

DOI:10.11918/202109034

BBR 拥塞控制算法的 RTT 公平性优化

潘婉苏^{1,2}, 李晓风^{1,2}, 谭海波¹, 许金林¹, 李哲茹¹

(1. 中国科学院 合肥物质科学研究院, 合肥 230031; 2. 中国科学技术大学, 合肥 230026)

摘要: Google 提出了一种基于瓶颈带宽和往返传播时间的拥塞控制算法 (bottleneck bandwidth and round-trip propagation time, BBR), 可以在网络链路中保持最大传输速率和最小延时。然而一些评估实验表明, BBR 算法会导致不同往返时间 (round trip time, RTT) 的数据流之间存在严重的公平性问题。为了优化这一问题, 研究分析了 BBR 算法探测机制所导致的发送速率与瓶颈带宽不匹配对 RTT 公平性的影响, 提出了一种基于起搏增益模型的优化算法 BBR-adaptive (BBR-A)。BBR-A 算法不再采用原 BBR 算法中固定的起搏增益, 而是利用 RTT 与起搏增益的关系, 构造一个基于反比例函数的起搏增益调节模型, 通过让向上和向下的起搏增益系数相互交错来平衡发送速率, 使每个 BBR 流可以公平地竞争带宽资源。网络模拟器 3 (network simulator 3, NS3) 仿真实验结果表明: BBR-A 算法的信道利用率比 BBR 算法有了小幅提升; 在 RTT 公平性的方面, BBR-A 缩小了不同 RTT 流之间的吞吐量差异, 在不同缓冲区和 RTT 差异下, Jain 公平指数至少提高了 1.5 倍; BBR-A 算法明显降低了重传率。因此通过自适应调整起搏增益系数, 可以平衡不同数据流之间的发送速率, 有效提升 BBR 算法的 RTT 公平性。

关键词: 拥塞控制; BBR; RTT 公平性; 起搏增益; 自适应算法

中图分类号: TP 393 **文献标志码:** A **文章编号:** 0367-6234(2022)11-0038-09

RTT fairness optimization of BBR congestion control algorithm

PAN Wansu^{1,2}, LI Xiaofeng^{1,2}, TAN Haibo¹, XU Jinlin¹, LI Xiru¹

(1. Hefei Institute of Physical Science, Chinese Academy of Sciences, Hefei 230031, China;

2. University of Science and Technology of China, Hefei 230026, China)

Abstract: Google proposed a congestion control algorithm based on bottleneck bandwidth and round-trip propagation time (BBR), which can maintain maximum transmission rate and minimum latency in a network link. However, the BBR algorithm was reported to cause serious round trip time (RTT) fairness problems by some evaluation experiments. The impact of the mismatch between pacing rate and bottleneck bandwidth caused by the asynchronous detection mechanism of BBR algorithm was analyzed to optimize the RTT fairness, and an optimized algorithm BBR-adaptive (BBR-A) was proposed based on pacing gain model. According to the relationship between RTT and pacing gain, a pacing gain adjustment model based on inverse proportional function was established, which replaces the fixed pacing gain coefficient in the original BBR algorithm. By interleaving the up and down pacing gain coefficients to adjust the pacing rate, each BBR flow could compete for bandwidth resources fairly. Experimental results of network simulator 3 (NS3) show that the channel utilization of BBR-A algorithm was slightly improved compared with BBR algorithm. In the experiment of RTT fairness, BBR-A reduced the throughput difference between different RTT flows, and Jain fairness index was at least 1.5 times higher than BBR algorithm with different buffer sizes and RTT differences. The retransmission rate of BBR-A algorithm was significantly reduced. By adaptively adjusting the pacing gain coefficient, the pacing rate between different flows was balanced, and the RTT fairness of BBR algorithm was improved.

Keywords: congestion control; BBR; RTT fairness; pacing gain; adaptive algorithm

TCP 拥塞控制对于互联网及其应用的发展和成功起到了重要作用。常用的拥塞控制算法是基于丢包反馈来调节拥塞窗口 (congestion window, CWND) 的大小, 如 Reno、BIC、CUBIC 等算法^[1-3]。但在高速、长距离的现代网络中, 基于丢包的算法会造成吞

吐量严重下降, 往往会浪费很多可用带宽^[4]。2016 年 Google 提出了 BBR (bottleneck bandwidth and round-trip propagation time) 算法^[5], 通过对瓶颈带宽和往返传播时间的估测来调整其发送行为, 实现了高吞吐量的同时减少了传输延迟。Google 公布的实施情况和一些公开发表的评测结果^[6-11]表明, BBR 可以显著提高 TCP 连接的吞吐量, 与传统 CUBIC 相比在吞吐量上具有显著优势, 但 BBR 仍存在着一些缺陷, 其中 RTT (round trip time) 公平性问题是比较

收稿日期: 2021-09-08

基金项目: 国家重点研发计划“区块链”重点专项 (2021YFB2700700)

作者简介: 潘婉苏 (1992—), 女, 博士研究生;

李晓风 (1966—), 男, 教授, 博士生导师

通信作者: 李晓风, xfli@hfcas.ac.cn

严重的问题之一。即当多个数据流共享同一瓶颈链路时, RTT 较长的 BBR 数据流具有带宽占用优势^[7,12-15]。

针对 RTT 公平性问题, 一些学者进行了改进。文献[7]提出了 BBQ 算法, 通过限制 RTT 较长的数据流的检测周期来降低长 RTT 流的带宽占用。当 $T > (1 + \beta) T_{\text{prop}}$ 时 (T 为 RTT, T_{prop} 为 RTprop, 即 round propagation time), 使用 $\min(\alpha; T_{\text{prop}})$ 作为探测周期的持续时间, 从而减小长 RTT 流发送的数据包数量, 其中 $\beta \in [0.5\%, 1\%]$, $\alpha = 3 \text{ ms}$ 。文献[12]通过建立理论模型证明了不同 RTT 流带宽占用受 RTT 比率的影响, 长 RTT 流的带宽占比具有优势。文献[13]提出了 RFBBR 算法, 在 PROBE_BW 阶段添加公平性因子 $\gamma = (T + T_{\text{prop}}) / (2 \times T)$ 调节拥塞窗口的大小, 进而改进 RTT 公平性。文献[14]提出了延迟感知 BBR (delay-aware BBR, DA-BBR) 算法, 根据 RTT 定义一个小于 1 的调节因子去限制链路中的带宽时延积 (bandwidth delay product, BDP), 缓解了 RTT 不公平性。文献[15]提出通过调整拥塞窗口增益缩小不同 RTT 流之间的窗口差距, 从而提升 RTT 公平性。以上这些算法在提高 RTT 的公平性方面取得了一定积极的效果, 但分析发现, 现有的这些改进方案都限制了 BBR 流的探测, 并且仍采用固定的起搏增益, 而未对起搏增益这一关键参数进行优化。

本文对 BBR 算法的探测过程进行建模分析, 利用 RTT 反馈调节 BBR 在探测带宽状态中的起搏增益, 建立起搏增益模型去平衡不同流之间的发送速率, 从而提升不同 RTT 流之间的公平性。网络模拟器 3 (network simulator 3, NS3) 的仿真实验表明, 改进算法 BBR-adaptive (BBR-A) 有效提升了算法的 RTT 公平性以及其它方面的性能。

1 BBR 算法概述

1.1 基本原理

不同于基于丢包的拥塞控制算法, BBR 算法通过交替测量链路中的最大带宽和最小延来解决寻找 Kleinrock 最佳操作点^[16]的问题。BBR 将最近 10 次往返中测得的最大带宽视为 B_{max} , 将过去 10 s 中测得的最小延时视为 T_{prop} , 然后根据 B_{max} 和 T_{prop} 估算 BDP^[16], 即

$$B_{\text{BDP}} = B_{\text{max}} \times T_{\text{prop}} \quad (1)$$

BBR 分别通过窗口增益和起搏增益来调整 CWND 和发送速率, 进而控制其发送行为。CWND 和发送速率的计算公式为

$$W_{\text{cwnd}} = \text{cwnd_gain} \times B_{\text{BDP}} \quad (2)$$

$$S = \text{pacing_gain} \times B_{\text{max}} \quad (3)$$

式中: W_{cwnd} 为 CWND, cwnd_gain 为窗口增益, S 为发送速率, pacing_gain 为起搏增益。

BBR 主要由 4 个状态构成, 见图 1。

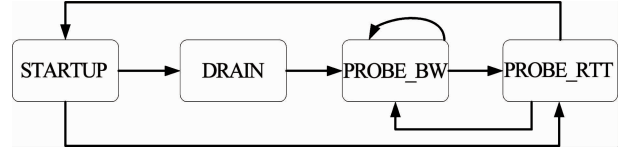


图 1 BBR 的状态模型

Fig. 1 State model of BBR

在 STARTUP 状态, 将 cwnd_gain 和 pacing_gain 设置为 $2/\ln 2$, 使 CWND 和发送速率指数增加, 积极的探测链路中的可用带宽。如果连续 3 个 RTT 内新估算的带宽增加没有超过 25%, BBR 则进入 DRAIN 状态。在 DRAIN 状态, BBR 通过将 pacing_gain 降低到 $\ln 2/2$ 来清除前一状态的剩余队列, cwnd_gain 保持不变 ($2/\ln 2$)。在 PROBE_BW 状态, BBR 循环 8 个周期 ($\text{pacing_gain}[] = [1.25; 0.75; 1; 1; 1; 1; 1; 1]$) 进行探测带宽, 每个周期持续时间为 T_{prop} 。在这个状态中 CWND 固定为 $2B_{\text{BDP}}$ 。BBR 每 10 s 进入一次 PROBE_RTT 状态, 在这个状态 CWND 被设置为 4MSS (Maximum Segment Size), 以确保对 T_{prop} 新值进行采样, 该状态持续 200 ms。在 PROBE_RTT 状态结束后, 通过瓶颈链路状态判断转换到 PROBE_BW 或 STARTUP 状态继续进行循环。

1.2 RTT 公平性原因分析

在 PROBE_BW 状态, BBR 周期性地以 $\text{pacing_gain} = 1.25$ 增加发送速率, 向瓶颈链路发送更多的数据包, 导致总发送速率大于瓶颈带宽, 在瓶颈上形成了持久队列。随后 $\text{pacing_gain} = 0.75$ 降低发送速率, 试图排空先前探测生成的多余的队列。但这种循环实际无法排空多余的数据, 仍然会形成队列积压。根据排队论, 一旦在瓶颈处形成持久队列, 不同数据流的吞吐量就取决于它的队列份额。

当不同 RTT 流通过瓶颈链路时, 长 RTT 流的 B_{BDP} 估算值大于短 RTT 流的 B_{BDP} 估算值, 其可以发送更多数据包, 因此在瓶颈队列中占主导地位。长 RTT 流的队列份额优势可以让其获得比短 RTT 流更高的发送速率, 使短 RTT 流的发送速率不断降低。持续的低传输速率将导致短 RTT 流在下一次带宽测量中获得更小的 B_{BDP} 估算值, 然后再次降低发送速率, 又将进一步减少其队列, 循环往复, 最终导致短 RTT 流的带宽占用严重下降。即使在探测过程中的一些数据流提前终止, 上面的结论仍然成立, 因为其他数据流将很快抢占空闲带宽。随着数

据流 RTT 差异的增加,公平性会进一步恶化。一些用户可能利用这个漏洞故意增加延迟去获得高带宽。因此,解决 BBR 算法中 RTT 公平性问题是必要的。

2 改进算法:BBR-A

2.1 模型分析

通过本文 RTT 公平性分析可以知道,队列积压的产生导致长 RTT 流的发送速率大于短 RTT 流。对 BBR 流建模,分析 RTT 与发送速率之间的关系。假设有 n 个不同数据流通过带宽为 C 的瓶颈链路, $f_i \in [1, n]$ 为第 i 个数据流, $d_i(t)$ 为 f_i 在 t 时刻的传递速率,在 t 时刻 f_i 的估算最大带宽为

$$B_{\max i}(t) = \max(d_i(t)) (T \in [t - 10R_{\text{RTT}}; t]) \quad (4)$$

在实际传输过程中,RTT 由传播时延和排队时延组成, f_i 在时间 t 时刻的 RTT 为

$$T_i(t) = q_i(t) + T_{\text{prop}i} \quad (5)$$

式中 $q_i(t)$ 为 f_i 在 t 时刻的排队时延。

设 $I_i(t)$ 为 f_i 的飞行中数据量,则 $I_i(t)$ 与 $d_i(t)$ 关系为

$$d_i(t) = \frac{I_i(t)}{T_i(t)} \quad (6)$$

进一步分析 RTT 与起搏增益之间的关系,根据 BBR 算法的带宽探测机制,在向上探测周期 $\text{pacing_gain} = 1.25$,因此在 t 时刻最大传递速率为

$$\max(d_i(t)) = \frac{\max(I_i(t))}{T_i(t)} = \frac{1.25 \times T_{\text{prop}} \times B_{\max i}(t - \Delta t)}{T_i(t)} \quad (7)$$

BBR 流的探测周期为 8 个 T_{prop} ,结合式(7)可以知道 f_i 在新一轮最大带宽估算时更新为

$$B_{\max i}(t) = \frac{1.25 \times T_{\text{prop}i} \times B_{\max i}(t - 8T_{\text{prop}i})}{T_i(t)} \quad (8)$$

对于共享同一瓶颈队列的数据流,其排队时延是相同的。一旦有队列的生成,实际带宽无法按照预期的 1.25 倍进行增加,实际的起搏增益为 $\frac{1.25 \times T_{\text{prop}}}{T} (< 1.25)$ 。通过实际增益系数可以看出传播时延和排队时延对不同 RTT 流带宽探测的影响,RTT 越大,BBR 流的有效起搏增益越大,实际带宽抢占能力越强。因此,可以通过平衡不同 RTT 流之间的起搏增益来消除 RTT 对发送速率的影响,从而提高 BBR 算法的 RTT 公平性。

2.2 算法设计

为了平衡不同 RTT 流的带宽占用,可以在现有增益系数基础上乘上一个 RTT 的减函数来消除 RTT 对发送速率的影响。如果一个长 RTT 数据流,

那么其发送速率缓慢地向上增加并且快速地向下减少。如果一个短 RTT 数据流,那么它的发送速率应该快速地向向上增加并且缓慢地向向下减少,从而实现起搏增益系数对发送速率的约束。根据 RTT 的测量机制,将 φ 定义为 f_i 当前延时与最大延时比值,计算公式为

$$\varphi_i = \frac{T_i}{T_{\max}}, \varphi_i \in (0, 1] \quad (9)$$

式中: T_i 为 f_i 从最后一个 ACK 中获得的当前延时, $T_{\max}$ 为 ACK 更新的最大延时。

通常 φ 可以量化瓶颈的链路利用率^[17-18], T_i 越大, φ 越大。仅在瓶颈链路容量和缓冲区即将满载时, T_i 才接近最大的 $T_{\max}$ 。通过 φ 构建关于 RTT 的减函数作为新的向上和向下起搏增益系数,进而约束不同 RTT 流的发送速率。

基于上述分析,分别定义了向上起搏增益系数 P_{up} 和向下起搏增益系数 P_{down} 来替代固定的起搏增益系数 1.25 和 0.75。对于 P_{up} ,相关函数 $P_{\text{up}}(\varphi)$ 应该为一个下凹的曲线,下渐近线为 $P_{\text{up}} = 1$,并且随着 φ 变大, $P_{\text{up}}(\varphi)$ 下降趋势变缓,以减少优势数据流的发送速率,给弱势数据流提供更多的可用带宽。对于 P_{down} ,相关函数 $P_{\text{down}}(\varphi)$ 应该为上凸的曲线,上渐近线为 $P_{\text{down}} = 1$,并且随 φ 越大, $P_{\text{down}}(\varphi)$ 下降趋势变快。此外, $P_{\text{up}}(\varphi)$ 和 $P_{\text{down}}(\varphi)$ 所需函数必须为低复杂度的函数,因为它要在 BBR 算法内核中实现。基于这些约束条件,测试了一些函数,发现反比例函数可以满足需要。基于反比例函数的特性,构造了 $P_{\text{up}}(\varphi)$ 和 $P_{\text{down}}(\varphi)$

$$P_{\text{up}}(\varphi_i) = \frac{3}{\varphi_i + 2}, P_{\text{up}}(\varphi_i) \in [1, 1.5] \quad (10)$$

$$P_{\text{down}}(\varphi_i) = \frac{3}{\varphi_i - 3} + 2, P_{\text{down}}(\varphi_i) \in [0.5, 1] \quad (11)$$

P_{up} 函数和 P_{down} 函数的变化趋势见图 2。随着 φ 的增大, $P_{\text{up}}(\varphi)$ 缓慢下降,且下降趋势变慢,取值在 $[1, 1.5]$ 之间;随着 φ 的增大, $P_{\text{down}}(\varphi)$ 缓慢下降,且下降趋势变快,取值范围在 $[0.5, 1]$ 之间,满足增益模型的设计需求。BBR-A 算法通过每个 ACK 更新 RTT 的信息调整原 BBR 中起搏增益的大小,从而自适应调节发送速率。此外,根据 BBR 的探测机制,将链路状态的分界点设置为 $1.25B_{\text{BDP}}$ ^[19]。BBR-A 具体细节见算法 1,当 $I_i < 1.25B_{\text{BDP}}$ 时,允许不同 RTT 流发送更多的数据包来占用空闲带宽。在这个向上探测周期内,采用 $\text{pacing_gain} = P_{\text{up}}$ 来增加发送速率。当飞行中的数据量大于 $1.25B_{\text{BDP}}$ 时,表明瓶颈链路没有额外的能力来传输更多的数据包,并会在缓冲区形成队列,此时通过 $\text{pacing_gain} = 1$ 维持

稳定的发送速率。此外,为了增加 BBR 算法对丢包的敏感性,增加丢包阈值 has_loss (阈值设定 2%) 表示是否存在严重丢包。如果 $I_i > 1.25B_{\text{BDP}}$ 并且有丢包产生,此时的向下探测周期要限制不同 RTT 流的发送速率,采用 $\text{pacing_gain} = P_{\text{down}}$ 降低发送速率,释放缓冲区。剩余周期,BBR-A 依然采用原 BBR 中默认的 pacing_gain 。

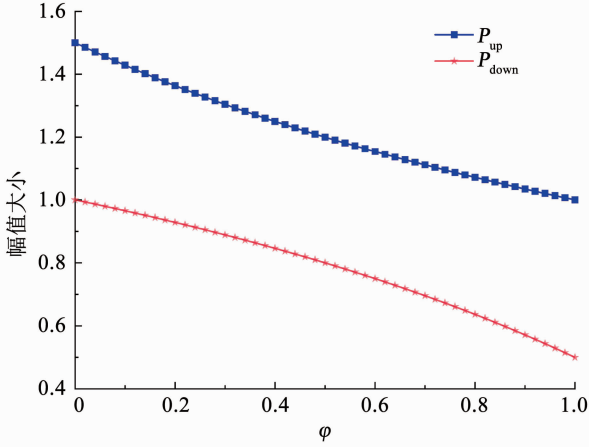


图 2 BBR-A 算法中 P_{up} 和 P_{down} 函数随着 φ 的变化趋势

Fig. 2 Variation of P_{up} and P_{down} functions with φ in BBR-A algorithm

算法 1: BBR-A 算法的 ProbeBW 状态

Input: I_i , has_loss , $T_{\text{max}} \leftarrow 0$, $T_{\text{min}} \leftarrow 0X7FFFFFFF$, $\text{rtt_us} // \text{BBR}$ 中每个 ACK 更新

Output: pacing_gain

```

1: For every ACK do
2:    $\text{rtt\_us} \leftarrow (\text{now sending time})$ 
3:   if  $\text{rtt\_us} < T_{\text{min}}$  then
4:      $T_{\text{min}} \leftarrow \text{rtt\_us}$ 
5:   end if
6:   if  $\text{rtt\_us} < T_{\text{max}}$  then
7:      $T_{\text{max}} \leftarrow \text{rtt\_us}$ 
8:   end if
9: end for

```

Phase 1: probe up

```

1: if  $I_i < 1.25B_{\text{BDP}}$  then
2:    $\text{pacing\_gain} = P_{\text{up}}$ 
3:   return
4: end if

```

Phase 2: probe down

```

1: if  $I_i > 1.25B_{\text{BDP}}$  or  $\text{has\_loss} > 2\%$  then
2:    $\text{pacing\_gain} = P_{\text{down}}$ 
3: else  $\text{pacing\_gain} = 1$ 
3: return
4: end if

```

Phase 3: next six cycles

```

1: if  $\text{pacing\_gain} = 1$  then
2:   return
3: end if

```

BBR-A 算法通过对起搏增益的调节,进一步增强了不同 RTT 流发送速率的自我调节能力。将带宽探测状态的向上起搏增益系数上限扩大到 1.5,提高 BBR 的探测带宽幅度,同时将向下起搏增益系数下限设置为 0.5,加快多余数据包的排空。通过让不同 RTT 流的探测和排空系数彼此交错,可以平衡不同数据流的发送速率,从而提高 RTT 公平性。

参考文献[13]进行 BBR-A 算法的复杂度分析, BBR-A 采用 if 循环语句判断链路状态,因此其时间复杂度为 $O(1)$ 。BBR-A 每个周期只存储起搏增益的计算结果,没有额外的内存消耗,其空间复杂度也为 $O(1)$ 。另外, BBR-A 算法的实现依旧基于原有的 BBR 框架,便于实施与部署。

3 实验结果分析

基于 BBR 版本^[20-21]在 NS3 上测试了 BBR 及优化算法的性能,网络拓扑见图 3。S0 ~ Sn 为发送端, R0 ~ Rn 为接收端,瓶颈带宽设置为 100 Mbit/s。

通过不同网络条件下测试所获得的仿真结果验证 BBR-A 算法有效性。性能指标包括吞吐量、信道利用率、Jain 公平指数和重传率,变量包括 RTT 和缓冲区大小。此外,将 BBQ^[7]和 DA-BBR^[14]作为基准算法加入到实验对比中。

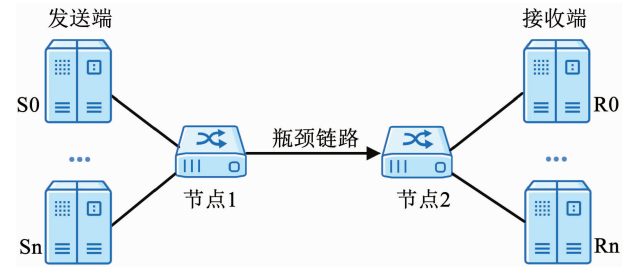


图 3 仿真模拟实验的网络拓扑

Fig. 3 Network topology of simulation experiment

3.1 信道利用率

信道利用率表示信道平均被占用的程度,可以一定程度反映网络利用率。通常通过测量数据流的吞吐量来计算瓶颈链路的信道利用率 η ,即

$$\eta = \frac{\sum_i Q_{\text{bytes}_i}}{C \times T_{\text{last}}} \quad (12)$$

式中: Q_{bytes_i} 为 f_i 在接收端已接收数据包的长度, C 为瓶颈链路的带宽, T_{last} 为持续的模拟运行时间(本文设定为 200 s)。

由于 BBR 算法在高丢包率下容易失速,因此本节测试具有不同随机丢包率下的信道利用率,以评估算法的抗丢包能力。缓冲区大小配置为 $1B_{\text{BDP}}$ 和 $5B_{\text{BDP}}$, 随机分组丢失率分别设置为 0%、1%、3% 和 6%。

BBR、BBQ、DA-BBR 和 BBR-A 算法的信道利用率的仿真结果见图 4。整体来看,4 种算法在 $1B_{BDP}$ 缓冲区的信道利用率要低于 $5B_{BDP}$ 缓冲区。当随机丢包率为 0% 时,4 种算法可以在不同的缓冲区中获得 94% 以上的信道利用率,其中 BBR-A 信道利用率最高, $1B_{BDP}$ 和 $5B_{BDP}$ 缓冲区中分别为 95.6% 和 96.2%。4 种算法的信道利用率均随着丢包率的增

加而降低,但 BBR-A 受丢包的影响相对较小。当随机丢包率为 6% 时,BBR 的信道利用率明显下降,而 BBR-A 的信道利用率降幅最小。BBR-A 算法加入了丢包反馈,并提高了起搏增益上限,可以在不同的缓冲区始终保持 90% 以上的信道利用率,尤其在高丢包的场景中获得更高的吞吐量。

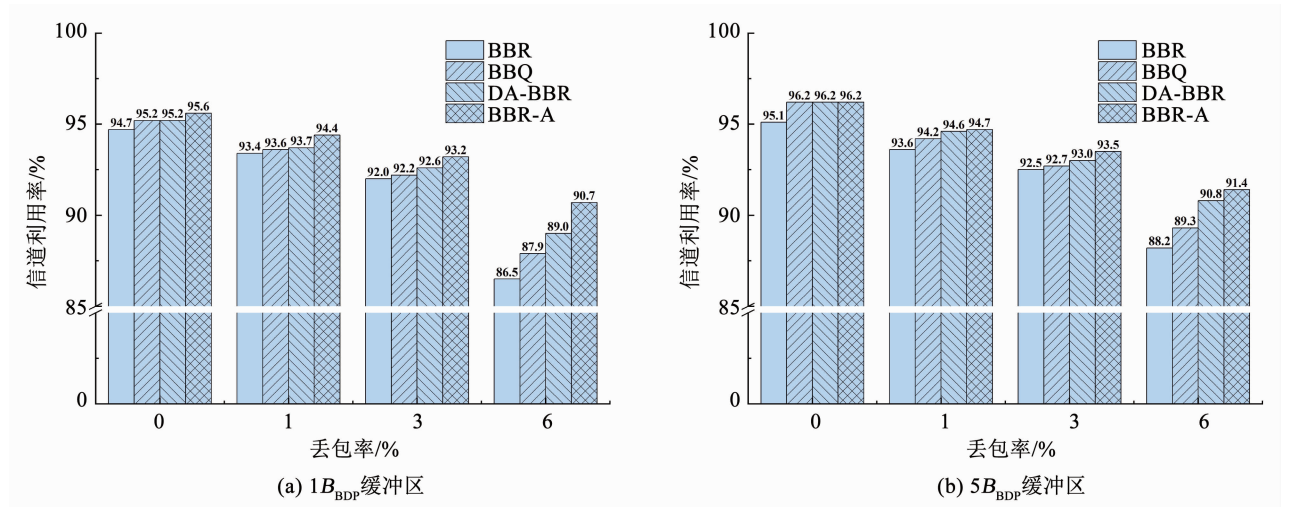


图 4 4 种算法的信道利用率
Fig. 4 Channel utilization of four algorithms

3.2 RTT 公平性

通过比较 BBR、BBQ、DA-BBR 和 BBR-A 算法不同 RTT 流之间的吞吐量,评估 BBR-A 是否缓解了 RTT 不公平性。实验设置 $0.5B_{BDP}$ 和 $5B_{BDP}$ 缓冲区大小,10 ms RTT 流与 50 ms RTT 流竞争瓶颈带宽。图 5 为 10 ms RTT 流和 50 ms RTT 流的吞吐量对比。

对于 BBR 算法,2 种缓冲区中 10 ms RTT 流和 50 ms RTT 流的吞吐量见图 5(a) 和 5(e)。在启动阶段 10 ms RTT 流可以快速获得吞吐量,但与 50 ms RTT 流短暂竞争后,其吞吐量快速下降,最终约为 39.7 Mbit/s,而 50 ms RTT 流吞吐量是其 1.4 倍。在图 5(e) 中,2 种 RTT 流的吞吐量差异明显变大,10 ms RTT 流的吞吐量仅为 13.8 Mbit/s 左右,50 ms RTT 流吞吐量是其 5.9 倍。图 5(b) 和 5(f) 为 BBQ 算法,在 $0.5B_{BDP}$ 缓冲区中 2 种 RTT 流的吞吐量差异相比 BBR 算法变小。10 ms RTT 流的吞吐量提高到 42.3 Mbit/s 左右,50 ms RTT 流的吞吐量约是其 1.3 倍。在 $5B_{BDP}$ 缓冲区(图 5(f)),2 种 RTT 流之间的吞吐量差异比图 5(b) 中的结果有所增加,50 ms RTT 流的吞吐量为 59.3 Mbit/s,是 10 ms RTT 流的 1.6 倍。DA-BBR 算法见图 5(c) 和 5(g),2 种缓冲区中数据流的吞吐量差异变化不大,50 ms RTT 流的吞吐量约是 10 ms RTT 流的 1.3 倍。对于 BBR-A

算法,在 $0.5B_{BDP}$ 缓冲区(图 5(d)),10 ms RTT 流和 50 ms RTT 流的吞吐量差异相比前 3 种算法进一步缩小。10 ms RTT 流和 50 ms RTT 流吞吐量分别为 45.3 Mbit/s 和 50.3 Mbit/s,相差 10%。在 $5B_{BDP}$ 缓冲区(图 5(h)),10 ms RTT 流的吞吐量波动相对较大,说明数据流积极地竞争可用带宽,50 ms RTT 流的吞吐量约是 10 ms RTT 流的 1.3 倍。

通过图 5 的对比结果可以看出,与 BBR 相比,BBQ 提高了 2 种 RTT 流之间的公平性,但在 $5B_{BDP}$ 缓冲区,50 ms RTT 流的吞吐量优势依旧明显。而 DA-BBR 算法也提高了 2 种 RTT 流之间的公平性,尤其是在 $5B_{BDP}$ 缓冲区中相比于 BBQ 算法有了进一步提升。BBR-A 改变了不同 RTT 流的起搏增益,提高了短 RTT 流的带宽竞争能力,缩小 2 种 RTT 流的吞吐量差异,在 4 种算法中实现了最好的 RTT 公平性,尤其在 $0.5B_{BDP}$ 缓冲区中具有明显公平性优势。

另外,通过上述实验结果发现不同的缓冲区大小对 RTT 的公平性有着明显的影响,因此继续开展相关实验比较 10 ms RTT 流和 50 ms RTT 流在不同缓冲区中的性能差异,进一步评估 4 种算法的 RTT 公平性。为了量化算法在不同缓冲区大小下的 RTT 公平性,本文引入了 Jain 公平指数^[22]。Jain 公平指数可以用来衡量带宽竞争中的公平性,计算方法如下:

$$J = \frac{(\sum_{i=1}^n x_i)^2}{n \sum_{i=1}^n x_i^2} \tag{13}$$

式中: n 为链接个数, x_i 为第 i 个链接的吞吐量。

Jain 公平指数(以下简称公平指数)能够很好地反映吞吐量的差异,取值范围在 $[0,1]$ 之间。该公平指数越接近 1,说明带宽分配的公平性就越好。

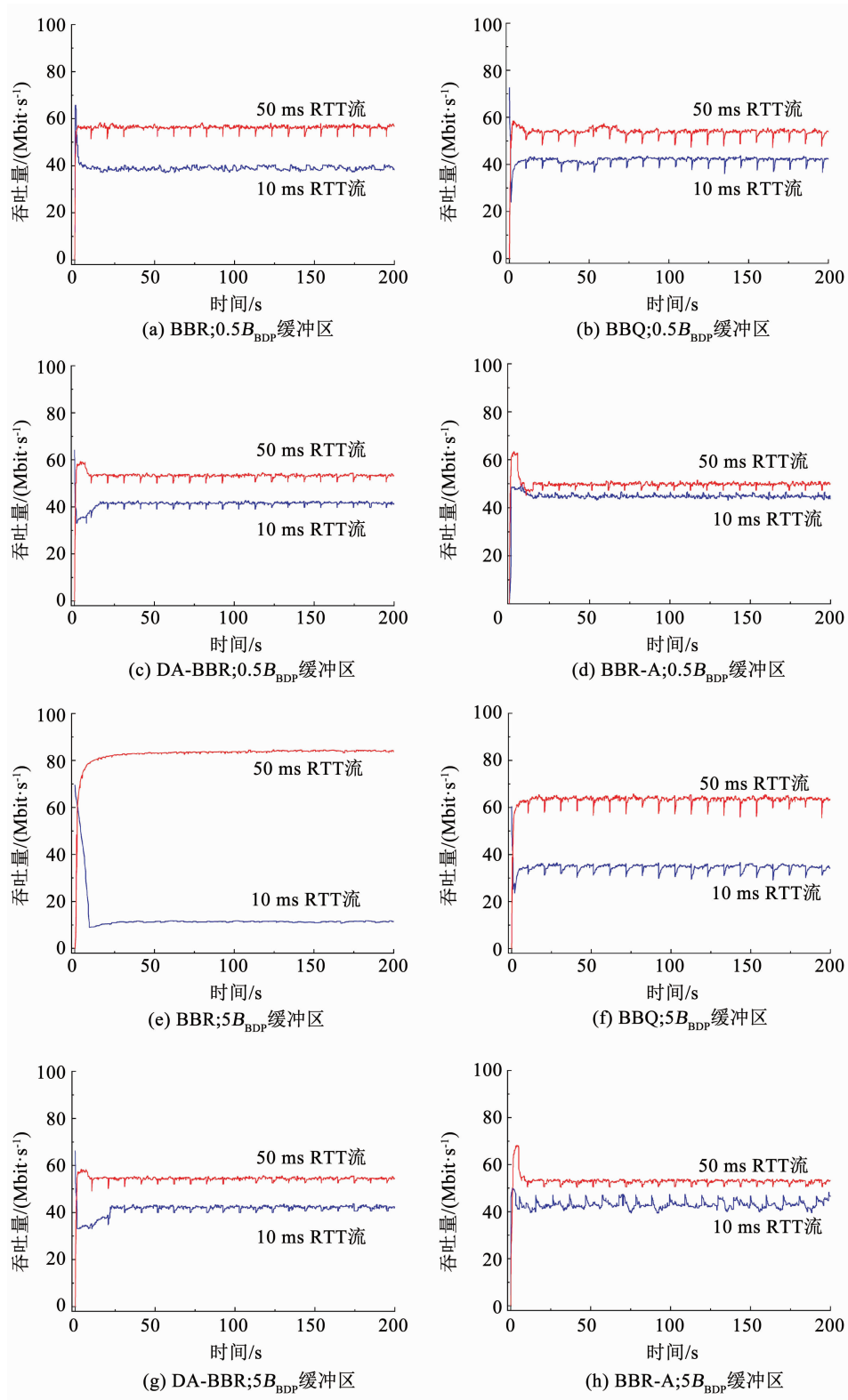


图 5 不同算法在不同缓冲区中 10 ms RTT 和 50 ms RTT 流的吞吐量对比

Fig. 5 Throughput comparison of 10 ms RTT flow and 50 ms RTT flow with different algorithms and buffer sizes

10 ms RTT 流和 50 ms RTT 流在 $0.1B_{BDP} \sim 100B_{BDP}$ 的缓冲区条件下,各自的平均吞吐量变化和公平指数见图 6。图 6(a)为 BBR 算法,随着缓冲区变大,2 种 RTT 流的吞吐量差异变大。当缓冲区大于 $6B_{BDP}$,50 ms RTT 流占用大部分带宽,公平指数下降到 0.63 左右。对于 BBQ 算法,见图 6(b)。相比 BBR,其公平性得到了很大的提升。虽然 2 种数据流竞争时吞吐量差异随着缓冲区变大而加大,公平指数也随着下降,但 BBQ 的公平指数仍能维持在 0.91 以上。图 6(c)中的 DA-BBR 算法,其公平指数最小维持在 0.95 左右。在图 6(d)中,BBR-A 算法明显缩小了 10 ms RTT 流和 50 ms RTT 流的吞吐量差异,其公平指数维持在 0.96 以上。总体而言,BBR-A 比 BBR 和 BBQ 算法 RTT 公平性更好,比 DA-BBR 算法略有优势。尤其在深缓冲区条件下,RTT 公平性相比 BBR 算法有了很大的提升,公平指数提高了约 1.5 倍,实现了更好的性能。

除了缓冲区大小对 RTT 公平性有着明显的影响,RTT 差异也会明显影响 RTT 公平性^[13]。仿真设置 10 ms RTT 流与不同 RTT 流在 $5B_{BDP}$ 缓冲区条件下进行带宽竞争,4 种算法中不同 RTT 流的吞吐量的变化趋势和公平指数见图 7。

图 7(a)为 BBR 算法,随着 RTT 差异的增加,长 RTT 流的吞吐量逐步占据优势,公平指数逐步下降。

当 RTT 比率达到 10 倍时,即 10 ms RTT 流与 100 ms RTT 流竞争时,公平指数仅为 0.59。在图 7(b)中,BBQ 算法中的 RTT 公平性呈现不规律变化。当 RTT 比率在 2~3 之间时,短 RTT 流的吞吐量占据优势地位,但当 RTT 比率大于 3 时,长 RTT 流的吞吐量逐步占据优势地位。图 7(c)的 DA-BBR 算法,不同 RTT 流之间的公平性有所提升,最终公平指数在 0.92 左右。BBR-A 算法见图 7(d),随着 RTT 差异的增加,虽然长 RTT 流的吞吐量占据优势,但其与 10 ms RTT 流之间的吞吐量差异不大,BBR-A 的公平指数可以维持在 0.95 左右。BBR-A 算法通过自适应的调整不同 RTT 流之间的起搏增益,可以使不同 RTT 数据流保持较高的公平性,尤其当 RTT 比率大于 4 时,BBR-A 算法的公平指数最高。

3.3 重传率

重传率是重传网络包的比例,链路中的重传率过高一般表示网络质量差,会影响数据的传输效率。通过对比缓冲区大小和竞争数据流的数量对数据包重传的影响,可以验证不同算法的重传率。实验发送方使用不同的算法将单个或多个数据流传输到接收方,缓冲区大小设置分别为 $0.1B_{BDP}$ 和 $1B_{BDP}$ 。4 种算法的重传率见图 8,起始点为单个 10 ms RTT 流的重传率,终点为 100 个 10 ms RTT 流的重传率。

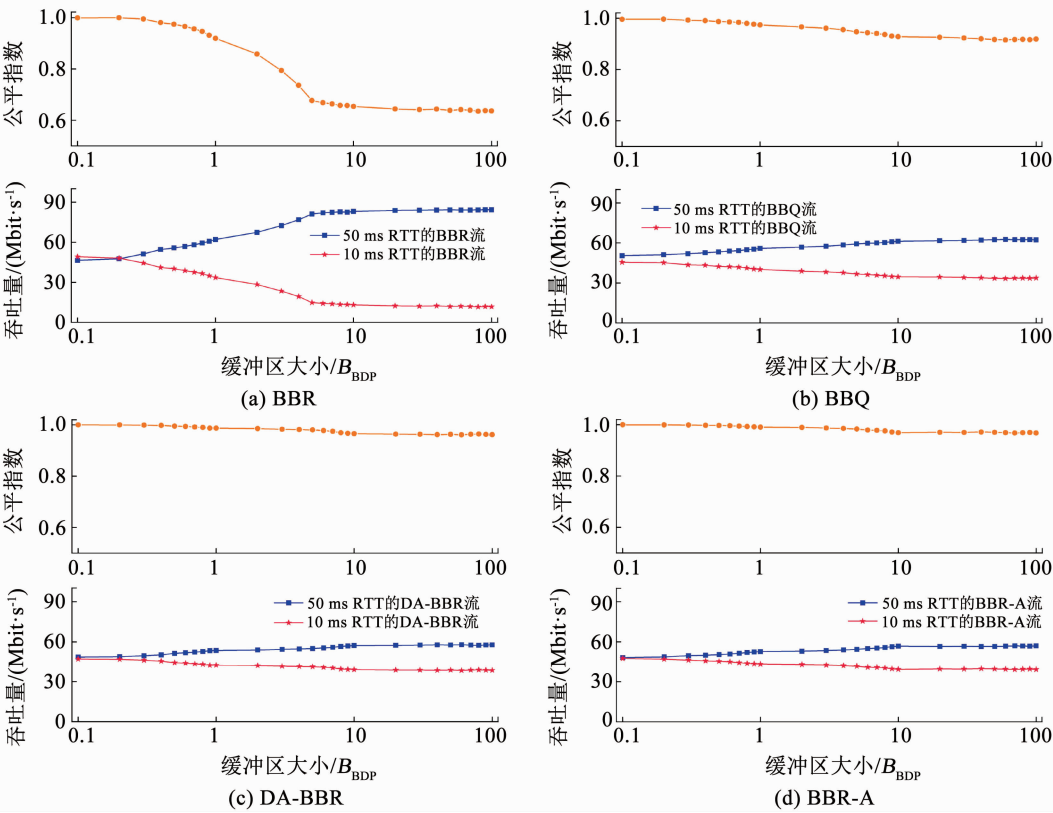


图 6 不同缓冲区大小下,10 ms RTT 流与 50 ms RTT 流竞争时的平均吞吐量和公平指数

Fig. 6 Average throughput and fairness index of 10 ms RTT flow competing with 50 ms RTT flow with different buffer sizes

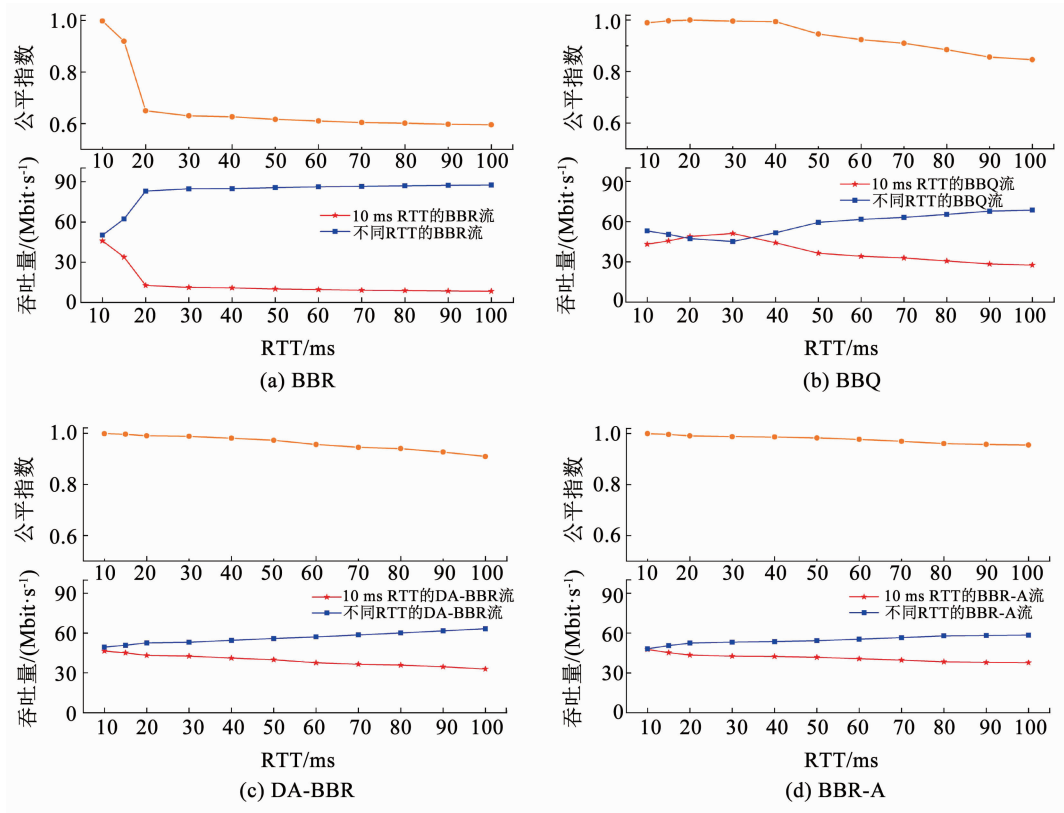


图 7 $5B_{BDP}$ 缓冲区大小下,10 ms RTT 流与不同 RTT 流共存时的平均吞吐量和公平指数

Fig. 7 Average throughput and fairness index of 10 ms RTT flow coexisting with different RTT flows in $5B_{BDP}$ buffer

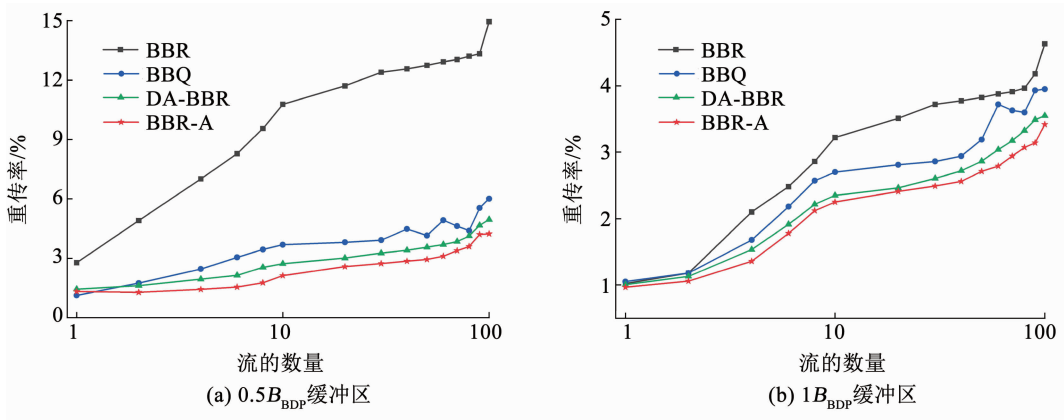


图 8 不同算法在不同缓冲区中的重传率

Fig. 8 Retransmission rates of different algorithms with different buffer sizes

在图 8(a)中,当缓冲区为 $0.1B_{BDP}$ 时, BBR 的重传率明显高于其他算法。在起始点, BBR 的重传率约为 2.8%, BBQ、DA-BBR 和 BBR-A 的重传率约为 1.52%、1.26% 和 1.31%。随着数据流量的增加,数据包出现了大量的重传,当流的数量为 100 时, BBR、BBQ 和 DA-BBR 的重传率分别为 14.9%、6.01% 和 4.86%, 而 BBR-A 的重传率约为 4.24%, 是 4 种算法中最小的。在图 8(b)中,缓冲区大小增加到 $1B_{BDP}$ 。BBR 的重传率相比 $0.1B_{BDP}$ 时明显下降,但仍是 4 种算法中最高的。在起始点, 4 种算法的重传率基本一致, 大约为 1%。随着流的数量增加, 4 种算法重传率都发生了增加。当数据流的

数量为 100 时, BBR 的重传率约为 4.63%, BBQ 和 DA-BBR 的重传率约为 3.95% 和 3.52%, BBR-A 的重传率约为 3.42%。BBR-A 算法加入了丢包反馈, 并建立对称的起搏增益模型, 避免激进的探测带宽方式导致的数据包重传。仿真实验结果证明了 BBR-A 算法可以减少数据包重传次数, 有效提高网络通信效率。

4 结 论

本文在研究 BBR 拥塞控制算法的基础上, 针对原 BBR 算法中的 RTT 公平性问题, 提出了改进算法 BBR-A。BBR-A 算法根据链路中的 RTT, 自适应

调整向上和向下起搏增益,取代了原有的固定起搏增益,从而平衡不同 RTT 流的发送速率。

NS3 仿真实验结果表明,BBR-A 算法中不同的 RTT 流可以公平竞争,公平指数明显高于 BBR。此外,在信道利用率的实验中,与原 BBR 算法相比,BBR-A 算法并没有牺牲信道利用率,甚至在高丢包的情况下信道利用率提升明显。在重传率的实验中,BBR-A 算法的表现也优于 BBR 算法,可以有效降低重传率。这些都表明本文提出的 BBR-A 算法可以有效提升 BBR 算法的网络拥塞控制性能。

参考文献

- [1] FLOYD S, HENDERSON T, GURTOV A. The New Reno modification to TCP's fast recovery algorithm: RFC 3782[R]. [S.l.]: The Internet Society, 2004
- [2] XU Lisong, HARFOUSH K, RHEE I. Binary increase congestion control (BIC) for fast long-distance networks[C]//Proceedings of IEEE INFOCOM 2004. Hong Kong: IEEE, 2004, 4. DOI:10.1109/INFCOM.2004.1354672
- [3] HA S, RHEE I, XU Lisong. CUBIC: a new TCP-friendly high-speed TCP variant[J]. ACM SIGOPS Operating Systems Review, 2008, 42(5): 64. DOI:10.1145/1400097.1400105
- [4] 董瀚泽, 郭志川. BBR 拥塞控制算法在无线网络中的性能改进[J]. 哈尔滨工业大学学报, 2019, 51(11): 63
DONG Hanzhe, GUO Zhichuan. Performance improvement of BBR congestion control algorithm in wireless network[J]. Journal of Harbin Institute of Technology, 2019, 51(11): 63. DOI:10.11918/j.issn.0367-6234.201901020
- [5] CARDWELL N, CHENG Y, GUNN C S, et al. BBR: congestion-based congestion control[J]. Communications of the ACM, 2017, 60(2): 58. DOI:10.1145/3009824
- [6] HOCK M, BLESS R, ZITTERBART M. Experimental evaluation of BBR congestion control[C]//Proceedings of 2017 IEEE 25th International Conference on Network Protocols (ICNP). Toronto: IEEE, 2017: 1. DOI:10.1109/ICNP.2017.8117540
- [7] MA Shiyao, JIANG Jingjie, WANG Wei, et al. Fairness of congestion-based congestion control: experimental evaluation and analysis[Z]. arXiv:1706.09115, 2017. DOI:10.48550/arXiv.1706.09115
- [8] SCHOLZ D, JAEGER B, SCHWAIGHOFER L, et al. Towards a deeper understanding of TCP BBR congestion control[C]//Proceedings of 2018 IFIP Networking Conference (IFIP Networking) and Workshops. Zurich: IEEE, 2018: 1. DOI:10.23919/IFIPNetworking.2018.8696830
- [9] SASAKI K, HANAI M, MIYAZAWA K, et al. TCP fairness among modern TCP congestion control algorithms including TCP BBR[C]//Proceedings of 2018 IEEE 7th International Conference on Cloud Networking (CloudNet). Tokyo: IEEE, 2018: 1. DOI:10.1109/CloudNet.2018.8549505
- [10] JAEGER B, SCHOLZ D, RAUMER D, et al. Reproducible measurements of TCP BBR congestion control[J]. Computer Communications, 2019(144): 31. DOI:10.1016/j.comcom.2019.05.011
- [11] WARE R, MUKERJEE M K, SESHAN S, et al. Modeling BBR's interactions with loss-based congestion control[C]//Proceedings of the Internet Measurement Conference. New York: ACM, 2019: 137. DOI:10.1145/3355369.3355604
- [12] TAO Yuecheng, JIANG Yingjie, MA Shaya, et al. Unraveling the RTT-fairness Problem for BBR: a queueing model[C]//Proceedings of 2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi: IEEE, 2018: 1. DOI:10.1109/GLOCOM.2018.8647260
- [13] 贾茗涵. 无线网络中 TCP-BBR 算法公平性的研究[D]. 大连: 大连理工大学, 2020
JIA Minghan. Research of fairness of TCP-BBR algorithm in wireless network[D]. Dalian: Dalian University of Technology, 2020. DOI:10.26991/d.cnki.gdllu.2020.000790
- [14] KIM G H, CHO Y Z. Delay-aware BBR congestion control algorithm for RTT fairness improvement[J]. IEEE Access, 2019(8): 4099. DOI:10.1109/ACCESS.2019.2962213
- [15] PAN Wansu, TAN Haibo, LI Xiru, et al. Improved RTT fairness of BBR congestion control algorithm based on adaptive congestion window[J]. Electronics, 2021, 10(5): 615. DOI:10.3390/electronics10050615
- [16] KLEINROCK L. Power and deterministic rules of thumb for probabilistic problems in computer communications[C]//Proceedings of the International Conference on Communications. Piscataway: IEEE, 1979, 3(43): 1
- [17] BRAKMO L S, PETERSON L L. TCP Vegas: end to end congestion avoidance on a global Internet[J]. IEEE Journal on Selected Areas in Communications, 1995, 13(8): 1465. DOI:10.1109/49.464716
- [18] AFANASYEV A, TILLEY N, REIHER P, et al. Host-to-host congestion control for TCP[J]. IEEE Communications Surveys & Tutorials, 2010, 12(3): 304. DOI:10.1109/SURV.2010.042710.00114
- [19] YANG Ming, YANG Peng, WEN Chaozhou, et al. Adaptive-BBR: fine-grained congestion control with improved fairness and low latency[C]//Proceedings of 2019 IEEE Wireless Communications and Networking Conference (WCNC). Marrakesh: IEEE, 2019: 1. DOI:10.1109/WCNC.2019.8885527
- [20] JAIN V, MITTAL V, TAHILIANI M P. Design and implementation of TCP BBR in ns-3[C]//Proceedings of the 10th Workshop on ns-3. New York: ACM, 2018: 16. DOI:10.1145/3199902.3199911
- [21] ZHANG Songyang. An evaluation of BBR and its variants[Z]. arXiv:1909.03673, 2019. DOI:10.48550/arXiv.1909.03673
- [22] JAIN R, CHIU D M, HAWE W. A quantitative measure of fairness and discrimination[Z]. arXiv:cs/9809099, 1984. DOI:10.48550/arXiv.cs/9809099

(编辑 苗秀芝)