Outline

- Principles behind Internet protocols
- IP
  - Addressing
  - Forwarding
  - Tunneling
- IP Protocols
  - ICMP
  - DNS
  - ARP
  - DHCP
- Routing
- IPv6
Internetworking

TCP/IP

- Born out of the ARPA net in the late 1960’s
- IP → Internet Protocol
- Transport Protocols
  - TCP → Transmission Control Protocol
  - UDP → User Datagram Protocol
- Open standard, runs on tablets, Smartphones, PC’s to supercomputers and others….

The Internet is more than IP

- A suite of protocols enable today’s Internet
  - IP
  - ARP
  - DHCP
  - DNS
  - ICMP
  - Routing
    - IGP’s (BGP)
    - EGP’s (OSPF)
Internetworking:
Internet Architecture

- Application, e.g., FTP, Telnet, e-mail
  Simple Mail Transfer Protocol, (SMTP)
- Service Provider, end-to-end communications (TCP, UDP or other)
- Internetwork, functions to connect networks and gateways into a total system, (IP)
- Subnetwork, e.g., ARPANET, Ethernet, ATM, Frame Relay, Wireless, others…

Internetworking

Host A
Application
Transport (TCP)
Network (IP)
Path/Physical

Gateway/Router
Network(IP)
Path/Physical

Subnetwork

Host B
Application
Transport (TCP)
Network (IP)
Path/Physical

Subnetwork
Internetworking

Internet Design Principles

- Make sure it works
  - Do prototypes
  - Do not wait until standard documents are completed

- Keep it Simple

- Make clear choices → goal to avoid multiple ways of accomplishing the same thing

- Exploit Modularity → protocol layers

- Expect Heterogeneity
  - Hardware
  - OSs
  - Transmission facilities
  - Applications
Internet Design Principles

- Avoid static options and parameters → best to negotiate or adapt
- Look for “good” design not optimum
- Be strict when sending and tolerant when receiving
- Scalability
  - # users
  - Geographic scope
  - Transmission speeds
- Consider performance and cost

IP Hourglass Architecture

From: Steve Deering
IP Hourglass Architecture

- Why an internet layer?
  - make a bigger network
  - global addressing
  - virtualize network to isolate end-to-end protocols from network details/changes

- Why a *single* internet protocol?
  - maximize interoperability
  - minimize number of service interfaces

- Why a *narrow* internet protocol?
  - assumes least common network functionality to maximize number of usable networks

From: Steve Deering

Problems with IP architecture

- End host assumptions
  - Not mobile
  - Address Binding → Coupling between IP address and end-host

- Security
  - Assumed friendly environment but in reality it is adversarial

- Economic model
  - Original architecture did not have an economic mode → Causes inter-carrier problems with providing QoS

- Narrow hourglass model prevents applications awareness → new applications placing demands for core functionality

- These are currently addressed via point solutions.
Internetworking: IP

IP is connectionless

- No call set up
- PDU’s may be lost
- Hides the subnet technology from the application to allow the use of many different subnet technologies

Internetworking: IPv4

IP packet header

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>8</td>
</tr>
<tr>
<td>IHL</td>
<td>4</td>
</tr>
<tr>
<td>Type of service</td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1</td>
</tr>
<tr>
<td>M</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>Fragment offset</td>
<td></td>
</tr>
<tr>
<td>Time to live</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
</tr>
<tr>
<td>Header checksum</td>
<td></td>
</tr>
<tr>
<td>Source address</td>
<td></td>
</tr>
<tr>
<td>Destination address</td>
<td></td>
</tr>
<tr>
<td>Options (0 or more words)</td>
<td></td>
</tr>
</tbody>
</table>

If no options then routers use “fast path” through hardware
Internetworking: IP
IP packet header-IPv4

- Version, enable transition between different versions of IP datagrams
- IHL= Number of 32 bit words in the header
- ToS= Type of Service, enables the use of priority queuing, basis for IP DiffServ
- Maximum length of IP datagram (including header) = 65,535 bytes
- TTL field decremented at each hop if 0 then drop packet
- Header Checksum verifies header only

Header Checksum

- IP header uses check bits to detect errors in the header
- A checksum is calculated for header contents
- Checksum recalculated at every router, so algorithm selected for ease of implementation in software
- Let header consist of L, 16-bit words, $b_0, b_1, b_2, \ldots, b_{L-1}$
- The algorithm appends a 16-bit checksum $b_L$
Checksum Calculation

The checksum $b_L$ is calculated as follows:
- Treating each 16-bit word as an integer, find
  \[ x = b_0 + b_1 + b_2 + \ldots + b_{L-1} \mod 2^{15-1} \]
- The checksum is then given by:
  \[ b_L = -x \mod 2^{15-1} \]
- This is the 16-bit 1’s complement sum of the $b$’s
- If checksum is 0, use all 1’s representation (all zeros reserved to indicate checksum was not calculated)
- Thus, the headers must satisfy the following pattern:
  \[ 0 = b_0 + b_1 + b_2 + \ldots + b_{L-1} + b_L \mod 2^{15-1} \]

In IPv4 Routers need to recalculate the checksum because the header changes.
Why does the header change at each router?

Differentiated Services - Concept

IP DiffServ

- Provide scalable service discrimination in the Internet
- No need to maintain per flow state or doing per hop signaling.
- Employs a small set of building blocks from which a variety of services can be built.
- These services can be either end-to-end or intra domain.
Differentiated Services - Concept

Differentiated Services provide a wide range of services through:

- Setting bits in the ToS at network edges and administrative boundaries,
- Using those bits to determine how packets are treated (Queued) by the routers inside the network, and
- Conditioning the marked packets at network boundaries in accordance with the requirements of each service.

Internetworking: IP Services

- Fragmentation and reassembly
  - If PDU size > MTU (Maximum Transfer Unit) for subnet the IP must fragment the PDU and reassemble at the destination
    - Ethernet ~1500 byte PDU's
Internetworking:
IP Addressing

- Every host and router interface has an IP address
- 32 bits/address ⇒ 4.295x10^9 addresses (IPv4)
- 128 bits/address ⇒ 3.4x10^34 addresses (IPv6)

Addresses contains
  - Host ID
    - Identifies a unique host on a network
  - Network ID
    - Identifies the network that the host is connected to
  - Initially five formats for IP addresses (Classfull IP Addressing)

Internetworking:
Classfull IP Addressing

<table>
<thead>
<tr>
<th>Class</th>
<th>Net_id</th>
<th>Range of host addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 0</td>
<td>Network</td>
<td>1.0.0.0 to 127.255.255.255</td>
</tr>
<tr>
<td>B 10</td>
<td>Network</td>
<td>128.0.0.0 to 191.255.255.255</td>
</tr>
<tr>
<td>C 110</td>
<td>Network</td>
<td>192.0.0.0 to 223.255.255.255</td>
</tr>
<tr>
<td>D 1110</td>
<td>Multicast address</td>
<td>224.0.0.0 to 239.255.255.255</td>
</tr>
<tr>
<td>E 1110</td>
<td>Reserved for future use</td>
<td>240.0.0.0 to 247.255.255.255</td>
</tr>
</tbody>
</table>

Internetworking:
Classfull IP Addressing

- Class A addresses
  - 127 Class A addresses
  - \(2^{24}\) hosts (16.77 Million) / Class A addresses
- Class B networks
  - 16383 Class B addresses (address ‘0’ is reserved)
  - \(2^{16}\) (65K) hosts/addresses
    - KU has a class B address
- Class C addresses
  - 2,097,152 Class C addresses (‘0’ and ‘2,907,151’ reserved)
    - 256 hosts/network.
- Class D is used for multicasting

Internetworking:
IP Addressing Notation

- 32 bits = 4 bytes
- Represent each byte by a decimal
- Example: 11.55.31.84
  - 00001011 . 00110111 . 00011111 . 1010100
  - 11 55 31 84
  - This is a Class A address
    - 0001011 is the network address
    - 001101111001111010100 is the host address
- Example: 129.237.125.27 is a KU class B address
Internetworking: Classfull IP Addressing Notation

Class B Address

The "xxx" represents the host-number field of the address which is assigned by the local network administrator.
Internetworking: Subnetting

Subnetting divides the standard classful host number into:
- Subnet number
- Host – number

Enables routing on subnet -number for more efficient routing
Provides an additional level of addressing hierarchy

Special addresses (Can not use for host):
- Address with host ID=all 0s refers to the network
- Address with host ID=all 1s refers to a broadcast packet, i.e.,
- it goes to all host on the network
### Internetworking: Subnetting

- **Base Net:** 11000001.00000001.00000001.00000000 = 193.1.1.0/24
- **Subnet #0:** 11000001.00000001.00000001.00000000 = 193.1.1.0/27
- **Subnet #1:** 11000001.00000001.00000001.00100000 = 193.1.1.32/27
- **Subnet #2:** 11000001.00000001.00000001.10000000 = 193.1.1.64/27
- **Subnet #3:** 11000001.00000001.00000001.10100000 = 193.1.1.96/27
- **Subnet #4:** 11000001.00000001.00000001.11000000 = 193.1.1.128/27
- **Subnet #5:** 11000001.00000001.00000001.11100000 = 193.1.1.160/27
- **Subnet #6:** 11000001.00000001.00000001.11110000 = 193.1.1.192/27
- **Subnet #7:** 11000001.00000001.00000001.11111000 = 193.1.1.224/27

**Number of host on /27:**

- $32 - 27$ = number of available bits = $5 (2^5)$
- $32 - 1$ (all 0’s host ID reserved for the network) = 31
- $31 - 1$ (all 1’s host ID reserved for broadcast) = 30

Number of host on /27 = 30

---

**Internetworking: Subnetting**

To identify the Subnet the router uses a “subnet mask”

Subnet mask has a “1” in each bit position of the address except the host ID

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Subnet Mask</th>
<th>Extended Network Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>130.5.5.25</td>
<td>255.255.255.0</td>
<td>11111111.11111111.11111111.00000000</td>
</tr>
</tbody>
</table>

**Logical AND** to find network prefix
**Internetworking: CIDR**

- Classless Interdomain Routing (CIDR)
- Removes the classful address restriction
- Extends the concept of subnetting to routers inside the Internet
- Partially relieves address exhaustion, allows more efficient use of IPv4 address space
- Supports deployment of arbitrarily sized networks
- Aggregation allows reduction in the size of routing tables


---

**Internetworking: CIDR:**

<table>
<thead>
<tr>
<th>CIDR Prefix Length</th>
<th>Dotted-Decimal</th>
<th># Individual Addresses</th>
<th># of Classful Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>/13</td>
<td>255.248.0.0</td>
<td>512 K</td>
<td>8 Bs or 2048 Cs</td>
</tr>
<tr>
<td>/14</td>
<td>255.252.0.0</td>
<td>256 K</td>
<td>4 Bs or 1024 Cs</td>
</tr>
<tr>
<td>/15</td>
<td>255.254.0.0</td>
<td>128 K</td>
<td>2 Bs or 512 Cs</td>
</tr>
<tr>
<td>/16</td>
<td>255.255.0.0</td>
<td>64 K</td>
<td>1 B or 256 Cs</td>
</tr>
<tr>
<td>/17</td>
<td>255.255.128.0</td>
<td>32 K</td>
<td>128 Cs</td>
</tr>
<tr>
<td>/18</td>
<td>255.255.192.0</td>
<td>16 K</td>
<td>64 Cs</td>
</tr>
<tr>
<td>/19</td>
<td>255.255.224.0</td>
<td>8 K</td>
<td>32 Cs</td>
</tr>
<tr>
<td>/20</td>
<td>255.255.240.0</td>
<td>4 K</td>
<td>16 Cs</td>
</tr>
<tr>
<td>/21</td>
<td>255.255.248.0</td>
<td>2 K</td>
<td>8 Cs</td>
</tr>
<tr>
<td>/22</td>
<td>255.255.252.0</td>
<td>1 K</td>
<td>4 Cs</td>
</tr>
<tr>
<td>/23</td>
<td>255.255.254.0</td>
<td>128 K</td>
<td>1 C</td>
</tr>
<tr>
<td>/24</td>
<td>255.255.255.0</td>
<td>512 1/2 C</td>
<td></td>
</tr>
<tr>
<td>/25</td>
<td>255.255.255.128</td>
<td>128 1/4 C</td>
<td></td>
</tr>
<tr>
<td>/26</td>
<td>255.255.255.192</td>
<td>64 1/8 C</td>
<td></td>
</tr>
<tr>
<td>/27</td>
<td>255.255.255.224</td>
<td>32 1/16 C</td>
<td></td>
</tr>
</tbody>
</table>

# of subnets

# hosts/

Subnet-2

Internetworking:

IP Addressing

- Domain Name Service (DNS)
  - Names ↔ IP translation
- Non-numeric form for IP addresses host naming
  - host.department.institution.domain
- Names are long and human understandable
  - Wastes space to carry them in packet headers
  - Hard to parse
- Numeric addresses are shorter and machine understandable
  - If fixed size, easy to carry in headers and parse
Internetworking: IP Addressing

- Indirection
  - Multiple names may point to same address
  - Can move a machine and just update the resolution table
- Names also constructed in hierarchy
- Domain name system (DNS) contain tables to convert:
  - host.department.institution.domain to a 32-bit address

Internetworking: IP Addressing-Common Domains

- Top level domains to naming authorities (see Internet Corporations for Assigned Names and Numbers- ICANN; http://www.icann.org)
  - .edu
  - .com
  - .mil
  - .org
  - .gov
  - .net
  - .biz
  - [country].il, .uk, .au
  - More
- DNS is a real time distributed data base
- Records in the DNS database include:
  - A → Maps name to IP
  - PTR (pointer) → Maps name to name (alias)
  - MX (mail exchange) → Maps name to name of mail server
Internetworking:

- IP Addressing -> Example
  gauss.eecs.ku.edu => 129.237.125.220

- A different IP address can be assigned to each physical interface on a host, note a physical interface will have a unique physical address, for IEEE 802.3 this is a 48-bit number

- A host can have multiple IP addresses: multihomed

Internet Control Protocols

- Internet Control Message Protocol: ICMP
  ➢ Purpose: Report unexpected events & test
### Principal ICMP message types

<table>
<thead>
<tr>
<th>Message type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination unreachable</td>
<td>Packet could not be delivered</td>
</tr>
<tr>
<td>Time exceeded</td>
<td>Time to live field hit 0</td>
</tr>
<tr>
<td>Parameter problem</td>
<td>Invalid header field</td>
</tr>
<tr>
<td>Source quench</td>
<td>Choke packet</td>
</tr>
<tr>
<td>Redirect</td>
<td>Teach a router about geography</td>
</tr>
<tr>
<td>Echo request</td>
<td>Ask a machine if it is alive</td>
</tr>
<tr>
<td>Echo reply</td>
<td>Yes, I am alive</td>
</tr>
<tr>
<td>Timestamp request</td>
<td>Same as Echo request, but with timestamp</td>
</tr>
<tr>
<td>Timestamp reply</td>
<td>Same as Echo reply, but with timestamp</td>
</tr>
</tbody>
</table>


### Host Configuration: Dynamic Host Configuration Protocol (DHCP)

- Every host needs an IP address
- Initial approach:
  - System Administrators manually configure host IP information (static)
- Management nightmare for large enterprise networks
- Management nightmare for “always on” public networks, e.g., cable modem systems
Host Configuration: Dynamic Host Configuration Protocol (DHCP)

- Solution: DHCP
- DHCP server maintains pool of IP addressed that are distributed on demand.
- The protocol governs the distribution of addresses
- DHCP enables the scaling of network management
PHY and IP Addresses

- Network interface PHY address (PHY or MAC)
- IP address
- Socket

PHY and IP Addresses and Networks

- Network interface PHY address (PHY or MAC)
- Host IP address
Internet Control Protocols: ARP (On Ethernet)

- Address Resolution Protocol (ARP)
  - Purpose: Map IP address to physical address (or link layer address)
- Want to talk to 129.237.116.75
- Send "broadcast" message: **Who owns 129.237.116.75**
- 129.237.116.75 will respond: **I do and here is my physical address**
- Reverse ARP (RARP)
- Maps Physical address into IP address

Tunneling

- A tunnel is a *virtual* point-to-point connection between a pair of nodes through an arbitrary number of networks
- Packet entering a tunnel is encapsulated into another packet
- Packet leaving the tunnel is de-encapsulated restoring the original packet format
### Tunneling: Example of IP-IP tunnel

**Goals:**
- Enable the use of Private Addressing Scheme inside enterprises
- Enable security, i.e., secure tunnel forming a Virtual Private Network (VPN)


### VPN

(a) A leased-line private network.  (b) A virtual private network.

Tunneling: Benefits & Penalties

- **Benefits**
  - Enables "virtual private networks"
  - Allows address independence in the enterprise
  - Enhances security (with encryption)
  - Enables gateway functionality, carry other PDUs formats (protocols) across an IP network

- **Penalties**
  - Increased overhead: packets are longer
  - Performance of edge routers: routers must add and remove encapsulation
  - Management: tunnel set up

Routing vs. Forwarding

[Diagram showing routing algorithm and local forwarding table]

- Value in arriving packet's header

Routing vs. Forwarding

- **Forwarding:**
  - Process of reading packet header, getting the destination address, looking up output port in forwarding table and send packet on its way
- **Routing:**
  - Process of building the forwarding table
- **Forwarding is local**

Routing vs. Forwarding

- **Routing is**
  - distributed (routers communicate using a routing protocol)
  - “learns” the network topology
  - finds “shortest” path
- **Routing is like exploring,**
  - Send explorers packets
  - They return with information of possible paths
  - Then calculate the best way to get from “here to there”
Delivery of an IP datagram

View at the data link layer (the physical interconnections):
- Internetwork is a collection of LANs or point-to-point links or switched networks that are connected by routers


Delivery of an IP datagram

View at the IP layer:
- An IP network is a logical entity with a network number
- We represent an IP network as a “cloud”
- The IP delivery service takes the view of clouds, and ignores the data link layer view

Tenets of end-to-end delivery of datagrams

The following conditions must hold so that an IP datagram can be successfully delivered:

- The network prefix of an IP destination address must correspond to a unique data link layer network (=LAN or point-to-point link or switched network).
- Routers and hosts that have a common network prefix must be able to directly exchange IP datagrams using a data link protocol (e.g., MAC, Ethernet, PPP).
- Every data link layer network must be connected to at least one other data link layer network via a router.


Forwarding Tables

- Each router and each host keeps a forwarding table which tells the router how to process an incoming packet.
- Main columns:
  1. **Destination address**: network where is the IP datagram going to?
  2. **Next hop**: how to send the IP datagram?
  3. **Interface**: what is the output port?
- Next hop and interface column can often be summarized as one column.
- Forwarding tables are set so that datagrams gets closer to its destination.

Forwarding Tables

Forwarding table of a host or router
IP datagrams can be directly delivered ("direct") or is sent to a router ("R4")
Will also include a default Next Hop, interface

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.0.0/24</td>
<td>direct</td>
<td>eth0</td>
</tr>
<tr>
<td>10.1.2.0/24</td>
<td>direct</td>
<td>eth0</td>
</tr>
<tr>
<td>10.2.1.0/24</td>
<td>R4</td>
<td>serial0</td>
</tr>
<tr>
<td>10.3.1.0/24</td>
<td>direct</td>
<td>eth1</td>
</tr>
<tr>
<td>20.1.0.0/16</td>
<td>R4</td>
<td>serial0</td>
</tr>
<tr>
<td>20.2.1.0/28</td>
<td>R4</td>
<td>serial0</td>
</tr>
</tbody>
</table>

Router

eth0
eth1
serial0

3 PHY Interfaces
eth0
eth1
serial0

Delivery with forwarding tables

Note: Not showing the interface in the table

Delivery of IP datagrams

- There are two distinct processes to delivering IP datagrams:
  1. **Forwarding**: How to pass a packet from an input interface to the output interface?
  2. **Routing**: How to find and setup the forwarding tables?

- Forwarding must be done as fast as possible:
  - on routers, is often done with support of hardware
  - on PCs, is done in kernel of the operating system

- Routing is less time-critical
  - Filling in the forwarding table using learned information
  - On a PC, routing is done as a background process

Processing of an IP datagram in IP

- Processing of IP datagrams is very similar on an IP router and a host
- Main difference:
  "IP forwarding" is enabled on router and disabled on host
- IP forwarding enabled
  ⇒ if a datagram is received, but it is not for the local system, the datagram will be sent to a different system
- IP forwarding disabled
  ⇒ if a datagram is received, but it is not for the local system, the datagram will be dropped


Processing of an IP datagram at a router

1. IP header validation
2. Process options in IP header
3. Parsing the destination IP address
4. Forwarding table lookup
5. Decrement TTL
6. Perform fragmentation (if necessary)
7. Calculate checksum
8. Transmit to next hop
9. Send ICMP packet (if necessary)

Type of forwarding table entries

- **Network route**
  - Destination addresses is a network address (e.g., 10.0.2.0/24)
  - Most entries are network routes

- **Host route**
  - Destination address is an interface address (e.g., 10.0.1.2/32)
  - Used to specify a separate route for certain hosts

- **Default route**
  - Used when no network or host route matches
  - The router that is listed as the next hop of the default route is the default gateway (for Cisco: “gateway of last resort”)

- **Loopback address**
  - Routing table for the loopback address (127.0.0.1)
  - The next hop lists the loopback (lo0) interface as outgoing interface

Forwarding table lookup: Longest Prefix Match

**How to forward** → 128.143.71.21

- **Longest Prefix Match**: Search for the forwarding table entry that has the longest match with the prefix of the destination IP address

1. Search for a match on all 32 bits
2. Search for a match for 31 bits
3. Search for a match on 30 bits
4. Search for a match on 29 bits
5. ....
32. Search for a match on 0 bits

<table>
<thead>
<tr>
<th>Destination address</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/8</td>
<td>R1</td>
</tr>
<tr>
<td>128.143.0.0/16</td>
<td>R2</td>
</tr>
<tr>
<td>128.143.64.0/20</td>
<td>R3</td>
</tr>
<tr>
<td>128.143.192.0/20</td>
<td>R3</td>
</tr>
<tr>
<td>128.143.71.0/24</td>
<td>R4</td>
</tr>
<tr>
<td>128.143.71.55/32</td>
<td>R3</td>
</tr>
<tr>
<td>default</td>
<td>R5</td>
</tr>
</tbody>
</table>

Host route, loopback entry → 32-bit prefix match
Default route is represented as 0.0.0.0/0 → 0-bit prefix match

The longest prefix match for 128.143.71.21 is for 24 bits with entry 128.143.71.0/24

Datagram will be sent to R4

### Forwarding table

<table>
<thead>
<tr>
<th>Destination Address Range</th>
<th>Output Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/21 0</td>
</tr>
<tr>
<td>11001000 00010111 00010111 11111111</td>
<td>104.23.24.0/24 1</td>
</tr>
<tr>
<td>11001000 00010111 00011000 00000000</td>
<td>104.23.25.0/21 2</td>
</tr>
<tr>
<td>otherwise (default)</td>
<td>3</td>
</tr>
</tbody>
</table>


### Longest prefix matching

<table>
<thead>
<tr>
<th>Prefix Match</th>
<th>Output Link Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>11001000 00010111 00010110</td>
<td>0</td>
</tr>
<tr>
<td>11001000 00010111 00010000</td>
<td>1</td>
</tr>
<tr>
<td>11001000 00010111 00011000</td>
<td>2</td>
</tr>
<tr>
<td>otherwise</td>
<td>3</td>
</tr>
</tbody>
</table>

**Examples**

| DA: 11001000 00010111 00011000 10100000 | interface? 0 |
| DA: 11001000 00010111 00011000 10101010 | interface? 1 |

DA = destination address

Forwarding process finds longest prefix in forwarding table and sends packet out interfaces that matches the longest prefix consistent with subnetting.

Example: IP Forwarding

Table:

<table>
<thead>
<tr>
<th>Net Address</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>201.4.16.0/22</td>
<td>Local</td>
<td>eth0</td>
</tr>
<tr>
<td>180.70.65.26</td>
<td>Router 1</td>
<td>eth0</td>
</tr>
<tr>
<td>180.70.65.36</td>
<td>Router 1</td>
<td>eth0</td>
</tr>
<tr>
<td>180.70.65.36</td>
<td>Router 2</td>
<td>eth0</td>
</tr>
<tr>
<td>223.1.1.0</td>
<td>Default</td>
<td>eth0</td>
</tr>
</tbody>
</table>

Diagram:

Network Layer...
Autonomous Systems (AS)

- Global Internet viewed as collection of autonomous systems.
- **Autonomous system (AS)** is a set of routers or networks administered by a single organization.
- Same routing protocol need not be run within an AS and between ASs.
- But, to the outside world, an AS should present a *consistent picture of what ASs are reachable* through it.
- **Stub AS**: has only a single connection to the outside world.
- **Multihomed AS**: has multiple connections to the outside world, but refuses to carry transit traffic.
- **Transit AS**: has multiple connections to the outside world, and can carry transit and local traffic.

From: Communication Networks: Fundamentals Concepts and Key Architectures
Authors: A. Leon-Garcia and I. Widjaja
AS Numbers (ASN)

- AS numbers are 16 bits
- Internet Assigned Numbers Authority (IANA) gives ASNs to regional internet registry (RIR), RIRs give ASNs to, ISPs and end-user organizations.
- RIRs
  - African Network Information Centre (AfriNIC) for Africa
  - American Registry for Internet Numbers (ARIN) for the United States, Canada, and several parts of the Caribbean region.
  - Asia-Pacific Network Information Centre (APNIC) for Asia, Australia, New Zealand, and neighboring countries
  - Latin America and Caribbean Network Information Centre (LACNIC) for Latin America and parts of the Caribbean region
  - Réseaux IP Européens Network Coordination Centre (RIPE) for Europe, the Middle East, and Central Asia
Internet Control Protocols: Routing

- Routing protocols are used to "set-up" the forwarding tables in IP routers
- Routing protocols "learn about the "state of the network" and communicate routing information between routers
- Routing protocols implement part of the IP "Control Plane"

Internet Routing Protocols

- Interior Gateway Router (IGP) Protocol
  - Routing protocol within "autonomous" systems, e.g., KU
    - Open Shortest Path First (OSPF)
    - Router Information Protocol (RIP)
  - An AS is usually own/controlled by one organization, e.g., an ISP
- Exterior Gateway Routing (EGP) Protocol
  - Routing between "autonomous" systems
    - Border Gateway Protocol (BGP)
  - EGP must work BETWEEN organizations, e.g., Sprint and ATT
- As of Summer 2010 there were over 35,000 AS’s.
Routing Protocols: Issues

- Coordinate a path (route)
- Route discovery
  - What does the network look like \(\rightarrow\) topology?
  - What routes are available?
- What information needs to be shared?
  - What are the characteristics of the paths, e.g., capacity, delay, loss, jitter, etc.
- How is network state information shared, e.g., flooding?
- How is network state information used?

The Routing Problem

- Routing algorithms attempt to build forwarding tables to "optimally" route traffic based on some knowledge of the network topology and state (e.g., link delay and loss)
- Practical problems:
  - Which shortest path algorithm to use?
  - How to learn the topology and network state?
  - How define an optimization metric (length or "distance")?
    - The bubble, change paths to reduce delays for some traffic may worsen performance for other traffic.
  - How to respond to:
    - Network element failures
    - Link failures
    - Changes in traffic, e.g., congestion
  - How to establish policies between AS’s?
- Different routing protocols answer these questions in different ways.
Routing-Shortest Path Algorithm

- What is distance (link weight)
  - Propagation delay \( \propto \) Physical distance
  - Number of hops, i.e., number of routers the packet hits between the source and destination
  - Other “cost”
    - Cost in $ 
    - Cost in “congestion”
    - Number of hops
    - Delay
    - Available capacity
    - Administratively set

Routing-Shortest Path Algorithm

- Shortest Path Algorithm finds the minimum distance path between nodes
- Input
  - Topology
  - Link “distances” (link weights)
- Output is a forwarding table
Routing-Shortest Path Algorithm

Example:

Find the shortest path routing table for all nodes

Exhaustive Search

- Find Shortest path from A to D
- List all possible paths and their lengths

<table>
<thead>
<tr>
<th>Path</th>
<th>Length</th>
<th># hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>A→B→C→D</td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>A→E→D</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>A→E→C→D</td>
<td>22</td>
<td>3</td>
</tr>
<tr>
<td>A→B→C→E→D</td>
<td>26</td>
<td>4</td>
</tr>
</tbody>
</table>
Exhaustive Search

- New link weights
- Find Shortest path from A to D
- List all possible paths and their lengths

<table>
<thead>
<tr>
<th>Path</th>
<th>Length</th>
<th># hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>A→B→C→D</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>A→E→D</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>A→E→C→D</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>A→B→C→E→D</td>
<td>13</td>
<td>4</td>
</tr>
</tbody>
</table>

Routing Algorithms

- Exhaustive Search does not scale with the size of the network
- Optimum shortest path algorithms exist to efficiently find the shortest path
- Routing Algorithms
  - Centralized
  - Distributed
- Examples:
  - Bellman-Ford Algorithm (one source/destination pair at a time)
  - Dijkstra’s Algorithm (source to all destinations)
Routing Algorithms

- Show example of Dijkstra’s Algorithm
  http://demonstrations.wolfram.com/ShortestPathsAndTheMinimumSpanningTreeOnAGraphWithCartesianE/

Shortest Path Approaches

**Distance Vector Protocols**
- Neighbors exchange list of distances to destinations
- Best next-hop determined for each destination
- Bellman-Ford (distributed) shortest path algorithm

**Link State Protocols**
- Link state information flooded to all routers
- Routers have complete topology information
- Shortest path (& hence next hop) calculated
- Dijkstra (centralized) shortest path algorithm

From: Communication Networks: Fundamentals Concepts and Key Architectures
Authors: A. Leon-Garcia and I. Widjaja
More on IP Routing:
Routing Information Protocol (RIP)

- IGP (within one AS)
- Distance vector protocol
- Distance is hop count
- Bellman-Ford Algorithm
  (Shortest Path Algorithm)
- Routers exchange view of network topology

More on IP Routing:
Open Shortest Path First (OSPF)

- IGP (within one AS)
- Link State routing protocol
- Routers discover
  - Their neighbors
  - The state of incident links
- Communicate state by periodically flooding the Link
  State Advertisements (LSA) throughout the network
- All routers converge to same map of the network topology
- Shortest path algorithm then used for routing.
  Distance can be more than just hop count.
More on IP Routing:
Border Gateway Protocol (BGP)

- EGP (Between AS’s)
- Path vector protocol, BGP advertises a sequence of AS #’s to the destination
- Routing information includes complete list of networks (AS’s) between source and destination
- Path vector info used to prevent routing loops
- Allows ranking of routes based on polices
- Polices are arbitrary rules, e.g., based on business agreements
- BGP enforces policy through selection of different paths to a destination and by control of redistribution of routing information
- Currently, it is common to have these manually configured
- Business agreements are reflected in BGP policies

Source Routing

- Source host selects path that is to be followed by a packet: sequence of nodes in path inserted into header
- Intermediate switches read next-hop address and remove address
- Source host needs link state information or access to a route server
- Source routing allows the host to control the paths that its information traverses in the network
- Potentially the means for customers to select what service providers they use
Both IPv4 and IPv6 allow source routing

IPv6

- In Spring of 2011 all IPv4 addresses were assigned
- IPv6 $\rightarrow$ Longer addresses
  - 128 bits/address (16 bytes) $\Rightarrow$ $3.4\times10^{38}$ addresses
  - Valid IPv6 address: 1002:DB78:7DF0:D5E9:976C:74ED:0FA1:89C1 (in hexadecimal)
- Simplified header -
  - 64 bit aligned
  - Longer but fewer fields
  - All fields are of fixed size
  - Easier to process at high speeds.
- Better options support $\rightarrow$ encoded in optional extension headers
- Flow label to support differentiated services
IPv6 Header Format

- Version field same size, same location
- Traffic class to support differentiated services
- Flow: sequence of packets from particular source to particular destination for which source requires special handling

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>12</th>
<th>16</th>
<th>24</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td><strong>Traffic Class</strong></td>
<td><strong>Flow Label</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Payload Length</strong></td>
<td><strong>Next Header</strong></td>
<td><strong>Hop Limit</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IPv6 Addressing

- **Address Categories**
  - **Unicast**: single network interface
  - **Multicast**: group of network interfaces, typically at different locations. Packet sent to all.
  - **Anycast**: group of network interfaces. Packet sent to only one interface in group, e.g. nearest.

- **Hexadecimal notation**
  - Groups of 16 bits represented by 4 hex digits
  - Separated by colons
  - Shortened forms:
    - `4BF5:0000:0000:0000:BA5F:039A:000A:2176`
    - To `4BF5:0:0:0:BA5F:39A:A:2176`
    - To `4BF5::BA5F:39A:A:2176`
  - Mixed notation:
    - `::FFFF:128.155.12.198`

---

IPv6

- **No checksum** (assumes other layers take care of it)
  - Lowers router processing, no longer have to recompute header checksum at each hop since TTL decremented.
  - Relieves resource burden on very fast links

- **No fragmentation in the network** – source must perform PATH MTU discovery
  - Send ICMP with requested MTU to destination, if get MTU to big response, decrement and retry. When destination replies, you have it.
  - Lowers router overhead – pushes complexity to edge
IPv6

- No broadcasts, replaced by multicasts
- ARPs, and ICMP combined/replaced with similar ICMPv6 functions.
- Security – IPsec available for v4, but is required to be available with IPv6 stack.
- Better support for mobility, auto configuration
  - No need for Network Address Translation (NAT)
  - Hosts have multiple addresses, can dynamically reconfigure without impact → easier plug-and-play
  - Router Solicitation, Router Advertisement – replaces DHCP, also includes duplicate address support
  - Enables stateless autoconfiguration → IPv6 address using a prefix obtained from a local router using an anycast message, eliminating the need for DHCP servers

Address Types based on Prefixes

<table>
<thead>
<tr>
<th>Binary prefix</th>
<th>Types</th>
<th>Percentage of address space</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
<td>Reserved</td>
<td>0.39</td>
</tr>
<tr>
<td>0000 0001</td>
<td>Unassigned</td>
<td>0.39</td>
</tr>
<tr>
<td>0000 001</td>
<td>ISO network addresses</td>
<td>0.78</td>
</tr>
<tr>
<td>0000 010</td>
<td>IPX network addresses</td>
<td>0.78</td>
</tr>
<tr>
<td>0000 011</td>
<td>Unassigned</td>
<td>0.78</td>
</tr>
<tr>
<td>0000 1</td>
<td>Unassigned</td>
<td>3.12</td>
</tr>
<tr>
<td>0001</td>
<td>Unassigned</td>
<td>6.25</td>
</tr>
<tr>
<td>0101</td>
<td>Aggregateable global unicast addresses</td>
<td>12.5</td>
</tr>
<tr>
<td>0110</td>
<td>Provider-based unicast addresses</td>
<td>12.5</td>
</tr>
<tr>
<td>0111</td>
<td>Unassigned</td>
<td>12.5</td>
</tr>
<tr>
<td>100</td>
<td>Geographic-based unicast addresses</td>
<td>12.5</td>
</tr>
<tr>
<td>101</td>
<td>Unassigned</td>
<td>12.5</td>
</tr>
<tr>
<td>110</td>
<td>Unassigned</td>
<td>12.5</td>
</tr>
<tr>
<td>1110</td>
<td>Unassigned</td>
<td>0.25</td>
</tr>
<tr>
<td>1111 0</td>
<td>Unassigned</td>
<td>3.12</td>
</tr>
<tr>
<td>1111 10</td>
<td>Unassigned</td>
<td>1.56</td>
</tr>
<tr>
<td>1111 110</td>
<td>Unassigned</td>
<td>0.78</td>
</tr>
<tr>
<td>1111 1110</td>
<td>Unassigned</td>
<td>0.2</td>
</tr>
<tr>
<td>1111 1110 10</td>
<td>Link local use addresses</td>
<td>0.008</td>
</tr>
<tr>
<td>1111 1110 11</td>
<td>Site local use addresses</td>
<td>0.038</td>
</tr>
<tr>
<td>1111 1111</td>
<td>Multicast addresses</td>
<td>0.39</td>
</tr>
</tbody>
</table>
Aggregatable global unicast addresses

- Identified by the Format Prefix (FP) of 001
- Same as public IPv4 addresses.
- Globally routable and reachable on the IPv6 Internet.
- Aggregatable global unicast addresses are also known as global addresses.
- For more details see:

Special Purpose Addresses

<table>
<thead>
<tr>
<th>n bits</th>
<th>m bits</th>
<th>o bits</th>
<th>p bits</th>
<th>(125-m-n-o-p) bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>Registry ID</td>
<td>Provider ID</td>
<td>Subscriber ID</td>
<td>Subnet ID</td>
</tr>
</tbody>
</table>

- Provider-based Addresses: 010 prefix
  - Assigned by providers to their customers
  - Hierarchical structure promotes aggregation
    - Registry ID: ARIN, RIPE, APNIC
    - ISP
    - Subscriber ID: subnet ID & interface ID
- IPv6 enables different hierarchical address structures to promote flexibility
Transition Mechanisms

- Dual stacks
  - Network elements running IPv4 and IPv6 at the same time
  - With translation between protocols
  - Some routers already doing this.
- Tunneling

Migration from IPv4 to IPv6

(a) IPv6 network IPv6 header IPv4 network IPv4 header Tunnel head-end Tunnel tail-end

(b) Source IPv6 network Link IPv6 network Destination

Modified From: Communication Networks: Fundamentals Concepts and Key Architectures
Authors: A. Leon-Garcia and I. Widjaja
IPv6 Deployment Aggregated Status

From: http://www.vyncke.org/ipv6status/

Look for World IPv6 Day