

# Programmable Networking: SDN Foundations

Raouf Boutaba

David R. Cheriton School of Computer Science  
University of Waterloo

Rogers TEP Workshop 3: Transport Networks  
Concepts #1

# SDN in a nutshell

- ▣ Three principles:
  - ▣ decoupling of control and data planes
  - ▣ implementing control plane in software as a (logically) centralized server
  - ▣ processing network traffic as flows

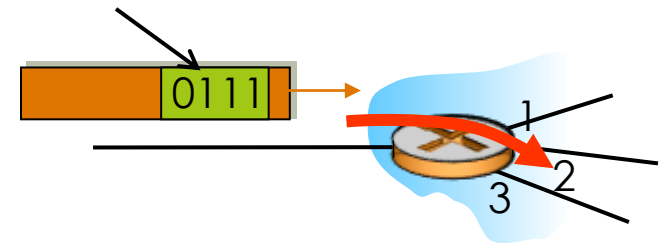
# Network Planes

- Data Plane
  - responsible for forwarding packets
- Control Plane
  - responsible for defining the behavior of the network
- Management Plane
  - configures routing protocol, monitors network devices and handle anomalies

# Data plane

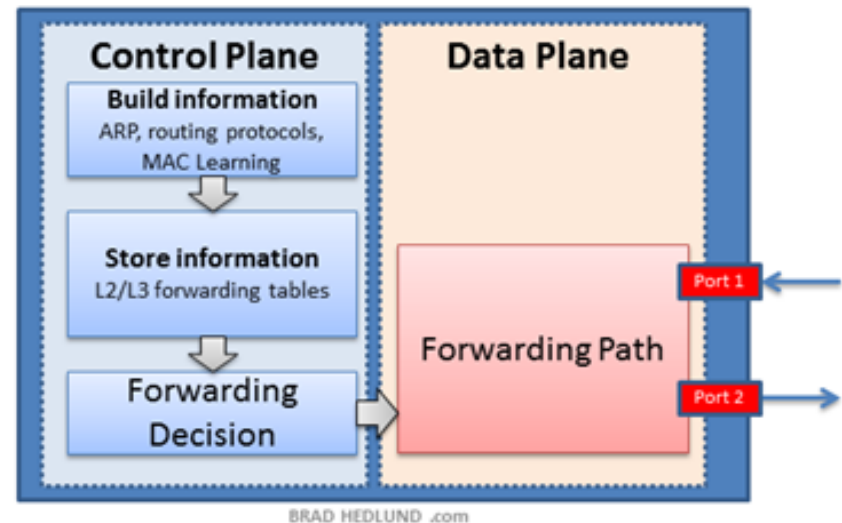
- Responsible for moving packets
  - local, per-router function
  - determines how packet arriving on router input port is forwarded to router output port
  - forwarding function
- Other data plane functions:
  - Packet scheduling and buffering
  - Packet fragmentation
  - Bandwidth metering
  - Access control
  - Packet cloning for multicasting
  - Traffic monitoring

values in arriving packet header



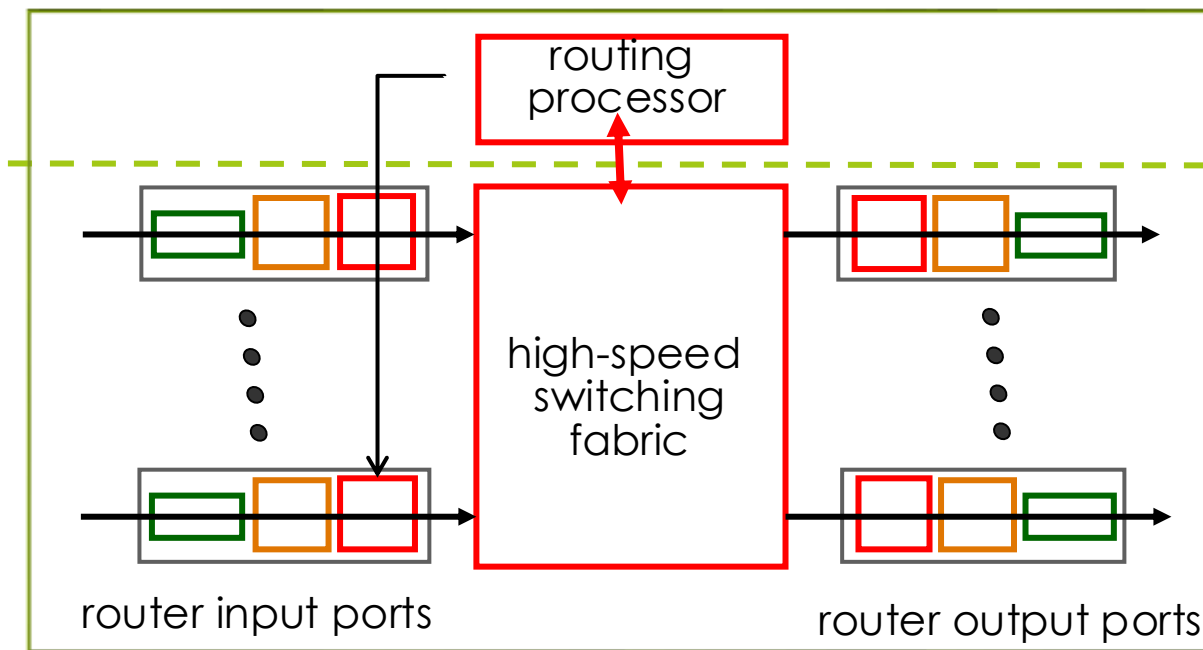
# Control plane

- ▣ Defines the network behavior
  - ▣ network-wide logic
  - ▣ determines how packet is routed along end-end path from source host to destination host
  - ▣ Route computation function
  
- ▣ Other control plane functions:
  - ▣ FIB installation in routers
  - ▣ Topology discovery
  - ▣ Access control list generation
  - ▣ QoS enforcement



# Traditional Networking

- Control plane is vertically integrated with data plane
  - Each router runs its own control plane
- Generic router architecture:

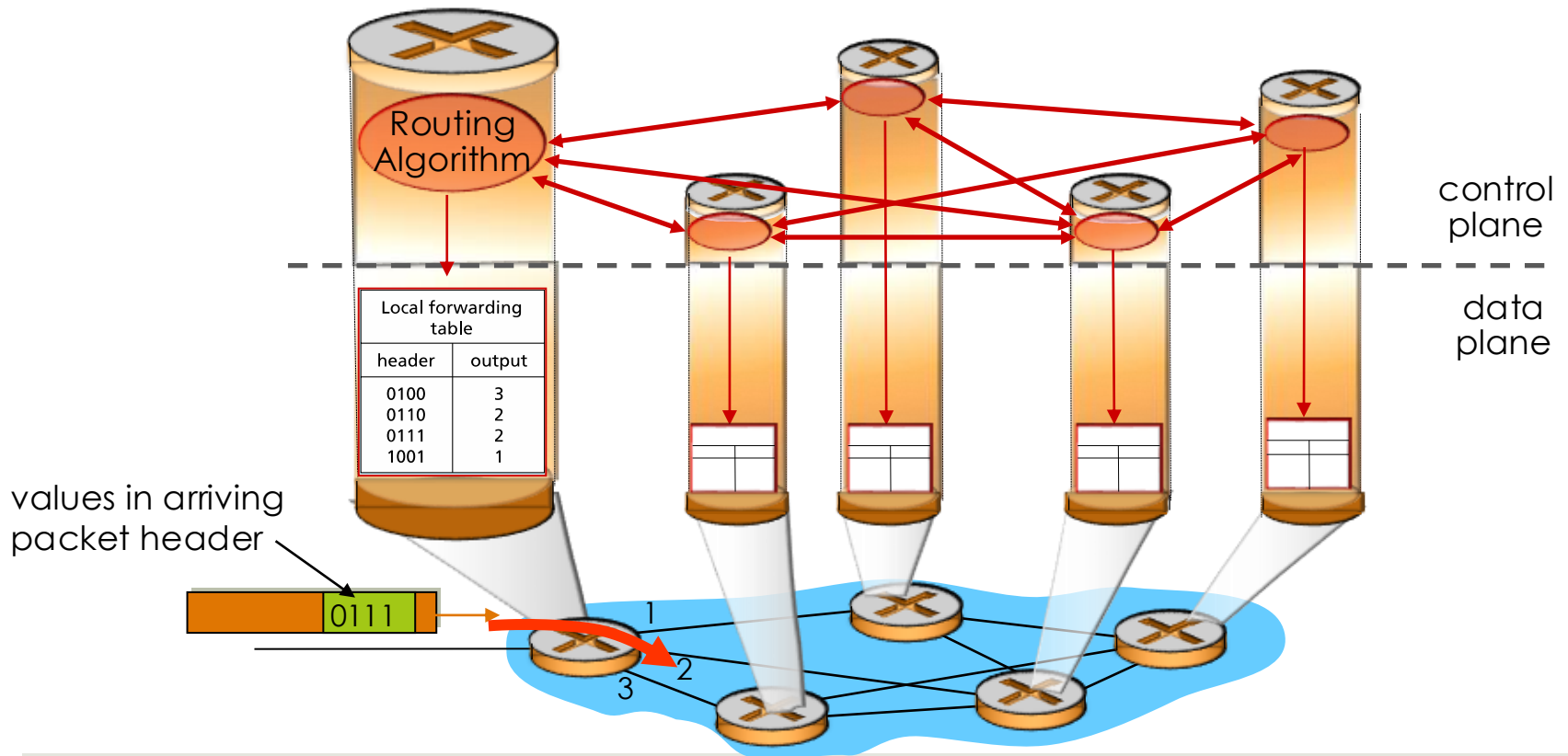


*routing control plane*  
(software) operates in  
millisecond time frame

*forwarding data plane*  
(hardware) operates in  
nanosecond timeframe

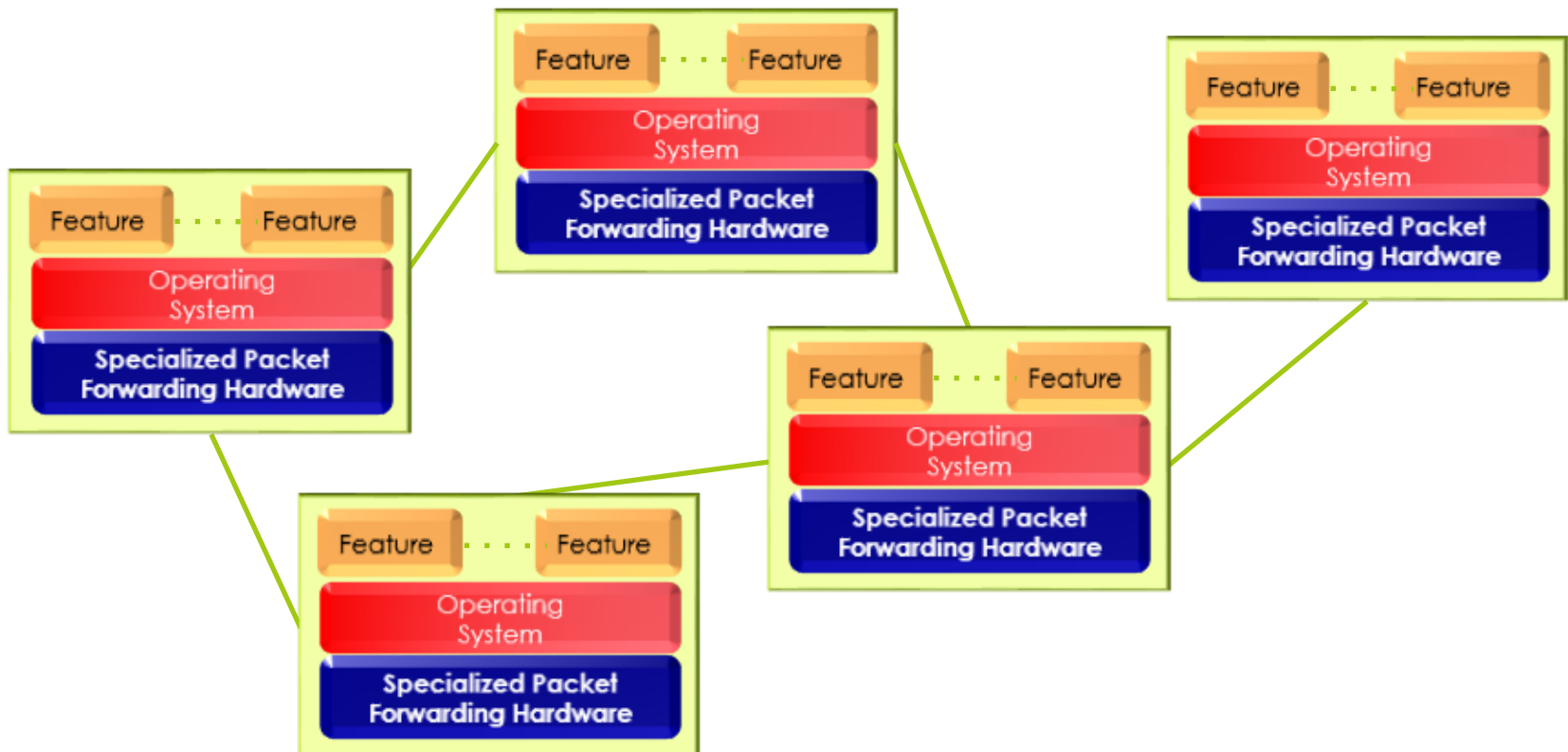
# Traditional Routing

- Traditional routing protocols are distributed
  - Routers exchange connectivity info. with each other
  - Each router builds its own routing table
  - Forwarding tables (FIB) are populated based on information available in the routing table

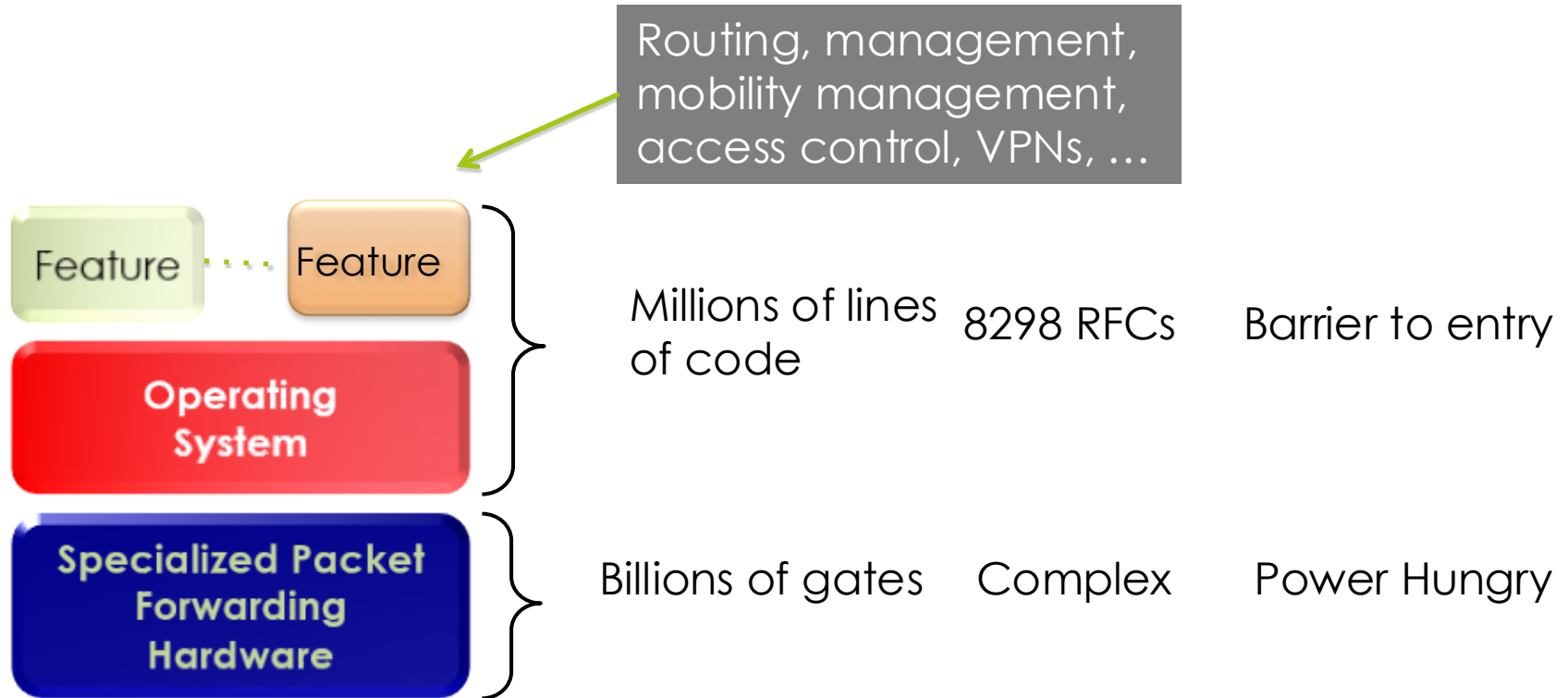


# Distributed Control Plane

- Distributed per-router control plane
  - Vertically integrated router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)



# Traditional Network Element



# Shortcomings of traditional networking

- ❑ Lack of programmability
  - ❑ Changes in the network requires manual reconfiguration
  - ❑ Error-prone and time consuming
  - ❑ Network becomes hard to scale
- ❑ Vendor dependency
  - ❑ Handful of vendors have monopoly in the market
  - ❑ Lack of open interface
  - ❑ Software vertically integrated with hardware
- ❑ Over specified
  - ❑ Slow protocol standardization

# Shortcomings of traditional networking

- ❑ Less opportunities for innovation
  - ❑ Only vendors can write the code
  - ❑ Slow time-to-market for new services/features
- ❑ High operational cost
  - ❑ More than 50%
  - ❑ Human error still causes most outages
- ❑ Buggy software
  - ❑ Hardware with 20+ million lines of code
  - ❑ Cascading failures, vulnerabilities, *etc.*

# The advent of SDN

- ~2005: renewed interest in rethinking network control plane
- Software Defined Networking (SDN) emerged as a new networking architecture that
  - Separates control plane from data plane
  - Software running on a *logically centralized* server remotely controls network hardware
  - Open standards (vendor neutral)
  - Directly programmable
  - Facilitates automation & programmability

# SDN architecture

Consistent, up-to-date global network view

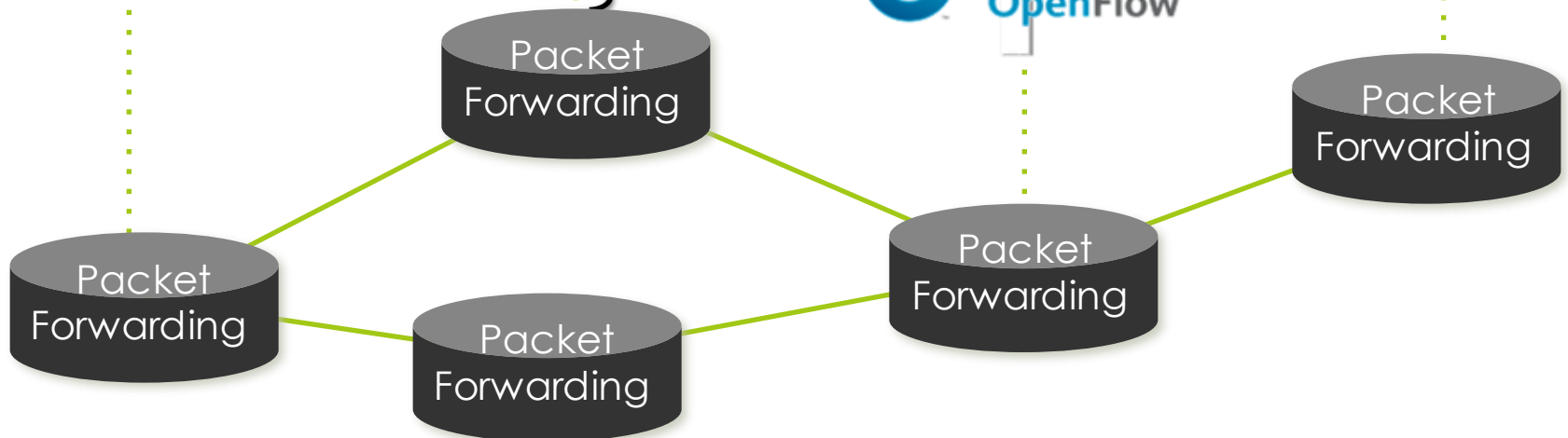
Control App 1

Control App 2

At least one Network OS probably many.  
Open- and closed-source

Network OS

Open interface for network device control



# Why SDN for future networks?

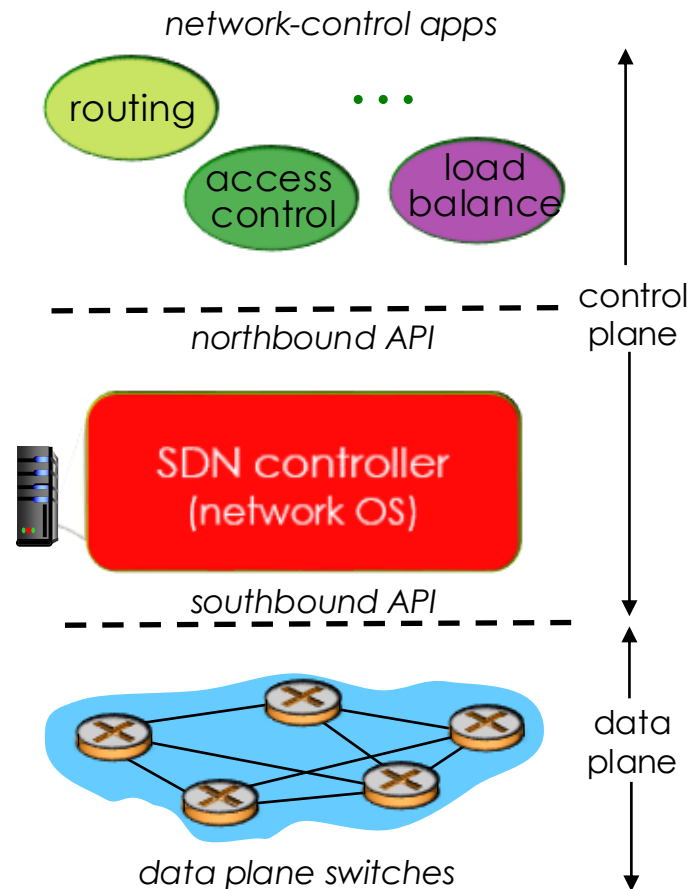
- Separation of control and data planes
  - Enables independent development of new features without hardware modification
- Global network view at control plane
  - Network-wide view allows easier application of policies
- Management Automation
  - Minimize human intervention
  - Lower operational cost
- Monitoring
  - On-demand monitoring of resources
  - Dynamically change what to monitor
  - Application performance can be improved by advanced monitoring features

# Why SDN for future networks?

- Elasticity
  - Resources can be scaled up or down on-demand
- More Flexible/Easier Performance Management
  - Traffic engineering
  - Capacity optimization
  - Load balancing
  - Faster failure recovery
- Network Function Integration
  - On-demand provisioning of network functions
    - Firewalls, IDS, IPS, etc.

# Components of SDN

- Data plane switches
  - SDN controlled switches
- South Bound API
  - Interface between control and data plane (e.g. OpenFlow)
- SDN Controller
- North Bound API
  - Interface between controller and control apps



# From Theory to Practice

- Traditional networks: control and data plane vertically integrated
- SDN separates them: control plane becomes software
  - Network behaviour defined programmatically
- Lab 1: SDN Data plane programmability
  - Mininet: network emulator, full topology on one machine
  - OVS: software switch, programmatically controlled
  - Flow rules: tool for defining network behaviour
  - No controller yet: you are the control plane