# 分组网上多媒体实时业务双时间尺度速率控制算法*)

张玉清[1] 蔡安妮[2] 孙景鳌[2]
(北京航空航天大学软件开发环境国家重点实验室 北京100083)[1]
(北京邮电大学电信工程学院 北京100876)[2]

## A Double Time-Scale Rate Control Algorithm for Multimedia Real-Time Applications in Packet Switched Networks

ZHANG Yu-Qing CAI An-Ni SUN Jing-Ao
(National Laboratory of Software Development Environment of BeiHang University, Beijing 100083)[1]
(School of Telecommunication Engineering, Beijing University of Posts & Telecommunications, Beijing 100876)[2]

**Abstract** By taking into account the self-similarity nature of video traffic, in this paper we propose a receiver-driven double time-scale rate control (DTSRC) algorithm for multimedia real-time applications based on direct bandwidth estimation, rather than on loss rate or delay time of packets. In this algorithm, the data rate is considered not significantly varying in a long time scale, and the actual rate is estimated in a short time scale. If the sending rate falls outside the estimated rate range, the receiver notifies the transmitter to change its rate and the adaptive QoS control is achieved. Simulation results show that by using this algorithm the network bandwidth is utilized sufficiently, and the packet loss rate is quite low.

**Keywords** Self-similarity, QoS, Rate control, Real-time applications

## 1 引言

随着IP网带宽的不断提升,视频会议和视频点播(VOD)等多媒体实时业务日益增多。这些应用具有实时性强、数据量大等特点,因此通常在传输层采用无连接的UDP协议。这些UDP数据流,容易引起网络的拥塞,于是人们从网络层和应用层两个方面开展了对QoS保障问题的研究。在网络层对这一问题的解决方法主要有使用资源预留协议(RSVP)和综合服务(IntServ)或区分服务(DiffServ)等。在网络层进行QoS控制至少有两方面的不足,其一是要求链路上的路由器都能支持该协议或服务;其二是,如果应用建立时网络比较繁忙,通过协商,用户只能获得较低水平上的服务,即使稍后网络变得空闲了,通常用户的服务水平却得不到相应的提升。

在IETF发布RTP(Real-time Transport Protocol)协议[1]之后,人们提出了一些基于应用层的QoS控制策略[2~4]。这些控制算法的基本思路是,在接收端检测丢包率,通过与设定阈值的比较估计网络状况,然后通知发送端调整发送速率。文[5]通过实际网络观测远距离传输视频流的特点,发现在网络出现拥塞时,首先表现出时延的增大,而不是丢包。于是,他们提出了基于RTT(Round Trip Time)的控制算法。该算法是在发送端检测发送数据包的RTT值,通过与理想值作比较,调整发送速率。以上这些算法都没有充分考虑业务本身的传输特点,控制参数的选取也与途经网络的拓扑结构直接相关。

最近几年,人们通过对实际网络的测试发现,业务流具有明显的长相关性[6~9]。因此,我们可以采用能表征长相关性的模型,如自相似过程模型,来描述网络业务。本文在分析视频业务流自相似特性的基础上,提出了针对实时业务的接收端双时间尺度速率控制(DTSR)算法。其基本思想是:接收端直接检测当前实际获得的带宽(速率),并以此作为控制发送端发送速率的依据。在对实际速率进行估值的过程中,必须兼顾准确性与实时性,并尽量减少计算量。因此,我们首先根据业务自相似的特点,划分较大的时间尺度(简称大尺度),认为在这个时间尺度内业务流量是相似的。为了快速估计出大尺度的平均速率,我们又在其中划分出较小的时间尺度(简称小尺度)。依据数理统计理论,我们可以通过若干个小尺度内获得的速率样本,估计出大尺度平均速率的置信区间,并以此作为控制发送速率的依据。仿真实验表明,本算法降低了丢包率,有效保障了QoS。本算法的另一个好处是控制参数与途经网络的拓扑结构无关。本文将以VOD业务为例,具体说明算法的原理和实施。

本文第2节介绍自相似过程及其特点;第3节分析VOD业务流的自相似特性,并提出划分大尺度的方法;第4节说明如何通过在小尺度内获得的速率样本,快速估计出大尺度平均速率的置信区间;第5节给出DTSRC算法,并以一个具有分层编码的MPEG2流为例,说明算法的实现过程;最后是实验结果和结论。

## 2 自相似过程模型及特征

在过去相当一段时间中,人们通常采用短相关模型,例如Markov链或Poisson过程,来描述网络业务流,即当前网络

*)国家自然科学基金资助项目(69972009)。张玉清 博士后,主要研究方向是多媒体通信,IPv6的典型应用,协同工作环境等。蔡安妮 教授,博士生导师,从事多媒体通信、图像处理和识别方面的教学和研究工作。孙景鳌 教授,博士生导师,从事多媒体通信、图像处理和成像技术方面的教学和研究工作。

流量仅有对过去的短时记忆。而近几年来，人们通过对 LAN、WAN、ATM、WWW 等不同网络进行的测试发现，实际网络业务流具有长相关性[6~9]，因此，应该采用能表征长相关性的模型，例如自相似过程模型，来描述网络业务。

**定义1**（长相关过程[10]） 给定随机过程 $X=(X_1,X_2,\cdots)$，如果它的自相关函数 $r$ 是不可和的，即 $\sum_k r(k)=+\infty$，则称 $X$ 为长相关过程。相应地，自相关函数可和的过程，则称为短相关过程。

自相似过程是长相关过程中最简单的模型。

**定义2**（自相似过程[10]） 给定随机过程 $X=(X_1,X_2,\cdots)$，如果它的自相关函数 $r(k)$ 满足下式

$$r(k)=\frac{1}{2}[(k+1)^{2H}-2k^{2H}+(k-1)^{2H}] \tag{1}$$

则称 $X$ 为自相似过程。(1)式中自相似参数 $H$ 又称为 *Hurst* 参数，它是描述自相似特性的唯一参数，其取值范围为(0.5，1)。$H$ 取值越大，表明随机过程的自相似程度越高。

Tuan 等[11]将业务流在时间上划分成等间隔的时间尺度，并对每个尺度内的平均流量进行估计。他们发现，对于长相关业务流，如果某一尺度内的平均流量值较大，则下一个尺度内的平均流量值较大的可能性也较大；反之亦然。这说明，长相关业务流在时间尺度上是可预测的。因此，我们可以根据自相关函数的变化趋势，选择合适的时间尺度，对流量进行估计。

对于不同的业务流，在用自相似过程进行描述时，会具有不同的 $H$ 参数。本文将以 VOD 业务为例，提出时间尺度的划分方法。

## 3 VOD 业务的自相似特点

实时性业务，如视频会议和 VOD 等，具有较大的数据量和严格的同步关系。这些业务在网络中的流量特性，具有明显自相似的特点[12~14]。Bashforth 等[14]曾对 VOD 业务进行仿真测试，发现 VOD 业务流的 Hurst 参数的均值为0.83。将此值带入公式(1)，得到流量的自相关函数 $r(k)$ 与 $k$ 之间的关系，如图1所示。

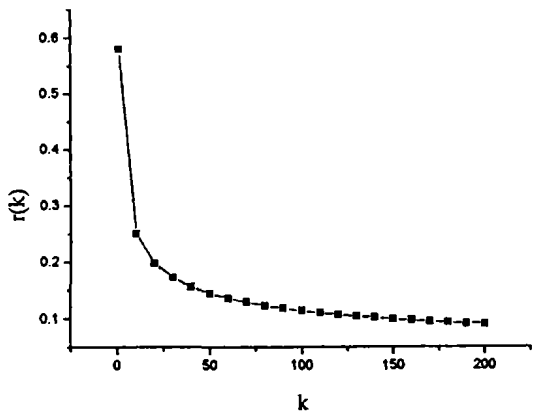


图1 $r(k)$与 $k$ 之间的关系($H=0.83$)

从图1中可以看出，当 $k$ 大于140后，自相关函数 $r(k)$ 将小于0.1，并表现出缓慢下降的特点(当 $k=1000$ 时，$r(k)=0.052314$；当 $k=10000$ 时，$r(k)=0.023912$)。所以，我们认为相邻的140个数据是比较相似的，并选择 $k=140$ 作为对流量进行估计的时间尺度。出于对磁盘效率和 CPU 占用率等方面的考虑，视频服务器通常周期性地发送数据。在实际应用中，我们选取的发送周期为50ms，所以当 $k=140$ 时，对应的时间长度为7s。我们选择7s 作为时间尺度，在接收端对接收数据的平均速率进行估值，并以此作为判断网络是否保障业务 QoS 的依据。

## 4 大尺度内速率估计方法

最简单的计算大尺度内速率的方法是将本尺度内接收到的数据比特数直接除以尺度时间。但这种方法的缺点是，接收端经过7s 才能发现网络状况的变化，即视频服务器只能在7s 之后才获得网络拥塞的信息，降低自己的发送速率，人们通常不能容忍如此长时间的 QoS 下降。因此，有必要研究尺度内快速准确的速率估计方法。在接收端发现接收速率有"很大可能"小于发送端发送速率时，及时通知发送端迅速调整发送速率，以适应网络的状况。

如前所述，在7s 的大尺度内有140个50ms 的小尺度。由自相似特性可知，这些小尺度内收到的数据量也应是比较接近的，所以我们可以采用数理统计的方法，由最初的若干个小尺度内接收速率的样本来估计大尺度的速率。

首先定义几个基本的变量：

$T$：大尺度长度7s；$t_0$：小尺度长度50ms；$t$：对接收速率进行估值的测试时间；$C_i$：第 i 个小尺度内收到的比特数；$b_i=\frac{C_i}{t_0}$：第 $i$ 个小尺度的速率；$N=\frac{T}{t_0}=140$：大尺度内包含小尺度的个数，即总体；$n=\frac{t}{t_0}$：测试过程所占用的小尺度个数；$\mu$：发送速率；$\bar{\mu}$：大尺度内的估计速率；$\bar{b}=\frac{\sum b_i}{n}$：$n$ 个小尺度实测速率的均值；$s^2=\frac{\sum(b_i-\bar{b})^2}{n-1}$：$n$ 个小尺度实测速率的方差。

我们的问题是，如何利用 $N,n,\bar{b}$ 和 $s^2$ 估计 $\bar{\mu}$ 的置信区间，并判断 $\mu$ 是否在这个区间内。

Sreenan 等[15]指出在分组交换的网络中，传输时延通常服从正态分布。因此，尽管视频服务器以 $t_0$ 为周期均匀地发送数据，但由于网络传输时延的影响，接收端在同样的时间周期内接收到的数据量 $C_i$ 是不尽相同的。根据传输时延遵从正态分布的特点，我们可以假定 $b_i$ 近似服从均值为 $\bar{\mu}$ 的正态分布(这一点我们还将在第6节的实验中加以验证)。由于未知总体方差，因此我们构造一个自由度为 $n-1$ 的 t 分布[16]：

$$t=\frac{\bar{\mu}-\bar{b}}{\frac{s}{\sqrt{n}}\sqrt{\frac{N-n}{N-1}}} \tag{2}$$

其中 $\sqrt{\frac{N-n}{N-1}}$ 是因总体数量有限，而引入的调整因子。

我们选择 t 分布的另一个重要原因是 t 分布条件比较宽松，它只要求 $b_i$ 近似服从正态分布即可。根据 t 分布的性质，当自由度 $n-1$ 大于30时，可用正态分布 $N(0,1)$ 来近似，有 $\bar{\mu}\approx\bar{b}$。因此我们设定 $n-1=31$，置信度 $100(1-\alpha)\%$ 为 95%，则置信区间为：

$$(\bar{b}-t_{(\frac{\alpha}{2},n-1)}\cdot\frac{s}{\sqrt{n}}\cdot\sqrt{\frac{N-n}{N-1}},\bar{b}+t_{(\frac{\alpha}{2},n-1)}\cdot\frac{s}{\sqrt{n}}\cdot\sqrt{\frac{N-n}{N-1}}) \tag{3}$$

其中 $N=140,n=32,t_{(0.025,31)}=2.0395$：

$$(\bar{b}-0.3178s,\bar{b}+0.3178s) \tag{4}$$

这种估计方法可以在较短时间内($t=1.6s$)，比较准确地(置信度为95%)估计出大尺度($T=7s$)的平均速率。

## 5 DTSRC 算法

总结前面两节的内容，我们提出 DTSRC 算法：首先将点播过程以7s 为单位，分割成一系列等距离大尺度，大尺度内又含有140个50ms 的小尺度。每个大尺度开始时，接收端通过检测前32个小尺度内的速率来估计本尺度速率的置信区间。如果发送速率 $\mu$ 在置信区间内，则发送端维持现有发送速率；否则，接收端将通知发送端，要求它降低发送速率。

针对 VOD 应用还有两个问题需要解决。其一是存储在视频服务器上的已压缩视频流所能提供的速率变化形式。以 MPEG2节目为例[17]，对于多层的可扩展性(Scalability)编码，一个3Mbps 的流可以由三个子层组合而成：一个基本层(BL)和两个增强层(E1和 E2)。BL 代表体现基本质量的主框架(main profile)速率0.32Mbps；E1是空间域的扩展层，速率为0.83Mbps；E2 是更高一级的信噪比扩展层，速率为1.85Mbps。因此发送端可调整的速率不是连续的而是只有三档：0.32Mbps(BL)、1.15Mbps(BL＋E1)和3Mbps(BL＋E1＋E2)。第二个问题是，在点播过程中，用户总是希望能够接收到较高质量，即较高速率下的节目。这就要求我们提出的速率控制算法，不仅能在网络拥塞时使发送端降低发送速率，还能在网络空闲时使发送端及时提高发送速率。通常的作法是，当网络能支持目前发送速率时，视频服务器将提高一档发送速率。由于相邻层次速率的差值较大(分别为0.83Mbps 和1.85Mbps)，如果网络只支持当前速率的稳定传输，而无力支持较高发送速率，则这种方法会造成发送速率的频繁调整，反而使观众的视觉感受受到影响，因此，我们在控制算法中增加了一个计数器，当接收端发现在非最高速率(BL 或 BL＋E1)下能持续10个大尺度稳定接收数据时，则通知发送端提高一档发送速率，并将 $\mu$ 调整为相应的值。这种作法可以有效地处理好上面提出的问题，因为接收端通过32个小尺度的测试就能及时了解网络是否支持速率的提升。由于不恰当地提升发送速率而引起的 QoS 下降，持续时间是1.6s 左右，与10个大尺度(70s)相比是很小的(2.3%＝1.6/70)，因此我们认为这种提升速率的方法是可行的。

综上所述，我们提出的 DTSRC 算法的流程如下：

1. 大尺度计时开始
2. 大尺度个数计数器 $N$ 加1
3. 获取32个小尺度内的数据 $C_i$，计算出 $b_i$、$\bar{b}$ 和 $s$
4. 计算大尺度平均接收速率的置信区间
5. 如果 $\mu$ 位于置信区间内，则：

·持续至大尺度结束；

·若 $N=10$，且 $\mu$ 不为最大值，则通知发送端提升一档发送速率，且令 $\mu$ 为新的发送速率，$N=0$；

·返回步骤1

否则，

·通知发送端以接近于置信区间的低一档速率发送数据；

·令 $\mu$ 为新的发送速率；

·$N=0$；

·返回步骤1

为了降低算法的计算量和提高计算的实时性，我们记录中间变量，$\sum_n b_i$ 和 $\sum_n b_i^2$，并对 $s^2$ 做如下处理：

$$s^2=\frac{\sum_N (b_i-\bar{b})^2}{n-1}=\frac{1}{n-1}\left(\sum_n b_i^2-n\bar{b}^2\right) \quad (5)$$

这样对 $s^2$ 计算就可分摊到各个小尺度内。由于引入本算法而增加的计算量为：前31个小尺度内增加了两个加法，一个除法，一个平方；第32个尺度内增加了三个加法，两个减法，两个乘法，三个除法，两个平方和一个开平方。这样的计算量，对于目前的微机系统来说，是微不足道的。

## 6 仿真实验及结论

为了考察上面提出的 DTSRC 算法的效果，我们采用如图2所示的仿真模型，并利用仿真软件 OPENT，进行了模拟实验。

图中两个路由器 A 和 B 之间的可用带宽为10Mbps，视频服务器(Video Server)向工作站(Client)发送视频节目，三个工作站 Backgrond1、Backgrond2和 Backgrond3向 Sink 发送具有自相似特性的背景流数据。背景流的参数是 $H=0.75$，生成方法参见文[18]。Client 端通过 DTSRC 算法，控制视频服务器的发送速率。图3显示了 Client 端的接收速率，路由器之间的数据流量(包括背景流和视频数据流)，以及视频数据的丢包率。

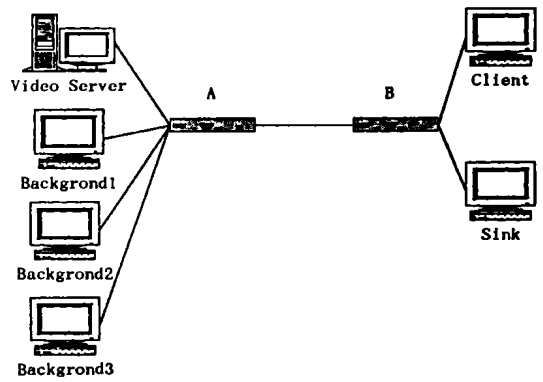


图2 仿真系统拓扑结构

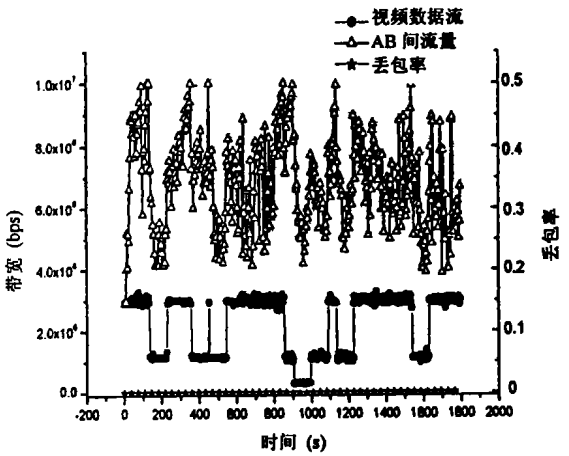


图3 仿真实验结果

从图3中可以看出，视频流的数据量随着网络拥塞状况自动调节，既能在网络拥塞时降低速率，又能在网络空闲时提升速率。由于我们在算法中设置了阈值为10的计数器，因此速率的调整并不十分频繁。例如，150s 时发送速率降至1.15Mbps，在稳定工作70s 后，才通知发送端重新将速率恢复到3Mbps。这么做可能会短时间内未充分利用一部分带宽，但避免了接

收质量的振荡。我们从图3中还发现，丢包率几乎一直保持为零。文[2]中的控制算法是以2%和4%的丢包率作为控制阈值的，只有先发生了丢包，控制机制才起作用。同时，文[2]中为了避免控制环路的振荡，在计算丢包率时采用了一阶低通滤波器，因此实际的瞬时丢包率会比4%高。实验表明，丢包对解码图像的主观质量产生较大的影响[19]。DTSRC 算法能有效地降低丢包率，因此能更好地保证 QoS。文[5]通过实测发现，网络拥塞首先表现为时延的增大，即使发展到无法满足业务实时性要求的情况，也没有出现丢包，所以作者提出了基于 RTT 的控制算法。由于 RTT 参数的选取与业务流途径的网络拓扑结构直接相关，因此这种算法不利于系统的扩展。即使在原有拓扑结构中，如果中间节点或链路的性能参数发生了变化，也必须调整 RTT 参数。我们的算法可以及时检测出严重时延的发生，因为传输时延的增加，直接导致速率的下降。与文[5]相比，DTSRC 算法的一个优点是控制参数与网络拓扑结构无关，具有很好的扩展性。

图3显示，在我们的仿真条件下，3Mbps 附近的样本点明显比其它速率下的样本点多，所以我们选择它们画出如图4的直方图。从图4中可以清楚地看出，接收端获得的速率确实近似服从以约3Mbps 为均值的正态分布，这就验证了第4节中提出的假设 b 近似服从正态分布的合理性。

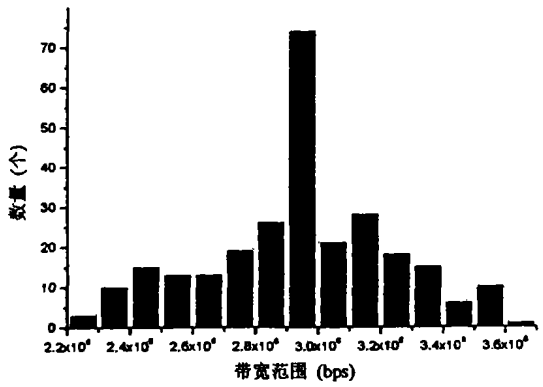


图4　测试速率的直方图

本文以 VOD 业务为例，说明了 DTSRC 算法的原理与实施。实际上，对于其它的实时应用，该方法是完全适用的，只不过由于 Hurst 参数不同，所选取的尺度大小可能不同。在将来的工作中，我们将进一步优化控制参数，使我们提出的速率控制算法不仅能有效保障实时流本身的 QoS，还能与其它业务流特别是 TCP 流公平分享带宽资源。

## 参 考 文 献


1 RFC 1889, RTP: A Transport Protocol for Real-Time Applications[S]. Jan. 1996

2 Busse I, Deffner B, Shulzrinne H. Dynamic QoS Control of Multimedia Applications Based on RTP [J]. Computer Communication, 1996, 19: 49～58

3 El-Marakby R, Hutchison D. Towards Managed Real-time Communications in the Internet Environment [A]. In: Proc. of Fourth IEEE Workshop on the Architecture and Implementation of High Performance Communication System [C], Greece, June 1997

4 Cen S, Pu C, Stachli R, Walple J. A Distributed Real-time MPEG Video Audio Player [A]. In: Proc. 1st workshop on Network and Operating systems Support for Digital Audio and Video [C], Duham, NH, Apr. 1995

5 朱利，周俊辉，郑志军，白跃彬. 基于 RTT 的自适应拥塞控制研究[J]. 计算机学报，2000，7：705～710

6 Willinger W, et al. Self-similarity through High-variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level [J]. IEEE/ACM Transactions on Networking, 1997, 5(1): 71～86

7 Feldmann A, Gilbert A C, Willinger W. Data Networks as Cascades: Investigating the Multifractal Nature of Internet WAN Traffic [J]. ACM Computer Communication Review, 1998, 28: 42～55

8 Moln'ar S, Vid'acs A. On Modeling and Shaping Self-Similar ATM Traffic [A]. 15th Intl. Teletraffic Congress [C], Washington, DC, USA, June 1997

9 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

10 Paxson V. Fast, Approximate Synthesis of Fractional Gaussian Noise for Generating Self-similar Network Traffic [J]. Computer Communication Review, 1997, 27(5): 5～18

11 Tuan T, Park K. Multiple Time Scale Congestion Control for Self-similar Network Traffic [R]. Performance Evaluation, 1999, 36(1): 359～386

12 Beran J, et al. Long-range Dependence in Variable-bit-rate Video Traffic. [J]. IEEE Transactions on Communications, 1995, 43: 1566～1579

13 Garrett M, Willinger W. Analysis, Modeling and Generation of Self-similar VBR Video Traffic [A]. In: Proc. of ACM SIGCOMM '94[C], London, UK, Aug. 1994. 269～280

14 Bashforth B, Williamson C. Statistical Multiplexing of Self-similar Video Streams: Simulation Study and Performance Results [A]. In: Proc. of the Sixth Intl. Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems [C], 1998

15 Sreenan C J, et al. Delay Reduction Techniques for Playout Buffering [J]. IEEE Transactions on Multimedia, 2000, 2(2): 88～100

16 Berenson M L, Levine D M. Statistics for Business and Economics 2nd edition[M]. Prentice Hall, Inc. 1993

17 Campbell A T, Elefteriadis, Aurrecoechea C. End-to-End QoS Management of Adaptive Flows [A]. IEEE Symposium of Multimedia Communications and Video Coding[C], New York, Oct. 1995

18 Sahinoglu Z, Tekinay S. On Multimedia Networks: Self-similar Traffic and Network Performance [J]. IEEE Communications Magazine, Jan. 1999. 48～52

19 ZHANG Yu-Qing, CAI An-Ni, SUN Jing-Ao. QoS Monitoring for Realtime Media Streams with Packet Loss and Jitter[A]. In: Proc. of the Intl. Conf. on Telecommunications 2002[C], pp. 1035～1039, Beijing