

一种应用于移动 P2P 网络的资源协作共享策略

牛新征,周明天,余 堃

(电子科技大学计算机科学与工程学院,四川成都 610054)

摘要: 移动 P2P 网络的移动节点设备资源短缺,网络动态多变,移动节点间的协作显得越发重要.本文提出了一种灵活的移动节点间资源协作共享方案,为资源的请求者设计了基于可靠性理论的请求资源预测算法.依据排队理论为资源的提供者,构建了层次型资源调度模型.数学分析和仿真结果表明该预测算法提高了节点请求任务执行成功率,层次型资源调度模型则提高了节点协作共享资源的服务能力,降低了资源请求丢失率.

关键词: 移动 P2P 网络; 排队模型; 协作共享; 可靠度函数

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 01-0018-07

A Cooperative Sharing Scheme for Resources in Mobile P2P Networks

NIU Xin-zheng, ZHOU Ming-tian, SHE Kun

(School of Computing Science and Engineering, UESTC, Chengdu, Sichuan 610054, China)

Abstract: Since the resources of mobile devices are scarcer and the network is dynamic, cooperation among mobile nodes is an important task. A novel flexible resource sharing and cooperating scheme among mobile nodes is proposed in the paper. A prediction algorithm of resources based on reliability theory for resources' requestor is also presented. According to the queue theory, a multi-level model of resource dispatching is put forward for resources' provider. While analysis and simulation results show that the prediction algorithm can increase success ratio of sharing resources' cooperation task, the resources dispatching model can significantly improve the service quality, and reduce the loss rate of resource requests.

Key words: mobile P2P networks; queue model; cooperative sharing; reliability function

1 引言

由于移动 P2P 网络节点资源的贫乏,网络的动态多变,因此,移动节点间的协作显得越发重要.对资源协作共享策略,目前已有多个研究成果.文献[1]提出了协作共享方案(COCA),移动用户可以通过邻居节点的缓存来取得需要的数据.文献[2]提出了一种新的协作共享方案-邻近区域共享的方案(PReCinCi).其协作策略不仅考虑共享数据块被自身节点使用的次数,也考虑其对于别的节点的作用.文献[3]对文献[2]做了进一步的优化工作.文献[4]评价了移动 ad hoc 网络中有效支持数据访问的协作共享方案,并且提出了协作共享数据(CacheData)和协作共享路径(CachePath)方案,以及混合(HybridCache)的方案.文献[5]提出了一个邻居协作计划,节点可以放置数据在资源丰富的邻居节点上,并且提出了选择合适邻居节点作为协作节点以及基于预测的协作节点排序选择算法.

上述研究成果主要是针对如何建立节点的协作共享资源模型,如何在移动网络中支持节点间的协作,如

何建立协作共享数据和协作共享节点等问题.本文提出了一种应用于移动 P2P 网络的资源协作共享策略.其中,提出了一种请求资源的预测算法,能及时地、准确地将移动节点的协作共享资源请求提交给资源的提供者,增加了移动节点间协作共享资源交互的成功率.同时,提出了不仅基于资源请求任务的优先值,而且充分考虑请求被提出执行的最初时间以及请求任务需求的资源业务类型的层次型共享协作资源调度模型.

2 资源协作问题描述

移动节点需要访问其余移动节点提供的协作共享资源,而移动节点同时也需要向外提供协作共享资源,完成其余移动节点的资源请求.整个移动 P2P 网络中移动节点资源的协作共享过程如图 1 所示.

2.1 资源请求

尽量保证第一次的请求能及时地到达资源提供者或者增加获得准确资源请求目的地址的可能性.对于资源请求者,本文通过基于可靠性理论的预测资源算法,增加了协作交互任务执行的成功率.

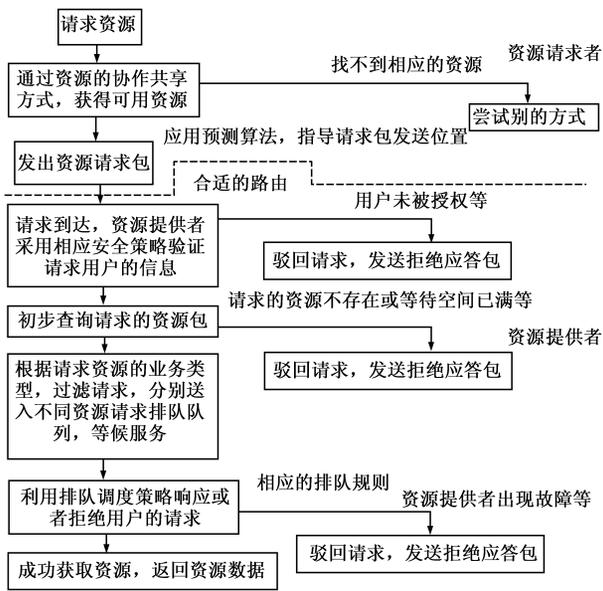


图1 移动节点间资源协作共享过程

2.2 资源提供

本文利用排队理论对网络资源提供者协作资源的请求调度执行这一系列的业务建立数学模型,并且根据其中的系统内排队顾客的平均数,系统损失概率等重要的模型参数,分析各种因素间的相互影响关系.使资源协作既能及时响应请求者的资源请求,又能最大限度地向外提供协作缓存资源.该策略充分考虑了用户由于贡献大小而赢得的优先值不同得到协作资源不同,用户的业务类型不同得到的资源时间先后不同,以及用户等待时间长短不同而不同等等.这种层次性的设计方式是根据请求任务执行的特性,保证用户的协作资源需求被合理及快速地响应,防止了对请求业务类型的不区分而造成获得的数据已过期,对用户的贡献大小的不区分造成自私节点的增加而可用协作资源越来越少.

2.3 排队模型

采用 R 队列(本文假设 $R=2$)和单服务窗口的方式,根据排队理论得到移动节点协作共享资源策略的排队模型如下:

本文的协作共享资源请求调度策略的排队模型是 $M/M/S/L_{\text{rong}}/\infty/\text{优先权服务} + \text{先到先服务} + \text{损失值}$.由于移动节点的处理能力和资源等非常有限,为了不影响节点本身的任务执行,对于资源请求的服务,服务台个数为 1,记为 $S(S=1)$.排队空间的容量,记为 L_{rong} ($L_{\text{rong}} \geq 1$),在下面公式组(6-8)表示为 m .该系统容量可调整,在某段时间内相对固定不变,并且是有限的.由于移动节点的任务过滤分发有一定的延迟,且专门设置的队列缓存与调度部分有缓存资源请求的能力,因此,系统容量等于 R 个等待队列能容纳的顾客总数

加上队列缓存与调度部分能容纳顾客数的总和.

3 资源预测算法

3.1 算法模型

根据对资源的协作共享行为的分析,参考了文献[6~9]的方法和思路,本文提出了一种基于可靠性理论的资源预测算法,通过引入可靠性理论对资源请求提供者的可靠性进行分析和量化评估,从而为增加获得资源的成功性提供了理论依据.

用 N 表示所有移动节点的总数,则网络中第 i 个移动节点唯一的网络标识 ID 可以表示为 $M_i, 1 \leq i \leq N$.可供移动节点 M_i 使用的协作资源表示为 $CR_j, 1 \leq j \leq N_{CR}, N_{CR}$ 为可以使用的协作资源种类数目, CT 表示提供资源 CR_j 的资源提供者总数,即 CT 个资源 CR_j 源.该可供移动节点 M_i 使用的 CT 个协作资源 CR_j 源中的任意一个资源 CR_j 可用一个 8 元组表示为: $Nd = (M_i, CR_j, RT_j, \theta_j, t_0, t_1, t_2, CT)$.其中,在某个协作资源 CR_j 源中,请求者 M_i 登记该资源块 CR_j 信息的初始时间为 t_0 ,该资源块最后一次被请求者使用到的时刻为 t_1 ,请求者 M_i 需要对它进行再次使用的时刻为 t_2 ,该资源 CR_j 被访问的总次数为 RT_j ,该资源 CR_j 被访问 RT_j 次能够成功获得资源的比率为 θ_j .

对 CT 个协作资源 CR_j 源中的任意一个 CR_j 资源的可靠性计算如下:首先,根据资源 CR_j 以前被请求者 M_i 访问的记录,可以求得单位时间该资源块被请求者成功访问的次数 τ_j ,简单地表示为

$$\tau_j = \frac{RT_j \times \theta_j}{t_2 - t_0} \quad (1)$$

其次,在可靠性理论中,通常会用一个非负的随机变量 X 描述产品的寿命, X 的分布函数为 $F(t) = P\{X \leq t\}, t \geq 0$,以 $F(t)$ 表示资源提供者提供的资源 CR_j 在时刻 t 以前都可用的概率.根据文献[8]和可靠性理论,可选取下面式子作为 X 分布函数.

$$F(t) = \begin{cases} \frac{1}{e^{\tau_j(t+1)}} & , t \geq 0 \\ 1 & , t < 0 \end{cases} \quad (2)$$

再者,产品的可靠度函数可表示为 $R(t) = P\{X > t\} = 1 - F(t)$, $R(t)$ 表示一个空闲时间片大于 t ,资源 CR_j 不失效的可用概率.

$$R(t) = 1 - F(t) = \begin{cases} 1 - \frac{1}{e^{\tau_j(t+1)}} & , t \geq 0 \\ 0 & , t < 0 \end{cases} \quad (3)$$

若资源 CR_j 在时刻 t 以前都可用,则可用度 $u(t)$ 为在 $(t, t + \Delta t]$ 时间内可使用的可能性大小, $r(t)$ 是 $R(t)$ 的概率密度.

$$u(t) = \frac{r(t)}{1 - R(t)} \quad (4)$$

将式(1)带入式(3),可得时刻 t_2 时资源 CR_j 的可用度:

$$u(t_2 - t_1) = \frac{RT_j \times \theta_j}{t_2 - t_0} + \frac{1}{t_2 - t_1 + 1} \quad (5)$$

式(5)为最终的可用度公式,可以通过该式预测求出时刻 t_2 该资源块 CR_j 可使用的可能性大小.显然,可用度越高,被成功获得资源的可能性越大.

因此,在资源请求者 M_i 需要再次使用该资源 CR_j 时,需计算资源 CR_j 的可用度 $u(t)$,对于资源 CR_j 的 CT 个资源提供者,资源 CR_j 的可用度 $u(t)$ 最高的资源提供者具有最大的可靠性,资源请求者 M_i 可以向该资源提供者发起资源请求,并且可以用一个 3 元组表示为 $Ur = (M_i, UR_j, t_2)$. UR_j 表示最后确定的可以使用的资源 CR_j 的信息.

3.2 算法设计

资源预测算法(RF)的输入为 Nd ,输出为 Ur ,主体为 Forecast()-预测函数.其功能是基于可靠性理论,通过可用资源的信息,预测资源可用性,指导资源协作.它能预测求出某个时刻某个资源源可被使用的可能性大小,并根据不同可用度排序,找到最值得信赖的资源源,向该资源源发出资源请求.同时,启动成功率刷新函数 $Rf(Nd, Ur)$,记录该资源请求的响应状况,Forecast()的伪代码如下:

```
Void Forecast (Nd, Ur) {
    获取  $CR_j$  源的数量  $CT$ ;
    if  $CT = 0$  then 结束该算法;
    if  $CT = 1$  then  $CR_j$  源信息存入  $UR_j$ , 并执行更新操作;
    for ( $i = 1; i < = CT; i++$ ) {
    if  $CR_j$  源的  $\theta_j = 0$  then 不计算其可用度;
        else {
        构造可用度公式  $u(t)$ ; //根据输入
        求出第  $i$  个  $CR_j$  源的可用度  $u(t)$ , 并存入临时数组  $U[]$ ; }
    } //end for 计算每个  $CR_j$  源可用度
    for ( $i = 1; i < = CT; i++$ ) //假设数组  $U[]$  大小为  $CT_U$  个 {
    把  $U[i]$  的  $u(t)$  从小到大排序;
    } //end for 寻找最大的可用度  $CR_j$  源
    if 可用度最大  $CR_j$  源唯一
    then 读取唯一  $CR_j$  源信息, 并存入  $UR_j$ ;
    else 随机选择一个  $CR_j$  源信息, 并存入  $UR_j$ ;
    增加  $RT_j$ , 修改  $t_1$ ; //选取  $CR_j$  源后的更新操作
     $Rf(Nd, Ur)$ , 并修改  $\theta_j$ ;
    //  $Rf(Nd, Ur)$  不细述 } //end Forecast ()
```

4 基于排队理论的协作资源请求调度

4.1 过程描述

应用排队理论^[10]结合移动 P2P 网络的实际情况,从以下几个方面评价、设计资源提供者的协作共享资

源的请求调度策略:(1)提出一个资源提供者协作共享资源的请求调度策略模型和分析方法;(2)对影响资源请求调度策略的几个重要的参数进行分析,并描述出他们之间的关系;(3)根据资源请求调度策略,证明该调度策略的可行性和有效性,为移动节点间的资源协作和共享的设计和规划提供依据.

本文用下面的 6 元组来标识资源请求包 $Re = (M_i, M_j, flag, seq, T_i, P_j(t))$, 节点 M_j 向节点 M_i 发出资源请求; $flag$ 表示该请求属于的业务类型标识, $flag = 0$, 表示为非实时数据请求, $flag = 1$, 表示为实时数据请求; seq 表示节点 M_i 为该资源请求包分配的序列号; T_i 为该资源请求包到达节点 M_i 的初始时间.在 t 时刻,节点 M_j 的优先值表示为 $P_j(t)$.优先值^[11]形式的虚拟货币就是移动节点贡献大小的具体体现.除此外, Pl_i 表示节点 M_i 中等待执行的资源请求的数目.优先值服务门限参数为 Γ (所有移动节点都使用同一个服务门限).

4.2 调度策略分析

本文的请求调度策略采用一种层次型的资源调度模型,主要分为两个方面的层次型设计.第一个方面:协作资源请求的响应方式采用了排队理论中即时制和等待制相结合的混合机制.首先,当 $Pl_i (1 \leq i \leq N) \geq L_{rong}$ 时,被请求的节点 M_i 对后续到来的请求包将采用即时制或称损失制的策略,拒绝后续的节点资源请求.其次,资源请求以 ϵ^n 的概率参加排队,中途可允许不耐烦的资源请求退出排队队列.并采用非抢占式优先权服务规则的等待制策略.第二个方面:协作资源请求的优先权层次性排序采用了三个判断机制,为优先值 $P_j(t)$, 业务种类 $flag$, 到达的时间先后 T_i .首先,当 $P_j(t) \geq \Gamma$ 时,移动节点 M_j 的资源请求才能被服务.其次,实时数据请求被响应的优先级高于非实时数据请求,应该在最短时间内响应其请求.最后,在考虑了请求的业务分类 $flag$ 后,以 T_i 为度量依据,应该响应等候服务时间最长的请求.

4.3 系统数学模型

为了提出协作共享资源请求调度策略的排队数学模型,先做下面的分析:(1)顾客流的概率分布可以用指数分布来描述,其强度表示为 λ .资源提供者的服务时间分布符合指数分布,其强度表示为 μ .资源服务台的“使用率”表示为 ρ , $\rho = \lambda/\mu$;(2)对于不同的资源请求通过过滤后分向不同业务类型排队等候队列 $R_i (i = 1, 2)$, 每个队列的优先权不同.同时, R_i 的排队队列空间大小 $L_{R_i(rong)}$ 可以根据系统的实际运行情况调整,必须满足系统容量 L_{rong} 等于 R 个等待队列中容纳的顾客总数加上队列缓存与调度部分中容纳顾客数的总和.因此,可以初步设置 $L_{R_2(rong)} \leq \frac{2}{3} L_{rong}$, $L_{R_1(rong)} =$

$\frac{1}{2}L_{R_2(ronq)}, L_{R_1(ronq)}$ 为实时数据请求队列的空间大小, $L_{R_2(ronq)}$ 为非实时数据请求队列空间大小.

根据排队理论^[10]的单通道混合制 ($M/M/1/m$) 和单通道混合制 (排队时间有限)^[12] 两种模型, 通过推导可以获得本文的排队模型. P_k 表示系统内有 k 位顾客 (资源请求) 的状态概率. 系统内顾客数目 L_{xi} 的数学期望等于正在被服务顾客的平均数 L_{fu} 加上系统内排队顾客的平均数 L_{dui} . 系统内排队顾客的平均数表示为:

$$L_{dui} = \varepsilon^0 \rho^1 P_0 + 2\varepsilon^0 \varepsilon^1 \rho^2 P_0 + \dots + m\varepsilon^0 \varepsilon^1 \dots \varepsilon^m \rho^{m+1} P_0$$

$$= P_0 \sum_{i=2}^{m+1} (i-1) \times \varepsilon^{\frac{i(i-1)}{2}} \rho^i \quad (6)$$

系统损失概率表示为:

$$P_{shun} = P_{m+1} = \rho^{m+1} \varepsilon^{\frac{m(m+1)}{2}} P_0 \quad (7)$$

资源请求在资源提供者中排队等待的时间, 其平均用值 W_{dui} 表示, 在服务提供者中的执行时间, 其平均用值 W_{fu} 表示. 资源请求在资源提供者中的运行时间 W_{xi} 等于排队等待时间加上执行时间. 顾客平均排队时间表示为:

$$W_{dui} = \varepsilon^0 \rho P_0 \frac{1}{\mu} + \varepsilon^0 \varepsilon^1 \rho^2 P_0 \frac{2}{\mu} + \dots + \varepsilon^0 \varepsilon^1 \dots \varepsilon^{m-1} \rho^m P_0 \frac{m}{\mu}$$

$$= \frac{P_0}{\mu} \sum_{i=1}^m i \times \varepsilon^{\frac{i(i-1)}{2}} \rho^i \quad (8)$$

4.4 调度策略步骤

本系统是采用混合制排队模型请求调度系统, 以图 2 为例描述调度策略.

第一步: 更新记录各种参数, 检测排队系统. 在某个时刻, 更新各个参数, 记录当前的各个参数的准确值, 了解目前排队模型的运行状况. 包括 L_{rong}, W_{dui} 等.

第二步: 根据目前资源请求的排队等候队列的大小, 从缓存调度系统中根据不同的业务种类分发请求, 利用混合机制对请求进行分发处理, 分发请求到不同的排队队列中等候, 直到下面条件之一发生, 就终止分发. (1) 资源请求排队等候队列已满; (2) 节点突发错误, 不能完成资源的请求执行动作; (3) 资源请求者需要的资源在资源提供者已经不存在. 其中对于很长时间未能服务的资源请求, 以及不耐烦的请求允许放弃执行.

第三步: 根据协作资源请求的优先权层次性排序原则对排队等候队列中的资源请求进行调度处理, 直到下面的条件之一发生, 就停止选择请求的执行. (1) 节点的资源服务窗口没有能力接受任务的执行; (2) 排队等候队列为空; (3) 资源服务窗口发生故障.

第四步: 如在第三步的条件 (2) 发生, 回第二步继续分发请求或处于就绪状态.

第五步: 如在第二步的条件 (1) 和第三步的条件 (1)

发生, 各阶段处于等候状态.

第六步: 如在第二步的条件 (2), (3) 和第三步条件 (3) 发生, 回第一步检测系统.

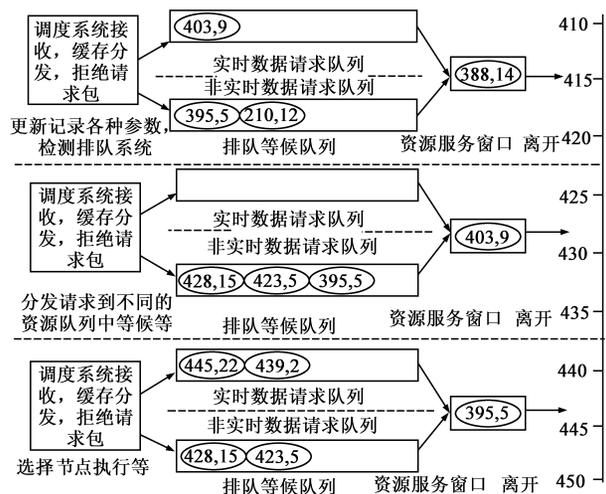


图2 请求任务的排队策略调度实例

图 2 展示了调度策略步骤, 整个排队策略随着时间的推移不断进行. 右侧的时间轴表示随着排队模型的步骤进行, 时间在推移, 时间轴是从节点开始运行时刻根据网络的标准时间开始计算的.

其中, 圆圈表示资源请求, 第一个参数表示请求排队等候的初始时间, 第二个参数表示请求需要占用服务窗口的大约时间. 非实时数据请求队列中的任务 X (被移动节点接收的初始时间戳为 210), 由于等待时间过长, 提前放弃请求的执行; 实时排队请求任务 Y (初始时间戳为 403) 尽管排队时间晚于非实时排队请求任务 Z (初始时间戳为 395), 仍给予优先执行; 实时排队请求任务 W (初始时间戳为 439) 不能抢占正在执行的非实时排队请求任务 Z 的资源服务窗口.

5 资源协作共享策略分析

5.1 排队模型分析

根据 4.3 部分推导出的排队数学模型, 分析了其中两个最重要的模型参数, 分别是系统内排队顾客的平均数, 系统损失概率, 其余的参数都是以这两个参数为基础变形的, 分别见图 3 与图 4. 其中 $\varepsilon = 0.9$ (可调整), 从图 3 可以看出当资源服务台的“使用率”比较小时, 表示服务的平均强度要大于请求到达的强度, 排队顾客不会太多, 排队顾客稳定在 12 ~ 13 个. 因此, 对于排队空间建议 $\rho = 0.6$ 时设置为 15 个, 就可以满足资源的排队请求, 排除由于排队空间不够而拒绝资源请求的情况发生. 从图 4 可以看出, 当 $\rho = 0.9$ 时, 由于排队空间需要的数量较大. 因此, 损失概率曲线下降较缓慢, 到达 25 左右的时候, 损失概率才降到 0 附近, 即没有由于排队空间不够而拒绝资源请求的情况发生. 同

时,由于其使用率较高,在排队空间最小时损失概率最大.上述分析表明,根据该资源请求调度模型,在确定一定的服务台使用率的情况下,能够稳定排队空间队列到一个理想的理论目标值,并获得较小的排队空间队列抖动.

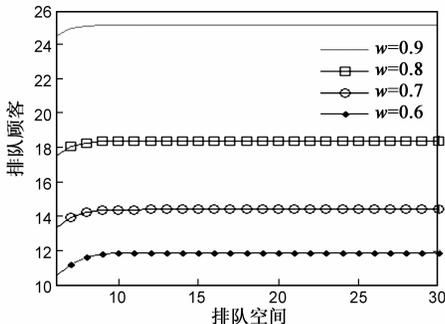


图3 排队空间与排队顾客的关系(w 即为 ρ)

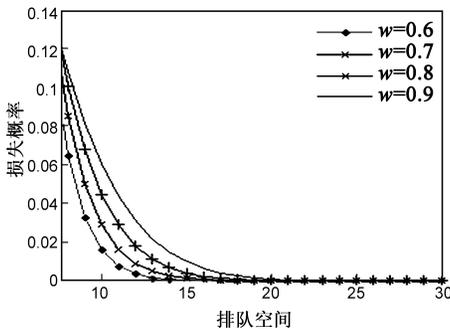


图4 排队空间与损失概率的关系

5.2 正确性证明

性质 1 如果某个协作资源存在资源提供者,根据资源预测算法,能找到该资源最大可靠性的提供者.

证明 假设请求者 M_i 需要的资源 CR_j 有 CT 个资源提供者,尽管对于某个资源提供者提供的资源 CR_j 在时刻 t 以前都可用,但是需要考虑在 $(t, t + \Delta t]$ 时间内可使用的可能性大小,即预测资源 CR_j 目前是否可用.本文应用可靠性理论,构造出若资源 CR_j 在时刻 t 以前都可用,则它在 $(t, t + \Delta t]$ 时间内可使用的可能性大小 - 可用度 $u(t)$.其中,从最终的可用度 $u(t)$ 公式可以发现,该公式不但考虑了该资源 CR_j 被成功使用的频率 τ_j , τ_j 增大,可用度 $u(t)$ 增大.并且考虑了该资源 CR_j 最久未被使用的时间间隔 Δt , Δt 增大,可用度 $u(t)$ 降低.也即表示该资源 CR_j 已经很久未被使用,对于动态性的移动 P2P 网络来说,该资源的可靠性应该降低.因此,可见该公式与几个时间因素和历史使用次数都有关,充分考虑了使用频率和使用间隔两种因素,较公平地评价出该资源目前是否可用,对于多个提供资源 CR_j 的提供者来说,应用可靠性理论求出资源 CR_j 可靠性最大的提供者,通过该预测算法能提高获取资源的成功率.

性质 2 基于排队模型的请求调度策略,能够找到合适的请求排队空间大小.

证明 基于单通道混合制和单通道混合制两种模型,可以推导出本文的排队模型.根据该排队模型,分析服务的平均强度 μ 和请求到达的强度 λ ,得到资源服务台的“使用率” $\rho (\rho = \lambda/\mu)$.并且根据请求等候的耐心强度 ϵ ,可以推导出排队顾客 L_{dui} 和排队空间 m 的关系,就能找到不同的系统损失概率 P_{shun} 下,排队空间的大小(如图 3,4).同时,可用相同的方法,对顾客平均排队时间 W_{dui} 等参数进行理论分析.

5.3 复杂性分析

请求资源预测算法的适用性主要取决于计算的复杂度.而预测算法由两部分组成:可供移动节点 M_i 使用的协作资源 $CR_j, 1 \leq j \leq N_{CR}$ 的 CT 个资源源的可用度计算,对计算出的可用度进行排序.可用度计算部分的复杂度主要是 CT 个资源源可用度的计算复杂度,通过算法可见复杂度是 $O(|CT|)$,同时,对计算出的每个可用度进行排序的复杂度最大是 $O(|CT|)$.由于移动节点可以获得资源 CR_j 的数量 CT 源是有限的.因此,复杂度仅随可获得的资源源数目呈线性增长,算法的扩展性良好,可应用于一定规模的移动 P2P 网络.

6 性能分析

正如在文章的第一部分提到的目前协作共享策略中资源发现策略和资源请求调度策略的缺乏和不成熟,因此,本文考虑引用经典的 P2P 资源发现策略和主动队列管理策略作为资源请求和调度策略,对本文提出的移动 P2P 网络的资源协作共享策略中资源预测算法(表示为 RPA)和资源的请求调度策略(表示为 RDA)进行验证和性能分析.其中,

表 1 仿真中使用的参数列表

参数	值
Network area	1000m × 200m
Number of Mobile nodes	100
Gnutella neighbor number	4
Gnutella query TTL	4
Chord/Gnutella Ping message rate	10 packets/s
Chord finger table updating time	5s

本文利用了文献 [13] 中采用的方法和仿真环境,使用 NS-2 作为仿真的平台,并以表 1 的内容作为仿真环境的参数.每个移动节点能够提供 5 个不同的资源,移动节点的移动模式符合随机停靠点(random waypoint)模型.该模型有两个重要参数:移动速度和停留时间.其中移动节点移动速度为 $0 \sim 8\text{m/s}$ 之间,两个移动之间移动节点停留 5s.

对于已有的 P2P 资源发现策略主要分为下面的两种.第一,早期设计的协议有 Napster 和 Gnutella.第二,基于分布式哈希表的算法比如 Chord.本文对上述提到的两种 P2P 资源发现策略进行分析和改进,选择在 Gnutella 策略的基础上,加入了资源预测算法 - RPA 部

分,验证本文提出的资源协作共享策略。

从图 5(a) 可以看到,随着节点移动速度的加快,Chord 不适应移动网络的特征凸现,命中率下降尤其明显.同时,在节点移动速度较快时,网络的移动性特征越发明显,仅凭借洪泛的方式来不能换来查询的高命中率.而 Gnutella_RPA 策略由于能够对可用资源的可用性进行预测分析,优势较突出,能够提高资源查找的命中率.

图 5(b) 展示了由于在 Gnutella 协议中,节点会转发收到的查询包到所有的邻居节点,而 Chord 仅仅依靠查询节点发出的查询包,非常容易发生包丢失而影响查询命中率. Gnutella 使用洪泛的方式查找资源, TTL 值的限定会限定了搜索的区域.而且,资源查找使用洪泛的方式,容易造成网络的拥塞和网络延迟,会引起查询包被阻塞和丢失的现象.正如图 5(b) 中当查询数目增长时,命中率下降.而 Gnutella_RPA 在 Gnutella 协议的基础上,分析目前可使用的资源源的可用度,合理利用了过去的和新搜索到的资源源信息,加入基于可靠性理论的资源预测,增加了资源查找的命中率.并且从图中可以发现,当查询数目增大时,效果尤其明显.

由于 Gnutella_RPA 能保证资源的请求者的资源请求能准确、迅速地提交给资源的提供者.因此,减少查询的时间,提高了搜寻的响应时间,从图 5(c) 可以看到, Gnutella_RPA 的性能提升非常明显.最后,图 5(d) 可以看到,尽管 Gnutella 和 Gnutella_RPA 的发送的消息数目要多于 Chord 协议,但能够获得高命中率.同时, Chord 定期更新和节点退出进入时,也会带来高开销.除此外, Gnutella 协议在协作的资源查找过程中,网络上的消息被复制,产生了消息的冗余,而 Gnutella_RPA 协议的高命中率减少了为查找而产生的消息冗余,也减少了网路中的消息数目.

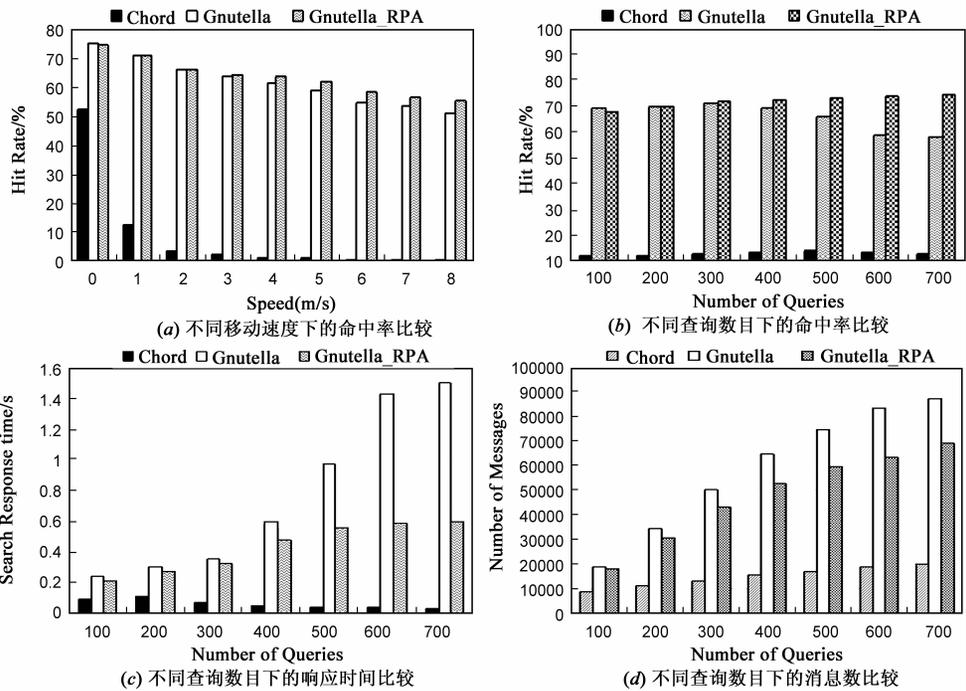


图5 资源发现策略比较

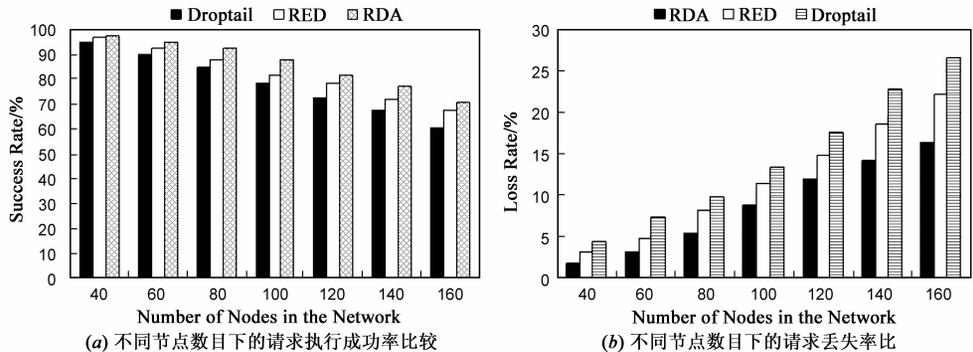


图6 资源请求调度策略比较

如在文献[14]中提到的主动队列管理策略分为 RED^[15]等,或文献[16]中比较的 Droptail 策略,是目前最常用的队列管理策略,用于管理排队系统中的队列分组.首先,从图 6(a) 可以看到,由于简单的尾部丢弃-Droptail 策略对队列处理显得过于简单,对所有请求保持同样的处理原则.而随机早期丢弃-RED 策略来说,同样存在缺乏公平性和分层次性优先级的缺陷.而对于基于排队论的资源队列调度策略-RDA,具有了层次性,简单性,综合性等优势,因此, RDA 策略的节点请求任务执行成功率最高, RDA 策略的资源请求丢失率也小于 Droptail 和 RED(见图 6(b)).当节点和发出请求增加时,策略优势更加突出.

7 结束语

本文提出基于可靠性理论请求资源的预测算法,和基于排队论的理论层次型资源调度模型,有效的提高资源协作共享的成功率,合理控制协作资源请求和

响应。

参考文献:

- [1] Chi-Yin Chow, Hong Va Leong, et al. Peer-to-peer cooperative caching in mobile environments[A]. Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04)[C], 2004. 528 – 533.
- [2] Huaping Shen, Mary Suchitha Joseph, Mohan Kumar, et al. PRcInCt: A scheme for cooperative caching in mobile peer-to-peer systems[A]. Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) [C], 2005. 57 – 60.
- [3] Joseph M S, et al. Energy efficient data retrieval and caching in mobile peer-to-peer networks[A]. Proceedings of the 3rd International Conference on Pervasive Computing and Communications Workshops (PerCom'05)[C], 2005. 50 – 54.
- [4] Liangzhong Yin, Guohong Cao. Supporting cooperative caching in Ad Hoc networks[J]. IEEE Transactions On Mobile Computing, 2006, 5(1): 77 – 89.
- [5] Joonho Cho, Seungtaek Oh, et al. Neighbor caching in multi-hop wireless Ad Hoc networks[J]. IEEE Communications Letters, 2003, 7(11): 525 – 527.
- [6] 赵春晓, 王光兴. 使用模糊线性回归的自组网有效洪泛[J]. 计算机学报, 2006, 29(5): 699 – 704.
Zhao Chun-xiao, Wang Guang-xing. Efficient flooding in MANET using fuzzy linear regression[J]. Chinese Journal of Computers, 2006, 29(5): 699 – 704. (in Chinese)
- [7] 冯萍慧, 连一峰, 戴英侠, 等. 基于可靠性理论的分布式系统脆弱性模型[J]. 软件学报, 2006, 17(7): 1633 – 1640.
Feng Ping-hui, Lian Yi-feng, Dai Ying-xia, et al. A vulnerability model of distributed systems based on reliability theory[J]. Journal of Software, 2006, 17(7): 1633 – 1640. (in Chinese)
- [8] Nelson WB. Accelerated life testing-step-stress models and data analysis[J]. IEEE Transactions of Reliability, 1980, 29(2): 103 – 108.
- [9] 唐三平, 赵娟, 陈兴蜀. 一种处理服务器连接的控制模型[J]. 计算机研究与发展, 2002, 39(6): 667 – 671.
Tang San-ping, Zhao Juan, Chen Xing-shu. A control model of dealing with proxies links[J]. Journal of Computer Research and Development, 2002, 39(6): 667 – 671. (in Chinese).
- [10] 陆凤山. 排队论及其应用[M]. 湖南: 湖南科学技术出版社, 1984.
Lu Feng-shan. Queue Theory and Application[M]. Hunan: Hunan Science & Technology Press, 1984. (in Chinese)
- [11] L Buttyan, J P Hubaux. Enforcing service availability in mobile ad-hoc WANS[A]. In Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc)[C]. Boston, MA, USA, August 2000. 87 – 96.

- [12] 王春江, 耿方萍, 刘元安, 等. 一种应用于 Ad Hoc 无线局域网的随机接入协议[J]. 电子学报, 2005, 33(1): 26 – 31.
Wang Chun-jiang, Geng Fang-ping, Liu Yuan-an, et al. A novel random access protocol for Ad Hoc wireless LANS[J]. Acta Electronica Sinica, 2005, 33(1): 26 – 31. (in Chinese)
- [13] Oliveira L B, Siqueira I G, et al. Evaluation of peer-to-peer network content discovery techniques over mobile ad hoc networks[A]. Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)[C], 2005. 51 – 56.
- [14] 刘明, 窦文华, 张鹤颖. 主动队列管理研究综述[J]. 计算机工程, 2006, 32(24): 84 – 86.
Liu Ming, Dou Wen-hua, Zhang He-ying. Survey of active queue management[J]. Computer Engineering, 2006, 32(24): 84 – 86. (in Chinese)
- [15] 张敬轅, 谢剑英, 傅春. 一种基于模糊逻辑的主动队列管理算法[J]. 电子学报, 2002, 30(8): 1246 – 1249.
Zhang Jing-yuan, Xie Jian-ying, Fu Chun. A fuzzy logic approach for active queue management[J]. Acta Electronica Sinica, 2002, 30(8): 1246 – 1249. (in Chinese)
- [16] Black M, et al. Deploying lightweight queue management for improving performance of mobile ad-hoc networks (MANETs)[A]. Proceedings of International conference on Networking and Services (ICNS'06)[C], 2006, 102 – 108.

作者简介:



牛新征 男, 1978 年 5 月生于贵阳, 2009 年在电子科技大学获博士学位, 毕业后在电子科技大学计算机科学与工程学院工作, 主要研究方向为网络计算、网络安全与移动数据库等。

E-mail: xinzhen@126.com



周明天 男, 1939 年生于广西, 教授, 博士生导师, 电子科技大学计算机科学与工程学院, 主要研究方向为分布式计算、信息安全等。

E-mail: mtzhou@uestc.edu.cn

余堃 男, 1969 年生于成都, 博士, 教授, 博士生导师, 电子科技大学计算机科学与工程学院, 主要研究方向为信息安全、移动计算。