

泛在末梢环境下均衡多业务的终端聚合机制

穆楠 芮兰兰* 郭少勇 熊翱

(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

摘要: 为了对异构终端进行有效聚合, 解决泛在末梢环境中网络资源和终端能力有限性带来的多用户效益冲突分配问题, 该文提出一种均衡多业务的终端聚合机制(DCMME)。首先以相对熵为均衡性指标设计了均衡服务质量效益函数, 建立了面向多业务的终端聚合问题模型(MODC); 然后, 该机制对问题模型进行降维简化, 提出了基于均衡的终端聚合算法(EDC); 最后, 通过 C++ 和 MATLAB 仿真表明, 在该机制下, 用户均衡性改善了 0.5%~20%, 由此证明该方法很好地权衡和保证了多泛在业务请求下各用户效益。

关键词: 泛在网络; 泛在末梢环境; 均衡多业务的终端聚合机制; 信息熵; 服务质量效益

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2014)09-2145-07

DOI: 10.3724/SP.J.1146.2013.01585

Device Composition Mechanism of Multi-service Equilibrium in Ubiquitous Stub Environments

Mu Nan Rui Lan-lan Guo Shao-yong Xiong Ao

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In ubiquitous stub environments, the limited network resources and device capabilities are always causing the conflict of multi-user utilities allocation. To deal with this problem, a Device Composition Mechanism of Multi-service Equilibrium (DCMME) is proposed to make effective composition with heterogeneous devices. Firstly, based on the equilibrium index of relative entropy, the equilibrium service quality utility function is designed, and furthermore, the model of the Multiservice-Oriented Device Composition (MODC) problem is established. Then, this paper proposes to use dimensionality reduction of MODC in order to produce an Equilibrium-based Device Composition (EDC) algorithm. Finally, the simulation is implemented with C++ programming and MATLAB. The results show that the DCMME improves the equilibrium of multi-user by 0.5%~20% and proves to perform well in balancing and ensuring each user's utility under multi-service circumstance.

Key words: Ubiquitous network; Ubiquitous stub environments; Device Composition Mechanism of Multi-service Equilibrium (DCMME); Information entropy; Service quality utility

1 引言

目前, 以泛在网络为背景的研究广泛开展^[1]。在泛在末梢环境中, 分布式的泛在业务需要多种终端聚合完成^[2], 而网络资源和终端能力有限, 导致难以均衡多用户请求业务时的质量效益。因此, 建立均衡多用户效益的终端聚合机制对泛在末梢环境具有重大意义。在泛在末梢环境下, 面向多业务的终端

聚合问题就是综合网络上上下文和业务上下文^[3], 为每个用户选取执行业务的终端集构成微微网^[4] (Capable Piconet, CP)。在研究问题的过程中, 主要面临两大挑战: 一是资源受限问题, 泛在环境下, 终端设备上的共享资源和通信链路上的共享带宽都是有限的, 不同泛在业务在不同质量等级对资源和带宽的需求量不同; 二是终端属性问题, 如业务距离、响应延时等, 也是影响服务质量的重要因素。因此, 多业务终端聚合必须同时考虑资源和终端的双重约束。

关于多业务下的终端聚合问题已有一些研究。文献[5]提出基于 Agent 协商技术的业务选择机制, 但仅适用于 MANETs 下的 P2P 业务模式, 难以满

2013-10-16 收到, 2014-02-17 改回

国家自然科学基金创新研究群体科学基金(61121061), 国家自然科学基金(61302078, 61271187), 国家 863 计划项目(2011AA 01A102), 教育部博士点基金(201110005110011)和北京高等学校青年英才计划(YETP0476)资助课题

*通信作者: 芮兰兰 llrui@bupt.edu.cn

足树形或图形逻辑复杂的分布式泛在业务的需求。文献[6]提出一种基于经济学理论的业务质量评价模型,但以贪心策略寻求任意时刻的效益最大化,无法保障业务的平滑性,同时没有将业务距离等 QoS 因子考虑在内。文献[7]通过将 Web 服务和软件代理技术集成到一个统一实体来建立规范的业务代理模型,进而提出一种分布式规划算法以实现低成本下的高质量服务。但是,这些方法都只追求系统整体性能的最优,而忽视了内部各用户体验质量间的差异,即用户均衡性。

基于已有研究,本文提出相对熵作为均衡性指标,以解决在资源最大化的前提下尽可能缩小用户差异的问题。首先,基于图的结构抽象出泛在业务模型(Ubiquitous Service Model, USM)和泛在网络模型(Ubiquitous Network Model, UNM),引入均衡性指标和均衡服务质量效益函数,建立面向多业务的终端聚合(Multiservice-Oriented Device Composition, MODC)数学模型。随后,提出均衡多业务的终端聚合机制(Device Composition Mechanism of Multi-service Equilibrium, DCMME),包括将 MODC 降维为 MMKP 问题和基于均衡的终端聚合(Equilibrium-based Device Composition, EDC)算法。最后,通过 C++ 和 MATLAB 仿真验证算法性能。该方法在兼顾用户服务质量系统综合性和内部均衡性方面的表现较优。

2 面向多业务的终端聚合问题

2.1 泛在业务和泛在网络模型

泛在末梢环境下,复杂的泛在业务需要分布式的多终端协同完成。如图 1 采用分层架构来描述一种为用户提供个性化睡音乐午觉的服务场景,在泛在网络层,该场景选取了 PDA, Shuffle, iPod Nano 和空调来构成一个 CP 为用户提供服务。面向多业务的终端聚合问题,就是为同时请求的 n 个用户从 $UG(D,L)$ 中选取支撑 $UG(S^i,A^i)$ 的最佳子图的问题。

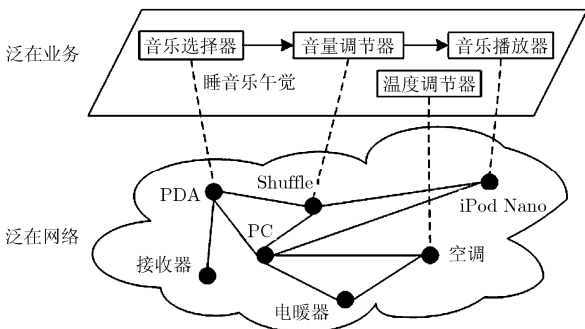


图1 睡音乐午觉服务

泛在业务模型(USM): 每个泛在业务 $i=1,2,\dots,n$ 用一个有向图 $UG(S^i,A^i)$ 表示,包含系统业务集合 $S^i = \{S_v^i | v = 1,2,\dots,v_i\}$ 和业务交互集合 $A^i = S^i \times S^i = \{a_e^i | e = 1,2,\dots,e_i\}$ 。其中, S_v^i 指业务 i 中的第 v 个系统业务, a_e^i 指系统业务间的第 e 条交互。对应的系统业务 S_v^i 在质量等级 q 的资源需求向量为 $r_v^{i,q} = \{r_{v,k}^{i,q} | k = 1,2,\dots,m\}$, 系统业务交互 a_e^i 在质量等级 q 的带宽需求为 $b_e^{i,q}$ 。

泛在网络模型(UNM): 泛在网络用一个无向图 $UG(D,L)$ 表示,包含终端集合 $D = \{d_t | t = 1,2,\dots,T\}$ 和通信链路集合 $L = \{l_s | s = 1,2,\dots,E\}$ 。其中, d_t 指网络中的第 t 个终端, l_s 指终端间的第 s 条通信链路。对应的终端 d_t 的最大资源向量为 $R^t = \{R_k^t | k = 1,2,\dots,m\}$, 通信链路 l_s 上的最大带宽为 B_s 。

2.2 问题模型

面向多业务终端聚合问题的目标是,在充分利用有限网络资源的条件下,使所有用户尽可能都享有最佳的业务体验。因此,首先定义目标函数——均衡服务质量效益函数来对多用户均衡性进行量化。

2.2.1 目标函数 泛在末梢环境下,当选定 CP 来执行业务 i 时,用户服务质量效益可由式(1)^[8]计算。

$$u_c^{i,q} = \alpha_i \cdot (q/q_i) + (1 - \alpha_i) \cdot \sum_{f=1}^{\beta} w_f^i \theta_{c,f}^i \quad (1)$$

其中 q 指 CP 的质量等级, q_i 指质量等级数; β 为终端 QoS 因子数, $\sum_{f=1}^{\beta} w_f^i = 1$, w_f^i 和 $\theta_{c,f}^i$ 分别为第 f 个 QoS 因子的权重和归一化效益值; α_i , $0 \leq \alpha_i \leq 1$ 是质量等级效益的权重,用户可以个性化地对其设置。

对 n 个泛在业务,其综合服务质量效益 ($U = \sum_{i=1}^n u_c^{i,q}$) 不能反映内部用户间的服务差异,因此需引入均衡性指标。传统方法大多采用方差来衡量数据波动性,但方差仅能表示用户效益与均值的偏离程度,而熵能更好地反映各用户效益的等值程度,故本文采用相对熵作为定量描述均衡性的指标。

根据信息熵^[9]原理,先对多业务效益进行归一化,令 $\rho_i = u_c^{i,q}/U$, 则归一化系统熵 $S = -\sum_{i=1}^n (\rho_i \ln \rho_i)$ 。由熵极值性可知,各值完全相等时熵取最大值 $S_{\max} = \ln n$, 则相对效益熵 $S_u = (S_{\max} - S)/S_{\max}$ 。使用相对效益熵 S_u ($S_u \in [0,1]$) 对综合服务质量效益进行修正,定义均衡服务质量效益(简称均衡总效益)函数如式(2)。

$$U_e = (1 - \lambda S_u) \sum_{i=1}^n u_c^{i,q} \quad (2)$$

可知: S_u 越小,各业务效益值越接近,结果越均衡;同时, S_u 越小,与综合效益相乘之后的值就

越大，系统效益就越好。所以，将相对熵作为量化均衡性的指标是合理的。其中， λ 为均衡性指标系数($\lambda S_u \in [0,1]$)。

2.2.2 模型建立 在资源和带宽约束下，以均衡服务质量效益为目标函数，抽象出 MODC 模型如式(3):

$$\left. \begin{aligned} \text{Max } U_e &= (1 - \lambda S_u) \sum_{i=1}^n \sum_{q=1}^{q_i} \sum_{c=1}^{c^{i,q}} u_c^{i,q} \cdot x_c^{i,q} \\ \text{s.t. } \sum_{i=1}^n \sum_{q=1}^{q_i} \sum_{c=1}^{c^{i,q}} \sum_{v=1}^{v_i} r_{v,k}^{i,q} \cdot y_c^{i,q}(v,t) &\leq R_k^t, \quad \forall t, k \\ \sum_{i=1}^n \sum_{q=1}^{q_i} \sum_{c=1}^{c^{i,q}} \sum_{e=1}^{e_i} b_e^{i,q} \cdot z_c^{i,q}(e,s) &\leq B_s, \quad \forall s \\ \sum_{q=1}^{q_i} \sum_{c=1}^{c^{i,q}} x_c^{i,q} &= 1, \quad \forall i \end{aligned} \right\} \quad (3)$$

其中， $c^{i,q}$ 指在质量等级 q 能执行业务 i 的 CP 数。第 1 个约束表示对终端 d_i 上第 k 种资源的需求和不能超过其最大容量 R_k^t ；第 2 个约束表示对链路 l_s 的带宽需求和不能超过其最大容量 B_s ；第 3 个约束表示仅能为每个泛在业务选取一个质量等级和在该质量等级的一个 CP 来执行该业务。0-1 变量 $x_c^{i,q}$ 表示在质量等级 q 的第 c 个候选 CP 被选中来执行业务 i 时为 1；0-1 变量 $y_c^{i,q}(v,t)$ 表示终端 d_i 被选中组成微微网 c 来执行系统业务 S_v^i 时为 1；0-1 变量 $z_c^{i,q}(e,s)$ 表示链路 l_s 被选中为微微网 c 的通信链路来执行业务交互 a_e^i 时为 1。

3 均衡多业务的终端聚合机制

为解决 MODC 模型描述的 NP-hard 问题，本节提出 DCMME 机制。DCMME 主要包含两个阶段：MODC 模型的降维处理及基于启发式的 EDC 算法。在降维处理阶段，主要通过多维约束的合并并将 MODC 转化为 MMKP，使其易于求解；EDC 算法中，ET-HEU(Equilibrium Total-utility Heuristic)以均衡总效益为启发式信息通过多次迭代搜索全局最优解，能够避免陷入局部最优解的情况。

3.1 降维处理

首先，对泛在业务 i ，将所有质量等级下的 CP 合并为 1 维的候选微微网序列，则候选微微网的个数变为 $c_i = \sum_{q=1}^{q_i} c^{i,q}$ 。对合并后新序列中的候选微微网 p ，若 $p = \sum_{q=1}^{q_i-1} c^{i,q} + c (0 \leq c \leq c^{i,q_i})$ ，则服务质量效益为 $\tilde{u}_p^i = u_c^{i,q}$ 。其次，将资源和带宽限制合并为 1 维约束，于是约束的总个数为 $k=mT+E$ 。目标函数中的相对效益熵变为 $\tilde{S}_u = S_u$ ，泛在业务需求和约束条件变化分别如式(4)和式(5)所示。

$$\tilde{r}_{p,\tilde{k}}^i = \left\{ \begin{aligned} \sum_{v=1}^{v_i} r_{v,(\tilde{k} \bmod m)}^{i,q} \cdot y_c^{i,q}(v, \lceil \tilde{k}/m \rceil), & \quad 1 \leq \tilde{k} \leq mT \\ \sum_{e=1}^{e_i} b_e^{i,q} \cdot z_c^{i,q}(e, (\tilde{k} - mT)), & \quad mT + 1 \leq \tilde{k} \leq mT + E \end{aligned} \right. \quad (4)$$

$$\tilde{R}_{\tilde{k}} = \left\{ \begin{aligned} R_{(\tilde{k} \bmod m)}^t, & \quad 1 \leq \tilde{k} \leq mT \\ B_{(\tilde{k} - mT)}, & \quad mT + 1 \leq \tilde{k} \leq mT + E \end{aligned} \right. \quad (5)$$

经过降维处理，MODC 简化为 MMKP 模型如式(6)所示。其中，0-1 变量 \tilde{x}_p^i 表示候选微微网 p 被选取来执行业务 i 时为 1，其他为 0。

$$\left. \begin{aligned} \text{Max } U_e &= (1 - \lambda \tilde{S}_u) \sum_{i=1}^n \sum_{p=1}^{c_i} \tilde{u}_p^i \cdot \tilde{x}_p^i \\ \text{s.t. } \sum_{i=1}^n \sum_{p=1}^{c_i} \tilde{r}_{p,\tilde{k}}^i \cdot \tilde{x}_p^i &\leq \tilde{R}_{\tilde{k}}, \quad \forall \tilde{k} \\ \sum_{p=1}^{c_i} \tilde{x}_p^i &= 1, \quad \forall i \end{aligned} \right\} \quad (6)$$

3.2 EDC 算法

3.2.1 总体流程 在研究的问题中，泛在业务有 3 种状态：等待，就绪和执行。处于等待状态的业务未被分配资源和带宽，处于就绪状态的业务已分配好相应资源和带宽并且待执行，处于执行状态的业务正占用着资源和带宽。网络资源和终端能力的有限性可能会引起多用户请求时的竞争问题，EDC 算法会根据业务层协议^[10](Service Layer Agreement, SLA)赋予每个泛在业务一个优先级(Priority, Pr)，以实现不同级别业务的区分服务^[11]。EDC 算法的总体流程如图 2，主要包括 4 个阶段。

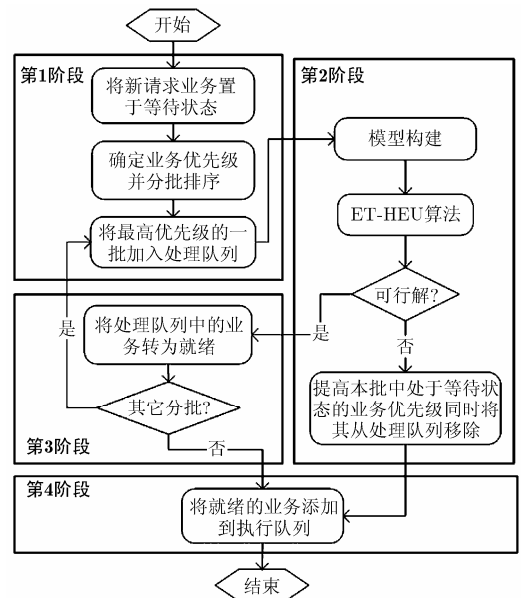


图2 EDC 算法流程图

第 1 阶段：预备阶段

- (1) 将所有新请求的泛在业务置于等待状态。
- (2) 确定处于等待状态的泛在业务的优先级，优先级相同的分为一批，并将批次按优先级递减排序。
- (3) 将优先级最高的一批加入处理队列。

第 2 阶段：终端聚合方案选择阶段

(1) 根据处理队列中泛在业务需求和泛在网络的当前状态，构建 MODC 模型，并降维为 MMKP 模型。

(2) 对 MMKP 模型使用 ET-HEU 算法进行求解。ET-HEU 算法将在 3.2.2 节详细介绍。

(3) 判断是否为泛在业务找到可行聚合方案并根据情况灵活调整优先级。若存在，则进入第 3 阶段；否则提高当前批次中处于等待状态业务的优先级，同时将它们从处理队列中移除。

第 3 阶段：等待执行阶段

(1) 将处理队列中的泛在业务转为就绪状态。

(2) 查看是否还有未处理的分批。若有，则转入预备阶段最后一步重复执行；否则，进入第 4 阶段。

第 4 阶段：执行阶段

将处于就绪状态的泛在业务添加到执行队列，开始执行。

3.2.2 ET-HEU 算法 ET-HEU 为 EDC 算法的核心，其主要思想是：先对 MMKP 问题进行预处理，按照启发式信息找到一个初始解，然后通过一系列升级和降级得到更好的可行解。这里的升级是指一次替换使方案解的均衡总效益增加，降级则反之。其中，方案解由选取的执行泛在业务的 CP 下标构成，是 1 维向量。该算法还使用了聚集资源 (aggregate resources)^[12] 的概念。ET-HEU 算法的详细步骤为：

步骤 1 数据预处理

(1) 根据业务和网络上下文，为泛在业务 $i \in [1, n]$ 选出 c_i 个候选 CP。

(2) 分别对泛在业务 $i \in [1, n]$ ，将其候选 CP 按照服务质量效益值 \tilde{u}_p^i 非递减排列成 n 个有序序列。

步骤 2 寻找初始解

(1) 对泛在业务 $i \in [1, n]$ ，令方案解 $X[i]=1$ ，即选择每个序列中 \tilde{u}_p^i 最小的 CP 作为初始解。

(2) 若解向量 \mathbf{X} 满足式(13)中的约束，那么转入步骤 3；否则解向量 \mathbf{X} 不可行。

(3) 考虑 CP 替换 ($\mathbf{X}' = \mathbf{X}, X'[i] = j$)，满足聚集资源节省量 $\Delta a = \max_{\forall i \in [1, n]; j \in [X[i], c_i]} \Delta a_{ij}$ ，其中， $\Delta a_{ij} = \sum_k (\tilde{r}_{X[i], \tilde{k}}^i - \tilde{r}_{j, \tilde{k}}^i) \cdot C_k / |\mathbf{C}|$ ， \mathbf{C} 为实际资源消耗向量。并且替换后的违背因子 $f'_k = C'_k / \tilde{R}_k$ 与替换前违背因子 f_k 若满足以下 3 条原则，则进行替换

($\mathbf{X} = \mathbf{X}'$) 并转入步骤 2(2)，否则不替换。

(a) 对满足 $f_\alpha = \max_{\forall k \in [1, mT+E]} f_k$ 的 α ， $f'_k < f_k$ ；

(b) 若 $\tilde{k} \neq \alpha$ 且 $f_k > 1$ ， $f'_k \leq f_k$ ；

(c) 若 $\tilde{k} \neq \alpha$ 且 $f_k \leq 1$ ， $f'_k \leq 1$ 。

(4) 初始可行解不存在，算法结束。

步骤 3 升级解向量

(1) 考虑一个升级替换 ($\mathbf{X}' = \mathbf{X}, X'[i] = j$) 具有最大的均衡总效益 $U_e = \max_{\forall i \in [1, n]; j \in [1, c_i]} U_e(i, j)$ 。若替换后解向量 \mathbf{X}' 可行，并且聚集资源节省量 Δa_{ij} 最大，则进行替换 ($\mathbf{X} = \mathbf{X}'$)。

(2) 若步骤 3(1) 中升级替换不存在，考虑替换 ($\mathbf{X}' = \mathbf{X}, X'[i] = j$) 具有最大的单位聚集资源节省量上的均衡总效益增量 $\Delta U_e(i, j) / \Delta a_{ij}$ ，若存在就进行替换 ($\mathbf{X} = \mathbf{X}'$)，并回到步骤 3(1)。

步骤 4 进行至少伴随一次降级的升级

(1) 忽略式(13)中的约束条件，找到一个升级替换 ($\rho = \mathbf{X}, \rho[\xi] = \eta$) 满足单位额外聚集资源上均衡总效益增量 $\Delta U_e(\xi, \eta) / \Delta a'_{\xi\eta}$ 最大，其中 $\Delta a'_{\xi\eta} = \sum_{\tilde{k}} (\tilde{r}_{X[\xi], \tilde{k}}^\xi - \tilde{r}_{\eta, \tilde{k}}^\xi) / (\tilde{R}_k - C_{\tilde{k}})$ 。

(2) 找到一个降级替换 ($\rho' = \rho, \rho'[\xi'] = \eta'$) 满足单位额外聚集资源上均衡总效益增量 $\Delta U_e(\xi', \eta') / \Delta a'_{\xi'\eta'}$ 最小，并且均衡总效益高于步骤 4(1) 升级前，即 $U_e(\xi', \eta') > U_e(\xi, X[\xi'])$ 。

(3) 若步骤 4(2) 中降级替换存在并且 ρ' 可行，那么就进行替换 ($\mathbf{X} = \rho'$) 并转入步骤 3；若降级替换存在但 ρ' 不可行，则转入步骤 4(2) 进行再次降级；否则，恢复到步骤 3 得到的解向量 \mathbf{X} ，算法结束。

3.3 复杂度分析

EDC 算法的复杂度主要由 ET-HEU 算法产生，本节将在最差情况下分析 ET-HEU 算法的时间复杂度。为便于分析，假设所有泛在业务具有相同的优先级 (Pr)、质量等级数 (q) 和候选 CP 数 (l)。首先，步骤 2 若找不到可行解或可行解是效益最高的候选 CP，那么算法复杂度将是最差的，此时步骤 2(3) 将执行 $n(l-1)$ 次，每执行一次就需要对所有违背因子进行检验。其次，当初始可行解为第 1 个候选 CP 时，步骤 3 的复杂度最高，此过程需要不断升级解向量，直到遍历完所有候选 CP。最后，在最坏情况下，步骤 4 进行一次升级后，又需要进行所有可能的降级，然后转入步骤 3 进行所有可能的升级。因此，最差的合并时间复杂度为 $O((mT+E)n^2(l-1)^2)$ 。事实上，每个步骤中的最坏情况都不会同时出现，因此 ET-HEU 算法的实际复杂度通常没有那么高。

4 实验分析

4.1 仿真环境

4.1.1 泛在网络 以智能家居系统为例(如图 3)，泛

在网络空间大小为 $100\text{ m} \times 150\text{ m}$ 。终端资源包括 CPU 频率(IPS)、内存(MB)、硬盘容量 (MB)，资源和带宽初始化值随机设置。终端 QoS 因子包括业务距离、响应延时、有效性和可靠性，4 种 QoS 因子的效益计算公式见文献[8]。

4.1.2 泛在业务 以系统中的具体应用“睡音乐午觉”服务为例(泛在业务逻辑如图 1)。假设多个用户同时发起该泛在业务请求，不同用户各质量等级下对资源和带宽的需求量随机赋值，需求间冲突大小可由实验结果来反映。

使用 C++ 和 MATLAB 仿真实验。为了反映用户不同的质量等级需求，将式(1)中质量等级效益权重 α_i 设置为 0, 0.3, 0.7, 1。为了反映用户对不同 QoS 因子的需求差异，将式(1)中的业务距离权重 ω_{sd} ，响应时延权重 ω_{rd} ，有效性权重 ω_a 和可靠性权重 ω_r 设置如表 1 所示。例如在音乐选择时，若用户请求高品质、流畅的乐曲，那么相应的质量等级效益和响应延时的权重就越大。

4.2 数据分析

通过用户数目、质量等级、收敛性和均衡性指标 4 个方面去分析。

(1)用户数目：从图 4 中可以看出，用户数目为

表 1 QoS 因子权重设置表

权重	方案 1	方案 2	方案 3	方案 4	方案 5
ω_{sd}	0.55	0.15	0.15	0.15	0.25
ω_{rd}	0.15	0.55	0.15	0.15	0.25
ω_a	0.15	0.15	0.55	0.15	0.25
ω_r	0.15	0.15	0.15	0.55	0.25

1 时，不存在资源竞争，此时当 α_i 变化时，总能选出 QoS 因子效益最大的聚合方案。随着用户数目的增加，出现了分配冲突(如图 6 所示)，有限的资源难以同时满足所有用户最高质量等级下的业务需求，一些泛在业务就必须选择其他服务质量效益值较小的 CP 以使更多的用户能得到服务。这样虽然在一定程度上降低了服务质量，但却提高了网络资源利用率。

(2)质量等级：图 5 中子图分别表示存在资源竞争时，相同用户数下， α_i 设置不同终端聚合方式不同。当 $\alpha_i=0$ 时，方案解中 4 种 QoS 因子效益值均为最大。随着 α_i 的增大，QoS 因子效益不断降低，这是由于业务的质量等级因素越来越多地被考虑，终端选择就要牺牲 QoS 因子效益来满足用户较高的质量等级需求。

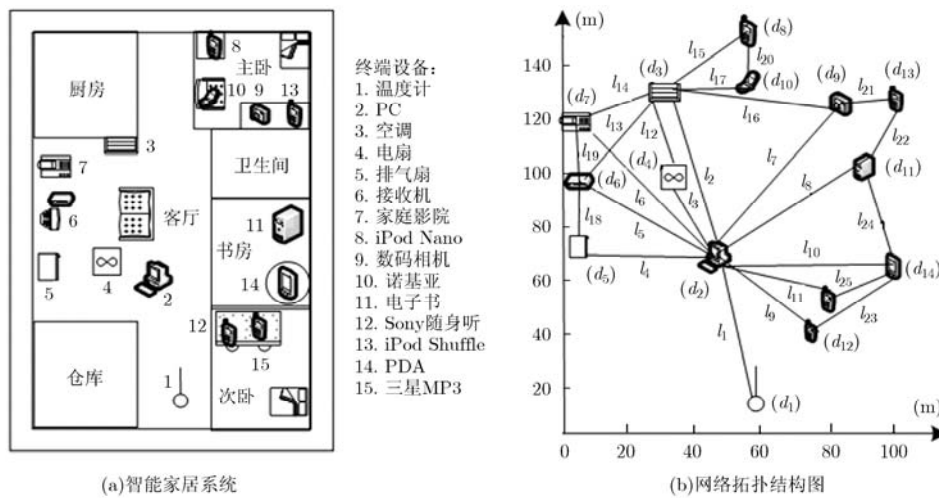


图 3 泛在网络

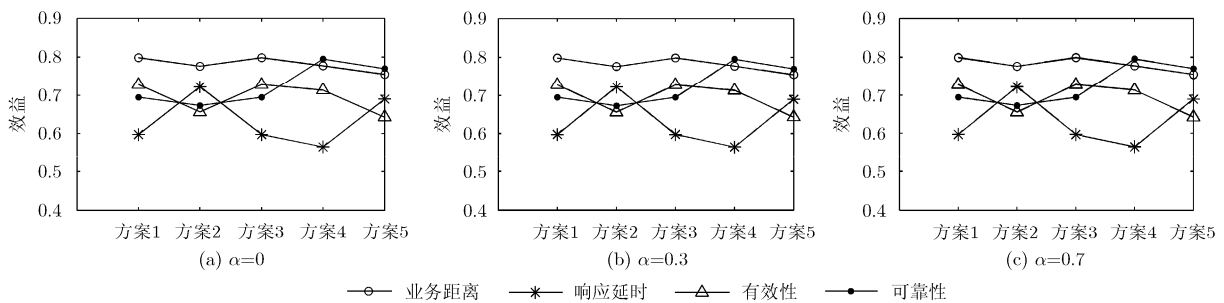


图 4 用户数目为 1

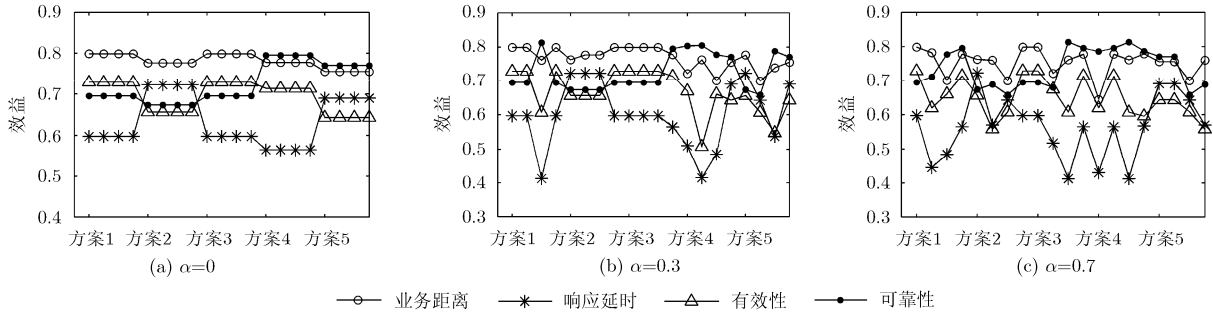


图 5 用户数目为 4

(3)收敛性: 图 6 显示了用户数目为 6 时的搜索过程。随着迭代次数的增加, 曲线呈波动上升趋势, 这是由于启发式搜索总是向着用户均衡总效益增大的方向进行。当迭代次数为 700 左右时, 曲线趋于平稳, 此时算法收敛到最优解, 得到的聚合方案具有最佳的用户均衡总效益, 由此说明算法具有良好的收敛性。

(4)均衡性指标: 当 $\lambda=0$ 时聚合过程未考虑均衡性。图 7 和图 8 反映了多用户请求下有均衡和未均衡的结果。由图 7 易见, 在相同的请求用户数下, 考虑均衡性后方案解的相对熵明显较小。同时由图

8 可以看出, 考虑均衡指标的曲线变化较为平缓, 各用户效益波动减小 0.5%~20%, 聚合效果更佳。

本实验说明 DCMME 在用户规模较小的系统(如: 智能家居系统)中的确更好地兼顾了系统综合性和均衡性, 也可进一步验证该算法同样适用于用户数量更大的场景(如: 智慧校园), 在此限于篇幅不再赘述。

5 结束语

本文从多用户多业务的角度出发, 结合泛在末梢网络中网络环境和终端能力的实际条件, 提出了面向多业务的终端聚合 MODC 模型。针对资源竞争

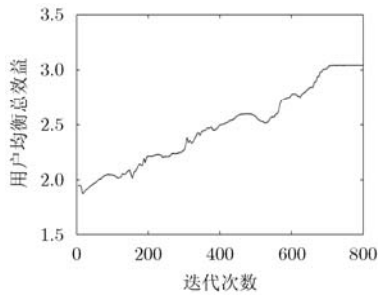


图 6 用户数目为 6 时仿真结果

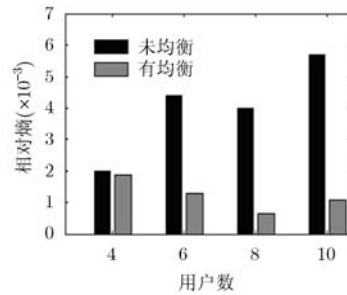


图 7 相对熵比较

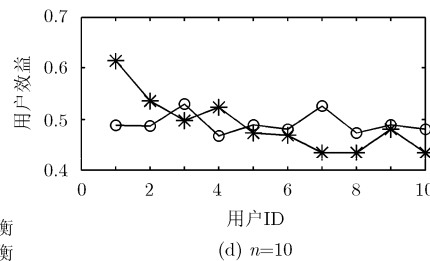
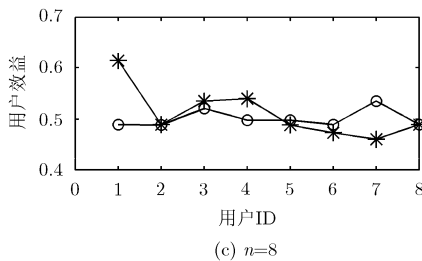
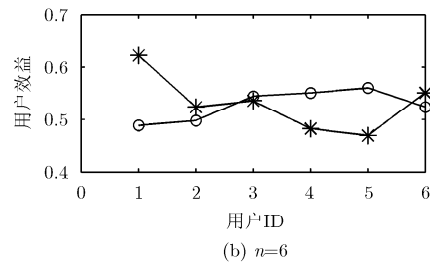
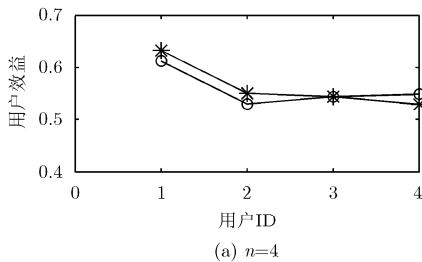


图 8 不同用户数下有均衡与未均衡效果对比

下用户质量差异性问题的，设计了均衡多业务的终端聚合机制 DCMME，该机制在现有研究的基础上，增加了相对效益熵的均衡性指标，并对 MODC 模型进行了降维处理。实验结果表明，EDC 算法在兼顾多用户请求业务时的系统综合性和均衡性方面的表现较优，这对未来研究泛在末梢网络中的业务最优化具有重要意义。下一步本文将在效益指标中增加影响服务质量的 QoS 因子，同时考虑终端移动性所带来的网络拓扑的变化，建立更加综合的目标函数和更加完备的终端聚合机制，从而为泛在末梢环境下的多用户提供最佳的业务体验。

参 考 文 献

- [1] Lu Jun-hong, Wang Chiung-ying, and Hwang Ren-hung. Open framework for distributed context management in ubiquitous environments[J]. *International Journal of Ad Hoc and Ubiquitous Computing*, 2012, 9(4): 199-210.
- [2] Guo Shao-yong, Rui Lan-lan, Qiu Xue-song, et al. An effective cooperation mechanism among multi-devices in ubiquitous network[C]. 2012 8th International Conference on Network and Service Management (CNSM), Las Vegas, NV, USA, 2012: 199-203.
- [3] Eddie Law K J and So S. QoS control framework for content satisfaction in ubiquitous multimedia computing[J]. *Journal of Ambient Intelligence and Humanized Computing*, 2012, 3(2): 103-112.
- [4] Alam M, Albano M, Radwan A, et al. Context based node discovery mechanism for energy efficiency in wireless networks[C]. 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 2012: 6008-6012.
- [5] Yang Yang, Qiu Xue-song, Meng Luo-ming, et al. Negotiation-based stimulating cooperation of selfish nodes in the ad hoc networks[J]. *Ad Hoc and Sensor Wireless Networks*, 2011, 13(3-4): 227-249.
- [6] Li Chun-lin and Li La-yuan. An economics-based negotiation scheme among mobile devices in mobile grid[J]. *Computer Standards and Interfaces*, 2011, 33(3): 220-231.
- [7] Tong Hong-xia, Cao Jian, Zhang Shen-sheng, et al. A distributed algorithm for web service composition based on service agent model[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2011, 22(12): 2008-2021.
- [8] Su Wei-tsung, Liao Ing-hsiu, Lee Kuan-rong, et al. Service-oriented device composition in resource-constrained ubiquitous environments[C]. 2008 IEEE Wireless Communications and Networking Conference (WCNC), Tainan, China, 2008: 3110-3114.
- [9] Henning P and Christian J S. Entropy search for information-efficient global optimization[J]. *The Journal of Machine Learning Research*, 2012, 13(1): 1809-1837.
- [10] Liu Hui, Bu Feng-lin, and Cai Hong-ming. SLA-based service composition model with semantic support[C]. IEEE Asia-Pacific Services Computing Conference (APSCC), Guilin, China, 2012: 374-379.
- [11] Fudzee M F M and Abawayj J H. Management of service level agreement for service-oriented content adaptation platform[J]. *Network and Traffic Engineering in Emerging Distributed Computing Applications*, 2012, 12(2): 21-42.
- [12] Crévits I, Hanafi S, Mansi R, et al. Iterative semi-continuous relaxation heuristics for the multiple-choice multidimensional knapsack problem[J]. *Computers and Operations Research*, 2012, 39(1): 32-41.

穆楠：女，1989年生，硕士生，研究方向为泛在末梢网与终端管理。

芮兰兰：女，1979年生，副教授，硕士生导师，研究方向为通信软件与网络管理。

郭少勇：男，1985年生，博士生，研究方向为终端管理与泛在网。

熊翱：男，1974年生，副教授，硕士生导师，研究方向为网络管理与通信软件。