Atmos. Meas. Tech. Discuss., 8, C4982–C4989, 2016
www.atmos-meas-tech-discuss.net/8/C4982/2016/

Atmospheric Measurement Techniques Discussions

Open Access

AMTD

8, C4982–C4989, 2016

Interactive Comment

Interactive comment on "Real time data acquisition of commercial microwave link networks for hydrometeorological applications" *by* C. Chwala et al.

C. Chwala

christian.chwala@kit.edu

Received and published: 19 January 2016

Dear Anonymous Referee 2,

thank you for your comments. Please excuse the late reply, but we wanted to take the opportunity of this response to improve our software with regards to memory consumption to better answer your comments on the scaling for larger networks of CMLs.

We will respond below (your text in black, our text in blue).

GENERAL COMMENTS:

The authors state that their data acquisition system is first of its kind; however similar

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper

system was already used for CML data acquisition by Fencl et al., (2015), although their software was not an open source. The novelty of the manuscript lays therefore rather than in presenting data acquisition system itself, in showing transferability and scalability of such a tool. The reviewer recommends to accept the manuscript after revisions specified bellow.

Thank you for pointing out this issue. We knew from personal communication that Fencl et al. also acquire data via SNMP requests. We were not aware of the fact that their work was published, though.

Based on the information from Fencl et al. (2015) there is however a huge difference between the capabilities of their data acquisition and ours. First, as they write, their software polls data serially. That is, the sampling rate will be heavily limited by the number of CMLs, which may explain why they only poll data for 14 CMLs. Second, they write that the data “is polled approximately five times per minute”. That is, there does not seem to be a fixed sampling rate, which may be attributed to the serial polling method which can cause temporal offsets according to the SNMP request round trip time or to CMLs that do not respond. In summary, from the information given in Fencl et al. (2015) we argue that their method would not be able to continuously query hundreds of CMLs with a reasonable temporal accuracy and resolution.

Nevertheless we have to agree with the reviewer that Fencl et al. (2015) were the first to publish the use of a custom data acquisition software in an operational CML network.

We will clarify this and highlight the unique feature of *pySNMPdaq* in the revised manuscript.

The reviewer recommends especially, to present more in detail the performance of authors’ system in terms of its effect on a CML network (data traffic), and also in terms of a computational demands on a server side. Authors might show the performance statistics from real operation of their system and the expected performance statistics for larger networks or higher temporal resolutions (authors mention they have done such calculations). This would be also in agreement with one of the authors' goals, to encourage operators to open their networks. Data traffic issues and security of a whole system, in terms of not hindering the primary function of a CML network, will be most probably operators' main concern.

We agree that a more detailed explanation of the computational demands and the effect of the data acquisition on the CML network is needed. Please see our responses to your corresponding specific comments.

SPECIFIC COMMENTS

12246, 15: The authors state that CML networks provide good coverage also in inhabited areas and refer to Overeem et al. (2013). Overeem et al. however state that CML coverage coincides (in global scale) with population density. Their country wide CML rainfall maps were reconstructed for Netherlands where there are rarely any larger inhabited areas. Please change the reference or the statement itself.

As far as we know (backed by available free online dictionaries), "inhabited" means that a place has inhabitants, i.e. the place is populated. Hence we state the same as Overeem et al. (2013): CML coverage is good in populated, that is inhabited, areas.

12247, 24: Isn't one of the limitations of the approach also variations in TX-level due to hardware issues (although it is assumed to be constant)?

We agree that the TX-level may be fluctuating due to hardware issues. But this problem is not unique to our old setup with the data loggers and the older generation CMLs with constant TX-level that we used. It is more of a general problem of CMLs, since they were not designed to be measurement devices. The recordings we get via SNMP, which may also be fixed constant TX-levels if ATPC (automatic transmit power control) is switched off, do not necessarily have to be 100

12249,19 and 12251, 1-8): To reviewers understanding the simultaneous SNMP requests are enabled by defining data acquisition process for each single CML. Does

AMTD
8, C4982–C4989, 2016

Interactive Comment

the approach, when each single CML is handled by its own process, restrict number of CMLs, which can be queried simultaneously? Country-wide CML networks might have one or two orders of magnitude more CMLs than amount, on which the application was tested (450). Could you comment on i) an ability of the *pySNMPdaq* to query thousands of CMLs without time synchronization problems ii) if CML networks can smoothly handle data traffic when thousands of SNMP requests are sent simultaneously?

The referee is correct, there is an individual process on the level of the operating system for each CML. And each new process comes with a certain cost, mostly memory consumption. In particular interprocess communication via Queue and Events consumes memory. That is, the limit of number of parallel SNMP requests is limited by the available RAM. We have recently updated our software in this regard to decrease the memory footprint (see attached figure). The update will we published on github in the next days, the latest when we submit the revised manuscript. With the updated version, 1000 parallel processes will require approximately 10 GB of free memory. This seems to scale linearly, but we did not have a computer with more than 8 GB RAM to test. We will look into further improving this scaling and elaborate on it in the revised manuscript.

Response concerning point i) Time synchronisation is not a problem. Our `smart_sleep()` function can trigger the SNMP requests with one second resolution with a reproducible accuracy of some milliseconds. The delay between requesting via SNMP and receiving the results will of course vary. Typical round-trip times (RTT) we have experienced are between 0.1 and 0.4 seconds. *pySNMPdaq* records the RTTs. That is, there is no software problem in sending out thousands of SNMP request at the exactly same second every minute. There may be constraints from the network topology, though (see our response to your comment to 12254, 9-13)

Response concerning point ii) We do not know how the CML network would react on sending 10.000 simultaneous request. For this order of magnitude, our plan is to do staggered simultaneous requests, each for approximately 1000 CMLs. With a typical SNMP request round-trip time of below 0.5 seconds, we could do the staggering with 1 second offsets. That is, for every minute there would be 60 sets of 1000 CMLs that would be requested simultaneously. The order would always be the same. Hence the sampling frequency for each individual CML would still be one minute. This way, we could realize 60.000 CML requests with minutely resolution. Please note that this is a future plan and we will not do this in one step.

The option for staggered request will be added to *pySNMPdaq* later this year since we plan to extend the acquisition to more than 1000 CMLs in the next step. We will elaborate on this issue in the revised manuscript when discussing the scaling of *pySNMPdaq*.

12252, 6-8: What kind of interference of pySNMPdaq with CML's normal operation can occur? Is there direct risk endangering flow of telecommunication data, or rather the risk of hindering smooth CML network management? To reviewer's knowledge there is a special reserved service channel for CML network management. Does pySNMPdaq use this channel to acquire TX/RX-level data?

We do not know if there is an industry standard, but from our information from Ericsson, CMLs have a SNMP service channel which is separated from the normal user communication channels and has a bandwidth from 64 kbps to 192 kbps (depending on the CML model). Hence, there is no risk to endanger the flow of user communication in a CML network. However, the network management via SNMP, could be harmed if the network is overloaded with SNMP requests. Our software does not specify how the SNMP requests are actually carried out. But from our exchange with Ericsson engineers we assume they propagate via the SNMP service channel.

12252, 21-12253, 11 and 12253, 22-27: Although presented data examples are interesting, they do not have straight connection to the data application system itself. The reviewer suggests to link these examples to the system e.g. by commenting on the gain

AMTD

8, C4982–C4989, 2016

of obtaining CML data with a 1 s and 1 min resolution in contrast to the computational or data traffic expense of such resolutions.

We do not understand how the data examples “do not have straight connection to the data application system itself”. We cannot show the data acquisition system in action, hence we show examples of resulting data. Furthermore we show data from two different CML types with two different temporal resolutions to highlight the flexibility. We do not want to judge the usefulness or necessity of the individual temporal resolutions. It will be up to the prospective user, bound by the limitations set by the CML providers, to select a temporal resolution. We only want to show what is possible.

12253, 9-11: Can the channels of single CML interfere with each other and cause these small variations?

All electronic systems are subject to interference due to electromagnetic radiation. The question is the order of magnitude of these interferences. Since we do not know the inner workings of the CMLs we cannot quantify them. From our experience with RF-systems, we however can assume that the CML receivers have an appropriate band-pass filter system. That is, the two directions of a CML, which typically use frequencies with approximately 1 GHz difference, should not interfere too much. The protection systems, on the other hand, are only redundant receivers and hence cannot cause interference.

12254, 9-13: There are additional calculations of data traffic mentioned in this paragraph, without further specification. The traffic issue is, however, main constraint possibly limiting number of CMLs, which can be polled from one server. Additional data traffic might impair CML network operation. Thus, traffic issues will be for cellular operators (and other CML owners) of the main interests when opening their network for such data acquisition system. The reviewer therefore suggests to specify data traffic calculations more in detail, show specific results and comment on them (e.g. in section 6).

As mentioned above, the SNMP service channel of CMLs is between 64 kbps and 192 kbps wide. A request of four values (TX- and RX-levels for the near end and the far end) via SNMP causes a data traffic of approximately 500 bytes, 250 bytes for the request, 250 bytes for the response, mostly caused by the package overhead. That is, even the slowest SNMP service channel with 64 kbps, equal to 8 kilobytes per second, could theoretically handle 16 of those requests every second. How this limits the data acquisition for the whole network, will depend on the network structure. We can only speak for Ericsson in Germany, but typically the cell phone backhaul network consists not only of CMLs. Typically there are small hubs or islands of CMLs which are connected via DSL to the IT center. That is, there is no big accumulation of SNMP request to one or very few CMLs. Only if the network would be a perfect tree structure of CMLs, one would definitely cause overload in the "trunk" (the CMLs closest to the IT center). We however doubt that any CML network is organized in a completely tree-like structure because this would also accumulate a lot of the data traffic towards the "trunk". Following the reviewer's recommendation we will discuss the details of the data traffic briefly in section 6.

12254, 17-18: Special designed application for weather data acquisition was also used by Fencl et al., (2015). Their application was used in operational conditions and provided data with temporal resolution approx. 12 s (Fencl et al., 2015).

Please see our corresponding response to your first general comment.

Interactive comment on Atmos. Meas. Tech. Discuss., 8, 12243, 2015.

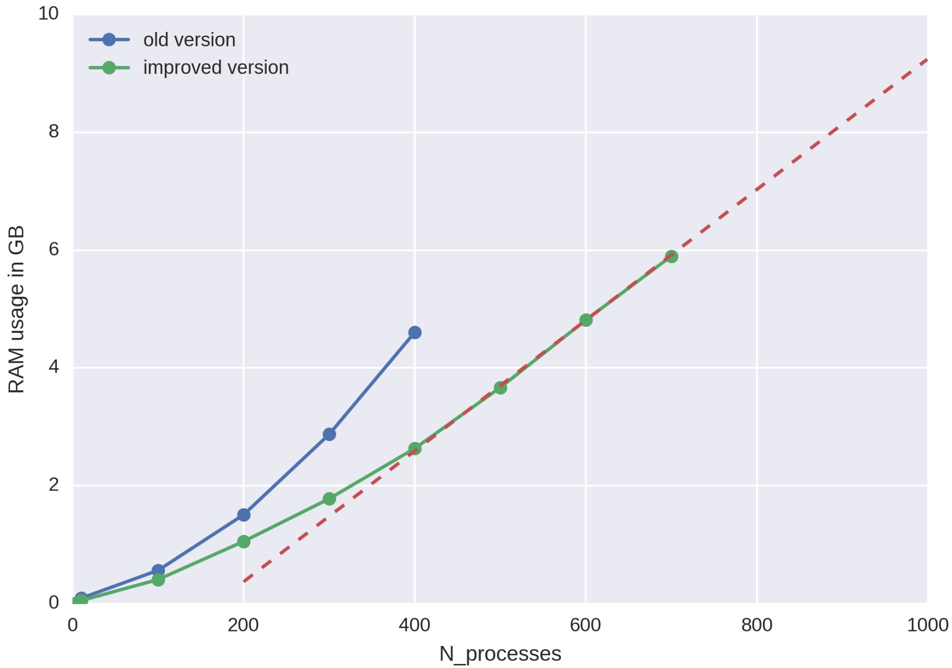

Fig. 1.

AMTD

8, C4982–C4989, 2016

Interactive Comment

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper