

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

这三种可靠传输实现机制的基本原理并不仅限于数据链路层，可以应用到计算机网络体系结构的各层协议中。

希望同学们在学习时，不要把思维局限在数据链路层，而应放眼于整个网络体系结构。

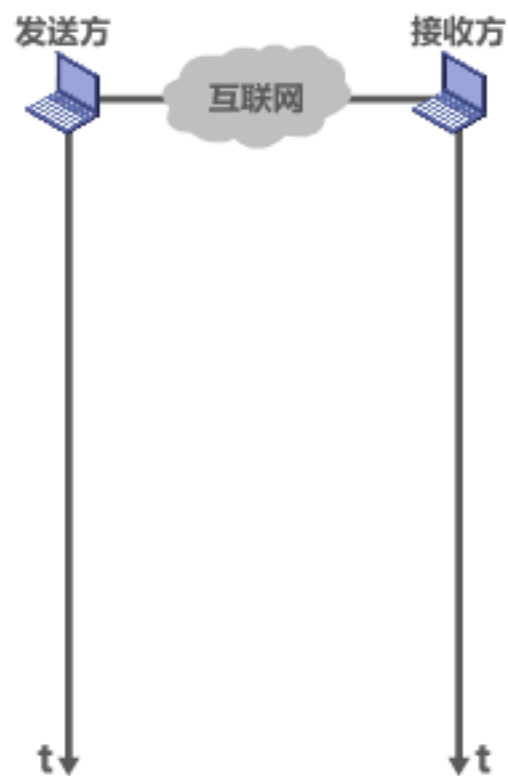


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

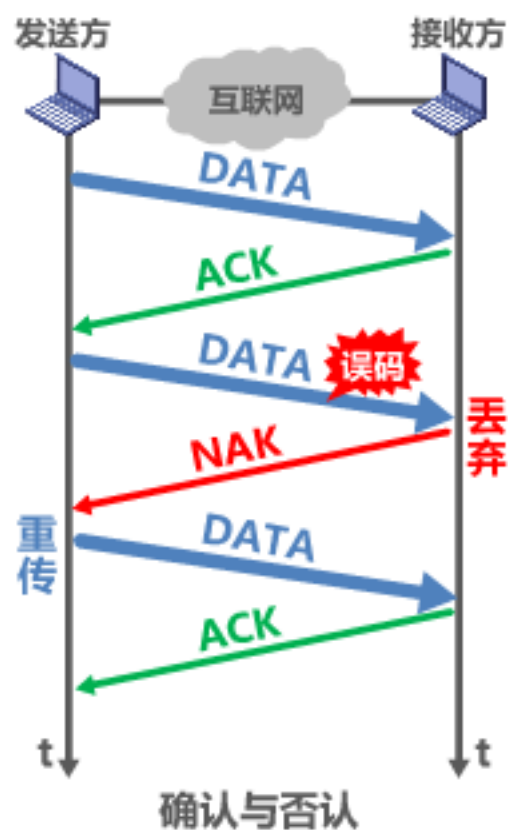


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

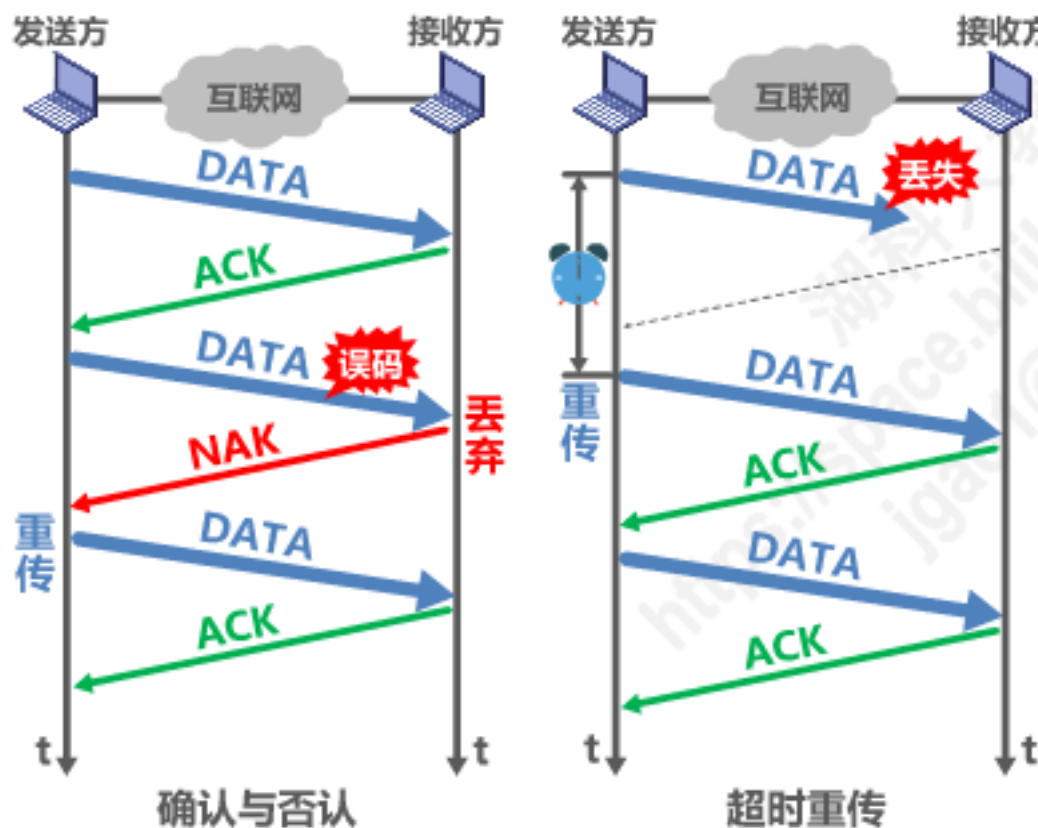


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR



接收方收不到数据分组，就不会发送ACK或NAK。如果不采取其他措施，发送方就会一直处于等待接收方ACK或NAK的状态。

为解决该问题，可以在发送方发送完一个数据分组时，启动一个**超时计时器**。若到了超时计时器所设置的**重传时间**而发送方仍收不到接收方的任何ACK或NAK，则重传原来的数据分组，这就叫做**超时重传**。

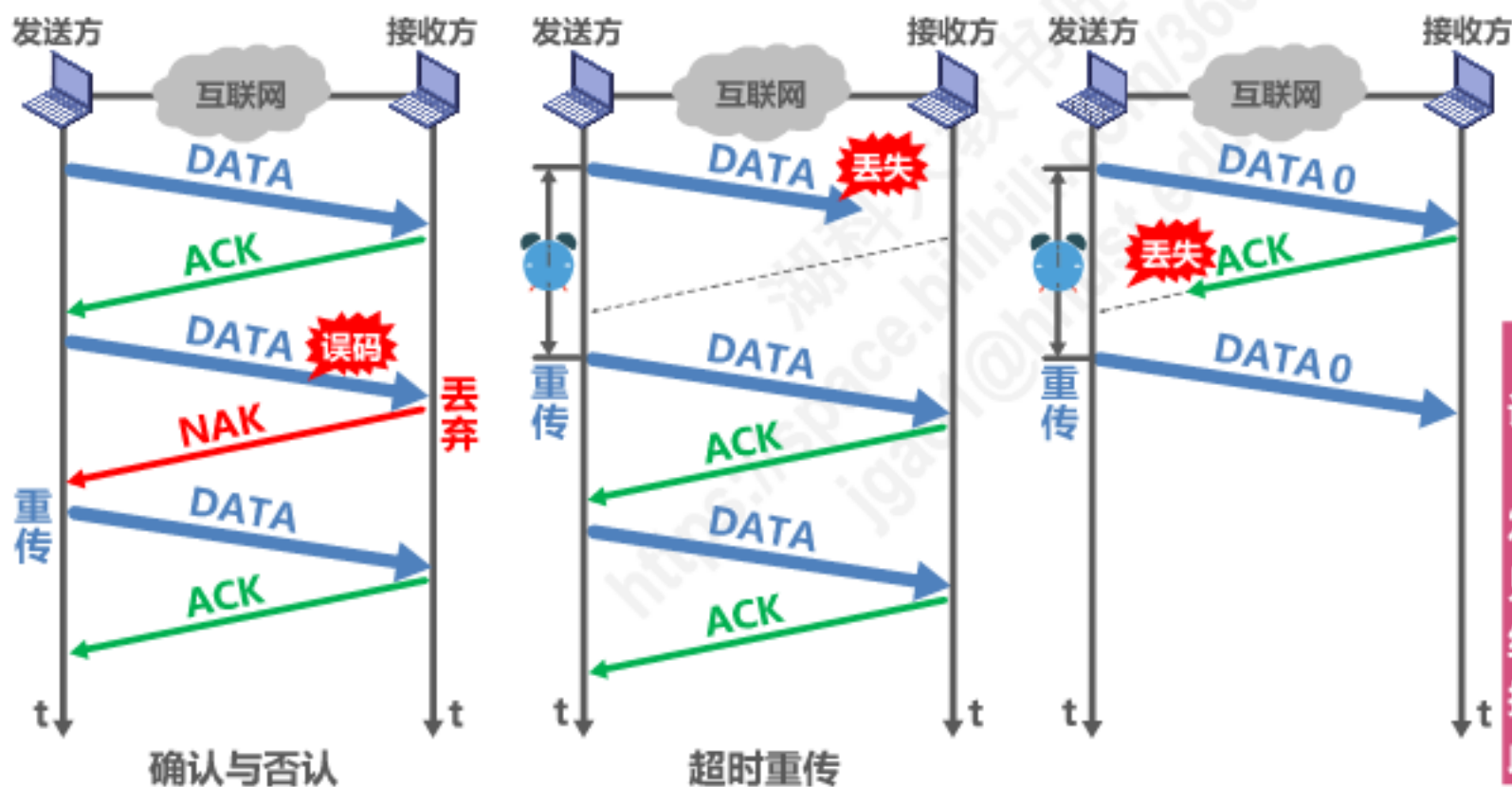
一般可将重传时间选为略大于“从发送方到接收方的平均往返时间”。

3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR



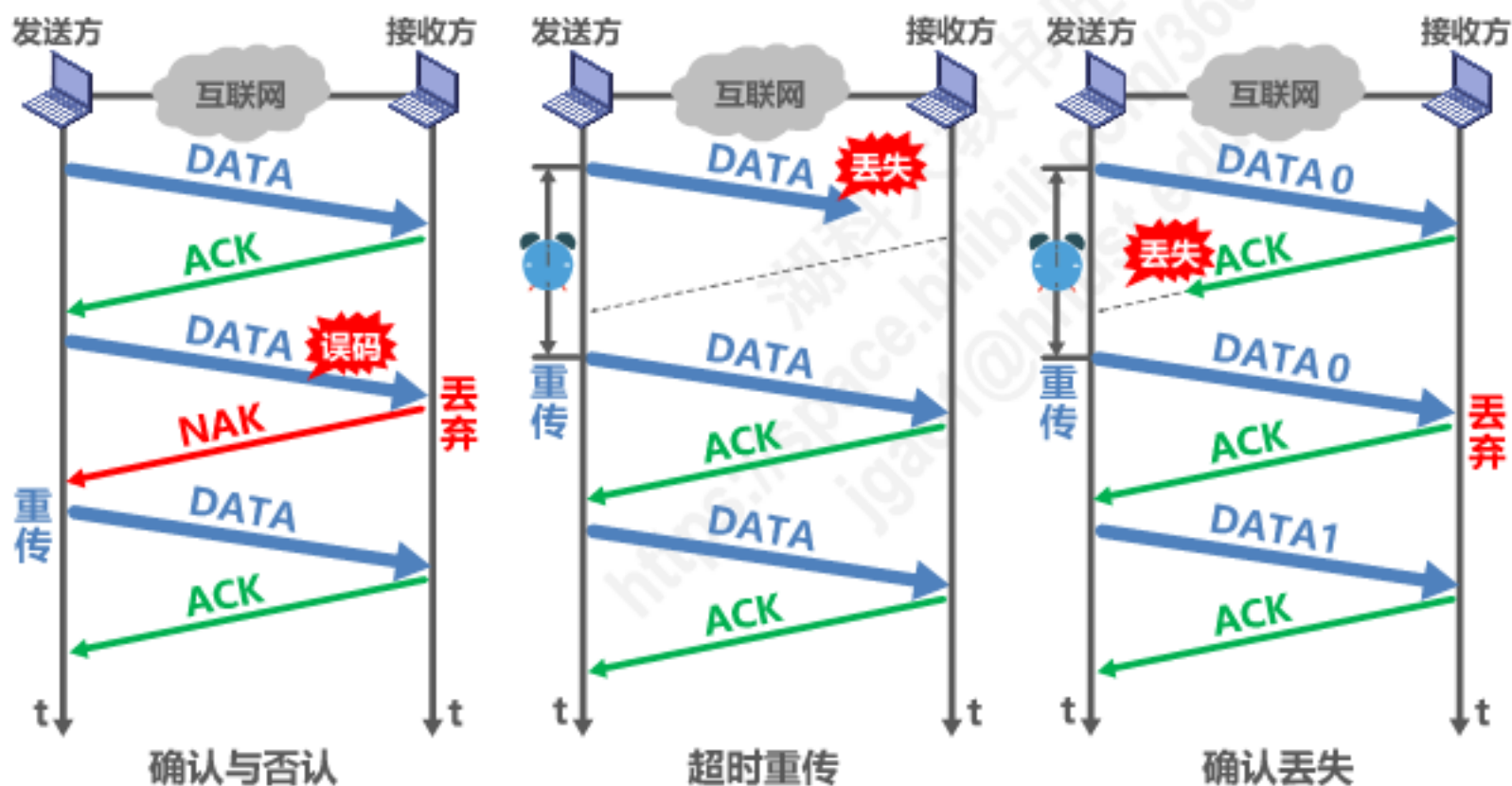
为避免分组重复这种传输错误，必须给每个分组带上序号。
对于停止-等待协议，由于每发送一个数据分组就停止等待，只要保证每发送一个新的数据分组，其发送序号与上次发送的数据分组的序号不同就可以了，因此用一个比特来编号就够了。

3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR



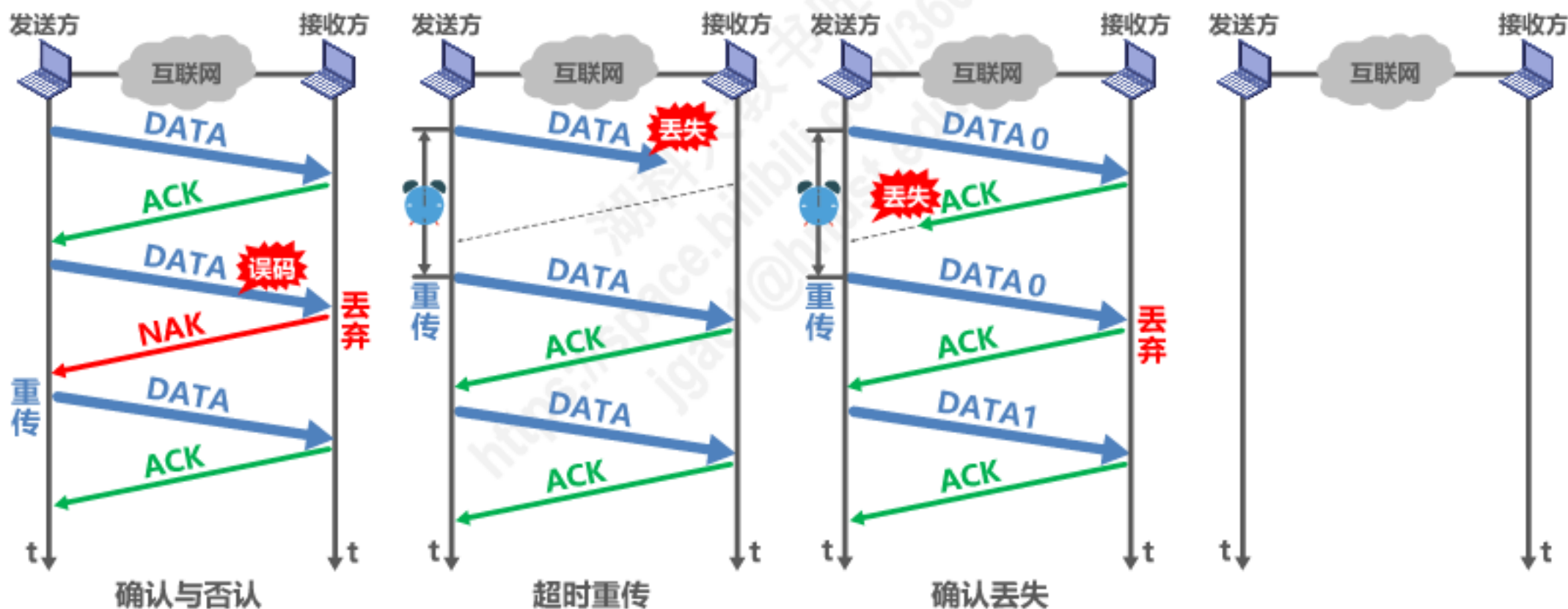
确认分组是否
需要编号?

3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

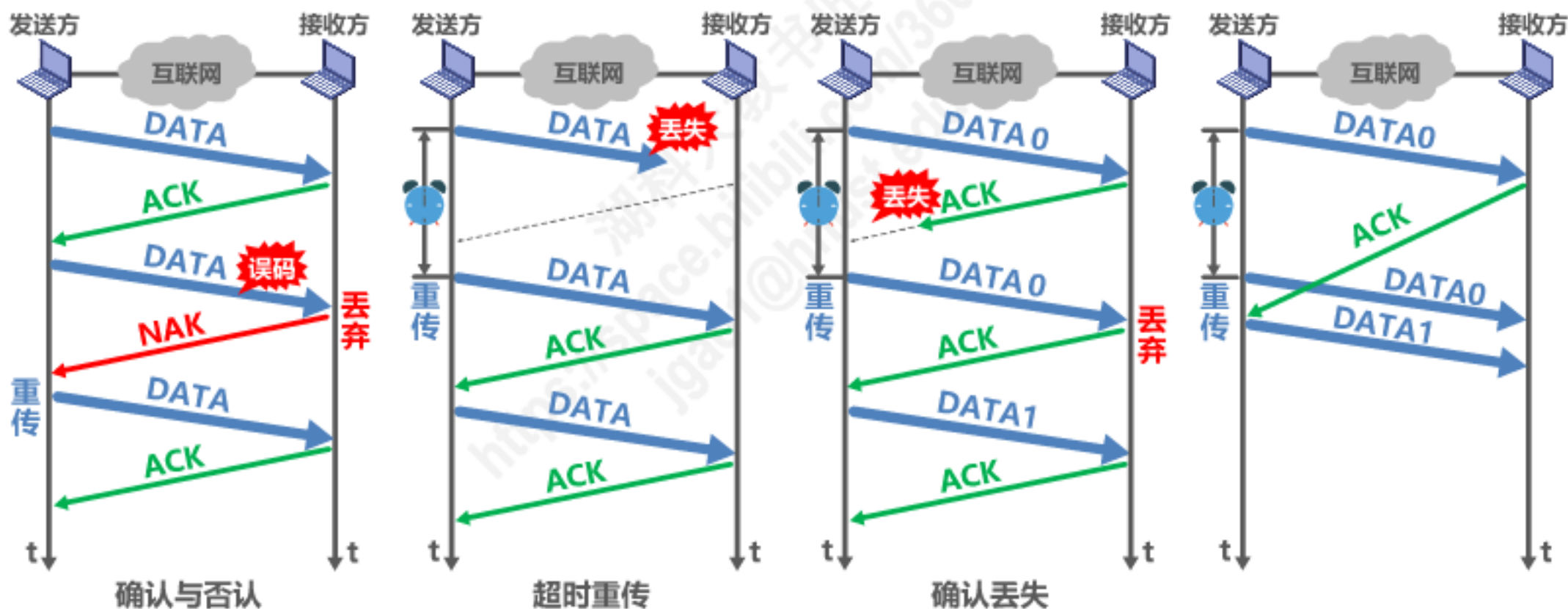


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

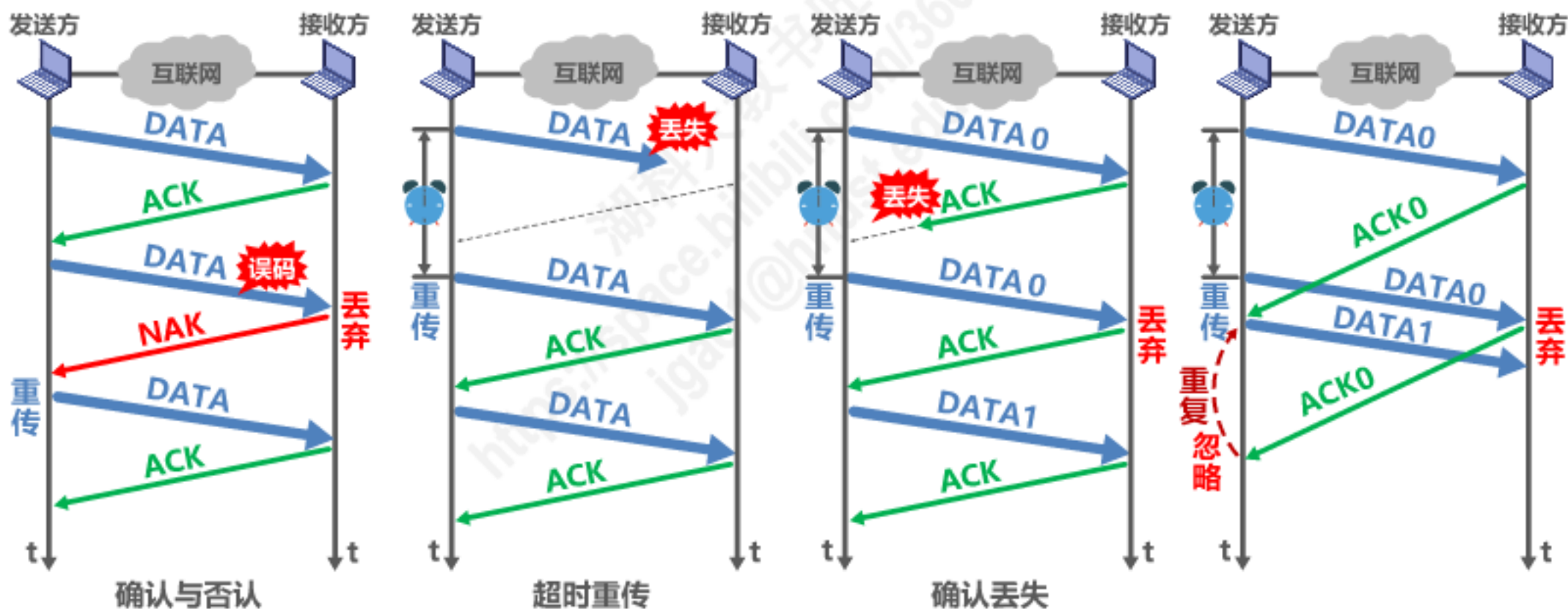


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

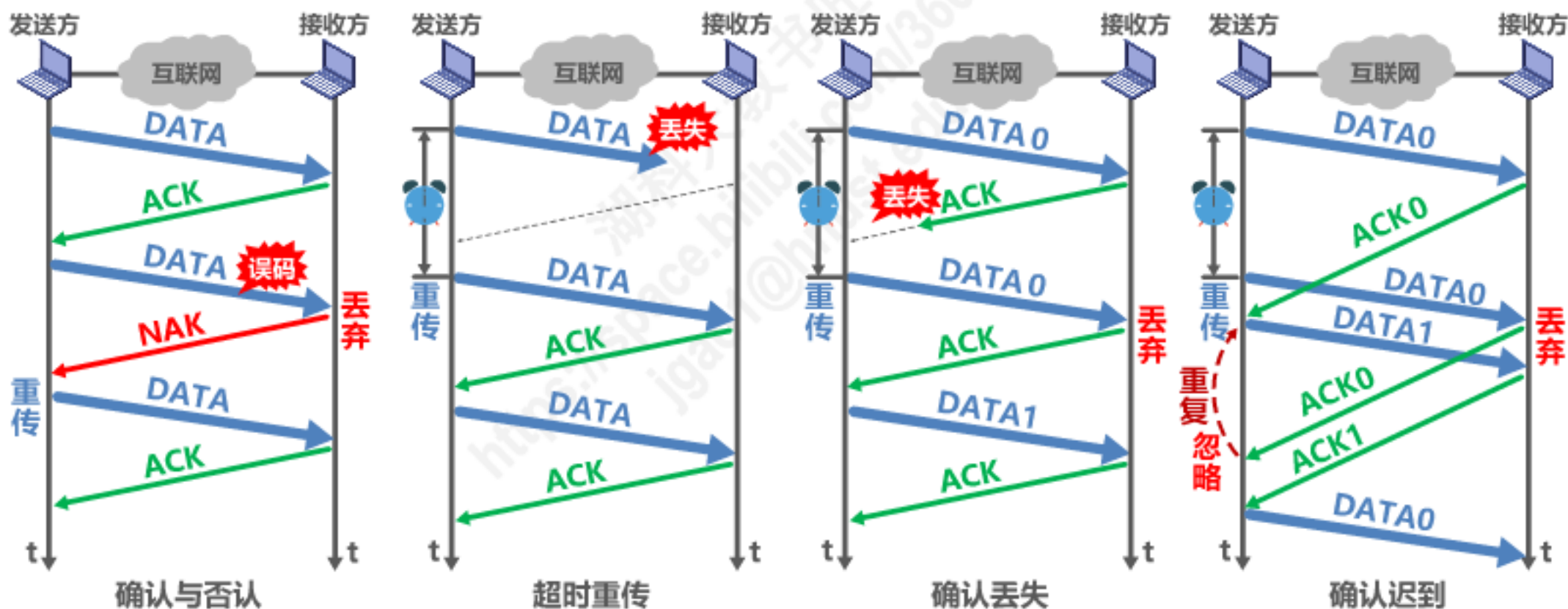


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

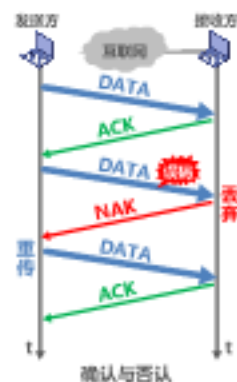
回退N帧协议GBN

选择重传协议SR

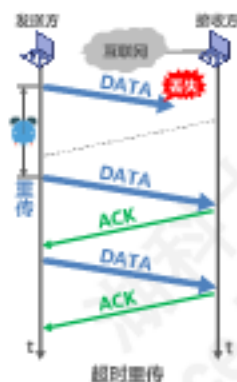


3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

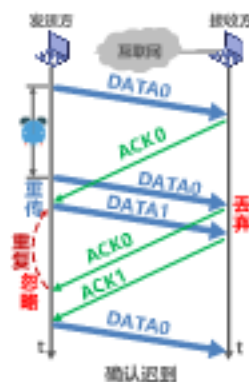
停止-等待协议SW



回退N帧协议GBN



选择重传协议SR



【注意事项】

- 接收端检测到数据分组有错误时，将其丢弃并等待发送方的超时重传。但对于误码率较高的点对点链路，为使发送方**尽早重传**，也可**给发送方发送NAK分组**。
- 为了让接收方能够判断所收到的数据分组是否是重复的，需要给**数据分组编号**。由于停止-等待协议的停等特性，**只需1个比特编号**就够了，即编号0和1。
- 为了让发送方能够判断所收到的ACK分组是否是重复的，需要给**ACK分组编号**，所用比特数量与**数据分组编号所用比特数量一样**。数据链路层一般不会出现ACK分组迟到的情况，因此在**数据链路层实现停止-等待协议可以不用给ACK分组编号**。
- 超时计时器设置的**重传时间**应仔细选择。一般可将重传时间选为**略大于“从发送方到接收方的平均往返时间”**。
 - ☐ 在数据链路层点对点的往返时间比较确定，重传时间比较好设定。
 - ☐ 然而在运输层，由于端到端往返时间非常不确定，设置合适的重传时间有时并不容易。

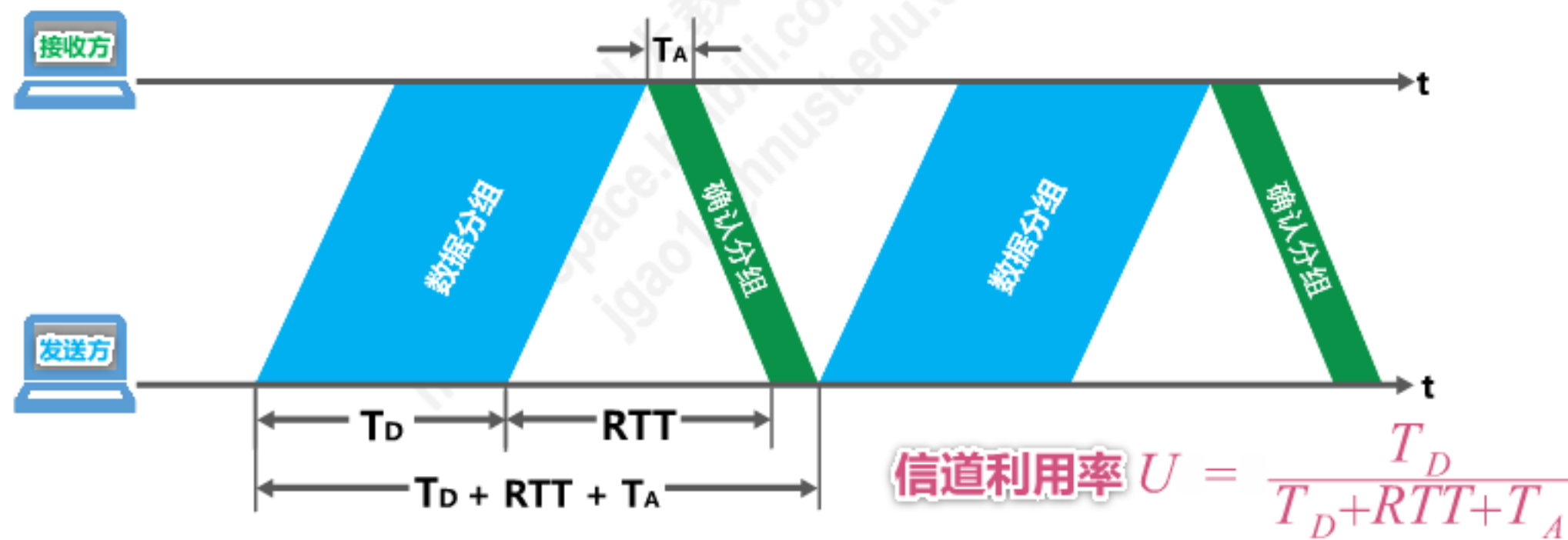
3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

停止-等待协议的信道利用率



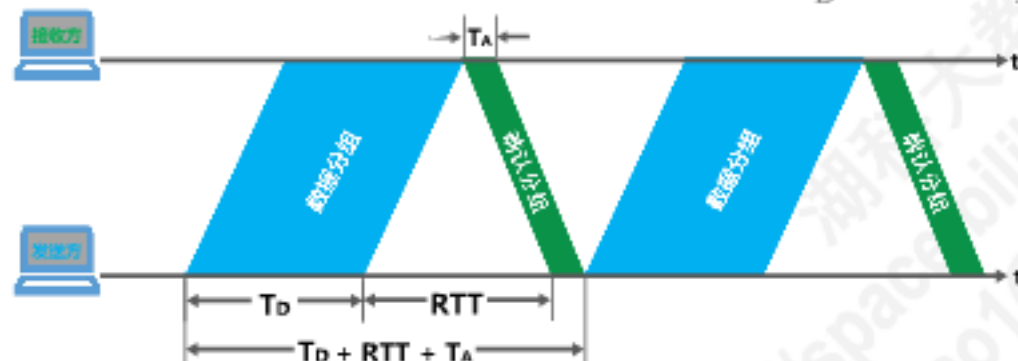
3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

停止-等待协议的信道利用率 $U = \frac{T_D}{T_D + RTT + T_A}$



假设信道长度2000km，数据分组长度1500B，发送速率10Mbit/s。
(忽略 T_A ，因为 T_A 一般都远小于 T_D)

$$U \approx \frac{T_D}{T_D + RTT} = \frac{\frac{1500 \times 8 \text{ bit}}{10 \times 10^6 \text{ bit/s}}}{\frac{1500 \times 8 \text{ bit}}{10 \times 10^6 \text{ bit/s}} + \frac{2000 \times 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}} \times 2} = \frac{1.2\text{ms}}{1.2\text{ms} + 20\text{ms}} \approx 5.66\%$$

若提高发送速率到100Mb/s

$$U \approx \frac{T_D}{T_D + RTT} = \frac{\frac{1500 \times 8 \text{ bit}}{100 \times 10^6 \text{ bit/s}}}{\frac{1500 \times 8 \text{ bit}}{100 \times 10^6 \text{ bit/s}} + \frac{2000 \times 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}} \times 2} = \frac{0.12\text{ms}}{0.12\text{ms} + 20\text{ms}} \approx 0.6\%$$

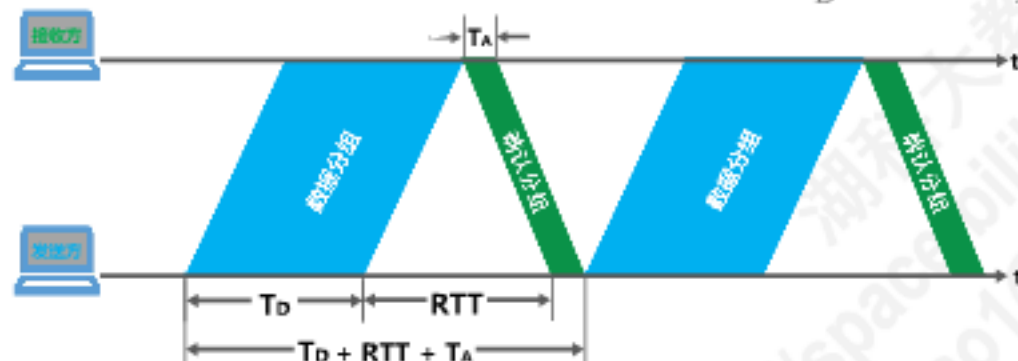
3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

停止-等待协议的信道利用率 $U = \frac{T_D}{T_D + RTT + T_A}$



假设信道长度2000km，数据分组长度1500B，发送速率10Mbit/s。
(忽略 T_A ，因为 T_A 一般都远小于 T_D)

$$U \approx \frac{T_D}{T_D + RTT} = \frac{\frac{1500 \times 8 \text{ bit}}{10 \times 10^6 \text{ bit/s}}}{\frac{1500 \times 8 \text{ bit}}{10 \times 10^6 \text{ bit/s}} + \frac{2000 \times 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}} \times 2} = \frac{1.2\text{ms}}{1.2\text{ms} + 20\text{ms}} \approx 5.66\%$$

若提高发送速率到100Mb/s

$$U \approx \frac{T_D}{T_D + RTT} = \frac{\frac{1500 \times 8 \text{ bit}}{100 \times 10^6 \text{ bit/s}}}{\frac{1500 \times 8 \text{ bit}}{100 \times 10^6 \text{ bit/s}} + \frac{2000 \times 10^3 \text{ m}}{2 \times 10^8 \text{ m/s}} \times 2} = \frac{0.12\text{ms}}{0.12\text{ms} + 20\text{ms}} \approx 0.6\%$$

当往返时延RTT远大于数据帧发送时延 T_D 时（例如使用卫星链路），信道利用率非常低。

若出现重传，则对于传送有用的数据信息来说，信道利用率还要降低。

为了克服停止-等待协议信道利用率很低的缺点，就产生了另外两种协议，即后退N帧协议GBN和选择重传协议SR。

3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

停止-等待协议SW

回退N帧协议GBN

选择重传协议SR

【2018年 题36】主机甲采用停-等协议向主机乙发送数据，数据传输速率是3kbps，单向传播延时是200ms，忽略确认帧的传输延时。当信道利用率等于40%时，数据帧的长度为 **D**

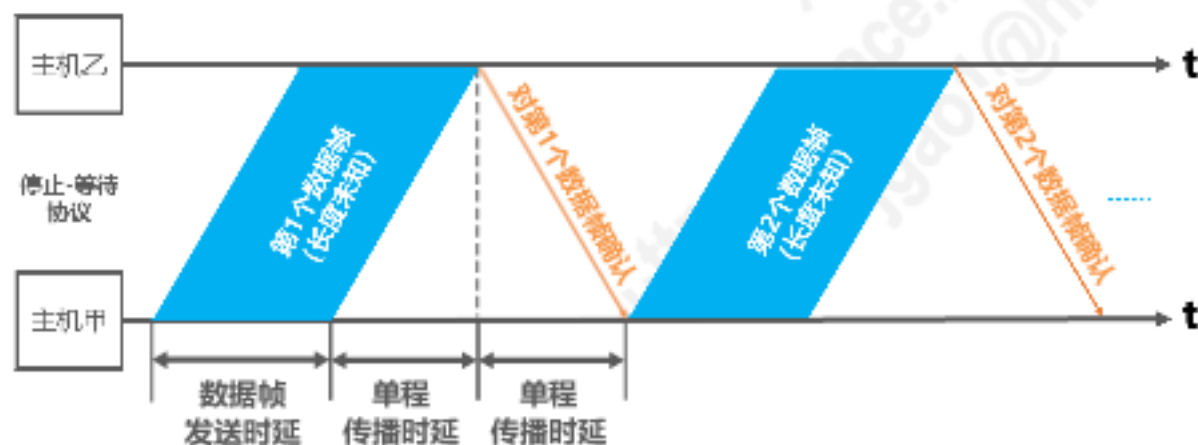
A. 240比特

B. 400比特

C. 480比特

D. 800比特

【解析】



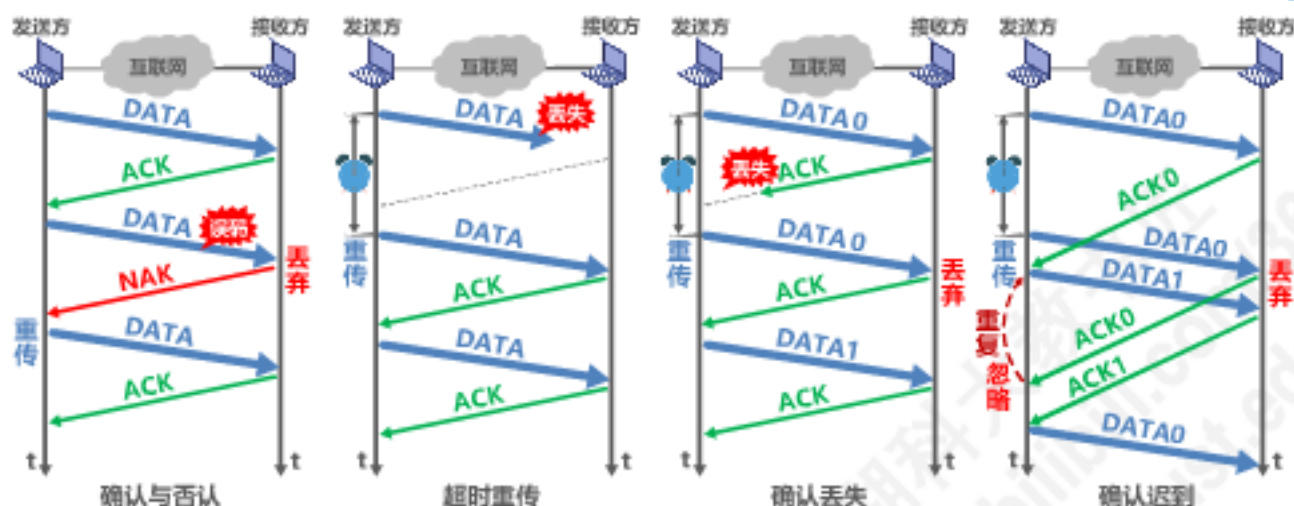
$$\text{信道利用率} = \frac{\text{数据帧发送时延}}{\text{数据帧发送时延} + \text{端到端往返时延}}$$

设数据帧长度为x比特

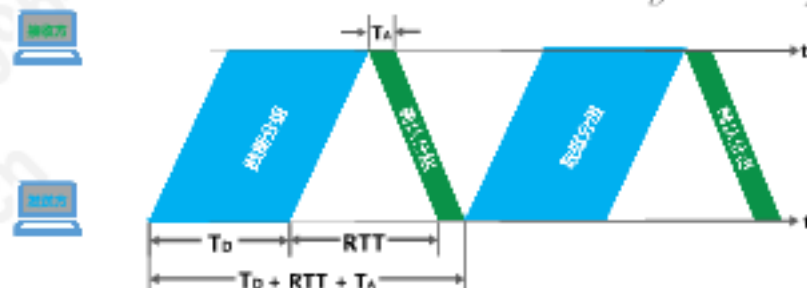
$$40\% = \frac{\frac{x}{3k} \frac{b}{s}}{\frac{x}{3k} \frac{b}{s} + 2 \times 200 \text{ ms}}$$

解得 x = 800比特

3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)

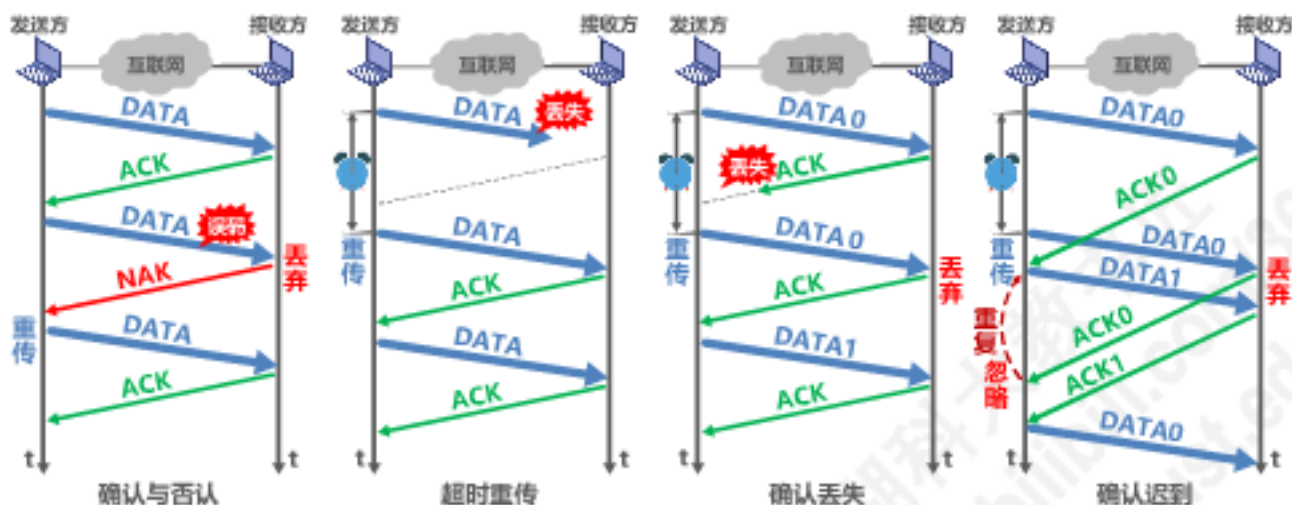


停止-等待协议的信道利用率 $U = \frac{T_D}{T_D + RTT + T_A}$

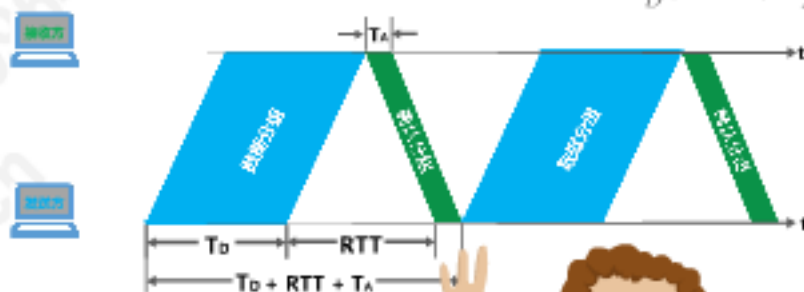


- 接收端检测到数据分组有错误时，将其丢弃并等待发送方的超时重传。但对于误码率较高的点对点链路，为使发送方**尽早重传**，也可**给发送方发送NAK分组**。
- 为了让接收方能够判断所收到的数据分组是否是重复的，需要给**数据分组编号**。由于停止-等待协议的停等特性，**只需1个比特编号**就够了，即编号0和1。
- 为了让发送方能够判断所收到的ACK分组是否是重复的，需要给**ACK分组编号**，所用比特数量与数据分组编号所用比特数量一样。数据链路层一般不会出现ACK分组迟到的情况，因此在**数据链路层实现停止-等待协议可以不用给ACK分组编号**。
- 超时计时器设置的**重传时间**应仔细选择。一般可将重传时间选为略大于“从发送方到接收方的平均往返时间”。
 - ☐ 在数据链路层点对点的往返时间比较确定，重传时间比较好设定。
 - ☐ 然而在运输层，由于端到端往返时间非常不确定，设置合适的重传时间有时并不容易。
- 当往返时延RTT远大于数据帧发送时延 T_D 时（例如使用卫星链路），信道利用率非常低。**
- 若出现重传，则对于传送有用的数据信息来说，信道利用率还要降低。
- 为了克服停止-等待协议信道利用率很低的缺点，就产生了另外两种协议，即后退N帧协议GBN和选择重传协议SR。

3.4.2 可靠传输的实现机制 —— 停止-等待协议SW(Stop-and-Wait)



停止-等待协议的信道利用率 $U = \frac{T_D}{T_D + RTT + T_A}$



- 接收端检测到数据分组有错误时，将其丢弃并等待发送方的超时重传。但对于误码率较高的点对点链路，为使发送方**尽早重传**，也可**给发送方发送NAK分组**。
- 为了让接收方能够判断所收到的数据分组是否是重复的，需要给**数据分组编号**。由于停止-等待协议的停等特性，**只需1个比特编号**就够了，即编号0和1。
- 为了让发送方能够判断所收到的ACK分组是否是重复的，需要给**ACK分组编号**，所用比特数量与数据分组编号所用比特数量一样。数据链路层一般不会出现ACK分组迟到的情况，因此在**数据链路层实现停止-等待协议可以不用给ACK分组编号**。
- 超时计时器设置的**重传时间**应仔细选择。一般可将重传时间选为略大于“从发送方到接收方的平均往返时间”。
 - ☐ 在数据链路层点对点的往返时间比较确定，重传时间比较好设定。
 - ☐ 然而在运输层，由于端到端往返时间非常不确定，设置合适的重传时间有时并不容易。
- 当往返时延RTT远大于数据帧发送时延 T_D 时（例如使用卫星链路），信道利用率非常低。
- 若出现重传，则对于传送有用的数据信息来说，信道利用率还要降低。
- 为了克服停止-等待协议信道利用率很低的缺点，就产生了另外两种协议，即后退N帧协议GBN和选择重传协议SR。

自动请求重传ARQ
(Automatic Repeat request)

