

慕课网
imooc.com

慕课网
imooc.com

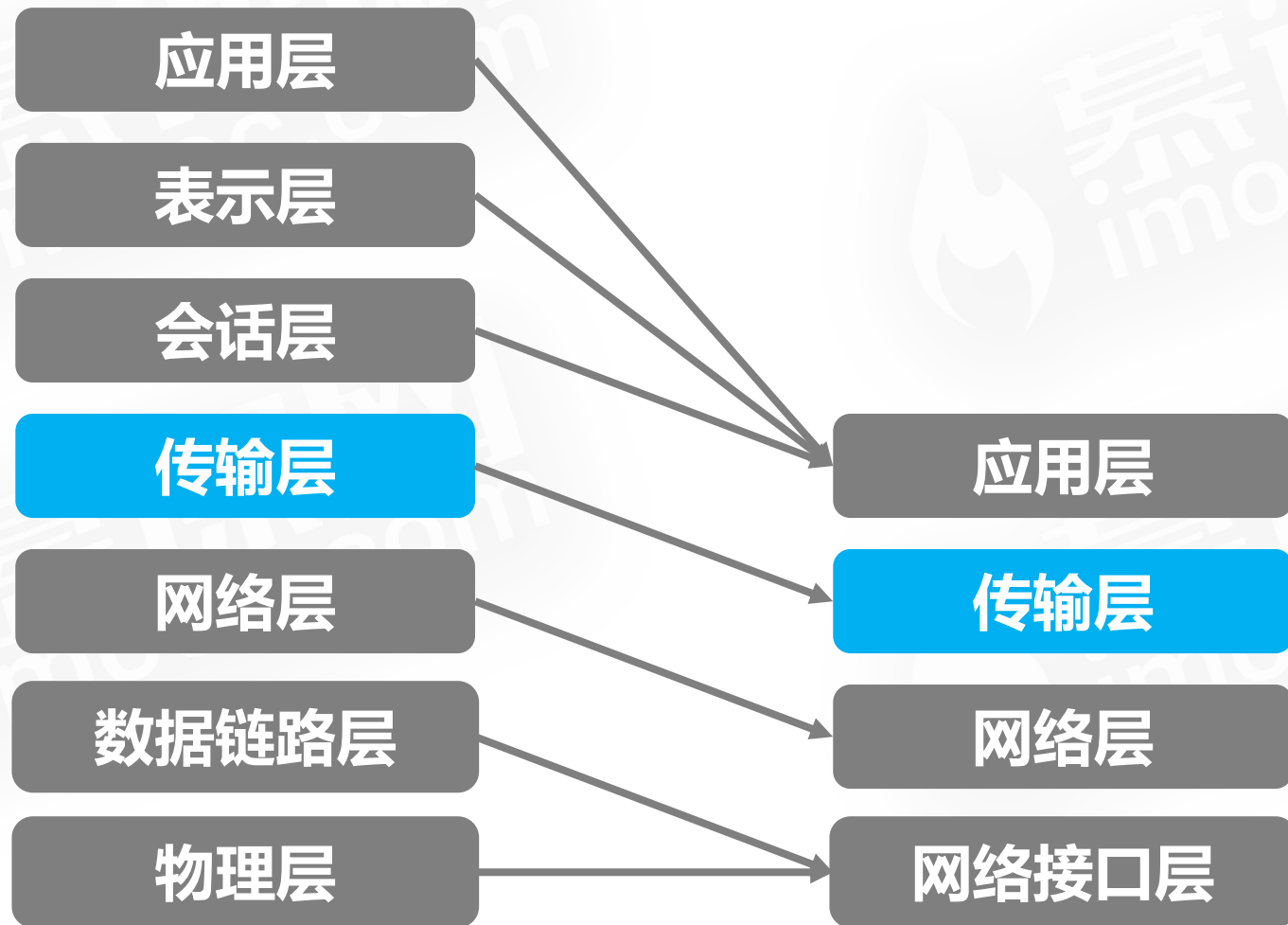
慕课网
imooc.com

慕课网
imooc.com

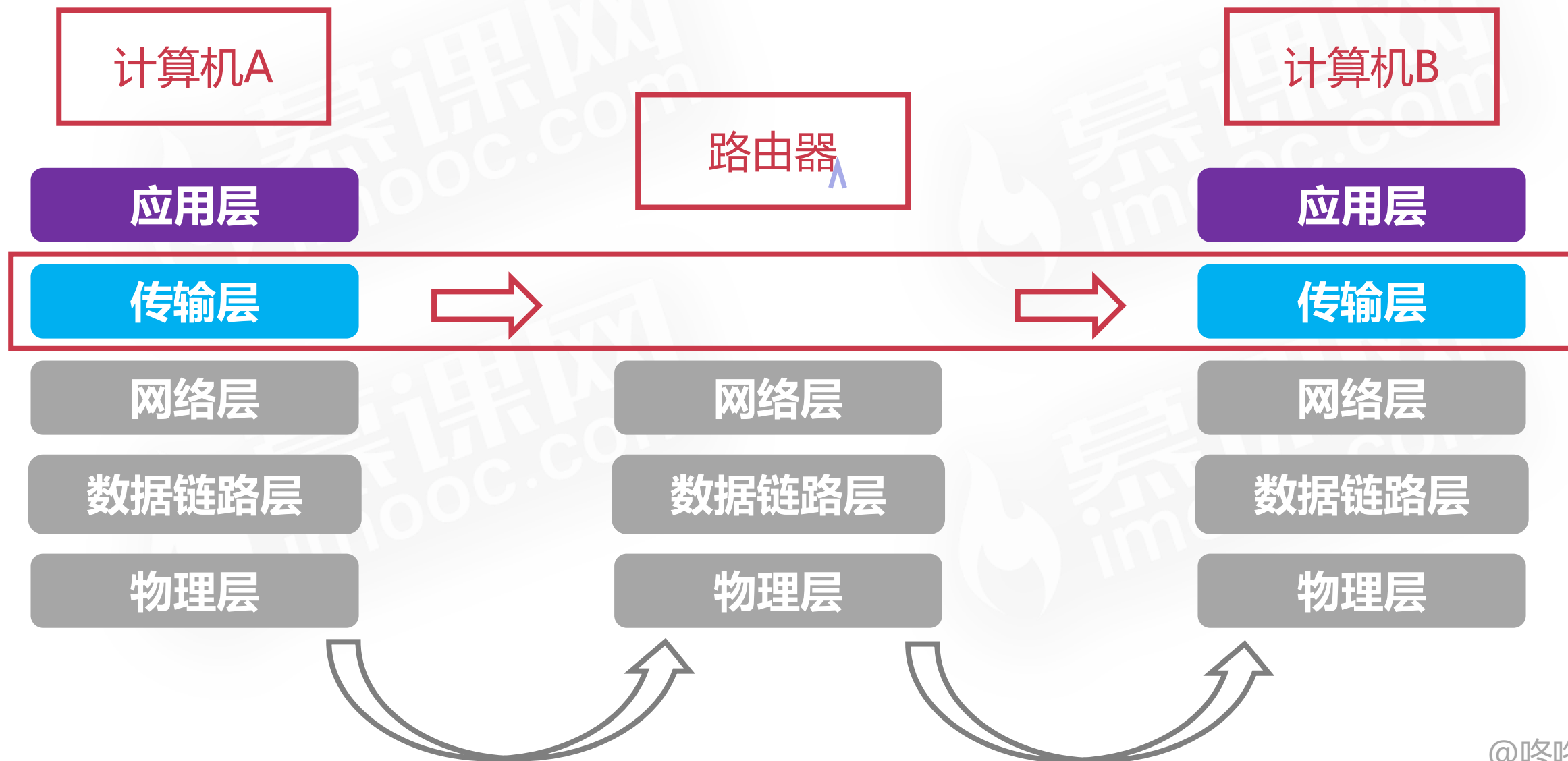
章节导学



传输层概述



传输层概述



传输层概述



传输层概述



进程与进程的通信

传输层概述

...

Unix域套接字

共享内存

网络通信

进程与进程的通信

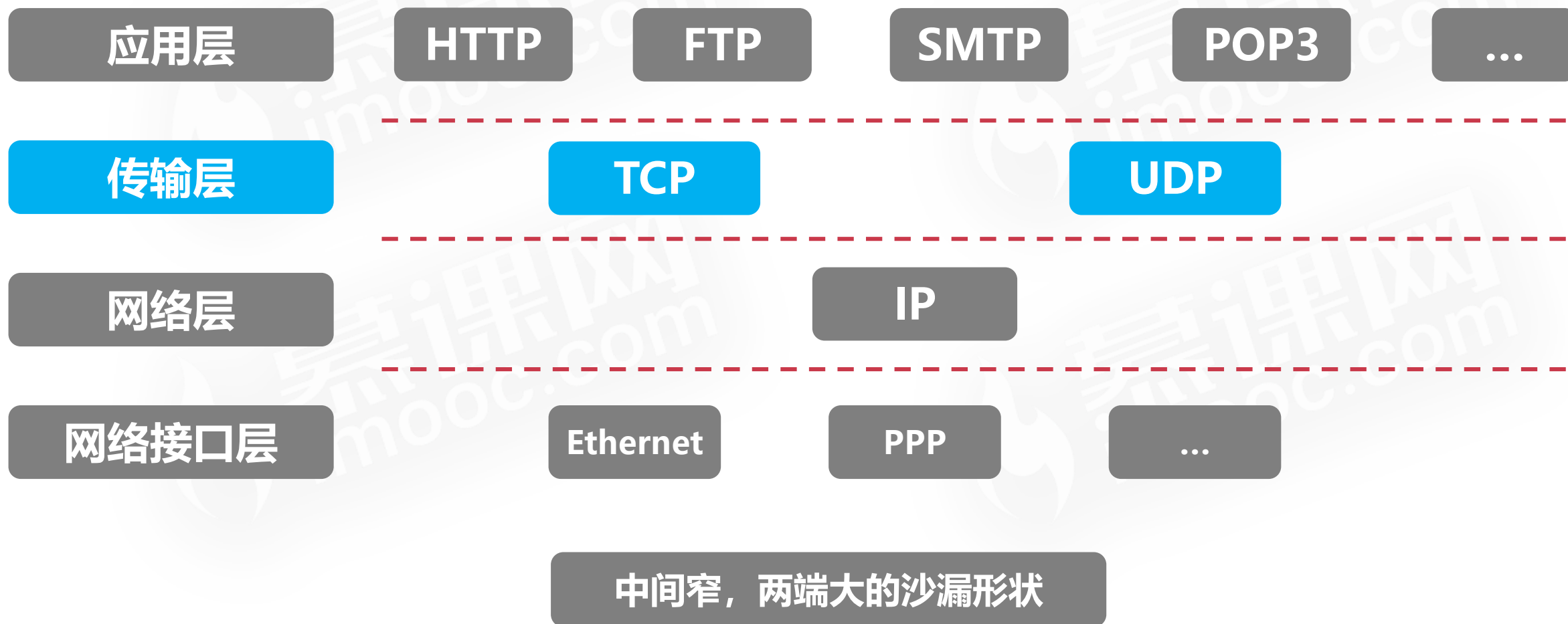
传输层概述

◆ 使用端口(Port)来标记不同的网络进程

◆ 端口(Port)使用16比特位表示(0~65535)

FTP	HTTP	HTTPS	DNS	TELNET
21	80	443	53	23

传输层概述



传输层概述



慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

UDP协议详解

- ◆ UDP(User Datagram Protocol: 用户数据报协议)
- ◆ UDP是一个非常简单的协议

数据报(Datagram)

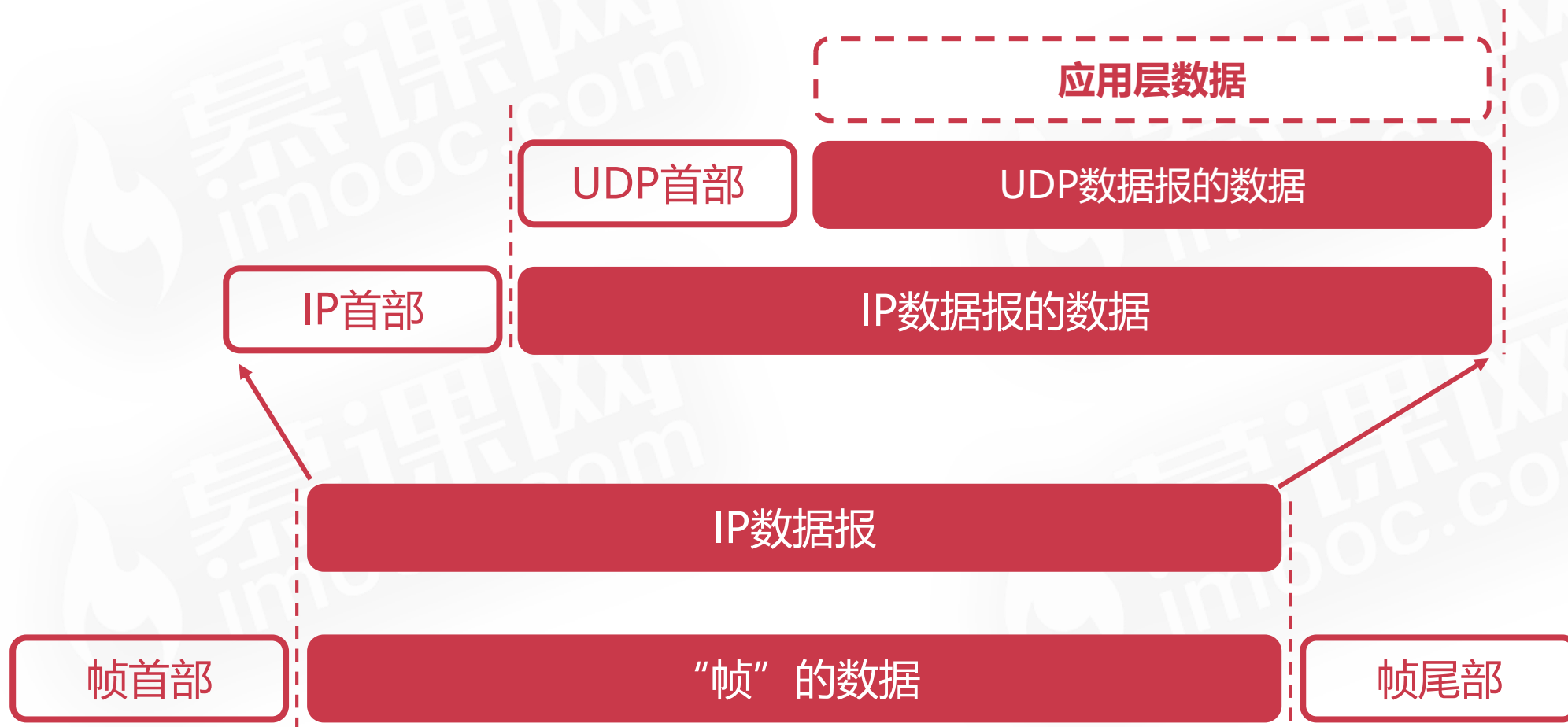
DATA

DATA

DATA

不合并 不拆分

UDP协议详解



UDP协议详解

16位源端口号	16位目的端口号
16位UDP长度	16位UDP校验和
UDP数据	

UDP协议详解

- ◆ UDP是**无连接**协议



UDP协议详解

- ◆ UDP不能保证可靠的交付数据

“想发就发”，“无法保证数据在网络中是否丢失”

UDP协议详解

- ◆ UDP是面向报文传输的



UDP协议详解

- ◆ UDP没有拥塞控制
- ◆ UDP的首部开销很小

UDP协议详解

- ◆ UDP协议
- ◆ UDP的特点

慕课网
imooc.com

慕课网
imooc.com

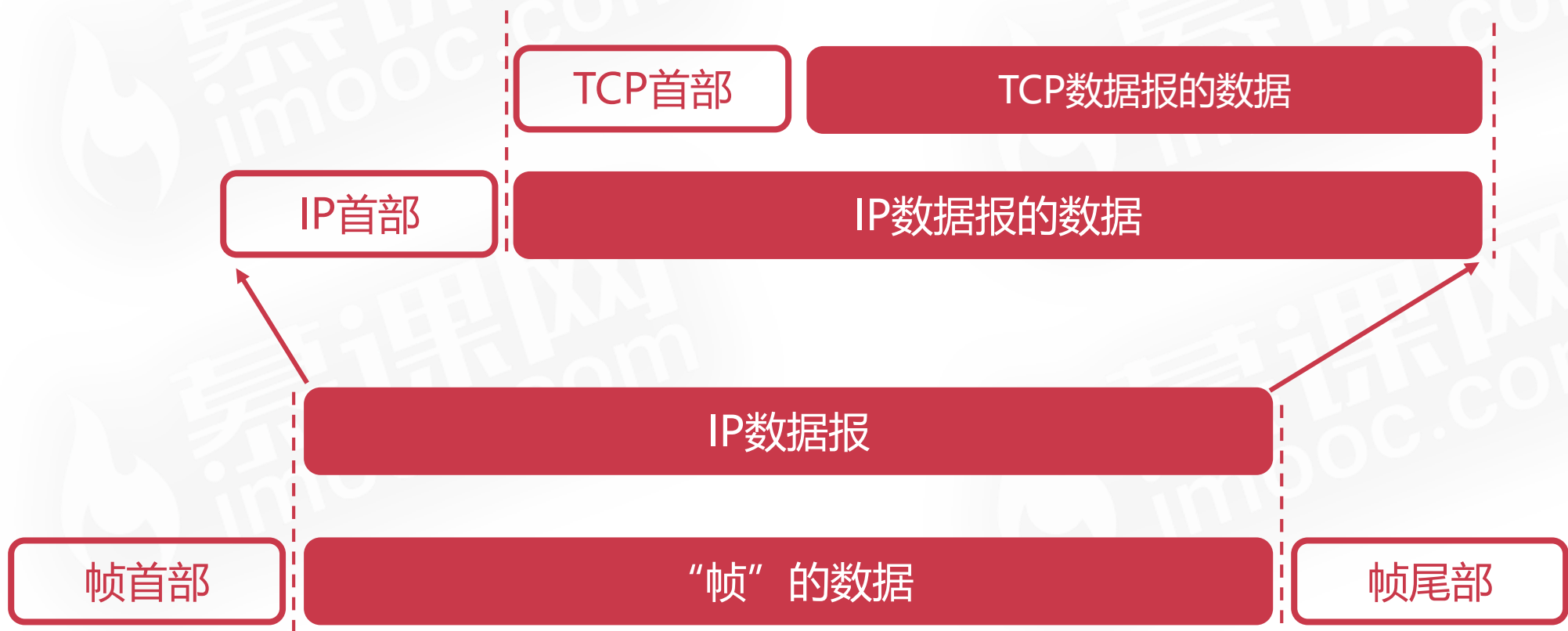
慕课网
imooc.com

慕课网
imooc.com

TCP协议详解

- ◆ TCP(Transmission Control Protocol: 传输控制协议)
- ◆ TCP协议是计算机网络中非常复杂的一个协议

TCP协议详解

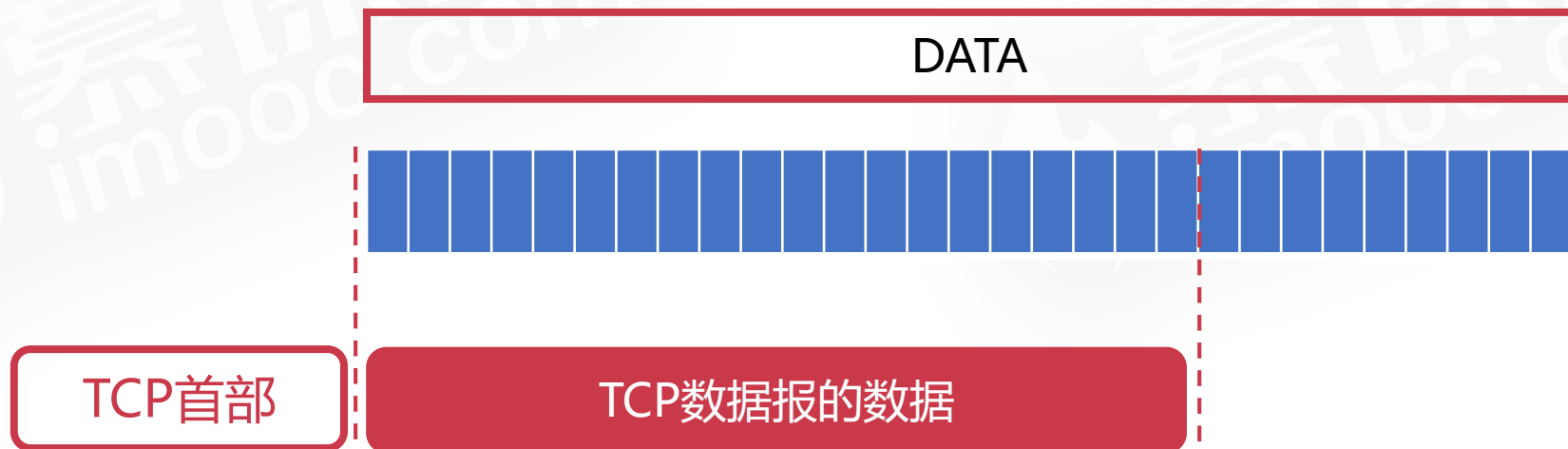


TCP协议详解

- ◆ TCP是**面向连接**的协议
- ◆ TCP的一个连接有两端（点对点通信）
- ◆ TCP**提供可靠**的传输服务

TCP协议详解

- ◆ TCP协议提供**全双工**的通信
- ◆ TCP是面向**字节流**的协议



TCP协议详解

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

固定20字节

TCP协议详解

序号

- ◆ $0 \sim 2^{32}-1$
- ◆ 一个字节一个序号
- ◆ 数据首字节序号

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP首部

TCP数据报的数据

TCP协议详解

确认号

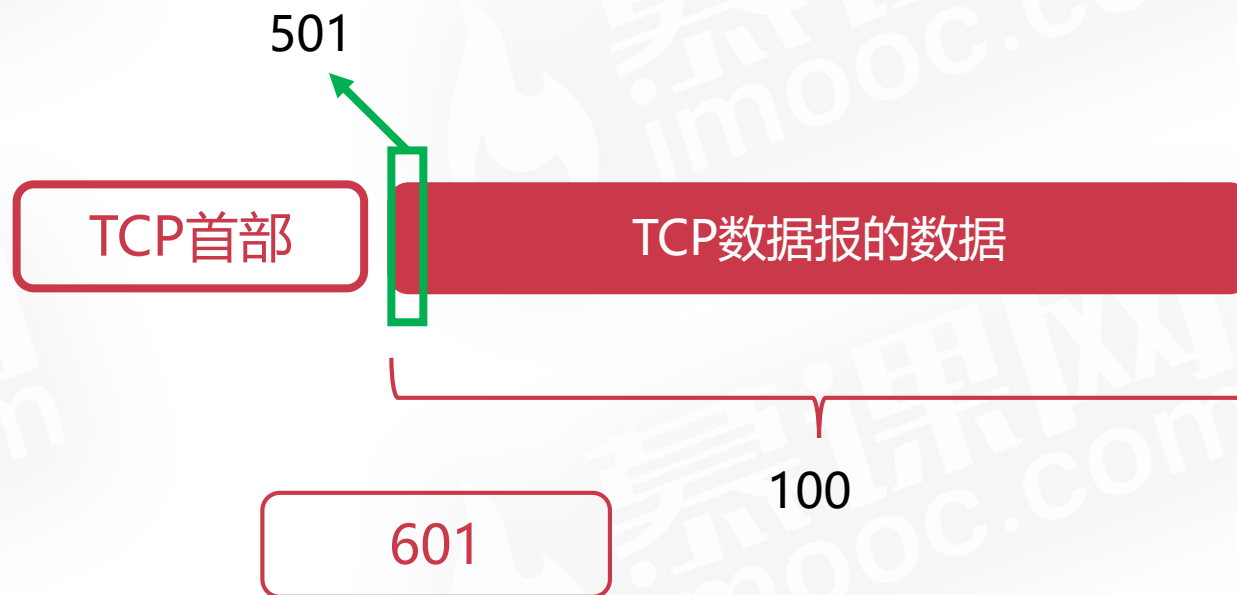
- ◆ $0 \sim 2^{32}-1$
- ◆ 一个字节一个序号
- ◆ 期望收到数据的首字节序号

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议详解

确认号

- ◆ $0 \sim 2^{32}-1$
- ◆ 一个字节一个序号
- ◆ 期望收到数据的首字节序号



确认号为N：则表示N-1序号的数据都已经收到

TCP协议详解

数据偏移

- ◆ 占4位：0~15，单位为：32位字
- ◆ 数据偏离首部的距离

20~60字节

TCP首部

TCP数据报的数据

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）			填充	

TCP协议详解

TCP标记

- ◆ 占6位，每位各有不同意义

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议详解

TCP标记

标记	含义
URG	Urgent: 紧急位, URG=1, 表示紧急数据
ACK	Acknowledgement: 确认位, ACK=1, 确认号才生效
PSH	Push: 推送位, PSH=1, 尽快地把数据交付给应用层
RST	Reset: 重置位, RST=1, 重新建立连接
SYN	Synchronization: 同步位, SYN=1 表示连接请求报文
FIN	Finish: 终止位, FIN=1 表示释放连接

TCP协议详解

窗口

◆ 占16位: $0 \sim 2^{16}-1$

◆ 窗口指明允许对方发送的数据量

501

1000

501~1500

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议详解

校验和

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议详解

紧急指针

- ◆ 紧急数据 (URG=1)
- ◆ 指定紧急数据在报文的位置

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议详解

TCP选项

- ◆ 最多40字节
- ◆ 支持未来的拓展

20~60字节

TCP首部

TCP数据报的数据

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议详解

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

固定20字节

TCP协议详解

- ◆ TCP是什么
- ◆ TCP的特点
- ◆ TCP的首部

慕课网
imooc.com

慕课网
imooc.com

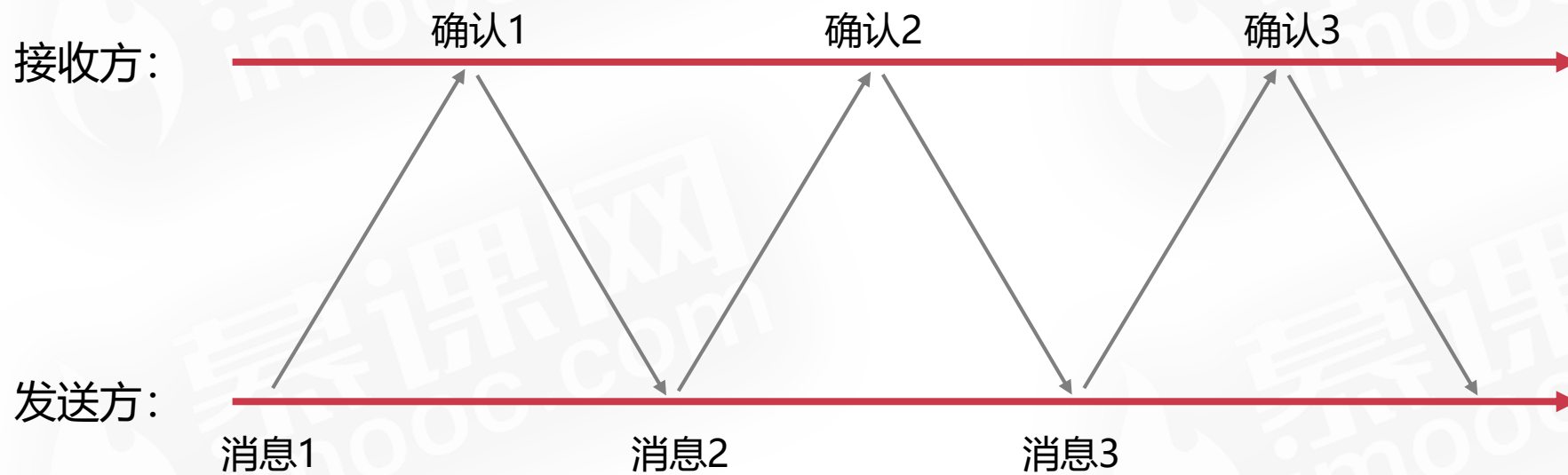
慕课网
imooc.com

慕课网
imooc.com

可靠传输的基本原理

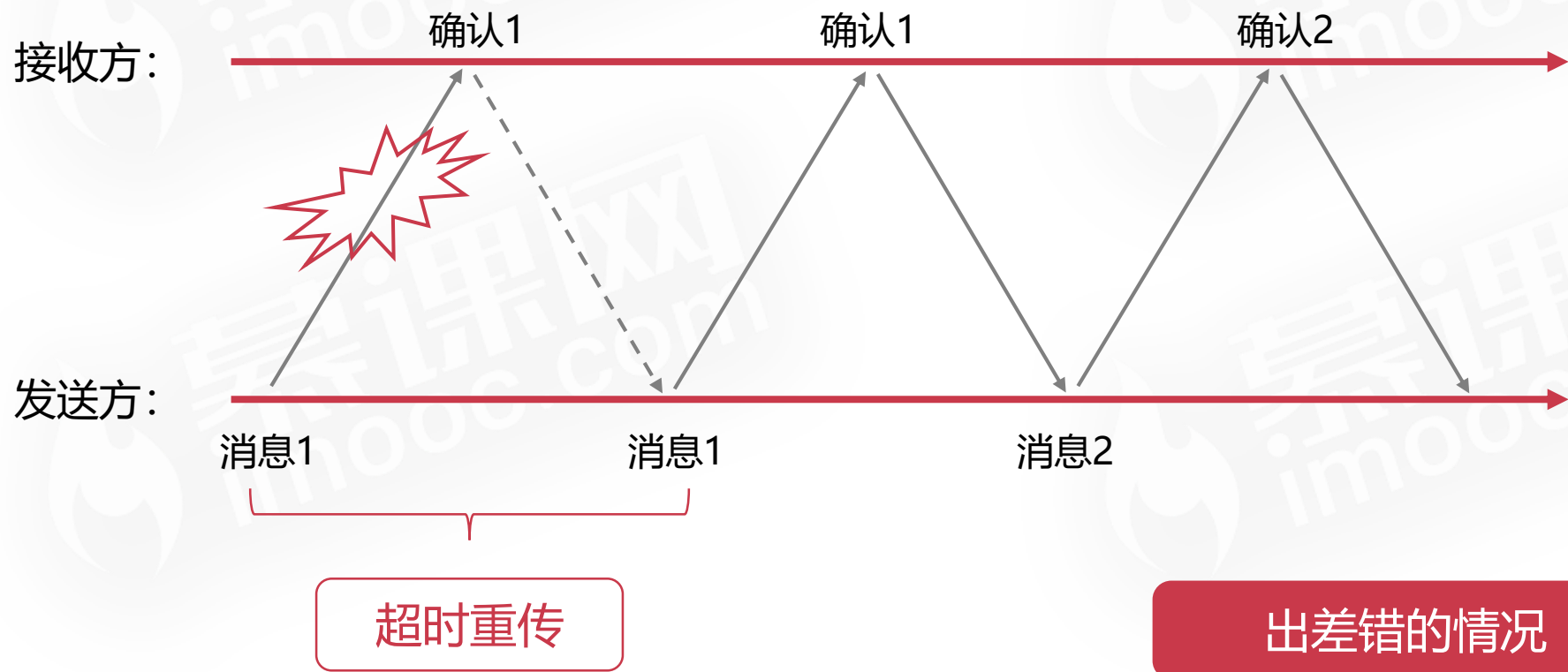
- ◆ 停止等待协议
- ◆ 连续ARQ协议

停止等待协议

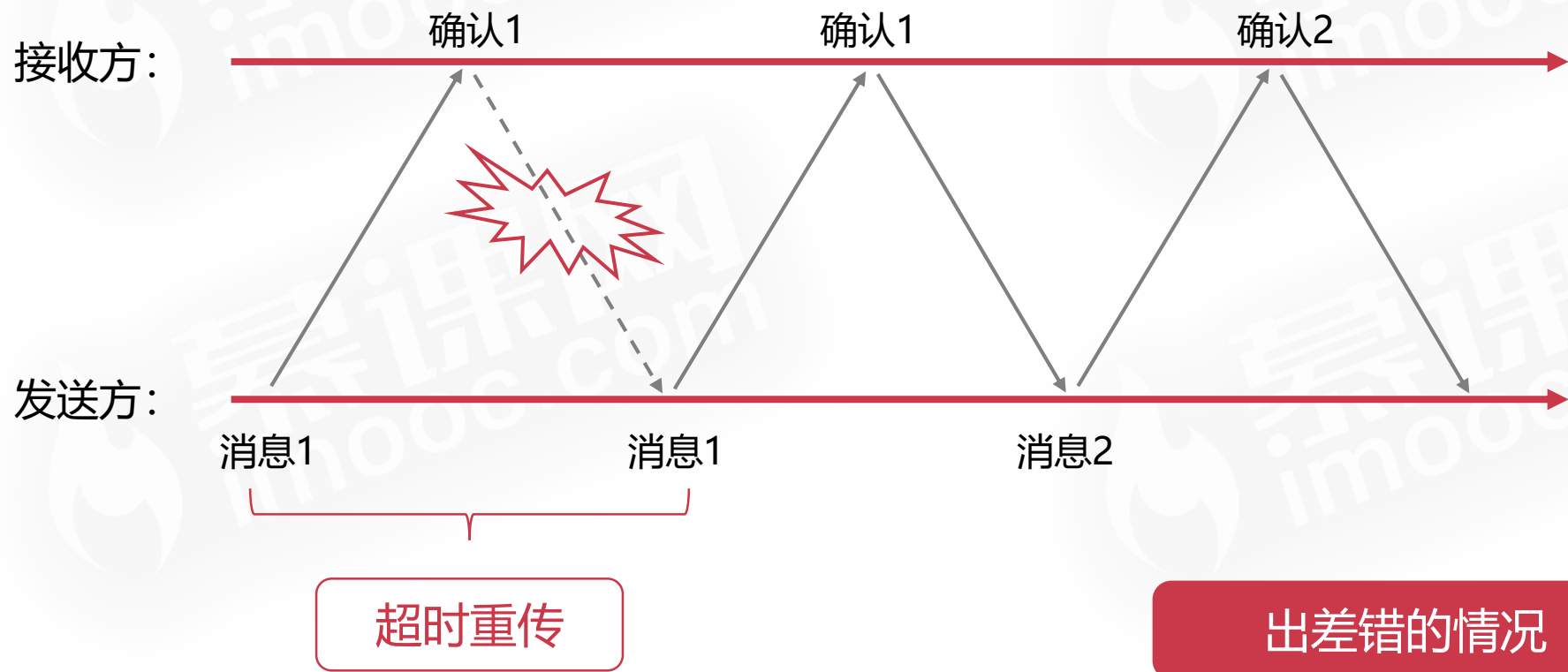


无差错的情况

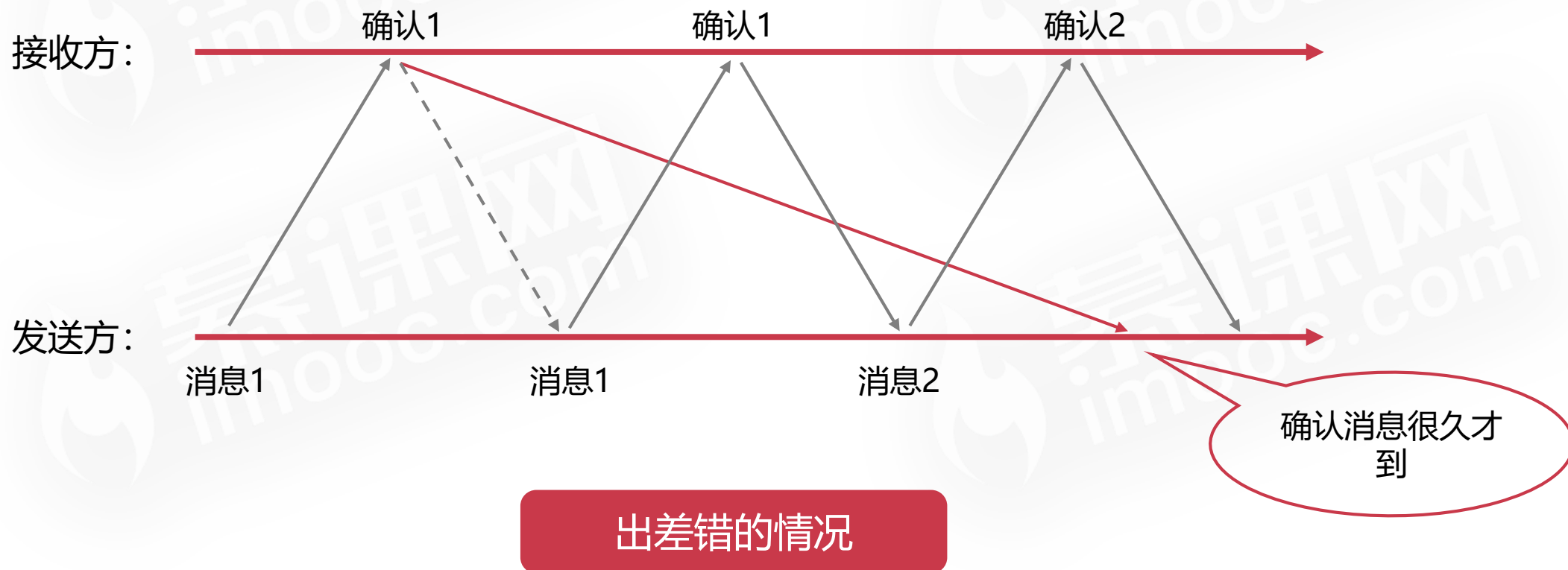
停止等待协议



停止等待协议



停止等待协议



停止等待协议

- ◆ 发送的消息在路上丢失了
- ◆ 确认的消息在路上丢失了
- ◆ 确认的消息很久才到

可靠传输

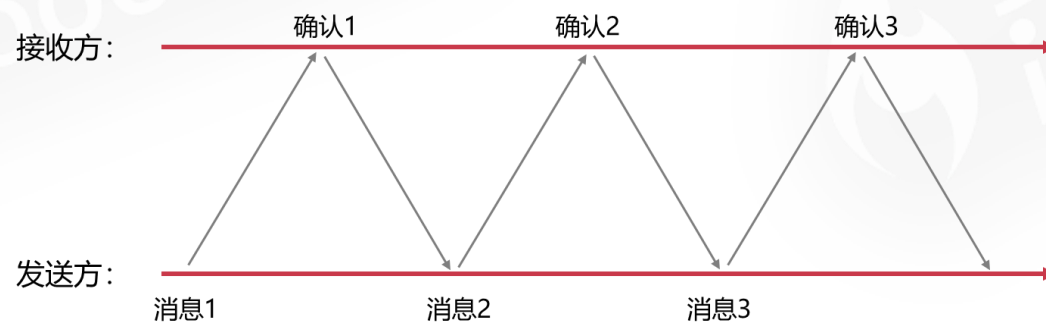
停止等待协议

- ◆ 每发送一个消息，都需要设置一个定时器

超时定时器

停止等待协议

- ◆ 停止等待协议是最简单的可靠传输协议
- ◆ 停止等待协议对信道的利用效率不高



可靠传输的基本原理

- ◆ 停止等待协议
- ◆ 连续ARQ协议

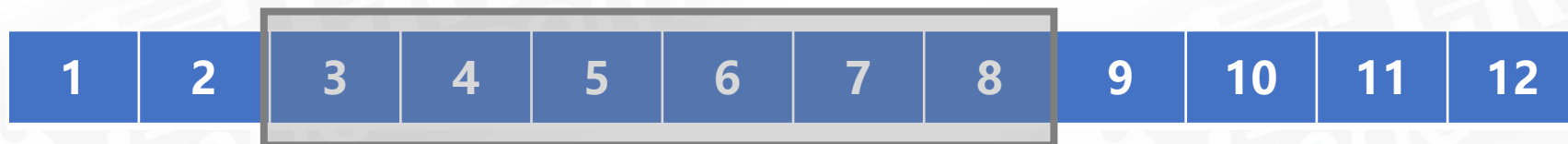
连续ARQ协议

- ◆ ARQ(Automatic Repeat reQuest: 自动重传请求)

既然单个发送和确认效率低，
可不可以批量发送和确认？

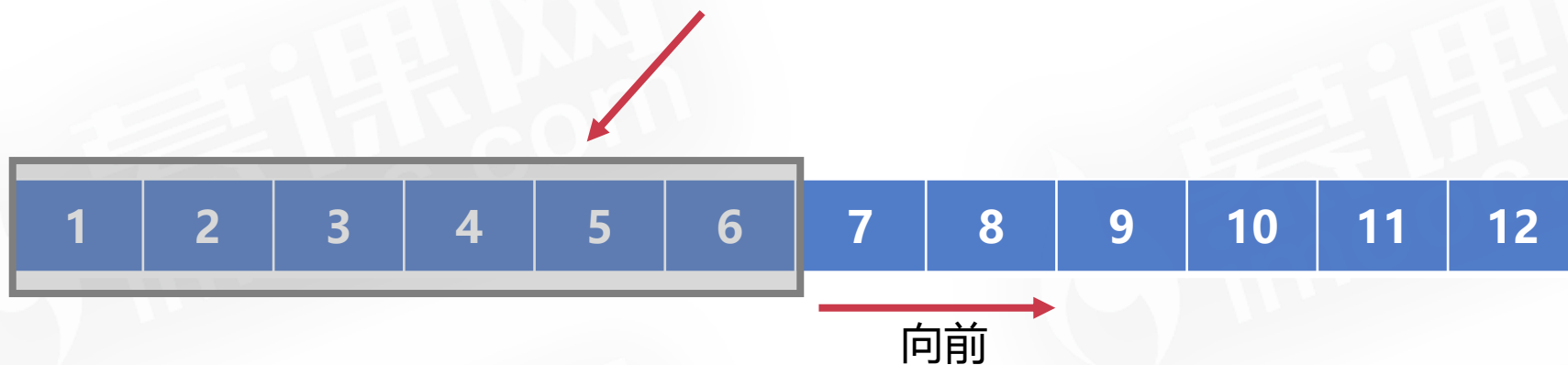


连续ARQ协议



滑动窗口

连续ARQ协议



累计确认

可靠传输的基本原理

- ◆ 停止等待协议
- ◆ 连续ARQ协议

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

TCP协议的可靠传输

◆ TCP的可靠传输基于连续ARQ协议

◆ TCP的滑动窗口以字节为单位

TCP协议的可靠传输

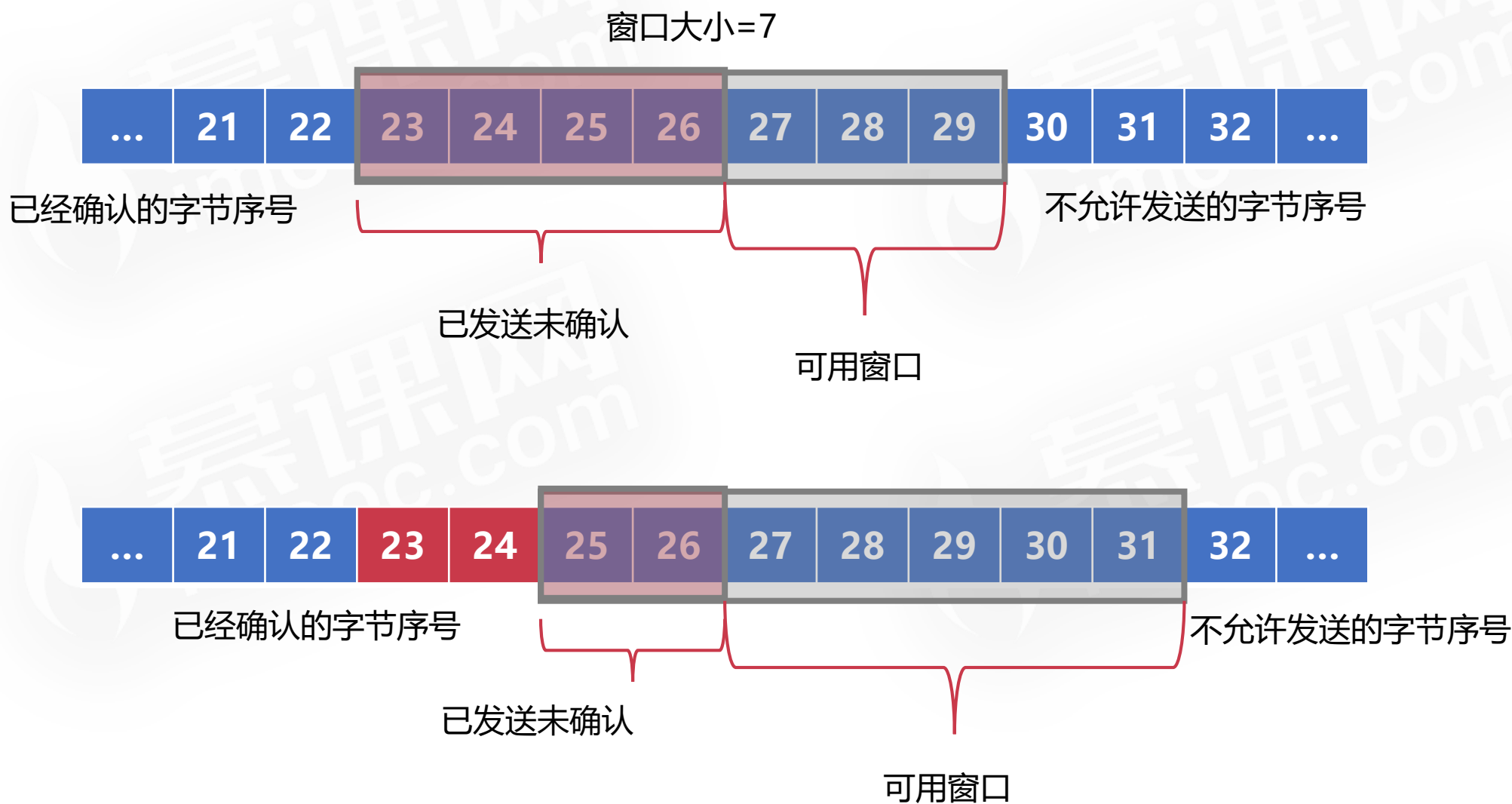
窗口大小=7



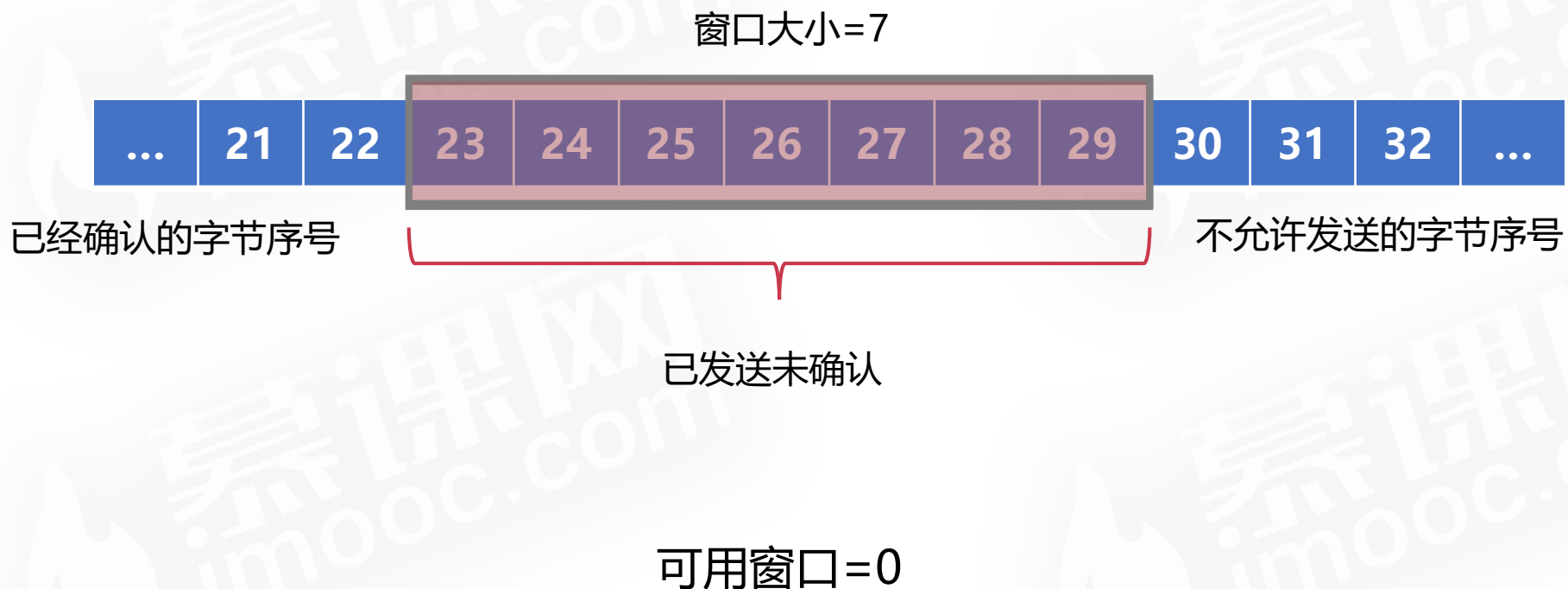
对方期待
收到的下
一个字节

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

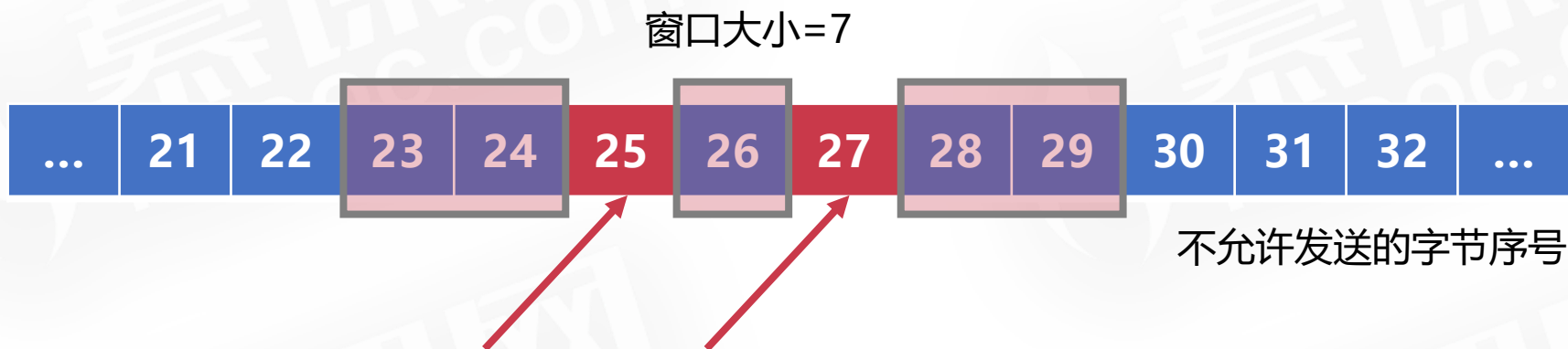
TCP协议的可靠传输



TCP协议的可靠传输



TCP协议的可靠传输



- ◆ 23、24没有收到
- ◆ 从23开始重传

选择重传

TCP协议的选择重传

- ◆ 选择重传需要指定需要重传的字节
- ◆ 每一个字节都有唯一的32位序号

TCP协议的选择重传



TCP协议的选择重传

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP协议的选择重传

数据偏移

- ◆ 占4位：0~15，单位为：32位字
- ◆ 数据偏离首部的距离

20~60字节

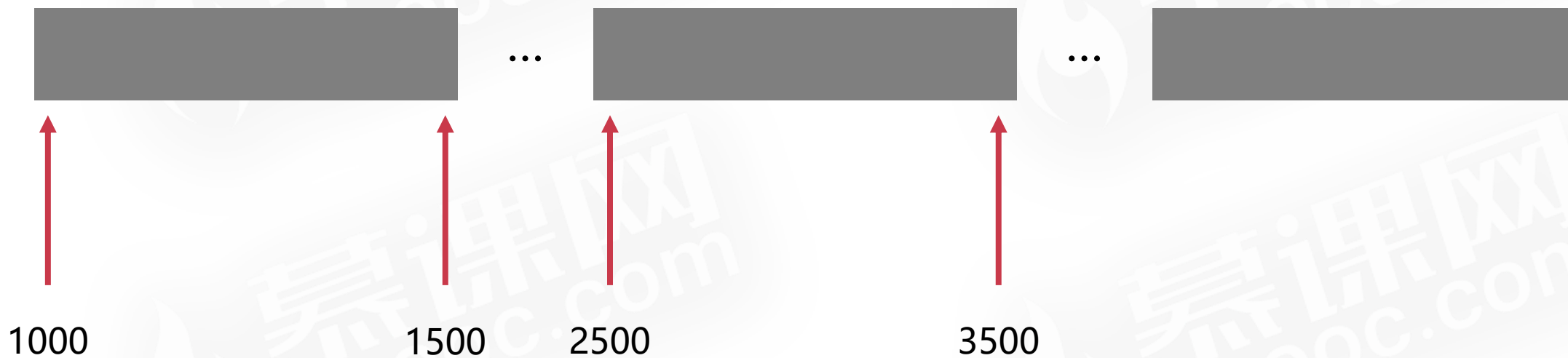
TCP首部



TCP选项最多40个字节

10个序号

TCP协议的选择重传



TCP协议的可靠传输

- ◆ TCP的可靠传输基于连续ARQ协议
- ◆ 选择重传

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

TCP协议的流量控制

◆ 流量控制指让发送方发送速率不要太快

◆ 流量控制是使用滑动窗口来实现的

TCP协议的流量控制

窗口

- ◆ 占16位: $0 \sim 2^{16}-1$
- ◆ 窗口指明允许对方发送的数据量

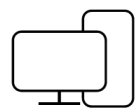
501

1000

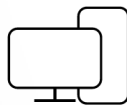
501~1500

16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

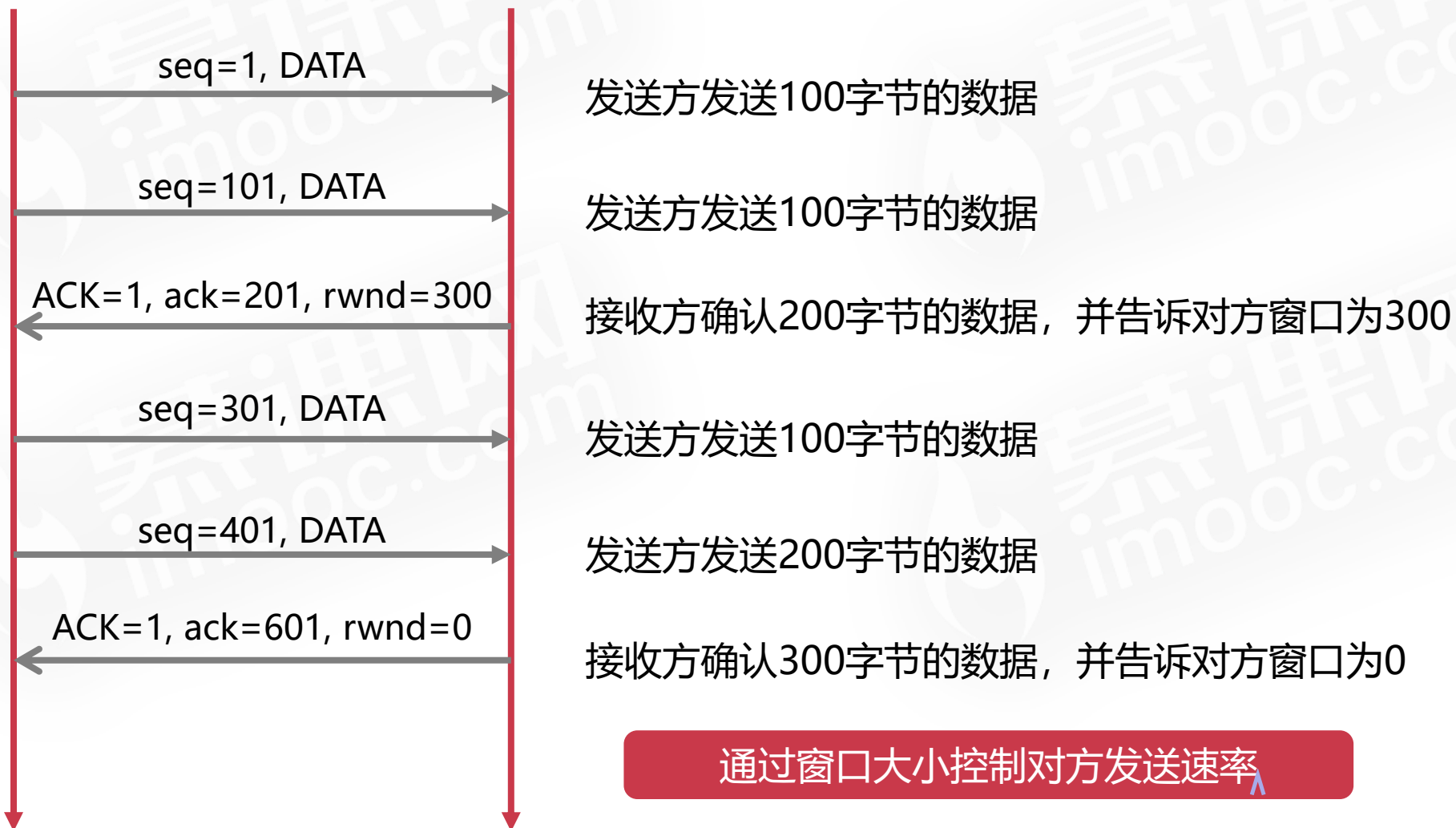
TCP协议的流量控制



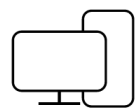
发送方:



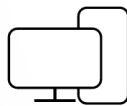
接收方:



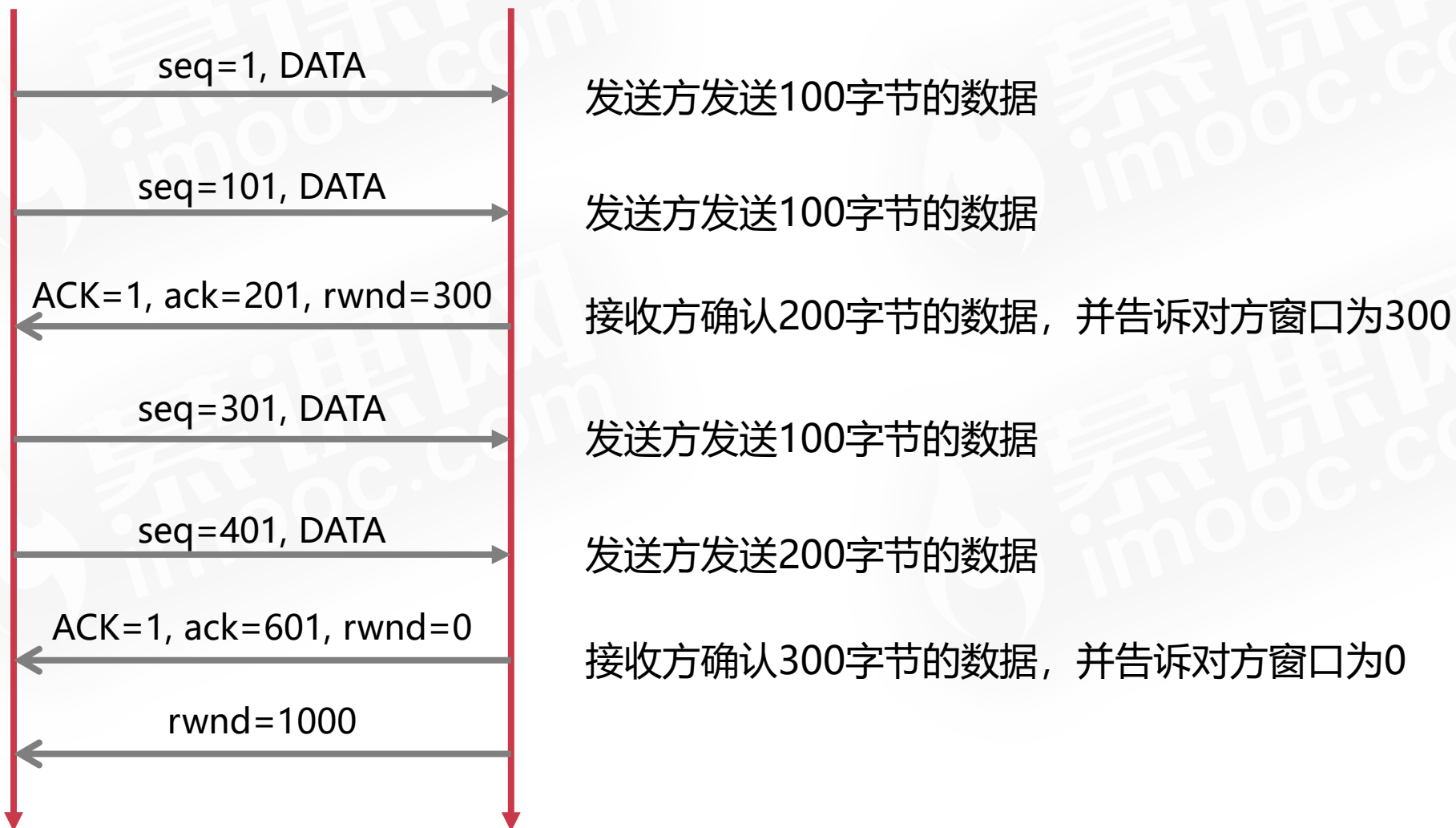
TCP协议的流量控制



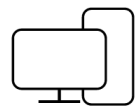
发送方:



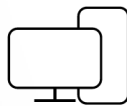
接收方:



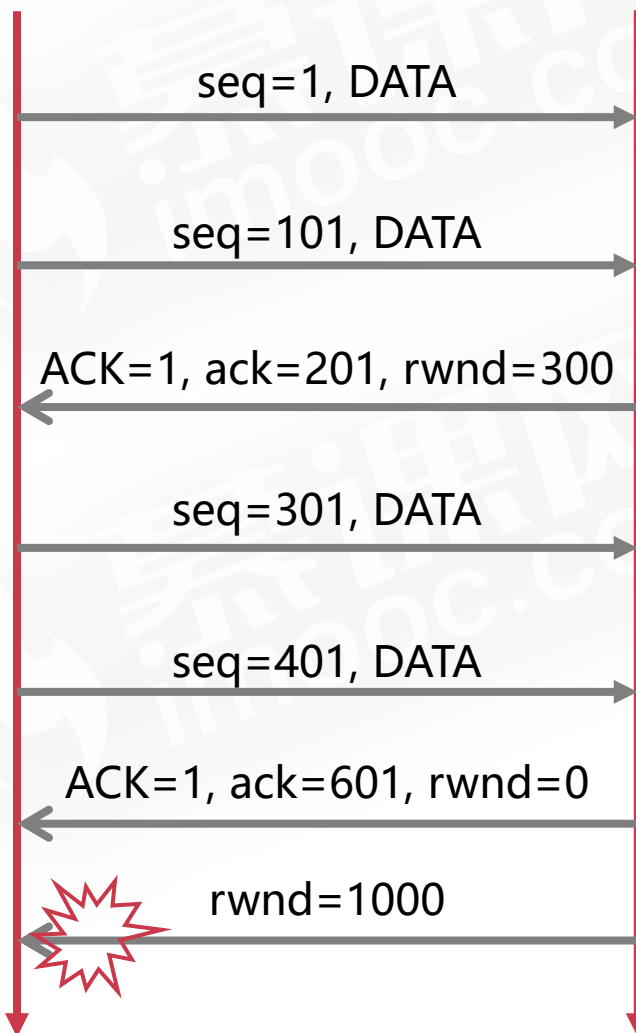
TCP协议的流量控制



发送方:



接收方:



发送方发送100字节的数据

发送方发送100字节的数据

接收方确认200字节的数据，并告诉对方窗口为300

发送方发送100字节的数据

发送方发送200字节的数据

接收方确认300字节的数据，并告诉对方窗口为0

坚持定时器

TCP协议的流量控制

坚持定时器

- ◆ 当接收到窗口为0的消息，则启动坚持定时器
- ◆ 坚持定时器每隔一段时间发送一个窗口探测报文

TCP协议的流量控制

- ◆ TCP协议使用滑动窗口实现流量控制
- ◆ 坚持定时器的作用

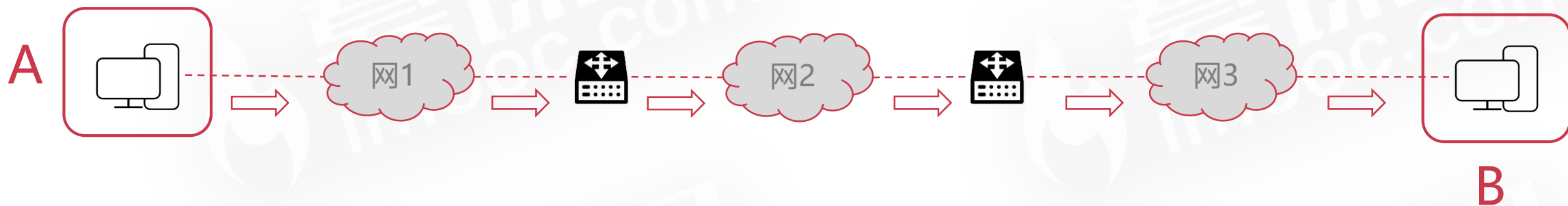
慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

TCP协议的拥塞控制



- ◆ 一条数据链路经过非常多的设备
- ◆ 数据链路中各个部分都有可能成为网路传输的瓶颈

TCP协议的拥塞控制

- ◆ 流量控制考虑点对点的通信量的控制
- ◆ 拥塞控制考虑整个网络，是全局性的考虑

报文超时则认为是拥塞

TCP协议的拥塞控制

慢启动算法

- ◆ 由小到大逐渐增加发送数据量
- ◆ 每收到一个报文确认，就加一

1 2 4 8 16

慢启动阈值(ssthresh)

指数增长

TCP协议的拥塞控制

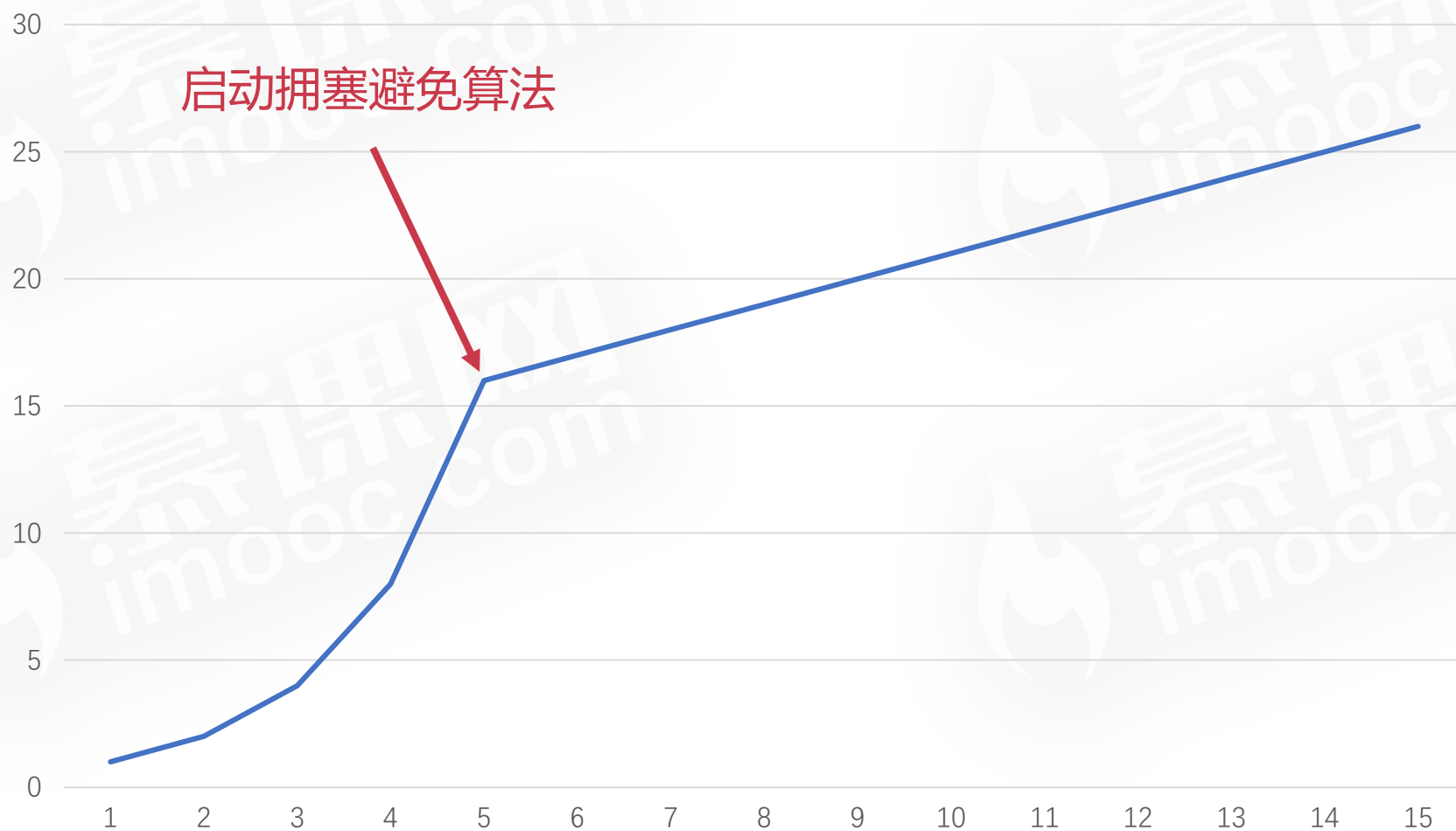
拥塞避免算法

- ◆ 维护一个拥塞窗口的变量
- ◆ 只要网络不拥塞，就试探着拥塞窗口调大

1	2	4	8	16
		17	18	19

慢启动阈值(ssthresh)

TCP协议的拥塞控制



TCP协议的拥塞控制

- ◆ 流量控制与拥塞控制的区别
- ◆ 拥塞控制的方法

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

TCP连接的建立

TCP标记

- ◆ 占6位，每位各有不同意义

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N

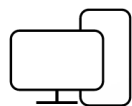
16位源端口			16位目的端口	
序号				
确认号				
数据偏移	保留字段	TCP标记	窗口	
校验和			紧急指针	
TCP选项（可选）				填充

TCP连接的建立

TCP标记

标记	含义
URG	Urgent: 紧急位, URG=1, 表示紧急数据
ACK	Acknowledgement: 确认位, ACK=1, 确认号才生效
PSH	Push: 推送位, PSH=1, 尽快地把数据交付给应用层
RST	Reset: 重置位, RST=1, 重新建立连接
SYN	Synchronization: 同步位, SYN=1 表示连接请求报文
FIN	Finish: 终止位, FIN=1 表示释放连接

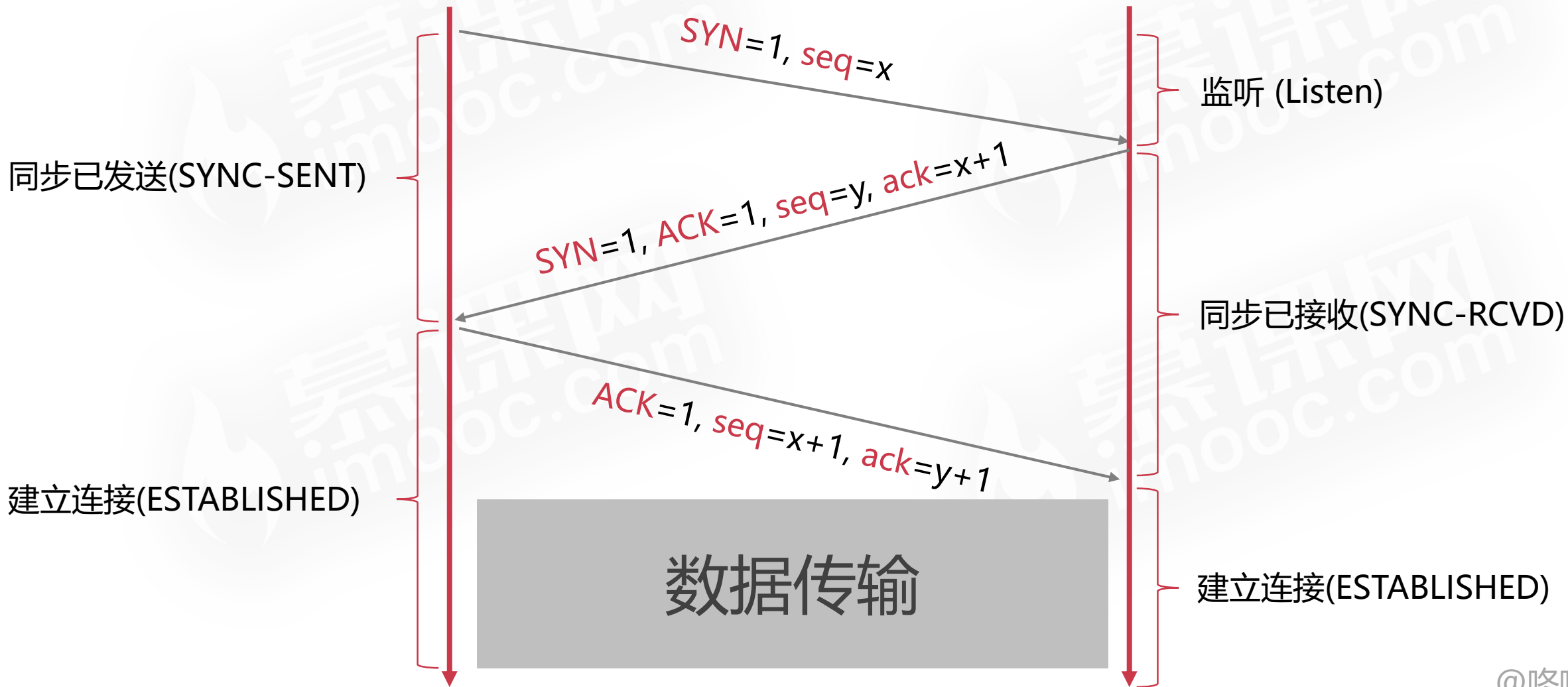
TCP连接的建立



发送方:



接收方:

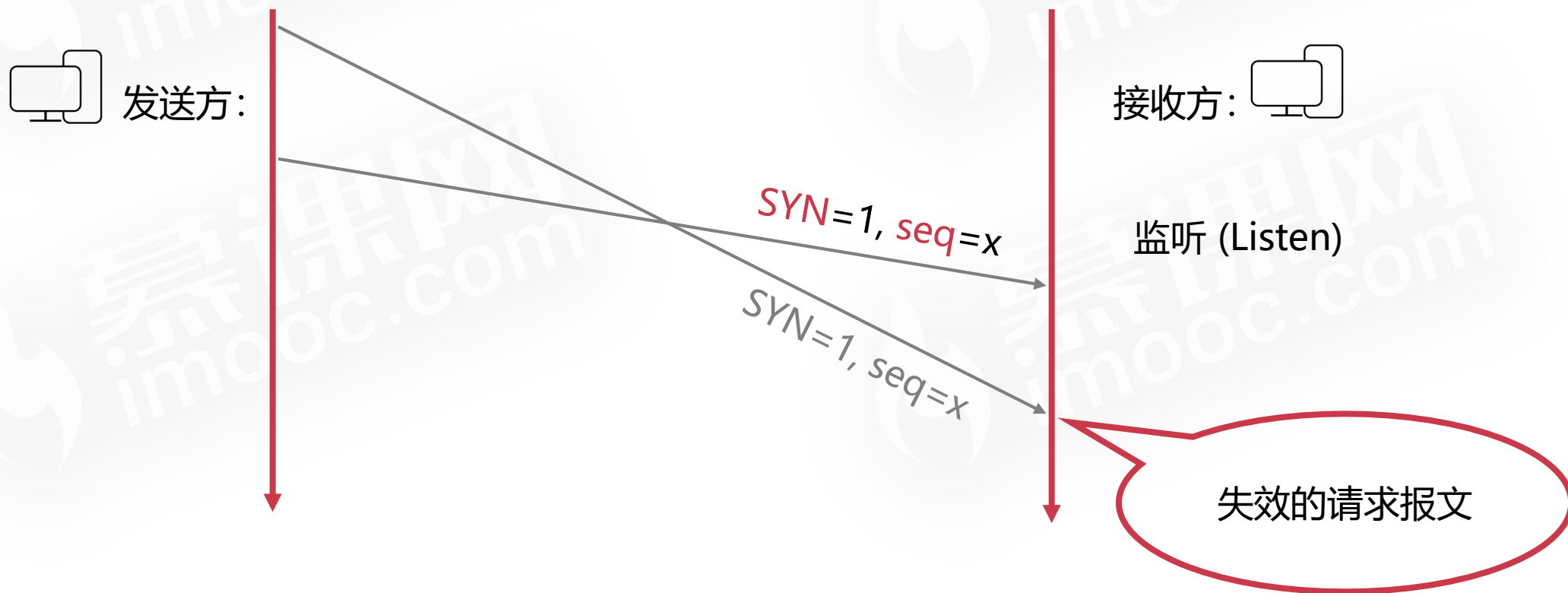


TCP连接的建立

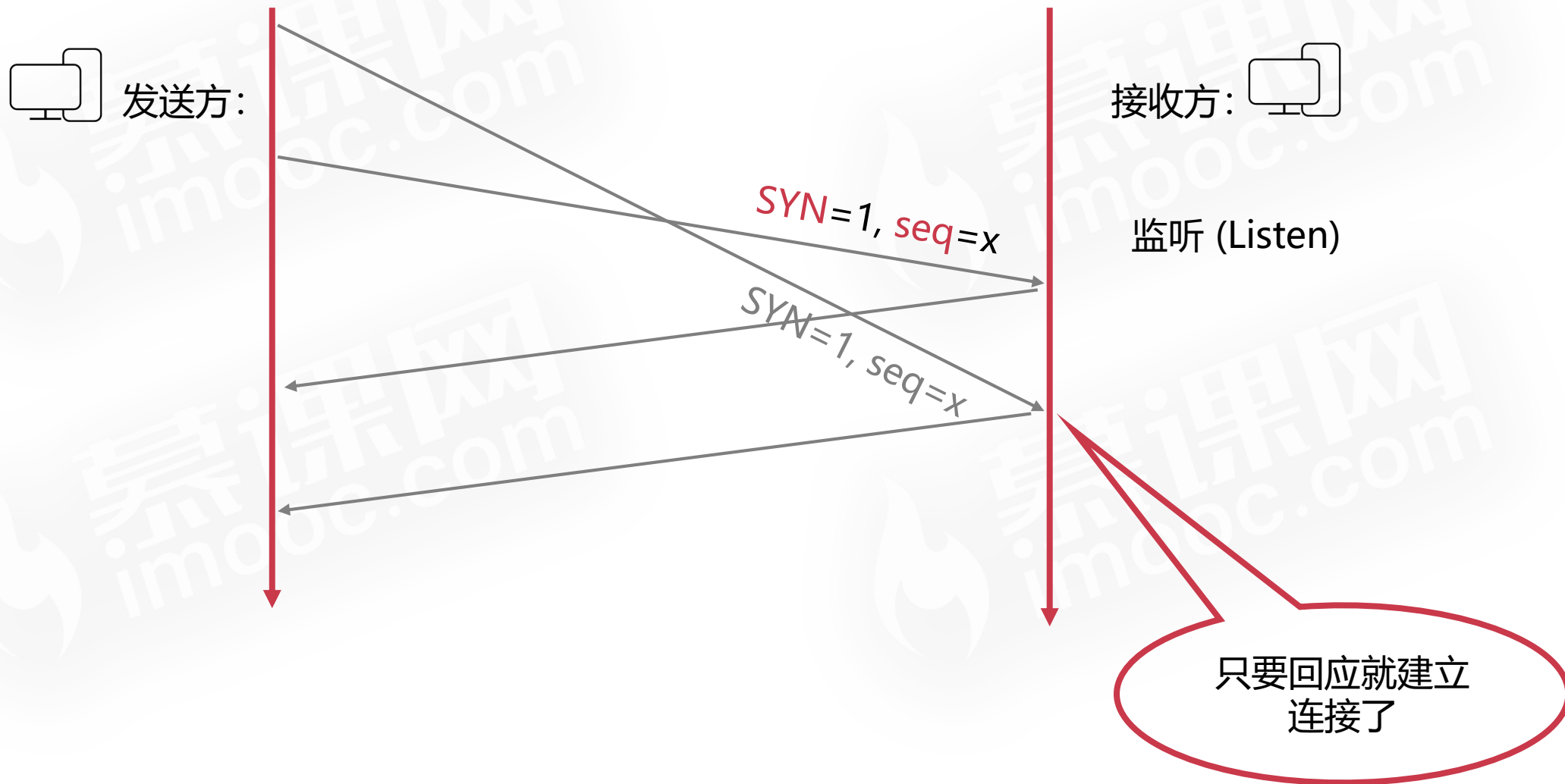
为什么发送方要发出第三个确认报文呢？

- ◆ 已经失效的连接请求报文传送到对方，引起错误

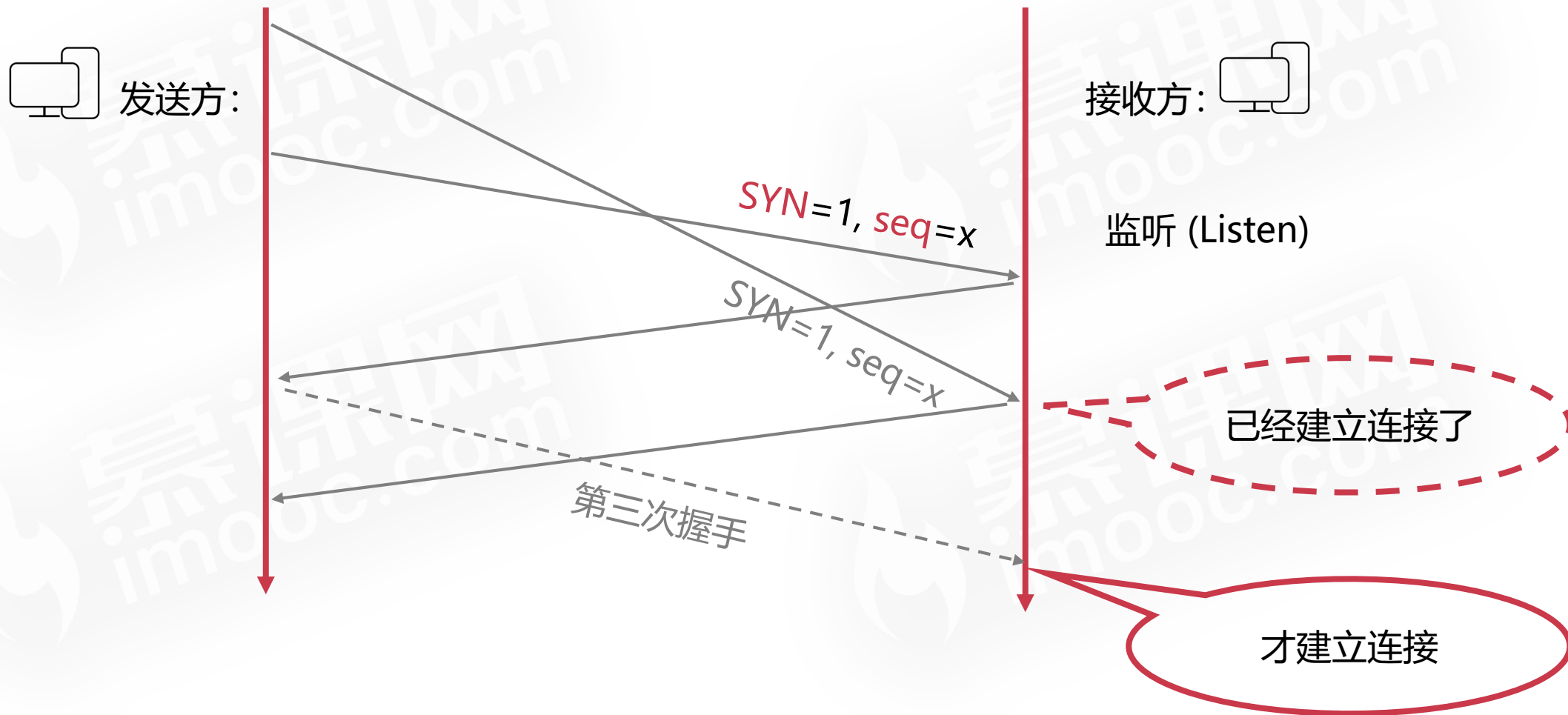
TCP连接的建立



TCP连接的建立



TCP连接的建立



TCP连接的建立

- ◆ 三次握手的过程
- ◆ 为什么需要第三次握手

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

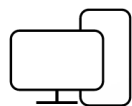
慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

TCP连接的释放



发送方:



接收方:

数据传输

$FIN=1, seq=u$

(FIN-WAIT-1)

$ACK=1, seq=v, ack=u+1$

关闭等待(CLOSE-WAIT)

(FIN-WAIT-2)

$FIN=1, ACK=1, seq=w, ack=u+1$

最后确认(LAST-ACK)

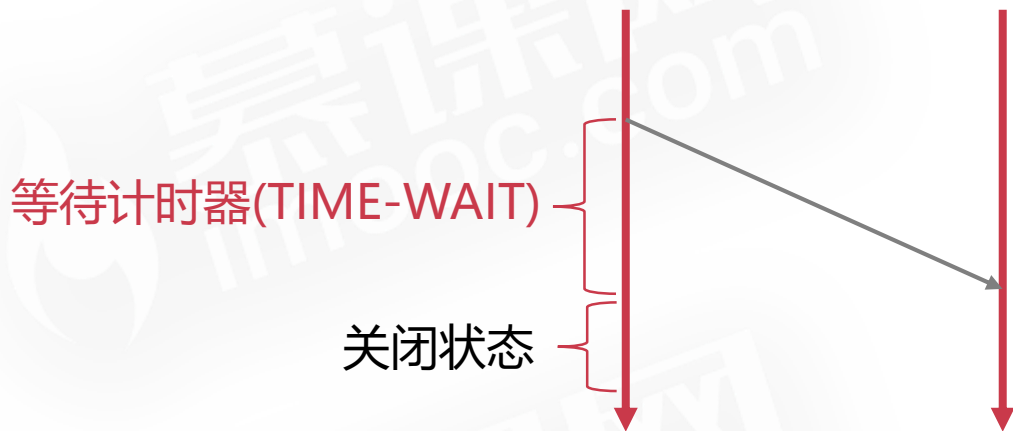
等待计时器(TIME-WAIT)

$ACK=1, seq=u+1, ack=w+1$

关闭状态

关闭状态

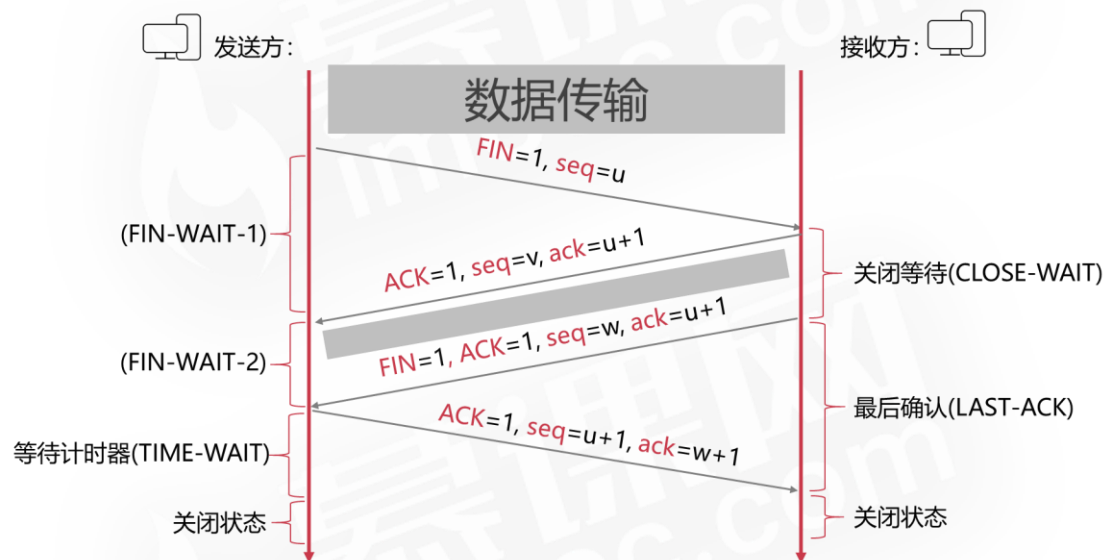
TCP连接的释放



2MSL

MSL(Max Segment Lifetime): 最长报文段寿命

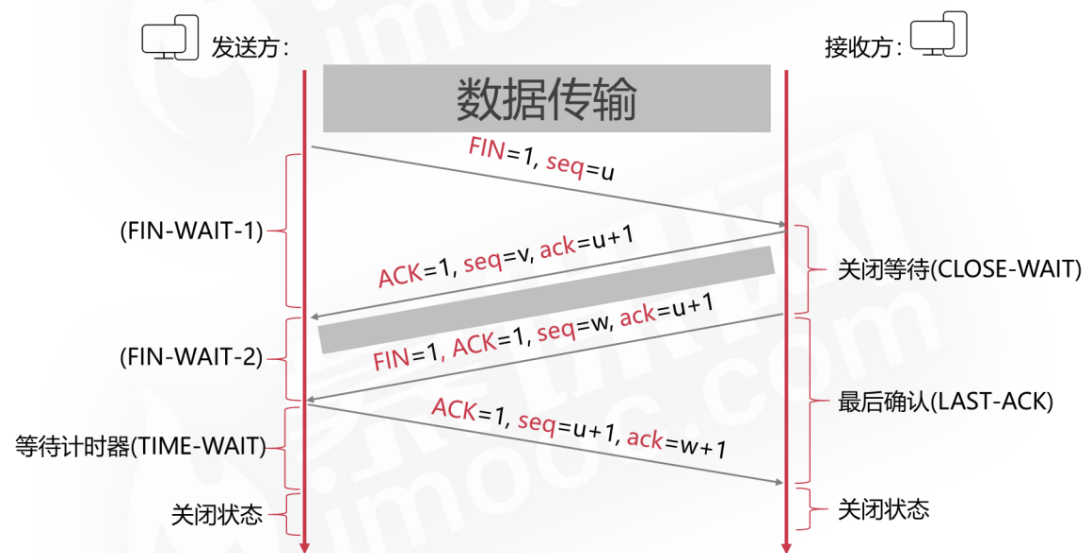
MSL建议设置为2分钟



TCP连接的释放

为什么需要等待2MSL?

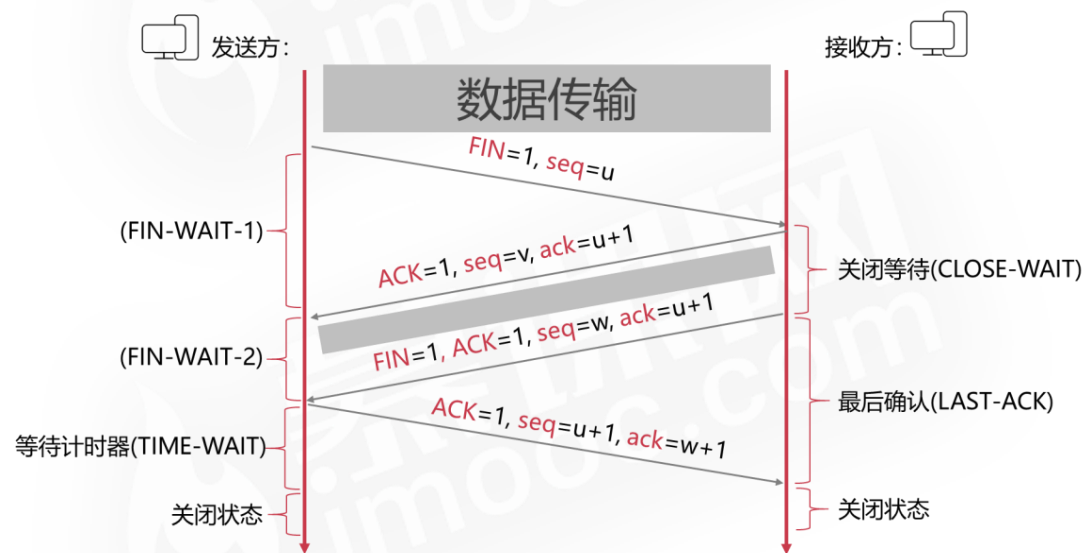
- ◆ 最后一个报文没有确认
- ◆ 确保发送方的ACK可以到达接收方
- ◆ 2MSL时间内没有收到，则接收方会重发



TCP连接的释放

为什么需要等待2MSL?

- ◆ 确保当前连接的所有报文都已经过期



TCP连接的释放

- ◆ 四次挥手的过程
- ◆ 等待计时器的作用



套接字与套接字编程

套接字与套接字编程

- ◆ 使用端口(Port)来标记不同的网络进程
- ◆ 端口(Port)使用16比特位表示(0~65535)

$\{ IP: Port \}$  套接字

套接字与套接字编程

- ◆ 套接字(Socket)是抽象概念，表示TCP连接的一端
- ◆ 通过套接字可以进行数据发送或接收

套接字与套接字编程

TCP

$= \{Socket1:Socket2\}$

$= \{\{ IP:Port \}\{ IP:Port \}\}$

TCP连接由两个套接字组成

套接字与套接字编程



套接字与套接字编程



套接字与套接字编程

网络套接字

VS

域套接字

套接字与套接字编程



域套接字文件

域套接字

套接字与套接字编程

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com

慕课网
imooc.com