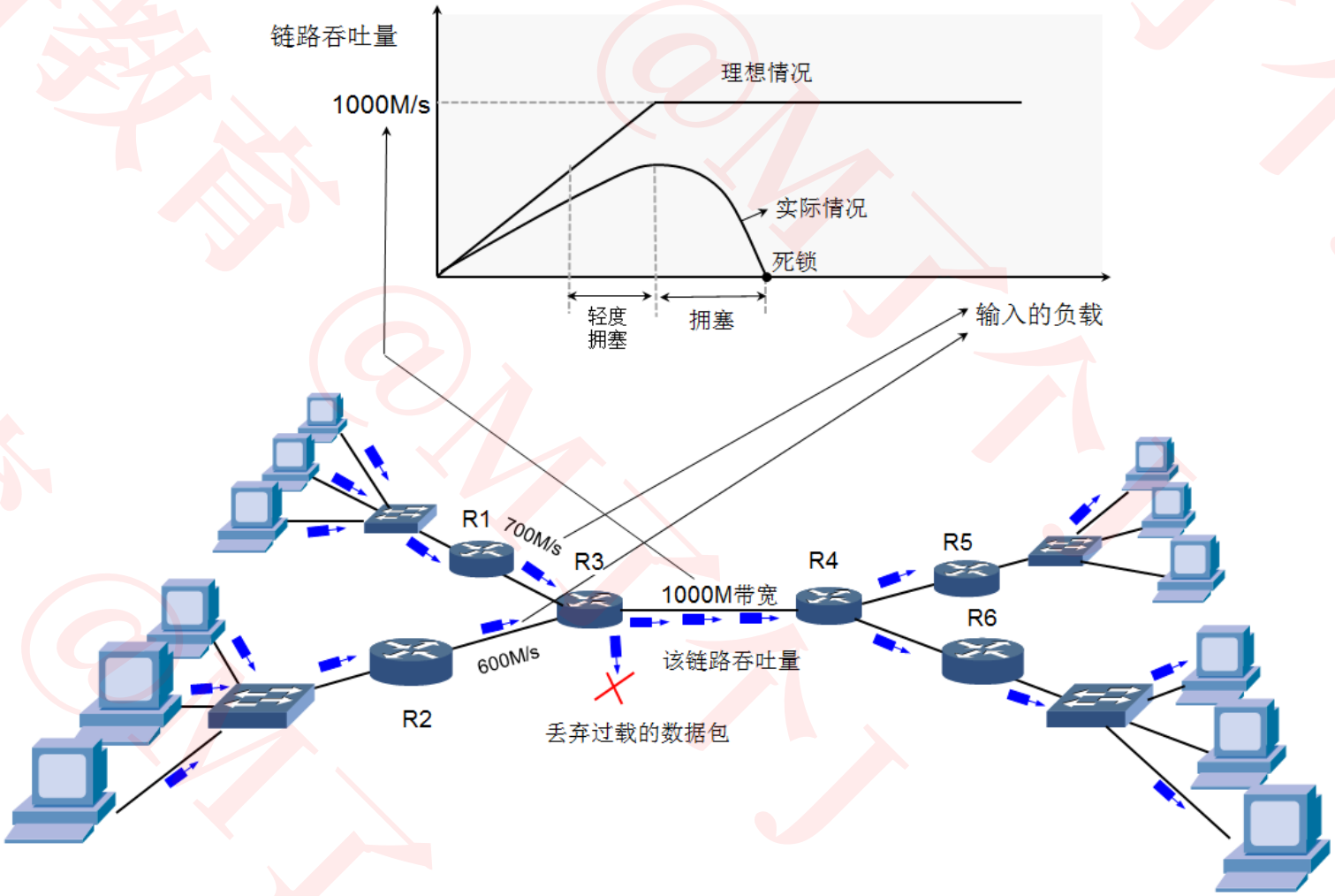
TCP – 拥塞控制

- 拥塞控制
  - 防止过多的数据注入到网络中
  - 避免网络中的路由器或链路过载

- 拥塞控制是一个全局性的过程
  - 涉及到所有的主机、路由器
  - 以及与降低网络传输性能有关的所有因素
  - 是大家共同努力的结果

- 相比而言，流量控制是点对点通信的控制

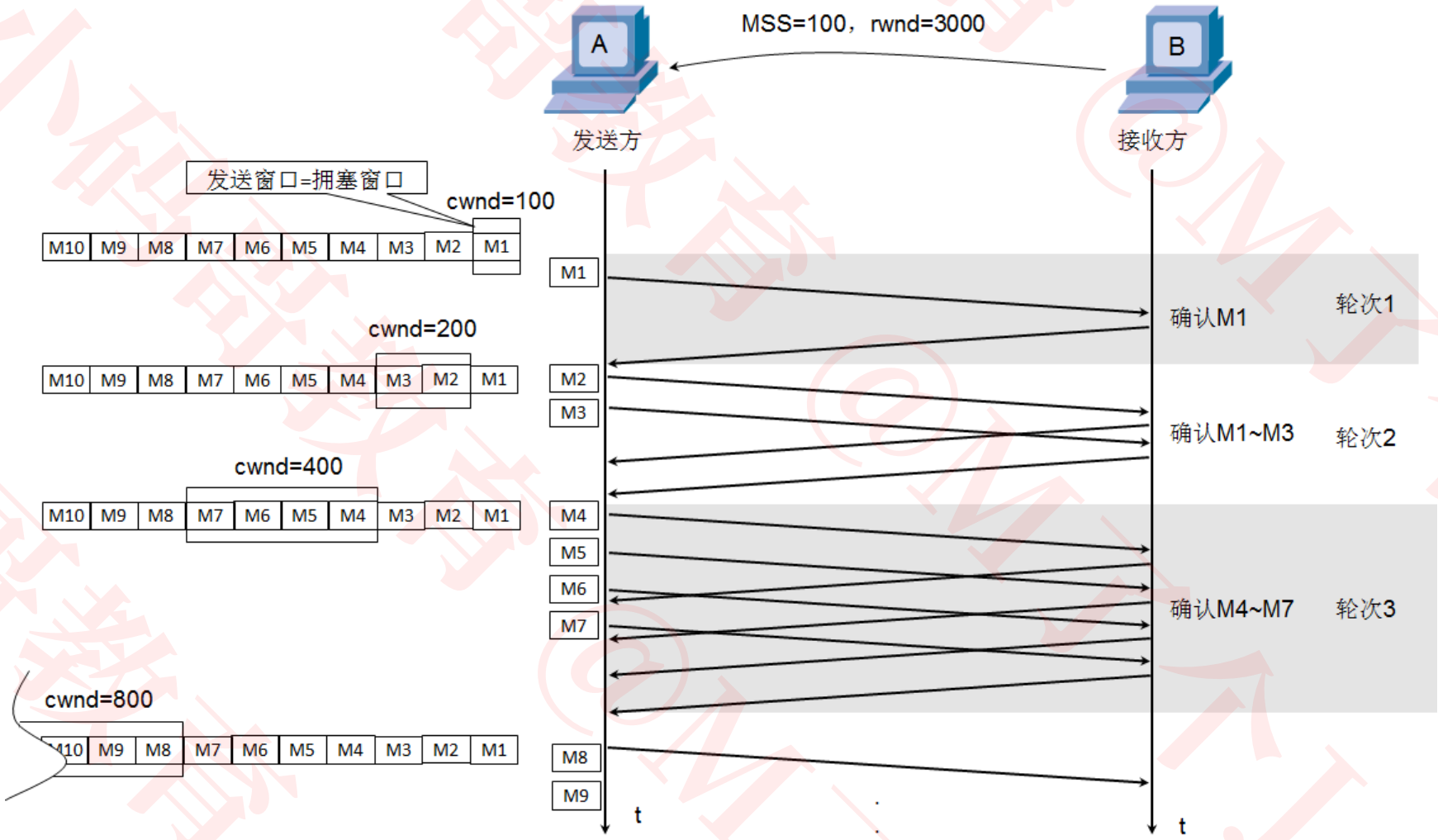


TCP – 拥塞控制 – 方法

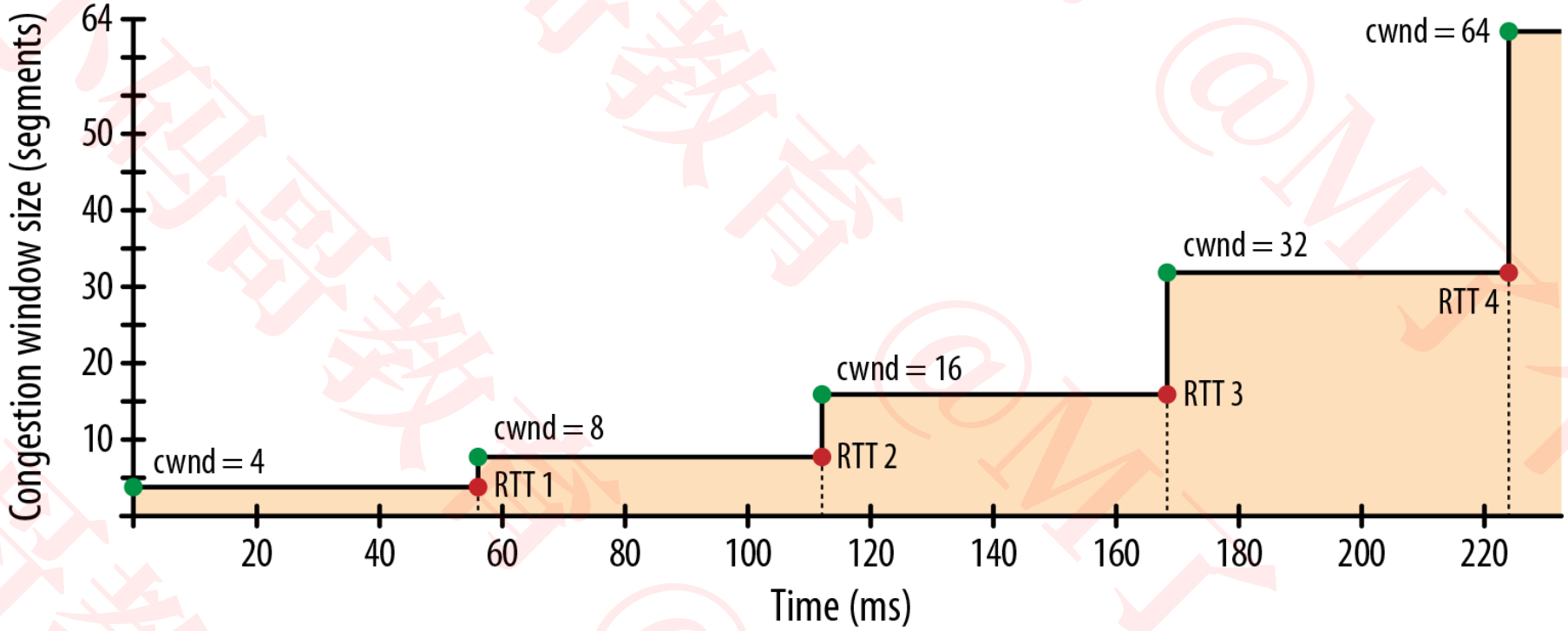
- 慢开始（slow start，慢启动）
- 拥塞避免（congestion avoidance）
- 快速重传（fast retransmit）
- 快速恢复（fast recovery）

- 几个缩写
  - MSS（Maximum Segment Size）：每个段最大的数据部分大小
    - ✓在建立连接时确定
  - cwnd（congestion window）：拥塞窗口
  - rwnd（receive window）：接收窗口
  - swnd（send window）：发送窗口
    - ✓swnd = min(cwnd, rwnd)

TCP – 拥塞控制 – 慢开始

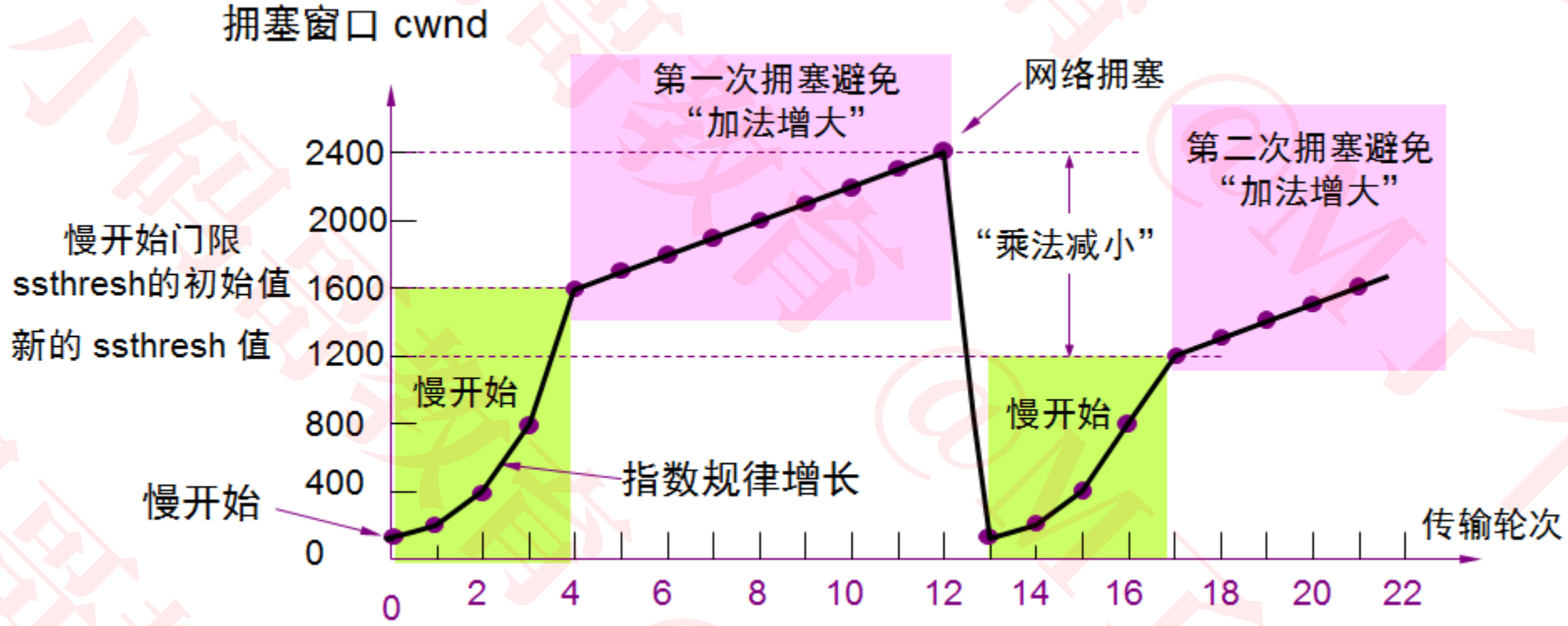

A
B
MSS=100，rwnd=3000
发送方
接收方
发送窗口=拥塞窗口
cwnd=100
M10 M9 M8 M7 M6 M5 M4 M3 M2 M1
cwnd=200
M10 M9 M8 M7 M6 M5 M4 M3 M2 M1
cwnd=400
M10 M9 M8 M7 M6 M5 M4 M3 M2 M1
cwnd=800
M10 M9 M8 M7 M6 M5 M4 M3 M2 M1
M1
M2
M3
M4
M5
M6
M7
M8
M9
确认M1
轮次1
确认M1~M3
轮次2
确认M4~M7
轮次3
t
t

TCP – 拥塞控制 – 慢开始



- cwnd的初始值比较小，然后随着数据包被接收方确认（收到一个ACK）
  - cwnd就成倍增长（指数级）

TCP – 拥塞控制 – 拥塞避免



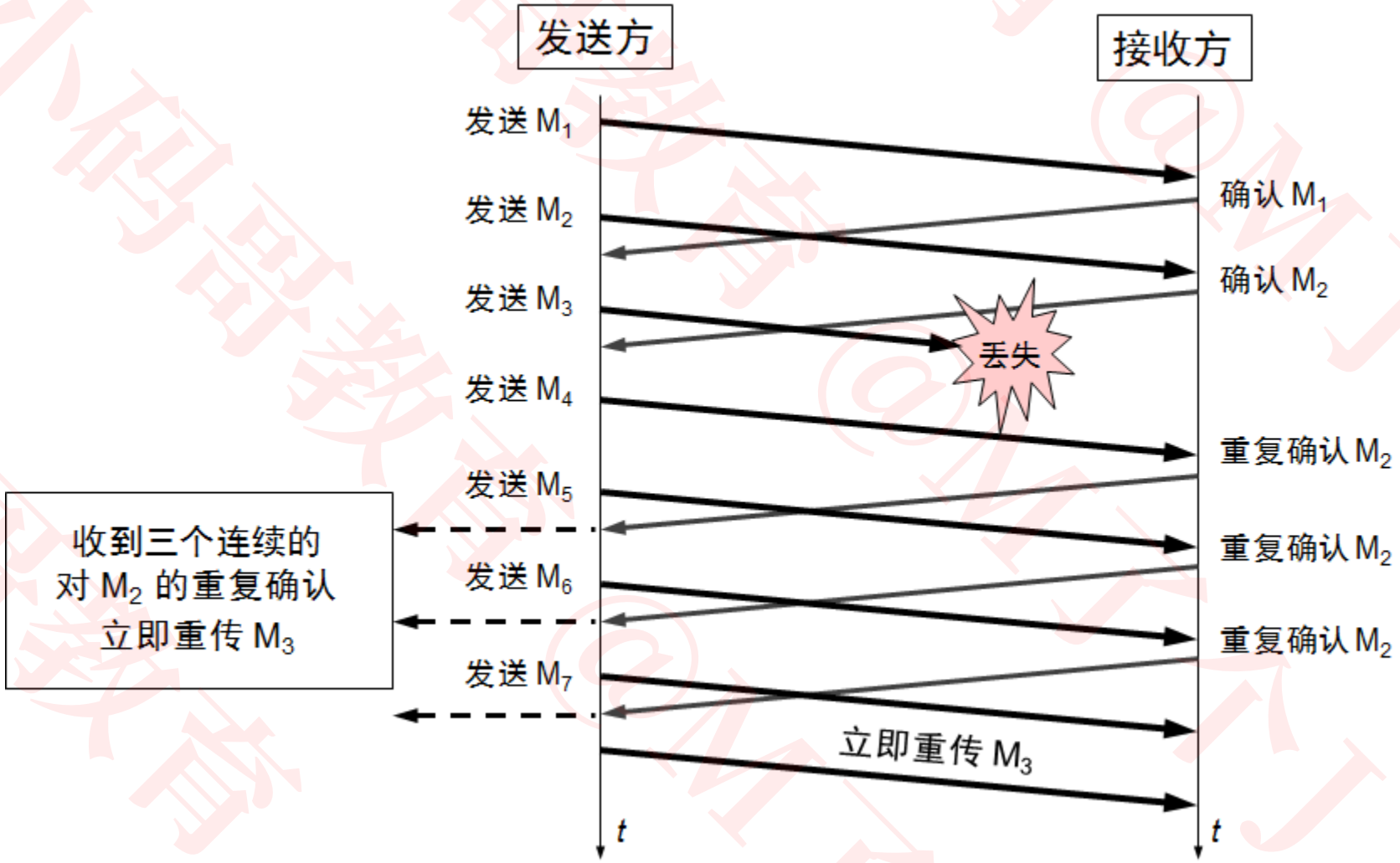
- ssthresh（slow start threshold）：慢开始阈值，cwnd达到阈值后，以线性方式增加
- 拥塞避免（加法增大）：拥塞窗口缓慢增大，以防止网络过早出现拥塞
- 乘法减小：只要网络出现拥塞，把ssthresh减为拥塞峰值的一半，同时执行慢开始算法（cwnd又恢复到初始值）
- 当网络出现频繁拥塞时，ssthresh值就下降的很快

TCP – 拥塞控制 – 快重传

- 接收方
  - 每收到一个失序的分组后就立即发出重复确认
  - 使发送方及时知道有分组没有到达
  - 而不要等待自己发送数据时才进行确认

- 发送方
  - 只要连续收到三个重复确认（总共4个相同的确认），就应当立即重传对方尚未收到的报文段
  - 而不必继续等待重传计时器到期后再重传

TCP – 拥塞控制 – 快重传

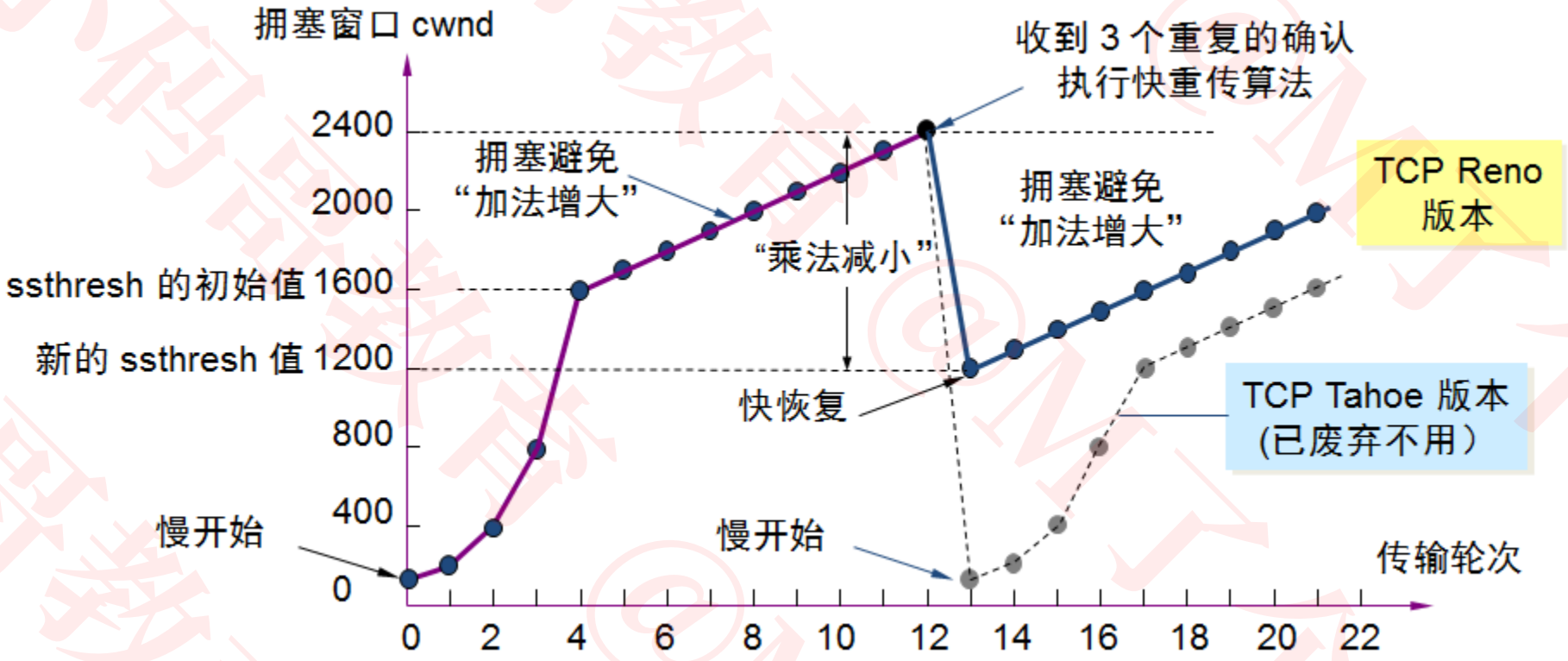

发送方
接收方
发送 M_1
发送 M_2
发送 M_3
发送 M_4
发送 M_5
发送 M_6
发送 M_7
确认 M_1
确认 M_2
丢失
重复确认 M_2
重复确认 M_2
重复确认 M_2
收到三个连续的
对 M_2 的重复确认
立即重传 M_3
立即重传 M_3
t
t

TCP – 拥塞控制 – 快恢复

- 当发送方连续收到三个重复确认，说明网络出现拥塞
  - 就执行“乘法减小”算法，把ssthresh减为拥塞峰值的一半

- 与慢开始不同之处是现在不执行慢开始算法，即cwnd现在不恢复到初始值
  - 而是把cwnd值设置为新的ssthresh值（减小后的值）
  - 然后开始执行拥塞避免算法（“加法增大”），使拥塞窗口缓慢地线性增大

TCP – 拥塞控制 – 快重传 + 快恢复



TCP – 拥塞控制 – 发送窗口的最大值

- 发送窗口的最大值：swnd = min(cwnd, rwnd)
- 当rwnd < cwnd时，是接收方的接收能力限制发送窗口的最大值
- 当cwnd < rwnd时，则是网络的拥塞限制发送窗口的最大值