Overview

Software Defined Networks

- Data plane and control plane separation
- SDN Controllers and Generalized Forwarding
  - Flow Tables vs. Forwarding Tables
- OpenFlow protocol

Network Layer Discussion

- Recalling chapters 4 and 5...
- What are the functions of network layer.
- What are the protocols.
- What might be part of the data plane and what part of the control plane?
- How do the added on functions such as NAT, firewalls and load balancing fit into this discussion?

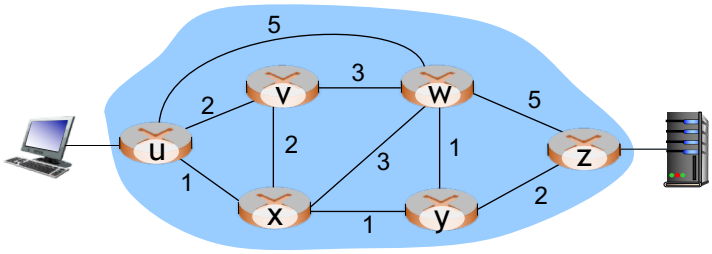
Network Layer Discussion

- A router consists of input ports, output ports, a switching fabric and a routing processor. Which of these functions are implemented in hardware and which are implemented in software?

Network Layer Discussion

- With evolution to the network layer's "data plane" and "control plane," which functions would be implemented in hardware and which in software?

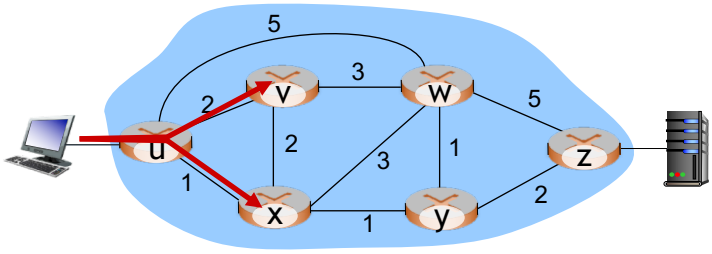
Difficult Traditional Routing



Q: What if network operator wants u-to-z traffic to flow along *uvwz*, x-to-z traffic to flow *xwyz*?

A: Need to define link weights so traffic routing algorithm computes routes accordingly (or need a new routing algorithm)

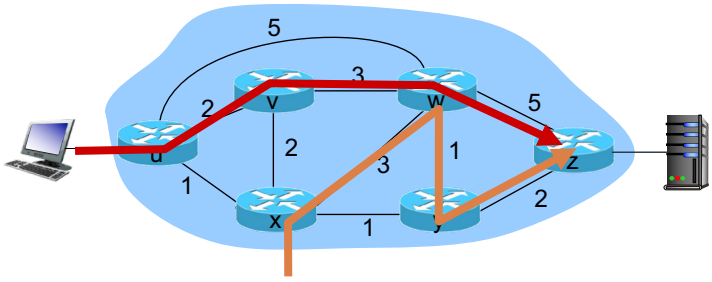
Traditional Routing: Load Balancing



Q: What if network operator wants to split u-to-z traffic along uvwz *and* uxyz?

A: Cannot do this (or need a new routing algorithm)

Difficult Traditional Routing



Q: What if w wants to route gold and red traffic differently?

A: Cannot do this with destination based forwarding, and LS, DV routing

Network-layer functions

Two main network-layer functions are:

- **Forwarding:** move packets from a router's input port to the appropriate output port = data plane
- **Routing:** determine the route taken by packets from source host to destination host = control plane

Network layer: data plane, control plane

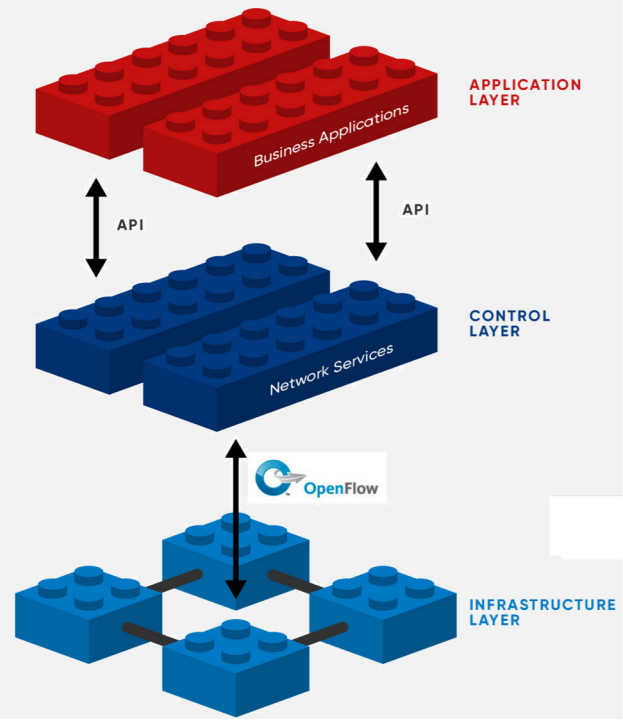
Data plane

- Local, per-router function
- Forwards from input to output port

Control plane

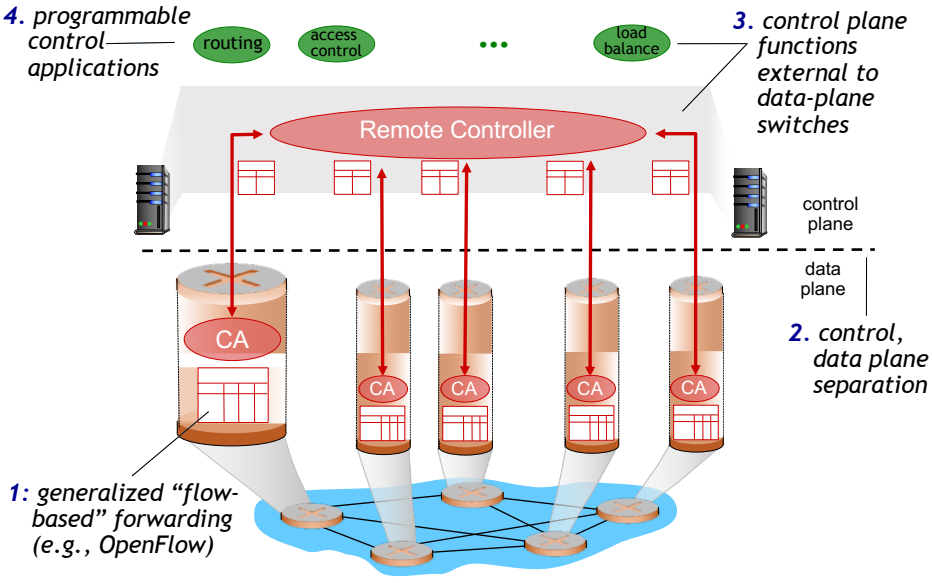
- Network-wide logic
- Determines how datagram is routed from source to destination
- Two control-plane approaches:
  - *Traditional routing algorithms:* implemented in routers
  - *Software-defined networking (SDN)*: implemented in (remote) servers

Software Defined Networking



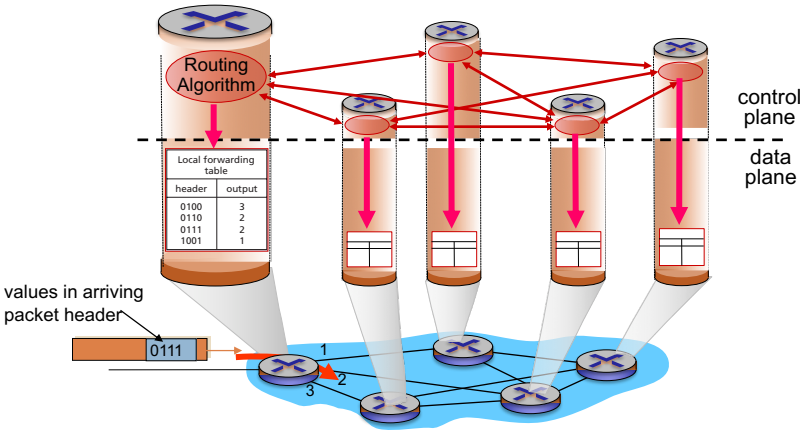
Opennetworking.org

Software defined networking



1) OLD Destination-Based Forwarding

- Individual routing algorithm is run *in each and every router.*
- Routers interact with each other in what we can now think of as a "control plane" to compute forwarding tables
- Traditional approach

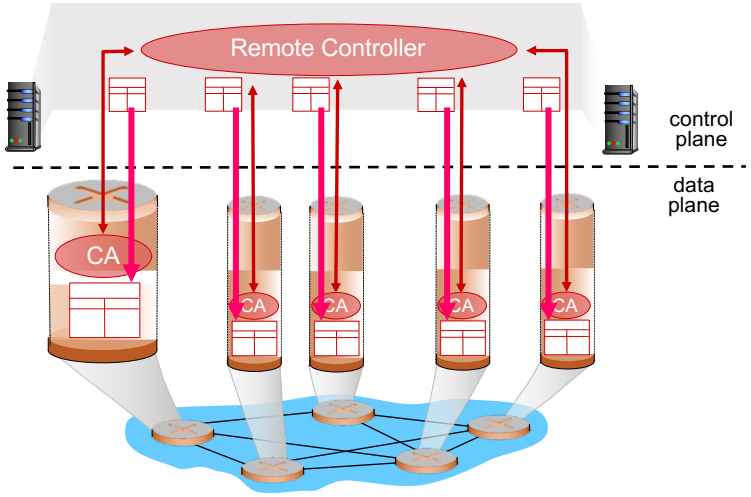


Open Networking Foundation

- The new SDN architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.
- The OpenFlow® protocol is a foundational element for building SDN solutions.

2) New SDN: Centralized Control Plane

A distinct (typically remote) controller interacts with local "control agents" (CAs, *i.e.*, CPUs) in routers to compute forwarding tables



Discussion Question

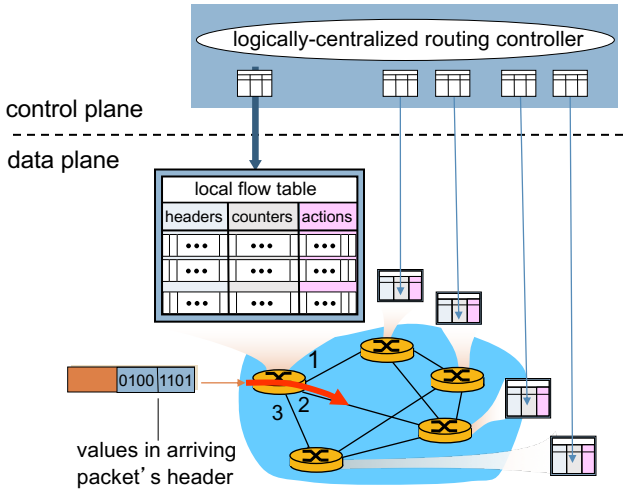
- How does generalized forwarding differ from destination-based forwarding?

Discussion Question

- How does generalized forwarding differ from destination-based forwarding?

3) Generalized Forwarding and SDN

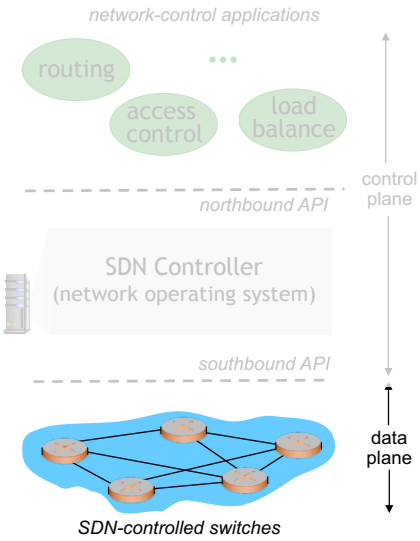
Each router contains a flow table that is computed and distributed by a centralized routing controller



SDN: data plane switches (routers)

Data plane switches

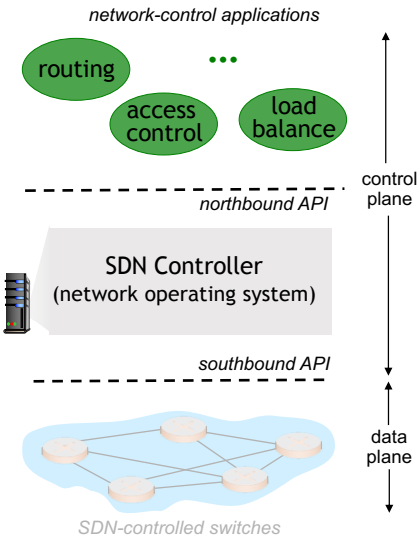
- Implement generalized data-plane forwarding in hardware
- Flow table computed & installed by controller
- Use the OpenFlow Protocol for communicating with the controller



SDN controller

Behaves as a network OS:

- Maintain network state information (links, congestion)
- Interact with network control applications
  - Routing, load balancing, priority access, hackers
- Interact with network switches (routers)



Implementation: OpenFlow protocol

- OLD: *individual* routers contain switching hardware and run Internet standard protocols
  - IP, Link state, Distance vector, OSPF, BGP
- Additional functionality has been evolving: NAT router, firewalls, load balancers, …
- Since 2005: Rethinking network control to directly incorporate these additional functions into a unified network layer framework

OpenFlow protocol

- *match+action:* unifies different kinds of devices

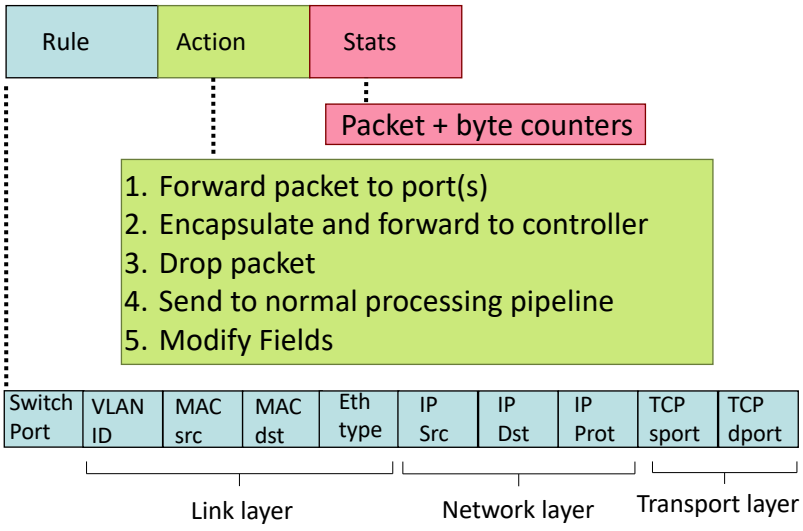
- Router
  - *match:* longest destination IP prefix
  - *action:* forward out a link
- Switch
  - *match:* destination MAC address
  - *action:* forward or flood
- Firewall
  - *match*: IP addresses and TCP/UDP port numbers
  - *action:* permit or deny
- NAT
  - *match:* IP address and port
  - *action:* rewrite address and port

OpenFlow protocol

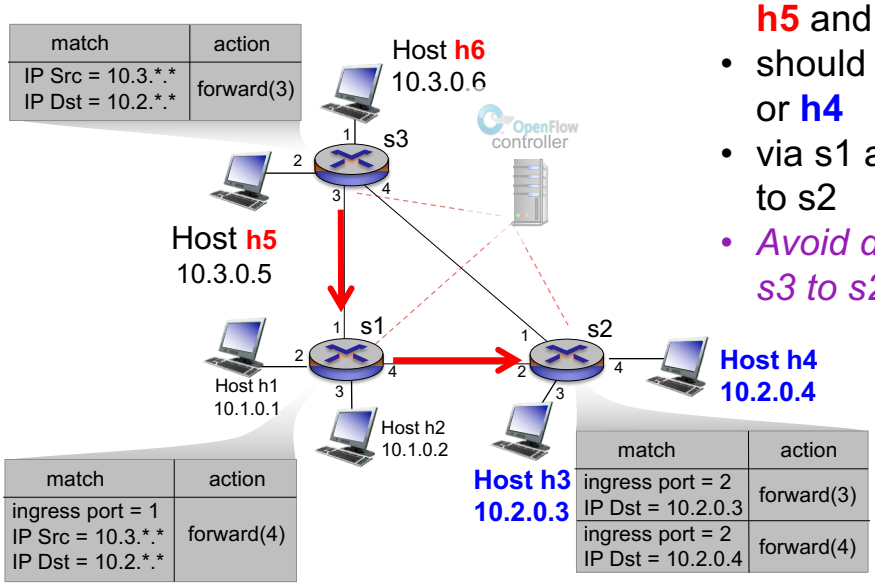
- *Flow* (i.e., route) defined by multiple packet header fields
- Generalized forwarding
  - *Pattern:* match values in up to eleven packet header fields (previously only IP address for traditional forwarding)
  - *Actions for matched packet:* forward, drop, duplicate, modify, matched packet or send matched packet to controller
    - *Prioritize* packets with overlapping packet matches
  - *Counters:* number of packets matched, time since last match…

Flow table in a router (computed and distributed by centralized controller) define each router's match+action rules

OpenFlow: Flow Table Entries



OpenFlow example



Example:

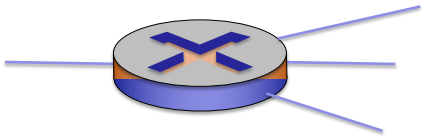
- datagrams from hosts **h5** and **h6**
- should be sent to **h3** or **h4**
- via s1 and from there to s2
- *Avoid direct link from s3 to s2*

OpenFlow Protocol

- *Flow* (i.e., route) defined by packet header fields
- Packet switch performs "match plus action"

1. Src = 1.2.*.*, dest = 3.4.5.* → drop
2. Src = *.*.*.*, dest = 3.4.*.* → forward(2)
3. Src = 10.1.2.3, dest = *.*.*.* → send to controller

* : wildcard



Examples

Destination-based forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port **6**

Destination-based layer 2 (switch) forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	port3

layer 2 frames from MAC address 22:A7:23:11:E1:02 should be forwarded to output port **6**

Examples

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Firewall:

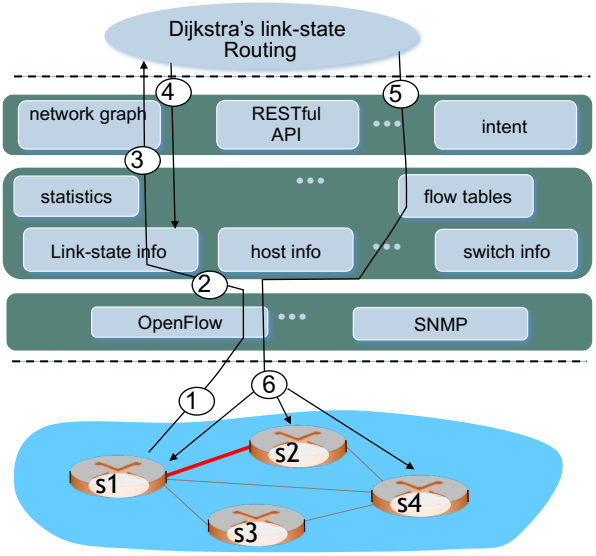
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

Discussion: Messages with OpenFlow

- Two types of messages from a controlled device to a controller:
  - Flow-removed message. Its purpose is to inform the controller that a flow table entry has been removed, for example, by a timeout or as the result of a received modify-state message.
  - Port-status message. Its purpose is to inform the controller of a change in port status.
- Two types of messages from a controller to a controlled device:
  - Modify-state. The purpose is to add/delete or modify entries in the switch's flow table, and to set switch port properties.
  - Read-state. The purpose is to collect statistics and counter values rom the switch's flow table and ports

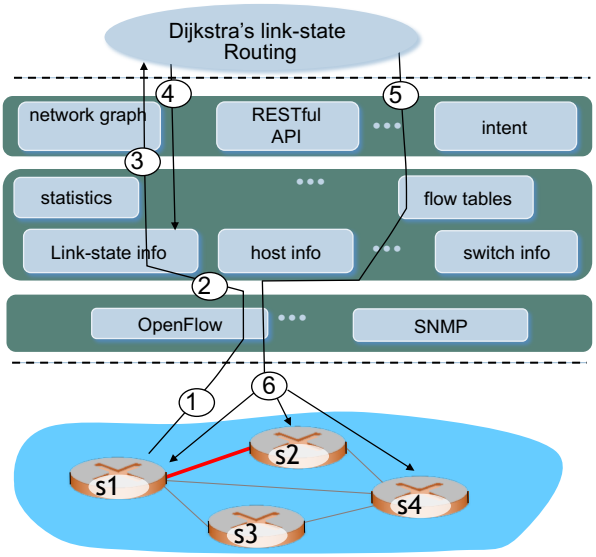
SDN: control/data plane interaction example



1. S1, experiencing link failure using OpenFlow port status message to notify controller
2. SDN controller receives OpenFlow message, updates link status info
3. Dijkstra's routing algorithm application has previously registered to be called when ever link status changes. It is called.
4. Dijkstra's routing algorithm access network graph info, link state info in controller, computes new routes

5-31

SDN: control/data plane interaction example



(5) link state routing app interacts with flow-table-computation component in SDN controller, which computes new flow tables needed

(6) Controller uses OpenFlow to install new tables in switches that need updating

5-32

Review Question

- What is the difference between a forwarding table for destination-based forwarding and OpenFlow's flow table?
- Each entry in the forwarding table of a destination-based forwarding contains
  1. Only an IP header field value and
  2. The outgoing link interface to which a packet (that matches the IP header field value) is to be forwarded.
- Each entry of the flow table in OpenFlow includes
  1. A set of header field values to which an incoming packet will be matched
  2. A set of counters that are updated as packets are matched to flow table entries (number of packets matched, time since last update…)
  3. A set of actions to be taken when a packet matches a flow table entry, such as forward, duplicate, drop, rewrite header field…

Software defined networking

- Explicitly separate data and control plane functions
- Centrally coordinate the individual router forwarding actions
  - "Generalized" forwarding determines output port based on many packet (protocol layer) header fields
  - Create 'flow' table rather than 'forwarding' table
- Implement the control plane function as a separate service, in a remote controller
  - "Software defined" because the controller is implemented via software
  - Evolving toward being able to "program" the Internet

SDN Reflection

- What might be drawbacks and/or limitations of software defined networking
  - Benefits and drawbacks of SDN?
  - Centralized v. Decentralized operations
  - Net neutrality.
  - Improved IDS functionality.

Summary

- Data plane and control plane
- SDN controller
- OpenFlow protocol