YANG-Based Service Models for Services over MPLS Networks

Adrian Farrel
Juniper Networks
Email: adrian@olddog.co.uk

www.isocore.com/2015
Agenda

- What is a Service Model?
- Why should we standardize Service Models?
- The Layer Three VPN Service Model (L3SM)
- Service Models in the SDN architecture
- What other Service Models could we develop?
- How will we measure success?
The Interface to the Operator

• A Service Model is part of the interface between a customer and the operator
  • That makes it one element of a business interface
  • Other aspects of the business interface are not in scope
    • Pricing
    • Billing
    • SLA
• Try to describe the services in a way that is common to multiple operators
  • Gives the customer a common point of reference
What is a Service?

• A collection of network functions provided by an operator to their customer

• Connectivity services
  • Internet connectivity
  • Virtual private wire
  • VPN

• Basic units of purchasable function
  • Available from multiple operators
  • Core characteristics the same
  • May be described and sold in different ways to maintain market differential
Benefits of a Common Approach

• A Service Model is a description of a service
  • A data model that can be represented in code
• Each operator could use their own data model
  • Would find a large overlap between models
• Try to standardise the common portions
  • Each operator uses the common model
    • Adds extensions for their own representation in the market
• Standard service model provides
  • Common base for customers
  • Opportunity for automation of service delivery
Using YANG for Service Models

- YANG is the data modelling language du jour
- There have been many modelling languages and there will probably be many more
- YANG is convenient for human and machine
- Not particularly good on the wire (it’s XML)
  - Easily mapped to other encodings such as JSON
- The main benefit is that it is widely understood
Modularity and Extensibility

- Two important features of data model design
- Modular
  - Possible to pull out components of the model
    - Leave them out completely
    - Re-use them in other models
- Extensible
  - Possible to extend (augment) the model
  - Allows new features to be added
  - Lets operator build on standard model
    - Add their own features
  - Achieve market differential
Layer Three VPN Service Model as an Example

• First attempt at a Service Model in the IETF
  • Unsure that a common description can be agreed
  • Pick a “simple” and “popular” service – L3VPN
• Built a team of network operators (Orange, BT, Verizon, AT&T) and let them get on with it
• Constrained discussion to PE-based L3VPN
• Basic blocks
  • Service identification (service name, service id, customer name)
  • VPN sites (many parameter!)
  • VPN topology (any-to-any, hub-spoke, hub-spoke-disjoint…)
  • Service provided (cloud, multicast…)
• Somewhat to our surprise, these operators have been able to agree
What Can I See From Where I’m Standing?

- Not a lot!
  - Customers can’t see under the hood of the network
  - Service models are not configuration models
Automation of Service Delivery

• Delivery of services can be a major hassle for operator
• Now a service can be expressed in code
  • Perhaps we can automate service delivery
• This brings us into the world of SDN
• Service orchestration
  • Take Service Model as input
  • Output network and device configuration models
• See this in *many* SDN architectures…
Service Models in RFC 4176

- Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management

| Service         | +----------------+ :          |
| Management      | | Service |<-----------------------:| Customer |
| Layer           | | Manager : | Agent |
+----------------+ :          |
| Network         | | +----------------+ :          |
| Management      | | | Provider :         |
| Layer           | | | Network | Customer |
|                | |<-------| Manager | Interface |
|                | +----------------+ :          |
| Network Element | | :          |
| Management      | | +-------+ :          |
| Layer           | | | CE |          |
|                | | PE | | device|          |
|                | | | of |          |
|                | |<--:--|VPN A|          |
|                | +-------+ :          |

---------------------------------------------->:<---------------------------------
Service Models in ABNO

• RFC 7491
  • A PCE-Based Architecture for Application-Based Network Operations
Service Models in the MEF

REFERENCE ARCHITECTURE

Customer Domain
- CANTATA (CUS:BUS)
- ALLEGRO (CUS:SOF)
  - Customer Application Coordinator

SP Domain
- Business Applications
  - LEGATO (BUS:SOF)
- Service Orchestration Functionality
  - PRESTO (SOF:ICM)
  - ADAGIO (ICM:ECM)
- Element Control and Management

Partner Domain
- Business Applications
  - SONATA (BUS:BUS)
- Service Orchestration Functionality
  - INTERLUDE (SOF:SOF)
  - PRESTO (SOF:ICM)
- Infrastructure Control and Management
  - ADAGIO (ICM:ECM)
- Element Control and Management

Network Infrastructure
How the L3SM Fits In

• draft-ietf-l3sm-l3vpn-service-model
  • YANG Data Model for L3VPN service delivery
Another View of the Architecture

- draft-wang-l3sm-service-automation-architecture
What Other Service Models Could We Work On?

• L2VPN is a popular candidate
  • But is it too complex to make common?
  • How about EVPN?
• Maybe some higher level commonality
  • A common VPN service model?
  • A data model for all services?
• Connectivity as a service
  • Maybe this is too simple?

• The big question: Why bother?
Other Related Work

• Following the principle of modularity
• IETF has work efforts on…
  • Policy
    • A key component of service description
    • Also relevant to configuration models
    • Is it possible to make a common description of policy?
    • A set of tools that could be used in other models
    • SUPA working group just formed
  • Security
    • Many different security functions in the network
    • These need to be configured and selected as services
    • I2NSF working group just formed
What Would Success Look Like?

• Can operators agree on a common subset of features?
  • Is this subset large enough to be useful?
• Is the resulting model extensible for operator use?
  • Can operators represent their different services?
• Can a Service Orchestrator be built to map to configuration models?
  • Might uncover some holes in the configuration models

• Prototypes have been built using early L3SM
  • Indicates that success is possible
Questions

adrian@olddog.co.uk