Review Comments on:
ALERT2 Application Layer Protocol Specification
Timothy J. Salo[1,2]
January 18, 2011

1. Introduction

This document contains the author's review comments for the *ALERT2 Application Layer Protocol Specification*, Version 1.0, dated September 1, 2010, published by the National Hydrologic Warning Council (NHWC). The primary intent of these comments is twofold: to improve the quality of the specification and to highlight specific technical issues that may warrant additional consideration.

This document is organized as follows:

- "General Comments" contains just that: comments that aren't clearly associated with specific sections of the specification.

- "Technical Comments" focuses on the technical aspects of the specification, including instances where the specification appears to be unclear or incomplete, and design decisions that may warrant reexamination.

- "Editorial Comments" includes comments that may help improve the readability and understandability of the specification.

- "Conclusions and Recommendations" contains general conclusions and recommendations about the specification.

These review comments focus on several aspects of the proposed protocol and its specification, including:

- **Interoperability** Is the specification complete, clear, and unambiguous enough to ensure that all implementations that conform to the specification are assured of interoperating with each other? Does the specification provide enough detail that a reasonably experienced engineer can reliably implement the specified protocol, without the need for any additional information about the protocol, beyond that what is contained in the specification?

- **Technical Correctness** Is the protocol, as specified, likely to achieve its apparent objectives?

[1] This work was supported by National Oceanic and Atmospheric Administration (NOAA) National Weather Service (NWS) Office of Climate, Water and Weather Services under Order Number DG133W10SE3038.

[2] The opinions expressed here are those of the author, and do not necessarily reflect those of any other individual or organization, including those that have funded, are funding, or may in the future fund the author's employer, Salo IT Solutions, Inc.

- **Functionality** What functionality is missing from the protocol? Is any functionality included in the protocol that ought to instead be implemented in some other protocol layer or device?

- **Extensibility** Does the protocol described in the specification permit new features and functionality to be easily added? In particular, can new functionality be easily added to the protocol without requiring that the installed base be upgraded prior to using this new functionality? What changes to the protocol ought to be considered in order to simplify the future evolution of the protocol?

- **Usability** Does the protocol described in the specification reasonably minimize the amount and complexity of the configuration that is required by system administrators? Does the protocol prevent or detect common misconfigurations?

- **Clarity** Is the specification clear and unambiguous? Is the specification easy to understand?

The protocol described in this specification represents a tremendous advance over the original ALERT protocol and addresses a number of the fundamental limitations of that protocol, including:

- Limited sensor value range
- Small, fixed message format
- Limited protocol extensibility
- Monolithic protocol
- Missing sensor data descriptions

On the whole, this specification appears to be reasonably complete, although the issues raised by these comments may lead to improvements in the protocol and in the specification.

2. General Comments

Several general issues warrant mention.

2.1. Technical Maturity

Collectively, the three available specifications (ALERT2 AirLink Protocol Specification, ALERT2 MANT Layer Protocol Specification, and ALERT2 Application Layer Protocol Specification) leave the impression that they *may* be snapshots of an evolving system design, rather than a consistent set of specifications. For example, this specification states that the address of the source node is contained in the *application-layer* protocol, and the MANT specification states that the modem inserts the source address in the *link-layer* header (which presumably the application-layer protocol could use), but the AirLink specification omits any mention of this requirement.

2.2. ALERT2 System Overview Description

The addition of an overview of ALERT2 systems would help the reader to understand how the various ALERT2 protocol specifications relate to each other. This overview could be included in each ALERT2 protocol specification, or could be a stand-alone document.

2.3. ALERT2 System Architecture Specification

It may be beneficial to create an ALERT2 system architecture specification document. This document could specify the technical framework within which the individual ALERT2 protocol specifications fit. The system architecture specification could also provide material that is common to or applicable to the other ALERT2 protocol specifications. Perhaps, the ALERT2 system overview could be contained in the system architecture specification document. See my comments on this topic in my review of the MANT protocol for additional information.

2.4. Protocol Evolution and System Migration

ALERT2 system administers would benefit if future versions of the ALERT2 protocol suit did not require that all nodes in an existing ALERT2 system be upgraded to the new, enhanced protocol simultaneously. The protocols in this initial release of the ALERT2 protocols ought to include mechanisms that permit the enhanced protocols and the unenhanced protocols to be used simultaneously in a single network. Specifically, the ALERT2 protocols should contain mechanisms that permit nodes that implement unenhanced protocols to ignore the portions of the new protocols that provide new functionality. That is, ALERT2 V1.0 nodes should be able to receive and process ALERT2 V2.0 messages, but should ignore the portions of the ALERT2 V2.0 messages that implement the new functionality.

The length field described in section 2.1.3 of this document and the sensor ID/length field described in section 3.1.3 may conflict with this objective, as detailed below.

2.5. Specification Completeness

The ALERT2 time-division, multiple-access (TDMA) medium access control (MAC) protocol does not appear to be documented in any of the available ALERT2 protocol specifications. The ALERT2 TDMA protocol needs to be documented somewhere, probably in a stand-alone specification.

2.6. Protocol Usability

The ALERT2 Protocols Technical Working Group might consider specifying additional details of the ALERT2 application-layer protocol. This may reduce the need for ALERT2 administrators to manually configure their systems and may help avoid some possible misconfigurations. As described in the detailed comments below, this specification might require that:

- Timestamps must use UTC.

- Some sensor ID values are specified in this specification, some values are reserved for future use in this specification, some values are reserved for use by vendors, and some values are reserved for use by customers.

- All stream stage data use particular units (inches, feet, or meters above a reference point that is unique to the gauge, pick one) and have the same precision (e.g., hundredths or a foot or meter).

- Temperatures must represent either Fahrenheit or Celsius (pick one or the other, but require that all ALERT2 systems use the same scale). Expressing temperatures in tenths of degrees may eliminate concern about accuracy that may be lost when temperatures are converted from Fahrenheit to Celsius or conversely.

These sorts of requirements will ensure that, for example, a remote station is not configured to use one scale, while an application assumes that the other scale is used.

3. Technical Comments

These comments suggest areas where the technical content could be made more complete or more clear. In a few instances, these comments suggest specific technical aspects of the proposed technology that may warrant additional examination.

3.1. ALERT2 Addressing Architecture

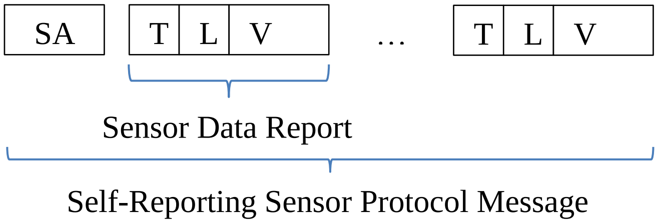
The inclusion in the ALERT2 application layer message of the address of the node that originated a packet (the SA field) suggests that the ALERT2 protocol suite may benefit from additional consideration about how addressing and routing will work. See my review comments on the MANT protocol for an extended discussion of this topic.

3.2. Single or Multiple ALERT2 Application Protocols

This specification appears to be ambiguous, or even contradictory, about how many application layer protocols are being specified. The second paragraph on page seven states: “The Self Reporting Sensor Protocol comprises a set of application layer ALERT2 message types.” It appears to me that, at least functionally, the document specifies a single application-layer protocol that transports data from a variety of sensors. It is not clear to me whether my confusion about the number of protocols described in the specification results from the protocol being described, results from the description of the protocol, or perhaps reflects evolving thoughts about the design of the protocol that are reflected in the specification.

I strongly recommend that the specification explicitly describe a single ALERT2 application-layer protocol that is responsible for transporting sensor data from a wide variety of sensors. I suggest that this single application-layer protocol for sensor data be referred to as the “Self Reporting Sensor Protocol” or “SRS protocol”. This [single] application protocol should be identified by a single application layer protocol id (ALPID) in the ALERT2 link-layer protocol header (part of the AirLink protocol). Additionally, in this document I refer to the type-length-value structures that generally contain sensor data as “sensor data reports”, in order to avoid using the heavily overloaded term “message”.

I suggest that the diagram in section 2.1 of the specification be enhanced as shown below:



3.3. Time

I recommend that the specification state that the values represented by timestamps must be in UTC. Explicitly stating that all timestamps are in UTC will avoid the risk that some ALERT2 implementations will use some other time, or that some parts of an ALERT2 system will assume UTC while other parts assume some other time representation.

3.4. ALERT2 Timestamps

In my view, the approach to timestamping sensor data described in the ALERT2 protocol specifications may be more complex and less accurate than necessary. For an extended discussion on this topic, see my review comments on the MANT protocol.

3.5. Simplified ALERT2 Multiplexing Protocol

It appears to me that it may be possible to extend the Self Reporting Sensor Protocol to support multiplexing. This *might* offer a much simpler solution than the MANT protocol. See my MANT protocol review comments for an outline of these extensions to the Self Reporting Sensor Protocol.

3.6. Detailed Technical Comments

Page 6, paragraph 1. Please provide a reference for the document on which this specification is based. Also, please make the document available on the NHWC Web site and provide a permanent URL for the document.

Page 7, paragraph 1. There are several ALERT2 protocols, not one. Language such as “ALERT2 protocol suite” or “suite of ALERT2 protocols” would make this clearer.

Page 7. A description of how the protocols described in this specification relates to the other ALERT2 protocols would be useful. One or more diagrams would help the reader understand how the ALERT2 protocols are related to each other.

Page 8, paragraph 2. “… uniquely identifies the transmitting site over the domain of operation of the subject system”. It might be worth noting that anomalous propagation conditions can occur at the VHF frequencies that are commonly used for ALERT2 systems, and that this should be taken into consideration when coordinating address assignments. Refer to my discussion of ALERT2 addressing in my MANT protocol review comments.

Page 8, section 2.1.3. “the size of the length field could be different for other message types.” I think that this presents some serious, potential problems. It requires that the receiver understand how long the length field is for every type code that a node might receive. In particular, it requires that, if a type code with a two-byte length field is specified, then *every* potential receiver must be upgraded to understand the new type code *before* it is used in a network. This defeats one of the major benefits of the TLV structure, namely that new type codes can be defined and used, even if some receivers don’t understand the internal details of the type code. If the length is always one byte, then every receiver can simply skip any TLV with a type code that it doesn't understand (because every receiver can correctly parse the length field). Conversely, if the length field is expanded for some type codes in the future, then existing implementations will be unable to parse these new TLV structures and skip the ones that they don't understand.

An alternative approach to providing a variable-length length field is to use a flag to indicate whether the length field is longer than one byte. For example, the length field might be seven-bits long, with the eighth bit indicating whether a second byte of length field is present. This permits new type codes to use longer length fields, without adversely affecting the installed base, assuming that the installed base understands how to parse two-byte length fields. (This mechanism could be recursive, permitting an essentially unlimited length field, although this is probably overkill.)

Page 8, paragraph 5. The specification should state whether the Message Value field must be present (i.e., have a non-zero length). This will inform implementers as to whether they need to check for zero-length Message Value fields.

Page 9, last paragraph. "The value of 0 is reserved for extensibility." I suggest that this specification explain *how* a value of zero will be used to enable extensibility. In particular, if any behavior is required of existing implementations when a value of zero is encountered, it should be documented so that all implementations respond appropriately when a zero value is used in the future. Again, implementations of version 1.0 of the ALERT2 application layer protocol should be able to understand this extension mechanism, so that they can skip TLVs that contain it, even if they don't understand how to process it.

Page 10, paragraph 1. "… but the ALERT2 protocol does not associate ID numbers with given types of sensors." I recommend that this specification associate ID numbers with the most common types of sensors. This will help reduce the manual configuration required by ALERT2 system administrators.

Page 10, paragraph 2. "0 Reserved for extensibility". Again, I suggest that this specification define how this value will be used to provide extensibility. What should an existing implementation do if it encounters a zero? Discard the whole sensor report? Unless existing implementations understand the extension mechanism, they might not be able to correctly skip the current SL-V structure and find the next one.

Page 10, paragraph 3. "0 Reserved for extensibility". See above comment.

Page 10, paragraph 3. "4 4-Byte unsigned integer indicating epoch time or seconds since 1970 not including leap seconds". I believe that this language, while incomplete, intends to say that the representation specified in the 2001 version of the POSIX 1 specification is to be used. As discussed above, I recommend that the specification state that the time *must* be in UTC.

Page 11, paragraph 3. "The values have the meaning "seconds before this transmission.". It might be beneficial to permit a timestamp to be included in the Tipping Bucket Rain Gage Report. Refer to my MANT protocol review comments for a discussion of timestamps. I also suggest specifying a 16-bit timestamp that represents the number of seconds since 12:00 a.m. or 12:00 p.m. UTC, whichever is more recent. This smaller-sized timestamp reduces the cost of using timestamps, rather than the times at which packets were received, may simplify the processing of timestamps, and may improve the accuracy of timestamps.

Page 12, paragraph 3. The Technical Working Group might consider specifying the value at which the accumulator rolls over. This would reduce the configuration required of ALERT2 systems and would reduce the opportunities for misconfiguration. (Since new software must be written to support the ALERT2 application layer protocol, this new software could be responsible for expanding the range of the accumulator, even if the rain gauge itself uses a more limited range.)

Page 12. The Technical Working Group might consider specifying a node status or node configuration message. Ideally, this message might contain every parameter in node's configuration. This could be used, for example, to inform application software about the configuration of a node, thereby reducing the need to manually configure the application software. Of course, it would also be nice to be able to have a similar message that *sets* parameter values in a remote node, thereby permitting nodes to be configured or reconfigured remotely.

Page 13, paragraph 4. The Technical Working Group might consider that stream gage data be specified in hundredths of a foot (or some other standard units). This will minimize the need for system configuration and the opportunities for system misconfiguration.

Page 13, paragraph 5. The Technical Working Group might consider requiring that all temperature data be specified in either Fahrenheit or Celsius.

Page 17, paragraph 2. I suggest considering making these sensor IDs standard. Perhaps, something of the form: "ALERT2 implementations must use the following IDs by default, and should permit system administrators to redefine these IDs".

Page 17, paragraph 2. It might be useful to explicitly reserve a range of IDs for use by system administrators. This will assure system administrators that future versions of the ALERT2 application protocol will not use IDs in this range, and thereby force them to reconfigure their networks. Likewise, a range of IDs might be reserved for use by vendors.

4. Editorial Comments

These comments are principally editorial.

4.1. Terminology

In this specification, “ALERT2” has a trademark symbol (™) appended. However, a search of the U.S. Patent and Trademark Office database did not return any information for “ALERT2”. Is “ALERT2” actually trademarked? Has a trademark application even been submitted?

4.2. Detailed Editorial Comments

- Page 5. Please fix “heading does not appear in the table of contents”.

- Page 6, Paragraph 1. Please delete “This heading appears in the table of contents but has no section number assigned to it.”

- Page 6, Paragraph 2. Please use consistent capitalization for “ALERT2 Protocol Technical Working Group.

- Page 6, Paragraph 2. This list of ALERT2 protocol technical working group differs from the list on page 3. Is that correct?

- Page 8, heading 2. I suggest that this heading be changed to “ALERT2 Self Reporting Sensor Protocol”.

- Page 8, heading 2. This specification might be easier to read if it stated that it describes only one ALERT2 application-layer protocol, the Self Reporting Sensor Protocol, rather than several, as the heading implies.

- Page 8, paragraph 1. Be more definitive than “We anticipate that…”. This is a specification, not a proposal.

- Page 8, paragraph 1. “ALERT2 payload” seems like an imprecise term. I suggest that the fields described in the diagram be called something like an “ALERT2 application-layer message” and that the individual TLV records be called something like “sensor reports”.

- Page 8, paragraph 1. “The Self Reporting Sensor Protocol comprises the set of application layer ALERT2 message types”. I recommend that the term “Self Reporting Sensor Protocol” be used consistently to refer to the application layer protocol described in this specification.

- Page 8, paragraph 1. I recommend that the General Sensor Report and Tipping Bucket Rain Gage Report *not* be referred to as “messages”. They might be referred to as “sensor reports” or something similar. In general usage, “message” refers an application-layer protocol data unit (PDU), which in this case is a source address field followed by one or more sensor reports.

- Page 8, paragraph 1. Removing the final sentence of this paragraph might make the document easier to update as the protocol is enhanced and new sensor report types are added. Or, perhaps a bulleted list or a table could be used.

- Page 8, heading 2.1. The fact that the Self Reporting Sensor Protocol is identified by an ALPID value of 001 doesn't belong in the header.

- Page 8, paragraph 3. The list of sensor report type codes would be better presented in a table.

- Page 9, heading 3. The General Sensor Report is specified in section 3, but the Tipping Bucket Rain Gage Report is specified in section 3.2. I recommend that both reports be described at the same outline level (e.g., 3.1 and 3.2).

- Page 9, paragraph 2. I recommend that "Message Value (V)" be called something else, since that term has already been used in this document for another purpose. Perhaps, this could be called "Sensor Value (SV)" or "Sensor Data (SD)".

- Page 9, heading 3.1.1. "t" -> "T".

- Page 10, paragraph 2. Suggest "8 digital sensors" -> "8 binary sensors".

- Page 10, heading 3.1.4. See previous comments about "Message Value (V).

- Page 11, paragraph 1. Suggest "tips" -> "bucket tips".

- Page 11, paragraph 1. Suggest "limits channel utilization" -> "reduces channel utilization".

- Page 11, paragraph 2. Define "RTU".

- Page 11, paragraph 2. Define "TDMA".

- Page 11, paragraph 2. Capitalize "ALOHA" consistently. I believe that the original papers used all capitals.

- Page 13, paragraph 2. Suggest "licenses" -> administers.

- Page 13, last paragraph. Most readers will be much happier if this information is presented as a diagram that labels the fields (e.g., a diagram that indicates that the first two bytes are the SA field, etc.).

- Page 14, paragraph 2. Suggest "licenses" -> administers.

- Page 14, last paragraph. Most readers will be much happier if this information is presented as a diagram that labels the fields (e.g., a diagram that indicates that the first two bytes are the SA field, etc.).

- Page 15, paragraph 2. Suggest "licenses" -> administers.

- Page 15, last paragraph. Most readers will be much happier if this information is presented as a diagram that labels the fields (e.g., a diagram that indicates that the first two bytes are the SA field, etc.).

- Page 16, table 1. The third column appears to be truncated by almost one digit on the right.

- Page 18. Please add references, including the other ALERT2 protocol specifications and the document on which this specification is based.

5. Conclusions and Recommendations

In my view, the protocol described in this specification represents a tremendous advance over the original ALERT protocol and addresses a number of the fundamental limitations of that protocol.

I recommend that the ALERT2 Protocol Technical Working Group consider the review comments contained in this document and update the ALERT2 Application Layer Protocol Specification document as it believes is appropriate.

In my view, this draft of the ALERT2 Application Layer Protocol Specification appears to be nearly complete enough for someone to implement the protocol without relying on other sources of information. Of course, the best test of the clarity, precision, and completeness of a protocol specification is for an independent group to implement the protocol based only on the information contained in the specification.

In addition to the technical commends included above, I recommend that several areas receive additional attention, including:

- Additional effort should be made to minimize the manual configuration required by ALERT2 administrators;

- Consideration be given to adding a 16-bit timestamp and reworking timestamp processing, as described in my MANT protocol review comments; and

- Consideration be given to replacing the multiplexing functionality of the MANT protocol with the enhanced Self Reporting Sensor Protocol that is outlined in my MANT protocol review comments.

I recommend that the ALERT2 specifications be adopted by the NHWC as full standards only *after* an independent implementation that was developed using only the information contained in these specifications has been shown to interoperate with the existing ALERT2 prototypes and products. This is perhaps the only way to ensure that the standards adopted by the NHWC are accurate enough and sufficiently detailed enough to permit independent implementations of these protocols to interoperate with each other. In the interim, the NHWC might adopt this specification as a “draft standard” or assign it a similar status that denotes that the NHWC believes that the specification is complete, but that the NHWC is awaiting feedback about implementation experiences before promoting it to full standard status.