SENSE: SDN for End-to-end Networked Science at the Exascale

Internet2 Global Summit
May 8, 2018
San Diego, California
SENSE Team

- **ESnet**
  - Inder Monga
  - Chin Guok
  - John MacAuley
  - Alex Sim

- **Fermilab**
  - Phil Demar

- **ANL**
  - Linda Winkler

- **NERSC**
  - Damian Hazen

- **Caltech**
  - Harvey Newman
  - Justas Balcas
  - Maria Spiropulu
  - Raimondas Sirvinskas
  - Shashwitha Puttaswamy

- **University of Maryland/ Mid-Atlantic Crossroads**
  - Tom Lehman
  - Xi Yang
  - Alberto Jimenez
What Research Problem(s) are We Solving?

• Distributed scientific workflows need end-to-end automation so the focus can be on science, and not infrastructure
  – Manual provisioning and infrastructure debugging takes time
  – No service consistency across domains
  – No service visibility or automated troubleshooting across domains
  – Lack of realtime information from domains impedes development of intelligent services
What Research Problem(s) are We Solving?

• Science application workflows need to drive network service provisioning
  – Network programming APIs usually not intuitive and require detailed network knowledge, some not pre-known
  – Detailed network information needed, usually not easily available
  – Efficient use of resources

• Multi-domain orchestration requires service visibility and troubleshooting
  – Data APIs across domains for applications, users, network administrators
  – Performance, service statistics, topology, capability, etc.
  – Exchange of ‘scoped’ and authorized information
Vision and Objectives

A new paradigm for Application to Network Interactions

• Intent Based – Abstract requests and questions in the context of the application objectives.
• Interactive – what is possible? what is recommended? let’s negotiate.
• Real-time – resource availability, provisioning options, service status, troubleshooting.
• End-to-End – multi-domain networks, end sites, and the network stack inside the end systems.
• Full Service Lifecycle Interactions – continuous conversation between application and network for the service duration.
Application to SENSE Interactions

Failure on a network element, problem fixed

Service is not working, please check status
available 10Gbps is ok

Request 20 Gbps P2P service between Caltech and Fermilab. If 20 Gbps not available 10Gbps is ok.

20 Gbps available for a P2P service between Caltech and Fermilab

15 Gbps P2P service between Caltech and Fermilab Instantiated

Service is not working, please check status

Failure on a network element, problem fixed

What is the maximum bandwidth available for a P2P service between Caltech and Fermilab?

Please provide a listing of all available provisioning Endpoints

Endpoint Listing

Please provide a listing of all available provisioning Endpoints
Application to SENSE Interactions

- Transforms the network into a first class resource for workflow planning and optimization
- Allows applications to “query and negotiate” with network
- The SENSE infrastructure is designed to develop these types of services in DevOps manner, and customize for individual application agents.
- Cannot do every computation possible, but can do any computation desired.

Please provide a listing of all available provisioning Endpoints

Endpoint Listing

What is the maximum bandwidth available for a P2P service between Caltech and Fermilab?

20 Gbps available for a P2P service between Caltech and Fermilab

Request 20 Gbps P2P service between Caltech and Fermilab. If 20 Gbps not available 10Gbps is ok.

15 Gbps P2P service between Caltech and Fermilab Instantiated

Service is not working, please check status

Failure on a network element, problem fixed
SENSE

Application Workflow Agent

SENSE Orchestrator

Intent Based APIs with Resource Discovery, Negotiation, Service Lifecycle Monitoring/Troubleshooting

Datafication of cyberinfrastructure to enable intelligent services

Model Driven SDN Control with Orchestration

SENSE End-to-End Model

Realtime system based on Resource Manager developed infrastructure and service models

ESnet MAX

http://sense.es.net/

SENSE Resource Manager
SENSE Resource Managers

Model Based Control and Orchestration

- Learn system states
- Read/Sync data
- Network Markup Language/RDF based Model Schema

SENSE Orchestrator

- Control
- Feedback / Awareness
- Automatic Operation
- Infrastructure

- Change system states
- Write / Sync data
- Multi-Resource Modeling

Allow the machines to automate, iterate, react, and adjust to find solutions and not bring the humans in until absolutely necessary

SENSE Resource Manager

ESnet

MAX

NERSC

Argonne

Fermilab

UMD

Caltech

DTN
SENSE Orchestrate

Types of Interactions
- What is possible?
- What is recommended?
- Requests with negotiation
- Service status and troubleshooting

- Resource Discovery Service
  - Service Discovery
  - End-Point Listing
- Connectivity Service
  - Point-to-Point
  - Multi-Point
- Resource Computation Service
- Monitoring and Troubleshooting Service
SENSE Orchestrator and Services

Data Plane Connectivity Services:
- Point-to-Point (Layer 2)
- Multi-Point (Layer 2)
- Layer 3 QoS/Priority Options
  - Layer 2 (with L3 addressing)
  - Layer 3 Routed Network Connections
  - Quality of Service (L2 and L3)
  - Scheduling
  - Preemption
  - Negotiation
  - Batch Service Request
  - Strict and Loose hops
  - Lifecycle monitoring and debug

SENSE-O is based on StackV open source (geni public license) orchestration system:
- github.com/MAX-UMD/stackV.community
Scheduling and QoS

- We are “orchestrating” automated resources
  - which sometimes do not have scheduling or QoS features
- Therefore we cannot provide scheduling and QoS for all resources. Our approach is:
  - put those resources under Schedule and QoS control that we can AND when it adds value to do so
- The DTN network interfaces are a resource where we think this adds value
  - enables “SENSE Priority Flow” services
  - may include management of other host resources (cpu, interrupts, others) in the future
SENSE Enabled DTNs

- SENSE DTNs can be deployed next to production DTNs
- No impact to standard DTN operations
- Just adds a “priority flow” feature

- Scheduled and guaranteed resources, network and end system
- Can be included as part of application workflow planning
End to End Provisioning

Workflow:
- SENSE-RM provide Resource models to SENSE-Orchestrator
- SENSE Orchestrator builds end-to-end Model
- Batch Provisioning of P2P connections
- Dataflow between end systems using FDT
SENSE Services – Two General Categories

- **User Services**: Provide services to application workflow agents, as discussed in previous slides
- **Resource Management**: Help network and end site owners manage resources more efficiently
  - Allow greater average utilization of network resources (network path management and adjustment)
  - Encourage users to plan and optimize their use of network resources
SENSE Development Testbed

Notes:
- The provisioning end points indicated by circles are those which are of primary interest for initial demonstrations and testing. All edges on any SENSE controlled element can also be a provisioning end point.
- For Layer 2 services the statically controlled elements will need to have VLANs provisioned to create connections between SENSE controlled elements.
SENSE Demonstration
SENSE Demonstration

• Contact someone from the SENSE team
• SENSE Demonstration movie available here:  
  – https://tinyurl.com/sense-demo
SENSE Use Cases

- LHC, CMS
  - 13 Tiers 1s, 170 Tier 2s

- Superfacility
SENSE Key Components

- SENSE Network Resource Manager
- SENSE EndSite/DTN Resource Manager
- SENSE Orchestrator
SENSE Network Resource Manager (RM)

SENSE-RM Functions/Roles:
• Responsible for a specific set of Network Resources
• Generate realtime MRML Model
• Evaluate and respond to SENSE Orchestrator information and service requests (including negotiation)
• Provision network resources in support of SENSE services
• Provide status, monitoring, and debug functions

Developed by ESnet
SENSE DTN/End Site Resource Manager

EndSite/DTN-RM Functions/Roles:
- Responsible for a specific set of EndSite and DTN Resources
- Generate realtime MRML Model
- Evaluate and respond to SENSE Orchestrator information and service requests (including negotiation)
- Provision EndSite/DTN resources in support of SENSE services (includes networking stack of end systems)
- Provide status, monitoring, and debug functions
- QoS provided via OpenFlow (Open vSwitch) flow prioritization and/or TC (FireQoS)

Developed by Caltech
SENSE Orchestrator

Software Architecture

Based on UMD/MAX StackV open source (geni public license) model driven orchestration system:
- [github.com/MAX-UMD/stackV.community](https://github.com/MAX-UMD/stackV.community)

### Functional View

**SENSE-O**

**Model Build**

**MRML Model Delta**

**MRML Model Delta (Driver Specific)**

**Service Workflow**

**Intent Interpretation**

**MRML Model of Service Request**

**Computation**

**Drivers**
- SENSE-EndSite-RM Driver
- SENSE-Network-RM Driver
- SENSE-EndSite-RM Driver
- Other Drivers
Thanks!
SENSE Service Features

• Real time model construction by Network-RMs and DTN-RMs
• SENSE Orchestrator collection of RM models, and construction of an end to end infrastructure model
• Intent based Request for point-to-point service between DTNs
  – can include negotiation between Application Workflow Agent and SENSE Orchestrator.
  – Multiple rounds of query regarding available bandwidth, application can decide when to transition from query to request.
  – Initial negotiation example, now that models are dynamically constructed, custom queries can be built. Will work with use cases to guide this.
• Provision of end-to-end service
  – Automated DTN configuration along with network provisioning
  – QoS on ESnet and DTNs to give priority to SENSE Flow
SENSE
Interesting Research Questions

• Tradeoffs between scaling and real-time performance
  – How to make this dynamic/configurable to adjust based on conditions and application needs

• Which information/states should be routinely exchanged between Resource Manager(s) and Orchestrator? Which information should be accessible on demand?
  – How to make this dynamic/configurable to adjust based on different Resource Manager capabilities and policies?

• What is the right level of abstraction for Application Agent to SENSE system interactions? Is it necessary to provide a variable levels from highly abstract to very detailed and resource specific?

SENSE Key Specifications

- Modeling Schema and Ontology; Multi-Resource Markup Language (MRML) extended from Network Markup Language
  - https://github.com/MAX-UMD/nml-mrml
- SENSE Resource Manager API
  - public specification available soon
- SENSE Orchestrator API
  - https://app.swaggerhub.com/apis/xi-yang/SENSE-O-Intent-API/0.9.0
SENSE Orchestrator API

1. Service Request
   a. Resource Discovery Service
      • Topology Discovery
      • Service Discovery
      • End-Point Listing
   b. Connectivity Service
      • Point-to-Point
      • Multi-Point
   c. Resource Computation Service
2. Service Status
3. Service Modify
4. Monitoring and Troubleshooting Service
5. Service Delete

https://app.swaggerhub.com/apis/xi-yang/SENSE-O-Intent-API/0.9.0

- Allows applications to “query and negotiate” with network
- Transforms the network into a first class resource for workflow planning and optimization
Resource Computation Service enables “what is possible” and “negotiation” interactions

“what is possible”: one or more rounds of computation requests, without a provisioning request

“negotiation”: one or more rounds of feedback driven computation requests and a subsequent provisioning request agreement

Time block-maximum bandwidth: user asks for a specific time block and would like to know the maximum bandwidth. There may be multiple rounds of this what is possible question, and then user requests the specific bandwidth desired.

Bandwidth Sliding Window: user asks for specific bandwidth and duration, and provides an acceptable time window, e.g., 10Gbps for a 10 hour time window, sometime in the next 3 days. SENSE selects a specific time block and provisions service.

The SENSE infrastructure is designed to develop these types of services in DevOps manner, and customize for individual application agents.

Cannot do every computation possible, but can do any computation desired.
Model Based Control and Orchestration

- Turnkey, customizable services
- Rapid resource integration via dynamic models
- Modular Computation Elements facilitate custom service workflow construction
- A DevOps model for service construction
- Generic control of infrastructure for custom construction of user facing services