

LOOM

An Open Flow Controller

Marc Sugiyama, Erlang Solutions

Problem

Build an extensible, robust OpenFlow Controller for thousands of switches

Why?

- Explosion in the number of IP devices
- Virtual Hosts, Mobile
- Internet of things: Sensors, Body Monitors, Cars, Lightbulbs
- “The network is the computer” - John Gage

OpenFlow

Separates the Controller...



... from the Switching hardware

Together they do what traditional ethernet switches can do.

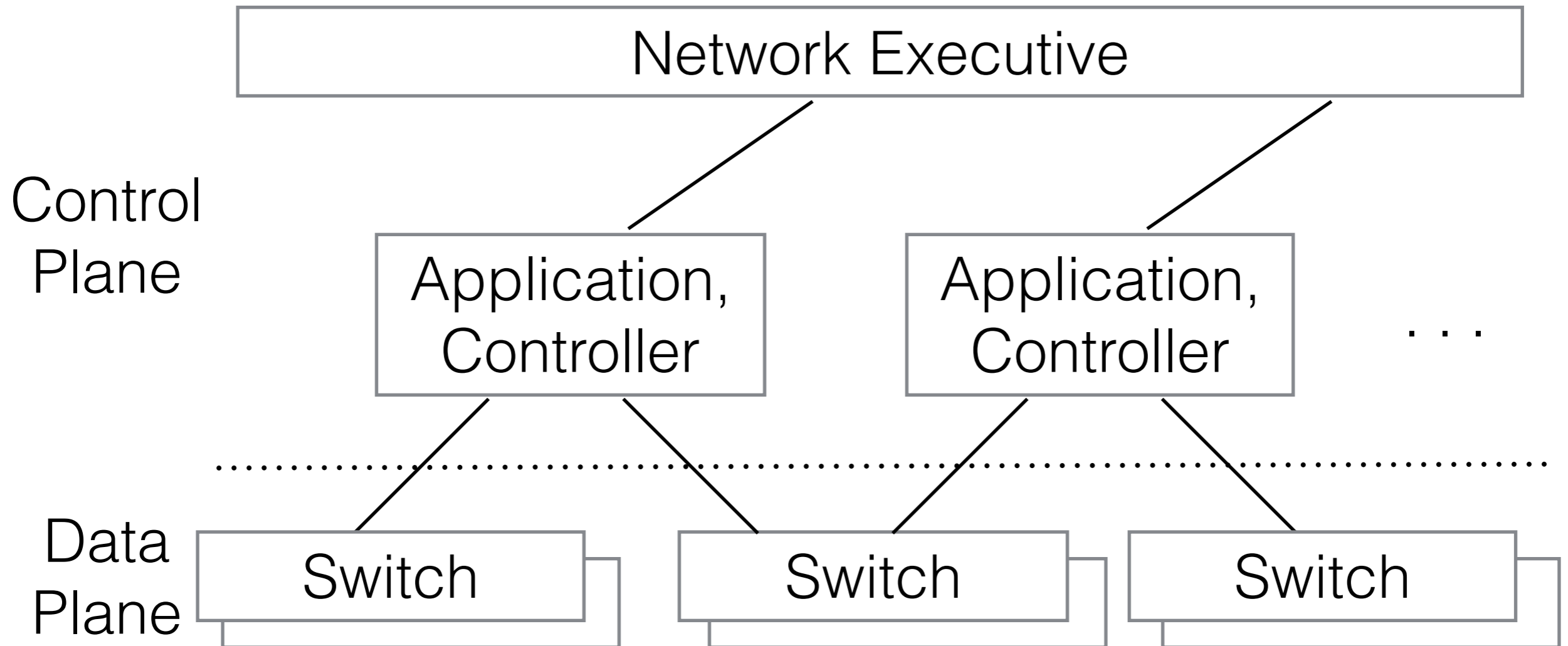


What is LOOM?

- Large scale OpenFlow controller
- Global view, deliberate, network aware applications
- Not a traditional IP switch controller
- Open Source

<https://github.com/FlowForwarding/loom>

LOOM Architecture



Where's the code?

- <http://github.com/FlowForwarding>
- Libraries: of_protocol, of_msg_lib, of_driver, ofs_handler, enetconf, of_config, ofs_config
- Examples: icontrol, tapestry, stats_poller

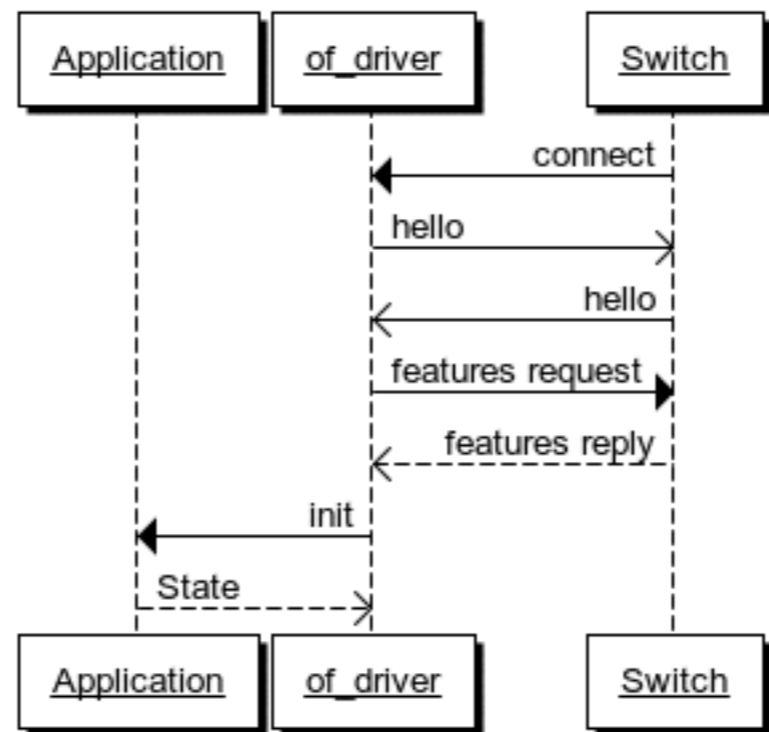
Getting Started

Custom Controller
ofs_handler
of_driver
of_msg_lib
of_protocol

Custom Config Point
ofs_config
enetconf
of_config

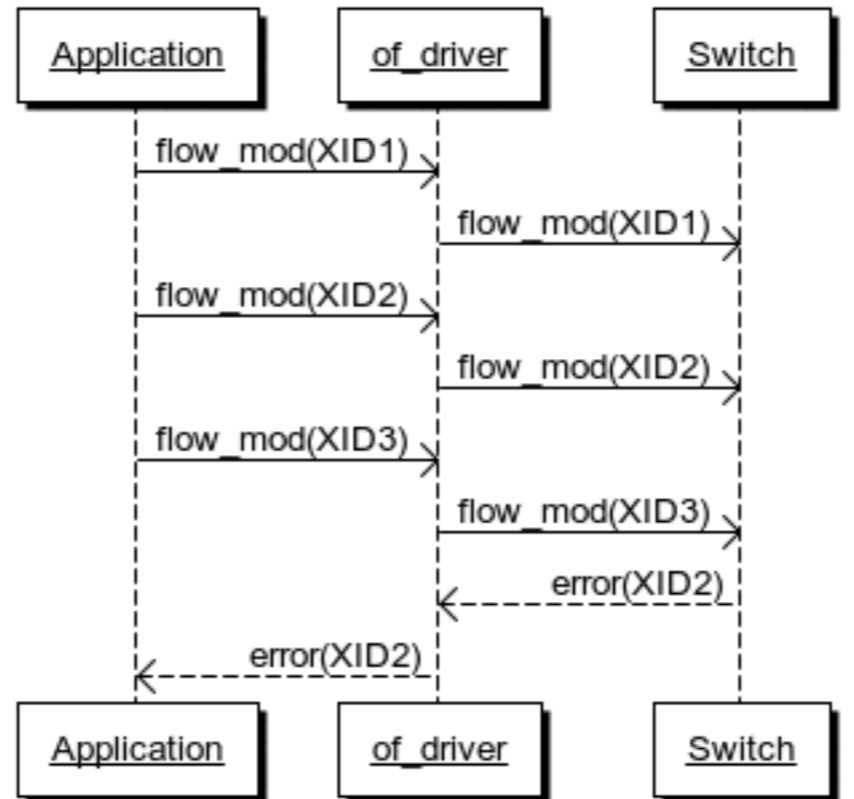
Connection

Protocol Flow



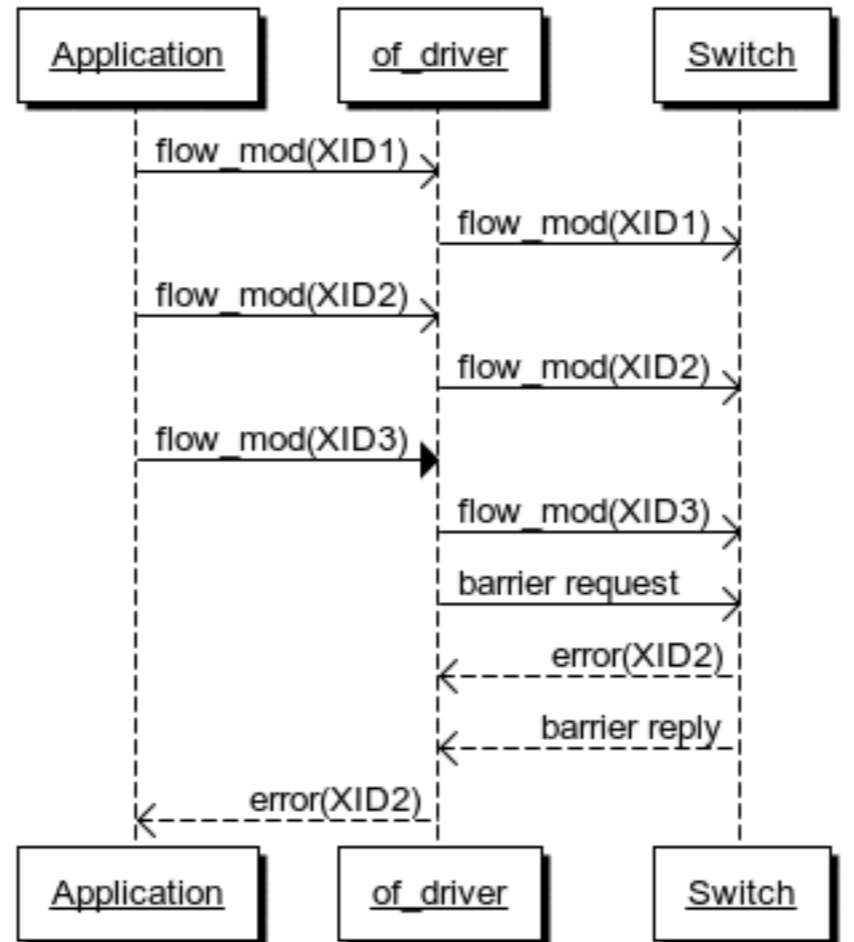
Asynchronous Send

Protocol Flow



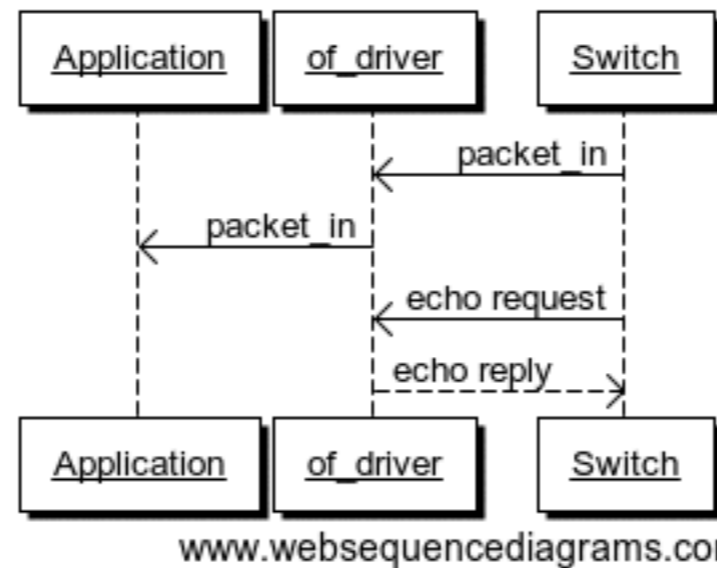
Synchronizing Send

Protocol Flow

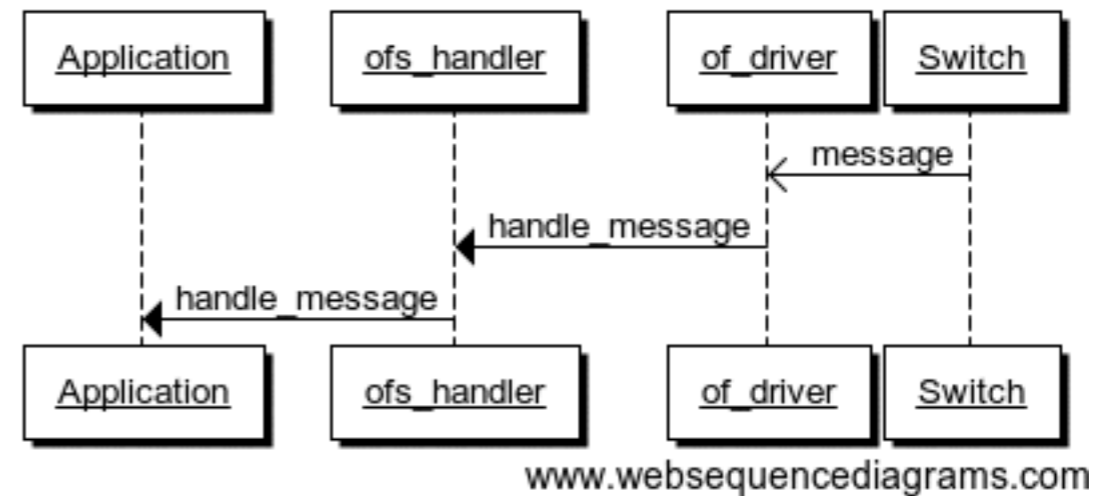
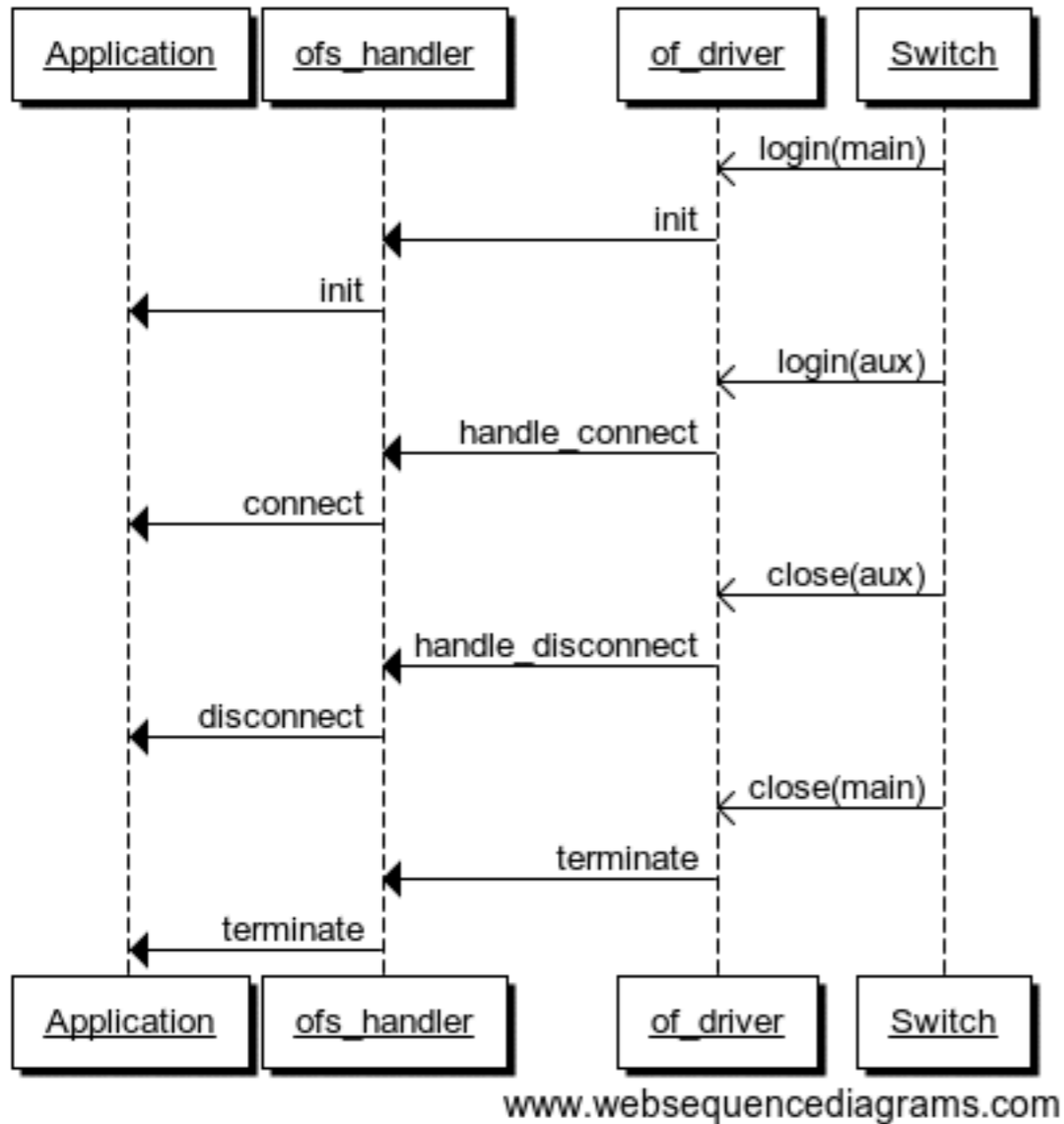


Asynchronous Messages

Protocol Flow



Callbacks



Sending Request

- Create message record with `of_msg_lib`
- `ofs_handler:sync_send`
- Process replies, if any

Subscriptions

- Handling async messages
- `ofs_handler:subscribe(DPID, Module, What)`
- `atom, {atom, fun/1}`

Example: adding flow

```
Matches = [{in_port, <<1:32>>}],
Instructions = [{apply_actions,
  [{output, 2, no_buffer}]}],
Opts = [{table_id, 0}, {priority, 100},
  {idle_timeout, 0},
  {cookie, <<0, 0, 0, 0, 0, 0, 0, 0, 10>>},
  {cookie_mask, <<0, 0, 0, 0, 0, 0, 0, 0>>}],
Request = of_msg_lib:flow_add(4,
  Matches, Instructions, Opts),
...
```


Example: adding flow

```
Reply = ofs_handler:sync_send(DPID,  
    Request).
```

Success:

```
{ok, noreply}
```

Example error:

```
{ok, {ofp_message, 4, error, 0,  
    ofp_error_msg, bad_action,  
    bad_out_port, <<>>}}}
```

packet_in

- `ofs_handler:subscribe(DPID, Module, packet_in)`
- `Module:handle_message({packet_in, Xid, Body}, State)`

Danger!

- Calls from callbacks can cause deadlocks

Current Status

- PlugFest Indianapolis May 2014
- Successfully tested with switches from five vendors, Ixia, and Spirent
- Managed 2000 simulated switches on Ixia - flows between 1000 switches
- Demonstrated stats_poller in large topology test

Future

- Fault tolerance
- Switch level data store
- Graph store
- Network Executive
- Flow compiler

Learn More

- <http://www.opennetworking.org>
- <http://www.flowforwarding.org>
- <http://github.com/FlowForwarding>

Contribute!

- Learning switch
- Topology discovery (LLDP)
- Expand icontrol, more stats poller integration
- OpenFlow Config
- Documentation

Questions?

- <http://github.com/FlowForwarding>
- Marc Sugiyama
marc.sugiyama@erlang-solutions.com