Distributed Resource Controllers

An SDN architecture with delegation, abstraction and support for multiple domains

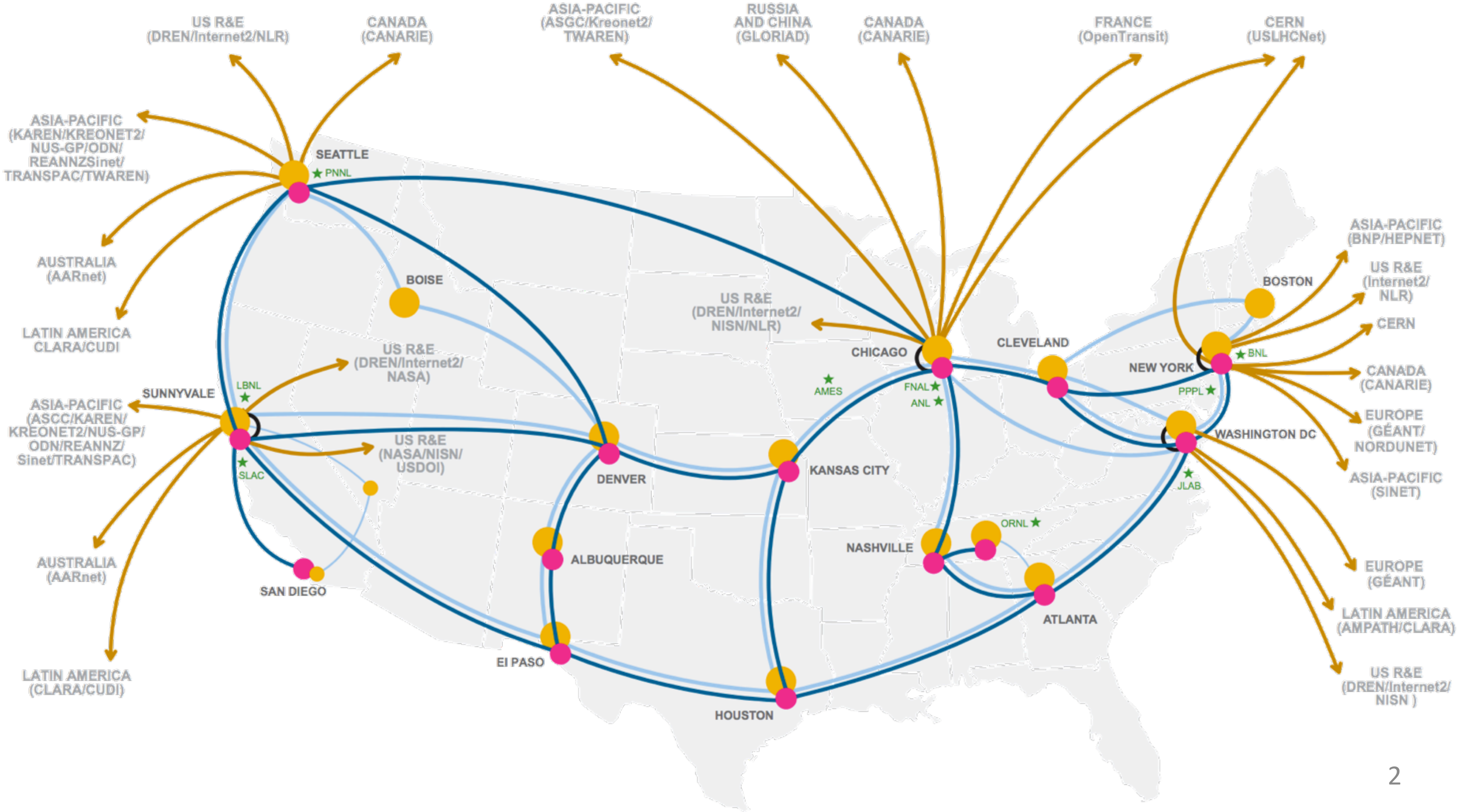
Dr. Gregory Lauer, Raytheon/BBN

Dr. Ilia Baldine, RENCI

March 18, 2013

ESNet

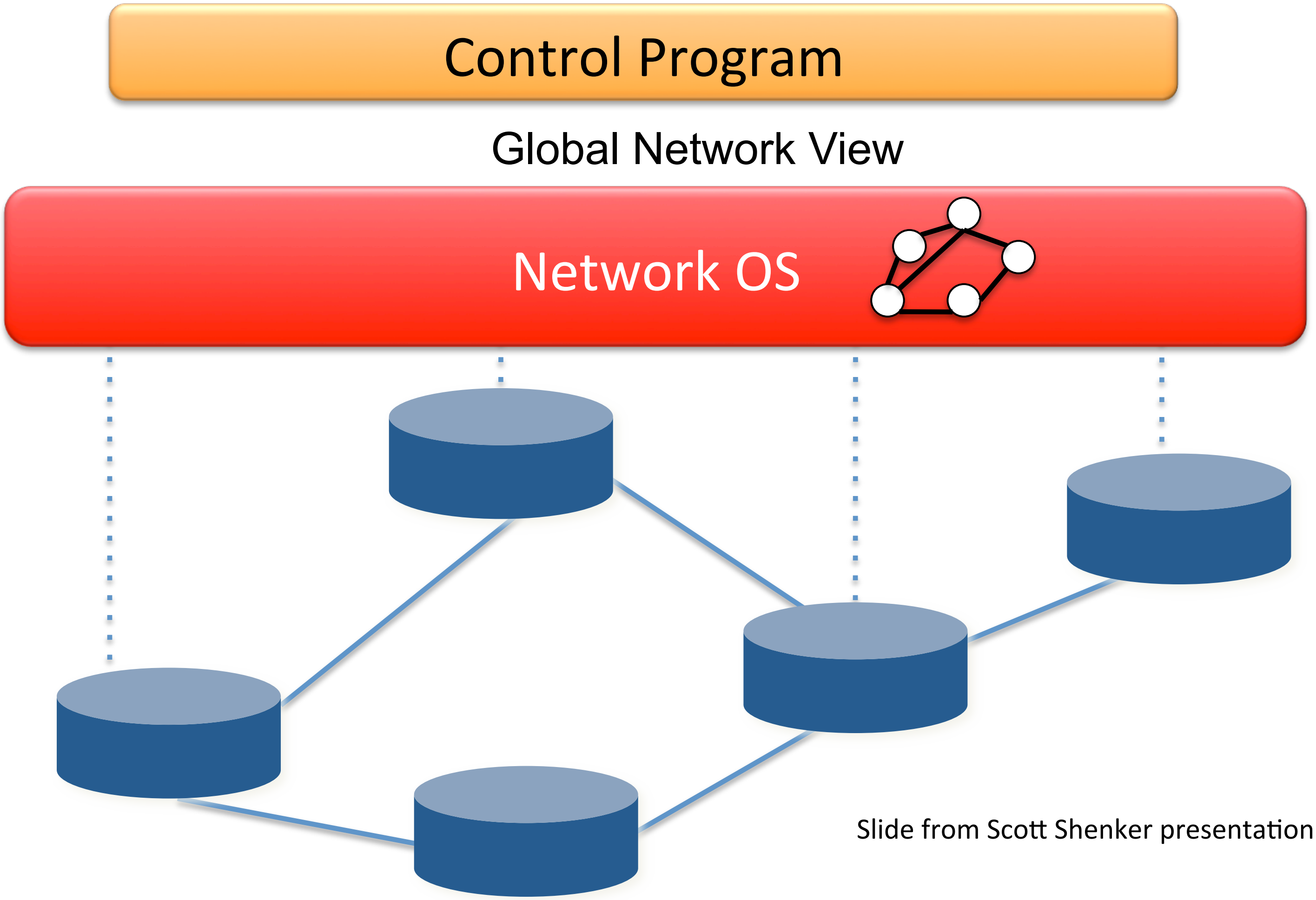
- Connects 40 DoE institutions to 100s of research and education networks
- Traffic growing 10x every 47 months


US R&E (DREN/Internet2/NLR)
CANADA (CANARIE)
ASIA-PACIFIC (ASGC/Kreonet2/ TWAREN)
RUSSIA AND CHINA (GLORIAD)
CANADA (CANARIE)
FRANCE (OpenTransit)
CERN (USLHCNet)
ASIA-PACIFIC (KAREN/KREONET2/ NUS-GP/ODN/ REANNZSinet/ TRANSPAC/TWAREN)
SEATTLE
PNNL
AUSTRALIA (AARnet)
LATIN AMERICA CLARA/CUDI
BOISE
US R&E (DREN/Internet2/ NASA)
SUNNYVALE
LBNL
ASIA-PACIFIC (ASCC/KAREN/ KREONET2/NUS-GP/ ODN/REANNZ/ Sinet/TRANSPAC)
SLAC
US R&E (NASA/NISN/ USDOI)
DENVER
AUSTRALIA (AARnet)
SAN DIEGO
ALBUQUERQUE
LATIN AMERICA (CLARA/CUDI)
EI PASO
HOUSTON
US R&E (DREN/Internet2/ NISN/NLR)
CHICAGO
AMES
FNAL
ANL
KANSAS CITY
NASHVILLE
ORNL
ATLANTA
CLEVELAND
BOSTON
NEW YORK
BNL
PPPL
WASHINGTON DC
JLAB
ASIA-PACIFIC (BNP/HEPNET)
US R&E (Internet2/ NLR)
CERN
CANADA (CANARIE)
EUROPE (GÉANT/ NORDUNET)
ASIA-PACIFIC (SINET)
EUROPE (GÉANT)
LATIN AMERICA (AMPATH/CLARA)
US R&E (DREN/Internet2/ NISN)

Key Issues

- Enable collaboration by thousands of remote users
  - Wide range of services for users
  - Need to share resources between different users
- Must coordinate resources from different organizations
  - Move lots of data long distances
- Need to deploy new services quickly
  - Empower end users

Software Defined Network 1.0



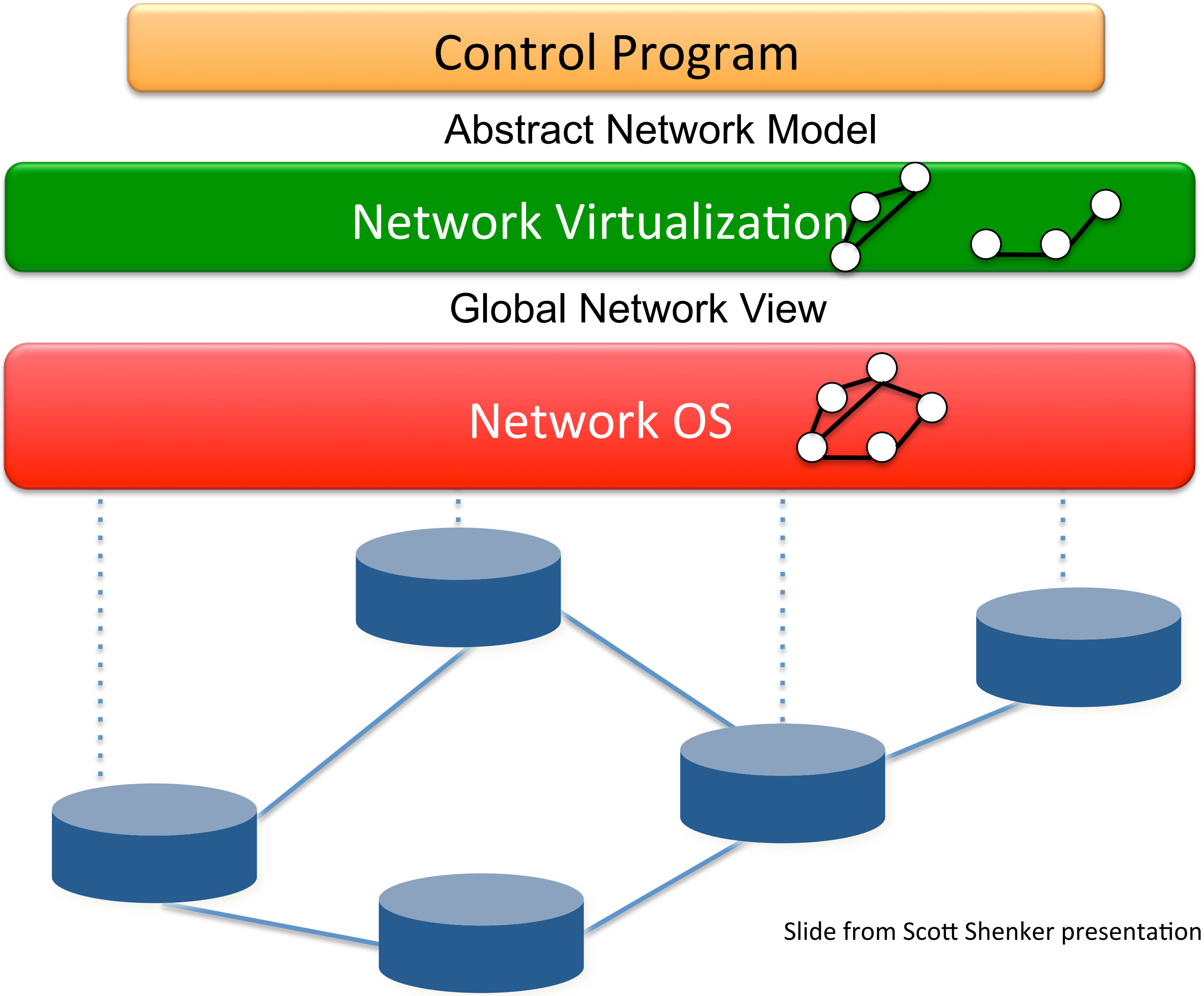
Control Program

Global Network View

Network OS

Slide from Scott Shenker presentation

Software Defined Network 2.0



Control Program

Abstract Network Model

Network Virtualization

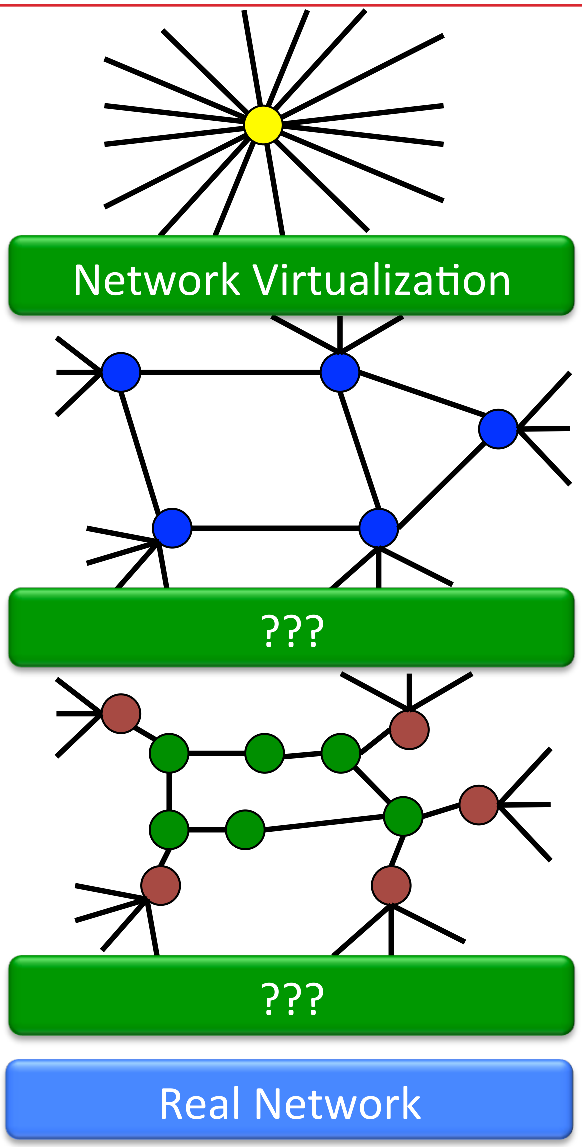
Global Network View

Network OS

Slide from Scott Shenker presentation

More on Abstractions

- Top level is virtual network view



Network Virtualization

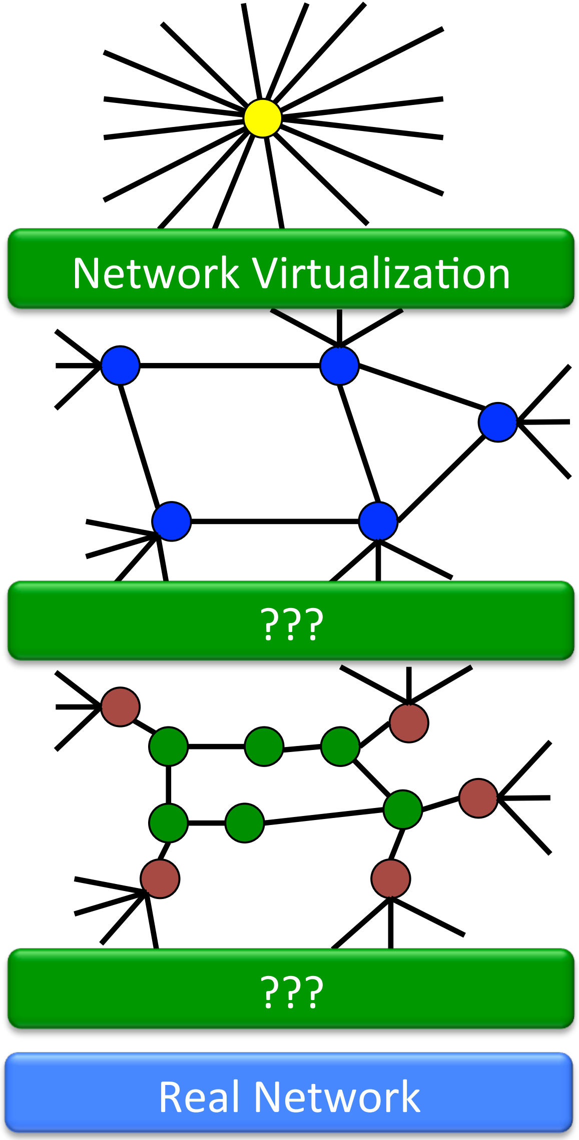
???

???

Real Network

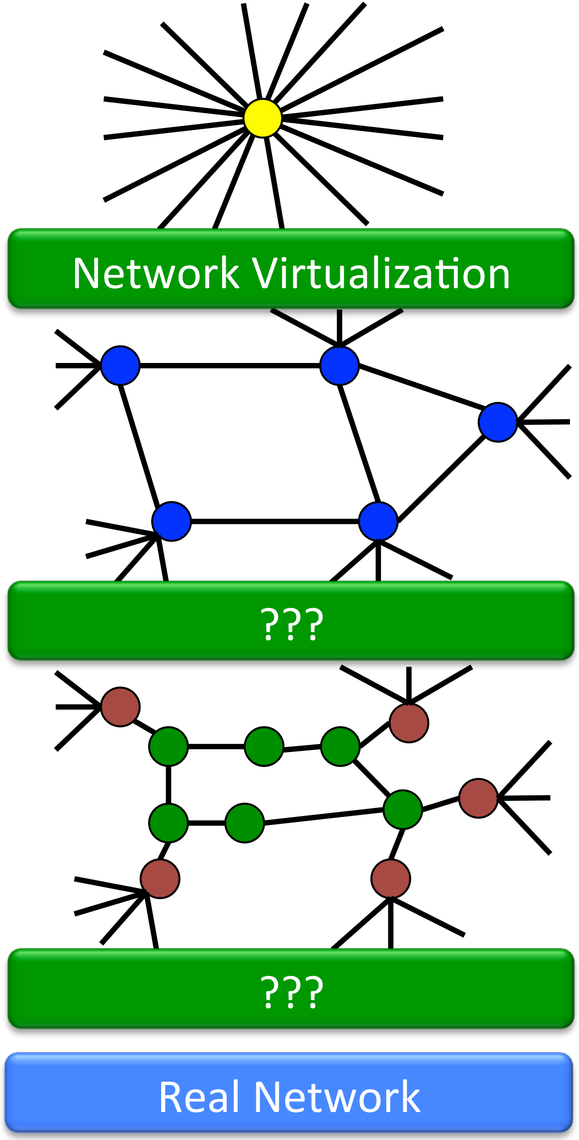
More on Abstractions

- Top level is virtual network view
- Lowest layer needs enough detail that it can be translated into switch settings



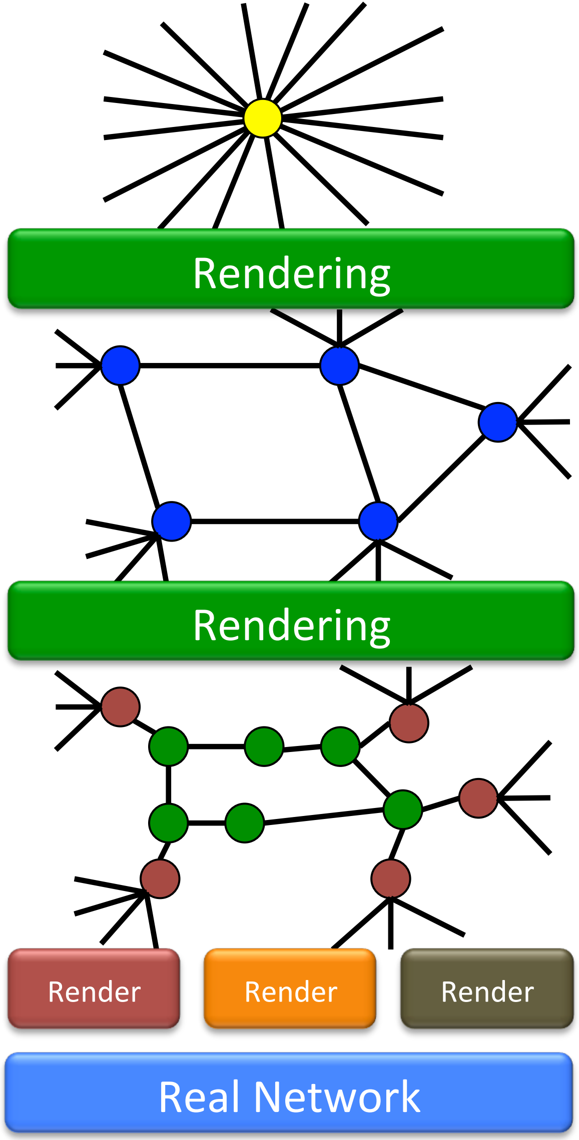
More on Abstractions

- Top level is virtual network view
- In-between are other layers of abstractions needed to provide programming APIs for network developers
- Lowest layer needs enough detail that it can be translated into switch settings



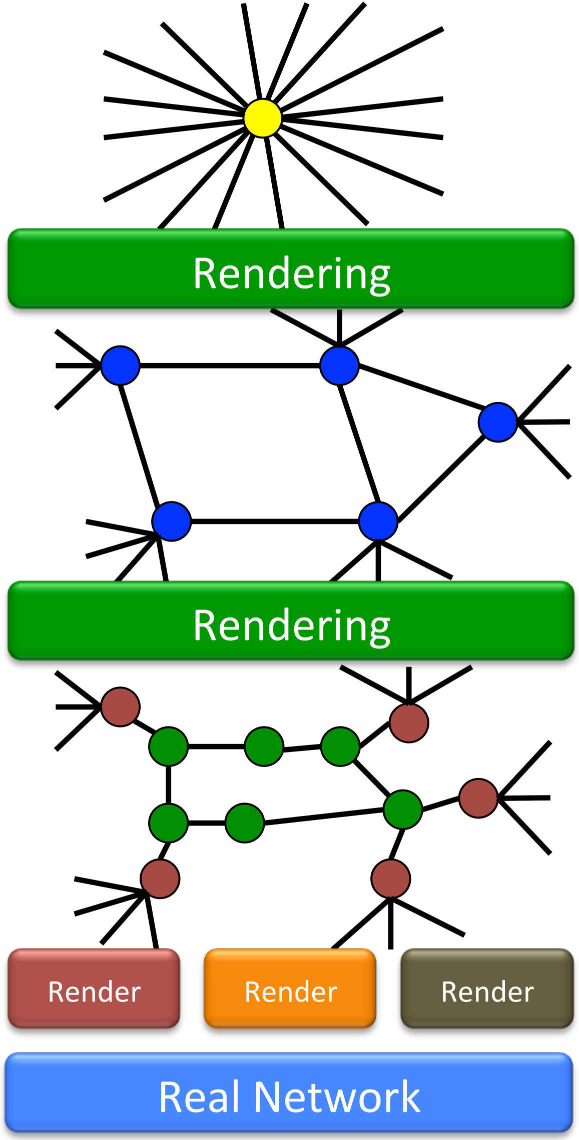
More on Abstractions

- Need to be able to *render* between different abstractions

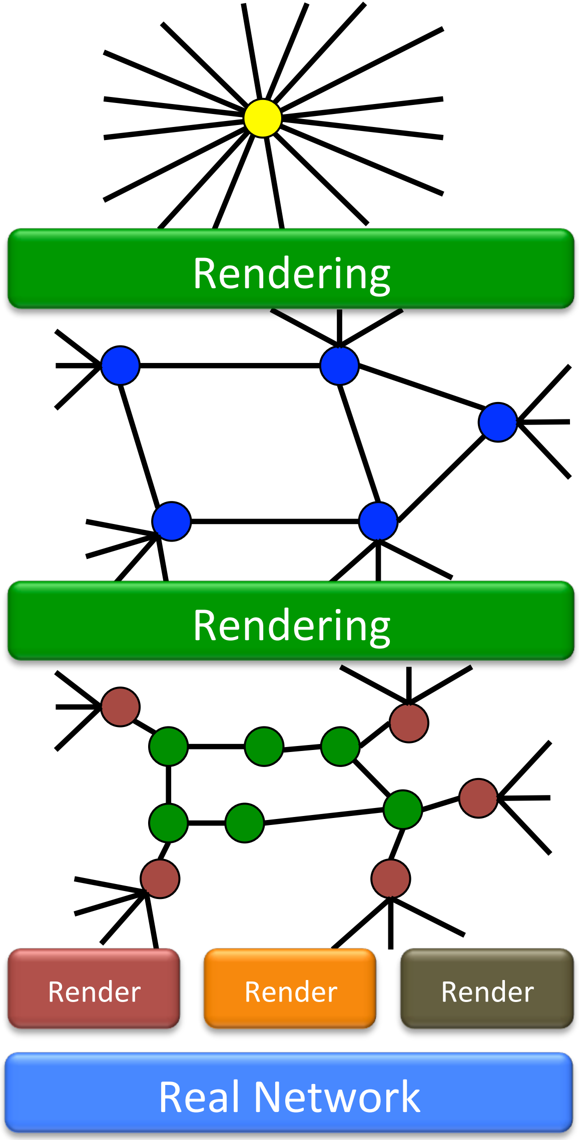


More on Abstractions

- Rendering is not translation
- Rendering is the set of algorithms that we write to:
  - Handle failures
  - Optimize energy use
  - Determine when to light up new lambdas
  - Decide when to adjust L2 topology to improve L3 performance
  - Etc.



More on Abstractions



- Need different modules to render between lowest layer abstractions and network element parameters
  - Want to encapsulate technology- and protocol-specific concerns

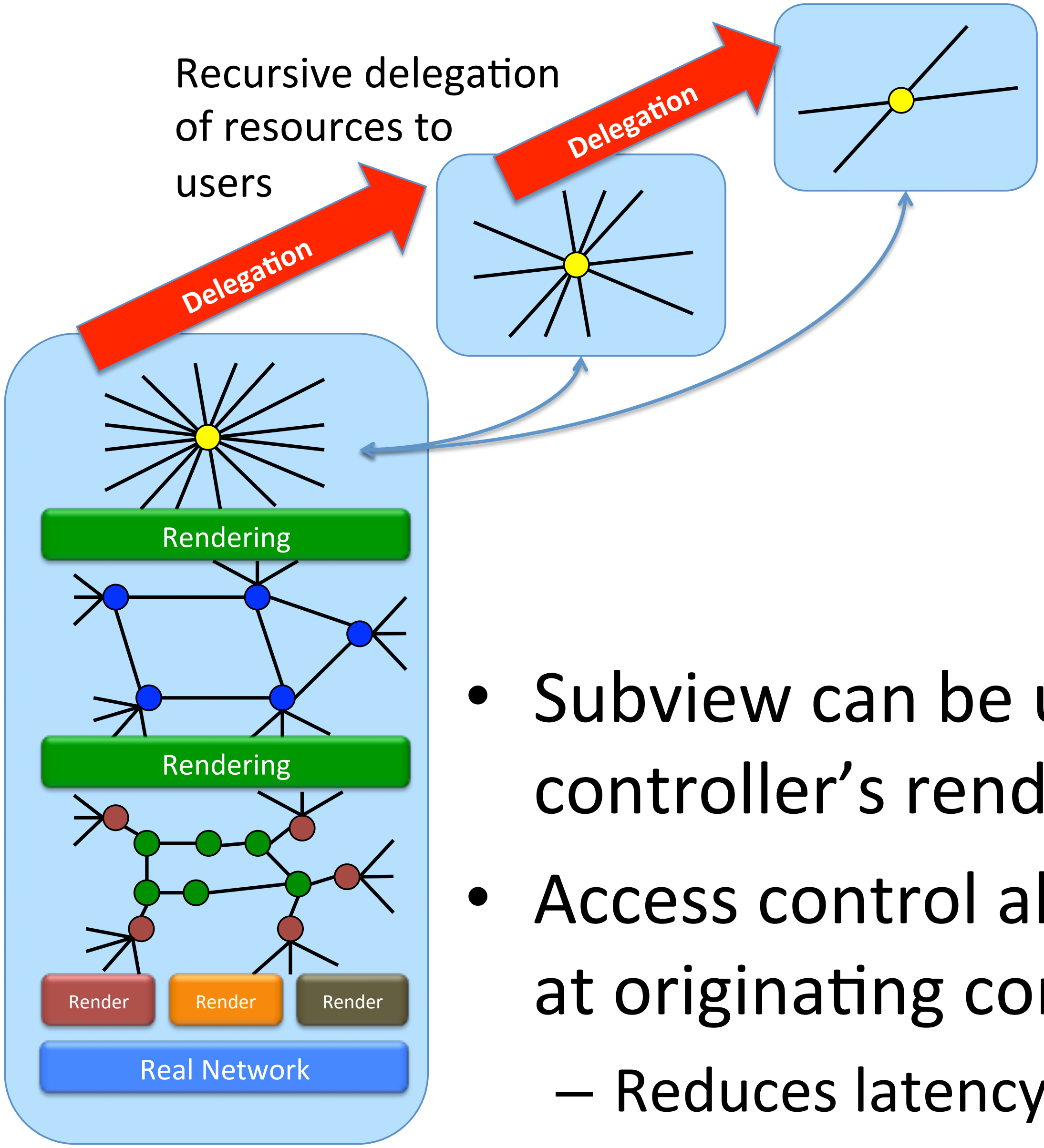
Non-Global Network Views

- When is a global view *not* OK?
  - When you can't/shouldn't share all the details

- Multiple users
- Multiple domains
- Scalability (multiple controllers)

- What's in common in these situations?
  - Need to share a "sub view" of abstraction(s)

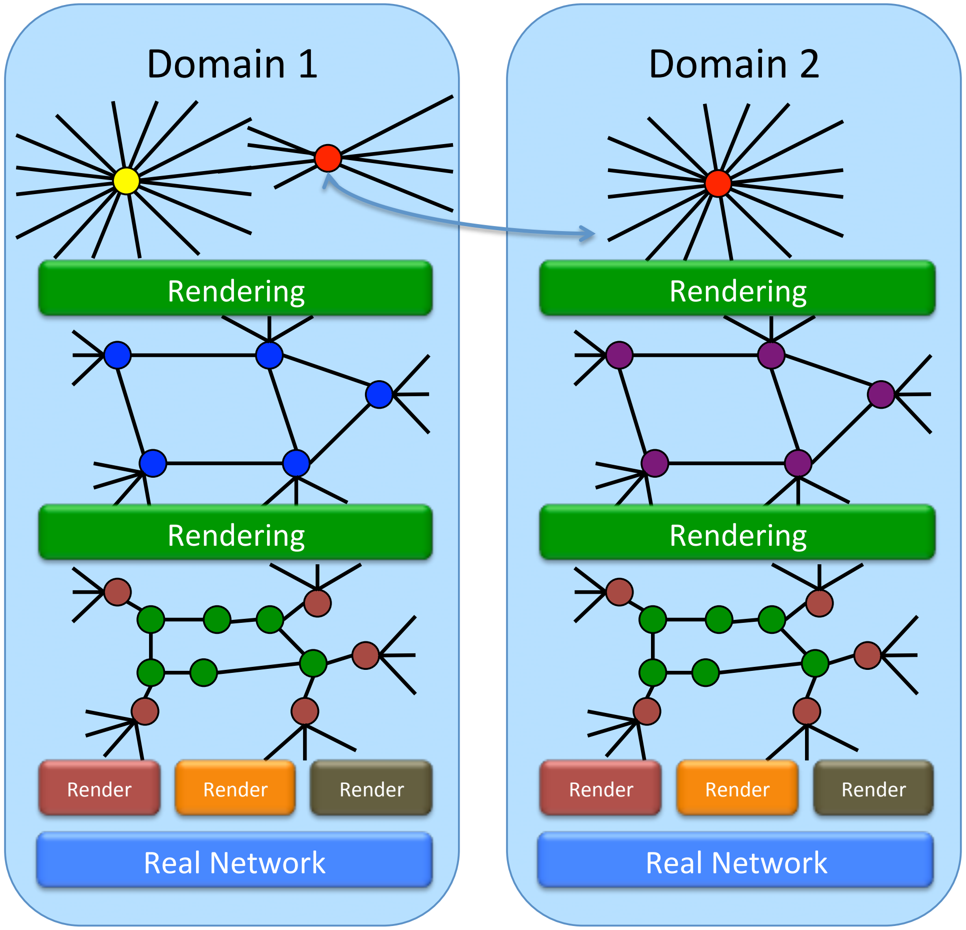
Multiple Users


Recursive delegation of resources to users

- Each user has a subview
  - Only able to affect its delegated resources, see its statistics, etc.
  - Empowered to develop own control programs (services)
- Subview can be updated by original controller's rendering engine
- Access control also can be enforced at originating controller
  - Reduces latency

Multiple Domains

- Subview restricted to resources its negotiated the right to use
- Could be static
  - Lambda/MPLS tunnel from X to Y
  - No hook to rendering required
- Could be dynamic
  - Request tunnels, OF slices in switches
  - Hook to rendering required



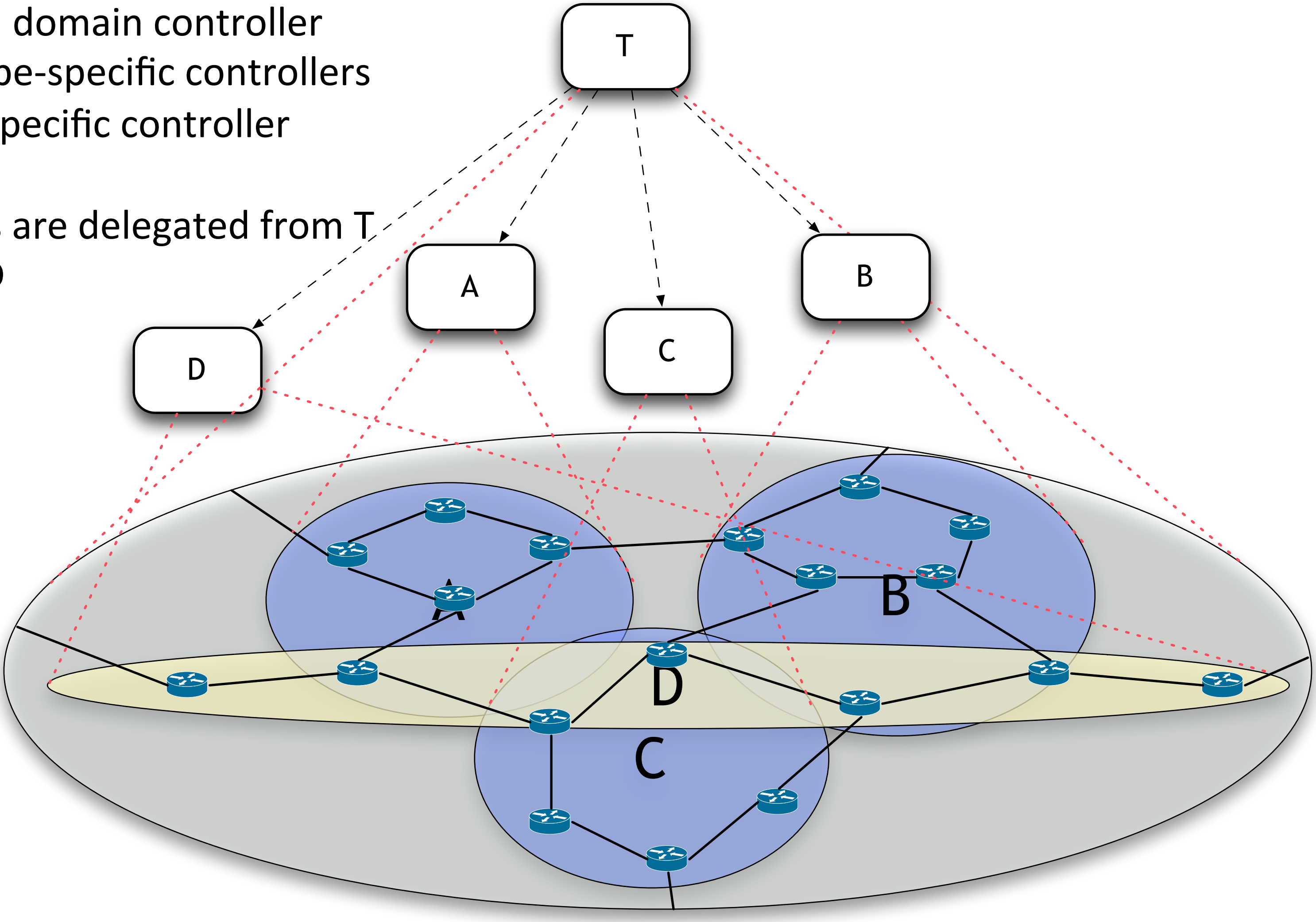
renci

Network under distributed SDN control

Raytheon
BBN Technologies

T: Top-level domain controller
A, B, C: scope-specific controllers
D: service-specific controller

- Resources are delegated from T to A, B, C, D



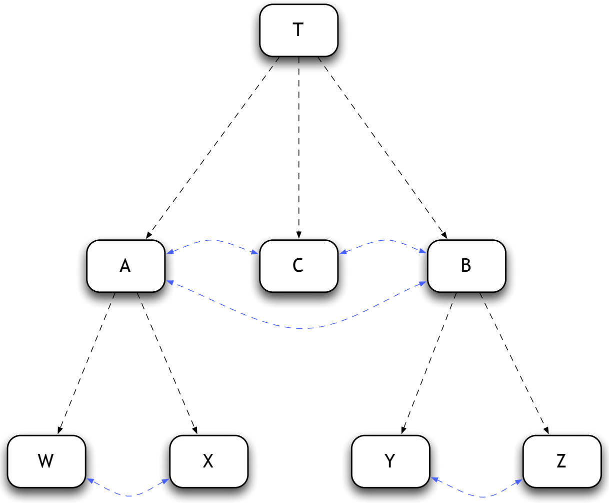
Resource delegation between controllers

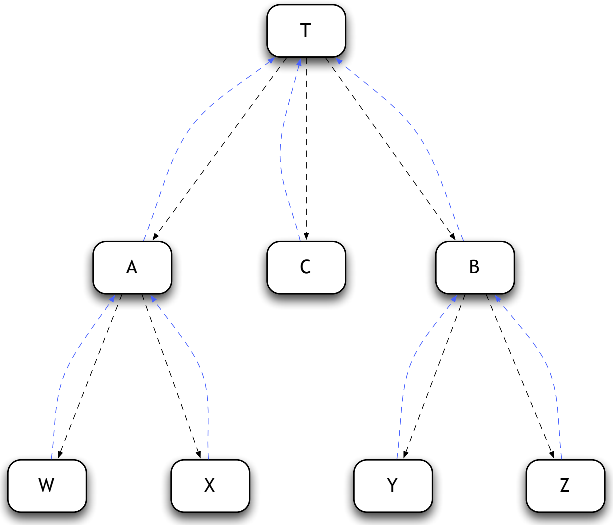
$$D_c^t =< INP_c^t, OUTP_c^t, R_c^t, T_c^t, Tr_c^t, B_c^t, W_c^t, G_c^t, F_c^t >$$

- Input ports
- Output ports
- Expected receive labels
- Authorized transmit labels
- Ability to translate labels
- Flow rule space, buffer space and outgoing bandwidth
- Topology information
- Meta information
  - Term and authorization attributes

Control resiliency with delegations

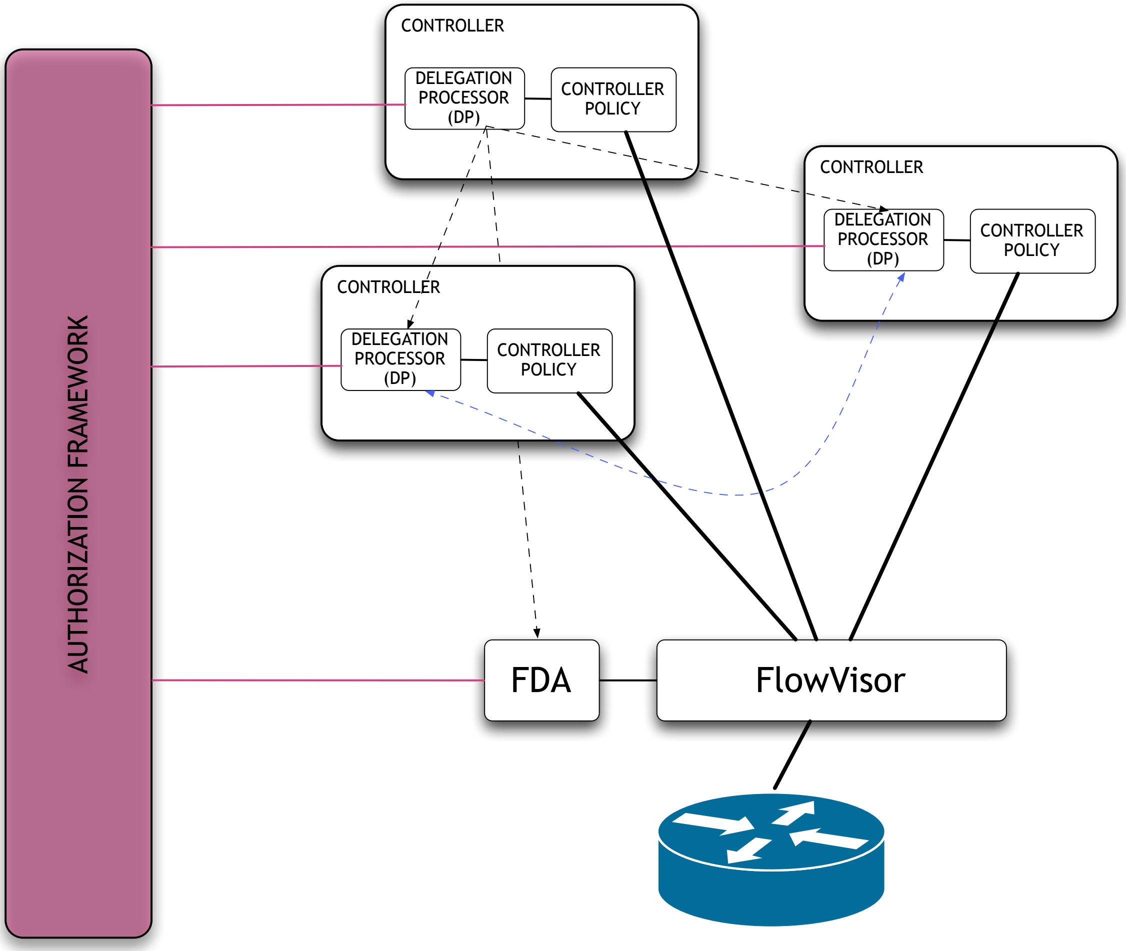
- Peer-to-peer
  - Peer controller fails, its delegation is taken over by a peer controller
  - By prior agreement
  - By consensus
- Hierarchical
  - When a subordinate controller fails, its parent controller takes over the scope



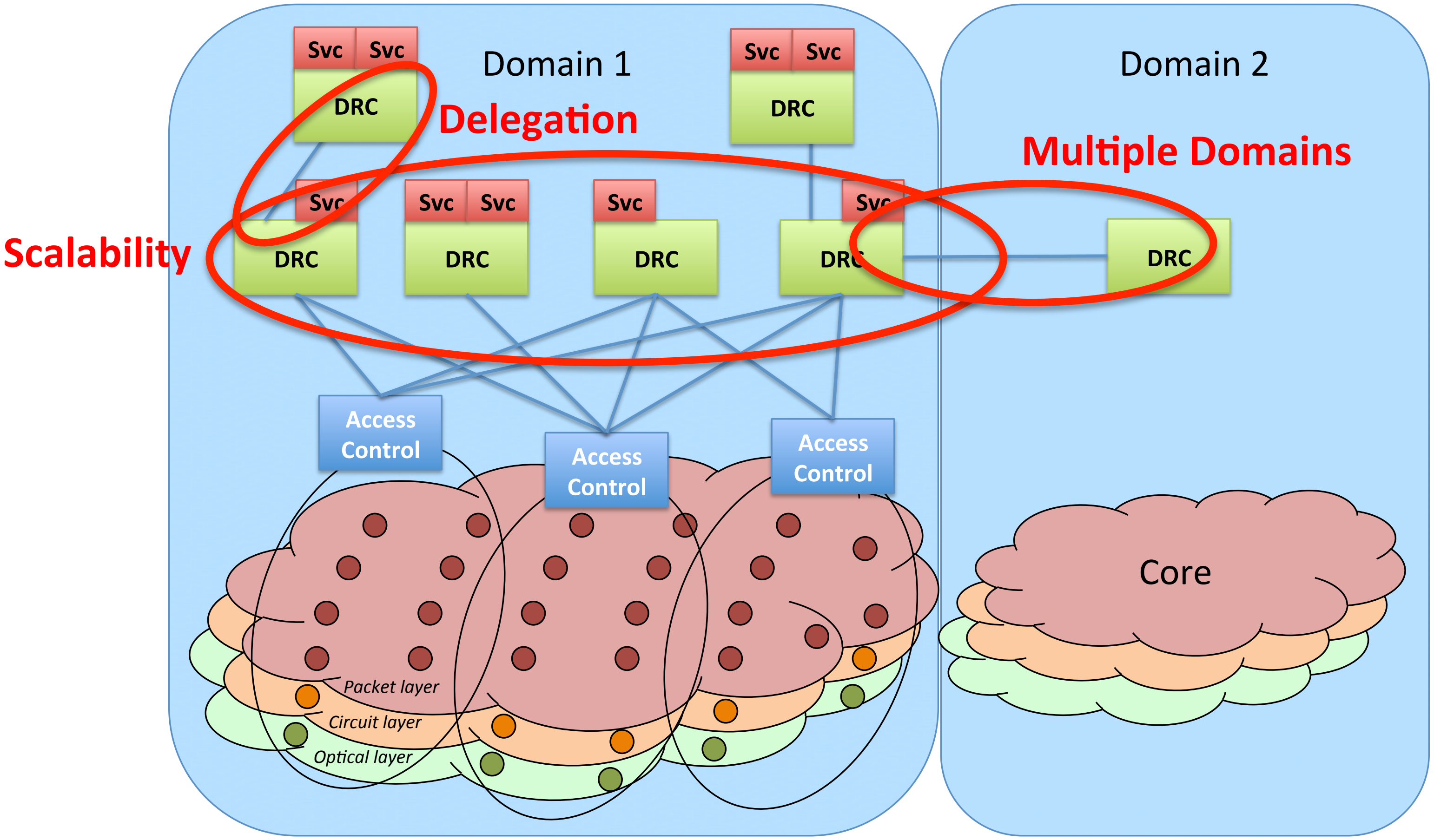


Control detail

- Multiple controllers can introduce flow policies into a switch
- Mitigated by a combination of FlowVisor-like element and delegation agent
- Authorization framework modulates access and delegation flows



Summary



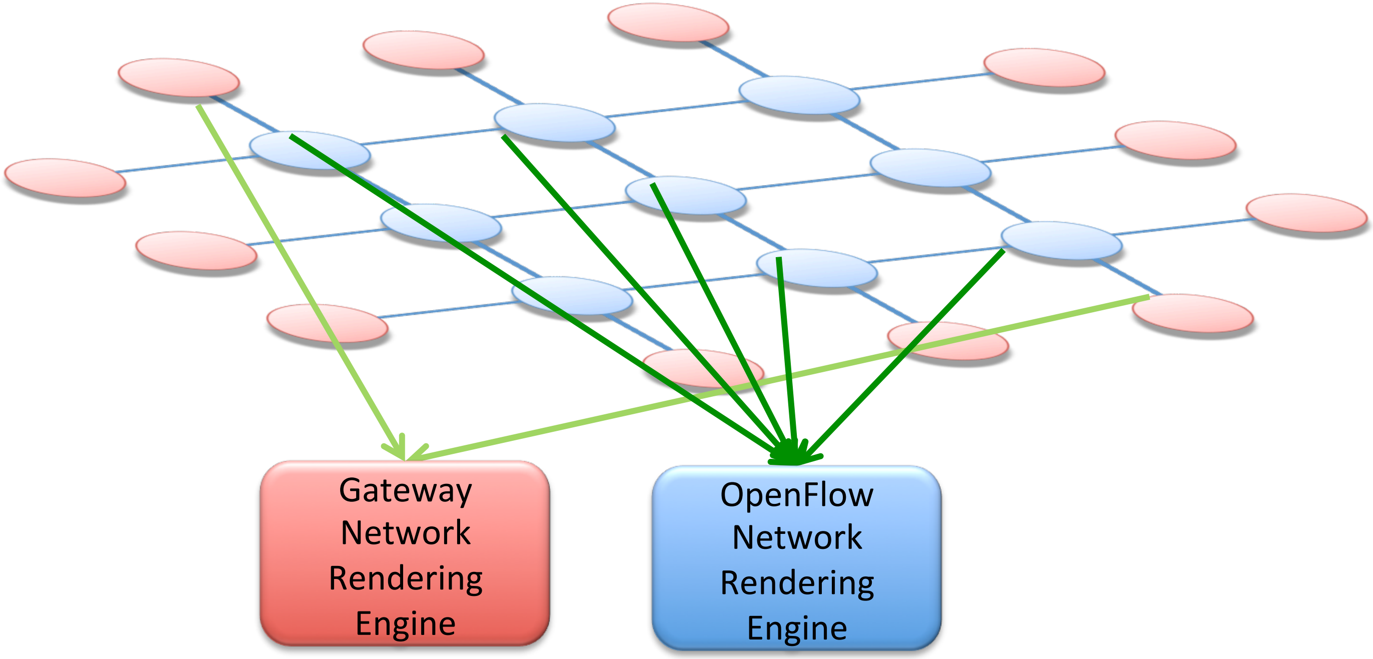
Abstractions, delegation and rendering are unifying concepts

Implementation Details

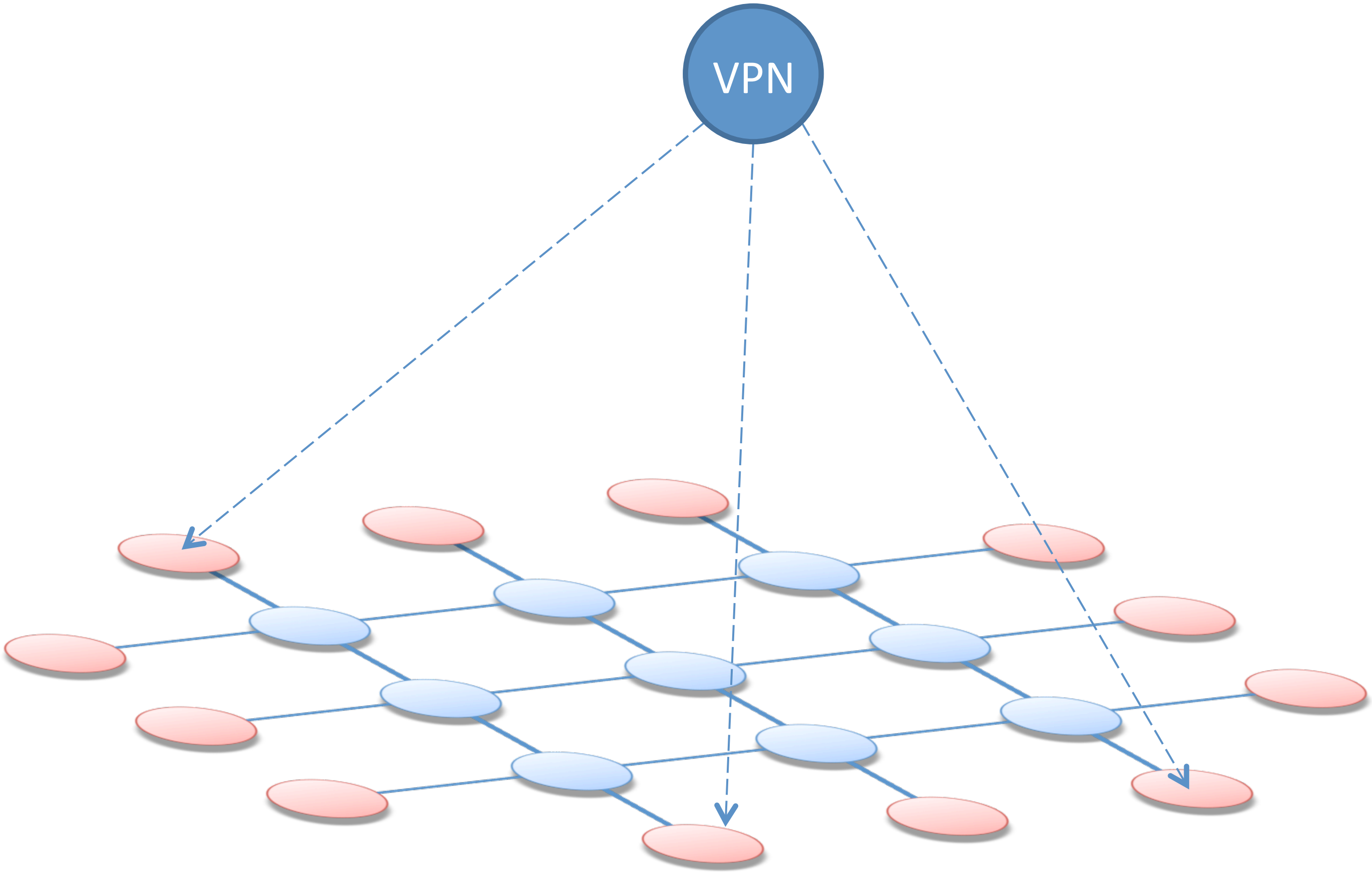
- Built around Graph Database
  - Natural representation of abstractions
  - Leverage high-scalability, network tools
  - Rich mechanisms for sharing subsets of information
- Triggers
  - Applications and rendering engines express their *interest* in an graph database action and are notified when it happens
    - Create a vertex of type V or an edge of type E
    - Attach an edge of type E to a vertex of type V, etc.
  - Changes in graph database impact apps/network and vice-versa
- Time/events as first-class citizens
  - Necessary for reservations and smooth transitions between configurations
  - Allows coalescing of triggers
- Implementation was a collaboration between BBN and NEC using ProtoGENI resources

VPN Example

Set up topology of nodes and links
Attach network rendering engines

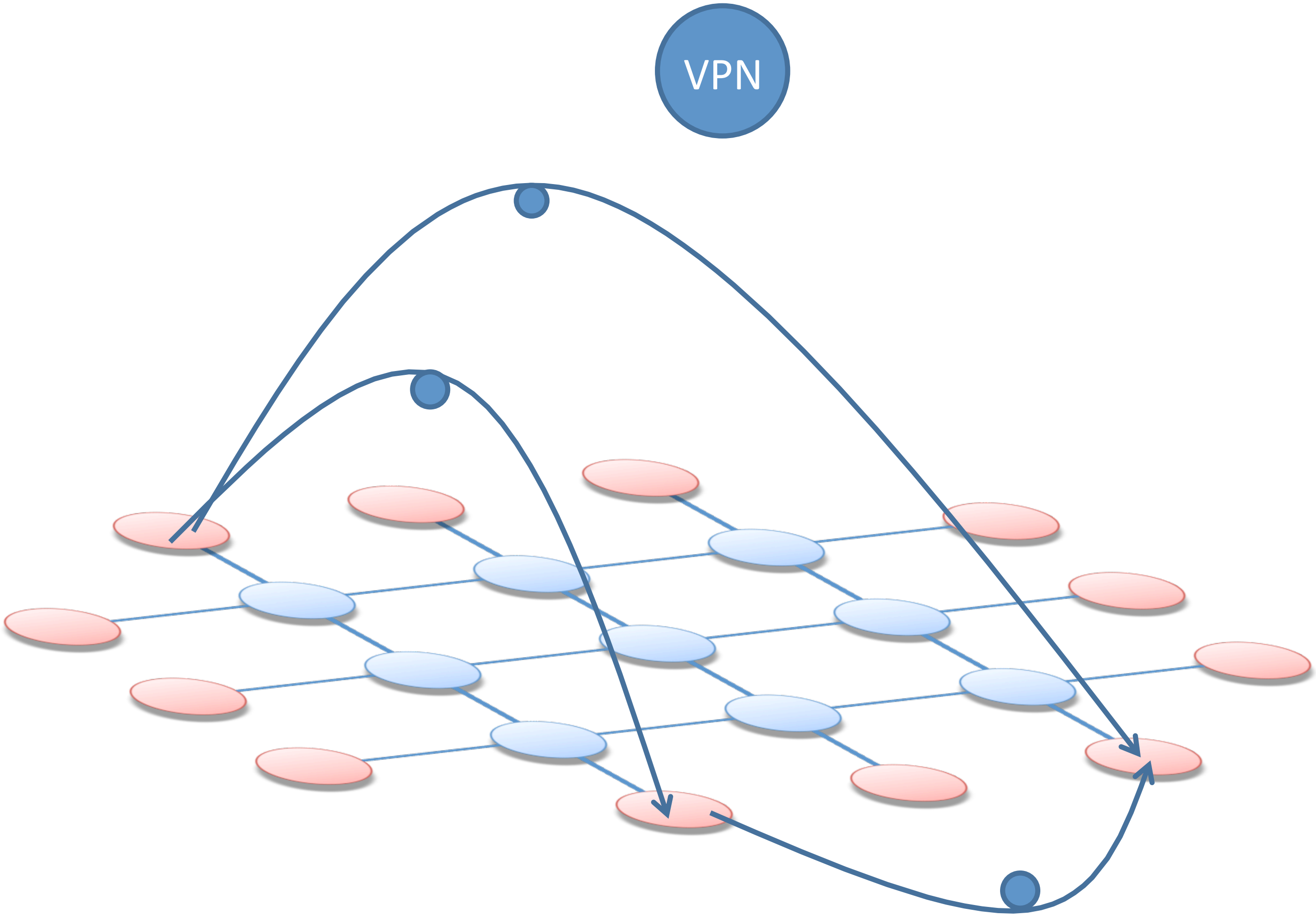

Gateway Network Rendering Engine
OpenFlow Network Rendering Engine

VPN Example



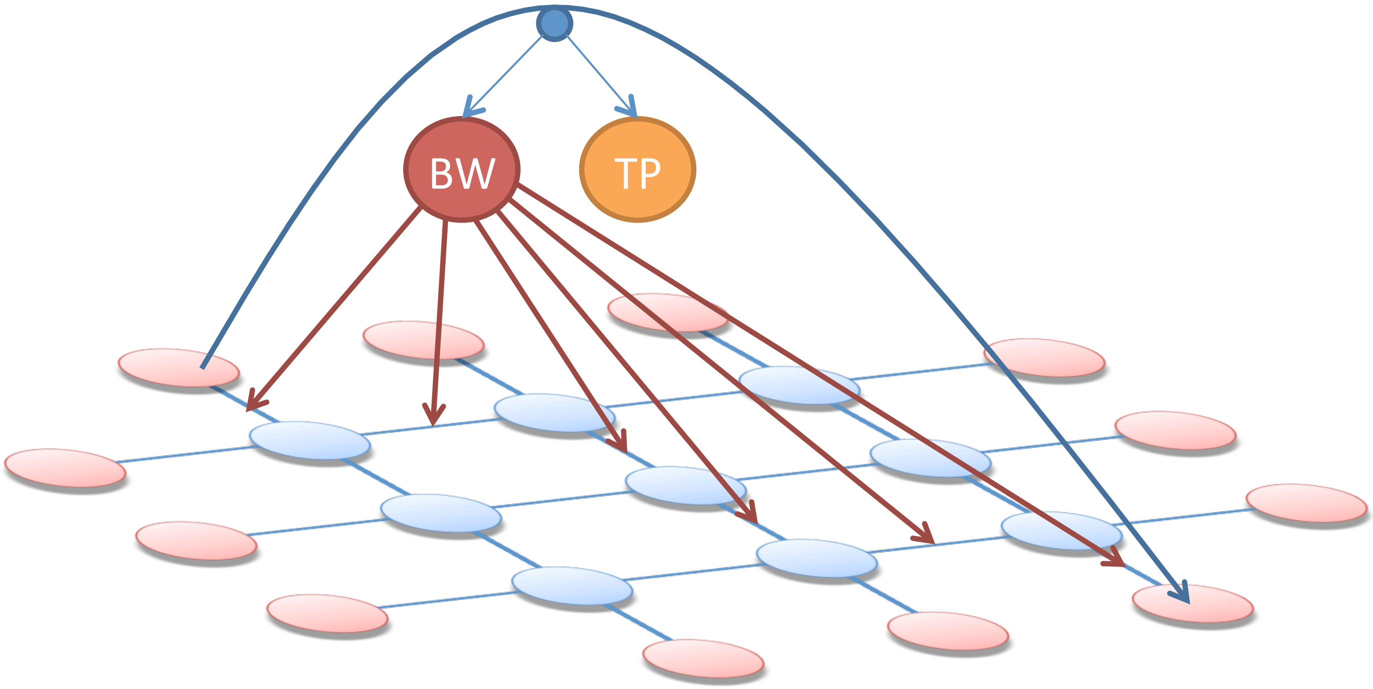
Specify the IP address ranges at sites to be connected

Site to Site Edges



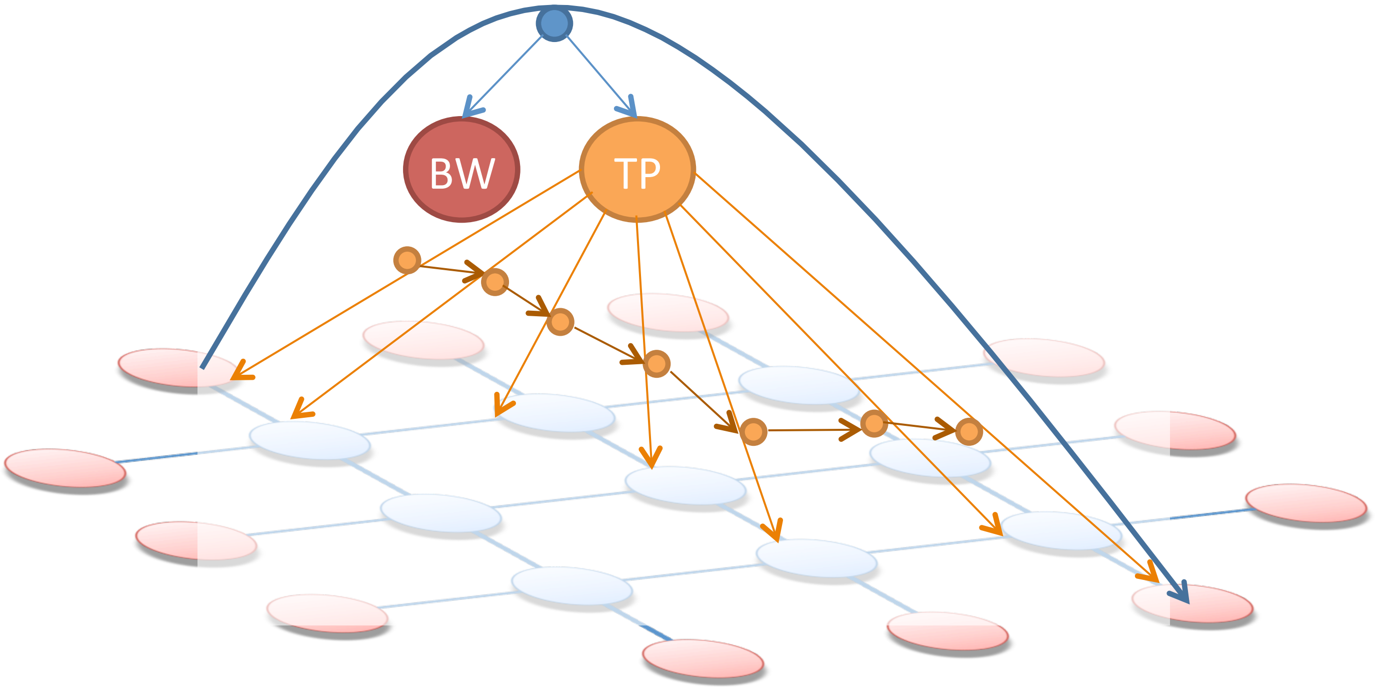
Specify the required site-to-site bandwidth

Site to Site Path



Bandwidth tracks remaining capacity on all links
including future commitments

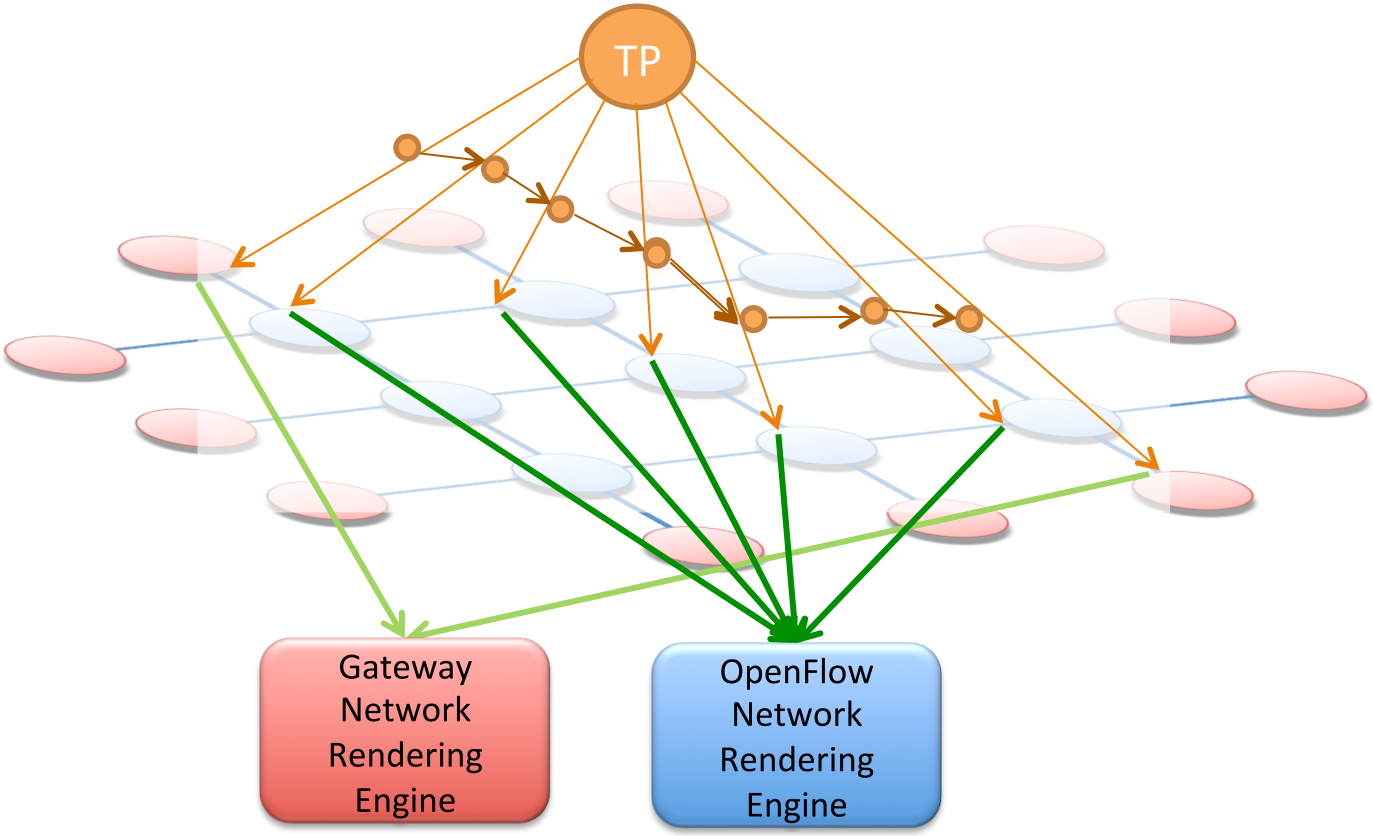
Traffic Classes & Forwarding Policies



Traffic Class tracks links used to setup path
Forwarding Policy determines use of Ethernet, IP or MPLS forwarding

Traffic Classes & Forwarding Policies

Network Rendering Engine triggers notify them when traffic policies change
Application triggers notify them when NREs modify graph DB to reflect topo changes


TP
Gateway Network Rendering Engine
OpenFlow Network Rendering Engine

Coupling IP network to SDN/ OpenFlow:

- Beyond IP gateways:
  - In a gateway a packet goes from L2 to L3 and back on a new link
  - SDN/OF allows simplifying this function
  - OF (v1.3) offers a 14-tuple packet label space that can be interpreted/rewritten as desired
    - Interpretation of labels doesn't have to be fixed
- ARP
  - An IP stack requires that L3 address is resolved to a L2 address
  - Controllers like Floodlight do this for a single domain
  - What happens in a multi-domain environment?

Bringing it together

- Apply the idea of controller resource delegations to a multi-domain transit environment built on SDN/OF
- Instead of the gateway, ingress controller rewrites part of the L2 header as path id
  - To keep things simple keep to one header
  - Keeps frame size constant, simple operation
- Intermediate controllers forward based on path id (with exceptions for nodes that moved)
- Explicit delegation of flow table space in switches to specific controllers/services helps isolation
- Egress controller can rewrite the L2 header as needed so destination network stack accepts it

Next Steps

- ARP
  - Implemented ARP in controller as a basic mechanism
- Delegation
  - Implement delegation processing
- Multiple domains
  - Prototype based on delegation of transit service by combining/ extending VPN
- More nuanced time/event handling
  - Support for lossless transition between configurations

- Eventually
  - Multiple layers of network resources (e.g., Optical)
  - Richer network abstraction
  - Add in non-network resources (compute, storage)