

Progetto Programmazione di Reti

Relazione progetto traccia 1
A.A 2020/2021

A cura di:

- Francesco Magnani (Matricola: 0000916594)
- Capelli Thomas (Matricola: 0000925150)
- Stricescu Ciprian (Matricola: 0000915547)

Indice

1	Analisi	2
1.1	Scelte di progetto	2
1.2	Strutture dati	3

Capitolo 1

Analisi

Progetto realizzato per il corso di Programmazione di Reti, svolgimento della traccia 1.

1.1 Scelte di progetto

Il codice è composto principalmente da tre parti:

- **4 Device:** Per facilitare la correzione e il debug abbiamo deciso di mantenere, consapevoli delle ripetizioni, sia il codice di 4 dispositivi differenti, ognuno con il proprio indirizzo di classe C prestabilito, sia quello di un unico dispositivo che è da avviare più volte e richiede in input un indirizzo IP.

Principalmente ogni device ha la funzione di campionare i dati necessari salvandoli su file secondo la corretta formattazione richiesta dalla consegna; inoltre, utilizza il protocollo UDP per comunicare con il gateway. In particolare la connessione viene simulata utilizzando la porta 10000 e localhost.

La generazione dei dati avviene in maniera dinamica attraverso l'utilizzo di:

- funzioni gaussiane per ottenere valori pseudo randomici (Temperatura e umidità).
 - utilizzo della libreria Time per ottenere la data relativa al campionamento.
- **4 Device (singolo script riutilizzabile):** Simile al codice dei 4 device separati ma presenta alcune differenze per poter riutilizzare lo stesso

script.

Per la creazione del file con i dati raccolti abbiamo deciso di non usare un file singolo, ma crearne uno in base all'IP del device.

La funzione `checkIP` prende da input l'IP dell'host e controlla che sia valido (indirizzo di classe C). Se l'indirizzo è corretto viene richiesto un check al gateway.

- **1 Gateway:** La gestione dei dati inviati dai device avviene attraverso una connessione UDP al gateway. I dati vengono manipolati e adattati per poter generare un header e mandarli al cloud attraverso una connessione TCP.

Per il corretto invio dei dati al cloud abbiamo ritenuto opportuno usare una metodo per "catalogare" i vari device usando un dizionario che associa ad ogni IP i dati raccolti dal device con lo stesso indirizzo. La connessione TCP avviene soltanto quando tutti e 4 i device inviano i loro dati, e in caso di più dati inviati dallo stesso device si limita soltanto ad aggiornare i dati nel gateway.

Qualvolta il gateway riceve un messaggio dai device esso viene controllato da `processData`. Se si usa lo script riutilizzabile i controlli principali si basano sulla presenza di una keyword nel messaggio (`Dev`) seguito dall'IP che il device richiede di utilizzare. Nel caso in cui questo IP sia utilizzabile il gateway rimanda un "ACK" al device, che ora può iniziare a mandare i suoi dati. Il gateway in questo modo è allo stesso momento sia un server UDP che un client TCP (utilizzando due porte differenti).

- **1 Cloud:** Per la realizzazione del cloud abbiamo fatto ricorso ad un classico server TCP, con le opportune modifiche, viste le necessità progettuali. La funzione del cloud è quella di ricevere i dati dal gateway attraverso una connessione TCP e formattarli in modo da presentarli all'utente.

1.2 Strutture dati

In questo progetto abbiamo utilizzato:

- **File:** Vengono utilizzati dai device per salvare le informazioni campionate che successivamente vengono lette per creare il messaggio da inviare.

- **Dictionary:** Usato nel gateway per gestire i dati inviati dai device. L'uso di associazione chiave-valore permette di poter semplicemente aggiornare i dati in caso di riinvio prematuro (indirizzo IP come chiave).
- **List:** Usate nel gateway per la gestione dei dati nella versione un solo client per i 4 device.