AIMD组播拥塞控制在卫星IP网络中的性能分析及改进*)

刘功亮[1,3] 顾学迈[1] 康文静[2,3] 郭 庆[1]

(哈尔滨工业大学通信技术研究所 哈尔滨 150001)[1]

(哈尔滨工业大学超精密光电仪器工程研究所 哈尔滨 150001)[2] (哈尔滨工业大学(威海) 威海 264209)[3]

摘 要 基于AIMD的组播拥塞控制算法由于采用了与TCP类似的拥塞控制策略,可以实现TCP友好性,因而在组播传输协议中得到了广泛应用。为了分析卫星网络长延时、高误码率特性对AIMD组播拥塞控制算法的影响,本文采用一种基于马尔可夫随机过程的理论模型,把拥塞发现时刻的拥塞窗口值作为马尔可夫链的状态;根据该理论模型,推导了系统吞吐量与卫星网络各种参数的关系式,从而分析了基于AIMD的组播拥塞控制算法在卫星网络中的性能。在此基础上,提出了采用接收者分组的方法来提高系统吞吐性能的改进方案,并对不同信道条件下的最优分组数量以及带来的吞吐量增益进行了研究。数学仿真结果表明,在高误码率、长延时的卫星网络中,采用最优分组可以显著提高组播系统的平均吞吐量。

关键词 下一代互联网,卫星网络,组播拥塞控制,加增乘减,马尔可夫随机过程

Performance Analysis and Improvement of AIMD-based Multicast Congestion Control in Satellite IP Networks

LIU Gong-Liang[1,3] GU Xue-Mai[1] KANG Wen-Jing[2,3] GUO Qing[1]

(Communications Research Center, Harbin Institute of Technology, Harbin 150001)[1]

(Institute of Ultra-precision Optoelectronic Instrument Engineering, Harbin Institute of Technology, Harbin 150001)[2]

(Harbin Institute of Technology (Weihai), Weihai 264209)[3]

Abstract Due to the TCP-like congestion control policy, AIMD-based multicast congestion control algorithms achieve TCP-friendly and are widely used in multicast transport protocols. In order to reveal the influence of long link delays and high link errors in satellite IP networks on AIMD-based multicast congestion control algorithms, a theoretical model based on Markov stochastic processes is studied, in which the congestion windows at congestion detection moment compose the states of the Markov chains. The throughput performance of AIMD-based multicast congestion control algorithms is analyzed with this theoretical model. Simulation results show that, in case that a large number of receivers are involved, the method of dividing the receivers into some groups and maintaining a different multicast session to each of the groups can achieve high throughput.

Keywords Next generation Internet, Satellite networks, Multicast congestion control, AIMD, Markov stochastic processes

从当前技术发展的趋势来看,下一代互联网(Next Generation Internet)将是一个天地一体化的综合网络。由于地面网络在建设成本、覆盖范围和网络配置灵活性等方面的局限性,作为下一代互联网天基部分的卫星互联网开始受到越来越广泛的关注。与地面网络相比,卫星网络具有覆盖面积广、信道广播性等优点,因而在IP组播应用中具有得天独厚的优势。

IP组播[1]是一种"尽力而为"类型的服务,不提供速率控制,组播流量会"侵略性"地耗光所有的网络资源,导致网络拥塞。为IP组播设计合适的拥塞控制算法是一个需要迫切解决的问题。组播拥塞控制算法的一个重要指标就是要有良好的TCP友好性[2],也就是组播流量要与TCP流量公平地共享网络带宽。由于采用了类似TCP的拥塞控制策略,基于AIMD的组播拥塞控制算法可以方便地实现TCP友好性,因此在目前已有的组播方案中得到了广泛应用[3~6]。然而,如果应用到卫星IP组播环境中,卫星信道的长延时、高误码率等特点,会对基于AIMD的组播拥塞控制算法带来一系列问题,影响系统的吞吐性能。

本文的主要工作,就是建立了一个基于Markov随机过程[7]的理论模型,把拥塞发现时刻的拥塞窗口值作为Markov链的状态,并基于此模型推导了系统平均吞吐量与信道误码率和往返时间等网络参数的关系式;通过模型分析和公式推导,深入研究了卫星网络环境的高误码率、长传播延时等因素对基于AIMD的组播拥塞控制算法吞吐性能的影响;针对卫星环境中AIMD组播拥塞控制算法性能下降的原因,给出一种提高卫星组播系统吞吐性能的改进措施,通过把接收者分成若干规模相近的组播组,降低了链路丢包导致拥塞窗口减小的概率,从而提高了系统吞吐量;本文对不同信道误码率条件下取得最优吞吐量所需的分组数量进行了深入的分析,并得出了一些有意义的结论。

本文在第1部分首先给出卫星组播系统结构;第2部分介绍了AIMD组播拥塞控制算法的基本原理和特点,详细描述了基于Markov随机过程的理论模型,并推导了吞吐量与各种网络参数的关系式;第3部分通过数学仿真分析了AIMD组播拥塞控制算法在卫星IP网络中的性能,并给出了改进方法;最后总结全文。

*)国家自然科学基金(60532030);国家发改委CNGI大规模路由和组播技术的研究与试验项目(CNGI-04-13-2T)。**刘功亮** 博士研究生;**顾学迈** 教授、博士生导师。

1 卫星组播系统结构

本文考虑的卫星组播系统结构如图1所示。地面有线网络中的组播源(Sender)经过一个卫星网关(Gateway)向卫星发送业务数据,在星上进行数据包的复制,并通过星上交换,将数据包组播到各个接收者(Receiver)。接收者通过卫星链路向组播源发送反馈数据包(ACK或NAK),组播源根据反馈数据包中的信息,来判断业务数据包的接收情况,从而相应地增大或者减小拥塞窗口。

设卫星前向链路(如图1中的 $l_0, l_1, l_2, \cdots, l_n$)的信道速率都为2.048Mbps,而地面链路信道速率为10Mbps,因此系统瓶颈在上行卫星网关,这种假设也与实际情况相符。为了简化分析,本文不考虑反馈数据包的丢失。

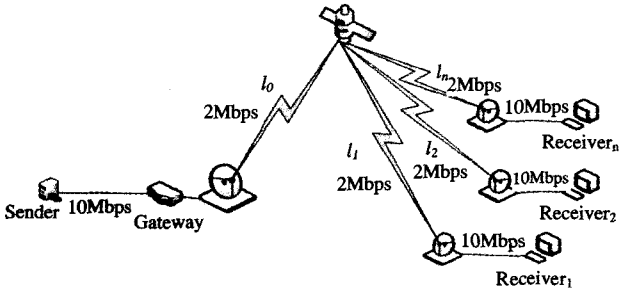

图1 卫星组播系统结构示意图

2 理论分析模型

2.1 加增乘减(AIMD)拥塞控制算法原理

AIMD算法的基本思想是:当没有拥塞发生时,每个RTT将拥塞窗口增加 α;而当拥塞发现时刻,立即将拥塞窗口减小 β 倍。这个过程可以表示为如下的数学表达式:

加性增加:$W_{t+RTT}=W_t+\alpha, \alpha>0$ (1)

乘性减少:$W_{t+\delta t}=(1-\beta)W_t, 0<\beta<1$ (2)

通常,用AIMD(α_0, β_0)来表示 $\alpha=\alpha_0, \beta=\beta_0$ 的AIMD算法。由于TCP采用AIMD(1,0.5)算法,因此在后面的分析中,考虑AIMD(1,0.5)的组播拥塞控制算法,这样可以方便地实现组播业务流和TCP业务流的带宽公平性。

2.2 Markov随机过程模型

首先给出两个定义:拥塞发现时刻和拥塞发现周期。

定义1(拥塞发现时刻) 用 $I_R(\tau)$ 表示时间 τ 后,从信源到接收者R的链路上第一次探测到拥塞的时刻;用 ϕ 表示所有接收者的集合。对于任意非负整数 m,定义第 m 个拥塞发现时刻 t_m 如下式:

$$t_m=\begin{cases}0, & m=0\\ \min_{R\in\phi}\{I_R(0)\}, & m=1\\ \min_{R\in\phi}\{I_R(t_{m-1}+RTT)\}, & m>1\end{cases} \quad (3)$$

定义2(拥塞发现周期) 对 $\forall m\geqslant 1$,将第 $m-1$ 个拥塞发现时刻 t_{m-1} 到第 m 个拥塞发现时刻 t_m 的时间间隔定义为第 m 个拥塞发现周期,用 $\Delta t(m)$ 表示,且 $\Delta t(m)=t_m-t_{m-1}$

由于 t_{m+1} 时刻的窗口值 W_{m+1} 只与 t_m 时刻的窗口值 W_m 有关,而与"过去"无关,所以拥塞发现时刻的系统可以看作一个Markov链,拥塞发生时刻的窗口值就是该Markov链的状态。那么,该Markov链的状态数等于所有可取的窗口值。窗口的最大值 W_{max} 由接收者缓冲区尺寸和卫星上行链路信道速率决定,即

$$W_{max}=\min([\frac{B\cdot RTT}{P_{size}}],[\frac{Q_{rcv}}{P_{size}}])$$

这里 B 为卫星前向链路的信道速率,RTT 为卫星信道往返时间,Q_{rcv} 为接收者缓冲区尺寸,P_{size} 为数据包尺寸,$[x]$ 表示对 x 取整。

设Markov链的状态转移矩阵为 Q,则

$$Q=\begin{bmatrix}p_{11} & p_{12} & \cdots & p_{1N}\\ p_{21} & p_{22} & \cdots & p_{2N}\\ \vdots & \vdots & \vdots & \vdots\\ p_{N1} & p_{N2} & \cdots & p_{NN}\end{bmatrix} \quad (4)$$

其中 N 为Markov链的状态数,显然 $N=W_{max}$。状态转移矩阵的元素 p_{ij} 表示一步转移概率,即

$$p_{ij}=P\{W_m=j|W_{m-1}=i\}, 1\leqslant i,j\leqslant N \quad (5)$$

设卫星网络中数据包的差错率为 PER,接收者的数量为 n,则

$$p_{ij}=(1-PER)^{n\cdot\sum_{l=\lceil i/2\rceil}^{j-1}l}\cdot(1-(1-PER)^{j\cdot n}) \quad (6)$$

而 PER 与信道误码率(BER)之间存在着如下关系:

$$PER=1-(1-BER)^d \quad (7)$$

式中 d 表示的是数据包的长度,单位是比特(bits)。

各状态的极限概率 $p=\{p_1, p_2, \cdots, p_N\}$ 为下面线性方程组的非零解:

$$\begin{cases}p_j=\sum_{i=1}^{N}p_ip_{ij} & (j=1,2,\cdots,N)\\ \sum_{i=1}^{N}p_i=1\end{cases} \quad (8)$$

2.3 吞吐量分析

根据上面建立的分析模型,AIMD组播拥塞控制算法的平均吞吐量可以用下式进行计算:

$$T=\lim_{t\to\infty}\frac{N_{pkt}(t)}{t}=\frac{\lim_{t\to\infty}N_{pkt}(t)}{\lim_{t\to\infty}N_{period}(t)} \quad (9)$$

上式中,T 是平均吞吐量,$N_{pkt}(t)$ 表示时间 t 内发送的数据包数量,$N_{period}(t)$ 表示时间 t 内包含的拥塞发现周期的数量。

那么,式(9)的分子 $\lim\frac{N_{pkt}(t)}{N_{period}(t)}$ 为平均每个拥塞发现周期成功传输的数据包数量,用 $E(X)$ 表示;分母 $\lim\frac{t}{N_{period}(t)}$ 为一个拥塞发现周期的平均时长,用 $E(\Delta t)$ 表示。

令 $X(i,j)$ 表示 $W_{m-1}=i, W_m=j$ 时第 m 个拥塞发现周期内成功传输的数据包的数量根据 i 和 j 的不同,$X(i,j)$ 有 N^2 种可能的取值,用 $\{x_s\}$ 表示,$s=1,2,\cdots,N^2$。那么平均每个拥塞发现周期成功传输的数据包数量为:

$$E(X)=\sum_{s=1}^{N^2}x_sp_s=\sum_{i=1}^{N}\sum_{j=1}^{N}p_iX(i,j)p_{i,j} \quad (10)$$

考虑与TCP拥塞控制相同的AIMD(1,0.5)算法,在不发生拥塞的情况下,每个 RTT 将 $cwnd$ 增加1;而在发生拥塞时,立刻将 $cwnd$ 变为原来的一半。所以

$$X(i,j)=[\frac{i}{2}]+([\frac{i}{2}]+1)+\cdots+j \quad (11)$$

令 $\Delta t(i,j)$ 表示 $W_{m-1}=i, W_m=j$ 时第 m 个拥塞发现周期的时间长度。同理,平均每个拥塞发现周期的时间长度为

$$E(\Delta t)=\sum_{s=1}^{N^2}\Delta t_sp_s=\sum_{i=1}^{N}\sum_{j=1}^{N}p_i\Delta t(i,j)p_{i,j} \quad (12)$$

其中

$$\Delta t(i,j)=\sum_{[i/2]}^{j} RTT(j-[i/2]+1) \tag{13}$$

综合(6)～(13)式可以看出，BER、RTT 等参数影响着AIMD组播拥塞控制算法的吞吐性能。那么，在高误码率、长延时的卫星网络中，吞吐性能将会发生什么样的变化呢？在下一节里，将通过数学仿真研究AIMD组播拥塞控制算法在卫星网络中的吞吐性能，并给出改进措施。

3 数学仿真及性能改进措施

为了分析卫星网络特点对基于AIMD的组播拥塞控制算法的影响，本节根据前面的理论模型，利用强大的数值计算工具MATLAB对吞吐性能进行了数学仿真。假设卫星前向链路的信道速率为2.048Mbps，接收者缓冲区尺寸为128kbytes，数据包尺寸为1kbytes。

3.1 卫星信道特性对吞吐量性能影响

图2给出了不同误码率下的吞吐性能曲线。从图中可以看出，随着信道误码率的升高(即信道条件变坏)，系统吞吐量显著下降。这是由于在高误码率的卫星环境中，基于AIMD的拥塞控制策略把信道差错引起的丢包当成拥塞的标志，从而减小拥塞窗口，造成吞吐量的下降。而卫星链路的长延时特性会进一步加重这个影响，如图3所示。

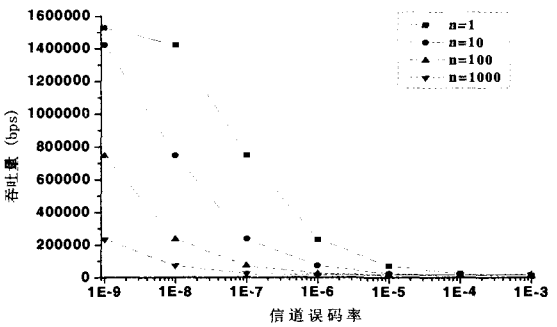

图2 信道误码率对吞吐性能的影响

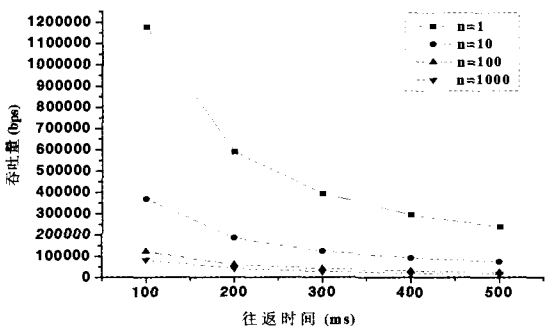

图3 链路往返时间对吞吐量的影响($BER=10^{-6}$)

3.2 利用接收者分组的方法提高系统吞吐量

图4给出了不同误码率情况下吞吐量随接收者数量变化的性能曲线。从图中可以看出，随着接收者数量增加，系统吞吐量呈明显的下降趋势。这是由于在本文考虑的组播系统中，只要有一个接收者没有正确收到数据包，就要减小拥塞窗口。当接收者数量很大时，数据包被所有接收者正确接收到的概率就要显著减小，这样就会造成系统吞吐量的下降。

针对这个问题，考虑在接收者数量较大的时候，将所有 R 个接收者分为 k 组，对每个组维持一个组播会话。分组的原则是尽量分成规模相同的若干个组播组。具体的分组方法是：若 k 整除 R，则每组拥有 R/k 个接收者；若 k 不整除 R，则 k 个组播组中，有 $k-(R-k\cdot[R/k])$ 个组拥有 $[R/k]$ 个用户，另外 $R-k\cdot[R/k]$ 个组拥有 $[R/k]+1$ 个用户。

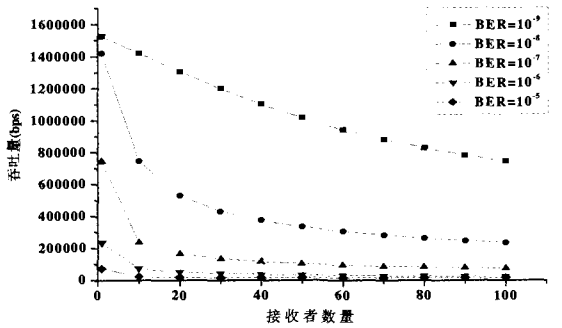

图4 接收者数量对吞吐性能的影响

分组的数量 k 会对系统吞吐量产生影响。这个影响是双方面的：一方面，随着分组数量的增加，发送者就需要建立更多的组播会话，增加前向链路的负担；而另一方面，每个组播会话涉及的接收者减少，丢包的概率就会下降。由于正反两方面的影响，所以在将接收者分组的时候，需要综合考虑，只有 k 值合适，才可以取得最大的吞吐量。

图5给出了接收者数量为100时，在不同信道误码率条件下系统吞吐量随着 k 值的变化而变化的曲线。从图中可以看出，吞吐量随着分组数量的增加呈现先增后减的趋势；当分组数量达到某个合适的值的时候，吞吐量达到峰值。观察图5还可以发现，在不同误码率条件下，最大吞吐量对应的 k 值(下面称为最优分组数)也不同。随着误码率的升高，最优分组数也随之增大。也就是说，在高误码率的卫星信道环境中，需要把一个组播会话的所有接收者分为更多的组，才能最大程度地提高系统平均吞吐量。这是由于信道误码率越高，链路丢包造成的拥塞窗口减小就越严重；将接收者分为多个组，每个组的成员数减少，就可以降低链路丢包发生的概率，从而提高吞吐量。

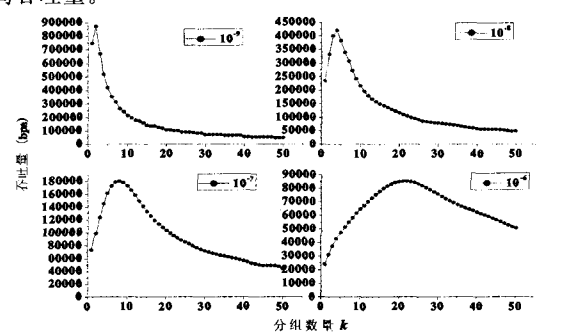

图5 不同误码率下分组数量对吞吐性能的影响($R=100$)

表1给出了不同误码率条件下的最优分组数，以及采用最优分组方案可实现的吞吐量增益。该表中的最优分组数，对应了图5中曲线顶点的横坐标值。吞吐量增益 A_{div} 表示采用最优分组方案可以带来的吞吐量的提高程度，可以通过下式计算。

$$A_{div}=\frac{T_{peak}-T_1}{T_1} \tag{14}$$

上式中，T_{peak} 为最优分组数对应的系统平均吞吐量，也就是图5中的吞吐量峰值；T_1 为不采用分组时的系统平均吞吐量，也就是图5中 $k=1$ 时对应的吞吐量。

表1 不同误码率下的最优分组数量和吞吐量增益

误码率	最优分组数	吞吐量增益 A_{div}(%)
0	1	0
10^{-9}	2	1.7
10^{-8}	4	79.1
10^{-7}	8	144.9
10^{-6}	22	253.4
10^{-5}	34	287.2

从表1可以看出，在信道条件较好时(误码率优于10^{-9})，分组方案带来的吞吐量增益很小；然而随着信道条件变差，吞吐量增益呈明显的上升趋势，在卫星信道的典型误码率10^{-6}情况下，采用最优分组方案带来的吞吐量增益可以达到253.4%，这说明本文提出的方法在卫星IP组播网络中具有非常显著的作用。

结束语 针对基于AIMD的组播拥塞控制算法，本文提出了一种基于Markov随机过程的理论分析方法并建立了数学模型。根据此模型，推导了系统平均吞吐量计算公式，给出了AIMD组播拥塞控制算法应用于卫星IP网络时的性能曲线。研究表明，高误码率、长延时的卫星网络会严重影响AIMD组播拥塞控制算法的性能，随着接收者数量的增大，吞吐性能会进一步恶化。

在以上分析的基础上，提出了通过接收者分组来提高系统吞吐量的方法。数学仿真结果表明，如果把接收者分为数量适当的组播组，对每个组建立一个组播会话，就可以最大程度的提高系统吞吐量。分组的数量是提高性能的关键。通过仿真和分析可以看出，最优分组数随着信道误码率的上升有明显的上升趋势，这就意味着在高误码率的卫星环境中，需要把一个组播会话的接收者分为更多的组，才能最大程度的提高系统平均吞吐量。在误码率低于10^{-9}的信道条件下，分组方案带来的吞吐量增益很小，但是在高误码率的卫星环境中，采用最优分组可以将系统平均吞吐量提高2.5倍以上。

本文的研究工作为基于AIMD的组播拥塞控制算法应用于卫星IP网络提供了一种有效的解决思路。本文仅考虑了所有接收者信道条件相同的情况，对于接收者信道条件不同的异构网络，最优分组方案将有所不同，这个问题将在以后的工作中进一步研究。

参考文献

1 Deering S. Multicast routing in a datagram internetwork[D]. Stanford University, 1991
2 Wang H A, Schwartz M. Achieving bounded fairness for multicast and TCP traffic in the internet[A]. In: Proc of ACM SIGCOMM 1998[C]. Vancouver, Canada, 1998. 81～92
3 Rhee I, Balaguru N, Rouskas G. MTCP: Scalable TCP-like congestion control for reliable multicast[J]. Computer Networks, 2002, 38(5): 553～575
4 Byers J W, Kwon G, Luby M, et al. Fine-grained Layered multicast with STAIR[J]. IEEE/ACM Transactions on Networking, 2006, 14(1): 81～93
5 Golestani S J, Sabnani K K. Fundamental observations on multicast congestion control in the Internet[A]. In: Proc of IEEE INFOCOM 1999. New York, USA, 1999. 990～1000
6 Rizzo L. PGMCC: a TCP-friendly single-rate multicast congestion control schemes[A]. In: Proc of ACM SIGCOMM 2000[C]. Stockholm, Sweden, 2000. 17～28
7 Cover T, Thomas J. Elements of Information Theory[M]. Wiley, 1991

(上接第36页)

制中所存在的缓存空间过小、替换频率过大、查询效率低下等问题，并讨论了其核心算法及其相关支撑技术。针对"尽力而为"缓存机制在某些情况下可能会以偶尔访问的对象替换查询频率较高的对象从而使得命中率下降的不足，SCOCM利用缓存对象间的语义信息解决此类问题。SCOCM根据语义信息对缓存内已有的对象进行分类，根据节点的分类信息主动地选择对象放置缓存，让每个缓存中缓存的对象之间尽可能地保持语义联系，使得缓存中尽可能缓存下一次访问命中概率大的对象。SCOCM机制的主要工作包括：1) 一种主动选择缓存内对象放置的方法以及使用这种方法的缓存管理机制，克服了以往算法中不考虑语义信息被动缓存对象的缺陷，并对网络的整体缓存能力实现了进一步地提升，在保障缓存命中率的前提下，大大减少了缓存的替换次数，减少存储器内的存储开销。2) 使得节点具有了语义信息，有助于动态地调整形成一个高兴趣度聚集的覆盖网络。3) 提出了一个缓存对象预取算法，使得SCOCM具有一定的学习功能，变被动缓存为主动，该机制可在网络负载较小时完成对存取对象的预取，合理分配网络开销。在今后的研究中，我们将针对语义信息在超节点P2P网络中的应用进一步展开深入研究，从而提升整个网络的存取效率并致力于解决其在各类P2P网络中应用时所碰到的实际问题。

参考文献

1 Yang B, Garcia-Molina H. Designing a Super-Peer Network. In: Proc of the 19th Intl Conf on Data Engineering, 2003
2 Zhuang Z, Liu Y, Xiao L. Dynamic Layer Management in Super-peer Architecture. In: Proc of the 2004 Intl Conf. on Parallel Processing (ICPP' 04)
3 Fiorano Software Inc. Super-Peer Architecture for Distributed Computing. White Paper, 2003
4 KaZaA website. http://www.kazaa.com
5 Garbacki P, Epema D H J, van Steen M. A Two-Level Semantic Caching Scheme for Super-Peer Networks. wcw, 2005
6 Mohammad, Salimullah, Raunak. A Survey of Cooperative Caching: [Technical report]. December 1999
7 Malpani R, Lorch J, Berger D. Making world wide web caching servers cooperate. In: Fourth International World Wide Web Conference, December 1995
8 Bowman C, Danzig P B, Hardy D R, et al. The harvest information discovery and access system. In: Second International World Wide Web Conference, October 1994
9 The Squid Project. 2004. http://www.squid-cache.org/
10 Zheng Wang. Cachemesh: A distributed cache system for world wide web. In: 2nd NLANR Web Caching Workshop, June 1997
11 Zhang L, Floyd S, Jacobson V. Adaptice Web Caching. In: Proceedings of the NLANR Web Cache Workshop, 1997
12 Garcés-Erice E W, Biersack P A, Felber K W, et al. Urvog-Keller Hierarchical Peer-to-peer Systems. In: Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par 2003)
13 Laoutaris N, Syntila S, Stavrakakis I. Meta Algorithms for Hierarchical Web Caches. IEEE IPCCC 2004, Phoenix, Arizona, April 2004
14 Korupolu M R, Dahlin M. Coordinated placement and replacement for large-scale distributed caches. In: Proc of the IEEE Workshop on Internet Applications. San Jose, CA: IEEE Computer Society Press, 1999. 62～71
15 Che H, Tung Y, Wang Z, Hierarchical web caching systems: modeling, design and experimental results. IEEE J Selected Areas Commun, 2002, 20 (7)
16 Daconta M C, Obrst L J, Smith K T. The semantic Web : a guide to the future of XML. In: Web services, and knowledge management . Wiley Pub, c2003
17 LIU G Z. Semantic vector space model : Implementation and evaluation. Journal of the American Society for Information Science, 1997, 48(5): 395～417
18 Wang J Z, Bhulawala V. Design and Implementation of A P2P Cooperative Proxy Cache System. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005), September 2005
19 http://peersim.sourceforge.net/