VOTER Protocol – Voice Observing Time Extension (for) Radio
Jim Dixon, WB6NIL
Version 1.0 – Aug. 11, 2013

VOTER is a completely connectionless UDP-based protocol (using IANA-assigned well-known port 667, originally assigned to me for a project I was doing for the United States Federal Election Commission, which, of course, deals mainly with an entirely different kind of voting). It provides for transmission of audio signals from remote radio receivers, along with ultra-precise (GPS-based) timing information and signal quality, allowing synchronization of multiple signals at the "head" end (hereinafter referred to as the "host") of the system and selection of which signal to use based upon the signal quality information and the consistency thereof. Audio and timing information may also optionally be sent back the remote location (hereinafter referred to as the "client") by the host to facilitate a multiple-site simulcast transmission system. In addition, it may serve as a simple, low-overhead audio stream protocol for general-purpose (non-GPS-based timed) purposes.

Security is provided by a Challenge/Response authentication technique using CRC-32 based digest information. Although it is by no means a terribly secure method (it is not even close to cryptographically-strong), it at least it provides somewhat of a deterrent for those attempting to circumvent the security.

This document applies to both GPS-based and general-purpose uses. All further references to these different modes of use shall be “GPS-based” (meaning that the Time data actually contains GPS-based accurate real time information) or “general-purpose” (which means the Time data contains audio packet sequence numbers, along with as accurate as possible (if any) representation of real time with whole-second accuracy).

Both the client and the host use the same exact messages as follows:

The VOTER protocol packet consists of a 24 octet header and optionally a payload, which currently can either be 160 samples (20ms) of Mu-law encoded audio, 320 samples (40ms) of ADPCM encoded audio, or GPS location and elevation information (typically sent once per second).

The VOTER protocol packet header consists of 24 octets as follows:

All date and time information is in GMT.

All multi-byte numeric fields are sent standard network-order (most significant byte first).

Octets 0-3: Current accurate (or as accurate as possible for “general-purpose” use) time in whole seconds, 32 bits (network order).
Octets 4-7: Current accurate fractional time in nanoseconds (or audio packet sequence number for “general-purpose” use), 32 bits (network order).
Octets 8-17: Authentication challenge, ASCII string, null-terminated (up to 9 chars long + null).
Octets 18-21: Authentication response (digest), 32 bits (network order)..
Octets 22-23: Payload type, 16 bits (network order).

Payload types are as follows (header octets 22-23):

Payload type 0 – Authentication plus flags. Flags must be sent by host. Flags may be sent by Client.

Octet 24: Flag bits, as follows (specified in decimal bit value):

1 – Flat Audio (NOT de-emphasized) if flag set, Processed audio (de-emphasized) if not set. (sent by host only)
2 - Send audio always, regardless of whether or not signal is detected if set, do not send audio unless signal is detected..if not set. (sent by host only)
4 – Do Not filter Sub-Audible tones from audio stream, if set. Filter Sub-Audible (<200Hz) tones from audio stream if set. (sent by host only)
8 – Master Timing Source Mode. If not set, payload 1,2 and 3 packets are delayed (approx 6 ms) to guarantee that the packets from the device designated as "Master Timing Source" are received by the host previous to packets from any other device. If set, no delay is performed, and the packets are sent immediately. This must be set on only one device in the network. (sent by host only)
16 – Send audio in 320 sample (40ms) packet IMA ADPCM (Intel/DVI block format) rather than 160 sample (20ms) packet Mu-law. (sent by host only)
32 – Operate in "general-purpose" mode (non-GPS-based). (sent by client, responded to by host)
(64-128) – RFU

Payload type 1 – RSSI information + Mu-law audio. Octet 24 is RSSI value 0-255 and Octets 25-184 are 160 samples of Mu-law encoded audio (20ms @ 8K samples/s Packets of this payload type are sent only when the receiver is receiving a valid signal (or the host is indication that a signal is to be transmitted).

Payload type 2 – GPS Information (optional) as follows:

Octets 24-32: Longitude value, in the format XXXX.DDY (eg 4807.038N), ASCII string, null-terminated.
Octets 33-42: Latitude value, in the format XXXXX.DDY (eg 01131.000E), ASCII string, null-terminated.
Octets 43-49: Elevation value in Meters (eg 545.4), ASCII string, null-terminated.

If Octets 24-49 are omitted, it is considered to be a "keep alive" packet (for maintaining NAT traversals and similar) and is done only by a "general-purpose" mode client without GPS information to send (since a GPS-based client or a "general-purpose" client with GPS would have GPS information to send which accomplishes the "keep alive" functionality as a side-affect).

Payload type 3 – RSSI information + IMA ADPCM audio in Intel/DVI block format. . Octet 24 is RSSI value 0-255 and Octets 25-187 are 163 octets (320 samples) of IMA ADPCM (Intel/DVI block format) encoded audio (40ms @ 8K samples/s Packets of this payload type are sent only when the receiver is receiving a valid signal (or the host is indication that a signal is to be transmitted).

Payload type 4 – RFU

Payload type 5 – “PING” (connectivity test). Octets 24 and up contain up to 200 bytes of payload for the purpose of evaluation of connectivity quality. When a client receives this packet, it is intended to be transmitted (with the payload information in tact) immediately back to the host from which it came. The actual contents of the payload are not specifically defined for the purposes of this protocol, and is entirely determined by the implementation of the applicable functionality in the host.

By definition, a digest value of 0 is an indication that the entity generating it has not received a validly recognizable digest from its peer. Therefore, a challenge must not be generated that would result in a digest value of 0.

A “Master Timing Source” must not use ADPCM audio, and obviously must not be using “general-purpose” mode. In a system where all clients are using “general-purpose” mode a “Master Timing Source” is not necessary.

The intended methodology for the use of the protocol is as follows:

Then the client starts operating, it starts periodically sending VOTER packets to the host containing a payload value of 0, the client's challenge string and an authentication response (digest) of 0 (to indicate that it has not as of yet gotten a valid response from the host). When such a packet is received by the host, it responds with a packet containing a payload value of 0, the host's challenge string, an authentication response (digest) based upon the calculated CRC-32 value of the challenge string it received from the client and the host's password., and option flags. If the client approves of the host's response, it may then start sending packets with a payload type of 1 or 3, the client's challenge string, and an authentication response (digest) based upon the calculated CRC-32 value of the challenge string it received from the host and the client's password, along with the 1 byte of RSSI information and 160 bytes of mu-law audio (and continues doing this every 20ms) as long as there is signal being received by the receiver.. If the host approves of the client's response, it then uses the audio and/or GPS information provided by the client (and the flow continues).

At any point, if either side does not approve of its peers authentication response, it must reply to its peer with a packet with a payload type of 0 (indicating that authentication need to take place).

The connection-less nature and the flexibility of this protocol allow the client to continue operating even of the host stops or changes its challenge string in mid-stream.

Basically, when a client starts up, it just starts sending out packets, whether or not the host is there, or responding or responding positively. Once it receives valid authentication information from the host, it then is to send periodic GPS packets (once every second), and audio packets (once every 20/40ms) when there is audio to be sent (a signal is present on the receiver, or there is a signal to transmit). If, at any point the client gets responses from the host that do no contain valid authentication, it merely goes back to sending authentication packets once again, and the whole process repeats. Similarly, any time the host receives a packet from the client that does not contain valid authentication, it sends an authentication packet to the client.

Therefore, the protocol is completely tolerant of interruptions in network connectivity, restarts of either or both sides of the connection, and accomplishes this quite easily and with very little complication and overhead.

In summary, there are 7 cases of packets:

24 octets, payload 0, digest 0 -- This is only sent by a client
24 octets, payload 0, digest (calculated) – This is only sent by a client
25 octets, payload 0, digest (calculated) ,flags
185 octets, payload 1, digest (calculated), 160 samples of Mu-law audio
24 octets, payload 2, digest (calculated) – This is only sent by client
50 octets, payload 2, digest (calculated),GPS data – This is only sent by client
188 octets, payload 3, digest (calculated), 320 samples of ADPCM audio
24-224 octets, payload 5, digest (calculated), up to 200 bytes for connectivity testing/verification

A host will not send a packet to a client until the client has send a packet for the host. Therefore the "24 octets, payload 0, digest 0" packets will only be sent by a client (since a digest of 0 indicates not yet hearing from a peer). A client may send as many payload 0 packets as is appropriate (there really is no limit). A host must respond to each of these packets.

Presence of payload 1 and/or 3 packets indicates valid signal received or to be transmitted.

For a "receive-only" implementation, the host may only send payload 0 packets in response to packets from the client. It need not send any payload 1 or 3 packets if doing so is inappropriate.
A client need not keep intrinsic track of date information (such as a real-time clock, etc). The client may, if no other means is available, determine current date (not including time of day) from the time information in the authentication packet last received from the host. Therefore, the host must send time information that has an accurate date (accurate time is not required).

“General-Purpose” mode operation:

For non-GPS-based uses, a client and host may send audio packet sequence numbers in lieu of accurate real-time information. In order to facilitate this operation, the client and host must negotiate “general-purpose” mode operation at authentication time by passing (and responding to) the “General-purpose” mode flag in the authentication packet.

Both the client and host must maintain a “sequence number” counter (the host must have a unique one for each client) which is intended to be initialized to zero upon initial connection to the host (completion of the authentication exchange), and be incremented every 20 milliseconds, synchronous with the transmission of any audio packets that may be sent (it must increment regardless of whether audio packets are actually being transmitted). Therefore, ADPCM (40 millisecond) audio streams will “skip” sequence numbers.

When an audio packet is transmitted, the local device's sequence number is represented as the “nanosecond” part of the real-time field in the packet header (as opposed to actual time). The receiver is intended to keep track of packet sequence numbers it receives from its peer, and “re-assemble” them as a “coherent stream” of audio based upon the sequential packet numbers generated at the sending point. If a device has whole-second real-time information (regardless of accuracy or sanity), it may send that data in the “seconds” part of the real-time field, although it is for informational purposes only and has no effect on any operation.

Since the sequence numbers are being sent as a 32 bit number, and it increments every 20 milliseconds, no session is intended to be up and valid for more than 1 year (365 days) (which is 1,576,800,000 Decimal (5DFC0F00 Hexadecimal) 20 millisecond intervals). If the session exceeds this limit, each peer is intended to re-establish communications (re-authenticate, etc).

It is important to keep in mind that in “general-purpose” mode, the 20 millisecond time interval is by definition approximate (although kept as accurate and consistent as possible). Unlike GPS-based mode, the timing is based entirely upon local processor clock sources, etc, and can (and probably will) vary. Therefore, accommodation must be made to allow tolerance of this.