

主动 Overlay 物联网 QoS 服务路由算法研究

曹怀虎 朱建明

(中央财经大学信息学院 北京 100081)

摘 要 在物联网环境中,服务可能由任意多个结点汇聚而成,而传统的尽力而为的通信服务不能保证服务质量(QoS)。为此,首先提出了主动 Overlay 物联网服务路由逻辑拓扑结构,然后对物联网服务路由问题进行了建模。在此基础上,设计了基于 Agent 和蚁群优化(ACO)的主动 Overlay 物联网 QoS 蚁群服务路由算法。该算法结合移动 Agent 对 ACO 进行了改进,实现了保证 QoS 的服务路径选择。最后从理论上证明了该算法的正确性和收敛性,同时通过仿真实验对该算法的实际性能进行了验证和比较。

关键词 物联网,服务路由,蚁群算法,主动 Overlay,QoS

中图分类号 TP393.01 **文献标识码** A

Research on QoS Service Routing Algorithm for Overlay Internet of Things

CAO Huai-hu ZHU Jian-ming

(School of Information, Central University Finance and Economics, Beijing 100081, China)

Abstract In the internet of things, the service may be provided by any nodes together, and the traditional best-effort communication service can not guarantee quality of service(QoS). This paper established an active services Overlay routing logic topology structure, and made a model of the service routing problem for internet of things. On this basis, this paper proposed an active Overlay QoS service routing optimization algorithm based on Agent and ACO, which was improved with mobile Agent and QoS guaranteed service path selection. It is proved theoretically that the algorithm is correct and convergent. And the actual performance of this algorithm was tested and compared by simulation experiments.

Keywords Internet of things, Service routing, Ant colony optimization, Active Overlay, QoS

1 引言

物联网本质上是一个服务集合体,是由各种异构、动态演化的业务网络(包括互联网、接入网、无线传感器网络等)组成的一个服务网络^[1-3]。在物联网上,服务可能不再只由一个结点提供,而可能由任意多个结点汇聚而成,传统的尽力而为的通信服务不能保证服务质量(QoS)。因此,如何融合现有的异构动态网络,向用户提供具有 QoS 的服务,是物联网将面临的一个重要挑战。对于这一难题,主动 Overlay 技术可能将是一个有效的方法。一个主动 Overlay Network 是由底层物理网络结点的子集所形成的,其能够在不改变底层网络的基础上,支持新的应用和协议。Overlay 结点之间的连接是由虚拟连接提供的,每一个通常是由一个或多个物理连接组成,作为 Overlay 应用通常是构筑在应用层的。它可以有效地使用底层基础设施,同时某些结点具有主动功能,对信息进行处理(如压缩、加密等),最终给高层用户提供具有附加值的服务^[4,5]。

目前的业务网络形成了基本的代价(Cost)模型(包括接纳控制、路由和资源预留等),并依赖这一模型设计了许多有效的路由协议,如“最短路径路由协议(Shortest Path Rou-

ting)”^[6,7]。但是物联网不仅仅只需要一般的信息传输功能,同时还必须通过中间结点对信息流进行处理。这种情况下,因为传输过程中随着流的属性的变化(如重新编码、混合),也在不断影响后续传输的代价,所以,传统的代价模型及其路由算法就不再适用于物联网。为此,本文提出了一个新的代价模型,并给出保证 QoS 的服务路由的解决方案及相关算法。

2 主动 Overlay 物联网服务路由逻辑拓扑结构

所提主动 Overlay 物联网服务路由逻辑拓扑结构如图 1 所示。在底层异构网络之上,通过 Overlay Network,建立一个逻辑的端到端的服务传输基础,再通过服务合约进行路由,用户使用物联网提供商的增值服务。在该逻辑拓扑结构中,基础网络层提供必要的网络通信和路由,包括端到端的运输层传输、点到点交换连接的网络层传输以及点到点直接连接的链路和物理层传输;主动 Overlay 层是一个在基础网络层上构成的虚拟层,它由提供服务的主动结点组成,并分为支持用户端的多样服务的主动结点和进行的 Overlay Network 内部路由的普通主动结点;用户应用层工作于主动覆盖层主动结点上,根据用户的需求,通过 QoS 路由提供用户保证质量的服务。

到稿日期:2011-10-12 返修日期:2012-02-20 本文受国家自然科学基金(60673162),北京自然科学基金(4073041)资助。

曹怀虎(1977—),男,博士,副教授,CCF 高级会员,主要研究方向为网络体系结构、网络计算,E-mail:caohu@163.com;朱建明(1965—),男,博士,教授,CCF 高级会员,主要研究方向为信息安全、系统可生存性。

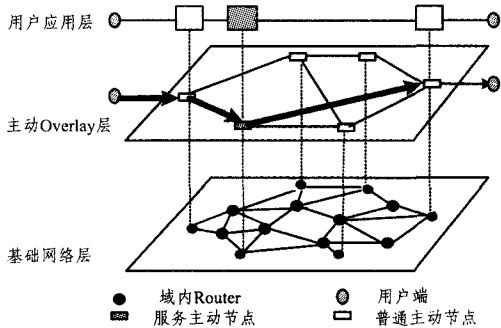


图1 主动 Overlay 物联网服务路由逻辑拓扑结构

3 主动 Overlay 物联网 QoS 服务路由优化模型

QoS 服务路由问题的网络模型可表示为带权图 $G=(V, E, R)$, V 是主动节点的集合, E 是边的集合, $R \subseteq V$ 为提供 Agent 服务的中间结点集合(结点或者子网), 每一条边 $e \in E$ 的代价为 $C(e)$; 主动结点 $r \in R$ 的结点处理代价为 $C(r)$ 。有一个源结点 s , 一个目的结点 d , 中间至少包括一个主动结点 r 加载 Agent。QoS 服务路由目标: 1) 寻找满足 QoS 约束条件的服务路径; 2) 降低寻路的计算开销并提高网络资源的利用率。服务路由优化问题可以表示如下:

$$\min f = \sum_{i \in R} C_i \times x(i) + \sum_{i \in E_p} (A_i \times \prod_{j \in E_p} (x(i) \times \lambda_i)) \quad (1)$$

式中, $x(i) = \begin{cases} 0, & \text{节点 } i \text{ 没有主动服务} \\ 1, & \text{节点 } i \text{ 承担主动服务} \end{cases}$, f 为最小的代价函数, C_i 表示结点提供服务的代价, λ_i 表示节点 i 对流的影响, A_i 表示流速率(单位为带宽)。

4 基于 Agent 和 ACO 的 QoS 服务路由优化算法

4.1 算法基本思想

蚁群算法是对自然界蚂蚁的寻路方式进行模拟而得出的一种仿生算法。目前蚁群算法的应用已经涉及多个领域, 如旅行商问题、指派问题、job-shop 调度问题、图着色问题等^[8-11]。蚁群算法中的运行主体是人工蚂蚁, 因此如何构造合适的人工蚂蚁是非常重要的一项任务。移动 Agent 在性能上充当人工蚂蚁有着许多优越性^[12,13]。本文结合这两种技术, 提出一种基于移动 Agent 的 QoS 蚁群服务路由算法(AQSR)。

4.2 AQSR 算法中移动 Agent 的路由机制

AQSR 算法利用经过改进的蚁群算法中的 3 个规则:

(1) 状态转移规则

根据链路信息素和链路代价来计算得到结点路由表中的链路选取概率数值。

$$p_{ij}^k = \begin{cases} \frac{\alpha \cdot \tau_{ij} + (1-\alpha)\xi_{ij}}{\sum_{i \in L(i)} [\alpha \cdot \tau_{iu} + (1-\alpha)\xi_{iu}]}, & j \in L(i) \\ 0, & j \notin L(i) \end{cases} \quad (2)$$

式中, $L(i)$ 表示第 k 个蚂蚁下一步可以选择的路径的集合, $p_{ij}^k(t)$ 表示第 k 个蚂蚁在 i 点选择经过路径 (i, j) 的概率, $\tau_{ij}(t)$ 表示在 t 时刻路径 (i, j) 上的信息素浓度, $\xi_{ij} = 1/C_{ij}$, C_{ij} 表示经过路径 (i, j) 的花费, α 用来控制信息素和路径长度的重要程度。

(2) 局部信息素刷新规则

当蚂蚁 (Agent) 每前进一条链路时, 通过该规则来刷新链路信息素的浓度。若该链路符合 QoS 约束条件, 则信息素浓度增加, 否则信息素处于挥发状态, 减少了该链路被后来蚂蚁选择的机会^[14-16]。

$$\tau_{ij} \leftarrow (1-\rho_1)\tau_{ij} + \Delta\tau_{ij}^k + \bar{\omega}_k \quad (3)$$

(3) 全局信息素刷新规则

当某蚂蚁成功找到一条符合 QoS 约束条件的路径返回到源结点时, 将采用该规则刷新整个路径的信息素浓度。若该蚂蚁寻路综合代价低, 则路径的信息素增加的浓度将更大。

$$\tau_{s,d} \leftarrow (1-\rho_2)\tau_{s,d} + \frac{\Delta\tau_{s,d}^k}{C_{s,d}} \quad (4)$$

4.3 AQSR 算法的实现

下面给出了 AQSR 算法中应用移动 Agent 进行路由的详细步骤。

若现有一连接请求 $R(B_w, D_w)$, 源结点为 s , 目的结点为 d , 其中 B_w 为 R 要求的最小带宽, D_w 为 R 要求的时延, 则 Agent 在探索路径 $P(N$ 为经过的结点集合) 的过程中要满足以下条件:

$$\begin{aligned} Delay(P) &= \sum_{l \in P} LD_l + \sum_{n \in N} ND_n \leq D_w \\ Bandwidth(P) &= \min_{l \in P} \{LB_l\} \geq B_w \end{aligned} \quad (5)$$

假设系统中每个结点都执行同样的路由选择算法。当某个源结点 s 接收到一个来自目的点 d 的带有参数 $R(B_w, D_w)$ 的连接请求时, 首先使用初始化的信息素值初始化网络中每个结点的信息素表 $adjphertable[\cdot]$ 和 $publicphertable[\cdot]$ 。为了进一步提高算法的速度, 在每个结点应用蚁群算法时, 先剪除该结点中不符合带宽约束条件的所有链路, 此后按照下面的步骤进一步实现。

Step1 源结点共产生 K 个 agent, 每个 agent 都有唯一的标识 ID, 设置并初始化这些 agent 的时间域 $agent. Elapstime=0$ 、结点集合域 $agent. Tabu=\{s\}$ 、目的结点标识域 $agent. destNode=d$ 、跳数 $agent. hopCount=0$ 。

Step2 源结点 s 周期性(不妨设时间周期为 $\Delta(\Delta \ll D_w)$) 地依次发送这些 agent。首先在源结点 s 的信息素表中选择具有最大概率的邻结点(根据式(2))作为 agent 移动的下一个结点(如果存在多个最大的选择概率, 就在其中随机地选取)。

Step3 设第 k 个 agent 到达某个结点 v (假设其前驱结点为 u), 如果 $v \in agent. Tabu$, 则 agent 返回到前一个结点 u , 并从 $agent. Tabu$ 中将 v 结点弹出, 同时通过在局部刷新公式(见式(2))中设置 $\Delta\tau_{uv}^k=0$ 来计算刷新链路 (u, v) 上的信息素强度(此时, 信息素处于挥发状态)。该 agent 在 u 点根据信息素表重新选取下一结点 v' , 并转本步开始处。否则, 进行下一步。

Step4 如果 $agent. Elapstime + t_k(u, v) \leq D_w$, 则利用局部刷新公式刷新链路 (u, v) 上的信息素强度, 同时刷新 agent 数据结构中的相关数据: $agent. Elapstime \leftarrow agent. Elapstime + t_k(u, v)$, $agent. Tabu \leftarrow \{agent. Tabu, v\}$, $agent. hopCount \leftarrow agent. hopCount + 1$ 。另外在链路 (u, v) 预留相应的资源, 进行下一步。否则, 利用局部刷新公式刷新链路 (u, v) 上的信息素强度(设置 $\Delta\tau_{uv}^k=0$, 即信息素处于挥发状态), 在 $agent. flag$ 中设置失败标识, 该 agent 沿原路返回, 同时释放所经过链路

上为自己预留的资源,转到 Step8。

Step5 如果 $v=agent, destNode$, 转 Step7; 否则, 进行下一步。

Step6 以结点 v 作为当前结点, 按照信息素浓度表, 选取下一结点, 转 Step3。

Step7 第 k 个 agent 到达目的点 d 后, 在 $agent, flag$ 中设置成功标识, 此时 $agent, Tabu$ 中的结点构成了符合 QoS 约束条件的路径 P 。之后, 该 agent 根据 $agent, Tabu$ 记录的结点, 反向沿原路返回源结点 s , 并根据携带的数据按全局刷新公式(见式(4))计算刷新路径 P 的信息素强度。

Step8 agent 返回源结点 s 后, 若 $k \geq m$, 且该过程中所有返回 agent 的 $agent, flag$ 中都为失败标识, 则表明网络不能满足连接请求的 QoS 约束条件, 拒绝连接请求。若 $k \geq m$, 且有返回成功标识的 agent, 则存在满足连接请求的路径。若有多个标识成功的 agent 返回, 则比较所有探索到的路径的信息素浓度, 选择出浓度最大的那条路径, 并释放其它路径上预留的资源。该连接开始服务数据传输, 并在服务结束后释放链路上的资源, 算法结束。

4.4 AQSR 算法正确性和收敛性分析

下面的定理给出 AQSR 算法正确性和收敛性证明。不失一般性, 先给出抽象的网络图 $G=(V, E)$, 用 $|V|$ 表示网络的结点数, $|E|$ 表示网络链路数, V 和 E 分别表示网络结点集合和链路集合。

定理 1 考虑网络 $G=(V, E)$ 基于提出的 AQSR 算法, 并满足如下假设:

- 网络 G 存在唯一从源结点 s 到目的结点 d , 且满足时延限制的路径 $P(s, d)$ 。
- 沿着路径 $P(s, d)$ 从源结点 s 到目的结点 d 的跳数为 H 。
- 假设 $P_f(t)$ 是第 k 个 agent 在时间 t 找到路径 $P(s, d)$ 。

则得出以下结论:

- $\lim_{t \rightarrow \infty} P_f(t) = 1$, 其中 x 是一个有限的正实数 ($\lim_{t \rightarrow \infty} P_f(t) = 1$ 说明本文提出的算法在有限的时间内一定可以找到 QoS 路径 $P(s, d)$)。

- 其收敛时间最大不超过 $O(|V|^3)$ 。

证明: 首先, 假设网络的所有结点中度最大的结点的度数为 D , 所有链路中传送时延的最大值为 d_{max} , 所有结点的最大处理时延为 d'_{max} 。如果结点选择了正确的结点 n (即结点 n 在路径 $P(s, d)$ 上), 则 agent 在结点选择正确链路所用的时间最长不超过 $D \times (d_{max} + d'_{max})$ 。下面用数学归纳法进行证明。

当 agent 经过 $H-1$ 跳时, 假设定理成立。 n 由以上的讨论可知, agent 选择下一个结点为目的结点所需要的时间最大不超过 $D \times (d_{max} + d'_{max})$ 。也就是说, 当 $t \leq D \times (d_{max} + d'_{max})$ 时, 可得 $P_f(t) \rightarrow 1$; 又由于 agent 选择某个结点作为其移动的下一个结点独立于选择其它结点, 故定理成立。

假设 agent 经过 $H-Z$ (Z 为整数且 $H \geq Z \geq 2$) 跳时, 定理为真。我们证明 agent 将在有限的时间内找到正确的下一条链路。事实上, agent 找到正确的下一条链路所用的时间最大不超过 $(|E| - H + Z) \times D \times (d_{max} + d'_{max})$, 因为 agent 巡遍某个结点的所有邻结点使用的的时间不超过 $D \times (d_{max} + d'_{max})$, 而 agent 选择的独立性很容易得到。结合以上两步, 定理 1 的第

一部分得证。

从上面的讨论中可以得出, agent 最差的收敛时间为 $|E| \times D \times (d_{max} + d'_{max})$, 而结点的最大度数 D , 链路最大传播时延 $(d_{max} + d'_{max})$, 只要通过恰当的结构设计和接入控制, 就可以使其为常数。所以 $|E| \times D \times (d_{max} + d'_{max}) \cong O(|E| \cdot D) \cong O(|V|^3)$, 定理得证。

5 仿真实验

5.1 实验设计

为了验证所提算法的有效性, 设计了仿真实验进行验证。仿真实验是在 aglet 移动代理系统环境中, 通过 Java 语言编程工具在局域网中来实现的。

图 2 是模拟实验中用到的网络拓扑结构, 它由 13 个结点组成, 每个结点代表一个网络结点, 用一元组 $[node-delay]$ 来表示, $node-delay$ 表示结点的时延; 每条边表示网络结点间的链路连接, 用三元组 (ld, lbw, lc) 来表示, 其中 3 元素分别表示链路时延、链路剩余带宽和链路代价。

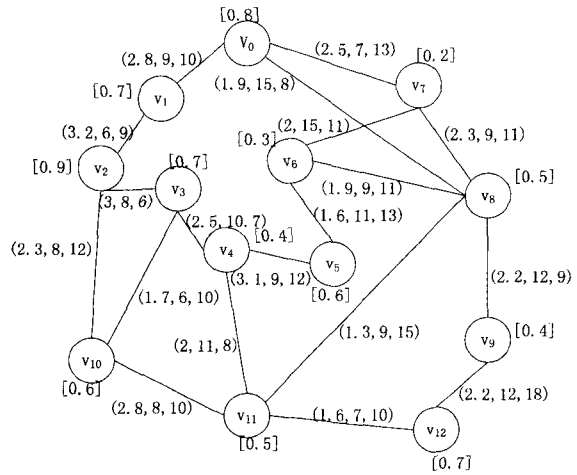


图 2 实验网络拓扑结构

AQSR 算法中的各个参数数值为: $\mu_0 = 0.9, \rho_1 = 0.45, \rho_2 = 0.45, \lambda_1 = 0.35, \lambda_2 = 0.25, \gamma = 0.15$, 各链路信息素初始值为 100, 每次服务连接产生的 agent 数量为 10。

5.2 实验结果及分析

(1) AQSR 算法与遗传算法比较

本实验验证了 AQSR 算法的性能, 并与遗传算法的性能进行了比较。图 3 是 AQSR 与遗传算法求解 QoS 路由的性能曲线图(为了集中比较, 对结果进行了无量纲化处理)。从图 3 可以看出, AQSR 的代价曲线比遗传算法的代价曲线平稳, 而且能更快地找到最优解(或近优解)。时延曲线虽然抖动要略微大一些, 但平稳后的时延小于遗传算法的。对于延迟抖动的曲线, 两者差距不大。

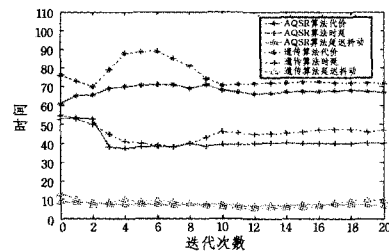


图 3 AQSR 算法与遗传算法的比较

(2) 本实验主要考察 AQSR 算法与最小代价优先 QoS 路由算法在业务会话连接的阻塞率及自适应性方面的差异。图 4、图 5 展示了通过仿真 120 个时间步, AQSR 算法与最小代价算法的性能比较。如图 4 所示, 在 20 步以后, AQSR 算法开始优于最小代价算法, 并在 80 步时达到最优, 这是因为蚁群算法需要一个对网络的适应过程; 另外在网络达到平稳后, 通过改变网络节点的服务请求来观察算法的适应性。图 5 展示了每个节点的服务呼叫数增大为原来的 5 倍时, 网络的波动情况。从图中可以看出, 基于蚁群的 AQSR 算法具有更好的适应性。

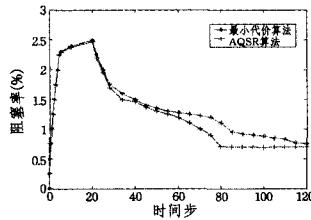


图 4 AQSR 算法与最小代价算法的平均阻塞率比较

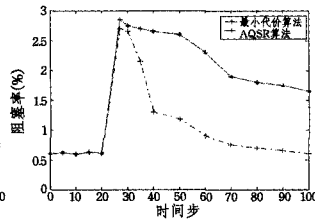


图 5 AQSR 算法与最小代价算法在突发服务需求时的自适应过程比较

结束语 本文首先建立了主动 Overlay 物联网服务路由逻辑拓扑结构, 然后对物联网服务路由问题进行了建模。在此基础上, 提出了一种基于 Agent 和 ACO 的主动 Overlay 物联网 QoS 蚁群服务路由优化算法。该算法结合移动 Agent 对蚁群算法进行了改进, 实现了保证 QoS 的服务路径选择。最后从理论上证明了该算法的正确性和收敛性, 同时通过仿真实验对该算法的实际性能进行了验证和比较, 实验结果表明, 该算法具有较好的性能, 同时能够保证网络负载均衡。

但是对于物联网环境下的 QoS 服务路由这样一个复杂的问题来说, 在本文研究的基础上还有许多值得继续研究的问题, 如有效的 Overlay 物联网的动态维护、资源预留、安全问题、结点物理层信息的获取等。

参考文献

[1] 孙其博, 刘杰, 黎葬, 等. 物联网: 概念、架构与关键技术研究综述[J]. 北京邮电大学学报, 2010, 33(3): 1-9

[2] 孔宁, 李晓东, 罗万明, 等. 物联网资源寻址模型[J]. 软件学报, (上接第 50 页)

[2] Gundavelli S, Leung K, Devarapalli V, et al. IETF RFC 5213 Proxy Mobile IPv6[S]. 2008

[3] Giaretta G. IETF draft-ietf-netlmm-mip-interactions-07 Interactions between PMIPv6 and MIPv6: Scenarios and related issues[S]. 2010

[4] Lee K-W, Seo W-K, Kum D-W, et al. Global Mobility Management Scheme with Interworking between PMIPv6 and MIPv6 [C]// IEEE International Conference on Wireless & Mobile Computing, Networking & Communication. Avignon, France, 2008: 153-158

[5] Yan Z-W, Zhou H-C, Wang H-C, et al. Design and implementation of a hybrid MIPv6/PMIPv6-based mobility management architecture[J]. Mathematical and Computer Modeling, 2011, 53

2010, 21(7): 1657-1666

[3] 李刚, 马修军, 韩燕波, 等. 动态网络环境下的透明服务组合[J]. 计算机学报, 2007, 30(4): 579-587

[4] 吴步丹, 金芝, 赵彬. 面向服务的建模: 一种全过程复用的方法[J]. 计算机学报, 2008, 31(8): 1293-1308

[5] 赵有健, 张铁蕾, 崔勇. 多约束服务质量路由中的路径压缩算法[J]. 计算机学报, 2007, 30(12): 2091-2100

[6] 冀俊忠, 黄振, 刘椿年. 基于变异和信息素扩散的多维背包问题的蚁群算法[J]. 计算机研究与发展, 2009, 46(4): 644-654

[7] Nilssen A. Security and privacy standardization in internet of things[S]. eMatch'09-Future Internet Workshop, Oslo, Norway, September 2009

[8] Wolf W. Cyber-physical Systems[J]. IEEE Computer, 2009, 42(3): 88-89

[9] Giusto D, Iera A, Morabito G, et al. The Internet of Things [M]. New York: Springer, 2010: 102-105

[10] Atzori L, Iera A, Giacomo Morabito. The Internet of Things: A survey[J]. Computer Networks, 2010, 54: 2787-2805

[11] Commission of the European Communities. Internet of Things—An action plan for Europe[R]. Commission of the European Communities, 2009

[12] ITU. ITU Internet Report 2005: The Internet of Things[R]. ITU, 2005

[13] Kranz M, Holleis P, Schmidt A. Embedded Interaction: Interacting with the Internet of Things[J]. IEEE Internet Computing, 2010, 14(2): 46-53

[14] Dorigo M, Maniezzo V, Colomi A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 1996, 26(1): 29-41

[15] Jain J, Gupta R, K T. Ant Colony Algorithm in MANET-Review and Alternate Approach for Further Modification[J]. Information Technology and Mobile Communication Communications in Computer and Information Science, 2011, 147(3): 344-347

[16] Cheng De-qi, Xun Yang-yang, Zhou Ting, et al. An Energy Aware Ant Colony Algorithm for the Routing of Wireless Sensor Networks[J]. Intelligent Computing and Information Science Communications in Computer and Information Science, 2011, 134: 395-401

(3/4): 421-442

[6] Koodli R. IETF RFC 5568[S]. Mobile IPv6 Fast Handovers. 2009

[7] Yokota H, Chowdhury K, Koodli R, et al. IETF RFC 5949[S]. Fast Handovers for Proxy Mobile IPv6, 2010

[8] Haberman B, Hinden R. IETF RFC 5175[S]. IPv6 Router Advertisement Flags Option, 2008

[9] Makaya C, Pierre S. An Analytical Framework for Performance Evaluation of IPv6-Based Mobility Management Protocols[J]. IEEE Transactions on Wireless Communications, 2008, 7(3): 972-983

[10] Sooyeol R, Misum K, Mun Y-S. Enhanced Fast Handovers for Proxy Mobile IPv6[C]//2009 International Conference on Computational Science and Its Applications. Yongin, 2009: 39-43