

边界网关协议 BGP4+ 的一致性测试研究^{*})

丁雪莲 李华 叶新铭

(内蒙古大学计算机学院 呼和浩特 010021)

摘要 随着互联网的快速发展,高效稳定的路由协议成为保障网络性能的重要因素。协议测试是保证协议实现的性能的有效手段。本文针对边界网关协议 BGP4+ 进行了一致性测试研究;对 BGP4 及其面向 IPv6 的扩展 BGP4+ 进行分析,给出了 BGP4+ 协议测试的有限状态机模型。运用形式化与非形式化相结合的方法生成了 BGP4+ 的测试例。最后给出了运用测试例对具体的协议实现进行测试的结果。

关键词 BGP4+,有限状态机,测试例,IUT

Conformance Test Study of Border Gateway Protocol-BGP4+

DING Xue-Lian LI Hua YE Xin-Ming

(College of Computer Science, Neimongol University, Hohhot 010021)

Abstract With the development of Internet, effective and steady routing protocol will be an important factor to ensure performance of network. Protocol Testing is an effective method to ensure performance of protocol implementation. We first analysis the feature of BGP4 and the extension of BGP4 oriented to IPv6, BGP4+. Moreover, FSM model of BGP4+ protocol is presented. Test cases of BGP4+ are generated by combining formal and informal method. At last, the test results for a real routing software, Zebra, are given.

Keywords BGP4+,FSM,Test case,IUT

1 引言

随着互联网的快速发展,高效稳定的路由协议成为保障网络性能的重要因素。协议测试是保证协议实现的性能有效手段。目前互联网域间路由协议采用的是 BGP4^[1]。在未来的 IPv6 网络中,域间路由协议将采用边界网关协议 BGP4 及其面向 IPv6 的扩展 BGP4+^[2]。但由于各个生产厂商对协议的理解不同,产生的协议实现也各不相同,所以对 BGP4 协议的测试工作也变得尤为重要。

协议测试就是对具体的协议实现进行验证,它包括许多方面,如功能测试、性能测试、一致性测试等。其中,协议一致性测试是检验协议的实现是否与协议标准相一致,进而确保不同厂商的协议实现之间能够互操作。而协议一致性测试中工作量最大、最耗时的就是测试例的生成,因而如何产生既具有较高检错能力又具有一定的覆盖范围的测试例的集合受到许多科研人员的关注。

针对边界网关协议的测试研究,国内外有多家科研院所从事这方面的工作,也发表了一些有关这方面的文章。在文[3]中,作者针对路由协议的特点,给出了路由协议的测试模式。在对 BGP4 协议进行测试的过程中,他们采用了形式化与非形式化相结合的思想开展测试。在文[4,5]中,针对 BGP4 作为路由协议的特点,对现有的测试方法进行了分析研究,并在自己开发的协议集成测试系统上对 BGP4 协议实现进行了测试。在文[6]中提出了一种测试目的自动生成的方法,将这一方法应用到路由协议的测试当中具有很好的指导意义。目前在 BGP4+ 协议测试研究方面的文章并不多

见。

2 BGP4 及其面向 IPv6 的扩展 BGP4+ 协议分析

Internet 是由大量自治系统(Autonomous System,简称 AS)构成的复杂网络。在 Internet 上,边界网关协议 BGP4 是目前广泛使用的 AS 间的路由协议。

BGP4 协议的主要功能是在运行 BGP4 的路由器之间交换路由信息,使得各个自治系统间能够正确地进行通信。这些运行 BGP4 的路由器称为 BGP 发言者。BGP 发言者与 BGP 发言者互称为对等体。RFC1771 是边界网关协议的标准文本说明,它规定了 BGP4 应该实现的功能。但随着 IPv6 的部署,为了使 BGP4 协议支持更多的网络层协议,IETF 修订完成了 RFC2858(Multiprotocol Extensions for BGP4,即 BGP4+)。在 BGP4 协议中有三部分与 IPv4 的信息相关,即 NEXT-HOP、NLRI 和 AGGREGATOR 属性。假设任何 BGP 发言者都有一个 IPv4 地址(包含在 AGGREGATOR 属性中),那么要使 BGP4 支持多网络层协议的路由只需要加入两个功能:把特定的网络层协议分别与 NEXT-HOP 和 NLRI 关联起来。为此在 BGP4 中增加了两个属性:MP-REACH-NLRI(Multi-protocol Reachable NLRI)和 MP-UNREACH-NLRI(Multiprotocol Unreachable NLRI)。这两个属性是可选非传递的,不支持多协议扩展的 BGP 发言者会忽略这两个属性中携带的信息并且不转发这些信息。

MP-REACH-NLRI 是结构比较复杂的属性,详细定义了所使用的网络层协议的类型、下一跳的长度和地址、子网接入点信息和 NLRI 信。如果在 UPDATE 消息中使用了 MP-

^{*}国家自然科学基金项目(60263002)、内蒙古科技攻关项目(2002061002)、内蒙古自然科学基金(200308020213)。丁雪莲 硕士研究生,主要研究方向为协议一致性测试;李华 副教授、硕士生导师,研究方向为协议测试及计算机网络应用;叶新铭 教授、博士生导师,主要从事计算机网络和分布式系统方面的研究。

REACH-NLRI属性就不需要原来的 IPv4 的 NEXT-HOP 属性了。而 MP_UNREACH_NLRI 属性相对简单得多,如果在 UPDATE 消息中使用了 MP_UNREACH_NLRI 属性,就不需要携带其它的路径属性了。这与 RFC1771 中规定的如果 UPDATE 消息中不包含强制属性就发送 NOTIFICATION 消息有了很大的不同。BGP4+ 中还增加了对这两个新属性的错误处理。RFC2858 中还采用了 BGP 能力通告协议^[7],可用于在 BGP 连接建立时决定是否与对等体使用多协议扩展传递路由信息。

3 测试方法的选择

ISO/IEC-9646 根据不同的控制观察点定义了不同的抽象测试方法,现有的端系统抽象测试方法基本上可以分为四类,即本地测试法、协调式测试法、分布式测试法和远程测试法^[8]。

由于远程测试法的主要特点是不要求能访问 IUT(被测实现)的上边界,也不要求显示的测试协调过程,只是把 IUT 当作黑盒。并且这种方法依赖于被测协议来实现 IUT 和 LT(下测试器)间的同步。测试判决是由基于下测试器对 IUT 提供的激励以及下测试器所观察到的 IUT 的响应做出的。所以我们在对 BGP4+ 进行测试的过程中选择了远程测试方法,测试环境如图 1 所示。

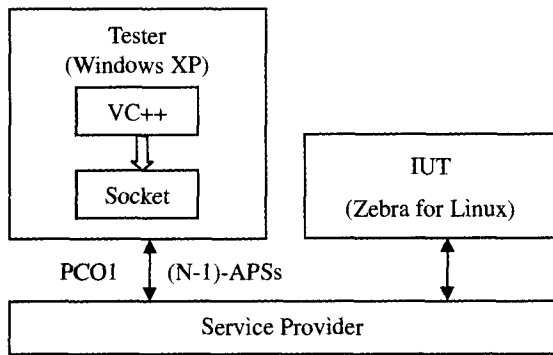


图 1 远程测试方法

4 测试 BGP4+

目前在一致性测试集生成技术的研究中,有限状态机(FSM)模型是对协议进行建模的一个强有力的工具。作为协议标准和生成的测试集之间的中间描述手段,它在描述协议的状态转换方面具有很强的能力。而唯一的输入/输出(UIO)序列是对有限状态机进行功能测试的有效手段。

4.1 基本概念

4.1.1 有限状态机定义

有限状态机 FSM 是个五元组 $(I, O, S, \delta, \lambda)$ 。其中, I 是输入标记集, O 是输出标记集, S 是状态集, $\delta: S \times I \rightarrow S$ 是状态变迁函数, $\lambda: S \times I \rightarrow O$ 是输出函数。 I, O 和 S 都是非空有限集。

4.1.2 UIO 序列的定义

状态 s 的 UIO 序列 x 是能够唯一地标志状态的输入向量序列,在该输入向量序列作用下,状态 s 的输出向量序列有别于任意其它状态 $t (\forall t \in S \wedge t \neq s)$ 的输出向量序列,即 $\lambda(t, x) \neq \lambda(s, x)$ 。

4.2 BGP4+ 协议测试的有限状态机模型

我们可以从协议标准得到用有限状态机表示的协议的形

式化描述。由于一个完整的 BGP4+ 互连行为的有限状态机较为复杂,如图 2 所示。图中的 S_1 到 S_6 分别对应协议的 Idle 状态、Connect 状态、Active 状态、OpenSent 状态、OpenConfirm 状态和 Establish 状态。为清晰起见,图 2 省略了各条变迁上的输入输出信息。图 3 中给出了另一个有限状态机模型,它是描述被测 BGP 实体主动建立连接模式下的有限状态机,是在图 2 的基础上去掉了 Connect 状态和 Active 状态得到的。在图 2 中,从 Idle 状态(S_1)到 OpenSent 状态(S_4)要经过 Connection 状态(S_2)和 Activate 状态(S_3),而触发状态转变的事件是发生在 TCP 层的,所以 Connection 状态和 Activate 状态属于不可测的状态,从 Idle 状态到 OpenSent 状态经过的变迁也就是不可测的变迁。省略了这些状态和变迁的模型如图 3 所示。

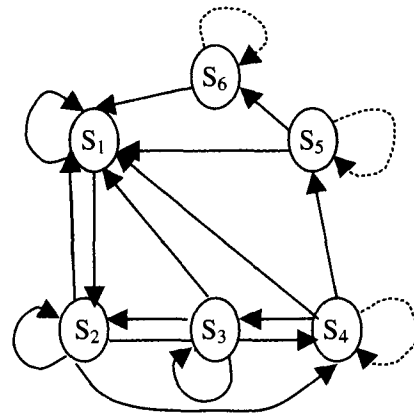


图 2 原有限状态机模型

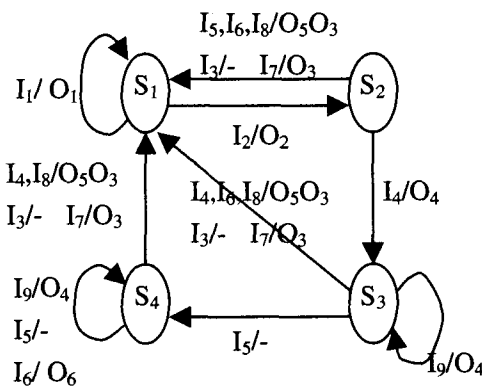


图 3 简化的有限状态机模型

下面列出了简化后的状态机模型的各个状态、变迁以及输入输出事件。

- S_1 : Idle S_2 : OpenSent
- S_3 : OpenConfirm S_4 : Established
- I_1 : ConnectRetry Timer 超时
- I_2 : Start 事件
- I_3 : 收到 Disconnect notification 消息
- I_4 : 收到 Open 消息
- I_5 : 收到 KEEPALIVE 消息
- I_6 : 收到 UPDATE 消息
- I_7 : 收到 NOTIFICATION 消息
- I_8 : Hold Timer 超时
- I_9 : KeepAlive Timer 超时

- O₁: 重启 ConnectRetry Timer
- O₂: 发出 OPEN 消息
- O₃: 释放资源
- O₄: 发出 KEEPALIVE 消息
- O₅: 发出 NOTIFICATION 消息
- O₆: 重新启动 Hold Timer

表 1 FSM 的 UIO 序列

State	UIO	Tail-State
S ₁	I ₁ /O ₁	S ₁
S ₂	I ₄ /O ₄	S ₃
S ₃	I ₉ /O ₄ I ₆ /O ₅ O ₃	S ₁
S ₄	I ₆ /O ₆	S ₄

4.3 测试例的生成与应用

从该状态机出发,使用 U 方法^[9]生成 BGP4+ 互连行为有限状态机中各条变迁的测试序列。在这个方法中,对有限状态机的每个状态都产生一个唯一的输入/输出(UIO)序列,如表 1 所示。从该状态机中共生成 22 条测试序列。例如,当 IUT 处于 OpenSent 状态(S₂)时,接收到 OPEN 消息(I₄),并发出 KEEPALIVE 消息(O₄)后,用来验证 IUT 是否从 OpenSent 状态(S₂)转移到 OpenConfirm 状态(S₃)的测试序列表示为:(r/- I₂/O₂) I₄/O₄(I₉/O₄I₆/O₅O₃)。这里 r 是复位操作,I₂/O₂ 是从初始状态 S₁ 到状态 S₂ 的前导序列,I₉/O₄I₆/O₅O₃ 是状态 S₃ 的 UIO 序列。对生成的 22 条测试序列进行分析和选择,删除 2 条被其它测试序列完全包含的测

试序列,从该状态机共得到 20 个测试例,如表 2 所示。

表 2 FSM 的测试例

NO.	Test Case	NO.	Test Case !
1	(r/-)I ₁ /O ₁ (I ₁ /O ₁)	11	(r/-I ₂ /O ₂ I ₄ /O ₄)I ₃ /-(I ₁ /O ₁)
2	(r/-I ₂ /O ₂)I ₅ /O ₅ O ₃ (I ₁ /O ₁)	12	(r/-I ₂ /O ₂ I ₄ /O ₄)I ₇ /O ₃ (I ₁ /O ₁)
3	(r/-I ₂ /O ₂)I ₆ /O ₅ O ₃ (I ₁ /O ₁)	13	(r/-I ₂ /O ₂ I ₄ /O ₄)I ₉ /O ₄ (I ₉ /O ₄ I ₆ /O ₅ O ₃)
4	(r/-I ₂ /O ₂)I ₈ /O ₅ O ₃ (I ₁ /O ₁)	14	(r/-I ₂ /O ₂ I ₁ /O ₁ I ₅ /-) I ₄ /O ₅ O ₃ (I ₁ /O ₁)
5	(r/-I ₂ /O ₂)I ₃ /-(I ₁ /O ₁)	15	(r/-I ₂ /O ₂ I ₄ /O ₄ I ₅ /-) I ₈ /O ₅ O ₃ (I ₁ /O ₁)
6	(r/-I ₂ /O ₂)I ₇ /O ₃ (I ₁ /O ₁)	16	(r/-I ₂ /O ₂ I ₄ /O ₄ I ₅ /-) I ₃ /-(I ₁ /O ₁)
7	(r/-I ₂ /O ₂)I ₄ /O ₄ (I ₉ /O ₄ I ₆ /O ₅ O ₃)	17	(r/-I ₂ /O ₂ I ₄ /O ₄ I ₅ /-) I ₇ /O ₃ (I ₁ /O ₁)
8	(r/-I ₂ /O ₂ I ₄ /O ₄)I ₄ /O ₅ O ₃ (I ₁ /O ₁)	18	(r/-I ₂ /O ₂ I ₄ /O ₄ I ₅ /-) I ₉ /O ₄ (I ₆ /O ₅)
9	(r/-I ₂ /O ₂ I ₄ /O ₄)I ₆ /O ₅ O ₃ (I ₁ /O ₁)	19	(r/-I ₂ /O ₂ I ₄ /O ₄ I ₅ /-) I ₅ /-(I ₆ /O ₅)
10	(r/-I ₂ /O ₂ I ₄ /O ₄)I ₈ /O ₅ O ₃ (I ₁ /O ₁)	20	(r/-I ₂ /O ₂ I ₄ /O ₄ I ₅ /-) I ₆ /O ₅ (I ₆ /O ₅)

对于无法抽取规范形式化描述的测试内容需要给出测试功能和测试目的,根据 RFC 协议标准手工抽取测试例。首先在详细阅读并理解 RFC 协议标准的基础上,对协议进行测试项目的划分。然后根据不同的测试项目确定该项目中的测试内容和测试目的。最后针对不同的测试目的设计合适的测试数据和测试配置。表 3 是我们采用手工抽取测试例的例子,表中列出了测试目的、测试配置、测试步骤以及预期的测试结果。

表 3 测试例的例子

测试项目 UPDATE 消息格式一致性测试
测试目的 保证被测设备收到只包含 MP_UNREACH_NLRI 属性,不包含其他 PATH-ATTRIBUTE 的 UPDATE 消息时,仍认为该消息有效。
测试配置
测试步骤 <ol style="list-style-type: none"> 1. 用测试端口 1 与被测设备建立外部 BGP4+ 连接。 用测试端口 2 与被测设备建立外部 BGP4+ 连接。 2. 由测试端口 1 向被测设备发送 UPDATE 消息,其中 MP_REACH_NLRI 属性的 NLRI 字段为 AS9-NLRI1,在测试端口 2 上监测被测设备发来的消息。 3. 由测试端口 1 向被测设备发送 UPDATE 消息,其中 MP_UNREACH_NLRI 属性的 NLRI 字段为 AS9-NLRI1,不包含其他 PATH-ATTRIBUTE 属性,在测试端口 2 上监测被测设备发来的消息。
预期测试结果 <ol style="list-style-type: none"> 1. 步骤 2 中,测试端口 2 收到被测设备发来的 UPDATE 消息,其中 MP_REACH_NLRI 属性的 NLRI 字段为 AS9-NLRI1。 2. 步骤 3 中,测试端口 2 收到被测设备发来的 UPDATE 消息,其中 MP_UNREACH_NLRI 属性的 NLRI 字段为 AS9-NLRI1。

参考 ISO/IEC9646 的建议,根据协议的测试目的和 BGP4+ 协议标准,我们将测试分为互连行为状态机测试、消息格式一致性测试、消息处理功能测试、路由处理功能测试、路由流量控制功能测试、BGP4+ 差错处理功能测试、路由反

射功能测试和 Community 属性处理功能测试。除去互连行为状态机测试部分自动生成的 20 个测试例,其它的测试部分我们手动提取了 92 个测试例。使用这些测试例我们对 Linux RedHat 9.0 Zebra 0.93b 中的 bgpd 进行了一致性测

试,在测试的过程中发现的与协议标准不一致的地方包括:消息格式一致性测试中对 ORIGIN 字段、AS-PATH 字段和 LOCAL-PREF 字段的处理;BGP4+消息处理功能测试中对 MP REACH_NLRI 属性中的 NEXT-HOP 字段的处理;路由处理功能测试中对下一跳不可达路由的处理、根据 BGP4+标识符或测试端口地址打破平局进行路由选择的处理以及对路由聚合的处理;路由反射功能测试中对不正确路由的忽略处理;路由流量控制功能测试中对根据 BGP4+标识符或测试端口地址打破平局进行路由流量控制的处理;BGP4+差错处理功能测试中对 UPDATE 消息头中的 Length 字段不正确情况的处理和对 AS 路由循环的处理。

结论 随着下一代互联网技术的部署,下一代路由协议将占据越来越重要的位置。其中广域网的域间路由协议大多数使用 BGP 协议转发路由信息,且 BGP 协议主要位于域间的路由器上。而 BGP4+路由协议作为下一代互联网中最基本和最重要的域间路由协议,对其协议实现的一致性进行测试具有重要的现实意义。本文首先对边界网关协议进行了分析,然后针对 BGP4+互连行为部分进行形式化建模,在形式化模型的基

础上自动生成测试例,并给出了测试例列表。对于无法形式化的内容,则采用手工设计测试数据的方式与参考实现相互配合开展测试。在运用这些测试例对 BGP4+协议实现,如 Zebra 进行测试的过程中,发现了与协议标准不一致的地方。在后续的工作中,将对于测试目的自动生成进行研究,同时针对测试中发现的不一致的地方对被测实现进行分析。

参考文献

- 1 RFC 1771. A Border Gateway Protocol 4 (BGP-4), March 1995
- 2 RFC 2858. Multiprotocol Extensions for BGP 4, June 2000
- 3 赵邑新,韩博,等. 路由协议测试研究. 计算机工程, 2000, 10(增刊): 769~774
- 4 赵邑新,吴建平,韩博. 路由协议测试研究——边界网关路由协议 BGP-4 测试. 通信学报, 2001, 22(9): 91~98
- 5 赵邑新,吴建平,韩博. 基于协议集成测试系统 BGP-4 协议测试. 计算机工程, 2001, 27(8): 12~14
- 6 Henniger O, Lu M, Ural H. Automatic Generation of Test Purposes for Testing Distributed Systems. FATES, 2003, 178~191
- 7 RFC 2842. Capabilities Advertisement With BGP-4, May 2000
- 8 ISO/IEC 9646-2. Information Processing Systems. Open System Interconnection, OSI Conformance Testing Methodology and Framework, Part 2: Abstract test suite specification, 1991
- 9 Sidhu D P, Leung T. Formal methods for protocol testing: A detailed study. IEEE Transactions on Software Engineering, 1989, 15(4): 413~426

(上接第 35 页)

```
STATUS sysFecEnetAddrGet ( UINT32 motCpmAddr, UCHAR *
addr)
{
.....
/* 'mac-valid' offset in NVRAM is 0xF0. The value 0x5a5a
says the mac in NVRAM is valid. */
if (sysNvRamGet_Block((char *)&mac_valid, 2, 0xf0) =
= OK)
{
if (mac_valid == 0x5a5a)
{
if (sysNvRamGet_Block((char *) (my_mac), 6,
0xf2) == OK)
{
tmp_mac = my_mac;
}
}
else
{
tmp_mac = sysFecEnetAddr;
}
}
.....
```

随后在文件 bootConfig.c 操作接口函数 bootCmdLoop() 中增加新命令“k”来执行修改 FLASH 中 MAC 地址的命令。

```
switch (* (pLine++)) {
.....
case 'k';
my_change_mac();
break;
.....
```

最后在 BSP 的 bootConfig.c 中加入函数 my_change_mac(), 用来改变 FLASH 中的 MAC 地址。

```
LOCAL void my_change_mac()
{
.....
if ( fioRdString ( STD_IN, input_string, MAX_INPUT )! =
EOF)
{
pInput = input_string;
tok = strtok_r (macaddr, "-", &holder);
addr[0] = strtol (tok, (char **)NULL, 16);
tok = strtok_r (NULL, "-", &holder);
addr[1] = strtol (tok, (char **)NULL, 16);
tok = strtok_r (NULL, "-", &holder);
.....
bcopy((char *)add, (char *) (my_mac), 6);
mac_valid = 0x5a5a;
sysNvRamSet_Block((char *) (my_mac), 6, 0xf2);
sysNvRamSet_Block((char *)&mac_valid, 2, 0xf0);
.....
```

3.2 IP 地址配置方法的实现

目标机的 IP 默认地址则通过 BSP 中的 config.h 文件进

行设置,其中 DEFAULT_BOOT_LINE 为 vxworks 下载

的默认路径, e 为目标机的默认 IP 地址。

```
#define DEFAULT_BOOT_LINE \
"motfec (0, 0) host; /target/config/105a/vxWorks e = 192.
168. 123. 1; ffff0000 h=192. 168. 200. 142 u=target pw=vx-
works"
```

当 FLASH 中 BOOT_LINE_ADRS 没有设置时, bootConfig.c 中 usrBootLineInit() 函数将 DEFAULT_BOOT_LINE 拷贝到 BOOT_LINE_ADRS。此时就采用默认的 IP 地址。

当需要改变下载的 IP 地址时,通过文件 bootConfig.c 操作接口函数 bootCmdLoop() 中命令“c”执行。

```
case 'c': /* change boot params */
bootParamsPrompt (BOOT_LINE_ADRS);
(void)sysNvRamSet_Block (BOOT_LINE_ADRS,
strlen (BOOT_LINE_ADRS) + 1, 0);
break;
```

结束语 通过以上 MAC 地址和 IP 地址的动态配置方法,可以有效地克服静态配置方法中存在的各种问题。相对于静态配置方法,动态配置方法结合了 FLASH 和 ROM 的灵活应用,有以下优点:

- ①能成批量生产内容一致的 ROM,降低了系统成本,缩短了生产周期。
- ②各个处理机板的调试步骤一致,简单易行。
- ③所申请的 MAC 地址得到有效利用,节省了地址资源。

随着分布式系统的应用日益广泛和设计的日益复杂,动态配置方法所提供的低生产成本、短生产周期和高地址资源利用率以及简单易行的调试方法等优点将日益凸显,因而具有良好的应用前景。

参考文献

- 1 Tanenbaum A S, van Steen M. Distributed Systems - Principle and Paradigms. Tsinghua University Press, 2002
- 2 Comer D E. Internetworking With TCP/IP. Prentice-Hall Inc, 2000
- 3 刘兵,刘欣. Internet 技术与应用教程. 北京:中国水利水电出版社, 2003
- 4 孔祥营, 柏桂枝. 嵌入式实时操作系统 VXWORKS 及其开发环境 TORNADO. 北京:中国电力出版社, 2001