COSMOS Optical Demo

https://wiki.cosmos-lab.org/wiki/tutorials/optical-network-example

Craig Gutterman, Columbia University
Artur Minakhmetov, Telecom ParisTech
Michael Sherman, Rutgers University
Jiakai Yu, University of Arizona

clg2168@columbia.edu, artur.minakhmetov@telecom-parsitech.fr,
msherman@winlab.rutgers.edu, jiakaiyu@email.arizona.edu
COSMOS Network Infrastructure
COSMOS Network Infrastructure

- Large-1
- Columbia
- Data Center @Columbia
- Large-2
- Colocation Site and Data Center @32 AoA

City College
Columbia
East Harlem

~11 km

32 AoA
Optical Architecture

- Enables configurable optical network
  - C-RAN
  - Edge computing
  - AR and VR applications

- Components
  - 10G Tunable Transceiver
  - 25G Ethernet interfaces
  - 100G FPGA connection
  - 320x320 Space Switch
  - Optical ROADM's (Reconfigurable Optical Add-Drop Multiplexer)
Optical Architecture

Central Space Switch (SS) @Columbia

To Test Ports
To BYOD
To ToR Switches

Data Center @Columbia

Colocation Site and Remote Data Center of NYU @ 32 Avenue of the Americas

20 x ROADM Units

PON VOA

2 x 1-to-10 PON Splits

1-to-32 PON Split

Single Fiber Pair
x Fiber Pairs

Whitebox ROADM Unit
Space Switch (SS)

R

Large-1
Large-2
Large-3
Large-4
Large-5
Large-6
Large-7
Large-8
Large-9

Columbia (on campus)
Columbia (residential)
City Assets
10 x Medium Nodes
To RoF Interfaces (i.e., Medium Nodes)

To Medium Nodes/RoF
SDRs
Serv.
Eth Sw
ToR

Large Node

To Medium Nodes/RoF

© COSMOS

RUTGERS
Columbia University
IN THE CITY OF NEW YORK
NYC
NYU
<sh> The City College of New York
IBM
Programmable Topologies
Programmable Topologies

CC @NYU – 32 AoA

CC @Columbia
Programmable Topologies
PON

CC @NYU – 32 AoA

10 km

CC @Columbia

OLT

1 km

PON Split

ONU

COSMOS

RUTGERS

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

NYU

IBM
Long Reach PON

CC @NYU – 32 AoA

CC @Columbia

OLT

ONU

CC Split

PON Split

1 km

10 km
WDM PON

CC @Columbia

OLT1
OLT2
PON Split

CC @NYU – 32 AoA

10 km
1 km

ONU
MidHaul Network

CC @NYU – 32 AoA

CC @Columbia

1 km

10 km
Converged mmWave/Fiber Transmission
mmWave Analog RoF

- Ultra-low latency, simple radio head
  - No digitization until data center
- COSMOS: Sub-6 GHz, plus select routes to 40 GHz
  - Can mix down from higher frequencies

(Figure courtesy of A. Kanno, NICT)
Optical Operation

• Remote experimentation
• User device insertion
• Today: configurable on request
• Future: user configurable
  – Basic topology controls
  – Advanced topology, power, components
    • Requires training to avoid damage to system
• Channels and links may be blocked for management purposes or due to other user reservations
Calient Space Switch
ROADM

- 3 Basic Sections
  - 96 chn MUX/DEMUX (WSS)
  - Booster Amplifier
  - Pre-Amplifier
- Single degree, bi-dir. ROADM
  - Combine to form multi-degree
- Python scripts
  - Booster/Preamp control
  - Booster/Preamp monitor
  - WSS connection Management
  - WSS connection monitor
- RYU SDN Controller
COSMOS Current State: Columbia Uni.

- Zussman’s lab
- Calient Switch S320
- ROADM 1
  - Line: 1.1.1..4
  - 1.1.8
  - 1.2.3.4
  - 5.5.1..4
  - 5.8.1..3
- ROADM 2
  - Line: 1.2.3.4
  - 1.4.1..4
  - 1.4.8
  - 1.4.1..4
- ROADM 3
  - Line: 1.2.3.4
  - 1.7.1..4
  - 1.7.8
  - 1.7.1..4
- ROADM 4
  - Line: 1.2.3.4
  - 2.2.1..4
  - 2.2.8
  - 2.2.1..4
- ToR
  - 29
  - 30
  - 31
  - 32
  - 3
  - 3
  - 3
  - 3
- srv1..2-co1
- srv1..4-lg1

29.34 – 1550 nm
COSMOS Current State: 32 AoA

- Fiber to 32 Ave of Americas facilitated by the city and ZenFi
Software Defined Optical Network

Optical Networks built in COSMOS could be SDN-controlled

Request types:
1. EDFA configuration
2. Ports configuration
3. Wavelength configuration

SDN functions:
1. RWA algorithm
2. NETCONF message
3. Resource allocation
Request Definition

1. EDFA configuration

\[
\text{traffic ID} + \text{message type} + \text{Node ID/IP} + \text{EDFA ID} + \text{configuration} \\
1 \quad \text{EDFAconfig} \quad 10.104.1.1 \quad 1 or 2 \quad \text{gain/power values} \\
\quad \text{in-service/out-of-service}
\]

2. Ports configuration

\[
\text{traffic ID} + \text{message type} + \text{Node ID/IP} + \text{port ID} + \text{configuration} \\
1 \quad \text{Portconfig} \quad 10.104.1.1 \quad 4101-4120 (4201) \quad \text{in-service/out-of-service} \\
\quad 5101-5120 (5201)
\]

3. Wavelength configuration

\[
\text{traffic ID} + \text{message type} + \text{Node ID/IP} + \text{Mux/Demux ID} + \text{configuration} \\
1 \quad \text{Add/TearDown} \quad 10.104.1.1 \quad 1 or 2 \quad \text{in-service/out-of-service} \\
\quad \text{block/not block} \\
\quad \text{start freq/end freq} \\
\quad \text{input-port/output-port} \\
\quad \text{connection ID}
\]
Optical SDN Control Flow

- Request Sender/Receiver:
  - generate request
  - send request
  - notice request sender

- Ryu SDN:
  - receive request and generate setup NETCONF message
  - send control message
  - deal with result message

- ROADM-20:
  - device setup
  - setup result message
  - send back result message
Experiment

Colocation Site and Data Center @32 AoA
Set up TOR Switch

• Each compute node has 2 25-Gb Ethernet connections to the TOR switch
• Configure the Interfaces to be set as VLAN switch ports
• Assign TOR and transceivers interfaces to VLANS
• Assign a wavelength (e.g., 1553.3 nm/193 Thz) to each transceiver

• Check VLANS

<table>
<thead>
<tr>
<th>NUM</th>
<th>Status</th>
<th>Description</th>
<th>Q Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>121</td>
<td>Active</td>
<td></td>
<td>U Te 1/32/1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U Tf 1/1/1</td>
</tr>
<tr>
<td>122</td>
<td>Active</td>
<td></td>
<td>U Te 1/29/1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U Tf 1/1/2</td>
</tr>
<tr>
<td>123</td>
<td>Active</td>
<td></td>
<td>U Te 1/31/1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>U Tf 1/1/3</td>
</tr>
</tbody>
</table>

• Configure VM interfaces and IP addresses
Establish Link 1

- **Preliminary Steps**
  - Connect line ports of ROADM4 and ROADM1 using the Calient Switch
  - Connect line Ports of ROADM2 and ROADM3 using the Calient Switch

- **Steps**
  - Add MUX/DEMUX connection from ROADM4 to TOR1
  - Add MUX/DEMUX connection from ROADM1 to TOR2

- **Example code**
  - python add_connection.py 10.104.1.4 1 10 in-service false 4102 4201 192950 193050 0 Exp1-FromTor1
  - python add_connection.py 10.104.1.4 2 10 in-service false 5101 5202 192950 193050 0 Exp1-TowardTor1
  - python add_connection.py 10.104.1.1 1 10 in-service false 4102 4201 192950 193050 0 Exp1-FromTor2
  - python add_connection.py 10.104.1.1 2 10 in-service false 5101 5202 192950 193050 0 Exp1-TowardTor2
Establish Link 1

python add_connection.py 10.104.1.4 1 10 in-service false 4102
4201 192950 193050 0 Exp1-FromTor1
Establish Link 1

python add_connection.py 10.104.1.4 2 10 in-service false 5101
5202 192950 193050 0 Exp1-TowardTor1
native@srv2-lg1:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.131 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.104 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.105 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.102 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.106 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=0.104 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=0.104 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=0.106 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=64 time=0.105 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=64 time=0.105 ms

--- 192.168.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9222ms
rtt min/avg/max/mdev = 0.102/0.107/0.131/0.010 ms
Establish Link 2

- Steps
  - Add MUX/DEMUX connection from ROADM4 to TOR1
  - Add MUX/DEMUX connection from ROADM1 to ROADM2
  - Add MUX/DEMUX connection from ROADM2 to ROADM1
  - Add MUX/DEMUX connection from ROADM3 to TOR3

- Example code
  - python add_connection.py 10.104.1.4 1 10 in-service false 4102 4201 192950 193050 0 Exp1-FromTor1
  - python add_connection.py 10.104.1.4 2 10 in-service false 5101 5202 192950 193050 0 Exp1-TorwardTor1
  - python add_connection.py 10.104.1.1 1 10 in-service false 4101 4201 192950 193050 0 Exp1-ROADM2
  - python add_connection.py 10.104.1.1 2 10 in-service false 5101 5201 192950 193050 0 Exp1-ROADM2
  - python add_connection.py 10.104.1.2 1 10 in-service false 4101 4201 192950 193050 0 Exp1-ROADM1
  - python add_connection.py 10.104.1.2 2 10 in-service false 5101 5201 192950 193050 0 Exp1-ROADM1
  - python add_connection.py 10.104.1.3 1 10 in-service false 4102 4201 192950 193050 0 Exp1-FromTor3
  - python add_connection.py 10.104.1.3 2 10 in-service false 5101 5202 192950 193050 0 Exp1-TorwardTor3
Establish Link 2

native@srv3-lg1:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.449 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.432 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.434 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.433 ms
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.425 ms
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=0.435 ms
64 bytes from 192.168.1.1: icmp_seq=7 ttl=64 time=0.434 ms
64 bytes from 192.168.1.1: icmp_seq=8 ttl=64 time=0.425 ms
64 bytes from 192.168.1.1: icmp_seq=9 ttl=64 time=0.426 ms
64 bytes from 192.168.1.1: icmp_seq=10 ttl=64 time=0.434 ms

--- 192.168.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9221ms
rtt min/avg/max/mdev = 0.425/0.432/0.449/0.025 ms