

MPTCP 与 CMT-SCTP 拥塞控制机制研究

(刘骥¹ 谭毓银¹ 符发¹ Thomas Dreibholz² 周星¹ 白勇¹ 杨雄¹ 杜文才¹)

(¹海南大学信息科学技术学院, 海口 570228. ²挪威国家实验室, 奥斯陆 0161)

摘要: 多路径 TCP (MPTCP) 与并发多路径传输 SCTP (CMT-SCTP) 是下一代互联网多路径并发传输的两种技术, 也是目前 IETF 国际组织讨论的热点。与传统的 TCP、SCTP 协议比较, 多路径传输协议的拥塞控制会面临公平性的问题。本文回顾了传统 TCP 拥塞控制算法, 针对多路径并发传输的不公平性问题, 基于资源池思想, 引入了慢启动阈值比因子、增长因子、抢占因子导出了解决公平性问题的新算法: CMT/RPv1 CC、CMT/RPv2 CC 与 MPTCP-like CC, 并通过仿真进行了对比分析。

关键词: 多路径传输, 拥塞控制, 公平性问题, 资源池, 增长因子、抢占因子、性能分析

Study on MPTCP and CMT-SCTP Congestion control mechanism

Ji Liu¹, Yuyin Tan¹, Fu Fa¹, Thomas Dreibholz², Xing Zhou¹, Yong Bai¹, Xiong Yang¹, Wencai Du¹

(¹College of Information Science and Technology, Hainan University, Haikou 570228, China)

(²The Simula Research Laboratory, Oslo 0161, Norway)

【Abstract】 Multi-path TCP (MPTCP) and Concurrent Multipath Transfer for SCTP (CMT-SCTP) are two technologies of the concurrent multi-path transfer for NGI, which are hot topics discussed by the IETF. Compared with the traditional TCP and SCTP, the congestion control of multi-path transport protocols will face the problem of unfairness. This paper begins with an overview of the traditional TCP congestion control algorithm, and then in view of the unfairness of concurrent multi-path transfer, basing on the resource pooling method[8], introducing the slow start threshold ratio factor, increase factor, per-flow aggressiveness factor, deducing three new algorithms to solve the problem of unfairness: CMT/RPv1 CC, CMT/RPv2 CC and MPTCP-like CC, and carrying out the contrastive analysis through simulation.

【Key words】 multi-path transfer, congestion control, fairness, resource pooling, increase factor, aggressiveness factor, Performance Analysis

1. 概述

随着网络接入技术的多样性与接入设备的低廉化, 越来越多的终端设备可同时享用多种接入方式 (如: IP 网络、3G 网络、WLAN, 卫星通信) 与 Internet 连接, 例如: 今天, 几乎所有手机、iPad 等移动设备都装备了 WLAN 和 3G/4G 接口, 这种具有多个网络接口的多宿主特性使设备具有了更好的可移动性、快速恢复能力、安全性和负载共享功能。负载共享是其中一个很重要的特性, 它能够聚合不同链路的带宽, 使得设备能获得更

大的网络吞吐量。负载共享功能可以在网络协议栈中的多个层上实现^[1], 目前正在 IETF 讨论的热点是传输层的协议和传输层协议的扩展, 比如多路径的 TCP (MPTCP^[2]) 和并行多路径传输的 SCTP (CMT-SCTP^[3])。

MPTCP (Multipath TCP) 是传统 TCP 协议的扩展, 即在传输层上使用多条路径实现同时进行数据传输的协议。MPTCP 使用 TCP 作为子流传输, 多个路径就意味着有多个 TCP 子流。而 CMT-SCTP (Concurrent Multipath Transfer for SCTP) 则是利用 SCTP 所特有的多宿主特性, 从而使用不同的路径同时进行数据传输^[4]。这两个协议

基金项目: 国家自然科学基金项目: “下一代互联网多宿主系统国际测试床构建与性能分析”(编号: 61363008)、“基于 RSerPool 应用的 SCTP 传输协议传输路径的优化研究”(项目编号: 61163014)、“基于网络效用最大化理论增强无线 Mesh 网 VoIP 传输性能的机制研究”(项目编号: 61261024); 海南省国际科技合作专项: 下一代互联网 SCTP 传输协议测试床构建与性能分析 (项目编号: KJHZ2013-20)。

通讯作者: 周星

作者简介: 刘骥(1990-), 男, 硕士, 研究方向: 下一代互联网多路径传输研究, E-mail: liuji@hainu.edu.cn; 谭毓银, 实验师; 符发, 实验师; 周星, 教授; 白勇, 教授; Thomas Dreibholz, 资深研究员; 杨雄, 教授; 杜文才, 教授。

收稿日期: 修回日期: E-mail: zhouxing@hainu.edu.cn

从理论上都能充分利用终端的多条路径同时进行数据传输, 以实现网络吞吐量的最大化。在实际的对比测试结果中, MPTCP 和 CMT-SCTP 也都能获得吞吐量的提高^[5]。

传输层协议 TCP 与 SCTP 使用的传统拥塞控制算法只适合单路径传输, 并不适合多路径传输, 因为在多路径传输上使用单路径拥塞控制机制^[2]会在多路径共享同一瓶颈链路时使资源分配不公平。因此, 需要适合多路径数据传输的拥塞控制机制。目前针对多路径 TCP 的 MPTCP^[6]和 CMT-SCTP 的 CMT/RP^[7]两种拥塞控制倍受 IETF 组织的关注。二者都使用了资源池 (Resource Pooling, RP) 思想来解决多路径的公平性问题。RP 可以将资源汇聚成一个整体, 协调每一条路径的拥塞控制, 实现负载均衡、提高网络的鲁棒性和最大限度的发挥有限容量的传输效率^[8]。

本文首先综述了 TCP 单路径拥塞控制机制; 然后引出导致多路径并发传输不公平性问题的原因, 采用资源池方法^[15]分析了解决多路径并发传输不公平性问题; 最后对多径拥塞控制算法进行了仿真分析。

2. 拥塞控制机制研究现状

拥塞控制算法一直是网络协议关注的课题。在传统的 TCP/IP 协议栈中, 拥塞控制机制不仅要在点到点的 IP 层上实现, 也在端到端的 TCP 层上实施。在 IP 层的拥塞控制中使用了被动的源站抑制报文机制, 即仅在收到通知后才减缓发送速度; 对于 TCP 的端设备而言, 拥塞表现为时延增加, 大多数传输层协议使用超时重传数据报来应对时延的增加, 而重传又会进一步加剧拥塞, 如不加以抑制, 如此循环往复会导致拥塞崩溃。解决端设备的“拥塞崩溃”的具体方法有: “慢启动”、“拥塞避免”、“快速重传”及“快速恢复”。但对于 MPTCP 与 CMT-SCTP 协议而言, 这些算法仍然有效吗?

2.1 TCP 单路径拥塞控制

2.1.1 慢启动与拥塞避免

传统的 TCP 协议是一种基于窗口的拥塞控制, 其拥塞控制窗口变化的原则被称为 AIMD^[1] (即加性增乘性减), 即通过流量控制来保证网络的稳定性以及带宽享用的公平性。TCP 发送端利用“慢启动 (slow start)”与“拥塞避免 (Congestion avoidance)”算法来控制正在向网络发送的数据量, 并为每个 TCP 连接提供了三个状态变量, 分别是: “接收窗口 r”、“拥塞窗口 c”和“慢启动门限 s”。

当 $c \leq s$ 时, 启用“慢启动”算法;

当 $c > s$ 时, 则使用“拥塞避免”算法;

当 $c = s$ 时, 是“慢启动”与“拥塞避免”阶段的转折点。

利用门限 s, TCP 连接将经历从慢启动到拥塞避免阶段的转换, 从而让发送数据的速度增长规律从指数规律变为线性规律; 当遇到超时重传时又会启用“加速递减”的算法, 并修正 s 的状态值为当前拥塞窗口的一半。

慢启动算法思想: 在新连接开始或拥塞解除后, 都仅以一个最大报文段长度 (Maximum Segment Size, MSS) 作为拥塞窗口 c 的初始值。此后, 发送方每收到一个确认, c 增加 1 个 MSS。当发送方发送第一个报文段 M_1 , 接收方收到后确认 M_1 。发送方收到对 M_1 的确认后, 把 c 从 1 个 MSS 增加到 2 个 MSS, 于是发送 M_2 和 M_3 两个报文段, 以此类推。因此使用慢启动算法后, 每经过一个传输轮次, 拥塞窗口 c 就加倍, 呈指数增长, 传输轮次所经历的时间其实就是往返时间 RTT。虽然慢启动初始窗口看似小, 但由于随后窗口按指数规律增长, 除非报文段丢失, 否则只需 4 个往返时间后, TCP 就可以连续发送 16 个报文段, 即很快就可以达到拥塞窗口的最大值。该算法使得互联网不会在拥塞清除后或新连接突然建立时被增加的通信量所淹没。

拥塞避免算法就是加法增大的思想。当窗口中的所有报文段被确认后, 才将 c 增加一个 MSS。也就是说, 发送方 c 每经过一个往返时间 RTT 仅增加一个 MSS 的大小, 这样 c 的增长规律由慢启动的指数规律变为线性规律, 比慢启动的增长缓慢得多。公式(1)描述了慢启动与拥塞避免算法思想, 其中 c 按分段函数规律增长。

$$c = c + \begin{cases} a \text{ MSS} & c \leq s \\ 1 \text{ MSS} & (c > s) \end{cases} \quad (1)$$

其中: a 是经接收端新确认的报文段个数, 慢启动阶段 $c + a$ 实际上是指加上 a 个 MSS 的大小, 单位是字节。

与此算法同时工作的算法是“加速递减”, 其思想是: 一旦通信在慢启动或拥塞避免阶段出现超时重传时, TCP 立即将慢启动门限值 s 设置为拥塞时的 c 值的一半 (按指数级递减), 如公式(2)所示, 同时将重传定时器的时限加倍 (按指数规律对重传定时器进行补偿)。

2.1.2 快速重传与快速恢复

快速重传与快速恢复是对慢启动与拥塞避免算法的改进。快速重传算法以连续 3 个重复 ACK 的到达作为一个报文段丢失的标志, 从而使 TCP 不用等待重传定时器超时就可重传看似已经丢弃的报文, 因而绕过重传定时器较长的等待而加快了重传的速度。快速恢复旨在

TCP 对带宽有更强的抢占性。据此,在 RFC 4960 标准中的 SCTP 只使用了其主路径,并运用了与 TCP 类似的拥塞控制机制。为了统一 CMT-SCTP 负载共享的扩展, TCP-Friendliness 被作为扩展 CMT-SCTP 协议的一个必须满足的强制性要求。最初定义 CMT-SCTP 协议的一个基本假设是:关联的所有路径都不相交,也就是说用来传输数据包的每个路径必须是不相交的^[3]。这就使得拥塞控制的实现变得容易,即每个路径只需按标准的 SCTP 拥塞控制机制独立处理,这种工作模式类似 SS7 的电话网络。从拥塞控制的角度看,为了避免不公平性,很有必要提前知道各路径是否不相交。图 2 的(a)与(b)说明了多路径是否相交的基本假设,图 2(a)就是共享瓶颈的多路径情况,而图 2(b)则是并发多路径不相交的情况。

3.2.2 基于资源池的方法

RP 方法^[8]可以将所有资源汇集表现为一个单一的资源整体,从而可实现多路径间的负载转移机制,协调每一条路径的拥塞控制,实现负载平衡、提高网络的鲁棒性和最大限度的发挥有限容量的传输效率。将 RP 概念运用到 CMT-SCTP 协议中的意图是:提高吞吐量、解决公平性问题及并平衡拥塞的目标。

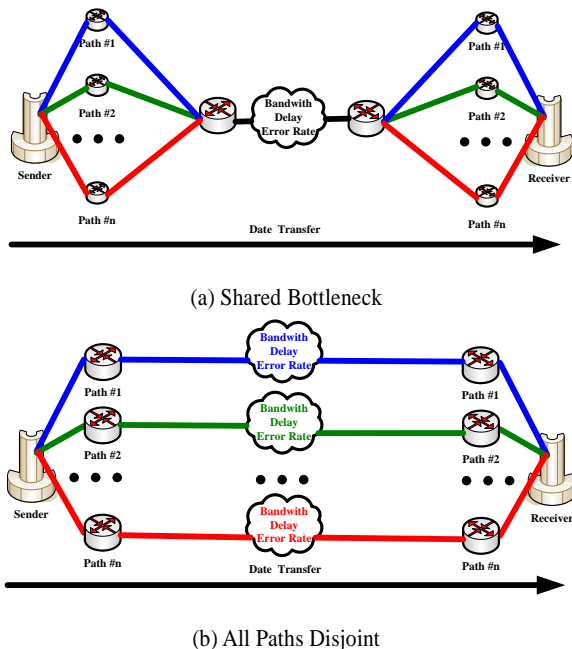


图 2 多路径是否相交的基本假设

为满足这三个目标,多路径传输中每一条支流(subPath)的拥塞窗口都是耦合的,每个 subPath 的拥塞窗口的大小应与标准 TCP Reno 的拥塞窗口的大小一致,

当一条 subPath 收到一个下降信号时,并不是按照当前窗口的大小减少,而是按照发送方的系统速率比例减小拥塞窗口的大小。另外,当有一条路径发生故障,该路径上的流量不应立即转移到其他路径,因为这很可能会导致整个网络范围内的不稳定性,而是逐步地在其他的路径的窗口相应增加多倍往返时间。

RP 是一个通用的方法,目前人们已提出三种平衡拥塞的机制,两个是基于资源池的 CMT-SCTP (分别为 CMT/RP v1 与 CMT/RP v2)算法,第三个是适用于 MPTCP 的拥塞控制 (MPTCP-like)。

3.2.3 CMT/RP v1

CMT/RP v1 算法引入了一个多路径慢启动阈值因子 \hat{s}_p ,并用该因子对前面的 plain CMT 的慢启拥塞窗口和快速重传的拥塞窗口的算法公式(3)与(4)分别进行了修正,如公式(5)与(6)所示。

$$c_p = c_p + \begin{cases} \lceil \hat{s}_p * \min\{a, MTU_p\} \rceil & (c_p \leq s_p) \\ \lceil \hat{s}_p * MTU_p \rceil & (c_p > s_p \wedge p_p > c_p) \end{cases} \quad (5)$$

$$s_p = \max \left\{ \left\lceil c_p - \frac{1}{2} * \sum_i c_i \right\rceil, \left\lceil \hat{s}_p * 4 * MTU_p \right\rceil, MTU_p \right\}$$

$$c_p = \begin{cases} s_p & (\text{FastRTX}) \\ MTU_p & (\text{Timer - Based RTX}) \end{cases} \quad (6)$$

其中, \hat{s}_p 是指多路径传输中第 p 条路径的慢启动阈值 s_p 与所有相关多路径慢启动阈值 s_i 和之比,即 $\hat{s}_p = \frac{s_p}{\sum_i s_i}$ ^[6]。用该阈值比因子乘以 $\min\{a, MTU_p\}$,再取顶函数来修正当前的拥塞窗口 c_p 。通过 \hat{s}_p 因子使慢启动拥塞窗口从单路径过渡到了多路径, \hat{s}_p 不再是原始第 p 条路径慢启动阈值,而是综合了所有路径慢启动阈值的一个比值。在公式(5)中使用顶函数 $\lceil \cdot \rceil$ 的目的是为了确保拥塞控制窗口 c_p 至少按一个单位的 MTU_p 增长,从而保证拥塞控制窗口 c_p 的变化满足 AIMD 的规律^[6]。

关于快速重传公式(6)中第 p 条路径的阈值 s_p ,除用 \hat{s}_p 对公式(4)修正以外,还考虑了所有关联路径上拥塞窗口 c_i 的总和。

3.2.4 CMT/RP v2

虽然公式(6)中的 \hat{s}_p 因子体现了多路径传输中的阈值比,但 CMT/RP v1 多路径拥塞控制算法比较适用于传输特性(如带宽、时延、误码率)接近的相似多路径传输通信^[6],当在传输特性相差甚远的多路径上通信时,会带来新的问题^[14]。例如 DSL 通信中,有可能出现在一条高速、低延迟的路径上,分配到一个小的 c_p 与 \hat{s}_p ,反之亦然。解决此问题的算法就是 CMT/RP v2。与 CMT/RP v1 相比, CMT/RP v2 算法引入了一个与 p 路径带宽有关的

增长因子 \hat{i}_p 。该因子从带宽的角度对 plain CMT CC 的慢启动拥塞窗口和快速重传拥塞窗口的算法公式(3)与(4)分别进行了修正,如公式(7)与(8)所示。

$$c_p = c_p + \begin{cases} \lceil \hat{i}_p * \min\{a, MTU_p\} \rceil & (c_p \leq s_p) \\ \lceil \hat{i}_p * MTU_p \rceil & (c_p > s_p \wedge p_p > c_p) \end{cases} \quad (7)$$

其中增长因子 \hat{i}_p 反映了当前路径带宽在多路径总带宽所占的份额或比值: $\hat{i}_p = \frac{\frac{c_p}{RTT_p}}{\sum_i \frac{c_i}{RTT_i}}$ 。 $\frac{c_p}{RTT_p}$ 表示 p 路径上的带宽, $\sum_i \frac{c_i}{RTT_i}$ 为所有相关路径的带宽之和。在公式 (7) 中使用顶函数 $\lceil \cdot \rceil$ 的目的是为了确保拥塞控制窗口 c_p 至少按一个单位的 MTU_p 增长,从而保证拥塞控制窗口 c_p 的变化满足 AIMD 的规律^[6]。

在快速重传算法中,为了减少在 P 路径上的丢包率,引入一个与带宽有关的衰减因子 \hat{d}_p 。 \hat{d}_p 表示为了将总的带宽减半, P_i 路径的带宽所应该减少的量。 \hat{d}_p 的公式如 (8):

$$\hat{d}_p = \max\left\{\frac{1}{2}, \frac{1}{2} * \frac{\sum_i \frac{c_i}{RTT_i}}{\frac{c_p}{RTT_p}}\right\} \quad (8)$$

可以看出 $\hat{d}_p \geq \frac{1}{2}$, 这是为了使慢启动阈值和拥塞窗口至少保持和单路径 SCTP 拥塞控制一致。 \hat{d}_p 的物理意义可以用这个例子来说明: 有两条路径分别为 p_1 (10Mbit/s) 和 p_2 (2Mbit/s), 在 p_1 路径上的丢包引起的衰减因子 $\hat{d}_1 = \frac{1}{2} * \frac{12}{10} = 0.6$; 在 p_2 路径上的丢包引起的衰减因子 $\hat{d}_2 = \frac{1}{2} * \frac{12}{2} = 3.0$ 。这个例子说明带宽越大的路径衰减因子小,反之亦然,体现了资源池的公平性。由此, s_p 和 c_p 按如下规律变化:

$$s_p = \max\{c_p - \lceil \hat{d}_p * c_p \rceil, 1 * MTU_p\}$$

$$c_p = \begin{cases} s_p & (\text{FastRTX}) \\ MTU_p & (\text{Timer - Based RTX}) \end{cases} \quad (9)$$

3.2.5 MPTCP-Like Congestion Control:

与 CMT/RP 一样, MPTCP 拥塞控制机制同样使用资源池 RP 的概念以保证公平性^{[6], [15]}。然而二者的拥塞控制的表现却不相同,当 p 路径丢包时, CMT/RP 试图减半总拥塞窗口和带宽;而 MPTCP 工作起来像是标准的 TCP 或 SCTP 一样,只将 p 路径上的 c_p 减少一半,这会导致不公平的带宽分配。MPTCP-Like 借用控制工程思想,为 subPath 引入了抢占因子 \hat{a} 来达到控制 c_p 的增减的目的,以达到拥塞平衡的目的。抢占因子 \hat{a} 表示 subPath 对总资源的抢占能力。

由于 MPTCP 的拥塞控制原理是基于数据包而不是字节,算法是基于 SCTP 而不是 TCP 的^[6],所以 MPTCP-Like 拥塞控制算法的改进也就应该基于 CMT-SCTP 的,而不是 TCP。在充分利用的拥塞窗口上,根据 p 路径上收到的 a 个新的确认字节, MPTCP-like 拥塞控制的 c_p 按公式(10)来计算。

$$c_p = c_p + \begin{cases} \min\left\{\lceil \frac{c_p * \hat{a} * \min\{a, MTU_p\}}{\sum_i c_i} \rceil, \min\{a, MTU_p\}\right\} & (c_p \leq s_p) \\ \min\left\{\lceil \frac{c_p * \hat{a} * MTU_p}{\sum_i c_i} \rceil, MTU_p\right\} & (c_p > s_p \wedge p_p \geq c_p) \end{cases} \quad (10)$$

其中, \hat{a} 为 subPath 抢占因子^[6], 公式如下:

$$\hat{a} = (\sum_i c_i) * \frac{\max_i\{\frac{c_i}{MTU_i}\}}{(\sum_i \frac{c_i}{MTU_i})^2}$$

拥塞窗口减少机制有一些不一样,如果在路径 P 上发生了重传时, c_p 和 s_p 改变如下:

$$s_p = \max\{c_p - \frac{1}{2} * c_p, MTU_p\}$$

$$c_p = \begin{cases} s_p & (\text{FastRTX}) \\ MTU_p & (\text{Timer - Based RTX}) \end{cases} \quad (11)$$

在这里 c_p 可能减少的是 MTU_p 而不是 $4 * MTU_p$ 。

4. 多路径拥塞控制算法测试分析

前面以资源池概念为基础从理论上对多路径拥塞控制策略机制进行了定量的分析,并有针对性地引入了适用于多路径传输协议的修正因子 (\hat{s} 、 \hat{i} 、 \hat{d}_p 与 \hat{a}) 以解决多路径并发传输过程中的公平性与友好性问题。为了分析上述三个拥塞控制策略的实用性,本文首先利用仿真平台构建了如图 2(a)、(b)所示的相似多路径共享瓶颈与不相似多路径不相交的仿真测试环境,分析了引入这些比例因子后对多路径并发传输的拥塞控制的影响。

4.1 仿真平台构建

测试1仿真平台配置: 基于INET框架, 本文利用 OMNET++ 仿真模型按图2(a)构建其拓扑结构,并配置各参数为: 1)路径数 $N_P=3$; 2)路径延迟 10 ms; 3)测试间隔 20s~30s; 4)路径带宽均设为相同(以体现多路径特性相似性); 5)发送与接收方缓冲大小设为 5,000,000 字节,以满足测试需求; 6)每次仿真运行时间设为 300 秒。

测试2仿真平台配置: 与测试1仿真平台不同的配置有: 1)按图2(b)构建其拓扑结构; 2)路径数 $N_P=2$; 3)有一个 CMT-SCTP 流和一个非 CMT 参考流。二者发送端都处于饱和状态, 1,452 字节的报文以 1500 MTU 无序传输; 4)为了体现多路径特性的非相似性, path#1 带宽为

6.25Mbit/s, 其资源完全用于CMT-SCTP 数据流, 且作为此流的第一路径, path#2 带宽取12.5Mbit/s, 并发传输CMT-SCTP 流和非CMT参考流, 也就是说, CMT-SCTP 流可同时使用这两条路径. 而通过非耦合的拥塞控制可使CMT-SCTP流与非CMT-SCTP流能均等地共享Path#2的带宽; 5) 引入RP路径阻塞功能开关。

4.2 相似多路径共享瓶颈传输拥塞控制

基于测试 1 仿真平台, 分别观察了时间间隔 $t=20s$ 到 $t=30s$ 之间了各路径的拥塞窗口 c_i 、慢启动门限值 s_i 的变化规律^[4], 结果如图 3 中的(a)、(b)、(c)、(d)所示。他们分别表达了 Plain CMT-SCTP、CMT/RP v1、CMT/RP v2 及 MPTCP-Like 拥塞控制策略的仿真结果。

图 3(a)中的蓝色曲线分别代表了 Path#1 的拥塞窗口 c_i 与慢启动门限值 s_i , 绿色与红色曲线对分别代表了 Path#2 与 Path#3。从中可以看出, 每条路径的行为表现类似于图 1 中的独立单路径拥塞控制曲线。但在 23s 处可以观察到一个有趣的现象, 在 1s 的超时时限后 Path#2 与 Path#3 发生了重传, 这就是在瓶颈处发生并发传输的结果。由于 Path#2 与 Path#3 超时重传的发生, 减少了对网络带宽的消耗, 所以 Path#1 获得了明显较大的拥塞窗口,

如图(a)中的蓝色曲线对所示。

图 3(b)、(c)分别为 CMT/RP v1 CC 和 CMT/RP v2 CC 两种拥塞控制策略的曲线表达。通过运用资源池方法, 当各路径的慢启动阈值具有可比性时, 二者的拥塞控制机制的表现很相似, 三条路径之间相互影响, 拥塞窗口 c_i “此消彼长”。一条路径获得了较长的拥塞窗口, 则另两条路径由于引入了 δ_p 或 \hat{i} 因子, 拥塞窗口将变小, 导致了拥塞窗口有个缓慢的增长态势。当表现强势的路径的拥塞窗口下降时, 这种强势效应可能会转移到另一条路径上。与 plain CMT-SCTP 相比较, CMT/RP v1 和 CMT/RP v2 拥塞发生的频率降低了, 拥塞时限变长了。

而MPTCP-like的表现与CMT/RP策略明显不同。其不同点在于MPTCP-like试图保持各路径拥塞窗口的平衡, 使得各路径表现出相似的行为, 好像是“均分”了资源。因为P 路径的拥塞窗口 c_p 很少下降到该路径的 MTU_p 的值, 所以 RP 路径阻塞功能很少被触发。因此, 一个拥塞窗口的下降只会导致轻微地吞吐量的下降, 这说明了MPTCP-like 的拥塞控制机制具有充分利用带宽资源的能力, 如图3(d)所示。

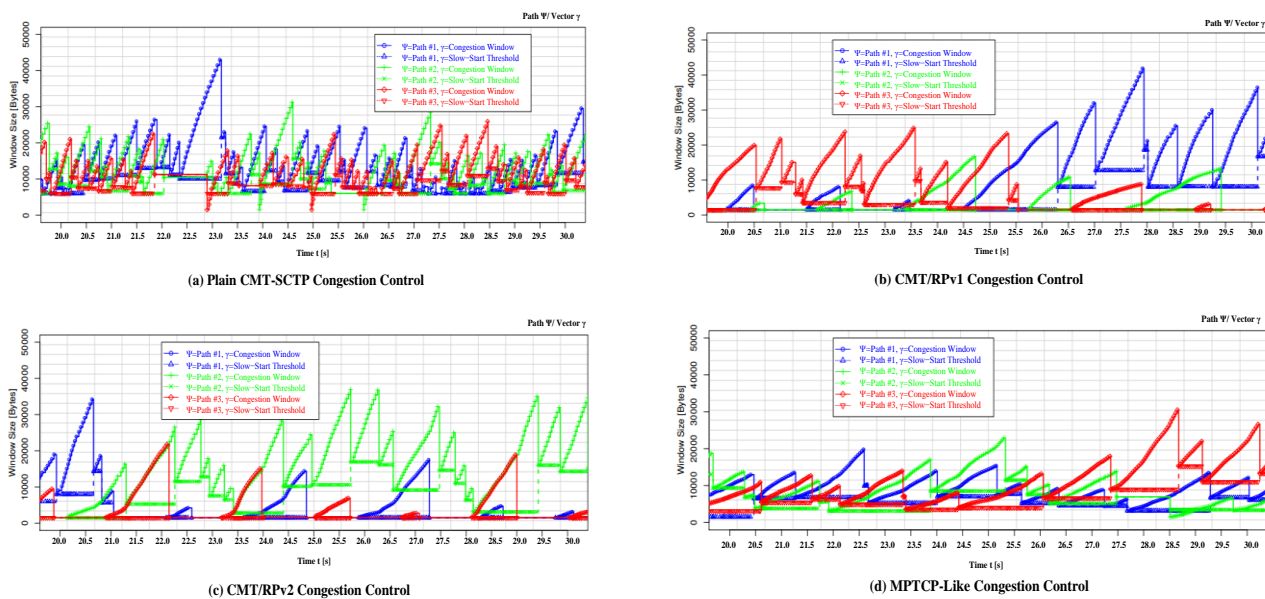


图 3 三条共享瓶颈路径

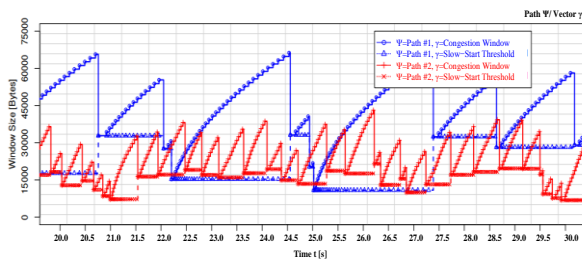
4.3 不相似多路径不相交传输拥塞控制

基于测试2仿真平台, 分别运用Plain CMT-SCTP、CMT/RPv1、CMT/RPv2及MPTCP-Like 策略观察了拥塞控制的行为表现, 结果如图4 (a)、(b)、(c)、(d)。

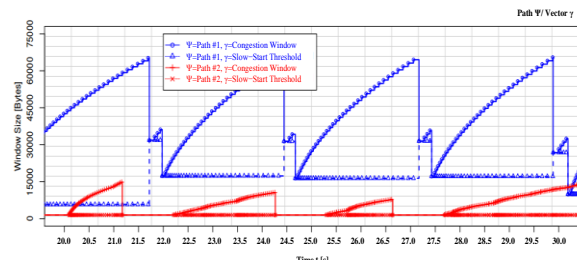
图4 (a) 中蓝色曲线对分别代表Path#1的拥塞窗口 c_i 与慢启动门限值 s_i ; 红色曲线对代表了Path#2的拥塞窗口与慢启动门限值。从中可以看出两条路径的行为表现类似于图1中传统单路径拥塞控制曲线, 并且相互独立。由于两路径传输特性的差异, 红线对和蓝线对所表示的拥塞控制窗口锯齿波的高度也不一样。难以理解的是: 尽管

path#1的带宽小于path#2的,但其拥塞窗口 c_i 与慢启动阈值 s_i 却明显大于path#2的,且拥塞频率也低于path#2。理由是虽然Path#1的带宽(为6.25Mbit/s)仅为path#2的一半,但path#1的带宽仅被CMT-SCTP流独享,而Path#2带宽被CMT-SCTP流和非CMT流共享,发生了带宽的抢占,导致拥塞频繁。因此,Path#1上没有由非CMT流引起的拥塞,也就没有Path#2那么容易产生拥塞。

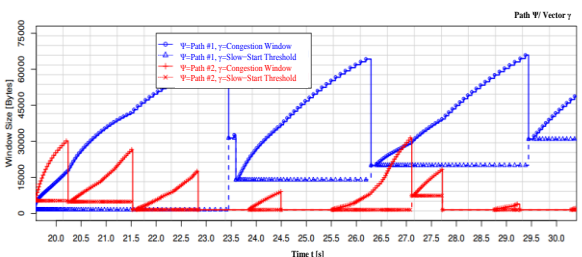
图4(b)反映了CMT/RPv1在处理传输特性不相似路径时的缺陷。因为慢启动阈值比 s_p 的关系,红色线对表示的拥塞控制窗口几乎不能增长,从表现上看CMT/RPv1不适合运用在不相似路径上。Path#2路径的拥塞窗口下降时,RP路径阻塞功能被触发,传输被暂停,1秒之后,开始超时重传。



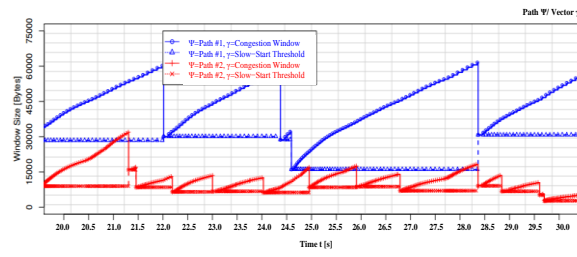
(a) Plain CMT-SCTP Congestion Control



(b) CMT/RPv1 Congestion Control



(c) CMT/RPv2 Congestion Control



(d) MPTCP-Like Congestion Control

图 4 两条不相交不相似路径

5. 结束语

本文首先综述了经典拥塞控制算法,然后基于资源池思想引入了三种用于多路径传输算法:CMT/RPv1 CC、CMT/RPv2 CC与MPTCP-like CC,其核心是使用“慢启动阈值比因子 s_p ”、与多路径传输带宽有关的“增长因子 i_p ”与“支流抢占因子 \hat{a} ”分别并对传统拥塞控制算法进行了扩充与修正。通过仿真测试观察到:三种拥塞控制机制都较好地解决了一般公平性、友好性问题;CMT/RPv1 CC适用于传输特性相似的场合;MPTCP-like CC和CMT/RPv2 CC适用性更宽,都能较好的适用于传输特性相异的不相交路径,但MPTCP-like CC带宽抢占性更强一些,因此也能更高效的利用带宽。用修正的拥塞控制算法实现吞吐量分析,观察到这些算法能有效地平衡多路径的拥塞控制、公平性并提高了网络吞吐量^[12]。

从图4(c)中可以看出,与CMT/RPv1策略相比,path#2在CMT/RPv2的作用下能更好的利用带宽。和Plain CMT-SCTP相比,path#2依然很小,也观察到RP路径阻塞现象但有较大幅度的降低。这就是path#2的绝大部分带宽被非CMT参考流分享,公平性得到了体现。

最后从图4(d)可以观察到MPTCP-like的拥塞控制显示了预期的行为特征。与Plain CMT-SCTP相比,Path#2上的拥塞窗口还是较小,但它并没有下降到两路径的最小MTU值,因此在图中没有观察到RP路径阻塞开关被启用的现象。在Path#2上的拥塞窗口值越大就是在该路径上获得吞吐量越大的原因,并且会导致非CMT参考流吞吐量的减少,带宽抢占减少,友好性得到改善。

下一步将进一步对多路径传输拥塞控制机制的公平性问题给予更精准的定义以更好地调整多路径传输的拥塞控制的性能,并开展实际网络——国际测试床上的部署测试。

参考文献

- [1] Michael Welzl. Network Congestion Control[M]. John Wiley & Sons.2005
- [2] Ford A, Raiciu C, Handley M, et al. Architectural guidelines for multipath TCP development[J]. IETF, Informational RFC, 2011, 6182(1): 2070-1721.
- [3] Iyengar J R, Amer P D, Stewart R. Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths[J]. Networking, IEEE/ACM Transactions on, 2006, 14(5): 951-964.
- [4] Thomas Dreiholz. Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol[D]. Essen/Germany: University of Duisburg-Essen,2012.

- [5] 符发, 周星, 杨雄等. MPTCP 与 CMT-SCTP 多路径传输协议性能分析研究[J]. 计算机工程与应用, 2013, 21:79-82.
- [6] Raiciu C, Wischik D, Handley M. University College London: Practical congestion control for multipath transport protocols[R]. University College London, London/United Kingdom, Tech. Rep, 2009.
- [7] Dreibholz T, Becke M, Adhari H, et al. On the impact of congestion control for concurrent multipath transfer on the transport layer[C]//Telecommunications (ConTEL), Proceedings of the 2011, 11th International Conference on. IEEE, 2011: 397-404.
- [8] Damon Wischik, Mark Handley, Marcelo Bagnulo Braun. Resource Pooling Principle. [M] SIGCOMM Computer Communication Review, 38(5):47-52, October 2008. ISSN 0146-4833.
- [9] Randall R. Stewart. Stream Control Transmission Protocol.[M] Standards Track RFC 4960, IETF, September 2007. ISSN 2070-1721.
- [10] Dreibholz T, Rathgeb E P, Rüngeler I, et al. Stream control transmission protocol: Past, current, and future standardization activities[J]. Communications Magazine, IEEE, 2011, 49(4): 82-88.
- [11] M. Allman, V. Paxson, and E. Blanton. TCP Congestion Control[M]. IETF, Standards Track RFC 5681, Sept. 2009, ISSN 2070-1721.
- [12] Becke M, Dreibholz T, Adhari H, et al. On the fairness of transport protocols in a multi-path environment[C]//Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012: 2666-2672.
- [13] Dreibholz T, Seggelmann R, Tüxen M, et al. Transmission scheduling optimizations for concurrent multipath transfer[C]//Proceedings of the 8th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT). 2010, 8.
- [14] Dreibholz T, Becke M, Pulinthanath J, et al. Applying TCP-friendly congestion control to concurrent multipath transfer[C]//Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. IEEE, 2010: 312-319.
- [15] Raiciu C, Handley M, Wischik D. Coupled multipath-aware congestion control. IETF draft draft-raiciu-mptcp-congestion-01. txt[S], 2010.