Proposal for ALERT2™ over IP

Presented to the ALERT2 Technical Working Group

By the ALERT2 over Non-AirLink Interface subcommittee
James Logan
Sam Utley
David Leader

With support from:
Ilse Gayl
Claus Strasburger
Glenn Hetchler
Carrie Lery

Introduction

ALERT2™ has proven to be a next generation improvement over ALERT, meeting the goal of creating a replacement that addressed ALERT's shortcomings. Upgrades include more efficient use of the channel, additional address space for sites and sensors, elimination of erroneous data, and almost no data loss.

The current ALERT2 AirLink specification uses radio frequencies (RF) as the physical transport layer, usually in the VHF/UHF RF bands. Although ALERT2 over RF is the logical replacement for ALERT, there are situations in which RF isn't practical, or where other forms of physical transport could provide alternatives superior to RF.

Examples of situations in which ALERT2 over other media would be useful include:

- No line-of-sight radio network is available (single site in a remote location, or group of sites but with no practical base station location).
- Highly-resilient communications could be achieved using redundant data feeds (Simultaneous transmission of radio and IP gives redundancy from a repeater site).
- ALERT2 decoder output, IND, over high cost-for-bandwidth networks (such as IP over satellite) is prohibitively expensive. IND messages are approximately four (4) times larger than the binary content alone.
- If a repeater location also has Internet access available, there is potential for a redundant data feed, thus eliminating an otherwise single channel as a single point of failure.

- A high value monitoring location or a small ALERT2 network could be connected to a base station via satellite, but today's satellite fees for transmitting full base station decoder output make that cost prohibitive. Transmission of only the MANT PDU ALERT2 message portion could reduce considerably the bandwidth required, yet provide the same information with the same (or better, if delivery is satellite-guaranteed) reliability.

Examples of potential Non-AirLink uses of ALERT2 include:

- ALERT2 over IP (includes ethernet, cellular data, other IP-based networks)
- ALERT2 binary for satellite (narrow-band, high cost)
- Internet of Things Sensors (includes LoRa WAN and other Low Power WAN (LPWAN) where we use ALERT2 content in application layer)

Overview

This document is a proposal for ALERT2 over IP. It is also a template for how ALERT2 over other below-MANT transport layers could be defined.

We want to be consistent with our original goals:

1. The protocol must reside in the public domain.
2. The protocol must provide a common (over each transport layer) interface to be available for manufacturers and system integrators.
3. The protocol must improve performance over ALERT, primarily low channel capacity and high data loss.

Additional goals include:

4. Maintain compatibility with existing ALERT2.
5. Extend use of ALERT2 over other transport for situations with no radio path or radio network.
6. Leverage extensibility of the existing ALERT2 protocol (app layer can continue to evolve).
7. Define gateway behavior for ALERT2 to convert over other transport layers.

ALERT2 Protocol Architecture and Proposed Update

The ALERT2 protocol is defined with three logical layers: The Application Layer, the MANT Layer, and the AirLink Layer. Each of the three layers has distinct responsibilities in support of the ALERT2 protocol.

Application Layer

The Application Layer supports the encoding and decoding of data into and out of formats and structures used by ALERT2 applications. Application layer protocols are formed by the sender software into structures understood by receiving application software.

MANT Layer

The MANT Layer provides addressing, port multiplexing, acknowledgement, and other services to logically transport application and network control data across an ALERT2 radio network.

AirLink Layer

The AirLink layer in the ALERT2 protocol corresponds in the ISO 7-layer stack to a Physical layer plus a Link Logic layer to put information on and take it off the radio channel. Using the ISO stack, for each physical substrate, the associated link logic must manage encoding, media access, etc., appropriate to that medium.

We propose to extend the ALERT2 protocol over other Link Logic-Physical layers; the RF-based AirLink layer is but one of the media on which the MANT and Application layers can complete their activities.

Thus we propose a new name for the bottom layer: Network Access Layer. Borrowed from the TCP/IP stack naming convention, the Network Access Layer encompasses the Physical Layer and the substrate-specific Link Logic that brings information into and out of the Physical Layer to pass to and from the MANT Layer.

Figure 1A, below, shows the original ALERT2 protocol stack and Figure 1B, the proposed stack.

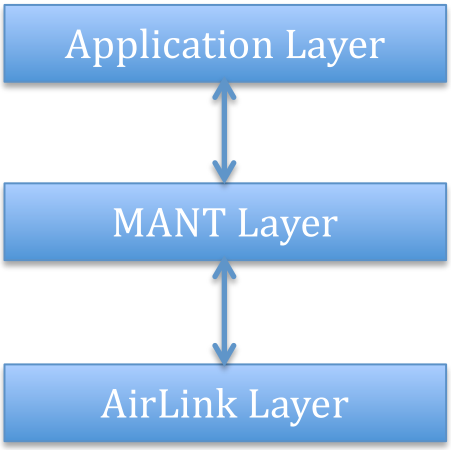

Figure 1A
Ver. 1.1 ALERT2 Stack

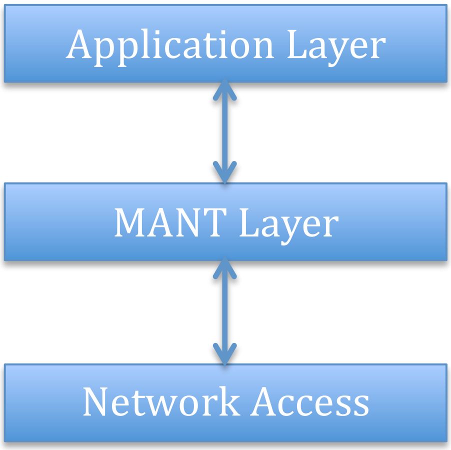

Figure 1B
Proposed ALERT2 Stack

Network Access Layer

For each physical substrate, the associated link logic must manage encoding, media access, etc., appropriate to that medium. The Network Access layer encompasses these functions.

Besides the existing AirLink protocol for the Network Access layer, other potential Network Access layers include Internet, Satellite, Cellular, and other RF formats. The AirLink and other potential new Network Access layer protocols are described in brief:

AirLink Protocol

The ALERT2 AirLink is an implementation of the Network Access layer on the RF Physical layer that transmits PDUs received from the MANT layer to a receiving device that reconstructs the PDU and delivers it to the MANT layer on the receiving device.

A2overIP Protocol (proposed)

The A2overIP is an implementation of the Network Access layer that uses IP networks to transmit the PDUs received from the MANT layer to a receiving logical-physical device that reconstructs the PDU and delivers it to the MANT layer on the receiving device.

Other potential Network Access layer protocols include:

- A2 over Short-burst satellite (for example, Inmarsat IDP or Iridium)
- A2 over IoT (for example, LoRa)
- A2 over Packet Radio (for example, P25 Radio)

Our objective in this document is to define the protocol for sending ALERT2 over Internet Protocol, or A2overIP.

A2overIP Protocol

Unique attributes needed for IP communications, not in existing ALERT2:

Transmit:
Source IP Address
Destination IP Address and Port
Receive:
Receive IP Address and Port

Transmitting ALERT2 over TCP/IP

The transmission control protocol (TCP) transmission method is a connection-oriented, session-based internet protocol (IP), guaranteeing delivery of each IP packet to its destination and assuring its contents were protected. Because of the care taken by TCP (waiting for and ordering missing packets, and potentially retrying for missing or damaged packets), this protocol is relatively costly in time and bandwidth required.

A TCP/IP socket is a session connection initiated by a client to a waiting server.

In the broadcast form of ALERT2, the remote site is configured as a client that connects to send its data in from the field, and the waiting server is the data receive point (sending-data-in scenario, see Figure 2, below).

We also define a scenario in which the remote site is configured as a server waiting to have its queued data received over a socket initiated by a data-receiving client (bringing-data-in, see Figure 3, below).

Which client-server configuration is best in a given situation will likely be determined by power resources and network security constraints.

For the sending-data-in scenario, two different approaches for use of A2overIP TCP-guaranteed, remote client-initiated socket connections are described to cover various ALERT2 data scenarios:

> **Discrete Push A2overIP** transmission connections are time-minimized. In the Discrete Push case, for example, distant or low-power stations with costly transmission time can push their data over short bursts, freeing up the channel and going to sleep between transmissions.
>
> **Streaming Push A2overIP** transmissions leverage high bandwidth and plentiful power, usually found at industrial repeater sites. They are highly efficient in that connections stay open and are ready for data as they are received. In the Streaming Push case, for example, a heavily-used ALERT2 RF repeater can also send its full data stream over a parallel land- or microwave-based IP network.

Both Discrete Push and Streaming Push use the same communications methodology. The only difference is the length of the connection. Termination of connection can be managed by the application, but maintaining a connection cannot fully be controlled by the application and has dependency on the network path and networking rules applied to components on the network path.

For the bringing-data-in scenario:

> **Receive Pull A2overIP** connections are initiated by the data receiver. The data receive client initiates a socket connection to the remote server. Once a socket connection is made, it is held open until the server end has streamed all of its data and then it is closed.

In all cases, the “data” transmitted are one or more whole MANT PDUs.

Keepalives / acks for TCP connections

TCP has a keep alive option for maintaining connections, but the state is not directly available to the application. Per the TCP specifications, connections can be maintained indefinitely and do not need resources if not in use.

But, we know from experience that firewalls and NAT routers break the spec by forcing termination of idle connections. Keepalives can be used to help maintain connections. For those situations where we want to maintain a connection but connections are unexpectedly broken, keepalives are one way of maintaining the connection.

Connection initiated by the data sender

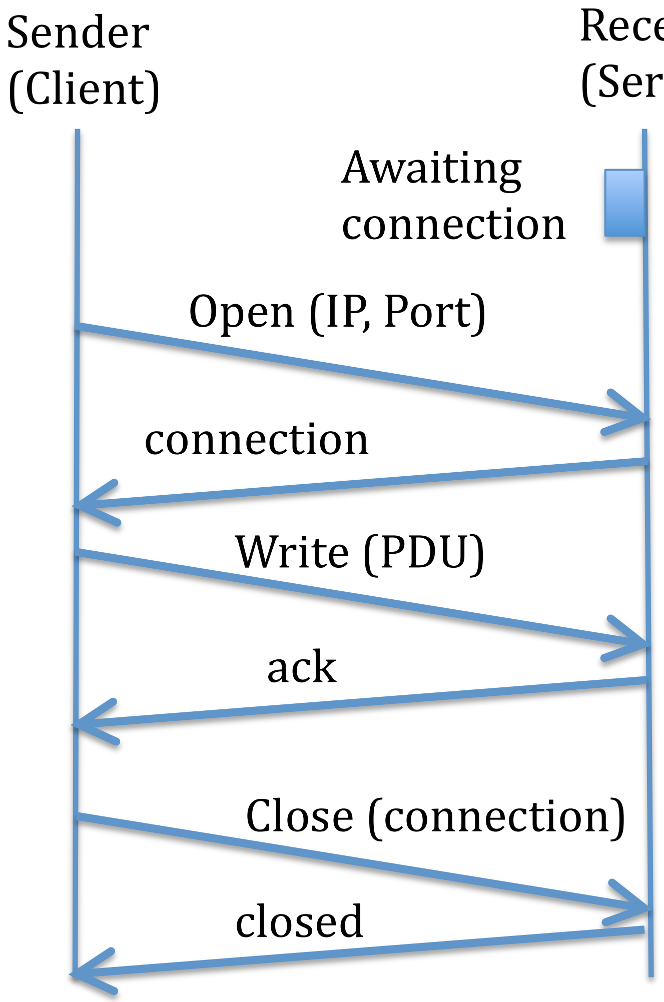

Figure 2. Sending-data-in scenario, in which the data sender is the TCP client and the receiver is the TCP server. The data sender initiates the TCP connection. The ack in the above example can either be the TCP ack, or the MANT ack if the MANT EERDS Protocol is enabled.

Connection initiated by the data receiver

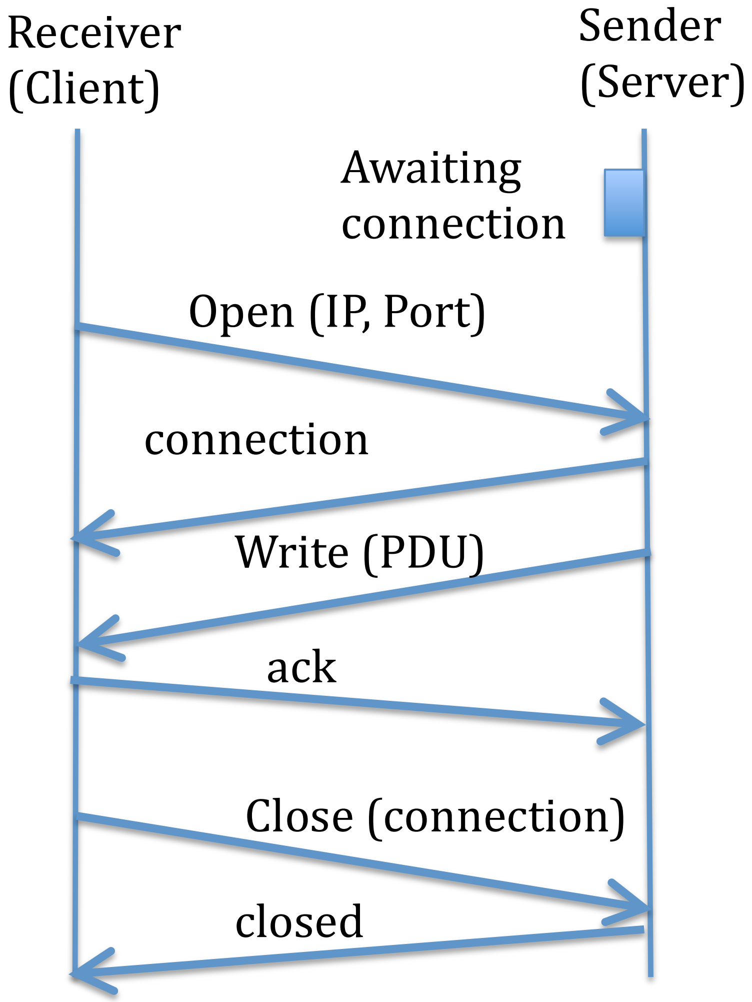

Figure 3. Bringing-data-in scenario, in which the data receiver is the TCP client that initiates the connection, and the remote data sender is the TCP server waiting for the client to connect and receive its data. The ack in the above example can either be the TCP ack, or the MANT ack if the MANT EERDS Protocol is enabled.

Functional Definitions: Discrete Push, Streaming Push, and Receiver Pull Types

Discrete Push: Whenever the A2overIP client device has something to send, it opens a connection, pushes its content, and closes the connection.

Streaming Push: When the A2overIP client device has data to transmit, it:

- Checks to see if it has a connection.
- If it does not have a connection, it creates a connection to the destination IP address and port.
- When it has a connection it continues pushing MANT PDUs through the connection until there is nothing more to push, triggering the timeout counter.
- The connection stays open as long as the timeout doesn't complete (timeout is the permissible connection time with no data to push).

Receiver Pull: Whenever the A2overIP receiver client device is ready to receive data from a sender, it opens a connection to the sender's server, pulls the sender's data across the connection, and closes the connection.

Transmitting ALERT2 over IP using UDP

User datagram protocol (UDP) connectionless internet protocol is the minimalist UDP form of internet protocol. It provides no guarantee the IP packet was delivered to its destination, or any information regarding a packet's welfare beyond it having been sent. A checksum is the only validation service offered at the datagram level at the receive end.

UDP is fast and cheap. User datagram packets are broadcast on the host network, there is no checking of delivery order of packets and thus no waiting for missing packets.

UDP is also insecure and often filtered out by firewalls and other security methods.

Using UDP, ALERT2's MANT-level end-to-end reliable datagram service protocol (EERDS) is the only method by which to guarantee delivery. The retry mechanism is defined in the ALERT2™ MANT Protocol Specification.

At this time, we don't believe UDP provides a relative benefit that makes it worth recommending for ALERT2 use.

Gateways: Transmitting ALERT2 over XXXX to ALERT2 over YYYY

We want to create gateway behaviors that enable ALERT2 transmission from one type of transport (XXXX) to a second transport (YYYY). A gateway allows the possibility of listening for MANT PDU packets on one transport layer (XXXX) and then retransmitting the PDU on a different transport layer (YYYY). The MANT PDU content is maintained across the gateway. This gateway method allows for interoperability between various transport layers.

A gateway, for example, could be listening to AirLink (RF) ALERT2 transmissions and retransmitting them over A2overIP through a satellite IP link, or vice versa.

Secure ALERT2 Transmission over IP

Some applications of A2overIP may have the need for data transmission security. SSL is a secure cryptographic protocol that allows two applications communicating on an IP network to have communications security. Transport Layer Security (TLS) or Secure Sockets Layer (SSL) are both referred to as SSL. The two sides of the connection share a cryptographic key. This protocol is used for HTTPS web browsing, secure email, and secure voice-over-IP (VoIP).

SSL is highly configurable, so both sides would need to have compatible configuration in order to connect properly.

In our view it is potentially insecure to use dynamic configuration sharing for the ALERT2 protocol. This document won't specify the configuration for SSL, but it does require that entities connecting A2overIP via SSL share their SSL configuration with each other in advance of going live, rather than on a session basis.

Glossary

IND	Intelligent Network Device Application Programming Interface
IP	Internet Protocol
PDU	Protocol Data Unit – a protocol described packet of data with header that includes type, and length and the enclosing data.
Port	For ALERT2 MANT Layer, Incoming serial port on the IND device
Port	For IP, Sub address used for internet connection IP Address and Port
Socket Connection	Communications on networks are mapped to IP Addresses and Ports
SSL	Common name given to TLS and SSL. Secure Sockets Layer is the older version, replaced by TLS.
TCP/IP	Transmission Control Protocol (TCP) - A set of rules that govern how two applications can maintain a conversation over Internet Protocol (IP). Includes error correction.
TLS	Transport Layer Security, the modern version of SSL
UDP/IP	User Datagram Protocol (UDP) - Low latency, loss tolerant protocol for communicating over the internet. Includes checksums. No handshake, no ordering, no duplicate protection. UDP protocol is appropriate when the application handles error checking and correction.

References:

ALERT2™ AirLink Layer Specification, Version 1.1, March 2012

ALERT2™ MANT Layer Protocol Specification, Version 1.1, March 2012

ALERT2™ Intelligent Network Device Application Programming Specification, Version 1.1, June 2015