CS 352
Software-Defined Networking: Intro

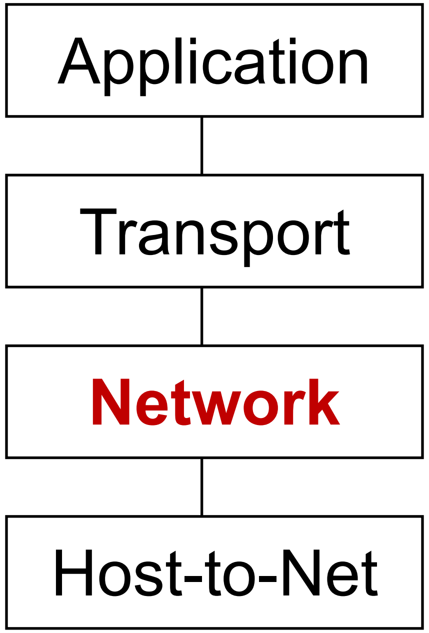
CS 352, Lecture 21.1

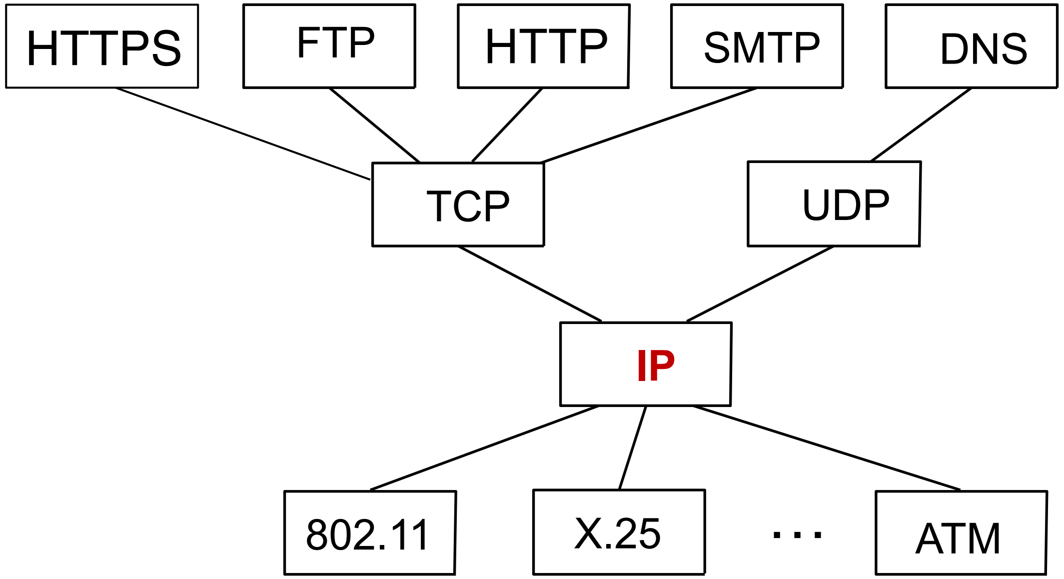
http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Network





The main function of the network layer is to
move packets from one endpoint to another.

Review: Network layer functions

- Forwarding: move packets from router's input to appropriate router output

- Routing: determine route taken by packets from source to destination

- Data Plane

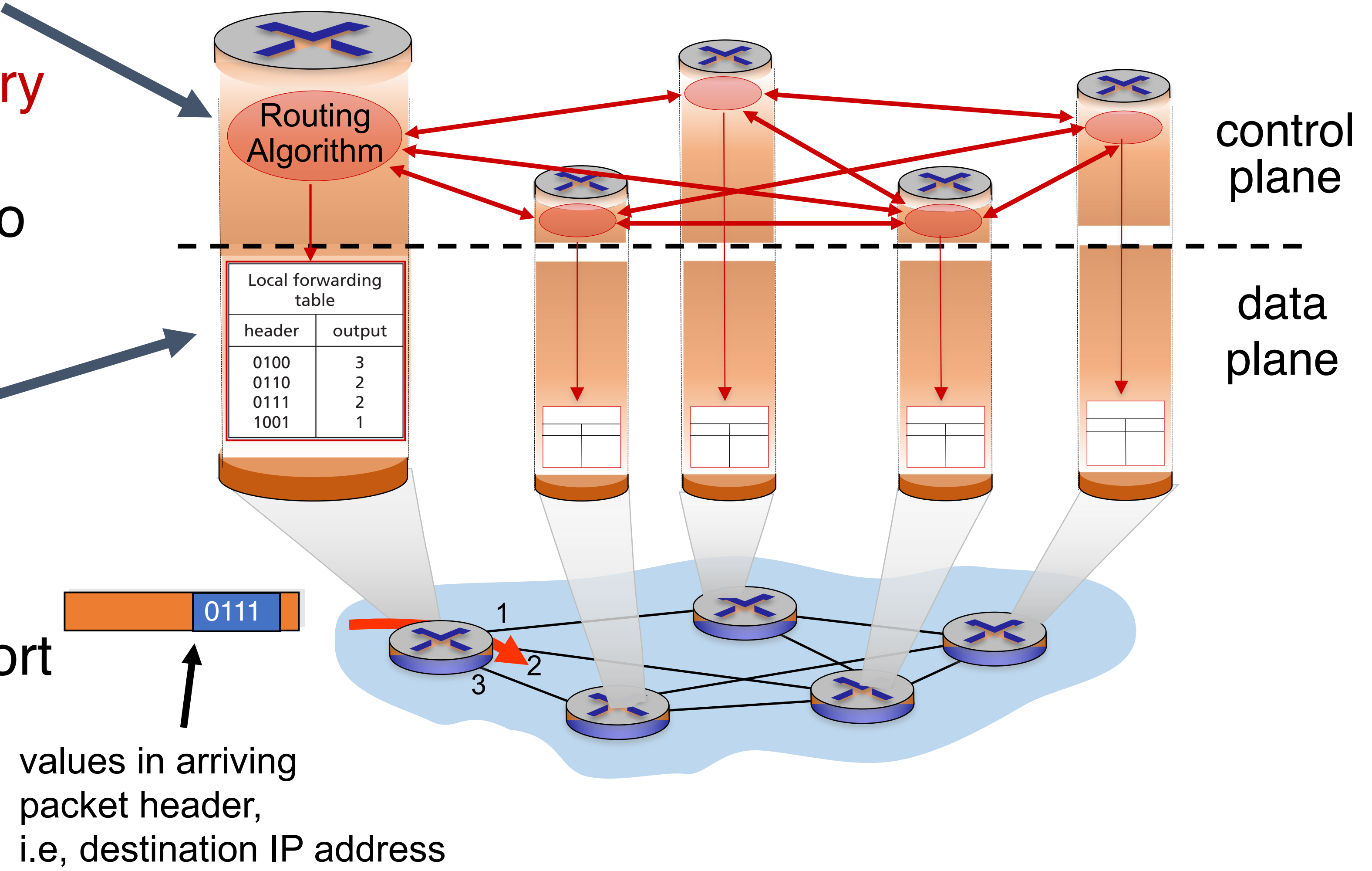
- Control Plane

- Two kinds of control planes:
  - Distributed per-router control
  - Logically centralized ← This lecture

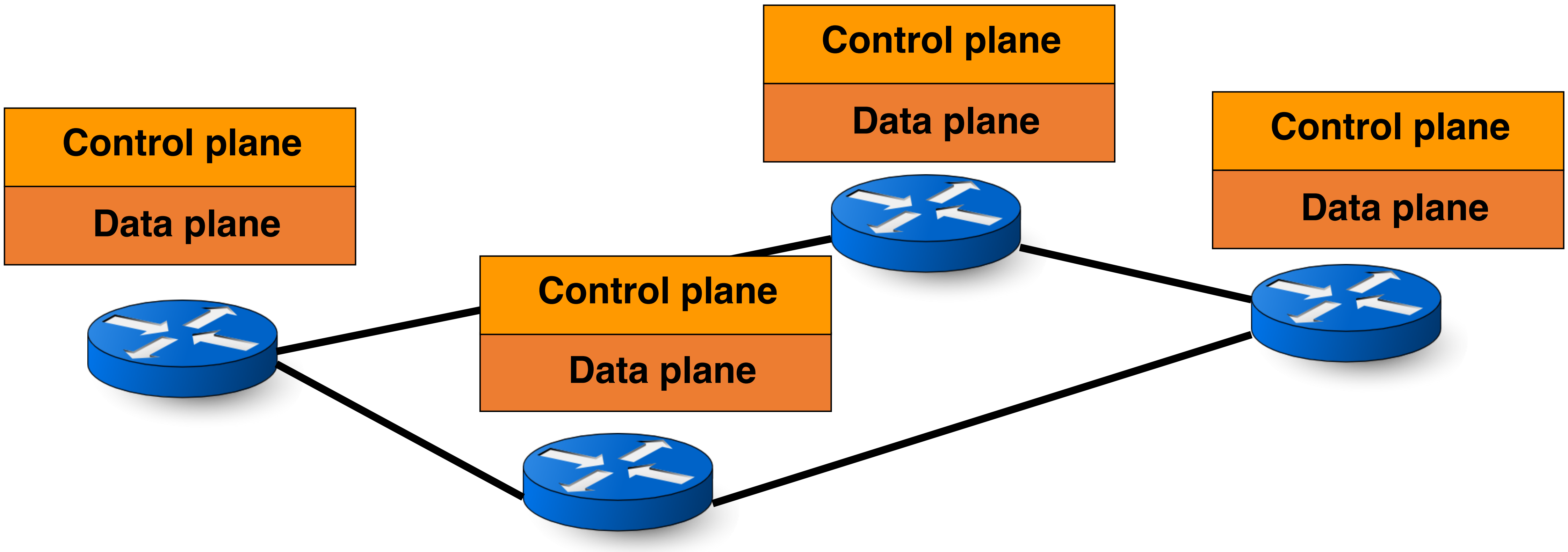
Review: Traditional distributed ctrl planes

Distributed control plane: Components in every router interact with other components to produce a routing outcome.

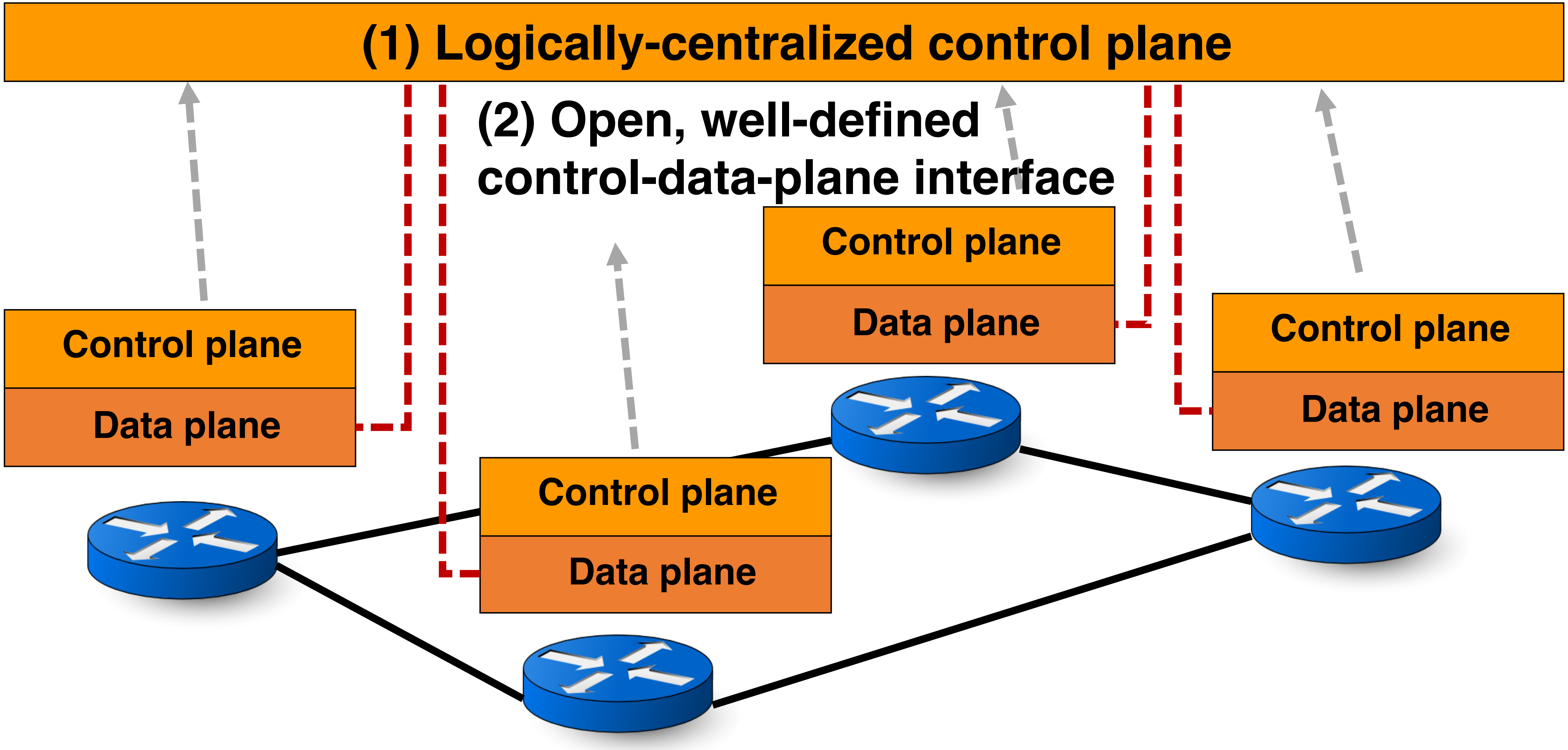
Data plane per-packet processing, moving packet from input port to output port



Traditional distributed control plane



Software-defined network (SDN)



Two components of an SDN

- Logically centralized control plane: the SDN controller
  - Function implemented in software: software-defined
  - General-purpose software to compute forwarding paths and policies
  - Not subject to restrictions of distributed protocols (more soon)
- Open control-data interface
  - Routers expose a general instruction set (next module)
  - Statistics, events (link changes, interface changes), control messages, etc.

Why SDN?

A reaction to two trends

Routers hard to change (~2005)

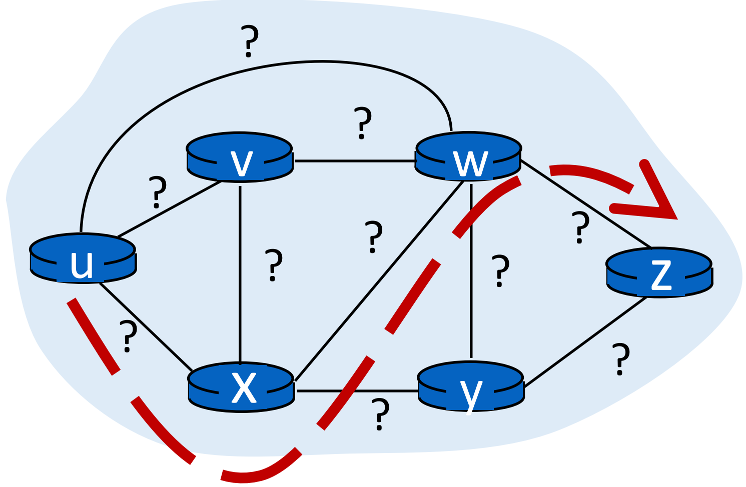
- Router vendors packaged routers with hardware for forwarding and software for the control plane processor
  - Proprietary software and interfaces to hardware
  - ISP operators relegated to a rigid, high-level command line interface
- The behavior of the routers (owned by an ISP) couldn’t be changed by the ISP’s network operators
  - Require vendor support
- Vendor software bloated and buggy
  - Including bugs from features that were never used by the ISP

Operators need better abstractions

- Larger networks come with harder problems
  - more devices to manage
  - higher data volumes
  - more failures

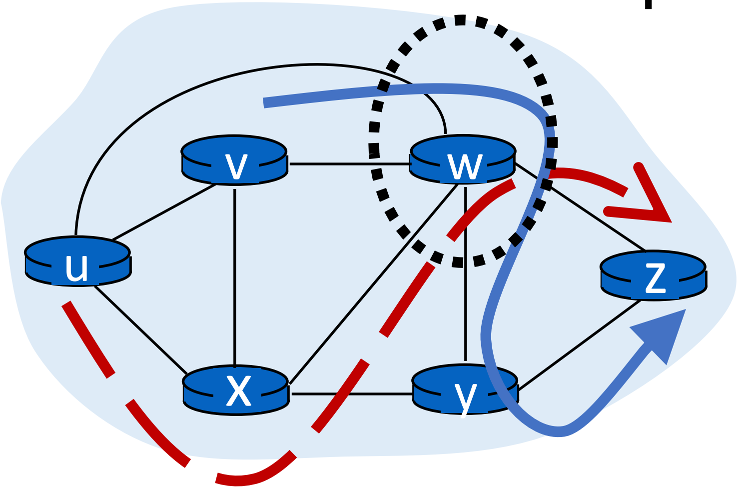
Operators need better abstractions

- Example 1: Choosing network paths freely
- Distributed intra-domain routing protocols compute least cost paths from link metrics
- Q: How to forward u → z data traffic along the path shown?
- Link weight inference problem: set link weights such that least cost paths are those that are desired



Operators need better abstractions

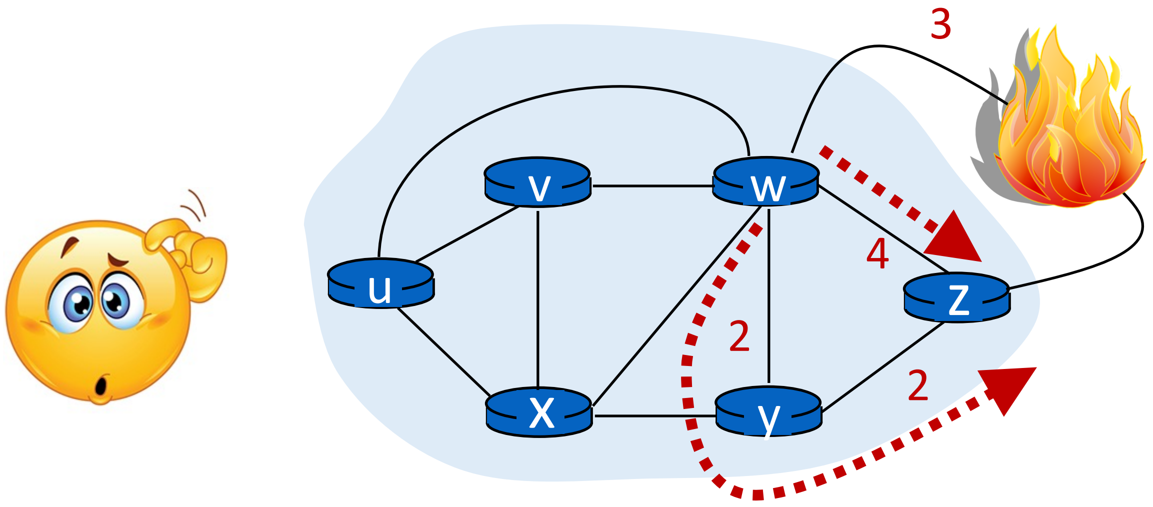
- Example 2: More flexible forwarding decisions
- Distributed routing protocols and data planes forward packets using destination address
- But might want to forward based on other criteria
  - Example: by both source and destination, to balance link usage
- Traditional routing: can't use different paths for traffic to the same destination



Requirement at node w:
Originate from u, dst z:
w → z
Originate from v, dst z:
w → y → z

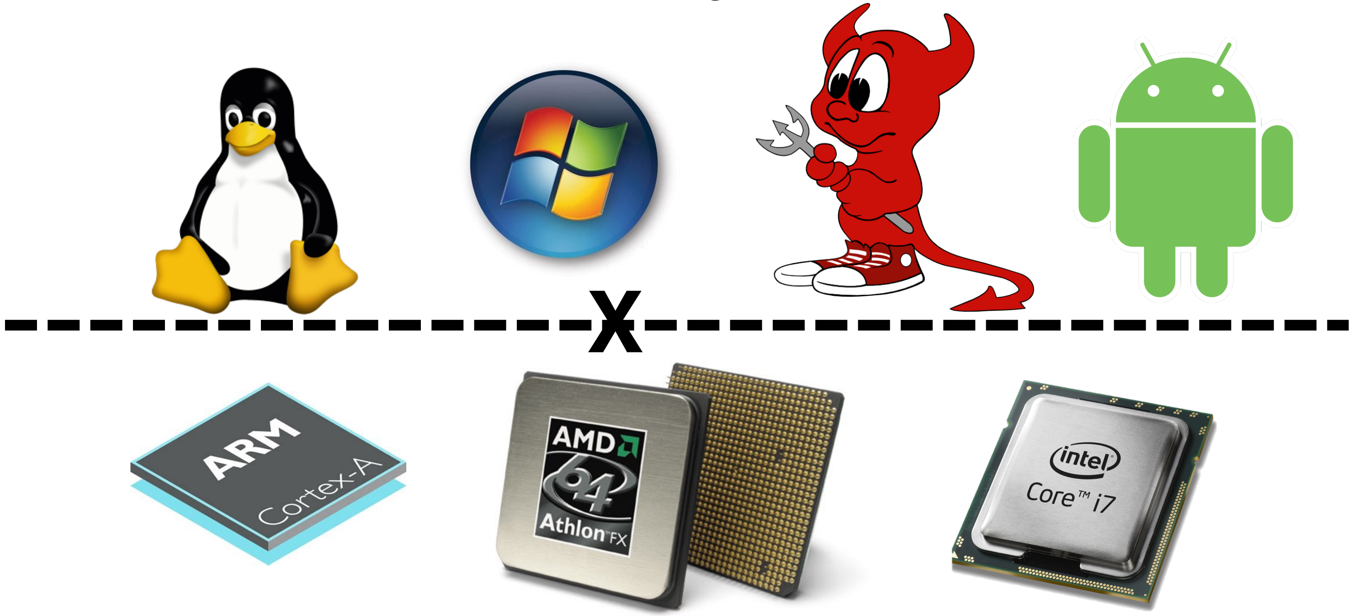
Operators need better abstractions

- Example 3: Reaction to network failures
- When a link fails, traditional routing protocols re-converge to an outcome based on link metrics on new network
- Two problems with this approach:
  - Slow convergence
  - Non-deterministic outcomes

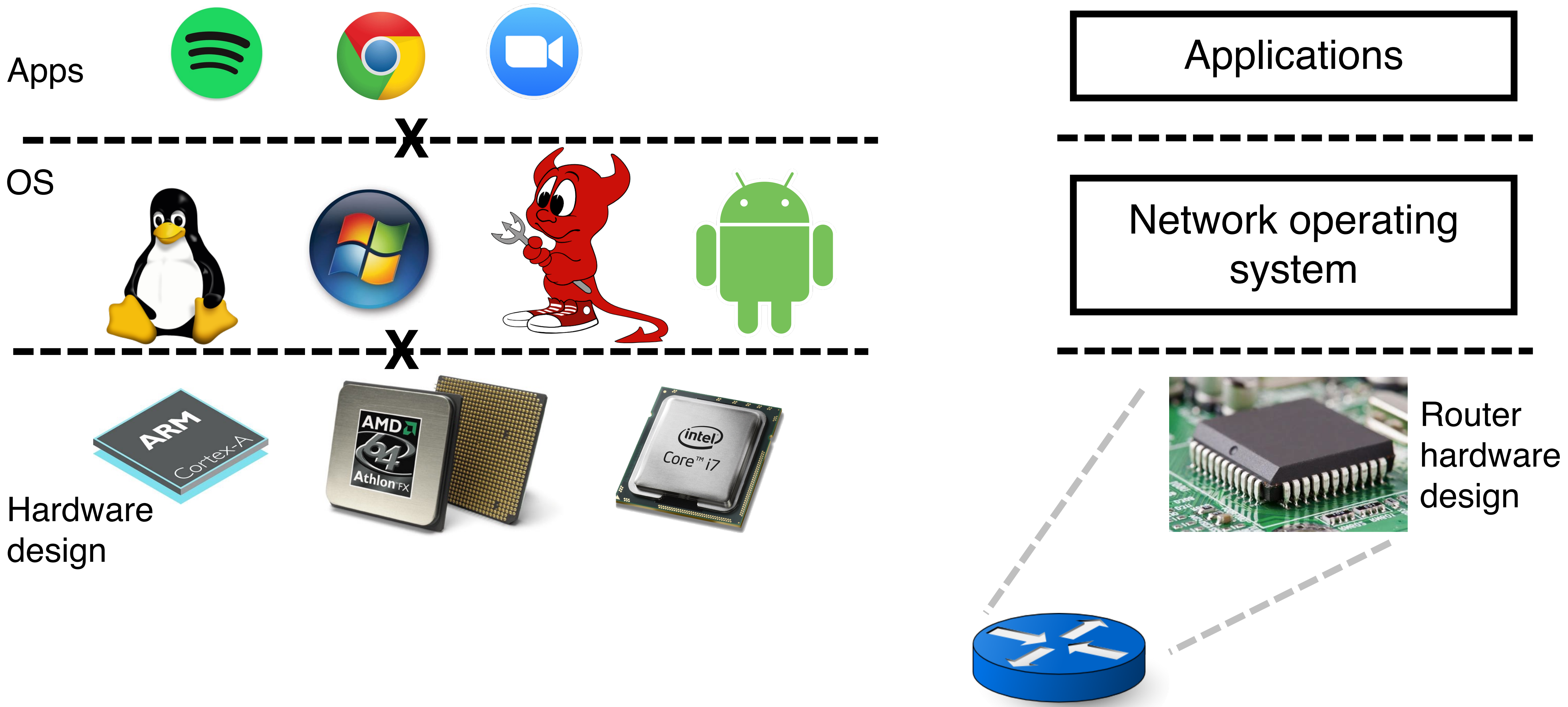


Benefits of SDN

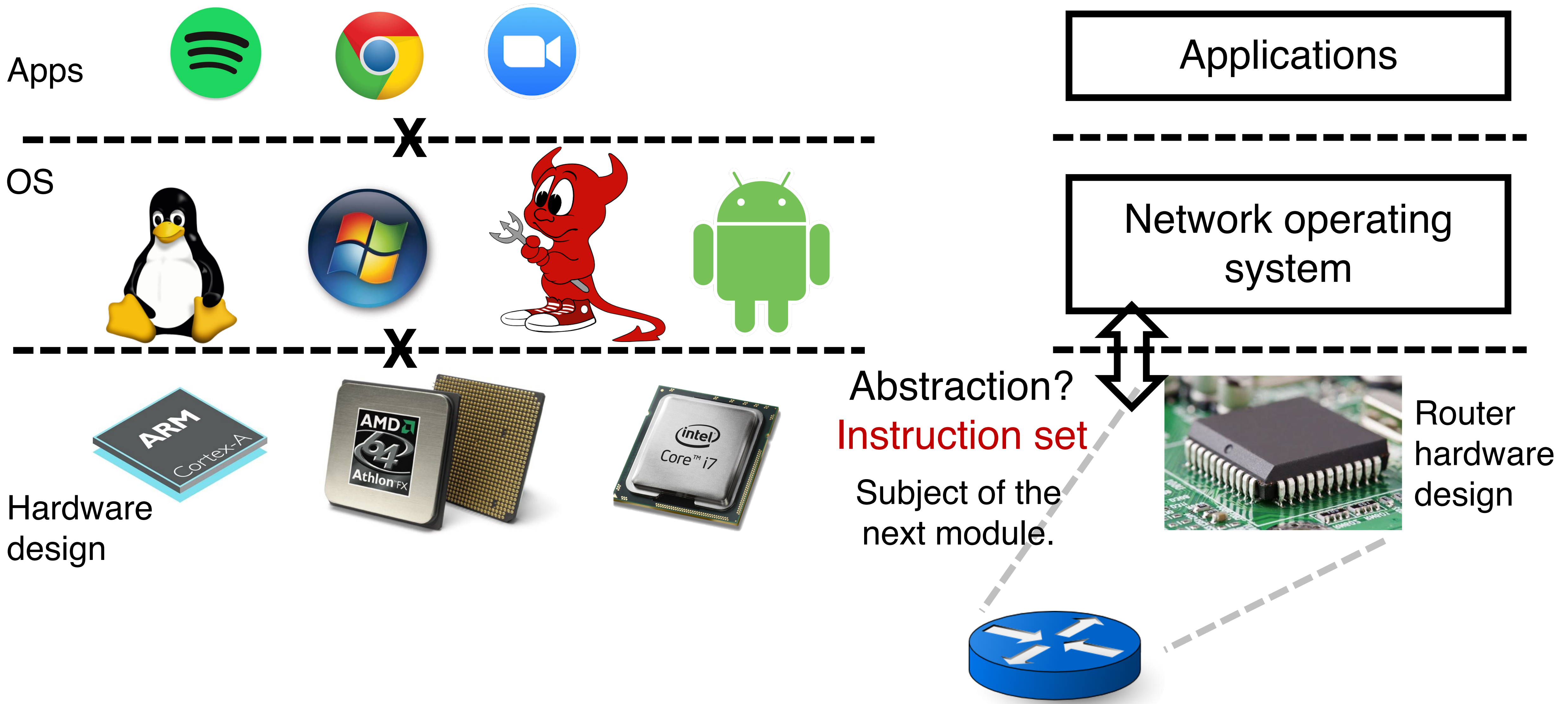
- Software controls the behavior, not distributed protocols
  - Implement flexible forwarding tables
- Interface to data plane is open and well-defined
  - Forward packets using a flexible set of fields
- Impact: like moving from mainframes to commodity PCs!



Design questions in SDN



Design questions in SDN



CS 352
Forwarding with Match-Action

CS 352, Lecture 21.2

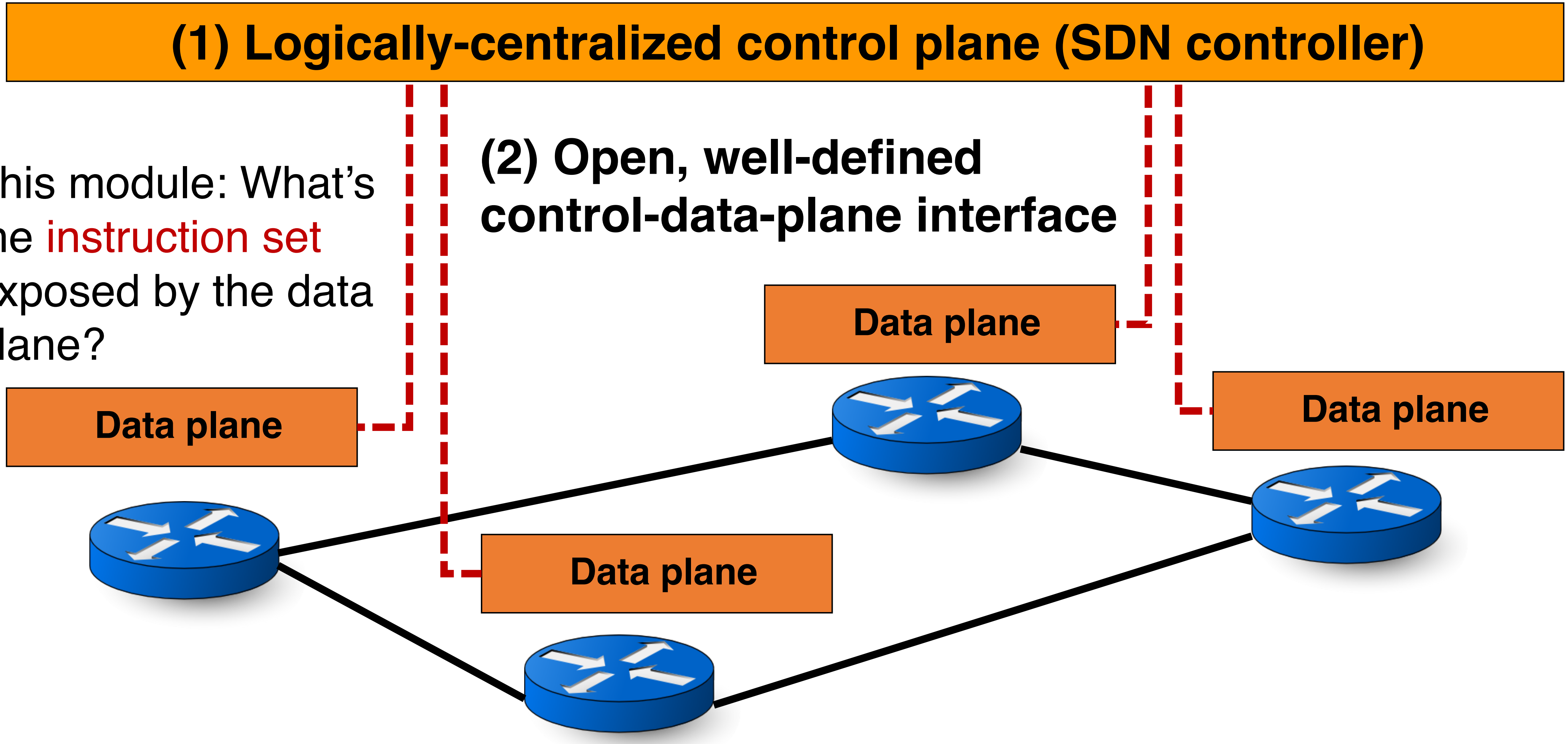
http://www.cs.rutgers.edu/~sn624/352

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Review: SDN

This module: What's the instruction set exposed by the data plane?



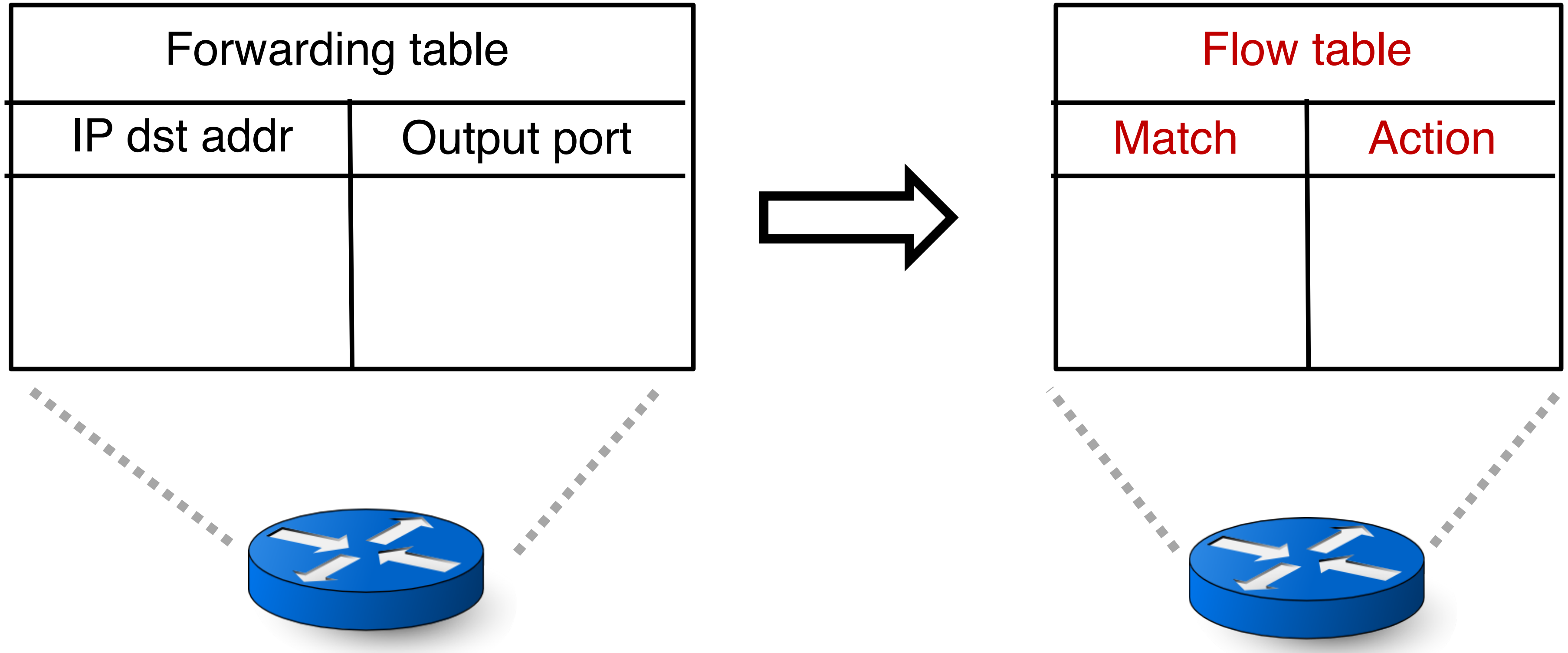
Data plane instruction set: Goals

- Want to match patterns on any part of the packet header
  - Flexible forwarding using any field (much more than dest address!)
  - Match a whole or a subset of each field
  - Allow any forwarding granularity (e.g., src-dst, application, etc.)

- Want to act on packets that match the pattern
  - Forward out a specific port
  - Modify, remove, or add fields
  - Drop packets
  - Count packets

Match-action flow tables

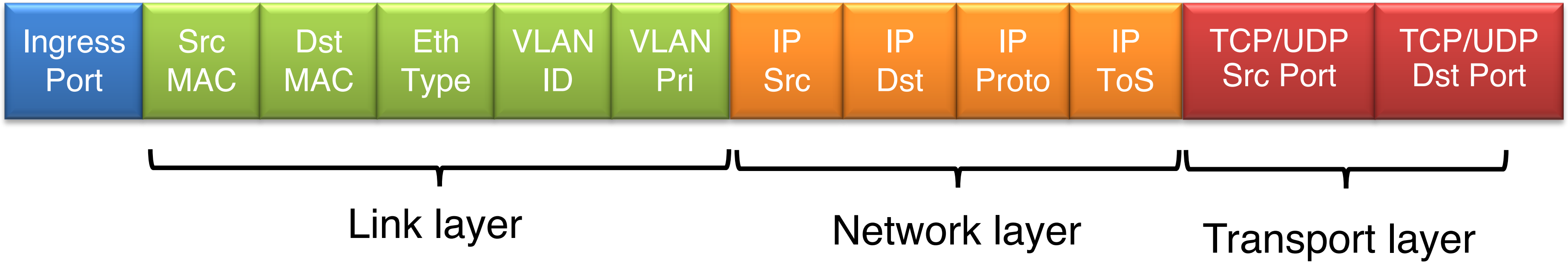
- A generalization of forwarding tables



Match-action flow tables

- For concreteness, let's consider Openflow 1.0
  - A specification of applicable match patterns
  - A specification of actions available in the data plane
- Flow table contains match-action rules
- Match: can occur on any subset of 12 fields

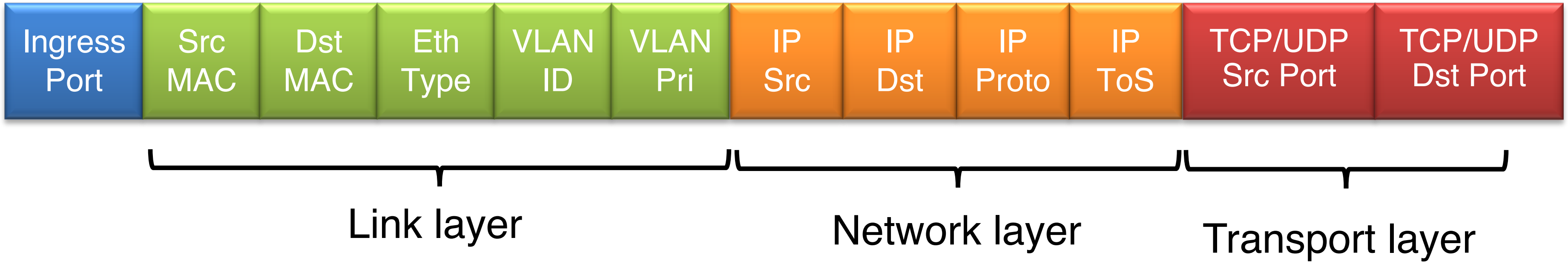
Flow table	
Match	Action



Match-action flow tables

- Examples of matches:
  - Dstip = 192.168.0.1
  - Srcip = 10.0.0/24, IP protocol = TCP, TCP dstport=80
  - DstMAC = 01:4f:3d:10:33:a4
- Any bit among those not listed is wildcarded
  - All values of those bits are accepted when matching

Flow table	
Match	Action

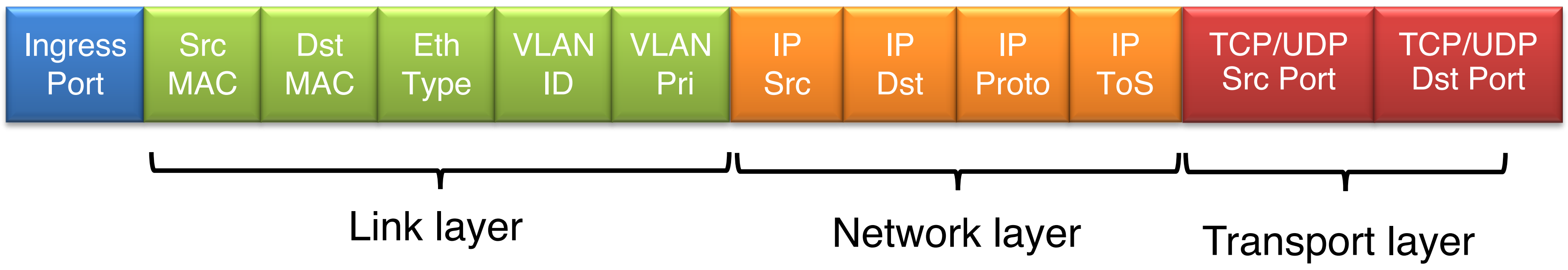


Match-action flow tables

- What if multiple rules match a packet?
  - Srcip = 10.0.0/24, IP proto=TCP, dstport=80
  - Dstip = 192.168.0.1
  - Can both match the same packet!
- Rule priority indicates the rule taking precedence

Flow table

Match	Action



Match-action flow tables

- Actions in Openflow 1.0 are simple

- Forward a packet through a specified port
- Drop the packet
- Send the packet to the SDN controller

Flow table	
Match	Action

Other match-action specifications

- Newer, more general flow table specifications exist
- Openflow 1.0 matches packets with a single table and a fixed set of packet fields (TCP/IP stack)
  - Later versions of Openflow: match using multiple tables
- P4: the ability to match on user-defined packet fields
  - Parse packets in a high-speed router using a program you wrote!
  - Highly flexible: introduce new protocol formats on packets in a network
  - Much more flexible actions: rewrite packets, do arithmetic on fields

Impact of SDN

- Real systems and deployments have been built using SDN principles
- Google: inter-cluster backbone network
  - High network utilization
  - Deterministic network behavior upon failures
  - Ability to pre-plan and test code by mirroring event streams from production networks
- Microsoft: Virtual Filtering Platform (VFP) to enforce policies on VM network traffic
  - Easily “serviceable” using software upgrades
  - High performance with hardware implementations