Protocollo Monitoraggio Vento

Stato di questo documento

Questo RFC è una possibile soluzione del 1° protocollo della verifica del 09/12/2022 ed è attualmente in stato di lavorazione.

Riepilogo

Il protocollo **Monitoraggio Vento** permette di monitorare le misurazioni della velocità e della direzione del vento rilevate tramite un sensore presso la sede del cliente. 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 sensore (lato client) e l'applicazione che colleziona le rilevazioni (lato server). Il sensore rileva i dati del vento ogni 5 secondi, per un totale di 17280 rilevazioni continuative al giorno. Vista la considerevole quantità di dati raccolta giornalmente, il protocollo è progettato in modo tale da non appesantire la comunicazione tra il client e il server, preferendo l'eventuale perdita di alcuni dati rispetto alla certezza del loro ricevimento. Per questa ragione si decide di **usare a livello trasporto il protocollo UDP** che non appesantisce la comunicazione con ulteriori controlli, demandando al protocollo **Monitoraggio Vento** qualsiasi caratteristica comunicativa aggiuntiva.

Il protocollo **prevede l'indirizzamento** sia per il server, sia per il client. Gli identificatori sono forniti dall'azienda produttrice prevedendo 2 byte per l'identificatore del server e 2 byte per l'identificatore del sensore client. Al momento del primo avvio il server e il client si assoceranno creando un unico codice a 4 byte, formati unendo il codice del server e il codice del client:

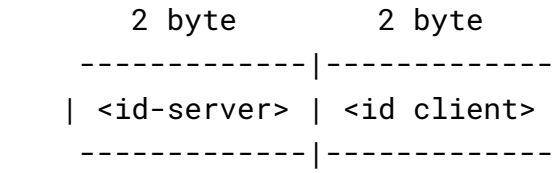


Figura 1

Questa operazione viene effettuata solo al primo avvio del sensore, successivamente il server contatterà il sensore usando il codice identificativo completo aggiungendolo ai propri comandi, e il sensore farà altrettanto quando invia i propri dati. In questo modo è possibile pensare ad una espansione futura del protocollo per la gestione contemporanea di più sensori, oltre ad evitare problemi di interferenze da parte di altri sensori nelle vicinanze. Il sensore quando sarà svegliato dal server risponderà con un messaggio di conferma e subito dopo inizierà a rilevare ed inviare i dati, aggiungendo ad ogni trasmissione il proprio identificatore.

Il protocollo **non prevede la frammentazione e il riassemblaggio** in quanto i dati trasmessi sono costituiti da pochi byte in formato testuale, prevedendo nello header un comando e l'identificatore, mentre l'eventuale payload sarà costituito anch'esso da pochi dati di rilevazione del vento.

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.

Il protocollo **è connesso** in quanto al primo avvio avviene una fase iniziale di associazione tra il server e il sensore client creando l'identificatore univoco, associazione che può essere terminata chiudendo la connessione ed eliminando il sensore dall'applicazione. Le fasi di terminazione e ri-avvio della rilevazione dei dati del vento, prevedono comunque uno scambio di messaggi di associazione tra il client e il server, ma non prevedono la chiusura della connessione iniziale.

Il protocollo **non è confermato durante la spedizione delle rilevazioni** per non creare traffico e perché per il sensore client non è necessario ricevere la conferma di ricezione dei dati inviati al server. È possibile che alcuni pacchetti vengano persi per interferenze momentanee, ma vista la rilevante quantità di dati inviati è accettabile questa perdita.

Il protocollo **prevede una conferma per ogni comando di controllo** per la gestione del sensore. Quest'ultimo invia un messaggio di conferma

per ogni comando relativo all'associazione o alla chiusura dell'associazione, definitiva o temporanea, con il server.

La **rilevazione degli errori è prevista** su diversi livelli. Il funzionamento regolare del sensore viene garantito dalle spedizioni dei dati ogni 5 secondi. Nel caso il sensore non effettuasse più le spedizioni si assumerà un malfunzionamento che dovrà essere gestito manualmente dall'utente. L'eventuale mancata associazione tra sensore e server, sia al primo avvio, sia ai successivi, sarà da considerare come un malfunzionamento da gestire manualmente dall'utente.

Il **controllo del flusso è previsto** intrinsecamente nel protocollo in quanto il sensore client invierà i dati rilevati ogni 5 secondi.

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 55555.

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

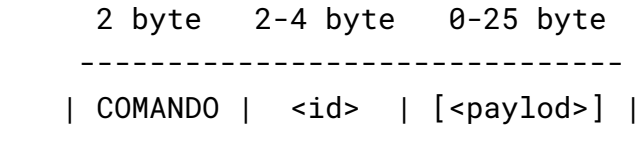


Figura 2

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

Il campo <id> identifica l'identificatore univoco del server in fase di prima connessione (2 byte) o l'identificatore univoco del sensore creato dall'unione dell'identificatore del server e del sensore client (4 byte).

Il campo payload può o meno essere presente a seconda del tipo di comando ed è adibito al contenimento dei dati sul vento.

Comandi del server

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

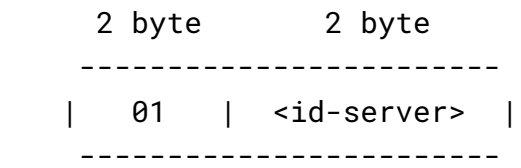


Figura 3

Il comando serve per effettuare la prima connessione con il sensore client. L'identificatore univoco <id> è l'identificatore del server.

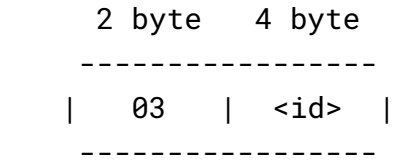


Figura 4

Il comando serve per indicare al sensore client di terminare temporaneamente la rilevazione dei dati del vento. L'identificatore univoco <id> è l'identificatore del sensore.

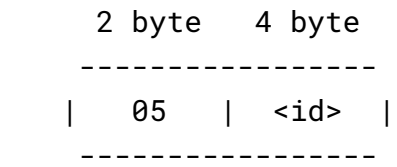


Figura 5

Il comando serve per indicare al sensore client di riprendere la rilevazione dei dati del vento. L'identificatore univoco <id> è l'identificatore del sensore.

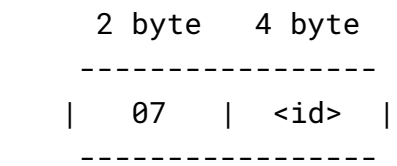


Figura 6

Il comando serve per terminare l'associazione tra il server e il sensore client, in modo tale che quest'ultimo possa essere sostituito o associato ad un altro server. L'identificatore univoco <id> è l'identificatore del sensore.

Comandi del client

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

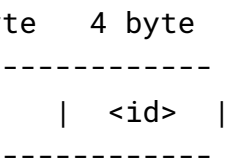


Figura 7

Il comando serve per effettuare la prima connessione con il server. L’identificatore univoco <id> è l’identificatore generato dall'unione dell’identificatore del server seguito dall’identificatore del sensore client. Questo identificatore servirà per individuare univocamente un sensore.

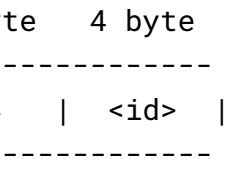


Figura 8

Il comando serve per segnalare che il sensore smette di rilevare i dati del vento avendo ricevuto il comando dal server. L’identificatore univoco <id> è l’identificatore del sensore.

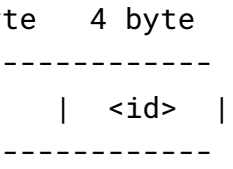


Figura 9

Il comando serve per segnalare che il sensore riprende la rilevazione dei dati del vento dopo aver ricevuto il relativo comando dal server. L’identificatore univoco <id> è l’identificatore del sensore.

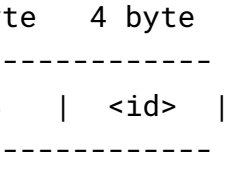


Figura 10

Il comando serve chiude l'associazione tra il sensore client e il server. L’identificatore univoco <id> è l’identificatore del sensore.

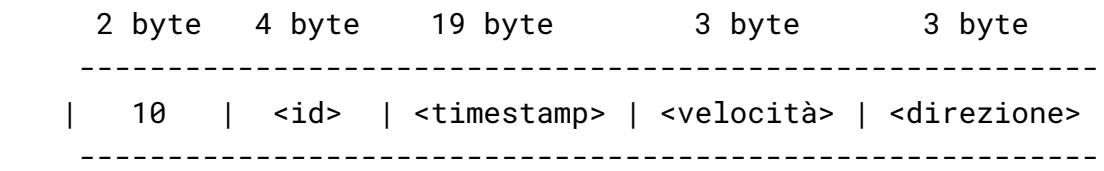


Figura 11

Il comando è usato per l’invio dei dati del vento ogni 5 secondi.

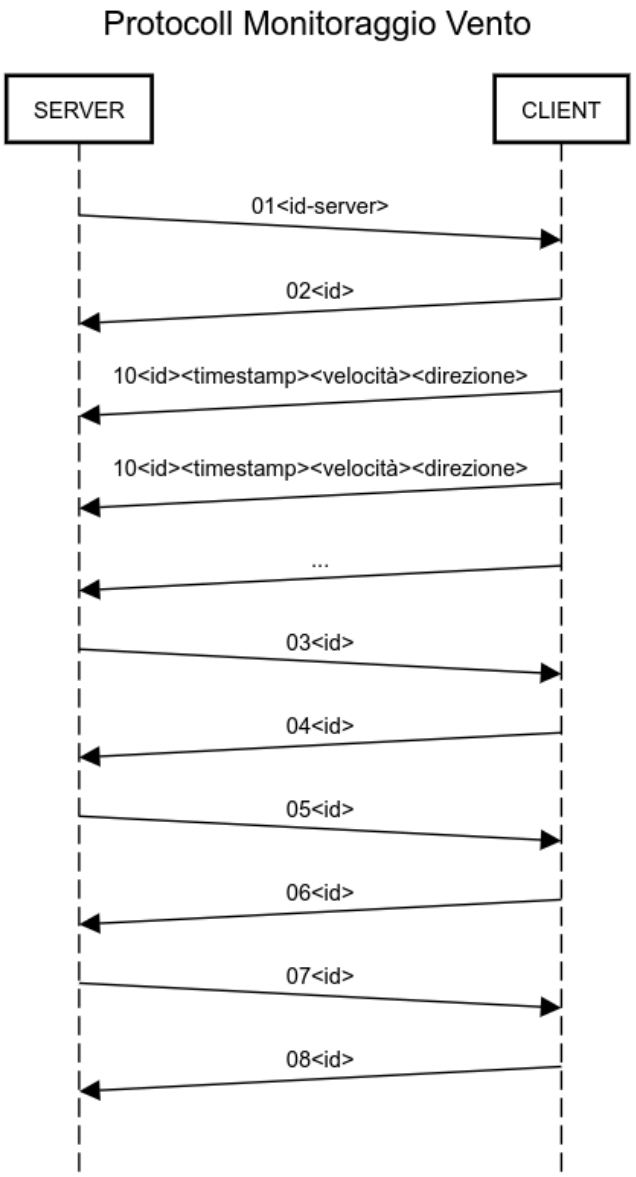
L’identificatore univoco <id> è l’identificatore generato dall'unione dell’identificatore del server seguito dall’identificatore del sensore client. Questo identificatore servirà per individuare univocamente un sensore.

Il formato del timestamp è yyyy-mm dd-hh.mm.ss costituito da cifre decimali in formato testuale.

Il formato della velocità in km/h è nnn, tre cifre decimali in formato testuale.

Il formato della direzione in gradi è nnn, tre cifre decimali in formato testuale.

Sequenza temporale[1]



Il codice usato su SequenceDiagram.org per ottenere la sequenza temporale illustrata sopra è il seguente:

```
title Protocoll Monitoraggio Vento

SERVER->(1)CLIENT: 01<id-server>
CLIENT->(1)SERVER: 02<id>
CLIENT->(1)SERVER:10<id><timestamp><velocità><direzione>
CLIENT->(1)SERVER:10<id><timestamp><velocità><direzione>
CLIENT->(1)SERVER:...
SERVER->(1)CLIENT:03<id>
CLIENT->(1)SERVER:04<id>
```

[1] Il sequence diagram è stato creato usando SequenceDiagramOrg

```
SERVER->(1)CLIENT:05<id>
CLIENT->(1)SERVER:06<id>
SERVER->(1)CLIENT:07<id>
CLIENT->(1)SERVER:08<id>
```