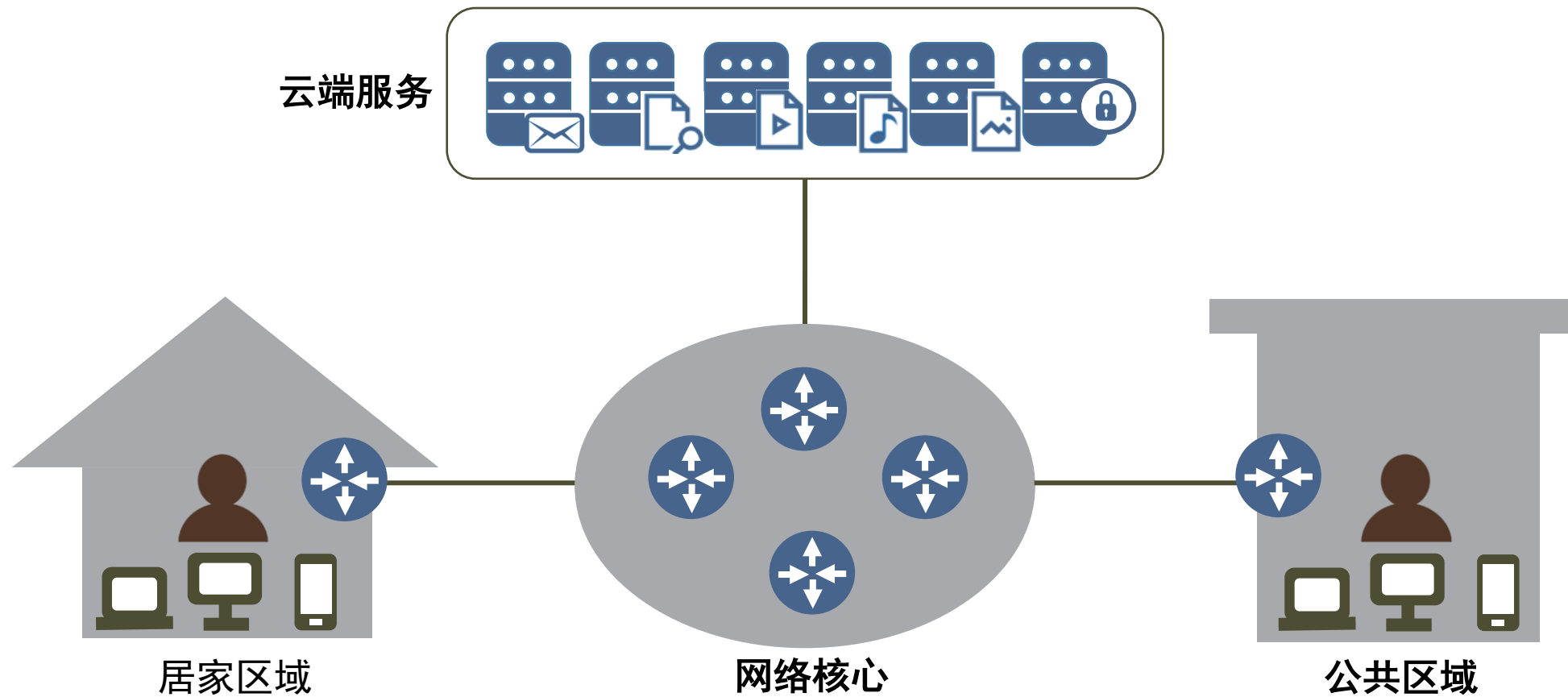


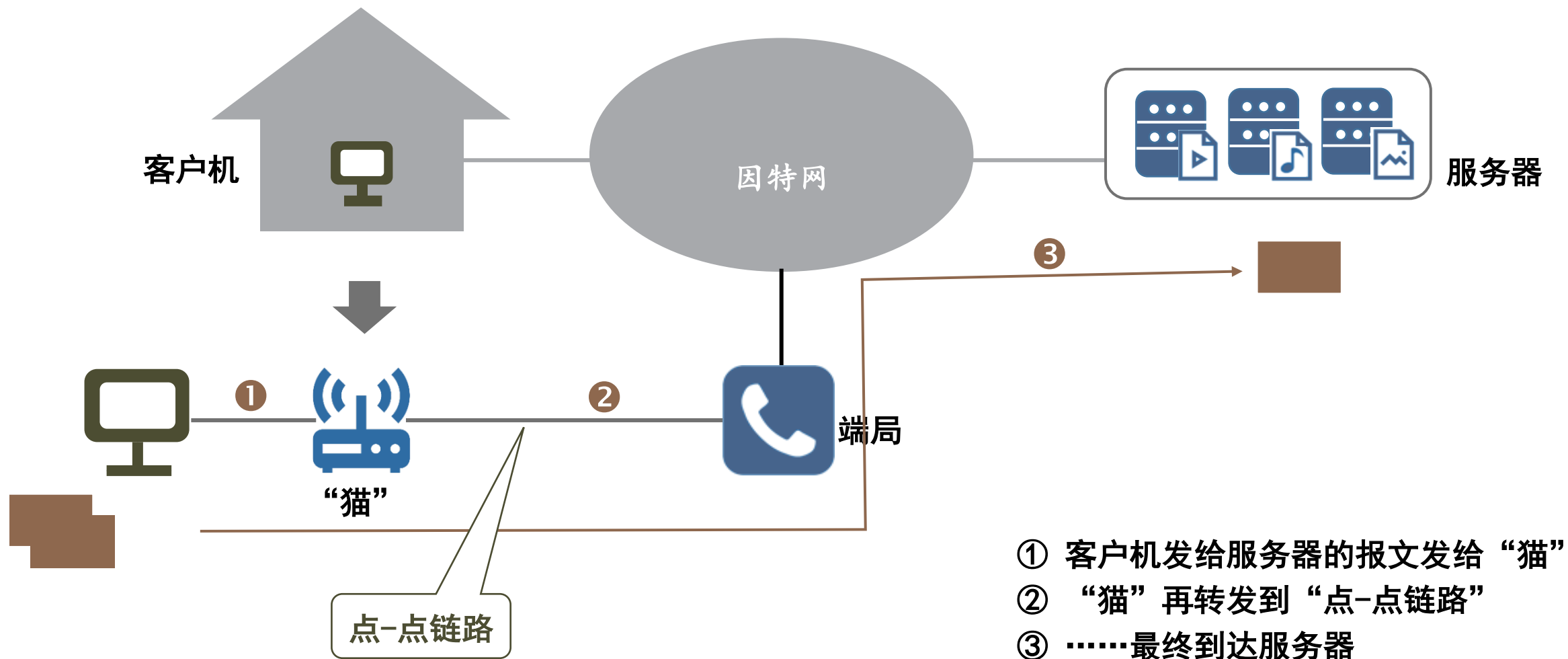
数据链路层概述



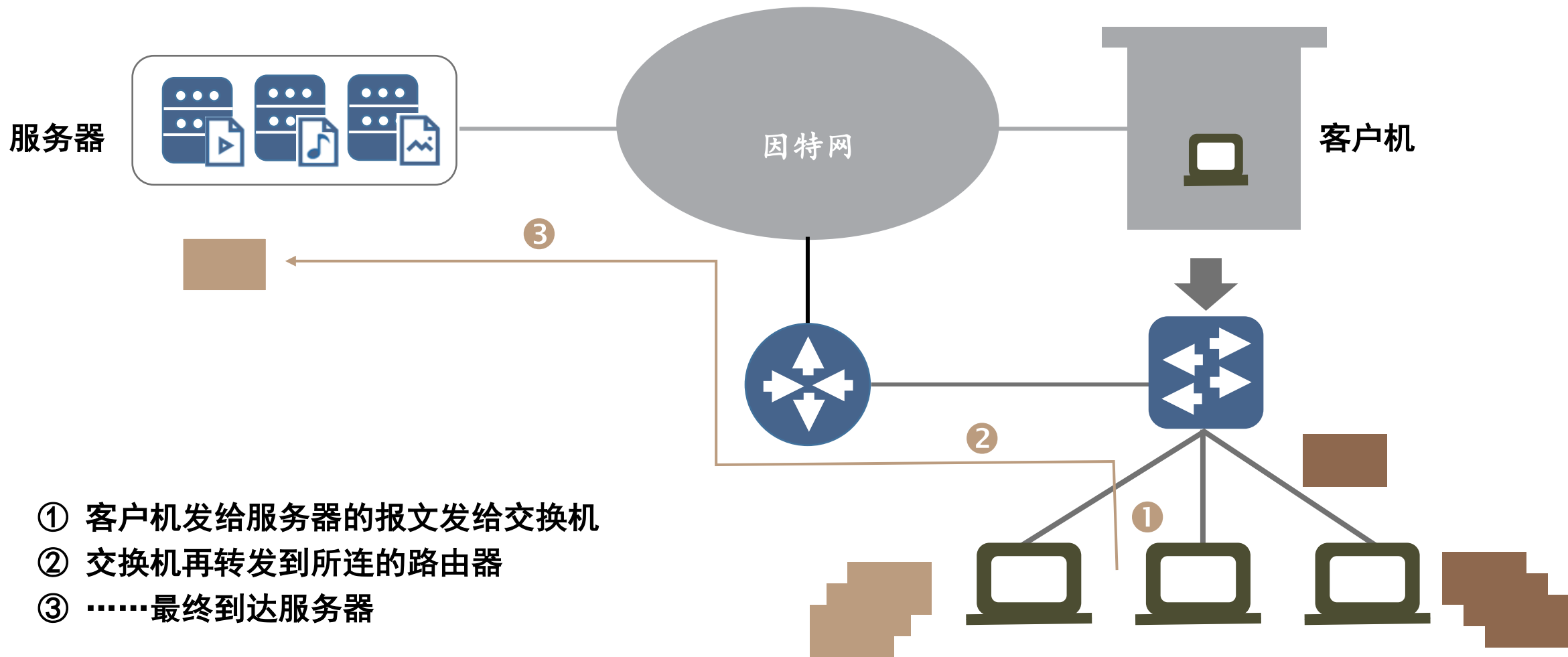
网络接入与信道访问



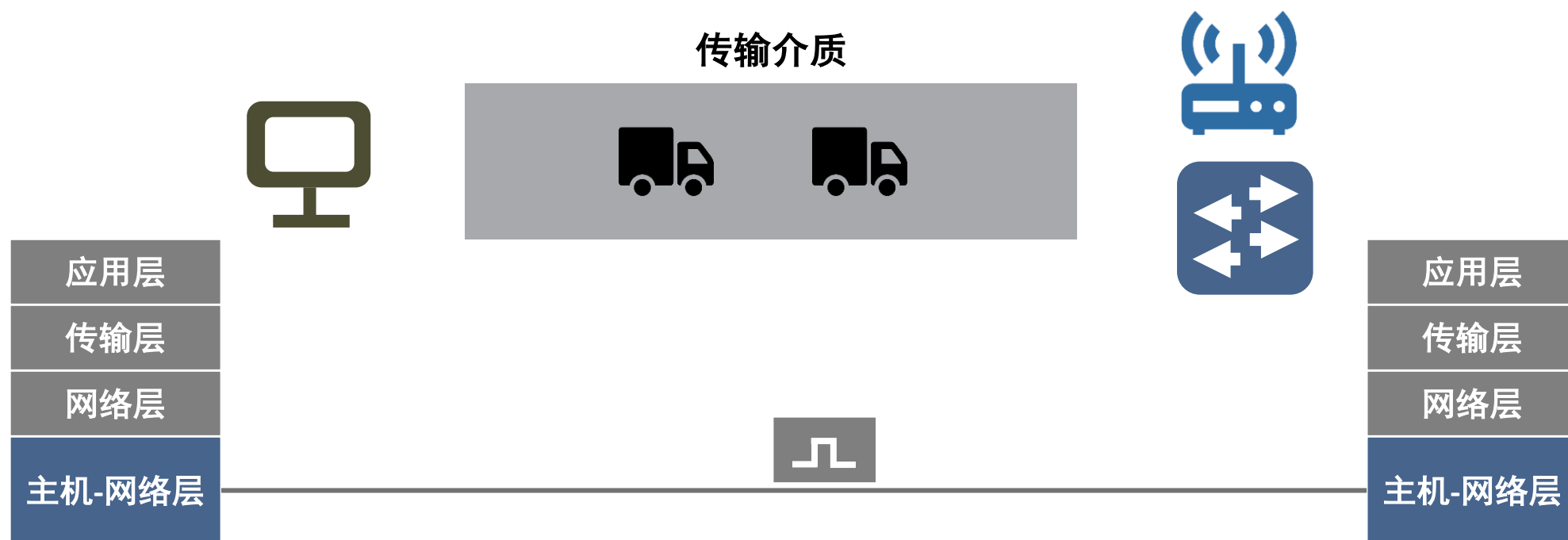
网络接入与点-点链路



信道访问与广播链路



如何通过介质发送报文



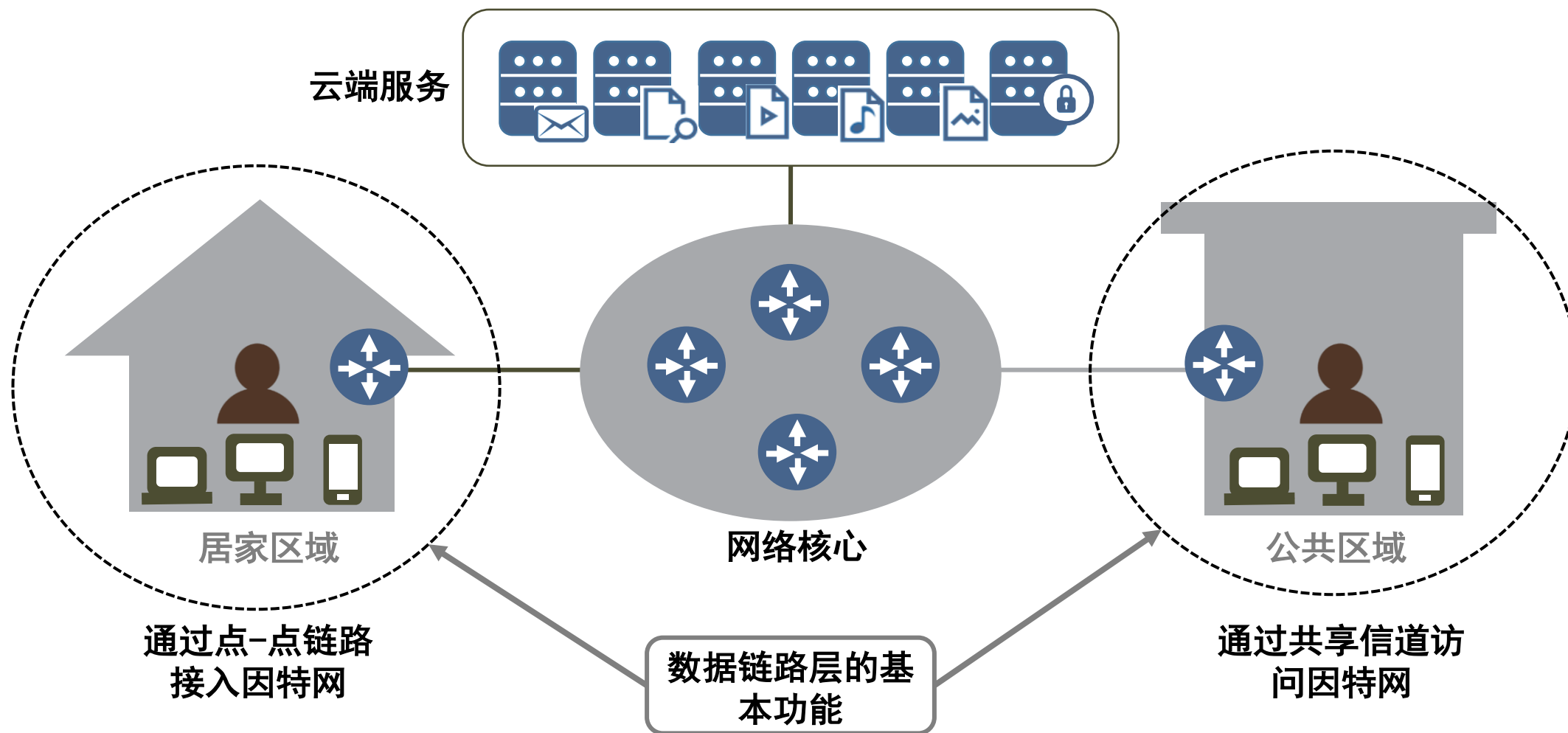
?

- 运载报文的“车”什么样？
- 如何使得“车”在介质上的随机信号中脱颖而出？

- 链路层的基本服务是将数据报从一个节点沿着通信链路移动到另一个节点
- 链路层服务的细节取决于该条链路所用的链路层协议



数据链路层的作用

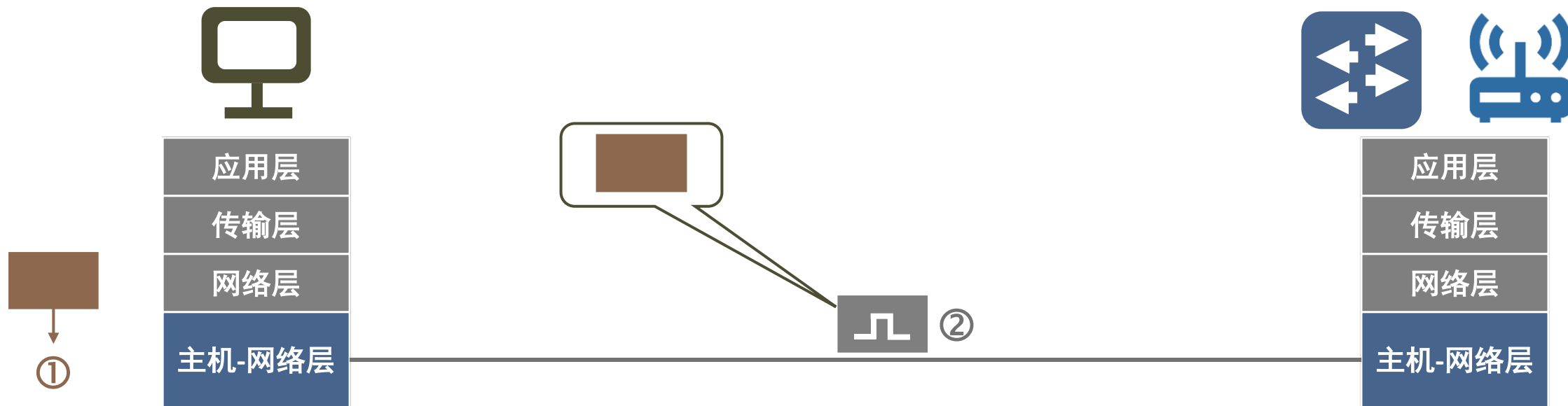


数据链路层面临的问题



数据链路层的上下层

帧：在数据链路上交换数据的单位。



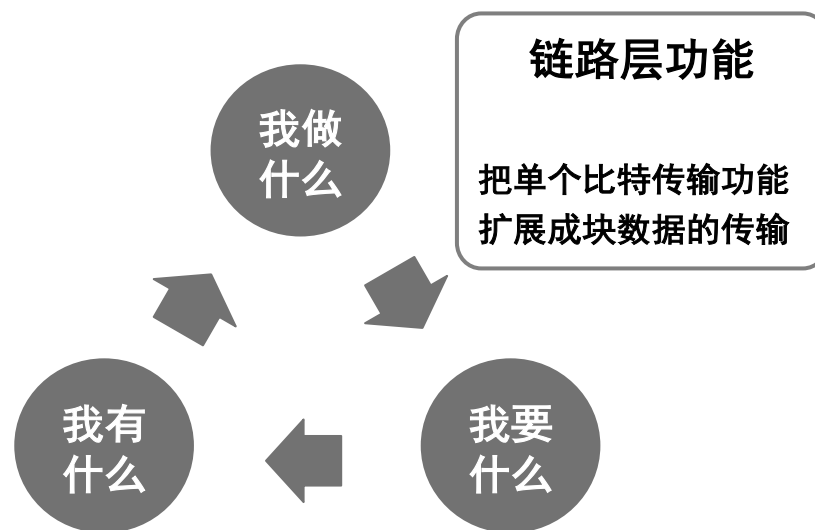
- ① 网络层通过接口把数据包交给链路层
- ② 链路层把包装在本层协议规定的格式中（帧）



数据链路层基本任务



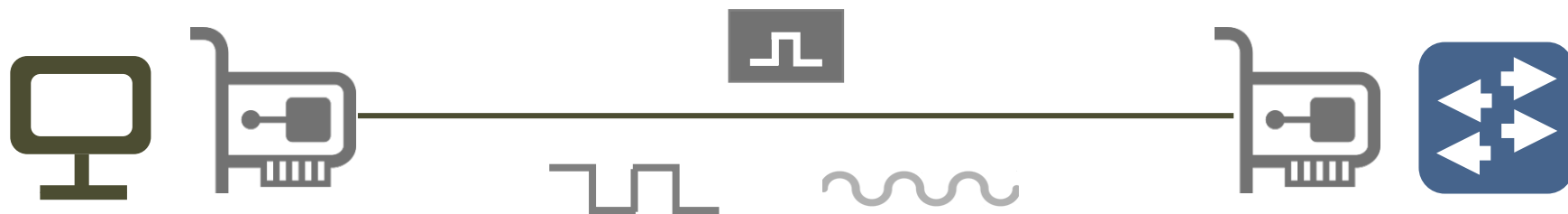
基础：网卡物理层针对具体介质产生相应信号，完成单个比特的传输。



目标：根据向网络层提供的服务类别，完成一块数据的传输，并达到相应服务质量。



帧的发送和接收同步



① 如何识别一个帧的开始？



需要有一种方法标识一个帧的开始和结束

成帧

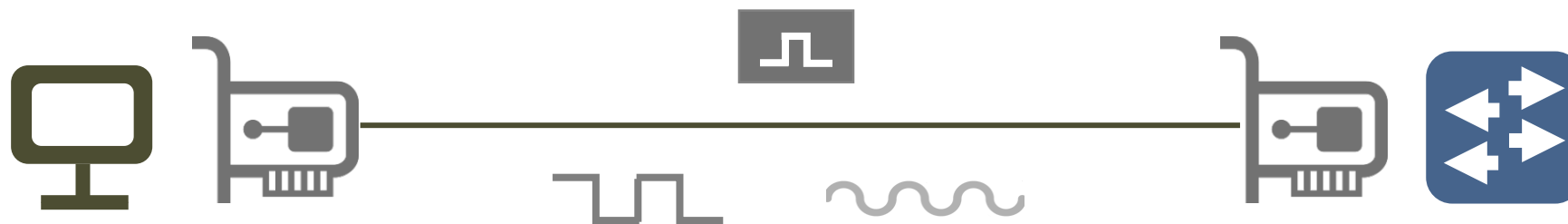


北京大学



Frame : 数据帧

传输差错的检测和处理



② 帧的传输出现错误、丢失或重复怎么办？

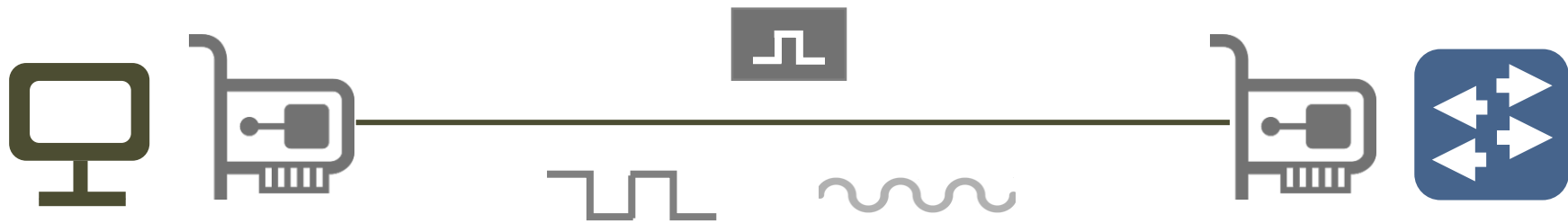


差错控制

- 需要一种方法，使得接收方在收到一串二进制后可以判断其接收的正确性
- 检测发现出现错误时要进行相应的处理



发送方和接收方能力均衡



③ 如果发送方和接收方的收发和处理能力不一致，出现一方快速发送另一方来不及接收怎么办？



需要一种控制机制，使得能力不一致的收发双方协同发送和接收行为，保证数据帧的正确传输。

流量
控制



链路的有效利用率



④ 当同时有控制信息和数据信息需要发送时先发哪个？

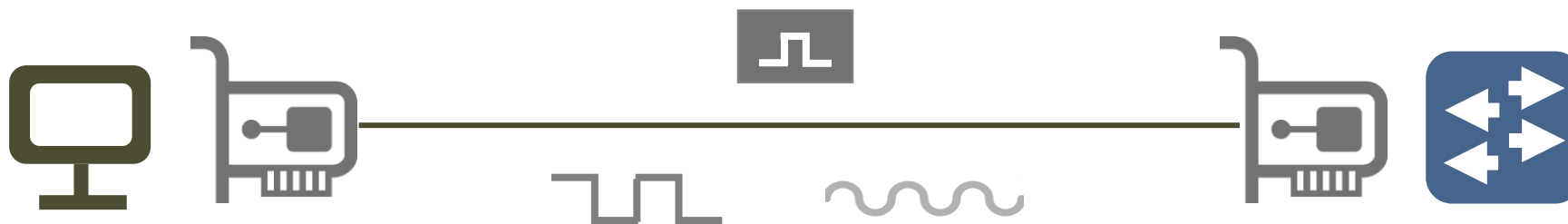


需要设计出一些技术机制，尽可能提高线路的利用率，由此降低传输成本。

滑动窗口



共享介质的访问控制



⑤ 如果收发双方所在的网络为广播网络，即所有节点通过同一个共享介质发送和接收，谁来决定该介质的使用权？

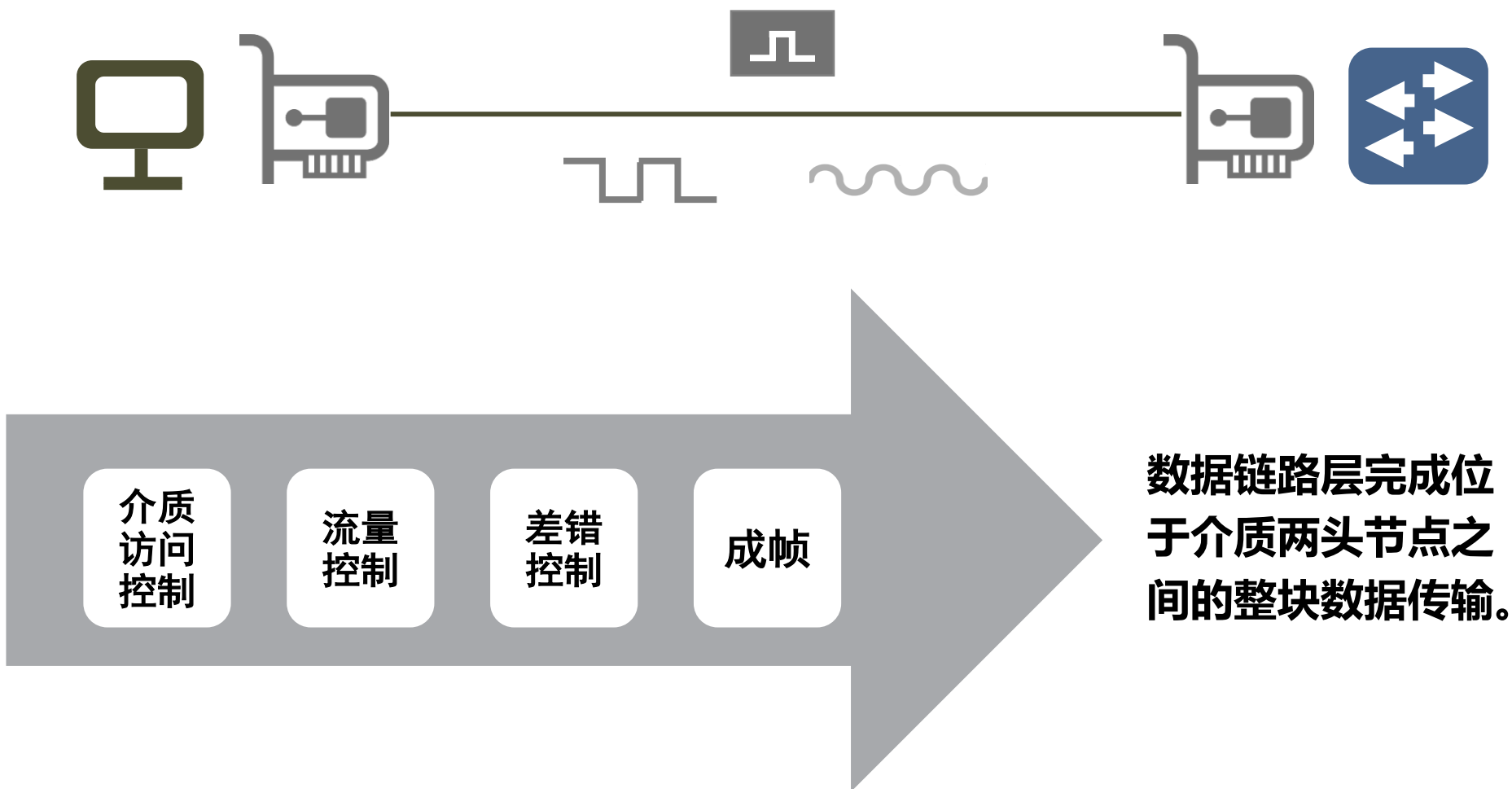


介质访问控制

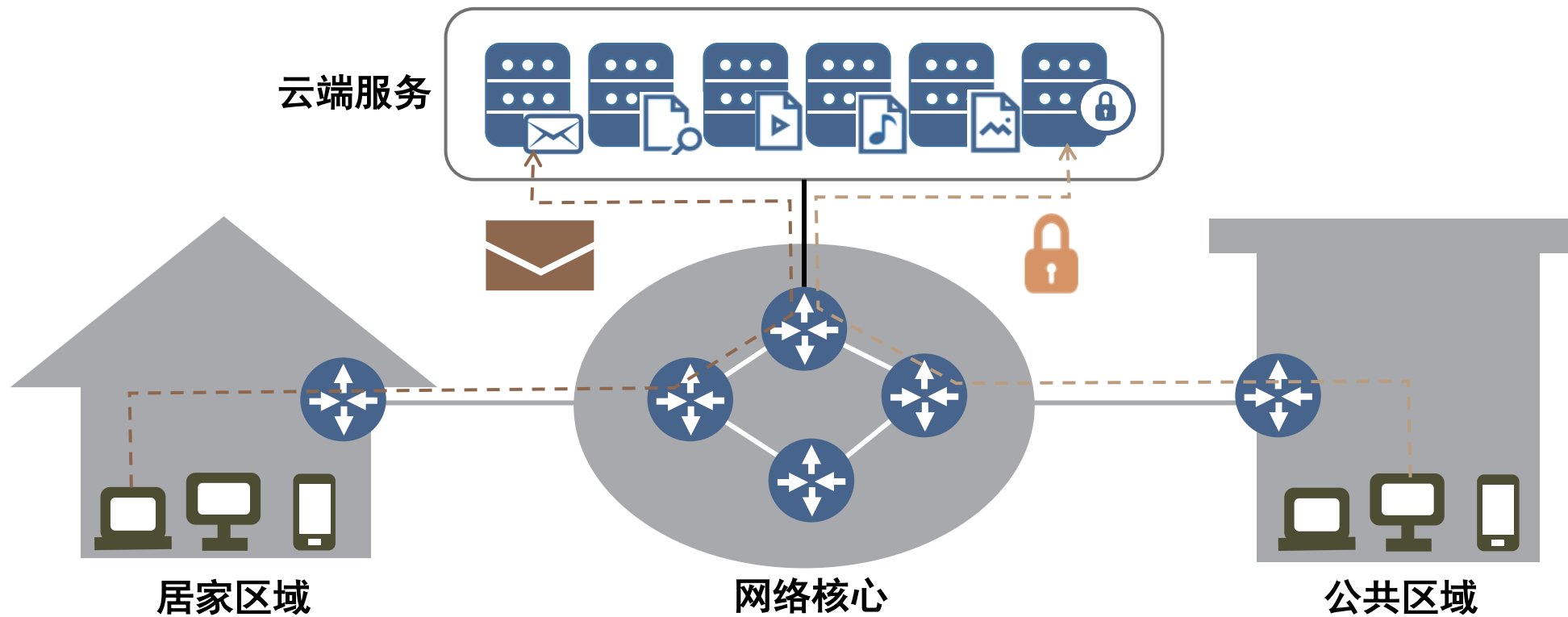
必须有一套控制或仲裁机制，公平地把共享介质的带宽分配给所需要的节点，同时要兼顾广播链路的利用率



数据链路层基本功能



数据链路层的作用



数据链路层负责整条路径上的“点-点”数据传输。

数据链路层是网络层与物理介质之间的桥梁。

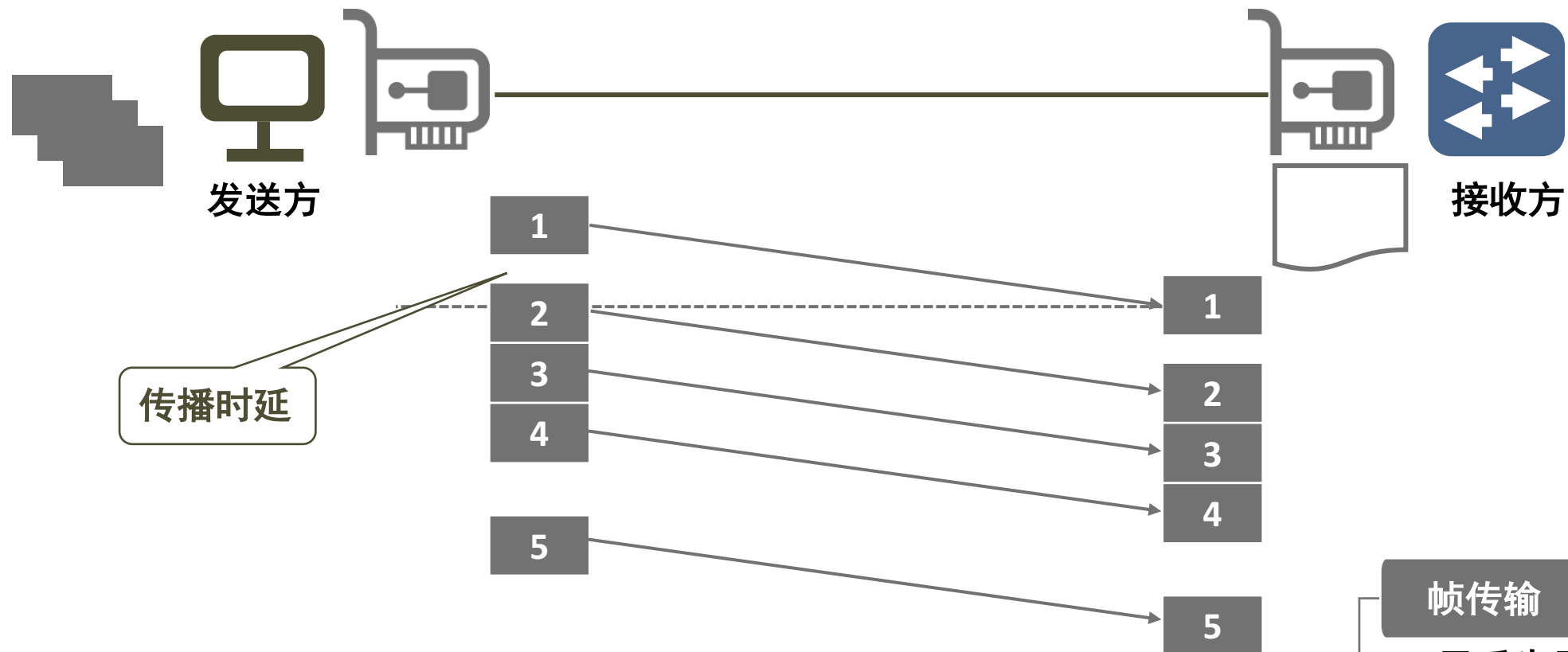
数据链路层为网络层的“存储-转发”提供支持。



数据链路层的 流量控制



帧传输模型



流量控制：用来确保发送实体发出的数据不会覆盖接收实体已接收的数据。

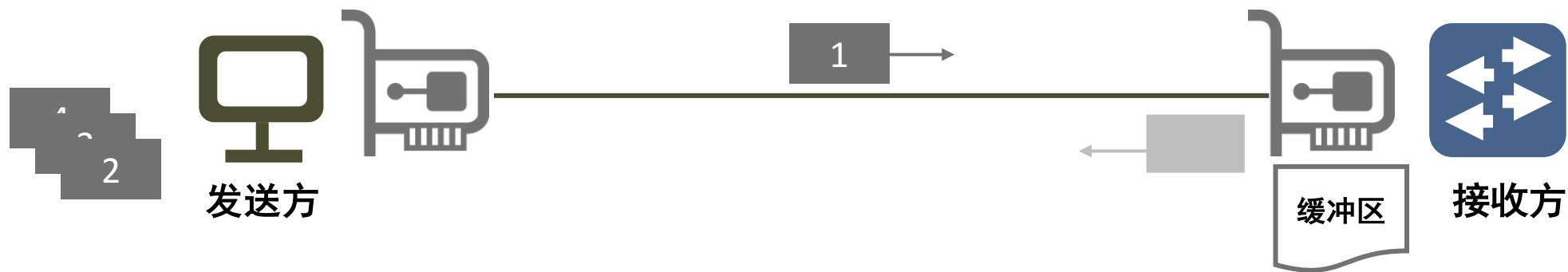
帧传输

- 无丢失无差错
- 按发出次序到达
- 具有不等长的延时



停一等流量控制

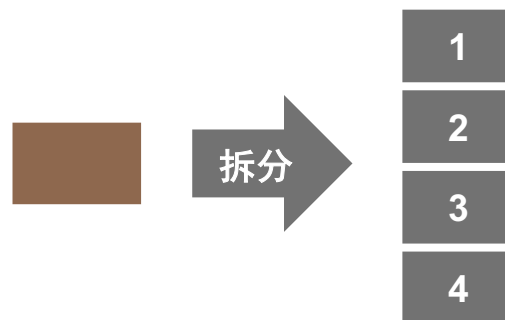
停等式控制：发送实体发送一个帧，接收实体收到后发回一个对该帧的确认，表示同意接收下一个帧；发送实体必须等待，收到确认后才能发送下一个帧。



接收实体简单地用停止发送确认的方式来阻止发送端的数据流。



停—等流量控制的特点



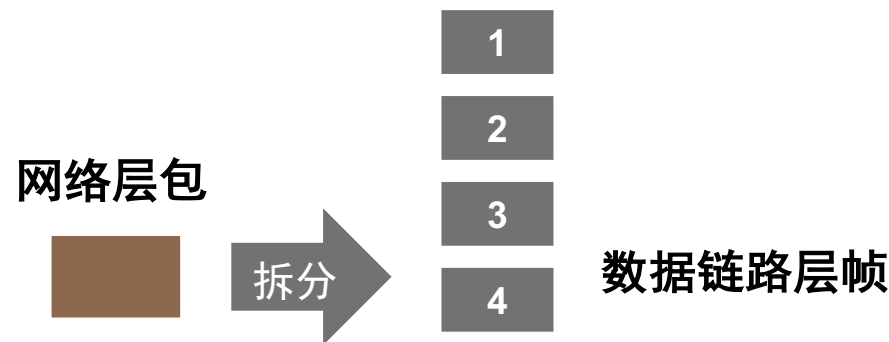
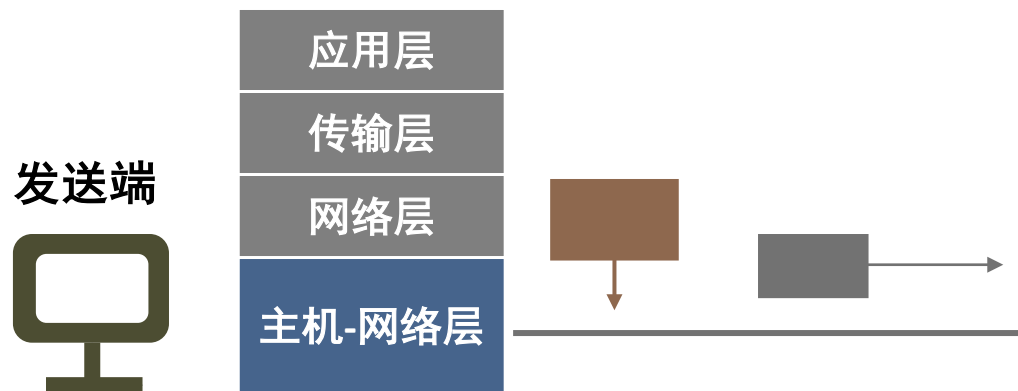
优点

- 控制简单
- 适用于当包被分成数量不多的帧发送时

缺点

- 效率不高
- 当一个包用多个帧发送且链路传播时延大于发送时间时

为什么要分拆数据包？



?

为什么数据被分成
许多个较小帧发送？

可能的原因

- 接收缓冲区大小受到限制
- 传输数据愈长出错可能性愈大
- 一个节点不能占用信道时间过长

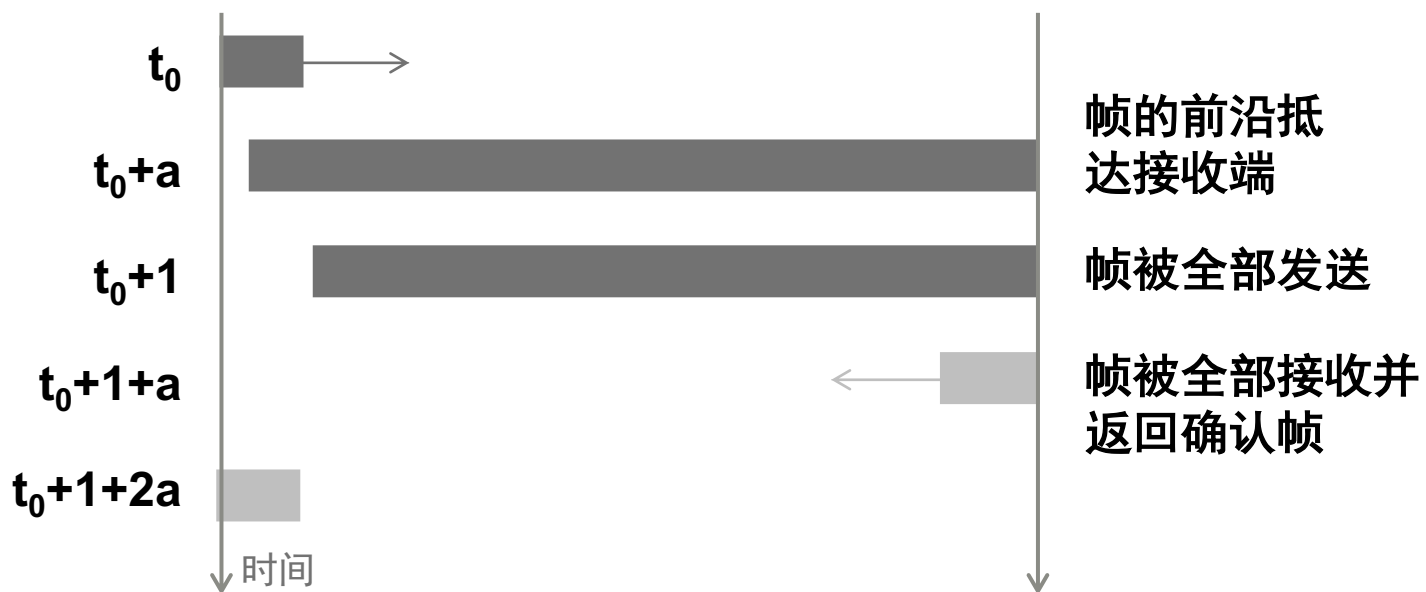


停一等流量控制机制性能



●当传播时延<发送时间

- 忽略确认帧的传输时间
- 信号的传播时延为 a
- 一帧的传输时间为1



链路空白处即为
链路空闲时间。

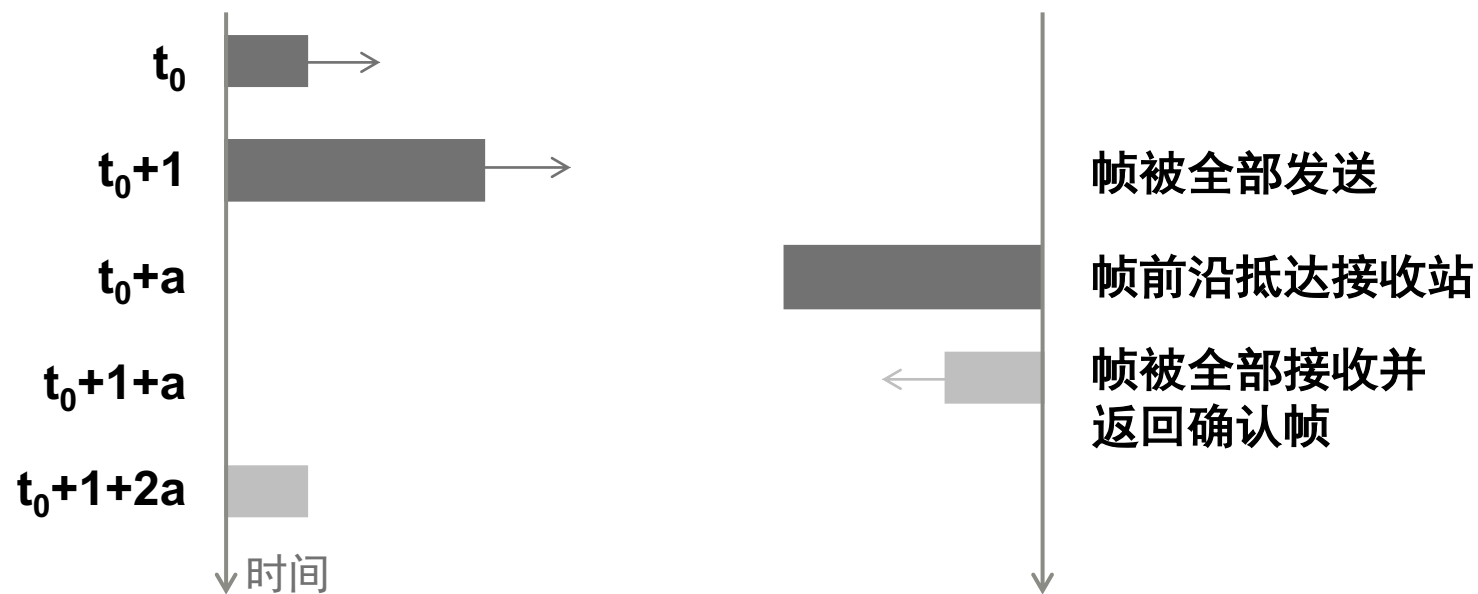


停一等流量控制机制性能（续）



●当传播时延>发送时间

- 忽略确认帧的传输时间
- 信号的传播时延为 a
- 一帧的传输时间为1



链路的大部分时间都
空闲！



长距离 vs. 短报文

示例1：在如图所示的网络中，两个网络设备通过**ATM**相连。

- 采用光纤作为传输介质
- 两端之间距离为**1000km**
- 数据帧长为**53B = 424b**
- 数据率为**155.52Mbps**



$$\text{发送时间} = 424 / (155.52 \times 10^6) = 2.7 \times 10^{-6} \text{ s}$$

$$\text{传播时延} = 10^6 / 3 \times 10^8 = 0.33 \times 10^{-2} \text{ s}$$

**距离越长，帧越短，
传输效率越低！**

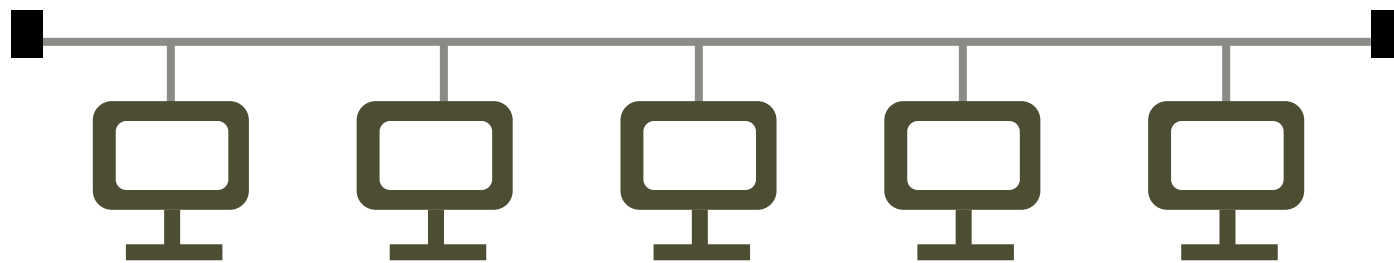
$$\text{线路传输效率} = \frac{\text{帧发送时间}}{2 \times \text{传播时延} + \text{帧发送时间}} = 0.004\%$$



高速率vs.近距离

示例2：在如图所示的广播网络中，5个节点共享一个传输介质。

- 两点之间的最大距离 100m；
- 数据速率100Mbps；
- 信号传播速率 2×10^8 m/s；
- 数据帧长1000b；



$$\text{发送时间 } t_{\text{frame}} = 1000 / (100 \times 10^6) = 10 \times 10^{-6} \text{ s}$$

$$\text{传播延迟 } t_{\text{prop}} = 100 / (2 \times 10^8) = 50 \times 10^{-6} \text{ s}$$

距离越短传播时延越小，
链路利用率越高！

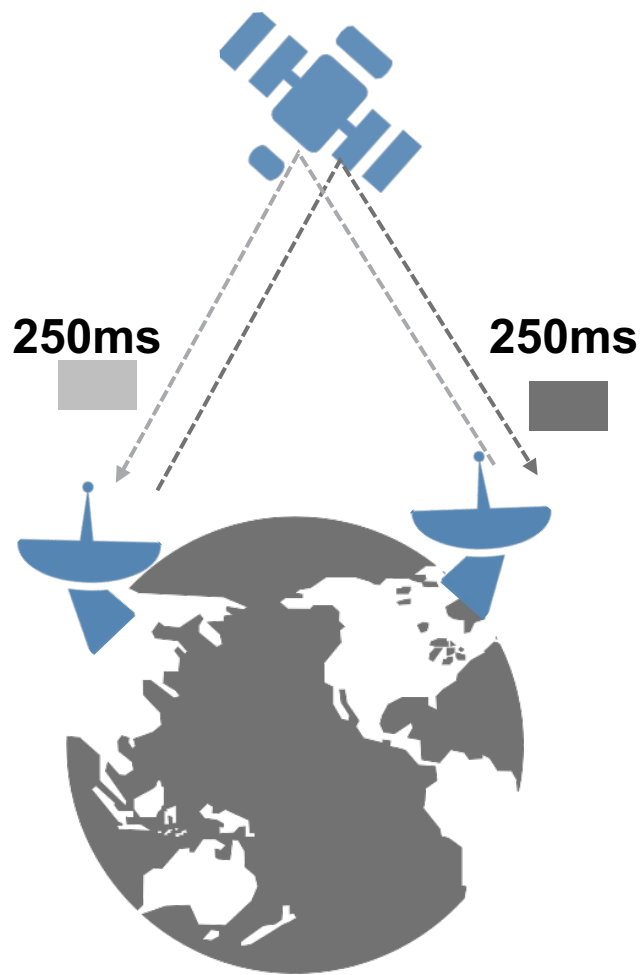
$$\text{线路传输效率} = \frac{\text{帧发送时间}}{2 \times \text{传播时延} + \text{帧发送时间}} = 99\%$$



滑动窗口控制机制



长距离vs.长报文



假设：在如图所示的卫星通信系统中，采用了停等式流量控制。

- 数据帧长1000b
- 发送速率为50kps
- 传播时延为250ms

- $t = 0$ 开始发送；
- $t = 1000/50000 = 20\text{ms}$ 后发送方发完第一帧
- $t = 20 + 250 = 270\text{ms}$ 后接收方接收完第一帧
- $t = 270 + 250 = 520\text{ms}$ 发送方收到第一帧的确认

$$\text{信道利用率} = 20/520 = 3.8\%$$

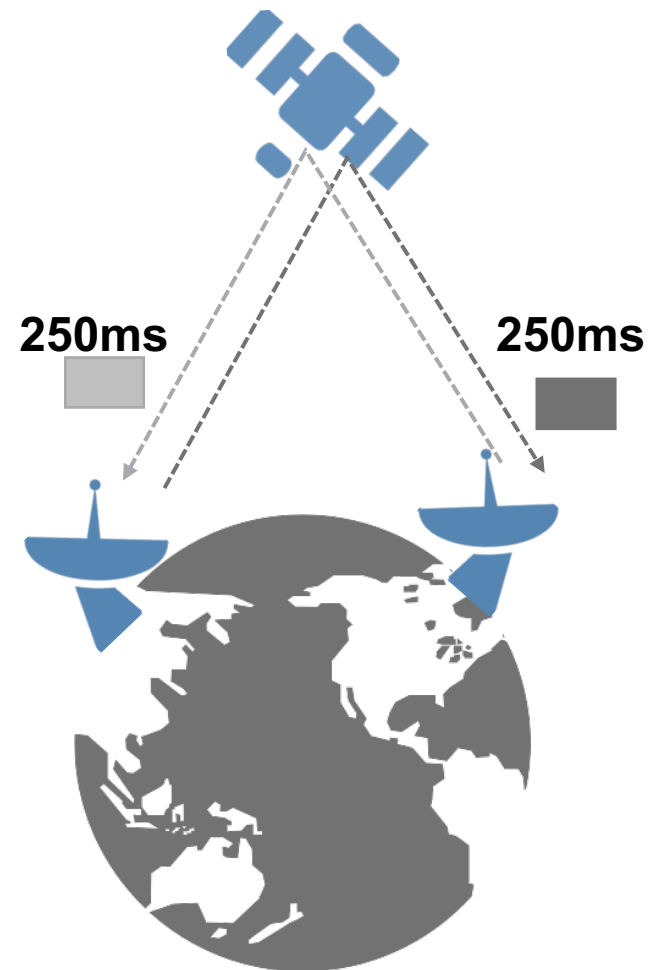


如何提高信道利用率

结论：使用停等式流量控制，高延迟的卫星信道信道利用率仅3.8%

原因	解决方法
信道时延使得发送方等待时间过长	利用传播延迟连续发送n帧(连续式)

$$n = 1 + \frac{\text{传播延迟}(500)}{\text{发送一帧所需时间}(20)} = 26$$



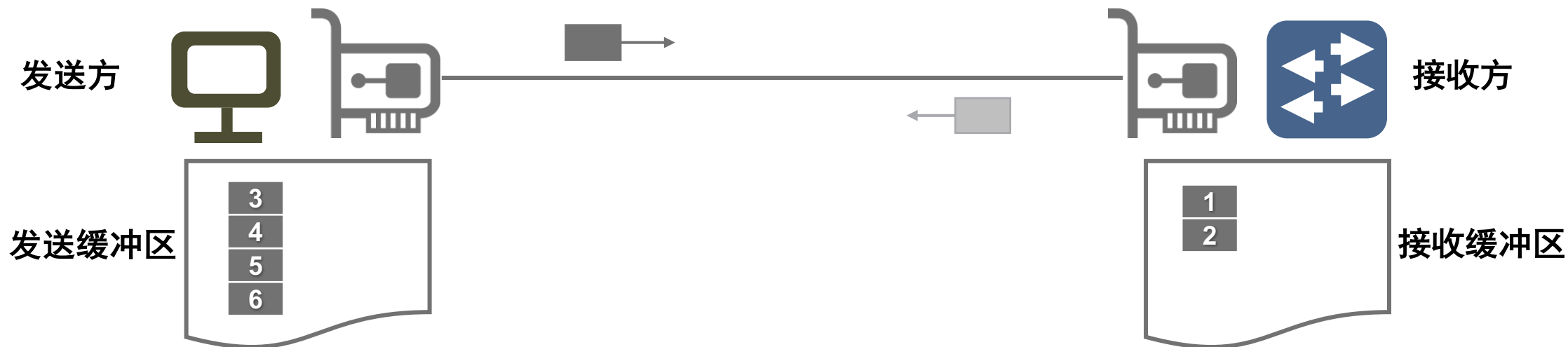
滑动窗口流量控制

滑动窗口控制：利用窗口控制连续发送的数据量。

接收方收到一个帧后发回一个确认，该确认帧包括了接收方想要的下一个帧号。

滑动窗口控制基本条件

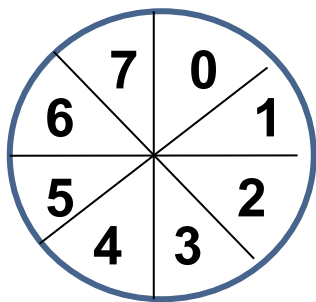
- 两个节点通过全双工链路连接
- 收发双方为n个帧分配缓冲区
- 每个数据帧拥有一个序号



序号以及序号空间特性

序号空间：帧序号的取值范围。通常协议规定了帧格式中表示序号的二进制比特数

例如：序号用二进制 n 位表示， 则取值范围是0、1、2、 \dots 、 2^n-1 。



序号空间循环特性

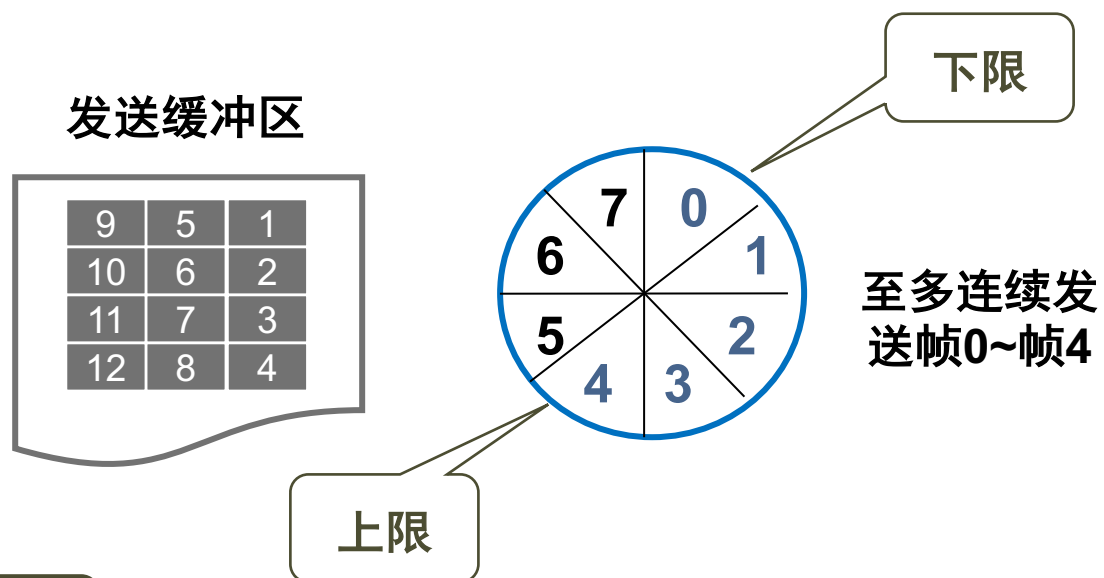
- 被发送的帧按顺序编号
- 当用尽最大序号后下一个帧的序号回滚到0

- 如果序号用二进制3个比特表示， 则序号取值范围是0, 1, 2, 3, 4, 5, 6, 7,
- 当最近发出帧的编号为7时，下一个待发送帧的编号为0.

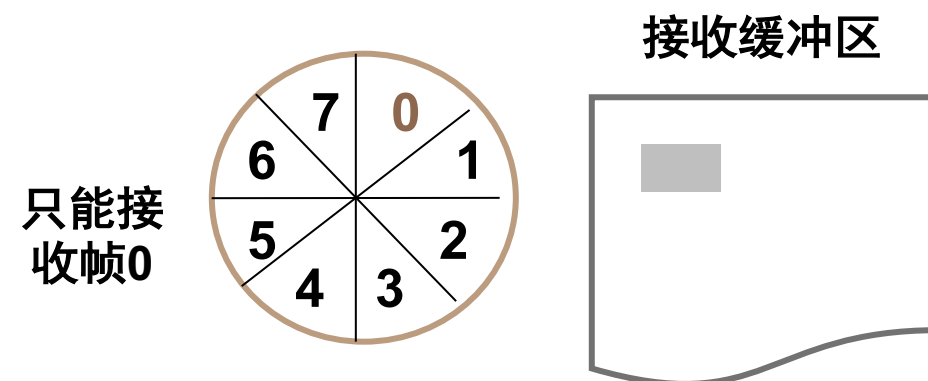


发送窗口 vs. 接收窗口

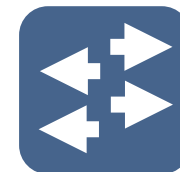
发送窗口(W_T)：给出允许发送方连续发送的帧序号。



接收窗口(W_R)：给出允许接收方接收的帧序号。



- 发送窗口和接收窗口可以不一样大
- 发送窗口和接收窗口沿顺时针方向滑动
- 窗口不是输出/输入缓冲区，只存放待发送/接收帧的序号



滑动窗口的控制原理

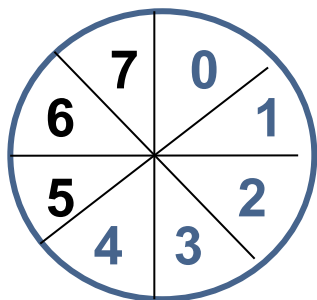
假设：发送窗口 $W_T=5$ ；接收窗口 $W_R=1$



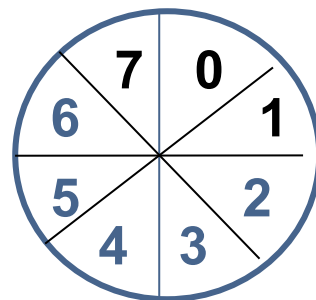
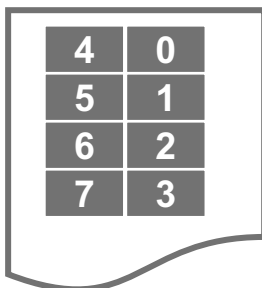
- 发送方：可以连续发送5个帧
- 接收方：只能一个帧一个帧地接收

发送窗口

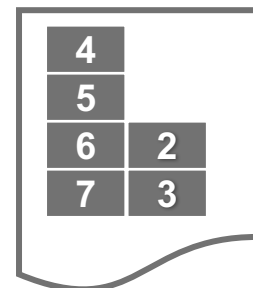
t_0 : 连续发帧0~4



发送缓冲区



发送缓冲区

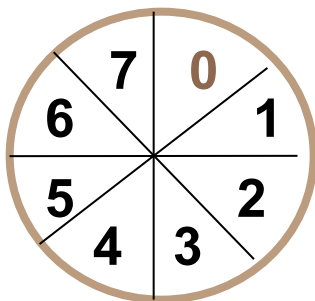


t_i : 收到帧1确认

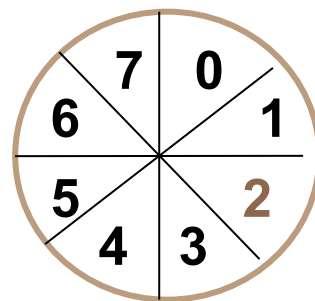
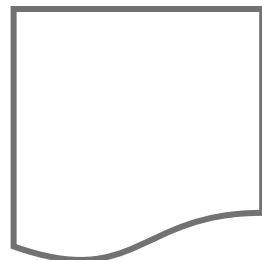
时间

接收窗口

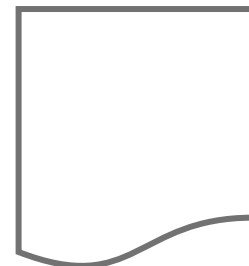
t_0 : 准备接收帧0



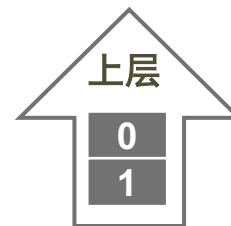
接收缓冲区



接收缓冲区



t_i : 准备接收帧2

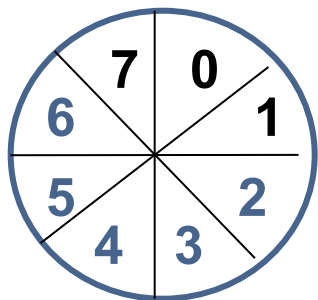


北京大学

滑动窗口的控制原理

窗口操作：在连续发送的情况下，利用模数运算，让序号循环地被使用，但要在收发两端进行适当的控制。

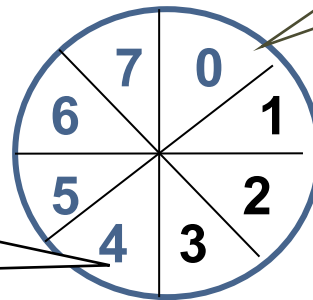
发送窗口



t_i : 收到帧1确认

t_{i+1} : 收到帧3确认

发出两个帧窗口下限前移2个序号

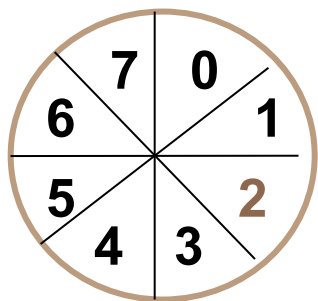
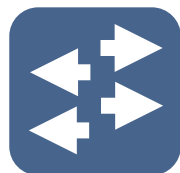


收到两个确认窗口上限前移2个序号

维持发送窗口大小为5

→ 时间

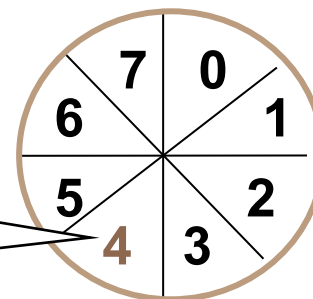
接收窗口



t_i : 准备接收帧2

t_{i+1} : 准备接收帧4

发出一个确认窗口前移1个序号



将收到的帧交给上层后再接收下一个帧



北京大学

滑动窗口基本原理

基本原理

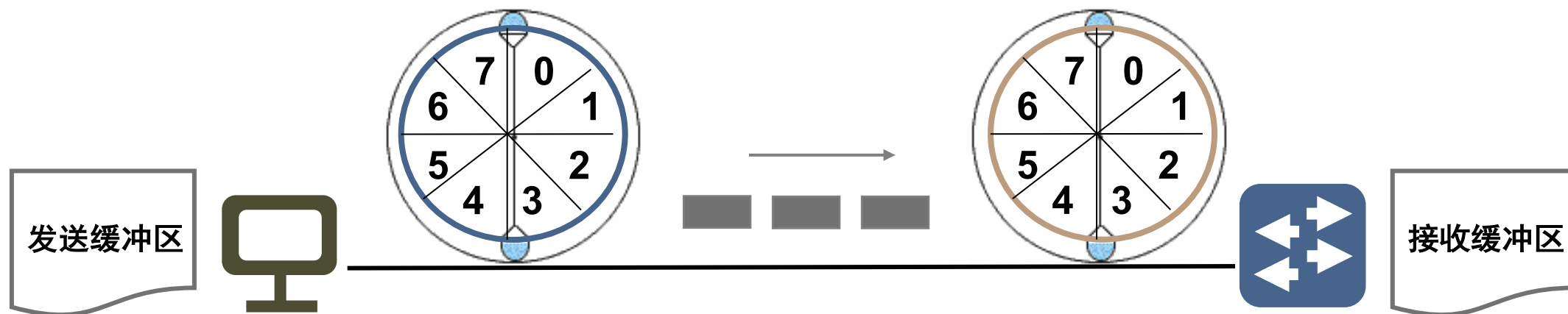
- 发送方和接收方的滑动窗口控制发送和接收
- 发送窗口和接收窗口在序号上滑动

发送方

发送方只能发送帧号落在发送窗口内的帧，收到确认后，将发送窗口向前推进一格

接收方

接收方每收到一个帧，校验正确且序号落在接收窗口内，就向前推进一格，并发出确认帧



基于滑动窗口的流量 控制机制



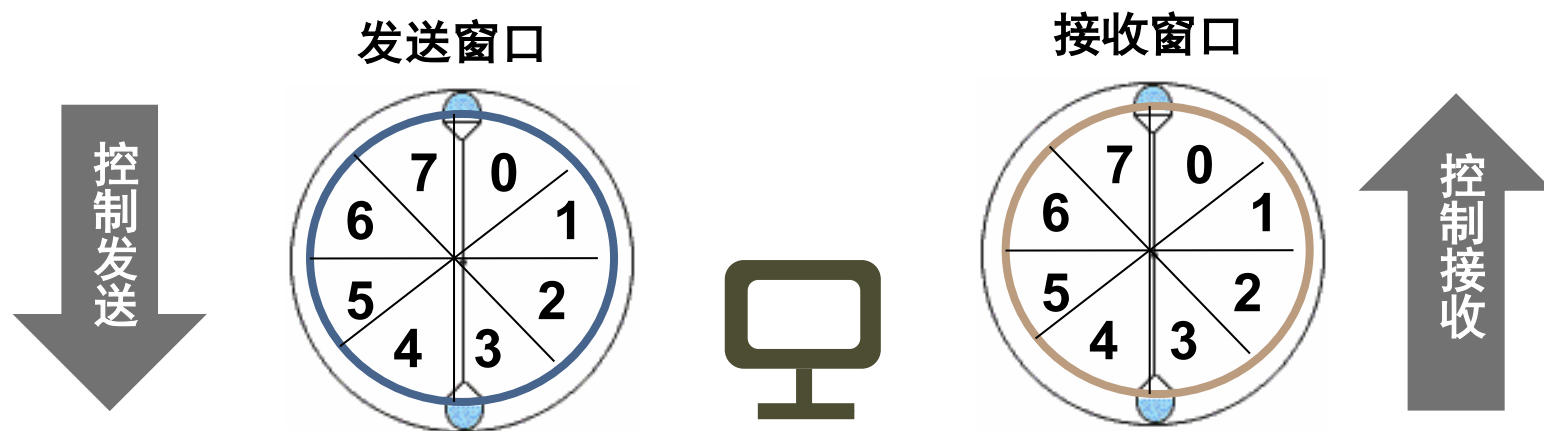
滑动窗口应用于流量控制

流量控制帧

- 肯定确认帧RRn: 准备接收从n开始的帧
- 否定确认帧RNRn: 已接收到 n-1的所有帧但不能再接收了

双向传输（全双工）

- 每个站都保持两个窗口
- 发送窗口控制正向发送
- 接收窗口控制逆向接收



捎带确认技术

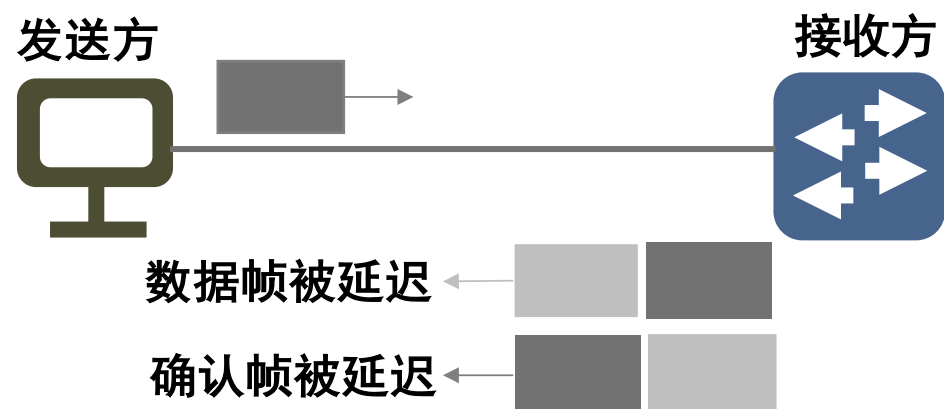
?

对于全双工操作来说，
接收方有数据要发，还
要回复确认帧，怎么办？

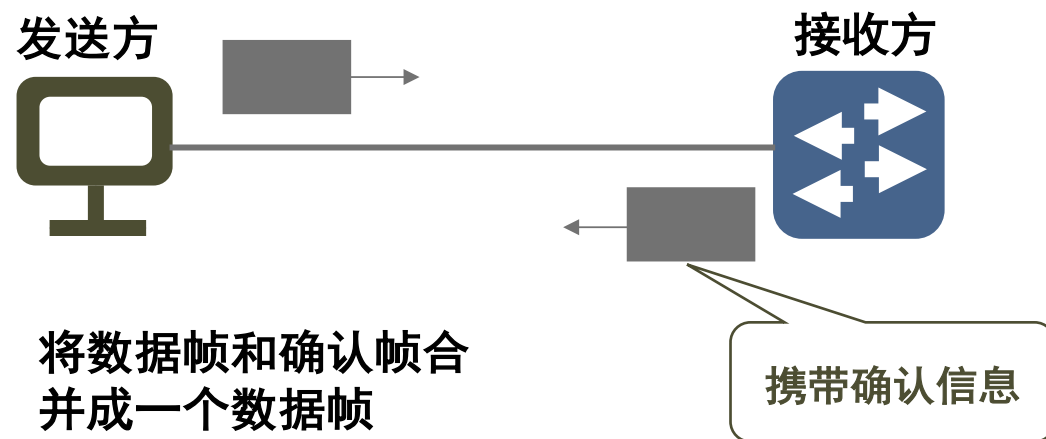
解决方案

**捎带技术：既有数据又有确认
时，将两者合在一个帧中发送，
即以数据帧捎带上确认信息。**

普通确认：数据帧含有序号
确认帧含有确认号



捎带确认：数据帧含有序号+确认号
确认帧含有确认号



北京大学

确认帧

数据帧

累计确认技术

普通确认：假设发送方发送了5个数据帧，接收方没有数据要发。则接收方要发送5个确认帧。

节省带宽

累计确认：接收方可对收到的K个帧 ($K < \text{发送窗口}$) 发一个确认，告知发送方已正确接收前(K-1)帧并期待第K个帧。

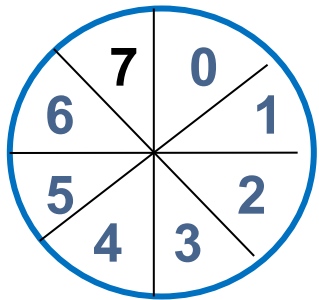


接收方用一个确认帧通知发送方自己的接收进展，由此节省了带宽。

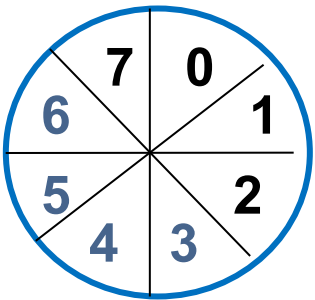


滑动窗口流量控制示例

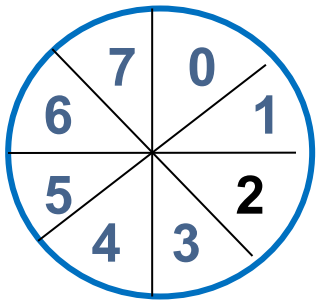
发送方



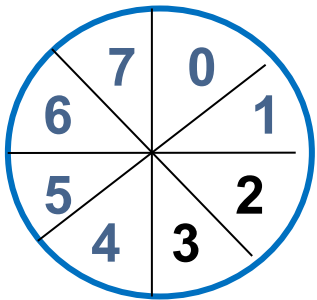
可发送帧0~帧6



连发帧0~帧2



收到确认帧



发送帧3~帧5

待发帧



t1



t2

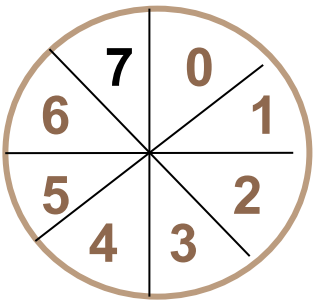


t4

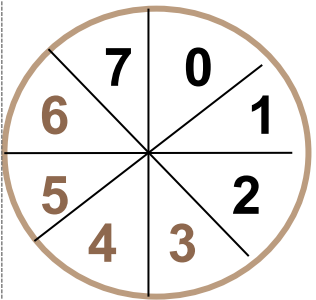
t5

时间

接收方



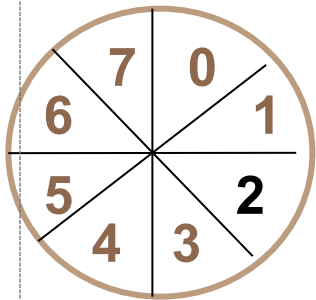
能接收帧0~帧6



接收帧0~帧2



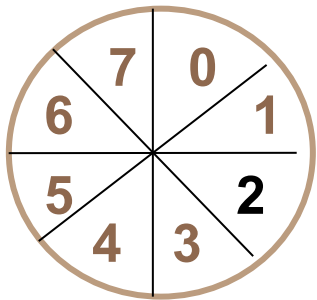
t3



反馈确认帧



t6



接收帧3~帧5

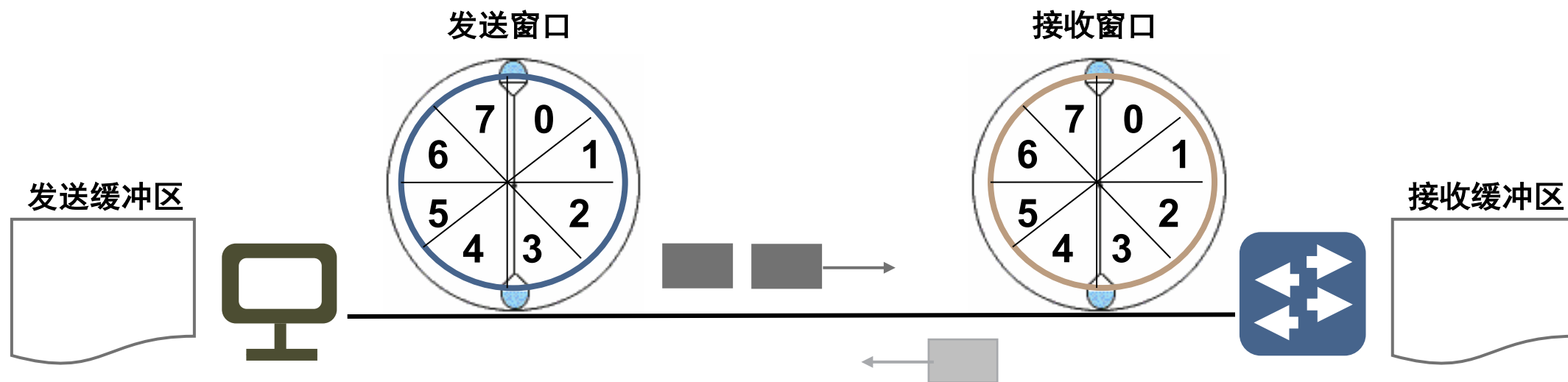


数据帧

差错控制vs. 停等式控制



滑动窗口控制机制的性能



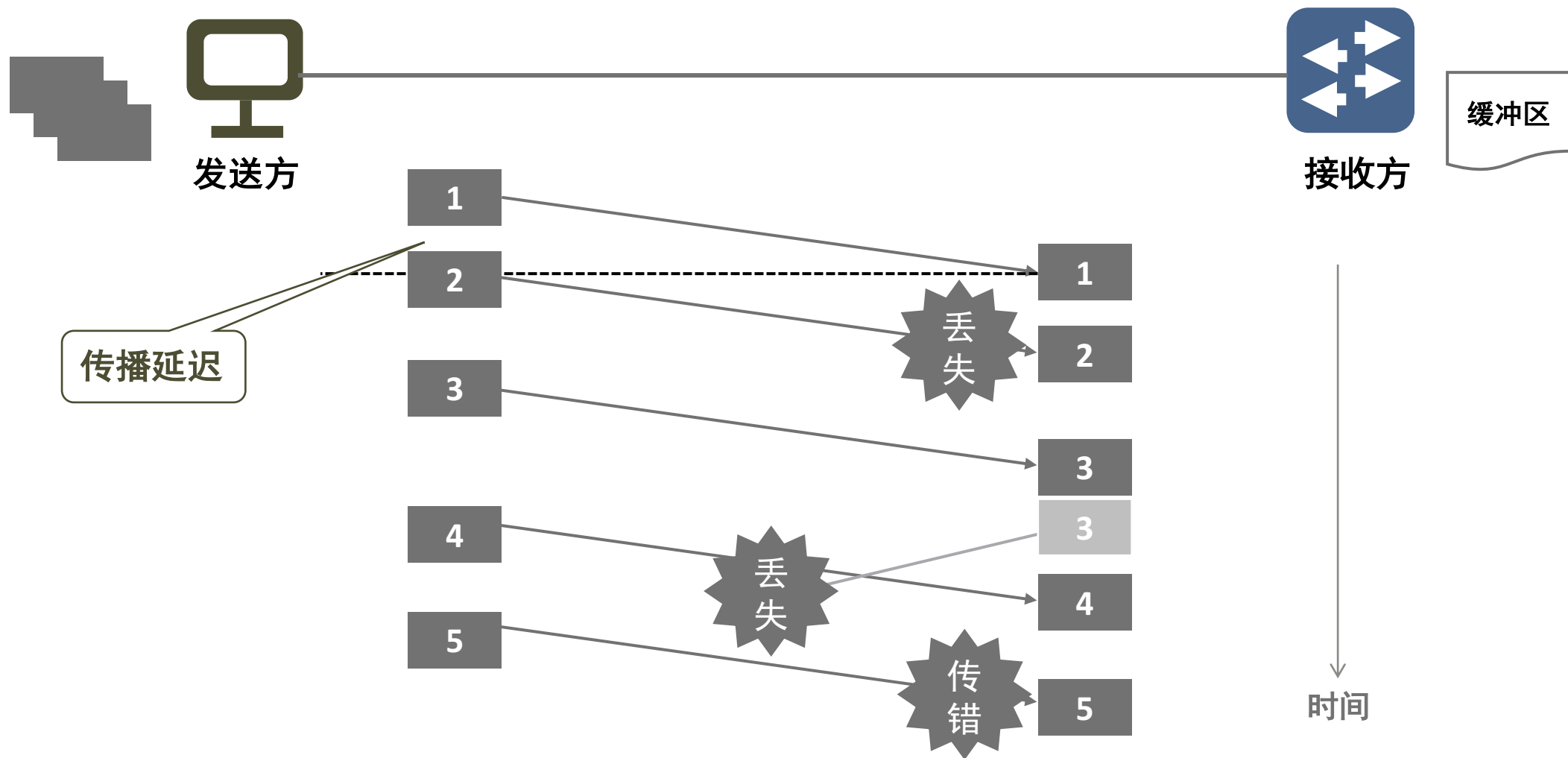
理想
链路利用率达到100%

差错控制

现实
帧的传输会出现错误

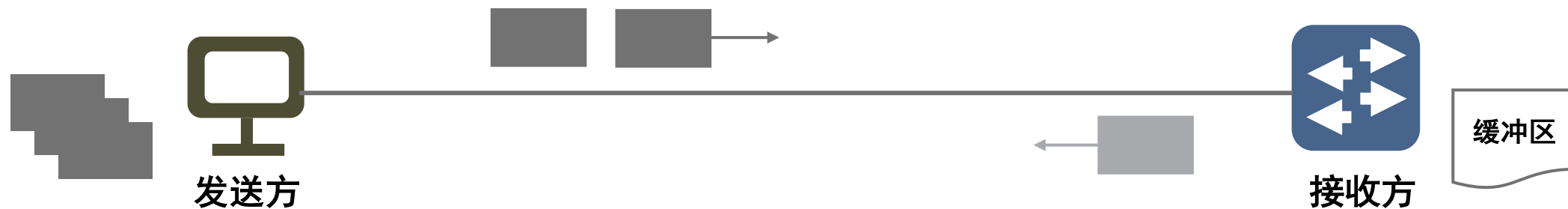


帧传输错误情形



差错控制

差错控制：指对传输的数据进行错误检测，并在发现错误时加以恰当的处理。



正确接收

帧按发出的次序到达，
且每个帧有不定长的
传输时延。

检错能力

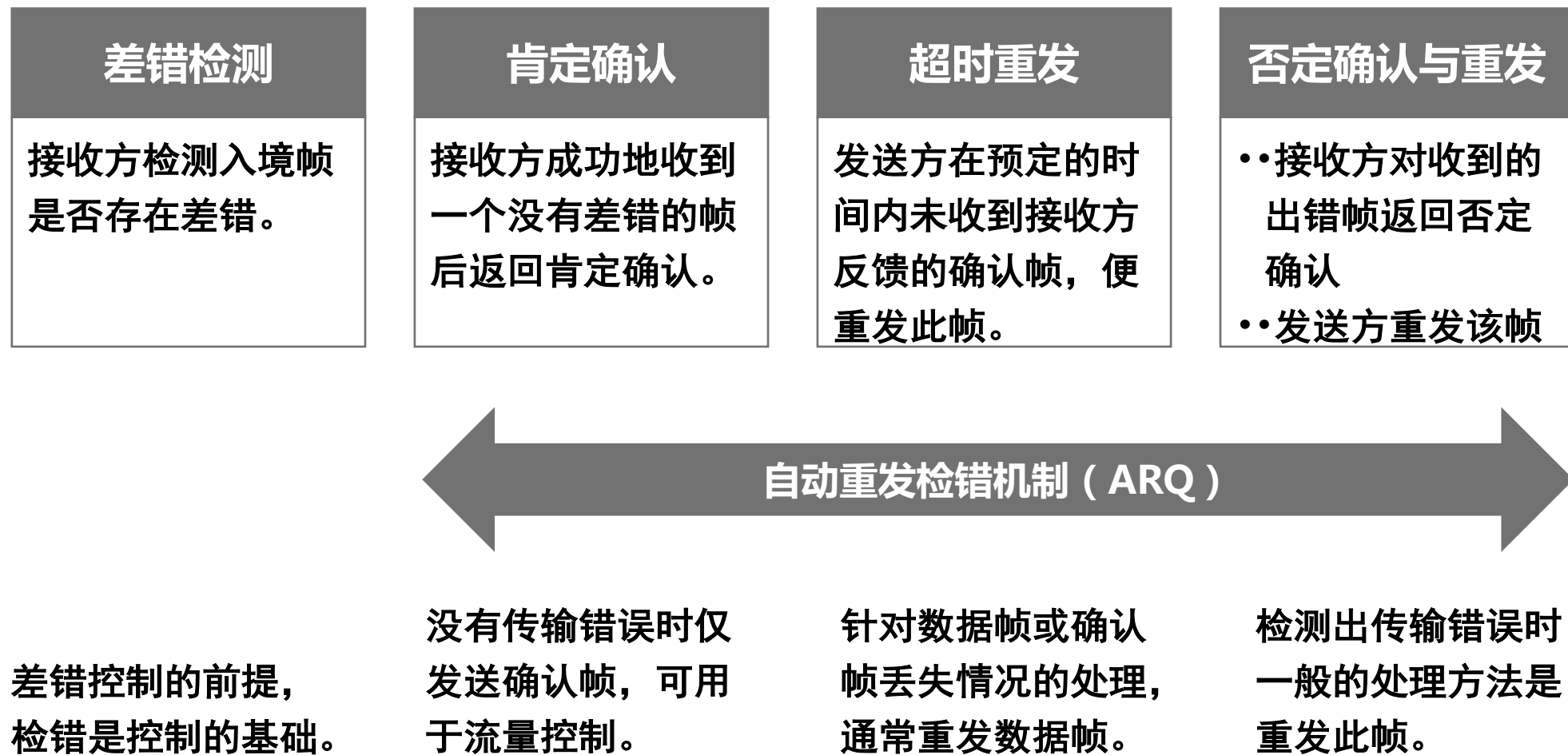
发现错误时，丢弃错
误帧，并要求发送方
重新传输该帧。

纠错能力

发现错误时，就地立
即加以改正，无需发
送方重传。



差错控制的基本功能



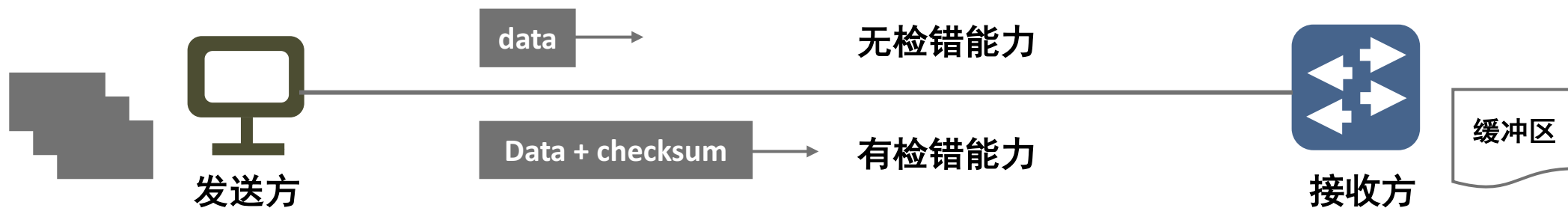
自动重发检错 (ARQ)

纠错编码：在信息序列中根据某种规则加入一定校验码。

校验码：根据待传数据按某种规律生成的额外比特，用来帮助接收方检验是否发生传输错误。

ARQ原理

- 发送方根据被传送数据，按一定规律加入一些校验码位，使数据和校验码有某种相关性，然后将数据和校验码都发送给对方；
- 接收方根据数据与校验码之间的相关规律进行检验，从而确定接收的数据是否出错，并通过反馈确认把检测结果告知发送方。



停-等式ARQ vs. 数据帧出错

停等式ARQ：发送方仅在收到当前帧的肯定确认后才能发送下一帧。

?

是否需要为帧分配编号

帧被破坏

接收方丢弃出错帧，发送方超时重发。

帧被丢失

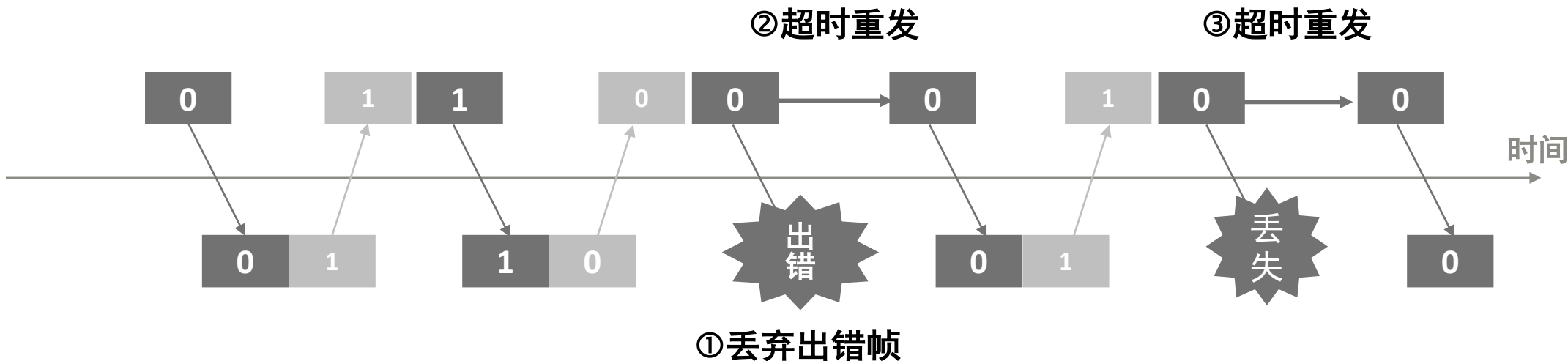
接收方不会有任何动作，发送方超时重发。



待发帧



接收缓冲区



确认帧

数据帧



北京大学

停一等式ARQ vs. 确认帧出错

解决方案

- 数据帧用序号0或1表示
- 确认帧用确认号0或1表示

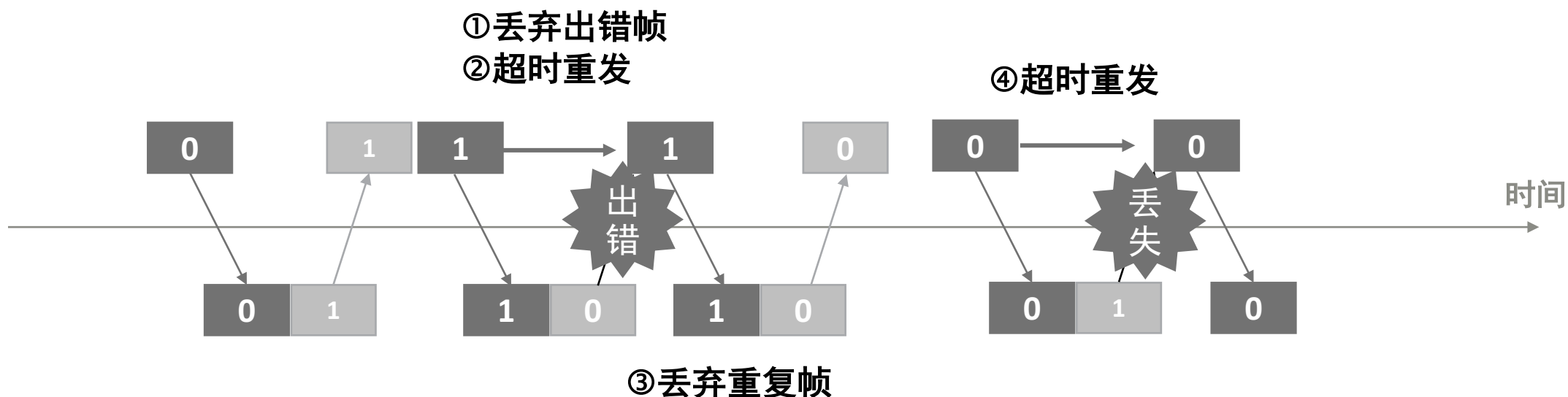
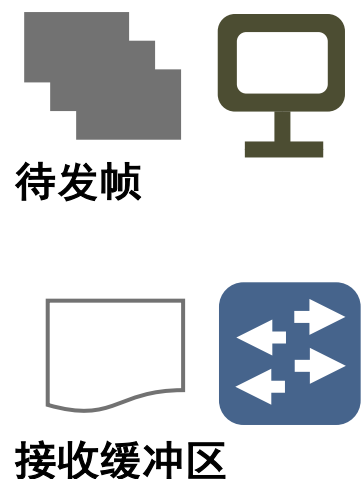
实现简单
效率低下！

ACK被破坏

- 出错确认被丢弃，发送方超时重发
- 接收方根据序号丢弃重复帧

ACK丢失

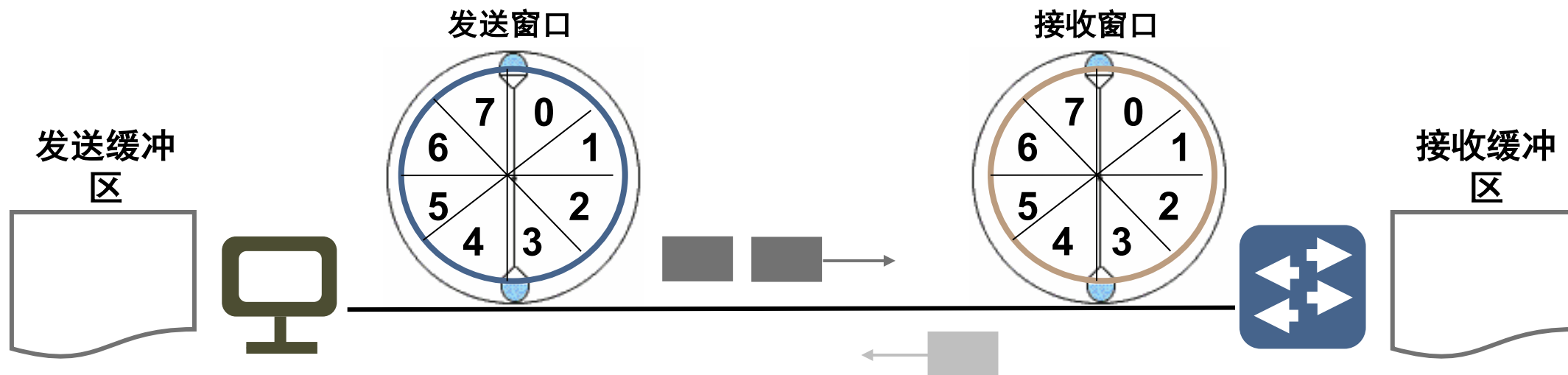
- 发送方超时重发
- 接收方根据序号丢弃重复帧



基于滑动窗口机制的 差错控制



基于滑动窗口机制的差错控制



发送方

- 连续发送由发送窗口指定的多个帧
- 每发出一个帧启动一个计时器
- 在计时器超时后仍未收到确认则重发该帧

接收方

- 每收到一个数据帧检查序号是否落在接收窗口
- 检验该帧传输是否出错
- 如果传输无误且允许接收则给发送方反馈一个肯定确认

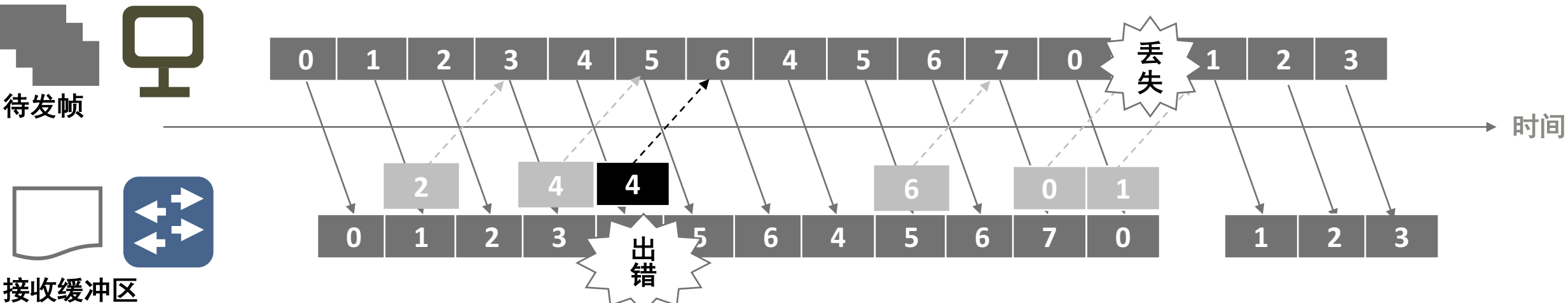


回退-N ARQ基本思想

顺序收发方式：接收方只能按照帧的序号接收数据帧。

回退N控制策略

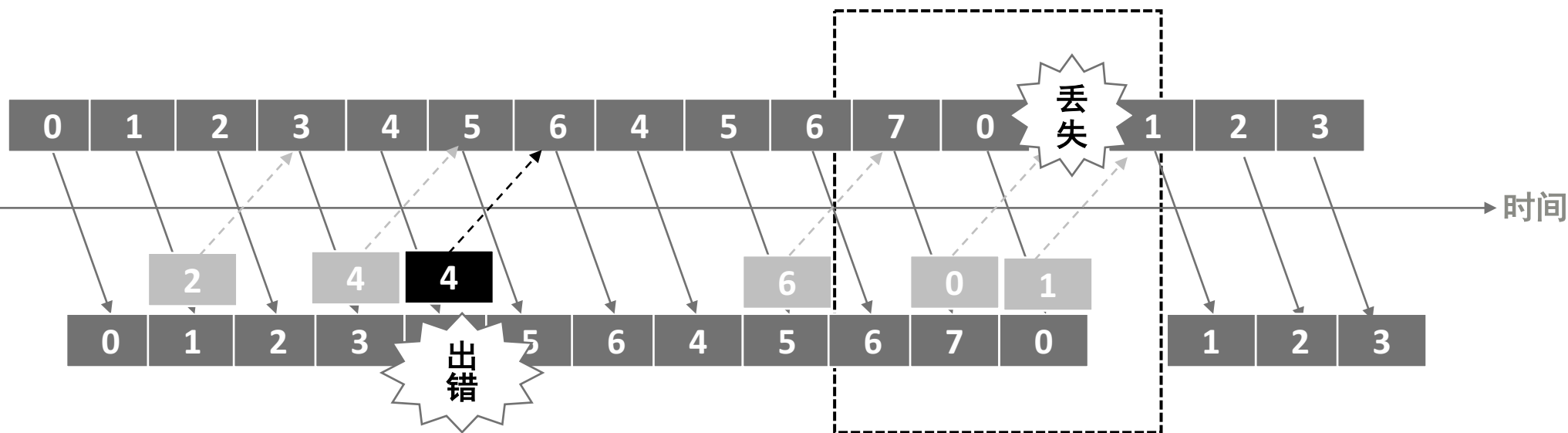
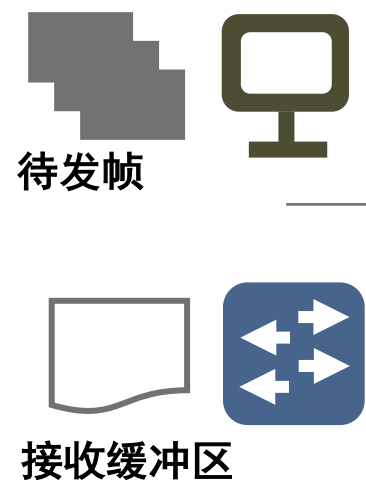
- 发送方连续发出N个帧，接收方以流水线方式顺序接收各个帧，并进行差错检测。
- 一旦某个帧有错，则丢弃该帧和它之后所收到的所有帧。



- ①接收方丢弃出错帧4之后的所有帧
- ②发送方重发出错帧4之后的所有帧
- ③后续确认可弥补丢失的确认

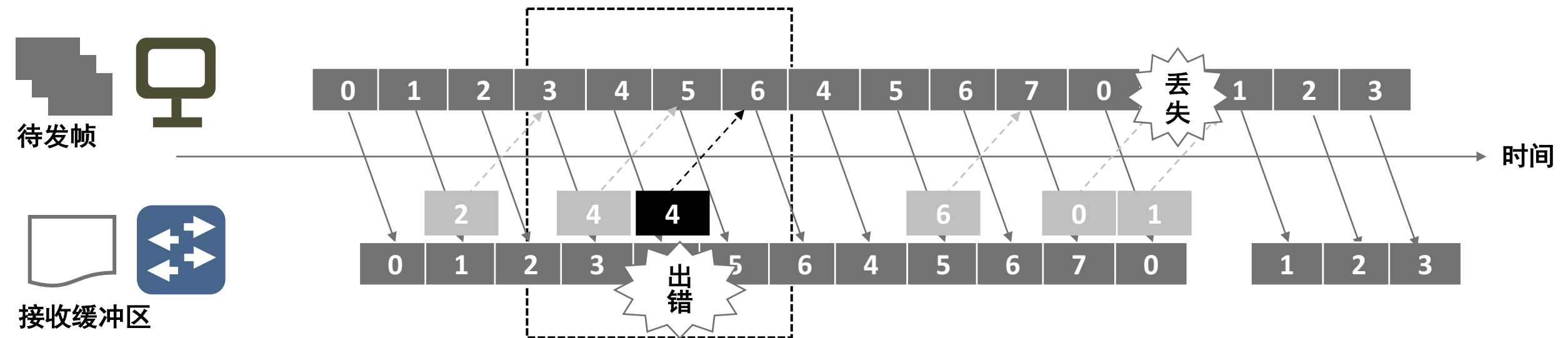
回退-N中的累计确认

1 肯定确认表示已经正确接收直到0的所有帧，
因而可弥补丢失的肯定确认 0



回退-N ARQ中的否定确认

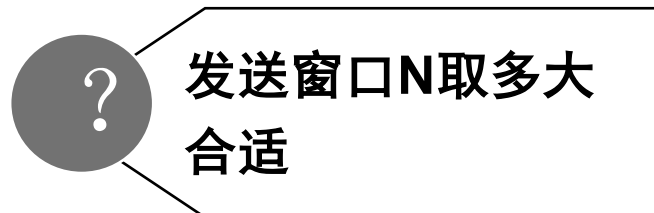
- 4 • 肯定确认表示已经正确接收直到3的所有帧，期待接收帧4
- 4 • 否定确认表示帧4出错，要求重发帧4



回退-N ARQ的发送窗口大小

假设：序号用 n 位表示

- 序号空间：0, 1, 2, 3, $2^n - 1$
- 模 $m = 2^n$
- 最大序号 $S_{\max} = m - 1 = 2^n - 1$



回退N-ARQ控制，最大发送窗口

$$N = 2^n - 1$$

- 窗口越大，发送方在接收方确认帧返回之前可发送的数据帧越多
- 接收方必须分配更多的资源和更大的缓冲区来应付入境数据帧



回退-N ARQ特性

优点

消除了停一等ARQ的等待确认时间。

缺点

正确帧的重发无疑浪费了信道。

特点

- 要求每一帧的确认在其后第N个帧尚未结束发送之前到达
- 发送方必须有存放N帧的缓存以便出错时重发
- 接收方只要求存放一帧大小的缓冲区
- 要求全双工链路

若第一帧的确认在第N帧发完以前尚未到达，表明信道往返时延较大，可加大帧的长度或增加N的值，否则发送方要空等。

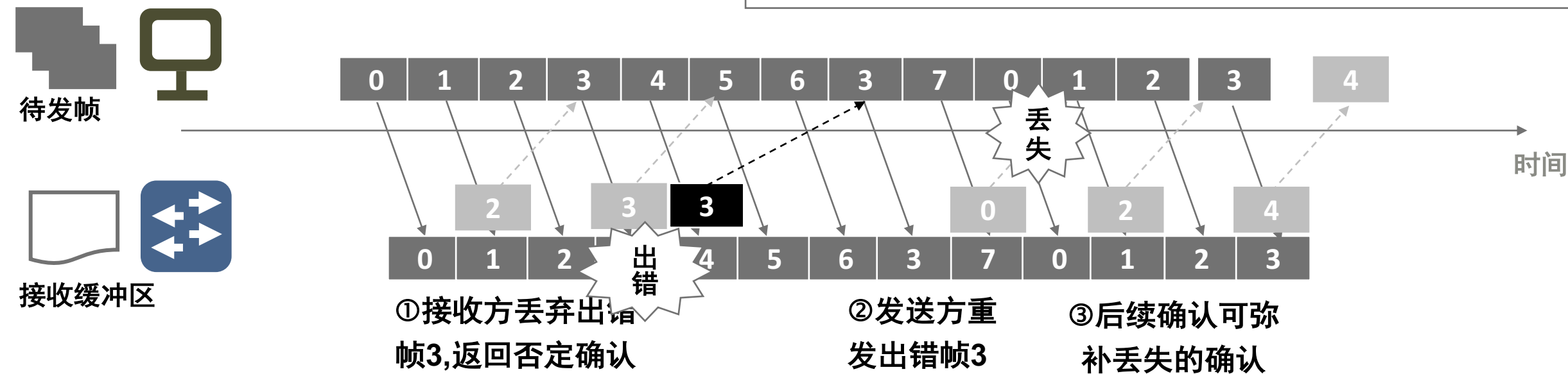


选择重传ARQ基本思想

选择重传ARQ可接收乱序帧，故被称为乱序收发方式。

选择重传控制策略

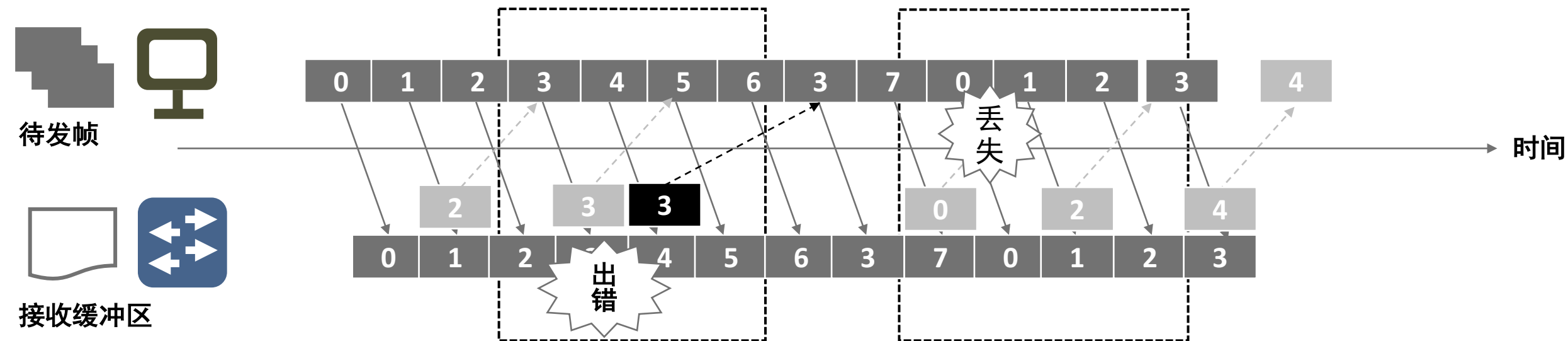
- 发送方可连续发送N个帧，接收方收到出错帧时给发送方反馈出错帧序号，要求发送方只重发出错的帧；
- 出错帧后续到达的帧被保存，待重发后的出错帧到达目的地，接收方把接收到的帧重新排序上交给网络层。



选择重传ARQ中的累计和否定确认

3 发送方针对否定确认采取的措施与回退N不同，发送方仅发送出错的帧3。

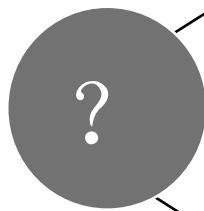
2 肯定确认表示已经正确接收直到1的所有帧，因而可弥补丢失的肯定确认 0



选择重传ARQ的滑动窗口大小

假设：序号用 n 位表示

- 序号空间：0, 1, 2, 3, $2^n - 1$
- 模 $m = 2^n$
- 最大序号 $S_{\max} = m - 1 = 2^n - 1$



选择重传的发送窗口
可以和回退N的发送
窗口一样

选择重传ARQ控制，最大发送窗口

$$N = 2^{n-1}$$

- 发送方要能处理所有可能的出错情况
(错1个帧、2个帧、。。。、 N 个帧)
- 重传时要确保接收方不会发生接收错误



选择重传ARQ特性

优点

- 消除了发送方的等待确认时候
- 信道利用率高

缺点

- 实现复杂
- 接收方要有存放N个帧的缓冲区

特点

- 发送方必须有存放N帧数据的缓存
- 接收方必须有足够存储空间以便缓存(N-1)个帧
- 接收方的接收顺序可能会打乱原发送顺序
- 要求全双工的工作链路

若第一帧的确认在第N帧发完以前尚未到达，表明信道往返时延较大，回退N可加大帧的长度或增加N的值，但选择重传只能加大帧长而不能增加N，否则接收方会发送接收错误。



两种ARQ控制的比较

回退N 的ARQ控制

- 因只能接收正确顺序的数据帧，相对控制简单
- 对接收方的缓存空间要求小，只要有存放一个数据帧的大小即可，
- 要重传出错帧后续的所有数据帧，浪费网络带宽，

以时间和带宽为代价

选择重传的ARQ控制

- 接收方的控制因乱序接收数据帧而复杂
- 对接收方的缓存空间要求大，必须具备和发送方一样大的缓冲区，
- 只重传出错的帧，相对节省网络带宽，

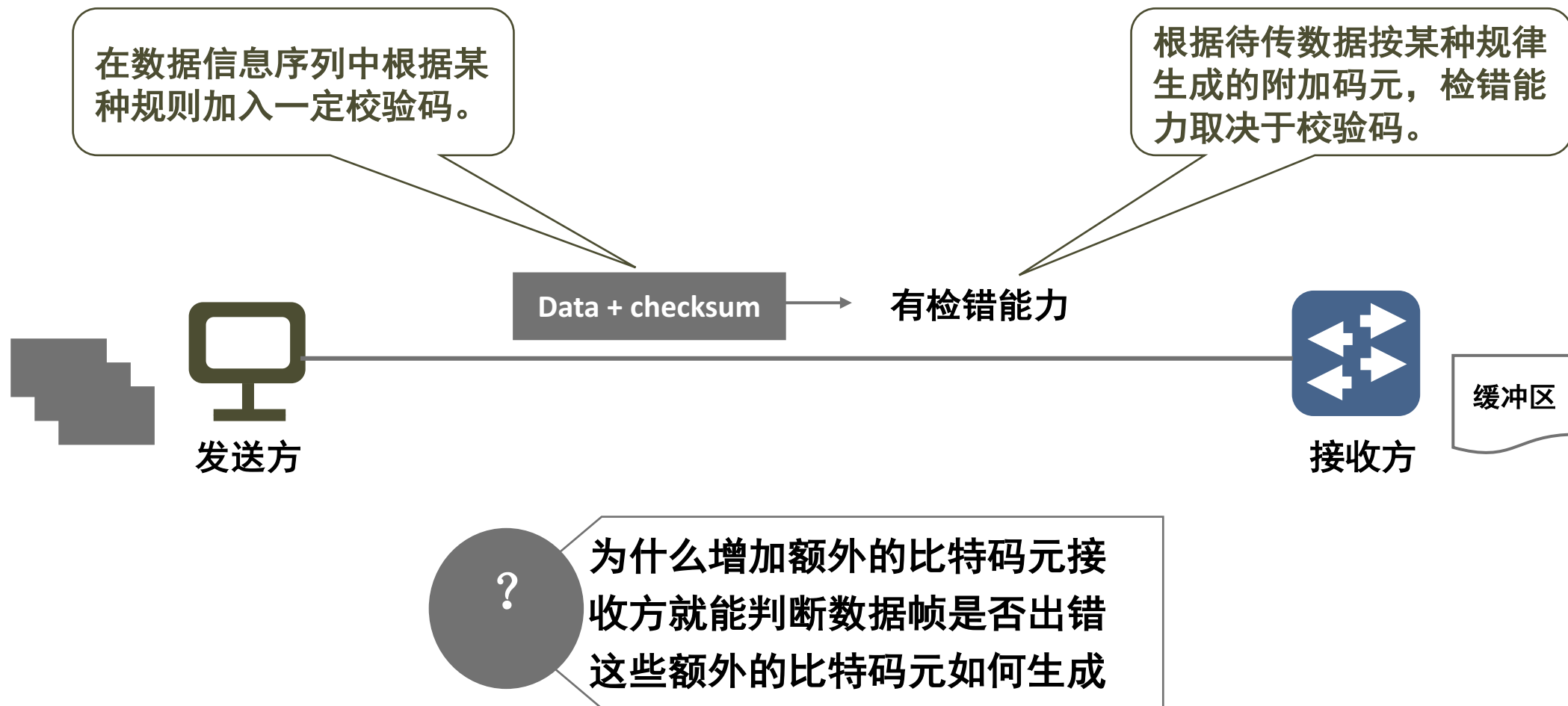
以空间和复杂性为代价



差错检测原理



自动重发检错 (ARQ) 基础



检错纠错原理(上)

假设：待传送的数据信息为“0”和“1”

- A为发“0”时的实际发送的码组
- B为发“1”时的实际发送的码组

- 码组：由n个码元(0,1)构成的每一组合
- 信息码：代表报文的0和1
- 校验码：插入的“0”和“1”

① 如果：不加任何校验码



发送方

待发送数据

0

1

校验码

实际发送数据

0

1

A: 0 B: 1

••既没有检错也纠错能力



北京大学

检错纠错原理(中)

② 如果：加入1个校验码

- 信息码：{0, 1}
- 校验码：{0, 1}
- 准用码组：{00, 11}
- 禁用码组：{01, 10}

当传输00, 11时，如果发生一位错码，则变成“01”或“10”，接收端的译码器将判决有错。



发送方

待发送数据

{0}

{1}

校验码

{0}

{1}

实际发送数据

00

11

A: 00 B: 11

- 具有了检出一位错码的能力
- 没有纠错能力



北京大学

检错纠错原理 (下)

③ 如果：加入2个校验码

- 信息码：代表信息的0和1;
- 校验码：插入的“00”和“11”;
- 准用码组={000, 111}
- 禁用码组={001,010,100,011,101,110}

- 当传输“000”，“111”时，如果发生两位错或一位错，结果都 \in 禁用码组，译码器将判决有错
- 如果只有一位错码，结果变成{001,010,100}或{110,101,011}，译码器可判决是哪位出错并加以纠正



发送方

待发送数据

{0}

{1}

校验码

{00}

{11}

实际发送数据

000

111

A: 000 B: 111

- 具有检出两位及两位以下错码的能力
- 具有纠正一位错码的能力



北京大学

汉明距离

码距：两个码组对应码位码元不同的个数。

- 码组1=(000)与(010) 的码距为1
- 码组2=(000)与(111) 的码距为3

汉明距离：一个码组集合中任意两个码组间的最小码距。

- 码组3={000, 010, 011, 100, 101, 110, 111}的汉明距离为1
- 码组4={000, 111}的汉明距离为3

一般来说，校验码引入越多，检错纠错能力越强，但信道的传输效率下降也越快。



差错检测的编码关系式

假设 编码集合的汉明距离是 d , e 和 t 均表示错码个数

检验目的\编码关系	要求编码集合汉明距离
检出 e 个错码	$d \geq e + 1$
纠正 t 个错码	$d \geq 2t + 1$
检出 e 个错码, 并纠正 t 个错码	$d \geq e + t + 1 \quad (e > t)$



纠错编码示例

示例1 给定某个至多出现2位错的传输信道，如果要具备检出2个错码，同时能纠正1个错码的能力。试问编码所用的码集汉明距离应该为多少？

$$e = 2, t = 1, \text{ 则 } d \geq e + t + 1 = 4$$

A: 0000 B: 1111

准用码组: {0000, 1111}

汉明距离 $d = 4$

能检2个错码，同时能纠1个错码

- 如果接收方收到的码组落在下列集合中 {0011, 0101, 0110, 1001, 1010, 1100}

肯定能检2位错

- 如果接收方收到的码组落在下列集合中 {0001, 0010, 0100, 1000, 1110, 1011, 1101, 0111}

肯定能纠1位错



纠错编码示例

示例2 给定某个至多出现2位错的传输信道，如果A编码为000，B编码为111。试问这种编码方法的检错能力和纠错能力分别为多少？

准用码组为 {000, 111}，汉明距离 $d = 3$

根据 $d \geq e+1$ ，解得 $e = 2$ ，能检出2个错码

根据 $d \geq 2t+1$ ，解得 $t = 1$ ，能纠正1个错码

d 不满足 $\geq e+t+1 = 4$

\therefore 不能同时既检出2个错码又能纠正1个错码



纠错编码的效率

编码效率：指一个码组中信息所占的比重，用R表示(这是衡量编码性能的一个重要参数)。

$$R = \frac{k}{n} = \frac{k}{k+r}$$

- k为信息码长度
- r为校验码长度
- n为编码后码组总长

示例1的编码效率=1/3

- 检出2位错，或纠正1位错

示例2的编码效率=1/4

- 检出2位错同时纠正1位错

特点

- 编码效率↑，R↑，但r↓；
- 检错纠错能力↑，r↑，但R↓



奇偶差错检测



奇偶校验编码

编码规则：先将所要传送的数据码元分组。在各组的数据后面附加一位校验位，使得该组码连校验位在内的码字中

“1” 的个数为偶数—偶校验

“1” 的个数为奇数—奇校验

例如 数据信息k为1101110

- 偶校验的校验码 $r = 1$

发送的码组n : 1101110 ①

- 奇校验的校验码 $r = 0$

发送的码组n : 1101110 ②

垂直
奇偶

水平
奇偶

垂直
水平

斜奇
偶

检错能力逐渐加强



垂直奇偶校验编码

垂直奇偶校验码基本原理

- 发送方在k位表示字符的信息位上附加一个第k+1位的校验码。
- 接收方根据收到的k位重新产生校验码，并与第k+1位作比较，相同则无错，否则存在错误。

设 $b_1 b_2 \dots b_{m-1}$ 是同一码组内的数据码元， b_m 为校验位

$$\text{偶校验: } b_1 \oplus b_2 \oplus \dots \oplus b_{m-1} \oplus b_m = 0$$

$$b_m = b_1 \oplus b_2 \oplus \dots \oplus b_{m-1}$$

$$\text{奇校验: } b_1 \oplus b_2 \oplus \dots \oplus b_{m-1} \oplus b_m = 1$$

$$b_m = b_1 \oplus b_2 \oplus \dots \oplus b_{m-1} \oplus 1$$



ISO 7码元垂直奇偶校验码

示例：假设发送方要发送“HELLO”，采用奇偶校验方法。

试问：实际发送的码组？

ISO编码标准

H - 1001000

E - 1000101

L - 1001100

O - 1010001

编码	b ₁	b ₂	b ₃	b ₄	b ₅	b ₆	b ₇	偶校验	奇校验
H	0	0	0	1	0	0	1	0	1
E	1	0	1	0	0	0	1	1	0
L	0	0	1	1	0	0	1	1	0
L	0	0	1	1	0	0	1	1	0
O	1	0	0	0	1	0	1	1	0

H的偶校验码 b₈ 的值应使得

$$b_1 \oplus b_2 \oplus \dots \oplus b_7 \oplus b_8 = 0 \rightarrow b_8 = 0$$

H的奇校验码 b₈ 的值应使得

$$b_1 \oplus b_2 \oplus \dots \oplus b_7 \oplus b_8 = 1 \rightarrow b_8 = 1$$



垂直奇偶校验码的检错能力与编码效率

编码	b_1	b_2	b_3	b_4	b_5	b_6	b_7	偶校验	奇校验
H	0	0	0	1	0	0	1	0	1
E	1	0	1	0	0	0	1	1	0
L	0	0	1	1	0	0	1	1	0
L	0	0	1	1	0	0	1	1	0
O	1	0	0	0	1	0	1	1	0

检错能力

- 能发现奇数个差错
- 无法发现偶数个差错

编码效率

$$R = k / (k+1)$$

K是数据信息位数

奇偶校验特点

- 实现简单
- 对随机错误的检测非常有效



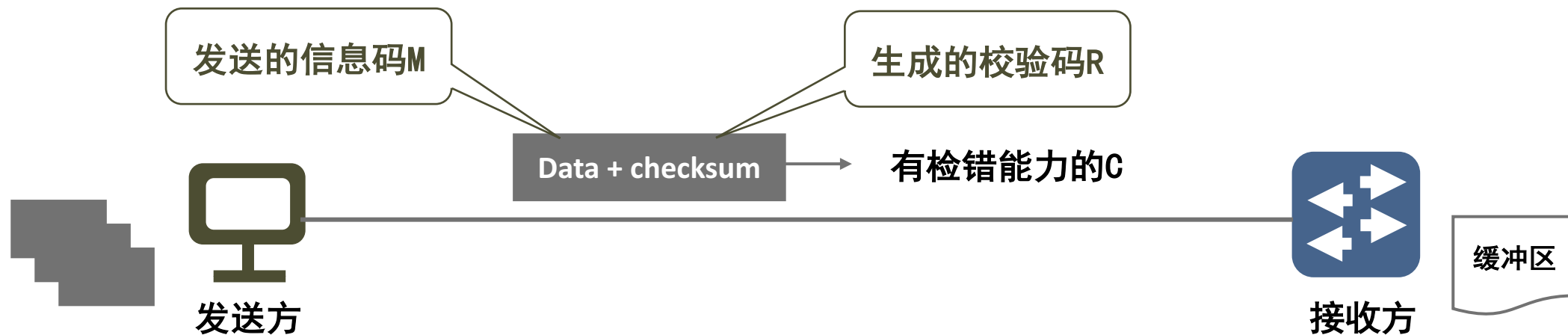
循环冗余差错校验



循环冗余校验码 (CRC)

基本思想

- 发送方将要传送的信息分成码组M，按某种约定的规律对每一个信息码组附加一些校验码R，形成新的码C，使得C中的码元之间具有一定的相关性，再传输到接收方；
- 接收方根据这种相关性或规律性来校验码C是否正确，还可对出错码组的错定位加以相应的纠正，最后将码C还原成信息码M。



码多项式及其算术运算

假设 码 $C = C_{n-1} C_{n-2} \dots C_2 C_1 C_0$ 长度为 n

C 的码多项式（称为 $n-1$ 次多项式）

$$C(x) = C_{n-1} x^{n-1} + C_{n-2} x^{n-2} + \dots + C_2 x^2 + C_1 x + C_0$$

例如： $C = 1100101$

$$C(x) = x^6 + x^5 + x^2 + 1$$

码多项式算术运算

- 模2加
- 模2减
- 模2乘
- 模2除

例如： $1101011011.0000 \div 10011$

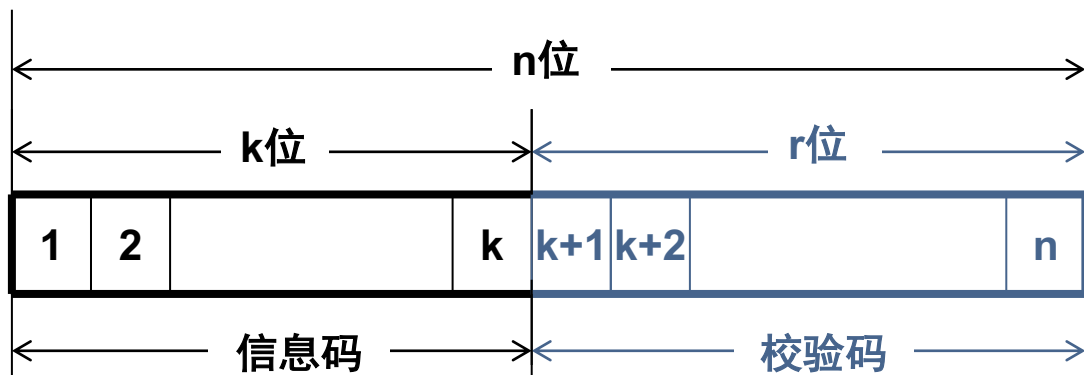
商数： 1100001010

余数： 1110



循环码与循环码生成多项式

对于一个码长为 n ，信息码为 k 位的循环码 (n, k) ，其构成形式为：



在一个 (n, k) 循环码中，存在一个且只有一个 $(n-k)$ 次的码多项式

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \dots + g_2x^2 + g_1x + 1$$

- 此循环码中任一码多项式都是 $g(x)$ 的倍式
- 任意一个 $(n-1)$ 次或 $(n-1)$ 次以下又是 $g(x)$ 倍式的多项式必定是此循环码的一个码多项式

循环码 (n, k) 用多项式表示：

$$C(x) = C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \dots + C_{n-k}x^{n-k} + C_{n-k-1}x^{n-k-1} \dots + C_2x^2 + C_1x + C_0$$

生成多项式: $(n-k)$ 次且最高次系数非0的码多项式 $g(x)$.



校验码的产生

- 设生成多项式 $G(x)$ 的最高幂次为 $r=n-k$
- 将待编码码元序列表示为 $m(x)$ ，乘以 x^r ，结果左移 r 位 $x^r \cdot m(x)$
- 用 $G(x)$ 去除 $x^r \cdot m(x)$ ，求得商式 $Q(x)$ 和余式 $R(x)$

$$\text{CRC-12} = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

$$\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$$

为除法运算后变成的循环
多项式 $C(x)$ 的表示式

$$[x^r \cdot m(x)] / G(x) = Q(x) + R(x) / G(x)$$

$$x^r \cdot m(x) = Q(x) \cdot G(x) + R(x)$$

$$x^r \cdot m(x) + R(x) = Q(x) \cdot G(x)$$

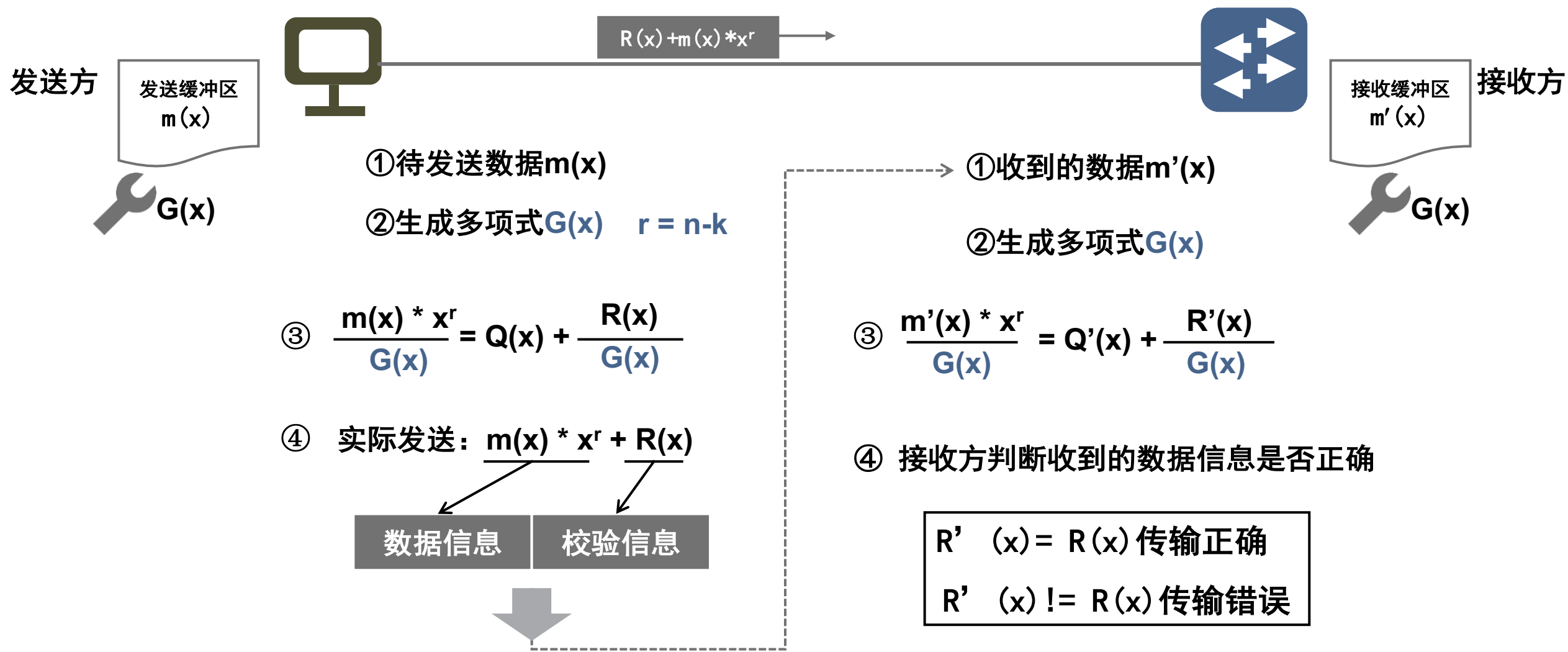
$R(x)$ 校验多项式。

其系数 $C_{r-1}, C_{r-2}, \dots, C_1, C_0$ 为校验码

$$\begin{aligned} C(x) &= C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \dots + C_{n-k}x^{n-k} + C_{n-k-1}x^{n-k-1} + \dots + C_2x^2 + C_1x + C_0 \\ &= \underbrace{C_{n-1}x^{n-1} + C_{n-2}x^{n-2} + \dots + C_{n-k}x^{n-k}}_{\text{信息码}} + \underbrace{C_{r-1}x^{r-1} + C_{r-2}x^{r-2} + \dots + C_2x^2 + C_1x + C_0}_{\text{校验码}} \end{aligned}$$



循环码校验工作过程



校验和计算实例

示例1：设编码的信息码元为1101011011.试问给定生成多项式 $g(x)$ 下发送的码组？

$$m(x) = x^9 + x^8 + x^6 + x^4 + x^3 + x + 1, \quad k = 10$$

① 假设 $g(x) = x^4 + x + 1$

系数形成的位串为10011 $r = 4 \rightarrow n = 14$

$r(x)$ 的最高幂次为 $r-1=3$

② $x^4 \cdot m(x) = 1101011011,0000$

③ $1101011011.0000 \div 10011$



商数：1100001010

余数：1110

$$r(x) = x^3 + x^2 + x + 0$$

④所需的循环编码 $C(x)$ 为

$$C(x) = x^r \cdot m(x) + r(x) = 1101011011, 1110$$



多项式计算实例

示例2：多项式除法

$1101011011.0000 \div 10011$

$$\begin{array}{r} \begin{array}{c} 10011 \\ \uparrow \\ \text{除数 } g(x) \end{array} \quad \begin{array}{r} 1100001010 \leftarrow \text{商数} \\ \hline 1101011011,0000 \leftarrow \text{被除数 } m(x) \\ \underline{10011} \\ 10011 \\ \underline{10011} \\ 10110 \\ \underline{10011} \\ 10100 \\ \underline{10011} \\ 1110 \leftarrow \text{余数 } r(x) \end{array} \end{array}$$

模2加=模2减

$$0+0 = 0; \quad 0+1 = 1+0 = 1; \quad 1+1 = 0;$$

$$0-0 = 0; \quad 0-1 = 1-0 = 1; \quad 1-1 = 0;$$

模2乘

$$0*0 = 0*1 = 1*0 = 0; \quad 1*1 = 1$$

模2除：模2乘的可逆运算



北京大学

循环校验码的检错能力

❑CRC循环校验码的检错能力

- 检查全部单个错（1位错）
- 检查全部离散的二位错（双错）
- 检查全部的奇数个错（1, 3, 5...个错）
- 检查全部长度等于 $(n-k)$ 或小于 $(n-k)$ 的突发错
- 以 $[1-2^{-(r+1)}]$ 的概率检查出长度为 $(r+1)$ 的突发错以及以 $[1-2^{-r}]$ 的概率检查出多于 $(r+1)$ 的突发错



链路层基本功能小结

链路层主要功能

- 组成帧(结合协议介绍)
- 流量控制
- 差错控制
- 共享介质访问控制

控制机制的本质

- 任何控制都有代价
- 流量/差错控制代价
 - ① 肯定/否定确认
 - ② 额外的校验码
 - ③ 缓冲区
 - ④ 计算负担

传输性能的提高

- 滑动窗口
- 累计确认
- 捎带确认
- 否定确认
- 肯定确认
- 计时器(超时重发)

