

NGI 中一种微观经济学模糊 QoS 单播路由机制^{*})

王兴伟 王 琦 黄 敏

(东北大学信息科学与工程学院 沈阳 110004)

摘 要 本文设计并仿真实现了一种微观经济学模糊 QoS 单播路由机制,通过适合隶属度函数对边做出模糊综合评判,通过博弈分析确定网络提供方与用户在边上的效用能否达到 Nash 均衡态,通过基于双向 Dijkstra 算法及中间链表加速方法的启发式选路算法,使得在找到的路径上不仅用户的 QoS 需求得到满足,而且双方的端到端效用达到或接近 Nash 均衡下的 Pareto 最优。仿真结果表明,同基于 Dijkstra 的路由机制和基于模糊塔的 QoS 单播路由机制相比,该机制具有较好的性能。

关键词 下一代互联网,服务质量,单播路由,微观经济学,模糊,Nash 均衡,Pareto 最优

A Microeconomics-based Fuzzy Unicast QoS Routing Scheme in NGI

WANG Xing-Wei WANG Qi HUANG Min

(School of Information Science and Engineering, Northeastern University, Shenyang 110004)

Abstract In this paper, a microeconomics based fuzzy QoS unicast routing scheme is proposed and has been implemented by simulation. It does fuzzy evaluation on the candidate edge comprehensively by introducing adaptability membership degree functions for edge parameters, determines whether the Nash equilibrium between the network provider utility and the user utility has been achieved on the candidate edge by gaming analysis, and attempts to make not only the user QoS requirements satisfied but also the Pareto-optimum under the Nash equilibrium on the network provider end-to-end utility and the user end-to-end utility achieved or approached along the found route by the proposed heuristic route selection algorithm based on the bi-directional Dijkstra algorithm and the intermediate list acceleration method. Simulation results have shown that the performance of the proposed scheme is better than its counterparts based on Dijkstra and fuzzy tower.

Keywords NGI, QoS, Unicast routing, Microeconomics, Fuzzy, Nash equilibrium, Pareto-optimum

1 引言

下一代互联网 NGI(Next-Generation Internet)支持任何地点、任何时间、使用任何类型信息在任何人或对象之间以固定或移动方式进行通信^[1]。为了向用户提供端到端服务质量 QoS(Quality of Service)保证,NGI 的各个组成部分都应当支持 QoS,无线与有线 QoS 应该无缝融合^[2]。由于 NGI 的大规模、异构、动态等特征,终端乃至网络移动的影响,拓扑聚集,信息传递不可避免的延迟及其不确定性,参数值的难以准确计量,特别是无线信道的不可靠与易受环境因素影响,以及 NGI 某些组成部分(如某些私有或专用网络)内部信息的隐蔽等,都必然导致其状态参数难以精确与完全表达。同时,随着 Internet 运营渐趋商业化,付费上网客观上要求以质论价,实现 QoS 计费。然而,网络提供方追求利润最大化,用户希望以最少费用获得最满意服务,双方存在利益冲突。因此,需要调和,实现双赢。如何从路由角度较好地满足上述需求是一个值得研究的问题。本文设计并仿真实现了一种基于微观经济学和模糊数学知识的 QoS 单播路由机制,使网络提供方和用户效用在找到的路由上达到或尽可能接近 Nash 均衡下

的 Pareto 最优^[3]。

2 问题描述

网络模型表示为一连通图 $G(V, E)$, V 是节点集, E 是边集。对 $\forall v_j \in V (j=1, 2, 3, \dots, |V|)$, 考虑如下参数:延迟、延迟抖动和出错率。对 $\forall v_i, v_j \in V$ 之间的边 $e_{ij} \in E (i, j=1, 2, 3, \dots, |V|)$, 考虑如下参数:可用带宽、延迟和出错率。简单起见,本算法把节点参数归并到在路径上其下游边的参数中。这样,边 e_{ij} 上的参数变为:可用带宽 bw_{ij} 、延迟 de_{ij} 、延迟抖动 jt_{ij} 和出错率 ls_{ij} 。

现指定源节点 $v_s \in V$, 目的节点 $v_d \in V$, 要求建立 v_s 与 v_d 之间的一条路径 l_{sd} , 使在 l_{sd} 上网络提供方端到端效用 TW 和用户端到端效用 TU 同时达到或者尽可能接近在 Nash 均衡下的 Pareto 最优,且满足以下 QoS 约束: l_{sd} 的可用带宽不小于 v_d 的带宽要求 bw_{-req} ; l_{sd} 的延迟不大于 v_d 的延迟要求 de_{-req} ; l_{sd} 的延迟抖动不大于 v_d 的延迟抖动要求 jt_{-req} ; l_{sd} 的出错率不大于 v_d 的出错率要求 ls_{-req} 。

数学模型描述如下:

$$TW + TU \rightarrow \max \{ TW + TU \} \quad (1)$$

$$TW \rightarrow \max \{ TW \} \quad (2)$$

^{*})国家自然科学基金资助项目(60673159),新世纪优秀人才支持计划资助项目;高等学校博士学科点专项科研基金资助课题;辽宁省自然科学基金资助项目(20062022);国家发改委 CNGI 示范工程资助项目(CNGI-04-13-2T、CNGI-04-06-2T)和(CNGI-04-15-7A)。王兴伟 博士,教授,博导,主要研究领域为下一代互联网、IP/DWDM 光因特网及移动无线因特网。

$$\begin{aligned}
TU &\rightarrow \max\{TU\} & (3) \\
s. t. \min_{e_{ij} \in l_{sd}} \{bw_{ij}\} &\geq bw_req & (4) \\
\sum_{e_{ij} \in l_{sd}} de_{ij} &\leq de_req & (5) \\
\sum_{e_{ij} \in l_{sd}} jt_{ij} &\leq jt_req & (6) \\
1 - \prod_{e_{ij} \in l_{sd}} (1 - ls_{ij}) &\leq ls_req & (7) \\
TW \text{ 和 } TU \text{ 计算如下:} & & \\
TW &= \sum_{e_{ij} \in l_{sd}} ws_{ij}(x_{ij}) & (8) \\
TU &= \sum_{e_{ij} \in l_{sd}} us_{ij}(x_{ij}) & (9)
\end{aligned}$$

其中, ws_{ij} 表示双方博弈策略确定后网络提供方在边 e_{ij} 上的效用, us_{ij} 表示双方博弈策略确定后用户在边 e_{ij} 上的效用。

上述问题是 NP 完全问题^[4], 本文采用启发式算法求解。

3 路由制描述

本文设计的路由机制由边评判、博弈分析和选路 3 部分组成, 分别描述如下。

3.1 边评判

由于网络状态难以精确描述, 因此引入适合隶属度函数来描述边对用户 QoS 需求的适合程度。

边带宽适合隶属函数定义如下:

$$g_1(bw_{ij}, bw_req) = \begin{cases} 0, & bw_{ij} < bw_req \\ \left(\frac{bw_{ij} - bw_req}{b - bw_req}\right)^k + f_1(bw_{ij}, bw_req), & b > bw_{ij} \geq bw_req \\ 1, & bw_{ij} \geq b \end{cases} \quad (10)$$

其中,

$$f_1(bw_{ij}, bw_req) = \begin{cases} \epsilon, & bw_{ij} = bw_req \\ 0, & \text{其他} \end{cases} \quad (11)$$

边延迟适合隶属函数定义如下:

$$g_2(Jp, de_{ij}, de_req) = \begin{cases} 0, & de_{ij} > de_req \\ 1 - e^{-\left(\frac{de_req - de_{ij}}{\sigma_1}\right)^2} + f_2(Jp, de_{ij}, de_req), & de_{ij} \leq de_req \end{cases} \quad (12)$$

其中,

$$f_2(Jp, de_{ij}, de_req) = \begin{cases} \epsilon, & Jp = 1 \wedge de_{ij} = de_req \\ 0, & \text{其他} \end{cases} \quad (13)$$

边延迟抖动适合隶属函数定义如下:

$$g_3(Jp, jt_{ij}, jt_req) = \begin{cases} 0, & jt_{ij} > jt_req \\ 1 - e^{-\left(\frac{jt_req - jt_{ij}}{\sigma_2}\right)^2} + f_3(Jp, jt_{ij}, jt_req), & jt_{ij} \leq jt_req \end{cases} \quad (14)$$

其中,

$$f_3(Jp, jt_{ij}, jt_req) = \begin{cases} \epsilon, & Jp = 1 \wedge jt_{ij} = jt_req \\ 0, & \text{其他} \end{cases} \quad (15)$$

边出错率适合隶属函数定义如下:

$$g_4(Jp, ls_{ij}, ls_req) = \begin{cases} 0, & ls_{ij} > ls_req \\ 1 - e^{-\left(\frac{ls_req - ls_{ij}}{\sigma_3}\right)^2} + f_4(Jp, ls_{ij}, ls_req), & ls_{ij} \leq ls_req \end{cases} \quad (16)$$

其中,

$$f_4(Jp, ls_{ij}, ls_req) = \begin{cases} \epsilon, & Jp = 1 \wedge ls_{ij} = ls_req \\ 0, & \text{其他} \end{cases} \quad (17)$$

式(10)、(12)、(14)和(16)均属高斯型隶属度函数^[5], 具

有光滑平稳的过渡特性。 f_i ($i=1, 2, 3, 4$) 为修正函数, 用来处理单跳路径的特殊情况; Jp 为一正整数, 表示端到端路径跳数; ϵ 是一个远小于 1 的正纯小数; k, b, σ_1, σ_2 和 σ_3 为正且 $k > 1$ 。使用式(10)、(12)、(14)和(16)得到边的评判矩阵 $R = [g_1, g_2, g_3, g_4]^T$ 。根据应用属性, 给出权值矩阵 $B = [b_1, b_2, b_3, b_4]$, b_1, b_2, b_3, b_4 是带宽、延迟、延迟抖动和出错率在用户 QoS 需求中相对重要性的权值, $0 < b_1, b_2, b_3, b_4 < 1$ 且 $b_1 + b_2 + b_3 + b_4 = 1$, 则用户就其 QoS 需求而言对边的综合评判值 ω 为

$$\omega = B \cdot R \quad (18)$$

显然, ω 值越大, 边越适合用户 QoS 需求。

3.2 博弈分析

如图 1 所示, 假设网络中共有 n_u 个用户, 分为两类: 高端用户和低端用户。前者的 QoS 需求较高, 愿意出高价, 有 s_u 个; 后者的 QoS 需求较低, 只愿意出低价, 有 $n_u - s_u$ 个。 k_u 为网络当前可用资源可满足的用户数(包括高端与低端用户)。显然, $n_u \geq k_u \geq s_u$ 。图 1 给出了网络资源紧张时的阶梯状需求曲线, 纵轴 Y 代表网络承载一单位数据流的价格, 横轴 X 代表数据流大小, c 为网络成本。生产者剩余 P_s 和消费者剩余 C_s 的计算公式如下^[6]:

$$C_s = a_1 \cdot s_u + a_2 \cdot \frac{k_u - s_u}{n_u - s_u} (n_u - s_u) \quad (19)$$

$$P_s = (a_1 + a_2) \cdot s_u + a_3 \cdot (k_u - s_u) \quad (20)$$

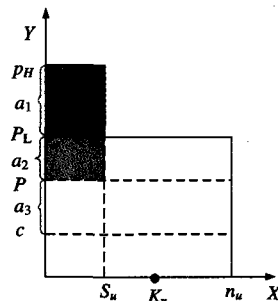


图 1 总需求曲线

根据 g_1 的值, 可以判断边的带宽资源相对 QoS 需求而言是否丰富: 若 $g_1 < h_1$ (h_1 为一个常数且 $0 < h_1 < 1$), 则说明带宽当前供不应求, 网络提供方可以适当提高价格来平抑需求; 若 $h_1 \leq g_1 < h_2$ (h_2 为一个常数且 $0 < h_1 < h_2 < 1$), 则说明带宽当前供求平衡; 若 $g_1 \geq h_2$, 则说明带宽当前供大于求, 网络提供方可以适当降低价格来吸引更多用户。因此, 增加如下价格调节系数 ρ ^[3,6]:

$$\rho = \begin{cases} \rho_1, & g_1 < h_1 \\ 1, & h_1 \leq g_1 < h_2 \\ \rho_2, & g_1 \geq h_2 \end{cases} \quad (21)$$

其中, $\rho_1 > 1, 0 < \rho_2 < 1$, 均取经验值。

网络提供方实际为用户分配的带宽 f_u 计算如下:

$$f_u = \frac{u_c}{\rho P} \quad (22)$$

其中, u_c 为用户所付费用, P 为资源基价。

使用二人博弈, 局中人是网络提供方和用户。网络提供方有二种策略 s_1 和 s_2 , 分别表示愿意和不愿意把当前边提供给用户。用户有二种策略 t_1 和 t_2 , 分别表示愿意和不愿意选择当前边。设网络提供方和用户效用矩阵分别为:

$$U1 = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (23)$$

$$U2 = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad (24)$$

$U1$ 和 $U2$ 的两个行向量分别对应于网络提供方的策略 s_1 和 s_2 , 两个列向量分别对应于用户的策略 t_1 和 t_2 。 $U1$ 的元素 a_{ij} ($i, j=1, 2$) 表示网络提供方选择策略 s_i 与用户选择策略 t_j 时网络提供方在边上的相对效用 (详后), $U2$ 的元素 b_{ij} ($i, j=1, 2$) 表示网络提供方选择策略 s_i 与用户选择策略 t_j 时用户在边上的相对效用 (详后)。

根据 3.1 节, 得到边的综合评判值 ω 。设经验评判值 ω_0 , 如果 $\omega > \omega_0$, 则认为当前边的状况超过用户期望; 如果 $\omega = \omega_0$, 则认为当前边的状况符合用户期望; 如果 $\omega < \omega_0$, 则认为当前边的状况低于用户期望。

$U1$ 和 $U2$ 元素取值如下:

$$U1 = \begin{bmatrix} \frac{(P_s \frac{\omega}{\omega_0} - P_s)}{f_u} & \frac{(P_s \frac{\omega}{\omega_0} - P_s)}{f_u} \\ -\mu \frac{(P_s \frac{\omega}{\omega_0} - P_s)}{f_u} & -\mu \frac{(P_s \frac{\omega}{\omega_0} - P_s)}{f_u} \end{bmatrix} \quad (25)$$

$$U2 = \begin{bmatrix} \frac{(C_s \frac{\omega}{\omega_0} - C_s)}{u_c} & -\mu \frac{(C_s \frac{\omega}{\omega_0} - C_s)}{u_c} \\ \frac{(C_s \frac{\omega}{\omega_0} - C_s)}{u_c} & -\mu \frac{(C_s \frac{\omega}{\omega_0} - C_s)}{u_c} \end{bmatrix} \quad (26)$$

在 $U1$ 中, $\frac{\omega}{\omega_0} P_s / f_u$ 表示网络提供方实际所得效用, P_s / f_u 表示依理论计算网络提供方所得效用, 二者之差作为网络提供方的相对效用, 表示网络提供方所得效用是否超出预期及其程度。 $U1_{21}$ 和 $U1_{22}$ 之所以添加负号是因为如果网络提供方拒绝用户的请求, 则会失去将要获得的效用。 μ 是一个惩罚因子, 表示拒绝愿意使用该网络提供方提供带宽的用户会对用户今后是否愿意使用该网络提供方的带宽造成较大的负面影响, 取大于 1 的值。 同样, 在 $U2$ 中, $\frac{\omega}{\omega_0} C_s / u_c$ 表示用户实际所得效用, C_s / u_c 表示依理论计算用户所得效用, 二者之差作为用户的相对效用, 表示用户所得效用是否超出预期及其程度。 $U2$ 元素中负号和 μ 的含义与 $U1$ 中的一致。 显然, 在 $U1$ 和 $U2$ 中, 如果 a_{ij} 或 b_{ij} 为负值, 则表示网络提供方和 (或) 用户不满意相应的策略对。

如果下列不等式满足:

$$\begin{cases} a_{i^* j^*} \geq a_{ij^*} \\ b_{i^* j^*} \geq b_{ij^*} \end{cases} \quad i=1, 2; j=1, 2 \quad (27)$$

则策略对 $\{s_{i^*}, t_{j^*}\}$ 组成一个非合作纯策略即 Nash 均衡解, i^* 和 j^* 代表某一 i 和 j , s_{i^*} 代表 s_1 或 s_2 , t_{j^*} 代表 t_1 或 t_2 。

3.3 选路

基本思路是为每条边计算一个启发式费用作为权值, 采用双向 Dijkstra 算法及中间链表加速方法^[7]选路。

(1) 启发式费用

通过边评判与博弈分析, 对边 e_{ij} 得到博弈结果, 将其转换成一权值, 用 Ω_{ij} 表示, 其定义如下:

$$\Omega_{ij} = \begin{cases} 1, & \text{Nash 均衡态} \\ > 1, & \text{非 Nash 均衡态} \end{cases} \quad (28)$$

边 e_{ij} 的启发式费用 $Tf_{ij}(\Omega_{ij}, ws_{ij}, us_{ij})$ 定义如下:

$$Tf_{ij}(\Omega_{ij}, ws_{ij}, us_{ij}) = \Omega_{ij} \left(q_1 \frac{1}{ws_{ij}} + q_2 \frac{1}{us_{ij}} \right) \quad (29)$$

在式 (29) 中, Ω_{ij} 反映是否达到 Nash 均衡态对选路的影响; q_1 和 q_2 分别为网络提供方倾斜权值和用户倾斜权值, 反映选路时是否需要向网络提供方或用户效用倾斜以及倾斜程度。 ws_{ij} 和 us_{ij} 分别为网络提供方和用户在边 e_{ij} 上的实际所得效用。 显然, 路径上各边启发式费用之和越小, 越接近在 Nash 均衡下的 Pareto 最优。 因此, 寻路时, 在满足约束 (4)~(7) 的前提下, 通过使路径上各边启发式费用之和最小, 即

$$\text{minimize} \left(\sum_{e_{ij} \in l_{sd}} Tf_{ij}(\Omega_{ij}, ws_{ij}, us_{ij}) \right) \quad (30)$$

而达到或尽可能达到目标 (1)~(3)。

(2) 算法描述

源节点和目的节点分别为 v_s 和 v_d 。 令 pc_1 和 Tc_1 分别表示从 v_s 开始搜索时节点 v 的 pc 标号和 Tc 标号, pc_2 和 Tc_2 分别表示从 v_d 开始搜索时 v 的 pc 标号和 Tc 标号, $pc_1(v)$ 表示在满足约束条件下从 v_s 到该节点的最小启发式费用, $Tc_1(v)$ 表示从 v_s 到该节点的最小启发式费用上限, $pc_2(v)$ 表示在满足约束条件下从 v_d 到该节点的最小启发式费用, $Tc_2(v)$ 表示从 v_d 到该节点的最小启发式费用上限。 S_i 表示第 i 步时具有 pc_1 标号的点的集合, S'_i 表示第 i 步时具有 pc_2 标号的点的集合。 为了在求出从 v_s (或 v_d) 到各节点的最小费用之和的同时, 也求出从 v_s (或 v_d) 到各节点的最小费用路径, 给每个节点 v 分配两个 λ 值 (λ_1 和 λ_2), $\lambda_1(v)$ 表示从 v_s 搜索的最短路径上 v 的前趋节点, $\lambda_2(v)$ 表示从 v_d 搜索的最短路径上 v 的前趋节点。 在算法中止时, 如果 $\lambda_1(v) = m$ (或 $\lambda_2(v) = m$), 则表示从 v_s (或 v_d) 到 v 的最小费用路径上 v 的前一个节点为 v_m ; 如果 $\lambda_1(v) = m'$ (或 $\lambda_2(v) = m'$), 则表示 G 中不存在从 v_s (或 v_d) 到 v 满足约束条件的路径; 如果 $\lambda_1(v) = 0$, 则表示 $v = v_s$, 如果 $\lambda_2(v) = 0$, 则表示 $v = v_d$ 。 另外, $\min bw_1(v_j)$ 为 v_j 到 v_s 的瓶颈带宽值, $de_1(v_j)$ 为 v_j 到 v_s 的总延迟, $jt_1(v_j)$ 为 v_j 到 v_s 的总延迟抖动, $ls_1(v_j)$ 为 v_j 到 v_s 的总出错率, $\min bw_2(v_j)$ 为 v_j 到 v_d 的瓶颈带宽值, $de_2(v_j)$ 为 v_j 到 v_d 的总延迟, $jt_2(v_j)$ 为 v_j 到 v_d 的总延迟抖动, $ls_2(v_j)$ 为 v_j 到 v_d 的总出错率。 Tf_{ij} 、 bw_{ij} 、 de_{ij} 、 jt_{ij} 和 ls_{ij} 分别表示边 e_{ij} 上的启发式费用、可用带宽、延迟、延迟抖动和出错率。 MLS_1 和 MLS_2 分别表示两个方向上的中间链表, 用来保存当前两个方向上的临时节点的序号。

基于文 [7] 中描述的算法, 设计如下路由算法:

Step0 初始化

开始 ($i=0$) 令 $S_0 = \{v_s\}$, $\lambda_1(v_s) = 0$; $S'_0 = \{v_d\}$, $\lambda_2(v_d) = 0$ 。 对每一个 $v \neq v_s$, 令 $Tc_1(v) = +\infty$, $\lambda_1(v) = m'$; 对每一个 $v \neq v_d$, 令 $Tc_2(v) = +\infty$, $\lambda_2(v) = m'$ 。 令 $k_1 = s$, $k_2 = d$ 。 令

$$(1) pc_1(v_{k_1}) = 0, pc_2(v_{k_2}) = 0;$$

$$(2) \min bw_1(v_{k_1}) = +\infty, \min bw_2(v_{k_2}) = +\infty;$$

$$(3) de_1(v_{k_1}) = 0, de_2(v_{k_2}) = 0;$$

$$(4) jt_1(v_{k_1}) = 0, jt_2(v_{k_2}) = 0;$$

$$(5) ls_1(v_{k_1}) = 0, ls_2(v_{k_2}) = 0;$$

把每个 $e_{k_1 j} \in E$ 且 $v_j \notin S_i$ 的节点 v_j 加入 MLS_1 , 并根据式 (10)~(29) 计算出 v_j 与其前趋节点 v_{k_1} 之间链路上的启发式费用 $Tf_{j k_1}$ 的值; 把每个 $e_{k_2 j'} \in E$ 且 $v_{j'} \notin S'_i$ 的节点 $v_{j'}$ 加入 MLS_2 , 并根据式 (10)~(29) 计算出 $v_{j'}$ 与其前趋节点 v_{k_2} 之间链路上的启发式费用 $Tf_{j' k_2}$ 的值。

Step1 标号过程

第一标号条件:

对于每个 $v_j \in MLS_1$, 为了实现目标 (30), 若 $Tc_1(v_j) >$

$pc_1(v_{k_1}) + T_{f_{j_{k_1}}}$, 则

- ① $pc'_1(v_j) = pc_1(v_{k_1}) + T_{f_{j_{k_1}}}$;
- ② $\min bw'_1(v_j) = \min\{\min bw(v_{k_1}), bw_{j_{k_1}}\}$;
- ③ $de'_1(v_j) = de(v_{k_1}) + de_{j_{k_1}}$;
- ④ $jt'_1(v_j) = jt(v_{k_1}) + jt_{j_{k_1}}$;
- ⑤ $ls'_1(v_j) = 1 - (1 - ls(v_{k_1}))(1 - ls_{j_{k_1}})$;

对于每个 $v_j \in MLS_2$, 为了实现目标(30), 若 $T_{C_2}(v_j) >$

$pc_2(v_{k_2}) + T_{f_{j_{k_2}}}$, 则

- ① $pc'_2(v_j) = pc_2(v_{k_2}) + T_{f_{j_{k_2}}}$;
- ② $\min bw'_2(v_j) = \min\{\min bw(v_{k_2}), bw_{j_{k_2}}\}$;
- ③ $de'_2(v_j) = de(v_{k_2}) + de_{j_{k_2}}$;
- ④ $jt'_2(v_j) = jt(v_{k_2}) + jt_{j_{k_2}}$;
- ⑤ $ls'_2(v_j) = 1 - (1 - ls(v_{k_2}))(1 - ls_{j_{k_2}})$;

第二标号条件:

为了满足约束(4)~(7), 如果

- ① $\min\{\min bw'_1(v_j), \min bw'_2(v_j)\} \geq bw_req$;
- ② $de'_1(v_j) + de'_2(v_j) \leq de_req$;
- ③ $jt'_1(v_j) + jt'_2(v_j) \leq jt_req$;
- ④ $1 - (1 - ls'_1(v_j))(1 - ls'_2(v_j)) \leq ls_req$;

则令

- ① $T_{C_1}(v_j) = pc'_1(v_j)$, $T_{C_2}(v_j) = pc'_2(v_j)$;
- ② $\min bw_1(v_j) = \min bw'_1(v_j)$, $\min bw_2(v_j) = \min bw'_2(v_j)$;

(v_j);

- ③ $de_1(v_j) = de'_1(v_j)$, $de_2(v_j) = de'_2(v_j)$;
- ④ $jt_1(v_j) = jt'_1(v_j)$, $jt_2(v_j) = jt'_2(v_j)$;
- ⑤ $ls_1(v_j) = ls'_1(v_j)$, $ls_2(v_j) = ls'_2(v_j)$;
- ⑥ $\lambda_1(v_j) = k_1$, $\lambda_2(v_j) = k_2$;

转 Step2; 否则, 与用户协商: 成功, 则转 Step2; 失败, 则算法结束。

Step2 修改过程

Step2.1 找出当前的工作节点

在 MLS_1 中找出 $T_{C_1}(v)$ 值最小 (即 $\min_{v_j \in MLS_1} \{T_{C_1}(v_j)\}$) 的节点 v_{p_1} , 将 v_{p_1} 的 T_c 标号变为 pc 标号, 即令 $pc_1(v_{p_1}) = T_{C_1}(v_{p_1})$, $S_{i+1} = S_i \cup \{v_{p_1}\}$, $k_1 = p_1$; 在 MLS_2 中找出 $T_{C_2}(v)$ 值最小 (即 $\min_{v_j \in MLS_2} \{T_{C_2}(v_j)\}$) 的节点 v_{p_2} , 将 v_{p_2} 的 T_c 标号变为 pc 标号, 即令 $pc_2(v_{p_2}) = T_{C_2}(v_{p_2})$, $S'_{i+1} = S'_i \cup \{v_{p_2}\}$, $k_2 = p_2$; $i = i + 1$ 。

Step2.2 计算启发式费用

对于每个 $e_{p_1j} \in E$ 且 $v_j \notin S_i$ 的节点 v_j , 根据式(10)~(29)计算出 v_j 与其前趋节点 v_{p_1} 之间链路上的启发式费用 $T_{f_{j_{p_1}}}$ 的值; 对于每个 $e_{k_2j} \in E$ 且 $v_j \notin S'_i$ 的节点 v_j , 根据式(10)~(29)计算出 v_j 与其前趋节点 v_{p_2} 之间链路上的启发式费用 $T_{f_{j_{p_2}}}$ 的值。

Step2.3 修改中间链表

将 v_{p_1} 从 MLS_1 中除去, 同时, 把每个 $e_{p_1j} \in E$ 且 $v_j \notin MLS_1$ 的节点 v_j 加入 MLS_1 ; 将 v_{p_2} 从 MLS_2 中除去, 同时, 把每个 $e_{p_2j} \in E$ 且 $v_j \notin MLS_2$ 的节点 v_j 加入 MLS_2 。

Step2.4 判断是否有相遇节点

遍历 S_i 中的每一个节点, 如果其中的所有节点都不在 S'_i 中 (即 $S_i \cap S'_i = \emptyset$), 则转 Step 1; 否则, 记 S_i 中第一个在 S'_i 中出现的节点为 v_b , 转 Step 2.5。

Step2.5 确定最终路径

如果 $MLS_1 \cap S'_i = \emptyset$, 则把由两个方向搜索到 v_b 的两条路径组合成一条从 v_s 到 v_d 的路径 l_{sd} , 判断 l_{sd} 是否满足 QoS 约束(4)~(7); 若满足, 则输出 l_{sd} 及各项指标, 算法结束; 否则, 与用户协商: 成功, 则输出 l_{sd} 及各项指标, 算法结束; 失败, 则无可行解, 算法结束。

如果 $MLS_1 \cap S'_i \neq \emptyset$, 设 $MLS_1 \cap S'_i = H$ (设 $|H| = q$, 即 H 中含有 q 个节点), 对于 H 中的每一个节点 v_{j_i} ($i = 1, 2, \dots, q$), v_s 到 v_{j_i} 的路径 l_{sj_i} 与 v_{j_i} 到 v_d 的路径 l_{j_id} 构成一条从 v_s 到 v_d 的路径记作 l_i ($i = 1, 2, \dots, q$), 共得到 $q + 1$ 条路径: $l_1, l_2, \dots, l_q, l_{sd}$ (v_s 到 v_b 的路径 l_{sb} 与 v_b 到 v_d 的路径 l_{bd} 构成一条从 v_s 到 v_d 的路径 l_{sd}); 从中找出启发式费用之和最小的路径 l_e ($e = 1, 2, \dots, q + 1$), 判断 l_e 是否满足 QoS 约束(4)~(7); 如果满足, 则输出 l_e 及各项指标, 算法结束; 否则, 与用户协商: 成功, 则输出 l_e 及各项指标, 算法结束; 失败, 则无可行解, 算法结束。

4 性能评价

基于 NS2(Network Simulator 2)^[8,9] 仿真实现了上述路由机制。在多个实际网络拓扑与虚拟网络拓扑上仿真运行本文设计的路由机制、基于模糊塔的 QoS 单播路由机制^[10]以及基于 Dijkstra 的路由机制 (以下分别简称 M、F 和 D 机制), 表 1 给出了三种机制在 CERNET2 拓扑 (拓扑 1)、CERNET 拓扑 (拓扑 2) 和使用 Waxman2 算法^[11] 生成的平均节点度数为 3.5 的拓扑 (拓扑 3) 上的性能比较结果。可以看出, 经济学机制各性能指标都比较好。

表 1 三种机制在三种拓扑上的比较结果

比较项	拓扑和机制	拓扑 1	拓扑 2	拓扑 3
		M : F : D	M : F : D	M : F : D
QoS 路由请求成功率		1.06563 : 1.05553 : 1.00000	1.00876 : 1.00425 : 1.00000	1.05435 : 1.04335 : 1.00000
用户端到端效用		1.02500 : 1.01924 : 1.00000	1.00722 : 1.00293 : 1.00000	1.02823 : 1.02122 : 1.00000
网络提供方端到端效用		1.15136 : 1.13134 : 1.00000	1.03211 : 1.02134 : 1.00000	1.11721 : 1.08876 : 1.00000
综合端到端效用		1.14708 : 1.12816 : 1.00000	1.03219 : 1.02236 : 1.00000	1.14086 : 1.12498 : 1.00000

结束语 本文基于微观经济学和模糊数学知识设计并仿真实现了一种 QoS 单播路由机制。该机制不仅能够适应 NGI 网络状态难以准确描述的实际情况, 而且使得找到的路径在满足用户 QoS 需求的同时网络提供方与用户效用达到或接近 Nash 均衡下的 Pareto 最优, 从路由角度支持效用双赢的实现。仿真研究表明, 本文设计的路由机制是有效的。算法与模型的进一步实用化、原型系统开发及其组播扩展将是今后工作的重点。此外, 本文设计的路由机制尚未考虑由

于用户 QoS 需求难以准确与完全描述而带来的模糊性问题, 这也将是今后工作的重要方向。

参考文献

- 1 Tachikawa K. A perspective on the evolution of mobile communications. IEEE Communications Magazine, 2003, 41 (10): 66~73
- 2 Alam M, Prasad R, Farserotu J R. Quality of service among IP-

based heterogeneous networks. IEEE Personal Communications, 2001, 8 (6): 18~24

- 3 全贤唐, 张健. 经济博弈论. 北京: 机械工业出版社, 2003
- 4 王兴伟, 王志军, 黄敏, 等. 基于服务质量的多媒体通信初始路由建立算法的研究. 计算机学报, 2001, 24 (8): 830~837
- 5 杨纶标. 模糊数学原理及应用. 第三版. 广州: 华南理工大学出版社, 2002
- 6 胡杨梅, 张金水. Internet 拥堵的差别定价策略研究. 数量经济技术经济研究, 2004, 35 (7): 81~85
- 7 靳晓强. 双向 Dijkstra 算法及中间链表加速方法. 计算机仿真, 2004, 21 (9): 78~81

2004, 21 (9): 78~81

- 8 Fall K, Varadhan K. The ns Manual. <http://www.isi.edu/nsnam/ns/doc/>
- 9 徐雷鸣, 庞博, 赵耀. NS 与网络模拟. 北京: 人民邮电出版社, 2003
- 10 Wang Xingwei, Yuan Changqing, Huang Min. A fuzzy-tower-based unicast QoS routing algorithm. In: Proceedings of EUC'04, 2004, 988~997
- 11 Waxman B M. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 1988, 6 (11): 478~489

(上接第 29 页)

所以簇头的稳定性较强, 分布也比较均匀, 并且可以根据节点的分布自适应的调节簇结构, 所以 P 值在 3 种算法中最低; LID 算法居于两者之间, 表明虽然用 LID 算法分簇, 簇头比较稳定, 但其分布还是不够均匀。

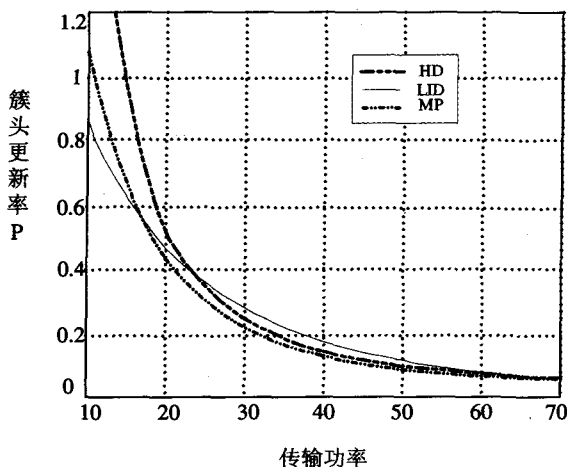


图 4 簇头更新率 P 随节点传输功率变化曲线

最后, 比较一下负载平衡因子 LBF 随节点传输功率变化的情况。模拟结果如图 5 ($LBF = \frac{n_c}{\sum_i (x_i - p)^2}$, 其中 x_i 是簇 i

的成员节点数)。从图中可以看出 3 种算法的变化趋势相同。当节点的传输功率较小时, 每个节点都单独成簇, 此时的 LBF 很大; 当节点的传输功率增大时, 簇头数目减少, 网络的负载平衡能力开始下降; 当节点的传输功率足够大时, 簇的数目几乎为 1, 这时 $LBF \rightarrow \infty$, 此时网络的连通性较好, 但簇头数目过少, 会增加计算开销, 空间重用率降低。从图中还可以看出 3 种算法的区别, MP 算法的 LBF 明显高于其它算法, 这是由于 MP 算法是以移动预测技术为依托, 以链路的连接强度为标准来进行分簇, 簇结构的稳定性强, 省去了许多更新簇头的开销; LID 算法由于产生的簇头数目较多, 更新频率快, 没有考虑负载平衡能力, 所以居中; HD 算法由于簇中成员分布不均匀而导致了较低的 LBF。由此可以看出, MP 算法具有较好的负载平衡能力, 从而可以减少网络拥塞和瓶颈节点出现的概率。

结论 基于分簇算法的分簇网络结构对于提高 Ad hoc 网络的性能有着重要的意义, 已经开始得到广泛的研究, 本文提出的基于移动预测的自适应分簇算法, 可以有效地弥补已有很多分簇算法只是考虑某一方面网络性能的缺点, 多方位

地考虑了影响网络性能的众多因素, 在保持分簇结构的稳定性和提高负载平衡能力等方面都收到了很好的效果。由于分簇结构既有利于移动管理、资源分配和信道接入, 又可以比较方便地采用主动式和按需式优点的混合式路由, 而且目前仍处于研究和模拟阶段, 所以分簇机制在实际网络中将有很大的应用前景。

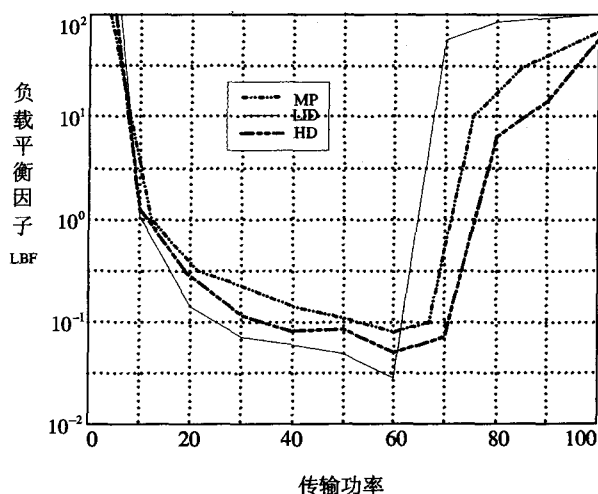


图 5 负载平衡因子 LBF 随节点传输功率变化曲线

参考文献

- 1 Amis A D, Prakash R. Load-Balancing Clusters in Wireless Ad Hoc Networks. In: Proceedings of ASSET 2000, March 2000
- 2 Lin C H R, Gerla M. A Distributed Architecture for Multimedia in Dynamic Wireless Networks [J]. IEEE GLOBECOM, 1995. 1468~1472
- 3 Gerla M, Tsai J T C. Multicluster, Mobile, Multimedia Radio Network [J]. Wireless Networks, 1995, 1(3): 255~265
- 4 Basagni S. Distributed Clustering for Ad Hoc Networks [A]. In: International Symposium on Parallel Architectures, Algorithms and Networks [C]. Perth, June 1999. 310~315
- 5 Su W, Lee S J, Gerla M. Mobility prediction in wireless networks. In: 21st Century Military Communications Conference Proceedings, Vol 1, 2000. 491~495
- 6 Chellappa R, Jennings A, Shenoy N. A comparative study of mobility prediction in fixed wireless networks and mobile ad hoc networks. In: IEEE International Conference on, Vol 2, 2003. 891~895
- 7 王海涛, 张学平. Ad hoc 网络中的分簇算法. 数据通信, 2003(4): 32~35