The Beacon OpenFlow Controller

David Erickson

Stanford PhD Candidate

Agenda

- Motivation
- Design Space
- Beacon
- Questions

Motivation

- Back to circa 2008-2009
- The controller world == NOX
  - Single threaded event based C++ with SWIG glue to Python
  - Enabled great research and demos (Thanks Nicira!)
- Python apps interesting with <1k LOC
  - But could have language level runtime errors
  - SWIG hard to use, needed to expose C++ code to Python
  - Much slower than C++
- C++ only apps
  - Solved the Python app problems
  - Cryptic compilation errors (STL, templates, etc)
  - Runtime segfaults and other memory related issues
- **I wanted to spend more time on interesting new features and less time fighting platform and/or language related challenges**

Motivation cont…

- Was it possible to build a controller with
  - Rapid or no compilation time
    - Human readable errors/warnings
  - Reduced scope of runtime errors
    - Managed code
    - Static type checking
  - High developer productivity
    - Mature toolchains
    - Code generation/auto complete
  - Cross platform
  - Performant
    - Within 50% of a fast C++ implementation
- Note, these were primarily language questions

Design space

- Candidate language: Java
  - No existing OpenFlow protocol bindings
  - Performance?
- Early basic tests with OpenFlowJ
  - Object Oriented OpenFlow 1.0 protocol library
  - Simple sample hub/switch controller

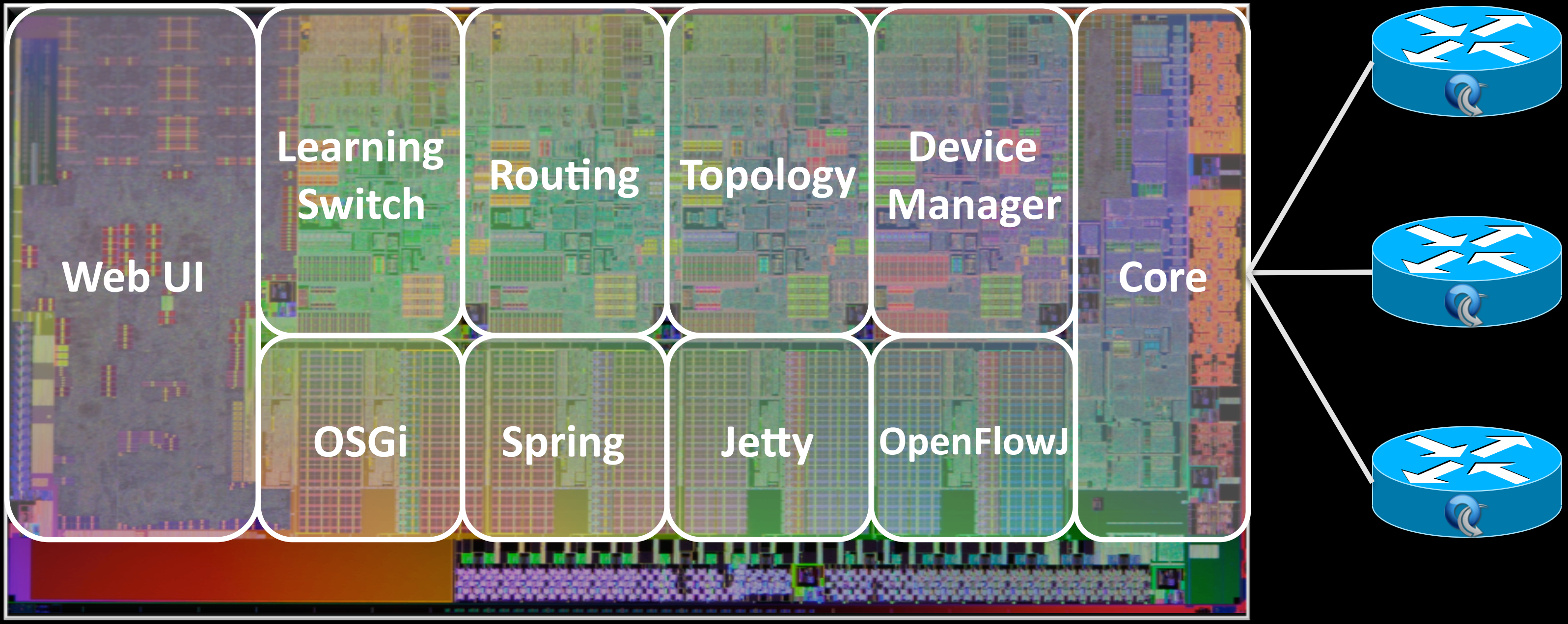
• NOX pyswitch	9369
• Simple Python Controller	21019
• NOX hub (C++)	124,897
• Reference Hub (C++)	214,591
• Java (1 thread)	252,246
• Java (2 threads)	287,567
• Java (4 threads)	348,762

Design space cont…

- Other desirable controller features
  - Fully multi-threaded
  - Build time modular
  - Run time modular
  - Easy to use and understand abstractions
  - Use existing familiar and well documented frameworks
  - Extensible Web UI and REST capabilities
- Possible to modify existing software?
  - Considered Tomcat

Beacon

- “Die Shot”



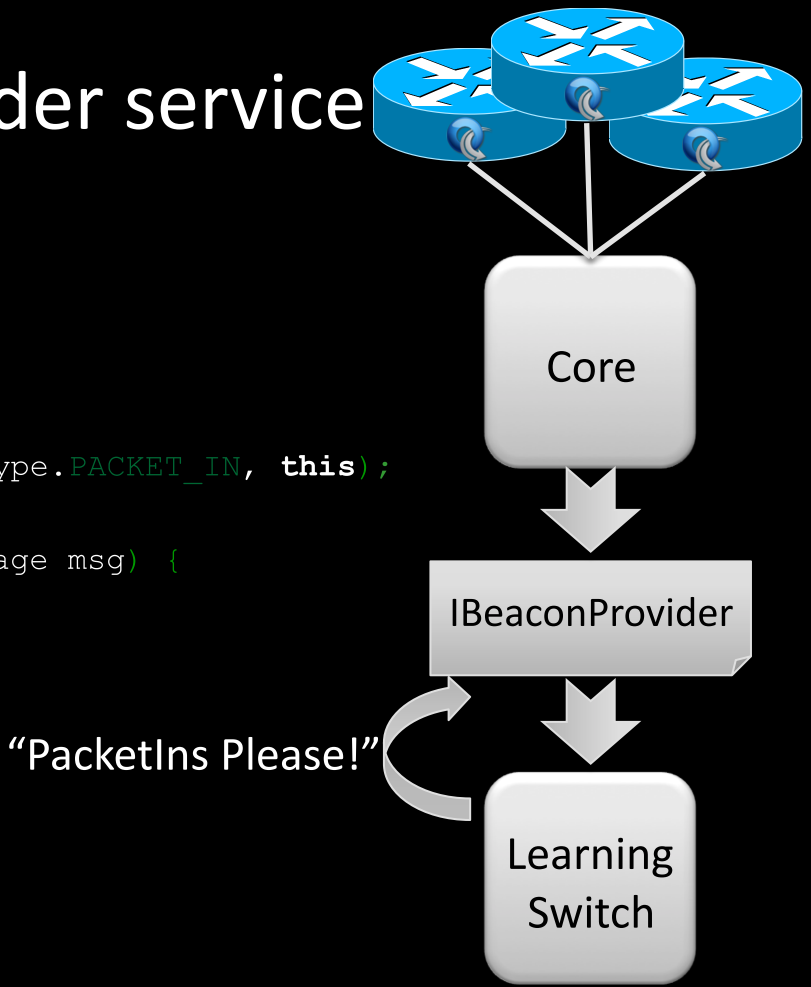
*Size not indicative of anything

How does the core work?

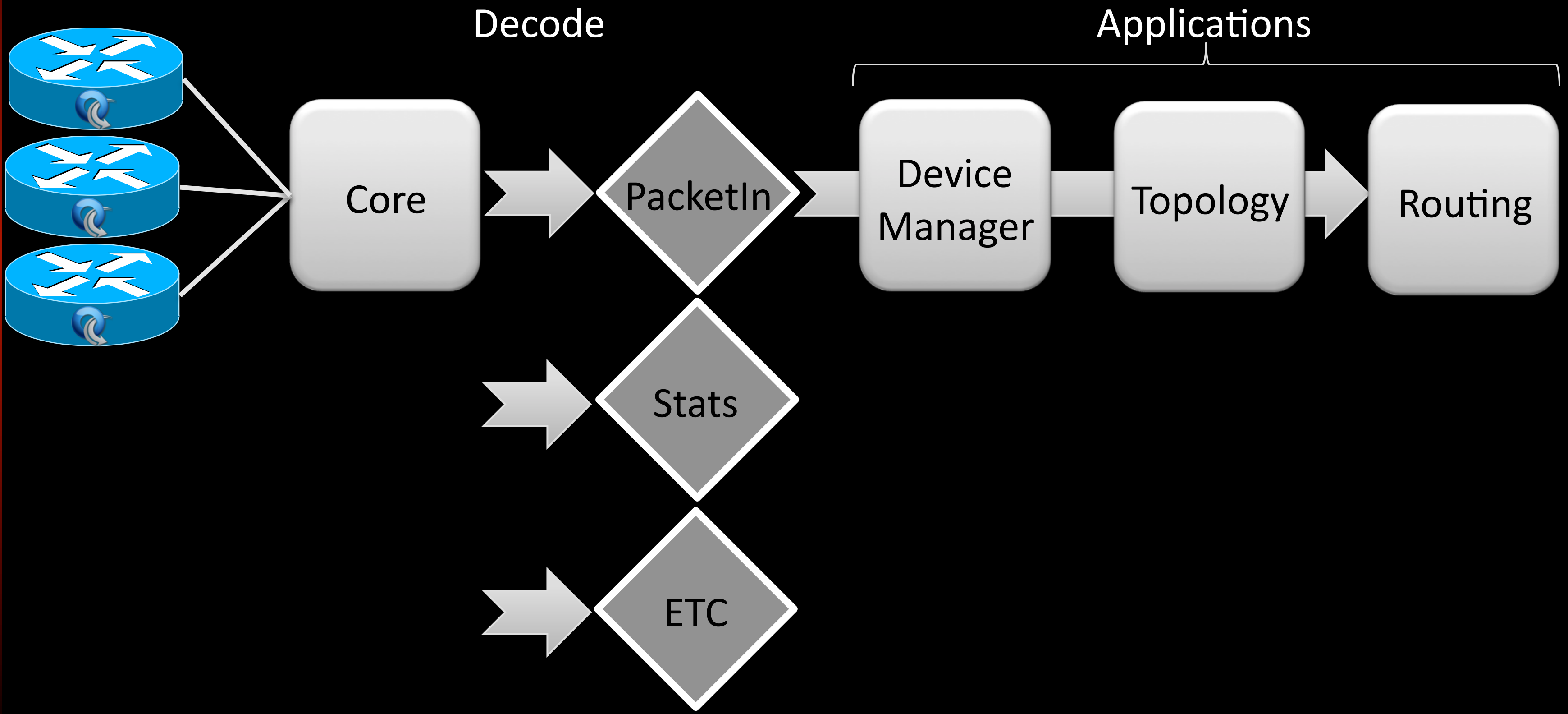
- Connects to switches
- Publishes IBeaconProvider service
- Other bundles use IBP

```
protected IBeaconProvider beaconProvider;

public void startUp() {
    beaconProvider.addOFMessageListener(OFType.PACKET_IN, this);
}

public Command receive(IOFSwitch sw, OFMessage msg) {
        OFPacketIn pi = (OFPacketIn) msg;
        …

        return Command.CONTINUE;
}
```

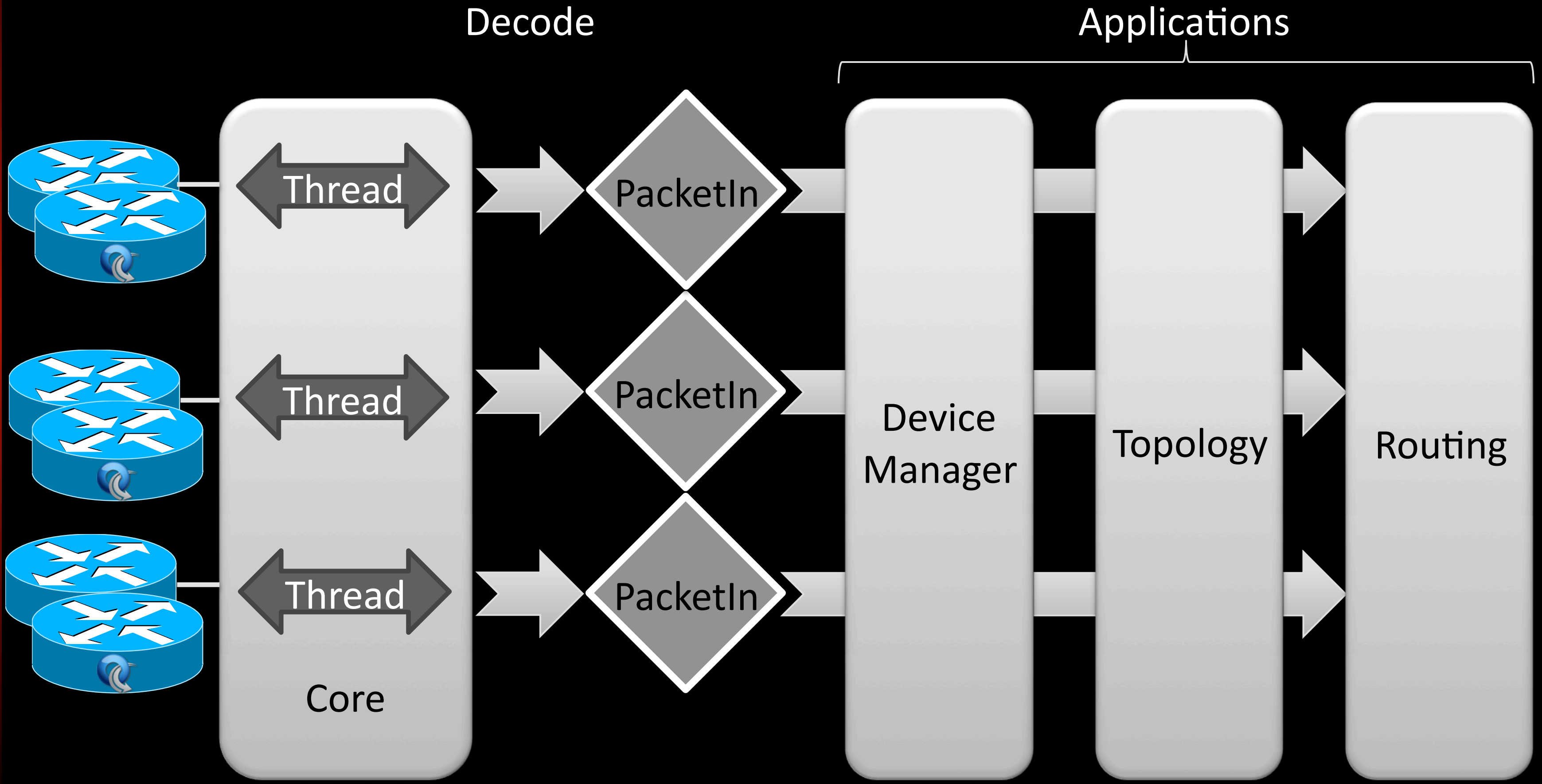
- Creates a pipeline…



Pipeline


Decode
Applications
Core
PacketIn
Stats
ETC
Device Manager
Topology
Routing

Fully Multithreaded



- Each app gets OFMessages from all threads

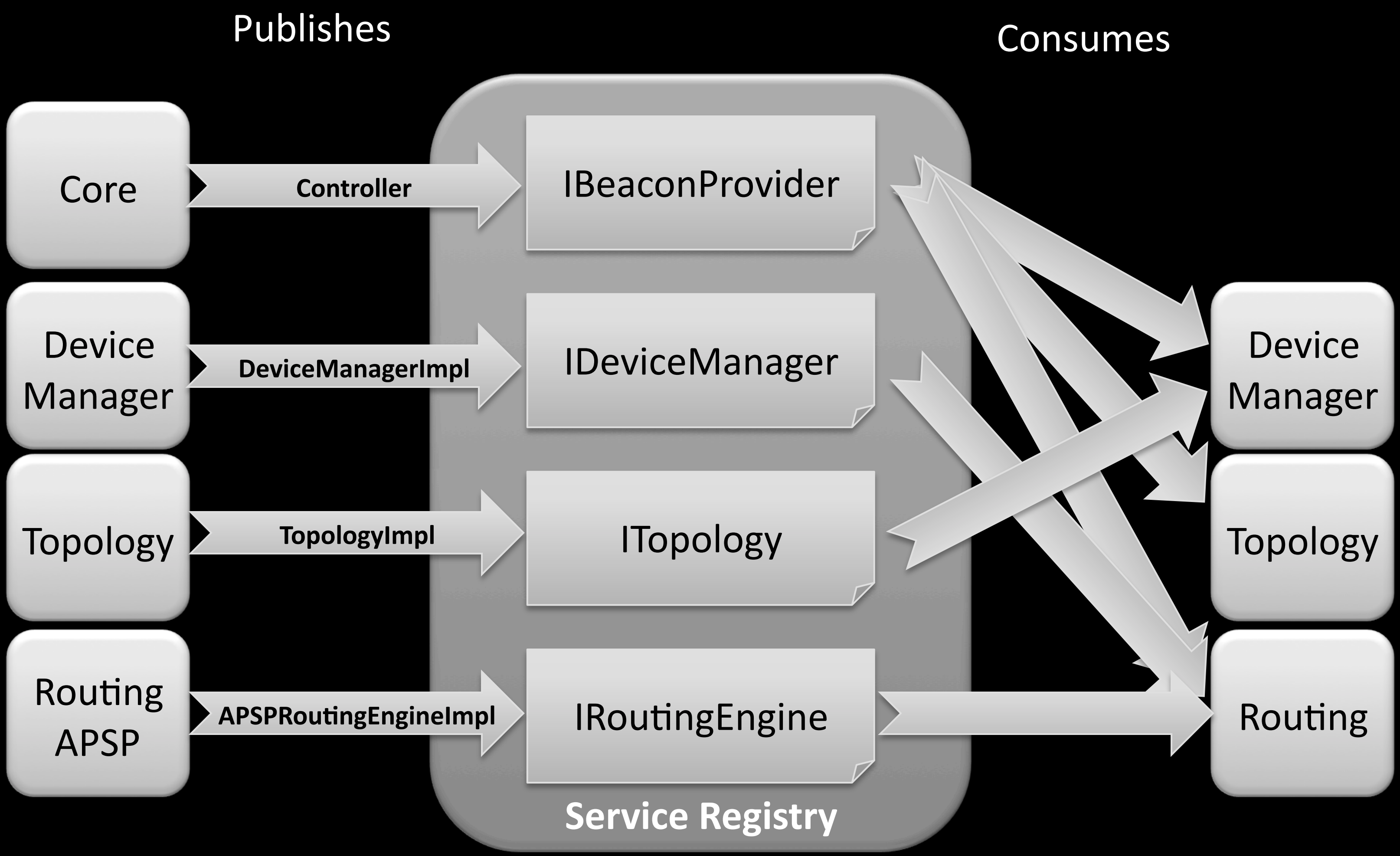
How do Bundles interact?

- Service abstraction
- Create an interface for service contract

```
public interface ITopology {
   /**
    * Retrieves a map of all known link connections between OpenFlow switches
    * and the last time each link was known to be functioning
    * @return
    */
   public Map<LinkTuple, Long> getLinks();
}
```

- Export an object instance that implements the interface to the service registry
- Other bundles’ objects import and use services
- Enables easy service extension

Service Registry Example



Service Examples

- Queryable
  - "Give me a list of all connected switches"
- Explicit Event Registration
  - "Add me as a listener for OFPacketIn messages"
  - "Notify me when switches connect/disconnect"
- Implicit Event Registration
  - Export an *Aware service interface, consuming services post relevant events to all implementers
  - ITopologyAware, all implementers receive link updates

What Bundles are available?

- Beacon centric
  - OpenFlowJ (OF 1.0 Protocol)
  - Packet encoder/decoder (Ethernet, ARP, IPv4, LLDP, TCP, UDP)
  - Core, Learning Switch, Hub, Device Manager
  - Topology, Layer 2 Shortest Path Routing
  - ARP Proxy, DHCP Proxy, Multicast eliminator
  - Declarative routing (upload a text file)
  - Web UI
- Third party, basically anything
  - Just a JAR file with Metadata
  - Some may need YOU to generate the Metadata
  - Logging, Web Server, JSON parsing, Web framework, etc

Is there a NIB?

- Decentralized
  - Relevant bundles store the data and export query and event interfaces
- Currently soft-state only
  - Persistence engines can be added to extend existing capabilities

Why Bundles?

- Unit of modularity in OSGi
- Basic Building Block
- JAR (zipfile)
- May Contain
  - Metadata*
    - META-INF/MANIFEST.MF
  - Java Classes
  - Resources (xml, etc)
  - Other JAR files

* Required

What can Bundles do?

- Share code with other packages
  - Export-Package: net.beaconcontroller.core
- Consume other Java Packages
  - Import-Package: org.openflow.protocol
- Extend other Bundles
  - Fragments
- Run Code

```
public void start() {
 listenSock = ServerSocketChannel.open();
 new Thread(...)
 ...
}
```

Advanced Bundles

- Dynamic
  - Stop, Start, Install, Replace while running
- Versioned
  - Can have multiple versions live simultaneously
- Explicit Dependencies
  - State which version(s) you need

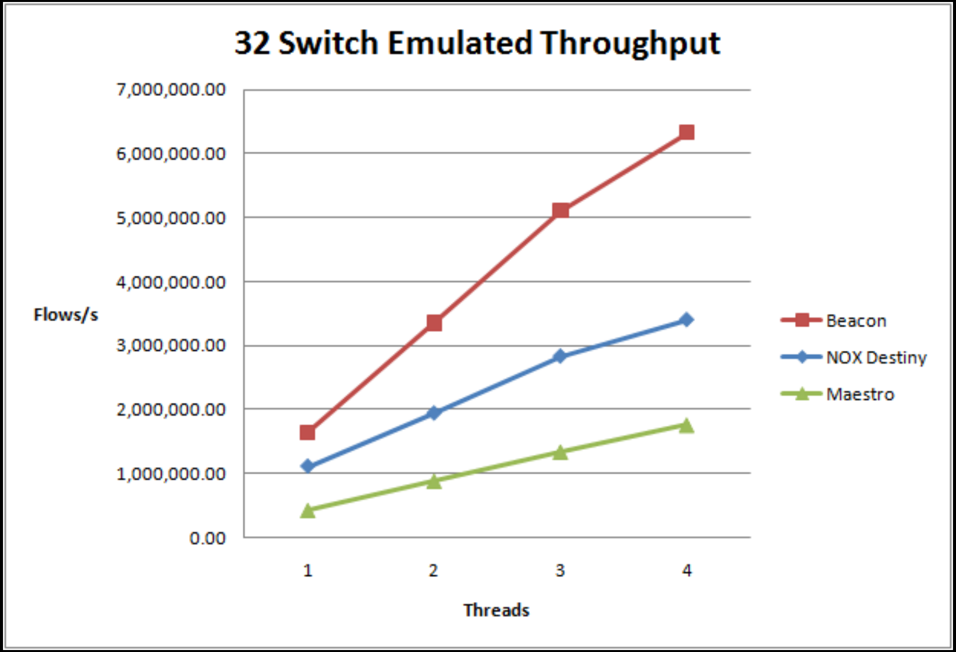
Performance

- Measured June 2011

Cbench Test, part of Oflops suite
- PacketIn to PacketOut/FlowMod throughput test, fills controller input buffers
- 10 loops, 32 switches, 10s per loop

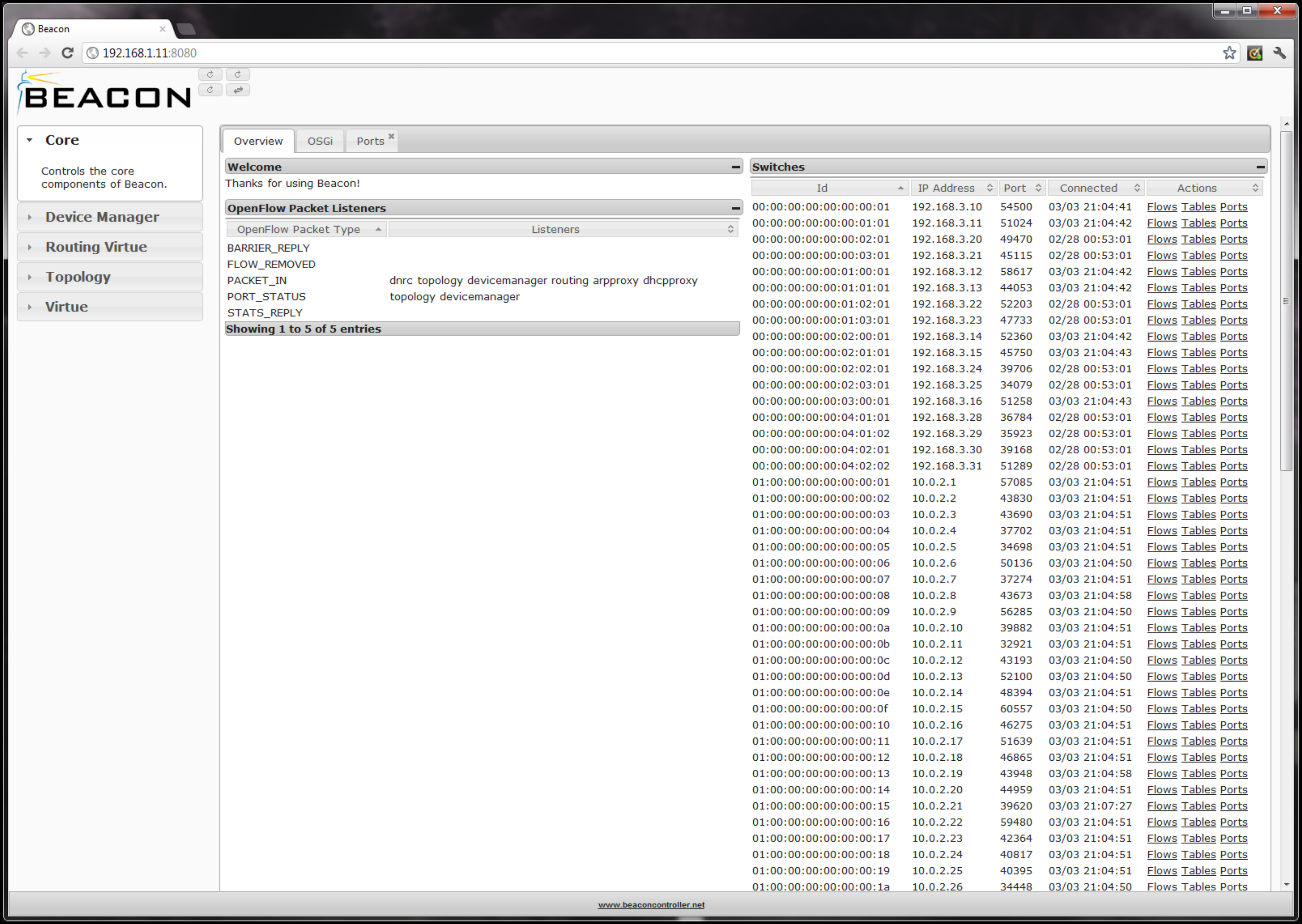
Test Machine
- CPU: 1 x Intel Core i7 930 @ 3.33ghz, 9GB RAM, Ubuntu 10.04.1 x64

Controllers
- Beacon, NOX (Destiny branch), Maestro



http://www.openflow.org/wk/index.php/Controller_Performance_Comparisons

Web UI



Beacon

192.168.1.11:8080

BEACON

Controls the core components of Beacon.

Welcome

Thanks for using Beacon!

OpenFlow Packet Listeners

OpenFlow Packet Type	Listeners
BARRIER_REPLY	
FLOW_REMOVED	
PACKET_IN	dnrc topology devicemanager routing arpproxy dhcpproxy
PORT_STATUS	topology devicemanager
STATS_REPLY	

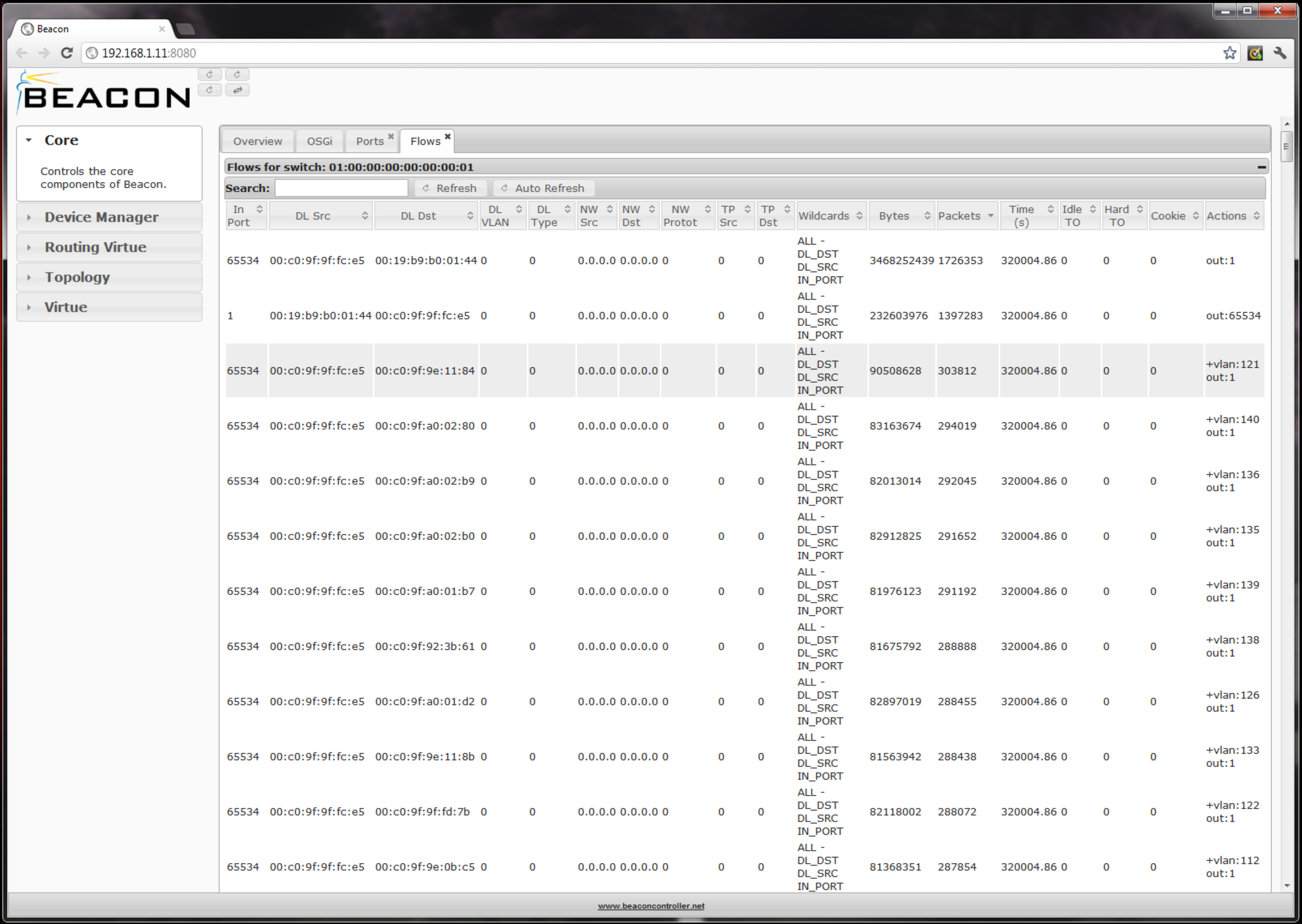
Showing 1 to 5 of 5 entries

Switches

Id	IP Address	Port	Connected	Actions
00:00:00:00:00:00:00:01	192.168.3.10	54500	03/03 21:04:41	Flows Tables Ports
00:00:00:00:00:00:01:01	192.168.3.11	51024	03/03 21:04:42	Flows Tables Ports
00:00:00:00:00:00:02:01	192.168.3.20	49470	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:00:03:01	192.168.3.21	45115	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:01:00:01	192.168.3.12	58617	03/03 21:04:42	Flows Tables Ports
00:00:00:00:00:01:01:01	192.168.3.13	44053	03/03 21:04:42	Flows Tables Ports
00:00:00:00:00:01:02:01	192.168.3.22	52203	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:01:03:01	192.168.3.23	47733	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:02:00:01	192.168.3.14	52360	03/03 21:04:42	Flows Tables Ports
00:00:00:00:00:02:01:01	192.168.3.15	45750	03/03 21:04:43	Flows Tables Ports
00:00:00:00:00:02:02:01	192.168.3.24	39706	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:02:03:01	192.168.3.25	34079	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:03:00:01	192.168.3.16	51258	03/03 21:04:43	Flows Tables Ports
00:00:00:00:00:04:01:01	192.168.3.28	36784	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:04:01:02	192.168.3.29	35923	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:04:02:01	192.168.3.30	39168	02/28 00:53:01	Flows Tables Ports
00:00:00:00:00:04:02:02	192.168.3.31	51289	02/28 00:53:01	Flows Tables Ports
01:00:00:00:00:00:00:01	10.0.2.1	57085	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:02	10.0.2.2	43830	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:03	10.0.2.3	43690	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:04	10.0.2.4	37702	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:05	10.0.2.5	34698	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:06	10.0.2.6	50136	03/03 21:04:50	Flows Tables Ports
01:00:00:00:00:00:00:07	10.0.2.7	37274	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:08	10.0.2.8	43673	03/03 21:04:58	Flows Tables Ports
01:00:00:00:00:00:00:09	10.0.2.9	56285	03/03 21:04:50	Flows Tables Ports
01:00:00:00:00:00:00:0a	10.0.2.10	39882	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:0b	10.0.2.11	32921	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:0c	10.0.2.12	43193	03/03 21:04:50	Flows Tables Ports
01:00:00:00:00:00:00:0d	10.0.2.13	52100	03/03 21:04:50	Flows Tables Ports
01:00:00:00:00:00:00:0e	10.0.2.14	48394	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:0f	10.0.2.15	60557	03/03 21:04:50	Flows Tables Ports
01:00:00:00:00:00:00:10	10.0.2.16	46275	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:11	10.0.2.17	51639	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:12	10.0.2.18	46865	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:13	10.0.2.19	43948	03/03 21:04:58	Flows Tables Ports
01:00:00:00:00:00:00:14	10.0.2.20	44959	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:15	10.0.2.21	39620	03/03 21:07:27	Flows Tables Ports
01:00:00:00:00:00:00:16	10.0.2.22	59480	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:17	10.0.2.23	42364	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:18	10.0.2.24	40817	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:19	10.0.2.25	40395	03/03 21:04:51	Flows Tables Ports
01:00:00:00:00:00:00:1a	10.0.2.26	34448	03/03 21:04:50	Flows Tables Ports

www.beaconcontroller.net

Web UI



Beacon

192.168.1.11:8080

BEACON

Core

Controls the core components of Beacon.

Device Manager

Routing Virtue

Topology

Virtue

Overview | OSGi | Ports | Flows

Flows for switch: 01:00:00:00:00:00:00:01

Search: Refresh | Auto Refresh

In Port	DL Src	DL Dst	DL VLAN	DL Type	NW Src	NW Dst	NW Protot	TP Src	TP Dst	Wildcards	Bytes	Packets	Time (s)	Idle TO	Hard TO	Cookie	Actions
65534	00:c0:9f:9f:fc:e5	00:19:b9:b0:01:44	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	3468252439	1726353	320004.86	0	0	0	out:1
1	00:19:b9:b0:01:44	00:c0:9f:9f:fc:e5	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	232603976	1397283	320004.86	0	0	0	out:65534
65534	00:c0:9f:9f:fc:e5	00:c0:9f:9e:11:84	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	90508628	303812	320004.86	0	0	0	+vlan:121 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:a0:02:80	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	83163674	294019	320004.86	0	0	0	+vlan:140 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:a0:02:b9	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	82013014	292045	320004.86	0	0	0	+vlan:136 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:a0:02:b0	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	82912825	291652	320004.86	0	0	0	+vlan:135 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:a0:01:b7	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	81976123	291192	320004.86	0	0	0	+vlan:139 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:92:3b:61	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	81675792	288888	320004.86	0	0	0	+vlan:138 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:a0:01:d2	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	82897019	288455	320004.86	0	0	0	+vlan:126 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:9e:11:8b	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	81563942	288438	320004.86	0	0	0	+vlan:133 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:9f:fd:7b	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	82118002	288072	320004.86	0	0	0	+vlan:122 out:1
65534	00:c0:9f:9f:fc:e5	00:c0:9f:9e:0b:c5	0	0	0.0.0.0	0.0.0.0	0	0	0	ALL - DL_DST DL_SRC IN_PORT	81368351	287854	320004.86	0	0	0	+vlan:112 out:1

www.beaconcontroller.net

Status

- 2010 April – 2011 September
  - Incubation and internal use
  - Limited external releases
- 2011 September 12
  - v1.0.0 Release
- Since
  - Ongoing active development
  - Accepting feature requests/suggestions/bug reports!
- Active user forum
- Many screencasts and guides available

Users?

- My research
  - Full time cluster of 80 machines
  - 97 switches (including vSwitches)
- Inside Big Switch Controller
  - Basis for Floodlight
- CS244 Stanford Graduate Networking course 2011
- FlowScale – load balancing

Lessons learned

- Met design goals
  - Productivity++
- Runtime dynamism does have a cost
- Seemingly minor changes can kill performance
  - 32 vs 64 bit
  - Spring proxies in the fast path
- Wide variety of I/O loop designs
  - With a correspondingly wide variety of fairness and performance consequences

Tutorial

- Unzip tutorial archive
- Launch Eclipse (eclipse subfolder)
  - File -> Import -> General -> Existing Projects into Workspace, Next
  - Point it to the src/ directory, Select All, Finish
- Follow tutorial instructions
  - http://goo.gl/Isuks :)

Agenda

- ~~Motivation~~
- ~~Design Space~~
- ~~Beacon~~
- Questions

Thanks!

davidERICKSON@cs.stanford.edu
http://www.beaconcontroller.net/