

SDV WG - uProtocol-Zenoh Weekly Meeting

Join Zoom Meeting

<https://eclipse.zoom.us/j/83187682865?pwd=CAxhUyIRY0PSXqcVw2XYuRxtvN1GVL.1>

Meeting ID: 831 8768 2865

Passcode: 497074

May 15, 2024 |

📅 [SDV WG] uProtocol & Zenoh weekly @ 14 CET

Cancelled no agenda topics

May 8, 2024 |

📅 [SDV WG] SDV WG - uProtocol & Zenoh weekly @ 17 CET

Apr 30, 2024 |

📅 [SDV WG] uProtocol & Zenoh weekly @ 14 CET

Houskeeping Items:

- None

Zenoh Performance Issues

- **All subscribers callbacks are called from the same thread**
 - <https://github.com/eclipse-uprotocol/up-client-zenoh-cpp/issues/30>
 - <https://github.com/eclipse-uprotocol/up-client-zenoh-rust/issues/25>

Configuration: router, client, peer

- Zenoh 0.11.0 is being released. Applications can load plugins via config parameters.
- <https://github.com/eclipse-uprotocol/up-spec/pull/103>

Source/Sink Address Mapping:


- <https://github.com/eclipse-uprotocol/up-rust/pull/92>
-

up-client-zenoh-cpp: Refactoring & revised API proposal

MAC Requirements:  uProtocol Message Authentication

Action items

- ☐ Steven: Add an issue to up-client-zenoh-cpp to elaborate on how to load well known zenoh configurations that we will use for the uProtocol Roadmap (peer, client, router). https://github.com/eclipse-zenoh/zenoh/blob/main/DEFAULT_CONFIG.json5 . Do we pass config file vs Zenoh Config structure.
- ☐ Steven: Review PR 92 from up-rust and share the uTransport changes with the rest of the committer community (and why)

Apr 24, 2024 |  SDV WG - uProtocol & Zenoh weekly

Houskeeping Items:

- Request meetings to rotate from 2pm to 5pm CET so others can join every other week.
- May 1st is a bank holiday in Europe and Taiwan => Move the meeting to April 30th, 2pm CEST.
- May 8th is bank holiday in France

Zenoh Performance Issues

- ~~Session startup time of ~700msec~~ Reduced to 4.5ms now once crashes are
- ~~Number of open threads~~ <https://github.com/eclipse-zenoh/zenoh/pull/893>
- **All subscribers callbacks are called from the same thread**
 - <https://github.com/eclipse-uprotocol/up-client-zenoh-cpp/issues/30>
 - <https://github.com/eclipse-uprotocol/up-client-zenoh-rust/issues/25>
- **High latency with 1 pub & 20 sub processes for 1K , 10K , 100K packets**
meirov.uzi@gmail.com to file an issue for this
- **Errors and data loss when running 1 pub & 20 sub** – This causes data not to be received by the subscribers
meirov.uzi@gmail.com to provide logs and file issue in zenoh-c as needed
- ~~All data is being sent/received on one socket~~
- ~~Shared memory is not cleaned after process termination~~ – even when not using shared memory (see /dev/shmem), see ticket <https://github.com/eclipse-zenoh/zenoh-c/issues/245>
This issue was solved by the SHM refactor (PR: <https://github.com/eclipse-zenoh/zenoh/pull/596>).
- ~~Possible memory leaks in zenoh~~ – Valgrind should possible lost/still reachable <https://github.com/eclipse-zenoh/zenoh-c/issues/188>
Memory leaks are fixed (PR: <https://github.com/eclipse-zenoh/zenoh/pull/846>).

Configuration: router, client, peer

- Zenoh 0.11.0 is being released. Applications can load plugins via config parameters.

Python attachment

- Merged into main. It will be available in Zenoh 0.11.0.

Conan recipes:

- <https://github.com/eclipse-zenoh/zenoh-c/pull/336>
- <https://github.com/eclipse-zenoh/zenoh-c/pull/334>
- Still need to figure out where to host recipes in eclipse-zenoh. Some intermediate step could be to host them on <https://github.com/ZettaScaleLabs/conan-recipes/>

up-client-zenoh-cpp: Refactoring

Action items

- ☐ Uzi to file issues/investigate problems reported above
- ☐ CY to provide an example of a C++ router that loads plugins
- ☐ Pete to file an issue / open a PR against **up-spec** to capture Kai's idea of including both src and sink authorities such that we can discriminate between publish / non-publish messages to make uStreamer & authentication cases simpler
- ☐ Steven to share the MAC requirements for zenoh

Apr 17, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Houskeeping Items:

- Request meetings to rotate from 2pm to 5pm CET so others can join every other week.

Conan

- Zenoh-C create recipe done however CCI refuses to accept the project as it requires the rust toolchain
- In discussion about where to house this and other projects other than CCI

Zenoh Performance Issues

- ~~Session startup time of ~700msec~~ Reduced to 4.5ms now once crashes are
- **Number of open threads** – <https://github.com/eclipse-zenoh/zenoh/pull/893>
- **All subscribers callbacks are called from the same thread**
 - <https://github.com/eclipse-uprotocol/up-client-zenoh-cpp/issues/30>
 - <https://github.com/eclipse-uprotocol/up-client-zenoh-rust/issues/25>
- **High latency with 1 pub & 20 sub processes for 1K , 10K , 100K packets**
meirov.uzi@gmail.com to file an issue for this
- **Errors and data loss when running 1 pub & 20 sub** – This causes data not to be received by the subscribers
meirov.uzi@gmail.com to provide logs and file issue in zenoh-c as needed
- ~~All data is being sent/received on one socket~~
- ~~Shared memory is not cleaned after process termination~~ – even when not using shared memory (see /dev/shmem), see ticket

<https://github.com/eclipse-zenoh/zenoh-c/issues/245>

This issue was solved by the SHM refactor (PR:

<https://github.com/eclipse-zenoh/zenoh/pull/596>).

- ~~Possible memory leaks in zenoh~~ – Valgrind should possible lost/still reachable

<https://github.com/eclipse-zenoh/zenoh-c/issues/188>

Memory leaks are fixed (PR: <https://github.com/eclipse-zenoh/zenoh/pull/846>).

C++ Router Plugin

- CY working on examples of how to do this from zenoh configuration script

Action items

- ☒ Greg: Test zenoh-c conan recipe: it works!
- ☐ Uzi to file issues/investigate problems reported above

Apr 10, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Houskeeping Items:

- Meeting shifts from 2pm CEST to 5pm CEST.

Zenoh Performance Issues

- **Session startup time of ~700msec** - Reduced to 4.5ms now once crashes are resolved. Reason why it is not faster than 4.5ms: 4-way handshake for zenoh and 3-way TCP handshake connecting to the router. Zettascale working on profiling the breakdown of the 4.5 msec, will share more details next week
- **Number of open threads** – <https://github.com/eclipse-zenoh/zenoh/pull/893>
 - AI for GM to try the 4 thread config
- **All subscribers callbacks are called from the same thread**
 - C++ & rust up-client needs to be updated to manage callback fifo/queue for callbacks (Steven Hartley to file issues in the different projects to capture this request). They are working on providing this functionality in zenoh proper (in lieu of up-client)
- **High latency with 1 pub & 20 sub processes for 1K , 10K , 100K packets**
meirov.uzi@gmail.com to file an issue for this
- **Errors and data loss when running 1 pub & 20 sub** – This causes data not to be received by the subscribers
meirov.uzi@gmail.com to provide logs and file issue in zenoh-c as needed
- **All data is being sent/received on one socket**
 - Actually one socket per-session so in clique topology there are multiple sockets (between P2P connections)

- Multiple links are supported in zenoh (for fallover). Traffic could be differentiated based on Qos. Could be exposed/expanded through `uSubscription EventDeliveryConfig`.
- Table this topic till we identify an actual performance issue
- **Shared memory is not cleaned after process termination** – even when not using shared memory (see `/dev/shmem`) , see ticket <https://github.com/eclipse-zenoh/zenoh-c/issues/245>
This issue was solved by the SHM refactor (PR: <https://github.com/eclipse-zenoh/zenoh/pull/596>).
- **Possible memory leaks in zenoh** – Valgrind should possible lost/still reachable <https://github.com/eclipse-zenoh/zenoh-c/issues/188>
Memory leaks are fixed (PR: <https://github.com/eclipse-zenoh/zenoh/pull/846>).

C++ Router Plugin

- Discussed the revised topology (v3) that will be C++ based streamer. Example provided and involves initializing zenoh in router mode in lieu of peer or client mode
- AI Luca to work to

Action items

- ☐ Luca: Add support to launch plugins from the configuration file
- ☐ Steven: File tickets in up-client-zenoh-cpp & rust for the callback fifo
- ☐ Uzi: Sync up with Michael for two items above on latency and data loss

Apr 3, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Houskeeping Items:

- Meeting shifts from 2pm CEST to 5pm CEST.

Zenoh Performance Issues

- **NEW RPC request round trip for 8 bytes of data is 600-700 microseconds**
- **Session startup time of ~700msec** - Current solution provided by zettascale causes process to crash (target should be < few 10s ms)
In the examples we provided we set the 2 current delays to 0. This should probably not crash but setting those values to 0 will cause issues anyway.
We removed those delays and reduced the startup time for clients in this branch: https://github.com/eclipse-zenoh/zenoh/tree/open_delay. The question is should this be done for peers short term as well?
Regarding the target, it's probably not possible to establish one or several secured Zenoh session(s) to remote peer(s)/router(s) in less than a ms. So the question is:

what should be the contract of the zenoh::open function ? What should be considered ready when it returns ?

- **Number of open threads** – Zenoh opens currently ~30 threads per process
We shared a PDF that shows how to reduce the number of threads to 4. This is valid for the async-std version. We are working on providing similar instructions for the tokio version.
- **All subscribers callbacks are called from the same thread** – this can cause delay, higher latency and frame drops
We are currently working on providing a z_sample_fifo handler that would allow users to spawn a thread for each subscriber and read samples from this fifo. This would be similar to our current z_get example. Note that this will increase latency.
- **High latency with 1 pub & 20 sub processes for 1K, 10K, 100K packets**
To be investigated.
- **Errors and data loss when running 1 pub & 20 sub** – This causes data not to be received by the subscribers
To be investigated.
- **All data is being sent/received on one socket**
- **Shared memory is not cleaned after process termination** – even when not using shared memory (see /dev/shmem), see ticket <https://github.com/eclipse-zenoh/zenoh-c/issues/245>
This issue was solved by the SHM refactor (PR: <https://github.com/eclipse-zenoh/zenoh/pull/596>).
- **Possible memory leaks in zenoh** – Valgrind should possible lost/still reachable <https://github.com/eclipse-zenoh/zenoh-c/issues/188>
Memory leaks are fixed (PR: <https://github.com/eclipse-zenoh/zenoh/pull/846>).

Zenoh protocol updates (6/24):

https://github.com/eclipse-zenoh/zenoh/tree/protocol_changes

Expected KPIs:

- Peer boot-up time:
 - If data is not lost: < 20ms per peer
 - If data could be lost the delay is ~0ms

Group management:

- https://github.com/eclipse-zenoh/zenoh/blob/protocol_changes/zenoh-ext/example/s/z_member.rs
- uTwin in a group so we can unblock the z_open() for publishers, to be investigated

Discussion on zenoh socket design, Zenoh reuses the same socket across all processes for communication

Action items

- ☐ Michael to provide instructions to reproduce the various issues above
- ☐ Investigate group management feature to unblock startup time
- ☐ Luca to provide updated instructions for configuring the # of threads using the new runtime tokio
- ☐ CY/Michael: update up-client-zenoh-cpp|rust to change the subscriber callback logic to use either callback, fifo or ring buffer to how to handle back pressure

Mar 27, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Attendees: steven.hartley@gm.com

c_2ampi2bmoka3qter4dceap1d5g@group.calendar.google.com igal.shnaider@gmail.com

Notes

-

Action items

- ☐

Mar 20, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Notes

- Up-client updates (rust, C++):
 - Configurations (peer/client/router)
- `registerListener()` for notifications:
 - <https://github.com/eclipse-uprotocol/up-spec/issues/81>
 -
- Roadmap discussion
- Issues

Action items

- ☐

Mar 13, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Attendees: steven.hartley@gm.com

c_2ampi2bmoka3qter4dceap1d5g@group.calendar.google.com igal.shnaider@gmail.com

Notes

- Topology Review
 - Discussed proposed plugin architecture using up-streamer-rust library
 - No issues identified
- Record & Playback
 - Requirements:
 - Support all message types: notifications, publications, requests, response
 - Both local & remote traffic: between zenoh peers, between devices (streamers), etc...
 - System state snapshot: current state of the vehicle (uTwin state?)
 - Zenoh supports distributed storage that could be used for recording
 - Zenoh Recording
 - Given the clique topology, we cannot simply inject a passive scanner into the “system” to record request/responses, these messages are sent directly between client & service
 - Need to introduce Recorder capabilities to UTransport to record what is sent and received into a common storage. Then services will record everything
 - Playback
 - Playback will also be tricky. When we playback we mock the think we are playing so we need to identify who we are (that we are mocking) when we call play(storage)

Action items

- ☐ CY, Peter, Michael: Zenoh configurations for peer vs client vs plugin

Mar 7, 2024 | 📅 SDV WG - uProtocol

Attendees: [Steven Hartley](#)

Notes

- uTwin Implementation for Zenoh:
 - Implemented with storage (plugin in zenoh)
 - Queryable Subscriber

Action items

- ☐ Steven update the zenoh ticket to include the diagram as to why we need local authority filtering in the routers (<https://github.com/eclipse-zenoh/zenoh/issues/773>)
- ☐ Daniel: COVESA uServices rust examples
- ☐ CY: Move the examples for upc-zenoh-rust to a new project
- ☐ Luca: to share more info on queryable subscriber so we can add this to upc-zenoh-cpp & rust

Feb 28, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Attendees: steven.hartley@gm.com

c_2ampi2bmoka3qter4dceap1d5g@group.calendar.google.com igal.shnaider@gmail.com

Notes

- Zenoh-java Updates Darius
 - Binder supports zero-copy and we are measuring the mean of ~1ms for sending uMessage from uE-->uBus-->uE vs 1.7ms for uE-->uE using zenoh-java.
 - Darius is investigating performance of two java apps talking over TCP to see if the issues are there or not.
- uProtocol-Zenoh Spec & Implementation
 - <https://github.com/eclipse-uprotocol/up-spec/pull/52>
 - UUri Mapping issue
 - Agreed on using two different namespaces for local vs remote: upl/ and upr/
 - CY will open issue over on eclipse-zenoh to document new behavior for Zenoh Router to only forward messages from "upr/**"
 - CY will update up-spec for Zenoh to highlight the above change for upl/ vs upr/ and also update how to put uAttributes into Zenoh user attachment
 - CY will update [up-client-zenoh-rust](#) initial PR to conform with upl/ vs upr/ namespace distinction
 - Upc-rust update CY
-

Action items



Feb 21, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Attendees: steven.hartley@gm.com

c_2ampi2bmoka3qter4dceap1d5g@group.calendar.google.com igal.shnaider@gmail.com

Notes

- Up-spec <https://github.com/eclipse-uprotocol/up-spec/pull/52>
- up-client-zenoh-rust
 - Updated to the latest up-rust changes that have been merged
 - Finalizing the uTransport interface, targeting initial PR review for this portion only, at the end of this week
 - We decided to stick to serializing via Protobuf the entire UAttributes as an attachment for a v1 – CY, what's the timeline on this?
 - We can in parallel ask CY to explore benchmarking alternatives such as putting each attribute individually via the field ID (the number) and only needing to fill out those that are not captured in Zenoh-specific header
 - Key expression mapping: Need to standardize the URI mapping to key expression and document in spec ASAP
- up-client-zenoh-cpp:
 - No updates
- up-streamer-rust
 -

Action items

- ☐ CY to work on sending full UAttributes & mapping of UUri
- ☐

Feb 14, 2024 | 📅 SDV WG - uProtocol & Zenoh weekly

Attendees: steven.hartley@gm.com

Notes

- Up-client-zenoh-rust
 - No updates, CY is on vacation
- up-client-zenoh-cpp:
 - Conan packages are done, no other update
- up-streamer

- using the up-clients zenoh & android and working with Misha & CY to work out all the kinks
- Zenoh: PR about priority mapping. Was not able to pull the priority from a query but the PR allows retrieving the priority “off the wire”. Once merged needs to be applied to the other bindings.
- Igal: Security, pub/sub implementation
-

Action items



Older Meeting Minutes:

<https://gitlab.eclipse.org/eclipse-wg/sdv-wg/sdv-technical-alignment/sdv-technical-topics/communication-protocols-sdv-topics/-/blob/main/MeetingNotes/uProtocol-zenoh.md>