Why ISPs need SDN: SDN-based Network Service Chaining and Software-defined Multicast

ZKI Herbsttagung, Kaiserslautern, Germany, 24. Sept. 2014

Jeremias Blendin, Julius Rückert, David Hausheer

Department of Electrical Engineering and Information Technology
Technische Universität Darmstadt

E-Mail: {jblendin|rueckert|hausheer}@ps.tu-darmstadt.de
Web: http://www.ps.tu-darmstadt.de/

Supported by DFG CRC 1053
A Quick Introduction

- PS Lab established in May 2011
  - Associated with Department of Electrical Engineering and Information Technology

- Research on networked systems research areas
  - Software Defined Networking
    - Cross-layer adaptation mechanisms in DFG CRC MAKI
    - Collaborative project with Deutsche Telekom on business potentials of SDN/NFV
    - Joint work with NEC on traffic management in broadband access networks (NOMS 2014)
    - PhD/master theses on various SDN aspects (energy modeling, performance management)
  - Social-aware content distribution (EU FP7 eCOUSIN)
  - Energy-efficient access to mobile Cloud services (EU FP7 SmartenIT)
  - Related economic aspects (e.g. incentive mechanisms)
Motivation

- How to deal with increasing Over-the-Top (OTT) live video streaming traffic?
  - Today, OTT video streams across multiple domains mostly based on IP unicast
  - CDNs used to improve global content delivery
    - Usually end at edge of an ISP network
    - High load on ISP border and internal network
  - IP multicast deployed mostly within single domains only (e.g. ISP-internal IPTV)
  - P2P delivery helps to reduce content provider and CDN load, but not necessarily ISP

- Research Question
  - How to support efficient large-scale OTT live video streaming inside ISP networks?
SDM: Software-Defined Multicast

- Our approach: Software-Defined Multicast (SDM)
  - An SDN application to push overlay live streams into the underlay
    - Using an OpenFlow-based cross-layer approach

- Goal: reduce *intra*-ISP traffic
  - Content delivery with performance and costs at the level of IP multicast
    - Fully transparent to receivers (end-users)
    - Under full control of ISP

- SDM: an ISP service for OTT live streaming providers
  - Requires OpenFlow-enabled switches, incremental deployment possible

SDM: High-level Concept

- Live streaming systems highly rely on tree-based overlays

SDM Approach

- SDM service API allows OTT providers to create a virtual presence inside ISP network
- Virtual identity is promoted to end-users inside ISP
- Overlay connections pass through network-layer proxies
- Content is pushed to ISP-internal end-users through OpenFlow-based network layer multicast

Picture source: [Bl13, RBH13a]
SDM: Architecture

- **SDM Controller**
  - Provides external SDM service API
  - Service provisioning and admission

- **SDM Application**
  - Network-layer multicast functionality
  - Planning and management of multicast tree instances

- **Virtual Peer Application**
  - Network-layer proxy functionality
  - Gives outside peer a virtual presence inside the ISP network

- **ISP Network**
  - OpenFlow-enabled switches/routers

Figure source: [Bl13, RBH13a]
**SDM: Implementation**

1. SDM calculates multicast tree to reach a set of given group members on central OF controller

2. Source peer sends packets to specific IP unicast address as defined by ISP

3. Ingress switches mark packet with group ID (e.g., by rewriting destination MAC address)

4. Internal switches are configured according to multicast tree (on matching group ID: forward or duplicate packet)

5. Egress switches remove mark and rewrite destination MAC and IP to receiver’s address

---

## SDM: Qualitative Comparison

<table>
<thead>
<tr>
<th>ISP control</th>
<th>IP Multicast</th>
<th>Overlay Multicast</th>
<th>SDM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Traffic control and admission hard to achieve</td>
<td>• None / very limited: High volumes of hard to control traffic</td>
<td>• Full traffic and admission control</td>
</tr>
<tr>
<td>Delivery efficiency</td>
<td>• <strong>High</strong>: Traffic duplication at routers (L3)</td>
<td>• <strong>Low</strong>: Traffic duplication at clients</td>
<td>• <strong>Very high</strong>: Traffic duplication at switches (L2)</td>
</tr>
<tr>
<td>Network Deployment Requirements</td>
<td>• Requires multicast-capable routers</td>
<td>• Directly deployed on the current Internet</td>
<td>• Requires OpenFlow support within the ISP network</td>
</tr>
<tr>
<td>Content Provider Requirements</td>
<td>• Use of multicast protocol</td>
<td>• Use of overlay multicast protocol</td>
<td>• usage of API by CP</td>
</tr>
<tr>
<td>Client-side Requirements</td>
<td>• Clients need to support IP multicast protocol</td>
<td>• Clients run overlay multicast application</td>
<td>• <strong>None</strong>: Transparent IP unicast delivery</td>
</tr>
<tr>
<td>Transport Protocol</td>
<td>• UDP only</td>
<td>• No limitation</td>
<td>• UDP only</td>
</tr>
<tr>
<td>Scalability</td>
<td>• <strong>Low</strong>: limited scalability of multicast-capable routers</td>
<td>• <strong>High</strong>: fully distributed and scalable protocols are available</td>
<td>• Depends on OpenFlow hardware and management architecture</td>
</tr>
</tbody>
</table>

Based on [Bl13,RBH13b]
Quantitative Evaluation

- Setting and tools
  - Mininet-based virtual network [LHM10]
  - Open vSwitch instances [PGP+10]
  - Ryu-based OpenFlow controller
  - Real UDP-based video stream
  - ISP-like network structures

- Focus of the evaluation
  - Influence of ISP/overlay topology
  - Transmission efficiency (network traffic per served peer)
    - Intra-ISP network traffic
    - Network traffic at ISP border
  - Costs: Number of OpenFlow rules

---

Figure source: [Bl13]
Quantitative Evaluation

- **ISP and Overlay Networks**
  - Three different ISP topology variants
  - Three types of overlay topologies
  - 30 repetitions per ISP and overlay

- **Parameters**
  - 180 peers: 90 inside, 90 outside the ISP
  - Video bitrate: 150 kbps
  - Video length: 60s

Figure source: [Bl13]
Evaluation Results
Intra-ISP Traffic Volume

- ISP topology shows clear influence on volumes
- SDM traffic is at level of IP multicast
- >30% better than underlay-aware
- ISP network topology has no significant influence on relative difference between mechanisms

Figure source: [RBH13b]
Evaluation Results
Intra-ISP Traffic Volume

Figure source: [RBH13b]
Evaluation Results: Link Stretch of Peers inside the ISP Network

Figure source: [RBH13b]
Evaluation Results
Cross-border Traffic Volume

- No significant influence of ISP topology
- SDM traffic is at level of IP multicast and below underlay-aware for DT Hosts
- Low number of valid measurements for IP multicast in case of DT Depths

Figure source: [RBH13b]
Evaluation Results: Costs

- OpenFlow Rules
  - OpenFlow is rule based
  - Number of supported rules is main resource limitation of devices

- Specific result for our case:
  - On average 3 new rules per peer

- In general:
  - Upper bound given by the longest path from SDM instance switch to any egress switch

Picture source: [RBH13b]
SDM Summary

- SDM is an efficient OpenFlow-based cross-layer approach to push overlay live streams into the underlay
  - Allows content delivery with performance / costs at level of IP multicast
  - >30% better than underlay-aware

- Benefits of SDM over IP multicast
  - Fully transparent to receivers (end-users)
  - Under full control of ISP

- SDM offers new business potentials for ISP
  - ISP can provide “SDM as a service” for OTT live streaming providers
  - OpenFlow-enabled switches can be deployed gradually (the more the better)

- BUT: Multicast is just one network service out of multiple services that ISPs need to support
Network Service Chaining

- SDN/NFV facilitates faster Service Deployment
  - Traditional network service examples: redirects, firewalls, traffic shaping, web proxy, video optimizer

- Network Service Chaining Challenge
  - Complex services composed of service functions require more flexible deployment models

- Goal
  - Demonstrating the feasibility of dynamic network service chaining using SDN/NFV in a telecom environment

- Benefits for ISPs
  - Use of standard hardware
  - Simplified management and operations
  - Faster service deployment for end-users
Service Chaining Terminology and Core Challenges

1. Per-user mapping to chain instance

Abstract definition of a chain consisting of service functions

Instantiation of a service chain including several Service Instances

Traffic

Service Function

Service Function

Service Function

Service Function

Service Chain Instance

Service Node

Service Instance

Service Instance

Service Instance

Service Instance

Traffic

2. Routing between SIs inside chain instance

3. SI integration and traffic identification

Instantiation of a service function on a physical service node

Abstract description of a network service (e.g. firewall)

4. Leaving chain and mapping of backward traffic in same chain

Terminology based on IETF [QN13]
Approach

- User mapping
  - Based in the IP address of the user device
- Routing
  - MAC address based, inspired by StEERING [ZBB+13]
- Traffic identification
  - Port based traffic identification
  - Use one service instance per user
    - Feasibility of operating a large number of SIs on one server is shown in [BDH+13], which based on clickOS [MAR+13]
    - Proof-of-concept implementation relies on Linux containers
- Service instance integration
  - Isolation of instances
  - Packet conditioning
Service Instance Interface

- Network interface
  - Traffic identification through switch ports
  - Two links per Service Instance to the NFC system: ingress and egress
  - Efficient and reliable traffic identification
  - Compared to other approaches using e.g. statistical analysis

- Instance isolation
  - Block management traffic (e.g. ARP)
  - Exchange address information through the management system

- Benefits
  - Efficient traffic identification
  - Simplified deployment
  - No per instances configuration required
  - Overlapping network configurations possible
Implemented Service Chaining Architecture

[Diagram showing the architecture of service chaining with various components and their interactions.]

Legend:
- GUI
- Service instance data
- Network control (OpenFlow)
- SFCC control
- Network data

Architecture Components

- Service Function Chaining Controller (SFCC)
  - Responsible for client to chain mapping
    - Done using IP address of the client
  - Instantiation and management of service chains

- Service Node Controller (SNC)
  - Instantiation and management of SIs
  - SI orchestration on single service node
    - Here: a rather simple script (not focus of demo!)
    - Could conceptually be replaced by VM orchestration framework

- Service Function Chain Router (SFCR)
  - Installs OpenFlow rules that creates routes between SIs
  - SI routing done based on MAC addresses (inspired by [ZB+13])
    - Rewriting of destination MAC address to address of next SI in the chain instance
    - Limits applicability to L2 domains, could be changed to L3 routing (requires IP rewriting)

[JZB+13] Zhang et al. (Ericsson Research), ICNP 2013
Potential Use Cases / Services

- Service 1 (Default access)
  - New client detected
  - Assigned to default chain (redirection to landing page)
  - Everything else is blocked
  - End-user books service class

- Service 2 (Standard quality video)
  - Service forces client to a standard video quality

- Service 3 (High quality video)
  - Provider-driven change to a higher video quality
Conclusion

Why ISPs need SDN (and NFV)?

- SDN/NFV allows use of standard hardware
  - OpenFlow-enabled switches can be deployed gradually in most cases
- SDN/NFV enables faster service deployment for end-users
  - SDN-based services like SDM are fully transparent to receivers (end-users)
- SDN/NFV offers new business potentials for ISPs
  - Eg. ISP can provide “SDM as a service” for OTT live streaming providers
  - SDN/NFV simplifies management and operations
- Full control remains with the ISP
Thank you for your attention!
Questions?

{jblendin|rueckert|hausheer}@ps.tu-darmstadt.de
http://www.ps.tu-darmstadt.de/