Protocollo invio dati

Documento di specifiche – versione documento 1.3

B.M. Tecnologie Industriali S.p.A. Società Benefit
Società soggetta all'attività di Direzione e Coordinamento di AlmavivA S.p.A.
Via Dell'Industria, 12- 35035 RUBANO (PD)- ITALY
Tel. +39(0)49-8841651 – Fax +39(0)49-8841654
e-mail: bm@bmtecnologie.it
web: www.bmtecnologie.it www.almaviva.it

REV.	Tipo revisione	Approvazione	Data
1.3	Modificata descrizione dei codici di stato	AL	19/03/2024
1.2	Aggiunti dettagli nuovi codici	AL	08/02/2024
1.1	Aggiunti dettagli nuovi codici	AL	28/02/2023
1.0	Prima versione in lingua italiana	AL	

1 Sommario

2 Introduzione

BM Tecnologie Industriali SpA sviluppa, produce e commercializza strumentazione che può essere dotata di modem integrato per connessione verso un sistema centrale di raccolta dati (SCADA/Cloud/…).
A seconda del modem montato sulla scheda elettronica, la connessione può avvenire su differenti reti: 2G e 4G (dual mode Cat M1/NB1 (NB-IoT) capability and 2G fallback) messe a disposizione dai provider telefonici.
La comunicazione tra strumentazione e server può avvenire utilizzando i più comuni protocolli di comunicazione: FTP, HTTP, MQTT o altro. All'interno di ogni protocollo di comunicazione esiste poi il formato e la sintassi del pacchetto (payload) utilizzato.
Lo scopo di questo documento è di descrivere il formato e la sintassi dei dati scambiati, al fine di guidare l'utente ad una corretta integrazione all'interno della propria infrastruttura.

2.1 Applicabilità

Gli strumenti a cui ci si riferisce all'interno di questo documento sono:

- **Bjong versione FLOW**, per applicazioni di portata con sensore Doppler o TTFM
- **Bjong versione STND**, per applicazioni con porta RS-485 standard

I dettagli riportati sono relativi ai protocolli di trasmissione:

- **MQTT**.

Negli anni sono state rilasciate diverse versioni firmware per gli strumenti. Per individuare la compatibilità con i protocolli, fare riferimento alla tabella seguente:

Strumento	Versione FW	Modem	Rete	TLS	HTTP	FTP	MQTT	CoAP	Note
BJONG FLOW/STND	Da versione FW1.1.x	2G/4G Cat M1/NB1 (NB-IoT)	2G				✓		Primo rilascio MQTT
	Da versione FW1.2.x		2G				✓		
	Da versione FW1.3.x		2G	✓			✓		Possibilità di aggiornare la configurazione da remoto
	Da versione FW1.3.3		2G	✓			✓		Possibilità di aggiornare software (strumento e modem) da remoto (FOTA)

Per versioni precedenti di FW, per strumenti o protocolli diversi, è necessario fare riferimento a documentazione alternativa specifica.

3 Generalità sulla trasmissione

Gli strumenti di BM Tecnologie Industriali SpA possono inviare molteplici informazioni, non limitatamente ai dati delle variabili memorizzate durante il normale funzionamento.
Tutti gli scambi eseguiti tramite protocollo MQTT sono accomunati dall'essere messaggi di testo che seguono la sintassi del formato JSON (https://www.json.org/json-it.html).
Gli strumenti inviano diversi tipi di messaggio, relativamente a:

- dati storici delle variabili
- allarmi basati su soglia
- stato di funzionamento (variabili di stato)
- configurazione dello strumento

Gli strumenti sono inoltre in grado di ricevere ed elaborare particolari messaggi per:

- aggiornamento da remoto della configurazione
- aggiornamento da remoto del Firmware

Nei capitoli seguenti verranno riportate le strutture dei singoli messaggi per poi, alla fine, aggiungere i dettagli sulla trasmissione in MQTT.

Ogni informazione è generalmente legata ad un preciso istante temporale, identificato tramite un "timestamp", ovvero una particolare rappresentazione di data e ora. I dettagli del formato sono riportati in un paragrafo dedicato.

Il formato del pacchetto può essere di 2 tipi:
FORMATO SINGOLO (singolo timestamp): ogni pacchetto contiene dati relativi ad un solo timestamp. Quando è necessario inviare dati relativi a più timestamp, questi verranno inseriti in più messaggi;
FORMATO AGGREGATO (timestamp multipli): un pacchetto può contenere uno o più timestamp, opportunamente identificabili.
I dettagli sono riportati in un paragrafo dedicato.

La durata della batteria è influenzata dal numero di pacchetti inviati.
In caso di invio di pacchetti con singolo timestamp il numero di pacchetti e i caratteri ridondanti aumentano notevolmente compromettendo la durata della batteria.
In fase di selezione del tipo di formato si prega di valutare attentamente e stimare il numero di pacchetti che lo strumento deve inviare in funzione del numero di variabili loggate e del periodo di campionamento.

È possibile che un messaggio relativo a uno stesso timestamp venga automaticamente suddiviso ed inviato in più pacchetti di protocollo.
Occorre quindi predisporre nel server il merge (unione) dei pacchetti con stesso timestamp.
Questa complicanza è resa necessaria dal fatto che potenzialmente ci potrebbero essere strumenti che registrano molte variabili (es.: 30 variabili) e per ogni variabile le proprietà

---

utilizzino più caratteri (sia per il campo valore che per il campo unità di misura), quindi non è possibile determinare a priori quanto lungo debba essere il pacchetto di protocollo per contenere l'intero messaggio.
La suddivisione è fatta in modo che ogni messaggio contenga una stringa JSON completa.

La dimensione massima di un singolo pacchetto è: 1200 Byte (ovvero 1200 caratteri)

4 Formato data e ora

La strumentazione di BM Tecnologie Industriali SpA equipaggiata di modem per la comunicazione dei dati è dotata di un RTC (Real Time Clock) interno tamponato per la gestione della data e dell'ora (timestamp). L'ora di questo orologio interno viene sincronizzata tramite protocollo NTP (Network Time Protocol) a quella di un server NTP che può essere impostato dall'utente. L'orologio interno è riferito al tempo UTC (Coordinated Universal Time).

I dati pubblicati dallo strumento sono sempre riferiti a UTC (UTC+0).
Non sono in alcun caso gestite l'ora solare/legale o eventuali ore locali.

4.1 Standard di riferimento

Lo standard internazionale di riferimento è la ISO8601.

In tutti i casi in cui lo strumento pubblica una data e ora viene utilizzata la medesima rappresentazione.
Es.: pacchetti di invio dati nei diversi protocolli, SMS, invio eventi, …

Tra le possibili rappresentazioni della data e ora (timestamp) secondo la ISO8601, il formato utilizzato nella strumentazione di BM Tecnologie Industriali è descritto di seguito. In linea di principio viene scelta la "forma base" anziché quella "estesa".

4.1.1 Data

Il formato della data segue la "forma base": [YYYY][MM][DD].
Dove [YYYY] indica l'anno, [MM] indica il mese e [DD] il giorno.
Ad esempio il "10 aprile 1985" è scritto in forma base come 19850410.

4.1.2 Ora

L'ISO 8601 usa il sistema a 24 ore.
Viene utilizzata la “forma base”: [hh][mm][ss].
Dove [hh] indica l'ora tra le 00 e le 23, [mm] si riferisce ai minuti da 00 a 59, e [ss] i secondi sempre tra 00 e 59.
La mezzanotte viene indicata con [00][00][00].
Ad esempio l’ora "10:26:01” è scritta in forma base come 102601.

4.1.3 Time Zone

Il time zone non viene specificato perché lo strumento riferisce i dati a UTC. L’ISO 8601 prevede una “Z” dopo l’ora in questo caso.

4.1.4 Data e ora completa

La rappresentazione combinata di data e ora avviene inserendo una "T" fra la data e l'ora.
Ad esempio, il 10 aprile 1985 alle ore 10:26:01 viene espresso come 19850410T102601Z.

4.1.5 Formato alternativo: Unix timestamp

A partire dalle versioni Firmware 1.3.10 del Bjong (standard e flow) è stata introdotta la possibilità di utilizzare un formato alternativo, ovvero l’Unix timestamp. Questo formato consiste in un numero intero che esprime il numero di secondi trascorsi a partire dal primo gennaio 1970.

5 Identificazione degli strumenti

Tutti gli strumenti sono identificati in maniera univoca dal proprio MAC address (brevemente detto MAC). Si tratta di una stringa di dodici caratteri esadecimali (numeri 0...9 e lettere A...F) che è riportata sull'etichetta esterna del case e non è modificabile dall'utente.
Esiste un ulteriore codice identificativo (ID) che può essere modificato dall'utente tramite applicazione EasySetup. Si tratta di un valore che viene mantenuto per compatibilità con vecchi sistemi di acquisizione dei dati.

In fase di realizzazione di una nuova architettura software per l'acquisizione dei dati, è opportuno basare l'identificazione degli strumenti sul solo MAC address, ignorando completamente il campo valore dell'ID. Questo garantisce infatti eventuali problemi legati all'assegnazione di identificativi non univoci.

In fase di trasmissione, i messaggi vengono inviati in modo che sia possibile associarli univocamente al MAC dello strumento, ovvero impostando uno specifico topic (MQTT).
In casi particolari è possibile configurare lo strumento in modo che ripeta le informazioni di MAC e ID all'interno della stringa JSON contenente i dati trasmessi (opzione "Ripetizione MAC" all'interno dell'app di configurazione EasySetup).
Ulteriori dettagli implementativi saranno specificati in seguito.

6 Identificazione delle variabili

Ogni strumento può inviare valori relativi a più variabili misurate. Il nome della variabile che identifica un valore letto è una **stringa di massimo 10 caratteri**, denominata TAG.
Ad esempio l'ingresso analogico 1, indipendentemente dalla dimensione fisica misurata (livello, pressione, …) viene identificato e trasmesso come AN1.

Lo strumento invia le sole variabili memorizzate in funzione della programmazione utente (abilitazione log).
Lo strumento può gestire diversi tipi di sensori e, a seconda del tipo di sensore collegato, invia specifiche variabili. I moduli ad oggi gestiti dagli strumenti sono:

1. Modulo TTFM: tempo di transito per tubi pieni per applicazioni nella rete acquedottistica
2. Modulo DOPPLER: sensore doppler per misure su rete fognaria
3. Modulo ANALISI: sonda per analisi chimico-fisiche di acque potabili e reflue
4. Nessun modulo: lo strumento lavora solo con gli I/O a bordo.

Per maggiori informazioni si rimanda al manuale dello strumento.

6.1 Nome variabile hardware (TAG)

TAG VARIABILE	DESCRIZIONE	UNITA' MISURA	RAPPRESENT.	LUNGHEZ. STRINGA
I/O A BORDO DATALOGGER				
DI1	Valore Ingresso Digitale 1 mediato sul periodo di campionamento	U.M. utente	Float	*1
TOT1	Totalizzatore Digitale 1	U.M. utente	Float	*1
DI2	Valore Ingresso Digitale 2 mediato sul periodo di campionamento	U.M. utente	Float	*1
TOT2	Totalizzatore Digitale 2	U.M. utente	Float	*1
DI3	Valore Ingresso Digitale 3 mediato sul periodo di campionamento	U.M. utente	Float	*1
TOT3	Totalizzatore Digitale 3	U.M. utente	Float	*1
DI4	Valore Ingresso Digitale 4 mediato sul periodo di campionamento	U.M. utente	Float	*1
TOT4	Totalizzatore Digitale 4	U.M. utente	Float	*1
AN1	Valore Ingresso Analogico 1	U.M. utente	Float	*1
AN1I	Ingresso Analogico 1 Curva	U.M. utente	Float	*1
AN2	Valore Ingresso Analogico 2	U.M. utente	Float	*1
AN2I	Ingresso Analogico 2 Curva	U.M. utente	Float	*1
DIEV1	Evento digitale 1 - ON/OFF (1/0)	U.M. utente	Bit	1
DIEV2	Evento digitale 2 - ON/OFF (1/0)	U.M. utente	Bit	1
DIEV3	Evento digitale 3 - ON/OFF (1/0)	U.M. utente	Bit	1
VARIABILI MODULO TTFM (FW versione FLOW)				
Q	Portata	U.M. utente	Float	*1
UP	Potenza sensore UP (non più supportata)	%	Float	*1
DN	Potenza sensore DN (non più supportata)	%	Float	*1
QUAL	Qualità (diagnostica installazione)	%	Float	*1
VEL	Velocità fluido	U.M. utente	Float	*1
NET	Totalizzatore netto	m^3	Float	*1
POS	Totalizzatore positivo	m^3	Float	*1
NEG	Totalizzatore negativo	m^3	Float	*1
DAYT	Totalizzatore giornaliero	m^3	Float	*1
TT	TOM/TOS (diagnostica installazione)	%	Float	*1
VARIABILI MODULO DOPPLER (FW versione FLOW)				
Q1	Portata	U.M. utente	Float	*1
V1	Velocità fluido	m/s	Float	*1
L1	Livello dal fondo (tirante)	m	Float	*1
LS1	Livello letto dal sensore doppler	m	Float	*1

T1	Temperatura fluido	°C	Float	*1
QL1	Qualità (diagnostica installazione)	%	Float	*1
VARIABILI SENSORE PRESSIONE INTEGRATO				
PRES	Pressione letta dal sensore di pressione integrato	Bar	Float	*1
VARIABILI MODBUS RS485 (FW versione STND)				
VAR1	Val. ingegnerizzato di VAR1 su RS485	U.M. utente	Float	*1
…			Float	*1
…			Float	*1
VAR20	Val. ingegnerizzato di VAR20 su RS485	U.M. utente	Float	*1
VARIABILI INTERNE				
HUM	Umidità interna scheda	%	Float	*1
TBRD	Temperatura interna scheda	°C	Float	*1
VBATT	Tensione di batteria	V	Float	*1

*1: La lunghezza della stringa inviata dipende dal valore letto e dal numero di cifre decimali impostate dall'utente.

7 Generalità sul formato dei messaggi

Il messaggio (o pacchetto) di base è sempre costituito dalla serializzazione in testo di un oggetto JSON composto da più coppie chiave:valore relative ad un singolo timestamp, ovvero ad un istante temporale a cui si riferisce la misurazione, l'evento o, più in generale, il dato contenuto nel pacchetto.
Il timestamp è sempre contenuto come valore della chiave "ts", secondo il formato data ora specificato nell'apposito capitolo.
Inoltre sono presenti una o più chiavi relative, ad esempio, alle variabili misurate con i relativi valori letti o ai codici di configurazione.

Lo strumento può essere impostato in modo da comporre i pacchetti in due differenti modalità:

- **Formato singolo**: prevede la presenza di un unico oggetto per messaggio (dati, configurazione, stato, …) corrispondente ad un unico timestamp con le ulteriori chiavi specifiche del messaggio. Ci si riferisce a questo formato anche con la notazione "singolo timestamp";
- **Formato aggregato**: prevede la presenza di una serie di oggetti omogenei (array) espressi come nel formato singolo. Ci si riferisce a questo formato anche con la notazione "timestamp multiplo".

In alcuni casi in cui non è mai presente una serie storica di informazioni (come ad esempio per i messaggi di stato), anche impostando il formato aggregato, nel messaggio si troverà un unico oggetto e non un array di oggetti. Tutti i dettagli sono riportati nei paragrafi successivi che analizzano i formati dei singoli tipi di messaggio.

È presente una ulteriore configurazione relativa alla possibilità di includere all'interno del messaggio le informazioni di MAC e ID dello strumento. In caso di attivazione di questa modalità, il messaggio conterrà 3 chiavi:

1. MAC
2. ID
3. Una chiave specifica in base al dato trasmesso (ad es. "HData" per i dati storici, "Alrm" per gli allarmi, …)

All'interno di questa ultima chiave sarà contenuto un singolo oggetto o un array di oggetti in base al formato singolo o aggregato scelto.

Combinando le due impostazioni, è quindi possibile configurare la trasmissione in quattro formati differenti che vengono riportati di seguito con dei semplici casi esplicativi.

Se non si imposta la ripetizione del MAC e ID, l'identificazione dello strumento tramite MAC address può essere eseguita tramite un meccanismo differente dipendente dal protocollo utilizzato. Nel caso di MQTT, l'informazione può essere reperita analizzando il topic di pubblicazione. La ripetizione è da evitare per non appesantire il messaggio con informazioni non strettamente necessarie.

8 Messaggio dei dati storici (HData)

I dati storici rappresentano i valori delle variabili memorizzate dallo strumento durante il funzionamento.

In caso di valore non numerico o NaN viene inviato il campo valore come “null”.

8.1 Formato singolo (singolo timestamp)

8.1.1 Formato singolo senza ripetizione MAC e ID

Si tratta del pacchetto più semplice che può essere trasmesso. Contiene una chiave “ts” ed uno o più chiavi equivalenti ai tag delle variabili misurate.

<table>
<tr><th colspan="3">PACCHETTO DATI SINGOLO SENZA RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>Nessuna chiave iniziale</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più TAG variabile></td><td>Valore ingegnerizzato della variabile. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è quello impostato dall’utente.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>

```
{
   "ts": "20200320T155600Z",
   "AN1": 2.7,
   "AN2": 3.4,
   "DI1": 1.8,
   "TOT1": 1523.3,
   "DI2": 1.22,
   "TOT2": 1466.3
}
```

</td>
<td>

JSON Viewer

JSON

- ts : 20200320T155600Z
- AN1 : 2.7
- AN2 : 3.4
- DI1 : 1.8
- TOT1 : 1523.3
- DI2 : 1.22
- TOT2 : 1466.3

</td>
</tr>
</table>

8.1.2 Formato singolo con ripetizione MAC e ID

<table>
<tr><th colspan="3">PACCHETTO DATI SINGOLO CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall’utente da 0 a 65.535. Deve essere univoco.</td><td>5</td></tr>
<tr><td>HData</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più TAG variabile></td><td>Valore ingegnerizzato della variabile. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è quello impostato dall’utente.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>

```
{
   "MAC": "E82A4452061C",
   "ID": "12",
   "HData": {
      "ts": "20200320T155600Z",
      "AN1": 2.7,
      "AN2": 3.4,
      "DI1": 1.8,
      "TOT1": 1523.3,
      "DI2": 1.22,
      "TOT2": 1466.3
   }
}
```

</td>
<td>

JSON Viewer

JSON
- MAC : E82A4452061C
- ID : 12
- HData
  - ts : 20200320T155600Z
  - AN1 : 2.7
  - AN2 : 3.4
  - DI1 : 1.8
  - TOT1 : 1523.3
  - DI2 : 1.22
  - TOT2 : 1466.3

</td>
</tr>
</table>

8.2 Formato aggregato (multi timestamp)

8.2.1 Formato aggregato senza ripetizione MAC e ID

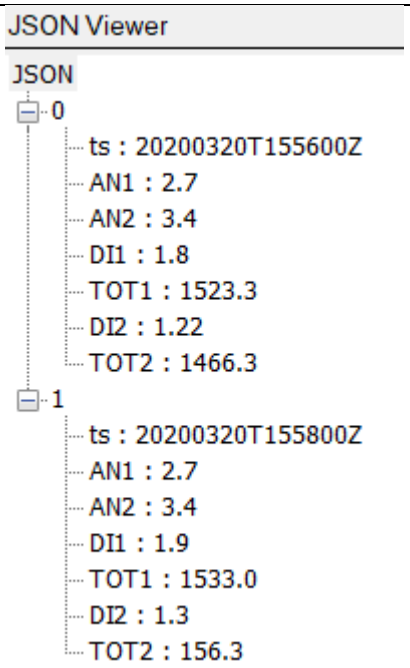
Il formato aggregato è equivalente alla versione singola ma, in questo caso, prevede un array di oggetti relativi a più timestamp.

<table>
<tr><th colspan="3">PACCHETTO DATI AGGRAGATO SENZA RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>Nessuna chiave iniziale</td><td>Array di uno o più oggetti costituiti da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più TAG variabile></td><td>Valore ingegnerizzato della variabile. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è quello impostato dall’utente.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>

```
[
  {
    "ts": "20200320T155600Z",
    "AN1": 2.7,
    "AN2": 3.4,
    "DI1": 1.8,
    "TOT1": 1523.3,
    "DI2": 1.22,
    "TOT2": 1466.3
  },
  {
    "ts": "20200320T155800Z",
    "AN1": 2.7,
    "AN2": 3.4,
    "DI1": 1.9,
    "TOT1": 1533.0,
    "DI2": 1.3,
    "TOT2": 156.3
  }
]
```

</td>
<td>



</td>
</tr>
</table>

8.2.2 Formato aggregato con ripetizione MAC e ID

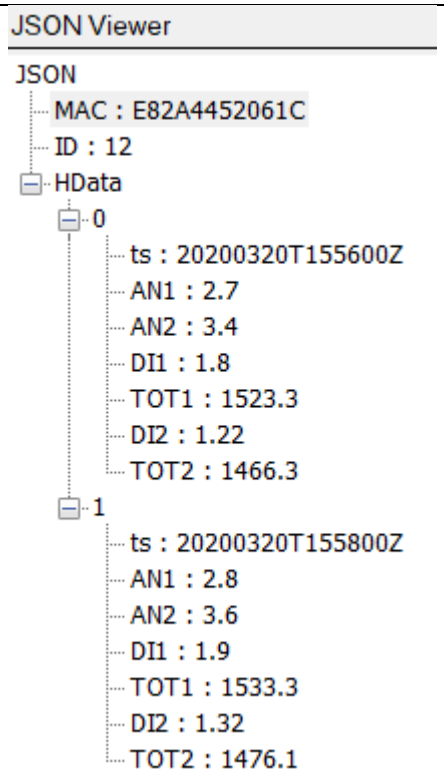
Il formato aggregato è equivalente alla versione singola ma, in questo caso, prevede un array di oggetti all'interno di HData.

<table>
<tr><th colspan="3">PACCHETTO DATI AGGREGATO CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall'utente da 0 a 65.535. Deve essere univoco.</td><td>5</td></tr>
<tr><td>HData</td><td>Array di uno o più oggetti costituiti da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più TAG variabile></td><td>Valore ingegnerizzato della variabile. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è quello impostato dall'utente.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>

```
{
   "MAC": "E82A4452061C",
   "ID": "12",
   "HData": [
      {
         "ts": "20200320T155600Z",
         "AN1": 2.7,
         "AN2": 3.4,
         "DI1": 1.8,
         "TOT1": 1523.3,
         "DI2": 1.22,
         "TOT2": 1466.3
      },
      {
         "ts": "20200320T155800Z",
         "AN1": 2.8,
         "AN2": 3.6,
         "DI1": 1.9,
         "TOT1": 1533.3,
         "DI2": 1.32,
         "TOT2": 1476.1
      }
   ]
}
```

</td>
<td>



</td>
</tr>
</table>

9 Messaggio degli allarmi (Alrm)

Su uno strumento l'evento di allarme si innesca (ON) quando il valore di una variabile supera una soglia impostata dall'utente. In modo analogo, l'allarme rientra (OFF) nel momento in cui il superamento della soglia non è più soddisfatto.
Se l'invio è abilitato, l'evento di allarme (o di rientro allarme) viene inviato non appena viene rilevato dallo strumento.

Su Bjong, la valutazione del superamento di una determinata soglia di allarme avviene solo al momento del campionamento delle variabili.
Sullo strumento è anche possibile gestire un ritardo, espresso in numero di campionamenti, in fase di attivazione o disattivazione di un allarme. Per maggiori dettagli sul meccanismo di generazione e rientro degli allarmi, consultare il manuale utente dello strumento.

Su Bjong esistono alcune opzioni che abilitano particolari funzioni quando un allarme viene attivato o disattivato. Ad esempio è possibile inviare dati storici presenti in memoria unitamente all'evento di allarme ("invio veloce su allarme") oppure cambiare il periodo di campionamento durante la permanenza dell'allarme in ON ("campionamento veloce su allarme").
Per maggiori dettagli, consultare il manuale utente dello strumento.

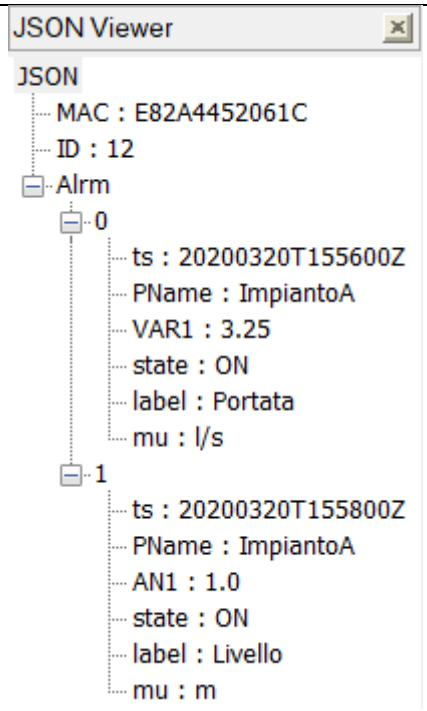
9.1.1 Formato aggregato con ripetizione MAC e ID

Di seguito, a titolo di esempio, è riportato un pacchetto di allarme in formato aggregato con ripetizione MAC. Per gli altri formati, valgono le stesse considerazioni fatte per il messaggio dei dati storici.

<table>
<tr><th colspan="3">PACCHETTO ALLARME AGGREGATO CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall'utente da 0 a 65.535.</td><td>5</td></tr>
<tr><td>Alrm</td><td>Un array di oggetti costituiti da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td>Pname</td><td>Nome impianto impostato dall'utente</td></tr>
<tr><td>state</td><td>Stato allarme "ON" o "OFF"
ON: allarme attivo
OFF: allarme disattivato</td></tr>
<tr><td>label</td><td>Etichetta impostata dall'utente</td></tr>
</table></td><td>ts: 16 caratteri
Pname: 20 caratteri
state: 3 caratteri
label: 20 caratteri
TAG: 10 caratteri
mu: 6 caratteri</td></tr>
</table>

	<TAG>	Valore ingegnerizzato della variabile al momento della generazione dell'allarme. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è quello impostato dall'utente.	
	mu	Viene riportata l'etichetta dell'unità di misura se impostata dall'utente (non vuota)	


```
{
   "MAC": "E82A4452061C",
   "ID": "12",
   "Alrm": [
      {
         "ts": "20200320T155600Z",
         "PName": "ImpiantoA",
         "VAR1": 3.25,
         "state": "ON",
         "label": "Portata",
         "mu": "l/s"
      },
      {
         "ts": "20200320T155800Z",
         "PName": "ImpiantoA",
         "AN1": 1.0,
         "state": "ON",
         "label": "Livello",
         "mu": "m"
      }
   ]
}
```

10 Messaggio delle variabili di stato (State)

I messaggi relativi allo stato dello strumento trasmettono informazioni relative allo strumento stesso, quindi non in relazione ai valori letti dagli I/O esterni ma da variabili interne.
L'invio dello stato può essere abilitato o disabilitato dall'utente.
I valori assunti sono di diverso tipo:

- Valori istantanei (come ad es. tensione della batteria) rilevati all'ultima misurazione prima dell'invio degli stati;
- Valori legati all'ultima trasmissione nel caso di stati relativi alla rete (qualità, livello del segnale, etc);
- Contatori giornalieri azzerati alla mezzanotte.

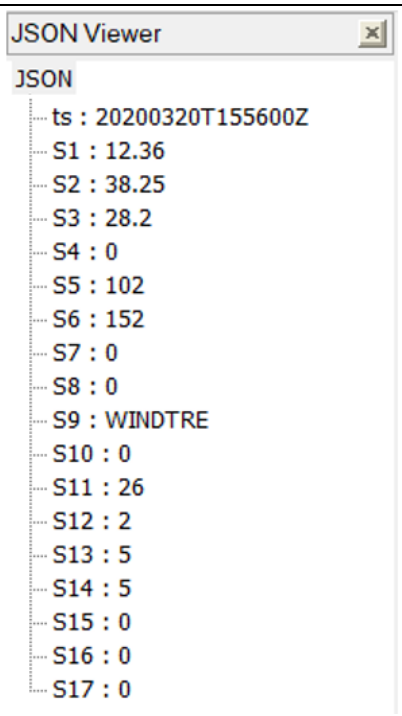
Le informazioni sullo stato vengono inviate al server, una volta al giorno, nella trasmissione della mezzanotte oppure in caso di esecuzione del "test invio dati".

10.1 Formato unico

I messaggi di stato non sono influenzati dall'impostazione del formato singolo/aggregato poiché vengono sempre inviati dati relativi ad un unico timestamp. Esiste dunque un unico formato nelle varianti con e senza ripetizione MAC.

10.1.1 Formato senza ripetizione MAC e ID

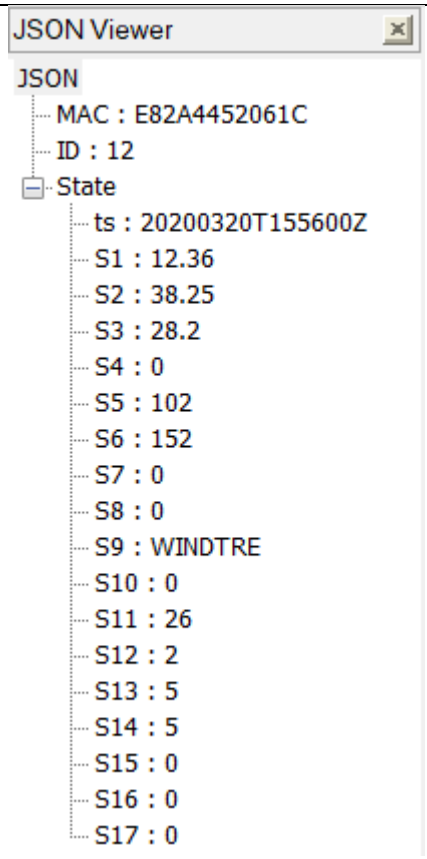
<table>
<tr><th colspan="3">PACCHETTO STATO SENZA RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>Nessuna chiave iniziale</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più codici di stato></td><td>Valore dello stato. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è fisso.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>
<pre>
{
  "ts": "20200320T155600Z",
  "S1": 12.36,
  "S2": 38.25,
  "S3": 28.2,
  "S4": 0,
  "S5": 102,
  "S6": 152,
  "S7": 0,
  "S8": 0,
  "S9": "WINDTRE",
  "S10": 0,
  "S11": 26,
  "S12": 2,
  "S13": 5,
  "S14": 5,
  "S15": 0,
  "S16": 0,
  "S17": 0
}
</pre>
</td>
<td>


</td>
</tr>
</table>

10.1.2 Formato con ripetizione MAC e ID

<table>
<tr><th colspan="3">PACCHETTO STATO CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall’utente da 0 a 65.535. Deve essere univoco.</td><td>5</td></tr>
<tr><td>State</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più codici di stato></td><td>Valore dello stato. I numeri con decimali sono divisi dal punto, dalla parte intera. Il numero di decimali è fisso.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>
<pre>
{
  "MAC": "E82A4452061C",
  "ID": "12",
  "State": {
    "ts": "20200320T155600Z",
    "S1": 12.36,
    "S2": 38.25,
    "S3": 28.2,
    "S4": 0,
    "S5": 102,
    "S6": 152,
    "S7": 0,
    "S8": 0,
    "S9": "WINDTRE",
    "S10": 0,
    "S11": 26,
    "S12": 2,
    "S13": 5,
    "S14": 5,
    "S15": 0,
    "S16": 0,
    "S17": 0
  }
}
</pre>
</td>
<td>


</td>
</tr>
</table>

10.2 Codifica stati

Gli stati vengono codificati con un indice numerico come descritto nella tabella che segue:

Chiave	Descrizione	Unità di misura	Rappresentazione	Lunghezza MAX stringa
S1	Tensione batteria	V	Float	*1
S2	Umidità interna	%	Float	*1
S3	Temperatura interna	°C	Float	*1
S4	SMS inviati/giorno	SMS	Float	*1
S5	Puntatore inizio buffer circolare Flash dati da inviare		Int	*1
S6	Puntatore finale buffer circolare Flash		Int	*1
S7	Rollover attivo		Bit	1
S8	Chiusura Case 0: correttamente chiuso 1: aperto		Bit	1
S9	Nome operatore corrente		String	32
S10	Tipo rete 0: 2G 1: 3G		Int	*1

	2: 4G-NBIoT			
S11	Qualità/Livello del segnale (Received Signal Strength Indication) 0: -113 dBm o meno 1: -111 dBm 2÷30: -109 dBm ... -53 dBm; 2 dBm per step 31: -51 dBm o maggiore 99: non conosciuto o non determinabile	N°/30	Int	*1
S12	**2G Networks:** Bit Error Rate 0: meno di 0.2% 1: 0.2%...0.4% 2: 0.4%...0.8% 3: 0.8%...1.6% 4: 1.6%...3.2% 5: 3.2%...6.4% 6: 6.4%...12.8% 7: più del 12.8% 99: non conosciuto o non determinabile **4G Networks: Reference Signal Received Quality** 0: -4...-3 dB 1: -6...-5 dB 2: -8...-7 dB 3: -10...-9 dB 4: -13...-11 dB 5: -15...-14 dB 6: -17...-16 dB 7: -19...-18 dB 99: non conosciuto o non determinabile	%/dB	Float	*1
S13	Registrazione rete (CREG: Network registration status) 0: not registered 1: registered, home network 2: not registered, but terminal is currently searching a new operator 3: registration denied 4: unknown 5: registered, roaming	State	Int	*1
S14	Registrazione rete GPRS (CGREG: GPRS Network Registration Status) 0: not registered 1: registered, home network 2: not registered, but terminal is currently searching a new operator 3: registration denied 4: unknown 5: registered, roaming	State	Int	*1

S15	Tecnologia di accesso alla rete (Access technology of the registered network) 0: GSM 1: CAT M1 2: NB IoT	State	Int	*1
S16	Numero errori giornaliero modem	N°	Int	*1
S17	Numero errori giornaliero aggiornamento NTP	N°	Int	*1
S18	SMS inviati/giorno (var. temporanea). Il valore è copiato da S4 prima dell'azzeramento	N°	Float	*1
S19	Data azzeramento SMS (solo giorno senza mese e anno)	-	Int	*1
S20	Numero sms da inviare	N°	Int	*1
S21	N° tentativi falliti di invio SMS. Il valore viene azzerato al primo invio riuscito	N°	Int	*1
S22	Contatore temporaneo dei campionamenti dopo allarme, usato per determinare la necessità dell'invio veloce su allarme	N°	Int	*1

*1: La lunghezza della stringa inviata dipende dal valore letto e dal numero di cifre decimali impostate dall'utente.

Nota:

- le variabili S18-S21 sono state introdotte nelle versioni Bjong Flow/Stnd 1.3.8
- la variabile S22 è stata introdotta nelle versioni Bjong Flow/Stnd 1.3.9

11 Messaggio di configurazione (Cfg)

I messaggi di configurazione descrivono informazioni relative alle impostazioni dello strumento. Attraverso queste informazioni l'utente può monitorare e controllare da remoto come lo strumento sia stato configurato.
L'invio della configurazione può essere abilitato o disabilitato dall'utente.

L'intera configurazione viene inviata al server, una volta al giorno, nella trasmissione della mezzanotte.
Se viene attivata anche la configurazione da remoto, la nuova configurazione viene inviata a seguito di una richiesta di modifica.
Inoltre l'invio viene eseguito in caso di esecuzione del "test invio dati".

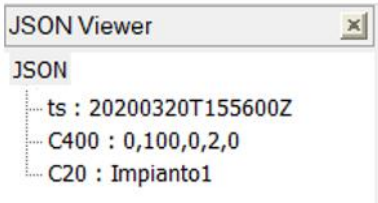
Come previsto in generale per tutti i tipi di messaggio, la configurazione può essere suddivisa in più parti se supera la lunghezza di caratteri massima prevista.

11.1 Formato unico

I messaggi di configurazione non sono influenzati dall'impostazione del formato singolo/aggregato poiché vengono sempre inviati dati relativi ad un unico timestamp. Esiste dunque un unico formato nelle varianti con e senza ripetizione MAC.

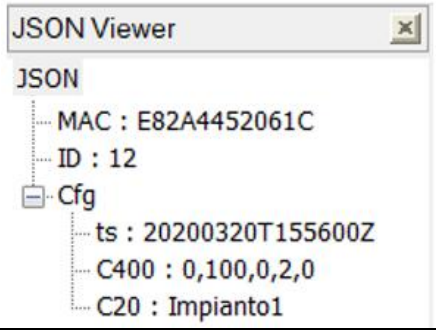
11.1.1 Formato senza ripetizione MAC e ID

<table>
<tr><th colspan="3">PACCHETTO CONFIGURAZIONE SENZA RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>Nessuna chiave iniziale</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più codici di configurazione></td><td>Valore della configurazione. Può trattarsi di una singola stringa, di un valore numerico o di un array di valori. I numeri con decimali sono divisi dal punto dalla parte intera.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

{
"ts": "20200320T155600Z",
"C400": "0,100,0,2,0",
"C20": "Impianto1"
}
	

11.1.2 Formato con ripetizione MAC e ID

<table>
<tr><th colspan="3">PACCHETTO CONFIGURAZIONE CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall’utente da 0 a 65.535. Deve essere univoco.</td><td>5</td></tr>
<tr><td>Cfg</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td>ts</td><td>Timestamp secondo ISO8601 descritto in precedenza</td></tr>
<tr><td><Uno o più codici di configurazione></td><td>Valore della configurazione. Può trattarsi di una singola stringa, di un valore numerico o di un array di valori. I numeri con decimali sono divisi dal punto dalla parte intera.</td></tr>
</table></td><td>ts: 16 caratteri.
La rimanente parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

{
"MAC": "E82A4452061C",
"ID": "12",
"Cfg": {
"ts": "20200320T155600Z",
"C400": "0,100,0,2,0",
"C20": "Impianto1"
}
}
	

11.2 Codifica variabili di configurazione

Nella tabella che segue sono riportati tutti i codici di configurazione esistenti.
Per ogni codice di configurazione (**codice**) viene fornita:

- ✓ una **descrizione**, il **formato** atteso con i possibili **valori** che quel particolare codice può assumere;
- ✓ informazioni relativamente a quando lo strumento ne esegue l'**invio (TX)**;
- ✓ informazioni relativamente a quando lo strumento ne accetta la **ricezione (RX)** per eseguire l'aggiornamento della propria configurazione

Per maggiori dettagli sull'utilizzo dei singoli parametri di configurazione, fare riferimento al manuale di istruzione dello strumento.

La disponibilità dei codici e la possibilità della modifica da remoto dipendono dalla versione implementata. L'elenco delle versioni è disponibile in fondo alla tabella dei codici.

Codice	Descrizione	TX	RX
Generali dello strumento			
C0	**Esito aggiornamento configurazione da remoto** Inviato dallo strumento solo nella prima trasmissione successiva ad una richiesta di aggiornamento da remoto. **Formato**: string o array [int, string] **Valori ammessi**: 0: aggiornamento della configurazione eseguito Es: "C0":"0" Array di due valori, di cui il primo pari a 1: errore di formattazione del pacchetto completato dal codice di configurazione contenente l'errore o dalla posizione del carattere ritenuto non conforme Es: "C0":["1","C4"] Es: "C0":["1","POS 100"] 2: Errore interno lettura pacchetto dal modem (rx-tx) 3: Errore in read function nel modem 4: Errore per modem non correttamente funzionante 5: Errore di checksum nella richiesta di cambio configurazione (parametro C10 non valido) 6: Errore ricezione più messaggi di configurazione (Clean Session=0) 7: incompatibilità tempi log/warm-up/pulizia	Solo dopo aggiorna-mento da remoto	No

C1	**Identificativo della richiesta di aggiornamento** Inviato allo strumento unitamente ad una richiesta di aggiornamento da remoto, serve per identificarla. Inviato come risposta dallo strumento, serve per riferire l'esito di aggiornamento espresso in C0 **Formato**: int **Valori ammessi**: da 0 a 255 (compresi)	Solo dopo aggiornamento da remoto	Sì
C2	**Identificativo numerico dello strumento** Selezionabile dall'utente **Formato**: int **Valori ammessi**: da 0 a 65535 (compresi)	Sempre	Sì
C3	**Modello prodotto** **Formato**: int **Valori ammessi**: 0: KAPTOR 1: KAPTORMINI 2: BJONG	Sempre	No
C4	**Tipo sensore collegato alla porta RS485** Ha significato solo su versioni FW FLOW **Formato**: int **Valori ammessi**: 0: Nessun sensore 1: Flowmeter 2: Doppler	Solo se versione Flow	Sì
C5	**Versione firmware strumento** Comprende in unica stringa, il tipo di strumento e le versioni FW e HW **Formato**: string **Esempio**: "C5":"BJONG FLOW 1.3.5-V1R1"	Sempre	No

C6	**Informazioni modem** **Formato**: array [string, string, string, string] **Parametri**: 1) modello modem 2) IMEI 3) Versione FW modem 4) Versione applicativo modem **Esempio**: "C6":["ME910","353081091510928","30.01.800","2.0.22/30.01.800"] **Note**: versione applicativo modem disponibile da FW strumento 1.3.4 (Flow/Stnd)	Sempre	No
C7	**ICCID della SIM** **Formato**: string	Sempre	No
C8	**Codice configurazione HW (8byte)** **Formato**: string	Sempre	No
C10	**Codice controllo coerenza configurazione** Si tratta di un codice calcolato in base al contenuto del messaggio inviato allo strumento in fase di aggiornamento della configurazione da remoto. **Formato**: int **Valori ammessi**: da 0 a 255 (compresi)	Mai	Sì
C20	**Nome impianto** **Formato**: string **Valori ammessi**: max 20 caratteri	Sempre	Sì A partire da v1.1
C21	**Modalità di funzionamento** **Formato**: int **Valori ammessi**: 0: Continuo 1: Intermittente	Sempre	Sì
C23	**Coordinate GPS impostate/rilevate dal modem** **Formato**: string **Esempio**: "C23":"N45 24.276 E11 55.920"	Sempre	Sì A partire da v1.1

Configurazione I/O			
C400	**Ingresso analogico 1 (AN1)** **Formato**: array [float, float, float, int, int, int] **Parametri**: 1) Inizio scala 2) Fondo scala 3) Offset 4) Tempo di warm up (secondi) 5) Tipo curva 6) Tipo ingresso **Valori ammessi**: 1) Valore numerico 2) Valore numerico 3) Valore numerico 4) int a scelta fra: 1, 2, 5, 6, 7, 8, 10, 15 5) 0: lineare 1: personalizzata 6) 0: 0-10 V 1: 4-20 mA	Se log abilitato	Sì
C401	**Ingresso analogico ingegneristico 1 (AN1I)** Corrisponde alla definizione dei punti per la generazione della curva personalizzata. Il suo utilizzo ha senso se il corrispondente ingresso analogico è impostato con curva personalizzata. La serializzazione è fatta mettendo consecutivamente la coppia valore letto/valore ingegneristico di ogni punto della curva. Necessari almeno due punti (4 valori), massimo venti (40 valori). I valori di AN1 devono essere inseriti in ordine crescente. **Formato**: array [float, float, …] **Parametri**: 1) Punto 1 - Valore AN1 2) Punto 1 - Valore ingegneristico 3) Punto 2 - Valore AN1 4) Punto 2 - Valore ingegneristico 5) … 39) Punto 20 - Valore AN1 40) Punto 20 - Valore ingegneristico **Valori ammessi**: Valori numerici. Minimo 4 (due punti), massimo 40 (20 punti)	Se log abilitato e curva personalizzata	Sì
C402	**Ingresso analogico 2 (AN2)** Come C400	Se log abilitato	Sì

C403	**Ingresso analogico ingegneristico 2 (AN2I)** Come C401	Se log abilitato e curva personalizzata	Sì
C421	**Ingresso digitale 1 (DI1)** **Formato**: array [int, float, int, float] **Parametri**: 1) Modo di funzionamento 2) Peso impulso 3) Calcolo della media 4) Moltiplicatore totalizzatore **Valori ammessi**: 1) 0: non attivo 1: stato 2: contatore 3: contatore con media su periodo campionamento 2) Valore numerico 3) 0: no 1: secondo 2: minuto 3: ora 4) Valore numerico **Note**: se il modo di funzionamento è diverso da 3 (contatore con media) il calcolo della media deve essere impostato a 0 (no)	Se log abilitato	Sì
C422	**Ingresso digitale 2 (DI2)** Come C421	Se log abilitato	Sì
C423	**Ingresso digitale 3 (DI3)** Come C421	Se log abilitato	Sì
C424	**Ingresso digitale 4 (DI4)** Come C421	Se log abilitato	Sì
Configurazione eventi digitali			
C441	**Ingresso digitale evento 1 (DIEV1)** **Formato**: int **Valori ammessi**: 1: ingresso abilitato 0: ingresso disabilitato	Se abilitato	Sì

C442	**Ingresso digitale evento 2 (DIEV2)** Come C441	Se abilitato	Sì
C443	**Ingresso digitale evento 3 (DIEV3)** Come C441	Se abilitato	Sì
C444	**Ingresso digitale cambio modalità** Come C441	Se abilitato	Sì
Configurazione pulizia			
C90	**Pulizia automatica** **Formato**: array [int, int, int, int] **Parametri**: 1) Abilitazione 2) Tempo pulizia (secondi) 3) Tempo attesa (secondi) 4) Intervallo di ripetizione (campionamenti) **Valori ammessi**: 1) 0: pulizia disabilitata 1: pulizia abilitata 2) da 1 a 255 (compresi) 3) da 0 a 255 (compresi) 4) da 1 a 60 (compresi) **Note**: se la pulizia è disabilitata, gli altri parametri sono ignorati	Se funzione abilitata	Sì
Configurazione variabili			
C300	**Periodo di campionamento** Valore comune a tutte le variabili (minuti) **Formato**: int **Valori ammessi**: int a scelta fra: 1, 2, 3,4, 5, 6, 10, 15, 20, 25, 60 (1h), 120 (2h), 240 (4h), 360 (6h), 480 (8h), 720 (12h)	Sempre	Sì
C301	**Periodo campionamento in allarme** Come C300	Se funzione abilitata	Sì

C311	**Tensione batteria (VBATT)** **Formato**: array [int, string, int, int] **Parametri**: 1) Abilitazione log 2) Unità di misura 3) Numero decimali 4) Grandezza fisica **Valori ammessi**: 1) 0: log disabilitato 1: log abilitato 2) string max 7 caratteri 3) da 0 a 4 (compresi) 4) 0: grandezza fisica generica (non modificabile) **Note**: per questo particolare codice, unità di misura e numero di decimali non sono modificabili	Se log abilitato	Sì
C312	**Umidità interna (HUM)** Come C311 **Note**: per questo particolare codice, unità di misura e numero di decimali non sono modificabili	Se log abilitato	Sì
C313	**Temperatura interna (TBRD)** Come C311 **Note**: per questo particolare codice, unità di misura e numero di decimali non sono modificabili	Se log abilitato	Sì
C314	**AN1** Come C311	Se log abilitato	Sì
C315	**AN1I** Come C311	Se log abilitato	Sì
C316	**AN2** Come C311	Se log abilitato	Sì
C317	**AN2I** Come C311	Se log abilitato	Sì
C318	**DI1** Come C311	Se log abilitato	Sì
C319	**DI2** Come C311	Se log abilitato	Sì

C320	**DI3** Come C311	Se log abilitato	Sì
C321	**DI4** Come C311	Se log abilitato	Sì
C322	**TOT1** Come C311	Se log abilitato	Sì
C323	**TOT2** Come C311	Se log abilitato	Sì
C324	**TOT3** Come C311	Se log abilitato	Sì
C325	**TOT4** Come C311	Se log abilitato	Sì
C326	**DIEV1** Come C311	Se log abilitato	Sì
C327	**DIEV2** Come C311	Se log abilitato	Sì
C328	**DIEV3** Come C311	Se log abilitato	Sì
C329	**PRES** Come C311 **Note**: per questo particolare codice, unità di misura e numero di decimali non sono modificabili	Se log abilitato	Sì
C330	**Q1** Come C311	Se log abilitato	Sì
C331	**V1** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì

C332	**L1** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
C333	**LS1** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
C334	**T1** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
C335	**QL1** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
Versione FLOW: Variabili modulo TTFM			
C337	**NET** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
C338	**POS** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
C339	**NEG** Come C311 **Note**: per questo particolare codice, unità di misura non è modificabile	Se log abilitato	Sì
C340	**Q** Come C311	Se log abilitato	Sì
C341	**VEL** Come C311	Se log abilitato	Sì

C342	**TT** Come C311	Se log abilitato	Sì
C343	**QUAL** Come C311	Se log abilitato	Sì
Configurazione di invio dati			
C800	**Periodo invio dati** Valore espresso in ore **Formato**: float o int (vedi note) **Valori ammessi**: float a scelta fra: 0.1 (6 minuti), 0.25 (15 minuti), 0.5 (30 minuti), 1, 2, 4, 6, 8, 10, 12, 24 **Note**: per le versioni FW Bjong FLOW/STND precedenti la 1.3.2 non è possibile utilizzare l'invio inferiore all'ora, quindi il valore è sempre espresso come int	Sempre	Sì
C801	**Invio dati storici** **Formato**: int **Valori ammessi**: 0: invio disabilitato 1: invio abilitato	Sempre	Sì
C802	**Invio configurazione** Come C801	Sempre	Sì
C803	**Invio eventi/allarmi** Come C801	Sempre	Sì
C804	**Invio variabili di stato** Come C801	Sempre	Sì
C805	**Flag invio Anomalie** Come C801 **Note**: parametro sempre a 0, implementazione futura	Sempre	Sì
C806	**Invio SMS** Come C801	Sempre	Sì

C807	**Max numero SMS/giorno** **Formato**: int **Valori ammessi**: da 0 a 255 (compresi)	Se invio SMS abilitato	Sì
C808	**Max tentativi invio SMS** **Formato**: int **Valori ammessi**: da 1 a 5 (compresi)	Se invio SMS abilitato	Sì
C809	**Ritardo fra tentativi** **Formato**: int **Valori ammessi**: da 1 a 60 (compresi)	Se invio SMS abilitato	Sì
C810	**Opzioni di trasmissione** **Formato**: array [int, int, int] **Parametri**: 1) Trasmissione timestamp aggregati 2) Ripetizione MAC 3) Grandezza fisica 4) Formato data/ora (a partire da v1.3) **Valori ammessi**: 1) 0: disabilitato (quindi formato TS singolo) 1: abilitato (quindi formato TS aggregati) 2) 0: disabilitata 1: abilitata 3) 0: invio disabilitato (non modificabile) 4) 0: ISO8601 1: Unix timestamp	Se trasmissione abilitata	Sì A partire da v1.1
C811	**Aggiornamento configurazione da remoto** **Formato**: int **Valori ammessi**: 0: disabilitato 1: abilitato – senza retain 2: abilitato – con retain **Note**: se la configurazione è disabilitata, non può essere abilitata da remoto (poiché lo strumento non controllerà mai eventuali messaggi di aggiornamento configurazione presenti). Se la configurazione è abilitata, può essere modificata la modalità (con o senza retain). La modalità con o senza retain deve essere coerente con quella espressa dal codice C812.	Se trasmissione abilitata	Sì A partire da v1.1

C812	**Abilitazione FOTA/OTA** **Formato**: int **Valori ammessi**: 0: tramite app 1: automatico da remoto – senza retain 2: automatico da remoto – con retain **Note**: la modalità con o senza retain deve essere coerente con quella espressa dal codice C811.	Se trasmissione abilitata	Sì A partire da v1.1
C820	**Tipo rete** **Formato**: int **Valori ammessi**: 0: 2G 1: 3G 2: 4G – NBIoT	Sempre	No
C825	**Sincronizzazione data/ora NTP** **Formato**: int **Valori ammessi**: 0: sincronizzazione disabilitata 1: sincronizzazione abilitata	Sempre	Sì A partire da v1.1
C826	**Parametri NTP** **Formato**: array [string, int, int] **Parametri**: 1) Host 2) Porta 3) Timeout (secondi) **Valori ammessi**: 1) String max 63 caratteri 2) Int da 0 a 65535 (compresi) 3) Int da 0 a 255 (compresi)	Se sincroniz. NTP abilitata	Sì A partire da v1.1
C830	**Frequenza invio veloce in caso allarme** **Formato**: array [int, int] **Parametri**: 1) Abilitazione 2) Frequenza invio (ogni n° campionamenti) **Valori ammessi**: 1) 0: invio veloce disabilitato 1: invio veloce abilitato 2) Da 3 a 99 (compresi)	Se abilitato	Sì

Configurazioni trasmissione dati			
C900	**Host name** **Formato**: string **Valori ammessi**: string max 31 caratteri	Se trasmissione abilitata	No
C901	**Modalità di connessione** **Formato**: int **Valori ammessi**: 0: HTTP (non supportato) 1: FTP (non supportato) 2: MQTT	Se trasmissione abilitata	No
C902	**Porta** **Formato**: int **Valori ammessi**: da 0 a 65535 (compresi)	Se trasmissione abilitata	No
C903	**Topic iniziale** **Formato**: string **Valori ammessi**: string max 31 caratteri **Note**: Il topic non deve terminare con il carattere “/”. Non possono essere presenti più di 5 caratteri “/”	Se trasmissione abilitata	Sì A partire da v1.1
C904	**APN SIM** **Formato**: string	Se trasmissione abilitata	No

C905	**Parametri connessione MQTT** **Formato**: array [int, int, int, int, int, int, int] **Parametri**: 1) QoS 2) Retain 3) Timeout (secondi) 4) KeepAlive (secondi) 5) Clean Session 6) Abilitazione last will 7) Modalità TLS **Valori ammessi**: 1) 0: al massimo una volta 1: almeno una volta 2: esattamente una volta 2) 0: retain disabilitato 1: reatin abilitato 3) Da 1 a 180 (compresi) 4) Da 5 a 180 (compresi) 5) 0: clean session disabilitata 1: clean session abilitata 6) 0: last will disabilitato 1: last will abilitato 7) 0: disabilitato 1: abilita solo cifratura senza nessun certificato (non vengono utilizzati certificati) 2: abilita solo certificato server (certificato server) 3: abilita certificati server e client con autenticazione certificato (certificati server, client, key) 4: abilita server e client self-signed (senza autenticazione) (certificato server, client, key) 5: abilita anche certificato pre-root con autenticazione (certificato server, client, key, e pre-root) **Note**: impostazioni last will e TLS non sono modificabili da remoto	Se trasmissione abilitata	Sì A partire da v1.1

C906	**Impostazioni last will** **Formato**: array [string, string, int, int] **Parametri**: 1) Topic 2) Messaggio 3) QoS 4) Retain **Valori ammessi**: 1) string max 31 caratteri 2) string max 31 caratteri 3) 0: al massimo una volta 1: almeno una volta 2: esattamente una volta 4) 0: retain disabilitato 1: reatin abilitato **Note**:Il topic non deve terminare con il carattere “/”. Non possono essere presenti più di 5 caratteri “/”	Se trasmissione e last will abilitati	No

<table>
<tr><th colspan="4">Configurazione allarmi</th></tr>
<tr><td>C600</td><td>Impostazione allarme 1

Formato: array [int, string, string, float, int, int, int]
Parametri:
1) Abilita
2) TAG variabile
3) Condizione
4) Soglia
5) Ritardo attivazione (numero campionamenti)
6) Ritardo rientro (numero campionamenti)
7) Abilitazione campionamento veloce
Valori ammessi:
1) 0: disabilitato
1: abilitato
2) TAG variabile (vedi apposito paragrafo)
3) Uno tra “<”, “>”, “&”, “!&”
4) Valore numerico
5) Da 1 a 150 (compresi)
6) Da 1 a 150 (compresi)
7) 0: disabilitato
1: abilitato
Note:
<u>Ingressi digitali evento</u>: per la corretta gestione degli allarmi è necessario tenere in considerazione che la condizione da usare è < o > utilizzando una corretta soglia.
Nello specifico per un allarme che deve scattare quando il digitale è a 0 (contatto aperto) deve essere impostato condizione < e soglia pari a 1. Al contrario, per un allarme che deve scattare quando il digitale è a 1 (contatto chiuso) deve essere impostato condizione > e soglia pari a 0.
<u>Ingressi digitali impostati come stato (aperto/chiuso)</u>: va seguito lo stesso criterio dei digitali evento.
<u>Variabili su modbus standard RS-485 con funzione coils (0x01) o discrete inputs (0x02)</u>: si tratta di variabili che implicitamente rappresentano più stati. Per semplicità di configurazione, l’allarme viene impostato sul singolo bit tramite le condizioni & (chiuso) e &! (aperto). Per specificare il bit di interesse è necessario procedere indicando il bit tramite rappresentazione in forma di maschera esadecimale convertita in decimale.
Esempio: allarme su bit 5 -> 10000 (binario) -> 16 (decimale) -> la soglia da inviare è 16. In pratica è possibile eseguire il calcolo 2^{bit-1}.</td><td>Se abilitato</td><td>Sì</td></tr>
</table>

	Utilizzando l'interfaccia web, per tutte e tre i casi sono previste le condizioni fittizie ON (chiuso) e OFF (aperto) che permettono l'impostazione veloce. Nel caso dei digitali non è necessario indicare nessuna soglia poiché viene calcolata in automatico; nel caso delle variabili modbus va specificato il bit di interesse che viene convertito in esadecimale.		
C601	**Impostazione allarme 2** Come C600	Se abilitato	Sì
C602	**Impostazione allarme 3** Come C600	Se abilitato	Sì
C603	**Impostazione allarme 4** Come C600	Se abilitato	Sì
C604	**Impostazione allarme 5** Come C600	Se abilitato	Sì
C605	**Impostazione allarme 6** Come C600	Se abilitato	Sì
C606	**Impostazione allarme 7** Come C600	Se abilitato	Sì
C607	**Impostazione allarme 8** Come C600	Se abilitato	Sì
C650	**Utente 1** **Formato**: array [int, string, string] **Parametri**: 1) Utente abilitato 2) Nome 3) Numero di telefono **Valori ammessi**: 1) 0: disabilitato 1: abilitato 2) String max 14 caratteri 3) String max 15 caratteri (solo cifre)	Se abilitato	Sì
C651	**Utente 2** Come C650	Se abilitato	Sì

C652	**Utente 3** Come C650	Se abilitato	Sì
Configurazione variabili modulo TTFM (non aggiornabili da remoto)			
C1000	**N° Seriale del modulo collegato allo strumento** **Formato**: string	Sempre	No
C1001	**Tipo liquido nel tubo** **Formato**: int **Valori ammessi**: 0: Acqua normale 1: Acqua di mare 2: 3: 4: 5: 6: 7: 8: Altri materiali	Sempre	No
C1002	**Diametro interno tubo** **Formato**: float	Sempre	No
C1003	**Diametro esterno tubo** **Formato**: float	Sempre	No
C1004	**Spessore tubo** **Formato**: float	Sempre	No
C1005	**Materiale condotta** **Formato**: int **Valori ammessi**: 1: Acciaio carbonio 2: Acciaio INOX 3: Ghisa 4: Ferro dolce 5: Rame 6: PVC 7: Alluminio 8: Fibrocemento 9: Fibra di vetro 10: Altri materiali	Sempre	No

C1006	**Materiale rivestimento tubo** **Formato**: int **Valori ammessi**: 0: Nessun rivestimento 1: Catrame epossidico 2: Gomma 3: Malta-Riv.cemento 4: Polipropilene 5: Polistirolo 6: Polistirene 7: Poliestere	Sempre	No
C1007	**Spessore rivestimento** **Formato**: float	Sempre	No
C1008	**Tipo trasduttori** **Formato**: int **Valori ammessi**: 1: Clamp-on TM-1 2: Clamp-on TS-2 3: Clamp-on TL-1 4: Inserzione B45	Sempre	No
C1009	**Tipo montaggio** **Formato**: int **Valori ammessi**: 1: V 2: Z 3: W 4: N	Sempre	No
C1010	**Smorzamento [s]** **Formato**: float	Sempre	No
C1011	**Cutoff [m/s]** **Formato**: float	Sempre	No
C1012	**Fattore di scala** **Formato**: float	Sempre	No

C1013	**Unità portata** **Formato**: int **Valori ammessi**: 0: m^3 1: l 2: g (Galloni US) 3: ig (Galoni UK) 4: cf (Piedi cubici)	Sempre	No
C1014	**Unità misura tempo** **Formato**: int **Valori ammessi**: 0: s 1: m 2: h 3: d	Sempre	No
C1015	**Warm-up [s]** **Formato**: float	Sempre	Sì
Configurazione variabili modulo misuratore di portata doppler			
C1100	**N° Seriale del modulo collegato allo strumento** **Formato**: string	Sempre	No
C1101	**Condotta** Impostazioni fisiche della condotta. Il primo parametro indica la forma geometrica. I parametri successivi sono in numero variabile in base alla forma e rappresentano le dimensioni (espresse in mm) **Formato**: array [int, int, …] **Valori ammessi (tra parentesi i parametri aggiuntivi attesi)**: 0: Circolare (raggio) 1: Rettangolare (base, altezza) 2: Trapezoidale (Base min., Base magg., altezza, altezza tot) 3: Profilo U, (raggio, altezza) 4: Ovoidale 3RN (raggio) 5: Ovoidale 3RM, (raggio) 6: Personalizzata (larghez.1, altez.1, larghez.2, altez.2, …) **Esempio**: “C1101”:[0,100] equivale a circolare, raggio 100mm “C1101”:[1,100,200] equivale a rettangolare, b 100mm, h 200mm	Sempre	Sì

C1102	**Altezza di montaggio [mm]** **Formato**: int **Valori ammessi**: Da 0 a 65535 (compresi) **Note**: il valore non deve eccedere l'altezza massima del canale	Sempre	Sì
C1103	**Calibrazione livello sensore [mm]** **Formato**: int **Valori ammessi**: Da -1000 a 1000 (compresi) **Note**: il valore non deve eccedere l'altezza massima del canale	Sempre	Sì
C1104	**Deposito fango [mm]** **Formato**: int **Valori ammessi**: Da 0 a 32767 (compresi) **Note**: il valore non deve eccedere l'altezza massima del canale e l'altezza di montaggio	Sempre	Sì
C1105	**Tubo pieno** **Formato**: int **Valori ammessi**: 0: no 1: si	Sempre	Sì
C1106	**Livello critico [mm]** **Formato**: int **Valori ammessi**: Da 40 a 65535 (compresi) **Note**: il valore non deve eccedere l'altezza massima del canale	Sempre	Sì
C1107	**Coefficiente di Strickler impostato** **Formato**: float **Valori ammessi**: maggiore o uguale a 0	Se abilitato l'utilizzo del coefficiente	Sì A partire da v1.2
C1108	**Unità portata** **Formato**: int **Valori ammessi**: 0: m^3 1: l 2: g 3: ig 4: cf	Sempre	Sì

C1109	**Unità misura tempo** **Formato**: int **Valori ammessi**: 0: s 1: m 2: h	Sempre	Sì
C1110	**Warm-up [s]** **Formato**: int **Valori ammessi**: uno a scelta fra 7, 8, 10, 11, 13, 15, 16, 18, 20	Sempre	Sì
C1111	**Calcolo automatico coefficiente di Strickler** **Formato**: int **Valori ammessi**: 0: calcolo disabilitato 1: calcolo abilitato	Se abilitato	Sì A partire da v1.2
C1112	**Livello massimo per il calcolo del coefficiente di Strickler [mm]** **Formato**: int **Valori ammessi**: Da 0 a 65535 (compresi) **Note**: il valore non deve eccedere l'altezza massima del canale e deve essere maggiore del livello critico	Se abilitato il calcolo automatico del coeff.	Sì A partire da v1.2
C1113	**Abilitazione Strickler sotto il livello critico** **Formato**: int **Valori ammessi**: 0: disabilitato 1: abilitato	Se abilitato	Sì A partire da v1.2
C1114	**Numero campioni per il calcolo del coefficiente di Strickler** **Formato**: int **Valori ammessi**: Da 0 a 255 (compresi)	Se abilitato il calcolo automatico del coeff.	Sì A partire da v1.2

Configurazione variabili lette tramite protocollo RS485 Modbus RTU			
C1200	**Impostazioni connessione RS485** **Formato**: array [int, int, int, int] **Parametri**: 1) Velocita 2) N° Bit Frame 3) Parità 4) N° Bit Stop **Valori ammessi**: 1) 0: 1200 1: 2400 2: 4800 3: 9600 4: 19200 5: 38400 6: 5700 7: 115200 2) 0: 7 (NON IMPOSTABILE) 1: 8 3) 0: nessuna 1: Even/pari 2: Odd/dispari 4) 0: 1 1: 2	Sempre	Sì
C1201	**Tempi RS485** **Formato**: array [int, int, int, int, int] **Parametri**: 1) Warm-up [s] 2) Timeout in modalità continua [ms] 3) Timeout in modalità intermittente [ms] 4) Ritardo richieste in modalità continua [ms] 5) Ritardo richieste in modalità intermittente [ms] **Valori ammessi**: 1) Da 0 a 255 (compresi) 2) Da 100 a 2000 (compresi) 3) Da 100 a 2000 (compresi) 4) Da 10 a 2000 (compresi) 5) Da 10 a 500 (compresi)	Sempre	Sì

C1203	**Impostazione variabile 1 (VAR1_RS485)** **Formato**: array [int, int, string, int, int, int, int, int, int, int, int, int] **Parametri**: 1) Abilitazione log 2) Funzione 3) Unità di misura 4) Numero decimali 5) Grandezza fisica 6) Tipo di dato 7) ID slave 8) Registro iniziale 9) Lunghezza (word da leggere) 10) PLC 11) Controllo errore 12) Tipo sonda **Valori ammessi**: 1) 0: log disabilitato (si possono omettere i campi successivi) 1: log abilitato 2) 0: Variabile da eliminare (si possono omettere i campi successivi in quanto qualsiasi altro campo viene automaticamente azzerato) 1: Read Coil 2: Read Discrete input 3: Read Holding register 4: Read Input register 3) String max 6 caratteri 4) Da 0 a 4 (compresi) 5) 0: grandezza fisica generica (non modificabile) 6) 0: S16 1: U16 2: Long ABCD 3: Long CDAB 4: Long BADC 5: Long DCBA 6: Real ABCD 7: Real CDAB 8: Real BADC 9: Real DCBA 7) Da 1 a 247 (compresi) 8) Da 1 a 65535 (compresi) 9) Da 1 a 2 (compresi) 10) 0: no 1: si 11) 0: no 1: si	Se funzione diversa da 0	Sì

	12) 0: 485 standard 1: Aqualabo		
C1204	**Impostazione variabile 2 (VAR2_RS485)** Come C1203	Se log abilitato	Sì
…	…	…	…
C1222	**Impostazione variabile 20 (VAR20_RS485)** Come C1203	Se log abilitato	Sì

11.3 Indicazioni di compatibilità

Versione	Descrizione	Compatibilità FW	Note
1.0	Versione iniziale	Bjong (Flow/Stnd) a partire da 1.1.0	
1.1	Aggiunta la possibilità di eseguire l'aggiornamento da remoto di codici aggiuntivi	Bjong (Flow/Stnd) a partire da 1.3.8	
1.2	Aggiunta la possibilità di eseguire l'aggiornamento da remoto di codici aggiuntivi	Bjong (Flow/Stnd) a partire da 1.3.9	
1.3	Aggiunta la possibilità di modificare il formato di data e ora		

12 Messaggio di aggiornamento di configurazione da remoto (UpdateCfg)

Lo strumento utilizza un messaggio specifico per eseguire l'aggiornamento della propria configurazione. La modalità di ricezione del messaggio dipende dal protocollo in uso. Per MQTT, avviene tramite la sottoscrizione al topic "UpdateCfg".

Lo strumento verifica la presenza di un messaggio di configurazione ad ogni invio dati. L'aggiornamento della configurazione da remoto deve essere abilitato dall'utente direttamente sullo strumento.

Se sullo strumento non è abilitata la funzionalità di aggiornamento da remoto, questa potrà essere modificata solo in locale.

La codifica dei codici contenuti nel messaggio di UpdateCfg è uguale a quella del messaggio che lo strumento invia. Vengono comunque aggiunti 2 codici specifici:

- "**C1**": indice numerico da 0 a 255 per codificare la configurazione generata dall'utente. Lo stesso indice viene inviato dallo strumento, nel medesimo campo, quando invia la nuova configurazione a seguito dell'update. In questo modo è possibile ricollegare l'esito dell'aggiornamento ad un messaggio di configurazione specifico.
- "**C10**": codice di controllo che viene aggiunto al pacchetto e che serve per verificare la congruità della configurazione. Il codice viene generato in automatico dall'applicativo web "BM Web Application" che BM Tecnologie Industriali mette a disposizione agli utenti per guidarli nella generazione dei messaggi di configurazione.

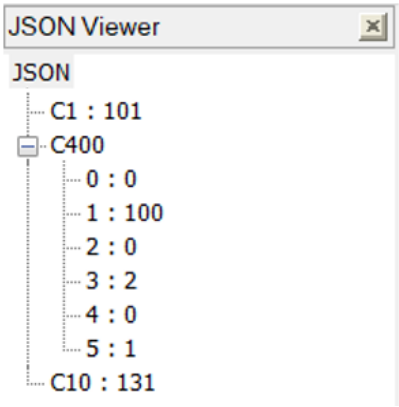
12.1 Formato unico

I messaggi di configurazione non sono influenzati dall'impostazione del formato singolo/aggregato poiché non è contemplato l'utilizzo del timestamp. Esiste dunque un unico formato nelle varianti con e senza ripetizione MAC.

12.1.1 Formato senza ripetizione MAC e ID

Il messaggio è rappresentato da un oggetto contenente i soli parametri che devono essere modificati.

<table>
<tr><th colspan="3">PACCHETTO AGGIORNAMENTO CONFIGURAZIONE SENZA RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>Nessuna chiave iniziale</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td><Uno o più codici di configurazione></td><td>Valore della configurazione. Può trattarsi di una singola stringa, di un valore numerico o di un array di valori. I numeri con decimali sono divisi dal punto, dalla parte intera.</td></tr>
</table></td><td>Parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>{
“C1”: 101,
“C400”: [
0,
100,
0,
2,
0,
1
],
“C10”: 131
}</td>
<td> 
</td>
</tr>
</table>

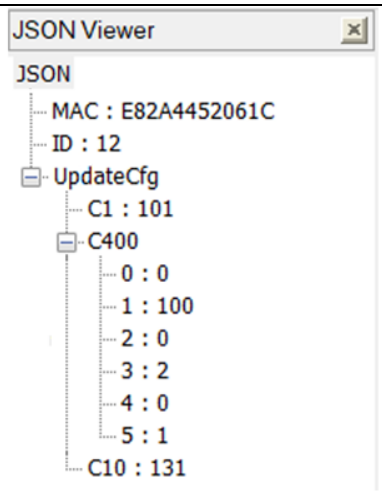
12.1.2 Formato con ripetizione MAC e ID

È possibile utilizzare il formato con ripetizione MAC, tuttavia le informazioni contenute in MAC e ID verranno ignorate dallo strumento.

<table>
<tr><th colspan="3">PACCHETTO AGGIORNAMENTO CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall’utente da 0 a 65.535. Deve essere univoco.</td><td>5</td></tr>
<tr><td>UpdateCfg</td><td>Oggetto costituito da uno o più codici di configurazione:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td><Uno o più codici di configurazione OTA></td><td>Valore della configurazione. Può trattarsi di una singola stringa, di un valore numerico o di un array di valori. I numeri con decimali sono divisi dal punto, dalla parte intera.</td></tr>
</table></td><td>Parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

<table>
<tr>
<td>

```
{
   "MAC": "E82A4452061C",
   "ID": "12",
   “UpdateCfg”: {
      “C1”: 101,
      “C400”: [
         0,
         100,
         0,
         2,
         0,
         1
      ],
      “C10”: 131
   }
}
```

</td>
<td>


</td>
</tr>
</table>

12.2 Esito configurazione da remoto

L’utente può controllare l’esito del processo di aggiornamento della configurazione tramite la configurazione che viene inviata dallo strumento subito dopo aver processato il messaggio.
In particolar modo è possibile analizzare la chiave “C0” contenente, appunto, l’esito (vedi formato messaggio di configurazione).

12.3 Generazione del messaggio di configurazione tramite applicazione WEB

BM Tecnologie Industriali mette a disposizione un tool raggiungibile da internet (web app) per generare il pacchetto di configurazione tramite interfaccia grafica. Il tool genera la nuova configurazione in modo che sia coerente con quella preesistente sullo strumento. Per poter verificare la coerenza e introdurre dei controlli è necessario partire dall'importazione dell'ultima configurazione nota dello strumento. L'importazione e l'esportazione della configurazione può avvenire in due modalità differenti:

- Gestione automatizzata: applicabile nel caso in cui il tool di generazione abbia accesso ai dati di configurazione. Questo è vero se, ad esempio, gli strumenti trasmettono in MQTT ad un broker BM o se, in caso di broker diverso, comunque ne venga fornito l'accesso. In questo modo l'applicativo è in grado di ricevere e salvare in memoria l'ultima configurazione inviata dallo strumento e la può utilizzare come base di partenza per le modifiche. La pubblicazione della nuova configurazione, allo stesso modo, può avvenire direttamente verso il broker senza nessuna ulteriore operazione manuale.
- Gestione manuale, basata su testo: qualora, per un qualunque motivo, la modalità precedente non sia applicabile, è possibile procedere in modo alternativo. Il tool è in grado di ricostruire una configurazione a partire da uno o più messaggi JSON, eseguendo copia/incolla a partire da un client MQTT connesso al broker verso cui trasmettono gli strumenti. A questo punto è possibile procedere alla determinazione dei parametri da modificare. Completata la modifica, è possibile visualizzare la stringa generata in modo da essere pubblicata manualmente sul broker.

La gestione automatizzata è più veloce e non affetta da possibili errori come, ad esempio, quelli legati alla pubblicazione del messaggio su un topic errato.

Per l'accesso all'applicazione web e per ulteriori dettagli, contattare il servizio clienti di BM Tecnologie Industriali

---

13 Messaggio di aggiornamento OTA da remoto (UpdateOTACfg)

Lo strumento utilizza un messaggio specifico per eseguire l'aggiornamento da remoto dei software (Firmware e Applicazioni). La modalità di ricezione del messaggio dipende dal protocollo in uso. Per MQTT, avviene tramite la sottoscrizione al topic "UpdateOTACfg". I parametri contenuti sono necessari allo strumento per reperire e scaricare (via FTP) i file necessari e per procedere all'aggiornamento.

13.1 Formato unico

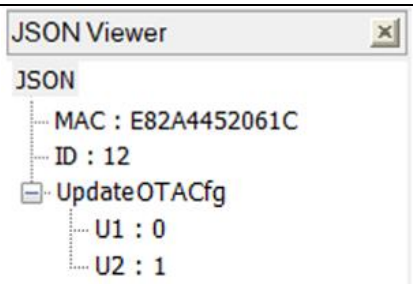
I messaggi di aggiornamento non sono influenzati dall'impostazione del formato singolo/aggregato poiché non è contemplato l'utilizzo del timestamp. Esiste dunque un unico formato nella variante con ripetizione MAC (non è prevista quella senza ripetizione MAC).

13.1.1 Formato con ripetizione MAC e ID

È necessario utilizzare il formato con ripetizione MAC, tuttavia le informazioni contenute in MAC e ID non verranno aggiornate dallo strumento.

<table>
<tr><th colspan="3">PACCHETTO AGGIORNAMENTO CON RIPETIZIONE MAC</th></tr>
<tr><th>Chiave</th><th>Descrizione</th><th>Lunghezza MAX Stringa</th></tr>
<tr><td>MAC</td><td>MAC Address dello strumento</td><td>12</td></tr>
<tr><td>ID</td><td>Identificativo strumento specificato dall'utente da 0 a 65.535. Deve essere univoco.</td><td>5</td></tr>
<tr><td>UpdateOTACfg</td><td>Oggetto costituito da:
<table>
<tr><th>Chiave</th><th>Descrizione</th></tr>
<tr><td><codici di aggiornamento OTA></td><td>Parametri indispensabili alla gestione del corretto aggiornamento.</td></tr>
</table></td><td>Parte variabile a seconda del numero di elementi presenti</td></tr>
</table>

```
{
   "MAC": "E82A4452061C",
   "ID": "12",
   " UpdateOTACfg ": {
      "U1": 0,
      "U2": 1,
         …
         …
   }
}
```



13.2 Codifica variabili di aggiornamento OTA

Nella tabella seguente sono riportati i parametri necessari per avviare una richiesta di aggiornamento del software:

Codice	Descrizione
U1	Numero intero (compreso tra 0 e 255) per l'identificazione della richiesta di aggiornamento
U2	Tipo aggiornamento: 0: nessuna operazione OTA (solo aggiornamento parametri OTA) 1: OTA Firmware modem (tipo outUpdPkg*.bin) 2: OTA Applicativo modem (mqttr.bin) 3: OTA Firmware strumento (mhx.bin)
U3	Host name
U4	Modalità di connessione FTP: 1: FTP passivo 0: FTP attivo
U5	Porta FTP
U6	Cartella FTP (dove è contenuto il file di aggiornamento)
U7	Utente
U8	Password
U9	Timeout di connessione, espresso in secondi (10..500)
U10	Codice controllo coerenza configurazione

13.3 Generazione del messaggio di aggiornamento tramite applicazione WEB

BM Tecnologie Industriali mette a disposizione un tool raggiungibile da internet (web app) per generare il pacchetto di richiesta di aggiornamento tramite interfaccia grafica. Per poter verificare la coerenza e introdurre dei controlli è necessario partire dall'importazione dell'ultima configurazione nota dello strumento. La logica di importazione della configurazione è analoga a quella relativa all'aggiornamento della configurazione (vedi).
Da interfaccia verranno proposti gli aggiornamenti compatibili (se presenti) dei diversi software presenti sullo strumento e sarà possibile:

- inoltrare il messaggio tramite la semplice pressione di un pulsante
- controllare l'esito degli aggiornamenti recenti

BM Tecnologie Industriali mette a disposizione una serie di cartelle FTP in cui lo strumento può reperire i file per l'aggiornamento.
Per ulteriori dettagli, contattare il servizio clienti di BM Tecnologie Industriali

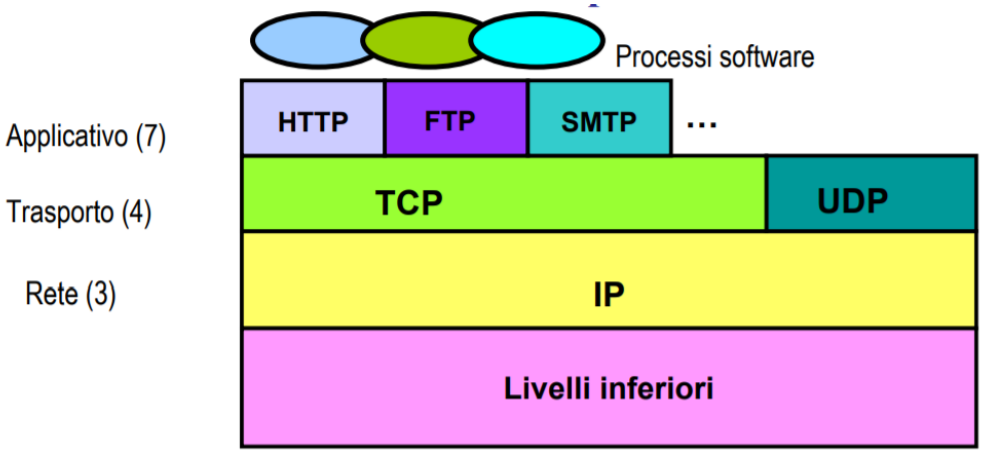
14 Messaggio di conferma OTA da remoto (OTACfg)

A seguito del tentativo di aggiornamento, lo strumento invia un messaggio specifico per fornire informazioni riguardo l'esito del processo. La modalità di ricezione del messaggio dipende dal protocollo in uso. Per MQTT, avviene tramite la pubblicazione nel topic "OTACfg".
La struttura del messaggio ed i parametri contenuti sono uguali a quelli presenti nel messaggio di aggiornamento UpdateOTACfg, con l'aggiunta di alcuni codici riportati nella tabella seguente:

Codice	Descrizione
U0	Esito aggiornamento Software. Il valore è uno dei seguenti: 0: ok 1: Errore formattazione pacchetto (codice configurazione contenente l'errore o posizione del carattere ritenuto non conforme o manca U1) Es.: {"U0":["1","POS 100"]} Es.: {"U0":["1","U4"]} Es.: {"U0" :["1","U1"]} 2: Errore interno lettura pacchetto dal modem (rx-tx) 3: Errore in read function nel modem 4: Errore per modem non correttamente funzionante 5: Errore di checksum nella richiesta di cambio configurazione 6: Errore ricezione più messaggi di configurazione (Clean Session=0) 7: Errore creazione packet data protocol context (PDP) 8: Errore ricerca operatori di rete 9: Errore attivazione APN negata (check parametri SIM) 10: Failed initializing ftp_client,check FTP timeout 11: Failed connecting: check ftp_host, ftp_port 12: Failed login: check ftp_user, ftp_password 13: Failed change directory 14: File missing: check source filename 15: File download failed 16: Incompatible versions 17: File rename not possible 18: Downloaded file not congruent 19: File reading not successful 20: Incompatible firmware version 21: Flash writing error
U1	Si tratta dell'identificativo riportato per associare questa risposta alla richiesta di aggiornamento che la ha generata
U11	Versione applicazione modem
U12	Versione FW modem
U13	Versione FW strumento

15 Protocolli di trasporto

I messaggi di testo fino ad ora analizzati sono quelli che gli strumenti possono inviare o ricevere, indipendentemente dal protocollo di scambio utilizzato.
In questo capitolo vengono approfonditi gli aspetti dei protocolli usati dagli strumenti di BM Tecnologie Industriali, specificando come i singoli messaggi debbano essere recapitati.
Nello specifico si fa riferimento a quei protocolli applicativi che utilizzano il livello di trasporto TCP. Il livello di trasporto ha il compito di instaurare un collegamento logico tra le applicazioni residenti su host remoti.
I processi in esecuzione su sistemi remoti possono scambiarsi informazioni e servizi mediante una rete: l'interazione avviene mediante lo scambio di messaggi; i protocolli applicativi sono le regole e i formati con i quali i processi costruiscono i messaggi e ne interpretano il significato.



15.1 MQTT

15.1.1 Generalità

MQTT (Message Queue Telemetry Transport) è un protocollo ISO standard (ISO/IEC PRF 20922) di messaggistica leggero basato sul meccanismo di pubblicazione/sottoscrizione (publish-subscribe).
Consente di eseguire la comunicazione tra entità in situazioni in cui è richiesto un basso consumo energetico e, generalmente, la banda è limitata.
A differenza del protocollo HTTP, in cui le entità che comunicano sono due con ruolo prefissato (client e server), in questo caso la comunicazione può avvenire tra più entità grazie alla mediazione di un cosiddetto broker, ovvero di un responsabile della distribuzione dei messaggi.
Ogni entità presente nel sistema può eseguire, di volta in volta ed in base alle esigenze, sia pubblicazione che sottoscrizione di messaggi. Il sistema di distribuzione gestito dal broker prevede la suddivisione dei messaggi pubblicati in topic (argomenti) eventualmente innestati a più livelli.

Un invio di messaggio è fondamentalmente costituito da:

- **topic**: argomento del messaggio che, nel caso applicativo, è prefissato in base alla configurazione dello strumento, al suo identificativo e al tipo di messaggio inviato;

- **payload**: ovvero il messaggio testuale (in JSON) che definisce le informazioni trasmesse.

Ulteriori informazioni sul protocollo MQTT sono inserite in appendice

15.1.2 Requisiti end point

L'end point deve essere predisposto per la ricezione (tramite sottoscrizione di topic) di messaggi dal broker.
Il contenuto del messaggio, come già visto, è in JSON. Il topic di pubblicazione corrisponde a <roottopic>/<MAC>/<tipo>, dove:

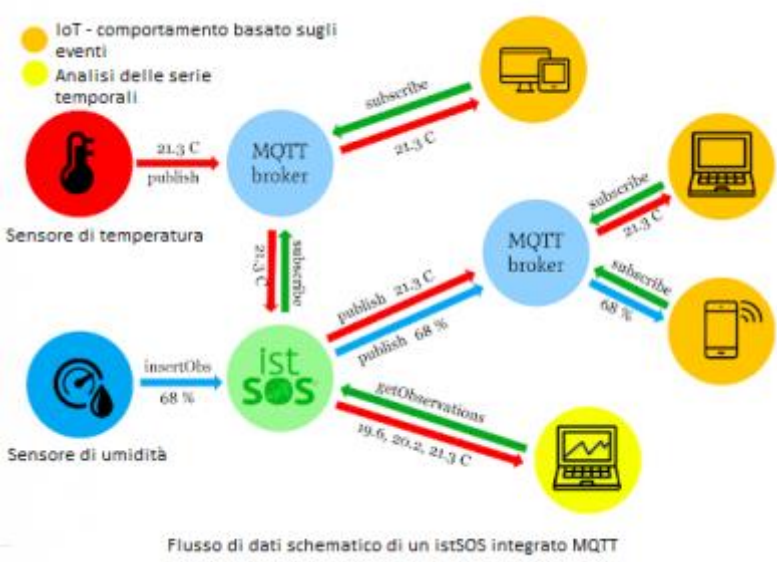
- rootopic: è l'argomento radice comune a tutte le trasmissioni eventualmente a più livelli (configurabile sullo strumento)
- MAC: è il MAC address dello strumento, determinato in modo automatico e non modificabile
- Tipo: corrisponde al tipo di messaggio trasmesso

I tipi di messaggio ed il contenuto del json inviati sono riportati nella tabella seguente:

Tipo	**Formato messaggio (json)**	**Operazione attesa**
HData	Messaggio dei dati storici	Salvataggio dei dati storici dello strumento
Alrm	Messaggio degli allarmi	Salvataggio degli allarmi su soglia generati dallo strumento
State	Messaggio di stato	Salvataggio delle informazioni di stato
Anom	Messaggio di anomalia	*non ancora implementato
Cfg	Messaggio di configurazione	Salvataggio delle informazioni di configurazione
UpdateCfg	Messaggio di aggiornamento configurazione	Pubblicazione da parte dell'utente per la lettura da parte dello strumento per l'aggiornamento della configurazione da remoto
OTACfg	Messaggio di esito di aggiornamento	Salvataggio (per consultazione) dell'esito dell'aggiornamento di FW e/o applicativi
UpdatOTACfg	Messaggio di aggiornamento da remoto	Pubblicazione da parte dell'utente per la lettura da parte dello strumento per l'aggiornamento di FW e/o applicativi da remoto

16 Appendice: generalità del protocollo MQTT

16.1 Introduzione



Flusso di dati schematico di un istSOS integrato MQTT

I concetti base dell'MQTT

MQTT sta per Message Queuing Telemetry Transport. È un semplice e leggero protocollo di messaggistica progettato per essere utilizzato con dispositivi a larghezza di banda ridotta, il che lo rende perfetto per le applicazioni IIoT (Industrial Internet of Things).

Cos'è l'MQTT?

L'MQTT è un protocollo di comunicazione aperto che è stato approvato come uno standard dall'Organizzazione per il Progresso degli Standard di Informazione Strutturate (OASIS) nel 2014. L'organizzazione descrive l'MQTT come un leggero, aperto e semplice protocollo di trasporto della messaggistica di pubblicazione/sottoscrizione. È una soluzione ottimale per le implementazioni machine-to-machine e IoT in cui si desidera un ingombro ridotto del codice e dove è disponibile una larghezza di banda di rete limitata. Alcune delle sue caratteristiche più distintive includono:

- Il modello di messagistica di tipo pub/sub (pubblicazione/sottoscrizione);
- Un trasporto di messaggistica indipendente dal contenuto del pacchetto (payload);
- Traffico di rete limitato con un piccolo sovraccarico di trasporto;
- La capacità di recapitare i messaggi almeno, al massimo o esattamente una volta per ottimizzare il modo in cui le informazioni specifiche vengono trasferite;
- Il servizio di notifica quando si verifica una disconnessione anomala.

Cosa differenzia l'MQTT dai tradizionali protocolli IoT?

La caratteristica più sorprendente che distingue l'MQTT dai tradizionali protocolli di comunicazione è la sua dipendenza dal modello di pub/sub piuttosto che dall'architettura client/server più familiare. Questo modello alternativo offre alcuni benefici sostanziali che rendono l'MQTT attraente per una varietà di

ragioni. In un'implementazione MQTT, sia i publisher che gli subscriber che eseguono una libreria MQTT sono indicati come client.

Nel modello client/server, i client comunicano direttamente con un end-point o un server. I modello di pubblicazione/sottoscrizione separa il mittente del messaggio (editore) dai destinatari (sottoscritti), i publisher e gli subscriber non si conoscono e non stabiliscono mai una connessione diretta. Un terzo componente della connessione, conosciuto come broker, filtra i messaggi in arrivo e li distribuisce agli subscriber appropriati.

Disaccoppiamento

Il disaccoppiamento di publisher/subscriber ha diverse dimensioni che contribuiscono alla flessibilità del protocollo MQTT.

- La separazione spaziale viene implementata tramite l'utilizzo del broker. I publisher e gli subscriber devono solo sapere come contattare il broker, non l'un l'altro.
- L'MQTT offre il disaccoppiamento in base al tempo. Ciò significa che un broker può memorizzare i messaggi per i client che non sono in linea e consegnarli quando la risorsa è disponibile.
- Il disaccoppiamento della sincronizzazione significa che le operazioni sui componenti non devono essere interrotte durante l'attesa di ricevere o pubblicare un messaggio in modo che coincida con la natura asincrona della maggior parte delle librerie client.

Filtraggio

Il filtraggio dei messaggi svolto dal broker è essenziale per l'efficienza dei sistemi utilizzando l'MQTT. Sono disponibili per il broker diverse opzioni di filtraggio.

- Il filtro basato sull'oggetto utilizza l'oggetto o l'argomento per determinare quali subscriber devono ricevere un messaggio specifico. I clienti si iscrivono al broker per argomenti di interesse. Tutti i client che si iscrivono a un particolare argomento riceveranno tutti i messaggi relativi a quell'argomento.
- Il filtro basato sul contenuto utilizza il contenuto del messaggio per determinare i destinatari. Il contenuto deve essere noto prima che il messaggio venga inviato e non può essere crittografato o modificato facilmente.
- Il filtro basato sul tipo si utilizza con il linguaggio orientato agli oggetti e utilizza il tipo di classe di un messaggio o evento per identificare gli subscriber corretti.

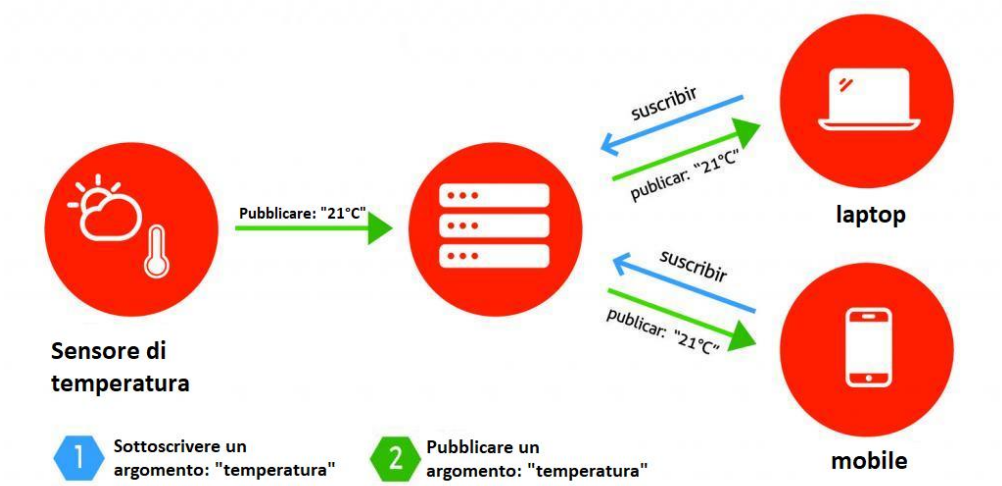
L'MQTT gestisce messaggi in un modo sostanzialmente diverso rispetto a un protocollo client/server tradizionale. In un'architettura client/server, ogni messaggio viene consumato da un singolo client, ma nel modello pub/sub molti subscriber possono ricevere lo stesso messaggio. I messaggi in un'implementazione MQTT non rimangono in coda fino a che non vengono consumati come nel caso del modello client/server. È possibile che nessun client elabori i messaggi se non ci sono iscritti per quell'argomento. Gli argomenti possono essere creati in modo dinamico con l'MQTT, che elimina la necessità di creare code con un nome prima che i messaggi possano essere inviati e ricevuti.

I vantaggi dell'MQTT

Le caratteristiche dell'MQTT e del modello pub/sub offrono alcuni vantaggi che dovrebbero essere considerati quando si implementano sistemi per IoT. Tra questi ci sono:

- L’abilità di distribuire informazioni in maniera più efficiente;
- Scalabilità aumentata attraverso l’abilità di parallelizzare le attività del broker;
- Leggero sovraccarico e requisiti di archiviazione del codice;
- Sicurezza basata sul permesso per proteggere la comunicazione;
- Requisiti minimi di larghezza di banda, consentendo alle organizzazioni di massimizzare le risorse date.

16.2 Funzionalità MQTT



L’MQTT è un protocollo di messaggistica pensato per un utilizzo con dispositivi a bassa larghezza di banda. La sua semplice e leggera natura rende il suo utilizzo perfetto nelle implementazioni di automazione industriale e IoT (Internet of Things). Un uso efficacie dell’MQTT richiede una comprensione di alcuni termini base e di come interagiscono all’interno del protocollo per inviare e ricevere i messaggi che possono controllare il funzionamento dei dispositivi automatizzati in rete.

Clients e Brokers

La maggioranza di protocolli di messaggistica tradizionale segue un’architettura client-server dove un client comunica direttamente con un end-point. Questa metodologia può essere un fattore limitante che non riesce ad affrontare adeguatamente i requisiti di un’infrastruttura di automazione industriale dinamica. L’MQTT offre una strategia di comunicazione innovativa che permette maggiore flessibilità attraverso l’uso di clients e brokers utilizzando la struttura publish e subscribe (pub/sub) per scambiare informazioni.

Il modello pub/sub disaccoppia il mittente e i destinatari di un messaggio che non sono mai in diretto contatto l’uno con l’altro. I broker sono responsabili del filtraggio e della direzione dei messaggi pubblicati ai sottoscritti appropriati.

Clients

Un client MQTT può essere definito come qualsiasi dispositivo che esegue una libreria MQTT che si connette a un broker MQTT tramite rete basata su TCP/IP. I dispositivi che pubblicano e sottoscrivono messaggi sono entrambi considerati client. La differenza è che un publisher sta generando e pubblicando

messaggi, mentre uno subscriber li sta ricevendo. Un singolo client può essere sia publisher che subscriber.
I clients possono variare di dimensioni, da piccoli dispositivi con risorse limitate a computer che eseguono client MQTT per i test. Il lato client del protocollo MQTT è facile da implementare ed è leggero, il che lo rende ideale per dispositivi basati su microcontrollore.

Brokers
Il broker in un'implementazione MQTT è responsabile di assicurare che i messaggi vengano inviati ai client corretti. Il broker è un server che riceve tutti i messaggi, esegue il filtraggio attraverso il quale determina i client che sono iscritti a un particolare messaggio e invia le informazioni a quel client. Milioni di clients possono essere connessi simultaneamente ad un broker.
Un broker MQTT esegue anche altre funzioni, come ad esempio conservare i dati per i clients che hanno sessioni permanente. È anche responsabile dell'autenticazione e autorizzazione dei clients nel sistema. Alcune caratteristiche importanti di un broker sono scalabilità, monitoraggio semplice, resistenza ai guasti e l'abilità di essere integrato a sistemi backend. L'integrazione è un fattore chiave, poiché il broker solitamente è accessibile direttamente da Internet e ha bisogno di avere la capacità di gestire potenzialmente molti clients.

16.2.1 Collegamento ad un broker

I clients non si collegano mai direttamente. Le connessioni MQTT avvengono tra un singolo client e il broker. Un client invia un messaggio CONNECT al broker per avviare la connessione e riceve in cambio un messaggio CONNACK e un codice di stato. Le connessioni stabilite vengono mantenute aperte dal broker fino a quando la connessione decade, o quando il client invia un comando di scollegamento.
La richiesta di collegamento contiene le informazioni necessarie per stabilire la connessione e un messaggio non valido comporterà la chiusura della connessione da parte del broker. Gli elementi più importanti contenuti in un messaggio CONNECT sono:

1. **Client ID** – questa è l'identificazione del client usata dal broker per identificare il client e il suo stato corrente.
2. **Clean session** – il flag di sessione "clean" determina se il client sta richiedendo una sessione permanente. Un'impostazione di false indica una sessione permanente in cui il broker archivia le sottoscrizioni e i messaggi persi per il client.
3. **Username/Password** – questo elemento viene utilizzato per l'autorizzazione e autenticazione del client e può essere inviato in modo trasparente o crittografato.
4. **Will message** – l'ultimo messaggio will (di volontà) viene utilizzato per notificare ad altri client quando un client si disconnette in modo inappropriato.
5. **KeepAlive** – questo parametro è un intervallo di tempo misurato in secondi che specifica il periodo di tempo più lungo che un broker e un client trascorrono senza inviare un messaggio. I client e il broker si scambiano un ping a vicenda per stabilire che sono ancora disponibili.

Il broker risponde al messaggio CONNECT del client con un messaggio CONNACK che contiene due pezzi di dati, che sono:

1. **Flag di sessione presente** – questo flag informa il client se c'è già una sessione persistente da una richiesta precedente.

2. **Codice di ritorno connessione** – il codice di ritorno indica se il tentativo di connessione è andato a buon fine. Se la connessione viene rifiutata, il codice di connessione indicherà un motivo, ad esempio un nome utente o una password errati.

16.2.2 Pubblicazione e sottoscrizione di argomenti (topics)

In un'implementazione MQTT i messaggi vengono scambiati attraverso il metodo pub/sub. I clients pubblicano messaggi ai quali altri clients si sottoscrivono.

Pubblicazione

Dopo il collegamento con il broker, un client può pubblicare messaggi. Ogni messaggio deve avere un argomento che può essere utilizzato per inoltrare il messaggio ai client sottoscritti. Il messaggio ha anche un payload che contiene dati che possono essere inviati in molteplici formati.
Un messaggio PUBLISH è composto dai seguenti elementi:

- **Identificatore del pacchetto** – identifica un messaggio mentre si sposta tra client e broker;
- **Nome dell'argomento** – una stringa di testo costruita gerarchicamente che utilizza barre come delimitatori;
- **QoS** – indica la qualità del livello di servizio del messaggio;
- **Retain flag** – indica se il messaggio deve essere salvato dal broker come ultimo valore valido noto per un particolare argomento;
- **Payload** – il messaggio da trasferire;
- **Flag di duplicazione** – indica se il messaggio è un duplicato inviato perché il messaggio originale non è stato riconosciuto.

Sottoscrizione

I clients inviano messaggi al broker, indicando l'argomento al quale sono interessati. Un messaggio di sottiscrizione contiene due componenti: un pacchetto identificativo e una lista di subscribers. Tantissime sottoscrizioni possono essere contenute in un singolo messaggio, ed ognuna specifica un argomento e il livello QoS.
Il broker risponde ad una sottoscrizione con un messaggio SUBACK per confermare la ricezione del messaggio e fornisce codici di ritorno che informano il client se una sottoscrizione è stata rifiutata e il livello di QoS che è stato concesso.

Annullamento della sottoscrizione

Un messaggio di DISISCRIZIONE cancella un'esistente sottoscrizione di un client. Contiene un pacchetto identificativo e una lista di argomenti che non sono più di interesse del client. Il broker risponde inviando un messaggio UNSUBACK di conferma al client.

16.3 Tutto sugli argomenti (topic)

Gli argomenti sono il meccanismo con cui il broker filtra i messaggi e determina che clients devono riceverli. Un argomento MQTT è una stringa UTF-8 costituita da uno o più livelli di argomento separati da

una barra. Gli argomenti non devono essere inizializzati prima dell'uso e possono essere creati dinamicamente immediatamente prima della pubblicazione.
Un argomento deve contenere almeno un carattere e una stringa può avere spazi vuoti. Fanno distinzione tra maiuscole e minuscole, quindi Questo/Argomento e questo/argomento sono due argomenti distinti. Una barra in avanti "/" di per sé è un argomento valido.
Sono disponibili caratteri jolly a livello singolo e multilivello per consentire ai client di iscriversi a più argomenti contemporaneamente. I caratteri jolly vengono utilizzati solo durante l'iscrizione agli argomenti.

Carattere jolly a livello singolo: +
Di seguito è riportato un esempio di un carattere jolly a livello singolo che sostituisce un livello di argomento in un messaggio di sottoscrizione.
Controllo/+/linea di base
Le sottoscrizioni che soddisfano queste richieste includono:

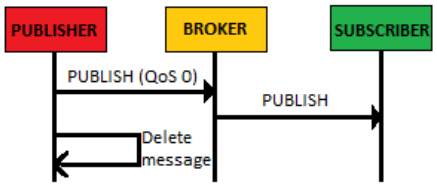
- Controllo/pressione/linea di base
- Controllo/temperatura/linea di base
- Controllo/tempismo/linea di base
- Etc.

Carattere jolly a più livelli: #
Un carattere jolly a più livelli può essere inserito solo come ultimo carattere in un argomento e deve essere preceduto da una barra. Le sottoscrizioni che includono un carattere jolly a più livelli riceveranno tutti i messaggi per l'argomento con il modello che precede il carattere jolly indipendentemente dalle dimensioni o dalla profondità dell'argomento. L'uso non corretto di un carattere jolly a più livelli può causare la sottoscrizione inavvertita di un client a molti argomenti irrilevanti.

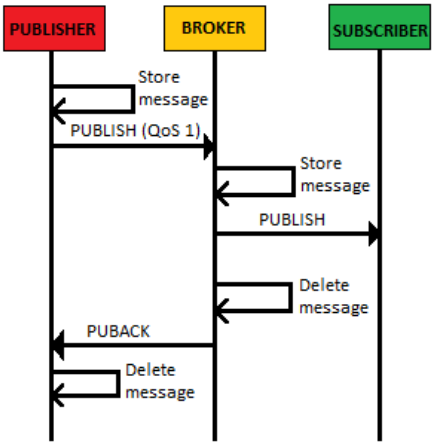
Il simbolo $ è riservato agli argomenti che riguardano le statistiche interne del broker MQTT. I client non possono pubblicare messaggi su questi argomenti.
Vale la pena prendere in considerazione alcuni suggerimenti sugli argomenti. Sebbene la barra in avanti (/) sia consentita nel linguaggio MQTT, non è consigliata. Lo stesso si può dire per l'utilizzo di spazi all'interno di un argomento. Questi costrutti introducono complessità non necessarie senza fornire alcun vantaggio.

16.4 Livelli QoS

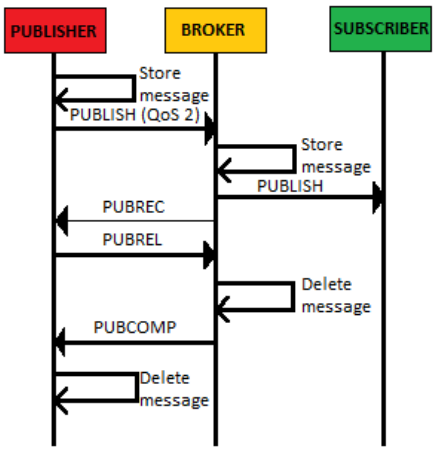
QoS 0: Al massimo una volta (fire & forget)



QoS 1: Almeno una volta



Qos 2: Esattamente una volta



16.4.1 La qualità del servizio MQTT

La possibilità di inviare messaggi che vengono gestiti in modi diversi è uno dei fattori che distinguono l'MQTT dagli altri protocolli di comunicazione. Questo servizio è chiamato QoS (Quality of Service) nell'MQTT. Questo paragrafo esaminerà in modo approfondito cosa sono i livelli QoS e come migliorano la comunicazione di rete.

Cos'è la QoS?

La QoS (Quality of Service) è un attributo che viene assegnato a un singolo messaggio quando viene pubblicato. La QoS è essenzialmente un accordo tra mittente e destinatario che definisce il modo in cui un messaggio viene consegnato. Consente ai client di selezionare un livello di servizio che tenga conto dell'affidabilità della rete e della logica di programmazione. L'MQTT è in grado di ritrasmettere messaggi e garantire la consegna, quindi l'aggiunta della QoS facilita la comunicazione utilizzando reti inaffidabili.
La QoS definisce fino a che punto il broker o il client si spingerà per garantire la ricezione di un messaggio. Il livello QoS iniziale è impostato dal client che pubblica il messaggio prima che venga inviato al broker. Il livello QoS può essere declassato dai sottoscritti ai messaggi. I broker consegnano messaggi in base alla QoS definita dal sottoscritto, non dall'editore.

16.4.2 QoS 0

Quando la QoS è impostata su 0, un messaggio viene recapitato al massimo una volta. Potrebbe non venire consegnato affatto. Non c'è conferma della consegna con questo livello di QoS. Il messaggio non viene archiviato e può essere perso se il client si disconnette o il server non funziona.
Questo livello di QoS viene talvolta definito "fire and forget" ed è la modalità di trasferimento dati più veloce in un'implementazione MQTT. I messaggi su QoS 0 non devono essere inoltrati a un client. Un messaggio può essere eliminato dal server se il client previsto viene disconnesso quando il server riceve la pubblicazione.

16.4.3 QoS 1

QoS 1 è la modalità di trasferimento predefinita dell'MQTT e indica che il messaggio viene sempre consegnato almeno una volta. La ricezione del messaggio deve essere confermata. La mancata ricezione di un riconoscimento comporta il reinvio del messaggio con il flag DUP impostato. Questo processo si ripete fino a quando il messaggio viene riconosciuto e può portare all'invio e all'elaborazione dello stesso messaggio più volte.
I messaggi impostati su QoS 1 vengono archiviati localmente presso il mittente e il destinatario fino a quando non vengono elaborati. Un destinatario elimina il messaggio dopo averlo elaborato. Quando il destinatario è un broker, il messaggio viene pubblicato ai sottoscritti appropriati. I client che ricevono un messaggio lo consegnano all'applicazione di sottoscrizione. Alla cancellazione del messaggio, il destinatario invia una conferma al mittente. I mittenti eliminano un messaggio a QoS 1 dopo aver accettato la conferma dal destinatario.

16.4.4 QoS 2

I messaggi inviati a QoS 2 vengono sempre consegnati esattamente una volta. È la modalità di trasferimento più affidabile e lenta in una rete MQTT. I messaggi vengono archiviati localmente presso il mittente e il destinatario finché non vengono elaborati. Tra il mittente e il destinatario vengono eseguite almeno due coppie di trasmissioni prima che un messaggio venga eliminato dal mittente. Il messaggio può essere elaborato dal destinatario dopo che la prima o la seconda trasmissione è stata completata, purché non lo elabori più di una volta.
La prima coppia di trasmissioni comprende il mittente che invia il messaggio e ottiene la conferma che il destinatario lo ha memorizzato. Se il mittente non riceve la conferma, i messaggi vengono ritrasmessi con il flag UDP impostato fino a quando non viene ricevuto un riconoscimento. Nella seconda coppia di trasmissioni, il mittente informa il destinatario che può completare l'elaborazione del messaggio inviando un messaggio PUBREL. Il messaggio PUBREL verrà inviato ripetutamente fino a quando non ne viene confermata la ricezione, a quel punto il mittente cancella il messaggio.
Quando il destinatario del messaggio è un broker, svolge la sua funzione e pubblica il messaggio agli subscribers. Un destinatario client consegna il messaggio all'applicazione di sottoscrizione. Quando il destinatario ha terminato l'elaborazione di un messaggio, invia un messaggio di completamento al mittente. Il livello QoS 2 viene utilizzato per garantire che i messaggi non vengano persi o elaborati più di una volta.

16.4.5 QoS in azione

QoS 0 fornisce la consegna best-effort senza alcuna garanzia di successo. Non c'è conferma della ricezione dei messaggi e non vengono archiviati o ritrasmessi in caso di errore.
QoS 1 garantisce che un messaggio venga recapitato con successo al destinatario almeno una volta. Un mittente memorizza un messaggio fino a quando il destinatario non conferma la ricezione restituendo un pacchetto PUBACK. L'identificatore del pacchetto PUBLISH viene utilizzato nella creazione del pacchetto PUBACK. Il pacchetto PUBLISH viene reinviato se il mittente non riceve il pacchetto PUBACK in un lasso di tempo ragionevole.
È necessario il processo four way handshake per completare l'elaborazione dei messaggi inviati a livello QoS 2. L'identificatore del pacchetto del messaggio PUBLISH originale viene utilizzato per coordinare la consegna. Il flusso dei messaggi è conforme al modello seguente. Un destinatario riceve un pacchetto PUBLISH da un mittente e risponde con un pacchetto PUBREC che conferma la ricezione del pacchetto originale. L'assenza di un pacchetto PUBREC fa sì che il messaggio venga rinviato con un flag DUP. Quando viene ricevuto il pacchetto PUBREC, il mittente può eliminare il pacchetto PUBLISH iniziale. Il mittente, quindi, memorizza il pacchetto PUBREC e risponde con un pacchetto PUBREL. Alla ricezione del pacchetto PUBREL, il destinatario scarta tutti gli stati memorizzati e risponde con un PUBCOMP al mittente. Il ricevitore memorizza un riferimento al pacchetto identificato nel messaggio PUBLISH originale fino all'invio del pacchetto PUBCOMP, per evitare di elaborare il messaggio una seconda volta.

16.4.6 Le pratiche migliori per la scelta di un livello di QoS

Il vantaggio di avere più livelli QoS è che possono essere utilizzati per affrontare diverse situazioni o requisiti. Di seguito sono riportate alcune linee guida su quando utilizzare ciascun livello di qualità del servizio (QoS).

- Il livello QoS 0 può essere utilizzato quando esiste una connessione affidabile e stabile tra il mittente e il destinatario, ad esempio su una rete cablata. Il sistema deve essere in grado di tollerare messaggi persi occasionali. Non ci sono messaggi in coda con questo livello di QoS.
- QoS 1 dovrebbe essere usato quando ogni messaggio deve passare e i duplicati possono essere gestiti dai destinatari. Questo è il livello di servizio più comunemente utilizzato poiché la consegna dei messaggi è garantita ed è più veloce del livello QoS 2.
- Infine, QoS 2 viene utilizzato quando un'applicazione deve assicurarsi di ricevere un messaggio solo una volta. Il sovraccarico di questo livello di servizio deve essere considerato quando viene selezionato, poiché l'elaborazione dei messaggi sarà più lenta rispetto ai livelli inferiori.

La disponibilità di diversi livelli QoS consente di ottimizzare i sistemi in base ai requisiti delle applicazioni sottostanti e all'affidabilità della rete. Essere in grado di selezionare la modalità di consegna dei messaggi è una delle caratteristiche più interessanti del protocollo MQTT e lo rende una scelta eccellente per le implementazioni IoT e di automazione industriale.

17 Appendice: TSL

Cosa serve?

TLS è un protocollo crittografico di comunicazione dati. È il successore del protocollo SSL, ormai deprecato. L'uso di TLS consente una comunicazione sicura end-to-end, dal mittente al destinatario, assicurando autenticazione, integrità dei dati e confidenzialità.

La connessione è Privata perché i dati sono crittografati tra il client e il server. Le parti comunicanti vengono autenticate per garantire che ciascuna parte stia parlando con l'ospite previsto. La connessione è affidabile in quanto nessuna modifica della comunicazione può avvenire senza rilevamento.

Il processo di trasmissione sicura dei dati tra il client e il server è gestito da ciò che il protocollo SSL/TLS definisce "cipher suite" (suite di cifratura). Questa suite di cifratura è composta da più parti con vari algoritmi per ciascuna parte:

Authentication Algorithm - Determina come viene eseguita l'autenticazione di entrambe le parti (relativa ai certificati) — Fornisce l'autenticazione

Key Exchange Algorithm - Determina come vengono scambiate le chiavi di crittografia (chiavi utilizzate per crittografare i dati)

Bulk Encryption Algorithm - Determina quale algoritmo utilizzare per crittografare i dati tra client e server — Rende i dati privati

Message Authentication Code Algorithm - Determina quale algoritmo per verificare l'integrità dei dati — Rende i dati affidabili

Prima che un'applicazione client e un server possano scambiare dati su una connessione TLS, queste due parti devono prima concordare un insieme comune di algoritmi per proteggere la connessione. Se le due parti non riescono a raggiungere un accordo, non verrà stabilita una connessione. Il processo di negoziazione avviene durante quella che è comunemente nota come stretta di mano SSL. Nell'handshake SSL/TLS, il client inizia informando il server quali suite di crittografia supporta. Le suite di crittografia sono generalmente disposte in ordine di sicurezza con la suite di crittografia più sicura come prima scelta. Il server confronta quindi quelle suite di crittografia con le suite di crittografia abilitate dalla sua parte. Non appena trova una corrispondenza, informa il client e gli algoritmi della suite di crittografia scelta utilizzati nella comunicazione successiva.

Ingredienti per il TLS

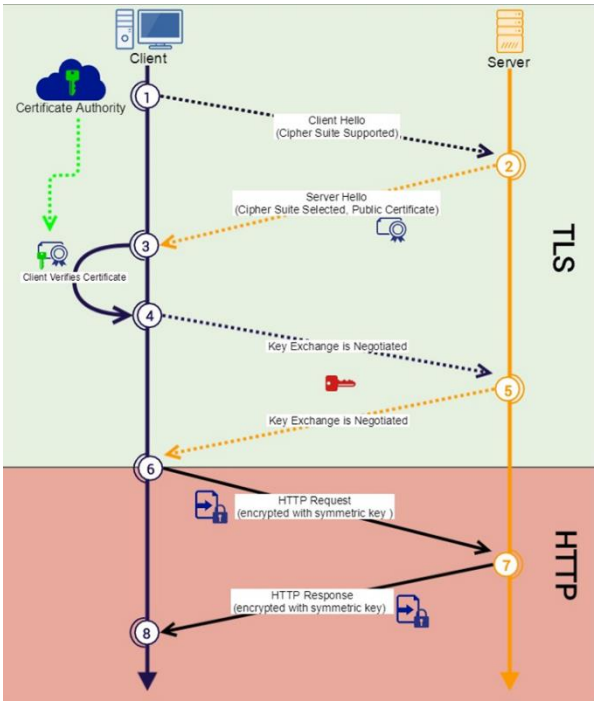
Quando un utente si connette a un server vuole essere sicuro di parlare con il server autorizzato e non con un imitatore. Per fare ciò, ogni server è in grado di ottenere un certificato che ne attesta l'autenticità che un client può controllare e verificare. Questi certificati non sono falsi poiché sono protetti crittograficamente da un'autorità di certificazione responsabile dell'emissione dei certificati. L'autorità di

certificazione (CA) è una delle poche organizzazioni situate a livello globale che tutti i dispositivi che si connettono a Internet accettano di considerare attendibili e accettano di considerare attendibili tutti i certificati emessi dall'autorità di certificazione. Un certificato emesso da un'autorità di certificazione, non può essere duplicato da un utente malintenzionato.

Quando un client si connette al server, il server risponde con il certificato che ha e se il nome di dominio (google.com) corrisponde al dominio elencato sul certificato, il client può essere sicuro di parlare con la parte corretta e non con qualche impostore.

Il certificato (CA Server) fornisce anche un modo per eseguire lo scambio di chiavi su un canale sicuro poiché il certificato utilizza una coppia di chiavi asimmetriche come parte dell'algoritmo di autenticazione. La matematica alla base di questo metodo esula dall'ambito di questo approfondimento, ma in sostanza ci sono due parti nella coppia di chiavi asimmetriche, una chiave privata che solo il server conosce e una chiave pubblica che è il certificato utilizzato per verificare l'autenticità del server. Le chiavi sono costruite in modo tale che quando i dati vengono "firmati" (crittografati) con la chiave privata del server, solo la chiave pubblica può decifrarli, e viceversa, dove qualcosa crittografato con la chiave pubblica è leggibile solo con la chiave privata.

Una volta che la chiave di sessione è stata scambiata tra il server e il client, il Bulk Encryption Algorithm protegge tutte le comunicazioni successive per la sessione di comunicazione. A differenza dell'algoritmo di autenticazione (Authentication Algorithm), il Bulk Encryption Algorithm utilizza un algoritmo di crittografia simmetrica, il che significa che la stessa chiave (che è già stata negoziata dall'algoritmo di scambio di chiavi) viene utilizzata sia per crittografare che per decrittografare i dati. Questo algoritmo utilizza una matematica intelligente per garantire che anche se un valore nel set di dati viene alterato, l'algoritmo rileverà l'alterazione e richiederà nuovamente i dati o semplicemente chiuderà la connessione.



Work flow

- Un client tenta di aprire una comunicazione con un server di destinazione. In una parte del saluto, il client invia un elenco delle sue suite di crittografia supportate.
- Il server risponde con la suite di crittografia più sicura supportata sia dal client che dal server. Il server invia anche il suo certificato pubblico.
- Il client quindi verifica che la firma provenga da un'autorità di certificazione attendibile. A seconda dell'algoritmo di autenticazione, il server crea un messaggio con la sua chiave privata. Se il client decifra il messaggio con la chiave pubblica del server e il messaggio è corretto, il client sa che il server è autorizzato.

- Il client tenterà quindi di negoziare una chiave simmetrica utilizzando l'algoritmo di scambio di chiavi (questo risulterà in una comunicazione avanti e indietro tra il client e il server. Il numero esatto di messaggi dipende dall'algoritmo).
- Una volta che la chiave è stata negoziata, entrambe le parti sono ora in grado di iniziare a comunicare in modo sicuro.
- Il client invia la sua prima richiesta crittografata con il proprio Bulk Encryption Algorithm. Il server è in grado di decifrare la richiesta con la chiave negoziata (stessa chiave del client). Il server verificherà anche l'integrità dei dati utilizzando il Message Authentication Code Algorithm.
- Il server invia quindi la sua risposta crittografata al client. Il client quindi decifra il messaggio e ne verifica l'integrità.
- Questo scambio di messaggi continuerà finché la sessione è attiva. Una volta terminata la sessione, sarà necessario scambiare nuove chiavi di sessione per avviare un nuovo ciclo di comunicazione sicura.

Chiavi

Crittografia “simmetrica”, ovvero tramite l’utilizzo di una sola chiave sia per cifrare che per decifrare. È un tipo di crittografia meno sicura poiché la chiave deve essere mantenuta segreta ma richiede meno sforzo computazionale.

Il messaggio di un client C viene crittografato prima dell’invio utilizzando la chiave ed il server riesce a decodificare il messaggio utilizzando la stessa chiave.

Crittografia “asimmetrica”, ovvero tramite l’utilizzo di due chiavi distinte: chiave pubblica (che viene distribuita pubblicamente) e chiave privata. Si tratta di un meccanismo più sicuro, ma che richiede più calcoli e funzioni computazionalmente onerose.

Il messaggio di un client C viene crittografato prima dell’invio utilizzando la chiave pubblica del server S.

Il server S è l’unico che può decodificare il messaggio perché per poterlo fare è necessaria la chiave privata.

Solo la crittografia non garantisce la non intercettazione dei messaggi poiché un potenziale intercettatore potrebbe “fingersi” il server, sostituendo la chiave pubblica inviata al client senza che questo se ne possa accorgere (e successivamente decodificare il messaggio).

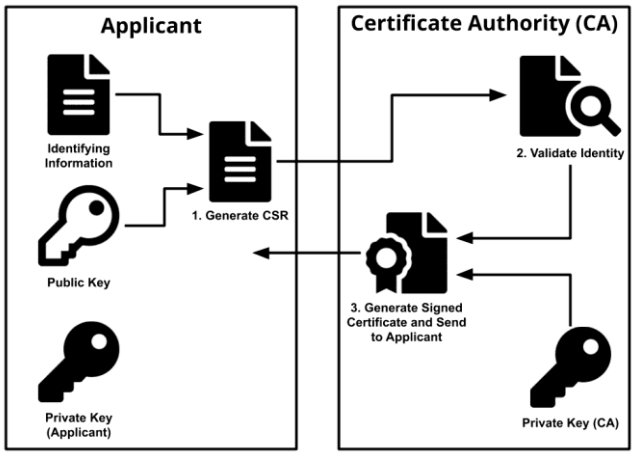
Certificati

Un sistema completo è costituito dai certificati. Essi permettono di garantire:

- **Autenticazione**, fungendo da credenziale per convalidare l'identità dell'entità a cui è stata emessa (un intercettatore non può essere in grado di dimostrare la finta identità);
- **Crittografia,** per comunicazioni sicure su reti non sicure come Internet, evitando le trasmissioni in chiaro;

- **Integrità** di documenti firmati con il certificato in modo che non possano essere alterati da terzi in transito (tramite controllo di una sorta di checksum).

Una Certification Authority (CA) è una società/organizzazione che emette i certificati e che ne può garantire in qualunque momento la validazione.



Una entità "Applicant" (ad esempio un server) può richiedere un certificato firmando tramite la propria chiave pubblica un apposito file che contiene le informazioni che lo identificano (come il nome dell'organizzazione, e l'indirizzo a cui è raggiungibile).

Il file così firmato (CSR) viene inviato alla CA.

La CA, previa validazione dell'identità, provvede a generare un certificato firmato tramite la propria chiave privata da inviare all'applicant.

Quando il server propone questo certificato ad un soggetto terzo (un client), questo soggetto può verificare la firma digitale della CA tramite la chiave pubblica della CA stessa. In questo modo il client ha conferma del fatto che il server abbia presentato un certificato (documento di identità) valido.

Con questo meccanismo di scambio, è solamente il server ad essersi autenticato presso il client (ovvero il client sa con esattezza chi è il server ma il server non sa chi è il client).

Qualora sia necessario che anche il client si autentichi, anche il client deve presentare un proprio certificato al server il quale deve provvedere alla validazione tramite CA (stesso meccanismo descritto in precedenza).

La comunicazione può a questo punto avvenire utilizzando una chiave simmetrica condivisa concordata tra client e server per la durata della comunicazione (sessione).