15-441 Computer Networking

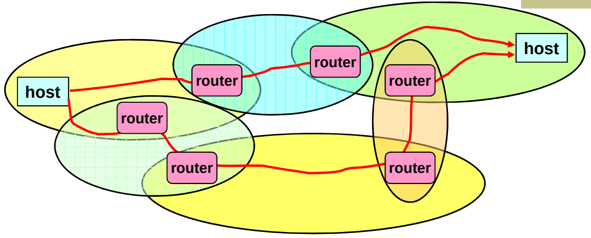
Lecture 7 –Internet design and IP Addressing

Peter Steenkiste

Fall 2014

www.cs.cmu.edu/~prs/15-441-F14

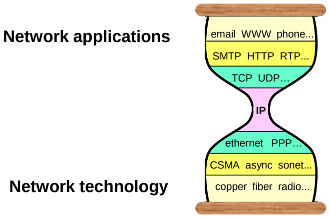
Logical Structure of an Internet



- Interconnection of separately managed networks using routers
  - Topology has emerged over time – not designed
  - Individual networks can use different (layer 1-2) technologies
  - The public Internet is a special (highly successful) example
- Send packets from source to destination by hopping through networks
  - “Network” layer responsiblity

Internet Protocol (IP)

- IP creates abstraction layer that hides underlying technology from network application software
  - Allows range of current & future technologies
- Can support many diverse types of applications
- IP is the shared language that is shared by all networks
  - Waist of the hourglass



Outline

- Discussion of the two papers
  - “The design philosophy of the DARPA Internet Protocols”, Dave Clark, SIGCOMM 88
  - “End-to-end arguments in system design”, Saltzer, Reed, and Clark, ACM Transactions on Computer Systems, November 1984
- Traditional IP addressing
- CIDR

Goals of the Internet [Clark88]

0 **Connect existing networks**

initially ARPANET and ARPA packet radio network

1. Survivability

ensure communication service even in the presence of network and router failures

2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Allow host attachment with a low level of effort
6. Be cost effective
7. Allow resource accountability

Goal 0: Connecting Networks

- How to internetwork various network technologies
  - ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links…
- Many differences between networks
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/pattern/handling
  - Routing

IP Standardization

- Alternative: translation “gateways”
- Minimum set of assumptions that underlying networks must meet to be part of the Internet
  - Minimum packet size
  - Reasonable delivery odds, but not 100%
  - Some form of addressing unless point to point
- Important non-assumptions:
  - Perfect reliability
  - Broadcast, multicast
  - Priority handling of traffic
  - Internal knowledge of delays, speeds, failures, etc
- Also achieves Goal 3: Supporting Varieties of Networks

Goal 1: Survivability

- If network is disrupted and reconfigured…
  - Communicating entities should not care!
  - No higher-level state reconfiguration
- How to achieve such reliability?
- Key question: where to keep communication state?

	Store in Network	Store on Host
Failure handling	Replicate the state	Natural “Fate sharing”
Switches …	Must maintain state	Are stateless
Net Engineering	Tough	Simple
Trust in host	Less	More

Principle: Fate Sharing

- Lose state information relevant to an entity's connections if and only if the entity itself is lost.
- Examples:
  - OK to lose TCP state if one endpoint crashes
    - NOT okay to lose it if an intermediate router reboots
  - Is this still true in today's network?
    - NATs and firewalls
- Tradeoffs
  - Survivability: Heterogeneous network → less information available to end hosts and Internet level recovery mechanisms
  - Trust: must trust endpoints more

Principle: Soft-state

- How can I not have state in the network, e.g., forwarding tables
- Solution: Soft-state
  - Announce state
  - Refresh state
  - Timeout state
- Penalty for timeout – poor performance
- Robust way to identify communication flows
  - Possible mechanism to provide non-best effort service
- Helps survivability

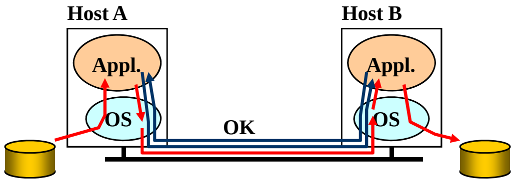
Principle: End-to-End Argument (Saltzer'81)

- Focus of the paper is "system"
  - Not a pure networking paper
- Deals with where to place functionality
  - Inside the network (in switching elements)
  - At the edges
- Argument: Some functions can only be correctly implemented by the endpoints – do not try to implement these elsewhere
  - Not a law – more of a "best practices"

Other Examples Throughout Course

- What should be done at the end points, and what by the network?
  - Reliable/sequenced delivery?
  - Addressing/routing?
  - Security?
  - Multicast?
  - Real-time guarantees?
  - Routing?

Example: Reliable File Transfer



- Solution 1: make each step reliable, and then concatenate them
- Solution 2: end-to-end check and retry

The “Other” goals

2. Types of service: only best effort service
   - Other services are optional, e.g., QoS
   - Or at end-points: TCP versus UDP
4. Decentralization: Internet – see BGP lecture
   - Allow distributed management of resources
5. Attaching a host
   - Host must implement hard part ☹ → transport services
6. Cost effectiveness: minimalist approach to IP
   - Packet overhead less important by the year
7. Accountability: “accounting” for resources
   - Accounting for billing purposes versus
   - Security: huge problem in the Internet today!

Changes Over Time → New Principles?

- Developed in simpler times
  - Common goals, consistent vision
- With success came changes in Internet goals
  - ISPs must talk to provide connectivity but are fierce competitors
  - Privacy of users vs. government’s need to monitor
  - User’s desire to exchange files vs. copyright owners
  - Security breaches vs. universal connectivity

- Provide choice → allow all parties to make choices on interactions – “tussle”
  - Can be tuned at different times: runtime, contract, hardware, ..
  - Creates competition, adjust for different contexts, ..
  - Fear between providers helps shape the tussle

Summary: Minimalist Approach

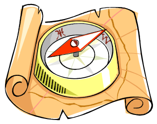
- Dumb network
  - IP provide minimal functionalities to support connectivity
    - Addressing, forwarding, routing
- Smart end system
  - Transport layer or application performs more sophisticated functionalities
    - Flow control, error control, congestion control
- Advantages
  - Accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - Support diverse applications (telnet, ftp, Web, X windows)
  - Decentralized network administration
- But the Internet has evolved since the early days …

Outline

- Discussion of the two papers
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
- CIDR

Getting to a Network Destination

- How do you get driving directions?
- Intersections→routers
- Roads→links/networks
- Roads change slowly



Directions

1. Start out going WEST on FORBES AVE toward S CRAIG ST.
2. Turn RIGHT onto S BELLEFIELD AVE.
3. Turn LEFT onto 5TH AVE.
4. Turn LEFT onto CRAFT AVE.
5. Turn RIGHT onto FORBES AVE.
6. Turn SLIGHT RIGHT onto BOULEVARD OF THE ALLIES/PA-885 N.
7. Take the I-376 W ramp toward DOWNTOWN/FORT PITT BRIDGE.
8. Merge onto US-22 W/US-30 W.
9. US-22 W/US-30 W becomes PA-60 N.

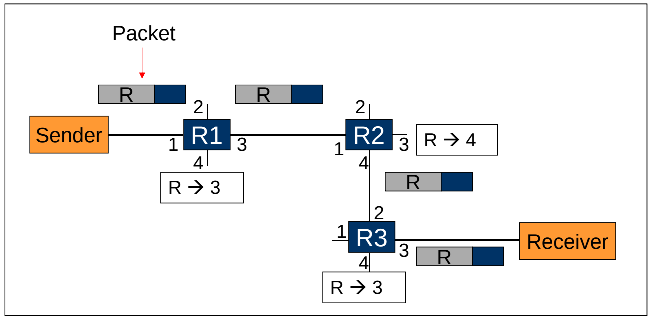
Forwarding Packets

- Table of virtual circuits
  - Inspired by telephone circuits
  - More on this later
- Source routing
  - Packet carries a path
- Table of global destination addresses (IP)
  - Routers keep next hop for destination
  - Packets carry destination address

Source Routing

- List entire path in packet
  - Driving directions (north 3 hops, east, etc..)
- Router processing
  - Strip first step from packet
  - Examine next step in directions and forward
- Rarely used
  - End points need to know a lot about network
  - Economic and security concerns
  - Variable header size

Global Address Example



Forwarding based on Global Addresses

- Advantages
  - Conceptually simple
  - Lines up with roles of actors (ISPs, endpoints)
  - Stateless – simple error recovery
- Disadvantages - challenges
  - Every switch knows about every destination
    - Potentially large tables - today
  - All packets to destination take same route
    - “Traffic engineering” – virtual circuits
  - Need routing protocol to fill table
    - Next couple of lectures

Addressing in IP

- IP addresses identify interfaces
  - E.g., 128.2.1.1
  - Multiple interfaces -> multiple IP addresses
- Domain Name System (DNS) names are names of hosts
  - E.g., www.cmu.edu
- DNS binds host names to interfaces
- Routing binds interface addresses to paths

Addressing Considerations

- Flat addresses – one address for every host
  - Peter Steenkiste: 123-45-6789
  - Does not scale – router table size explodes
  - 630M (1/09) entries,doubling every 2.5 years
  - Why does it work for Ethernet?

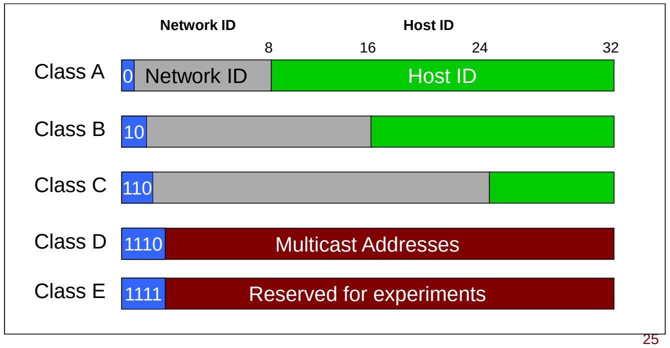
- Hierarchical – add structure
  - Pennsylvania / Pittsburgh / Oakland / CMU / Steenkiste
    or Peter Steenkiste: (412)268-0000
  - Common “trick” to simplify forwarding, reduce forwarding table

- What type of Hierarchy?
  - How many levels?
  - Same hierarchy depth for everyone?
  - Who controls the hierarchy?

IP Address Structure

Challenge: Accommodate networks of different very sizes

Initially: classful structure (1981) (not relevant now!!!)



Original IP Route Lookup

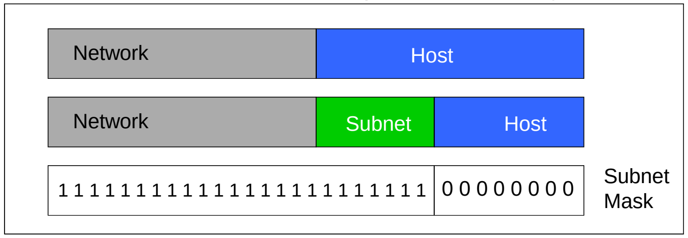
- Address specifies prefix for forwarding table
  - Extract address type and network ID
- Forwarding table contains
  - List of class+network entries
  - A few fixed prefix lengths (8/16/24)
  - Prefix – part of address that really matters for routing
- www.cmu.edu address 128.2.11.43
  - Class B address – class + network is 128.2
  - Lookup 128.2 in forwarding table for class B
- Tables are still large!
  - 2 Million class C networks

Subnet Addressing RFC917 (1984)

- Class A & B networks too big
  - Very few LANs have close to 64K hosts
  - For electrical/LAN limitations, performance or administrative reasons
- Need simple way to get multiple “networks”
  - Use bridging, multiple IP networks or split up single network address ranges (subnet)
- CMU case study in RFC
  - Chose not to adopt – concern that it would not be widely supported ☺

Subnetting

- Add another layer to hierarchy
- Variable length subnet masks
  - Could subnet a class B into several chunks
- Subnetting is done internally in the organization
  - It is not visible outside – important for management

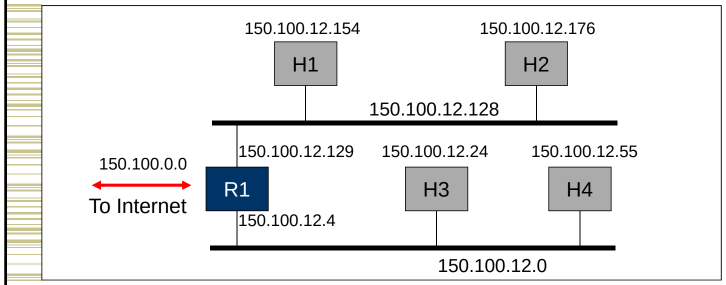


Subnetting Example

- Assume an organization was assigned address 150.100
- Assume < 100 hosts per subnet
- How many host bits do we need?
  - Seven
- What is the network mask?
  - 11111111 11111111 11111111 10000000
  - 255.255.255.128

Forwarding Example

- Assume a packet arrives with address 150.100.12.176
- Step 1: AND address with class + subnet mask
  - Subnet masks stored on router



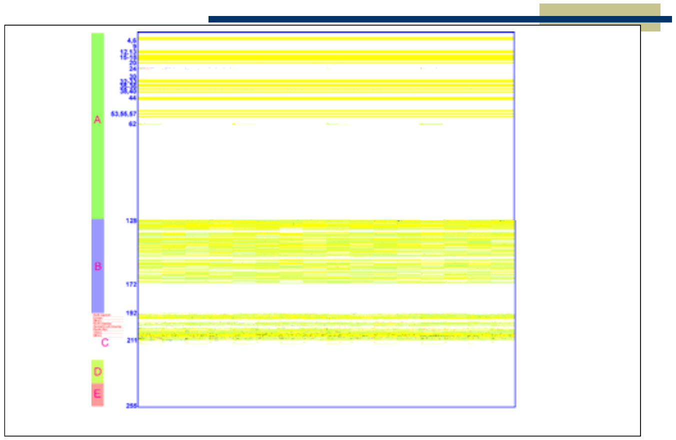
Outline

- Discussion of the two papers
- Traditional IP addressing
- CIDR
  - Motivation
  - Classless address
  - Example

IP Address Problem (1991)

- Address space depletion
  - Suppose you need 2^{16} + 1 addresses?
  - In danger of running out of classes A and B
    - Class C too small for most domains
    - Very few class A – very careful about using them
    - Class B – greatest problem
- Class B networks sparsely populated
  - But people refuse to give it back
- Large forwarding tables
  - 2 Million possible class C groups

IP Address Utilization ('97)

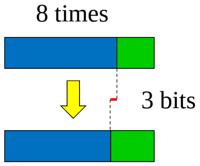


Classless Inter-Domain Routing (CIDR) – RFC1338

- Arbitrary split between network & host part of address → more efficient use of address space
  - Do not use classes to determine network ID
  - Use "prefix" that is propagated by routing protocol
  - E.g., addresses 192.4.16 - 192.4.31 have the first 20 bits in common. Thus, we use these 20 bits as the prefix (network number) → 192.4.16/20
- Merge forwarding entries → smaller tables
  - Use single entry for range in forwarding tables even if they belong to different destination networks
    - "Adjacent" in address space and same egress

CIDR Example

- Network is allocated 8 class C chunks, 200.10.0.0 to 200.10.7.255
  - Move 3 bits of class C address to host address
  - Network address is 21 bits: 201.10.0.0/21
- Replaces 8 class C routing entries with 1 entry
- But how do routers know size of network address?
  - Routing protocols must carry prefix length with address



IP Addresses: How to Get One?

Network (network portion):

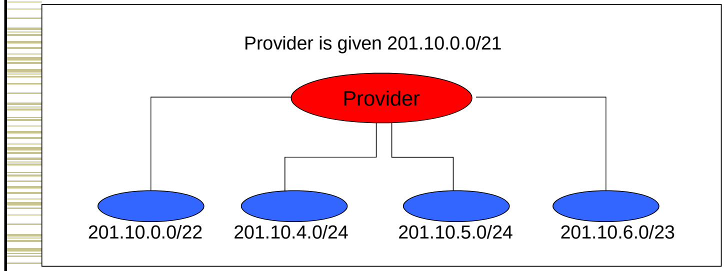
- Get allocated portion of ISP's address space:

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

IP Addresses: How to Get One?

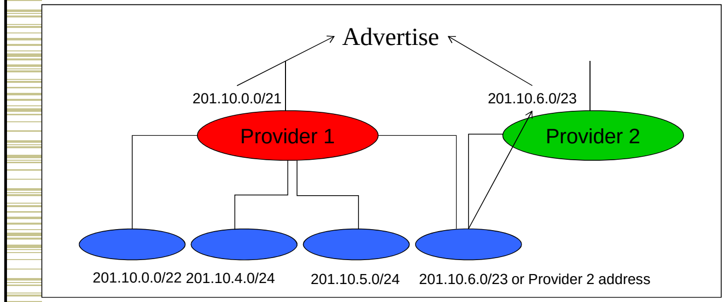
- How does an ISP get block of addresses?
  - From **Regional Internet Registries** (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)
- How about a single host?
  - Assigned by sys admin (static or dynamic)
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: “plug-and-play”
    - Host broadcasts “DHCP discover” msg
    - DHCP server responds with “DHCP offer” msg
    - Host requests IP address: “DHCP request” msg
    - DHCP server sends address: “DHCP ack” msg

CIDR Illustration



CIDR Implication: Longest Prefix Match

- How to deal with multi-homing, legacy addresses, …



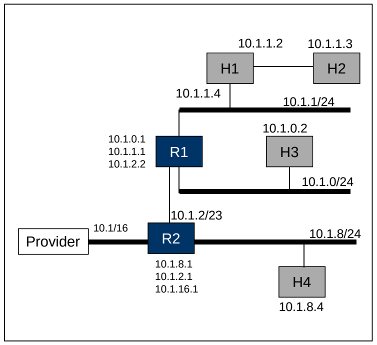
Host Routing Table Example

```
Destination     Gateway         Genmask         Iface
128.2.209.100   0.0.0.0         255.255.255.255 eth0
128.2.0.0       0.0.0.0         255.255.0.0     eth0
127.0.0.0       0.0.0.0         255.0.0.0       lo
0.0.0.0         128.2.254.36    0.0.0.0         eth0
```

- From “netstat –rn”
- Host 128.2.209.100 when plugged into CS ethernet
- Dest 128.2.209.100 → routing to same machine
- Dest 128.2.0.0 → other hosts on same ethernet
- Dest 127.0.0.0 → special loopback address
- Dest 0.0.0.0 → default route to rest of Internet
  - Main CS router: gigrouter.net.cs.cmu.edu (128.2.254.36)

Routing to the Network

- Packet to 10.1.1.3 arrives
- Path is R2 – R1 – H1 – H2
- H1 serves as a router for the 10.1.1.2/31 network

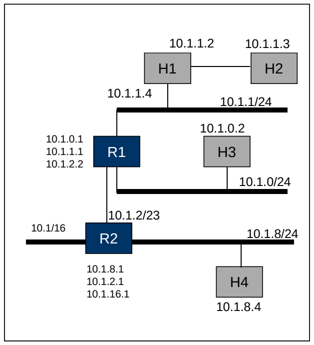


Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.0.0/23

Routing table at R2

Destination	Next Hop	Interface
127.0.0.1	-	lo0
Default or 0/0	provider	10.1.16.1
10.1.8.0/24	-	10.1.8.1
10.1.2.0/23	-	10.1.2.1
10.1.0.0/23	10.1.2.2	10.1.2.1

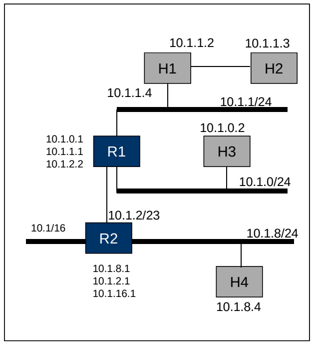


Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.1.2/31
  - Longest prefix match

Routing table at R1

Destination	Next Hop	Interface
127.0.0.1	-	lo0
Default or 0/0	10.1.2.1	10.1.2.2
10.1.2.0/23	10.1.2.1	10.1.2.2
10.1.0.0/24	-	10.1.0.1
10.1.1.0/24	-	10.1.1.1
10.1.1.2/31	10.1.1.4	10.1.1.1

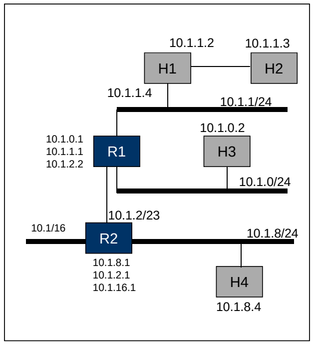


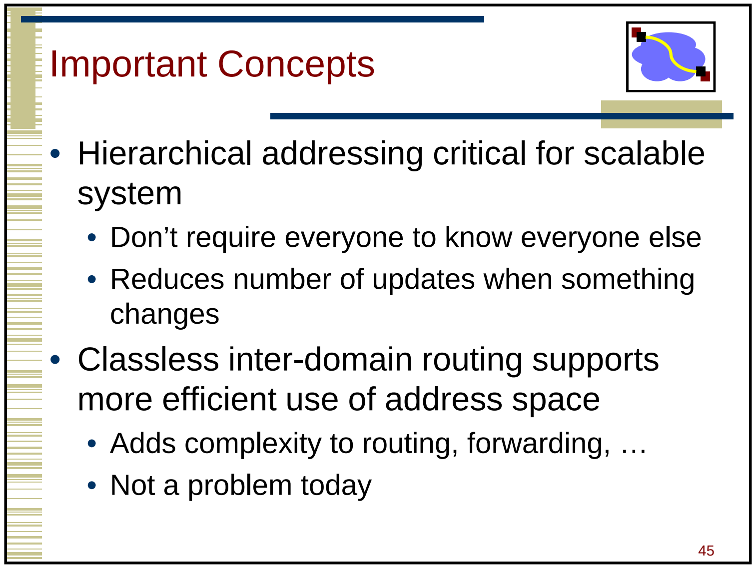
Routing Within the Subnet

- Packet to 10.1.1.3
- Direct route
  - Longest prefix match

Routing table at H1

Destination	Next Hop	Interface
127.0.0.1	-	lo0
Default or 0/0	10.1.1.1	10.1.1.4
10.1.1.0/24	-	10.1.1.1
10.1.1.2/31	-	10.1.1.2





Important Concepts

- Hierarchical addressing critical for scalable system
  - Don’t require everyone to know everyone else
  - Reduces number of updates when something changes
- Classless inter-domain routing supports more efficient use of address space
  - Adds complexity to routing, forwarding, …
  - Not a problem today