SELECTED TOPICS IN ADVANCED NETWORKING FOR SCIENTIFIC APPLICATIONS

Artur Barczyk/Caltech
Varenna School
Varenna, July 29th, 2014
Agenda

- Introduction

- Part I: Application-Aware Networks
  - Dynamic Circuits
  - OGF Network Services Interface
  - Examples (ANSE)

- Part II: Software Defined Networking
  - Introduction to SDN
  - OpenFlow
  - Programmable Networks
  - Use cases

- Part III:
  - Content Centric Networking

- Additional Resources
  - Networks for experimentation
INTRODUCTION
Different Views of a Network

... what a user might see:
Different views of a Network

... what a network engineer would see:
Different Views of a Network

... what an SDN network engineer would see:
But first…

• A refresher of basics and terms…
  (not a formal course on networking, just a collection of terms and definitions needed for the discussion later)
Circuit vs Packet Switched Networks

- **Circuit Switching**
  - Dedicated communication path between two stations
  - Set up prior to data exchange
  - Usually through several nodes in the network
  - Example: telephone network

- **Packet Switching**
  - Data sent in packets
  - Each packet’s header is inspected at each network node
  - Packets are passed from node to node based on header information and (local) routing database
  - Example: IP network

- **Virtual Circuit Switching**
  - Emulation of circuit switching on packet switched infrastructure
Network Layers

- Communication happens between corresponding entities in a layered structure

<table>
<thead>
<tr>
<th>OSI Reference Layers</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. physical</td>
<td>Physical characteristics</td>
</tr>
<tr>
<td>2. data link</td>
<td>Data delivery across a link or medium</td>
</tr>
<tr>
<td>3. network</td>
<td>End-to-end delivery of packets</td>
</tr>
<tr>
<td>4. transport</td>
<td>Segmentation and reassembly, error recovery</td>
</tr>
<tr>
<td>5. session</td>
<td>Start/control/stop sessions</td>
</tr>
<tr>
<td>6. presentation</td>
<td>Data formats, encryption</td>
</tr>
<tr>
<td>7. application</td>
<td>Interface between application and communication sw</td>
</tr>
</tbody>
</table>

Packet Router

<table>
<thead>
<tr>
<th>End System</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>End System</td>
<td>http</td>
</tr>
<tr>
<td>Packet Router</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>SONET</td>
<td>IP</td>
</tr>
</tbody>
</table>

July 29, 2014

Artur.Barcyk@cern.ch
Control & Data Planes

- **Data Plane**: processing of incoming data packets
  - Inspect, forward or drop
- **Control plane**: processes to build topology (RIB) and forwarding tables (FIB)
  - Needed to populate Forwarding Information Base used in the data plane
- In a traditional networks, each node operates processes in both control and data plane
Multiplexing

- Multiplexing is used to enable sharing of transmission medium between multiple devices
- Most common multiplexing schemes:
  - Wavelength-Division multiplexing
  - Time-Division Multiplexing
  - Statistical multiplexing
  - But also Space-Division multiplexing…
Circuit switching revisited

- Optical circuit switching equipment operate at Layer 1
  - Or even at “Layer 0” like e.g. MEMS switches

- Layer 1 optical equipment can switch based on wavelengths - Called Lightpath or Lambda-switching

- **Virtual circuit** connections above physical layer
  - SONET/SDH: TDM channels with defined capacity
  - MPLS emulates circuit connections using bandwidth profiles
  - TCP: a logical circuit connection between two end-systems
    - As opposed to UDP’s datagrams
Packets, Circuits and Flows

• Packet switches and routers forward based on each individual packet’s header information
  – In IP networks, typically only IP Destination address is matched against the Routing Information Base
    • plus QoS fields
    • sometimes also source address (PBR)

• Flow-based forwarding on the other hand…
  – Flow definition based on a set of parameters, such as e.g. \{IP\_SRC, IP\_DST, TCP\_PORT\}
  – Network device forwards packets based on forwarding database information for that flow – each packet in the flow takes the same path

• Flow-based forwarding is encountered in e.g. Link Aggregation scenarios (LAG, ECMP), as well as being the basis of OpenFlow
Other contents of a network environment

• In practical deployment, networks do not just forward packets

• Other services needed to function:
  – DNS
  – Possibly DHCP
  – AAA
    • NIS, LDAP, Shibboleth, etc.
  – Monitoring

• Networks deliver content…
Goal: reduce WAN latencies for data delivery
Strategically placed Cache Servers
Data replicated from the Origin Server(s)
Application-level technology
Usually an overlay on top of existing IP network infrastructure
APPLICATION AWARE NETWORKING

And network-aware applications
Network – Application Interface

• Any distributed system needs some form of network interaction

• Basic programming interface: Sockets
  – “puts bits on wire”
  – Restricted QoS

• Network Control
  – Reserve capacity
    • usually a NOC procedure, unless BoD system used
  – Prioritize traffic

• Network Monitoring and Analytics
  – To base smart decisions on
    • Reachability
    • Topology
    • Available capacity
OGF Network Services Interface (NSI)

- A standardized service interface between network domains
  - Note: A computing site is also a network domain!
- Open Grid Forum Working Group (NSI-WG)
**NSI multi-domain service construction**

- Two ways defined for “chaining” services: tree and chain

- Note: “ultimate Requester Agent” can be an end-user app

---

**Domain A**  **Domain B**  **Domain C**

- **NSA**
- **Aggregator NSA**
- **ultimate RA**
- **uRA**
- **Aggregator/ uPA**
- **Domain A**  **Domain B**  **Domain C**

pictures C. Guok, ESnet
NSI Services

- Currently foreseen services:
  - Connection Service (NSI-CS)
  - Topology Service (NSI-TS)
  - Discovery Service (NSI-DS)
  - Switching Service (NSI-SS)

- Future Services:
  - Monitoring Service
  - Protection Service
  - Verification Service
  - ...

July 29, 2014

Artur.Barczyk@cern.ch
NSI Connection Service

• NSI-CS: first NSI service standardized, currently v2.0
• Advance-reservation protocol
  – Mandatory Reservation parameters:
    • A-point, Z-point
  – Optional parameters:
    • Start time, end time
    • Bandwidth
    • Labels/VLAN IDs

• V2.0 supports optional modification of a reservation
  – Start time, end time and bandwidth
<table>
<thead>
<tr>
<th>NSI CS Message (abbreviation)</th>
<th>SM</th>
<th>Synch./Asynch.</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reserve (rsv.rq)</td>
<td>RSM</td>
<td>Asynch</td>
<td>The <em>reserve</em> message allows an RA to send a request to reserve network resources to build a Connection between two STP's.</td>
</tr>
<tr>
<td>reserveCommit (rsvcommit.rq)</td>
<td>RSM</td>
<td>Asynch</td>
<td>The <em>reserveCommit</em> message allows an RA to request the PA commit a previously allocated Connection reservation or modify an existing Connection reservation.</td>
</tr>
<tr>
<td>reserveAbort (rsvabort.rq)</td>
<td>RSM</td>
<td>Asynch</td>
<td>The <em>reserveAbort</em> message allows an RA to request the PA to abort a previously requested Connection that was made using the <em>reserve</em> message.</td>
</tr>
<tr>
<td>provision (prov.rq)</td>
<td>PSM</td>
<td>Asynch</td>
<td>The <em>provision</em> message allows RA to request the PA to transition a previously requested Connection into the Provisioned state. A Connection in Provisioned state will activate associated data plane resources during the scheduled reservation time.</td>
</tr>
<tr>
<td>release (release.rq)</td>
<td>PSM</td>
<td>Asynch</td>
<td>The <em>release</em> message allows an RA to request the PA to transition a previously provisioned Connection into Released state. A Connection in a Released state will deactivate the associated resources in the data plane. The reservation is not affected.</td>
</tr>
<tr>
<td>terminate (term.rq)</td>
<td>LSM</td>
<td>Asynch</td>
<td>The <em>terminate</em> message allows an RA to request the PA to transition a previously requested Connection into Terminated state. A Connection in Terminated state will release associated resources and allow the PA to clean up the RSM, PSM and all related data structures.</td>
</tr>
</tbody>
</table>

NSI Service Agent implementations

- **AutoBAHN**: GÉANT (EU)
- **BoD**: SURFnet (NL)
- **DynamicKL**: KISTI (KR)
- **G-LAMBDA-A**: AIST (JP)
- **G-LAMBDA-K**: KDDI Labs (JP)
- **OpenNSA**: NORDUnet (DK, SE, NO, FI, IS)
- **OSCARS**: ESnet (US)
Automated GOLE Fabric

John MacAuley, ESnet
• GLIF Open Lightpath Exchanges using automated provisioning
• Currently in R&D
• Demonstrated e.g. at SC’13
• Next demo planned for GLIF and SC’14

AutoGOLE Topology SC’13
Source: John MacAuley
Network of radio telescopes

Image by Paul Boven (boven@jive.nl). Satellite image: Blue Marble Next Generation, courtesy of Nasa Visible Earth (visibleearth.nasa.gov).

More info on http://www.evlbi.org/evlbi/evlbi.html
A typical eVLBI run

- 8-12 radio telescopes
- 1Gbps per telescope (future: 4Gbps)
  - Steady streams of data from antennas to correlator
  - Low jitter very important
- 8-12 hours
- 30-65 TB
NSI in Action: eVLBI

- Service provided by NRENs
- Standard client interface
- Client software developed by eVLBI collaboration
LHC Open Network Environment
  - VRF for current multipoint production use
  - Experiment/demonstration: Bandwidth-on-Demand
• Target: demonstrate multi-domain bandwidth reservation capability

Status: under construction
  - Multi-domain service created first, then connect end-sites
Building Network Awareness in LHC

• General: network monitoring in OSG and WLCG

• Based on information from monitoring systems such as MonALISA and PerfSONAR

• Specific project in CMS and ATLAS Experiments: Advanced Network Services for Experiments
  – Network Integration into
    • Workflow management (PanDA)
    • Data movement management (PhEDEx)
  – Measurement: PerfSONAR and MonALISA integration
  – Control: interface to provisioning systems (DYNES/OSCARS, NSI)
PanDA Workload Management

Production managers define task/job repository (Production DB)

Submitter (bamboo/JEDI)

PanDA server

Data Management System

Logging System

Local Replica Catalog

NDGF

ARC Interface (aCT)

pilot

pilot

Worker Nodes

End-user

Pull

https

OSG

condor-g

https

EGI

https

ARC Interface (aCT)

pilot scheduler (autopifactory)

Kaushik De

July 4, 2014
Network Integration in PanDA

- Concept: utilize network as a resource like other resources such as CPU, disk storage
  - Use network information for FAX brokerage
    - Brokerage should use concept of nearby sites
    - Jobs are sent to site with best weight, not necessarily the site with local data or available CPUs
  - Use network information for cloud selection
    - Best T2D site should be selected based on throughput from T1 to T2D measurements
- Network measurements are available at SSB (Site Status Board, Network view)
  - FAX xrdcp rate metric used for FAX brokerage
  - DDM Sonar metrics used for cloud selection
ATLAS Computing Model

- 11 Clouds
  - 10 T1s + 1 T0 (CERN)
  - Cloud = T1 + T2s + T2Ds (except CERN)
  - T2D = multi-cloud T2 sites
- 2-16 T2s in each Cloud

Task → Cloud
Task brokerage

Jobs → Sites
Job brokerage

Kaushik De, UTA
ANSE/Panda Use Cases

• Based on
  – Important to PanDA users
  – Enhance workload management through use of network
  – Should provide clear metrics for success/failure

1. Improve User Analysis Workflow
  – Include network information for routing of jobs to T1/T2 sites

2. Cloud Selection:
  – Optimize choice of T1-T2 pairings
  – Automate using network information

• Both use cases are development and testing phase
PhEDEx is the CMS data management toolkit
ANSE/PhEDEx – Dynamic Circuits

- Several points of circuit integration into PhEDEx
  - At the transfer-job level
  - At the link level (FileDownload agent)
  - At the instance level (FileRouter agent)

- Currently using a distributed testbed
  - T2_ANSE_CERN_1 & T2_ANSE_CERN_2
    - Both PhEDEx and storage nodes
  - T2_ANSE_Geneva & T2_ANSE_Amsterdam
    - PhEDEx and storage nodes separate
    - High speed link between AMS & GVA
    - 4x4 SSD software RAID 0 arrays
• FDT transfer tool integrates IDCP (OSCARS) calls

• Integrating FDT as transfer tool in PhEDEx naturally includes BoD capability

• Work ongoing on integration at FileDownload agent level

Circuits in PhEDEx at transfer level
Background traffic on the shared path limits rate to 5Gbps.
Network-Aware Applications Summary

- Network awareness can improve overall system performance
  - through acting on precise, real-time data on network state
  - through creating application-specific topologies such as point-to-point virtual circuits
- Network Services Interface (NSI) standard released, several implementations in development at many NRENs
- More examples of network-application interaction in the SDN part later
SDN
SOFTWARE DEFINED NETWORKING

Where we encounter OpenFlow and intelligent networks
Network business the traditional way

- Proprietary hardware, proprietary software
  - IPR
  - provide business edge
  - vendor lock-in

- Effects:
  - closed software
  - innovation slow, driven by vendors only
  - difficult to develop and evaluate new ideas
Drivers behind SDN

• Change in traffic patterns
  – away from single client - single server
  – local as well as wide area

• Appearance of cloud services
  – need for security, flexibility, scalability

• Manage complexity on large scales
  – Appearance of huge data centers
    • Multi-tenant facilities
  – Often global connectivity requirements

More in ONF SDN whitepaper at http://www.opennetworking.org
Software Defined Networking

• Basic SDN Paradigm:
  Separation of Network control plane from the data plane

- Enables network control by applications; provides an API to programmatically define network functionality
Central control

- Logically centralized control:
  - simplified operation
  - cost reduction
  - faster reconfiguration -> increased efficiency

- Physically distributed infrastructure:
  - scalability
  - redundancy
Network Programmability

- Network devices expose interface to third-party applications
- Applications provide the value
  - Networks applications:
    - Routing
    - Traffic Engineering
    - Flow Management
    - Network load balancing
  - End-user and service provider applications:
    - Access control and filtering
    - Computing resource load balancing
- Standards provide uniformity across vendor platforms
The Building Blocks

SDN BUILDING BLOCKS

ORCHESTRATION ↔ NFV AND APPLICATIONS ↔ MONITORING AND ANALYTICS

CONTROLLER ↔ CONTROLLER

NETWORK OS ↔ NETWORK OS

SWITCH ↔ SERVER ↔ NETWORK APPLIANCE

NORTHBOUND APIs

SOUTHBOUND APIs

SOURCE: RAYNO REPORT (WWW.RAYNOREPORT.COM)
What is SDN good for

- SDNs are used for
  - Network virtualization
    - Scalability
    - Robustness
    - Security
    - Logical separation (multi-tenant environments)
  - Centralization of management
    - Simplify operational aspects and workload
  - R&D
    - Fast development and deployment of new or non-IP protocols

- SDNs are/can/will be used in
  - Data center networks
  - Cloud systems (intra-/inter-site)
  - WANs
  - Transport networks
Example: Multi-Tenant Datacenter

- Some challenges in multi-tenant large data centers are
  - scalability
  - change management in large/complex deployments
  - elasticity, fast
  - ...
Current techniques are limiting performance:
  – Spanning Tree for loop avoidance
  – LAGs are link-local
  – scaling up involves much configuration work on each involved device
Multipath in Data Center

• Multipath can be achieved in several ways, e.g.
  – Multipath-TCP (IETF RFC 6824)
  – TRILL (IETF RFC 6325)
  – SPB (IEEE 802.1aq)
  – And/Or Load Balancing algorithms in SDN!
Example: Multi-Tenant Datacenter

- In addition, virtualization enables
  - host sharing
  - client-specific topologies
Example: Orchestration

- (Wikipedia: “…automated arrangement, coordination, and management of complex computer systems, middleware, and services”)
- For full service deployment need to orchestrate Storage, Compute and Network resources
- E.g. OpenStack
OpenFlow – SDN’s favourite protocol

- OpenFlow ≠ SDN
  - SDN is (technically, and with limitations) possible with SNMP, CLI, etc.

- OpenFlow = open standard for
  - Protocol for controller – device communication
  - Definition of packet processing in the switch

- Standardized by the Open Networking Foundation
OpenFlow switch components

• For packet look-up and forwarding
  – Flow Tables
  – Group Tables

• Control Channel
  – add, update, remove flow table entries

• OpenFlow Switch Ports:
  – Physical
  – Logical
    • e.g. LAG, tunnel, etc.
  – Reserved
    • ALL, CONTROLLER, TABLE, etc.
Flow Tables

- Matching fields at Layer 2, 2.5, 3 and 4
- Wildcards allowed
- Table miss entry - default action:
  - forward to controller, port or drop (default)

<table>
<thead>
<tr>
<th>Match fields</th>
<th>Priority</th>
<th>Counters</th>
<th>Instructions</th>
<th>Timeouts</th>
<th>Cookie</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC src</td>
<td>MAC dst</td>
<td>IP src</td>
<td>IP dst</td>
<td>TCP dport</td>
<td>Count</td>
<td>Instructions</td>
</tr>
<tr>
<td>*</td>
<td>50:25:.</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>531</td>
<td>Out port 7</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>1.2.3.*</td>
<td>80</td>
<td>77</td>
<td>local</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>2755</td>
<td>Controller</td>
</tr>
</tbody>
</table>

It is what the controller writes into the flow tables that determines the network behaviour.
Tables are processed in a pipeline

For each table:
1. Find highest priority matching flow entry
2. Apply Instructions
   1. apply actions
   2. update action set
   3. update metadata
3. Send match data and action set to next table
Instructions and Actions

• (Some) Instructions:
  – Apply actions <actions>
  – Clear actions <actions>
  – Write actions <actions>
  – Meter <actions>
  – Goto <table>

• (Some) Actions:
  – Output <port>
  – Drop
  – Push tag
  – Pop tag

• Tags specified (v1.3) can be
  – VLAN
  – MPLS
  – PBB

ONF OpenFlow Standard v1.3
For full document, see http://www.opennetworking.org
OpenFlow - The Controller

- Software typically running on commodity hardware
- Provides the API to user applications
  - Aka Northbound interface

http://www.opennetworking.org
Path to multi-domain SDN

• Not to forget: interactions between administrative domains

ONF SDN Architecture v1.0; [http://www.opennetworking.org](http://www.opennetworking.org)
Popular Controller Examples

- NOX (C++)  
  - http://www.noxrepo.org/

- POX (Python)  
  - http://www.noxrepo.org/

- Ryu (Python)  
  - http://osrg.github.io/ryu/

- Floodlight (Java)  
  - http://www.projectfloodlight.org/floodlight/

- OpenDaylight (Java)  
  - http://www.opendaylight.org/
OpenDaylight – Industry Driven

BROCADE  
CISCO  
CITRIX  
QOSMOS
Your Network is Information  
radware  
VERSA
NETWORKS
ERICSSON  
hp  
IBM  
ZTE 中兴  
FUJITSU  
guavus  
H3C
JUNIPER
NETWORKS  
Microsoft  
redhat
OIN  
ADVA
Optical Networking  
ARISTA  
AVAYA  
ciena
HUAWEI  
INocybe
TECHNOLOGIES  
intel
ARISTA  
AVAYA  
ciena
HUAWEI  
INocybe
TECHNOLOGIES  
intel
AZURE  
ADVA
Optical Networking  
ARISTA  
AVAYA  
ciena
HUAWEI  
INocybe
TECHNOLOGIES  
intel
CONTExTREAM  
Coriant  
CYAN
 PantheOn
Technologies  
PLEXXi  
PLUMgrid
DELL  
Extreme
Networks  
FLEXTRONICS  
NEC  
vmware
Artur.Barczyk@cern.ch
OpenDaylight – OpenFlow and beyond

VTN: Virtual Tenant Network
oDMC: Open Dove Management Console
D4A: Defense4All Protection
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Communication Protocol
SNMP: Simple Network Management Protocol
FRM: Forwarding Rules Manager
ARP: Address Resolution Protocol

Base Network Service Functions
- Topology Manager
- Stats Manager
- Switch Manager
- FRM
- Host Tracker
- ARP Handler

Service Abstraction Layer (SAL)
(Plugin Manager, Capability Abstractions, Flow Programming, Inventory, etc.)

OpenFlow
1.0
1.3

OVSDB
NETCONF
LISP
BGP
PCEP
SNMP

OpenFlow Enabled Devices

Open vSwitches

Additional Virtual & Physical Devices

Controller Platform

Network Applications Orchestrations & Services

Southbound Interfaces & Protocol Plugins

Data Plane Elements (Virtual Switches, Physical Device Interfaces)
Some Important Components

• Northbound interface: REST and OSGi

• BGP-LS (BGP-Link State)

• PCEP (Path Computation Engine Protocol)

• Southbound interface supporting OpenFlow and non-OpenFlow devices
Application Example: Multipath Controller

- **OpenFlow Link-layer Multi-Path Switching, OLiMPS**
- DOE funded project
- Extending capabilities of the Floodlight controller
- Load-balancing traffic over multiple possible end-to-end paths

SC’12 – The OLiMPS Demo Topology
Meshed Networks, Multipath

- Data connections (TCP) are point-to-point
- Classical IP routing constrains flows to a single path
- Reality: Networks are meshed, many paths possible, only one used

- Multipath forwarding helps increasing network efficiency
- Application “telling” the network controller its intentions increases efficiency even further
- Implemented using Floodlight controller
- Paper to be presented at HotSDN 2104

Example of WAN demo topology

From OLiMPS project – multipath with openflow

Local testbed with Application-Network interface
ESnet’s OSCARS management system using OpenFlow controller for traffic optimization

Caltech’s OLiMPS project created an interface between Floodlight OpenFlow controller and the OSCARS dynamic circuit system.

Additional capability of the controller: Create additional paths between OpenFlow devices.

I.e. create a topology optimized to the load distribution in the network.

Fits OpenDaylight architecture.
• SDN provides a new possibility for programmatic network interaction

• HEP computing should be involved in defining services provided by the networks, built on SDN
CONTENT CENTRIC NETWORKING

Where we meet CCN, NDN and friends
When Internet was invented, it was connecting resources
- TCP/IP: point-to-point connections between two entities
- IP: delivering packets to destination hosts

Today’s applications, ours included, care about content
• Applications deal with “what”, while the network deals with “where”
  – Lots of middleware needed to match these
  – Web services, CDNs, P2P, …

• Complexity arises when dealing with failover, security, etc.
  – E.g. if the server at A.B.C.D does not respond, it’s the application to react and possibly find a backup source for the data
  – E.g. you trust the server, but it’s the content that’s potentially dangerous

• Lot of the work in CDNs, redirection, caching deals with this mismatch

• Can we do a better design instead?
• Identify data rather than hosts?
• CCN is one of the **Future Internet Architectures** being developed and studied

• Specific projects include
  – Content-Centric Networking – CCN
    • Project at PARC
    • Code base developed: CCNx
  – Named Data Networking – NDN
    • NSF funded project since 2010, recently extended
    • Collaboration including PARC
  – and several other similar projects

• I will focus on the Named Data Networking (NDN) project in the following
Named Data Networking

• Basic Principle: Name Data instead of naming end-hosts

• Today’s Internet delivers packets to a destination address

• NDN delivers content identified by a given name to the client

• This is a basic change in semantics of the network service
Two NDN packet types

• Communication is driven by the receiving end
• **Interest Packets:**
  – Sent out by the data consumer, identifies desired data
• **Data Packets:**
  – Sent back by the node which has the desired content

---

**Interest packet**

<table>
<thead>
<tr>
<th>Content Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selector</td>
</tr>
<tr>
<td>(order preference, publisher filter, scope, ...)</td>
</tr>
<tr>
<td>Nonce</td>
</tr>
</tbody>
</table>

**Data packet**

<table>
<thead>
<tr>
<th>Content Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature</td>
</tr>
<tr>
<td>(digest algorithm, witness, ...)</td>
</tr>
<tr>
<td>Signed Info</td>
</tr>
<tr>
<td>(publisher ID, key locator, stale time, ...)</td>
</tr>
<tr>
<td>Data</td>
</tr>
</tbody>
</table>
Protocols

- Basis for NDN:
  - Named data replaces named end-points
  - Keeping the thin waist approach
Network node operation

- NDN routers remember the interface the request came in the PIT
- Forwards the Interest Packet looking up the name in the FIB
  - Populated by routing protocol
- Once the Interest Packet reaches a node that has the content, a Data Packet is sent back following the reverse path (as stored in the PIT)
Better-than Multicast

• Data Packets are cached in the routers’ Content Store

• Data Packets are then forwarded to all interfaces with registered interest in the router’s PIT
  – i.e. if multiple IPs received for same content – multicast!

• When next interest packet arrives for a named data in the content store (cache), a Data Packet is sent from the router, rather than forwarding the IP to the data source

• This provides for additional multicast-like operation

• With one big difference: no multicast request or protocol is necessary

• Added benefit: because of caching, destinations do not have to be synchronised - fits a pull model as opposed to push
NDN Operation – New request

Data Server (Publisher)

Interest Packets

Data Packets

Client(s)
NDN Operation – New request

Interest Packets

Data Server (Publisher)

Subsequent Interests are satisfied from the first node having the data stored

Data Packets

Client(s)
Why should we investigate use of NDN?

• A potential candidate technology to solve several issues, but do so at the network layer:
  – Optimal data distribution
  – Data caching
  – Popularity based data placement
  – Latency optimization for remote data access

• NDN could simply be the way the Internet works in the future

• How will this change the way we access and process data?
Some topics for investigation

- Caching strategies
  - Could we reduce the storage at end-sites to only permanent copies?
  - Rely on caching in the network?
- What is the correct data chunk?
  - File? Block? Event?
- Bulk data transfer strategies
  - E.g. multipath, multi-source, multicast
- Multipath forwarding
- Network-Application interface
  - Sockets?
  - Calendaring?
- Impact on workflows and job scheduling
  - Reduced latency through caching
  - Rely on remote data access as default?
- QoS and flow prioritization
- ...
Combining SDN and NDN

- For starters, SDN can be an easy way to create a high performance NDN test bed
- In which we want to investigate a possible design suitable for HEP data (and other data intensive science fields)

Caltech and Northeastern proposal to NSF: NDN Architecture for Data Intensive Science (NADIS)
SDN/NDN - Summary

- Software Defined Networking provides a powerful way to interact with the network
  - Needs engagement and collaboration with the network operators

- Named Data Networking is a fresh approach at the design of the Internet of the future
  - Designed with the content rather than end-points as basis for communication
  - Has many features which can benefit LHC data processing
  - Despite it being rather new, basic implementation and a test bed are available
    - The underlying ideas match very well with distributed data and computing models as in HEP computing
    - Impact on the LHC data processing models needs careful study
NETWORK TESTBeds

A non-exclusive list of examples for people interested in practical network innovation

July 29, 2014
If you want to test new network ideas, several testbeds might be available:

- **GENI** – generic network testbed, mostly US
- **OFELIA** – European Openflow testbed
- **GEANT OpenFlow test facility**
- **FELIX** – EU-JAPAN testbed for FI research
- ...
• Global Environment for Network Innovations
• “virtual laboratory for networking and distributed systems research and education”
• Mainly US based initiative, but not only
• GENI racks (3 “models”) installed on several campuses

Example: InstaGENI rack layout

http://www.geni.net
GENI allows experimenters to:

- Obtain compute resources from locations around the US
- Connect compute resources using Layer 2 networks in topologies best suited to their experiments
- Install custom software or even custom operating systems on these compute resources
- Control how devices in their experiment handle traffic flows
- Run their own Layer 3 and above protocols

http://www.geni.net
• European FP7 Project
• European OpenFlow Testbed Facility
• Project ended in 2013, but the GEANT Openflow Facility continues
• FEderated Test-beds for Large-scale Infrastructure eXperiments
• EU-JAPAN Project
• SDN-oriented service architecture for federating Future Internet facilities like OFELIA and JGN-X RISE
• Use high-capacity NSI-enabled networks as substrate
  – JGN-X, GEANT, GLIF, NRENs
• On-demand setup of “OpenFlow based network slices”
  – including network, compute and storage

http://www.ict-felix.eu
NDN Testbed
Summary and Conclusions

- Networks are not any more providing only transmission of bits between a pair of hosts
- New developments are in areas above providing bandwidth
- In development of distributed computing systems, we should leverage the new capabilities of the network systems
- Engagement with the network service providers (NRENs) is necessary in order to benefit most from it
QUESTIONS & DISCUSSION

Artur.Barczyk@cern.ch
Some Resources

1) OGF NSI WG http://redmine.ogf.org/projects/nsi-wg
2) Open Networking Foundation: http://www.opennetworking.org
3) Floodlight controller: http://www.projectfloodlight.org/floodlight/
4) OpenDaylight: http://www.opendaylight.org/
5) Named Data Networking: http://named-data.net/
6) CCNx: http://www.ccnx.org/