Axon: A Flexible Substrate for Source-routed Ethernet

Jeffrey Shafer
Brent Stephens
Michael Foss
Scott Rixner
Alan L. Cox
Ethernet Tradeoffs

**Strengths**
- Cheap
- Simple
- High data rate
- Ubiquitous

**Weaknesses**
- Loop-free forwarding topology – limits bandwidth
- Broadcasts and packet flooding for location discovery – limits scalability

Scale up Ethernet to work effectively in a modern datacenter?
Ethernet in the Datacenter

- Traditional solution:
  - Small Ethernet LANs + **IP routers**
  - Increases network complexity
  - Hinders *live* virtual machine migration

- Recent proposals
  - Many VLAN overlays (*see SPAIN*)
  - Re-writing MAC addresses to add hierarchy (*see PortLand or MOOSE*)
  - New non-broadcast location service (*see SEATTLE*)

Existing techniques keep Ethernet **frame format** 30 years old! Let’s **replace** it!
What is the Axon Device?

Take a traditional datacenter network

- Border Router
- Internet

Router

Switch

Server Rack

Server Rack

Server Rack

Server Rack

Server Rack
What is the Axon Device?

Replace all the interior network devices...
(Ethernet switches, IP routers)
What is the Axon Device?

... with an arbitrary graph of Axons
Axon Overview

Axons deploy a new datalink-layer protocol: **source-routed Ethernet**
- Full path placed in packet header
- Used internally between Axons (Axon↔Axon)
- Standard Ethernet PHYs

Axons maintain **compatibility with unmodified hosts**
- Abstraction of a single large subnet
- Traditional Ethernet used externally (Host↔Axon)
- Packets are transparently rewritten by Axons
Advantages of Source-routed Ethernet

- Flexibility in network **topology**
  - Support arbitrary paths, including **loops**!
  - In traditional Ethernet, STP disables redundant links (cannot carry data)

- Flexibility in **routing algorithms**
  - Shortest-path? Congestion-aware?

- Improved **scalability**
  - Each Axon only stores routes for locally-connected hosts
  - Interior Axons just follow route in packet header
  - In traditional switches/routers, a lookup must be performed at every hop along the path
1. Introduction

2. Design Overview
   - Source-routed Ethernet
   - Compatibility with Existing Hosts

3. Evaluation
   - Hardware Prototype
   - Software Simulator
Traditional Ethernet

<table>
<thead>
<tr>
<th>Bytes:</th>
<th>(6)</th>
<th>(6)</th>
<th>(2)</th>
<th>(Varies)</th>
<th>(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest MAC Addr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Src MAC Addr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type/Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Forwarding**

- At each hop, must lookup destination address in a forwarding table to obtain output port (CAM lookup)

In contrast, *source-routed Ethernet* has a new header containing the full path list. How to obtain transparent compatibility?
Axons present illusion that entire network is simply a large Ethernet segment

Host A wishes to communicate with Host Z
Host A issues ARP request to locate Host Z
Axon A intercepts broadcast ARP request
Axon A begins establishing route with Axon Z
Axon Z sends ARP request to Host Z
Host Z responds with ARP reply (captured by Axon Z)
Axon Z installs route to Host A
Axon A installs route to Host Z
Axon A issues ARP reply
Host A sends data to Host Z

Axon A looks up route and encapsulates data for transport

Source routing used internally (Axon↔Axon)

Data unpacked for delivery
Packet header contains two routes:
- Forward route from current Axon to destination
  - Grows shorter at each hop
- Reverse route from current Axon to source
  - Grows longer at each hop
- Each 1-byte route item specifies an output port

Forwarding
- At each hop, read header to obtain next output port
- Prepend arrival port to reverse route header

Works with standard Ethernet PHYs and MACs by using jumbo frames
Route Generation

- Generate a route on the first ARP for flow
- Cache at local Axon for subsequent packets
- Prototype design
  - Central route controller with full topology knowledge
    - Inspired by Ethane and Tesseract projects
    - Could also implement a distributed mechanism
  - Routing algorithm: Shortest-path or congestion aware
- Key point: source routing allows for arbitrary topologies, arbitrary paths (including loops), and arbitrary routing algorithms

Casado et. al., Ethane: taking control of the enterprise, SIGCOMM’2007
Yan et. al, Tesseract: A 4D network control plane, NSDI’2007
1. Introduction

2. Design Overview
   - Source-routed Ethernet
   - Compatibility with Existing Hosts

3. Evaluation
   - Hardware Prototype
     - Measure performance
     - Demonstrate compatibility
   - Software Simulator
Hardware Prototype

- **Data plane**
  - 4-port NetFPGA
  - Custom verilog
  - Packet forwarding and translation

- **Control plane**
  - Intel Atom processor on mini-ITX board
  - Linux + application program
Test Networks

Line Topology:

Host  
Axon  
Host  
Axon  
Host

Ring Topology:

Host  
Axon  
Host  
Axon  
Host

(Can’t build with conventional Ethernet!)
Higher Bandwidth

- Test setup: Used both ring and line topology
  - 1 TCP or UDP flow from each host to a host on a different Axon

- Measured aggregate bandwidth (Mbit/s)

<table>
<thead>
<tr>
<th></th>
<th>UDP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Line</td>
<td>Line</td>
</tr>
<tr>
<td>Ring</td>
<td>2906</td>
<td>2425</td>
</tr>
<tr>
<td></td>
<td>5690</td>
<td>3951</td>
</tr>
</tbody>
</table>

Shows bandwidth benefit of using redundant links
Lower Latency

- Measured forwarding latency
  
<table>
<thead>
<tr>
<th>Axon ↔ Axon</th>
<th>Axon → Host</th>
<th>Host → Axon</th>
</tr>
</thead>
<tbody>
<tr>
<td>520ns</td>
<td>520ns</td>
<td>720ns</td>
</tr>
</tbody>
</table>

- Compares favorably against gigabit Ethernet switch
  - 7-28us per hop (varies with packet size)

- Latency advantage in Axon design
  - Cut-through forwarding instead of store-and-forward
  - Forwarding table lookup only at first hop (to obtain route)
  - Traditional Ethernet switches do lookup at every hop
Lower Latency in Applications

- Test setup
  - PostMark benchmark
  - Line topology with Axons or switches
- Each Axon adds a smaller per-hop latency compared to an Ethernet switch
- Only first Axon does a route lookup
Host Compatibility

- Demonstrated compatibility with unmodified hosts
- Windows, Mac OS X, FreeBSD, Linux, Netgear switch, Cisco IP router, Linksys wireless access point, ...
1. Introduction

2. Design Overview
   - Source-routed Ethernet
   - Compatibility with Existing Hosts

3. Evaluation
   - Hardware Prototype
   - Software Simulator
     - Evaluate design at large scales and arbitrary topologies
Custom software simulator

- Simulated Axons, hosts, and links
- Based on prototype
  - Each simulated Axon runs same control software
  - Each simulated host represented by ARP generator
    - ARPs from host trigger route generation, which is the overhead we are most concerned about
Lower Control Overhead

- Characterize **overhead bandwidth** used for Axon **control**
  - Network topology maintenance (discovery and heartbeat messages)
  - Route generation and dissemination

- Simulator Setup
  - Topologies: Torus, Fat tree, Flattened-butterfly, Random
  - Up to 50,000 hosts and 5,000 Axons
  - Each host generates 10 ARPs/sec (new flows only!)
    - Conservative choice compared to peak of 0.5 ARPs/sec reported in Ethane network and LBNL trace
Showing torus topology

- Max link has highest overhead
  - Attached to central controller
- Average Axon link has less overhead than average Ethernet link
  - ARPs not broadcast
- Torus is worst case topology for Axons
  - Highest average distance from controller

![Graph showing Mbps per Link vs Number of Hosts]

- Max Axon Overhead
- Avg Ethernet Overhead
- Avg Axon Overhead
Overhead Comparison

- Compared against PortLand architecture
  - Fat tree topology

- Axon host discovery protocol more efficient

- Very similar average link overhead to Axons once PortLand has warmed up
  - Axon packets are slightly larger due to source routes

Flexible Route Selection

Implemented **weighted shortest path routing** in central controller (similar to SPAIN)
- Weight is number of flows across a link
- Disperses flows across many links (congestion avoidance!)

Demonstrated Axon flexibility to easily support alternate route selection algorithms

Results
- Average route length increases by 0.1 hops
- Busiest link (measured by flow count) has the number of flows cut in half!

Mudigonda et. al., SPAIN: COTS data-center Ethernet for multipathing over arbitrary topologies, NSDI’2010.
Source-routed Ethernet is **flexible**
- Supports arbitrary topologies and routing algorithms

Axons unlock this flexibility for existing hosts
- Abstraction – giant Ethernet segment (flat IP address space)
- Migrate a VM from any point to any point in the entire network
- Transparent packet rewriting

FPGA prototype demonstrated design is simple and practical

Simulator demonstrated reasonable control overhead for real-world network sizes
- Control overhead on a 50,000 host network is only 0.25% of total link bandwidth
Lower Control Overhead

- Showing average overhead for all topologies
- Torus has highest average distance from controller
  - Thus highest overhead
- Even the torus was a significant win over conventional Ethernet
Byte Overhead of Source-Routed Ethernet
Evaluation – Memory Requirements

- How large of a CAM does each Axon need to support all locally-attached hosts?

- Worst-case scenario
  - Axon attached to the border router (to reach public Internet) must have routes to all internal hosts with an active flow

- Best-case scenario
  - Core Axons – no attached hosts at all!

- Wrote custom trace analyzer to measure re-use distance between messages to the same destination IP address
Evaluation – Memory Requirements

- Traces examined
  - LBNL
  - NCAR-I
  - CESCA-I
  - Link connecting scientific ring to public Internet

- 4k CAM entries sufficient
  - Commercial switches already have 8k+ entries

- Many datacenter flows will be internal (and thus avoid the worst-case Axon)
Axon Compatibility

- The first Axon (connected to a sending host) has several functions
  - Intercept ARP and DHCP packets
  - Transparently rewrite packet from traditional to source Ethernet format

- Interior Axons just follow route in packet header

- The last Axon (connected to a receiving host) transparently rewrites packet back to traditional Ethernet format