

RIP 一致性测试系统的设计与实现

宇 佳, 赵保华, 周 颢

(中国科学技术大学计算机科学技术系, 合肥 230027)

摘 要: 针对协议一致性测试的要求和特点, 提出了一种 RIP 协议一致性测试系统的设计, 并给出了具体的实现。该系统采用虚拟测试法来完成对被测体的测试, 利用有限状态机以确保测试套的完备性, 并使用 TCL 语言作为测试例的脚本描述语言。文中给出了一个基于该系统的测试实例。试验结果表明, 这一系统具有较好的测试能力及可靠性高、可移植性好等特点, 为今后相关协议测试系统的实现提供了有益的借鉴。

关键词: RIP; 一致性测试; 测试系统; TCL

Design and Realization for Conformance Test System of RIP

YU Jia, ZHAO Baohua, ZHOU Hao

(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)

【Abstract】 According to the characteristic and requirement of protocol conformance testing, design and realization for conformance testing system of RIP protocol are presented. The system uses virtual testing method to test the implement under test, uses finite state machine to make sure the integrality of test sequences, uses TCL as the description language for the testing scripts. An example illustrating how to implement a test case using this system is given out. The testing result shows that the system has good testing ability, reliability and flexibility. It can be used for reference when developing relevant protocol testing systems in future.

【Key words】 RIP; Conformance test; Test system; TCL

协议是网络的内在灵魂, 协议实现的正确与否对保障网络性能有着重要的作用。作为检测协议实现正确性的一种基本手段, 协议一致性测试保证了各种网络路由产品相互连通和正确的相互操作。协议一致性测试是一种黑盒测试, 即在一定的网络环境下, 通过分析被测系统对外部输入信息做出的实际输出与期望输出的异同, 判断该系统是否正确实现了协议要求的功能。

RIP(Routing Information Protocol)是在协议规范 RFC 1058 中提出的一种 Internet 中常用内部网关动态路由协议。RIP 遵循距离矢量算法来构造路由表。它收集所有可到达目的地的不同路径, 并且只保存到达每个目的地的经过最少路由器的路由信息, 其它信息均予以丢弃。同时 RIP 路由器也定时把这些路由信息通知相邻的路由器, 使正确的路由信息逐渐扩散到了整个网络^[1]。基本的 RIP 协议(RIPv1)并非安全性的协议, 为了使它与更多的现代化的路由协议一致, 在改进的 RIPv2 中扩展了认证机制^[2]。本文针对 RIP 协议, 设计并实现了它的一致性测试系统, 成功地完成对其基本一致性测试。

1 RIP 协议一致性测试系统的设计

1.1 测试步骤

进行协议一致性测试的做法一般是首先要根据协议规范(Protocol Specification)取得严格的形式化描述; 接着从协议的形式化描述出发, 依照一致性测试的测试套生成方法, 产生相应的测试序列(test sequence), 再结合具体的测试数据(test data)和测试环境使用描述语言生成测试例; 最后将测试例在搭建的测试系统上执行, 根据测试记录参考 PICS 和 PIXIT 对被测系统进行评估, 生成测试报告。其中, PICS(Protocol Implementation Conformance Statement)是协议

一致性说明, 它和 PIXIT(Protocol Implementation Extra Information for Testing)协议实施附加说明一起决定了测试套的选择和测试准备中的测试参数设定^[3]。

1.2 测试方法

对路由协议的测试通常采用远程测试法^[4]。但这种测试方法通常需要多个测试体的参与, 所以会给实际组网带来困难。而且位于多个物理机器上的虚拟测试体之间的同步问题也会影响测试的可靠性。所以在对 RIP 协议的一致性测试中, 作者采用了虚拟测试法, 该方法的基本框架如图 1 所示。其中 IUT (Implementation Under Test)为待测协议实体, LT(Lower Tester)为下层测试体, 它经由 N-1 层服务提供者(Service Provider)发送 N-1 层抽象服务原语 ASP(Abstract Service Primitives)给 IUT, 以支持 IUT 的行动。在 ASP 中可能承载协议数据单元 PDU(Protocol Data Unit)。具体实现方法是在一台 LT 上设置多个物理网络适配器(Adapter), 同时用一个物理网络适配器来虚拟出多个网络适配器。可以在每个虚拟的网络适配器上配置多个 IP 地址, 从而将它们看成多个独立的测试体。这样理论上可以方便地利用较少设备组成任意网络拓扑结构。而且, 在这种虚拟测试环境中进行测试的效果和在具有相同拓扑结构的物理环境中进行测试得到的结果

基金项目: 国家自然科学基金资助重大研究计划项目(90104010); 国家自然科学基金资助项目(60241004); 国家“973”计划基金资助项目(2003CB314801)

作者简介: 宇 佳(1980-), 女, 硕士生, 主研方向: 通信软件, 协议工程; 赵保华, 教授、博导; 周 颢, 硕士、讲师

收稿日期: 2006-03-20 **E-mail:** pure2mail.ustc.edu.cn

相同。

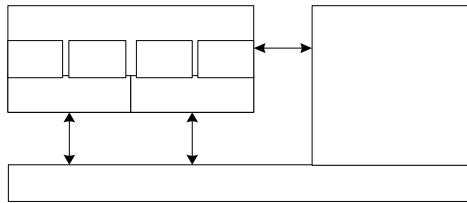


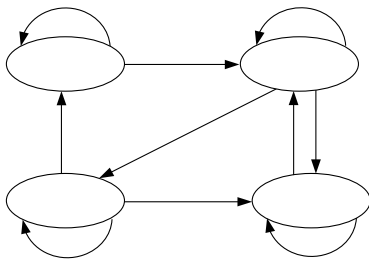
图 1 虚拟测试法的基本框架

1.3 测试套的生成

根据 RIP 协议的特点和一致性测试的目的, 可将 RIP 协议一致性测试划分为 3 种类型:

- (1) 报文格式测试: 测试在特定的情况下路由器产生的报文格式是否合法, 例如, 回复报文(response)中的 cmd 域值是否是 2, 以及收到格式不正确的报文时, 路由器的处理是否与 RIP 协议规范一致。
- (2) 路由信息交换测试: 测试路由信息的产生、接收和发布是否与 RIP 协议规范一致。
- (3) 路由表计算测试: 测试 RIP 路由表的计算是否与协议规范一致。

上述 3 种不同测试内容的抽象测试套的生成也有一定差异: 报文格式测试, 如何根据协议描述选用合适的数据结构表示报文结构是测试关键。对于路由信息交换测试和路由表计算测试, 由于路由信息的产生和计算与协议描述的不同状态间的转化紧密相关, 因此在形式化描述的基础上生成完备的测试套是测试关键。作者在 RIP 协议一致性测试系统中使用了有限状态机(FSM)来辅助产生完备的测试套, 因为 FSM 的描述可以产生一个完整的、没有歧义的协议模型, 由它产生的测试套可以达到很高的错误覆盖率。另外, 从形式化规范中更容易抽取模型化的信息, 这使得对测试中产生的问题寻求公式化和理论答案提供了很大的便利^[2,5]。构造出的 RIP 的有限状态机如图 2 所示。



1: 定时更新; 2: Timeout 计时器超时; 3: 垃圾清理计时器时间到; 4: 随机偏移计时器时间到; 5: 收到新路由; 6: 路由建立完成; 7: 收到更新路由; 8: 收到请求; 9: 删除路由完成; 10: 路由更改标记置位; 11: 收到无效路由或非更佳路由; O1: 发出响应; O2: 触发更新; O3: 正常更新; #: NONE; ~: OR; &: AND; a/b: b occurs after a

图 2 RIP 有限状态机

1.4 测试套的编写

文章介绍的一致性测试系统采用一种简洁高效的脚本解释语言 TCL 来描述测试套, 与传统的 TTCN 规范描述相比, 它有着执行效率高、描述能力强和可移植性好等特点。TCL 可以按照基于组件的方法(Component Approach)将程序分割

成一个个小而完备、可以重复使用的组件。不同的组件有不同的功能, 用于不同的目的, 并且能为其它应用程序所利用。可以针对某一特定应用领域对 TCL 语言的核心命令集进行扩展, 加入适合于自己应用领域的扩展命令, 扩展后的 TCL 语言仍然保留有 TCL 核心部分的所有功能。可以将扩展命令的实现封装在一个动态链接库里, 在一个输出函数里使用 TCL 提供的函数 TCL_CreateCommand 向 TCL 注册扩展命令。使用时, 只要在 TCL 平台上调用 TCL 内建命令 loadcmd 载入该库即可使用这些扩展命令。

2 RIP 协议一致性测试系统的具体实现

2.1 底层辅助命令和扩展命令的开发

根据测试需求, 先用 C/C++ 编写扩展命令函数, 再将这些扩展命令集成到 TCL 平台上。因为所需的扩展命令中包括 RIP 报文的组包、解包, 所以首先需要设计并实现一些底层辅助的报文缓冲区编解码支持命令, 如 pack_set, pack_new 等, 如图 3 所示。

例 1 一些辅助的底层支持命令

```
pack_new size; //申请一块新的缓冲区
pack_set pId -p addr spld ?size?; // 按指定格式设置缓冲区指定
//域的内容
pack_comb pId1 pId2 ?...?; //将多个缓冲区合并生成一个新的缓
//冲区
pack_get pId -b addr bit; //按指定格式获取缓冲区指定域的内容
pack_free pId; //释放缓冲区
pack_len pId; //获取缓冲区的大小
```

这些支持命令可以实现基本的管理和操作报文缓冲区的操作, 包括缓冲区的申请和释放, 以及按各种格式填写缓冲区和获取缓冲区的内容等。

有了这些操作命令, 就能够根据测试的具体需要编写扩展命令。对于 RIP 协议来说, 主要是添加一些 RIPv1、RIPv2 的报文编解码函数, 以及报文收发接口支持函数。表 1 给出了一些 RIPv2 部分的扩展命令。

表 1 部分扩展命令介绍

| 命令名 | 功能 | 返回值 |
|------------------|--------------|--------------------------------|
| rip2_sdu | 构造一个报文 | 对应报文缓冲区 |
| rip2_routeentry | 构造一个路由条目 | 对应路由缓冲区 |
| md5_encode | 进行 md5 加密 | 获取的 16 进制认证数据 |
| rip2_getentry | 取得指定路由条目 | 对应条目缓冲区 |
| rip2_getcmd | 取得报文的头部命令区域值 | 1: request 报文 2: respond 报文 |
| rip2_getmetric | 获取该路由条目的度量 | 相应的度量值 |
| rip2_getauthtype | 获取报文的认证类型 | 2: 采用明文验证 3: 采用 MD5 认证 |
| rip2_getauthdata | 获取报文加密后的密文 | 报文中的认证数据 |
| RIP::Init | 创建发送接收接口 | 套接口描述符 |
| RIP::Send | 发送一个 RIP 报文 | 无 |
| RIP::Receive | 接收一个 RIP 报文 | 报文载荷的缓冲区 |

2.2 测试实例

本节给出一个测试实例, 该测试例测试在启用 MD5 认证的 RIPv2 协议环境中, 路由器产生并传播的路由信息是否正确。相关测试例如下:

(1) 测试目的

将发送、接收接口配置成支持 MD5 认证的时候, 路由器接收发送来的带有 MD5 认证的报文, 将其中新的路由条目添加到路由表中, 并组播更新后的路由表。

(2) 测试过程

1) 测试设备连接如图 3, 测试主机 PC1 远程登录到路由

器 RUT 上, 将 RIP 版本配置成带 MD5 认证的 RIPv2。

2)配置 RUT 的 2 个接口 send_sock 和 recv_sock,使它们运行 MD5 认证,采用的密码为 key1。

3)测试主机 PC1 向路由器发送一个带有 MD5 认证,并含有一条新的路由条目的 respond 报文。

4)虚拟主机 PC2 在 recv_sock 接口接收路由器发出的更新报文,并对其进行解析。

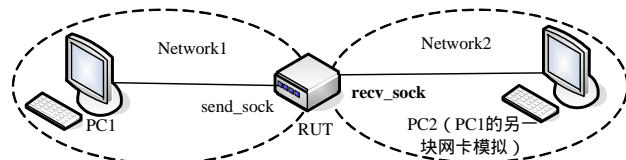


图 3 逻辑组网

(3)测试脚本

由于完整的测试例比较复杂,而且会涉及到很多未介绍的测试命令和繁杂的路由器配置,因此对测试例的开头,结尾和中间不重要的过程做了适当的省略。

```
#step1:加载本测试脚本需要 base 库并绑定主机
source loadcmd.tcl
arpd $s_adapter -start
arpd $s_adapter -add $s_addr1 $s_mac1
arpd $s_adapter -add $s_addr2 $s_mac2
#step2:配置路由器,使其支持 MD5 认证,输入 key1 为密钥
#step3:创建发送和接收 RIP 报文的接口
set send_sock [RIP::Init $s_adapter $s_mac1 $multimac $s_addr1
$s_port1 $multiaddr $d_port1]
set recv_sock [RIP::Init $s_adapter $multimac $d_mac2
$multiaddr $s_port2 $d_addr2 $d_port2]
#构造路由表条目
set entrylist ""
set re [rip2_routeentry $ip4addr $ip4mask $s_addr1 $metric]
set entrylist [rip2_addentry $entrylist $re]
set send_pck [rip2_sdu $cmd $auth $keyid $keychain $seqnum
$entrylist]
#step4:PC1 发送报文,PC2 接收报文并将其解析
RIP::Send $send_sock $send_pck
set recv_pck [RIP::Receive $recv_sock 20000 "" "" ""]
puts $recv_pck
if { $recv_pck == "" } {
#超时,未收到报文时的处理
} else {
set rip2_cmd [rip2_getcmd $recv_pck]
#检查收到的报文是否是来自 IUT 的更新报文且其是否为 MD5
认证类型
if { $rip2_cmd==2 && [rip2_getauthtype $recv_pck]==3 } {
set num [rip2_getentrycnt $recv_pck]
set i 0
while { $i < $num } {
set reentry [rip2_getentry $recv_pck $i]
#检查当前从所收到报文的路由表里取出的路由条目是否为刚
才向 IUT 通告的条目。
if { [string compare $ip4addr [rip2_getdstaddr
$reentry]] == 0
&& [string compare $ip4mask
[rip2_getnetmask $reentry]] == 0 && [string compare $defaultnexthop
[rip2_getnexthop $reentry]] == 0
&& [rip2_getmetric $reentry] == $metric+1
&& [rip2_getauthdata $recv_pck] ==
```

```
[rip2_getcalcauthdata $recv_pck $keychain]] {
set success 1
break
}
incr i 1
}
}
```

#路由添加失败时的处理,略

#step5:关闭接口,将本测试例的路由配置清空并断开与路由器的连接,略。

这个测试例的预期效果是 PC2 收到 RUT 通过 Network2 发来的路由通告,其中包含了先前 PC1 发给 RUT 的那条路由信息,且该条路由的度量(metric)加 1。

2.3 测试结果

和上面的例子相似,可以用 TCL 脚本实现并在实际的测试环境中测试其他测试例,从而得到测试结果。利用开发的扩展命令,作者在 Cisco 路由器上对 RIPv1, RIPv2 协议的主要功能点进行了初步的测试,测试结果如表 2。

表 2 在一种路由器上的测试结果

| 测试类型 | 测试功能点 | 总数 | 成功 | 失败 |
|--------|--------|-----|-----|----|
| 报文格式 | 报文合法性 | 12 | 12 | 0 |
| 路由信息交换 | 路由请求响应 | 5 | 5 | 0 |
| | 路由重分配 | 20 | 20 | 0 |
| | 路由聚合 | 4 | 4 | 0 |
| | MD5 认证 | 6 | 4 | 2 |
| | 水平分割 | 8 | 8 | 0 |
| | 路由收敛 | 6 | 6 | 0 |
| 路由表计算 | 定时器 | 14 | 14 | 0 |
| | 路由更新 | 37 | 37 | 0 |
| 合计 | | 112 | 110 | 2 |

其中,没有通过的 2 个测试例是因为协议的实现和协议规范不一致造成的。协议规范规定,在一个接口上只能有一个密钥生效,但是实际上在接口定义的 255 个密钥可同时生效。

3 结论

在分析协议一致性测试原理和 RIP 路由协议特点的基础上,本文提出了一种 RIP 一致性测试系统的设计思路,并在这种思路的指导下具体实现了这一测试系统。在 Cisco 路由器上的试验结果表明,该测试系统能够较好地完成对 RIP 路由协议的一致性测试,并具有可靠性高、可移植性好等特点。今后的工作方向主要集中在使用 TCL 语言开发一个更通用的测试平台,使其能对更多的相关协议,如 RIPng(RIP for next generation)协议进行测试,并且可以实现测试例的自动产生、选择,以及协助调试测试脚本等功能。

参考文献

- Hedrick C. Routing Information Protocol[S]. RFC1058, 1988-06.
- Malkin G. RIP Version 2[S]. RFC2453, 1998-11.
- Dahbura A T, Sabnani K, Uyar M Ü. Formal Methods for Generating Protocol Conformance Test Sequences[J]. Proceedings of the IEEE, 1990, 78(8): 1317-1326.
- ISO/IEC 9646-2 Information Processing Systems, Open System Interconnection, OSI Conformance Testing Methodology and Framework, Part 2: Abstract Test Suite Specification[S]. 1994.
- Baker F, Atkinson R. RIP-2 MD5 Authentication[S]. RFC2082, 1997-01.