Conventional Networks

- Based on ideas ARPANET
- Network of networks
- IP protocol ad unifier

Packets routing based on *routing tables*

Routing tables updated with *routing protocols*:
Distance-Vector, Link-State and Inter-Domain

De-centralized control: robust but...

Do they support new technologies?
What if we need new approaches (network slicing)?

Sofware Define Networks

- Designed to make easier administration
- Centralized controller & global overview
- Separation between *data plane* and *control plane*

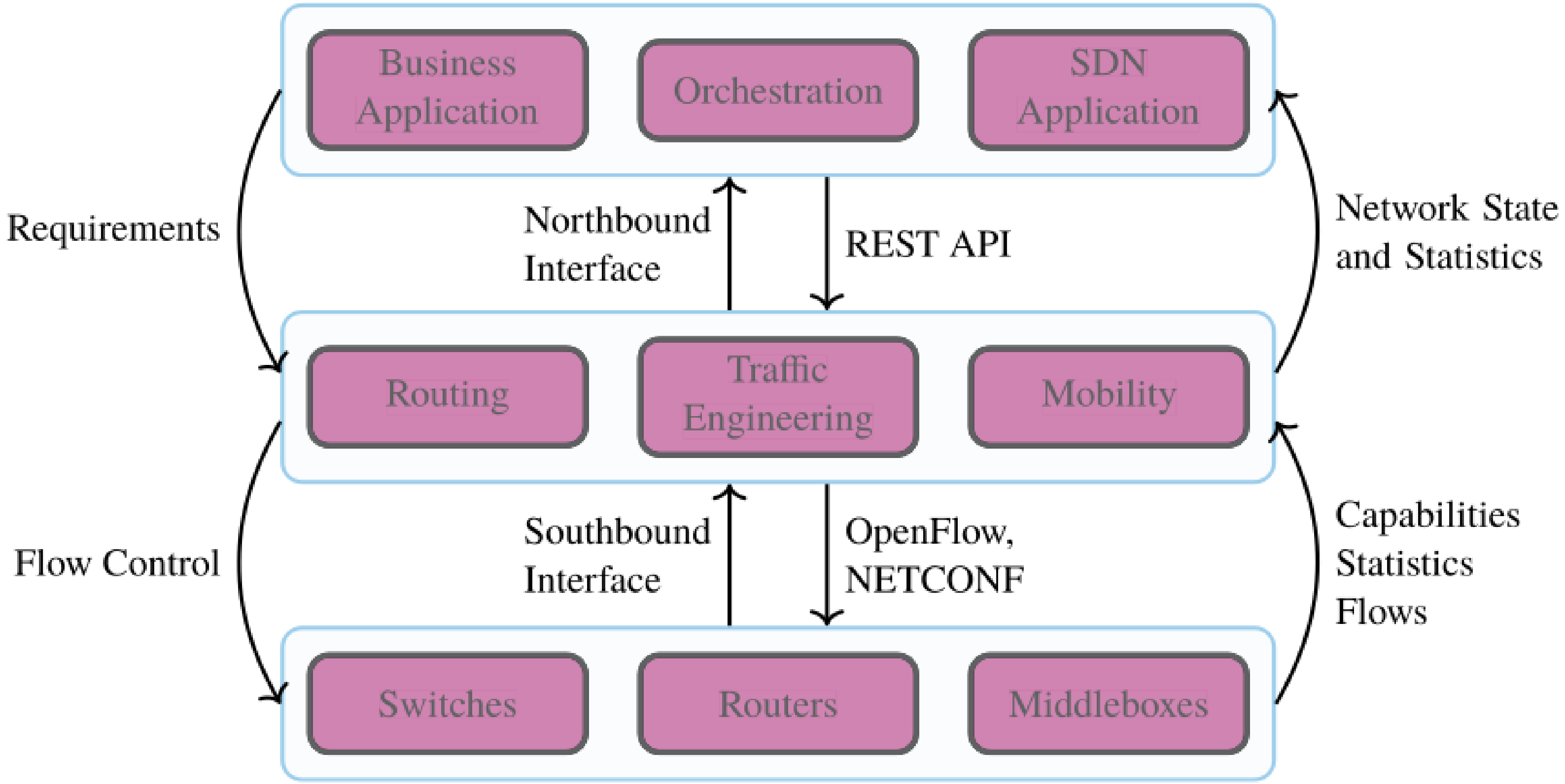
Components:

- centralized SDN controller
- SDN-capable switches
- a management protocol

Open and vendor neutral standards are needed

A new architecture

SDN architecture overview



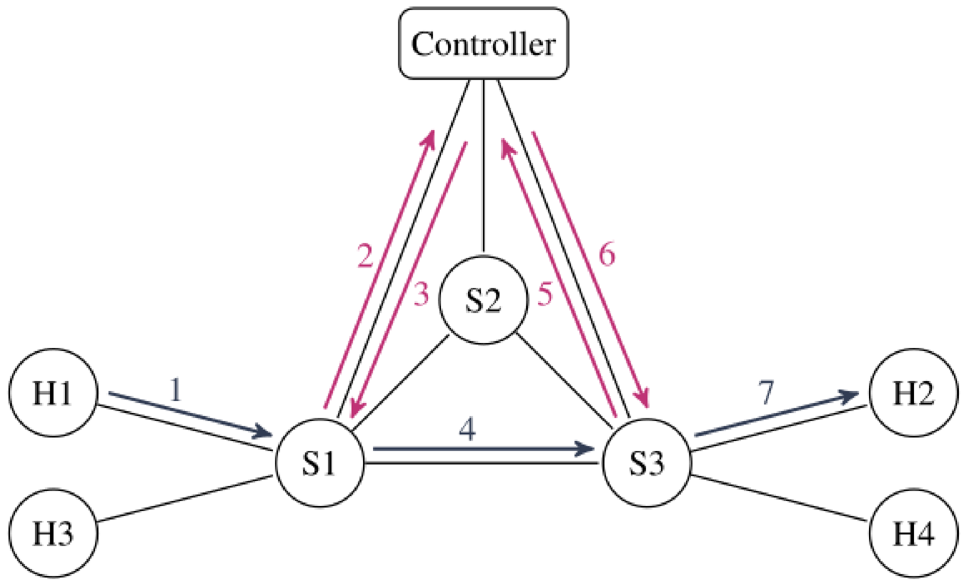
SDN Controllers

Controller	Domain	Configuration type
Nox	Research & Industry	C++ application
Pox	Research	Python application
Beacon	Research	Java *bundles*
Floodlight	Industry	REST API, Java modules
Opendaylight	Industry	REST API, YANG data modeling
Faucet	Industry	YAML-based configuration file
Ryu	Research & Industry	Python application

OpenFlow

OSI	Proto.	Action	Header	Example application
Layer 1	Output	Forw.	Port ID	Drop, flood, or forward packet
	Queue	Set	Queue ID	Bandwidth shaping
Layer 2	Ethernet	Set	VLAN ID	Manipulate VLAN tags
Layer 3	IPv4	Set	Src./Dst.	Network address translation
Layer 3	IPv4	Decr.	TTL	Decrement Time-To-Live
Layer 4	TCP	Set	Port	Port address translation

Workflow of OpenFlow



Flow Initialization

(1) a first packet is sent from H1

(2) switch asks directions

(3)* "forward packet to S3"

(4) packet is forwarded

(5,6)* similar procedure of (2,3)

(7) the packet arrives to H2

Flow Continuation

*** the rule is stored and repeated**

Credits (full text available form Unipd intranet)
https://www.sciencedirect.com/book/9780128204887/computing-in-communication-networks