Programmable Edge Node with Hybrid Multi-core Processors

PI: Yan Luo

University of Massachusetts Lowell
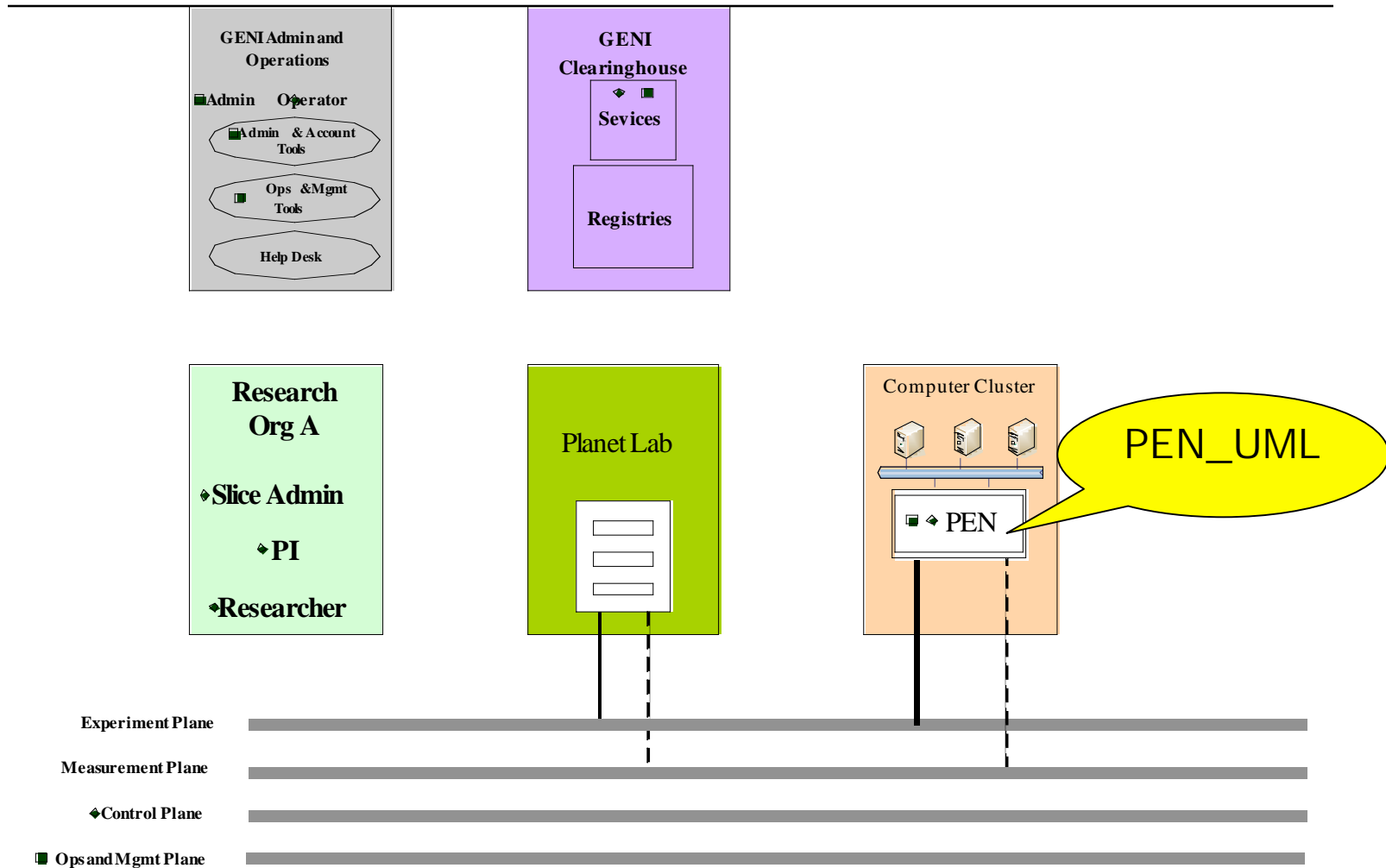
University of Massachusetts Lowell

Outline

- Project Goals of PEN_UML
- How PEN_UML fits in GENI
- Architecture and Benefits of PEN_UML
- Preliminary Data
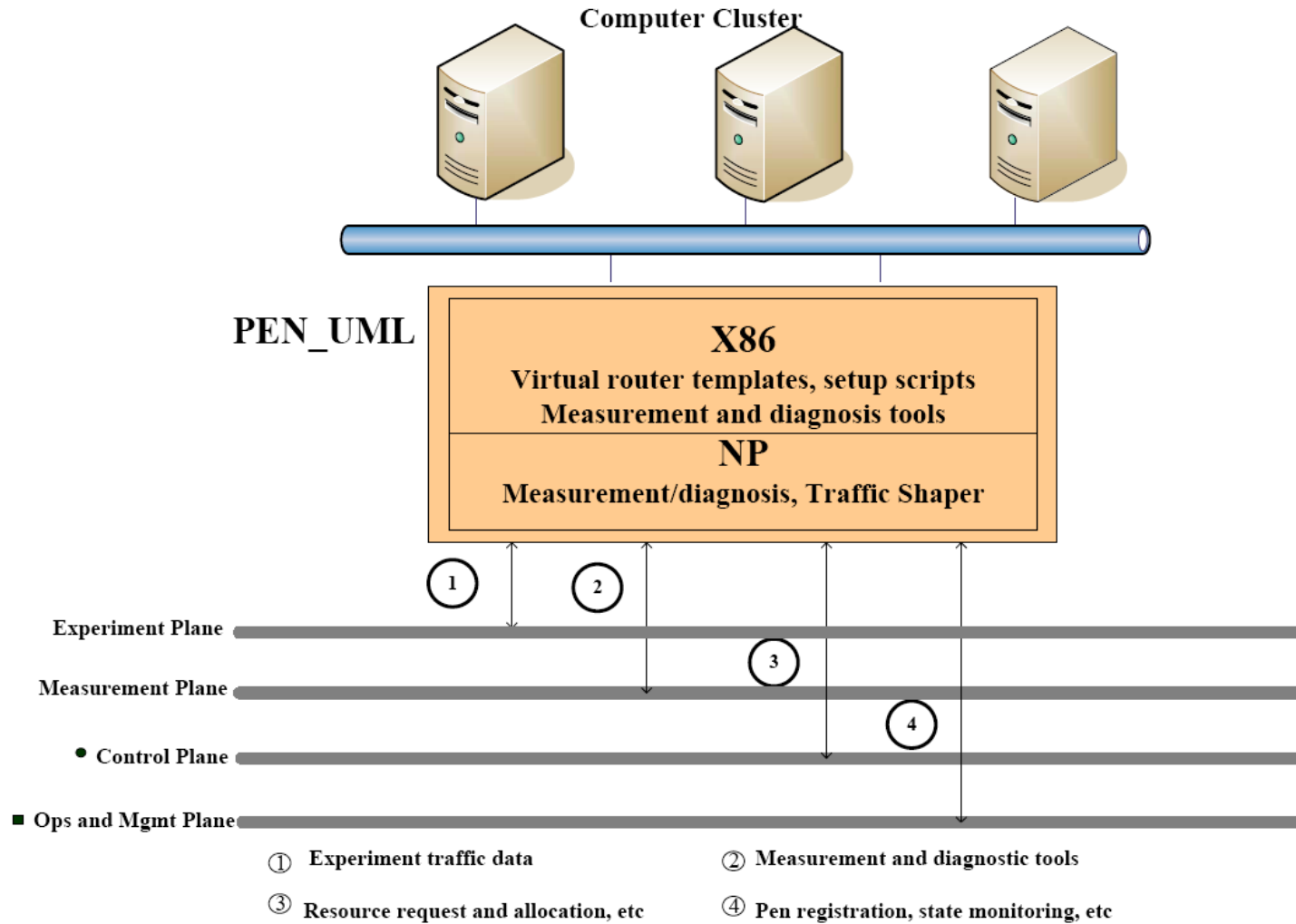- Problems Under Study
- In 6-12 months

Project Goals of Programmable Edge Node at UMass Lowell (PEN_UML)

- Design and implement a Programmable Edge Node (PEN) with hybrid multi-core processors (x86 and network processors)

- Provide control & management, measurement, diagnosis tools

- Trial integration with a GENI control framework
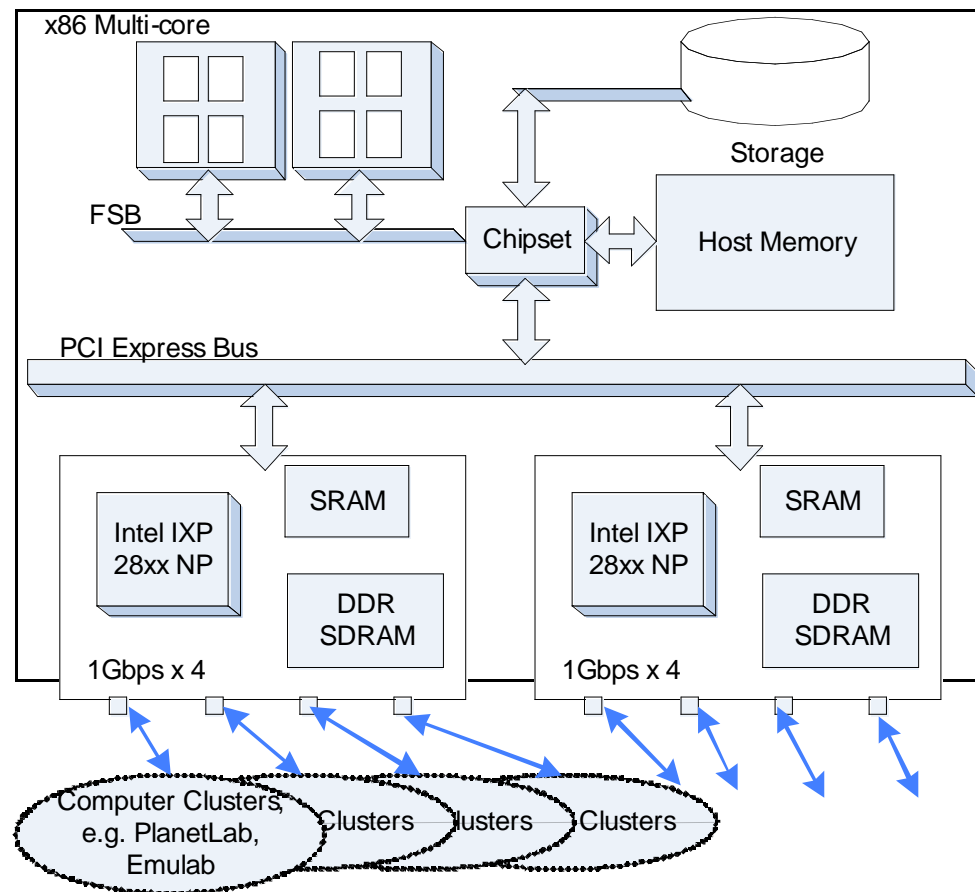
How PEN_UML Fits In GENI

GENI Admin and Operations
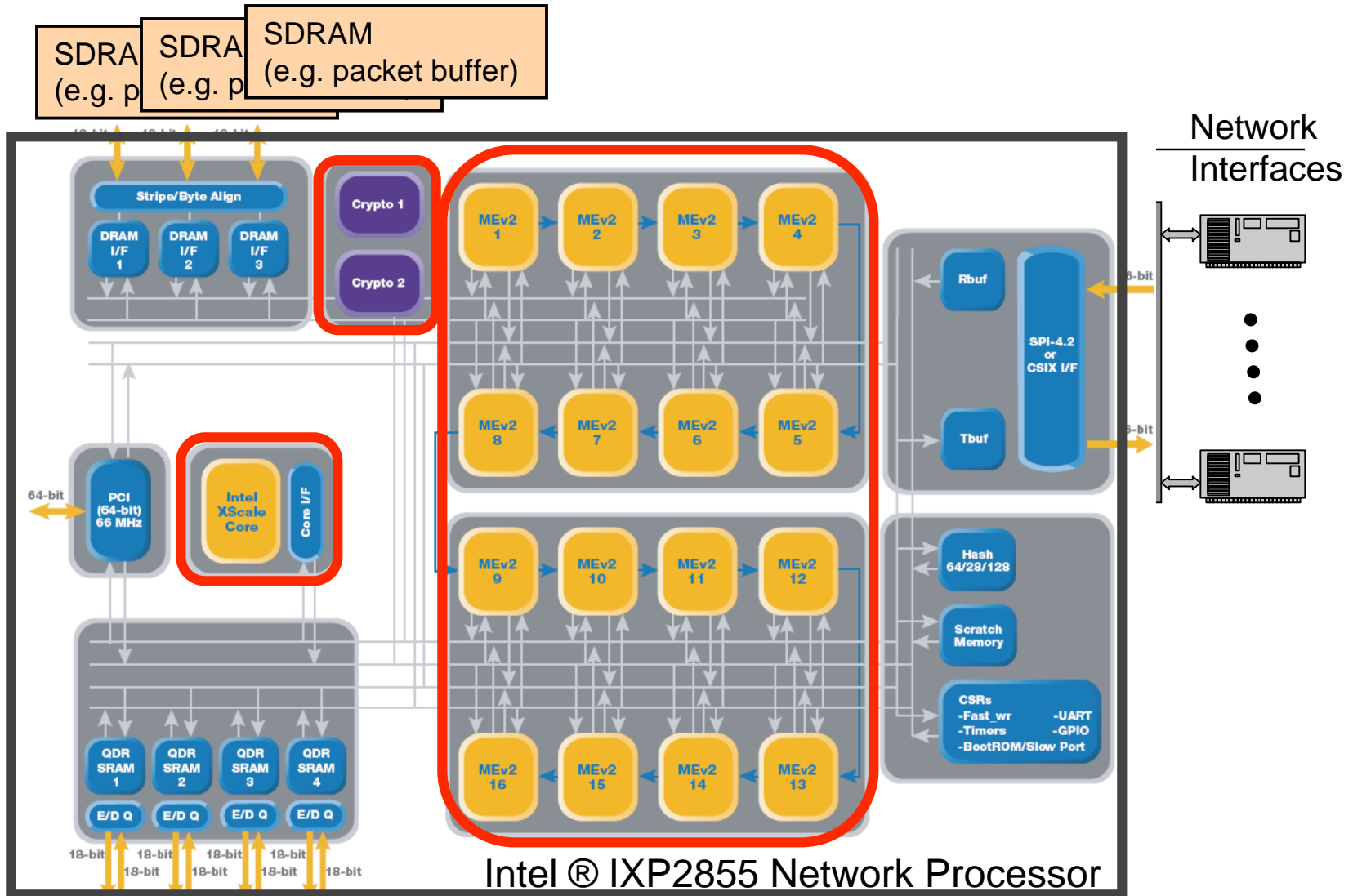
■ Admin   Operator

- Admin & Account Tools
- Ops & Mgmt Tools
- Help Desk

GENI Clearinghouse

Sevices

Registries

Research Org A

- Slice Admin
- PI
- Researcher

Planet Lab

Computer Cluster

■ ◆ PEN

PEN_UML

Experiment Plane

Measurement Plane

◆ Control Plane

■ Ops and Mgmt Plane

PEN_UML Operation Overview

PEN_UML Hardware Overview

Programmable Edge Node
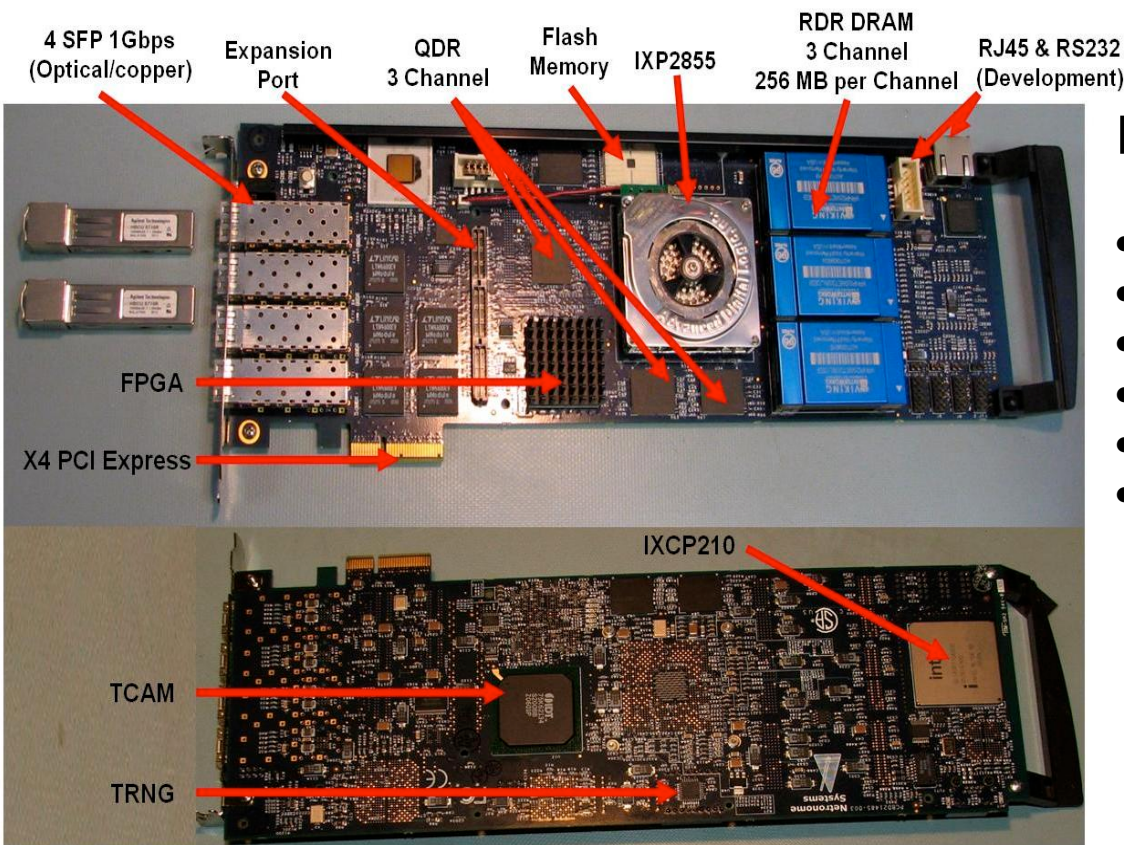(hardware architecture with off -the-shelf components)

x86 Multi-core

Storage

FSB

Chipset

Host Memory

PCI Express Bus

SRAM

Intel IXP
28xx NP

DDR
SDRAM

1Gbps x 4

SRAM

Intel IXP
28xx NP

DDR
SDRAM

1Gbps x 4

Computer Clusters,
e.g. PlanetLab,
Emulab

Clusters

Clusters

Clusters

Programmable Network Processor Architecture

Intel ® IXP2855 Network Processor

Netronome's Network Processor Card

NFE: Hardware Overview

4 SFP 1Gbps (Optical/copper) · Expansion Port · QDR 3 Channel · Flash Memory · IXP2855 · RDR DRAM 3 Channel 256 MB per Channel · RJ45 & RS232 (Development) · FPGA · X4 PCI Express · IXCP210 · TCAM · TRNG
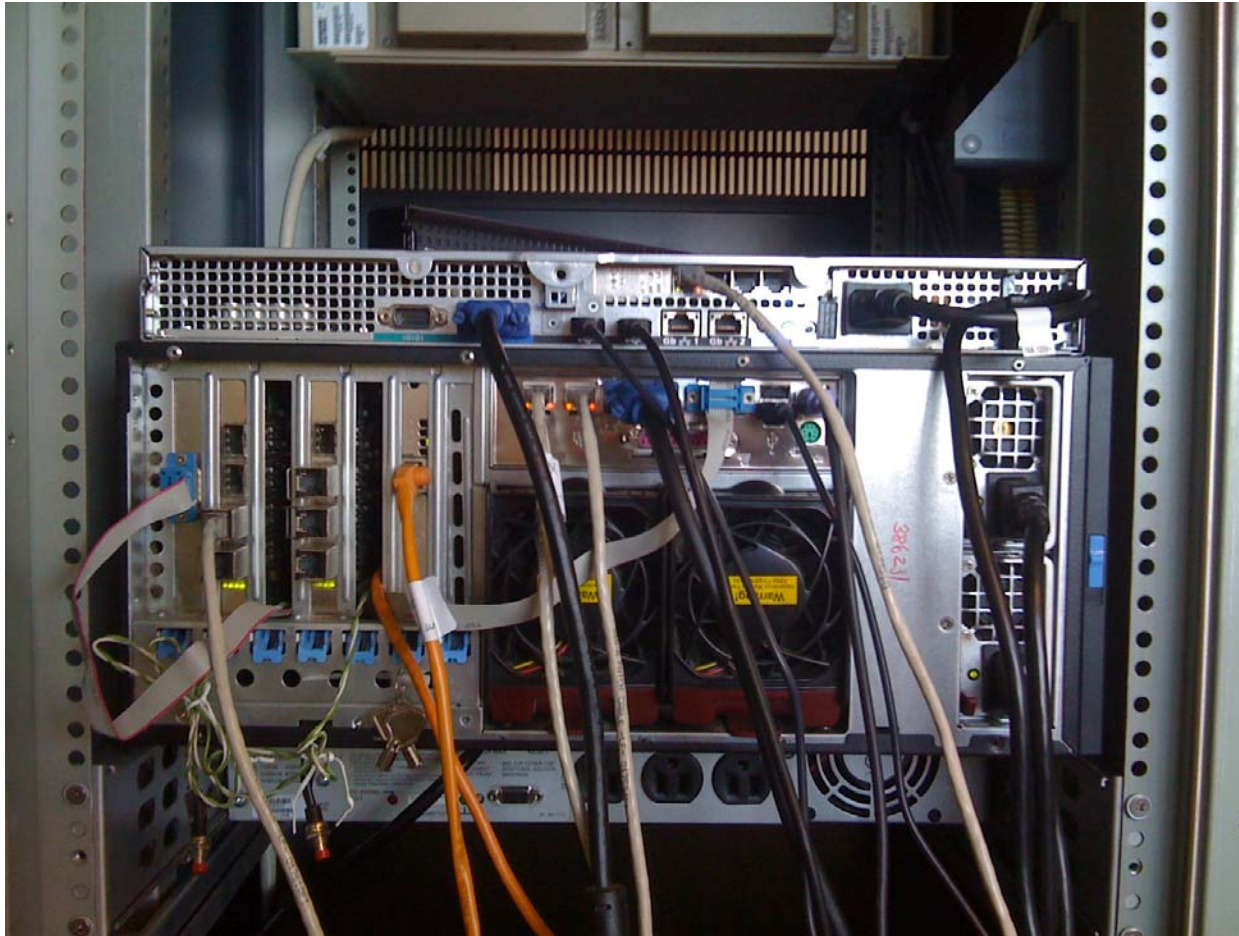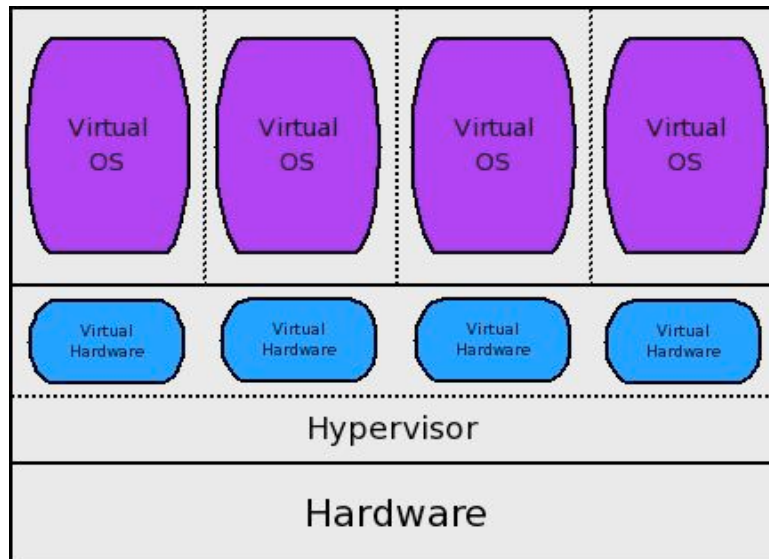
NFE-i8000 Specifications

- 1.4GHz IXP2855 NPU
- crypto units
- 4x1Gbps Ethernet ports
- 9Mb TCAM
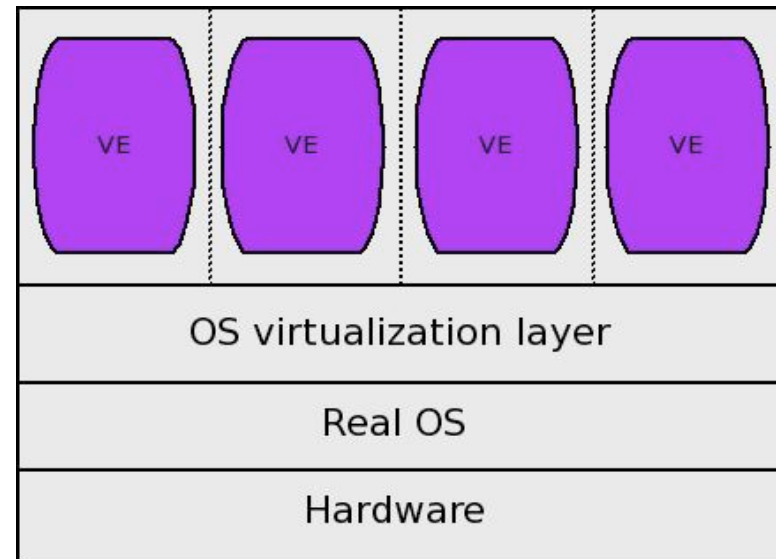- 40MB QDR2 SRAM
- 768MB RDRAM

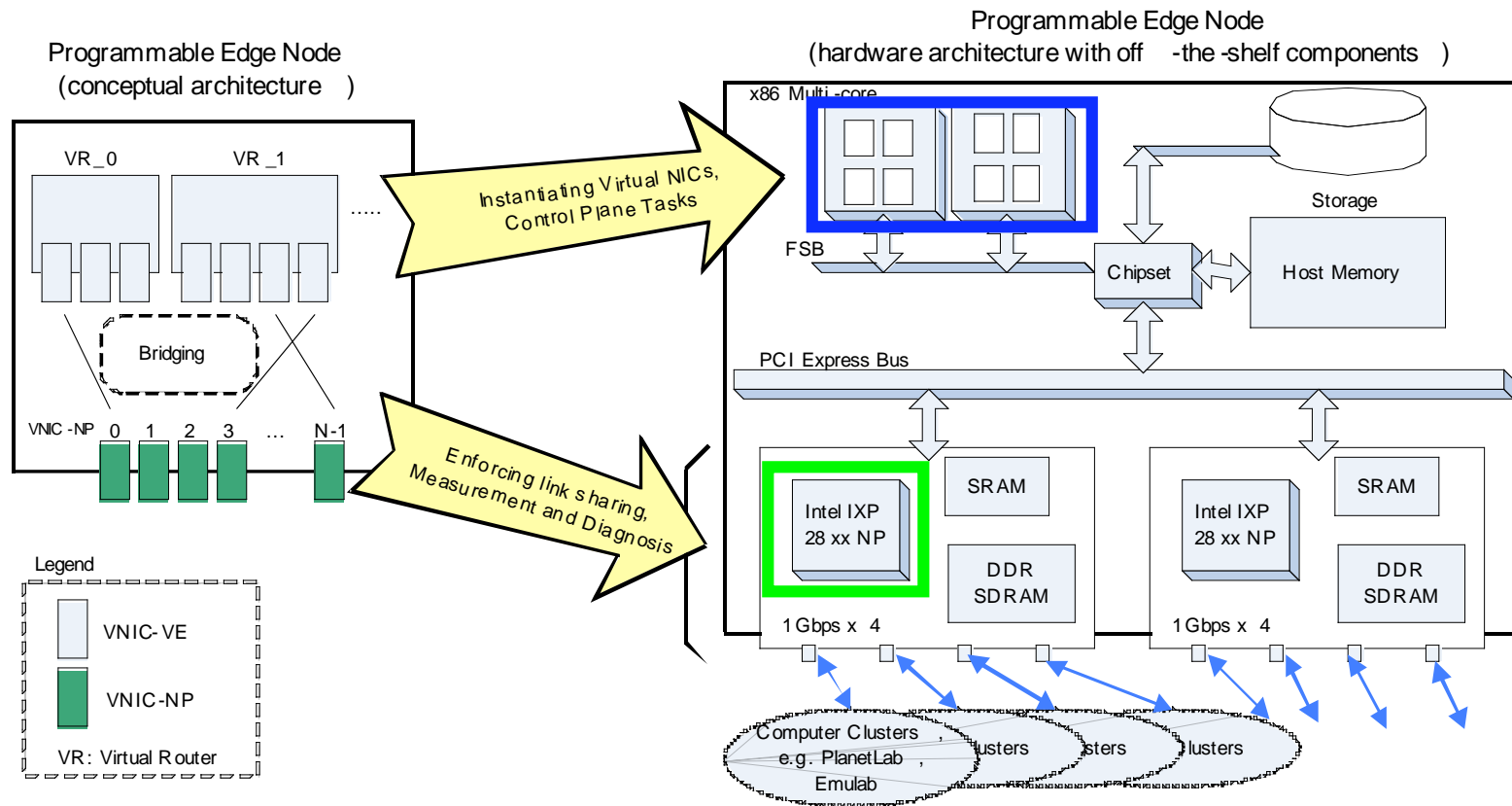# Picture of the Hardware

Virtualization Technologies

Full Virtualization
(e.g. VMWare, Xen)

OS Level Virtualization
(e.g. VServer, OpenVZ)

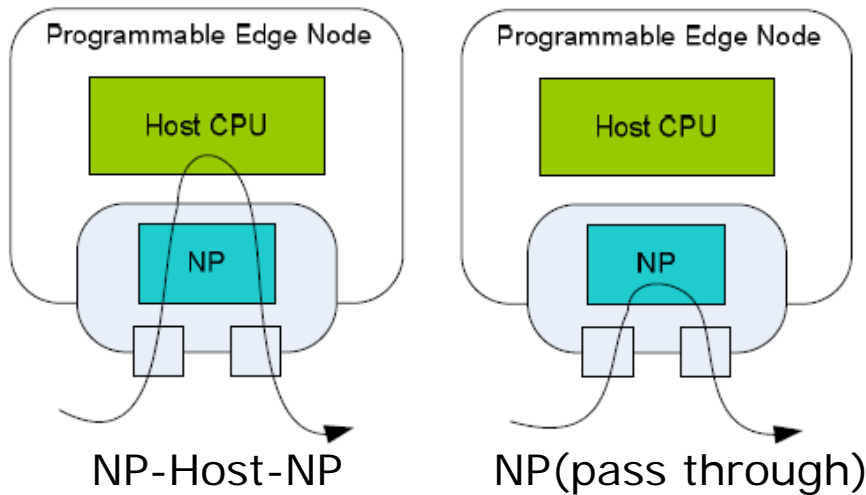# Virtual Routers Based on Hybrid Multi-core Processors of PEN_UML

Benefits of PEN_UML

○ Enforcement of Link Bandwidth Sharing
  - Host CPU supports virtual routers and NPs provide virtual NICs
  - NPs can offload traffic shaping from host CPU
○ Isolation of Experiment and Measurement
  - Experiment measurement can be performed on NP
  - Host CPU carries out user experiments
○ Flexible Number of Virtual NICs provided by NP
  - Virtual NICs on VEs can be one-to-one mapped to a VNIC-NP so that ARP broadcasts are avoided
○ Acceleration of Complex Packet Processing Workloads
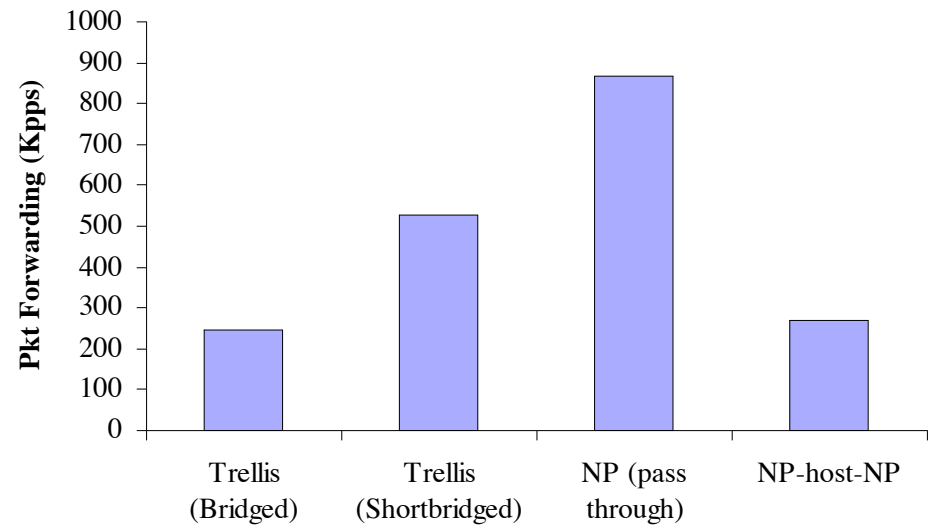 - NPs can be programmed to accelerate heavy packet processing load

Performance Evaluation (I) - Packet Forwarding

Node-Under-Test

eth0 — eth1

Source → Sink

Trellis[1]
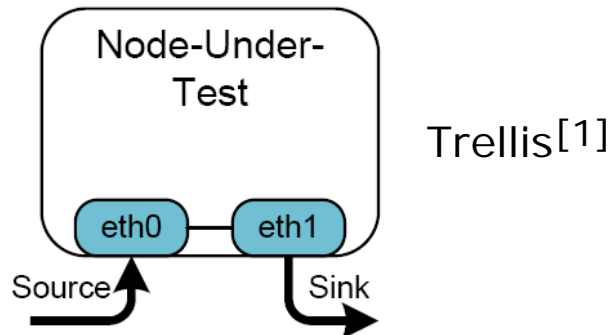
[1] Sapan Bhatia, et al. Hosting virtual networks on commodity hardware. Technical Report GTCS-07-10, Georgia Tech, Jan 2008.

Programmable Edge Node

Host CPU

NP

NP-Host-NP
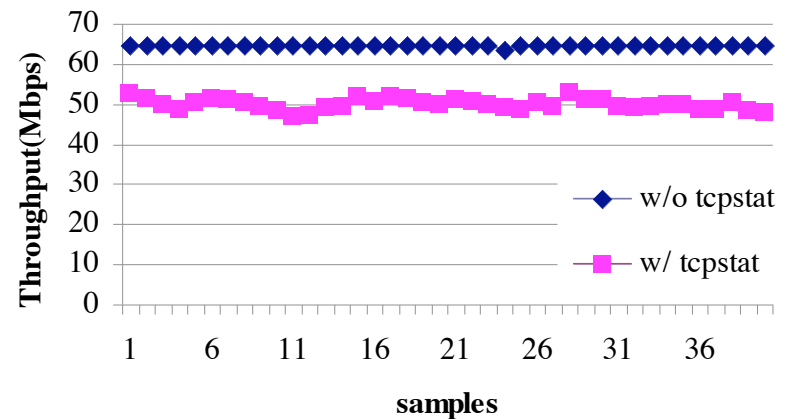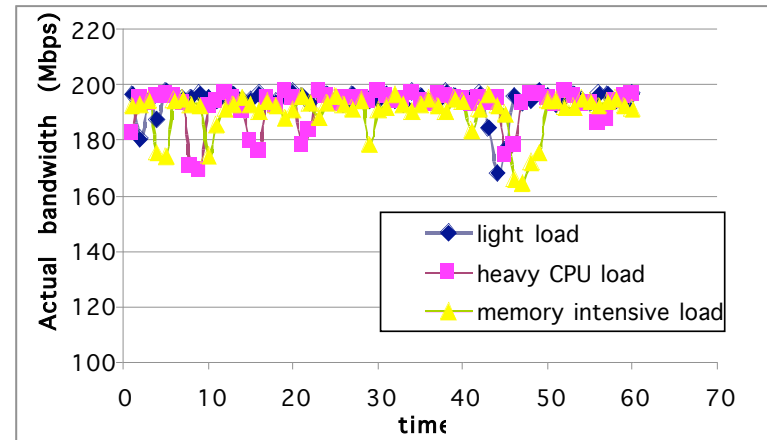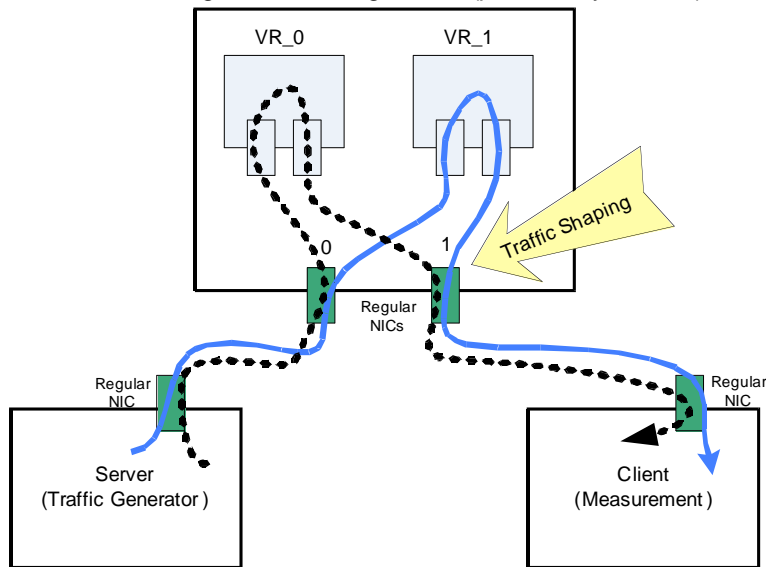
Programmable Edge Node

Host CPU

NP

NP(pass through)

[2] Y. Luo and C. Zhang, The Design of A Programmable Network Edge Node with Hybrid Multi-core Processors for Virtual Networks, ICCCN, Aug 2008

Pkt Forwarding (Kpps)

Trellis (Bridged)	~245
Trellis (Shortbridged)	~525
NP (pass through)	~865
NP-host-NP	~270

Performance Evaluation (II) - Bandwidth Allocation Enforcement & Isolation of Experiment and Measurement

*Conventional architecture **w/o** NP*

Programmable Edge Node (preliminary testbed)

VR_0   VR_1

Traffic Shaping

Regular NICs

0   1

Regular NIC

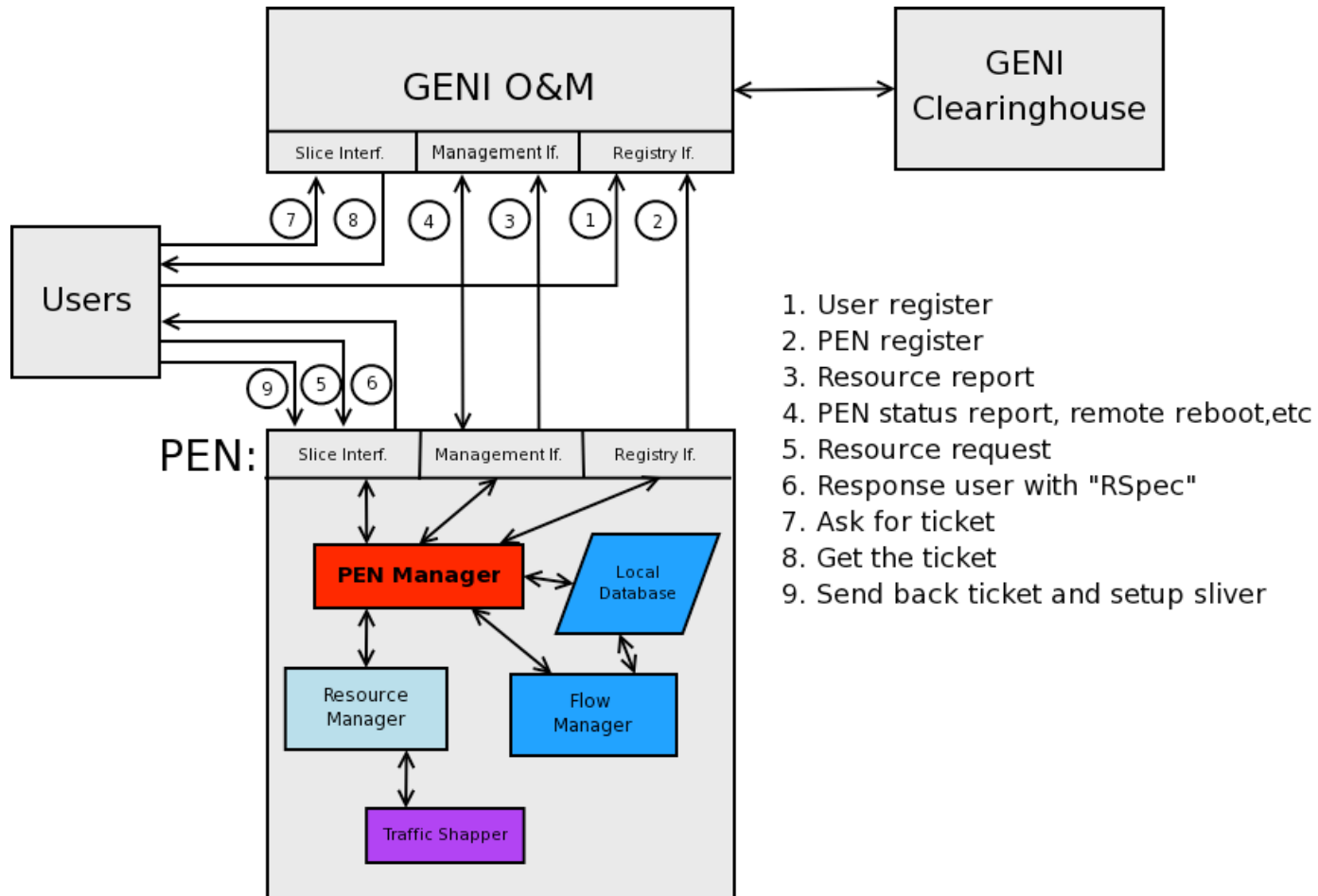Server (Traffic Generator)

Regular NIC

Client (Measurement)

Y. Luo and C. Zhang, The Design of A Programmable Network Edge Node with Hybrid Multi-core Processors for Virtual Networks, IEEE ICCCN'08

Issues Under Study

- Control software and APIs
  - Develop generic control software and APIs to fit in a specified GENI control framework.
- Measurement and diagnostic tools
  - Develop measurement and diagnostic tools as elements of GIMS and user help tools within a specified GENI control framework
- Network bandwidth sharing
  - Implement traffic shapers on hybrid multi-core processors to enforce link sharing within a specific GENI control framework.

PEN with GENI O & M

1. User register
2. PEN register
3. Resource report
4. PEN status report, remote reboot,etc
5. Resource request
6. Response user with "RSpec"
7. Ask for ticket
8. Get the ticket
9. Send back ticket and setup sliver

In 6-12 Months …

- Design and implementation
  - Virtual router templates & setup scripts
  - Control APIs and manager s/w
  - Measurement tools
  - Diagnosis tools
  - Traffic shapers on x86
- Trial integration with a control framework (e.g. ProtoGENI)
  - Conversation already started