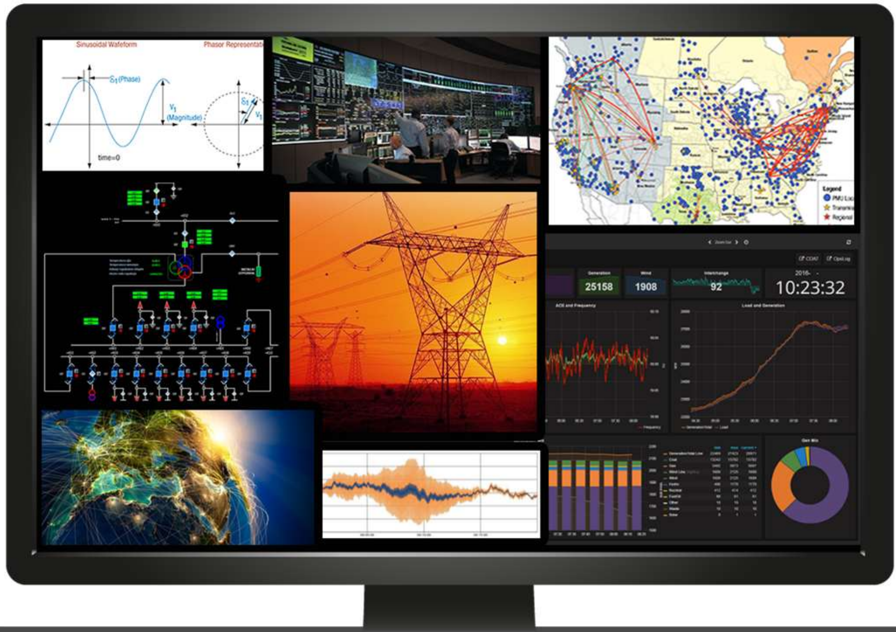
October 8th 2025



Using IEEE 2664 for Streaming Point on Wave Data

PoWA Conference

IEEE 2664-2024 (STTP)

Streaming Telemetry Transport Protocol

- US DOE Funded Project
- Intrinsically reduces losses and latency compared to frame-based protocols
- Allows the safe co-mingling of phasor data with other operational data network traffic
- Detailed metadata exchanged as part of protocol
- Includes lossless compression to reduce bandwidth utilization
- Security-first design with strong authentication and option for encryption
- Designed for Synchrophasor data

STTP Ideal for PoW Transmission, Continuous or Periodic

- Time-series special compression (TSCC) – *included in IEEE 2664 standard*
  - TSCC works well for synchrophasor and high-resolution time-series data
- Supports configurable high-resolution timestamps
  - Uses periodically updating base time offsets to reduce bandwidth
- Allows for both real-time streaming and historical playback
  - Subscriptions with time constraint can retrieve history at desired speed
- Supports standard and custom data types
  - User commands allow for notifications of triggers and events
- Includes extensible metadata
  - Standard XML tabular data sets can fully describe data available for subscription

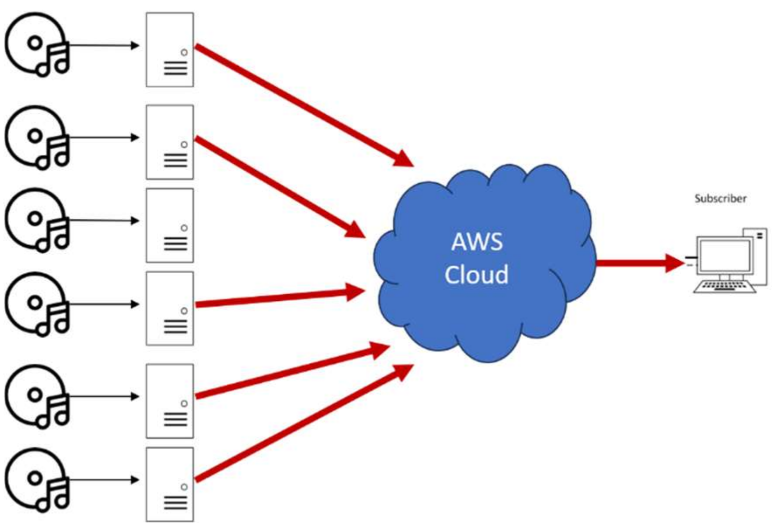
http://sttp.info

GRID PROTECTION ALLIANCE

© 2024 Grid Protection Alliance

STTP at Scale

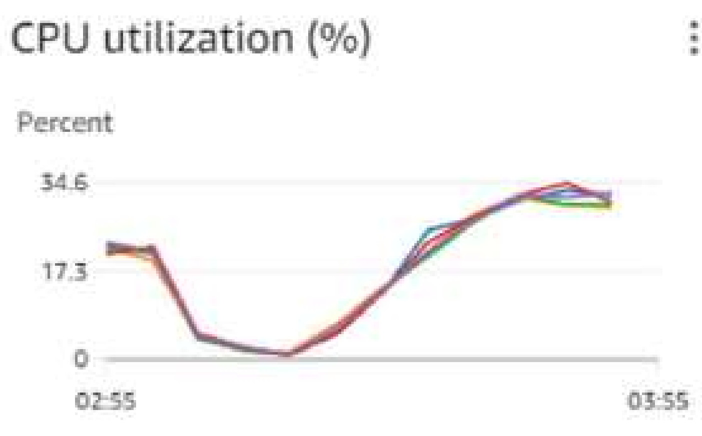
- Testing using real time time-synchronized data
  - Audio stream replay useful to demonstrate fidelity, i.e., not skipped or missed frames
- 2,812 points (two audio channels, left and right) at 44.1kHz / point
  - Corresponds to 1,406 audio sources, i.e., CD quality songs
  - ***Over 124 million samples per second***

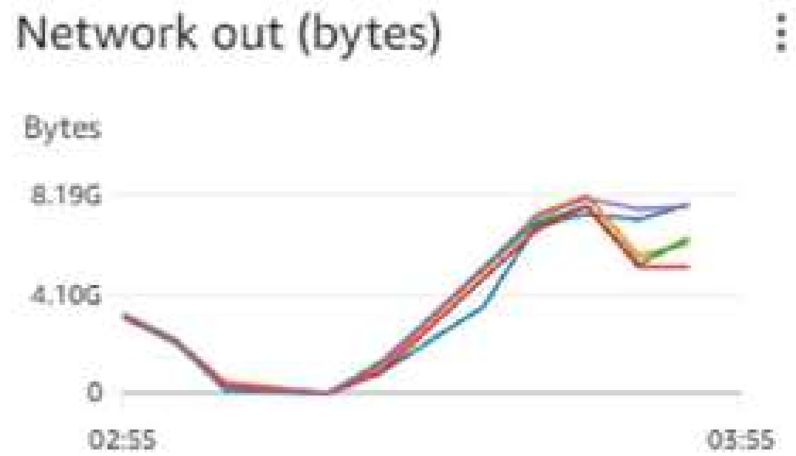


GRID PROTECTION ALLIANCE

© 2024 Grid Protection Alliance

STTP at Scale Test Results





- Average CPU usage < 35%
- Maximum Network Throughput < 28G bytes/s
- Compression rate > 93%

GRID PROTECTION ALLIANCE

© 2024 Grid Protection Alliance

TSSC Testing with Point on Wave

- Use of STTP TSSC for Point on Wave (PoW) data
  - we collected some POW sample data and ran tests
- Compression is very good for streaming phasor data
  - Low latency, low CPU impact, and fast
- Tests with streaming audio data also compressed well
  - Streaming signals at 44,100Hz data compressed well
- TSSC was expected to perform well with point on wave data…
  - Test data was recorded at 960Hz

It did not…

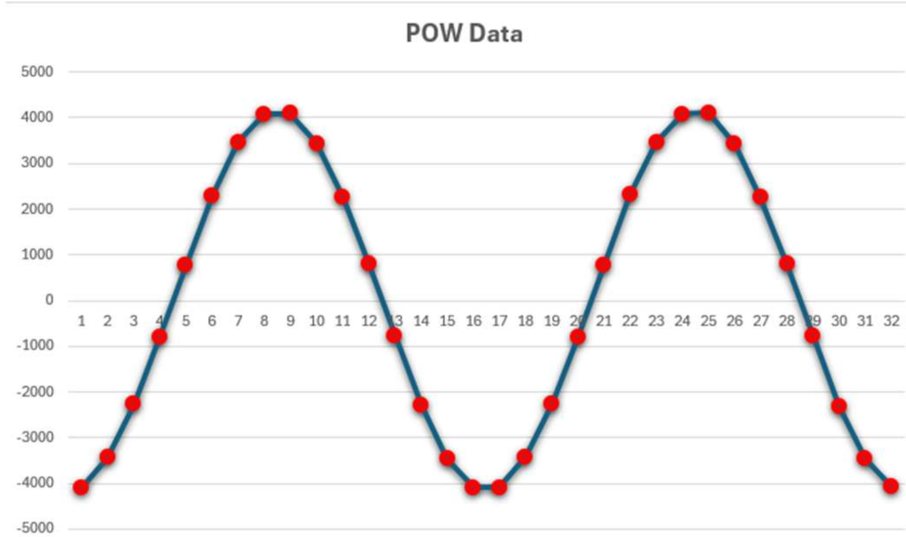
© 2024 Grid Protection Alliance

Why? Rate of Change

- TSSC performs well for data sets where there is a slow gradient of change:
  - This works well for phasor data (30/60Hz)
  - This works well for audio data (44100Hz)
  - ***Higher resolution compresses better!***
- What makes 960Hz special?

• Within 16 measurements, you move through 360 degrees →

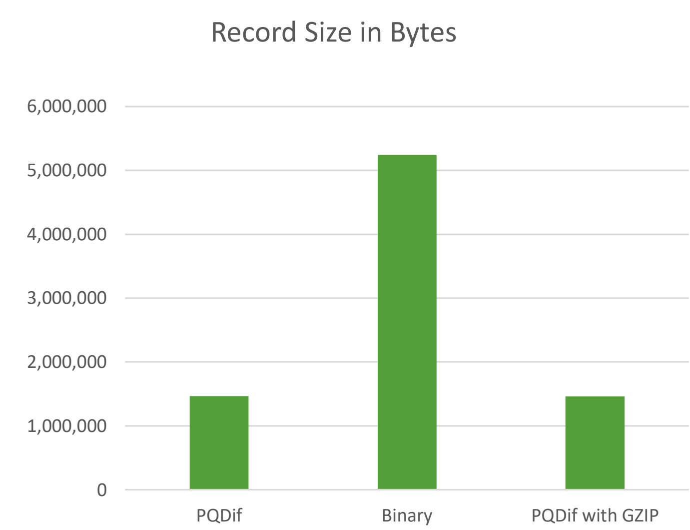
WARNING: Curves Ahead!



© 2024 Grid Protection Alliance

Compression Options for *Lower* Resolution PoW

- After playing around with many compression techniques, you find that standard compression algorithms work well.
- Of all the common players, LZMA (a.k.a. 7-zip) seemed to do the best job (empirically tested).
  - For a 5.6GB POW file representing a full day of data, 7-zip would reduce size by 65.6% (34.44% compression ratio)
- In terms of compression, we felt like this ratio could be improved, especially by understanding the sinusoidal nature of the data – something LZMA would not “assume”



© 2024 Grid Protection Alliance

Trying to match the curve…

- Started with a goal of trying to emulate the source curve as close as possible
- Tried lots of frequency estimators with simple sine wave:
  - Zero crossing / FFT / and just assuming fixed 60hz
- For several sample files, narrowed in on the following solution
  - Disclaimer: *This work was based on empirical work and intuition, more math could produce better results*
- Emulating the PoW curves with harmonic estimation, narrowing in on the 8th harmonic – simply because it produced the best match to original curve
  - For available data sources, anything higher or lower did not do as well

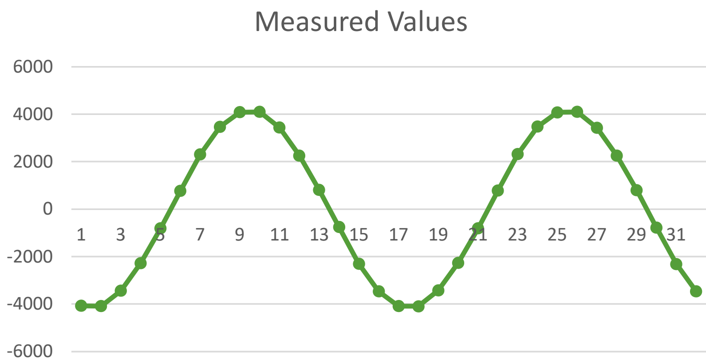
GRID PROTECTION ALLIANCE

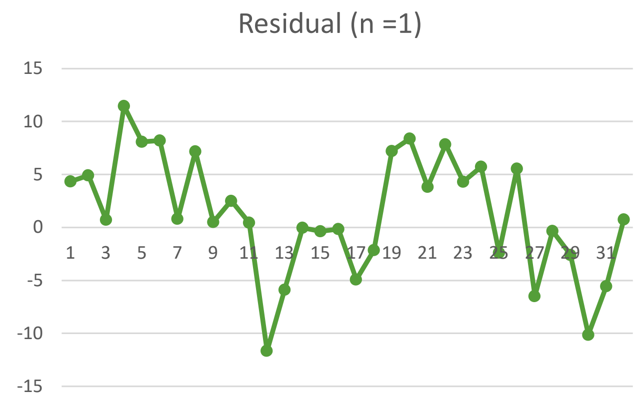
© 2024 Grid Protection Alliance

Trying to match the curve…

$$A * \sin(\omega t + \phi)$$

+





For available test data sets, 8th harmonic produced smallest residuals for 960hz

With many residual values being very tiny, e.g., -8 to +8, you can fit value into 4-bits with sign, so commonly two residual values in a single byte, making the 960Hz data much more compressible:

So, for two four-byte values, highest possible compression ratio becomes 8:1

GRID PROTECTION ALLIANCE

© 2024 Grid Protection Alliance

Everything is peaches and cream…

- Smaller compression ratios values are better:

Example 1:
Encoded Size: 39.82 megabytes / 158 megabytes (**25.17%**)
Supplemental Compression: 1,028 / 1,265 (81.26%)

Example 2:
Encoded Size: 39.47 megabytes / 158 megabytes (**24.95%**)
Supplemental Compression: 319 / 1,265 (25.22%)

Example 3:
Encoded Size: 36.25 megabytes / 158 megabytes (**22.91%**)
Supplemental Compression: 436 / 1,265 (34.47%)

Example 4:
Encoded Size: 40.01 megabytes / 158 megabytes (**25.29%**)
Supplemental Compression: 0 / 1,265 (0.00%)

Example 5:
Encoded Size: 40.21 megabytes / 158 megabytes (**25.42%**)
Supplemental Compression: 0 / 1,265 (0.00%)

Example 6:
Encoded Size: 40 megabytes / 158 megabytes (**25.29%**)
Supplemental Compression: 598 / 1,265 (47.27%)

Example 7:
Encoded Size: 39.76 megabytes / 158 megabytes (**25.13%**)
Supplemental Compression: 518 / 1,265 (40.95%)

Example 8:
Encoded Size: 39.58 megabytes / 158 megabytes (**25.02%**)
Supplemental Compression: 0 / 1,265 (0.00%)

Example 9:
Encoded Size: 40.36 megabytes / 158 megabytes (**25.51%**)
Supplemental Compression: 0 / 1,265 (0.00%)

Example 10:
Encoded Size: 40.13 megabytes / 158 megabytes (**25.37%**)
Supplemental Compression: 0 / 1,265 (0.00%)

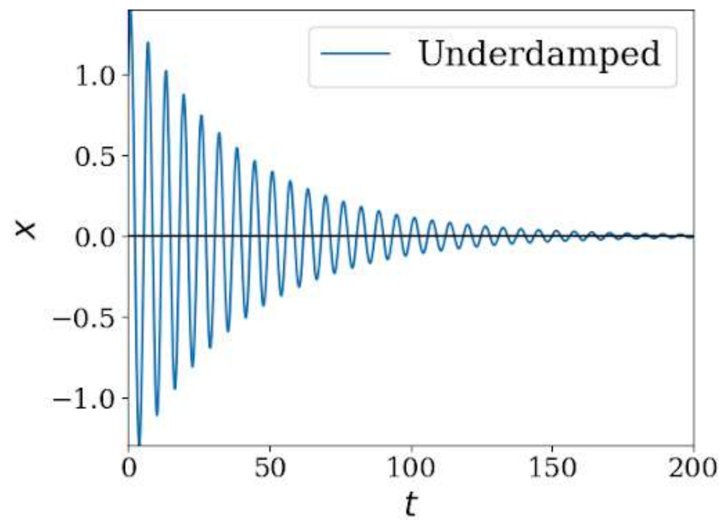
Note: results excluded for tests cases that used around 100% supplemental compression

© 2024 Grid Protection Alliance

Everything is peaches and cream…

Until it's not…

- Pretty, predictable curves aren't always so pretty
- Sometimes they get angry and noisy
- So, you need a "plan B" for compression in these cases
  - As previously tested, LZMA is a good "general choice" for compression
- When things don't compress well, e.g., less than a target of 26%, use a common compression algorithm, e.g., LZMA



© 2024 Grid Protection Alliance

Conclusion

- For a sinusoidal inputs, results were better than LZMA alone
- For wave forms that didn’t “fit”, LZMA produced better results
- The current implementation operates by using both, again, when ratio is less than (configurable) 26%, use LZMA

- ***Pros***:
  - Good compression, ~25%
  - Suitable for streaming compression, e.g., STTP
  - Reduces bandwidth for streaming and file transfers in reduced bandwidth environments, e.g., sub-station

- ***Cons***:
  - CPU costs are high – lots of calculation required – so better suited for single device streams, i.e., fewer signals
  - More compression would be better; more work to be done on improving algorithm results

© 2024 Grid Protection Alliance

IEEE 2664-2024 (STTP)

sttp IEEE 2664

Streaming Telemetry Transport Protocol

http://sttp.info