CS356 Cloud Computing - 3 - Networking

David Pouliot
Southern Oregon University

Network Layer

Network layer

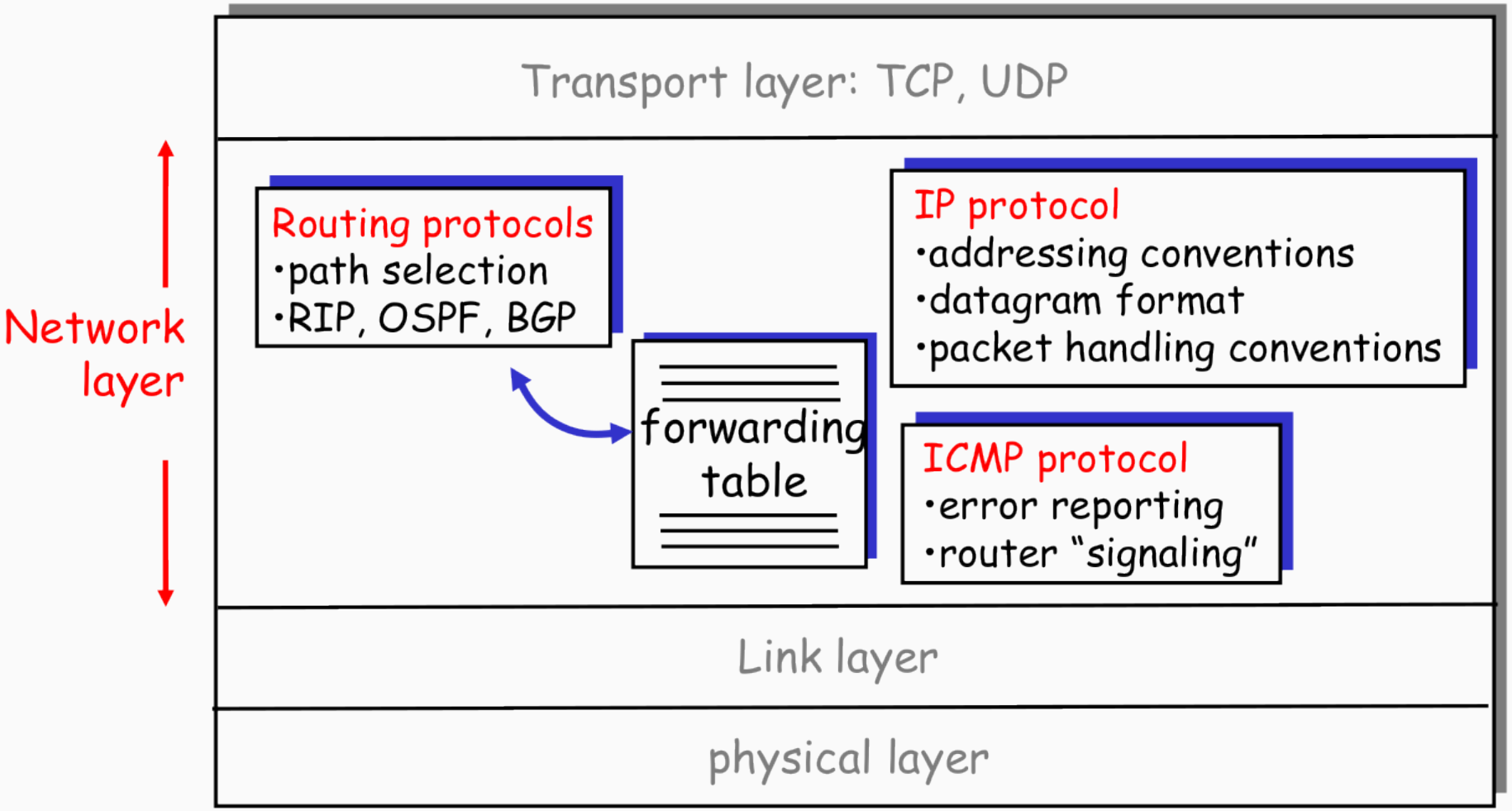
- Send datagram from one host to another
- Network layer protocols in every host, router

Network layer functions

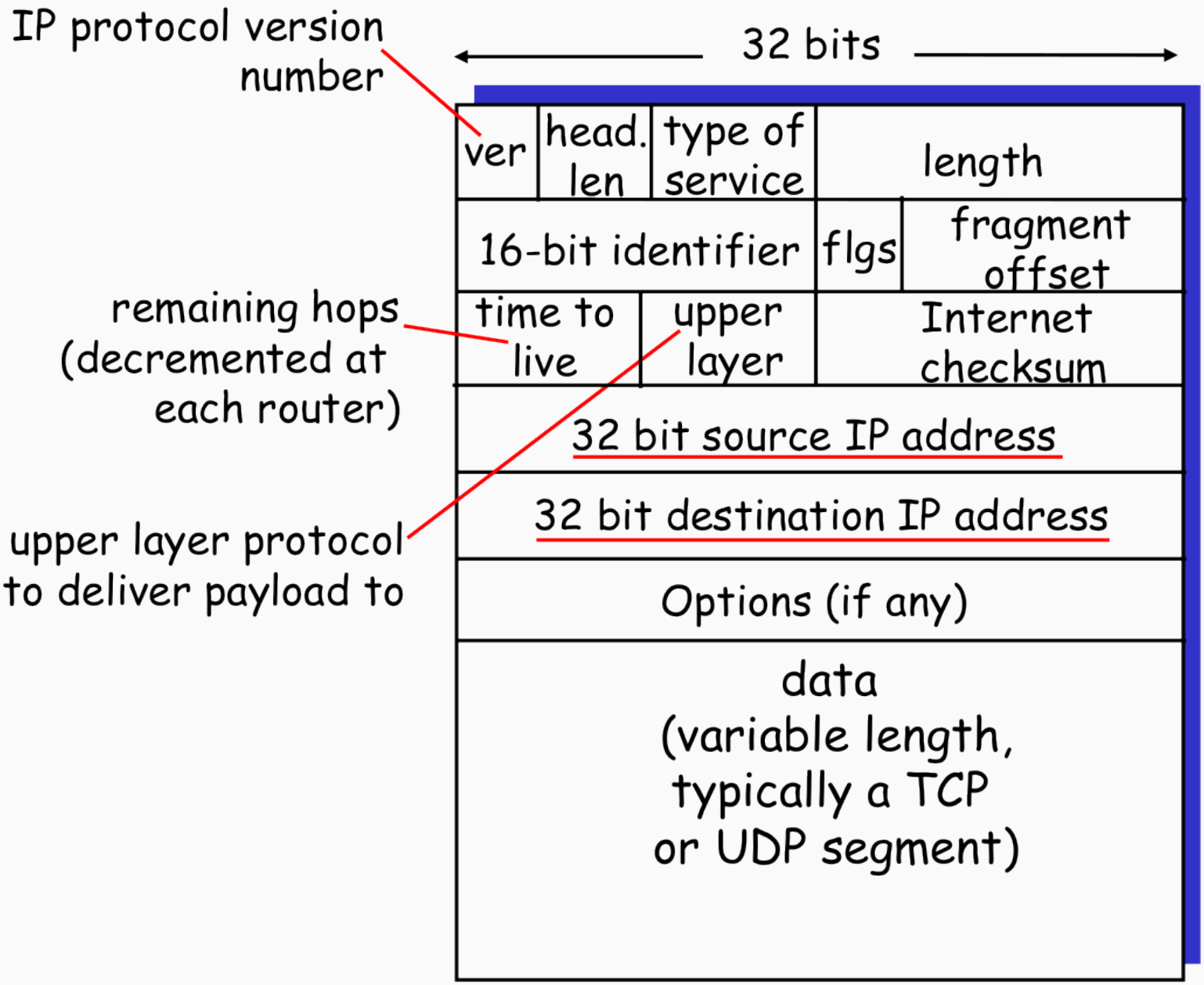
- Connection support
- Delivery semantics
- Security
- Demux to upper layer
- Routing
- Addressing
- Many historical examples, but only one really matters ...

The Internet Network layer

Host, router network layer functions:



IP datagram format



IP connection setup

- Hourglass design
- No support for network layer connections
  - Datagram service
  - Connection semantics only at higher layer
  - Compare to phone network...

IP delivery semantics

- No reliability guarantees
- No ordering guarantees
- No broadcast (255.255.255.255) not forwarded
- No multicast (supported in address space, but no longer used)
  - 224.0.0.0 to 239.255.255.255
- Mostly unicast
- Recently, anycast
  - IP address that has many machines associated with it
  - "Reach any one of them"
  - Done with some routing protocol hacks....

Example: Cloudflare's 1.1.1.1

THE VERGE TECH SCIENCE CULTURE CARS REVIEWS MORE

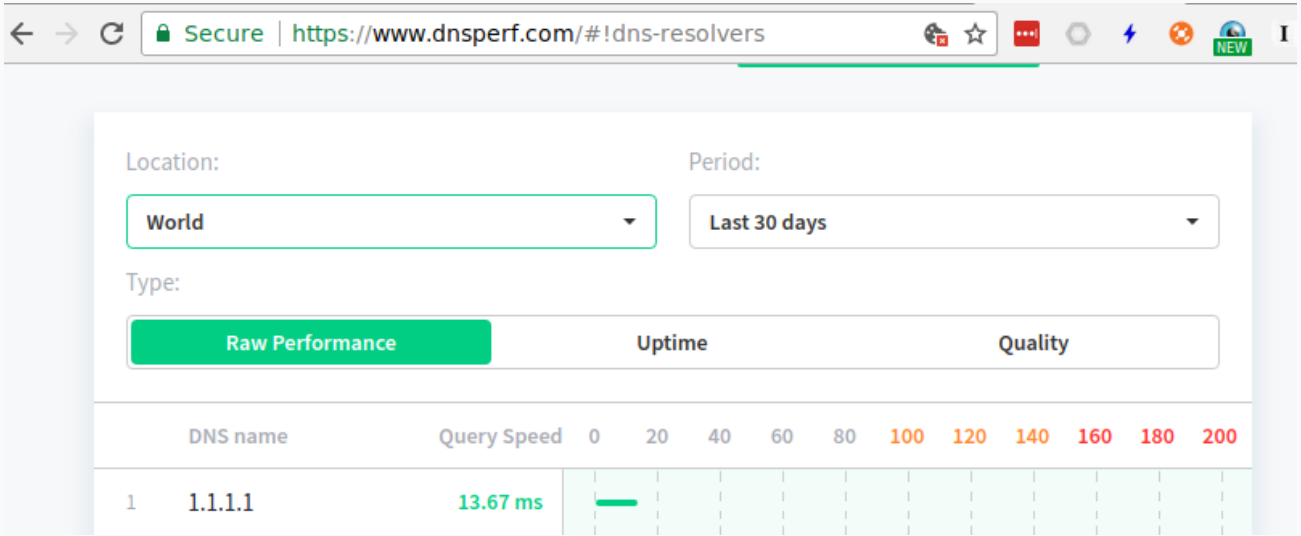
TECH CYBERSECURITY

Cloudflare launches 1.1.1.1 DNS service that will speed up your internet

84

Not an April Fools' prank

By Tom Warren | @tomwarren | Apr 1, 2018, 10:15am EDT



IP security

- Weak support for integrity
  - IP header checksum
  - Leaves data integrity to TCP/UDP

IP security

- Weak support for integrity
  - IP header checksum
  - Leaves data integrity to TCP/UDP
- No support for secrecy

IP security

- Weak support for integrity
  - IP header checksum
  - Leaves data integrity to TCP/UDP
- No support for secrecy
- No support for authenticity
  - Even source IP address can be faked!
  - Hosts trusted to provide legitimate address in packets (Leads to IP spoofing attacks)

IP security

- Weak support for integrity
  - IP header checksum
  - Leaves data integrity to TCP/UDP
- No support for secrecy
- No support for authenticity
  - Even source IP address can be faked!
  - Hosts trusted to provide legitimate address in packets (Leads to IP spoofing attacks)
- IPsec
  - Retrofit IP network layer with encryption and authentication

IP security

- Weak support for integrity
  - IP header checksum
  - Leaves data integrity to TCP/UDP
- No support for secrecy
- No support for authenticity
  - Even source IP address can be faked!
  - Hosts trusted to provide legitimate address in packets (Leads to IP spoofing attacks)
- IPsec
  - Retrofit IP network layer with encryption and authentication
- Similar issues as with WPA
  - On other side of IPSec, payload decrypted

IP demux to upper layer

- Protocol type field (e.g. next innermost doll)
- Control messaging
  - 1 = ICMP
- Transport layers
  - 6 = TCP
  - 17 = UDP
- Tunneling (often used to create virtual networks on cloud platforms)
  - 41 = IPv6 encapsulation within IPv4
  - 47 = GRE (Generic Routing Encapsulation)
- Routing
  - 88 = EIGRP
  - 89 = OSPF

https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers

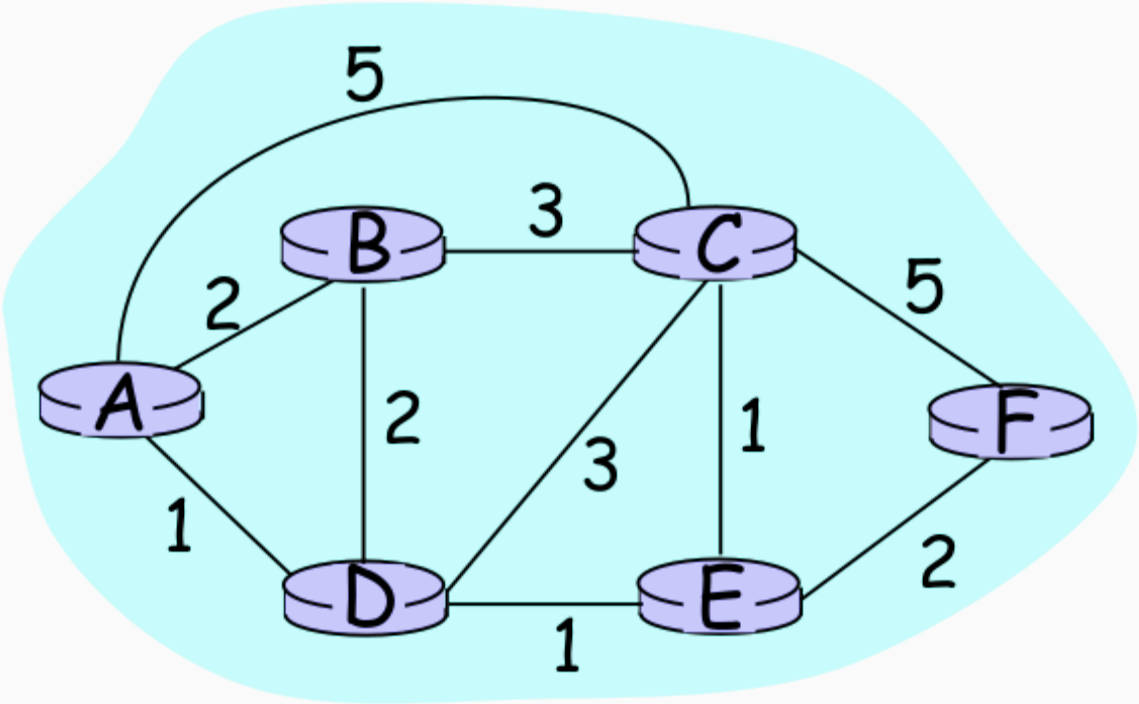
IP routing

- Internet routing done via hop-by-hop forwarding based on destination IP address
- Each router builds forwarding table of..
  - Destination IP $=>$ Next-Hop IP address
- Each router runs a routing protocol and algorithm to create forwarding table

Routing protocols and algorithms

Goal: determine "good" path (sequence of routers) thru network from source to dest.

- Graph abstraction for routing algorithms:
- Routing algorithms find minimum cost paths through graph



Two main kinds of routing algorithms

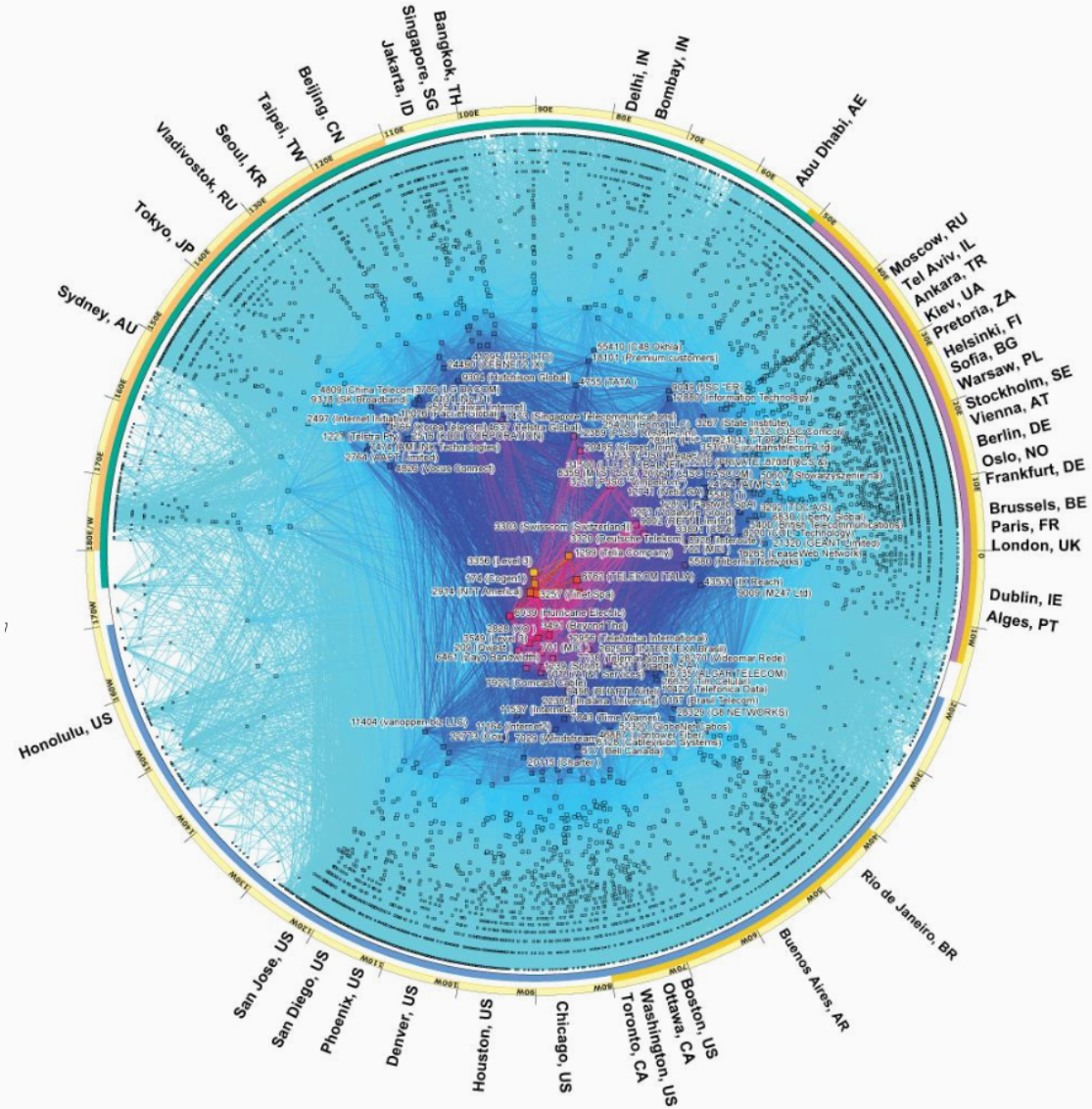
- Link-state algorithms
  - e.g. Dikjstra's shortest-path algorithm
  - Global information
    - Broadcast link cost information to all routers in network
    - Have each router calculate shortest path to destinations
  - Typically done on smaller, edge networks

Two main kinds of routing algorithms

- Link-state algorithms
  - e.g. Dikjstra's shortest-path algorithm
  - Global information
    - Broadcast link cost information to all routers in network
    - Have each router calculate shortest path to destinations
  - Typically done on smaller, edge networks
- Distance-vector algorithms
  - e.g. Bellman-Ford algorithm
  - Decentralized information
  - Router knows physically-connected neighbors, link costs to neighbors
    - Iteratively exchange information with neighbors and recompute routes
  - Done within and between large, backbone networks

Routing issue #1: Scale

- Flat routing doesn't scale
  - 200 million+ destinations
  - Storage
    - Can't store all dest's in routing tables
  - Computation
    - Algorithms perform poorly at that scale
  - Bandwidth
    - Link and routing table exchanges would swamp links!



Routing issue #2: Autonomy

- Network admins need to control routing in their own networks they manage
  - Require administrative autonomy
  - Require isolation of networks from each other
  - Route changes within SOU should be hidden to anyone outside of SOU
- Motivates...

Internet Routing Hierarchy

- Key observation
  - Need less information with increasing distance to destination
  - Saves table size and reduces update traffic

Internet Routing Hierarchy

- Key observation
  - Need less information with increasing distance to destination
  - Saves table size and reduces update traffic
- Implemented via "autonomous systems" (AS)
  - Network divided into regions with administrative autonomy

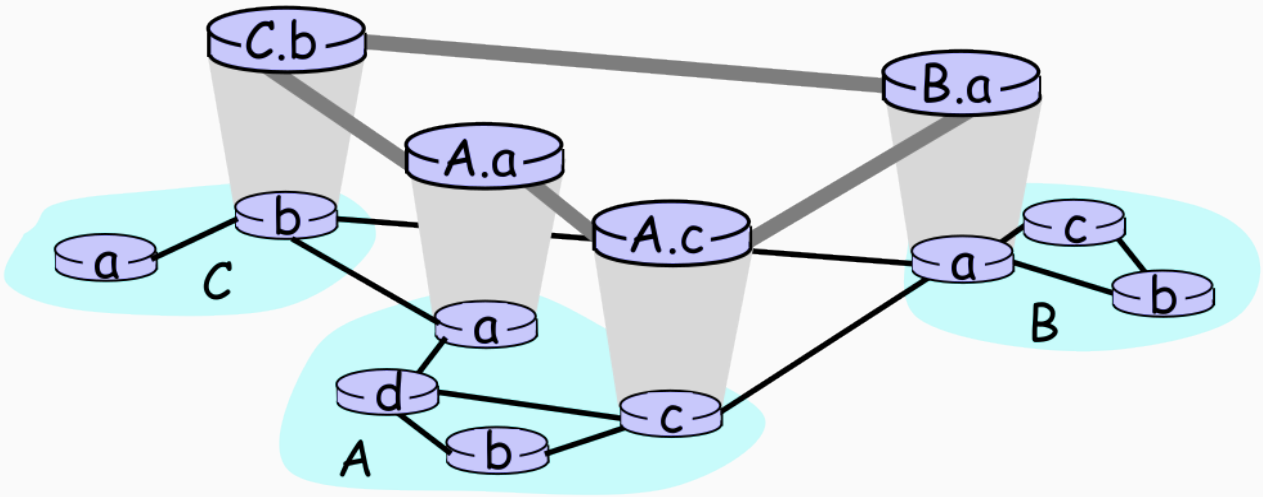
Internet Routing Hierarchy

- Key observation
  - Need less information with increasing distance to destination
  - Saves table size and reduces update traffic
- Implemented via "autonomous systems" (AS)
  - Network divided into regions with administrative autonomy
- Within AS
  - Routers run same routing protocol
    - "Intra-AS" or Interior Gateway routing protocol (IGP)
  - Each node has routes to every other node in area
  - Each node has routes to get to any nodes outside of area
    - Done via a "border router"
    - Packets destined outside area routed to nearest border router

Internet Routing Hierarchy

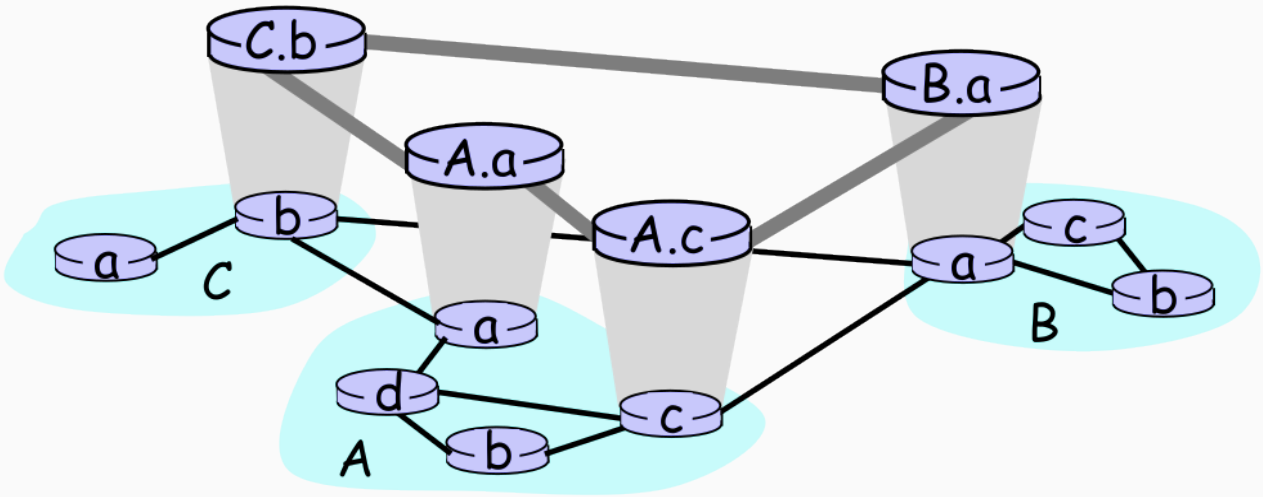
- Key observation
  - Need less information with increasing distance to destination
  - Saves table size and reduces update traffic
- Implemented via "autonomous systems" (AS)
  - Network divided into regions with administrative autonomy
- Within AS
  - Routers run same routing protocol
    - "Intra-AS" or Interior Gateway routing protocol (IGP)
  - Each node has routes to every other node in area
  - Each node has routes to get to any nodes outside of area
    - Done via a "border router"
    - Packets destined outside area routed to nearest border router
- Between ASes
  - Border routers run "Inter-AS" or Border Gateway routing protocol (BGP) with border routers in other AS's

Internet Routing Hierarchy example



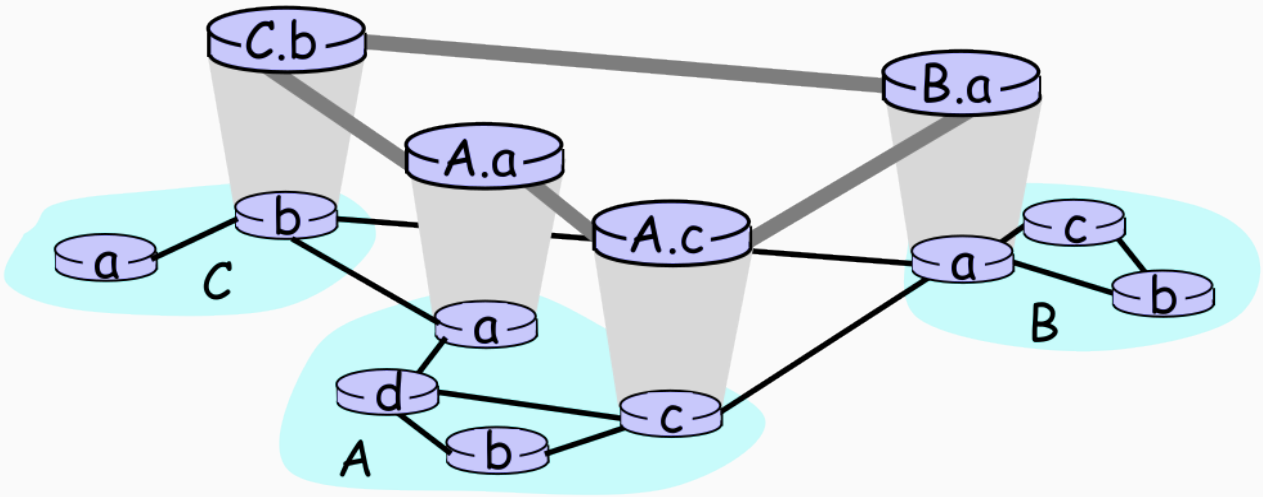
- Addresses in B combined into single route entry pointing to B.a

Internet Routing Hierarchy example



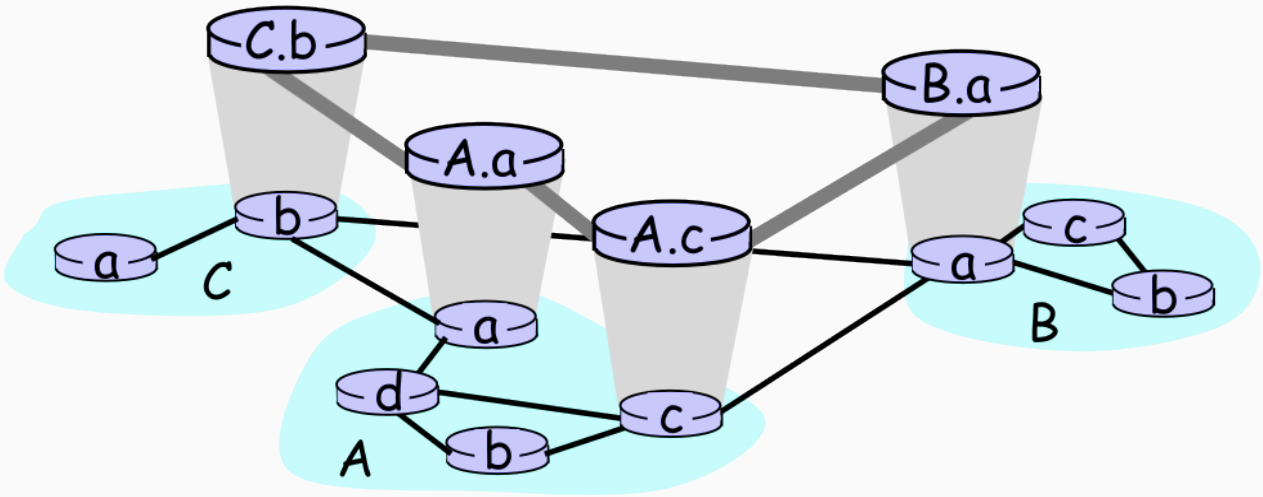
- Addresses in B combined into single route entry pointing to B.a
- Addresses in C combined into single route entry pointing to C.b

Internet Routing Hierarchy example



- Addresses in B combined into single route entry pointing to B.a
- Addresses in C combined into single route entry pointing to C.b
- Addresses in A combined into two route entries pointing to A.a and A.c

Internet Routing Hierarchy example



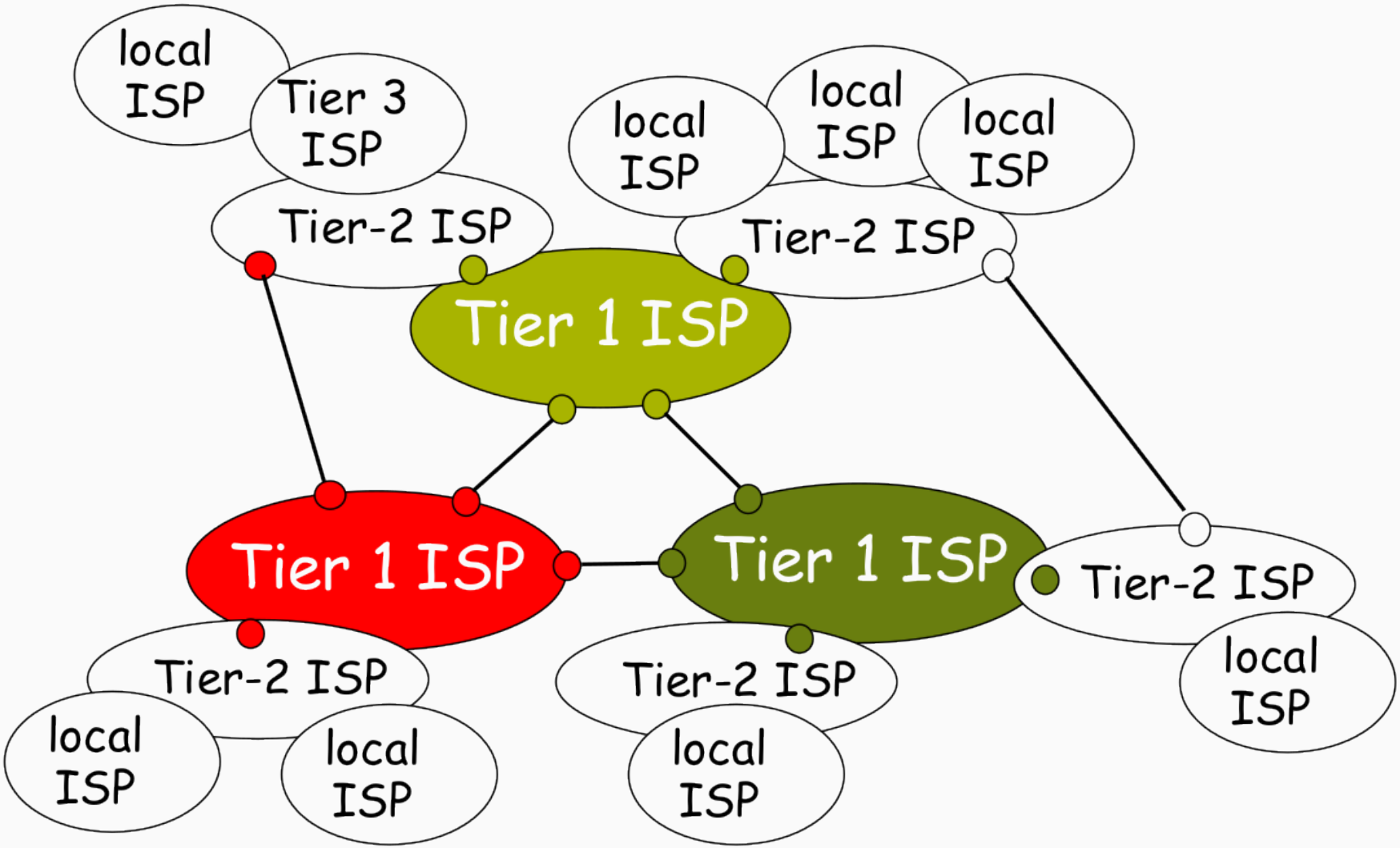
- Addresses in B combined into single route entry pointing to B.a
- Addresses in C combined into single route entry pointing to C.b
- Addresses in A combined into two route entries pointing to A.a and A.c
- Nodes in A, B, and C have no information about individual nodes in other ASes C (only an aggregate route to them)
- Routing done between aggregates

Internet routing hierachy

- At top of hierarchy: “tier-1” ISPs
  - Verizon, Sprint, CenturyLink, AT&T, Cable and Wireless, Google
  - National/international coverage
- Peer with each other in multiple geographic locations in major cities
- ISPs at any tier with a well-known, unique ”AS number”
  - AS numbers, the IP addresses they ”own”, and their location/country well-known
  - Important for attribution of attacks and mistakes

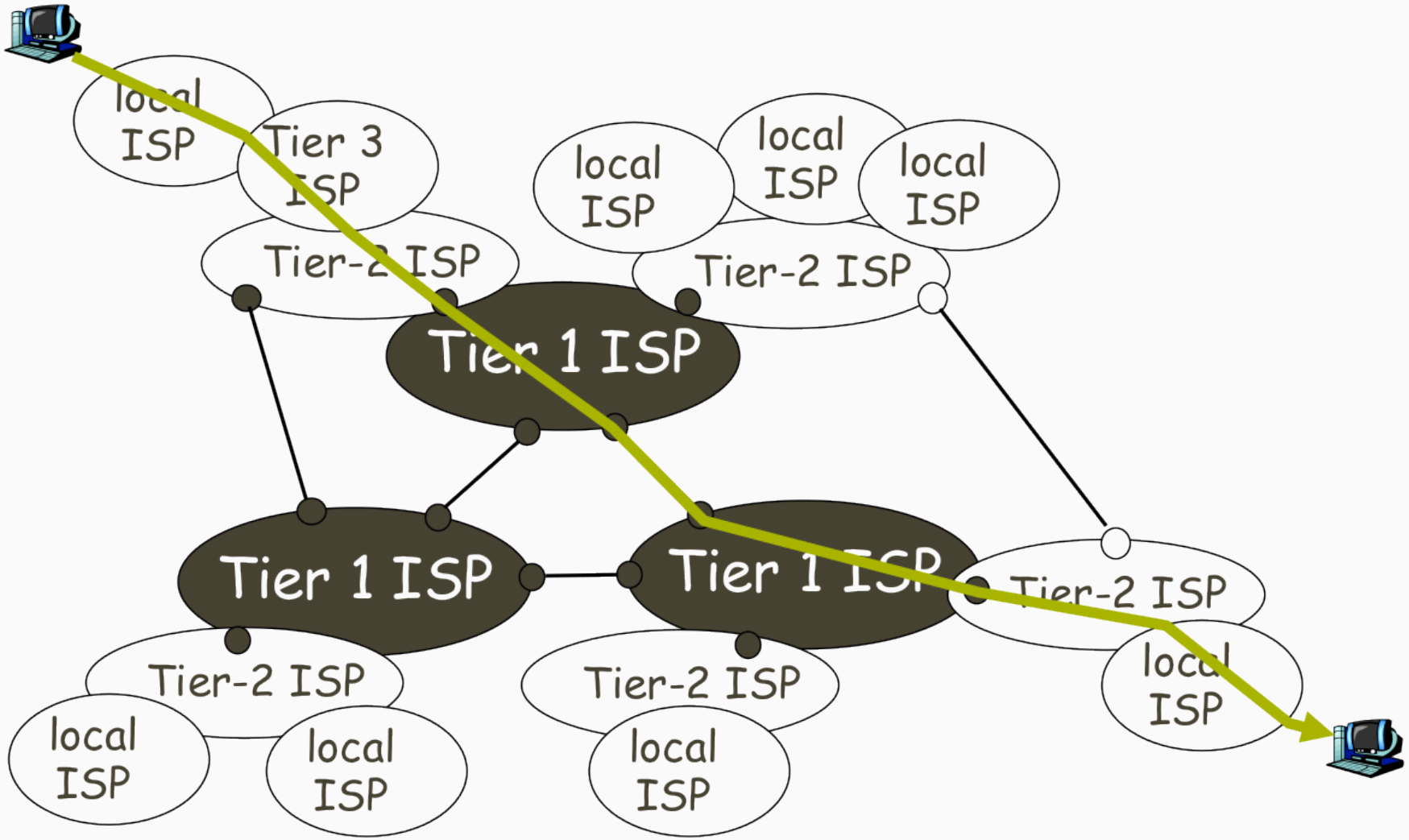
Internet routing hierarchy

- "Tier-2" ISPs: smaller (often regional) ISPs
- "Tier-3" ISPs and local ISPs

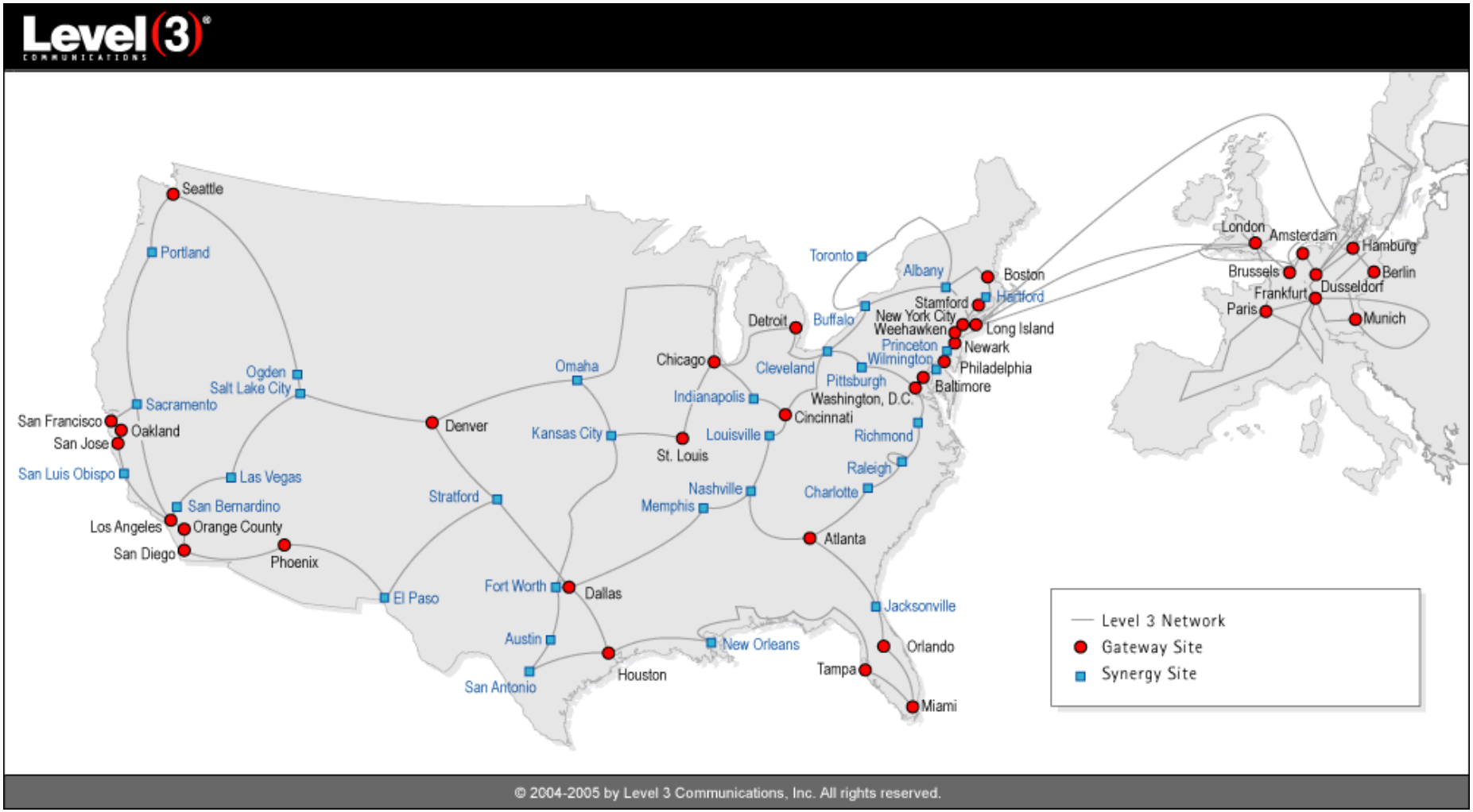


Internet structure: network of networks

- a packet passes through many networks!



Example Tier-1 ISP: Level 3 / CenturyLink



Inter-AS routing

- Routing in between different autonomous systems in hierarchy
- Done using BGP (Border Gateway Protocol)
  - Uses distance-vector style algorithms
  - Treats each AS as a node in a graph
  - Each AS has a route that aggregates all destinations within it (for scale)
- BGP messages exchanged using TCP
  - Simplifies BGP, but ....

Routing issue #3: Trust

- **Authentication not part of TCP/IP**
  - BGP TCP spoofing attack
  - Bogus route advertisements
    - Route advertisements are not authenticated (no public-key infrastructure)
- Result: Anyone can advertise a route to a site in order to redirect traffic towards itself!

IP addressing

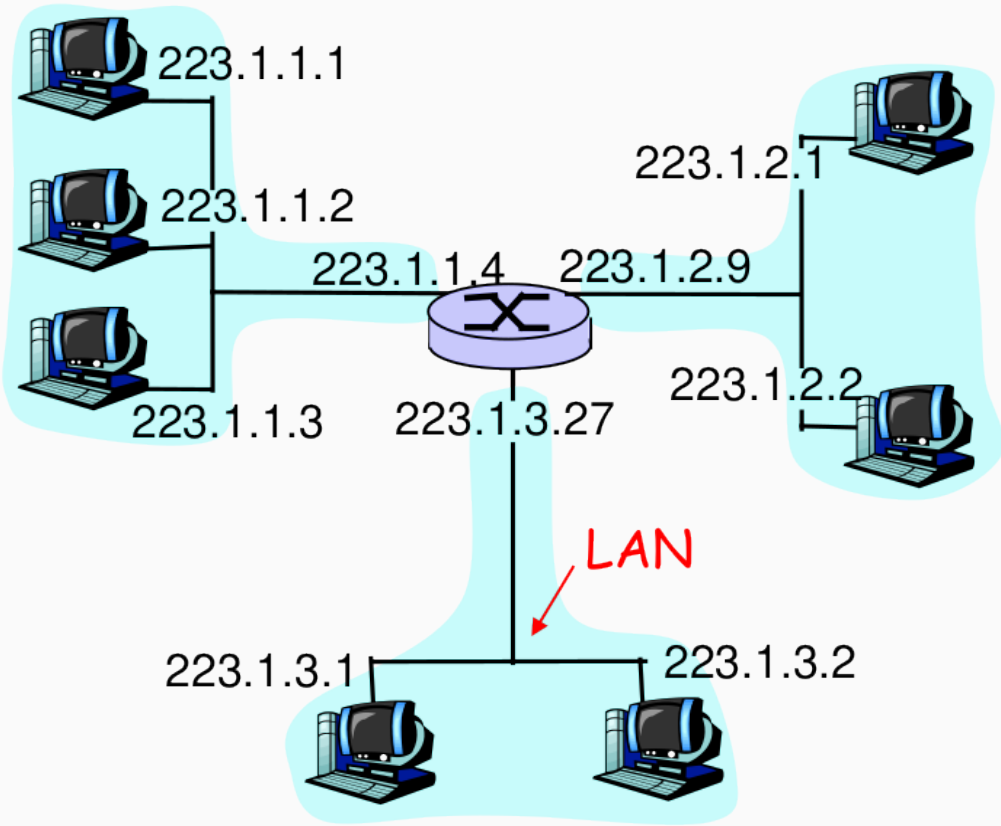
- IP address:
  - 32-bit identifier for host/router network interface
    - Specified by 4 individual bytes from 0 to 255 (0x00 to 0xFF)

      131.252.220.1 = 10000011 11111100 11011100 00000001

      131 252 220 1
    - Recall: What is the total number of addresses available for use?
- IP addresses associated with an interface, not host or router
  - Routers typically have multiple interfaces
  - Host may have multiple interfaces (both real and virtual!)

IP addressing

- Aggregation done to support routing scalability
- IP address split into two
  - Network part (high order bits)
  - Host part (low order bits)
- What's a network ?
  - all interfaces that can physically reach each other without intervening router
  - each interface shares the same network part of IP address



network consisting of 3 IP networks (for IP addresses starting with 223, first 24 bits are network address)

IP addressing and networks via CIDR

- Classless Inter-Domain Routing
  - Network part of IP address can be arbitrarily sized
  - Done throughout routing infrastructure to implement hierarchy

IP addressing and networks via CIDR

- Classless Inter-Domain Routing
  - Network part of IP address can be arbitrarily sized
  - Done throughout routing infrastructure to implement hierarchy
- Mechanism
  - Aggregate adjacent addresses together (supernet)
  - Split adjacent addresses apart (subnet)

IP addressing and networks via CIDR

- Classless Inter-Domain Routing
  - Network part of IP address can be arbitrarily sized
  - Done throughout routing infrastructure to implement hierarchy
- Mechanism
  - Aggregate adjacent addresses together (supernet)
  - Split adjacent addresses apart (subnet)
  - Single integer specifies the demarcation between network and host parts
  - Notation:`<prefix>/<bits in prefix>` $\rightarrow$ 200.23.16.0/23

IP addressing and networks via CIDR

- Classless Inter-Domain Routing
  - Network part of IP address can be arbitrarily sized
  - Done throughout routing infrastructure to implement hierarchy
- Mechanism
  - Aggregate adjacent addresses together (supernet)
  - Split adjacent addresses apart (subnet)
  - Single integer specifies the demarcation between network and host parts
  - Notation:`<prefix>/<bits in prefix>` $\rightarrow$ 200.23.16.0/23

$\leftarrow$ variable network part $\rightarrow$ $\leftarrow$ host part $\rightarrow$

11001000 00010111 00010000 00000000

IP addressing and networks via CIDR

- Classless Inter-Domain Routing
  - Network part of IP address can be arbitrarily sized
  - Done throughout routing infrastructure to implement hierarchy
- Mechanism
  - Aggregate adjacent addresses together (supernet)
  - Split adjacent addresses apart (subnet)
  - Single integer specifies the demarcation between network and host parts
  - Notation:`<prefix>/<bits in prefix>` → 200.23.16.0/23

    ← variable network part → ← host part →

    11001000 00010111 00010000 00000000
    - Host range is a power of 2 ranging from all 0s to all 1s.

      11001000 00010111 00010001 11111111

      200.23.16.0 to 200.23.17.255 (512 addresses)

Subnetting walkthrough

- Take large block and split into smaller sub-networks

Subnetting walkthrough

- Take large block and split into smaller sub-networks
- Example: Split the following network into 4 equal subnetworks
  - 131.252.0.0/22
  - Expand out address. . .

    10000011 . 11111100 . 00000000 . 00000000
  - Q1: How many hosts are on this network?
  - Q2: How many hosts will be on each subnetwork?

Subnetting walkthrough

- Take large block and split into smaller sub-networks
- Example: Split the following network into 4 equal subnetworks
  - 131.252.0.0/22
  - Expand out address...

    10000011 . 11111100 . 00000000 . 00000000
  - Q1: How many hosts are on this network?
  - Q2: How many hosts will be on each subnetwork?
- Split into 4 parts using next 2 significant bits

  10000011 . 11111100 . 000000**00** . 00000000

  10000011 . 11111100 . 000000**01** . 00000000

  10000011 . 11111100 . 000000**10** . 00000000

  10000011 . 11111100 . 000000**11** . 00000000

Subnetting walkthrough

- Take large block and split into smaller sub-networks
- Example: Split the following network into 4 equal subnetworks
  - 131.252.0.0/22
  - Expand out address...

    10000011 . 11111100 . 00000000 . 00000000
  - Q1: How many hosts are on this network?
  - Q2: How many hosts will be on each subnetwork?
- Split into 4 parts using next 2 significant bits

  10000011 . 11111100 . 000000**00** . 00000000

  10000011 . 11111100 . 000000**01** . 00000000

  10000011 . 11111100 . 000000**10** . 00000000

  10000011 . 11111100 . 000000**11** . 00000000
  - Solution – 131.252.0.0/24 131.252.1.0/24

    131.252.2.0/24 131.252.3.0/24

Subnetting problem

- With the person sitting next to you, split the following network into 16 equal subnetworks
  - 131.252.128.0/17
  - 10000011 . 11111100 . 10000000 . 00000000

Subnetting problem

- With the person sitting next to you, split the following network into 16 equal subnetworks
  - 131.252.128.0/17
  - 10000011 . 11111100 . 10000000 . 00000000
- Split into 16 parts
  - 10000011 . 11111100 . 1**0000**000 . 00000000
  - 10000011 . 11111100 . 1**0001**000 . 00000000
  - 10000011 . 11111100 . 1**0010**000 . 00000000
  - etc
- Solution
  - 131.252.128.0/21
  - 131.252.136.0/21
  - 131.252.144.0/21
  - etc

Supernetting walkthrough

- Take small blocks and combine to reduce routes (supernetting)
- Combine the following class C networks into one larger network
  - 131.252.0.0/24
  - 131.252.1.0/24

Supernetting walkthrough

- Take small blocks and combine to reduce routes (supernetting)
- Combine the following class C networks into one larger network
  - 131.252.0.0/24
  - 131.252.1.0/24

    10000011.11111100.00000000.*

    10000011.11111100.00000001.*

Supernetting walkthrough

- Take small blocks and combine to reduce routes (supernetting)
- Combine the following class C networks into one larger network
  - 131.252.0.0/24
  - 131.252.1.0/24

    10000011.11111100.0000000**0**.*

    10000011.11111100.0000000**1**.*
    - Answer:

      10000011.11111100.0000000*.*

      131.252.0.0/**23**

Supernetting walkthrough

- Can you combine the following class C networks into a larger /23?
  - 131.252.1.0/24 10000011.11111100.00000001.*
  - 131.252.2.0/24 10000011.11111100.00000010.*

Supernetting walkthrough

- Can you combine the following class C networks into a larger /23?
  - 131.252.1.0/24 10000011.11111100.00000001.*
  - 131.252.2.0/24 10000011.11111100.00000010.*
- No, they do not share the same address prefix!
- Ranges must be aligned properly to be supernetted.
  - Only (131.252.0.0/24 + 131.252.1.0/24) and (131.252.2.0/24 + 131.252.3.0/24) can be combined into a larger /23.

10000011.11111100.00000000.*	10000011.11111100.00000001.*
10000011.11111100.00000010.*	10000011.11111100.00000011.*
131.252.0.0/23	131.252.2.0/23

Supernetting problem

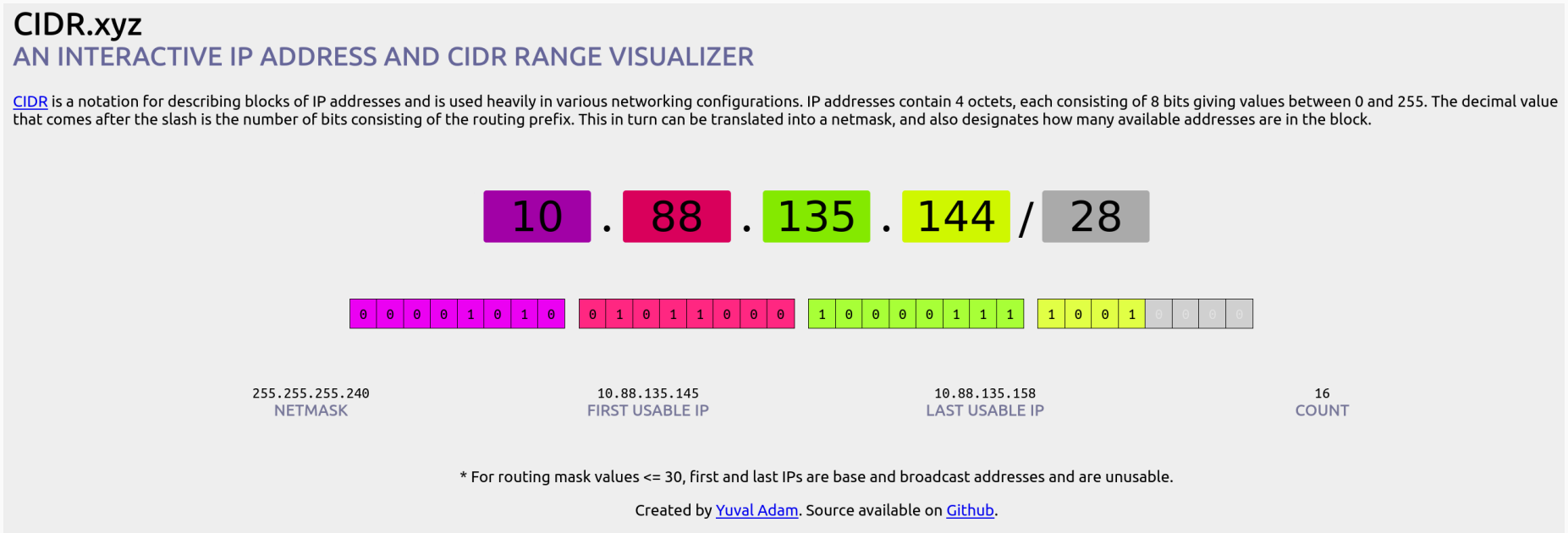
- Combine the following class C networks into one larger network
  - 131.252.0.0/24
  - 131.252.1.0/24
  - 131.252.2.0/24
  - 131.252.3.0/24
  - 131.252.4.0/24
  - 131.252.5.0/24
  - 131.252.6.0/24
  - 131.252.7.0/24

Supernetting problem

- Combine the following class C networks into one larger network
  - 131.252.0.0/24
  - 131.252.1.0/24
  - 131.252.2.0/24
  - 131.252.3.0/24
  - 131.252.4.0/24
  - 131.252.5.0/24
  - 131.252.6.0/24
  - 131.252.7.0/24
- Answer: 131.252.0.0/21

Demo Site

cidr.xyz



Demo

Demo IP CIDR for cloud firewall rules

Special IP Addresses: Loopback

- 127.0.0.1: localhost
  - The self-talk address
  - The "`lo`" interface via `ifconfig`

```
catron <~> 11:47AM % ifconfig
eno1      Link encap:Ethernet  HWaddr 98:90:96:d8:56:e7
          inet addr:10.218.103.22  Bcast:10.218.103.255
           Mask:255.255.255.0
          inet6 addr: fe80::9a90:96ff:fed8:56e7/64 Scope:Link
          inet6 addr: 2610:10:20:1103::22/128 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
...
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
...
catron <~> 11:48AM %
```

Special IP Addresses: Private

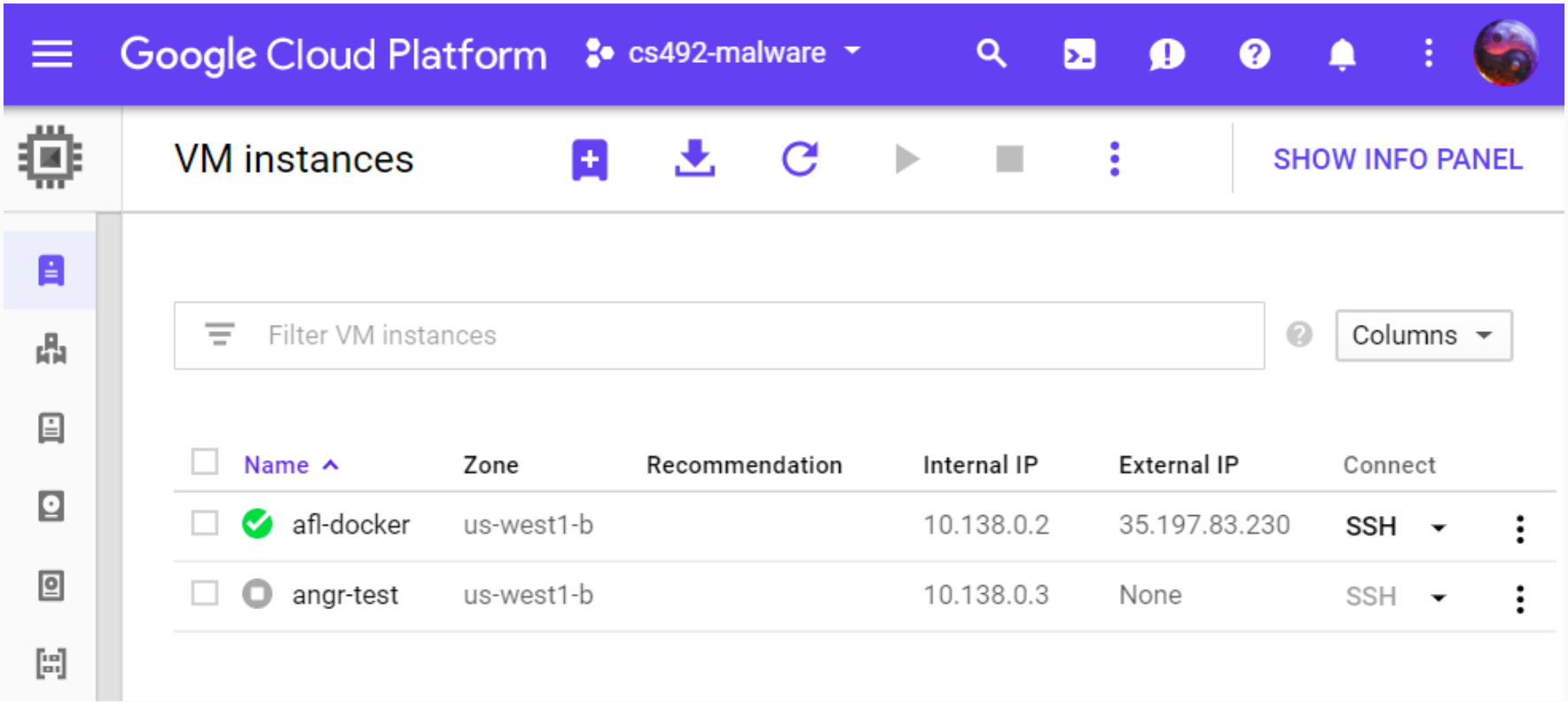
- Private addresses (not globally routable)
- Class A: 10.0.0.0 - 10.255.255.255 (10.0.0.0/8 prefix)
- Class B: 172.16.0.0 - 172.31.255.255 (172.16.0.0/12 prefix)
- Class C: 192.168.0.0 - 192.168.255.255 (192.168.0.0/16 prefix)

Special IP Addresses: Private

- Must go through a machine that has a globally routable IP address

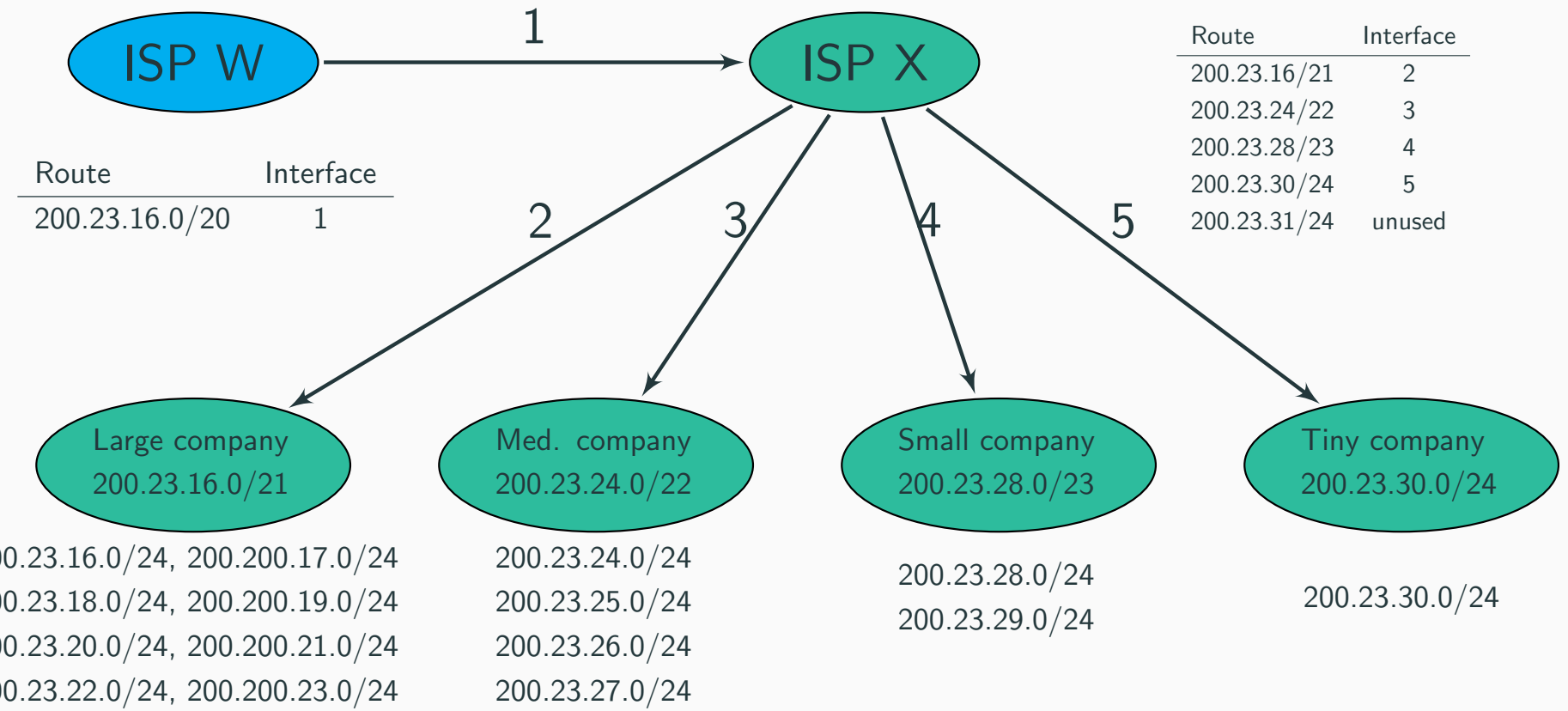
Special IP Addresses: Private

- All Google Cloud internal interfaces use them



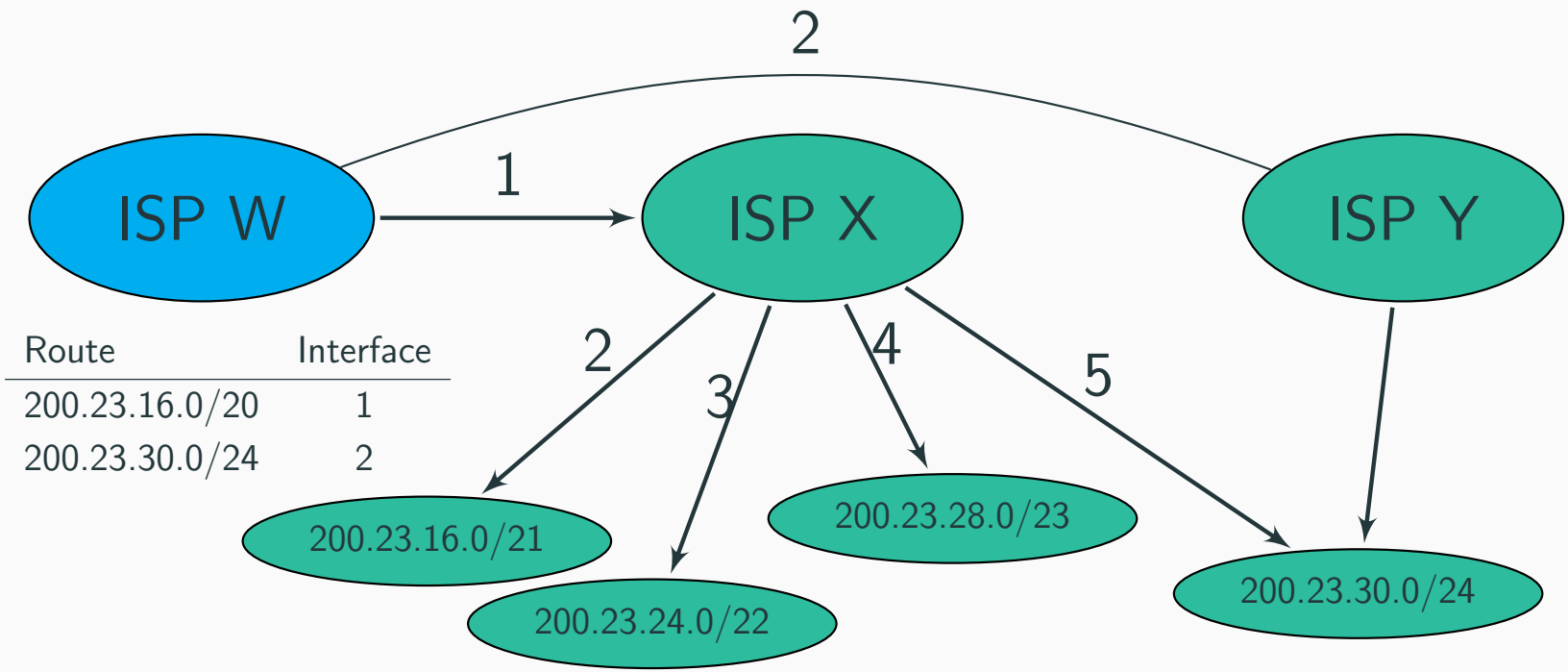
CIDR in practice

ISP X given 16 class C networks (200.23.16.* to 200.23.31.*)
Can advertise a single CIDR route to ISP W (200.23.16.0/20)



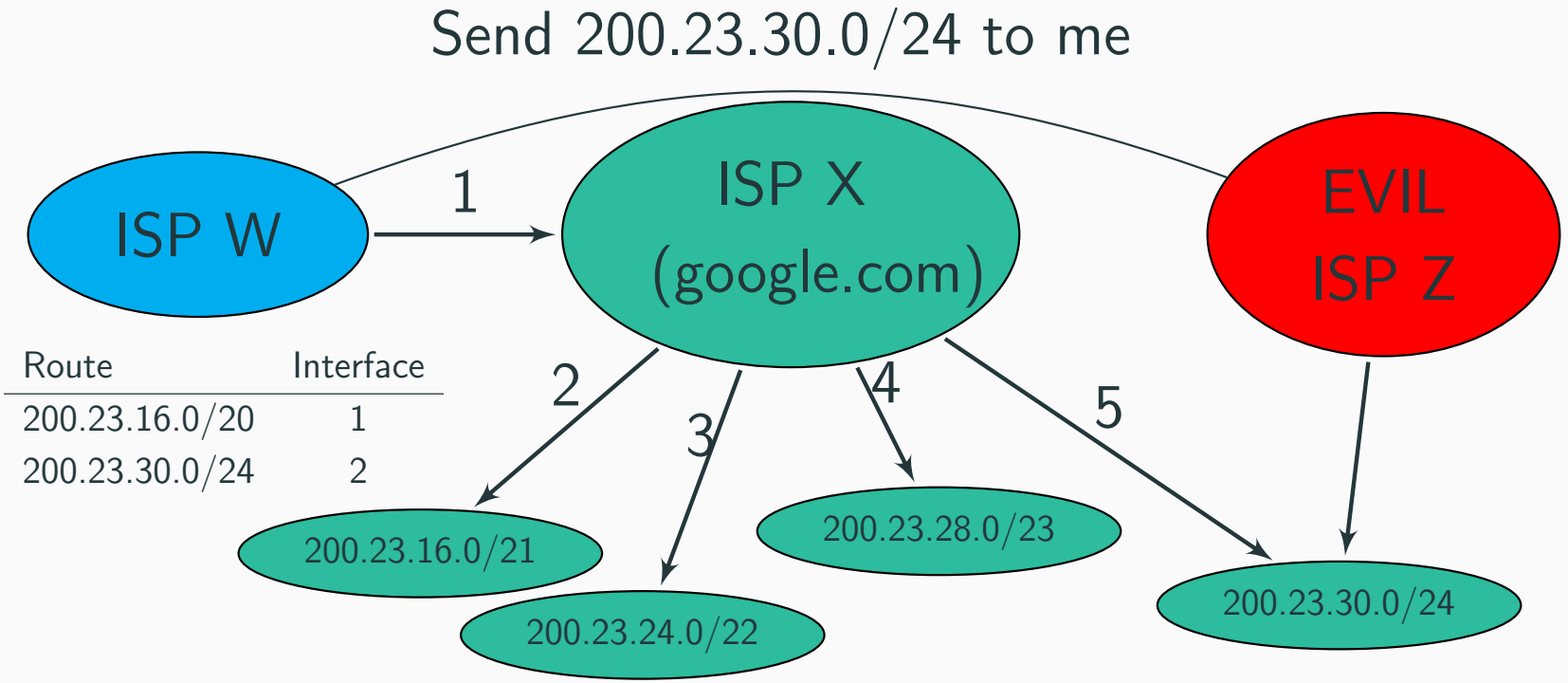
Overlapping routes

- Consider multi-homing for tiny company going to ISP Y
  - 200.23.16.0/20 through ISP X to ISP W, but tiny company would advertise 200.23.30.0/24 to ISP Y, which advertises it to ISP W
- How does ISP W handle overlapping routes from ISP X and Y?
  - Choice is to always follow the most specific route (e.g. /24) when a packet matches both



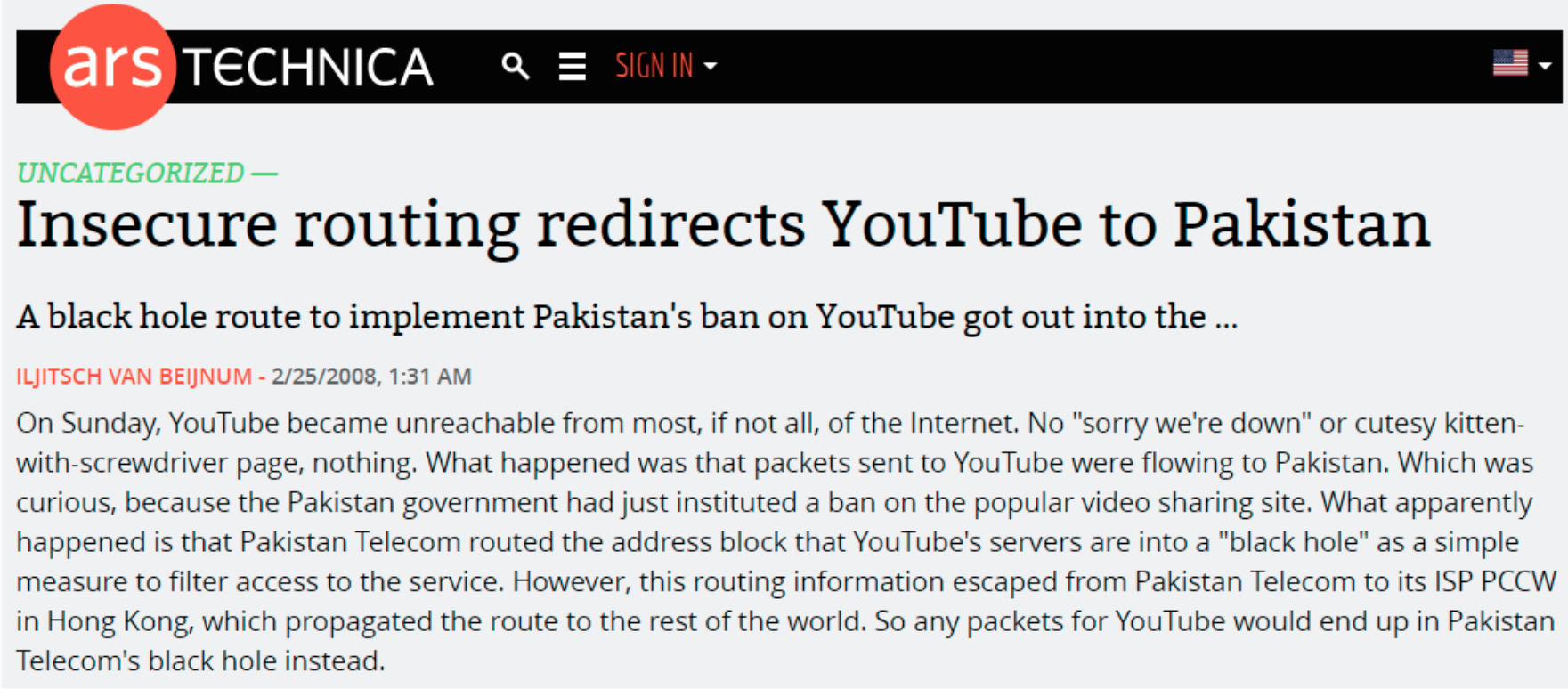
What if?

- YouTube with its network traffic going through X
- Has someone else advertise its prefix?



Trust and routing

- Like Pakistan did (2008)



ars TECHNICA SIGN IN

UNCATEGORIZED —

Insecure routing redirects YouTube to Pakistan

A black hole route to implement Pakistan's ban on YouTube got out into the ...

ILJITSCH VAN BEIJNUM - 2/25/2008, 1:31 AM

On Sunday, YouTube became unreachable from most, if not all, of the Internet. No "sorry we're down" or cutesy kitten-with-screwdriver page, nothing. What happened was that packets sent to YouTube were flowing to Pakistan. Which was curious, because the Pakistan government had just instituted a ban on the popular video sharing site. What apparently happened is that Pakistan Telecom routed the address block that YouTube's servers are into a "black hole" as a simple measure to filter access to the service. However, this routing information escaped from Pakistan Telecom to its ISP PCCW in Hong Kong, which propagated the route to the rest of the world. So any packets for YouTube would end up in Pakistan Telecom's black hole instead.

Trust and routing

- Like Google did to Japan (2017!)

{* NETWORKS *}

Google routing blunder sent Japan's Internet dark on Friday

Another big BGP blunder

Richard Chirgwin Sun 27 Aug 2017 // 22:35 UTC SHARE

Last Friday, someone in Google fat-thumbed a border gateway protocol (BGP) advertisement and sent Japanese Internet traffic into a black hole.

Trust and routing

- Or Russia and Iran did?

The Register®
Biting the hand that feeds IT

'Suspicious' BGP event routed big traffic sites through Russia

Google, Facebook and Microsoft routed through PutinGrad, for no good reason

By Richard Chirgwin 13 Dec 2017 at 08:02

A Border Gateway Protocol (BGP) routing incident saw a bunch of high-profile Internet destinations mis-routed through Russia on Tuesday, US time.

Data Centre ▸ Networks

Hey, don't route the messenger! Telegram redirected through Iran by baffling BGP leak

Fat thumb – or government intervention?

By Richard Chirgwin 1 Aug 2018 at 03:03 18 SHARE

Updated A bunch of Telegram messages went the long way round on Monday: a BGP leak sent people's Telegram chat communications via systems in Iran.

Not likely to be solved

- But we still keep on trying

SecurityIntelligence

NEWS October 12, 2017 @ 10:45 AM

Agencies, Assemble! NIST and DHS Join Forces for BGP Security

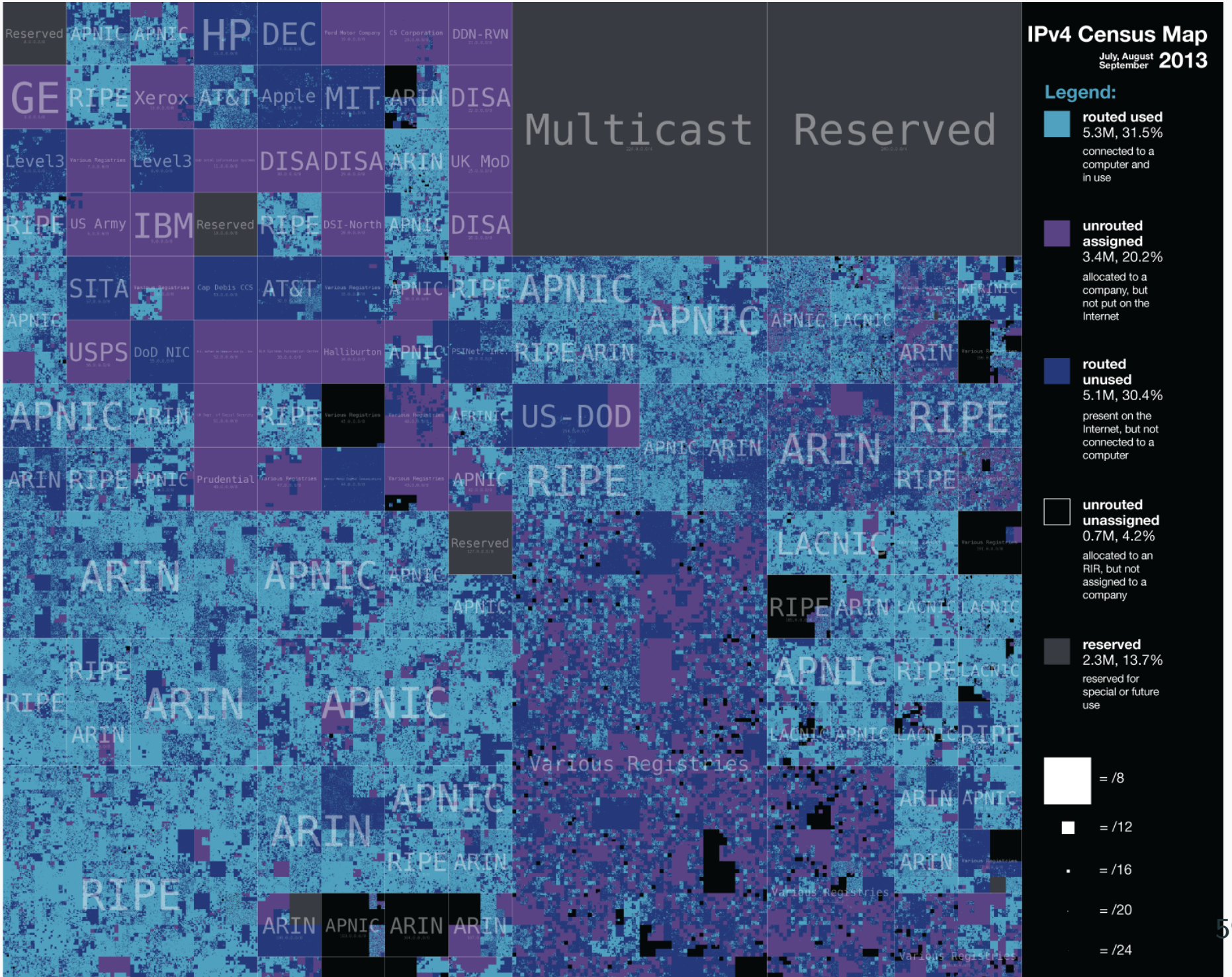
By Douglas Bonderud

The protocol was developed in the late 1980s, when data security wasn't top of mind for emerging internet entities. As a result, BGP hijacks are now, as Wired put it, "the internet's biggest security hole." This prompted two government agencies — the National Institute of Standards and Technology (NIST) and the Department of Homeland Security (DHS) — to team up and develop a new set of security standards to close border gateway loopholes.

Network Address Translation

CIDR and IPv4 address allocation

- CIDR designed to make more efficient use of addresses
  - Many large blocks with few hosts in them
  - CIDR allows one to break blocks up into smaller units


IPv4 Census Map
July, August September 2013
Legend:
routed used
5.3M, 31.5%
connected to a computer and in use
unrouted assigned
3.4M, 20.2%
allocated to a company, but not put on the Internet
routed unused
5.1M, 30.4%
present on the Internet, but not connected to a computer
unrouted unassigned
0.7M, 4.2%
allocated to an RIR, but not assigned to a company
reserved
2.3M, 13.7%
reserved for special or future use
= /8
= /12
= /16
= /20
= /24
Multicast
Reserved
Various Registries

But

- Even with CIDR, IPv4 address space running out

CORE NETWORKING

By Scott Hogg, Network World | SEP 22, 2015 7:25 AM PDT

About

Scott Hogg is a co-founder of HexaBuild.io, an IPv6 consulting and training firm, and has over 25 years of cloud, networking and security experience.

OPINION

ARIN Finally Runs Out of IPv4 Addresses

IPv4 Address Cupboards are Bare in North America.

Data Centre ▸ Networks

OK, this time it's for real: The last available IPv4 address block has gone

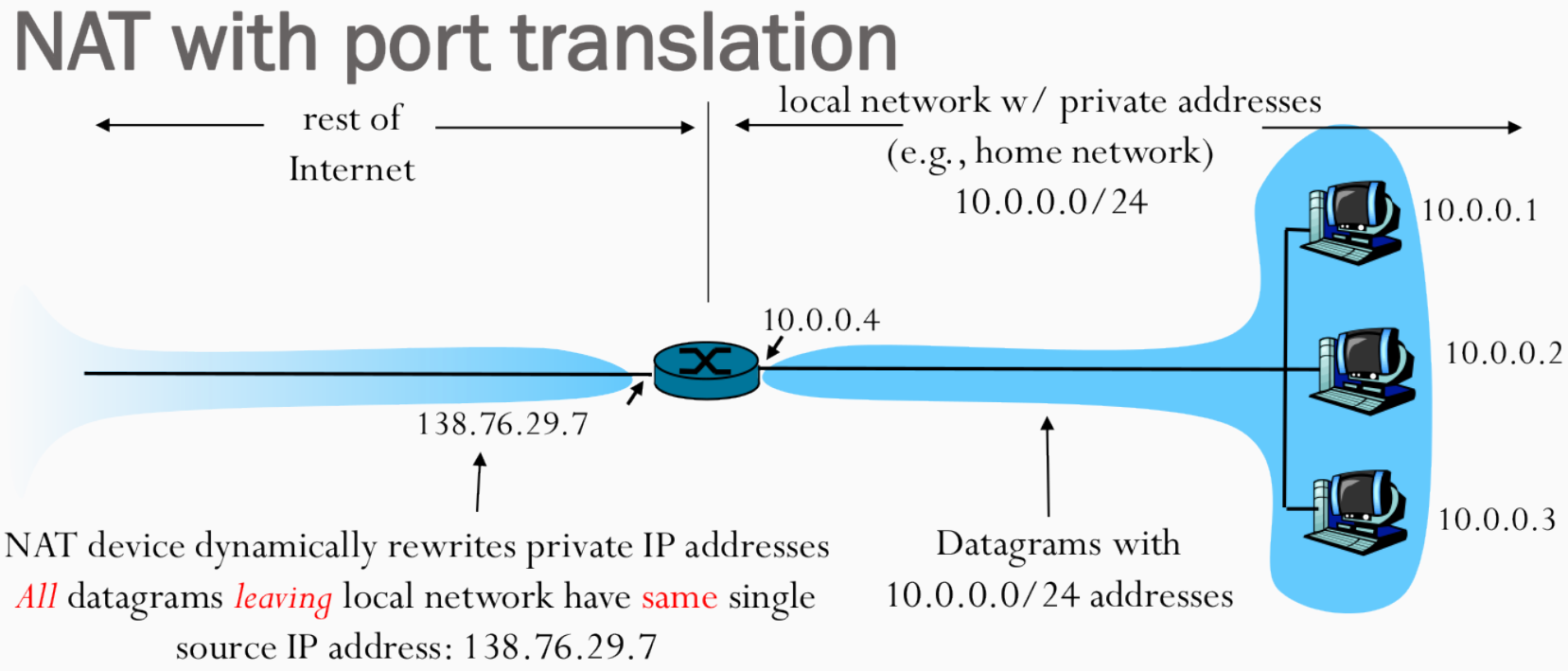
Now for the last time, will you all please shift to IPv6?!

By Kieren McCarthy in San Francisco 18 Apr 2018 at 22:10 211 SHARE ▼

Network Address Translaction (NAT)

- One solution to address space depletion problem
  - **Make it appear that all connections coming from a local network comes from a single IP address**
  - "Statistically multiplex" address/port usage across multiple machines
- Examples
  - Built-into your cable modem at home
  - "NAT and Internet Gateway" in AWS
  - "Cloud NAT" in GCP

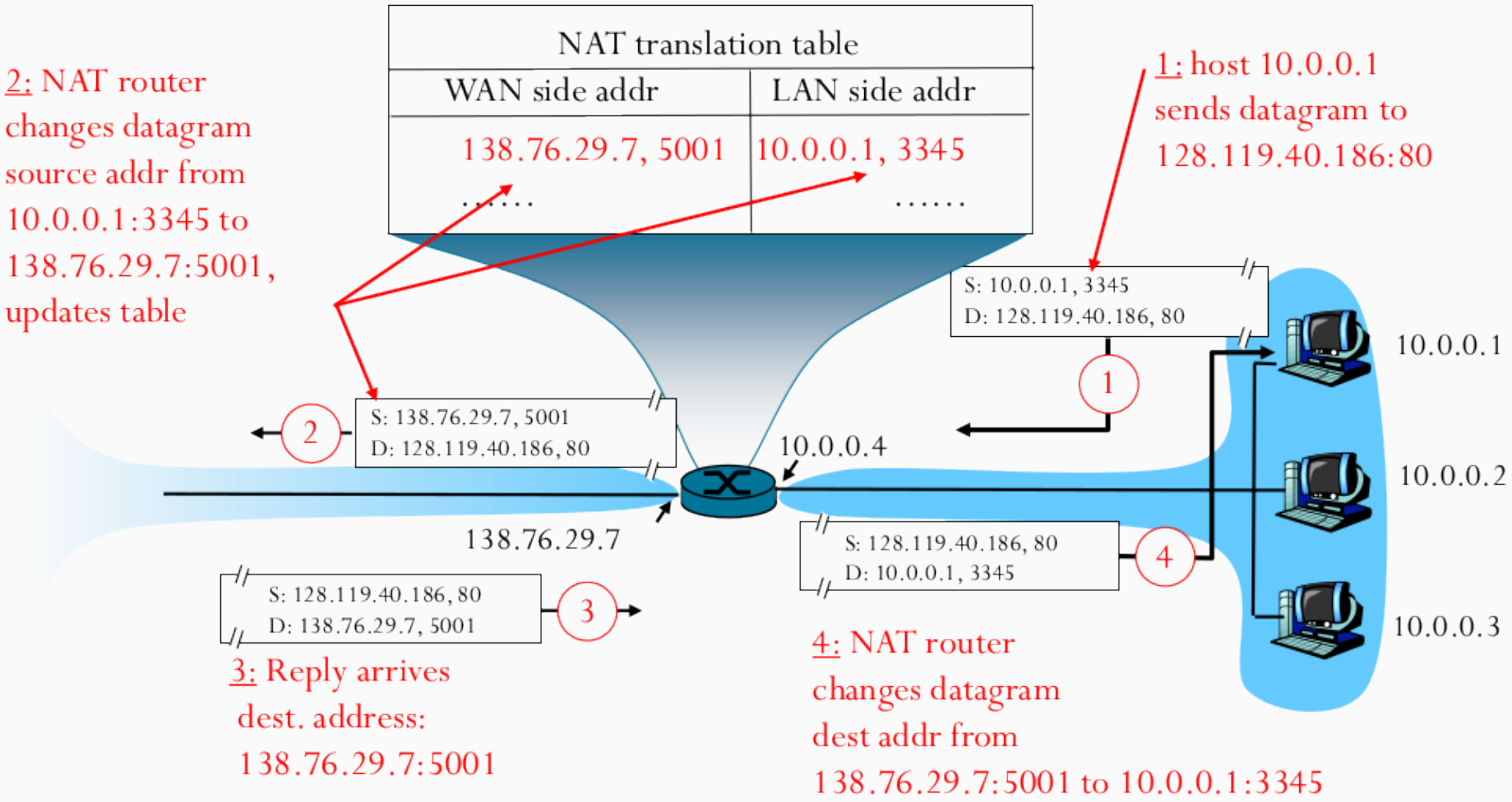
NAT with port translation



But, must ensure unique identifiers per connection to work.

- Connections from 3 machines to google.com must not get confused with each other
- Use the 16-bit transport layer port-number field
- NAT device dynamically rewrites private source IP addresses and source port numbers on connections to the Internet to global IP address and port numbers
- To Internet, entire network looks like 1 machine (with a lot of connections!)

NAT operation

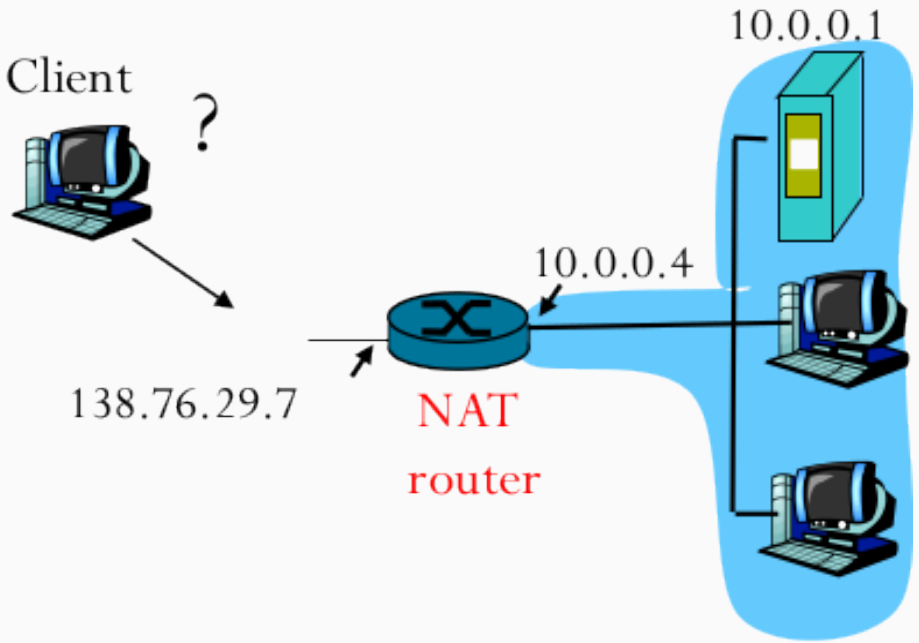


NAT advantages

- Only a single IP address needed from ISP to network multiple devices (avoid depletion)
  - Most cloud providers charge you for an external IP address!
  - GCP charges differently for VMs with
    - No external IP address
    - Ephemeral external IP address (ones that go away when VM is shut off)
    - Static, reserved IP address (ones that stay with VM even when shut off)
- Portability
  - Can change ISP without changing addresses of devices in local network
- Security
  - Devices inside local net not explicitly addressable or visible by outside world

NAT issue #1: No inbound connection

- Client wants to connect to web server at address 10.0.0.1
- Web server has private LAN address not reachable externally
- Only externally visible address: 138.76.29.7
- Must be taken into account for P2P applications



NAT issue #2: Loss of transparency

- Implicit assumption that network header is unchanged in network
  - Key feature that allows one to deploy any application without coordinating with network infrastructure
  - Breaks applications that assume network only touches layer 3
  - **Transport-layer source port numbers re-written by network-layer device (layer violation)**
- New applications
  - Can no longer assume their own IP address is valid!
  - Must never carry IP addresses and ports in application payloads
  - Example: ftp's PORT command
    - To initiate file transfer, client sends its IP address and a port number for ftp server to connect to
    - With client behind a NAT, private address sent!
    - NAT breaks protocol by breaking network transparency!

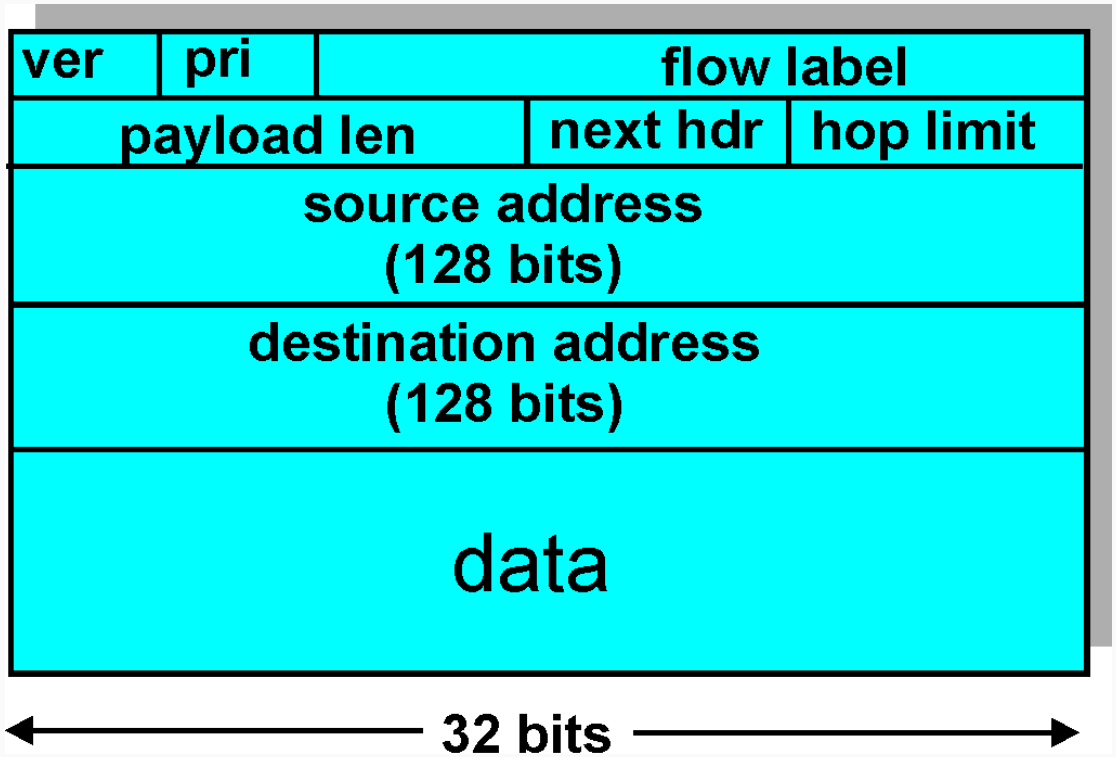
IPv6

IPv6

- Address shortage should instead be solved by IPv6
- Expands address space without using NAT
- Redesign protocol
- What changes should be made in ...
  - IP addressing
  - IP delivery semantics
  - IP quality of service
  - IP security
  - IP routing
  - IP fragmentation
  - IP error detection

Main IPv6 Changes

- Addresses are 128 bit
- Protocol simplified (end-to-end principle)
  - Removes checksum
  - Eliminates fragmentation



Parsing an IPv6 address

- Specified as 8, 2-byte (4 hex digit) numbers
  `2610:10:20:220:45c7:8fb6:7430:bcb6`
  - Note, leading 0s omitted for brevity
- Double-colon notation
  - Can be used exactly once in an address to specify a wildcard of all 0s in address
  - Fills address with enough nulls to create a 128-bit address
  - Example:
    - `2610:10:20:1103::22`
    - `2610:10:20:1103:0:0:0:22`

Transition From IPv4 To IPv6

- Gradual deployment
  - Eventually, want to have all devices run dual stacks
- But, what happens when you run into a cloud of IPv4-only routers?

Transition From IPv4 To IPv6

- Gradual deployment
  - Eventually, want to have all devices run dual stacks
- But, what happens when you run into a cloud of IPv4-only routers?
- Tunneling
  - IPv6 carried as payload in an IPv4 datagram among IPv4 routers

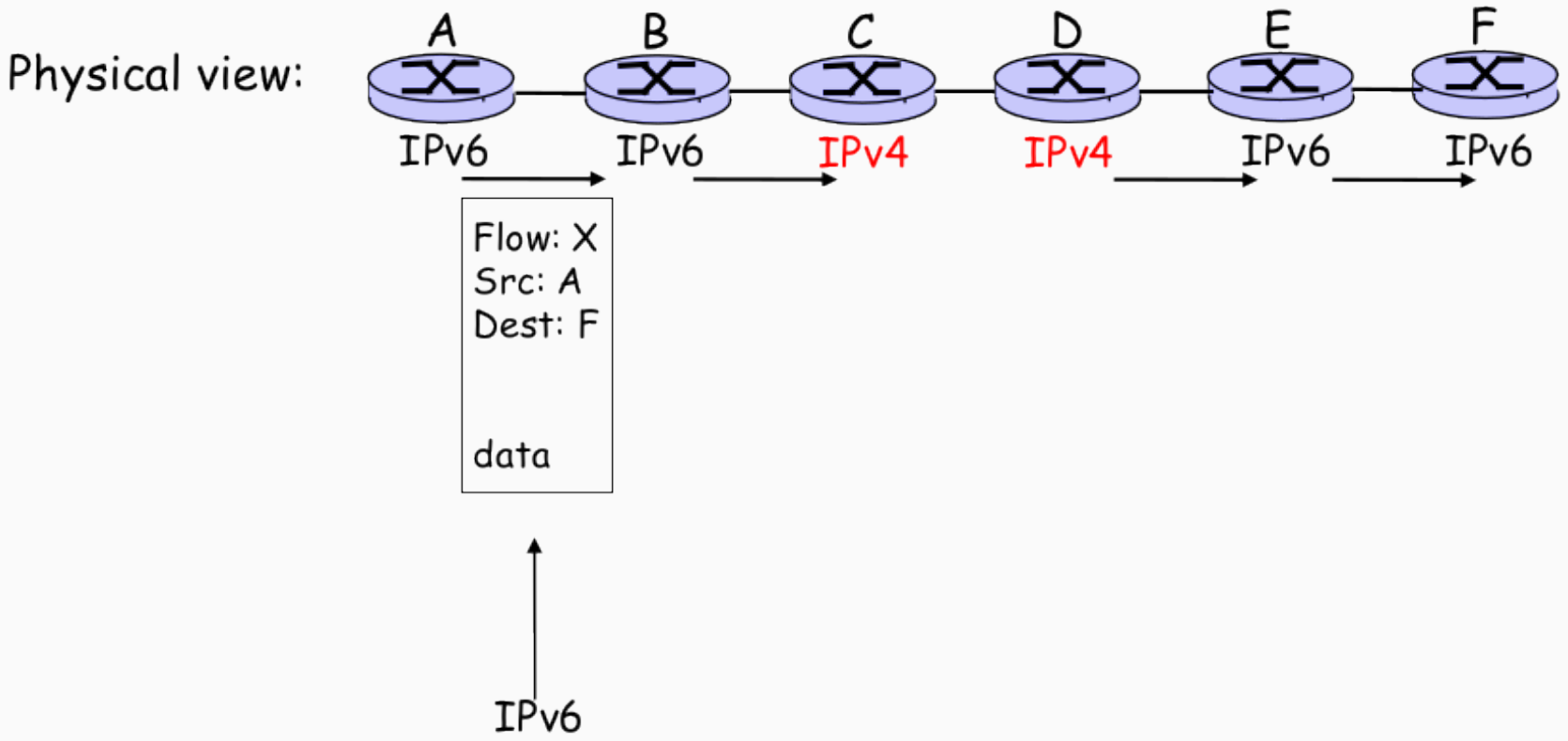
Transition From IPv4 To IPv6

- Gradual deployment
  - Eventually, want to have all devices run dual stacks
- But, what happens when you run into a cloud of IPv4-only routers?
- Tunneling
  - IPv6 carried as payload in an IPv4 datagram among IPv4 routers
  - Wraps an IPv4 Russian doll around the IPv6 one
    - For IPv6, treats the entire IPv4 network as a single data-link!
    - e.g. IPv4 is a framing protocol for the IPv4 "data-link" layer

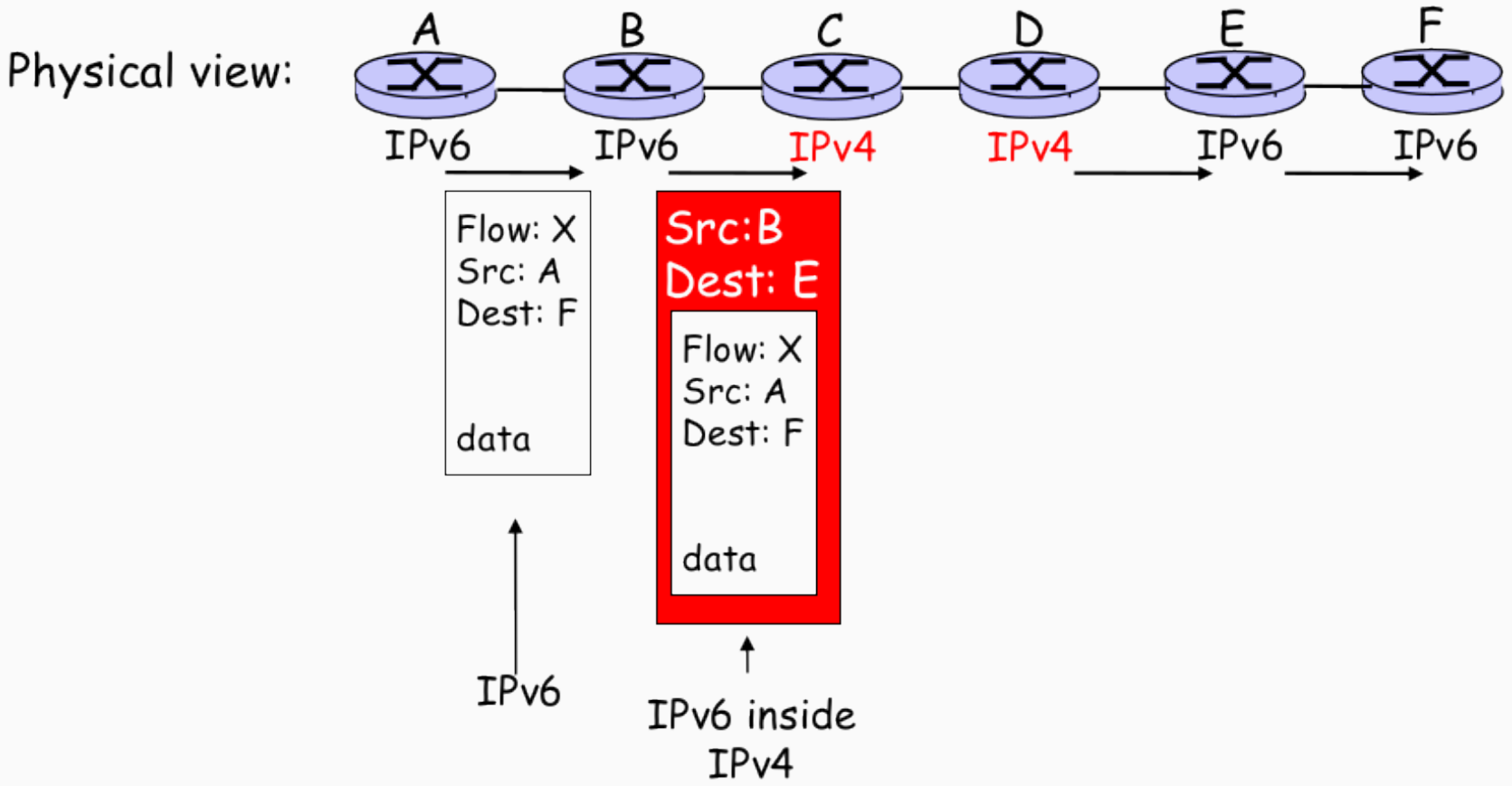
Transition From IPv4 To IPv6

- Gradual deployment
  - Eventually, want to have all devices run dual stacks
- But, what happens when you run into a cloud of IPv4-only routers?
- Tunneling
  - IPv6 carried as payload in an IPv4 datagram among IPv4 routers
  - Wraps an IPv4 Russian doll around the IPv6 one
    - For IPv6, treats the entire IPv4 network as a single data-link!
    - e.g. IPv4 is a framing protocol for the IPv4 "data-link" layer
  - Builds a virtual IPv6 network link on top of a multi-hop IPv4 network

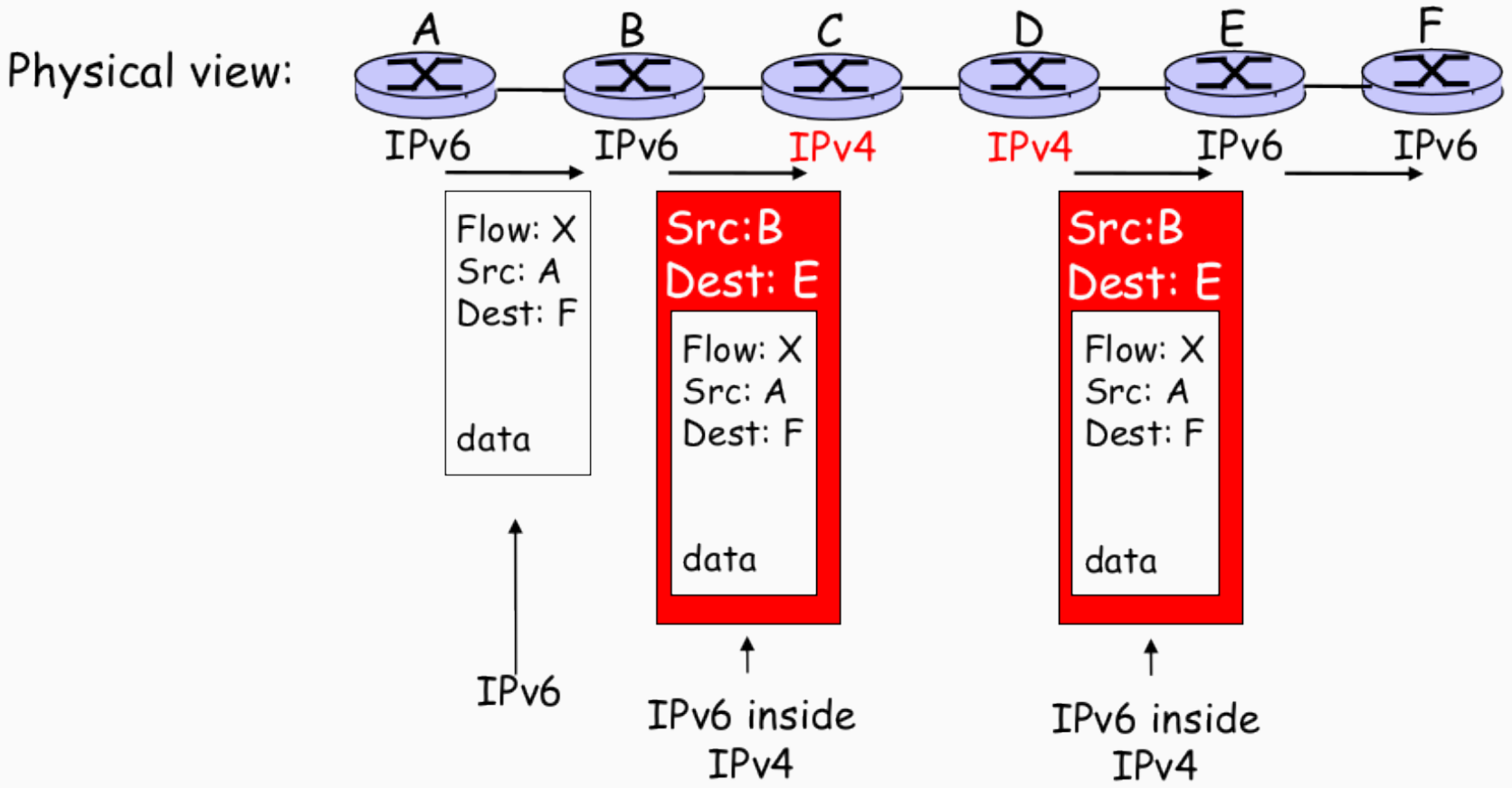
Tunneling


Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
IPv6

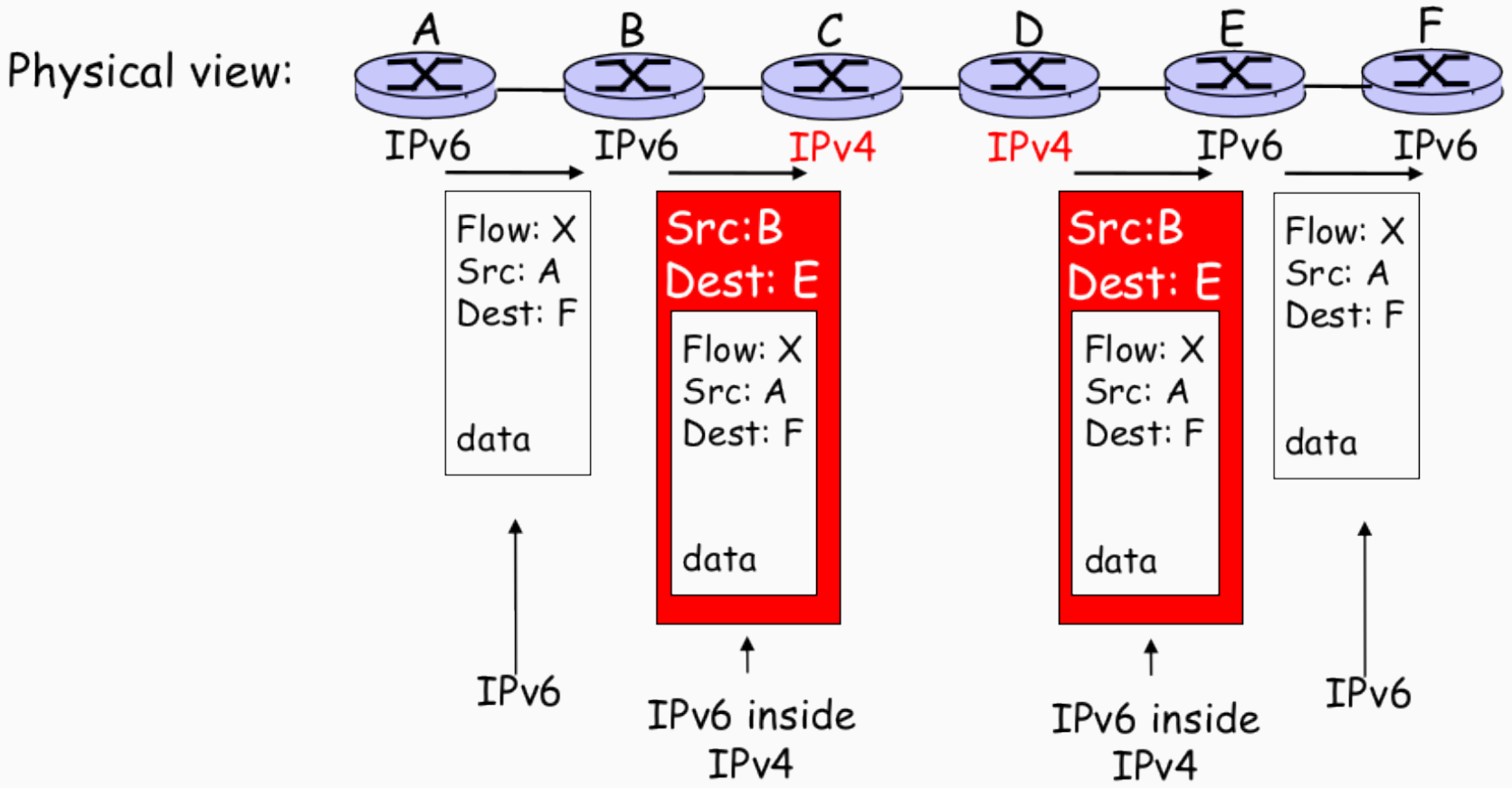
Tunneling


Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
IPv6
IPv6 inside
IPv4

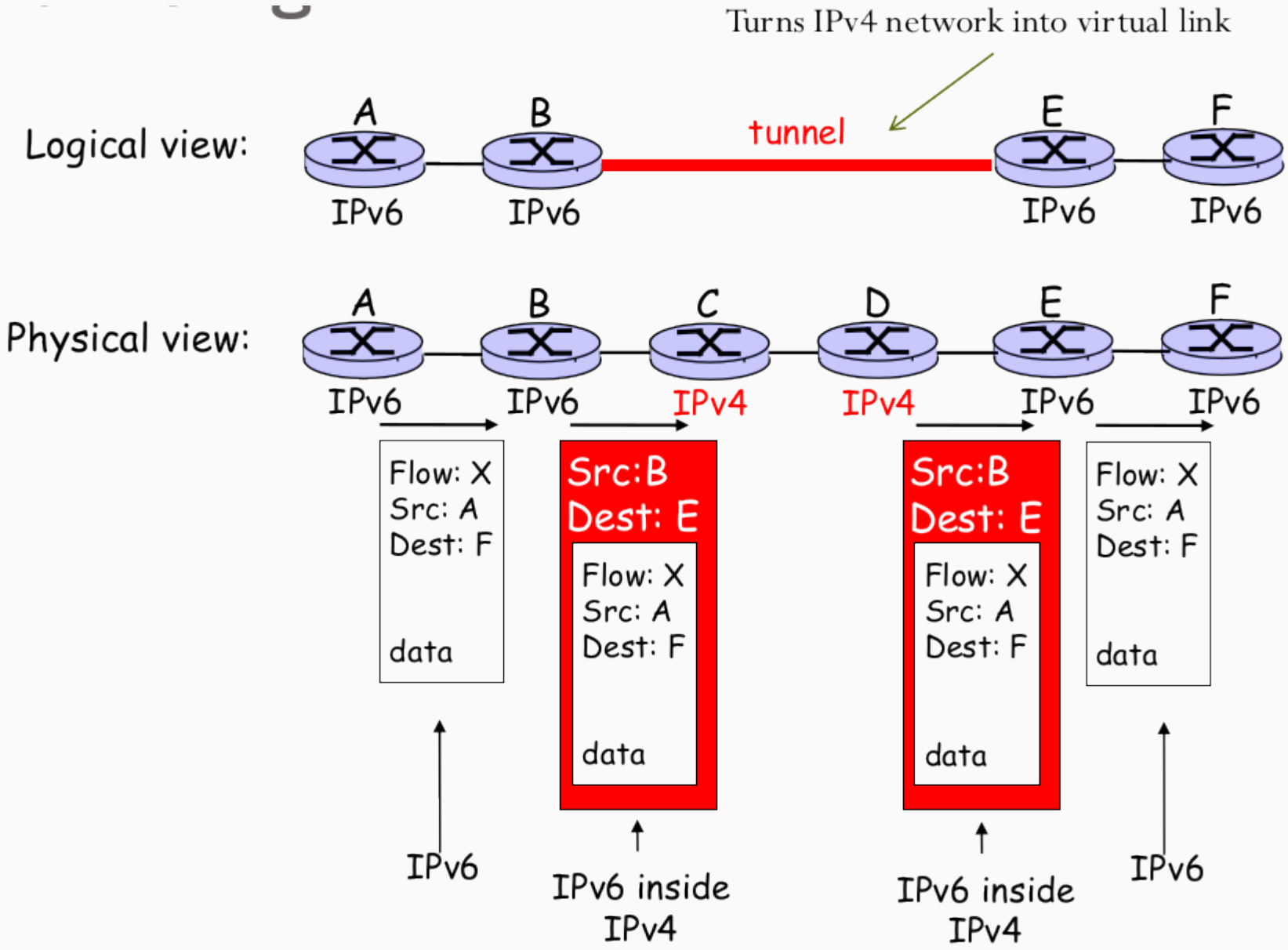
Tunneling


Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
IPv6
IPv6 inside
IPv4
IPv6 inside
IPv4

Tunneling


Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Flow: X
Src: A
Dest: F
data
IPv6
IPv6 inside
IPv4
IPv6 inside
IPv4
IPv6

Tunneling


Turns IPv4 network into virtual link
Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Flow: X
Src: A
Dest: F
data
IPv6
IPv6 inside
IPv4
IPv6 inside
IPv4
IPv6