

HermesNet



Ferramenta para configuração
e gerenciamento de redes
OpenFlow aberta e extensível

Marcial Porto Fernandez
marcial@larces.uece.br

Objetivos

- Propor um projeto de curto/médio prazo de se construir um plataforma de gerência de redes flexível e aberta.
- Diferencial: focar na arquitetura OpenFlow
 - Arquitetura proposta para Internet do Futuro
 - Pouco trabalho em plataforma de gerência OpenFlow
 - Apesar de ser possível gerenciar qualquer dispositivo de rede....

Objetivos

- Sugestões de vários projetos finais e dissertações de mestrado.
- Oferecer uma plataforma de teste para novas dissertações e projetos finais.
 - Um novo algoritmo de roteamento, um novo mecanismos de acesso a BD, etc
- Divulgar o Larces e a UECE
- Contribuir para o mundo Open Source

Sobre o nome

- Hermes é o mensageiro dos deuses na mitologia grega. Também conhecido como Mercurio na mitologia romana e Thoth no Egito
- Por essa função é considerado o guardião dos viajantes e das comunicações (dentre outros)
- HermesNet, como uma ferramenta de gerenciamento, pretende ser o guardião da rede.



What is Openflow?

- Part of The Stanford Clean Slate Program initiative
 - Nick McKeown (chair)
 - <http://openflowswitch.org/>
- Motivation:
 - Experiments we'd like to do
 - Experiments on L2 and L3
 - New naming/addressing schemes
 - Network access control
 - Problem with our network
 - Paths are fixed (by the network provider)
 - IP-only
 - Addresses dictated by DNS, DHCP, etc
 - No means to add our own processing in network nodes

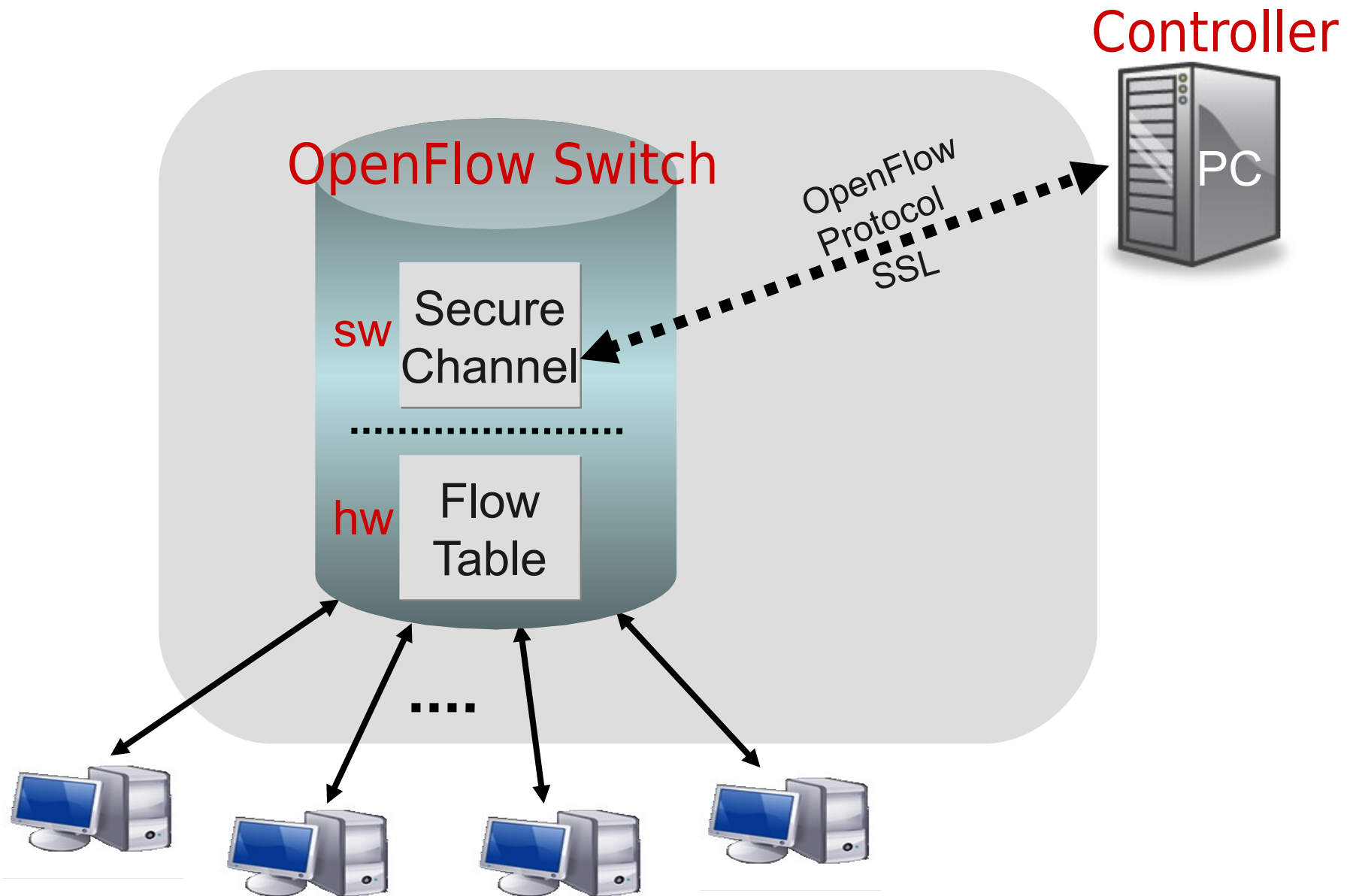
What is Openflow?

- A “pragmatic” compromise
 - Allow researchers to run experiments in their network...
 - ...without requiring vendors to expose internal workings.
- Basic Idea
 - An open protocol to remotely add/remove flow entries in on-the-shelf switches

Openflow Approach

- Commercial vendor won't open software and hardware development environment
 - Complexity of support
 - Market protection and barrier to entry
 - NDAs from chip-sets manufactures (Broadcom, Marvel, ..)
- Hard to build my own
 - Prototypes are flakey...
 - Software only: Too slow
 - Fanout too small

Openflow Approach



Openflow Flow Table Entry



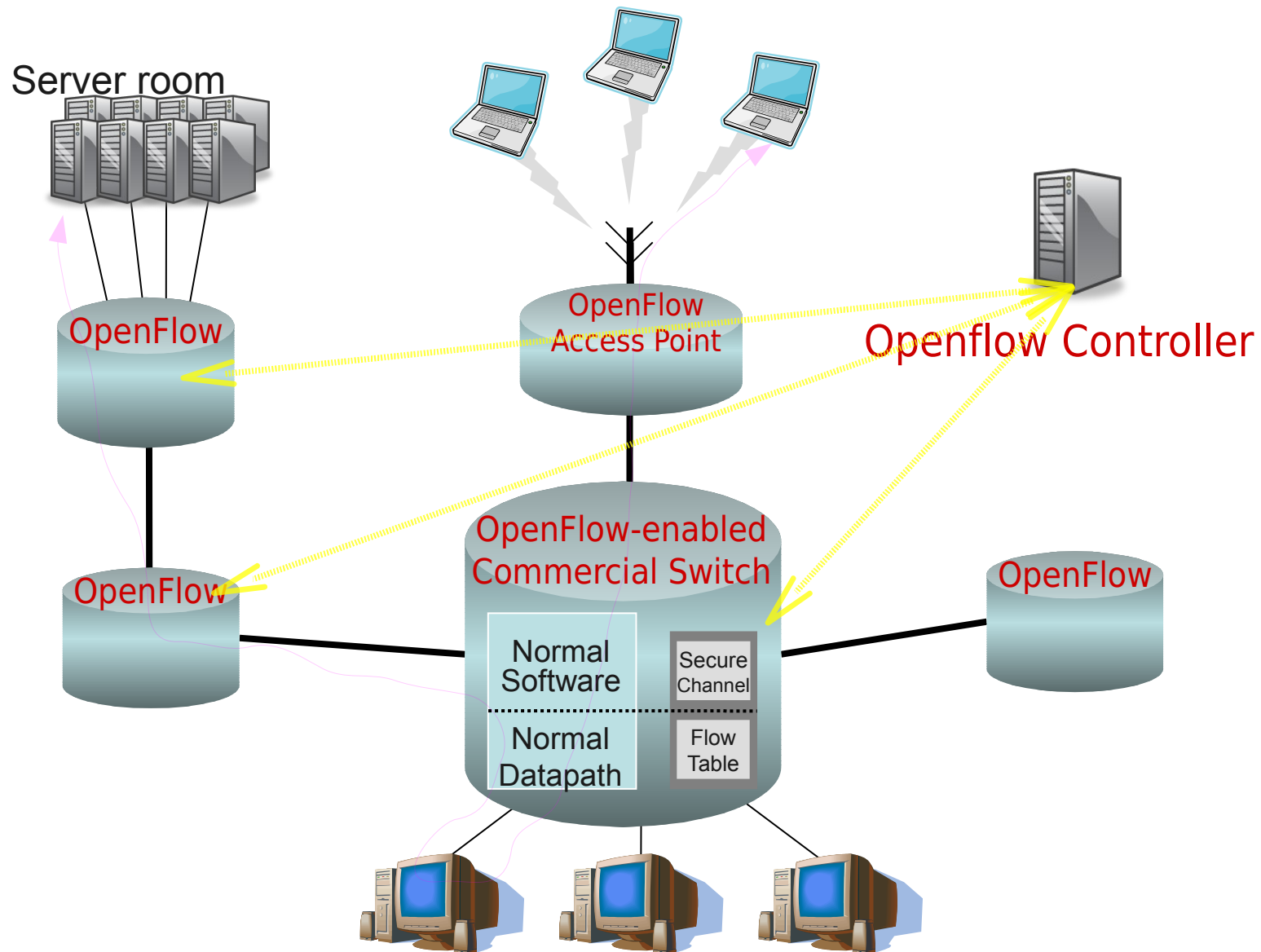
Packet + byte counters

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Rewrite header
5. Send to normal processing pipeline

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport
-------------	---------	---------	----------	---------	--------	--------	---------	-----------	-----------

+ mask

Openflow Switch Usage



Openflow Switch Status

- OpenFlow Specification Version 1.1.0
 - Published February 28, 2011
 - License: Free for research and commercial use
- OpenFlow Implementations
 - Open Virtual Switch (<http://openvswitch.org/>)
 - Linux OpenWRT (Access Point)
 - Pronto Switch / Indigo Firmware: Openflow switch based on Broadcom Reference Design.
 - NetFPGA

Openflow Switch Status

- Openflow Controller
 - NOX OpenFlow controller (Python and C++)
 - SNAC Web Based OpenFlow Policy Controller (GUI for NOX)
 - Beacon Java based OpenFlow Controller
 - FlowVisor Openflow Controller Proxy (C++)

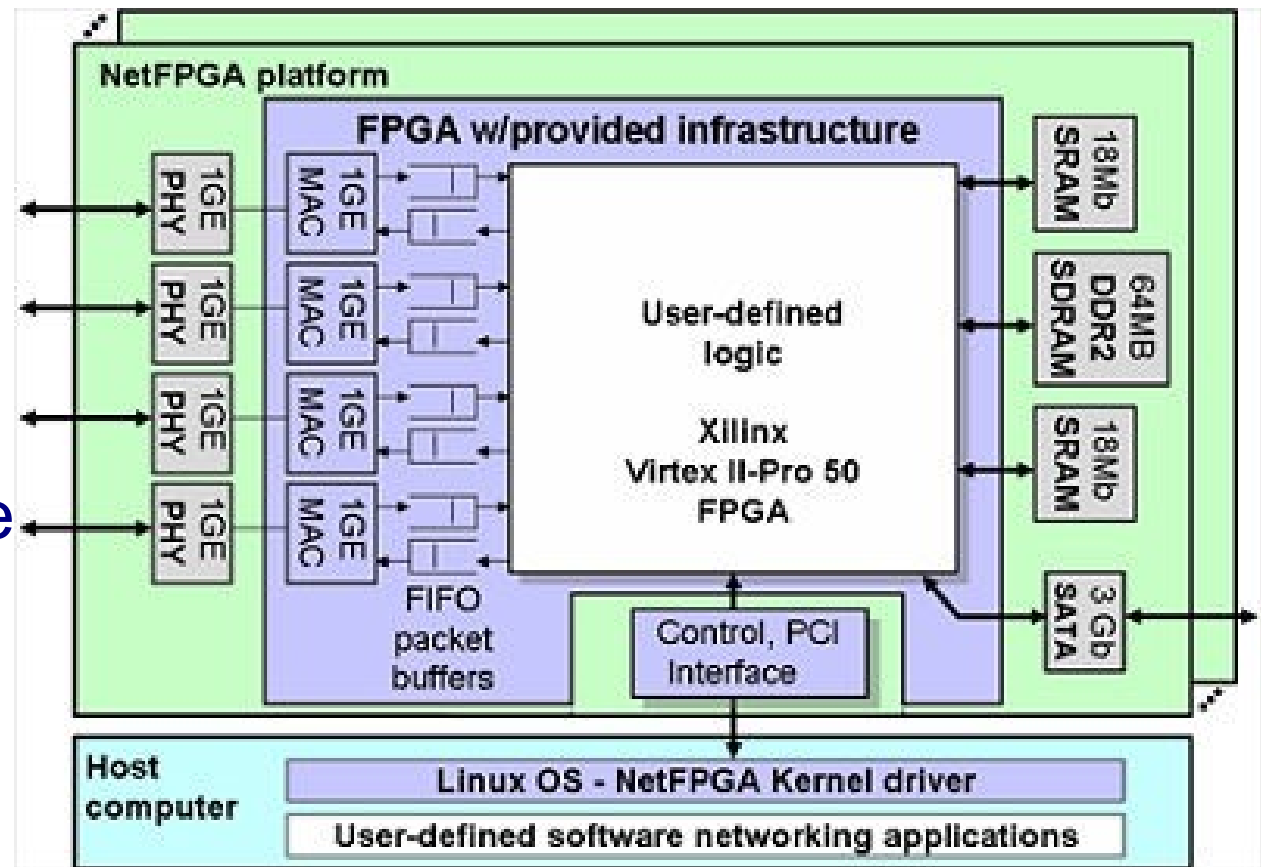
NetFPGA Pr

- Network FPGA Experimental Board
 - 4 x 1 GBPS Ethernet ports
 - PCI interface
- FPGA forward packets (fast path)
 - Verilog/VHDL
- Linux OS (slow path)
 - Linux driver
- Board vendor
 - Digilent US\$ 499,00
- System vendor
 - Accent Technology US\$ 1.500,0



NetFPGA^

- Xilinx Virtex II FPGA
- 4 x 1 GBPS Interfaces
- 4.5 MB SRAM
- 64 MB DRAM
- PCI Interface
- SATA interface
- Linux Kernel Drive



NetFPGA System

- Supermicro Case and Board
- Intel or AMD Dual/Quad Core processor
- 4 GB DDR
- 500 GB SATA
- CentOS pre-instal
- 1 NetFPGA board



NetFPGA - 1U

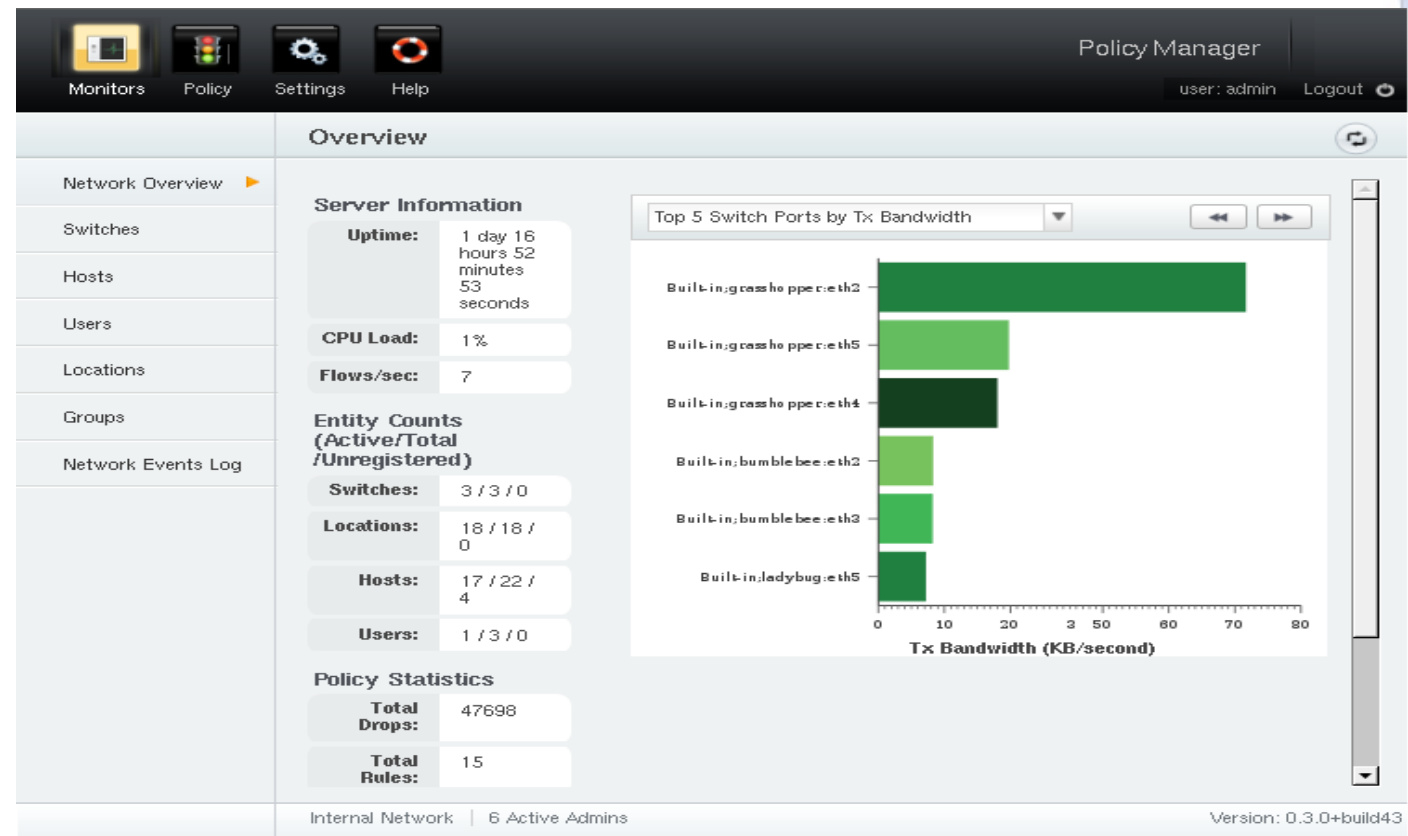
Pronto Switch



- Supports OF ver 1.0
- 48 x 1Gbps ports, 4 SFP+ ports
- External GE and serial management ports
- Bootable CF card slot
- Based on Broadcom 56534 switch hardware
- 8541 PPC CPU running at 825 MHz
- Use Indigo Firmware

SNAC Web Based OpenFlow Policy Controller

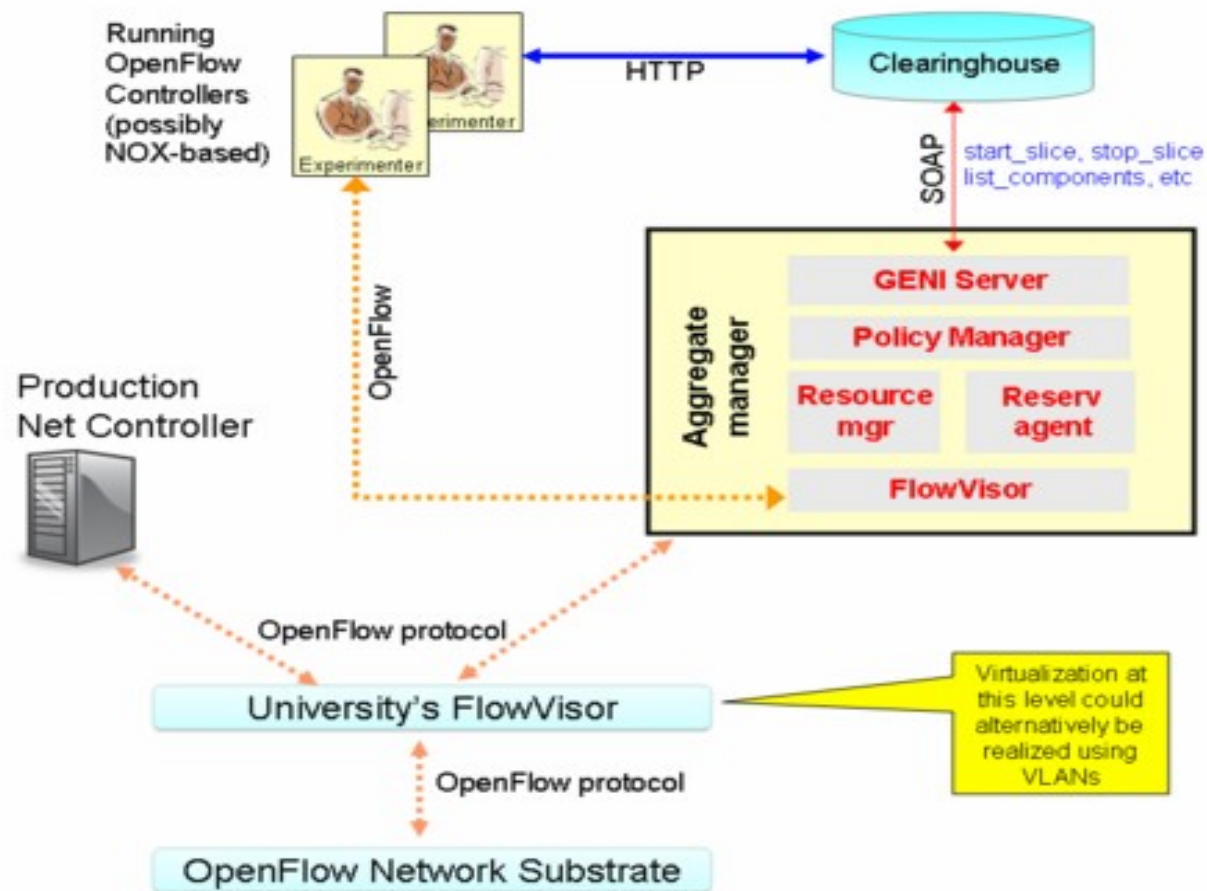
- Web Based Openflow controler
- Features
 - Monitors
 - Policy
 - Settings
- Use NOX
- Other GUI Tools
 - Lavi
 - Envi



Future projects

- Liboftrace
 - Lib to monitor and control Openflow traffic
- Oflop
 - Openflow benchmarking controller
- OpenFlow Click
 - Openflow proxy for Click software router
- Config Protocol (Schema)
- OpenFlowMPLS
 - OpenFlow support for MPLS switch (Ericson)

E-GENI Testbed



- Create a testbed of OpenFlow switch in GENI/PL using SFA interface
 - SOAP-based SFA interface between ClearingHouse and the E-GENI Aggregate Manager
 - OpenFlow protocol between the experiment's controller and the E-GENI FlowVisor

Características Gerais

- Utilizar o máximo de projetos Open Source
 - Facilita a implementação
- Utilizar uma licença aberta GPLv3 ou Apache 2.0
- Hospedar no SourceForge.
- Linguagem de desenvolvimento:
 - C/C++ para as partes sensíveis ao desempenho
 - Java para o demais módulos

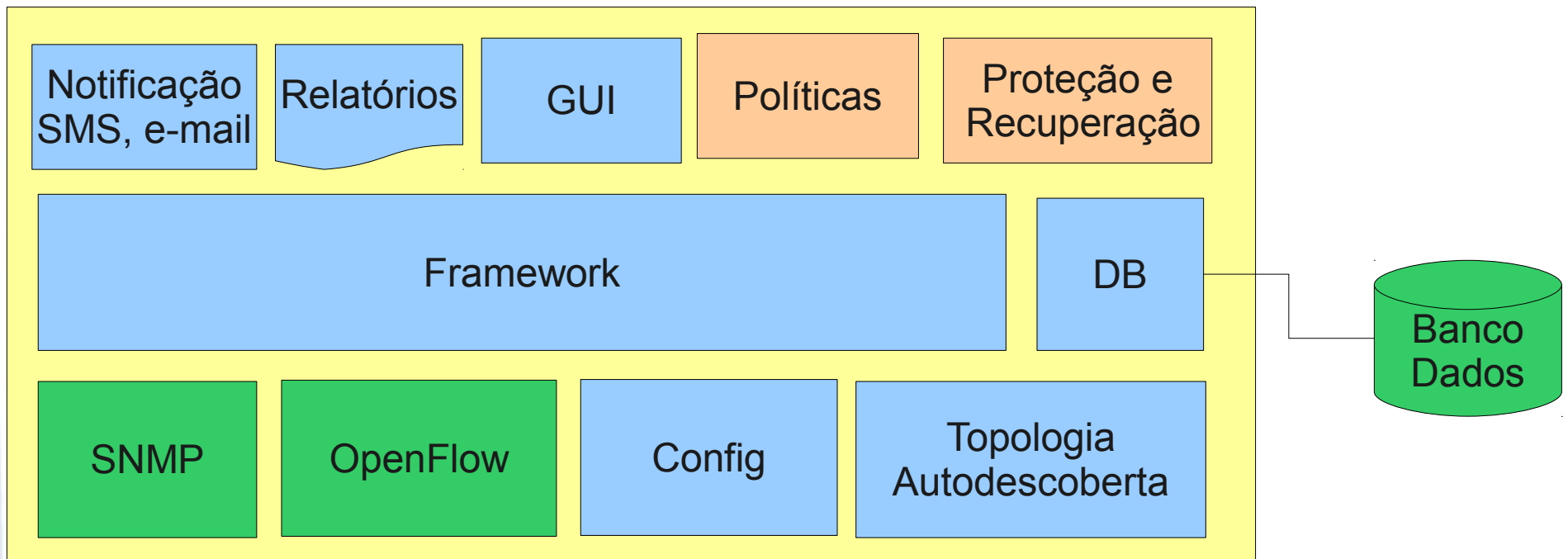
Características Gerais

- Sistemas Operacionais em foco:
 - Linux Ubuntu (deb) e Red Hat (rpm)
 - Windows
- Interface: Web+JavaScript, Web Start, ...?
- Apesar de ser uma plataforma experimental, ela deve focar uso profissional “carrier grade”.
- Framework Java: Struts, Spring?

Características Gerais

- Usar Maven para build e documentação.
- Por que não usar o OpenNMS?
 - Única ferramenta gerenciamento OpenSource “carrier grade”.
 - Arquitetura do projeto muito interessante (copiar as boas ideias...)
 - Ferramenta de gerenciamento genérica e complexa demais...
 - Monitoramento baseado em ping. :-)

Arquitetura HermesNet



Framework

- Oferecer recursos para facilitar desenvolvimento de GUI e gerar relatórios.
 - Enviar SMS de alerta?
- Implementar um daemon para receber notificações Trap/SNMP.
- Definir uma API para as aplicações: descoberta de topologia, políticas, etc.
- Outras funcionalidades requeridas para um framework....

SNMP

- O SNMP é necessário para coletar informações de gerenciamento dos switches.
- Net-SNMP oferece uma biblioteca C que permite todas as funções SNMP.
 - Incluir no pacote HermesNet?
- Será necessário desenvolver aplicações:
 - LLDP para descoberta de MAC address de hosts
 - IPFIX, NetFlow, IPFlow para desempenho

OpenFlow

- Aproveitar código controlador do protocolo OpenFlow.
- É um ponto onde o desempenho é importante:
 - Quanto maior o desempenho do controlador maior é a rede que ele é capaz de gerenciar.
- Sugestões: NOX (C++) e Beacon (Java)

Config Protocol

- Protocolo para configurar dispositivos OpenFlow
 - Enviar arquivo de configuração para dispositivos.
 - O servidor (que fica no switch) deve ser leve e escrito em C
 - O cliente deve ser Java (módulo HermesNet)
- Deve ser flexível para permitir inclusão e alteração de novos parâmetros.
- Formatos possíveis: XML, JSON, YAML

Descoberta Topologia

- A descoberta de topologia é importante para qualquer ferramenta de gerenciamento.
 - Particularmente para OpenFlow, baseado em conexão L2, é essencial para roteamento.
- Deve usar SNMP:
 - LLDP para descobrir as ligações e montar o grafo da topologia.
 - SNMP para atualizar estado.
- Armazenar a topologia no BD.

Interface Banco de Dados

- Usar Postgres
 - Questiono a longevidade do MySQL na mão da Oracle....
 - Outra opção?
- Deve ter flexibilidade para incluir novos dados de MIB
- Deve ser bem desenhada para não impactar no desempenho.
- Deve ter forte interação com framework

Políticas

- Módulo para implementar políticas:
- Analisar dados de desempenho, por exemplo, IP-FIX, S-Flow, etc.
- Definir regras para configurar switches.
- Muitas dissertações.....

Proteção e Recuperação

- Implementar mecanismo para recuperar falhas.
- Recebe notificação de falhas via SNMP
- Alterar tabelas de roteamento.
- Pode-se usar vários algoritmos com resultados diferentes.
- Muitas dissertações.....

Situação atual

- Criado página no SourceForge:
- <https://sourceforge.net/p/hermesnet/home/>
 - Acesso via Git
 - Usar Wiki e Ticket
- Meta de lançamento primeira versão: final de 2011

Ambiente de Teste

- Servidores e desktops do Larces.
- Dispositivos OpenFlow:
 - VM de desenvolvimento
 - PC com OpenV Switch (disponível)
 - PC com placa NetFPGA (disponível)
 - Pronto Switch (comprado)
 - Switch Netgear (aguardando mais informações)
 - OpenWRT (algum voluntário?)

Sugestões de Trabalho

- Framework: escolha framework;
 - definições de API;
 - daemon.
- Protocolo de configuração, sistema de configuração e cliente (dispositivo).
- Estrutura e interface BD
- Descoberta de topologia.
- Motor de políticas
- Mecanismo de recuperação.

Conclusão

- Não acredito que o projeto seja um sucesso!!!
 - Pelo menos espero que seja tema de várias monografias e dissertações...
 - Em princípio apenas para UECE, tornar publico futuramente.
- Entender como criar e manter um projeto Open Source.
- Quando ficar pronta uma versão completa, procurar empresas para financiar.
 - Podem surgir propostas de emprego....