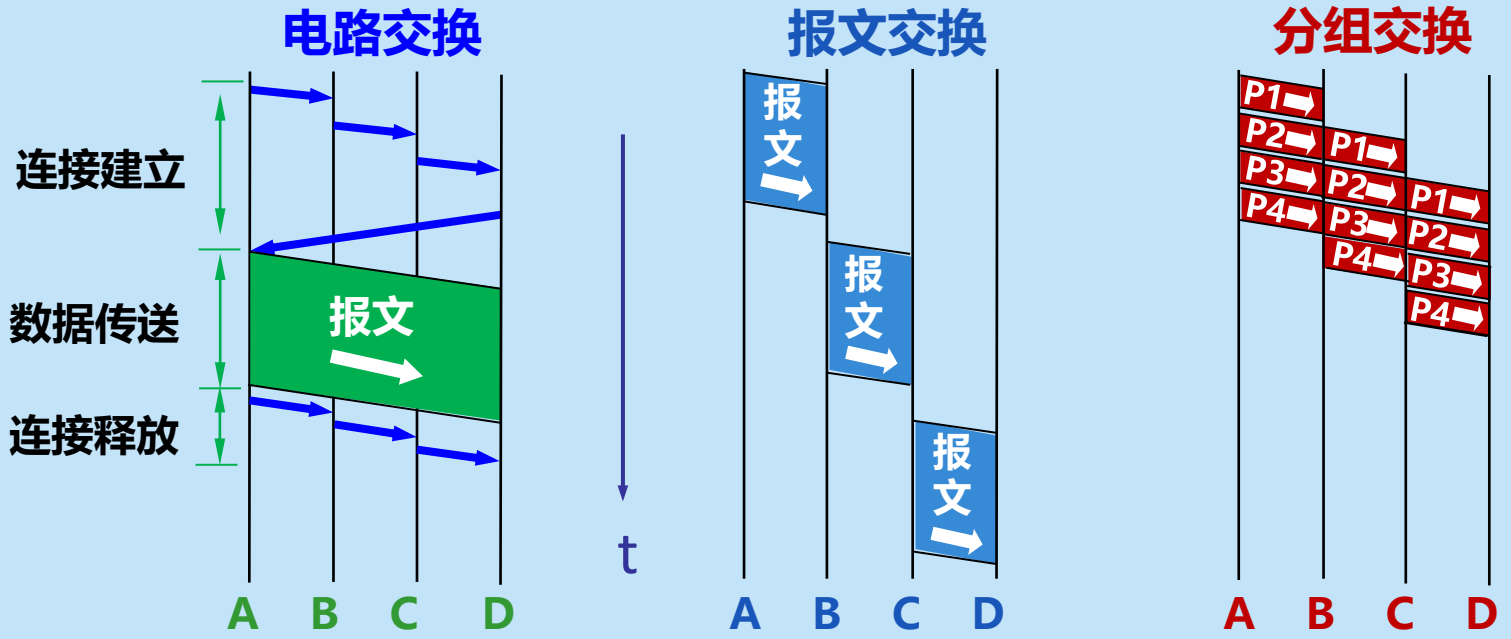
网络基础课程要点

路由器



数据传送特点

比特流直达终点

报文 报文 报文

存储转发 存储转发

分组 分组 分组

存储转发 存储转发

4. 时延

- 时延 (delay 或 latency) 是指数据（一个报文或分组，甚至比特）从网络（或链路）的一端传送到另一端所需的时间。
- 有时也称为**延迟**或迟延。
- 网络中的时延由以下几个不同的部分组成：
  ① 发送时延
  ② 传播时延
  ③ 处理时延
  ④ 排队时延

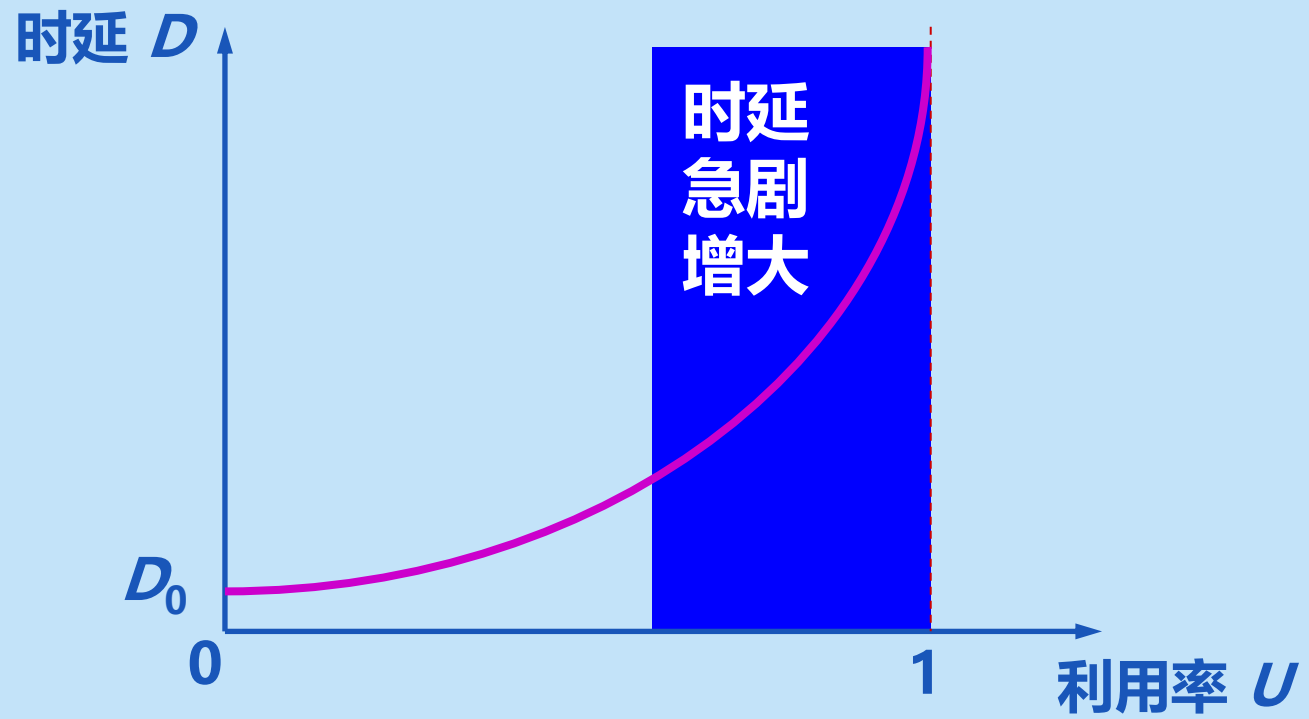
时延与网络利用率的关系

- 根据排队论的理论，当某信道的利用率增大时，该信道引起的时延也就迅速增加。
- 若令 D_0 表示网络空闲时的时延，D 表示网络当前的时延，则在适当的假定条件下，可以用下面的简单公式表示 D 和 D_0 之间的关系：

$$D = \frac{D_0}{1 - U}$$

其中：U 是网络的利用率，数值在 0 到 1 之间。

时延与网络利用率的关系



当信道的利用率增大时，该信道引起的时延迅速增加。

网络协议的三个组成要素

- **语法**：数据与控制信息的结构或格式 。
- **语义**：需要发出何种控制信息，完成何种动作以及做出何种响应。
- **同步**：事件实现顺序的详细说明。

由此可见，网络协议是计算机网络的不可缺少的组成部分。

分层的好处与缺点

好处

- 各层之间是独立的。
- 灵活性好。
- 结构上可分割开。
- 易于实现和维护。
- 能促进标准化工作。

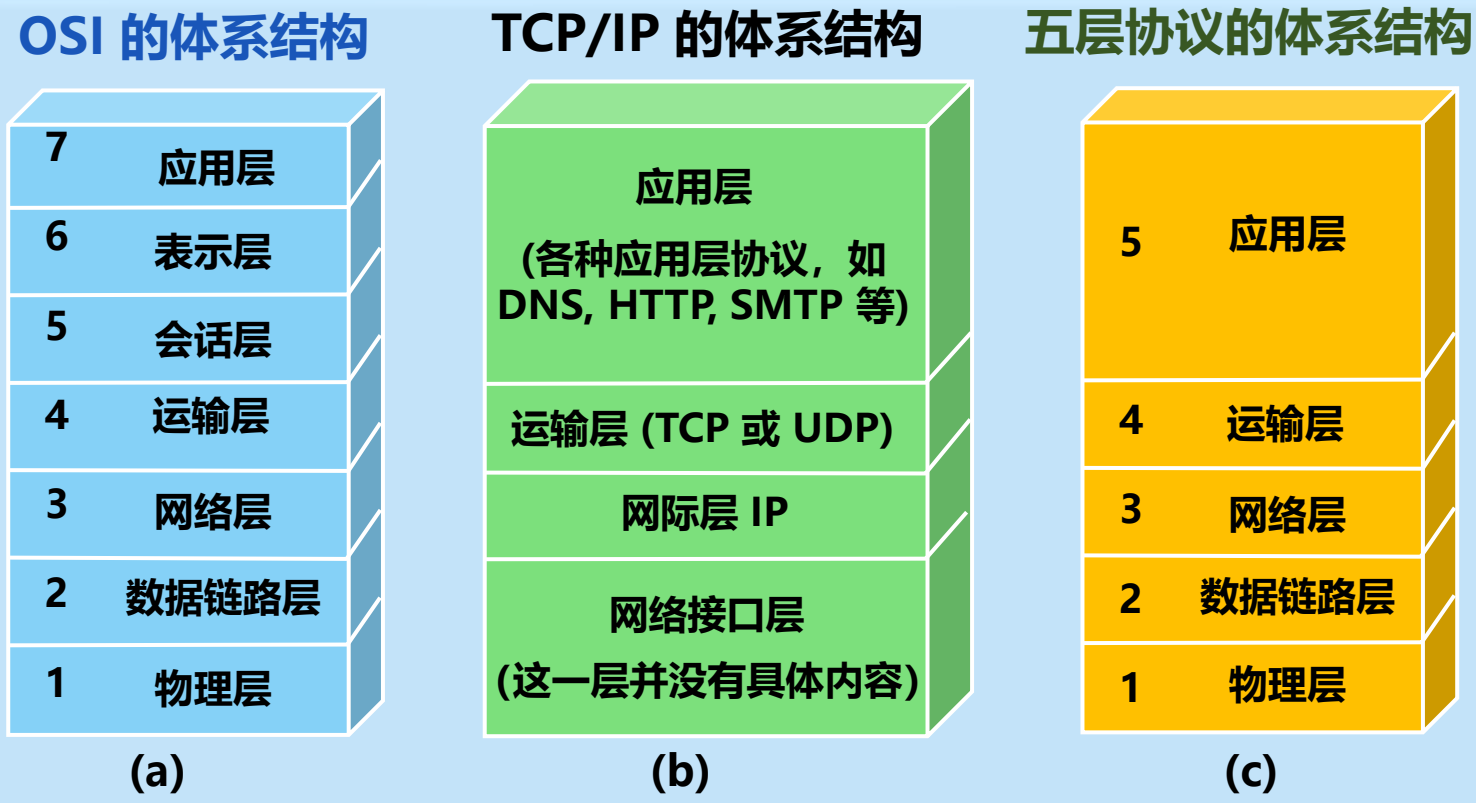
缺点

- 降低效率。
- 有些功能会在不同的层次中重复出现，因而产生了额外开销。

各层完成的主要功能

- **差错控制：**使相应层次对等方的通信更加可靠。
- **流量控制：**发送端的发送速率必须使接收端来得及接收，不要太快。
- **分段和重装：**发送端将要发送的数据块划分为更小的单位，在接收端将其还原。
- **复用和分用：**发送端几个高层会话复用一条低层的连接，在接收端再进行分用。
- **连接建立和释放：**交换数据前先建立一条逻辑连接，数据传送结束后释放连接。

1.7.3 具有五层协议的体系结构



计算机网络体系结构：(a) OSI 的七层协议；(b) TCP/IP 的四层协议；(c) 五层协议

物理层

(2) 信噪比

- 1984年，香农 (Shannon) 用信息论的理论推导出了带宽受限且有高斯白噪声干扰的信道的极限、无差错的信息传输速率（香农公式）。
- 信道的极限信息传输速率 C 可表达为：

$$C = W\log_2(1+S/N) \quad \text{(bit/s)}$$

其中：W 为信道的带宽（以 Hz 为单位）；

S 为信道内所传信号的平均功率；

N 为信道内部的高斯噪声功率。

2.3 物理层下面的传输媒体

- 传输媒体也称为传输介质或传输媒介，它就是数据传输系统中在发送器和接收器之间的物理通路。
- 传输媒体可分为两大类，即导引型传输媒体和非导引型传输媒体。
- 在导引型传输媒体中，电磁波被导引沿着固体媒体（铜线或光纤）传播。
- 非导引型传输媒体就是指自由空间。在非导引型传输媒体中，电磁波的传输常称为无线传输。

2.3 物理层下面的传输媒体

- 传输媒体也称为传输介质或传输媒介，它就是数据传输系统中在发送器和接收器之间的物理通路。
- 传输媒体可分为两大类，即导引型传输媒体和非导引型传输媒体。
- 在导引型传输媒体中，电磁波被导引沿着固体媒体（铜线或光纤）传播。
- 非导引型传输媒体就是指自由空间。在非导引型传输媒体中，电磁波的传输常称为无线传输。

码片序列的正交关系

- 令向量 S 表示站 S 的码片向量，令 T 表示其他任何站的码片向量。
- 两个不同站的码片序列正交，就是向量 S 和T 的规格化内积 (inner product) 等于 0：

$$\mathbf{S} \bullet \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^{m} S_i T_i = 0$$

正交关系的另一个重要特性

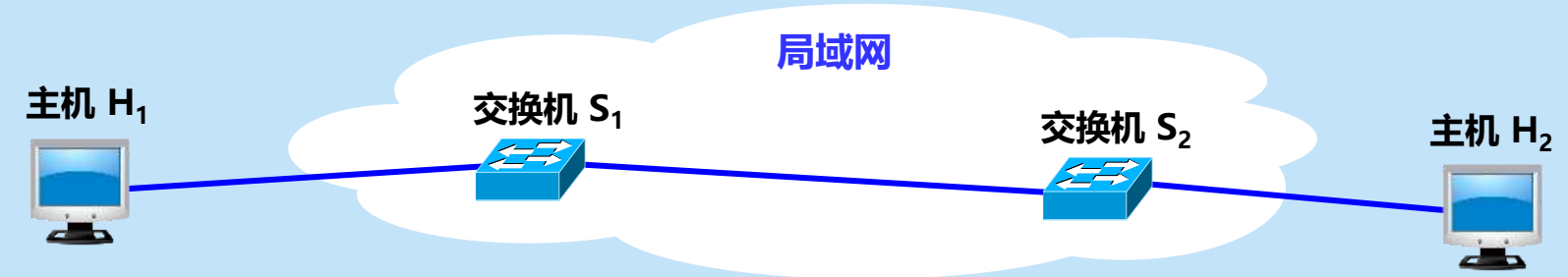
- 任何一个码片向量和该码片向量自己的规格化内积都是 1 。

$$\mathbf{S} \bullet \mathbf{S} = \frac{1}{m}\sum_{i=1}^{m} S_i S_i = \frac{1}{m}\sum_{i=1}^{m} S_i^2 = \frac{1}{m}\sum_{i=1}^{m} (\pm 1)^2 = 1$$

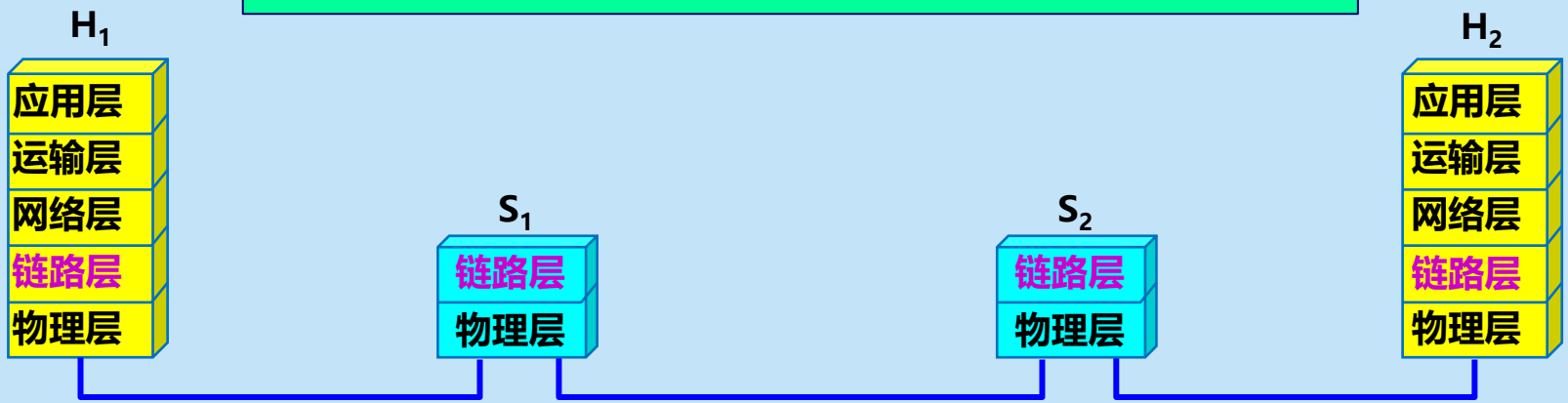
- 一个码片向量和该码片反码的向量的规格化内积值是 –1。

链路层

数据链路层是实现设备之间通信的非常重要的一层



局域网中的主机、交换机等都必须实现数据链路层



3.1.2 三个基本问题

- 数据链路层协议有许多种，但有三个基本问题则是共同的。这三个基本问题是：
  1. 封装成帧
  2. 透明传输
  3. 差错控制

2. PPP 协议不需要的功能

- 纠错
- 流量控制
- 序号
- 多点线路
- 半双工或单工链路

2. 适配器的作用

- 网络接口板又称为通信适配器 (adapter) 或网络接口卡 NIC (Network Interface Card)，或“网卡”。
- 适配器的重要功能：
  1. 进行串行/并行转换。
  2. 对数据进行缓存。
  3. 在计算机的操作系统安装设备驱动程序。
  4. 实现以太网协议。

以太网提供的服务

- CSMA/CD 含义：**载波监听多点接入 / 碰撞检测**（Carrier Sense Multiple Access with Collision Detection）。
- “**多点接入**”表示许多计算机以多点接入的方式连接在一根总线上。
- “**载波监听**”是指每一个站在发送数据之前先要检测一下总线上是否有其他计算机在发送数据，如果有，则暂时不要发送数据，以免发生碰撞。
- 总线上并没有什么“载波”。因此，**“载波监听”就是用电子技术检测总线上有没有其他计算机发送的数据信号。**

检测到碰撞后

- 在发生碰撞时，总线上传输的信号产生了严重的失真，无法从中恢复出有用的信息来。
- 每一个正在发送数据的站，一旦发现总线上出现了碰撞，就要立即停止发送，免得继续浪费网络资源，然后等待一段随机时间后再次发送。

二进制指数类型退避算法 (truncated binary exponential type)

- 发生碰撞的站在停止发送数据后，要推迟（退避）一个随机时间才能再发送数据。
  1. 基本退避时间取为争用期 2τ。
  2. 从整数集合 [0, 1, … , $(2^k - 1)$] 中随机地取出一个数，记为 r。重传所需的时延就是 r 倍的基本退避时间。
  3. 参数 k 按下面的公式计算：

     $$k = \text{Min}[\text{重传次数}, 10]$$

  4. 当 $k \le 10$ 时，参数 k 等于重传次数。
  5. 当重传达 16 次仍不能成功时即丢弃该帧，并向高层报告。

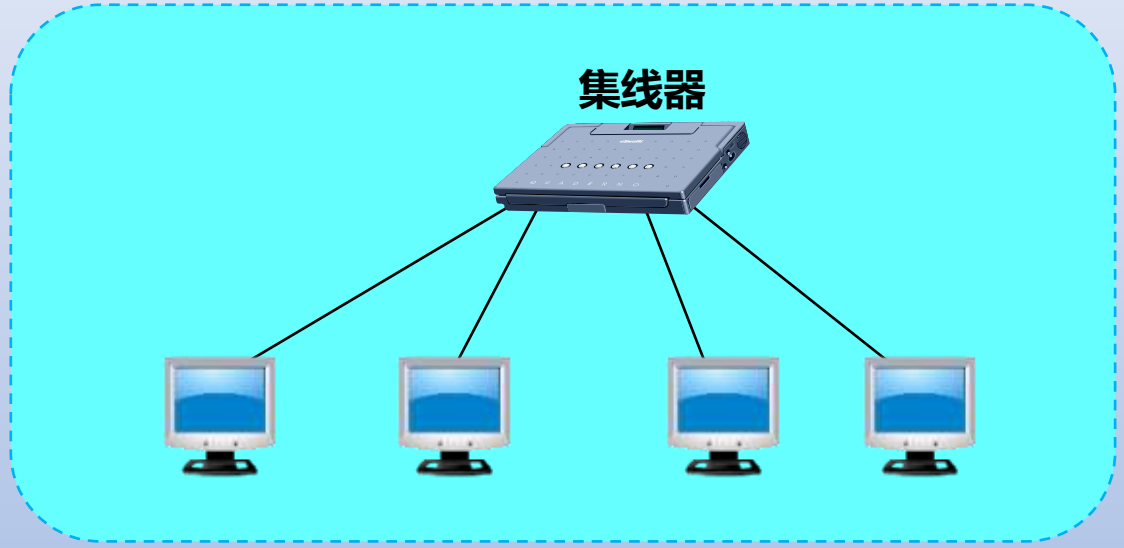
1．MAC 层的硬件地址

- 在局域网中，**硬件地址**又称为**物理地址**，或 **MAC 地址**。
- 802 标准所说的“地址”严格地讲应当是每一个站的“**名字**”或**标识符**。
- 但鉴于大家都早已习惯了将这种 48 位的“名字”称为“地址”，所以本书也采用这种习惯用法，尽管这种说法并不太严格。

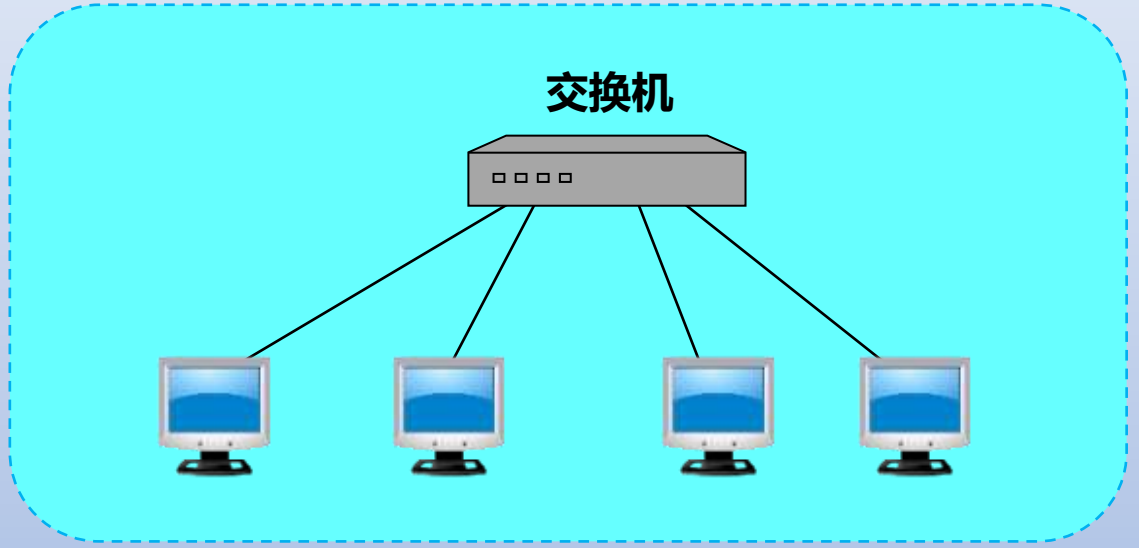
请注意，如果连接在局域网上的主机或路由器安装有多个适配器，那么这样的主机或路由器就有多个“地址”。更准确些说，这种 48 位“地址”应当是某个接口的标识符。

以太网交换机的优点

- 用户独享带宽，增加了总容量。



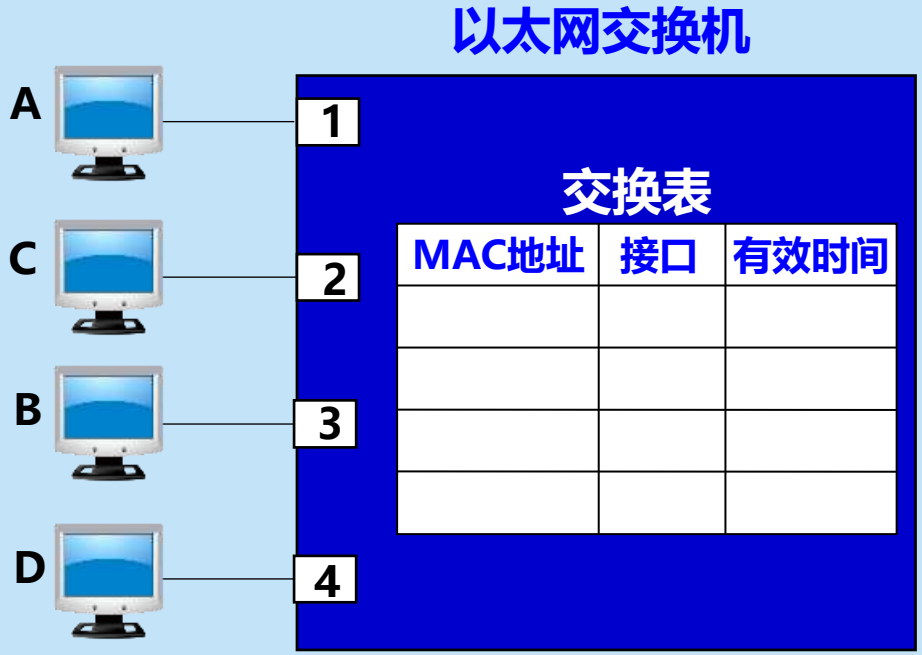
- N 个用户共享集线器提供的带宽 B。
- 平均每个用户仅占有 B/N 的带宽。



- 交换机为每个端口提供带宽 B。
- N个用户，每个用户独占带宽 B。
- 交换机总带宽达 B × N 。

2. 以太网交换机的自学习功能

- 以太网交换机运行自学习算法自动维护交换表。



开始时，交换表是空的

PPPoE

- **PPPoE** (PPP over Ethernet) 的意思是“在以太网上运行 PPP”，它把 PPP 协议与以太网协议结合起来 —— 将 PPP 帧再封装到以太网中来传输。
- 现在的光纤宽带接入 FTTx 都要使用 PPPoE 的方式进行接入。在 PPPoE 弹出的窗口中键入在网络运营商购买的用户名和密码，就可以进行宽带上网了。
- 利用 ADSL 进行宽带上网时，从用户个人电脑到家中的 ADSL 调制解调器之间，也是使用 RJ-45 和 5 类线（即以太网使用的网线）进行连接的，并且也是使用 PPPoE 弹出的窗口进行拨号连接的。

9.1.3 802.11 局域网的 MAC 层协议

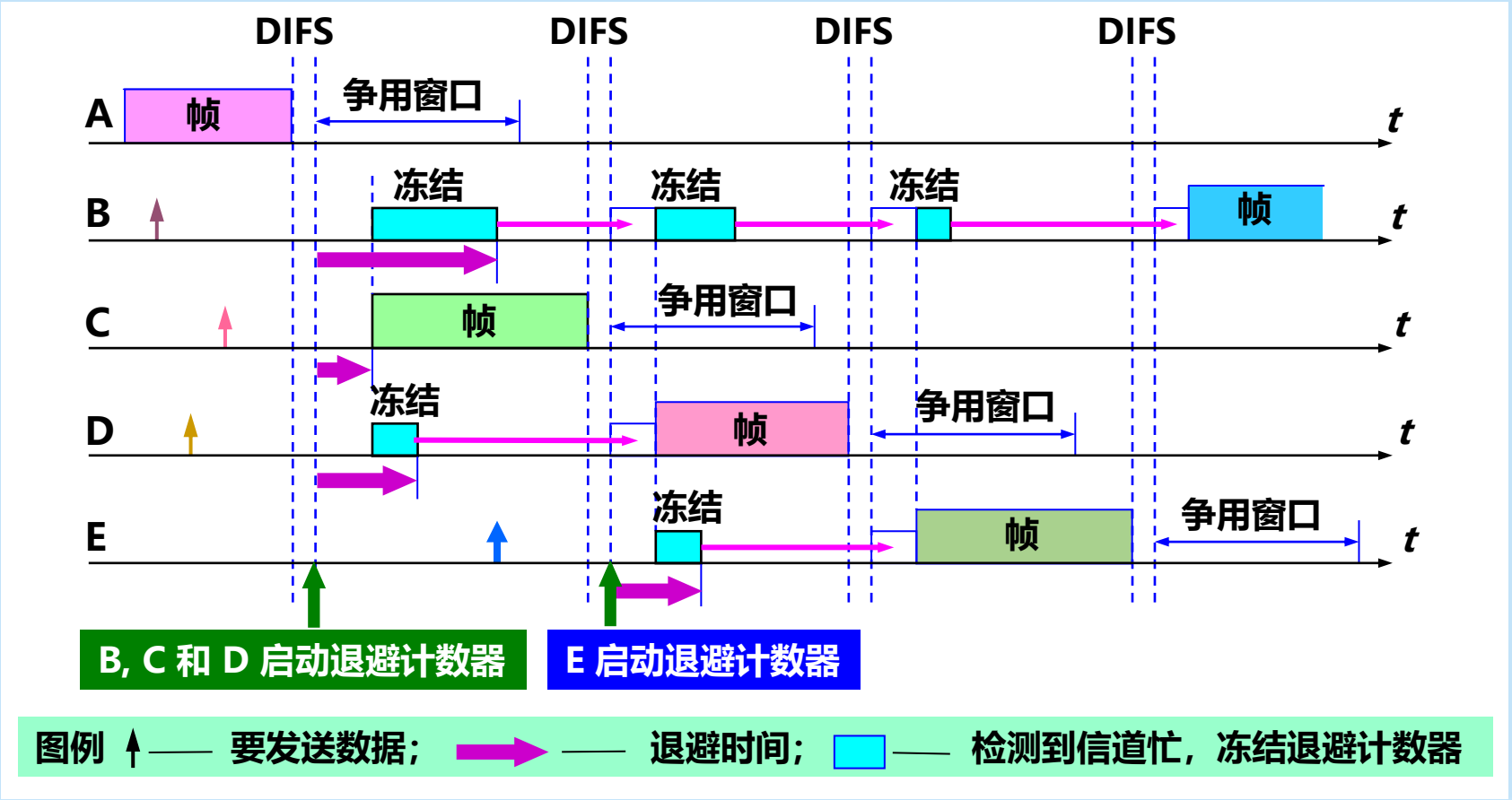
1. CSMA/CA 协议

- 无线局域网不能简单地搬用 CSMA/CD 协议。因为：
  1. “碰撞检测”要求一个站点在发送本站数据的同时，还必须不间断地检测信道，但接收到的信号强度往往会远远小于发送信号的强度，在无线局域网的设备中要实现这种功能就**花费过大**。
  2. 即使能够实现碰撞检测的功能，并且在发送数据时检测到信道是空闲的时候，在接收端仍然有**可能发生碰撞**。

CSMA/CA 协议

- 无线局域网不能使用 CSMA/CD，而只能使用改进的 CSMA 协议。
- 改进的办法是把 CSMA 增加一个**碰撞避免** CA (Collision Avoidance) 功能。
- **802.11 就使用 CSMA/CA 协议**。在使用 CSMA/CA 的同时，还增加使用**停止等待**协议。

802.11 的退避机制


DIFS
DIFS
DIFS
DIFS
A
帧
争用窗口
t
B
冻结
冻结
冻结
帧
t
C
帧
争用窗口
t
D
冻结
帧
争用窗口
t
E
冻结
帧
争用窗口
t
B, C 和 D 启动退避计数器
E 启动退避计数器
图例 —— 要发送数据； —— 退避时间； —— 检测到信道忙，冻结退避计数器

CSMA/CA算法归纳

① 若站点最初有数据要发送（而不是发送不成功再进行重传），且检测到信道空闲，在等待时间 DIFS 后，就发送整个数据帧。

② 否则，站点就要等检测到信道空闲并经过时间 DIFS 后，执行 CSMA/CA 协议的退避算法，启动退避计数器。在退避计数器减少到零之前，一旦检测到信道忙，就冻结退避计时器。一旦信道空闲，退避计时器就进行倒计时。

③ 当退避计时器时间减少到零时（这时信道只可能是空闲的），站点就发送整个的帧并等待确认。

④ 发送站若收到确认，就知道已发送的帧被目的站正确收到了。这时如果要发送第二帧，就要从上面的步骤 (2) 开始，执行 CSMA/CA 协议的退避算法，随机选定一段退避时间。若源站在规定时间内没有收到确认帧 ACK（由重传计时器控制这段时间），就必须重传此帧 （再次使用 CSMA/CA 协议争用接入信道），直到收到确认为止，或者经过若干次的重传失败后放弃发送。

网络层

另一种观点：网络提供数据报服务

- 互联网的先驱者提出了一种崭新的网络设计思路。
- 网络层向上只提供简单灵活的、**无连接的、尽最大努力交付**的**数据报服务**。
- 网络在发送分组时不需要先建立连接。每一个分组（即 IP 数据报）独立发送，与其前后的分组无关（不进行编号）。
- **网络层不提供服务质量的承诺**。即所传送的分组可能出错、丢失、重复和失序（不按序到达终点），当然也不保证分组传送的时限。

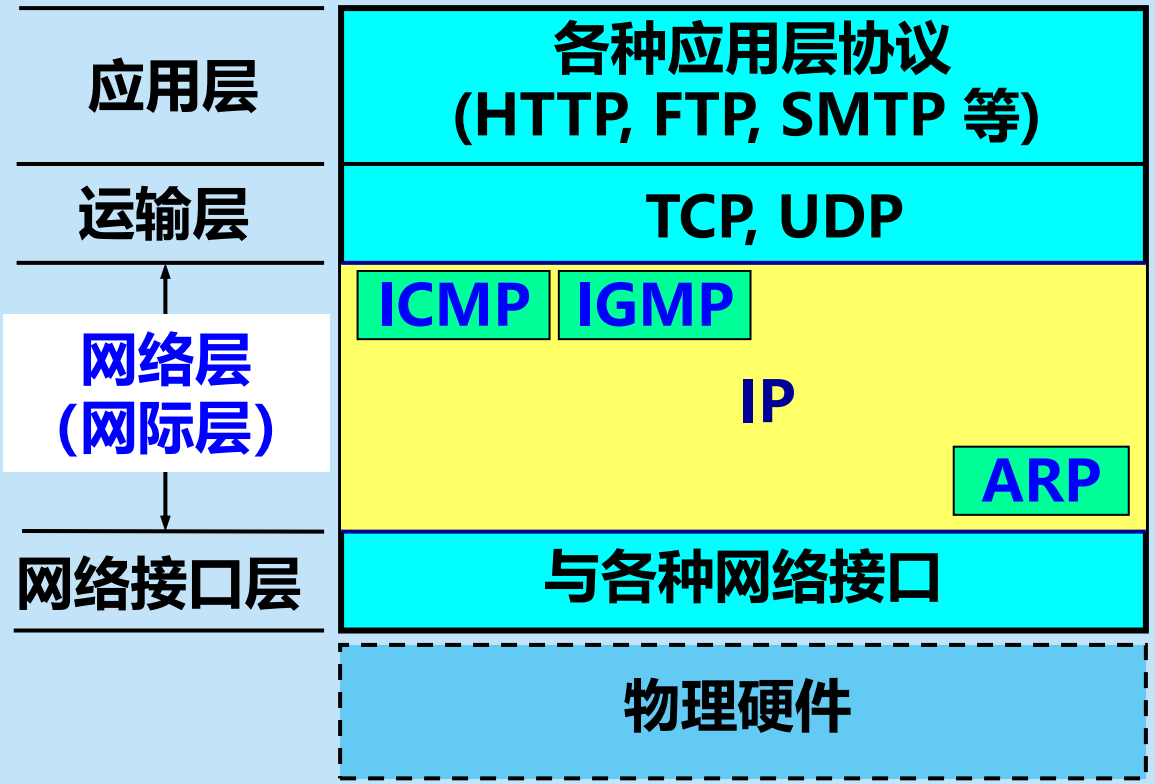
尽最大努力交付

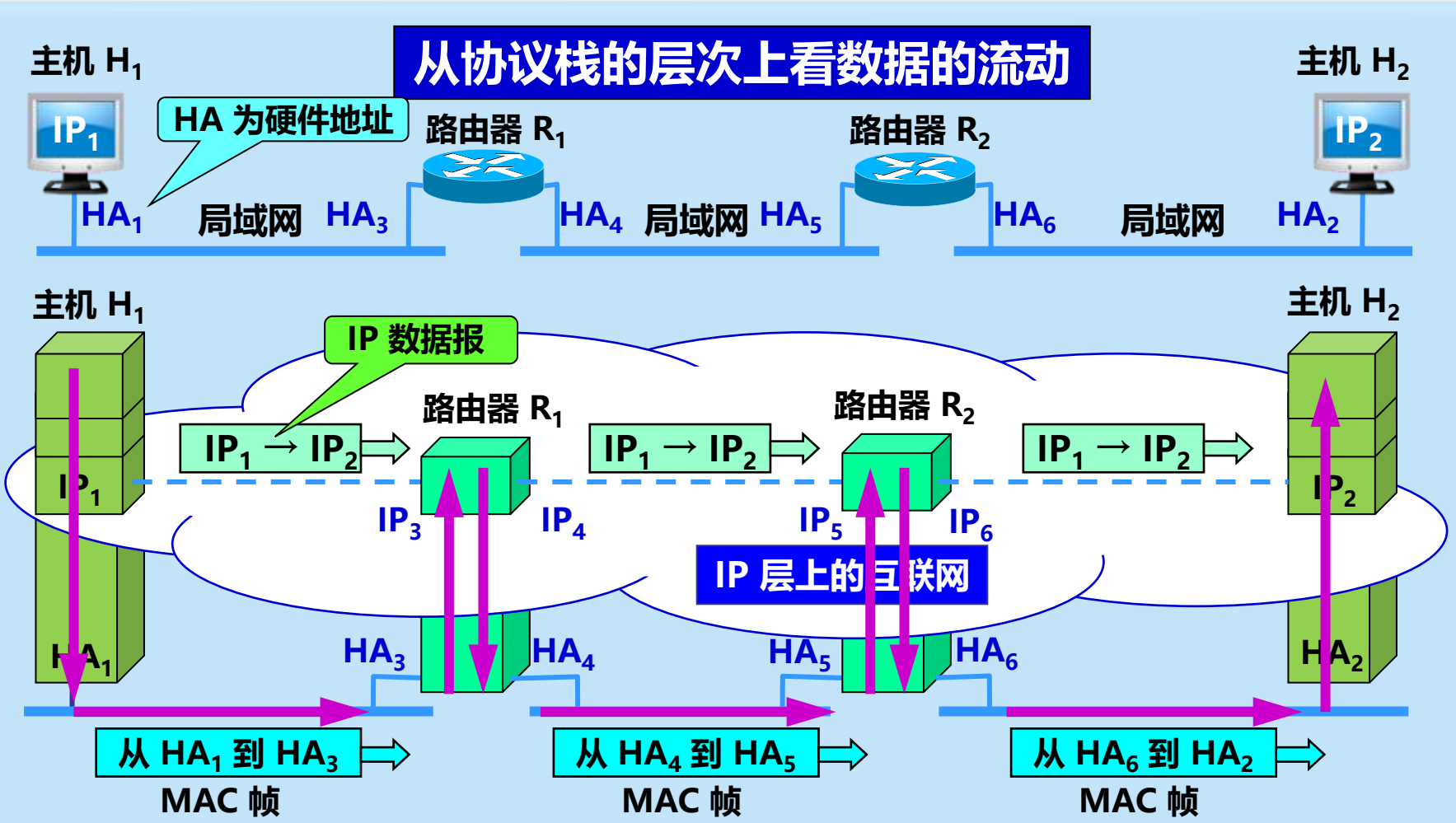
- 由于传输网络不提供端到端的可靠传输服务，这就使网络中的路由器可以做得比较简单，而且价格低廉（与电信网的交换机相比较）。
- 如果主机（即端系统）中的进程之间的通信需要是可靠的，那么就由网络的主机中的运输层负责可靠交付（包括差错处理、流量控制等）。
- 采用这种设计思路的好处是：网络的造价大大降低，运行方式灵活，能够适应多种应用。
- 互连网能够发展到今日的规模，充分证明了当初采用这种设计思路的正确性。

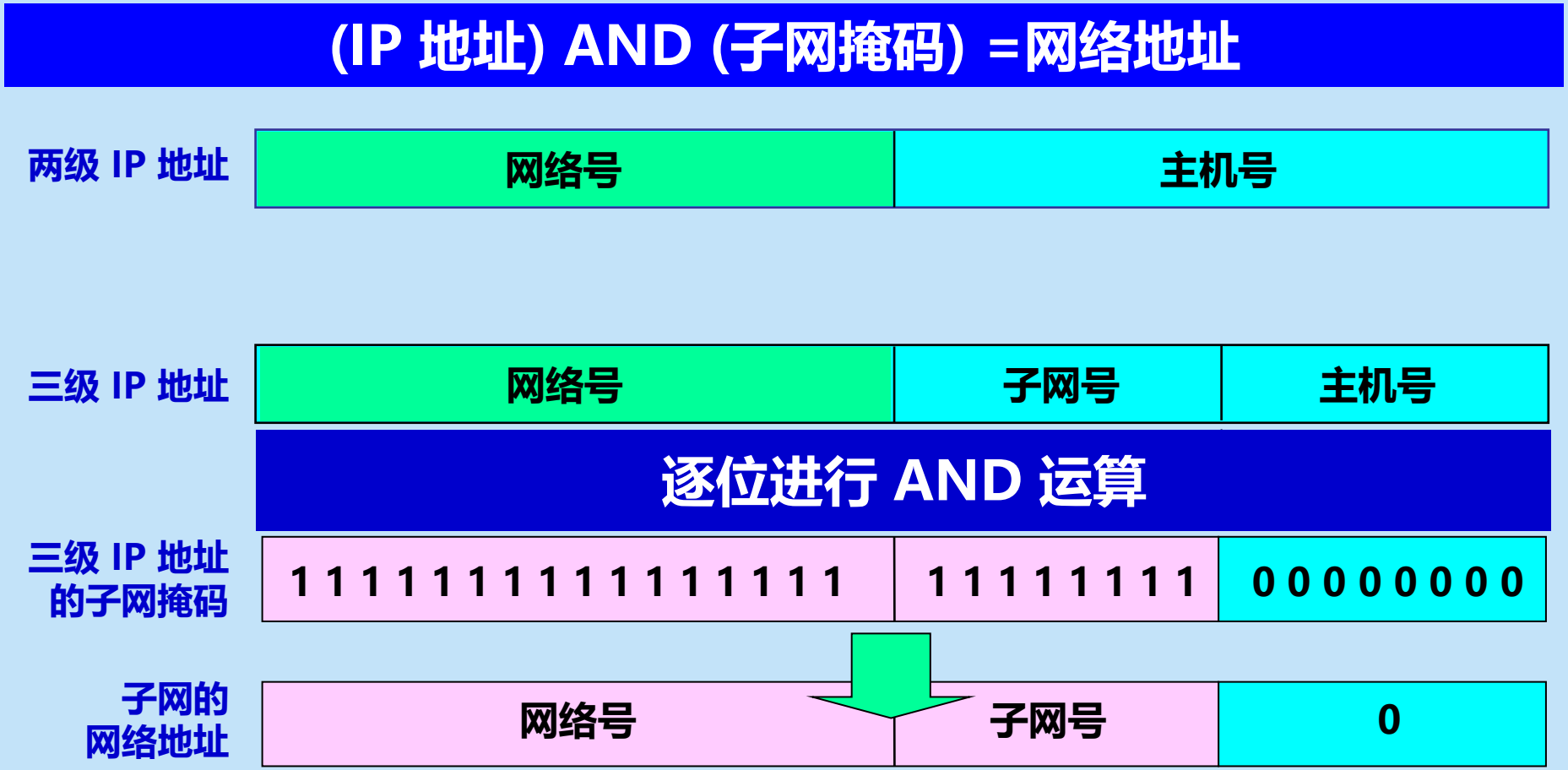
虚电路服务与数据报服务的对比

对比的方面	虚电路服务	数据报服务
思路	可靠通信应当由网络来保证	可靠通信应当由用户主机来保证
连接的建立	必须有	不需要
终点地址	仅在连接建立阶段使用，每个分组使用短的虚电路号	每个分组都有终点的完整地址
分组的转发	属于同一条虚电路的分组均按照同一路由进行转发	每个分组独立选择路由进行转发
当结点出故障时	所有通过出故障的结点的虚电路均不能工作	出故障的结点可能会丢失分组，一些路由可能会发生变化
分组的顺序	总是按发送顺序到达终点	到达终点时不一定按发送顺序
端到端的差错处理和流量控制	可以由网络负责，也可以由用户主机负责	由用户主机负责

网际层的 IP 协议及配套协议




从协议栈的层次上看数据的流动
主机 H1
主机 H2
IP1
IP2
HA 为硬件地址
路由器 R1
路由器 R2
HA1
局域网
HA3
HA4
局域网
HA5
HA6
局域网
HA2
IP 数据报
IP1 → IP2
IP1 → IP2
IP1 → IP2
IP3
IP4
IP5
IP6
IP 层上的互联网
从 HA1 到 HA3
从 HA4 到 HA5
从 HA6 到 HA2
MAC 帧
MAC 帧
MAC 帧


(IP 地址) AND (子网掩码) =网络地址
两级 IP 地址
网络号
主机号
三级 IP 地址
网络号
子网号
主机号
逐位进行 AND 运算
三级 IP 地址
的子网掩码
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0
子网的
网络地址
网络号
子网号
0

默认子网掩码


A类地址
网络地址
网络号
主机号为全 0
默认子网掩码
255.0.0.0
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
B类地址
网络地址
网络号
主机号为全 0
默认子网掩码
255.255.0.0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
C类地址
网络地址
网络号
主机号为全 0
默认子网掩码
255.255.255.0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0

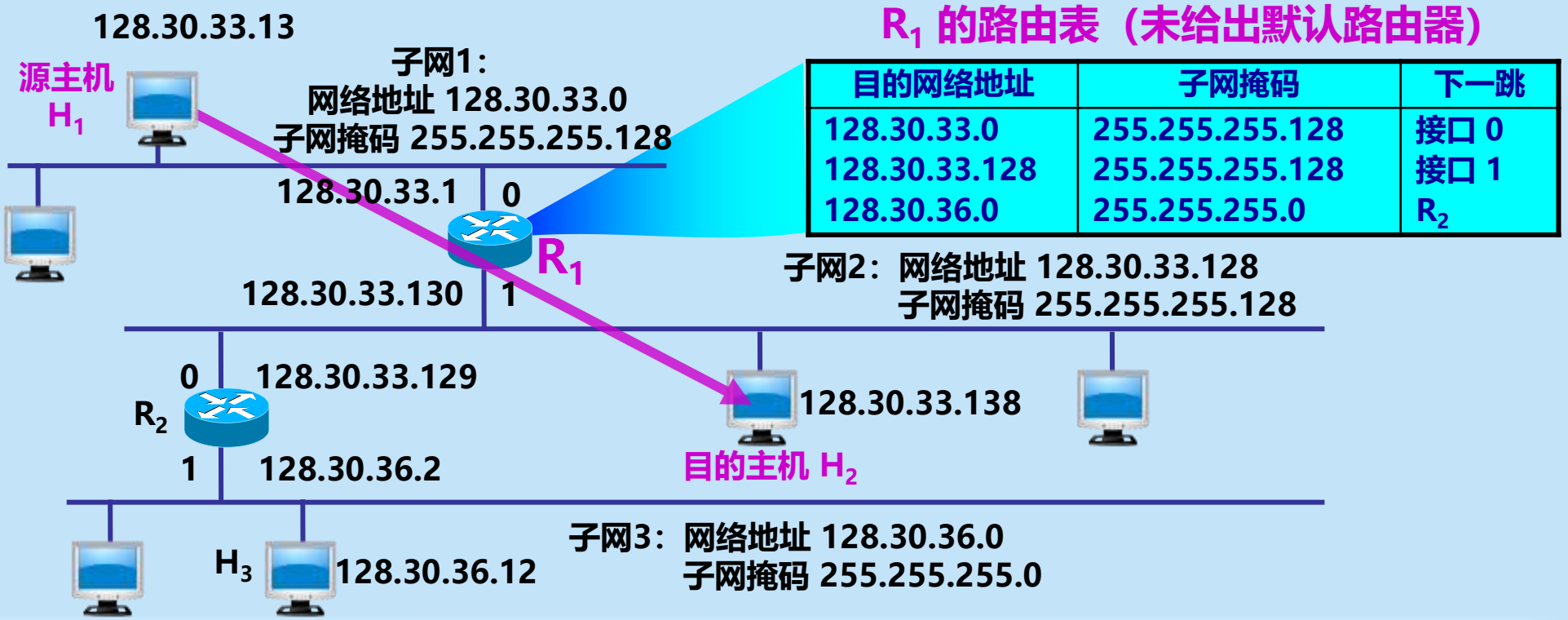
子网掩码是一个重要属性

- **子网掩码是一个网络或一个子网的重要属性。**
- **路由器在和相邻路由器交换路由信息时，必须把自己所在网络（或子网）的子网掩码告诉相邻路由器。**
- **路由器的路由表中的每一个项目，除了要给出目的网络地址外，还必须同时给出该网络的子网掩码。**
- **若一个路由器连接在两个子网上，就拥有两个网络地址和两个子网掩码。**

子网划分方法

- 有固定长度子网和变长子网两种子网划分方法。
- 在采用固定长度子网时，所划分的所有子网的子网掩码都是相同的。
- 虽然根据已成为互联网标准协议的 RFC 950 文档，子网号不能为全 1 或全 0，但随着无分类域间路由选择 CIDR 的广泛使用，现在全 1 和全 0 的子网号也可以使用了，但一定要谨慎使用，确认你的路由器所用的路由选择软件是否支持全 0 或全 1 的子网号这种较新的用法。
- 划分子网增加了灵活性，但却减少了能够连接在网络上的主机总数。

【例4-4】已知互联网和路由器 R_1 中的路由表。
主机 H_1 向 H_2 发送分组。
试讨论 R_1 收到 H_1 向 H_2 发送的分组后查找路由表的过程。

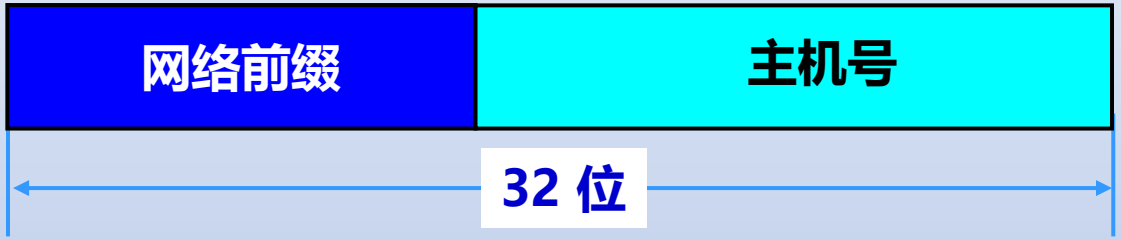


R_1 的路由表（未给出默认路由器）

目的网络地址	子网掩码	下一跳
128.30.33.0	255.255.255.128	接口 0
128.30.33.128	255.255.255.128	接口 1
128.30.36.0	255.255.255.0	R_2

无分类的两级编址

- 无分类的两级编址的记法是：



IP地址 ::= {<网络前缀>，<主机号>}　　(4-3)

- CIDR 使用“斜线记法”(slash notation)，它又称为 CIDR 记法，即在 IP 地址面加上一个斜线“/”，然后写上网络前缀所占的位数（这个数值对应于三级编址中子网掩码中 1 的个数）。例如： 220.78.168.0/24

CIDR 地址块

- CIDR 把网络前缀都相同的连续的 IP 地址组成“CIDR 地址块”。
- 128.14.32.0/20 表示的地址块共有 2^{12} 个地址（因为斜线后面的 20 是网络前缀的位数，所以这个地址的主机号是 12 位）。
  1. 这个地址块的起始地址是 128.14.32.0。
  2. 在不需要指出地址块的起始地址时，也可将这样的地址块简称为“/20 地址块”。
  3. 128.14.32.0/20 地址块的最小地址：128.14.32.0
  4. 128.14.32.0/20 地址块的最大地址：128.14.47.255
  5. 全 0 和全 1 的主机号地址一般不使用。

2. 最长前缀匹配

- 使用 CIDR 时，路由表中的每个项目由“网络前缀”和“下一跳地址”组成。**在查找路由表时可能会得到不止一个匹配结果。**
- 应当从匹配结果中选择具有最长网络前缀的路由：**最长前缀匹配** (longest-prefix matching)。
- 网络前缀越长，其地址块就越小，因而路由就越具体 (more specific) 。
- 最长前缀匹配又称为**最长匹配**或**最佳匹配**。

最长前缀匹配举例

收到的分组的目的地址 *D* = 206.0.71.130

路由表中的项目：206.0.68.0/22　　1

206.0.71.128/25　　2

查找路由表中的第 1 个项目：

第 1 个项目 206.0.68.0/22 的掩码 *M* 有 22 个连续的 1。

M = 11111111 11111111 11111100 00000000

因此只需把 *D* 的第 3 个字节转换成二进制。

	M =	11111111	11111111	11111100	00000000
AND	*D* =	206.	0.	01000111.	130
		206.	0.	01000100.	0

与 206.0.68.0/22 匹配!

最长前缀匹配举例

收到的分组的目的地址 *D* = 206.0.71.130

路由表中的项目：206.0.68.0/22　　1

206.0.71.128/25　　2

查找路由表中的第 2 个项目：

第 2 个项目 206.0.71.128/25 的掩码 M 有 25 个连续的 1。

M = 11111111 11111111 11111111 10000000

因此只需把 *D* 的第 4 个字节转换成二进制。

	M =	11111111	11111111	11111111	10000000
AND	D =	206.	0.	71.	10000010
		206.	0.	71.	10000000

与 206.0.71.128/25 匹配!

最长前缀匹配举例

D AND (11111111 11111111 11111100 00000000)
= 206.0.68.0/22　　匹配

D AND (11111111 11111111 11111111 10000000)
= 206.0.71.128/25　　匹配

选择两个匹配的地址中更具体的一个，即选择最长前缀的地址。

4.5.2 内部网关协议 RIP

1. 工作原理

- 路由信息协议 RIP (Routing Information Protocol) 是内部网关协议 IGP 中最先得到广泛使用的协议。
- RIP 是一种分布式的、基于距离向量的路由选择协议。
- RIP 协议要求网络中的每一个路由器都要维护从它自己到其他每一个目的网络的距离记录。

RIP 协议的三个特点

1. 仅和相邻路由器交换信息。
2. 交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
3. 按固定的时间间隔交换路由信息，例如，每隔 30 秒。当网络拓扑发生变化时，路由器也及时向相邻路由器通告拓扑变化后的路由信息。

路由表的建立

- 路由器在刚刚开始工作时，只知道到直接连接的网络的距离（此距离定义为 1）。它的路由表是空的。
- 以后，每一个路由器也只和数目非常有限的相邻路由器交换并更新路由信息。
- 经过若干次更新后，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。
- RIP 协议的收敛 (convergence) 过程较快。“收敛”就是在自治系统中所有的结点都得到正确的路由选择信息的过程。

2. 距离向量算法

路由器收到相邻路由器（其地址为 X）的一个 RIP 报文：

(1) 先修改此 RIP 报文中的所有项目：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。

(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：

若项目中的目的网络不在路由表中，则把该项目加到路由表中。

否则

若下一跳字段给出的路由器地址是同样的，则把收到的项目替换原路由表中的项目。

否则

若收到项目中的距离小于路由表中的距离，则进行更新，

否则，什么也不做。

(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为 16（表示不可达）。

(4) 返回。

【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) R4 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

计算更新

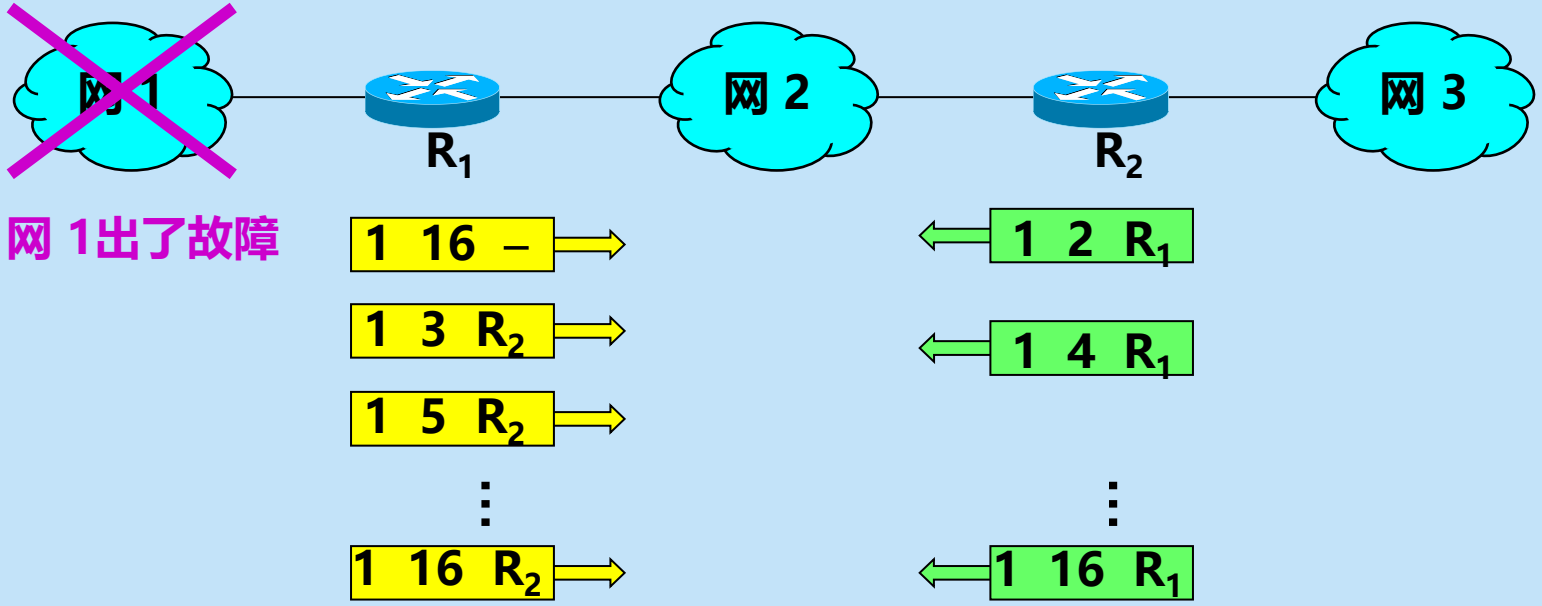
表 4-9(c) 修改后的表 4-9(b)

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

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
…	…	…

这就是好消息传播得快，而坏消息传播得慢。网络出故障的传播时间往往需要较长的时间(例如数分钟)。这是 RIP 的一个主要缺点。

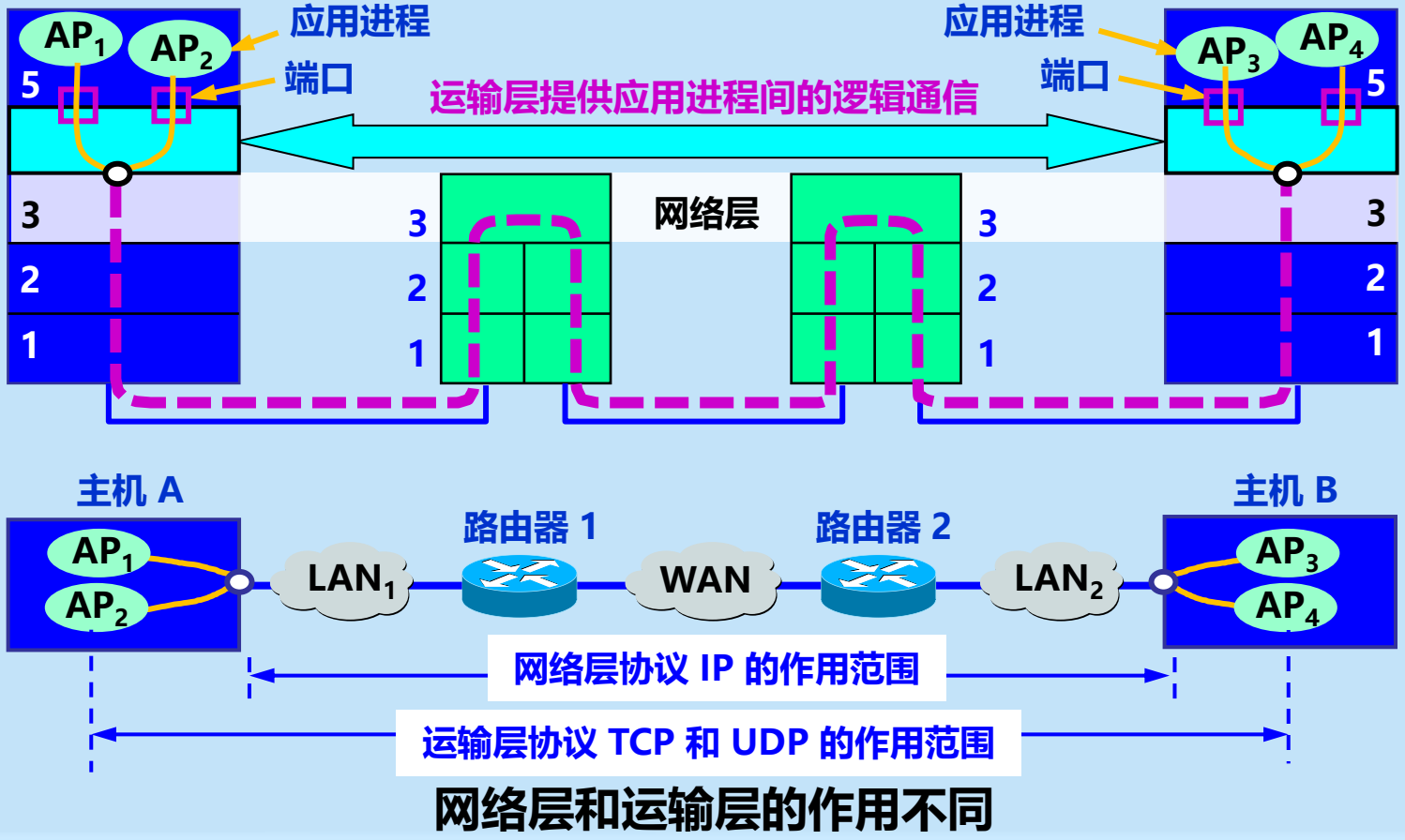


这样不断更新下去，直到 R_1 和 R_2 到网 1 的距离都增大到 16 时，R_1 和 R_2才知道网 1 是不可达的。

运输层

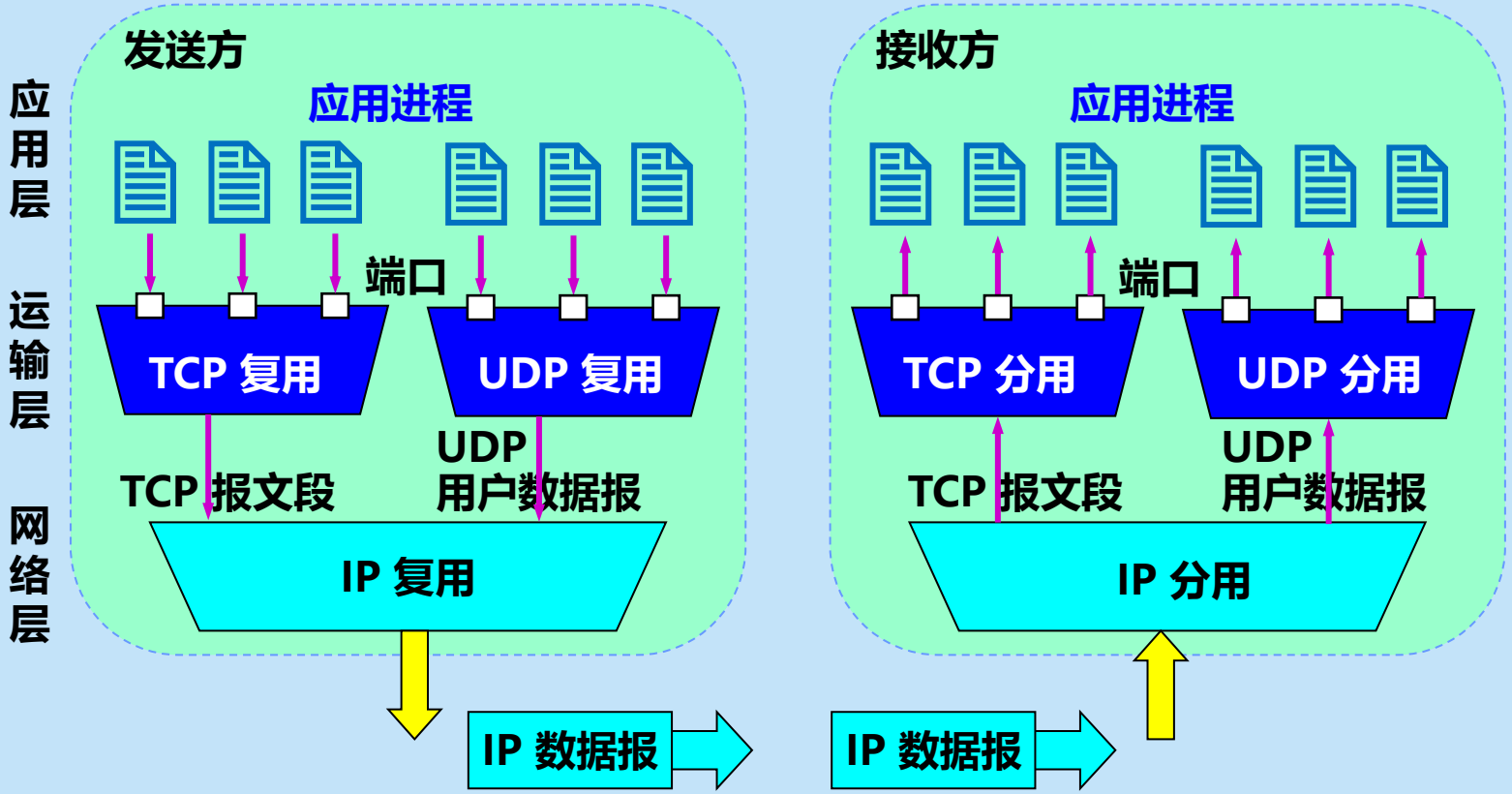
网络层和运输层有明显的区别

网络层是为主机之间提供逻辑通信，运输层为应用进程之间提供端到端的逻辑通信

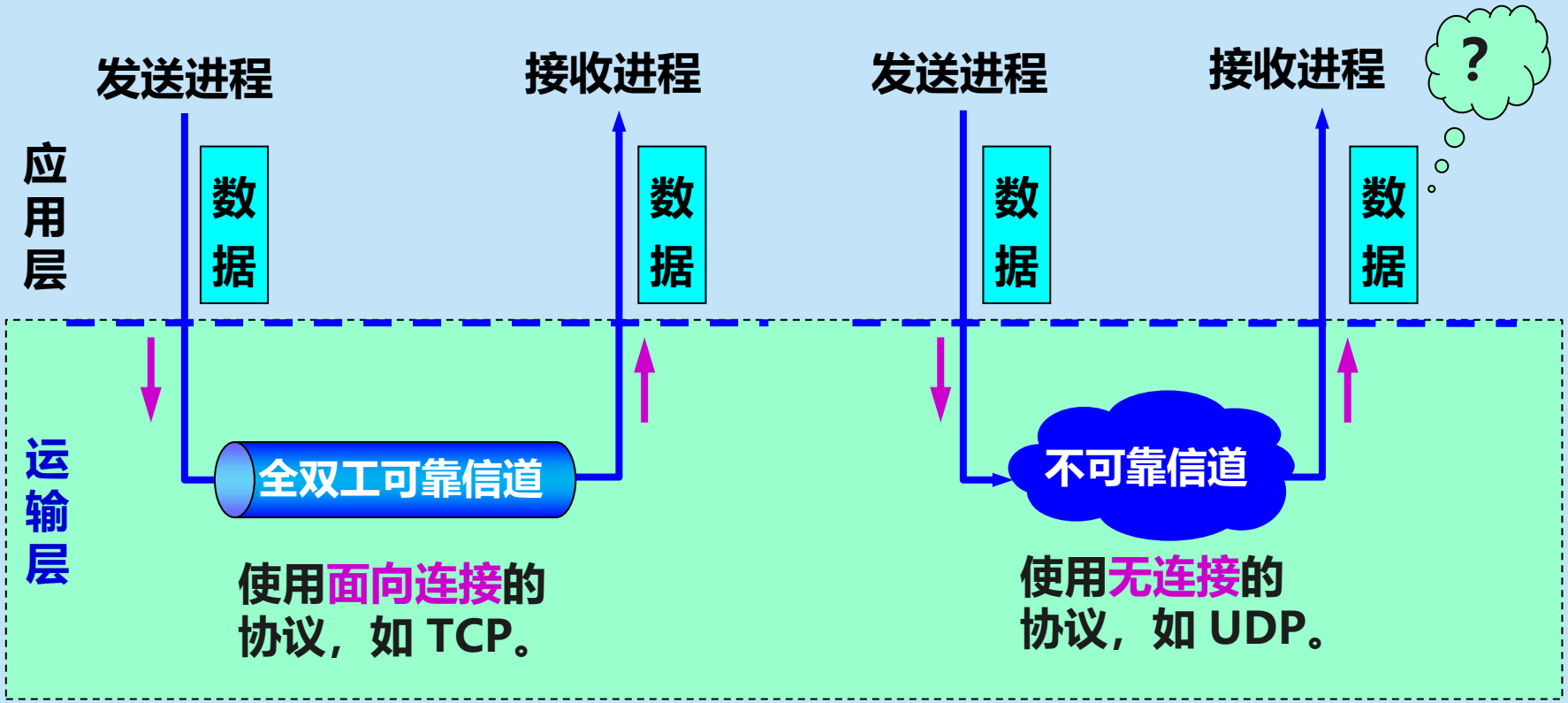


网络层和运输层的作用不同

基于端口的复用和分用功能



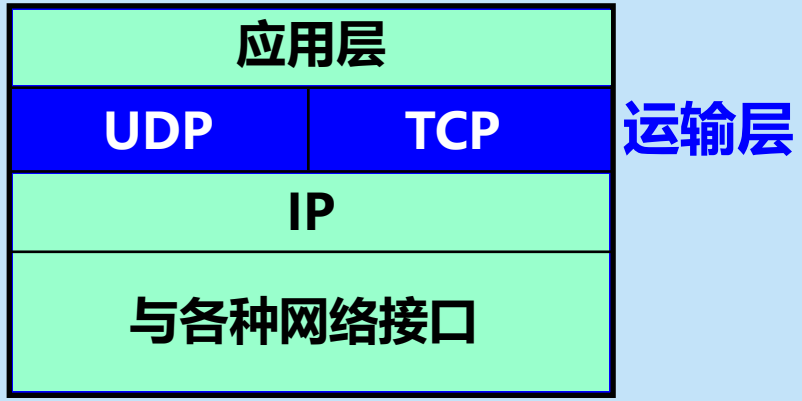
可靠信道与不可靠信道


发送进程
接收进程
发送进程
接收进程
?
应用层
数据
数据
数据
数据
运输层
全双工可靠信道
不可靠信道
使用面向连接的
协议，如 TCP。
使用无连接的
协议，如 UDP。

5.1.2　运输层的两个主要协议

TCP/IP 的运输层有两个主要协议：

1. 用户数据报协议 UDP (User Datagram Protocol)
2. 传输控制协议 TCP (Transmission Control Protocol)



TCP/IP 体系中的运输层协议

UDP 与 TCP

UDP

无连接的协议，提供无连接服务；

其传送的运输协议数据单元TPDU是UDP 报文或用户数据报；

支持单播、多播、广播；

不提供可靠交付；

简单。适用于很多应用，如：多媒体应用等。

TCP

面向连接的协议，提供面向连接服务；

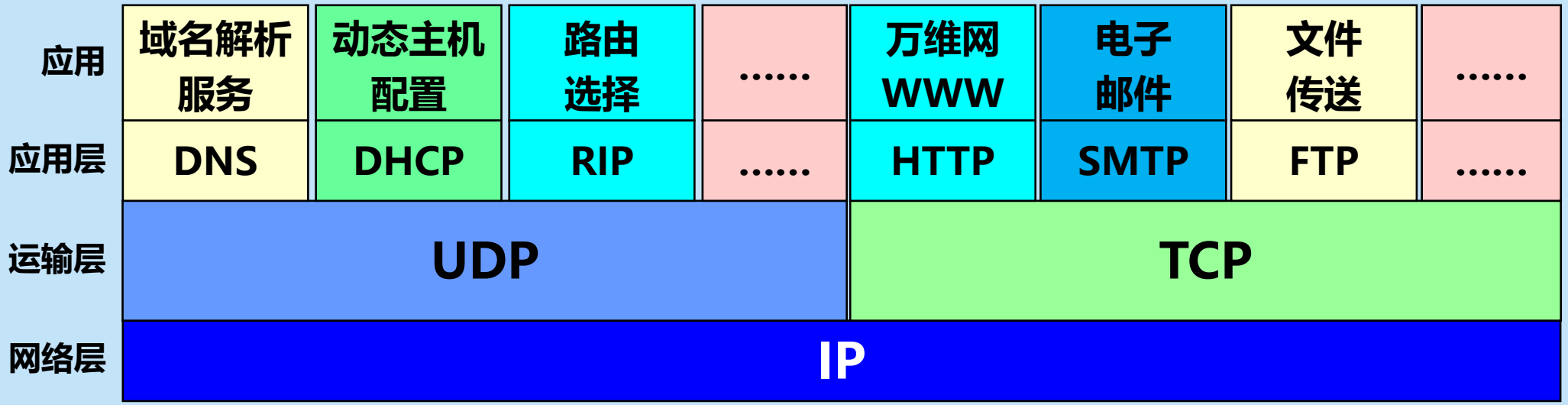
其传送的运输协议数据单元TPDU是TCP 报文；

支持点对点单播，不支持多播、广播；

提供可靠服务；

复杂。用于大多数应用，如：万维网、电子邮件、文件传送等。

使用 UDP 和 TCP 的典型应用和应用层协议



套接字 (socket)

套接字 socket = (IP地址 : 端口号)　　(5-1)

例如:

套接字 socket = (192.169.1.20 : 2028)

5.4.1 停止等待协议

- “停止等待”就是每发送完一个分组就停止发送，等待对方的确认。在收到确认后再发送下一个分组。
- 全双工通信的双方既是发送方也是接收方。
- 为了讨论问题的方便，我们仅考虑 A 发送数据，而 B 接收数据并发送确认。因此 A 叫做发送方，而 B 叫做接收方。

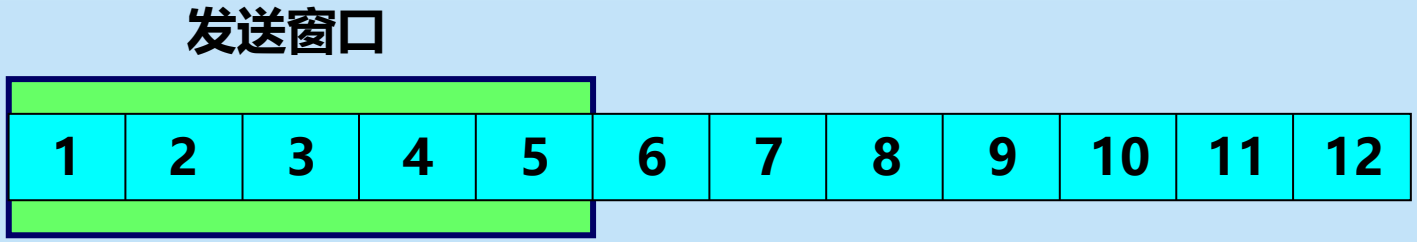
停止等待协议要点

- **停止等待**。发送方每次只发送一个分组。在收到确认后再发送下一个分组。
- **编号**。对发送的每个分组和确认都进行编号。
- **自动重传请求**。发送方为每个发送的分组设置一个超时计时器。若超时计时器超时，发送方会自动重传分组。
- 简单，但信道利用率太低。

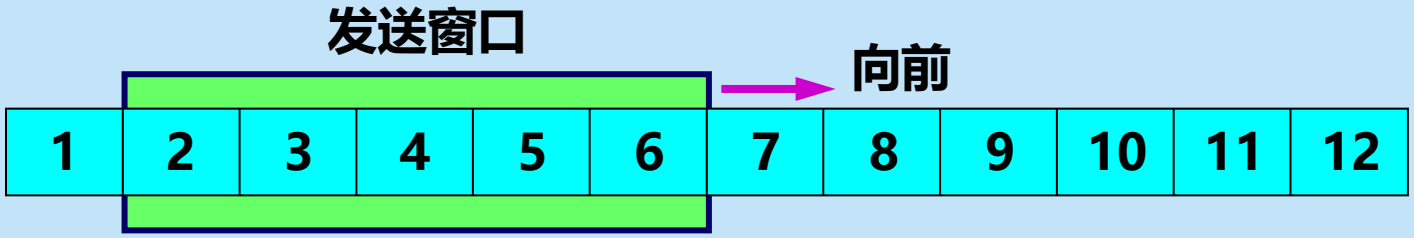
5.4.2　连续 ARQ 协议

- **基本思想：**
- **发送方一次可以发出多个分组。**
- **使用滑动窗口协议控制发送方和接收方所能发送和接收的分组的数量和编号。**
- **每收到一个确认，发送方就把发送窗口向前滑动。**
- **接收方一般采用累积确认的方式。**
- **采用回退N（Go-Back-N）方法进行重传。**

5.4.2　连续 ARQ 协议



(a) 发送方维持发送窗口（发送窗口是 5）



(b) 收到一个确认后发送窗口向前滑动

连续 ARQ 协议的工作原理

累积确认

- 接收方一般采用累积确认的方式。即不必对收到的分组逐个发送确认，而是对按序到达的最后一个分组发送确认，这样就表示：到这个分组为止的所有分组都已正确收到了。
- 优点：容易实现，即使确认丢失也不必重传。
- 缺点：不能向发送方反映出接收方已经正确收到的所有分组的信息。

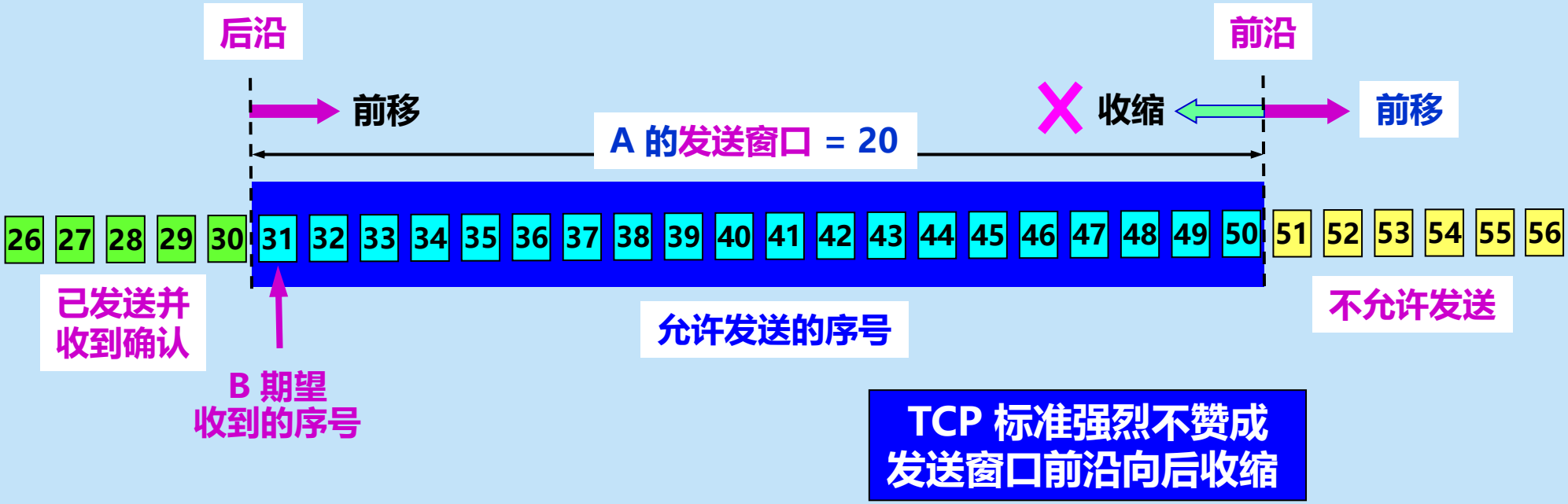
连续 ARQ 协议与停止等待协议

	连续ARQ协议	停止等待协议
发送的分组数量	一次发送多个分组	一次发送一个分组
传输控制	滑动窗口协议	停等-等待
确认	单独确认 + 累积确认	单独确认
超时定时器	每个发送的分组	每个发送的分组
编号	每个发送的分组	每个发送的分组
重传	回退N，多个分组	一个分组

5.6.1　以字节为单位的滑动窗口

- TCP 使用流水线传输和滑动窗口协议实现高效、可靠的传输。
- TCP 的滑动窗口是**以字节为单位**的。
- 发送方 A 和接收方 B 分别维持一个发送窗口和一个接收窗口。
- **发送窗口表示**：在没有收到确认的情况下，可以连续把窗口内的数据全部发送出去。
- **接收窗口表示**：只允许接收落入窗口内的数据。

- 根据 B 给出的窗口值，A 构造出自己的发送窗口。
- **发送窗口表示**：在没有收到 B 的确认的情况下，A 可以连续把窗口内的数据都发送出去。
- 发送窗口里面的序号表示允许发送的序号。
- 显然，窗口越大，发送方就可以在收到对方确认之前连续发送更多的数据，因而可能获得更高的传输效率。



TCP 标准强烈不赞成发送窗口前沿向后收缩

5.6.2　超时重传时间的选择

- 重传机制是 TCP 中最重要和最复杂的问题之一。
- TCP 每发送一个报文段，就对这个报文段设置一次计时器。
- 只要计时器设置的重传时间到但还没有收到确认，就要重传这一报文段。
- 重传时间的选择是 TCP 最复杂的问题之一。

加权平均往返时间

- TCP保留了RTT的一个**加权平均往返时间**RTT_S（这又称为**平滑的往返时间**）。
- 第一次测量到 RTT 样本时，RTT_S 值就取为所测量到的 RTT 样本值。以后每测量到一个新的 RTT 样本，就按下式重新计算一次 RTT_S：

$$\text{新的}RTT_S = (1-\alpha) \times (\text{旧的}RTT_S) + \alpha \times (\text{新的RTT样本}) \qquad (5\text{-}4)$$

- 式中，$0 \le \alpha < 1$。若 α 很接近于零，表示 RTT 值更新较慢。若选择 α 接近于 1，则表示 RTT 值更新较快。
- RFC 6298 推荐的 α 值为 1/8，即 0.125。

超时重传时间 RTO

- RTO (Retransmission Time-Out) 应略大于上面得出的加权平均往返时间 RTT_S。
- RFC 6298 建议使用下式计算 RTO：

$$RTO = RTT_S + 4 \times RTT_D \qquad (5\text{-}5)$$

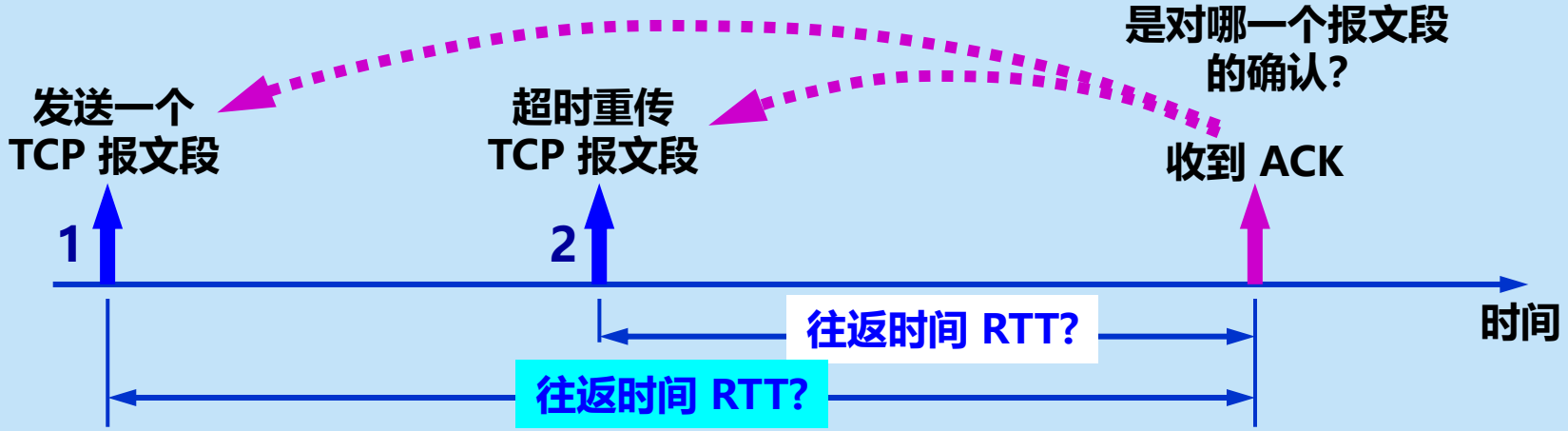
- RTT_D 是 RTT 的偏差的加权平均值。
- RFC 6298 建议这样计算RTT_D 。第一次测量时， RTT_D 值取为测量到的 RTT 样本值的一半。在以后的测量中，则使用下式计算加权平均的 RTT_D ：

$$\text{新的 } RTT_D = (1 - \beta) \times (\text{旧的} RTT_D) + \beta \times |RTT_S - \text{新的 RTT 样本}| \qquad (5\text{-}6)$$

- β是个小于 1 的系数，其推荐值是 1/4，即 0.25。

往返时间 (RTT) 的测量相当复杂

- TCP 报文段 1 没有收到确认。重传（即报文段 2）后，收到了确认报文段 ACK。
- **如何判定此确认报文段是对原来的报文段 1 的确认，还是对重传的报文段 2 的确认?**



5.7.1　利用滑动窗口实现流量控制

- 一般说来，我们总是希望数据传输得更快一些。但如果发送方把数据发送得过快，接收方就可能来不及接收，这就会造成数据的丢失。
- **流量控制** (flow control) 就是让发送方的发送速率不要太快，既要让接收方来得及接收，也不要使网络发生拥塞。
- 利用**滑动窗口机制**可以很方便地在 TCP 连接上实现流量控制。

利用可变窗口进行流量控制举例

A 向 B 发送数据。在连接建立时，B 告诉 A：
“我的接收窗口 rwnd = 400（字节）”。

A 发送了序号 1 至 100，还能发送 300 字节

A 发送了序号 101 至 200，还能发送 200 字节

允许 A 发送序号 201 至 500 共 300 字节

A 发送了序号 301 至 400，还能再发送 100 字节新数据

A 发送了序号 401 至 500，不能再发送新数据了

A 超时重传旧的数据，但不能发送新的数据

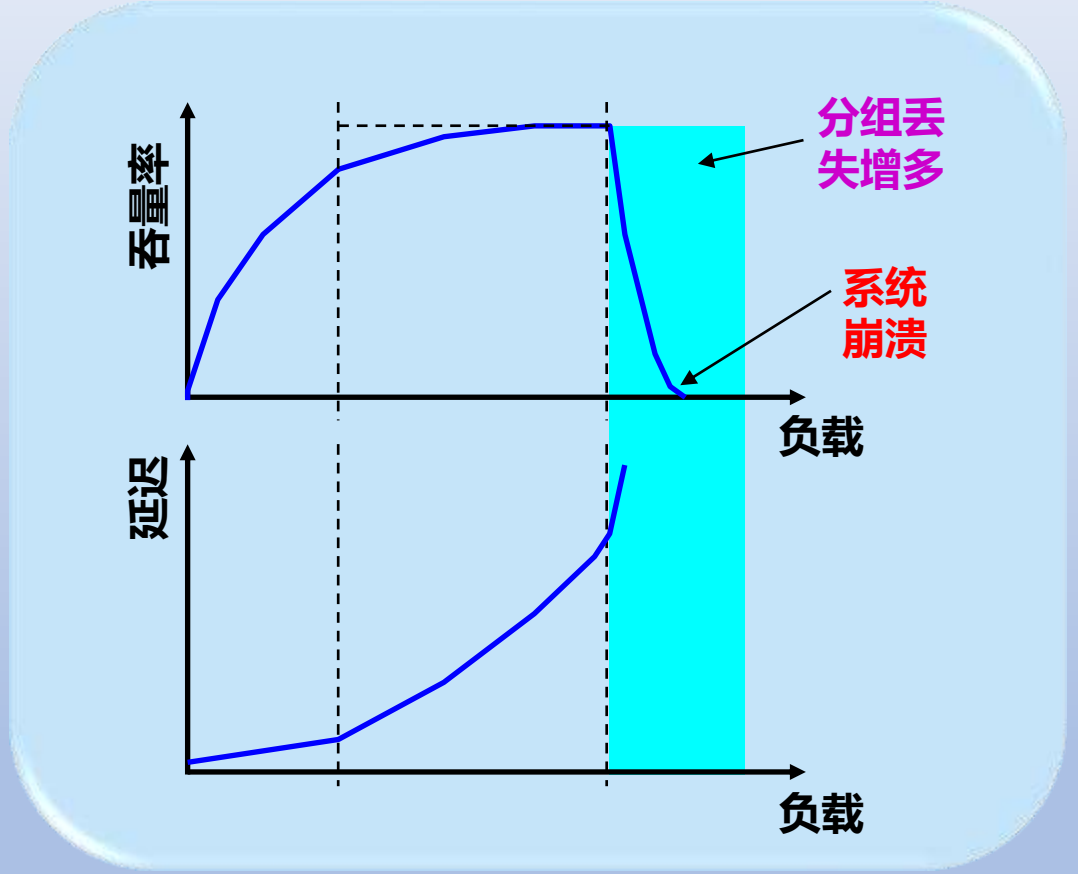
允许 A 发送序号 501 至 600 共 100 字节

A 发送了序号 501 至 600，不能再发送了

不允许 A 再发送（到序号 600 为止的数据都收到了）

5.8.1 拥塞控制的一般原理

- 在某段时间，若对网络中某资源的需求超过了该资源所能提供的可用部分，网络的性能就要变坏。这种现象称为**拥塞 (congestion)**。
- 最坏结果：**系统崩溃**。

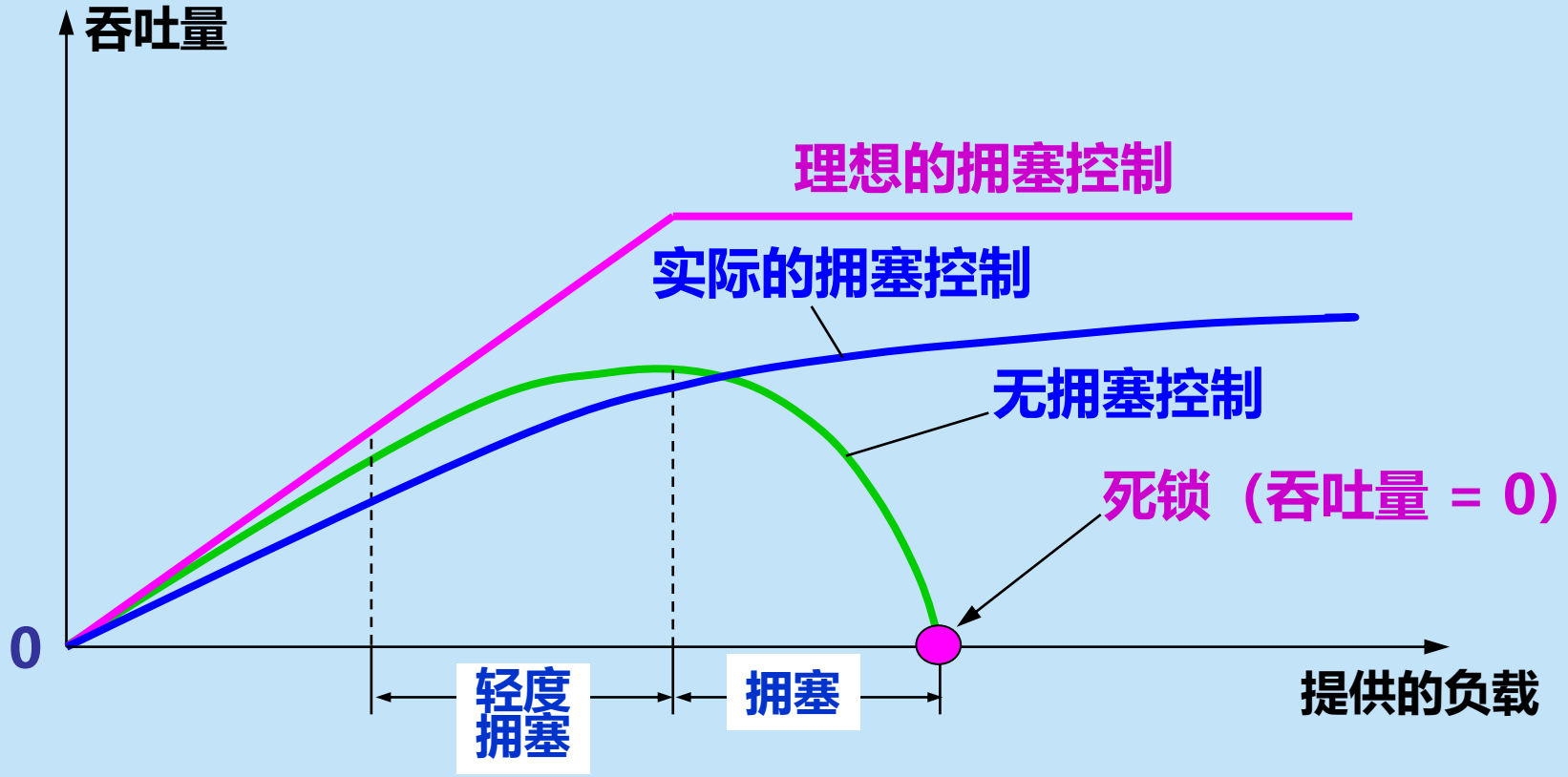


拥塞产生的原因

- 网络拥塞往往是由许多因素引起的。例如：
  1. 点缓存的容量太小；
  2. 链路的容量不足；
  3. 处理机处理的速率太慢；
  4. 拥塞本身会进一步加剧拥塞；
- 出现拥塞的原因：

$$\sum \text{对资源需求} > \text{可用资源} \qquad (5\text{-}7)$$

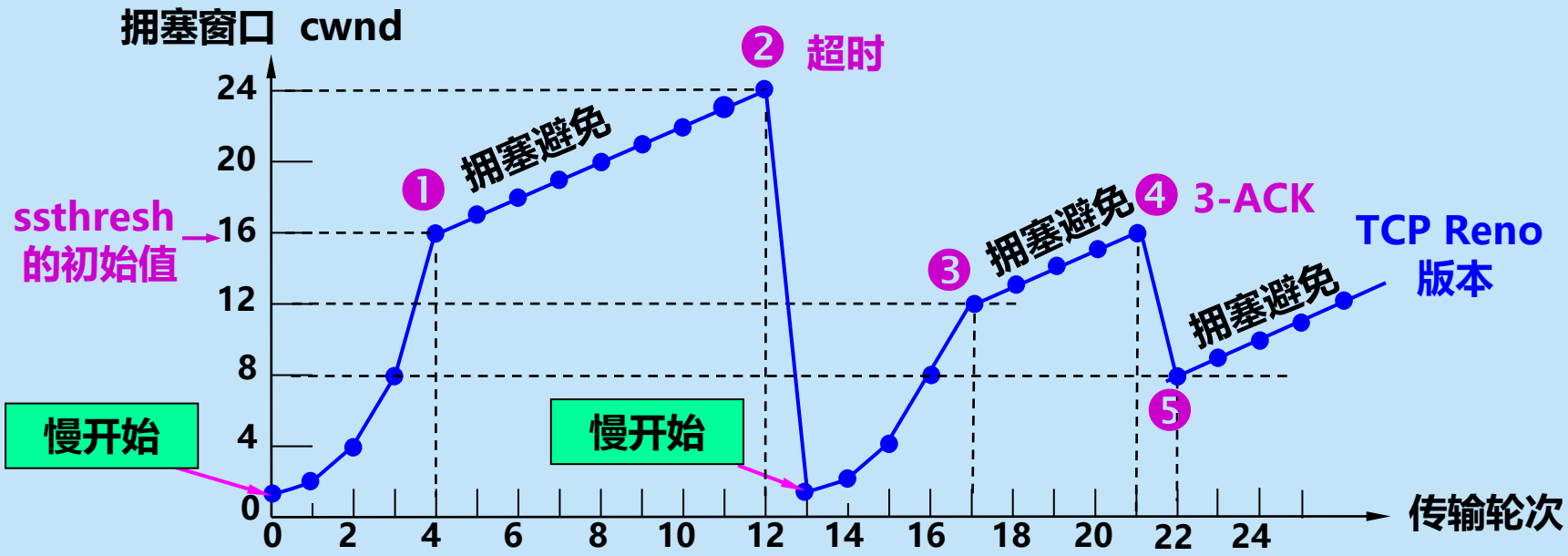
拥塞控制所起的作用



TCP拥塞控制算法

- 四种拥塞控制算法（RFC 5681）：
  - 慢开始 (slow-start)
  - 拥塞避免 (congestion avoidance)
  - 快重传 (fast retransmit)
  - 快恢复 (fast recovery)

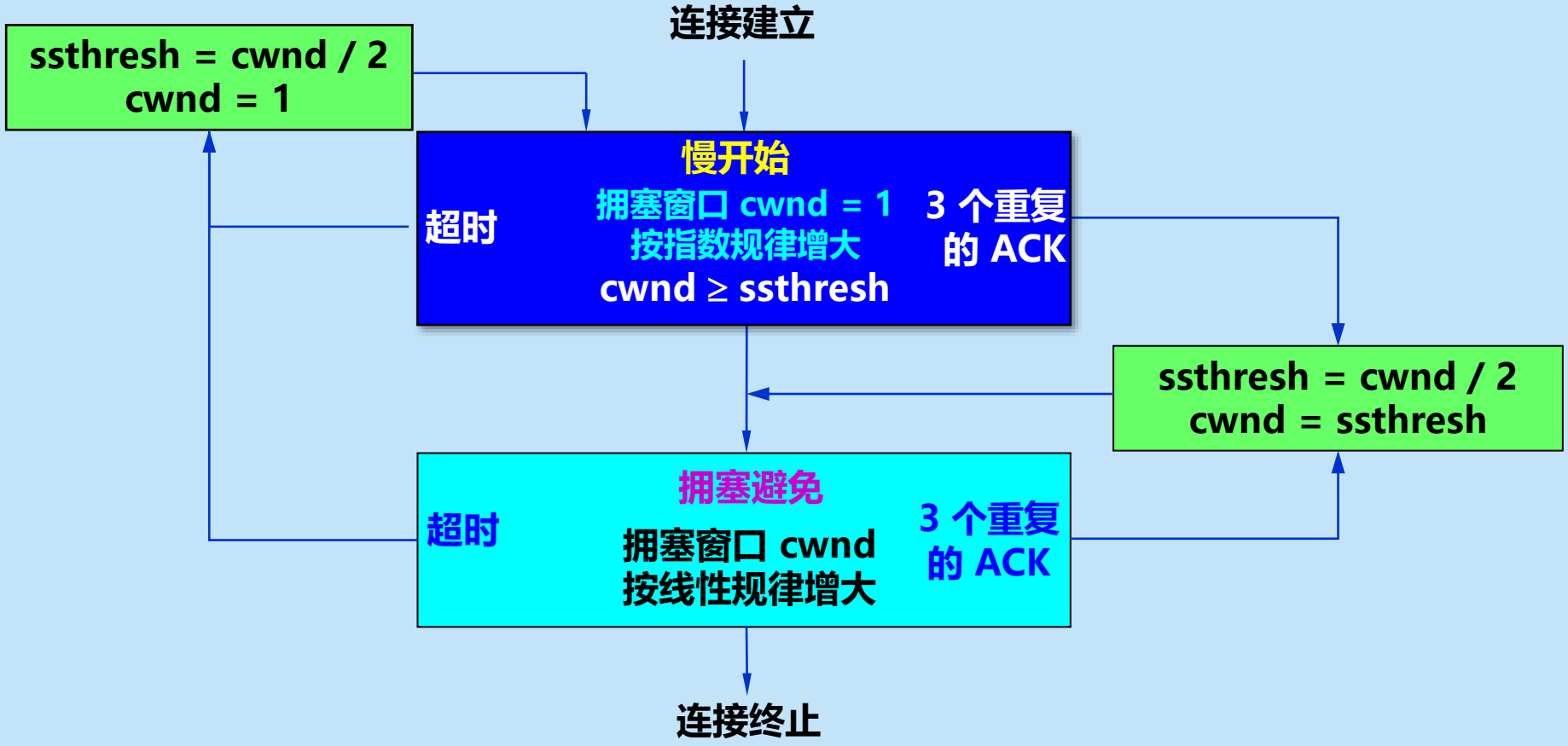
慢开始和拥塞避免算法的实现举例



当 TCP 连接进行初始化时，将拥塞窗口置为 1。图中的窗口单位不使用字节而使用报文段。

慢开始门限的初始值设置为 16 个报文段，即 ssthresh = 16。

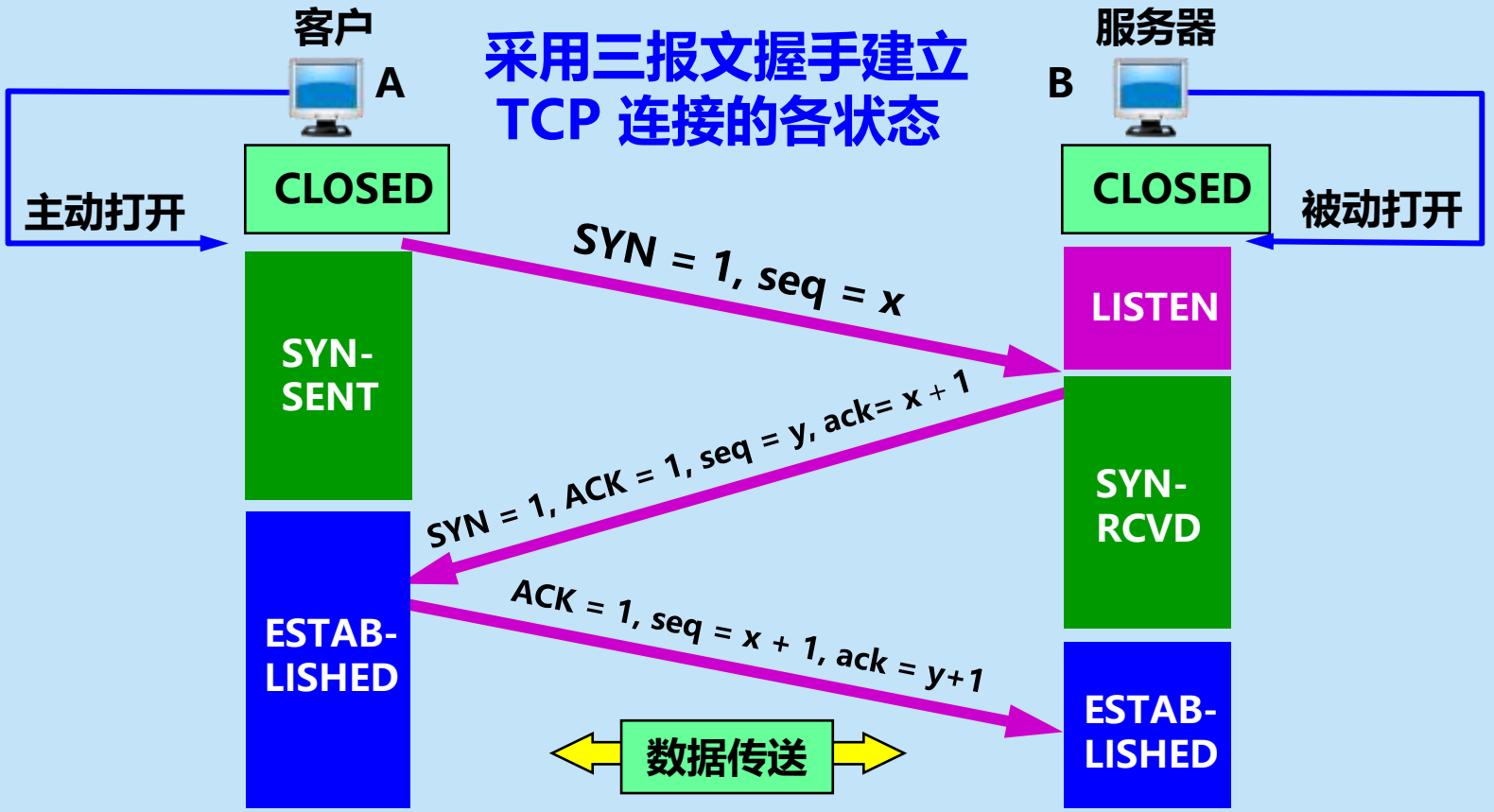
TCP拥塞控制流程图


连接建立
ssthresh = cwnd / 2
cwnd = 1
慢开始
拥塞窗口 cwnd = 1
按指数规律增大
cwnd ≥ ssthresh
超时
3 个重复
的 ACK
ssthresh = cwnd / 2
cwnd = ssthresh
拥塞避免
拥塞窗口 cwnd
按线性规律增大
超时
3 个重复
的 ACK
连接终止

5.9.1 TCP 的连接建立

- TCP 建立连接的过程叫做**握手**。
- 握手需要在客户和服务器之间交换三个 TCP 报文段。称之为**三报文握手**。
- 采用**三报文握手**主要是为了防止已失效的连接请求报文段突然又传送到了，因而产生错误。

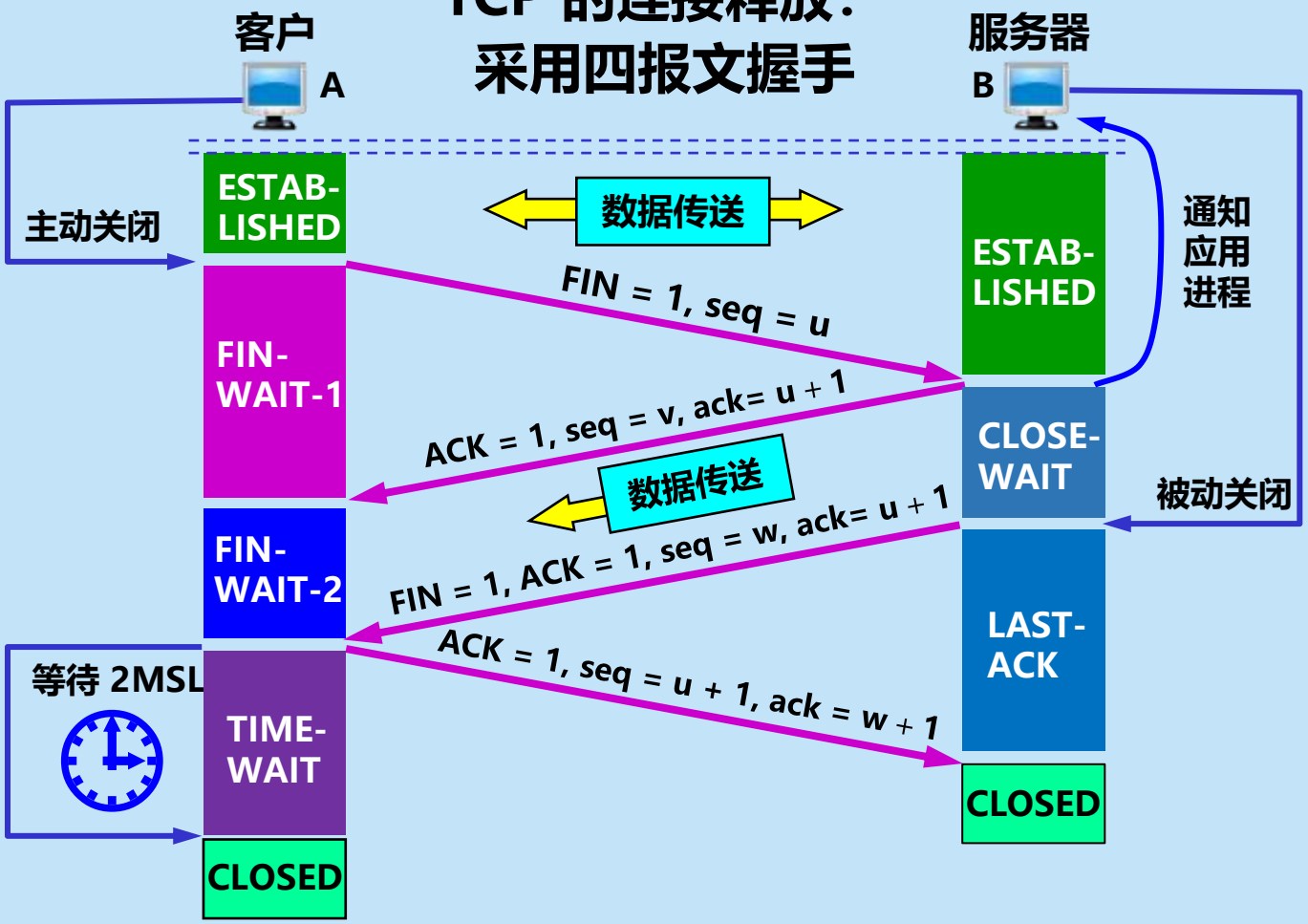
TCP 的连接建立：采用三报文握手



5.9.2 TCP 的连接释放

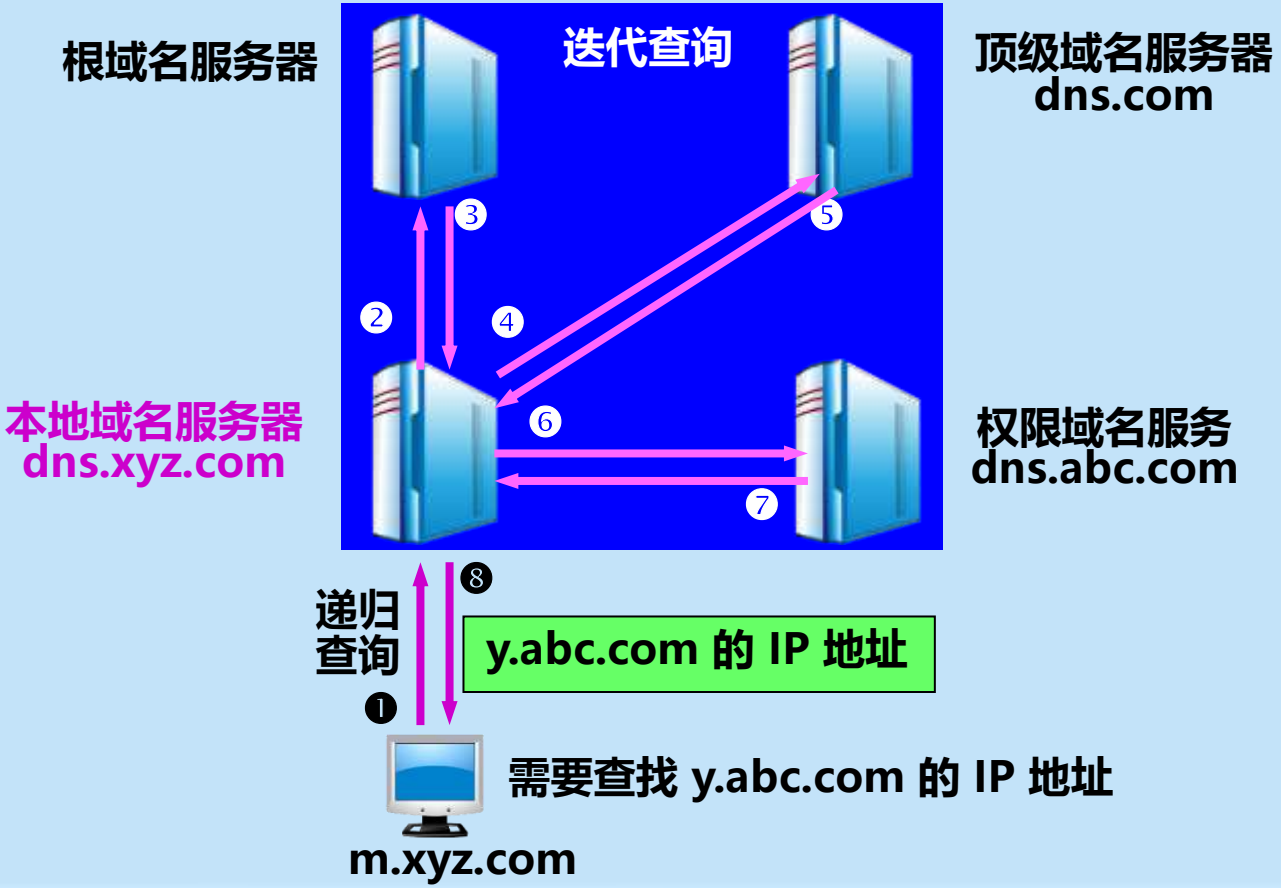
- TCP 连接释放过程比较复杂。
- 数据传输结束后，通信的双方都可释放连接。
- TCP 连接释放过程是**四报文握手**。

TCP 的连接释放：采用四报文握手



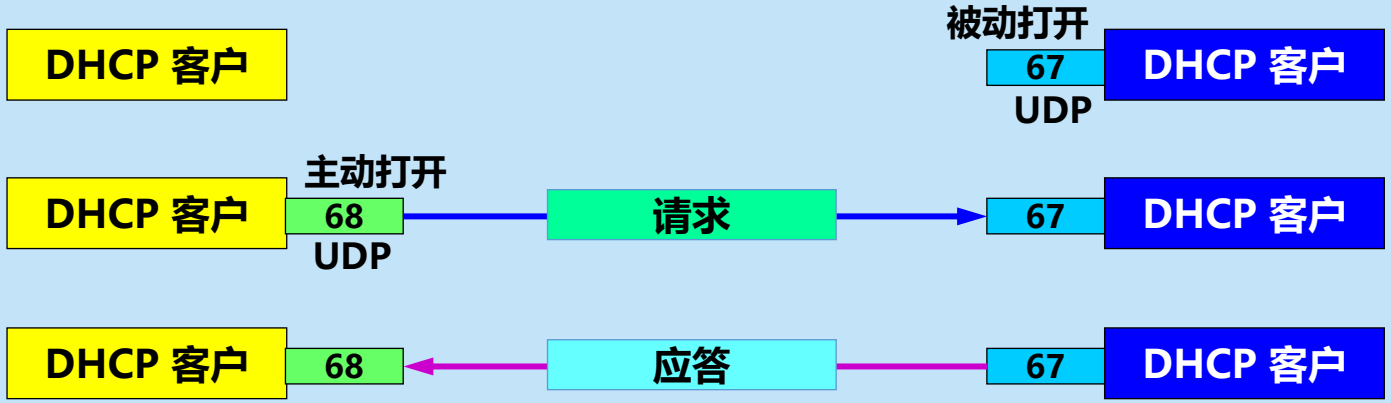
应用层

本地域名服务器采用迭代查询

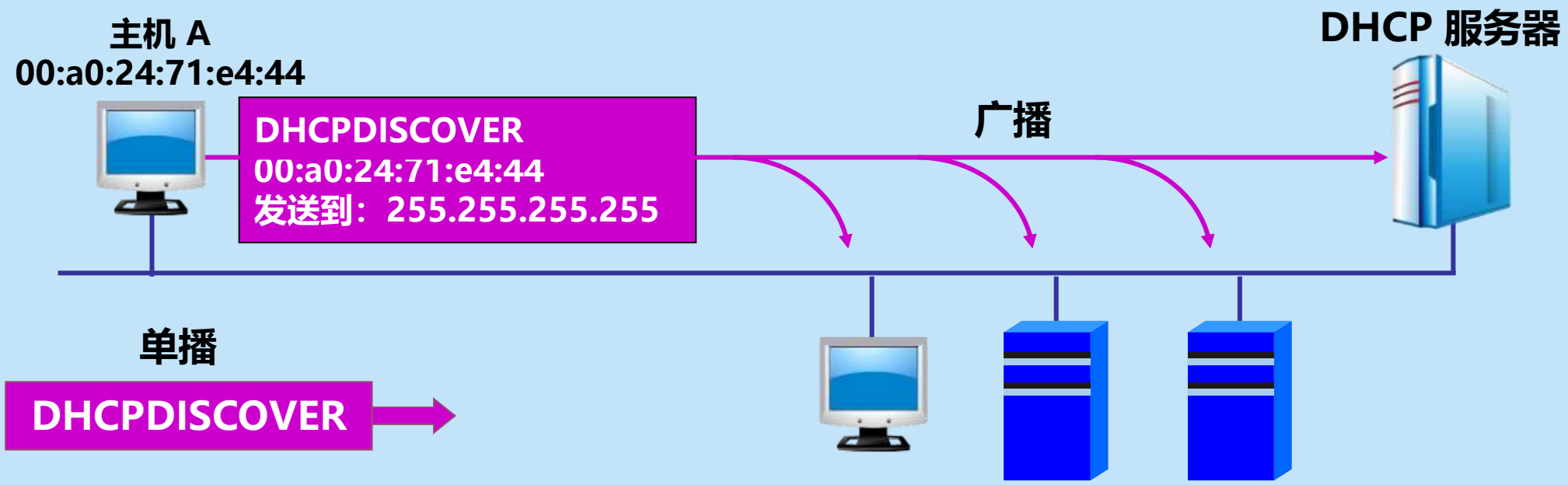


DHCP 工作方式

- DHCP 使用客户-服务器方式，采用请求/应答方式工作。
- DHCP 基于 UDP 工作，DHCP 服务器运行在 67 号端口， DHCP 客户运行在 68 号端口。

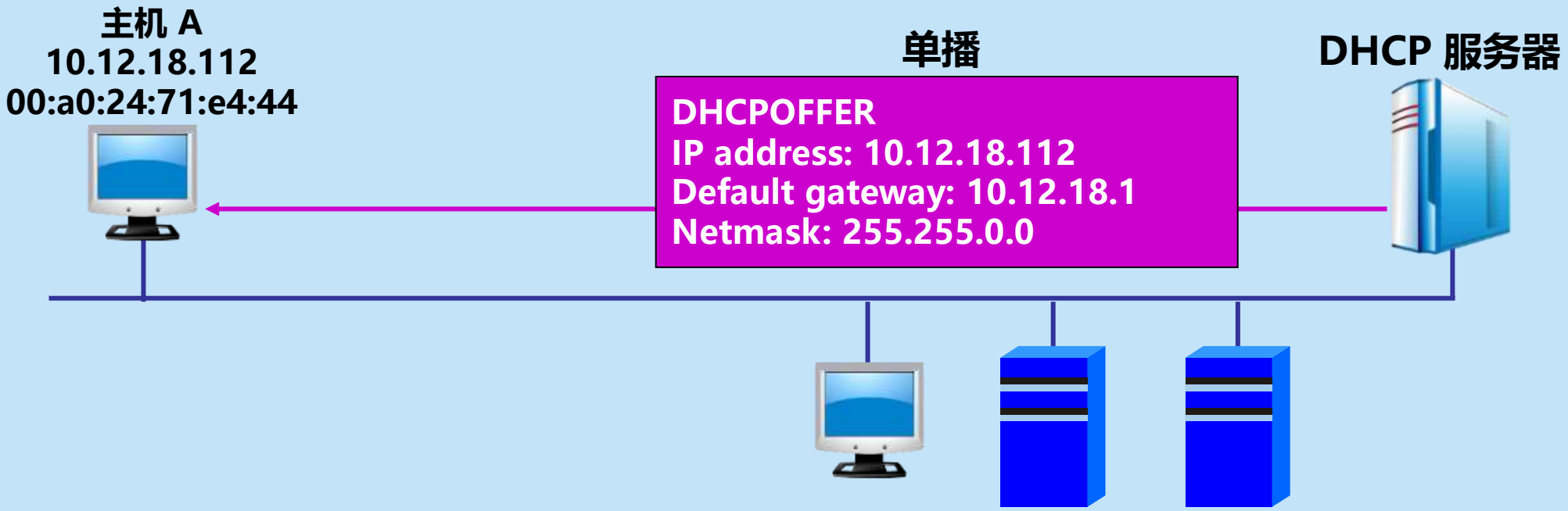


DHCP 工作方式



需要IP地址的主机向DHCP服务器广播发送发现报文 (DHCPDISCOVER) 。

DHCP 工作方式



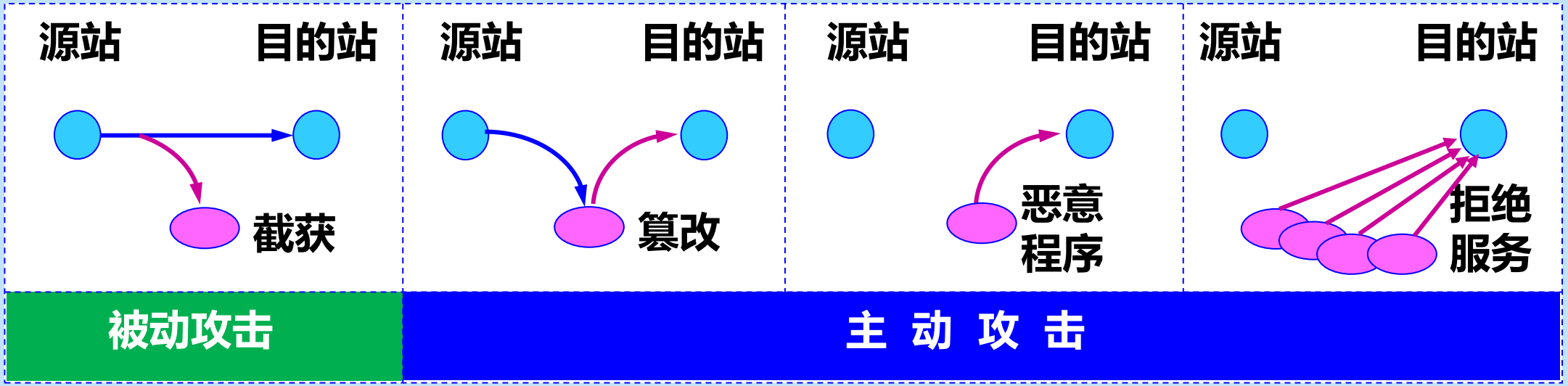
DHCP服务器回答提供报文 (DHCPOFFER)，表示“提供”IP地址等配置信息。

P2P 工作方式概述

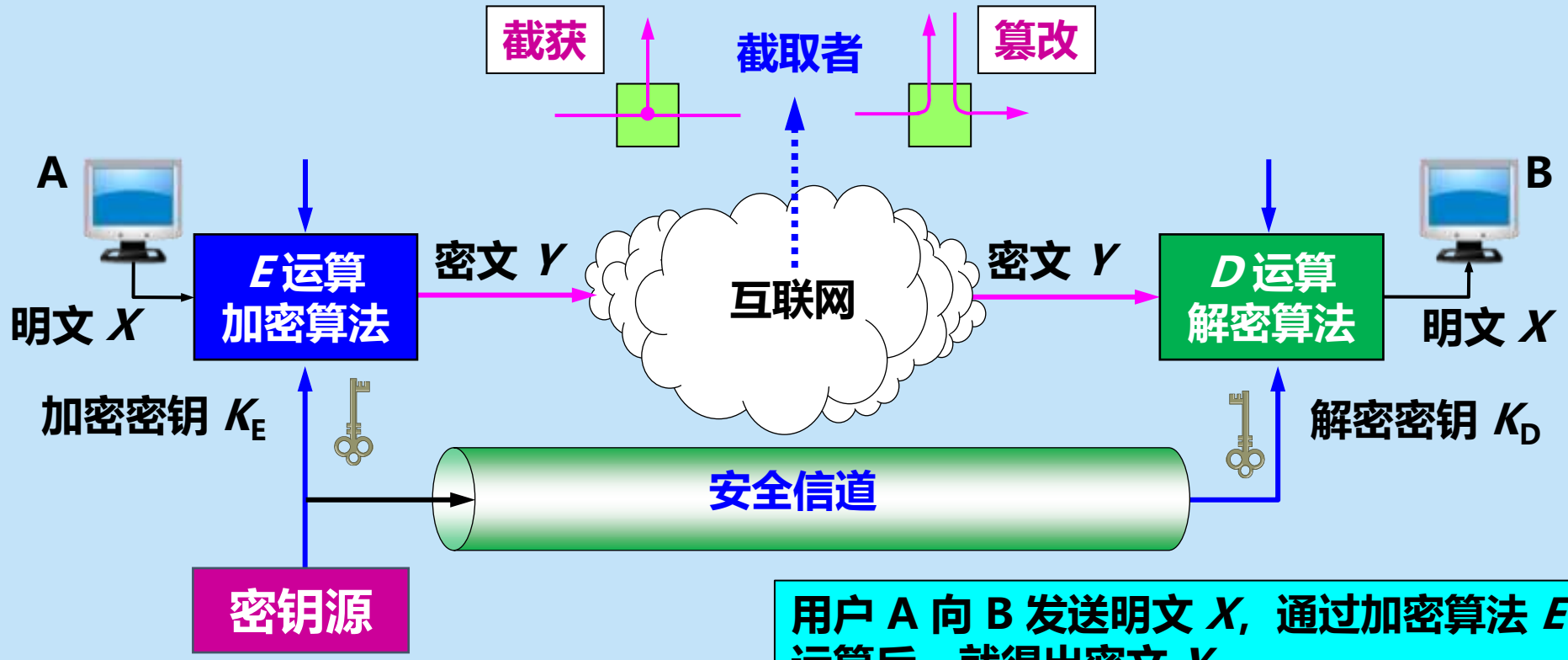
网络安全

7.1.1 计算机网络面临的安全性威胁

- 计算机网络上的通信面临以下两大类威胁：**被动攻击**和**主动攻击**。



数据加密模型



用户 A 向 B 发送明文 *X*，通过加密算法 *E* 运算后，就得出密文 *Y*。

密钥

- 加密和解密用的密钥 K (key) 是一串秘密的字符串（即比特串）。
- 明文通过加密算法 E 和加密密钥 K 变成密文：

$$Y = E_K(X) \qquad (7\text{-}1)$$

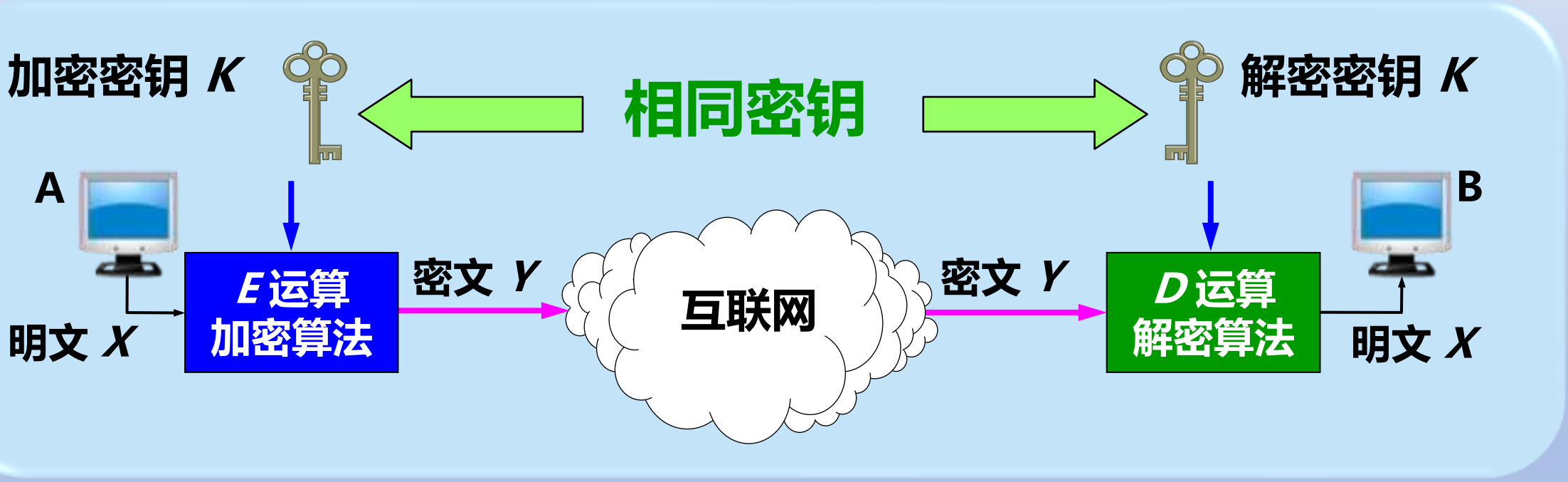
- 接收端利用解密算法 D 运算和解密密钥 K 解出明文 X。解密算法是加密算法的逆运算。

$$D_K(Y) = D_K(E_K(X)) = X \qquad (7\text{-}2)$$

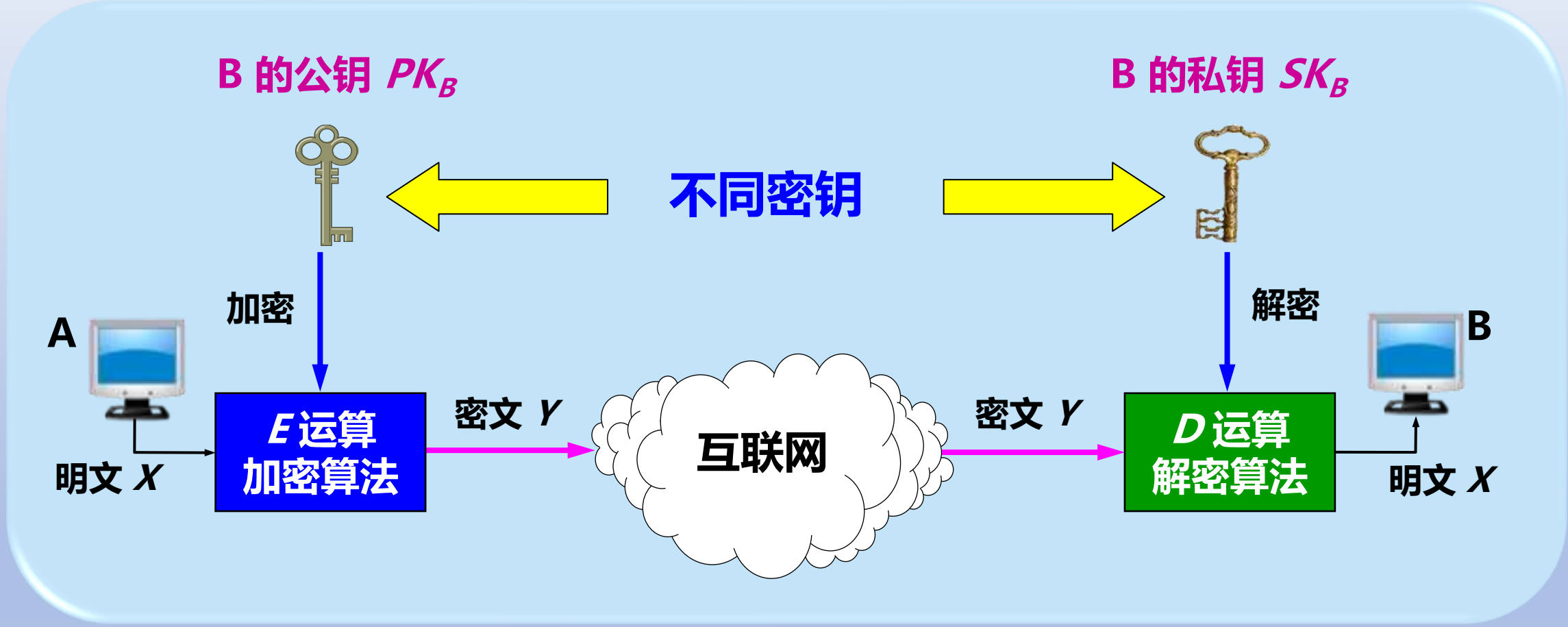
- 加密密钥和解密密钥可以一样，也可以不一样。
- 密钥通常由密钥中心提供。
- 当密钥需要向远地传送时，一定要通过另一个安全信道。

7.2.1　对称密钥密码体制

- 所谓常规密钥密码体制，即加密密钥与解密密钥是相同的密码体制。
- 这种加密系统又称为对称密钥系统。



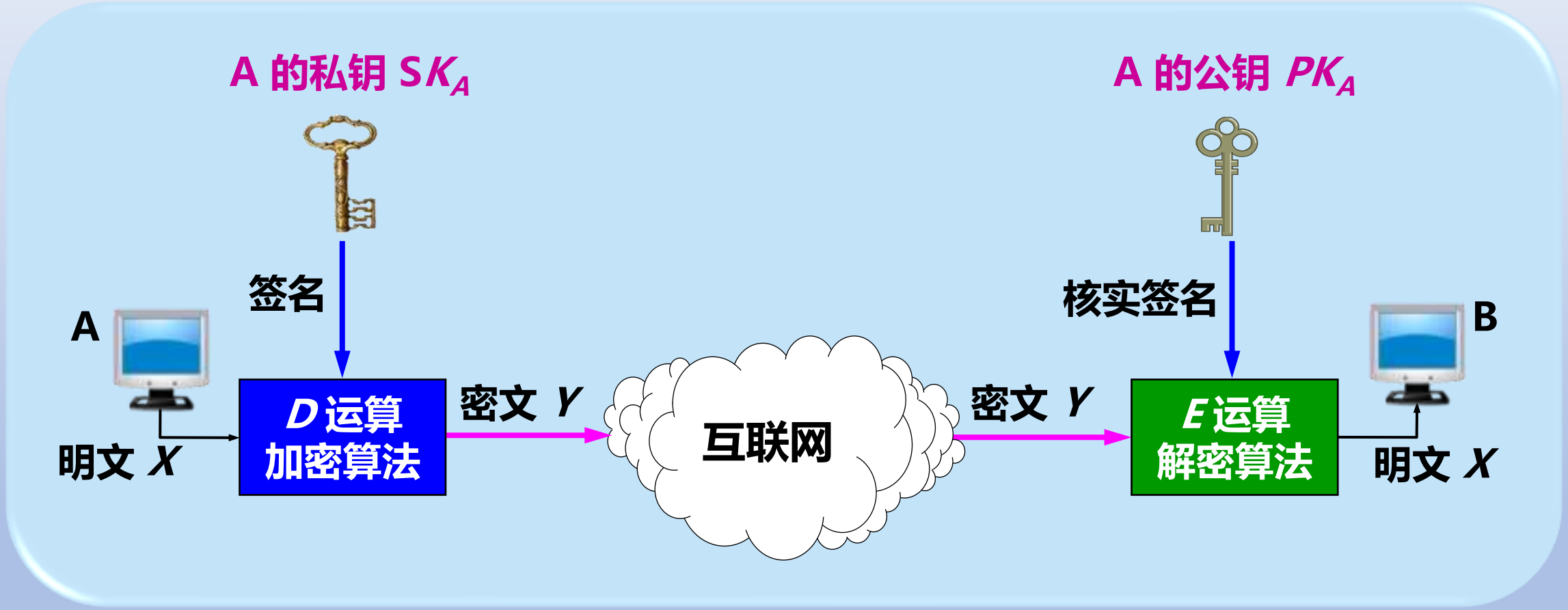
公钥密码体制


B 的公钥 PK_B
B 的私钥 SK_B
不同密钥
加密
解密
A
B
E 运算
加密算法
密文 Y
互联网
密文 Y
D 运算
解密算法
明文 X
明文 X

7.3 数字签名

- 用于证明真实性。
- 数字签名必须保证以下三点：
  1. 报文鉴别——接收者能够核实发送者对报文的签名（证明来源）；
  2. 报文的完整性——发送者事后不能抵赖对报文的签名（防否认）；
  3. 不可否认——接收者不能伪造对报文的签名（防伪造）。
- 现在已有多种实现各种数字签名的方法。但采用公钥算法更容易实现。

基于公钥的数字签名的实现



基于公钥的数字签名的实现

- 因为除 A 外没有别人能具有 A 的私钥，所以除 A 外没有别人能产生这个密文。因此 B 相信报文 X 是 A 签名发送的。
- 若 A 要抵赖曾发送报文给 B，B 可将明文和对应的密文出示给第三者。第三者很容易用 A 的公钥去证实 A 确实发送 X 给 B。
- 反之，若 B 将 X 伪造成 X'，则 B 不能在第三者前出示对应的密文。这样就证明了 B 伪造了报文。

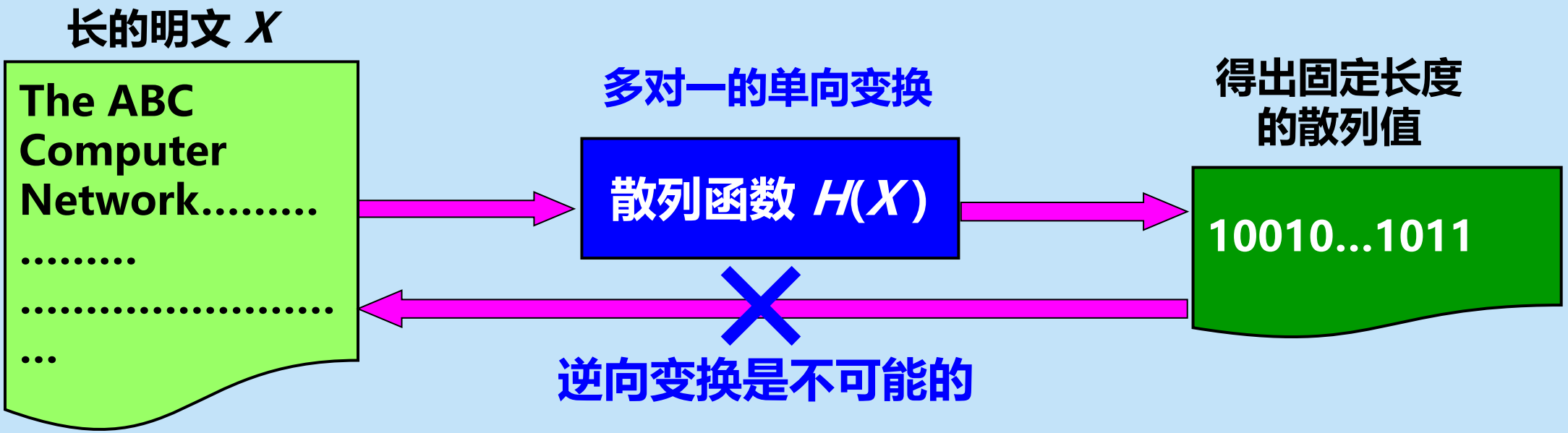
7.4.1 报文鉴别

- 许多报文并不需要加密，但却需要数字签名，以便让报文的接收者能够**鉴别报文的真伪**。
- 然而**对很长的报文进行数字签名**会使计算机增加很大的负担（需要进行很长时间的运算）。
- 当我们传送不需要加密的报文时，**应当使接收者能用很简单的方法鉴别报文的真伪。**

1. 密码散列函数

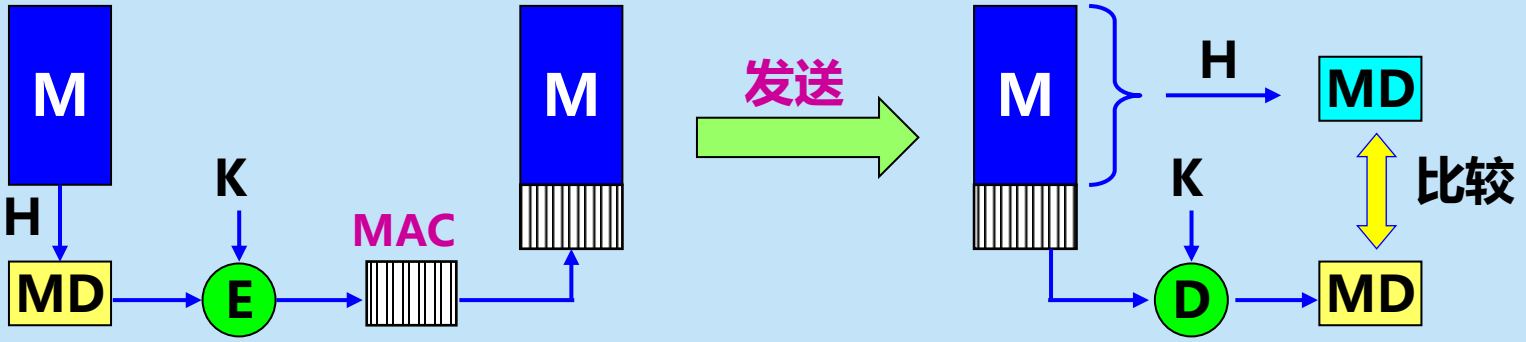
- 数字签名就能够实现对报文的鉴别。
- 但这种方法有一个很大的**缺点**：对较长的报文（这是很常见的）进行数字签名会使计算机增加非常大的负担，因为这需要较多的时间来进行运算。
- **密码散列函数** (cryptographic hash function)是一种相对简单的对报文进行鉴别的方法。

密码散列函数的特点

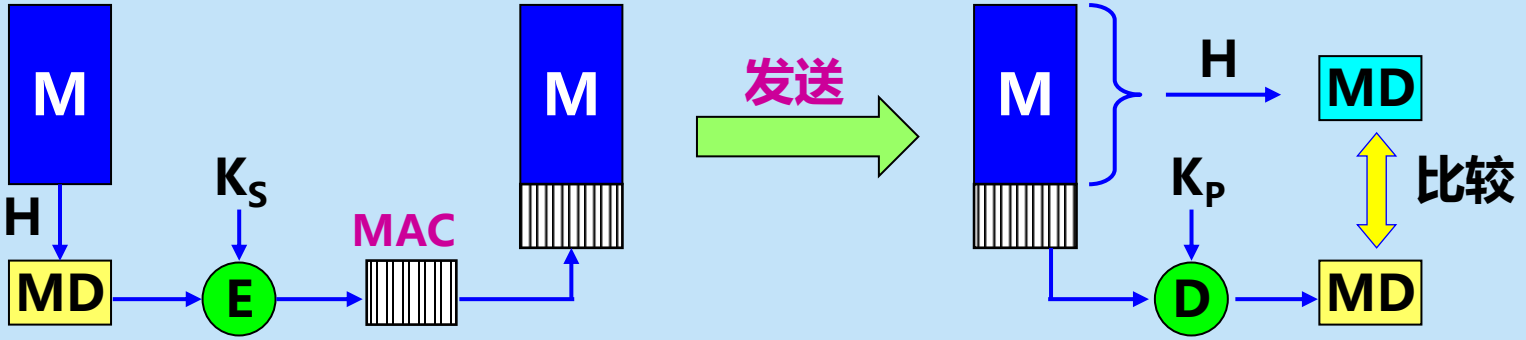


散列 *H(X)* 可用来保护明文 *X* 的完整性，防篡改和伪造。

3. 报文鉴别码 MAC



使用传统加密方法鉴别报文（防伪造）



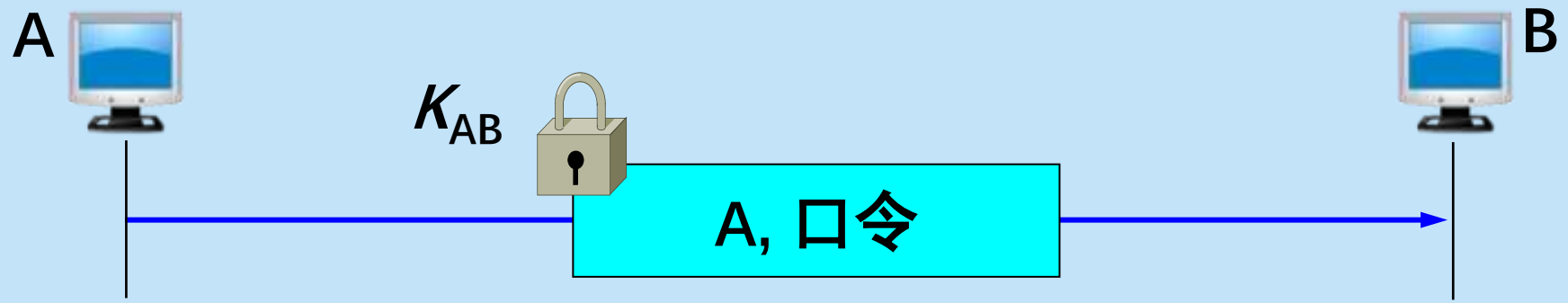
使用公开密钥密码体制鉴别报文（防伪造，防否认）

7.4.2 实体鉴别

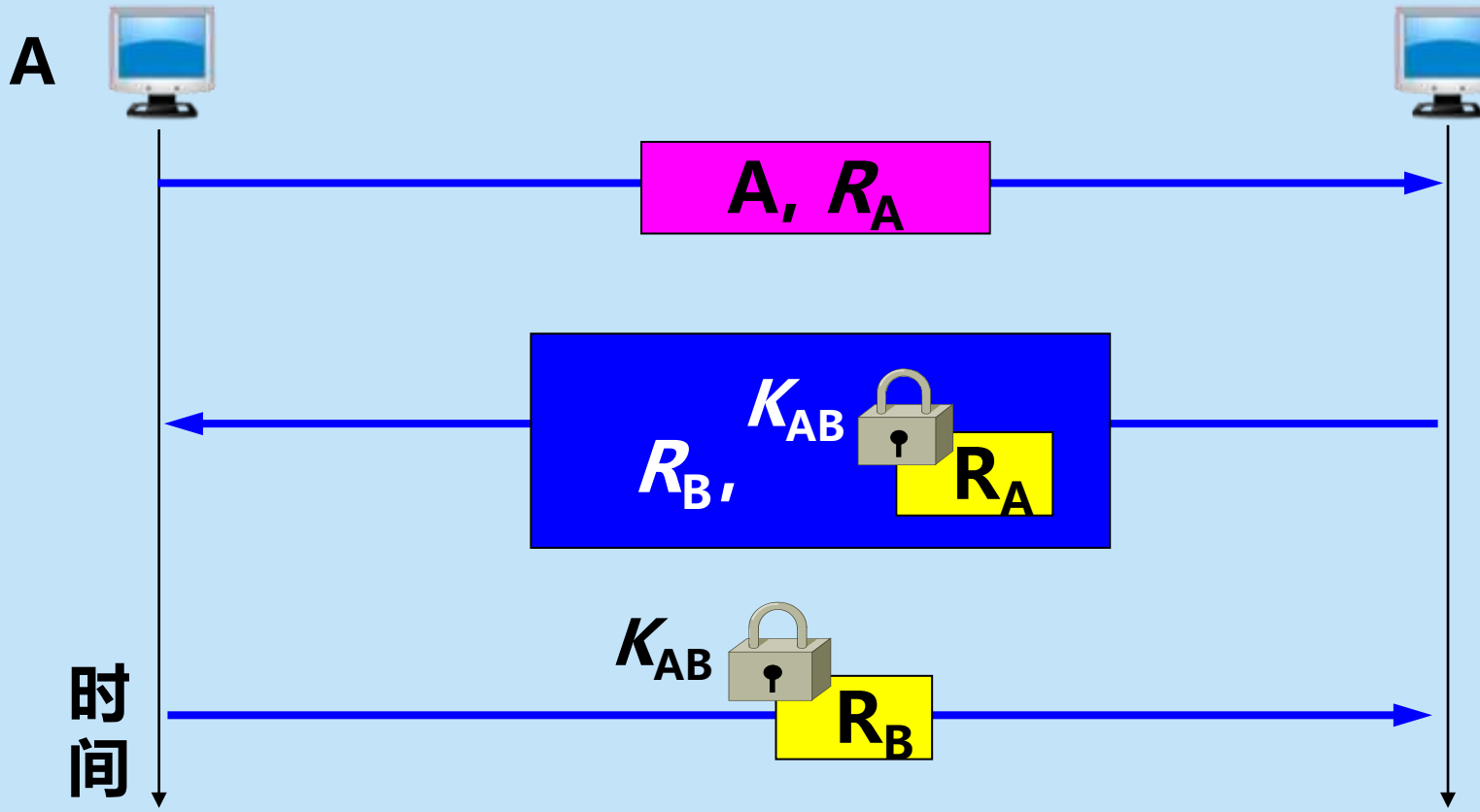
- 实体鉴别与报文鉴别不同。
- 报文鉴别是对每一个收到的报文都要鉴别报文的发送者。
- 实体鉴别是在系统接入的全部持续时间内对和自己通信的对方实体只需验证一次。

最简单的实体鉴别过程

- 可以使用共享的对称密钥实现实体鉴别。
- A 发送给 B 的报文的被加密，使用的是对称密钥 K_{AB}。
- B 收到此报文后，用共享对称密钥 K_{AB} 进行解密，因而鉴别了实体 A 的身份。 因为该密钥只有 A 和 B 知道。

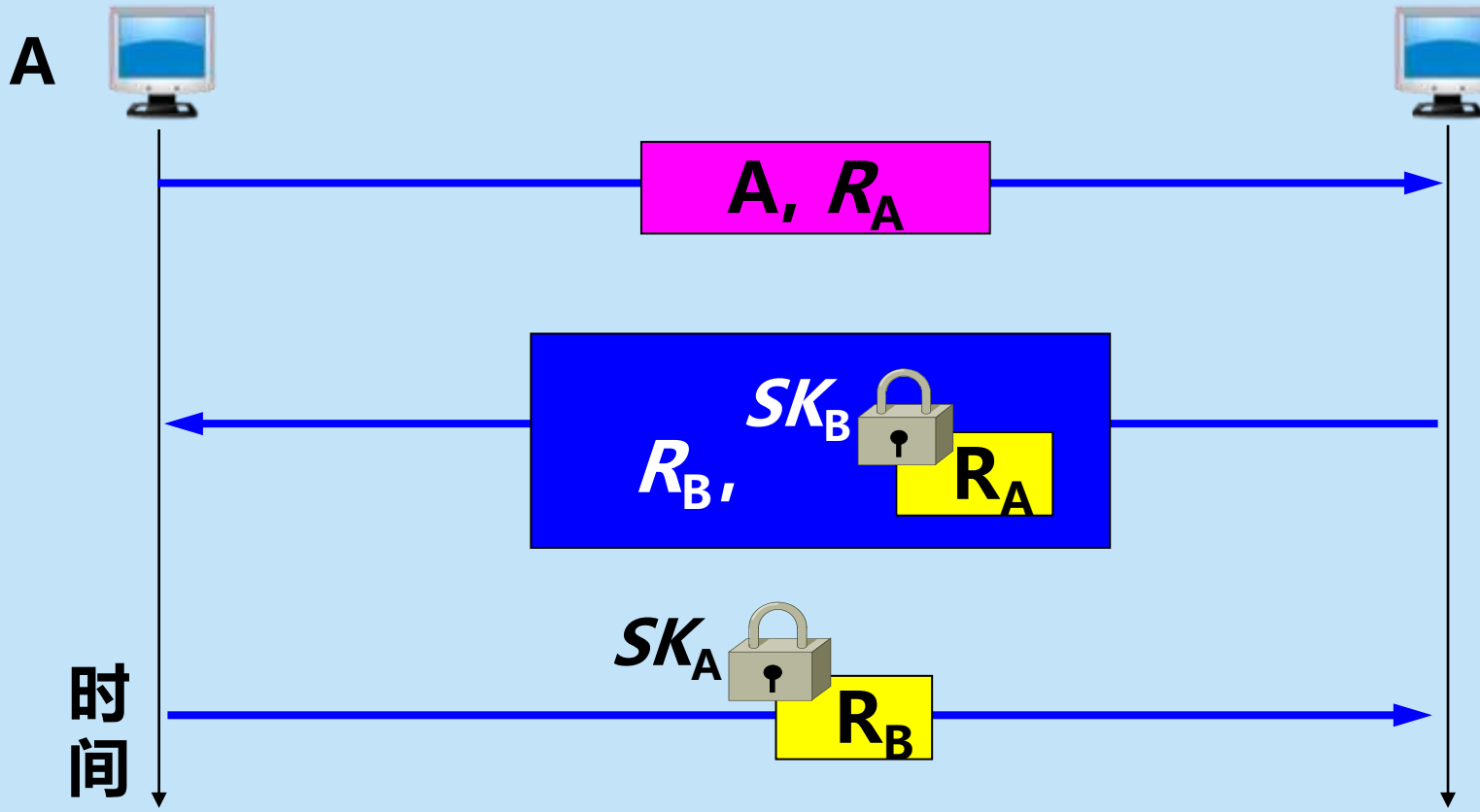


使用不重数进行鉴别



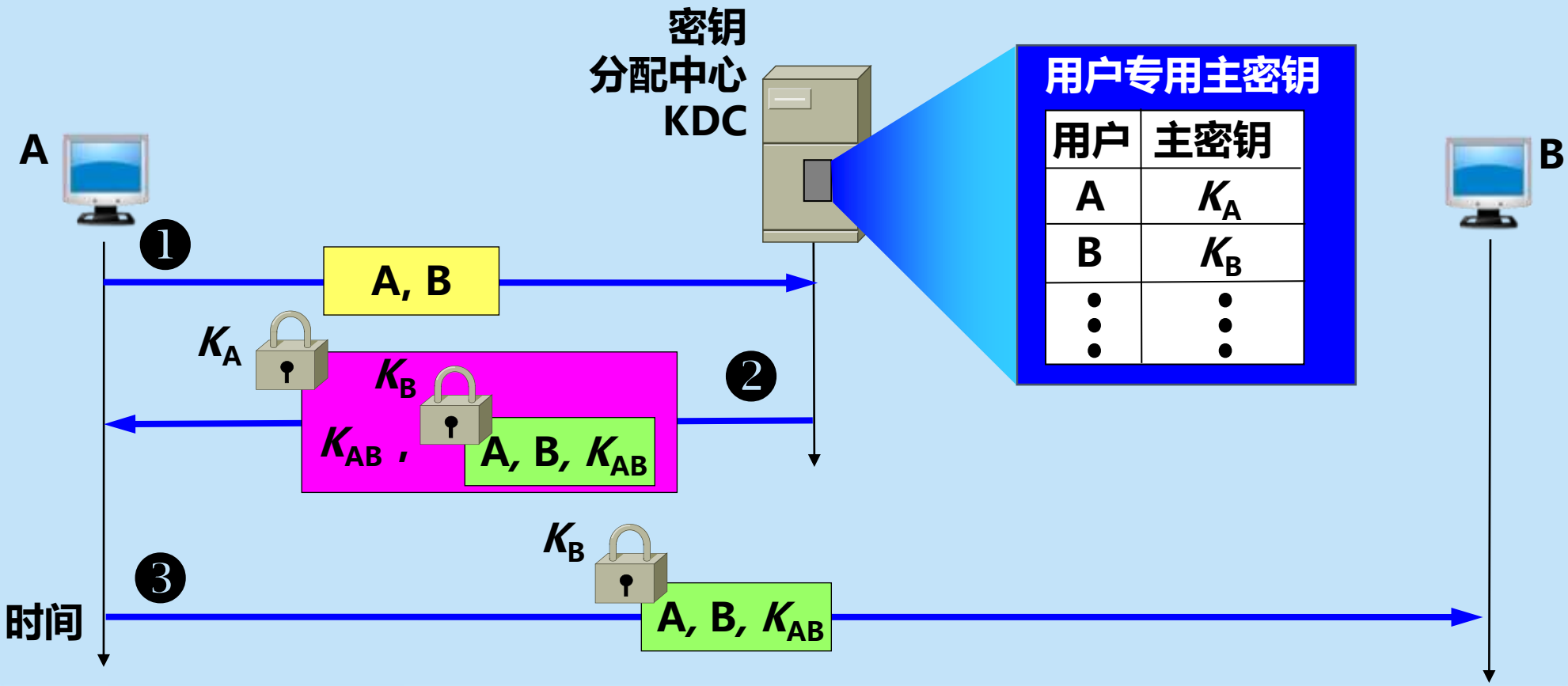
不重数 (nonce) 就是一个不重复使用的大随机数，即“一次一数”。由于不重数不能重复使用，所以 C 在进行重放攻击时无法重复使用所截获的不重数。

使用公钥体制进行不重数鉴别



不重数 (nonce) 就是一个不重复使用的大随机数，即“一次一数”。由于不重数不能重复使用，所以 C 在进行重放攻击时无法重复使用所截获的不重数。

对称密钥的分配


密钥
分配中心
KDC
用户专用主密钥
用户
主密钥
A
K_A
B
K_B
A
B
❶
A, B
K_A
K_B
K_{AB} ,
A, B, K_{AB}
❷
K_B
❸
A, B, K_{AB}
时间

7.5.2 公钥的分配

- 需要有一个值得信赖的机构——即认证中心 CA (Certification Authority)，来将公钥与其对应的实体（人或机器）进行绑定(binding)。
- 认证中心一般由政府出资建立。
- 每个实体都有 CA 发来的证书 (certificate)，里面有公钥及其拥有者的标识信息。
- 此证书被 CA 进行了数字签名，是不可伪造的，可以信任。
- 证书是一种身份证明，用于解决信任问题。

7.6 互联网使用的安全协议

1．IPsec 协议

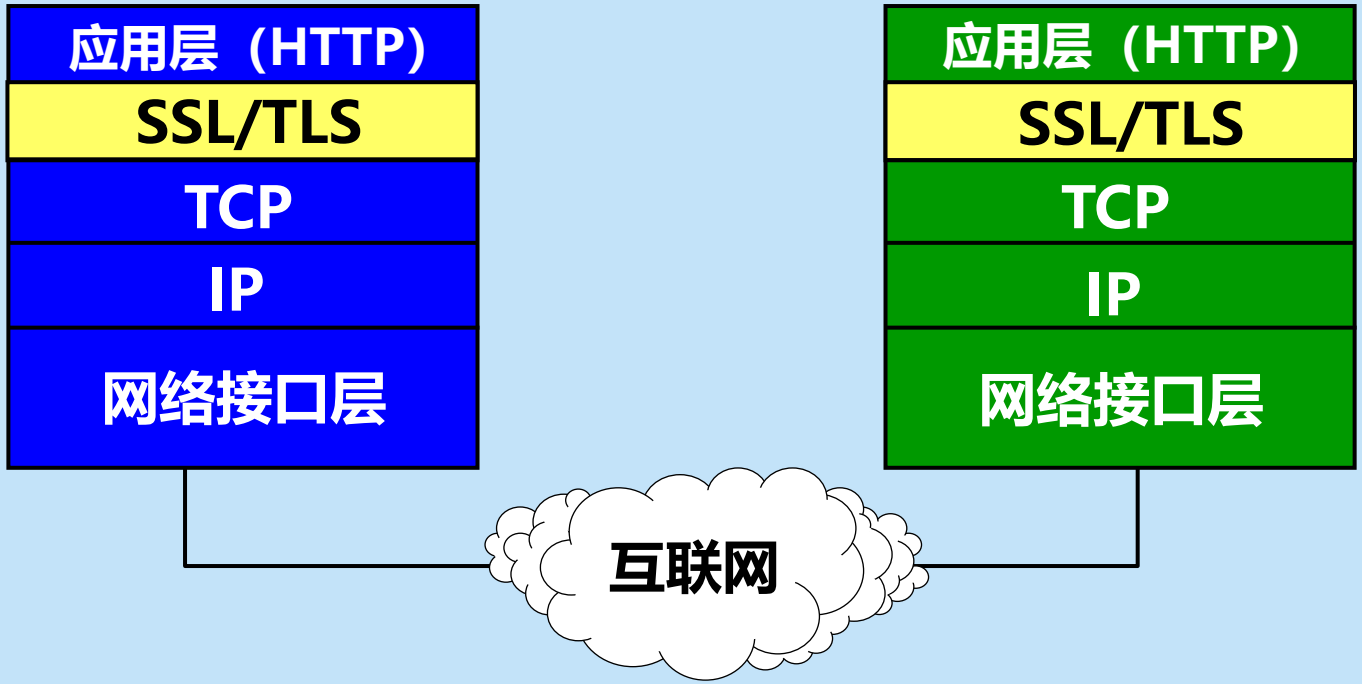
- IPsec 就是“IP 安全 (security)”的缩写。
- IPsec 并不是一个单个的协议，而是能够在 IP 层提供互联网通信安全的协议族。
- IPsec 是个框架，它允许通信双方选择合适的算法和参数（例如，密钥长度）。
- 为保证互操作性，IPsec 还包含了所有 IPsec 的实现都必须有的一套加密算法。

7.6.2 运输层安全协议

现在广泛使用的有以下两个协议：

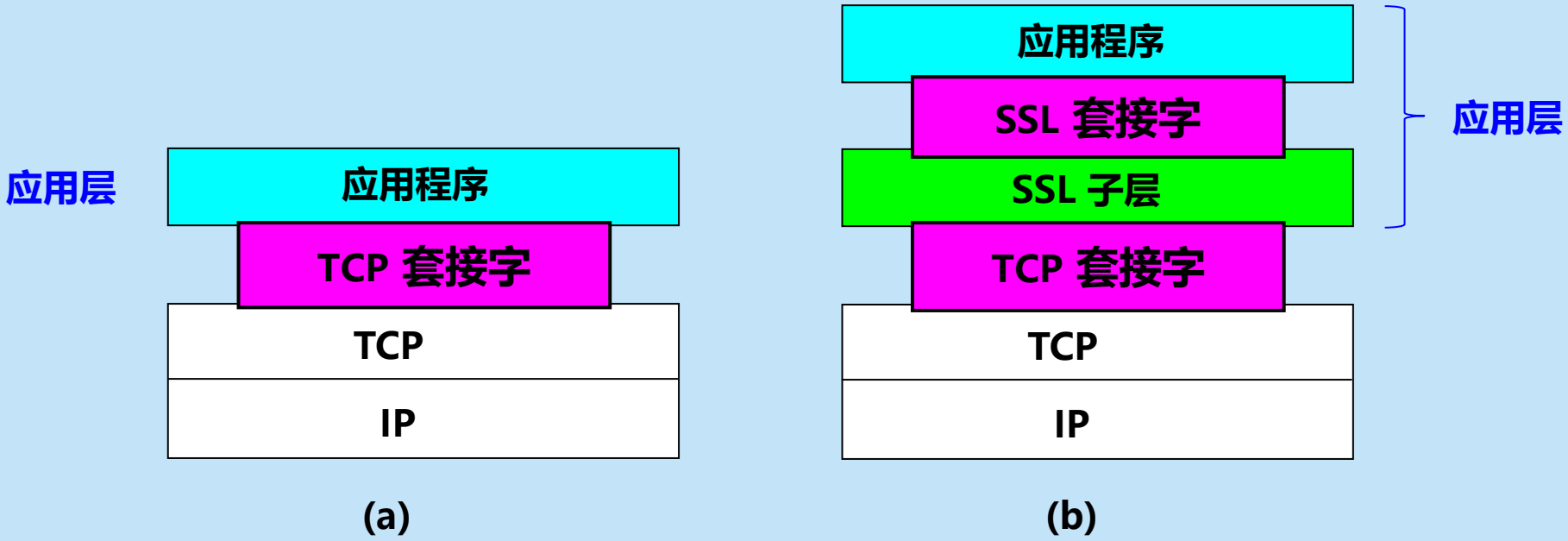
- 安全套接字层 SSL (Secure Socket Layer)
- 运输层安全 TLS (Transport Layer Security)

SSL / TLS 的位置



在发送方，SSL 接收应用层的数据，对数据进行加密，然后把加了密的数据送往 TCP 套接字。在接收方，SSL 从 TCP 套接字读取数据，解密后把数据交给应用层。

运输层不使用安全协议和使用安全协议的对比



什么是IOT

- 物联网（The Internet of Things，简称IOT）
- 物联网是一个基于互联网、它让所有能够被独立寻址的普通物理对象形成互联互通的网络 。

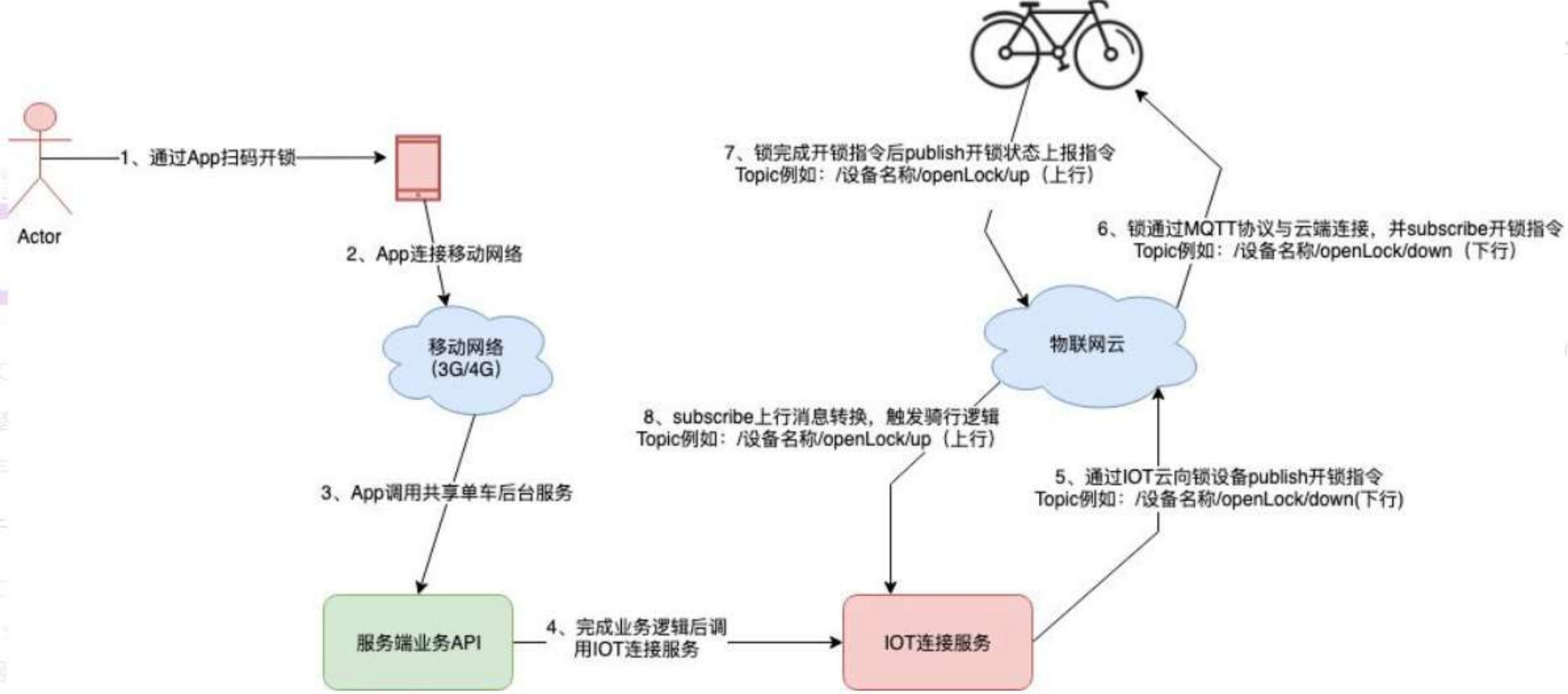
物联网（IoT，Internet of things）即“万物相连的互联网”，是互联网基础上的延伸和扩展的网络，将各种信息传感设备与互联网结合起来而形成的一个巨大网络，实现在任何时间、任何地点，人、机、物的互联互通

IOT应用层协议

- MQTT协议
  - MQTT（Message Queuing Telemetry Transport，消息队列遥测传输协议）
  - 是一种基于发布/订阅（publish/subscribe）模式的轻量级协议
  - 该协议构建于TCP/IP协议之上，MQTT最大优点在于，可以以极少的代码和有限的带宽，为连接远程设备提供实时可靠的消息服务。作为一种低开销、低带宽占用的即时通讯协议，使其在物联网、小型设备、移动应用等方面有较广泛的应用。
  - MQTT是一个基于客户端-服务器的消息发布/订阅传输协议。

MQTT实现方式

- 实现MQTT协议需要：客户端和服务器端。
- MQTT协议中有三种身份：发布者（Publish）、代理（Broker）（服务器）、订阅者（Subscribe）。其中，消息的发布者和订阅者都是客户端，消息代理是服务器，消息发布者可以同时是订阅者。


Actor
1、通过App扫码开锁
2、App连接移动网络
移动网络
(3G/4G)
3、App调用共享单车后台服务
服务端业务API
4、完成业务逻辑后调
用IOT连接服务
IOT连接服务
5、通过IOT云向锁设备publish开锁指令
Topic例如：/设备名称/openLock/down(下行)
6、锁通过MQTT协议与云端连接，并subscribe开锁指令
Topic例如：/设备名称/openLock/down（下行）
物联网云
7、锁完成开锁指令后publish开锁状态上报指令
Topic例如：/设备名称/openLock/up（上行）
8、subscribe上行消息转换，触发骑行逻辑
Topic例如：/设备名称/openLock/up（上行）

共享单车例子