

# DTN Web 服务器基于自适应控制的成比例延迟区分服务

孙启禄<sup>1</sup> 戴冠中<sup>1</sup> 潘文平<sup>2</sup>

(西北工业大学自动化学院 西安 710072)<sup>1</sup> (南京航空航天大学自动化学院 南京 210016)<sup>2</sup>

**摘 要** DTN 网络的出现为未来的 Internet 应用提供了一种全新的平台,在 DTN 网络上部署支持服务区分服务的 Web 服务器方面的研究仍为空白。首先设计了一种支持区分服务的 DTN Web 服务器,并在其上实现了基于自适应控制的成比例延迟区分服务。自适应控制器根据预设的延迟区分参数,通过动态计算并调节各个客户类别的服务线程数目,可保证 DTN Web 服务器上高优先级客户具有较低的平均连接延迟而各个客户类别的平均延迟比保持不变。仿真结果表明,在动态变化的负载、不同的负载分布以及变化的参考输入之下,控制器作用下的 DTN Web 服务器都能可靠地提供成比例延迟区分服务。

**关键词** DTN 网络, Web 服务器, 自适应控制, 成比例延迟区分服务

**中图法分类号** TP393 **文献标识码** A

## Adaptive Control-based Proportional Delay Differentiated Service on DTN Web Servers

SUN Qi-lu<sup>1</sup> DAI Guan-zhong<sup>1</sup> PAN Wen-ping<sup>2</sup>

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)<sup>1</sup>

(College of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)<sup>2</sup>

**Abstract** The DTNs construct a new platform for the future network applications, but nothing has been done for the research to deploy the Web servers which can provide service differentiation on DTNs. This paper designed a DTN Web server supporting differentiated service, and implemented adaptive control-based proportional delay differentiated service. Dynamically the adaptive controller can calculate and adjust the number of service threads in Web server for different types of clients according to the pre-specified delay differentiation parameters. Simulation results demonstrate that proportional delay differentiation can be achieved on the DTN Web server even under dynamic workloads, different workload distributions and varying references.

**Keywords** DTNs, Web servers, Adaptive control, Proportional delay differentiated service

近年来, Web 应用得到了飞速发展, Internet 上 Web 流量超过了 Internet 数据总流量的 65%<sup>[1]</sup>。同时,随着基础网络从相对静态同构的环境(有线网络)变得更加动态异构(例如 Wi-Fi 接入网、蓝牙接入网和卫星网络),迫切需要新的架构和协议来提供更好的 Web 服务。DTN (Delay/Disruption Tolerant Network)<sup>[2]</sup> 网络的出现为未来的 Internet 应用架构提供了一种可能的平台。DTN 网络技术旨在为缺乏持续端到端连接、往返传输时间(RTT)很高、信道质量低劣或者非对称数据传输的极端和挑战性网络(Challenged Network)环境提供通信保证。Internet 的 TCP 协议在 DTN 网络中被包裹传输协议(Bundle Protocol)取代,而 Internet 的 HTTP-Over-TCP 机制在 DTN 网络中演化为 HTTP-DTN<sup>[3]</sup> 或者 HTTP-Over-Bundle<sup>[4]</sup> 机制。但是,两者都未涉及网络服务质量保证(QoS)问题。考虑到未来 DTN Web 服务的广泛性,以及 DTN 应用领域的特殊性,对 DTN Web 服务器 QoS 问题进行研究具有现实和长远意义。

与传统 Internet 网络 QoS 技术相比, Web 服务器端系统

提供区分服务和性能保证,可以为 Web 客户提供更为令人满意的服务性能。响应延迟是评价 Web 服务器性能的关键性能指标,而成比例延迟区分服务<sup>[5]</sup>是 Internet 上的重要区分服务模型。DTN 网络的应用环境使得优先客户优先获得资源访问权更有必要,例如在军用战术网中,指挥官需要优先获取战场传感器资源数据;在深空网络中,航天器测控信息必须保证最小的响应时延。同时,相对于传统 Internet, DTN 网络节点间通信机会很少,特别是在网络极度稀疏的条件下, Web 服务器的优先服务更为关键,更容易获得较高的 QoS。因而,除了在网络及传输层进行 QoS 控制,在 DTN 网络 Web 服务器端提供区分服务不可或缺。但当前 DTN 网络研究仍停留在协议设计和仿真验证阶段,对该领域的研究尚为空白。

本文在分析成比例延迟区分服务模型以及 DTN 网络 Web 应用延迟构成的基础上,总结了相关的研究成果,基于 HTTP-Over-Bundle 的原型 DTN Web 服务器,通过在线系统辨识,建立了服务器的二阶 ARMA 模型,并在 DTN Web 服务器上实现了基于自适应控制的成比例区分服务,最后通过

到稿日期:2009-03-25 返修日期:2009-06-10 本文受国防基础科研项目(C2720061361)资助。

孙启禄(1977-),男,博士生,主要研究方向为 DTN 网络及 QoS 等, E-mail: tigersql@gmail.com;戴冠中(1937-),教授,博士生导师,主要研究方向为自动控制、信息安全;潘文平(1980-),博士,主要研究方向为 QoS 等。

系统仿真给出了结论。

## 1 成比例延迟区分服务及相关工作

### 1.1 成比例延迟区分服务

文献[5]最早提出成比例延迟区分 (proportional delay differentiation, PDD) 服务模型。假设网络中的数据包或请求可以划分为  $n$  个服务类别, 其中第  $i$  ( $1 \leq i \leq n$ ) 个服务类别的延迟区分参数 (differentiation parameter) 与平均排队延迟分别为  $w_i$  和  $\bar{d}_i$ , 则网络节点 (如路由器等) 在繁忙时应保证各个服务类别的平均延迟比满足  $\frac{\bar{d}_i}{\bar{d}_j} = \frac{w_i}{w_j}$  ( $1 \leq i \leq n, 1 \leq j \leq n$ )。

比例延迟保证属于相对区分服务范畴, 因此并不保证特定服务类别的平均排队延迟低于某个阈值。但是可以保证在任意负载强度之下, 高优先级的服务类别总是具有相对较低的平均排队延迟, 即保证高优先级服务类别获得更优的服务。

### 1.2 DTN 网络 Web 服务延迟特性分析

PDD 主要集中在网络层或传输层研究。本文将成比例区分服务引入到部署 Web 应用的 DTN 服务器端系统中, 为此首先分析一下 DTN Web 应用的延迟构成。

当 Web 客户通过 Bundle 协议访问 Web 站点时, 其感受到的响应延迟包含以下几个方面: Web 客户发送请求数据包裹 (Bundle) 到 DTN Web 服务器需要的时间  $t_1$ ; DTN Web 服务器启动服务进程或服务线程的连接延迟时间  $t_2$ ; 读取请求包裹, 解析请求, 生成响应包裹的延迟时间  $t_3$ ; DTN Web 服务器将响应包裹通过 DTN 网络传输到 Web 客户所需的时间  $t_4$ 。其中  $t_1$  和  $t_4$  由 DTN 网络间断或者拥塞状况决定, DTN 的网络特性决定了两者数值波动较大, 在中间节点进行 QoS 控制需要结合路由协议进行, 超过了本文讨论的范围;  $t_3$  与请求类型及服务器端系统的软硬件配置相关, 另外 DTN 网络尽量减少 Web 客户和服务器之间的交互, 因而一次请求的响应数据包裹往往很大, 可能将相关网页、脚本、图片甚至视频全数打包到一个包裹中, 导致  $t_3$  较大;  $t_2$  主要与 Web 服务器中连接管理模块的工作方式相关, 但由于  $t_3$  的不可忽略和网络连接的间断, 使得在 DTN 网络, 尤其是低带宽环境里,  $t_2$  对 QoS 影响比 Internet 网络明显增大。同时, 由于 DTN 往往不能预知下一次连接机会何时到来, 并且无法预测响应 Bundle 的大小, 保证某一服务的排队时延低于某一阈值并没有实际意义, 亦即绝对区分服务并不适用于 DTN。

基于上述分析, 本文为 DTN Web 服务器系统建立了针对  $t_2$  的基于自适应控制的成比例延迟区分服务, 即保证不同类别的客户在繁忙的 DTN Web 服务器上具有不同的连接延迟。

### 1.3 相关工作

为了能够在 Web 服务器端系统上实现以延迟作为评价指标的性能保证和区分服务, 研究者们做了大量工作。文献 [6, 7] 等, 通过不同的方法, 实现了 Web 服务器端相对和绝对连接延迟保证。文献 [8] 则以延迟的另外一种形式, 即 Slowdown 作为评价指标, 实现了基于 Slowdown 的面向多个服务类别的区分服务。由于静态 HTTP 请求的大小呈重尾分布 (heavy-tailed distribution), 因此文献 [9] 在网络层运用 SRPF (shortest remain processing first) 调度算法, 将 HTTP 响应的 IP 包按照其所对应请求的大小划分为多个不同优先级的队

列, 通过控制不同 HTTP 响应的发送次序, 从整体上降低了静态请求的平均响应时间。另外, 有不少研究工作是针对其它区分服务性能指标的, 比如文献 [10] 提出了基于 Web 缓存相对命中率 (relative hit ratio) 的区分服务, 文献 [11] 提出了针对 Web 请求成比例拒绝率 (proportional rejection rate) 的区分服务等。

## 2 支持 PDD 服务的 DTN Web 服务器实现

### 2.1 DTN Web 服务器的设计

HTTP-DTN<sup>[3]</sup> 概念架构引入了新的 Content-\* 头对 HTTP 数据包进行封装, 不再需要双向会话, 其优点是概念非常简单, 不改变已有的 Web 技术体制, 可以复用大部分已有代码和功能, 但该机制尚在草案阶段。Jörg Ott 等在文献 [4] 提出了 3 种 HTTP-Over-Bundle 实现方式: 端到端、基于代理和基于网关, 并对后两者进行了原型实现。HTTP-DTN 和 HTTP-Over-Bundle 机制的共同点是使用 Multipart MIME 对 Web 信息进行封装, 利用 Bundle 协议进行传送。两种机制为在 DTN 网络中实现 Web 服务提供了可行的方案, 但是两种方案均不能涵盖所有的应用领域。

基于已有的研究成果, 本文总结了 4 种 DTN Web 信息获取的方案, 如图 1 所示。A 方案与 HTTP-DTN 方案类似, 不改变现有的客户端和服务器, 直接用网关的方式实现 Web 数据包的封装和解包, 用 Bundle 协议再通过路由器 (BR) 在 DTN 网络上传递。B 方案不改变客户端配置, 将 Internet Web 服务器升级为可直接处理 Bundle 的 Web 服务器, 同时在 Web 客户端使用代理 (Proxy) 完成 Http 数据包和 bundle 的转换。C 方案与 B 相反, 将客户端进行升级, 而服务器端保持不变, 在服务器端使用代理。D 方案是最终演化方案, 客户端和服务器端都升级为 DTN 架构, 直接通过 BR 进行数据传递。A 方案力求保持底层 Internet 端系统架构不变, 方便部署和兼容, 但网关的存在势必带来巨大的性能瓶颈, 尤其是服务器端网关承受着巨大的负载压力时却不能改变底层进行优化。D 方案作为最终解决方案, 能够提供最好的性能, 但是系统配置部署开销较大, 在当前 DTN 技术尚不成熟的前提下全面采用也不明智。B 方案和 C 方案可以看成 A 方案和 D 方案的折衷, 采用代理的方式, 只改变部分端系统结构, 可以对 DTN 和 TCP 保持较好的兼容, 并且实现难度较低。同时, B/S 架构的非对称性使得 B 方案相对 C 方案具有一定优势, 采用现有的成熟 Web 客户端, 可以大大减少部署代价。表 1 对 4 种方案的优缺点进行了简单比较。

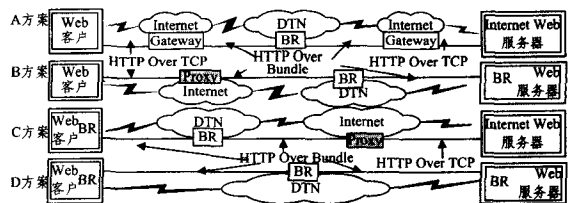


图 1 DTN Web 信息获取备选方案

表 1 4 种方案优缺点比较

方案	A	B	C	D
服务器端	不改变	改变	不改变	改变
客户端	不改变	不改变	改变	改变
网络性能	低	中	中	高

配置代价	低	低	中	高
兼容性	高	中	中	低

本文基于 B 方案设计了支持 PDD 服务的 DTN Web 服务器。采用和文献[12]类似的方式,使用 DTN2 作为底层,负责 Bundle 在 DTN 网络中的存储转发;DTN-Ruby 作为 Ruby 接口层,负责从 DTN2 接收和发送 Bundle,并实现与上层 Ruby 应用程序的交互;采用 Mongrel 作为上层 Http 服务提供者。DTN2 是 DTN 网络架构和 Bundle 协议(BP)的参考,实现了 BP 的大部分功能,为上层应用提供简单的 API 接口,当前版本为 2.6.0。DTN-Ruby 是 DTN2 的一个 Ruby 语言绑定扩展库,能够为上层 Ruby 应用提供 DTN2 的功能接口。Mongrel 是用 Ruby 语言实现的一个 HTTP 库和简单 Web 服务器,基于单进程-多线程架构。我们对 Mongrel 进行扩展,使之在支持 BP 的基础上,通过增加线程管理模块(Thread Management Module, TMM)实现区分服务和性能保证,如图 2 所示。

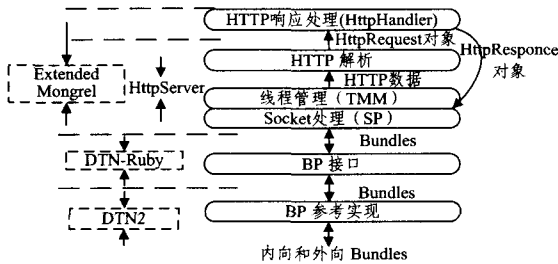


图 2 支持 PDD 服务的 DTN Web 服务器架构

## 2.2 支持 PDD 服务的 DTN Web 服务器实现

Mongrel 是提供 Web 服务的关键,主要由 3 个功能组件 HttpServer, HttpHandler 及 URIClassifier 组成,并维护 HttpRequest, HttpResponse 两个关键对象。其中 HttpHandler 负责 Http 资源的处理,接收 HttpRequest 对象,生成 HttpResponse 对象;URIClassifier 负责请求路径和 Handler 列表的映射(图中未标出);HttpServer 负责接收并解析收到的 Http 请求,并进行 Socket 处理(SP)。对 HttpHandler 和 SP 用类似于文献[12]的方式加以改进,使之能够支持 BP 协议,在此不赘述。

Mongrel 的连接管理在 HttpServer 组件中实现,该组件可以简化为一个包含多个工作者线程和一个主进程的线程队列模型。每个工作线程可对应一个客户连接,主进程接受到新的客户连接时,将创建新的工作者线程并按照 FIFO 的方式加入队列。由此可见,Mongrel 并不支持区分服务与性能保证。本文对 Mongrel 的 Http Server 进行改进,增加 TMM,使之能够同时服务于两个或者多个不同的客户类别(为简化分析,本文仅针对两个优先级级别,如图 3 所示)。具体改进如下:

(1)增加优先级分类器,用于将新到的  $n$  个客户连接划分到相应的客户队列。

(2)使用双队列管理客户连接,每个队列对应一个客户类别,其中 A 类为高优先级类别,管理  $k$  个高优先级工作者线程;B 类为低优先级类别,管理剩余  $n-k$  个工作者线程。

(3)增加与工作者线程并列的控制线程,用于调节每个客户队列实际的工作者线程数目。

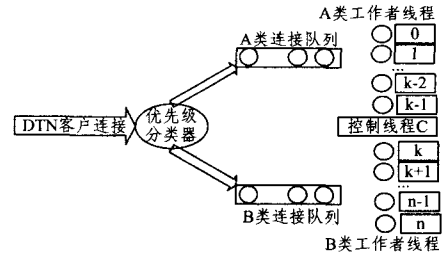


图 3 线程管理模块(TMM)

## 3 自适应控制算法及闭环系统实现

### 3.1 在线系统辨识与自校正控制算法

考虑用以下 ARMA 模型描述 TMM 模块(暂不考虑控制线程)。其中  $\{y(t)\}$  和  $\{u(t)\}$  分别表示系统模型的输出和输入序列,即 A, B 两个客户连接类别的平均排队延迟比  $\frac{\bar{d}_A(t)}{\bar{d}_B(t)}$

和相应的工作者线程数目比  $\frac{\bar{r}_B(t)}{\bar{r}_A(t)}$ ,  $z^{-1}$  为单位后移算子。

$$A(z)y(t) = B(z)u(t) \quad (1)$$

其中,  $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$ ,  $B(z) = b_1 z^{-1} + b_2 z^{-2}$ 。

另外,定义参数向量  $\theta$  和信息向量  $\varphi(t)$  分别为

$$\theta^T = (a_1, a_2, b_1, b_2)$$

$$\varphi^T(t) = (-y(t-1), -y(t-2), u(t-1), u(t-2))$$

根据参数估计的投影算法<sup>[13,14]</sup>,可按式(2)在线递推系统模型的参数。

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\varphi(t)}{\|\varphi(t)\|^2} (y(t) - \varphi^T(t) \hat{\theta}(t-1)) \quad (2)$$

其中,  $\hat{\theta}(t)$  为  $t$  时刻模型参数  $\theta$  的估计,  $\hat{\theta}(0)$  为界实向量,范数  $\|x\|^2 = \text{tr}(x^T x)$ 。对于确定系统,投影算法给出的参数估计可按指数速度收敛于真实参数。

再考虑包含控制线程的 TMM 模型。设 PDD 服务的参考值为  $y^*$  (即  $\frac{\bar{d}_A}{\bar{d}_B} = y^*$ ),按照投影自校正控制算法<sup>[13,14]</sup>,若在  $t$  时刻按式(3)修正系统控制律,则最终系统输出的跟踪误差为零。

$$y^* = \varphi^T(t) \hat{\theta}(t) \quad (3)$$

令  $\varphi^T(t) = (u(t), \varphi_0^T(t))$ ,  $\hat{\theta}^T(t) = (\beta(t), \hat{\theta}_0^T(t))$ ,可得  $t$  时刻控制律为

$$u(t) = \frac{1}{\beta(t)} (y^* - \varphi_0^T(t) \hat{\theta}_0(t)) \quad (4)$$

### 3.2 闭环系统实现的相关问题

#### 3.2.1 延迟时间的计算

在 TMM 中,记录每个线程的时间信息 Thread[:start\_on] 和 Thread[:time\_to], 分别对应  $t_c$  和  $t_d$ 。连接建立后记录 Thread[:start\_on] (利用 Ruby 的 Time.now 方法获取当前系统时间), 当此连接获得处理时填充 time\_to。time\_to 与 start\_on 之差,即为该客户的延迟。另外,定义 QUEUE TIME 结构类型的变量,用于记录采样周期内两个连接队列上被分配到工作者线程连接的总个数及其总延迟时间。各个线程使用 QUEUE TIME 类型的变量时,使用互斥量达到线程同步。新定义的 QueueTime 数据结构表述如下:

$$\text{QUEUE TIME} = \text{OpenStruct.new}(\dots)$$

```

:time_all => 0, //总延迟时间
:connection_counter => 0 //客户连接数
)

```

### 3.2.2 采样及线程比调整

利用 Ruby 的 Time.now 方法,在每个采样时刻,计算各类客户的平均延迟后对各类服务线程比做相应的调整。HT-TP1.1 的超时时间默认为 15s,故将采样时间设定为 15s。

使用数组变量 ThreadStatus,用于记录  $n$  个线程的状态。ThreadStatus[ $i$ ]( $0 \leq i < n$ ) 为 0,则线程  $i$  服务于 A 类客户,为 1 则服务于 B 类客户。控制线程在每个采样时刻根据  $u(t)$  的值调整该数组的值,以达到调整线程比的目的。

## 4 闭环系统仿真

### 4.1 仿真试验床

在仿真实验中,服务器端处理器为 Intel 奔腾双核 E5200 2.5GHz,内存 1GB,操作系统为 Ubuntu Linux 8.04;客户端处理器为 Pentium 4 2.8GHz,内存 512MB,操作系统为 Ubuntu Linux 8.04。服务器端运行加入 TMM 模块的 Mongrel 服务器。客户端使用 SURGE 1.0 (Scalable URL Reference Generator)<sup>[15]</sup>作为网络流量发生器,同时模拟多个并发的 Web 客户向服务器发送请求和接收响应,以引发具有自相似(突发性)特征的网络流量<sup>[16]</sup>。两台客户机  $C_A$  和  $C_B$  上的 SURGE1.0 分别模拟 A,B 两类客户。在客户机和服务器间采用一台与  $C_A$  配置相同的主机 T,其上运行 DTN2 2.6,模拟 DTN 路由器节点(BR),以实现图 1 所示 B 方案的 DTN 网络环境。

### 4.2 仿真试验及其结果分析

为验证支持 PDD 服务的 DTN Web 服务器闭环系统的性能,设计 3 类试验。第一类试验在均匀分布 (Uniform Distribution) 工作负载 (Workload) 下,将动态参数的自适应控制器与具有固定参数的 PI 控制器的性能进行比较,第二类试验分别使用均匀分布工作负载和重尾分布工作负载检验自适应控制器对于不同类型负载的适应能力;第三类试验工作使用动态变化的参考输入,用于评价闭环系统对动态变化的参考输入的适应能力。以下分别予以论述。

#### 4.2.1 自适应控制与 PI 控制下的比例延迟区分服务

为了比较自适应控制器与 PI 控制器的性能,本文按文献 [7] 通过离线系统辨识和控制器设计实现了一个固定参数的 PI 控制器。

DTN Web 服务器上 A,B 两类客户的工作者线程数量比设定为 1:1,同时设定服务器最大并发线程数 (Num\_processors) 为 90。SURGE 产生的工作负载为均匀分布,参考值设定为 0.5,即  $\frac{\bar{d}_A}{\bar{d}_B} = y^* = 0.5$ 。开始测试时,A,B 两类客户的客

户数均为 90 个。两台客户机上 SURGE 程序同时启动。控制器 30s (待客户机运行稳定) 后启动。在 375s,在客户机  $C_B$  上启动另一个 SURGE 进程,使得 B 类客户增加 90 个。测试时间为 960s。仿真结果如图 4 所示。其中,(a) 和 (c) 分别描述 PI 控制和自适应控制下两类客户的延迟波动,(b) 和 (d) 是根据 (a) 和 (c) 计算的平均延迟比与参考值偏离情况。显然,两类控制器均可使两类客户的平均延迟比收敛到参考值,实现 PDD 服务。但较 PI 控制器而言,在自适应控制器的作用下,平均延迟比波动更小。这表明通过在线系统辨识,自适应控制算法可快速修正控制律,调整服务线程数目,使得闭环系

统在负载变化时快速收敛到参考值,取得比 PI 控制器更优的动态性能。

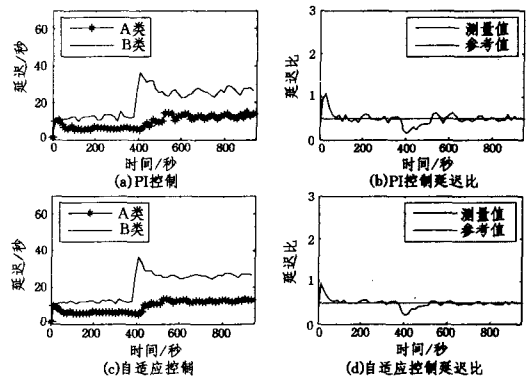


图 4 自适应控制器与 PI 控制器性能比较

#### 4.2.2 不同分布工作负载下的自适应比例延迟区分服务

第二类试验用于评价闭环系统对于不同分布的工作负载的适应能力。我们使用改进的 SURGE 分别产生均匀分布和重尾分布工作负载。系统配置、测试方式和客户机启动方式均与第一类试验相同。仿真结果如图 5 所示。其中 (a) 和 (c) 分别描述均匀分布工作负载和重尾分布工作负载下两类客户的延迟波动,(b) 和 (d) 是根据 (a) 和 (c) 计算的平均延迟比与参考值偏离情况。显然,闭环系统在自适应控制器的作用下,都能较快达到新的平衡,实现比例延迟保证。但是与均匀负载相比,重尾负载由于大小样本差别较大,单位采样周期内大尺寸样本的数目会显著影响平均延迟,因此两类客户的平均延迟波动较大。

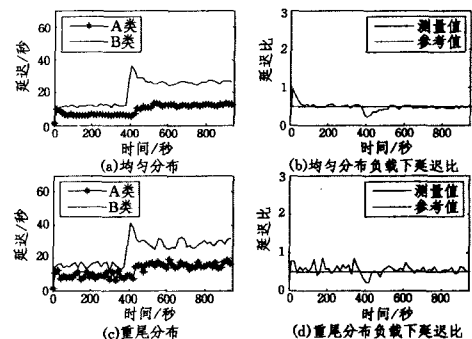


图 5 不同分布负载下的闭环系统性能比较

#### 4.2.3 动态参考输入下的比例延迟区分服务

第三类试验用于评价闭环系统对于动态参考输入变化的适应能力。系统配置、测试方式和客户机启动方式均与第一类试验相同。在整个实验过程中,参考输入按图 6 中虚线变化。由图 6 可知,自适应闭环控制系统能根据参考输入的变化动态调整控制输出,使得系统快速达到新的平衡,实现 PDD 服务。

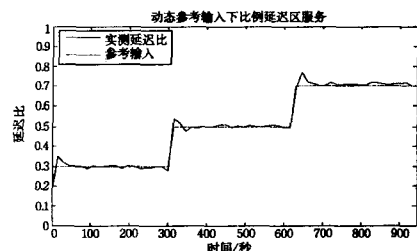


图 6 动态参考输入下的比例延迟保证

结束语 本文在论述 DTN Web 服务和 PDD 服务研究

进展的基础上,设计实现了基于自适应控制的 DTN Web 服务器。通过新增连接管理模块,实现了 DTN Web 服务器上的 PDD 服务。试验结果表明,即使在动态变化的负载条件和不同分布的负载下,闭环系统中的 DTN Web 服务器仍然能较好地为用户提供 PDD 服务。较固定参数的 PI 控制器而言,自适应控制器使得控制指标在动态负载下波动更小。而且,自适应闭环控制系统可适应系统参考输入的变化,实现面向两个服务类别的 PDD 服务。

在进一步的工作中,将着重研究面向多个服务类别的 PDD 服务在 DTN Web 服务器上的实现。同时,考虑到仿真的复杂度,只检验了 DTN 参考实现 DTN2 下的 PDD 服务,在未来的工作中,将结合 DTN 网络的高延时、高中断率等特征开展深入研究。

## 参考文献

[1] 单志广,林闯,肖人毅,等. Web QoS 控制研究综述[J]. 计算机学报,2004(2)

[2] Cerf V, et al. Delay-tolerant Network Architecture [S]. IETF RFC 4838. informational, April 2007

[3] Wood L, Holliday P. Using HTTP for delivery in Delay/Disruption-Tolerant Networks[EB/OL]. Internet-Draft, 2008

[4] Ott J, Kutscher D. Bundling the Web: HTTP over DTN[C]// WNEPT 2006 Workshop on Networking in Public Transport, QShine Conference, 2006

[5] Dovrolis C, Stiliadis D, et al. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling[C]// Proceedings of the ACM SIGCOMM, 1999

[6] Abdelzaher T F, Stankovic J A, Lu Chengyang, et al. Feedback

Performance Control in Software Services [J]. IEEE Control Systems, 2003, 23(3)

[7] Lu Chengyang, Abdelzaher T F, Stankovic J A, et al. A Feedback Control Approach for Guaranteeing Relative Delays in Web Servers[C]// Proceedings of IEEE Real-time Technology and Applications Symposium, Taiwan, June 2001

[8] Wei Jian-bin, Xu Cheng-zhong. Design and Implementation of a Feedback Controller for Slowdown Differentiation on Internet Servers[C]// WWW 2005. Chiba, Japan, 2005

[9] Lu Y, Abdelzaher T F, Saxena A. Design, implementation, and evaluation of differentiated caching services[J]. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(5): 440-452

[10] Harchol-balter M, Schroeder B, Bansal N, et al. Size - based Scheduling to Improve Web Performance[J]. ACM Transactions on Computer Systems, 2003, 21(2), 207-233

[11] The Apache Software Foundation [J/OL]. [http:// www. apache. org](http://www.apache.org)

[12] Peltola L. Enabling DTN-based Web Access : the Server Side [D]. Department of Communications and Networking, Helsinki University of Technology, April 2008

[13] 韩曾. 自适应控制[M]. 北京:清华大学出版社, 1995

[14] 谢新民, 丁锋. 自适应控制系统[M]. 北京:清华大学出版社, 2004

[15] Barford P, Crovella M E. Generating Representative Web Workloads for Network and Server Performance Evaluation [J]. Measurement and Modeling of Computer Systems, 1998: 151-160

[16] Crovella M E, Bestavros A. Self-similarity in World Wide Web Traffic: Evidence and Possible Causes[J]. IEEE/ACM Transactions on Networking, 1997, 5(6): 835-846

(上接第 96 页)

1 所列。

表 1 融合网络中恶意实体的信任变化统计表

信任值=0.5		会话的失败次数						
融合值	实体数	1	3	4	5	6	7	8
0.2	3	0.1632	-0.0150					
0.3	4	0.2632	0.0850	-0.0115				
0.4	5	0.3632	0.1850	0.0885	-0.0094			
0.5	5	0.4632	0.2850	0.1885	0.0906	-0.1068		
0.6	2	0.5632	0.3850	0.2885	0.1906	0.0921	-0.0068	
0.7	1	0.6632	0.4850	0.3885	0.2906	0.1921	0.0932	-0.0069
恶意实体数		0	3	7	12	17	19	20

本文的网络信任域是对称的。当更新后的信任值小于零时,该信任模型认为该实体为恶意实体。由表 1 可知,在异构融合模式中,假设信任度为 0.5,当实体的初始融合值较低(0.2~0.4)时,通过较少的交易行为就可以判断出实体是否存在恶意实体;当实体的初始融合值较高(0.5~0.7)时,需要通过更多的交易行为来判断实体是否为恶意实体。该信任模型对于实体的行为能够迅速准确地反映在信任值上,因此这种判断方法能够保证将恶意实体所带来的损失降到最低。

**结束语** 无线传感器网络和计算机网络融合是异构网络融合的典型代表,网络结构模型是异构网络互联融合的关键。本文在综合考虑计算机网络和无线传感器网络特点的基础上,提出并分析了互联、隧道和异构网络等 3 种融合模型,并重点研究了第 3 种模型的移动性管理和路由协议架构。

下一步计划在异构网络融合结构模型的基础上,重点开展可信路由控制、资源管理与访问控制、信任管理以及行为检测控制等关键技术研究,并进行仿真验证,通过关键技术研究

进一步促进异构网络模型的完善。

## 参考文献

[1] 3GPP TS 23.234 V6.0.0. 3GPP System to Wireless Local Area Network (WLAN) interworking; system description (Release 6) [S]. 2004

[2] 刘侠,蒋铃鸽,何晨. 一种无线异构网络的垂直切换算法[J]. 上海交通大学学报, 2006, 40(5): 742-746

[3] 崔莉,鞠海玲,苗勇,等. 无线传感器网络研究进展[J]. 计算机研究与发展, 2005, 42(1): 163-174

[4] Patwardhan A, Parker J, Iorga M, et al. Secure Routing and Intrusion Detection in Ad Hoc Networks[C]// Proceedings of the 3rd IEEE Int'l Conf. on Pervasive Computing and Communications (PerCom 2005), 2005

[5] Jiang Yuan-xi, Zhao Bao-hua. A Secure Routing Protocol with Malicious Nodes Detecting and Diagnosing Mechanism for Wireless Sensor Networks[C]// IEEE Asia-Pacific Services Computing Conference, 2007

[6] Guo C, Guo Z, Zhang Q, et al. A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks[J]. IEEE Journal on Selected Areas in Communications, 2004, 22: 834-848

[7] Song Qingyang, Jamalipour A. A Network Selection Mechanism for Next Generation Networks[C]// IEEE ICC, 2005: 1418-1422

[8] Jiang Yuan-xi, Zhao Bao-hua. A Secure Routing Protocol with Malicious Nodes Detecting and Diagnosing Mechanism for Wireless Sensor Networks[C]// IEEE Asia-Pacific Services Computing Conference, 2007