0254-6124/2019/39(1)-111–07 *Chin. J. Space Sci.* 空间科学学报

LIU Pan, CAI Yanxia, LU Guorui. Space environment data transfer system based on BBR congestion control algorithm (in Chinese). *Chin. J. Space Sci.*, 2019, **39**(1): 111-117. DOI:10.11728/cjss2019.01.111

基于 BBR 拥塞控制算法的空间环境数据传输系统*

刘 盼[1,2] 蔡燕霞[1,2] 鲁国瑞[1]

1(中国科学院国家空间科学中心 北京 100190)

2(中国科学院大学 北京 100049)

摘 要 空间环境监测网需要通过广域网将海量空间环境数据从地面监测台站传输到数据中心. 本文研究了 TCP 拥塞控制机制, 采用 BBR 拥塞控制算法解决数据传输存在的问题. 针对空间环境数据传输需求, 设计了空间环境数据传输的工作流模型, 建立了空间环境数据传输系统. 实验证明该系统适于进行广域网链路上的空间环境数据传输.

关键词 空间环境, 数据传输, TCP, 拥塞控制, BBR

中图分类号 V 443, TP 393.2

Space Environment Data Transfer System Based on BBR Congestion Control Algorithm

LIU Pan[1,2] CAI Yanxia[1,2] LU Guorui[1]

1(*National Space Science Center, Chinese Academy of Sciences, Beijing* 100190)

2(*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract Space environment data transfer system needs to transfer massive space environment data from the ground monitoring station to the data center through Wide-Area Network (WAN). High loss rate and large delay of WAN influence the normal operation of the space environment data transfer system. In this paper, TCP congestion control mechanism is studied and the BBR congestion control algorithm is used to solve the problems of space environment data transmission. Aiming at the requirements of space environment data transmission, a workflow model of space environment data transfer is designed using Petri-net, and a space environment data transfer system based on BBR congestion control algorithm is established. Finally, it is proved by the experiments that the system is applicable to space environment data transfer on WAN.

Key words Space environment, Data transfer, TCP, Congestion control, BBR

---

* 国家自然科学基金面上项目 (41574181) 和国家重点研究发展计划项目 (2016YFB0501503) 共同资助

2017-12-14 收到原稿, 2018-05-11 收到修定稿

E-mail: pliu@whu.edu.cn

0 引言

空间环境地基监测是进行空间环境预报和研究的重要手段 [1]. 中国科学院空间环境监测网由分布在全国范围内的地基监测台站和数据中心组成, 能够对太阳活动、宇宙线、地磁、电离层和中高层大气等空间环境要素进行监测, 监测台站将监测数据通过广域网实时传送到数据中心. 由于不同监测台站接入的网络条件不同, 数据传输的速率和稳定性受网络条件制约, 通过公网传输数据会出现传输中断、数据积压和接收到的数据不完整等问题. 随着监测设备分辨率的提高, 数据文件越来越大, 单个文件达到上百 MByte 甚至更大. 此外, 在空间天气事件发生时, 需要增大对空间环境监测的时间频率, 数据文件数量随之增多. 现有数据传输系统传输效率不高, 无法满足海量空间环境数据的传输需求.

为解决空间环境数据传输面临的问题, 本文研究了 TCP 拥塞控制机制, 将 Cardwell 等 [2]2016 年提出的基于拥塞的拥塞控制 (BBR) 算法应用于空间环境数据传输中, 在此基础上建立空间环境数据传输工作流模型, 以提高数据传输的速率和稳定性.

1 TCP 拥塞控制机制

1.1 TCP 拥塞控制算法

中国科学院空间环境监测网的空间环境数据传输使用 TCP 协议, 通过拥塞控制机制保证整个网络的服务质量 (Quality of Service, QoS), 防止主机向网络中发送过多数据而造成网络性能下降 [3].

目前广泛使用的拥塞控制算法主要包括 TCP Tahoe[4], TCP Reno[5], TCP NewReno[6], SACK[7] 及 CUBIC[8] 等. 这几种算法都是基于丢包反馈的拥塞控制算法, 即以丢包的发生作为判断网络拥塞的标志. 当发生丢包时, TCP 会减小拥塞窗口, 降低发送速率, 从而减轻网络拥塞程度. 然而随着网络硬件和网络技术的发展, 丢包反馈很多时候会对网络拥塞进行误判, 盲目减小拥塞窗口, 降低发送速率, 导致 TCP 传输效率低, 吞吐量小. 因此, 对 TCP 拥塞控制算法进行改进逐渐成为研究热点.

Brakmo 等 [9] 提出的 TCP Vegas 算法通过持续测量往返时延 (Round-Trip Time, RTT) 判断网络是否出现拥塞来进行拥塞控制. 但是 TCP Vegas 算法带宽竞争力极弱, 不适用于公网中的数据传输 [10]. Padmanabhan 等 [11,12] 提出的 TCP Fast Start 算法对 TCP 慢启动过程进行改进, 根据网络情况决定初始慢启动阈值, 可以提高 HTTP 等短连接数据传输的效率, 但对于需要长连接的空间环境数据持续传输作用不大. Floyd[13] 提出的 ECN 算法通过数据包中的标志位判断拥塞, 而不是完全依据丢包判断网络拥塞. 该算法有助于将网络中数据包损坏造成的丢包与拥塞造成的丢包区分开, 但是需要专用路由设备的支持, 而且在数据包中添加标志位会增加数据量 [14]. Cardwell 等提出的 BBR 算法能够在广域网通过测量带宽和延迟, 较准确地对网络拥塞进行判断. 因此, 这里选择使用 BBR 拥塞控制算法对 TCP 进行加速.

1.2 BBR 拥塞控制算法

Cardwell 等认为, 可以将端到端的网络链路看作一条管道, 该管道管径即为瓶颈带宽 ω_{BtlBw}, 数据包从源端发出到收到目的端对数据包的确认所经历的时间 τ_{RTprop} (不包括节点排队和延迟确认花费的时间) 相当于管道的长度, 则 BDP (带宽延迟积) 为管道容积. 为实现 TCP 在达到最大吞吐量的同时延迟最小, 要求数据的发包速率等于瓶颈带宽, 并保证链路上的数据量等于 BDP. 只要得到任意时刻的 ω_{BtlBw} 和 τ_{RTprop}, 就能在不造成拥塞的前提下以最大速率发送数据. 事实上, ω_{BtlBw} 和 τ_{RTprop} 是无法准确获得的, 只能进行粗略估计.

BBR 算法以一段时间内计算的最大传输速率作为 ω_{BtlBw} 的估计, 以测得的最小延迟作为 τ_{RTprop} 的估计. 计算方法如下.

$$\hat{\omega}_{\mathrm{BtlBw}} = \max(v_t), \tag{1}$$

$$\hat{\tau}_{\mathrm{RTprop}} = \min(RTT_t). \tag{2}$$

其中, v_t 为任意时刻 t 的传输速率. BBR 算法通过持续计算带宽和延迟来感知网络拥塞变化情况, 进而采取相应策略来调整数据发送行为, 其工作原理如图 1 所示.

1.3 BBR 算法与 CUBIC 算法的比较分析

目前最新 Linux 4.15 内核使用的是 CUBIC 算法. 该算法由 Ha 等 [8] 于 2008 年提出, 其在基于丢包反馈类拥塞控制算法里性能最优. 为了验证 BBR 算法的优越性, 在 10 Mbps 带宽和 100 ms 延迟的模

拟网络链路进行了一系列实验, 使用 WinSCP 进行数据传输, 改变链路的丢包率, 统计各丢包率下数据传输的平均吞吐量, 实验结果如图 2 所示.

图 2 横坐标为链路丢包率, 纵坐标为平均吞吐量. 从图 2 可以看出, 在 10 Mbps 带宽和 100 ms 延迟的网络环境下, 丢包率大于 0.05% 时, CUBIC 算法的平均吞吐量随丢包率的增加逐渐下降, 丢包率大于 2% 时平均吞吐量接近为 0, 此时数据传输几乎停止. 而 BBR 算法在同样的丢包率下明显优于 CUBIC 算法, 丢包率小于 5% 时, 平均吞吐量受丢包率影响较小, 能够充分利用带宽, 丢包率大于 5% 时, 平均吞吐量随丢包率的增加迅速下降, 丢包率大于 20% 时, 平均吞吐量接近 0. 这是因为 CUBIC 算法将丢包作为网络拥塞的标志, 每次检测到丢包后立即成倍地减小拥塞窗口, 在丢包率较高时这样频繁缩小拥塞窗口将导致链路利用率降低, 吞吐量小. 而 BBR 算法的发送速率只与 BBR 探测到的可用带宽有关, 如果网络并没有发生拥塞, 丢包率也在一定范围内, 那么估计带宽就能接近实际带宽. 事实上, 广域网链路的丢包率一般不会超过 5%. 因此, 使用 BBR 算法作为 TCP 拥塞控制算法进行空间环境数据传输更稳定高效.

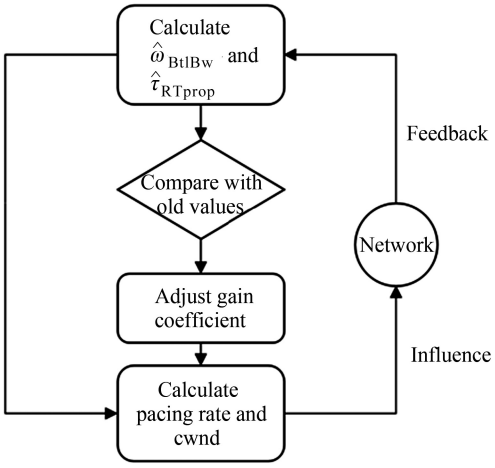


图 1　BBR 算法工作原理

Fig. 1　Working principle of BBR algorithm

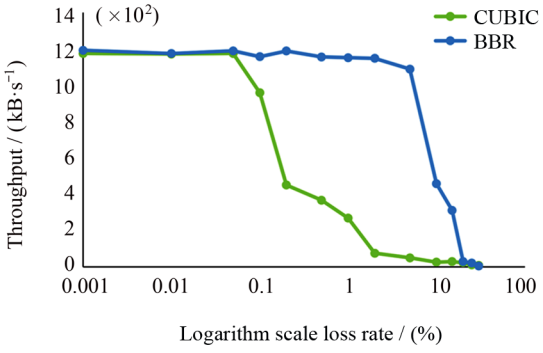


图 2　BBR 算法和 CUBIC 算法在不同丢包率下的数据传输实验结果

Fig. 2　Data transfer experimental results of BBR and CUBIC algorithms at different loss rates

2　空间环境数据传输工作流模型

2.1　空间环境数据传输需求

中国科学院空间环境监测网传输的数据包括太阳活动、宇宙线、地磁、电离层、中高层大气等多种类型. 每种数据由多个台站的不同观测设备提供, 不同设备产生数据文件的格式和频率也不同. 数据文件产生的频率范围为 1 分钟至 1 天, 单个数据文件大小相差很大, 小则几 KB, 大到上百 MB. 海量多源异构的空间环境监测数据对空间环境数据传输提出了以下需求.

(1) 空间环境数据传输系统稳定性. 空间环境预报业务需要稳定、实时的空间环境监测数据, 因此系统需要长期 7×24 h 稳定运行.

(2) 空间环境数据发送速度. 能在短时间内处理大量小文件; 对于大文件的传输, 要保证足够的发送速度以避免数据发送不及时导致数据堆积.

(3) 空间环境数据完整性. 对于数据发送端, 数据发送中断后, 能够自动对数据进行补发; 对于数据接收端, 需要保证数据接收的完整性.

(4) 数据并发接收. 数据接收端能够同时接收多台站发送的数据.

空间环境数据传输系统是一个复杂的系统, 为使系统具有高可靠性, 有必要对空间环境数据传输的工作流进行建模.

2.2　空间环境数据传输工作流建模

工作流的建模方法主要有三种: 一是基于图形符号, 例如 UML, EPC; 二是基于数学, 例如 CTR, ECA; 三是基于图形符号和数学, 例如 Petri 网 [15]. Petri 网既有严格的数学表述方式, 也有直观的图形表达方式, 能够较准确地描述复杂系统的分布、并行、不确定和随机特征 [16]. 这里基于 Petri 网构建空间

环境数据传输工作流模型, 进行系统的设计与实现.

2.2.1 Petri 定义

满足以下条件的三元组 $N = (P, T, F)$ 是一个 Petri 网.

(1) P 为库所的有限集合, 库所用来描述系统状态.

(2) T 为变迁的有限集合, 变迁用来描述改变系统状态的事件.

(3) $P \cap T = \phi$, $P \cup T \neq \emptyset$.

(4) $F = F^+ \cup F^-$, $F \subseteq (P \times T) \cup (T \times P)$, F 表示 P 与 T 之间流关系的有向弧集, 描述执行业务过程所经过的活动时序排列. 其中 "×" 为笛卡儿积, $F^+ = \{f | f \subseteq (P \times T)\}$ 为从库所指向变迁的有向弧集合, $F^- = \{f | f \subseteq (T \times P)\}$ 为从变迁指向库所的有向弧集合 [17].

2.2.2 空间环境数据传输工作流模型

根据空间环境数据传输业务流程和 Petri 网的建模规则, 设计空间环境数据传输工作流模型, 如图 3 所示. 图 3 中各个库所和变迁所代表的具体含义列于表 1.

3 系统架构

基于空间环境数据传输工作流模型设计了空间环境数据传输系统架构, 如图 4 所示, 其包括客户层、接口层、服务层和存储层.

3.1 客户层

客户层是空间环境数据传输发送端, 分布在各空间环境监测台站, 由发送模块和 TCP 加速模块组成. 发送模块负责数据文件监视以及数据发送任务的调度和执行. 当数据文件产生时, 新的发送任务会被加入到任务列表, 由控制器依次调度, 数据的发送由发送器执行, TCP 加速模块负责控制数据发送的速度.

TCP 加速模块使用 BBR 拥塞控制算法控制数据发送速度. BBR 算法定义了 4 种 TCP 状态, 通过计算 BtlBw, RTprop 和增益系数, 转移 TCP 状态来改变数据发送行为, 如图 5 所示. 各状态下的增益系数列于表 2.

图 5 中箭头的指向表明状态转移路径, 各编号所代表的状态转移条件如下: ①为按照带宽 × 增益系数计算的速率发送数据, 结果带宽增加; ②为按照带宽 × 增益系数计算的速率发送数据, 结果带宽不增加; ③为已发送但未被 ACK 的数据量大于计算的 BDP; ④为已发送但未被 ACK 的数据量不大于计算的 BDP; ⑤为条件⑥不满足; ⑥⑨⑩为 10 s 内没有出现更小的 RTT; ⑦为延迟探测未结束; ⑧为延迟探测结束且带宽已达到最大; ⑪ 为延迟探测结束且带宽未达到最大.

3.2 接口层

接口层是台站与数据中心进行通信的接口. 这里

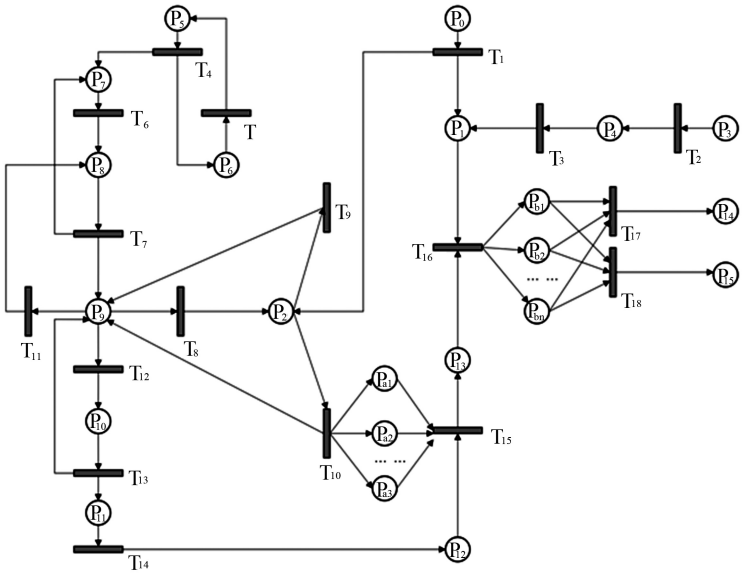


图 3 空间环境数据传输工作流模型

Fig. 3 Space environment data transfer workflow model

表 1 图 3 中库所和变迁的具体含义

Table 1 Specific meaning of the places and the transitions in Fig. 3

库所	含义	变迁	含义
P_0	服务端就绪	T_1	初始化服务端
P_1	控制器就绪	T_2	配置协议
P_2	监听器就绪	T_3	获取协议
P_3	配置器就绪	T_4	初始化客户端
P_4	协议栈就绪	T_5	连接超时
P_5	客户端就绪	T_6	启动定时器
P_6	心跳探测器启动	T_7	遍历文件
P_7	定时器就绪	T_8	发起协议验证
P_8	监测文件存储位置	T_9	未通过协议验证, 返回错误信息
P_9	待发送文件列表	T_{10}	通过协议验证, 分配接收器
$\{P_{ai}\}$	并行数据接收器	T_{11}	未通过协议验证, 返回 P_8
P_{10}	待发送文件	T_{12}	通过协议验证, 开启发送器
P_{11}	发送缓冲区	T_{13}	发送文件
P_{12}	接收缓冲区	T_{14}	BBR 加速
P_{13}	文件池	T_{15}	接收文件并存储至文件池
$\{P_{bi}\}$	并行文件解析器	T_{16}	启动文件解析器
P_{14}	文件服务器	T_{17}	二进制类数据解析入库
P_{15}	数据库	T_{18}	记录类数据解析入库

表 2 BBR 各状态增益系数

Table 2 Gain coefficient of each state of BBR

状态	增益系数
STARTUP	2∼3
DRAIN	1000/2885
PROBE_BW	5/4, 1, 3/4
PROBE_RTT	1

定义了台站与数据中心的文件传输协议, 包括访问控制验证、断点续传、文件命名规范和文件格式定义. 访问控制验证规定客户层在发送数据文件之前需向服务层发送用户名和密码, 验证通过后才能发送数据文件; 断点续传机制降低了网络不稳定对数据传输的影响; 文件命名规范规定了客户层允许发送的数据文件命名规则; 文件格式定义规定了数据在文件中的存储方式.

3.3 服务层

服务层是空间环境数据传输接收端, 由监听器、文件接收器、文件池、控制器以及数据处理引擎组成. 监听器接收客户层的访问请求, 为文件接收器分配客户层并发数据传输任务; 文件接收器负责数据文件的接收, 并将接收完成的数据文件放入文件池; 控制器为数据处理引擎分配文件池并行处理任务; 数据处理引擎对文件池中的数据文件进行解析处理.

3.4 存储层

存储层存储处理后的空间环境监测数据. 数据存储方式有两种: 二进制和图片类数据以文件形式存储在文件服务器中, 文件索引记录在数据库中; 记录类数据直接以数据库表的形式存储.

4 系统实现与实验

4.1 系统实现

空间环境数据传输系统采用 Java 语言编写, 界面使用 JavaFX Scene Builder 交互式设计, 生成的 FXML 文件通过标记语言描述用户界面. 使用 Apa-

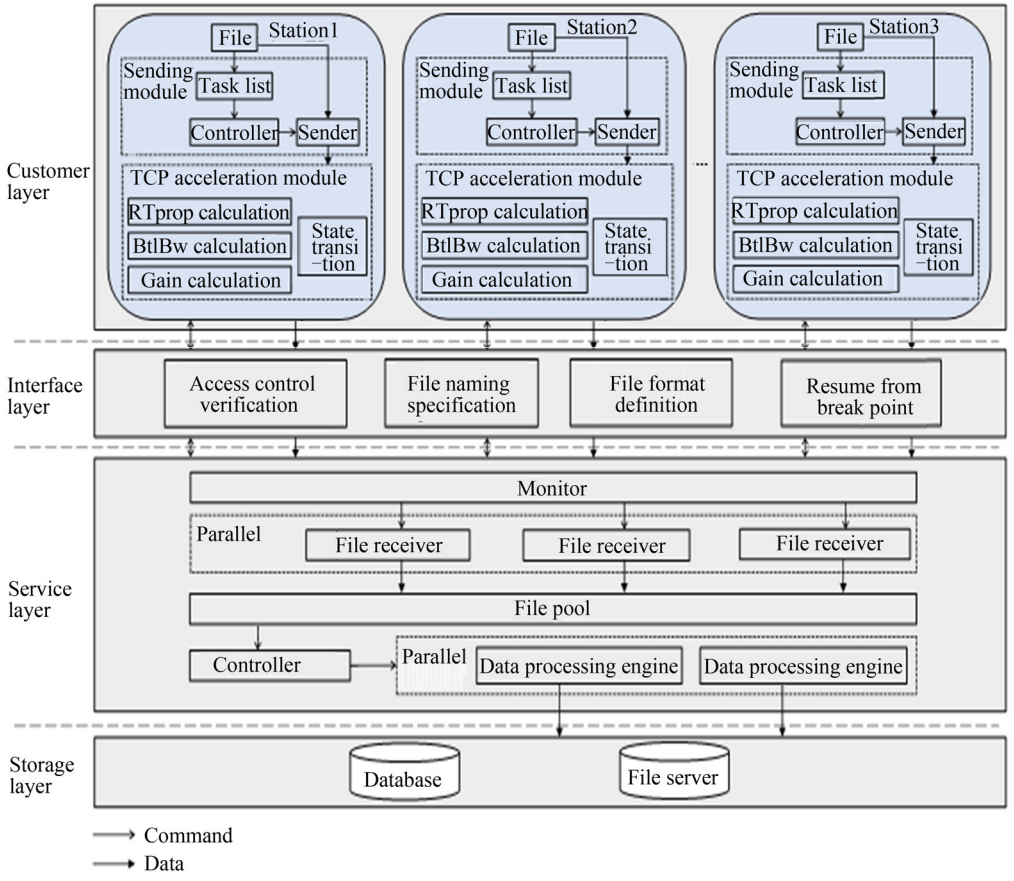


图 4 空间环境数据传输系统架构

Fig. 4 Architecture of space environment data transfer system

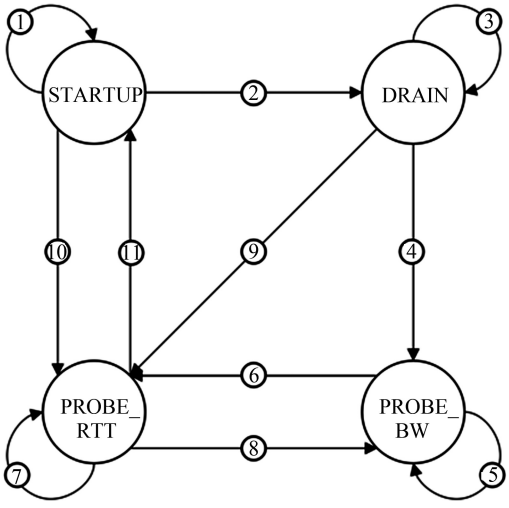


图 5 TCP 状态转移

Fig. 5 TCP state transition

che Commons IO 库实现对数据文件的监控, 当设定的目录下有新增文件时, 即将该文件加入任务列表中. 使用 Apache Commons Logging 库实现日志记录功能, 每天产生一个日志文件, 该文件以日期命名, 记录系统运行和数据传输情况, 同时系统窗口输出一些关键信息.

4.2 数据传输实验

为验证系统的有效性, 模拟空间环境监测台站与数据中心的数据传输, 进行了空间环境数据传输实验, 在本地模拟广域网复杂网络环境, 测试各类空间环境数据在该系统内的传输表现.

真实网络情况复杂, 网络参数不受控制, 因此选在本地进行实验, 模拟广域网高延迟、高丢包率环境下空间环境监测台站与数据中心的数据传输. 实验机器为 DELL OPTIPLEX 9010, 利用 Vmware 建立了 3 台 Ubuntu 虚拟机与宿主机组建局域网. 设置链路带宽为 10 Mbit·s^{-1}, 延迟为 100 ms, 丢包率为 10%, 模拟 3 个台站同时向数据中心传输数据. 实验选取某一天的空间环境数据分 3 个台站进行发送, 兼顾了大量小数据文件与大文件, 数据发送中断 10 s 后自动重连, 实验结果列于表 3. 传输数据文件 2385 个, 总

表 3 数据传输实验结果

Table 3 Results of data transfer experiment

台站 (虚拟机) 编号	文件数量	文件总大小/Byte	中断次数	总用时/s
1	272	1 104 708 832	2	1410
2	673	117 358 123	1	825
3	1440	352 416 742	3	1946

字节数 1 574 483 697, 无数据丢失.

5 结语

针对空间环境数据传输的传输频繁中断、数据不完整、传输速率低等问题, 分析了广域网传统 TCP 协议的不足, 采用 BBR 拥塞控制算法对 TCP 进行优化, 设计了空间环境数据传输的工作流模型, 并建立了空间环境数据传输系统. 实验结果表明, 该系统能够有效提高空间环境数据在广域网上的传输效率, 解决数据传输不完整和传输速率低的问题, 并能在传输中断后及时对数据进行补发, 保证了空间环境数据的可靠传输.

参考文献

[1] XIAO Jianjun, GONG Jiancun. Foreign space weather support capability building and the enlightenments to our country [J]. *Spacecraft Environ. Eng.*, 2015, **32**(1): 9-13 (肖建军, 龚建村. 国外空间天气保障能力建设及对我国的启示 [J]. 航天器环境工程, 2015, **32**(1): 9-13)

[2] CARDWELL N, CHENG Y, GUNN C S, *et al*. BBR: Congestion-based congestion control [J]. *ACM Queue*, 2016, **14**(5): 20-53

[3] KONG Jinsheng, REN Pingying. Summary of TCP network congestion control research [J]. *Comput. Technol. Dev.*, 2014, **24**(1): 43-46 (孔金生, 任平英. TCP 网络拥塞控制研究 [J]. 计算机技术与发展, 2014, **24**(1): 43-46)

[4] JACOBSON V. Congestion avoidance and control [J]. *ACM SIGCOMM Comput. Comm. Rev.*, 1988, **18**(4): 314-329

[5] STEVENS W. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms [S/OL]. IETF RFC 2001, 1999. https://www.rfc-editor.org/rfc/rfc2001.txt

[6] FLOYD S, HENDERSON T, Berkeley U C. The NewReno modification to TCP's fast recovery algorithm [S]. IETF RFC 2582, 1999. https://www.rfc-editor.org/rfc/rfc2582.txt

[7] MATHIS M, MAHDAVI J, FLOYD S, *et al*. TCP selective acknowledgment options [S]. IETF RFC 2018, 1996. https://www.rfc-editor.org/rfc/rfc2018.txt

[8] HA S, RHEE I, XU L. CUBIC: A new TCP-friendly high-speed TCP variant [J]. *ACM SIGOPS Operat. Syst. Rev.*, 2008, **42**(5): 64-74

[9] BRAKMO L S, PETERSON L L. TCP vegas: End to end congestion avoidance on a global internet [J]. *IEEE J. Selected Areas Commun.*, 1995, **13**(8): 1465-1480

[10] LIU H, XIA Y. Optimal resource allocation under TCP reno and vegas in complex communication networks [C]// IEEE International Symposium on Circuits and Systems. Piscataway, NJ: IEEE, 2015: 2676-2679

[11] PADMANABHAN V N, KATZ R H. Addressing the Challenges of Web Data Transport [D]. Berkeley, CA: University of California, Berkeley, 1998

[12] DAVERN P, NASHID N, SREENAN C J, *et al*. HTTPEP: A HTTP performance enhancing proxy for satellite systems [J]. *Int. J. Next-Gener. Comput.*, 2011, **2**(3): 242-256

[13] FLOYD S. TCP and explicit congestion notification [J]. *ACM Comput. Commun. Rev.*, 1994, **24**(5): 8-23

[14] BRISCOE B, BRUNSTROM A, PETLUND A, *et al*. Reducing internet latency: A survey of techniques and their merits [J]. *IEEE Commun. Surv. Tutor.*, 2016, **18**(3): 2149-2196

[15] CHEN Guangzhi, PAN Rong, LI Lei. Survey and research trends of workflow modeling techniques [J]. *Comput. Sci.*, 2014, **41**(S1): 11-17, 23 (陈广智, 潘嵘, 李磊. 工作流建模技术综述及其研究趋势 [J]. 计算机科学, 2014, **41**(S1): 11-17, 23)

[16] MOU Naixia, LI Jiling, LIU Feng, *et al*. Study on the methodology of oil spill contingency response workflow modeling based on the petri net [J]. *Mar. Sci. Bull.*, 2015, **34**(1): 72-75, 82 (牟乃夏, 李继领, 刘峰, 等. 基于 petri 网的海上溢油应急处置工作流建模方法研究 [J]. 海洋通报, 2015, **34**(1): 72-75, 82)

[17] CHEN Y F, LI Z W, ZHOU M C. Optimal supervisory control of flexible manufacturing systems by petri nets: a set classification approach [J]. *IEEE Trans. Autom. Sci. and Eng.*, 2014, **11**(2): 549-563