<!--

- @Title:
- @Description:
- @Author: SoulCompiler
- @Email: yangtw7@gmail.com
- @Blog: SoulCompiler.github.io
- @Creation Date: 2021-06-15 18:17:23
- @LastEditors: SoulCompiler
- @LastEditTime: 2021-06-15 18:17:23

-->

第 24 章 拥塞控制和服务质量

1. 网络性能通常由两个网络因素来衡量：吞吐量和延迟

吞吐量和延迟 [IMP]

- 通常是衡量网络性能的因素
- 吞吐量:
    - 单位时间内通过网络的分组数量
    - 载荷小于网络容量时, 吞吐量随载荷增加成比例增长
    - 载荷达到网络容量时, 吞吐量会急剧下降 (一些分组被丢弃导致重传)
- 延迟: 当载荷远小于网络容量时，延迟最小
    - 最小延迟是由传播延迟和处理延迟所组成的，且都可以忽略不计
    - 载荷达到网络容量时, 延迟急剧增加 (分组需要排队)

吞吐量和延迟 [IMP]

- 通常是衡量网络性能的因素
- 吞吐量:
    - 单位时间内通过网络的分组数量

        - 载荷小于网络容量时, 吞吐量随载荷增加成比例增长
        - 载荷达到网络容量时, 吞吐量会急剧下降 (一些分组被丢弃导致重传)
    - 延迟: 当载荷远小于网络容量时，延迟最小
        - 最小延迟是由传播延迟和处理延迟所组成的，且都可以忽略不计
        - 载荷达到网络容量时, 延迟急剧增加 (分组需要排队)

延迟和载荷

- 当载荷比网络容量小得多时，延迟最小。最小延迟是由传播延迟和处理延迟所组成的，并且它们都可以忽略不计。
- 当载荷达到网络容量时，延迟就会急剧增加，因为现在需要将在队列（对路径中所有的路由器）中等待的时间加到总的延迟中。
- 当载荷大于网络容量时，延迟会变为无穷大。

二. 典型的拥塞控制方法

开环拥塞控制

- 重传 (例如 TCP)
- 窗口 (比如选择性重复 ARQ)
- 确认 (ACK, 使用一个 ACK 确认多个分组与停止-等待 ARQ)
- 丢弃 (按照优先级)
- 许可 (虚电路建立连接时检查是否有拥塞, 有则拒绝建立连接)

闭环拥塞控制

- 背压: 从目的端方向的节点逐个拒绝接受上行节点的数据, 直到源端
- 抑制分组: 从发生拥塞的节点发送警告信息到源端
- 隐含信令: 进行推测, 例如过长时间没有收到确认
- 显示信令: 把信号包含在携带数据的分组中, 分前向信令 (向发生拥塞的方向), 后向信令 (原理发生拥塞的方向)

开环拥塞控制

- 在拥塞发生之前，应用某种策略来预防拥塞现象的发生。
- 重传策略
- 窗口策略
- 确认策略
- 丢弃策略
- 许可策略

闭环拥塞控制

- 在拥塞发生之后，采用闭环拥塞控制可以缓解拥塞状况。
- 背压：
    - 一个拥塞点停止接收来自直接上行节点或一些近邻节点的数据。这会引起上行节点或一些近邻节点发生拥塞，它们依此拒绝它们的上行节点或一些近邻节点的数据，依此类推。
    - 背压是点到点拥塞控制。
    - 背压技术仅用于虚电路网络。
- 抑制分组：
    - 抑制分组是一个分组，该分组由节点发送给源端，通知它发生拥塞的情况。
    - 与背压的区别：
    - 警告从已经发生拥塞的路由器直接传到源端，该分组经过的那些中间的节点没被警告。但在背压方法中，警告从一个节点到它的上行节点，虽然警告可能最后到达源端。
- 隐含信令：
    - 在隐含信令中，拥塞节点或节点与源端之间没有通信。
    - 源端能从其他有关征兆中察觉出在网络某处有拥塞。
- 显式信令：
    - 发生拥塞的节点能发送一种显式信令通知源端或目的端发生了拥塞。
    - 与抑制分组的不同：
        - 在抑制分组方法中，有一个单独的分组用于此目的。而在显式信令方法中，信号包含在携带数据的分组中。

    - 后向信令将一个位设置在分组中，并使之向与拥塞发生方向相反的方向移动。该位提示源端网络发生了拥塞，并提示它需要放慢发送速度，以避免分组的丢失。
    - 前向信令反之。

TCP中的拥塞控制（接上一章）

- 拥塞窗口：
    - 发送方窗口大小不仅取决于接收方，而且还取决于网络拥塞的情况。
    - 发送方有两种信息：接收方通告的窗口大小和拥塞窗口大小。实际的窗口大小是这两者中的最小者。
- 如何确定拥塞窗口大小（拥塞策略）：
    - 基于三个阶段：慢速启动、拥塞避免和拥塞检测。
    - 在慢速启动阶段，发送方用很慢的传输速率启动，但迅速地增加到阈值。
    - 在达到阈值时，为了避免拥塞而降低数据速率。
    - 如果检测到拥塞，则发送方基于如何检测拥塞而返回到慢速启动或拥塞避免阶段。