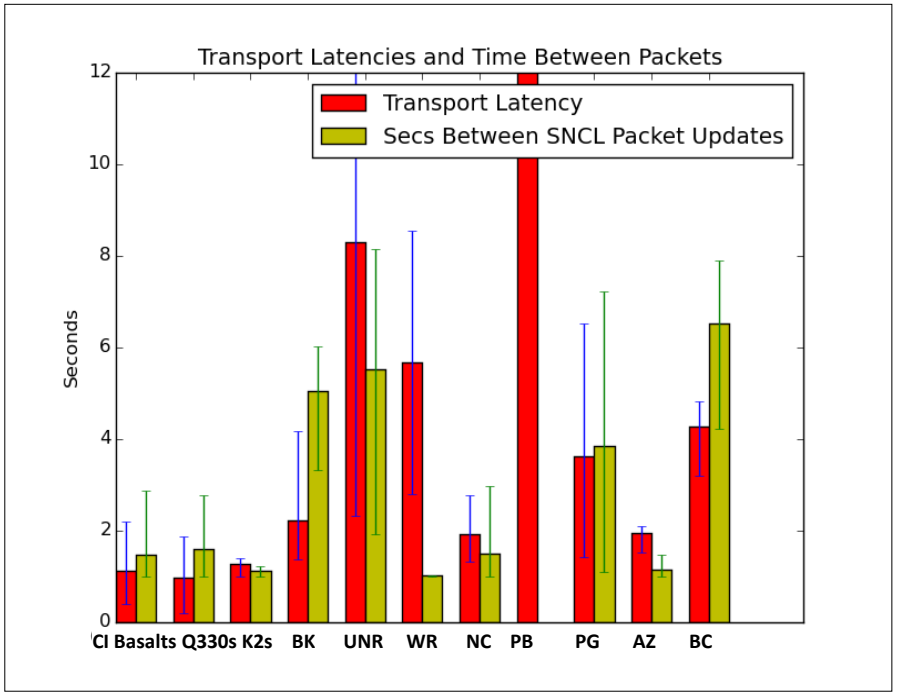
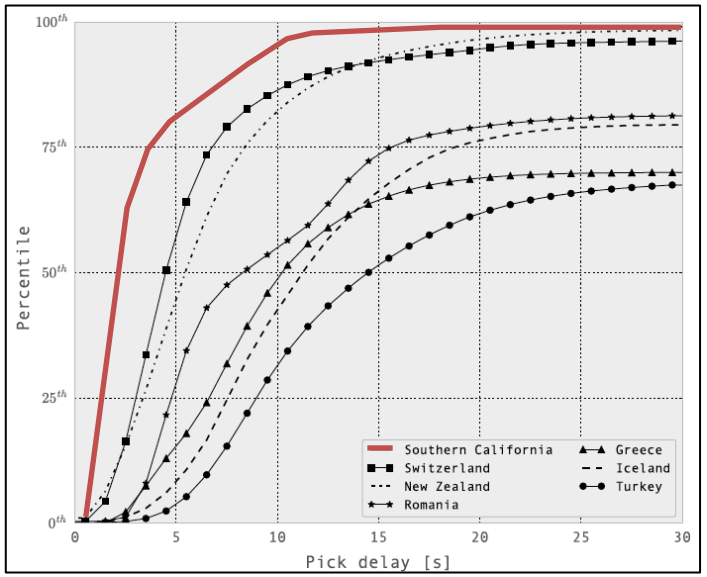
Comparison of different network latencies

Recently, *Behr et al., SRL, 2016* analyzed data from several regional seismic networks deployed around the globe. They showed that the **SCSN** had the **smallest data communication delays or latency**.



Comparison of data transport latencies measured at Caltech among several large networks. The latencies maybe different at the recording sites of the operating networks.

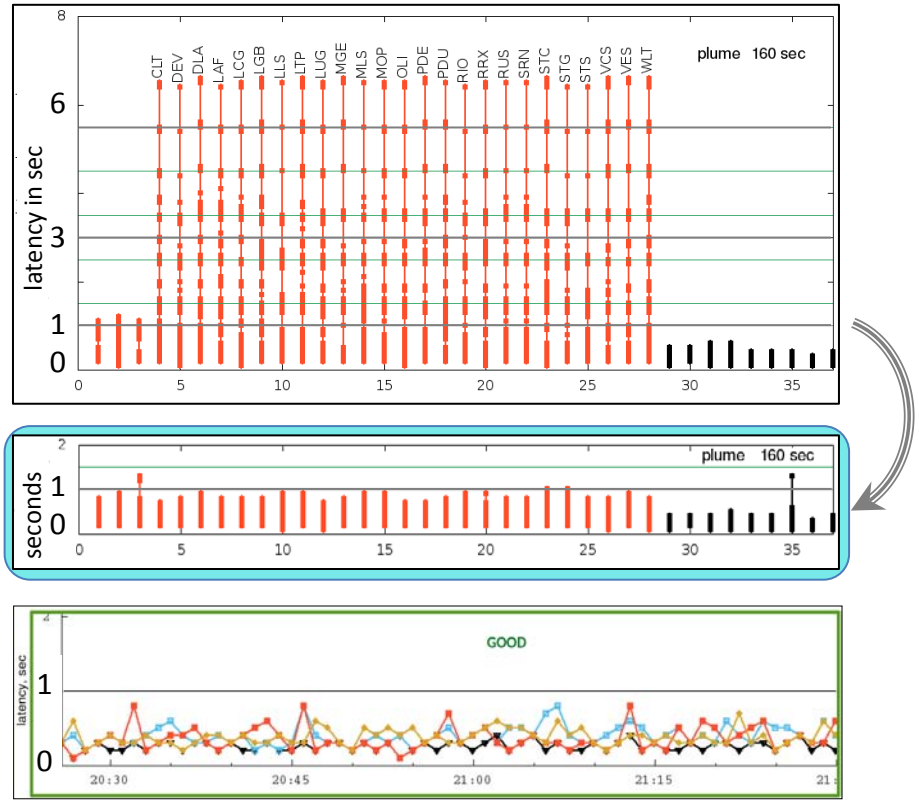


A figure from *Behr et al., SRL, 2016*. Cumulative distribution functions for pick delays measured for each network as the time between the detection of the P-wave arrival at the processing hub and the recording of the corresponding waveform data at the datalogger.

Improving the Q330 latencies

Since the beginning of 2016, we have reduced further the telemetry delays for many of the 330 current sites: on average from 2-6 sec to ~0.4 sec by tuning the datalogger parameters and/or deploying software upgrades (for Basalts).

Example: the company's high bandwidth circuits were redesigned for VOIP (Voice over IP) applications. The VOIP network requirements are different from the regular Ethernet network with the packets delivered like a continuous voice stream: smaller more frequent bundles of data.



Top: VOIP High bandwidth Q330 station latencies before and after the improvements (in red). ~ 6 sec -> < 1 sec. *Bottom:* 1-sec packet latency per packet for 4 different sites after the improvements.

What we changed?

Settings (min)	Default	New values
Min timeout (0.2s)	1.0	0.2
Max timeout (0.2s)	10.0	2.0
ACK timeout (0.2s)	0.3	0.2
ACK Grouping (1)	4.0	1.0
Max latency, sec	*6.5*	*1.0*
Min latency, sec	*0.2*	*0.15*

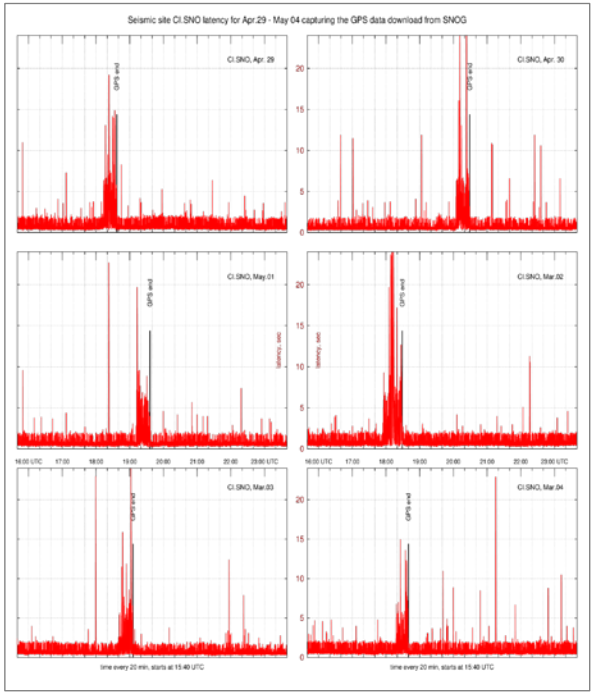
Conclusions: for different networking environments with challenging telemetry, the Q330 Data Port 1 settings should be customized and/or tuned accordingly. In most of the cases, the Standard data logger default settings will suffice.

Check: The *packet re-sent number* is *small* when the settings are correct.

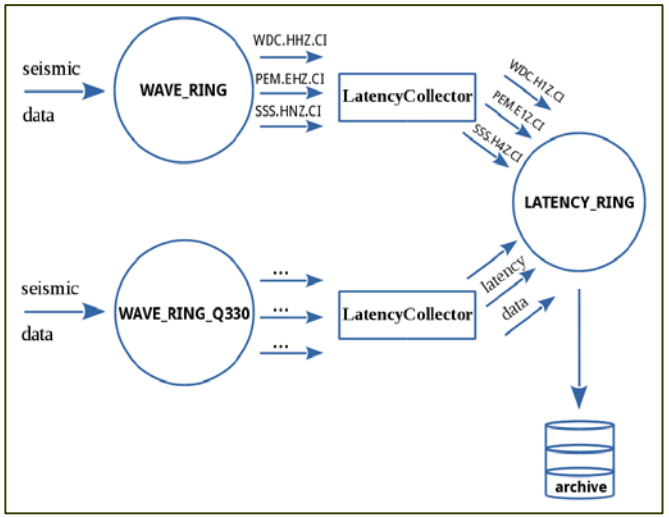
Archiving per-packet latencies in mseed format

Recognizing the latency data as one of the crucial parameters in EEW, we have started **archiving the per-packet latencies in mseed format** for all the participating sites.

Benefits: 1) understand and document long-term changes in performance of the telemetry links,
2) retroactively investigate how latent the waveform data were during a specific event or a specific time period,
3) compare different telemetry technologies and identify the behavioral similarities in a large pool of telemetry links by using seismic software tools to process the latency archived data, like filtering, cross correlation, etc.



Analysis of packet latencies while investigating the effect of the GPS daily data downloads over shared with the seismic sites telemetry links.



Flow: transport latency D of one Z channel per site from a **sniffwave** output -> save interpolation in **mseed** -> **tracebuff2** -> **multicast** to archive.

For the 1-sec packets, use their time.
For the randomly sized packet, apply a linear piece-wise interpolation and save the 1-sec apart values.

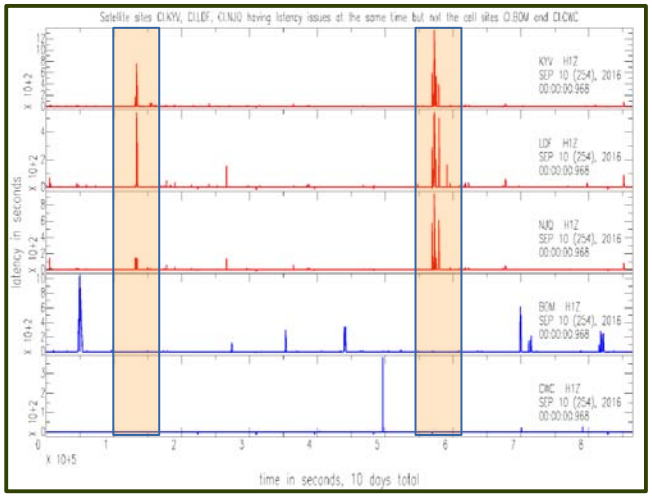
Latency channel naming: replace the instrument code letter (the second letter, like ***N*** in *HN**Z***) with 1 for H, 4 for N, i.e.: HHZ channel latency -> H1Z.

Output: 10 min long mseed H1Z, H4Z, and E4Z data in LATENCY_RING.

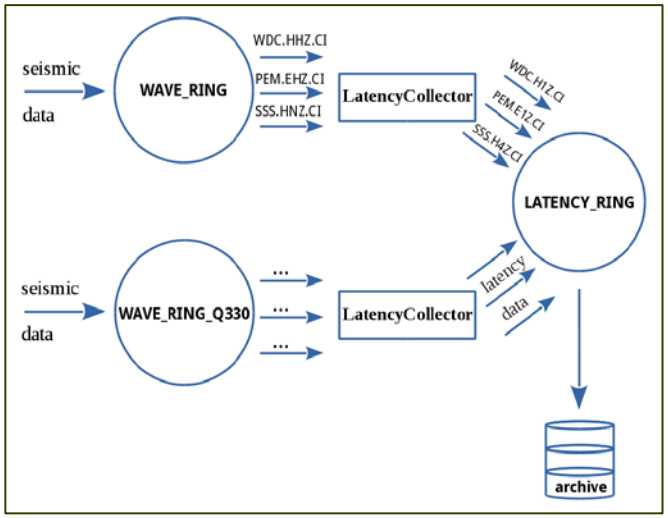
Archiving per-packet latencies in mseed format

Recognizing the latency data as one of the crucial parameters in EEW, we have started **archiving the per-packet latencies in mseed format** for all the participating sites.

Benefits: 1) understand and document long-term changes in performance of the telemetry links,
2) retroactively investigate how latent the waveform data were during a specific event or a specific time period,
3) compare different telemetry technologies and identify the behavioral similarities in a large pool of telemetry links by using seismic software tools to process the latency archived data, like filtering, cross correlation, etc.



Data latency inspection of the sites with satellite connections (red) vs. the radio sites (blue). Note the disruptions that happen at the same time. Exact cause is unknown, possibly a bad weather or solar magnetic storms.



Flow: transport latency D of one Z channel per site from a **sniffwave** output -> save interpolation in **mseed** -> **tracebuff2** -> **multicast** to archive.

For the 1-sec packets, use their time.
For the randomly sized packet, apply a linear piece-wise interpolation and save the 1-sec apart values.

Latency channel naming: replace the instrument code letter (the second letter, like ***N*** in *HN**Z***) with 1 for H, 4 for N, i.e.: HHZ channel latency -> H1Z.

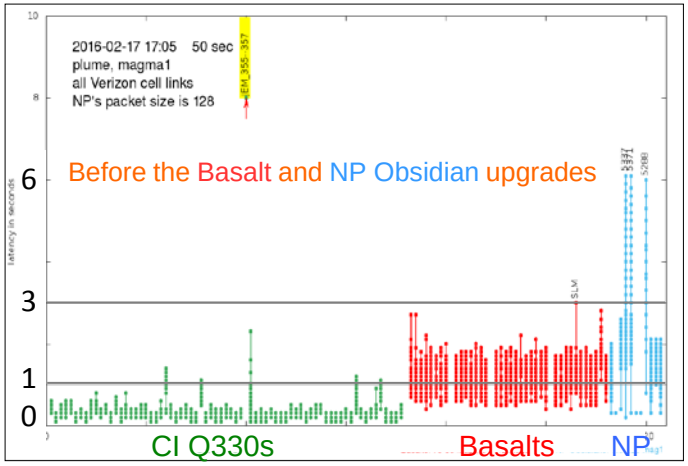
Output: 10 min long mseed H1Z, H4Z, and E4Z data in LATENCY_RING.

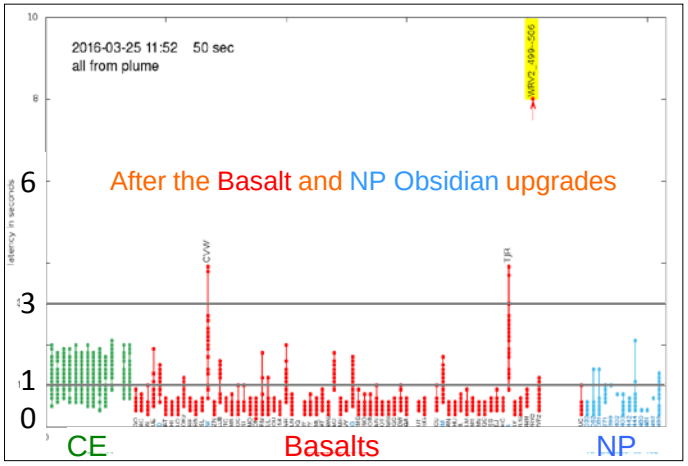
Improving the Basalt and Obsidian latencies

Collaborated with Dennis Pumphrey from Kinemetrics to produce a new Basalt/Obsidian firmware tuned for the EEW to
1) start delivering 1 sec packets instead of randomly sized,
2) improve the data delivery latency to be much closer to the Q330 type of instruments' performance.

KMI released the **3.14.2f** firmware version with the following modifications:
1. Tuned the loops to hand off the telemetry data because occasionally the 1 second packets get delayed by another second.
2. On 4 channel units, serviced the DSPs (producing the data) more frequently.
3. Disabled "nageling" on 4 channel systems to reduce TCP transmission delays (by default, TCP uses Nagle's algorithm to collect small outgoing packets to send them all at once. This may have a detrimental effect on latency).

To benefit from those changes, a few datalogger values should be modified:
a) Seed Link Record Size - changed from 128 to 512;
b) Samples Force Encode - changed from 0 to 100





Noticeable downsides:
1. the FW version 3.14.2f is too aggressive for the weak telemetry links and make the latency even worse;
2. the total monthly data volume for a typical EEW station *may* increase up to 200 MB/day=6 GB/month, depending on a site, since we force the Seed Link Record Size to be 512 bytes.

Noticeable improvements:
1. much lower latency (from ~2 sec to ~0.5 sec);
2. shorter range between the low and high latency;
3. smaller lower bound values;
4. consistency in the transport times.