

浅谈拥塞控制与 BBR 算法

摘要

先帝创业未半而中道崩殒

关键词： 关键词 1； 关键词 2； 关键词 3

Abstract

The founding emperor passed away before his endeavor was half completed, and now the empire is divided into three parts. Yizhou is exhausted and in decline, and this is truly a critical moment of survival or destruction. Ho

Key Words: Keyword 1; Keyword 2; Keyword 3

1 绪论

1.1 研究背景

随着互联网的普及和应用范围的不断扩大，网络中的信息呈现爆炸式增长，对网络及其基础设施提出了更高的要求。网络数据传输的条件也在不断变化：在地面互联网中，传输带宽越来越高，链路构成越来越复杂；而在一些特殊网络中，如卫星网络中，通信距离越来越远，传播延时越来越大。这都为当前的传输协议带来了前所未有的挑战^[1]。

TCP 作为一种可靠传输协议，通过在数据传输中引入确认和重传机制保证可靠性。当链路不稳定时，丢包率和错误率都会上升；当网络拥挤时，排队的分组总量超出了路由器缓存的大小，导致后续到达的分组被丢弃。由此，大量数据需要重传，从而加剧了网络拥塞。

1983 年，TCP 被默认部署在当时的互联网——ARPANet 上。1986 年 10 月，由于 ARPANET 网络中的一个路由器出现了故障，大量数据包被重复发送。在加州大学伯克利分校和 400 码外的劳伦斯伯克利国家实验室之间的 32 kbps 链路上检测到互联网拥塞崩溃，在此期间吞吐量下降了近 1000 倍，降至 40 bps^[2]。这一现象引起了网络研究者的注意，因此提出了拥塞控制，使其成为一门新的研究领域。

拥塞控制算法是 TCP 协议中非常重要的一部分。在网络拥塞时，主机不再贪婪地发出数据，而是采取更加保守的策略，降低自身发送数据包的速率，从而保证互联网的可用性，尽可能提高带宽利用率。

1.2 主机通信模型

本文不考虑路由器的转发寻址功能，因此将复杂的计算机网络抽象为点对点的通信链路，如图 1 所示。

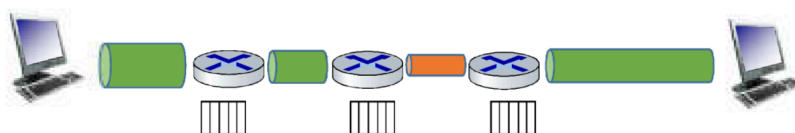


图 1：网络抽象模型

每个路由器具有一个缓存池，用于缓存路由器接收到的分组。若将链路当成水管，每个数据包分组作为水流，则缓存相当于水池，具有积蓄作用。当某个路由器的输入速率大于输出速率时，多余的分组将在缓存中排队。若分组到达路由器，而缓存已满，该分组将会被丢弃。

在点对点数据传输中所能达到的最高速率取决于瓶颈链路的带宽，也就是传输能力最弱的链路的带宽。

空闲网络中的分组增加，可以提高网络吞吐量。但是当拥塞发生时，网络中的每台主机都需要降低自身的数据发送速率。这必然会导致网络的总体利用率变低，就像在堵车时，道路的车流量实际是减少的。

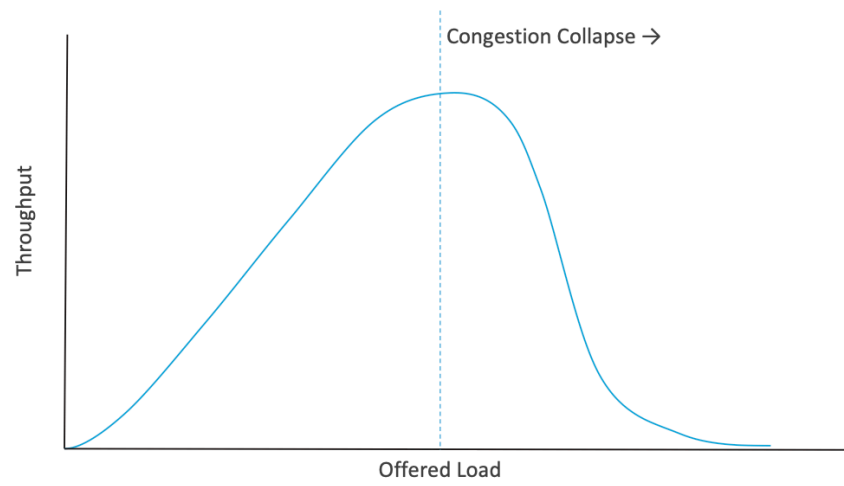


图 2: 随着负载的增加, 吞吐量会上升, 然后在拥塞崩溃点下降。^[3]

2 拥塞控制

2.1 拥塞控制概述

计算机与路由器构成了一个庞大的网络，任一时间任一节点的数据传输量都在变化。拥塞控制算法运行于主机之上，（在不改变现有网络协议与结构条件下）不能直接访问路由的缓存状态。因此拥塞控制算法需要通过其他办法探知网络的拥挤程度，并控制主机行为缓解拥塞。

2.2 拥塞模型

从 TCP 的角度来看，任意复杂的路径都表现为具有相同 RTT（Round-trip time，往返时间）和瓶颈速率的单个链路。RTprop（往返传播时间）和 BtlBw（瓶颈带宽）这两个物理约束限制了传输性能。如果网络路径是物理管道，则 RTprop 将是其长度，BtlBw 将是其最小直径。^[4]

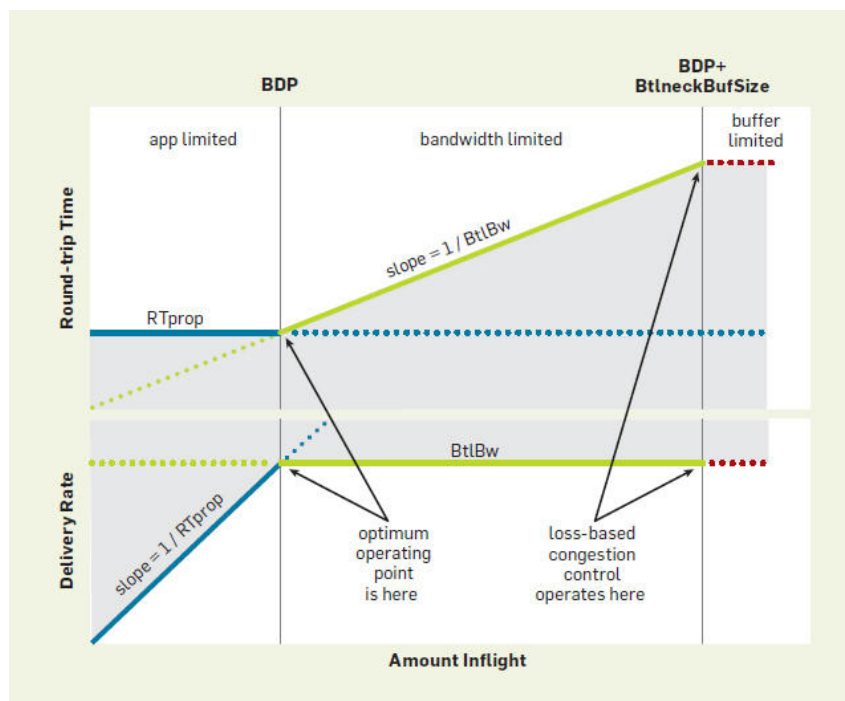


图 3：传输速率与往返时间的关系^[4]

图 3 展示了网络运行过程中的三个阶段。第一个阶段，网络处于轻载状态，由于分组无需排队，往返延迟不变，而吞吐量完全取决于主机的发送速率，发送方发送得多快，接收方就能收得多快。第二个阶段，应用进程向网络中的注入速率超过了瓶颈链路的带宽，这个带宽本质上是吞吐量的上限。分组在缓存中排队，而往返延迟随排队的长度增加而增加。此时网络已经出现了拥塞。第三个阶段，网络负载继续增加，排队分组数量超过了缓存容量，多余分组被丢弃，数据需要重传，因此往返延迟与吞吐量是未知的。

2.3 拥塞控制算法

3 BBR 算法

3.1 算法概述

BBR 拥塞控制算法实时测量网络瓶颈带宽和最小延迟，通过计算带宽延时积（Bandwidth-Delay Product, BDP）来调整发送速率，实现了高吞吐量和低延时。因此，BBR 算法被认为开创了拥塞控制的新纪元。

BBR 算法由 Google 在 2016 年提出。

3.2 算法性能

参考文献

- [1] 曹涛涛. 拥塞控制算法的性能评估及公平性分析. 2017.
- [2] NETLAB C. Communication Networks[EB/OL]. (2023). <http://netlab.caltech.edu/research/communication-networks/>.
- [3] FISHER B, MICHELONI G, BEMMEL J van, 等. Introduction[EB/OL]. (2023). <https://tcpcc.systemsapproach.org/intro.html>.
- [4] CARDWELL N, CHENG Y, GUNN C S, 等. BBR: Congestion-Based Congestion Control[J/OL]. ACM Queue, 2016: 20–53. <http://queue.acm.org/detail.cfm?id=3022184>.