

## UMIR 系统设计与实现

秦董洪<sup>1,2</sup>, 杨家海<sup>2</sup>, 杨洋<sup>3</sup>, 张辉<sup>2</sup>

(1. 桂林电子科技大学 计算机科学与工程学院, 广西 桂林 541004; 2. 清华大学 网络科学与网络空间研究院, 北京 100084;  
3. 清华大学 计算机科学与技术系, 北京 100084)

**摘要:** 随着互联网应用快速增长, 域间路由系统的选路灵活性和路由质量面临着重大挑战。多路径路由是提高互联网可靠性和顽健性、满足用户选路灵活性的重要方法。基于前期的研究成果以及面向用户定制路由服务的需求, 设计并开发了一套域间多路径路由系统 UMIR。该路由系统包括控制管理、路由发现与计算和路由激励三大功能, 具体由控制管理子系统、路由控制子系统、路由激励子系统等构成。在美国 Utah 大学的 Emulab 网络测试平台上, 对 UMIR 系统进行了实际部署与测试实验, 结果表明该路由系统运行正确, 达到了系统预期的设计目标。

**关键词:** 域间路由; UMIR; 路由系统; 设计与实现

中图分类号: TP393.4

文献标识码: A

文章编号: 1000-436X(2013)Z2-0027-09

## Design and implementation of user-customizing oriented multipath inter-domain routing system

QIN Dong-hong<sup>1,2</sup>, YANG Jia-hai<sup>2</sup>, YANG Yang<sup>3</sup>, ZHANG Hui<sup>2</sup>

(1. School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China;  
2. Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China;  
3. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

**Abstract:** With the rapid growth of Internet applications, the routing flexibility and quality of the inter-domain routing system face some significant challenges. Multipath routing is an important method of improving the reliability and robustness of the Internet. Based on previous researches, orienting user customizing routing service, a multi-path inter-domain routing system named UMIR was developed. The routing system includes three service functions of controlling and management center, route discovery and computing protocol and routing incentive, and it is made of control and management subsystem, routing protocol subsystem and routing incentive subsystems. Based on the Emulab network test platform of University of Utah, the UMIR routing system was deployed and tested. The experimental results show that the routing system is running correctly to achieve our design goals.

**Key words:** interdomain routing; UMIR; routing system; design and implementation

收稿日期: 2013-09-10

基金项目: 国家重点基础研究发展计划(“973”计划)基金资助项目(2009CB320505); 国家科技支撑计划基金资助项目(2008BAH37B05); 国家高技术研究发展计划(“863”计划)基金资助项目(2008AA01A303, 2009AA01Z251); 国家自然科学基金基金资助项目(61170211, 61202356); 教育部博士学科专项基金资助项目(20110002110056); 广西可信软件重点实验室基金资助项目(KX201307)

**Foundation Items:** The National Basic Research Program of China (973 Program) (2009CB320505); The National Science and Technology Supporting Plan of China (2008BAH37B05); The National High Technology Research and Development Program of China (863 Program) (2008AA01A303, 2009AA01Z251); The National Natural Science Foundation of China (61170211, 61202356); The Specialized Research Fund for the Doctoral Program of Higher Education (20110002110056); The Fund of Guangxi Key Laboratory of Trusted Software(KX201307)

## 1 引言

互联网是由数以万计独立的自治系统(AS, autonomous systems)互联构成的分布式系统. AS 内部以优化路由效率为目标, 运行内部网关协议(比如 OSPF、RIP2 和 IS-IS)来规划和管理网络资源; AS 之间以网络可达与路由策略为目标, 运行外部网关协议(如 BGP<sup>[1]</sup>)实现互联互通. 为了确保域间路由协议具有良好的扩展性, BGP 仅为目的网络筛选一条最佳路径, 并将它宣告给邻居对等路由器. 由于 BGP 采用单路径路由模式, 故它存在着可靠性差、无法使用次优路径以及负载均衡支持弱等不足<sup>[1,2]</sup>. 互联网需要其他灵活与高效的新型路由模式.

随着新型互联网应用的快速发展, 很多实时应用如流媒体(streaming media)和数据传输应用如数据中心(data center)对单路径路由模式(BGP)提出了许多新挑战<sup>[1,2]</sup>. 例如有些用户要求多样化和个性化的路由需求, 希望互联网服务提供商 ISP(internet service provider)提供灵活与高效的多路径路由服务. 不同类型的应用对网络性能有着不同的要求: 实时应用偏好低延迟和低抖动的路由, 数据传输偏好高带宽的路由. 传统单路径路由模式不能满足用户路由多样化的需求, 需要高效灵活的多路径路由技术的支持. 研究表明多路径路由具有以下优点<sup>[2,3]</sup>: 1)根据不同应用的性能需求, 用户可以灵活定制路由服务; 2)实现故障避免改善网络传输的可靠性; 3)利用多条路径实现负载均衡; 4)增强通信安全防止出现单路径窃听. 由此可见, 多路径路由可以弥补单路径路由的不足, 它有着重要的应用需求和市场前景, 不仅是现有互联网路由体系的补充, 也是下一代互联网体系的重要组成.

前期研究中, 提出了一种面向用户定制的域间多路径路由(UMIR, user-customizing oriented multipath inter-domain routing)协议<sup>[3,4]</sup>以及路由激励方法<sup>[5,6]</sup>. UMIR 协议基本思想是通过主动收集网络各节点的路由信息, 构建互联网的自组织拓扑, 为定制用户计算满足其要求的路径. 另外, 为了激励各 ISP 部署和提供高质量的路由服务, 提出 2 种路由激励方法, 其核心思想是: 通过收取定制用户的路由服务费用, 然后将费用通过收益分配算法合理公平地分配给路径上各节点. 上述的路由协议和激励方法成为域间多路径路由系统的核心技术. 本文对上述研究成果进行系统实现, 设计并开发了 UMIR 路由系

统. 该路由系统包括控制管理, 路由发现与计算和路由激励三大功能, 具体由控制管理子系统、路由控制子系统、路由激励子系统构成. 然后在美国 Utah 大学的 Emulab 网络测试平台<sup>[7]</sup>上, 对该 UMIR 路由系统进行了部署与测试实验, 结果表明该路由系统运行正确, 达到了系统的预期设计目标.

## 2 UMIR 系统概述

### 2.1 UMIR 协议

为了满足用户的选路灵活性和路径多样化的需求, 提出一种面向用户定制的域间多路径路由协议 UMIR<sup>[3,4,6]</sup>, 满足了互联网域间多路径路由的需求. 针对域间多路径路由存在的扩展性问题和选路灵活性要求, 该协议通过控制节点规模和边(或路段)的数量, 提出了相应的解决方案. 其核心思想是: 根据用户定制请求, 主控节点请求协作节点提供路由信息, 构成一定规模的局部拓扑, 然后进行路由计算. 局部拓扑形成包括了节点选择和节点路段构造过程. 通过选择节点度值较大的节点, 提高了路径多样化的潜力. 节点构造路段时综合考虑了路段的物理属性、策略属性和代价信息<sup>[6]</sup>. 通过控制节点的路段可视度  $R$  和拓扑节点数  $N$  可以调节路由协议的性能与开销. 路由计算过程可以使用多种路由计算算法(例如策略路径算法和最短路径算法等)以提高路由计算的能力. 关于 UMIR 协议的详细介绍见文献<sup>[3,4,6]</sup>.

### 2.2 路由激励

路由激励是 ISP 实施互联网高效路由的必要保证, 公平合理的激励方案是促使 ISP 进行路由合作以及保持路由稳定的重要基础<sup>[8]</sup>. 研究如何利用路由激励机制来激励 ISP 部署和实现高效的路由服务, 是互联网路由领域面临的重大挑战. UMIR 协议包括 2 种基本路由模式<sup>[6]</sup>: 策略路由和最小代价路由. 针对这 2 种路由模式, 分别设计了 2 种不同的激励模型: 针对 UMIR 的策略路由, 提出了一种基于合作博弈的路由激励模型<sup>[6]</sup>; 针对最小代价路由, 提出了一种激励兼容的最小代价路由机制<sup>[5]</sup>, 该机制能激励各协作节点真实宣告路由信息.

在文献<sup>[6]</sup>中, 利用合作博弈模型描述了 UMIR 网络中协作节点(AS)之间的合作与竞争关系, 将 UMIR 建模为 AS 间的合作博弈过程, 提出了基于合作博弈的路由激励模型(CGIM, cooperative-game-based incentive model). 全体 AS 是 UMIR

博弈中的博弈者集合，AS 通过宣告路由信息参与路由博弈，从路由博弈系统中获得一定的经济收益。其核心思路如下：通过考察路由博弈中 AS 形成路由联盟的理性基础以及联盟稳定的条件，提出了路由联盟和子联盟的概念，设计了路由联盟特征函数的计算方法；运用 Shapley 值方法及其核心概念解决了联盟收益的分配问题，设计开发了一种公平且可行的联盟收益分配算法。

针对 UMIR 协议的最小代价路由<sup>[5,6]</sup>，从机制设计的角度，提出了一种最小代价路由机制(LCPM, lowest cost path mechanism)，LCPM 是一种兼容激励的可信机制。LCPM 的主要特点是：1) 具有可信性，能激励参与人采取“诚实”策略参与机制运行；2) 有较低的过度支付。研究表明：在 LCPM 下，“诚实”是参与人的优势策略，它能最大化参与人的效用；而“撒谎”会造成系统性能的不稳定而且降低了参与人的效用值。另外，LCPM 具有较大的节俭率，该特性将使 LCPM 更具有实用性。

### 2.3 系统总体设计

为了方便系统设计以及后续功能扩展，UMIR 系统的设计与开发采用了模块化原则。路由系统主要由控制管理（CAM, controlling and management）模块、路由控制模块（RCP, routing control platform）和路由激励（RIM, routing incentive management）模块组成<sup>[6]</sup>，如图 1 所示。控制管理模块 CAM 通过消息队列获取各个模块的运行状态，负责监控调度其他模块，为路由系统提供用户界面，通过消息队列将控制台终端的各种命令消息发送给其他模块。路由协议模块（RCP）的主要功能是根据路由定制参数使用不同方法构造局部拓扑，以此为基础计算满足所给定制参数的路由，RCP 主要包括拓扑发现（TID, topology information discovery）子模块、路由计算（PCC, path computation center）子模块、路段构造（PLC, path-let constructing）子模块和路由管理（RTM, routing table management）子模块。路由激励模块（RIM）的主要功能通过调用不同的方法为所定制路径计算路由价格。该模块中已经实现的 2 种方法：最小代价路由(LCPM)激励法和基于合作博弈的路由联盟激励法(CGIM)。

## 3 控制管理系统

控制管理 CAM（controlling and management）模块<sup>[6]</sup>是整个 UMIR 系统的控制中心，它负责接收键盘

输入、各模块之间的通信以及监控调度其他模块。

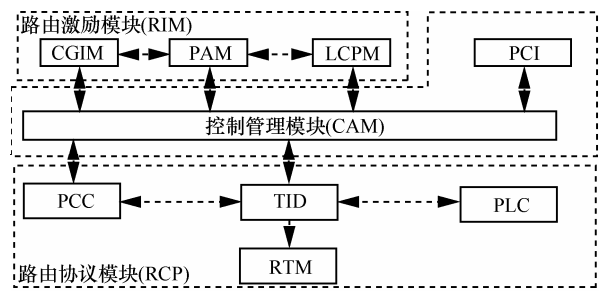


图 1 UMIR 系统框架

### 3.1 工作机制

UMIR 系统主要包括“控制管理”与“模块处理”2 个过程，将 2 个过程抽象成“CAM 监听”和“模块处理”，从而构成一个有限状态机，图 2 给出了 CAM 状态的转换。为了及时响应用户命令的输入，CAM 模块必须使用一个具有较高优先级的进程负责从键盘上接收用户命令，然后将这些用户命令再发给相应处理模块。

CAM 处理过程如图 2 所示。UMIR 启动后处于 CAM 监听状态，并以提示符“UMIR>”表示 UMIR 系统已经启动。UMIR 负责监听、解析和处理用户输入命令。当用户输入某一个“模块名”（module），状态机转入相应模块的运行状态；同时输出“module>”。例如，当用户输入路由控制模块(RCP)时，状态机转入 RCP 运行；同时输出“RCP>”。在“module>”提出符下，用户输入所有非“exit”命令都由该模块进行处理；模块收到命令后进行处理时，如果需要向屏幕输出内容时，直接采用 printf 输出结果，处理每个命令后该模块负责输出提示符“module>”。当用户在“module>”下输入 exit 命令时，状态机重新回到 CAM 模块，并输出“UMIR>”提示符，此时如果用户输入“exit”命令，则退出 UMIR 系统。

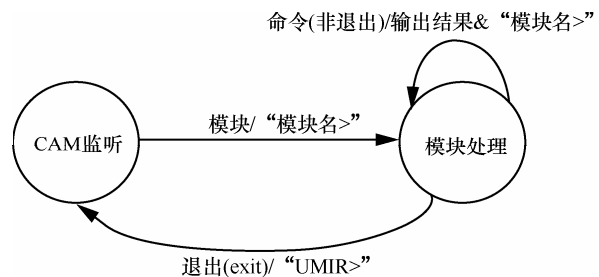


图 2 CAM 状态的转换

### 3.2 数据结构与处理函数

下面简要介绍 CAM 模块中的基本数据结构

和主要处理函数，详见系统研究报告<sup>[6]</sup>。CAM 模块通过消息队列与其他模块进行通信。CAM 接收用户的输入命令，并将该消息命令发给需要配置的模块。CAM 消息结构主要包括 5 个数据域：模块类型(`mdl_type`)、消息类型(`type`)、消息长度(`length`)、消息命令(`message`)。为了向其他模块发送消息命令，CAM 模块必须保存运行模块的名称及其消息队列等信息，该模块主要包括模块表项和模块表 2 个数据结构。CAM 模块主要包括 3 个处理函数：`umirCamMDLRegister` 函数，UMIR 系统中初始化每个模块时要调用该函数进行注册，以便在 CAM 模块表中就保留有每个模块的信息；`umirCamInputTask` 函数负责从接收用户输入并进行处理；`umirCamParse` 函数负责分析用户命令并发送给相应模块进行处理。

#### 4 路由控制子系统(RCP)

##### 4.1 模块设计

路由控制模块(RCP)的基本结构如图 1 所示，主要由拓扑发现 (TID, `topology information discovery`) 模块、路段构造 (PLC, `path-let constructing`) 模块、路径计算 (PCC, `path computation center`) 模块、路由管理 (RTM, `routing table management`) 模块等构成。

1) 拓扑发现模块。根据所定制的目的网络，使用节点选择算法选择相应的协作节点，并请求各节点构造路段信息，基于收集到的路段信息生成局部拓扑。

2) 路段构造模块。基于 BGP 路由表构造到目的网络的路段信息，利用路段封装算法得到路段报文。

3) 路径计算模块。基于局部拓扑以及用户路由定制需求，使用相应的路由算法计算符合要求的路径。

4) 路由管理模块。负责路由的实时监控、路由表的维护和检索以及隧道的建立与维护。

##### 4.2 报文格式

路由控制系统的上述 4 个功能模块中，拓扑发现模块和路径计算模块定制了重要的数据报文格式，拓扑发现模块定制了请求报文格式，路径计算模块 PLC 定制了路段报文格式。UMIR 系统启动后，首先初始化 CAM 模块并输入路由定制命令，然后 CAM 模块将定制命令消息发送给拓扑发现模块 TID 以及路由计算模块 PCC。由于 PCC 没有检索到局部拓扑信息而处于等待状态，系统调用 TID 模

块启动拓扑发现。因此该系统中重要的数据结构类型就是向协作节点所发送的请求报文。请求报文格式如图 3 所示。

uint Request_AS
uint Response_AS
char* User_Net
char* Destination_Prefix
uint Pathlet_Num
unit Incentive_Model

图 3 路由请求报文格式

**Request\_AS:** 表明发出报文的节点编号。

**Response\_AS:** 表示接受 Request\_AS 请求的节点编号。

**User\_Net:** 表示定制用户的网络标识，一般由网络前缀或自治号表示。

**Destination\_Prefix:** 表示用户定制的目的网络前缀或自治号。

**Pathlet\_Num:** 用于配置请求协作节点提供的路段数目。

**Incentive\_Model:** 所用的激励方法，可以是 STEM、LCPM 和 CGIM。

另外，协作节点接收到路由请求消息后，基于 BGP 路由表封装可用路径，得到路段项报文。路段项的格式定义如图 4 所示。

uint Pathlet_ID
char Destination_Prefix
char Pathlet_type
float[] Metrics
float Pathlet_cost
unit Incentive_Model

图 4 路段项格式

**Pathlet\_ID:** 标识路段，用路段相邻 2 个自治号表示。

**Destination\_Prefix:** 表示用户定制路由的目的网络。

**Metrics:** 表示路段的物理属性，例如带宽、延迟、丢失率等。

**Incentvie\_Model:** 所使用的激励方法，例如 STEM、LCPM 和 CGIM。

**Pathlet\_Type:** 表示路段的策略属性。

Pathlet\_Cost: 表示该路段的代价信息。

### 4.3 基本数据结构

为了保存用户定制的路由表，RCP 模块必须保存路由由计算模块为用户计算出的路由，故该模块包括路由表项和路由表 2 个数据结构。路由表项的数据结构如下所示。

```

Typedef struct umirRtEntry
{
    struct umirRtEntry *rt_next;
    char *destination_prefix;
    char *user;
    char *path;
    struct addras rt_nexthop_as;
} UMIR_RT_ENTRY;

```

模块表项用 umirRtEntry 表示，每条 umirRtEntry 表示一条目的网络的路由表项。所有的 umirRtEntry 用 rt\_next 由指针链接起来。destination\_prefix 表示目的网络前缀。user 表示用户网络。path 表示所定制的路径。rt\_nexthop\_as 表示下一跳 AS 地址。

路由表数据结构如下所示：

```

Typedef struct umirRtTable
{
    struct umirRtEntry *pHead;
    uint umirRtNum;
} UMIR_RT_TABLE;

```

其中 pHead 表示路由表项链表的头部，umirRtNum 记录路由表中路由条数。

### 4.4 主要处理函数

作为路由系统的核心路由算法包括拓扑发现算法和路由计算算法。前者包括不同激励方法下的局部拓扑发现算法。后者包括文献[6]中的路段构建算法和路由计算算法。RCP 模块包含主要处理函数如表 1 所示。

表 1 RCP 模块的主要处理函数

函数	描述
umirRcpNodeSelect	选择 AS 节点组成局部拓扑
umirRcpPathlet Encap	路段构造与封装
umirRcpPathComp	路径计算
umirRcpTopoDiscovery	基于路由表进行拓扑发现
umirRcpTunnelCreate	不兼容策略路径建立隧道

#### 1) umirRcpTopoDiscovery 函数

在 RCP 模块中，该函数的主要功能是构建局部拓扑。基本思想是：首先调用函数 umirRcpNodeSelect 获得局部拓扑节点集合；然后请求所得集合

中的协作节点构造路由信息并返回，即得到主控节点的局部拓扑。该函数流程如图 5 所示。

```

struct umirRcpTopoDiscovery
(char *destination_prefix /*目的网络*/)
1) 调用 umirRcpNodeSelect 函数选择协作节点，构造局部拓扑节点集合；
2) 对于所述 AS 集合中的各个节点发出构造路段项请求报文；
3) 调用 umirRcpPathlet Encap 函数/* 对到达目的网络的可行路径进行编码形成路段项，并将这些路段项返回给主控节点*/
4) 获得所有 AS 节点的路段项信息，形成局部拓扑

```

图 5 umirRcpTopoDiscovery 函数流程

#### 2) umirRcpNodeSelect 函数

在 CAM 模块中输入路由定制命令后，CAM 模块将该命令消息转发给 RCP 进行路由定制任务。此时，RCP 调用 umirRcpNodeSelect 函数，根据特定的路径选择策略和节点选择策略，选择 AS 节点构成局部拓扑。该函数流程如图 6 所示。

```

array umirRcpNodeSelect
(char * route_custom_para /*路由定制参数*/,
char *path_norm /*路径选择策略*/,
char *node_norm/*节点选择策略*/)
1) 根据路由定制参数，从路由表中获得到达目的网络的所有路径，确定 AS 节点候选集合；
2) 在节点候选集合的基础上选择 AS 节点组成局部拓扑的 AS 节点集合

```

图 6 umirRcpNodeSelect 函数流程

#### 3) umirRcpPathletEncap 函数

AS 节点收到路由请求消息后，基于其 BGP 路由表、路由定制的若干参数以及激励方法，可以生成不同的路段信息。该函数流程如图 7 所示。

#### 4) umirRcpPathComp 函数

该函数是系统获得局部拓扑后，被调用来计算所定制的路由。该函数根据定制参数例如目的网络、路径约束以及定制路径数等要求，计算出特定路径。该函数流程如图 8 所示。

#### 5) umirRcpTunnelCreate 函数

该函数的主要功能是：如果所计算的路径是一条不兼容中间协作节点的选路策略的路径，则必须与其协作节点进行协商并建立隧道。该函数流程如图 9 所示。

```

struct umirRcpPathletEncap
(char *destination_prefix; /*目的网络*/,
char *pathlet_vis_degree; /*路段可视度*/)
1) 提取目的前缀的所有路径;
2) if (umirModel==STEM)
{将主路径构造造成 type=1 的路段项, 重复将候选路径构造造成 type=2
的路段项, 直到路段项数目达到要求}
3) else if (umirModel==CRIM)
{包括主路径在内的有效路径构造造成 type=0 的路段项, 直到路段项
数目达到要求}
4) 返回路段项集合
    
```

图 7 umirRcpPathletEncap 函数流程

```

struct umirRcpPathComp
(char *destination_as /*目的 as 节点*/,
char *route_constraints /*路径约束*/,
uint route_num /*路径数目*/)
1) for (num=1; num<route_num; num++)
path[num]=compute_route(destination_as, route_constraints);
/* 计算到达目的节点的满足约束条件的路径*/
2) 返回路径
    
```

图 8 umirRcpPathComp 函数流程

```

struct umirRcpTunnelCreate
(char *path_vrpp /*不兼容协作节点的选路策略的路径*/)
1) 检查路径中的每条路段;
2) if (该路段是节点主路径)
    重复步骤 2, 直到该路径上所有不兼容策略的节点建立隧道; /*
使得整条路径由级联隧道构成可用路径*/
    else if (该路段不是节点主路径)
        向所述节点发出“建立隧道”请求; /*请求包括主控节点的隧道记
录*/
3) 返回所述节点将已建隧道; /*节点分配隧道 ID, 将自身控制的
路径信息加入隧道记录中, 并在其隧道表中安装, 然后将隧道记录
返回*/
4) 用返回隧道记录替换旧隧道记录
    
```

图 9 umirRcpTunnelCreate 函数流程

#### 4.5 节点交互机制

UMIR 系统的每一次路由定制任务都涉及主控节点与协作节点的频繁地信息交互过程。由于主控节点和协作节点的地位是相对的, 其身份取决于路由定制的发起方。在不同的路由计算中, 一个节点可以是一次路由计算的主控节点, 同时是另一次路

由计算的协作节点。路由计算(定制)过程主要涉及到主控节点与协作节点的信息交互。一般来说, 主控节点的工作流程如下。

1) 通过 CAM 模块输入路由定制命令, 触发 RCP 模块发起“路由计算”任务, 通过 TID 模块收集局部拓扑信息。

2) 在 TID 模块中, 调用 umirRcpNodeSelect 函数, 构造局部拓扑节点集。

3) 在 TID 模块中, 向各协作节点发出请求路段构造消息; 等待协作节点路由信息的返回。

4) 收集到协作节点的路段信息后, 即形成局部拓扑。

5) RCP 模块调用路径计算模块(PCC), 基于局部拓扑和定制参数计算符合要求的路径。

6) 对于不兼容策略的路径, 调用 RTM 模块建立隧道路径。

相应地, 协作节点的工作过程包括对主控节点请求的响应:

1) 若接收到“路段构造”请求消息, 调用 umirRcpPathletEncap 按要求构造路段信息, 将所构造的路由信息返回给主控节点;

2) 若接收到主控节点“建立隧道”请求时, 调用 umirRcpTunnelCreate 函数建立相应隧道。

## 5 路由激励系统(RIM)

### 5.1 基本数据结构

为了保存定制路径的路由价格, RIM 模块的主要数据结构是路由价格表项和价格表 2 个。路由价格表项的数据结构如下所示:

```

typedef struct umirPriceEntry
{
    struct umirPriceEntry *next_price;
    char *as_name;
    char *user;
    char *destination_prefix;
    float price;
} UMIR_PRICE_ENTRY;
    
```

价格表项用 umirPriceEntry 表示, 每个 umirPriceEntry 表示一个节点的路由价格。所有的 umirPriceEntry 用 next\_price 链接起来, 构成一条路径的路由价格。as\_name 表示自治系统编号。user 表示用户网络。destination\_prefix 表示定制的目的网络。price 表示 as\_name 所管理的路由价格。

价格表数据结构如下所示:

```

typedef struct umirPriceTable
{
    struct umirPriceEntry *pHead;
    uint umirPriceNum;
} UMIR_PRICE_TABLE;
    
```

其中 pHead 表示价格表项链表的头部，umirPrice Num 记录价格表中定制的目的网络数目。

### 5.2 路由价格计算

路由激励模块(RIM) 的主要功能是计算路由的价格。RIM 模块包括 3 个主要处理函数，即 umirRimPricing、umirRimLCPM 和 umirRimCGIM。

#### 1) umirRimPricing 函数

该函数是 RIM 模块的主要处理函数，用来计算所定制路径的路由价格。RIM 模块根据定制路径的不同类型，分别调用不同算法为路径计算路由价格。该函数流程如图 10 所示。

```

void umirRimPricing
(char *m_rim /*激励模型名称*/,
char *rt_path /*所定制的路径*/)
1. if (m_rim==LCPM) umirRimLCPM(rt_path);
2. else if (m_rim==CGIM)
umirRimCGIM(rt_path)
    
```

图 10 umirRimPricing 函数流程

#### 2) umirRimLCPM 函数

在 RIM 模块中，对于最小代价路由算法得到的路径，其路由价格的计算方法使用文献[5]中介绍的 LCPM 机制的支付算法。该函数流程如图 11 所示。

```

struct umirRimLCPM
(char *path_low /*最小代价路径*/)
1. 从最小代价路径 LCP 解析出 AS 节点;
2. 对于每个 AS 节点，执行如下两步:
2.1 计算子联盟 S(i,k), D{S(i,k)=0},d(i)=0和 d(i)=∞);
2.2 代入支付公式计算 p(i);
3. 返回路由价格
    
```

图 11 umirRimLCPM 函数流程

#### 3) umirRimCGIM 函数

在 RIM 模块中，对于策略路径，其路由价格的计算方法使用如文献[6]中给出的收益分配算法。该函数流程如图 12 所示。

## 6 系统部署与测试

系统部署与测试是检验路由系统能否正常运

行的必要手段，系统测试也需要一系列的测试标准，由于本文所设计与开发的 UMIR 系统尚处于初步试验阶段，缺乏通用的协议标准。本节利用美国 Utah 大学开发的 Emulab 测试平台<sup>[7]</sup>部署 UMIR 系统，通过设置具体的网络拓扑环境，运用测试场景来验证 UMIR 系统的有效性以及路由协议和路由激励方法的有效性和可行性。

```

struct umirRimCGIM
(char *path_policy /*UMIR 策略路径*/)
1) 从策略路径 path_policy 得到路由联盟 S;
2) 确定联盟 S 的子联盟及其总数;
3) 分别计算子联盟 M 的收益值: 对于任意子联盟 mj, 计算该子联盟的收益;
4) 计算路由联盟中各 AS 节点的收益值
    
```

图 12 umirRimCGIM 函数流程

### 6.1 网络环境

#### 1) Emulab 测试平台

Emulab 平台是由美国 Utah 大学开发的网络系统测试平台<sup>[7]</sup>，它是基于软硬设备的实验环境，根据配置文件为用户设置一个实际存在的物理拓扑，它模拟出来的系统与当前被模拟系统具有高度一致性。Emulab 根据用户提交的拓扑配置文件为用户建立一个真实存在的网络系统，并可以直接对网络系统中的每个节点安装 OS 系统以及其他系统软件等。用户将该节点视为一个真实网络节点并在其上部署需要验证的协议或系统，然后运用测试场景对被测系统进行实验验证，获得直接测试实验结果并对其进行分析，从而可以验证协议系统是否达到了预期设计的目的。

#### 2) 网络拓扑与配置

UMIR 系统所用的测试拓扑如图 13 所示，它由 8 个节点 12 条链路构成，每个节点都部署了 UMIR 系统。Emulab 使用 TCL 脚本编写拓扑配置文件。8 个网络节点全部采用了如下具体配置：机器型号 Pc3000，主频为 3 GHz，内存为 2 GB；OS 系统采用 RHL90-STD。12 条链路两端地址如表 2 所示。

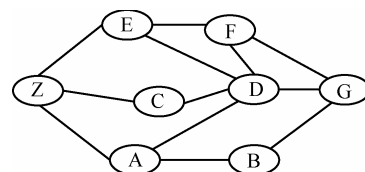


图 13 UMIR 路由网络

表 2 链路地址配置

编号	链路接口地址	编号	链路接口地址
Z-A	10.1.1.2-10.1.1.3	E-D	10.1.7.2-10.1.7.3
Z-C	10.1.2.2-10.1.2.3	A-B	10.1.8.2-10.1.8.3
Z-E	10.1.3.2-10.1.3.3	A-D	10.1.9.2-10.1.9.3
B-G	10.1.4.2-10.1.4.3	C-D	10.1.10.2-10.1.10.3
D-G	10.1.5.2-10.1.5.3	E-F	10.1.12.2-10.1.12.3
F-G	10.1.6.2-10.1.6.3	D-F	10.1.11.2-10.1.11.3

3) 测试场景的设置

为了验证系统及其各模块是否正常工作，设置了如下 3 个测试场景。

测试场景 1：定制一条从 Z 到 G 的最小路由代价和的路径。

测试场景 2：定制一条从 Z 到 G 的策略路径，其策略要求是所定制的路径必须经过 C 和 D 节点。

测试场景 3：定制 2 条从 Z 到 G 的策略路由，其策略分别是，一条路径要经过 E 和 D 节点，而另一条路径要避免节点 D。

根据编写的脚本文件，在 Emulab 测试平台上构建如图 14 所示的实际网络环境，通过登录网络节点并部署了 UMIR 系统。按照 3 种测试场景进行系统验证实验，然后对获得的实验观察结果进行分析。

6.2 测试结果与分析

根据上述的 3 个测试场景，当 UMIR 系统运行趋于稳定后，通过节点 Z 的 CAM 模块查看其实验结果，包括路由表和路由价格表的实验数据，其结

果如表 3 所示。

表 3 实验测试结果

场景	路径结果	路段价格
测试场景 I	Z-E-F-G	3.17, 3.3, 3.17
测试场景 II	Z-C-D-G	4.67, 1.67, 7.67
测试场景 III	Z-E-D-G Z-A-B-G	3.67, 4.5, 8.17 6, 3, 6

测试场景 I 的主要目的是为了验证系统的最小代价路由算法和 LCPM 的支付功能是否正常运行。从实验结果来看，定制路径为 Z-E-F-G，其路径代价之和为 2+1+2=5，分别少于路径 Z-E-D-G 的代价和 7 以及路径 Z-A-B-G 的代价和 10，故它是路径代价最小的路径，表明了最小代价路由算法的正确性。定制路径包括(Z-E)、(E-F)和(F-G) 3 段，这 3 段的价格分别为 3.17、3.3 和 3.17，实验结果与文献[6]中的计算结果刚好一致。该实验验证了 LCPM 的支付算法的正确性。

测试场景 II 的主要目的是为了验证系统的策略路由算法和 CGIM 的收益分配算法是否正常运行。从实验结果看，所定制路径为 Z-C-D-G，由于从网络拓扑中可以看出，满足定制策略要求，即“路径经过 C 和 D 节点”的路径只有一条 Z-C-D-G，故验证了策略路由算法的正确。所定制的策略路径包括(Z-C)、(C-D)和(D-G) 3 段，这 3 段的价格分别为 4.67、1.67、7.67，实验结果与文献[6]中的计算结果刚好一致。该实验验证了 CGIM 的收益分

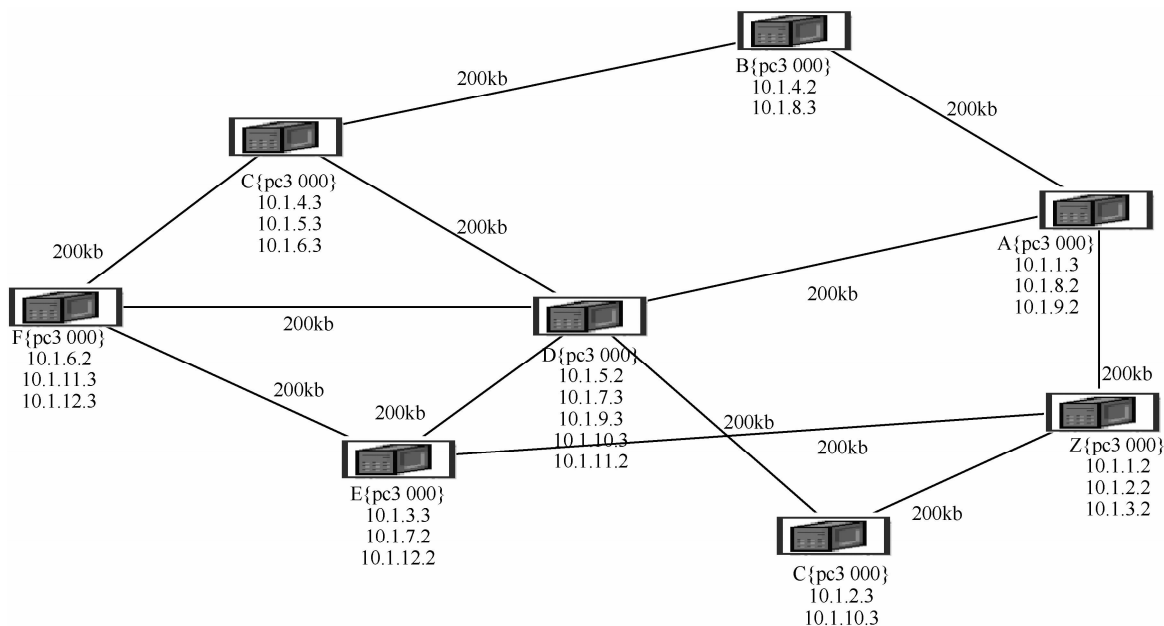


图 14 Emulab 的网络环境

配算法的正确性。

测试场景 III 的主要目的是用来验证系统是否具有多路径计算能力。从实验结果看, 2 条定制路径为 Z-E-D-G 和 Z-A-B-G, 这 2 条路径分别符合了定制要求, 即满足了“经过 E 和 D 节点”和“避免 D 节点”的策略要求。该实验验证了 UMIR 系统的路由算法能计算出多条符合策略的路径。

综上, 通过 3 个测试场景从不同角度进行系统测试, 不仅表明了系统的各个功能模块工作正常, 而且表明各模块之间的互操作正常。另外, 系统集成测试表明了各单元模块组合在一起能够按既定意图工作。初步的系统部署与测试表明 UMIR 系统达到了预期的设计目标。

## 7 结束语

在前期研究基础上, 设计和开发了 UMIR 系统, 它可以为用户提供多样化的路径计算和路由服务。UMIR 系统采用了模块化设计理念, 整个系统包括控制管理子系统、路由控制子系统和路由激励子系统。经过小规模的系统部署和初步测试, 验证了系统功能正常以及子系统与各模块的互操作性正常。在未来的研究中, 基于这个初步的路由系统, 计划进一步增加其他模块来完善系统的功能, 最终实现功能完备的、实用的域间多路径路由系统。

## 参考文献:

- [1] REKHTER Y, LI T, HARES S. A Border Gateway Protocol[R]. RFC 4271, Internet Engineering Task Force, 2006.
- [2] HE J, REXFORD J. Towards Internet-wide multipath routing[J]. IEEE Network Magazine, 2008, 22(2):16-21.
- [3] QIN D, YANG J, WANG H, *et al.* Multipath inter-domain routing via deviation from primary path[A]. Proceedings of the International Conference on Information Networking[C]. 2012. 222-227.
- [4] QIN D, YANG J, WANG H. Another multipath inter-domain routing [A]. Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications[C]. 2012. 581-588.
- [5] QIN D, YANG J, WANG H. Incentive compatible mechanism for

multipath inter-domain routing[J]. The International Journal of Advancements in Computing Technology, 2013, 5(7):339-345.

- [6] QIN D, YANG Y. Research Report of Multipath Inter-Domain Routing System[R]. Technical Report NOM-TR-001, 2012.
- [7] Emulab testbed[EB/OL]. <http://www.emulab.net/>.
- [8] HALABI S. Internet Routing Architectures[M]. Eastland, USA: Cisco Press, 2001.

## 作者简介:



秦董洪 (1974-), 男, 广西桂林人, 博士, 桂林电子科技大学副教授, 主要研究方向为计算机网络、路由协议、网络测量与网络安全、云计算等。



杨家海 (1966-), 男, 浙江云和人, 博士, 清华大学网络科学与网络空间研究院研究室主任, 主要研究方向为计算机网络、网络管理与测量、网络安全、云计算等。



杨洋 (1980-), 男, 江苏无锡人, 清华大学博士生, 主要研究方向为多路径路由及流量工程。



张辉 (1973-), 男, 甘肃兰州人, 硕士, 清华大学高级工程师, 主要研究方向为网络测量和网络管理。