

PaFloMon

A Slice Aware Passive Flow Monitoring Framework for
OpenFlow Enabled Experimental Facilities

Christos Argyropoulos
Network Management & Optimal Design Laboratory (NETMODE)
National Technical University of Athens

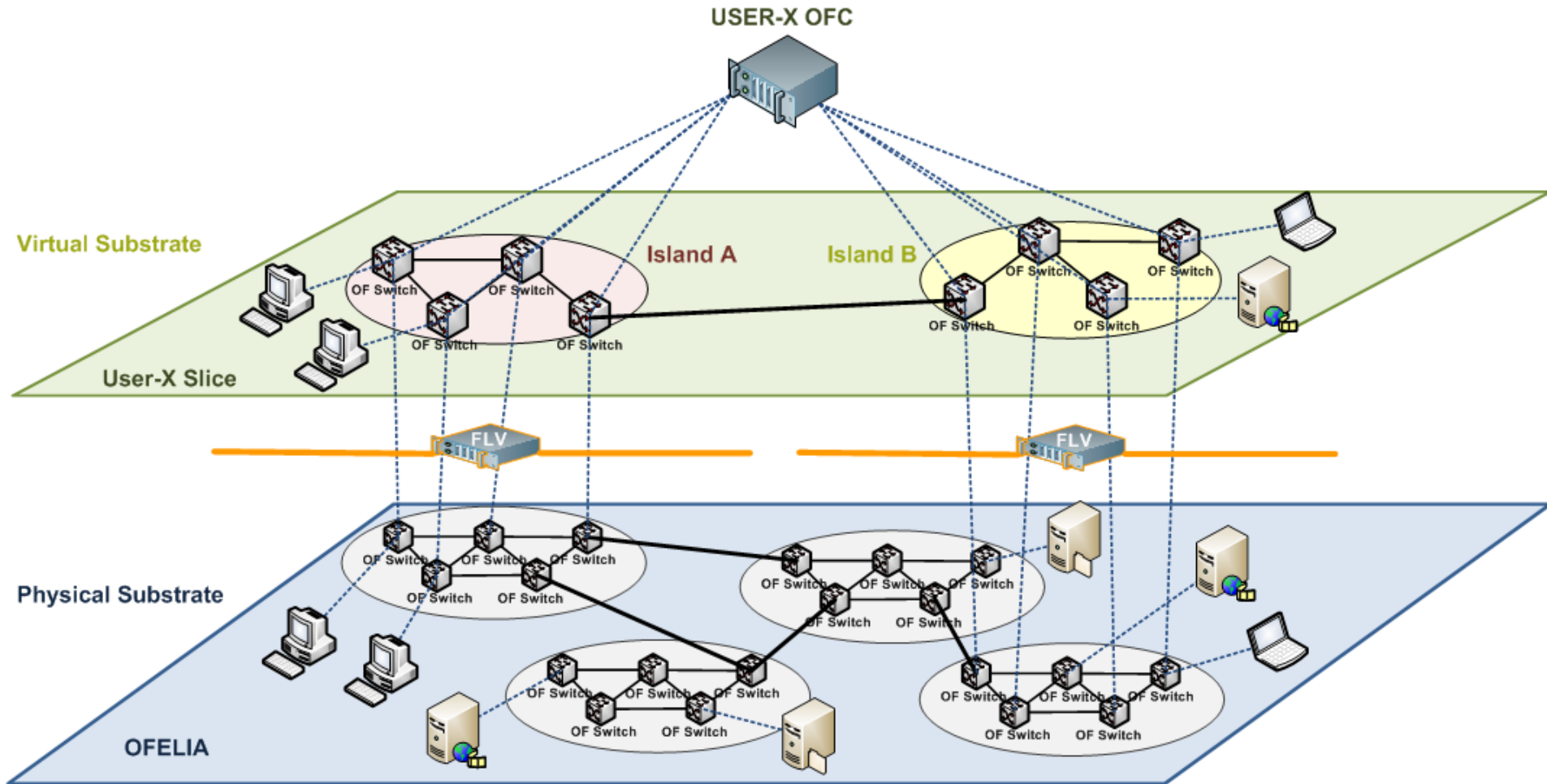
European Workshop on Software Defined Networks
October 26th , 2012
Darmstadt, Germany

Baseline

openflow (OF) enabled experimental facilities

- Management Plane
 - Compute resources provisioning
 - Flow-space provisioning upon network elements
 - Access control (authentication & authorization)
 - Topology creation
- Control Plane
 - Multi-tenant environment with isolation (slicing)
 - Compute resources isolation
 - Data plane isolation
 - Data plane exposure to user's control logic (via proxy OF controller – e.g. FlowVisor)

OFELIA case



Is anything missing?

Monitoring

- Infrastructure Monitoring
 - Management & control plane services availability
 - Compute resources state
 - OpenFlow switches state
 - FlowVisor state
 - PHY substrate link state
- Virtual Substrate/Slice Monitoring



Slice Monitoring

taking into consideration..

- Slice-centric monitoring on heterogeneous network infrastructures
- Role-based monitoring data view and easy access for the experimenters
- Existence of a monitoring specification that will promote homogeneity and compatibility
- Usability (web access & remote calls)

OpenFlow for Slice-based Monitoring?

OpenFlow data gathering support

- OpenFlow counters
 - Counters per-table, per-flow, per-port, per queue
- Easy fit with the slice-based concept
 - Slice-based data gathering based on flow-space definition
 - Proxy Controller permits per-slice counter view to the users' controllers

It seems like a dream!
OpenFlow fits everywhere

Does it ?

OpenFlow constraints

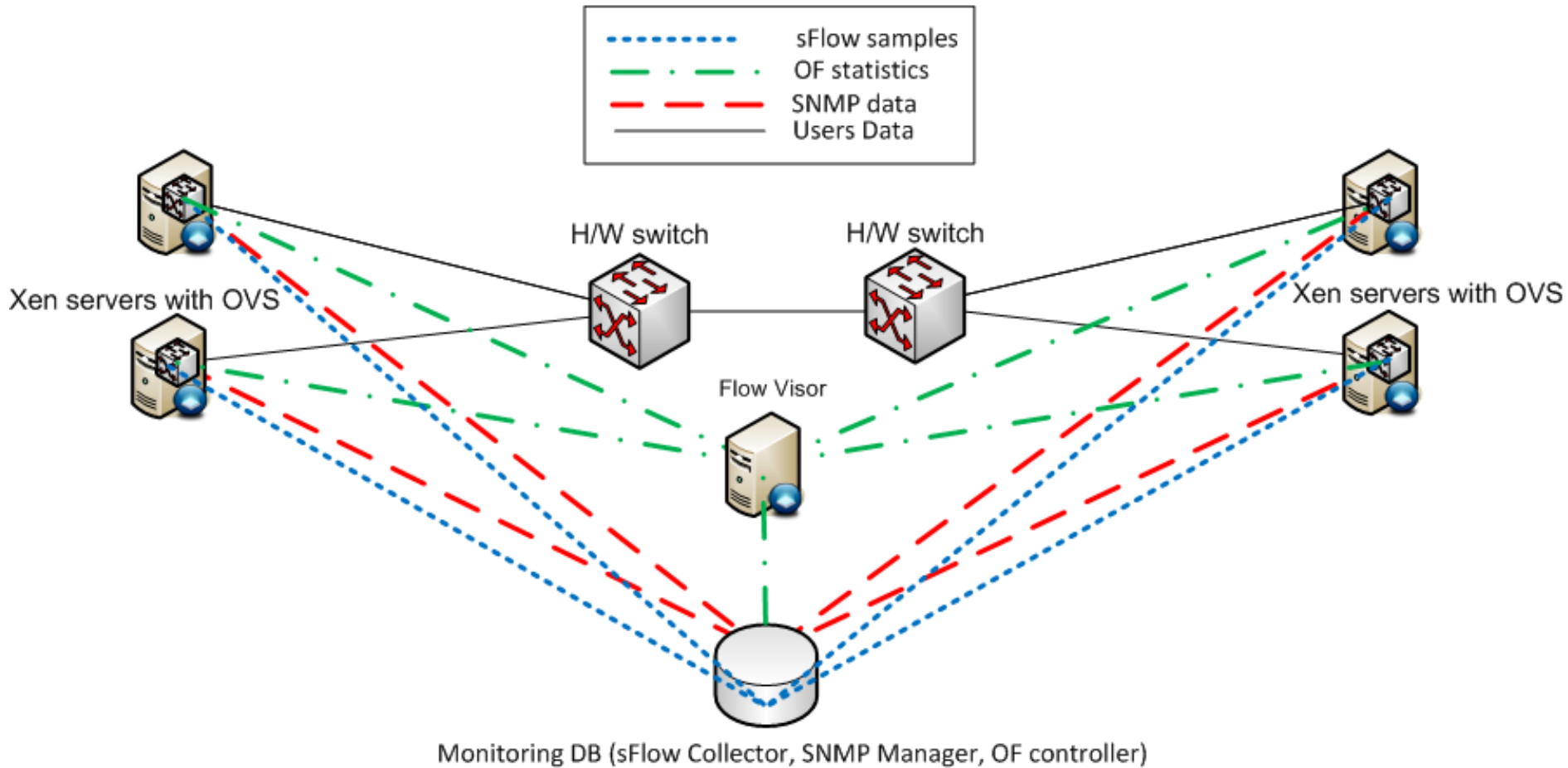
- OpenFlow counters
 - Flow forwarding table polling for statistics is not a good idea (control plane overloading)
 - You cannot gather flow statistics for a flow you do not have to the forwarding table
 - Aggregated flow matching during the forwarding process prohibits later on drilling down to micro-flows statistics
- Micro-flows use for data gathering (a very bad idea)
 - OpenFlow forwarding table overflow
 - High rate of *packet-in* messages from the OF switches to the OF Controllers (proxy controller bombing)

PaFloMon enablers

- Taking advantage of existing passive monitoring capabilities
 - sFlow
 - Purpose: packet sampling, flow-based exposure to the user
 - SNMP
 - Purpose: management info
 - OpenFlow stats
 - Purpose: flow-based stats for the applied forwarding rules

PaFloMon

data gathering on ofelia

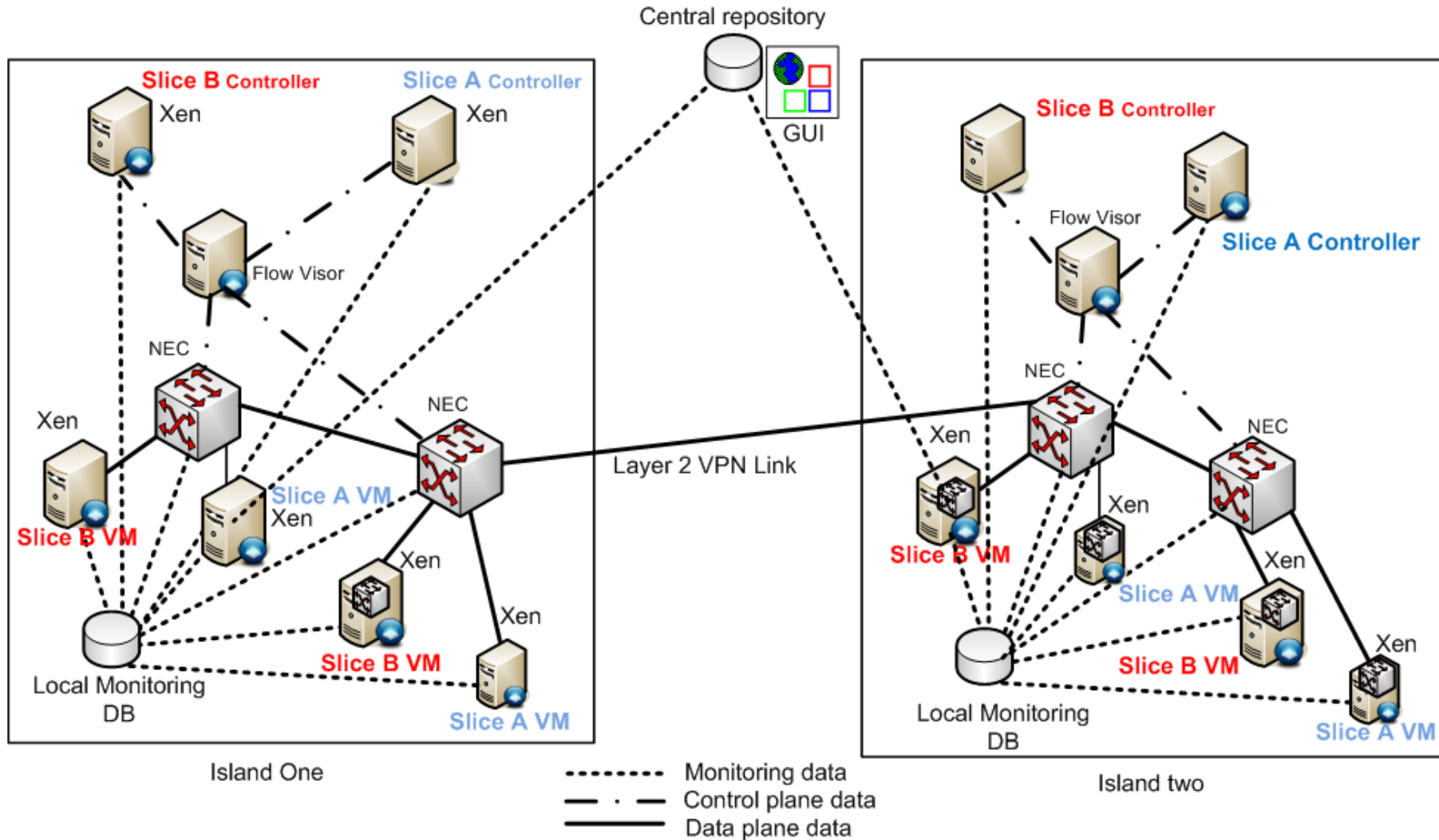


PaFloMon for Slices taking into consideration..

- *Slice-centric monitoring on heterogeneous networks*
 - ➔ Compound wired/wireless (e.g. sFlow on NEC/OpenWRT)
- *Role-based monitoring data view and easy access for the experimenters*
 - ➔ Roles correspond to slice user-view and admin-view
- *Monitoring spec for homogeneity and compatibility*
 - ➔ Monitoring RSpec for integration of monitoring requests
- *Usability (web access & remote calls)*
 - ➔ Web-based and XML-RPC access of the stored data

PaFloMon

high level overview



PaFloMon

feasibility tests on ofelia gear

- Micro-flows to the NEC switch
 - Rates that lead to tables' overflow affect both packet forwarding & sFlow sampling
- “*flow Number of flow entries per slice per dpid (switch)*” parameter
 - Must be carefully configured to avoid performance degradation of the entire infrastructure
 - The sum of the maximum numbers of flows per slice per switch, permitted by the FlowVisor, should not exceed the switch flow-table size limit

$$\left(\sum_{\kappa} \text{flows}_{\text{max, slice } \kappa, \text{ switch } \lambda} \leq \text{table_size}_{\text{switch } \lambda} \right)$$

Thank You!