

Protocollo Vento in Cloud

Stato di questo documento

Questo RFC è una possibile soluzione del 2° protocollo della verifica del 09/12/2022 ed è attualmente in stato di lavorazione.

Riepilogo

Il protocollo **Vento in Cloud** permette di raccogliere nel cloud le misurazioni della velocità e della direzione del vento rilevate tramite un sensore presso la sede dei vari clienti. Questo documento descrive il protocollo e i pacchetti usati per la comunicazione. Inoltre spiega anche le ragioni delle scelte fatte nella sua progettazione.

Ringraziamenti

Si ringrazia tutta la comunità che condivide su Internet il proprio lavoro, la vera conoscenza è possibile solo tramite una comunione di menti.

Panoramica del protocollo

Il protocollo prevede la comunicazione tra un server in cloud che offre il servizio di raccolta dati, e i client che inviano i dati raccolti sulle rilevazioni di velocità e direzione del vento. Dato che i sensori presso le sedi locali dei client rilevano i dati del vento ogni 5 secondi, per un totale di 17280 rilevazioni continuative al giorno, la quantità di dati inviata da un client può essere considerevole, e per essere certi che tutti i dati inviati al server siano ricevuti si decide di **usare a livello trasporto il protocollo TCP** che implementa già tutte le caratteristiche di affidabilità e mantenimento della sessione necessarie per il trasferimento di file anche di grandi dimensioni. Inoltre il protocollo TCP effettua autonomamente la frammentazione e il riassettaggio del messaggio nel caso fosse necessario, usando una modalità che risulta adatta per la tipologia di trasferimento richiesto. Le uniche caratteristiche che il protocollo **Vento in Cloud** implementa sono l'identificazione dell'utente tramite un identificatore di 2 byte fornito direttamente dall'azienda produttrice, l'identificazione del tempo delle rilevazioni usando un timestamp di inizio e uno di fine, e l'identificazione del luogo di rilevazione dei dati usando la

latitudine, la longitudine e l'altitudine dal livello del mare del punto di rilevazione.

Il protocollo **prevede l'indirizzamento** per il client utilizzando un identificatore di 2 byte dall'azienda produttrice che sarà aggiunto ad ogni messaggio inviato dal client.

Il protocollo **non prevede la frammentazione e il riassettaggio** in quanto questa funzionalità è demandata al protocollo TCP.

Il protocollo **prevede l'incapsulamento** in quanto tutti i messaggi scambiati tra il client ed il server prevedono un header contenente un comando e l'identificatore del client.

Il protocollo **non è connesso** in quanto la sessione viene mantenuta solo dal protocollo TCP.

Il protocollo **è confermato** prevedendo l'invio di una conferma di ricezione del file una volta conclusa la spedizione da parte del client.

Il protocollo **prevede la rilevazione di un errore**, cioè il non riconoscimento dell'identificatore del client da parte del server. Inoltre indirettamente la mancata ricezione della conferma di ricezione del file da parte del client rappresenta intrinsecamente il rilevamento di un problema di spedizione o di ricezione. Tutti gli altri errori sono demandati demandata completamente al protocollo TCP.

Il protocollo **non prevede controllo del flusso** demandandolo completamente al protocollo TCP.

Non è previsto il multiplexing in quanto non è prevista aggregazione di dati.

Il protocollo **non prevede priorità dei pacchetti, non offre qualità del servizio e non gestisce la sicurezza**. Per quest'ultimo aspetto, se necessario, dovranno essere previste misure di sicurezza aggiuntive usando protocolli appositi a corredo di questo.

Il server ascolterà sulla porta 44444.

Il protocollo è completamente testuale con un formato dei comandi del seguente tipo:

```
      2 byte      2 byte      n byte
-----
| COMANDO |  <id>  | [<payload>] |
-----
```

Figura 1

Il campo COMANDO è costituito da un codice numerico che individua il tipo di operazione.

Il campo <id> identifica l'identificatore univoco del client usato sia in fase di spedizione del messaggio, sia in fase di conferma da parte del server.

Il campo payload può o meno essere presente a seconda del tipo di comando ed è adibito al contenimento dei dati relativi al periodo di rilevazione, all'esatta localizzazione del punto di rilevazione e ai dati del vento.

Comandi del client

Il comando descritto di seguito è inviato dal client al server (client → server).

```
-----
| <cmd> |  <id-client>  |  <timestamp-start> |  <timestamp-end> |
-----
|  <latitudine>  |  <longitudine>  |  <altitudine>  |
-----
| <velocità-1> | <direzione-1> | ... | <velocità-n | <direzione-n>|
-----
```

Figura 2

Il comando serve per effettuare la trasmissione dei dati del vento rilevati in una precisa posizione identificata da <latitudine>, <longitudine> e <altitudine>, all'interno di due date indicate da <timestamp-start> e <timestamp-end>.

Il significato e la lunghezza in byte dei campi sono i seguenti:

- <cmd>: lunghezza 2 byte, valore pari a 12 che indica la spedizione di un messaggio;
- <id-client>: lunghezza 2 byte, è l'identificativo dell'utente assegnato dall'azienda produttrice;

- <timestamp-start>: lunghezza 19 byte, timestamp di inizio delle rilevazioni inviate, il formato è 'YYYY-MM-DD hh:mm:ss';
- <timestamp-end>: lunghezza 19 byte, timestamp di fine delle rilevazioni inviate, il formato è 'YYYY-MM-DD hh:mm:ss';
- <latitudine>: lunghezza 4 byte, rappresentazione in codifica IEEE-754 a 32 bit della latitudine;
- <longitudine>: lunghezza 4 byte, rappresentazione in codifica IEEE-754 a 32 bit della longitudine;
- <altitudine>: lunghezza 4 byte, rappresentazione in codifica IEEE-754 a 32 bit dell'altitudine rispetto al livello del mare;
- velocità-n: lunghezza 3 byte, velocità del vento rilevata, i valori sono molteplici;
- direzione-n: lunghezza 3 byte, direzione del vento rilevata, i valori sono molteplici.

Comandi del server

I comandi descritti di seguito sono quelli inviati dal server al client (server → client)

```

    2 byte    2 byte
    -----
|   11   | <id> |
    -----

```

Figura 3

Il comando serve per confermare la ricezione di un messaggio contenente i dati inviati da un client identificato dall'identificatore <id>.

Segnalazione di errore

Nel caso in cui il server non riconosca l'identificatore del client, spedirà un messaggio di errore con il seguente formato.

```

    2 byte    2 byte
    -----
|   13   | <id> |
    -----

```

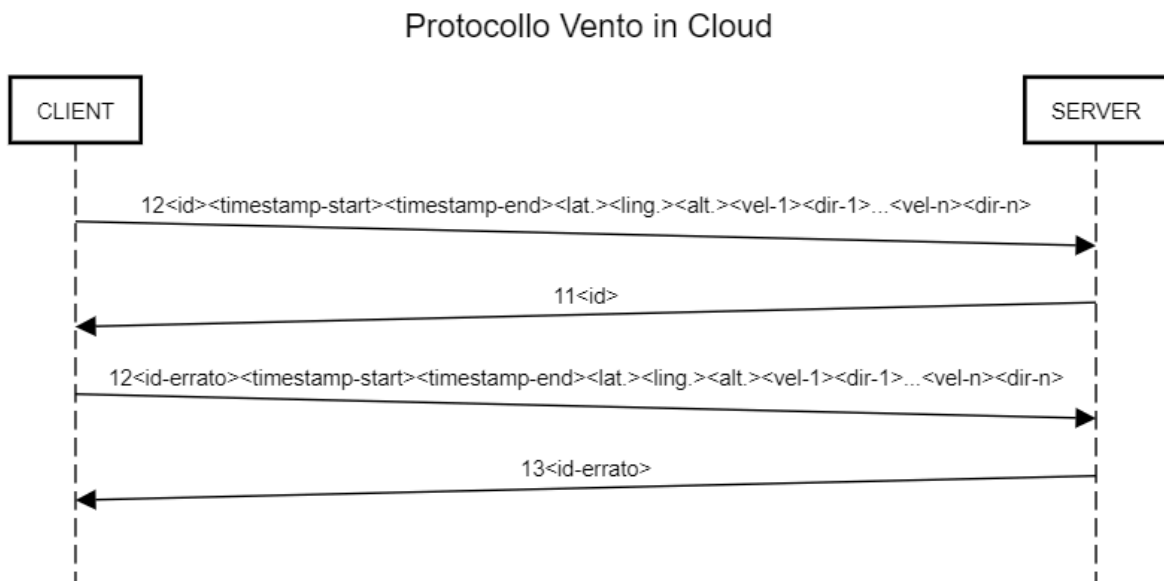
Figura 4

Il significato e la lunghezza in byte dei campi sono i seguenti:

- 13 è il codice del comando di errore e occupa 2 byte;

- <id>: lunghezza 2 byte, rappresenta l'identificatore errato inviato dal client;

Sequenza temporale¹



Il codice usato su [SequenceDiagram.org](https://sequencediagram.org) per ottenere la sequenza temporale illustrata sopra è il seguente:

```
title Protocollo Vento in Cloud
```

```
CLIENT->>(1)SERVER:
```

```
12<id><timestamp-start><timestamp-end><lat.><ling.><alt.><vel-1><dir-1>...<vel-n><dir-n>
```

```
SERVER->>(1)CLIENT:11<id>
```

```
CLIENT->>(1)SERVER:
```

```
12<id-errato><timestamp-start><timestamp-end><lat.><ling.><alt.><vel-1><dir-1>...<vel-n><dir-n>
```

```
SERVER->>(1)CLIENT:13<id-errato>
```

¹ Il sequence diagram è stato creato usando [SequenceDiagramOrg](https://sequencediagram.org)