An Introduction to Software-Defined Networking (SDN)

Zhang Fu

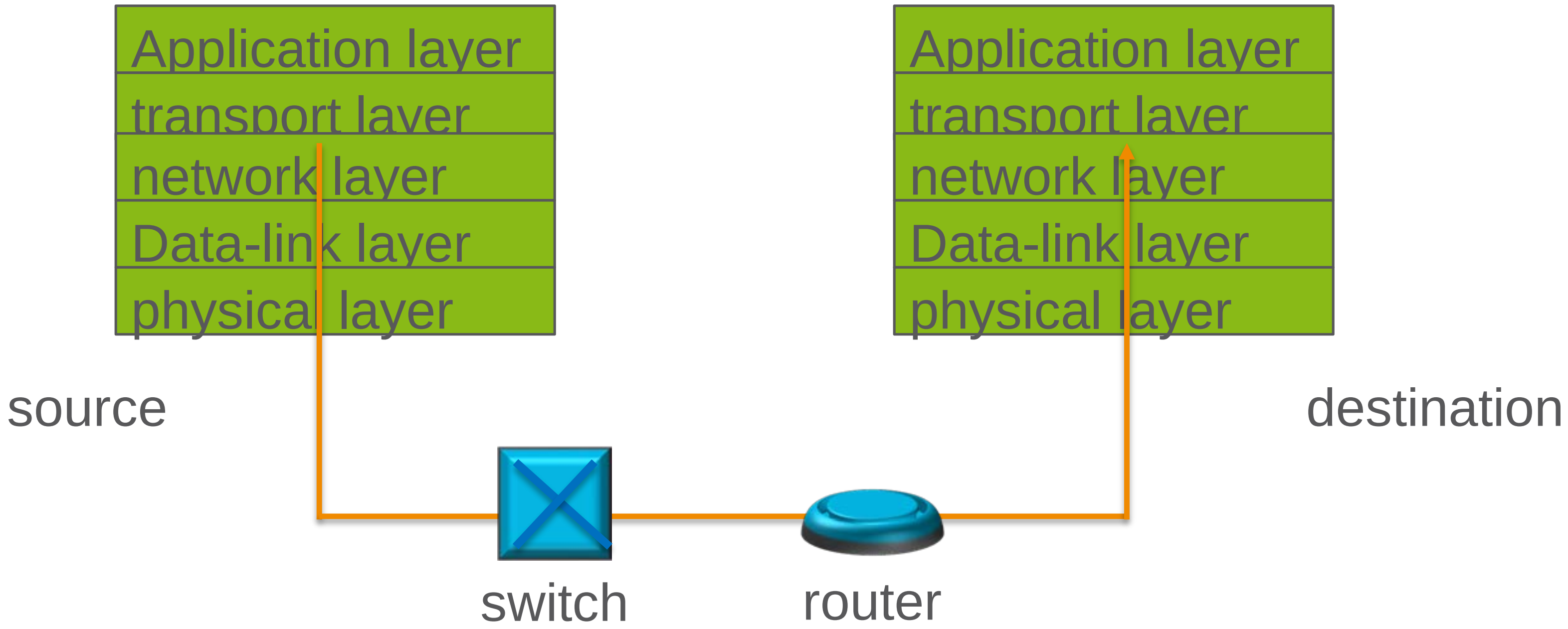
Ericsson Research
Feb 2016

Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little touch on Openflow

Reviewing traditional networking

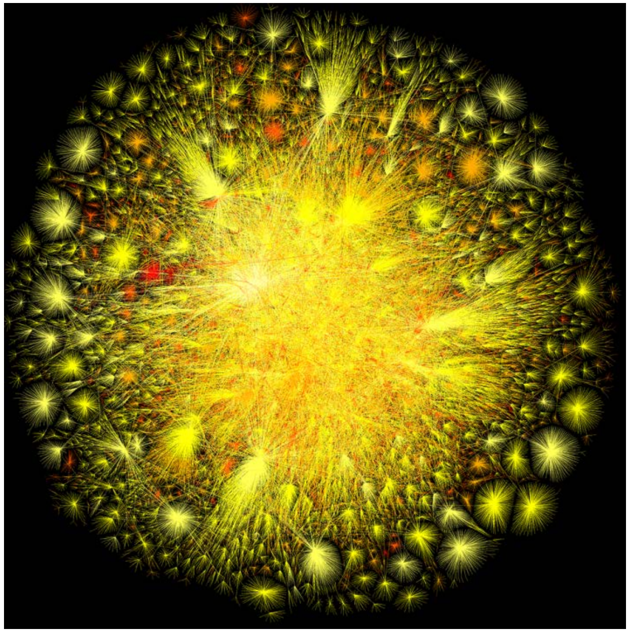
- Network layers



Why layers? Good abstraction, transparency…

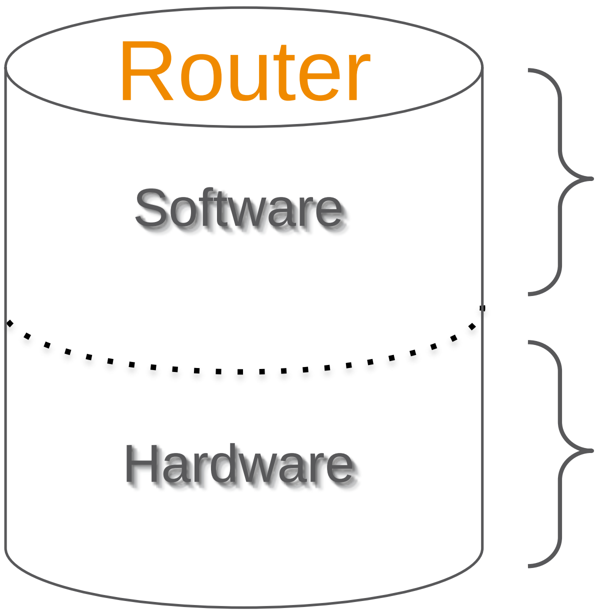
Reviewing traditional networking

- Design principles of Internet
  - Simple
  - Intelligent end-points
  - Distributed control
- Resulting in huge complex network and hard to manage
  - Billions of computers
  - Tens of thousands of ASes
  - Great business for selling routers



Reviewing traditional networking

› Complex routers



Million of lines
of source code

Hard to extend

Tens of billions of transistors

Power Hungry

10Gbytes RAM

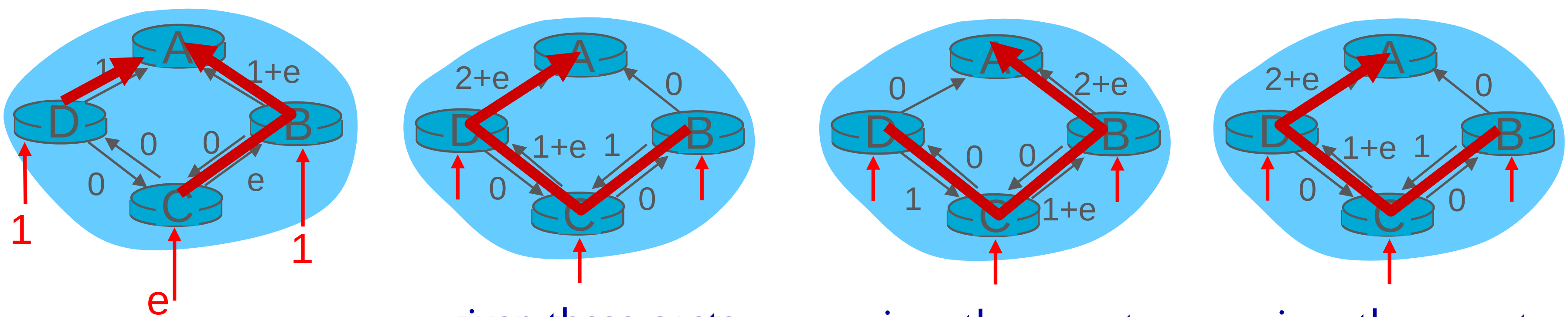
Vertically integration with many complex functions: *OSPF, BGP, multicast, QoS, Traffic Engineering, NAT, firewalls, MPLS…*

Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little bite of Openflow

Example: oscillation problem

› Link cost equals the amount of carried traffic



initially

given these costs,
find new routing….
resulting in new costs

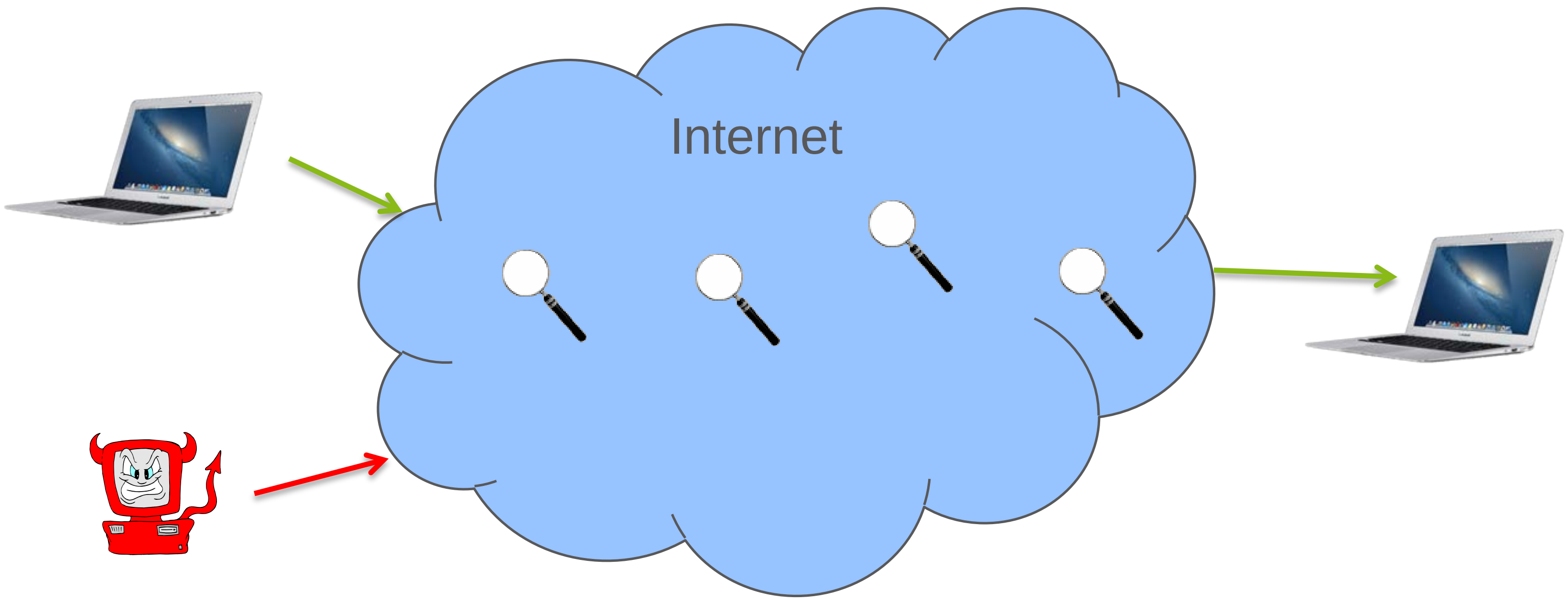
given these costs,
find new routing….
resulting in new costs

given these costs,
find new routing….
resulting in new costs

How to achieve optimal routing dynamically?

Example: mitigating attacks

› Checking the validity of packets by middle boxes

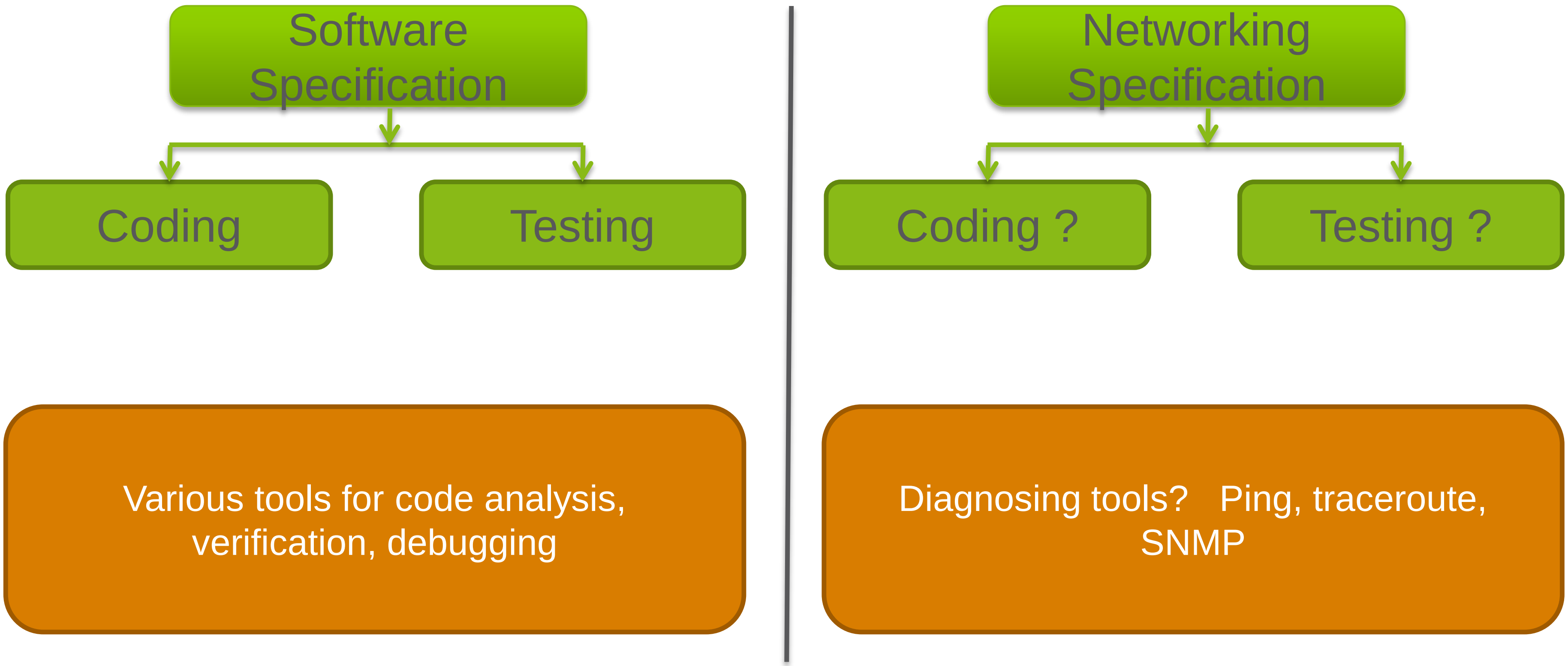


How to route the packets through a series of middle boxes?

Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little bite of Openflow

Software development VS Network diagnosing



- The life cycle for network protocols is much longer than that for software
- Timely research does not find its way into practice

Network substrate

› We want to mimic the success in software industry

– Has simple common substrate

– Building OS on top the hardware, which enables easy deployment of networking applications

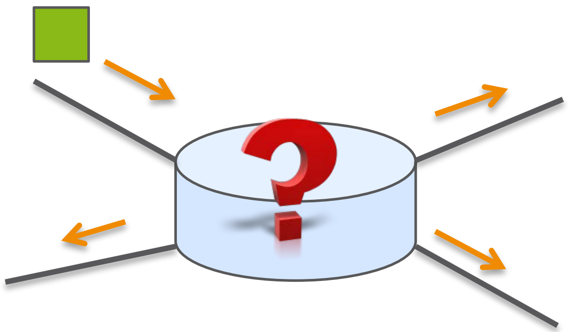
SDN

- A network in which the control plane is physically separate from the data plane.

- A single control plane controls several forwarding devices.

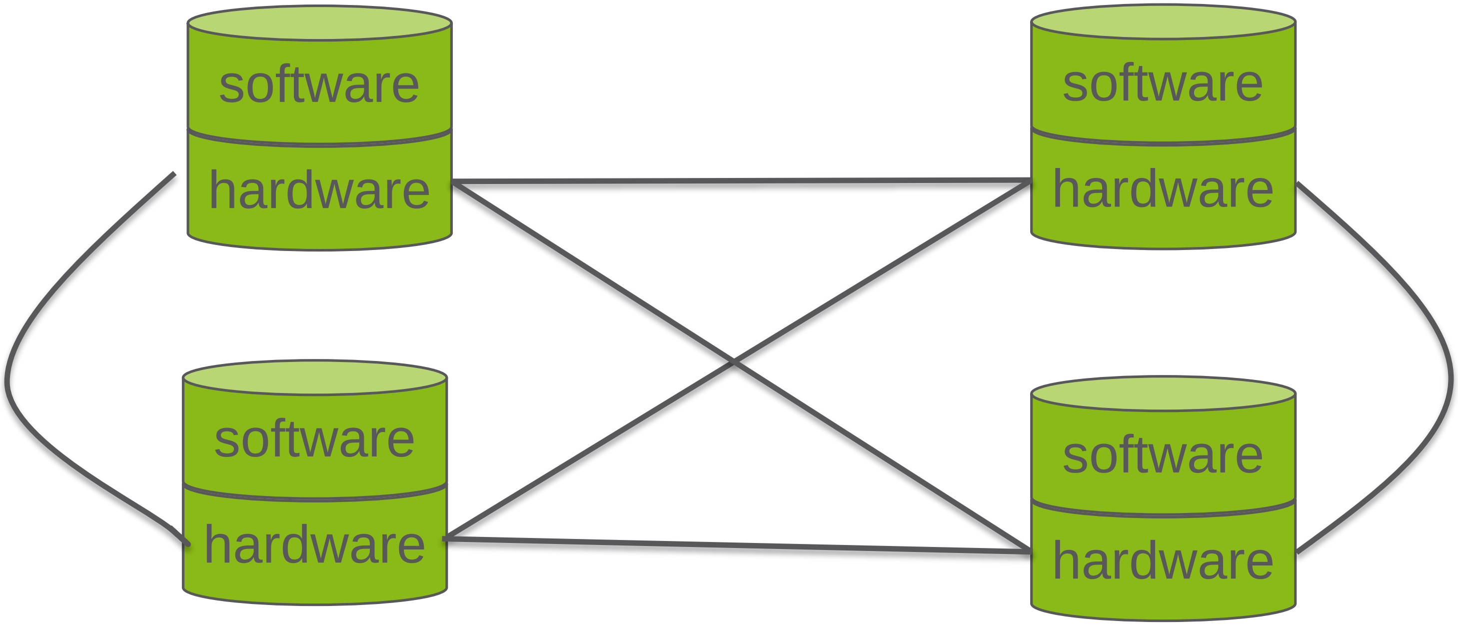
Network substrate

› Router Example

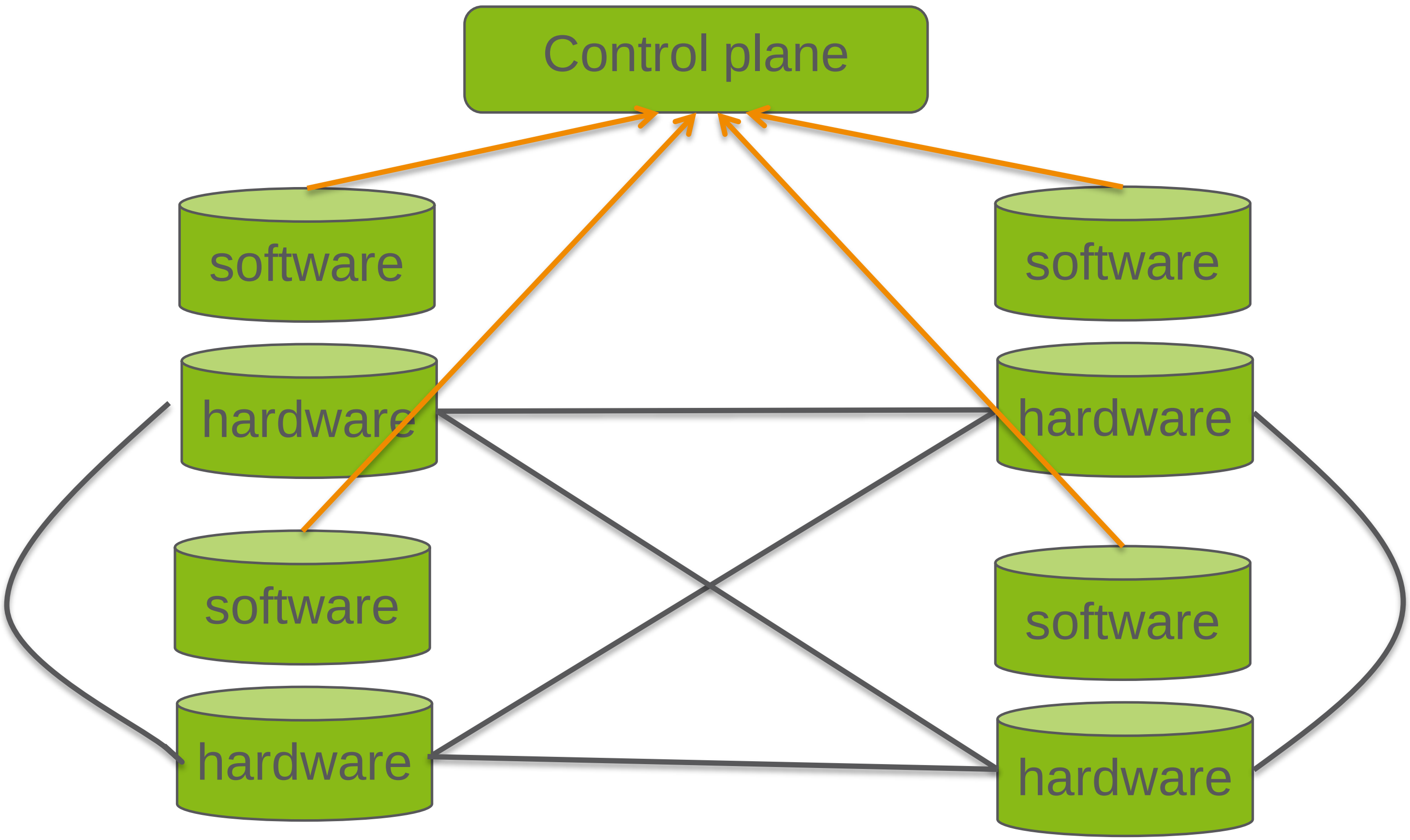


- Basic job of the router: *receiving packets, checking the routing table, forwarding the packets out*
- In order to build the routing table, the router has to understand BGP, OSPF, RIP, etc.
- What about getting the routing table from somewhere else?

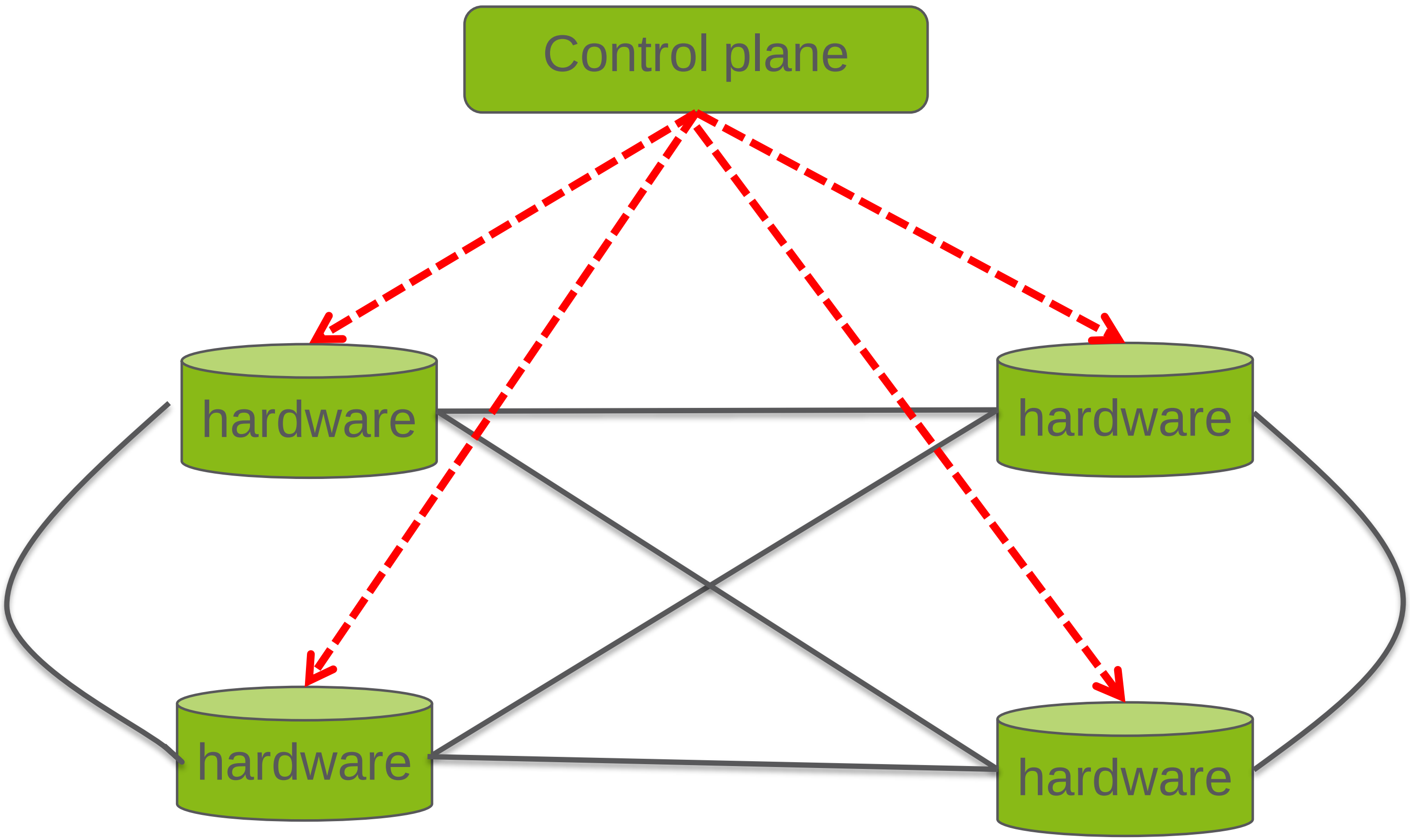
Separate data and control plane



Separate data and control plane



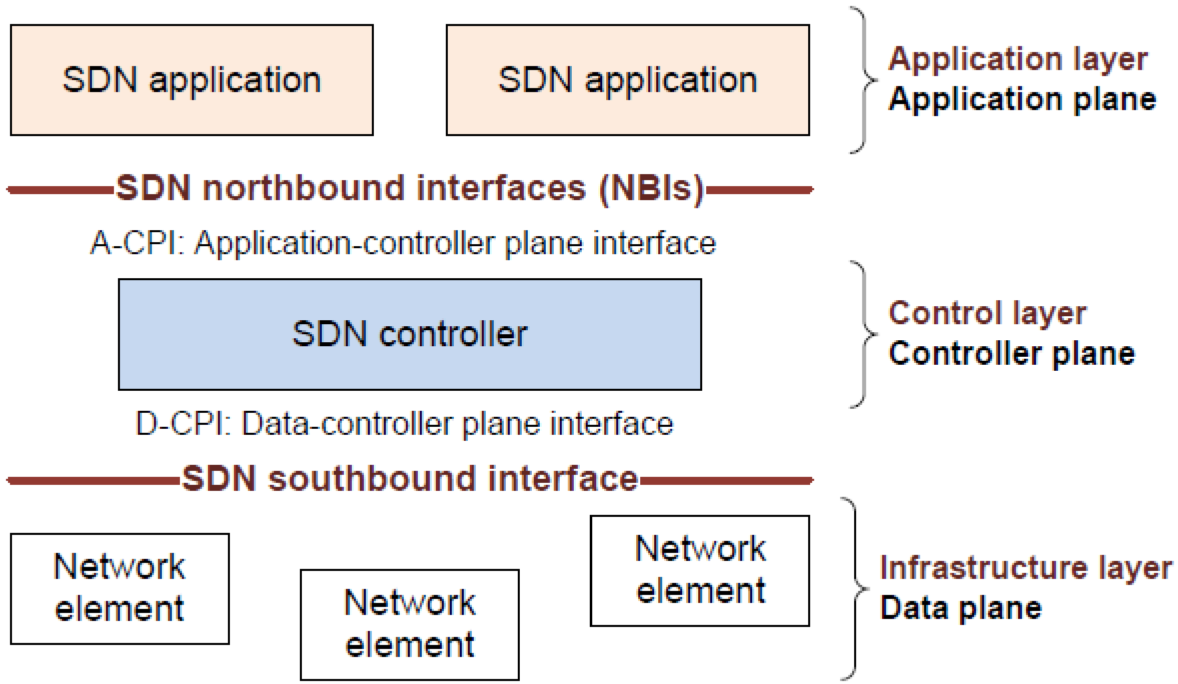
Separate data and control plane



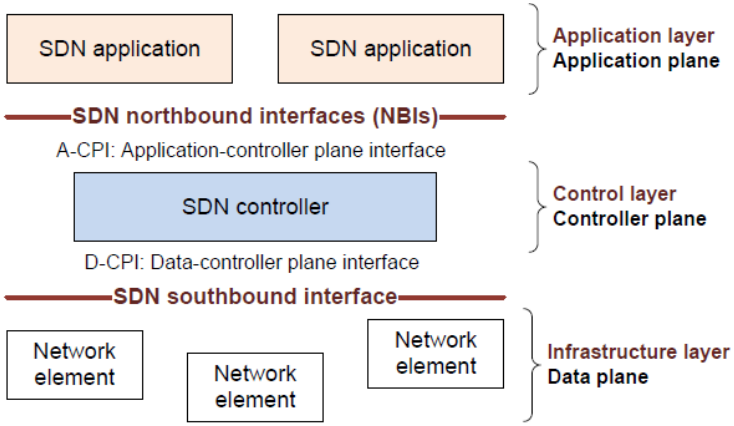
Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little bite of Openflow

SDN architecture



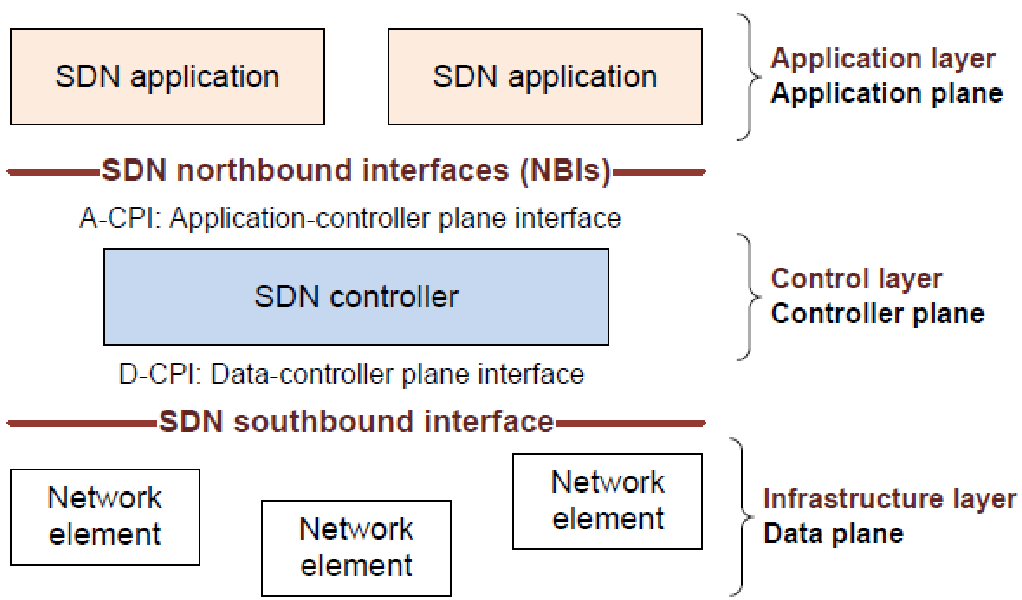
SDN architecture



- The data plane consists of network elements, which expose their capabilities to the control plane via southbound interface
- The SDN applications are in the application plane and communicate their network requirements toward the control plane via northbound interface
- The control plane sits in the middle
  - translate the applications' requirements and exerts low-level control over the network elements
  - Provide network information to the applications
  - Orchestrate different applications

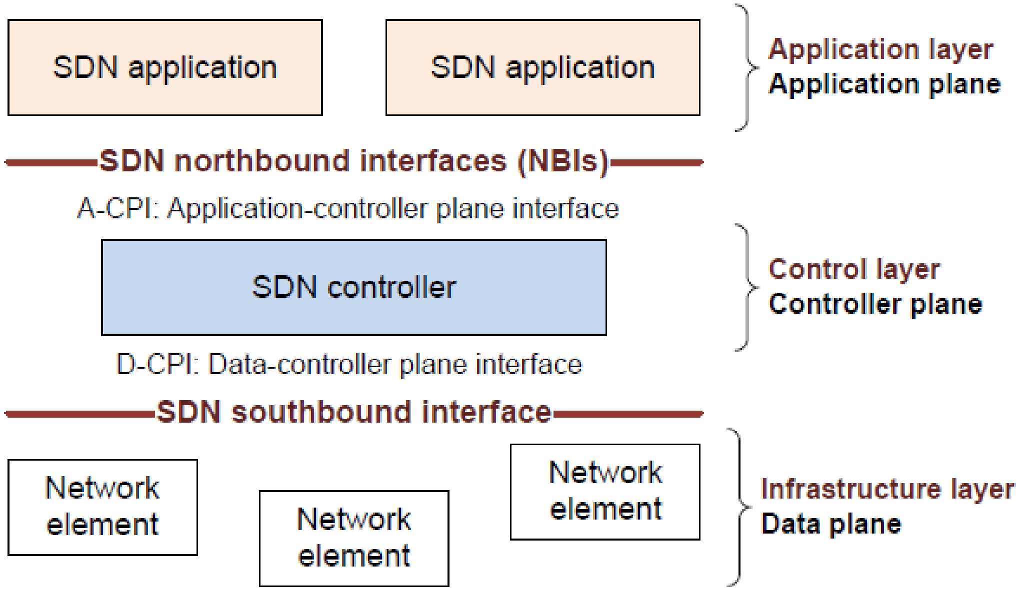
Data-plane

- Data sources and sinks
- Traffic forwarding/processing engine
  - May have the ability to handle some types of protocol, e.g. ARP, LLDP.
- Provide interfaces communicating to the control plane
  - Programmatic control of all functions offered by the network element
  - Capability advertisement
  - Event notification



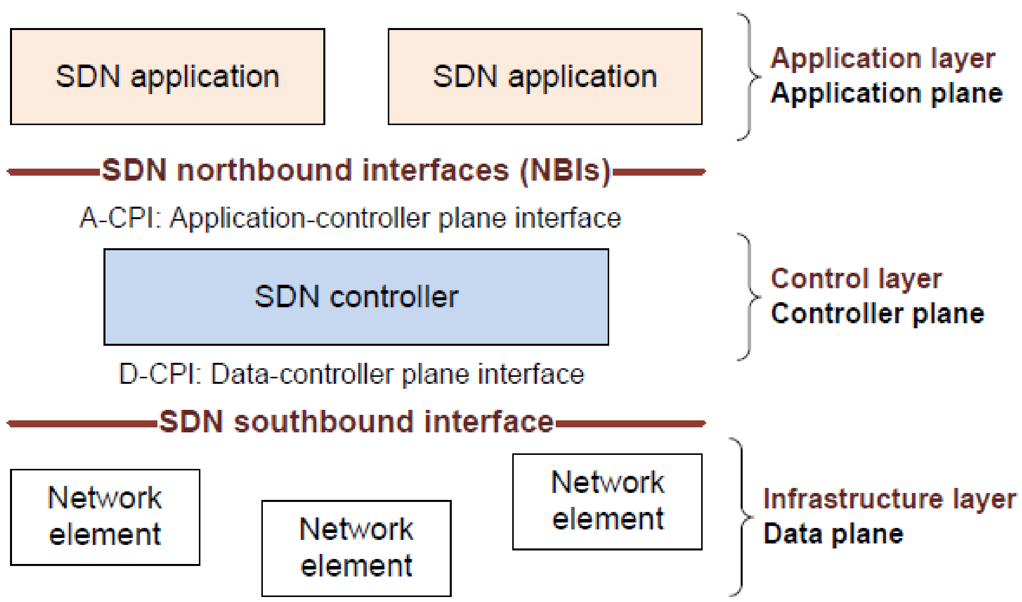
Control-plane

- Logically centralized
- Core functionality
  - Topology and network state information
  - Device discovery
  - Path computation
  - Security mechanism
- Coordination among different controllers
- Interfaces to the application plane



Application-plane

- Applications specify the resources and behaviors required from the network, with the context of business and policy agreement
- It may need to orchestrate multiple-controllers to achieve the objectives
- Programming languages help developing applications, e.g. Pyretic, FatTire,etc.



Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little bite of Openflow

Use cases

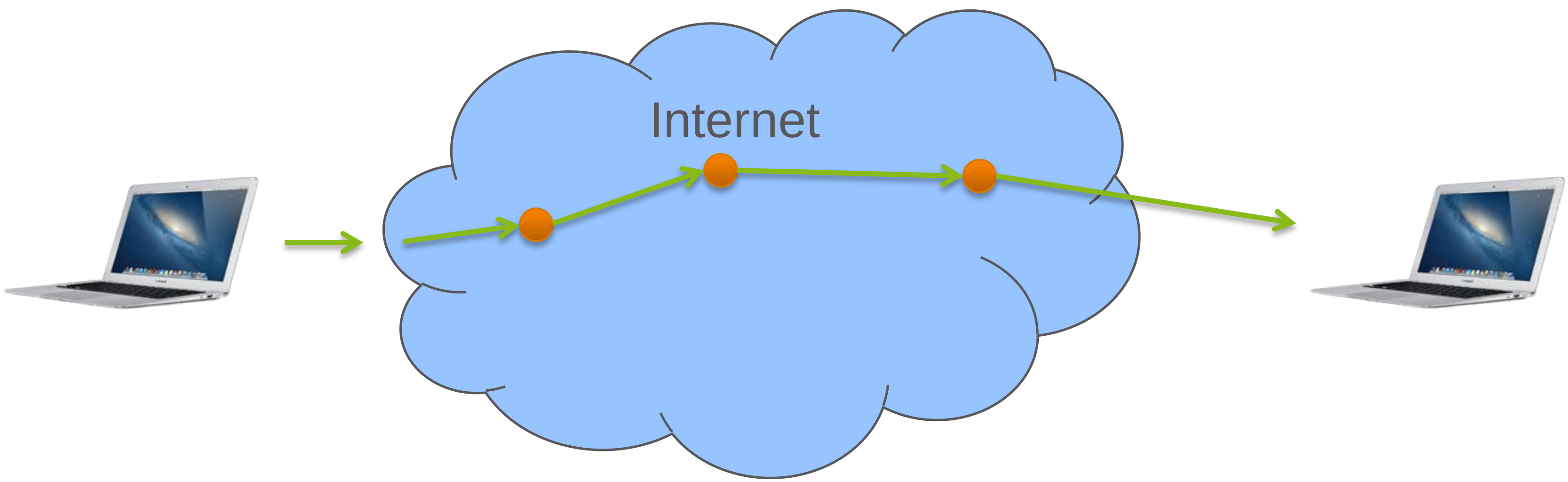
- Traffic engineering
  - Avoid congestion
  - Adaptive to different policies, QoS
- Mobility and wireless
  - Seamless mobility
  - SDN based Core network
- Security
  - Packets going through a set checking boxes
- Data center networking
  - Enhancing link utilization
  - Saving energy

Example: routing

App: build path
<a,b,QoS>

Controller:
Find path <a,b>,
set QoS requirements,
Commands to devices

Devices:
Configure hardware,
Installing forwarding rules



Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little bite of Openflow

Challenges and research problems

- Switch design
  - Find common abstraction
  - Flow table capacity
  - Throughput
- Controller platform
  - Distributed vs centralized
  - Flexibility
- Dependability and security
  - Attack to data plane
  - Attack to control plane
  - Trust, privacy issues

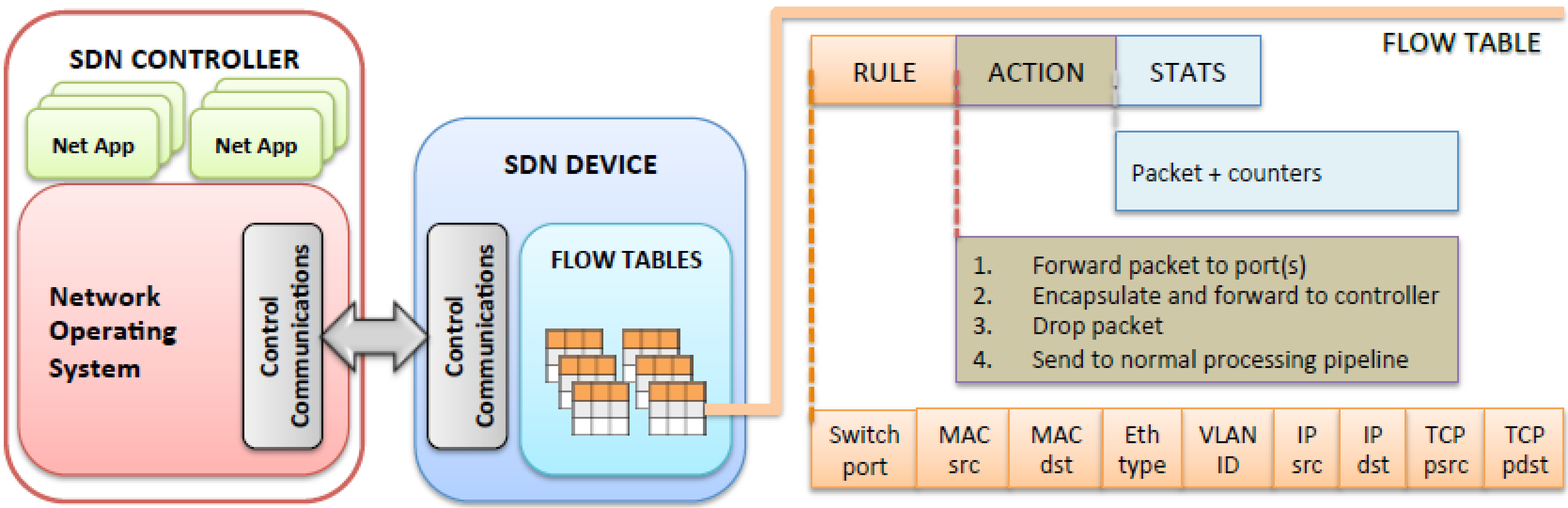
Roadmap

- Reviewing traditional networking
- Examples for motivating SDN
- Enabling networking as developing softwares
- SDN architecture
- Use cases
- Challenges and research problems
- Little bite of Openflow

Openflow

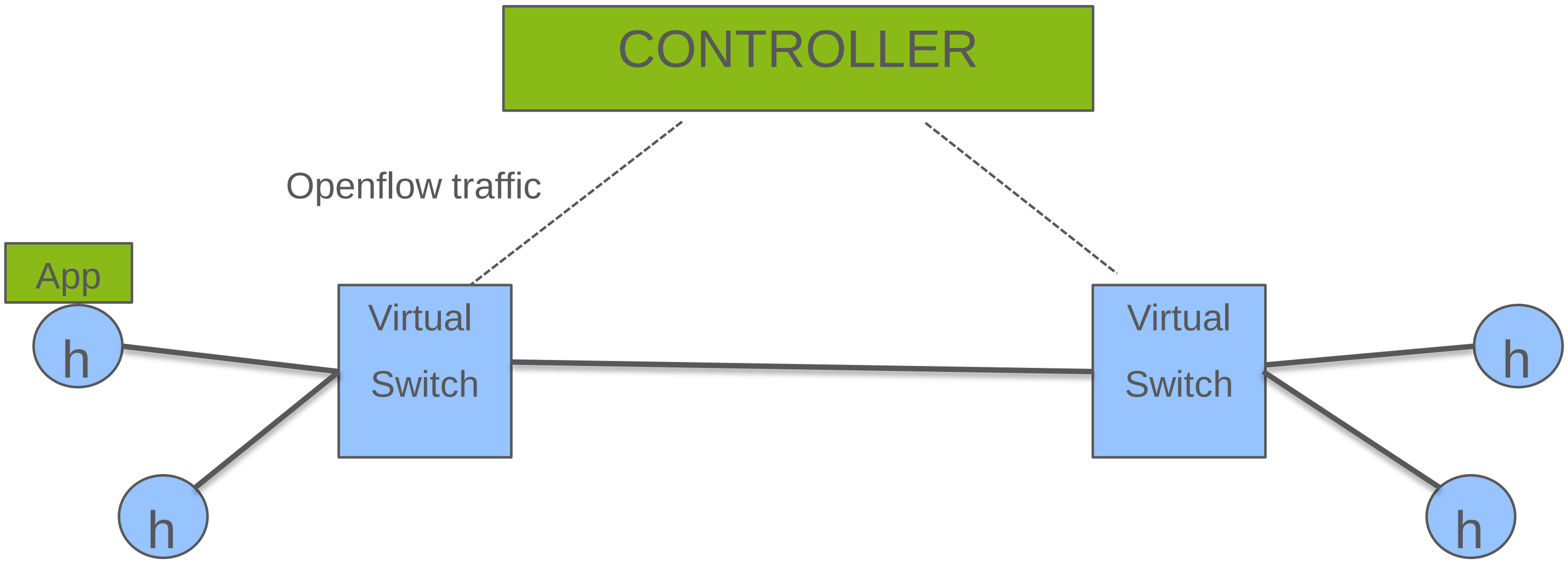
An southbound standard:

- Provide specification to implement Openflow-enabled forwarding devices
- Communication channel between data and control plane



Mininet

- Provide tools to create virtualized network with OVS
- CLI for manipulating network dynamically
- Virtualized hosts



```
fuzhang@fuzhangVM:~/mininet$ sudo mn
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:

mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=38.192s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=38,
priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=d6:13:79:41:63:43,dl_dst=7e:6c:76:0d:89:c9,nw_src=10.0.0.2,nw_dst=10
.0.0.1,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:1
 cookie=0x0, duration=38.190s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=38,
priority=65535,icmp,in_port=2,vlan_tci=0x0000,dl_src=d6:13:79:41:63:43,dl_dst=7e:6c:76:0d:89:c9,nw_src=10.0.0.2,nw_dst=10
.0.0.1,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:1
 cookie=0x0, duration=38.189s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=38,
priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=7e:6c:76:0d:89:c9,dl_dst=d6:13:79:41:63:43,nw_src=10.0.0.1,nw_dst=10
.0.0.2,nw_tos=0,icmp_type=0,icmp_code=0 actions=output:2
 cookie=0x0, duration=38.192s, table=0, n_packets=1, n_bytes=98, idle_timeout=60, idle_age=38,
priority=65535,icmp,in_port=1,vlan_tci=0x0000,dl_src=7e:6c:76:0d:89:c9,dl_dst=d6:13:79:41:63:43,nw_src=10.0.0.1,nw_dst=10
.0.0.2,nw_tos=0,icmp_type=8,icmp_code=0 actions=output:2
 cookie=0x0, duration=33.190s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=33,
priority=65535,arp,in_port=1,vlan_tci=0x0000,dl_src=7e:6c:76:0d:89:c9,dl_dst=d6:13:79:41:63:43,arp_spa=10.0.0.1,arp_tpa=10
.0.0.2,arp_op=2 actions=output:2
 cookie=0x0, duration=38.193s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=38,
priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=d6:13:79:41:63:43,dl_dst=7e:6c:76:0d:89:c9,arp_spa=10.0.0.2,arp_tpa=10
.0.0.1,arp_op=2 actions=output:1
 cookie=0x0, duration=33.191s, table=0, n_packets=1, n_bytes=42, idle_timeout=60, idle_age=33,
priority=65535,arp,in_port=2,vlan_tci=0x0000,dl_src=d6:13:79:41:63:43,dl_dst=7e:6c:76:0d:89:c9,arp_spa=10.0.0.2,arp_tpa=10
.0.0.1,arp_op=1 actions=output:1
```

References

- "The road to SDN" http://queue.acm.org/detail.cfm?ref=rss&id=2560327
- Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey.*proceedings of the IEEE*, *103*(1), 14-76.
- “SDN Architecture 1.0 - Open Networking Foundation”, https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf
- Openflow specification v1.0, https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf
- Mininet, mininet.org
- POX controller, http://www.noxrepo.org/

ERICSSON