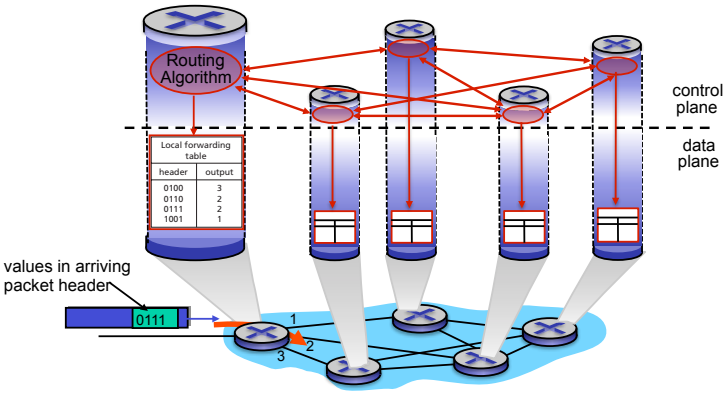
Network Layer (I)

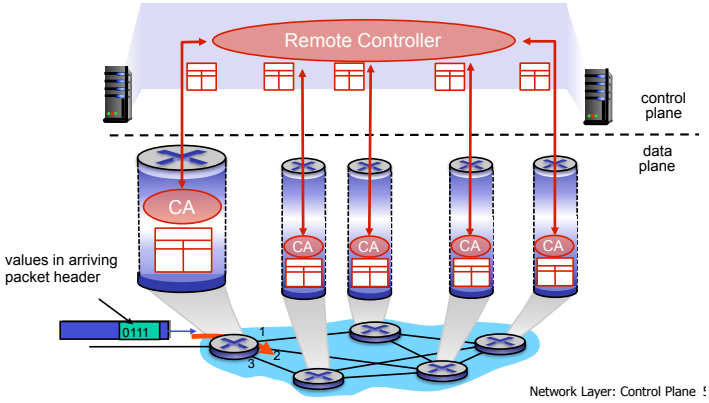
Network-layer Functions, Data Plane and Control Plane

- **Forwarding**:
  - Task of the **data plane**
    - Data plane: move packets from router's input to appropriate router output
- **Routing**:
  - Task of the **control plane**
    - Control plane: determine route taken by packets from source to destination

- *Two approaches* to structuring network control plane:
  - per-router control (traditional)
    - Individual routing algorithm components in each and every router interact in the control plane



  - logically centralized control (software defined networking)
    - A distinct (typically remote) controller interacts with local control agents (CAs)
    - The routing path is computed by the remote controller



Network Layer General Functions

- Addressing: IP address
- Error control: rerouting, ICMP (Internet Control Message Protocol)
- QoS: TOS (Type of Service) field and Differentiated Services

ICMP

- **Internet Control Message Protocol**
  - used by hosts & routers to communicate network-level information
  - typically used for diagnostic/control purpose/generated in response to errors in IP
- Runs on top of IP but still within the network layer

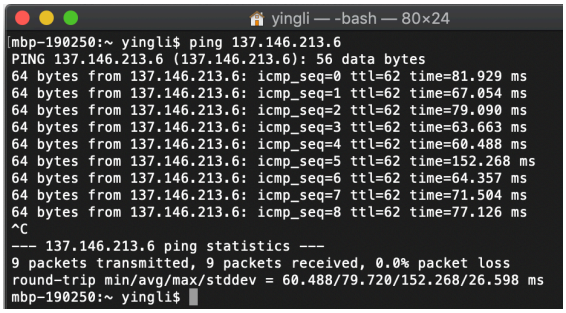
- **Examples**:
  - ping (Echo Request/Reply)
    - uses ICMP to send requests to the destination and replies to the source
  - traceroute (Time-to-live Exceeded, Destination Unreachable)
    - every node on the path sends three responses to measure the RTT using ICMP or to tell the destination is unreachable.
  - “TTL is 0” (Packet Drop)
    - a router uses ICMP packet to indicate the source of a packet that the lifetime exceeded and the packet is dropped.

- ICMP messages use its type and code fields to specify its purposes. These values are well defined in the protocol. The list below shows a subset of the values as an example.

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

Ping

- ping 137.146.213.6 and open Wireshark filter on ip.src == 137.146.103.160 && icmp



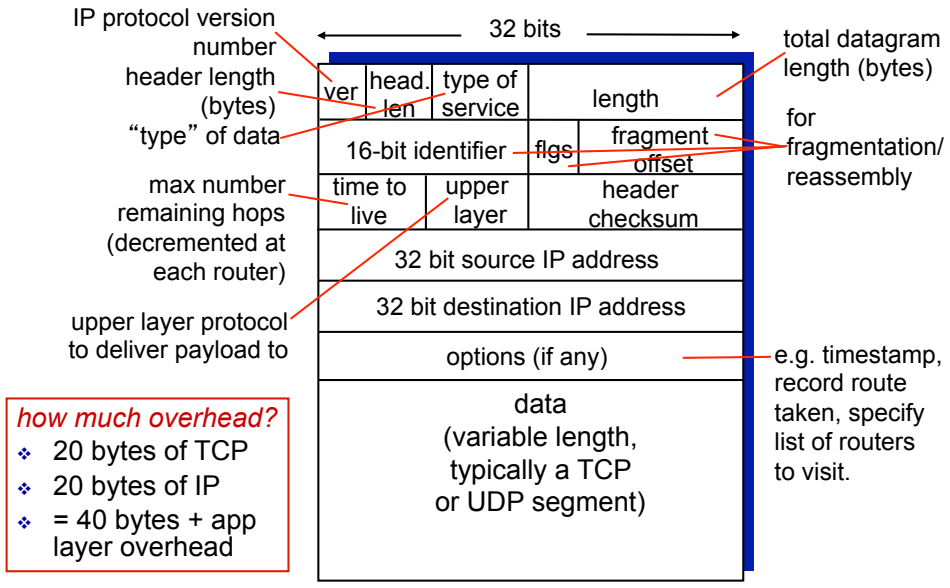
- ping is a network tool to check the status of the network
- It uses ICMP and the client-server model
- The client sends an ICMP request (type 8 code 0), and the server sends an ICMP response (type 0 code 0) back to the client after receiving the request.
- The ping program calculates the RRT for each ping packet based on the request and response messages
- After ctrl-c, the ping program calculates the statistics based on the RTT of all ping packets
- Then, we can use the statistics to tell the status of the network. (packet loss and RRT).

IP Design Goals

- Should have fields for source and destination IP addresses
- Needs means for error control:
  - detection of packet header corruption (L2 does the heavy lifting)
  - limiting the lifespan of a packet (TTL not in TCP)
- Fragmentation
  - carrying transport layer messages that are longer than what L2 can support

IPv4 Header

- version: v4 or v6; v6 has a different format, so decide how to parse the binary sequence
- TOS: distinguish different types of IP datagrams.
  - e.g. it might be useful to distinguish real-time datagrams (such as those used by an IP telephony application) from non-real-time traffic (e.g. FTP). The specific level of service to be provided is a policy issue determined and configured by the network administrator for that router.
- upper layer: used only when an IP datagram reaches its final destination, indicating the transport layer protocol, 6 - TCP, 17 - UDP, 1 - ICMP, so that the destination will know how to parse the data.



Fragmentation

- Problem:
  - network layer needs to deliver a PDU (protocol data unit) that is longer than what the link layer permits
- Solutions:
  - drop and inform (default behavior of IPv6)
  - break the PDU to smaller units (fragments) and reassemble them at the destination (default behavior of IPv4)

IPv4 Fragmentation

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Identification         |0 D M|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- Identification (16 bits)
  - identifies the original fragmented packet
  - sender generates it and increments the number for each packet it sends
- Fragment Offset (13 bits)
  - specifies the location (in 8-byte blocks) of the fragment in the packet
- M (1 bit) - More Fragments
- D (1 bit) - Do not Fragments

Fragmentation Example

- Example:

- Transport layer Packet length 4,500 bytes, MTU 2,500 bytes, IP header 20 bytes:

Fragment	Total bytes	Header bytes	Data bytes	"More fragments" flag	Fragment offset (bytes)
1	2500	20	2480	1	0
2	2040	20	2020	0	310

  - Offset Calculation
    - 0
    - 0 + 2480/8 = 310

Disclaimer: Notes adapted from the textbook and online resources.