

SISTEMI INTEGRATI PER IL MONITORAGGIO, L'EARLY WARNING
E LA MITIGAZIONE DEL RISCHIO IDROGEOLOGICO
LUNGO LE GRANDI VIE DI COMUNICAZIONE



UNIONE EUROPEA
Fondo europeo di sviluppo regionale



PON Ricerca e Competitività
2007-2013



Ministero dell'Università e della Ricerca



Ministero dello Sviluppo Economico

investiamo nel vostro futuro

PROGETTO PON01_01503



Quaderno
16 PON LEWIS
CAED.
ACQUISIZIONE DATI:
ARCHITETTURA DEL SISTEMA



autostrade//Tech



DELIVERABLE WP 6.1

CAED. Acquisizione dati: architettura del sistema

Sistemi integrati per il monitoraggio, l'early warning e la mitigazione del rischio idrogeologico lungo le grandi vie di comunicazione"

Premessa

Frane e inondazioni sono un problema di grande rilevanza nel nostro Paese. Negli ultimi anni le vittime e i danni dei disastri idrogeologici hanno raggiunto livelli inaccettabili e impongono un grande e immediato impegno della comunità nazionale per cercare di mitigare il livello di rischio, utilizzando strategie articolate ed efficaci capaci di integrare, in una visione organica, interventi strutturali e non strutturali.

Su questi temi l'Università della Calabria è impegnata da anni in attività di studio e di ricerca di rilevanza nazionale e internazionale e nella diffusione e promozione della cultura della previsione e prevenzione del rischio idrogeologico. Nel 2011 insieme ad altri partner, ha promosso un progetto di ricerca triennale, "Sistemi integrati per il monitoraggio, l'early warning e la mitigazione del rischio idrogeologico lungo le grandi vie di comunicazione", finalizzato allo sviluppo di un sistema complesso e articolato di preannuncio delle frane da impiegare per le fasi di previsione/prevenzione del rischio idrogeologico.

Il Progetto, indicato con l'acronimo LEWIS (Landslide Early Warning Integrated System), è stato svolto, nel periodo 2012-2014, nel quadro del Programma Operativo Nazionale 2007-13 "Ricerca e Competitività".

I risultati conseguiti sono descritti in questa collana di Quaderni PON LEWIS.

Il progetto è stato sviluppato dall'Università della Calabria e Autostrade Tech S.p.A. insieme ai partner industriali Strago e TDGroup, alle Università di Catania, di Reggio Calabria e di Firenze e al CINID (Consorzio Interuniversitario per l'Idrologia). Per l'Ateneo calabrese hanno partecipato diversi laboratori e gruppi di ricerca: CAMILab (con funzione di coordinamento), μ Wave, Geomatica, Nems, Geotecnica, Dipartimento di matematica.

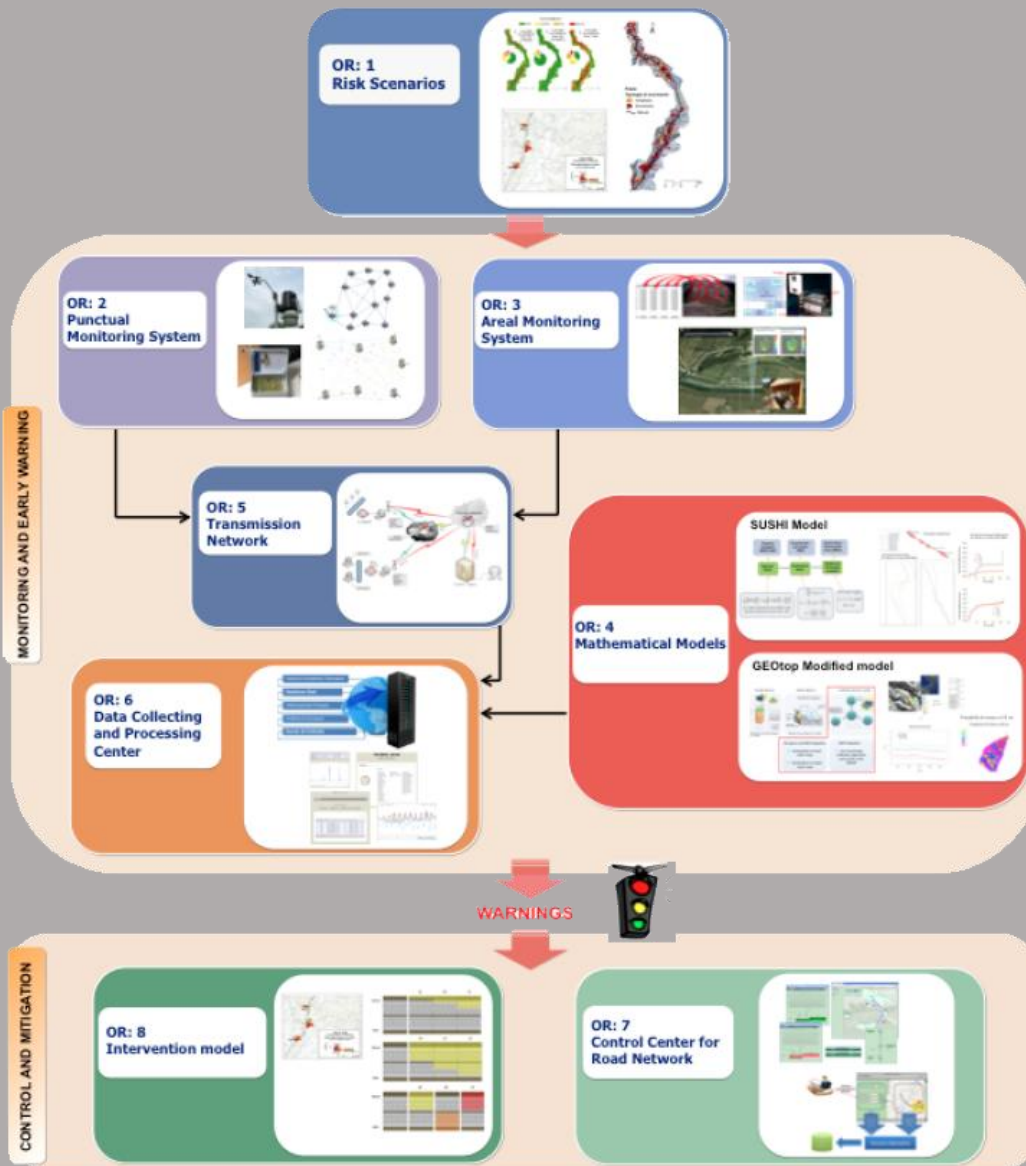


Figura 1 - Articolazione del sistema integrato di monitoraggio dei versanti e di preannuncio dei movimenti franosi

Il progetto è finalizzato allo sviluppo di un sistema di monitoraggio dei versanti e di preannuncio dei movimenti franosi che possono interessare le grandi vie di comunicazione e all'identificazione dei conseguenti interventi non strutturali di mitigazione.

Il sistema è articolato in due sottosistemi (fig. 1):

- ✓ Monitoraggio e preannuncio,
- ✓ Controllo e mitigazione,

che richiedono la preventiva individuazione degli scenari di rischio ossia dei danni che l'eventuale attivazione di una frana può produrre sugli elementi a rischio presenti (infrastruttura viaria, autoveicoli, persone). La procedura originale sviluppata nell'ambito del progetto prevede l'identificazione, lungo il tratto autostradale di interesse, delle aree soggette a movimenti franosi e la conseguente definizione dei relativi scenari di evento e di rischio.

Il sottosistema *Monitoraggio e preannuncio* è formato da diverse componenti: rete di monitoraggio "puntuale" che comprende sensori che misurano localmente l'inizio degli spostamenti superficiali o profondi; rete di monitoraggio "areale" che include sensori che controllano a distanza il fenomeno franoso con tecniche radar; modelli matematici di simulazione dell'innesco e della propagazione dei movimenti franosi. Nel progetto LEWIS sono state sviluppate numerose componenti innovative e sono state modificate e migliorate altre componenti già esistenti. In particolare tra i sensori puntuali sono stati sviluppati i sistemi SMAMID e POIS; tra quelli areali sono stati realizzati un radar in banda L, uno scatterometro, un interferometro; tra i modelli si sono sviluppati e/o migliorati: GEOtop, SUSHI, SCIDDICA.

La raccolta dei dati misurati dai sensori è affidata ad un unico sistema di trasmissione dati che trasmette anche le informazioni necessarie per il funzionamento dei modelli. Il sottosistema è completato da un Centro di acquisizione ed elaborazione dei dati (CAED) che, sulla base dei dati misurati dai sensori e delle indicazioni dei modelli, valuta la situazione di pericolo lungo il tronco autostradale emettendo i relativi livelli di criticità.

I livelli di criticità emessi dal CAED sono l'elemento di collegamento tra il sottosistema *Monitoraggio e preannuncio* e il sottosistema *Controllo e mitigazione*. Gli avvisi di criticità sono acquisiti dal Centro di comando e controllo del traffico (CCCT) che, sulla base di un modello di intervento predefinito, attiva le procedure standardizzate per la mitigazione del rischio, che vanno dalla

sorveglianza diretta del tratto di interesse da parte di squadre tecniche all'interruzione del traffico su entrambe le direzioni di marcia.

Il progetto prevede anche lo sviluppo di attività sperimentali su tre tronchi autostradali lungo la A3, la A16 e la A18, nonché l'erogazione di un Master di secondo livello denominato ESPRI (ESperto in Previsione/Prevenzione Rischio Idrogeologico).

Il progetto di ricerca è stato organizzato in Obiettivi Realizzativi (OR), ciascuno dei quali suddiviso in Work Package (WP), a loro volta articolati in Attività Elementari (AE). In totale erano previste 11 OR, 47 WP e 243 AE. In particolare le OR 1-8 riguardano la ricerca e si articolano in 26 WP e 139 AE. Le OR 9-11 sono dedicate a sperimentazione, governance e trasferimento tecnologico, integrazione e aggiornamento dell'attività di ricerca nella fase di Sviluppo Sperimentale e si articolano complessivamente in 21 WP e 104 AE.

I Quaderni che compongono questa collana sono stati costruiti con riferimento ai singoli WP, per la parte che riguarda la ricerca, e quindi ogni Quaderno contiene la descrizione dei risultati conseguiti nel WP, articolata in base alle AE previste.

Sono, inoltre, previsti altri tre Quaderni:

Quaderno 0 che contiene una descrizione di sintesi, in inglese, dei risultati conseguiti nell'ambito del progetto.

Quaderno 28 che contiene l'informazione relativa alle attività di divulgazione dei risultati scientifici.

Quaderno 29 che contiene la descrizione dei risultati conseguiti con l'attività formativa.

Il Quadro editoriale complessivo è riportato in tabella 1:

QUADERNO	OR	WP	TITOLO
0	-	-	Research outcomes
01 Parte prima	1	1.1	Linee guida per l'identificazione di scenari di rischio
01 Parte seconda	1	1.1	Linee guida per l'identificazione di scenari di rischio
02	2	2.1	Monitoraggio idrogeologico
03 Parte prima	2	2.2	Monitoraggio con unità accelerometriche (Sistema SMAMID)
03 Parte seconda	2	2.2	Monitoraggio con unità accelerometriche (Sistema SMAMID)
04	2	2.3	Circuiti integrati a bassa potenza per sistemi di monitoraggio con unità accelerometriche
05	2	2.4	Monitoraggio con sensori puntuali di posizione e inclinazione (Sistema POIS)
06	3	3.1	Sviluppo di uno scatterometro a risoluzione variabile
07	3	3.2	Elettronica di bordo dello scatterometro ed inclinazione
08	3	3.3	Sviluppo di un radar in banda L
09	3	3.4	Tecniche di analisi e sintesi di segnali radar per la simulazione accurata di scenari complessi
10	3	3.5	Elettronica di bordo del radar in banda L

QUADERNO	OR	WP	TITOLO
11	3	3.6	Sistemi interferometrici radar ad apertura sintetica basati a terra
12	4	4.1	Modello areale per il preannuncio delle frane da innesco pluviale (Modello GEOtop)
13	4	4.2	Modelli completi di versante di tipo puntuale per il preannuncio di movimenti franosi (Modello SUSHI)
14	4	4.3	Modelli di propagazione delle frane tipo colate (Modello SCIDDICA)
15	5	5.1	Rete Wireless di Telecomunicazioni: sviluppo e scelta dei parametri di progetto
16	6	6.1	CAED. Acquisizione dati: architettura del sistema
17	6	6.2	CAED. Elaborazione dei dati
18	7	7.1	CCCT. Progettazione
19	7	7.2	CCCT. Interfaccia verso il centro di acquisizione ed elaborazione dati
20	7	7.3	CCCT. Interfaccia con altre centrali operative e canali di diffusione delle notizie
21	7	7.4	CCCT. Modulo per la presentazione e convalida delle allerte
22	7	7.5	CCCT. Modulo per la gestione delle informazioni di traffico
23	7	7.6	CCCT. Integrazioni con moduli speciali

QUADERNO	OR	WP	TITOLO
24	8	8.1	Definizione del modello di intervento e predisposizione del Piano di Emergenza
25	8	8.2	CCCT. Gestione delivery allerte e attivazione squadre d'intervento
26	8	8.3	CCCT. Gestione percorsi alternativi
27	9	9.1 - 9.11	Sperimentazione
28	10	10.1 - 10.2	Piano di comunicazione e diffusione
29	-	-	Master ESPRI (Esperto in Previsione/Prevenzione Rischio Idrogeologico)

Tabella 1 - Quadro editoriale complessivo della collana di Quaderni PON LEWIS

31 dicembre 2014

Il Responsabile Scientifico del progetto PON LEWIS

Pasquale Versace

INDICE

ATTIVITA' ELEMENTARE 6.1.1

1 **Analisi delle WSN presenti in letteratura**

1 **6.1.1.A Ricerca bibliografica sulle WSN**

2 **6.1.1.B WSN: introduzione e campi di applicazione**

ATTIVITA' ELEMENTARE 6.1.2

3 **Supporto alla scelta dei dispositivi**

3 **6.1.2.A Tipologia di reti wireless**

6 **6.1.2.B Unità Hardware**

7 **6.1.2.C Requisiti Hardware e Software dei dispositivi**

ATTIVITA' ELEMENTARE 6.1.3

11 **Modalità invio dati**

ATTIVITA' ELEMENTARE 6.1.4

14 **Modalità connessione e scambio dati**

14 **6.1.4.A Servizio di acquisizione dei dati: AqSERV**

26 **6.1.4.B Message Protocol**

ATTIVITA' ELEMENTARE 6.1.5

30 **Testing piattaforma acquisizione WSN**

30 **6.1.5.A Simulatore di reti WSN**

39 **6.1.5.B Rete WSN di laboratorio**

45 **6.1.5.C Testing con i progettisti delle reti di sensori**

ATTIVITA' ELEMENTARE 6.1.6

58 **Architettura CAED**

59 **6.1.6.A Architettura hardware**

60 **6.1.6.B Architettura software CAED**

61 **6.1.6.C Implementazione del database**

62 **6.1.6.D Modello relazionale**

ATTIVITA' ELEMENTARE 6.1.7

63 **Trattamento dei dati**

63 **6.1.7.A Estrazione dei dati dai messaggi**

65 **6.1.7.B Log e storage dei messaggi grezzi**

66 **6.1.7.C Acquisizioni campionarie**

ATTIVITA' ELEMENTARE 6.1.8

67 **Classificazione dei codici di errore e registrazione eventi (Event Log System)**

ATTIVITA' ELEMENTARE 6.1.9

72 **Acquisizione integrazione e test**

74 **6.1.9.A Allegati**

ALLEGATO A

77 **Moduli registrazione dispositivi**

ALLEGATO B

97 **Moduli registrazione modelli di preannuncio**



Analisi delle WSN presenti in letteratura

6.1.1.A RICERCA BIBLIOGRAFICA SULLE WSN

Preliminarmente alle problematiche della progettazione e realizzazione del centro di acquisizione ed elaborazione dati (CAED), è stato necessario conoscere ed approfondire l'architettura delle reti di monitoraggio implementate nel presente progetto: le Wireless Sensor Network (WSN). Le applicazioni WSN, nell'ultimo decennio, hanno subito un notevole incremento di utilizzo grazie alla loro semplicità di applicazione unitamente ai bassi costi di installazione, produzione e manutenzione. Nella prima parte dell'attività elementare è stata fondamentale una ricerca bibliografica sulle applicazioni delle WSN nel monitoraggio ambientale. Di seguito le risorse di letteratura consultate:

- Seongeun Yoo, Jaeon Kim, Taehong Kim, Sungjin Ahn, Jongwoo Sung, Daeyoung Kim, "A2S: Automated Agriculture System based on WSN". In ISCE2007, Dallas, Texas, 20~23 June, 2007.
- Hui Liu, Zhijun Meng, Shuanghu Cui, "A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses". In Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference, 2125 Sept. 2007. Page(s): 2344 – 2347.
- Aline Baggio, "Wireless sensor networks in precision agriculture". In Proc. ACM Workshop RealWorld Wireless Sensor Networks, 2005.
- R. CardellOliver, K. Smettem, M. Kranz, and K. Mayer, "A Reactive Soil Moisture Sensor Network: Design and Field Evaluation". In Distributed Sensor Networks, vol.1, no.2, pp.149 – 162 , 2005.
- M. Minami, T. Morito, H. Morikawa, and T. Aoyama, "Solar biscuit: A batteryless WirelessSensor Network system for environmental monitoring applications". In The 2nd International Workshop on Networked Sensing Systems, 2005.
- Barrenetxea G., Ingelrest F., Schaefer G., Vetterli M., Couch O., Parlange M., "SensorScope: OutoftheBox Environmental Monitoring".

In Information Processing in Sensor Networks, 2008. IPSN '08. Publication Date: 22~24 April 2008 On page(s): 332 – 343. ISBN: 9780769531571.

Le pubblicazioni sono risultate fondamentali per conoscere i termini tecnici delle WSN, modalità di progettazione, limiti hardware e software dell'infrastruttura ed i possibili protocolli utilizzati per collegare le reti di sensori al CAED.

6.1.1.B WSN: INTRODUZIONE E CAMPI DI APPLICAZIONE

Le reti di sensori wireless (WSN) sono costituite da una rete di piccoli dispositivi elettronici, dotati di processori a bassissime prestazioni, con piccole memorie RAM, che distribuiti sul territorio da monitorare, forniscono le caratteristiche fisiche dell'ambiente. Tipicamente le soluzioni cablate (wired) per il monitoraggio ambientale sono state quelle maggiormente utilizzate nello scorso secolo. Le limitazioni delle connessioni cablate, quali gli ambienti ospitali ed i costi di installazione, sono state superate efficacemente dalle connessioni wireless e risultano essere quelle maggiormente utilizzate per questo tipo di problemi.

Ogni dispositivo, costituente la rete, è dotato di una serie di diversi sensori, con i quali valuta periodicamente le condizioni ambientali in cui si trova per poi inviare tali misure ad un nodo centrale. Tale nodo centrale, nell'ambito del progetto, è costituito dal CAED che ha il compito di raccogliere le misure dai sensori ed elaborarli in una *informazione* per la previsione di eventi in atto o potenziali.

Nonostante i dispositivi siano dotati di un hardware limitato (paragonato ai normali personal computer) devono implementare numerose funzionalità per poter permettere un monitoraggio continuo. Ogni nodo, infatti, deve ottimizzare i flussi di comunicazione provenienti dai moduli radio associati, minimizzare i consumi energetici per garantire l'autonomia di ciascun dispositivo ed effettuare le misurazioni quando richiesto ovvero ad intervalli predeterminati.

In base ai sensori che vengono utilizzati per il monitoraggio e alle elaborazioni effettuate, le WSN possono essere applicate al campo militare, ambientale, sanitario, urbano e domestico. In particolare, numerose applicazioni sono state effettuate in campo ambientale e nello specifico, grazie ad una crescente offerta tecnologica, al monitoraggio in real-time delle grandezze fisiche che possono

fornire informazioni sull'evoluzione dei fenomeni naturali come quelle relative al rischio di frana.

Accanto ai numerosi vantaggi nell'utilizzo delle WSN per il monitoraggio ambientale, per la progettazione della comunicazione ottimale tra il CAED e le WSN, è stato necessario conoscere gli svantaggi di queste reti. Le WSN sono caratterizzate da alcuni aspetti chiave come i vincoli di potenza elaborativa, la durata limitata della batteria, un passo duty-cycle e connessioni molti a uno, che creano delle sfide progettuali nell'ambito del trasporto dei dati, gestione della rete, disponibilità, confidenzialità e integrità. Queste considerazioni sono state utilizzate come base per la progettazione del protocollo di comunicazione verso il CAED e per la corretta gestione da remoto delle reti (par. 6.1.5).

ATTIVITÀ ELEMENTARE 6.1.2

Supporto alla scelta dei dispositivi

Lo scopo principale di questa attività è di fornire supporto ai progettisti delle WSN, per una integrazione ottimale con i servizi messi a disposizione dal CAED. Nella prima parte dell'attività, in concomitanza con l'attività precedente (AE 6.1.1), sono descritte le topologie delle reti da dover gestire, le unità hardware che compongono i dispositivi delle WSN e sono indicati i requisiti minimi da dover soddisfare da parte dei progettisti di WSN, sia per l'acquisizione dei dati che per la gestione da remoto delle reti da parte degli operatori del CAED.

6.1.2.A TIPOLOGIA DI RETI WIRELESS

Tutti i benefici derivanti dalle reti di sensori wireless dipendono dall'architettura che queste reti assumono per assolvere i loro compiti, obiettivi diversi impongono schemi e modelli applicativi differenti per superare le difficoltà legate ai consumi energetici e all'affidabilità generale del sistema. In base agli obiettivi del sistema si può disporre di WSN dense, composte da tante unità che svolgono il medesimo compito, allo scopo di produrre informazioni statisticamente rilevanti, oppure WSN composte da meno unità computazionali ma con data-rate (tasso di

trasmissione dati) superiore e logiche applicative più evolute. È quindi possibile classificare le reti wireless anche in base al consumo e al data-rate utilizzato. Il rapporto tra consumo e velocità di trasmissione dati risulta essere proporzionale: in generale, alti data-rate implicano consumi elevati e viceversa (Fig. 6.1.1).

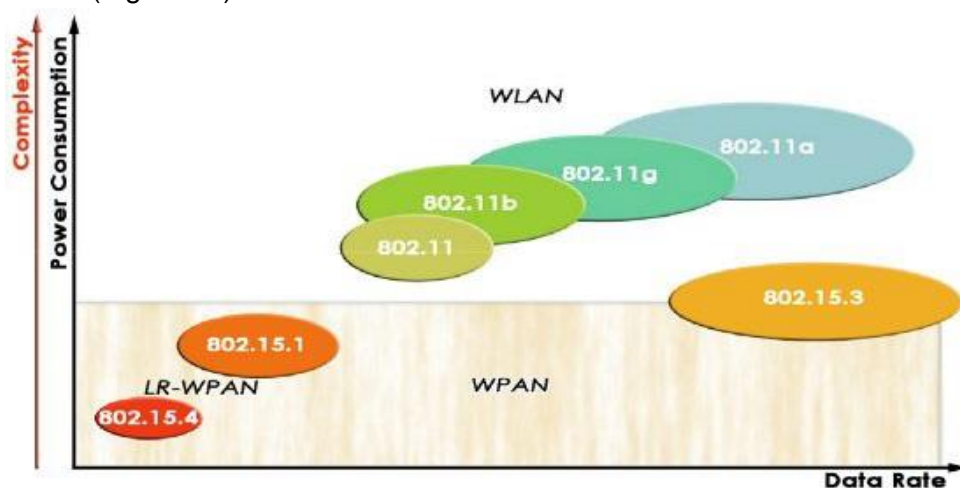


Fig. 6.1.1 – Relazione tra le varie tipologie di protocolli wireless, rispetto al data-rate e ai consumi energetici.

Considerando il primo caso (elevato numero di numero di unità) le reti di sensori wireless sono composte da mini computer o sistemi embedded dotate di almeno un'interfaccia di rete per stabilire un collegamento di scambio dati. Il collegamento è wireless a radio frequenza, al quale viene spesso adoperato un protocollo di comunicazione costruito sul popolare standard IEEE 802.15.4 (Low-Rate Wireless PAN), come ad esempio ZigBee, ZigBee PRO, WirelessHART, MiWi, o altre soluzioni proprietarie. La scelta del protocollo di comunicazione è sempre vincolato da due fattori fondamentali: il data-rate desiderato ed i consumi che le unità possono sostenere. In questo genere di WSN, l'uso di protocolli *general purpose*, come ad esempio la suite di protocolli Internet (TCP/IP), è solitamente evitato a causa di un impatto elevato dal punto di vista computazionale derivato dalla sua implementazione, decisamente oltre le esigenze di una WSN. Queste reti sono collegate direttamente ad un'altra unità computazionale di tipo superiore, che raccoglie i dati trasmessi dalle unità foglia (**nodì di 1° livello**) e, per mezzo di una interfaccia di rete TCP/IP accede nelle

VPN (Virtual Private Network) della trasmissione dati tramite Internet. Questa unità denominata **nodo coordinatore**, solitamente è dotata di diverse interfacce di rete (GSM/3G, 802.11, 2.4 GHz/5 GHz) per la connettività verso l'esterno della WSN. Il nodo coordinatore raccoglie le informazioni in arrivo dagli altri nodi e invia i dati al CAED.

La collocazione dei dispositivi ed i collegamenti fisici e logici che interconnettono i vari nodi della rete definiscono le varie topologie che si possono implementare: rete a stella, mesh e ad albero.

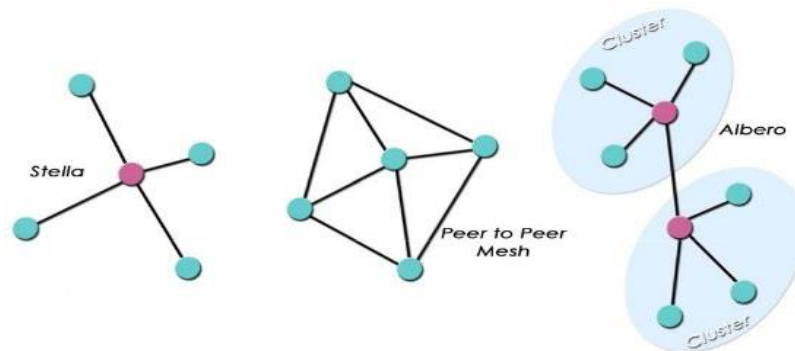


Fig. 6.1.2 – Topologie delle reti impiegate nel progetto.

La rete a stella è una rete che pone un dispositivo di rete coordinatore al quale sono collegati fisicamente tutti gli altri dispositivi. Questa tipologia è sicuramente quella migliore dal punto di vista del risparmio energetico tenendo in considerazione, però, che il dispositivo al quale sono collegati tutti gli altri consumerà inevitabilmente più energia rispetto agli altri. La rete a maglia (mesh) è una rete dove ogni dispositivo è collegato a tutti i suoi più prossimi e ognuno di questi svolge, all'occorrenza, compiti di routing. Sebbene la rete mesh sia logicamente più efficiente, consentendo di coprire vaste estensioni territoriali mediante configurazioni generalizzate, presenta però consumi maggiori e una minore adeguatezza nel caso in cui l'installazione debba essere fissa e non mobile. In queste condizioni è preferibile adottare soluzioni di reti a stella, collegate fra loro per mezzo di una dorsale ottenendo risultati con consumi estremamente ridotti. Infine, la struttura ad albero è una topologia di rete che combina la struttura a stella e quella a mesh. Più nodi, che fanno capo ad un unico nodo, formano un gruppo chiamato cluster. I nodi-coordinatori possono far riferimento a loro volta ad un nodo-coordinatore di livello superiore costruendo, così, vari livelli di gerarchia.

6.1.2.B UNITÀ HARDWARE

La specificità della tipologia di rete progettata impone Hardware Embedded, ovvero Hardware di elaborazione a microprocessore progettato appositamente per una determinata applicazione (special purpose), inteso come piattaforma hardware ad hoc. I componenti hardware dei nodi di una WSN comprendono CPU, memoria, radio, sensori, attuatori e fonte di alimentazione.

- **Controller** è la CPU del nodo e si occupa di rilevare i dati tramite i sensori, elaborarli, decidere dove e quando spedirli, ricevere dati da altri nodi, gestire gli attuatori e i sensori. I componenti più adatti a questo scopo sono i microcontrollori, ovvero processori per sistemi embedded, poiché richiedono poca energia, hanno spesso delle memoria integrata e non hanno controller di memoria, si integrano facilmente con i sensori e gli attuatori e presentano delle modalità di risparmio energetico;
- **Memoria** memoria volatile e di massa, in questo caso proprie del microcontrollore, che vengono utilizzate rispettivamente per salvare le variabili durante l'esecuzione e per memorizzare il codice da eseguire;
- **Modulo radio** componente fondamentale delle WSN costituito da un transceiver ad onde radio che, solitamente, opera nella banda ISM secondo lo standard IEEE 802.15.4;
- **Sensori e attuatori** sono le unità fondamentali per valutare le grandezze fisiche dell'ambiente circostante (sensori passivi e attivi) ovvero effettuare delle modifiche a particolari circuiti (attuatori);
- **Unità energetica** il nodo è di norma alimentato da una batteria (preferibilmente al litio) che ne assicura il funzionamento anche in luoghi poco accessibili per una durata pari a quella della WSN. Per limitare i consumi delle unità si devono prevedere degli algoritmi specifici sia per le unità trasmissive che per le unità computazionali, per le prime si ricorre spesso ad un sistema di wake-up/sleep per le seconde si preferisce abbassare il clock del controllore, in relazione al carico computazionale richiesto. Molto spesso, la garanzia della continuità dell'alimentazione è garantita da piccole celle fotovoltaiche.

La piattaforma così composta può assumere, all'interno della WSN, il ruolo di nodo di primo livello (nel seguito denominato *mote*) o nodo coordinatore (nel seguito denominato *sink*). I mote sono i nodi ordinari della WSN, raccolgono dati attraverso la sensoristica e la inviano all'interno della rete. I sink svolgono la

funzione di nodi concentratori, la differenza tra questi ed i primi è che i sink non montano sensori e solitamente sono dotati della possibilità di comunicare attraverso diversi protocolli, su altrettante interfacce di rete, al fine di compiere la funzione di gateway verso livelli di rete superiori. Spesso i sink sono composti da Hardware embedded più complesso o di maggiore capacità computazionale che, oltre ad inglobare le unità comuni ai mote, implementano Architetture Hardware che consentono l'utilizzo di un sistema operativo evoluto come ad esempio GNU/Linux o simili Unix Based.

6.1.2.C REQUISITI HARDWARE E SOFTWARE DEI DISPOSITIVI

Sulla base delle conoscenze acquisite dalla ricerca bibliografica e in base agli obiettivi richiesti dal progetto di ricerca sono state indicate le caratteristiche minime che devono possedere le WSN per poter comunicare correttamente con il CAED. Tali caratteristiche sono:

- **Quality of Service (QoS):** qualità e quantità dei dati ricevuti da un nodo coordinatore di una specifica area (es. affidabilità nel rilevamento di alcuni eventi, priorità dei pacchetti più recenti, rilievo delle grandezze anche in assenza di connessione verso il CAED, ecc.);
- **Tolleranza ai guasti:** nel caso di fault da parte di un nodo (esaurimento dell'alimentazione, rottura fisica o perdita di collegamento) l'intera rete non deve risentire di questo malfunzionamento e si deve prevedere un altro nodo che svolga le sue funzioni;
- **Lifetime:** l'obiettivo delle WSN sono di rimanere attive per più tempo possibile e perciò si devono prevedere tutti gli accorgimenti idonei per soddisfare questo requisito, in base a quanto già discusso precedentemente. Questo aspetto deve trovare il giusto compromesso con gli eventi monitorati: in fase ordinaria (assenza di eventi significativi) si possono utilizzare soluzioni conservative, mentre in fase di criticità è necessario avere un numero elevato di acquisizioni;
- **Scalabilità:** numero di nodi elevato con architettura in grado di gestirli senza eccessive difficoltà;
- **Densità di nodi:** la densità dei nodi può variare nello spazio e nel tempo e la rete deve sapersi adattare a queste nuove conformazioni;

- **Programmabilità:** i nodi devono essere in grado di cambiare i propri compiti in relazione alle richieste dal CAED ovvero devono poter essere riprogrammabili;
- **Manutenzione:** la rete deve essere in grado di adattarsi, monitorando il proprio stato di salute, aggiornando i propri parametri, decidendo tra nuovi compromessi (ad esempio diminuendo la qualità del servizio quando l'energia sta per terminare).

Per soddisfare queste caratteristiche è stata fornita una serie di requisiti che il mote ed il sink *dovrebbero* possedere (tab. 6.1.1, 6.1.2).

Priorità	Requisito	Descrizione
M1	Intestazione del messaggio completa	Gli header dei messaggi devono contenere Data, ora, minuti, secondi. Il MAC address del dispositivo può essere contenuto in questo oppure essere specificato nel campo successivo.
M2	Uniformità del messaggio	Tutti i motes di tutte le WSN devono produrre un messaggio con medesima struttura sintattica, utilizzando il medesimo separatore di campo per il raggruppamento dei campi e il medesimo separatore di campo, ma diverso dal primo, per le variabili multiple acquisite da un sensore specifico. Esempio: [2012-03-08 23:20:13354]-mac:0013a20040307f9c - x:27,y:23,z:1023 -temp:28 -bat:97% In questo caso l'intestazione è fra parentesi quadre. Il delimitatore di campo dei dati è lo spazio. Il delimitatore delle variabili di un sensore è la virgola, come nell'esempio dell'accelerometro (-x,y,z).
M3	Univocità dei campi delle acquisizioni	I nomi delle variabili delle acquisizioni devono essere univoci affinché non coesistano all'interno di un messaggio possibili collisioni di significato. Ogni variabile deve avere un nome univoco (-temp, -bat, -x,y,z)
M4	Tipologia di rete multipla	I mote possono essere configurati per comunicare a stella, ad albero o a mesh.
M5	Risparmio e autonomia energetica	Di comune interesse per tutte le OR. Possibilità di alimentazione della batteria tramite piccoli pannelli solari.
M6	Accessibilità, usabilità	Documentazione, risorse e plug agevolato su postazioni di sviluppo al fine di condurre analisi, diagnosi o migliorie software della piattaforma autonomamente.

M7	Modularità	Sensori ed espansioni possono essere montate, gestite e modificate indipendentemente da come fornite dal produttore
M8	Riprogrammazione e Remota	Programming-over-the-air per consentire di modificare scheduling e applicazione dei sensori da remoto, agendo dalla sala di comando del CAED direttamente sul dispositivo Coordinatore (Sinks)

Tab 6.1.1 – Requisiti del mote.

Priorità	Requisito	Definizione
S1	Multiprotocollo	Il Sink deve possedere connettività: <ol style="list-style-type: none"> 1. ZigBee-PRO 2. Wifi 2.4GHz 802.11 a-g 3. Wifi Mesh 2.4GHz/5GHz 4. GPRS 5. Ethernet
S2	Alimentazione multipla	Consentire il collegamento sia alla rete di distribuzione ordinaria (AC 220V) che a quella a batteria fotovoltaica/autovettura (DC 12V)
S3	Sistema Operativo GNU/Linux	Consentire di implementare procedure scritte con linguaggi di alto livello, scheduling e controlli personalizzati. Consentire di sfruttare la potenza e la flessibilità lato network del kernel Linux
S4	Antenne sostituibili	Consentire di sostituire una antenna o applicare un amplificatore in caso di trasmissione a distanza elevata (>1km)
S5	Capacità computazionale adeguata	Processore min 500mhz Ram min 256MB Storage min 8GB
S6	Gestione remota Facilitata	Accesso web, accesso ssh. Possibilità di sviluppare strumenti amministrativi ad-hoc

Tab 6.1.2 – Requisiti del sink.

I requisiti delle tab. 6.1.1 e 6.1.2 sono stati definiti nella prima parte del progetto, quando ancora erano disponibili poche informazioni da parte degli sviluppatori delle WSN. Molte delle caratteristiche sopra citate sono state

soddisfatte, altre, in una visione indipendente delle reti di monitoraggio rispetto al CAED non sono state garantite.

Nonostante queste limitazioni, da parte degli sviluppatori/progettisti delle reti di monitoraggio è stata garantita la programmabilità dei nodi e del comportamento della rete in relazione alle esigenze operative del CAED.

Nel caso di malfunzionamenti della rete, il CAED comunica tempestivamente ai singoli gestori delle reti i nodi che presentano comportamenti anomali, ma non può, in nessun caso, intervenire per il ripristino da remoto.

Modalità invio dati

L'obiettivo della presente attività è stato di esaminare le tipologie di connessione tra le WSN ed il CAED, fornendo indicazioni alla rete di trasmissione (OR 5) da implementare sui tratti di autostrada da monitorare.

Lungo le unità geomorfologiche monitorate, per come è stata realizzato il progetto di ricerca, possono coesistere più reti WSN. Le reti WSN, per come descritte nel paragrafo precedente, sono dotate di un nodo coordinatore che può comunicare direttamente con il CAED oppure può utilizzare un *nodo concentratore*, dotato di connessione ad internet, che si occupa di raccogliere tutte le informazioni presenti nelle sottoreti e di inviarle al CAED (Fig. 6.1.3).

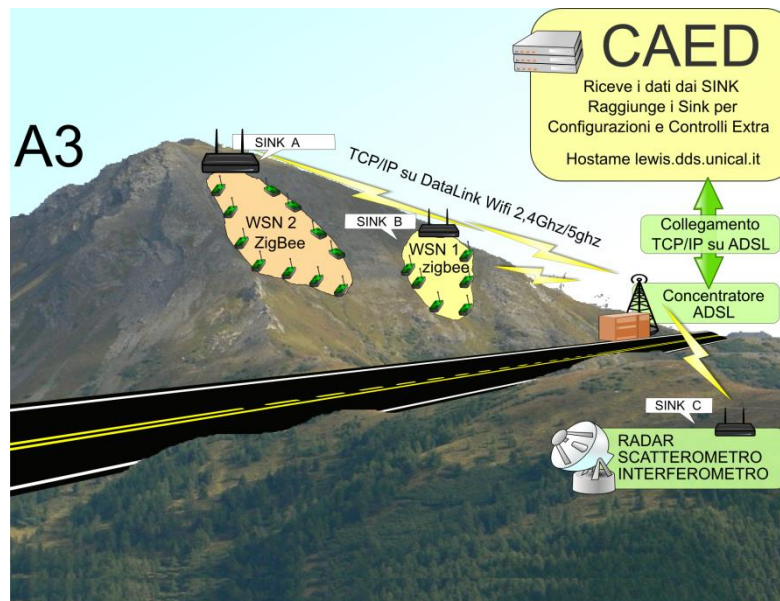


Fig. 6.1.3 – Rappresentazione schematica delle modalità di invio dati su un tronco di autostrada ricoperto da più reti WSN.

Secondo questa rappresentazione, lo schema di collegamento tra la rete di sensori (WSN) ed il CAED può essere descritto attraverso due segmenti distinti:

1. Segmento A: dai sink al concentratore adsl;

2. Segmento B: dal concentratore adsl al CAED.

Lo studio dei due segmenti, così come per le reti WSN descritte nei paragrafi precedenti, non rientra specificatamente nei compiti del CAED, ma è stato esaminato per conoscere i pregi ed i difetti di tutte le possibili combinazioni di comunicazione in ingresso e fornire alle varie WP interessati, spunti per la progettazione dell'infrastruttura di rete.

Nel caso del **segmento A**, può essere utilizzato:

- cablaggio fisico o RF;
- dataLink Wifi 2.4GHz 802.11ag o Mesh 5GHz 802.11ag;
- trasporto TCP/IPv4.

I dispositivi che possono essere utilizzati in questo segmento sono:

- sink multiprotocollo (ZigBee, Wifi 2.4/5GHz, GPRS);
- eventuale amplificatore di segnale RF;
- dispositivo AP/Bridge Wifi 2.4/5GHz con Ethernet 1000Mbps.

Il **segmento B**, invece, comunica direttamente con il CAED e pertanto si consiglia di utilizzare i seguenti dispositivi:

- sink multiprotocollo Wifi 2.4/5GHz (AP/MESH olsr) 802.11a-b-g;
- ADSL/SHDSL "Cisco 2951 Integrated Services Router" o "Cisco 1900 Router" o "Cisco1841-SHDSL-V3 " o equivalente.

Le alternative richiedono attente valutazioni sui costi/opportunità e sulla possibilità di utilizzare sui dispositivi moduli di espansione per integrare nuove interfacce per eventuali link di backup.

In merito al cablaggio fisico del Router ADSL/SHDSL è preferibile adottare una linea sincrona SHDSL per garantire la stessa capacità sia in download che in upload, affinché i dati in uscita dai sinks verso il CAED (Upload) abbiano a disposizione una banda adeguata. In merito al dimensionamento della banda è necessario considerare il data-rate delle WSN ed il livello di attenuazione che la società di telecomunicazioni può garantire.

Come alternativa al concentratore ADSL è possibile adottare un concentratore satellitare bidirezionale. Questa tecnologia è sicuramente una risposta alle difficoltà, o ai costi elevati, che si possono riscontrare in zone che ancora non sono coperte dall'ADSL terrestre. La precedente tecnologia satellitare unidirezionale presentava dei problemi, quali l'esigenza di possedere comunque un collegamento telefonico. Questi solo di recente sono stati risolti con l'avvento delle comunicazioni via satellite bidirezionale. Il tipo di collegamento satellitare

bidirezionale a internet consente di avere una connessione a banda larga ovunque con prestazioni più che buone ad un costo di abbonamento “minimo”, senza che, per il suo funzionamento, sia necessario avere una linea telefonica. Inoltre, grazie alle sue particolari caratteristiche non richiede alcuna specifica autorizzazione per essere installato. Per ottenere un collegamento satellitare bidirezionale bisogna stipulare un contratto con una delle aziende operanti in questo settore. A questo segue per necessità la strumentazione minima, ovvero il modem offerto da contratto, l'antenna satellitare parabolica, una bussola e un inclinometro per ottenere il corretto puntamento della parabola.

Le soluzioni proposte riguardano esclusivamente il caso in cui più reti comunicano con il CAED attraverso un nodo concentratore. Nel caso, invece, ogni sink di ciascuna WSN, voglia essere indipendente dalle altre reti è necessario assicurargli una connessione stabile e persistente attraverso un collegamento GPRS, ADSL/SHDSL ovvero satellitare, previa valutazione dei costi e delle coperture della rete sui tratti di studio. Inoltre è necessario valutare preventivamente la quantità di dati trasportata da ogni WSN, dipendente dalla tipologia di sensori installata (i sensori areali poiché ricoprono una porzione di territorio da investigare ampia trasportano maggiori pacchetti rispetto ai sensori puntuali).

Le informazioni acquisite durante questa attività elementare sono state utili al CAED, per vagliare tutte le possibili modalità di configurazione e comunicazione delle WSN e fornire utili informazioni alla progettazione delle sottoreti, nel caso ci sia la possibilità di utilizzare un nodo concentratore lungo i tratti di sperimentazione.

Modalità connessione e scambio dati

La presente attività definisce il protocollo applicativo realizzato dal CAED per ricevere i dati registrati dalle WSN ed effettuare delle operazioni di programmabilità da remoto delle reti, nel caso le condizioni di operatività lo richiedano. I dispositivi che compongono le reti WSN, all'interno del progetto, per come è stato già evidenziato nel par. 6.1.2, sono dei microprocessori dotati di una limitata capacità hardware. I partner, sviluppatori di hardware, durante i primi incontri, hanno fatto esplicita richiesta di una modalità di immissione dei dati all'interno del database che risiede al CAED, preferibilmente semplice. Inizialmente era stato proposto di installare, sul sink che comunica con la base dati, un client sql per l'inserimento diretto dei messaggi di ciascun nodo su una specifica tabella; tale soluzione, in prima istanza, era risultata la più idonea poiché avrebbe soddisfatto i requisiti di adeguatezza, affidabilità, standardizzazione, autenticità, stabilità e sicurezza. Le limitate capacità dei nodi coordinatori, però, non hanno permesso di implementare questa strategia.

Per permettere a tutti i nodi coordinatori di comunicare con il CAED, indipendentemente dalle specifiche architetture, è stato deciso di realizzare un protocollo di comunicazione standard con semplici pacchetti di scambio, che al loro interno contengono tutte le informazioni registrate dai sensori. Il servizio di acquisizione dei dati è stato denominato AqSERV, le cui caratteristiche sono descritte nel paragrafo successivo.

6.1.4.A SERVIZIO DI ACQUISIZIONE DEI DATI: AQSERV

AqServ è il servizio di acquisizione dati progettato e sviluppato dalla OR 6. Il suo impiego è quello di ricevere i dati dai Sink (nodi concentratori) delle WSN e di mediare tra le reti WSN e la base dati LewisDB, al fine di semplificare le sessioni di salvataggio dati e valutarne l'autenticità.

Caratteristiche tecniche

AqServ adopera una Socket (Stream socket) TCP/IP raggiungibile sulle porte 812/1,2,3,4,5 (ciascuna porta è specifica dei vari progettisti delle WSN che compongono il progetto). Il flusso dei dati non ha incapsulamento, i dati viaggiano in chiaro a differenza delle informazioni di sicurezza, questi viaggiano con un sistema di offuscamento e criptazione in seguito esposta. Questo aspetto è relativo alla sicurezza e all'autorevolezza della sessione di salvataggio.

Altre misure di sicurezza abbracciano gli aspetti di operatività ed efficienza del servizio. Di fatto AqServ è un servizio di rete realizzato per livellare le condizioni di congestionamento, di sovraccarico o di attesa, tutti tipici aspetti relativi ai servizi Internet. Il software del CAED, data la delicatezza dell'Early Warning, tende a comprendere e minimizzare tutte le condizioni di disservizio, malfunzionamento e rallentamento (collo di bottiglia) che potrebbero avere luogo nella fase di produzione.

In risposta a questi scenari AqServ introduce il design pattern informatico denominato Reattore (Reactor) con lo scopo di realizzare un ciclo (loop) non bloccante. Attraverso l'implementazione del Reactor in un contesto di servizio Internet possiamo gestire N connessioni in ingresso, senza che alcune di queste pregiudichino o rallentino eventuali altre concorrenti. AqServ, in altre parole, consente la gestione degli eventi, per ogni richiesta di servizio effettuata contemporaneamente ad altre (connessioni di rete), in modalità asincrona secondo la modalità di recupero "Polling" delle informazioni, al fine di gestire i dati in base alla loro disponibilità ed evitare inopportune attese.

AqServ può gestire simultaneamente innumerevoli connessioni (il limite è determinato dalle caratteristiche della piattaforma Hardware e dal sistema operativo su cui opera il servizio). Nel caso in cui una sessione producesse un errore interno, la cosiddetta eccezione inaspettata, il resto delle sessioni non sarebbe comunque compromesso (assenza di crash del core).

Protocollo di comunicazione e livello applicativo

Il protocollo di comunicazione adottato presenta delle modalità di iterazione fortemente Human-readable, pertanto le fasi di una sessione di AqServ sono illustrate utilizzando direttamente gli screen shots di una sessione tipica. AqServ è un servizio TCP/IP e ogni immissione dei dati deve essere seguita da 2 bytes che identificano il return word, ovvero '\r\n', corrispondenti ai bytes di carriage return (0xd) ed end of line (0xa).

Come illustrato nelle prossime figure AqServ produce due tipi di informazioni in OUTPUT, che possiamo distinguere dalla stringa che le precedono:

- **OUT**: rappresentano le risposte successive alle acquisizioni dei dati;
- **IN**: rappresentano i messaggi di acquisizione che AqServ ha ricevuto dal sink per come gli sono stati forniti, ai quali segue l'elaborazione vera e propria. Queste informazioni sono necessarie esclusivamente ai fini di debug.

Ogni client deve essere registrato preventivamente nel sistema Lewis, per questo i partner coinvolti hanno l'obbligo di compilare e consegnare il modello MRD013 (modello di registrazione dei dispositivi, discusso nel paragrafo 6.1.5).

A seguito della consegna del modello MRD013 lo staff tecnico OR6 provvede a consegnare al referente tecnico le seguenti informazioni:

- Numero identificazione dispositivo, di tipo intero;
- Pre-shared Key, 16bytes.

Queste verranno configurate nel client, ogni dispositivo ha un identificativo numerico univoco e una chiave PSK univoca.

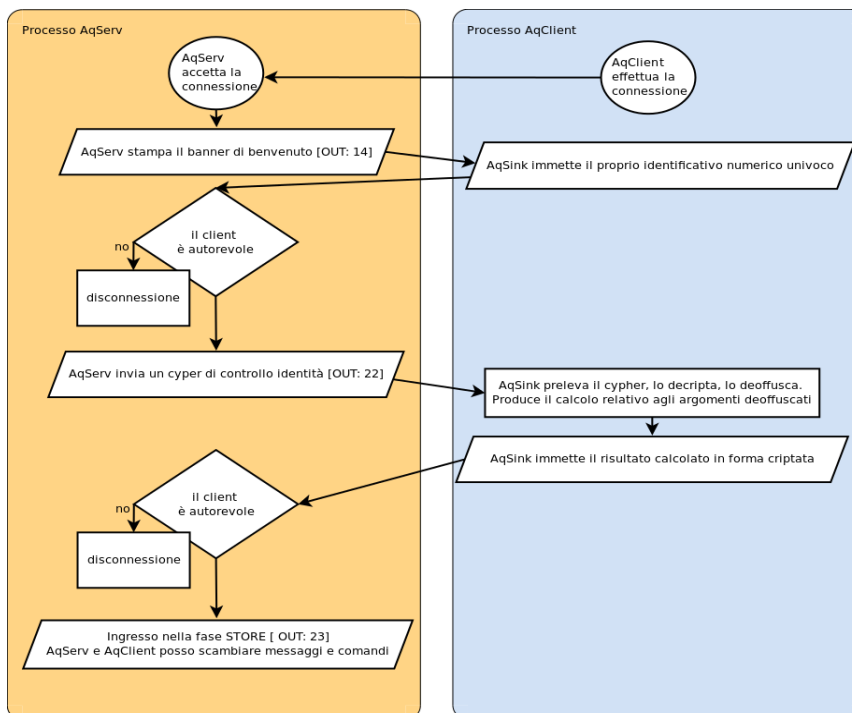


Fig. 6.1.4 – Diagramma aggregato che descrive la fase di autenticazione.

Le fasi di autenticazione client/server sono rappresentate in fig. 6.1.4:

1. Il client (nel seguito denominato AqClient) si connette ad AqServ, tramite la specifica porta;
2. AqServ stampa il banner di benvenuto e chiede l'identificativo univoco del sink;
3. AqClient immette l'identificativo univoco:
 - a. se è registrato all'interno del database del CAED, AqServ invia un cypher per il controllo dell'identità;
 - b. se l'identificativo univoco è sbagliato, AqServ disconnette AqClient dalla connessione;
4. Nel caso di risposta esatta nel punto 3 (a), AqClient preleva il cypher, lo decripta, lo deoffusca ed esegue il calcolo della domanda proposta;
5. AqClient invia il risultato, in forma criptata;
6. AqServ verifica che il risultato immesso sia corretto:
 - a. la risposta è corretta, AqServ entra nella fase di STORE e possono essere scambiati i messaggi tra client e server;
 - b. la risposta è sbagliata, AqServ disconnette il AqClient dalla connessione.

```
OUT: 7 - Welcome to CAED - today it's 2012-07-23 15:05:54.977004
OUT: 0 - write \q for quit

OUT: 14 - Who are you? Prompt your id number please:
```

Fig. 6.1.5 - La sessione con AqServ è stata avviata, il servizio saluta il sink e chiede il numero di identificazione.

```
OUT: 21 - Welcome EWARNET

OUT: 22 - I have a question for you: bIXN0XfLzU5bAD1/hFsPhLrcU2vv+6Mcn57gGcsrDTo=
iDee17BEPd02m40qBnZVww==
```

Fig. 6.1.6 – Dopo aver richiesto e ricevuto l'identificativo univoco di autenticazione, AqServ fornisce una frase criptata al client. Il client la decripta e ne estrae il contenuto. La frase cambia ad ogni connessione.

L'immissione della frase segreta da parte di AqServ consente al Client di estrarre il contenuto offuscato, elaborarlo e fornire una risposta; questa serve per effettuare un confronto numerico e valutare l'autenticità della sessione. La frase

segreta viene comunicata con un semplice metodo di offuscamento, al fine di implementare un livello di sicurezza anti-sabotaggio minimale ma facilmente estendibile. Nella fig. 6.1.5 è visibile la risposta del Client, ovvero:

```
iDee17BEPdO2m40qBnZVww==
```

Questa altro non è che il risultato, criptato, della elaborazione. Il sistema di criptazione adottato possiede alcuni attributi. I client, per eseguire una autenticazione valida devono rispettare scrupolosamente le seguenti caratteristiche:

- AES-128bit (rijndael-128)
- modo CBC
- vettore IV statico pari a:
- `uint8_t iv = { 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08, 0x08 };`

Il sistema di padding adottato rispecchia uno standard Microsoft, ovvero adopera una codifica PKCS7; questa scelta rispecchia la volontà di facilitare l'integrazione con gli altri staff.

I passi procedurali per compiere adeguatamente la fase di autenticazione sono:

1. AqServ fornisce la frase segreta:
`bIXN0XfLzU5bAD1/hFsPhLrcU2vv+6Mcn57gGcsrDTo=`
2. AqClient legge la frase e la decifra in:
`so_ 7.28 (94 5.01`
3. AqClient estrae i numeri dalla frase e li somma:
 $7.28 + 5.01 = 12.29$
4. AqClient fornisce in forma criptata il risultato codificato in base64 (12.29):
`iDee17BEPdO2m40qBnZVww==`

Il metodo di offuscamento introduce 3 bytes di dati random (so_) prima del primo numero, a seguito uno spazio (0x20) e il primo numero (7.28). A seguire un altro spazio, 3 bytes di dati random, un altro spazio e l'ultimo numero (5.01). Le modalità di estrazione dei numeri possono avere innumerevoli strategie e si lascia l'implementazione del metodo agli sviluppatori del client (split basato su carattere, espressione regolare, ...).

Invio messaggi

Una volta effettuata la fase di autenticazione, AqClient può iniziare ad inviare i dati ad AqServ. Per effettuare una standardizzazione i dati sono inviati nel formato JSON, ovvero in formato hash-table (o dizionario).

```
{'hops':'1,2,3,4',  
'sink_insert_datetime':'2013-03-12 14:44:50',  
'message':'[0013A200408C9D03 2013-03-06 12:09:52 #sample] -x:26,y:6,z:985 -  
int_temp:26.2 -bat:65\n ;}'
```

Fig. 6.1.7 – Struttura di un messaggio in formato JSON.

Di questo esempio solo il campo “*message*” è indispensabile ai fini dell'accettazione del dato da parte di AqServ, gli altri campi sono opzionali. Pertanto un invio valido, con profilo minimo, può essere formattato come segue:

```
{'message':'[0013A200408C9D03 2013-03-06 12:09:52 #sample] -  
x:26,y:6,z:985 -int_temp:26.2 -bat:65\n ;}'
```

Gli altri campi, seppure opzionali, consentono un maggiore livello di dettaglio sul routing e sulla data di presa in consegna dei messaggi inviati dai dispositivi da parte del coordinatore. Il messaggio, strutturato come in fig. 6.1.8, inviato ad AqSERV è sottoposto ad una serie di procedure di validazione, discusse in dettaglio nel paragrafo 6.2.1.

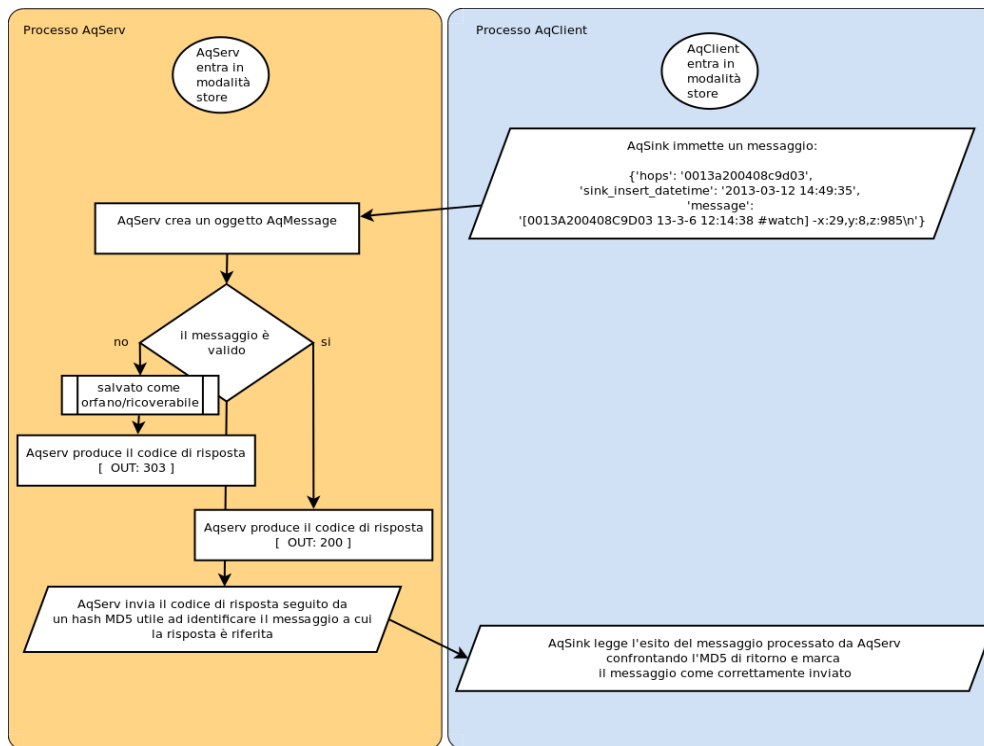


Fig. 6.1.8: La sessione di salvataggio messaggi inviati da parte di AqClient verso AqServ.

Nel caso il messaggio inviato ha superato tutte le fasi di validazione (Fig.6.1.9), AqServ risponde con un codice 200 associando un argomento di 16 bytes, rappresentato in fig. 6.1.9 con formato esadecimale (es. "4479dde17f56ccdc73b190d9deb82ca3"); questo argomento altro non è che l'hash MD5 del messaggio inviato.

```
Server sent 1 possible commands - [['200', ' Saved [43523fcd9f4584d0721f367e66215105]\r' ]]
['43523fcd9f4584d0721f367e66215105'] marked as sent
```

Fig. 6.1.9: Schermata relativa al processo Client AqSink. Il Client riceve la risposta di avvenuto salvataggio [200], preleva l'hash MD5 di controllo integrità e marca il messaggio come correttamente inviato

L'acronimo MD5 (Message Digest algorithm 5) indica un algoritmo crittografico di hashing realizzato da Ronald Rivest nel 1991 e standardizzato con la RFC 1321. Questo tipo di codifica prende in input una stringa di lunghezza arbitraria e ne produce in output un'altra a 128 bit. La codifica avviene molto velocemente e l'output (noto anche come "MD5 Checksum" o "MD5 Hash") restituito è tale per

cui è altamente improbabile ottenere con due diverse stringhe in input uno stesso valore hash in output.

Questo metodo consente al sink, previo confronto degli HASH MD5, di capire a quale messaggio è riferita la conferma di salvataggio da parte di AqServ. Questo metodo è indispensabile in caso di invii differiti per il quale non è possibile risalire alla corrispondenza invio-conferma.

Message Aggregation

Per ottimizzare il processo di inserimento di numerosi messaggi in un'unica sessione di transizione è stata realizzata un'estensione alla struttura dei messaggi, analizzata in fig. 6.1.8. Tale estensione, denominata *message aggregation*, ha una struttura simile a quella per gli invii singoli; a differenza di quest'ultimo dove il messaggio è identificato dal campo 'message' è la voce 'messages' ad attivare la message aggregation e includere una lista di messaggi al suo interno.

```
{'messages':  
  [{ 'message': '[TDgroup_meteo_1 14-02-03 12:56:34 #status]  
bAABAgMEBVLvkPUAQA==;'},  
  { 'message': '[TDgroup_meteo_2 14-03-25 16:43:11 #sample] -  
whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;'},  
  { 'message': '[TDgroup_meteo_2 14-02-25 16:43:06 #sample] -  
whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;'},  
  { 'message': '[TDgroup_meteo_2 14-02-25 16:43:07 #sample] -  
whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;'},  
  { 'message': '[TDgroup_meteo_2 14-02-25 16:43:08 #sample] -  
whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;'},  
  { 'message': '[TDgroup_meteo_2 14-02-25 16:43:09 #sample] -  
whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;'},
```

```
{'message': '[TDgroup_meteo_2 14-02-25 16:43:10 #sample] -  
whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;']  
}
```

Fig. 6.1.10 – Aggregazione di più messaggi per utilizzare il servizio esposto da AqServ per l'inserimento contemporaneo di un numero di acquisizioni maggiore ad uno.

Nel caso di utilizzo della message aggregation, a seguito di AqClient dell'inserimento del messaggio, AqServ risponde con un codice [78] per notificare al client che tale estensione è stata attivata. Il codice finale di tutta la transazione è 27 che indica al client un riepilogo delle procedure dei singoli messaggi.

```
{'messages': [{'message': '[TDgroup_meteo_1 14-02-03 12:56:21 #status]  
bAABAgMEBVLvkPUAQA==;'},{'message': '[TDgroup_meteo_2 14-01-25  
16:43:27 #sample] -whr_rad:+0.097459 -  
whr_temp:+8.575839,whr_humi:+67.518653,whr_dewpoint:+2.898416;'}]}
```

OUT: 78 - Message aggregation acknowledged [
7f7fd69373ffdddba034426210a968c4]

OUT: 306 - Message [a09585cb2fc61514551815d5f93dcb94] saved as status

OUT: 200 - Saved [c2eaa60a3df4c0ab54995a669569a947]

OUT: 27 - 0 2/2 aggregated messages was succesfull saved

In successione: il codice 78 indica la message aggregation, il primo messaggio salvato che stampa codice 306 perché il tipo di messaggio - #status – non è utile ai fini del monitoraggio ambientale, dopodiché viene stampato un codice 200 per il messaggio salvato e il codice 27 che indica l'esito generale. Se anche solo un messaggio dovesse fallire il codice 27 tornerebbe 1 e la WSN potrebbe reinviare singolarmente i messaggi o fare il parse dei codici di risposta dei singoli messaggi per rilevare quali fra questi, nello specifico, siano falliti lungo la prima fase di validazione.

Acquisizione comandi CAED to WSN

La programmabilità delle WSN da parte del CAED (per soddisfare uno dei requisiti analizzati nel paragrafo 6.1.2) è stata garantita ed implementata secondo il protocollo di messaggi *server to client*, realizzato in AqServ.

AqServ può inviare comandi remoti al fine di riconfigurare o interagire con le reti di monitoraggio ambientale. I comandi possono richiedere una specifica acquisizione, possono modificare l'intervallo di acquisizione di un dato sensore, possono eseguire comandi arbitrari sui dispositivi remoti gestiti dal nodo coordinatore di riferimento, quali controlli sugli integrati, modifiche dei parametri di rete, modifica dell'orario o qualsiasi comando implementato nel rispettivo firmware del dispositivo.

Il formato di un comando inviato dal CAED ad un nodo coordinatore, per mezzo di AqServ, presenta la seguente struttura:

```
{'id':1, 'destination':'0013A200408C9D03', 'command': 14, 'arguments':[ '2013-03-17 12:05:17', 'UTC' ] }
```

Questo esempio indica al coordinatore, che ha precedentemente stabilito la connessione con AqServ, di inviare al dispositivo '0013A200408C9D03' il comando 14 con gli argomenti sopra esposti. Ogni dispositivo, operante nella WSN remota, ha un set di comandi e argomenti possibili, questi devono essere classificati preventivamente presso il CAED. Il campo id indica l'identificativo univoco del comando inviato all'interno del sistema Lewis.

```
OUT: 205 - { 'cmd_id': 1, 'destination':'0013A200408C9D03', 'command': '14', 'arguments':[ '2013-03-17 12:05:17', 'UTC' ] }
```

Fig. 6.1.11: Comando inviato da AqServ verso il client remoto.

A seguito dell'acquisizione del messaggio il client risponde con la stringa di controllo MD5, al fine di validare l'autenticità del dato acquisito.

```
{ 'id': 1, 'md5':'817741e67931b173ce6986801553888b' }
```

L'hash MD5 è valutato sull'intera stringa JSON:

```
MD5({ 'id': 1, 'destination':'0013A200408C9D03', 'command': 14, 'arguments':[ '2013-03-17 12:05:17', 'UTC' ] })
```

la precedente funzione ritorna la seguente serie di bytes esadecimali:

```
817741e67931b173ce6986801553888b
```

Il coordinatore, esponendo l'id accoppiato con l'hash md5 chiede il controllo ad AqServ che il dato acquisito sia autorevole. Nel seguente esempio AqServ conferma che il comando recepito dal client è stato correttamente acquisito e

aggiorna lo stato di invio del comando all'interno della propria base dati. Se il valore md5 inviato dal client ad AqServ non rispecchia quello autentico, la risposta che ne segue è un codice di fallito recepimento del comando.

```
AqServ response:
IN: 186 - { 'cmd_id': 1, 'md5':'817741e67931b173ce6986801553888b' } (from 160.97.12.228)

OUT: 207 - 817741e67931b173ce6986801553888b

Server sent 1 possible commands - [['207', '817741e67931b173ce6986801553888b\r']]
```

Fig. 6.1.12: AqServ risponde con un 186 (di debug) e con un codice 207, che conferma l'autenticità del dato acquisito dal client. Il codice di debug 186 indica cosa ha immesso il client, il codice 207 è la risposta, in questo caso positiva, del recepimento del comando.

```
AqServ response:
IN: 186 - { 'cmd_id': 1, 'md5':'817741e77931b173ce6986801553888b' } (from 160.97.12.228)

OUT: 209 - Error in remote command acquisition.

Server sent 1 possible commands - [['209', 'Error in remote command acquisition.\r']]
```

Fig. 6.1.13: Il codice 209 indica al client che il controllo sul comando acquisito è fallito, pertanto verrà inviato nuovamente fino a N tentativi, prima di produrre un messaggio di attenzione per lo staff del CAED

Se il comando acquisito dal client ha un esito positivo e produce una risposta da inviare al CAED è possibile inviare tale risposta secondo il seguente formato:

```
{'hops': '1', 'sink_insert_datetime': '2013-03-12 14:44:50', 'message':
'[0013A200408C9D03 13-3-6 12:9:52 #response #r1] -x:26,y:6,z:985 -
int_temp:26.2 -bat:65\n ;}
```

nella intestazione del message è stato configurato il tipo di messaggio “#response”, questo aspetto è a discrezione degli sviluppatori del firmware. Tale livello di dettaglio è opzionale. Il campo #r1, invece, indica al CAED di associare il messaggio come conseguenza del comando precedentemente inviato.

Limitazioni del protocollo

AqServ accetta esclusivamente caratteri ascii, eventuali immissioni di stringhe codificate saranno troncate ed il risultato di tale troncamento è visionabile per mezzo del codice IN: 186. Questo codice è utilizzato esclusivamente per scopi di debug e può essere omesso in fase di produzione.

Un coordinatore prima di riconnettersi ad AqServ deve aver chiuso la connessione precedente, nel caso in cui lo stesso coordinatore, che non ha

chiuso la connessione, si identifichi nuovamente presso AqServ, questo gli negherebbe la sessione con relativa motivazione.

Sicurezza ed Operatività

Il metodo di criptazione adottato da AqServ è AES-128, ovvero criptazione a chiave sincrona con blocchi da 16 bytes. È stata scartata l'opportunità di realizzare una sessione TLS (ssl) per due motivi:

1. avrebbe comportato lo sviluppo di un client TLS lato firmware per i progettisti Hardware delle WSN;
2. avrebbe ridotto la banda disponibile. Operando in UMTS/GPRS questo avrebbe comportato un rallentamento generale, specie nelle condizioni meteorologiche avverse quali le precipitazioni.

La criptazione si avvale di una chiave condivisa generata presso il CAED e fornita ai partner che sviluppano le piattaforme hardware embedded.

Messaggi e codici di risposta

Nella tab. 6.1.3 sono riportati tutti i codici di risposta che AqServ invia ai client durante le sessioni di immissione dei messaggi e di invio/risposta dei comandi remoti per la riprogrammabilità delle WSN.

COD	Descrizione
0	Messaggio di esempio, banner, informazione, descrizione, commento.
7	Messaggio di benvenuto
9	Connessione scaduta per inattività (timeout)
12	Messaggio di logout
14	Richiesta identificativo univoco Sink
17	Identificativo Sink non valido
21	Riconoscimento Sink
22	Immissione password criptata, segue aggiornamento IP dinamico in LewisDB
23	Modalità acquisizione (STORE), si possono immettere le acquisizioni
27	Riepilogo salvataggio Message Aggregation
28	Password aggiornata correttamente in LewisDB
31	Disconnessione per inserimento password errata

186	Invio dati dal client al server, codice di avvenuta ricezione (solo per DEBUG)
200	Salvataggio avvenuto
205	Invio comando CAED to WSN
207	Ricezione comando correttamente avvenuta
209	Errore nella ricezione del comando
301	Messaggio incompleto (nel caso in cui non siano consentite misurazioni assenti nel messaggio)
303	Salvataggio avvenuto come messaggio orfano: la validazione è fallita
305	Salvataggio non avvenuto: duplicato
306	Notifica ricezione tipo messaggio non utile ai fini del monitoraggio
307	Messaggio malformato
309	Duplicazione acquisizione in Message Aggregation (il codice è esposto solo se la message aggregation è in verbose_mode = True)
403	Lo stesso sink risulta aver stabilito una sessione e questa è ancora attiva. Transazione negata, disconnessione.

Tab. 6.1.3 – Codici di risposta di AqServ.

6.1.4.B MESSAGE PROTOCOL

.....

Il messaggio è l'informazione utile contenuta nel trasporto di rete WSN. Come diversi protocolli di incapsulamento presenti in letteratura, il CAED Message Protocol è un protocollo basato sulla composizione dei dati trasportati e non sulla tipologia dei frame inviati in rete. Questa strategia consente di adottare un protocollo di scambio di dati e comandi indipendente dal protocollo di rete adottato, come nel caso di ZigBee o altro, offrendo completa portabilità su sistemi radio diversi. Pertanto il CAED Message Protocol è un protocollo di livello superiore a quello di rete, incapsulato nei dati (payload) trasportati dai frame di rete, e rispecchia le seguenti caratteristiche:

1. Leggibili umanamente, sono stringhe di testo di facile interpretazione;
2. Composti, contengono diverse misure e caratteristiche di sensori differenti;
3. Marcati temporalmente, ogni messaggio ha una marcatura temporale nel formato europeo ISO 8601;
4. Identificabili ad un dispositivo univoco, per distinguerne l'origine.

Il message protocol adottato all'interno del firmware, per poter soddisfare i requisiti sopra elencati è composto da un'intestazione racchiusa tra parentesi quadre in cui è presente l'identificativo di rete (mac address o altro, purché univoco all'interno della rete) del dispositivo, la data e l'ora ed il tipo di messaggio prodotto, a seguire sono presenti i valori registrati dai sensori o eventuali variabili del dispositivo a seguito di una interrogazione. Un esempio di message protocol secondo le regole precedenti è descritto nella fig. 6.1.14.

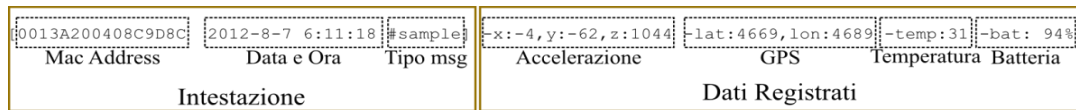


Fig. 6.1.14 - Message protocol adottato per l'invio dei dati dal dispositivo al nodo coordinatore.

Nel caso dei messaggi ZigBEE inviati all'interno di una rete WSN, la struttura utilizzata per trasmettere i pacchetti è descritta in fig. 6.1.15. Analizzando la figura, i dati che devono essere inviati, seguendo le caratteristiche precedenti, sono disposti nella cella DATA, ovvero quella predisposta al trasporto del payload. Ecco un esempio di messaggio:

[0013A200408C9CF9 13-3-4 12:38:22 #watch] -x:16,y:-93,z:1044;

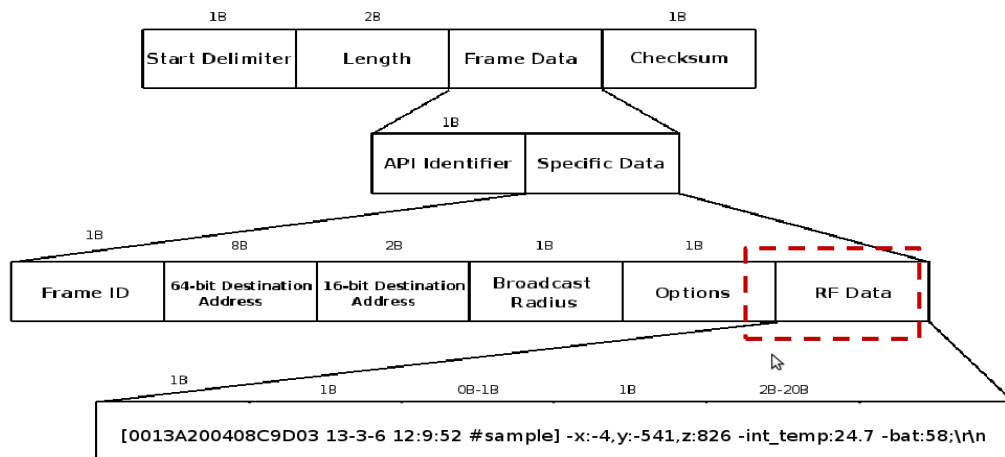


Fig. 6.1.15 - Struttura del pacchetto ZigBEE inviato da un dispositivo (in rosso la collocazione all'interno del pacchetto dei dati inviati dal dispositivo).

Il frame dei Dati Registrati è composto da più elementi, ogni elemento è rappresentato da un carattere iniziale “-” seguito dal nome della variabile e dal valore di risposta; il valore è separato dal nome della variabile per mezzo del carattere “:”. Nel caso in cui un sensore produca due o più variabili, ognuna di esse deve essere separata per mezzo del simbolo “,”, come nel caso dell'accelerometro per esempio:

-x:-4,y:-62,z:1044

- Carattere del gruppo
- : Carattere di assegnazione
variabile/valore
- , Carattere di separazione variabili
gruppo

Tipi di messaggi

I messaggi che i dispositivi inviano al CAED sono di diversa natura e possono contenere informazioni diverse. La distinzione dei diversi messaggi all'interno di un pacchetto ZigBEE, seguendo le indicazioni del message protocol adottato, è contenuta all'interno dell'intestazione del messaggio (i dati racchiusi nelle parentesi quadre: [0013A200408C9D03 13-3-6 12:9:52 #**sample**]). I messaggi che il dispositivo può inviare sono al massimo di sei tipologie, ognuno dei quali si differenzia dal tipo di variabile contenuta nell'intestazione, questa può essere numerica o letterale ed è preceduta dal carattere '#'. Nella tab. 6.1.4 sono definite le 6 variabili che possono caratterizzare un singolo messaggio.

Letterale	Numerico
#ack	#1
#response	#2
#sample	#3
#warning	#4
#watch	#5
#status	#6

Tab. 6.1.4 - Tipi di messaggi inviati del message protocol del firmware.

- **#ack:** definisce un messaggio di conferma/recepimento comando di riconfigurazione o interrogazione (CAED to WSN)
[2013-11-13 1:0:34 #ack] comando;
- **#response:** definisce un messaggio di risposta ad una richiesta (CAED to WSN) di invio acquisizione dati
[2013-11-13 1:0:34 #response] risposta;
- **#sample:** definisce un messaggio contenente i dati acquisiti ad un certo intervallo di campionamento
[2013-11-13 1:0:34 #sample] valori campionati;
- **#warning:** definisce un messaggio contenente un avvertimento, quale ad esempio il livello critico della batteria
[2013-11-13 1:0:34 #warning] warning segnalato;
- **#watch:** definisce un messaggio di avviso evento a seguito del superamento della soglia di uno o più sensori sentinella (accelerometro)
[2013-11-13 1:0:34 #watch] valori sensore sentinella;
- **#status:** definisce un messaggio di cambiamento di stato del dispositivo, ad esempio 1 indica l'avvio del dispositivo e l'ingresso in rete
[2013-11-13 1:0:34 #status] 1;

Esempi

boot del dispositivo avvenuto correttamente

[0013A200408C9D8C 2014-1-8 10:22:47 #status] 1

messaggio sentinella su un evento

[0013A200408C9D8C 2014-1-8 10:22:47 #watch] -temp:22.06 -hum:82 -soil_temp:20.09

messaggio di campionamento

[0013A200408C9D8C 2014-1-8 10:22:47 #sample] -x:20,y:36,z:1047 -int_temp:26 -bat:77

Ogni sviluppatore può utilizzare message protocol diversi da quello proposto, basta che siano sempre verificate le specifiche descritte nei punti 1, 2, 3 e 4. In linea generale, quasi tutti si sono adattati a questo modello, con piccole modifiche soprattutto per la netta distinzione delle variabili fisiche registrate dai sensori areali rispetto a quelli puntuali.

Testing piattaforma acquisizione WSN

Lo scopo di questa attività è stato di valutare le prestazioni del servizio di acquisizione e delle procedure di estrazione delle informazioni dai messaggi trasportati. L'attività elementare è stata strutturata in tre fasi: nella prima parte il test è stata effettuato realizzando preliminarmente un simulatore di WSN, in grado di replicare le acquisizioni da parte di un set di sensori; nella seconda parte è stata costruita una stazione meteorologica ed acquisiti ed inviati al CAED i dati in tempo reale; nella terza ed ultima parte, infine, è stato effettuato il test con le reti di sensori che andranno ad essere installati lungo i tratti di sperimentazione.

6.1.5.A SIMULATORE DI RETI WSN

L'idea iniziale per la verifica delle procedure di acquisizione ed estrazione dei dati, nella prima parte del progetto in cui era ancora in corso lo sviluppo dei firmware dei progetti delle reti di sensori, è stata di realizzare una piattaforma ad-hoc in grado di simulare il comportamento dei nodi (motes) della WSN e l'acquisizione da parte del nodo concentratore ed, infine, il successivo salvataggio di questi sulla base dati del sistema Lewis. La piattaforma, sviluppata interamente nel linguaggio di programmazione python, è stata denominata **pyMOTESim**. La simulazione consente di ricostruire una rete WSN configurata in LEWIS; per ogni sensore configurato nella rete vengono prodotti dei valori relativi a delle serie di andamento, queste ultime sono generate secondo alcune funzioni matematiche registrate nel sistema. I dati simulati dalla piattaforma, pertanto, seguono un andamento pseudo-casuale che dipende dalla funzione matematica applicata al sensore.

Ogni dispositivo produce in output un messaggio composto da più variabili, quanti sono i suoi sensori, alla frequenza stabilita di acquisizione (wsn-scheduling), mentre il salvataggio nella base dati avviene anch'esso ad intervalli prestabiliti che possono essere differenti da quelli di acquisizione.

In principio pyMOTESim era nato come una utility di supporto al progetto, pertanto il suo funzionamento era autonomo e i motes venivano generati sulla

base di alcune classi che ne descrivevano modello e composizione. A tutt'oggi, pyMOTESim è stato integrato nel progetto Lewis di conseguenza diverse opzioni di inizializzazione, quali le credenziali di accesso, le tabelle e le colonne di destinazione, sono implicitamente fornite da Lewis, essendo pyMOTESim un modulo del progetto principale.

Il suo scopo principale è quello di poter testare, successivamente alla configurazione di una nuova rete con nuovi dispositivi e nuovi sensori, una WSN senza che questa venga dispiegata realmente in produzione. Questa strategia ha consentito allo staff del CAED di controllare le coerenze di validazione, estrazione e classificazione delle informazioni, con le regole di trattamento dei dati di LEWIS, prima che le reti vengano installate realmente.

Un altro obiettivo di pyMOTESim è offrire un metodo solido e flessibile per la produzione di dati esemplificativi, coerenti con le ipotetiche realtà del futuro, ovvero quando le reti WSN saranno in produzione.

Nel seguito del paragrafo sono discusse tutte le funzionalità del simulatore, attraverso degli screenshot. In breve, per inizializzare il simulatore è necessario definire la configurazione della rete: sink, mote, sensori e a ciascun sensore associare un comportamento pseudo-causale (già presente nella base dati oppure crearne uno nuovo, per casi specifici); a seguito della inizializzazione è possibile eseguire il simulatore impostando l'intervallo di campionamento dei sensori e quello di inserimento dei messaggi all'interno della base dati.

pyMOTESim: Simulatore Wireless Sensor Network

Esempi d'uso

Motesim consente tre tipi di esecuzione e diversi parametri operativi. Se eseguito senza parametri pyMOTESim produce in output il suo manuale in linea e alcuni suggerimenti.

```
<codice>
difsuolo@difsuolo-
linux:~/Dev/LewisGIS/LewisGIS/simulatore/pyMOTESim$ ./motesim

usage: motesim [-h]
               [-run
               {test,store,examples,list_sinks,list_motes,list_sensors,list_behaviour}]
```

```

        [--sink SINK_ID] [--store-interval INTERVAL]
        [--wsn-scheduling SCHEDULING] [--behave-map
BEHAVE_MAP]
        [--version]

Simulate MOTE Sensor Acquisitions and Store them to a DataBase

optional arguments:
-h, --help      show this help message and exit
-run
{test,store,examples,list_sinks,list_motes,list_sensors,list_behaviour}
        Test Produce only a simulation without store datas
        into DB.
--sink SINK_ID  Emulate WSN motes handled by a specific Sink
--store-interval INTERVAL
                Seconds between one Sink Data Send and another
--wsn-scheduling SCHEDULING
                Frequency of Motes Sensor Data Measurements
--behave-map BEHAVE_MAP
                --behave-map mote_id,x:behave_id,y:behave_id
--version      show program's version number and exit

This may help: motesim -run examples
</codice>

```

Di seguito le descrizioni delle modalità di esecuzione (-run) con un esempio del loro output.

- “*list_sink*”: lista dei Sink registrati in Lewis e disponibili per la simulazione, la loro rappresentazione include l'identificativo univoco numerico, seguito dal suo nome e dalla descrizione;

```

<codice>
$ ./motesim -run list_sinks

Active Sinks per Wireless Sensor Networks

ID Number. Wsn name, Sink name

```

```
2. A16_idro, Sink_A16
3. A20_gi, Sink_A20
1. WSN Rogliano test 1, Sink_Rogliano_Test_1
```

</codice>

- “*list_motes*”: lista dei motes che fanno riferimento al sink selezionato;

<codice>

```
$. /motesim -run list_motes --sink 1
```

Active Motes per Sinks

ID number. Wsn name, Sink name, Mote Name

```
4. WSN Rogliano test 1, Sink_Rogliano_Test_1, Libelum_Waspmote
00163e6216d4
3. WSN Rogliano test 1, Sink_Rogliano_Test_1, Libelum_Waspmote
00163e5c9001
2. WSN Rogliano test 1, Sink_Rogliano_Test_1, Libelum_Waspmote
00163e374b82
1. WSN Rogliano test 1, Sink_Rogliano_Test_1, Libelum_Waspmote
00163e45b67c
```

</codice>

- “*list_sensors*”: lista dei sensori per ogni dispositivo. L'output dell'esempio è troncato per limiti di spazio;

<codice>

```
./motesim -run list_sensors --sink 1
```

Active Motes per Sinks

ID number. Wsn name, Sink name, Mote Name

```
4. WSN Rogliano test 1, Sink_Rogliano_Test_1, Libelum_Waspmote
00163e6216d4
```

Sensors on board:

```
ID_number. name, sensor_vars
```


1. Accelerometro_Waspmote, [y,x,z]
2. Termometro_WaspMote, [temp]
3. LivelloBatteria_Waspmote, [bat]

[.... altri troncati ...]

</codice>

- “*list_behaviour*”: lista delle funzioni che possono caratterizzare le serie di acquisizioni prodotte da ogni sensore. Per applicare una funzione comportamentale ad un sensore bisogna specificare l'identificativo del mote e per ogni variabile che si desidera popolare bisogna associare l'identificativo di un behaviour. Esempio:

Esempio: --behave-map 1,x:14 --behave-map 3,y:16,x:14,z:14

<codice>

```
./motesim --run list_behaviour
```

Behaviours applicable for every Sensor value

ID number. Name, description

1. decrescente,
2. stazionario_33,
14. Random, durante le simulazioni la distribuzione random varia di volta in volta
6. abs,
15. Calm,
9. sen_1,
8. abs_1,
16. Random basso accelerometro,

</codice>

L'esempio precedente applica al mote 1, alla variabile x, la funzione Random; anche al mote 3 è stata associata alle variabili x e z la funzione Random, mentre alla sua variabile y è stato associato un andamento Random con scarto

quadratico (funzione Random basso accelerometro). Tutte le altre variabili non contemplate dalla opzione behave-map eseguiranno un andamento stazionario.

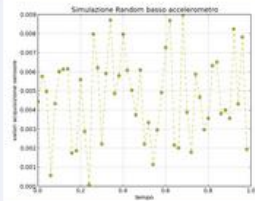
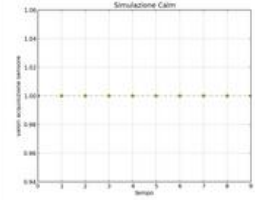
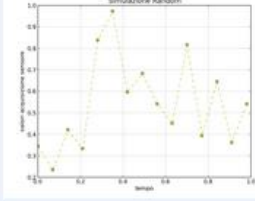
Nome	Funzione	Durata	Step	Grafico Comportamento
Random basso accelerometro	FunzioneRandomRangeBassoAccelerometro	1,0	0.02	
Calm	FunzioneCalm	10,0	1	
Random	FunzioneRandom	1,0	0.07	

Fig. 6.1.16 – Tre diverse funzioni generatrici registrate in Lewis; ogni comportamento necessita di una funzione generatrice e due argomenti che sono la durata temporale e lo step.

Le funzioni elencate nella lista dei comportamenti di pyMOTESim vengono classificate nella base dati di Lewis, alla voce *Simulatore* → *Comportamento variabili Sensori*. In questa sezione l'utente può registrare nuovi comportamenti e ottenere una anteprima di quelle correnti (Fig. 6.1.16).

- “*example*”: questa opzione offre in output degli esempi di uso tipico di pyMOTESim. L'opzione test e l'opzione store hanno una sintassi simile, a differenziarli è il comportamento del sink simulato. Con l'opzione test, infatti, i messaggi vengono soltanto stampati a schermo (standard output) mentre solo con l'opzione store viene prodotta una query di salvataggio da parte del sink sul server DataBase.

```
<codice>
./motesim -run examples
```

Es:

```
motesim -run list_sinks
motesim -run list_behaviour
motesim -run list_motes --sink 1
motesim -run list_sensors --sink 1
motesim -run test --sink 1 --store-interval 7 --wsn-scheduling 5
motesim -run test --sink 1 --store-interval 7 --wsn-scheduling 3 --
behave-map 1,x:14
motesim -run store --sink 1 --store-interval 7 --wsn-scheduling 3 --
behave-map 1,x:1
</codice>
```

Esempi di Output

Nell'esempio successivo ogni mote produce il proprio messaggio e il sink, all'intervallo prestabilito, raggruppa tutti i messaggi e li salva nella base dati. Durante l'esecuzione pyMOTESim crea tanti processi paralleli quante i nodi della rete WSN. Inoltre, per ogni messaggio è possibile misurare la dimensione in bytes.

```
<codice>
./motesim -run store --sink 1 --store-interval 330 --wsn-scheduling 300
--behave-map 3,y:16,x:16,z:16 --behave-map 2,y:16,x:16,z:15 --
behave-map 1,y:16,x:15,z:16 --behave-map 4,y:16,x:16,z:16
```

```
Sink_Rogliano_Test_1
```

```
Starting.
```

```
Motes number: 4
```

```
Sink Acquisition Interval set to: 330
```

```
Motes sensor acquisition set to: 300
```

Sensors on site: Accelerometro_Waspmote, Termometro_WaspMote,
LivelloBatteria_Waspmote

[ID:4 2012-10-09 10:42:00]-mac:00163e6216d4 -
x:77.7676780001,y:85.0046176293,z:81.4072287005 -temp:22.6 -
bat:52.0% [130 bytes]

[ID:3 2012-10-09 10:42:00]-mac:00163e5c9001 -
x:82.8481871991,y:84.8617644554,z:74.9529280407 -temp:22.6 -
bat:52.0% [129 bytes]

[ID:1 2012-10-09 10:42:00]-mac:00163e45b67c -
x:81.92,y:79.4774898787,z:83.4729406182 -temp:22.6 -bat:52.0%
[119 bytes]

[ID:2 2012-10-09 10:42:00]-mac:00163e374b82 -
x:84.1457112852,y:86.2329962618,z:81.92 -temp:22.6 -bat:52.0%
[127 bytes]

Stored 4 Messages [513 bytes] into DataBase

[ID:1 2012-10-09 10:47:00]-mac:00163e45b67c -
x:81.92,y:84.3299964821,z:84.2931375961 -temp:22.6 -bat:52.0%
[456 bytes]

</codice>

id tabella serial	acquisizione text	data_inserimento timestamp with time zone	sink_id integer
1	105 [HEADER]-mac:00163e22428a x:91,y:48,z:117 temp:105 bat:16%		1
2	106 [HEADER]-mac:00163e73c58e x:179,y:195,z:129 temp:35 bat:69%		1
3	107 [HEADER]-mac:00163e7785d7 x:175,y:30,z:192 temp:119 bat:78%		1
4	108 [HEADER]-mac:00163e796d7a x:103,y:123,z:78 temp:12 bat:91%		1
5	109 [HEADER]-mac:00163e1a0191 x:0,y:7,z:65 temp:145 bat:92%		1
6	110 [HEADER]-mac:00163e2057cd x:45,y:161,z:73 temp:109 bat:27%		1
7	111 [HEADER]-mac:00163e363805 x:45,y:113,z:187 temp:26 bat:12%		1
8	112 [HEADER]-mac:00163e441470 x:133,y:107,z:176 temp:69 bat:80%		1
9	113 [HEADER]-mac:00163e017dc0 x:140,y:52,z:5 temp:136 bat:86%		1
10	114 [HEADER]-mac:00163e73c58e x:37,y:76,z:65 temp:39 bat:78%		1
11	115 [HEADER]-mac:00163e7785d7 x:128,y:92,z:13 temp:174 bat:74%		1
12	116 [HEADER]-mac:00163e22428a x:15,y:36,z:112 temp:179 bat:97%		1
13	117 [HEADER]-mac:00163e1a0191 x:63,y:83,z:82 temp:144 bat:14%		1
14	118 [HEADER]-mac:00163e796d7a x:102,y:111,z:56 temp:126 bat:20%		1
15	119 [HEADER]-mac:00163e2057cd x:19,y:14,z:83 temp:106 bat:43%		1
16	120 [HEADER]-mac:00163e363805 x:70,y:179,z:58 temp:145 bat:75%		1
17	121 [HEADER]-mac:00163e441470 x:138,y:33,z:92 temp:142 bat:33%		1
18	122 [HEADER]-mac:00163e017dc0 x:91,y:25,z:127 temp:112 bat:32%		1
19	123 [HEADER]-mac:00163e73c58e x:119,y:3,z:184 temp:119 bat:42%		1
20	124 [HEADER]-mac:00163e7785d7 x:91,y:114,z:149 temp:143 bat:48%		1
21	125 [HEADER]-mac:00163e22428a x:155,y:38,z:199 temp:131 bat:90%		1
22	126 [HEADER]-mac:00163e796d7a x:90,y:144,z:21 temp:118 bat:51%		1
23	127 [HEADER]-mac:00163e1a0191 x:75,y:96,z:198 temp:14 bat:30%		1
24	128 [HEADER]-mac:00163e363805 x:161,y:21,z:28 temp:148 bat:24%		1
25	129 [HEADER]-mac:00163e2057cd x:52,y:40,z:52 temp:122 bat:48%		1
26	130 [HEADER]-mac:00163e441470 x:5,y:44,z:25 temp:196 bat:27%		1
27	131 [HEADER]-mac:00163e017dc0 x:44,y:105,z:135 temp:85 bat:31%		1
28	132 [HEADER]-mac:00163e293bcd x:168,y:191,z:158 temp:55 bat:31%		1
29	133 [HEADER]-mac:00163e6eda67 x:25,y:85,z:163 temp:106 bat:40%		1
30	134 [HEADER]-mac:00163e555a3b x:11,x:52,z:0 temp:128 bat:14%		1

Fig. 6.1.17 – Salvataggio dei dati nel DataBase Postgres prelevati dalla rete virtuale di MOTESim.

Nella prima versione del simulatore i messaggi erano inseriti direttamente nel database e sono state realizzate, parallelamente allo sviluppo del simulatore, le procedure di estrazione dei dati trasportati e di validazione. Questo poiché, preliminarmente, come discusso nel paragrafo 6.1.4 il CAED aveva proposto di inserire direttamente le acquisizioni nel DB attraverso delle query insert. Con lo sviluppo di AqSERV, per l'ottimizzazione del processo di acquisizione dei dati dai sistemi embedded, i messaggi generati dal simulatore sono inviati tramite AqClient e verificata tutta la procedura descritta nel paragrafo precedente.

Il simulatore realizzato, non considera in nessun conto, le problematiche di routing all'interno delle WSN poiché di interesse ad altre OR del presente progetto; l'obiettivo principale per la realizzazione è stato di creare uno strumento semplice che ricostruisca le gerarchie delle WSN (sink e mote) ed applichi il message protocol per la formattazione dei messaggi.

6.1.5.B RETE WSN DI LABORATORIO

Il simulatore, per le verifiche delle procedure realizzate dal CAED, è stato utilizzato specificatamente nella prima parte del progetto; nella seconda parte, solamente le acquisizioni reali, con tutte le problematiche del caso, potevano dare risalto ad eventuali incoerenze delle idee progettuali, della struttura del data base ovvero delle acquisizioni e dei comandi remoti, inoltrate tramite AqServ.

Per superare i limiti oggettivi di una rete simulata e testare la filiera procedurale di acquisizione dei dati all'interno dell'applicativo LEWIS, è stata realizzata una WSN di laboratorio composta da dei nodi di primo livello (mote) in cui sono installati un numero predefinito di sensori. I dispositivi, per riprodurre realmente le reti WSN del progetto, comunicano i dati registrati dai sensori ad un nodo coordinatore (sink), che trasporta il contenuto informativo al Centro di Elaborazione Dati.

I nodi di primo livello utilizzati per la piattaforma di sviluppo sono microcontrollori (MCU) della famiglia AVR, prodotti dalla Atmel. Questa tipologia di MCU è stata preferita rispetto ad altre per la popolarità del prodotto più che per le reali prestazioni in termini di processazione e di consumi energetici: è stato possibile realizzare una rete di monitoraggio in tempi relativamente ridotti (circa tre mesi) composta da sensori puntuali con protocollo di comunicazione, verso il nodo coordinatore, di tipo ZigBEE. In particolare, i prodotti che sono stati utilizzati sono stati forniti dalla Libelium (denominati *waspmote*)¹, che opera specificatamente nel settore delle WSN per applicazioni ambientali, smart-city, applicazioni sanitarie ecc.

Specifiche tecniche del WASPMOTE

- Denominazione: WAPMOTE V1.1
- Microcontrollore: ATmega1281
- Frequenza: 8MHz
- SRAM: 8KB
- EEPROM: 4KB
- FLASH: 128KB
- SD Card: 2GB
- Range di Temperatura [-20°C, +65°C]

¹ <http://www.libelium.com/>

Per una descrizione dettagliata dell'architettura del dispositivo, dei sensori montati direttamente sul Waspote (accelerometro, sensore di temperatura interna, ecc.), delle modalità di comunicazione e di trasferimento dei dati e di altre specifiche e procedure si rimanda alla lettura della documentazione ufficiale.

Progettazione firmware

I dispositivi adottati dal CAED si prestano adeguatamente per il monitoraggio di variabili ambientali utilizzando una rete di sensori di tipo wireless (WSN); una WSN ha due funzioni principali: misurazione e comunicazione. Il protocollo utilizzato per la comunicazione all'interno di ciascuna rete di monitoraggio è quello ZigBEE PRO: i waspmote sono dotati di moduli radio DIGI Xbee, con frequenza di 2.4 GHz. L'utilizzo di antenne con protocollo di comunicazione di tipo ZigBEE per il monitoraggio di variabili ambientali consente una strategia di risparmio energetico, i moduli alternano una fase di sleep a quella di wake-up (sleep-cycle) solo durante l'invio dei dati ed eventualmente per la ricezione dei comandi da parte dei nodi di livello superiore (sink). Nel caso del firmware realizzato, dato che si tratta di monitoraggio ambientale per sistemi di early warning, la fase di sleep dei dispositivi è stata abbandonata per garantire un monitoraggio continuo delle variabili di interesse ed una comunicazione full-duplex con il Centro di Elaborazione dei Dati, che può interrogare in qualsiasi momento la rete di monitoraggio. Per eliminare lo sleep intrinseco dei nodi end-device, ogni dispositivo è stato dotato di una antenna configurata come router: il dispositivo può ricevere eventuali comandi in qualsiasi momento, senza dover aspettare la fase di wake-up.

Per quanto riguarda i sensori utilizzati per il monitoraggio è stata eseguita una distinzione tra i sensori che vengono utilizzati per un eventuale allertamento (accelerometro) e sensori per il monitoraggio di variabili ambientali quali pluviometro, sensore di temperatura, sensore di pressione atmosferica, da utilizzare ad esempio come input per i modelli di preannuncio. I sensori che sono stati implementati per la lettura delle variabili di interesse nel firmware sono elencati nella tab. 6.1.5. I sensori di allertamento sono sempre attivi ed effettuano un controllo continuo delle variabili di interesse, comparando la misura precedente con quella successiva per verificare se un'eventuale soglia di attenzione è stata superata o meno. Nel caso dell'accelerazione, qualora questa soglia sia stata superata si ottiene un motion detection (*watch*).

I sensori di monitoraggio delle variabili ambientali, invece, effettuano le letture secondo un intervallo di campionamento predefinito (*sample*). Per tale motivo, questi sensori sono tenuti in fase di sleep prima del campionamento e si risvegliano ed effettuano le letture solamente per un breve periodo di tempo, proporzionale al tempo di risposta del sensore. Il firmware scritto secondo questa logica risulta essere univoco per tutti i dispositivi, eventuali sensori aggiunti o rimossi da un dispositivo sono gestiti attraverso dei flag booleani. La configurazione di un singolo dispositivo, nella fase iniziale, è gestita dalla SD Card, attraverso dei flag (1 Vero, 0 Falso) così come gli intervalli di acquisizione e le soglie di allertamento. Il file, denominato 'MOTE.conf', deve essere contenuto nella directory root della SD Card. Tutti i flag, nella fase successiva, invece, sono gestiti dal CAED che può eventualmente modificarli, qualora se ne rilevi la necessità, da remoto per mezzo dei comandi definiti nel protocollo di messaging. Il firmware, infine, è dotato di una procedura di salvataggio di eventuali pacchetti non inviati e di recupero di quest'ultimi ad un prefissato intervallo temporale qualora le condizioni di comunicazione vengano ripristinate.

Funzioni principali del firmware

La possibilità di adottare un unico firmware per tutti i dispositivi è stato reso possibile utilizzando il file di configurazione MOTE.conf. Il file è composto da 33 righe: la prima riga è rappresentata da un flag relativo alla presenza della comunicazione radio attraverso protocollo ZIGBEE, le successive undici righe sono flag relativi ai sensori delle variabili ambientali, dodici righe successive sono bianche per eventuali future integrazioni di ulteriori sensori rispetto a quelli utilizzati in questa fase; le altre nove righe, infine si riferiscono agli intervalli di acquisizione e di recupero, alla soglia di allerta dell'accelerometro ed all'intervallo minimo tra due allerte successive, e due flag uno relativo al salvataggio dei dati nel caso di errore di trasmissione e l'altro alla stampa di eventuali dati in seriale (USB).

La funzione preposta per la lettura della configurazione è `SD_CAED_Setup()` che legge le righe della configurazione ed assegna a ciascuna variabile globale il valore corrispondente. Il firmware è composto da tre parti fondamentali: le variabili globali, la funzione `setup()` e la funzione `loop()`: nel `setup` vengono inizializzati i sensori utilizzati per il monitoraggio, la trasmissione radio e/o seriale, la SD Card ed il Real Time of Clock (RTC). Nel `loop` invece, sono definite le procedure per l'acquisizione e l'invio dei dati, le regole di motion detection ed è

istanziata la funzione per l'acquisizione dei comandi inviati dal CAED da parte del dispositivo. I sensori per il monitoraggio ambientale sono acquisiti, come evidenziato in precedenza, secondo un predefinito intervallo di campionamento: al fine del risparmio energetico i sensori vengono spenti quando non devono acquisire. L'unico sensore che rimane sempre attivo è il pluviometro per avere informazioni continue della precipitazione. Il messaggio da inviare si basa su payload (vedi Message Protocol) ed è composto da un header e dai dati da inviare: l'header è formato dall'identificativo univoco del dispositivo (MAC Address), dalla data e ora di acquisizione e dal tipo di messaggio inviato. Il tipo di messaggio per campionamento è identificato con la dicitura #sample.

La costruzione del messaggio è affidata ad una classe esterna, denominata Message.cpp, che ricostruisce il messaggio in base ai valori assunti dai sensori montati sul dispositivo. La classe Message.cpp è utilizzata per ricostruire tutti i messaggi che devono essere inviati con il firmware. Con questo approccio è stato possibile avere un risparmio di memoria all'interno del dispositivo, da poter utilizzare per l'invio del messaggio con tecnologia ZigBEE.

Il messaggio realizzato con la classe descritta precedentemente è inviato ad un nodo coordinatore, individuato attraverso l'identificativo univoco (*destinazione*). Nel caso l'invio del messaggio vada a buon fine la funzione ritorna uno stato pari a zero (state = 0); nel caso, invece, vengono riscontrati errori di trasmissione, il messaggio viene salvato all'interno della memoria secondaria di massa (SD card). Quest'ultima azione viene effettuata solamente se il flag all'interno del file di configurazione è stato posto pari ad uno (SD_log), altrimenti non viene eseguita nessun azione di salvataggio del messaggio non inviato. Il salvataggio dei messaggi è eseguito su file di testo nominati in ordine numerico crescente, contenenti ciascuno, al massimo tre pacchetti non inviati. Il recupero dei file avviene secondo una procedura di tipo LIFO (Last In First Out) a livello di directory, FIFO (First In First Out) a livello di file: quando la procedura è richiamata all'interno del loop viene recuperato ed inviato un singolo file contenente al massimo tre pacchetti. Per non appesantire la trasmissione ed i normali processi del microcontrollore è stato scelto di recuperare eventuali pacchetti non inviati secondo un intervallo di tempo definito preliminarmente. Quando tutti e tre i pacchetti del file in corso di recupero sono stati inviati, il file viene cancellato per liberare memoria dalla SD Card. Per eseguire un risparmio energetico, il pin della SD viene acceso e alla fine del recupero viene spento.

Oltre al messaggio di campionamento, particolare attenzione nella scrittura del firmware è stata data alla realizzazione di una procedura, quanto più semplice possibile per identificare condizioni che sono indicative di possibili movimenti di un corpo di frana. Il sensore utilizzato per individuare le condizioni di innesco è l'accelerometro: nel loop del firmware viene comparato, per ogni singolo asse, il valore dell'accelerazione registrata allo step precedente e quella allo step successivo. Nel caso la differenza tra i due valori è superiore ad una certa soglia, il messaggio è contrassegnato con la dicitura **#watch** e contiene, oltre all'intestazione, i valori di accelerazione che hanno determinato una situazione di criticità. Data la necessità di avere continue informazioni da parte dell'accelerometro, per individuare le condizioni di innesco, questo sensore è tenuto costantemente in funzione.

Invio dei comandi dal CAED al dispositivo

Le funzioni analizzate precedentemente effettuano un campionamento delle variabili ambientali, ad un determinato istante temporale, ed un confronto in continuo dei valori registrati dal sensore accelerometrico, generando dei messaggi di tipo **#sample** nel caso del campionamento, **#watch** nel caso di superamento di una certa soglia. La gestione di tutto il firmware, come evidenziato, è affidato completamente al file di configurazione MOTE.conf: intervalli di campionamento, soglie di controllo, flag di trasmissione, recupero e sensori installati. Per avere una completa gestione del dispositivo da parte del CAED sono state implementate alcune funzioni che riescono a ricevere e processare comandi impartiti da remoto. La gestione in tempo reale del dispositivo può essere eseguita solamente se il collegamento radio risulta essere raggiungibile in ogni istante, per tale motivo il modulo ZigBEE deve essere sempre funzionante e mai in fase di sleep, anche per i nodi end-device che teoricamente possono avere questo comportamento. Affinché la radio ZigBEE non vada in sleep bisogna implementare il firmware di tipo Router (per cambiare configurazione alle antenne ZigBEE-Pro da end-device a router è stato utilizzato il software X-CTU specifico per le antenne che sono in dotazione, XBee® ZB). L'invio di comandi dal CAED al dispositivo avviene secondo una procedura standardizzata: il comando è composto da due argomenti il primo è relativo ad un codice che identifica il comando, il secondo, di tipo opzionale, può essere lungo al massimo 10 valori e può essere rappresentativo di un ulteriore comando ovvero di un intervallo temporale espresso in millisecondi. Il comando ed il suo attributo opzionale,

all'interno di un pacchetto ZigBEE sono inseriti nel payload e sono così descritti: ;comando;;attributo;; Il comando, una volta giunto al dispositivo e processato, genera un messaggio di tipo **#ack** è inviato al nodo coordinatore. Il messaggio di tipo **#ack** è costituito da un'intestazione e dal comando inviato (eventuali attributi non vengono inseriti in questo messaggio); il comando inviato può essere un comando che deve contenere una risposta ovvero un comando di cambio o settaggio dei parametri del dispositivo. Nel primo caso, il nodo coordinatore, oltre ad un messaggio di **#ack** attende che arrivi un messaggio di **#response**; nel secondo caso, invece, solamente il messaggio di acknowledge è utile per capire se il comando è stato ricevuto dal firmware.

Sensori Implementati nel firmware

I sensori che possono essere eventualmente aggiunti o eliminati dal dispositivo, utilizzando i flag del file di configurazione, sono riassunti in tab. 6.1.5. Per ognuno dei sensori è indicato il modello del sensore, il tipo di misurazione eseguito, la variabile che viene utilizzata all'interno del message protocol, l'unità di misura e il range di funzionamento.

Modello Sensore	Tipo Misurazione	Variabile	Unità di Misura	Range	
				Minimo	Massimo
RTC Temperature	Temperatura Interna	-int_temp:	°C	-40	85
LIS3LV02DL	Accelerazione (Proper Acceleration)	-x:,y:,z:	LSB/g		
MPX4115A	Pressione Atmosferica	-press:	kPa	15	115
MCP9700A	Temperatura Atmosferica	-temp:	°C	-40	125
SHT75	Temperatura Atmosferica	-temp:	°C	-40	123,8
	Umidità Atmosferica	-hum:	%	0	100
Watermark	Tensione di Suzione	-suct:	cb	0	200
PT-1000	Temperatura del suolo	-soil_temp:	°C	-50	300
SQ-110	Radiazione solare	-solar_rad:	$\mu\text{mol}\cdot\text{m}^{-2}\text{s}^{-1}$	0	2000
Anemometer	Velocità del Vento	-wind_vel:	Km/h	0	240
Wind Vane	Direzione del Vento	-wind_dir:	Direzione	N, E, S, W ...	
Pluviometer	Precipitazione	-prec:	mm ²	0	-

Tab. 6.1.5 - Sensori che sono stati inseriti nel firmware per la lettura delle variabili di interesse.

² Precipitazione cumulata nell'intervallo di campionamento adottato.

Scambio acquisizioni/comandi SINK to CAED

La rete di laboratorio realizzata dispone di un nodo coordinatore, con caratteristiche hardware e prestazioni superiori a quelli dei motes, che si occupa di raccogliere tutti i messaggi prodotti dai dispositivi ed inviarli tramite protocollo di comunicazione TCP/IP al CAED. All'interno del nodo coordinatore sono stati realizzati due processi:

1. *AqSink*, client di *AqServ* per la comunicazione con il CAED (protocollo di comunicazione TCP/IP);
2. *ZBFetcherSender*, comunica con la rete di monitoraggio in protocollo ZigBEE.

La rete di laboratorio, seppure a piccola scala, riproduce esattamente quanto realizzato nel corso della sperimentazione dai partner che si occupano della realizzazione dei firmware dei sensori. La realizzazione della rete di laboratorio ha permesso di valutare le prestazioni di *AqServ*, la struttura dei comandi remoti e l'interazione tra il CAED e la rete ed infine di realizzare procedure di validazione dipendenti dalla tipologia del messaggio. Inoltre, sono state definite le priorità che devono possedere i firmware nel caso di assenza del collegamento, per il recupero dei messaggi non inviati al CAED, condivise con i partner.

6.1.5.C TESTING CON I PROGETTISTI DELLE RETI DI SENSORI

Nell'ultima parte della presente attività elementare, in seguito ai test con il simulatore e la rete di laboratorio sono state effettuate delle robuste prove con tutti i partner che devono comunicare direttamente con il CAED.

In base alle politiche di indipendenza di ciascuna rete, alcuni partner hanno deciso di sviluppare autonomamente il client che comunica con il CAED, altri invece hanno utilizzato il prodotto realizzato dalla OR 5 (reti di trasmissione) adattato alle specifiche esigenze (Fig. 6.1.19).

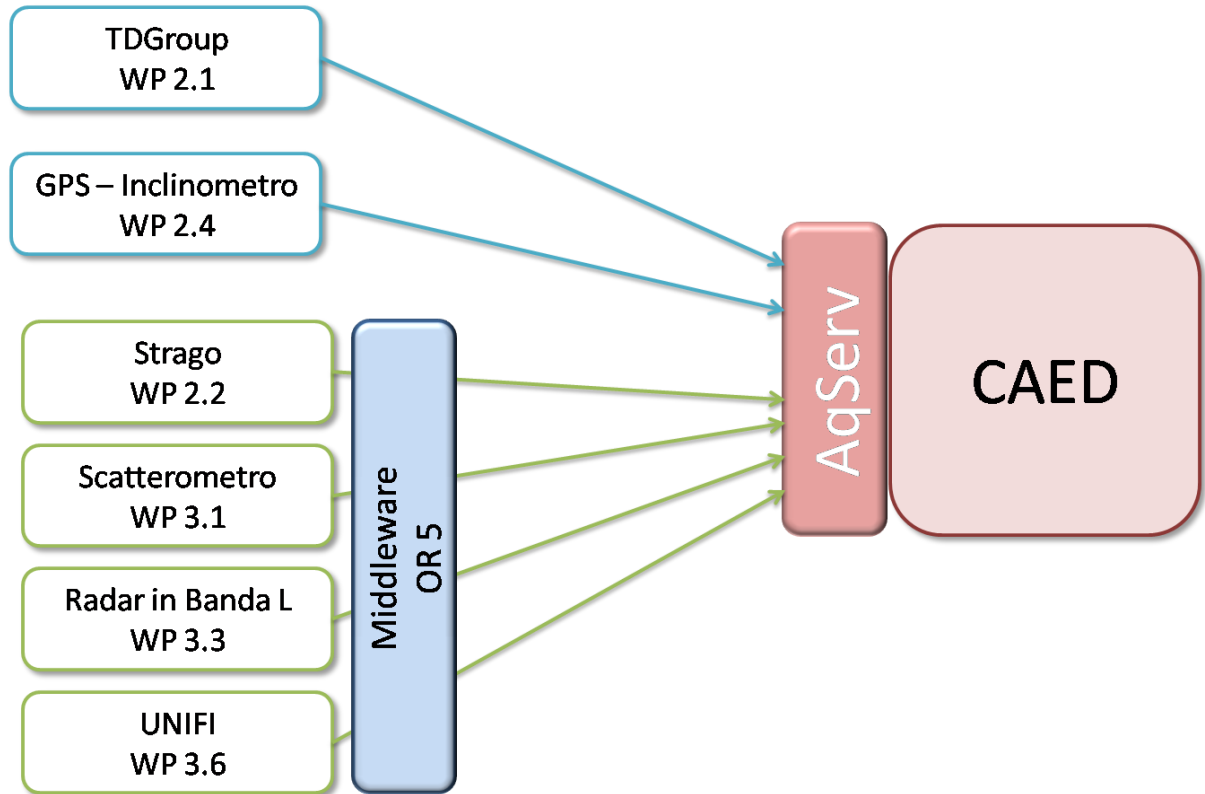


Fig. 6.1.18 – Implementazione dei client per lo scambio dei messaggi tra le WSN e il CAED.

Per ciascuno dei partner, preliminarmente alla messa in produzione del sistema, bisogna verificare tutta la filiera procedurale:

1. autenticazione;
2. acquisizione;
3. validazione;
4. estrazione contenuto informativo;
5. invio/recepimento dei comandi remoti.

Come evidenziato dalla figura 6.1.18, due partner (Td Group ed il gruppo del WP 2.4) hanno sviluppato separatamente il client di connessione verso AqServ, gli altri partner hanno utilizzato il middlewre della OR5. Per ciascun partner è stata definita preliminarmente la porta di connessione, per distinguere i relativi

log e per distribuire il carico computazionale e di rete in maniera uniforme, evitando i sovraccarichi ed i possibili colli di bottiglia.

Le porte sono state assegnate nel seguente modo:

- porta **8121**, UNIFI WP 3.6;
- porta **8122**, OR 3 WP 3.1 e 3.3;
- porta **8123**, Strago WP 2.2;
- porta **8124**, TDGroup WP 2.1;
- porta **8125**, Artese WP 2.4.

Nel seguito della trattazione sono discusse le tipologie delle reti utilizzate come test ed i passi procedurali che hanno permesso il test del sistema di acquisizione in tutte le sue componenti. Per ciascuna delle reti è stato imprescindibile registrarle in Lewis prima del test, dopo la compilazione da parte degli staff di un modello realizzato dal CAED denominato MRD013 – Moduli registrazione dispositivi monitoraggio ambientale.

UNIFI WP 3.6

1. Registrazione della rete

La tipologia di rete utilizzata per lo scambio di messaggi tra il CAED e l'interferometro è quella che realmente sarà dispiegata lungo i tratti di monitoraggio. Essa è composta da un nodo coordinatore, rappresentato da un personal computer con architettura x86, dotato di un modem GSM che comunica direttamente con il CAED. Il nodo coordinatore si occupa delle funzioni principali di elaborazione della scena interferometrica che sono: 1) acquisizione matrice binaria dallo strumento; 2) conversione in formato ascii della matrice; 3) correzione atmosferica; 4) formattazione dei messaggi da inviare; 5) invio de messaggi e verifica consegna; 6) elaborazione delle scene per intervalli di aggregazione ad 8 e 24 ore.

Il nodo coordinatore è collegato direttamente allo strumento che acquisisce la scena interferometrica con passo temporale di circa 5 minuti.

Le elaborazioni riguardo, inoltre, la determinazione dei punti di controllo all'interno della matrice interferometrica e la relativa formattazione nei messaggi da scambiare con il CAED. Anche per i punti di controllo, all'interno del nodo coordinatore è effettuata una aggregazione a 8 e 24 ore.

2. Test del sistema di autenticazione

A seguito della registrazione della rete di test/sperimentazione³, il CAED ha fornito le credenziali di accesso ad AqServ inserite, dal gruppo UNIFI, nel file di configurazione del middleware (OR 5). A seguito di queste procedure sono iniziati i primi test di connessione e verificate le fasi di autenticazione discusse in fig. 6.1.4.

3. Definizione del message protocol

La struttura utilizzata dall'UNIFI è quella proposta dal CAED ed è simile alla seguente:

[0024E8BE5355 14-07-24 01.37.03 #sample] -var: (1001),(1001)

Dove – var indica la variabile inviata che può essere: interferometro a 5 minuti, 8 ore e 24 ore, mentre le acquisizioni vengono inviate in forma matriciale con numero di righe e di colonne pari a 1001. In maniera del tutto analoga vengono inviati i control point:

[0024E8BE5355 14-07-24 01.37.03 #sample] -ctrl0_8:3.2355, ...

Nell'esempio precedente, i punti di controllo, attraverso il suffisso 8 sono relativi alla serie storica delle matrici interferometriche aggregate ad 8 ore.

4. Inserimento acquisizioni

Registrata la rete, verificato il sistema di autenticazione e definite tutte le tipologie di messaggi inviati al CAED (matrici e control point, per diverse serie temporali) sono iniziati i test di trasmissione con l'invio delle acquisizioni da parte dello strumento in campo aperto. Dopo le prime verifiche ed i primi affinamenti ad AqServ è possibile ricevere da parte dello strumento, fino a 2GB di acquisizioni giornaliere.

³ La rete di sperimentazione ha la stessa tipologia di quella di test.

OR 3 WP 3.1 e 3.3

1. Registrazione della rete

I due strumenti prodotti dalla OR 3 sono lo scatterometro ed il radar in banda L. Entrambi gli strumenti sono dotati di un nodo coordinatore, equipaggiato con un modem GSM, che comunica direttamente con il CAED per l'invio dei messaggi. Le acquisizioni, trattandosi di strumenti areali, come nel caso dell'interferometro, sono configurate per essere acquisite in 3 dimensioni, le prime due spaziali e la terza temporale. Dal punto di vista spaziale, gli strumenti danno informazioni di 4 macro settori irradiati, rispetto all'area di investigazione, e ciascun settore è stato configurato separatamente all'interno di Lewis:

- Funzionamento on demand;
- Risoluzione spaziale: n° 4 settori;
- Risoluzione temporale: campionamento.

Al contrario dell'interferometro, i due radar rimangono in fase di quiescenza fin quando non vengono attivati dal CAED con l'invio di un sms, per il verificarsi di condizioni diverse da quelle di criticità nulla.

I due strumenti, ovvero il nodo coordinatore, non effettuano aggregazioni temporali delle acquisizioni campionarie per cui è stato necessario registrare solamente il singolo sensore ed i quattro sottosectori assieme al nodo coordinatore e quello di primo livello.

2. Test del sistema di autenticazione

Sono state fornite le credenziali di accesso del nodo coordinatore, utilizzate dal middleware per comunicare con il CAED e verificate le connessioni e le disconnessioni in assenza di acquisizioni. La verifica ha dato esito positivo.

3. Definizione del message protocol

Scatterometro

```
[PNA8041_12105101 14-07-16 10.51.22 #sample]  
sect_1:[((0,1500);(0,1))*5001]
```

Radar in banda L

```
[EAR12VCNW 14-07-16 10.51.22 #sample]  
sect_1:[((0,1000);(0,2))*167]
```


Nei due esempi precedenti i messaggi si riferiscono ad un solo settore rappresentato da una lista di tuple; ogni tupla è relativa al campo elettrico (0 – 2 o 0 – 1) trasmesso ad una specifica distanza.

4. Inserimento acquisizioni

A seguito dei numerosi test di autenticazione (il radar in banda L è stato utilizzato dalla OR 5 come primo sensore per lo sviluppo del middleware) sono stati inviati i messaggi di ciascun settore sia in laboratorio che in campo aperto. Rispetto all'interferometro, i messaggi inviati sono dell'ordine delle centinaia di KB, acquisiti solamente su richiesta dal CAED.

Strago WP 2.2

1. Registrazione della rete

La rete Strago (SMAMID) utilizzata come test di trasmissione ed acquisizione verso il CAED è composta da un nodo coordinatore e da 4 nodi di primo livello (mote) dotati di un accelerometro triassiale. Il nodo coordinatore acquisisce i dati dai nodi di primo livello, tramite collegamento wireless proprietario, ed effettua due elaborazioni: la prima è relativa al singolo nodo poiché calcola l'inclinazione rispetto alla verticale, la seconda è relativa alla rete, attraverso un valore identificante lo stato della rete (compreso tra 0 e 3).

Il CAED è predisposto per inviare due tipi di comandi remoti alla rete SMAMID per la riconfigurazione:

1. Intervallo di campionamento tra due acquisizioni successive (sec.);
2. Intervallo di acquisizione per il calcolo dell'accelerazione media su ogni asse (sec.).

Il nodo coordinatore, come nel caso dell'interferometro, è un nodo con architettura x86.

2. Test del sistema di autenticazione

Sono state fornite le credenziali di accesso del nodo coordinatore, utilizzate dal middleware per comunicare con il CAED e verificate le connessioni e le disconnessioni in assenza di acquisizioni. La verifica ha dato esito positivo.

3. Definizione del message protocol

Ogni nodo, per ogni intervallo di campionamento produce un messaggio contenente le accelerazioni lungo i tre assi, le inclinazioni, la temperatura dell'aria e la differenza di potenziale della batteria che li alimenta. Il messaggio è inviato al CAED secondo la seguente struttura:

```
[SM103 14-07-20 11.01.00 #sample] -acc_x:-217.77,-acc_y:-109.19,-  
acc_z:948.24,-wht_temp:24.10,-bat_v:0.01,-incl_x:102.85,-  
incl_y:96.40,-incl_z:14.41
```

Il messaggio inviato dal nodo coordinatore, che da informazioni generali sullo stato della rete, per la valutazione delle criticità sui singoli corpi di frana, è il seguente:

```
[SM14S0A812 14-07-20 11.01.00 #sample] -ew_stat:0;
```

4. Inserimento acquisizioni

Le acquisizioni sono inviate al CAED ad intervallo temporale di 3 minuti, nella configurazione standard, per ciascuna acquisizione sono inviati 5 messaggi distinti: 4 per i sensori montati sui nodi di primo livello ed 1 per il nodo coordinatore. I messaggi, correttamente formattati, sono acquisiti da AqServ ed archiviati in Lewis. La scalabilità con cui è stato progettato e realizzato Lewis permette di essere indipendente dal numero di nodi e di messaggi. Nel caso di nodi superiori a dieci, possono essere riscontrate delle latenze, facilmente risolvibili estendendo il middleware per l'utilizzo della message aggregation.

Oltre alle acquisizioni, sono state verificate le procedure per l'acquisizione dei comandi remoti da parte del nodo coordinatore. I messaggi inviati per la riconfigurazione hanno la seguente struttura (formato JSON):

OUT: 205 - {'id':48,'destination':'SM14S0A812','command':1,'arguments':[10]}

OUT: 205 - {'id':49,'destination':'SM14S0A812','command':2,'arguments':[240]}

TDGroup WP 2.1

1. Registrazione della rete

Le reti che TDGroup ha utilizzato per effettuare i test di acquisizione sono due. Entrambe sono delle stazioni meteorologiche che comunicano singolarmente con il CAED tramite un sistema embedded progettato e realizzato dalla TDGroup. La comunicazione con AqServ è assicurata da un collegamento GSM, mentre il client di connessione è stato sviluppato indipendentemente dalla OR 5 (fig. 6.1.18).

La stazione meteorologica è a sua volta composta da più sensori: termometro, igrometro, pluviometro, anemometro e radiometro. La stazione è inoltre dotata di sensori per la misura delle variabili del suolo quali il contenuto volumetrico di acqua.

Il funzionamento della stazione meteo è quello classico: ad un intervallo di campionamento vengono acquisite le misure ed inviate al CAED. Inviare i messaggi allo stesso istante di campionamento, nel caso dei sistemi embedded, ha due limiti: 1) quando vengono inviati i dati non possono essere effettuate acquisizioni; 2) consumi energetici eccessivi. Per ottimizzare il sistema e minimizzare gli effetti di un campionamento/invio frequente, insieme alla TDGroup, è stato deciso di trasmettere, ad un predefinito intervallo di tempo, più acquisizioni.

Per riconfigurare la rete da parte del CAED, TDGroup ha predisposto due comandi remoti:

- a) intervallo di campionamento;
- b) intervallo di trasmissione.

Altre informazioni, di carattere diagnostico/elettronico sono inviate ad AqServ ma non utilizzate direttamente per la gestione del sistema di early warning.

2. Test del sistema di autenticazione

A seguito della registrazione delle due WSN, sono state fornite le credenziali di accesso ad AqServ e verificate le connessioni in ingresso e le disconnessioni. La verifica ha dato esito positivo senza nessun problema particolare.

3. Definizione del message protocol

I messaggi sono formattati secondo la struttura vista nei precedenti paragrafi, di tipo #sample, suddivisi in base alle variabili acquisite:

a) precipitazione;

```
[TDgroup_meteo_2 14-10-19 23:55:00 #sample] -  
wh_t_rain:+7.112000,wh_t_rainh:+0.0
```

b) umidità, temperatura e radiazione dell'aria;

```
[TDgroup_meteo_2 14-10-19 23:45:00 #sample] -  
wh_t_rad:+0.006235
```

-

```
wh_t_temp:+16.035839,wh_t_humi:+99.562637,wh_t_dewpoint:+1  
5.967138;
```

c) velocità e direzione del vento;

```
[TDgroup_meteo_2 14-10-20 00:00:00 #sample] -  
wh_t_direction:+212.240066  
-wh_t_speed:+0.503074,wh_t_gust:+0.574181
```

d) suolo;

```
[TDgroup_soil__2 14-10-19 23:35:00 #sample]
```

-

```
soil_perm:+8.170000,soil_cond:+0.09,soil_temp:+21.0,soil_vw  
c:+0.151197;
```

4. Inserimento acquisizioni

Le impostazioni di default delle due WSN di test sono di acquisire i dati ogni 5 minuti ed inviarli al CAED ogni 30 minuti. Con la struttura analizzata nel sottoparagrafo precedente ad ogni trasmissione devono essere inviati 24 messaggi, più eventuali messaggi di diagnostica. Inviare i singoli messaggi richiedeva molto tempo e per tale motivo è stato utilizzata l'estensione della message aggregation per la trasmissione di

tutti i messaggi in un unico istante. In poco meno di 3 secondi vengono acquisiti tutti i messaggi, validati ed inseriti nel database.

La verifica dei comandi remoti è stata fatta parallelamente alle acquisizioni. I comandi che AqServ invia sono:

```
OUT: 205 - {'id': 47,'destination':'TDgroup_meteo_2',  
'command':20,'arguments':[300]}
```

```
OUT: 205 - {'id':32,'destination':'TDgroup_meteo_2',  
'command':39,'arguments':[1800]}
```

Artese WP 2.4

1. Registrazione della rete

La rete progettata dalla WP 2.4 (denominata *Artese* in riferimento al responsabile di WP) è stata realizzata in maniera classica: un nodo coordinatore che gestisce n nodi di primo livello a cui sono associati un sensore di inclinazione ed un sensore di posizione (GPS). La rete è stata interamente realizzata con dispositivi embedded autoalimentata attraverso dei pannelli solari; anche in questo caso devono essere ottimizzate le connessioni verso il CAED per garantire una durata della WSN quanto più lunga possibile, senza manutenzione. La rete utilizzata come test è dotata di un nodo coordinatore, che comunica con il CAED tramite connessione GSM e da un nodo di primo livello.

Nessun comando remoto registrato.

Client di connessione verso AqServ realizzato indipendentemente dalla OR 5 (fig. 6.1.18).

2. Test del sistema di autenticazione

Preliminarmente è stato registrato il nodo coordinatore e sono state fornite le credenziali di accesso. A seguito di alcuni test il client sviluppato è stato allineato ad AqServ e sono state verificate correttamente le connessioni in ingresso e le disconnessioni.

3. Definizione del message protocol

Ogni messaggio trasporta con se 5 variabili, due relative all'inclinometro (pitch e roll) e tre al GPS (X, Y, Z). La struttura del messaggio è simile a quanto analizzato precedentemente per gli altri partner:

```
[b827eb1a4366 2014-06-10 13:14:45 #sample] -incl_pitch:-0.05131 -  
incl_roll:0.01394 -x: 0.000000,y: 0.000000,z: 0.000000
```

4. Inserimento acquisizioni

Le acquisizioni sono state effettuate solamente da un unico nodo, ma per il principio di scalabilità di Lewis, una volta registrati gli altri sensori e nodi che compongono la rete, verranno acquisiti senza particolari problemi. Da verificare la possibilità di implementare, da parte del client, la message aggregation per l'invio di più messaggi. In maniera analoga alla rete SMAMID, ogni acquisizione dell'intera rete deve essere corredata da una opportuna variabile (ew_stat) che indichi possibili condizioni di movimento del corpo di frana monitorato.

MRD013 – Moduli registrazione dispositivi monitoraggio ambientale

Staff di competenza

Gruppo di Lavoro o Società di Appartenenza	
Referente tecnico, Nome e Cognome	
Data Compilazione	
Email di Riferimento	
Contatto Telefonico	

Scheda da replicare per ogni dispositivo

Tipo dispositivo	End device/Router/Coordinatore/Gateway/Sink (Coordinatore+Gateway)
Identificativo univoco ^[1]	
Prodotto da ^[2]	
Nome Modello	
Versione Software/Firmware	
Modelli sensori montati ^[3]	
Coordinate di Posizionamento ^[4]	
Operante in WSN (nome dato alla rete WSN)	

Scheda da replicare per ogni sensore

Prodotto da	
Montato su dispositivo (identificativo univoco)	
Tipo di misurazione ^[5]	
Modello	
Unità di misura ^[6]	
Variabili della misurazione (nomi) ^[7]	
Range acquisizione ^[8]	
Link web (url) datasheet	

- [1] preferibile MAC address, lunghezza massima 16 caratteri
- [2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
- [3] es, separati da virgola: LIS3LV02DL, SHT75.
- [4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
- [5] es. proper acceleracion, rainfall, atmospheric temperature
- [6] es. percent (%), gradi celsius, millimeters, voltage
- [7] es. "x,y,z" oppure "temp", oppure "wind_vel"
- [8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo: -20;80
se indicato nel seguente formato indica un set di valori:
-20, 0, 20, 32.26, 52.19, 84.44

Architettura CAED

Il Centro di Elaborazione ed Acquisizione dei Dati è composto da Hardware e Software, tale insieme è indicato come Infrastruttura Telematica (all'interno del progetto tale infrastruttura è stata denominata *LEWIS*). Nella presente attività elementare, sono descritte le componenti principali dell'Hardware e del Software che costituiscono il Centro di Elaborazione dei Dati. Per quanto riguarda l'architettura Hardware, sono definiti tutti i componenti per la progettazione fisica acquistati nella fase di sperimentazione del progetto. Mentre, per il software, è definita l'architettura che utilizza il CAED per l'acquisizione e l'elaborazione dei dati.

L'Hardware è composto da:

- cablaggi;
- dispositivi di networking;
- computer di fascia alta predisposti come server.

Il Software è composto da:

- sistemi operativi;
- ambiente di sviluppo;
- programmi;
- volumi di dati.

Tipicamente una macchina hardware server può implementare logicamente uno o più server software. Nel seguito sono indicati come Server Hardware in riferimento alle macchine hardware server, mentre in riferimento ai Servizi vengono denominati semplicemente Server. Il Server Hardware è un computer specifico appartenente alla fascia di mercato dedicata all'uso come server, caratterizzato da alta affidabilità e maggiori prestazioni. Il Server Hardware con l'ausilio di un software specifico (Server Web, Server FTP, Server Mappa, RDBMS, Procedure, Modelli ecc.) fornisce uno o più servizi parallelamente, insieme alle procedure di acquisizione, elaborazione e rappresentazione dei dati.

6.1.6.A ARCHITETTURA HARDWARE

Il deployment della infrastruttura del CAED mira ad ottimizzare il rapporto tra il risultato atteso e le risorse impiegate. Il CAED adotta accorgimenti volti a garantire alta affidabilità, ovvero continuità di servizio, robustezza ai guasti e alta sicurezza, ovvero protezione da abusi informatici. L'architettura telematica del CAED richiede:

- utilizzo di hardware di classe superiore, con prestazioni elevate ed elementi ridondanti (dischi RAID, alimentatori ridondanti);
- protezione dell'alimentazione elettrica mediante gruppo di continuità;
- protezione fisica mediante localizzazione in un locale apposito ad accesso ristretto ("sala server") o in housing;
- connessione di rete preferenziale, di capacità superiore, ridondante;
- climatizzazione;
- configurazioni del sistema operativo volte a garantire maggiore affidabilità e sicurezza (hardening);
- gestione da parte di sistemisti esperti.

Per poter garantire queste caratteristiche sono stati scelti 3 server, di tipo rack, i primi due di gestione dei servizi, degli applicativi e dei dati, il terzo invece, denominato server di supporto, agisce come backup dei due server principali o, in caso di malfunzionamenti, può essere utilizzato per la gestione di servizi specifici (ad esempio, replicazione di AqSERV).

I tre server individuati per la realizzazione del CAED sono:

- Dell PowerEdge R620 x10 Base, Intel® Xeon® E5-26XX v2 Processors;
- Dell PowerEdge R720xd, Intel® Xeon® E5-26XX v2 Processors;
- Dell PowerEdge R210 II Chassis, 2x3.5 Cabled HDDs, LED Diagnostics;

Entrambi i server sono dotati di dischi ridondanti e indipendenti che condividono e replicano le informazioni attraverso il controller RAID, aumentando l'affidabilità e la velocità in lettura. I due server principali sono stati progettati con

dischi da 1.2 TB, SAS 6Gbps (5 dischi per il primo server e 4 per il secondo server), mentre il server di supporto è stato munito di dischi SAS 6Gbps con capacità di 4 TB. La velocità delle elaborazioni è garantita da processori con frequenza di clock pari a 3.3 GHz (3.1 per il server di supporto), mentre l'elevato scambio di informazioni in rete è assicurato da schede di rete da 10Gb/1Gb.

Inoltre, altri accorgimenti per la progettazione ottimale dell'architettura telematica sono:

- a) Protezione dell'alimentazione elettrica mediante due gruppi di continuità da 1920 W da rack;
- b) Alimentatori ridondati (1+1), 750W;
- c) Armadio rack da 24U.

I computer scelti per essere usati come server, disposti in una sala server di dimensioni medio-grandi, sono montati all'interno dell'armadio sopra citato, al fine di risparmiare spazio e concentrare in un unico volume le soluzioni di raffreddamento, climatizzazione e messa in sicurezza dagli accessi fisici.

6.1.6.B ARCHITETTURA SOFTWARE CAED

I sistemi operativi adottati dal CAED si basano su Debian GNU/Linux. L'adozione di sistemi GNU/Linux è conseguente alla valutazione prestazionale di questi sistemi in ambito network e alla flessibilità introdotta negli ambiti sperimentali e di sviluppo software server side network oriented.

Il sistema operativo, su entrambi i nodi del server, è configurato per svolgere funzioni di hypervisor: si tratta di un virtual machine manager per poter sfruttare, a pieno, le potenzialità hardware in dotazione. In questo modo, i principali servizi che eroga il CAED sono soddisfatti da specifiche macchine virtuali, che emulano il comportamento di una macchina fisica ed ottimizzano le risorse a disposizione, ed in caso di malfunzionamenti o corruzione del sistema non pregiudicano il comportamento della macchina fisica. L'hypervisor, inoltre, permette di destinare le risorse hardware in maniera dinamica alle diverse macchine virtuali, di ridurre il tempo di messa in esecuzione di sistemi operativi e di gestire, in maniera semplice, il sistema tramite interfaccia grafica ovvero testuale (shell).

Lo sviluppo software, per l'infrastruttura IT del CAED, è realizzato attraverso i seguenti linguaggi di programmazione: Python, ANSI C/C++, javascript, SQL 99/2003, Fortran. Lo sviluppo server è stato realizzato interamente in Python (come già visto per lo sviluppo di AqServ), mentre l'applicativo web è stato implementato con il framework Django, che implementa il pattern Model-View-Controller (fig. 6.1.19).



Fig. 6.1.19 – Pattern MVC implementato in Django, per la realizzazione di applicativi web.

6.1.6.C IMPLEMENTAZIONE DEL DATABASE

Il Database Management System (abbreviato in DBMS) è il sistema software adottato per consentire la creazione e la gestione efficiente del database (ovvero la collezione dei dati strutturati) relativo alle acquisizioni e alle elaborazioni condotte dal CAED.

Il CAED ha progettato ed implementato un modello logico di tipo Relazionale e il linguaggio di interrogazione (query language) è l'SQL. Per mezzo di SQL i dati sono conservati in un numero definito di tabelle (come da modello) con chiavi esterne e costrizioni al fine di collegare le informazioni secondo vincoli di integrità.

La piattaforma adoperata come relational DBMS (RDBMS) del CAED è PostgreSQL, questa supporta la maggior parte dello standard SQL2003 e fornisce un'ampia gamma di estensioni proprie. La sua caratteristica object-relational, offre la possibilità di estendere il sistema tramite la definizione di nuovi tipi di dati, operatori e funzioni, all'interno di uno specifico database. Per mezzo di questo approccio e per via della sua estendibilità, PostgreSQL offre soluzioni

valide anche in campi come i sistemi informatici territoriali, attraverso estensioni orientate al GIS. L'estensione utilizzata dal CAED è denominata PostGIS: è stata integrata nell'attuale modello relazionale e permette la manipolazione e gestione di tipi di dati come raster e shape file, oltre che funzione e ulteriori facilitazioni per lo sviluppo di sistemi geospaziali.

Con l'ausilio di questa piattaforma il CAED ha adottato sul proprio RDBMS:

- viste
- schemi
- trigger
- tablespace
- transazioni
- rules (ovvero rewrite del comportamento delle query SQL)
- stored procedure (in diversi linguaggi di programmazione)
- interfacce di connessione (in particolare ODBC e JDBC)
- codifica caratteri (unicode)
- storage elevato di dati.

6.1.6.D MODELLO RELAZIONALE

Il modello relazionale che il CAED ha progettato ed implementato, pone lo schema di salvataggio dei dati e le relazioni interne a questi.

Secondo il modello relazionale adoperato, il CAED memorizza ed estrae i dati in apposite tabelle fra loro complementari secondo i seguenti principi:

- **normalizzazione:** procedimento volto all'eliminazione della ridondanza del database;
- **integrità:** vincolo che esclude incoerenze o perdite di informazioni dovute all'inserimento o alla rimozione di informazioni;
- **non eccedenza:** principio che vincola lo sviluppo del sistema a non favorire la duplicazione dei dati, evitare la ridondanza delle informazioni al suo interno;
- **generalizzazione:** ha la funzione di attenuatore di varietà sui dati acquisiti o elaborati allo scopo di semplificarne la gestione.

Il modello relazionale è stato realizzato secondo il paradigma MVC, descritto in fig. 6.1.19, attraverso una serie di applicativi e di classi implementate in python. Le classi forniscono una rappresentazione delle tabelle del database, facilitando le operazioni di CRUD (**C**reate, **R**ead, **U**ppdate, **D**eleate) sui dati, al posto delle classiche query SQL; in questo caso si parla di ORM (Object-Relational Mapping).

La complessità del modello relazionale è rappresentato da circa un centinaio di tabelle di configurazione, mentre le altre tabelle vengono create dinamicamente da AqServ, durante la prima sessione di salvataggio delle acquisizioni dalle WSN.

ATTIVITÀ ELEMENTARE 6.1.7

Trattamento dei dati

Lo scopo dell'attività elementare denominata "trattamento dei dati" è la realizzazione di procedure automatiche e semiautomatiche per estrarre l'informazione tipica dai dati strutturati, rendendola disponibile e direttamente utilizzabile sia dagli operatori del CAED attraverso interfacce grafiche ovvero testuali che nel sistema generale di Early Warning.

L'uso del message protocol ha permesso di normalizzare la struttura dei messaggi scambiati con le reti di sensori siano essi costituiti dalle acquisizioni dei sensori puntuali o dei sensori areali. Le estrazioni delle informazioni contenute nei messaggi è stata strutturata identificando, preliminarmente, i dati di interesse di ogni sensore utilizzato all'interno del progetto, distinguendo tra sensori puntuali e sensori areali.

6.1.7.A ESTRAZIONE DEI DATI DAI MESSAGGI

Il messaggio che giunge dal singolo nodo coordinatore, di ciascuna WSN segue una serie di regole di estrazione e validazione che sono discusse in dettaglio nelle AE 6.2.1. Strutturalmente, le estrazioni dei dati avvengono in due fasi:

1. individuazione del message protocol associato al nodo coordinatore;

2. topologia dei nodi di primo livello che compongono la WSN.

Tutti i messaggi grezzi, validati, sono archiviati all'interno dello schema⁴ "messages" e, perseguendo il principio di scalabilità orizzontale, automaticamente sono generate tante tabelle quanti sono i nodi di primo livello configurati nel database. Ogni tabella è denominata accoppiando l'identificativo univoco della WSN ed il nome univoco del nodo di primo livello; gli attributi principali della tabella sono (fig 6.1.21):

- *message*;
- *insert ed acquisition datetime*;
- *ip sink*;
- *sended by sink*;
- MD5.

	id_tabella	message	insert_datetime	sink_insert	acquisition_datetime	hops	source_ip	sended_by	md5
	[PK] integer	character varying(512)	timestamp without time zone	timestamp	timestamp without time zone	character	character	integer	character
5497	5533	('2014/1/26/w7_SM1450A812', 1173	2014-01-26 04:21:53.03		2014-01-26 03:21:28	2	109.113.4		69e4dbf8
5498	5534	('2014/1/26/w7_SM1450A812', 1178	2014-01-26 04:22:06.00		2014-01-26 03:21:38	2	109.113.4		16662319
5499	5535	('2014/1/26/w7_SM1450A812', 1183	2014-01-26 04:23:34.62		2014-01-26 03:23:09	2	109.113.4		b96c5b55
5500	5536	('2014/1/26/w7_SM1450A812', 1188	2014-01-26 04:23:44.92		2014-01-26 03:23:20	2	109.113.4		cb72ef9c
5501	5537	('2014/1/26/w7_SM1450A812', 1193	2014-01-26 04:25:17.20		2014-01-26 03:24:52	2	109.113.4		80723e77
5502	5538	('2014/1/26/w7_SM1450A812', 1198	2014-01-26 04:25:27.78		2014-01-26 03:25:02	2	109.113.4		8a6296eb
5503	5539	('2014/1/26/w7_SM1450A812', 1203	2014-01-26 04:25:37.98		2014-01-26 03:25:12	2	109.113.4		dd8ba513

Fig. 6.1.21 – Screenshot di una tabella dello schema MESSAGES.

Per non appesantire la base dati, i messaggi grezzi non sono conservati direttamente nel database ma vengono archiviati giornalmente in un file di testo esterno, distinto per dispositivo. Il link del messaggio grezzo con il database è garantito dall'attributo *message*, costituito da una tupla contenente il nome del file ed i riferimenti iniziali e finali, all'interno del file, di ogni singolo messaggio. Queste tre informazioni permettono di risalire, attraverso script realizzati ad-hoc, al messaggio originale inviato dal client del nodo coordinatore. L'attributo MD5, inoltre, garantisce il controllo di integrazione del messaggio: non possono essere inseriti all'interno della tabella messaggi aventi hash MD5 identici (univocità del

⁴ Gli schemi di PostgreSQL sono stati introdotti per superare i limiti di effettuare le query su due diversi database. Gli schemi sono entità distinte (proprie tabelle, estensioni e stored procedure) ma è possibile effettuare query su due schemi dello stesso database.

messaggio). L'uso di questa struttura, per l'archiviazione dei messaggi grezzi ha permesso di normalizzare la procedura sia per i sensori puntuali che per quelli areali, dipendente solamente dalla topologia della WSN (identificativo e dispositivi della rete).

6.1.7.B LOG E STORAGE DEI MESSAGGI GREZZI

Il CAED per garantire la persistenza dei dati ed analizzare le singole sessioni di acquisizione ha progettato e realizzato un sistema di log giornaliero ed una conservazione dei messaggi grezzi in file giornalieri (discusso nel paragrafo precedente).

I log giornalieri sono distinti in base allo specifico staff che si occupa della progettazione e realizzazione delle reti di monitoraggio, attraverso un suffisso (tab. 6.1.6).

Partner	Log Giornaliero
UNIFI	AqServ.daily_UNIFI.log
Strago	AqServ.daily_STRAGO.log
OR 3	AqServ.daily_OR3.log
TDGroup	AqServ.daily_TDGROUP.log
Artese WP 2.4	AqServ.daily_ARTESE.log

Tab. 6.1.6 – Log giornalieri delle acquisizioni da parte di AqServ.

All'interno del log file sono presenti le singole sessioni di connessione con le autenticazione, i messaggi scambiati, le procedure di acquisizioni e di estrazione e le eventuali eccezioni che possono essere riscontrate.

I log, alla mezzanotte di ogni nuovo giorno, sono archiviati attraverso un suffisso che indica il giorno corrispondente, per essere facilmente consultabili dagli operatori del CAED.

I messaggi grezzi sono archiviati all'interno di una struttura gerarchica di directory: 1) anno, 2) mese, 3) giorno, 4) dispositivo (nodo di primo livello). Nel caso dei sensori puntuali i messaggi grezzi di ogni singolo dispositivo possono avere una dimensione dell'ordine della decina di megabyte; di contro i sensori areali, soprattutto l'interferometro, in fase di criticità a pieno regime di invii, possono essere generati file dell'ordine di 1-2 GB.

La strategia dei messaggi grezzi, archiviati giornalmente, è stata implementata per garantire un sistema di ripristino della base dati in caso di malfunzionamenti del sistema: da parte del CAED è possibile riprodurre esattamente il client dei vari partner e rinviare i messaggi grezzi in una nuova base dati.

6.1.7.C ACQUISIZIONI CAMPIONARIE

I messaggi che sono stati salvati, nello schema *messages*, subiscono il processo di estrazione, di validazione e di archiviazione, dipendenti dal message protocol implementato da ogni singolo nodo coordinatore. Introducendo la tipizzazione dei messaggi (par. 6.1.4.B) che sono di campionamento, come nella maggior parte delle acquisizioni, ma possono essere anche di preelaborazione e di stato, la costruzione delle tabelle dinamiche segue questa procedura:

1. selezione del tipo di messaggio;
2. selezione dei sensori che appartengono al dispositivo;
3. costruzione della tabella dinamica nello schema acquisition secondo il seguente formato:
wWSN_ddispositivo_ttipomessaggio_mtipomisurazione_sensore;
4. se la tabella è già presente sono inseriti i valori registrati, altrimenti viene creata la tabella secondo il formato descritto al punto 3.

Ogni record di ciascuna tabella da informazioni della data e dell'ora di inserimento ed acquisizione e del valore acquisito dal sensore che può essere indicativo di una singola variabile (es. temperatura dell'aria) o di più di una variabile (es. accelerazione lungo x, y e z). La denominazione di ciascuna colonna della/e variabile/i è fatta automaticamente a seguito della prima acquisizione, in accordo con la configurazione delle informazioni del messaggio trasportato.

I dati campionari sono salvati in ciascun record solamente per i sensori puntuali, che rappresentano le situazioni classiche del monitoraggio; nel caso dei sensori areali è stato deciso di salvare solamente il percorso ad un file esterno contenente le acquisizioni nel campo bidimensionale. Questa strategia è stata perseguita per rendere più performante il database, in fase di consultazione, evitando di introdurre tipi di variabili complesse, quali gli array di array.

Classificazione dei codici di errore e registrazione eventi (Event Log System)

Lo scopo della presente attività è quella di definire dei codici di tipo alfanumerico, univoci, che identificano tutte le possibili eccezioni sollevate durante le procedure di acquisizione, validazione dei messaggi e dei dati e dell'invio dei comandi remoti, per la configurazione delle WSN.

Nella presente attività elementare sono descritte tutte le eccezioni possibili che sono state ottenute durante la fase di testing e la loro codifica dettagliata. Tale codifica è stata inserita all'interno della base dati e qualora una delle procedure, descritte nelle attività precedenti, sollevi un'eccezione, deve esserne tenuto traccia, registrandola correttamente in Lewis.

Il codice dell'evento sollevato dall'eccezione/evento segue una notazione standard per quanto riguarda la simbologia di tipo alfabetica. Per ogni codice di evento sono descritti gli argomenti che sono stati registrati all'interno della base dati.

Le eccezioni che riguardano l'*acquisizione* dei dati vengono indicati con la lettera **A** e sono tre: la prima (A00) indica l'acquisizione avvenuta con successo, la seconda (A03) indica l'acquisizione non avvenuta per cause imputabili al sink, il terzo (A09) è relativo alla mancanza di messaggi da parte di un dispositivo imputabili al malfunzionamento del mote.

Le eccezioni che vengono riferite alla rete di trasmissione sono indicate con la lettere **N** e sono anch'esse due: N03 il sink non risponde per possibili problemi nel collegamento o malfunzionamento del dispositivo, N06 il concentratore non comunica con il CAED o i Sink.

Il CAED ha la possibilità di modificare la configurazione della rete ed ogni errore di configurazione deve essere tracciato, il codice evento è preceduto dalla lettera **M**, mentre le eccezioni riscontrate sono pari a 4: M00 il CAED invia al sink della WSN la nuova configurazione della rete, il sink se accetta la nuova configurazione invia al CAED il codice M06 (nuova configurazione accettata), se rifiuta questa configurazione il sink emette il codice M03, mentre se la nuova configurazione fallisce per cause imputabili alla rete il sink invia un codice M09.

Ulteriori codici di evento, invece, sono da riferire alla validazione dei dati (la cui procedura verrà discussa nel dettaglio nella WP 6.2), che sono preceduti dalla lettera **V**. A seguito dell'acquisizione il sistema controlla una serie di parametri del messaggio inviato e nel caso la procedura vada a buon fine codifica la validazione con il codice V00. Nel caso il dato o il gruppo di dati non arrivassero con le specifiche assegnate sono contrassegnate con il codice V03; V09 i dati estratti non sono coerenti con l'indice statistico di riferimento; V13 non è stato possibile validare un sensore poiché il sensore è mancante nella stringa di acquisizione; V14 l'acquisizione ha prodotto un NoData (NaN); V15 le informazioni contenute presente valori anomali; V17 i dati acquisiti non rientrano nel range di valori del sensore; V19 il codice identificativo del messaggio (mac address, intestazione messaggio) è inesistente e non è possibile associare nessun sensore; V21 non è possibile identificare il mote; V23 la data di acquisizione non è stata rilevata, i dati non registrati non possono essere collocati temporalmente nel database.

Una serie di codici alfanumerici, invece, sono da riferire all'esecuzione dei modelli, alle chiamate da parte del CAED agli eseguibili o alle funzioni dei modelli di preannuncio e a tutte le possibili eccezioni che possono essere sollevate durante tali chiamate. I codici di errore sono preceduti dalla lettera **E**.

Nel caso in cui la chiamata del CAED vada a buon fine e gli argomenti di input sono stati inviati correttamente al modello il codice rappresentativo è indicato da E00, mentre se la chiamata del modello è terminata in modo inaspettato o l'esecuzione del modello è fallita i codici identificativi sono rispettivamente E03 e E09. Eseguito correttamente il modello può restituire un collettivo di dati corretto (E12) che deve essere salvato sulla base di dati correttamente (E18); nel caso, invece, in cui il collettivo di dati non può essere salvato sulle BaseDati il codice identificativo è E15, mentre con il codice E21 viene indicata l'eccezione sollevata quando i dati di output del modello non sono validati (presenza di valori anomali o non coerenti con il range dei dati di output del modello).

I livelli di allerta o gli scenari di rischio elaborati dal CAED, a seguito dei valori registrati dai sensori puntuali ed areali e dei modelli di preannuncio, devono essere comunicati al centro di comando e controllo identificato con il termine di "Decision System". I codici di errore sono preceduti dalla lettera **D**. La comunicazione dei livelli di allerta al Decision System può determinare una chiamata corretta D00 e due eccezioni non corrette D03, D09 che identificano rispettivamente la chiamata fallita e l'arresto inaspettato della chiamata.

Ogni eccezione sollevata durante lo svolgimento delle procedure argomentate è registrata all'interno del database ed associata la data di inserimento ed un livello di priorità da dover comunicare ai vari soggetti ed all'operatore del CAED.

Nella tab. 6.1.7 sono presenti tutti i codici descritti nel presente paragrafo, il nome, una breve descrizione e gli argomenti di interesse per classificare il codice

Cod. Evento	Nome	Descrizione	Argomenti
A00	Acquisizione Avvenuta	I dati della WSN sono stati acquisiti e salvati	Sink mittente (IPv4), frequenza acquisizione (minuti), Bytes ricevuti, DateTime acquisizione
A03	Acquisizione non Avvenuta	I dati della WSN non sono pervenuti, segue un tentativo di recupero	Sink mittente, frequenza, Datetime fallimento
A09	Mote scomparso/i	L'acquisizione presenta la mancanza di messaggi da parte di un dispositivo Mote. Possibili cause: malfunzionamento o caduta di batteria (spegnimento)	MAC Address, Mote scomparso, Sink mittente, frequenza acquisizione, DateTime, numero fallimento consecutivo
N03	Connessione al Sink fallita	Il Sink non risponde. Questo può indicare caduta di collegamento (data-link) o Malfunzionamento del dispositivo Sink	Sink mittente, frequenza acquisizione, numero fallimento consecutivo
N06	Connessione al concentratore Fallita	Il concentratore non risponde. Può indicare caduta di collegamento (data-link) o malfunzionamento del dispositivo	Indirizzo concentratore (Ipv4), DateTime, numero fallimento consecutivo
M00	Avvio Modifica Configurazione Sink	Il CAED invia la nuova configurazione della WSN al Sink coordinatore	Sink, comando, argomenti, DateTime

M06	Nuova configurazione Sink Accettata	Il Sink ha acquisito la nuova configurazione correttamente	Sink, comando, argomenti, DateTime
M03	Nuova configurazione Sink Rifiutata		Sink, comando, argomenti, DateTime
M09	Nuova configurazione Sink Fallita		Sink, comando, argomenti, DateTime
V00	Validazione Sensori Corretta	I Dati sono coerenti con le specifiche dei sensori	CallBack: la tupla dei dati viene spuntata con un flag di validazione
V03	Validazione Sensori Errata	I Dati sono incoerenti con le specifiche dei sensori	CallBack: la tupla dei dati viene catalogata con attributo V03
V09	Validazione Storica Incoerente	L'indice statistico con riferimento al periodo di tempo precedente non risulta coerente	CallBack: la tupla dei dati viene catalogata con attributo V06
V13	Sensore mancante	Non è stato possibile trovare e validare un sensore	
V17	Valori variabili sensore non coerenti	Validazione intervallo valori sensore fallita	
V15	Match dei valori è fallito, presenza di valori anomali		
V19	Mac address del messaggio inesistente		
V21	Identificazione Mote Fallita		
V23	Data acquisizione non rilevata		
E00	Chiamata Modello Avvenuta	La chiamata al modello è avvenuta correttamente, gli argomenti sono stati accettati dal modello	Collettivo di dati (Array), Chiamata al modello (eseguibile o funzione)

E03	Esecuzione Modello uscita con codice errore	La chiamata al modello è terminata inaspettatamente	Collettivo di dati (Array), Chiamata al modello (eseguibile o funzione), codice di esecuzione (1 o 2), DateTime
E09	Esecuzione Modello fallita	La chiamata al modello è fallita, errore di esecuzione	Chiamata al modello
E12	Elaborazione modello completato con successo	Il Modello ha restituito un Collettivo di dati	Chiamata al modello, DateTime, Bytes Collettivo dati, Tempo esecuzione
E15	Salvataggio Collettivo dati da Modello fallito	Il collettivo dati non può essere Salvato sulla BaseDati	Codice errore SQL, DateTime, Inserimento dati fallito (SQL)
E18	Salvataggio Collettivo dati Riuscito	Il collettivo dati è stato salvato nella BaseDati	Datetime CallBack: Decision System
E21	Validazione Dati da modello fallita	Validazione Dati da modello fallita	Collettivo dati, Datetime
D00	Chiamata a Decision System	Decision System Avviato Correttamente	DateTime, Dati Modello, Chiamata
D03	Chiamata Decision System Fallita	Avvio Decision System Fallito	DateTime, Chiamata
D09	Decision System arrestato inaspettatamente	Decision System arrestato inaspettatamente	Datetime, Chiamata

Tab. 6.1.7 - Classificazione dei codici di evento all'interno di Lewis.

Acquisizione integrazione e test

L'attività 6.1.9, denominata "acquisizione, integrazione e test", è stata introdotta durante gli ultimi due semestri dell'attività di ricerca per effettuare l'integrazione completa, con Lewis, dei sistemi di monitoraggio e dei modelli di preannuncio.

Le attività che sono state descritte nell'AE 6.1.6.C in parte rientrano anche in questa attività elementare, ma essendo già state descritte dettagliatamente non sono riportate nel presente paragrafo.

L'altra integrazione è relativa ai modelli di preannuncio sviluppati dalla OR 4. I modelli di preannuncio necessitano dei dati acquisiti dai sensori che sono forniti dal CAED, mentre gli output sono inviati dai modelli al CAED e l'informazione utilizzata per valutare lo stato all'interno del sistema di preannuncio.

I modelli di preannuncio, nella visione integrata del progetto, sono trattati come se fossero dei particolari sensori, che usufruiscono delle informazioni delle reti di monitoraggio per determinare la situazione di criticità, sul corpo di frana analizzato, sia al tempo t che durante gli intervalli di tempo successivi (previsione del modello).

I modelli risiedono su macchine dedicate, non residenti all'interno del CAED, e lo scambio di informazioni deve avvenire necessariamente in rete. Dopo una prima proposta con lo staff della OR 4 per lo scambio dei dati attraverso il protocollo stateless HTTP, la scelta più comoda è ricaduta sulla configurazione di uno specifico server che si occupi dello scambio di file (fig. 6.1.21). Il file server non è gestito direttamente dal CAED che non può diagnosticare malfunzionamenti hardware e/o software. L'accesso al sistema e allo spazio di condivisione è garantito da un sistema di autenticazione dotato di username e password.

All'interno del file di configurazione, sia il CAED che le applicazioni possono aggiungere e modificare ed eliminare sia le directory che i file in esse contenute.

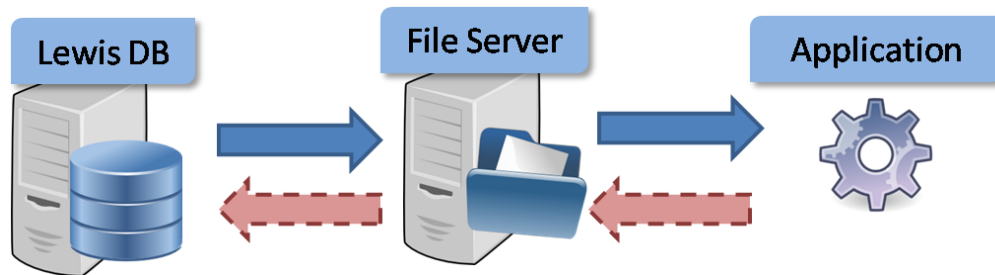


Fig. 6.1.20 – Sistema distribuito in rete per lo scambio dei dati acquisiti dal CAED con i modelli di preannuncio con l’ausilio di un server di condivisione.

L’integrazione dei modelli all’interno di Lewis, così come per i sensori, è effettuato a seguito della compilazione di uno specifico modulo per la classificazione delle caratteristiche operative di ciascuna applicazione (allegato 6.1.9.A): sensori in input, tipologia di funzionamento, output ed eventuali comandi che il CAED può inviare ai modelli, per controllarne lo stato di esecuzione.

I modelli di preannuncio, che sono utilizzati all’interno del progetto, sono principalmente tre: modello areale, puntuale e di propagazione. Ogni modello può essere eseguito su uno o più versanti (unità geomorfologiche) ed ogni singola applicazione è identificata all’interno di Lewis con una istanza (identificativo univoco del singolo modello su una specifica unità geomorfologica).

A seguito della registrazione dei modelli, un applicativo specifico, denominato “processing model”, analizza i singoli modelli istanziati e trasmette le informazioni dei sensori all’interno di una specifica cartella, la cui denominazione tiene conto dell’identificativo univoco del modello, della sua denominazione e dalla UGM/SUGM su cui viene applicato. Nella directory sono presenti tanti file (in formato csv) quanti sono i sensori di interesse per ciascun modello, contenenti la serie delle acquisizioni delle ultime 24 ore, con intervallo di aggregazione a 5 minuti.

L’applicativo, inoltre, valida la serie dei dati, prima di pubblicarla sul file server, e nel caso di assenza di acquisizioni, dovuti ad un malfunzionamento della WSN, è introdotta la dicitura **NA** (Not Available) e l’eccezione gestita direttamente dal modello.

Nella fase di ricerca, la procedura di inserimento dei dati è stata verificata utilizzando una delle stazioni meteorologiche della TDGroup (par. 6.1.5).

6.1.9.A ALLEGATI

RMP013 – Modulo Registrazione Modelli di Preannuncio

Modello*:

*Denominazione del modello

Funzionamento in Continuo: **SI** **NO**

Identificativo Modello

Id	TAS	SUGM

Descrizione attributi

1. Id identificativo numerico dell'istanza del modello operante su una singola unità geomorfologica;
2. TAS tratto autostradale di interesse (A3, A18, A16);
3. SUGM denominazione unità geomorfologica modellata (es. Mancarelli, Fiego, ...).

Sensori di Input

Sensore	Unità di misura	<input type="checkbox"/> t acquisizione* (default)	<input type="checkbox"/> t acquisizione* (minimo)	Eventuale Descrizione

Descrizione attributi

1. Sensore (precipitazione, temperatura dell'aria, suzione, ...);
2. Unità di misura (mm, °C, kPa, ...);
3. Dt acquisizione in minuti (passo temporale di acquisizione del singolo sensore),;
4. Dt acquisizione minimo in minuti (passo temporale di acquisizione minimo per ciascun sensore);
5. Eventuale descrizione.

Nel caso non abbia importanza questa informazione scrivere (CAMPIONAMENTO) in riferimento al DT di acquisizione della rete.

Output del Modello

Nome	Formato	Unità di Misura	Dt output	Dimensione del file (kB)	Eventuale Descrizione

Descrizione attributi

1. Nome denominazione dell'output del modello (matrice spostamenti, fattore sicurezza, matrice di suzione, ...);
2. Formato tipologia del formato di output del modello (raster, vettoriale, singola informazione, stringa, ...);
3. Unità di misura dell'output del file (mm, adimensionale, kPa, ...);
4. Dt output intervallo temporale di uscita dei modelli, in minuti, che eventualmente lavorano in continuo;
5. Dimensione approssimativa del file output del modello in kB (semplice stima);
6. Eventuale descrizione.

Preelaborazioni dell'output

Breve descrizione su un'eventuale preelaborazione dell'output.

Invio dei comandi da parte del CAED per avviare un modello e controllarne lo stato

Codice comando	Comando	Argomenti	Codice risposta del Modello	Descrizione

Descrizione attributi

1. Codice comando valore progressivo (1, 2, ..., n), identifica un numero corrispondente al comando;
2. Comando denominazione del comando che invia il CAED ad un modello;
3. Argomenti eventuali argomenti da dover inviare insieme al comando separati da virgola, es: con comando "data" potremmo avere un argomento "2013-09-18".
Altro esempio: con comando delta_t potremmo avere come argomento "20";
4. Codice risposta del Modello : codice di presa in consegna, es:
 - a) 200: presa in consegna;
 - b) 500: errore interno;
 - c) 404: comando non trovato;
 - d) altro, da definire in base alle eccezioni interne gestite dalla implementazione del codice di calcolo.
5. Descrizione del comando inviato dal CAED.

ALLEGATI

Eventuali formati di input ed output che sono utilizzati dai modelli.

ALLEGATO A

Moduli registrazione dispositivi



Centro di Acquisizione Elaborazione Dati

Staff di competenza

Gruppo di lavoro o Società di appartenenza	TD Group SpA
Referente tecnico, nome cognome	Marco De Marinis
Data compilazione	21/10/2013
Email di riferimento	m.demarinis@tdnet.it
Contatto telefonico	+39 349 2108617

Scheda da replicare per ogni dispositivo

tipo dispositivo	Gateway + End Node
identificativo univoco [1]	<i>TDgroup_meteo_1</i>
prodotto da [2]	TDGroup SpA
nome modello	METEO_STATION_STANDALONE
versione software/firmware/middleware	1.0.2332
modelli sensori montati [3]	DAVIS Vantage Pro2, Decagon 5TE
coordinate di posizionamento [4]	<i>CUSTOMER SPECIFIC</i>
operante in WSN (nome dato alla rete wsn)	LEW_001_DEMO

SENSORE

prodotto da	DAVIS Instruments (http://www.davisnet.com)
montato su dispositivo [identificativo univoco]	METEO_STATION (all models)
tipo di misurazione [5]	Temperatura, Umidità [, Temperatura di Rugiada] (derived measure)
modello	Temperature and Humidity 6382 (Alias Sensirion SHT15)
unità di misura [6]	°C , RH , °C
variabili della misurazione (nomi) [7]	'wht_temp:', 'wht_humi:', 'wht_dewpoint:'
range acquisizione [8]	wht_temp: -40.0 ... 123.8 °C wht_humi: +0.0 ... 100.0 % wht_dewpoint: -40.0 ... 123.8 °C 'fail' if error (each measures independent)
Link web (url) datasheet	http://www.davisnet.com/weather/products/wx_product_docs.asp?pnun=06382

SENSORE

prodotto da	DAVIS Instruments
montato su dispositivo [identificativo univoco]	METEO_STATION (all models)
tipo di misurazione [5]	Radiazione solare totale
modello	Solar Sensor (6450)
unità di misura [6]	W/m ²
variabili della misurazione (nomi) [7]	'wht_rad:'
range acquisizione [8]	wht_rad: 0.0 ... 1800,0 'fail' if error
Link web (url) datasheet	http://www.davisnet.com/weather/products/wx_product_docs.asp?pnum=06450

SENSORE

prodotto da	DAVIS Instruments
montato su dispositivo [identificativo univoco]	METEO_STATION (all models)
tipo di misurazione [5]	Velocità del vento
modello	Anemometer (7911)
unità di misura [6]	m/s, m/s
variabili della misurazione (nomi) [7]	'wht_speed:', 'wht_gust:'
range acquisizione [8]	wht_speed: 0.5 ... 89.0 m/s (acc. ±1 m/s) wht_gust: 0.5 ... 89.0 m/s (acc. ±1 m/s) 'fail' if error (each measures independent)
Link web (url) datasheet	http://www.davisnet.com/weather/products/wx_product_docs.asp?pnum=07911

SENSORE

prodotto da	DAVIS Instruments
montato su dispositivo [identificativo univoco]	METEO_STATION (all models)
tipo di misurazione [5]	Direzione del vento
modello	Anemometer (7911)
unità di misura [6]	degrees
variabili della misurazione (nomi) [7]	'wht_direction:'
range acquisizione [8]	wht_direction: 0.0 ... 359.9deg. (acc. ±3°) 'fail' if error (each measures independent)
Link web (url) datasheet	http://www.davisnet.com/weather/products/wx_product_docs.asp?pnum=07911

SENSORE

prodotto da	DAVIS Instruments
montato su dispositivo [identificativo univoco]	METEO_STATION (all models)
tipo di misurazione [5]	Pioggia accumulata, pioggia oraria
modello	Rain Collector (7852)
unità di misura [6]	mm (h ₂ O)/m ² , mm (h ₂ O)/m ² /h
variabili della misurazione (nomi) [7]	'wht_rain:', 'wht_rainh:'
range acquisizione [8]	wht_rain: +0.0 .. 999.9 mm wht_rainh: +0.0 .. 999.9 mm 'fail' if error (each measures independent)
Link web (url) datasheet	http://www.davisnet.com/weather/products/wx_product_docs.asp?pnum=07852

SENSORE

prodotto da	Decagon
montato su dispositivo [identificativo univoco]	METEO_STATION_STANDALONE
tipo di misurazione [5]	Permittivity, Conductivity, Temperature, VWC (derived measure)
modello	5TE
unità di misura [6]	-, dS/m, °C, -
variabili della misurazione (nomi) [7]	'soil_perm:', 'soil_cond:', 'soil_temp:', 'soil_vwc:'
range acquisizione [8]	soil_perm: +1.00 ... 81.88 soil_cond: +0.0 ... 23.0 dS/m soil_temp: -40.0 .. 50.0 °C soil_vwc: +0.0 .. 1.0 m ³ /m ³ 'fail' if error (each measures independent)
Link web (url) datasheet	http://manuals.decagon.com/Manuals/13509_5TE_Web.pdf

SENSORE (STATO MODULO GSM/GPRS)

prodotto da	SIMCOM
montato su dispositivo [identificativo univoco]	METEO_STATION_STANDALONE
tipo di misurazione [5]	Stato interno, batteria, link radio
modello	SIM340
unità di misura [6]	s, -, -, -, %, mV, -
variabili della misurazione (nomi) [7]	's5p5_gsm_period:' 's5p5_gsm_rssi:' 's5p5_gsm_ber:' 's5p5_gsm_bcs:' 's5p5_gsm_bcl:' 's5p5_gsm_batt:' 's5p5_gsm_err:'
range acquisizione [8]	s5p5_gsm_period: 1... 65536 s5p5_gsm_rssi: 0 ... 99 s5p5_gsm_ber: 0 ... 99 s5p5_gsm_bcs: 0 1 2 s5p5_gsm_bcl: 0 ... 99 % s5p5_gsm_batt: 0 ... 4500 mV s5p5_gsm_err: 255 if error
Link web (url) datasheet	na

SENSORE (STATO MODULO S5P5)

prodotto da	TD Group SpA
montato su dispositivo [identificativo univoco]	METEO_STATION_STANDALONE
tipo di misurazione [5]	Stato interno batteria
modello	S5P5
unità di misura [6]	mV
variabili della misurazione (nomi) [7]	's5p5_vbatt:'
range acquisizione [8]	s5p5_vbatt:: 0 ... 4500 mV
Link web (url) datasheet	na

- [1] preferibile MAC address, lunghezza massima 16 caratteri
[2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
[3] es, separati da virgola: LIS3LV02DL, SHT75.
[4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
[5] es. proper acceleration, rainfall, atmospheric temperature
[6] es. percent (%), gradi celsius, millimeters, voltage
[7] es. "x,y,z" oppure "temp", oppure "wind_vel"
[8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo: -20;80
se indicato nel seguente formato indica un set di valori: -20, 0, 20, 32.26, 52.19, 84.44
nel caso di vettore di valori indicare lunghezza e range: [-20;80]*NumeroCampioni

ESEMPI DI MESSAGGI:

```
>> {'message': '[TDgroup_meteo_1 13-11-10 14:11:24 #sample] -wht_rad:+1234.010009765625 -
wht_temp:+23.450000762939453125,wht_humi:+56.779998779296875,wht_dewpoint:+12.34000015258789
0625;'}
OUT: 200 - Saved [ d6d22a82bf75da8441e6ddd0ce9793a6 ]
>> {'message': '[TDgroup_meteo_1 13-11-10 14:11:24 #sample] -wht_direction:+234.55999755859375 -
wht_speed:+23.450000762939453125,wht_gust:+34.560001373291015625;'}
OUT: 200 - Saved [ f4f99038f099cbf663575f689504c3ff ]
>> {'message': '[TDgroup_meteo_1 13-11-10 14:11:24 #sample] -
wht_rain:+1234.010009765625,wht_rainh:+0.11999999237060546875;'}
OUT: 200 - Saved [ a5a4d64b7fed5cdf1040770d83429fd1 ]
>> {'message': '[TDgroup_meteo_1 13-11-10 14:11:24 #sample] -
soil_perm:+12.340000152587890625,soil_cond:+23.450000762939453125,soil_temp:+12.34000015258789
0625,soil_vwc:+0.11999999237060546875;'}
OUT: 200 - Saved [ 848e5c1cb3e2c47f70af593caf42f7fa ]
>> {'message': '[TDgroup_meteo_1 13-11-10 14:11:24 #sample] -
s5p5_vbatt:+2.3450000286102294921875 -
s5p5_gsm_period:120,s5p5_gsm_rssi:17,s5p5_gsm_ber:0,s5p5_gsm_bcs:0,s5p5_gsm_bcl:87,s5p5_gsm_b
att:4567,s5p5_gsm_err:0;'}
OUT: 200 - Saved [ d8bac4c54e28bc783bace755abb0d248 ]
```

Come si può notare, i campi sono raggruppati per settore in modo da soddisfare il requisito attuale di non superare i 256 caratteri per messaggio. Probabilmente tale suddivisione verrà ulteriormente rivista in modo da portare la lunghezza sotto i 120 caratteri. Questo è un requisito specifico del protocollo zStack della Texas (solo in caso si incontrino problemi durante la formazione dei messaggi per i nuovi sensori).

I messaggi di servizio sono stati assimilati alle misure. Si richiede che vengano conservati al fine di poter analizzare in dettaglio l'andamento dei segnali e quindi risalire alle cause di un eventuale guasto. In particolare:

RSSI	s5p5_gsm_rssi
GSM Vbatt	s5p5_gsm_batt
S5P5 Vbatt	s5p5_vbatt

Da notare che i Float sono espressione diretta (fino all'ultimo digit) di Float a 32 bit (IEEE 754). L'implementazione attuale non effettua l'arrotondamento od il troncamento.

COMANDI:

1. *Allineamento temporale:*

```
OUT: 205 - { 'id': 1, 'destination': 'TDgroup_meteo_1', 'command': 14, 'arguments': [ 1384092684 ] }  
>> { 'id': 1, 'md5': '817741e67931b173ce6986801553888b' }
```

Da notare che il tempo UTC è in formato UNIX timestamp = 32bit integer (seconds since Jan 1st 1970). Il massimo è: 2147483647 pari al 19 gennaio 2038 03:14:08 UTC

2. *Cambio Temporizzazione:*

```
OUT: 205 - { 'id': 1, 'destination': 'TDgroup_meteo_1', 'command': 15, 'arguments': [ 1 120 ] }  
>> { 'id': 1, 'md5': '817741e67931b173ce6986801553888b' }
```

Il primo termine rappresenta il tipo di messaggio. Nel caso sopra esposto:

1 = Radiazione solare + (Temperatura & Umidità)

2 = Vento

3 = Pioggia

4 = Suolo

5 = 'Service message'

Il secondo termine rappresenta il numero di secondi: **1...65535**

Il valore ha un minimo dovuto al tipo di sensore (tempo di misura). In caso il valore richiesto sia inferiore al minimo, il sistema imposterà automaticamente tale valore.

Da notare che la temporizzazione è relativa ad ogni singolo messaggio: se due sensori sono raggruppati nello stesso messaggio, conseguentemente avranno stessa temporizzazione.

Non c'è risposta al comando se non l'ACK con MD5

ALLARMI:

Ad oggi non sono stati richiesti allarmi specifici. Se richiesto, verranno implementati previa approvazione tecnica.

MESSAGGI DI LOG:

```
>> { 'message': '[TDgroup_meteo_1 13-11-10 14:11:24 #status] rQx5DT2/heKrs7kssGte5g==;' }
```

```
OUT: 200 - Saved [ d8bac4c54e28bc783bace755abb0d248 ]
```

La signature (BASE64) è un packet binario di lunghezza variabile.

Si prega di loggare tali messaggi al fine della ricostruzione degli eventi critici del nodo.

Il timestamp è relativo al tempo di invio del messaggio e non è correlato al log stesso.

Centro di Acquisizione Elaborazione Dati

Staff di competenza

Gruppo di lavoro o Società di appartenenza	STRAGO SpA
Referente tecnico, nome cognome	Maddalena Autiero
Data compilazione	08_10_2013
Email di riferimento	maddalena.autierostrago.it
Contatto telefonico	0815240667

Scheda da replicare per ogni dispositivo

tipo dispositivo	Master
identificativo univoco [1]	S/N SM14S0A812
prodotto da [2]	STRAGO SpA
nome modello	SMAMID v1.4B
versione software/firmware/middleware	Firmware v 1.1.5 strago_earlywarning
modelli sensori montati [3]	strago_earlywarning
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	SWAN

Scheda da replicare per ogni dispositivo

tipo dispositivo	slave
identificativo univoco [1]	S/N SM14S0CF12
prodotto da [2]	STRAGO SpA
nome modello	SMAMID v1.4B
versione software/firmware/middleware	Firmware v 1.1.5
modelli sensori montati [3]	STM LIS3LV02DQ
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	SWAN

Scheda da replicare per ogni dispositivo

tipo dispositivo	slave
identificativo univoco [1]	S/N SM14S0CE12
prodotto da [2]	STRAGO SpA
nome modello	SMAMID v1.4B
versione software/firmware/middleware	Firmware v 1.1.5
modelli sensori montati [3]	STM LIS3LV02DQ
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	SWAN

Scheda da replicare per ogni dispositivo

tipo dispositivo	slave
identificativo univoco [1]	S/N SM14S0A212
prodotto da [2]	STRAGO SpA
nome modello	SMAMID v1.4B
versione software/firmware/middleware	Firmware v 1.1.5
modelli sensori montati [3]	STM LIS3LV02DQ
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	SWAN

Scheda da replicare per ogni dispositivo

tipo dispositivo	slave
identificativo univoco [1]	S/N SM14S0A612
prodotto da [2]	STRAGO SpA
nome modello	SMAMID v1.4B
versione software/firmware/middleware	Firmware v 1.1.5
modelli sensori montati [3]	STM LIS3LV02DQ
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	SWAN

Sensore accelerometrico

prodotto da	STMicroelectronics
montato su dispositivo [identificativo univoco]	S/N SMAMID
tipo di misurazione [5]	Tre componenti ortogonali dell'accelerazione
modello	LIS3LV02DQ mg
unità di misura [6]	mg
variabili della misurazione (nomi) [7]	acc_x, acc_y, acc_z
range acquisizione [8]	-2000,2000
Link web (url) datasheet	http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00047926.pdf

Sensore temperatura

prodotto da	Microchip
montato su dispositivo [identificativo univoco]	S/N SMAMID
tipo di misurazione [5]	Temperatura
modello	MCP9700A
unità di misura [6]	°C
variabili della misurazione (nomi) [7]	wht_temp
range acquisizione [8]	-40 - 125

Sensore tensione

prodotto da	Microchip
montato su dispositivo [identificativo univoco]	S/N SMAMID
tipo di misurazione [5]	Tensione sulla batteria di alimentazione
modello	La tensione viene letta a bordo del microcontrollore PIC18LF8723
unità di misura [6]	V
variabili della misurazione (nomi) [7]	bat_v
range acquisizione [8]	0-10.6 ⁵

⁵ La tensione minima a bordo scheda è 2.8 V mentre la massima è 10 V.

Sensore software Inclinazioni

prodotto da	STRAGO
tipo di misurazione [5]	Inclinazioni
unità di misura [6]	°
variabili della misurazione (nomi) [7]	incl_x, incl_y, incl_Z ⁶
range acquisizione [8]	-180, +180

Sensore software System_status

prodotto da	STRAGO Software numeric comparison
tipo di misurazione [5]	Software numeric comparison (Stato del sistema rilievo franoso)
modello	strago_earlywarning
unità di misura [6]	[n] strago_earlywarning ew_stat
variabili della misurazione (nomi) [7]	ew_stat
range acquisizione [8]	0-3

Sensore software Sensor_status

prodotto da	STRAGO
tipo di misurazione [5]	Stato del singolo sensore accelerometrico
unità di misura [6]	[n]
variabili della misurazione (nomi) [7]	sensor_status
range acquisizione [8]	0-2

⁶ Vengono forniti i valori degli angoli che la verticale forma con gli assi accelerometrici. Ad es per l'asse x: $Inclinazione_x = \arccos \left(\frac{Acc_x}{\sqrt{Acc_x^2 + Acc_y^2 + Acc_z^2}} \right)$, essendo Acc_x, Acc_y e Acc_z le componenti dell'accelerazione lungo le direzioni x, y e z definite dagli assi del sensore accelerometrico

Sensore software Acquisition_status

prodotto da	STRAGO
tipo di misurazione [5]	Esito del download dei dati accelerometrici
unità di misura [6]	[n]
variabili della misurazione (nomi) [7]	acquisition_status
range acquisizione [8]	0,1

- [1] preferibile MAC address, lunghezza massima 16 caratteri
- [2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
- [3] es, separati da virgola: LIS3LV02DL, SHT75.
- [4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
- [5] es. proper acceleration, rainfall, atmospheric temperature
- [6] es. percent (%), gradi celsius, millimeters, voltage
- [7] es. "x,y,z" oppure "temp", oppure "wind_vel"
- [8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo:
-20;80
se indicato nel seguente formato indica un set di valori:
-20, 0, 20, 32.26, 52.19, 84.44
nel caso di vettore di valori indicare lunghezza e range
[-20;80]*NumeroCampioni

Centro di Acquisizione Elaborazione Dati

Staff di competenza

Gruppo di lavoro o Società di appartenenza	Gruppo di geomatica
Referente tecnico, nome cognome	Artese Giuseppe
Data compilazione	21/05/2014
Email di riferimento	g.artese@unical.it
Contatto telefonico	0984496768

Scheda da replicare per ogni dispositivo

tipo dispositivo	Coordinatore
identificativo univoco [1]	b827eb1a4366
prodotto da [2]	Laboratorio di geomatica - UNICAL
nome modello	OptNivGNSS_c
versione software/firmware/middleware	1.0
modelli sensori montati [3]	1
coordinate di posizionamento [4]	Cartesiane Geocentriche
operante in WSN (nome dato alla rete wsn)	Test1

Scheda da replicare per ogni sensore

prodotto da	Laboratorio di geomatica -UNICAL
montato su dispositivo [identificativo univoco]	b827eb1a4366
tipo di misurazione [5]	Inclinazione
modello	OptNivGNSS_ed
unità di misura [6]	Gradi sessadecimali
variabili della misurazione (nomi) [7]	InclYaw,InclRoll
range acquisizione [8]	[-0.05:0.05], [-0.05:0.05]
Link web (url) datasheet	

Scheda da replicare per ogni sensore

prodotto da	Laboratorio di geomatica -UNICAL
montato su dispositivo [identificativo univoco]	b827eb1a4366
tipo di misurazione [5]	Posizionamento GNSS
modello	OptNivGNSS_ed
unità di misura [6]	metri
variabili della misurazione (nomi) [7]	X,Y,Z
range acquisizione [8]	[-6.379.400,00 6.379.400,00], [-6.379.400,00 6.379.400,00] [-6.379.400,00 6.379.400,00]
Link web (url) datasheet	

- [1] preferibile MAC address, lunghezza massima 16 caratteri
- [2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
- [3] es, separati da virgola: LIS3LV02DL, SHT75.
- [4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
- [5] es. proper acceleracion, rainfall, atmospheric temperature
- [6] es. percent (%), gradi celsius, millimeters, voltage
- [7] es. "x,y,z" oppure "temp", oppure "wind_vel"
- [8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo: -20;80
se indicato nel seguente formato indica un set di valori: -20, 0, 20, 32.26, 52.19, 84.44
nel caso di vettore di valori indicare lunghezza e range: [-20;80]*NumeroCampioni

Centro di Acquisizione Elaborazione Dati

Staff di competenza

Gruppo di lavoro o Società di appartenenza	Dipartimento Scienze della terra, UniFI
Referente tecnico, nome cognome	Emanuele Inrieri
Data compilazione	04/11/2013
Email di riferimento	emanuele.inrieri@gmail.com
Contatto telefonico	055 275 7784

Scheda da replicare per ogni dispositivo

tipo dispositivo	Sink (coordinatore + gateway)
identificativo univoco [1]	00-24-E8-BE-53-55
prodotto da [2]	PC
nome modello	PC
versione software/firmware/middleware	LEARNET_FRAMEWORK_MW_v080b
modelli sensori montati [3]	GBInSAR
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	

Scheda da replicare per ogni sensore

prodotto da	ELLEGI Lisalab
montato su dispositivo [identificativo univoco]	00-24-E8-BE-53-55
tipo di misurazione [5]	Displacement map
modello	GBInSAR
unità di misura [6]	Millimeters
variabili della misurazione (nomi) [7]	ctrl[1...]
range acquisizione [8]	-5;+5
Link web (url) datasheet	

Scheda da replicare per ogni sensore

prodotto da	ELLEGI Lisalab
montato su dispositivo [identificativo univoco]	00-24-E8-BE-53-55
tipo di misurazione [5]	Displacement map
modello	GBInSAR
unità di misura [6]	Millimeters
variabili della misurazione (nomi) [7]	itf
range acquisizione [8]	
Link web (url) datasheet	

Scheda da replicare per ogni sensore

prodotto da	ELLEGI Lisalab
montato su dispositivo [identificativo univoco]	00-24-E8-BE-53-55
tipo di misurazione [5]	Displacement map 8h
modello	GBInSAR
unità di misura [6]	Millimeters
variabili della misurazione (nomi) [7]	itf_8
range acquisizione [8]	[-5;+5]* 1001
Link web (url) datasheet	

Scheda da replicare per ogni sensore

prodotto da	ELLEGI Lisalab
montato su dispositivo [identificativo univoco]	00-24-E8-BE-53-55
tipo di misurazione [5]	Displacement map 24h
modello	GBInSAR
unità di misura [6]	Millimeters
variabili della misurazione (nomi) [7]	itf_24
range acquisizione [8]	[-5;+5]* 1001
Link web (url) datasheet	

- [1] preferibile MAC address, lunghezza massima 16 caratteri
- [2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
- [3] es, separati da virgola: LIS3LV02DL, SHT75.
- [4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
- [5] es. proper acceleracion, rainfall, atmospheric temperature
- [6] es. percent (%), gradi celsius, millimeters, voltage
- [7] es. "x,y,z" oppure "temp", oppure "wind_vel"
- [8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo:
-20;80
se indicato nel seguente formato indica un set di valori:
-20, 0, 20, 32.26, 52.19, 84.44
nel caso di vettore di valori indicare lunghezza e range
[-20;80]*NumeroCampioni

Centro di Acquisizione Elaborazione Dati

Staff di competenza

Gruppo di lavoro o Società di appartenenza	μWaveLab – Unical
Referente tecnico, nome cognome	Giuseppe Di Massa
Data compilazione	16/11/2013
Email di riferimento	dimassa@deis.unical.it
Contatto telefonico	0984494700

Scheda da replicare per ogni dispositivo

tipo dispositivo	Embedded PC
identificativo univoco [1]	PNA12105101
prodotto da [2]	Adlinktech
nome modello	MXE 5000
versione software/firmware/middleware	LEWARNET_v084
modelli sensori montati [3]	GBInSAR
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	

Scheda da replicare per ogni sensore

prodotto da	μWaveLab – Unical
montato su dispositivo [identificativo univoco]	PNA12105101
tipo di misurazione [5]	Distanza
Modello	Scatterometro
unità di misura [6]	[m; V/m]
variabili della misurazione (nomi) [7]	SEC_N [X_range; Y _{k/N} _HRRP] k = 1:N dove N è il n° di settori
range acquisizione [8]	Float 0; 1000 – Float 0;1
Link web (url) datasheet	N.A.

- [1] preferibile MAC address, lunghezza massima 16 caratteri
- [2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
- [3] es, separati da virgola: LIS3LV02DL, SHT75.
- [4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
- [5] es. proper acceleracion, rainfall, atmospheric temperature
- [6] es. percent (%), gradi celsius, millimeters, voltage
- [7] es. "x,y,z" oppure "temp", oppure "wind_vel"
- [8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo:
-20;80
se indicato nel seguente formato indica un set di valori:
-20, 0, 20, 32.26, 52.19, 84.44
nel caso di vettore di valori indicare lunghezza e range
[-20;80]*NumeroCampioni

Centro di Acquisizione Elaborazione Dati

Staff di competenza

Gruppo di lavoro o Società di appartenenza	μWaveLab – Unical
Referente tecnico, nome cognome	Giuseppe Di Massa
Data compilazione	16/11/2013
Email di riferimento	dimassa@deis.unical.it
Contatto telefonico	0984494700

Scheda da replicare per ogni dispositivo

tipo dispositivo	Embedded PC
identificativo univoco [1]	EAR12VCNW
prodotto da [2]	Adlinktech
nome modello	MXE 5000
versione software/firmware/middleware	LEWARNET_v084
modelli sensori montati [3]	GBInSAR
coordinate di posizionamento [4]	
operante in WSN (nome dato alla rete wsn)	

Scheda da replicare per ogni sensore

prodotto da	μWaveLab – Unical
montato su dispositivo [identificativo univoco]	EAR12VCNW
tipo di misurazione [5]	Distanza
Modello	SDRadar
unità di misura [6]	[m; V/mHz]
variabili della misurazione (nomi) [7]	SEC_N [X_range; Y _{k/N} _HRRP] k = 1:N dove N è il n° di settori
range acquisizione [8]	Float 0; 1000 – Float 0;2
Link web (url) datasheet	N.A.

- [1] preferibile MAC address, lunghezza massima 16 caratteri
- [2] es. Campbell, ADLINK. Se è un prototipo di laboratorio specificare Staff
- [3] es, separati da virgola: LIS3LV02DL, SHT75.
- [4] Geografiche (epsg:4326 long/lat) o Cartesiane (epsg:32633 est/nord)
- [5] es. proper acceleracion, rainfall, atmospheric temperature
- [6] es. percent (%), gradi celsius, millimeters, voltage
- [7] es. "x,y,z" oppure "temp", oppure "wind_vel"
- [8] es. tipo intero o float; se indicato nel seguente formato indica minimo-massimo:
-20;80
se indicato nel seguente formato indica un set di valori:
-20, 0, 20, 32.26, 52.19, 84.44
nel caso di vettore di valori indicare lunghezza e range
[-20;80]*NumeroCampioni

ALLEGATO B

Moduli registrazione modelli di preannuncio

Modello*: GEOTop

*Denominazione del modello

Funzionamento in Continuo: **SI****Identificativo Modello**

Id	TAS	SUGM
GEOtop01	A3	Mancarelli
GEOtop02	A16	

Descrizione attributi

- Id identificativo numerico dell'istanza del modello operante su una singola unità geomorfologica;
- TAS tratto autostradale di interesse (A3, A18, A16);
- SUGM denominazione unità geomorfologica modellata (es. Mancarelli, Fiego, ...).

Sensori di Input

Sensore	Unità di misura	□t acquisizione* (default)	□t acquisizione* (minimo)	Eventuale Descrizione
Precipitazione	mm	Orario	orario	Necessario per esecuzione modello
Temperatura dell'aria	C	Orario	Orario	Necessario per esecuzione modello
Radiazione Solare	W/m2	Orario	Orario	Necessario per esecuzione modello
Umidità dell'aria	%	Orario	Orario	Necessario per esecuzione modello
Contenuto d'acqua	[-]	Orario	Orario	Necessario per verifica del modello
Suzione	mm	Orario	Orario	Necessario per verifica del modello

Descrizione attributi

1. Sensore (precipitazione, temperatura dell'aria, suzione, ...);
2. Unità di misura (mm, °C, kPa, ...);
3. □t acquisizione in minuti (passo temporale di acquisizione del singolo sensore),;
4. □t acquisizione minimo in minuti (passo temporale di acquisizione minimo per ciascun sensore);
5. Eventuale descrizione.

* nel caso non abbia importanza questa informazione scrivere (CAMPIONAMENTO) in riferimento al □T di acquisizione della rete.

Output del Modello

Nome	Formato	Unità di Misura	□t output	Dimensione del file (kB)	Eventuale Descrizione
Mappa FS	.asc	[-]	Orario	100-1000 kB	
Mappe di suzione	.asc	mm		100-1000 kB	Se necessario per il CAED
Mappe di contenuto d'acqua	.asc	[-]		100-1000 kB	Se necessario per il CAED

Descrizione attributi

1. Nome denominazione dell'output del modello (matrice spostamenti, fattore sicurezza, matrice di suzione, ...);
2. Formato tipologia del formato di output del modello (raster, vettoriale, singola informazione, stringa, ...);
3. Unità di misura dell'output del file (mm, adimensionale, kPa, ...);
4. □t output intervallo temporale di uscita dei modelli, in minuti, che eventualmente lavorano in continuo;
5. Dimensione approssimativa del file output del modello in kB (semplice stima);
6. Eventuale descrizione.

Preelaborazioni dell'output

Breve descrizione su un'eventuale preelaborazione dell'output.

Invio dei comandi da parte del CAED per avviare un modello e controllarne lo stato

Da definire in modo congiunto tra responsabile scientifico, CAED e responsabili di OR

Codice comando	Comando	Argomenti	Codice risposta del Modello	Descrizione

Descrizione attributi

- Codice comando valore progressivo (1, 2, ..., n), identifica un numero corrispondente al comando;
- Comando denominazione del comando che invia il CAED ad un modello;
- Argomenti eventuali argomenti da dover inviare insieme al comando separati da virgola, es: con comando "data" potremmo avere un argomento "2013-09-18". Altro esempio: con comando delta_t potremmo avere come argomento "20";
- Codice risposta del Modello : codice di presa in consegna, es:

- 200: presa in consegna;
- 500: errore interno;
- 404: comando non trovato;
- altro, da definire in base alle eccezioni interne gestite dalla implementazione del codice di calcolo
- Descrizione del comando inviato dal CAED.

ALLEGATI

- 1) File meteo.txt
- 2) File di output: watercontent.asc

Modello*: SUSHI

*Denominazione del modello

Funzionamento in Continuo: **SI**

Identificativo Modello

Id	TAS	SUGM
	A3	Fiego

Descrizione attributi

- Id identificativo numerico dell'istanza del modello operante su una singola unità geomorfologica;
- TAS tratto autostradale di interesse (A3, A18, A16);
- SUGM denominazione unità geomorfologica modellata (es. Mancarelli, Fiego, ...).

Sensori di Input

Sensore	Unità di misura	□t acquisizione* (default)	□t acquisizione* (minimo)	Eventuale Descrizione
Precipitazione	mm	15 min	10 min	Unico dato di input necessario per l'esecuzione Dato utilizzato per validare l'output del modello e quindi non necessariamente richiesto ad intervalli temporali prestabiliti
Contenuto d'acqua	-			
Suzione	m			
Spostamenti	mm			
Altezza di falda	m			
Precipitazione	mm	15 min	10 min	

Descrizione attributi

6. Sensore (precipitazione, temperatura dell'aria, suzione, ...);
7. Unità di misura (mm, °C, kPa, ...);
8. Δt acquisizione in minuti (passo temporale di acquisizione del singolo sensore);
9. Δt acquisizione minimo in minuti (passo temporale di acquisizione minimo per ciascun sensore);
10. Eventuale descrizione.

* nel caso non abbia importanza questa informazione scrivere (CAMPIONAMENTO) in riferimento al ΔT di acquisizione della rete.

Output del Modello

Nome	Formato	Unità di Misura	Δt output	Dimensione del file (kB)	Eventuale Descrizione
Fattore di Sicurezza	.txt	-	15 min	1--100 kB	
Mappa di suzioni	.txt	m	15 min	100-1000 kB	Se richiesto dal CAED
Mappa di spostamenti	.txt	mm	15 min	100-1000 kB	Se richiesto dal CAED

Descrizione attributi

7. Nome denominazione dell'output del modello (matrice spostamenti, fattore sicurezza, matrice di suzione, ...);
8. Formato tipologia del formato di output del modello (raster, vettoriale, singola informazione, stringa, ...);
9. Unità di misura dell'output del file (mm, adimensionale, kPa, ...);
10. Δt output intervallo temporale di uscita dei modelli, in minuti, che eventualmente lavorano in continuo;
11. Dimensione approssimativa del file output del modello in kB (semplice stima);
12. Eventuale descrizione.

Preelaborazioni dell'output

Breve descrizione su un'eventuale preelaborazione dell'output.

Invio dei comandi da parte del CAED per avviare un modello e controllarne lo stato

Da definire in modo congiunto tra responsabile scientifico, CAED e responsabili di OR

Codice comando	Comando	Argomenti	Codice risposta del Modello	Descrizione

Descrizione attributi

- Codice comando valore progressivo (1, 2, ..., n), identifica un numero corrispondente al comando;
- Comando denominazione del comando che invia il CAED ad un modello;
- Argomenti eventuali argomenti da dover inviare insieme al comando separati da virgola, es: con comando "data" potremmo avere un argomento "2013-09-18". Altro esempio: con comando delta_t potremmo avere come argomento "20";
- Codice risposta del Modello : codice di presa in consegna, es:
 - 200: presa in consegna;
 - 500: errore interno;
 - 404: comando non trovato;
 - altro, da definire in base alle eccezioni interne gestite dalla implementazione del codice di calcolo
- Descrizione del comando inviato dal CAED.

ALLEGATI

In allegato bozza del file di input e di output (solo relativo al Fattore di Sicurezza).

NOTE

Si tratta di un livello basilare di definizione degli input e output che sarà aggiornato in fase di sperimentazione del modello in funzione delle necessità di volta in volta riscontrate

