

第四章 数据链路层基础

授课教师：崔勇

清华大学



计 算 机 网 络
教 案 社 区

致谢社区成员

安徽大学 王贵竹	华北科技学院 陈振国
华东交通大学 王艳	福州大学 张浩
中国传媒大学 林卫国	枣庄学院 徐涛
华南理工大学 王昊翔	南开大学 徐敬东



回顾与思考



清华大学
Tsinghua University



计算机网络教案社区

➤ 数据链路层的目标

- 实现点到点的可靠通信
- 有什么可以借鉴的设计思路?

➤ 人们互相说话会遇到哪些问题?

- 能互相听到了 (物理层已经通畅)
- 啥? 没听清 (单个字)? 有口音?
- 说得太快? 咱俩抢着说?
- 如何解决?

➤ 数据链路层要解决哪些问题?

- 在传输中会出错吗? 如何检测出错误?
- 是尝试纠正还是要求重传?
- 数据过多, 超出接收方的处理能力?
-



➤ 物理层 (Physical Layer)

- 定义如何在信道上传输0、1 : Bits on the wire



本节目标



清华大学
Tsinghua University



计算机网络教案社区

- 了解数据链路层在网络体系结构中的位置及基本功能和服务
- 掌握差错检测和纠正的基本原理和典型的编码方法（核心内容）
- 掌握无错信道和有错信道上停等协议的设计和实现方法（核心内容）



本节内容



4.1 数据链路层的设计问题

4.2 差错检测和纠正

4.3 基本的数据链路层协议

1. 数据链路层在协议栈中的位置
2. 数据链路层的功能
3. 数据链路层提供的服务
4. 成帧
5. 差错控制
6. 流量控制



数据链路层在协议栈中的位置

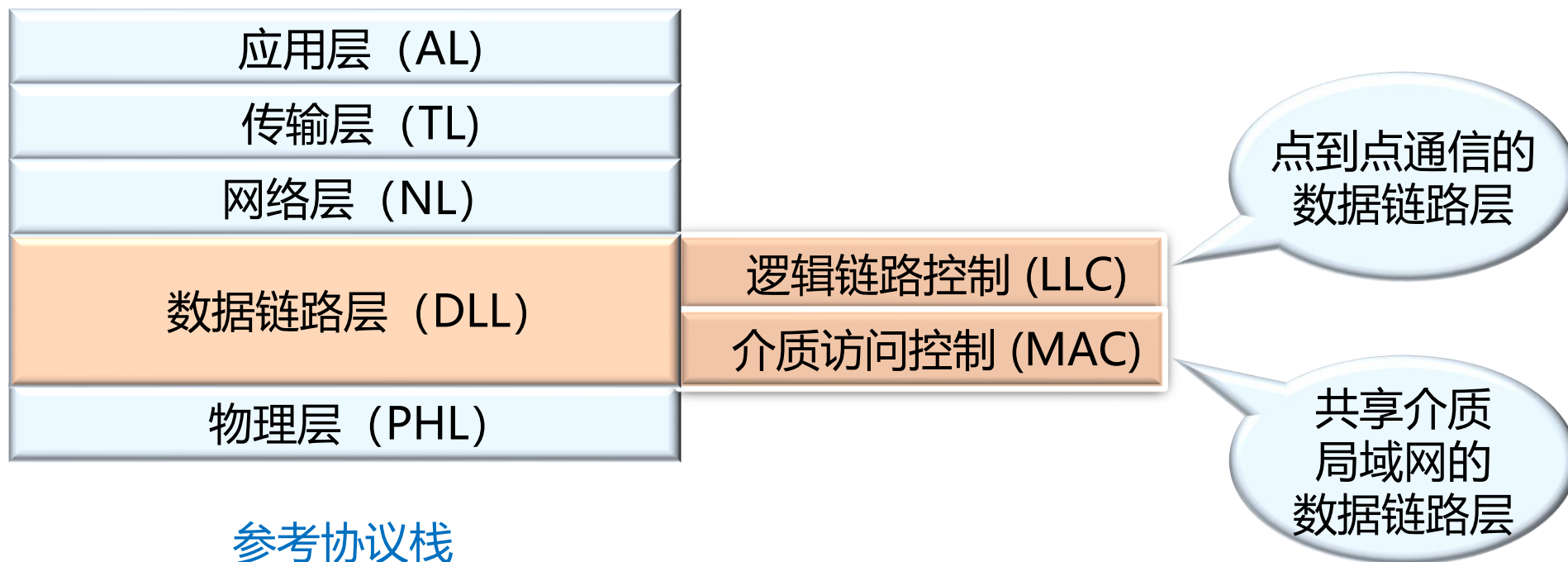


清华大学
Tsinghua University



计算机网络教案社区

- 向下：利用物理层提供的位流服务
- 向上：向网络层提供明确的 (well-defined) 服务接口





物理层、数据链路层和网络层的实现



清华大学
Tsinghua University



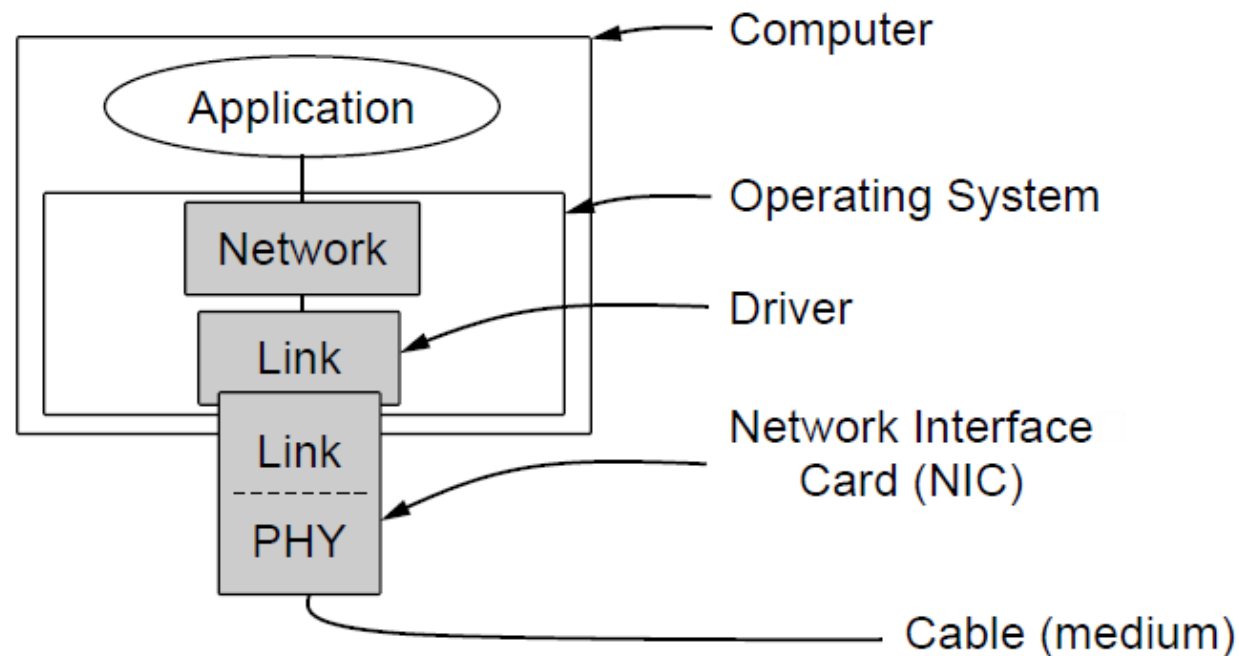
计算机网络教案社区

➤ NIC

- NIC, Network Interface Card
- 物理层进程和某些数据链路层进程, 运行在专用硬件上

➤ 协议栈与驱动

- 数据链路层进程的其他部分和网络层进程, 作为操作系统的一部分 (协议栈), 运行在CPU上
- 数据链路层进程的软件通常以设备驱动的形式存在





关键假设



清华大学
Tsinghua University



计算机网络教案社区

➤ 分层进程独立假设

- 网络层、数据链路层、物理层为独立的进程
- 进程间通过传递消息实现通信

➤ 提供可靠服务假设

- 提供可靠的、面向连接的服务
- 数据链路层要发送的数据可从网络层获得

➤ 只处理通信错误假设

- 仅处理通信错误
- 假设机器不会崩溃，不考虑断电、重启等引起的问题



数据链路层的功能



清华大学
Tsinghua University



计算机网络教案社区

➤ 成帧 (Framing)

- 将比特流划分成 “帧” 的主要目的是为了检测和纠正物理层在比特传输中可能出现的错误，数据链路层功能需借助 “帧” 的各个域来实现

➤ 差错控制 (Error Control)

- 处理传输中出现的差错，如位错误、丢失等

➤ 流量控制 (Flow Control)

- 确保发送方的发送速率，不大于接收方的处理速率
 - 避免接收缓冲区溢出



数据链路层提供的服务



清华大学
Tsinghua University



计算机网络教案社区

- 无确认 无连接 服务 (Unacknowledged connectionless)
 - 接收方不对收到的帧进行确认
 - 适用场景：误码率低的可靠信道；实时通信；
 - 网络实例：以太网
- 有确认 无连接 服务 (Acknowledged connectionless)
 - 每一帧都得到单独的确认
 - 适用场景：不可靠的信道（无线信道）
 - 网络实例：802.11
- 有确认 有连接 服务 (Acknowledged connection-oriented)
 - 适用场景：长延迟的不可靠信道



成帧 (Framing)



清华大学
Tsinghua University



计算机网络教案社区

- 关键问题：从源源不绝的比特流，到帧？
 - 接收方必须能从物理层接收的比特流中明确区分出一帧的开始和结束
 - 这个问题被称为帧同步或帧定界
- 成帧 (framing) 的方式
 - 字节计数法 (Byte count)
 - 带字节填充的定界符法 (Flag bytes with byte stuffing)
 - 带比特填充的定界符法 (Flag bits with bit stuffing)
 - 物理层编码违例 (Physical layer coding violations)



成帧 (Framing)

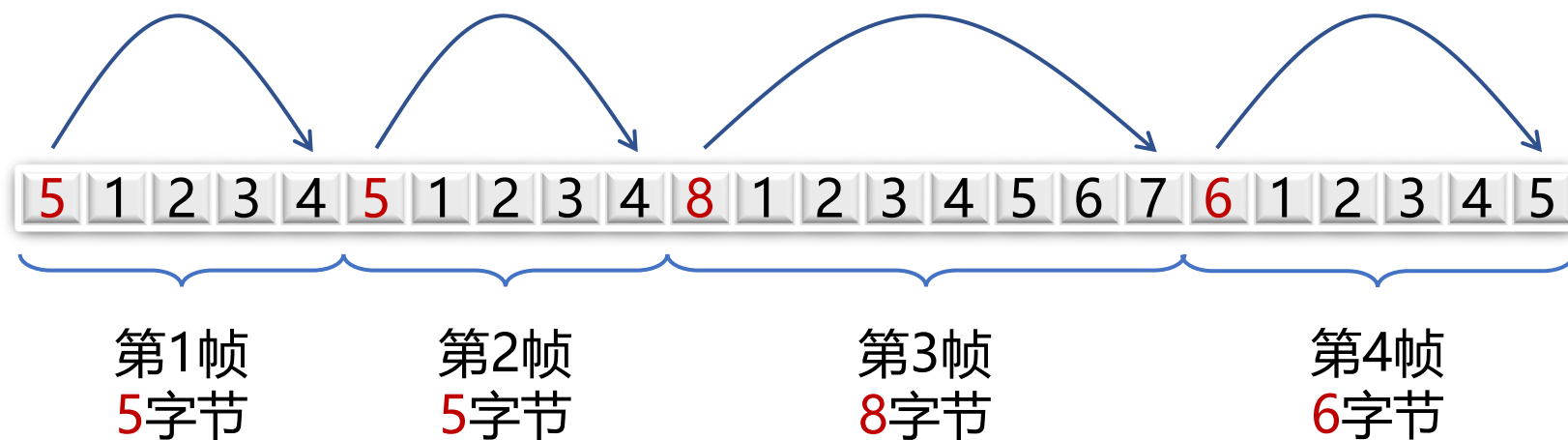


清华大学
Tsinghua University



计算机网络教案社区

➤ 字节计数法：无差错传输的情形



问题：如果某个计数字节出错会发生什么情况？



成帧 (Framing)

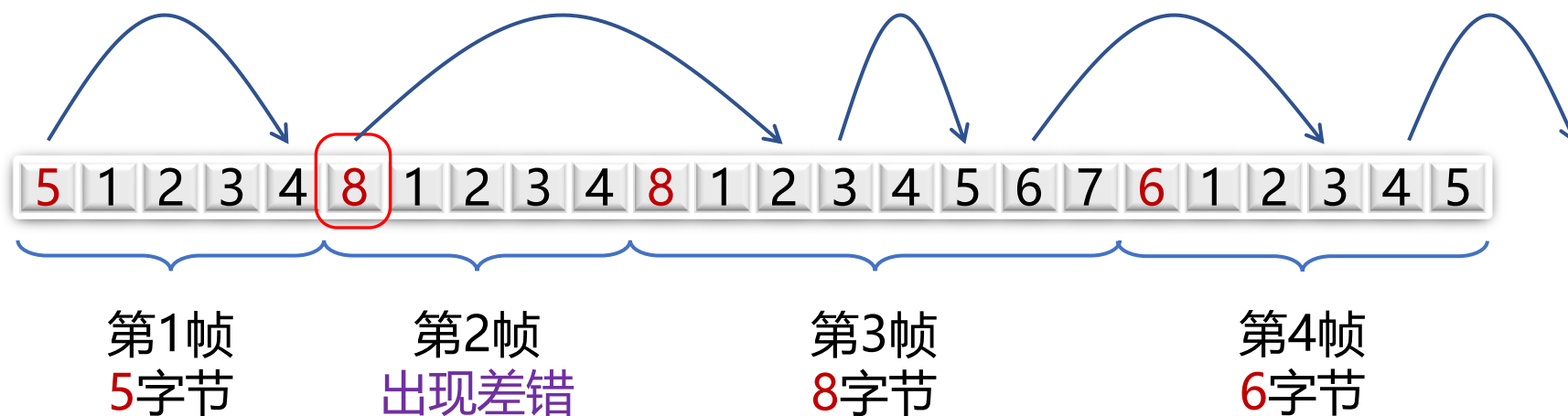


清华大学
Tsinghua University



计算机网络教案社区

➤ 字节计数法：出现了一个字节**差错**的情形



破坏了帧的边界，导致一连串帧的错误！怎么办？

透过现象看本质，连问三个为什么



成帧 (Framing)



清华大学
Tsinghua University



计算机网络教案社区

➤ 带字节填充的定界符法

- 定界符 (FLAG)：一个特殊的字节，比如 01111110，即 0x7E，用于区分前后两个不同的帧



问题：如果有效载荷部分包含与“定界符”相同的字节会有什么问题？



成帧 (Framing)

➤ 带字节填充的定界符法：发送方检查有效载荷，进行字节填充





成帧 (Framing)



清华大学
Tsinghua University



计算机网络教案社区

➤ 带字节填充的定界符法：接收方的处理

- 逐个检查收到的每一个字节
- 收到ESC
 - 则后一字节无条件成为有效载荷，不予检查
- 收到FLAG
 - 则为帧的边界



成帧 (Framing)



清华大学
Tsinghua University



计算机网络教案社区

➤ 带字节填充的定界符法：收方的处理示例

接收方收到的数据

处理后的数据

A ESC FLAG B



A FLAG B

A ESC ESC B



A ESC B

A ESC ESC ESC FLAG B



A ESC ESC FLAG B

A ESC ESC ESC ESC B



A ESC ESC B

问题：带字节填充的定界符法不够灵活怎么办？



成帧 (Framing)



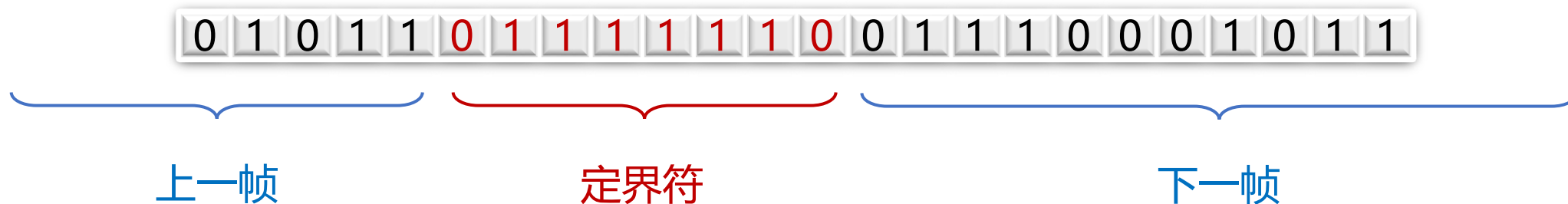
清华大学
Tsinghua University



计算机网络教案社区

➤ 带比特填充的定界符法

- 定界符：两个0比特之间，连续6个1比特，即**01111110**，0x7E



问题：如果有效载荷部分包含与“**定界符**”相同的位组合如何解决？



成帧 (Framing)



清华大学
Tsinghua University



计算机网络教案社区

- **带比特填充的定界符法**：发送方检查有效载荷
 - 若在有效载荷中出现连续5个1比特，则直接插入1个0比特

原始载荷

0 1 1 1 1 1 0 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 0

载荷填充

0 1 1 1 1 1 0 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0 0 0 1 0



成帧 (Framing)



清华大学
Tsinghua University

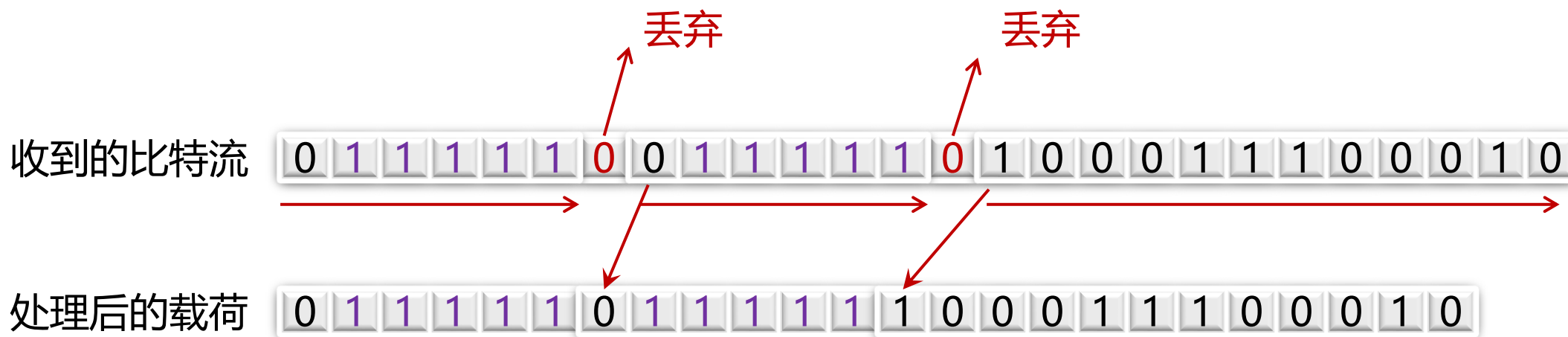


计算机网络教案社区

➤ 带比特填充的定界符法：接收方的处理

- 若出现连续5个1比特

- 若下一比特为0，则为有效载荷，直接丢弃0比特
- 若下一比特为1，则连同后一比特的0，构成定界符，一帧结束





成帧 (Framing)



➤ 物理层编码违例

- 核心思想：选择的定界符不会在数据部分出现
- 4B/5B编码方案
 - 4比特数据映射成5比特编码，剩余的一半码字（16个码字）未使用，可以用做帧定界符
 - 例如：00110组合不包含在4B/5B编码中，可做帧定界符
- 前导码
 - 存在很长的 前导码 (preamble)，可以用作定界符（并同步时钟）
 - 例如：传统以太网、802.11
- 曼切斯特编码 / 差分曼切斯特编码
 - 正常的信号在周期中间有跳变，持续的高电平（或低电平）为违例码，可以用作定界符
 - 例如：802.5令牌环网



成帧 (Framing)

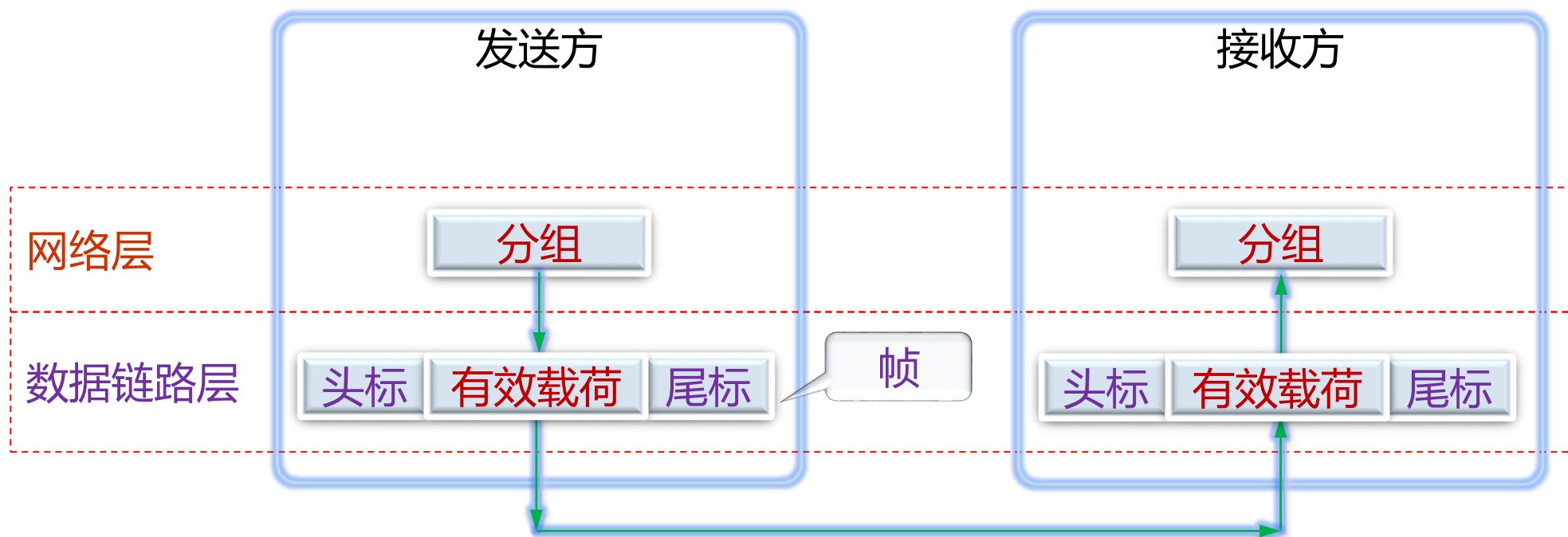


清华大学
Tsinghua University



计算机网络教案社区

➤ 分组 (packet) 与 帧(frame)的关系





小结：数据链路层的设计问题



清华大学
Tsinghua University



计算机网络教案社区

➤ 数据链路层的功能

- 成帧，差错控制，流量控制

➤ 帧的定界方法

- 字节计数法
- 带字节填充的定界符法：借助转义字节 ESC 和定界符 FLAG
- 带比特填充的定界符法：借助连续出现的 1 的个数



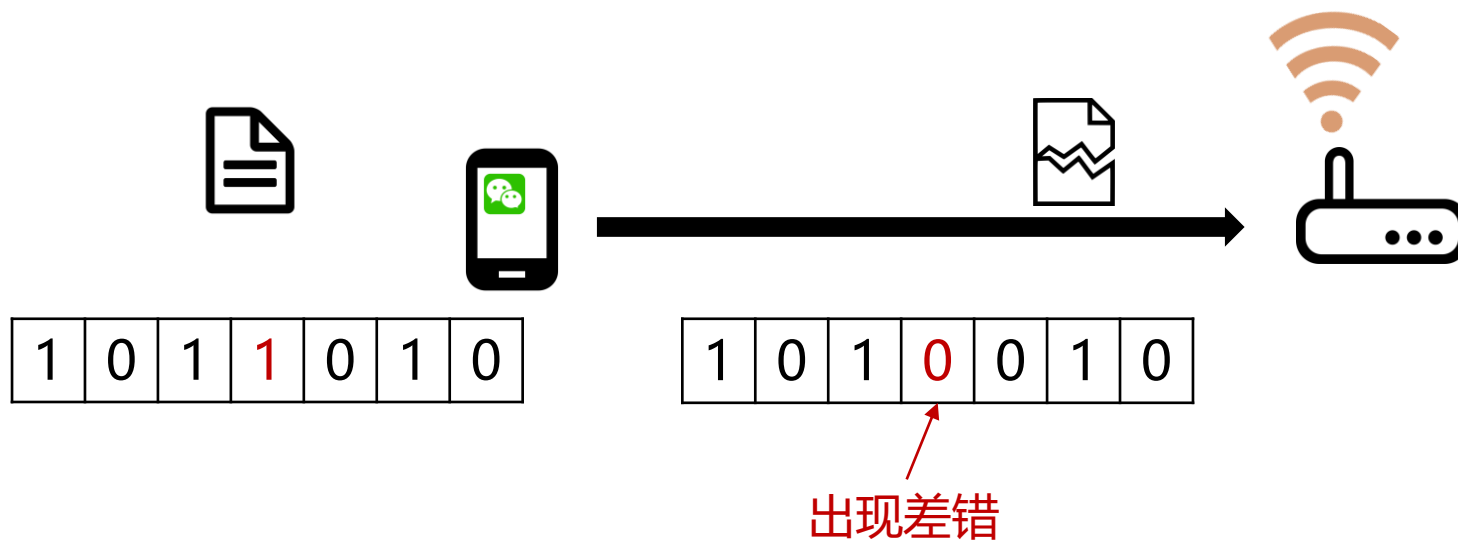
思考



清华大学
Tsinghua University



计算机网络教案社区



传输过程中出现差错，怎么办？

如何能检测发现错误？是否能够纠正错误？



本节内容



清华大学
Tsinghua University



计算机网络教案社区

4.1 数据链路层的设计问题

4.2 差错检测和纠正

4.3 基本的数据链路层协议

1. 差错检测与纠正概述
2. 典型的纠错码
3. 典型的检错码



差错控制



清华大学
Tsinghua University



计算机网络教案社区

- 大胆思考，都会有什么样的差错？不仅仅是1比特错误
 - 差错（incorrect）：数据发生错误
 - 丢失（lost）：接收方未收到
 - 乱序（out of order）：先发后到，后发先到
 - 重复（repeatedly delivery）：一次发送，多次接收
- 解决方案：差错检测与纠正、确认重传
 - 编码：检错与纠错编码
 - 确认：接收方校验数据（差错校验），并给发送方应答，防止差错
 - 定时器：发送方启动定时器，防止丢失
 - 顺序号：接收方检查序号，防止乱序递交、重复递交



差错检测和纠正概述



➤ 如何解决信道传输差错问题

- 通常采用**增加冗余信息**（或称**校验信息**）的策略
- 简单示例：每个比特传三份，如果每比特的三份中有一位出错，可以纠正

0	1	0	0	1	0	1	0	1	0	1	1	0	1
0	1	0	0	1	0	1	0	1	0	1	1	0	1
0	1	0	0	1	0	1	0	1	0	1	1	0	1

原始数据

冗余信息

33.3%

66.7%

设计具体编码前
来点理论基础？

携带2/3的冗余信息，冗余太多怎么办？



差错检测和纠正概述



清华大学
Tsinghua University



计算机网络教案社区

➤ 码字 (code word)

- 一个 $n=m+r$ 位单元 (含 m 个数据位和 r 个校验位) , 描述为 (n, m) 码

➤ 码率 (code rate)

- 码字中不含冗余部分所占的比例, 可以用 m/n 表示

➤ 海明距离 (Hamming distance)

- 两个码字之间对应不同比特的数目
- 如果两个码字的海明距离为 d , 则需要 d 个单比特错误就可以把一个码字转换成另一个码字
- 海明距离为 **d** 的编码, 可以**检测 $d-1$** 个单比特错误
- 海明距离为 **d** 的编码, 可以**纠正 $(d-1)/2$** 个单比特错误

码字A: 00000

码字B: 11111

A和B的海明距离为5

收到码字

0000**1**

000**11**

00**111**

0**1111**

如果是3位
错误呢?



典型检错码



清华大学
Tsinghua University



计算机网络教案社区

➤ 检错码 (error-detecting code)

- 在被发送的数据块中，包含一些冗余信息，但这些信息只能使接收方推断是否发生错误，但不能推断哪位发生错误，接收方可以请求发送方重传数据
- 主要用在高可靠、误码率较低的信道上，例如光纤链路
- 偶尔发生的差错，可以通过重传解决差错问题

➤ 常用检错码

- **奇偶检验** (Parity Check): 1位奇偶校验是最简单、最基础的检错码
- **校验和** (Checksum): 主要用于TCP/IP体系中的网络层和传输层
- **循环冗余校验** (Cyclic Redundancy Check, CRC): 数据链路层广泛使用的校验方法



典型检错码—奇偶校验



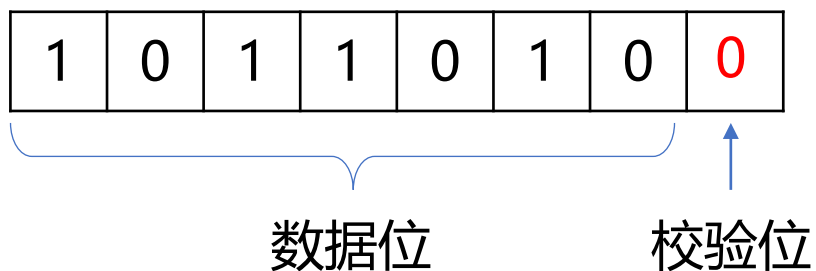
清华大学
Tsinghua University



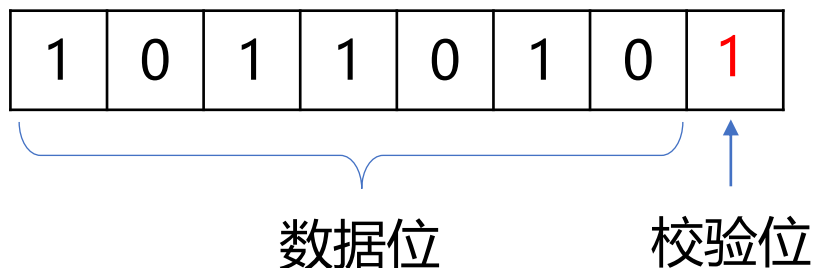
计算机网络教案社区

➤ 1位奇偶校验：增加1位校验位，可以检查奇数位错误

- 偶校验：保证1的个数为偶数个



- 奇校验：保证1的个数为奇数个



奇偶校验思路简单，
但效率低下！怎么办？

1	0	1	1	0	1	0	0
1	0	1	0	0	1	0	1
1	0	1	1	0	1	0	0
1	0	1	0	0	1	0	1

会有什么问题？
连续突发错误？



典型检错码—校验和



清华大学
Tsinghua University



计算机网络教案社区

➤ 以TCP/IP体系中主要采用的校验方法为例

发送方：进行 16 位二进制补码求和运算，计算结果取反，随数据一同发送

接收方：进行 16 位二进制补码求和运算（包含校验和）。若结果非全1，则检测到错误

数据

1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

数据

补码求和

1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0

1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1
1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0

同样计算
(补码求和)

校验和
(取反)

0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

校验和

优缺点？ 错1位成功检测到，错2位呢？

未检测到错误



典型检错码—循环冗余校验CRC



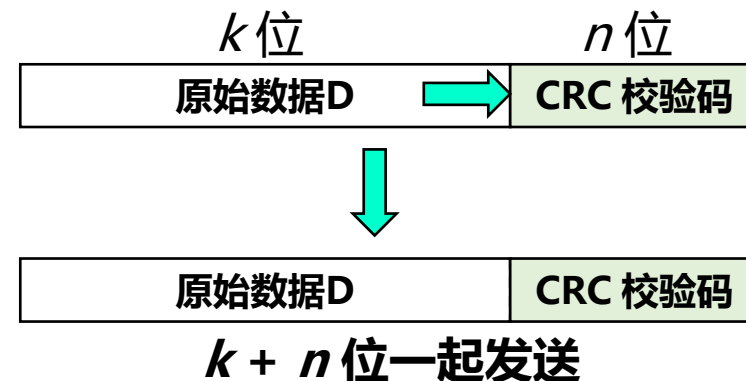
清华大学
Tsinghua University



计算机网络教案社区

➤ CRC校验码计算方法

- 设原始数据D为k位二进制 (如D = 1010001101)
- **生成多项式G**: 如果要产生n位CRC校验码, 事先选定一个n+1位二进制G (收发双方提前商定)
- G最高位为1 (如G=110101, 即 $G=x^5+x^4+x^2+1$)
- 将原始数据D乘以 2^n (如101000110100000), 产生k+n位二进制, 用G对该数做模2除, 得到余数R即为CRC校验码 (n位, 不足n位前面用0补齐)



接收端如何计算?



检错码—循环冗余校验CRC



清华大学
Tsinghua University

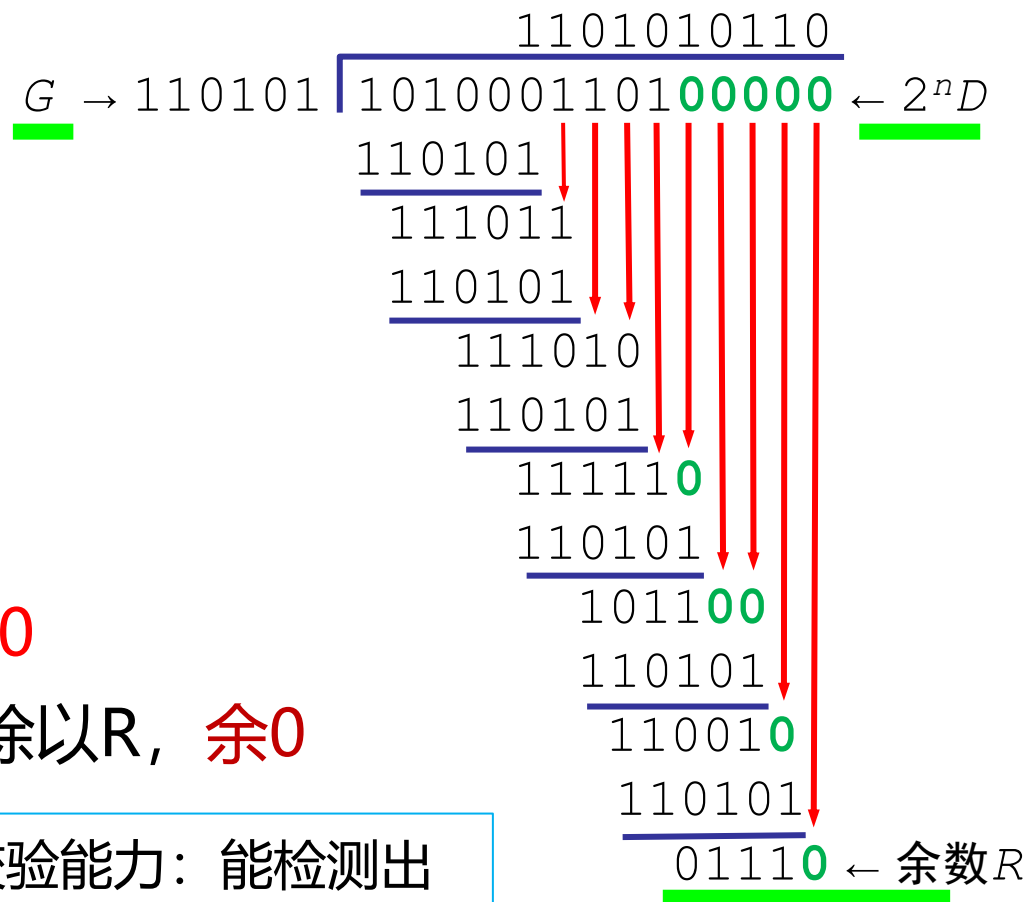


计算机网络教案社区

➤ CRC校验码计算示例

- 数据: $D = 1010001101$
- $n = 5$
- 生成多项式: $G = 110101$,
即 $G = x^5 + x^4 + x^2 + 1$
- 余数: $R = 01110$
- 发送数据: $T = 101000110101110$
- 接收方: $T = 101000110101110$ 除以 R , 余0
- 有差错情况: $(T+E)/R=0?$

CRC校验能力: 能检测出
少于 $n+1$ 位的突发错误





检错码—循环冗余校验CRC



清华大学
Tsinghua University



计算机网络教案社区

➤ 四个国际标准生成多项式

- $\text{CRC-12} = x^{12} + x^{11} + x^3 + x^2 + x + 1$
- $\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$
- $\text{CRC-CCITT} = x^{16} + x^{12} + x^5 + 1$
- $\text{CRC-32} = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

例如，以太网、无线局域网使用CRC-32生成多项式



典型纠错码



清华大学
Tsinghua University



计算机网络教案社区

- 纠错码 (error-correcting code)
 - 主要用于错误发生比较频繁的信道上，如无线链路
 - 也经常用于物理层，以及更高层（例如，实时流媒体应用和内容分发）
 - 使用纠错码的技术通常称为**前向纠错** FEC (Forward Error Correction)
 - 发送方在每个数据块中加入足够的冗余信息，使得接收方能够判断接收到的数据是否有错，并能纠正错误（**定位出错的位置**）

考虑二维奇偶校验，
是否可以进行纠错？



以二维偶校验为例：

1	0	1	0	1		1
1	1	1	1	0		0
0	1	1	1	0		1
0	0	1	0	1		0

无错情况

1	0	1	0	1		1
1	0	1	1	0		0
0	1	1	1	0		1
0	0	1	0	1		0

有错情况



典型纠错码—海明码



清华大学
Tsinghua University



计算机网络教案社区

➤ 纠错码设计目标

- 找到出错位置，从而提供纠错能力
- 如何缩小并定位错误的范围？
- 每个校验位对数据位的子集做校验？
- 问题：如何设置和计算校验位？

➤ 海明码编码过程

- 校验位：2的幂次方位（记为p1, p2, p4, p8）
- 例：11比特的数据01011001101按顺序放入数据位
- 数据位k对哪些校验位有影响？
将k写成2的幂的和，例：11 = 1 + 2 + 8

			0
	1	2	3
	1	0	1
4	5	6	7
	1	0	0
8	9	10	11
1	1	0	1
12	13	14	15

校验位 数据位

以 (15, 11)海明码为例
右下角数字为码字中位序号

校验位：1、2、4、8

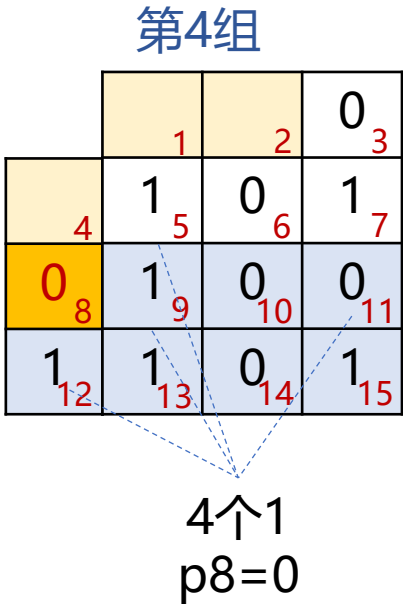
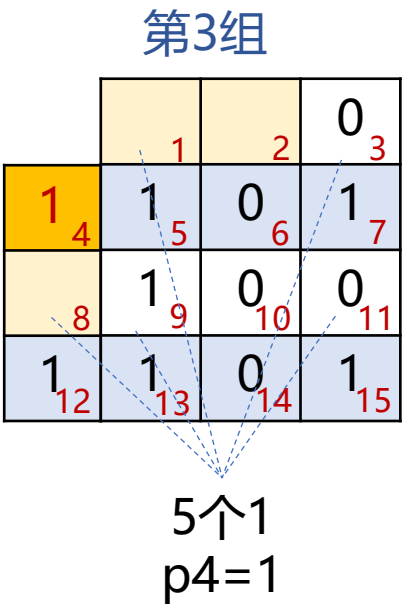
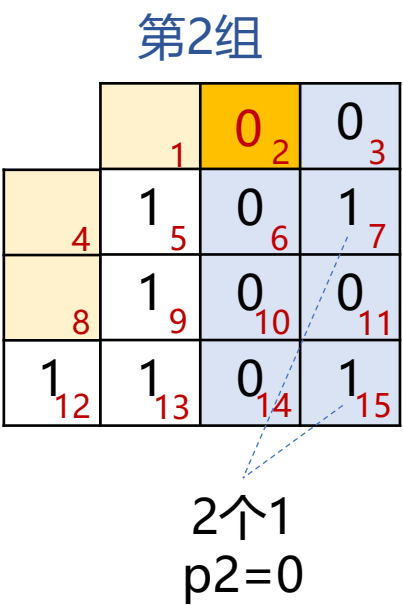
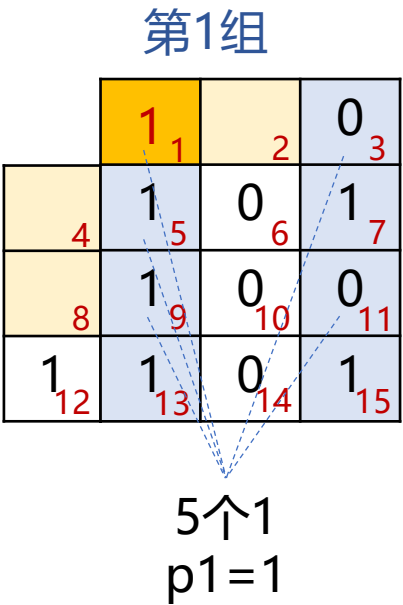
1	2	3	4	5	6	7	8	9	10	11...
		1		1		1		1		1
		2			2	2			2	2
				4	4	4				
								8	8	8



典型纠错码—海明码

- 子集的选择与校验位计算
- 海明码缺省为偶校验（也可以使用奇校验）

1	2	3	4	5	6	7	8	9	10	11...
		1		1		1		1		1
		2			2	2			2	2
				4	4	4				
								8	8	8



每组的数据位 每组的校验位



典型纠错码—海明码



清华大学
Tsinghua University



计算机网络教案社区

➤ 码字的发送与接收

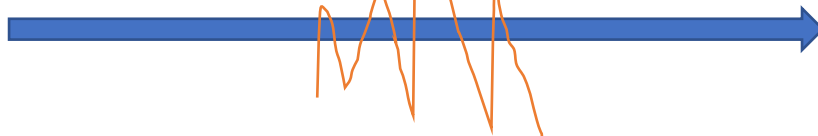
- 如果发送过程中第7位出现差错，如何定位错误？

	1 ₁	0 ₂	0 ₃
1 ₄	1 ₅	0 ₆	1 ₇
0 ₈	1 ₉	0 ₁₀	0 ₁₁
1 ₁₂	1 ₁₃	0 ₁₄	1 ₁₅

校验位 数据位

发送方

第7位出现差错



	1 ₁	0 ₂	0 ₃
1 ₄	1 ₅	0 ₆	0 ₇
0 ₈	1 ₉	0 ₁₀	0 ₁₁
1 ₁₂	1 ₁₃	0 ₁₄	1 ₁₅

校验位 数据位

接收方



典型纠错码—海明码



清华大学
Tsinghua University



计算机网络教案社区

➤ 定位错误与纠正

- 组1和组2的校验结果，**定位错误位于第4列**
- 组3校验结果指明存在错误，组4校验结果指明无错误，**可定位错误位于第2行**
- 结论：定位第2行、第4列（第7比特错误）

1	2	3	4	5	6	7	8	9	10	11...
		1		1		1		1		1
		2			2	2			2	2
				4	4	4				
							8	8	8	

第1组

	1 ₁		0 ₃
4	1 ₅	0 ₆	0 ₇
8	1 ₉	0 ₁₀	0 ₁₁
1 ₁₂	1 ₁₃	0 ₁₄	1 ₁₅

5个1
有错误

第2组

		0 ₂	0 ₃
4	1 ₅	0 ₆	0 ₇
8	1 ₉	0 ₁₀	0 ₁₁
1 ₁₂	1 ₁₃	0 ₁₄	1 ₁₅

1个1
有错误

第3组

			0 ₃
1 ₄	1 ₅	0 ₆	0 ₇
8	1 ₉	0 ₁₀	0 ₁₁
1 ₁₂	1 ₁₃	0 ₁₄	1 ₁₅

5个1
有错误

第4组

			0 ₃
4	1 ₅	0 ₆	0 ₇
0 ₈	1 ₉	0 ₁₀	0 ₁₁
1 ₁₂	1 ₁₃	0 ₁₄	1 ₁₅

4个1
无错误



典型纠错码—海明码

- 定位错误与纠正 (续)
- 总体的定位错误列表

1	2	3	4	5	6	7	8	9	10	11
		1		1		1		1		1
		2			2	2			2	2
				4	4	4				
							8	8	8	

出错位	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
组1-1	×	√	×	√	×	√	×	√	×	√	×	√	×	√	×
组2-2	√	×	×	√	√	×	×	√	√	×	×	√	√	×	×
组3-4	√	√	√	×	×	×	×	√	√	√	√	×	×	×	×
组4-8	√	√	√	√	√	√	√	×	×	×	×	×	×	×	×

用海明码如何纠错?



典型纠错码—海明码



清华大学
Tsinghua University



计算机网络教案社区

➤ 海明码纠正的实现过程

- 每个码字到来前，接收方计数器清零
- 接收方检查每个校验位 k ($k = 1, 2, 4 \dots$) 的奇偶值是否正确（每组运算）
- 若 p_k 奇偶值不对，计数器加 k
- 所有校验位检查完后，若计数器值为0，则码字有效；若计数器值为 j ，则第 j 位出错
- 例：若校验位 p_1 、 p_2 、 p_4 出错，则第7位变反

1	2	3	4	5	6	7	8	9	10	11
		1		1		1		1		1
		2			2	2			2	2
				4	4	4				
								8	8	8



5G新标准——编码的力量



清华大学
Tsinghua University



计算机网络教案社区

➤ 华为主导的5G标准：极化码—Polar码

- Polar码最早由土耳其教授E. Arikan在2007年提出
- 华为慧眼识珠，积极发展并应用Polar码
- 5G标准的师徒之争：华为推崇极化码；
美国高通推崇低密度奇偶校验（LDPC）码
- LDPC码：数据信道编码方案；极化码：控制信道编码方案



“国王般的礼遇”

➤ 极化码的基本思路

- 信道极化理论是Polar编码理论的核心，包括信道组合和信道分解部分
- 信道极化过程本质上是一种信道等效变换的过程，当信道的数目趋于无穷大时，出现极化现象：一部分信道将趋于无噪信道，另外一部分则趋于全噪信道
- 无噪信道的传输速率将会达到信道容量 $I(W)$ ，而全噪信道的传输速率趋于零
- Polar码的编码策略正是应用了这种现象的特性，利用无噪信道传输用户有用的信息，全噪信道传输约定的信息或者不传信息



小结：差错检测和纠正



清华大学
Tsinghua University



计算机网络教案社区

➤ 检错和纠错

- 通过增加冗余信息，实现检错码和纠错码
- 海明距离：两个码字之间不同对应比特的数目

➤ 检错码

- 奇偶检验：增加1位校验位，可以检查奇数位错误
- 校验和：如TCP/IP 校验和，使用 16 位二进制补码求和，而后取反
- 循环冗余校验CRC

➤ 纠错码

- 海明码：以奇偶校验为基础，提供1位纠错能力



思考



清华大学
Tsinghua University



计算机网络教案社区

分块处理比特，方便
数据链路层功能的实现



成 帧

- 字节计数法
- 带字节填充的定界符法（借助转义字节 ESC 和定界符 FLAG）
- 带比特填充的定界符法（借助连续出现的 1 的个数）
- 其它

检测和纠正错误



差错控制

- 海明距离
- 检错码：奇偶校验，校验和，循环冗余校验
- 纠错码：海明码

避免数据多发、少收



流量控制



➤ 啥？你说太快？

- 发端到收端，发送速率比处理速率大，数据来不及处理。怎么办？
- 是暂存多出的数据，还是直接丢弃？
- 或是从根本上避免多发的可能性？

➤ 发出的帧丢失怎么办？

- 发端如何知道帧丢失，从而重发？
- 发端如何知道要重发哪个帧？
- 谁丢失了什么？



本节内容



4.1 数据链路层的设计问题

4.2 差错检测和纠正

4.3 基本的数据链路层协议

步步深入，越来越现实

1. 乌托邦式单工协议
2. 无错信道单工停止-等待协议
3. 有错信道单工停止-等待协议

发明互联网协议而不仅仅是理解



基本的协议定义

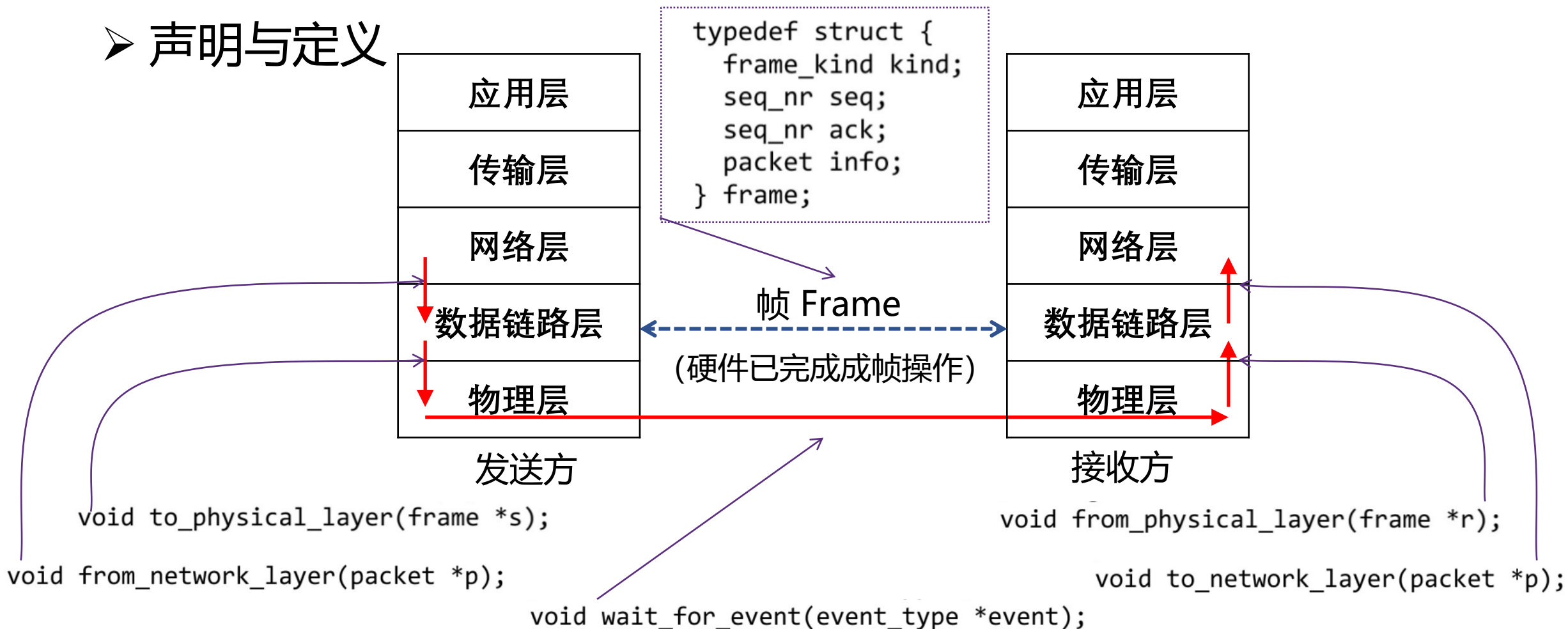


清华大学
Tsinghua University



计算机网络教案社区

➤ 声明与定义



event_type: 帧到达frame_arrival、帧出错cksum_err、超时timeout等



乌托邦式单工协议P1



清华大学
Tsinghua University



计算机网络教案社区

➤ 从理想而简单的假设开始设计

- 单工 (Simplex) 协议：数据单向传输
- 完美信道：帧不会丢失或受损
- 始终就绪：发送方/接收方的网络层始终处于就绪状态
- 瞬间完成：发送方/接收方能够生成/处理无穷多的数据

➤ 乌托邦：完美但不现实的协议

- 不处理任何流量控制或纠错工作
- 接近于无确认的无连接服务，必须依赖更高层次解决上述问题



乌托邦式单工协议P1



➤发送方

- 在循环中不停发送
- 从网络层获得数据
- 封装成帧
- 交给物理层
- 完成一次发送

缺点?
比比看谁快? !

```
frame s;  
packet buffer;  
  
while (true) {  
    from_network_layer(&buffer);  
    s.info = buffer;  
    to_physical_layer(&s);  
}
```

发送程序：取数据，构成帧，发送帧

➤接收方

- 在循环中持续接收
- 等待帧到达 (frame_arrival)
- 从物理层获得帧
- 解封装，将帧中的数据传递给网络层
- 完成一次接收

```
frame r;  
event_type event;  
  
while (true) {  
    wait_for_event(&event);  
    from_physical_layer(&r);  
    to_network_layer(&r.info);  
}
```

接收程序：等待，接收帧，送数据给高层



流量控制



清华大学
Tsinghua University



计算机网络教案社区

➤ 链路层存在的问题：接收方的处理速率

- 发送方发送帧的速度过快（大于接收方处理速度），将导致接收方被“淹没” (overwhelming)
- 基于速率 (rate-based) 的流量控制
 - 发送方根据内建机制，自行限速
- 基于反馈 (feedback-based) 的流量控制
 - 接收方反馈，发送方调整发送速率

如何能让接收方
不被淹没？

➤ 仍然假设

- 通信信道不会出错 (Error-Free)
- 数据传输保持单向, 但是需要双向传输链路 (半双工物理信道)



无错信道上的停等式协议P2



清华大学
Tsinghua University



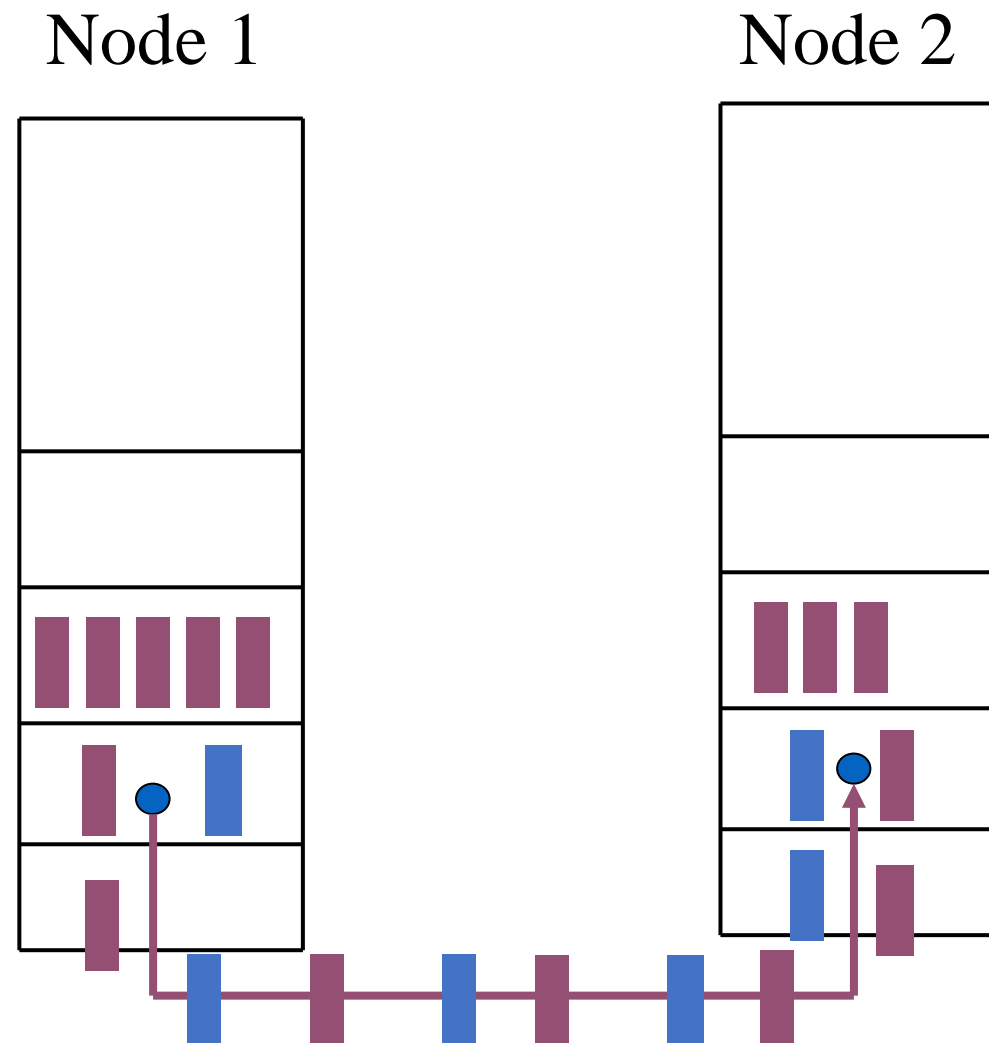
计算机网络教案社区

➤ 停-等式协议 (stop-and-wait)

- 增加**确认**机制: **Acknowledgment**
- 发送方发送一帧后暂停, 等待确认到达后发送下一帧
- 接收方完成接收后, 回复确认接收

➤ 哑帧确认 (dummy frame)

- 确认帧的内容是不重要的





无错信道上的停等式协议P2



清华大学
Tsinghua University



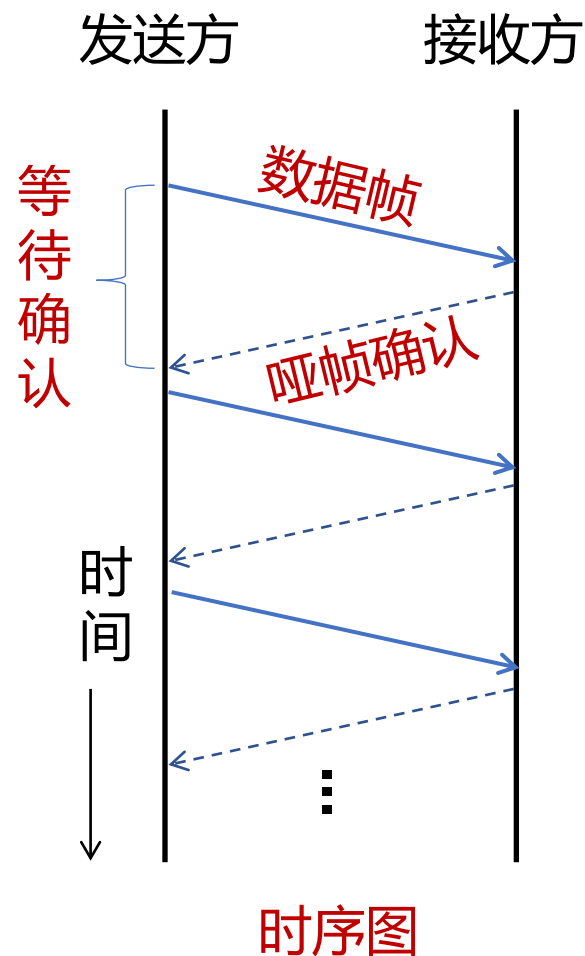
计算机网络教案社区

➤ 停-等式协议 (stop-and-wait)

- 增加**确认**机制: **Acknowledgment**
- 发送方发送一帧后暂停, 等待确认到达后发送下一帧
- 接收方完成接收后, 回复确认接收

➤ 哑帧确认 (dummy frame)

- 确认帧的内容是不重要的





无错信道上的停等式协议P2



清华大学
Tsinghua University



计算机网络教案社区

➤ 发送方

- 完成一帧发送后
- 等待确认到达
- 确认到达后，发送下一帧

```
while (true) {  
    from_network_layer(&buffer);  
    s.info = buffer;  
    to_physical_layer(&s);  
    wait_for_event(&event);  
}
```

发送程序：取数据，组帧，发送帧，等待确认帧

➤ 接收方

- 完成一帧接收后
- 交给物理层一个哑帧
- 作为成功接收上一帧的确认

```
while (true) {  
    wait_for_event(&event);  
    from_physical_layer(&r);  
    to_network_layer(&r.info);  
    to_physical_layer(&s);  
}
```

接收程序：
等待，接收帧，送数据给高层，回送确认帧

发现缺点？通信信道会出错吗？



有错信道上的单工停等式协议P3



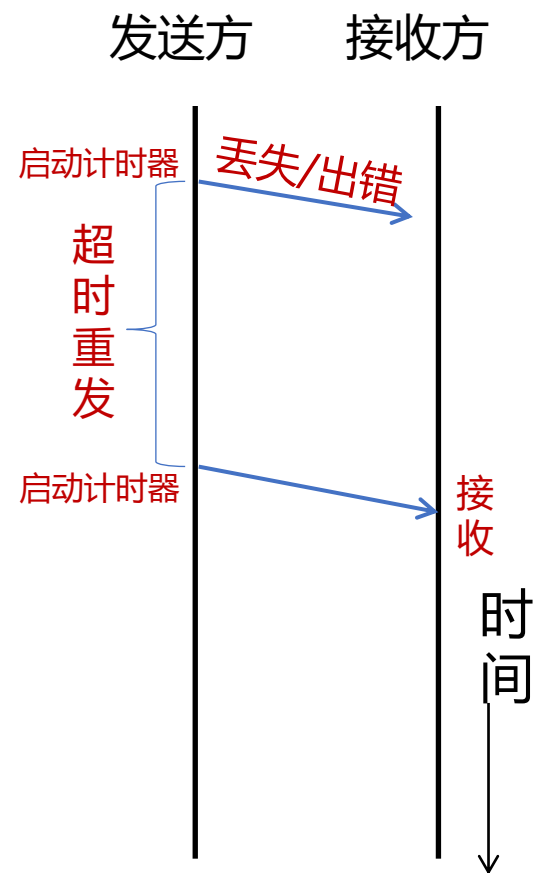
清华大学
Tsinghua University



计算机网络教案社区

- 通信信道出错会导致什么结果?
 - 帧在传输过程中可能会被损坏，接收方能够检测出来
 - 帧在传输过程中可能会丢失，永远不可能到达接收方
- 发明解决方案?
 - 发送方增加一个**计时器(timer)**，如果经过一段时间没有收到确认，发送方将超时，于是再次发送该帧
- 除了数据会丢失，还有其他什么问题？

反向信道出错会有什么结果？





有错信道上的单工停等式协议P3



清华大学
Tsinghua University



计算机网络教案社区

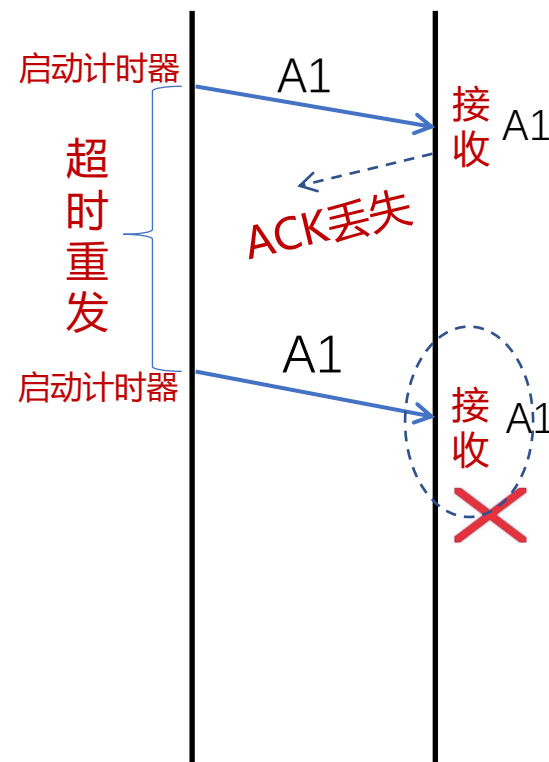
➤ 考虑一个特别场景

- A发送帧A1
- B收到了A1
- B生成确认ACK
- ACK在传输中丢失
- A超时，重发A1
- B收到A1的另一个副本（并把它交给网络层）

➤ 其他的场景

- 另一个导致副本产生的场景是过长的延时 (long delay)

发送方A 接收方B



接收端如何识别重复的报文？



有错信道上的单工停等式协议P3



清华大学
Tsinghua University



计算机网络教案社区

➤ 序号 (SEQ: sequence number)

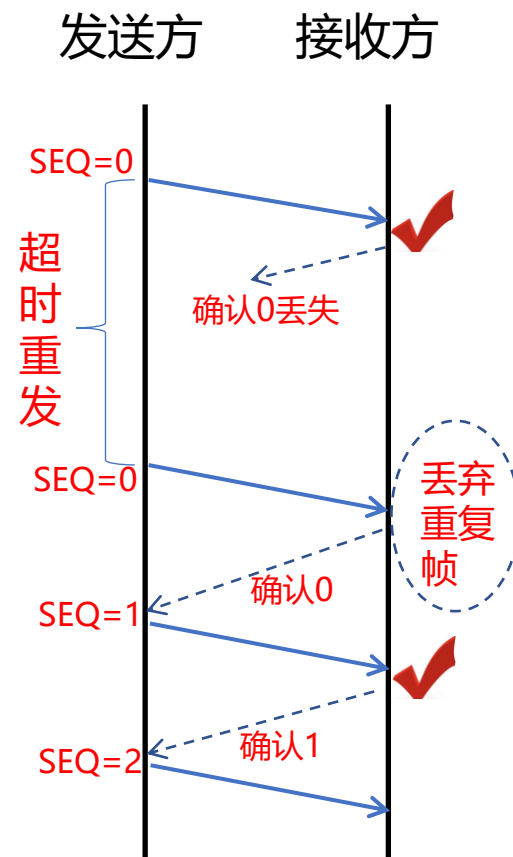
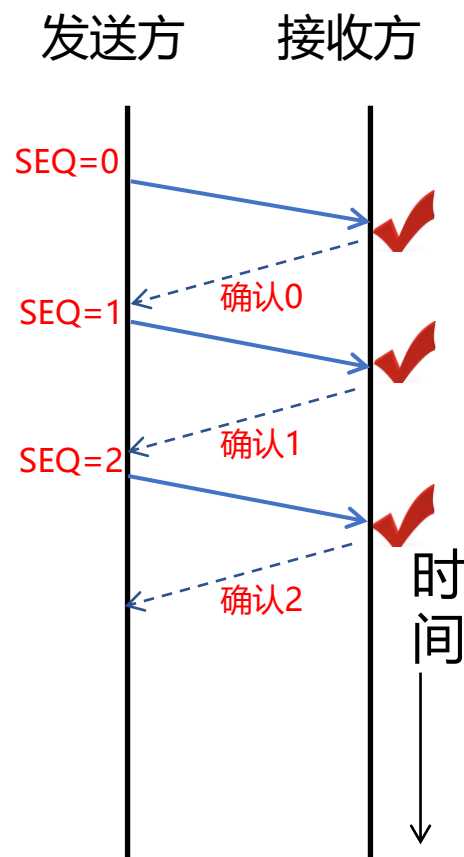
- 接收方需区分到达的帧是否为第一次发来的新帧
- 让发送方在发送的帧的头部放一个序号，接收方可以检查它所收到的帧序号
- 由此判断这是新帧还是应该被丢弃的重复帧

➤ 为精简帧头，序号最小多少位？

- 1 bit序号(0或1)可满足要求

➤ 带有重传的肯定确认

- 发送方在发送下一个数据前，需要得到确认



SEQ=2改为SEQ=0



有错信道上的单工停等式协议P3



清华大学
Tsinghua University



计算机网络教案社区

➤ 序号 (SEQ: sequence number)

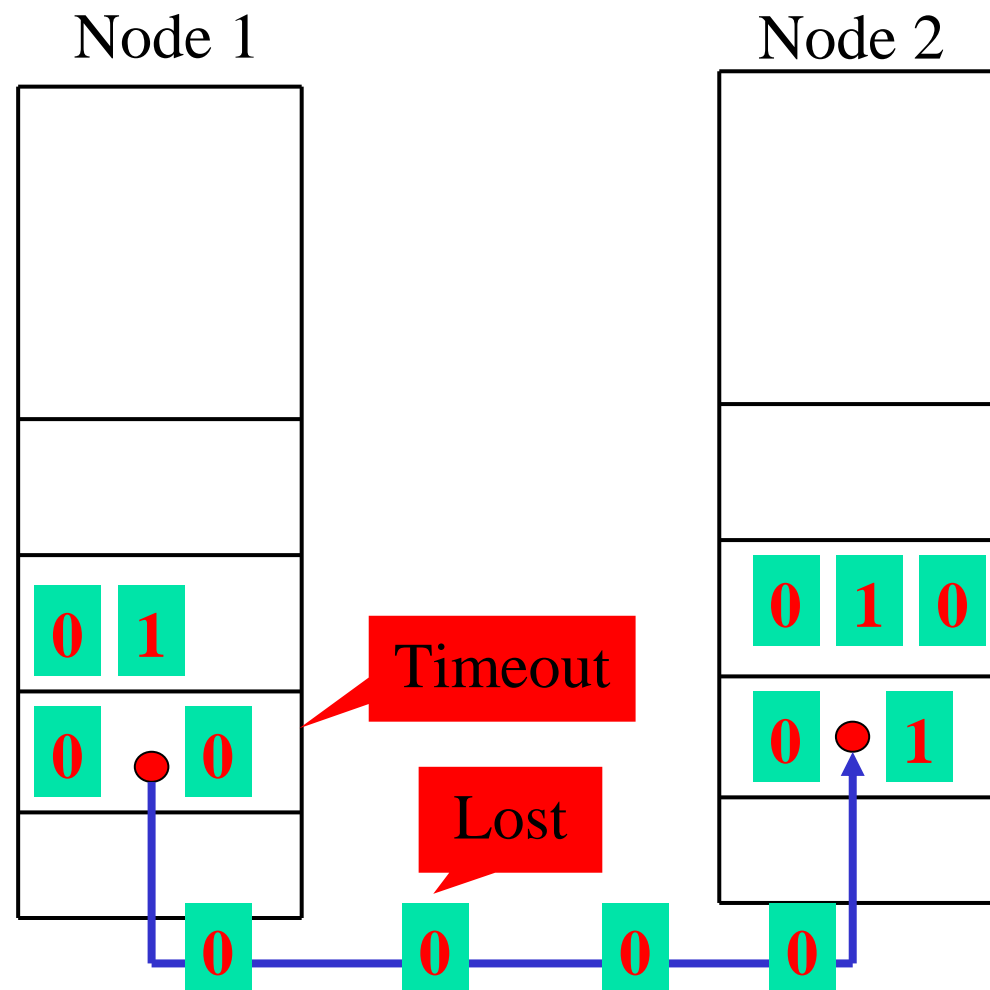
- 接收方需区分到达的帧是否为第一次发来的新帧
- 让发送方在发送的帧的头部放一个序号，接收方可以检查它所收到的帧序号
- 由此判断这是新帧还是应该被丢弃的重复帧

➤ 为精简帧头，序号最小多少位？

- 1 bit序号(0或1)可满足要求

➤ 带有重传的肯定确认

- 发送方在发送下一个数据前，需要得到确认



确认：期望收到的下一帧序号



有错信道上的单工停等式协议P3



➤ 发送方

- 初始化帧序号0，发送帧
- 等待：正确的确认/错误的确认/超时
- 正确确认：发送下一帧
- 超时/错误确认：重发

```
next_frame_to_send = 0;
from_network_layer(&buffer);
while (true) {
    s.info = buffer;
    to_physical_layer(&s);
    start_timer(s.seq);
    wait_for_event(&event);
    if (event == frame_arrival){
        from_physical_layer(&s);
        if (s.ack == next_frame_to_send){
            from_network_layer(&buffer);
            inc(next_frame_to_send);
        }
    }
}
```

➤ 接收方

- 初始化期待0号帧
- 等待帧达到
- 正确帧：交给网络层，并发送该帧确认
- 错误帧：发送上一个成功接收帧的确认

```
frame_expected = 0;
while (true) {
    wait_for_event(&event);
    if (event == frame_arrival){
        from_physical_layer(&r);
        if (r.seq == frame_expected){
            交给网络层 to_network_layer(&r.info);
            inc(frame_expected);
        }
        s.ack = 1 - frame_expected;
        to_physical_layer(&s);
    }
}
```




有错信道上的单工停等式协议P3



清华大学
Tsinghua University



计算机网络教案社区

➤ 效率的评估

- F = 帧大小 (bits)
- R = 链路容量 (Bandwidth in bits/second)
- I = propagation delay + processor service time (second)
- 每帧发送时间 (Time to transmit a single frame) = F/R , 即停止等待协议的发送工作时间
- 发送空闲时间: $D = 2I$, 即为RTT
- 信道利用率 (line utilization) = $F/(F + R \cdot D)$
- 当 $F < DR$ 时, 信道利用率 $< 50\%$

$R \cdot D = 2$ 倍时延带宽积

在什么情况下
利用率特别低?



有错信道上的单工停等式协议P3



清华大学
Tsinghua University



计算机网络教案社区

➤ 停止等待协议的效率问题

- 举例:
 - Short frame length: 1000 bit frames
 - High bandwidth: 1 Mbps channel (卫星信道)
 - Long transit time: 270 ms propagation delay (传播延迟)
- 每一帧的发送时间是 1 毫秒 ($1000 \text{ bits} / (1,000,000 \text{ bits/sec})$)
- 由于传播延迟较长, 发送者在541毫秒之后才能收到确认
- 信道利用率1/541
- 停止等待协议的问题是只能有一个没有被确认的帧在发送中



有错信道上的单工停等式协议P3



清华大学
Tsinghua University



计算机网络教案社区

➤ 信道利用率 = $F / (F + R \cdot D)$ ，帧大小 F ， $R \cdot D$ 为时延带宽积的2倍

➤ 信道利用率很低

长肥网络 (LFN, Long Fat Network): 如果一个网络的带宽-时延积 (bandwidth-delay product) 很明显的大于 10^5 bits (~ 12 kB), 则可以被认为是长肥网络

➤ 提高效率的方法?

- 尝试使用更大的帧?
- 但帧最大长度受信道比特错误率的限制
- 帧越大, 传输出错概率越高, 导致更多重传

停等协议, 效率太低!
如何优化? 下回分解😊



小结



清华大学
Tsinghua University



计算机网络教案社区

➤ 数据链路层关键假设

- 分层进程独立假设
- 提供可靠服务假设
- 只处理通信错误假设

➤ 数据链路层基本协议

- 乌托邦式单工协议P1：完美信道，双方始终就绪，收发瞬间完成
- 无错信道上的停等式协议P2：发送后停下等待ACK，接收后回复ACK
- 有错信道上的单工停等式协议P3：引入计时器和帧序号，处理帧和ACK丢失

➤ 信道利用率计算

- $F/(F + R \cdot D)$ ，F是帧大小，D是发送空闲时间，R是信道容量



总结



清华大学
Tsinghua University



计算机网络教案社区

➤ 成帧的方式

- 字节计数法，带字节填充的定界符法，带比特填充的定界符法

➤ 差错检测和纠正

- 海明距离
- 检错码：奇偶校验，校验和，循环冗余校验
- 纠错码：海明码

➤ 基本的数据链路层协议

- 乌托邦式单工协议P1
- 无错信道上的停等式协议P2
- 有错信道上的单工停等式协议P3
- 信道利用率计算



作业



清华大学
Tsinghua University



计算机网络教案社区

1. 一个上层数据包被分成 10 个帧，每一帧有 80% 的机会无损地到达目的地。如果数据链路协议没有提供错误控制，试问，该报文平均需要发送多少次才能完整地到达接收方？
2. 数据链路协议使用了下面的字符编码：

A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000

为了传输一个包含 4 个字符的帧：A B ESC FLAG，试问使用下面的成帧方法时所发送的比特序列（用二进制表达）是什么？

- (a) 字节计数。
 - (b) 字节填充的标志字节。
 - (c) 比特填充的首尾标志字节。
3. 一个数据流中出现了这样的数据段：A B ESC C ESC FLAG FLAG D，假设采用本章介绍的字节填充算法，试问经过填充之后的输出是什么？
 9. 假设使用海明码来传输 16 位的报文。试问，需要多少个校验位才能确保接收方能同时检测并纠正单个比特错误？对于报文 1101001100110101，试给出传输的比特模式。假设在海明码中使用了偶校验。



作业



清华大学
Tsinghua University



计算机网络教案社区

10. 接收方收到一个 12 位的海明码，其 16 进制值为 0xE4F。试问该码的原始值是多少？
(假设发生错误的bit数不超过1)
17. 使用本章介绍的标准 CRC 方法传输比特流 10011101。生成多项式为 x^3+1 。试问实际传输的位串是什么？假设左边开始的第三个比特在传输过程中变反了。请说明这个错误可以在接收方被检测出来。给出一个该比特流传输错误的实例，使得接收方无法检测出该错误。
20. 考虑一个具有 4 kbps 速率和 20 毫秒传输延迟的信道。试问帧的大小在什么范围内，停-等式协议才能获得至少 50%的效率？



致谢社区本章贡献者



清华大学
Tsinghua University



计算机网络教案社区



王贵竹



王艳



林卫国



王昊翔



陈振国



张浩



徐涛



徐敬东