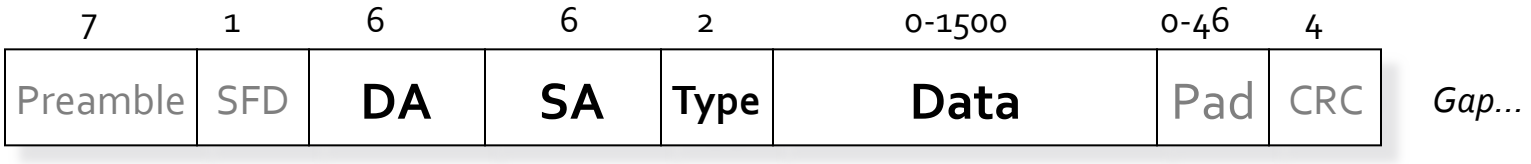
ELEC / COMP 177 – Fall 2014

Computer Networking → Global Communication

Some slides from Kurose and Ross, *Computer Networking*, 5th Edition

Recap – Ethernet Frame



- Destination MAC address
- Source MAC address
- Type (of encapsulated data)
- The data!
- **Who assigns the source address?**
  - **Does it contain information on network location?**
- **If I just have an Ethernet frame, where can I send data to?**

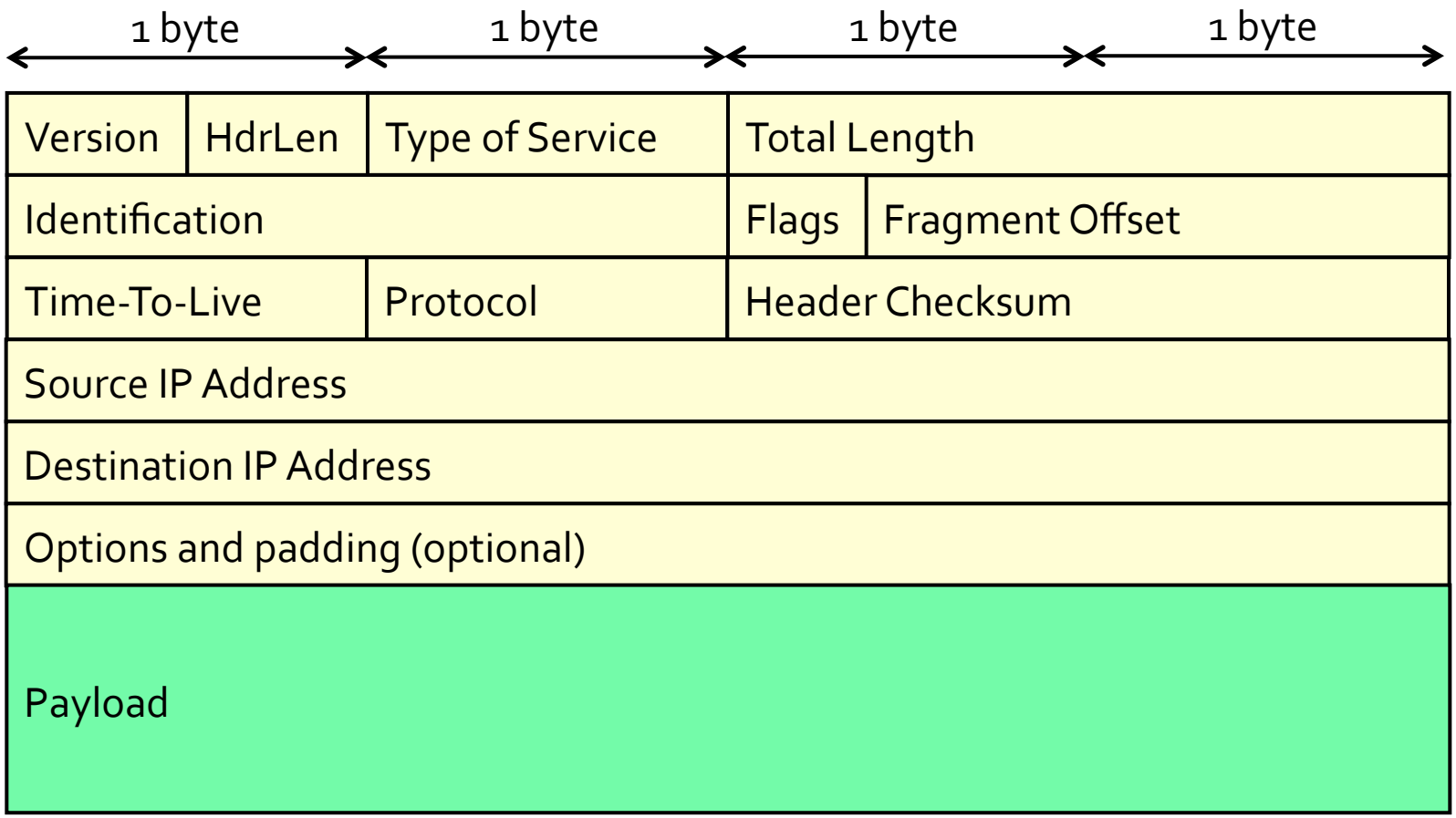
Recap – Ethernet Switch

- How does a switch learn the location of computers on the network? (what *field*)
- What is stored in the forwarding table?
  - MAC address, output port
- What happens if a switch has no match in its forwarding table?

Recap – Ethernet

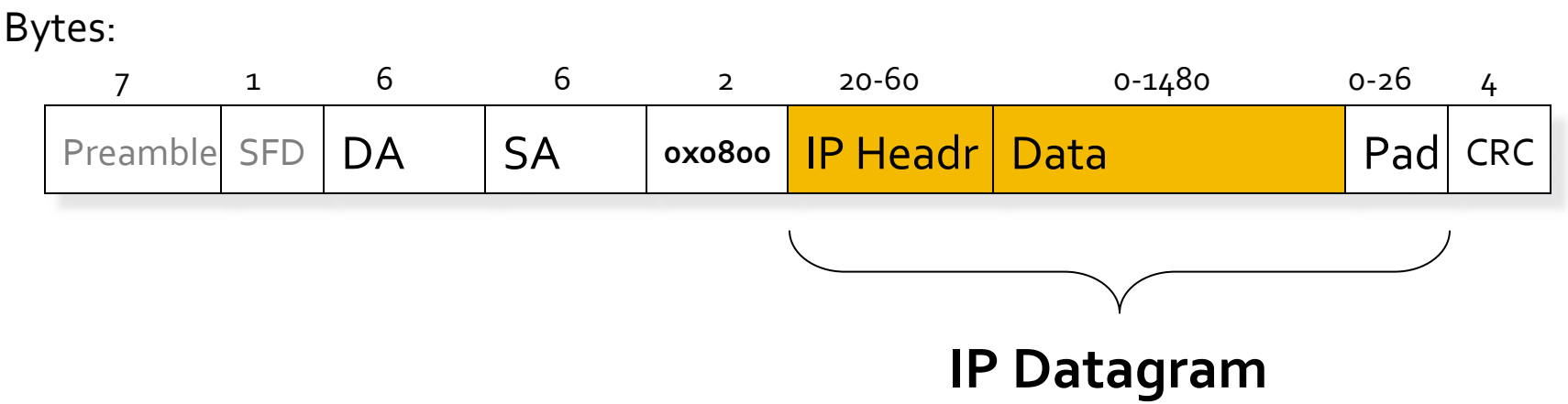
- **Why can't we use Ethernet for global communication?**
  - Broadcasts to find location of computers – too much bandwidth to do worldwide
  - Loops – Ethernet uses spanning tree to prevent loops
    - Can't have a single "root" of the Internet!
  - **Address contains no information about location on network**
    - Would need to have a forwarding table with one entry for every PC on the Internet we want to communicate with
    - i.e. a single worldwide "phonebook" with no shortcuts!

Recap – IP Datagram



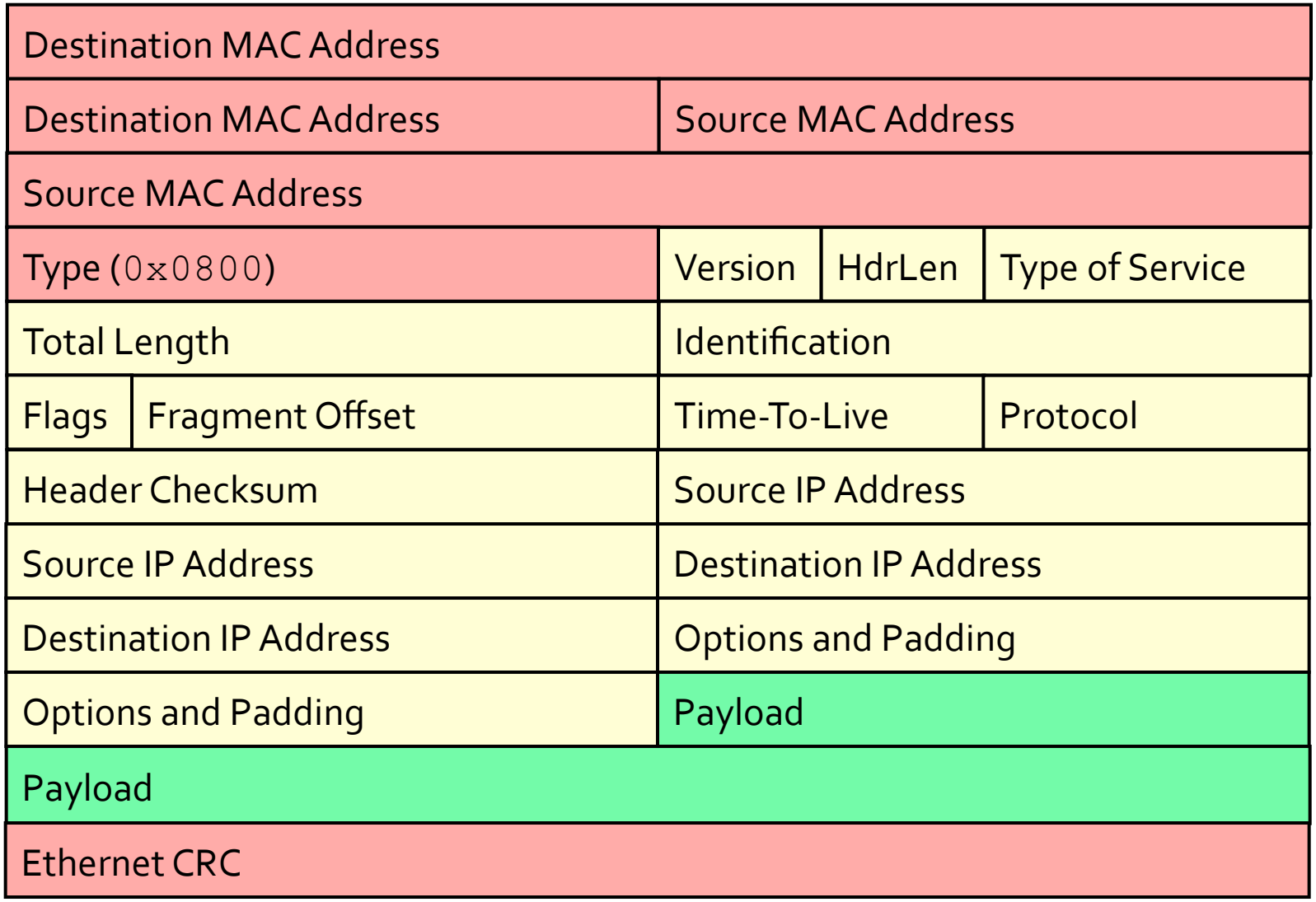
Recap – IP Datagram

- Are IP packets separate from Ethernet frames?



- Time-to-live field: what's it used for?

Recap – IP encapsulated in Ethernet



Recap – IP Datagram

- **Where does the source IP address come from?**
  - DHCP (possibly running on the router)
- **Where does the destination IP address come from?**
  - DNS can be used to translate a host name from the *user* (e.g. www.pacific.edu) into an IP address (e.g. `138.9.110.12`)

Recap – IP Routers

- Ethernet switches forward packets based on destination MAC address
- **What do routers forward packets based on?**
  - Destination IP address
- **What is in the router's forwarding table?**
  - Prefixes, e.g. `138.16.9/24`
  - Next hop IP
  - Exit port
- **What happens if more than one prefix matches the destination IP address?**
  - **Longest prefix match** determines winner

Recap – Forwarding versus Routing

FORWARDING

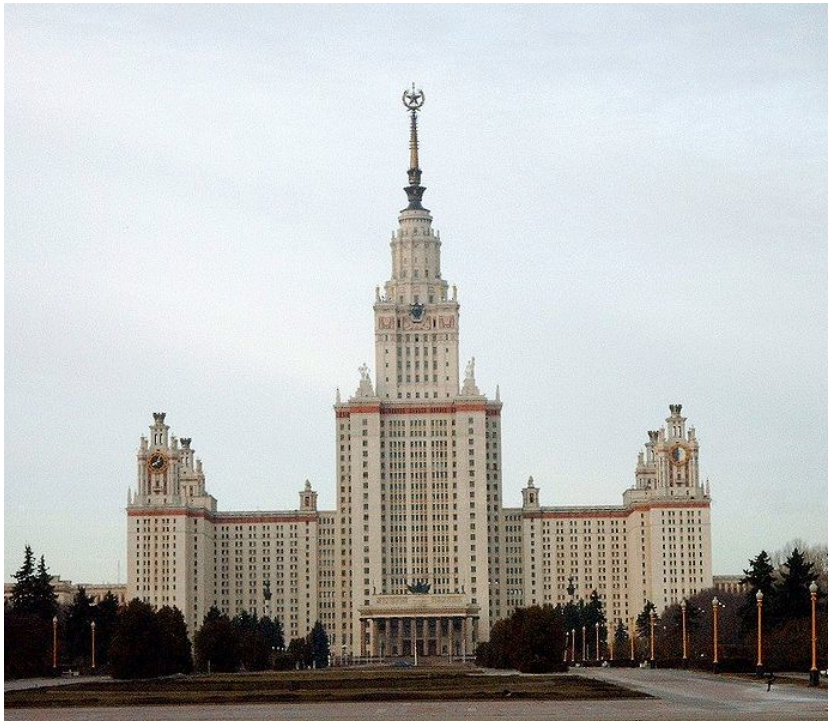
- Move packets from router's input to appropriate router output
- *Longest prefix match* (LPM)

ROUTING

- Determine path (route) taken by packets from source to destination
- Routing algorithms such as RIP and OSPF

Example

- Send a single IP packet from Pacific to the main Moscow State University web server
- My IP:
  - `138.9.253.252`
- MSU's IP:
  - `93.180.0.18`



Traceroute

How does this actually **work**?

```
dhcp-10-6-162-134:~ shafer$ traceroute -q 1 www.msu.ru
traceroute to www.msu.ru (93.180.0.18), 64 hops max, 52 byte packets
 1  10.6.163.254 (10.6.163.254)  1.677 ms
 2  10.0.0.141 (10.0.0.141)  1.116 ms
 3  10.0.0.90 (10.0.0.90)  1.053 ms
 4  138.9.253.252 (138.9.253.252)  5.200 ms
 5  74.202.6.5 (74.202.6.5)  8.137 ms
 6  pao1-pr1-xe-1-2-0-0.us.twtelecom.net (66.192.242.70)  13.241 ms
 7  te-9-4.car1.sanjose2.level3.net (4.59.0.229)  92.772 ms
 8  vlan70.csw2.sanjose1.level3.net (4.69.152.126)  8.440 ms
 9  ae-71-71.ebr1.sanjose1.level3.net (4.69.153.5)  11.130 ms
10  ae-2-2.ebr2.newyork1.level3.net (4.69.135.186)  80.992 ms
11  ae-82-82.csw3.newyork1.level3.net (4.69.148.42)  77.316 ms
12  ae-61-61.ebr1.newyork1.level3.net (4.69.134.65)  74.584 ms
13  ae-41-41.ebr2.london1.level3.net (4.69.137.65)  147.127 ms
14  ae-48-48.ebr2.amsterdam1.level3.net (4.69.143.81)  151.779 ms
15  ae-1-100.ebr1.amsterdam1.level3.net (4.69.141.169)  152.848 ms
16  ae-48-48.ebr2.dusseldorf1.level3.net (4.69.143.210)  156.349 ms
17  4.69.200.174 (4.69.200.174)  168.386 ms
18  ae-1-100.ebr1.berlin1.level3.net (4.69.148.205)  167.652 ms
19  ae-4-9.bar1.stockholm1.level3.net (4.69.200.253)  192.668 ms
20  213.242.110.198 (213.242.110.198)  176.501 ms
21  b57-1-gw.spb.runnet.ru (194.85.40.129)  198.827 ms
22  m9-1-gw.msk.runnet.ru (194.85.40.133)  204.276 ms
23  msu.msk.runnet.ru (194.190.254.118)  202.454 ms
24  93.180.0.158 (93.180.0.158)  201.358 ms
25  93.180.0.170 (93.180.0.170)  200.257 ms
26  www.msu.ru (93.180.0.18)  204.045 ms !Z
```

Companies Handling Our Packet

Number	Name
1)	University of the Pacific
2)	Time Warner Telecom
3)	Level 3 Communications
4)	Runnet - State Institute of Information Technologies & Telecommunications (SIIT&T "Informika")
5)	Moscow State University

Assumptions

- Assume that I know
  - My own MAC address (hardwired on the NIC)
  - My own IP address (assigned via DHCP to be within my local subnet)
  - The subnet mask for my local network
  - The IP address of my gateway router leading "outside"
  - The IP address of MSU that I want to send a message to

Step 1

- **What happens first?**
  - Compare destination IP with my IP and subnet mask
    - My IP: `138.9.110.104`
    - My subnet mask: `255.255.255.0`
    - Thus, my subnet is `138.9.110/24`
  - Destination IP of `93.180.0.18` is (way!) outside my LAN

Step 2

- **The destination is outside of my LAN. What happens next?**
  - Need to send packet to gateway router
- **What does the Ethernet/IP packet look like?**
  - Destination MAC: ???
  - Source MAC: My MAC
  - Destination IP: MSU's IP
  - Source IP: My IP
  - TTL: 64 (a reasonable default)

Step 3

- **How do I get the MAC address of the router port attached to my LAN?**
  - I know my gateway router's IP address
  - Use ARP (Address Resolution Protocol)
- **Who receives my ARP request?**
  - Everyone – broadcast to all hosts on LAN
  - *"Who has 138.16.110.1? Tell 138.9.110.104"*
- **Who replies to my ARP request?**
  - Only the host (if any) with the requested IP address. This should be the router

Step 4

- Assume there is an Ethernet switch between you and the router
- **What happens if the switch has seen the MAC address of the router before?**
  - Packet is sent out only the port that faces the router
- **What happens if the switch has *not* seen the MAC address before?**
  - Packet is broadcast out all ports
- Switch **always** learns (or re-learns) from each packet

Step 5

- The packet reaches your gateway router (first router between here and MSU)
- **What does the router do?**
  - Verify checksums
  - Longest prefix match on destination IP address
- **What information is returned from router's forwarding table?**
  - Next hop IP address
    - (of subsequent router, or final host)
  - Output port

Step 6

- Assume the next hop is also connected to this router via Ethernet
- **What do we need to know to send a message to this router?**
  - Its MAC address
- **How do we find this?**
  - Router does ARP (just like hosts do ARP)

Step 7

- **How does the router modify the packet when retransmitting?**
  - Destination MAC = *change* to be MAC of next hop
  - Source MAC = *change* to be MAC of this router
  - Destination IP = unchanged
  - Source IP = unchanged
  - TTL = *decrement* by 1
  - Checksum = *recalculate*

Step 8

- This process of re-transmitting a packet repeats for many routers across the network
  - *26 in this example*
- Eventually, however, the "next hop" in the forwarding table is the actual destination computer
  - Packet has arrived!
- **Is that all the complexity in the Internet?**
  - **No – forwarding tables in the router aren't created by magic!**

Routing

- In addition to forwarding packets, routers are busy (*asynchronously*) calculating **least-cost** routes to destinations
  - Goal: Have the forwarding table ready by the time your packet arrives with a specific destination
- **What happens if the forwarding table isn't ready, and there is no entry for your destination?**
  - Packet is dropped – you lose

Hierarchical Routing

- Our routing discussion thus far has been idealized
  - All routers are identical
  - The network is "flat"
- This is not true in practice!

- Problem 1 – **Scale**
  - Hundreds of millions of destinations:
  - Can't store all destinations in routing tables!
  - Routing table exchange would swamp links!
  - Distance-vector would never converge
- Problem 2 - **Administrative autonomy**
  - Internet = network of networks
  - Each network admin wants to control routing in his/her own network

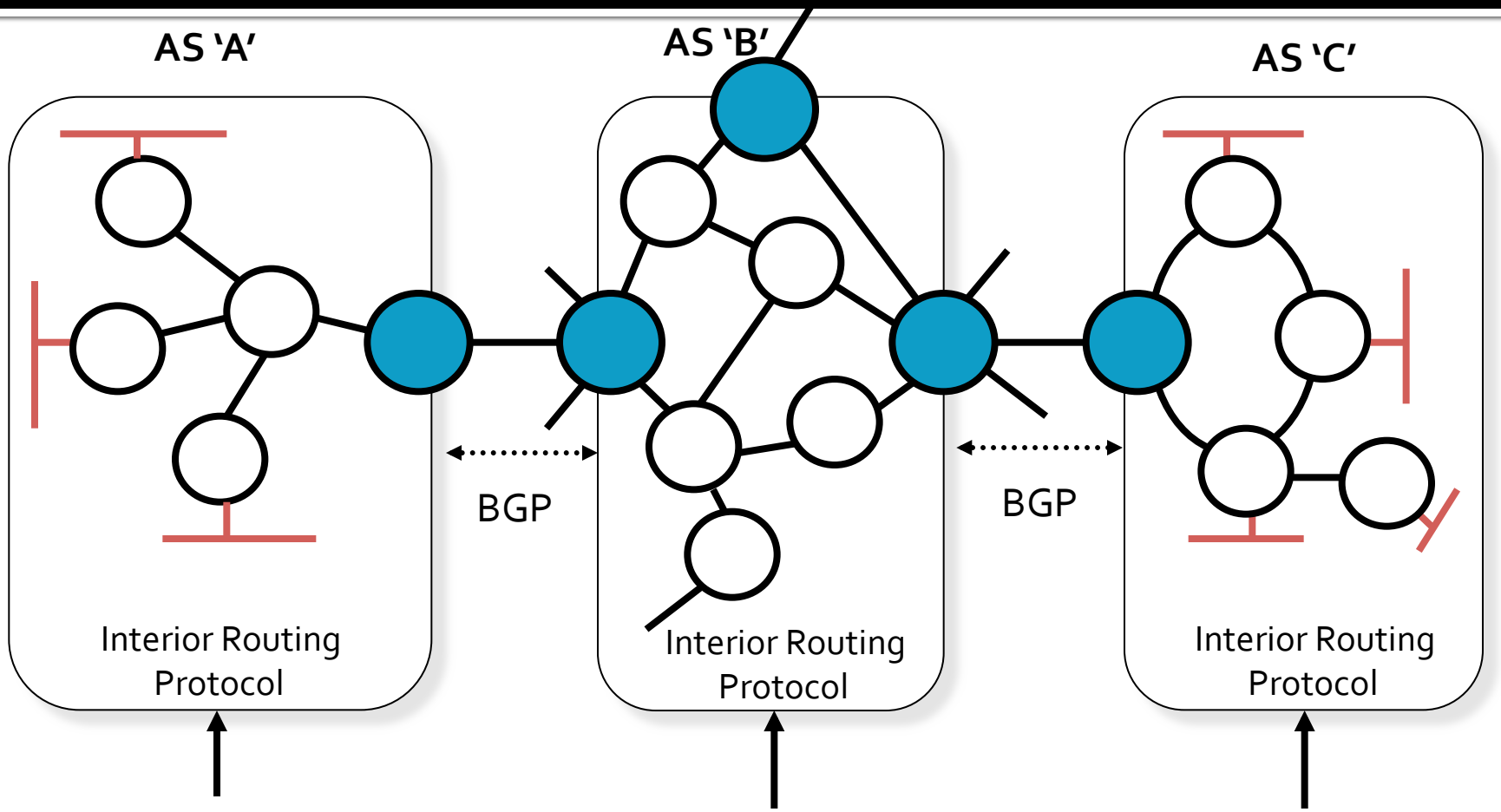
Hierarchical Routing

- Aggregate routers into regions (aka "**autonomous systems**" - AS)
- Routers inside autonomous system run same routing protocol
  - "Intra-AS" routing protocol
  - Routers in different AS can run different intra-AS routing protocol
- Border Router
  - Direct link to router in another AS

Routing in the Internet

- The Internet uses hierarchical routing
- The Internet is split into Autonomous Systems
  - "Independent" networks on the Internet
  - Typically owned/controlled by a single entity
  - Share a common routing policy
- Example autonomous systems
  - Pacific (18663), Exxon (1766), IBM (16807), Level3 (3356)
- Different routing protocols within and between autonomous systems
  - Interior gateway/routing protocol (e.g. OSPF)
  - Border gateway protocol (e.g. BGP)

Autonomous Systems



Traceroute

```
dhcp-10-6-162-134:~ shafer$ traceroute -a -q 1 www.msu.ru
traceroute to www.msu.ru (93.180.0.18), 64 hops max, 52 byte packets
 1  [AS65534] 10.6.163.254 (10.6.163.254)  1.677 ms
 2  [AS1] 10.0.0.141 (10.0.0.141)  1.116 ms
 3  [AS1] 10.0.0.90 (10.0.0.90)  1.053 ms
 4  [AS0] 138.9.253.252 (138.9.253.252)  5.200 ms
 5  [AS0] 74.202.6.5 (74.202.6.5)  8.137 ms
 6  [AS4323] pao1-pr1-xe-1-2-0-0.us.twtelecom.net (66.192.242.70)  13.241 ms
 7  [AS3356] te-9-4.car1.sanjose2.level3.net (4.59.0.229)  92.772 ms
 8  [AS3356] vlan70.csw2.sanjose1.level3.net (4.69.152.126)  8.440 ms
 9  [AS3356] ae-71-71.ebr1.sanjose1.level3.net (4.69.153.5)  11.130 ms
10  [AS3356] ae-2-2.ebr2.newyork1.level3.net (4.69.135.186)  80.992 ms
11  [AS3356] ae-82-82.csw3.newyork1.level3.net (4.69.148.42)  77.316 ms
12  [AS3356] ae-61-61.ebr1.newyork1.level3.net (4.69.134.65)  74.584 ms
13  [AS3356] ae-41-41.ebr2.london1.level3.net (4.69.137.65)  147.127 ms
14  [AS3356] ae-48-48.ebr2.amsterdam1.level3.net (4.69.143.81)  151.779 ms
15  [AS3356] ae-1-100.ebr1.amsterdam1.level3.net (4.69.141.169)  152.848 ms
16  [AS3356] ae-48-48.ebr2.dusseldorf1.level3.net (4.69.143.210)  156.349 ms
17  [AS3356] 4.69.200.174 (4.69.200.174)  168.386 ms
18  [AS3356] ae-1-100.ebr1.berlin1.level3.net (4.69.148.205)  167.652 ms
19  [AS3356] ae-4-9.bar1.stockholm1.level3.net (4.69.200.253)  192.668 ms
20  [AS3356] 213.242.110.198 (213.242.110.198)  176.501 ms
21  [AS3267] b57-1-gw.spb.runnet.ru (194.85.40.129)  198.827 ms
22  [AS3267] m9-1-gw.msk.runnet.ru (194.85.40.133)  204.276 ms
23  [AS3267] msu.msk.runnet.ru (194.190.254.118)  202.454 ms
24  [AS2848] 93.180.0.158 (93.180.0.158)  201.358 ms
25  [AS2848] 93.180.0.170 (93.180.0.170)  200.257 ms
26  [AS2848] www.msu.ru (93.180.0.18)  204.045 ms !Z
```

AS Numbers in Traceroute

AS	Name
0	Reserved (local use)
18663	University of the Pacific *(Traceroute didn't resolve this due to missing information in address registry…)*
4323	Time Warner Telecom
3356	Level 3 Communications
3267	Runnet - State Institute of Information Technologies & Telecommunications (SIIT&T "Informika")
2848	Moscow State University

First AS

- First AS is Pacific's (AS18663)
- Do a lookup on the AS
  - http://www.ripe.net/data-tools/stats/ris/routing-information-service
  - https://www.dan.me.uk/bgplookup
  - http://www.peeringdb.com/
    - Among other places…
- Pacific's gateway(s) to the Internet advertise a BGP prefix (aka subnet)
  - `138.9.0.0/16`

First AS

- An advertisement is a *promise:*
  - If you give me packets destined for IP addresses in this range, I will move them closer to their destination.
  - In this case, we *are* the destination!
  - This advertisement *originates* from our AS

Second AS

- Pacific buys Internet service from Time Warner (AS4323) , which has border routers that speak BGP
  - Pacific's routers talk to their routers, and they learn of our advertisement for 138.9.0.0/16
  - Now, Time Warner knows how to reach Pacific's IPs
  - We also learn of their advertisements!
    - Both for prefixes *originating* at those ISPs, and prefixes *reachable* through those ISPs

Announcements

- **When Time Warner give our routers their BGP announcements, do we get lots of tiny entries like 138.9.0.0/16?**
  - Maybe
  - But, routes can be aggregated together and expressed with smaller prefixes, e.g. `138.0.0.0/8`
    - Reduces communication time plus router CPU and memory requirements

Second AS (continued)

- Pacific had only 1 announcement
- Time Warner *originates* ~159 announcements (as of Nov 2012)
  - Some are large, e.g. `173.226.0.0/15`
  - Some are small, e.g. `159.157.233.0/24`
- Time Warner also provides transit to their *downstream* customers' prefixes, including Pacific's prefix
  - Total of ~6395 announcements (as of Nov 2012)
  - We get this full list, as does ever other (BGP-speaking) AS connected to Time Warner

Third AS

- Time Warner (AS4323) can move this packet to San Jose, where it enters the Equinix Internet Exchange (See https://www.peeringdb.com)
  - Private location to peer ("exchange traffic") with dozens of other companies
  - Akamai, Apple, Amazon, Facebook, Google, Microsoft, many ISPs, etc…
- Time Warner connects with Level 3 (AS3356)
  - *Do they pay, or is this free?*
  - Same sharing of BGP announcements occurs here

Last AS

- **The same thing is happening over in Eurasia**
- Last AS of our path is Moscow State University (AS2848)
- MSU's gateway(s) to the Internet advertise a BGP prefix for `93.180.0.0/18` (along with 3 others that *originate* in this AS)
  - That encompasses the destination IP of `93.180.0.18`

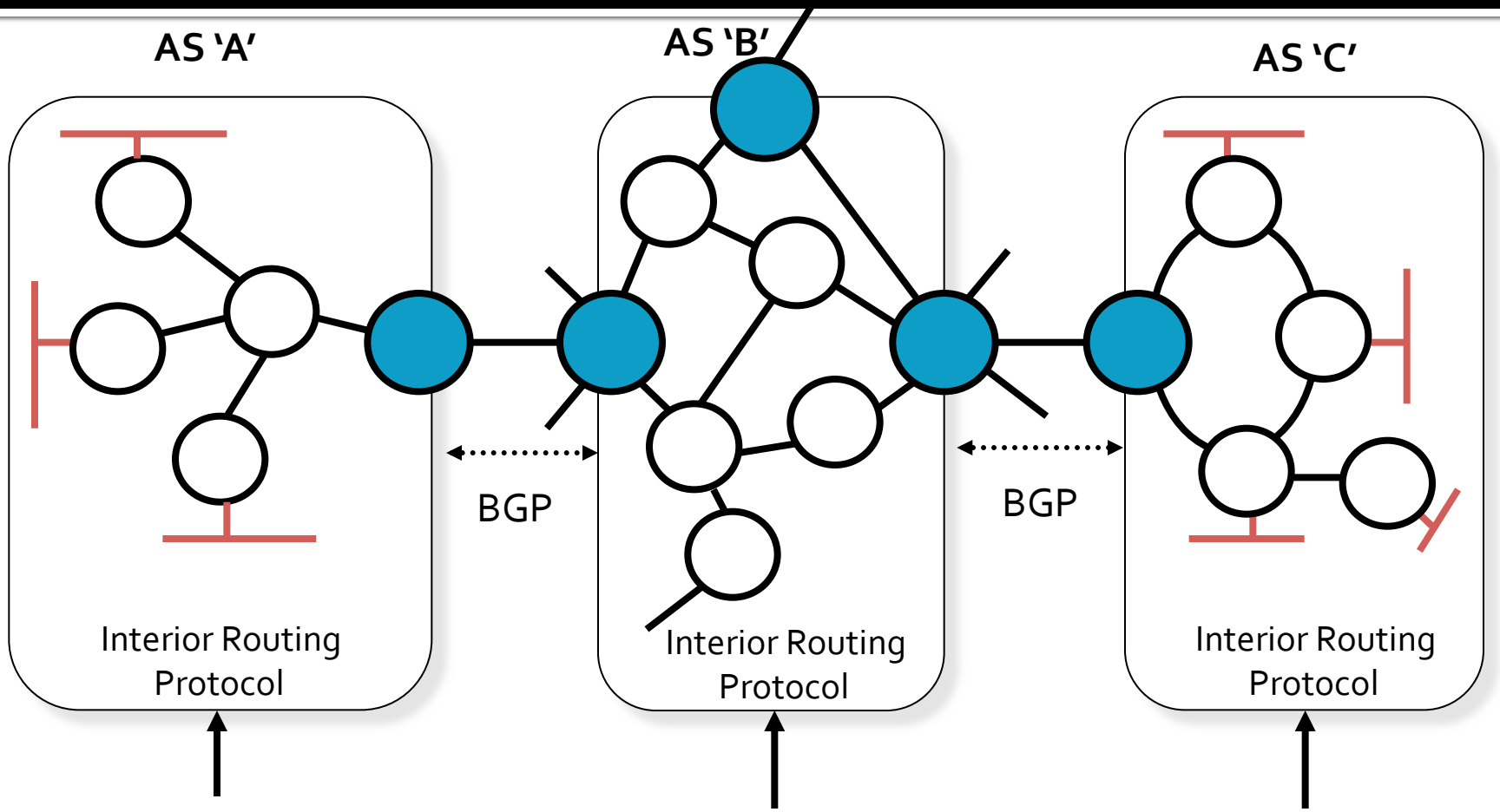
Next-to-last AS

- Moscow State University connects to Runnet (AS3267)
  - Runnet announces prefix `93.180.0.0/18` (along with 291 others reachable *downstream*, and 13 that *originate* in this AS)
  - Runnet now knows how to reach IPs that belong to MSU
- Runnet obtains transit through Level3, so our link is complete!

What's Missing?

- The forwarding table!
  - We keep forgetting to generate the forwarding table!
- Need more information
  - BGP tells us links between autonomous systems
  - Other protocols (RIP, OSPF) tell us paths within autonomous systems

Autonomous Systems



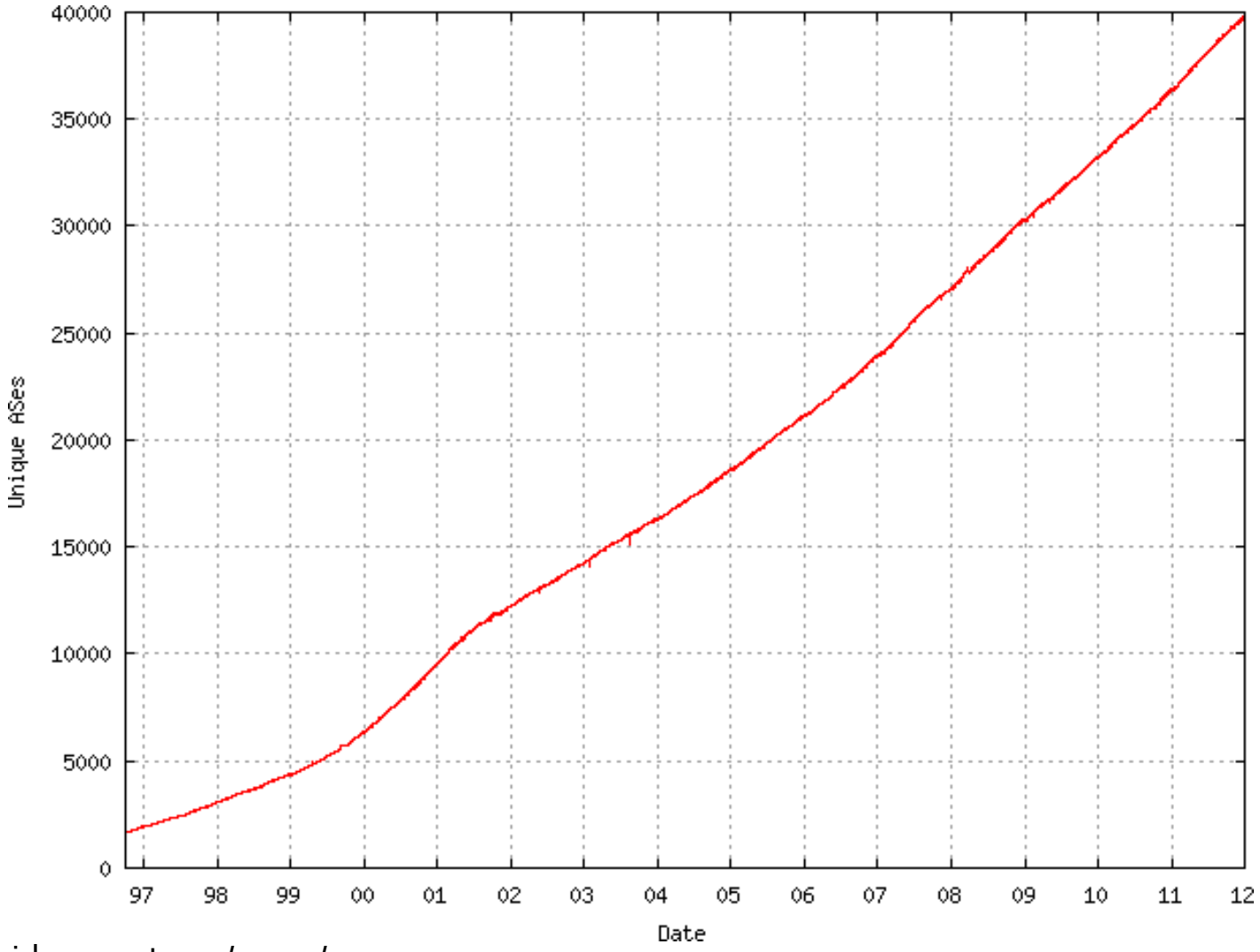
Interior Routing Protocol

- **Option 1: Global Information** (example: OSPF)
  - All routers have complete topology, link cost info
  - **"Link state" algorithms** *(Dijkstra's algorithm)*
- **Option 2: Decentralized** (example: RIP)
  - Router only knows physically-connected neighbors and link costs to neighbors
  - Iterative process of computation, exchange of info with neighbors
  - **"Distance vector" algorithms** *(Bellman-Ford Algorithm)*

Interior Routing Protocol

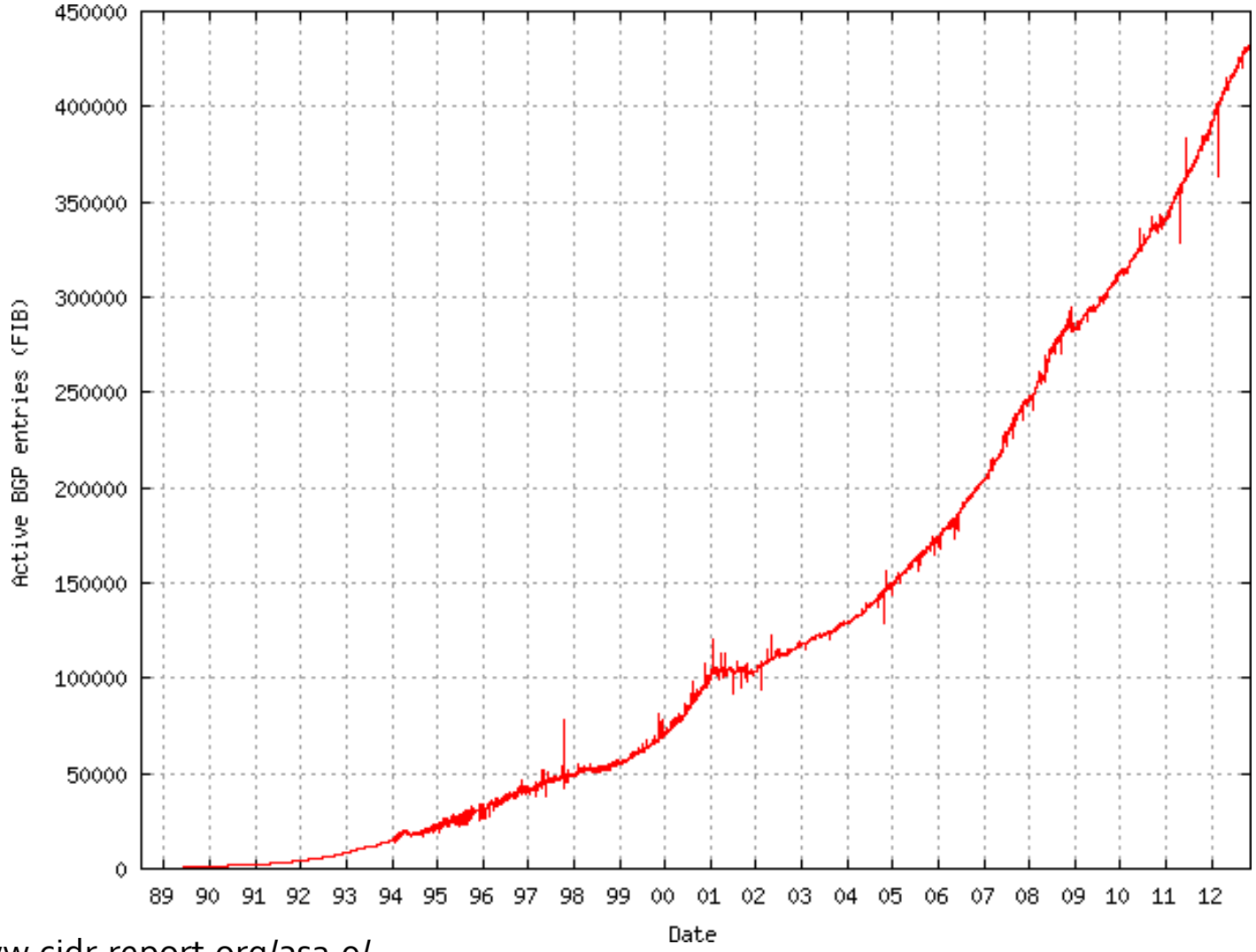
- Each router inside the AS updates its own forwarding table to direct BGP prefixes to the appropriate gateway router to the next AS
  - Rules might be very simple, i.e. just forward everything not destined to this AS to the same gateway router
  - Or rules might be complicated…
- **End result is a forwarding table for the router**
  - Prefix (for LPM)
  - Next-hop IP
  - Exit port

Growth of Internet – AS's



http://www.cidr-report.org/as2.0/

Growth of Internet – BGP Entries (prefixes)



http://www.cidr-report.org/as2.0/

Growth

- **What does this growth mean for routers on the BGP-speaking Internet?**
  - They need to grow too! (more memory, faster CPUs, etc…)

What Does *My* Computer Do?

- **Does *my* computer speak BGP?**
  - No – your ISP's external gateway router does
- **Does *my* computer speak RIP or OSPF?**
  - No – your ISP's internal routers do
- **Does *my* computer speak ARP?**
  - Yes
- **Does *my* computer speak IP?**
  - Yes
- **Does *my* computer speak Ethernet?**
  - Yes

Milestone

- **Successfully sent a single IP packet across the global Internet**
  - Now know all of the key protocols and standards necessary to accomplish that task
- **Now can I waste time watching LOLcats?**

Milestone

- Not quite. One IP packet by itself is not enough to transmit an entire image
- **What else do we need?**
  - Method to **link multiple IP packets together** and deliver them to the **correct process** on the receiver
    - **Transport layer**: UDP, TCP (TCP also provides **reliability!**)
  - **Applications** need to be written to use this reliable network communication, and they need protocols of their own!
    - Web = HTTP, Email = POP / IMAP / SMTP, …