Network Virtualization in Multi-tenant Datacenters

Teemu Koponen, Keith Amidon, Peter Balland, Martín Casado, Anupam Chanda,
Bryan Fulton, Igor Ganichev, Jesse Gross, Natasha Gude, Paul Ingram, Ethan Jackson,
Andrew Lambeth, Romain Lenglet, Shih-Hao Li, Amar Padmanabhan, Justin Pettit,
Ben Pfaff, Rajiv Ramanathan, Scott Shenker*, Alan Shieh, Jeremy Stribling,
Pankaj Thakkar, Dan Wendlandt, Alexander Yip, Ronghua Zhang

February 27, 2014

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Background

- Two Technology Trends:
  - Datacenters
    - In-house services -> Computing resource provision
  - Virtualization
    - Physical infrastructure -> Logical computing unit

=> Multi-tenant datacenter (MTD)

Background

- Successful Story: Amazon
  - Infrastructure for selling and shipping books
  - Amazon Web Service (AWS)
    - Computing, Storage, Database, MessageQueue…

      – Steve Yegge's Google Platforms Rant

Background

- Compute and storage
  - have been successfully abstracted
- Network
  - Is VLAN suitable?
    - Fairly static manner
  - NVP solution provides full generality
    - Software-Defined Networking (SDN) based
    - Network hypervisor built on controller cluster
    - Datapath enhancement

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

System Design

- Overview
  - Abstractions
    - Control abstraction
    - Packet abstraction
  - Virtualization Architecture
    - Software switch
    - Service / Gateway node
    - Controller cluster

System Design

- Abstractions: provided by network hypervisor
  - Control abstraction:
    - Logical datapath: forwarding pipeline in the form of a sequence of lookup tables
    - Arbitrary matching
  - Packet abstraction:
    - Same switching, routing, and filtering services
    - MAC learning, unicast / multicast / broadcast flow

System Design

- Virtualization Architecture
  - Host hypervisor

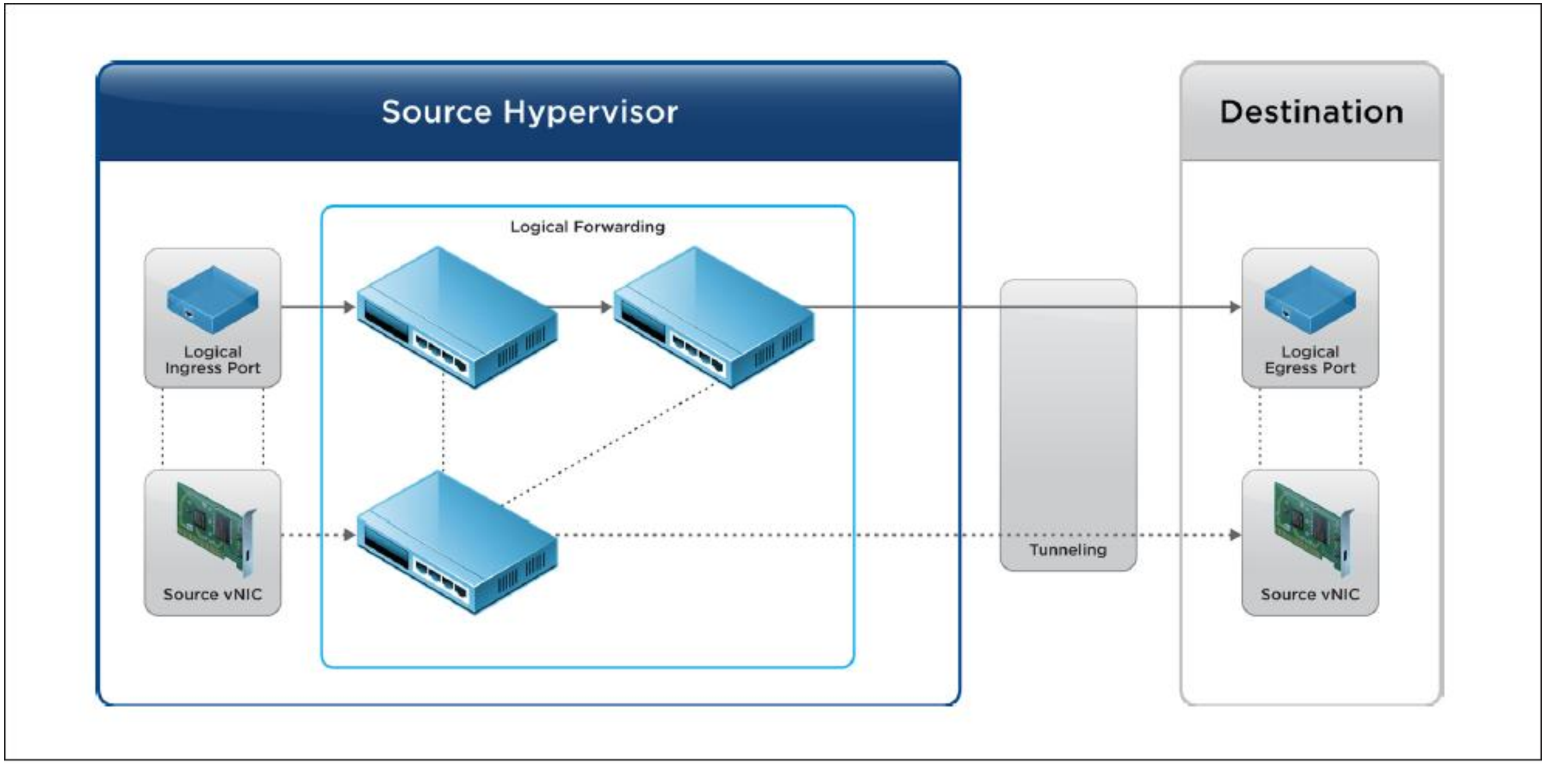


Figure 1. Logical forwarding is implemented by the virtual switch of the originating host hypervisor. After the logical forwarding decision is made, the packet is tunneled across the physical network to the receiving host hypervisor for delivery to the destination VM.

System Design

- Virtualization Architecture
  - Overall

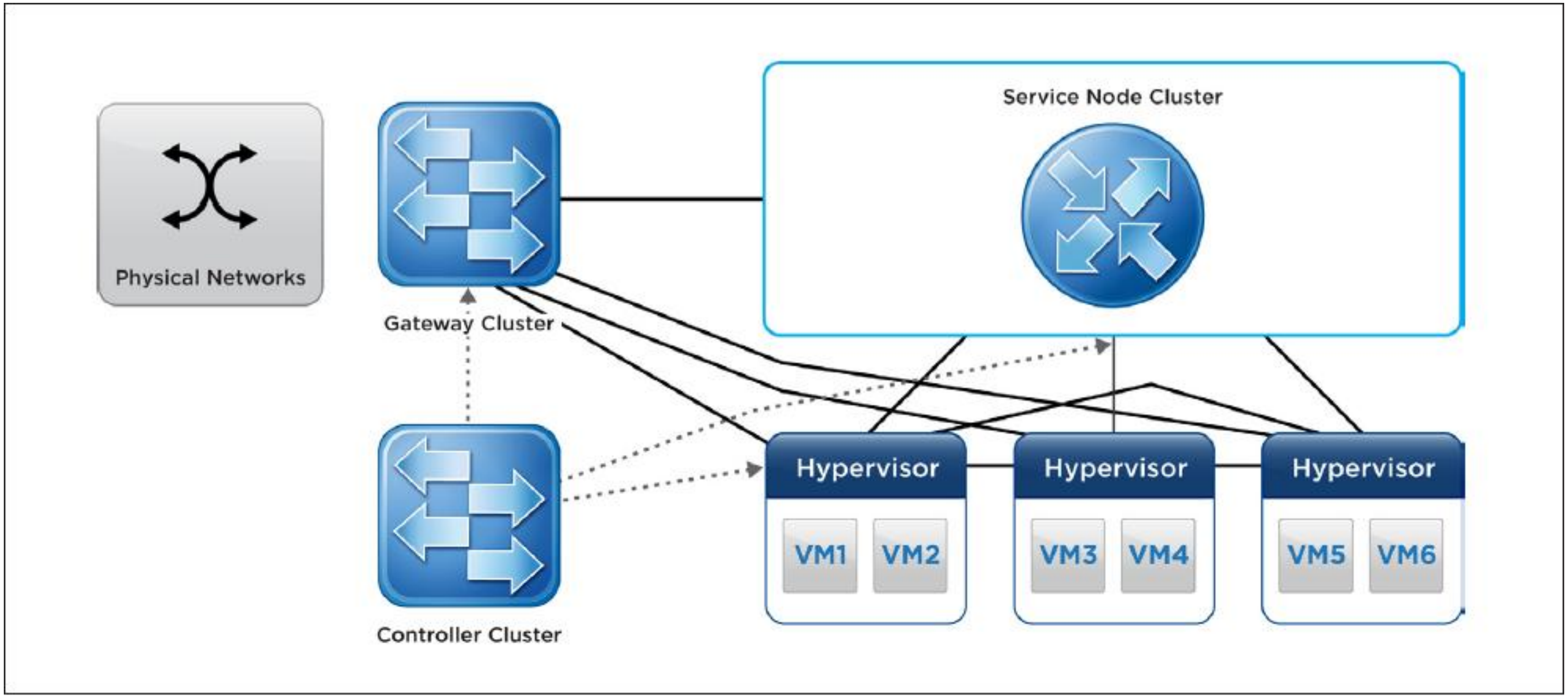


Figure 2. In NVP, a controller cluster manages the forwarding state at all transport nodes (hypervisors, gateways, service nodes), but is not involved in packet routing. Transport nodes are fully meshed over IP tunnels (solid lines). Gateways integrate the logical networks with physical networks, and service nodes provide replication for logical multicast/broadcast.

System Design

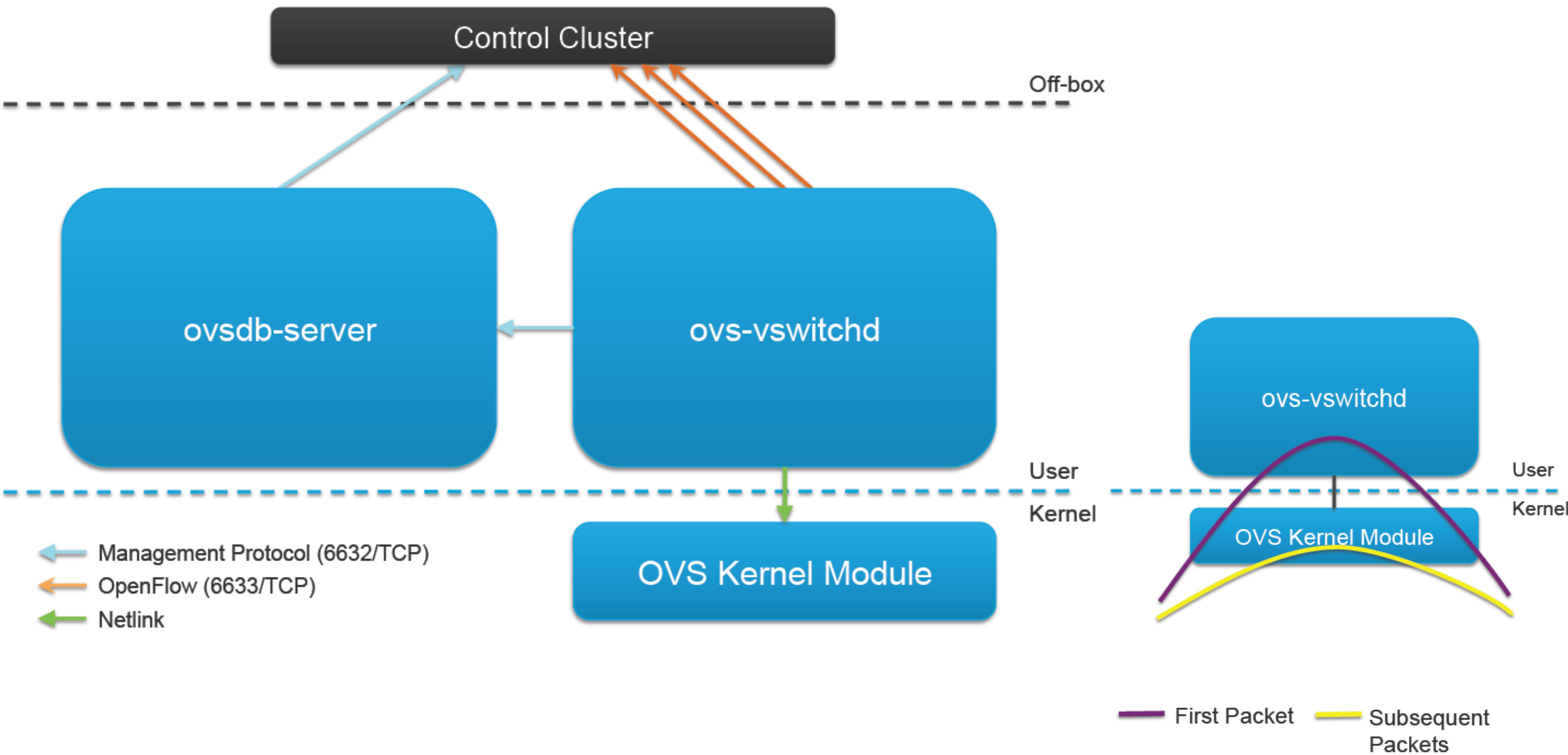
- Design Challenges
  - Datapath design and acceleration
    - OvS, STT, fast failover
  - Forwarding State Computation
    - Declarative, new domain-specific language
  - Scaling the cluster computation
    - Distribution, Availability……

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Network Edge Support

- Open vSwitch



Network Edge Support

- Logical datapath
  - A series of flow tables (uuid)
    - A set of flow entries
      - Match: packet header
      - Action: fwd, mod, drop
- Open vSwitch datapath
  - Single flow table
    - A set of flow entries
      - Match: packet header, packet metadata
      - Action: fwd, mod, drop, mod metadata, resubmit

Network Edge Support

- Forwarding performance
  - Matching mechanism
    - First packet: userspace, hash classifier
    - Subsequent packets: kernel, exact match
    - Similar to firewall, not a big problem
      - Miniflow in userspace, megaflow in kernel

Network Edge Support

- Forwarding performance
  - Encapsulation mechanism
    - Significant overhead
    - Two standard offload mechanisms:
      - TCP Segmentation Offload (TSO)
      - Large Receive Offload (LRO)
    - Problem: NICs do not support offloading in the presence of IP encapsulation

Network Edge Support

- Stateless Transport Tunneling (STT)
  - Fake TCP header after the physical IP header
  - STT header (only once)
    - Pros. No handshake, bandwidth saving
    - Cons. Difficult for HW parsing

Network Edge Support

- Fast Failovers
  - Hypervisor
  - Service and gateway nodes
  - Solutions:
    - Multi-path, ECMP
    - Multiple instance
    - Heartbeat

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Forwarding State Computation

- Computational Structure of Controller
  - Single controller

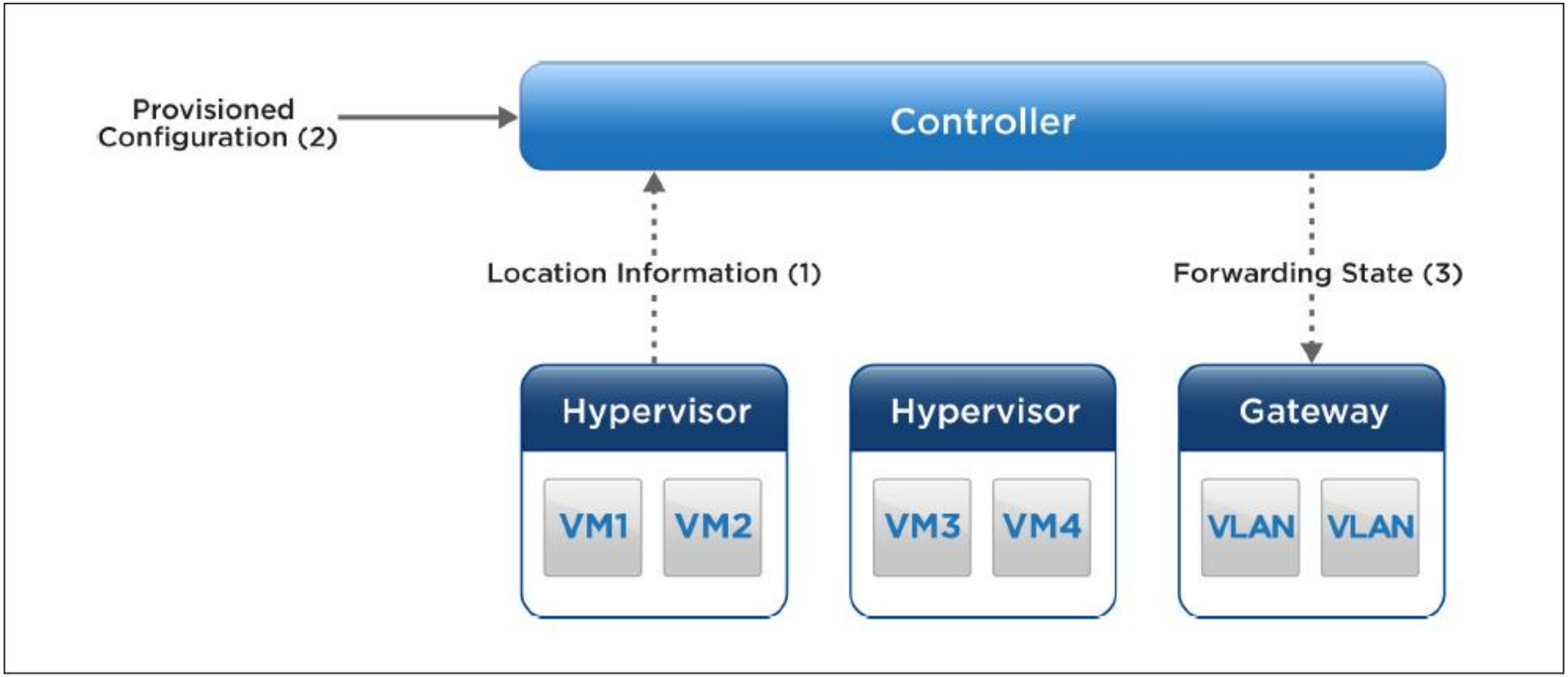


Figure 3. Inputs and outputs to the forwarding state computation process.

Forwarding State Computation

- Computational Challenge
  - Input: 123 types, device states & user config, proportional to the size of MTD
  - Output: 81 types
  - Solutions:
    - Full input-output translation is inefficient
    - Incremental update:
      - Hand-written state machine is impractical

Forwarding State Computation

- Automatically Generating an Incremental State Machine
  - nlog: domain-specific, declarative language
  - Separate logical spec from state transition
  - Logical spec: declarative, a function mapping the input to output
  - State transition: event processing code generated by compiler

Forwarding State Computation

System input types	123
System output types	81
All tables (internal, input, and output)	889
Declarative rules	1224
Lines of declarative code	27000

Table 1. nlog stats.

```
tunnel(phys_switch_id, encap_type, dst_ipaddr) :-
    log_port_present(log_port_id, phys_switch_id),
    log_port_present(log_port_id_2,
                     other_phys_switch_id),
    phys_switch_connector(other_phys_switch_id,
                     encap_type, dst_ipaddr),
    log_port(log_port_id, log_datapath_id),
    log_port(log_port_id_2, log_datapath_id),
    log_datapath_encap(log_datapath_id, encap_type);
```

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Controller Cluster

- Scaling
  - Two-layer hierarchy
  - Both are implemented in nlog

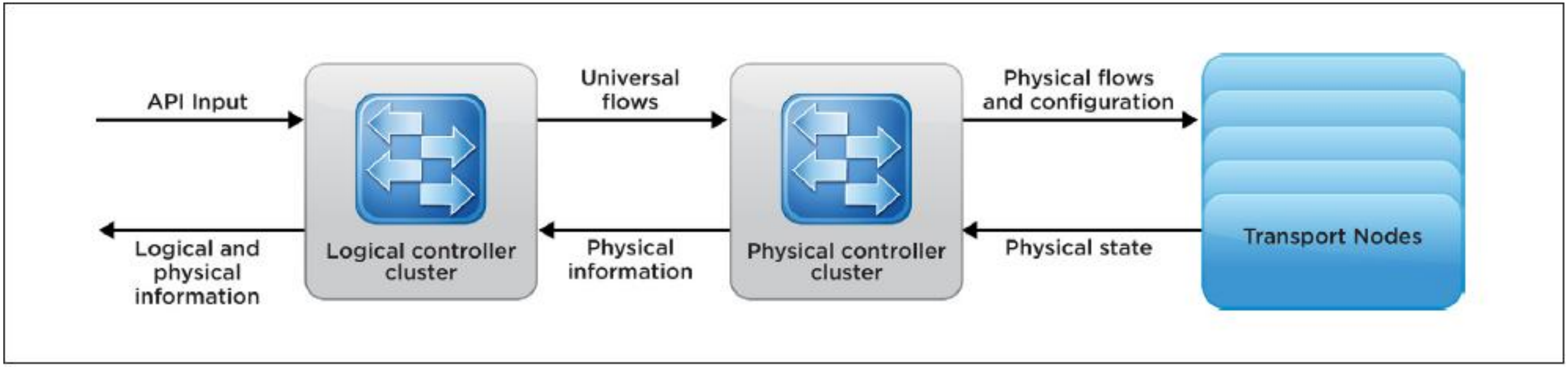


Figure 4. NVP controller cluster.

Controller Cluster

- Availability
  - Hot standbys for both logical and physical controller
  - For every shard, one controller is master, and the others are slave and candidates
  - OvS are configured to connect to both the master and slave

Controller Cluster

- Distributed Services
  - Use zookeeper for
    - leader election
    - label allocation
    - configuration storage
- API for Service Providers
  - Over HTTP
  - Blocking API

Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Evaluation

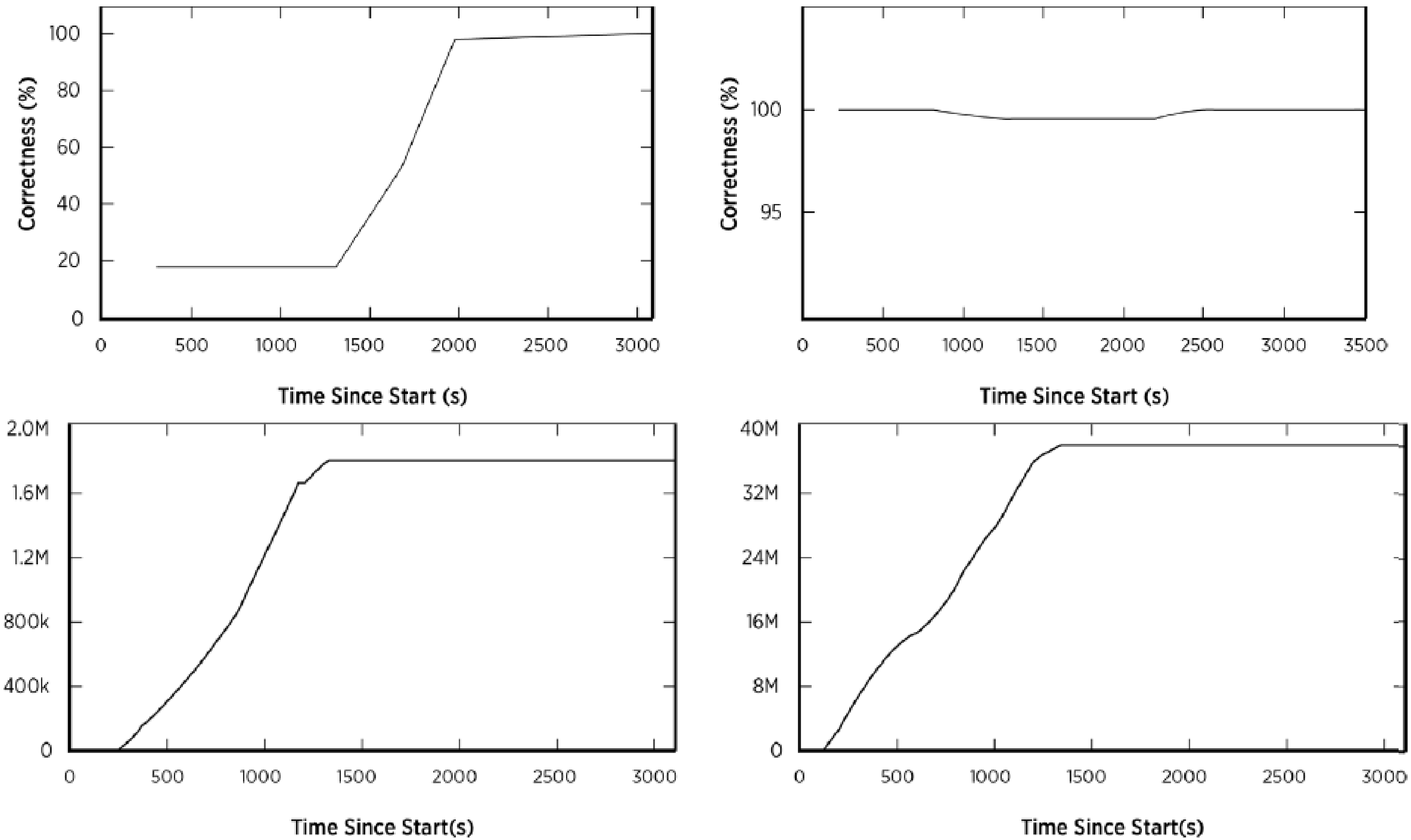
- Controller Cluster
  - Setup
    - 3000 simulated hypervisors, each with 21 vNICs
    - 7000 logical datapath, 9 ports / ldp on average
    - ~50000 ACLs
    - Controller: 12 core Xeon E5645 2.4GHz, 96GB memory, 400GB Hdisk
    - Simulated hypervisors on top of XenServer 5.6

Evaluation

- Controller Cluster
  - Test
    - a set of pings between logical endpoints
    - continue to send ping until all pings have the desired outcome

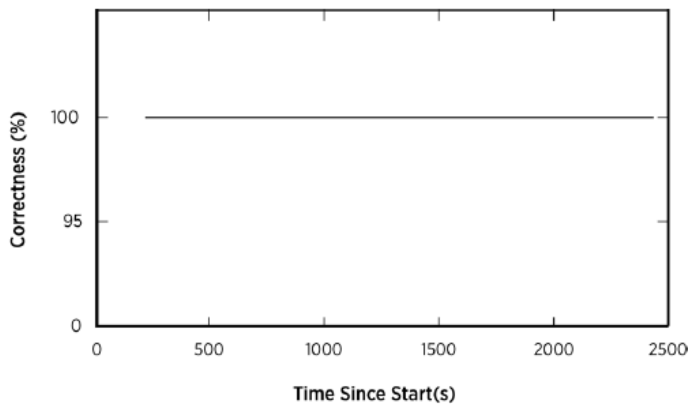
Evaluation

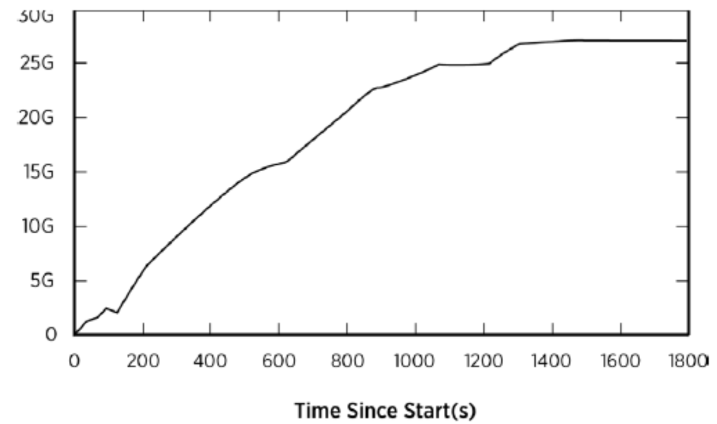
- Controller Cluster (Correctness & memory)



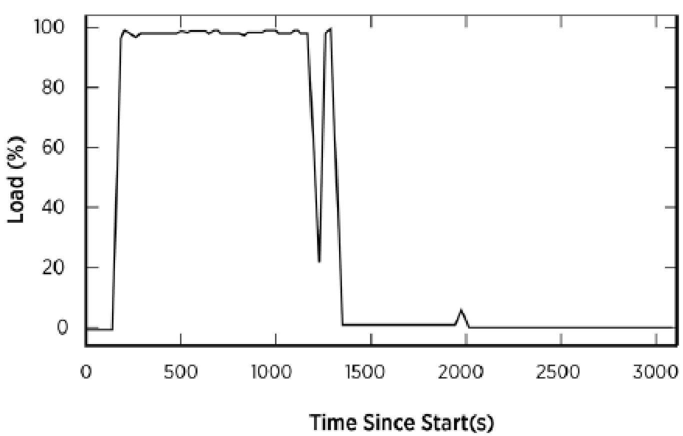
Evaluation

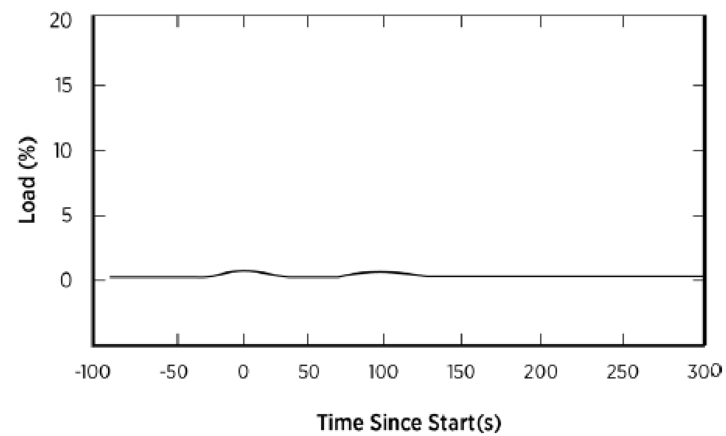
- Controller Cluster (Correctness & memory)





- Controller Cluster (Load)



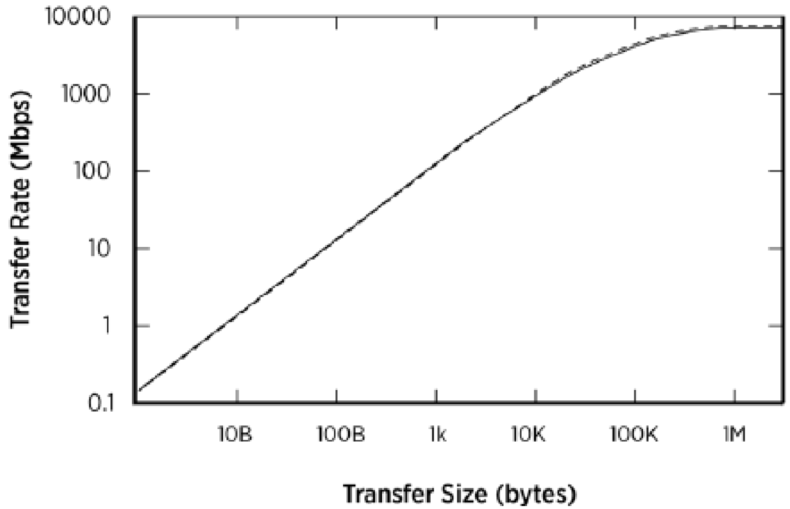


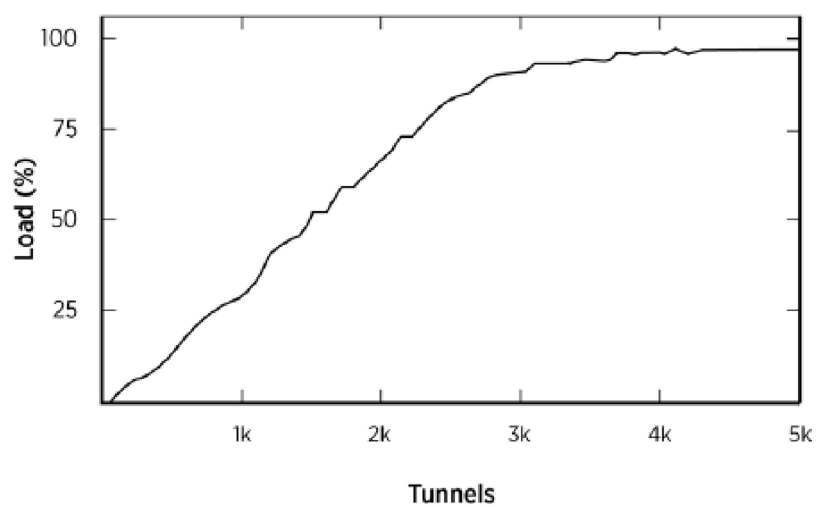
Evaluation

- Transport Nodes (Tunnel Performance)

	NO ENCAP	STT	GRE
TX CPU load	49%	49%	85%
RX CPU load	72%	119%	183%
Throughput	9.3Gbps	9.3Gbps	2.4Gbps

Table 2. Non-tunneled, GRE, and STT performance benchmark.





Outline

- Background
- System Design
  - Network Edge Support
  - Forwarding State Computation
  - Controller Cluster
- Evaluation
- Discussion

Discussion

- Seeds of NVP’s Success
  - Basing NVP on a familiar abstraction
    - Similar with legacy network, no modification
  - Declarative state computation
  - Leveraging the flexibility of OSS
    - No haggling over standards
  - Others
    - NetMap, Intel DPDK, NG NIC…

Discussion

- Implications for SDN
  - Canonical SDN: OpenFlow, Controller, APP
    - Simplify the implementation of network management
  - Network Virtualization
    - Conceal the complexity of coping with underlying physical infrastructure

Thanks

February 27, 2014