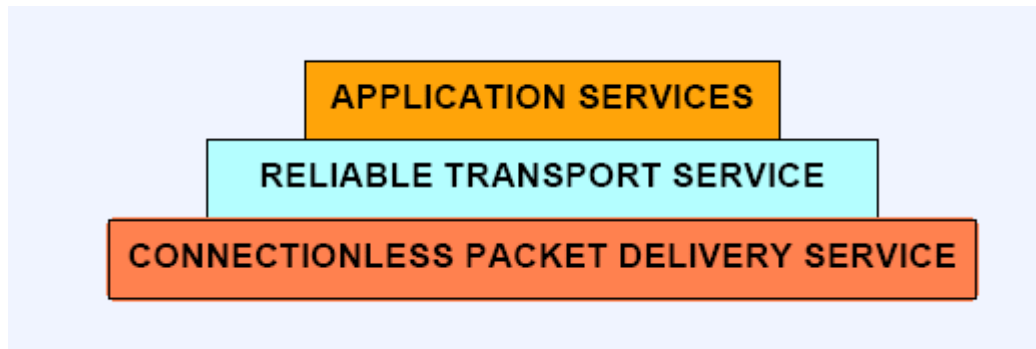


Internet Protocols (IP)

(1) Internet Protocols

❖ Internet Architecture and Philosophy

- A TCP/IP internet provides three sets of services as shown in the following figure



❖ Connectionless Delivery System

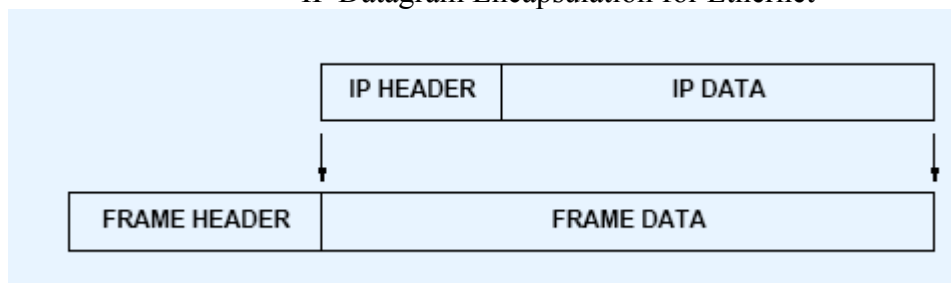
- The most fundamental internet service consists of a packet deliver system, which is *unreliable*, *best-effort*, and *connectionless*.
- *Unreliable*: packets may be lost, duplicated, delayed, or delivered out of order.
- *Connectionless*: each packet is treated independently from all others.
- *Best-effort*: the Internet software makes an earnest attempt to deliver packets.

❖ Purpose of the Internet Protocol

- The IP protocol defines the basic unit of data transfer (IP datagram)
- IP software performs the *routing* function
- IP includes a set of rules that embody the idea of unreliable packet delivery:
 - How hosts and routers should process packets
 - How and when error messages should be generated
 - The conditions under which packets can be discarded.

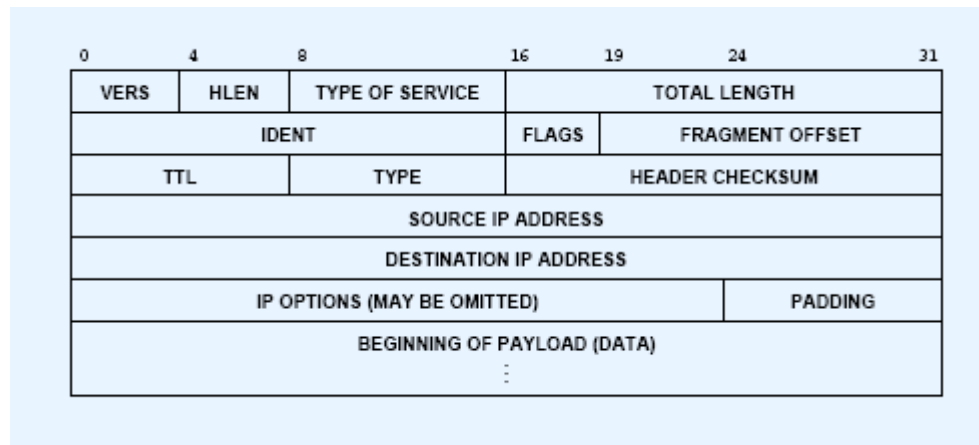
❖ IP Datagram Encapsulation

IP Datagram Encapsulation for Ethernet



(2) IP Header

❖ IP Header Format



- ❖ VERS: current version is 4, I.e. IPv4
 - proposal for IPv6, which will have a different header
- ❖ HLEN: header length in # 32-bit words
 - Normally = 5, i.e. 20 octet IP headers
 - Max 60 bytes
 - Header can be variable length (IP option)
- ❖ TYPE OF SERVICE 3-bit precedence field (unused), 4 TOS bits, 1 unused bit set to 0
 - TOS bit 1 (min delay), 2 (max throughput), 3 (max reliability), 4 (min cost): only one can be set
 - typically all are zero, for best-effort service
 - DiffServ proposes to use TOS for IP QOS
- ❖ TOTAL LENGTH: of datagram, in bytes
 - Max size is 65535 bytes (64K – 1)
- ❖ IDENT, FLAGS, FRAGMENT OFFSET:
 - Used for fragmentation and reassembly, will talk about this later
- ❖ TTL (Time To Live): upper limit on # routers that a datagram may pass through
 - Initialized by sender, and decremented by each router. When zero, discard datagram. This can stop routing loops
 - Example: `ping -t TTL IP` allows us to specify the TTL field
 - Question: normal users are not supposed to be able to modify the TTL field, how does ping do that? (the *SetUID* concept)

- Question: How to implement `traceroute`? i.e., how to find the routers to a destination (without using IP options)?
 - Use `TTL=1,2,3,...`

```
% ping -t 20 www.dell.com

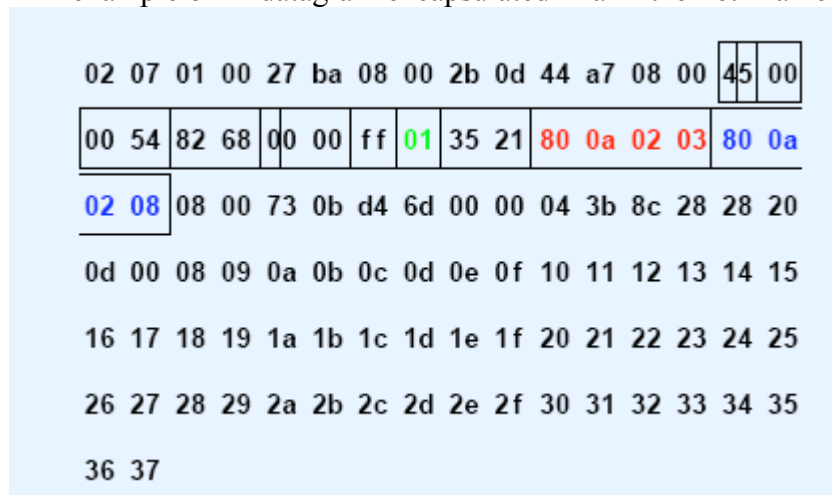
Output: www.dell.com is alive

-----
% ping -t 10 www.dell.com

Output: ICMP Time exceeded in transit from
        hagg-01-ge-1-3-0-508.ausu.twtelecom.net (66.192.253.165)
        for icmp from enyo (128.230.208.110) to www.dell.com
        (143.166.83.230)
```

- ❖ TYPE: IP needs to know to what protocol it should hand the received IP datagram
 - In essence, it specifies the format of the DATA area
 - Demultiplexes incoming IP datagrams into either UDP, TCP, ICMP...
- ❖ HEADER CHECKSUM
 - 16-bit 1's complement checksum
 - Calculated only over header
 - Recomputed at each hop
- ❖ An example of IP datagram
 - Header length: 20 octet
 - TYPE: 01 (ICMP)
 - Source IP: 128.10.2.3
 - Destination IP: 128.10.2.8

An example of IP datagram encapsulated in an Ethernet Frame



❖ IP OPTIONS

- IP OPTIONS field is not required in every datagram
- Options are included primarily for network testing or debugging.
- The length of IP OPTIONS field varies depending on which options are selected.

❖ Record Route Option

- The sender allocates enough space in the option to hold IP addresses of the routers (i.e., an empty list is included in the option field)
- Each router records its IP address to the record route list
- If the list is full, router will stop adding to the list
- Example: `ping -R` (on Solaris)

```
% ping -R -v -s www.yahoo.com
```

```
Output:
```

```
IP options: <record route> 128.230.93.1, 128.230.85.1,  
67.99.63.126, L0.a1.nwyk.broadwing.net (216.140.10.58),  
216.140.10.197, L0.a1.nwak.broadwing.net (216.140.8.250),  
Broadwing-Level3-oc12.NewYork1.Level3.net (63.211.54.70),  
ge-5-0.core1.NewYork1.Level3.net (4.68.97.40),  
lo-0.bbr2.NewYork1.Level3.net (209.247.8.252)
```

❖ Timestamp Option

- Works like the record route option
- Each router along the path fills in a 32-bit integer timestamp

❖ Source Routing

- It provides a way for the sender to dictate a path through the Internet.
- Strict Source Routing
 - The list of addresses specifies the exact path the datagram must follow to reach its destination
 - An error results if a router cannot follow a strict source route
- Loose Source Routing
 - The list of addresses specifies that the datagram must follow the sequence of IP addresses, but allows multiple network hops between successive addresses on the list
- Question: how are these two types of source routing implemented?

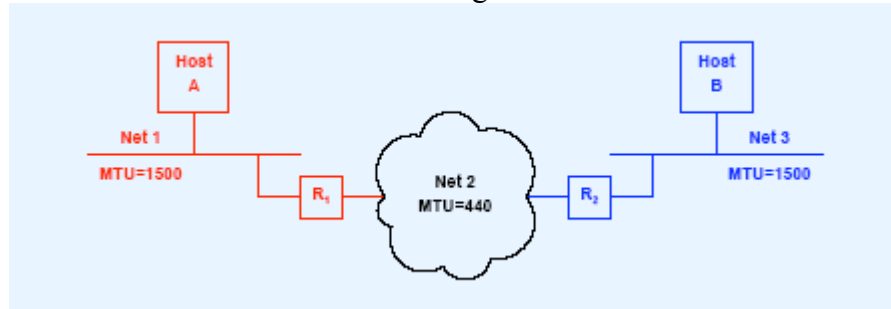
(3) IP Fragmentation

❖ Why do we need fragmentation?

- MTU: Maximum Transmission Unit
- An IP datagram can contain up to 65535 total octets (including header)

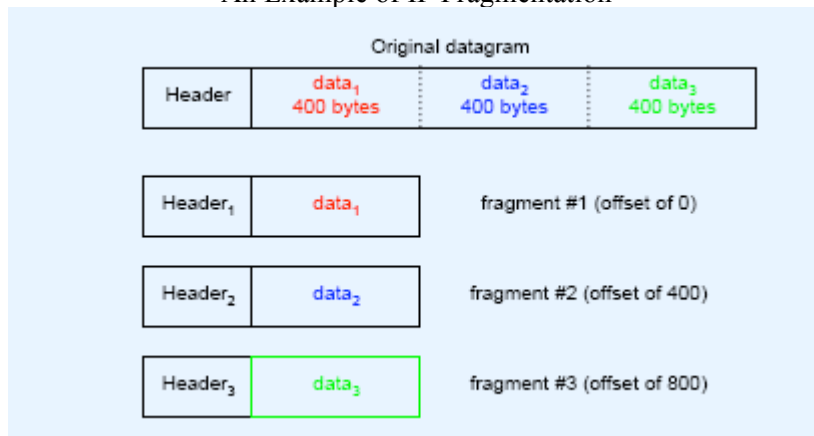
- Network hardware limits maximum size of frame (e.g., Ethernet limited to 1500 octets, i.e., MTU=1500; FDDI limited to approximately 4470 octets/frame)

Illustration of When Fragmentation is Needed



- ❖ IP fragmentation
 - Routers divide an IP datagram into several smaller fragments based on MTU
 - Fragment uses same header format as datagram
 - Each fragment is routed independently
- ❖ How is an IP datagram fragmented?
 - IDENT: unique number to identify an IP datagram; fragments with the same identifier belong to the same IP datagram
 - FRAGMENT OFFSET:
 - Specifies where data belongs in the original datagram
 - Multiple of 8 octets
 - FLAGS:
 - bit 0: reserved
 - bit 1: do not fragment
 - bit 2: more fragments. This bit is turned off in the last fragment (Q: why do we need this bit? A: the TOTAL LENGTH field in each fragment refers to the size of the fragment and not to the size of the original datagram, so without this bit, the destination does not know the size of the IP datagram)

An Example of IP Fragmentation



- Example: Header + 400 + 400 + 400
 - Header 1: FLAGS=001 and OFFSET = 0
 - Header 2: FLAGS=001 and OFFSET = 400/8 = 50
 - Header 2: FLAGS=000 and OFFSET = 800/8 = 100

- ❖ How are IP fragments reassembled?
 - All the IP fragments of a datagram will be assembled before the datagram is delivered to the layers above.
 - Where should they be assembled? At routers or the destination?
 - They are assembled at the destination.
 - IP reassembly uses a timer. If timer expires and there are still missing fragments, all the fragments will be discarded.

- ❖ Question: if you are implementing the IP fragmentation, what (**malicious**) situations do you need to consider? Malicious situations are those that are intentionally created by adversaries, rather than occurring naturally.
 - What do you do if you never get the last missing piece?
 - What do you do if you get overlapping fragments?
 - What do you do if the last byte of a fragment would go over the maximum size of an IP packet, i.e., if the size of all reassembled fragments is larger than the maximum size of an IP packet?

- ❖ Attack 1: Denial of Service Attack
 - 1st fragment: offset = 0
 - 2nd fragment: offset = 64800
 - Result: The target machine will allocate 64 kilobytes of memory, which is typically held for 15 to 255 seconds. Windows 2000, XP, and almost all versions of Unix are vulnerable.

- ❖ Attack 2: TearDrop
 - Send a packet with:
 - offset = 0
 - payload size N
 - More Fragments bit on
 - Second packet:
 - More Fragments bit off
 - offset + payload size < N
 - i.e., the 2nd fragment fits entirely inside the first one.
 - When OS tries to put these two fragments together, it crashes.

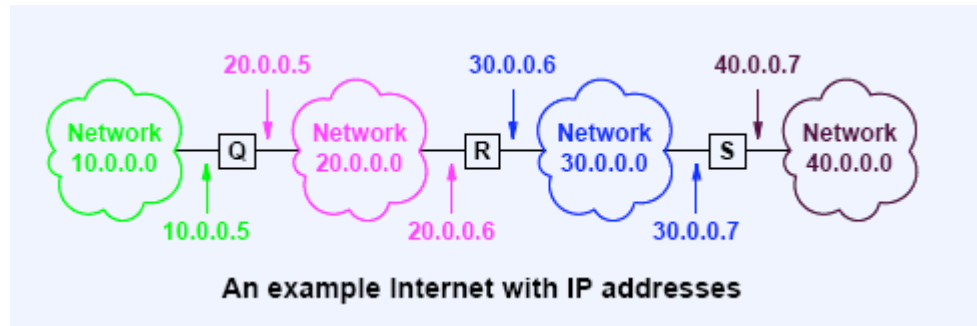
- ❖ Overlapping attacks against firewalls
 - Many firewalls inspect packet separately. When the filtering rule is based on TCP header, but the TCP header is fragmented, the rule will fail
 - TCP header is at the beginning of the data area of an IP packet.
 - Firewalls often check TCP header: for example, SYN packet for connection request.
 - Tiny Fragment Attack: Assumption: firewalls only check the packets with offset=0.
 - Overlapping attacks: Assumption: firewalls only check the packets with offset=0.

(4) IP Spoofing

- ❖ Spoofing:
 - Any host can send packets pretending to be from any IP address
 - Replies will be routed to the appropriate subnet.
 - ❖ Egress (outgoing) Filtering
 - Remove packets that couldn't be coming from your network; however it doesn't benefit you directly, so few people do it.
 - ❖ Ingress (incoming) Filtering: remove packets from invalid (e.g. local) addresses.
 - ❖ To conduct IP spoofing, one needs the superuser privilege.
-

(5) Routing

- ❖ Router vs. Host
 - A router has direct connections to two or more networks, has multiple network cards and multiple IP addresses.
 - A host usually connects directly to one physical network.
- ❖ Direct and Indirect Delivery
 - Direct delivery: ultimate destination can be reached over one network
 - Indirect delivery: requires intermediary (router)
- ❖ Routing table
 - Used by routers to decide how to send datagram
 - Only stores address of next router along the path
 - Scheme is known as next-hop routing
 - (We will discuss later on how to construct routing tables)
- ❖ Next-Hop Routing
 - The destination IP address will not change, the next hop's MAC address is used.
 - Routing table entries (the router R's IP is 20.0.0.6 and 30.0.0.6):



TO REACH NETWORK	ROUTE TO THIS ADDRESS
20.0.0.0 / 8	DELIVER DIRECT
30.0.0.0 / 8	DELIVER DIRECT
10.0.0.0 / 8	20.0.0.5
40.0.0.0 / 8	30.0.0.7

The routing table for router R

- ❖ Host-Specific Routes:
 - Allows per-host routes to be specified as a special case
- ❖ Default Routes
 - Only selected if no other match in table
 - Especially for hosts.
- ❖ IP Routing Algorithm

1. Extract destination IP address D, and compute the network prefix, N;
2. Is N the same network?
3. Is there a specific route for D?
4. Is there a route for N?
5. Is there a default route?
6. Report error.

- ❖ Handling Incoming Datagrams
 - Host: accept or drop. Don't forward. Hosts are forbidden from attempting to forward datagrams that are accidentally routed to the wrong machine. Why?
 - Router: accept or forward.
 - Forwarding: decrease TTL field, recompute the header checksum.
 - Dropping: TTL=0; send an error message to the source.
- ❖ Manipulate routing tables: the `route` command (Linux, Windows, Solaris)