Orchestrating Intercontinental Advance Reservations with Software-Defined Exchanges

INNOVATING THE NETWORK FOR DATA INTENSIVE SCIENCE (INDIS) 2017

BY JOAQUIN CHUNG, RAJKUMAR KETTIMUTHU, NAM PHO, RUSS CLARK, HENRY OWEN

NOVEMBER 12, 2017
Motivation
Motivation

- Advance reservations are not flexible [1]
- International advance reservations typically follow a single path across a single domain
- Reservation success rate is low [2,3]

Software-Defined Exchange (SDX)

An SDX is a novel cyberinfrastructure that allows multiple independent administrative domains to share computing, storage, and networking resources in a programmatic way.
Software-Defined Exchange (SDX)
Agenda

1. Motivation

2. Background

3. Related Work

4. Architecture Overview

5. Design

6. Evaluation

7. Conclusions
What is SDN?

Software Defined Networking (SDN) separates the control plane from the data plane.
Agenda

1. Motivation
2. Background
3. Related Work
4. Architecture Overview
5. Design
6. Evaluation
7. Conclusions
Related Work

Multi-domain SDN Architectures
- Multi-domain network resource management [1] → Service level specifications
- Service provider SDN (SP-SDN) [2] → Technology domains (e.g., mobile, transport, data center, etc.)

Network Resource Management
- Resource Negotiation and Pricing Protocol (RNAP) [3]
- Service Negotiation and Acquisition Protocol (SNAP) [4]

Multi-path Advance Reservations
- OpenFlow Link-layer MultiPath Switching (OLiMPS) [5]
- Multi-path extension for OSCARS client [6]
Agenda

1. Motivation
2. Background
3. Related Work
4. Architecture Overview
5. Design
6. Evaluation
7. Conclusions
Architecture Overview

Science Network Services Orchestrator

Orchestrator

Interdomain links
Intradomain links
D-O: Domain to Orchestrator
U-O: User/Application to Orchestrator

End-to-End Science Network Services

Scientist User

Data Transfer Application

U-O Interface

D-O Interface
Agenda

1. Motivation
2. Background
3. Related Work
4. Architecture Overview
5. Design
6. Evaluation
7. Conclusions
Design – General Workflow
Design – Negotiation Protocol

START

Receive reservation request from user

Perform Path Computation

Decompose request on per-domain and per-SDX requests

Request individual offers

End-to-end service composition

Is service possible?

YES
Commit offers
END

NO
Abort offers

Negotiation Protocol
Design – Negotiation

Types of Domains:

- **Visible domains**: provide bandwidth offers (query available bandwidth)
- **Blind domains**: cannot provide bandwidth offers (i.e., traditional advance reservation systems)

Visibility scenarios for a negotiation protocol considering $N$ participant domains, with $M$ visible domains and $N - M$ blind domains:

1. **No visibility ($M = 0$)**: All participant domains are blind domains
2. **Full visibility ($M = N$)**: All participant domains are visible domains
3. **Partial visibility ($M \neq N$)**: blind domains and visible domains participate in the orchestration process
Negotiation Strategies

1. **Equal Splitting**: In this approach the orchestrator *divides* the original bandwidth request in equal parts among the participant domains.

2. **Partial Offers**: In this approach the orchestrator *contacts the visible domains* for bandwidth offers. If the orchestrator is able to compose an end-to-end service with these offers only, the orchestrator provisions the offers. Otherwise, the orchestrator *tries* to request the *remaining* bandwidth from *blind domains*.

3. **Full Offers**: In this approach the orchestrator *contacts all participant domains* for bandwidth offers. If the orchestrator is able to compose an end-to-end service with these offers, the orchestrator proceeds with provisioning, otherwise the reservation request fails.
Design – Provisioning (SDX Rules)

START

Receive reservation request from user

Perform Path Computation

Decompose request on per-domain and per-SDX requests

Request individual offers

End-to-end service composition

Is service possible?

YES

Commit offers

END

No

Abort offers

Provisioning
Design – SDX Rules Provisioning

SDX as interconnection points
Key insights:
1. Adv. reservations over VLANs
2. Data transfer protocols use multiple TCP streams
Agenda

1. Motivation
2. Background
3. Related Work
4. Architecture Overview
5. Design
6. Evaluation
7. Conclusions
Evaluation – Negotiation Protocol

- Simulation of random user requests to an orchestrator with 2, 3, and 4 participant domains
- With 3 domains we obtained 95% success rate for any negotiation strategy
- Full offers can achieve 99% success rate with 4 domains/paths available
### SDX Testbed Topology

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corsa DP2100</td>
<td>OpenFlow 1.5, multiple flow tables, multi-context virtualization, 48 Gb packet buffer, 10 Gbps line-rate</td>
</tr>
<tr>
<td>Dell PowerEdge R220</td>
<td>Ubuntu Server 16.04, 16 GB RAM, four Intel(R) Xeon(R) CPU E3-1220 v3 @ 3.10GHz processors, four port Gigabit Ethernet card</td>
</tr>
<tr>
<td>Customized Supermicro</td>
<td>Ubuntu Server 16.04, 8 GB RAM, four Intel(R) Xeon(R) CPU X3430 @ 2.40GHz, two Gigabit Ethernet interfaces</td>
</tr>
</tbody>
</table>

90 ms RTT between endpoints
**Bandwidth Splitting and TCP Streams**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS1</td>
<td>Tunnel 1: 100 Mbps, Tunnel 2: 900 Mbps</td>
</tr>
<tr>
<td>SS2</td>
<td>Tunnel 1: 200 Mbps, Tunnel 2: 800 Mbps</td>
</tr>
<tr>
<td>SS3</td>
<td>Tunnel 1: 300 Mbps, Tunnel 2: 700 Mbps</td>
</tr>
<tr>
<td>SS4</td>
<td>Tunnel 1: 400 Mbps, Tunnel 2: 600 Mbps</td>
</tr>
<tr>
<td>SS5</td>
<td>Tunnel 1: 500 Mbps, Tunnel 2: 500 Mbps</td>
</tr>
</tbody>
</table>

![Graph showing streams per tunnels with different throughput values for SS1 to SS5]
Agenda

1. Motivation
2. Background
3. Related Work
4. Architecture Overview
5. Design
6. Evaluation
7. Conclusions
Conclusions

Contributions

◦ An architecture for orchestrating international multi-path, multi-domain advance reservations in science networks and SDXs.

◦ Our orchestration architecture and negotiation protocols increases the reservation success rate from approximately 50% using single path to approximately 99% when four paths are available.

◦ Architectural approaches at the SDX level that enable novel science network services, while enhancing the performance of science data transfers over traditional approaches.

Future Work

◦ Large scale deployments and evaluations

◦ Novel science network services: scheduled migrations, multipoint-to-multipoint advance reservations
References


Thanks! Questions?
Bandwidth Splitting Service

By introducing SDXs in the provisioning process, we will be able to create multi-path, multi-domain advance reservations by splitting a bandwidth request among multiple participants.
Design – Negotiation Protocol

Phase 1

- Reservation
  - Request decomposition
    - Contact visible domains. Keep blind domains as backup
    - Compose end-to-end service from offers. If not possible, contact blind domains for remaining resources

User

Orchestrator

Domain 1 ... Domain N

SDXs

- ReqOffers
  - ...

- SendOffers
  - ...

11/11/2017
Design – Negotiation Protocol

[Diagram showing a sequence of messages between User, Orchestrator, Domain 1, ..., Domain N, and SDXs. Phases and messages include:
- **ReservationPrep**
- **ReservationResp**
- **Commit/Abort**
- **ReservationResp**

Key messages include:
- **Resource hold**
- **Path provisioning / Resource release**]
Orchestrator Implementation

Written in **Python** using an **agent-based approach**

- We control the WAN communication channel
- Site controller can provide their own API

Orchestrator communicates with the agents using the general remote procedure call (**gRPC**) protocol

- HTTP/2
- Protocol buffers
SDX Implementation

**AtlanticWave/SDX** controller: written in Python, using the Ryu SDN Framework, and OpenFlow
- REST API
- L2 Tunnels over VLANs
- Bandwidth offers

**Ryu SDN controller + Open vSwitch (OVS)** at each end for bandwidth splitting and aggregation
Evaluation – Negotiation Protocol

![Diagram showing network connections between RN1, RN2, RN3, Site A, Site B, Site C, and Site D, with R&E-1 and R&E-2 as clouds]

![Bar chart showing success rates for R&E-1, R&E-2, and MP-MD]
Throughput Baseline

Single L2 Tunnel @ 1 Gbps

Two L2 Tunnels @ 500 Mbps

m2m: memory-to-memory
d2d: disk-to-disk