Review Comments on:
ALERT2 MANT Layer Protocol Specification
Timothy J. Salo[1,2]
Version 1.0 – January 18, 2011

1. Introduction

This document contains the author's review comments for the *ALERT2 MANT 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?

I believe that this specification would benefit from a substantial reorganization, additional editing, and the inclusion of significant additional detail.

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, the application-layer protocol 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.

Because of their importance, a description of how addressing and routing work in the initial version of the ALERT2 protocol suite should be included in the system architecture specification. Additionally, the system architecture specification ought to describe how the addressing and routing functionality contained in the initial release of the ALERT2 protocol suite will be enhanced to support new functionality, such as two-way communications. Possible future functionality that ought to be included in this architecture specification includes: two-way protocols, dynamic (rather than manually configured) routing, and zero-configuration ALERT2 networks (ALERT2 networks that configure themselves to minimize the amount of manual configuration required). Specifying these future capabilities now may help minimize the changes that will be required in the future implementations of these initial versions of the ALERT2 protocols in order to support future versions of the ALERT2 protocols.

2.4. Protocol Evolution and System Migration

ALERT2 system administrators would benefit if future versions of the ALERT2 protocol suite 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.

2.5. Specification Completeness

The ALERT2 time-division, multiple-access (TDMA) media 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.

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. Assignment of Functionality to Subsystems

This specification appears to assign functionality to the AirLink modem beyond that which is typically implemented in modems. For example, the AirLink modem appears to be responsible for knowing a node's node address and inserting it in the appropriate link-layer header. I recommend that these decisions be carefully reexamined to ensure that they minimize the following risks.

- **Increased configuration complexity** It would be highly desirable if the AirLink modem did not increase the complexity of the configuration of ALERT2 nodes. While it may not be possible to eliminate the need to configure ALERT2 hosts (the processor or intelligent portion of an ALERT2 node that uses the AirLink modem), it is probably possible to eliminate the need for a separate configuration process for the AirLink modem. One approach to achieving this objective is to 1) minimize the functionality implemented the AirLink modem to the physical layer and a portion of the link layer, and 2) pass all configuration information from the ALERT2 host to the AirLink modem. For example, rather than configuring an AirLink modem with a node's node address, the host could pass its node address to the AirLink modem when a frame is transmitted. A mechanism for passing node addresses to the AirLink modem will be needed if two-way communication is implemented in ALERT2, so implementing a capability to pass addresses now, may simplify the future implementation of two-way protocols. One benefit of a zero-configuration AirLink modem is that a failing AirLink modem could swapped out for a replacement unit without the need for any reconfiguration (because all of configuration information is contained in the host, and no configuration information is contained in the modem). This benefit will be even greater if it avoids the need to program AirLink modems in the field, or if field programming of AirLink modems is unwieldy.

- **Increased Complexity of Enhancing ALERT2 Protocols** To the extent that AirLink modems are responsible for implementing network-layer functionality such as routing, then enhancing the ALERT2 protocols becomes more difficult when the AirLink modems need to be upgraded to implement this functionality. For example, if traditional modems are used, then upgrading the ALERT2 protocols to support two-way communications would not require modifications to the modems.

- **Increased Complexity of Supporting Non-AirLink Modems** Implementing non-traditional, ALERT2-specific functionality in AirLink modems will increase the complexity of using the ALERT2 protocols with non-AirLink modems. That is, in order for an ALERT2 host to use a non-AirLink modem, the host must implement the unique functionality that is implemented in AirLink modems (e.g., the host must insert the node address in frames, because the non-AirLink modem doesn't). Conversely, if ALERT2 AirLink modems, similar to most other modems, generally appear to be devices that simply transmit frames, then it would be much easier for ALERT2 nodes to support other modems. For example, it would be nice if ALERT2 systems could use commercial, off-the-shelf (COTS) 6.25 kHz modems without major changes to ALERT2 hosts.

It is also possible that the term "modem", as used in these specifications, is ambiguous. It may, at times, describe an intelligent network device that implements a fairly complete ALERT2 protocol stack, in addition to the functionality traditionally found in modems. Again, an ALERT2 systems architecture specification document, which describes how functionality is assigned to subsystems, would help clarify this issue.

3.2. 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. These comments suggest some topics that warrant additional examination.

Each ALERT2 node will be configured with a 16-bit address, which is referred to here as a "node address". A node address identifies a specific node within a network; other mechanisms are used to identify functionality or subsystems within a node. I suggest that ALERT2 and ALERT addresses share the same 16-bit address space. If this is the case, then a 16-bit ALERT2/ALERT address uniquely identifies a node, and doesn't need to be tagged as either an ALERT2 address or an ALERT address. This notion needs to be expanded to account for the smaller size of ALERT addresses, but I think the general principle is evident.

(A term such as "node address" or "node ID" should be used to identify the 16-bit address that is assigned to an ALERT2 node. This will avoid confusion with the "source address field", which may contain a node address. At some point, there may also be a "destination address field", which may also contain a node address. Using the term "node address" may prevent unwieldy phrases of the form: "the destination address field contains the source address of the node that is the destination of the packet" or "the source address of the destination node".)

Unlike some network architectures, a single 16-bit node address will be used by all of the ALERT2 protocol layers, including:

- the ALERT2 link-layer protocol (which is part of the AirLink protocol, and is responsible for forwarding frames from one node to an adjacent node),
- the ALERT2 network-layer protocol (which doesn't appear to explicitly exist, but is the functionality of ALERT2 MANT protocol that is responsible for forwarding a packet through nodes until it reaches its final destination), and
- the ALERT2 application-layer protocol.

I strongly recommend that a source address (SA) field be added to the ALERT2 link-layer protocol (the AirLink protocol) and that the SA field be present in *all* AirLink frames. In my view, *every* link-layer frame transmitted in an ALERT2 network should contain its source address, the node address of the node that originated the packet. Transmitting any packet without a source address is likely, sooner or later, to make diagnosing anomalous network behaviors extremely difficulty. The ALERT2 protocols appear to be evolving in this direction, but I can't tell if a source address field is required in every link-layer header, because this isn't mentioned in either the AirLink link-layer protocol specification or the ALERT2 application-layer protocol specification.

I recommend that ALERT2 application-layer protocols use the source address contained in the ALERT2 link-layer or network-layer header, wherever possible. (Note that the simplified application-layer multiplexing protocol suggested below *does* include node addresses in some application-layer packets. But, these are cases where the correct address can't be found in the lower-layer headers.)

I also recommend that the source node address be passed down to lower protocol layers when a packet is transmitted, and be passed up to higher protocol layers when a packet is received. (Note that when the ALERT2 protocols are extended to support two-way communication, a similar mechanism will be needed to pass destination addresses down and up the protocol stack. Therefore, it may make sense to create this mechanism now and extend it in the future, as necessary).

I believe that these recommendations will dramatically simplify the evolution and extension of the ALERT2 protocols to provide additional functionality. In particular, it would be very desirable to avoid having the lower level protocols inspect a SA field in the application layer protocol, because this will complicate the evolution of both the link-layer and the application-layer protocols.

I also recommend that a mechanism be defined now that optionally carries the address of the node to which a packet is destined. I refer to this field here as the destination address or DA field. This DA field will be necessary if the ALERT2 protocols are extended to permit two-way communication or to enable control of actuators at remote nodes. The DA field should be defined in the ALERT2 link-layer (AirLink) protocol, since it will be used to route the packet to the destination node (which will generally not be the base station). It would be useful to specify a mechanism that would permit the DA field to be elided when it is not needed, perhaps by defining a flag in the ALERT2 link-layer header that indicates whether a DA field is present. Implementations of version 1.0 of the ALERT2 protocols should be capable of skipping this field, if it is present.

I suggest that an explicit, optional, network-layer header be specified now, in order to support the future evolution and enhancement of the ALERT2 protocols. The network-layer packet header ought to be located between the link-layer header and the application-layer message. A flag in the link-layer header could be used to indicate whether the network layer header is present. Some mechanism is needed to permit current ALERT2 implementations to skip the network-layer header when it is present, such as a length field or defining the fixed length of the network-layer header. Again, implementations of version 1.0 of the ALERT2 protocols should understand how to skip this field.

Elsewhere, I have suggested that these 16-bit addresses be divided in to two fields: a network address and a local node address. The network address portion of the 16-bit address would be assigned to networks, while the local node address would identify a node within a network. This convention could be used to help ensure that any message received from another network, perhaps because of anomalous propagation conditions, is quickly discarded. This would help avoid peculiar behaviors that might result if such a message is accepted by a network for which it is not intended. This approach would also minimize the coordination that is necessary between systems in the same geographic area (because only a network number, rather than a range of addresses, must be assigned to each network).

3.3. 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. I believe that the objective of ALERT2 timestamps should be to record, as accurately as possible, the time at which a sensor reading was taken.

I recommend that the ALERT2 timestamping strategy be thoroughly reexamined, with the objective of making timestamps simpler to create and process, as well as more accurate. I also recommend that the ALERT2 timestamping architecture be thoroughly documented in one place, perhaps an ALERT2 system architecture document.

I suggest that a 16-bit timestamp be specified that is defined as the number of seconds since 12:00 a.m. or p.m. UTC, whichever is more recent. This timestamp is half the size of the 64-bit POSIX timestamp currently specified for some ALERT2 timestamps, but is large enough to avoid ambiguity. These smaller timestamps would be sufficient for everything except for sensor data reports that are more than 12 hours old, in which case the 64-bit POSIX timestamp could be used. This smaller timestamp *may* make it possible to use timestamps, rather than estimates of when a packet was received, without significantly increasing the amount of data that must be transmitted.

The ALERT2 protocols might be simpler if eight-bit time offset values referred to a 16-bit, seconds-since-noon/midnight-UTC timestamp contained in a message, rather than to the time that a frame was transmitted or received. Having the eight-bit time off refer to a 16-bit absolute timestamp would simplify the computation of when a sensor reading was actually taken. For example, it would no longer be necessary to recursively accumulate time offsets for messages that have been forwarded through more than one repeater. Likewise, special processing would not be required for sensor readings that miss a TDMA timeslot because the sensor data would not fit in the current TDMA time slot, perhaps because a large volume of sensor data was generated during a significant rain event. Furthermore, using these 16-bit UTC timestamps, rather than receive times, would avoid inaccuracies due to variable latency that results from queuing delays within a node or the variable time required to decode FEC encoding.

The ALERT2 specifications should probably specify how accurately a node’s clock must be synchronized with UTC. Techniques should be employed to ensure that the node’s clock remains synchronized within the specified tolerance. For example, nodes should measure the drift of their clocks, and correct their time-of-day clocks to account for that drift. Nodes might also correct their clocks based on the ambient temperature. A node might power up its GPS receiver occasionally to resynchronize its clock, depending on the historic drift of the node's clock. Of course, time synchronization would be much simpler if ALERT2 "modems" could receive as well as transmit.

I suggest that ALERT2 repeaters be required to have clocks that are synchronized with UTC.

To accommodate ALERT nodes and ALERT2 nodes that don't have clocks that are synchronized with UTC, the first ALERT2 node that has a UTC-synchronized clock that receives a packet without a valid timestamp, should set the timestamp.

3.4. Simplified ALERT2 Multiplexing Protocol

It appears to me that the recursive structure of the ALERT2 multiplexing protocol *may* be more complicated than necessary. It might be worth considering two alternatives: an iterative, rather than recursive, structure and an application-layer multiplexing protocol, which is outlined below.

An application-layer multiplexing strategy might be much simpler than what is described in this specification. It has the disadvantage of making ALERT2 repeaters application-layer devices (i.e., ALERT2 repeaters must understand at least the structure, if not the meaning, of application-layer messages). But, this cost doesn't seem particularly large, plus ALERT2 repeaters are already required to understand what are essentially ALERT application-layer messages.

This simpler ALERT2, application-layer multiplexing protocol could be easily constructed if some additional message type codes were added to the Self Reporting Sensor (SRS) Protocol, if explicit link-layer and network-layer headers were specified, and if ALERT and ALERT2 node addresses shared a common 16-bit address space.

The following Self Reporting Sensor Protocol TLV type codes might be sufficient to permit sensor reports to be multiplexed into a single ALERT2 SRS message. The SRS TLV type code would indicate:

- Whether the value field of this TLV contains a general sensor report, a tipping bucket rain gage report, a compressed ALERT message, or a sequence of Self Reporting Sensor Protocol TLVs;

- Whether the value field of this TLV contains an contains a 16-bit ALERT/ALERT2 node id; if the TLV doesn't contain a node id, then the node ID in the link- or network-layer header should be used;

- Whether the value field of this TLV contains a 16-bit UTC timestamp; and

- Whether the value field of this TLV contains an 8-bit offset from the "nearest" nested 16-bit UTC timestamp.

I think that this approach might yield an application-layer message format that replaces all of the existing ALPIDs (2, 3, 4, 6 and 7).

A disadvantage of this approach is that it may not provide multiplexing for future ALERT2 application-layer protocols. On the other hand, if these new application-layer protocols don't need timestamps (e.g., a bulk data transfer protocol), then it would probably be beneficial to create a new multiplexing mechanism for the new application-layer protocol anyway.

The following diagrams and text outline the Simplified Self-Reporting Sensor (S2RS) protocol. The S2RS, once it is fully developed, may be able to replace all of the existing application layer protocol and multiplexing protocols with a single, simplified protocol. Presumably, this single, simplified protocol would be much easier to implement, compared the two protocols and five ALERT2 message types (ALPID type codes) that it replaces.

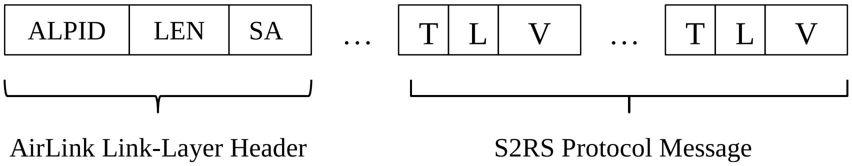

Figure 1. Structure of Proposed Simplified Self Reporting Sensor (S2RS) Protocol Message

ALPID Application Layer Protocol ID for Simplified Self Reporting Sensor (S2RS) Protocol
LEN AirLink frame length
SA Source address in AirLink link-layer header

T S2RS sensor report type code
L S2RS sensor report length
V S2RS sensor report value

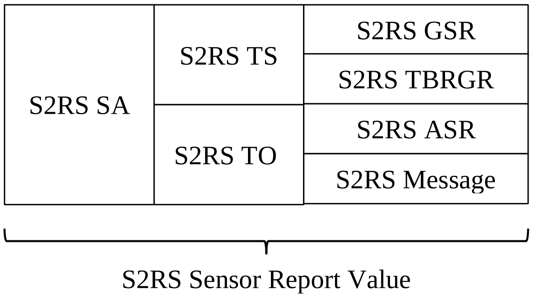

Figure 2.Structure of Proposed S2RS Protocol Sensor Report Value

S2RS SA Source address of node originating sensor data; optional; presence indicated by S2RS type code

S2RS TS	16-bit timestamp; optional; presence indicated by S2RS type code
S2RS TO	8-bit time offset; optional; presence indicated by S2RS type code; offset from nearest enclosing S2RS TS value, or the time the packet was received, if no S2RS TS field is present
S2RS GSR	General sensor report value (SL/SV structure)
S2RS TBRGR	Tipping bucket rain gage report value (SL/Accum/TO structure)
S2RS ASR	ALERT sensor report (compressed ALERT report value)
S2RS Message	One or more S2RS sensor reports (T/L/V structure)

3.5. Detailed Technical Comments

Title. I suggest that the title of this document be something like: “ALERT2 Media Access Network Transport (MANT) Protocol Specification, (complaints below about the use of the term “transport” notwithstanding). This specification describes a protocol, not a “layer”. (More precisely, this protocol is probably a sublayer above the network layer.)

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. The Media Access Network Transport (MANT) is a protocol, not a layer. Language such as “ALERT2 protocol suite” or “suite of ALERT2 protocols” would make this clearer.

Page 7, paragraph 1. “Media Access Network Transport (MANT)”. The term “transport protocol” has a precise meaning, namely an end-to-end protocol than manages the transfer of application data. I suggest that “transport” not be used in the name of this protocol.

Page 7, paragraph 2. “The MANT Layer is the layer …”. MANT is a protocol.

Page 7, paragraph 2. The term “end user data” is ambiguous. Perhaps, say something like: “The MANT protocol is responsible for forwarding ALERT2 packets through an ALERT2 network.” Perhaps, continue with something like: “The MANT protocol includes a multiplexing capability that permits multiple ALERT2 packets or multiple ALERT messages to be transported in one ALERT2 packet.

Page 7. A description of how the protocol 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, section 2. Please rewrite this section for clarity. This section should provide an introduction to the rest of the document. It should describe, in general terms, what the MANT protocol *does*, rather than *how* it operates. It appears to me that the MANT protocol provides two broad services:

- It forwards ALERT2 packets through an ALERT2 network to their destinations (specifically the network's base station).

- It can multiplex multiple ALERT2 messages into a single ALERT2 message, and can multiplex multiple ALERT messages into a single ALERT2 message.

Page 8, paragraph 1. Please rewrite this paragraph for clarity.

Page 8, paragraph 1. "independent of the ALERT2 Protocol Message Type". I don't understand what the point of this phrase is. Is this text trying to say that the MANT protocol can multiplex multiple ALERT2 application-layer messages (e.g., Self Reporting Sensor Protocol messages) into one ALERT2 link-/network-layer packet? If so, just say so.

Page 8, paragraph 1. "ALERT2 Application Layer data payload". It might work better to simply say "ALERT2 application layer messages".

Page 8, paragraph 2. Please rewrite for clarity. This notion probably requires more than two sentences to describe. I think that the text is trying to say that ALERT2 nodes can add additional headers after the link-layer header. (I think the link-layer header is recreated at each hop, but I'm not certain.) If this is the case, then I would ensure that this added information is described as "headers".

Page 8, paragraph 3. Please rewrite for clarity. The concept of "added headers" hasn't really been described yet.

Page 8, paragraph 3. "…added headers can be used by the ALERT2 modems and repeater nodes …". I would consider simply saying "added headers can be used by ALERT2 nodes". I think that it may not be prudent to put all of this functionality into the ALERT2 modem. I think that some of this information will need to be provided by the host or processor in the node (e.g., the destination address, should this functionality be added). I think that ALERT2 systems will be much easier to configure if the AirLink modem requires little or no configuration. Furthermore, minimizing the functionality implemented in the AirLink modem will make it easier to use different modems in the future, such as commercial, off-the-shelf modems that support very narrowband, 6.25 kHz channels.

Page 8, paragraph 3. "This intermediate layer processing is similar to the processing done in the Media Access, Network and Transport layers of standard protocol stacks …". Not really. The MANT protocol appears to be a primitive network-layer protocol that also provides a multiplexing service (which might be described as a sub-layer just above the traditional network layer). The MANT protocol isn't similar to traditional transport protocols, and does not obviate the need for a future ALERT2 transport protocol, should one be needed.

Page 8, paragraph 3. See earlier comment about MANT being a protocol, not a layer. Also, ALERT2 is a suite of protocols, not a single protocol.

Page 8, paragraph 4. "This document discusses an initial use of this ALPID and A2Protocol Payload independence …". Please rewrite for clarity. (What is "A2Protocol Payload independence”?) I think the text might be trying to say that this document discusses the multiplexing service provided by the MANT protocol.

Page 8, paragraph 5. “An example network architecture is used to show how the independence of ALPIDs from the A2Protocol Payload allows robust communications capabilities.” First, what follows is an example network *configuration*, not a network *architecture*. It is not obvious that the claim of robustness is supported by the text of this document. Perhaps, consider adding a description of the benefits of this protocol, although it is not entirely clear that such material is appropriate for a protocol specification. Perhaps, a discussion of the benefits of the ALERT2 protocols could be put in an ALERT2 overview document. Furthermore, I think the text should talk about the multiplexing capabilities of the MANT protocol, rather than “how the independence of ALPIDs from the A2Protocol Payload” enables this functionality. This proclaimed independence is a very subtle feature that many readers won’t understand, at least until the presentation of this mechanism is enhanced considerably. Plus, it might be better to explain what is being provided, rather than how it is being provided (or the mechanism that makes it easy to provide).

Page 8, paragraph 5. This material really needs one or more diagrams in order to be easily understood.

Page 9, first unnumbered bullet. It should probably be noted that the repeater will need both a traditional, 300 bps ALERT-compatible modem and an AirLink modem.

Page 9, paragraph 2. “architecture” -> “configuration.

Page 9, paragraph 3. “architecture” -> “configuration.

Page 9, paragraph 3. “What is described is hopefully applicable to … two-way communications …”. I believe that the MANT protocol will require considerable enhancement to support two-way communications.

Page 9, paragraph 3. “… two-way communications (for reliable communications) …”. Two-way communications will provide more than reliable communications. It will enable the remote administration of ALERT2 nodes, will permit ALERT2 networks to control actuators at remote nodes, will allow system administrators to download log files from remote nodes, will enable system administrators to upload new software to remote ALERT2 nodes, and will generally make ALERT2 networks much more functional and simplify the administration of these networks.

Page 9, section 2.1. I don’t understand the purpose of this paragraph. Yes, the contents of ALERT2 application-layer messages should be opaque to the lower protocol layers. But, what relevance does the “ALERT2 Message Type field” (presumably the message type code used in the sensor reports transported by the ALERT2 Self Reporting Sensor Protocol) have to do with this concept?

Page 9, last paragraph. Is a pass-list processing (packet filtering) capability part of the MANT protocol? Are ALERT2 repeaters required to implement this capability? As I understand it, the pass-list facility is a simple means of manually configuring routing, and so should probably use addresses in the link- or network-layer header. If packet filtering is part of the ALERT2 protocols, is should be specified somewhere.

Page 9, last paragraph. Does this paragraph imply that packet filtering uses the source address field in the ALERT2 Self Reporting Sensor Protocol messages? If so, I believe that this is a bad idea and will handicap the evolution of the ALERT2 protocols. For additional discussion on the topic, see the section "ALERT2 Addressing Architecture" section at the beginning of this section.

Page 10, paragraph 1. "The data payload is processed …". Should this say: "ALERT2 application-layer protocol messages are processed"?

Page 10, paragraph 1. The second sentence doesn't make much sense. Please rewrite for clarity.

Page 10, paragraph 1. Please add a reference.

Page 10, paragraph 2. "Except for address processing, ALERT2 Applications Layer Protocol Specification can be done independently…". Again, does the MANT protocol or do ALERT2 repeaters access the source address field in the ALERT2 application-layer protocol (and presumably future ALERT2 application-layer protocols)? If so, this is probably a bad idea, as noted above. Also, see comments on a possible simplified multiplexing protocol.

Page 11, section 3. This section might be much easier to understand if it started by describing the syntax of MANT protocol messages. This would seem to include ALERT2 messages with ALPID values of 3, 4, 6, and 7. Then, the descriptions of how these packets are processed will be much easier to understand.

Page 11, paragraph 1. Please add a reference for this material.

Page 11, paragraph 3. "A source address (SA) is added to each ALERT2 message …". Is this the SA that the AirLink modem writes just after the AirLink reserved bits?

Page 11, paragraph 3. What does an AirLink frame look like after it has passed through several repeaters? A detailed example with diagrams would be valuable here.

Page 11, paragraph 4. "The originating modem adds its source address to an ALERT2 message." Is this behavior identical to that described in the preceding paragraph? If so, why not just say that all AirLink modems insert the node's SA just after the AirLink reserved bits?

Page 11, paragraph 4. I don't believe that the AirLink specification requires an AirLink modem to write the SA in an outgoing frame.

Page 11, paragraph 5. “In the future, the source address may be used in conjunction with a destination address for routing and two way communications networking.” This should probably be moved to another section that describes future enhancements, including how two-way communication will work.

Page 11, paragraph 5. “The source address is independent of the A2 Protocol Payload id information.” Does this mean to say that the link-layer SA may be different than the SA value contained in the Self Reporting Sensor protocol messages? If so, just say it.

Page 13, paragraph 1. “The repeater receives both ALERT and ALERT2 transmissions on a single inbound aloha channel …”. Please note that two modems, an ALERT modem and an ALERT2 modem, are required. Otherwise, some readers might wonder whether the ALERT2 modem can simultaneously receive ALERT and ALERT2 frames.

Page 13, paragraph 1. “The repeater holds the received messages until its TDMA transmission time slot.” Must the outbound channel of an ALERT2 repeater use TDMA?

Page 13, paragraph 2. Please specify the translation between a four-byte ALERT message and the three-byte compressed format.

Page 13, paragraph 2. This paragraph would be easier to understand if diagrams of the time-tagged, three-byte compressed ALERT messages and of the ALERT Concentration Protocol messages were included here.

Page 13 paragraph 3. Again, including and using a diagram will make this text much easier to understand.

Page 13, paragraph 3. “ALERT2 transmissions received are decoded, time tagged with the Time Offset parameter …”. Is there any provision for the case when a received message can’t be transmitted in the next TDMA time slot?

Page 13, paragraph 4. “The “ALERT2 message” is comprised of …” I suggest using a more unique term for this structure. It seems that I have read numerous uses of this term. However, I’m not sure that they all mean the same thing.

Page 14, paragraph 1. See my earlier comments about a potential, simplified multiplexing protocol.

Page 14, paragraph 2. The second sentence is true. Is this capability used in this specification, or is this merely an observation?

Page 15, paragraph 1. “… transmission occurs immediately so the time offset is zero”. This is true only if the outbound packet is generated immediately prior to transmission. In some configurations, it seems that it may be possible for the outbound channel of a repeater to be overrun, in which case queuing and maybe even packet loss may occur.

Page 17, paragraph 2. This API needs to be documented. It appears to be different than what is documented in the ALERT2 AirLink Protocol Specification.

Page 19, paragraph 1. How is the “errored message forwarding” enabled? Local, manual configuration of the AirLink modem? Or, can this be done remotely?

Page 20. It appears that this glossary was copied directly from the AirLink specification. The usage of some terms in this document appear to conflict with this glossary. There are probably some terms used in this specification that ought to appear in this glossary.

Page 20, “Block”. This describes a structure within an AirLink frame, and doesn’t appear to be applicable to, or even used in, this specification.

Page 20, “Header”. This definition of “header” does not match its use in this specification (or common usage).

Page 20, “Packet”. This is usually called a “frame”, not a “packet” (or at least the modulated portion of this signal is commonly called a frame).

Page 21, References. Please add the appropriate references.

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 an application even been submitted?

4.2. Detailed Editorial Comments

These editorial comments are somewhat limited, and assume that this specification will be substantially reorganized, rewritten, and expanded, and may be made available for another period of public review and comment.

- 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 7, paragraph 2. “This protocol initially supports three message types …” Consider using bullets to highlight the three message types.

- Page 8, paragraph 1. “what we called the PID”. This information is extraneous. The document might read better without this phrase.

- Page 8, paragraph 4. I recommend using the term “remote node” rather than “gage”. An ALERT2 node is more than just a gage.

- Page 12, paragraph 2. Please don’t rely on colors. Readers may print this specification on a black and white printer.

- Page 12, paragraph 2. I suggest that the physical layer structure of an ALERT2 frame be displayed once, and then the physical layer header and trailer (CO, BS, FS, and FEC fields) can be deleted from the remaining diagrams. This will help simplify the diagrams and make them easier to understand.

5. Conclusions and Recommendations

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

In my view, this draft of the ALERT2 MANT Layer Protocol Specification lacks the detail and clarity necessary to ensure that implementations of the MANT protocol developed based only on the information contained in the specification are likely to interoperate.

I recommend that the ALERT2 MANT specification be substantially reorganized, rewritten, expanded, and edited, as outlined in the comments above. The Technical Working Group may wish to release the updated document for another period of public review and comment.

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.