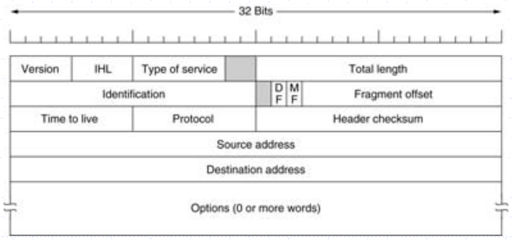
ECE453 – Introduction to Computer Networks

Lecture 12 – Network Layer (IV)

IP Datagram Format



An Example (tcpdump)

```
4500 0054 0000 4000 3401 eb82 982d 0469 a024 1e6c 0800 57a3 ce1b 0000
|||  |    |    ||   | |  |    |         |         | |
|||  |    |    ||   | |  |    32-bit source IP: 152.45.4.105
|||  |    |    ||   | |  Header checksum (16 bits)
|||  |    |    ||   | Upper layer protocol: 01
|||  |    |    ||   TTL
|||  |    |    |13-bit fragmentation offset
|||  |    |    3-bit flag
|||  |    identifier
|||  datagram length:
||type of service (1 byte)
||header:4*5=20bytes (4 bits)
V4  (4 bits)
```

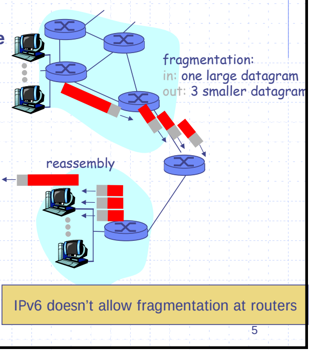
Analyze Network: `tcpdump`

- You need root privilege
- -i : listen to a specific interface, e.g., eth0
- -w: write the raw packet to a file rather than print them out, -r can be used to read packet from a file
- -s num: get num bytes of data from each packet rather than the default value of 68
- src host xyz: true if IP destination field of the packet is xyz
- dst host xyz: true if IP source field of the packet is xyz
- ip proto xyz: true if the packet is an IP packet and protocol type is xyz
- Example: /usr/sbin/tcpdump src host 152.45.4.11 and icmp

4

IP Fragmentation & Reassembly

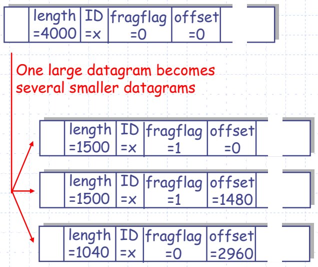
- **Link-layer protocols can only carry packets of a limited size**
- **Different link-layer protocols may carry packets of different size**
  - Ethernet: 1,500 bytes
  - Others: 576 bytes
- **MTU: maximum transfer unit**
- large IP datagram divided (“fragmented”) within net
  - one datagram becomes several datagrams
  - “reassembled” only at final destination
  - IP header bits used to identify, order related fragments



IPv6 doesn’t allow fragmentation at routers

5

IP Fragmentation and Reassembly



length =4000 | ID =x | fragflag =0 | offset =0

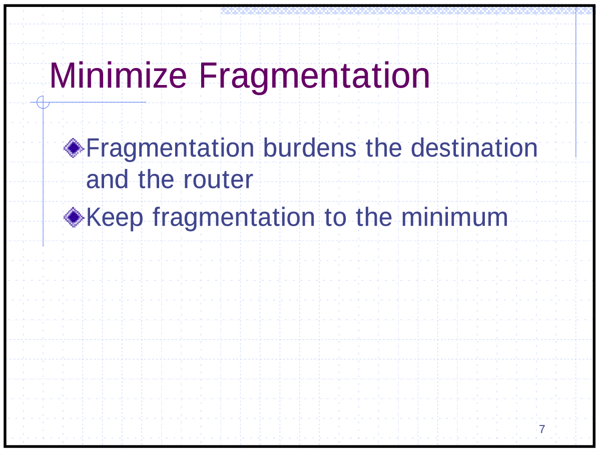
One large datagram becomes several smaller datagrams

length =1500 | ID =x | fragflag =1 | offset =0

length =1500 | ID =x | fragflag =1 | offset =1480

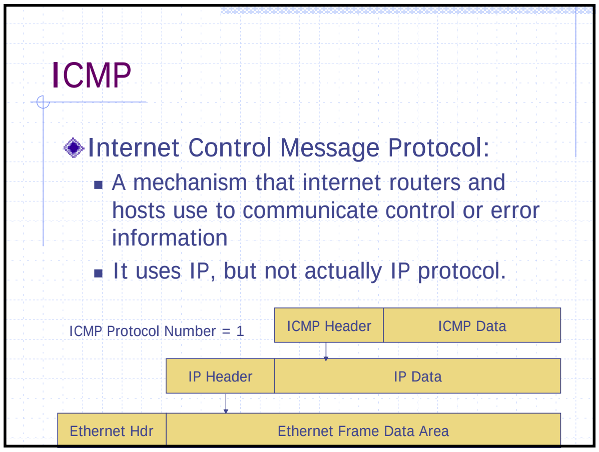
length =1040 | ID =x | fragflag =0 | offset =2960

6



Minimize Fragmentation

- Fragmentation burdens the destination and the router
- Keep fragmentation to the minimum



ICMP

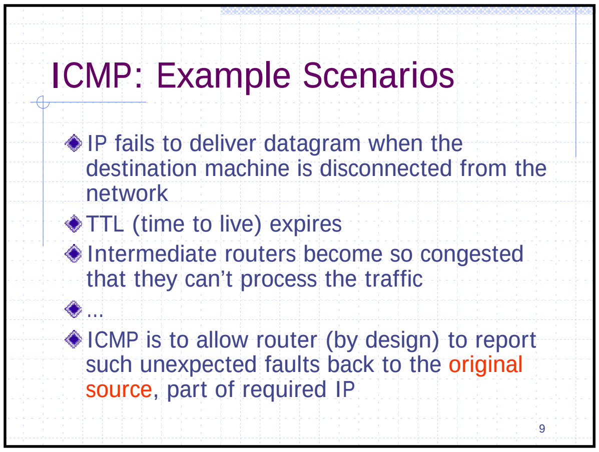
- Internet Control Message Protocol:
  - A mechanism that internet routers and hosts use to communicate control or error information
  - It uses IP, but not actually IP protocol.

ICMP Protocol Number = 1

ICMP Header	ICMP Data

IP Header	IP Data

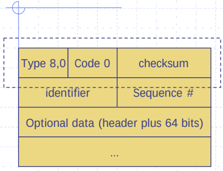
Ethernet Hdr	Ethernet Frame Data Area



ICMP: Example Scenarios

- IP fails to deliver datagram when the destination machine is disconnected from the network
- TTL (time to live) expires
- Intermediate routers become so congested that they can’t process the traffic
- ...
- ICMP is to allow router (by design) to report such unexpected faults back to the original source, part of required IP

ICMP Message Format



Type 8,0	Code 0	checksum
identifier		Sequence #
Optional data (header plus 64 bits)		
...		

Echo request / reply

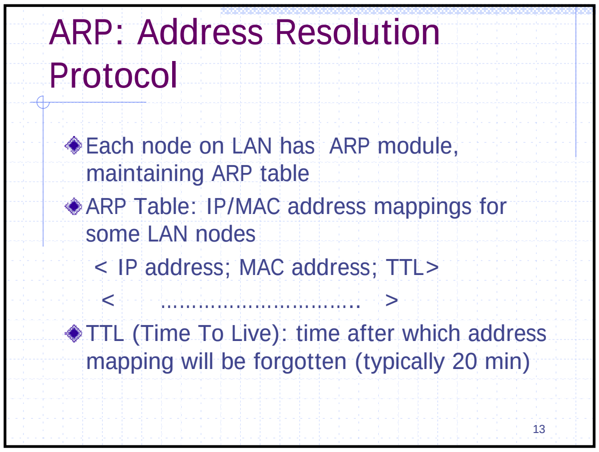
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Test Reachability (Ping)

- A host or router sends an ICMP echo request message to a specified destination
- Any machine that receives echo request must formulate an echo reply message and send to sender
- Successful receipt of a reply verifies that major pieces of transport system work

Traceroute

```
traceroute: Warning: cn.yahoo.com has multiple addresses; using 61.135.128.50
traceroute to cn.yahoo.com (61.135.128.50), 30 hops max, 38 byte packets
 1  r6hm01v163.ns.utk.edu (160.36.30.1)  1.373 ms  0.332 ms  0.322 ms
 2  bsm01v200.ns.utk.edu (160.36.1.104)  0.417 ms  0.515 ms  0.393 ms
 3  atl-edge-19.inet.qwest.net (216.207.16.33)  5.452 ms  5.547 ms  5.484 ms
 4  atl-core-03.inet.qwest.net (205.171.21.125)  5.486 ms  5.688 ms  5.520 ms
 5  atl-core-01.inet.qwest.net (205.171.21.153)  5.836 ms  5.905 ms  5.830 ms
 6  iah-core-03.inet.qwest.net (205.171.8.145)  25.322 ms  25.348 ms  25.325 ms
 7  iah-core-02.inet.qwest.net (205.171.31.41)  25.321 ms  25.419 ms  25.299 ms
 8  bur-core-01.inet.qwest.net (205.171.205.25)  56.697 ms  56.746 ms  56.713 ms
 9  lax-core-01.inet.qwest.net (205.171.8.41)  57.019 ms  57.058 ms  57.022 ms
10  lax-brdr-01.inet.qwest.net (205.171.19.38)  57.064 ms  57.099 ms  57.020 ms
11  202.97.48.65 (202.97.48.65)  264.265 ms  259.337 ms  257.330 ms
12  202.97.51.193 (202.97.51.193)  492.494 ms  470.912 ms  464.106 ms
13  p-13-0-r1-c-bjbj-1.cn.net (202.97.33.9)  958.715 ms  1012.859 ms  1016.328 ms
18  202.108.61.2 (202.108.61.2)  298.953 ms  293.484 ms  300.453 ms
19  cn.yahoo.com (61.135.128.50)  1908.846 ms  1892.476 ms  1953.833 ms
```



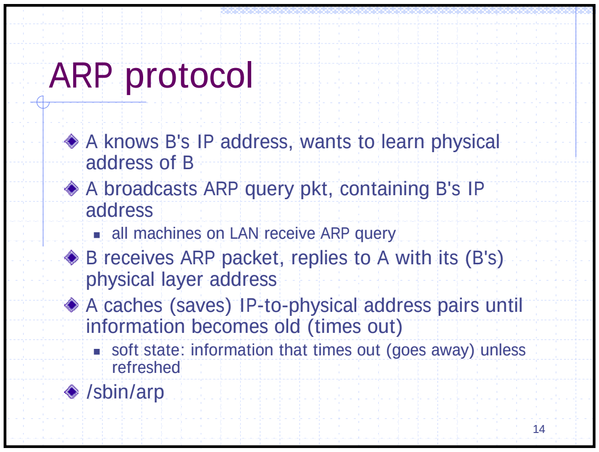
ARP: Address Resolution Protocol

- Each node on LAN has ARP module, maintaining ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

  < IP address; MAC address; TTL>

  < ……………………….. >
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

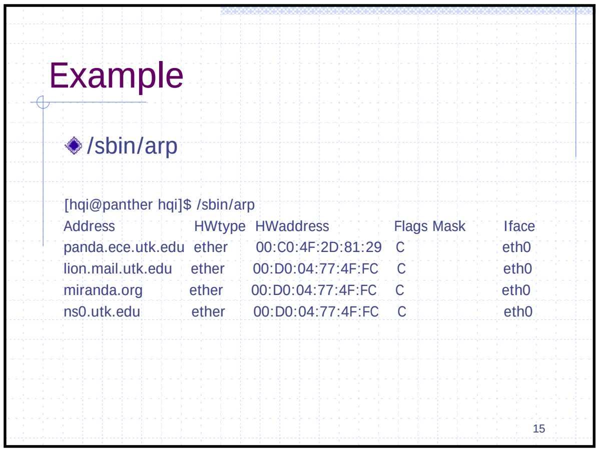
13



ARP protocol

- A knows B's IP address, wants to learn physical address of B
- A broadcasts ARP query pkt, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) physical layer address
- A caches (saves) IP-to-physical address pairs until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed
- /sbin/arp

14



Example

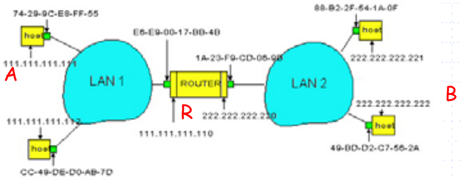
- /sbin/arp

```
[hqi@panther hqi]$ /sbin/arp
Address            HWtype  HWaddress          Flags Mask     Iface
panda.ece.utk.edu  ether   00:C0:4F:2D:81:29  C              eth0
lion.mail.utk.edu  ether   00:D0:04:77:4F:FC  C              eth0
miranda.org        ether   00:D0:04:77:4F:FC  C              eth0
ns0.utk.edu        ether   00:D0:04:77:4F:FC  C              eth0
```

15

Routing to another LAN

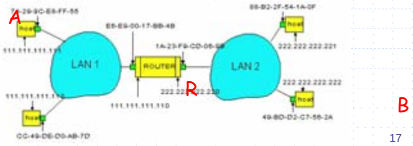
walkthrough: routing from A to B via R



- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B of the router

16

- A creates IP packet with source A, destination B
- A uses ARP to get R's physical layer address for 111.111.111.110
- A creates Ethernet frame with R's physical address as dest, Ethernet frame contains A-to-B IP datagram
- A's data link layer sends Ethernet frame
- R's data link layer receives Ethernet frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's physical layer address
- R creates frame containing A-to-B IP datagram sends to B



17

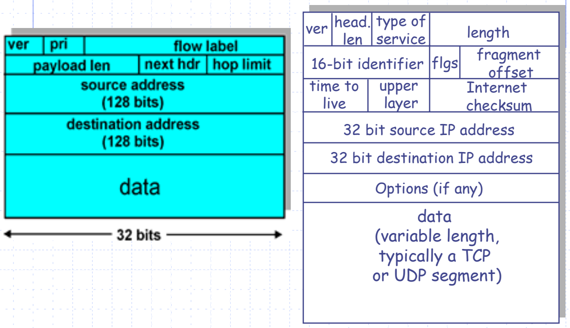
IPv6

18

IPv6

- Initial motivation: 32-bit address space completely allocated by 2008.
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
    - The concept of flow
  - new “anycast” address: route to “best” of several replicated servers

IPv6 Header



ver	pri	flow label
payload len	next hdr	hop limit
source address (128 bits)		
destination address (128 bits)		
data		

ver	head. len	type of service	length
16-bit identifier		flgs	fragment offset
time to live	upper layer		Internet checksum
32 bit source IP address			
32 bit destination IP address			
Options (if any)			
data (variable length, typically a TCP or UDP segment)			

Other Changes from IPv4

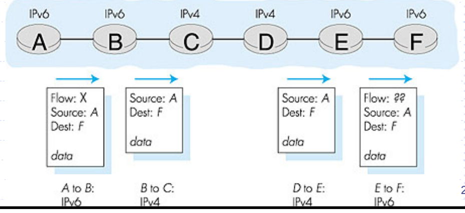
- *Length field*: fixed-length 40 byte header
- *No fragmentation allowed*
- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by “Next Header” field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. “Packet Too Big”
  - Subsumes multicast group management functions (IGMP – Internet Group management Protocol)
  - “Unrecognized IPv6 option”

Transition From IPv4 To IPv6

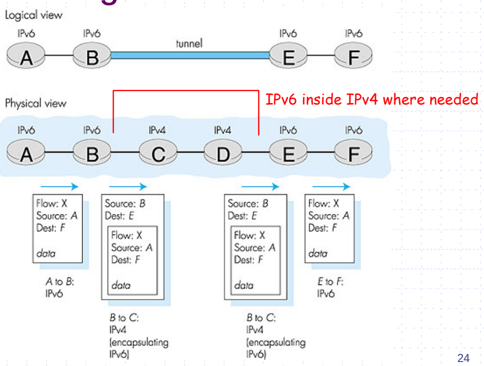
- Flag day?
- Dual stack
  - some routers with dual stack (v6, v4) can “translate” between formats (IPv6/IPv4 nodes)
- Tunneling:
  - IPv6 carried as payload of IPv4 datagram among IPv4 routers

Dual Stack Approach

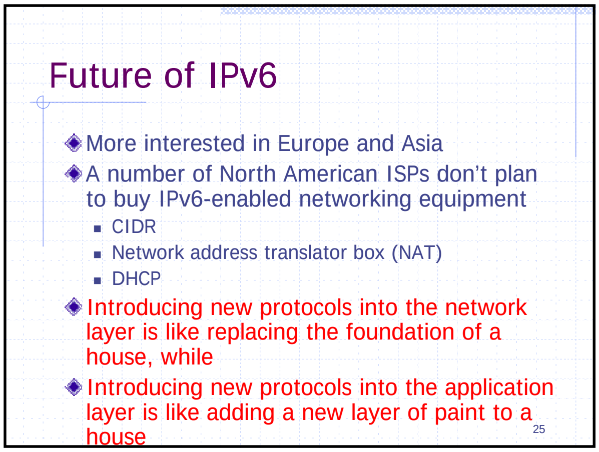
- IPv6/IPv4 nodes must have both IPv6 and IPv4 addresses
- Be able to determine whether another node is IPv6-capable or IPv4-only



Tunneling

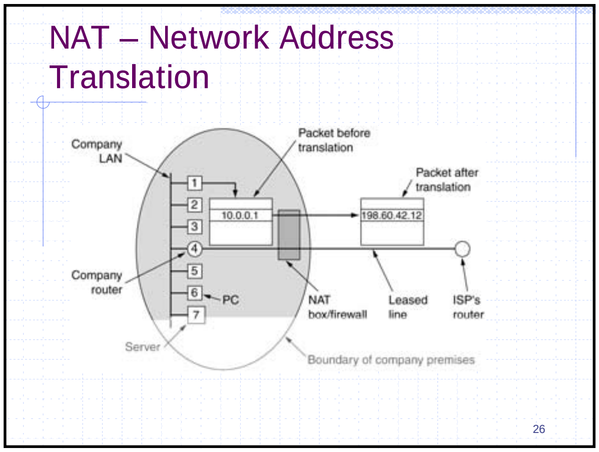


IPv6 inside IPv4 where needed

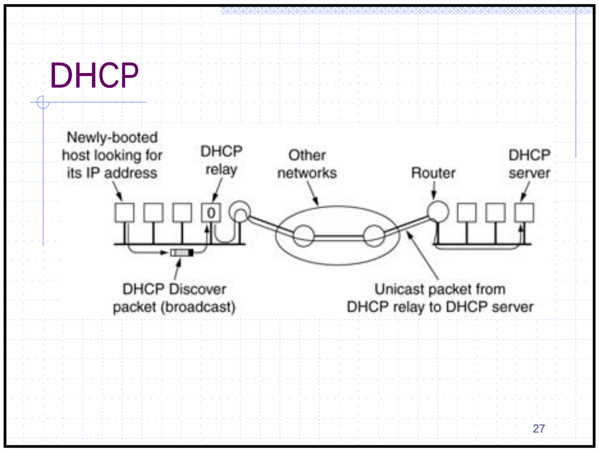


Future of IPv6

- More interested in Europe and Asia
- A number of North American ISPs don't plan to buy IPv6-enabled networking equipment
  - CIDR
  - Network address translator box (NAT)
  - DHCP
- Introducing new protocols into the network layer is like replacing the foundation of a house, while
- Introducing new protocols into the application layer is like adding a new layer of paint to a house



NAT – Network Address Translation



DHCP

Newly-booted host looking for its IP address
DHCP relay
Other networks
Router
DHCP server
0
DHCP Discover packet (broadcast)
Unicast packet from DHCP relay to DHCP server