UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

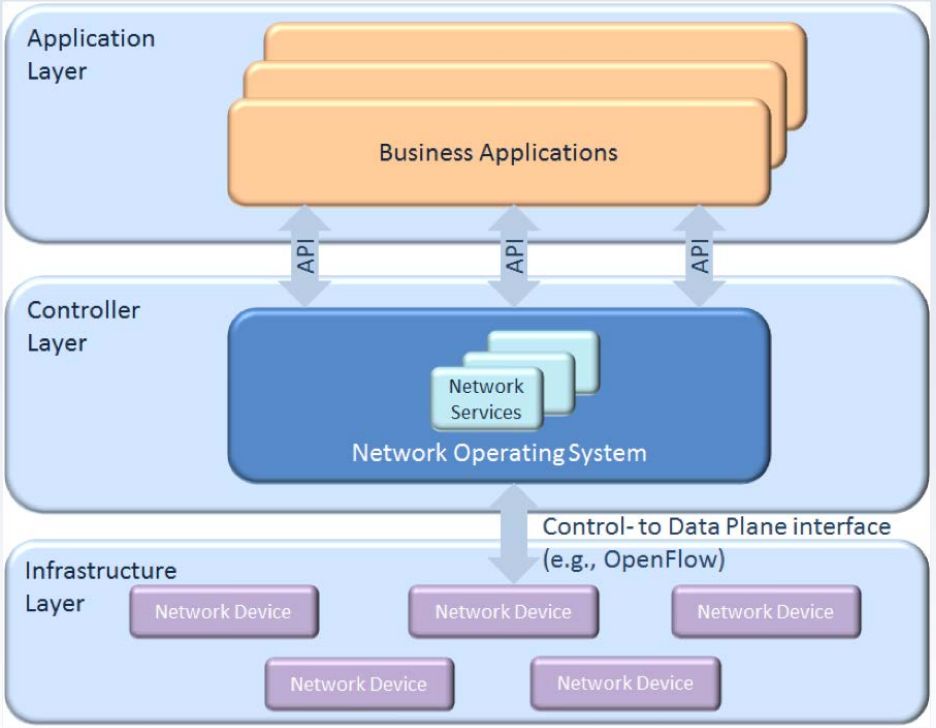
Past, Present, and Future Topics in Software-Defined Networking

John Bellessa and Prof. Roy H. Campbell

illinois.edu

Software-Defined Networking

- Basic principles
  - Decouples control plane and data plane
  - Centralizes view/control of the network
  - Makes network programmable
- Concept of Network "Operating System" emerges



NetODESSA: Dynamic Policy Monitoring with Inference

illinois.edu

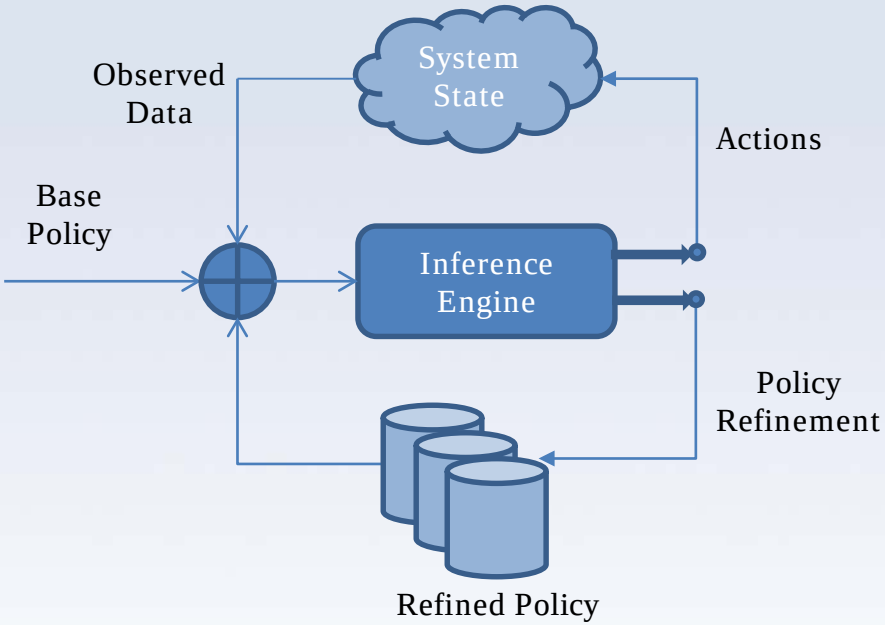
Dynamic Policies

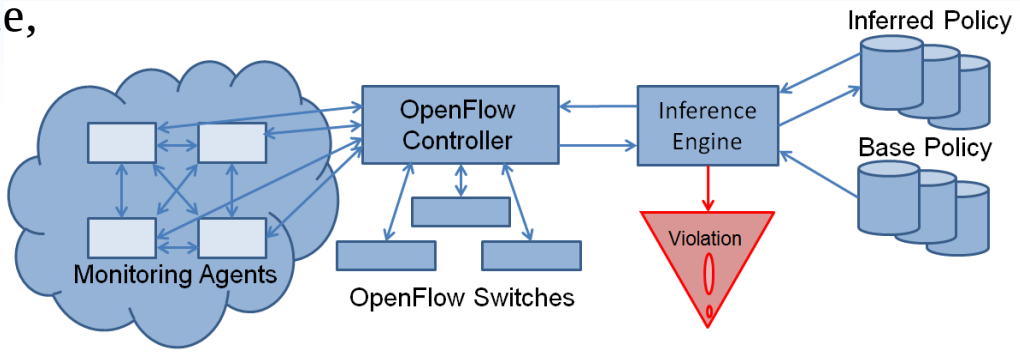
- Modern networking environments are increasingly dynamic

- Network policies should be highly expressive without requiring explicit rules
  - Should not require admin intervention
  - Should be able to *infer* explicit rules

Inference-based Dynamic Policies

- Define general *base policies* as logical assertions
- Monitor and report events that occur in the network
- Use inference engine to determine whether assertions are violated
  - As system observes events, inference engine generates *refined policy*
  - In order to ensure compliance, *actions* are generated





Example

Base Policy:
Anonymous hosts can't use SSH

+

Event:
Anonymous host A joins the network

=

Refined Policy:
Anonymous host A can't use SSH

Refined Policy:
Anonymous host A can't use SSH

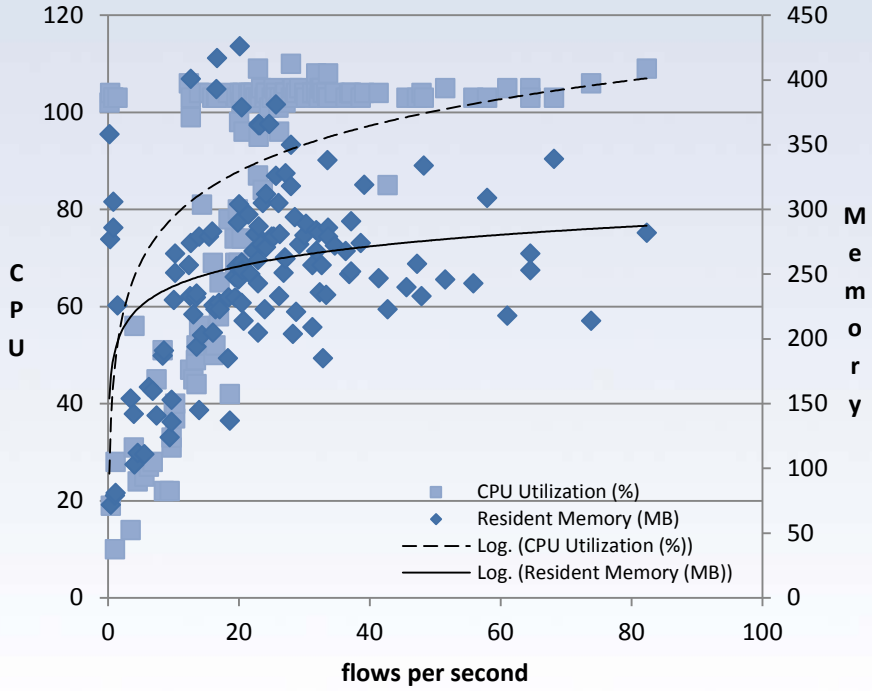
+

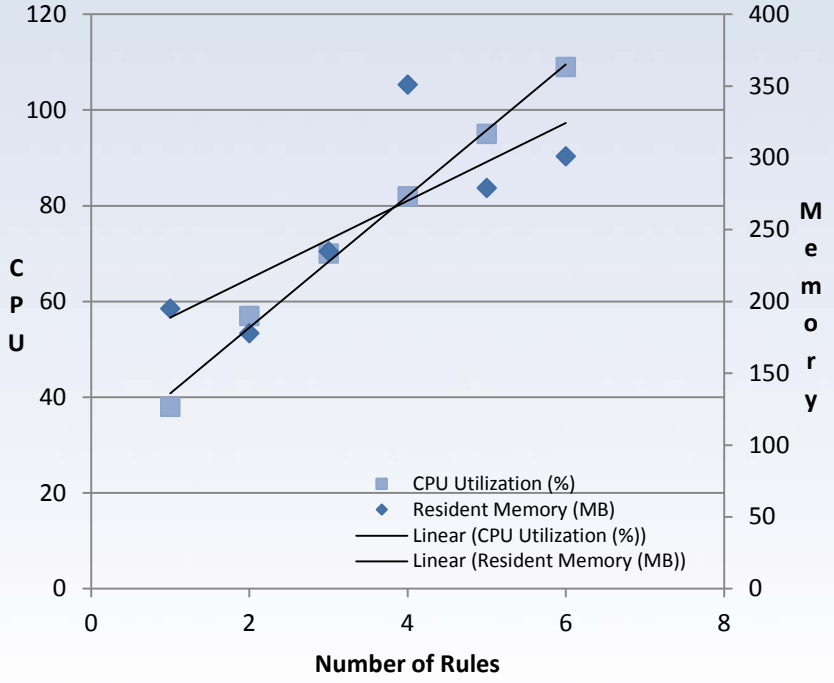
Event:
Anonymous host A tries to use SSH

=

Action
SSH flow blocked

Results





illinois.edu

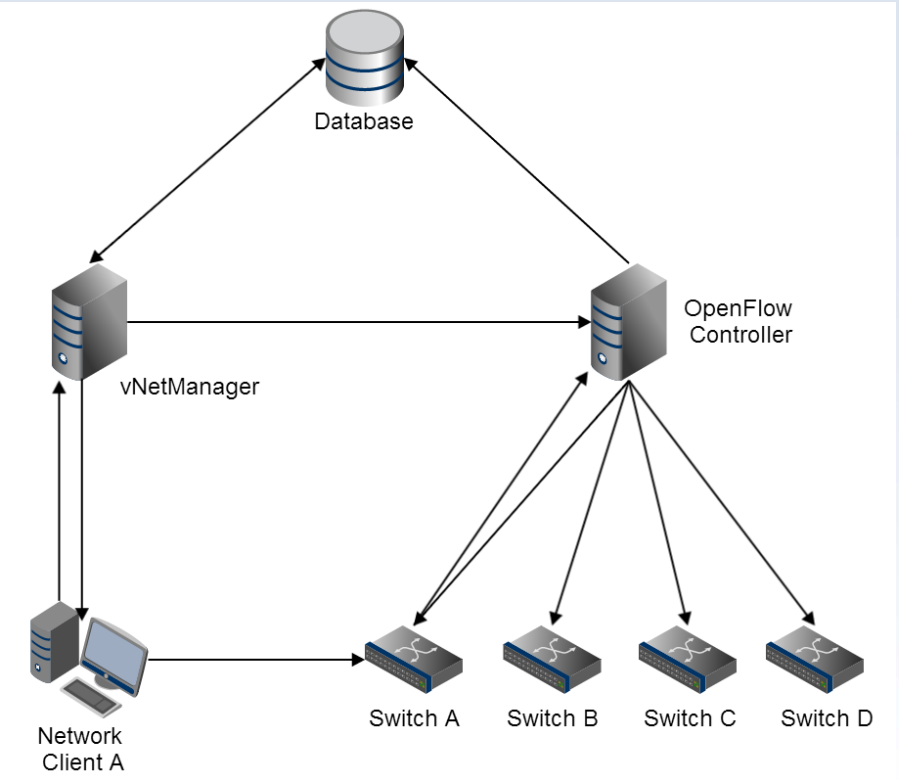
Dynamic Network Virtualization

SDN-based Network Virtualization

- Goal: Dynamic virtualization infrastructure
  - Single, domain-wide configuration
  - User-based access control
  - Flow-based enforcement
- Effectively the same functionality as VLANs, but…
  - Simplified deployment, less prone to error, and easier to debug
  - Management becomes less difficult to reason about
  - Rapid/automated deployment of behavior alterations

System Architecture

- vNetManager application
- OpenFlow Controller monitors physical network and pushes flows to switches
- Database stores configuration and network state
- Tunnel endpoints within each domain



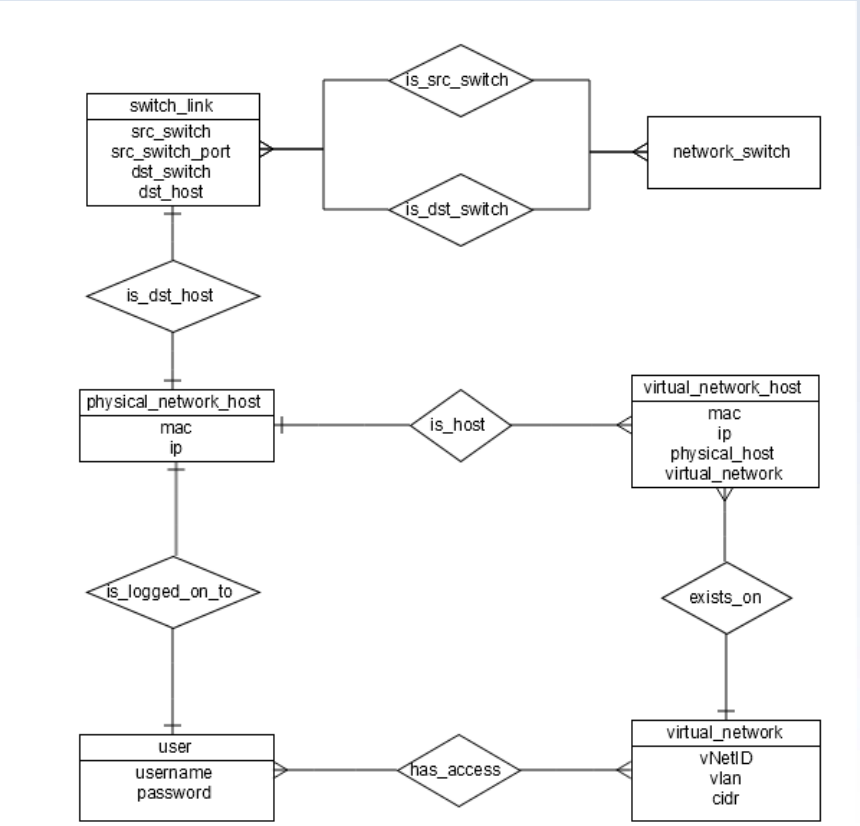
illinois.edu

vNetManager

- Implemented as a REST service in Python
- Exposes REST API using the Flask framework
  - Authenticate user
  - Deauthenticate user
  - List allowed networks
  - Join network
  - Leave network
- Changes in network state invoked by user actions

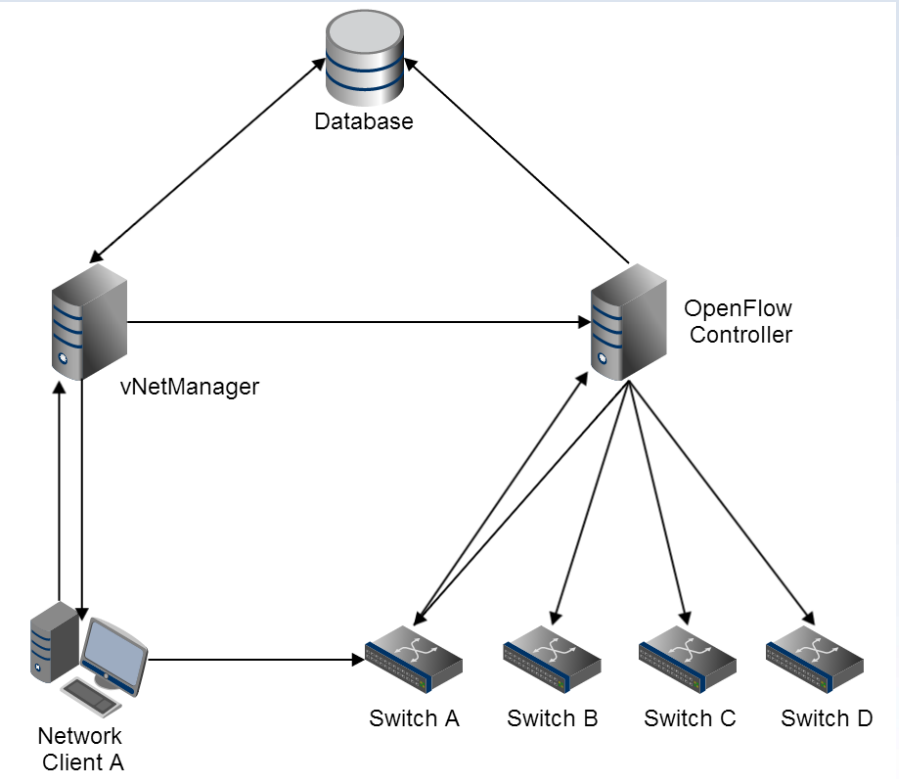
Database

- Implemented in MySQL
- Stores domain-wide configuration
  - Virtual network configurations
  - User access control
- Maintains network state
  - Switch graph
  - Physical host information
  - Virtual host bindings
  - Active users



System Operation

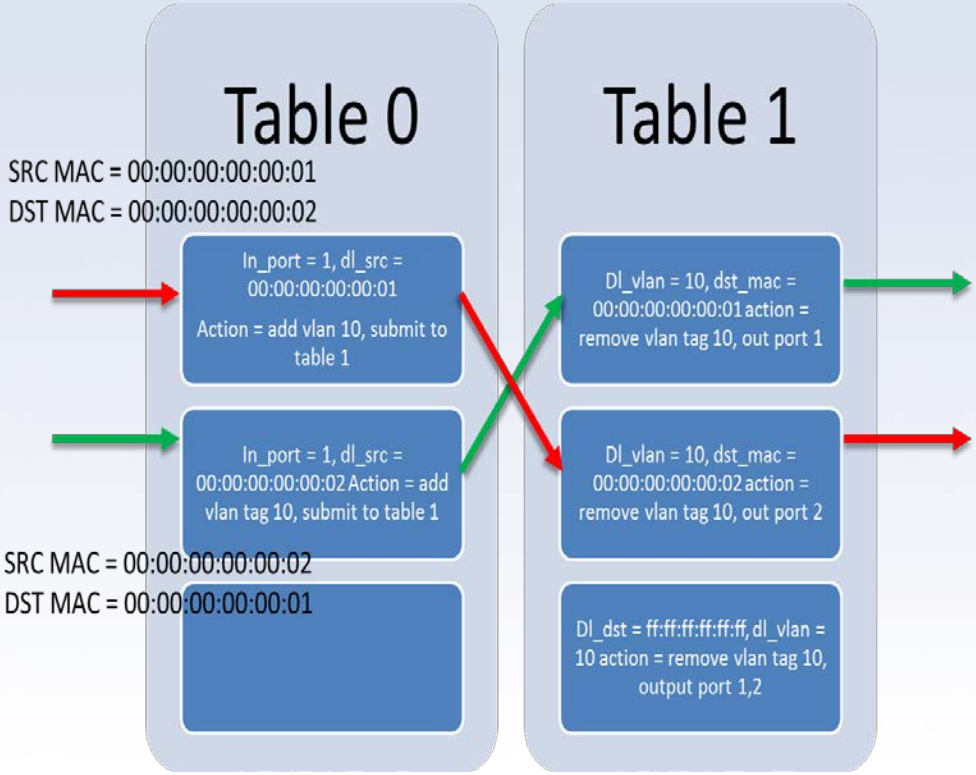
- When hosts become active, the controller notes their location
- Users log-in and join “virtual networks” via client-side application that communicates with vNetManager
- Flows are installed to ensure connectivity among hosts on the same virtual networks



illinois.edu

System Operation

- Leverages OpenFlow 1.1's multi-table processing feature
- Flows are calculated and pushed to switches when users join virtual networks
- Paths determined using Dijkstra's single-source shortest path algorithm



illinois.edu

What Lies Ahead?

New SDN Lab

- Established with support from HP and Matrix Integration

- Infrastructure
  - 13 Switches (all support OpenFlow 1.3)
    - 2x HP 12900 series
    - 2x HP 5900 series
    - 3x HP 5406 series
    - 4x HP 3800 series
    - 2x HP 2920 series
  - 4x HP MSR3024 routers
  - 3x HP 830 WAPs

HP FlexFabric 12900 Core Switch

New SDN Lab

- Will enable new research
  - Datacenter and inter-datacenter simulation
  - Inter-domain control systems
  - High-performance benchmarking
  - Controller scalability stress testing
  - Advanced OpenFlow features at line rate

New SDN Lab

- Challenges/Opportunities
  - Allocating resources and access
  - Enabling rapid experimentation
  - Ensuring reproducibility
  - Ensuring experimental isolation
  - Exposing switch-level logging and debugging

New SDN Lab

- Next steps
  - Understanding capabilities and limitations of HP VAN SDN Controller and Open Daylight
    - Built-in virtualization
    - Controller chaining
  - Deploy SDN applications
  - Generate traffic

Future Topics

- Elasticity-promoting Abstractions
  - Decouple distributed services from underlying resources
  - Use network addresses to identify services rather than hosts on which services run
  - Allows elastic allocation of resources to services with requiring updates to clients

Future Topics

- SDN-Based Load Balancing
  - Is the concept viable?
  - Do existing solutions meet requirements at scale?
  - Is it always necessary to defer to controller?
  - May offer a more “holistic” approach

Future Topics

- Network “Context Switching”
  - Treat network like a CPU: preempt network operation with different configuration and restore state
  - If implemented properly, and with the cooperation of network hosts, this would allow multiple experiments to make progress without interfering with one another.
  - This would present significant challenges such as saving and restoring state across the network between context switches.

Future Topics

- High-Scale Rule Consistency
  - A well-known challenge in SDN is rule consistency.
  - When controllers are making system-wide control decisions in large networks, inconsistent rule states may result in transient errors or leave the network susceptible to failure.
  - One area of research is determining strategies to guarantee high degrees of consistency among switches and controllers

Thank You!

illinois.edu