

一种异构网络 TCP 拥塞控制算法

王志明* 曾孝平 刘学 陈礼 郭博文

(重庆大学通信工程学院 重庆 400044)

摘要: 针对互联网中端对端带宽、时延和丢包率等的差异性日益加剧, 导致 TCP 传输性能严重退化, 该文提出一种链路自适应 TCP 拥塞控制算法(INVS)。INVS 在拥塞避免阶段初期采用基于指数函数的凸窗口增长函数, 以提高链路利用率; 在窗口增长函数中引入了自适应增长因子实现窗口增长速率与链路状态相匹配; 采用了自适应队列门限的丢包区分策略以提高无线环境下 TCP 的性能。性能分析和评估表明, INVS 提高了 TCP 拥塞控制算法的吞吐量、公平性、链路利用率和 RTT 公平性。

关键词: 异构网; 传输控制协议; 拥塞控制; 时延带宽积

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2016)04-0780-07

DOI: 10.11999/JEIT150774

TCP Congestion Control Algorithm for Heterogeneous Networks

WANG Zhiming ZENG Xiaoping LIU Xue CHEN Li GUO Bowen

(College of Communication Engineering, Chongqing University, Chongqing 400044, China)

Abstract: The end-to-end bandwidth, the Round Trip Time (RTT) and the packet loss rate vary over many orders of magnitude in Internet, which cause significant performance degradation of Transmission Control Protocol (TCP). To cope with this, a path-adaptive TCP congestion control protocol, called INverse Sigmoid (INVS), is proposed. INVS employs an exponential function based convex growth function of congestion window to improve the path utilization. An adaptive increase factor in the growth function is introduced to match the window growth rate with the path condition. An adaptive queue-threshold in the loss distinguishing scheme is used to enhance the performance of TCP over wireless links. The performance analysis and evaluation show that INVS improves the throughput, the fairness, the total link utilization and the RTT-fairness of TCP congestion control algorithm.

Key words: Heterogeneous networks; Transmission Control Protocol (TCP); Congestion control; Bandwidth Delay Product (BDP)

1 引言

随着通信技术的发展和普及, 传统的通信网络已发展成为包含有线网络、无线网络和卫星网络等多种 IP 网络的全球异构互联网^[1]。高速光链路、卫星长时延链路和各种无线链路的共存使得链路带宽在无线链路中比较稀缺, 在光纤链路中非常充足; 端对端往返时延(Round Trip Time, RTT)在有线网络中小到甚至低于 1 ms, 在卫星网络中大到甚至超

过 1 s; 链路丢包率在光纤链路上可能低于 10^{-6} , 在无线链路上则可能高于 1%^[2]。简言之, 互联网中端对端带宽、RTT 和丢包率的差异很大, 这也是其异构性的具体体现。TCP 在尽力交付的 IP 网络中为上层业务提供可靠的端到端传输服务, 承载着约 90% 的互联网流量。然而, 随着互联网异构性的增加, TCP 协议的传输性能面临严峻挑战^[2]。目前, 网络异构性已成为互联网拥塞控制研究的热点。

目前已经有大量针对异构性的网络拥塞控制算法研究, 但大多数只针对异构性某一方面或需要网络辅助。如针对 RTT 差异性造成的公平性问题, 文献[3]提出了 Hybla, 大大改善了 RTT 公平性, 但在高速链路和高随机丢包链路上 Hybla 却不能充分利用网络带宽。针对带宽和 RTT 的差异性造成传统 TCP 在高速和大时延, 即大时延带宽积(Bandwidth-Delay Product, BDP)^[4], 网络中不能充分利用带宽的改进算法有 HS-TCP, STCP,

收稿日期: 2015-06-29; 改回日期: 2015-11-17; 网络出版: 2016-01-04

*通信作者: 王志明 jamewzm@163.com

基金项目: 国家自然科学基金(61171089, 61302054), 国家自然科学基金重大项目重大研究计划(91438104), 重庆大学研究生科研创新基金(CYS14005)

Foundation Items: The National Natural Science Foundation of China (61171089, 61302054), The Training Program of the Major Research Plan of the National Natural Science Foundation of China (91438104), The Scientific Research Innovation Program of Chongqing University for Graduate Students (CYS14005)

Illinois, CUBIC 和 Compound TCP(CTCP)等^[4-6], 其中 CUBIC 和 CTCP 已经在互联网上广泛部署使用。然而, 这些算法不适用于随机丢包率高的无线链路, 同时也存在公平性问题。如 CUBIC, STCP 和 HS-TCP 等在与 TCP Reno 共享链路时, 存在严重的公平性问题^[7]; CTCP 与 CUBIC 等算法共享链路时, 不足以竞争到公平的带宽资源^[7]。文献[8]提出了基于时延和丢包率的拥塞控制方法, 其以丢包率作为衡量拥塞的指标, 但这种方法在有随机丢包的无线网络中易造成误判。其次, 由于 LTE 和 WIFI 等网络的无线链路丢包率远高于光纤链路, 特别是 3G 网络无线链路在信号较好时丢包都高达 1%, 手机、平板等无线设备的 TCP 协议(大多数默认采用 CUBIC)传输效率遭受严重影响^[9,10]。针对丢包率的差异性, 如无线链路丢包造成 TCP 性能退化的问题^[4], 目前改进算法有 Veno, JTCP, Westwood 等, 这些算法采用固定队列长度或者时延来区分拥塞丢包和非拥塞丢包^[4]。然而, 随着链路缓存大小差异性的增加^[11,12], 采用固定队列(或时延)门限区分丢包方法的准确性日趋下降。此外, 也有一些算法采用了特殊策略, 如 PEPsal 和 indirect-TCP 将 TCP 连接分成几段以屏蔽无线链路丢包对 TCP 性能的影响^[4]; TCP-Jersey 采用路由辅助的显式拥塞通告策略判定丢包类型^[4]; 网络编码 TCP 通过对数据包进行编码抵抗随机丢包^[4]。由于部署应用对网络的特殊要求, 这些特殊策略难以在互联网上被广泛使用。

在同时考虑带宽、时延和丢包差异性方面, 文献[13]提出了一种根据链路类型选择对应拥塞控制算法的策略, 即卫星链路用 Vegas, 无线链路用 Westwood、高速链路用 HS-TCP 和一般链路用 NewReno。由于其链路信息的获取需要中继路由器将链路类型加入 IP 报文, 在互联网上部署应用的非常困难。文献[14]基于时延抖动将网络拥塞分为 5 级以区分链路丢包、一般拥塞和严重拥塞。由于 TCP BRJ (enhanced TCP congestion control based on Bandwidth estimation and RTT Jitter for heterogeneous networks)完全依赖于时延抖动, 在异构网络中不同链路的时延抖动差别很大, 采用固定时延抖动分级的可靠性尚待验证。因此, 本文提出一种链路自适应 TCP 拥塞控制算法(INVerse Sigmoid, INVS), 能够根据链路自适应调整窗口增长速率和退避策略, 避免 TCP 由于不同链路带宽和时延的差异性以及高无线链路丢包造成 TCP 传输性能退化。此外, INVS 只需修改发端协议, 易于在互联网上逐步部署应用。

2 新的拥塞控制算法

为实现拥塞控制策略与链路状态相匹配以取得较好的利用率和公平性, INVS 采用基于指数函数的自适应窗口增长函数, 并引入了自适应队列门限的丢包区分策略。

2.1 拥塞窗口增长函数

首先, 为了保证在大 BDP 网络中的传输效率, INVS 在拥塞避免阶段 (Congestion Avoidance Phase, CAP) 初期采用基于指数函数的凸窗口增长函数, 在探测网络阶段采用凹窗口增长函数。基于指数函数的凸窗口增长函数允许 TCP 发送端在 CAP 初期快速增长拥塞窗口以尽快充分利用带宽, 当接近拥塞时缓慢增长窗口(如图 1)。拥塞避免阶段初期 t 时刻的窗口为

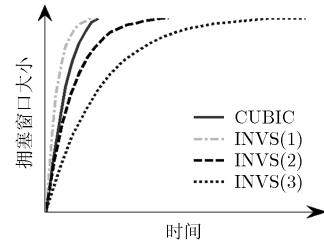


图1 凸窗口增长函数

$$cwnd(t) = cwnd_{sp} (1 - (1 - \beta)\alpha^t), \quad 0 < \alpha < 1 \quad (1)$$

其中, β 是乘性减因子, t 为从 CAP 开始所经历的时间, $cwnd_{sp}$ 为链路资源已充分利用时(饱和点)的拥塞窗口大小。

在实际操作中, 拥塞控制算法每收到一个 ACK, 更新一次拥塞窗口。INVS 采用如下拥塞窗口更新策略, 实现在 CAP 初期按凸函数增长拥塞窗口和在探测阶段按凹函数增长窗口及其过度。

$cwnd$

$$= cwnd + \begin{cases} \frac{cwnd_{sp} - cwnd}{k \cdot cwnd_{sp}}, & cwnd \leq cwnd_{sp} - 1 \\ \frac{cwnd - cwnd_{sp}}{k \cdot cwnd}, & cwnd \geq cwnd_{sp} + 1 \\ \frac{1}{cwnd}, & cwnd_{sp} - 1 < cwnd < cwnd_{sp} + 1 \end{cases} \quad (2)$$

其中, $cwnd$ 为拥塞窗口大小。 k 为一个自适应增长因子, 与式(1)中 α 的关系为 $\alpha = \beta(k - \beta)/k$ 。

式(2)第 1 个表达式实现式中的凸函数增长过程; 第 2 个表达式实现到达饱和点后的带宽探测过程; 最后一个表达式实现凸函数增长阶段到网络探

测阶段的过度。根据式(2), INVS 在 CAP 开始时, 快速增加 $cwnd$, 当 $cwnd$ 接近饱和点($cwnd_{sp}$)时逐渐减慢; 当 $cwnd$ 超过饱和点时, 逐渐加速探测带宽。饱和点($cwnd_{sp}$)为带宽已充分利用时的窗口值。换言之, 如果发送端继续增大拥塞窗口, 将可能发生拥塞; 否则说明仍有可用带宽。

式(2)保证了在给定链路状态下 TCP 的传输效率。在异构的互联网中, TCP 业务流既可能由高速光纤链路承载, 也可能由低速无线链路或大时延的洲际链路承载。然而, 现有大多数 TCP 拥塞控制算法的窗口增长策略独立于链路状态, 使得在高速和大时延等大 BDP 网络中, 拥塞窗口增长耗时长, 不能充分利用链路带宽; 在低速小时延网络中, 窗口增长过快, 导致网络频繁拥塞。INVS 引入了自适应增长因子 k , 实现在大 BDP 网络中以较快的速率增加窗口, 在低速小时延网络中以较小的速率增加窗口, 即窗口增长速率与链路状态相匹配。图 1 给出了采用基于指数函数的凸窗口增长函数与采用三次方函数(CUBIC)的对比。图中 INVS(1), INVS(2)和 INVS(3)分别为采用不同参数 k 的窗口增长过程函数。对比 INVS(1)和 CUBIC 可以看出, 在增长阶段耗时相同时, 采用基于指数函数的凸窗口函数效率更高。从图中可以看出, 增大因子 k , 采用指数函数可使得 TCP 拥塞周期变长, 而且会较长时间处于接近饱和状态, 这既可以避免频繁拥塞, 又保障传输效率。

链路状态参数主要有端对端带宽、RTT 和丢包率。对于链路(随机)丢包, INVS 采用改进的丢包区分策略避免无线链路丢包造成不必要的拥塞退避。因此, 为实现因子 k 根据链路自适应调整, k 取为链路带宽和最小 RTT 的函数, 如式(3)。其中采用对数函数实现将带宽和 RTT 的大范围变化转化为较小范围的变化, 使得 k 可作为调节增长速率的比例因子。在不同链路上, INVS 能够通过因子 k 获得与其链路带宽和 RTT 相匹配的窗口增长速率。

$$k = c(\log_2(r_{bw} r_{rtt}^\gamma) + 1), \quad 0 < \gamma < 1 \quad (3)$$

其中, $r_{bw} = \max\left[\frac{BW_{ref}}{BW_{est}}, 1\right]$, $r_{rtt} = \max\left[\frac{RTT_{ref}}{RTT_{min}}, 1\right]$, c 是常数比例因子, 因子 γ 用于减小 RTT 增大的不利影响, BW_{ref} 是参考带宽(默认值为 500 Mbps), BW_{est} 为估计带宽, RTT_{ref} 是参考 RTT(默认值为 0.5 s), RTT_{min} 为 RTT 最小值。为获得实时带宽估计, TCP Westwood 每收到一个 ACK 估计一次带宽, 而式(3)中 BW_{est} 用于反映链路环境, 采用比较稳定的平均带宽就足够。因此, INVS 一个 RTT 估计一次带宽, 并在估计中采用最小估计周期和低通滤波器消除估

计带宽的抖动, 减小了 ACK 时延抖动的影响, 也避免频繁计算 k 。

从式(3)可以看出, k 随着带宽或 RTT 的增大而减小。带宽或者 RTT 的增大会导致网络 BDP 的增加, 即允许更多的报文同时在网络中传输。为了更有效充分利用网络资源, 每收到一个 ACK, $cwnd$ 增加量随着网络 BDP 的增大而增大, 结合式(2)即 k 应减小。因此, 在式(3)中, 当 BDP 增大时, k 减小。当 $BW_{est} > BW_{ref}$ 且 $RTT_{min} > RTT_{ref}$, k 恒为 c 。这时由于窗口增长速率已经很快, 即使带宽和 RTT 增大, k 也不再减小。

其次, 带宽增加和时延增加都会导致 BDP 增加, 但 RTT 增大也会导致接收到网络拥塞信息反馈的时延增加, 拥塞信息延迟会造成加剧每一次拥塞的程度, 导致更多的丢包。因此, INVS 在式(3)中引入了因子 γ (默认为 0.75)来降低 RTT 增大的不利影响。 γ 越小, RTT 增大的不利影响越小。

此外, 由于在小时延带宽网络中, k 会很大, 使得拥塞窗口的增长速率与 TCP Reno 接近, 甚至会小于 TCP Reno。因此, 为保证与 TCP Reno 的友好性, 当 INVS 的增长速率小于 TCP Reno 时, INVS 使用 TCP Reno 的窗口值。TCP Reno 的拥塞窗口表示为

$$W_{TCP}(t) = 3 \frac{1+\beta}{1-\beta} \frac{t}{RTT} + \beta \cdot cwnd_{sp} \quad (4)$$

其中, t 是 CAP 开始的时间。

2.2 丢包响应及丢包区分策略

在收到 3 个重复 ACK 时, TCP 会执行拥塞退避策略。针对不同类型的丢包, INVS 按照式(5)确定新的 $cwnd$, $ssthresh$ 和 $cwnd_{sp}$:

$$cwnd = ssthresh = \begin{cases} \beta \cdot cwnd, \\ \text{buffer}_{est} \geq \min(\delta, \text{maxbuffer}) \\ \min(BDP_{est}, cwnd), \\ \text{其它} \end{cases} \quad (5)$$

其中, 退避因子 β 默认为 0.75, BDP_{est} 是估计的链路 BDP。 buffer_{est} 是当前网络中排队报文数的估计值, 通过当前 RTT、最小 RTT 和估计带宽获得。 δ 是预设的大容量链路的队列溢出门限, maxbuffer 以 buffer_{est} 的最大值作为队列门限, 超时后重新估计。

在丢包区分策略中, 由于不同链路缓存大小差异大, 为避免与固定门限值相比较造成误判, INVS 以 $\min(\delta, \text{maxbuffer})$ 作为新的队列门限值来区分拥塞丢包和非拥塞丢包。 maxbuffer 用于确保丢包分类方法在小缓存链路上能正确判定拥塞, 如式(5)所

示。当收到 3 个重复 ACK，且当前队列长度大于 $\min(\delta, \text{maxbuffer})$ ，则判定当前丢包由拥塞引起，并采用传统的退避机制；否则，判为丢包属于非拥塞丢包。cwnd 和 ssthresh 的值设为 BDP_{est} 和 cwnd 的最小值。

当有新数据流的加入或数据传输终止时，链路带宽等会发生变化。cwnd_{sp} 用于跟踪网络饱和时的 cwnd。当丢包发生时，cwnd < cwnd_{sp} 表示在这一个拥塞避免阶段中拥塞发生比上一次更早，这就说明可用的带宽等资源正在减少。为加快拥塞窗口的收敛，当 cwnd < cwnd_{sp} 时，当前跟踪到的网络饱和点 cwnd_{sp} 向下调整，设为小于 cwnd 的值(如式(6))；否则 cwnd_{sp} 就设为当前窗口大小 cwnd 以跟踪当前的网络饱和点。

$$\text{cwnd}_{\text{sp}} = \begin{cases} \frac{1+\beta}{2} \cdot \text{cwnd}, & \text{cwnd} < \text{cwnd}_{\text{sp}} \\ \text{cwnd}, & \text{其它} \end{cases} \quad (6)$$

3 性能分析

3.1 INVS 的稳态吞吐量

TCP 稳态吞吐量建模分析基于 TCP 的长时间传输、不考虑慢启动阶段和超时的情况，通过分析拥塞避免阶段和发现丢包后的快重传快恢复阶段发送报文的总数及其持续时间，得出 TCP 的稳态吞吐量。

在给定链路时，链路带宽和最小 RTT 不变，因此 k 可看作常数。稳态时，cwnd 周期性地从 $\beta \cdot \text{cwnd}_{\text{sp}}$ 增长到 cwnd_{sp}。其中，cwnd_{sp} 为拥塞避免阶段(CAP) cwnd 的最大值。从式(2)可以得出，在第 i 轮 RTT 开始时，cwnd 可以表示为

$$\text{cwnd}(i) = \text{cwnd}_{\text{sp}} \left(1 - \left(\frac{\beta(k-\beta)}{k} \right)^{i-1} (1-\beta) \right) \quad (7)$$

其中， i 表示从 CAP 开始经过的 RTT 轮数。

记 n 为 cwnd 增长到 cwnd_{sp} 所经过的 RTT 轮数。根据式(2)可以得出，当 $\text{cwnd}(i) \geq \text{cwnd}_{\text{sp}} - 1$ 时，在第 i 轮结束时 cwnd 就会等于 cwnd_{sp}。因此，轮数 n 可以表示为

$$n = \log_{\frac{k}{\beta(k-\beta)}} (\text{cwnd}_{\text{sp}} (1-\beta)) \quad (8)$$

因此，一个 CAP 阶段发送的总报文数为

$$\begin{aligned} Y &= \sum_{i=1}^n \text{cwnd}(i) \\ &= \text{cwnd}_{\text{sp}} \left[n - \frac{k((1-\beta)\text{cwnd}_{\text{sp}} - 1)}{(k - k\beta + \beta^2)\text{cwnd}_{\text{sp}}} \right] \end{aligned} \quad (9)$$

稳态时，一个 CAP 阶段可以发送的报文总数也可用网络拥塞概率 p 表示为

$$Y = 1/p \quad (10)$$

由式(9)和式(10)可以得出

$$\text{cwnd}_{\text{sp}} \left[n - \frac{k((1-\beta)\text{cwnd}_{\text{sp}} - 1)}{(k - k\beta + \beta^2)\text{cwnd}_{\text{sp}}} \right] = \frac{1}{p} \quad (11)$$

因此，

$$\text{cwnd}_{\text{sp}} = \frac{k(1-\beta-p) + \beta^2}{p((k - k\beta + \beta^2)n - k(1-\beta))} \quad (12)$$

将式(8)代入式(12)，利用 Lambert W 函数是 $f(W) = We^W$ 的反函数可以求得 cwnd_{sp} 的表达式为

$$\text{cwnd}_{\text{sp}} = \frac{(k(1-\beta-p) + \beta^2) \ln \left(\frac{k}{\beta(k-\beta)} \right)}{p(k - k\beta + \beta^2) \text{lambertW}(f)} \quad (13)$$

其中，

$$\begin{aligned} f &= \frac{k(1-\beta-p) + \beta^2}{p(k - k\beta + \beta^2)} (1-\beta) \ln \left(\frac{k}{\beta(k-\beta)} \right) \\ &\quad \cdot e^{-\frac{k(1-\beta)}{k - k\beta + \beta^2} \ln \left(\frac{k}{\beta(k-\beta)} \right)} \end{aligned}$$

因此，可以推得平均吞吐量的表达式为

$$\text{Th} = \frac{Y}{n \cdot \text{RTT}} = \frac{\ln \frac{k}{\beta(k-\beta)}}{p \cdot \text{RTT} \cdot \ln(\text{cwnd}_{\text{sp}} (1-\beta))} \quad (14)$$

图 2 所示为不同网络拥塞概率下 INVS 与其 TCP 拥塞控制协议的响应函数^[5]。由于目前尚无 CUBIC, Illinois, Hybla 和 Westwood 的响应函数，因此后文通过仿真对比分析。AIMD(32, 0.125)表示使用增量因子 32 和退避因子为 0.125。INVS(500)和 INVS(50)分别表示在瓶颈带宽为 500 Mbps 和 50 Mbps INVS 流的结果。从图中可以看出，给定拥塞概率下，INVS 流在高带宽网络中取得更大的吞吐量性能。

3.2 TCP 公平性

为分析 INVS 的公平性，图 3 给出了相同链路上两个 INVS 流的窗口变化矢量图^[15]。假设 1 号流先启动，已取得较大 cwnd，2 号流启动晚，瓶颈链路拥塞后退避至图中 A 点。根据式(6)，1 号流的 cwnd_{sp} 设为 $0.5(1+\beta)\text{cwnd}_1$ ，2 号流的设为 cwnd₂。cwnd₁ 和 cwnd₂ 分别为上一 CAP 中的最大拥塞窗口。根据式(8)，2 号流比 1 号流先将 cwnd 增加到 cwnd_{sp}。直到再次拥塞时(图中 B 点)，2 号流就获得了 1 号流释放的带宽。退避到 C 点后，在下一 CAP 拥塞时(D 点)，根据式(2)，2 号流获得更多的带宽。通过数次拥塞和退避，1 号流逐步释放带宽给 2 号流，直到它们近似平等地共享带宽。

图 4 给出了同一网络下的两个 INVS 流的拥塞

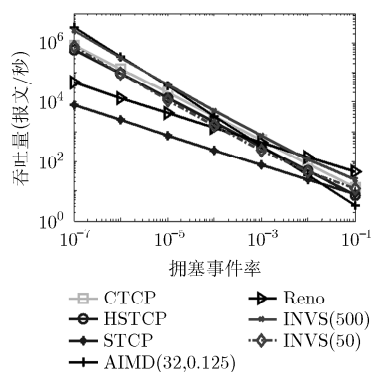


图 2 TCP 协议的响应函数

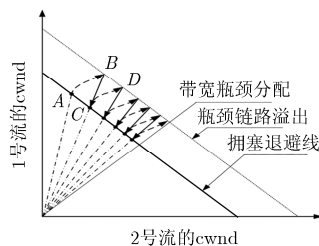


图 3 公平性示意图

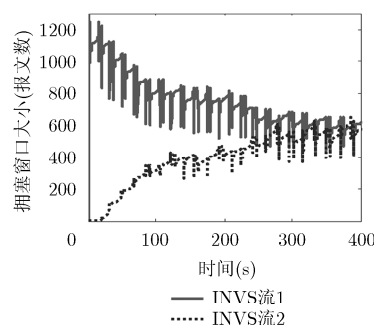


图 4 两个 INVS 流的拥塞窗口

窗口。仿真场景是瓶颈容量为 100 Mbps, RTT 为 80 ms 的哑铃网络。从图中可以看出, 两个 INVS 流的拥塞窗口从 300 s 开始趋于相同。仿真结果也验证了当丢包发生时, 采用式(6)加快了竞争流间的窗口收敛速度。

4 性能评估

为评估 INVS 在异构性网络中的性能, 本节使用了哑铃拓扑和混合拓扑两种网络拓扑。哑铃拓扑网络采用 100 Mbps 的瓶颈带宽和 1000 Mbps 的支路带宽。混合网络拓扑和配置如图 5 所示。仿真中节点缓冲大小默认为链路 BDP。

4.1 单个 TCP 流场景

为评估无线链路丢包下 INVS 的性能, 本节分别在高速网络和卫星网络两种网络拓扑仿真了单个 INVS 流的吞吐量。哑铃网络拓扑用于模拟高速网络, 其瓶颈带宽为 500 Mbps, RTT 为 20 ms, 瓶颈链路缓存为 4 Mbit。混合网络拓扑用于模拟卫星大时延网络, 其中卫星前向链路(Satellite Forward Link, SFL)的瓶颈带宽为 4 Mbps。图 6 和图 7 分别为高速网络和卫星网络下随着丢包率增加时单个流的平均吞吐量。从图中我们可以得出:

(1)随机丢包率越高, 所有协议的性能都越来越

差;

(2)当链路丢包为 0 时(即仅有拥塞丢包), Reno 不能够充分利用网络;

(3)对比图 5 和图 6 发现, Hybla 和 CUBIC 在有链路丢包的高速网络中不能充分利用网络带宽。这是因为 Hybla 和 CUBIC: (1)窗口增长独立于 RTT; (2)均不区分拥塞丢包和链路丢包。

(4)随着随机丢包率的增加, INVS 在高速网络和卫星网络中均取得较好的性能。

4.2 两个 TCP 流场景: RTT 公平性

时延差异是异构网络中的一个重要方面, 主要影响 TCP 拥塞控制算法的公平性, 本节采用哑铃网络拓扑评估算法的 RTT 公平性, 瓶颈带宽为 100 Mbps, 支路带宽为 1000 Mbps, 瓶颈链路缓存大小为 3 Mbit, 仿真时间持续 600 s。两个 TCP 流共享瓶颈带宽, 1 号流的 RTT 为 80 ms, 2 号流的 RTT 从 20 ms 到 320 ms 变化。2 号流在 1 号流启动 20 s 后启动。

图 8 为两个流中吞吐量中较高的与较低的比值。图中比值越接近 1, 则该协议的 RTT 公平性越好。从图中可以看出, 当两个流的 RTT 相等时, Reno 和 CTCP 的公平性最好; 由于采用完全 RTT

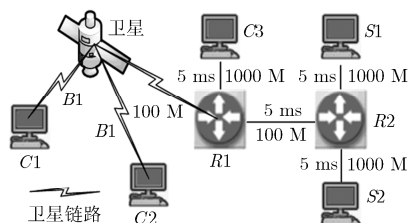


图 5 混合网络拓扑

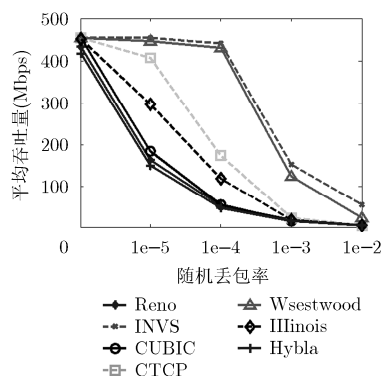


图 6 高速网络吞吐量

独立的增长速率, Hybla 的 RTT 公平性最好; 当两个流的 RTT 差增大时, INVS 的公平性比除 Hybla 外的其他协议公平性好, 这是因为 INVS 在窗口增长阶段引入了自适应增长因子等。

4.3 混合 TCP 流场景

本节在混合网络拓扑下采用混合 TCP 背景流评估了 INVS 的性能。为模拟实际网络流量, 基于文献[16]对互联网上服务器使用 TCP 协议的估计结果, 结合 Windows7 以上系统(默认 CTCP)和智能手机(默认 CUBIC)的广泛使用, 背景流由 Reno 流(20%), Cubic 流(50%)和 CTCP 流(30%)组成。仿真中在客户端 C1 和服务器 S1 之间建立 5 个测试协议流, 在 C3 和 S2 之间建立 20 个背景流, 所有的 TCP 流都在 0 s 到 5 s 内随机启动。当 SFL 带宽分别为 10 Mbps, 40 Mbps 和 80 Mbps 时, 卫星 TCP 流的平均带宽为 2 Mbps, 8 Mbps 和 16 Mbps, 有线瓶颈链路上 TCP 流的平均带宽为 4 Mbps。

图 9 和 10 分别为卫星链路上 TCP 流的平均吞吐量。对比可见, 随着 SFL 带宽的增加, 即 SFL 不再是卫星 TCP 流的瓶颈链路, 卫星 INVS 流逐渐获得与有线 TCP 流相近的平均吞吐量; 当 SFL 带宽充足时, 卫星 CUBIC 和 Hybla 流严重抢

占了有线链路 TCP 流的带宽, 而其它卫星 TCP 流(CTCP, Illinois, Reno 和 Westwood)又不足以与有线链路背景流竞争。图 11 为公平性指标, 结果显示, 当 SFL 带宽为 10 Mbps 和 80 Mbps 时, INVS 的公平性最好; 当 SFL 带宽为 40 Mbps 时 CUBIC 的公平性最好。图 12 为异构网络下总的链路利用率, 为了衡量卫星链路和有线链路的总利用率, 结果为卫星链路利用率和有线链路利用率的平均值。从图中可以看出, INVS 的总链路利用率最高。总的来说, 在混合网络拓扑下 INVS 提高了混合 TCP 流的公平性和总链路利用率。

5 总结

本文提出了一种异构网络的自适应 TCP 拥塞控制算法——INVS。INVS 在拥塞避免阶段初期采用基于指数函数的凸窗口增长函数, 提高了链路利用率; 在窗口增长函数中引入了自适应增长因子, 实现了窗口增长速率与链路状态相匹配, 并引入了自适应队列门限的丢包区分策略区分拥塞丢包和非拥塞丢包, 提高了无线高链路下 TCP 的性能。此外, INVS 还采用了加速拥塞窗口收敛的策略, 并且也只需修改发送端的 TCP 协议。最后, 性能分析表明 INVS 具有较高的吞吐量响应和良好的公平性。仿真

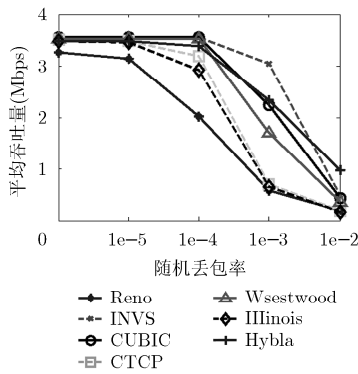


图7 卫星网络吞吐量

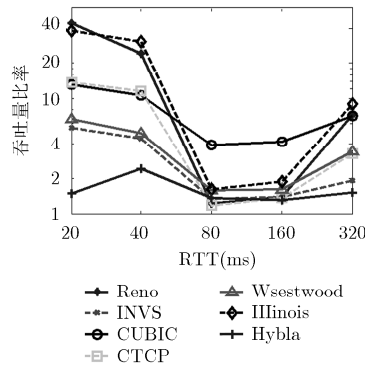


图8 RTT公平性

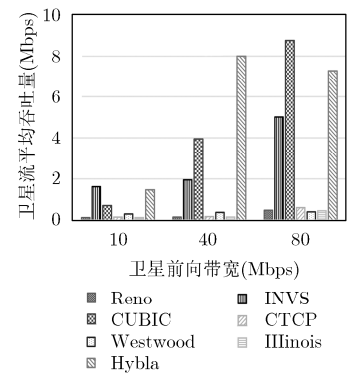


图9 卫星链路吞吐量

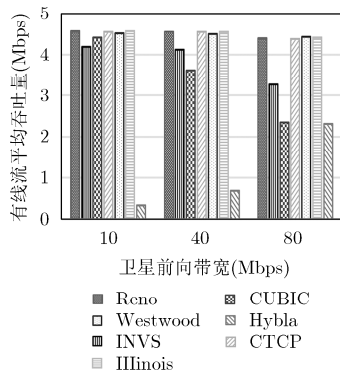


图10 有线链路吞吐量

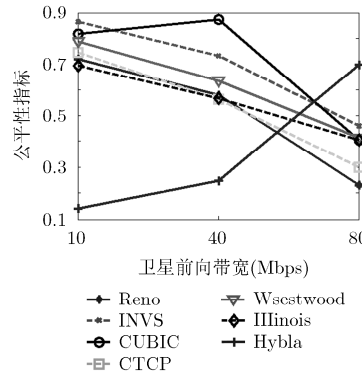


图11 公平性

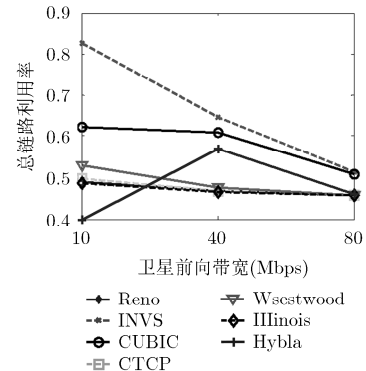


图12 总链路利用率

实验表明：在有无线链路丢包的高速网络和卫星网络的场景中，相对于 Westwood, CUBIC, CTCP 等算法，INVS 的吞吐量性能最好；在两个流和混合流场景中，INVS 取得很好的 RTT 公平性、总链路利用率、有效吞吐量和公平性。

参考文献

- [1] LI P and FANG Y G. On the throughput capacity of heterogeneous wireless networks[J]. *IEEE Transactions on Mobile Computing*, 2012, 11(12): 2073–2086. doi: 10.1109/TMC.2011.239.
 - [2] PAPADIMITRIOU D, WELZL M, SCHARF M, *et al.* Open research issues in Internet congestion control[R]. RFC 6077: Internet Research Task Force (IRTF), 2011.
 - [3] CAINI C and FIRRINCIELI R. TCP Hybla: a TCP enhancement for heterogeneous networks[J]. *International Journal of Satellite Communications and Networking*, 2004, 22(5): 547–566. doi: 10.1002/sat.799.
 - [4] ALRSHAH M A, OTHMAN M, ALI B, *et al.* Comparative study of high-speed Linux TCP variants over high-BDP networks[J]. *Journal of Network and Computer Applications*, 2014, 43: 66–75. doi: 10.1016/j.jnca.2014.04.007.
 - [5] AFANASYEV A, TILLEY N, REIHER P, *et al.* Host-to-host congestion control for TCP[J]. *IEEE Communications Surveys & Tutorials*, 2010, 12(3): 304–342. doi: 10.1109/SURV.2010.042710.00114.
 - [6] MOLIA H K and AGRAWAL R. A conceptual exploration of TCP variants[C]. 2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), Surat, 2014: 1–6. doi: 10.1109/ET2ECN.2014.7044985.
 - [7] WANG J Y, WEN J T, HAN Y X, *et al.* CUBIC-FIT: a high performance and TCP CUBIC friendly congestion control algorithm[J]. *IEEE Communications Letters*, 2013, 17(8): 1664–1667. doi: 10.1109/LCOMM.2013.060513.130664.
 - [8] 谢钧, 俞璐, 金凤林. 基于排队时延和丢包率的拥塞控制[J]. *电子与信息学报*, 2010, 32(9): 2058–2064. doi: 10.3724/SP.J.1146.2009.01228.
XIE J, YU L, and JIN F L. Congestion control based on queuing delay and packet loss probability[J]. *Journal of Electronics & Information Technology*, 2010, 32(9): 2058–2064. doi: 10.3724/SP.J.1146.2009.01228.
 - [9] WAN Y J, ZHANG L, WU Z M, *et al.* Uplink performance in 3G networks[C]. 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), Ottawa, 2014: 1213–1216. doi: 10.1109/WARTIA.2014.6976498.
 - [10] CHEN Y. C, Nahum E M, Gibbens R J, *et al.* Measuring cellular networks: Characterizing 3G, 4G, and path diversity[R]. Technical Report, UMass Amherst Technical Report: UM-CS-2012-022.
 - [11] LI M, LUKYANENKO A, TARKOMA S, *et al.* Tolerating path heterogeneity in multipath TCP with bounded receive buffers[J]. *Computer Networks*, 2014, 64(0): 1–14. doi: 10.1016/j.comnet.2014.01.011.
 - [12] 刘伟彦, 孙雁飞, 张顺颐, 等. 一种参数自适应的主动队列管理算法——自适应 BLUE[J]. *电子与信息学报*, 2009, 31(2): 462–466.
LIU W Y, SUN Y F, ZHANG S Y, *et al.* An adaptive active queue management algorithm——self-adaptive BLUE[J]. *Journal of Electronics & Information Technology*, 2009, 31(2): 462–466.
 - [13] 毕元梅. 异构网络下的 TCP 性能增强技术研究[D]. [硕士学位论文], 重庆大学, 2009.
BI Y M. A study of enhancing TCP performance over heterogeneous networks[D]. [Master dissertation], Chongqing University, 2009.
 - [14] DING N, WU R Q, and JIE H. TCP BRJ: Enhanced TCP congestion control based on bandwidth estimation and RTT jitter for heterogeneous networks[C]. The Proceedings of the Third International Conference on Communications, Signal Processing, and Systems. Switzerland, 2015: 623–632. doi: 10.1007/978-3-319-08991-1_64.
 - [15] BERTRAN R, GONZÁLEZ M, MARTORELL X, *et al.* Counter-based power modeling methods: Top-down vs. bottom-up[J]. *The Computer Journal*, 2013, 56(2): 198–213. doi: 10.1093/comjnl/bxs116.
 - [16] YANG P, SHAO J, LUO W, *et al.* TCP congestion avoidance algorithm identification[J]. *IEEE/ACM Transactions on Networking*, 2014, 22(4): 1311–1324. doi: 10.1109/TNET.2013.2278271.
- 王志明：男，1986 年生，博士，研究方向为 TCP 拥塞控制与建模分析、航空自组网组网及优化。
- 曾孝平：男，1956 年生，教授，研究方向为航空移动通信、计算机网络、信号与信息处理。
- 刘学：男，1983 年生，博士生，研究方向为航空自组网连通性研究、计算机网络。
- 陈礼：男，1982 年生，博士生，研究方向为民用航空移动通信网络。
- 郭博文：男，1992 年生，硕士生，研究方向为航空自组网 MAC 协议及资源调度算法。