KNUST **RANKS NO.1 GLOBALLY** FOR THE PROVISION OF QUALITY EDUCATION **(SDG 4)** By: THE WORLD UNIVERSITY RANKINGS

SDG 4

QUALITY EDUCATION

TE 156
Introduction to Communication Networks

Programmable Networks

Outline

- Philosophy of Traditional Computer Networking
- Limitations of Traditional Computer Networking

Traditional Computer Networking

- Computer Network Design Rule and Philosophy
  - Networks and networking devices are designed to overcome RARE but severe challenges
    - The philosophy is thus SURVIVABILITY

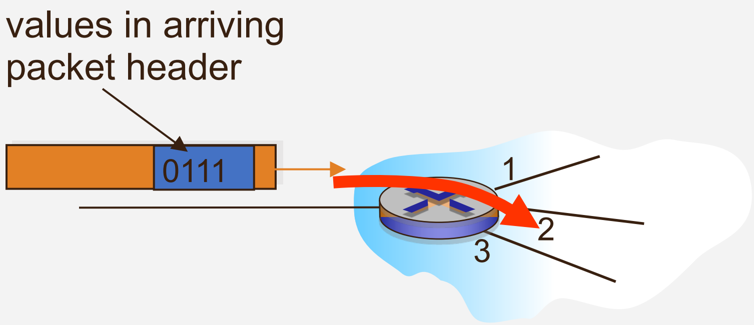
Current State in Networking

- Computer networks can be divided in three planes of functionality: the data, control and management planes
  - ***The data plane*** corresponds to the networking devices, which are responsible for (efficiently) forwarding data.
  - ***The control plane*** represents the protocols used to populate the forwarding tables of the data plane elements.
  - ***The management plane*** includes the software services, such as SNMP-based tools used to remotely monitor and configure the control functionality

Network layer: data plane, control plane

Data plane

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port
- forwarding function



Control plane

- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- Current control-plane approach:
  - *traditional routing algorithms:* implemented in routers

Current State in Networking

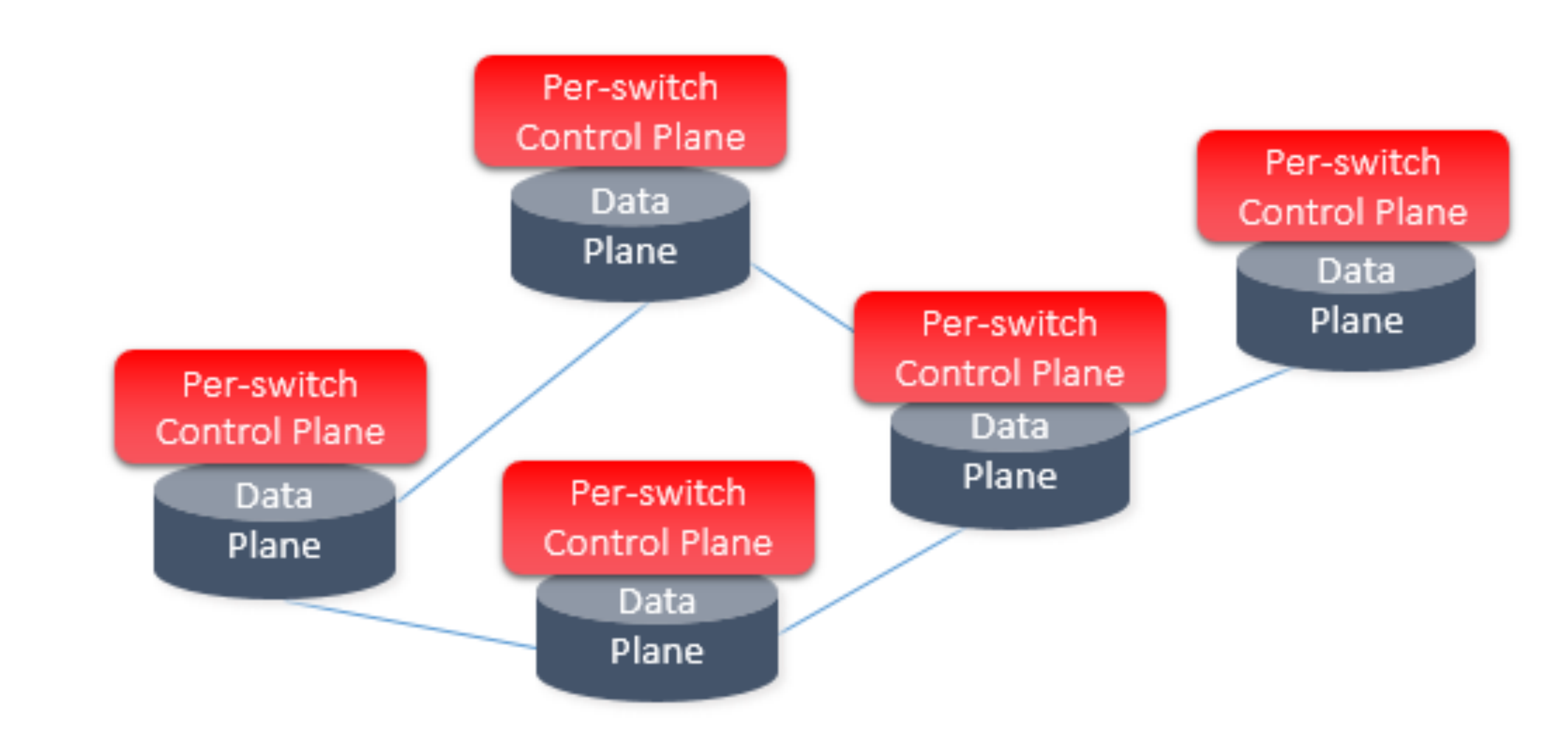
- Network policy is
  - defined in the management plane
    - the control plane enforces the policy
      - and the data plane executes it by forwarding data accordingly

Current State in Networking

- In traditional IP networks
  - the control and data planes are tightly coupled, embedded in the same networking devices, and the whole structure is highly decentralized.
- The outcome is a
  - very complex and relatively static architecture
- It is also the fundamental reason why traditional networks are
  - rigid, and complex to manage and control.
- innovation is difficult.

Traditional Networking

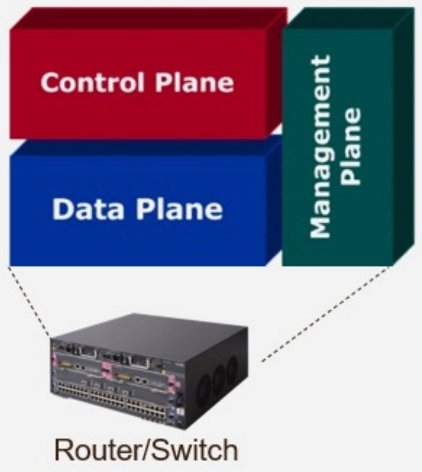
- The data and control planes baked into one box



Networking Planes

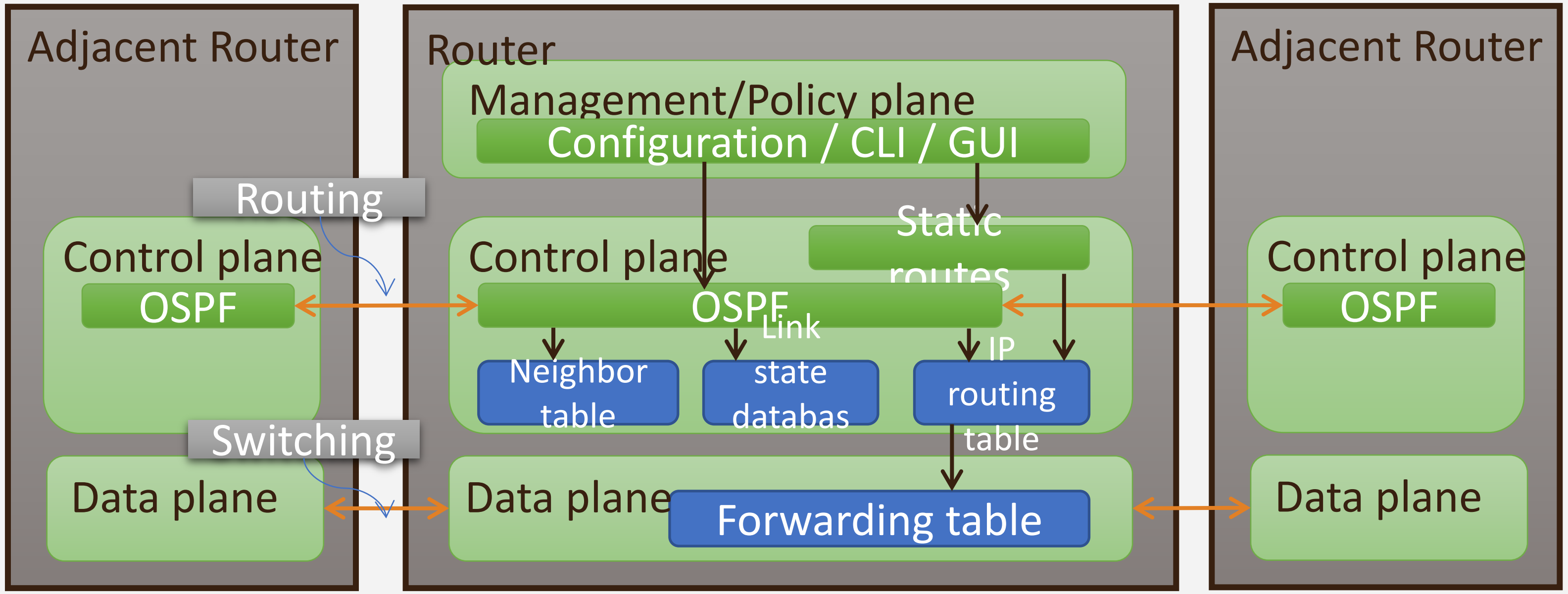
Typical Networking Software

- Management plane
- Control Plane – The brain/decision maker
- Data Plane – Packet forwarder



Traditional Network Router

- Router can be partitioned into control and data plane
  - Management plane/ configuration
  - Control plane / Decision:
  - Data plane / Forwarding



Network-layer functions

Recall: two network-layer functions:

- *forwarding:* move packets from router's input to appropriate router output — *data plane*
- *routing:* determine route taken by packets from source to destination — *control plane*

Traditional approach to structuring network control plane:

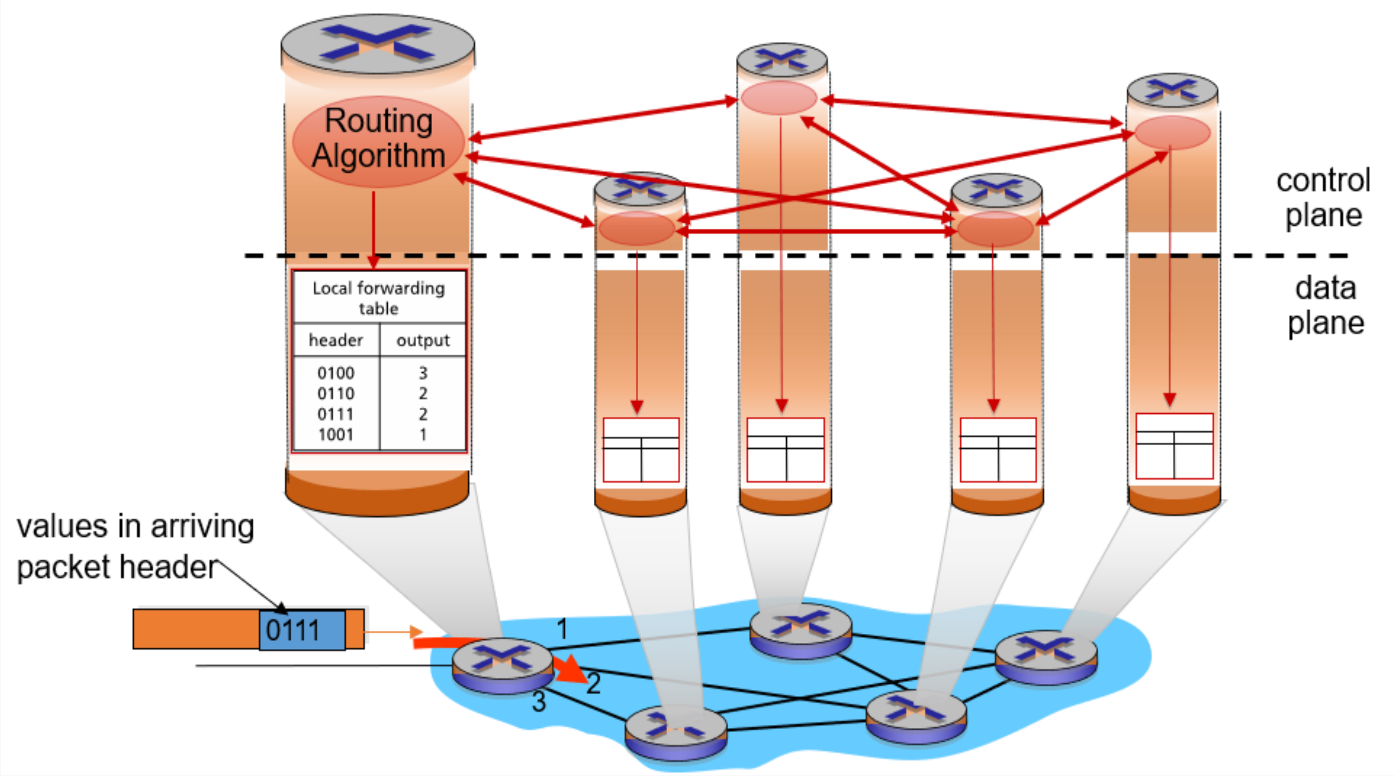
- per-router control

The Networking "Planes"

- **Data plane**: processing and delivery of packets with local forwarding state
  - Forwarding state + packet header → forwarding decision
  - Filtering, buffering, scheduling
- **Control plane**: computing the forwarding state in routers
  - Determines how and where packets are forwarded
  - Routing, traffic engineering, failure detection/recovery, …
- **Management plane**: configuring and tuning the network
  - Traffic engineering, ACL config, device provisioning, …

Per-router control plane

- Individual routing algorithm components *in each and every router* interact in the control plane



Traditional networking

- Internet network layer: historically has been implemented via distributed, per-router approach
  - *monolithic* 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)
  - different “middleboxes” for different network layer functions: firewalls, load balancers, NAT boxes, ..

Two Key Definitions

- **Data Plane**: processing and delivery of packets
  - Based on state in routers and endpoints
  - E.g., IP, TCP, Ethernet, etc.
  - Fast timescales (per-packet)
- **Control Plane**: establishing the state in routers
  - Determines how and where packets are forwarded
  - Routing, traffic engineering, firewall state, …
  - Slow time-scales (per control event)

Limitations

- Networks are hard to manage
- Networks are hard to evolve
  - Networks are stuck in the past
    - Routing algorithms change very slowly
    - Network management extremely primitive
- Networks design not based on formal principles
  - Networking courses teach a big bag of protocols
    - No formal principles, just general design guidelines

Challenges

(Too) many task-specific control mechanisms

- No modularity, limited functionality

Indirect control

- Must invert protocol behavior, “coax” it to do what you want
- Ex. Changing weights instead of paths for TE

Uncoordinated control

- Cannot control which router updates first

Interacting protocols and mechanisms

- Routing, addressing, access control, QoS

Network Control/Management Plane

A very different story!

- There are *no general principles or abstractions* guiding the design of network *control/management* plane
  - in contrast, e.g., to distributed systems or database systems
- Control plane is basically composed of various distributed “control” protocols
  - For each new feature/functionality needed, a new protocol is designed → new software run by “flimsy” CPU in routers
- Worse, network boxes are “closed” equipment
  - (mostly proprietary) software bundled with hardware
  - vendor-specific interfaces;
  - slow protocol standardization – often years!

Managing Networks …

- Networks used to be simple: Ethernet, IP, TCP….
- New **control** requirements led to great complexity
  - Isolation → VLANs, ACLs
  - Traffic engineering → MPLS, ECMP, Weights
  - Packet processing → Firewalls, NATs, middleboxes
  - Payload analysis → Deep packet inspection (DPI)
  - …..
- Mechanisms designed and deployed independently
  - Complicated “control plane” design, primitive functionality
  - Stark contrast to the elegantly modular “data plane”

Limitations of current Networks

- Cannot dynamically change according to network conditions

Infrastructure Still Works!

- **Only** because of “our” ability to master complexity

What is the Point?

- The ability to ***master complexity*** is not the same as the ability to ***extract simplicity***

- When first getting systems to work….
  - Focus on mastering complexity

- When making system easy to use and understand
  - Focus on extracting simplicity

- **You will never succeed in extracting simplicity**
  - If don't recognize it is different from mastering complexity

What Is *My* Point?

- Networking still focused on mastering complexity
  - Little emphasis on extracting simplicity from control plane
  - No recognition that there's a difference….
- Extracting simplicity builds intellectual foundations
  - **Necessary for creating a discipline….**
  - **That's why networking lags behind**
- **Abstractions key to extracting simplicity**
- *What abstractions do we have in networking?*

- Abstraction
  - Is the act of representing essential features without including the background details or explanations

Data Plane Abstractions: Layers

Applications

...built on...

Reliable (or unreliable) transport

...built on...

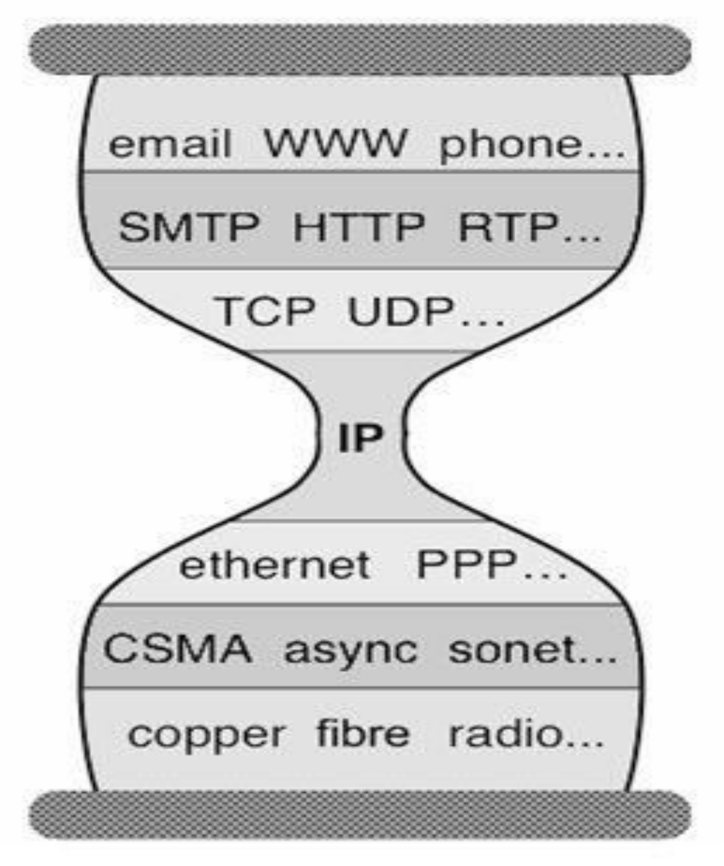
Best-effort global packet delivery

...built on...

Best-effort local packet delivery

...built on...

Local physical transfer of bits



Control Plane Abstractions

?

Control Plane Abstractions

- How do we find these abstractions?
- Define our problem, and then decompose it

The Control Plane Problem

- What is the control plane problem?

The Control Plane Problem

- Control plane must compute forwarding state. To accomplish its task, the control plane must:
  1. Figure out what the network looks like (topology)
  2. Figure out how to accomplish goal on given topology
  3. Tell the swtiches what to do (configure forwarding state)
- What components do we want to reuse?
  1. Determining the topology information
  3. Configuring forwarding state on routers/switches

Two Control Plane Abstractions

- Abstraction: **global network view**
  - Provides information about current network

- Abstraction: **forwarding model**
  - Provides standard way of defining forwarding state

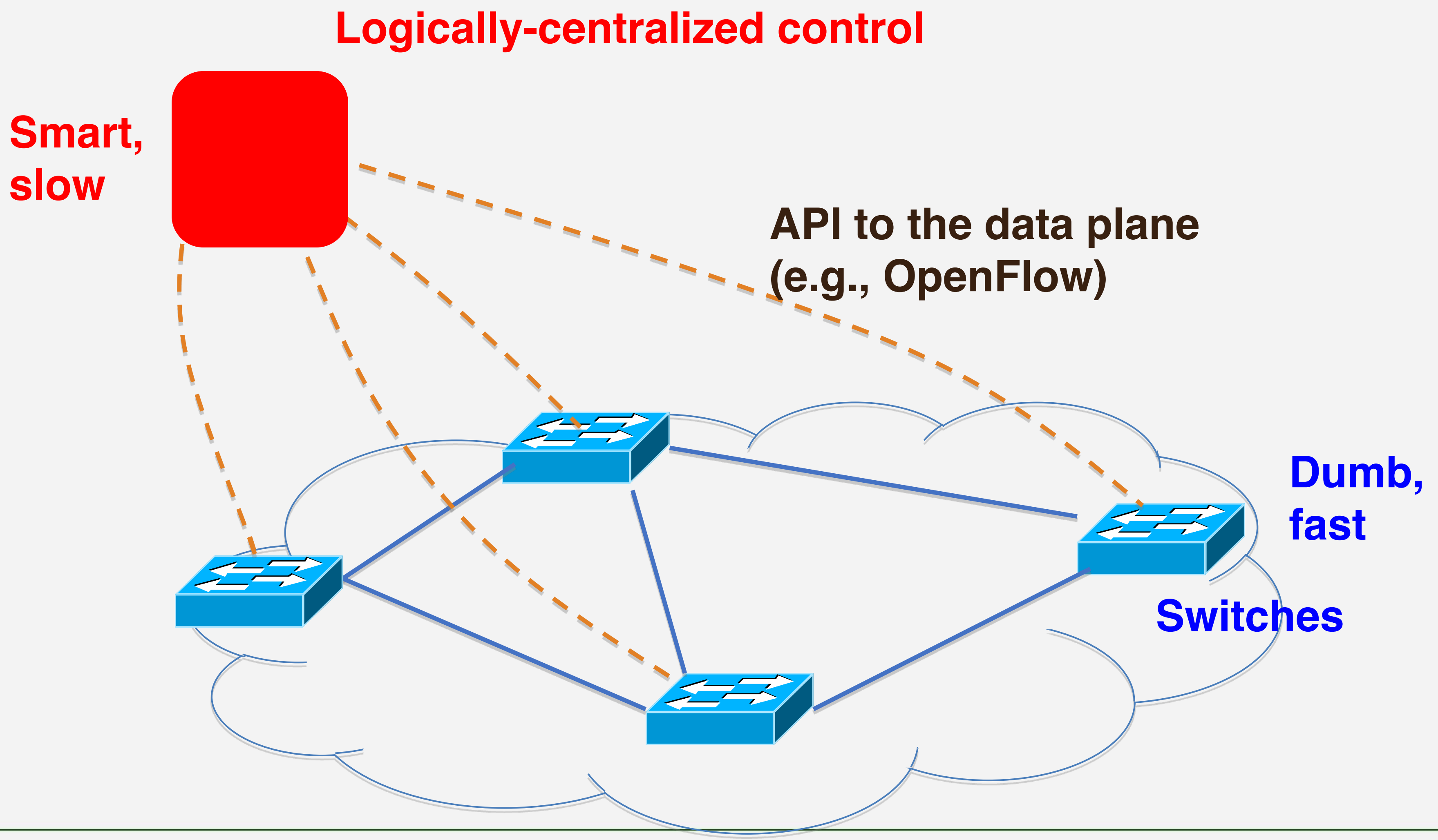
SDN: Two Control Plane Abstractions

- Abstraction: **global network view**
  - Provides information about current network
  - **Implementation**: “Network Operating System”
    - Runs on servers in network (replicated for reliability)
- Abstraction: **forwarding model**
  - Provides standard way of defining forwarding state
  - This is OpenFlow
    - Specification of <match,action> flow entries

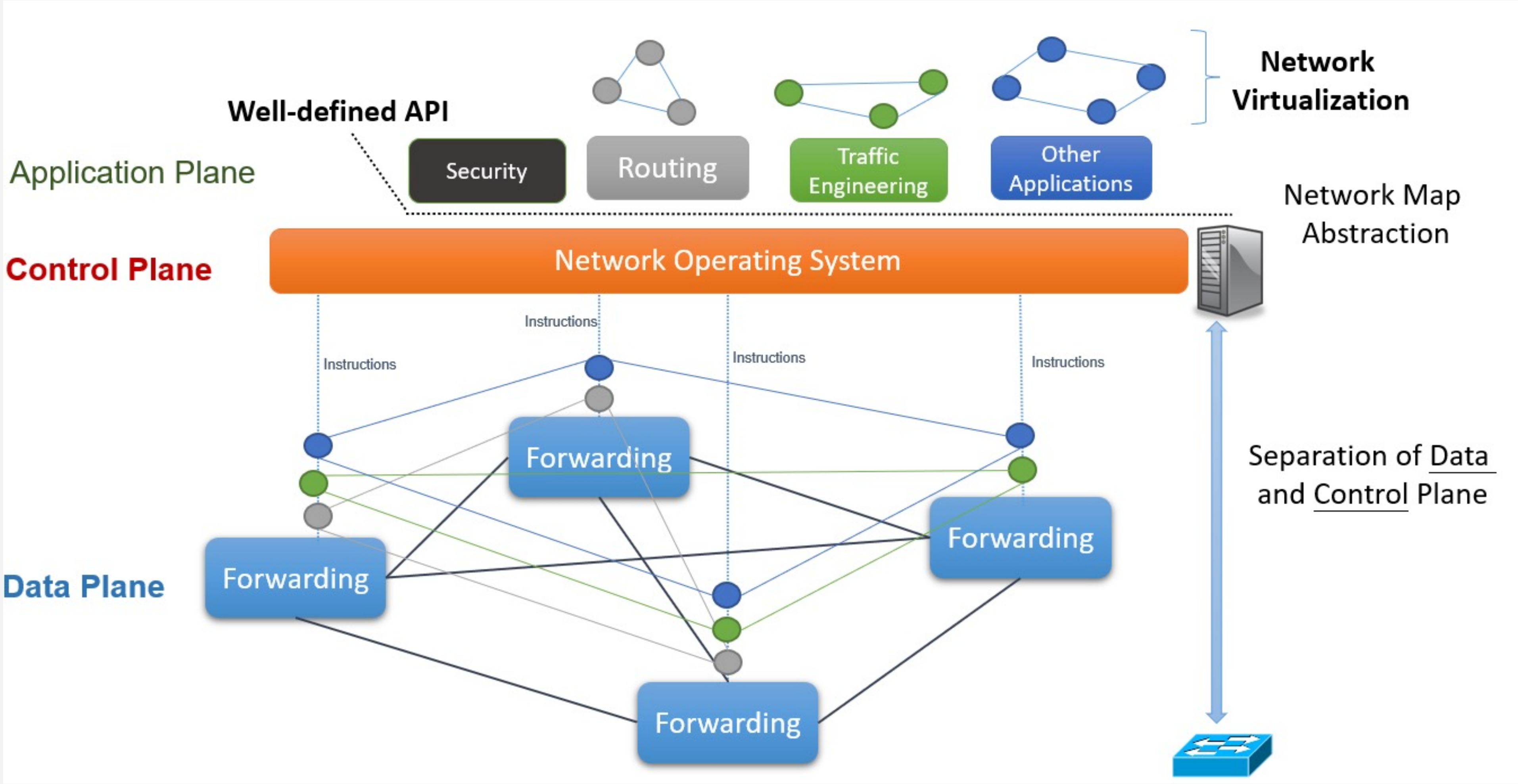
SDN Basic Concept

- Separate Control plane and Data plane entities
  - Network intelligence and state are logically centralized.
  - The underlying network infrastructure is abstracted from the applications.
- Execute or run Control plane software on general purpose hardware
  - Decouple from specific networking hardware
  - Use commodity servers and switches
- Have programmable data planes
  - Maintain, control and program data plane state from a central entity
- An architecture to
  - control not just a networking device but an entire network

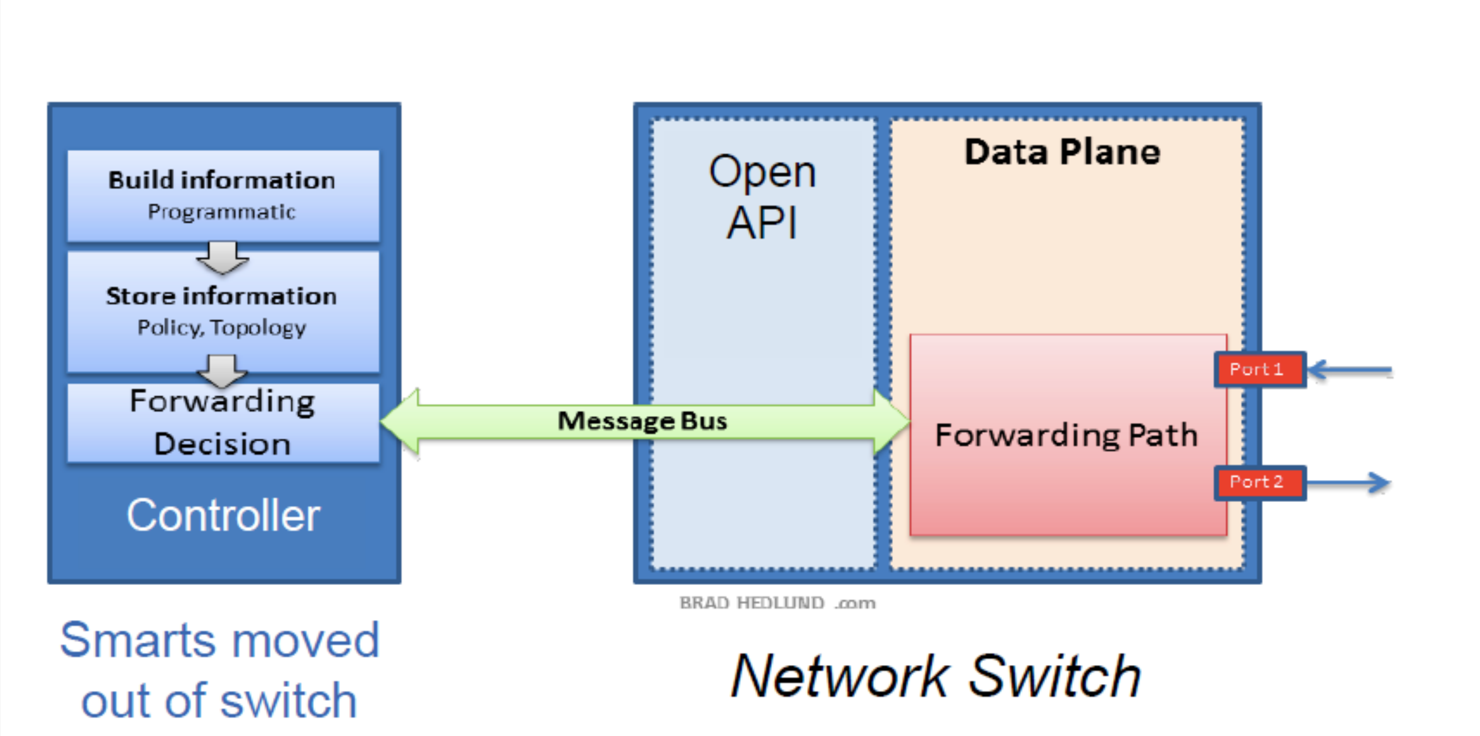
Software Defined Networking (SDN)



Software-Defined Network with key Abstractions



SDN Basics


Build information
Programmatic
Store information
Policy, Topology
Forwarding Decision
Controller
Message Bus
Open API
Data Plane
Forwarding Path
Port 1
Port 2
BRAD HEDLUND .com
Smarts moved out of switch
Network Switch

Specification Abstraction

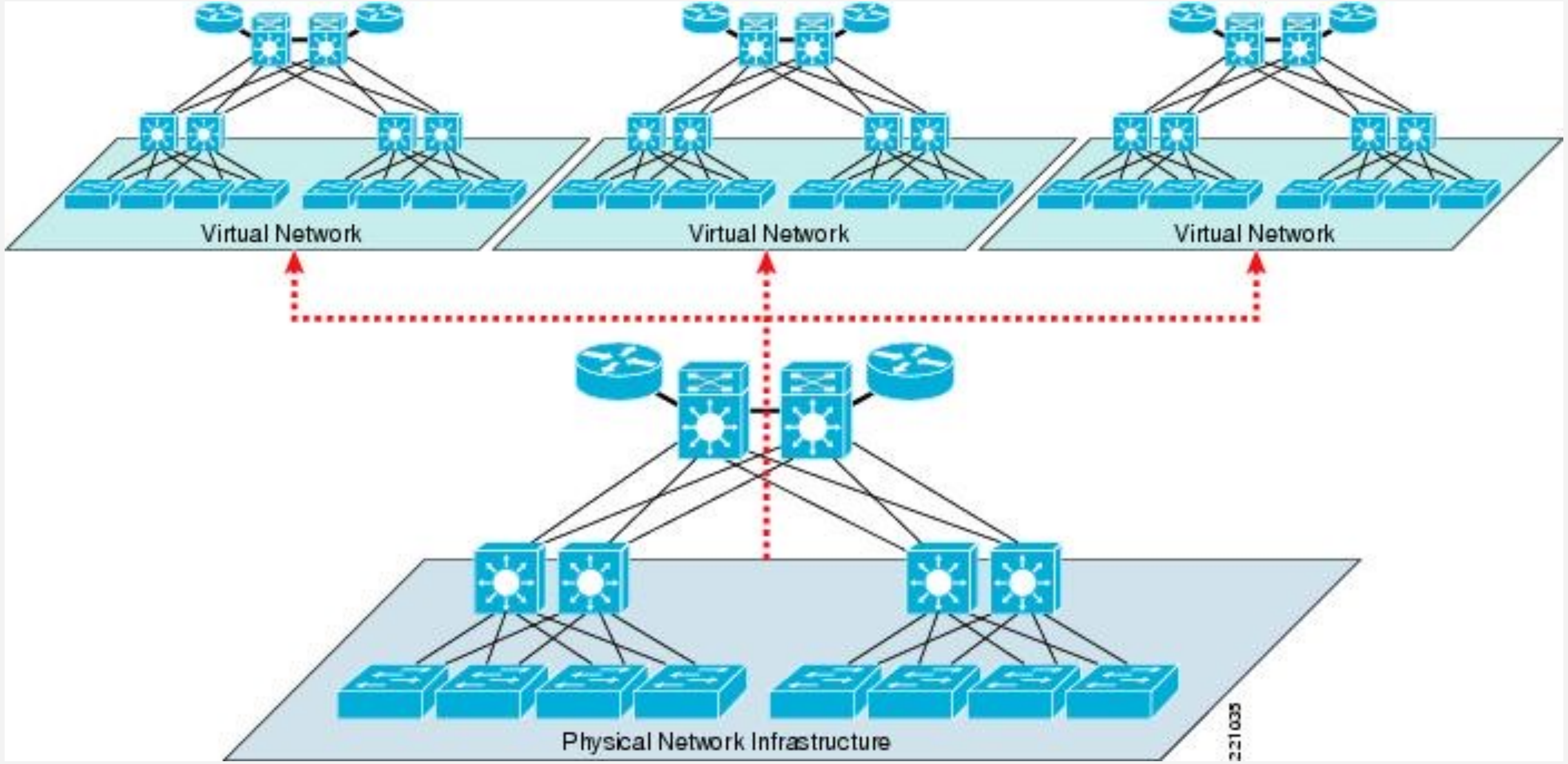
- Control program must ***express*** desired behavior
  - Whether it be isolation, access control, or QoS

- It should not be responsible for ***implementing*** that behavior on physical network infrastructure
  - Requires configuring the forwarding tables in each switch

- Proposed abstraction: **Virtual Topology** of network
  - Virtual Topology models only enough detail to <u>*specify goals*</u>

Network Virtualization

- What is network virtualization ?

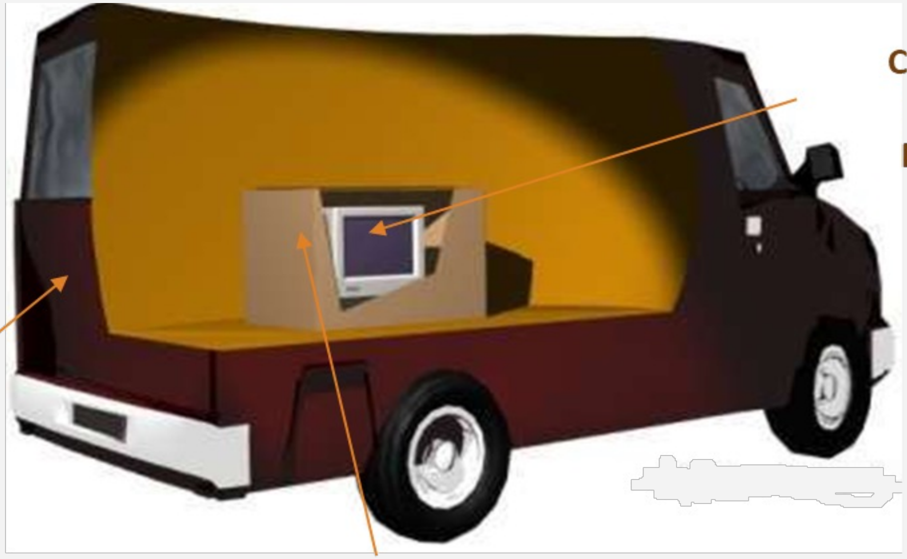


Network Virtualization

- Introduce new abstraction and new SDN layer
- Abstraction: **Virtual Topology**
  - Allows operator to express requirements and policies
  - Via a set of logical switches and their configurations
- Layer: **Network Hypervisor**
  - Translates those requirements into switch configurations
  - "Compiler" for virtual topologies

Network Design Not Based On Formal Principles

- Network courses teach
  - a big bag of PROTOCOLS



Networks are Hard to Manage

- Networks are still notoriously difficult to manage

Networks are Hard to Evolve

- Networks are stuck in the past
  - Routing algorithms change very slowly
  - Network management extremely primitive

THANK YOU

Kwame Nkrumah University of Science and Technology, Kumasi | Leaders In Change

Visit us at www.knust.edu.gh

uro@knust.edu.gh | Follow KNUST on: