

分组交换：源将报文分为小数据报（分组），每个分组都要通过通信链路和分组交换机。分组交换机分为路由器和链路交换机。**电路交换：**端系统通信期间预留了端系统间通信所需的资源。在发送方能够发送信息之前，该网络必须在发送方和接收方之间建立一条连接。这是一个名副其实的连接，因为此时沿着发送方和接收方之间路径上的交换机都将为该连接维护连接状态。该连接被称为一条电。当网络创建这种电路时，它也在连接期间在该网络链路上预留了恒定的传输速率（表示为每条链路传输容量的一部分），从而确保发送方能以恒定的速率向接收方发送数据。**存储转发交换：**交换机在能够输出该分组的第一个比特之前，必须接收到该分组的全部数据。

帧复用和时分复用：FDM：连接期间链路为每个连接分配一个频段。TDM 时域被分为帧。每个帧有 4 个时隙，每个时隙由一个特定的发送方接收方对专用。对于 TDM，一条电路的传输速率等于帧速率乘以一个时隙中的比特数量。例如，如果链路每秒传输 8000 个帧，每个时隙由 8 个比特组成，则每条电路的传输速率是 64kbps。

分组交换与电路交换的对比：分组交换不适合实时服务（例如，电话和视频会议），因为它的端到端时延是可变的和不可预测的（主要是因为排队时延的变动和不可预测所致）。分组交换的优点：（1）它提供了比电路交换更好的带宽共享；（2）它比电路交换更简单、更有效，实现成本更低。

几种时延：1. 处理时延：检查分组首部和决定将该分组导向何处所需要的时间是处理时延的一部分。处理时延也能包括其他因素，如检查比特级别的差错所需的时间，该差错出现在从上游结点向路由器 A 传输这些分组比特的过程中。2. 排队时延：在队列中，当分组在链路上等待传输时，它经受排队时延。一个特定分组的排队时延将取决于先期到达的正在排队等待向链路传输的分组数量。3. 传输时延：假定分组以先到先服务方式传输，仅当所有已经到达的分组都被传输后，才能传输刚到达的分组。用 L 表示该分组的长度，用 R （bps 即 b/s）表示从路由器 A 到路由器 B 的链路传输速率。例如，对于一条 $L=10Mb$ 的以太网链路，速率 $R=10Mbps$ ；对于 $100Mbps$ 的以太网链路，速率 $R=100Mbps$ 。传输时延是 L/R 。这是将所有分组的比特（传输）向链路所需要的时间。4. 传播时延：一旦一个比特被推向链路，该比特需要向路由器 B 传播。从该链路的起点到路由器 B 传播所需要的时间是传播时延。该比特以该链路的传播速率传播。该传播速率取决于该链路的物理媒体（即光纤、双绞线等），其速率范围是 $2 \times 10^8 \sim 3 \times 10^8$ m/s，这等于或略小于光速。该传播时延等于两台路由器之间的距离除以传播速率。

瞬时吞吐量：主机接收到文件的速率（bps）| 平均吞吐量：主机接收到所有 F 比特用去 T 秒，F/T

消息首先通过 HTTP 从 Alice 的主机发送到她的邮件服务器。

然后，Alice 的邮件服务器通过 SMTP 应用层

向 Bob 的邮件服务器发送消息。

然后 Bob 通过 POP3 将消息从他的邮件服务器表示层

传输到他的主机。

应用层

会话层

运输层

网络层

链路层

物理层

5 种非专用的因特网应用及它们所使用的因特网协议。

Web 应用和 HTTP 协议、文件传输 FTP

电子邮件应用和 SMTP（简单邮件传输协议）、

因特网的目录服务 DNS 和 DNS 协议、

P2P 应用和 P2P 协议、

远程终端访问 Telnet

运输层提供的服务

1. 可靠数据传输，TCP 提供了可靠的端到端数据传输服务，而 UDP 没有。

2. 吞吐量，TCP 和 UDP 均为提供此服务。

3. 定时，TCP 和 UDP 均为提供此服务。

4. 安全性，TCP 在应用层可以很容易地通过 SSL 提供安全服务，而 UDP 没有。

a) 五层因特网协议栈

b) 七层 ISO OSI 参考模型

病毒：是一种需要某种形式的用户交互来感染用户设备的恶意软件。典型的例子是包含恶意可执行代码的电子邮件附件。蠕虫，是一种无需任何明显用户交互就能进入设备的恶意软件。例如，用户运行一个某攻击者能够发送恶意软件的蠕虫网络应用程序。在某些情况下，没有用户的任何干预，该应用程序能从因特网接收恶意软件并运行它。新近感染设备的蠕虫则能扫描因特网，搜索其他运行相同易受感染的网络应用程序的主机。当它发现其他易受感染的主机时，便向这些主机发送一个它自身的副本。

第二章：应用层

客户-服务器体系结构：有一个总是打开的主机称为服务器，它服务于其他客户主机的请求，客户彼此间不互相影响，如 Web 服务。

P2P 体系结构：对专用服务器有很小甚至没有依赖，直接连接的主机之间直接通信。自拓展性：对等方之间相互传输文件，增加系统服务能力。成本有效：不需要庞大的服务器和大带宽，如 Skype 因特网电话、迅雷。

套接字：又叫应用程序编程接口（Application Programming Interface, API），一个进程通过套接字向网络发送报文和从网络接收报文。套接字是进程进化的“门”，是进程与计算机网络之间的接口。

进程寻址：标识接受进程的两个信息：1. 主机的地址：IP 地址 2. 定义在主机中进程的标识符：端口号（portnumber）Web: 80 邮件服务器 SMTP: 25

可供应用程序使用的运输服务：1. 应用程序将报文推入套接字 2. 运输层协议将报文推入接收进程的套接字。

运输层协议提供的服务：1. 可靠数据传输：一个协议提供了确保数据交付服务：可靠数据传输，只要进入套接字的数据都会无差别地传输。不提供可靠数据传输：可能造成容忍丢失的应用（多媒体类）接受。2. 吞吐量：发送进程向接收进程交付比特的速率以特定速率提供确保可用的吞吐量吞吐量不足，应用将降低编码的速率。带宽敏感应用：对吞吐量有要求，如多媒体应用，因特网电话。弹性应用：根据情况灵活利用可用的吞吐量，如电子邮件、文件传输、Web 传输。3. 定时：保证小时的时延，吸引交互式实时应用因特网电话虚拟环境多方游戏电话会议。

4. 安全性：加密发送进程传输的所有数据交付给接收进程之前解密所有数据，包括数据完整性，端点鉴别。

因特网运输协议提供的服务 TCP：1. 面向连接的服务，握手：客户和服务器交换运输层信息 | 全双工连接：双方可以同时在连接上收发。应用层结束报文发送，必须拆除连接 2. 可靠的数据传送服务，无差错 | 无丢失 | 无冗余 | 按照一定顺序交付数据 3. 拥塞控制机制，出现拥塞，抑制发送进程 | 限制每一个 TCP 连接，达到公平共享带宽的目的。UDP 1. 轻量级运输协议，只提供最小服务 2. 没有握手过程，提供一种不可靠数据传送服务 | 不保证数据能到达 | 不保证数据到达的顺序 3. 无拥塞控制机制，可以以任何速率（实际上应该无法达到，因为有拥塞和带宽的限制）向下层（网络层）注入数据。

应用	应用层协议	支撑的运输协议	POP3
电子邮件	SMTP [RFC 5321]	TCP TCP 的拥塞控制会在拥塞时限制应用层的发送速率。通常，IP	
远程终端访问	Telnet [RFC 854]	TCP 电话和视频会议选择在 UDP 上运行	
Web	HTTP [RFC 2616]	他们的应用层另外，有些应用程序	
文件传输	FTP [RFC 959]	不需要	
流式多媒体	HTTP (如 YouTube)	TCP 提供的可靠数据传输。	
因特网电话	SIP [RFC 3261]、RTP [RFC 3505] 或专用的（如 Skype）	UDP 或 TCP	

因特网电话 1. 带宽敏感 2. 能够容忍丢失，但必须保证传输速率，否则无法正确解码 3. 采用 UDP 可以避免 TCP 的拥塞控制和分组带来的时间开销 4. 由于防火墙会阻挡 UDP，通常设置为如果 UDP 失败就采用 TCP。由于大多数防火墙都被配置为阻止 UDP 通信，因此使用 TCP 进行视频和语音通信可以让通信通过防火墙。

HTTP：是 Web 的应用层协议。多数 Web 页面含有一个 HTML 基本文件和几个引用对象。使用 TCP 作为支撑运输协议。不保存客户信息，是无状态协议。

图 2-5 流行的因特网应用及其应用层协议和支撑的运输协议

持续 | 非持续连接：1. 非持续，一个对象一次 TCP 连接，发完之后就关闭连接，要发送 11 个对象，要 11 个 TCP 连接。2. 持续，几个对象甚至几个 Web 页面都用单个持续 TCP 连接。如果一段时间未被使用，就关闭连接。HTTP 默认是带流水线的持续连接。

RTT 往返时间：一个短分组从客户到服务器再返回客户所需的时间。RTT 包括四种时延。TCP 三次握手，总的响应时间是传输文件时间 + 2RTT。

HTTP 请求报文：1. 请求行，方法字段，url 字段，HTTP 版本字段。2. 首部行 Host：提供主机信息，虽有 TCP 协议，但这是 Web 代理高速缓存要求的信息。Connection：closed 要求发送完被请求的对象后就关闭连接，而不是持续打开。User-Agent 指明用户代理（浏览器类型），方便服务器发送相同对象的不同版本。Accept-Language：如果有符合该语言要求的版本就发送，否则发送默认版本。

HTTP 响应报文：a. 1 个初始状态行：协议版本信息 | 状态码：200OK：请求成功，301 Moved Permanently：文件被永久移除，客户端会自动获取位于 Location 中的新 URL，400 Bad Request：该请求不能被服务器理解，404 Not Found：该文件不在服务器上，505 HTTP Version Not Supported：服务器不支持使用的 HTTP 版本相适应信息 b. 6 个首部行：Connection：close，发送完后关闭 TCP，Date：服务器产生并发送该报文的日期时间，Server：指示报文由哪一种服务器产生，Last-Modified：对象创建或者最后修改的日期时间，Content-Length：发送内容的字节数，Content-Type：指示对象文件类型 c. 实体体：Post 请求中客户提交的信息。

用户与服务器交互 | Cookie：Cookie 由一个 Cookie 文件，由用户浏览器管理。4. Web 的一个后端数据库。演示：访问 Amazon，发送请求报文，Web 产生唯一识别码，作为数据库索引，响应报文

中 Set-Cookie+ 识别码，用户浏览器根据识别码在 Cookie 文件中创建一个条目，以后浏览一个网站，浏览器就会在请求报文加入这个网站的识别码。

WEB 缓存器：同时是服务器又是客户。1. 可以大大减少对客户请求的响应时间。2. Web 缓存器能够大大减少一个机构的接入链路到因特网的通信量。通过减少通信量，该机构就不必增加带宽，因此降低了费用。3. Web 缓存器能从整体上大大减低因特网上的 Web 流量，从而改善了所有应用的性能。

因特网电子邮件系统由用户代理、邮件服务器，简单邮件传输协议 SMTP 组成。邮件发送过程：1. 发送方的用户代理将邮件发送到发送方的邮件服务器 2. 发送方的服务器将邮件传输到接收方的邮件服务器 3. 接收方要在邮箱里接收到报文时，接收方的服务器通过用户名和口令来识别它 4. 发送方的服务器要能处理不能将邮件发送给接收方服务器的情况，要有一个报文序列号。

SMTP 因特网电子邮件中的主要的应用层协议，使用 TCP 可靠传输，有服务端和客户端，一个邮箱服务器上既有服务端也有客户端。传输过程 1. 客户 SMTP 在 25 端口建立一个到服务器端口的 TCP 连接，如果服务器没有开机，则客户稍后会重新尝试 2. SMTP 客户向服务器指示发送方的邮件地址和接收方的邮件地址

3. SMTP 通过 TCP 将邮件无差错传输到接收服务器 4. 若该客户有另外的报文要发送则在相同的 TCP 上发送，否则关闭这条连接。

与 HTTP 对比相同点：都从一台主机向另一台主机传输文件，HTTP 从一个 Web 服务器到 Web 客户，SMTP 从一个邮箱服务器到另一个邮箱服务器；持续的 HTTP 和 SMTP 都使用持续连接。区别：HTTP 是一个拉协议，TCP 连接是由想要接收文件的人发起的。SMTP 基本是一个推协议，TCP 连接是由想发送文件的人发起的；SMTP 要求报文格式为 7bit ASCII 码；HTTP 把每个对象封装到自己的 HTTP 响应报文中，而 SMTP 则把所有报文对象放在一个报文中。MTA 代表邮件传输代理（Mail Transfer Agent）。主机将消息发送给 MTA 邮件首部“Received”表示此行描述了邮件在传输过程中的一个步骤。“fromasus-4b96”表示邮件从名为“asus-4b96”的主机发出，“[localhost 127.0.0.1]”表示该主机将自身视为本地主机 localhost，IP 地址为 127.0.0.1。这是一个特殊的回环地址，通常用于表示本地计算机。“bybarmail.cs.umass.edu(SpamFirewall)”表示邮件经由位于 barmail.cs.umass.edu 的服务器接收，并通过该服务器上的垃圾邮件防火墙进行了处理。“forhg@cs.umass.edu”表示邮件的最终接收者邮箱地址是 hg@cs.umass.edu，“Fri.”是邮件接收日期的缩写，表示星期五。

DNS 识别主机的两种方式：主机名，IP 地址。DNS 是一个由分层的 DNS 服务器实现的分布式数据库，一个使得主机可以查询分布式数据库的应用层协议，它将主机名解析为 IP 地址。DNS 的其他三个服务：主机别名，邮件服务器别名，负裁分配。

DNS 运行过程：用主机如何获得 school.edu.cn 的 IP 地址？同一台用户主机上运行着 DNS 应用的客户端。浏览器从上述 URL 中抽取出主机名 www.someschool.edu，并将这台主机名传给 DNS 应用的客户端。DNS 客户通过 UDP 向 DNS 服务器发送一个包含主机名的请求。DNS 客户最终会收到一份回答报文，其中含有对应于该主机名的 IP 地址。一旦浏览器接收到来自 DNS 的该 IP 地址，它能够向位于该 IP 地址 80 端口的 HTTP 服务器进程发起一个 TCP 连接。

单 DNS 的缺点：1. 单点故障：一旦坏了，整个互联网崩溃 2. 通信容量：一个 DNS 服务器处理所有 DNS 查询 3. 远距离的集中式数据库：对离服务器远的地方延时会很大 4. 维护不易

根 DNS 服务器

com DNS 服务器 org DNS 服务器 edu DNS 服务器

yahoo.com DNS 服务器 amazon.com DNS 服务器 pbs.org DNS 服务器 poly.edu DNS 服务器 umass.edu DNS 服务器

请求机 client 1 client 2 client 3 client 4 client 5

本地 DNS 服务器 dns1.cs.umass.edu dns2.cs.umass.edu dns3.cs.umass.edu

请求机 client 6 client 7 client 8 client 9 client 10

TLD DNS 服务器 dns4.cs.umass.edu dns5.cs.umass.edu dns6.cs.umass.edu

请求机 client 11 client 12 client 13 client 14 client 15

根 DNS 服务器

client 16 client 17 client 18 client 19 client 20

请求机 client 21 client 22 client 23 client 24 client 25

请求机 client 26 client 27 client 28 client 29 client 30

请求机 client 31 client 32 client 33 client 34 client 35

请求机 client 36 client 37 client 38 client 39 client 40

请求机 client 41 client 42 client 43 client 44 client 45

请求机 client 46 client 47 client 48 client 49 client 50

请求机 client 51 client 52 client 53 client 54 client 55

请求机 client 56 client 57 client 58 client 59 client 60

请求机 client 61 client 62 client 63 client 64 client 65

请求机 client 66 client 67 client 68 client 69 client 70

请求机 client 71 client 72 client 73 client 74 client 75

请求机 client 76 client 77 client 78 client 79 client 80

请求机 client 81 client 82 client 83 client 84 client 85

请求机 client 86 client 87 client 88 client 89 client 90

请求机 client 91 client 92 client 93 client 94 client 95

请求机 client 96 client 97 client 98 client 99 client 100

请求机 client 101 client 102 client 103 client 104 client 105

请求机 client 106 client 107 client 108 client 109 client 110

请求机 client 111 client 112 client 113 client 114 client 115

请求机 client 116 client 117 client 118 client 119 client 120

请求机 client 121 client 122 client 123 client 124 client 125

请求机 client 126 client 127 client 128 client 129 client 130

请求机 client 131 client 132 client 133 client 134 client 135

请求机 client 136 client 137 client 138 client 139 client 140

请求机 client 141 client 142 client 143 client 144 client 145

请求机 client 146 client 147 client 148 client 149 client 150

请求机 client 151 client 152 client 153 client 154 client 155

请求机 client 156 client 157 client 158 client 159 client 160

请求机 client 161 client 162 client 163 client 164 client 165

请求机 client 166 client 167 client 168 client 169 client 170

请求机 client 171 client 172 client 173 client 174 client 175

请求机 client 176 client 177 client 178 client 179 client 180

请求机 client 181 client 182 client 183 client 184 client 185

请求机 client 186 client 187 client 188 client 189 client 190

请求机 client 191 client 192 client 193 client 194 client 195

请求机 client 196 client 197 client 198 client 199 client 200

请求机 client 201 client 202 client 203 client 204 client 205

请求机 client 206 client 207 client 208 client 209 client 210

请求机 client 211 client 212 client 213 client 214 client 215

请求机 client 216 client 217 client 218 client 219 client 220

请求机 client 221 client 222 client 223 client 224 client 225

请求机 client 226 client 227 client 228 client 229 client 230

请求机 client 231 client 232 client 233 client 234 client 235

请求机 client 236 client 237 client 238 client 239 client 240

请求机 client 241 client 242 client 243 client 244 client 245

请求机 client 246 client 247 client 248 client 249 client 250

请求机 client 251 client 252 client 253 client 254 client 255

请求机 client 256 client 257 client 258 client 259 client 260

请求机 client 261 client 262 client 263 client 264 client 265

请求机 client 266 client 267 client 268 client 269 client 270

请求机 client 271 client 272 client 273 client 274 client 275

请求机 client 276 client 277 client 278 client 279 client 280

请求机 client 281 client 282 client 283 client 284 client 285

请求机 client 286 client 287 client 288 client 289 client 290

请求机 client 291 client 292 client 293 client 294 client 295

请求机 client 296 client 297 client 298 client 299 client 300

请求机 client 301 client 302 client 303 client 304 client 305

请求机 client 306 client 307 client 308 client 309 client 310

请求机 client 311 client 312 client 313 client 314 client 315

请求机 client 316 client 317 client 318 client 319 client 320

请求机 client 321 client 322 client 323 client 324 client 325

请求机 client 326 client 327 client 328 client 329 client 330

请求机 client 331 client 332 client 333 client 334 client 335

请求机 client 336 client 337 client 338 client 339 client 340

请求机 client 341 client 342 client 343 client 344 client 345

请求机 client 346 client 347 client 348 client 349 client 350

请求机 client 351 client 352 client 353 client 354 client 355

请求机 client 356 client 357 client 358 client 359 client 360

请求机 client 361 client 362 client 363 client 364 client 365

请求机 client 366 client 367 client 368 client 369 client 370

请求机 client 371 client 372 client 373 client 374 client 375

请求机 client 376 client 377 client 378 client 379 client 380

请求机 client 381 client 382 client 383 client 384 client 385

请求机 client 386 client 387 client 388 client 389 client 390

请求机 client 391 client 392 client 393 client 394 client 395

捎带确认对客户到服务器的数据的确认被装载在一个承载服务器到客户的数据的报文段中；这种确认被称为是被捎带在服务器到客户的数据报文段中的。

$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}, \alpha = 0.125$$

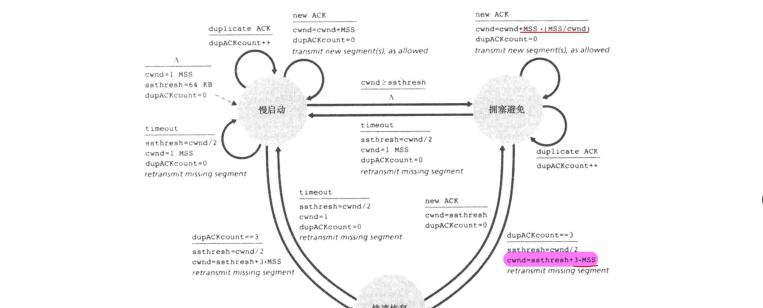
$$\text{DevRTT} = (1 - \beta) \cdot (\text{DevRTT} + \beta \cdot |\text{SampleRTT} - \text{EstimatedRTT}|), \beta = 0.25$$

估算的 RTT 和样本值的偏差。TimeoutInterval=EstimatedRTT+4*DevRTT
快速重传：如果 TCP 发送方收到对相同数据的 3 个冗余 ACK，说明跟在这个数据后的分组丢失，那么我们就在该报文的定时器过期时重传丢失的报文。

TCP 一方面采取累积确认，另一方面也用选择确认，是 SR 和 GBN 的混合体

TCP 拥塞控制：TCP 连接的每一端都是由一个接收缓存、一个发送缓存和几个变量（LastByteRead、rwnd 等）组成，运行在发送方的 TCP 拥塞控制机制跟踪一个额外的变量拥塞窗口 cwnd，它对一个 TCP 发送方能向网络中发送流量的速率进行了限制。特点是，在一个发送方中未被确认的数据量不会超过 cwnd 与 rwnd 中的最小值，即 $\text{LastByteSent} - \text{LastByteAcked} \leq \min\{\text{cwnd}, \text{rwnd}\}$ 。在每个往返时间 (RTT) 的起始点，上面的限制条件允许发送方将连接发送 cwnd 个字节的数据，在该 RTT 结束时发送方接收对数据的确认报文。因此，该发送方的发送速率大概是 cwnd/RTT 字节/秒。

TCP 拥塞控制算法



b)如果W足够大, $\frac{W^2}{3} \gg \frac{1}{W}$ 。因此, $L \approx 8/3W$ 或者 $W \approx \sqrt{\frac{L}{8}}$

因此可以有平均传输速率:

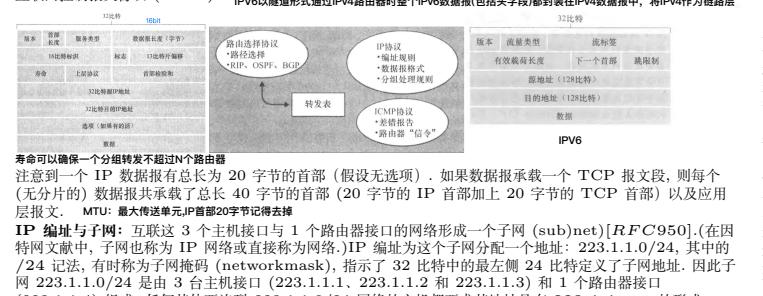
$$\frac{1}{4}W \cdot \text{MSS} = \frac{1}{4}RTT \cdot MSS$$

$$(简化) \text{一条连接的平均吞吐量} = \frac{0.75}{RTT}, \text{一条连接的平均吞吐量} = \frac{1.22 \times MSS}{RTT \sqrt{L}}$$

第四、五章：网络层 转发：当一个分组到达一个路由器的输入链路时，路由器要把它移动到合适的输出链路。路由选择：当分组从发送方流向接收方时，网络层必须决定这些分组所采用的路由或路径。计算这些路径的算法被称为路由选择算法。仅在网络层提供连接服务的计算机网络称为虚电路（Virtual-Circuit, VC）网络；仅在网络层提供无连接服务的计算机网络称为数据报网络。因特网是一个数据报网络（datagram network）。

网络层的虚电路建立与运输层的连接建立之间的区别：运输层的连接建立仅涉及两个端系统。在运输层的连接建立期间，两个端系统独自决定运输层连接的参数。虽然这两个端系统已经知道该运输层连接，但网络中的路由器则对这些完全不知情。在另一方面，对于一个虚电路网络层，沿两个端系统之间路径上的路由器都要参与虚电路的建立，且每台路由器都完全知道经过它的所有虚电路。

网络层的三个组件：1. IP 协议 2. 路由选择 3. 报告数据报中的差错和对某些网络层信息请求进行响应的设施即互联网控制报文协议（ICMP）。IPV6 以隧道形式通过 IPV4 路由器时整个 IPV6 数据报（包括头字段）都封装在 IPV4 数据报中，将 IPV4 作为链路层



寿命可以确保一个分组转发不超过N个路由器

注意到一个 IP 数据报有总长为 20 字节的首部（假设无选项）。如果数据报承载一个 TCP 报文段，则每个（无分片）的数据报共承载了总长 40 字节的首部（20 字节的 IP 首部加上 20 字节的 TCP 首部）以及应用层报文。

编址与子网：互联 3 个主机接口与 1 个路由器接口的网络形成一个子网（subnet）[RFC 950]。（在因特网文献中，子网也称为 IP 网络或直接称为网段。）IP 编址为这个子网分配一个地址：223.1.1.0/24，其中的 /24 记法，有时称为子网掩码（networkmask），指示了 32 比特中的最左侧 24 比特定义了子网地址。因此网段 223.1.1.0/24 是由 3 台主干接口（223.1.1.1、223.1.1.2 和 223.1.1.3）和 1 个路由器接口（223.1.1.4）组成，任何其他要连到 223.1.1.0/24 网络的主机都要求其地址具有 223.1.1.xxx 的形式。



循环冗余检测 d 比特数据 D，发送方和接收方协商一个 $r+1$ 比特生成多项式 G，G 最左边是 1，然后发送方计算 r 个附加比特 R, $R = D \cdot 2^r/G$ 的余数。这里的加减计算无进位和退位。

CSMA/CD

动态主机配置协议：某组织一旦获得了一块地址，它就可为本组织内的主机与路由器接口逐个分配 IP 地址。系统管理员通常手工配置路由器中的 IP 地址。主机地址也能手动配置，但是这项任务目前通常更多的是使用动态主机配置协议（Dynamic Host Configuration, DHCP）来完成。DHCP 允许主机自动获取（被分配）一个 IP 地址。网络管理员能够配置 DHCP，以便某组织的每一个 IP 地址，或者某主机将被分配一个临时的 IP 地址，该地址在每次与网络连接时也许是不同的。除了主机 IP 地址分配外，DHCP 还允许一台主机得知其他信息，例如它的子网掩码、它的第一跳路由器地址（常称为默认网关）与它的本地 DNS 服务器地址。DHCP 也被称为即插即用协议。DHCP 是一个客户-服务器协议，客户是新到达的主机，服务器是 DHCP 服务器。

UPnP 允许外部主机使用 TCP 或 UDP 向 NAT 化的主机发起通信会话。

插入端口出现分组丢失
如果数据包到达交换矩阵的速率超过交换矩阵速率，则数据包需要在输入端口排队。如果此速率不匹配仍然存在，则队列会变得更大和更大，并最终溢出输入端口缓冲区，从而导致数据包丢失。如果丢弃结构速度至少为 n，则可以消除分组丢失并提高入线速。
其中 n 为输入端口的数据量。

输出端口出现分组丢失
假设输入和输出线速率相同，如果数据包到达单个输出端口的速率超出了线速率，则仍可能发生丢包。如果速率不匹配仍然存在，队列将变得越来越大，并最终溢出输出端口缓冲区，从而导致数据包丢失。
请提高开关结构的速度并防止此问题的发生。

CIDR：因特网的地址分配策略被称为无类别域间路由选择（Classless Interdomain Routing, CIDR）。CIDR 将子网地址的概念一般化了。因为对于子网地址，32 比特的 IP 地址被划分为两部分，并且也具有点分十进制数形式 a.b.c.d/x，其中 x 指示了地址的第一部分中的比特数。CIDR 采用之前，子网地址则只能是 8,16,24bit。

NAT 网络地址转换
NAT 是中间盒的一种。
中间盒并不执行传统的数据报转发，而是执行诸如 NAT、流量流的负载均衡、流量防火墙等功能。

SDN
数据平面的主要作用是从其输入链路向其输出链路转发数据报；控制平面的主要作用是协调这些本地的每个路由器的转发动作，使得数据报沿着源和目的主机之间的路径在路由器继续进行端到端的传送。

MTU：最大发送报文
图 4-29 OpenFlow 1.0 流表的分组匹配字段

端口号是与到达接口相同，丢弃 3. 与到达接口不同，转发。自学习：1. 交换机表初始化为全 0。2. 在每个接口接收到的每个人帧，该交换机在其表中存储；(1) 在该帧源地址字段中的 MAC 地址；(2) 该帧到达的接口；(3) 当前时间，交换机以这种方式在它的表中记录了发送节点所在的局域网网段。如果在局域网上的每个主机最终都发送了一个帧，则每个主机最终将在这张表中留有记录。3. 如果在一段时间（称为老化期）后，交换机没有接收到以该地址作为源地址的帧，就在表中删除这个地址。以这种方式，如果一台 PC 被另一台 PC（具有不同的适配器）代替，原来 PC 的 MAC 地址将最终从该交换机表中被清除掉。

数据中心 1. 负载均衡。为了支持来自外部客户的请求，每一个应用都与一个公开可见的 IP 地址关联，外部用户向该地址发送其请求并从该地址接收响应。在数据中心内部，外部请求首先被定向到一个负载均衡器。负载均衡器的任务是向主机分发请求，以主机当前的负载作为函数来在主机之间均衡负载。负载均衡器将该请求分发到处理该应用的某一主机上。当主机处理完该请求后，向负载均衡器回送响应，再由负载均衡器将其继发给外部客户。负载均衡器不仅平衡主机间的工作负载，而且还提供类似 NAT 的功能，将外部 IP 地址转换为内部适当主机的 IP 地址。然后将反向方向向客户分组按照相反的转换进行处理，这防止客户直接接触主机，从而避免了直接访问。在实现中，为了性能或成本原因，一个流表可以由多个流表实现 [Boswell 2013]，但我们在这里只关注一个流表的实现。数据报可以由许多流表实现 [Boswell 2013]，但我们在实现中只关注一个流表的实现。当分组匹配表项与数据报匹配时的动作命令，这些动作可能将分组转发到新的输出端口，丢弃该分组、复制分组和它们发送到多个输出端口，和/或重写所选的首部字节。

端口号是与到达接口相同，丢弃 3. 与到达接口不同，转发。自学习：1. 交换机表初始化为全 0。2. 在每个接口接收到的每个人帧，该交换机在其表中存储；(1) 在该帧源地址字段中的 MAC 地址；(2) 该帧到达的接口；(3) 当前时间，交换机以这种方式在它的表中记录了发送节点所在的局域网网段。如果在局域网上的每个主机最终都发送了一个帧，则每个主机最终将在这张表中留有记录。3. 如果在一段时间（称为老化期）后，交换机没有接收到以该地址作为源地址的帧，就在表中删除这个地址。以这种方式，如果一台 PC 被另一台 PC（具有不同的适配器）代替，原来 PC 的 MAC 地址将最终从该交换机表中被清除掉。

松耦合：松耦合是指在数据中心中各个组件或子系统之间相互独立、相对解耦的设计方式。松耦合的设计具有以下特点：模块化、异步通信、弹性和平滑性。

紧耦合：紧耦合是指数据中心中各个组件或子系统之间高度依赖、相互密切连接的设计方式。紧耦合的设计具有以下特点：直接依赖、同步通信、高性能。紧耦合的设计可以提供更高的性能和效率。特别适用于需要实时数据处理或密集计算的场景。但是，紧耦合也可能导致系统的可扩展性和可维护性较差，因为各个组件之间的依赖性较高。通路可达通路可达是指数据中心中各个模块之间相互连接的可靠性。在数据中心中，不同的模块之间需要通过网络或其他通信方式进行数据交换和协作。

虚拟局域网 VLAN 在一个基于端口的 VLAN 中，交换机的端口（接口）由网络管理员划分为组。每个组成为一个 VLAN，在每个 VLAN 中的端口形成一个广播域（即来自一个端口的广播流量仅能到达该组中的其他端口）。特殊端口连接外部路由器来解决一台交换机上多个 VLAN 互连的问题：加入 VLAN 干线来解决不同交换机上相同 VLAN 的连接问题。VLAN 的选择由 4 字节的 VLAN 标签决定。

第八章：网络安全 在对称密钥系统中，Alice 和 Bob 的密钥是相同的并且是秘密的。在公开密钥系统中，使用一对密钥：一个密钥为 Bob 和 Alice 俩人所知（实际上为全世界所知），另一个密钥只有 Bob 或 Alice 知道（而不是双方都知道）。

RSA 生成 RSA 的公钥和私钥，Bob 执行如下步骤：1) 选择两个大素数 p 和 q ，那么 p 和 q 应该多大呢？该值越大，破解 RSA 越困难，而执行加密和解密所用的时间也越长。2) 计算 $n = pq$ 和 $z = (p-1)(q-1)$ 。3) 选择小于 n 的一个数 e ，且使 e 和 n 没有（非 1）的公因数。（这时称 e 与 z 互素。）使用字母 d 表示是因为这个值将被用于加密。4) 求一个数 d ，使得 $ed \equiv 1 \pmod{z}$ （就是说，没有余数）。使用字母 d 表示是因为这个值将用于解密。换句话说，给定 e ，我们选择 d ，使得 $ed = 1 \pmod{z}$ 。Bob 使外界可用的公钥 K_B^+ 是一对 (n, e) ，其私钥 K_B^- 是一对 (n, d) 。Alice 执行的加密和 Bob 执行的解密过程如下：假设 Alice 要给 Bob 发送一个由整数 m 表示的比特组合，且 $m < n$ 。为了对明文报文 m 加密，Bob 计算： $m = c^e$ 对于这个密文 c 的比特模式发送给 Bob。-为了对收到的密文报文 c 解密，Bob 计算： $m = c^d$ 。这要求使用他的私钥 (n, d) 。

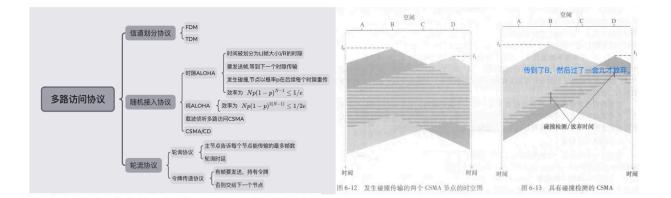
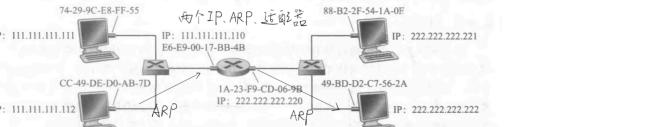


图 6-12 发生碰撞的两个 CSMA 节点的时间空隙

图 6-13 具有碰撞检测的 CSMA

二进制指数后退当传输一个给定帧时，在该帧经历了一连串的 n 次碰撞后，节点随机地从 $\{0, 1, 2, \dots, 2^n - 1\}$ 中选择一个 K 值。因此，一个帧经历的碰撞越多， K 选择的间隔越大。对于以太网，一个节点等待的实际时间量是 $K \cdot 512$ 比特时间（即发送 512 比特进入以太网所需时间的 K 倍）， n 能够取的最大值在 10 以内。

ARP
以太网帧结构



CSMA/CD 效率 令 d_{prop} 表示信号能在任意两个适配器之间传播所需的最大时间。令 d_{trans} 表示传输一个最大长度的以太网帧的时间（对于 10Mbps 的以太网，该时间近似为 1.2 微秒）。CSMA/CD 效率的近似式：效率 = $\frac{1}{1+5d_{prop}/d_{trans}}$

链路层交换机 1. 表中没有表项，广播 2. 表项与到达接口相同，丢弃 3. 与到达接口不同，转发。自学习：1. 交换机表初始化为全 0。2. 在每个接口接收到的每个人帧，该交换机在其表中存储；(1) 在该帧源地址字段中的 MAC 地址；(2) 该帧到达的接口；(3) 当前时间，交换机以这种方式在它的表中记录了发送节点所在的局域网网段。如果在局域网上的每个主机最终都发送了一个帧，则每个主机最终将在这张表中留有记录。3. 如果在一段时间（称为老化期）后，交换机没有接收到以该地址作为源地址的帧，就在表中删除这个地址。以这种方式，如果一台 PC 被另一台 PC（具有不同的适配器）代替，原来 PC 的 MAC 地址将最终从该交换机表中被清除掉。

数据中心 1. 负载均衡。为了支持来自外部客户的请求，每一个应用都与一个公开可见的 IP 地址关联，外部用户向该地址发送其请求并从该地址接收响应。在数据中心内部，外部请求首先被定向到一个负载均衡器。负载均衡器的任务是向主机分发请求，以主机当前的负载作为函数来在主机之间均衡负载。负载均衡器将该请求分发到处理该应用的某一主机上。当主机处理完该请求后，向负载均衡器回送响应，再由负载均衡器将其继发给外部客户。负载均衡器不仅平衡主机间的工作负载，而且还提供类似 NAT 的功能，将外部 IP 地址转换为内部适当主机的 IP 地址。然后将反向方向向客户分组按照相反的转换进行处理，这防止客户直接接触主机，从而避免了直接访问。在实现中，为了性能或成本原因，一个流表可以由多个流表实现 [Boswell 2013]，但我们在实现中只关注一个流表的实现。当分组匹配表项与数据报匹配时的动作命令，这些动作可能将分组转发到新的输出端口，丢弃该分组、复制分组和它们发送到多个输出端口，和/或重写所选的首部字节。

松耦合：松耦合是指在数据中心中各个组件或子系统之间相互独立、相对解耦的设计方式。松耦合的设计具有以下特点：模块化、异步通信、弹性和平滑性。

紧耦合：紧耦合是指数据中心中各个组件或子系统之间高度依赖、相互密切连接的设计方式。紧耦合的设计可以提供更高的性能和效率。特别适用于需要实时数据处理或密集计算的场景。但是，紧耦合也可能导致系统的可扩展性和可维护性较差，因为各个组件之间的依赖性较高。通路可达通路可达是指数据中心中各个模块之间相互连接的可靠性。在数据中心中，不同的模块之间需要通过网络或其他通信方式进行数据交换和协作。

虚拟局域网 VLAN 在一个基于端口的 VLAN 中，交换机的端口（接口）由网络管理员划分为组。每个组成为一个 VLAN，在每个 VLAN 中的端口形成一个广播域（即来自一个端口的广播流量仅能到达该组中的其他端口）。特殊端口连接外部路由器来解决一台交换机上多个 VLAN 互连的问题：加入 VLAN 干线来解决不同交换机上相同 VLAN 的连接问题。VLAN 的选择由 4 字节的 VLAN 标签决定。

第八章：网络安全 在对称密钥系统中，Alice 和 Bob 的密钥是相同的并且是秘密的。在公开密钥系统中，使用一对密钥：一个密钥为 Bob 和 Alice 俩人所知（实际上为全世界所知），另一个密钥只有 Bob 或 Alice 知道（而不是双方都知道）。

RSA 生成 RSA 的公钥和私钥，Bob 执行如下步骤：1) 选择两个大素数 p 和 q ，那么 p 和 q 应该多大呢？该值越大，破解 RSA 越困难，而执行加密和解密所用的时间也越长。2) 计算 $n = pq$ 和 $z = (p-1)(q-1)$ 。3) 选择小于 n 的一个数 e ，且使 e 和 n 没有（非 1）的公因数。（这时称 e 与 z 互素。）使用字母 d 表示是因为这个值将被用于加密。4) 求一个数 d ，使得 $ed \equiv 1 \pmod{z}$ （就是说，没有余数）。使用字母 d 表示是因为这个值将用于解密。换句话说，给定 e ，我们选择 d ，使得 $ed = 1 \pmod{z}$ 。Bob 使外界可用的公钥 K_B^+ 是一对 (n, e) ，其私钥 K_B^- 是一对 (n, d) 。Alice 执行的加密和 Bob 执行的解密过程如下：假设 Alice 要给 Bob 发送一个由整数 m 表示的比特组合，且 $m < n$ 。为了对明文报文 m 加密，Bob 计算： $m = c^e$ 对于这个密文 c 的比特模式发送给 Bob。-为了对收到的密文报文 c 解密，Bob 计算： $m = c^d$ 。这要求使用他的私钥 (n, d) 。