

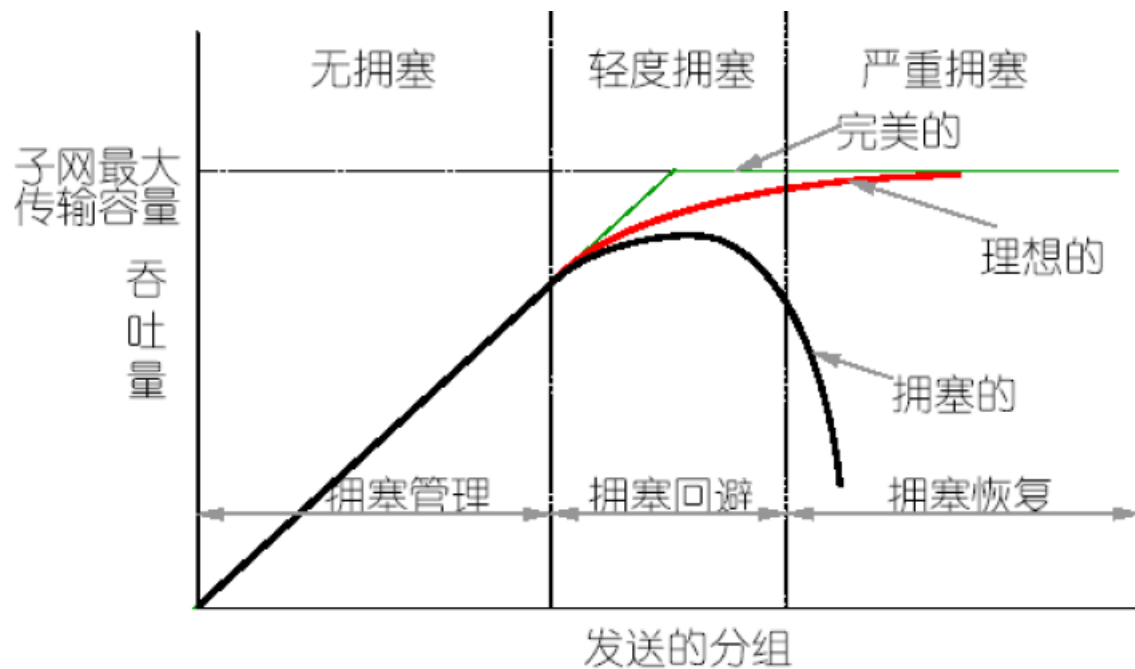
第五章 网络层

拥塞控制



什么是拥塞？

当一个子网或子网的一部分出现太多分组的时候，网络的性能急剧下降，这就是拥塞（**Congestion**）



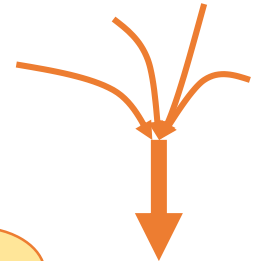


导致拥塞的因素



输入流量大于输出线路的容量

- 如：几根输入线路同时向一个输出线路转发分组
- 增加内存可以提高性能到某种程度



如果增加内存到无穷？



慢速的处理器也可能引起拥塞，如线路容量充足，但处理器来不及处理



线路容量和处理器能力需要平衡

拥塞控制 VS 流控

拥塞控制 (Congestion control)	流控 (Flow control)
任务：确保子网能够承载所到达	只与特定的发送方和接收方之间
一台主机会收到 “slow down” 消息，不管是因为接收者无法处理	分组还是网络无法处理
路由器，存储转发的过程等方方面面的问题	确保一个快速的发送方不会持续地以超过接受方接收能力的速率传输数据



拥塞控制的通用原则

开环 (Open loop)

- 试图用良好的设计来解决问题，本质是从一开始就保证问题不会发生
- 开环决策制定不考虑网络的当前状态

不是不明白，这世界变换太快！

闭环 (Closed loop)

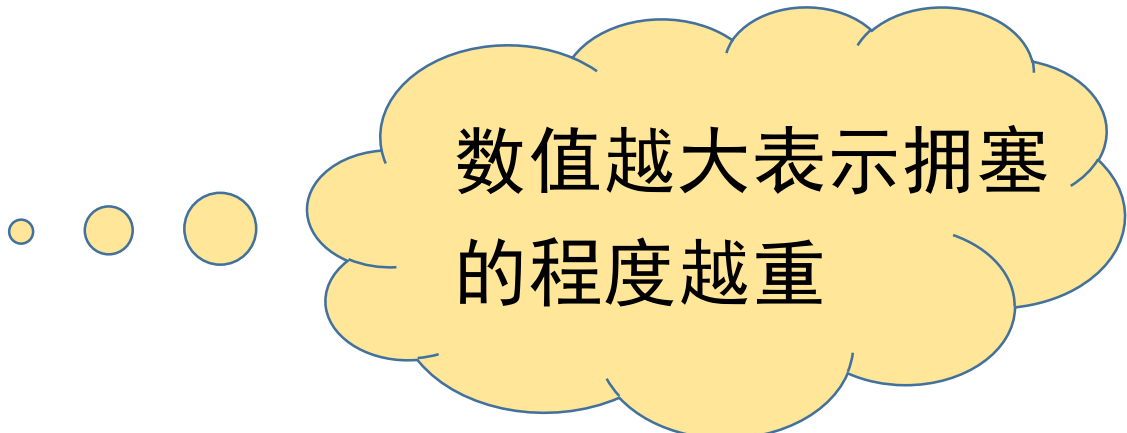
建立在反馈环路的概念上，分三个步骤：

- 监视系统，**检测**何时何地发生了拥塞
- 把这些信息**传递**到能够采取行动的地方
- 调整系统的运行，以**改正**问题

怎么知道拥塞了？

拥塞量度

- 因为缺乏缓存空间而丢弃的分组百分比
- 平均队列长度
- 超时和重传的分组数
- 平均分组延迟
- 分组延迟的标准方差 (standard deviation)



数值越大表示拥塞的程度越重

检测到拥塞后怎么办？

拥塞信息传播

- 检测到拥塞的路由器发送一个警告分组给流量源，但是……
- 其他方法：每个分组可以保留一位或一个域，当拥塞度量超过阈值的时候，路由器填充该位或域，以此警告它的邻居。
- 其他方法：主机或路由器周期性地向外发送探测分组，显式地询问有关拥塞的情况，然后，在有问题的区域利用回收的信息来路由流量（类比：交通电台）

怎样解决拥塞问题？

拥塞根源：负载 > 资源

增加资源

- 在某些点之间使用更多的通道增加带宽（广深）
- 把流量分散到多条路径
- 启用空闲或备份的路由器

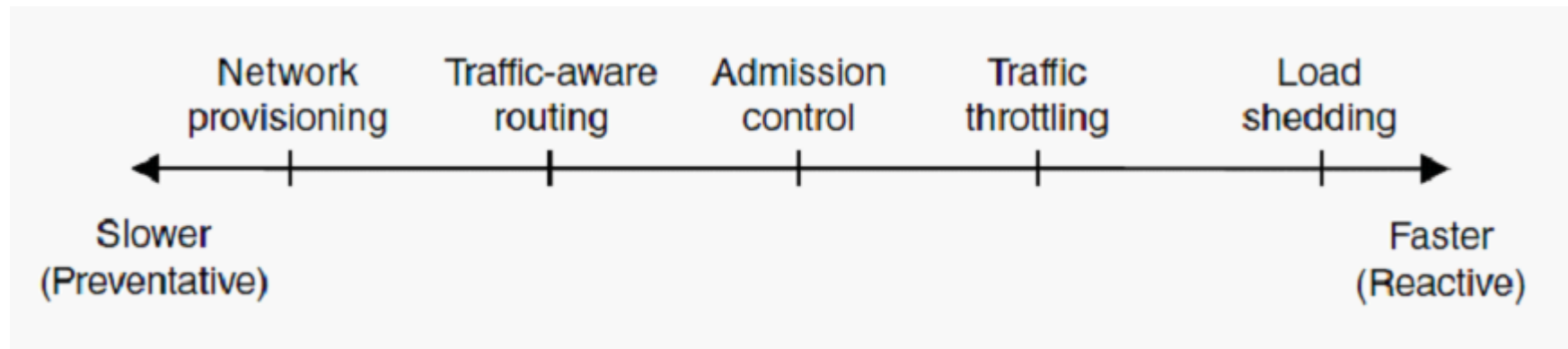
降低负载

- 拒绝为某些用户提供服务（春节车票提价，限行）
- 给某些用户的服务降低等级（黄金周旅游）
- 让用户更有预见性地安排他们的需求（年假制）



开环系统中的拥塞控制方法

- 一开始就将拥塞发生的可能性降到最低，而不是随它发生了之后再去反应
- 在各个层面上采用适当的策略达成这个目标





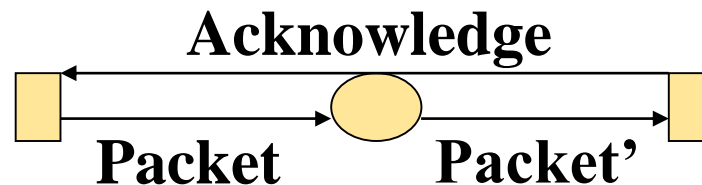
数据报子网中的拥塞控制

- 每台路由器可以监视它的输出线路和其它资源的使用情况
- 每条线路和一个实变量 u 关联在一起，其值位于(0.0 -1.0)之间
- 无论何时，只要 u 超出了阈值，对应的线路就进入到警告
“warning” 状态
- 每个新到达的分组都将被检查，看它的输出线路是否处于
“警告状态”

处于警告状态后可以采取的措施

□ The Warning Bit (警告位)

- 分组头部中的一个特殊的位
- 该位被复制到下一个确认分组中，被传回源
- 源机会监视设置了警告位的分组的比例，相应地调整它的发送速度



处于警告状态后可以采取的措施

□ Choke Packets（抑制分组）

- 路由器给源机发回一个抑制分组，并在抑制分组中指明原分组的目标地址
- 当源机收到抑制分组后，它会以某种百分比减少它发向该目标的流量



处于警告状态后可以采取的措施

□ Choke Packets（抑制分组）

- 在一段固定的时间内，源机会忽略同样一个目标的抑制分组
- 过了这段时间，会继续侦听有无新的抑制分组，如果没有，源机将增加流量



处于警告状态后可以采取的措施

▣ 逐跳抑制分组（Hop-by-Hop Choke Packets）

当网络速度很高或者路由器离源主机的距离很远的时候，给源主机发送抑制分组并不能很好地起作用，因为反应太慢

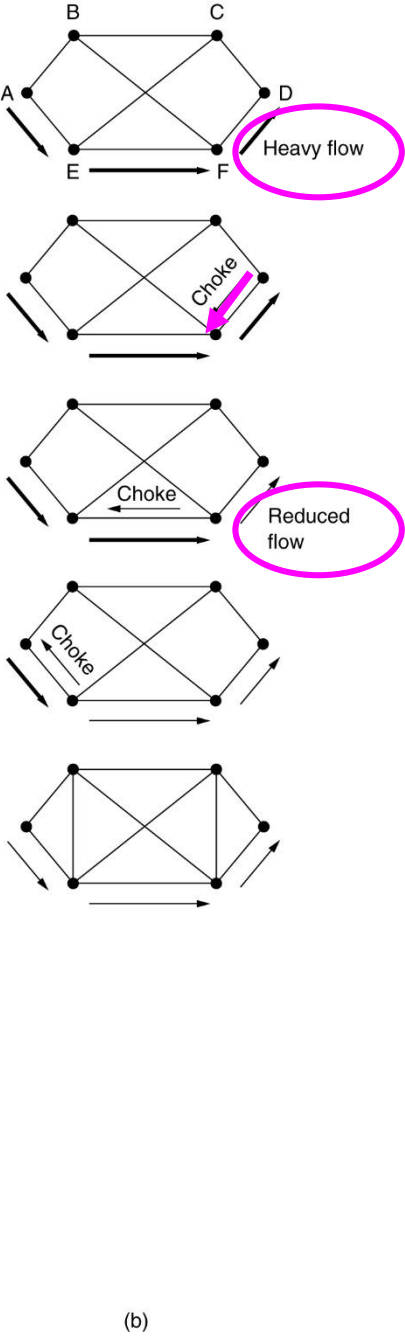
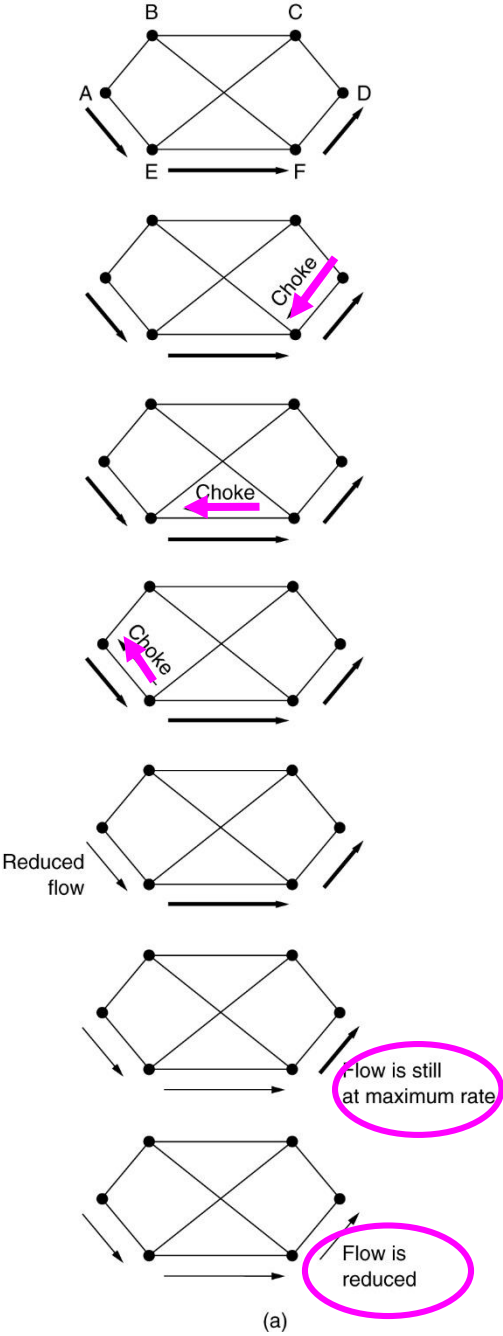
改进方法

让抑制分组路径的每个路由器都采取相应的措施

逐跳抑制的效果

拥塞点上的拥塞很快得到了缓解，但是付出的代价就是，上游路由器需要更多的缓存空间

抑制分组机制





负载丢弃/载荷脱落

- 处理拥塞的最极端的方法
- 当路由器收到的分组超载了，一些分组会被丢掉



丢掉哪些分组呢



负载丢弃/载荷脱落

丢掉分组：

- 随机丢弃 (random)
- 丢弃新到达的 (葡萄酒策略, 适合文件传输类)
- 丢弃早到达的分组 (牛奶策略, 适合多媒体类)
- 丢弃不太重要的 (less important) 分组
 - 需要发送方在它们的分组中标明优先级



随机早期检测RED（防患于未然）

- 当情况变得恶化无可救药之前就开始丢弃分组
- 为了确定什么时候开始丢弃分组，路由器维护着最近的队列平均长度
- 当某条线上的队列平均长度超过了某阈值时，该线路被认定是拥塞的，可以采取相应的措施



小结

- ❑ 拥塞的根源是负载大过资源。
- ❑ 抑制分组生效时间慢，逐跳抑制可以快速缓解拥塞点压力。
- ❑ 载荷脱落是见效快的粗暴方法，可根据应用选择不同的脱落策略。
- ❑ 随机早起检测可以防患于未然。

思考题

- 拥塞的量度有哪些？
- 怎么控制拥塞？
- 抑制分组和逐条抑制分组是怎么控制拥塞的？
- 载荷脱落有哪些策略？ 分别适合什么样的应用？

谢谢观看

致谢

本课程课件中的部分素材来自于：（1）清华大学出版社出版的翻译教材《计算机网络》（原著作者：Andrew S. Tanenbaum, David J. Wetherall）；（2）思科网络技术学院教程；（3）网络上搜到的其他资料。在此，对清华大学出版社、思科网络技术学院、人民邮电出版社、以及其它提供本课程引用资料的个人表示衷心的感谢！

对于本课程引用的素材，仅用于课程学习，如有任何问题，请与我们联系！