Relazione del progetto di Informatica III



Laurea Magistrale in Ingegneria Informatica

I Anno

A.A. 2021/2022

Modulo di Progettazione e

Algoritmi

Prof. Patrizia Scandurra

cod. 38068

Wasim Essbai

Matteo Locatelli

Nicola Zambelli 1053015

Sommario

[Introduzione 5](#_Toc96338507)

[Obiettivo 6](#_Toc96338508)

[W3C 7](#_Toc96338509)

[LoRaWan 8](#_Toc96338510)

[MAPE-K loop 10](#_Toc96338511)

[Iterazione 0 11](#_Toc96338512)

[Analisi dei requisiti 12](#_Toc96338513)

[R0 – Azioni di riconfigurazione per la minimizzazione dell’overhead e allungamento del ciclo di vita delle sentinelle: 12](#_Toc96338514)

[R1 – Azioni di auto-diagnosi manutentiva di risoluzione autonoma dei guasti: 12](#_Toc96338515)

[REQUISITO R0: 13](#_Toc96338516)

[REQUISITI R1: 13](#_Toc96338517)

[Ping-Echo 14](#_Toc96338518)

[User-Stories 15](#_Toc96338519)

[LATO NODO EDGE: 15](#_Toc96338520)

[LATO APPLICATION SERVER: 16](#_Toc96338521)

[LATO SENTINELLA: 17](#_Toc96338522)

# Introduzione

## Obiettivo

Il progetto ha come finalità la creazione di una piattaforma edge-cloud con la funzione di monitoraggio ambientale. Parte integrante del progetto sono i dispositivi IoT (internet of Things), capaci di raccogliere dati con uno scarso consumo di risorse.

I vantaggi dell’utilizzo di una rete di dispositivi IoT per la raccolta di dati ambientali sono molteplici, come la vasta scalabilità della rete e il basso costo delle spese di manutenzione, rispetto ad una classica stazione di monitoraggio ambientale.

I protagonisti della rete di acquisizione sono i watchdog, gli edge-node ed infine il cloud server:

* I **watchdog** sono i nodi sentinella, hanno il compito di campionare i dati e trasmetterli ai nodi edge. Tali dispositivi sono mobili e alimentati a batteria, perciò l’obbiettivo è quello di massimizzare il loro ciclo di vita, minimizzando il consumo di risorse. Un protocollo di comunicazione wireless a basso profilo è il *LoRaWan* che sarà discusso in seguito. Tale tecnologia ci permette di connettere i watchdog ai nodi edge con un modesto impiego di energia.
* Gli **edge-node** hanno un ruolo centrale nel progetto, il loro compito è quello di fare da *bridge* tra i watchdog ed il cloud server. Essi si interfacciano con i watchdog attraverso il protocollo LoRaWan e con il cloud server per mezzo del protocollo MQTT, anch’esso uno standard ISO per la messaggistica leggera, che si appoggia al livello TCP/IP dell’internet.
* Il **cloud server** ha la funzione di acquisire e immagazzinare le informazioni che arrivano dagli edge-node, inoltre sarà capace di analizzare e interpretare i dati, pianificando delle operazioni *self-adaptive* in modo migliorare la qualità di *dependability,* al fine di minimizzare ed ottimizzare l’intervento umano sui dispostivi fisici. Questo tipo di approccio permette di ottenere una maggiore qualità delle acquisizioni dei dati e di risparmiare sui costi di manutenzione e riparazione.

## W3C

Il World Wide Web Consortium, anche conosciuto come W3C, è un'organizzazione non governativa internazionale che ha come scopo quello di favorire lo sviluppo di tutte le potenzialità del World Wide Web e diffondere la cultura dell'accessibilità della Rete. Al fine di riuscire nel proprio intento, la principale attività svolta dal W3C consiste nello stabilire standard tecnici per il World Wide Web inerenti sia i linguaggi di markup che i protocolli di comunicazione.

Nel nostro caso siamo interessati al paragrafo 4.1.10 della guida, sul monitoraggio ambientale:

*“Il monitoraggio dell'ambiente si basa in genere su molti sensori distribuiti che inviano i propri dati di misurazione a gateway comuni, dispositivi perimetrali e servizi cloud.*

*Il monitoraggio dell'inquinamento atmosferico, dell'inquinamento idrico e di altri fattori di rischio ambientale come polveri sottili, ozono, composti organici volatili, radioattività, temperatura, umidità per rilevare condizioni ambientali critiche può prevenire danni irreparabili alla salute o all'ambiente.”*

Un altro paragrafo di nostro interesse è il 4.2.6, quello sui Digital Twins, utile per simulare le componenti.

*“Un gemello digitale è una rappresentazione virtuale, ovvero un modello di un dispositivo o un gruppo di dispositivi che risiede su un server cloud o su un dispositivo edge. Può essere utilizzato per rappresentare dispositivi del mondo reale che potrebbero non essere continuamente online o per eseguire simulazioni di nuove applicazioni e servizi, prima che vengano distribuiti ai dispositivi reali.”*



## LoRaWan

La specifica LoRaWan è un protocollo di rete LPWA (Low Power, Wide Area) progettato per connettere in modalità wireless *"things"* alimentati a batteria in reti regionali, nazionali o globali e si rivolge ai requisiti chiave dell'Internet of Things, come bi-servizi di comunicazione direzionale, sicurezza end-to-end, mobilità e localizzazione.

*Perché proprio LoRaWan?*

LoRaWan offre una durata della batteria pluriennale ed è progettato per sensori e applicazioni che richiedono di inviare piccole quantità di dati su lunghe distanze. Di seguito si può osservare un confronto tra LoRaWan e le comunicazioni wireless maggiormente adottate, come le reti LAN(local Area Network) e le reti *Cellular*



WiFi e BTLE sono standard ampiamente adottati ma servono alle applicazioni che comunicano con dispositivi personali su piccole distanze. La tecnologia cellulare è perfetta per applicazioni che richiedono un elevato throughput di dati, ma devono disporre di una notevole fonte di alimentazione.

LoRaWan definisce il protocollo di comunicazione e l'architettura di sistema (livello applicativo del modello ISO/OSI) ,mentre LoRa è lo strato fisico della telecomunicazione (livello1 del modello OSI), che consente il collegamento a lungo raggio.

Immagine che contiene tavolo

Descrizione generata automaticamente

*LoRaWan per l'Europa*

LoRaWan definisce dieci canali, otto dei quali sono multi-data da 250 bps a 5,5 kbps, un singolo canale LoRa ad alta velocità di trasmissione dati, a 11 kbps e un singolo canale FSK a 50 kbps. La potenza di uscita massima consentita da ETSI (European Telecommunications Standards Institute) in Europa è +14dBM. Ci sono restrizioni sul ciclo di lavoro ma senza limiti di tempo massimo di trasmissione o di permanenza del canale.

## MAPE-K loop

*MAPE-K control loop* è stato introdotto da IBM nel loro *white paper*: *“An architectural blueprint for autonomic computing”.* L’intento è quello di creare un ambiente informatico con le capacità di autogestione e auto-adattamento dinamico a seconda delle *business policies* implementate. Il ciclo e suddiviso in 4 aree principali:

* Monitor: raccoglie i dati delle risorse gestite
* Analyze: Esegue complesse analisi dei dati e valuta i segnali dalla funzione monitor.
* Plan: Struttura le azioni necessarie per raggiungere gli obiettivi richiesti,

essa crea o seleziona una procedura da attuare per la configurazione desiderata nella risorsa gestita.

* Execute: modifica il comportamento della risorsa gestita utilizzando gli effettori, sulla base delle azioni consigliate dalla funzione di piano.



Il nostro obbiettivo è quello di utilizzare questo modello concettuale allo scopo di creare un MAPE-Kper ogni *adaptation-goals,* inoltre provvederemo a valutare quale attore della rete prende in carico una particolare funzione del MAPE-K.

# Iterazione 0

## Analisi dei requisiti

Di seguito vengono elencati gli obiettivi che si vogliono raggiungere con questo progetto.

### R0 – Azioni di riconfigurazione per la minimizzazione dell’overhead e allungamento del ciclo di vita delle sentinelle:

Dal momento che i dispositivi IoT(le sentinelle) sono portatili, quindi alimentati da batteria, si vuole ottimizzare la durata della batteria intraprendendo delle azioni correttive sulla loro configurazione. Delle possibili azioni potrebbero intervenire sulla gestione della potenza e della frequenza di trasmissione.

I nodi edge sui quali viene distribuito il software d’interesse, sono nodi computazionali a basse prestazioni. Sarebbe opportuno gestire eventuali situazioni di congestione della rete in corrispondenza dei nodi edge configurando opportunamente la frequenza di trasmissione delle sentinelle a cui si interfaccia.

### R1 – Azioni di auto-diagnosi manutentiva di risoluzione autonoma dei guasti:

Si vuole rendere possibile la rilevazione automatica di guasti sulle sentinelle, identificando la sentinella danneggiata con le sue caratteristiche tecniche e geografiche. A seguito di un guasto si possono intraprendere due diverse tipologie di azioni:

Correzione: insieme di azioni, che il sistema compie autonomamente, volte a ripristinare il corretto funzionamento del dispositivo.

Allerta: se le azioni correttive non sono efficaci, viene mandato un messaggio di allarme per richiedere un intervento manuale.

Oltre a rendere robusta la rete di sentinelle, si vuole tutelare il sistema da possibili malfunzionamenti dovuti a guasti dei nodi edge. Questo può essere fatto introducendo diversi nodi edge che comunicano tra loro e cooperano, in modo che, a fronte di un guasto su uno di essi, intervenga un altro nodo edge funzionante per prendere in carico i compiti di quello guasto.

### REQUISITO R0:

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Goal | Fenomeno | Azione |
| S1 | Massimizzare la durata della batteria dei watchdog. | La batteria dei watchdog scende al di sotto di determinate soglie. (50%, 30%, 15%, 10%) | Al superamento di ogni soglia è necessario:   * Diminuire la frequenza di trasmissione dei dati da parte dei watchdog. * Diminuire la potenza di trasmissione dei watchdog. |
| S2 | Evitare fenomeni di congestione in corrispondenza dei nodi edge. | Il nodo edge riceve un numero di dati superiore rispetto a quelli che riesce a elaborare e trasmettere al server, causando effetti indesiderati. | Diminuire la frequenza di trasmissione dei watchdog che comunicano con il nodo edge nella situazione di congestione. |

### REQUISITI R1:

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Goal | Fenomeno | Azione |
| S3 | Rendere il sistema in grado di gestire i guasti sui watchdog in maniera autonoma.  (*self-recovery)* | Watchdog silente, cioè non trasmette dati per un certo intervallo di tempo.  (Ad es. 60s) | Il nodo edge tenta ripristinare il corretto funzionamento del watchdog guasto tramite il suo riavvio. |
| S4 | Allerta in caso di guasto ad un nodo watchdog e identificazione esatta delle sue caratteristiche fisiche e geografiche. | Guasto ad un watchdog non recuperabile. | Il nodo edge allerta un operatore per un intervento manuale sul nodo watchdog .guasto indicando le sue caratteristiche. |
| S5 | Rendere il sistema in grado di gestire guasti ai nodi edge in maniera autonoma.  (*self-recovery*) | Guasto di un nodo edge: non riceve i dati inviati dai watchdog e, quindi, non può inoltrarli al server. | * L*’application server* ridistribuisce il traffico di dati che arrivava al nodo edge guasto verso altri nodi edge. * L*’application server* notifica il personale del guasto avvenuto. |
| S6 | Limitare la perdita di informazioni dovute a gusti all’*application server.* | Malfunzionamenti dell’*application server* che quindi non è in grado di ricevere e/o elaborare i dati trasmessi dai gateway. | * I nodi edge devono mantenere i dati ricevuti dai watchdog in un buffer locale finchè il server non riprende la sua operatività. * I nodi edge devono notificare il guasto ad un indirizzo di rete dedicato e ritenuto affidabile. |

L’avvenimento di guasti viene fatto secondo la tattica ping-echo. In particolare, le condizioni di verificabilità dei vari scenari sono:

* S3: ping mandato da un nodo edge, echo non mandato dal watchdog
* S5: ping mandato dal server e/o da un nodo edge, echo non mandato dal nodo edge destinatario
* S6: ping mandato dai nodi edge, echo non mandato dal server

## Ping-Echo

TODO

## User-Stories

Le user stories sono suddivise in tre raggruppamenti, uno per attore della rete, questa soluzione ci permette di indentificare i diversi *use-case* che saranno essenziali nello *use case diagram UML.*

### LATO NODO EDGE:

-Requisito R0:

* Come nodo edge, voglio essere in grado di conoscere il livello della batteria delle sentinelle in modo da sapere quando il livello di batteria scende sotto determinate soglie (M)
* Come nodo edge, voglio poter diminuire la frequenza di trasmissione delle sentinelle in modo da allungare la durata della loro batteria. (E)
* Come nodo edge, voglio poter diminuire la potenza del segnale di trasmissione delle sentinelle in modo da allungare la durata della loro batteria. (E)
* Come nodo edge, voglio poter comunicare il livello del mio traffico di dati in arrivo in modo da rilevare eventuali situazioni di overhead. (M)
* Come nodo edge, voglio poter diminuire la frequenza di trasmissione delle sentinelle in modo da diminuire il flusso di dati in arrivo. (E)

-Requisito R1:

* Come nodo edge, voglio essere in grado di sapere quando una sentinella è silente in modo da poter rilevare sui guasti. (M)
* Come nodo edge, voglio essere in grado di riavviare una sentinella guasta in modo da ripristinare il suo funzionamento. (E)
* Come nodo edge, voglio essere in grado di mandare messaggi a del personale in modo da avvisare che una sentinella è guasta e comunicare i suoi dati. (E)
* Come nodo edge, voglio poter mandare un messaggio di ping ad altri nodi edge in modo da verificare che siano in funzione. (M)
* Come nodo edge, voglio poter subentrare al posto di un altro nodo edge in modo da svolgere i suoi compiti. (E)
* Come nodo edge, voglio poter mandare dei messaggi di ping all’application server in modo da avere informazioni sul suo funzionamento. (M)
* Come nodo edge, voglio poter analizzare le risposte ai messaggi di ping mandati all’application server in modo rilevare se sta mal funzionando. (A)
* Come nodo edge, voglio poter mantenere i dati che ricevo in un buffer locale in modo da evitare perdite di dati quando l’application server non li può ricevere.(E)
* Come nodo edge, voglio essere in grado di mandare messaggi ad un indirizzo di rete assegnato in modo da avvisare che l’application server è malfunzionante.(E)

### LATO APPLICATION SERVER:

Requisito R0:

* Come application server voglio essere in grado di valutare lo stato di carica delle sentinelle in modo da sapere quando la loro autonomia oltrepassa una certa soglia. (A)
* Come application server voglio essere in grado di determinare la frequenza e potenza del segnale di trasmissione per le sentinelle in modo da allungare il loro ciclo di vita. (P)
* Come application server voglio essere in grado di analizzare il livello di traffico in ingresso ad un nodo edge in modo da rilevare situazioni di congestione del nodo. (A)
* Come application server voglio essere in grado di determinare la frequenza di trasmissione delle sentinelle in modo da diminuire il flusso di dati in arrivo ad un nodo edge congestionato. (P)
* Come application server voglio essere in grado di analizzare il livello di traffico di un nodo edge in modo da rilevare situazioni di sottoutilizzo della rete. (A)
* Come application server voglio essere in grado di determinare la frequenza di trasmissione delle sentinelle in modo da aumentare il flusso di dati in arrivo ad un nodo edge. (P)

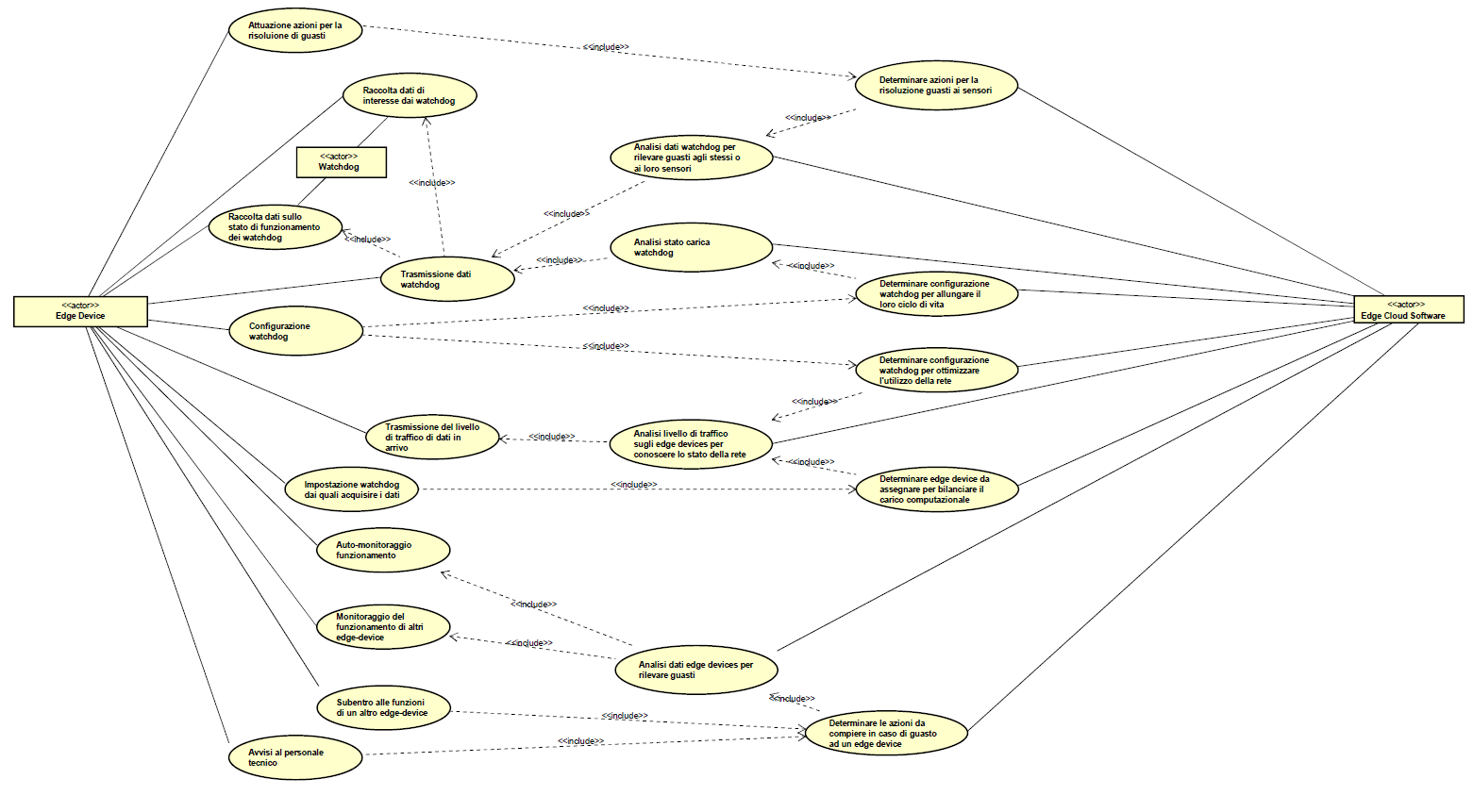
Requisito R1:

* Come application server, voglio poter fare delle analisi sui dati(le risposte ai ping) delle sentinelle ricevuti dal nodo edge in modo da sapere quando una sentinella è guasta. (A)
* Come application server, voglio poter fare delle analisi sui dati(le risposte ai ping) delle sentinelle ricevuti dal nodo edge in modo da sapere quando un sensore di una sentinella è guasto (A)
* Come application server, voglio poter comunicare ad un nodo edge di riavviare una sentinella in modo da ripristinare il suo funzionamento. (P)
* Come nodo edge, voglio poter mandare un messaggio di ping ai nodi edge in modo da avere le informazioni sul funzionamento dei nodo edge. (M)
* Come nodo edge, voglio poter analizzare le risposte ai ping mandati ai nodi edge in modo da rilevare quando un nodo edge sta mal funzionando. (A)
* Come application server, voglio poter decidere quale nodo edge assegnare ad un gruppo di sentinelle in modo da subentrare ai compiti di un nodo edge malfunzionante o guasto. (P)
* Come application server, voglio poter inviare un messaggio di allerta in modo da notificare il malfunzionamento di un nodo edge e comunicare i suoi dati. (E)

### LATO SENTINELLA:

* Come sentinella, voglio poter generare dei dati fittizi in modo da simulare una rilevazione di dati da parte di sensori.
* Come sentinella, voglio poter simulare diversi stati di funzionamento in modo da simulare un dispositivo elettronico fisico.

## Use Case Diagram



## Deployment Diagram Informale



## Deployment Diagram UML

