ATSC 3.0 Broadcast Positioning System (BPS) Mesh Network

Mark T. Corl
Triveni Digital, Inc.
Princeton, New Jersey, United States
mcorl@trivenidigital.com

Vladimir Anishchenko
Avateq Corp.
Markham, Ontario, Canada
vlad_a@avateq.com

Francisco Girela Lopez
Safran Electronics and Defense
Rochester, New York, United States
Francisco.girela@nav-timing.safrangroup.com

Tariq Mondal
National Association of Broadcasters
Washington, District of Columbia, United States
tmondal@nab.org

Abstract – The Broadcast Positioning System (BPS) concept and related proof-of-concept technologies were introduced at the 2023 NAB Broadcast Engineering and Information Technology (BEIT) conference. Work on the BPS project has continued through 2023 and into 2024 with a focus on better accuracy and traceability but also on how such a system would be put into operation across one or more regions. This paper describes the design of a BPS transmitter mesh network that would allow nationwide time synchronization based entirely on ATSC 3.0 broadcasts. This network could also be used to obtain position information for devices receiving three or more BPS broadcasts. To accomplish this design, the proof-of-concept implementations presented at the 2023 NAB BEIT Conference were redesigned to meet the needs of this mesh network. The paper will describe the overall architecture and equipment needed to allow such a mesh network to be implemented. A nationwide monitoring system necessary to manage the mesh network will also be discussed. Finally, the paper will suggest areas of design and development needed before such a system could be realized.

Background

The preamble of an ATSC 3.0 frame carries the bootstrap emission timestamp. If this timestamp is accurate, a receiver can synchronize its clock using the ATSC 3.0 signal. Moreover, since the ATSC 3.0 signal can carry data, the location of the transmission antenna can be sent with the signal. Armed with the precise bootstrap emission time and the location of the antenna, a receiver at a known location can maintain a very accurate clock.

This paper is excerpted from the Proceedings of the 2024 NAB Broadcast Engineering and Information Technology (BEIT) Conference, © 2024, National Association of Broadcasters, 1 M Street SE, Washington, DC 20003 USA.

A prototype system was described at the NAB BEIT conference in 2023 [1] which demonstrated the ability to maintain the time within the ATSC 3.0 broadcast to an accuracy of ± 300ns. A new system has been subsequently developed that has improved this number by an order of magnitude. This new system will be described in more detail in the sections below.

Furthermore, another paper [2] on the BPS topic was also presented at the NAB BEIT 2023 conference that discussed the potential coverage of a BPS signal at various antenna heights and broadcast robustness. That paper showed that within the continental United States, if an antenna were mounted on each transmission tower at a height of 50m, it would be able to receive as many as 70 transmissions at a broad range of frequencies.

The new synchronization system along with the potential for receiving the signal from so many neighboring transmitters leads to an opportunity to build an extremely robust system of multiple transmitters, using multiple disparate frequencies and traceable time sources.

Phase II Receiver/Synchronizer Description

The ATSC 3.0 BPS Synchronizer is constructed from components derived from the original Avateq receiver used in the proof-of-concept. The solution is based on the Avateq RF front-end with an FPGA-based DSP chain contained within a 1RU box and a second unit for post-processing. Additional development of the FPGA-based DSP core was required to meet the phase II project objectives. The basic architecture was maintained from the original receiver proof-of-concept and shown below in Figure 2. The intention of the new design was:

a) to meet the phase II project technical requirements in full
b) to provide more flexibility for development, verification and debugging the synchronization chain and its components at all levels
c) to expose a set of standard interfaces for off-the-shelf available instrumental hardware - PCIe cards for additional NICs, PTP, I/O, Timing, GPS (if required), etc.
d) to deliver a versatile platform suitable for potential application related to the BPS and not limited to the time synchronization task.

The same platform is used to provide a receiver unit which allows validation of the broadcast signal. The use of existing hardware and software platforms allowed the end of 2023 delivery target to be met.

Figure 1 provides a diagram of the systems and functions contained within a synchronized BPS transmission system. The numbered paragraphs that follow the diagram correspond to the callouts in the diagram.

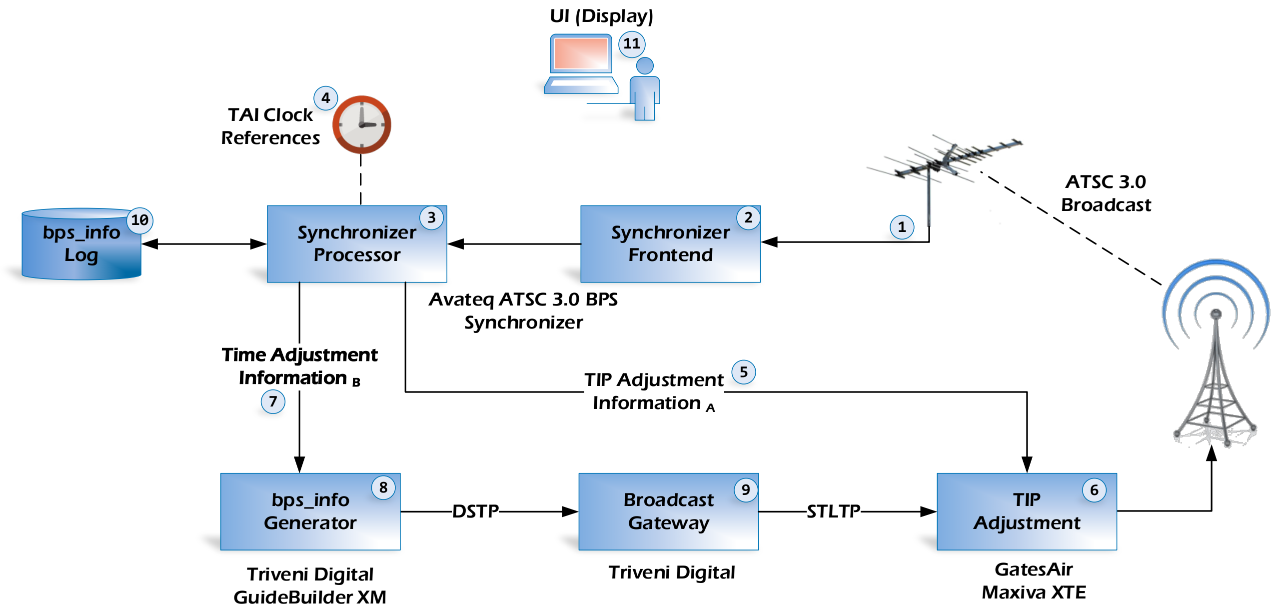

FIGURE 1: BPS TRANSMISSION SYSTEM DIAGRAM.

① The ATSC 3.0 transmission. Legal transmissions include a time stamp based on the current TAI time in the detailed preamble (L1D) according to A/322 [3]. Leap seconds are communicated in other signaling (e.g., LLS). The time represented by the L1D timestamp is referred to as the "Time Information Position" or "TIP" which describes the instant in time at the center of the leading edge of the first sample of the first symbol of the bootstrap when the wave form was emitted from the transmitter antenna. The previous version of the system allowed a delay factor to be entered to account for fixed delays in reception and that capability was maintained.

② The Avateq AVQ1050 ATSC 3.0 BPS Synchronizer Front End. This is a separate component that implements the functionality block on the left of Figure 2. This component was purpose-built by Avateq based on current technology they have deployed in other products.

③ The Avateq AVQ1050 ATSC 3.0 BPS Synchronizer Processor is a high-performance computing engine that allows further experimentation to be carried out without constraint. It allows external interfaces through PCIe to high-precision clocks or other devices that may prove useful as the project moves through additional phases in the future. The Synchronizer Processor is implemented by Avateq as the second component of the overall AVQ1050 ATSC 3.0 BPS Synchronizer.

④ Use of the dedicated processor enables many types of interfaces to reference clocks that were not available to the proof-of-concept device. Since this system continues to operate as an experimental platform for the BPS concept, the general-purpose CPU provides more precision clock interface options.

⑤ As with the proof-of-concept implementation, an adjustment signal needs to be generated by the receiver system and forwarded back to the studio equipment to "close the feedback loop," In this phase, a new interface was developed to the GatesAir Maxiva XTE exciter to allow the TIP adjustment to be as close as possible to the next emitted frame. The previous system provided this information to the Broadcast Gateway where there are at least three frames of buffering before emission and often many more due to other loop delays. A very simple interface to provide a short list of ± time offsets in nanoseconds representing the last few frame errors was developed in collaboration with GatesAir. This interface was also tested with Rohde & Schwarz SDE900/ TCE901 exciters.

⑥ The TIP adjustment is applied by the exciter on the next available frame after it receives the TIP adjustment information. This tightens the adjustment loop and directly controls the device generating the preamble data, namely, the exciter.

⑦ The Time Adjustment Information B is provided to the bps_info Generator function integrated with the Triveni Digital GuideBuilder system. This information is distinctly different from the information sent to the exciter in step ⑤ above. The information can be a more comprehensive set of data to allow the bps_info structure to carry history and other information helpful to receivers. In comparison, the Time Adjustment Information A is used as a direct control signal into the exciter.

⑧ The bps_info Generator uses the latest time differential information along with preset location values to create a bps_info data structure. This data structure is logged and sent to the Broadcast Gateway as a separate data table in a robust PLP. The PLP is a configuration element of the Broadcast Gateway. This function is integrated into GuideBuilder XM to allow the bps_info to be signed to avoid spoofing.
Note that it would also be possible to encrypt the bps_info structure but this would require a much more sophisticated public/private key infrastructure (PKI) when deployed.
For this phase, the bps_info structure was redesigned into three distinct sections to support different transmission requirements. The timing measurement was separated from the transmitter details and the timing source details to allow higher frequency, dynamic information to be distributed separately from more static data. The data structure is described in detail later in this paper.

⑨ The Broadcast Gateway can operate without knowledge that it is part of the BPS synchronization loop; however, a time adjustment is still available as a lower performance backup.

⑩ The bps_info log function is part of the Synchronizer Processor. It collects received bps_info structures into a log for post-processing.

⑪ A user interface (UI) is provided via a web interface (HTTP) to allow logging information to be displayed as well as to allow the system to be controlled. Control connections are not shown to reduce diagram complexity. Of particular interest will be a user interface that provides a graph of the time differential value over time to verify that the system is operating correctly. An example of this graphical display is shown in Figure 5.

Figure 2 provides a functional block diagram of the 2-chassis synchronizer of the BPS time optimization loop. The number callouts are represented in the list that follows the diagram.

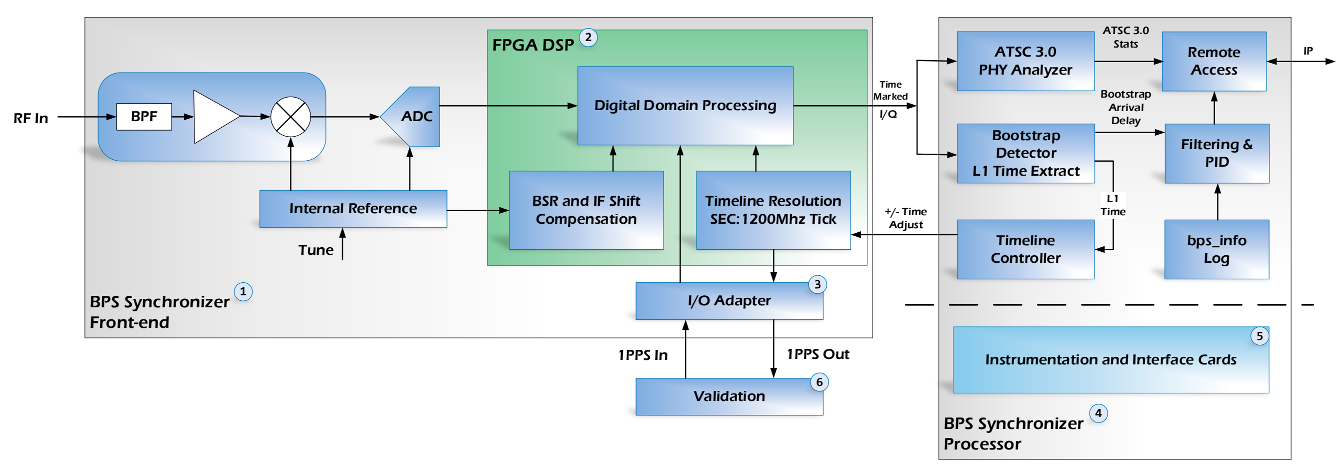

FIGURE 2: BPS RECEIVER / SYNCHRONIZER FUNCTIONAL BLOCK DIAGRAM.

① BPS Synchronizer Front-end combines several processing stages of the received RF signal analog and digital domains. The front-end allows an RF signal in the frequency range 100 to 1000 MHz and a power range of 0 to -60 dBm to be down converted, sampled, and processed in a complex DSP chain implemented in an FPGA fabric. The DSP chain outputs I/Q samples over a high-speed digital interface for further processing. The I/Q stream is "time-marked" with a counter in the SEC:120MHz tick format.

② FPGA DSP is the central processing core of the sampled signal. The digital domain processing is done under the control of the BSP Processor module. The Time-line Restoration block is built around a clocking core which keeps the time in SEC:120MHz format providing time values with a precision of 0.008 nanoseconds. The SEC portion of the clock is implemented as a 64-bit counter allowing counting UTC time in seconds. The Time-line Restoration block output is used for time-marking the I/Q stream in SEC:120MHz format. The block 1PPS output can be used by an external time validation instrument (e.g., an oscilloscope) for timing verification. Generated by an external source, a 1PPS input signal is used for the Front-end analog and digital processing chains calibration delay. The purpose of the calibration is to define a delay introduced by the chains. The known value of the delay is used for a correction of the Bootstrap arrival time.

③ I/O Adapter hardware performs signal conditioning (levels and logic adaptation, circuitry protection, etc.) to interface the internal components of the BPS Synchronizer Front-end with external equipment.

④ The time-marked and time-aligned I/Q stream is further processed in the BPS Synchronizer Processor block. The main processing stages include:

a) ATSC 3.0 PHY Layer Analyzer performs validation, restoration, and estimation of the signal framing components – Bootstrap, L1 signaling, PLPs including LLS information, if present. The Analyzer provides a set of data which can be used for the signal quality estimation and troubleshooting if any of the expected parameters are out of range.

b) Bootstrap Detector finds the first sample of the first symbol of each ATSC 3.0 frame and matches the sample number with the timeline of the I/Q stream. The TAI time extracted from the preamble L1 detail, and the time of Bootstrap arrival are used for adjusting clocks of the Time-line Restoration block performed by the Time-line Controller.

c) The Time-of-Arrival (TOA) estimation is based on correlation between reference and received Preamble patterns in time domain. The correlation produces a Channel Impulse Response (CIR). The CIR main peak corresponds to the earliest multipath component of the Preamble signal – the Preamble TOA. As the number of preceding Bootstrap samples is known, the whole frame TOA (Bootstrap Arrival Delay) can be calculated. The same procedure is applicable to SFN environment where each CIR peak is additionally marked with corresponding TxID.

   Note: Avateq introduced a so-called "SFN drift" estimation tools in 2015. The same tools were adapted to ATSC 3.0 in 2017. It allows the receive time of ATSC 3.0 frame to be verified by detecting any delays or shifts in Bootstrap TOA.

   As a result of performed experimental development, Avateq designed and implemented a proprietary algorithm for estimating the TOA of the earliest significant multipath component for ATSC 3.0 Physical Layer signals.

   The algorithm is based on the CIR deconvolution technique and provides accuracy of the TOA estimation in the range of a fraction of a nanosecond.

d) The Filtering & PID block processes the Bootstrap Arrival Delay information and converts the delay into TIP adjustments. The block functionality is the same as implemented in Phase 1 of the project with the "D" controller loop added.

e) The bps_info Log block collects and stores the received bps_info structures into a log for post-processing.

f) The BPS Receiver and Synchronizer run Avateq's proprietary framework. The framework integrates a library of ATSC 3.0 PHY Layer processing routines, a WEB-based GUI for remote access, internal data storing and logging. The remote interface provides full control of the device functionality. No special software is required on the Host side to communicate with the device.

⑤ The BPS Synchronizer provides a set of standard interfaces which can be used for additional instrumentation or diagnostic equipment.

⑥ The BPS Receiver provides matching functionality to view the 1PPS clock generated by the BPS synchronizer with an external system clock.

The details of the Avateq AVQ1050 BPS Synchronizer and Receiver are covered in the following tables.

Title	Reference
RF Input	50 Ohm, N-type Power Levels: -70 … +5 dBm (off-air) -50 … +5 dBm (Tx output) Freq Range: 100 … 1000 MHz Freq Tuning Step: 1kHz
10MHz	Input: 50 Ohm, BNC, 1Vp-p, sine Output: 50 Ohm, BNC, 5Vp-p, sine
1PPS	Input: 50 Ohm, BNC, TTL compatible Output: 50 Ohm, BNC, TTL compatible
IRIG-B126 output	High density DB26
GPS Receiver	Optional module
Ethernet	3x RJ45 10/100/1000
Form Factor	1U RF Front-end, and 2U Processing module
Operating temperature	0 … 50 C
Power Supply	110 – 250V, 50/60 Hz, AC

TABLE 1: TECHNICAL SPECIFICATIONS.

Title	Reference
Parameters	MER/SNR; MER for Bootstrap, L1, and selected PLP; Signal PAPR; Bandwidth; Freq and Sampling rate shifts; Shoulder attenuation; Emission/Spectral mask compliance; Groups Delay across bandwidth; ATSC 3.0 frame structure; Bootstrap, L1D, and L1B signaling info; LDPC BER for L1 and selected PLP.

Plots	Spectrum of the main lobe and in-band interference; Channel Amplitude/ Phase and Impulse responses; CCDF; Raw, Bootstrap, L1, and selected PLP constellations; Channel Impulse Response/ SFN/ Echo profile; Group delay.
Alarms	Input signal level; Spectrum shoulder levels; Signal MER; CIR/ Echo profile variations; Freq shift; etc.
Data logging and reporting	Detailed report with parameters and plots; Event and alarm log; Main parameters time interval log.
Software interfaces	WebGUI; SNMP v2c and v3; Email; MqTT
Additional tools	- Selected service streaming: • Reassembled DASH/ROUTE (HTTP) • ALP packets for sel PLP (UDP) - ATSC 3.0 STLTP – verification and detailed analysis - Network tools for connection verification and troubleshooting

TABLE 2: PHY LAYER ANALYSIS.

Title	Reference
Modes of operation	- BPS Synchronizer: Transmission path delay estimation and adjustment - BPS Receiver: Test, validation, and characterization of BPS deployment
Supported features	- Communication interface with Clock Reconciliation and Broadcast Studio block and PID based feedback loop to adjust ATSC 3.0 TIP - ATSC 3.0 Bootstrap Time of Arrival measurement and report to be included in bps_info - Extracting and decoding ATSC 3.0 L1 time info - Synchronization the receiver clock with ATSC 3.0 L1 time info and available as a reference to redistribute - Extracting, decoding, displaying, and logging bps_info
GPS-BPS hybrid functionality	- Additional development is required. - Scan and report BPS enabled signals presence - Estimate both position and time

TABLE 3: BPS FUNCTIONALITY.

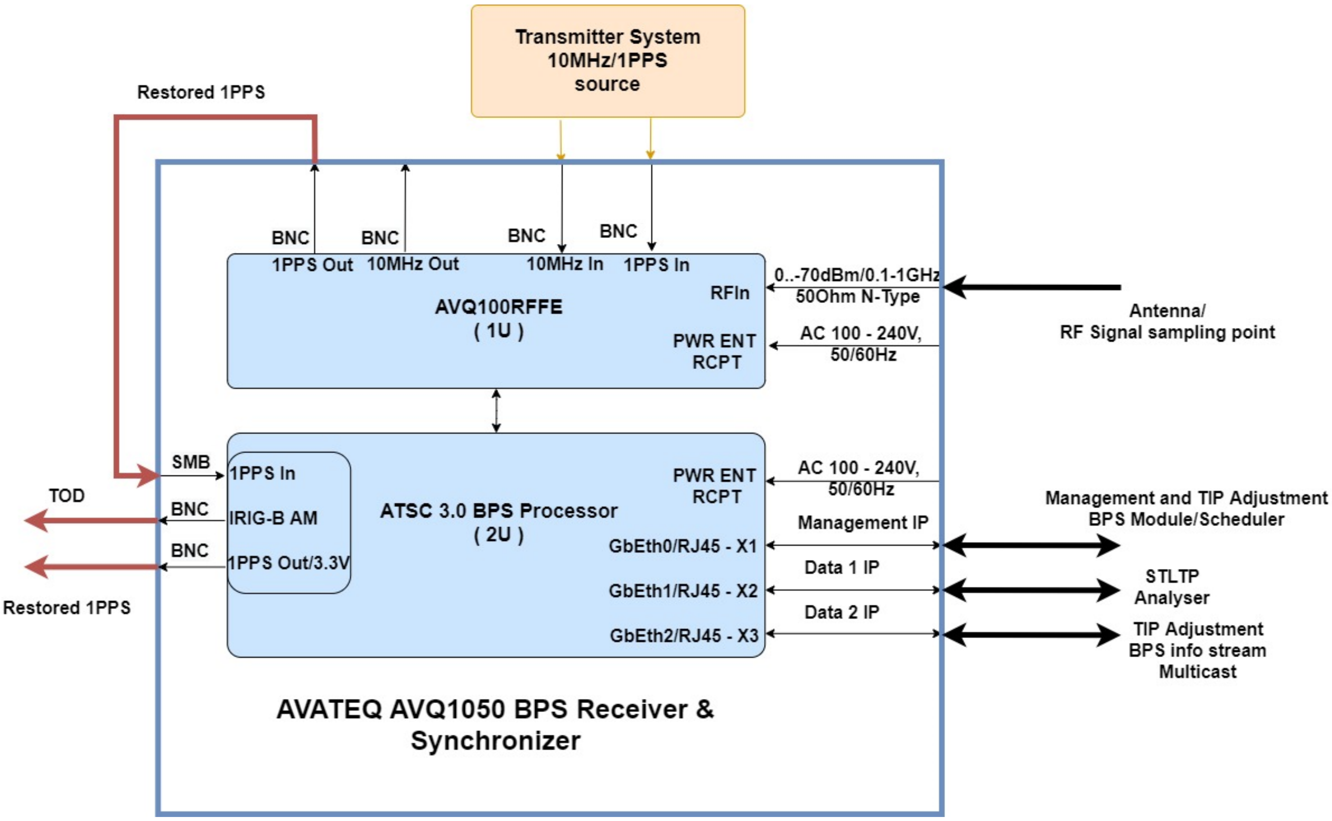

FIGURE 3: AVQ1050 ATSC 3.0 BPS RECEIVER AND SYNCHRONIZER FUNCTIONAL DIAGRAM.

System Performance

The proof-of-concept developed by the partner companies demonstrated that a feedback control loop can increase the accuracy of the transmitted timestamps of an existing ATSC 3.0 transmission chain. While the phase I proof-of-concept receiver synchronization provided an instantaneous time delay of the stabilized system at less than 300 ns the new system reduces that number to under 10 ns. We learned that the different exciters had different time response characteristics, meaning they had different transfer functions. We found that the optimal PI controller settings were different in each case. Our prototypes were stable for many days.

The phase II systems included the previous broadcast gateway software used in phase I but also introduced new firmware into the exciter to allow the synchronizer to control the emitted time directly at its insertion point. The higher performance of the synchronizer / receiver combined with the tighter control loop resulted in an order-of-magnitude improvement in system performance.

The solution demonstrated remains 100% compliant with the ATSC 3.0 standard as was the phase I system. No change in the standard was necessary for this phase of the BPS project. We have also demonstrated the solution with transmission equipment from multiple ATSC 3.0 vendors, reinforcing the standard compliance. Finally, we continue to learn how to make the system more accurate. We have identified the improvement opportunities and will include them in the next phase of development.

Synchronizer Performance

The yellow and blue curves exemplified in Figure 4 correspond to raw measurements of the Bootstrap arrival delay and the filtered and PID-processed TIP adjustment, respectively, provided by the phase I receiver. Due to the implementation of the receiver hardware, the time of arrival delay measurement resolution is limited to approximately 144 ns. However, the sophisticated processing of the raw measurements implemented in the receiver allows the TIP adjustment variations to be within approximately ±70 ns.

Figure 5 provides a view of the BPS Time Arrival Graph provided by the Phase II synchronizer. Note that the scale on the y axis of the second graph ranges between ± 0.004 µsec, that is, 4 ns, while the y axis in the graph in Figure 4 ranges between ± 0.28 µsec, i.e., 280 ns.

It should be noted that the blue line in the graph in Figure 5 shows the variation after PID filtering. This value remains consistently below 1 ns.

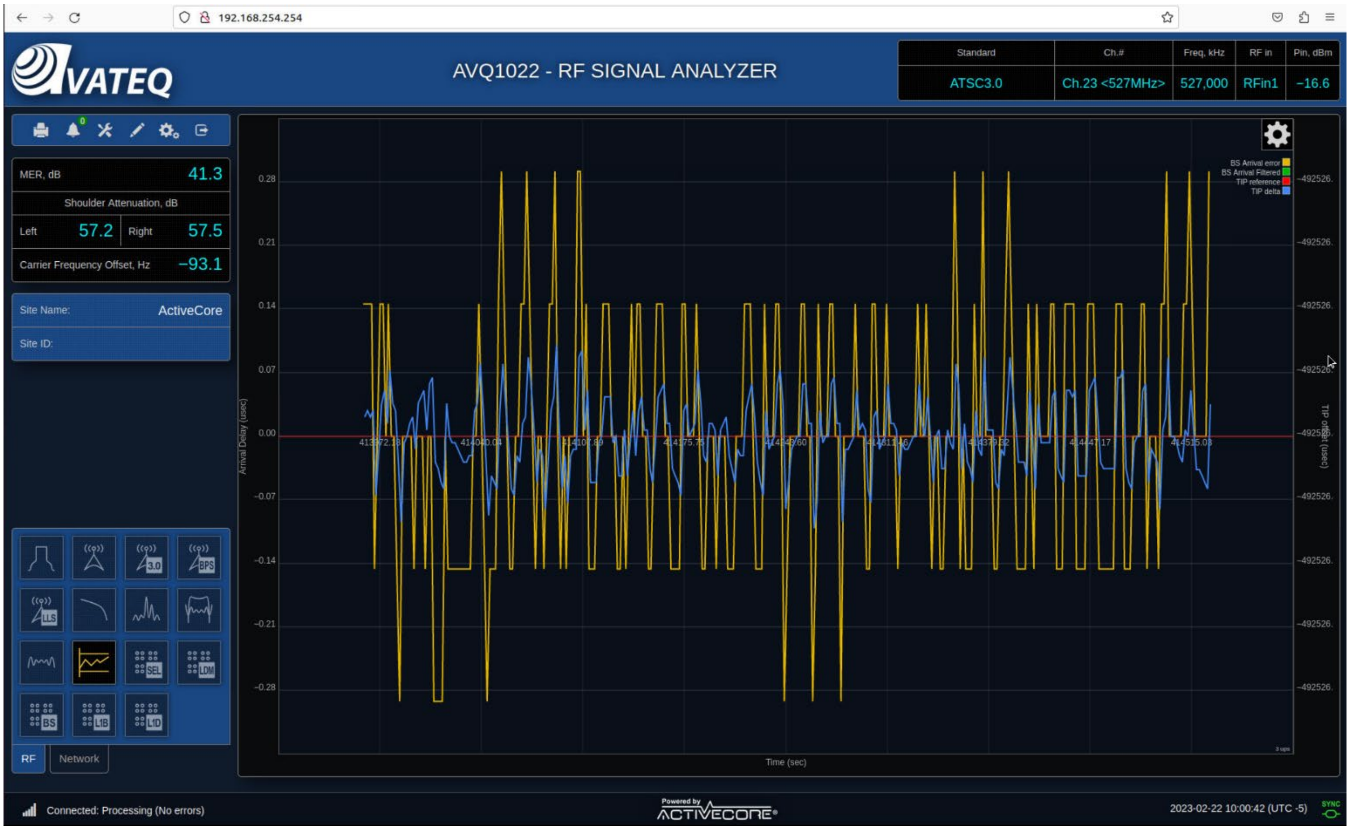

FIGURE 4: EXAMPLE PHASE I RECEIVER BPS TIME ARRIVAL GRAPH.

FIGURE 5: EXAMPLE PHASE II RECEIVER BPS TIME ARRIVAL GRAPH.

BPS Transmitter Description

In a nationwide deployment of the BPS network, it is expected that the functionality at each node, that is, each transmitter, would be essentially the same with some "leader" transmitters having time traceable to TAI. It is also expected that by adding additional time-keeping equipment, transmitters can move from "followers" to "leaders" and vice versa. This paper does not dwell on single frequency networks except to point out the difficulties of synchronizing multiple transmitters within an SFN. Also, providing unique measurement information for each transmitter in an SFN is still a topic of exploration. For the bulk of the document, readers can assume that all transmitters within receiving range are on separate frequencies, that is, form a multiple frequency network (MFN).

Figure 6 provides a diagram of an exemplary BPS network using four transmitters. The detailed diagram shown associated with one of the transmitters would apply to all the transmitter nodes in the BPS network. In this case, the node would be designated as a "Leader" since it would have another source of traceable time. Figure 6 provides a logical diagram of the systems and functions developed. The numbered paragraphs that follow the diagram correspond to the callouts in the diagram.

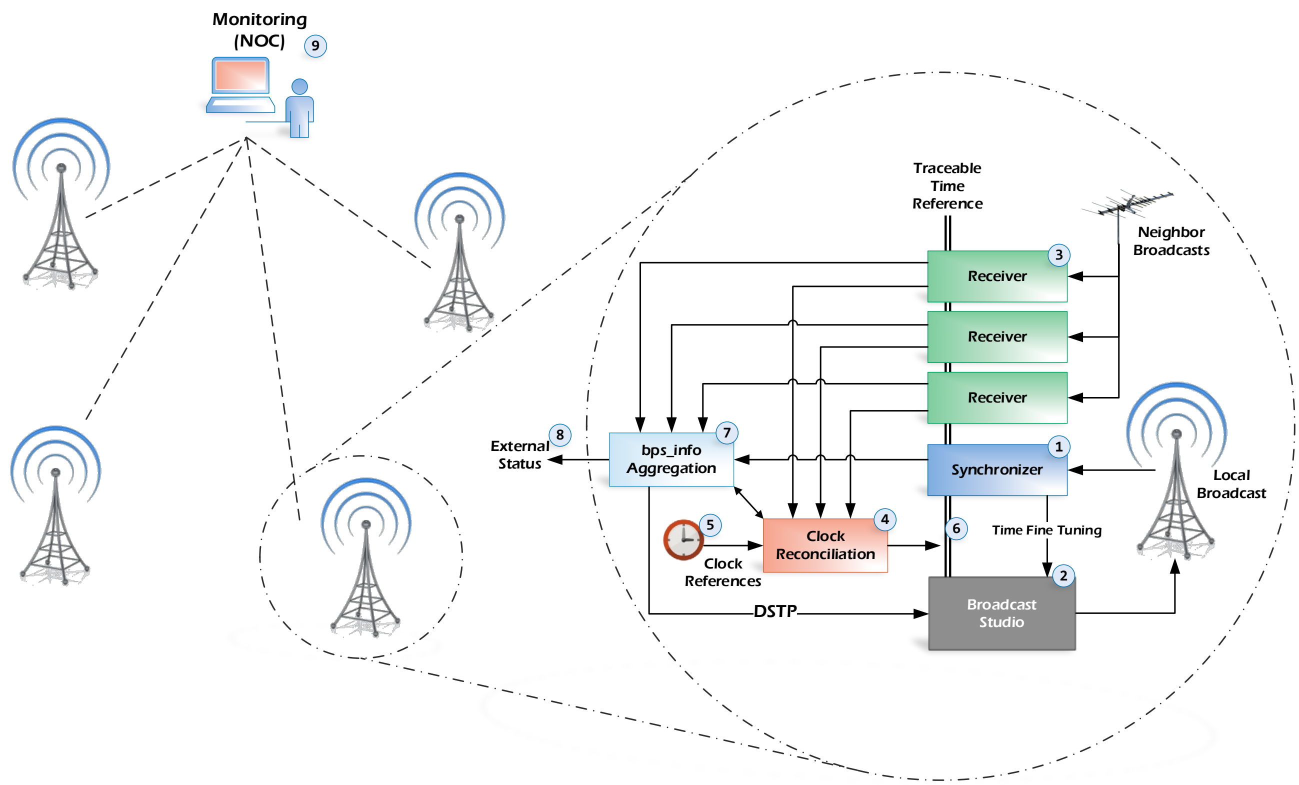

FIGURE 6: BPS TRANSMISSION NODE DIAGRAM.

① This block represents the Avateq AVQ1050 ATSC 3.0 BPS Synchronizer described in previous sections. This device receives the local broadcast, measures the bootstrap emission time with the time signaled in the L1D preamble and reports differences. It also extracts the bps_info data from the broadcast. The time difference is passed to the Broadcast Studio air chain ②, specifically, the exciter modified to accept the fine-tuning protocol provided by the synchronizer. Both GatesAir Maxiva XTE and Rohde & Schwarz SDE900/TCE901 exciters provide support for this protocol. All information about the basic processing of the time information is forwarded to the bps_info Aggregation ⑦. While it would be possible to send detailed clock information from the Synchronizer to the Clock Reconciliation function ④, this would likely not be useful since this information is self-referential and already known to the reconciliation function since it was the source of time for this "leader" node. Other "follower" nodes would not have an external source of traceable time and would rely on the leader for their time.

② The Broadcast Studio represents an ATSC 3.0 air chain. Internal components of the air chain of primary interest to this document are the Broadcast Gateway and the exciter. The bps_info Aggregation function ⑦ provides a multicast or DSTP stream containing bps_info that is formed into a dedicated PLP by the gateway. The output of the gateway is fed, via STLTP, to the exciter which generates the ATSC 3.0 wave form. In addition, the exciter receives fine-tuning information from the Synchronizer to make small adjustments to the emission time of each frame to more accurately generate the time carried in the L1D preamble of each frame.

③ The Avateq AVQ1050 ATSC 3.0 BPS Receiver and Synchronizer can also be used to obtain timing information from neighboring transmitters. In this case, the device is configured to operate as a "Receiver." For this example, three separate devices are required to receive the broadcast from three neighbor transmitters. The number of receivers deployed at any transmitter is limited only by cost since the number of transmissions visible to an ATSC 3.0 transmission tower are likely in the dozens once ATSC 3.0 is fully deployed. The robustness and resiliency of the network increases every time a receiver is added. Like the local synchronizer, these receivers will send all timing information including extracted bps_info to the bps_info Aggregation ⑦ function for processing. The clock signals out from the receiver are 1PPS and 10MHz reference signals, and date/time info in IRIG-B126 (IRIG-B AM). Support for HAVEQUICK and IRIG-B DCLS(LVCMOS levels) formats for the date/time info are optional and will be defined based on the trial results. This information is provided to the Clock Reconciliation function ④ for use in determining the actual best traceable time source.

④ The Clock Reconciliation function will be provided by a Safran SecureSync 2400. This function takes input from each receiver and one or more clock references using them as Input References. The SecureSync 2400 can be synchronized to different time and frequency sources that are referred to as Input References, or just References. References can be a GNSS receiver, or other sources delivered into the SecureSync 2400 unit via dedicated (mostly optional) inputs. It is also possible to enter a system time manually, which SecureSync 2400 can then synchronize to. For the SecureSync 2400 to declare synchronization, it needs both a valid 1PPS, and a Time reference.

⑤ The concept of Reference Priority allows the ranking of multiple references for redundancy purposes. This allows the SecureSync to gracefully fall back upon a lower ranking 1PPS or Time reference without transitioning into Holdover in case a reference becomes unavailable or invalid. The priority order assigned to the available references typically is a function of their accuracy and reliability. The different acceptable Input references can be found in Table 4.

Title	Reference
ASCII Timecode	ASCII serial timecode input
External 1PPS input	External 1PPS input (BPS)
Frequency	External Frequency input (BPS)
GNSS	GNSS input
PTP	PTP input
IRIG	IRIG timecode input (BPS)
Local System	Built-in clock OR internal 1PPS generation
NTP	NTP input
User	Host (time is manually set by the user)
HAVEQUICK	HAVEQUICK input

TABLE 4: SECURESYNC ACCEPTABLE INPUT REFERENCES

To get Input References into the SecureSync 2400 unit, the configurable connectors in the base unit or option cards can be used. The base unit includes the GNSS connector with the following characteristics:

- Connector: Type SMA with Type N cable adapter, +5V to power active antenna (+3.3V for optional SAASM)
- Frequency: GPS L1 (1575.42 MHz), Galileo E1 (1575.42 MHz), GLONASS L1 (1602.0 MHz), BeiDou B1 (1561.1 MHz), QZSS L1 (1575.42 MHz).

- Satellite tracking: 1 to 72, T-RAIM satellite error management
- Synchronization time: cold start < 15 minutes (includes almanac download), warm start < 5 minutes (assumes almanac downloaded)
- Antenna system: Separate from the SecureSync 2400, GNSS antennas (including anti-jamming) and a variety of accessories (surge protectors, splitter, cables…) are available to deploy the antenna system.

In addition to the GNSS connector, the multi I/O connector can be configured to accept 1PPS, IRIG, HaveQuick and ASCII Time Code inputs.

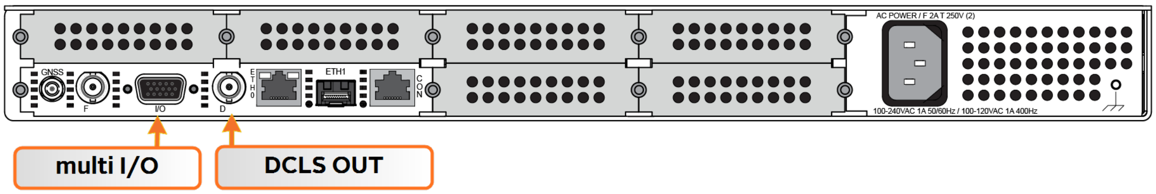

FIGURE 7: SECURESYNC 2400 BACK PANEL CONNECTOR DIAGRAM

It is notable that the SecureSync 2400 accepts up to six option cards that can be used for additional Input or Output references. In order to obtain compatibility with the IRIG - B126 (IRIG-B AM) signal coming from each Avateq AVQ 1050 ATSC 3.0 BPS Receiver and Synchronizer, an IRIG option card needs to be included in the SeucreSync 2400 unit.

The quality of input references can be assessed by comparing their phase offsets against the current system reference, and against each other. This is called Reference Monitoring.

Reference Monitoring helps to understand and predict system behavior and is an interference mitigation tool. It can also be used to manually re-organize reference priorities e.g., by assigning a lower reference priority to a noisy reference or a reference with a significant phase offset, or to automatically failover to a different reference if certain quality thresholds are no longer met. SecureSync 2400 allows Reference Monitoring by comparing the phase data of references against the System Ontime Point. The phase values shown are the filtered phase differences between each input reference 1PPS, and the internal disciplined 1PPS.

The data is plotted in a graph in real-time. The plot also allows historic data to be displayed and can zoom in on any data range or on a specific reference. A data set can be exported or deleted.

When comparing the multiple Input sources coming from the neighboring transmitters against traceable Input sources (i.e., GNSS or Time over Fiber systems), this graph will show the timing performance of the different elements in the system and will be able to provide traceability to UTC time. The active Input reference in the SecureSync 2400 will be used to synchronize all the equipment used in the broadcast and BPS systems. The SecureSync 2400 can export its monitoring data, including to the bps_info Aggregation ⑦, using the following mechanisms:

- IPv4/IPv6: Dual stack
- VLAN support
- DHCPv4/DHCPv6 (AUTOCONF)/SLAAC: Automatic IP address assignment
- Syslog: Logging
- HTTP(S): Browser-based configuration and monitoring
- REST API configuration and monitoring
- (S)FTP Server: Access to files (logs, etc.)
- SNMP: Supports v1, v2c, and v3 (no auth/auth/priv) with Enterprise MIB
- SMTP: Email

It also includes multiple security features to prevent cyberattacks during management and monitoring:

- Configurable Password Policy
- Authentication: LDAP, RADIUS, TACACS+
- Enable/Block protocols
- Access Control Lists
- HTTP Strict Transport Security (HSTS) support
- SSL/SSH
- TLS v1.2, v1.3
- SFTP/SCP: Securely transfers files to and from the time server over an SSH session
- SNMP v3: Remotely configure and manage over an encrypted connection
- Alert notifications via SNMP traps and email
- Signed software updates

The function receives information from the aggregation function the indicates which inputs are followers or leaders since this information is contained in the bps_info received from each transmitter. This information can be used to configure the Input priorities of the SecureSync 2400.

⑤ The clock icon represents one local clock reference and is backed up by the Input References that can be configured on a priority basis. Additionally, different Holdover oscillator options are available. The 1PPS and 10 MHz frequency output performance for these oscillators is provided in Table 5 and Table 6.

1 PPS Timing Output

	TCXO	OCXO	OCXO LPN	Rubidium	Rub LPN
Accuracy to UTC (locked to GPS, @ 1 sigma)	±50 ns	±25 ns	±15 ns	±15 ns	±15 ns
Holdover (constant temp after 2 weeks of GPS lock)					
After 4 hours	12 μs	1 μs	0.5 μs	0.2 μs	0.2 μs
After 24 hours	450 μs	25 μs	10 μs	1 μs	1 μs
Signal Waveform & Levels: TTL (5 V_{P-P}), into 50 ohm, BNC					

TABLE 5: 1PPS OUTPUT HOLDOVER OPTIONS.

10 MHz Frequency Output:

Internal Oscillator	TCXO	OCXO	OCXO LPN	Rubidium	Rub LPN
Accuracy (average over 24 hours when GPS locked)	5x10^{-12}	4x10^{-12}	3x10^{-12}	1x10^{-12}	1x10^{-12}
Medium Term Stability (without GPS after 2 weeks ofGPS lock)	1x10^{-8} / day	5x10^{-10} /day	2x10^{-10} / day	5x10^{-11} /month (3x10^{-11} /month typ)	
Short Term Stability (Allan Deviation)					
1 sec	2.5x10^{-9}	1x10^{-11}	1x10^{-11}	1x10^{-11}	1x10^{-11}
10 sec	1x10^{-9}	9x10^{-12}	9x10^{-12}	9x10^{-12}	1x10^{-11}
100 sec	5x10^{-10}	9x10^{-12}	8x10^{-12}	4x10^{-12}	5x10^{-12}
Temperature Stability (peak-to-peak)	1x10^{-6}	5x10^{-9}	2x10^{-9}	1x10^{-10}	1x10^{-10}
Phase Noise (dBc/Hz)					
@1 Hz	—	-95	-100	-80	-100
@10 Hz	—	-123	-128	-98	-128
@100 Hz	-110	-140	-148	-120	-148
@1 kHz	-135	-145	-150	-140	-150
@10 kHz	-140	-150	-150	-140	-150
Signal Waveform & Levels: +13 dBm Sine into 50 ohm, BNC					

TABLE 6: 10MHZ OUTPUT HOLDOVER OPTIONS.

The modularity of the SecureSync 2400 in terms of option cards, Holdover oscillators and power supply options plus the software flexibility to work with a variety of Input References allow this unit to be configured for both the leader and the following towers. This way, both ④ and ⑤ can be integrated within the same unit, avoiding an additional box to be managed in the system.

Thanks to the variety of Input References, alternative to GNSS technologies for timing such as atomic clocks, Time over Fiber (NTP, PTP or White Rabbit), LEO receivers or pseudolites can be optionally used to complement the resiliency and/or independence from GPS.

As an example of a traceable solution which would provide long-term holdover capabilities (up to several months) and could be used as a Time Reference for the SecureSync 2400, the combination of a GNSS receiver with a stable external clock like a maser has been proved as feasible backup solution, although not GNSS-independent. Taking advantage of the common view data provided by a National Metrology Institute to correct the data obtained from the GPS information, the UTC traceable time error related to the maser is obtained in the GNSS receiver by using the corrected GPS data (UTC) as a reference.

That reference can also be used to discipline the maser by applying the proper corrections on daily basis. The measurements correspond to the raw data measured for 80 days of operation. In this period, the maximum time error measured was lower than 280 ns. This provided a time drift average of 3.5 ns/day. This result confirms that the requirement of less than 1 µs of synchronization error in wide area networks during 100 days of operation without GNSS reference is possible when using high performance time sources.

⑥ The double line represents station synchronized time. This time is sourced from the Clock Reconciliation function ④ and used by all devices as a traceable time source. Note the clock information may be provided in many forms including:

- NTP v2, v3, v4: Conforms with RFC 1305 and 5905. Supports Unicast, Broadcast, Multicast, Symmetric Key Encryption, Peering, Stratum 2
- SNTP v3, v4: Conforms with RFC 1769, 2030, 4330, and 5905

- PTP v2: Conforms with IEEE 1588:2019. Supports Master, Slave, E2E, IPv4/v6, Multicast, Unicast, Hybrid modes
- PTP supported profiles: default, power (IEEE 61850-9-3, IEEE C37.238-2011/2017), telecom (ITU-T G.8265.1, G.8275.2 master only)

Additional profiles or protocols, including High Accuracy alternatives, and 1GbE/10 GbE support are available through Option Cards.

⑦ The bps_info Aggregation function is an extension of the bps_info generator developed for Phase I by Triveni Digital. It receives information regarding the bps_info from each receiver, the synchronizer and the clock reconciliation system and combines them into a single data stream. There are two types of data reported by the receivers: detailed time information, specifically how each transmitter is varying from the correct time, and static information about each transmitter location and type. Transmitter information is indexed via a unique transmitter ID (TXID), facility ID and frequency.
The combined bps_info stream along with the detailed timing information are merged into a single multicast IP stream fed into the Triveni Digital Broadcast Gateway. This forms the complete data source of a dedicated BPS PLP. The bps_info aggregation function generates new time information structures and possibly includes them in the signaling layer of the BPS PLP or simply as dedicated UDP/IP multicasts. This decision can be made once field trials and receivers are developed. Portions of this data may update as quickly as the fastest frame rate detected by the synchronizer and receivers. The static information, such as transmitter location, will be emitted less frequently, for example, every second or two. This will require very little bandwidth so the PLP can use a very robust modulation / coding (ModCod) setting. Specifics of this protocol are provided below.

⑧ In addition to providing bps_info streams to the broadcast gateway, the bps_info aggregation function ⑦ also provides status via an external network connection. This is intended to be rolled up into a central management system which would be used to monitor the entire BPS network operation ⑨. Each device in the diagram also provides a local, web-based command, control, and status user interface as is typically provided by each of these devices today which the NOC can use to configure and control each discrete device if desired if some aberration is detected.

⑨ Ultimately, the entire network will require one or more centralized "Network Operations Center(s)" (NOCs) to monitor the overall behavior of the BPS network. The NOC would identify outages and issues with the various devices across the operating area. There would likely be multiple NOCs separated geographically perhaps monitoring selected regions of BPS systems. Further, the NOC could manage the various devices in the BPS system by accessing their command-and-control websites. This would likely be done over a dedicated VPN but these details can be determined once the system becomes operational. For a live market trial, an example interface can be provided to demonstrate the basic NOC functionality.

BPS PLP Data Definition

Overview

This data structure definition described here divides the `bps_info` structure originally defined by NAB in [1] into multiple fragments and provides a rationale for doing so. The `bps_info` outer structure simply contains a CRC_32 checksum for the one or more `bps_info_fragments` contained within it. The fragments are defined as three different types: 1) a timing measurement fragment, 2) a timing source fragment and 3) a transmitter description fragment. This allows either a single fragment to be sent or multiple fragments to be combined into a single message reducing overhead. Fragments are independent of one another and are differentiated by transmitter ID, facility ID, and frequency. Fragments can be included in a bps_info structure in any way conducive to the local architecture. For example, measurement fragments and timing source fragments can be created by the receivers, synchronizer and clock reconciliation and sent independently of the station location information fragments. This flexibility allows all types of architecture to be realized in future deployments.

`bps_info` Data Structure Definition

Table 4 provides the syntax of the outer loop for the `bps_info` structure. The structure may contain as many fragments as desired up to the size of the fragments field but may constrain the total number of bytes used based on the underlying data carrier.

Syntax	No. of Bits	Format	Value
bps_info() {			
protocol_version	2	uimsbf	
fragments	6	uimsbf	
for (i=1; i<fragments; i++) {			
bps_info_fragment()	variable	Table 5	
}			
CRC_32	32	uimsbf	
}			

TABLE 7: `BPS_INFO()` DATA STRUCTURE.

protocol_version – The protocol version field defines the bps_info protocol version in use. Presently, this field shall be set to '1' indicating the first version of the BPS information protocol.

fragments – The fragments field provides the number of fragments contained in the `bps_info` structure. The `bps_info` structure shall contain a minimum of one fragment.

CRC_32 – This is a 32-bit field that contains the CRC value that gives a zero output of the calculation in the receiver as defined in ISO/IEC 13818-1 [5], Annex A after processing the entire `bps_info()` structure (including the **CRC_32** field).

In a future revision of this data structure, a signature can be added instead of a CRC_32 value to allow the entire data structure to be signed. Signing can be accomplished in the same way as other signaling in the ATSC 3.0 broadcast, for example, by using the same PKI and signaling support such as the CDT.

`bps_info_fragment` Data Structure Definition

Table 8 provides the syntax of the proposed `bps_info_fragment` data structure.

Syntax	No. of Bits	Format	Value
bps_info_fragment() {			
version	4	uimsbf	
fragment_length	12	uimsbf	
fragment_type	3	uimsbf	Table 1
TxID	13	uimsbf	
tx_freq	16	uimsbf	
facility_id	32	uimsbf	
if (fragment_type == Measurement) {			
forward_flag	1	boolean	

Syntax	No. of Bits	Format	Value
reserved	7	uimsbf	'0000000'
if (forward_flag == '1') {			
reported_bootstrap_time_sec	32	uimsbf	
reported_bootstrap_time_msec	10	uimsbf	
reported_bootstrap_time_usec	10	uimsbf	
reported_bootstrap_time_nsec	10	uimsbf	
reserved	2	uimsbf	'00'
bootstrap_toa_offset	32	simsbf	
}			
prev_bootstrap_time_sec	32	uimsbf	
prev_bootstrap_time_msec	10	uimsbf	
prev_bootstrap_time_usec	10	uimsbf	
prev_bootstrap_time_nsec	10	uimsbf	
reserved	2	uimsbf	'00'
prev_bootstrap_time_error_nsec	32	simsbf	
}			
else if (fragment_type == Timing_Source) {			
sync_hierarchy	8	uimsbf	
expected_accuracy	16	uimsbf	
timing_source_used	4	uimsbf	
num_timing_sources	4	uimsbf	
for (i=0; i<num_timing_sources; i++) {			
timing_source_type	4	uimsbf	Table 3
}			
if (num_timing_sources % 2 == 1) {			
padding	4		'1111'
}			
}			
else if (fragment_type == Description) {			
gain_flag	1	boolean	
position_flag	1	boolean	
power_flag	1	boolean	
pattern_flag	1	boolean	
reserved	2		'00'
if (gain_flag == 1) {			
max_gain_direction	10	Fixed point	
} else {			
reserved	10		0x3FF
}			
if (position_flag == 1) {			
geodetic_lat	64	double	
geodetic_lon	64	double	

Syntax	No. of Bits	Format	Value
geodetic_height	64	double	
}			
if (power_flag == 1) {			
radiated_power	32	float	
}			
if (pattern_flag == 1) {			
for (i=0; i<36; i++) {			
antenna_pattern_relative_field	7	uimsbf	
}			
reserved	4		'0000'
}			
}			
}			

TABLE 8: BPS_INFO_FRAGMENT() DATA STRUCTURE.

version – The version field provides the version of the `bps_info_fragment` structure. If the fragment structure changes in any way, the version shall be incremented modulus 16.

fragment_length – This **fragment_length** field provides the total number of bytes in the fragment after the length field. The total size in bytes of the fragment is thus **fragment_length** + 2 considering the size of the **version** and **fragment_length** fields.

fragment_type – This **fragment_type** field indicates the type of BPS information fragment that follows. Available types are defined in Table 9.

Type	Value
Measurement	0
Timing_Source	1
Description	2
reserved	3-7

TABLE 9: BPS_INFO_FRAGMENT() TYPES.

TxID – This field represents the `txid_address` field as it is defined in ATSC A/322 [3] with specific codes assigned according to the table at txid.nabpilot.org. The **TxID** indicates to which transmission the BPS information fragment applies.

tx_freq – This field indicates the middle of the assigned television channel frequency in units of MHz.

facility_id – This field provides the facility ID as recorded by the FCC. It is used to identify the particular transmitter. It is expected that this value will be provided as part of the configuration when a receiver or synchronizer is tuned to a particular frequency in a particular region.

The transmitter ID, transmission frequency and facility ID provide a unique tuple that identifies every transmitter in the northern hemisphere. Using these three values, the BPS management system can provide description information regarding that transmitter from its database.

Measurement Fragment Definition

The following fields are part of the measurement fragment which describes the results of measuring the actual time of the arriving signal with reported time as well as the previous time. The measurement fragment is used to provide information from the received transmitter (the source) and forwarded timing information from neighboring transmitters. The reported time of the received transmitter is not required in the `bps_info_fragment()` since the fields can be obtained directly from the L1D preamble values and calculated.

It is expected that measurement fragments will be inserted into the broadcast at the frame rate of the emitting transmitter. Neighbor transmitter measurement fragments will be forwarded at the rate they are received, if possible, that is, at the frame rate of the neighboring transmitter.

forward_flag – If the **forward_flag** is set to '1', then this fragment contains measurement information provided by a neighboring tower. The received source transmission is forwarding the information. If the flag value is '0', then only the previous bootstrap time is reported by the source transmission.

reported_bootstrap_time_sec – This field will be populated with the `L1D_time_sec` value, as defined in ATSC A/322 [3], of the most recent frame transmitted by a neighbor tower and received at the transmitting tower.

reported_bootstrap_time_msec – This field will be populated with the `L1D_time_msec` value, as defined in ATSC A/322 [3], of the most recent frame transmitted by a neighbor tower and received at the transmitting tower.

reported_bootstrap_time_usec – This field will be populated with the `L1D_time_usec` value, as defined in ATSC A/322 [3], of the most recent frame transmitted by a neighbor tower and received at the transmitting tower.

reported_bootstrap_time_nsec – This field will be populated with the `L1D_time_nsec` value, as defined in ATSC A/322 [3], of the most recent frame transmitted by a neighbor tower and received at the transmitting tower.

bootstrap_toa_offset – This field represents the difference between the time of arrival at the transmitting (self) antenna of the neighboring bootstrap signal and the reported timestamp in `L1D_time_sec`, `L1D_time_msec`, `L1D_time_usec`, and `L1D_time_nsec` fields as defined in ATSC A/322 [3]. The unit of this time offset is nanoseconds.

prev_bootstrap_time_sec – This field will be populated with the `L1D_time_sec` value, as defined in ATSC A/322 [3], of the immediately previous transmitted frame. When the **forward_flag** is '0', the transmitting tower will provide this information. When **forward_flag** is '1', the transmitting tower will receive the value in a `bps_info_fragment` from a neighboring tower and report that value in this field.

prev_bootstrap_time_msec – This field will be populated with the `L1D_time_msec` value, as defined in ATSC A/322 [3], of the immediately previous transmitted frame. When the **forward_flag** is '0', the transmitting tower will provide this information. When **forward_flag** is '1', the transmitting tower will receive the value in a `bps_info_fragment` from a neighboring tower and report that value in this field.

prev_bootstrap_time_usec – This field will be populated with the `L1D_time_usec` value, as defined in ATSC A/322 [3], of the immediately previous transmitted frame. When the **forward_flag** is '0', the transmitting tower will provide this information. When **forward_flag** is '1', the transmitting tower will receive the value in a `bps_info_fragment` from a neighboring tower and report that value in this field.

prev_bootstrap_time_nsec – This field will be populated with the `L1D_time_nsec` value, as defined in ATSC A/322 [3], of the immediately previous transmitted frame. When the **forward_flag** is '0', the transmitting tower will provide this information. When **forward_flag** is '1', the transmitting tower will receive the value in a `bps_info_fragment` from a neighboring tower and report that value in this field.

prev_bootstrap_time_error_nsec – This is the difference between the actual time when the first sample of the first symbol of the bootstrap was transmitted and the reported bootstrap transmission time in the `L1D_time_sec`, `L1D_time_msec`, `L1D_time_usec`, and `L1D_time_nsec` fields mentioned in ATSC A/322 [3]. The time difference is measured as actual transmission time minus reported transmission time in nanosecond units. When the **forward_flag** is '0', the transmitting tower will measure and provide this information. When the **forward_flag** is '1', the transmitting tower will receive the value in a `bps_info_fragment()` from a neighboring tower and report that value in this field.

Timing Source Fragment Definition

The timing source fragment describes how the particular transmitter obtained the timing signal it used when emitting frames. This information may change occasionally if the clock synchronization system detects a failure in one of its inputs. It may also be generated in systems separate from the main studio site making it more convenient to send just this timing source information separate from the transmitter description and measurement information.

The following fields comprise the Timing Source type `bps_info_fragment()`.

sync_hierarchy – This field indicates the number of hops needed to transfer time from a reference ATSC 3.0 transmitter to the transmitter that is using neighboring transmitter's ATSC 3.0 signal as a timing reference. If an ATSC 3.0 transmitter does not use the timing information from any neighboring transmitter to synchronize its own clock, its hierarchy is '0'. For example, transmitters that receive traceable time from NIST or USNO using satellite, fiber, or other methods but do not use any neighboring transmitter’s signal for synchronization will report 0 in this field. If a transmitter is using a **sync_hierarchy** = '0' transmitter’s signal for timing reference, that transmitter will report its **sync_hierarchy** as '1'. Generally, if a transmitter is listening to the signals from neighboring transmitters for clock synchronization, and if 'n' is the lowest **sync_hierarchy** among the received signals, that transmitter will report 'n+1' as its own **sync_hierarchy**.

expected_accuracy – This field indicates the accuracy of the TV station clock 99% of the time compared to UTC. The unit is nanoseconds. A value of 200 means that the TV station clock is synchronized in such a way that the local clock is within 200 ns of UTC 99% of the time.

timing_source_used – This field indicates the type of timing source used at the transmission facility. If the transmitter clock is derived from an ensemble of timing devices, a value of '15' will be reported. If the transmitter uses just one of the clocks available to it, that source will be reported using the table of the available **timing_source_type** values.

num_timing_sources – This field indicates the number of independent timing sources used at the ensemble clock deployed at the transmitter location for clock synchronization. For example, if a TV station uses GPS, one cesium clock, eLORAN, and 5 neighboring transmitters as input to the ensemble clock, that transmitter will report the number 8 in this field. If a transmitter does not use an ensemble clock but uses another backup clock if its primary clock fails, the reported value will be 1. The corresponding list of **timing_source_type** values will be updated accordingly in that case.

timing_source_type – The list of **timing_source_type** values provides the various timing sources used as reference at the ATSC 3.0 transmission facility. The types found in Table 10 are used to indicate the various clock sources used by the facility. If the transmission facility uses an ensemble of clocks, it will list the most accurate timing source of the ensemble as its source in the **timing_source_used** field.

Value	**Timing Source Type**
0	UTC from NIST or USNO securely delivered to the TV station
1	GPS
2	ATSC 3.0 signal emitted from neighboring transmitters
3	Local clock, such as cesium or rubidium
4	eLORAN
5-14	reserved
15	An ensemble of clocks is reported (**timing_source_used** only)

TABLE 10: TIMING SOURCE TYPES.

Description Fragment Definition

The information carried in the BPS Description fragment is expected to be relatively static since it describes the physical position of the transmitter and other configuration parameters that are not likely to change often. The frequency of transmission of the description fragment will determine the signal acquisition time, or the time it will take to lock on to a broadcast and start using the time signal as an accurate clock. Note that there are alternatives to providing this information only over broadcast for receivers that have broadband access. It is conceivable that a nationwide database could be made available, and the in-band broadcast could be used occasionally to limit bandwidth usage.

The following fields comprise the Description type `bps_info_fragment()`.

gain_flag – If the **gain_flag** is set to '1', then the **max_gain_direction** field shall be present in the information fragment. A value of '0' shall indicate that the 10 bits should be treated as reserved.

position_flag – If the **position_flag** is set to '1', then the **geodetic_lat**, **geodetic_lon** and **geodetic_height** fields shall be present in the information fragment. A value of '0' shall indicate that the three fields are not present. Since each field spans 8 bytes, there is no need for padding and no reserved fields are required when the fields are not present.

power_flag – If the **power_flag** is set to '1', then the **radiated_power** shall be present in the information fragment. A value of '0' shall indicate that the 32-bit value will not be present. Since the field is 32 bits long, no reserved padding is required when the field is not present.

pattern_flag – If the **pattern_flag** is set to '1', then the antenna pattern table with its 36 **antenna_pattern_relative_field** values shall be present in the information fragment. A value of '0' shall indicate that the table will not be present. Since the table occupies an integer number of bytes, no additional padding is required when the table is not present.

max_gain_direction – This field represents the direction where relative field strength is maximum, i.e., 1.0. The angle is measured clockwise from true north. The angle is reported as a fixed-point value such that 360 degrees is scaled to 1023 in fixed point notation.

geodetic_lat – This field represents geodetic latitude of the midpoint of the transmit antenna in WGS 84 LLA convention.

geodetic_lon – This field represents geodetic longitude of the midpoint of the transmit antenna in WGS 84 LLA convention.

geodetic_height – This field represents geodetic height of the midpoint of the transmit antenna in WGS 84 LLA convention.

radiated_power – This field represents ERP of the radiating antenna in units of kilowatts.

antenna_pattern_relative_field – The table containing these 36 values represents the antenna pattern by the relative field values reported every 10 degrees. The values are arranged clockwise starting at true north, meaning that the 1st value of the 36-element array represents the relative field strength in the direction north, the 2nd value represents the relative field strength in the direction 10 degrees clockwise from north, and so on. The relative field values are scaled such that the maximum value, which is 1.0, is scaled to the integer 127 in fixed point representation.

BPS Information Usage Discussion

It is expected that the `bps_info` data structures will be carried in one or more UDP/IP multicast streams contained in a single, dedicated PLP. It may be advantageous to designate one of the PLP numbers to BPS, for example, PLP 63. This would allow receivers to discover the BPS PLP without need for additional signaling. Within this PLP, every IP packet will contain a `bps_info` data structure with 1 or more `bps_info_fragments`. The IP address and port of the UDP/IP multicast can be any that meet the requirements of multicast addressing. It may be possible to move the Measurement and perhaps the Timing Source fragments into the ALP signaling level as user-defined signaling. It is not certain this is necessary. Clearly, there is no reason to carry Description fragments at that layer. If PLP 63 is used and is standardized as the BPS PLP, receivers could then simply process any UDP/IP packets found as `bps_info` packets.

By using a dedicated PLP, it is also possible to move this PLP to LDM making it possible to support SFNs. It might be useful in the case of an SFN to broadcast the Description fragment in the base stream and the Measurement and Timing Source fragments in the LDM PLP instead of trying to merge the PLPs somehow. If a single set of `bps_info` data is carried across all transmitters, reflecting every transmitter in the SFN network, then having separate `bps_info` fragments per transmitter is not necessary. It will depend on connectivity between the transmitters if combining all measurement and timing fragments would be conceivable.

Network Operating Center (NOC) System

The diagram in Figure 6 shows each transmission node communicating to a network operating center (NOC) ⑨. While each node could contain a reference time source from another network ⑤, in other words, every node could be a "Leader," a more economical solution would be to build a mesh of leader nodes and follower nodes where the followers rely on the leaders as a source of traceable time. The inclusion of a clock reconciliation device, namely the Safran SecureSync 2400 ④, at every node allows each follower to determine the best source of traceable time from its neighbors without necessarily including expensive local clock references.

Constructing the BPS mesh network in this fashion requires that a central system be used to monitor how well time is being distributed through the network and if each transmission node is transmitting time to a necessary accuracy. The `bps_info` aggregation function ⑦ is designed to obtain information from all receivers, the synchronizer and the clock reconciliation system and generate a summary of the performance of a transmission node. Since detailed information would not be useful for reporting to the NOC due to the volume of information, the aggregation function is expected to generate a summary of the information. For example, jitter maximums, drift, and other first derivative values are expected to be more indicative of the behavior of the node to the NOC software than the details themselves. This could be done at the NOC itself but will be more efficient if this processing is distributed to the edge, that is, the transmission node itself.

The summary information will be reported to the NOC system which will provide both comprehensive logging plus visualization of the summary information provided by the nodes. As the mesh grows larger, the amount of information accumulated will be large requiring pattern recognition and other techniques to assist operators in determining issues with the network. The NOC will also need alarm capability to notify operators that an issue is occurring at a particular node or set of nodes. Further, the NOC will have connectivity to each reporting node so can provide access to the user interfaces of all the devices within the transmission node for detailed troubleshooting purposes, if necessary.

It is expected that a single NOC will not be sufficient, and that the NOC software will be capable of being deployed in a hierarchy so that regions can be managed individually. Ultimately, it is expected that the NOC system software will evolve as the network evolves and change as necessary to support issues and outages that are not yet understood. It is also expected that the NOC software will be deployed on a cloud infrastructure and, therefore, could run and be managed from any location across the country.

Conclusion

In 2022, a proof-of-concept prototype was constructed to demonstrate that time stabilization of a BPS-enabled ATSC 3.0 transmission facility was achievable. We used commercial-grade exciters and other transmission chain equipment to prove compliance with the ATSC 3.0 standard. This resulted in a system that was functional but had limitations. In 2023, a new Receiver / Synchronizer system was purpose-built to improve the performance demonstrated by the proof-of-concept systems. This new prototype and the new connections developed tightening the feedback loop demonstrated marked improvement in jitter and accuracy over the previous prototype along with the ability to connect ATSC 3.0 broadcast systems to traceable time.

These results show that an independent mesh network of ATSC 3.0 television transmitters can be used to provide highly accurate, traceable time across multiple frequencies covering the continental United States. Further development of software to bind these transmitters into an operational network needs to continue but the technology described by this paper shows that it is eminently feasible at a modest cost.

References

[1] Corl, M., et al, "BPS ATSC 3.0 Broadcast Emission Time Stabilization System Proof-of-concept," Proceedings of the 2023 NAB Broadcast Engineering and Information Technology (BEIT) Conference.

[2] Diamond, P., et al, "Delivering Traceable Reference Time for ATSC 3.0-based Broadcast Positioning System (BPS)," Proceedings of the 2023 NAB Broadcast Engineering and Information Technology (BEIT) Conference.

[3] ATSC: "ATSC Standard: Physical Layer Protocol," Doc. A/322:2022-11, Advanced Television Systems Committee, Washington, DC, 14 November 2022.

[4] ATSC: "ATSC Standard: Signaling, Delivery, Synchronization, and Error Protection," Doc. A/331:2022-11, Advanced Television Systems Committee, Washington, DC, 30 November 2022.

[5] ITU: "Information Technology — Generic coding of moving pictures and associated audio – Part 1: systems," Doc. ITU-T Rec. H.222.0 | ISO/IEC 13818-1:2007, International Telecommunications Union, Geneva.