Network Data Plane
Network layer

- delivers segments from sending to receiving host
  - sender encapsulates segments into datagrams
  - Receiver de-encapsulates and delivers segments to transport layer

- network layer in every host, every router

- Router examines IP header field in every passing datagram (exception: routers running MPLS)
Key Network-Layer Functions

- **forwarding**: move a packet from router’s input interface to an appropriate output interface

- **routing**: determine route taken by packets from source to destination
  - routing protocols (*intra-AS and inter-AS*) where AS is acronym for “Autonomous System”
  - every AS runs the same inter-AS protocol
Virtual-circuit networks need 3rd function

- Before datagrams can flow, end hosts and routers between them establish a virtual circuit
  - Routers maintain state info
  - Earlier networks designed initially to compete with IP:
    - ATM, frame relay, X.25 (from old to very old)
  - MPLS protocol designed more recently to provide virtual circuits supported by IP routers (typically within the same AS)

- Today, such virtual circuits serve as virtual links in Internet
Network layer: data plane, control plane

**Data plane**
- local, per-router function
- determines how datagram arriving on an input port is *forwarded* to an output port

**Control plane**
- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- main approach:
  - *routing protocols* implemented in routers
- new approach
  - *software-defined networking (SDN)*: implemented in logically centralized server(s)
**Per-router control plane**

Individual routing process *in every router*. They interact by exchanging routing protocol messages.
Logically centralized control plane

A distinct (typically remote) controller interacts with local control agents (CAs). The controller computes routes.
Datagram networks

- IPv4, IPv6
- No network-level concept of “connection” or “flow”
- Each packet forwarded independently using destination host address
  - Packets between same source-dest pair may take different paths
IPv4 addressing: CIDR

Classful addressing (now obsolete): fixed-length subnet portion of 8, 16 or 24 bits

**CIDR:** Classless InterDomain Routing

- subnet portion of address of variable length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address

```
11001000 00010111 00010000 00000000
```

200.23.16.0/23
## Forwarding table

4 billion possible entries

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>11001000 00010111 00010000 00000000</code> through <code>11001000 00010111 00011000 00000000</code></td>
<td>0</td>
</tr>
<tr>
<td><code>11001000 00010111 00010111 11111111</code> through <code>11001000 00010111 00011000 00000000</code></td>
<td>1</td>
</tr>
<tr>
<td><code>11001000 00010111 00011000 00000000</code> through <code>11001000 00010111 00011111 11111111</code></td>
<td>2</td>
</tr>
<tr>
<td><code>11001000 00010111 00011111 11111111</code> otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

1/12/2017
## Longest prefix match

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00011000</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

### Examples

- **DA:** 11001000 00010111 000110110 10100001
  - Which interface?

- **DA:** 11001000 00010111 00011000 10101010
  - Which interface?

An forwarding table in an Internet core router has more than 500,000 IP prefixes.

*Fast implementation uses Ternary Content Addressable Memory (TCAM), prefixes sorted in decreasing order*
Virtual circuits: signaling protocols

- used to set up, maintain, tear down VC
- not used in Internet’s network layer, but may be used underneath the IP layer to provide a virtual link (e.g., MPLS tunnel)
Virtual circuit (VC)

- call setup, teardown for each call before data can flow
- each packet carries a VC identifier which
  - is fixed length and short
  - only needs to be unique for a link
  - is carried in an additional header inserted between link and network layer headers (called layer 2½)

- every router on source-dest path maintains state information for each passing VC
  - incoming and outgoing VC identifiers,
  - resources allocated to VC (bandwidth, buffers)
### VC Forwarding table

**Forwarding table in northwest router:**

<table>
<thead>
<tr>
<th>Incoming interface</th>
<th>Incoming VC #</th>
<th>Outgoing interface</th>
<th>Outgoing VC #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>3</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>63</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>2</td>
<td>17</td>
</tr>
<tr>
<td>1</td>
<td>97</td>
<td>3</td>
<td>87</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Forwarding is fast because short fixed-length VC numbers are used** vs. IP forwarding table with variable-length prefixes. (This is not forwarding in IP layer but it is considered to be in *data plane*.)

- **May have additional state information about service guarantees**
The Internet Network layer

Host, router network layer functions:

- Transport layer: TCP, UDP
  - Routing protocols
    - path selection
    - RIP, OSPF, BGP
  - IP protocol
    - addressing conventions
    - datagram format
    - packet handling conventions
  - ICMP protocol
    - error reporting
    - router “signaling”

Network layer

Link layer

physical layer

1/12/2017

Network Data Plane (S. S. Lam)
**IP datagram format**

- **IP protocol version number**
- **Header length**
- "**Type**" of data
- Max number of remaining hops (decremented at each router)
- Upper layer protocol to deliver payload to

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total datagram length</td>
<td>Number of bytes for the entire datagram</td>
</tr>
<tr>
<td>Fragment/Reassembly</td>
<td>Indicate if datagram is fragmented or not</td>
</tr>
<tr>
<td>Source IP address</td>
<td>32-bit address of the source node</td>
</tr>
<tr>
<td>Destination IP address</td>
<td>32-bit address of the destination node</td>
</tr>
<tr>
<td>Options</td>
<td>Additional data (if any)</td>
</tr>
<tr>
<td>Data</td>
<td>Variable length, typically a TCP or UDP segment</td>
</tr>
</tbody>
</table>

E.g. timestamp, record route taken, specify list of routers to visit.
**IP address prefix: how to get one?**

**A:** Typically, a customer network gets allocated a portion of its provider ISP’s address space

<table>
<thead>
<tr>
<th>ISP's block</th>
<th>11001000 00010111 00010000 00000000</th>
<th>200.23.16.0/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization 0</td>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00010111 00010010 00000000</td>
<td>200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>11001000 00010111 00010100 00000000</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>...</td>
<td>.....</td>
<td>.....</td>
</tr>
<tr>
<td>Organization 7</td>
<td>11001000 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>

1/12/2017

Network Data Plane (S. S. Lam)
Hierarchical addressing: route aggregation allows efficient advertisement of routing information

Organization 0
200.23.16.0/23

Organization 1
200.23.18.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23

Fly-By-Night-ISP

ISPs-R-Us

“Send me anything with address beginning 200.23.16.0/20 ”

“Send me anything with address beginning 199.31.0.0/16 ”

Internet
Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1

Hole(s) in a block of addresses ← reason for longest prefix match

Organization 0
200.23.16.0/23

Organization 2
200.23.20.0/23

Organization 7
200.23.30.0/23

Organization 1
200.23.18.0/23

Fly-By-Night-ISP

"Send me anything with address beginning 200.23.16.0/20"

"Send me anything with address beginning 199.31.0.0/16 or 200.23.18.0/23"

Internet

Network Data Plane (S. S. Lam)
**Access Control List (ACL)**

- lists of rules used in firewalls and for guarding input ports and output ports
- *first match* determines action to take on packet

<table>
<thead>
<tr>
<th>action</th>
<th>source address</th>
<th>dest address</th>
<th>protocol</th>
<th>source port</th>
<th>dest port</th>
<th>flag bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow</td>
<td>222.22/16</td>
<td>outside of 222.22/16</td>
<td>TCP</td>
<td>&gt; 1023</td>
<td>80</td>
<td>any</td>
</tr>
<tr>
<td>allow</td>
<td>outside of 222.22/16</td>
<td>222.22/16</td>
<td>TCP</td>
<td>80</td>
<td>&gt; 1023</td>
<td>ACK</td>
</tr>
<tr>
<td>allow</td>
<td>222.22/16</td>
<td>outside of 222.22/16</td>
<td>UDP</td>
<td>&gt; 1023</td>
<td>53</td>
<td>---</td>
</tr>
<tr>
<td>allow</td>
<td>outside of 222.22/16</td>
<td>222.22/16</td>
<td>UDP</td>
<td>53</td>
<td>&gt; 1023</td>
<td>----</td>
</tr>
<tr>
<td>deny</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
<td>all</td>
</tr>
</tbody>
</table>
Packet filters and transformers in the data plane

- Let the packet universe be the set of all possible bit strings representing all feasible packet headers (or packets). A packet filter let a subset of packets pass through, while dropping all other packets.
- Forwarding tables and ACLs can be modeled as packet filters.
- We next consider network devices that transform packet headers.
NAT: Network Address Translation

rest of Internet → local network 10.0.0/24

138.76.29.7

10.0.0.4

10.0.0.1
10.0.0.2
10.0.0.3

All datagrams leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination within network have 10.0.0/24 addresses for source, destination
NAT: Network Address Translation

**Motivation:** local network uses just one IP address as far as outside world is concerned

- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly addressable/visible by outside world (a security plus).
NAT: Network Address Translation

1: host 10.0.0.1 sends datagram with port number 3345

2: NAT router changes datagram's source addr and port number

3: Reply arrives for 138.76.29.7, 5001

4: NAT router changes datagram's dest addr and port number to 10.0.0.1, 3345

Network Data Plane (S. S. Lam) 24
NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000+ simultaneous connections with a single IP address

- NAT is controversial:
  - routers should only process up to layer 3
    - violates “end-to-end argument”
  - NAT possibility must be taken into account by app designers, e.g., IPsec, P2P applications, etc.
  - address shortage should instead be solved by IPv6
IPv6

- Initial motivation: 32-bit address space soon to be completely allocated.
- Additional motivation:
  - simpler header format to speed up processing/forwarding
  - header change to facilitate QoS
- IPv6 datagram format:
  - fixed-length 40 byte header
  - no fragmentation allowed
**IPv6 Header (Cont)**

*Priority*: identify priority of datagrams within flow or in different apps

*Flow Label*: identify datagrams in same “flow.”
   (concept of “flow” not defined).

*Next header*: identify upper layer protocol for data

---

### IPv6 Header Diagram

```
<table>
<thead>
<tr>
<th>ver</th>
<th>pri</th>
<th>flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>payload len</td>
</tr>
<tr>
<td></td>
<td></td>
<td>next hdr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hop limit</td>
</tr>
</tbody>
</table>
```

- **source address** (128 bits)
- **destination address** (128 bits)

---

**data**

32 bits
Other Changes from IPv4

- **Checksum**: removed entirely to reduce processing time at each hop
- **Options**: allowed, but outside of header, indicated by “Next Header” field
- **ICMPv6**: new version of ICMP
  - additional message types, e.g. “Packet Too Big”
  - including *multicast group management* functions
Transition From IPv4 To IPv6

- Not all routers can be upgraded simultaneously
  - no “flag day”
  - How will the network operate with mixed IPv4 and IPv6 routers?

- **Tunneling**: IPv6 carried as payload in IPv4 datagram among IPv4 routers (also vice versa)
Tunneling

Logical view:

Physical view:
Tunneling

Logical view:

Physical view:

Routers B and E have dual stacks.

In this example, B encapsulates v6 packet in v4 packet.

E extracts v6 packet from v4 packet.
Concept - Tunnel as a virtual link

Many possibilities:

- IPv4 in MPLS tunnel (virtual circuit)
- IPv6 in IPv4 tunnel (previous example)
- IPv4 in IPv6 tunnel
- IPv4 in IPv4 tunnel (new routing path)

...
Link Virtualization: A Network as a Link

Virtual circuits provided by

- ATM, frame relay, which are packet-switching networks in their own right (obsolete)
  - with service models, addressing, routing different from Internet
- A subnet of MPLS capable routers

Each is viewed as a link connecting two IP nodes
Multiprotocol label switching (MPLS)

- **initial goal**: speed up IP forwarding by using fixed-length label (instead of variable-length IP prefix) to do forwarding
  - borrowing ideas from Virtual Circuit (VC) approach
  - MPLS routers insert and remove MPLS header but IP datagram still keeps IP address
MPLS capable routers

- a.k.a. label-switched router
- forward packets to outgoing interface based only on label value (*does not inspect IP address*)
  - Much faster than longest prefix match
  - MPLS forwarding table distinct from IP forwarding tables
- **flexibility**: MPLS forwarding decisions can differ from those of IP
MPLS forwarding tables

There are two predetermined routes from R4 to A.
MPLS applications

- **Fast failure recovery** - rerouting flows quickly to pre-computed backup paths (useful for VoIP)

- **Traffic engineering** - network operator can override IP routing and allocate traffic toward the same destination to multiple paths

- **Resource provision for virtual private networks**
Generalized Forwarding in Software Defined Networking (SDN)

Each router contains a **flow table** that is computed and distributed by a **logically centralized routing controller**.
OpenFlow abstraction

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

- **Router** (layer 3)
  - **match**: longest destination IP prefix
  - **action**: forward to a port

- **Switch** (layer 2)
  - **match**: destination MAC/VLAN address
  - **action**: forward to port or flood

- **Firewall**
  - **match**: IP addresses and protocol field, TCP/UDP port numbers
  - **action**: permit or deny

- **NAT**
  - **match**: IP address and port
  - **action**: rewrite address and port
OpenFlow data plane abstraction

- **flow**: defined by header fields (for link, network, transport layers)
- **generalized forwarding**
  - *Flow entry*: match fields, priority, counters, instructions
  - *Actions*: for matched packet - drop, forward, modify the packet, or send it to controller

*Flow table in a router/switch (computed and distributed by controller) defines router’s match+action rules*
OpenFlow: Flow Table Entries

1. Drop packet
2. Forward packet to port(s)
3. Modify Fields
4. Encapsulate and send to controller

Switch Port | VLAN ID | MAC src | MAC dst | Eth type | IP Src | IP Dst | IP Prot | TCP Src_port | TCP Dst_port
--- | --- | --- | --- | --- | --- | --- | --- | --- | ---

Link layer | Network layer | Transport layer

1/12/2017

Network Data Plane (S. S. Lam)
The End