The new IEEE Standard for the Streaming Telemetry Transport Protocol (STTP)
Agenda

Technical Intro to STTP (Ritchie Carroll, GPA)
STTP as IEEE standard (Ken Martin, EPG)
STTP Security Features (Scott Mix, PNNL)
STTP at Dominion (Kevin Jones, Dominion)
STTP at SDG&E (Jared Bestebreur, SEL)
STTP at SPP (Mike Nugent, SPP)
STTP at ComED (Mike Brown, PingThings)
Humble Beginnings

NASPI Meeting

J. Ritchie Carroll

April 5, 2023

IEEE P2664
Original Streaming Data Flows:
- All configured data broadcast from point A to point B

Common Issue:
- Bandwidth / processing overload

Information Needs:
- Commonly a visualization or computation only needs certain data

Idea for Solution:
- Find a way to only “subscribe” to desired data

Problem -- No Protocol Did This:
- Invent one!

How Did We Get Here?
NASPI Meeting – April 2023
The “invented” protocol started as the “Gateway Exchange Protocol” (GEP)

Early on, many utilities invested time, energy and funding into helping flesh out the needs for this new Pub/Sub protocol – here’s a few:

- Entergy / Dominion / TVA
- SDG&E / Peak RC / SOCO
- SPP / OG&E / MISO / PJM

Thankless and unacknowledged countless hours went into this initial testing and development – the industry owes the existence of this new protocol to these key players!
US DOE Project
- Helped fund effort to develop and standardize STTP

Industry Recognized Need
- Many partners assisted >>

Advanced Synchrophasor Protocol Project
STTP
Streaming Telemetry Transport Protocol

NASPI Meeting – April 2023
Atomic Measurement Packets
- Reduced Data Loss
- Lossless Compression
- Scalability (to hardware limits)

- Publish / Subscribe Model
- Publisher Data Access Control
- IP Level Security
- Configurable Connection Origin
### API Status

<table>
<thead>
<tr>
<th></th>
<th>Subscriber</th>
<th>TSCC</th>
<th>Filter Expressions</th>
<th>Reverse Subscriber</th>
<th>Publisher</th>
<th>Reverse Publisher</th>
<th>TLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSF</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>C++</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Go</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Python</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rust</td>
<td>✔️</td>
<td></td>
<td>Ongoing progress on STTP API language targets...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- All API language targets expected to be function complete by end of year in time for IEEE release
- However, no need to wait, **start integrating now!**
Chose one and go!

Python STTP Implementation
https://github.com/sttp/pyapi

Go STTP Implementation
https://github.com/sttp/goapi

.NET STTP Implementation
https://github.com/sttp/dotnetapi

C++ STTP Implementation
https://github.com/sttp/cppapi

STTP Connection Tester
https://github.com/sttp/connection-tester

Open Source
All STTP reference implementations are Open Source Software (OSS) published on GitHub under the permissive MIT license.