Zenoh across all the busses

Robotics and Pub Sub

Robotics and Publish/Subscribe have a long history.

- Orca Robotics (mid 2000s) ZeroC Ice
- Player/Stage (mid 2000s) custom TCP
- ROS1 custom TCP protocol
- ROS2 initially DDS and now Zenoh (IP protocols)

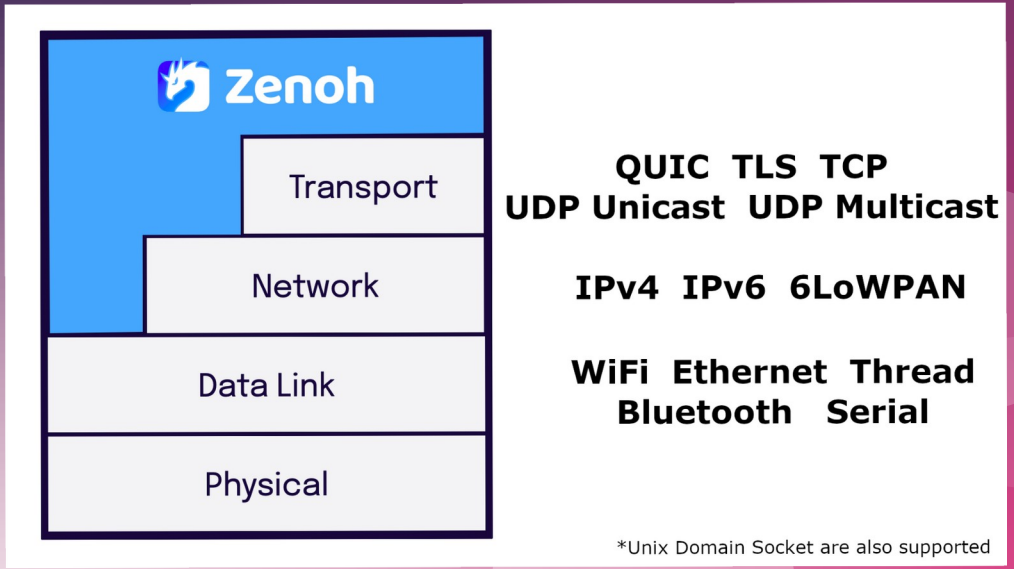
What about microcontrollers!?

What is Zenoh

- Highly efficient messaging protocol
- Topic/Key oriented with verbs (pub, sub, get, put)
- Topics are “compressed” over the wire (index referenced)
- Routed, Peer to Peer, and Mesh network topologies
- “Smallest implementation in 300 bytes on Atmel”
- Like DDS, HTTP, MQTT combined but better and more flexible

Zenoh all the busses

- Zenoh is nicely layered
- zenoh-pico does assume unix socket like APIs exist to open and listen for unicast like connections
- Can we do this over other busses?
- **YES**



Zenoh all the busses

- What about SPI, UART, and I2C? USB?
- We need a new link layer.
- Needs to kind of look like Unix sockets (open, listen, accept, write, read)

MCTP

- Messaging protocol with 8 bit network addressing
- Routing/Bridging across nodes and busses
- Bindings (Specifications) to many physical transports (USB, PCIe, CXL, Serial, I2C, I3C, etc)

Zenoh on MCTP

MCTP as a link layer in Zenoh means Zenoh over all the physical layers MCTP supports.

Zenoh on all the busses (almost)

MCTP as a Zenoh Link Layer

```
171 z_result_t _z_new_link_mctp(_z_link_t *zl, _z_endpoint_t endpoint) {
172     z_result_t ret = _Z_RES_OK;
173     zl->_type = _Z_LINK_TYPE_MCTP;
174     zl->_cap._transport = Z_LINK_CAP_TRANSPORT_UNICAST;
175     zl->_cap._flow = Z_LINK_CAP_FLOW_STREAM;
176     zl->_cap._is_reliable = false;
177
178     zl->_mtu = _z_get_link_mtu_mctp();
179
180     zl->_endpoint = endpoint;
181
182     zl->_open_f = _z_f_link_open_mctp;
183     zl->_listen_f = _z_f_link_listen_mctp;
184     zl->_close_f = _z_f_link_close_mctp;
185     zl->_free_f = _z_f_link_free_mctp;
186
187     zl->_write_f = _z_f_link_write_mctp;
188     zl->_write_all_f = _z_f_link_write_all_mctp;
189     zl->_read_f = _z_f_link_read_mctp;
190     zl->_read_exact_f = _z_f_link_read_exact_mctp;
191     zl->_read_socket_f = _z_f_link_read_socket_mctp;
192
193     return ret;
194 }
```

Potential of Zenoh Everywhere

- Zenoh over every bus means bridging/routing across many cores and devices
- Shared memory on the same device? Across low speed busses? Clusters of small cores?
- Paper it over with Zenoh, talk in **topics** not in custom messages with custom bridges.