

Interactive comment on “Automated observatory in Antarctica: real-time data transfer on constrained networks in practice” by Stephan Bracke et al.

Stephan Bracke et al.

stephan.bracke@meteo.be

Received and published: 10 May 2017

Thanks for all comments,

General comments on 56 kbits/sec : You are probably right on the fact that I still have a lot of extra bandwidth with my first message structures being 8 bytes and 16 bytes , but in first implementation I pushed it to the limit so I had these small messages. In current versions I have already bigger messages but still small where I pass also temperature info (24 bytes vector and 12 bytes for scalar) Althouhg it still is small extra bandwidth can come in handy when we get bursts of messages whenever connections got stale or lost. General comments on considering to write my own protocol ? I never considered to write it myself, because when I would do it, it would never be something

like a message oriented middleware (MOM) with a broker that comes with complete the decoupling of publisher/subscriber but rather be a request/reply protocol. Also writing the code would mean for me writing it again in for the moment three languages (C#, javascript, python) and maintaining/testing it in these three languages (and maybe more). The code of mqtt publisher/subscriber is simple and easy to integrate in most popular languages. The complexity lies in the broker which is done for you (it comes with a configuration and management effort). the liberty of adding different subscribers to the same topics doing different obs (one for database insertion, one for alerts on different patterns: gps failures, temperature etc) opens up different possibilities without messing up code. The instruments now default always log to file, by enabling mqtt they can send data to topics without change default behaviours.

Allowing loss of network connection for 4 hours is this enough ? As I stated already I indeed log data locally to files , this is a must because in the unmanned observatory it can be that connection of satellite is lost and only can be switched back on in the next summer. When I began to use the library mqtt.js in 2014 this library was doing the retransmit of messages itself even when connection was lost (not in mqtt standard), however doing this it stored the messages in memory while testing two days of connection failure it got an out of memory. I adapted this code by rewriting handlers so I can now limit the messages that it will store (and is configurable in a file for each topic). I looked at it today and see that the config is currently 1 hour for second data and 4 hours for minute data. In the current version of this api they make a storage available and you can use levelDB (key value nosql database) to store the not sent messages which now gives the opportunity to go way beyond several days (also not standardized and not necessary in api). But then again after some reasoning of what exactly should be done to use mqtt probably one day is the maximum (the real-time aspect is already lost we only use it to deliver every message, for me one hour was sufficient) . To recover more then this I fallback to ftp approaches because the data is available at the antarctica site. There is another aspect that also becomes important on this level and it is the scalability of your broker. Having a burst of one week second

[Printer-friendly version](#)[Discussion paper](#)

data needs to scale well and messages need to be consumed quickly. Each topic can be limited to a maximum of queued messages, etc. This all depends on the choice of broker. The broker I use is not the best on scalability but then again I didn't test it for more than four hours. Does MQTT allow for prioritisation? No, MQTT doesn't have prioritisation in its standard, it is available in AMQP (which also has message routing and etc) but this protocol is not lightweight. You could e.g. use another topic for messages that were not sent during a long connection loss while not influencing the real-time topics.

Technical Corrections Most of the technical corrections are included in the new version. Some extra notes: Line 15: Is it well documented? All technical schemes can be found on <https://github.com/beagleboard/BeagleBone-Black> and http://www.elinux.org/Beagleboard:BeagleBoneBlack#Hardware_Files Line 20: What has the software been re-written from? Was there a Windows version of the software that had to be re-written? The rewritten is true for the lemi-025 I rewrote everything that was possible in the windows by studying the user interface and remake it into a web interface (possibility to see current measurements, update dacs, stop/start measurements, change parameters, etc) Isn't MQTT and ISO standard (ISO/IEC 20922:2016)? yes it is, I didn't know it and added it in the paper. Line 24: What does this mean? Does MQTT include metadata that describes the topic? I rewrote this section because it was not clear. MQTT doesn't include metadata it is up to the analyst/developer to document message structures (this can be binary formats (what I use) or text (JSON if we want but messages will be bigger in this case)) there is no WSDL like in soap webservice. Best regards, Stephan

Interactive comment on Geosci. Instrum. Method. Data Syst. Discuss., doi:10.5194/gi-2017-17, 2017.

[Printer-friendly version](#)[Discussion paper](#)