Global Internet

Computer Networks
Dr. Jorge A. Cobb

UTD

Where Are We?

- Internet
  - Connect heterogeneous collection of networks
  - Simple addressing hierarchy
- Scalability Challenges
  - Several challenges exist to make the Internet of global scale
  - We address several of these in these notes.

Global Internet – Summary of Topics

- IP address hierarchy evolution
  - Subnetting
  - CIDR
- Evolution of Internet structure
- Virtual geographies
  - Networks
  - Domains (Autonomous Systems)
- Routing with domains
  - Intradomain routing
  - Interdomain routing

Problems of Scale

1. Inefficient address allocation
   - Most physical networks (i.e. LANs) have only about 100 hosts or less
   - Assigning class A or B to them would be wasteful.
   - Solution: subnetting

2. Too many networks for routing
   - Networks at the core need to be "aware" of each class A, B, and C network number
   - Too many networks! Routing tables don't scale
   - Solution: CIDR

- We will tackle each of these in turn (i.e. we start with 1 above)

Subnetting

- Assume an organization is given a class B network #
  - Class B has two bytes for network # and two bytes for the host #
  - E.g., 128.174.0.0 (recall, network #'s have 0's for the host bits)
- Think of this as an "**address block**" of 2^16 addresses
- First address in the block is 128.174.0.0

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- Last address in the block is 128.174.255.255

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- Actually, 2^16 – 2 addresses (128.174.0.0 and 128.174.255.255 cannot be used ☺)
- I sometimes draw this block as follows

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Subnetting continued …

- Assume the organization has multiple physical networks with a few hundred hosts in each physical network.

- The class B network # can be broken into smaller “sub”networks

- Idea: take a single IP network number
  - Break its block of addresses into smaller blocks (subnets).
  - Allocate a smaller block of IP addresses to each physical network.
  - **Not all blocks are of the same size!**

Example

- Consider again network 128.174.0.0

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

- One sub-block of addresses could be as follows:

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	X	X	X	X	X	X	X

- First 25 bits identify the network and the subnetwork
- The last 7 bits identify the host within the subnetwork
- This subnetwork is identified as 128.174.**142**.**128** (zero for the host bits).
- It also has associated with it a subnet “mask”

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

1 – network or subnet bit, 0 – host bit

Another subnet

- Consider again network 128.174.0.0

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

- Another sub-block of addresses could be as follows:

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	0	1	X	X	X	X	X	X	X	X

- First 24 bits identify the network and the subnetwork
- The last 8 bits identify the host within the subnetwork
- This subnetwork is identified as 128.174.**141**.**0** (zero for the host bits).
- It also has associated with it a subnet “mask”

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

1 – network or subnet bit, 0 – host bit

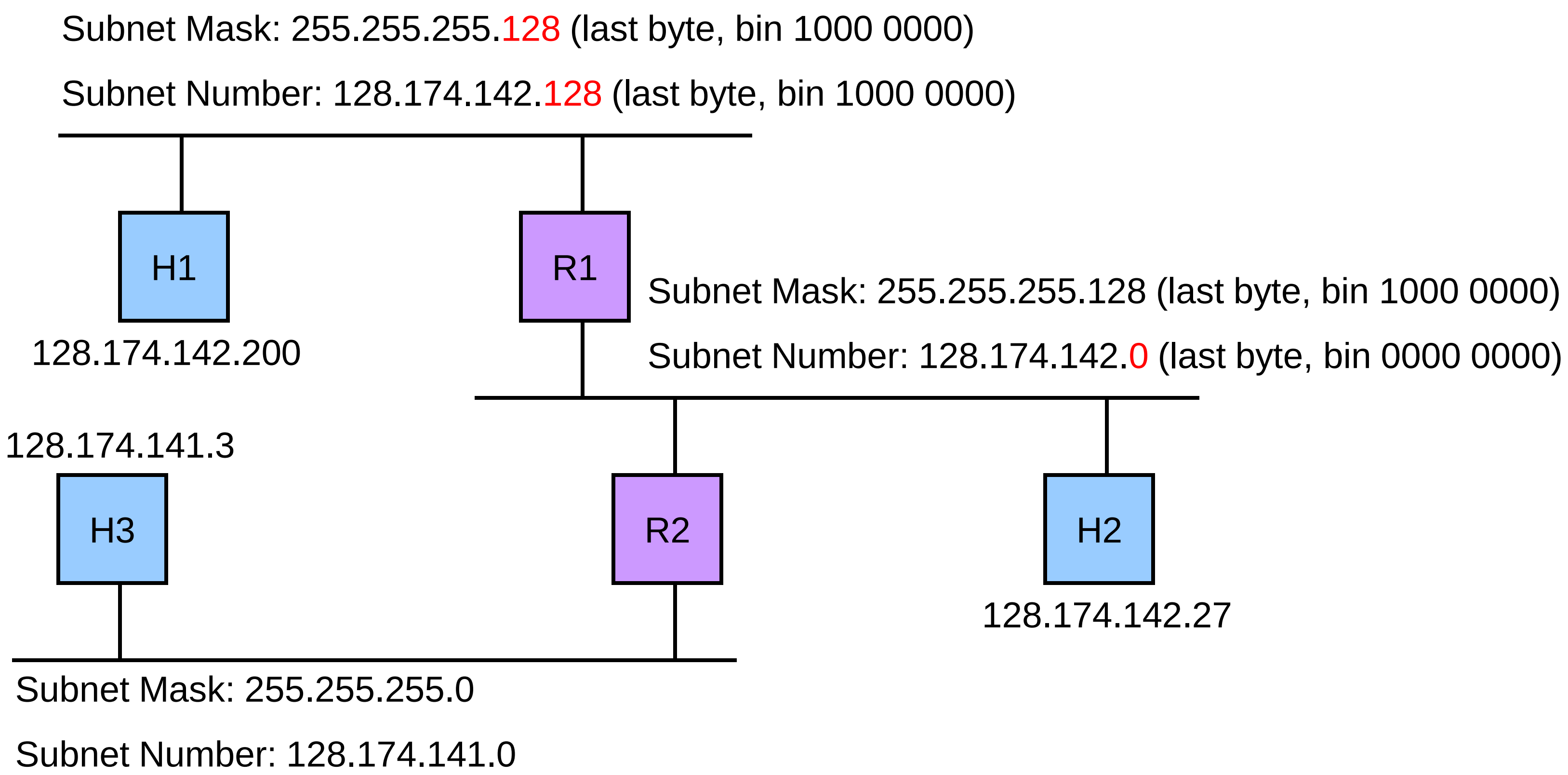
Why a subnet mask?

- Assume I tell you about a subnet number:
  - 128.174.141.0
- How many bits are for the network/subnet and how many bits are for the host? Can you tell?
- Routing table entries within the organization include the subnet number AND its associated subnet mask

Subnetting – in summary

- Assumptions
  - Subnets are close together (same company)
    - Looks like a single network to routers outside the organization
    - Hence, outside routers only have a single entry in their table for the organization.
- IP with Subnetting:
  - All hosts in the same company have the same network#
  - All hosts on the same physical network must have the same subnet # (an “extension” of network #)

Subnetting Example



Note: subnet masks are not of the same length
Can we replace the subnet mask of 128.174.142.0 by 255.255.255.0?
Can we have an additional subnet 128.174.142.0 with mask 255.255.255.0?
Can we have an additional subnet 128.174.140.0? How many bits can we have in the mask?

Subnetting (host IP XOR SM = SN #)

Host 3: 128.174.141.3

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	1	1

Subnet Mask 255.255.255.0

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Subnet # 128.174.141.0

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0

Host 1: 128.174.142.200

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	1	0	0	1	0	0	0

Subnet Mask 255.255.255.128

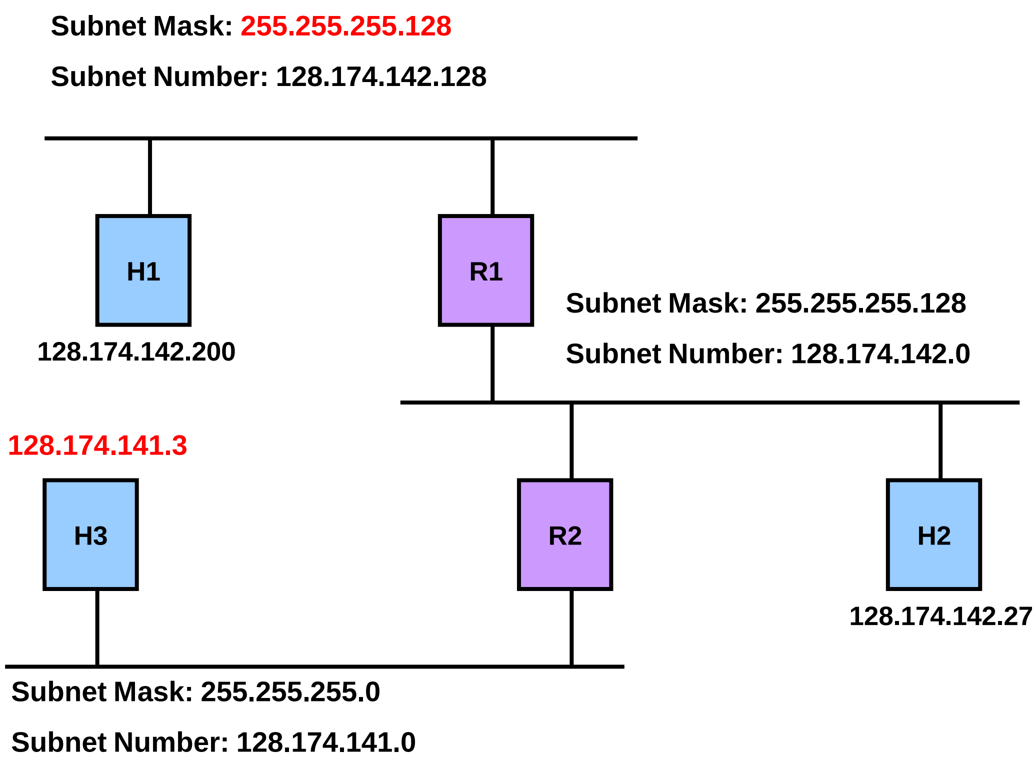
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0

Subnet # 128.174.142.128

1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0

Subnetting

Send an IP packet from H1 to H3: send directly or via a router?



- At H1:
- Compute (H3 "subnet number")
  - 128.174.141.3 AND 255.255.255.128 = 128.174.141.0 (≠ 128.174.142.128 = H1's subnet #)
- If result = H1's subnet number
  - then H3 and H1 are on the same subnet
- Else
  - route through appropriate router

Subnetting

Subnet #	Subnet Mask	Next Hop
128.174.141.0	255.255.255.0	Interface 0
128.174.142.0	255.255.255.128	Interface 1
128.174.142.128	255.255.255.128	R1
128.174.0.0	255.255.0.0	R3
0 (Default)	0.0.0.0	R3

- Example Table from R2
  - Next hop (AND against subnet mask, compare to subnet #)
    - **128.174.142.196 to R1 (why not R3?**)
    - 128.174.142.95 to Interface 1
    - 128.174.141.137 to Interface 0
    - 129.174.145.18 to R3
    - 131.126.244.15 to R3

Subnetting

- Notes
  - Non-contiguous subnets are difficult to administer
  - Multiple subnets on one physical network
    - Must be routed through router
- Pros
  - Helps address consumption
    - Better use of class A and B addresses
  - Helps reduce routing table size of routers outside the organization
    - one network # per company rather than multiple network numbers per company (one per physical network)

Problems of Scale: need for CIDR

- Most companies plan to have more than 255 machines (Class C)
  - Thus they choose a class B
  - Otherwise, renumbering is time consuming and can interrupt service.

- Class B networks aren't very efficient
  - Approximately 16,000 class B networks available
  - Few organizations have O(10,000) machines
    - More likely use O(1,000) of the 65,000 addresses

- What about multiple class C networks per company?
  - Routing tables in the core don't scale in this case (too many networks)
  - Protocols do not scale beyond O(10,000) networks

Solution – Classless Interdomain Routing (CIDR)

- <u>Eliminate class notation</u> (A, B, and C are gone!)
- <u>Generalize</u> subnet notion and subnet masks (just mask).
- Allow only contiguous masks
- Specify network by (network no. **/** no. of bits in mask)
- The mask is <u>dynamic</u>
  - Think of a network as a "region" where all machines have the same prefix (network number) in their IP address.
  - As you get closer to the destination, the mask size (prefix length) "increases", I.e. more details are available.
- Aggregate routes in routing tables – contiguous blocks of network numbers along the same path are aggregated into one network with a shorter mask (prefix length)

Starting from the bottom ☺

- Organization X was given a “class C” address block **192.4.16.0/24**
- But a physical network in the org. has only 6 hosts
- Thus, for this phys. netw., X uses 3 bits for the host (29 for netw #)

192.4.16.0/29

- Remaining addresses may be given to other physical networks

192.4.16.0/24 Class C address block given to organization X

1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	x	x	x	x	x	x	x	x

24 bits identify the class C network

192.4.16.0/29 Network number of small physical network

1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	x	x	x

29 bits identify the physical network

Further Aggregation

- Assume organization X is given a block of 16 contiguous C netw numbers.
  - E.g. 192.4.16.0/24 – 192.4.31.0/24 (see picture below)
- Organization X is identified by **192.4.16.0/20**, where network number 192.4.16.0, and prefix length (mask) is 20 bits.
- Routers outside the organization have one entry, rather than 16 entries
- Block size (# of addresses) must be a power of 2
- Network number may be any number of bits

192.4.16.0/24 First class C network #

1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	X	X	X	X	X	X	X	X

192.4.31.0/24 Last class C network #

1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	X	X	X	X	X	X	X	X

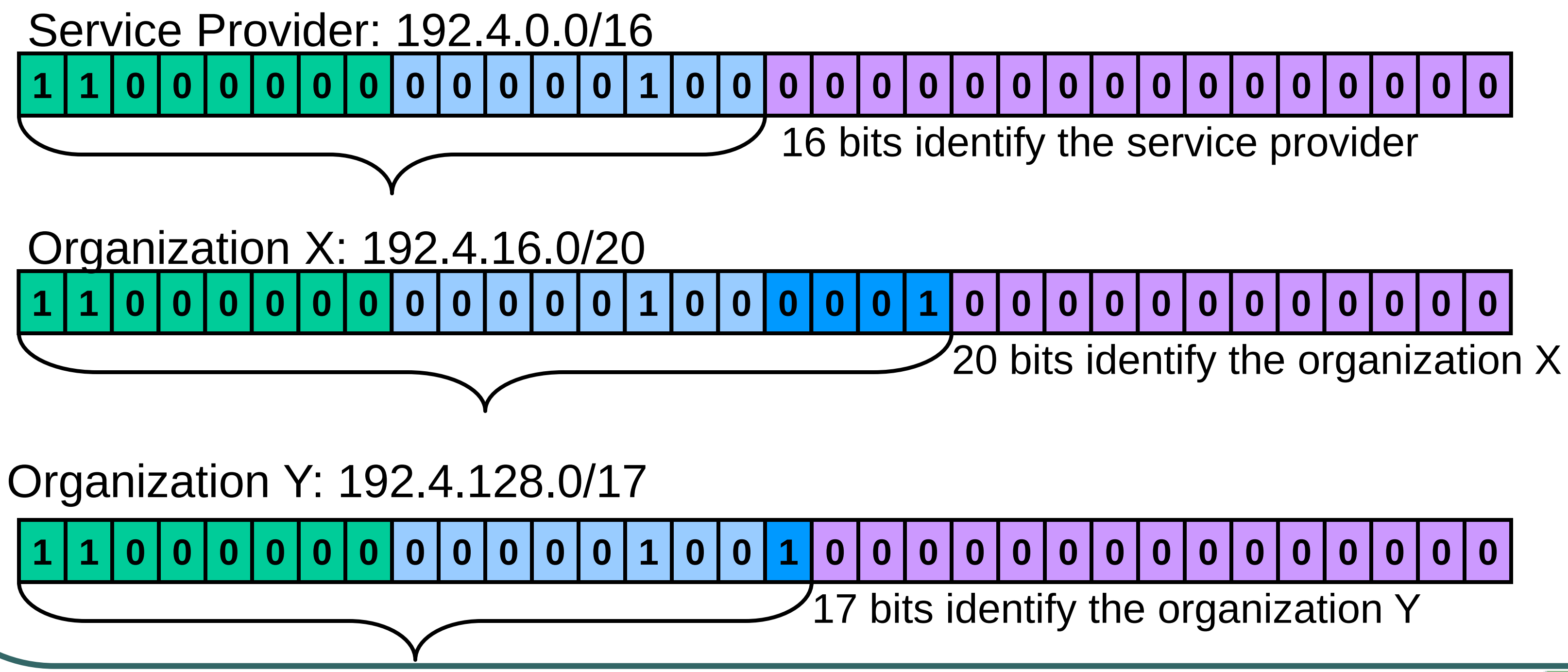
192.4.16.0/20 X's block of addresses

1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X

20 bits identify the organization X

Even More Aggregation

- A service provider could be assigned the network address 192.4.0.0/16
- It further breaks this address into <u>contiguous blocks of addresses</u> and gives them to its client organizations.
- One of them is 192.4.16.0/20 for organization X as before.
- Routers outside the service provider have only a single entry for the entire service provider, I.e., 192.4.0.0/16



Routing with CIDR

- Core routers will only care about the first 16 bits of 192.4.0.0/16 to reach the service provider
- Service provider routers care about the first 20 bits in 192.4.16.0/20 to reach organization X
- X cares about the first 29 bits in 192.4.16.0/29 to reach the specific physical network (Ethernet)

Longest Match Prefix (weird)

- Assume a service provider P has a contiguous group of addresses.
- This group was split into different organizations, X, Y, Z, etc.
- What if X changes to service provider Q? Its IP addresses would change (bad, lots of renumbering)
- X is allowed to keep its IP addresses.
- P advertises to outside routers that it can reach the contiguous group of addresses (including X!)
- Q advertises the more specific group of addresses of X.
- Routers must follow path to Q, since Q has more "specific" information (longest prefix).

CIDR

Subnet # / length	Next Hop
128.174.141.0 / 24	Interface 0
128.174.142.192 / 27	Interface 1
128.174.142.128 / 25	R1
128.174.0.0 / 16	R3
Default	R3

- Trend is for increasing amounts of overlap in routing table entries
- Example: 128.174.142.200
  - Matches second, third and fourth lines
  - Route to entry with longest match (always!)

Need for More Scalability

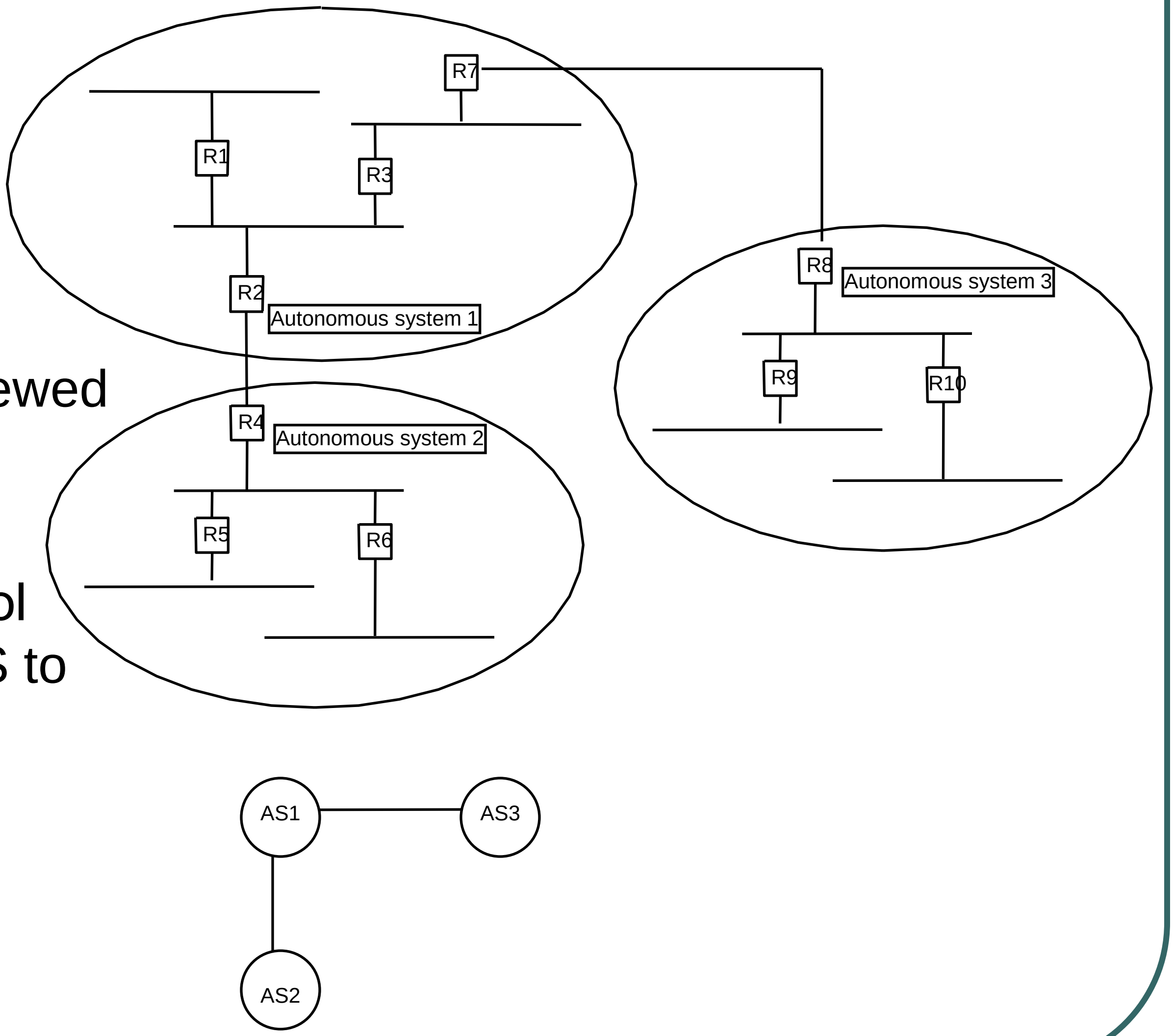
- Even with CIDR, it is not scalable enough
- For a routing protocol to work, all routers in a network are aware of all other routers in the network
- My router here in UTD should not need to know about or talk to routers in Hong Kong.
- We need to break the Internet into pieces, or “routing domains”

Autonomous Systems (Routing Domains)

- The Internet is subdivided into Autonomous Systems
- There are currently about 13000 active ASM’s
- Based on notion of autonomy of control
- E.g.: company, university, etc
- Each AS has a unique 16 bit ID

AS Picture and meta picture

- Enables hierarchical aggregation of routing information
- An entire AS may be viewed as a single "node"
- A "meta" routing protocol finds paths from one AS to another



Autonomous Systems

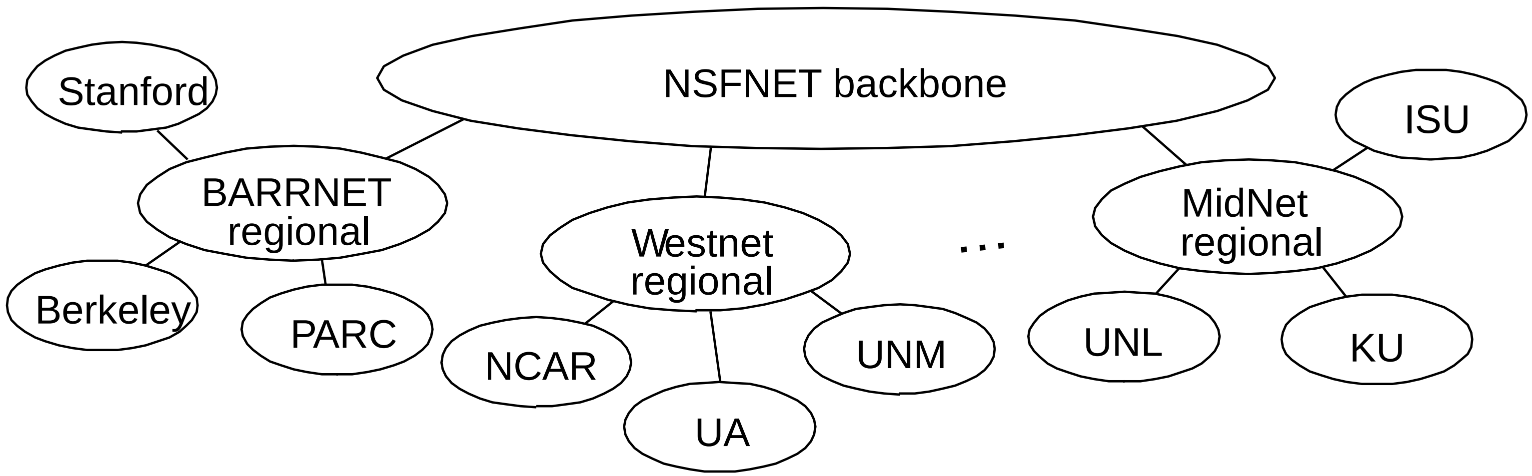
- Intradomain Routing (within an AS)
  - Performed using domain-specific algorithm (e.g. OSPF, RIP)
  - Selected by domain administrators
  - Allows heterogeneous interior gateway protocols

- Interdomain Routing (between ASM's)
  - Performed using standard global algorithm
  - Nodes in the routing table are AS's
  - Homogeneous exterior gateway protocol (why?)
  - Main goal: reachability

Standard Interdomain Routing Protocols

- General aspects
  - Very complex and difficult
  - Large scale (140,000 network prefixes in the core of the network, and about 14,000 AS numbers)
  - Focuses on reachability rather than optimality
  - Must be loop-free
  - Specify how reachability information should be exchanged

Old Tree-Structure

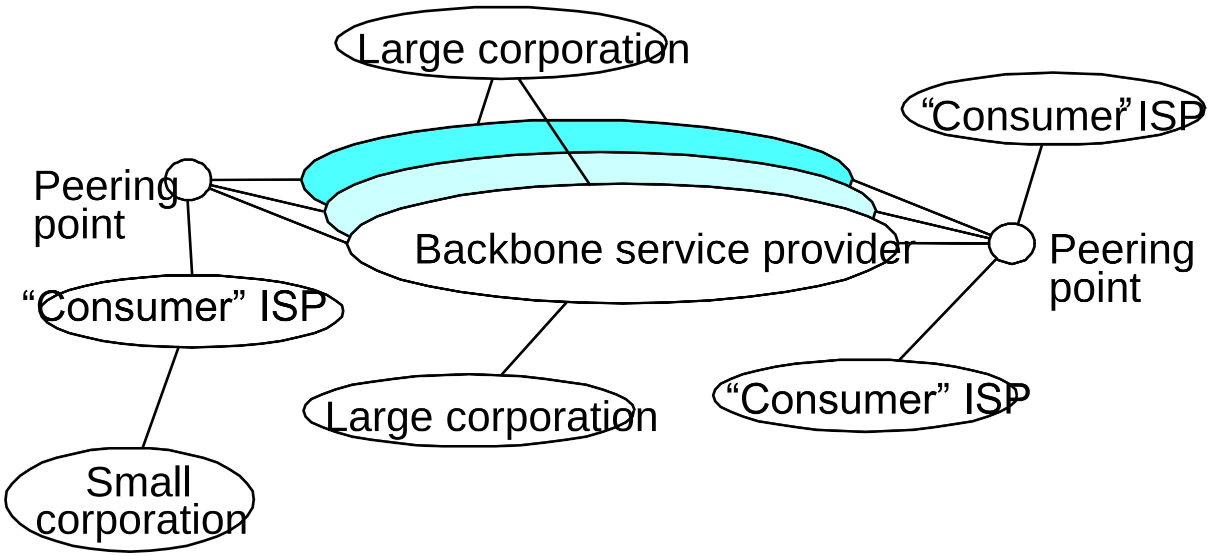


- Internet before 1990 had a tree-structure
- Main “core” was the NSF Backbone (NSFNET)
- An AS could have a parent and/or children
- No “peering”, strictly hierarchical.
- A low level AS could have a default route to its parent (and thus need not know the whole Internet).
- However, the NSFNET needed to know the whole Internet

EGP (OLD)

- Defined on the Internet having a tree structure
- Embodied (and enforced) tree structure
- Each AS must learn how to reach every AS in its sub-tree
- Thus, the core network learns the path to every AS
- Distance vector updates
- Had to be replaced eventually

Privatization of the Internet



- Mid 1990's, NSF relinquished control of the Internet backbone
- We have many commercial backbone providers (UUNet/Worldcomm, Sprint, MCI, ...)
- Mesh connectivity, multi-homed networks,
- Loops galore
- Need a new solution

Types of AS

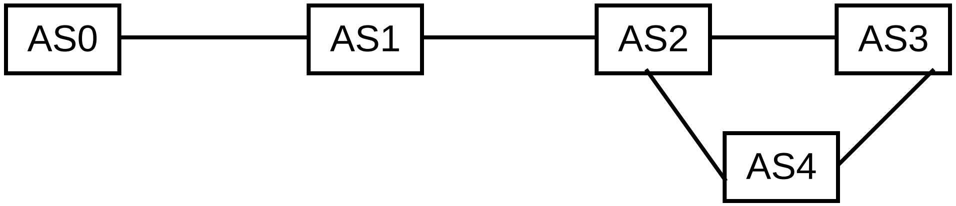
- Stub: only connected to one other AS; carries local traffic only (can use default path to its parent AS, and does not necessarily have an AS number)
- Multi-homed: connected to multiple ASM's, but refuses to carry transit traffic
- Transit: allows traffic from other ASM's to cross it.

- Need a protocol that can handle this general connectivity

Reachability vs Optimality

- Each domain can choose its own interior routing protocol (intradomain)
- It can choose any scheme to assign metrics (i.e. costs) to its interior paths
- No consistency between ASM's
  - Value of 1000 in one AS may be great, but awful at another
- Impossible to find the least cost path to a destination AS.
- Best you can do is find "a path".
- Each AS advertises "reachability information"
  - I can reach AS n and it contains networks 129.18.0.0/16 and 100.18.0.0/16
  - No cost is given.

Loop Freedom (must avoid loops)



- Short-lived or long-lived routing loops are devastating
  - The traffic of an entire AS could be interrupted
  - Looping packets would cause congestion
- Example
  - AS2 announces to AS3 it can reach AS0
  - AS3 announces to AS4 it can reach AS0
  - AS4 announces to AS2 it can reach AS0
  - AS2 chooses AS4 as its next hop to AS0

Need for Flexible Routing Policies

- We could choose the path with least # of AS-hops
- However, other factors may be more important:
  - An AS may prefer a neighboring AS according to:
    - Some economic relationship
    - Level of trust (may not trust some ASM’s)
  - Also, each AS is not forced to disclose its reachability
    - <u>Multi-homed ASM’s prevent through traffic by not announcing reachability</u>

- Thus, domains should have the freedom to choose their path to each destination domain (flexible routing policy)
- The whole point is to find a “good” path not an optimal one.

Review: intradomain routing protocols

- Common intradomain routing protocols
  - Routing Information Protocol (RIP)
    - From the early Internet
    - Part of Berkeley Software Distribution (BSD) Unix
    - <u>Distance vector algorithm</u>
    - Based on hop count (infinity set to 16 hops)
  - Open Shortest Path First (OSPF)
    - Internet Standard (RFC 2328)
    - <u>Link state algorithm</u>
    - Authenticates messages
    - Load balances across links

Why not DV or Link-State for interdomain routing?

- Link-State
  - Too much overhead $O(N^2)$ message and processing overhead.
  - Even for intradomain routing it has too much overhead
- Distance Vector
  - Slow to converge (counting to infinity)
    - We may choose all links to have a cost of "1", but short-lived loops still would exist
- Need for flexible policies
  - Neither of the above two supports flexible policies, because the entire path information is needed.
  - Both of the above look for <u>min-cost paths</u>, not flexible enough to implement routing policies.

BGP-4

- Current standard interdomain routing protocol
- Assumption

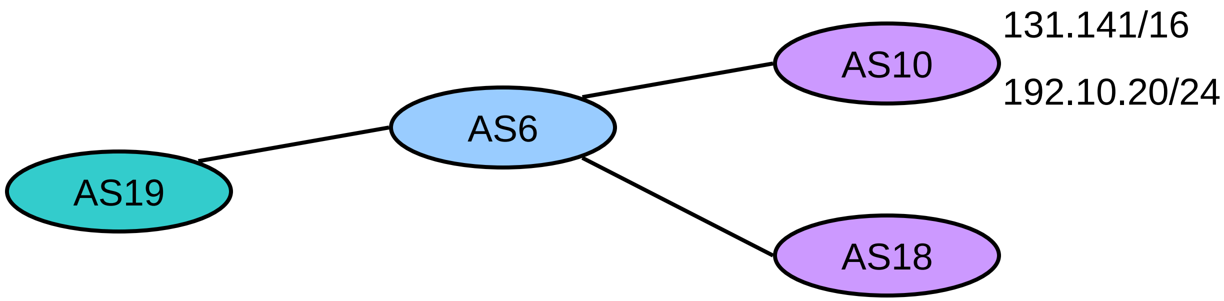
  Internet is an arbitrarily interconnected set of AS's
- Traffic
  - Local: Begins or ends within an AS
  - Transit: Moves through an AS
- Each AS has
  - Border routers (one or more)
    - Connects an AS to the Internet
    - Used for default external route
  - BGP speakers (one or more)
    - Routers that participate in the interdomain routing protocol

BGP-4

- Neither link-state nor distance-vector
- BGP speakers advertise, for each network N,
  - The full path (list) of ASM's to reach network N
- Loops are avoided by not choosing a neighbor's path if it contains your own AS ID.
- Only one path is advertised even if many are available.
  - Advertise the route that you have chosen according to your routing policies.

BGP-4 path advertisements

- Example
  - AS10 advertises 131.141/16 and 192.10.20/24 as local networks
  - AS6 advertises same networks with path (AS6, AS10)
  - AS19 advertises same networks with path (AS19, AS6, AS10)

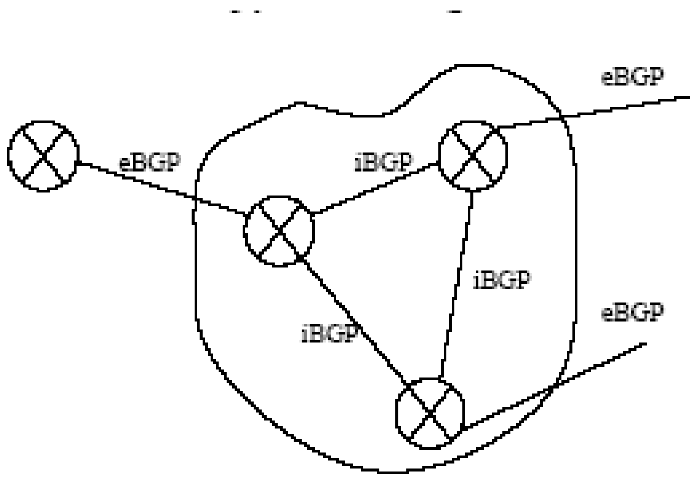


TCP Connectivity

- Neighboring (peer) BGP speakers connect to each other using TCP (reliable)
- Information is not “refreshed”
- Only “keep-alive” messages are sent periodically to each neighbor.
- Advertised paths, if no longer existent, must be withdrawn explicitly.
  - Thus, a router has two explicitly tell its neighbors a path is no longer available.

Multiple BGP Speakers per AS

- Two types of neighbors
  - Internal: in the same AS, use iBGP protocol
  - External: in a different AS, use eBGP protocol
- Full TCP connectivity with internal neighbors.
  - All internal neighbors share the routes they learned from outside ASM's.
  - <u>Internal neighbors may be multiple hops away!</u>



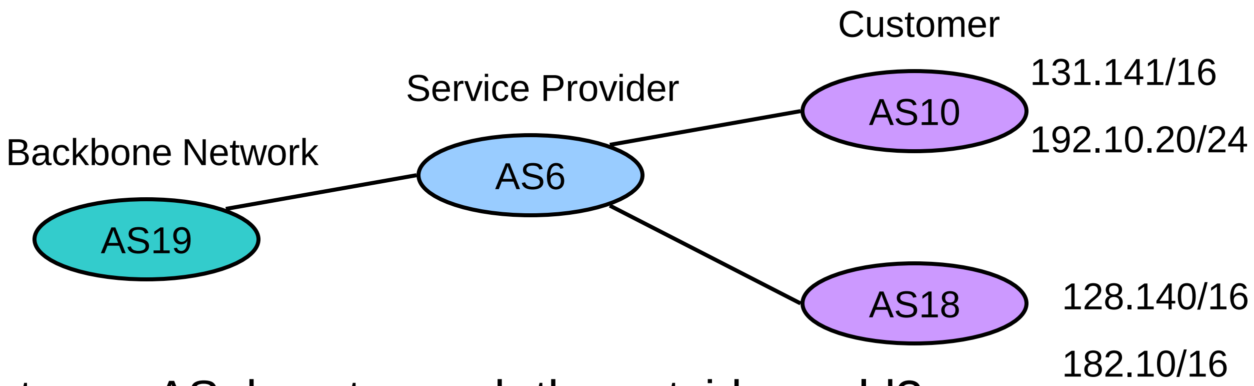
Why is everything more scalable?

- Two levels of the hierarchy
  - Outside of an AS (interdomain)
  - Within an AS (intradomain)

- Interdomain: Finding the next-hop AS and its border router is scalable: there are less ASms than networks.

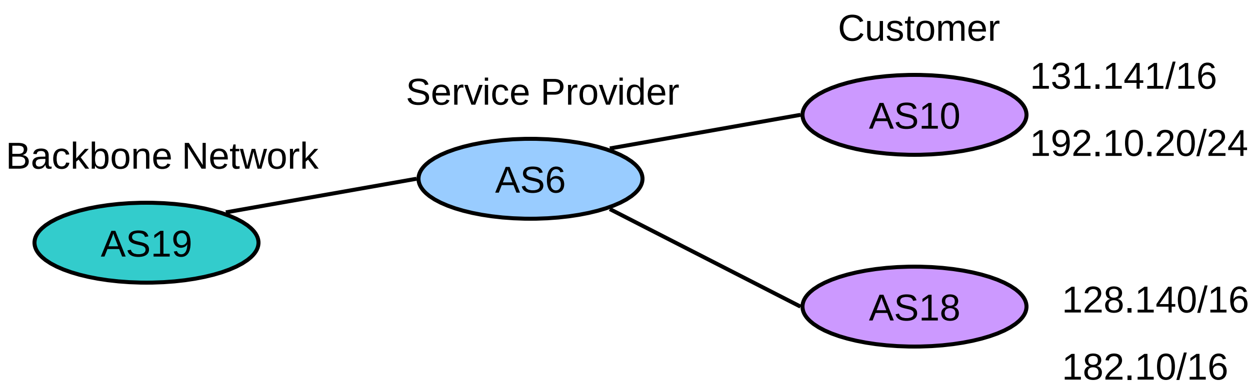
- Intradomain: Within an AS you must find a path to the border router (done by, e.g., OSPF protocol)

Integrating Interdomain and Intradomain



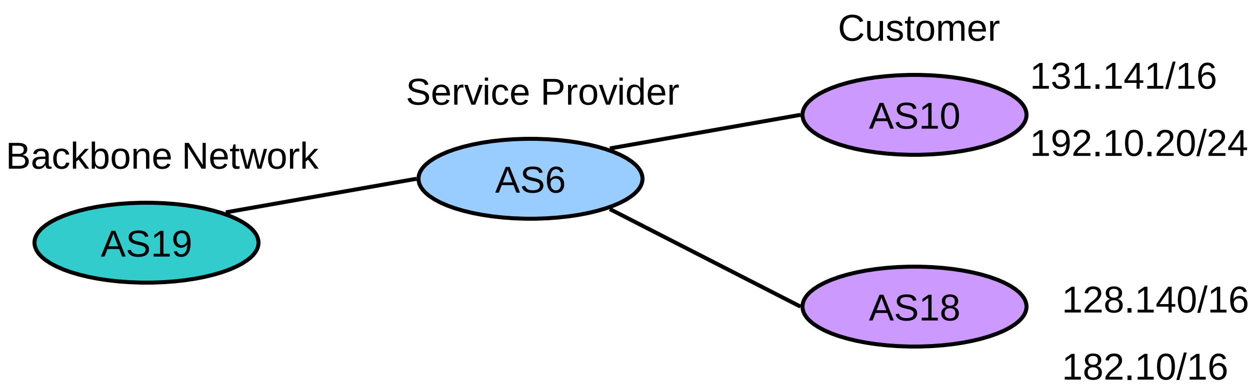
- Customer AS: how to reach the outside world?
  - It is a stub AS
  - Assume only the border router speaks BGP (interior routers do not)
  - How do non-BGP speakers in AS10 learn how to reach the Internet?
  - The border router "injects" a default path into the intradomain protocol (e.g. OSPF or RIP)
  - All traffic from customer is sent via the border router to the service provider

Integrating Interdomain and Intradomain



- Service Provider AS6: how to reach customer? (from non-BGP routers inside AS6)
  - The AS6 router bordering with AS10 “injects” into its intradomain protocol that it has a “link” of cost X to networks 131.141/16 and 192.10.20/24
  - The AS6 router bordering with AS18 “injects” into its intradomain protocol that it has a “link” of cost X to networks 128.140/16 and 182.10/216
  - Any non-BGP-Speaker router insider the service provider will reach these networks via the border routers.

Integrating Interdomain and Intradomain



- Backbone networks: how to reach customer
  - Too many prefixes to inject into intradomain protocol
  - All routers speak BGP
  - Via the iBGP protocol, they share their reachability information (the networks and their AS path to get to them)
  - To reach a specific network, iBGP says which border router is needed, and the intradomain protocol gives the next hop to this router.

Problems with BGP

- Instability
  - Route flapping
  - Arbitrary path decisions can lead to route flapping
  - Not guaranteed to converge
- Over 100,000 network prefixes in some routers (without using defaults)