

无线网络环境下的缓存策略研究

林 晨^{1,2} 黄 宇³ 金蓓弘¹

(中国科学院软件研究所软件工程技术中心 北京 100190)¹ (中国科学院研究生院 北京 100049)²
(南京大学计算机科学与技术系软件新技术国家重点实验室 南京 210093)³

摘 要 日益普及的无线网络和移动设备为普适的 Internet 接入提供了良好的硬件基础,缓存技术是普适 Internet 场景中人们进行信息共享和分发的关键技术之一。综述了两类无线网络环境,即基于基础设施无线网络环境和移动自组网中的缓存策略,具体包括缓存放置策略、缓存替换策略、缓存一致性维护算法。进而分析、讨论了上述策略的优缺点,并指出了缓存策略的发展趋势。

关键词 无线网络,缓存放置,替换,一致性

Research on Caching Strategies for Wireless Network Environments

LIN Chen^{1,2} HUANG Yu³ JIN Bei-hong¹

(Technology Center of Software Engineering, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)¹

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

(State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology,

Nanjing University, Nanjing 210093, China)³

Abstract The growing wireless networks and mobile devices provided a good basis of the hardware for universal access to the Internet. Caching technology is one of the key technologies which enables people to share and distribute information in pervasive Internet environments. This paper summarizes two types of wireless network environment, that is, caching strategies for the infrastructure-based wireless network and mobile ad hoc network. The advantages and disadvantages of specific strategies, including cache placement strategy, cache replacement strategy and cache consistency maintenance algorithm are analyzed and discussed. This paper also points out the trends of caching strategies.

Keywords Wireless network, Cache placement, Replacement, Consistency maintenance

1 引言

无线通信技术(如无线通信协议 802.11 等)以及移动嵌入式设备(智能手机, PDA 等)制造技术的发展,使得基于移动终端的普适 Internet 接入,即让用户随时随地地接入 Internet 访问信息成为可能。具体而言,用户可以利用基于基础设施无线网络环境(Infrastructure-based Wireless Network)或者利用移动自组网(MANET, Mobile Ad hoc Network)接入 Internet。前者是指用户的手机、PDA、笔记本电脑等移动设备通过基站(Base Station)或是 Internet 接入点(Access Point)接入 Internet;后者是指用户自发地使用手持的移动设备组成移动自组网,并以其中的若干移动设备作为 Internet 接入的网关结点从而接入 Internet。

缓存技术是缓解数据访问的压力,减少网络通讯流量,缩短用户访问延迟的经典手段,它已经在传统的分布式系统中得到了大范围的成功应用。利用缓存系统,可以在用户下次访问相同信息时避免重新访问网络,从而提高用户的数据访问性能。同样,缓存技术也是普适 Internet 场景中人们进行信息共享和分发的关键技术之一。

基本的缓存策略包括缓存放置策略、缓存替换策略、缓存一致性维护策略等。缓存放置算法用于决定网络中的哪些节点应该缓存数据,以及这些节点应该缓存什么样的数据。缓存替换策略用于在缓存已满而需要加入新的对象时,决定如何替换出缓存中的一些对象,从而为新对象腾出空间。缓存一致性维护策略是为了保证缓存系统中数据访问的有效性。

基于基础设施的无线网络环境和 MANET 是当前主流的无线网络环境,图 1 是这两类网络的示意图。本文综述了这两类无线网络环境中的缓存策略,包括缓存放置策略、缓存替换策略、缓存一致性维护算法,最后指出了缓存策略的发展趋势。

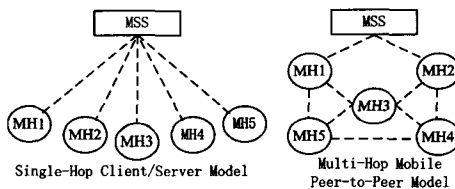


图 1 基于基础设施无线网络环境和 MANET 环境

到稿日期:2008-05-13 本文受 863 国家重点基金项目(2006AA01Z231),国家自然科学基金(60673123)资助。

林 晨 博士研究生,研究方向为分布式计算、软件工程,E-mail:linchen04@otcaix.iscas.ac.cn;黄 宇 博士,讲师,研究方向为普适计算、移动计算、分布式计算、算法设计与分析;金蓓弘 博士,研究员,研究方向为分布式计算、软件工程。

2 缓存放置策略

基于基础设施无线网络环境下的缓存放置基本上是直接使用有线网中的算法,因此我们主要关注 MANET 环境下的缓存放置策略。

针对 MANET 环境,Nuggehalli P 和 Tang B 等分别在文献[1]和文献[2]中对缓存放置问题进行了数学建模,并研究了缓存放置问题的计算复杂度。Nuggehalli P 等在文献[1]中证明了在缓存一致性维护代价的限制下求解最优缓存放置机制是一个 NP 完全问题。Tang B 等在文献[2]中证明了在缓存空间大小的限制下求解最优缓存放置机制同样是一个 NP 完全问题。

Tang X 等在文献[3]中研究了采用基于超时值的缓存一致性维护机制时的缓存放置问题,并给出此时求解最优放置策略的算法。CGA (Centralized Greedy Algorithm)、DGA (Distributed Greedy Algorithm)^[3]、ECHO (Efficient Caching Heuristic Optimization)^[4]是移动自组网环境下的基于贪心算法的缓存放置策略。在移动自组网中,缓存放置问题转化为在连通图中选择哪些节点存放数据会带来更大的收益。上述 3 个算法的主旨是通过获益函数的设置,每一次从某一根节点(即数据源节点)出发,选择能带来最大获益的节点作为缓存放置节点,依次继续,直到所有节点均能够连接上缓存放置节点,如何设置获益函数是这几个放置算法的核心。

在设计缓存放置机制时,除了考虑哪些节点应该放置缓存数据,我们还需要研究缓存节点应该缓存什么样的数据,例如缓存节点可以选择缓存数据本身,也可以选择缓存持有某份数据的节点的标识(例如节点的 ID),即缓存指向某份缓存数据的指针。Yin 等在文献[5]中着重研究了移动节点应如何动态地决定缓存何种类型的数据。作者集中研究了两种缓存机制:CacheData 和 CachePath。在 CacheData 机制中,缓存节点缓存数据本身,而在 CachePath 机制中,缓存节点缓存其它持有缓存数据的节点的 ID,而不是数据本身。作者通过详细的理论分析研究了 CachePath 和 CacheData 这两种机制各自的特点,并根据理论分析的结果设计了启发式算法,由网络中的移动节点动态决定自己应该采用何种缓存放置机制。

3 缓存替换策略

已有的缓存替换算法主要包括 3 大类:基于局部性(Locality-based)、基于缓存替换代价(Cost-based)和基于语义(Semantic-based)^[6,7]。基于局部性的缓存替换算法主要是基于这样的基本假设:在一个缓存系统中,近期被访问的数据很有可能被再次访问。基于替换代价的缓存替换算法的基本原理是对缓存替换带来的各种开销进行建模,并根据事先建立的代价模型综合考虑缓存替换的各种开销,并选择代价最低的替换策略。基于语义的缓存替换算法的设计原理是在缓存数据之间建立语义联系,并优先替换与其它被缓存的数据关系最疏远的数据。当前无线网络环境使用最多的是基于替换代价的缓存替换算法。

基于替换代价的替换策略综合考虑了对象的新旧程度、访问频率、大小等因素,根据这些因素构造函数,通过计算产生代表代价的权值,然后根据权值来进行替换。在无线网络环境中,由于受到设备以及环境的限制,函数的构造不宜复

杂,往往是由几个关键参数构造组成函数。这些参数包括:访问频率,即历史记录中访问对象的频率;对象大小,即被访问对象的实际大小;更新频率,即对缓存对象进行一致性检验的频率;获取延迟,即从数据源端获取对象的延迟时间。目前无线环境下的一些主流的基于替换代价缓存替换算法,比如 SAIU 替换策略^[8]、Min-SAUD 替换策略^[9]、LA2U 替换策略和 LAUD 替换策略^[10]、SXO 替换策略^[4]基本考虑上述参数。前 3 个是基于基础设施无线网络环境下的缓存替换策略,SXO 是 MANET 中的缓存替换策略。

3.1 SAIU 替换策略

SAIU (Stretch Access-rate Inverse Update-frequency) 的主旨是替换那些访问率低、更新率高、空间占用大的数据项。它使用获益函数大小来决定对哪些数据项进行替换。获益函数 $gain(i) = L_i \times A_i / (s_i \times U_i)$, 其中, L_i 是获取数据条目 i 的延迟, A_i 是 i 的访问频率, U_i 是 i 的更新频率, s_i 是 i 的大小。令 $S_i = L_i / (s_i / B)$, 其中 B 是带宽的大小, s_i / B 是 i 所占带宽的比例,称 S_i 为获取 i 所花费的单位时间代价。由此得出 $gain(i)$ 与 $S_i \times A_i / U_i$ 成正比。SAIU 通过对 $gain(i)$ 的计算,把 $gain$ 值大的条目放入缓存,而替换出 $gain$ 值小的条目。SAIU 替换策略是用于无线广播下的客户端的强一致性的缓存替换,考虑了数据更新频率的因素,而且计算复杂度并不高,适合移动客户端使用。

3.2 Min-SAUD 替换策略

Min-SAUD (Minimum Stretch integrated with Access rates, Update frequencies, and cache validation Delay) 是 SAIU 替换策略的一种改进。它在 SAIU 的基础上增加了缓存验证延迟这个参数。Min-SAUD 获益函数如下: $gain(i) = p_i / s_i \times (b_i / (1 + x_i) - v)$, 其中, p_i 是数据条目 i 的访问概率, x_i 是 i 需要更新数据的概率, b_i 是 i 从服务器端获取数据的延迟, v 是 i 缓存有效检验的延迟, s_i 是 i 的大小。通过对 $gain(i)$ 的计算,把 $gain$ 值大的条目放入缓存,而替换出 $gain$ 值小的条目。由于 Min-SAUD 替换策略考虑了缓存验证延迟对替换带来的影响,获益函数的设定更为科学,性能上优于 SAIU。

3.3 LA2U 替换策略和 LAUD 替换策略

LA2U (Least Access-to-Update Ratio)、LAUD (Least Access-to-Update Difference) 这两个算法都是基于访问频率的。在传统的 LFU 算法中,缓存选取近期访问频率低的对象进行替换,而保留那些访问频率高的对象。当考虑到访问对象更新频率的情况时,会发现当某一对象的更新频率很高的时候,缓存这些对象的效率是不高的。因此,访问频率与更新频率成为一对矛盾的因素。选取那些访问频率高而更新频率低的对象,替换那些访问频率低而更新频率高的对象,是解决这一矛盾的方法。

令 r_j^{AU} 是对象 j 在客户端 i 上的访问更新比值, $f_j^{AU} = f_i^j / f_j^j$, 其中 f_i^j 是对象 j 在客户端 i 上的访问频率, f_j^j 是 j 在服务器端的更新频率。 $f_i^j = n_i^j / t$, $f_j^j = n_j^j / t$, 其中 n_i^j 是 t 时间里客户端 i 访问 j 的次数, n_j^j 是服务器端对 j 的更新次数。因此,在 t 时间内, $r_j^{AU} = n_i^j / n_j^j$ 。LA2U 选择那些 r_j^{AU} 大的对象 j 放入缓存,替换那些 r_j^{AU} 小的对象 j 。而 LAUD 与 LA2U 不同的是,LAUD 选取标准是访问更新差值 $d_j^{AU} = f_i^j - f_j^j = (n_i^j - n_j^j) / t$, LAUD 选择那些 d_j^{AU} 大的对象 j 放入缓存,替换那些 d_j^{AU} 小的对象 j 。

LA2U 替换策略和 LAUD 替换策略的优点在于,同时考虑到了对象访问频率和更新频率,这是对传统 LFU 算法的极大补充。另外,LA2U 与 LAUD 的计算复杂度不高,很适合移动环境下客户端缓存的使用。

3.4 SXO 替换策略

SXO 替换策略着重考虑了两个因素:一个是数据对象 i 的大小 s_i ;一个是访问兴趣 Order (d_i),即数据 d_i 被访问的频率排名。令 $k = \text{Order}(d_i)$,表示 d_i 是第 k 个被频繁访问的数据。SXO 替换策略的主旨是寻找那些空间占用大且访问频率排名大的数据对象进行替换,即替换 Size \times Order 值大的数据对象,函数是 $\text{value}(d_i) = s_i \times \text{Order}(d_i)$ 。在 $\text{value}(d_i)$ 的计算中,数据对象的大小 s_i 是已知的,而 $\text{Order}(d_i)$ 无法直接得到,故使用 d_i 的访问频率 a_i 来代替,并通过 $a_i = K/(T - T_{a_i}(k))$ 来计算,其中 T 为当前时间, $T_{a_i}(k)$ 是第 k 个最近访问最多的数据的时间。

SXO 是针对 MANET 下协作缓存的替换策略。由于 MANET 下客户端的计算能力有限,因此采用 SXO 这种计算量小的替换算法有利于整体性能。

4 缓存一致性维护策略

缓存一致性维护策略负责在源数据被更新、而同时又有用户请求缓存数据时如何维护源数据与缓存拷贝之间的一致性。设计缓存一致性维护机制的目标是以最低的代价满足用户对缓存一致性的需求。

缓存一致性维护策略包括缓存一致性模型(Consistency model)设计、缓存一致性维护算法设计。

4.1 缓存一致性模型

设计缓存一致性模型的首要目标是让用户能够准确、便捷地表述自己的缓存一致性需求。传统的缓存一致性模型包括强一致性(Strong consistency)模型和弱一致性(Weak consistency)模型。缓存强一致性是指缓存拷贝始终与源数据保持一致,而缓存弱一致性是指缓存系统尽量(Best effort)维护缓存一致性,但是不能对源数据与缓存拷贝之间的一致性作出任何保证。在动态的网络环境中,用户往往随着周围环境的变化而有不同的缓存一致性需求,为了让用户能灵活地表述自己的一致性需求,Urgaonkar 等提出了 Delta 一致性(Delta consistency)模型^[17]。在 Delta 一致性模型下,用户可以任意指定自己可以接受的源数据和缓存拷贝之间最大的不一致,例如源数据和缓存拷贝之间所允许的最大值的差异程度、缓存拷贝所允许的最大失效时间值等。

除了缓存 Delta 一致性模型之外,另一种让用户灵活设定自己的缓存一致性需求的方法是提供概率一致性模型(Probabilistic consistency),即对于事先设定的某个一致性模型(例如强一致性),让用户指定自己可以接受的某个既定一致性模型被满足的最小概率 p 。

Delta 一致性模型和概率一致性模型让用户从两个正交的维度来设定自己的缓存一致性需求,因而这两个模型可以被组合。组合后的缓存一致性模型能够同时继承 Delta 一致性模型和概率一致性模型的灵活性。PDC(Probabilistic Delta Consistency)模型^[12]组合了已有的 Delta 一致性模型和概率一致性模型,允许用户从两个维度——数据源和缓存拷贝之间的差异和满足既定缓存一致性需求的缓存访问所占的比

例来指定自己的一致性需求。

4.2 缓存一致性维护算法

缓存一致性维护的基本方式包括 Push 策略和 Pull 策略。Push 策略是指数据源节点主动通知缓存节点源数据的更新,而 Pull 策略是指缓存节点从数据源节点或者其它缓存节点处获取数据更新。为了满足用户不同的一致性需求,研究者提出了许多缓存一致性维护算法。支持强一致性模型的缓存一致性维护算法包括 Pull each read^[13],Invalidation^[14],Lease^[15,16]和 UIR-based Cache Invalidation^[17]。这些缓存一致性维护算法只能支持单一的一致性模型,当用户需要灵活地设定自己的一致性需求的时候,这些缓存一致性维护算法一般不可用或者是效率太低。Bhide M 等在文献^[18]中提出了 Push-and-Pull 和 Push-or-Pull 算法来支持 Delta 一致性模型。在文献^[19]中,Zhu 等提出了一种自适应的 Pull 算法以支持概率一致性模型,但这个算法仅仅支持数值数据(例如股票价格等)的一致性维护。

在无线网络环境中,维护强一致性的开销太大,所以大部分的研究工作集中在提供弱一致性上。在有基础设施的移动网中,数据源节点的 Push 请求可以通过数据源节点对所有移动节点的单跳无线广播缓存失效通知(Invalidation report)来实现^[20-23],在该类移动网中缓存一致性维护的触发简化为数据源节点广播时机选择的问题。在有基础设施的移动网中,Push 策略的触发方式主要有两种:同步的和异步的。在同步 Push 策略中,数据源节点每隔固定时间周期性地向移动节点 Push 数据更新;而在异步 Push 策略中,数据源节点在源数据被更新的时候向移动节点 Push 数据更新。有许多基于 Push 的一致性维护算法^[24-26]主要使用同步的 Push 策略,让数据源定期地向缓存节点发送数据更新。为了更好地工作在移动网络环境中,文献^[20]中的一致性维护机制基于缓存命中率 and 数据更新的频率定期调整 Push 的频率。不过,文献^[20]中的 Push 频率调整策略每次仅能将 Push 的频率增加或者减少一个常数值,没有充分考虑周围网络环境的变化程度。在 MANET 中,其网络体系结构与有基础设施的移动网有着本质的不同,节点间采用更为对等的但是通信带宽和稳定性有限的多跳无线通信方式。在 MANET 下,文献^[27]提出面向移动自组网的有状态的缓存一致性维护算法 Greedy Walk-based Selective Push(GWSP),数据源节点通过维护各个缓存拷贝的状态,选择性地向最需要数据更新的节点分发更新,来保证缓存的 Delta 一致性。具体地说,GWSP 算法首先为每个缓存拷贝维护超时值 TTR ,然后根据各个缓存拷贝的状态(TTR 值以及缓存请求率),数据源节点在源数据被更新时,动态地选择需要接收数据更新的缓存节点。

在缓存一致性维护策略 Push 和 Pull 的触发时机决定以后,如何分发数据更新(Push 消息和 Pull 消息),成为一个关键性的问题。

在有基础设施的移动网中,源节点的 Push 请求可以通过数据源节点对所有移动节点的单跳无线广播缓存失效通知来实现。该网络中的缓存失效通知分发算法主要考虑缓存失效通知的内容、缓存失效通知分发机制(包括缓存失效通知的粒度、缓存失效通知分发的参与者)和数据更新历史维护^[24]。在移动自组网中,节点间通信依赖多跳的无线连接,并且数据源节点和缓存节点之间的关系相对有基础设施的移动网来说

更为对等,节点间通信还受到移动自组网中资源(网络带宽、电池能量等)和节点自主移动的限制。

Sailhan F 等在文献[28]中提出了移动自组网协作缓存机制,缓存节点为每个缓存拷贝维护一个 TTR(Time to Refresh)值,缓存节点可以从附近的基站或者是附近的缓存节点处获取源数据更新。在文献[5]提出的缓存机制中,TTR 机制同样被用来标记失效的缓存拷贝,每个缓存节点检查自己转发的数据,以更新自己缓存的数据。但在这种机制中,每个缓存节点被动地检查自己转发的数据,以更新自己的缓存拷贝,因而往往不能满足用户对响应延迟的要求。而引入主动的数据更新协作分发机制可以减少用户获取满足自己一致性需求的缓存拷贝前的响应时间。在文献[29]中,具有更多的电池能量、更多的缓存访问率的节点被选为 Relay Peer。数据源节点向 Relay Peer 分发数据更新,而普通的缓存节点(非 Relay Peer 缓存节点)则向 Relay Peer 发送 Pull 请求以获取数据更新。Relay Peer 的选取对基于 Relay Peer 的缓存一致性维护算法的性能有关键性的影响。然而,在高度动态的移动自组网环境下,如何动态地选出合适的 Relay Peer,是一个有待解决的难题。在文献[30]中提出了 PCC 算法,PCC 算法采用预测机制来动态地决定 Push 和 Pull 的时机。但是,PCC 算法使用简单的单播机制来分发数据更新,因而在数据分发方面的性能有待改进。

结束语 本文对基于基础设施无线网络环境和 MANET 网络环境下的缓存策略从缓存放置策略、缓存替换策略、缓存一致性维护策略几个方面进行了总结。随着移动设备能力的增强,无线网络环境下的应用将越来越丰富,人们对数据访问的速度、准确性的要求将越来越高,作为解决这类问题的关键技术,缓存策略研究将在充分利用各种上下文的基础上进一步深入展开。例如,可以利用网络、设备上下文、开发上下文感知的预取策略,并将其作为缓存替换获益度的评价参数,从而提高客户数据访问的智能性。又例如,基于基础设施的无线网络环境下,一般上行通信与下行相比,其带宽非常有限,所以缓存一致性维护算法的设计都是基于无状态的。而 MANET 下,数据源节点和缓存节点之间是相对更为对等的通信方式,所以可以考虑利用缓存的状态信息设计算法来选择更合适的一致性维护的时机。有线网络环境下的缓存技术目前非常完整成熟,虽然无线网络环境下的缓存技术并不是一个新课题,但它还是有很大方面值得进一步研究。

参 考 文 献

[1] Nuggehalli P, Srinivasan V, Chiasserini C. Energy-efficient Caching Strategies in Ad hoc Wireless Networks // Proc. of Mobile Computing and Networking (MobiCom), 2003

[2] Tang Bing, Gupta H, Das S. Benefit-based Data Caching in Ad Hoc Networks // Proc. of the 14th Intl. Conf. on Network Protocols (ICNP), 2006

[3] Tang X, Chanson S T. Analysis of Replica Placement under Expiration-based Consistency Management. IEEE Trans. on Parallel and Distributed Systems, 2006, 17(11): 1253-1263

[4] Nuggehalli P, Srinivasan V, Chiasserini C F, et al. Efficient Cache Placement in Multi-Hop Wireless Networks. IEEE/ACM Transactions on Networking, 2006, 14(5): 1045-1055

[5] YIN Liangzhong, CAO Gonghong. Supporting Cooperative Caching in Ad Hoc Networks. IEEE Trans. on Mobile Computing, 2006, 5(1): 77-89

[6] Ren Qun, Dunham M H. Using semantic caching to manage location dependent data in mobile computing // Proc. of the 6th Annual Intl. Conf. on Mobile Computing and Networking (MobiCom), 2000

[7] Du Y, Gupta S K S. Cache Management in Wireless and Mobile Computing Environments. Handbook of Mobile Computing, CRC Press, 2004: 337-360

[8] Xu Jiangliang, Hu Qinghong, Lee D L, et al. SAIU: An Efficient Cache Replacement Policy for Wireless On-demand Broadcasts // Proc. of the 9th Intl. Conf on Information and Knowledge Management, 2000

[9] Xu Jianliang, Hu Qinglong, Lee W C, et al. Performance Evaluation of an Optimal Cache Replacement Policy for Wireless Data Dissemination. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(1): 125-139

[10] Chen Hui, Xiao Yang, Shen Xuemin. Update-Based Cache Access and Replacement in Wireless Data Access. IEEE Transactions on Mobile Computing, 2006, 5(12): 1734-1748

[11] Urgaonkar B, Ninan A, Raunak M, S, et al. Maintaining Mutual Consistency for Cached Web Objects // Proc. of the 21st Intl. Conf. on Distributed Computing Systems (ICDCS), 2001

[12] Huang Yu, Cao Jiannong, Wang Zhijun, et al. Achieving Flexible Cache Consistency for Pervasive Internet Access // Proc. of the 5th Annual IEEE Intl. Conf. on Pervasive Computing and Communications (PerCom), 2007

[13] Howard J, Kazar M, et al. Scale and Performance in a Distributed File System. ACM Trans. on Computer Systems, 1988, 6(1): 51-81

[14] Cao Pei, Liu Chengjie. Maintaining Strong Cache Consistency in the World Wide Web. IEEE. Trans. on Computers, 1998, 47(4): 445-457

[15] Gray C, Cheriton D. Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency. ACM SIGOPS Operating Systems Review, 1989, 23(5): 202-210

[16] Duvvuri V, Shenoy P, Tewari R. Adaptive Leases: A Strong Consistency Mechanism for the World Wide Web // IEEE Trans. on Knowledge and Data Engineering, 2003, 15(5): 1266-1276

[17] Cao Gonghong. Proactive Power-aware Cache Management for Mobile Computing Systems. IEEE Trans. on Computers, 2002, 51(6): 608-621

[18] Bhide M, Deolasee P, Katkar A, et al. Adaptive Push-Pull: Disseminating Dynamic Web Data. IEEE. Trans. on Computers, June 2002, 51(6): 652-668

[19] Zhu Shanzhong, Ravishankar C. Stochastic Consistency and Scalable Pull-Based Caching for Erratic Data Stream Sources // Proc. of the 30th VLDB Conf. 2004

[20] Barbara D, Imielinski T. Sleeper and Workaholics: Caching Strategy in Mobile Environments // Proc. of ACM SIGMOD Conf. on Management of Data, 1994

[21] Jing J, Elmagarmid A, Helal A, et al. Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments. Mobile Networks and Applications, 1997, 2(2): 115-127

- peer lookup service for internet applications//Proceedings of the ACM SIGCOMM '01 Conference. San Diego, California, August 2001; 149-160
- [4] Gupta R, Somani A K. Reputation Management Framework and Its Use as Currency in Large-scale Peer-to-Peer Networks // Fourth International Conference on Peer-to-Peer Computing (P2P'04). 2004
- [5] Dewan P, Dasgupta P. Securing P2P networks using peer reputations: Is there a silver bullet // IEEE Consumer Communications and Networking Conference (CCNC). 2005
- [6] Marti S, Garcia-Molina H. Limited Reputation Sharing in P2P Systems // ACM Conference on Electronic Commerce (EC'04). 2004
- [7] Buchegger S, Le Boudec Jean-Yves. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. P2PEcon, 2004
- [8] Jaramillo J J , Srikant R. DARWIN : Distributed and Adaptive Reputation mechanism for WIREless ad-hoc Networks. MobiCom'07. Montréal, Québec, Canada, September 2007
- [9] Olmedilla D, Rana O, Matthews B, et al. Security and trust issues in semantic grids // Proceedings of the Dagstuhl Seminar. Semantic Grid: The Convergence of Technologies, 2005, 05271
- [10] Aringhieri R, Damiani E, Paraboschi S, et al. Fuzzy Techniques for Trust and Reputation Management in Anonymous Peer-to-Peer Systems. Journal of the American Society for Information Science and Technology, 2006, 57(4)
- [11] Lamsal. Understanding trust and security. <http://www.cs.helsinki.fi/u/lamsal/papers/>
- [12] Romano DM. The Nature of Trust : Conceptual and Operational Clarification. PhD Thesis, Louisiana State University, 2003
- [13] Jøsang A. Trust and Reputation Systems. Foundations of Security Analysis and Design IV, FOSAD 2006/2007 Tutorial Lectures. Springer LNCS 4677. Bertinoro, Italy, September 2007
- [14] Jøsang A, Ismail. A survey of trust and reputation system for online service provision. Decision Support Systems, 2007, 43(2) : 618-644
- [15] Theodorakopoulos G, Baras JS. On Trust models and trust evaluation metrics for ad-hoc networks. IEEE Journal on Selected Areas in Communications, 2006, 24(2) : 318-328
- [16] Theodorakopoulos G. Distributed trust evaluation in ad-hoc networks. MS Thesis. 2004
- [17] Page L, Brin S, Motwani R, et al. The PageRank Citation Ranking: Bringing Order to the Web. Technical report. Stanford Digital Library Technologies Project, 1998
- [18] Ziegler C-N, Lausen G. Spreading Activation Models for Trust Propagation // Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE '04). Taipei, March 2004
- [19] Levien R. Attack Resistant Trust Metrics. PhD thesis. University of California at Berkeley, 2004
- [20] Newsome J. The Sybil Attack in Sensor Networks: Analysis & Defenses // the Third International Symposium on Information Processing in Sensor Networks (IPSN). April 2004
- [21] Wang YuFeng , Hori Yoshiaki . On securing open networks through trust and reputation-architecture, challenges and solutions. Technical Report. Institute of Electronics, Information and Communication Engineers, 2006, 106(340) : 1-6
- [22] Danezis G, Schiener S. On network formation, (Sybil attacks and Reputation systems). <http://dimacs.rutgers.edu/Workshops/InformationSecurity/slides/gamesandreputation.pdf>
- [23] Sun Y, Yu W, Han Z. Information theoretic framework of trust modeling and evaluation for ad hoc networks. IEEE Journal on Selected Areas in Communications, 2006, 24(2) : 674-679
- [24] Li Ruidong, Li Jie, Liu Peng, et al. An Objective Trust Management Framework for Mobile Ad Hoc Networks // Vehicular Technology Conference, 2007. 2007; 56-60
- [25] Jiang Tao, Baras J S. Trust Evaluation in Anarchy: A Case Study on Autonomous Networks // Infocom. Barcelona, Spain, April 2006
- [26] Mundinger J, Jean-Yves. Reputation in Self-organized Communication Systems and Beyond // Proceedings from the 2006 Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communications Systems. Pisa, Italy, 2006
- [27] Mundinger J , Jean - Yves . Analysis of a Reputation System for Mobile Ad-hoc Networks with Liars // Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005, Third International Symposium on. 2005; 41-46
- [28] Mundinger J, Jean-Yves. The Impact of Liars on Reputation in Social Networks. Social Network Analysis: Advances and Empirical Applications Forum, 2005

(上接第 4 页)

- [22] Cai Jun, Tan K, Ooi B. On Incremental Cache Coherency Schemes in Mobile Computing Environment // Proc. of 13th Intl. Conf. Data Engineering. 1997; 114-123
- [23] Wu K, Yu P, Chen M. Energy-Efficient Caching for Wireless Mobile Computing // Proc. of 20th Intl. Conf. Data Engineering. 1996; 336-343
- [24] Tan K, Cai Jun, Ooi B. An Evaluation of Cache Invalidation Strategies in Wireless Environments. IEEE Trans. Parallel and Distributed Systems, Aug. 2001, 12(8) : 789-807
- [25] Lan Jiang, Liu Xiaotao, Shenoy P, et al. Consistency Maintenance in Peer-to-Peer File Sharing Networks // Proc. of the 3rd IEEE Workshop on Internet Applications. 2003
- [26] Tan K, Cai J. Broadcast-based Group Invalidation: An Energy Efficient Cache Invalidation Scheme. Information Sciences, 1997, 100(1) : 229-254
- [27] Huang Yu , Jin Beihong , Cao Jiannong , et al . A Selective Push Algorithm for Cooperative Cache Consistency Maintenance in MANETs // IFIP Intl. Conf. on Embedded and Ubiquitous Computing (EUC), 2007
- [28] Sailhan F, Issarny V. Cooperative Caching in Ad hoc Networks // IEEE Intl. Conf. on Mobile Data Management (MDM). 2003
- [29] Cao Jiannong, Zhang Yang, Xie Li, et al. Consistency of Cooperative Caching in Mobile Peer-to-Peer Systems Over MANET // IEEE Intl. Conf. on Distributed Computing Systems Workshops. 2005
- [30] Huang Yu, Cao Jiannong, Jin Beihong. A Predictive Approach to Achieving Consistency in Cooperative Caching in MANET // Proc. of the 1st Intl. Conf. on Scalable Information Systems, P2PIM workshop session, ACM Intl. Conf. 2006