TinyNet:
A Lightweight Networking Strategy for Networked Control Systems

6.829 Computer Networks

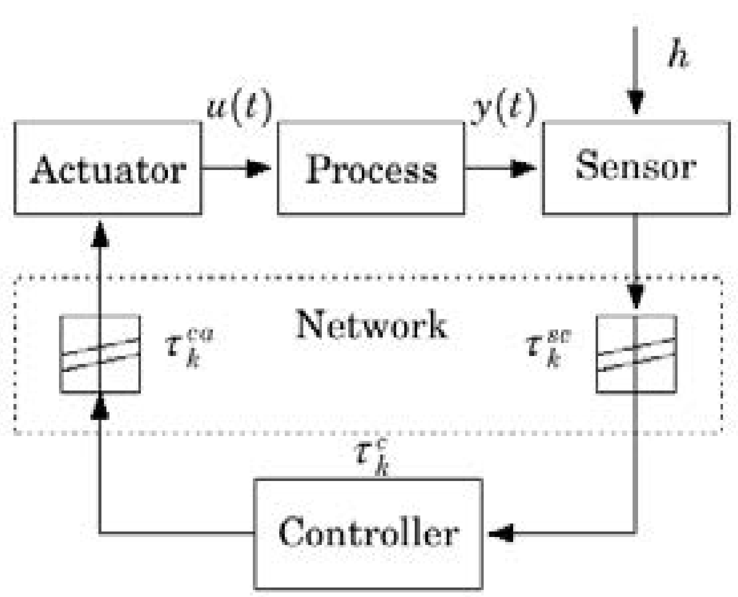
Jake Read, Nick Selby, Dougie Kogut, Patrick Wahl

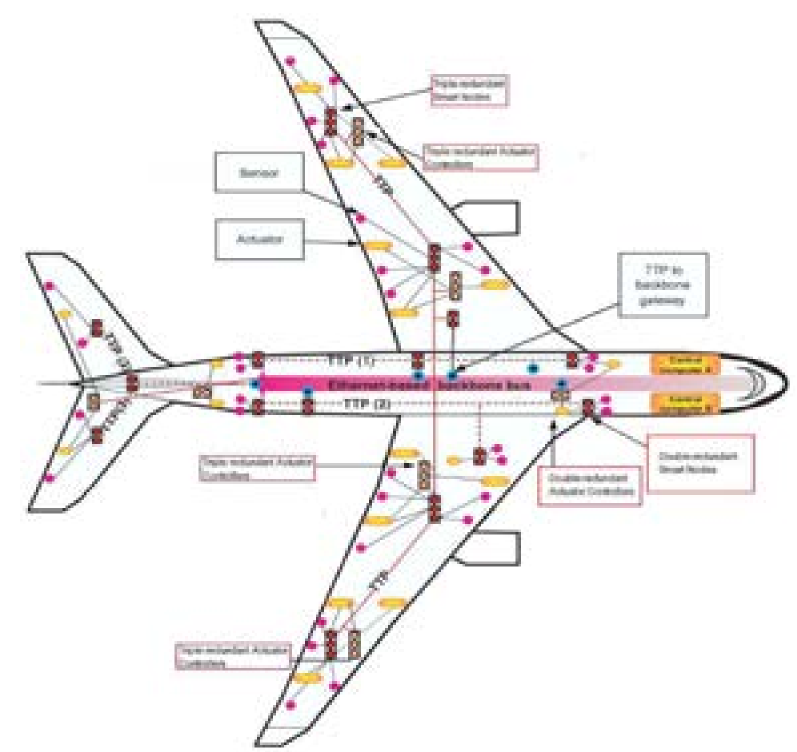
Networked Control Systems

NCS: Performing control on physical systems
- Sensors, Actuators and Controllers communicate over on network link.
- Network Message Delivery Time = Control Loop Quality

Common in Manufacturing, Robotics, Avionics, etc
- Relative systems level complexity as researchers discover advantages of ‘many small’ rather than ‘a few large’ actuators, etc

Different Priorities!
- Relative most other networking, NCS demands Message Delivery Determinism above all other metrics:
- Loop Closing, Bode & System Instability





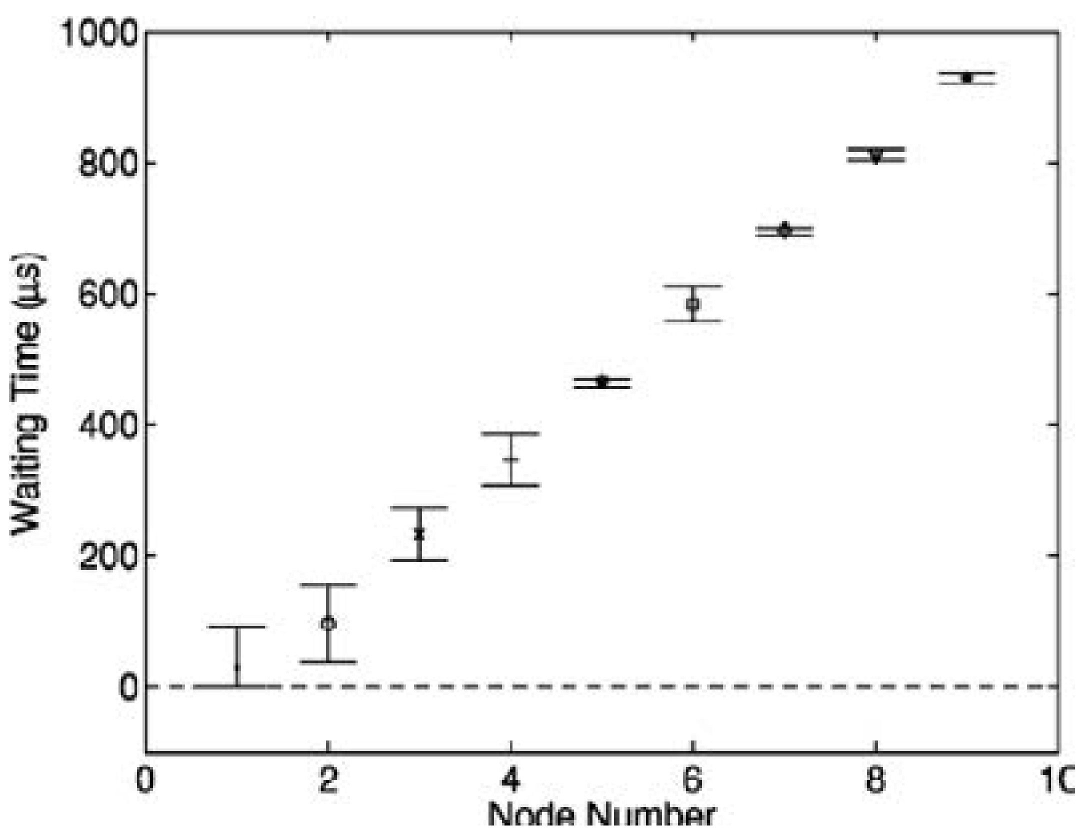
Move towards Switched Ethernet from Fieldbusses

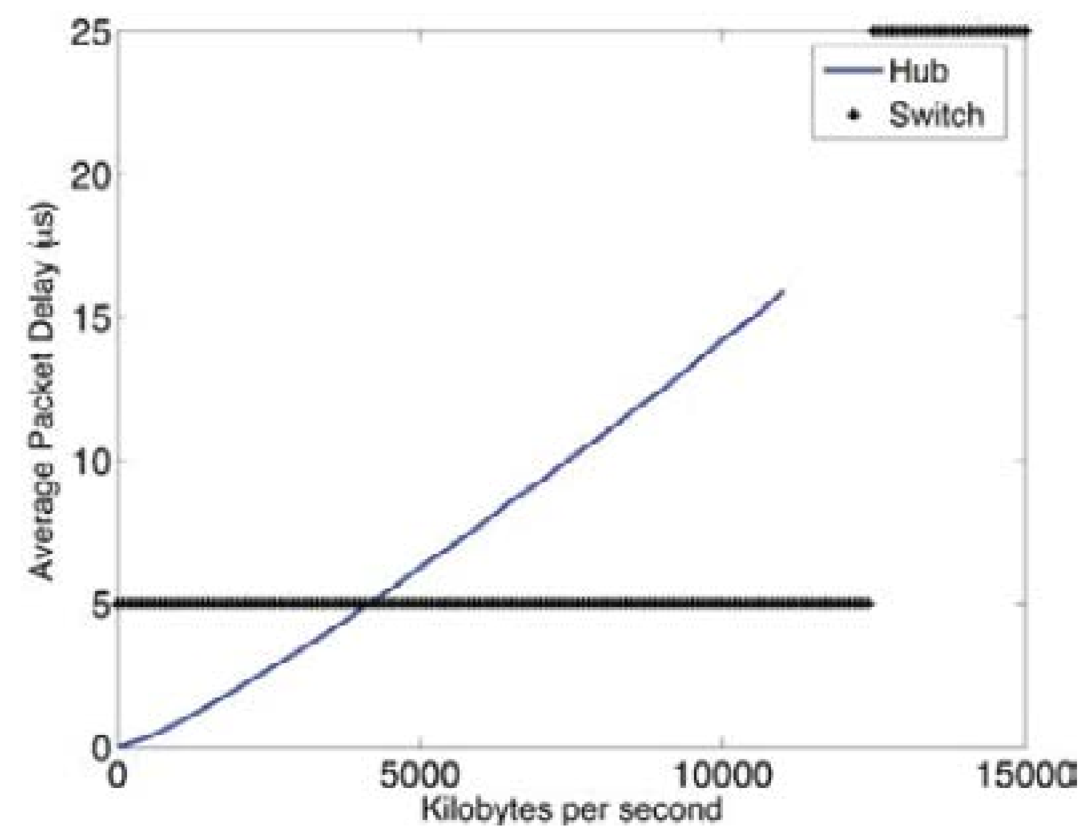
Fieldbusses are multi-drop lines common in realtime control systems

- See an increasing message delay time with added nodes: hard limit on complexity of systems, interoperability, development

Switched Ethernet Permeates industrial NCS (~ 80%)

- A Layer 2 'protocol'
- Often simply implemented with TCP/IP stack, other proprietary protocols exist using SE





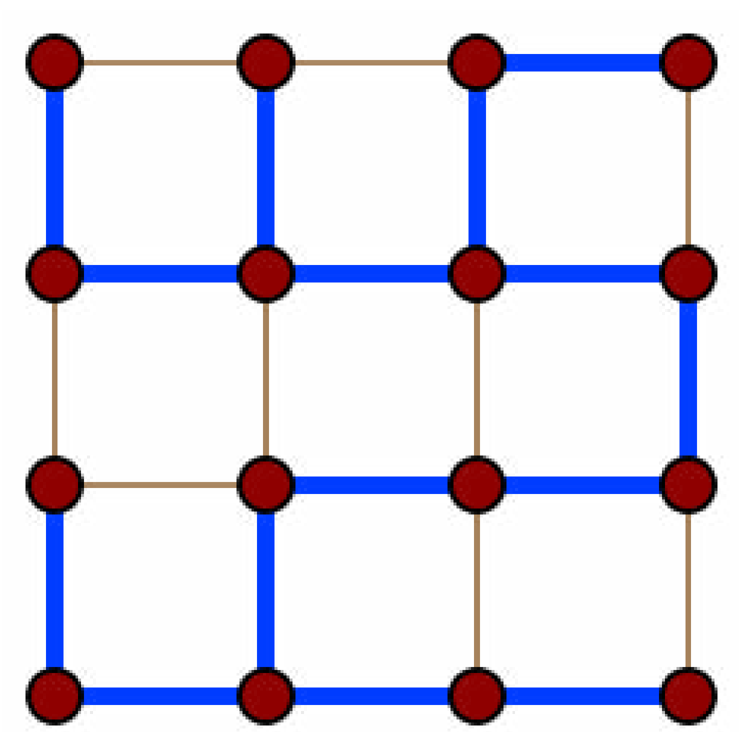
Limitations of Switched Ethernet

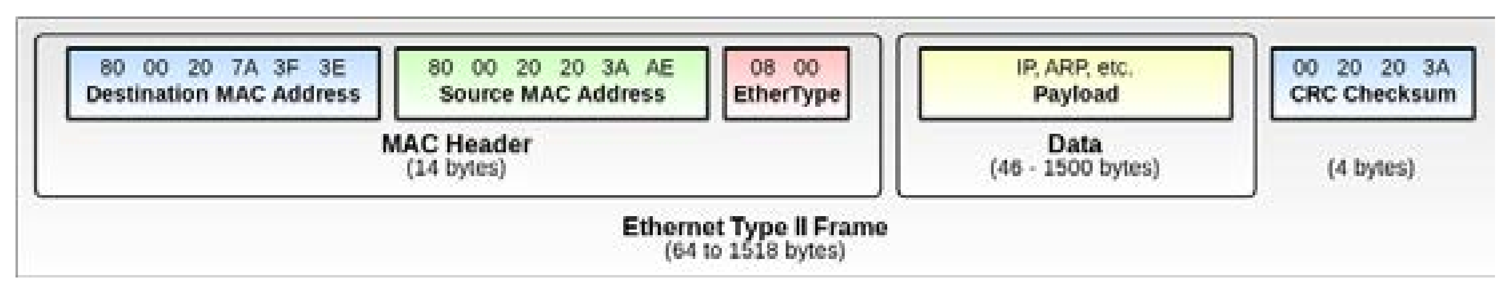
Constructs Spanning Tree Network Graph

- Only one possible route to each endpoint
- Contains Single Points of Failure (SPoF)
- Switches in upper-levels of graph can become busy

Large Message Overhead

- the Ethernet Header is 64 Bytes, NCS packages are typically between 1-8 Bytes





Multipath: Why not use Datacenter Techniques?

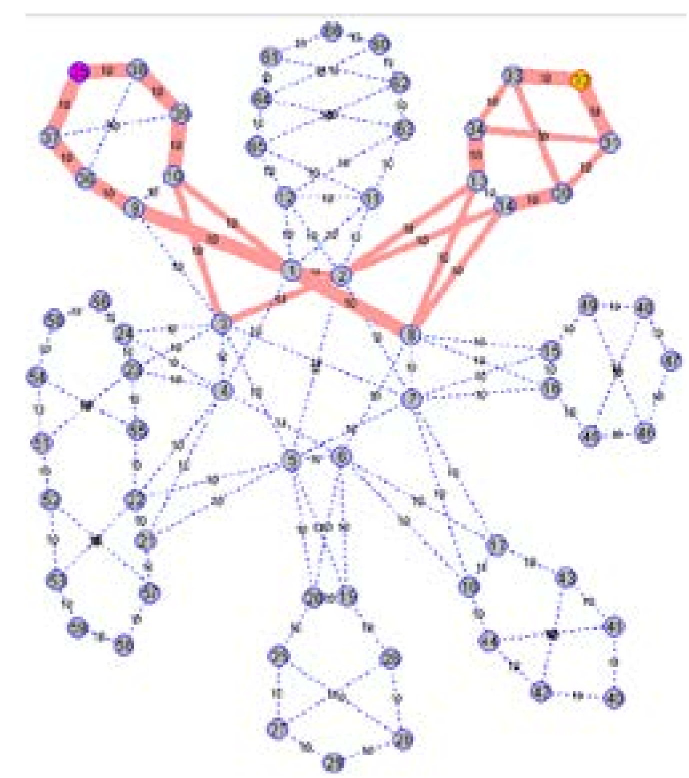
Many strategies exist for Multipath Routing
- Most are developed for use in Datacenter message passing
- OSPF, TRILL, etc...

Link-Layer Protocols, requiring state of entire network graph
- Network Controllers, or Routers, must contain state information about the entire graph

Reconfiguration of Routing Table Rules decreases Message Delivery Determinism
- On changes to the network graph, or increased busyness at routers, networks must undergo reconfiguration
- This is a process that takes 200ms on the lower bound

Not real-time!

Implementations not readily available for small systems



Our Approach

Use Stateless Routing, but maintain enough intelligence to perform greedy multipath

- Statelessness allows routing to take place without large bursts of network-configuration traffic

Use next-hop buffer dept, i.e. 'Busyness Metric' to determine best port forwarding

- Lambda function incorporates known route lengths with information about next-hop buffer wait time

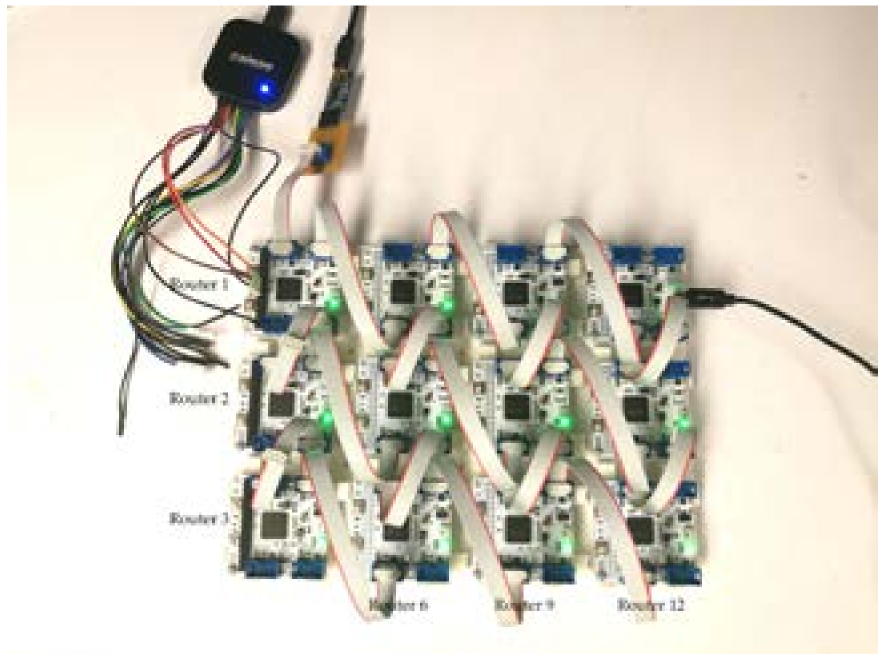
Implement on the same hardware used in endpoint control

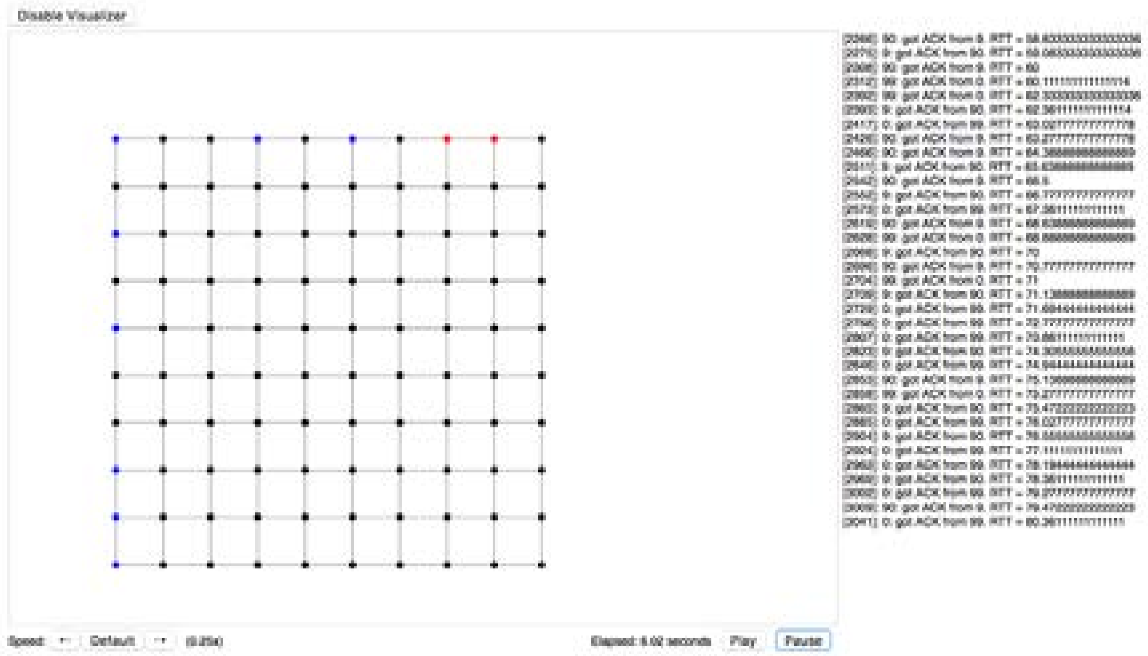
- TinyNet runs in C on easily accessible microcontrollers
- Systems do not require special IC's to become attached to the network: only a software stack

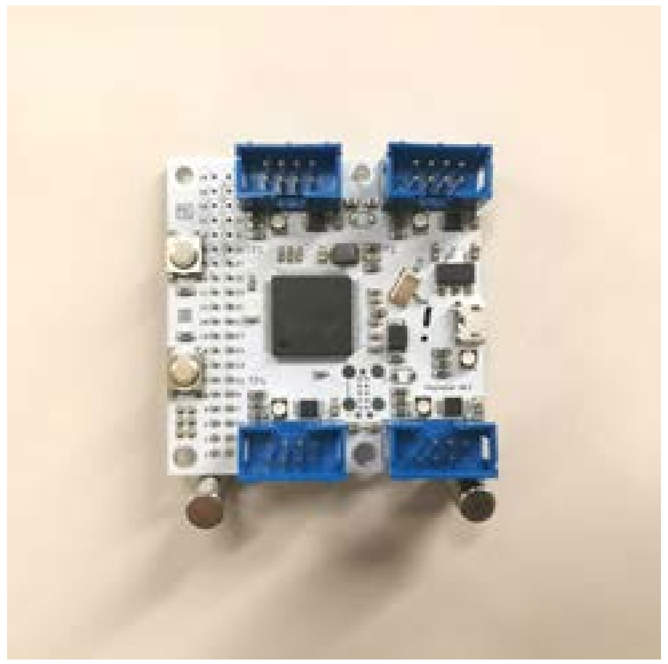
Implementation

Develop hardware to perform routing
- Uses ATSAMS70 microcontroller with UART links
- Use hardware to characterize performance

Develop a Simulation to test larger networks

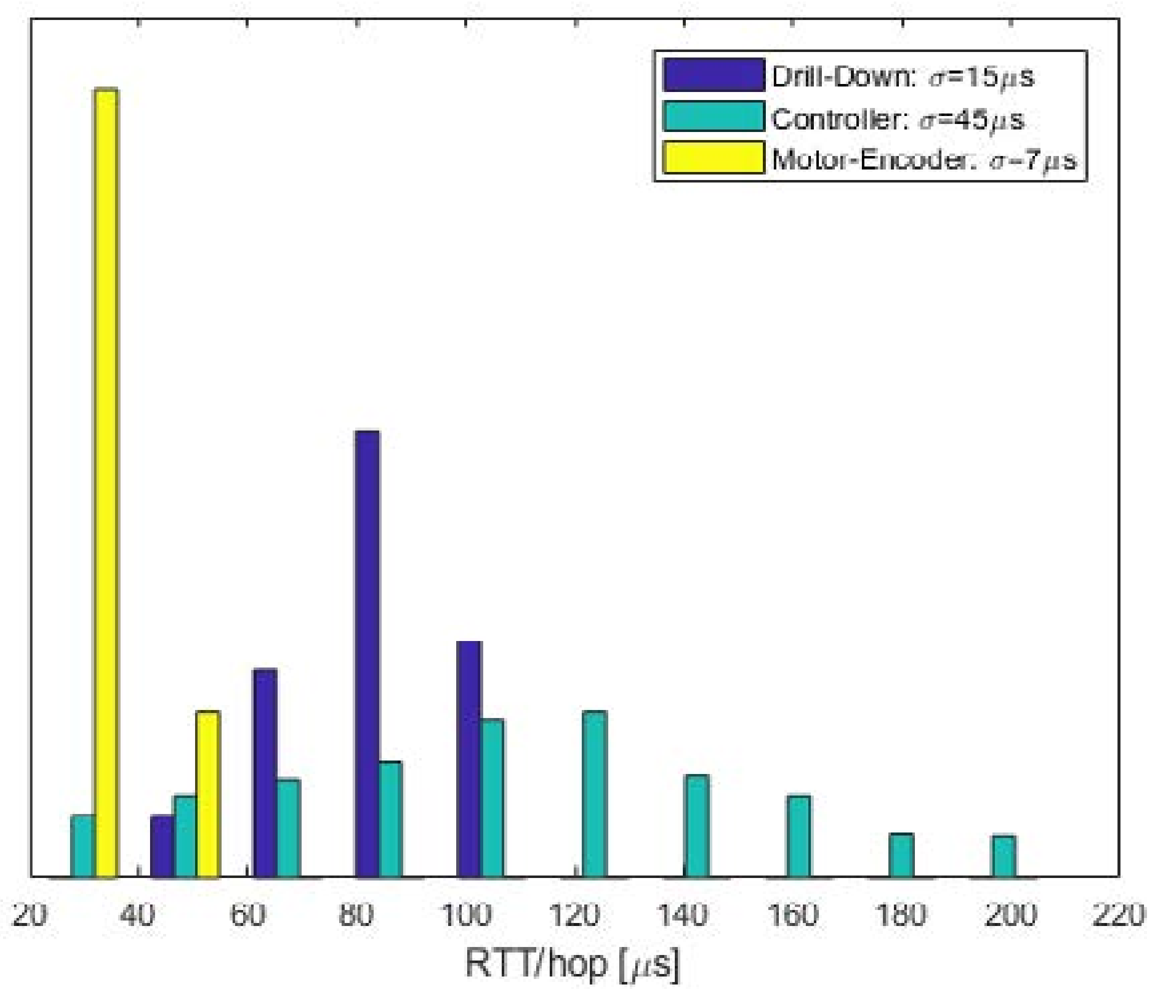


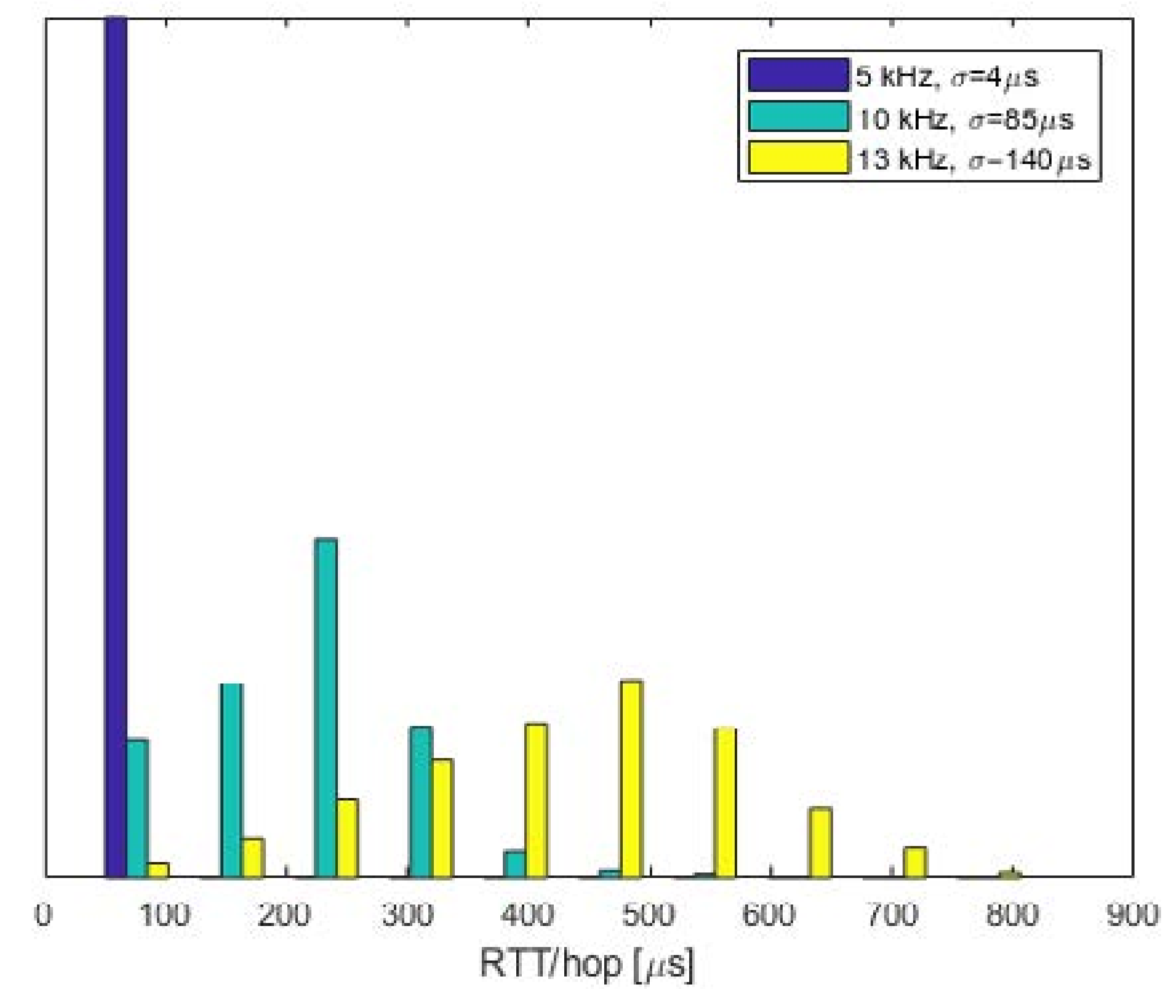




Results

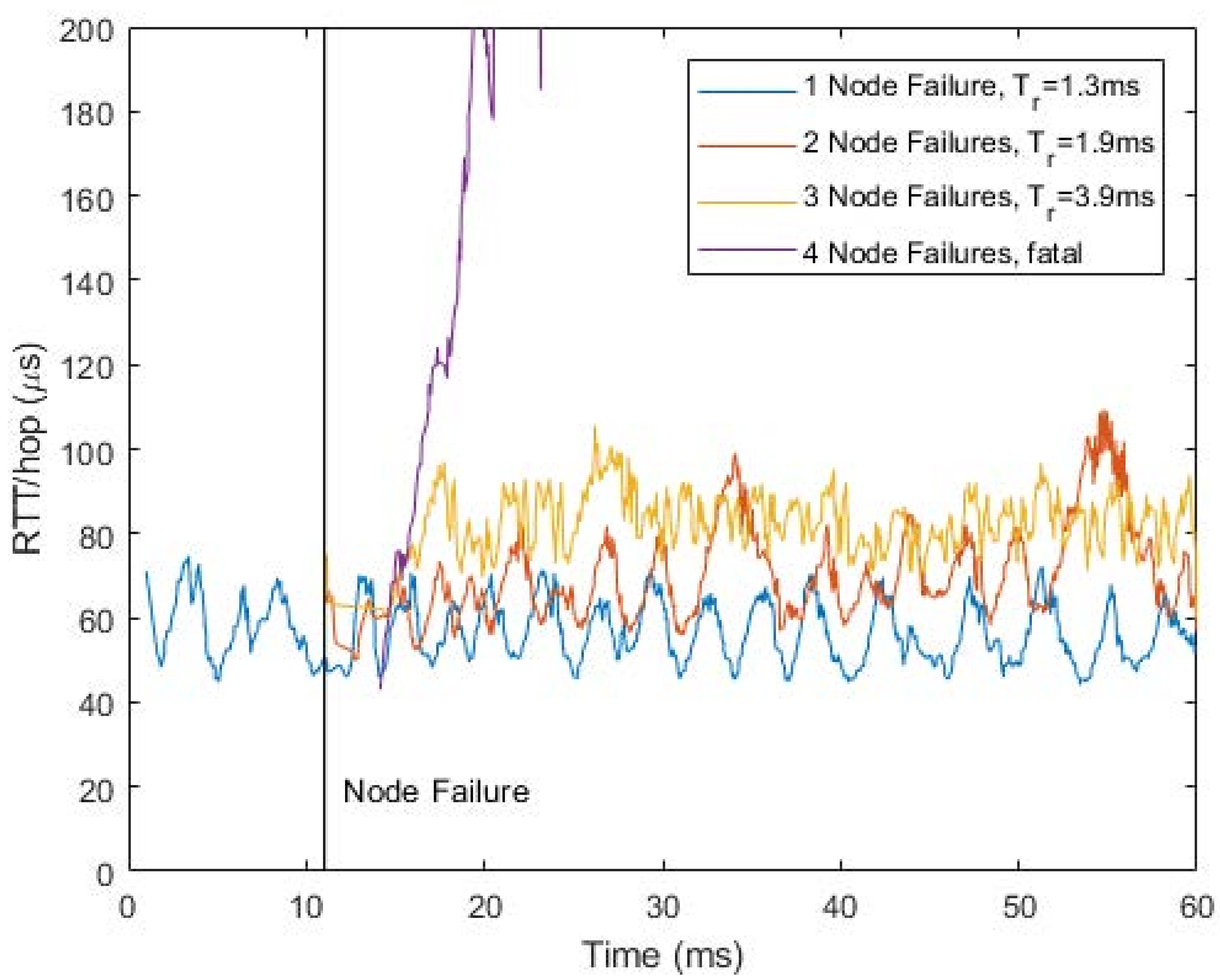
Message Delivery Determinism





Results

Robustness to Link failure



Future Work

Contacted Industry Professionals
- No data! Please, your data?

FPGAs for Stateless Link Layer
- UART is 'stateful' or requires configuration in that bitrate must be selected per port
- We implemented testbed FPGA 'coclocking' technique

FPGA Routing
- TinyNet is simple enough to write into Verilog ?

Learning Lambda
- Difficult to ascertain what a good Lambda Function is (to account for hop-count and buffer depth simultaneously)
- Can we learn this by also measuring packet delay time from each node (using acks) ?

Publish and Implement
- to open source robotics community
- implement hardware endpoints, control them, incorporate distributed controls research

Demo!