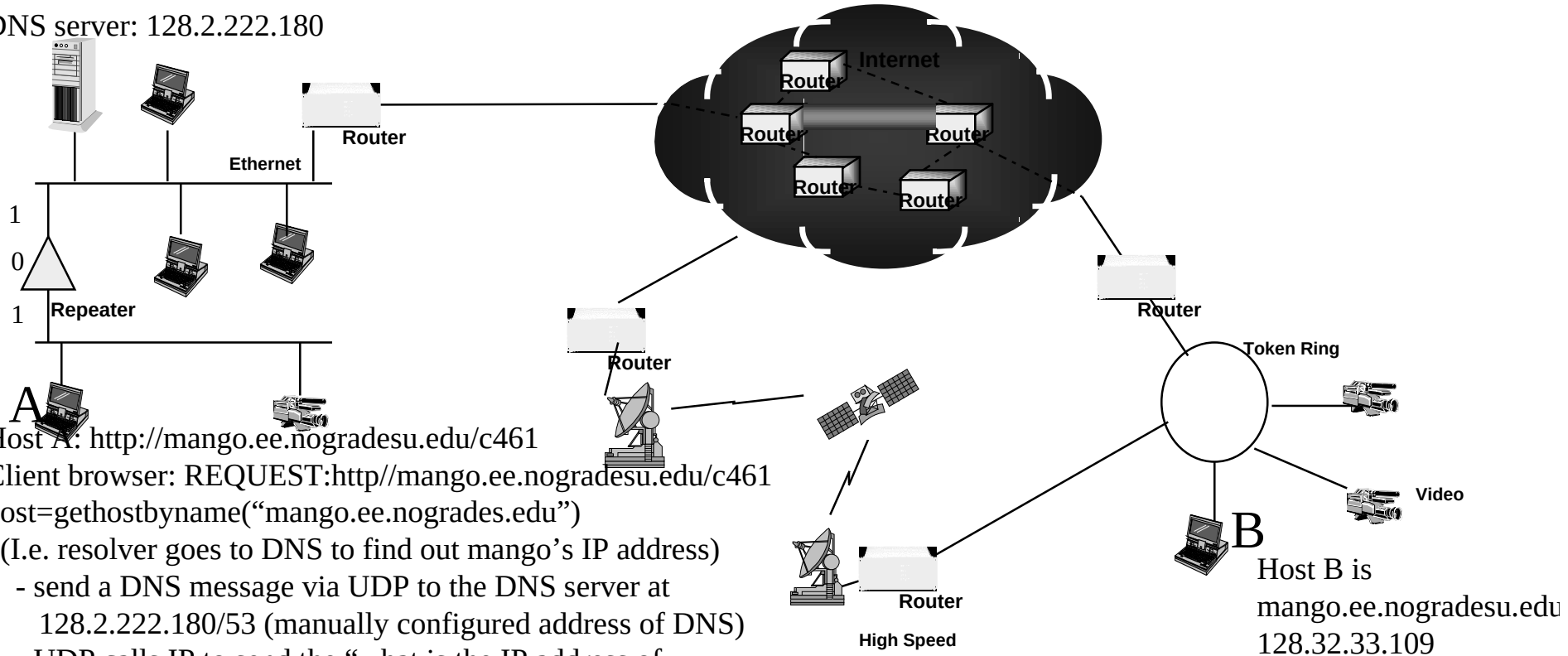
http:// mango.ee.nogradesu.edu/c461

DNS server: 128.2.222.180



Host A: http://mango.ee.nogradesu.edu/c461
Client browser: REQUEST:http//mango.ee.nogradesu.edu/c461
host=gethostbyname("mango.ee.nogrades.edu")
(I.e. resolver goes to DNS to find out mango's IP address)

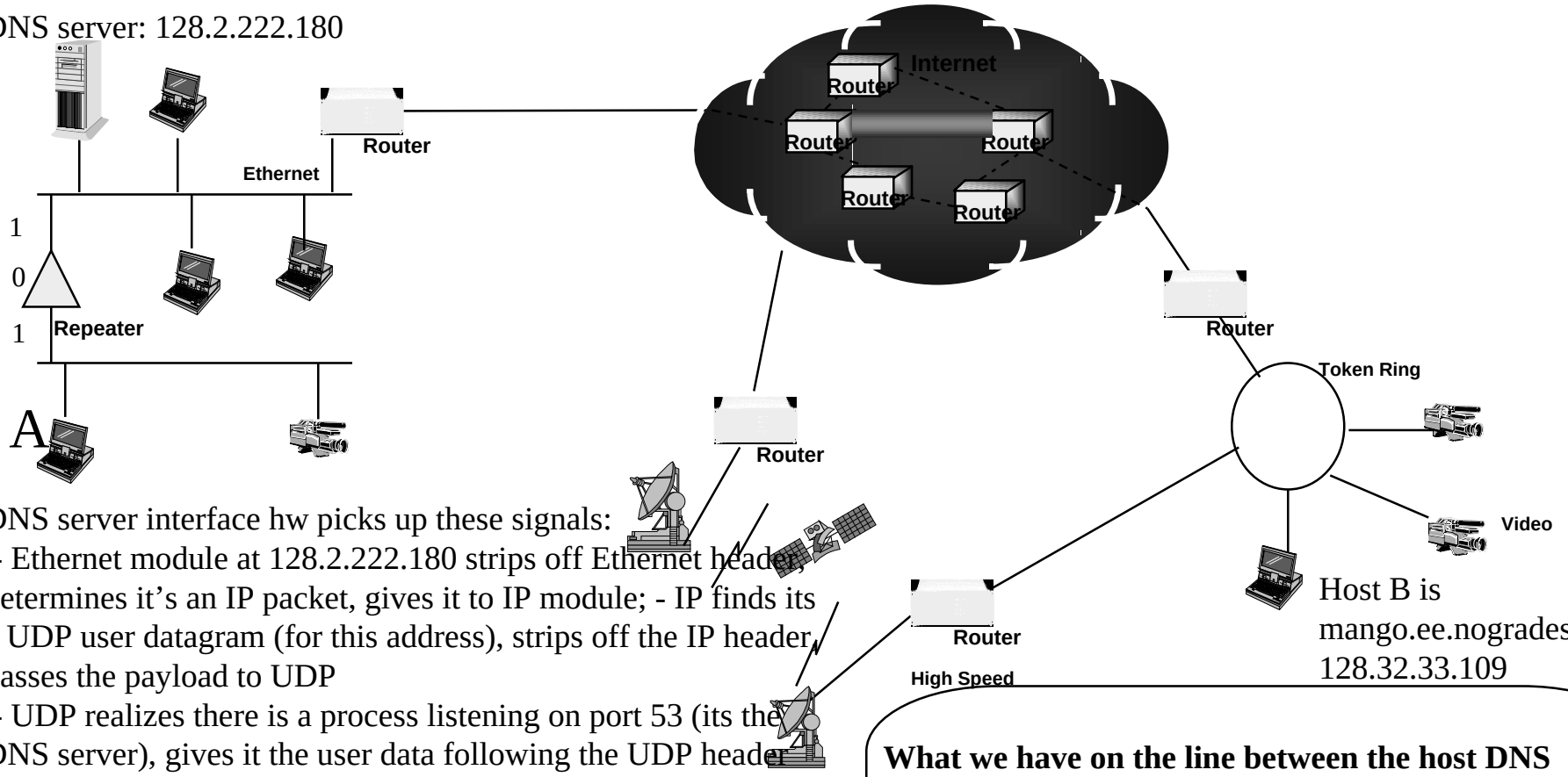
- send a DNS message via UDP to the DNS server at 128.2.222.180/53 (manually configured address of DNS)
- UDP calls IP to send the "what is the IP address of mango.ee.nogradesu.edu" DNS message to 128.2.222.180
- Turns out 128.2.222.180 on the same Ethernet
- call ARP to find Ethernet address of 128.2.222.180: it is 8:0:20:75:42:df
- IP puts UDP datagram into IP datagram payload and asks Ethernet driver to send this IP datagram to 8:0:20:75:42:df
- Ethernet driver puts the IP datagram in an Ethernet frame and broadcasts on the bus
- Interface hw transmits the bits as signals

Host B is mango.ee.nogradesu.edu
128.32.33.109

What we have on the line between the host A and the DNS server (I.e. on the Ethernet bus) is a DNS query message within a UDP user datagram enclosed in an IP datagram addressed to 128.2.222.180 (destination address in the IP header) inside an Ethernet frame with the destination address 8:0:20:75:42:df, all this as signals encoding the bits (that the repeater repeats bit by bit)

http://mango.ee.nogradesu.edu/c461

DNS server: 128.2.222.180



DNS server interface hw picks up these signals:
- Ethernet module at 128.2.222.180 strips off Ethernet header, determines it's an IP packet, gives it to IP module; - IP finds its a UDP user datagram (for this address), strips off the IP header, passes the payload to UDP
- UDP realizes there is a process listening on port 53 (its the DNS server), gives it the user data following the UDP header

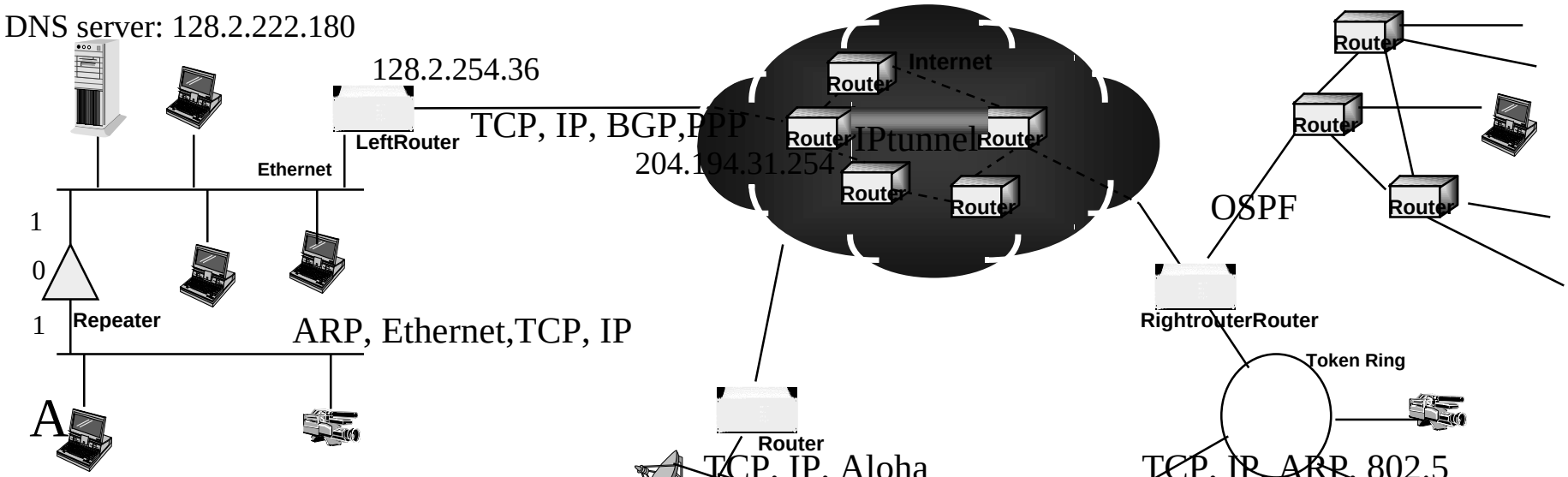
The user data consisted of a DNS query: " what is the IP address of mango.ee.nogradesu.edu"
DNS server: Asks UDP to send a response "It is 128.32.33.109" back the same way.

Host A: a web browser client process will get this response: it is going to set up a TCP connection to 128.32.33.109

Host B is mango.ee.nogradesu.edu 128.32.33.109

What we have on the line between the host DNS server and Host A is a message from the DNS server process "it's 128.32.33.109" sent as a UDP user datagram enclosed in an IP datagram addressed to host A (destination address in the IP header is host A's address) inside an Ethernet frame with destination address an Ethernet address of A

http://mango.ee.nogradesu.edu/c461



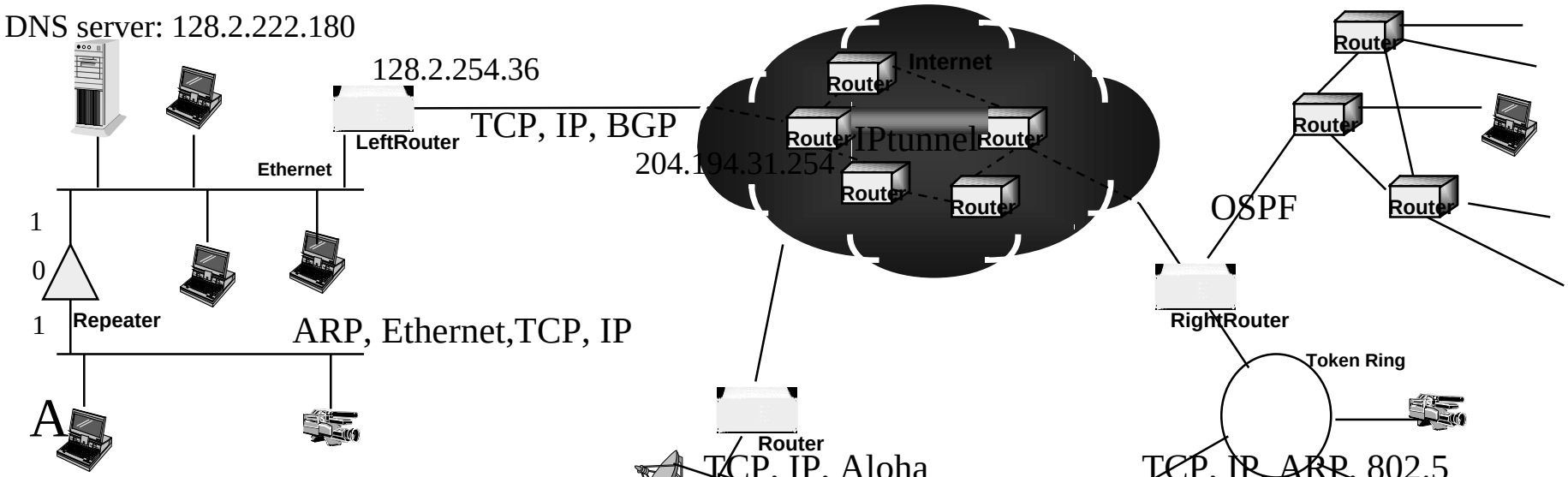
Host A:
web browser process: HTTP protocol starts to set up a TCP connection to 128.32.33.109
TCP calls IP to send a datagram to 128.32.33.109, with a TCP SYN in it for port 80 (well known web server port)
IP needs to find out where to send the packet; finds out A (itself) and 128.32.33.109 are not directly connected
IP on Host A sends this datagram to the default router Leftrouter 128.2.254.36 (needs to find the Ethernet address for it …)
IP gives this datagram to the Ethernet module, Ethernet bradcasts on the bus, 128.2.254.36 picks it up
128.2.254.36 decides this datagram is for a remote host
128.2.254.36 IP module consults the forwarding table
128.2.254.36 forwards this datagram to 204.194.31.254

Host B is mango.ee.nogradesu.edu 128.32.33.109

What we have on the pt-pt line between 128.2.254.36 and 204.194.31.254 is a TCP SYN segment, inside an IP datagram addressed from A (source address in the IP header) to 128.32.33.109 (destination address in the IP header) encapsulated in a PPP frame which is put onto the line as signals representing bits ...

Note that BGP, OSPF traffic is flowing independently all the time, building and updating the routing tables

http://mango.ee.nogradesu.edu/c461



Eventually, this datagram gets to the Rightrouter. The Rightrouter: Checks the IP datagram's destination address - it's on its own network! (Beyond the Rightrouter is an AS which uses OSPF to provide routing tables for its hosts). The rightrouter:

- uses ARP to find the Token Ring address for 128.32.33.109
- encloses the datagram into an 802.5 frame addressed to Host B and broadcasts on the ring
- Host B picks up the 802.5 frame (it recognizes its 48-bit address in the frame header), sees its an IP datagram and gives it to the IP module, the IP module sees that next protocol is TCP, gives it to TCP … to port 80 … all is OK … web server process accepts and TCP sends an ACK back to A - work the rest of the story for yourself.

Host B is mango.ee.nogradesu.edu 128.32.33.109

What is you had a bridge somewhere?
What if you had to tunnel IPv6 through an IPv4 network?
What if you lost an IP datagram (or a piece of it)?
What if you are typing a "hello" during the session , and all you end up with is "hol" given to the remote TCP by IP?