The Internet Protocol Suite

an overview with details

Matt Crawford

crawdad@fnal.gov

Data Communications

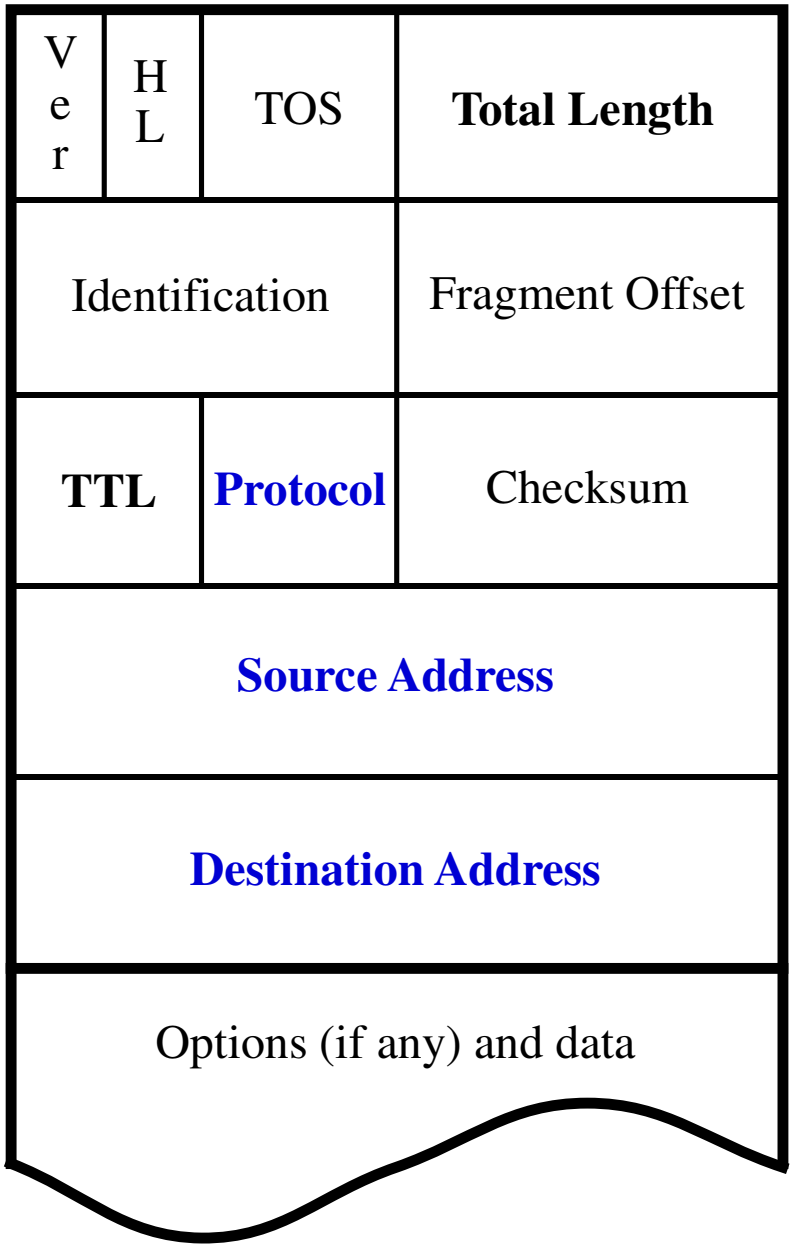
Architecture of the Internet Protocol

The network provides an *unreliable, connectionless* datagram service. Being connectionless, the "network" does not have to remember anything between one packet and the next. Unreliability leaves it up to the end nodes to arrange detection and retransmission of lost packets.

This model is close to what many physical media provide, and allows simple tasks to be performed simply. Complicated tasks must be performed by the end nodes running a higher-level protocol (such as TCP) on top of IP.

IP uses a 32-bit address for each network interface of each network node. Automatic configuration is virtually non-existent (contrast with Appletalk), so machines must have their addresses configured manually.

IP Packet Header Format



Version = 4

HL = Header Length in 4 byte units

TOS = Type of Service

Total Length is bytes, including header

Identification is for reassembly or error reply

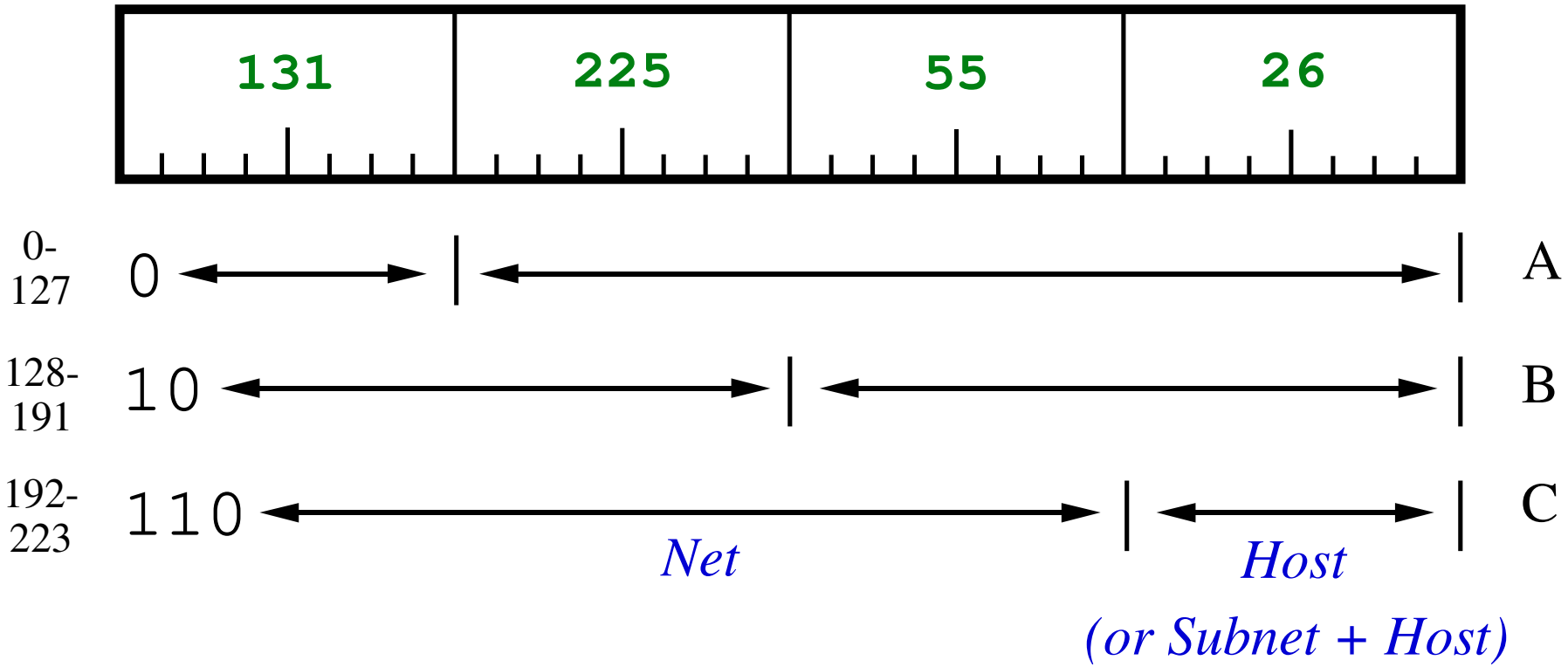
Fragment Offset locates this fragment within the reassembled datagram

TTL = Time To Live, used as hop limit

Protocol designates the next layer up -- TCP, UDP, etc.

Options, if present, are padded to 4 byte multiple

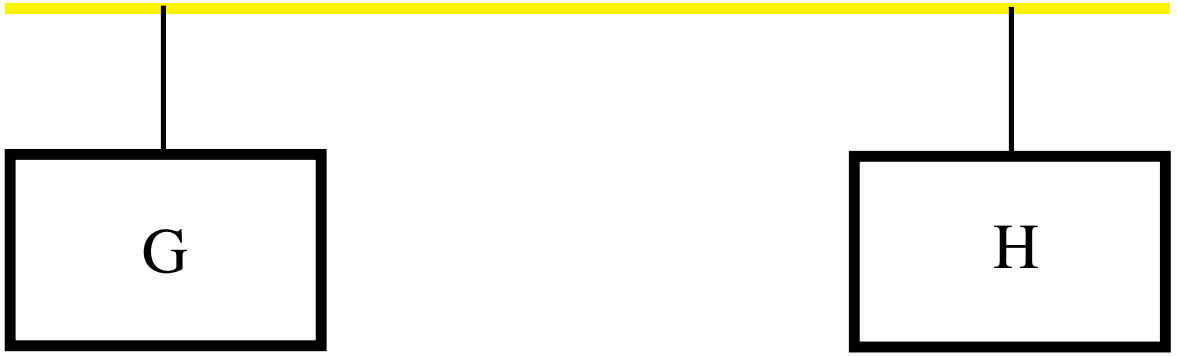
IP Addresses



Why are IP addresses different than Ethernet addresses?
IP was not invented for Ethernet, nor vice-versa.

OK, so how do you translate between them?
With ARP -- the Address Resolution Protocol.

ARP Example - Two Systems



IP 131.225.1.1
Ether 08-00-69-01-23-45

IP 131.225.1.2
Ether 00-00-A7-12-3C-E7

Arp Query

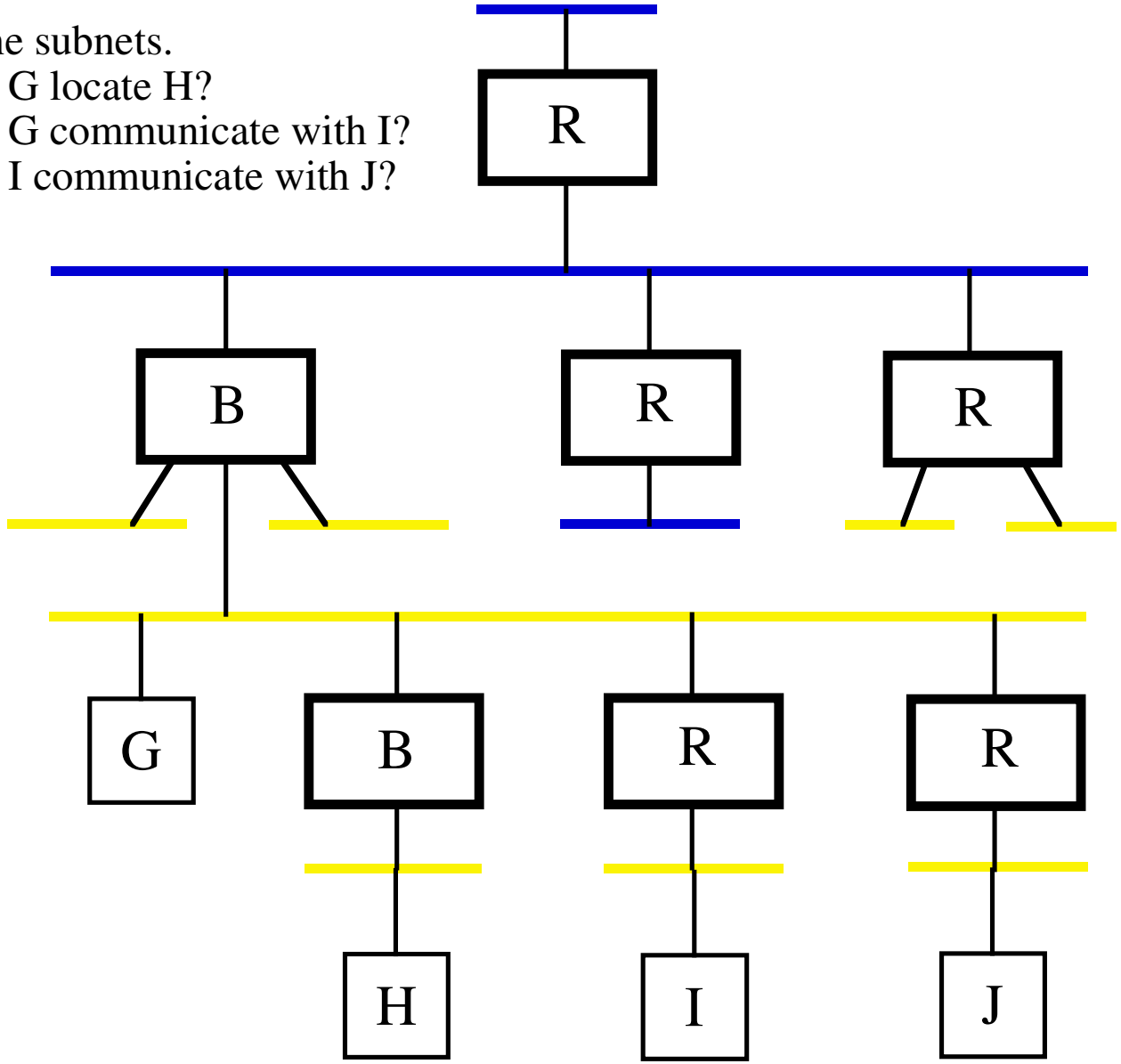
Dst	FF-FF-FF-FF-FF-FF
Src	08-00-69-01-23-45
Hardware=Ethernet	Proto=IP
Sender Proto	131.225.1.1
Sender H'ware	08-00-69-01-23-45
Target Proto	131.225.1.2
Target H'ware	(blank)

Arp Reply

Dst	08-00-69-01-23-45
Src	00-00-A7-12-3C-E7
Hardware=Ethernet	Proto=IP
Sender Proto	131.225.1.2
Sender H'ware	00-00-A7-12-3C-E7
Target Proto	131.225.1.1
Target H'ware	08-00-69-01-23-45

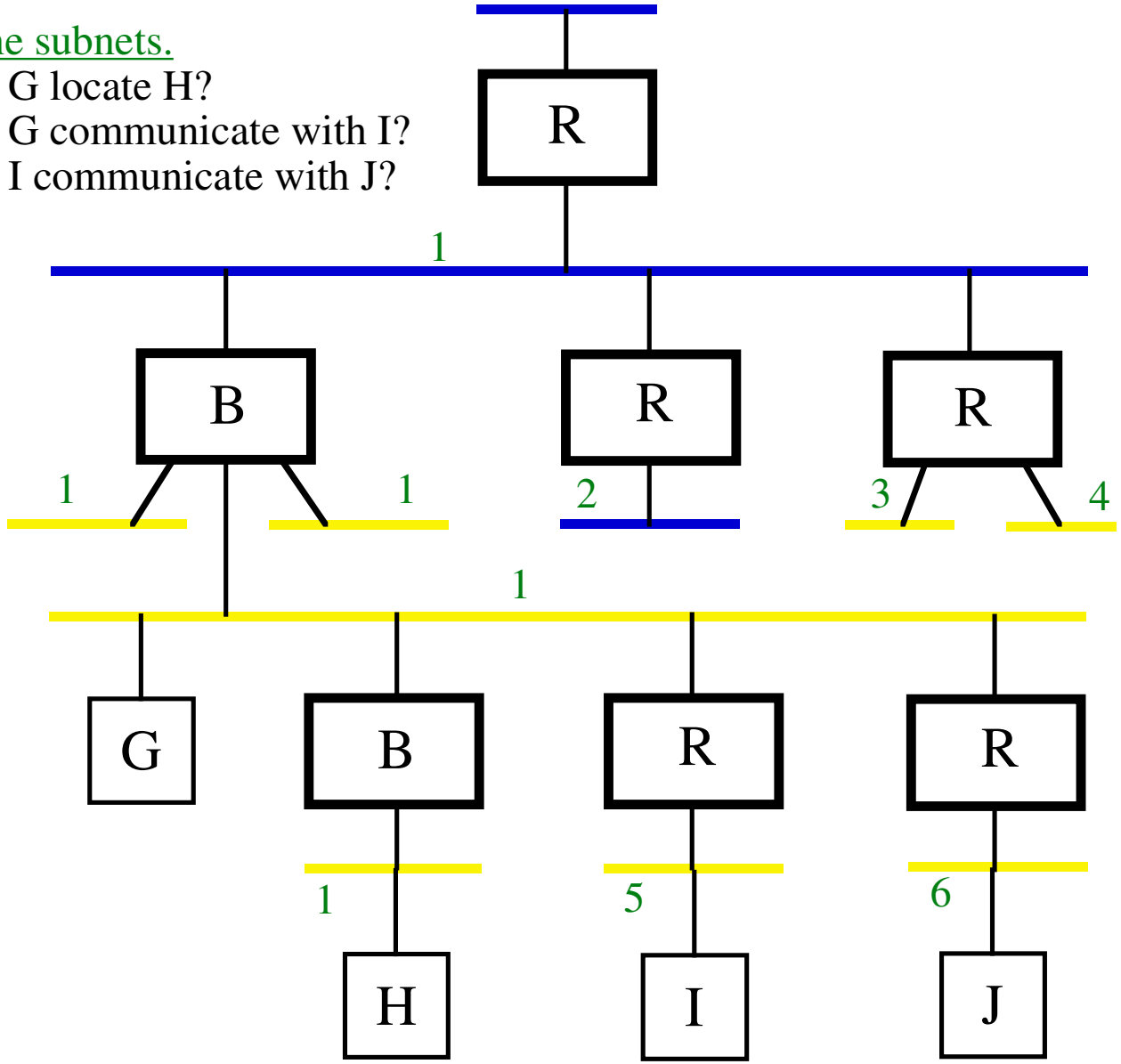
Example - A Larger Network

- Identify the subnets.
- How does G locate H?
- How does G communicate with I?
- How does I communicate with J?



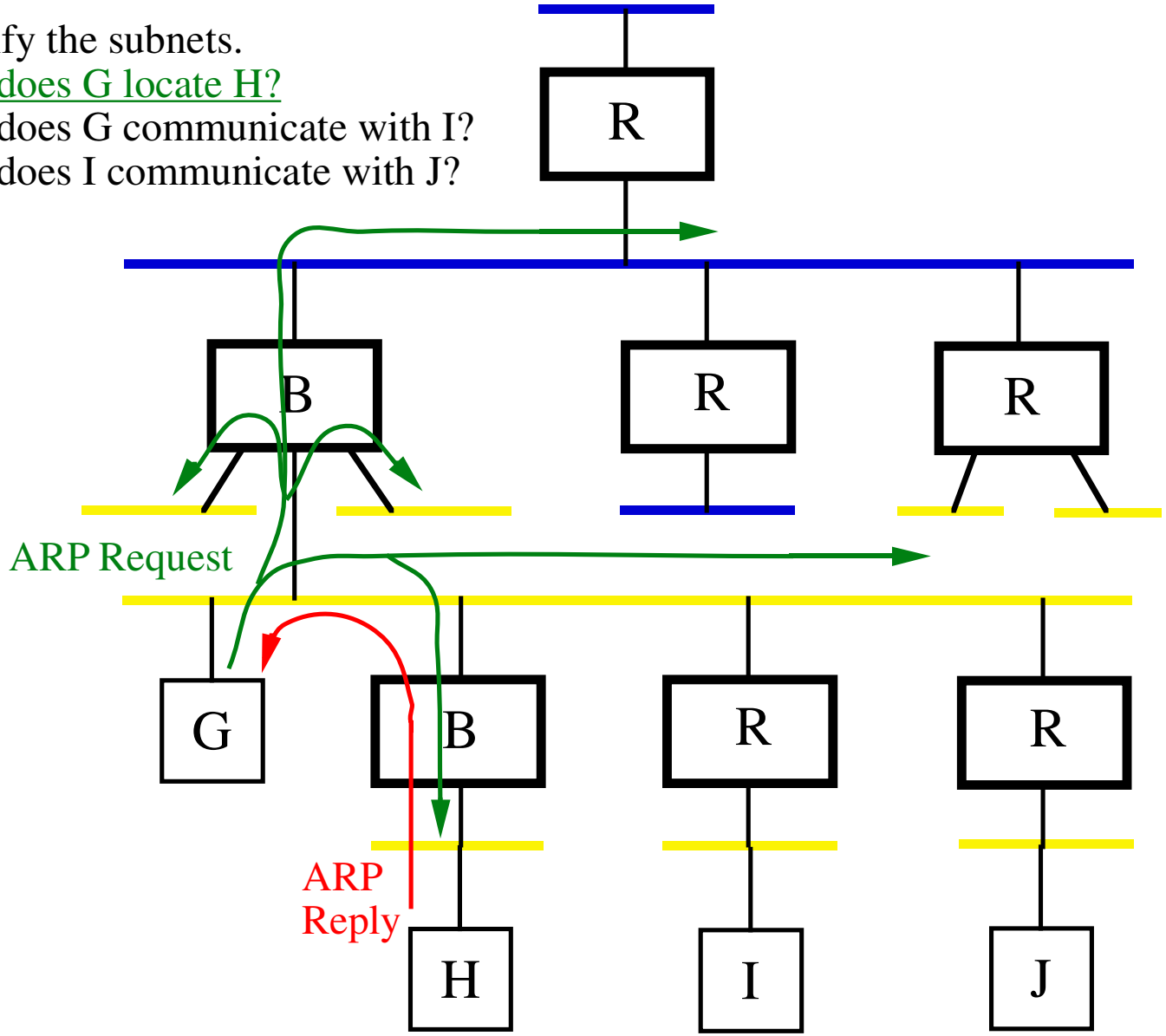
Example - A Larger Network

- Identify the subnets.
- How does G locate H?
- How does G communicate with I?
- How does I communicate with J?



Example - A Larger Network

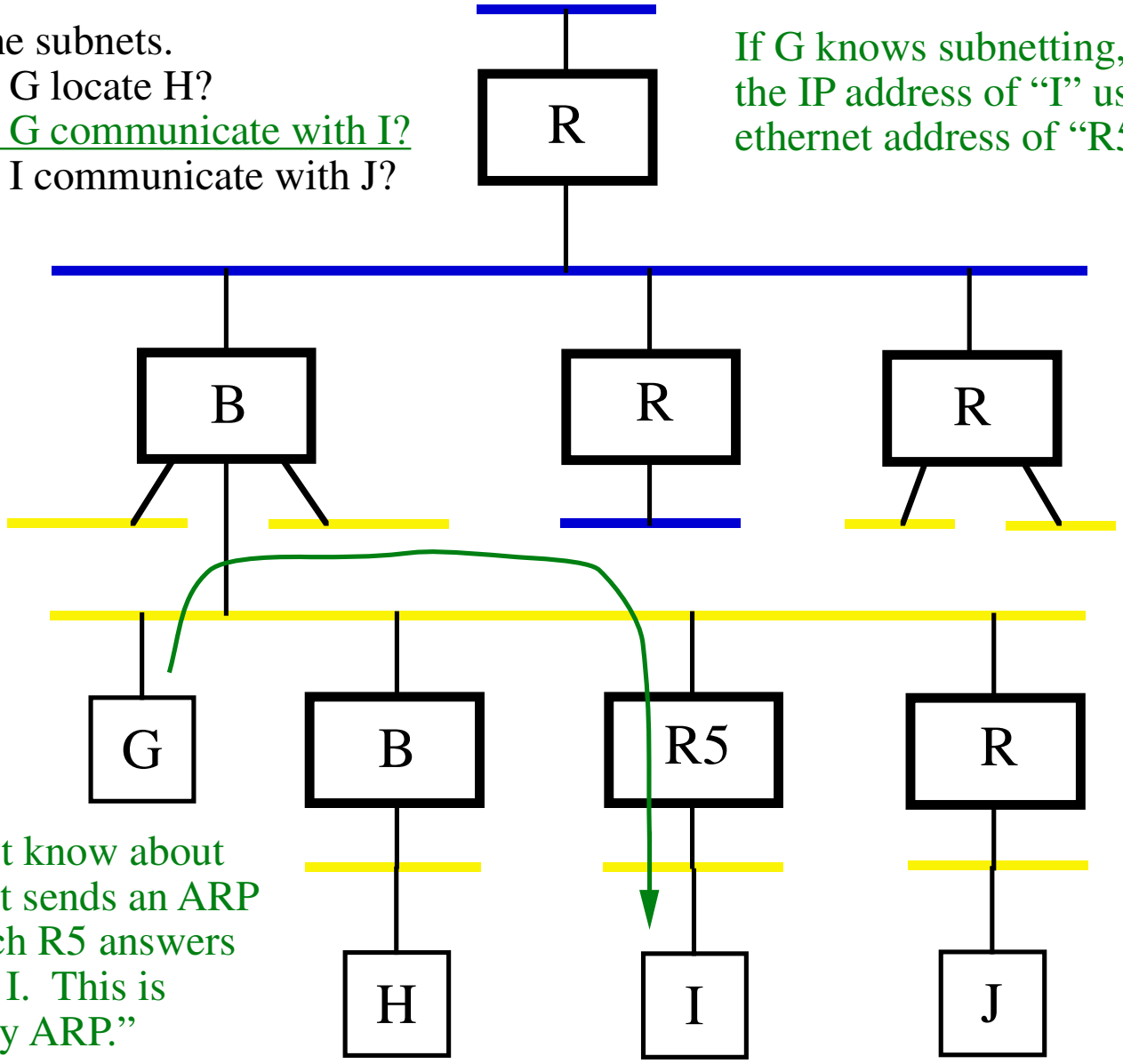
- Identify the subnets.
- How does G locate H?
- How does G communicate with I?
- How does I communicate with J?



Example - A Larger Network

- Identify the subnets.
- How does G locate H?
- How does G communicate with I?
- How does I communicate with J?

If G knows subnetting, it sends to the IP address of "I" using the ethernet address of "R5."

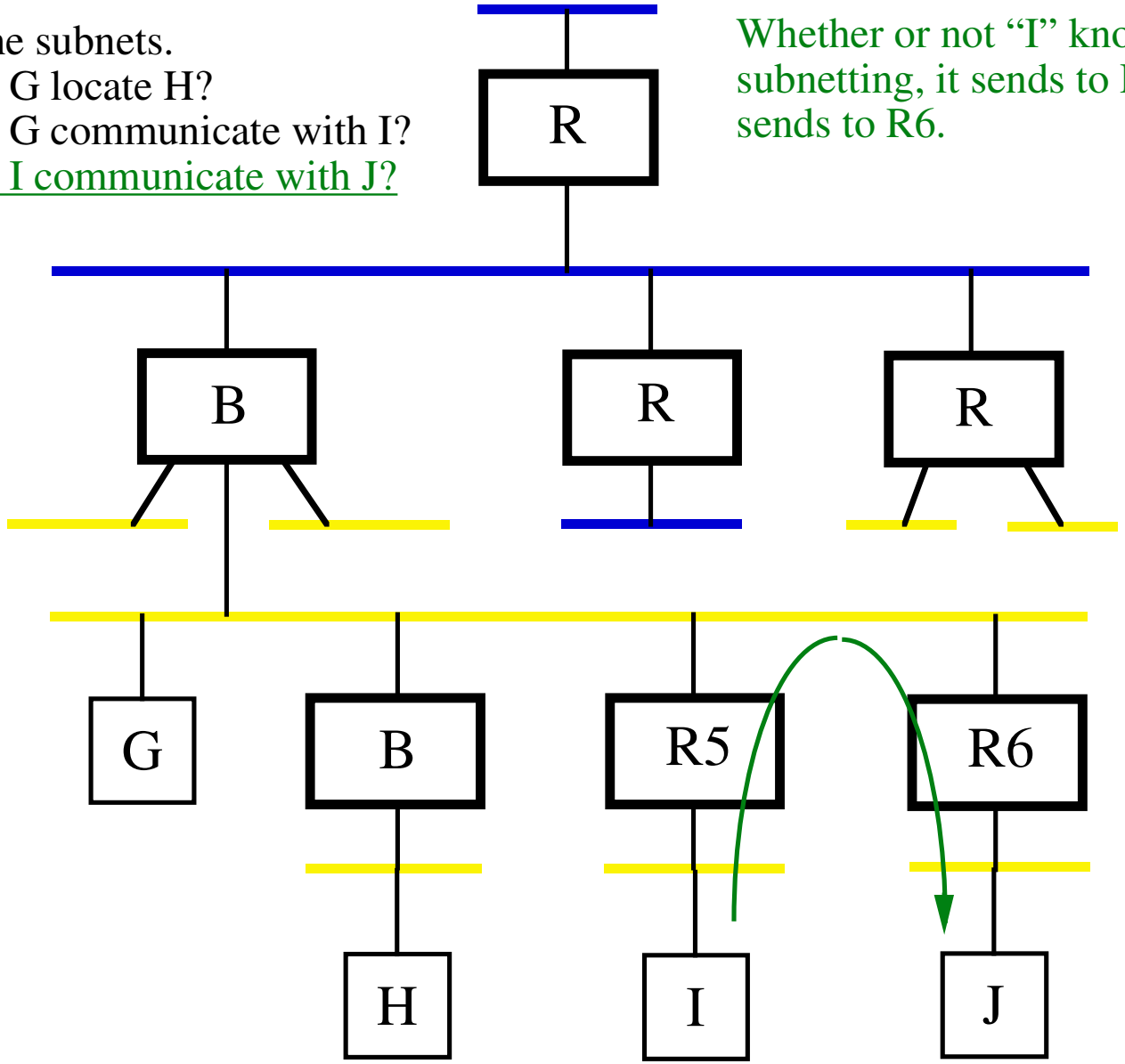


If G does not know about subnetting, it sends an ARP request which R5 answers on behalf of I. This is called "proxy ARP."

Example - A Larger Network

- Identify the subnets.
- How does G locate H?
- How does G communicate with I?
- How does I communicate with J?

Whether or not "I" knows about subnetting, it sends to R5, which sends to R6.



Higher Level Protocols - UDP

Services provided beyond IP

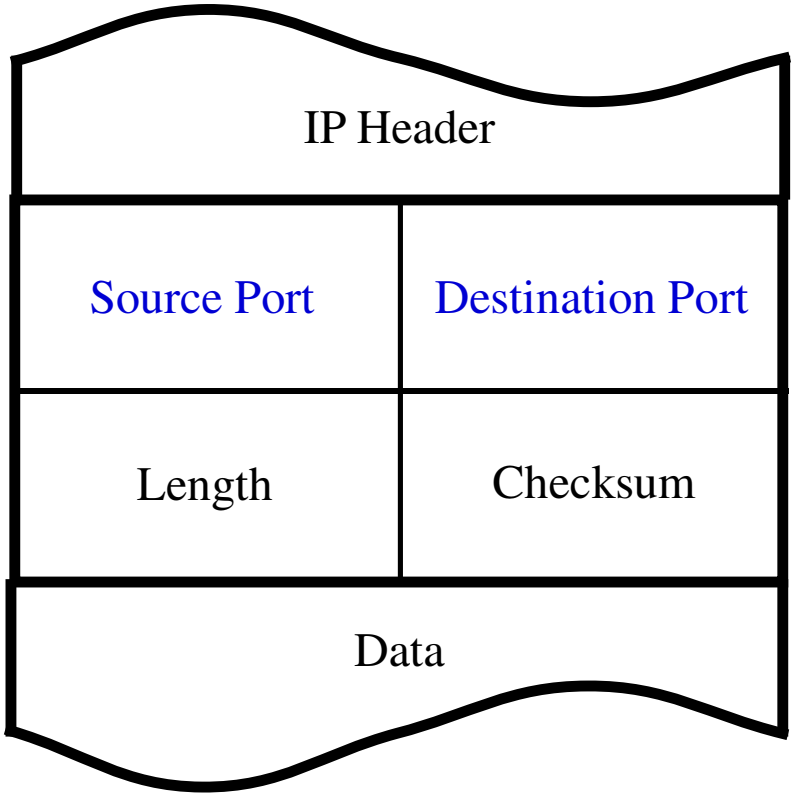
- Demultiplexing by source and destination *port numbers*
- Checksum protection of data

Additional header

- Eight bytes

Certain destination port numbers are reserved for *well-known services*.

Other protocols built on top of UDP include RPC, DNS, NTP, BOOTP.



Higher Level Protocols - TCP

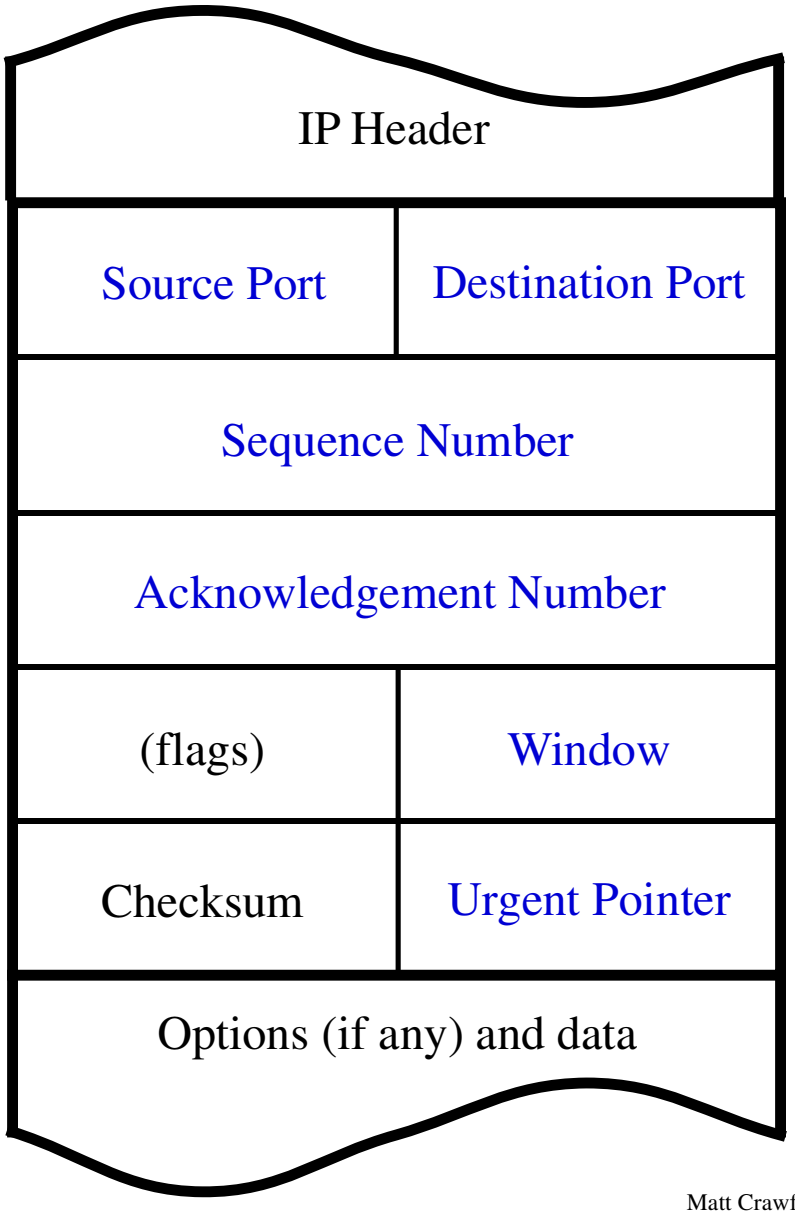
Services provided beyond IP

Reliable, flow-controlled, sequenced stream of bytes with provision for "out of band" data.

Demultiplexing by source and destination *port numbers*.

Additional header

20 bytes, plus options

Certain destination port numbers are reserved for *well known services*.

Higher level protocols build on TCP include telnet, X, ftp, SMTP.