

卫星网络中并发 TCP 的复用机制

王维¹, 郝晓强², 段桂华¹, 黄家玮¹, 王建新¹

- (1. 中南大学 信息科学与工程学院, 湖南 长沙, 410083;
2. 航天恒星科技有限公司, 北京, 100190)

摘要: 研究卫星网络中并发 TCP 流的复用机制, 在通过模型分析得到最优复用比的基础上, 基于 epoll 事件驱动机制设计并实现卫星网络的并发 TCP 复用加速网关。研究表明: 该加速网关可以对来自不同客户端的多条 TCP 流按照一定的复用比进行复用, 这样, 每条客户端流可以通过避免 3 次握手并充分利用复用流的窗口加速数据传输; 所设计的加速网关均能将典型门户网站页面的加载时间和首字节到达时间最高缩短 50%, 可满足卫星网络的加速需求。

关键词: 卫星网络; 传输控制协议; 加速网关

中图分类号: TP393

文献标志码: A

文章编号: 1672-7207(2017)03-0712-09

Multiplexing scheme for concurrent TCPs in satellite networks

WANG Wei¹, HAO Xiaoqiang², DUAN Guihua¹, HUANG Jiawei¹, WANG Jianxin¹

- (1. School of Information Science and Engineering, Central South University, Changsha 410083, China;
2. Space Star Technology Co.Ltd., Beijing 100190, China)

Abstract: A multiplexing mechanism for the concurrent TCP in satellite networks was designed, the optimal multiplexing ratio was modeled, and an acceleration gateway based on event-driven epoll mechanism was implemented. The results show that acceleration gateway can multiplex multiple TCP flows of different clients on a TCP flow with a ratio, so that every client's flow can accelerate the data transfer by avoid three-way handshake overhead and take full advantage of the window of multiplexed flow. The accelerate gateway reduces by 50% of the page load time and the arrival time of the first byte for the typical portal web sites, which meets the acceleration requirement of satellite network.

Key words: satellite networks; TCP; acceleration gateway

随着互联网的飞速发展和大规模普及, 各种新型网络应用为用户提供了方便和快速的网络服务。其中, Web 应用是目前使用最广泛的网络应用, 用户可以使用各种网络终端通过浏览器等访问远程的 Web 服务器以获取网络信息和资料。但在大延时和高丢包的卫星网络环境中, 网络传输速度变得十分缓慢。同时,

在 Web 应用等网络服务中, 绝大部分网络流量是由小于 20 kB 的小数据流组成。在卫星网络传输中, 这些小流的传输控制开销相对于有效数据传输较大, 这使得典型网站在卫星网络中的访问时间会从传统广域网中的几秒剧增到几十秒。SmartBear 公司的统计结果表明^[1], 57%的 Web 应用用户在 3 s 内还没加载完就会

收稿日期: 2016-04-10; 修回日期: 2016-06-22

基金项目(Foundation item): 国家自然科学基金资助项目(61572530, 61462007) (Projects(61572530, 61462007) supported by the Natural National Science Foundation of China)

通信作者: 段桂华, 博士, 教授, 从事计算机网络、网络安全研究; E-mail: duangh@csu.edu.cn

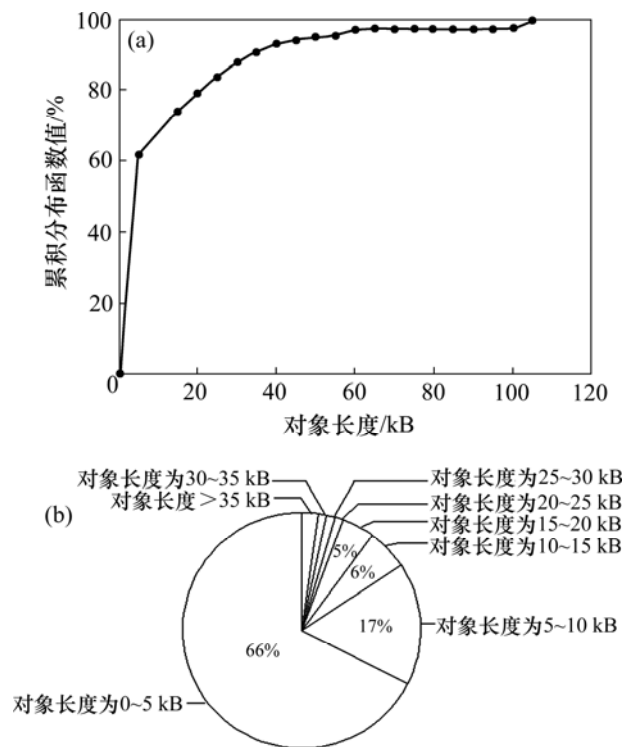
放弃。因此, 如何在卫星网络中提高传输效率、降低响应时间、提高用户的体验就变得十分重要。当前针对卫星网络的加速研究包含网络的多个层面, 其中有使用灰色系统优化 ALOHA 协议中的退避算法^[2]、网络层优化时固定缓冲条件下的最大吞吐率路由算法^[3]、近地卫星网络中的流量预测路由算法^[4]和基于效率最大化的路由算法^[5]。传输层协议优化包括: 通过 RTT(round-trip time, 往返时延)动态指数增窗和动态的降窗算法来修改 TCP 中的拥塞控制策略^[6]; 通过随机早期检测和窗口大小调整进行数据流量的 TCP 传输控制^[7]; 提升延迟敏感服务质量的包调度策略^[8]; 针对近地卫星网络中卫星频繁切换的特点使用 MPTCP 平滑网络切换延迟^[9]。在应用协议的研究中, HTTP PEP^[10]是专门针对 HTTP 协议进行加速, 其主要使用 HTTP 对象缓存、HTTP 预取等; FFD^[11]是针对卫星网络中文件传输而设计的协议, 该协议通过定时请求未确认数据包和重传丢失数据包充分利用链路带宽, 从而可以快速传输文件; GOOGLE 提出了 SPDY 协议^[12], 该协议通过优先加载、多路复用、压缩、推送等技术加速 HTTP 的访问, 在卫星网络中可以缩短页面加载时间^[13-15]。其中, SPDY 协议的多路复用是将当前 HTTP 协议下多条 TCP 连接上的数据复用到 1 条 TCP 连接上发送以避免建立 TCP 连接的延迟。但该协议需要在浏览器和服务端同步部署, 且在高误码率的卫星网络中该协议的传输性能由于 TCP 的拥塞控制而产生抖动。总体而言, 目前基于传输层协议的优化未充分考虑典型应用的数据流量特性, 导致传输控制开销较高。而基于应用协议主要针对某一种特定的应用场景和应用层协议, 无法对其他或场景提供加速服务, 普适性不强。为此, 本文作者针对卫星网络的特点, 基于 TCP 协议的工作流程, 研究卫星网络中并发 TCP 流的复用机制。

1 设计思路

1.1 问题分析

目前, Web 应用是使用最广泛的互联网服务且 Web 应用的数据流通常是由小对象组成。对使用 Google Chrome 浏览器访问“新浪”、“网易”、“搜狐”等目前中国 Top 5 的门户网站主页的对象长度分布进行统计, 对象的长度分布如图 1 所示。从图 1(a)可以看出: 超过 70%的对象小于 20 kB。从图 1(b)可看出: 小于 5 kB 的对象占 66%, 只有 34%的对象大于 5 kB。HTTP Archive 提供了全球范围内 Top 1000

的网站对象长度分布统计^[16]。其结果显示, 55%的 Web 页面中包含的对象数超过 76 个, 所有对象中有 91%是静态对象, 而这些静态对象平均小于 23 kB。



(a) 新浪首页的对象长度累积分布;

(b) 访问量 top5 门户网站的对象长度分布

图 1 对象长度分布

Fig. 1 Distribution of object's size

各大浏览器厂商和 Web 应用提供方使用多种方式来加速 Web 应用的访问。主流的浏览器规定每个域名同时可以建立 6 条 TCP 连接, 通过增大并行度以降低请求与响应的停等式传输带来的影响。而 Web 应用提供方为了进一步加快访问速度, 将不同的对象部署在不同的域名下进一步增大并行度。这些加速机制在低延迟的网络可以加快 Web 访问速度, 但也导致 Web 应用传输的网页对象进一步分散到不同的 TCP 流中, 形成了大量的短数据流。

图 2 所示为测试的对地同步卫星网络拓扑。同步卫星需要对地面保持位置不变, 因此, 保持在离地比较高的运行轨道。其传播延迟被大大拉长, 往返延迟 RTT 达到 500 ms 以上, 而且误码率高于 10^{-5} 。当用户访问首先通过高速、低延迟的地面网络连接到卫星小站时, 小站通过大延迟、高丢包的卫星链路连接到卫星中心站, 最后卫星中心站通过高速、低延迟的地面网络连接到互联网。一般多个小站会共用同一个中心站, 各小站被分配的带宽设定为 100 MB/s。

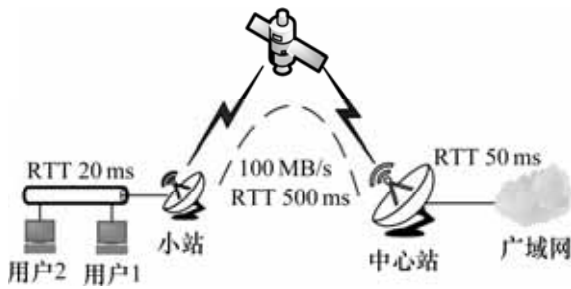


图 2 卫星网络拓扑

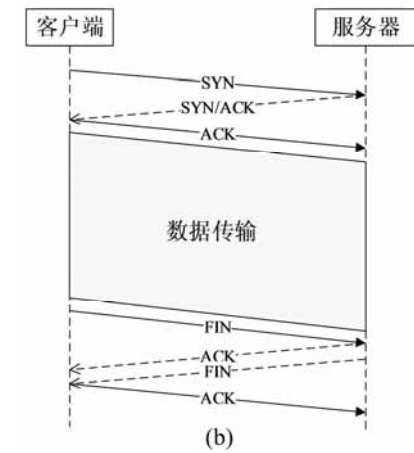
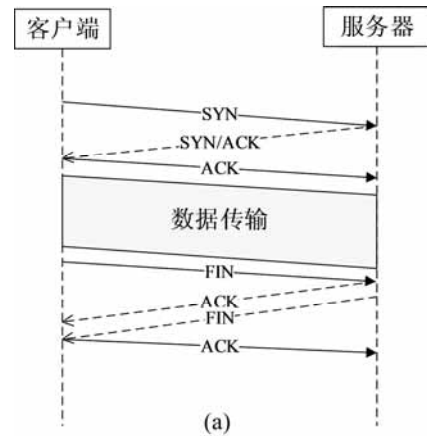
Fig. 2 Topology of satellite network

用户通过卫星网络访问 Web 对象时,大多数小对象会经历 3 次握手的延迟、从请求到响应的延迟和服务端等待客户端确认的延迟。因此,多数对象需要至少 2 个 RTT 才能完成传输,稍大的对象则需要更多 RTT,而这种延迟在 RTT 大于 500 ms 时将非常明显。

1.2 设计思想

从 HTTP Archive 的统计结果可看出 Web 应用的访问场景中主要包含 2 种数据:平均长度小于 20 kB 的对象和平均长度大于 250 kB 的视频流数据。由于视频流数据比较大,浏览器通常采取缓冲策略以通过当前的延迟提升视频播放中的流畅度,所以,视频流对延迟的敏感度低于其他小对象对延迟的敏感度。为了提高 Web 应用中普遍存在的小对象的传输效率,研究卫星网络并发 TCP 流复用机制。其中,传输层有多种变种 TCP 可供选择,包括 TCP Hybla 和 TCP Cubic 等。文献[17]对 TCP Hybla 和 TCP Cubic 在卫星网络中的传输性能进行了对比测试,测试结果显示 TCP Cubic 的性能更优,因此,选用 TCP Cubic 进行传输层的拥塞控制。并发 TCP 流复用机制的设计思想是:将多条 TCP 流传输的小对象复用汇聚到 1 条 TCP 流中,从而降低卫星网络中相对过高的传输控制开销,加速网络数据传输。复用加速前后的工作原理和加速效果见图 3。假设单流的握手和挥手时间为 2 个 RTT,而数据传输延时也是 2 个 RTT。图 3(a)表明单流的传输效率仅为 50%,而图 3(b)显示了复用加速的效果。其中,复用比定义为复用到同一条流的客户端单流数量。这里,设定复用比为 4,即 4 条单流复用到 1 条流,复用后传输效率提升为 80%。

进一步测试加大复用比的传输性能,并统计每条单流的吞吐量。当 RTT 为 500 ms,误码率为 10^{-5} 的模拟卫星网络中,多条客户端流在不同复用比的条件下连续请求多个 20 kB 对象过程中每条客户端流的平均吞吐量示意图见图 4。从图 4 可以看出:当复用比从 1 增长到 5 的过程中,每条客户端流的平均吞吐量



(a) TCP 单流; (b) TCP 复用流

图 3 TCP 传输时序图

Fig. 3 Sequence diagram of TCP transmission

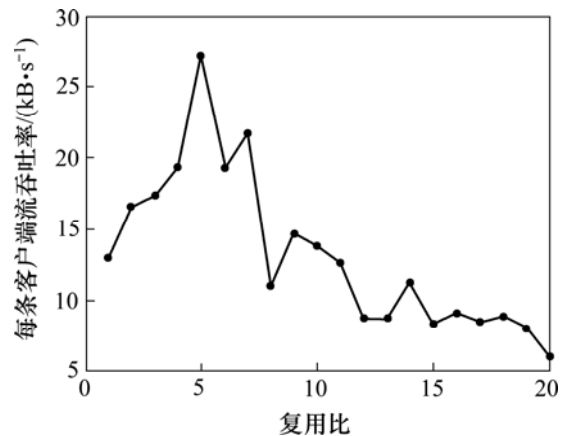


图 4 每条客户端流的吞吐量与复用比的关系
 Fig. 4 Relationship between multiplexing ratio and throughput of each client flow

不断提高,此时,多个客户端流可以利用并增大复用后 TCP 的窗口,从而使每条客户端流获得较高的平均吞吐量;但当复用比从 5 到 20 时,每条客户端流的平均吞吐量呈现下降趋势,其原因在于卫星网络的

误码率高; 随着复用比增加, 每条复用后 TCP 连接上传输的数据量变大, 而卫星网络的高误码率引起的丢包会导致 TCP 减小拥塞窗口以降低发送速率, 这样, 不断增大复用比会导致每条客户端流分得的带宽降低; 当复用比大于 10 时, 每条单流在复用后的 TCP 连接上传输吞吐率甚至低于复用前的吞吐率。

从以上分析可以看出: 卫星网络的大延时和高丢包特性共同决定了复用的效率, 过高或过低的复用比都不能得到最好的加速效果。这就需要进一步分析在特性网络环境中如何得到最优的复用比, 以获取最优传输性能。

2 协议设计

TCP 复用的关键是将多条 TCP 上的数据通过 1 条 TCP 连接进行传输, 提高了传输效率。为此, 设计一种并发 TCP 流的复用机制。该机制处于传输层之上, 对每个 TCP 连接上的数据进行封装并分配会话, 实现 TCP 复用。在复用过程中, 复用比决定了传输的加速性能。因此, 首先通过建模分析得到最优的复用比, 然后研究复用网关的设计结构, 并描述实现细节和关键技术。

2.1 理论分析

最优复用比的计算问题可以转化成如何得到特定网络环境下最佳的并发 TCP 流数量。使用并发 TCP 流拥塞控制模型分析得到最佳的并发 TCP 流数。系统模型中使用的变量见表 1。

表 1 系统模型变量说明

Table 1 Description of variables in system model

变量	含义
m	所有客户端的 TCP 并发数量
n	复用后的并发 TCP 流数量
w_n	n 条 TCP 流的总拥塞窗口长度
d	拥塞窗口降窗因子, 默认为 0.5
i	拥塞窗口增窗因子, 默认为 1
P	丢包率
t	卫星网络往返延迟
s	n 条 TCP 流总传输数据量
R	n 条 TCP 流总平均速率
T_0	TCP 连接握手和挥手时间
r	每条 TCP 流的平均吞吐率
b	链路带宽
M	最大报文段长度

图 5 所示为 n 条 TCP 流的总拥塞窗口 w_n 变化。其中: i 和 d 分别为拥塞窗口的增窗因子和降窗因子, P 为链路丢包率。若没有发生丢包, 则在 1 个往返延时 t 内 n 条流共增长窗口 ni 。当发生误码导致丢包时, 其中 1 条 TCP 流降低其拥塞窗口, 另外 $n-1$ 条流窗口不变, 因此, 总拥塞窗口变为 $(n-d)w_n/n$ 。丢包后, 共花费 $dw_n/(n^2i)$ 个往返延时 t 将总窗口恢复到 w_n 。

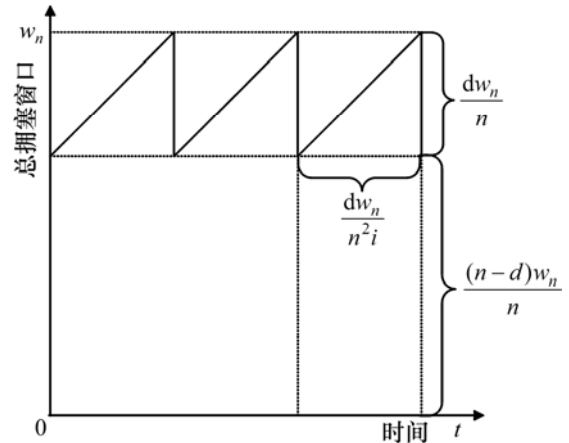


图 5 并发 TCP 流的总拥塞窗口

Fig. 5 Total congestion window of concurrent TCP flows

在各增窗周期内, 各流发包总量为 $1/P$ 。由图 5 可知

$$\frac{dw_n}{n^2i} \times \frac{(n-d)w_n}{n} + \frac{1}{2} \times \frac{dw_n}{n^2i} \times \frac{dw_n}{n} = \frac{1}{P} \quad (1)$$

得到 n 条 TCP 流的总拥塞窗口 w_n 为

$$w_n = n \sqrt{\frac{2ni}{(2n-d)Pd}} \quad (2)$$

根据 w_n 可以得出 n 条 TCP 流的总吞吐率 R 为

$$R = \frac{(1/P)}{(dw_n/n^2i)} \times \frac{M}{t} \quad (3)$$

代入式(1)和(2)得

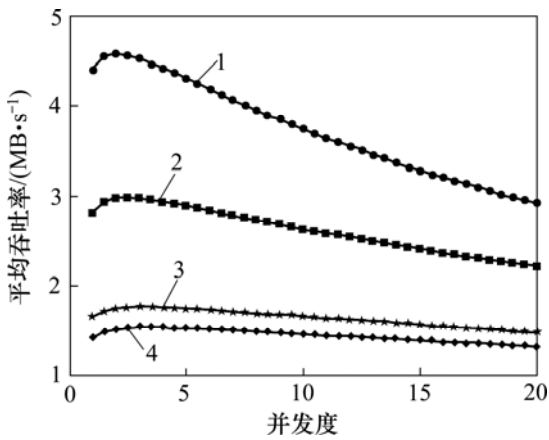
$$R = \sqrt{\frac{ni(2n-d)}{2Pd}} \times \frac{M}{t} \quad (4)$$

在 i 和 d 为默认值时, 随着 n 增加, 总吞吐率 R 也不断增加。这说明在不考虑 TCP 流控制开销时, TCP 流的并发度越高, 传输的总速率也越大。但实际上, 卫星网络中过高的传播延时下 TCP 的控制开销很高, 严重地影响着传输效率。而且随着并发度提高, 复用比不断降低, 这意味着 TCP 连接控制开销在整个传输过程中占据越来越大的比例, 整体上降低了加速效率。

假设 TCP 连接握手和挥手时间为 T_0 , s 为 n 条 TCP 流的总传输量, 由此可以计算出每条流的平均吞吐率 r :

$$r = \frac{s}{nT_0 + \frac{sn}{R}} \quad (5)$$

每条 TCP 流的平均吞吐率 r 见图 6。从图 6 可见: 丢包率越低, 吞吐率越高; 不论是哪种丢包率, 当并发度 n 变化时, TCP 流的吞吐率存在 1 个最高值。这说明存在 1 个最优的复用率, 使得 TCP 流的传输效率最高。下面计算最优的并发 TCP 流数 n_{opt} 。



P: 1—0.000 10; 2—0.000 25; 3—0.000 75; 4—0.001 00。

图 6 每条并发流的平均吞吐率

Fig. 6 Average throughput of each TCP flow

首先取增窗因子 i 和降窗因子 d 为默认值, 并令

$$\alpha = \frac{M}{\sqrt{2Pdt}} \quad (6)$$

得

$$r = \frac{s}{\frac{2sn}{\alpha\sqrt{4n^2-2n}} + nT_0} \quad (7)$$

为了得到最优的并发 TCP 流数 n_{opt} , 将 r 对 n 求导得

$$\frac{dr}{dn} = -\frac{s}{\left(\frac{2sn}{\alpha\sqrt{4n^2-2n}} + nT_0\right)^2} \cdot \left(\frac{2s}{\alpha\sqrt{4n^2-2n}} - \frac{sn(8n-2)}{a\sqrt{(4n^2-2n)^3}} + T_0\right) \quad (8)$$

因为 $\sqrt{4n^2-2n}$ 与 $2n$ 在 n 的取值范围内误差小于 1, 所以, 可以将式(8)简化为

$$\frac{dr}{dn} \approx -\frac{s}{\left(\frac{s}{a} + nT_0\right)^2} \left(\frac{s}{an} - \frac{sn(8n-2)}{a(8n^3-4n^2)} + T_0\right) \quad (9)$$

令 $\frac{dr}{dn} = 0$, 根据 n 的取值范围得到, 得最优的并发 TCP 流数 n_{opt} :

$$n_{opt} = \frac{1}{4T_0a} \left(T_0a + \sqrt{T_0^2a^2 + 4T_0as}\right) \quad (10)$$

得到最优的并发 TCP 流数 n_{opt} 后, 可得到复用比 α 为

$$\alpha = \frac{m}{n_{opt}} \quad (11)$$

2.2 加速网关架构

TCP 复用加速网关分为小站网关和中心站网关 2 部分。加速网关工作于传输层和应用层之间, 主要由 3 部分构成。

- 1) 事件驱动。采用 `epoll` 的事件驱动机制, 负责处理所有的 TCP 连接的事件。
- 2) 连接池。维护小站网关与中心站网关之间正在使用的 TCP 连接。
- 3) 会话层。负责维护用户与小站网关和中心站网关与广域网之间的会话。

图 7 所示为 TCP 复用的主要流程。首先, 小站网关启动以后会阻塞在监听连接处等待客户端发起 TCP 连接。客户端发起 TCP 连接后, 此时需要判断连接池中是否有到中心站的 TCP 连接。若有且该复用流的当前复用数小于计算得到的最优复用比, 则会将该客户

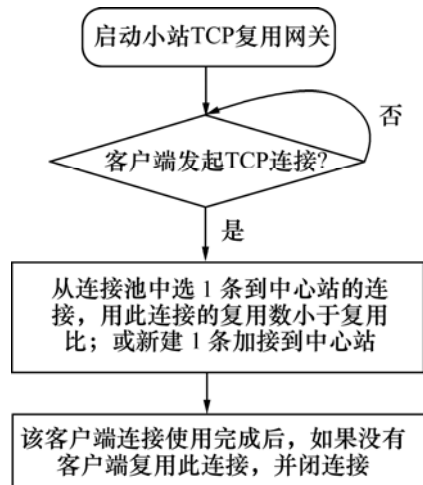


图 7 TCP 复用流程

Fig. 7 Process of TCP multiplexing

端的数据在此连接上传输; 否则, 创建 1 条新的连接到中心站, 在这条连接上传输该客户端数据。每次客户端连接的数据传输完成后, 网关查看是否有其他客户端连接在使用, 若没有, 则关闭此链接。

TCP 复用加速网关基于 Linux2.6 内核, 用 libuv 库和 C 语言实现。其中 libuv 库是 1 个异步 IO 库, 可以简化大并发网络程序的开发, 使用的关键技术包括 TCP 流映射、会话同步、连接池维护。

2.3 TCP 流映射

系统中所有的 TCP 建流请求都是由客户端发起。当小站网关接收到 1 个建流请求时, 会为该流创建 1 个会话 ID, 分配或创建小站网关和中心站网关之间的 1 条 TCP 连接, 并将这种对应关系存储在小站网关的映射表中。这样, 当客户端 TCP 连接上有数据时, 可以根据文件描述符对应的结构体获得会话 ID 和复用的 TCP 流的信息。当复用流上有数据到达时, 可以根据会话 ID 获取到客户端 TCP 连接的信息, 完成客户端映射。此后, 小站网关发送数据到中心站网关后, 中心站网关会根据小站网关发送的会话 ID 也建立 1 个中心站的映射关系并存储在中心站的映射表中, 当复用流上有数据时, 可以根据会话 ID 获取到对应服务器端连接信息。当服务器端连接上有数据时, 可以根据文件描述符对应的结构体得到会话 ID 和复用的 TCP 流的信息。

2.4 会话层

会话层使用了自定义的协议, 该协议的数据包格式如图 8 所示。图 8(a)中数据包分为头部和数据区, 其中头部包含 4 字节会话 ID、1 字节控制命令和 2 字节数据长度, 数据区在一般情况下只包含有效负载。客户端向小站网关发起请求后, 小站为该 TCP 连接创建会话, 会根据客户端发送给它的目标地址和端口等信息初始化会话。会话的数据格式如图 8(b)所示。中心站接收到该数据包以后会解析出目标域名和端口,

并与该地址建立 TCP 连接, 然后将数据发送给该地址。

中心站接收到远端服务器返回的响应后, 构造普通的包后发送到小站。小站网关接收到数据包后根据会话 ID 找出对应的客户端连接, 然后将数据发送给客户端。

当连接异常关闭后, 网关要实施会话同步。客户端的连接和服务器的连接都有可能因为某种原因关闭。当连接关闭时, 需要将小站和中心站网关的会话都删除。若客户端连接首先异常关闭, 则小站网关会发送 1 个不包含数据区的会话包到中心站网关, 表示关闭该 ID 的会话, 同时将自身的相关资源释放。中心站网关收到该包后, 释放会话对应的资源。若服务器的连接异常关闭, 则中心站网关通告小站网关释放对应会话的资源, 同时将自身的相关资源释放。

2.5 连接池

小站网关和中心站网关之间保持正在被客户端连接使用的 TCP 连接, 这些连接由小站网关主动发起并维护, 多条连接之间的负载均衡也是由小站网关完成。当客户端发起 1 条 TCP 连接时, 小站网关使用最少客户端使用的负载均衡算法将该 TCP 连接与连接池中的某条 TCP 连接关联。此后, 该客户端连接上的所有数据都由选定的 TCP 连接处理。这样, 只要有客户端正在使用该连接, 该连接就不会被主动断开。当新的客户端连接到来时就有机会使用这条连接, 但若最后 1 个使用该连接的客户端连接关闭后, 连接也随之关闭。

3 实验分析

实验分为本地测试、广域网测试和对比测试。本地测试和广域网测试分别测试可控环境下和真实环境下的复用网关加速性能, 对比测试用于比较本文的并发 TCP 复用机制和 SPDY 协议的传输性能。其中, 采用前面的实验拓扑结构进行测试。小站网关和中心站网关均部署在普通 PC 上, 配置如下: CPU 为 Intel(R)CoreTMi3-3220 CPU @ 3.30 GHz, 内存为 4 GB, 网卡为千兆网卡。卫星网络模拟器使用 WANem 3.0, 卫星网络模拟器设置带宽为 100 MB/s。客户端使用 chrome 浏览器访问测试网站 10 次, 然后统计平均测试结果。

实验中, 测试网页加载时间和首字节到达时间。



(a) 普通数据包格式; (b) 初始化数据包格式

图 8 数据包格式

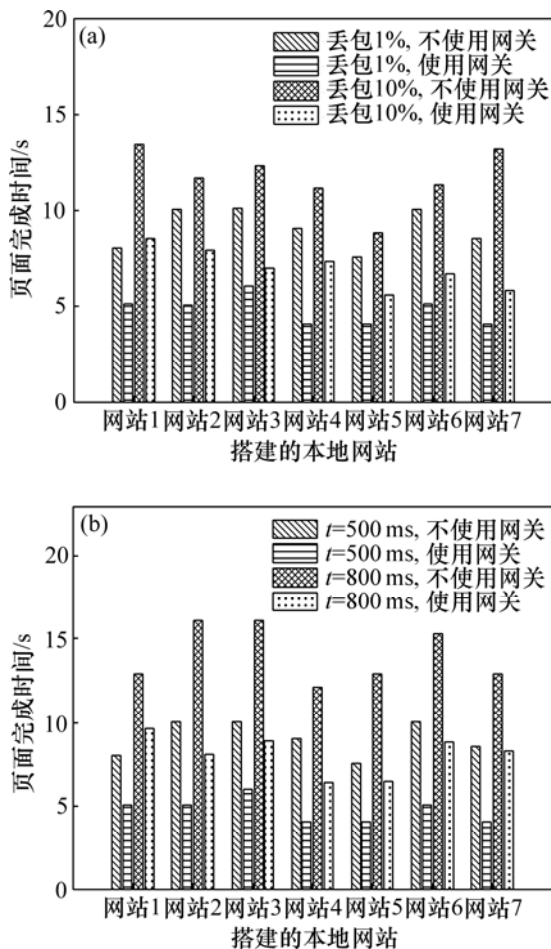
Fig. 8 Packet formats

其中, 网页加载时间是页面所有对象元素都到达客户端的最终时间, 是决定网站用户体验的关键指标。而首字节达到时间为响应页面的第 1 个字节到达客户端的最快时间, 包括 DNS, socket 连接和请求响应时间, 是反映服务端响应速度的重要指标。

3.1 本地测试

本地测试是在局域网中搭建网站, 然后客户端通过 WANem 访问这些网站。客户端采用 chrome 浏览器测试各个网站, 然后从 chrome 浏览器访问各个网站所产生的日志文件中统计网页的加载时间。通过实验得出每个网站在不同丢包和往返延时条件下, 使用加速网关和不使用加速网关的网页加载时间对比如图 9 所示。其中, 图 9(a)中, 往返延时 t 为 500 ms; 图 9(b)中, 丢包率为 1%。

从图 9(a)可见: 使用加速网关最低可以缩短 37% 的页面加载时间, 最高可以缩短 55% 的页面加载时间;



(a) 不同丢包率条件下的网页加载时间; (b) 不同往返延时条件下的网页加载时间

图 9 局域网中网页加载时间测试结果

Fig. 9 Test results of web page load time in LAN

增大丢包率后, 较大的丢包率对使用网关和不使用网关的网页访问都产生了负面影响, 导致页面加载时间波动变大, 其中使用加速网关最低可以缩短 36% 的页面加载时间, 最高可以缩短 56% 的页面加载时间。从图 9(b)可见: 增大往返延时 t 后依然保持较高的加速比, 最低缩短 25% 的页面加载时间, 最高缩短 49% 的页面加载时间。

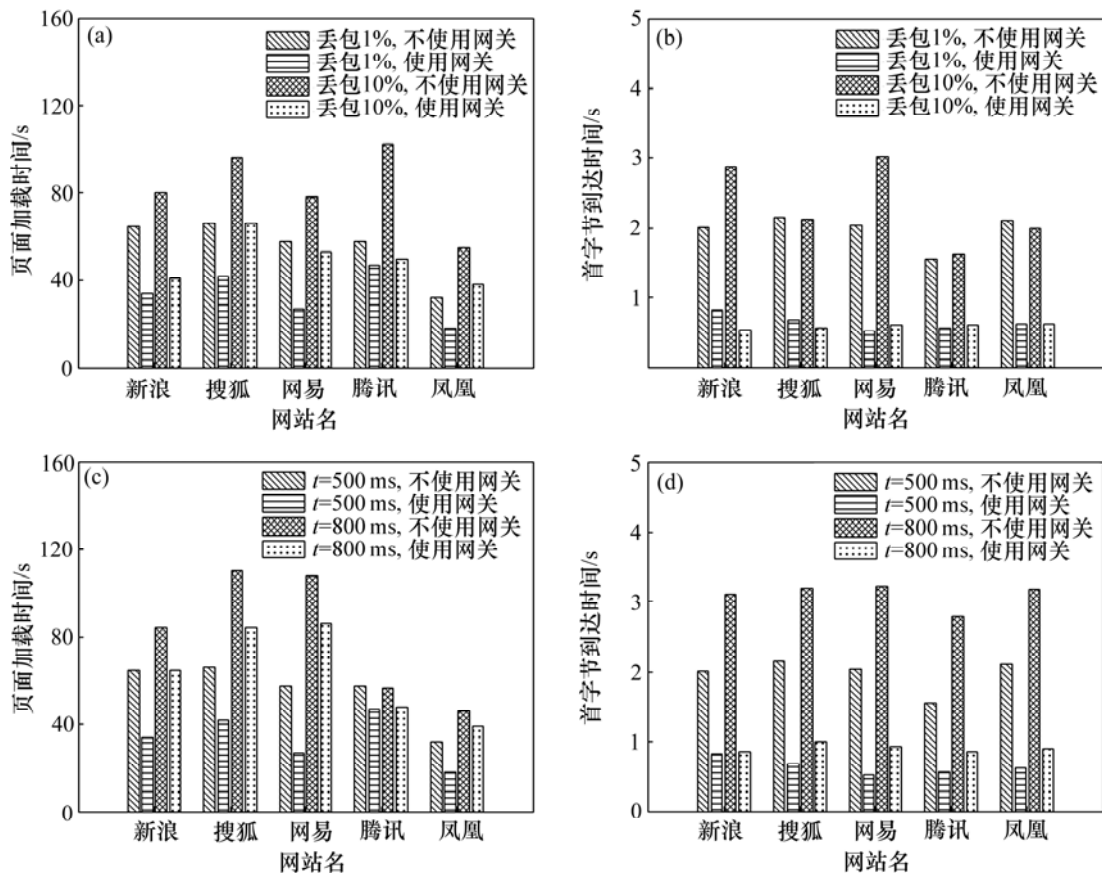
3.2 广域网测试

在广域网测试中, 客户端通过 WANem 访问广域网中访问量 top5 的门户网站。通过多次实验得出在不同丢包率和往返延时条件下, 不同门户网站主页的页面加载时间和首字节到达时间, 统计结果如图 10 所示。从图 10(a)可见: 当往返延时 t 为 500 ms, 丢包分别为 1% 和 10% 时, 使用网关比不使用网关访问 5 个门户网站首页的页面加载时间明显缩短。从图 10(b)可见: 随着丢包率增大, 首字节到达时间更短。这主要是因为不使用网关的访问中域名的解析也需要通过卫星网络, 因此, 严重加大了门户网站时的首字节到达时间; 而采用加速网关后, 能有效地降低 DNS 解析和流传输的时间。保持 1% 丢包不变, 增大往返延时 t 的测试结果如图 10(c) 和图 10(d) 所示, 可见页面加载时间总体保持加速, 而首字节到达时间加速效果更明显。

3.3 对比测试

由于 SPDY 是针对当前 HTTP 协议进行优化, 且使用了 TCP 复用技术, 因此, 对比测试中选用的 SPDY 协议作为比较对象。在对比测试中, 客户端分别通过并发 TCP 复用机制和 SPDY 协议访问局域网中部署的页面, 其中不同页面中包含不同数量的长度为 20 kB 的对象。通过多次实验得出通过并发 TCP 复用机制和 SPDY 协议访问不同页面时的页面加载时间, 统计结果如图 11 所示。

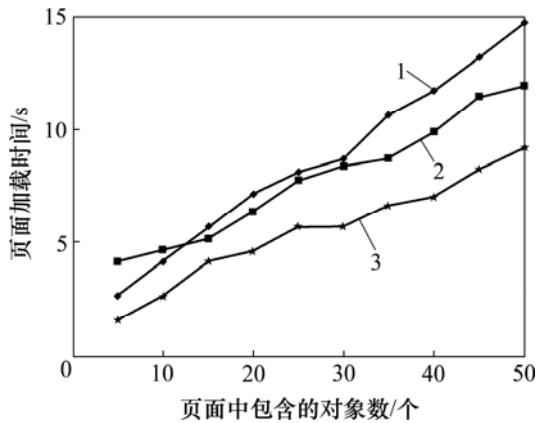
图 11 中 Cubic 是不使用任何加速技术的页面访问状态。从图 11 可以看出: 在页面中包含不同数量对象的条件下, 使用并发 TCP 复用可以缩短页面的加载时间, 且比使用 SPDY 的加速效率更高。这主要是因为 SPDY 采用的是将所有对象复用到 1 条流的策略, 其传输性能受 TCP 拥塞控制的影响, 而且 SPDY 自身复杂的控制流程导致在页面对象数小于 15 时的页面加载时间大于采用 Cubic 访问时的页面加载时间。并发 TCP 复用通过合理的复用比降低了 TCP 拥塞控制的影响, 因此, 可以获得更好的加速性能。



(a) 不同丢包下的主页页面加载时间; (b) 不同丢包下的首字节到达时间;
 (c) 不同往返延时的主页页面加载时间; (d) 不同往返延时的首字节到达时间

图 10 广域网中不同门户网站网页加载和首字节到达时间测试结果

Fig. 10 Test results for web page load time and first byte arrival time of different portals in WAN



1—Cubic; 2—SPDY; 3—并发 TCP 复用。

图 11 网页加载时间对比测试结果

Fig. 11 Comparison test results of web page load time

4 结论

1) 以传输 Web 应用对象为例分析了卫星网络中

基于 TCP 协议的传输效率低的问题, 根据典型门户网站的数据对象的长度分布和 Web 应用数据传输的流量特性, 建模分析得到保证传输效率最高的最优复用比的基础上设计了并发 TCP 流的复用机制, 并实现了并发 TCP 流的复用加速网关。网关均能将典型门户网页页面的加载时间和首字节到达时间最高缩短 50%。

2) 下一步将进一步研究卫星网络中更为高效的传输方法, 以及基于之上的压缩、预取等其他加速方法, 并将测试大规模真实系统。

参考文献:

[1] Mashable. Slow website cost retailers billion[EB/OL]. [2013-04-22]. <http://mashable.com/2012/11/22/slow-website/>.
 [2] QI Xiaohang, ZHANG Qi, TIAN Qinghua, et al. Design of adaptive backoff algorithm for satellite network using grey system[J]. The Journal of China Universities of Posts and Telecommunications, 2015, 22(1): 24-56.

- [3] LIN Bing, LI Hongyan, LONG Yan. A buffer-limited maximum throughput routing algorithm for satellite network[C]// 22nd International Conference on Telecommunications (ICT2015). Sydney, Australia, 2015: 378–383.
- [4] NA Zhenyu, GAO Zihe, CUI Yang, et al. Agent-based distributed routing algorithm with traffic prediction for LEO satellite network[J]. International Journal of Future Generation Communication & Networking, 2013, 6(3): 67–84.
- [5] LU Yong, ZANG Jun, ZHANG Tao. UMR: A utility-maximizing routing algorithm for delay-sensitive service in LEO satellite networks[J]. Chinese Journal of Aeronautics, 2015, 28(2): 499–507.
- [6] SHIN M, PARK M, OH D, et al. RTT fair congestion control algorithm for satellite interactive networks[C]// Computing, Communications and Applications Conference (ComComAp2012). Hong Kong, China, 2012: 339–344.
- [7] TORRES R, BORDER J, STEVENS P, et al. Method and system for controlling TCP traffic with random early detection and window size adjustments: US Patent 8,705,357[P]. 2014–04–22.
- [8] LU Yong, ZHANG Jun, ZHANG Tao. UPQ: An effective packet scheduling algorithm for delay-sensitive service in N GEO satellite network[C]// 9th International Conference on Communications and Networking in China (CHINACOM2014). Maoming, China, 2014: 517–522.
- [9] DU Pengyuan, LI Xiao, LU You, et al. Multipath TCP over LEO satellite networks[C]// 11th International Wireless Communications and Mobile Computing Conference (IWCMC 2015). Dubrovnik, Croatia, 2015: 1–6.
- [10] DAVERN P, NASHID N, SREENAN C J, et al. Httppe: a http performance enhancing proxy for satellite systems[J]. International Journal of Next-Generation Computing, 2011, 2(3): 242–256.
- [11] LIU Chang, WU Chunqing, YU Wanrong, et al. FFDP: a full-load file delivery protocol in satellite network communication[C]// Wireless Algorithms, Systems, and Applications. Berlin: Springer International Publishing, 2015: 377–385.
- [12] GOOGLECOM. SPDY: an experimental protocol for a faster web[R]. [2016–03–30]. <http://dev.chromium.org/spdy/spdy-whitepaper>.
- [13] ROSETI C, SALAM A A, LUGLIO M, et al. SPDY over satellite: performance optimization through an end-to-end technology[C]// 38th International Conference on Telecommunications and Signal Processing (TSP2015). Prague, Czech Republic, 2015: 1–6.
- [14] CARDACI A, CAVIGLIONE L, GOTTA A, et al. Performance evaluation of SPDY over high latency satellite channels[M]// Personal Satellite Services. Berlin: Springer International Publishing, 2013: 123–134.
- [15] SALAM A A, LUGLIO M, ROSETI C, et al. SPDY multiplexing approach on long-latency links[C]// IEEE Wireless Communications and Networking Conference (WCNC). Istanbul, Turkey, 2014: 3450–3455.
- [16] Http Archive. Interesting stats[EB/OL]. [2015–10–01]. <http://httparchive.org/interesting.php>.
- [17] TRIVEDI S, JAISWAL S, KUMAR R, et al. Comparative performance evaluation of TCP Hybla and TCP Cubic for satellite communication under low error conditions[C]// IEEE 4th International Conference on Internet Multimedia Services Architecture and Application (IMSAA2010). Bangalore, India, 2010: 1–5.

(编辑 陈灿华)