ECE 356/COMPSI 356
Computer Network Architecture

IP Fragmentation, ARP, and ICMP

Wednesday September 26, 2019

Recap

- Last lecture: IP protocol. IP addressing, IP forwarding
- Materials for this lecture:
  - **PD 3.2**: Fragmentation and Reassembly
  - **PD: 3.2.6, 3.2.8**

Lecture Outline

- IP fragmentation
- ARP
- ICMP

Need for IP Fragmentation and Reassembly

- Packets can go through different types of links
- Each network has some **Maximum Transmission Unit (MTU)**, the largest IP datagram that it can carry in a frame
  - Ethernet: 1500 bytes, FDDI: 4500 bytes
- Would be inefficient to always send the smallest packets possible over all potentially encountered technologies

Duke UNIVERSITY

IP Fragmentation and Reassembly: Strategy (1/2)

- Fragmentation occurs in a router when it receives a datagram that it wants to forward over a network which has (MTU < datagram)
- Reassembly is done at the *receiving host*
  - Not at the intermediate routers

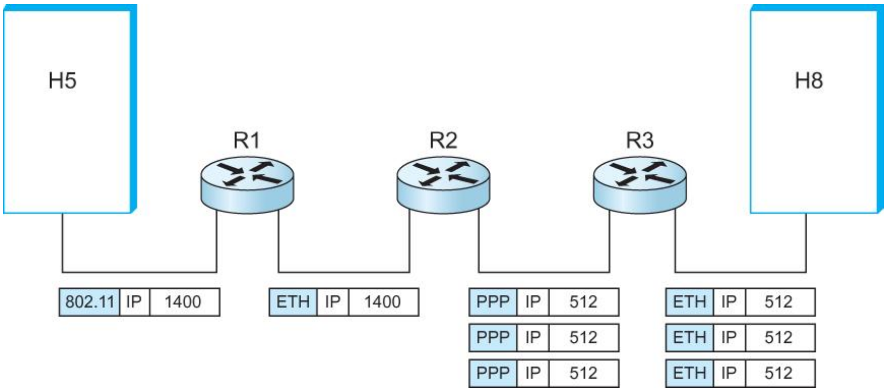
Duke UNIVERSITY

IP Fragmentation and Reassembly: Strategy (2/2)

- All the fragments carry the same identifier in the *Ident* field
- Fragments are self-contained datagrams
- IP does not recover from missing fragments
  - Fragments discarded if a part of the frame is missing

Duke UNIVERSITY

An Example: an IP Datagram Traversing a Sequence of Physical Networks

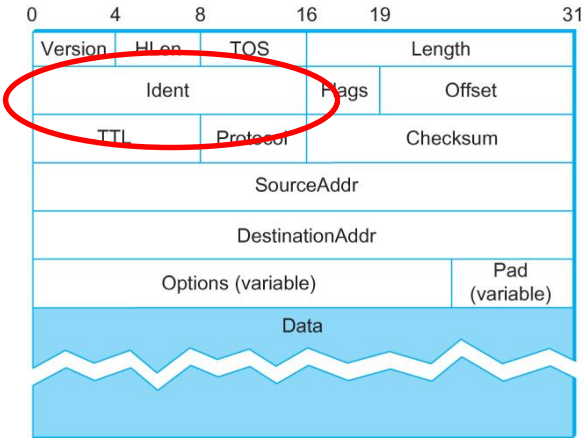


IP Header Format: Fragmentation and Reassembly Fields

- Identification, Flags, Fragment offset
  - Fragmentation and reassembly

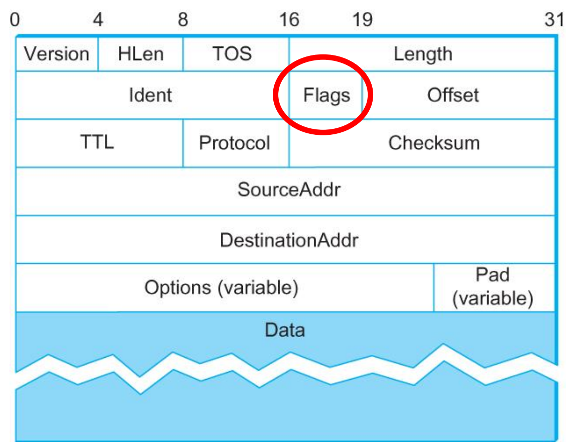
IP Header Format: Identification

- Unique datagram identifier from a host
  - ➢ Incremented whenever a datagram is transmitted (in some OS)
  - ➢ Used by many researchers for various purposes



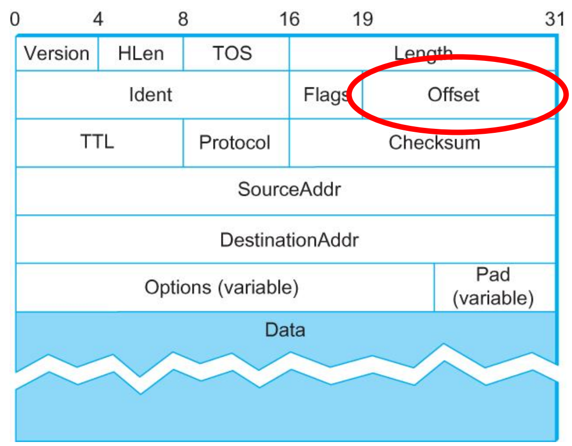
IP Header Format: Flags

- 3 bits:
  - ➢ First bit always set to 0
  - ➢ DF bit (Do not fragment)
  - ➢ MF bit (More fragments)



IP Header Format: Offset

- Fragment offset (13 bits)



Duke UNIVERSITY

Example of Fragmentation

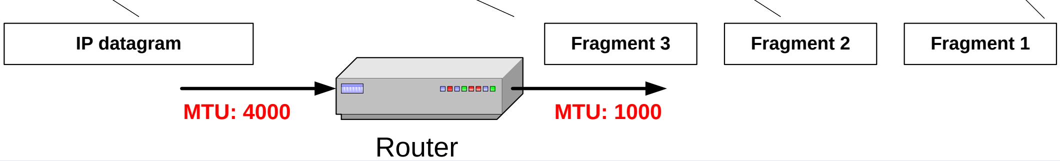
- A datagram with size 2400 bytes must be fragmented according to an MTU limit of 1000 bytes

Header length: 20
Total length: 2400
Identification: 0xa428
DF flag: 0
MF flag: 0
Fragment offset: 0

Header length: 20
Total length: 448
Identification: 0xa428
DF flag: 0
MF flag: 0
Fragment offset: 244

Header length: 20
Total length: 996
Identification: 0xa428
DF flag: 0
MF flag: 1
Fragment offset: 122

Header length: 20
Total length: 996
Identification: 0xa428
DF flag: 0
MF flag: 1
fragment offset: 0



12

Duke UNIVERSITY

Determining the Length of Fragments (1/2)

- Maximum payload length = 1000 – 20 = 980 bytes
- Offset specifies the bytes in multiple of 8 bytes. So the payload must be a multiple of 8 bytes
- 980 - 980 % 8 = 976 (the largest number that is less than 980 and divisible by 8)
- The payload for the first fragment is 976 and has bytes 0 ~ 975 of the original IP datagram. The offset is 0

Determining the Length of Fragments (2/2)

- The payload for the second fragment is 976 and has bytes 976 ~ 1951 of the original IP datagram. The offset is 976 / 8 = 122
- The payload of the last fragment is 2400 – 976 * 2 = 428 bytes and has bytes 1952 ~ 2400 of the original IP datagram. The offset is 244
- Total length of three fragments: 996 + 996 + 448 = 2440 > 2400
  - Why?

Alternative to Fragmentation: Path MTU Discovery

- Fragmentation slows down the router
  - Would be more efficient for the host to send appropriately sized packets in the first place
- How does a sender know the MTU of a path?
  - A host only knows the MTU of its links
- Solution:
  - Sends large packets with DF set
  - If receives ICMP Fragmentation needed messages, reduces maximum segment size

Lecture Outline

- IP fragmentation
- **Address translation (ARP)**
- ICMP

Need for the Address Translation Protocol (ARP)

- How do we find out host's Ethernet address after knowing its IP address?
- The Internet is based on IP addresses
- Data link protocols (Ethernet, FDDI, ATM) may have different MAC addresses
- The ARP protocol perform the **translation between IP addresses and MAC layer addresses**

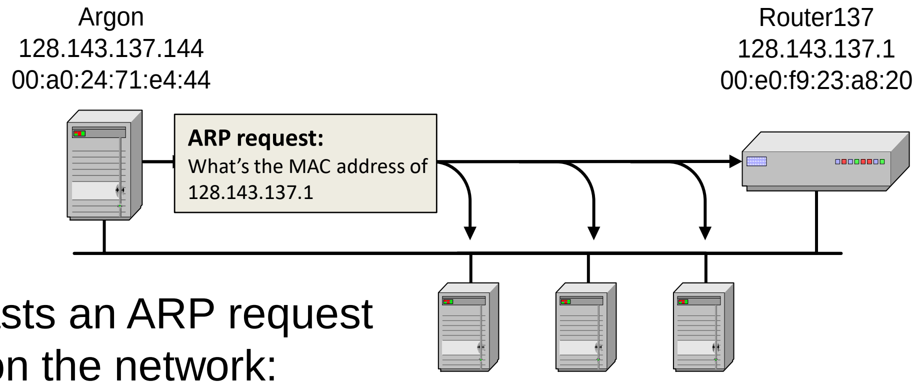
Duke UNIVERSITY

ARP

- In this lecture: ARP for broadcast LANs, particularly Ethernet LANs
  - RFC 826

Duke UNIVERSITY

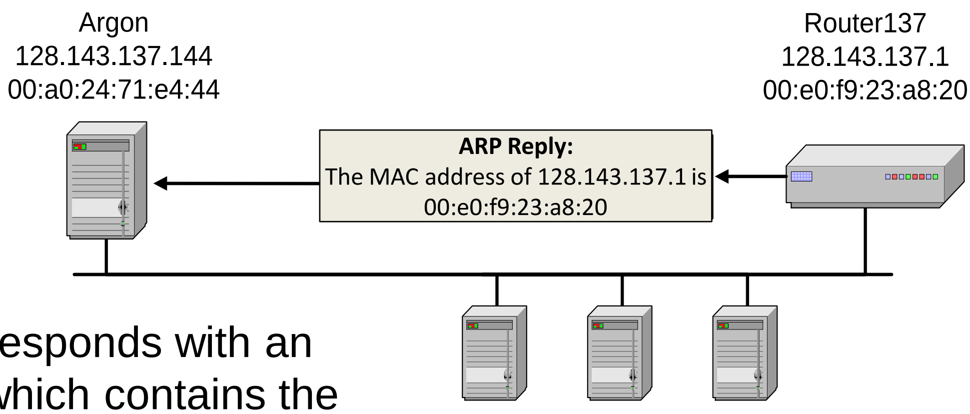
Address Translation with ARP: ARP Request



- Argon broadcasts an ARP request to all stations on the network: **"What is the hardware address of 128.143.137.1?"**

Duke UNIVERSITY

Address Translation with ARP: ARP Reply



- Router 137 responds with an ARP Reply which contains the hardware address

Duke UNIVERSITY

ARP Packet Format (1/9)

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0–3)			
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)	
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)	
TargetHardwareAddr (bytes 2–5)			
TargetProtocolAddr (bytes 0–3)			

- Physical network: Ethernet

ARP Packet Format (2/9)

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0–3)			
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)	
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)	
TargetHardwareAddr (bytes 2–5)			
TargetProtocolAddr (bytes 0–3)			

- Higher-layer protocol type: IP

ARP Packet Format (3/9)

0	8	16 … 31
Hardware type = 1		ProtocolType = 0x0800
HLen = 48	PLen = 32	Operation
SourceHardwareAddr (bytes 0–3)		
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)
TargetHardwareAddr (bytes 2–5)		
TargetProtocolAddr (bytes 0–3)		

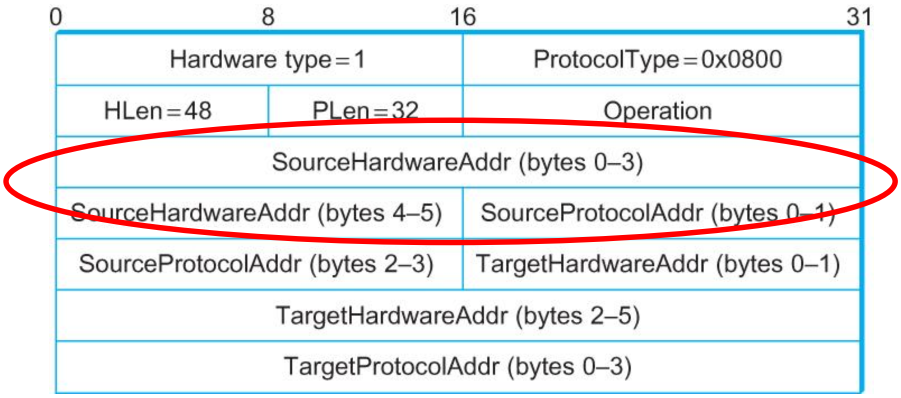
- “Hardware” and “protocol” header lengths
  - Ethernet 48 bits, IP 32 bits

ARP Packet Format (4/9)

0	8	16 … 31
Hardware type = 1		ProtocolType = 0x0800
HLen = 48	PLen = 32	Operation
SourceHardwareAddr (bytes 0–3)		
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)
TargetHardwareAddr (bytes 2–5)		
TargetProtocolAddr (bytes 0–3)		

- Opcode: ARP request 1, ARP reply 2

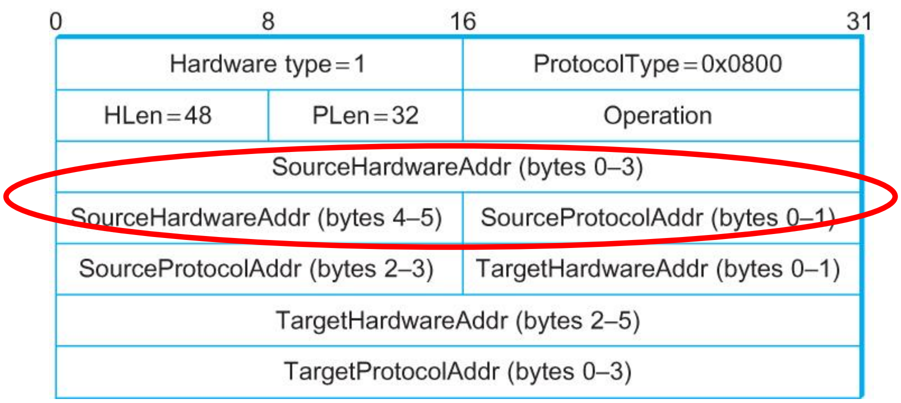
ARP Packet Format (5/9)



- Source hardware address

Duke UNIVERSITY

ARP Packet Format (6/9)



- Source hardware address

Duke UNIVERSITY

ARP Packet Format (7/9)

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0–3)			
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)	
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)	
TargetHardwareAddr (bytes 2–5)			
TargetProtocolAddr (bytes 0–3)			

- Source protocol address

Duke UNIVERSITY

ARP Packet Format (8/9)

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0–3)			
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)	
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)	
TargetHardwareAddr (bytes 2–5)			
TargetProtocolAddr (bytes 0–3)			

- Target hardware address
  - ➢ Request: empty, reply: target Ethernet address

Duke UNIVERSITY

ARP Packet Format (9/9)

0	8	16	31
Hardware type = 1		ProtocolType = 0x0800	
HLen = 48	PLen = 32	Operation	
SourceHardwareAddr (bytes 0–3)			
SourceHardwareAddr (bytes 4–5)		SourceProtocolAddr (bytes 0–1)	
SourceProtocolAddr (bytes 2–3)		TargetHardwareAddr (bytes 0–1)	
TargetHardwareAddr (bytes 2–5)			
TargetProtocolAddr (bytes 0–3)			

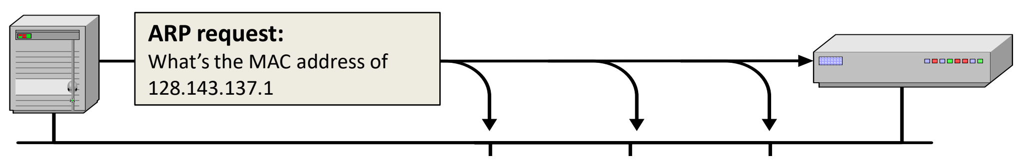
- Target protocol address
  - ➢ Request: target IP address, reply: destination IP address

Duke UNIVERSITY

An Example (1/2)

Argon
128.143.137.144
00:a0:24:71:e4:44

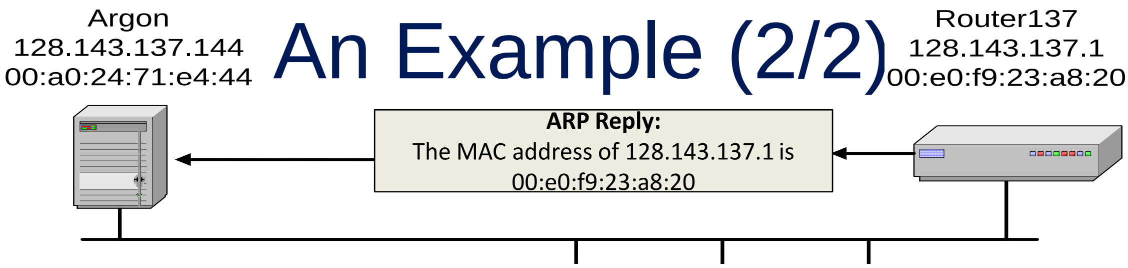
Router137
128.143.137.1
00:e0:f9:23:a8:20



- ARP Request from Argon is broadcasted:
  - ➢ Source addr in Ethernet header: 00:a0:24:71:e4:44
  - ➢ Destination addr in Ethernet header: FF:FF:FF:FF:FF:FF
- Source hardware address: 00:a0:24:71:e4:44
- Source protocol address: 128.143.137.144
- Target hardware address: 00:00:00:00:00:00
- Target protocol address: 128.143.137.1

Duke UNIVERSITY

An Example (2/2)



- ARP Reply from Router137 is unicasted:
  - Source addr: 00:e0:f9:23:a8:20
  - Dst addr: 00:a0:24:71:e4:44
- Source hardware address: 00:e0:f9:23:a8:20
- Source protocol address: 128.143.137.1
- Target hardware address: 00:a0:24:71:e4:44
- Target protocol address: 128.143.137.144

Duke UNIVERSITY

ARP: Comments

- ARP requests: broadcast
  - Other hosts learn the source IP/MAC mapping
- ARP replies: unicast

Duke UNIVERSITY

ARP Table / ARP Cache

- Since sending an ARP request/reply for each IP datagram is inefficient, hosts maintain a cache (ARP Cache) of current entries
  - Entries expire after a time interval
- Linux, Windows, macOS: arp -a

Duke UNIVERSITY

Putting it Together: IP Forwarding Logistics, Lab 2 (1/2)

1. Sanity-check
   - Meets minimum length and has correct checksum
2. Update header
   - Decrement the TTL by 1, and compute the packet checksum over the modified header
3. Next hop IP lookup
   - Find out which entry in the routing table has the longest prefix match with the destination IP address

Duke UNIVERSITY

Putting It Together: IP Forwarding Logistics, Lab 2 (2/2)

4. Next hop MAC lookup
   - Check the ARP cache for the next-hop MAC address corresponding to the next-hop IP. If it's there, send it. Otherwise, send an ARP request for the next-hop IP (if one hasn't been sent within the last second), and add the packet to the queue of packets waiting on this ARP request.
5. Error reporting

Duke UNIVERSITY

Lecture Outline

- IP fragmentation
- Address translation (ARP)
- **Error reporting (ICMP)**

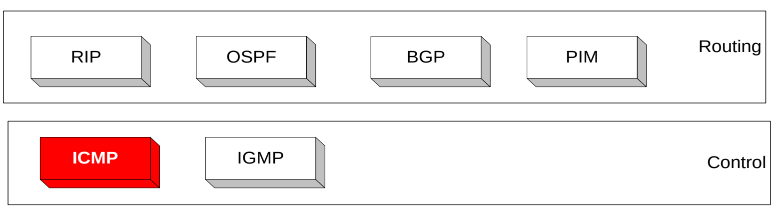
Duke UNIVERSITY

Error Reporting

- Internet Control Message Protocol (ICMP)
  - Ill-formatted packets
  - TTL == 0
  - ARP receives no reply
  - No protocol or application running at the destination
  - No routing table match
  - …

Location in the Protocol Stack

- The Internet Protocol relies on several other protocols to perform necessary control and routing functions:
  - Control functions (ICMP)
  - Multicast signaling (IGMP)
  - Setting up forwarding tables (RIP, OSPF, BGP, PIM, …)



ICMP: An Overview

- The **Internet Control Message Protocol (ICMP)** is a helper protocol that supports IP with facility for:
  - Simple queries
  - Error reporting
- ICMP messages are encapsulated as IP datagrams
  - Often considered part of IP, but architecturally lies above it

IP header	ICMP message
	← IP payload →

ICMP Message Format

bit # 0–7	8–15	16–23	24–31
type	code	checksum	
additional information or 0x00000000			

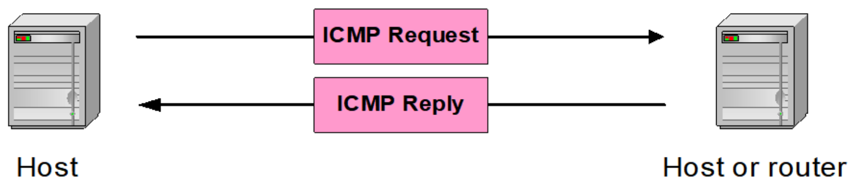
4 byte header:

- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum. Checksum is calculated over the entire ICMP message

If there is no additional data, there are 4 bytes set to zero

→ Each ICMP message is at least 8 bytes long

ICMP Query Message



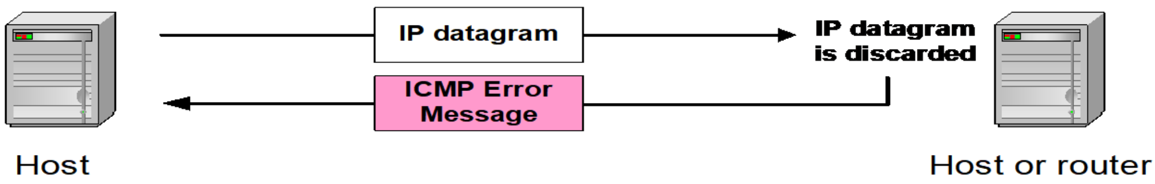
ICMP query:

- Request sent by host to a router or host
- Reply sent back to querying host

Example of ICMP Queries

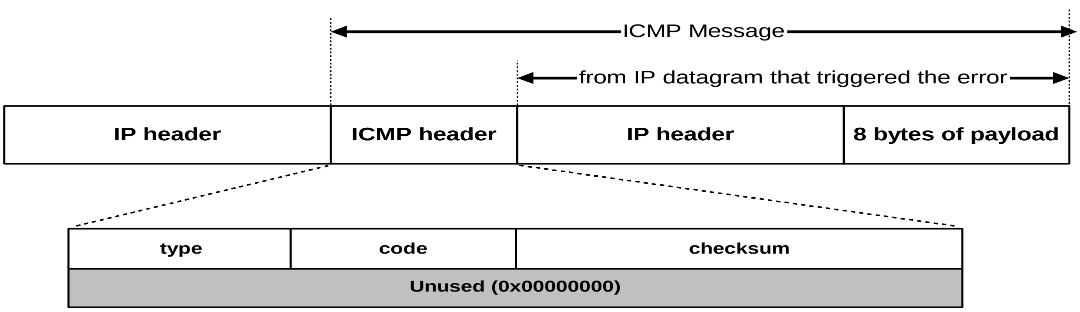
Type/Code:	Description	
8/0	Echo Request	The ping command uses Echo Request/ Echo Reply
0/0	Echo Reply	
13/0	Timestamp Request	
14/0	Timestamp Reply	

ICMP Error Message



- ICMP error messages report error conditions
- Typically sent when a datagram is discarded
- Error message is often passed from ICMP to the application program

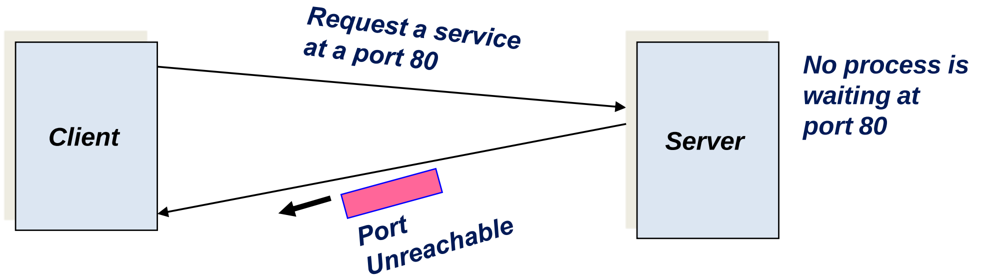
ICMP Error Message



- ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)

Example: ICMP Port Unreachable

- RFC 792: If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a "*destination unreachable*" message to the source host.



Common ICMP Error Messages

Type	Code	Description	
3	0–5	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.
11	0, 1	Time exceeded	Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)
12	0, 1	Parameter problem	Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)

Some Subtypes of the "Destination Unreachable"

Code	Description	Reason for Sending
0	Network Unreachable	No routing table entry is available for the destination network.
1	Host Unreachable	Destination host should be directly reachable, but does not respond to ARP Requests.
2	Protocol Unreachable	The protocol in the protocol field of the IP header is not supported at the destination.
3	Port Unreachable	The transport protocol at the destination host cannot pass the datagram to an application.
4	Fragmentation Needed and DF Bit Set	IP datagram must be fragmented, but the DF bit in the IP header is set. (MTU discovery)
5	Source route failed	The source routing option has failed.

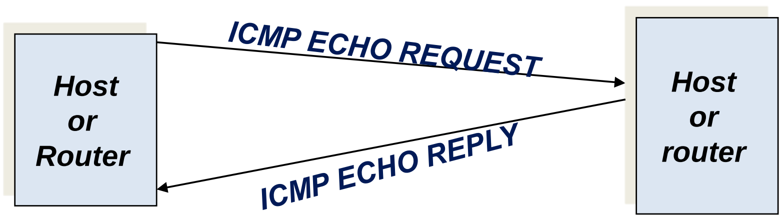
Duke UNIVERSITY

ICMP Applications

- Ping
  - **ping** www.duke.edu
- Traceroute
  - **traceroute** nytimes.com
- MTU discovery

Duke UNIVERSITY

Ping: Echo Request and Reply



- Pings are handled directly by the kernel
- Each ping is translated into an ICMP Echo Request
- Pinged host responds with an ICMP Echo Reply

Duke UNIVERSITY

Traceroute

- Linux and MAC OS: traceroute google.com
- Windows: tracert google.com

Duke UNIVERSITY

Traceroute Algorithm

- Sends out UDP packets with TTL= 1, 2, …, n, with an unlikely port number, starts timers for them
  - Standard implementation: 3 packets for each TTL value
- Each router on the path sends ICMP “Time exceeded” message (type 11, code 0)
  - Includes the name and the address of the router
  - Sender calculates the round-trip time
- Destination replies with a “Port unreachable” ICMP message (type 3, code 3). The process stops.

Duke UNIVERSITY

Traceroute: An Example

```
C:\Users\maria>tracert nytimes.com

Tracing route to nytimes.com [151.101.129.164]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  10.197.0.2
  2     *        *        *     Request timed out.
  3     2 ms     7 ms    18 ms  10.236.254.226
  4     2 ms     2 ms     6 ms  tel1-sp-wireless-vrf-v4311.netcom.duke.edu [10.236.242.130]
  5     5 ms     3 ms     4 ms  tel-edge-gw1-t0-0-0-1.netcom.duke.edu [10.236.254.102]
  6     8 ms     9 ms    11 ms  hntvl-gw-to-duke-tel-edge.ncren.net [128.109.247.93]
  7    11 ms    10 ms    11 ms  rtp-gw-to-hntvl-gw.ncren.net [128.109.9.5]
  8    12 ms    11 ms    11 ms  et-3-3-0.582.rtsw.rale.net.internet2.edu [198.71.47.221]
  9    96 ms    30 ms    17 ms  ae-1.4079.rtsw.wash.net.internet2.edu [162.252.70.121]
 10     *      320 ms    17 ms  23.235.41.187
 11     *        *        *     Request timed out.
 12    17 ms    18 ms    16 ms  151.101.129.164

Trace complete.
```

Duke UNIVERSITY

Path MTU Discovery Algorithm

- Send packets with DF bit set
- If receive an ICMP error message, reduce the packet size

Duke UNIVERSITY

Summary

- IP fragmentation
- ARP
- ICMP

Duke UNIVERSITY

Next Lecture

- Introduction to Lab 2
- Routing: Dynamic Routing Protocol

Duke UNIVERSITY