Recall the two main network-layer functions:
- **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*

**Two approaches to structuring the network control plane:**
- Individual, distributed router control (traditional)
- Centralized control (software defined networking)
Traditionally, *individual* routers contain switching hardware and run Internet standard protocols (IP, Link state, Distance vector, OSPF, BGP).

Different “middleboxes” for different network layer functions have been evolving: NAT router, firewalls, load balancers, ...

-2005: renewed interest in rethinking the network control plane to directly incorporate the functions of the ‘middleboxes’

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
1. Generalized "flow-based" forwarding (e.g., OpenFlow)

2. Control, data plane separation

3. Control plane functions external to data plane switches

4. Programmable control applications

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)

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)

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

A: Cannot do this with destination based forwarding, and LS, DV routing
**Data plane switches**

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

**SDN controller (network OS):**

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

**Protocol**

- Operates between centralized controller and router (packet switch)
- TCP used to exchange messages
  - Optional encryption
- OpenFlow messages:
  - Controller-to-switch
  - Switch to controller

- Rule
- Action
- Stats

1. Forward packet to port(s)
2. Encapsulate and forward to controller
3. Drop packet
4. Send to normal processing pipeline
5. Modify Fields

<table>
<thead>
<tr>
<th>Switch</th>
<th>VLAN ID</th>
<th>MAC src</th>
<th>MAC dst</th>
<th>Eth type</th>
<th>IP src</th>
<th>IP dst</th>
<th>TCP sport</th>
<th>TCP port</th>
</tr>
</thead>
</table>

Packet + byte counters

**Link layer**

**Network layer**

**Transport layer**
Flow (i.e., route) defined by packet header fields

Generalized forwarding
- Pattern: match values in up to eleven packet header fields (i.e., 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

Destination-based forwarding:

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC Src</th>
<th>MAC Dst</th>
<th>Eth Type</th>
<th>VLAN Id</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP Port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>51.6.0.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>port 6</td>
</tr>
</tbody>
</table>

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

Destination-based layer 2 (switch) forwarding:

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC Src</th>
<th>MAC Dst</th>
<th>Eth Type</th>
<th>VLAN Id</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP Port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>22:A7:23</td>
<td>11:E1:02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>port 3</td>
</tr>
</tbody>
</table>

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

Firewall:

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC Src</th>
<th>MAC Dst</th>
<th>Eth Type</th>
<th>VLAN Id</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP Port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>128.119.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>drop</td>
</tr>
</tbody>
</table>

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

Firewall:

<table>
<thead>
<tr>
<th>Switch Port</th>
<th>MAC Src</th>
<th>MAC Dst</th>
<th>Eth Type</th>
<th>VLAN Id</th>
<th>IP Src</th>
<th>IP Dst</th>
<th>TCP Port</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>22</td>
<td>128.119.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>drop</td>
</tr>
</tbody>
</table>

Do not forward (block) all datagrams destined to TCP port 22
- **Match+action**: unifies different kinds of devices
  - Network layer Router
    - **match**: longest destination IP prefix
    - **action**: forward out a link
  - Link layer Switch
    - **match**: destination MAC address
    - **action**: forward or flood
  - Firewall
    - **match**: IP addresses and TCP/UDP port numbers
    - **action**: permit or deny
  - Home/office NAT router
    - **match**: IP address and port
    - **action**: rewrite address and port

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

**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