绝对延迟保证在 Web 应用服务器数据库连接池中的实现

吕健波，戴冠中，慕德俊
（西北工业大学 自动化学院 控制与网络研究所，西安 710072）

摘　要： 网络 QoS 机制不足以提供完全的端到端的性能保证，在由三层结构构成的电子商务网站中，基于反馈控制理论，提出并实现了在应用服务器的数据库连接池中的绝对延迟保证，对数据库连接池作出了改进，确保带有高优先级请求的平均排队延迟不超过设定的阈值。通过系统辨识建立了数据库连接池的近似线性时不变模型（LTI），并设计了绝对延迟保证控制器，为 Tomcat Web 应用服务器实现了数据库连接池中闭环系统所有部件。测试结果表明，即使在并发请求数量剧烈变化时，控制器的设计也是有效的。

关键词： Web 应用服务器；反馈控制；绝对延迟保证；系统辨识；数据库连接池
中图分类号： TP393　　**文献标志码：** A　　**文章编号：** 1001-3695(2012)05-1838-04
doi:10.3969/j.issn.1001-3695.2012.05.063

Implementation of absolute delay guarantee in database connection pool in Web application servers

LV Jian-bo, DAI Guan-zhong, MU De-jun
(*Control & Networks Institute, College of Automation, Northwestern Polytechnical University, Xi' an* 710072, *China*)

Abstract: Network QoS mechanisms are insufficient to provide complete end to end performance guarantees. In e-commerce Web sites composed of a three-tiered architecture, this paper proposed and implemented absolute delay guarantee in the database connection pool(DBCP) in Web application servers based on feedback control. It improved the database connection pool to ensure that the response time for requests with high priority was no more than the threshold specified. It set up the approximate linear time-invariant models of the DBCP through system identification and designed the absolute delay guarantee controller. It implemented all the components of closed-loops in a real DBCP for the Tomcat Web application server. Test results show that the controllers designing is effectively even if the number of concurrent requests varies severely.

Key words: Web application server; feedback control; absolute delay guarantee; system identification; database connection pool

电子商务网站通常由前端 Web 服务器、应用服务器和后端数据库三层架构组成。前端 Web 服务器处理工作负载的静态组成部分，如图像和不常变化的 HTML 页面。应用服务器提供一个环境来调用执行应用的业务和表示层逻辑的方法，这些方法包括 PHP scripts、active server pages(ASPs)和 Java servlets。应用逻辑发送大量的查询到存储了网站动态部分的数据库。应用服务器格式化被返回的数据库查询结果作为 HTML 页面，传回给前端 Web 服务器。最后，Web 服务器向客户端返回聚合内容。对于 Web 应用，其端到端的延迟主要由以下几个方面构成：a)连接延迟[1]，即 TCP 连接从建立到被服务线程或服务进程接收的时间间隔；b)处理延迟，即服务器读取客户请求，生成响应所需时间；c)传输延迟，主要指在客户与服务器间通过三次握手建立 TCP 连接，以及通过 TCP 连接传输请求和响应所需的时间。

1　相关工作

对于服务器端的 QoS 控制，很多研究者做了大量的工作，其研究表明，当发送请求的并行客户数目多于 Web application 服务器(如 Tomcat 等)设置的最大服务线程或服务进程数目时，连接延迟会大幅度增加。文献[1,2]通过不同的方法实现了 Web 服务器端相对连接延迟保证。文献[3]进一步提出了 percentile delay，并实现了相对和绝对的 percentile delay 保证。文献[4]则以延迟的另外一种形式——slowdown 作为评价指标，实现了基于 slowdown 的区分服务。针对静态请求大小呈重尾分布的特点，文献[5]在 IP 层将响应的 IP 包按照其对应请求的大小分为多个不同优先级的队列，运用 SRPF 调度算法，有效地控制了不同响应的发送次序，从而从整体上降低了静态请求的平均响应时间。文献[6,7]提供了在 Web 服务器上尽力延迟区分，其工作不是提供在不同服务级别之间差别程度的保证。

以上方法都是针对 HTTP 静态请求的，并且所有工作都是通过修改操作系统相关代码实现的。对于动态请求，当 Web 应用服务器繁忙时，处理延迟将大幅度增加。而现有的处理动态请求的 Web 应用服务器，如 Tomcat、IBM Web Sphere 等，都

收稿日期： 2011-09-08；**修回日期：** 2011-10-24

作者简介： 吕健波(1970-)，男，吉林人，讲师，博士，主要研究方向为 Web QoS、Web GIS 等(jianbolu@hotmail.com)；戴冠中(1937-)，男，上海人，教授，博导，主要研究方向为控制理论与控制工程、复杂网络中的控制问题；慕德俊(1963-)，男，山东人，教授，博导，主要研究方向为并行计算、网络安全.

不能提供类似于 Web 服务器下的延迟区分服务。

由于多数动态请求都包含数据库操作,本文从改进数据库连接池出发,提出了在 Web 应用服务器上实现区分服务的方法,即保证不同类别客户请求在申请数据库连接时享受到不同的延迟。

2　数据库连接池及其改进

2.1　数据库连接池的基本原理[8]

数据库连接池内维护着多个可以重复使用的数据库连接。初始化时,连接池生成一定数量的数据库连接,并将其状态全部设置为空闲。当客户请求申请连接时,如果连接池内存在空闲的连接,则连接池立即查找并返回给客户请求一个空闲的连接,同时将该连接的状态设置为繁忙;如果连接池内没有空闲的连接,并且连接池维护的总连接数未达到预设的最大值,连接池将新建一些数据库连接(通常按照默认的递增值,如 2 个),并将其返回给客户请求;如果此时连接池内的总连接数达到了预设的最大值并且没有空闲的连接,连接池将在等待一定时间后,再次查找空闲连接,并将结果返回给客户请求。客户请求使用完连接,将其退还给连接池时,连接池不会将该连接关闭,而是重新将该连接的状态设置为空闲,供其他客户请求使用。数据库连接池避免了过多的数据库连接的建立与关闭操作,提高了连接的利用率,降低了动态请求的处理时间,因而得到了广泛的应用。

2.2　数据库连接池的改进

对于原始的数据库连接池,在服务器繁忙时,不同的请求在申请空闲的连接时会等待相同的时间。为实现以等待时间为参考指标的区分服务,对数据库连接池进行改进。

改进模型 1 如图 1 所示。a)增加请求分类器。分类器按客户请求的源 IP、会话 ID 或请求的 http-header 等信息将客户请求分为普通请求和优先请求两类。每类请求形成一个 FIFO 队列,等待空闲的数据库连接。b)将数据库连接池划分为两个子池。将数据库连接池内的连接总数目固定,并在逻辑上划分为两个子池,每个子池服务于一类请求。将每个子池与一个信号量(semaphore)关联[8],每个信号量中许可(permit)的数目就是该子池中连接的数目,客户请求只有在获得一个信号量许可后,才能申请到一个空闲的连接。当客户请求将连接退还给子池时,将同时释放其获取的信号量许可。客户请求对空闲数据库连接的使用完全由有限的信号量许可控制。c)建立数据库连接池控制线程。运用一定的控制方法,动态调节两个信号量中许可的数目,即调节每组连接的数目 n_a 和 n_b,以达到不同数目的连接服务于不同类别的请求,使得在服务器繁忙时不同类别的请求在申请数据库连接时享受到不同的延迟。

改进模型 2 如图 2 所示。a)增加请求分类器,与模型 1 相同。b)建立数据库连接池控制线程。假定当前为第 k 个采样时刻,控制线程将根据第 $k-1$ 个采样周期每类请求的平均延迟,调节第 k 个周期中每类请求获取空闲连接的概率 $p(a)$ 和 $1-p(a)$,即保证不同类别的请求获得不同比例的资源。c)建立请求分派器。在数据库连接池出现空闲连接时,分派器将先生成一个随机数 x,该随机数是来自 $U(0,1)$ 分布的一个样本,比较 x 与 p_a 后,分派器将空闲的连接分派给相应类别的一个客户请求。如果第 k 个采样周期连接池累计服务了 S 个请求,则 A 类请求占 p_a,而 B 类请求占 $1-p_a$。

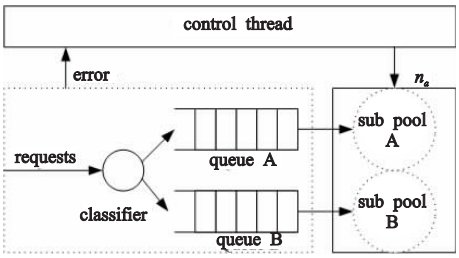

图1　改进模型1

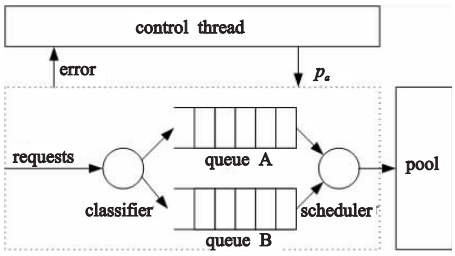

图2　改进模型2

2.3　改进后数据库连接池的特点

对于改进后的数据库连接池,本文设计了多组实验来分析两类请求的数据库连接数目比与延迟比之间的关系。数据库连接池中连接总数目设为 20,以请求的源 IP 作为分类标准,实验结果如表 1 所示。对于改进模型 1,由于资源调节时以数据库连接为单位,粒度较大,很难达到较为合适的延迟比例。对于改进模型 2,由于控制器的输出是每类请求获取空闲连接的概率值,并且在单位采样周期(本文设定为 15 s)内连接池接收的请求数目远多于有限的数据库连接数,因而能较好地实现细粒度的资源分配。因此,本文针对改进模型 2,设计了两种不同的控制器,分别用于实现客户请求在申请数据库连接时的绝对延迟保证和相对延迟保证。

表 1　改进模型 1 的性能

延迟比		数据库连接数目比			
		10:10	9:11	8:12	7:13
客户连接数目比	50:50	1.00	1.47	3.51	6.50
	50:60	0.72	1.06	2.25	5.10
	50:70	0.12	0.78	1.18	2.98

3　控制器的设计与闭环系统仿真

近年来,控制理论在网络服务器系统中的应用引起了人们的广泛关注。文献[9]将反馈控制方法运用到邮件服务器中,根据一定的利润模型,通过控制器动态调节服务器参数,提高了邮件服务器给运营商带来的收益。文献[2,3]将控制方法运用到 Web 服务器的性能改进之中。本文则运用反馈控制的相关理论,设计绝对延迟控制器,用于改进 Web 应用服务器的服务性能,为用户提供区分服务和性能保证。

3.1　绝对延迟保证控制器

文献[3,10]分别提出了 absolute connection delay guarantee 和 absolute percentile delay assurance,这两种延迟保证都是针对 TCP 连接延迟的。在 HTTP 1.1 协议下,多个客户请求可以通过一个持续连接(persistent connection)次序发送,连接延迟只会对该连接的第一个请求的反应时间产生影响。因此,保证高优先级请求的处理延迟比保证其所属 TCP 连接的连接延迟更具有意义。当多个并发的动态请求同时申请数据库连接时,绝

对延迟保证控制器应保证高优先级客户请求的平均延迟在给定的参考值之下。系统的输入为高优先级客户请求获取空闲数据库连接的概率,输出为此类客户请求的平均延迟。同样将闭环系统的性能指标定为:稳态误差 $e_{ss}=0$,调节时间 $t_s<300$ s。与 2.1 节类似,选择 PI 控制器作为设计目标,经过系统辨识、极点配置,得到绝对延迟保证控制器的参数为 $K_P=-0.5$,$K_I=-0.033$,采用增量式控制算法有 $\Delta x(k)=-e(k)-0.5e(k-1)$。

3.2 系统辨识

假定改进后的数据库连接池可以等效为一个线性模型(图 2)。该被控模型的输出是不同类别客户请求的平均延迟比,输入是不同客户请求队列获取下一个空闲数据库连接的概率比。假定 m 阶线性模型能达到较好的近似程度,那么相应的差分方程可记为

$$Y(k)=\sum_{i=1}^{m}[a_iY(k-i)+b_iX(k-i)] \tag{1}$$

其对应的 Z 域传递函数为

$$G(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{i=1}^{m}b_iz^{m-i}}{z^m-\sum_{i=1}^{m}a_iz^{m-i}} \tag{2}$$

按 3.4 节配置实验平台,数据库连接池中连接总数目设为 20;以请求的源 IP 作为分类标准;两台客户机分别模拟 50 个客户,利用 SURGE 向服务器发送动态请求。服务器在第 k 个采样时刻,计算采样周期内 A、B 类客户(假定 A 类为高优先级)的平均延迟比 $Y(k)$,并根据伪随机二位式序列当前的值将不同类别客户请求下一周期获取空闲数据库连接的概率比 $X(k)$ 设置为 1 或 1.5。输入的伪随机二位式序列为

$$\varepsilon(k)=\varepsilon(k-p)+\varepsilon(k-q)(\mathrm{mod}\ 2) \tag{3}$$

当 p 取 8,q 取 5 时,得到周期为 255 的输入序列。当 $\varepsilon(k)=1$时,调整 $X(k)$ 为 1;当 $\varepsilon(k)=0$ 时,调整 $X(k)$ 为 1.5。实验共进行了 40 min(2400 s)。所有实验数据均由控制线程写到服务器的日志文件。根据递推最小二乘估计的思想,对于此 m 阶系统,设由前 $N+m$ 次采样数据得到的参数的最小二乘估计为 θ_N,则在第 $N+m+1$ 次采样后,θ_N 可以修正为

$$\theta_{N+1}=\theta_N+\frac{P_N\varphi_{N+1}}{\varphi_{N+1}^{\mathrm{T}}P_N\varphi_{N+1}+1}[Y(N+m+1)-\varphi_{N+1}^{\mathrm{T}}\theta_N]\quad N>0 \tag{4}$$

$$P_{N+1}=P_N-\frac{P_N\varphi_{N+1}\varphi_{N+1}^{\mathrm{T}}P_N}{\varphi_{N+1}^{\mathrm{T}}P_N\varphi_{N+1}+1} \tag{5}$$

其中:

$$\varphi_N^{\mathrm{T}}=(Y(N+m-1),\cdots,Y(N),X(N+m-1),\cdots,X(N)) \tag{6}$$

取 $\theta_0=0$,$P_0=15I$,可得到系统参数的估计值 $\hat{\theta}^{\mathrm{T}}=(\hat{a}_1,\cdots,\hat{a}_m,\hat{b}_1,\cdots,\hat{b}_m)$。估计值与真实值间存在误差,可定义损失函数为

$$J(m)=\sum_{k=m+1}^{m+N}\{y(k)-\sum_{i=1}^{m}[a_iY(k-i)+b_iX(k-i)]\} \tag{7}$$

采用 F 检验法确定系统的阶数。设 n_1 和 $n_2(n_1<n_2)$是模型的两个近似阶数,构造统计量为

$$F=\frac{J(n_1)-J(n_2)}{J(n_2)}\times\frac{N-2n_2}{2(n_2-n_1)} \tag{8}$$

其中:N 为样本数目,$2n_1$ 和 $2n_2$ 分别对应系统阶数为 n_1 和 n_2 时系统参数的数目。文献[11]证明:当 N 充分大且 $n_2>n_1\geqslant m$ 时,F 服从分布 $F(2(n_2-n_1),N-2n_2)$。设 $1\leqslant m\leqslant 6$,按式(4)(7)(8)分别计算 $\hat{\theta}^{\mathrm{T}}$、$J(m)$以及 F 的值。取置信度 $\alpha=5\%$,则 $F_{0.05}(2,86)=3.10$。因为 $F_{0.05}(2,86)>1.9981$,即 m 取 1 和 2 时,$J(m)$无显著变化,所以 $m=1$ 时,式(1)即达到了足够的近似程度。因此原系统为一阶线性系统,其中系统参数 $\hat{\theta}^{\mathrm{T}}=(-0.0172,0.7463)$。

3.3 实验平台与实验配置

在 Testbed 中,所有 PC 机均配置有 Pentium 4 2.8 GHz 处理器,512 MB 内存。改进的 SURGE[3] 作为测试工具,运行在 Redhat 9 上,向 Web 服务器发送动态请求。Web 服务器为 Apache(版本为 httpd2.0.53),静态请求可由 Apache 直接处理。在 Apache 内,mod_jk 模块将动态请求按一定方式(如加权方式)分派给后端的多个应用服务器,以实现负载均衡。Web 应用服务器为 Tomcat(版本为 5.5.17)。由于本文主要目的不是 Web 负载均衡问题,故在实验中只使用一台 Web 应用服务器。后端数据库为 Oracle 9i。服务器端结构如图 3 所示。

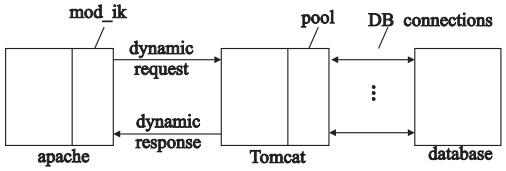

图3 服务器端结构

3.4 动态负载的分布

对于 HTTP 静态请求,可以根据请求对应文件的大小估算请求的处理时间。而动态请求通常并不与静态文件关联,而且需要利用较多的系统资源,因此难以统计其处理时间的分布。本文使用三种属于不同分布的动态负载,测试控制器作用下闭环系统的性能,三种负载大小的均值都为 350 ms。第一种负载为定长负载,即假定每个请求都需要 350 ms 的处理时间;第二种负载的大小服从均匀分布,即假定每个请求的处理时间为来自 $U(0,700)$的一个样本;第三种负载的大小服从重尾分布,假定有界帕累托分布(bounded-Perato,其累积分布函数如式(9))为实际的重尾分布模型。另外由于双峰分布(bimodal,其密度函数如式(10))在 $B>>E(X)>A$ 且 $\alpha\approx 1$ 时,具有明显的重尾特征,并且能很好地近似其他类型的重尾分布,如有界帕累托分布和幂率分布(power-law)[12]。因此,本文在实际程序中使用双峰分布生成第三种负载,并取 $A=50$,$B=6050$,$\alpha=0.95$,以保证负载的均值为 350 ms。在程序设计时,使用服务线程对象的函数成员 sleep()模拟实际的请求处理时间。

$$F(x)=\frac{1-(m/x)^{\gamma}}{1-(m/M)^{\gamma}} \tag{9}$$

其中:$M>>m$, $M\geqslant x\geqslant m$,$\gamma\in(0,2)$

$$f(x)=\alpha\delta(x-A)+(1-\alpha)\delta(x-B)$$

$$\delta(x)=\begin{cases}1 & x=0\\ 0 & \text{其他}\end{cases} \tag{10}$$

3.5 实验配置

本文共设计两组实验,用于分析闭环系统在不同的参考值和不同类别的客户并发请求变化的情况下的性能,并使用了第二种测试负载。第一组实验中,参考值设定为 1.5 s,即保证高优先级客户请求的延迟不高于 1.5 s,客户机配置如表 2 所示,第三台客户机使得高优先级的并发客户请求发生变化。在第二组实验中,参考值设定为 2 s,客户机配置如表 3 所示,第三台客户机使得低优先级的并发客户请求发生变化。

在所有实验开始时,p_a 设置为 0.5,即分派器按 50% 的概

率将空闲的数据库连接分别分派给两类请求。

表 2　绝对延迟保证实验客户机设置 1

客户机	优先级	客户数目	运行时间/s
1	高(A)	70	1800
2	低(B)	70	1800
3	高(A)	20	600

表 3　绝对延迟保证实验客户机设置 2

客户机	优先级	客户数目	运行时间/s
1	高(A)	80	1800
2	低(B)	80	1800
3	低(B)	40	600

3.6　实验结果及其分析

如图 4 和 5 所示,在两组实验中,系统稳定时 p_a 能较好地保持在固定的数值上,达到对资源的较细粒度的分配。对于不同的参考值和不同的负载变化情况(高优先级并发请求数目变化、低优先级并发请求数目变化),闭环控制系统都能保证高优先级请求申请连接时的平均延迟在参考值附近。

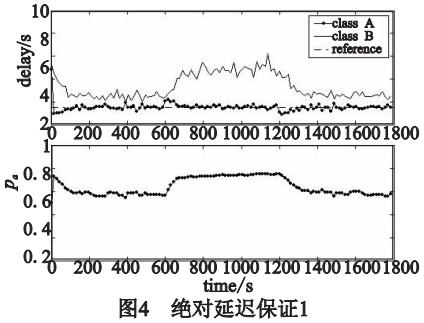

图4　绝对延迟保证1

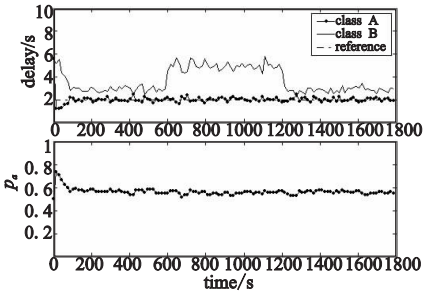

图5　绝对延迟保证2

4　结束语

端到端的延迟是评价 Web 应用的一项重要指标。在支撑 Web 站点的应用服务器中,数据库连接申请延迟构成了服务器繁忙时端到端延迟的重要部分。本文针对 Web 应用服务器中数据库连接池管理的不足,结合控制理论的基本原理,设计了绝对延迟控制器,实现了服务器上的区分服务。本文首次将反馈控制方法运用到了 Web 应用服务器的性能改进中。由于本文的改进方法由 Java 实现,因此可以方便地移植到 Tomcat 之外的其他 Web 应用服务器中。实验结果显示,在控制器的作用下,动态负载下的 Web 应用服务器保证了不同类别的客户请求申请数据库连接时享受到了不同的延迟,能够较好地为客户提供区分服务和性能保证。

此外,根据 HTTP 协议和服务器软件设计上的特点,Web 服务器和 Web 应用服务器的工作流程可以划分为 TCP 连接管理、HTTP 请求调度、数据库连接等资源管理和 HTTP 响应内容控制四个层次。因此,在下一步的工作中,将把反馈控制方法综合运用到不同的层次上,从而从整体上提高服务器的服务性能,更好地实现服务器系统上的区分服务和性能保证。

参考文献:

[1] CHANDRA A, PRADHAN P, TEWARI R, *et al*. An observation based approach towards self-managing Web servers[J]. Computer Communications,2006,29(8):1-15.

[2] ABDELZAHER T F, STANKOVIC J A, LU Cheng-yang, *et al*. Feedback performance control in software services[J]. IEEE Control Systems,2003,23(3):74-90.

[3] KANODIA V, KNIGHTTLY E W. Ensuring latency targets in multi-class Web servers[J]. IEEE Trans on Parallel and Distributed Systems,2003,14(1):84-93.

[4] WEI Jian-bin, XU Cheng-zhong. Design and implementation of a feedback controller for slowdown differentiation on Internet servers [C]//Proc of Special Interest Track and Posters of the 14th International Conference on World Wide Web. New Yook: ACM,2005.

[5] HARCHOL-BALTER M, SCHROEDER B, BANSAL N, *et al*. Size-based scheduling to improve Web performance[J]. ACM Trans on Computer Systems,2003,21(2):207-233.

[6] ABDELZAHER T F, BHATTI N. Web server QoS management by adaptive content delivery[C]//Proc of International Workshop on Quality of Service. 1999:216-225.

[7] ALMEDIA J, DABU M, MANIKNTTY A, *et al*. Providing differentiated levels of service in Web content hosting[C]//Proc of the 1st Workshop on Internet Server Performance. 1998.

[8] LEA D. Concurrent programming in Java: design principles and patterns[M]. 2nd ed. [S.l.]: Addison Wesley Longman,2000.

[9] DIAO Yi-xin, HELLERSTEIN J L, PAREKH S S. Using fuzzy control to maximize profits in service level management[J]. IBM Systems Journal,2002,41(3):403-420.

[10] LU Cheng-yang, ABDELZAHER T F, STANKOVIC J A, *et al*. A feedback control approach for guaranteeing relative delays in Web servers[C]//Proc of IEEE Real-Time Technology and Applications Symposium. 2001.

[11] ASTROM K J, EYKHOFF P E. System identification: a survey[J]. Automatica,1971,7(2):123-162.

[12] PSOUNIS K, MOLINERO-FERNANDEX P, PRABHAKAR B, *et al*. Systems with multiple servers under heavy-tailed workloads[J]. Performance Evaluation,2005,62(1-4):456-474.

[13] BARFORD P, CROVELLA M E. Generating representative Web workloads for network and server performance evaluation[J]. Measurement and Modeling of Computer Systems,1998,26(1):151-160.

[14] CROVELLA M E, BESTAYROS A. Self-similarity in world wide Web traffic: evidence and possible causes[J]. IEEE/ACM Trans on Networking,1997,5(6):835-846.

[15] The apache software foundation[EB/OL]. http://www. apache. org.

[16] WOOD L, HOLLIDAY P. Internet-Draft-06, Using HTTP for delay/disruption-tolerant networks[S]. 2010.

[17] ZHANG Guo-hua, LIU Yong-he. Congestion management in delay tolerant networks[C]//Proc of the 4th Annual International Conference on Wireless Internet. 2008.

[18] AL-SIYABI M, CRUICKSHANK H, SUN Zhi-li. Quality of service provisioning for delay tolerant network by implementing admission control model for aircrafts bundles data transmission[C]//Proc of the 6th International Wireless Communications and Mobile Computing Conference. 2010.