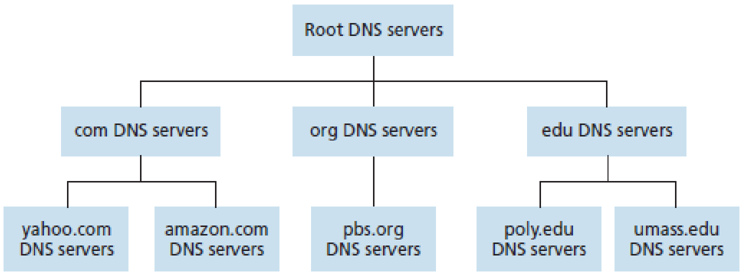
计算机网络期末复习纲要

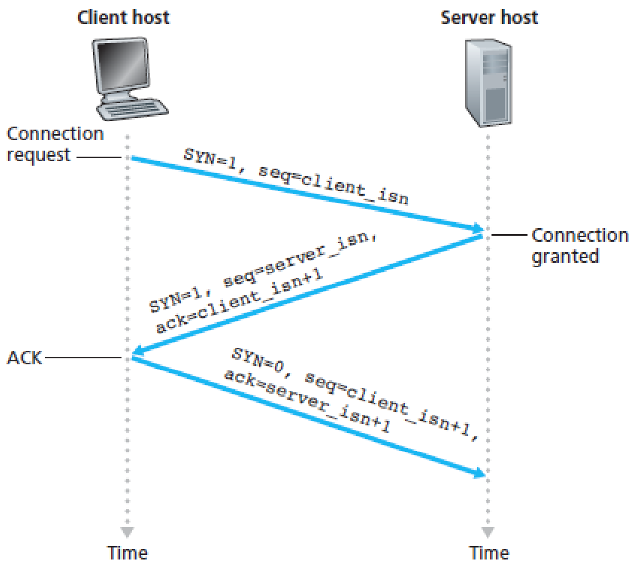
考试时间：2013 年 7 月 11 日 14:00−16:00

一、选择题（共 30 题，每题 1 分）

1. 局域网（LAN）:概念，划分。（11 页，297 页）
2. 域名解析服务(DNS）三层结构:根(root)DNS 服务器,顶级域(top-level domain)服务器,权威(authoritative）DNS 服务器。DNS 记录结构。离你最近的 DNS 服务器上存放着什么？（87 页，90 页）



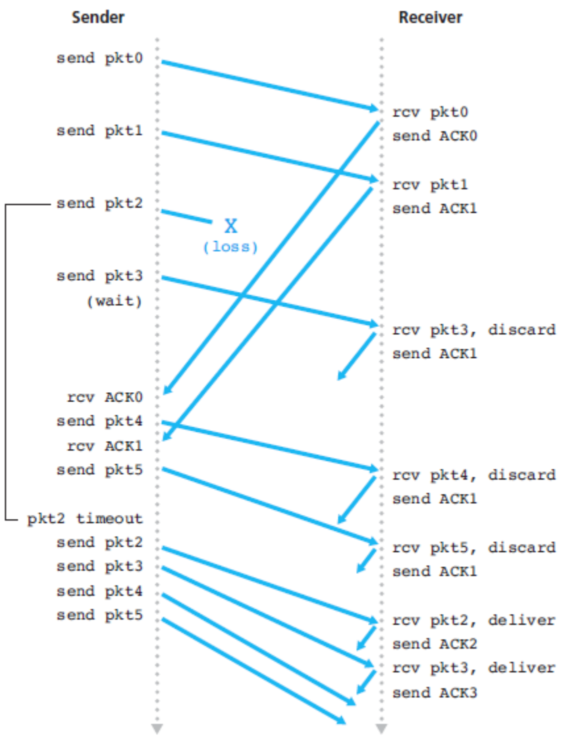
3. TCP 连接的建立：三次握手（three-way handshaking），SYN 和 ACK 标志位，顺序号和确认号。（169 页）



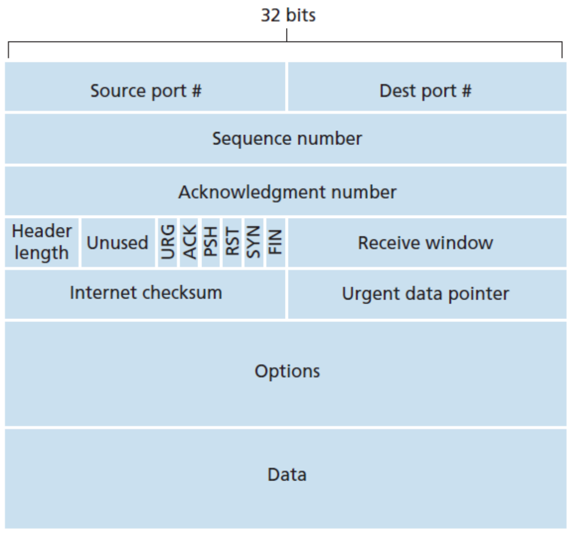
4. SMTP：推协议（push protocol），邮件服务器间投递邮件，不使用中间邮件服务器，25 端口，只能用 7 位 ASCII 码表示。（77 页）
5. TCP/IP 协议族（TCP/IP protocol suite）

层次	协议
应用层（application layer）	HTTP SMTP FTP
传输层（transport layer）	TCP UDP
网络层（network layer）	IP RIP OSPF BGP
数据链路层（data link layer）	ARP PPP

6. 流水线（pipelining）：允许发送方发送多个分组而无需等待确认，但必须增加序号范围且缓存分组。（146 页）
7. URL 的格式（format）：<协议>://<主机名>:<端口>/<路径> （谢希仁第五版教材 238 页）
8. C/S 结构与 P2P 结构（7 页，53 页）
9. 分用（demultiplexing）和复用（multiplexing）（128 页）
10. 后退 N 帧协议（go-back-N）：滑动窗口（sliding-window）协议。累积确认（cumulative acknowledgement）：接收方对目前收到的最后一个正确有序的分组发出确认，丢弃所有乱序分组。如果超时，发送方将重传所有已发送但还未被确认的分组。接收窗口大小为 1。（147 页）



11. 端口号(port number)的范围:0-65535,周知端口号(well-known port number)0-1023,登记端口号(registered port）1024-49151，短暂端口号(ephemeral port)49152-65535。(129 页，谢希仁第五版教材 183 页)
12. TCP 顺序号（sequence number）和确认号(acknowledge number)：顺序号是该报文段首字节的字节流编号，确认号是期望收到的下一个报文段的顺序号。(157 页）
13. 发送窗口：受制于流量控制和拥塞控制，拥塞控制中的拥塞窗口（congestion window）和阈值（threshold）（147 页，180 页，183 页）
14. TCP 报文段结构，各字段的名称、作用和值。TCP 报文段首部的长度范围：20-60 字节。(157 页）

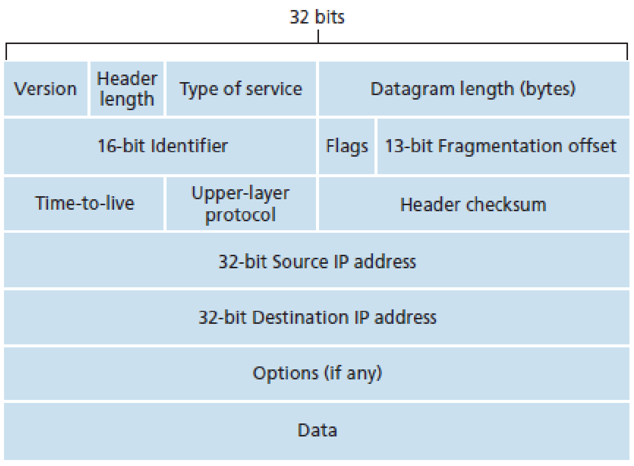


15. 拥塞控制（congestion control)：慢启动（slow start)，拥塞避免（congestion avoidance)，加性增乘性减（AIMD)。(180-184 页）
16. 路由表的查询：最长前缀匹配。(207 页）
17. 网络流量（network traffic）和吞吐量（throughput)。(28 页，174 页，184 页）
18. RIP 协议：基于距离向量（DV）算法，以跳数（hop）作为代价度量（metrics)。(242 页，251 页）
19. OSPF 协议：基于链路状态（LS）算法。(239 页，253 页）
20. BGP 协议：自治系统（AS）间的选路。(255 页）
21. 差错控制方法（error detection method)：奇偶校验（parity check)，校验和（checksum)，循环冗余校验

（CRC），各自的特点和计算方法。（284 页）
22. 循环冗余校验的计算。（287 页）
23. 随机接入（random access）协议：时隙 ALOHA，ALOHA，CSMA，各自的特点。（292 页）
24. ARP 的过程：广播（broadcast）发送 ARP 分组，单播（unicast）发送响应。（301 页）
25. IP 地址与 MAC 地址的映射（mapping）：NAT，ARP，如何获得下一跳的地址？（228 页，300 页）
26. MAC 地址的格式，二进制到十六进制的转换。
27. 特殊的 IP 地址：

全网网络地址	0.0.0.0
广播地址	255.255.255.255
本机回送（loopback）地址	127.0.0.1
组播地址	224.0.0.1
私网地址	10.0.0.0---10.255.255.255　172.16.0.0---172.31.255.255　192.168.0.0---192.168.255.255

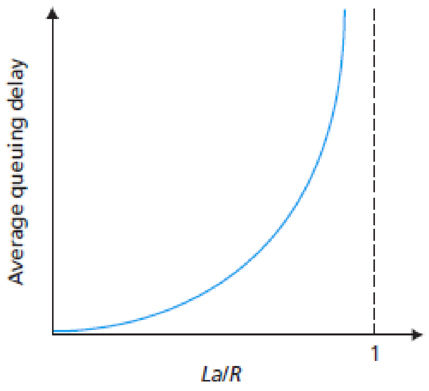
28. IP 数据报格式，各字段的名称、作用和值。哪些字段会随着数据报的转发而改变？TTL，checksum。（216 页）



29. TCP 平均吞吐量=0.75*W/RTT。（184 页）
30. 二进制指数退避算法（binary exponential backoff algorithm）（308 页）

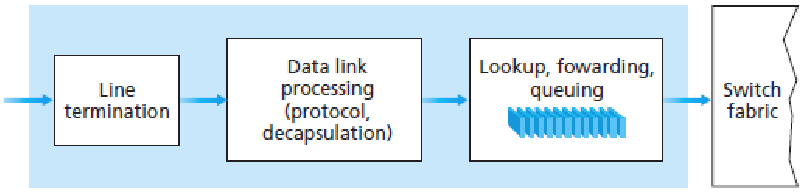
二、填空题（共 10 题，每题 1 分）

31. 协议的定义：protocols define format, order of massages sent and received among network entities, and actions taken on massage transmission, receipt.（6 页）
32. 流量强度（traffic intensity）= La/R R=链路带宽　L=分组大小　a=平均分组到达速率（25 页）



33. FTP：控制连接 21 端口，数据连接 20 端口，控制信息带外（out of band）传送（数据和控制信息不使用同一 TCP 连接），保留用户状态（state）信息，每个数据连接只能传送一个文件（非持久 non-persistent 连接）。（73 页）
34. POP3 的过程和工作模式：下载并删除（IMAP 下载并保留），USER PASS LIST RETR DELE QUIT。（82 页）

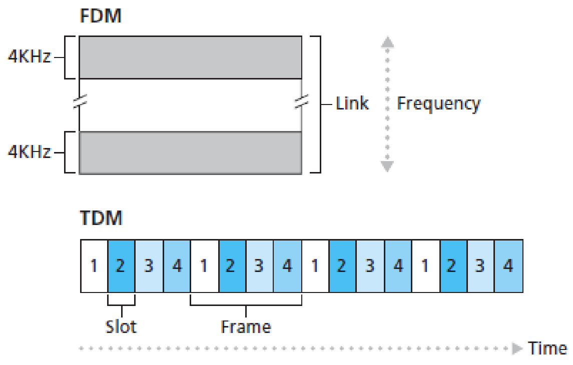
35. TCP 流量控制公式：LastByteRcvd – LastByteRead ≤ RcvBuffer
    RcvWindow= RcvBuffer-[LastByteRcvd - LastByteRead]
    LastByteSent – LastByteAcked ≤ RcvWindow （167-168 页）
36. 对于 SR 协议而言，窗口长度必须小于等于序号空间大小的一半。The window size must be less than or equal to half the size of the sequence number space for SR protocols.（152 页）
37. 路由器的输入端口：数据链路处理（协议，拆封），查找、转发、排队。虽然转发表是由选路处理器计算的，但通常一份转发表的影子拷贝会被存放在每个输入端口，而且会被更新。（210 页）


Line termination
Data link processing (protocol, decapsulation)
Lookup, fowarding, queuing
Switch fabric

38. 因特网的路由方式。
39. 链路类型：点对点链路和广播链路（中文版书 288 页）。
40. MAC 地址。

三、计算题（共 2 题，共 15 分）

41. 1）传输延时（transmission delay）与传播延时（propagation delay）：传输延时指分组从设备发送到链路上所经历的时间，等于分组大小/网络带宽（L/R）。传播延时等于分组从物理链路一端发送到另一端所经历的时间，等于链路长度/传播速度（d/s）。（23 页）
    2）分组交换（packet switch）：存储转发（store and forward）是指在交换机能够开始向输出链路传输该分组的第一个比特之前，必须接收到整个分组。（18 页）
    3）时分多路复用（TDM）与频分多路复用（FDM）。（17 页）



Example:
For the following questions suppose we are sending a 30-Mbit MP3 file from a source host to a destination host. All the links in the path between source and destination have a transmission rate of 10 Mbps. Assume that the propagation speed is 2X10^8 meters/sec and the distance between source and destination is 10000 km. Initially suppose there is only one link between the source and the destination. Also suppose that the entire MP3 file is sent as one packet.

Question 1
The transmission delay is:
Correct: 3 seconds

Question 2
Referring to the above question, the end-to-end delay (transmission delay plus propagation delay) is

Correct: 3.05 seconds

Question 3

Referring to the above question, how many bits will the source have transmitted when the first bit arrives at the destination.

Correct: 500,000 bits

Question 4

Now suppose there are two links between source and destination, with one router connecting the two links. Each link is 5,000 km long. Again suppose the MP3 file is sent as one packet. Suppose there is no congestion, so that the packet is transmitted onto the second link as soon as the router receives the entire packet. The end-to-end delay is

Correct: 6.05 seconds

Question 5

Now suppose that the MP3 file is broken into 3 packets, each of 10 Mbits. Ignore headers that may be added to these packets. Also ignore router processing delays. Assuming store and forward packet switching at the router, the total delay is

Correct: 4.05 seconds

Question 6

Now suppose there is only one link between source and destination, and there are 10 TDM channels in the link. The MP3 file is sent over one of the channels. The end-to-end delay is

Correct: 30.05 seconds

Question 7

Now suppose there is only one link between source and destination, and there are 10 FDM channels in the link. The MP3 file is sent over one of the channels. The end-to-end delay is

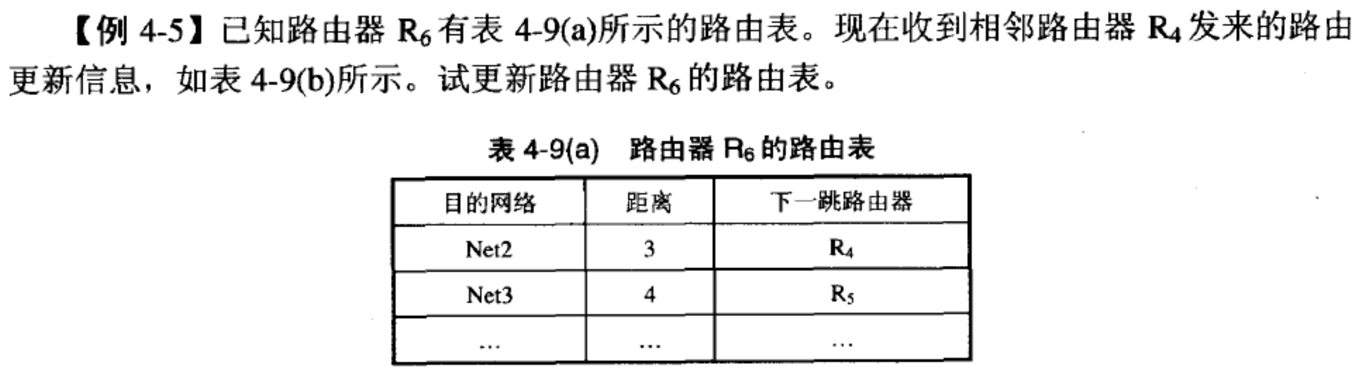
Correct: 30.05 seconds

----

42. 子网划分（subnetting）

四、简答题（共 45 分）

43. RIP 路由表的更新（谢希仁第五版教材 148 页）



【例 4-5】已知路由器 R_6 有表 4-9(a)所示的路由表。现在收到相邻路由器 R_4 发来的路由更新信息，如表 4-9(b)所示。试更新路由器 R_6 的路由表。

表 4-9(a)　路由器 R_6 的路由表

目的网络	距离	下一跳路由器
Net2	3	R_4
Net3	4	R_5
…	…	…

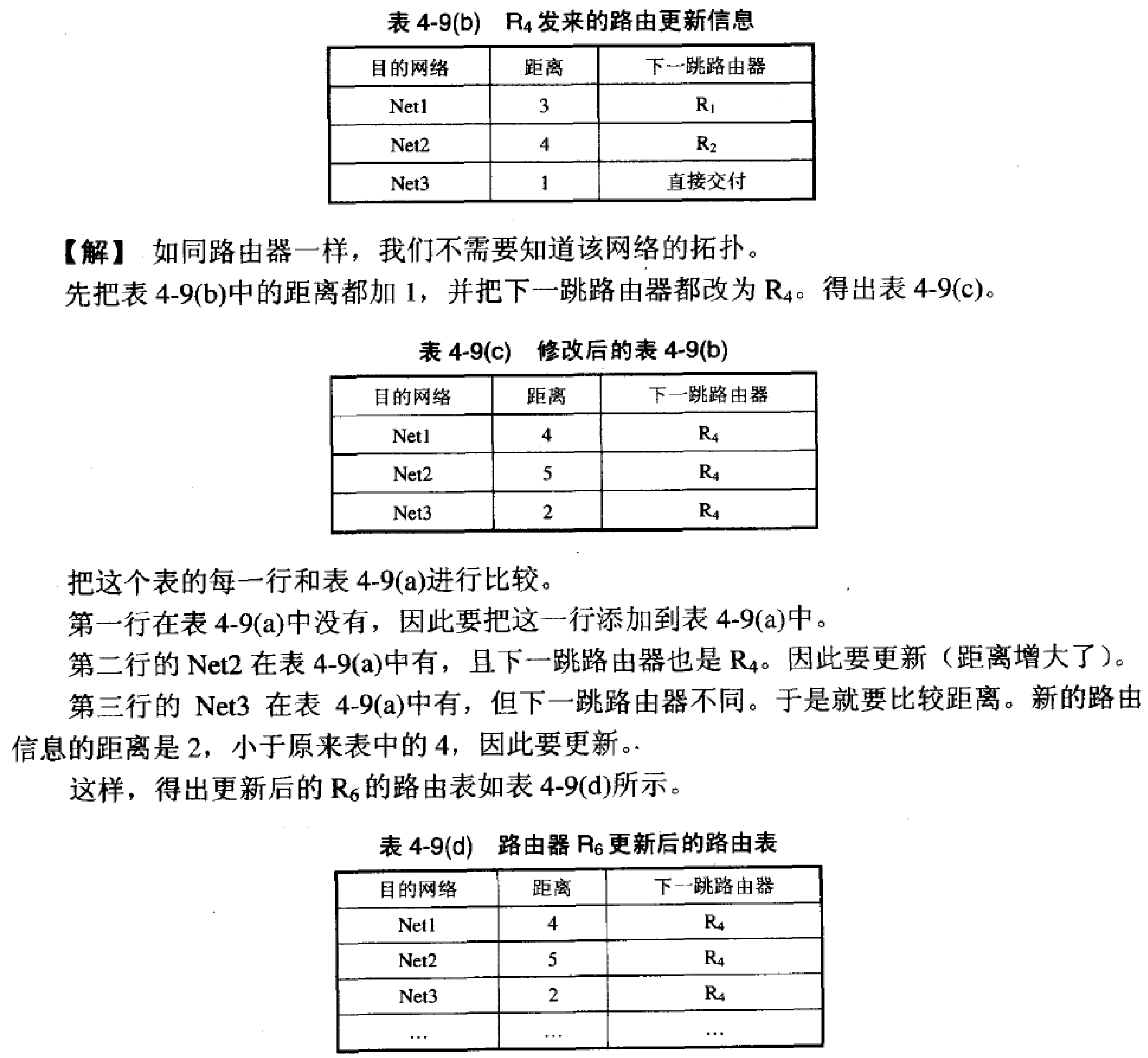


表 4-9(b)　R_4 发来的路由更新信息

目的网络	距离	下一跳路由器
Net1	3	R_1
Net2	4	R_2
Net3	1	直接交付

【解】 如同路由器一样，我们不需要知道该网络的拓扑。

先把表 4-9(b)中的距离都加 1，并把下一跳路由器都改为 R_4。得出表 4-9(c)。

表 4-9(c)　修改后的表 4-9(b)

目的网络	距离	下一跳路由器
Net1	4	R_4
Net2	5	R_4
Net3	2	R_4

把这个表的每一行和表 4-9(a)进行比较。

第一行在表 4-9(a)中没有，因此要把这一行添加到表 4-9(a)中。

第二行的 Net2 在表 4-9(a)中有，且下一跳路由器也是 R_4。因此要更新（距离增大了）。

第三行的 Net3 在表 4-9(a)中有，但下一跳路由器不同。于是就要比较距离。新的路由信息的距离是 2，小于原来表中的 4，因此要更新。

这样，得出更新后的 R_6 的路由表如表 4-9(d)所示。

表 4-9(d)　路由器 R_6 更新后的路由表

目的网络	距离	下一跳路由器
Net1	4	R_4
Net2	5	R_4
Net3	2	R_4
…	…	…

44. 一个体系结构（one architecture）

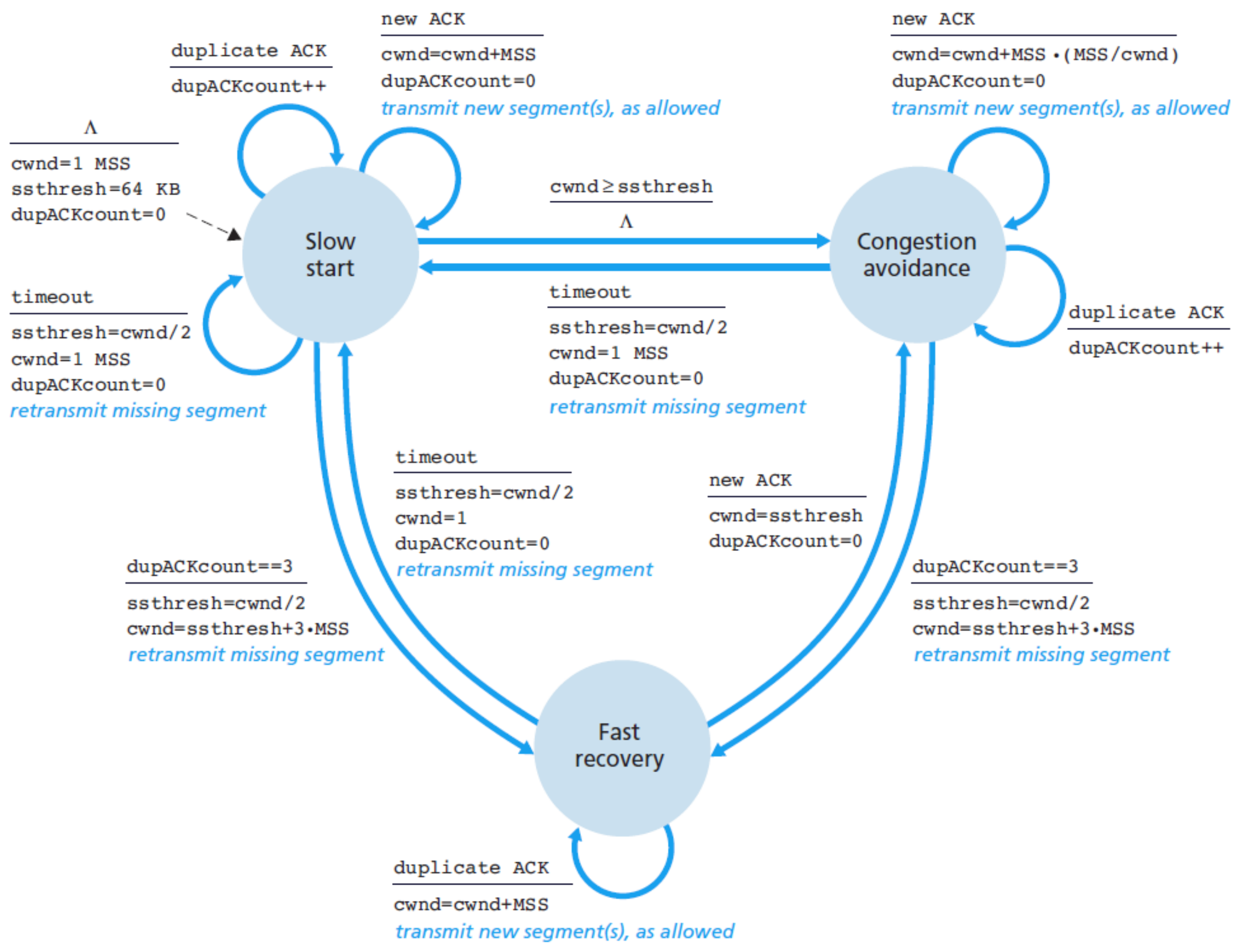
两个参考模型（two reference model）：ISO/OSI 参考模型，TCP/IP 参考模型（32 页）

三个地址（three address）：URL，IP 地址，MAC 地址

四种控制（four control）：差错控制（error control），流量控制（flow control），拥塞控制（congestion control），介质访问控制（medium access control）

45. 自治系统（AS）间的路由：外部 BGP 会话（eBGP session）和内部 BGP 会话（iBGP session）。（256 页，275 页 29 题）

46. 拥塞控制（congestion control）的有限状态自动机（FSM）。



47. When you enter an URL in your browser and press ENTER, what happens? Hints: DHCP, UDP, IP, Ethernet, DNS, ARP, routing, TCP, HTTP

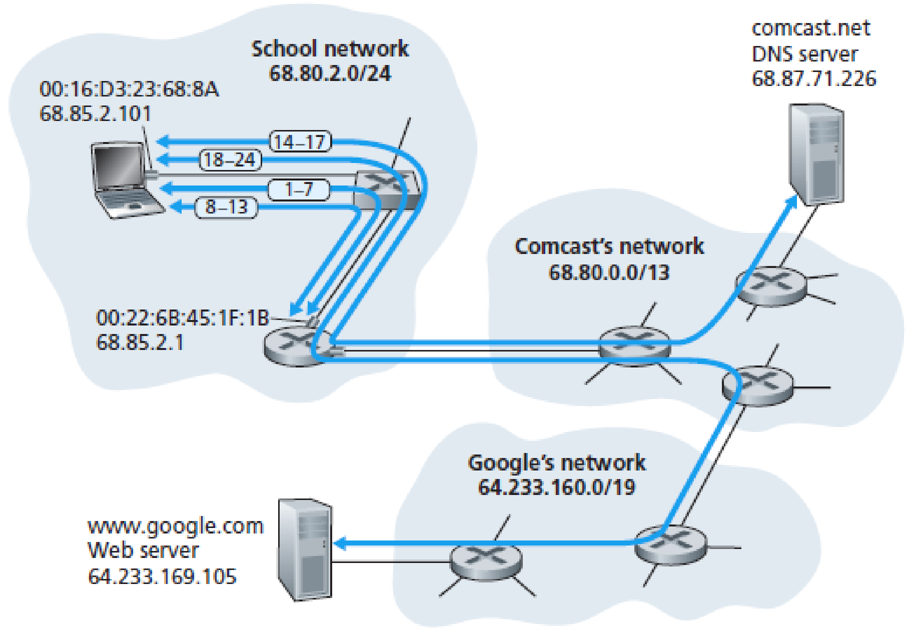


Figure 5.32 ♦ A day in the life of a Web page request: network setting and actions

Retrospective: A Day in the Life of a Web Page Request

Now that we've covered the link layer in this chapter, and the network, transport and application layers in earlier chapters, our journey down the protocol stack is complete! In the very beginning of this book (Section 1.1), we wrote "much of this book is concerned with computer network protocols," and in the first five chapters, we've certainly seen that this is indeed the case! Before heading into the topical chapters in second part of this book, we'd like to wrap up our journey down the protocol stack by taking an integrated, holistic view of the protocols we've learned about so far. One way then to take this "big picture" view is to identify the many (many!) protocols that are involved in satisfying even the simplest request: downloading a web page. Figure 5.32 illustrates our setting: a student, Bob, connects a laptop to his school's Ethernet switch and downloads a web page (say the home page of www.google.com). As we now know, there's a lot going on "under the hood" to satisfy this seemingly simple request. A Wireshark lab at the end of this chapter examines trace files containing a number of the packets involved in similar scenarios in more detail.

1. Getting Started: DHCP, UDP, IP and Ethernet

Let's suppose that Bob boots up his laptop and then connects it to an Ethernet cable connected to the school's Ethernet switch, which in turn is connected to the school's router, as shown in Figure 5.32. The school's router is connected to an ISP, in this example, comcast.net. In this example, comcast.net is providing the DNS service for the school; thus, the DNS server resides in the Comcast network rather than the school network. We'll assume that the DHCP server is running within the router, as is often the case.

When Bob first connects his laptop to the network, he can't do anything (e.g., download a Web page) without an IP address. Thus, the first network-related action taken by Bob's laptop is to run the DHCP protocol to obtain an IP address, as well as other information, from the local DHCP server:

1. The operating system on Bob's laptop creates a DHCP request message (Section 4.4.2) and puts this message within a UDP segment (Section 3.3) with destination port 67 (DHCP server) and source port 68 (DHCP client). The UDP segment is then placed within an IP datagram (Section 4.4.1) with a broadcast IP destination address (255.255.255.255) and a source IP address of 0.0.0.0, since Bob's laptop doesn't yet have an IP address.
2. The IP datagram containing the DHCP request message is then placed within an Ethernet frame (Section 5.5). The Ethernet frame has a destination MAC addresses of FF:FF:FF:FF:FF:FF so that the frame will be broadcast to all devices connected to the switch (hopefully including a DHCP server); the frame's source MAC address is that of Bob's laptop, 00:16:D3:23:68:8A.
3. The broadcast Ethernet frame containing the DHCP request is the first frame sent by Bob's laptop to the Ethernet switch. The switch broadcasts the incoming frame on all outgoing ports, including the port connected to the router.
4. The router receives the broadcast Ethernet frame containing the DHCP request on its interface with MAC address 00:22:6B:45:1F:1B and the IP datagram is extracted from the Ethernet frame. The datagram's broadcast IP destination address indicates that this IP datagram should be processed by upper layer protocols at this node, so the datagram's payload (a UDP segment) is thus demultiplexed (Section 3.2) up to UDP, and the DHCP request message is extracted from the UDP segment. The DHCP server now has the DHCP request message.
5. Let's suppose that the DHCP server running within the router can allocate IP addresses in the CIDR (Section 4.4.2) block 68.85.2.0/24. In this example, all IP addresses used within the school are thus within Comcast's address block. Let's suppose the DHCP server allocates address 68.85.2.101 to Bob's laptop. The DHCP server creates a DHCP ACK message (Section 4.4.2) containing this IP address, as well as the IP address of the DNS server (68.87.71.226), the IP address for the default gateway router (68.85.2.1), and the subnet block (68.85.2.0/24) (equivalently, the "network mask"). The DHCP message is put inside a UDP segment, which is put inside an IP datagram, which is put inside an Ethernet frame. The Ethernet frame has a source MAC address of the router's interface to the home network (00:22:6B:45:1F:1B) and a

destination MAC address of Bob’s laptop (00:16:D3:23:68:8A).

6. The Ethernet frame containing the DHCP ACK is sent (unicast) by the router to the switch. Because the switch is self-learning (Section 5.6.2) and previously received an Ethernet frame (containing the DHCP request) from Bob’s laptop, the switch knows to forward a frame addressed to 00:16:D3:23:68:8A only to the output port leading to Bob’s laptop.
7. Bob’s laptop receives the Ethernet frame containing the DHCP ACK, extracts the IP datagram from the Ethernet frame, extracts the UDP segment from the IP datagram, and extracts the DHCP ACK message from the UDP segment. Bob’s DHCP client then records its IP address and the IP address of its DNS server. It also installs the address of the default gateway into its IP forwarding table (Section 4.1). Bob’s laptop will send all datagrams with destination address outside of its subnet 68.85.2.0/24 to the default gateway. At this point, Bob’s laptop has initialized its networking components and is ready to begin processing the Web page fetch. (Note that only the last two DHCP steps of the four presented in Chapter 4 are actually necessary.)

2. Still Getting Started: DNS and ARP

When Bob types the URL for www.google.com into his Web browser, he begins the long chain of events that will eventually result in Google’s home page being displayed by his Web browser. Bob’s Web browser begins the process by creating a TCP socket (Section 2.7) that will be used to send the HTTP request (Section 2.2) to www.google.com. In order to create the socket, Bob’s laptop will need to know the IP address of www.google.com. We learned in Section 2.5, that the DNS protocol is used to provide this name-to-IP-address translation service.

8. The operating system on Bob’s laptop thus creates a DNS query message (Section 2.5.3), putting the string “www.google.com” in the question section of the DNS message. This DNS message is then placed within a UDP segment with a destination port of 53 (DNS server). The UDP segment is then placed within an IP datagram with an IP destination address of 68.87.71.226 (the address of the DNS server returned in the DHCP ACK in step 5) and a source IP address of 68.85.2.101.
9. Bob’s laptop then places the datagram containing the DNS query message in an Ethernet frame. This frame will be sent (addressed, at the link layer) to the gateway router in Bob’s school’s network. However, even though Bob’s laptop knows the IP address of the school’s gateway router (68.85.2.1) via the DHCP ACK message in step 5 above, it doesn’t know the gateway router’s MAC address. In order to obtain the MAC address of the gateway router, Bob’s laptop will need to use the ARP protocol (Section 5.4.1).
10. Bob’s laptop creates an ARP query message with a target IP address of 68.85.2.1 (the default gateway), places the ARP message within an Ethernet frame with a broadcast destination address (FF:FF:FF:FF:FF:FF) and sends the Ethernet frame to the switch, which delivers the frame to all connected devices, including the gateway router.
11. The gateway router receives the frame containing the ARP query message on the interface to the school network, and finds that the target IP address of 68.85.2.1 in the ARP message matches the IP address of its interface. The gateway router thus prepares an ARP reply, indicating that its MAC address of 00:22:6B:45:1F:1B corresponds to IP address 68.85.2.1. It places the ARP reply message in an Ethernet frame, with a destination address of 00:16:D3:23:68:8A (Bob’s laptop) and sends the frame to the switch, which delivers the frame to Bob’s laptop.
12. Bob’s laptop receives the frame containing the ARP reply message and extracts the MAC address of the gateway router (00:22:6B:45:1F:1B) from the ARP reply message.
13. Bob’s laptop can now (finally!) address the Ethernet frame containing the DNS query to the gateway router’s MAC address. Note that the IP datagram in this frame has an IP destination address of 68.87.71.226 (the DNS server), while the frame has a destination address of 00:22:6B:45:1F:1B (the gateway router). Bob’s laptop sends this frame to the switch, which delivers the frame to the gateway router.

3. Still Getting Started: Intra-Domain Routing to the DNS Server

14. The gateway router receives the frame and extracts the IP datagram containing the DNS query. The router looks up the

destination address of this datagram (68.87.71.226) and determines from its forwarding table that the datagram should be sent to the leftmost router in the Comcast network in Figure 5.32. The IP datagram is placed inside a link-layer frame appropriate for the link connecting the school's router to the leftmost Comcast router and the frame is sent over this link.

15. The leftmost router in the Comcast network receives the frame, extracts the IP datagram, examines the datagram's destination address (68.87.71.226) and determines the outgoing interface on which to forward the datagram towards the DNS server from its forwarding table, which has been filled in by Comcast's intra-domain protocol (such as RIP, OSPF or IS-IS, Section 4.6) as well as the Internet's inter-domain protocol, BGP.
16. Eventually the IP datagram containing the DNS query arrives at the DNS server. The DNS server extracts the DNS query message, looks up the name www.google.com in its DNS database (Section 2.5), and finds the DNS resource record that contains the IP address (64.233.169.105) for www.google.com. (assuming that it is currently cached in the DNS server). Recall that this cached data originated in the authoritative DNS server (Section 2.5.2) for google.com. The DNS server forms a DNS reply message containing this hostname-to-IP address mapping, and places the DNS reply message in a UDP segment, and the segment within an IP datagram addressed to Bob's laptop (68.85.2.101). This datagram will be forwarded back through the Comcast network to the school's router and from there, via the Ethernet switch to Bob's laptop.
17. Bob's laptop extracts the IP address of the server www.google.com from the DNS message. Finally, after a lot of work, Bob's laptop is now ready to contact the www.google.com server!

4. Web Client-Server Interaction: TCP and HTTP

18. Now that Bob's laptop has the IP address of www.google.com, it can create the TCP socket (Section 2.7) that will be used to send the HTTP GET message (Section 2.2.3) to www.google.com. When Bob creates the TCP socket, the TCP in Bob's laptop must first perform a three-way handshake (Section 3.5.6) with the TCP in www.google.com. Bob's laptop thus first creates a TCP SYN segment with destination port 80 (for HTTP), places the TCP segment inside an IP datagram with a destination IP address of 64.233.169.105 (www.google.com), places the datagram inside a frame with a destination MAC address of 00:22:6B:45:1F:1B (the gateway router) and sends the frame to the switch.
19. The routers in the school network, Comcast's network, and Google's network forward the datagram containing the TCP SYN towards www.google.com, using the forwarding table in each router, as in steps 14–16 above. Recall that the router forwarding table entries governing forwarding of packets over the inter-domain link between the Comcast and Google networks are determined by the BGP protocol (Section 4.6.3).
20. Eventually, the datagram containing the TCP SYN arrives at www.google.com. The TCP SYN message is extracted from the datagram and demultiplexed to the welcome socket associated with port 80. A connection socket (Section 2.7) is created for the TCP connection between the Google HTTP server and Bob's laptop. ATCP SYNACK (Section 3.5.6) segment is generated, placed inside a datagram addressed to Bob's laptop, and finally placed inside a link-layer frame appropriate for the link connecting www.google.com to its first-hop router.
21. The datagram containing the TCP SYNACK segment is forwarded through the Google, Comcast, and school networks, eventually arriving at the Ethernet card in Bob's laptop. The datagram is demultiplexed within the operating system to the TCP socket created in step 18, which enters the connected state.
22. With the socket on Bob's laptop now (finally!) ready to send bytes to www.google.com, Bob's browser creates the HTTP GET message (Section 2.2.3) containing the URL to be fetched. The HTTP GET message is then written into the socket, with the GET message becoming the payload of a TCP segment. The TCP segment is placed in a datagram and sent and delivered to www.google.com as in steps 18–20 above.
23. The HTTP server at www.google.com reads the HTTP GET message from the TCP socket, creates an HTTP response message (Section 2.2), places the requested Web page content in the body of the HTTP response message, and sends the message into the TCP socket.
24. The datagram containing the HTTP reply message is forwarded through the Google, Comcast, and school networks, and

arrives at Bob’s laptop. Bob’s Web browser program reads the HTTP response from the socket, extracts the html for the Web page from the body of the HTTP response, and finally (finally!) displays the Web page!

Our scenario above has covered a lot of networking ground! If you’ve understood most or all of the above example, then you’ve also covered a lot of ground since you first read Section 1.1, where we wrote “much of this book is concerned with computer network protocols” and you may have wondered what a protocol actually was! As detailed as the above example might seem, we’ve omitted a number of possible additional protocols (e.g., NAT running in the school’s gateway router, wireless access to the school’s network, security protocols for accessing the school network or encrypting segments or datagrams, network management protocols), and considerations (Web caching, the DNS hierarchy) that one would encounter in the public Internet. We’ll cover a number of these topics and more in the second part of this book.

Lastly, we note that our example above was an integrated and holistic, but also very “nuts and bolts,” view of many of the protocols that we’ve studied in the first part of this book. The example focused more on the “how” than the “why.” For a broader, more reflective view on the design of network protocols in general, see [Clark 1988, RFC 5218].

（Source: Computer Networking, A Top-Down Approach, 6th Edition by James F.Kurose and Keith W.Ross. Number of sections in the article has transmitted to textbook of 4th edition.）

摘要：本题要求描述在浏览器地址栏中输入一个 URL 并按回车后将要发生的事情。大致过程包括：

1. 发送 DHCP request 消息：封装到 UDP 数据报和 IP 数据报，源 IP 0.0.0.0，目的 IP 255.255.255.255。
2. IP 数据报封装为以太网帧，源 MAC 地址 00:16:D3:23:68:8A，目的 MAC 地址 FF:FF:FF:FF:FF:FF。
3. 交换机收到广播消息并向所有端口转发。
4. 作为 DHCP 服务器的路由器收到这一消息，解封装并交付上层。
5. 分配 IP 地址，将 IP 地址、DNS 地址和网关地址放入 DHCP ACK 消息，封装并发送。
6. 交换机经过自学习，直接发送到目的地址。
7. 客户机收到分配的 IP 地址，可以上网。
8. 请求 google 的 IP 地址，封装 DNS query 消息。
9. 想要发送 DNS query，发现不知道网关的 MAC 地址。
10. 广播发送 ARP query，根据网关的 IP 请求网关的 MAC 地址。
11. 网关发送 ARP reply，告知自己的 MAC 地址。
12. 客户机收到 ARP reply。
13. 客户机将 DNS query 发送到网关。
14. 网关根据 DNS query 报文中携带的 DNS 服务器地址转发 DNS query 报文。
15. 中间路由器的转发。
16. 本地 DNS 服务器收到 query，查询到已被缓存的 google 的 IP 地址，经查询权威 DNS 知道 IP 没有改动，发送 reply。
17. 客户机收到 DNS reply，获得了 google 的 IP 地址。
18. 客户机开启 socket，发送 TCP SYN 的握手消息以建立 TCP 连接。
19. 数据报经过复用，选路，转发，分用，到达服务器。
20. 服务器开启 socket，发送 TCP SYNACK 的握手消息以建立 TCP 连接。
21. 服务器返回消息。
22. 连接建立，发送 HTTP GET 请求消息。
23. 服务器收到请求消息，将所请求的内容放入 response 消息中返回。
24. 客户端收到响应消息，解封装后在浏览器中显示。

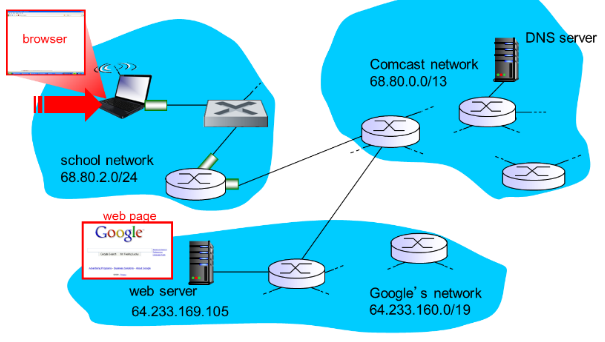
原文中省略了一些步骤。包括 DHCP discover 和 DHCP offer 报文，以及第三次握手的消息等等。请参见原文给出的章节编号自行查看。

下列各图说明了前文描述的过程。

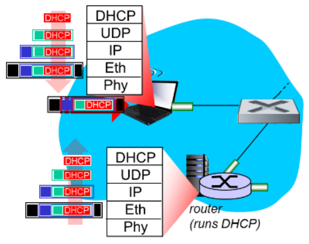
Synthesis: a day in the life of a web request

- journey down protocol stack complete!
  - application, transport, network, link
- putting-it-all-together: synthesis!
  - *goal:* identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
  - *scenario:* student attaches laptop to campus network, requests/receives www.google.com

A day in the life: scenario

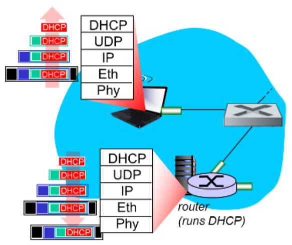


A day in the life… connecting to the Internet



- connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use *DHCP*
- DHCP request *encapsulated* in *UDP*, encapsulated in *IP*, encapsulated in *802.3* Ethernet
- Ethernet frame *broadcast* (dest: FFFFFFFFFFFF) on LAN, received at router running *DHCP* server
- Ethernet *demuxed* to IP demuxed, UDP demuxed to DHCP

A day in the life… connecting to the Internet

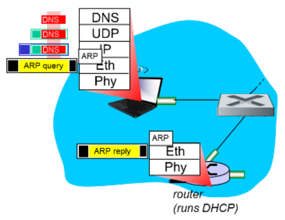


- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (*switch learning*) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

4

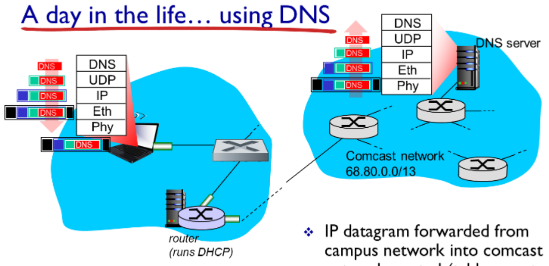
A day in the life… ARP (before DNS, before HTTP)



- before sending *HTTP* request, need IP address of www.google.com: *DNS*
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. To send frame to router, need MAC address of router interface: *ARP*
- *ARP query* broadcast, received by router, which replies with *ARP reply* giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query

5

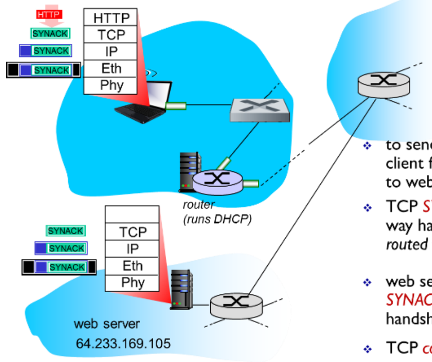
A day in the life… using DNS



- IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router
- IP datagram forwarded from campus network into comcast network, routed (tables created by *RIP, OSPF, IS-IS* and/or *BGP* routing protocols) to DNS server
- demux'ed to DNS server
- DNS server replies to client with IP address of www.google.com

6

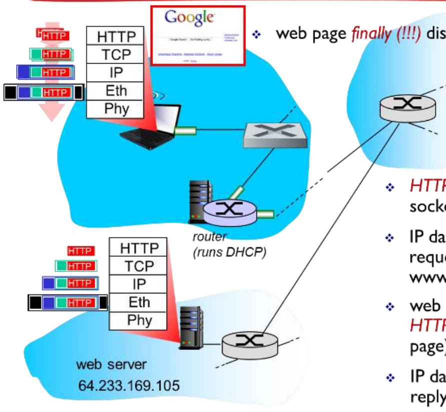
A day in the life…TCP connection carrying HTTP



- to send HTTP request, client first opens *TCP socket* to web server
- TCP *SYN segment* (step 1 in 3-way handshake) *inter-domain routed* to web server
- web server responds with *TCP SYNACK* (step 2 in 3-way handshake)
- TCP *connection established!*

7

A day in the life… HTTP request/reply



- web page *finally (!!!)* displayed
- *HTTP request* sent into TCP socket
- IP datagram containing HTTP request routed to www.google.com
- web server responds with *HTTP reply* (containing web page)
- IP datagram containing HTTP reply routed back to client

说明：内容仅供参考。标注的页码为知识点在中文版教材的位置。