lecture 18: network virtualization platform (NVP)
5590: software defined networking

anduo wang, Temple University
TTLMAN 401B, R 17:30-20:00

Network Virtualization in multi-tenant Datacenters

Teemu Koponen., et al. "Network Virtualization in Multi-tenant Datacenters"

(partial) server virtualization

server virtualization

- managing computational resources by exposing the software abstraction of a server to users

partially realized

- new application / environment requires an associated change in the network

(partial) server virtualization

partially realized

- new application / environment requires an associated change in the network and services
- different workloads demand different *topology*
  - flat L2, L3, L4-L7
- virtualized workloads operate in the physical *address space*
  - arbitrary location, address type,…

server virtualization requires network virtualization

- no single unifying abstraction, invoked in a global manner

(partial) server virtualization

computation is virtualized, the network is not

network virtualization primitives

- VLAN — virtualize L2 domain
- VRFs — virtualize L3 FIB
- NAT — virtualize IP space
- MPLS — virtualize path

(partial) server virtualization

computation is virtualized, the network is not

network virtualization primitives

- VLAN — virtualize L2 domain
- VRFs — virtualize L3 FIB
- NAT — virtualize IP space
- MPLS — virtualize path

but

- traditional configured box-by-box
- no single unifying abstraction, invoked in a global manner

solution — network virtualization

virtual networks over the same physical network

- each with independent service models
- topologies
- addressing architectures

the creation and management

- done through global abstractions, rather than pieced together through box-by-box configuration

solution — network virtualization

virtual networks over the same physical network

- each with independent service models
- topologies
- addressing architectures

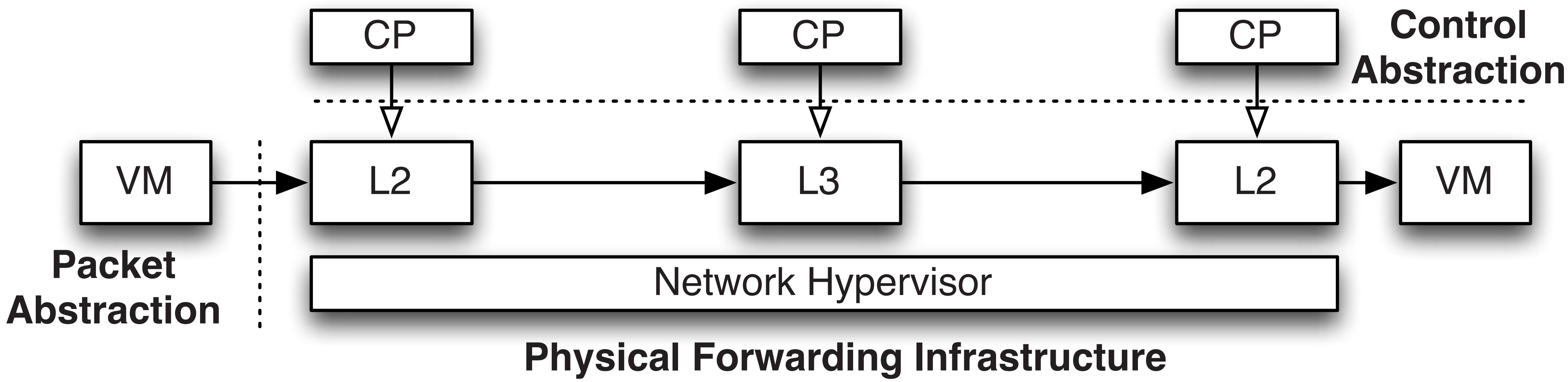
the creation and management

- done through global abstractions, rather than pieced together through box-by-box configuration

NVP

- deployed in dozens of production environments
- last few years
- tens of thousands of virtual networks and virtual machines
- enterprise network

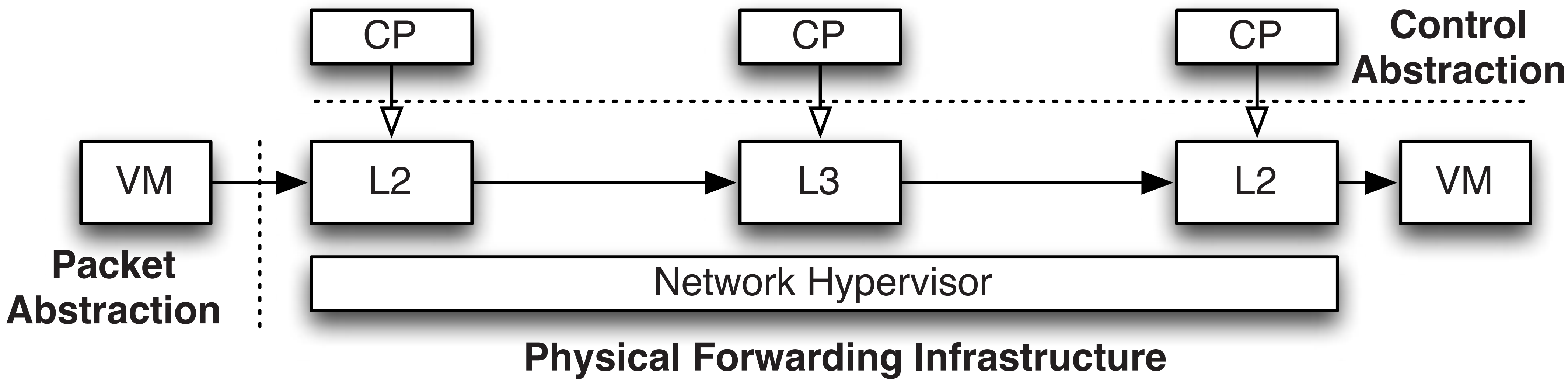
multi-tenant datacenter (MTD)



hosts connected by physical network

- each host hosts many VMs, connected by a virtual switch

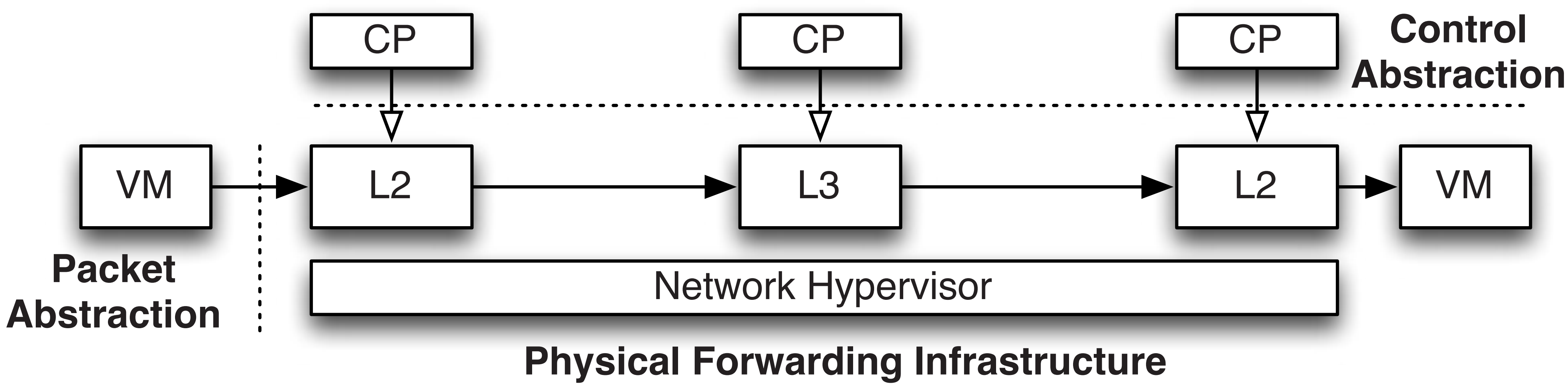
MTD — control abstraction



logical datapath

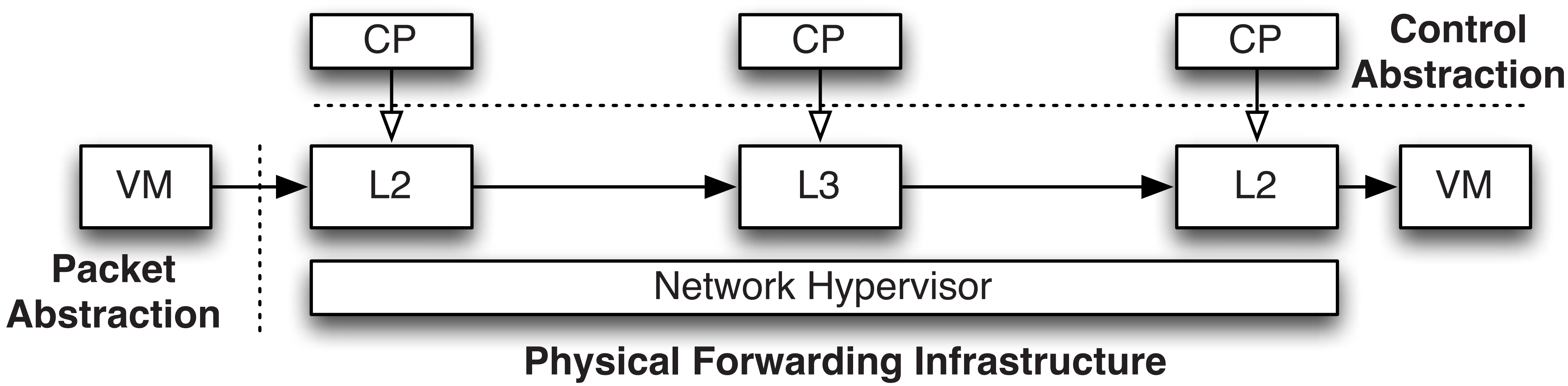
- set of logical network elements
- = a packet forwarding pipeline interface,
- = a sequence of lookup tables
- = resulting in a forwarding decision

MTD — packet abstraction



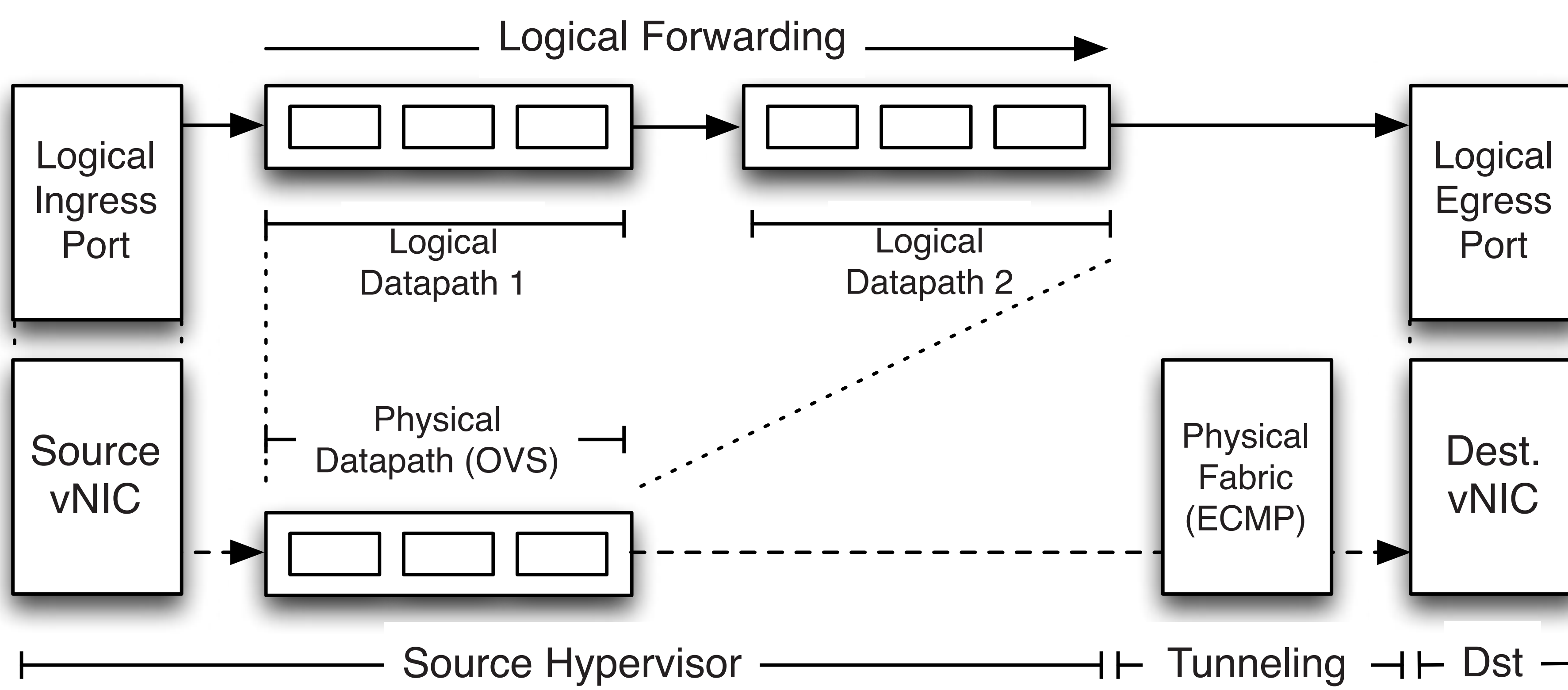
packets sent by endpoints in the MTB have the same switching, routing, and filtering services as in a physical network

control- and packet- abstractions



hypervisor implements the abstractions by implementing tenant-specific logical data paths over the provider's physical network

implementing control- and packet- abstractions



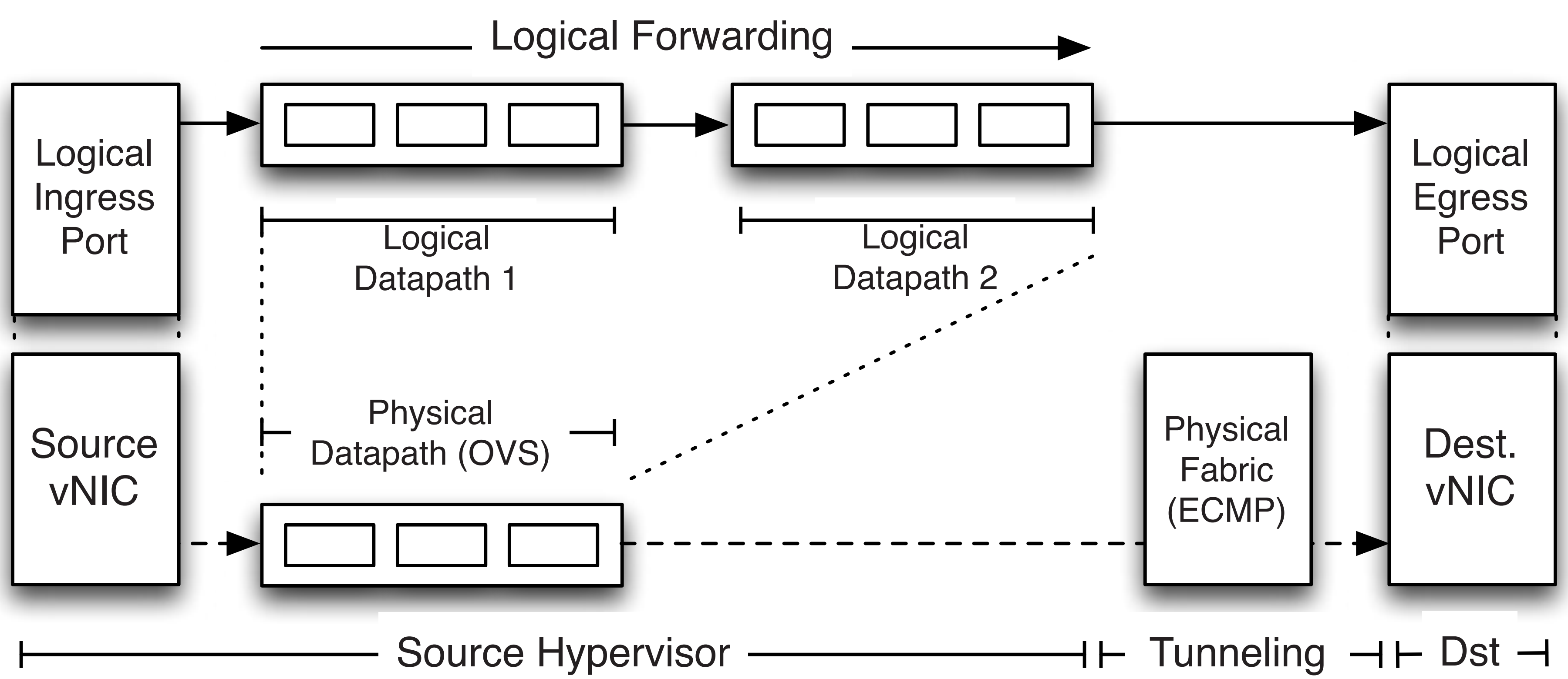
OVS on the sender VM

- implements the logical data path
- (after forwarding decision) tunnels to the receiving host hypervisor

the receiving hypervisor

- decapsulates the packet and sends it the destination VM

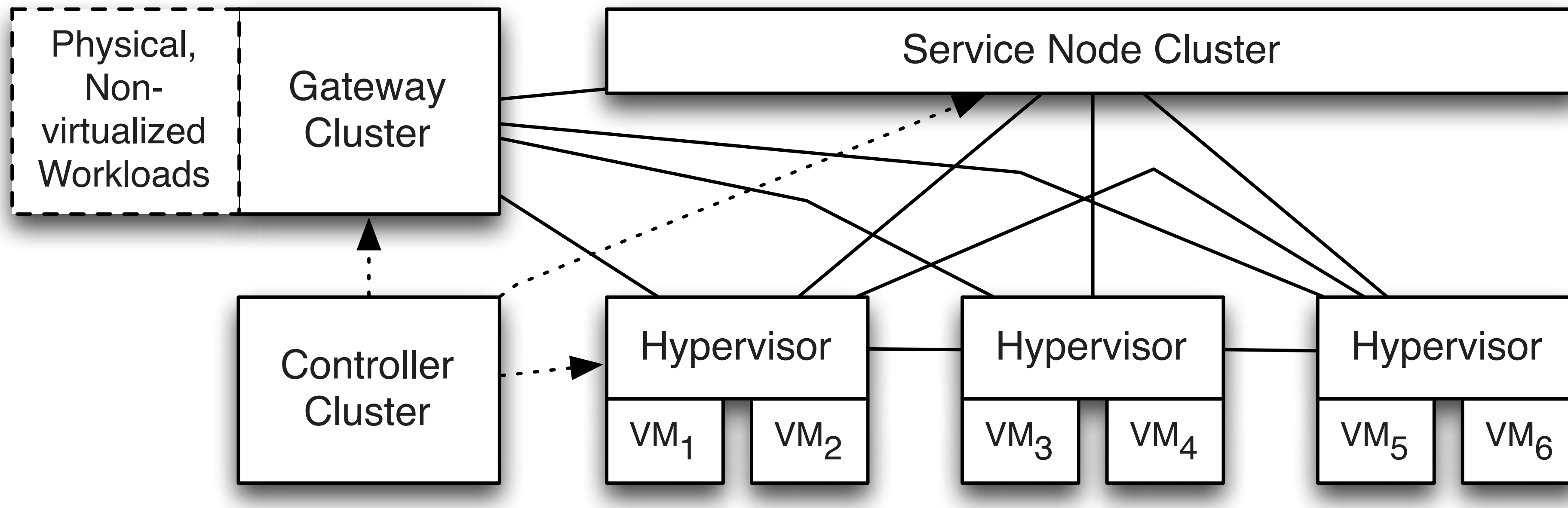
implementing control- and packet- abstractions



OVS on the sender VM

- configured by a centralized SDN controller

implementing control- and packet- abstractions

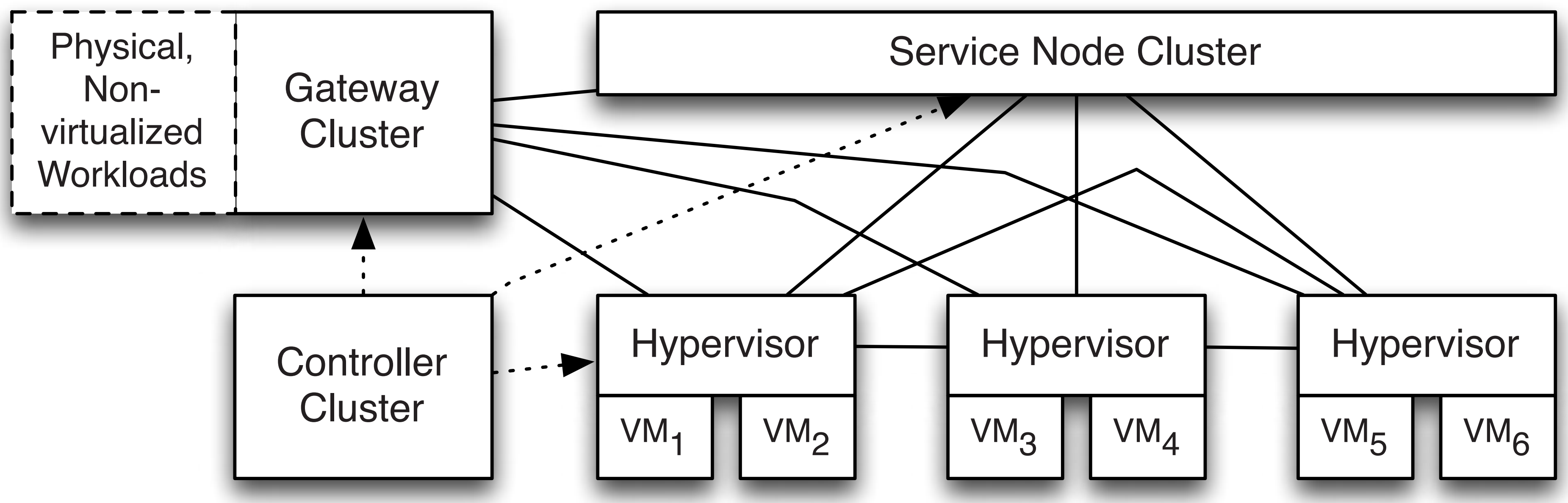


tunnels between every pair of host-hypervisors

- logical point-to-point
- logical broadcast, multicast

implemented by service nodes

virtualization architecture

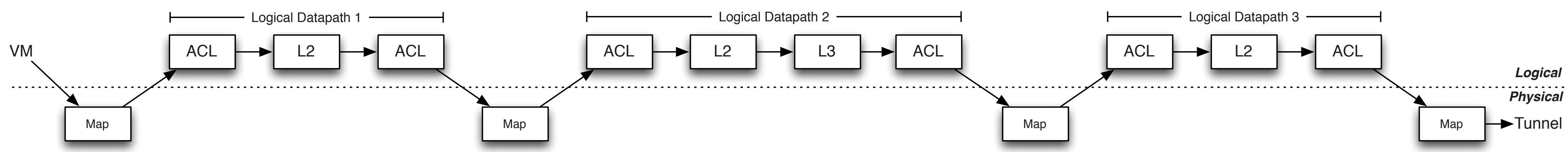


transport nodes

- service nodes, hypervisors, gateways

virtualization at the edge

logical datapath



flow tables

- similar to OVS flow-tables
- metadata registers (identifier of a logical path)

forwarding performance

problem: fast packet classification with wildcards

- TCAM not available on OVS

OVS solution: traffic locality

- kernel module: sends first packet of a new flow to userspace
  - follow-up packets quickly matched by kernel
- user module: matched against the full flow table
  - install on kernel exact match

fast failovers

hypervisor failures

- hypervisor-to-hypervisor tunnel cannot survive

service node failure

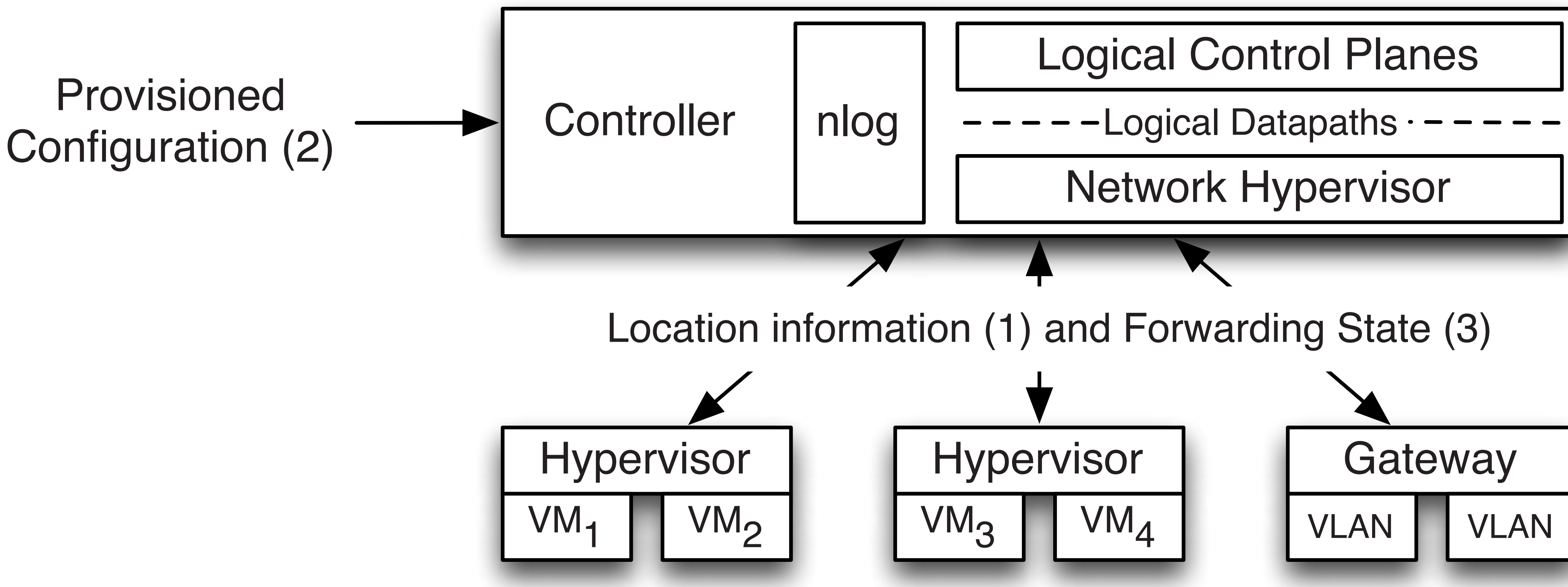
- controller load-balance traffic across many service nodes

gateway nodes

- many gateway nodes for each physical network
- failover to backup (leader/backup to prevent loop)

forwarding state computation on a single controller

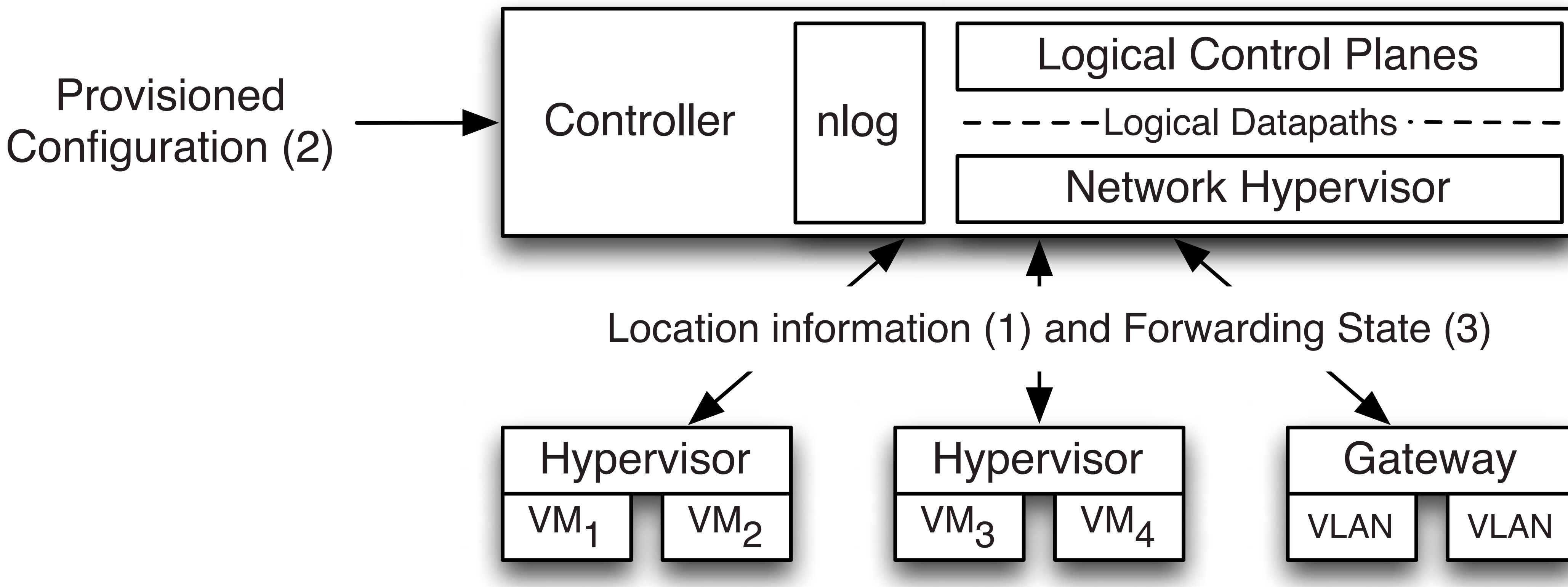
forwarding state computation



inputs

- (1) location of vNICs
  - through OVS, update as VM migrates

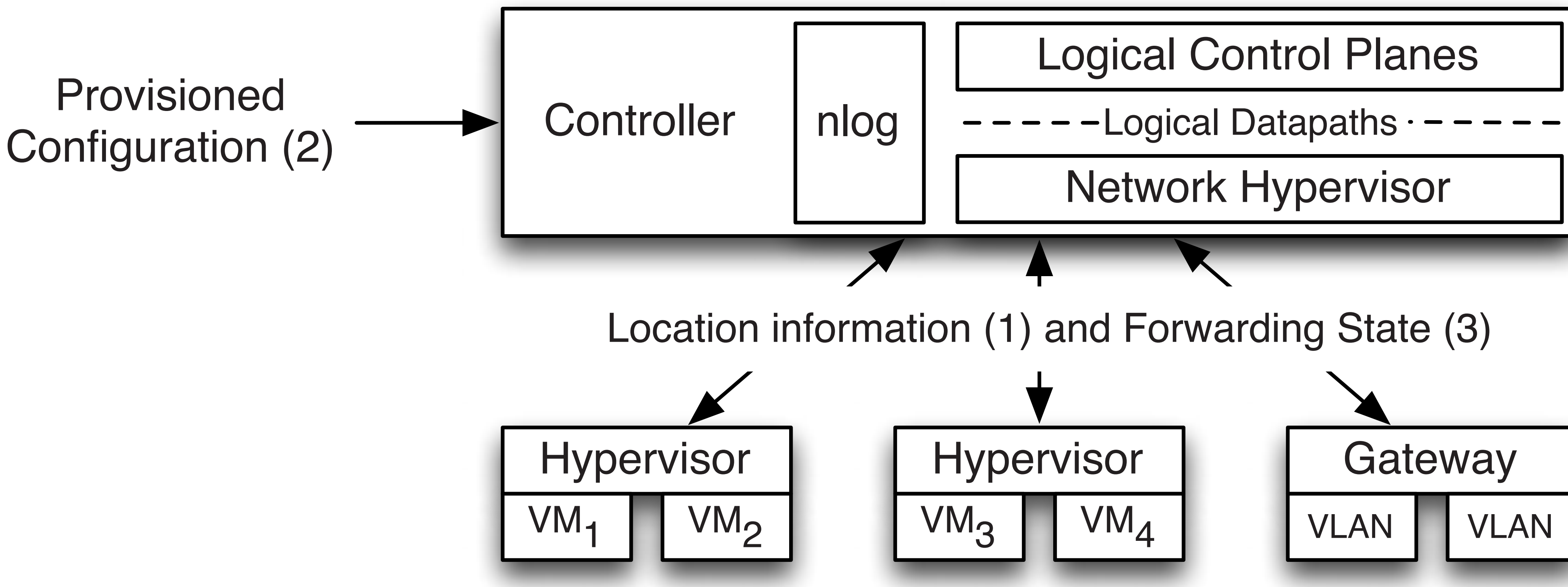
forwarding state computation



inputs

- (1) location of vNICs
  - through OVS, update as VM migrates
- (2) service provider configuration
  - through NVP API, update as tenant's (virtual) network and/or physical network change

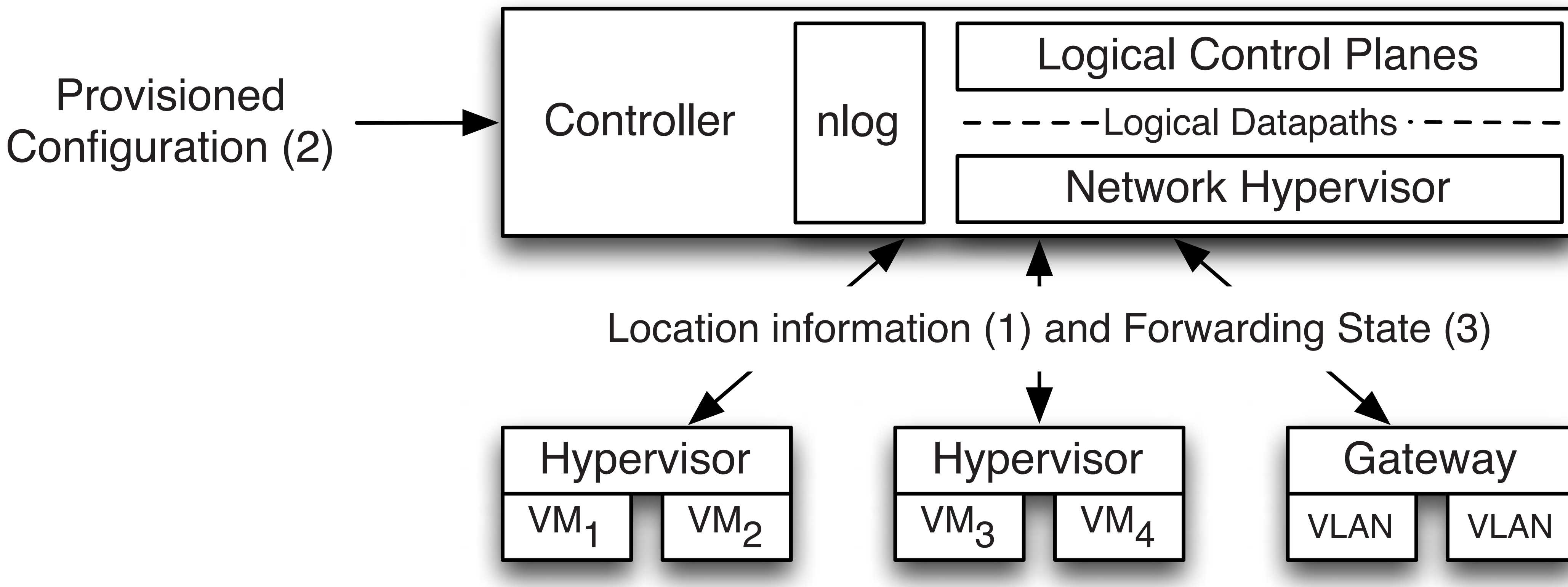
forwarding state computation



output

- logical lookup tables
- (3) (transformed by the hypervisor into the physical) forwarding states, pushed to transport nodes through OpenFlow and OVS

forwarding state computation



(proactive) output

- (3) proactively compute forwarding states, and push to transport nodes
- do not process any packets
- benefits: scaling, failure isolation

computation challenge

large total input size

- 123 types of input
  - eg., a particular type of logical ACL, location of a vNIC
- 81 types of output
  - eg., a single type of attribute being configured by OVS

frequent, localized changes

computation challenge

incremental computation with hand-written state machine infeasible

- number of event types
- arbitrary interleaving

incremental computation with nlog

head_table :- joined_table, joined_table, …

- maps controller input to output

types	tables
output types (or immediate results)	head table
input types (or immediate results)	joined table

incremental computation with nlog

head_table :- joined_table, joined_table, …

- maps controller input to output

types	tables
output types (or immediate results)	head table
input types (or immediate results)	joined table

- change in the joined tables results in (incremental) re-evaluation
  - inserting into or removing from head table

incremental computation with nlog

- non-recursive
- 1200 declarations and 900 tables (all three types)

benefits

- separate (logic) spec from the (state machine) implementation
- incremental evaluation without worrying about state transition, input event ordering

extending nlog with functions

extending nlog with functions

datalog re-structure column data

extending nlog with functions

datalog re-structure column data

nlog adds *function* table

- certain columns of a row is a stateless function of others
- NVP primitive function tables
  - match over flow, sequence of actions

extending nlog with functions

datalog re-structure column data

nlog adds *function* table

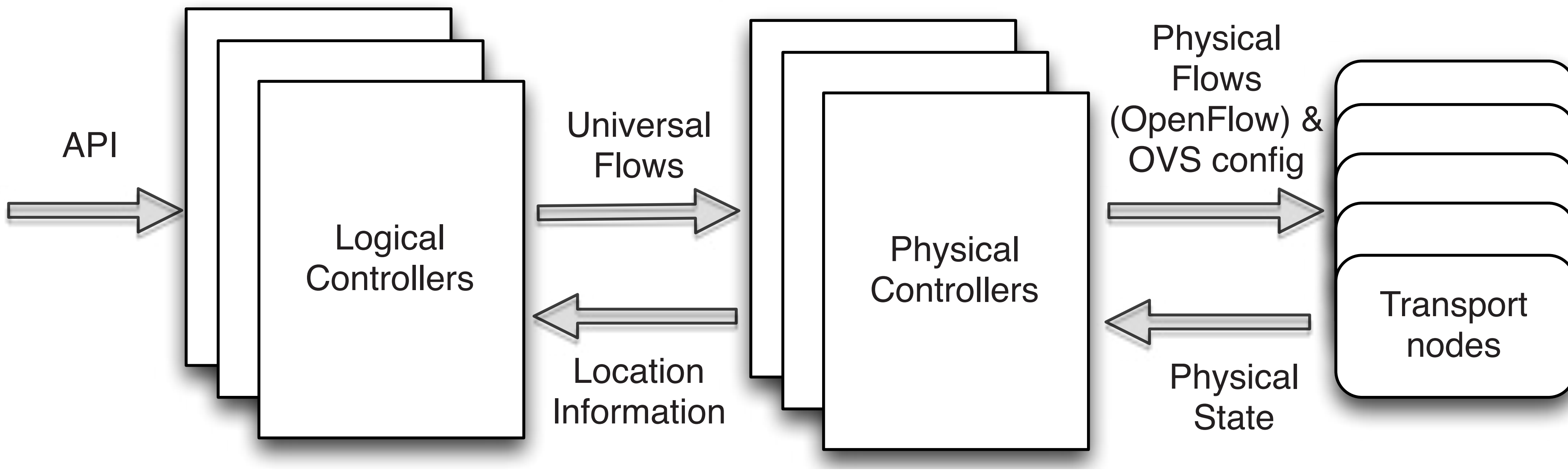
- certain columns of a row is a stateless function of others
- NVP primitive function tables
  - match over flow, sequence of actions

hook up output and input table by arbitrary C++

- output table tuples → C++ implementation
- C++ implementation processing
- C++ implementation → tuples
- → input table

distribution — controller cluster

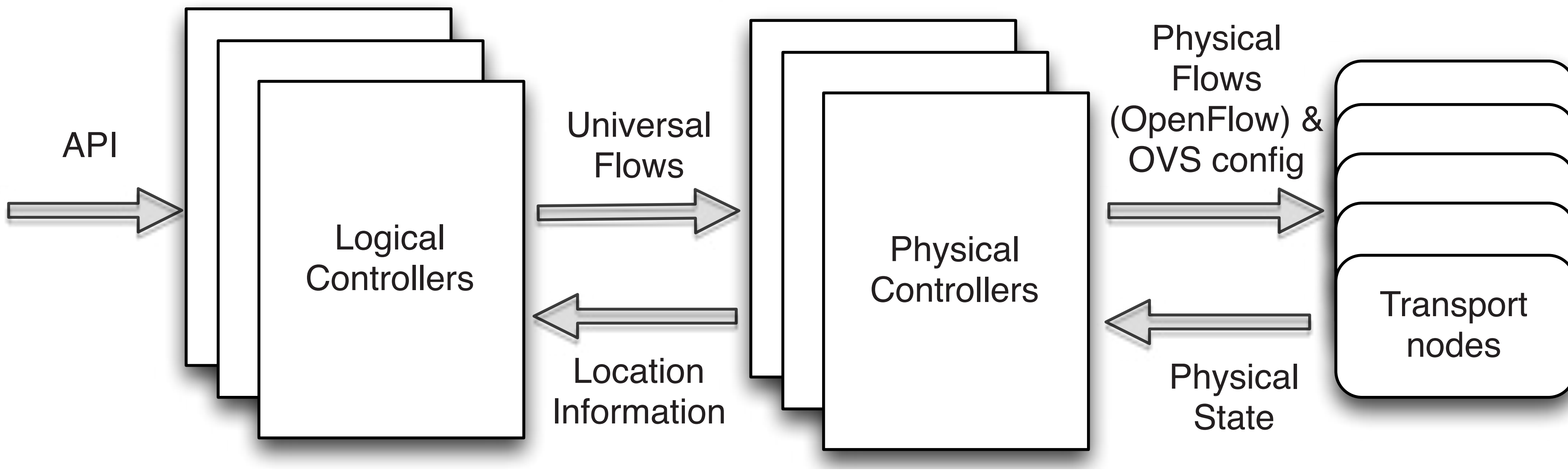
distribution of computation



distribute computation of logical data path among controllers

- sharding logical datapath using its identifier
- each controller computes
  - lookup tables + tunnels (universal flows)
- universal flows published over RPC to physical controllers

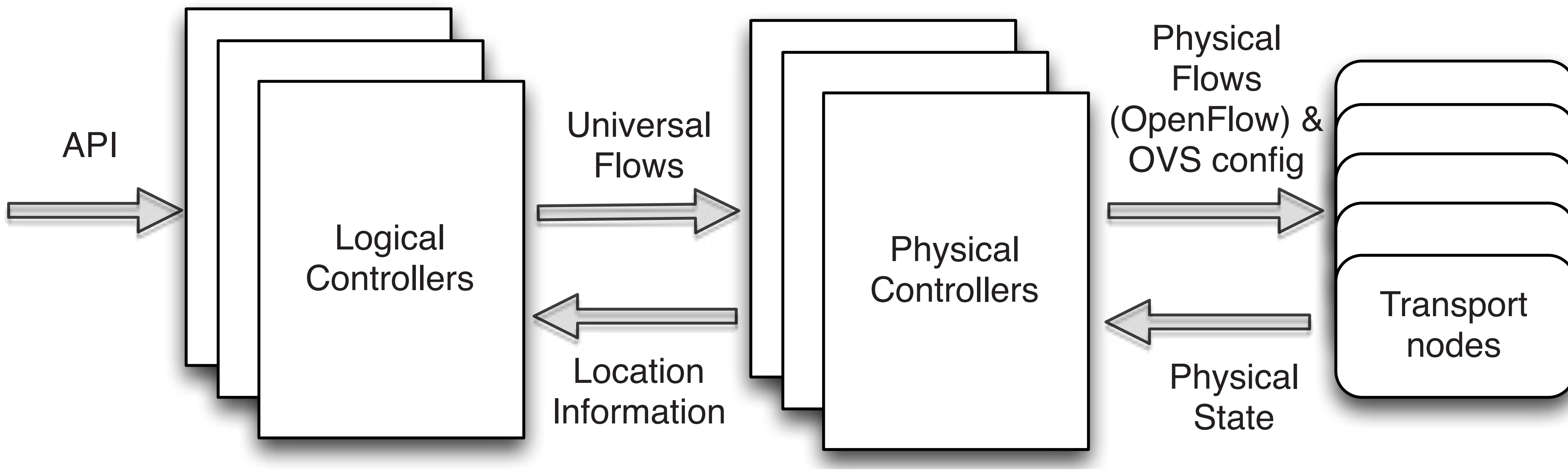
distribution of computation



distribute universal-to-physical translation among physical controllers

- sharding the translation for transport nodes among controllers
  - translation independent for each transport node

distribution of computation



logical-/physical- controller failover by *hot standbys*

- one highly-available controller acts as sharding coordinator
  - elected using Zookeeper
- maintain a (master, standby) pair
  - if master fails, promotes standby to master, find new standby
  - if standby fails, coordinator assigns a new standby

extended onix-distributed services

NVP uses Onix

- replicated transactional database to persist configuration state

(extend Onix by) Zookeeper

- elects sharding coordinator
- assigns globally unique label (for logical egress port) among the many controllers