# Software Defined Networking and Architectures

**Eugen Borcoci**

University POLITEHNICA Bucharest,
Eugen.Borcoci@elcom.pub.ro

# Software Defined Networking and Architectures

## Acknowledgement

The **overview (State of the Art) part is compiled**, based on several public documents and different authors' and groups work: Future Internet conferences public material, research papers and projects, overviews, tutorials, etc.: (see Reference list).

# Software Defined Networking and Architectures

## Motivation of this talk

- Future Internet challenges -> need to solve the current Internet limitation and ossification- as to support global integration of various forms of communications
  - **Evolutionary approach**
  - **Clean slate approach**
  - **New trends**

    **Software Defined Networks→ Software Defined Internet Architectures**

    **Cloud computing**

    **(ICN/CCN) Information/Content Centric Networking, (CON) Content Oriented Networking, (CAN) Content Aware Networking, …**

    **Combinations**

**UPB Master Courses 2015-2016**

# Software Defined Networking and Architectures

## Motivation of this talk (cont'd)

- **This tutorial**
  - Overview of recent architectural proposals and technologies, studied in research groups but also included in industry development, aiming to bring more flexibility and efficiency to IP networking and even to the whole Internet architecture.
- **Topics**

  - **Software Defined Networks (SDN) architecture**
    - Control and data planes are decoupled
    - Increased flexibility
    - Network intelligence is more centralized
      - better and more flexible control of the resource management
      - overall image of the system in the control plane
      - programmability of the network resources.
    - *OpenFlow* protocol for communication between planes
    - Attractive also for media-oriented and real time apps/services

# Software Defined Networking and Architectures

## Motivation of this talk (cont'd)

- **Topics**
  - **SDN links to other technologies**

    - **Cloud computing**
      - Infrastructure as a Service (IaaS) and Network as a Service (NaaS)

    - **Content/information oriented/centric networking**
      - propose to significantly (revolutionary) change the traditional approach
      - by decoupling the content and location at network level
      - creating the possibility for media objects to be directly leveraged in network nodes

  - **The above approaches : SDN/ Cloud computing/ ICN**
    - can be seen and developed as complementary
    - cooperating and supporting each other
    - aiming finally towards re-architecting the Internet

# CONTENTS

1. Software Defined Networks (Basics)
2. SDN Applications
3. SDN-OpenFlow
4. SDN Extensions and  Advanced Architectures
5. Other recent technologies- SDN approach
6. Conclusions

# CONTENTS

1. ☞ **Software Defined Networks (Basics)**
2. SDN Applications
3. SDN-OpenFlow
4. SDN Extensions and Advanced Architectures
5. Other recent technologies- SDN approach
6. Conclusions

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.1 Introduction**
- Current network architectures only partially meet today's requirements

- **Current network technologies limitations**

  - **Complexity that leads to stasis:**
    - Current status : many discrete sets protocols, separately defined for specific purposes
    - No fundamental network abstractions -> complexity
    - To add/move any device, IT admin. must (re) configure multiple specific HW/SW entities using device-level management tools
    - Today's networks reconfigurations are performed relatively in static way (to minimize the risk of service disruption)
  - The static nature of networks
    - not good for today's dynamic server environment, (server virtualization, VM migration)
      - applications are distributed across multiple virtual machines (VMs), which exchange traffic flows with each other.
      - VM migration : challenge for many aspects of traditional networking (addressing schemes, namespaces segmented, routing-based design).

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.1 Introduction**

- **Current network technologies limitations (cont'd)**

- **Limited capability for dynamic differentiated QoS** levels because of – usually static provisioning; not enough capability for dynamic adaptation to changing traffic, application, and user demands.

- **Inconsistent policies:**
  - Network-wide policy implementation -> need to configure ~$10^3$ - $10^4$ devices and mechanisms
  - Today's networks complexity → difficult to apply a consistent set of access, security, QoS, and other policies

- **Scalability issues:**
  - Complex network (10**5 network devices in data centers)

  - Over-subscription based on predictable traffic patterns is no more good;
    - in today's virtualized data centres, traffic patterns are highly dynamic and it is difficult to predict

  - Mega-operators (e.g. Google, Yahoo!, Facebook): scalability challenges
    - The number of of computing elements exploded
    - data-set exchanges among compute nodes can reach petabytes

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.1 Introduction**

- **Current network technologies limitations (cont'd)**

- **Scalability issues (cont'd)**

    - Need "hyper-scale" networks to provide high-performance, low-cost connectivity among many physical servers (need automation)
    - Carriers have to deliver better-differentiated services to customers
    - Multi-tenancy : the network must serve **large groups** of users with different applications and needs

- **Vendor dependency**
    - Carriers/enterprises want rapid response to changing business needs or user demands

    - They are limited by vendors' equipment product cycles (years)

    - Lack of standard, open I/F - limits the network operators ability to tailor the network to their individual environments

# 1. Software Defined Networking

- **1.1 Introduction**

- **Need for a new network architecture** **to answer to:**

  - **Changing traffic patterns:**
    - *Traffic patterns have changed significantly* within the enterprise data center: today's applications access different DBs and servers, creating a high M2M traffic before returning data to the end user device (different from classic client-server applications)

    - *Users- network traffic patterns changing:* they want access to corporate content and apps. from **any type of device, anywhere, at any time**

    - Enterprises : need of flexible computing model: *private public or hybrid cloud,* → additional traffic across the WANs

  - **Need of flexible access to IT resources:**
    - *Increasing usage of mobile personal devices* such as smart-phones, tablets, and notebooks to access the corporate network

    - Need to accommodate these personal devices while *protecting corporate data and intellectual property* and meeting compliance mandates

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.1 Introduction (cont'd)**
- **Need for a new network architecture (cont'd)**

  - **Cloud services development:**
    - Significant growth of public and private cloud services ( SaaS, PaaS, IaaS, NaaS,..)  on demand and à la carte
    - IT's needs for cloud services : security, compliance, auditing requirements, elastic scaling of computing, storage, and network resources,etc.

  - **Need for more bandwidth:**
    - today's  high volume of  data requires massive parallel processing on thousands of inter-connected servers
    - demand for additional network capacity in the data center
    - data center networks : need of scaling to very large size, while maintaining any-to-any connectivity
    - Media/content traffic high increase- need of more bandwidth

# 1. Software Defined Networking

- **1.1 Introduction**

- **Recent industry/research - new approaches:**
    - **Software- Defined Networking (SDN)** – aiming to transform networking architecture
    - **Open Networking Foundation** (ONF- non-profit industry consortium ) → OpenFlow I/F specifications for SDN

- **SDN major characteristics:**
    - the *Control Plane (CPl)* and *Data Planes (DPl)* are decoupled
    - network intelligence and state are logically centralized
    - underlying network infrastructure is abstracted from the applications
    - network programability

- **Promises for enterprises and carriers** :
    - higher programmability opportunities, automation, and network control
    - enabling them to build highly scalable, flexible networks
    - fast adapt to changing business needs

- *Source: Software-Defined Networking: The New Norm for Networks ONF White Paper April 13, 2012*

- *Note: after many years of strongly defending a completely distributed control approach in TCP/IP architecture- now a more centralized approach is proposed ….*

# 1. Software Defined Networking

- **1.1 Introduction**
- SDN + OpenFlow I/F(first standard) advantages:

  - *high-performance, granular traffic control* across multiple vendors' network devices; ability to apply comprehensive and wide-ranging policies at the session, user, device, and application levels
  - *centralized M&C* improving automation and management

  - *common APIs abstracting the underlying networking* details from the orchestration and provisioning systems and applications;
  - *flexibility:* new network capabilities and services with no need to configure individual devices or wait for vendor releases
  - *programmability* by operators, enterprises, independent software vendors, and users (not just equipment manufacturers) using common programming environments
  - *Increased network reliability* and *security* as a result of centralized and automated management of network devices, uniform policy enforcement, and fewer configuration errors

# 1. Software Defined Networking

- **1.1 Introduction**
- SDN + OpenFlow advantages (cont'd):
  - *better end-user experience* as applications exploit centralized network state information to seamlessly adapt network behavior to user needs
  - *protects existing investments* while future-proofing the network
  - SDN→ evolution to an extensible service delivery platform capable of responding rapidly to changing business, end-user, and market needs.
- **SDN – problems/open issues**
  - **centralisation- ( single point of failures)**
  - **horizontal and vertical scalability**
  - **backward compatibility**
  - **security**
  - **Real time capability od network control**
  - **Manufacturers/oparator resiliency**
  - **etc.**

# 1. Software Defined Networking

- **1.2 Earlier technologies related to SDN**
- **Open Signaling [3]**
  - **OPENSIG  WG** (~1995)- attempt to make Internet, ATM, and mobile networks more open, extensible, and programmable"
  - Ideas: separation between the communication HW and control SW
- **IETF WG** - > General Switch Management Protocol (GSMP
  - GSMPv3, June 2002, WG has been  concluded
  - general purpose protocol to control a label switch.
  - establish and release connections across the switch
- **Active Networking** [4] ~1995-2000
  - - programmable network infrastructure (for customized serviceses)
    - (1) user-programmable switches, in-band data transfer and out-of-band management channels
    - (2) control information organized in "capsules", which were program fragments that could be carried in user messages; program fragments would then be interpreted and executed by routers.
  - No large scale / significant success in practice - issues: security and perf.

# 1. Software Defined Networking

- **1.2 Earlier technologies related to SDN** (cont'd)
- **4D Project** [5]~2004, a clean slate design
  - separation between the routing decision logic and the protocols governing the interaction between network elements.
  - Consequences: NOX = operating system for networks in OF context

- **NETCONF,** [6], 2006
  - IETF Network Configuration WG (still active) : NETCONF defined a management protocol for modifying the configuration of network devices.
    - network devices have APIs - (to send /retrieve) configuration data
  - still - no separation Control/Data Plane

- **Ethane,** [8], 2006- precursor to SDN
  - new network architecture for enterprise networks
  - centralized controller to manage policy and security in a network
  - two components:
    - a controller to decide if a packet should be forwarded
    - Ethane switch : a flow table and a secure channel to the controller

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.2. Earlier technologies related to SDN (cont'd)**

- **IETF WG ForCES Forwarding and Control Element Seperaration, 2003, [7].**
    - A parallel approach to SDN
    - some common goals with SDN and ONF
    - Differences:
        - ForCES: the internal network device architecture is redefined as the control element separated from the forwarding element, but the combined entity is still represented as a single network element to the outside world
            - Aim: to combine new forwarding hardware with third-party control within a single network device where the separation is kept within close proximity (e.g., same box or room)
        - **SDN: Contrl Plane (CPI) is totally moved from net device**
    - FORCES published docs on : arch. framework, interactions, modelling language, forwarding  element (FE) functions, protocol between Ctrl and FE

**UPB Master Courses 2015-2016**

- **1.3 Early SDN products**
- Early  SDN **products and activities examples**

- 2008: Software-Defined Networking (SDN) : NOX Network Operating System [Nicira]; OpenFlow switch interface [Stanford/Nicira]

- 2011: Open Networking Foundation (72 members) : Board: Google, Yahoo, Verizon, DT, Msoft, F'book, NTT ; Members: Cisco, Juniper, HP, Dell, Broadcom, IBM,…..

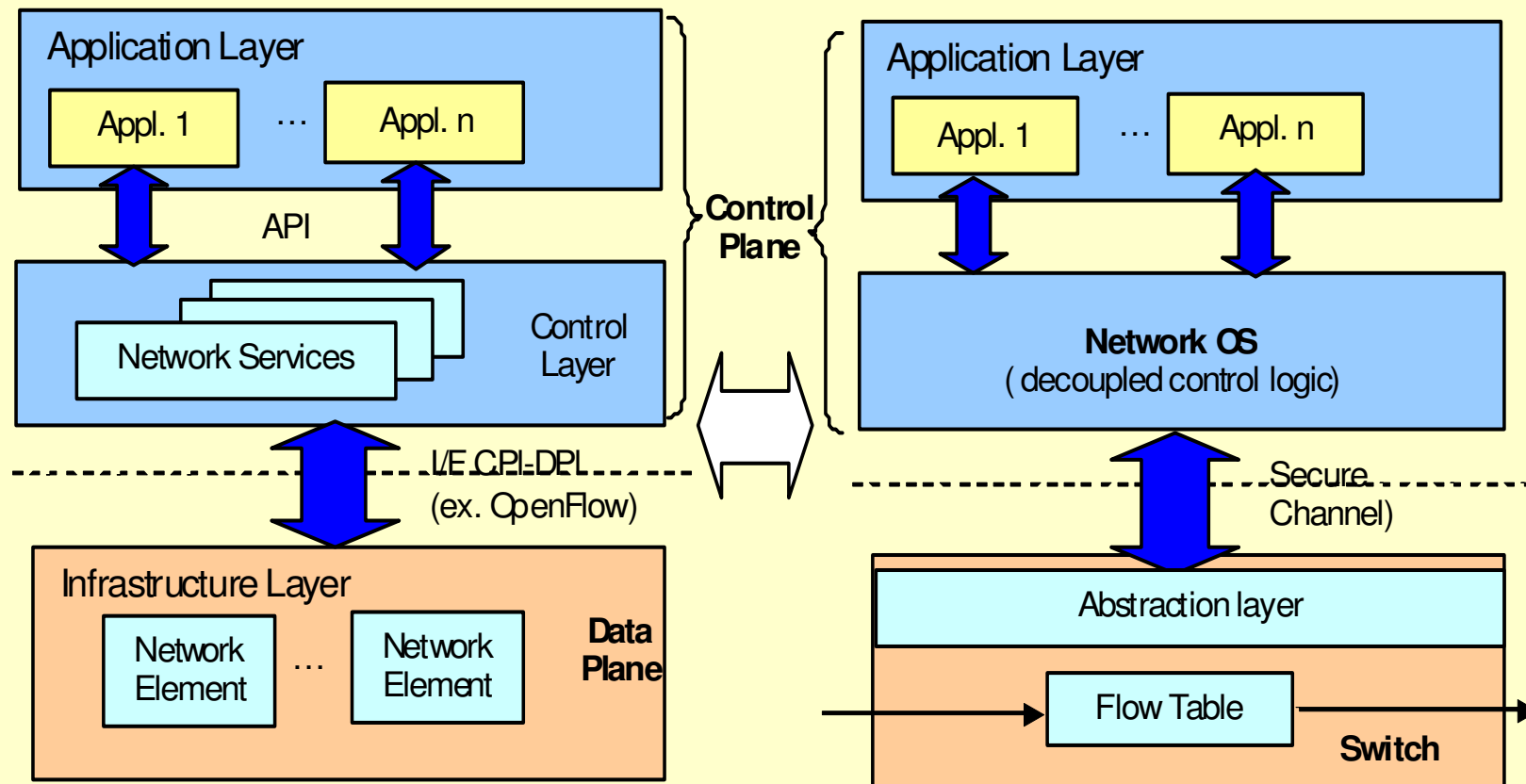# 1. Software Defined Networking

- **1.4 SDN Basic Architecture**

  - **Evolutionary**

  - **CPI and DPI are separated**

  - Network intelligence is (logically) centralized in SW-based SDN controllers, which maintain a global view of the network.

  - Execute CPI SW on general purpose HW
    - Decoupled from specific networking HW
    - CPI can use use commodity servers

  - Data Plane (DPI ) is programmable

  - Maintain, control and program data plane state from a central entity

  - The architecture defines the control for a network (and not for a network device) The network appears to the applications and policy engines as a single, logical switch

  - This simplified network abstraction can be efficiently programmed

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

## 1.4 SDN Basic Architecture



**SDN Generic Architecture**

UPB Master Courses 2015-2016

# 1. Software Defined Networking

- **1.4 SDN Basic Architecture**
- **Control Plane**
    - **Control Applications/Program**
        - operates on view of network **:**
        - performs different functions ( routing, traffic engineering, QoS, security, etc.)
        - **Input**: global network view (graph/database)
        - **Output**: configuration of each network device
        - Control program is not a distributed system
            Abstraction hides details of distributed state
    - **Network OS:** distributed system that creates a consistent, global and up-to-date network view
        - In SDN it runs can on controllers ( servers)  in the network
        - It creates the "lower layer" of the Control Plane
        - Examples: NOX, ONIX, Trema,  Beacon, Maestro, …

- **Data Plane** : forwarders/switches ( Forwarding elements -FE)
    - NOS uses some abstraction to:
        - Get state information from FE
        - Give control directives to FE

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.4 SDN Basic Architecture**
- **Advantages**
- **Centralization allows:**
  - To alter network behavior in real-time and faster deploy new applications and network services (hours, days not weeks or months as today).

  - flexibility to configure, manage, secure, and optimize network resources via dynamic, automated SDN programs ( not waiting for vendors) .

- APIs facilitate implementation of:
  - common network services: routing, multicast, security, access control, bandwidth management, QoS, traffic engineering, processor and storage optimization, energy usage
  - policy management, custom tailored to meet business objectives
    - Easy to define and enforce consistent policies across both wired and wireless connections on a campus

- Manage the entire network : intelligent orchestration and provisioning systems

# 1. Software Defined Networking

- **1.4 SDN Basic Architecture**
- **Advantages ( cont'd)**

  - ONF studies open APIs to promote multi-vendor management:
    - possibility for **on-demand resource allocation, self-service provisioning,** truly virtualized networking, and secure cloud services.
  - SDN control and applications layers, business apps can operate on an **abstraction of the network**, leveraging network services and capabilities without being tied to the details of their implementation.

- Open SDN issues/problems
  - Balance between distribution – centralization ( physical/logical)

  - Scalability
    - How many controllers
    - Their location
    - Synchronization

  - Reliability

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.4 SDN Basic Architecture**

- **Network OS:**
  - **Distributed system that creates a consistent, updated network view**

  - **Executed on servers (controllers) in the network**

  - **Examples: NOX, ONIX, HyperFlow, Floodlight, Trema, Kandoo, Beacon, Maestro,..**

- **Uses forwarding abstraction in order to:**
  - **Collect state information from FE**
  - **Generate commands to FE**



**UPB Master Courses 2015-2016**
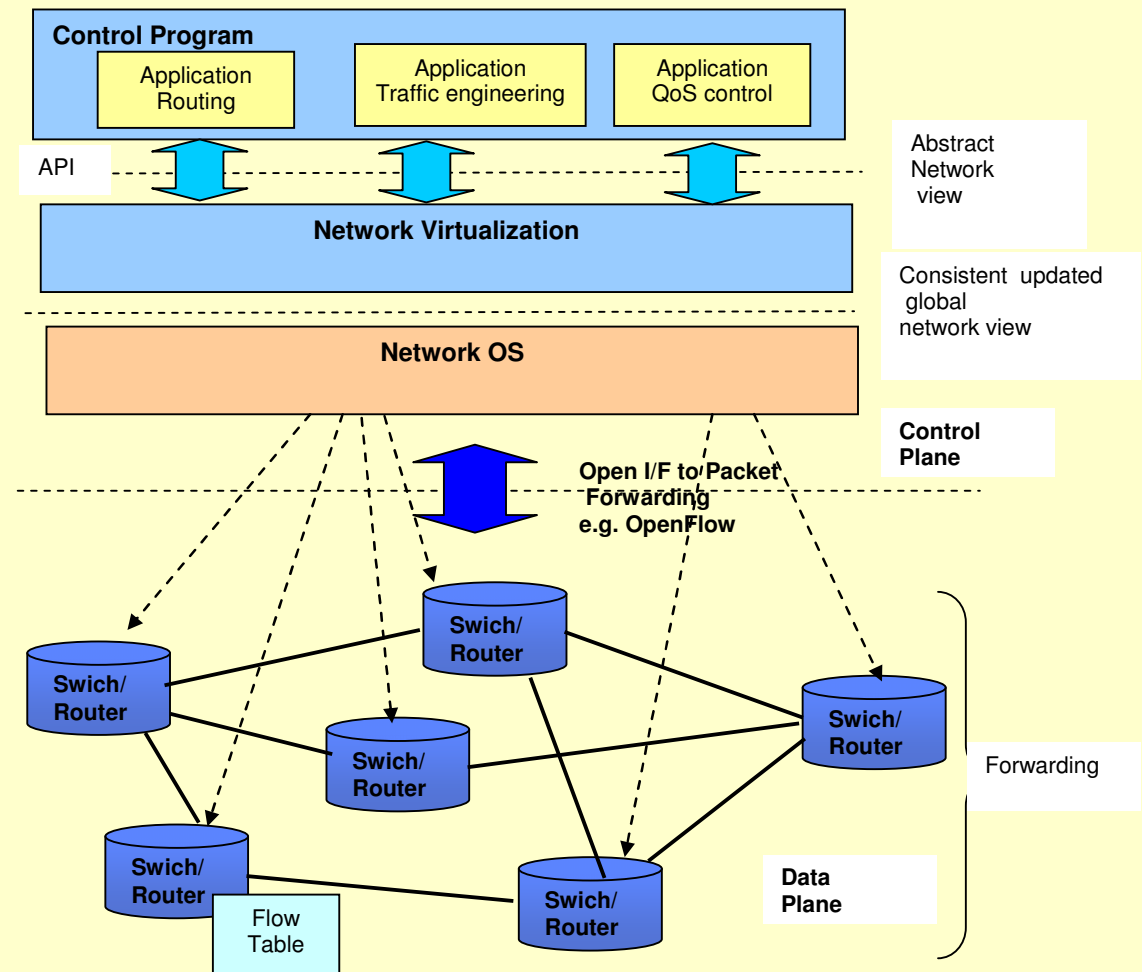
# 1. Software Defined Networking

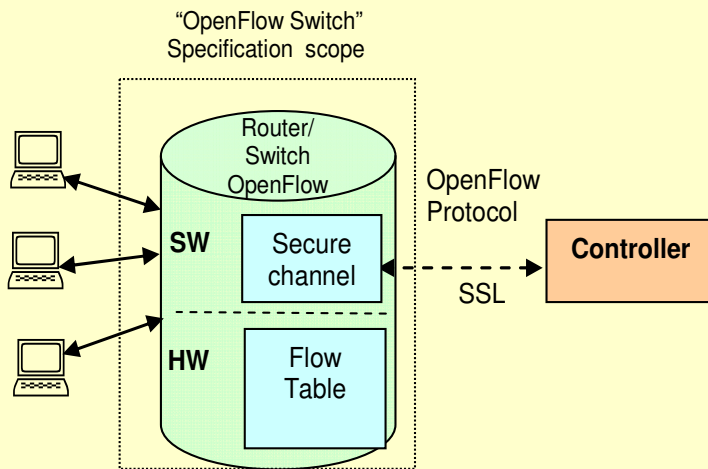- **1.4 SDN Basic Architecture**
- **OpenFlow summary**
  - the **first SDN standard** communications: CPI-DPI I/F

  - allows direct access to the Fwd. Plane of network devices (FE = switches or routers), both physical and virtual (hypervisor-based)

  - allows to move CPI out of the FEs to logically centralized control software

  - specifies basic primitives to be used by an external SW application to program the FwdPl (~ instruction set of a CPU)

  - *Flow concept : to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SDN control SW*

  - network can be programmed on a per-flow basis ( provides – if wanted- extremely granular control), enabling the network to respond to real-time changes at the application, user, and session levels

  - allows IT admin to define how traffic should flow through FEs based on parameters such as usage patterns, applications, and cloud resources

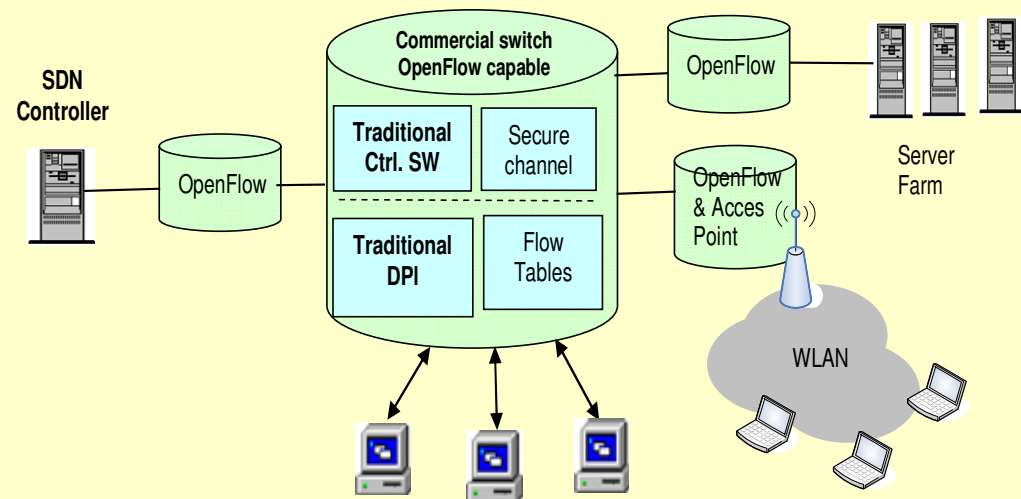**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- ## 1.4 SDN Basic Architecture: Open Flow
- ### *Source Ref1 [1]: "OpenFlow: Enabling Innovation in Campus Networks"- N.McKeown, T.Anderson, H.Balakrishnan, G.Parulkar, L.Peterson, J.Rexford, S.Shenker, J.Turner, 2008*
  - http://www.openflow.org/documents/openflow-wp-latest.pdf.

"OpenFlow Switch"
Specification scope

Router/
Switch
OpenFlow

**SW**
Secure channel

OpenFlow
Protocol

**Controller**

SSL

**HW**
Flow Table

Ref1: Figure 1: Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel.

| In Port | VLAN ID | Ethernet | | | IP | | | TCP | |
|---------|---------|----|----|------|----|----|-------|-----|-----|
| | | SA | DA | Type | SA | DA | Proto | Src | Dst |

Table 1: The header fields matched in a "Type 0" OpenFlow switch.

**SDN Controller**

OpenFlow

Commercial switch
OpenFlow capable

**Traditional Ctrl. SW**

Secure channel

OpenFlow
& Acces Point

OpenFlow

Server Farm

**Traditional DPI**

Flow Tables

WLAN

Ref1: Figure 2: Example of a network of OpenFlow-enabled commercial switches and routers.

# CONTENTS

1. Software Defined Networks (Basics)
2. ☞ **SDN Applications**
3. SDN-OpenFlow
4. SDN Extensions and Advanced Architectures
5. Other recent technologies- SDN approach
6. Conclusions

# 2. SDN Applications

- **Enterprise Networks**
- SDN – unify and improve M&C;
  - support to  programmatically enforce/ adjust network policies as well as help net monitoring and tune performance.

  - eliminate middleboxes (NAT, firewalls, load balancers, access control, DPI, etc.) by  integrating their functionality within the controller

  - configuration changes (currently- common networks instability source) can be performed in a more flexible and consistent way

  - a set of high-level abstractions are proposed that allow admin to update the entire network

  - packets  are processed in consistent way at network level based on flow concepts

# 2. SDN Applications

- **Data Centers**
  - High volume and dynamic traffic, large scale
  - Management and policy enforcement is critical especially to avoid service disruption.
  - Still some Data Center are provisioned based on estimation of peak demand
    - (-) high percentage of time under-utilization
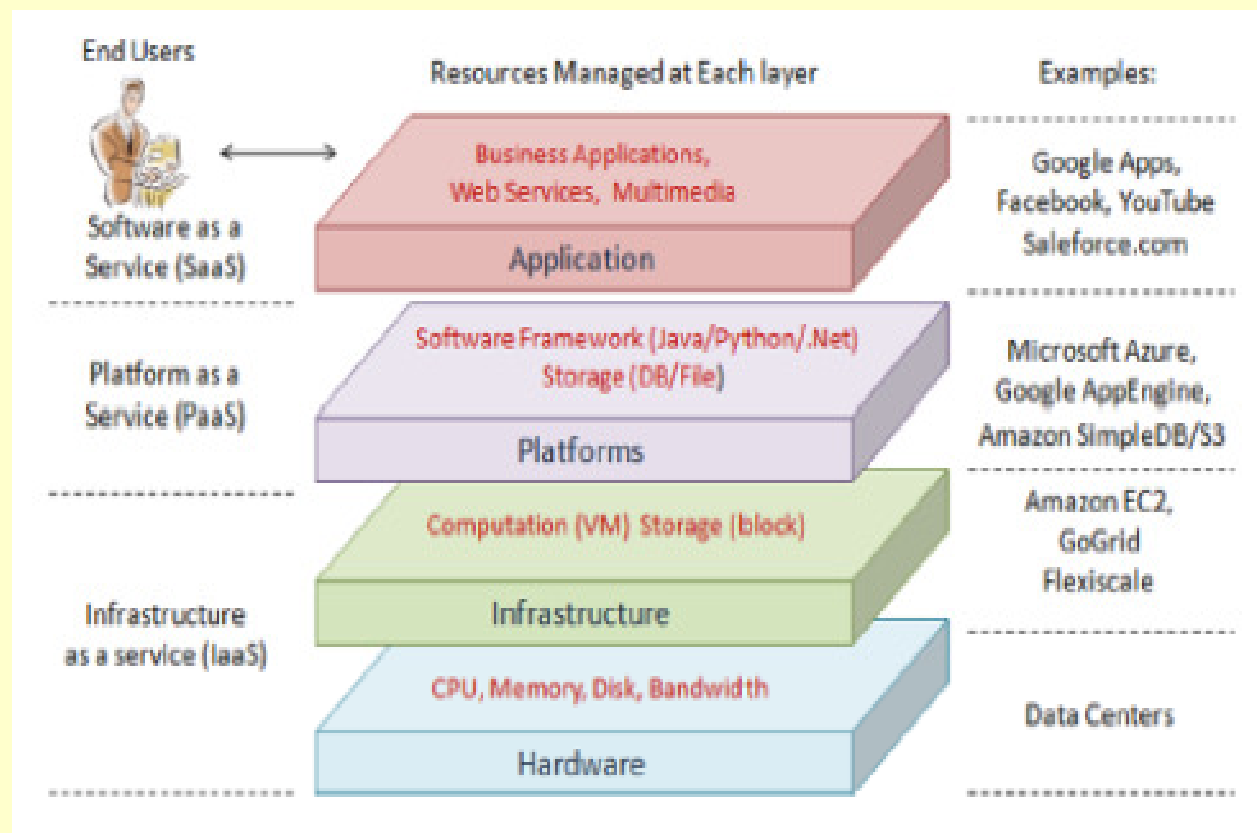    - (+) but answer to high demand is very fast

Basic layered design of data center network infrastructure



**UPB Master Courses 2015-2016**
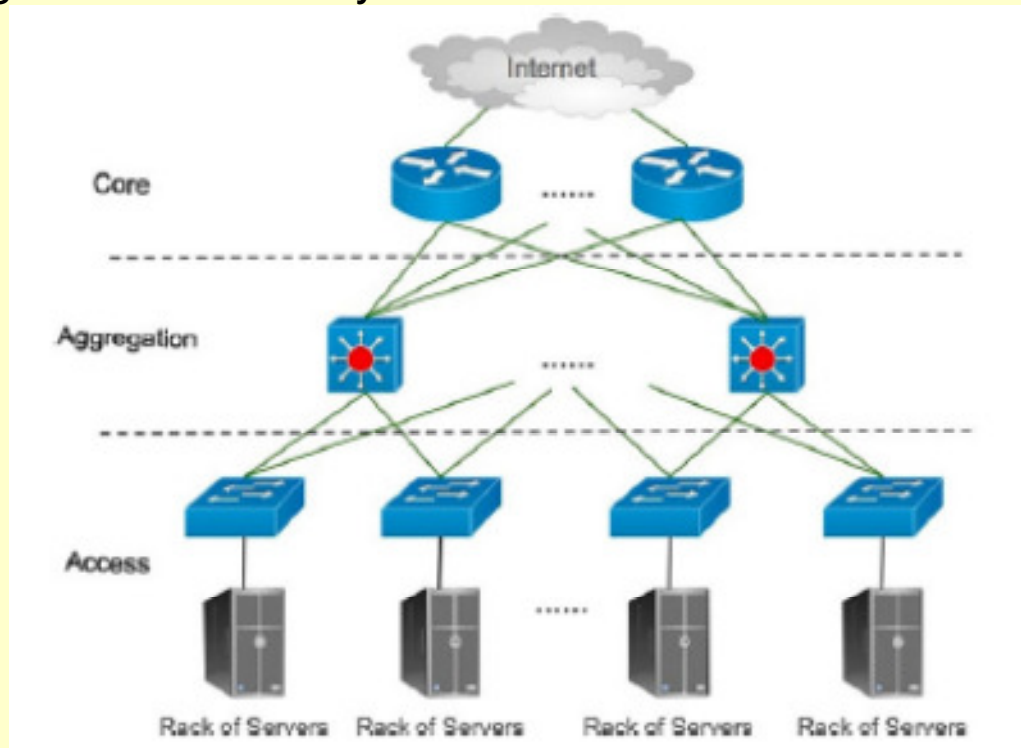
# 2. SDN Applications

- **Data Centers ( cont'd)**

- Cloud computing Architecture
  - need to map the SDN architecture onto this one

# 2. SDN Applications

- **Data Centers ( cont'd)**

- **Energy consumption** – important in big Data Centers (10-20% for networking) → need of better energy management
  - Proposal: SDN based Network-wide power management, (elastic tree, savings 20-65% depending on traffic conditions-have been shown)

  - Savings can be increased if used in cooperation with server management and virtualization
    - controlling the migration of VMs as to increase the number of machines and switches that can be shut down
    - however such traffic management must be balanced with scalability and performance overheads.

**UPB Master Courses 2015-2016**
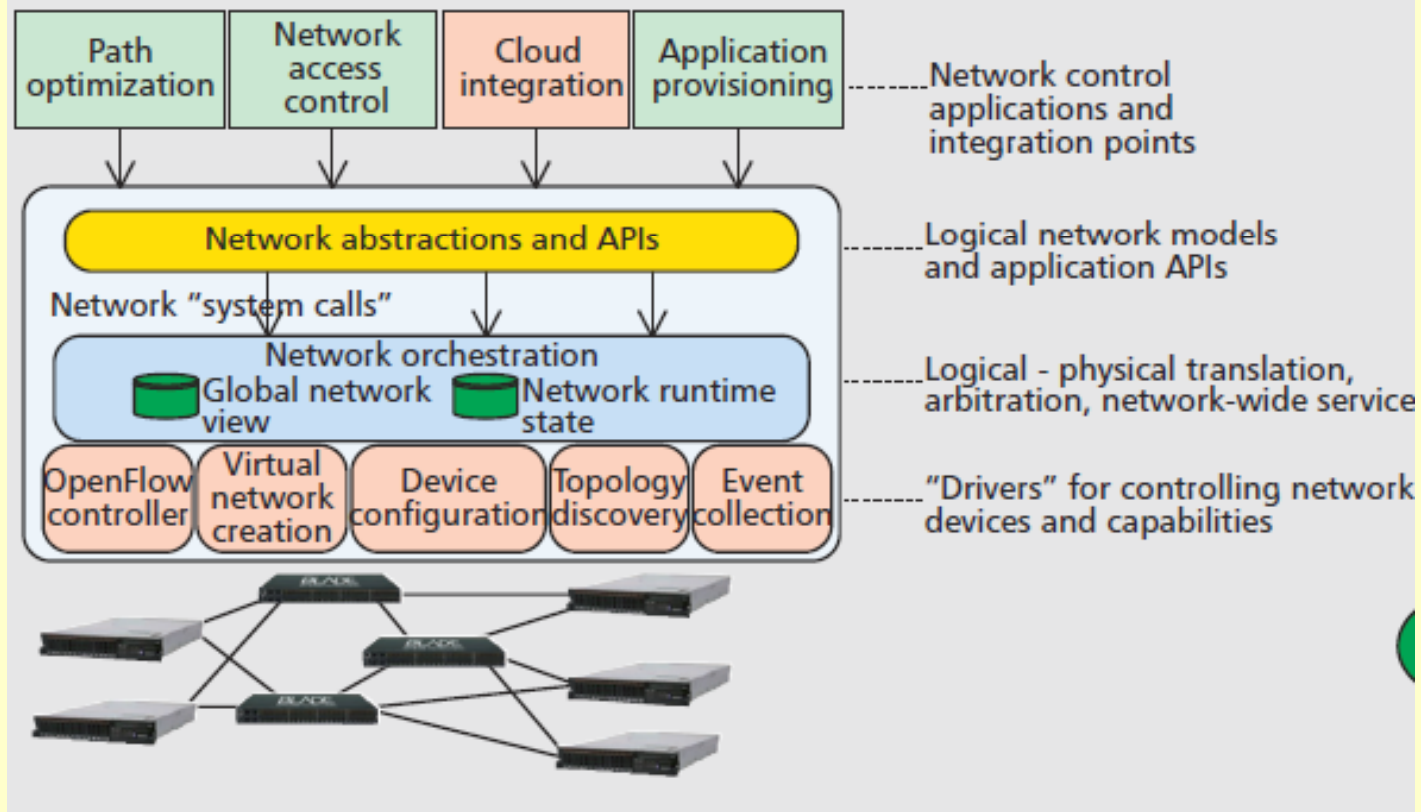
# 2. SDN Applications

- **Data Centers (cont'd)**
  - Issues: [2] :OF sometimes excessively centralises control processing
    - while only few "significant" flows would be needed to be managed → bottlenecks in the control communication ( if fine granularity is wanted)

  - Solutions: proactive policies and wild-card rules, but the cost is paid with less to manage traffic and gather statistics.

  - Proposals done:  design changes like
    - keep control of flows as much as possible in the data plane while maintaining enough visibility at controller level for effective flow management.
    - pushing back again responsibility on many flows to the switches and adding more efficient statistics collection mechanisms, for significant" flows (e.g. long-lived, high-throughput) identified and managed by the controller.
    - Effect: reducing the control overhead and having fewer flow table entries

# 2. SDN Applications

- ## Data Centers (cont'd)- Examples

  - *Ref: Mohammad Banikazemi, David Olshefski, Anees Shaikh, John Tracey, and Guohui Wang,"* **Meridian: An SDN Platform for Cloud Network Services",** IEEE Communications Magazine • February 2013
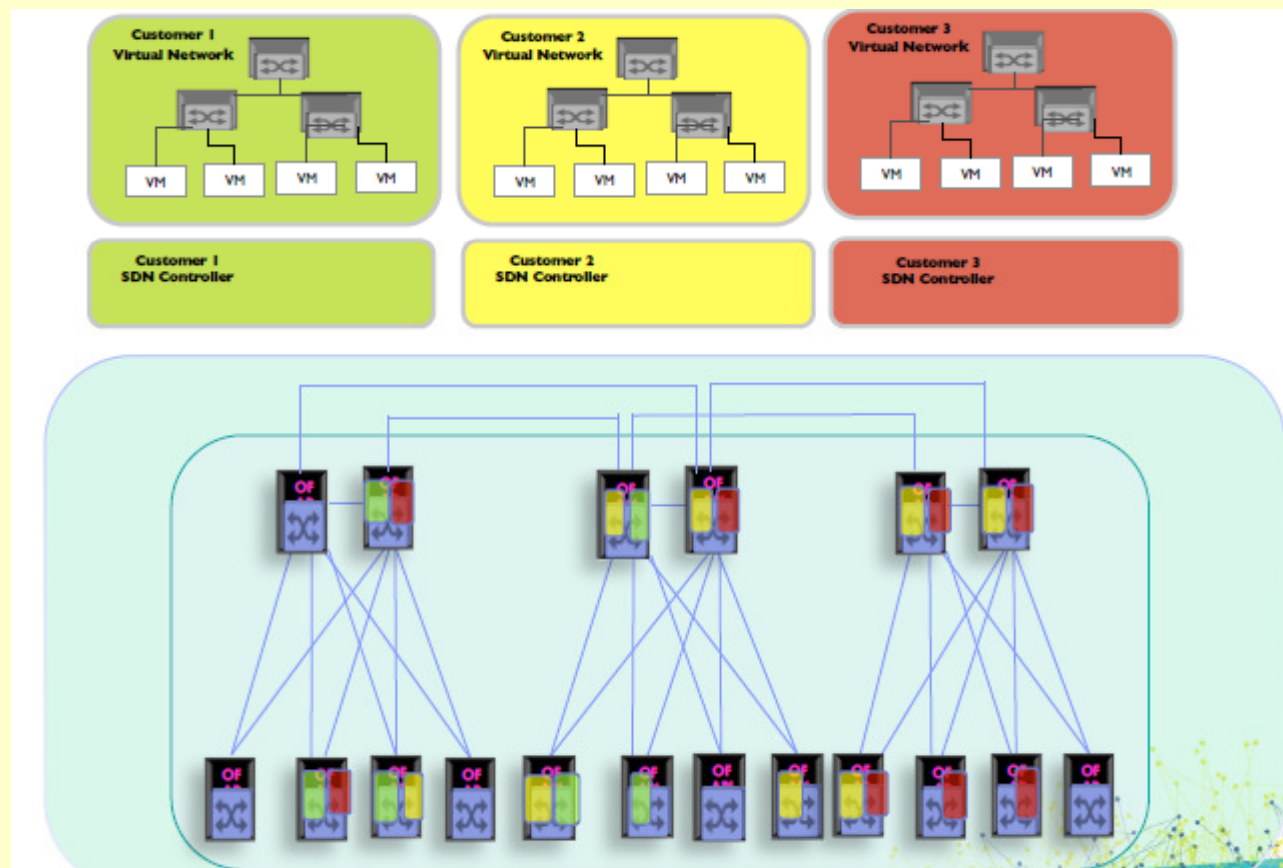
## Data Centers (cont'd)

- **Source:** R.Saha, IBM, A.Agarwal, Google "SDN Approach to Large Scale Global Data Centers"
- **Cloud – multi-tenncy**
- **Network as a Service**



**UPB Master Courses 2015-2016**
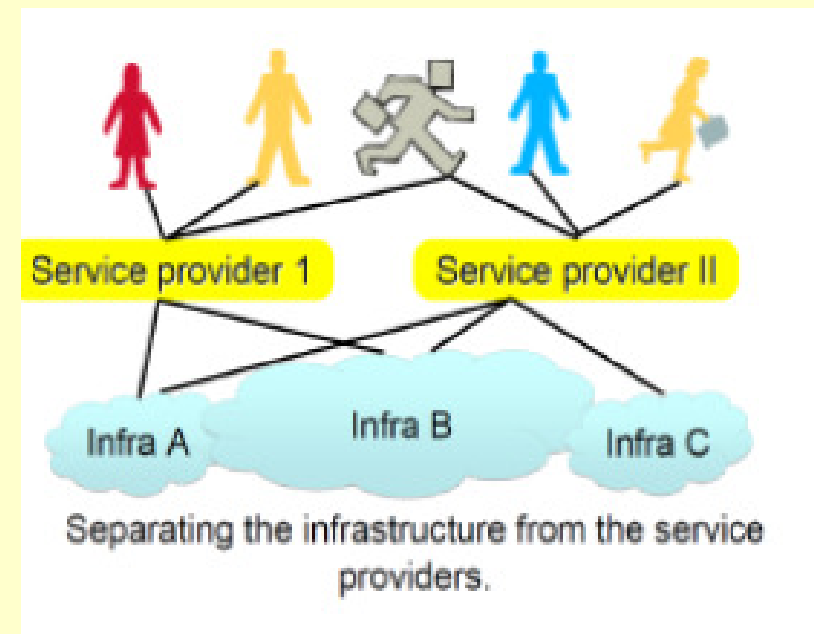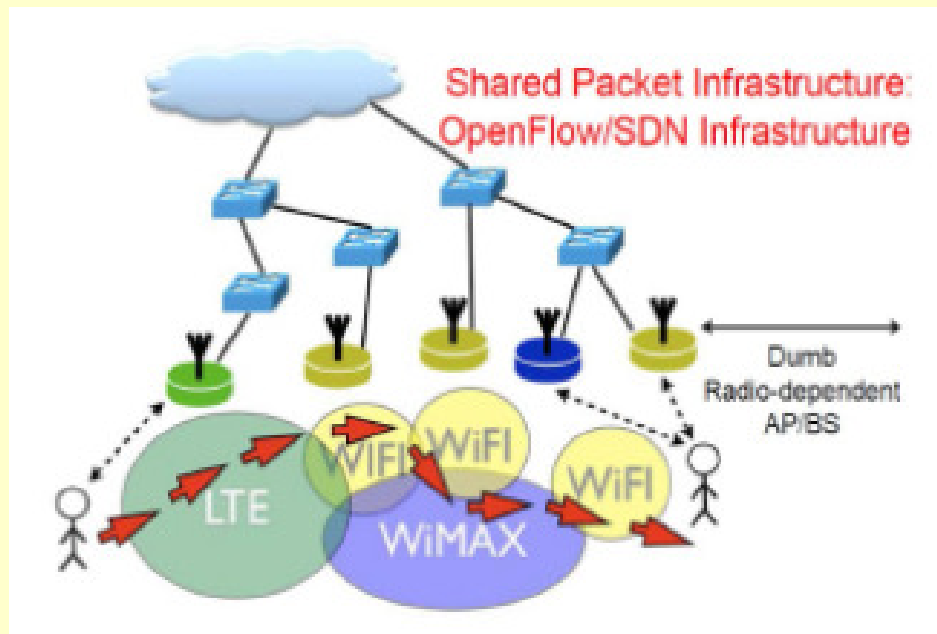
# 2. SDN Applications

- **Infrastructure-based Wireless Access Networks**

- **OpenRoads project [21]**
  -

  - ## Goal: Provide efficient wireless network services with SDN
    - users move across different wireless infrastructures, managed by various providers.
    - SDN-based architecture, backwards-compatible, yet open and sharable between different SPs
    - testbed using OF-enabled wireless devices such as
      - WiFi APs and
      - WiMAX base stations
    - controlled by NOX and Flowvisor controllers

    - Result: improved performance on handover events.
- Current work is done for: specific requirements and challenges in deploying a software-defined cellular network.

# 2. SDN Applications

- **Infrastructure-based Wireless Access Networks**
- **OpenRoads project [21] (cont'd)**
  - Goals ( cont'd)
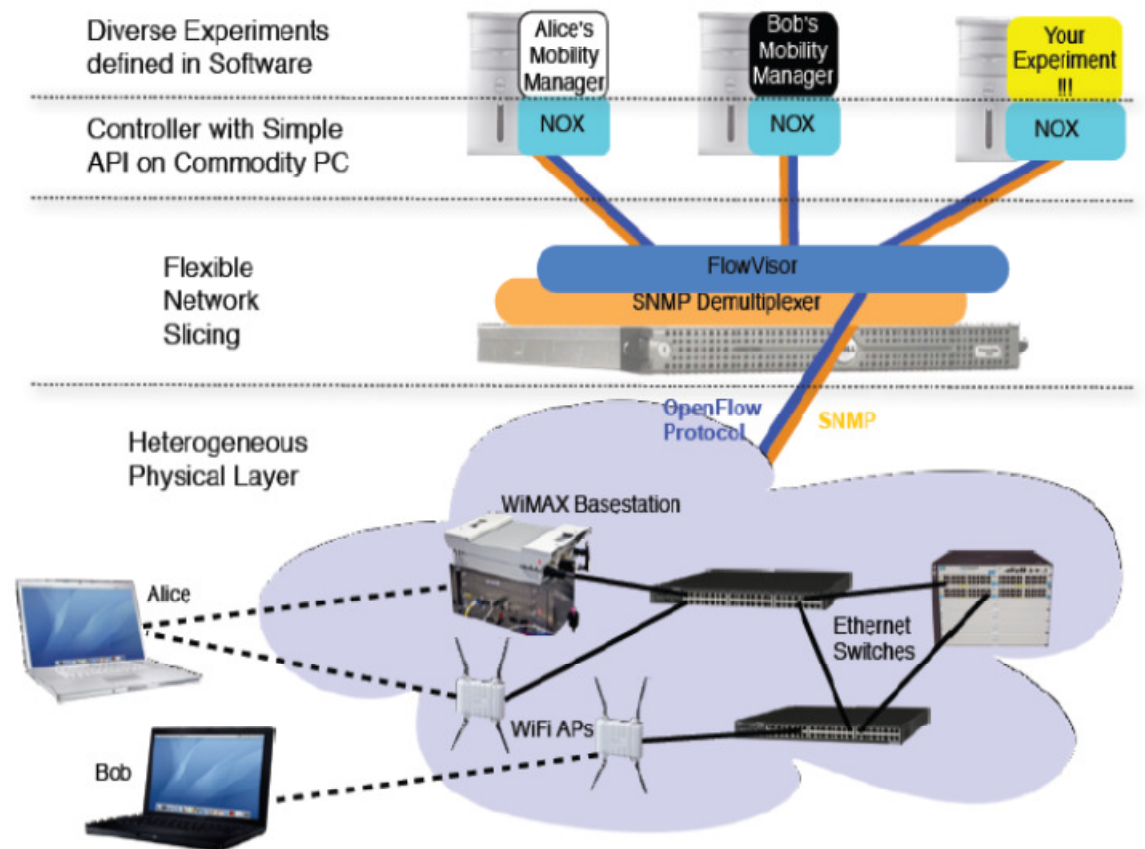
# 2. SDN Applications

- **Infrastructure-based Wireless Access Networks**
- **OpenRoads project [21] (cont'd)**

Overall scenario Use SNMP protocol to check the availability of Aps

Each network controller operates different network applications

Each user can share APs, but they can see different network

Network virtualization
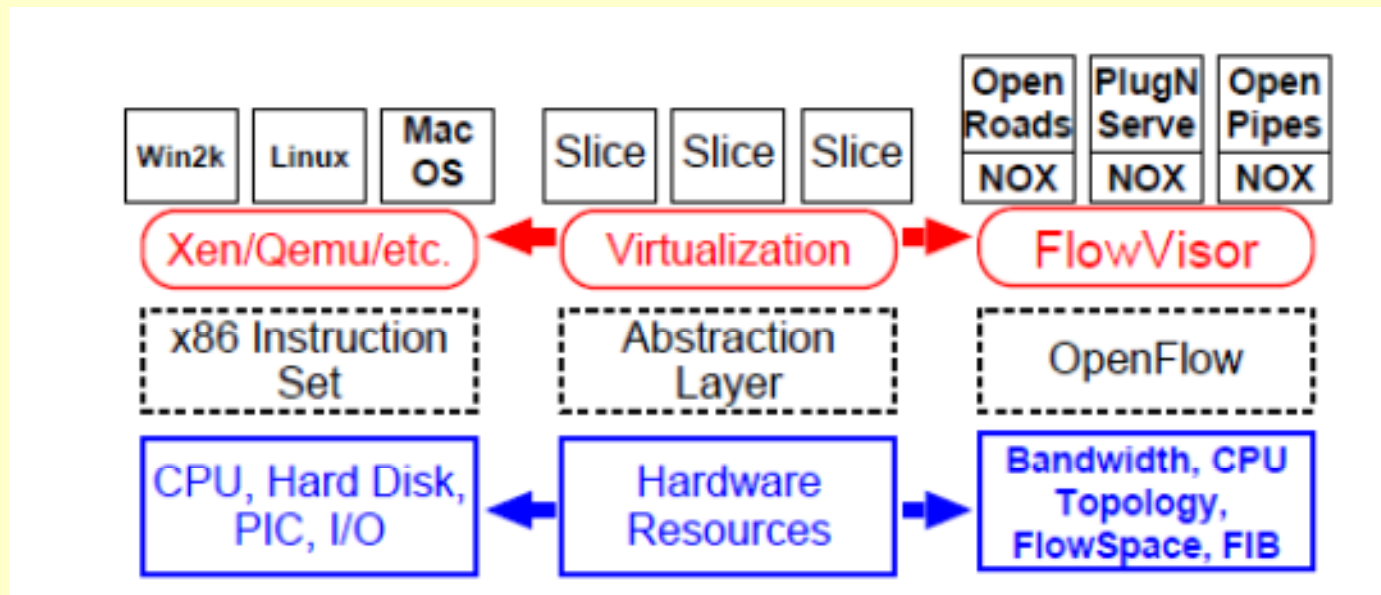
# 2. SDN Applications

- **Complements**
- **Flowvisor**
- originally developed at Stanford University
- used in experimental Research & Education networks to support **slicing**
- multiple experimenters get their own isolated slice of the infrastructure
  - and **control it using their own network OS** and a set of control and management applications.

- **FlowVisor is a special purpose OpenFlow controller**
  - **acting as a transparent proxy between OpenFlow switches and multiple other OpenFlow controllers**
- **creates rich "slices" of network resources and delegates control of each slice to a different controller**

  - **Slices can be defined by any combination of switch ports (layer 1), src/dst ethernet address or type (layer 2), src/dst IP address or type (layer 3), and src/dst TCP/UDP port or ICMP code/type (layer 4).**

- **FlowVisor enforces isolation between each slice, i.e., one slice cannot control another's traffic**

# 2. SDN Applications

- **Complements**
- **Flowvisor**
- Similar to computer virtualization, **FlowVisor is a network virtualization** layer that resides between the hardware and software architectural components.

- OpenRoads, PlugNServe, and OpenPipes are examples of virtual network controllers built on NOX(x 6).



**UPB Master Courses 2015-2016**

# 2. SDN Applications

- ## **Infrastructure-based Wireless Access Networks**

  - **Odin[22] : programmability in enterprise wireless LAN environments.**
    - - it builds an Access Point (AP) abstraction at controller level,
      - separating the association state from the physical AP
      - enabling proactive mobility management and load balancing without changes to the client.

  - **OpenRadio[23] : programmable wireless data plane**

    - flexibility at the PHY and MAC layers
    - provide modular I/Fs able to process traffic subsets using WiFi, WiMAX, 3GPP LTE-Advanced, etc.

    - Separation of the decision and forwarding planes allows:
      - an operator may express decision plane rules and corresponding actions
      - assembled from processing plane modules (e.g., FFT, decoding, etc.)

# 2. SDN Applications

- ## Service Provider -SDN Approach
  - There is a strong interest of SP/NP for SDN developments



**Service exposure**
Northbound APIs to allow networks to respond dynamically to application/service requirements

**Orchestrated network and cloud management**
Unified legacy and advanced network, cloud management system and OSS/BSS to implement SDN in step-by-step upgrade

**Integrated network control**
Control of entire network from radio to edge to core to data center for superior performance

**Service Provider SDN**

**Service provider needs:**
- Accelerated service innovation
- Advanced public/hybrid enterprise and consumer cloud services
- Improved QoE
- Opex reduction through centralized management
- Capex control

*Source: SDN: the service provider perspective, Ericsson Review, February 21, 2013*

**UPB Master Courses 2015-2016**
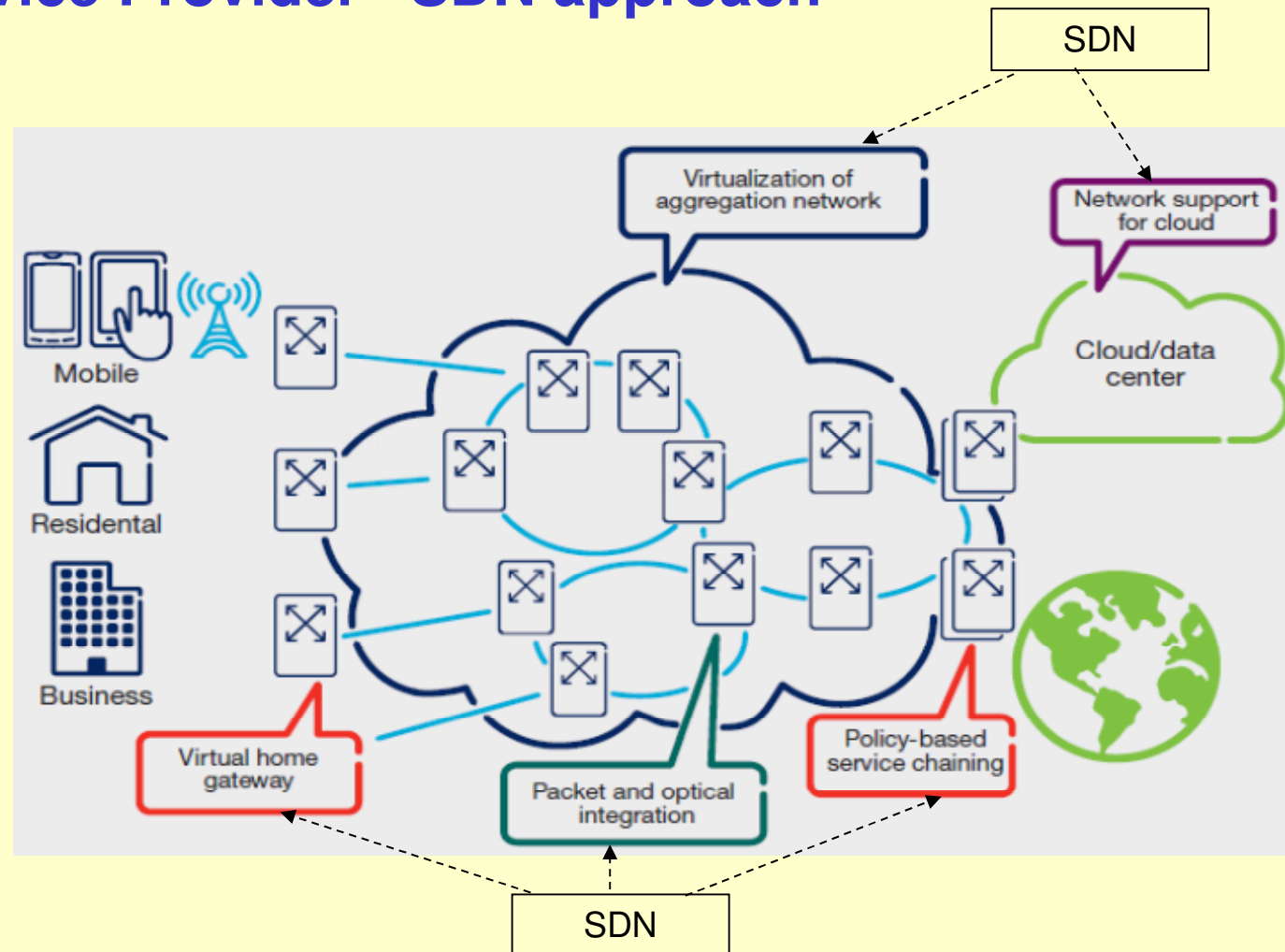
# 2. SDN Applications

- **Service Provider –SDN approach**



Source: *SDN: the service provider perspective, Ericsson Review, February 21, 2013*

# 2. SDN Applications

- **Service Provider -SDN Approach**

- **1. Aggregation/acces and mobile-backhaul networks (AAN)**
  - large number of nodes and relatively static tunnels – for traffic grooming of many flows
  - stringent requirements : reliability and short recovery times.
  - Generic backhaul solution technologies : L2, IP, IP/MPLS
  - Usually AAN is configured statically : - centralized management system + point of touch to *every* network element

  - **Building a centralized SDN controller is natural for backhaul solutions.**
    - Controller element (CE) can be hosted on a telecom-grade server or on an edge router
  - Operational difference (AAN- SDN network)- (AAN-traditional network) : the number of touch points required to provision and operate the domain
    - *SDN: only a few points are needed to control the connectivity for the entire network.*

# 2. SDN Applications

- **Service Provider –SDN approach**
- **Aggregation/acces  and mobile-backhaul networks  (AAN) (cont'd)**
  - **Example:**
  - *Current:* AAN: 10\*\*2- 10\*\*3 nodes ; distributed IGP, LDP/MPLS

  - *New:* SDN can simplify and increase the scalability of provisioning and operating by collecting the configuration of the whole network into just a few control points.
  - **Control Element  (CE) treats the underlying FE as remote line cards** of the same system and control them via  OpenFlow
- SDN: any kind of connectivity model is feasible (i.e. FE: L2, L3);  from a forwarding point of view, the same model is used

- **Network resilience**
  - CE pre-computes and pre-installs back-up routes and then protection switching is handled by the network elements for fast failover

  - or, CE can reroute around failures, in case multiple failures occur, or in scenarios having less stringent recovery requirements.

- From the outside**, the entire *network segment appears to be one big Provider Edge (PE) router***
  - neighbors  of the SDN-controlled area cannot tell the difference between it and a traditional network.

**UPB Master Courses 2015-2016**

# 2. SDN Applications

- **Service Provider –SDN approach**

- **2.Dynamic service-chaining** ( source: Ericsson)
  - Usually : for inline services (DPI, firewalls (FWs), NAT, load balancers, etc.), special middle-boxes are used hosted on HW/VMs.
  - *Service chaining =* forwarding a traffic flow traffic through ≥ one service.

  - Today: static or non-flexible solutions
    - no protocols or tools to perform flexible, **dynamic traffic steering**

  - **Dynamic service-chaining**
    - special services use – optimization: **selectively steering traffic through specific services or bypassing them** (avoid over-dimensioning-> capex savings)

    - operators can offer : virus scanning, firewalls, content filters through an automatic selection and subscribe portal, etc.

## Service Provider –SDN approach

- **Dynamic service-chaining (cont'd)**

- SDN - can support **dynamic service chaining** ( e.g. Ericsson - proof of concepts)
- logically centralized OF-CE manage switches and middleboxes
  - service chains can be differentiated on subscriber behavior, application, traditional 5-tuple, required service

  - Service paths are *unidirectional-* can be different for upstream and downstream traffic.
  - Traffic steering has two phases
    - 1. Classification of the incoming packets; assignment of a service path to them, based on predefined policies.
    - 2. Packets are then forwarded to the next service, based on the current position in their assigned service path
  - No repeating classification is required; hence the solution is scalable.

- The SDN CE can flexibly set up and reconfigure service chains

- The dynamic reconfiguration of service chains needs a mechanism to handle notifications sent from middle-boxes to the controller.
  - -e.g. , the DPI engine notifies CE ( via extended OF) that it has recognized a video flow.

# 2. SDN Applications

- **Service Provider - SDN approach**
- **Dynamic service-chaining (cont'd)**

- **Use Case for Service Chaining: Virtual Network System (VNS) -** concept ( source: Ericsson)
    - VNS: domain of the network with centralized CPl (this excludes some traditional control agents)
        - API- OpenFlow, controls the forwarders
        - VNS can create north-bound I/Fs APIs to support creation of new features, e.g. service chaining

    - The services provided may reside on devices located in different parts of the network, or within an edge router –e.g. Ericsson's *Smart Services Router (SSR)*

    - Service chains are programmed (cf. operator policies) based on a combination of information from the different layers (L2-L4. ..)
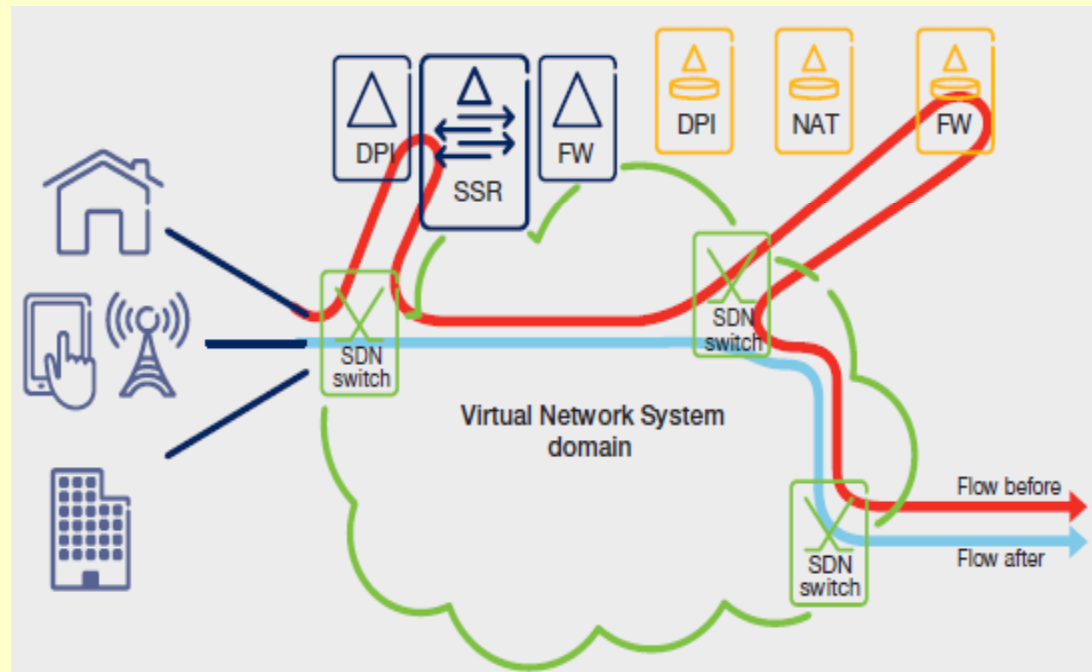
# 2. SDN Applications

- **Service Provider -SDN approach**
- **Dynamic service-chaining (cont'd)** - example
  - **Traffic goes first thrugh DPI + FW**
  - **After flow type has been determined (Deep Packet Inspection- DPI) the operator may decide to modify the services applied to it.**

**Example:  internet video stream flow**

-it may no longer need to pass the FW service after the service type has been detected

- the sub-sequent packets of the same flow may no longer need to pass the DPI service either ( shorter- blue path)



Source: *SDN: the service provider perspective, Ericsson Review, February 21, 2013*

**UPB Master Courses 2015-2016**
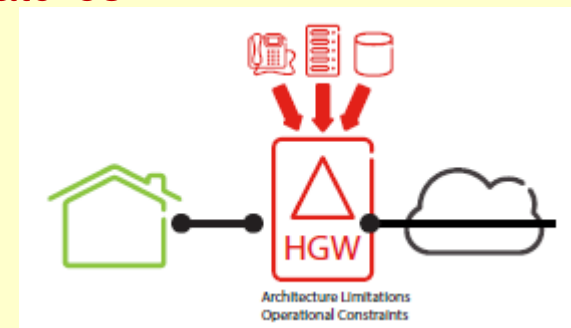
# 2. SDN Applications

- **Service Provider -SDN Approach**

- **3. Packet-optical (P-O) integration**

  - SDN : can solve some *optical packet networking challenges*
    - simplify multi-layer coordination and
    - optimize resource allocation at each layer
    - by redirecting, based on the specific traffic requirements and the best serving layer.

  - Today: layered set of separated media coordinated in a static manner

  - **SDN :** P-O infrastructure can be more fluid, with a unified recovery approach and an allocation scheme based on real-time link utilization and traffic composition

  - ONF still has to adapt OpenFlow to cope with optical constraints.

  - **Example of a a *hybrid architecture* proposal**
    - **OpenFlow drives the packet domain**
    - **GMPLS still controls the optical domain**

    - **advantage:**
      - one still utilizes the extensive optical capabilities of GMPLS
      - instead working to extend OF with optical capabilities, it allows focus on the actual integration of optical and packet domains applications that utilize the flexibility of a unified SDN controller.

**UPB Master Courses 2015-2016**

# 2. SDN Applications

- **4. *Virtual Home Gateway (VHG) /Virtual CPE***

- The customer premises equipment (CPE)/Home Gateway (HGW) in a residence
  - is (traditionally) the device that connects the home LAN to the broadband operator's WAN for Internet and media connectivity
  - The CPE began as a simple device, and the average subscriber connected few devices to it

  - CPE quickly evolved to become complex and feature-rich
    - It is connection point for many devices
    - using many different interfaces and protocols.

  - Increasing complexity in the CPEs led to increasing operational costs for the broadband operator
    - in the form of costs associated with device upgrade or replacement, network upgrades, and support for new features.

Traditional HGW location in relation to the broadband operator

# 2. SDN Applications

- **4.Virtual Home Gateway (VHG)/Virtual CPE** concept : new home-network architecture improving service delivery and management.

  - SDN can be used between the Residential Gateway (RG) and the edge network – moving most GW functionalities into an embedded execution environment.

  - **RG Virtualization**
    - reduces its complexity
    - provides greater granularity in remote-control management, extensible to every home device
    - increases the RG life, cutting maintenance costs, accelerating time to market for new services

  - **VHG can offer/provide**
    - seamless and secure remote profile instantiation extending the boundaries of a home network without compromising security
    - tools to configure and reconfigure middle-boxes dynamically,
    - specific connectivity requirements for a third-party service
    - enables operators to offer personalized applications to sub-scribers.

- **The architecture places an *operator-controlled bridge* (e.g. Ericsson) at the customer's premises instead of a complex router,**
  - the L3-L7 functionalities migrate to the IP edge or into the operator cloud

- Using SDN between the IP edge and the switch → fine-grained control for switch dynamic configuration

**UPB Master Courses 2015-2016**

# 2. SDN Applications

- **Home gateway control**

- ***4.Virtual Home Gateway (VHG)/Virtual CPE***



What users may want from a broadband operator.

The HGW possible functions:
:
• Monitor and manage the home environment including utilities

• Ensure residential/business perimeter and fire/gas safety

• Monitor and manage personal health and fitness

• Connect with others with voice and video over IP

• Interact with professionals and healthcare providers over secure video

• Play interactive HD games

•
Enforce policy rules and parental controls for a household by user

•
Access work resources

# 2. SDN Applications

- *4.Virtual Home Gateway (VHG)/Virtual CPE*

- A Virtual CPE is the combination of
    - a bridge installed in the residence and
    - a remotely-hosted Virtual CPE for that residence that connects to machines in the Cloud.

- The Virtual CPE contains the subscribers' profiles and account information

- The bridge sends all traffic to the broadband operator's WAN for processing

- Depending on where the Virtual CPE is placed within the WAN
    - (access node, Broadband Network Gateway, or Cloud with Service Chaining), processing occurs, and the correct services can be accessed.

Location of Virtual CPE in
a residential scenario



**UPB Master Courses 2015-2016**
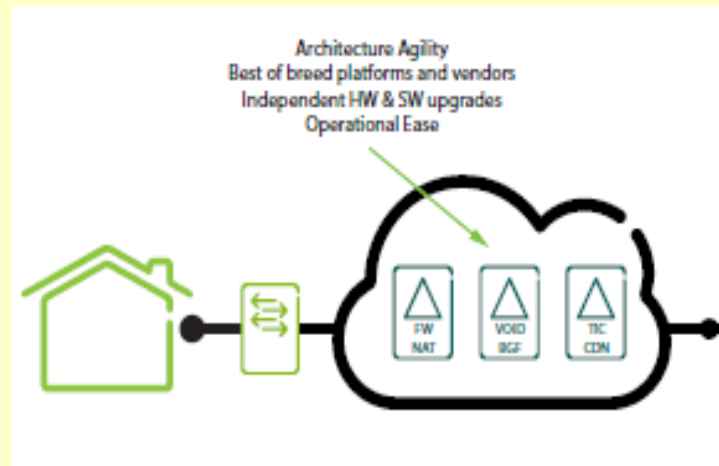
- *4.Virtual Home Gateway (VHG)/Virtual CPE*

- When a device at a subscriber's home connects to the Internet,
    - the Virtual CPE allows or rejects the connection request based on device type or other parameters, such as an individual user.

- With this model many residences can be hosted on a single machine, and the operator can:
    - Aggregate and load balance subscriber traffic
    - Use a Flow Control device for SDN
    - Upgrade the software or hardware independent of the subscriber
    - Granularly monitor service usage to generate service- specific revenue

- The virtual CPE provides a path for extending cloud-based services to a subscriber base.

Location of Virtual CPE in
a residential scenario

**UPB Master Courses 2015-2016**

# 2. SDN Applications

- **4.Virtual Home Gateway (VHG)/Virtual CPE**

- Virtual CPE with dynamic service chaining ( source Ericsson)
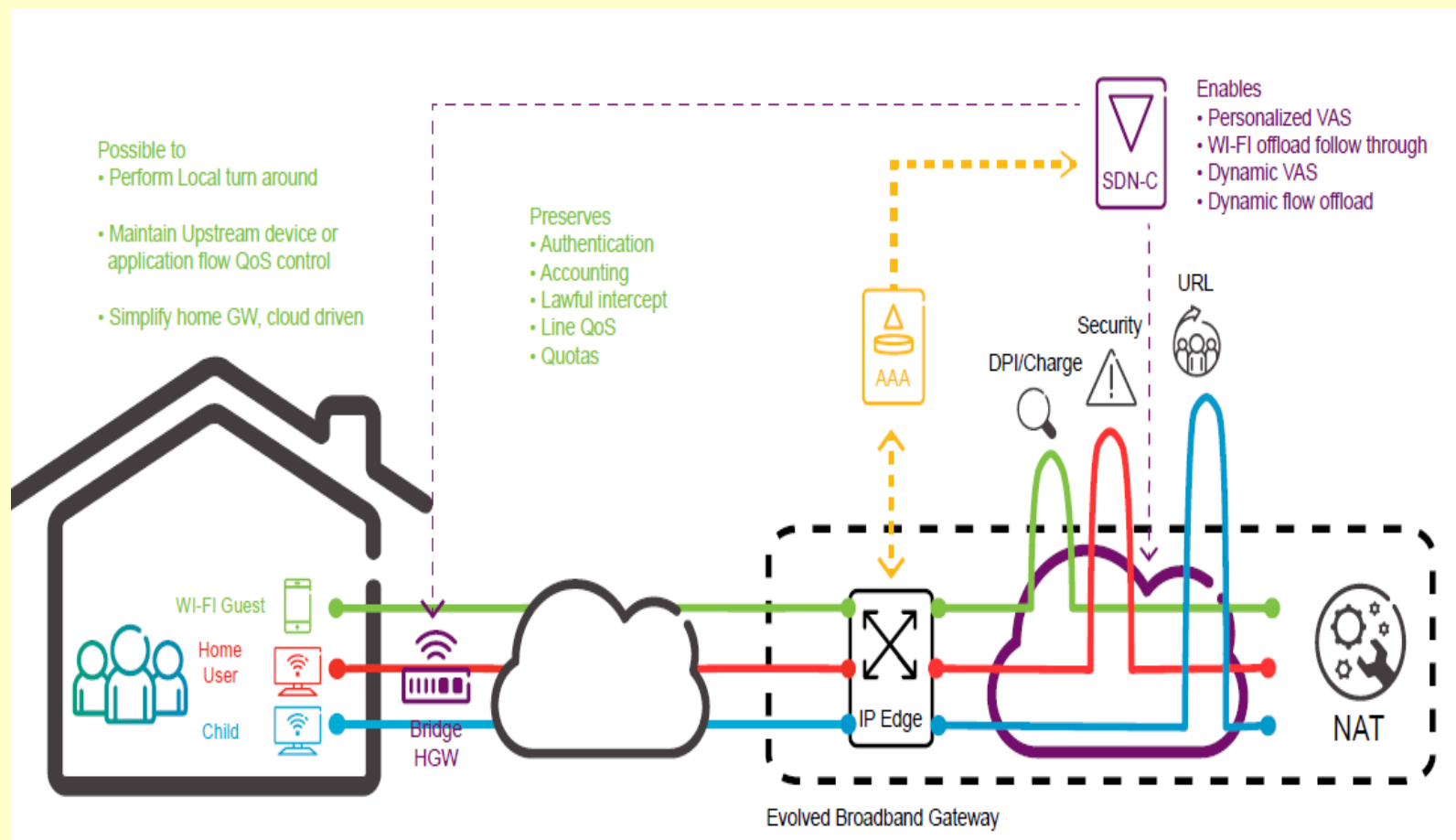
# 2. SDN Applications

- **5.Home Networks (HN) and Small Business**

- **Several projects examples** :
  - SDN used in smaller networks, (home or small businesses)
  - need for more careful network management and tighter security
  - avoid complex admin at each home/business

  - **Managing HNs** [25] by making the **network gateway/controller to act as a HN Data Recorder** (e.g. logs for troubleshooting or other purposes).

  - **Outsourcing management to third-party experts** [26]: remote control of programmable switches and the appl. of distributed network monitoring and inference algorithms used to detect security problems.

  - **Alternative approach** [27]: a HN can be managed by the users who better understand the dynamics and needs of their environment.
  - SDN can provide users a view into their network (single point of control)

  - **Anomaly Detection System (ADS)** [28] in a programmable HN can accurate identify malicious activity as compared to one deployed at the ISP.
    - The ADS algorithm could operate alongside other controller services, such as a HomeOS that may react to suspicious activity and report anomalies to the ISP or local administrator.

# 2. SDN Applications

## 5. Bandwidth on Demand ( BWoD)

- WAN bandwidth demand ratio *peak info rate/mean rate* ~ 10 to 20 (cloud networking, ad hoc inter-enterprise collaboration, etc.)
  - peaks duration – very variable: 1 hour or less to several weeks or more.
- Contracting *Peak Information Rate* ( PIR) is costly and wasteful.

- *Bandwidth on Demand (BWoD)* – dynamically adjustable if wanted ( pay what you consume)
  - Connection types: subscribers; subscriber to a service GW (e.g., a cloud data center); subscriber to a third-party interconnect point.

- Current model of BWoD services (limited number of operators)
  - Lack of automation → difficult to roll out self-provisioned services and respond to time-sensitive changes in bandwidth requirements.
  - customers are given some control; they invoke the services through a portal but very limited in scope.
  - Frequent changes in a distributed control environment sometimes lead to transient overloads → congestion and instability.
  - Lack of a standard I/F → operators today must interface their OSS/BSS systems to a vendor-specific network infrastructure. (need to redesign control applications for each vendor)

**UPB Master Courses 2015-2016**

- **Bandwidth on Demand ( BWoD)**

- **SDN Solution :** BWoD from an OF - SDN architecture with a programmatic north API
  - operators have centralized, granular control over the networking infrastructure.

- Customers can *automatically request dynamic changes to bandwidth allocation* and other QoS parameters at the packet and/or optical layers, either immediately or scheduled in the future.

- The SDN control layer can leverage topology-aware path computation to cost-effectively enable bandwidth on demand.

- SDN : real-time topological view of the network,
  - enables network virtualization, and
  - allows *network bandwidth reservation* to provide guaranteed *performance on a per-connection or flow basis to meet SLA requirements.*

**UPB Master Courses 2015-2016**

# 2. SDN Applications

- **Bandwidth on Demand (BWoD)**

- SDN's global view of network resource supply and customer demands → intelligent and dynamic BWoD pricing.

  - An *analytics engine* could evaluate current supply and demand as well as historical temporal demand peaks and supply.

  - Through continual learning of the price elasticity of demand, these adjustments can become more refined, enabling the analytics engine to maximize network revenue per bit.

- Network virtualization allows operators to leverage the same networking and operational infrastructure on which they deliver traditional services to create BWoD services and new billing models.

# 2. SDN Applications
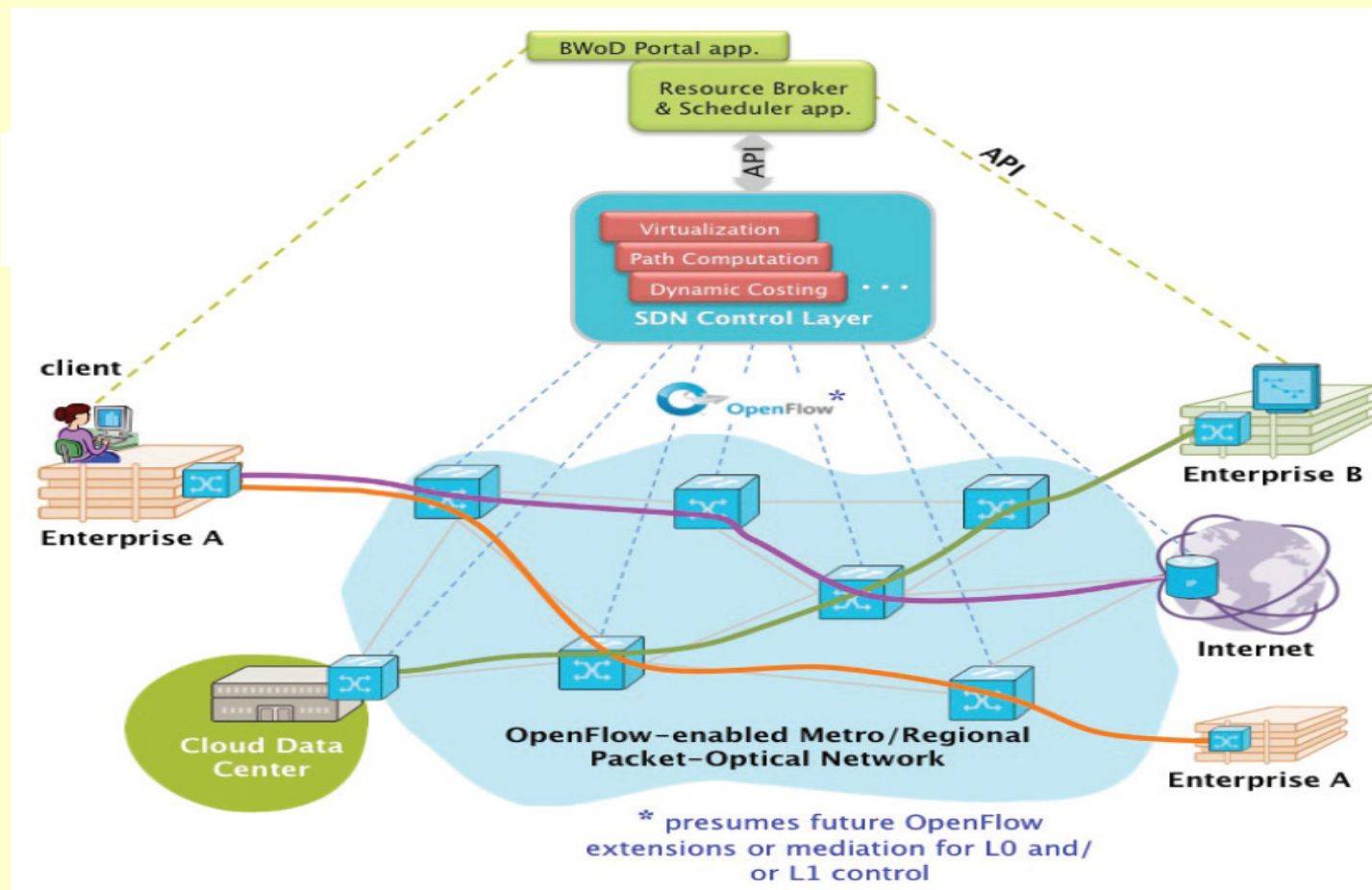
- **Bandwidth on Demand (BWoD)**



*Source: Operator Network Monetization Through OpenFlow-Enabled SDN, ONF Solution Brief,*
*April 3, 2013, https://www.opennetworking.org/images/stories/downloads/sdn-*
*resources/solution-briefs/sb-network-monetization.pdf*

**UPB Master Courses 2015-2016**

# CONTENTS

1. Software Defined Networks (Basics)
2. SDN Applications
3. ☞ **SDN-OpenFlow**
4. SDN Extensions and Advanced Architectures
5. Other recent technologies- SDN approach
6. Conclusions

- **OpenFlow switch ( forwarder)- architecture**
- *Source: OpenFlow Switch Specification, V1.3.0 and V 1.4.0 (Wire Protocol 0x05 )Oct. 2014 ( only for Ethernet)*

- **ONF Key Definitions for OF switch (partial list)**

- **Action:** an operation that forwards the packet to a port, modifies the packet (such as decrementing the TTL  field) or change its state (such as associating it with a queue).

    - Actions may be specified as part of the instruction set associated with a flow entry or in an action bucket associated with a group entry.
    - **Actions may be accumulated in the Action Set of the packet or applied immediately to the packet**

- **Action Bucket:** a set of actions and associated parameters in a group. The group will select one (or more) buckets for each packet.

- **Action Set:** a set of actions associated with the packet that are accumulated while the packet is processed by each table and that are executed when the instruction set instructs the packet to exit the processing pipeline

- **Control Channel:** The aggregation of components of an OF logical switch that manage communication with controllers.
    - The control channel includes one OF channel per OF controller.

# 3. SDN-OpenFlow

- **ONF Definitions for OF switch (partial list)**

- **Counter:** counts the number of packets and bytes at various specific points of the pipeline, such as on a port or on a flow entry

- **Flow: a sequence of packets having one or more fields identical in various headers**
  - **( do not associate a flow with a route!)**

- **Flow Entry:** an element in a flow table used to match and process packets.
  - It contains
    - a set of match fields for matching packets,
    - a priority for matching precedence,
    - a set of counters to track packets,
    - a set of instructions to apply

- **Flow Table:** a stage of the pipeline. It contains flow entries.

- **Group:** a list of action buckets and some means of choosing one or more of those buckets to apply on a per-packet basis.

- **Instruction:** are attached to a flow entry and describe the OF processing that happens when a packet matches the flow entry.
  - An instruction
    - either modifies pipeline processing, such as directing the packet to another flow table,
    - or contains a set of actions to add to the action set,
    - or contains a list of actions to apply immediately to the packet.

**UPB Master Courses 2015-2016**

# 3. SDN-OpenFlow

- **ONF Definitions for OF switch (partial list)** (cont'd)

- **Match Field**:a field against which a packet is matched, including packet headers, the ingress port, and the metadata value. A match field may be wildcarded (match any value) and in some cases bitmasked.

- **Matching:** comparing the set of header fields and pipeline fields of a packet to the match fields of a flow entry

- **Metadata:** a maskable register value that is used to carry information from one table to the next.

- **Message:** OF protocol unit sent over an OF connection. May be a **request**, a **reply**, a **control message** or a **status event.**

- **Meter:** an element that can measure and control the **rate of packets**. The meter can trigger a meter band if the packet rate or byte rate passing through the meter exceeds a predefined threshold.
  - **If the meter band drops the packet – it is called Rate Limiter**

- **OF Logical Switch**: A set of OF resources that can be managed as a single entity;  includes a data path and a control channel.

# 3. SDN-OpenFlow

- **ONF Definitions for OF switch (partial list)**

- **Packet:** an Ethernet frame, including header and payload.

- **Pipeline:** the set of **linked flow tables**
  - that provide **matching, forwarding, and packet modification** in an OF switch

- **Port:** point where packets enter and exit the OpenFlow pipeline May be a **physical port**, a **logical port** defined by the switch, or a reserved port defined by the OpenFlow protocol.

- **Pipeline fields**:
  - set of values attached to the packet during pipeline processing
  - They are not header fields; include the ingress port, the metadata value and the Tunnel-ID value.

- **Queue:** Logical Queue to schedule packets according to their priority on an output port to provide Quality-of-Service (QoS).

- **Tag:** a **header that can be inserted or removed from a packet** via push and pop actions.

- **Outermost Tag**: the tag that appears closest to the beginning of a packet.

# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions**
  - **"On demand" style of flow processing**
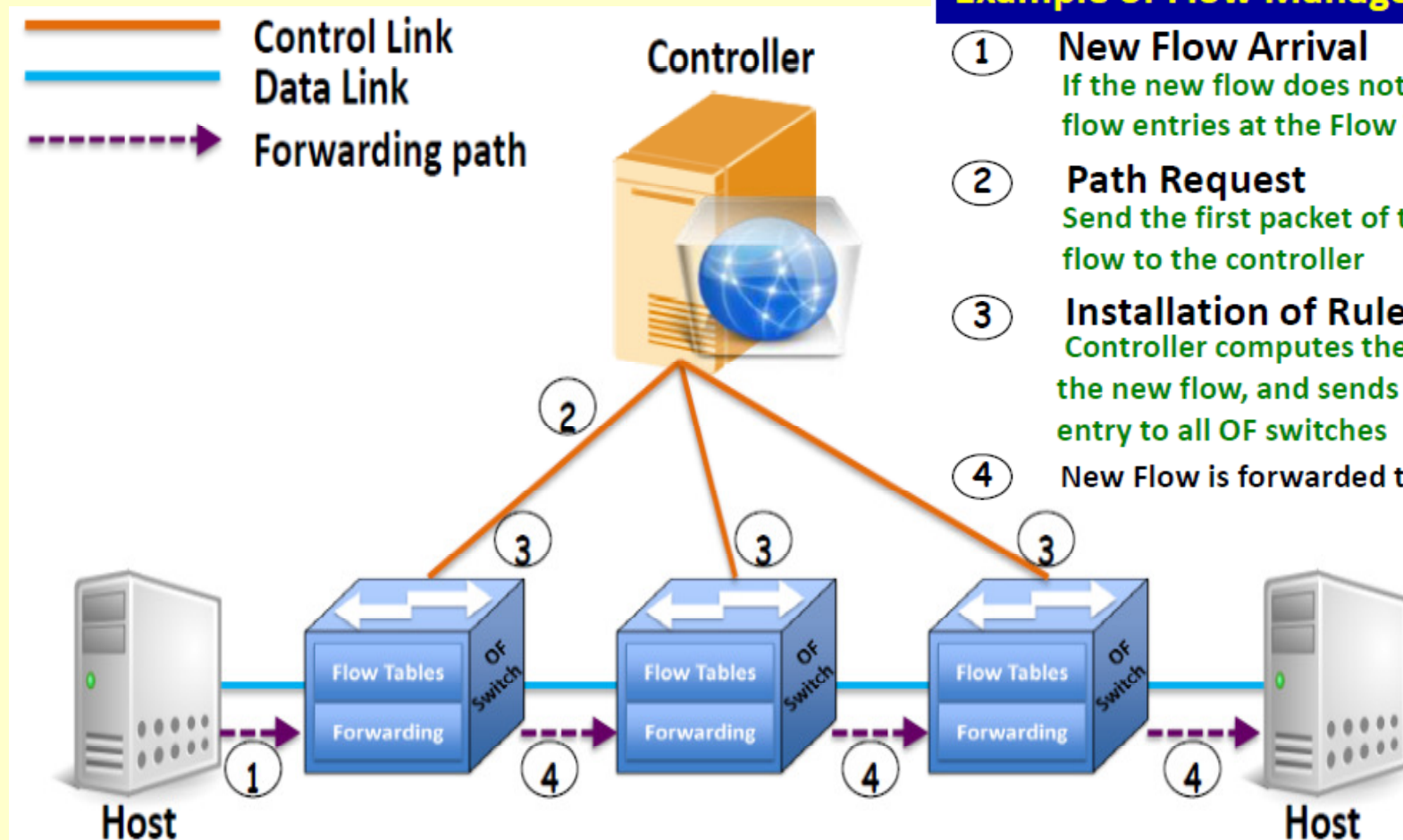


**Example of Flow Management**

① **New Flow Arrival**
If the new flow does not match the flow entries at the Flow Tables

② **Path Request**
Send the first packet of the new flow to the controller

③ **Installation of Rules**
Controller computes the path for the new flow, and sends the flow entry to all OF switches

④ New Flow is forwarded to destination

**Control Link**
**Data Link**
**Forwarding path**

Controller

Flow Tables / Forwarding / OF Switch

Host

# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions**

  - OpenFlow (OF) protocol runs over the *Secure Sockets Layer* (SSL).
  - Each switch is connected to other OF switches and, possibly, to end-user devices that are the sources and destinations of packet flows.

  - In a switch, a **series of tables—implemented in HW or firmware**—are used to manage the flows of packets through the switch.

  - **The switch**
    - encapsulates and forwards the first packet of a flow to an SDN controller, which decides whether the **flow should be added to the switch flow table**.

    - It forwards incoming packets out the appropriate port based on the flow table.
      - The flow table may include priority information dictated by the controller

    - It can drop packets on a particular flow, temporarily or permanently, as dictated by the controller.
      - Packet dropping can be used for security purposes, to solve *Denial-of-Service* (DoS) attacks or traffic management requirements.

# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions**
- **An OF Switch contains :**
  - one or more *flow tables* and a **group table**, which perform packet lookups and forwarding,
  - and an *OF channel to an external controller*

- **The controller manages the switch via the OF protocol.**
  - The controller can add, update, and delete flow entries in flow tables, both reactively (in response to packets) and proactively.

- **Flow table** : a set of flow entries; each flow entry consists of **match fields, counters, and a set of instructions** to apply to matching packets
  - Matching starts at the first flow table and may continue to additional flow tables
  - *Flow entries match packets in priority order*, with the first matching entry in each table being used.
  - **If a *matching entry is found*, the instructions associated with the specific flow entry are executed.**
  - If no *match is found in a flow table,* the outcome depends on configuration of the **table-miss flow entry**:
    - for example, the packet may be forwarded to the controller over the OF channel, dropped, or may continue to the next flow table
  - A *Meter Table* can trigger a variety of performance-related actions on a flow

**UPB Master Courses 2015-2016**

# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions ( cont'd)**

- *Instructions* associated with each flow entry either contain actions or modify pipeline processing
- *Actions* included in instructions describe packet forwarding, packet modification and group table processing.
- *Pipeline processing instructions* allow packets to be sent to subsequent tables for further processing and allow information, in the form of metadata, **to be communicated between tables.**
    - Table pipeline processing stops when the instruction set associated with a matching flow entry does not specify a next table; at this point the packet is usually modified and forwarded

- *Flow entries may forward to a port* (physical port/logical/reserved port )
    - *Reserved ports* may specify **generic forwarding actions** such as sending to the controller, flooding, or forwarding using non-OF methods, such as "normal" switch processing
    - while *switch-defined logical ports* may specify link aggregation groups, tunnels or loopback interfaces

# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions ( cont'd)**

  - *Actions associated with flow entries* may also direct packets to a group, which specifies additional processing
  - A FT may direct a flow to a *Group Table,* which may trigger a variety of actions that **affect one or more flows**
  - The *group table* contains *group entries;*
    - each group entry contains a list of action buckets with specific semantics dependent on group type
    - The actions in one or more action buckets are applied to packets sent to the group.
  - *Groups* represent sets of actions for flooding, or more complex forwarding semantics (e.g. multipath, fast reroute, and link aggregation).
    - As a general layer of indirection, groups also enable multiple flow entries to forward to a single identifier (e.g. IP forwarding to a common next hop).
    - This abstraction allows common output actions across flow entries to be changed efficiently.

# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions ( cont'd)**

- Switch designers are **free to implement the internals in any way convenient,** provided that correct match and instruction semantics are preserved.
  - Ex. 1: while a flow entry may use an all group to forward to multiple ports, a switch designer may choose to implement this as a single bitmask within the hardware forwarding table.

  - Ex. 2: is matching; the pipeline exposed by an OF switch may be physically implemented with a different number of hardware tables.

Main components
of an OpenFlow switch
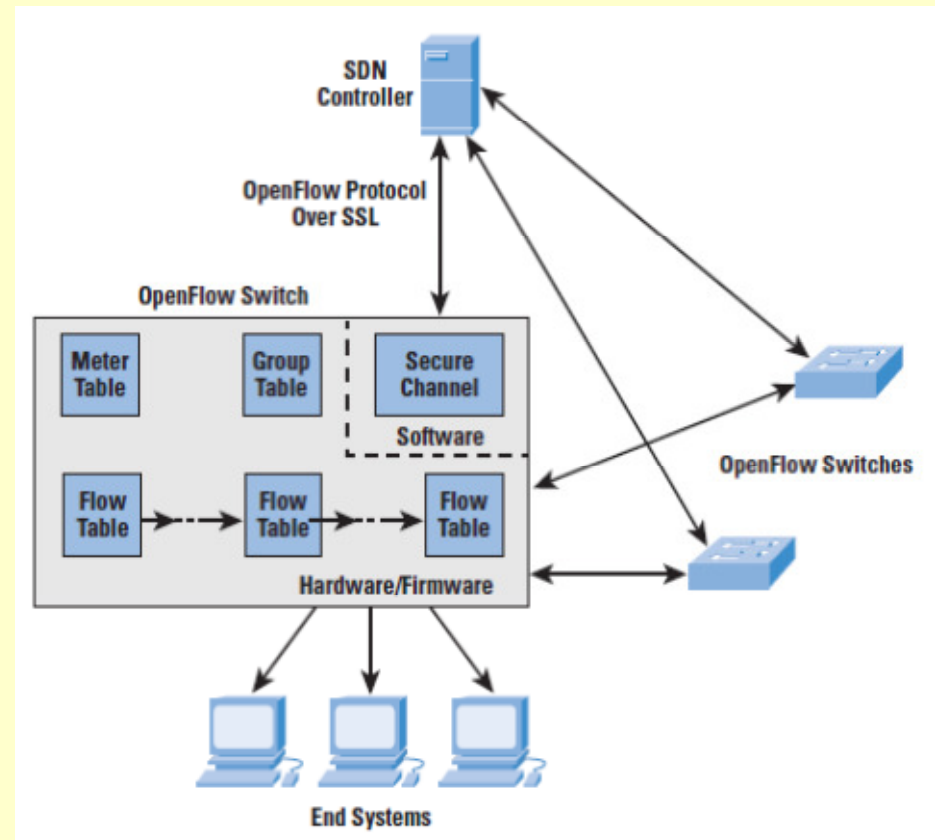
# 3. SDN-OpenFlow

- **OpenFlow switch components and basic functions ( cont'd)**
- **A flow table consists of flow entries.**
- Each flow table entry contains:
    - *match fields*: to match against packets. These consist of the ingress port and packet headers and optionally metadata specfied by a previous table.
    - *priority:* matching precedence of the flow entry.

    - *counters:* updated when packets are matched.
        - Examples include the number of received bytes and packets per port, per flow table, and per flow-table entry; number of dropped packets; and duration of a flow.
    - *instructions:* to modify the action set or pipeline processing.
    - *timeouts:* maximum amount of time or idle time before flow is expired by the switch.

- **OpenFlow switch components and basic functions ( cont'd)**
- Each flow table entry contains:

  - *cookie:* opaque data value chosen by the controller.
    - May be used by the controller to filter flow statistics, flow modification and flow deletion. Not used when processing packets.

  - A flow table may include a ***table-miss* flow entry,** which renders all Match Fields wildcards (every field is a match regardless of value) and has the lowest priority (priority 0).

    - **Example: Main components of a flow entry in a flow table**.

| Match Fields | Priority | Counters | Instructions | Timeouts | Cookie |
|---|---|---|---|---|---|

# 3. SDN-OpenFlow

## Basic Flow table functionalities

① Find highest-priority matching flow entry

② Apply instructions:
i. Modify packet & update match fields
(apply actions instruction)

ii. Update action set (clear actions and/or
write actions instructions)

iii. Update metadata

③ **Send match data and action set to next table**
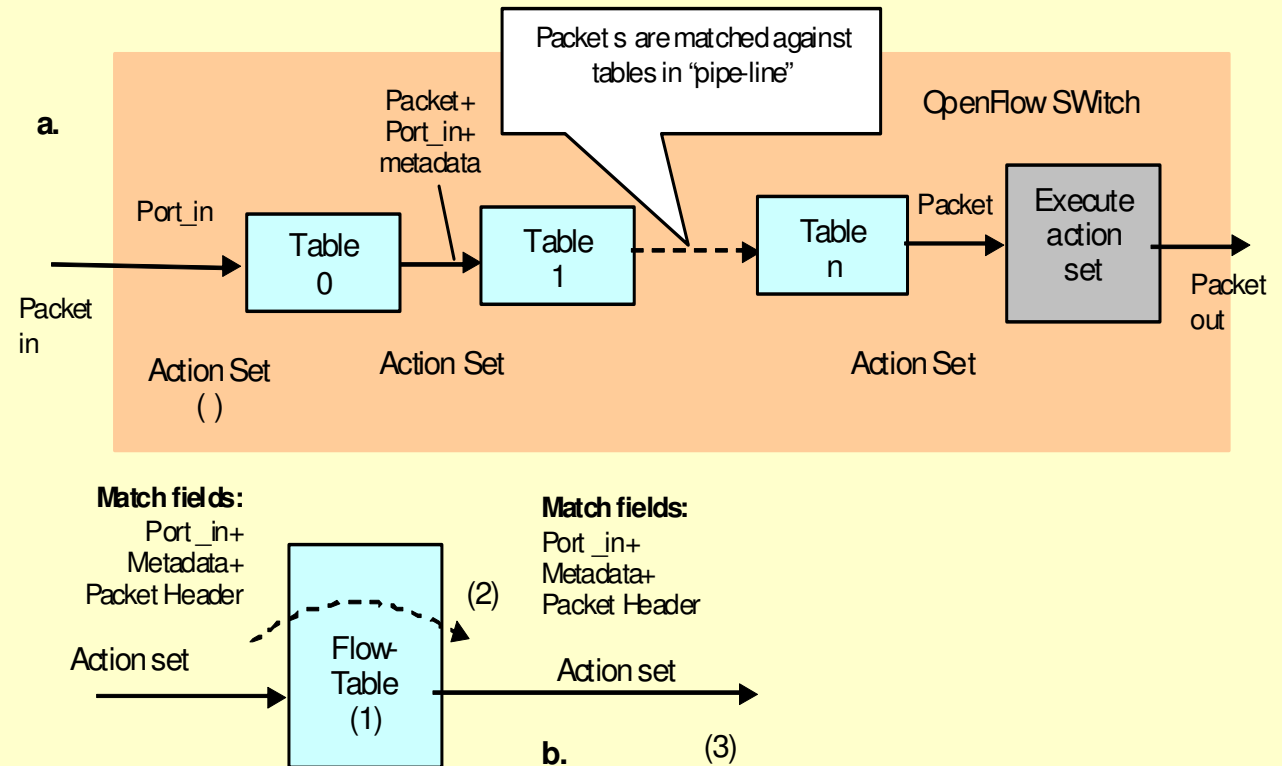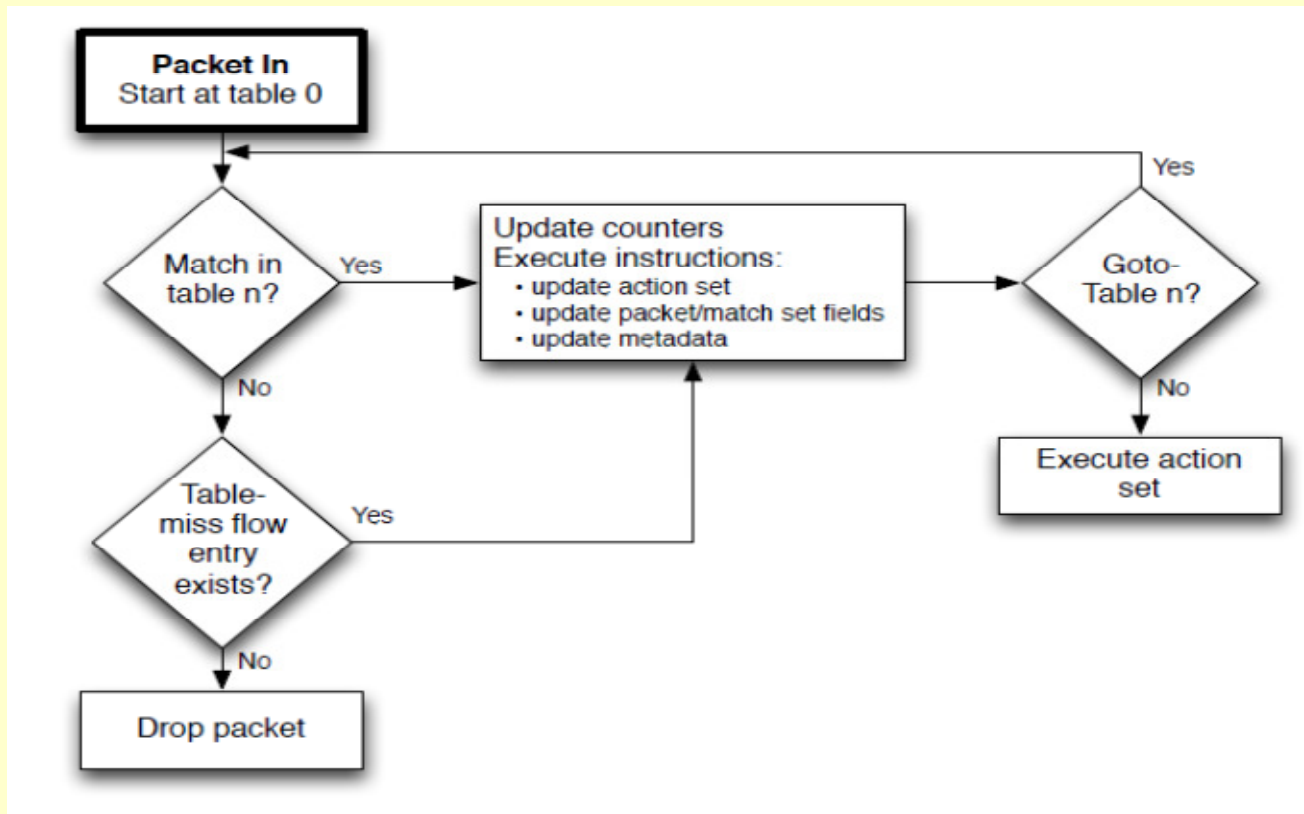


Figure : Packet flow through the processing pipeline.

# 3. SDN-OpenFlow

- **OpenFlow**

- Flowchart detailing **packet flow through an OpenFlow switch. ( v1.4.0)**
- The switch starts by performing a table lookup in the first flow table, and based on pipeline processing, may perform table lookups in other flow tables

- **OpenFlow**
- Alternative view of packet flow in the OF switch
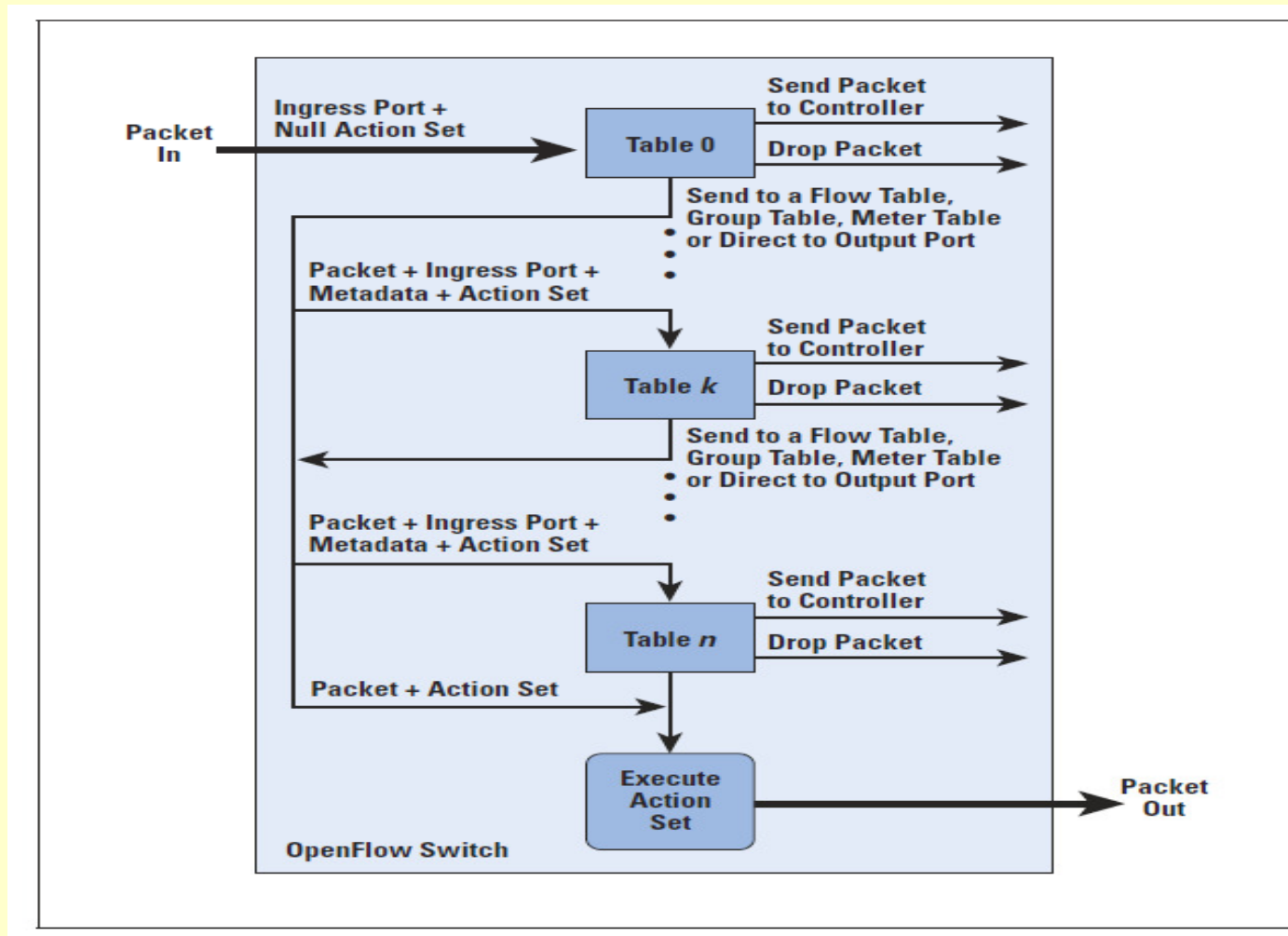
# 3. SDN-OpenFlow

- **OpenFlow : Flow Table components**
- The Match Fields component of a table entry consists of the following **required fields**:

- *Ingress Port:* The identifier of the port on the switch where the packet arrived. It may be a physical port or a switch-defined virtual port.
- *Ethernet Source and Destination Addresses*: Each entry can be an exact address, a bitmasked value for which only some of the address bits are checked, or a wildcard value (match any value).
- *IPv4 or IPv6 Protocol Number*: A protocol number value, indicating the next header in the packet.
- *IPv4 or IPv6 Source Address and Destination Address*: Each entry can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.

- *TCP Source and Destination Ports:* Exact match or wildcard value.

- *User Datagram Protocol (UDP) Source and Destination Ports:* Exact match or wildcard value.

| In Port | VLAN ID | Ethernet | | | IP | | | TCP | |
|---------|---------|----|----|------|----|----|-------|-----|-----|
| | | SA | DA | Type | SA | DA | Proto | Src | Dst |

Table 1: The header fields matched in a "Type 0" OpenFlow switch.

# 3. SDN-OpenFlow

- **OpenFlow : Flow Table components**
- **Optional fields**:

- *Physical Port:* Used to designate underlying physical port when packet is received on a logical port.

- *Metadata:* Additional information that can be passed from one table to another during the processing of a packet. Its use is discussed subsequently.

- *Ethernet Type:* Ethernet Type field.

- *VLAN ID and VLAN User Priority:* Fields in the IEEE 802.1Q Virtual LAN header.

- *IPv4 or IPv6 DS and ECN:* Differentiated Services and Explicit Congestion Notification fields.

- *Stream Control Transmission Protocol (SCTP) Source and Destination Ports:* Exact match or wildcard value.

- *Internet Control Message Protocol (ICMP) Type and Code Fields:* Exact match or wildcard value.

# 3. SDN-OpenFlow

- **OpenFlow : Flow Table components**
- **Optional fields**:

- *Address Resolution Protocol (ARP) Opcode:* Exact match in Ethernet Type field.

- *Source and Target IPv4 Addresses in Address Resolution Protocol (ARP) Payload:* Can be an exact address, a bitmasked value, a subnet mask value, or a wildcard value.

- *IPv6 Flow Label:* Exact match or wildcard.

- *ICMPv6 Type and Code fields:* Exact match or wildcard value.

- *IPv6 Neighbor Discovery Target Address:* In an IPv6 Neighbor Discovery message.

- *IPv6 Neighbor Discovery Source and Target Addresses:* Link-layer address options in an IPv6 Neighbor Discovery message.

- *Multiprotocol Label Switching (MPLS) Label Value, Traffic Class, and Bottom of Stack (BoS):* Fields in the top label of an MPLS label stack.

# 3. SDN-OpenFlow

- **OpenFlow : Flow Table components**
- The *instructions component* of a table entry consists of a set of instructions that are executed if the packet matches the entry.

- The OpenFlow specification includes the following actions:
  - *Output:* Forward packet to specified port.
  - *Set-Queue:* Sets the queue ID for a packet.
    - When the packet is forwarded to a port using the output action, the queue id determines which queue attached to this port is used for scheduling and forwarding the packet.
    - Forwarding behavior is dictated by the configuration of the queue and is used to provide basic QoS support.

  - *Group:* Process packet through specified group.

  - *Push-Tag/Pop-Tag:* Push or pop a tag field for a VLAN or MPLS packet.

  - *Set-Field:* The various Set-Field actions are identified by their field type; they modify the values of respective header fields in the packet.

  - *Change-TTL:* The various Change-TTL actions modify the values of the IPv4 Time To Live (TTL), IPv6 Hop Limit, or MPLS TTL in the packet.

# 3. SDN-OpenFlow

- **OpenFlow Protocol messages**
- ***Controller-to-Switch:*** initiated by the controller and, in some cases, require a response from the switch.
    - This class of messages enables the controller to manage the logical state of the switch, including its configuration and details of flow- and group-table entries.

    - Also included in this class is the *Packet-out* message.
        - This message is used when a switch sends a packet to the controller and the controller decides not to drop the packet but to direct it to a switch output port.

- ***Asynchronous:*** are sent without solicitation from the controller.
    - This class includes various status messages **to the controller**.
    - Also included is the *Packet-in* message, which may be used by the switch to send a packet to the controller when there is no flow-table match.

- ***Symmetric: sent without solicitation from either the controller or the switch.***

    - **Hello** messages are typically sent back and forth between the controller and switch when the connection is first established.

    - **Echo** request and reply messages can be used by either the switch or controller to measure the latency or bandwidth of a controller switch connection or just verify that the device is operating. The Experimenter message is used to stage features to be built into future versions of OpenFlow.

# 3. SDN-OpenFlow

- **OpenFlow Protocol messages**

| Message | Description |
|---|---|
| **Controller-to-Switch** | |
| Features | Request the capabilities of a switch. Switch responds with a features reply that specifies its capabilities. |
| Configuration | Set and query configuration parameters. Switch responds with parameter settings. |
| Modify-State | Add, delete, and modify flow/group entries and set switch port properties. |
| Read-State | Collect information from switch, such as current configuration, statistics, and capabilities. |
| Packet-out | Direct packet to a specified port on the switch. |
| Barrier | Barrier request/reply messages are used by the controller to ensure message dependencies have been met or to receive notifications for completed operations. |
| Role-Request | Set or query role of the OpenFlow channel. Useful when switch connects to multiple controllers. |
| Asynchronous-Configuration | Set filter on asynchronous messages or query that filter. Useful when switch connects to multiple controllers. |
| **Asynchronous** | |
| Packet-in | Transfer packet to controller. |
| Flow-Removed | Inform the controller about the removal of a flow entry from a flow table. |
| Port-Status | Inform the controller of a change on a port. |
| Error | Notify controller of error or problem condition. |
| **Symmetric** | |
| Hello | Exchanged between the switch and controller upon connection startup. |
| Echo | Echo request/reply messages can be sent from either the switch or the controller, and they must return an echo reply. |
| Experimenter | For additional functions. |

# 3. SDN-OpenFlow

- **OpenFlow**
- Available Software Switch Platforms
- SDN software switches
  - can be used to run a SDN testbed or when developing services over SDN.

| Software Switch | Implementation | Overview | Version |
|---|---|---|---|
| Open vSwitch [14] | C/Python | Open source software switch that aims to implement a switch platform in virtualized server environments. Supports standard management interfaces and enables programmatic extension and control of the forwarding functions. Can be ported into ASIC switches. | v1.0 |
| Pantou/OpenWRT [15] | C | Turns a commercial wireless router or Access Point into an OpenFlow-enabled switch. | v1.0 |
| ofsoftswitch13 [11] | C/C++ | OpenFlow 1.3 compatible user-space software switch implementation. | v1.3 |
| Indigo [6] | C | Open source OpenFlow implementation that runs on physical switches and uses the hardware features of Ethernet switch ASICs to run OpenFlow. | v1.0 |

Current software switch examples compliant with the OpenFlow standard

# 3. SDN-OpenFlow

**Examples of native SDN switches compliant with the OpenFlow standard**

| Provider | Switch Model | Version |
|----------|--------------|---------|
| HP | 8200zl, 6600, 6200zl, v1.0 5400zl, and 3500/3500yl | v1.0 |
| Brocade | NetIron CES 2000 Series | v1.0 |
| IBM | RackSwitch G8264 | v1.0 |
| NEC | PF5240 PF5820 | v1.0 |
| Pronto | 3290 and 3780 | v1.0 |
| Juniper | Junos MX-Series | v1.0 |
| Pica8 | P-3290, P-3295, P-3780 and P-3920 | v1.2 |

*Source [2]: M.Mendonca, et. al., A Survey of SDN: Past, Present, and Future of Programmable Networks http://hal.inria.fr/docs/00/83/50/14/PDF/bare_jrnl.pdf*

**UPB Master Courses 2015-2016**

# 3. SDN-OpenFlow

- **Controller Implementation Examples**

*Source:M.Mendonca, et.al., A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks http://hal.inria.fr/docs/00/83/50/14/PDF/bare_jrnl.pdf*

| Controller name | Implem. | Open Source | Developer | Characteristics |
|---|---|---|---|---|
| NOX | Python/C++ | y | Nicira | General, first SDN controller |
| POX | Python | Y | Nicira | General |
| MUL | C | Y | Kulcloud | Multi-threaded infrastructure, multi-level north-bound I/F |
| Beacon | Java | Y | Stanford | Cross-platform, modular, event-based and threaded operation |
| Trema | Ruby/C | Y | NEC | Framework for developing OpenFlow Ctrl. |
| Maestro | Java | Y | Rice University | NOS, provide I/F to develop modular network control |
| Jaxon | Java | Y | Independent | Based on NOX |
| Floodlight | Java | Y | BigSwitch | Based on the Beacon; works with PHY/V OF switches. |

**UPB Master Courses 2015-2016**

# 3. SDN-OpenFlow

- **Controller Implementation Examples**

*Source:M.Mendonca, et.al., A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks http://hal.inria.fr/docs/00/83/50/14/PDF/bare_jrnl.pdf*

| Controller name | Implem. | Open Source | Developer | Characteristics |
|---|---|---|---|---|
| SNAC | C++ | No | Nicira | Based on NOX; web-based, user-friendly policy manager: manage configure, monitoring |
| ovs-controller | C | Y | Independent | Simple OF ctrl. Ref. implem.with Open vSwitch ; manages any number of switches through the OF protocol; |
| Ryu | Python | Y | NTT,OSRG group | SDN OS ; provide logically centralized control and APIs to create new network M&C applications. Supports OpenFlow v1.0, v1.2, v1.3, and the Nicira Extensions. |
| NodeFlow | JavaScript | Yes | Independent | Written in JavaScript for Node.JS |
| Flowvisor | C | Y | Stanford/Nicira | Transparent proxy between OF switches and multiple OF controllers; can create network slices and delegate control of each slice to a different controller; isolation between slices. |
| RouteFlow | C++ | Y | CPQD | Provide virtualized IP routing over OF capable hardware. It is composed by an OF Ctrl. Appl., an independent server, and a VNet environment reproducing the connectivity of a PHY infrastructure; it runs IP routing engines. |

**UPB Master Courses 2015-2016**

# CONTENTS

- **SDN Scalability issues**

- **Why scalability-related questions"?**
  - **Reasons:**
    - centralized (more or less)  control plane
    - signaling overhead ( forwqarders- controllers)
  - A single  central controller will not scale for larger networks (no. of switches, flows, bandwidth, etc.)
    - E.g.NOX (the first SDN controller), can process max. 30,000 flow initiations per sec, [see NOX Refs] if less than 10 ms install time per flow is wanted
- Early SDN experiments and proposals have been flow-based → additional flow initiation delay .
- However, recent studies  show that SDN scalability is manageable and does not raise more problems than traditional networking control plane design

  - Source[29]:*S. H.Yeganeh, A.Tootoonchian, Y. Ganjali , On Scalability of Software-Defined Networking,  IEEE Comm. Magazine, February 2013.*

# 4. SDN Extensions and Advanced Architectures

- **SDN Scalability issues**
- CPl/Dpl decoupling issues
  - need standard API between Cpl/DPI - to allow their independent evolutions - not so simple
  - switch manufacturers should adopt the same APIs ( compatibility reasons)
  - moving control far away from switches/routers → may create signaling overhead ( both directions)
- **Controller scalability**
  - a single controller will be for sure a bottleneck if the number of switches and flows increase
- ***Solution proposals examples:***
  - **Direct solution [30]:** increase the controller processing power ( through better management, multicore processors)
    - *Source: A. Tootoonchian et al., "On Controller Performance in Software-Defined Networks," Proc. USENIX Hot-ICE '12, 2012, pp. 10–10.*
  - ( **DIFFANE** [31]) : Proactive pushing the states to the data paths
    - *Source: M. Yu et al., "Scalable Flow-Based Networking with DIFANE," Proc. ACM SIGCOMM 2010 Conf., 2010, pp. 351–62.*

# 4. SDN Extensions and Advanced Architectures

- **SDN Scalability issues**

- **DevoFlow** [32]: **Dividing the flows in two classes** ( to reduce the controller tasks)-
  - Short lived flows – handled in Data path only
  - Long lived/larger flows- forwarded to the controller
  - *Source: A. R. Curtis et al., "DevoFlow: Scaling Flow Management for High-Performance Networks," Proc. ACM SIGCOMM '11, 2011, pp. 254–65.*

- **Multiple controllers- solution for large networks**
  - **Several controllers ( distributed DPI)**
  - However maintaining the unified view on the network ( to benefit from SDN advantages)
  - Need to maintain consistency between them
    - Full/strong consistency is difficult to achieve ( affects the control plane response time)
    - Define a convenient consistency level (while maintaining availability and partition tolerance)
    - The necessary degree of consistency between several controllers depends on the type of control applications
    - No standard protocols defined yet for communication/synchronization between controllers

# 4. SDN Extensions and Advanced Architectures

- **SDN Scalability issues**
- **Examples:**

- **Onix [33] :**
  - distributed control platform implementing a **distributed CPI**
  - It provides general APIs for control appl. to access network state (NIB), which is distributed over ONix instances.
  - *Source: T. Koponen et al., "Onix: A Distributed Control Platform for Large-Scale Production Networks," Proc. 9th USENIX, OSDI Conf., 2010,*

- **Kandoo [ 34]**
  - **distributed the control plane**
  - It defines a scope of operations → applications with different requirements can coexist:
    - **locally scoped applications** (they can operate using the switch local state) are deployed close to the data path and shield other parts of the control plane from the load
  - A root controller, maintains a network-wide state and coordinates local controllers.
  - *Source: S. H.Yeganeh and Y. Ganjali, "Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications," Proc. HotSDN '12 Wksp., 2012*

**UPB Master Courses 2015-2016**

# 4. SDN Extensions and Advanced Architectures

- **SDN Scalability issues**

  - **HyperFlow [ 35]**
    - multiple controller
    - it synchronizes network state among controller instances
    - the control applications (running on every controller instance) see a virtual single control over the whole network.
    - *Source: A. Tootoonchian et.al., "Hyperflow: A Distributed Control Plane for OpenFlow," Proc. 2010 INMConf., 2010.*

- **Flow processing issues**

- **Early SDN proposals :**
  - *Reactive style* of flow handling  ( i.e all flows are first processed in the controller) → high flexibility (fine-grained high-level network-wide policy enforcement) but it introduces a flow setup delay and  limits the scalability.

- **Alternative solution:**
  - *Proactive style-* forwarding entries are set up before the initiation of actual flows can avoid the flow setup delay penalty altogether.

## 4. SDN Extensions and Advanced Architectures

- **SDN Scalability issues**

- **Other methods to improve scalability**

  - Placing the controllers in the proximity of the group of switches they control- may enhance the response time

    - Open issue: how to geo-distribute the controllers as to prepare upgrading of the network in different regions

- **Aggregation of rules:**
  - The control program may
    - define aggregate rules( i.e per classes like in DiffServ technology)  matching a large number of micro-flows,

    - proactively install rules in the forwarders to provide E2E connectivity and identify quality of service (QoS) classes,

    - classification and reactive labeling flows  may be performed at the edges

**UPB Master Courses 2015-2016**

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture**

  - **SDN is a promise for enhancing the networking flexibility and performance**
  - **Going further: define a flexible architecture "software defined"**

  - Recent proposal:

  - **Main ideas:**
    - Make the architectural evolution more flexible through software

  - General comment: attempts to solve incrementally the Internet defficiencies, including "clean slate" ones – had limited success
    - By decoupling the architecture w.r.t infrastructure

    - Authors (*) claim that even after recent advances (including "clean slate-ICN/CCN, etc. and SDN) the *architecture* remained *coupled* with *infrastructure*
      - Architecture: IP protocols and packet handling rules
      - Infrastructure: PHY equipments
    - Coupling means that changes at IP level will need some changes in the routers (e.g. because lack of ASIC flexibility)

  - *(*) Source [36]: B. Raghavan, T, ".Koponen, A.Ghodsi, M.Casado, S.Ratnasamy, S.ShenkerSoftware-Defined Internet Architecture: Decoupling Architecture from Infrastructure, Hotnets '12, October 29–30, 2012, Seattle, WA, USA."*

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture (cont'd)**

  - OpenFlow has increased the flexibility but still does not solve the decoupling;architecture/infrastructure
  - to support a wide range of architectures, the forwarders should support very general set of matching rules and fwd. actions.
    - Big header size, cost

- Proposal in (*) considers useful features of several technologies and tries combine them in an intelligent way as to realize that decoupling:

  - **MPLS** : (distinction :edge/core, partial separation DPI/CPI )

  - **SDN:** separation CPL/DPI, I/F through which the CPI can program the forwarders

  - **Middleboxes** (perform tasks beyond IP fwd.)

  - **SW forwarding** (based on fast processors)- the other extreme is ASIC based routers ( highest ratio cost/perf)

**UPB Master Courses 2015-2016**

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture (cont'd)**

  - Data Plane (DPI) splitted in
    - Core network ( its own addressing scheme)
    - Edge network

  - Architectural dependencies – placed at the edges

  - SW forwarding in the edge (assure flexibility)

  - Control Plane (CPI) uses SDN-like control to edge routers (can be OpenFlow- based but not mandatory)

  - Each core network domain has its own design

  - **The approach allows a top-down perspective**

    - Still SDN style of control is proposed
    - Openflow or equivalent is needed to be standardized

    - However – no need to specify beforehand the behavior of each box- because the controller assures interoperation

**UPB Master Courses 2015-2016**

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture (cont'd)**

- Top-down perspective
  - Tasks ( to get E2E connectivity):
    - *Interdomain: Domain A-Domain B*
    - *intradomain transit*
    - *intradomain delivery ( from domain edges to/from hosts or between hosts)*
  - *Main suggestion: separation between* intradomain and interdomain addressing
  - interdomain addressing :
    - some form of domain identifiers, to support interdomain task
    - no ref. to any intradomain addresses (each domain can choose its own internal addressing scheme
    - this important choice can be solved in "clean-slate" style or some specific solutions can be applied ( e.g. using the IPv6 flow ID as the interdomain Id.)
  - each domain is represented by a single logical server in the algorithm to compute interdomain routes
    - the server may be replicated for reliability, but a single logical entity represents the domain for interdomain routing algorithm
      - Routing alg.: BGP-like or *any other new algorithm*

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture (cont'd)**

- Intradomain tasks: *edge-to-edge transit, edge-to-H delivery, and H-to-H delivery*
    - -implemented independently w.r.t. interdomain task
    - different domains can use different implementations for intradomain tasks ( e.g MPLS)
    - the core can use *any internal fwd and control plane* ( SDN…. traditional protocols)
    - each domain's core can use their own internal addressing scheme.

- The edge uses SW fwd.
    - commodity processors managed by an SDN edge controller
    - SDN edge controller knows the core requirements to insert the appropriate packet headers to achieve internal or edge delivery.

- Result: highly modularity

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture (cont'd)**

  - **Advantages:**
    - Only the edge routers need to understand interdomain addressing
    - Core routers need to understand intradomain addressing in their domain only
    - Only the edge-controller participate in the interdomain route computation
    - Only the core Cpl needs to determine the internal routes
    - The only components needed to forward packets based on interdomain addresses are edge routers, which use software forwarding.

  - **Result: high architectural freedom**

  - Question: SW fwd- is it realistic in this context?
    - apparently yes , encouraging results [ ]: longest-prefix match forwarding on minimum-sized packets, including checksum verification and TTL adjustment, can be done at 6.7Gbps on a single 3.3Ghz core.

# 4. SDN Extensions and Advanced Architectures

- **Software Defined Internet Architecture (cont'd)**

- **SDIA defines an Interdomain Service Model (ISM) [ 36]:**
  - only edge controllers  (one per domain) are involved in  interdomain task
  - and edge routers (controlled by the edge controller)
- **Implications:**
  - Interdomain routing changes ( e.g. BGP to  other) only involve changing SW in the edge controllers
  - Changing how domains are addressed → a change only to the controller SW
  - Changing how hosts are addressed, (e.g. IP to IPv6), is done per domain.

- **ISM in SDIA main requirements:**
  - A distributed interdomain algorithm between *the edge controllers* that computes whatever state the controllers need to implement the service model; (e.g. BGP)

  - A set of forwarding actions to be sent to the edge routers by the edge-controllers.
  - Allow incremental/partial  deployment; need a basic unicast packet-delivery ISM (such as supplied by IP and BGP), so that non-peering domains can set up tunnels with each other

  - a discovery mechanism : domains participating in an ISM are aware of each other.

# 4. SDN Extensions and Advanced Architectures

- Software Defined Internet Architecture (cont'd)

  - Illustration of the ISM principles
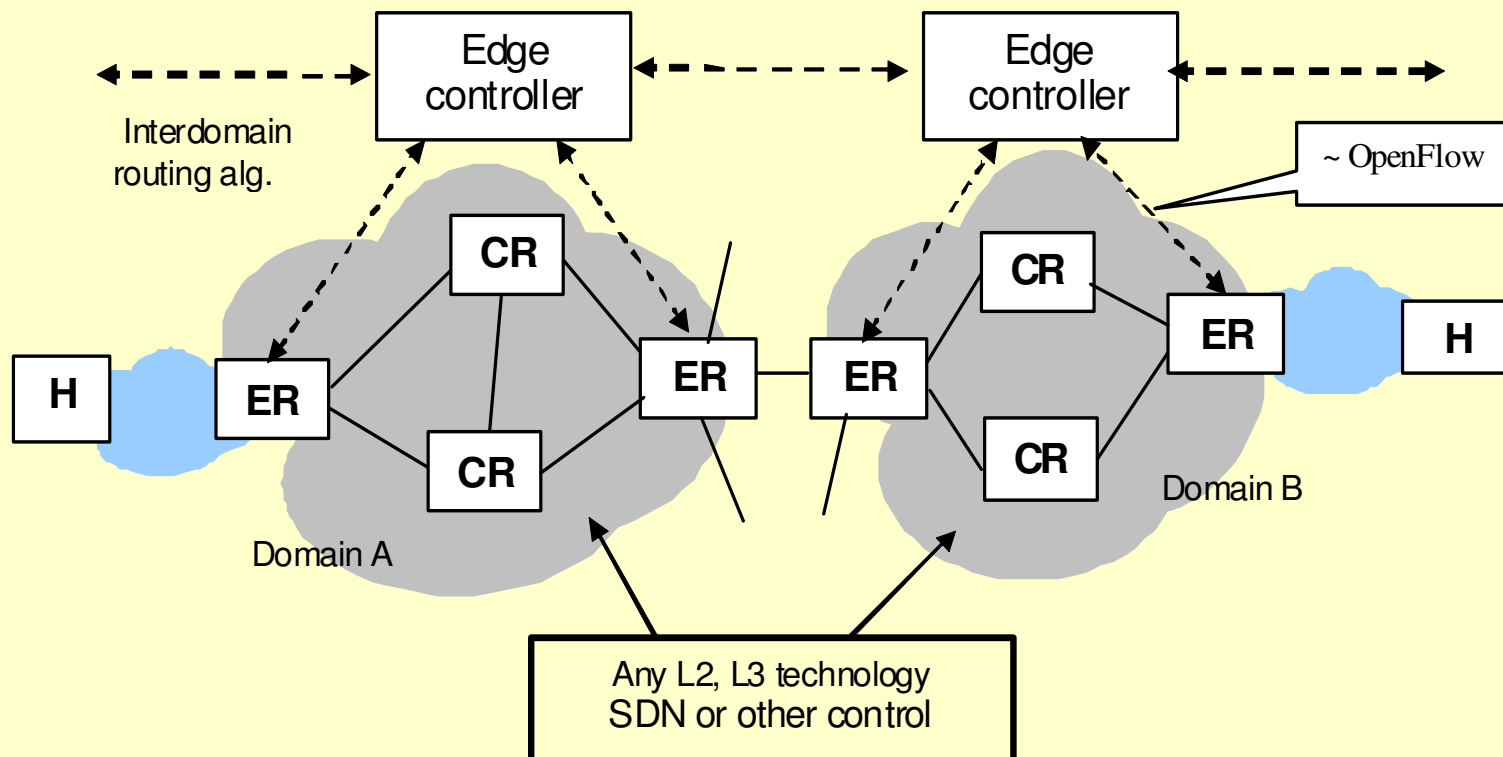
# CONTENTS

1. Software Defined Networks (Basics)
2. SDN Applications
3. SDN – OpenFlow
4. SDN Extensions and Advanced Architectures
5. ☞ **Other recent technologies- SDN approach**
6. Conclusions

## 5. Other recent technologies - SDN approach

- **Cloud Computing summary**

  - **The NIST Definition of Cloud Computing (CC)**
  - (NIST: National Institute of Standards and Technology USA) [37,38]

- CC: a model for enabling ubiquitous, convenient, *on-demand* network access
  - to *a shared pool* of *configurable computing resources* (e.g., networks, servers, storage, applications, and services)
  - that can be *rapidly provisioned and released* with minimal management effort or service provider interaction.

- Cloud model : five essential characteristics , three service models,  four deployment models.

- Basic Cloud Characteristics [38 ]
  - *On-demand self-service*
  - *Broad network access*
  - *Resource pooling*
  - *Rapid elasticity.*
  - *Measured service.*



Access a Web based Application from Any connected devices using:
1. Web Browser
2. Internet /VPN network connectivity
3. Secure ID & Payment

**UPB Master Courses 2015-2016**
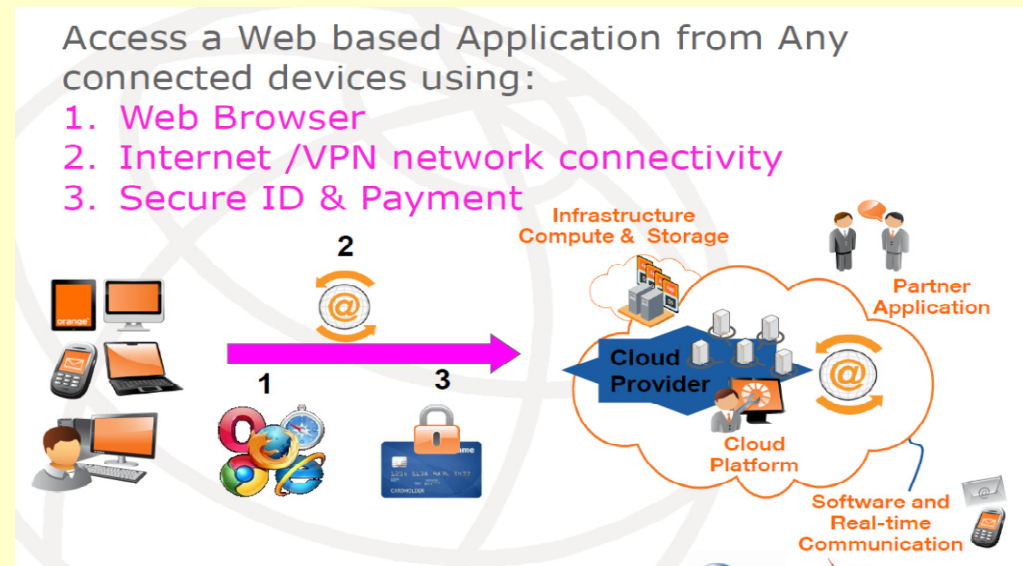
# 5. Other recent technologies - SDN approach

- Cloud Computing summary
  - **Service Models**
    - *Software as a Service (SaaS)*
      - Consumer can use the **provider's applications running on a cloud**
      - Applications are accessible from client devices (thin client I/F, such as a *web browser* or a *program interface.*
      - The consumer *does not manage or control the underlying cloud infrastructure* (network, servers, OS, storage, or even individual application capabilities)
        - possible exception : limited user-specific application configuration settings
    - *Platform as a Service (PaaS)*
      - Consumer
        - can deploy on the cloud infrastructure **consumer-created or acquired applications created using programming languages, libraries, services, and tools** supported by the provider
        - **does not** manage or control the *underlying cloud infrastructure* (network, servers, OS , storage)
        - but *has control over the deployed applications* and possibly configuration settings for the application-hosting environment
    - *Infrastructure as a Service (IaaS)*
      - Consumer
        - can **provision processing, storage, networks**, and other computing resources
        - *Can deploy and run arbitrary software*, (including OS and applications)
        - **does not** manage or control the underlying cloud infrastructure
        - but has control over OS, storage, and deployed applications
        - and possibly *limited control* of select networking components (e.g., host firewalls)

**UPB Master Courses 2015-2016**

# 5. Other recent technologies - SDN approach

- **Cloud Computing summary**
  - **Deployment Models**
  - *Private cloud*
    - The *cloud infrastructure is provisioned for exclusive use* by a *single organization*
  - *Community cloud*
    - The cloud infrastructure is provisioned for exclusive use by a *specific community* of consumers from organizations that have *shared concerns* (e.g., mission, security requirements, policy, and compliance considerations).
  - *Public cloud*
    - The cloud infrastructure is provisioned for *open use by the general public.*
    - It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
  - *Hybrid cloud*
    - The cloud infrastructure is a *composition of two or more distinct cloud infrastructures* (private, community, or public) that remain unique entities

**UPB Master Courses 2015-2016**

# 5. Other recent technologies - SDN approach

- **Cloud Computing summary: NIST Reference architecture [38]**
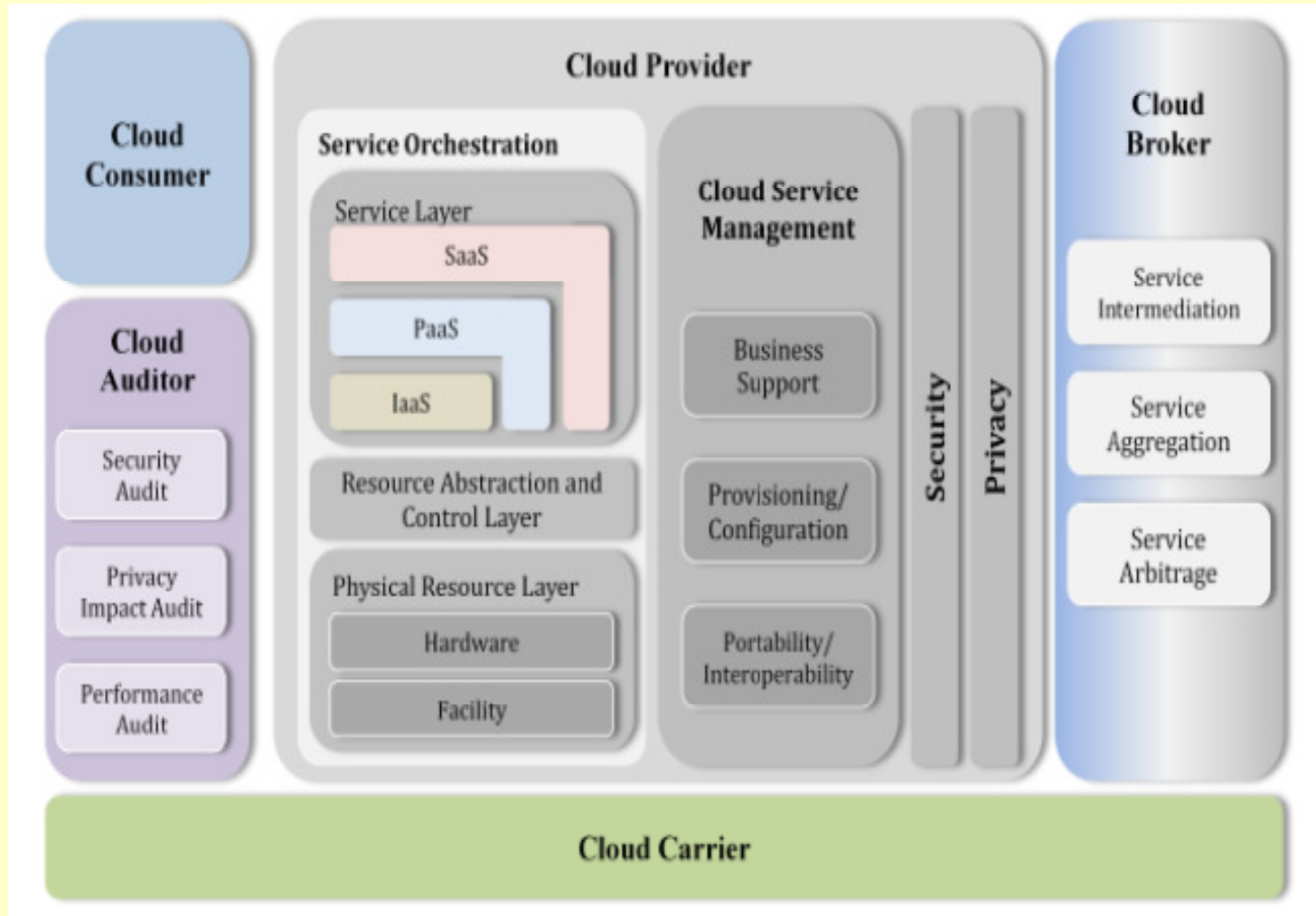
# 5. Other recent technologies - SDN approach

- Cloud Computing summary: NIST Reference architecture
  - Main actors

| Actor | Definition |
|---|---|
| **Cloud Consumer** | A person or organization that maintains a business relationship with, and uses service from, *Cloud Providers*. |
| **Cloud Provider** | A person, organization, or entity responsible for *making a service available* to interested parties. |
| **Cloud Auditor** | A party that can conduct *independent assessment* of cloud services, information system operations, performance and security of the cloud implementation. |
| **Cloud Broker** | An entity that *manages the use, performance and delivery of cloud services, and negotiates relationships* between *Cloud Providers* and *Cloud Consumers*. |
| **Cloud Carrier** | An *intermediary* that provides *connectivity and transport of cloud services* from *Cloud Providers* to *Cloud Consumers*. |

**5.** **Other recent technologies -**
**SDN approach**

- **Cloud Computing summary**
- ITU-T position on Cloud Computing [40, 41] :
  - *Source: ITU-T* Focus Group on Cloud Computing Technical Report (2012)
  - **Cloud Eco-system**
    - **Cloud Service Provider (CSP)** An organization that provides and maintains delivered cloud services
    - **Cloud Service User (CSU)** A person or organization that consumes delivered cloud services
      - Consumer, Enterprise (including enterprise administrator), Governmental/public institution
    - **Cloud Service Partner (CSN)** A person or organization that provides support to the building of the service offer of a cloud service provider (e.g. service integration).
      - Application developer, Content provider, Software provider, Hardware provider,
      - Equipment provider, System integrator, Auditor

# 5. Other recent technologies - SDN approach

- **Cloud Computing summary**
- ITU-T position on Cloud Computing:

  - **New types of Cloud Services defined by ITU-T**
    - ***Communication as a Service - CaaS*** : real-time communication and collaboration services
      - audio/video communication services (VoIP, A/VC), collaborative services, unified communications, e-mail, instant messaging, data sharing (web conference)

    - ***Network as a Service – NaaS*** : transport connectivity services and/or inter-cloud network connectivity services.
      - **Managed Internet** (guaranteed speed **,** availability, etc.) virtualized networks (VPNs), coupled with cloud computing services, flexible and on demand bandwidth

- **Cloud Computing summary**
- ITU-T position on Cloud Computing:

| Business roles versus Actors | Application CSP | Platform CSP | Infrastructure CSP | Cloud Service Partners | Cloud Service Users | Inter-cloud |
|---|---|---|---|---|---|---|
| Telecom SP | X | X | X | X | (X) | X |
| Internet SP | X | X | X | (X) | (X) | X |
| 3rd Party Provider | (X) | (X) | (X) | X | (X) | |
| User | | | | | X | |

ITU-T
Actors of a Cloud Ecosystem

Possible roles in cloud scenario
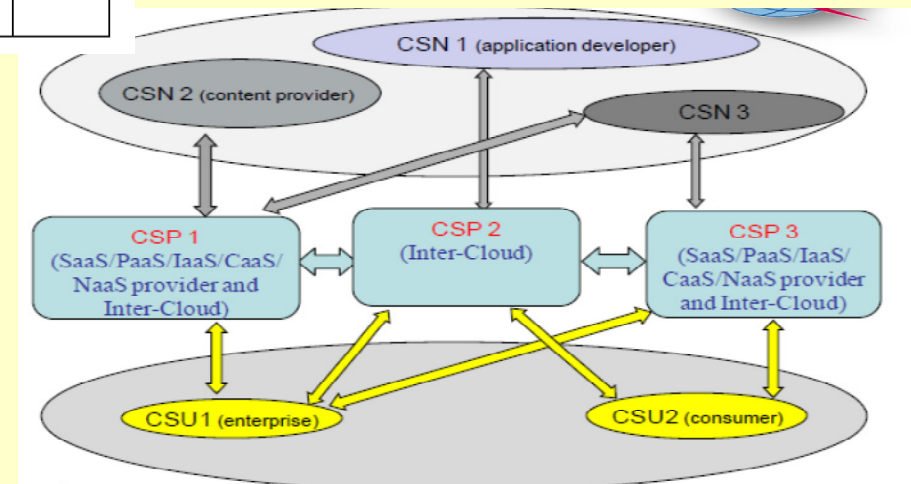
## 5. Other recent technologies - SDN approach

- **SDN in Cloud technology?**

  - SDN: Admin can change dynamically network switch's rules - prioritizing, de-prioritizing or even blocking specific types of packets (granular level of control)
    - This is helpful in a cloud computing multi-tenant architecture
      - administrator can manage traffic loads in a flexible and more efficient manner

  - Some key requirements for IaaS "cloud networking"/SDN:
    - Multi-tenancy
    - L2, L3 isolation
    - Scalable control plane
    - NAT (floating IP)
    - ACLs and Stateful (L4) firewall
    - VPN
    - BGP gateway
    - RESTful API
    - Integration with CMS (like OpenStack)

**UPB Master Courses 2015-2016**

- **Example: Bandwidth Exchange service**
- Useful for Cloud Carriers also but more general

- **Problem**: the sourcing of raw bandwidth to deliver networking services is a constant challenge for *Network Operators ( NO) and Virtual Network Operators (VNOs).*
  - - need to avoiding excessive idle inventory.

- *Predictive business models* are used to evaluate bandwidth requirements on existing and future extensions of their networks.

- **Models usage:**
  - VNOs determine when and where to *lease* wholesale capacity.
  - NO drive their *"buy/build versus lease"* decisions:
    - *buy* decisions involve CAPEX/OPEX but provide operators with the greatest control over their network infrastructure
    - bandwidth *lease* decisions may make sense where capacity does not warrant dedicated builds or is not expected to grow substantially over time.

- **Bandwidth Exchange service**

- However, planning and forecasting are by nature imperfect.
    - The amount, timing, and location of bandwidth demands are difficult to predict, leading to network segments or links with either excessive or insufficient capacity.
    - Lengthy lead times from equipment suppliers or traditional wholesale bandwidth providers can exacerbate the issue,

- Idle bandwidth (not generating revenue) is undesirable, but also the ability to competitively/rapidly respond to new opportunities is needed for NO

- *Bandwidth exchange marketplaces* : alternative means for NO to address capacity planning challenges.
    - bandwidth suppliers and buyers can trade bandwidth like a commodity (with options and futures), and provide a means *to transfer the control of real bandwidth resources* between parties.
    - facilitate a common inventory of bandwidth contracts that can be priced and exchanged in an automated fashion.

**UPB Master Courses 2015-2016**

## 5. Other recent technologies - SDN approach

- **Bandwidth Exchange service**

- **Bandwidth exchange markets provide other benefits:**
  - Quicker ROI from projects requiring capacity expansion
  - Reduced operational costs ← automation of bandwidth acquisition and provisioning
  - Optimized network expansion costs associated with "leased" bandwidth, taking advantage of general market pricing efficiency and elasticity

- Bandwidth exchange markets challenges/needs:
  - *Registration, tracking, and mgmt.* of available time-based bandwidth inventory across multiple network domains
  - *Secure and automated orchestration, scheduling, coordination, and provisioning* of resources between multiple supply and demand entities
  - *Integration of network operators' management* systems into the bandwidth exchange BSS/OSS
  - *Policy management and admin.* of bandwidth resources according to parameters such as time, duration, volume, and location
  - *Monitoring and enforcement* of standardized SLAs across the market

## 5. Other recent technologies - SDN approach

- **Bandwidth Exchange service**

- **Next Figure** illustrates the concept of bandwidth exchange markets and a *reference implementation architecture.*

- **Suppliers and buyers meet at common, neutral exchange "points" where bandwidth transactions can occur, such as for carrier Ethernet or optical transport bandwidth.**

- **Examples**
  - Enterprise companies, may utilize the bandwidth exchange to acquire the most cost-effective leased bandwidth service to **Cloud Provider A** for a transaction-oriented cloud operation.

  - **Operator A** may use the bandwidth exchange to locate cost-effective bandwidth to interconnect two of its disjoint networks.

  - Diverse bandwidth requirements could be met by leasing bandwidth from different exchange points over different suppliers' networks (**Operator B and Bandwidth Provider C).**
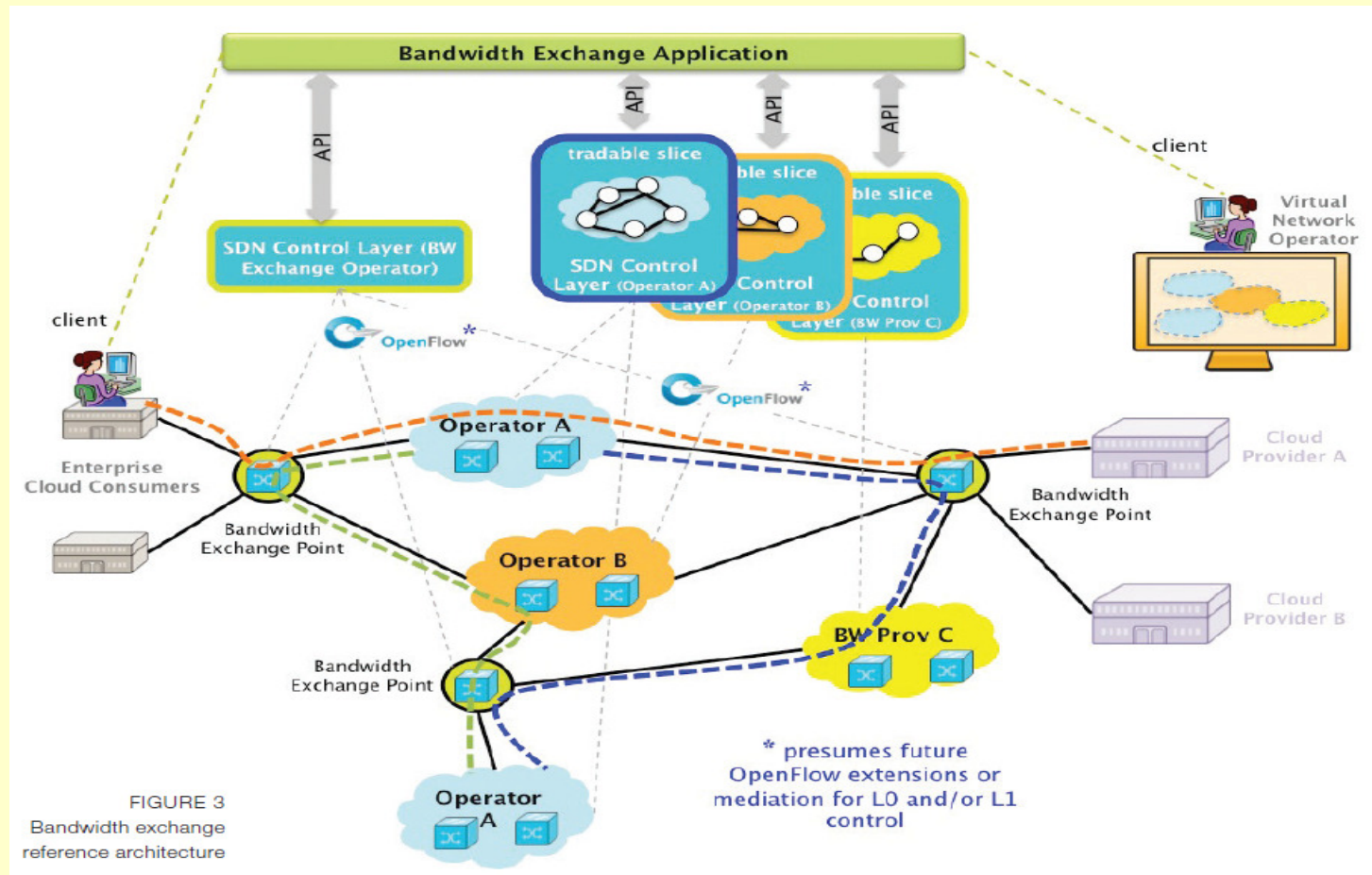
**UPB Master Courses 2015-2016**

- Bandwidth Exchange



FIGURE 3
Bandwidth exchange
reference architecture

*Source: Operator Network Monetization Through OpenFlow-Enabled SDN, ONF Solution Brief, April 3, 2013, https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-network-monetization.pdf*

**UPB Master Courses 2015-2016**

## 5. Other recent technologies - SDN approach

- **Bandwidth Exchange service**
- ***SDN –OF provides essential tools for enabling automation*** of the processes necessary to achieve real-time bandwidth trading.
  - It can play a key role in supplier networks as well as in the bandwidth exchange market.
  - With bandwidth resources spanning multiple networks, *canonical abstractions* of available network services and bandwidth resources across multivendor environments are essential.

- ***SDN directly controls elements of the network architecture***
  - provisioning the bandwidth exchange point switches interconnecting these networks
  - slicing and dynamically allocating bandwidth in the supplier's network.

- Standardized representation and configurability of bandwidth flows are essential for enabling a bandwidth exchange marketplace to uniformly manage bandwidth inventory and provision flows in a multitenant environment.

- **Bandwidth Exchange service**

- **Conclusion on SDN approach**
  - allows bandwidth suppliers **to partition their networks** into "public/tradable" and "private/non-tradable" network slices.

  - provide **secure access and control** of their designated tradable bandwidth to the bandwidth exchange application, where their bandwidth can be pooled with tradable resources from other suppliers

  - the **matching of demand with supply**, based on duration and temporal availability, can be achieved through the logically centralized SDN control layer's global perspective.

  - The **open API and Open Flow I/Fs facilitate the rapid, on-demand provisioning** of resources and the automation of workflow processes

- **ICN/CON/CCN- versus SDN**

    - recent attention : research /industry / operators
    - propose some fundamental changes for TCP/IP networking
      - claiming several advantages in the perspective of Future Internet

    - **Still open questions:**
        - what significant benefits does ICN designs offer?
        - are ICN designs the best solution to achieve those benefits?
        - Is the current technology prepared to introduce soon these changes?
        - Seamless dvelopment?

    - **Terminology**
        - Not standardised, different (overlapping) semantics…

            - **ICN/CCN  - Information/Content  Centric Networking**
            - **CON - Content Oriented Networking**
            - **DON - Data Oriented Networking**
            - **CAN - Content Aware Networking**
            - **NDN - Named Data Networking**

        - **Examples of ICN/CON Projects**
            - EUROPE : PSIRP, 4WARD, PURSUIT, SAIL, …
            - USA: CCN , DONA , NDN, CCNx, …

- **ICN/CON/CCN- versus SDN**

- The content-oriented paradigm: *content-oriented, content centric,content-based, data-oriented,* or *data-centric network* are considered to be equivalent in that they focus on not the communication party but the content or data itself

  - **However**
    - there is little common terminology between different ICN/CON/CCN, …proposals
    - no common framework → the focus is often on low-level mechanisms
    - many studies accentuate the differences between their design and others
    - they do not clarify enough the construction of the ICN assembly

      - *Source[46]: A.Ghodsi, T.Koponen, B.Raghavan, S.Shenker, A.Singla, J.Wilcox, Information-Centric Networking: Seeing the Forest for the Trees, http://www.icsi.berkeley.edu/~barath/papers/icn-hotnets11.pdf*

# 5. Other recent technologies - SDN approach

- **ICN**
  - Infrastructure providing in-network caching
  - Content is distributed in a scalable, cost-efficient & secure manner
  - Receiver-driven model – subscribe/get objects of interest
  - Support for location transparency, mobility & intermittent connectivity
  - Still need to support interactivity (A/V) and location oriented services (e.g. similar service as telnet)
    - *Source [48]: G. Pavlou, Information-Centric Networking: Overview, Current State and Key Challenges, IEEE ISCC 2011 Keynote,* http://www.ee.ucl.ac.uk/~gpavlou/

  - **ICN:**
  - **the principal paradigm is not E2E communication between hosts**
  - **high amount of content need efficient distribution**
    - **information objects as a first-class abstraction;**
    - **focusing on the properties of such objects and receivers' interests to achieve efficient and reliable distribution of such objects**
    - **In-network storage, multiparty communication through replication, and interaction**
    - **publish-subscribe models generally available for all kinds of applications,**
    - **No more need of dedicated systems such as peer-to-peer overlays and proprietary CDNs**

    - *Source[53]: D. Kutscher, B.Ahlgren, H.Karl, B. Ohlman, S.Oueslati I.Solis, Information-Centric Networking— Dagstuhl Seminar — 2011*

# 5. Other recent technologies - SDN approach

- **CON**
  - Decoupling contents from hosts (or their locations) not at the application but at the network level
  - Hope to solve or mitigate also other Internet problems (mobility, security).
  - Free application/service developers from reinventing application-specific delivery mechanisms
  - Scalable and efficient delivery of requested contents (e.g., by supporting multicast/ broadcast/anycast)
    - CON: dealing with *content objects*: *naming, locating/routing, deliver/disseminate, caching in-network*
    - CON ~ ICN~CCN
    - *Source [43]: J. Choi, J. Han, E.Cho, T.Kwon, and Y.Choi "A Survey on Content-Oriented Networking for Efficient Content Delivery"IEEE Communications Magazine, March 2011pp. 121- 127*

- **CCN**
  - **CCN treats content as a primitive** – decoupling location from identity, security and access, and retrieving content by name
  - New approaches to routing named content,
  - derived from IP, one can achieve scalability, security and performance
    - *[50] Van Jacobson, D.K. Smetters, J.D. Thornton, M. F. Plass, NH. Briggs, R.L. Braynard, Networking Named Content, Palo Alto Research Center, Palo Alto, CA, October 2009*
- **CAN-NAA**
  - Content awareness at network level and content oriented processing
  - Network awareness at service / application layer

# 5. Other recent technologies - SDN approach

- **Content-oriented concepts**
  - CON node : **routing by content names**, not by (host) locators
    - *hosts Identification* is replaced by *content identification*.
    - content file location - independent of its name
    - *content naming* and *routing – independent of* location
    - free from mobility and multi-homing problems

  - **Publish/subscribe (P/S)** communication model
    - Essential in CON:
    - A content source announces (or *publishes*) a content file
    - An user requests (or *subscribes* to) the content file.
    - P/S
      - **decouples the content generation and consumption** in time and space
      - so contents are delivered efficiently and scalably (e.g., multicast/anycast)

      - *Source [43]: J.Choi, Jinyoung Han, E.Cho, Ted Kwon, and Y.Choi,A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Communications Magazine • March 2011*

**UPB Master Courses 2015-2016**

# 5. Other recent technologies - SDN approach

- **ICN/CON/CCN- versus SDN**

  - ICN: information objects names are unique and independent of locations, applications, storages and distribution → allows ubiquitous information retrieval.

  - **Typical ICN architectures/deployments**
    - 1. ICN over IP (encapsulate ICN protocol data in IP or UDP/TCP packets or take ICN protocol information using IP options;
    - 2. ICN over L2, (completely replace IP layer and L2 protocols (Ethernet, IEEE 802.x)
    - 3. ICN over virtualized network (exploit network virtualization technologies, e.g. SDN)

  - The *draft-icn-implementation-sdn-00* – **proposes a unified framework based on SDN concepts**
    - *Source: W. Liu, J. Ren, J. Wang, draft-icn-implementation-sdn-00, „A Unified Framework for Software-Defined Information-Centric Network" August 09, 2013*

# 5. Other recent technologies - SDN approach

## Example of CCN stack

CCN : new "thin waist" of the Internet: IP → to chunks of named content
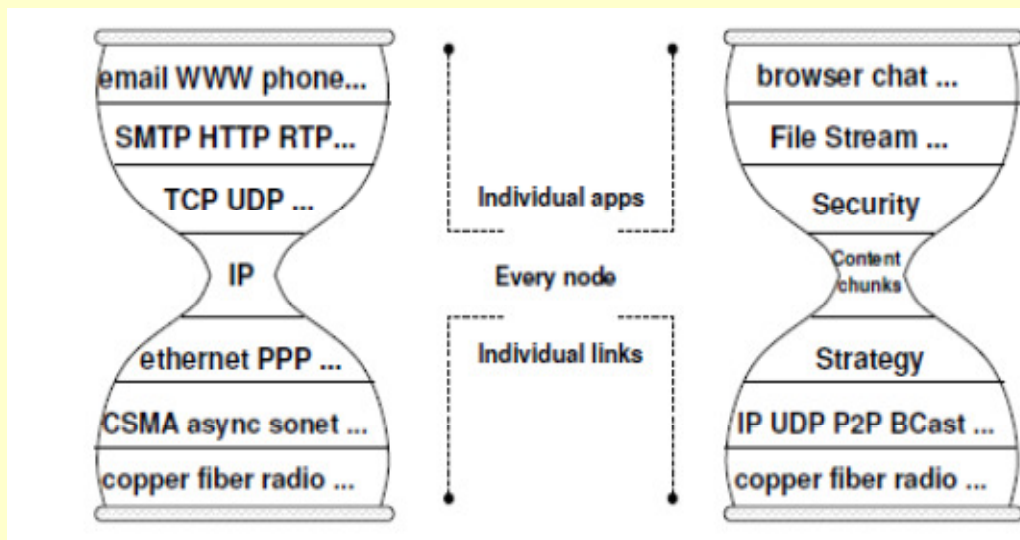


Traditional
TCP/IP stack

→
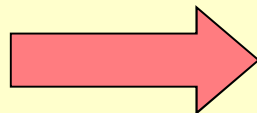
**Original
picture
CCN**

Source [50]: Van Jacobson Diana K. Smetters James D.
Thornton Michael F. Plass, Nicholas H. Briggs Rebecca L.
Braynard, Networking Named Content, Palo Alto Research
Center, Palo Alto, CA, October 2009

| Application | Applications: browser chat, file stream: |
|---|---|
| | **Security** |
| | **Content chunks** |
| | **Strategy** |
| | P2P, .. |
| TCP, UDP, … | UDP |
| IP | Intra-domain routing:OSPF, .. Inter-domain routing: BGP, ... (placed here to show their role) |
| Data link | Any Layer 2 |
| Physical Layer (wireline, wireless) | Any PHY |

Alternative view of CCN stack
(if it runs on top of IP)

# 5. Other recent technologies - SDN approach

- **Unified ICN-SDN framework**
  - *Source [42] Lin et.al, : draft-icn-implementation-sdn-00, „A Unified Framework for Software-Defined Information-Centric Network" August 09, 2013*
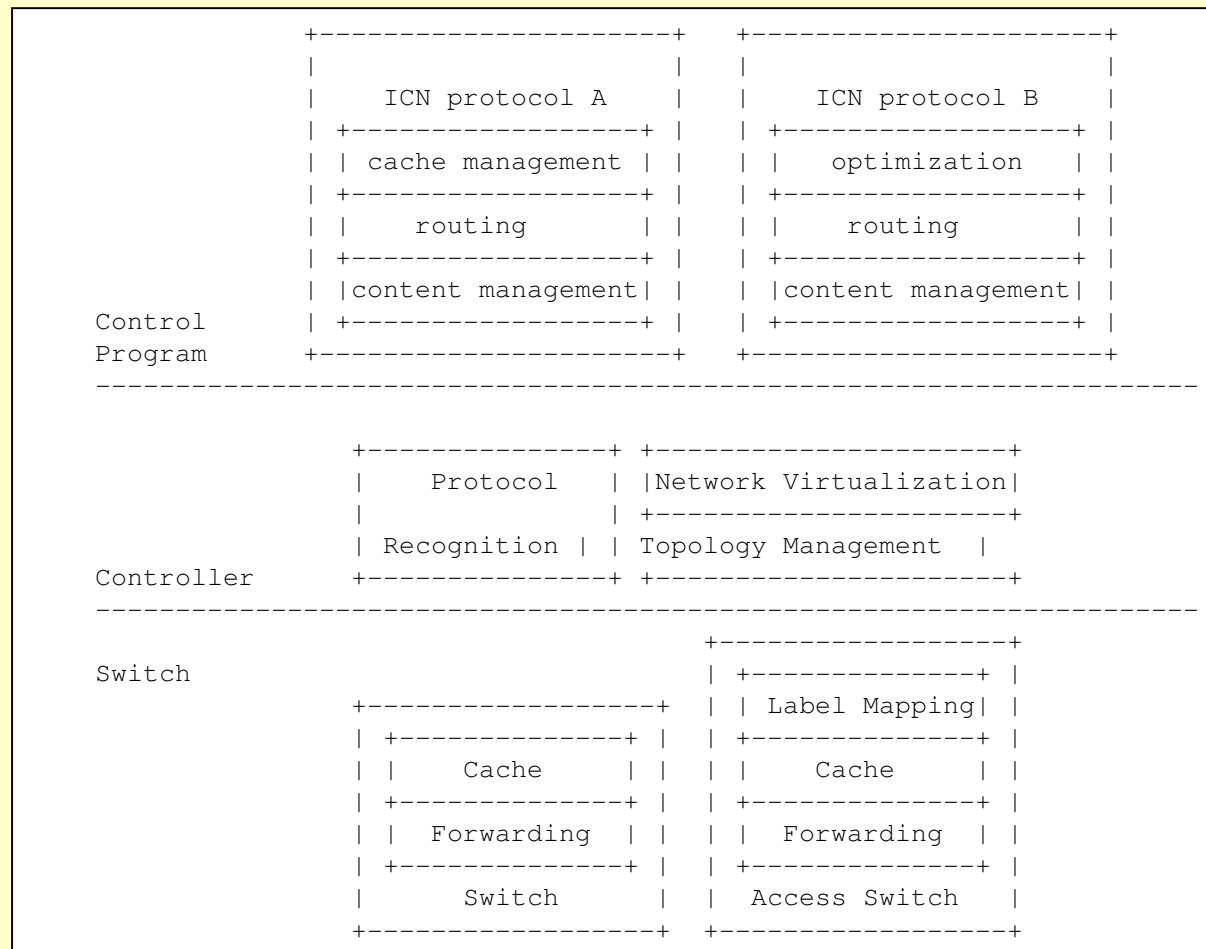
```
                +-------------------+   +-------------------+
                |                   |   |                   |
                |   ICN protocol A  |   |   ICN protocol B  |
                | +---------------+ |   | +---------------+ |
                | | cache management | |   | |  optimization  | |
                | +---------------+ |   | +---------------+ |
                | |    routing    | |   | |    routing    | |
                | +---------------+ |   | +---------------+ |
                | |content management| |   | |content management| |
Control         | +---------------+ |   | +---------------+ |
Program         +-------------------+   +-------------------+
          --------------------------------------------------------------


                +-------------+ +---------------------+
                |   Protocol  | |Network Virtualization|
                |             | +---------------------+
                | Recognition | | Topology Management  |
Controller      +-------------+ +---------------------+
          --------------------------------------------------------------
                                 +-----------------+
Switch                           | +-------------+ |
                +---------------+ | | Label Mapping| |
                | +-----------+ | | +-------------+ |
                | |   Cache   | | | |    Cache    | |
                | +-----------+ | | +-------------+ |
                | | Forwarding| | | | Forwarding  | |
                | +-----------+ | | +-------------+ |
                |    Switch     | | Access Switch  |
                +---------------+ +-----------------+
```

**UPB Master Courses 2015-2016**

# CONTENTS

1. Software Defined Networks (Basics)
2. SDN Applications
3. SDN-OpenFlow
4. SDN Extensions and Advanced Architectures
5. Other recent technologies- SDN approach
6. Conclusions

# 6. Conclusions

- **SDN – technology opening new perspective to networking:**

  - Flexibility ( decoupling DPL/CPl)
  - Programability
  - Better resource and policy management
  - Network equipment vendor independency
  - No huge problems of scalability ( despite some opinions)
  - Can be extended to SDIA

  - Incremental deployment – possible
  - Strong support from industry

  - **Open - research issues**
    - Distributed control plane versus – unique logical representation- in large networks
    - Coordination among several controllers
    - Reliability
    - Universality of API and OpenFlow – like protocol

- # Thank you !
- # Questions?

- # Backup slides

# 1. Software Defined Networking

- **1.1 Introduction (details)**

- Current network architectures only partially meet today's requirements of enterprises, carriers, and end users: open architectures, flexibility, programability, QoS, flexible resource management, etc.

- **Current network technologies limitations**

  - **Complexity that leads to stasis:**
    - Current networking : many discrete sets of protocols connecting hosts reliably over arbitrary distances, link speeds, and topologies

    - Protocols are defined in isolation, to solve a specific problem

    - No benefit of any fundamental abstractions -> complexity

    - To add or move any device, IT admin. must (re)configure multiple HW/SW entities using device-level management tools
      - should consider topology, vendor switch model, SW version, etc.

    - Network complexity ->  today's networks reconfigurations are performed relatively in static way (to minimize the risk of service disruption)

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.1 Introduction**

- **Current network technologies limitations (cont'd)**
- **Complexity that leads to stasis:**

  - The static nature of networks
    - It is not good for today's dynamic server environment, (server virtualization, VM migration)
      - applications are distributed across multiple virtual machines (VMs), which exchange traffic flows with each other.

      - VM migration : challenge for many aspects of traditional networking (addressing schemes, namespaces segmented, routing-based design).

- Limited capability for dynamic differentiated QoS levels because of – usually static provisioning

- Not enough capability to dynamically adapt to changing traffic, application, and user demands.

# 1. Software Defined Networking

- **1.1 Introduction**

- **Current network technologies limitations (cont'd)**

- **Inconsistent policies:**
  - Network-wide policy implementation  -> have to configure thousands of devices and mechanisms
  - The complexity of today's networks makes it very difficult to apply a consistent set of access, security, QoS, and other policies

- **Scalability issues:**
  - Complex network (10**5 network devices in data centers)

  - Over-subscription based on predictable traffic patterns is not working well; in today's virtualized data centres, traffic patterns are highly dynamic and it is difficult to predict

  - Mega-operators (e.g. Google, Yahoo!, Facebook), face scalability challenges
    - The number of of computing elements exploded
    - data-set exchanges among compute nodes can reach petabytes

# 1. Software Defined Networking

- **1.1 Introduction**

- **Current network technologies limitations (cont'd)**

- **Scalability issues (cont'd)**

  - Need "hyper-scale" networks to provide high-performance, low-cost connectivity among many physical servers (need automation)
  - Carriers have to deliver ever-higher value, better-differentiated services to customers
  - Multi-tenancy : the network must serve large groups of users with different applications and needs

- **Vendor dependency**
  - Carriers/enterprises want rapid response to changing business needs or user demands

  - Their ability to respond is limited by vendors' equipment product cycles (years)

  - Lack of standard, open I/F - limits the ability of network operators to tailor the network to their individual environments

# 1. Software Defined Networking

- **1.1 Introduction**

- **Need for a new network architecture**

  - **Changing traffic patterns:**
    - *Traffic patterns have changed significantly* within the enterprise data center: today's applications access different DBs and servers, creating a high M2M traffic before returning data to the end user device (different from classic client-server applications)

    - *Users- network traffic patterns changing:* they want access to corporate content and apps. from any type of device, anywhere, at any time

    - Enterprises : need of computing model, which might include a *private public or hybrid cloud,* resulting in additional traffic across the wide area network

  - **Need of flexible access to IT resources:**
    - *Increasing usage of mobile personal devices* such as smart-phones, tablets, and notebooks to access the corporate network

    - Need to accommodate these personal devices while *protecting corporate data and intellectual property* and meeting compliance mandates

**UPB Master Courses 2015-2016**

# 1. Software Defined Networking

- **1.1 Introduction (cont'd)**
- **Need for a new network architecture (cont'd)**

  - **Cloud services development:**
    - Significant growth of public and private cloud services ( SaaS, PaaS, IaaS, NaaS,..)  on demand and à la carte
    - IT's needs for cloud services : security, compliance, auditing requirements, elastic scaling of computing, storage, and network resources,etc.

  - **Need for more bandwidth:**
    - today's  high volume of  data requires massive parallel processing on thousands of inter-connected servers
    - demand for additional network capacity in the data center
    - data center networks : need of scaling to very large size, while maintaining any-to-any connectivity
    - Media/content traffic high increase- need of more bandwidth

# 1. Software Defined Networking

- **1.1 Introduction**

- Recent industry/research effort resulted in new approaches:
    - **Software- Defined Networking (SDN)** – aiming to transform networking architecture
    - **Open Networking Foundation** (ONF- non-profit industry consortium ) → OpenFlow I/F specifications for SDN

- **SDN architecture major characteristics:**
    - the *Control Plane (CPI)* and *Data Planes (DPI)* are decoupled
    - network intelligence and state are logically centralized
    - underlying network infrastructure is abstracted from the applications

- **Promises for enterprises and carriers** :
    - higher programmability opportunities, automation, and network control
    - enabling them to build highly scalable, flexible networks
    - fast adapt to changing business needs

- *Source: Software-Defined Networking: The New Norm for Networks ONF White Paper April 13, 2012*

- *Note: after many years of strongly defending a completely distributed control approach in TCP/IP architecture- now a more centralized approach is proposed ....*

# 1. Software Defined Networking

- **1.2 Earlier technologies related to SDN - details**
- **Open Signaling [3]**
  - **OPENSIG WG** (~1995)- attempt to make Internet, ATM, and mobile networks more open, extensible, and programmable"
  - Ideas: separation between the communication HW and control SW
  - Proposal: access to the network HW via open, programmable network I/Fs
    - allow new services deployment through a distributed programming environment.
- **IETF WG** - > General Switch Management Protocol (GSMP)
  - general purpose protocol to control a label switch.
  - establish and release connections across the switch
  - add/delete leaves on a multicast connection
  - manage switch ports, request configuration information, statistics
  - manage switch resources
  - GSMPv3, June 2002, WG has been concluded

# 1. Software Defined Networking

- **1.2 Earlier technologies related to SDN** (cont'd)
- **Active Networking** [4]
  - ~1995-2000 - idea of a programmable network infrastructure (for customized services)
  - Approaches
    - (1) user-programmable switches, in-band data transfer and out-of-band management channels
    - (2) control information organized in "capsules", which were program fragments that could be carried in user messages; program fragments would then be interpreted and executed by routers.
  - No large scale / significant success in practice - issues: security and perf.

- **4D Project** [5]
  - ~2004, a clean slate design
  - separation between the routing decision logic and the protocols governing the interaction between network elements.
  - "decision" plane having a global view of the network,
  - serviced by a "dissemination" and "discovery" plane, for control of a "data plane" forwarding
  - Consequences: NOX = operating system for networks in OF context

# 1. Software Defined Networking

- **1.2 Earlier technologies related to SDN (cont'd)**
- **NETCONF,** [6], 2006
  - IETF Network Configuration WG (still active) : NETCONF defined a management protocol for modifying the configuration of network devices.
    - network devices have APIs - (to send /retrieve) configuration data
  - still - no separation Control/Data Plane
  - A network with NETCONF is not fully programmable (new functionality should be implemented at both the network device and the manager
  - NETCONF primarily aid automated configuration and not for enabling direct control of state data
  - It can be used in parallel on hybrid switches supporting other solutions that enable programmable networking

- **Ethane,** [8], 2006- precursor to SDN
  - new network architecture for enterprise networks
  - centralized controller to manage policy and security in a network
  - two components:
    - a controller to decide if a packet should be forwarded
    - Ethane switch : a flow table and a secure channel to the controller

## 1. Software Defined Networking

- **1.2. Earlier technologies related to SDN (cont'd)**

- **IETF WG ForCES Forwarding and Control Element Seperaration, 2003, [7].**
  - A parallel approach to SDN
  - some common goals with SDN and ONF
  - Differences:
    - ForCES: the internal network device architecture is redefined as the control element separated from the forwarding element, but the combined entity is still represented as a single network element to the outside world
      - Aim: to combine new forwarding hardware with third-party control within a single network device where the separation is kept within close proximity (e.g., same box or room)
    - **SDN: Contrl Plane (CPI) is totally moved from net device**
  - FORCES published docs on : arch. framework, interactions, modelling language, forwarding  element (FE) functions, protocol between Ctrl and FE

**UPB Master Courses 2015-2016**

# References-0

1. N.McKeown, T.Anderson, et. Al., OpenFlow: Enabling Innovation în Campus Networks, - http://www.openflow.org/documents/openflow-wp-latest.pdf.
2. M.Mendonca, et. al., A Survey of Software-Defined Networking: Past, Present,and Future of Programmable Networks, http://hal.inria.fr/hal-00825087/
3. A. Campbell, I. Katzela, K. Miki, andJ. Vicente. Open signaling for atm, internet and mobile networks (opensig'98). ACM SIGCOMM Computer Communication Review, 29(1):97-108,1999.
4. D. Tennenhouse, et.al., A survey of active network research, Communications Magazine,IEEE, 35(1):80-86, 1997.
5. A. Greenberg, et. al., A clean slate 4d approach to network control and management. ACM SIGCOMM Computer Communication Review, 35(5):41-54, 2005.
6. R. Enns, NETCONF Conguration Protocol RFC 4741 (Proposed Standard), Dec. 2006, Obsoleted by RFC 6241.
7. A. Doria, et al., Forwarding and Control Element Separation (ForCES) Protocol Specification. RFC 5810 (Proposed Standard), Mar. 2010
8. M. Casado, et al., Ethane: Taking control of the enterprise. ACM SIGCOMM Computer Communication Review, 37(4):1-12, 2007.
9. M. Casado, et.al., Fabric: a retrospective on evolving sdn, Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 85{90, New York, NY, USA, 2012. ACM.
10. Open networking foundation, https://www.opennetworking.org/about.
11. Open Networking Research Center (ONRC). , http://onrc.net.
12. Open vswitch and ovs-controller, http://openvswitch.org/.
13. Pantou: Openow 1.0 for openwrt, http://www.openow.org/wk/index.php/

14. OpenFlow switch specification v1.0. , http://openflow.org/wp/documents/
15. OpenFlow Switch Specification, V 1.3.0 (Wire Protocol 0x04 ) June 25, 2012
16. HP SDN/Openflow Technology Solutions:
    http://h17007.www1.hp.com/us/en/solutions/technology/openflow/index
17. SDN Controller Product Fact Sheet:
    http://h17007.www1.hp.com/docs/interopny/4AA4-3881ENW.PDF
18. SDN for cloud providers and enterprises:
    http://h17007.www1.hp.com/docs/interopny/4AA4-3872ENW.pdf
19. SDN Technical White Paper http://h17007.www1.hp.com/docs/interopny/4AA4-3871ENW.pdf
20. Software-Defined Networking: The New Norm for Networks ONF White Paper April 13, 2012
21. K. Yap, et.al., Blueprint for introducing innovation into wireless mobile networks, proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures, pages 25-32. ACM, 2010.
22. L. Suresh, et.al., Towards programmable enterprise wlans with odin, Proceedings of the rst workshop on Hot topics inSDN , HotSDN '12, pages115-120, New York, NY, USA, 2012. ACM
23. M. Bansal, et. al.,Openradio: a programmable wireless dataplane, Proceedings of the rst workshop on Hot topics in software dened networks, pages 109-114, ACM, 2012
24. SDN: the service provider perspective, Ericsson Review, February 21, 2013
25. K. Calvert, et.al., Instrumenting home networks. ACM SIGCOMM Computer Communication Review, 41(1):84-89, 2011

26.   N. Feamster. Outsourcing home network security, Proceedings of the 2010 ACM SIGCOMM workshop on Home networks, pages 37-42. ACM, 2010.

27.   R. Mortier,et.al., Control and understanding: Owning your home network, communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on, pages 1-10. IEEE, 2012.

28.   S. Mehdi,. Revisiting trac anomaly detection using SDN, Recent Advances in Intrusion Detection, pages 161-180. Springer, 2011.

29.   S. H.Yeganeh, et.al., On Scalability of Software-Defined Networking,  IEEE Comm. Magazine, February 2013

30.   A. Tootoonchian et al., "On Controller Performance in Software-Defined Networks," Proc. USENIX Hot-ICE '12, 2012, pp. 10–10

31.   M. Yu et al., "Scalable Flow-Based Networking with DIFANE," Proc. ACM SIGCOMM 2010 Conf., 2010, pp. 351–62.

32.   A. R. Curtis et al., "DevoFlow: Scaling Flow Management for High-Performance Networks," Proc. ACM SIGCOMM '11, 2011, pp. 254–65.

33.   T. Koponen et al., "Onix: A Distributed Control Platform for Large-Scale Production Networks," Proc. 9th USENIX, OSDI Conf., 2010.

34.   S. H.Yeganeh and Y. Ganjali, "Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications," Proc. HotSDN '12 Wksp., 2012.

35.   A. Tootoonchian et.al., "Hyperflow: A Distributed Control Plane for OpenFlow," Proc. 2010 INMConf., 2010.

36.   B. Raghavan, T, et.al., Software-Defined Internet Architecture: Decoupling Architecture from Infrastructure, Hotnets '12, October 29–30, 2012, Seattle, WA, USA

37.  Peter Mell , Timothy Grance,  The NIST Definition of Cloud Computing, Special Publication 800-145, Recommendations of the National Institute of Standards and Technology , 2011
38.   Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf, Recommendations of the National Institute of Standards and Technology,  NIST "Cloud Computing Reference Architecture", Special Publication 500-292 , 2011
39.  Jamil CHAWKI,  "Cloud Computing Standards: Overview and ITU-T positioning", ITU Workshop on "Cloud Computing" (Tunis, Tunisia, 18-19 June 2012)
40.  Marco CARUGI Cloud Computing technology in Telecommunication ecosystems and recent ITU-T standardization efforts, International Workshop "Innovative research directions in the field of telecommunications in the world" within ITU-ZNIIS ITTC joint project 21-22 July 2011, Moscow, Russia
41.  ITU-T  FG Cloud TR Part 2: Functional requirements and reference architecture
42.  W. Liu, J. Ren. J. Wang, IETF, Draft-icn-implementation-sdn-00, „A Unified Framework for Software-Defined Information-Centric Network" August 09, 2013
43.  J.Choi, Jinyoung Han, E.Cho, Ted Kwon, and Y.Choi,A Survey on Content-Oriented Networking for Efficient Content Delivery, IEEE Communications Magazine • March 2011
44.  T. Koponen et al., "A Data-Oriented (and Beyond) Network Architecture," SIGCOMM '07, 2007, pp. 181–92
45.  D. Kutscher, B.Ahlgren,  H.Karl, B. Ohlman, S.Oueslati  I.Solis, Information-Centric Networking— Dagstuhl Seminar — 2011
46.  A.Ghodsi, T.Koponen, B.Raghavan, S.Shenker, A.Singla, J.Wilcox, Information-Centric Networking: Seeing the Forest for the Trees, http://www.icsi.berkeley.edu/~barath/papers/icn-hotnets11.pdf

47. Operator Network Monetization Through OpenFlow-Enabled SDN, ONF Solution Brief, April 3, 2013, https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-network-monetization.pdf

48. G. Pavlou, Information-Centric Networking: Overview, Current State and Key Challenges, IEEE ISCC 2011 Keynote  http://www.ee.ucl.ac.uk/~gpavlou/

49. D. Kutscher, B.Ahlgren,  H.Karl, B. Ohlman, S.Oueslati  I.Solis, Information-Centric Networking— Dagstuhl Seminar — 2011

50. Van Jacobson, D.K. Smetters,  J.D. Thornton, M. F. Plass, NH. Briggs,  R.L. Braynard, Networking Named Content, Palo Alto Research Center, Palo Alto, CA, October 2009

51. H. Koumaras,et.al., "Media Ecosystems: A Novel Approach for Content-Awareness in Future Networks", Future Internet: Achievements and Promising Technology, Springer Verlag, pp369-380, May 2011

52. E.Borcoci (UPB), et.al., "A Novel Architecture for Multimedia Distribution based on Content-Aware Networking", Proc. of the 3rd Int'l. Conference on Communication Theory, Reliability, and Quality of Service (CTRQ 2010), Athens/Glyfada, Greece, June 13-19, 2010,

53. E.Borcoci, R.Iorga (UPB) "A Management Architecture for a Multi-domain Content-Aware Network", TEMU Conf, Int'l Conf. on Telecommunications and Multimedia, 14-16 July, 2010, Crete, Greece.

54. E.Borcoci, et. al., "Resource Management in Multi-Domain Content-Aware Networks for Multimedia Applications" , Int'l. Journal on Advances in Networks and Services", vol 5 no 1 & 2, year 2012, http://www.iariajournals.org/networks_and_services/