.

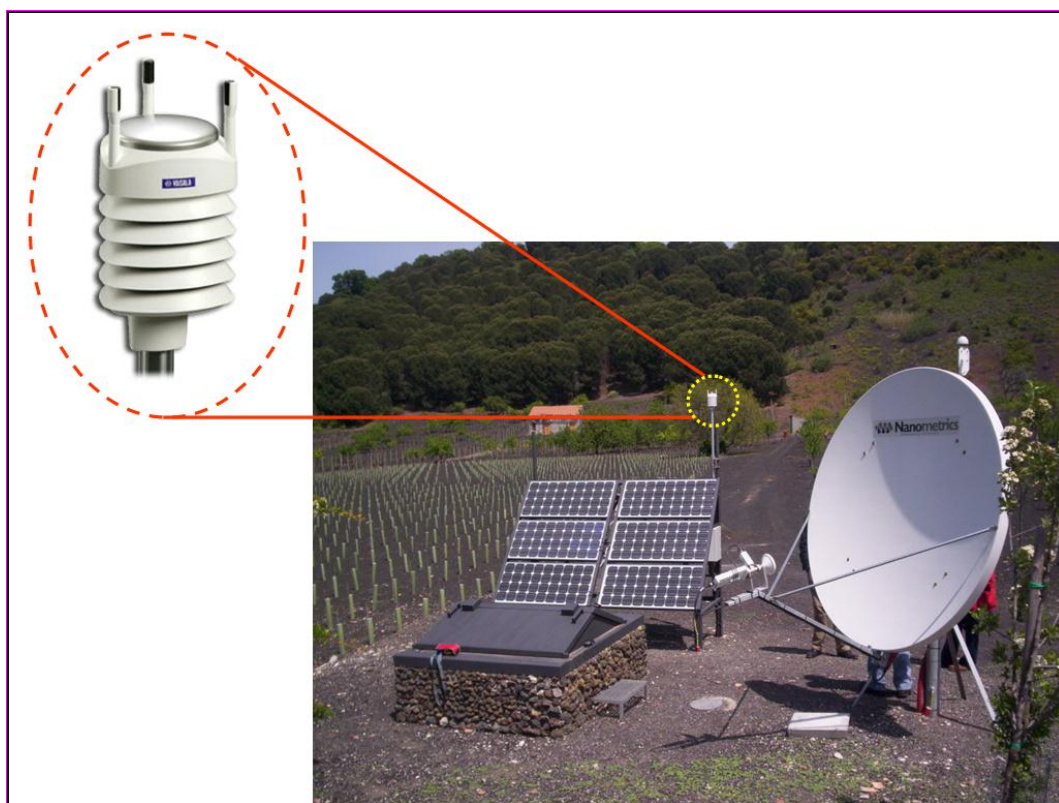


Figura A1: stazione ENIC con in evidenza la stazione meteo

La stazione è fornita di una connessione seriale attraverso cui trasferisce i dati meteo ad un acquisitore esterno. L'acquisitore nel nostro test è stato un Nanometrics[®] Cygnus modificato per acquisire questi dati da seriale.

La stazione trasferisce via satellite i dati meteo fino al centro acquisizione dati dove è presente il server ADNAQS che si occupa dello *storage*.

Il software ***StreamingClient*** è stato realizzato in linguaggio C++.

Esso consiste in un client che si connette alla porta 28000 di ADNAQS o qualunque altro server in cui è installato il software NaqsServer. Quest'ultimo risponde alle richieste di dati provenienti dai client.

Utilizzando uno scambio di messaggi, è possibile richiedere qualunque tipo di dato acquisito con NaqsServer in *near real time*.

Il software presenta un'interfaccia grafica di semplice utilizzo per l'utente finale. Questa consiste di:

- una sezione dedicata alle informazioni del vento, contenente una bussola grafica che visualizza in real time la direzione del vento;
- una sezione dedicata alla pressione atmosferica, temperatura e umidità;
- una sezione dedicata alle precipitazioni e alle loro statistiche;
- un textfield per inserire l'indirizzo o il nome DNS del server dal quale bisogna scaricare i dati;
- una list box per scegliere il canale da acquisire;
- un textfield in cui vengono visualizzati i messaggi di interfaccia con il server e alcune informazioni riguardanti il singolo pacchetto acquisito, come il tipo di dato richiesto e il timestamp;
- un pulsante per iniziare l'acquisizione, un altro per stopparla ed un terzo per uscire dal programma.



Figura A2: interfaccia grafica software StreamingClient

Una difficoltà nell'implementazione di tale software è stata quella riguardante l'animazione della bussola per la direzione del vento. Poiché il software impiega tutto il suo tempo per occuparsi della gestione dell'acquisizione dei pacchetti, l'interfaccia grafica risultava sempre bloccata.

Questo inconveniente è stato risolto utilizzando i **thread**, che garantiscono l'esecuzione contemporanea di azioni diverse.

Nel caso specifico, ho utilizzato due thread paralleli che gestivano uno l'acquisizione dei pacchetti, creato appositamente da me, e l'altro l'animazione della bussola e dell'interfaccia grafica, creato e gestito dall'applicazione in automatico.

```

void CStreamingClientDlg::OnConnetti()
{
    CString st;
    GetDlgItemText(IDC_STATIONS,st);
    if (st=="")
    {
        messaggio.Format("Immetti prima il canale da acquisire!!!");
        for(int u=0;u<=4;u++)
        {
            SetDlgItemText(IDC_MESSAGGIO,messaggio);
            Sleep(500);
            SetDlgItemText(IDC_MESSAGGIO,"");
            Sleep(500);
            SetDlgItemText(IDC_MESSAGGIO,messaggio);
        }
    }
    else
    {
        CreaFile("ptu");
        //si aggiunge l'estensione
        st += ".MET";
        canalerichiesto = st;
        //viene passato il puntatore this alla funzione MainConnection
        miothread = AfxBeginThread(MainConnection,this);
    }
}

```

Figura A3: in evidenza la chiamata al Thread per la connessione e acquisizione dati gestita attraverso la procedura **MainConnection**

La parte fondamentale del software è quella relativa alla comunicazione con il NaqsServer, vi sono diverse funzioni che operano per la lettura dei messaggi provenienti dal server e per il loro processing.

receiveHeader, receiveMessage, receiveData, receiveError, receiveTermination, receiveChannelList sono funzioni che esaminano i messaggi ricevuti dal NaqsServer e procedono con l'operazione adeguata, come quella di individuare una conclusione di comunicazione da parte del server, consentire di ricevere la lista dei canali scaricabili nel server e cominciare a processare le stringhe di dati. **requestChannel, requestSerialChannel, requestSohChannel,**

requestDataChannel, **requestTypeChannel** sono funzioni che abilitano il client ad eseguire una richiesta di qualche tipo.

processData è la funzione che si occupa di analizzare il pacchetto dati per riconoscere cosa si sta ricevendo.

Di seguito si presenta la parte che riguarda l'estrazione delle informazioni contenute nell'header dei pacchetti dati e, nella seconda figura, il processing dei vari tipi di dato.

```
void CStreamingClientDlg::processData(char* buffer, int length)
{
    int bundles = (length - 21) / 17;
    int type = buffer[4];
    int oldestSeq = 0;
    int sequence = 0;
    int timeSecs = 0;
    short timeFrac = 0;
    short instrument = 0;
    short channel = 0;
    short byteCount = 0;

    double pktTime = 0;

    struct tm *newtime;

    // le informazioni contenute dell'header vengono selezionate e copiate in variabili
    // sono nel formato little endian (LSB first)
    memcpy(&oldestSeq, &buffer[0], 4);
    memcpy(&timeSecs, &buffer[5], 4);
    memcpy(&timeFrac, &buffer[9], 2);
    memcpy(&instrument, &buffer[11], 2);
    memcpy(&sequence, &buffer[13], 4);
    memcpy(&byteCount, &buffer[17], 2);
    memcpy(&channel, &buffer[19], 2);
}
```

Figura A4: in evidenza la selezione delle informazioni contenute nell'header dei pacchetti di dati

```

if (type == TIMSER_TYPE)
{
    channel = buffer[17] & 0x07;
    messaggio.Format("Rx time series inst %u:%d seq %6u time %.4f bundles %d",
        instrument, channel, sequence, pktTime, bundles);
}

else if (type == SOH_TYPE)
{
    messaggio.Format("Rx SOH inst %u:soh seq %6u time %.4f bundles %d",
        instrument, sequence, pktTime, bundles);
}

else if (type == SERIAL_TYPE)
{
    char* data = &buffer[21];
    int ix;

    //per quanto riguarda il tempo *****
    CTime tempo(timeSecs);
    giorno = tempo.GetDay();
    mese = tempo.GetMonth();
    anno = tempo.GetYear();
    ora = tempo.GetHour();
    minuti = tempo.GetMinute();
    secondi = tempo.GetSecond();
    messaggio.Format("data e tempo pacchetto: %d/%d/%d time:%d:%d:%d",giorno,mese,anno,ora,minuti,secondi);
    free(&time);
    SetDlgItemText(IDC_MEMO,messaggio);

    nonef.Format("%s%d%d%d",DEFAULTPATH,anno,mese,giorno);
}

```

Figura A5: il processing dei vari tipi di dato

ProcessaMeteo è la procedura che si occupa di decodificare le informazioni contenute nella stringa in formato ASCII creata dalla stazione meteo.

```

OR1, Dn=104D, Dm=119D, Dx=137D, Sn=0.3M, Sm=0.7M, Sx=1.1M
OR1, Dn=324D, Dm=017D, Dx=119D, Sn=0.4M, Sm=0.7M, Sx=1.0M
OR5, Th=20.2C, Vh=0.0#, Vs=13.2V, Vr=3.523V
OR1, Dn=006D, Dm=116D, Dx=139D, Sn=0.3M, Sm=0.7M, Sx=1.0M
OR1, Dn=009D, Dm=029D, Dx=116D, Sn=0.5M, Sm=0.7M, Sx=0.9M
OR5, Th=20.0C, Vh=0.0#, Vs=13.2V, Vr=3.524V
OR1, Dn=307D, Dm=034D, Dx=130D, Sn=0.1M, Sm=0.4M, Sx=0.9M
OR5, Th=19.8C, Vh=0.0#, Vs=13.2V, Vr=3.525V
OR1, Dn=034D, Dm=128D, Dx=144D, Sn=0.6M, Sm=0.9M, Sx=1.3M
OR1, Dn=003D, Dm=036D, Dx=128D, Sn=0.2M, Sm=0.7M, Sx=1.1M
OR1, Dn=302D, Dm=358D, Dx=036D, Sn=0.2M, Sm=0.4M, Sx=0.6M
OR5, Th=20.0C, Vh=0.0#, Vs=13.2V, Vr=3.521V

```

Figura A6: Le stringhe ASCII prodotte dalla stazione meteo

```

void CStreamingClientDlg::ProcessaMeteo(CString input, CTime tempo)
{
    char tipomessaggio;

    int index,posizione,step,fine,postemp;
    CString temp;
    tipomessaggio= input[2];
    step = 3;
    if (input.Find("OR1")!= -1)
    {
        //estrazione direzione minima
        index = input.Find("Dn");
        posizione = index + step;
        postemp = RitornaIndice(input,posizione,'D') -1;
        fine = postemp - posizione;
        windmessage.Dn = input.Mid(posizione,fine);
    }
}

```

Figura A7: La codifica delle stringhe meteo

Le informazioni opportunamente trattate, vengono quindi inserite in strutture dati create per lo storage di gruppi classificabili di informazioni.

```

//estrazione direzione minima
index = input.Find("Dn");
posizione = index + step;
postemp = RitornaIndice(input,posizione,'D') -1;
fine = postemp - posizione;
windmessage.Dn = input.Mid(posizione,fine);

```

Figura A8: lo storage delle informazioni nelle strutture

Alla fine del decoding delle stringhe, si procede con la visualizzazione di queste ultime nell'interfaccia grafica, e alla loro archiviazione in file ASCII giornalieri memorizzati in hard disk.

Le procedure che si occupano dell'archiviazione in hard disk sono **Scrivi**, **ScriviFileptu**, **ScriviFilewnd** e **ScriviFileptr**.

Time	Dm	Dn	Dx	Sm	Sn	Sx
2006512_17:51:49	082	270	241D	6.2	0.2N	14.0N
2006512_17:52:19	090	262	260D	5.2	0.7N	12.4N
2006512_17:52:49	065	278	262D	4.3	1.0N	11.4N
2006512_17:53:19	083	247	239D	8.2	0.4N	15.3N
2006512_17:53:49	100	263	260D	10.5	1.4N	15.4N
2006512_17:54:19	087	301	270D	6.6	1.2N	12.8N
2006512_17:54:49	083	271	265D	8.4	0.8N	13.1N
2006512_17:55:19	076	278	226D	7.4	0.9N	14.9N
2006512_17:55:49	067	274	250D	5.9	1.6N	13.7N
2006512_17:56:19	055	287	213D	4.0	1.2N	8.5N
2006512_17:56:49	062	259	216D	4.5	0.2N	10.3N
2006512_17:57:19	053	272	225D	5.0	1.3N	10.0N
2006512_17:57:49	068	276	230D	4.7	0.5N	10.5N
2006512_17:58:19	063	281	220D	4.5	0.5N	13.3N
2006512_17:58:49	069	262	231D	5.5	0.3N	13.7N
2006512_17:59:19	070	270	220D	3.9	0.5N	12.5N
2006512_17:59:49	070	251	246D	7.7	1.4N	14.3N

Figura A9: un file di archiviazione

Bibliografia

- [1] Susanna Falsaperla, Salvo Alparone, Salvo D'Amico, Giuseppe Di grazia, Ferruccio Ferrari, Horst Langer, Tiziana Sgroi, Salvo Spampinato. *Volcanic Tremor at Mt. Etna, Italy, Preceding and Accompanying the Eruption of July-August, 2001* Pageoph, n° 11, Vol. 162, pp. 2111-2132, 2005.
- [2] Susanna Falsaperla, Giuseppe Di Grazia, Horst Langer. *Volcanic Tremor Location during the 2004 Mount Etna Lava effusion* Geophysical Research Letters, Vol. 33, L04304, doi:10.1029/2005GL025177, 2006.
- [3] Nanometrics[®] Inc. Kanata, Ontario, Canada. *Nanometrics System Software Manuals*, 15 Marzo 2004.
- [4] Nanometrics[®] Inc. Kanata, Ontario, Canada. *Nanometrics System Hardware Manuals*, 15 Marzo 2004.
- [5] Nanometrics[®] Inc. Kanata, Ontario, Canada. *Trillium Seismometer User Guide*, Part Number 13912R9, 03 Giugno 2003.
- [6] Mark Hayman, Ontario, Canada. *HRD24 Manual*, 24 Maggio 2000.
- [7] Vaisala[®] Oyj. *Weather Transmitter WXT510 User's Guide*. M210470EN-C, Luglio 2005.
- [8] Gérard Maral, *VSAT Networks – Second edition*. Wiley 2003.
- [9] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest *Introduzione agli algoritmi*. 2003.
- [10] Brian W. Kernighan, Dennis M. Ritchie. *Linguaggio C –seconda edizione* , Jackson Libri 1989.
- [11] Guido Buzzi Ferraris. *Microsoft Visual C++ Applicazioni scientifiche*. Mondadori Informatica, febbraio 2000.
- [12] Giuseppe Biondo, Enrico Sacchi. *Manuale di elettronica e telecomunicazioni*. HOEPLI, 1994.
- [13] Wikipedia. *Audio Data Compression*.

http://en.wikipedia.org/wiki/Audio_data_compression.

[14] Greg Goebel, Introduction / Lossless Data Compression, v1.1.1, chapter 1, 1 aprile 2005.
<http://www.vectorsite.net/ttdcmp1.html#m3>.

[15] Massachusetts Institute of Technology Centre for Space Research Cambridge. *Huffman Coding of ACIS Pixel Data Part Number 36-56102, Version 01°*, 29 Luglio 1996.
<http://acis.mit.edu/acis/huff/>

[16] D.A.Huffman, *A method for the Construction of Minimum-Redundancy Codes*, *Proceedings of the Institute of Radio Engineers*, vol. 40, pp. 1098-1101, 1952.