# PartⅠ面向黑盒测试的测试用例设计方法

黑盒测试即数据驱动的测试，其测试数据完全来源于软件规范

## 等价类划分

使用等价划分方法设计测试用例主要有两个步骤: (1) 确定等价类; (2)生成测试用例。

**确定等价类**

确定等价类是选取每一个输入条件并将其划分为两个或更多的组。注意，我们确定了两类等价类:有效等价类代表对程序的有效输入，而无效等价类代表的则是其他任何可能的输入条件(即不正确的输入值)。这样，我们遵循了本书第阐述的测试原则，即要注意无效和未预料到的输入情况。

在给定了输入或外部条件之后，确定等价类大体上是一个启发式的过程。下面给出了一些指导原则:

1.如果输入条件规定了一个取值范围(例如，"数量可以是从 999") ，那么就应确定出一个有效等价类( 1<数量<999) ，以及两个无效等价类(数量<1数量>999)

2. 如果输入条件规定了取值的个数(例如，"汽车可登记一至六名车主" ) ，那么就应确定出→个有效等价类和两个无效等价类(没有车主，或车主多于六个)。

3. 如果输入条件规定了一个输入值的集合，而且有理由认为程序会对每个值进行不同处理(例如，"交通工具的类型必须是公共汽车、卡车、出租车、火车或摩托车勺，那么就应为每个输入值确定一个有效等价类和一个无效等价类(例如，"拖车")。

1. 如果存在输入条件规定了"必须是"的情况，例如"标识符的第一个字符必须是字母"，那么就应确定一个有效等价类(首字符是宇母)和一个无效等价类(首字符不是字母)。如果有任何理由可以认为程序并未等同地处理等价类中的元素，那么应该将这个等价类再划分为小一些的等价类。

**生成测试用例**

第二步是使用等价类来生成测试用例，其过程如下:

1.为每个等价类设置一个不同的编号。

2. 编写新的测试用例，尽可能多地覆盖那些尚未被涵盖的有效等价类，直到所有的有效等价类都被测试用例所覆盖(包含进去)。

3. 编写新的用例，覆盖一个且仅一个尚未被涵盖的无效等价类，直到所有的无效等价类都被测试用例所覆盖。

用单个测试用例覆盖无效等价类，是因为某些特定的输入错误检查可能会屏蔽或取代其他输入错误检查。举例来说，如果规格说明规定了 "请输入书籍类型(硬皮、软皮或目页)及数量 (1- 999)" ，代表两个错误输入(书籍类型错误，数量错误)的测试用例 "XYZ 0" ，很可能不会执行对数量的检查，因为程序也许会提示"XYZ是未知的书籍类型"，就不检查输入的其余部分了。

实例：PPPoE上网功能测试

测试点分析

输入项：账号、密码

预期输出结果：登录成功，登录失败

用户名可以为：正确的用户名（有效）和不存在的用户名以及为空

密码可以为：匹配的密码（有效）和不正确的密码以及为空

其他场景：重复操作；权限问题...

**等价类划分--基于输入输出（功能层面）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 编号 | 有效等价类 | 编号 | 无效等价类 |
| 用户名 | 1 | 用户名正确 | 2 | 不存在的用户名 |
|  |  | 3 | 用户名为空 |
| 密码 | 4 | 密码正确 | 5 | 密码错误 |
|  |  | 6 | 密码为空 |
| 登录按钮 | 7 | 按下按钮 | 8 | 不按按钮 |
|  |  |  |  |

登录的测试点（用例）：

1. 正确的用户名和匹配的密码，按下按钮，登录成功； 覆盖：1、4、7
2. 不存在的用户名，按下按钮，登录失败；覆盖2、7
3. 为空的用户名，按下按钮，登录失败，提示用户名不能为空；覆盖3、7
4. 不正确的密码，按下按钮，登录失败，提示账号或密码错误，按下按钮；覆盖 5、7
5. 密码为空，按下按钮，登录失败；覆盖 6、7
6. 正确的用户名和匹配的密码，不按按钮，登录失败，无错误提示； 覆盖：1、4、8

**错误猜测--基于场景（功能层面）**

登录的场景用例

1. 密码连续错误多次，账号锁定；
2. 如果是锁定的账号，登录会提示该用户已被锁定

**登录的非功能用例（非功能层面）**

* 性能用例：关注快不快，也就是响应时间
* 检查登录的响应时间和登录效率（是否在1s内完成操作）
* 并发多个用户进行登录，检查响应时间等待特性是否满足(1s,3s,5s)
* 安全性测试用例：
* 密码显示应该为密文，有被加密；
* 验证码的测试....
* 界面测试:
* 检查界面文字是否正确，是否存在错别字
* 检查输入框、按钮是否对应，是否可以输入合适的内容
* 整个系统的界面包括登录，是否做到了风格和格式的统一
* 兼容性测试：
* 检查登录功能是否能兼容业界主流的浏览器（**IE、谷歌**、360、Firefox等）
* 如果是APP，则是测试不同手机设备：安卓、苹果
* 易用性测试
* 检查登录时，整个操作过程是否方便用户使用，提示信息是否友好。
* 接口测试（没有界面，直接通过测试工具提交数据到后端）
* 检查登录时，接口是否正确，接口测试工具：Postman
* 检查的接口入参是用户名、密码

标准的测试点描述---沟通面试

1. 不要说“测试用户名对不对”“测试密码”之类的

应该像测试用例的标题那样描述：数据+预期结果，如：输入正确的用户名和密码，登录成功；

什么是接口测试？（接口实现功能，界面显示数据）

一个系统但凡有功能的地方都是接口，针对每一个功能去抓一个http请求，把这个请求放在Jmeter、postman里面去测试就是接口测试。

为什么要做接口测试？

通过接口可以跳过（越过）UI层面的校验，从而可以达到UI层面测试不到的东西，我们在接口层面可以去测试；接口可以更早的介入我们的测试，如果说我们只是做功能或者UI测试的话，我们是不是一定得要等这个系统已经做好了并且已经部署好了之后，我们才能够进行？因为这个东西没出来的话，你怎么去做功能测试，怎么去输入数据，怎么去得出结果去判断这功能是不是正确的。但是我们在接口测试的时候，你这个系统的界面没有完全做出来，我们也可以去做测试，只要你后台的一些接口调用已经是OK的，那么我们就可以去进行接口测试。这就相当于比我们的功能测试要提前介入了一定的时间，既然可以提前介入进行接口测试的话，是不是在一定程度上保证了我们系统的稳定性？

## 边界值分析

经验证明，考虑了边界条件的测试用例与其他没有考虑边界条件的测试用例相比，具有更高的测试回报率。所谓边界条件，是指输入和输出等价类中那些恰好处于边界、或超过边界、或在边界以的状态。边界值分析方法与等价划分方住存在两方面的不同:

1. 与从等价类中挑选出任意一个元素作为代表不同，边界值分析需要选择一个或多个元素，以便等价类的每个边界都经过一次测试。
2. 与仅仅关注输入条件(输入空间)不同，还需要考虑从结果空间(输出等价类)设计测试用例。

一些通用指南:

1. 如果输入条件规定了一个输入值范围，那么应针对范围的边界设计测试用例，针对刚刚越界的情况设计无效输入测试用例。举例来说，如果输入值的有效范围是-1. 0至+1.0 ，那么应针对-1.0、+1.0 、-1.001 、+1.001 的情况设计测试用例。
2. 如果输入条件规定了输入值的数量，那么应针对最小数量输入值、最大数量输入值，以及比最小数量少一个、比最大数量多一个的情况设计测试用例。举例来说，如果某个输入文件可容纳1-255 条记录，那么应根据0、1、255、256 条记录的情况设计测试用例。
3. 对每个输出条件应用指南a。举例来说，如果某个程序按月计算FICA 的扣除额，且最小金额是$0.00 ，最大金额为$1 165.25 ，那么应该设计测试用例来测试扣除$0.00和 $1 165.25 的情况。此外，还应观察是否可能设计出导致扣除金额为负数或超过$1 165.25 的测试用例。

注意，检查结果空间的边界很重要，因为输入范围的边界井不总是能代表输出范围的边界情况。

1. 对每个输出条件应用指南b。如果某个信息检索系统根据输入请求显示关联程度最高的信息摘要，而摘要的数量从未超过4条，则应编写测试用例， 使程序显示0条、1条和4条摘要，还应设计测试用例，导致程序错误地显示5条摘要。
2. 如果程序的输入或输出是一个有序序列(例如顺序的文件、线性列表或表格) ，则应特别注意该序列的第一个和最后一个元素。
3. 此外，发挥聪明才智找出其他的边界条件。

## 因果图分析

定义

因果图（Cuase-effect Graph）是一种描述输入条件的组合以及每种组合对应的输出的图形化工具。在因果图的基础上可以设计测试用例。

因果图法设计测试用例的步骤

1、分析程序的规格说明书中哪些是原因，哪些是结果。所谓原因，是指输入条件或输入条件的等价类，而结果是指输出条件。给每一个原因和结果赋一个标识符。

2、分析程序规格说明书中的语义，确定原因与原因，原因与结果之间的关系，画出因果图。

3、由于语法环境的限制，一些原因与原因之间，原因与结果之间的组合不能出现。对于这些特殊情况，在因果图中用一些记号标明约束或限制条件。

4、将因果图转化为判定表。

5、根据判定表的每一列设计测试用例。

当然，若能直接得到判定表，可以直接根据判定表设计测试用例。

实例1 WAN口PPPoE登录功能需求

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*因果图\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

输入： 输出：

1. 输入正确用户名 1、登录成功
2. 输入错误用户名 2、登录失败，错误提示
3. 输入用户名为空 3、错误提示

4、输入密码错误 4、登录失败，无提示

5、输入密码正确

6、输入密码为空

7、点击应用按钮 条件约束：(1,2)(1,3)(2,3)(4,5)(4,5)

8、未点击应用按钮 (4,6)(5,6)(7,8)(9,10)

9、只输入用户名 (9,11)(10,11)不可组合

10、只输入密码

11、只按应用按钮

条件项

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 条件桩 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |  |
| 正确用户名 | Y | Y | Y | Y | Y | Y |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 错误用户名 |  |  |  |  |  |  | Y | Y | Y | Y | Y | Y |  |  |  |  |  |  |  |
| 用户名为空 |  |  |  |  |  |  |  |  |  |  |  |  | Y | Y | Y | Y | Y | Y |  |
| 密码错误 | Y | Y |  |  |  |  | Y | Y |  |  |  |  | Y | Y |  |  |  |  |  |
| 密码正确 |  |  | Y | Y |  |  |  |  | Y | Y |  |  |  |  | Y | Y |  |  |  |
| 密码为空 |  |  |  |  | Y | Y |  |  |  |  | Y | Y |  |  |  |  | Y | Y |  |
| 点击按钮 | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  |  |
| 未点击按钮 |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  | Y |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 动作桩 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 登录成功 |  |  | √ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 登录失败，错误提示 | √ |  |  |  | √ |  | √ |  | √ |  | √ |  | √ |  | √ |  | √ |  |  |
| 登录失败，无提示 |  | √ |  | √ |  | √ |  | √ |  | √ |  | √ |  | √ |  | √ |  | √ |  |

通过因果图画出的判定表没必要也不能简化这个表

测试用例

|  |  |  |
| --- | --- | --- |
| 测试用例 | 输入 | 输出 |
| 01 | 正确用户名，错误密码，点击按钮 | 登录失败，错误提示 |
| 02 | 正确用户名，错误密码，不点击按钮 | 登录失败，无错误提示 |
| 03 | 正确用户名，正确密码，点击按钮 | 登录成功 |
| 04 | 正确用户名，正确密码，不点击按钮 | 登录失败，无错误提示 |
| 05 | 正确用户名，密码为空，点击按钮 | 登录失败，错误提示 |
| 06 | 正确用户名，密码为空，不点击按钮 | 登录失败，无错误提示 |
| 07 | 错误用户名，错误密码，点击按钮 | 登录失败，错误提示 |
| 08 | 错误用户名，错误密码，不点击按钮 | 登录失败，无错误提示 |
| 09 | 错误用户名，正确密码，点击按钮 | 登录失败，错误提示 |
| 10 | 错误用户名，正确密码，不点击按钮 | 登录失败，无错误提示 |
| 11 | 错误用户名，密码为空，点击按钮 | 登录失败，错误提示 |
| 12 | 错误用户名，密码为空，不点击按钮 | 登录失败，无错误提示 |
| 13 | 用户名为空，错误密码，点击按钮 | 登录失败，错误提示 |
| 14 | 用户名为空，错误密码，不点击按钮 | 登录失败，无错误提示 |
| 15 | 用户名为空，正确密码，点击按钮 | 登录失败，错误提示 |
| 16 | 用户名为空，正确密码，不点击按钮 | 登录失败，无错误提示 |
| 17 | 用户名为空，密码为空，点击按钮 | 登录失败，错误提示 |
| 18 | 用户名为空，密码为空，不点击按钮 | 登录失败，无错误提示 |

## 错误猜测

## 接口测试

1. **冒烟测试**

冒烟测试是指对一个软件进行系统大规模的测试之前，先验证一下软件的基本功能是否能实现，是否具备可测性

# PartⅡ面向白盒测试的测试用例设计方法

一、语句覆盖

二、判定覆盖

三、条件覆盖

四、判定/条件覆盖

五、多重条件覆盖

**Part Ⅲ 更高级别的测试**

## 3.1功能测试

等价类划分方桂、边界值分析方怯、因果图分析方法和错误猜测方法尤其适合于功能测试

## 3.2系统测试

# Part Ⅳ 测试工作的一般流程

## 一、测试策略

先使用黑盒测试方式来设计测试用例，然后视情况需要使用白盒测试方式来设计补充的测试用例。

针对某个物品/功能进行测试。

万能思路：功能测试 + 界面测试 + 性能测试 + 兼容性测试 + 易用性测试 + 安全测试

在编写测试用例时，选择使用哪种测试方法（如等价类划分、因果图、边界值分析等）通常取决于测试的目标、需求的复杂性以及测试的类型。不同的测试方法可以相互补充，以确保测试覆盖面广且有效。

**一种策略**

1.如果规格说明中包含输入条件组合的情况，应首先使用因果图分析方法。

2. 在任何情况下都应使用边界值分析方注。应记住，这是对输入和输出边界进行的分析。边界值分析可以产生一系列补充的测试条件，但是，也正如"因果图分析"一节所述，多数甚至全部条件都可以被整合到因果图分析中。

3. 应为输入和输出确定有效和无效等价类，在必要情况下对上面确认的测试用例进行补充。

4. 使用错误猜测技术增加更多的测试用例。

5. 针对上述测试用例集检查程序的逻辑结构。应使用判定覆盖、条件覆盖、判定/条件覆盖或多重条件覆盖准则(最后的一个最为完整)。如果覆盖准则未能被前四个步骤中确定的测试用例所满足，并且满足准则也并非不可能(由于程序的性质限制，某些条件的组合也许是不可能实现的) ，那么增加足够数量的测试用例，以使覆盖准则得到满足。

## 二、回归测试

**原则7: 应避免测试用例用后即弃，除非软件本身就是一个一次性的软件**。

这个问题在采用交互式系统来测试软件时最常见。人们通常会坐在终端前，匆忙地编写测试用例，然后将这些用例交由程序执行。这样做的问题在于，饱含我们宝贵投入的测试用例，在测试结束后就消失了。一旦软件需要重新测试(例如，当改正了某个错误或作了某种改进后) ，又必须重新设计这些测试用例。情况往往是这样的，由于重新设计测试用例需要投入大量的工作，人们总是避免这样做。因此，对该程序的重新测试极少会同上次一样严格。这就意味着，如果对程序的更改导致了程序某个先前可以执行的部分发生了故障，这个故障往往是不会被发现的。保留测试用例，当程序其他部件发生更动后重新执行，这就是我们所谓的"回归测试"。--**保留测试用例，二次执行**

## 三、测试结束准则

软件测试的结束点可以由以下10个原则（条件）确定：

1、基于“测试阶段”的原则

每个软件的测试一般都要经过单元测试、集成测试、系统测试这几个阶段，我们可以分别对单元测试、集成测试和系统测试制定详细的测试结束点。每个测试阶段符合结束标准后，再进行后面一个阶段的测试。举个例子来说：单元测试，我们要求测试结束点必须满足“核心代码100％经过Code Review”、“功能覆盖率达到100％”、“代码行覆盖率不低于80％”、“不存在A、B类缺陷”、“所有发现缺陷至少60％都纳入缺陷追踪系统且各级缺陷修复率达到标准”等等标准。集成测试和系统测试的结束点也应这样制定。

2、基于“测试用例”的原则

测试设计人员设计完测试用例，并邀请项目组成员参与评审，测试用例一旦评审通过，后面测试时，就可以作为测试结束的一个参考标准。比如说在测试过程中，如果发现测试用例通过率太低，可以拒绝继续测试，待开发人员修复后再继续。在功能测试用例通过率达到100%，非功能性测试用例达到95%以上，允许正常结束测试。

使用该原则作测试结束点时，把握好测试用例的质量，非常关键。

3、基于“缺陷收敛趋势”的原则

软件测试的生命周期中随着测试时间的推移，测试发现的缺陷图线，首先成逐渐上升趋势，然后测试到一定阶段，缺陷又成下降趋势，直到发现的缺陷几乎为零或者很难发现缺陷为止。我们可以通过缺陷的趋势图线的走向，来定测试是否可以结束，这也是一个判定标准。

这个原则对迭代测试版本控制有较高的要求，如不建议超过5次迭代，否则在时间上得不偿失。

4、基于“缺陷修复率”的原则

软件缺陷（bug）在测试生命周期中我们分成几个严重等级，如：致命级、严重级、主要、较小和测试建议5种。那我们在确定测试结束点时，严重bug和致命bug的缺陷修复率必须达到100%；主要bug和较小bug的缺陷修复率必须达到85%以上，允许存在少量功能缺陷，后面版本解决；测试建议的缺陷修复率最好达到50%~70%以上。

该原则的使用，按具体公司具体项目的实际情况而定。比如测试建议，有的公司（或项目）相当重视，可能会有较高修复率的要求，而一些公司（或项目）可能不会做要求。bug等级划分上，各公司的划分也不尽相同。

5、基于“验收测试”的原则

很多公司都是做项目软件，如果这种要确定测试结束点，最好测试到一定阶段，达到或接近测试部门指定的标准后，就递交用户做验收测试。如果通过用户的测试验收，就可以立即终止测试部门的测试；如果客户验收测试时，发现了部分缺陷，就可以针对性的修改缺陷后，验证通过后递交客户，相应测试也可以结束。

此原则适用于非自主性开发项目，或有明确客户的项目。

6、基于“覆盖率”的原则

对于测试“覆盖率”的原则，个人觉得只要测试用例的“覆盖率”覆盖了客户提出全部的软件需求，包括行业隐性需求、功能需求和性能需求等等，只要测试用例执行的覆盖率达到100%，基本上测试就可以结束。如果你不放心，非得要看看测试用例的执行效果，检查是否有用例被漏执行的情况，可以对常用的功能进行“抽样测试 ”和“随机测试”。对于覆盖率在单元测试、集成测试和系统测试，每个阶段都不能忽略。

7、基于“项目计划”的原则

大多数情况下，每个项目从开始就要编写开发和测试的计划，相应的在测试计划中也会对应每个里程碑，对测试进度和测试结束点做一个限制，一般来说都要和项目组成员（开发、管理、测试、市场、销售人员）达成共识，团队集体同意后制定一个标准结束点。如果项目的某个环节延迟了，测试时间就相应缩短。大多数情况下是所有规定的测试内容和回归测试都已经运行完成，就可以作为一个结束点。很多不规范的软件公司，都是把项目计划作为一个测试结束点。如果把它作为一个结束点，测试风险较大，软件质量很难得到保证。此种情况，建议遵循“2·8”定律，优先级较高的功能模块测试覆盖率应尽量达到100%。

8、基于“缺陷度量”的原则

这个原则也许大家用的不是很多，了解比较少。我们可以对已经发现的缺陷，运用常用的缺陷分析技术和缺陷分析工具，用图表统计出来，方便查阅，分时间段对缺陷进行度量。我们可以把 “测试期缺陷密度”和 “运行期缺陷密度”作为一个结束点。当然，最合适的测试结束的准则应该是“缺陷数控制在一个可以接受的范围内”。比如说：一万行代码最多允许存在多少个什么严重等级的错误，这样比较好量化，比较好实施，成为测试缺陷度量的主流。

9、基于“质量成本”的原则

一个软件往往要从“质量、成本、进度”三方面取得平衡后就停止。至于这三方面哪一项占主要地位，就要看是什么软件了。比如说是：人命关天的航天航空软件, 那还是质量重要些，就算多花点钱、推迟一下进度，也要保证较高质量以后才能终止测试，发布版本。如果是一般的常用软件，由于利益和市场的原因，哪怕有bug，也必须得先推出产品。通常最主要的参考依据是：“找缺陷耗费的代价和这个缺陷可能导致的损失做一个权衡”。

具体操作的时候，可以根据公司实际情况来定义什么样的情况下算是“测试花费的代价最划算、最合理”，同时保证公司利益最大化。如果找bug的成本比用户发现bug的成本还高，也可以终止测试。

10、基于“测试行业经验”的原则

很多情况下，测试行业的一些经验，也可以为我们的测试提供借鉴。比如说测试人员对行业业务的熟悉程度，测试人员的工作能力，测试的工作效率等等都会影响到整个测试计划的执行。如果一个测试团队中，每个人都没有项目行业经验数据积累，拿到一个新的项目，自然是一头雾水，不知道从何处开始，测试质量自然不会很高。因此通过测试者的经验，对确认测试执行和结束点也会起到关键性的作用。

这条原则谨慎使用，在创业性项目可以考虑。

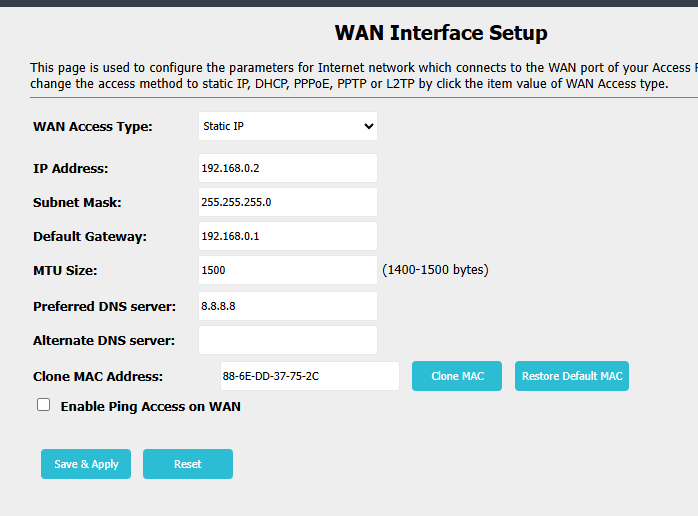
通常自主产品研发项目会采用1、2、4、6、9；非自主产品项目会采用1、2、4、5、6、7；以前的传统模式开发项目还会加上3。当然，这些原则是可以根据具体情况自由组合的

# Part Ⅴ 路由器测试用例编写实例

## 功能子项：WAN口静态IP上网

实例2 WAN口静态IP上网功能

设置界面如下：

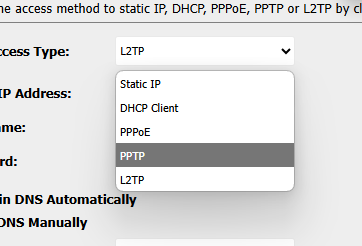


分析：

输入框功能校验：1、多个输入组合，应该用因果图分析（该方法主要适用于各种输入条件之间存在某种相互制约关系或输出结果依赖于各种输入条件的组合时的情况）；2、只要有输入框，必用边界值

1. 必要情况下用等价类补充 4、错误猜测 5、检查程序的逻辑结构

其他方面：界面测试 + 性能测试 + 兼容性测试 + 易用性测试（自动填充掩码） + 安全测试



因果图分析

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*因果图\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

输入： 输出：1、设置成功

1、IP Address正确 2、设置失败，无提示

2、IP Address错误（包括空）（与网关不在一个网段） 3、设置失败，有错误提示

3、Subnet Mask正确 4、清空设置

4、Subnet Mask错误（包括空）

5、Default Gateway正确 条件约束（1、2、3、4）（5、6、7）

6、Default Gateway错误（包括空） （8、9、10）（11、12、13）

7、Preferred DNS server正确（合法IP，但不一定可用） （20、21）（22、23）两两互斥

8、Preferred DNS server错误（包括空） （非法IP）

9、Alternate DNS server正确

10、Alternate DNS server错误（包括空）

11、点击Save & Apply 按钮

12、不点击Save & Apply按钮

13、点击Reset按钮

路由器LAN口没有IP地址，整个路由器只有一个网关（LAN侧IP），和一个WAN侧IP地址，网关地址是连接不同LAN侧；，路由器的LAN口本身并不需要有自己的IP地址，因为它们主要用于连接内部网络设备。但是，路由器通常会有一个管理IP地址，用于远程登录和配置路由器

1. 编写新的测试用例，尽可能多地覆盖那些尚未被涵盖的有效等价类，直到所有的有效等价类都被测试用例所覆盖(包含进去)。

2. 编写新的用例，覆盖一个且仅一个尚未被涵盖的无效等价类，直到所有的无效等价类都被测试用例所覆盖。

PS：根据IP没办法判断子网掩码设置得对不对

**等价类划分--基于输入输出（功能层面）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入条件 | 编号 | 有效等价类 | 编号 | 无效等价类 |
| IP Address | 1 | IP正确 | 2 | IP错误 |
|  |  | 3 | IP为空 |
| Subnet Mask | 4 | 掩码正确 | 5 | 掩码错误 |
|  |  | 6 | 掩码为空 |
| Default Gateway | 7 | 网关正确 | 8 | 网关错误 |
|  |  | 9 | 网关为空 |
| Preferred DNS server | 10 | 首选DNS正确 | 11 | 首选DNS错误 |
|  |  | 12 | 首选DNS为空 |
| Alternate DNS server | 13 | 备选DNS正确 | 14 | 备选DNS错误 |
|  |  | 15 | 备选DNS为空 |
| Save & Apply 按钮 | 16 | 按下按钮 | 17 | 不按按钮 |
|  |  |  |  |
| Reset按钮 | 18 | 按下按钮 | 19 | 不按按钮 |
|  |  |  |  |

注：表中的正确或错误是指针对WAN口所连的网络，若WAN口所连上级路由器的网关是192.168.55.1，子网掩码是255.255.255.0；则WAN口的掩码长度不能低于上级网络的掩码长度，可大于等于;

若配置WAN口静态IP上网时，其子网掩码长度低于上行网络，则为“掩码错误”

测试用例

1. 正确IP，正确掩码、正确网关、首选DNS正确、备选DNS正确、按下按钮，设置成功；覆盖1、4、7、10、13、16；
2. 正确IP，错误掩码、正确网关、首选DNS正确、备选DNS正确、按下按钮，设置失败，提示“掩码错误”；覆盖1、5、7、10、13、16；
3. 正确IP，空掩码、正确网关、首选DNS正确、备选DNS正确、按下按钮，设置失败，提示“掩码不能为空”；覆盖1、6、7、10、13、16；
4. 错误IP，正确掩码、正确网关、首选DNS正确、备选DNS正确、按下按钮，设置失败，提示“IP地址应该和网关在一个网段”；覆盖2、4、7、10、13、16
5. 空IP，正确掩码、正确网关、首选DNS正确、备选DNS正确、按下按钮，设置失败，提示“IP地址不能为空”；覆盖3、4、7、10、13、16；
6. 正确IP，正确掩码、错误网关、首选DNS正确、备选DNS正确、按下按钮，设置失败，提示“网关不合法，应该跟IP地址在同一个网段”；覆盖1、4、8、10、13、16；
7. 正确IP，正确掩码、空网关、首选DNS正确、备选DNS正确、按下按钮，设置失败，提示“网关不能为空”；覆盖1、4、9、10、13、16；

7-1、正确IP，正确掩码、空网关、首选DNS正确、备选DNS正确、按下按钮，设置成功，自动填充网关地址为0.0.0.0；覆盖1、4、9、10、13、16；

1. 正确IP，正确掩码、正确网关、首选DNS错误、备选DNS正确、按下按钮，设置成功；覆盖1、4、7、11、13、16；
2. 正确IP，正确掩码、正确网关、首选DNS为空、备选DNS正确、按下按钮，设置失败，提示“首选DNS不能为空”；覆盖1、4、7、12、13、16
3. 正确IP，正确掩码、正确网关、首选DNS正确、备选DNS错误、按下按钮，设置成功；覆盖1、4、7、10、14、16；
4. 正确IP，正确掩码、正确网关、首选DNS正确、备选DNS为空、按下按钮，设置成功；覆盖1、4、7、10、15、16；
5. 正确IP，正确掩码、正确网关、首选DNS正确、备选DNS正确、不按按钮，设置失败，不提示任何信息；覆盖1、4、7、10、15、17；
6. 正确IP，正确掩码、正确网关、首选DNS正确、备选DNS正确、按下Reset按钮，设置失败，不提示任何信息，输入内容清空；覆盖1、4、7、10、13、18；
7. 正确IP，正确掩码、正确网关、首选DNS正确、备选DNS正确、不按Reset按钮，设置失败，不提示任何信息；覆盖1、4、7、10、13、19；

其他情况下的用例补充：

两个DNS地址都错误

1. 正确IP，正确掩码、正确网关、首选DNS错误、备选DNS错误、按下按钮，设置成功；覆盖1、4、7、11、14、16；

性能测试用例：

界面测试用例：

兼容性测试用例：

易用性测试用例（自动填充掩码）

当条件桩组合起来数量比较多的时候，可以采取以下策略：

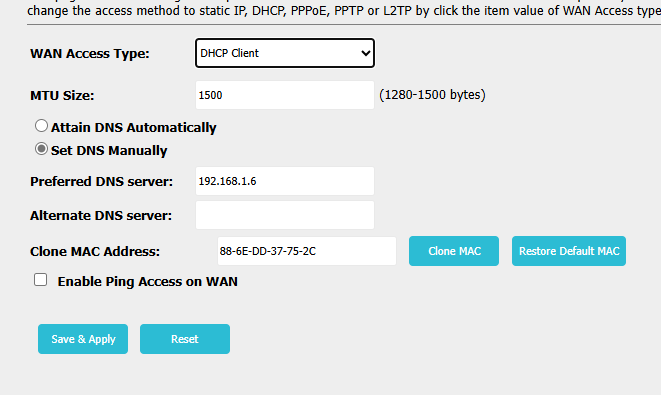
1. 优先级排序：根据业务逻辑的重要性和风险评估，对条件组合进行优先级排序。优先测试那些对业务影响最大的条件组合。
2. 将条件桩划分为等价类，每个等价类代表一组具有相同行为的输入条件。选择一个或几个代表性的条件组合来代表整个等价类。
3. 测试的目的是发现缺陷和验证功能，而不是生成尽可能多的测试用例。因此，选择和设计测试用例时，应该注重质量而非数量
4. 通过分析因果图，尝试减少测试用例的数量，同时仍然保持足够的覆盖率。识别并消除冗余的测试用例。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 条件桩 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| IP正确 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| IP错误 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 掩码正确 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 掩码错误 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 网关正确 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 网关错误 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 首DNS正确 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 首DNS错误 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 备DNS正确 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 备DNS错误 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 点击Save & Apply按钮 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 不点击Save & Apply按钮 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 点击Reset按钮 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 动作桩 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 设置成功 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 设置失败，有错误提示 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 设置失败，无错误提示 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 清空设置 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 二、功能子项--WAN口DHCP上网功能

实例3 WAN口DHCP上网功能

设置界面如下：



因果图

输入：

1. 自动获取DNS
2. 手工设置DNS
3. 首选DNS正确
4. 首选DNS错误
5. 首选DNS为空
6. 备选DNS正确
7. 备选DNS错误
8. 备选DNS为空
9. 保存应用按钮

输出：

1. 设置成功，能访问外网
2. 设置成功，不能访问外网
3. 设置失败，有提示信息

测试用例

1. 自动获取DNS，按下“保存并应用”，设置成功，能访问外网；
2. 手工设置DNS，首选DNS正确，备选DNS正确，按下“保存并应用”，设置成功，能访问外网。
3. 手工设置DNS，首选DNS错误，备选DNS正确，按下“保存并应用”，设置成功，能访问外网
4. 手工设置DNS，首选DNS为空，备选DNS正确，按下“保存并应用”，设置失败，提示“首选DNS不能为空”
5. 手工设置DNS，首选DNS正确，备选DNS错误，按下“保存并应用”，设置成功，能访问外网。
6. 手工设置DNS，首选DNS正确，备选DNS为空，按下“保存并应用”，设置成功，能访问外网。
7. 手工设置DNS，首选DNS错误，备选DNS错误，按下“保存并应用”，设置成功，不能访问外网；