Gnutella 网络的连接管理*

庄 雷[1,2+], 潘春建[3], 郭永强[1], 王从银[1]

[1](郑州大学 信息工程学院,河南 郑州 450052)

[2](国家数字交换系统工程技术研究中心,河南 郑州 450000)

[3](中国科学院 声学研究所 网络与数字信号处理技术研究中心,北京 100080)

Connection Management Based on Gnutella Network

ZHUANG Lei[1,2+], PAN Chun-Jian[3], GUO Yong-Qiang[1], WANG Cong-Yin[1]

[1](College of Information Engineering, Zhengzhou University, Zhengzhou 450052, China)

[2](National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450000, China)

[3](Engineering Research Center of Digital Signal Processing, Institute of Acoustics, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-371-7763077, E-mail: ielzhuang@zzu.edu.cn, http://www.zzu.edu.cn

Received 2003-10-20; Accepted 2004-01-14

Zhuang L, Pan CJ, Guo YQ, Wang CY. Connection management based on Gnutella network. *Journal of Software*, 2005,16(1):158−164. http://www.jos.org.cn/1000-9825/16/158.htm

Abstract: Gnutella is a fully decentralized and unstructured peer-to-peer network. It uses the message broadcasting mechanism of flooding. However, while bringing Gnutella network the characters of high degree of robustness and dynamic, this broadcasting mechanism makes the network give redundant messages that increase exponentially. On basis of resolving Gnutella network message broadcasting mechanism, the paper points out the necessity and feasibility of Gnutella network losing contact, and then bring forward the means which can compartmentalize Gnutella network message's PRI according to the transmitting bandwidth, the time, and the resources which are consumed by servents dealing with all kinds of messages. F-Measure is introduced to connection management, which is usually used to evaluate the performance of searching engine. The paper provides a discarding connection management algorithm, which discards the redundant connection by computation and ensures the maximal attainability of message simultaneously. Finally, the arithmetic example and discussion are given.

Key words: Gnutella; servent; F-Measure; DCMA (discarding connection management algorithm)

摘 要: Gnutella 是完全分布式、无结构的对等网络.它采用洪泛式的消息广播机制,使网络具有高鲁棒性和高动

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA111141 (国家高技术研究发展计划(863))

作者简介: 庄雷(1963－),女,山东日照人,教授,主要研究领域为计算机网络,自动机理论;潘春建(1979－),男,博士生,主要研究领域为宽带网多媒体接入技术;郭永强(1977－),男,硕士,主要研究领域为分布式计算;王从银(1971－),男,硕士生,主要研究领域为计算机网络.

© 中国科学院软件研究所 http://www.jos.org.cn

态性的同时,也使网络产生了呈指数级增长的冗余消息.在解析 Gnutella 网络消息广播机制的基础上,指出了 Gnutella 网络丢弃某些连接的必要性和可行性,提出了根据传输带宽和机器处理各种消息时所耗费的时间和资源,来划分Gnutella网络中消息的优先级.把评价搜索引擎性能的F-Measure参数引入连接管理中,在保障消息可达率的同时,通过计算丢弃某些冗余连接.该解决方案由丢弃连接管理算法(discarding connection management algorithm,简称 DCMA)实现,还给出了算法实例和对算法的讨论.

关键词: Gnutella;servent;F-Measure;DCMA(discarding connection management algorithm)

目前,互联网上的多数服务都是基于客户机/服务器的模式,它以一些大的网站为中心,不同地域的客户端通过连接服务器进行信息的浏览和下载.随着网络规模的扩大,服务器已难以满足日益增加的客户端的需求.如果每台机器都能为其他计算机提供服务,而不像客户机/服务器那样仅依赖中央服务器提供服务,这将极大地改善网络的性能,提高服务质量,减轻服务器的压力,减少网络拥塞,方便、灵活地实现机器间的通信及资源共享.这里的每台机器既是客户机又是服务器,称为 servent,即对等网络的思想(peer-to-peer,简称 P2P).

P2P 让人们通过互联网直接交互,真正地消除了中间商.互联网上最基本的协议 TCP/IP 并没有客户机和服务器的概念,所有设备都是通信中平等的一端.所以,P2P 使互联网技术返璞归真,重返“非中心化”,把权力交还给用户.P2P 引导网络计算模式从集中式向分布式偏移,也就是说,网络应用的核心从中央服务器向网络边缘的终端设备扩散:服务器到服务器、服务器到 PC 机、PC 机到 PC 机、PC 机到手机,…,所有网络节点上的设备都可以建立 P2P 对话.它使人们在 Internet 上的共享行为被提到了一个更高的层次,以更主动、更深刻的方式参与到网络中去.当前,P2P 主要应用在大范围的资源共享和搜索领域[1],它能够很好地解决网络上 4 大类型的应用:对等计算、协同工作、搜索引擎、文件交换.

Gnutella[2]作为一种典型的 P2P 网络通信协议,由 AOL 的 Nullsoft 部门开发.它具有完全分布式的特点,能够有效地消除单点瓶颈,使网络具有更好的鲁棒性.与此同时,该协议也产生了呈指数级增长的冗余消息,效率低下,只能在小规模的网络上使用,很难成为应用的主流.

本文首先解析了 Gnutella 网络的消息广播机制,接着分析了 Gnutella 网络丢弃冗余连接的必要性和可行性,提出了 4 种消息的优先级划分,将评价搜索引擎性能的 F-Measure 参数引入连接管理中来,在此基础上,给出了丢弃连接管理算法(discarding connection management algorithm,简称 DCMA),并给出了算法实例和对算法的讨论.

1 Gnutella 网络消息广播机制解析

所有利用 Gnutella[3]协议运行的计算机组成的网络称为 Gnutella 网络.网络中的每个节点是一个 servent.Gnutella 网络的消息传播机制如下:

(1) 向前广播入消息

Gnutella 网络采用公平简单的方法向周围主机节点广播消息.Gnutella 网络中的每个节点都管理着若干个连接.对于每个 servent,除了知道本地连接外,并不知道网络的拓扑[4],网络中的每个 servent 将接收到的消息发送到除消息来源节点之外的所有与其直接相连的 servent 节点上.

(2) 丢弃已见过的入消息

一个 servent 从其他 servent 接收到消息时,它仅仅检查这个消息是否是先前已经处理过的.如果是,它就丢弃该消息,不再向前广播;否则,就将这个消息广播到和它直接相连的所有 servent 上.为了确定消息是否是先前处理过的,每个消息都需要有一个唯一的标识 ID 号,每个节点需要维护一个近来接收到的消息队列.如果这个新近接收到的消息 ID 在其维护的消息队列中未出现,就对其进行广播,否则将其丢弃.

(3) 丢弃 TTL=1 的入消息

传输的消息每经过一个 servent,其消息的 TTL 值减少 1,hop counts 的值增加 1,当发现该消息的 TTL 值等于 1 时,则停止向前广播该消息.

© 中国科学院软件研究所 http://www.jos.org.cn

(4) 原路返回应答消息

应答消息根据节点记录的历史信息沿原路径返回.Gnutella 网络消息原路返回机制[5]:当目标 servent 对 request 消息产生出 response 响应后,就将该 response 消息传递给和它直接相连的 servent,并且该 response 消息和请求的 request 消息具有一致的消息 ID 号,接收到该 response 消息的 servent 将该消息的 ID 号和自己保存的消息列表中的 ID 号进行比较,并判断该消息的类型,如果该消息的类型是 response 而且消息列表中又具有该消息 ID 号时,才进行传递;否则,丢弃,直至到达请求的原 servent 节点为止.

(5) 节点自己产生的消息向所有连接广播

节点自身产生的消息,将向其所有邻接的 servent 进行广播.

2 Gnutella 网络丢弃连接的必要性与可行性

理想情况下,消息在整个 Gnutella 网络中广播,到达网络的所有节点.但是,随着网络规模的扩大,网络节点的增多,网络中的冗余消息数目将会是海量的,对于用有限带宽来处理这些无限增长的消息量,过载是一种必然的结果.

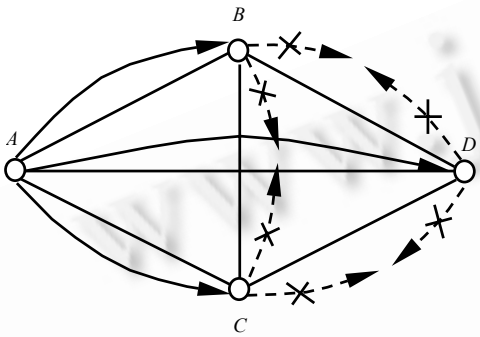

Fig.1 Connected graph of four nodes

图 1 具有 4 个连通节点的示意图

下面我们用图来示例冗余消息的产生过程,如图 1 所示,假设节点 A,B,C,D 建立连接(图中实线所示),在某一时刻,节点 A 接收到网络中的一个广播消息,根据 Gnutella 协议,它将该消息向前传播到和其直接相连的节点 B,C 和 D 中(图中实线箭头所示),节点 B,C 和 D 分别检查自己的消息列表,发现该消息没有在其消息列表中出现过,就继续向前广播,其中包括由 B 到 C、由 C 到 B、由 B 到 D、由 D 到 B、由 C 到 D、由 D 到 C 的传播路径,结果在这些连接中传播的消息(图中虚线箭头所示)都将被丢弃.

随着节点间连接的增加,网络中消息冗余量的增加会更明显.我们将其推广到一般情况,在具有 n 个连通节点的 Gnutella 网络中,一条消息最多可以产生 $P_2^{n-1}-(n-1)$ 个冗余的消息.这些冗余的消息将极大地耗费机器的处理时间,吞噬网络带宽[6],这也是 Gnutella 网络不能适用于大规模应用的弊病所在.

通过分析这些冗余消息的产生过程可以知道:Gnutella 网络的消息广播机制导致了冗余消息的产生.由于 Gnutella 网络的完全分布性和高动态性的特点,使得网络中的某些连接成为冗余,因此,我们可以通过丢弃这些相对冗余的连接,有效地节约带宽和机器的处理时间.同时,只要我们消息的 TTL 值设置合理,即使丢弃掉一些连接,我们仍有机会寻找到更快捷的到达该节点的路径.

3 Gnutella 网络消息优先级的划分

要确定丢弃哪一条连接,就得对丢弃连接的影响进行评估.丢弃连接意味着要丢弃该连接上的消息,Gnutella 网络中的消息可以划分为 4 种类型:自身产生的 request 消息;接收到的 request 消息;自身产生的 response 消息;接收到的 response 消息.为了对这 4 类消息进行优先级划分,我们分析丢弃某类消息对网络的影响:

(1) response 消息的优先级高于 request 消息

对一个 request 消息 R,一个节点 D 最多只会给出一个应答 response 消息.如果丢弃掉一个 response 消息,发起该查询的节点 S 在本次查询中就不可能得到该应答,与其这样,还不如不转发相应的 request 消息.一个 servent 节点丢弃 request 消息,还有可能通过其他路径到达目标 servent 节点,同时,还可以提高机器的处理性能,提高网络传输带宽的利用率.因此,接收到的 response 消息应该具有最高的消息优先级.

(2) 接收到消息的优先级高于自身产生的消息

消息到达本节点时已经使用了一些网络资源,而自身产生的消息尚未使用网络资源,所以说丢弃接收到的

© 中国科学院软件研究所 http://www.jos.org.cn

消息比丢弃自身产生的消息对网络的影响要大.

(3) 自身产生的 response 消息优先级高于接收到的 request 消息

节点产生出 response 消息表明已经耗费了一定的网络资源和机器对此消息的处理时间,而接收到 request 消息仅仅表明耗费了一部分网络资源(忽略节点对消息的底层处理时间),故前者的优先级更高.

根据以上分析,4 种消息的优先级由高到低的顺序是:

接收到的 response 消息→自身产生的 response 消息→接收到的 request 消息→自身产生的 request 消息.

4 连接管理的评估方法及实例

上一节我们对消息进行了优先级的划分,目的就是为了评价丢弃连接对网络的影响.本节我们把评价搜索引擎性能的 F-Measure 参数引入到连接管理中来,并根据消息的优先级划分对该参数进行改进,得到连接管理参数,最后给出一个例子以说明计算连接管理参数的方法和过程.

4.1 F-Measure

F-Measure[7]参数由 Van Rijsbergen 于 1979 年提出,当前,它主要用于信息的检索,是搜索引擎性能评价的重要标准.下面我们来介绍 F-Measure 参数在搜索引擎中的具体含义.

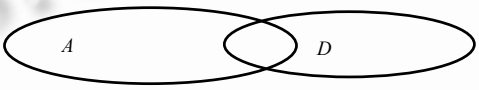

Fig.2 Actual results and desired resources

图 2 返回结果集和期望结果集图示

如图 2 所示,A 代表一次搜索返回的所有结果集,D 代表所有期望得到的结果集.查准率 p(precision)是指已查出的有效结果占所有搜索到结果的百分比.查全率 r(recall)是指已查出的有效结果占所有期望得到的结果集的百分比.分别定义如下:

$$p = p(D/A) = \frac{P(A \cap D)}{P(A)} \tag{1}$$

$$r = p(A/D) = \frac{P(A \cap D)}{P(D)} \tag{2}$$

这两个参数反映了搜索引擎的检索效果,按照 F-Measure 参数对其属性的定义,我们可以将这两个参数结合成一种简单的标量尺度 f,定义如下:

$$f = \frac{2 \cdot p \cdot r}{p + r} = \frac{2 \cdot p(A/D) \cdot p(D/A)}{p(A/D) + p(D/A)} = \frac{2 \cdot P(A \cap D)}{P(A) + P(D)} \tag{3}$$

4.2 连接管理参数

在搜索引擎中,每个检索的关键字被封装成一个请求的查询信息,然后由搜索引擎的服务器对此请求进行处理,获取请求资源的机器信息.在 Gnutella 网络中,request 消息也是一种请求消息,节点产生 request 消息后,按照其消息处理机制进行广播,然后由目标节点对此消息进行响应.由于 Gnutella 网络弱化了服务器功能,每个 servent 节点既具有客户机的功能,又具有服务器的功能,因此,Gnutella 网络中的 request 消息和搜索引擎的查询信息在本质上具有一致性.

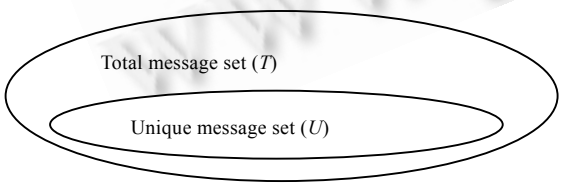

Fig.3 Message set in Gnutella

图 3 Gnutella 网络消息集合

我们收集某节点自身产生的和接收到的 request 消息的历史信息,并做出统计.用 U 表示唯一消息集合(即节点收到的某消息 ID 仅有一次).用 T 表示所有消息集合(即消息 ID 相同的消息重复计算).集合 U 和 T 的关系如图 3 所示.

将 F-Measure 参数应用到基于 Gnutella 协议的 P2P 网络连接管理上,唯一消息集合 U 相当于图 2 中的 D,所有消息集合 T 相当于图 2 中的 A.根据集合 U 和 T 的关系,我们可以得到如下的改进公式:

© 中国科学院软件研究所 http://www.jos.org.cn

$$p' = p(U/T) = \frac{P(U \cap T)}{P(T)} \tag{4}$$

$$r' = p(T/U) = \frac{P(U \cap T)}{P(U)} \tag{5}$$

$$f' = \frac{2 \cdot p' \cdot r'}{p' + r'} = \frac{2P(U)}{P(U) + P(T)} \tag{6}$$

公式(6)可算出每个节点的 f'值,参数 f'的值越大,表明该节点连接上唯一消息的可达率越高,冗余度越小,它是冗余率和可达率之间的折衷,f'值越大,网络节点的综合效能越好.

设节点共有 n 条连接,我们对该节点所有连接的历史信息进行统计,按照公式(6)计算得到参数 f'(Overall).对该节点除去连接 i 之外其他连接的历史信息进行统计,按照公式(6)计算得到参数 f'(Discard-i).若 f'(Discard-j)=Max(f'(Discard-1),f'(Discard-2),…,f'(Discard-n)),则丢弃连接 j 后该节点的效能最高.我们将参数 f'称为连接管理参数,因为在进行连接管理时,参数 f'用于确定哪条连接将被丢弃.

4.3 计算连接管理参数的方法和过程

Table 1 Collecting messages

表 1 采集消息信息表

Message ID	Type	Connection		
		A	*B*	*C*
000001	α	Y	Y	Y
000002	β			Y
000003	β	Y	Y	Y
000004	β	Y		
000005	β	Y	Y	
000006	β		Y	Y
000007	β	Y	Y	Y
000008	β	Y	Y	
000009	β	Y	Y	Y
000010	β		Y	Y
000011	β			Y
000012	α	Y	Y	Y
000013	β	Y	Y	
000014	β			Y
000015	β	Y	Y	Y
000016	α	Y	Y	Y
000017	β	Y	Y	
000018	β		Y	Y

下面的实例用于说明计算连接管理参数的方法和过程.假设某个节点维护 3 个连接,它在某时刻得到如表 1 所示的历史记录,其中α表示自身产生的 request 消息,β表示接收其他节点的 request 消息.

首先,统计每条连接上的消息,$|A|$=12,$|B|$=14,$|C|$=13,$|A \cap B|$=11,$|A \cap C|$=7,$|B \cap C|$=10,$|A \cap B \cap C|$=7,得到数据的 Venn 图,如图 4 所示.

由于接收到的 request 消息数目和自身发出的 request 消息数目决定了节点主机在网络中的角色,因此,这两种消息应该具有不同的权值,其权值应该根据节点主机的机器性能和连接所具有的带宽来决定.假定该节点确定自身产生的 request 消息的权值为 0.3,接收其他节点的 request 消息的权值为 0.7.将上述统计结果进行加权计算,$|A|$=7.2,$|B|$=8.6,$|C|$=7.9,$|A \cap B|$=6.5,$|A \cap C|$=3.7,$|B \cap C|$=5.8,$|A \cap B \cap C|$=3.7,得到的结果如图 5 所示.

然后,我们可计算连接管理参数如下:

(1) 计算所有连接的参数

$P(U)=|A|+|B|+|C|-|A \cap B|-|A \cap C|-|B \cap C|+|A \cap B \cap C|$

$=3.7+7.2+8.6+7.9-6.5-3.7-5.8+3.7=11.4.$

$P(T)=|A|+|B|+|C|=7.2+8.6+7.9=23.7.$

f'(Overall)=2*$P(U)$/($P(U)$+$P(T)$

=2*11.4/(11.4+23.7)

=22.8/35.1=65%.

(2) 计算丢弃连接 A 后的参数

$P(U)=|B|+|C|-|B \cap C|=8.6+7.9-5.8=10.7,$

$P(T)=|B|+|C|=8.6+7.9=16.5,$

f'(Discard-A)=2*$P(U)$/($P(U)$+$P(T)$=2*10.7/(10.7+16.5)=21.4/27.2=79%.

(3) 计算丢弃连接 B 后的参数

$P(U)=|A|+|C|-|A \cap C|=|7.2+7.9-3.7=11.4,$

$P(T)=|A|+|C|=7.2+7.9=15.1,$

f'(Discard-B)=2*$P(U)$/($P(U)$+$P(T)$=2*11.4/(11.4+15.1)=22.8/26.5=86%.

(4) 计算丢弃连接 C 后的参数

© 中国科学院软件研究所 http://www.jos.org.cn

$P(U)=|A|+|B|-|A\cap B|=7.2+8.6-6.5=9.3$,

$P(T)=|A|+|B|=7.2+8.6=15.8$,

$f'(\text{Discard-}C)=2*P(U)/(P(U)+P(T)=2*9.3/(9.3+15.8)=18.6/25.1=74\%$.

比较上述各计算值,Max(f'(Discard-A), f'(Discard-B), f'(Discard-C))=f'(Discard-B),连接 B 被确定为丢弃的连接.由f'(Discard-B)–f'(overall)=21%可知,连接 B 被丢弃后,节点的f'参数值提高 21%.

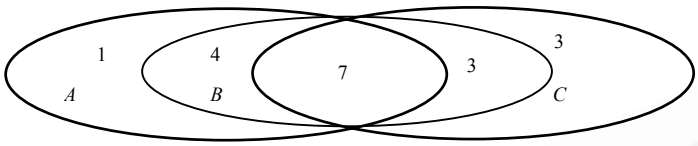

Fig.4 Results of statistics message in table 1

图 4 表 1 历史信息的统计结果

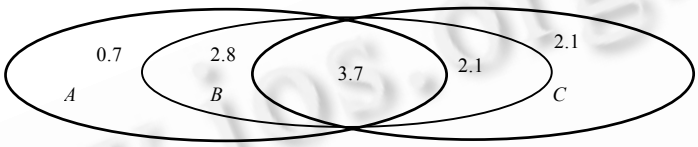

Fig.5 Results of calculate with power

图 5 加权计算结果

5 丢弃连接管理算法

DCMA 算法的基本思想是:节点通过在给定的时间段内,对从 n 条连接上接收到的 request 消息和自身发出 request 消息分别进行记录,其处理方法不是像 Gnutella 协议那样,将接收到的重复消息丢弃,而是根据消息连接的不同来源分别进行记录,再进行统计.由于接收到的 request 消息数目和自身发出的 request 消息数目决定了节点主机在网络中的角色,因此,这两种消息应该具有不同的权值,其权值应该根据节点主机的机器性能和连接所具有的带宽来决定.分别计算该节点在现有连接和去掉一条连接后的 f'值.根据 f'值决定是否丢弃某条连接,以及应该丢弃哪条连接.

现将算法的核心步骤描述如下(假设节点要维护 n 条连接):

Step 1. 如果已经建立起 n 条连接,转 Step 2;否则,等待,直至建立起 n 条连接.

Step 2. 开始记录这 n 条连接的历史信息,并设定定时器;定时器到时以后转 Step 3 进行计算.

Step 3. 节点将自身产生的 request 消息和接收到的 request 消息按照不同的连接来源分别进行统计.

Step 4. 将不同连接上接收到的消息,按照设定的权值进行加权求和.

Step 5. 按照改进后的 F-Measure 参数计算公式,分别计算该节点在现有连接的f'(overall)和去掉连接 i(i=1,…,n)后的 f'(Discard-i).

Step 6. 设f'(Discard-j)=Max(f'(Discard-1),…,f'(Discard-n)),如果f'(Discard-j)–f'(overall)>ε,丢弃连接j;否则,不丢弃任何连接.

Step 7. 回到 Step 1.

Gnutella 网络具有高度动态性.这种动态性表现在网络中各个节点的加入和离开完全是自主决定的.随着节点的加入和离开,网络连接也会动态地建立和断开.动态性是 Gnutella 网络的固有特性.但是,这种动态性会导致网络资源的浪费.本文的算法是通过丢弃连接来减少冗余消息的数量.但是,如果丢弃连接过于频繁,会大大增加网络的动态性,增加网络资源的浪费.因此,我们使用定时器限制丢弃连接的频率.

一条连接从建立到断开所经历的时间,称为该连接的寿命.节点自加入 Gnutella 网络时,开始记录它的各条连接的寿命,并计算连接的平均寿命.随着新的连接的建立和已有连接的断开,求取的平均连接寿命不断变化.所以平均连接寿命是一个动态的变量,它在一定程度上反映了网络的动态性.当节点开始记录历史信息时,根据

© 中国科学院软件研究所 http://www.jos.org.cn

平均连接寿命来设置定时器.当定时器到时,开始进行统计和计算.因此,定时器的设置也是动态的,它与网络连接的动态性基本同步.

算法涉及到的另一个问题是:如果 f'(Discard-1),f'(Discard-2),…,f'(Discard-n)与 f'(Overall)相差都不大,甚至都为 0,说明丢弃连接并不能显著提高效能,甚至还得为之付出一定的代价.因此,我们引入参数ε,它相当于一个阈值,用于最后确定是否丢弃某条连接.

设 max_diff=max(f'(Discard-i)–f'(overall)).节点加入网络后,首先把ε设置为一个指定值ε_0(比如 10%).然后,第 n(n>0)次使用参数ε时,把它设置为

$$\varepsilon_n = (\varepsilon_0 + \sum_{i=1}^{n} \max\_diff_i)/(n+1) \tag{7}$$

上式可以简化为

$$\varepsilon_n = (\varepsilon_0 + \sum_{i=1}^{n-1} \max\_diff_i + \max\_diff_n)/(n+1) = (n \cdot \varepsilon_{n-1} + \max\_diff_n)/(n+1) \tag{8}$$

可见,参数ε的设定也是动态的,能够动态跟踪节点的实际运行状况.

由第 2 节我们知道,Gnutella 网络中的消息数目是呈指数级增长的,随着网络中连接数量的增多,TTL 值的增大,request 消息可以到达的节点数目会很大,网络中的带宽很大程度上被冗余的消息所吞噬.芝加哥大学的研究小组对 Gnutella 网络进行研究后发现:Gnutella 网络的基础架构[8]不符合互联网的拓扑学原理,网络节点间的连接是造成其效率低下的主要原因.如果网络中的每个节点都使用 DCMA 算法,当节点的连接足够多时,每个节点丢弃一条冗余消息量大的连接的话,将极大地改善整个网络的性能.

6 总结与展望

Gnutella 网络作为一种典型的 P2P 网络,可以有效地消除单点瓶颈,但同时也产生了呈指数级增长的冗余消息,严重影响了该网络的进一步发展.本文在解析其消息广播机制的基础上,提出采用丢弃连接的方法来减少网络中的冗余消息.首先,对采用这种方法的必要性和可行性进行了分析,进而对消息的优先级进行了划分,并给出了基于此思想的 DCMA 算法.

与此同时,Gnutella 网络还有许多值得深入研究之处,如对 request 与 response 之间的关系的量化,具有较高优先级的 response 类消息的处理;物理连接的非对称,输入带宽和输出带宽的不平衡;缓冲时间的设定等问题,本课题组也作了深入探讨.

References:

[1] Abdul-Rahman A, Hailes S. A distributed trust model. In: Proc. of the 1997 New Security Paradigms Workshop. ACM, 1997. 48−60. http://bikmrdc.lm.fju.edu.tw/files/ADistributedTrustModel.pdf

[2] Bordignon F, Tolosa G. Gnutella: Distributed system for information storage and searching model description. 2002. http://sise.ttu.ee/it/vorgutarkvara/wav4101/paper_final_gnutella_english.pdf

[3] Gnutella Protocol Specification. 2002. http://www.gnutella.co.uk/library/pdf/gnutella_protocol_0.4.pdf

[4] Kleinberg J, Kumar R, Raghavan P, Rajagopalan S, Tomkins A. The Web as a graph: Measurements, models, and methods. In: Proc. of the 5th Annual Int'l Conf. on Computing and Combinatorics. 1999,1627:1−7. http://www.cs.cornell.edu/home/kleinber/web-graph.ps

[5] Barabôsi A, Albert R. Emergence of scaling in random networks. Science, 1999,286:509−512.

[6] Jovanovic M. Modeling large-scale peer-to-peer networks and case study of gnutella [MS. Thesis]. Cincinnati, Ohio: University of Cincinnati, 2001. http://www.ececs.uc.edu/~mjovanov/thesis/thesis.ps

[7] Ishioka T. Evaluation of criteria for information retrieval. 2002. http://www.rd.dnc.ac.jp/~tunenori/doc/ishiokat_criteria.ps

[8] Ripeanu M, Foster I, Iamnitchi A. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. 2002. http://people.cs.uchicago.edu/~matei/PAPERS/ic.pdf

© 中国科学院软件研究所 http://www.jos.org.cn