SDN Peering with XSP

Ezra Kissel

Indiana University

Internet2 Joint Techs / TIP2013

January 2013

Overview

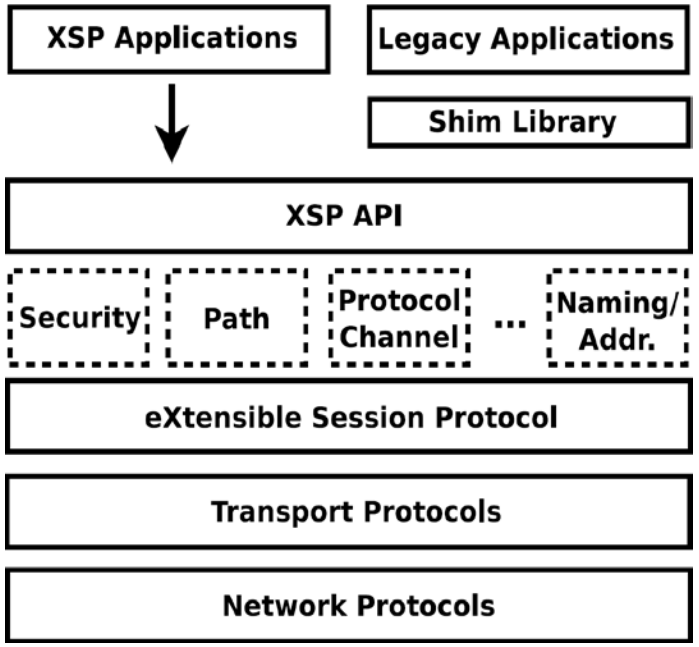
- Software Defined Networking and OpenFlow
  - Fine-grained control of forwarding in the data plane
  - Tremendous interest in R&E as well as industry.
- What exactly is SDN peering?
- A few key characteristics:
  - Advertise and update topology of OpenFlow-based networks (most likely dynamic)
  - Provide a trust model (authentication and authorization)
  - Bridge existing inter-domain technologies
- XSP and SDN: session-based protocol framework for interaction between applications and network services

eXtensible Session Protocol (XSP)

- In the spirit of the ITU-T Recommendation X.225 connection-oriented session protocol specification:
  - *" … a single protocol for the transfer of data and control information from one session entity to a peer session entity..."*
- XSP resides above the transport layer
  - Can encapsulate control and data PDUs into session layer PDUs
- Provides a common set of features used by applications
  - Rather than being defined specifically for each application
- We can also think of a session in the most literal sense:
  - "*a period of time devoted to a particular activity*"

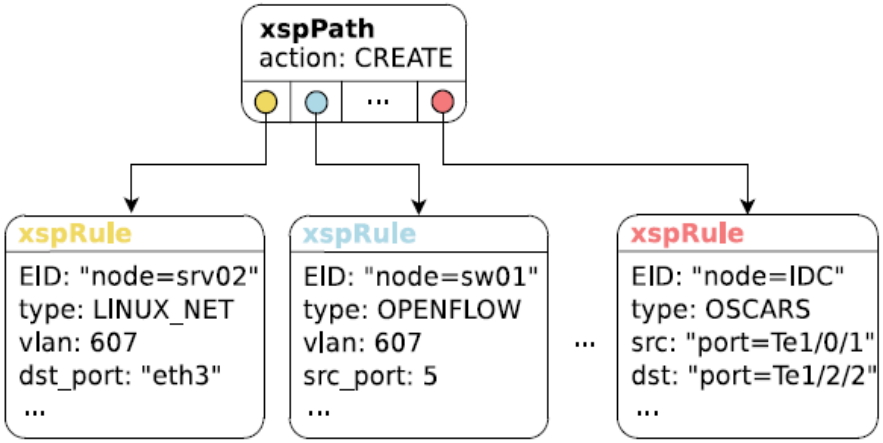
What is in a session layer?

- End-to-end state
  - Session state for parallel or serial transport connections
  - Context for mobility, multipath
- Authorization and authentication
  - SSL/TLS, X.509, and SSH
- Naming and addressing
  - Session **`connect()`** not tied to l3/l4 addresses
- Explicit signaling and data channel optimizations
  - Phoebus WAN acceleration
- Flexibility for new approaches and technologies
  - SDN
  - Delay tolerant flows for performance (buffer and burst)



XSP as a network configuration interface

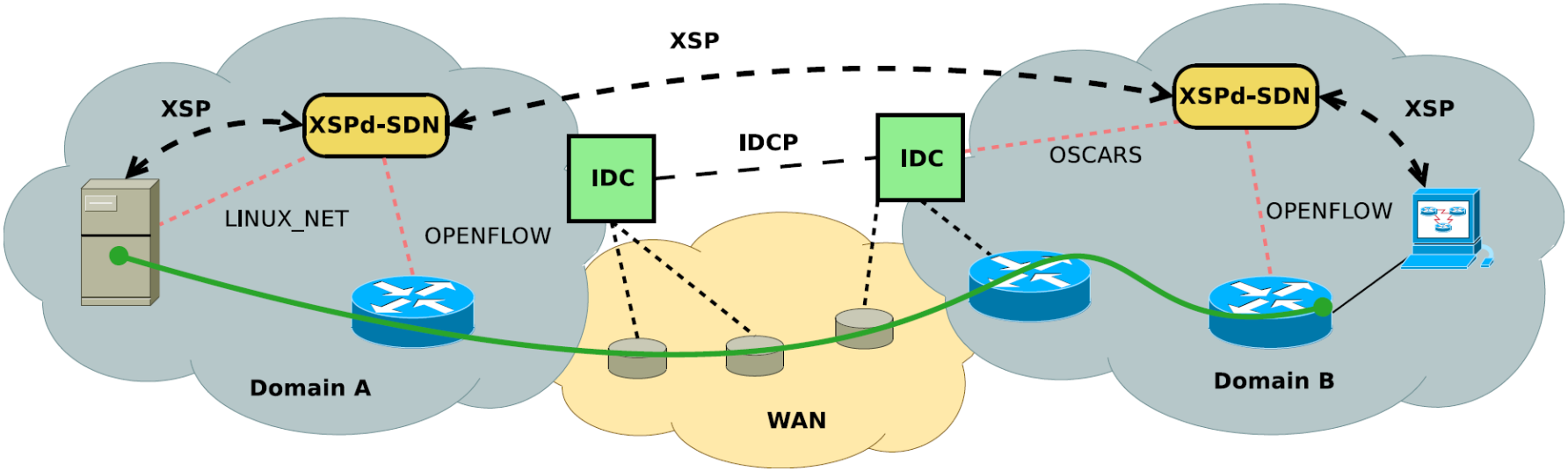
- *Path* framework
  - Modular service handlers for different technologies



- A path is a set of applied rules in the network
- API provides a consistent abstraction
  - Access to technology-specific fields if needed
- Service called XSPd-SDN for SDN deployments

XSP for application-driven networking

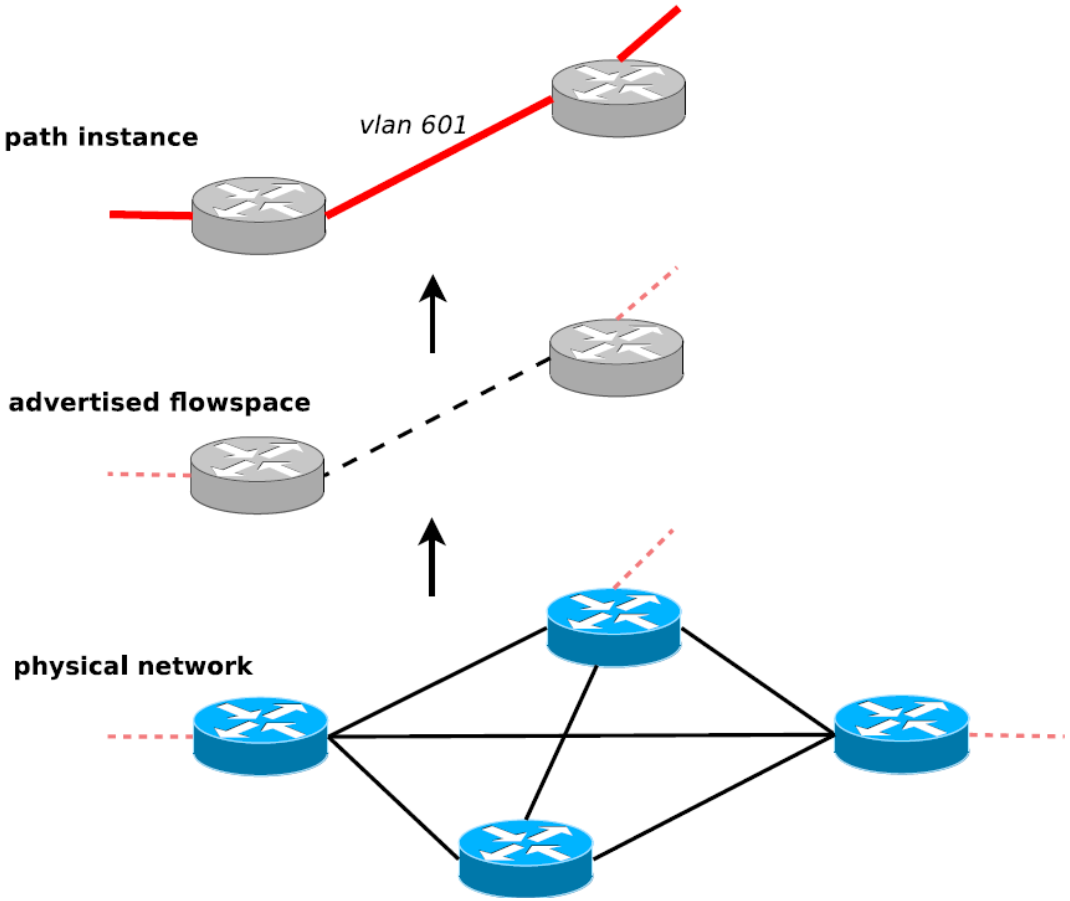
- Common interface for explicit path provisioning
  - Interaction with OSCARS, OpenFlow, (e.g. Floodlight interface), Linux networking
  - Planned and/or in development: OESS and NSI
- Build end-to-end paths using XSP client API (wrapper, *libxsp*)
- Prototype deployment for DYNES
  - GridFTP support via *xio-xsp* driver



A case for SDN peering

- Exchange dynamic campus and data center network topologies in a secure manner
- XSP session provides for a period of authorized use
  - Secure context for exchanging credentials and updating peers
- SDN also presents additional challenges
  - More complexity
  - Level of control offered
  - Is mutual economic incentive enough (i.e. BGP model)?
- Allow for expressive policy

Representing SDN flowspace

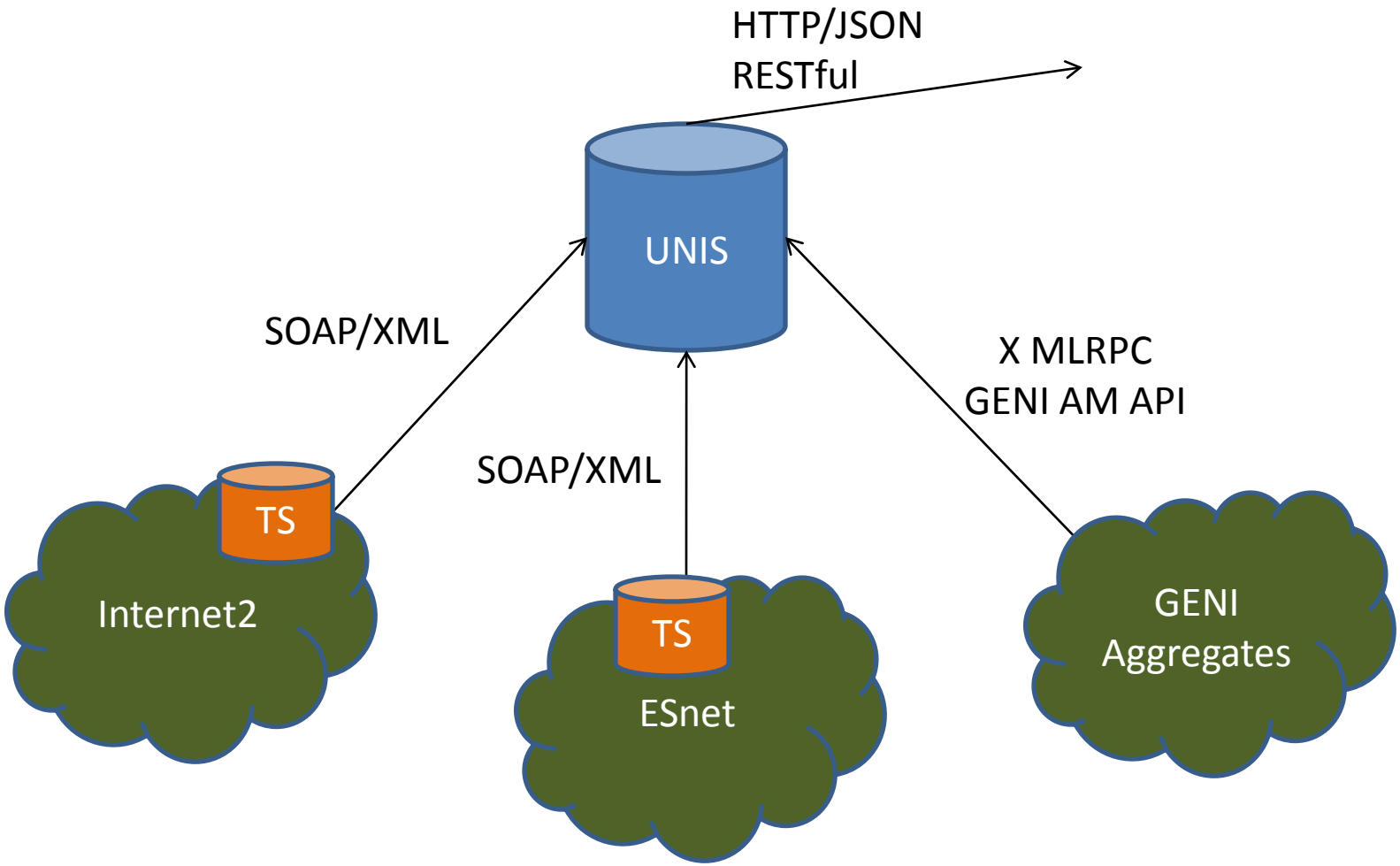


- Flowspace advertised as a virtual network
- State can change as devices are discovered or removed
- Use of the flowspace network may also be updated between peers
- Track associated controllers

Unified Network Information Service

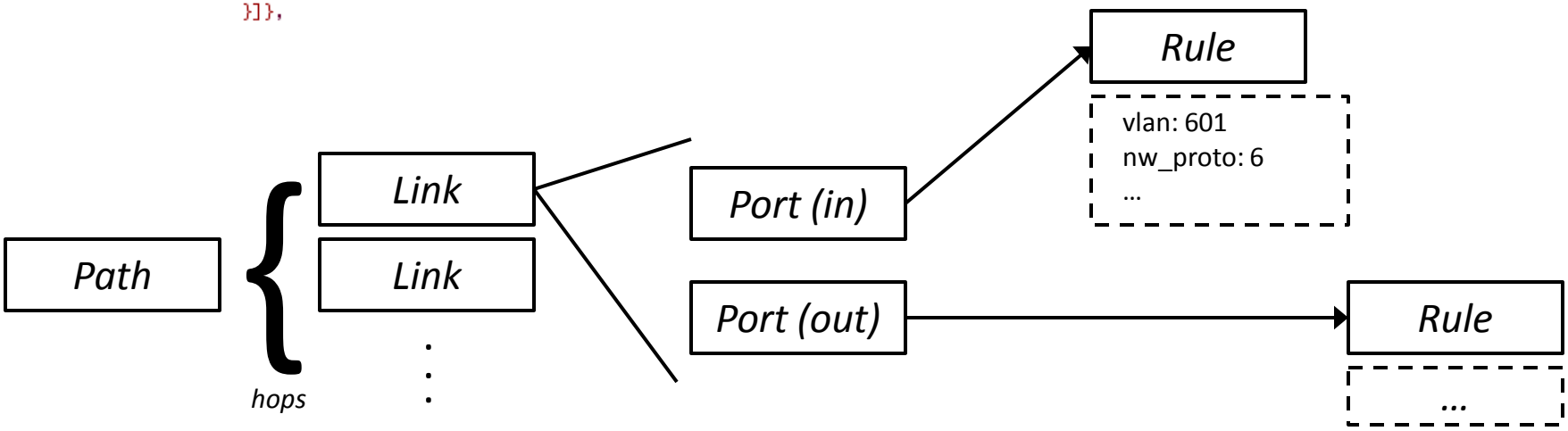
- Topology and service information store
- RESTful re-implementation of the perfSONAR protocols
  - Re-implementation vs. reworking based on experience and current thinking
  - JSON schema, BSON on the wire and storage for efficiency
- Built-in AuthN/AuthZ using PKI
  - Attribute Based Access Control (ABAC)
  - UNIS.rReadOnly ← UNIS.rTrusted.rReadOnly (delegation)
- Clients may also subscribe to network resources

UNIS: towards a common network model



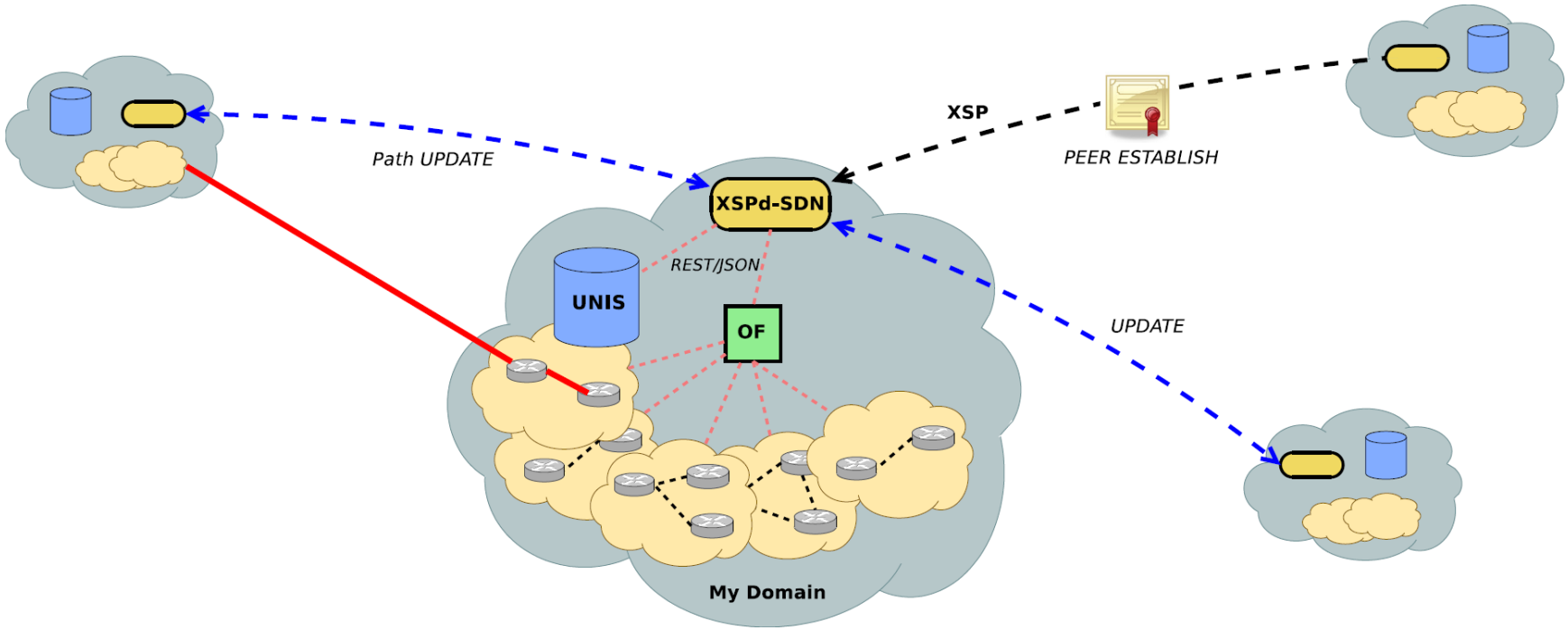
UNIS example


```
"networks": [
    {
        "$schema": "http://unis.incntre.iu.edu/schema/20120709/network#",
        "urn": "urn:publicid:IDN+mydomain+network+flowspace_XX",
        "id": "mydomain.net_network_flowspace_XX",
        "name": "flowspace_XX",
        "address": {
            "type": "openflow",
            "value": "*:*:*:*:0x800:600:*:*:*:*:*:*:*:*"
        },
        "properties": {"openflow": {
            "controller": "tcp://mycontroller.mydomain.net:6633",
            "vlan-range": "600-620"
        }},
        "nodes": [
            {
                "name": "edge-sw-flowspace_XX",
                "urn": "urn:publicid:IDN+mydomain.net+network+flowspace_XX+node+edge-sw-flowspace_XX",
                "relations": {"over": [{
                    "href": "https://unis.mydomain.com/nodes/edge-sw.mydomain.net",
                    "rel": "full"
                }]},
```

Multi-domain SDN peering with XSP and UNIS

- Sessions for negotiating authorized exchange of topology
  - Peering policy determines advertisement scope
  - Should distinguish between public and protected topologies



What does this give us?

- Framework for defining peering arrangements
  - Extensible and flexible
  - Consistent with existing XSP *Path* features
- XSP abstraction enables both peer negotiation and exchange
  - Forwarding of advertisements
  - Valid over session lifetime
- However, we may still require (per-domain):
  - Policy engine, path computation
  - Resource management, allocation
  - Additionally a network programming language abstraction

Future work

- Define peering policy in line with best practices
- Demonstrate real world use-cases
- OpenFlow islands bridged with WAN
  - Make use of existing WAN provisioning services
- Trans-continental paths
  - TransPAC3, ACE, I2 100G

Summary

- XSP provides an interface for configuring advanced networks
  - SDNs including (but not limited to) those implemented with OpenFlow
  - “Network as a service”
- XSP sessions provide a natural context for peering networks
  - Secure connections, negotiation of credentials
  - State over a period of time
- UNIS gives us a consistent representation of the network
  - Campus, data center, WAN
  - Extensible schema with pub/sub and AuthN/AuthZ
- Works with existing technologies