



Introduction to Software Defined Networks

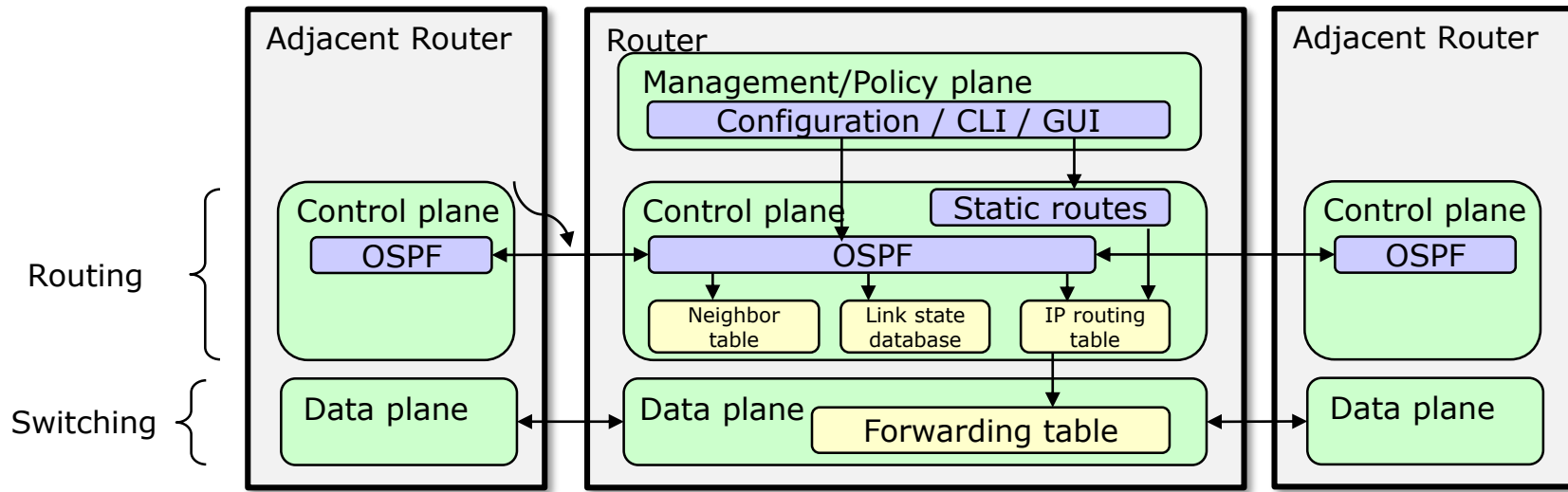
Fulvio Riso

Politecnico di Torino



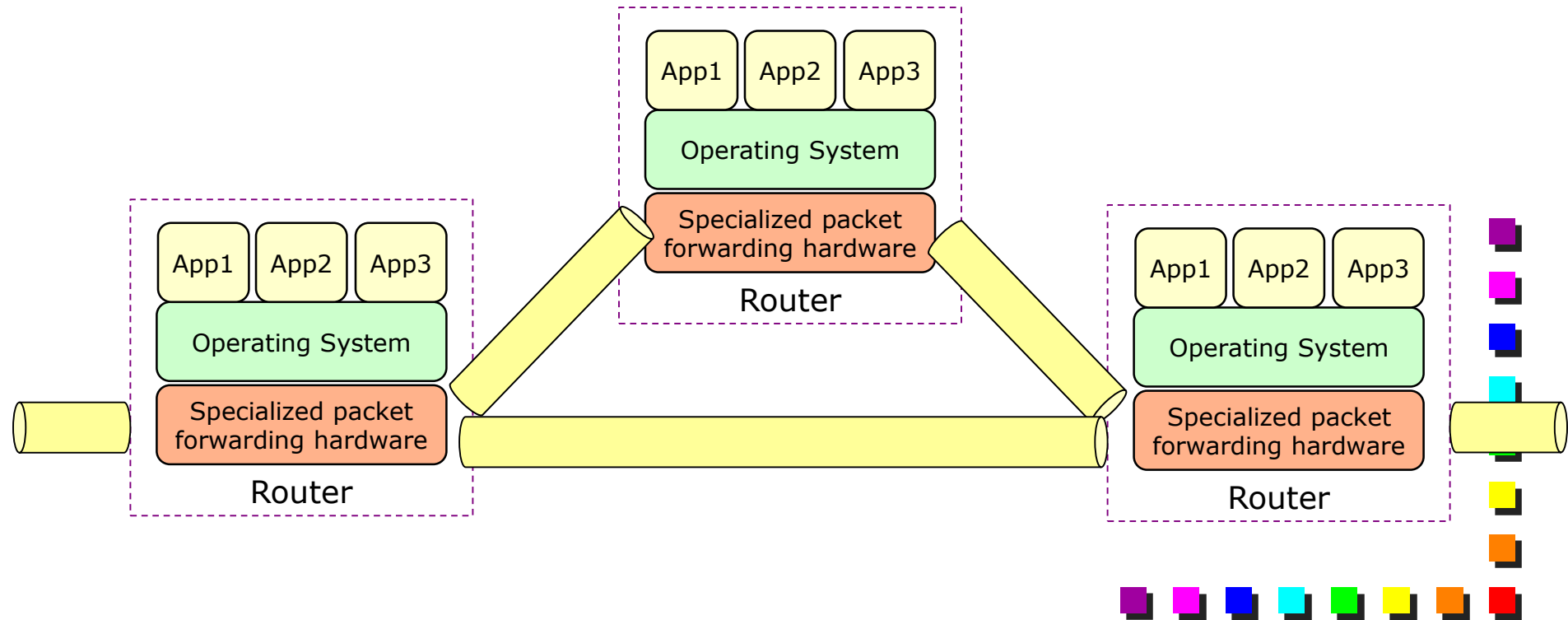
Traditional Network Router

- Router (but in general, any network device) can be partitioned into a **control** and **data plane**
 - Management plane: configuration
 - Control plane: decision (e.g., routing, OSPF)
 - Data plane: forwarding

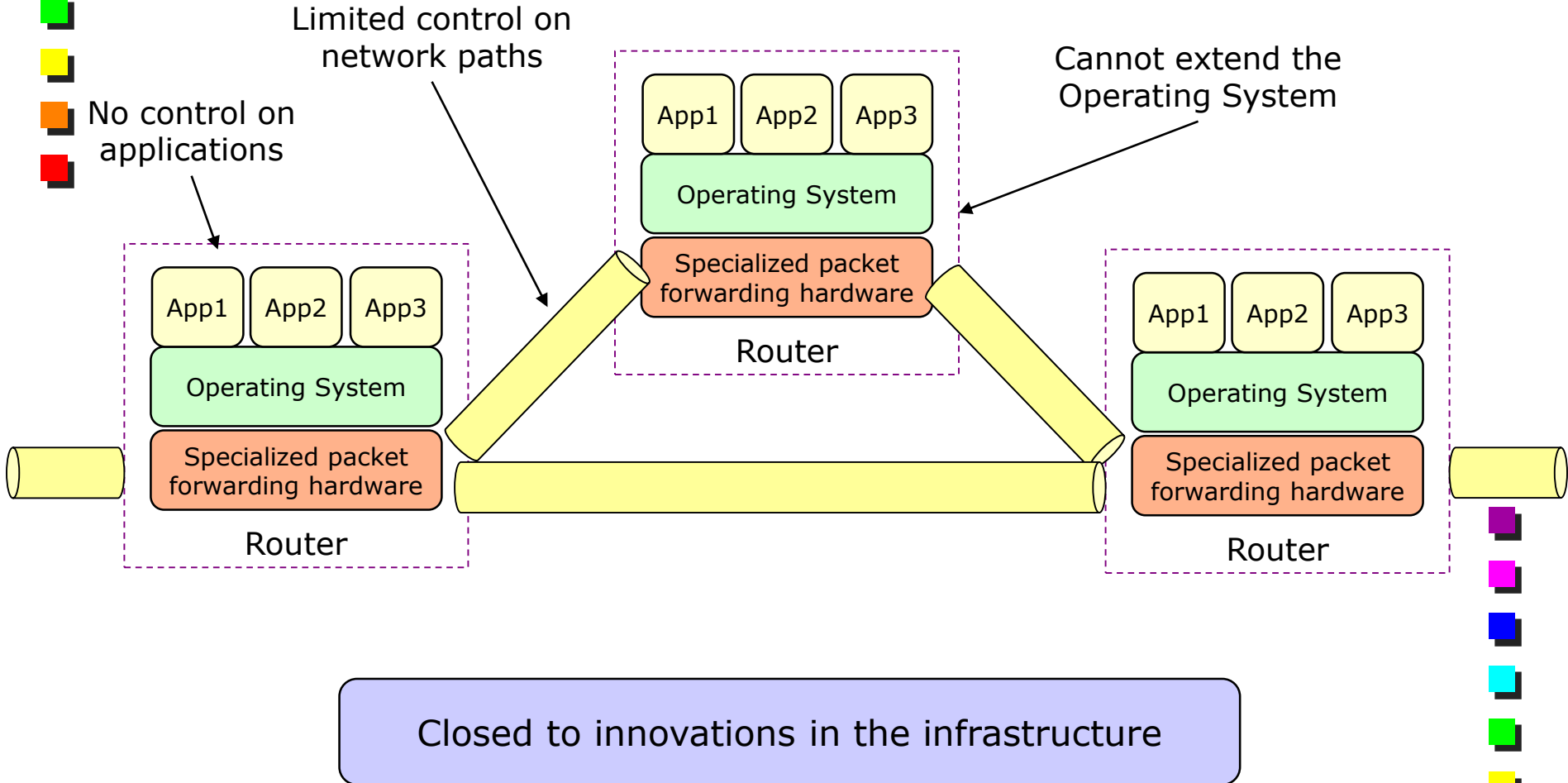


The Internet, after 30+ years

- Internet is still the one we defined 30 years ago
 - Almost the same protocols, same philosophy
- Internet is a very efficient **pipe** that transports **bits** at very **high speed**

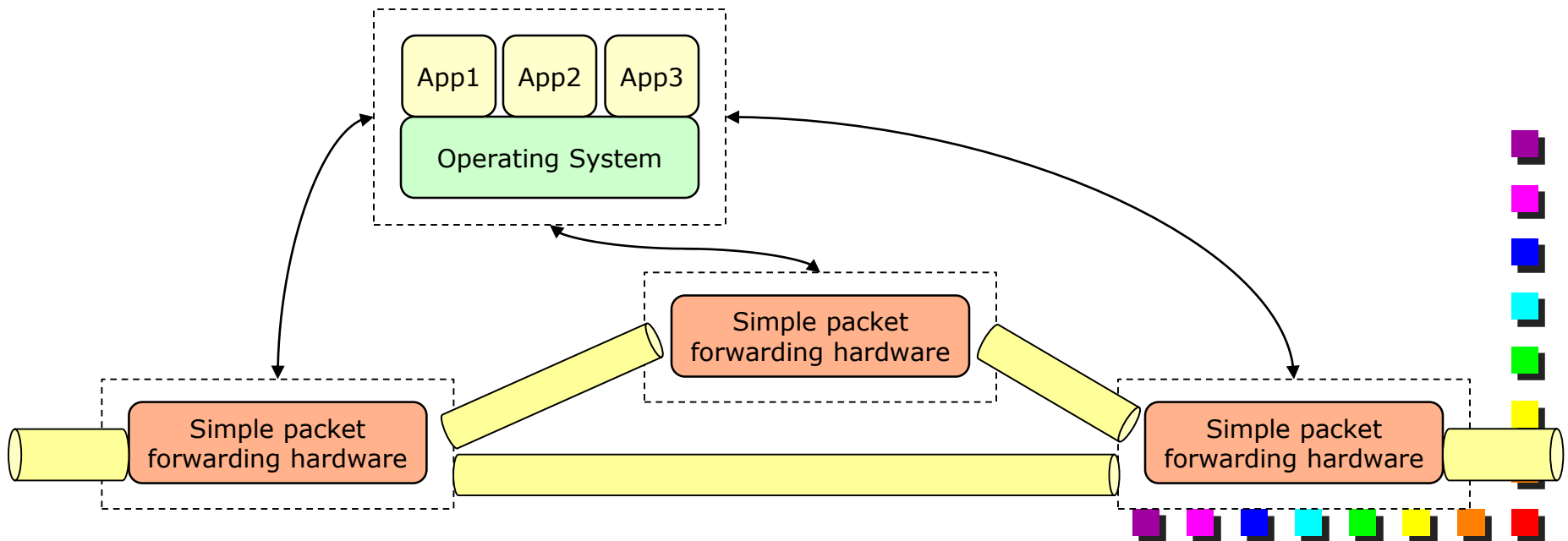


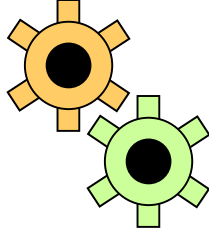
The Internet and the difficulties to innovate



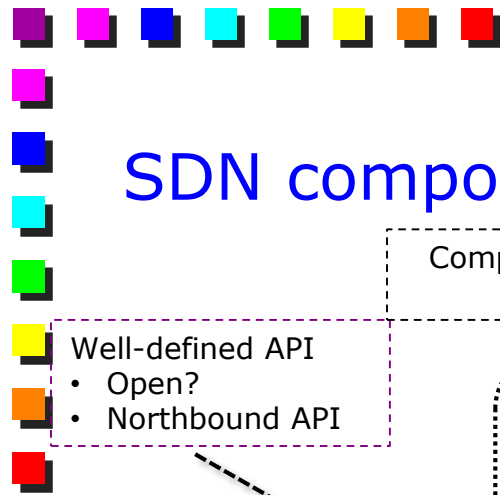
Software Defined Networks

- Paradigm that introduces the possibility to **program** the **network**
- Based on three pillars
 - Separation of control and forwarding functions
 - Centralization of control
 - Well-defined interfaces (northbound and southbound)





SDN components



Complex (user-defined) applications

Well-defined API

- Open?
- Northbound API

Firewall

WAN load balancer

"Routing" protocol

Operating System

- Extensible
- Open source?
- NetOS

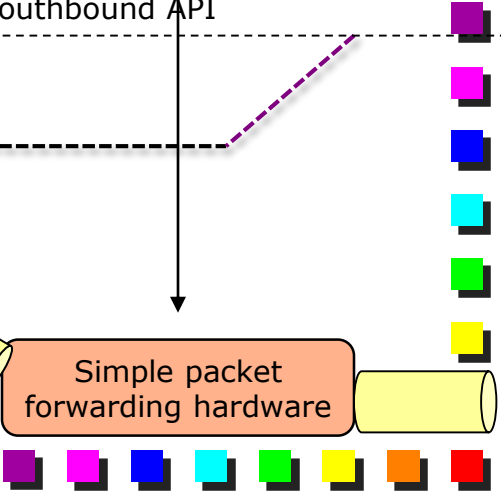
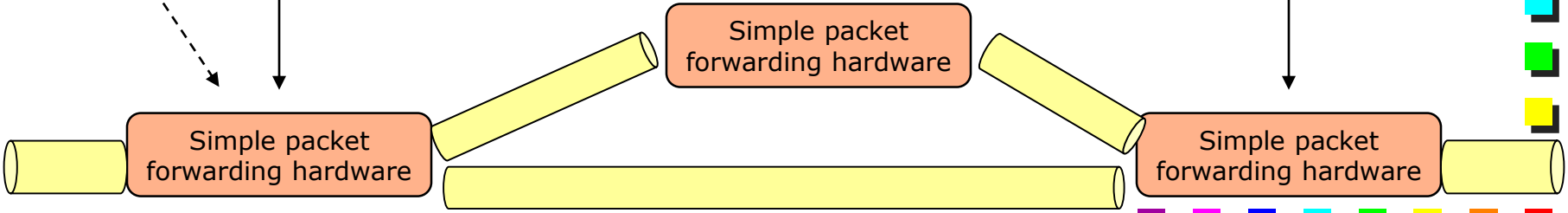
Operating System

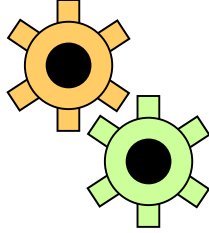
Execution environment for control applications (*controller*)

Open interface to the hardware

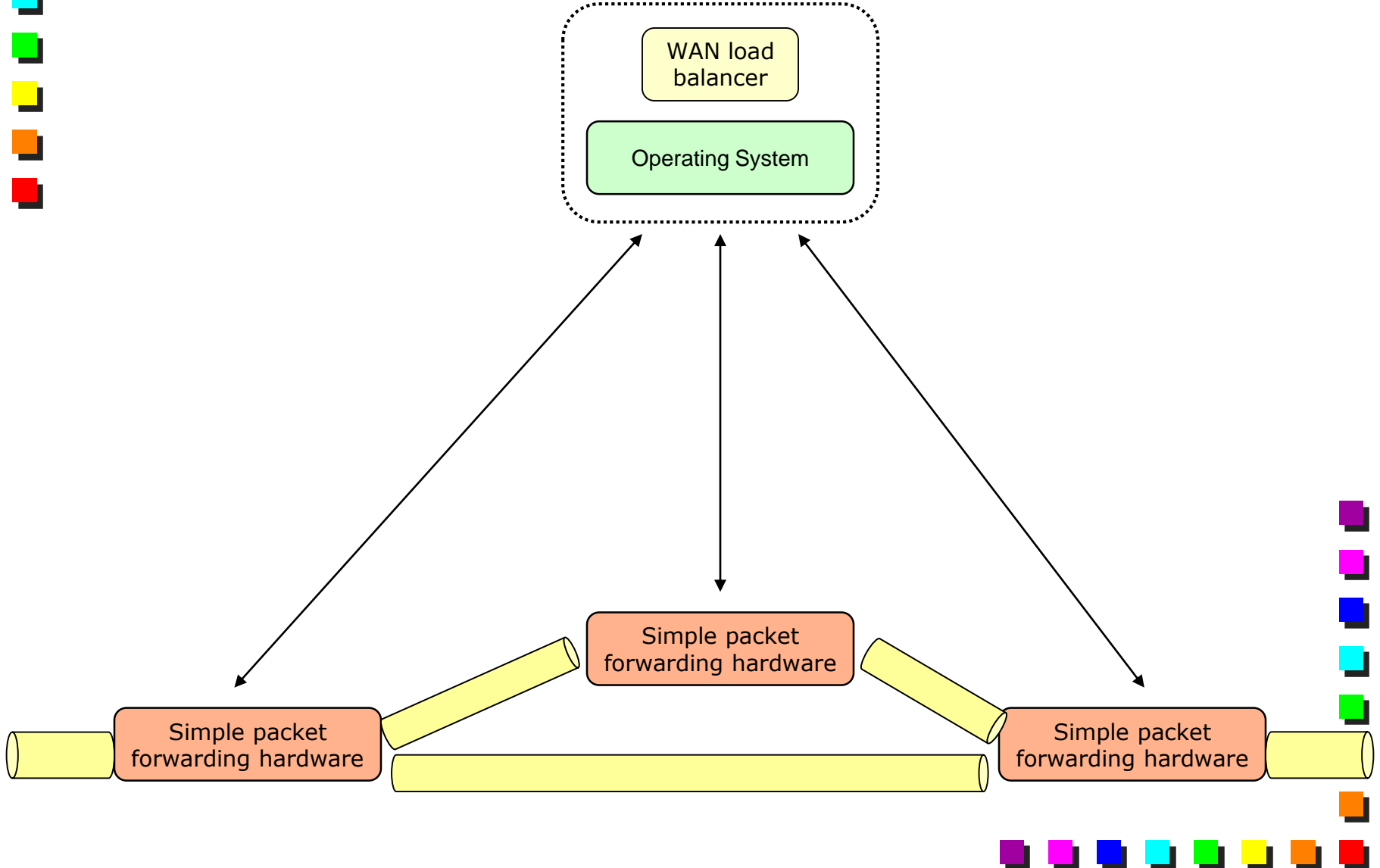
- Standard protocol between controller and data plane switches
- Southbound API

Simple forwarding switches (in principle, a lookup table)
Simpler, faster, cheaper



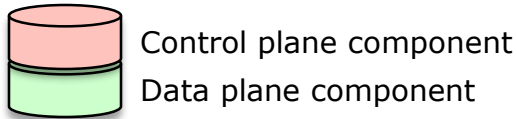
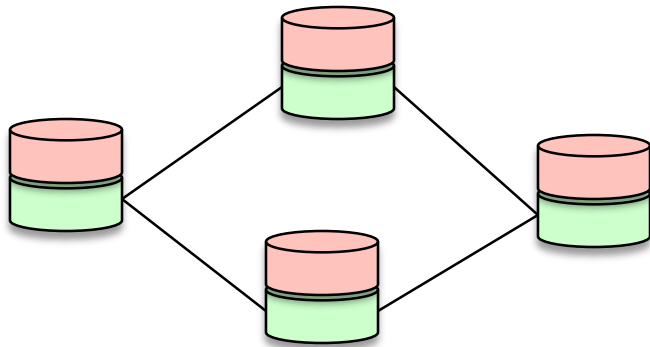


An SDN packet flow example

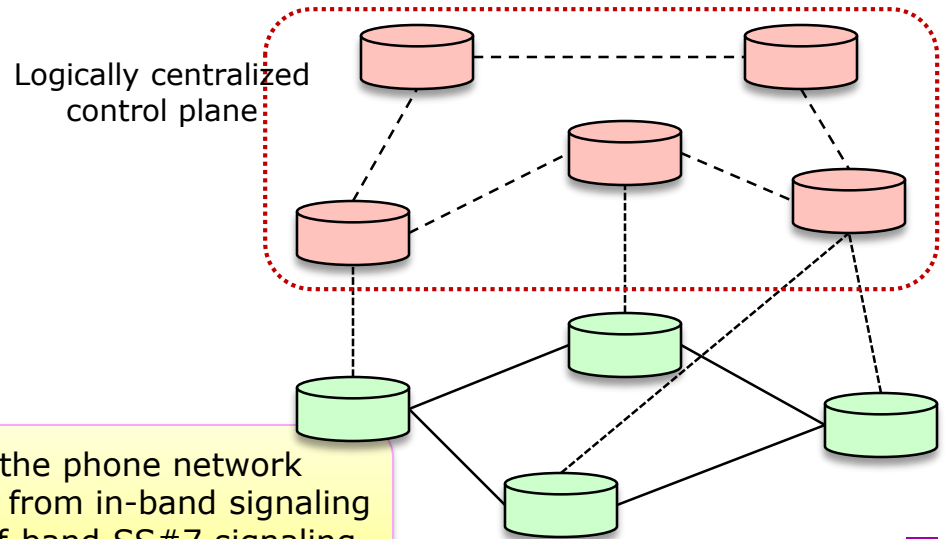


Software Defined Networks in essence

Traditional control plane architecture



Control plane architecture with SDN




Also the phone network migrated from in-band signaling to out-of-band SS#7 signaling some years ago

SDN is an approach to architecting the network control plane, where the behavior of the network is determined by software which is **logically separated** from the network devices.

This software could run in the network devices, a set/cluster of dedicated central server, application servers – or any combination of those.

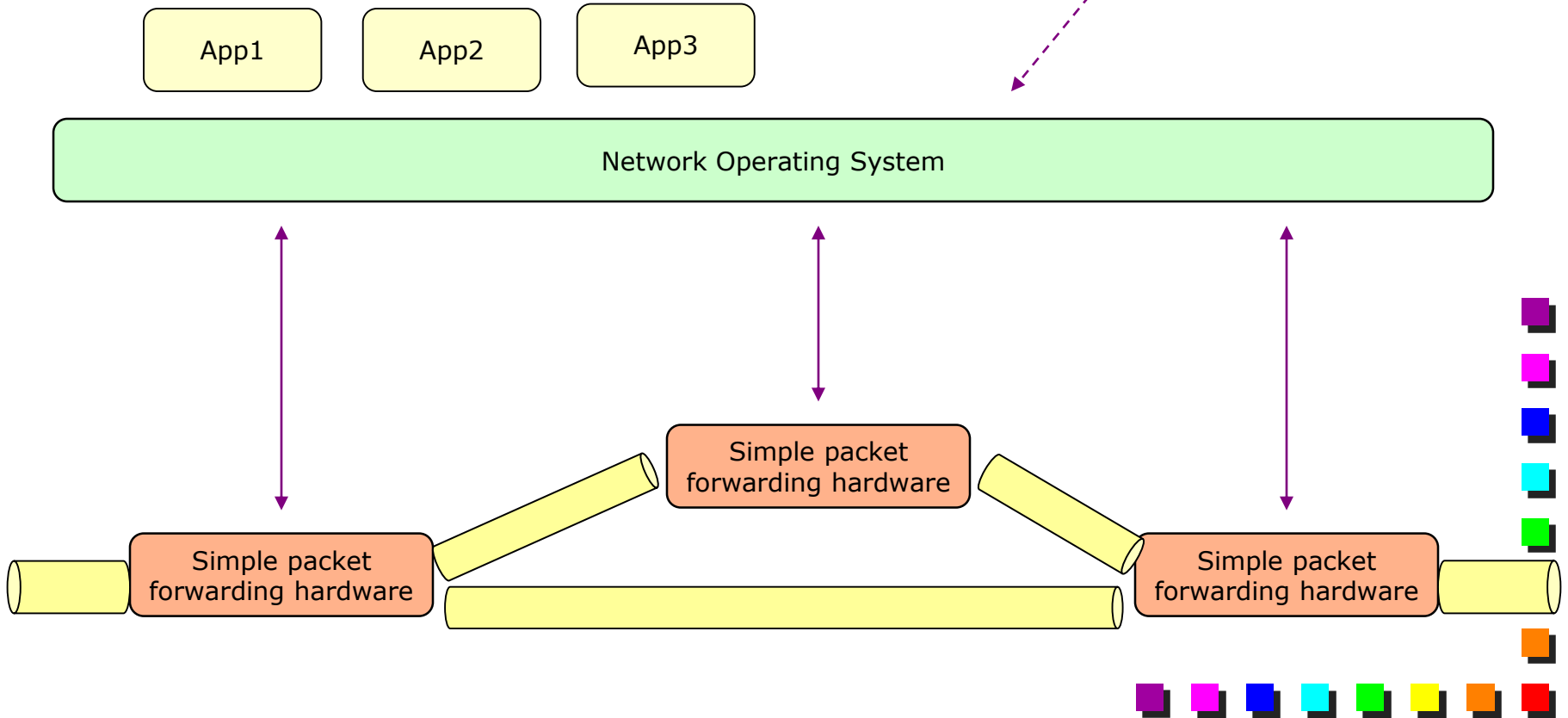


Software Defined Networks

- Software from third parties (e.g., network operators, residential customers, enterprise managers, datacenter operators) can be installed that can control network devices
 - What does “control” mean?
 - Currently, “control” means mainly “control plane”
 - We can implement new “routing” protocols (e.g., customize paths for network engineering)
 - Create network slices “private” to different entities (e.g., virtual network operators, application service providers such as video streaming, CDN, etc)
- 

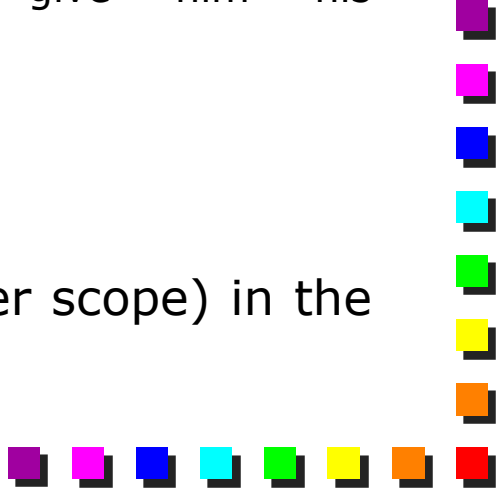
Network Operating System

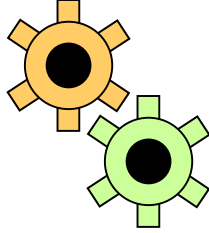
- Possible functions
- Synchronization between different physical controllers
 - Virtualization of the network topology
 - Single "logical" configuration transformed into multiple "physical" commands





Network Operating System

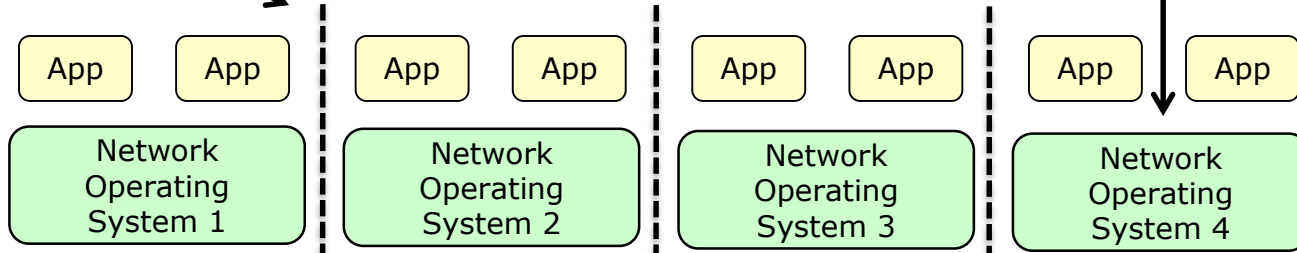
- Software layer that offers a “global” view of the network to upper applications
 - Right now, we have to configure each single device
 - VLAN, access lists, policies, QoS, ...
 - The configuration may be incoherent on different devices
 - A NetOS wants to give us the possibility to setup an application that operates across the whole network
 - E.g., check that a new user that connects to the network (from any port) is not infected; if so, give him his privileges/configuration
 - “Easy” to implement in a centralized controller
 - NetOS may have more features (and a broader scope) in the future
- 



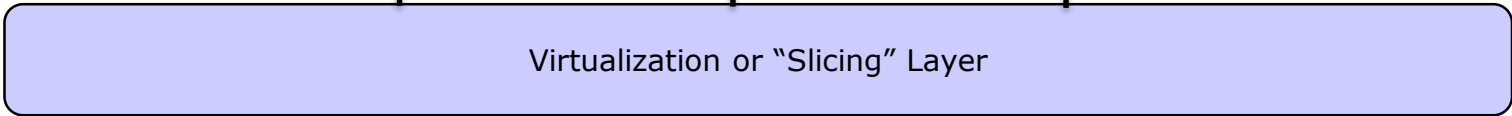
SDN and slices

Isolated "slices"

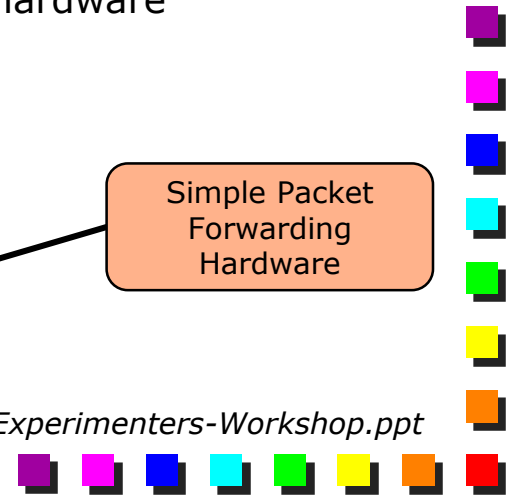
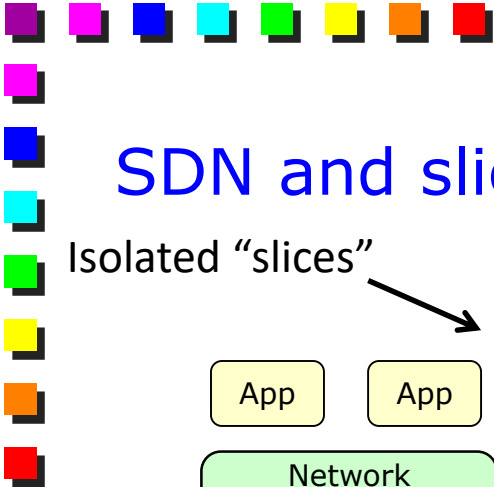
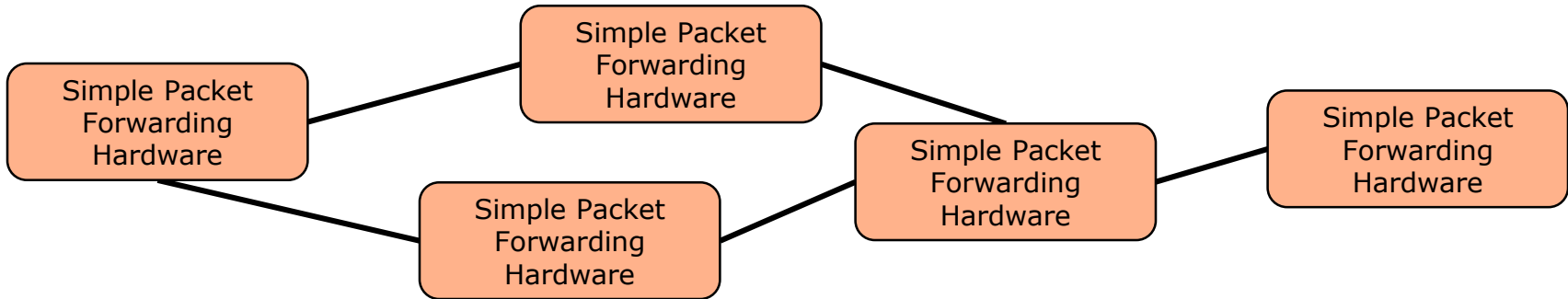
Many operating systems, or
Many versions



} Open interface to hardware



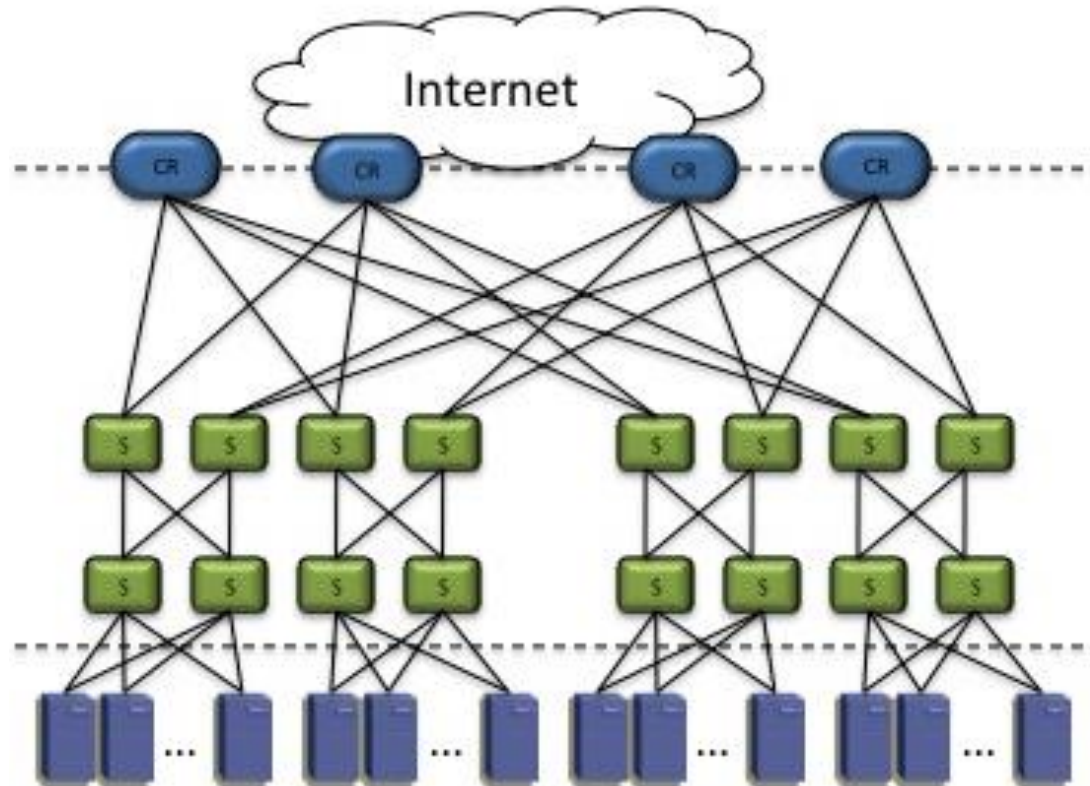
} Open interface to hardware





SDN applications

- Path optimization
 - Fat trees
- Traffic engineering
 - Per flow paths
- Virtual circuit emulation
 - VM migration
- Service chaining
 - (presented later)





SDN deployment

- Stable, local networks
 - Datacenters
 - Enterprise
- Edge services (possible)
 - Service chaining model
- No network service providers so far
 - Except Google





The good and the bad of SDN

Good

- Network virtualization (slicing)
 - E.g., virtual operators, VPNs with network (not only edge) support
- Network operating system
- Network API
- Network as an unique entity

Bad

- Simple packet forwarding hardware
 - Against the interest of major network vendors
 - Not scalable (hardware speedup needed)
 - Focus only on the *network*
- 