

自私感知的数据驱动覆盖网络

王 淼^{1,2}, 张玉军¹, 李国杰¹

(1. 中国科学院计算技术研究所, 北京 100190; 2. 中国科学院研究生院, 北京 100039)

摘 要: 为了解决数据驱动覆盖网络(DONet)中节点自私性问题,提出了一种自私感知的 DONet,简称 SA-DONet 机制. SA-DONet 赋予每个节点一个利他值,记录着该节点对其他节点的转发贡献;基于节点利他值设计的数据段请求和发送算法,保证利他值高的节点被优先服务. SA-DONet 具有以下特点:首先,使用分布式方法来检测自私节点并动态的调整数据段请求和发送策略;其次,自私的评价(利他值)基于节点的历史数据,无需发送额外的探测和测量包;最后,SA-DONet 保持着和 DONet 相当的计算复杂度. 仿真结果表明,和 DONet 相比,即使存在一定比例的自私节点,SA-DONet 机制也能提高应用层组播的总体性能,并且具有较低的控制负荷.

关键词: 利他值; 应用层组播; 数据驱动; 分布式算法; 自私性

中图分类号: TP393.09 **文献标识码:** A **文章编号:** 0372-2112 (2010) 03-0534-07

Selfishness-Aware Data-Driven Overlay Network

WANG Miao^{1,2}, ZHANG Yu-jun¹, LI Guo-jie¹

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China;

2. Graduate School of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: To address the selfishness issue in Data-driven overlay network (DONet), a selfishness-aware DONet (SA-DONet) is proposed in this paper. SA-DONet allows each node associative with an altruism value for its contributions to peers. Based on the altruism value, segment requesting and sending algorithms are designed to ensure the more altruistic nodes will have more chances to be served. The primary characteristic of our mechanism lies in three aspects. Firstly, SA-DONet can discover the selfish nodes in a decentralized manner and adjust the segment sending and requesting strategy dynamically. Secondly, selfish assessment (altruism value) comes from the node's history and doesn't require any extra probe and measuring packets. Lastly, our algorithms remain comparable computing complexity to DONet. Simulation results show that compared with DONet, even with a significant portion of nodes being selfish, SA-DONet can improve the streaming quality of global multicast session with low control overhead.

Key words: altruism value; application-layer multicast; data-driven; distributed algorithm; selfishness

1 引言

现有的应用层组播协议可以分为基于树状的和基于网状的两类. 在基于树状的应用层组播协议中, 无论是网优先(mesh first)^[1]、树优先(tree first)^[2]还是隐式法(implicit approach)^[3], 都需要构造一棵组播树, 组播分组沿着树进行传输. 然而, 一般来说, 主机作为转发节点很不稳定, 它们可能频繁地加入、离开或者失效, 因此, 应用层组播树可能经常改变, 这会影响组播应用的性能, 尤其是流媒体类应用的性能, 因为这类应用对连续性要求很高.

在基于网状的应用层组播协议中, 节点之间不再有

固定的父/子关系, 数据路径呈现网状, 因此能更好的容忍节点动态变化. 典型的基于网状的应用层组播协议包括基于 gossip 协议的^[4,5]和数据驱动覆盖网络(DONet)^[6]等. 在基于 gossip 的组播协议中, 每个节点随机地选择一组邻居节点转发消息, 直到消息被传递给所有的节点为止. 在 DONet 中, 每个节点选择一部分邻居节点作为其伙伴节点, 并根据网络状态进行动态调整. 数据被分割成大小相同的数据段, 节点用一个缓存映射表(BM)来表示缓存中存在那些数据段. 节点通过不断地和它的伙伴节点交换 BM, 寻找数据段的潜在提供者, 并决定从那个伙伴节点获取那些数据段. 因为在 DONet 中, 伙伴关系和数据传输方向都不固定, 而且和基于

gossip 协议的相比, DONet 中数据采用请求-应答的方式传递, 不会有大量数据消耗网络带宽, 因此, 它尤其适合实时的流媒体应用。

然而, 应用层组播中的节点不仅是不稳定的, 而且是具有独立利益和策略的。自私是独立节点的天性。例如, 在 Gnutella 网络中, 66% 的节点根本不共享任何文件, 73% 的节点共享的文件数目少于 10; 而且, 1% 的共享节点提供着 47% 的文件请求, 25% 的共享节点提供着 98% 的文件请求^[7]。Hughes 等人进一步分析了 Gnutella 网络, 指出这种搭便车行为极大地影响了网络的性能^[8]。类似的, 在 DONet 中, 节点可能有意(为了减少数据转发负担)或者无意(因为节点处理能力或者带宽限制)的停止或者延迟转发某些数据段, 我们称之为节点的自私行为, 这种行为会降低组播会话的总体性能。

为了解决 DONet 中节点自私行为, 提高流媒体会话的总体服务质量, 本文设计了一种自私感知的 DONet (Selfishness-Aware Data-driven Overlay Network, 简称 SA-DONet 机制), 包括利他值计算算法以及数据段请求和发送算法。利他值与节点的转发贡献相关。数据段请求和发送算法根据利他值区分节点, 使得利他值高的节点被优先选择和服务。具体来说, 当请求数据段时, 优先选择利他值高的节点; 当多个请求同时到达数据段提供节点时, 优先处理利他值高的节点的请求。仿真结果表明, 即使存在一定比例的自私节点, 使用 SA-DONet 机制也能获得较好的应用层组播性能。

2 相关工作

如前所述, 应用层组播节点是具有独立利益的, 它们可能是自私的。Chu 等人分析了 P2P 流媒体中利他行为的重要性, 指出一小部分节点的利他行为就可以显著改善整个系统的性能^[9]。Habib 等人评价了系统在节点搭便车时的性能, 指出如果一些节点不合作, 应用层组播的服务质量就会变差^[10]。Shetty 等人给出两种检测应用层组播中恶意节点的方法: SA 和 RM^[11], 但是, 这两种方法都需要一个集中式的管理器, 来保存和计算节点的可信度。

为了解决自私性问题, chu 等人提出了一种税收模型, 使得资源丰富的节点为系统贡献更多的带宽, 以资助资源贫乏的节点。通过资源再分配, 提高了整个系统的效用^[12]。但是, 该模型假定流媒体源有权对节点征税。

Jun 等人提出了一种信任感知的组播 TAM^[13]。TAM 根据节点对不合作行为的报告来评估节点的信任值, 并根据节点的信任值调整组播树, 使得信任值高的节点离数据源更近。因为无法区分谁是恶意节点, 当收到负面报告时, TAM 同时惩罚报告节点和其父亲节点。而

且, 在 TAM 中, 节点的信任值采用集中式算法来计算。

Tan 等人设计了一种用于 P2P 网络中的, 基于支付的激励和服务区分策略^[14]。使用该策略, 节点在向其他节点发送数据段时争取积分。数据流的发送分为若干个周期, 在每个周期开始时, 节点使用积分, 采用“市场”机制相互竞争, 赢取最好的数据提供节点。但是, 该方法要求部署一些集中式的安全支付系统, 例如一些银行服务器来管理用户帐户和支付过程。

Li 等人提出了一种用于 DONet 网络中的服务质量感知的流构造策略^[15], 这是我们所知的唯一的关于 DONet 中节点自私行为的相关研究。为了得到最好的服务质量, 节点必须在请求每个数据包时, 发送一些探测包来测量到相关节点的距离和带宽, 因此, 引入了大量的主动测量, 而且, 实际上, 基于单个时间点的测量结果通常是不准确的。

3 SA-DONet 机制

为了解决 DONet 中节点的自私性行为, 本文提出了 SA-DONet 机制。目标是通过历史统计数据来检测自私节点, 而不用进行额外的测量; 并且使得系统尽可能分布式。本节首先给出利他值的定义和计算方法, 然后描述数据段请求和发送算法。使用这些方法, 利他值高的节点将被优先选择。

3.1 利他值的定义和计算

3.1.1 利他值的定义

我们统计在某个固定的时间 t 内, 节点 u 从节点 v 请求和接收的数据段数目。假设 S_{uv} 为节点 u 向节点 v 请求并且在 u 设定的播放截止时间内到达的数据段数目, F_{uv} 为节点 u 向节点 v 请求但没有在 u 设定的播放截止时间内到达的数据段数目, 然后, 我们给出如下定义:

定义 1 称二元组 $\langle S_{uv}, F_{uv} \rangle$ 为节点 u 对节点 v 的评价。

定义 2 称 $C_{uv} = \frac{S_{uv}}{S_{uv} + F_{uv}}$ 为节点 u 对节点 v 的推荐度。

定义 3 假设 K_v 为向节点 v 请求过数据段的节点集合, 称 $T_v = \frac{\sum_{r \in K_v} (C_{rv} \times T_r)}{\sum_{r \in K_v} T_r}$ 为节点 v 的利他值。

定义 2 中推荐度的定义只考虑了节点的直接经验, 而直接经验通常具有随机性和主观性。在 DONet 网络中, 一个节点可能需要从另外一个节点处请求数据, 而之前却没有和该节点进行过交互或者交互次数很少。这种情况下, 就没有直接经验或者直接经验不太可靠, 而使用定义 3 中的利他值定义, 通过加权其他节点的推

荐进行利他值的计算,就可以解决这个问题.在上述定义中, S_{uv} 保证接收到的数据包满足时间限制, C_{uv} 反映了丢包率,因此,利他值的定义包含了两个主要的网络指标:延迟和丢包率.注意:数据源的利他值永远为 1,其他节点的利他值初始化为 0.5.

定理 1 根据定义 3 求解节点利他值,解存在.

证明 定义 $D = \{(T_1, T_2, \dots, T_N) \mid 0 \leq T_i \leq 1, i = 1, 2, \dots, N\}$, 令: $A_j = \min_{i \in K_j} C_{ij}$, $B_j = \max_{i \in K_j} C_{ij}$, $D_0 = \{(T_1, T_2, \dots, T_N) \mid A_j \leq T_j \leq B_j, j = 1, 2, \dots, N\}$, $\mathbf{T} = (T_1, T_2, \dots, T_N)^T$ (右上角 T 表示转置).

构造映射 G 为: $G: D \subset \mathbb{R}^N \rightarrow \mathbb{R}^N$, $G(\mathbf{T}) = (g_1(\mathbf{T}),$

$$g_2(\mathbf{T}), \dots, g_N(\mathbf{T}))^T, g_j(\mathbf{T}) = \frac{\sum_{r \in K_j} (C_{rj} \times T_r)}{\sum_{r \in K_j} T_r}.$$

容易验证:

(1) D_0 为有界闭凸集;

(2) 由定义知 $g_j(\mathbf{T})$ 在 D_0 上连续,从而 G 在 D_0 上连续;

$$(3) \forall \mathbf{T} \in D_0, A_j = \frac{\sum_{r \in K_j} (A_j \times T_r)}{\sum_{r \in K_j} T_r} \leq \frac{\sum_{r \in K_j} (C_{rj} \times T_r)}{\sum_{r \in K_j} T_r} \leq$$

$$\frac{\sum_{r \in K_j} (B_j \times T_r)}{\sum_{r \in K_j} T_r} = B_j, \text{ 即 } A_j \leq g_j(\mathbf{T}) \leq B_j, \text{ 也即 } G(\mathbf{T}) \in D_0.$$

根据非线性分析中的 Brouwer 不动点定理(设 $G: D \subset \mathbb{R}^N \rightarrow \mathbb{R}^N$ 在有界闭凸集 $D_0 \subset D$ 上连续,且 $G(D_0) \subset D_0$,则 G 在 D_0 上至少有一个不动点,即存在一个点 $x^* \in D_0$,使得 $x^* = G(x^*)$)可知, G 在 D_0 上至少有一个不动点,即存在一个点 $\mathbf{T}^* \in D_0$,使得 $\mathbf{T}^* = G(\mathbf{T}^*)$,即定义 3 中非线性方程组的解存在,也就是说定义 3 的非线性方程通过若干次叠代收斂.

3.1.2 利他值的计算

本文设计了分布式求解节点利他值的方法.通过分布哈希表(DHT)机制,选择网络中的一些节点作为档案点来存储和计算利他值.DHT 使用 P-Grid 方法^[16].P-Grid 通过虚拟二叉树,分发数据备份到多个节点,并提供有效的检索.使用 P-Grid,档案点通过哈希节点的 ID 获得.每个节点存储多个数据,并维护一个路由表.当接收到查询或者更新请求时,根据路由表进行转发.P-Grid 机制可以保证在节点动态变化或者失效时,数据不丢失.P-Grid 方法的查询复杂度为 $O(\log N)$, N 为系统规模.

假设节点 u 向节点 v 请求数据段,每经过时间 t , u 对 v 进行评价,执行的 $\text{Eval}(\text{ID}_v)$ 过程如下:

(1) 统计 S_{uv} 和 F_{uv} ;

(2) 将 ID_u 、 S_{uv} 和 F_{uv} 写入节点 v 的档案点,并触发 v 的档案点进行利他值的计算.

假设节点 w 为节点 v 的档案点.节点 w 包含一个如表 1 所示的数据结构.其中,节点 v 的 ID 和利他值 (ID_v 和 T_v) 存入表 1 的第一行. ID_k 为评价节点, S_{kv} 和 F_{kv} 为 ID_k 对 v 的评价, LT_k 为 ID_k 最后更新评价的时间.每收到一个新节点的评价,在表 1 中增加一行.当收到已存在节点的评价时,更新表 1 中的相关数据.如果某行数据很长时间(例如 1 小时)都没有更新,删除该行数据.

表 1 档案点的数据结构

ID_v		T_v	
ID_{k_1}	$S_{k_1 v}$	$F_{k_1 v}$	LT_{k_1}
ID_{k_2}	$S_{k_2 v}$	$F_{k_2 v}$	LT_{k_2}
...
ID_{k_r}	$S_{k_r v}$	$F_{k_r v}$	LT_{k_r}

每收到一个评价, w 运行 $\text{Comp}T(\text{ID}_v)$ 计算 v 的利他值. $\text{Comp}T(\text{ID}_v)$ 的工作过程如下:

(1) 对于表 1 中的每个 $S_{k_r v}$ 和 $F_{k_r v}$, 按照定义 2 计算

$$C_{k_r v}, \text{ 即 } C_{k_r v} = \frac{S_{k_r v}}{S_{k_r v} + F_{k_r v}};$$

(2) 根据 ID_{k_r} 查询 T_{k_r} ;

$$(3) \text{ 按照定义 3 计算 } T_v, \text{ 即 } T_v = \frac{\sum_r (C_{k_r v} \times T_{k_r})}{\sum_r T_{k_r}}.$$

在上述分布求解算法中,任意节点 u 每经过时间 t , 通过 $\text{Eval}(\text{ID}_v)$ 将新的评价写入节点 v 的档案点(消息复杂度为 $O(\log N)$). v 的档案点每收到一个评价,通过 $\text{Comp}T(\text{ID}_v)$ 对 v 的利他值进行重新计算,在重新计算的过程中需要获取对 v 进行评价的节点的利他值(消息复杂度为 $O(\log N)$).因此,整个 DONet 网络中节点的利他值是通过叠代计算出来的.通过调整 t 的取值,可以控制评价粒度和计算量.

3.2 数据段请求和发送算法

有了节点的利他值,就可以在数据段请求和发送时,对不同利他值的节点作不同的处理.

3.2.1 数据段请求算法

如前所述,在 DONet 中,伙伴关系和数据的传输方向都是动态的.节点不断地和它的伙伴交换各自拥有的数据段信息,并决定从那个伙伴节点请求那些数据段.DONet 采用一个快速响应的启发式算法.在这个算法中,首先计算出数据段的潜在提供者的数量.因为如果一个数据段对应的潜在提供者越少,那么它越难在播放截止时间之前到达.算法会从仅有一个潜在提供者的分段开始确定某一分段的提供者,之后是对应有两个潜在提供者的分段,以此类推.如果一个分段对应

着多个潜在提供者,那么具有最高带宽和足够可用时间的提供者会被选中。

SA-DONet 机制对上述 DONet 的算法进行了改进。基于节点利他值,SA-DONet 的数据段请求策略如下:节点首先调度只有一个提供者的数据块,再获取有两个提供者的数据块,依次类推。如果有多个伙伴节点拥有所需的数据段,在保证时间限制的情况下,选择利他值最大的提供者。时间的计算会考虑带宽、距离等因素。

如前所述,数据源的利他值永远为 1。对于第一次加入组播组的节点,其利他值为 0.5;对于离线若干时间后,重新加入组播组的节点,其利他值通过 3.1 节介绍的算法获得。另外,为了避免数据流的抖动,节点需要一段缓存区来缓存数据(称为缓存窗口),而且在收到一定数量的数据段后才开始播放(称为播放等待窗口)。

SA-DONet 的数据段请求算法如图 1 所示,计算复杂度为 $O(B * M)$,其中 B 为缓存窗口大小, M 为节点维护的伙伴节点个数。

Procedure segment Requesting()
Begin
//计算分段潜在提供者的数量
for $i \in \text{expected_set}$ do //expected_set 为需要获取的数据段集合
$n = 0$
for $j = 1$ to num_partners do //num_partners 为伙伴节点数目
$n = n + bm[j, i]$ //bm[j, i] 表示伙伴节点 j 的 BM 关于数据段 i 的取值
end for j
//只有一个潜在提供者的数据段
if $n = 1$ then
$k = \arg, \{bm[r, i] = 1\}$ //k 为数据段 i 的提供节点
SendReq(k, i) //向节点 k 请求数据段 i
else
dup set[n] = dup_set[n] \cup { i } //dup_set[n] 表示有 n 个潜在提供者的数据段集合
end if
end for i
//有多个潜在提供者的数据段
for $n = 2$ to num_partners do
for each $i \in \text{dup_set}[n]$ do
//k 为数据段 i 的提供节点; T_j 为节点 j 的利他值;
$t[j, i]$ 为节点 j 传输数据段 i 所需的时间;
//DL[i] 为数据段 i 的播放截止时间; set_partners
为伙伴节点集合
$k = \arg, \{T_r > T_r' \mid t[r, i] < DL[i], t[r', i] < DL$
[i], $r, r' \in \text{set_partners}\}$
if $k \neq \text{NULL}$ then
SendReq(k, i) //向节点 k 请求数据段 i
end if
end for i
end for n
End

图 1 数据段请求算法

3.2.2 数据段发送算法

在 DONet 中,当请求到达数据段提供节点时,先进入请求队列。如果有多个请求,数据段提供节点必须考

虑如何对请求进行排队。通常,数据段提供节点按照请求的到达次序对请求进行排序,越早到达的请求,越接近对头,称为 FIFO 方法。使用 SA-DONet 机制,数据段提供节点按照请求节点的利他值对请求进行排序,利他值高的节点的请求将更接近队头。

数据段提供节点处理数据段请求时,从请求队列的队头取出请求节点(其利他值最大)和所请求的数据段,然后发送相应的数据段给请求节点。相对于 FIFO 方法,使用 SA-DONet 机制的排队策略,可以使利他值高的节点的请求得到优先处理。

SA-DONet 的请求排序算法和数据段发送算法如图 2 所示。如果有 n 个请求同时到达,请求排序算法和数据段发送算法的计算复杂度分别为 $O(n \log n)$ 和 $O(n)$ 。

Procedure requestOrdering(request)
Begin
node = request.source
seg = request.segment
$T = \text{QueryT}(node)$ //查询节点利他值
InsertByT(node, seg) //按照利他值的由大到小的顺序,插入请求队列
End
Procedure segmentSending()
Begin
while true do
if queue is not NULL then //如果队列非空
QueueOut(&seg, &node) //从对头取出请求节点和所请求的数据段
SendSeg(seg, node) //发送 seg 给 node
end if
end while
End

图 2 请求排序算法和数据段发送算法

4 仿真结果

本文使用网络模拟器(NS2)^[17]对 SA-DONet 机制进行了仿真,分析了 SA-DONet 机制对 DONet 中应用层组播性能的影响。网络拓扑是使用 GT-ITM^[18]产生 Transit-Stub 模式的层次化结构。自私节点被模拟为随机的停止或者延迟转发某些数据段。仿真中涉及的参数及其取值见表 2。所有的仿真结果取 5 次运行的平均值。

在参考文献[6]中,zhang 等人提出了 DONet 机制。在该文中,作者分析了连续性指标和伙伴数量 M 的关系(见 DONet^[6]的图 7)。结果显示连续性会随着 M 的增长而加大,因为每个结点都会有更多的资源提供者以供选择;但是在伙伴超过 4 个之后,这种增长将变得不

再明显.另外,该文还将连续性指标表示为流速率的函数,并且发现,即使在高速率下,伙伴数量为4时仍然会达到很好的效果(见 DONet^[6]的图8).再考虑到控制开销会随着伙伴数量的增长而加大(见 DONet^[6]的图6),最后作者给出结论: $M=4$ 在实际中将会是一个很好的选择.所以,本文我们也只仿真了 $M=4$ 的典型场景.

表 2 模拟参数及其取值

Parameter	Description	Value
N	Number of nodes in the overlay network	500-1500P
p	Selfish node percentage	0-80%
t	Evaluating interval	15-120(s)
M	Number of partners each node maintains	4
B	Buffer window size	70
W	Playing waiting segment number	30
S	Number of segments of the stream	6000

4.1 评价间隔对 SA-DONet 性能的影响

本小节考察评价间隔(t)的取值对 SA-DONet 性能的影响.在下面的实验中, $N=500, p=50\%$.

这里,我们定义请求成功率为接收到的数据段占所请求的数据段的比例.从图3可以看出,随着评价间隔增大,请求成功率减小.这是因为利他值更新越慢,向自私节点请求数据段的机会越大.

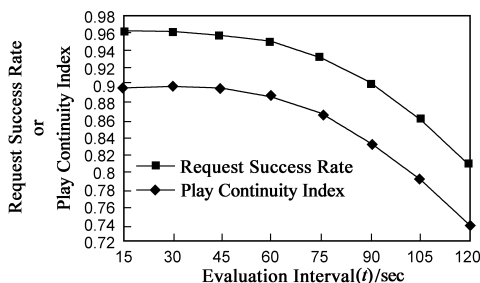


图3 请求成功率/播放连续因子随 t 的变化情况($N=500, p=50\%$)

播放连续因子被定义为在播放截止时间前到达的数据段占有所有数据段的比例,反映了用户接收到的流媒体的连续程度,记为 PCI.假设数据段 s 到达节点 k 的时间为 $ArrivalTime(k, s)$,节点 k 中段 s 播放截止时间为 $Deadline(k, s)$,那么播放连续因子可以表示为 PCI

$$PCI = \frac{|\{s | s \leq S, \text{ and } ArrivalTime(k, s) \leq Deadline(k, s)\}|}{S * N}$$

图3显示, t 取值越大,播放连续因子越小.这是由于利他值的更新间隔越长,节点接收到的数据段来自低利他值的节点的几率越大,从而连续性越差.

我们用归一化控制负荷(即控制消息流量占总流量的比值)来测量控制负荷.当利他值更新间隔变长时,虽然和利他值计算相关的控制消息减少了,但由于请求成功率的下降,获取数据段的控制消息反而增加了,结果,如图4所示,SA-DONet的归一化控制负荷随 t

的增大变化不大.

从图3和图4的结果可知, t 取值越小,SA-DONet的服务质量更高,而归一化控制负荷变化不大.从第3.1节利他值的计算可知, t 取值越小,档案点的计算量越大.所以, t 的具体取值,可以根据情况,在性能和计算量之间折中.

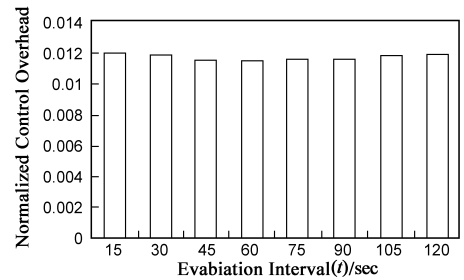


图4 归一化控制负荷随 t 的变化情况($N=500, p=50\%$)

4.2 SA-DONet 和 DONet 的性能比较

本小节对 SA-DONet 和 DONet 的性能进行比较.在下面的仿真中, t 均取60秒.

图5比较了在不同网络规模下,SA-DONet 和 DONet 的请求成功率随自私节点比例的变化情况.图5说明,当没有自私节点时,SA-DONet 和 DONet 的请求成功率都最大.随着自私节点比例的增加,SA-DONet 和 DONet 的请求成功率都减小,但是使用 SA-DONet 机制可以明显提高请求成功率.即使存在80%的自私节点时,SA-DONet 仍能获得较高的请求成功率(大于90%).另外,网络规模越大,请求成功率通常越大.

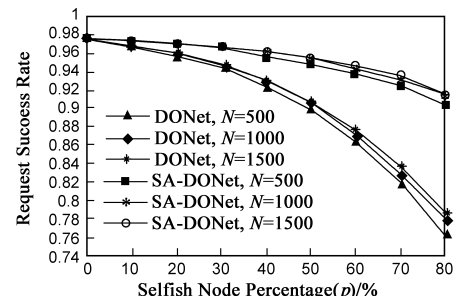


图5 不同网络规模下,请求成功率随自私节点比例的变化情况($t=60$ sec)

图6比较了在不同网络规模下,SA-DONet 和 DONet 的播放连续因子随自私节点比例的变化情况.当没有自私节点时,SA-DONet 和 DONet 的播放连续因子最大.当自私节点增多时,SA-DONet 和 DONet 的播放连续因子都减少,而且几乎所有的 SA-DONet 的播放连续因子都大于 DONet.即使80%的节点为自私节点,SA-DONet 算法仍能保持比较高的播放连续因子(大于82%).同样,网络规模越大,一般来说,播放连续性越好.

控制负荷的比较见图7.SA-DONet 和 DONet 的归一化控制负荷都随着自私节点比例的增加而变大.但是,相对于流媒体数据包来说,控制负荷还是相当小.图7

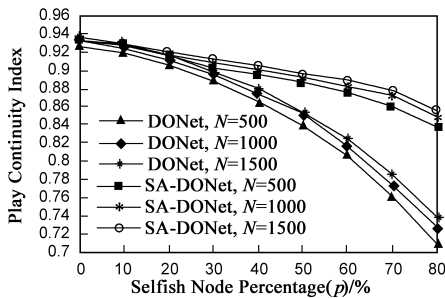


图6 不同网络规模下,播放连续因子随自私节点比例的变化情况($t=60$ sec)

还表明,SA-DONet 比 DONet 多引入约 6% 的控制负荷。另外,随着组播组规模的增大,SA-DONet 和 DONet 的控制负荷都变化不大,因此,SA-DONet 和 DONet 具有很好的扩展性,可以应用到大规模的组播环境中。

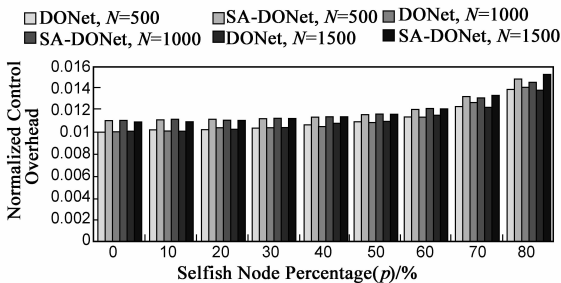


图7 不同网络规模下,归一化控制负荷随自私节点比例的变化情况($t=60$ sec)

5 结论

和其他 P2P 网络一样,DONet 面临着节点自私性问题。自私节点可能有意或者无意的停止或延迟转发某些数据段,导致流媒体质量下降。为了解决这个问题,本文提出了 SA-DONet 机制。SA-DONet 根据一个节点对其他节点的转发贡献,为该节点计算出一个利他值,并设计了相应的数据段请求和发送算法。SA-DONet 机制的特点表现在如下方面:利他值采用分布式方法计算,而且无需进行额外的测量;利他驱动的数据段请求和发送策略取得了较好的应用层组播性能,但保持和 DONet 差不多的计算复杂度。仿真结果表明,使用 SA-DONet 机制,可以提高组播会话的总体性能,并且具有较低的控制负荷。

参考文献:

[1] Chu Y H, Rao S G, Seshan S, Zhang H. A case for end system multicast[J]. ACM SIGMETRICS Performance Evaluation Review, 2000, 28(1): 1 - 12.

[2] Zhang B, Jamin S, Zhang L X. Host multicast: A framework for delivering multicast to end users[A]. Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)[C]. New York, NY, USA: IEEE Press, 2002. 1366 - 1375.

[3] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast[J]. ACM SIGCOMM Computer Communication Review, 2002, 32(4): 205 - 217.

[4] Eugster P T, Guerraoui R, Kermarrec A M, Massoulié L. From epidemics to distributed computing[J]. IEEE Computer, 2004, 37(5): 60 - 67.

[5] Banerjee S, Lee S, Bhattacharjee B, Srinivasan A. Resilient multicast using overlays[J]. ACM SIGMETRICS Performance Evaluation Review, 2003, 31(1): 102 - 113.

[6] Zhang X Y, Liu J C, Li B, Yum T P. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming[A]. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)[C]. Miami, FL, USA: IEEE Press, 2005. 2102 - 2111.

[7] Adar E, Huberman B. Free riding on Gnutella[J]. First Monday, 2000, 5(10): 1 - 22.

[8] Hughes D, Coulson G, Walkerdine J. Free riding on Gnutella revisited: the bell tolls[J]. IEEE Distributed Systems Online, 2005, 6(6): 1 - 18.

[9] Chu Y, Zhang H. Considering altruism in Peer-to-Peer Internet streaming broadcast[A]. Proceedings of the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2004)[C]. Cork, Ireland: ACM Press, 2004. 10 - 15.

[10] Habib A, Chuang J. Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming[J]. IEEE Transactions on Multimedia, 2006, 8(3): 610 - 621.

[11] Shetty S, Galdames P, Tavanapong W, Cai Y. Detecting malicious peers in overlay multicast streaming[A]. Proceedings the 31st IEEE Conference on Local Computer Networks (LCN 2006)[C]. Tampa, FL, USA: IEEE Press, 2006. 499 - 506.

[12] Chu Y, Chuang J, Zhang H. A case for taxation in Peer-to-Peer streaming broadcast[A]. Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems (PINS 2004)[C]. New York: ACM Press, 2004. 205 - 212.

[13] Jun S, Ahmad M, Jun X. Robust information dissemination in uncooperative environments[A]. Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005)[C]. Washington, DC, USA: IEEE Computer Society, 2005. 293 - 302.

[14] Tan G, Jarvis S A. A payment-based incentive and service differentiation scheme for peer-to-peer streaming broadcast[J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(7): 940 - 953.

[15] Li D, Wu J P, Cui Y, Liu J C. QoS-aware streaming in overlay multicast considering the selfishness in construction action [A]. Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)[C]. An-

chorage, USA: IEEE Press, 2007. 1154 – 1162.

- [16] Aberer K. P-Grid: a self-organizing access structure for P2P information systems[A]. Proceedings of the 6th International Conference on Cooperative Information Systems[C]. Berlin: Springer, 2001. 179 – 194.
- [17] VINT Group. The Network simulator—ns-2 [EB/OL].

<http://www.isi.edu/nsnam/ns/index.html>, 2009-06-18.

- [18] Zegura E W, Calvert K L, Bhattacharjee S. How to model an internetwork[A]. Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer Societies (INFOCOM'96) [C]. San Francisco, USA: IEEE Press, 1996. 594 – 602.

作者简介:



王 森 女, 1975 年 11 月出生于陕西省乾县. 现任中国科学院计算技术研究所助理研究员, 主要研究方向为分布式计算、覆盖网络.

E-mail: wangm@ict.ac.cn

张玉军 男, 1976 年 6 月出生于河北省衡水. 现任中国科学院计算技术研究所副研究员, 中国计算机学会高级会员, 主要研究方向为下一代网络、移动计算. E-mail: zhmj@ict.ac.cn

李国杰 男, 1943 年 5 月生于湖南省邵阳市. 现任中国工程院院士、中国科学院计算技术研究所所长、中国计算机学会理事长、中国工程院信息与电子学部主任、863 信息领域专家委员会副主任、国家信息化专家咨询委员会委员、全国人大代表. 主要研究方向为并行处理、计算机体系结构、人工智能. E-mail: lig@ict.ac.cn