Tiny Nets
A Small Network with Industrial Dreams

Jake Read, Dougie Kogut, Nick Selby, Patrick Wahl

October 2017

1 Motivation

1.1 Networked Control Systems

We are motivated by the lack of purpose-built networking solutions for embedded hardware and control. In a Networked Control Systems (NCS), nodes are coincident with physical hardware (I.E. motors, encoders, sensors and switches), and the network is used for Control, Diagnostics and Safety [MT07]. NCS are commonly found on a factory floor where production must be coordinated between multiple machines and material handling systems, in robotics applications where motion must be coordinated across multiple joints or degrees of freedom, and in avionics or other large, complex control systems. NCS also have applications in Building Control.

Interest in NCS has increased in the last ten years, as Figure 1 showcases.

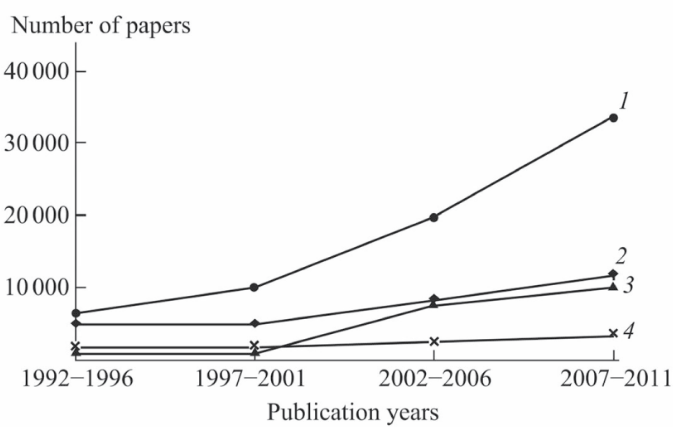

Figure 1: Publications in main subfield of control theory in Web of Science indexed journals: (1) control in networks, (2) adaptive control, (3) intelligent control systems, (4) robust control [PF16]

Interest is increasing for a few reasons. First, industry initiatives for reconfigurable factories and on-demand manufacturing is drawing economic demand for hardware networking solutions that are fast, adaptable, and interoperable where existing proprietary fieldbus technologies have historically fallen short. Danielis et all [Dan+14] suggest that such a network should support up to 10,000 endpoints, while maintaining $< 10ms$ delivery times - and $< 1ms$ delivery times within branches of the network. In their survey of the field, they conclude that "the existing solutions will not fulfill the future challenges in terms of reliability, scalability, and flexibility as they are right now."

1.2 Switched Ethernet

The dominant technology in NCS is Switched Ethernet - in one study from 2007, 80% of companies surveyed were already using the technology [MT07]. Switched Ethernet is seen as the successor to technologies that were commonly referred to as Fieldbuses. These were typically proprietary systems that offered little interoperability between vendors [LLL06], and had little extensibility: adding new nodes to the network required re-engineering at a systems' top level, and increased wait times for other nodes on the network. Figure 2 showcases increasing wait times on an expanding fieldbus.

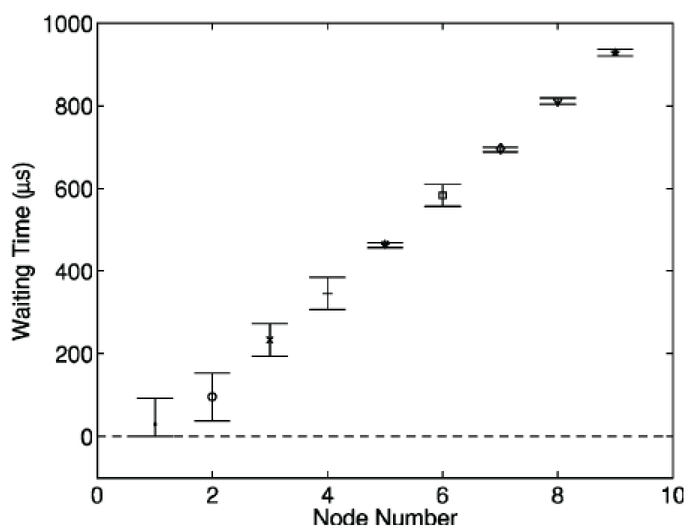

Figure 2: Wait times relative the number of nodes on a DeviceNet Fieldbus Network [MT07]

Switched Ethernet has been successful because it overcomes many of these issues. Because it is based on an open standard, vendor interoperability is easily achieved. It uses a split medium, so an arbitrary number of nodes can be added without collisions and wait times increasing, as more medium is also dynamically added to the network. Figure 3 charts packet delay with increasing traffic towards switched- and a hub- based Ethernet networks. This illustrates the difference between using split medium and shared medium: a switch disconnects the medium, and uses buffers to pass messages between them, which allows traffic on different sections of the medium to increase without causing additional delays on the rest of the network.

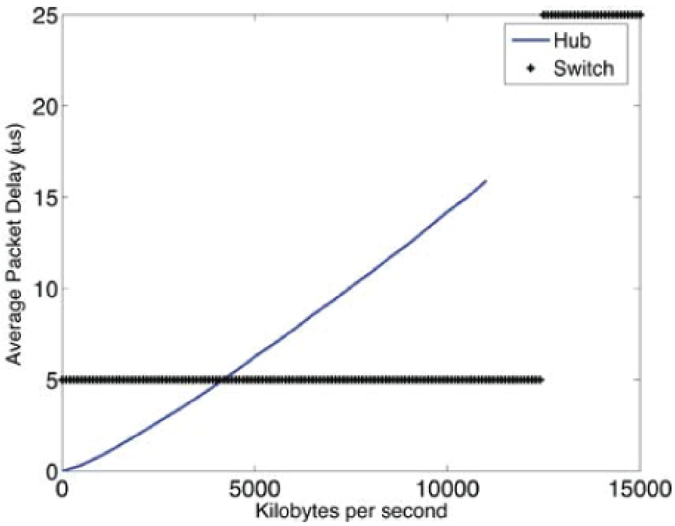

Figure 3: Per-packet delay as traffic increases on a hub vs. a switched Ethernet network.

1.3 Disadvantages of Switched Ethernet

However, Ethernet was not originally developed with control systems in mind, and there are aspects of its design that demand attention.

1. **Topology Limitations and Static Routing:**
   Ethernet switches use Broadcast packets to discover network topology, and occasionally to send packets to all devices on the network. When there are multiple possible routes between endpoints, broadcast packets can loop infinitely through switches, occupying 100% of the available medium. To overcome this, Ethernet implements a Spanning Tree Protocol to construct a network graph where there is only one route to every endpoint.

   A result of the Spanning Tree Protocol is that routes cannot be adapted in the face of heavy network traffic. In the example where a switch having four endpoints is receiving a frame from one of its endpoints, other nodes' frames are buffered at the switch, and must wait to transmit. This issue, which is largely responsible for NCS's indeterminacy (inability to ensure message delivery within a specific timeframe) is discussed by Moyne and Tilbury on p. 39 [MT07] and by Danielis et all on p. 2 [Dan+14].

2. **Packet Overhead:**
   Ethernet has a minimum packet size of 84 bytes (672 bits), consisting of an 8 byte preamble and frame delimiter, two 6 byte Addresses, a 2 byte Ethernet Type flag, a minimum 46 byte payload size, a 4 byte Cyclic Redundancy Check, and a 12 byte interframe gap[16]. In many controls applications, data sizes on control networks tend to be relatively small and extremely low latencies are preferred[MT07]. As a result, even though Ethernet features high bit-rates, a large percentage of useful time is occupied with unnecessary overhead.

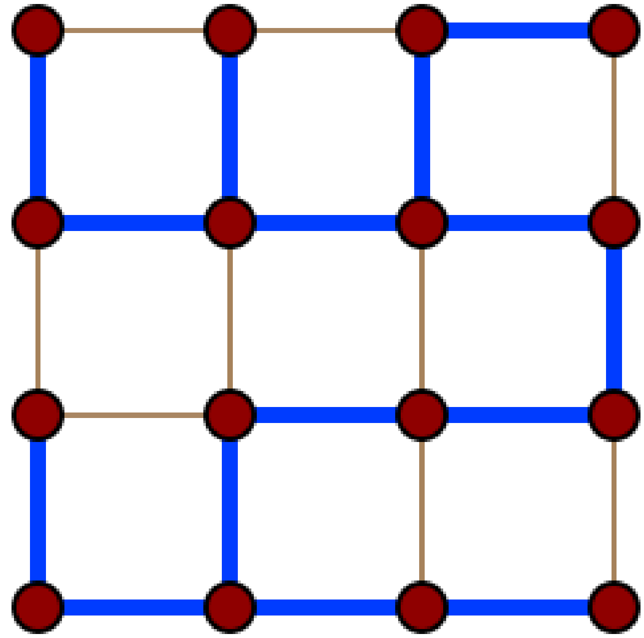

Figure 4: Example of a spanning tree, where active connections are highlighted against culled connections.

3. **Black Box Implementation and Hardware Size and Cost:**
   While Ethernet marks a drastic improvement over fieldbus technologies in terms of interoperability, its implementation involves purchasing specific chipsets from silicon manufacturers, and using large RJ45 Jacks that have not changed in size since 1988. This means that systems designers are unable to push networking hardware into smaller form factors as is often demanded in micro robotics and avionics systems. It also prevents further system integration, where network computing devices (switches and endpoints) cannot be modified to also perform application level work. We will propose a system architecture where network communication can be implemented on any given microprocessor, requiring only one UART Peripheral. This allows systems designers to implement a NCS with minimum hardware overhead and an easily included software stack.

2 Proposal

We propose a proof-of-concept network architecture that overcomes all three of Switched Ethernet's limitations as posed above and outlined in the referenced literature. We will develop a simulated network based on this architecture, and use the simulation to test network quality of service in a mixture of cross traffic conditions. We will implement the network protocol on ubiquitous micro-controllers, demonstrating the protocol's portability and interoperability between hardware, and will use results from physical experiments to inform our simulation.

2.1 Measurements and Metrics

As outlined in Table I from Danielis et all, the critical metric for Networked Control Systems is Message Delivery Time [Dan+14]. Delivery time depends on packet size and bitrate (the time it takes to send a message across the medium) as well as the time it takes to route between switches. This will be the touchstone metric for our network architecture.

We will measure delivery time between two nodes as the number of switches between those nodes increases, as explored by Lee, Lee and Lee in [LLL06] for Switched Ethernet. We will also measure delivery time between two nodes as cross-traffic between them increases, i.e. as the switches in their routing paths become busy. We will experiment with different routing procedures and switching protocols in order to maximize determinacy in the system.

Other metrics in Danielis et all's table include tokens for Reliability (whether or not the network has a singular point of failure), Scalability (how many devices can share the network), and Self Configurability. We will demonstrate that our network architecture has No Single Point of Failure (i.e. is robust to 'lost nodes'), that it is scalable (up to $2^1$6 devices, outpacing Switched Ethernet and surpassing Danielis et all's target of 10000) and that is self-configures.

2.2 Implementing Dynamic Routing

We will implement a packet-switching algorithm that uses feedback from other switches regarding their status (busy or clear) in order to intelligently route packets through the network. In order to stop flood packets from ringing through the network, we attach to the packet information regarding its own history, so that switches can also intelligently stop ringing flood packets. Intelligent multi-path routing will allow us to develop a network that has no single point of failure (messages can find alternate routes) and a network that is more determinant. When switches are busy, instead of waiting in a queue, messages can find alternate routes to their destination.

2.3 Reducing Packet Overhead

We will implement a network architecture that minimizes unnecessary overhead in the packet, minimizing message delivery time in the network.

2.4 Open Hardware and Software:

Where Ethernet requires specific ICs and plugs to implement, we us readily available GPIO pins available on almost all micro-controllers. This means that our network can be seamlessly integrated to small and micro-size robotics projects like that exhibited in pebble [GKR10].

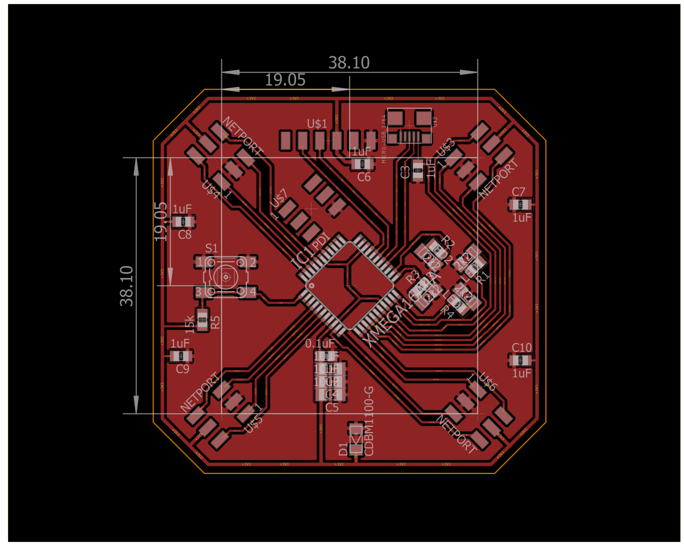

Figure 5: A prototype scale network switch, implemented on an XMega Microprocessor, $5 total hardware cost.

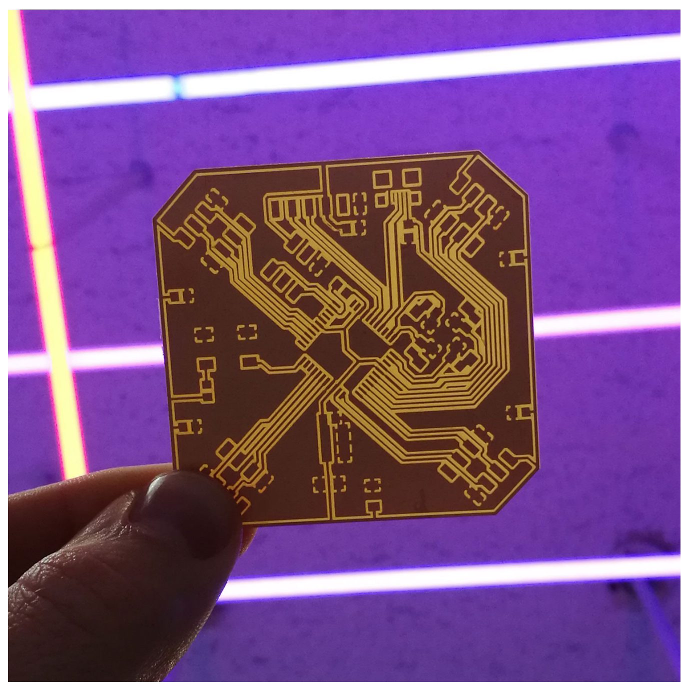

Figure 6: The board, fabbed.

3 Milestones / Schedule Outline

1. **October 12th**
   Hardware v0.1

   - Bring hardware and IDEs online.

2. **October 23rd**
   Project Report 1
   - Hardware v0.2
   - Measure peer-to-peer packet delay times
   - Measure packet delay times with increasing edge hops
   - Develop a Simulation Environment
3. **November 7th**
   Project Meetings
   - Demonstrate a small graph network with some version of addressing, packet protocol, and routing.
   - Use data from previous experiments to improve our simulation
4. **November 21st**
   Project Report 2
   - More Nodes, Bigger Graphs
   - Graph Discovery
   - Route Optimizations
   - Measure packet delay times in the face of increasing cross-traffic.
5. **November 30th**
   Work on hardware for demos.
6. **December 7th**
   Demo development and project documentation.
7. **December 12th**
   Project Presentations
8. **January 3rd**
   Paper Submissions to SIGCOMM

References

[LLL06] K. C. Lee, S. Lee, and M. H. Lee. "Worst Case Communication Delay of Real-Time Industrial Switched Ethernet With Multiple Levels". In: *IEEE Transactions on Industrial Electronics* 53.5 (2006), pp. 1669–1676. ISSN: 0278-0046. DOI: 10.1109/TIE.2006.881986.

[MT07] J. R. Moyne and D. M. Tilbury. "The Emergence of Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data". In: *Proceedings of the IEEE* 95.1 (2007), pp. 29–47. ISSN: 0018-9219. DOI: 10.1109/JPROC.2006.887325.

[GKR10] K. Gilpin, A. Knaian, and D. Rus. "Robot pebbles: One centimeter modules for programmable matter through self-disassembly". In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 2485–2492. DOI: 10.1109/ROBOT.2010.5509817.

[Dan+14] P. Danielis et al. "Survey on real-time communication via ethernet in industrial automation environments". In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 2014, pp. 1–8. DOI: 10.1109/ETFA.2014.7005074.

[16] "IEEE Standard for Ethernet". In: *IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012)* (Mar. 2016), pp. 1–4017. DOI: 10.1109/IEEESTD.2016.7428776.

[PF16] A. V. Proskurnikov and A. L. Fradkov. "Problems and methods of network control". In: *Automation and Remote Control* 77.10 (Oct. 2016), pp. 1711–1740. ISSN: 1608-3032. DOI: 10.1134/S0005117916100015. URL: https://doi.org/10.1134/S0005117916100015.