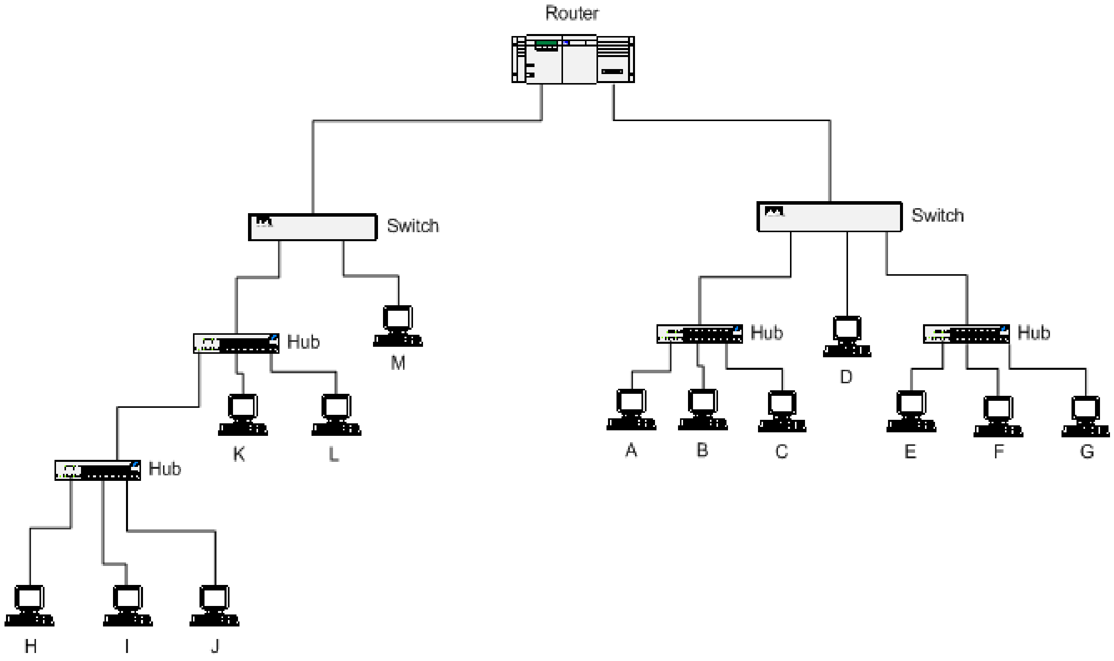
I. Introduction to the Internet Protocol (IP)

- Ethernet addresses are layer 2 (data link) addresses
- IP addresses are layer 3 (network) addresses
- each station/node on a network has two addresses (one of each) with a 1-1 correspondence, per NIC
- **hub** – dumb device, simple repeaters, operate at layer **?**
- **switch** – operate at layer **?**, so only care about **?** addresses
- **router** – a device that connects 2 or more networks
  - can connect networks of different physical topologies
  - has 2 or more protocol interface layers
  - only makes (routing) decisions based upon layer ? (3) addresses
- **internetwork** (or internet) – a collection of networks of similar protocols (note this is not the same as the **I**nternet
- **network segment** – portion of the network where the medium is shared
  - hub: entire hub is a segment
  - switch: each port/host is a segment
  - router: each host is a separate network
  - see Wikipedia for network segment
- routers and switches have data link (MAC) addresses
- routers also have network (IP) addresses
- see handouts



Thus given the diagram above:

How many networks?

How many segments on NW 1?

How many segments on NW 2?

II. Introduction to IP Addresses

- IP addresses are 4 octets in size,? bits for IPv4
- IP addresses are written as 4 decimal numbers, separated by dots…referred to as **dotted-decimal** notation, e.g. 196.128.4.96
- each decimal number represents 8 bits, so the largest decimal number portion of any IP address is ? (2^8 -1) = 255
- part of the IP address describes the source/destination network, part of the address describes the source/destination host

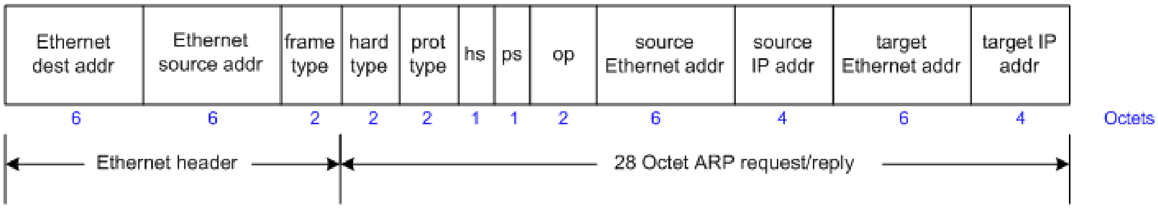
III. Introduction to Address Resolution Protocol (ARP)

- each station on a network has 2 corresponding addresses, a **network** address and a **hardware** address, for delivery across networks, must have both

- given a network address, we use ARP to get hardware address

- ARP is a 2½ layer protocol, or a helper protocol between layers 2 & 3

- ARP packet/datagram (see RFC 826)



Ethernet dest addr	Ethernet source addr	frame type	hard type	prot type	hs	ps	op	source Ethernet addr	source IP addr	target Ethernet addr	target IP addr	
6	6	2	2	2	1	1	2	6	4	6	4	Octets

- **hardware type** (2 octets) – identifies the hardware medium type address, DIX Ethernet = 1, see RFC 1700 (search for hrd)

- **protocol type** (2 octets) – identifies the upper layer (sending) protocol, typically x0800 for IP

- **hardware size** (1 octet) – size of hardware address, in octets, for ht = 1, hs = ? (6)

- **protocol size** (1 octet) – size of the protocol type address in octets, for x0800, ps = ? (4)

- **operation** (2 octet) – identifies the type of ARP operation, 1 = request, 2 = reply

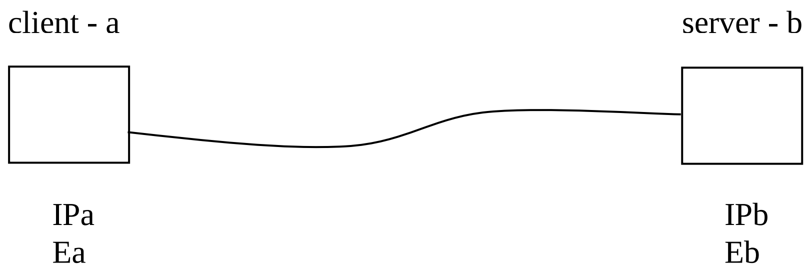
- **source hardware address** (hs octets) – source hardware address

- **source network address** (ps octets) – source network address

- **destination hardware address** (hs octets) – destination hardware address, when op = 1, this is 00:00:00:00:00:00

- **destination network address** (ps octets) – destination network address

What happens when we request data from a web server?



1. type in URL (in words, typically), hit <CR> or GO Layer ? (5)

2. need a network address, not words, so use DNS to resolve name and give network address, but still don’t have hardware address Layer ? (5)

3. to get hardware address, send data down stack and build ARP packet Layer ? (2½)

HT = 1	PT = x0800	HS = 6	PS = 4	OP = 1
Src HW: Ea	Src NW: IPa	Dest HW: ?	Dest NW: IPb	

4. send to data link layer, add MAC header and trailer and send across wire:

   MAC dest: ? MAC src: Ea Ethertype: x0806

   - packet is then broadcast to everyone (Who has hardware address ? tell me)
   - node with that IP is the only node to respond

5. ARP on target machine then:

   - changes OP to ? (2)
   - switch source and destination addresses, adding missing HW address, which is now in the source field

6. add MAC data and send back

Note: ARP packets never leave source network! Why? Either go to a host on your network or to a router

Broadcast packets **NEVER** cross routers!