低速网络中实时补偿型差额循环调度算法的设计和实现

孙力娟① 李 超① 张登银① 王汝传①②
①(南京邮电学院计算机科学与技术系 南京 210003)
②(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 服务质量(QoS)是目前网络应用研究的一个热点。由于低速链路在当前整个网络中占有相当大的比例，因此研究如何在低速链路上为用户提供具有 QoS 保证的实时业务已经成为一个重要的课题，其中采取何种调度算法则是实现 QoS 保证的关键因素之一。该文根据低速链路的特点，提出了一种适合实时分组转发的公平排队调度算法——实时补偿型差额循环调度(RCDRR)算法，并用 ns2 软件对 RCDRR 算法和 DRR 算法进行了模拟对比。实验及仿真结果表明： RCDRR 调度算法具有公平性好、算法复杂度低、可以降低实时分组在低速链路下的排队时延等特点。

关键词 低速网络，QoS 保证，分组调度算法
中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2006)10-1935-05

Design and Implementation of RCDRR Scheduling Algorithm within Low Speed Networks

Sun Li-juan① Li Chao① Zhang Deng-yin① Wang Ru-chuan①②
①(*Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing* 210003, *China*)
②(*State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing* 210093, *China*)

Abstract QoS is one of the hot points among the network study fields. Currently , because of Low Speed Network (LSN) occupying a very big specific weight among the whole networks, how to provide the real-time services with QoS guarantee for the LSN's customers becomes an important research concern.. In this paper, a scheduling algorithm called Real time Compensation Deficit RoundRobin (RCDRR) according to the LSN's characteristics is proposed to be suitable for realtime packets' transmition, and comparing the RCDRR algorithm with DRR algorithm by using the ns2 software. Experiments and simulation results show that RCDRR scheduling algorithm possesses good fairness,low complexity,and can reduce the queuing delay of realtime packets in LSN.

Key words Low speed networks, QoS guarantee, Cheduling algorithm

1 引言

随着网络技术的发展，诸如IP Phone，视频会议等多媒体业务均要求网络设备能够为这些业务提供QoS(Quality of Service)保证，而采取何种调度算法则是能否实现QoS保证的关键因素之一。从 20 世纪 80 年代末以来，国际上对分组公平调度算法(Packet Fair Queuing，PFQ)进行了大量的研究，提出了许多算法[1-16]，其中严格优先权(Priority Scheduling，PS)，权值公平排队(Weithted Fair Queuing，WFQ)[4]，循环(Round Robin，RR)调度算法[5]和差额循环(Deficit Round Robin，DRR)调度算法[2]是目前较为流行的调度算法。

上述算法在低速网络环境下进行调度分组存在一些缺点。比如，严格优先权调度算法公平性比较差；WFQ调度算法虽然提供了与GPS(Generalized Processor Sharing)相当的特性，但存在计算复杂度为$O(N)$、最坏情况公平指数(WFI，Worst-case Fair Index)与系统中的连接数N成正比，因此造成时延抖动增加、硬件实现比较复杂，设备成本较高等致命的弱点；RR调度算法公平性较好，但是它不能体现不同服务队列的不同带宽需求，尤其当不同服务队列使用不同长度分组时，会产生不公平；DRR调度算法解决了RR调度算法的不足，但DRR算法在低速网络环境下应用也会存在以下一些缺点[6]：

(1)预约带宽越大的连接其一轮服务定额越大，导致经过调度器输出后连接的突发性越大，不利于连接通过后续的节点。

(2)当连接的流量符合漏桶限制时，连接分组的最大时延不仅取决于连接本身的流量特性，还与连接数等其它参数有关，因此 DRR 算法不能很好地支持实时业务。

针对 DRR 算法存在的上述问题，本文结合低速链路这

2005-02-17 收到，2005-08-02 改回
国家自然科学基金(70271050)，江苏省自然科学基金(BK2005146)，江苏省自然科学基金预研项目(BK2004218)，江苏省高技术研究计划(BG2004004，BG2005038)，江苏省计算机信息处理技术重点实验室基金(kjs05001)和江苏省高校自然科学研究计划(04KJB520095)资助课题

一特定的环境，提出一种新的适合低速链路 QoS 的公平调度算法——实时补偿差额循环调度算法 RCDRR(Realtime Compensate DRR)。

RCDRR 算法原则上与 DRR 算法类似，为每一个连接配备一个差额计数器和服务定额寄存器，在一轮中为一个连接发送总长度为服务定额的分组。它与 DRR 算法有以下不同：

(1)RCDRR 能够实时跟踪特定服务队列的长度并以此来调整量子值从而改变预约带宽，满足一些特定业务类型的服务质量要求；

(2)将随机早期丢弃技术加入到RCDRR中，这样RCDRR可以有效地预防在网络发生拥塞时，发送端还会有过多的分组发送过来，从而降低了分组丢弃率；

(3)当实时分组排在一个长非实时分组后时，可以根据实时分组的紧急情况，吊起非实时分组，优先发送实时分组。

2 RCDRR 调度算法描述

2.1 术语

连接 i 表示第 i 个连接，i =0,1,…, N−1；weight_i 为连接 i 的权值，权值越大，表明其优先级越高；$\mathrm{Quantum}_i$ 为连接 i 的一轮服务定额；$\mathrm{Deficitcounter}_i$ 为连接 i 的差额计数器，表示连接 i 在一次服务后剩余的服务量，初始值为 $\mathrm{Quantum}_i$；C_i 为实时分组补偿参数，C_i $\mathrm{Quantum}_i$ 表示高优先级的实时分组在一轮中可以根据队列长度的情况多发出的字节数；B_i 为连接 i 的预约带宽，$B_i = \mathrm{Quantum}_i$；轮(one round)为调度器将所有连接轮循一周，称为一轮。

2.2 定义

定义 1 $\mathrm{FQ}_i = \mathrm{Max}\left(\lim_{t->\infty}\frac{\mathrm{sent}_{i,t}}{\mathrm{sent}_t}\right)$，其中 $\mathrm{sent}_{i,t}$ 为到时刻 t 为止流 i 所发送的字节数，$\mathrm{sent}_{i,t}$ 为所有 n 个流到时刻 t 所发送的字节总数；

定义 2 $\mathrm{IFQ}_i = \frac{\sum_{j=1}^{n} f_j}{f_i}$，$\mathrm{IFQ}_i$表示管理者分配给第 i 个队列应得的带宽比例，其中 f_i 表示流 i 的份额，此处 $f_i = \mathrm{Quantum}_i$；

定义 3 $\mathrm{FairInedx}_i = \mathrm{FQ}_i / \mathrm{IFQ}_i = \frac{\mathrm{FQ}_i \sum_{j=1}^{n} f_j}{f_i}$，$\mathrm{FairInedx}_i$ 为流 i 的公平指数。

2.3 RCDRR 算法描述

(a) 当连接 i 的队列空时，$\mathrm{Deficitcounter}_i$ 为零。

(b) 连接 i 开始获得服务时，该队列的差额计数器初始化为其量子值，即 $\mathrm{Deficitcounter}_i = \mathrm{Quantum}_i$。

(c)连接 i 的队列分组获得服务后，差额计数器减去已服务分组的长度。只要差额计数器大于 0，该队列总能获得服务，否则就轮到下一个连接(连接号为 $k = (i+1) \bmod n$)服务。

(d)每当新一轮开始时，所有非空队列的差额计数器都将加上其量子值。

(e)连接 i 对应一个优先级，当其队列长度的变化越过设定的观察值时，可根据其优先级调整 $\mathrm{Quantum}_i$ 值；也可以根据随机早期丢弃策略丢弃一定的分组，以达到控制时延和时延抖动的目的；同时优先级非常高的实时分组如果排在一个长的非实时分组后面，可以暂停非实时分组的发送，优先发送实时分组，其中，每一个高优先级的实时队列只能在一轮中有一次这样的机会，且发送的分组数不超过 C_i $\mathrm{Quantum}_i$，其中 $0 \le C_i \le 1$。

3 RCDRR 调度算法理论分析

定理 1 采用实时补偿型循环调度算法时，任一队列 i 在第 1 到第 k 轮所调度输出理想字节数与实际字节数差小于 $(\mathrm{Max} - \mathrm{KC}_i\mathrm{Quantum}_i)$。

证明 设$\mathrm{byte}_{i,t}$表示队列 i 在第k轮所发送的字节数，$\mathrm{Sent}_{i,k}$ 表示第 1 到第k轮从队列 i 调度输出的字节数，$\mathrm{Deficitcounter}_{i,k}$ 是队列 i 在第k轮结束时的差额计数器的值。根据实时补偿型循环调度算法，可知

$$\mathrm{bytes}_{i,k} + \mathrm{Deficitcounter}_{i,k} = \mathrm{Quantum}_i(1+C_i) + \mathrm{Deficitcounter}_{i,k+1} \tag{1}$$

由式(1)得

$$\mathrm{bytes}_{i,k} = \mathrm{Quantum}_i(1+C_i) + \mathrm{Deficitcounter}_{i,k-1} - \mathrm{Deficitcounter}_{i,k} \tag{2}$$

因为 $\mathrm{Sent}_{i,k} = \sum_{k=1}^{K}\mathrm{bytes}_{i,k}$，且 $\mathrm{Deficitcounter}_{i,0} = 0$ 所以

$$\mathrm{Sent}_{i,k} = k\mathrm{Quantum}_i(1+C_i) - \mathrm{Deficitcounter}_{i,k} \tag{3}$$

因为 $\mathrm{Deficitcounter}_{i,k} < \mathrm{Max}$，式(3)变为

$$K\mathrm{Quantum}_i - \mathrm{Sent}_{i,k} = \mathrm{Deficitcounter}_{i,k} - k\mathrm{Quantum}_i C_i < \mathrm{Max} - k\mathrm{Quantum}_i C_i$$

证毕

定理 2 采用实时补偿型循环调度算法时，在足够长的观察时间内，任一队列 i 的公平指数 $\mathrm{FairIndex}_i$ 在区间[1/2,2]内。

证明 设有 n 个队列，$\mathrm{Sent}_{i,k}$ 表示第 1 到第 k 轮从队列调度输出的字节数，Sent_k 表示所有队列从第 1 到第 K 轮调度输出的字节数，对于实时补偿型循环调度算法来说，对任一队列 i，第 1 到第 k 轮调度输出的字节数 $\mathrm{Sent}_{i,k}$ 为

$$K\mathrm{Quantum}_i - \mathrm{Max} \le \mathrm{Sent}_{i,k} \le K\mathrm{Quantum}_i(1+C_i) \tag{4}$$

而所有队列在第 1 到第 k 轮调度输出的字节数 Sent_k 为

$$k\sum_{i=1}^{n}\mathrm{Quantum}_i - n\cdot\mathrm{Max} \le \mathrm{Sent}_k \le k\sum_{i=1}^{n}\mathrm{Quantum}_i(1+C_i) \tag{5}$$

由式(4)和式(5)知

$$\frac{K\mathrm{Quantum}_i - \mathrm{Max}}{K\sum_{i=1}^{n}\mathrm{Quantum}_i(1+C_i)} \le \frac{\mathrm{Sent}_{i,k}}{\mathrm{Sent}_k} \le \frac{K\mathrm{Quantum}_i(1+C_i)}{K\sum_{i=1}^{n}\mathrm{Quantum}_i - n\mathrm{Max}} \tag{6}$$

当 $t \to \infty$, $k \to \infty$ 时，由定义 1 知

$$\frac{\text{Quantum}_i}{\sum_{i=1}^{n}\text{Quantum}_i(1+C_i)} \le \text{FQ}_i \le \frac{\text{Quantum}_i(1+C_i)}{\sum_{i=1}^{n}\text{Quantum}_i} \tag{7}$$

所以

$$\frac{\text{Quantum}_i}{\sum_{i=1}^{n}\text{Quantum}_i(1+C_i)} \cdot \frac{\sum_{j=1}^{n} f_j}{f_i} \le \frac{\text{FQ}_i}{\text{IFQ}_i} = \text{FairIndex}_i$$

$$\le \frac{\text{Quantum}_i(1+C_i)}{\sum_{i=1}^{n}\text{Quantum}_i} \cdot \frac{\sum_{j=1}^{n} f_j}{f_i} \tag{8}$$

由定义 2 知 $f_i = \text{Quantum}_i$，代入式(8)

$$\frac{\sum_{i=1}^{n}\text{Quantum}_i}{\sum_{i=1}^{n}\text{Quantum}_i(1+C_i)} \le \text{FairIndex}_i \le 1+C_i \tag{9}$$

因为 $0 \le C_i \le 1$，所以

$$0.5 \le \frac{\sum_{i=1}^{n}\text{Quantum}_i}{\sum_{i=1}^{n}\text{Quantum}_i(1+C_i)} \le \text{FairIndex}_i \le 1+C_i \le 2 \tag{10}$$

因此　$\text{FairIndex}_i \in [0.5,2]$。　　证毕

4　RCDRR 调度算法网络仿真及分析

ns2 由美国加州 Lawrence Berkeley 国家实验室于 1989 年开发成功，它是一种可扩展、易配置和编程的网络仿真工具。ns2 基于事件驱动模型，支持协议广泛，采用了开放的体系结构，用户很容易根据自己的需要开发出新的协议和算法。目前，ns2 被广泛应用于各种网络协议和算法的仿真[17,18]。

(1)RCDRR 网络仿真拓扑结构　在本文中，网络测试拓扑结构如图 1 所示。

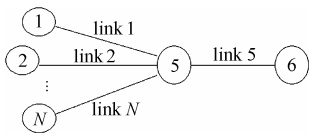

图 1　RCDRR 网络仿真拓扑结构

Fig.1 Topology of RCDRR network simulation

(2)RCDRR 和 DRR 在不同链路容量下实时流时延性能比较　设图 1 中 N=4, 节点 $n1$, $n2$, $n3$, $n4$ 发送数据源为呈指数分布的 on/off 随机数据源，速率分别为 300kbit/s, 400kbit/s, 700kbit/s, 1Mbit/s, 峰值时间 30s, 空闲时间 5s, 分组大小分别为 100byte, 200byte, 300byte 和 500byte，由 $n1$, $n2$, $n3$, $n4$ 至 $n5$ 节点的传输时延均为 10ms，链路 1, 2, 3 ,4 的带宽为 1Mb, 链路 5 的带宽分别设为 0.5Mb, 1Mb, 1.5Mb 和 4Mb。整个仿真时间为 50s，仿真结果如图 2(a)和图 2(b)所示。

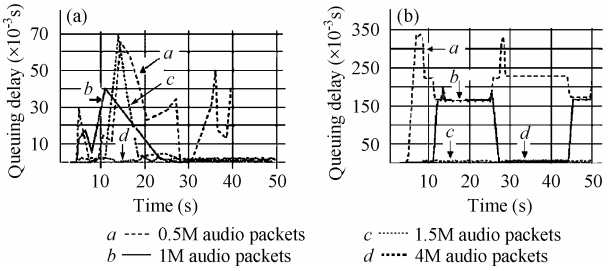

图 2　(a) 采用 RCDRR 调度算法在不同容量出口时音频流分组排队时延(b) 采用 DRR 调度算法在不同容量出口时音频流分组排队时延

Fig.2 (a) The queuing delay of audio packets at the different capacities deploying RCDRR algorithm (b) The queuing delay of audio packets at the different capacities deploying DRR algorithm

仿真结论：从图 2(a)和图 2(b)可见，采用 RCDRR 调度算法音频流分组排队时延比 DRR 调度算法小。

原因分析：在链路空闲时(如输出链路容量为 4M)，RCDRR 调度算法和 DRR 调度算法，均能较好将音频数据转发出去。但采用 DRR 调度算法时，音频流数据要取得发送权，必须严格遵守轮循的原则，一种极端情况是：当音频流队列把调度权交到下一队列手中时，从上游收到新的音频流数据，但必须要等到本轮其它队列调度完后，才能获得发送数据的机会；而 RCDRR 调度算法采用实时补偿发送机制，则可以暂停其它队列的发送，取得调度权，从而能减少分组排队的时延。

当输出链路较为拥挤时，RCDRR 调度算法仍然可以较好地为实时流数据服务，如链路输出容量为 0.5M 时，音频流数据仍然可以得到 200kb/s 速率，而采用 DRR 算法时仅有 150kb/s 速率，前者的传输时延最大为 70ms，后者高达 350ms。

(3) RCDRR 和 DRR 在不同链路容量下对分组丢失率的影响从图 3(a)至图 3(d)可得以下结论：

(a) 当输出链路容量较大时(如 4M 和 1.5M)，采用 DRR 调度算法和采用 RCDRR 调度算法时的实时分组丢失率相当接近；当链路容量较小时(如 0.5M), 采用 RCDRR 调度算法时的实时分组丢失率略大于 DRR 调度算法时的实时分组丢失率；

(b)采用 RCDRR 调度算法，可以降低重要非实时分组的丢失率。

原因分析：DRR 算法对各个流的处理是统一的，它计算为每一个流发送的分组字节数，当网络发生拥塞时，找出已发送分组数最大的流，将其队列分组丢失，并不针对每一个流的特点来处理；RCDRR 调度算法根据各个流的特点采用不同的丢失方式，如：对于音频流数据，允许一定的丢失率来换取分组排队时延的减少；对于一些非常重要的非实时数据，如电子邮件等，可以先将其保存下来，等链路空闲时再发送，也就是用时延来换取较低的丢失率。当网络空闲(输出链路容量为 4M)时，此类分组的丢失率为 0；当网络发生拥塞(输出链路容量为 1.5M，1M 和 0.5M)时，丢失率也比采用

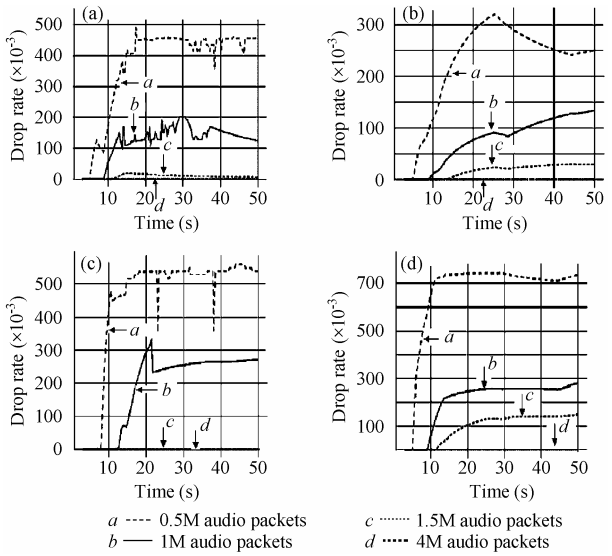

图 3 (a) 采用 RCDRR 调度算法在不同容量出口时音频流分组丢失率 (b) 采用 DRR 调度算法在不同容量出口时音频流分组丢失率 (c)采用 RCDRR 调度算法在不同容量出口时重要非实时分组丢失率 (d) 采用 DRR 调度算法在不同容量出口时重要非实时分组丢失率

Fig.3 (a) The drop rate of audio packets at the different capacities deploying RCDRR algorithm (b) The drop rate of audio packets at the different capacities deploying DRR algorithm (c) The drop rate of none real-time important packets at the different capacities deploying RCDRR algorithm (d) The drop rate of none real-time important packets at the different capacities deploying DRR algorithm

DRR 算法的丢失率低。当然网络发生拥塞时，不重要的非实时分组首先被丢失，这是必须付出的代价。为了减少非实时分组被丢失，可以采取下面的措施：当网络发生拥塞时，路由器可以给非实时分组用户发送一个网络拥塞告知包，此包到达用户端后，可以让终端用户减少分组的发送，以避免进一步加重网络拥塞。

在 RCDRR 算法中，采用 RED 丢失策略来丢失分组，在本文仿真过程中，各个流的最小阈值和最大阈值及其它相关参数初始值设置如表 1：

表 1 实验相关参数

Tab.1 Parameters of the experiments

	音频流	视频流	重要非实时分组流	不重要的非实时分组流
最小阈值	5	10	200	40
最大阈值	30	80	700	100
C_i	0.6	0.6	0	0

5 结束语

根据低速链路的特点，提出了一种适合实时分组转发的公平排队调度算法——实时补偿型差额循环调度算法 RCDRR，并用 ns2 软件对本文提出的排队调度算法进行了仿真，验证该算法的有效性和完备性。理论分析和仿真结果表明，该算法具有下列特点：

(1)继承了 DRR 算法在平均吞吐率上的公平性；

(2)在时延性能上比 DRR 算法有显著改善，而且随着连接数的增大，变化较小；

(3)当网络发生拥塞时，发送端还会有过多的分组发送，RCDRR 算法结合随机早期丢失技术可以有效地降低分组被丢失的概率；

(4)当实时分组排在一个长的非实时分组后时，可以根据实时分组的紧急情况，吊起非实时分组，优先发送实时分组；

(5)实现方式简单，算法复杂度低(为 $O(1)$)；

(6)能够实时跟踪特定服务队列的长度，并以此来调整量子值从而改变预约带宽，满足一些特定业务类型的服务质量。

当然，本文提出的算法和解决方案还有待进一步深入研究。希望在不远的将来，可以推出一个功能更加完善的低速链路 QoS 软件，以使低速终端用户可以更加充分地享受 Internet 所带来的无限乐趣。

参 考 文 献

[1] Zhang H. Service disciplines for guaranteed performance service in packet-switching networks. *Proc. IEEE*, 1995, 83(10): 1374−1396.

[2] Shreedhar M, Varghese G. Efficient fair queuing using deficit round robin. *ACM SIGCOMM Computer Communication Review* 1995, 25(4): 231−242.

[3] Parekh A K, Gallager R G. A generalized processor sharing approach to flow control in integrated services networks. *IEEE Trans. on Networking*, 1993, 1(3): 344−357.

[4] Demers A, Keshav S, Shenker S. Analysis and simulation of a fair queuing algorithm. *ACM SIGCOMM Computer communication Review*, 1989, 19(4): 1−13.

[5] Nagle J B. On packet switches with infinite storage. *IEEE Trans. on Communications*, 1997, COM-35(4): 435−438.

[6] 涂晓东，李乐民. 一类基于调度表的公平轮循调度算法. 电子学报, 2001, 29(9): 1290−1293.

[7] 郭传雄，郑少仁. 对 Linux 操作系统中 TCP/IP 网络协议的 IP 层排队分析. 计算机学报, 2001, 24 (8): 860−865.

[8] 高强，董立岩. GPRS 中的分组调度算法. 吉林工业大学自然科学学报, 2000, 30(4): 29−33.

[9] Floyd S, Jacobson V. Sharing and resource management models for packet networks. *IEEE/ACM Trans. on Networking*, 1995, 3(4): 365−386.

[10] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Trans. on Networkings*, 1993, 1(4): 397−413.

[11] Stiliadis D, Varm A. General methodology for designing efficient traffic scheduling and shaping algorithms. Proc. IEEE INFOCOM'97, Kobe, Japan, 1997: 326−335.

[12] Stiliadis D, Varm A. Efficient fair queueing algorithms for packet-switched networks. *IEEE/ACM Trans. on Networking*, 1998, 6(2): 175−185.

[13] Zhang Hui, Ferrari D. Rate-controlled static priority queueing. Proc. IEEE INFOCOM'93, San Francisco, CA, 1993: 227−236.

[14] Floyd S, Jacobson V. Link-sharing and resource management model for packet networks. *IEEE/ACM Trans. on Networking*, 1995, 3(4): 265−386.

[15] Goyal P. Generalized guaranteed rate scheduling algorithms: A framework. *IEEE/ACM Trans. on Networking*, 1997, 5(4): 561−571.

[16] Zhang Hui, Keshav S. Comparison of rate-based service disciplines. Proc. ACM SIGCOMM'91, Zurich, 1991: 113−121.

[17] Hanle C, Hofmann M. Performance comparision of reliable

[18] multicast protocols osing the network simulator NS-2[EB/OL]. http://www.isi.edu/ns/, 2003-06-28.

[19] Bresau L, Estrin D, *et al*.. Advances in network simulation. *IEEE Computer*, 2000, 33(5): 59−67.

孙力娟：女, 1963 年生, 博士生，研究方向为计算机网络、计算机软件在通信中应用等.

李　超：男，1975 年生, 硕士，助教，研究方向为计算机网络等.

张登银：男，1964 年生，副教授，研究方向为计算机网络技术等.

王汝传：男，1943 年生，教授, 博士生导师，主要研究方向为计算机软件、计算机网络、信息安全、移动代理和虚拟现实技术等.