Performance and Applications of Synchronized Waveform Data Compression

Steven Blair
Jason Costello

April 2022
Agenda

1. Industry challenges
2. Slipstream: data compression
3. Performance and comparison
4. Applications
Industry challenges

**Power grids**
- Reduced system inertia
- Increasing levels of IBRs
- Extremes of weather/wildfires

**Monitoring technologies**
- Phasor-based monitoring has limits
- Data comms and analysis can be inefficient
- Multiple competing standards
Digitising waveform measurements
Digitising waveform measurements

The data value (e.g., it's 40 volts)

Did something go wrong?

When was it

total: 16 bytes
Digitising waveform measurements

up to 30 passive voltage and current sensors per fibre
(equivalent to 5-10 Merging Units)

total: 480 bytes, every sample
Digitising waveform measurements

up to 30 passive voltage and current sensors per fibre
(equivalent to 5-10 Merging Units)

do this every 69-250 microseconds

total: 6.9 megabytes every second
(55.2 Mbps)
The data problem

4 kHz sampling:

- 1.9 megabytes every second
- 7 gigabytes every hour
- 61 terabytes every year

14.4 kHz sampling:

- 6.9 megabytes every second
- 25 gigabytes every hour
- 218 terabytes every year

14.4 kHz is equivalent to:

- 4K high-quality video stream
- ~46 Microsoft Teams video streams
Synchronized waveform data compression

Objectives:

• Designed for **streaming waveform data**, similar to IEC 61850-9-2 or IEC 61869-9 SV
• Optimised for **smallest message size**
• **Low overhead** compared to SV
• **Lossless**: must not add errors or distortion
• **Flexible**: variable number of samples per message for different applications
• Compress each data stream separately
• Assume out-of-band comms will arrange meta information (like IEEE C37.118.2)

IEC 61850-9-2 SV payload format

(~219 bytes for 2 samples)
Other compression principles

- Preserve time sync and data quality information
- Produce a byte stream for any transport method (e.g. UDP, TCP, Ethernet, HTTP, WebSocket, raw data file)
- Prefer efficient and fast encoding/decoding methods (simple linear arithmetic)
- Error or loss of a message must not affect other messages
- Open source project: [https://github.com/synaptecltd/slipstream](https://github.com/synaptecltd/slipstream)
Slipstream compression approach

1. delta-delta (multi-layer) with zig-zag encoding
2. simple-8b encoding
3. Optional gzip compression (for large data captures only)
4. Run-length encoding of data quality values

Header
- UUID (16 bytes)
- Starting timestamp (8 bytes)
- Number of samples encoded (max. 4 bytes)

First sample
- Stream 1 (4 bytes)
- Stream 2 (4 bytes)
- Stream n (4 bytes)

Remaining samples (delta-delta encoded)
- Stream 1 delta (~1 byte)
- Stream 2 delta (~1 byte)
- Stream n delta (~1 byte)
- ... (~1 byte)
- ... (~1 byte)
- ... (~1 byte)

Quality (RLE)
- Stream 1 (>=2 bytes)
- Stream 2 (>=2 bytes)
- Stream n (>=2 bytes)
Delta-delta encoding

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>Signal</th>
<th>delta</th>
<th>delta-delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00025</td>
<td>1554</td>
<td>1554</td>
<td>1554</td>
</tr>
<tr>
<td>0.00050</td>
<td>3140</td>
<td>1586</td>
<td>32</td>
</tr>
<tr>
<td>0.00075</td>
<td>4664</td>
<td>1524</td>
<td>-62</td>
</tr>
<tr>
<td>0.00100</td>
<td>6193</td>
<td>1529</td>
<td>5</td>
</tr>
<tr>
<td>0.00125</td>
<td>7653</td>
<td>1460</td>
<td>-69</td>
</tr>
<tr>
<td>0.00150</td>
<td>9086</td>
<td>1433</td>
<td>-27</td>
</tr>
<tr>
<td>0.00175</td>
<td>10454</td>
<td>1368</td>
<td>-65</td>
</tr>
<tr>
<td>0.00200</td>
<td>11763</td>
<td>1309</td>
<td>-59</td>
</tr>
<tr>
<td>0.00225</td>
<td>12991</td>
<td>1228</td>
<td>-80</td>
</tr>
<tr>
<td>0.00250</td>
<td>14151</td>
<td>1160</td>
<td>-68</td>
</tr>
<tr>
<td>0.00275</td>
<td>15209</td>
<td>1058</td>
<td>-103</td>
</tr>
<tr>
<td>0.00300</td>
<td>16169</td>
<td>960</td>
<td>-98</td>
</tr>
<tr>
<td>0.00325</td>
<td>17054</td>
<td>885</td>
<td>-74</td>
</tr>
<tr>
<td>0.00350</td>
<td>17821</td>
<td>767</td>
<td>-119</td>
</tr>
<tr>
<td>0.00375</td>
<td>18462</td>
<td>641</td>
<td>-126</td>
</tr>
</tbody>
</table>
## Compression performance

<table>
<thead>
<tr>
<th>Sampling rate (Hz)</th>
<th>Samples per message</th>
<th>Message size (bytes)</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>10</td>
<td>210</td>
<td>16.4%</td>
</tr>
<tr>
<td>4000</td>
<td>4000</td>
<td>40778</td>
<td>8%</td>
</tr>
<tr>
<td>14400</td>
<td>6</td>
<td>123</td>
<td>16.9%</td>
</tr>
<tr>
<td>14400</td>
<td>14400</td>
<td>2812</td>
<td>0.2%</td>
</tr>
<tr>
<td>150000</td>
<td>150000</td>
<td>7238</td>
<td>&lt;0.1%</td>
</tr>
</tbody>
</table>

Previous work only achieved ~54% ratio


Slipstream: 0.67 Mbps
IEC 61850-9-2 SV: ~20 Mbps

Assuming a dataset of three-phase voltage and current signals, at 400 kV and 500 A.
Compression performance (with noise and harmonics)

<table>
<thead>
<tr>
<th>Sampling rate (Hz)</th>
<th>Samples per message</th>
<th>Message size (bytes)</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000</td>
<td>10</td>
<td>236</td>
<td>18.4%</td>
</tr>
<tr>
<td>4000</td>
<td>4000</td>
<td>123738</td>
<td>12.1%</td>
</tr>
<tr>
<td>14400</td>
<td>6</td>
<td>141</td>
<td>18.3%</td>
</tr>
<tr>
<td>14400</td>
<td>14400</td>
<td>123213</td>
<td>6.7%</td>
</tr>
<tr>
<td>150000</td>
<td>150000</td>
<td>779918</td>
<td>4.1%</td>
</tr>
</tbody>
</table>
Demo: Ten 14.4 kHz streams over WSS in <2 Mbps
### Compression is slow?

<table>
<thead>
<tr>
<th>Data storage method</th>
<th>Sampling rate (Hz)</th>
<th>Samples per message</th>
<th>Message size</th>
<th>Size</th>
<th>Time to encode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw data</td>
<td>14400</td>
<td>144000</td>
<td>10.4 MB</td>
<td>56.3%</td>
<td>47 ms</td>
</tr>
<tr>
<td>Slipstream</td>
<td>14400</td>
<td>144000</td>
<td>0.8 MB</td>
<td>4.4%</td>
<td>37 ms</td>
</tr>
<tr>
<td>CSV</td>
<td>14400</td>
<td>144000</td>
<td>12.6 MB</td>
<td>68.1%</td>
<td>379 ms</td>
</tr>
<tr>
<td>CSV (+gzip)</td>
<td>14400</td>
<td>144000</td>
<td>4.2 MB</td>
<td>22.5%</td>
<td>527 ms</td>
</tr>
</tbody>
</table>

Compressing is faster than not compressing!
## Comparison with industry standards

<table>
<thead>
<tr>
<th></th>
<th>Slipstream</th>
<th>IEC 61850-9-2 Sampled Values</th>
<th>IEEE C37.118.2</th>
<th>IEEE P2664 (STTP)</th>
<th>PQDIF</th>
<th>COMTRADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time or storage?</td>
<td>Both</td>
<td>Real-time</td>
<td>Real-time</td>
<td>Real-time</td>
<td>Storage</td>
<td>Storage</td>
</tr>
<tr>
<td>Suitable for waveform data?</td>
<td>Yes</td>
<td>Yes</td>
<td>Can be adapted, but not optimal</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Payload definition</td>
<td>Fixed datasets</td>
<td>Fixed datasets</td>
<td>Fixed datasets</td>
<td>Dynamic subscription of data points</td>
<td>Arbitrary</td>
<td>Arbitrary</td>
</tr>
<tr>
<td>Sampling rate</td>
<td>Any, fixed per dataset</td>
<td>Any, fixed per dataset</td>
<td>Certain reporting rates expected</td>
<td>Any per data point</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>Open source?</td>
<td>Yes</td>
<td>No, but some implementations are</td>
<td>No, but some implementations are</td>
<td>No, but some implementations are</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Security</td>
<td>Handled by transport layer</td>
<td>Auth + encryption possible</td>
<td>Handled by transport layer</td>
<td>Auth + encryption possible</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Bandwidth/storage use</td>
<td>Low</td>
<td>Very high</td>
<td>Medium/high</td>
<td>Low/medium</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>
Future improvements

- Optimisation of internal parameters for best compression performance
- Clearly define timestamp and quality fields
- Expand supported data types
- Formalise metadata format and exchange
Applications

New protection methods based on waveforms, not just phasors

Continuous monitoring of harmonics and other PQ metrics

Analysis of transients – condition monitoring of primary assets

Automated event classification and location

Q: What is the source location of this momentary arc / fault?


Tianshu Bi, NCEPU, IEEE SGSMA 2021

https://doi.org/10.1109/PSCE.2004.1397508

https://www.naspi.org/sites/default/files/2021-04/D1S1_02_wang_dominion_naspi_20210413.pdf
Contact info

Dr Steven Blair
Head of Power Systems Technologies

steven.blair@synapt.ec

https://github.com/synaptecltd/slipstream