

文章编号:1007-2985(2011)05-0048-04

一种基于 TCL/TK 的自动化测试框架^{*}

唐赞玉¹, 贾志宏²

(1. 吉首大学信息科学与工程学院, 湖南 吉首 416000; 2. 中铁二十局集团海外工程保障中心, 上海 201900)

摘 要:依据自动化测试理论,采用 TCL/TK,设计与实现了自动化测试框架,构建了自动化测试系统.实验证明,基于本文自动化测试框架构建的自动化测试系统能自动连接到测试仪,对 DUT 设备进行自动配置,并对配置结果进行判断,控制测试仪收发数据帧,对收到的结果进行判断并给出最终的测试结果.

关键词:软件测试;框架;自动化;TCL/TK;函数库

中图分类号:TP311.5

文献标志码:A

软件测试框架是用于指导软件测试实施的理论模型.该模型的优劣直接影响着测试的进度与质量,因此正确选取过程框架是开展软件测试需要考虑的首要问题^[1-2].自动化测试借助于测试软件和工具部分或全部代替人工进行测试,包括对测试活动的自动化管理与实施,自动化测试脚本的开发与执行,以及使用自动化测试工具来对测试需求进行验证.对于测试过程中需要大量重复或人工测试比较困难的测试过程,自动化测试可以提高测试效率,缩短产品发布时间,提高产品的质量^[3-4].

目前国内外存在多种多类型的软件测试工具或软件测框架^[5-7].其中 LoadRunner 是一种预测系统行为和性能的负载测试工具,通过模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题,能对整个企业架构进行测试. Mercury Interactive 公司的 WinRunner 是一种企业级的功能测试工具,用于检测应用程序是否能够达到预期的功能及正常运行,通过自动录制、检测和回放用户的应用操作,能有效地帮助测试人员对复杂的企业级应用的不同发布版进行测试,提高测试人员的工作效率和质量,确保跨平台的、复杂的企业级应用无故障发布及长期稳定运行. IBM Rational Solutions 是在 RUP 测试方法论的基础上构建的软件自动化测试工具集. TesDirector 是业界第 1 个基于 Web 的测试管理系统,可在公司内部或外部进行全球范围内测试的管理,通过在一个整体的应用系统中集成测试管理的各个部分,包括需求管理,测试计划,测试执行以及错误跟踪等功能.北京大学的王立教授在 2000 年提出一种界面类对象建模技术,体现了对象的封装、通信、动态特征.浙江大学的谭健荣教授在 2002 年提出界面构件关联图,利用构件的相互关系,提出测试覆盖准则和测试用例生成方法.笔者针对实际需要,依据自动化测试理论,采用 TCL/TK 设计与实现了自动化测试框架,构建了自动化测试系统.这对于软件测试具体应用具有重要的价值与意义.

* 收稿日期:2011-06-18

基金项目:湖南省科技厅基金项目(10c1099)

作者简介:唐赞玉(1978-),女,湖南桃江人,吉首大学信息科学与工程学院讲师,主要从事计算机网络、软件测试、通信安全等研究.

1 测试拓扑

自动化测试系统网络通常由 2 种相对独立的以太网构成:控制网络和测试网络.其中:通过 1 台或数台控制交换机连接各个设备构成控制网络,专门用于主控设备与各设备之间的通讯和控制,完成自动化配置设备、自动化操作设备、自动化设备状态检查、自动化收集日志;通过 1 台测试交换机来连接各测试仪器与待测设备中参与测试数据传输的各端口,构成测试网络,通过在测试交换机上设置虚拟局域网的方法来动态构造测试拓扑.

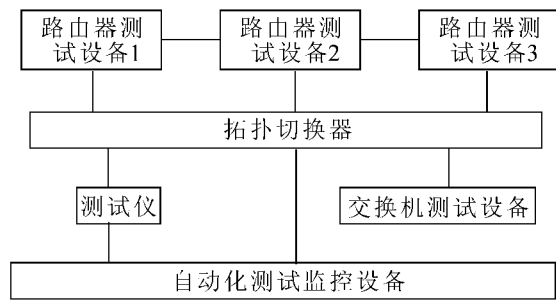


图 1 物理拓扑图

测试用的物理拓扑如图 1 所示,物理拓扑设备包括测试监控设备、拓扑切换器、测试仪、DUT(路由器测试设备和交换机测试设备).

- (1) 测试监控设备. 是运行自动化测试系统的工作 PC, 也可能是远程登陆专用的测试 PC.
- (2) 拓扑切换器. 由交换机堆叠而成, 实现一个透明连接. 通过将测试仪的端口和 DUT 端口划分到同一个 VLAN 来实现测试仪的端口与 DUT 的端口连接.
- (3) 测试仪. IXIA, 用来实现自动化测试发送和接收报文的测试.
- (4) DUT. 需要参与测试的路由器或交换机.
- (5) 控制网络. 测试监控设备、拓扑切换器、测试仪(IXIA)都连接到同一个网段, 由测试监控设备 PC 来实现对其他设备的通讯与控制.

(6) 测试网络. 测试仪(IXIA)、DUT 都与测试拓扑切换器相连, DUT 与 DUT 相连. 与拓扑切换器相连的 IXIA 端口和 DUT 端口划分到同一个 VALN 下实现测试拓扑的创建.

2 测试框架

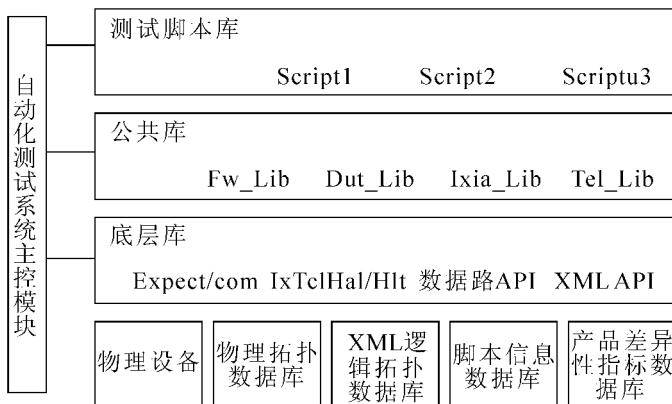


图 2 测试框架图

构建的自动测试框架结构如图 2 所示, 包括 1 个主控模块和 9 个库.

(1) 主控模块. 该模块负责控制整个自动化测试系统的运行, 负责加载公共库、底层库. 通过底层库 API 去连接和释放物理设备, 通过数据库 API 去读取数据库信息并进行初始化, 控制和监控测试脚本的运行过程.

(2) 测试脚本库. 测试脚本库是测试脚本的集合, 测试脚本是自动化测试人员根据测试用例编写的测试脚本.

(3) 公共库. 为了减少编写脚本的代码冗

余, 使编写人员更容易更快地编写测试脚本, 将一些公用的内容和经常使用的功能封装成函数.

- (4) 底层库. 直接跟物理设备、数据库、XML 操作相连的接口.
- (5) 物理设备库. 参与测试的设备, 包括测试仪 IXIA、DUT 设备、拓扑切换器.
- (6) 物理拓扑数据库. 保存着物理拓扑设备之间的连接关系, 设备相关属性定义的数据.
- (7) XML 逻辑拓扑数据库. 保存着测试脚本使用的一个逻辑拓扑. 逻辑拓扑上定义的设备名称和端口都是采用逻辑名称. 一个测试脚本必须有一个 XML 逻辑拓扑文件, 一个 XML 逻辑拓扑文件可以同时被几个测试脚本共用.
- (8) 测试脚本信息数据库. 该数据库保存着测试脚本的相关属性信息.
- (9) 产品差异指标数据库. 该数据库保存在测试不同网络产品时, 产品之间指标差异值.

3 处理流程

利用上述自动测试框架构建的自动测试系统的处理流程如图 3 所示.

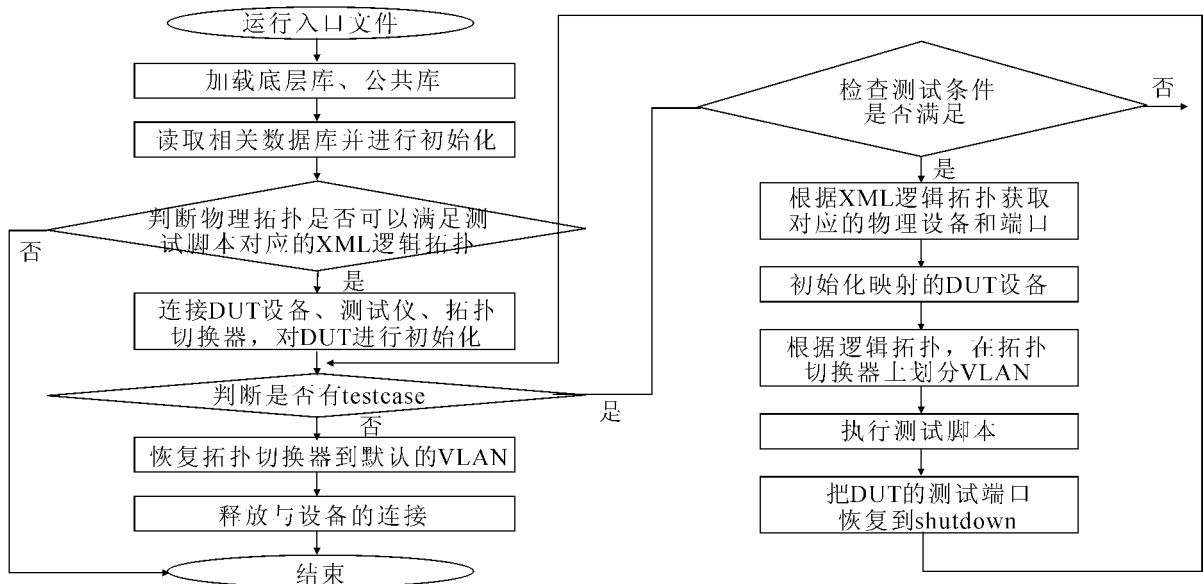


图 3 处理流程图

首先运行自动化测试系统主控模块 `main.tcl` 入口文件, 加载底层库和公共库, 通过底层库和公共库函数来读取数据库信息并进行初始化. 然后进行判断物理拓扑是否能够满足该测试脚本对应的 XML 逻辑拓扑. 如果满足就继续执行, 否则就结束. 通过底层库和公共库连接 DUT 设备、拓扑切换器、测试仪等, 并初始化 DUT 和拓扑切换器, 同时把 DUT 设备上的所有端口设置为 DOWN 状态.

通过读取全局设置的 `TestCaseList` 列表调用相应的测试脚本进行执行. 调用一个 `TestCase` 执行时, 先检查测试条件是否满足, 然后根据测试脚本信息数据库里设置的对应的 xml 逻辑拓扑到物理拓扑中去查找对应的测试仪及测试端口和 DUT 设备及测试端口等信息. 然后在拓扑切换器上切换 XML 对应的拓扑, 将参与测试的 DUT 设备的端口设置为 UP 状态. 最后开始 `TestCase` 内容的测试, 实现配置 DUT、控制测试仪收发数据帧、检测测试仪端口的帧收发情况、对结果进行判断等功能, 并输出 `TestCase` 的测试记录与 PASS/FAIL 的测试结果.

测试结束后自动化测试系统负责对 DUT 设备测试端口恢复 Down 状态, 把拓扑切换器的所有端口都恢复为默认的 VLAN 下, 释放测试仪和 DUT 设备的连接. 最后结束该测试用例的执行.

4 测试流程

具体的测试流程分为以下 7 个步骤:

(1) 定义测试脚本中测试拓扑. 定义测试脚本中测试拓扑, 统一用测试用例描述接口来定义, 目前已经统一定义的使用 `portGroup` 的数组结构来定义, 然后调用 `Fw_Create_PortGroup` 函数来获取实际的物理端口.

(2) 获取 `portGroup` 测试端口. `portGroup` 数组经过调用 `Fw_Create_ProtGroup` 处理后, `portGroup` 下的接口都获取到了对应的物理端口, 可以通过下面的 `portGroup` 数组结构来获取相关的测试端口等信息. `Fw_Create_ProGroup` 根据测试脚本里设置的以虚拟接口为索引的 `PortGroup` 数组关系, 从逻辑拓扑中获取测试端口, 设置到 `PortGroup` 数组中.

(3) 获取物理拓扑设备和连接关系. 获取物理拓扑的设备属性值, 可以通过调用 `fw_common.tcl` 里的 `Fw_Get_Device`. * 开头相关函数. 获取物理拓扑的设备端口连接和属性值, 可以通过调用 `fw_common.tcl` 里的 `Fw_Get_Top`. * 开头相关函数.

(4) 获取 XML 逻辑拓扑文件中信息. 获取 XML 逻辑拓扑文件中相关信息, 可以通过调用 `fw_com-`

mon.tcl 里的 Fw_Get_Xml.* 开头相关函数。

(5) 获取 tc_sat.csv 文件中信息. 获取 tc_sat.csv 文件中相关信息,可以通过调用 fw_common.tcl 里的 Fw_Get_Tc.* 开头相关函数。

(6) 获取 srs_sat.csv 文件和全局变量信息. 获取 srs_sat.csv 文件中相关信息和框架全局变量信息,可以通过调用 fw_common.tcl 里的 Fw_Get_Global.* 开头相关函数。

(7) 将测试信息输出到 LOG 文件中. 输出自动化测试执行过程中相关的 LOG 信息,可以通过调用 fw_common.tcl 里的 Fw_Print_.* 开头相关函数。

5 结语

根据文中构建的自动化测试系统能够实现自动连接到测试仪(IXIA)、对 DUT 进行自动配置、对配置结果进行判断、控制测试仪进行收发帧、对接收结果进行判断并最终给出测试例的测试结果(PASS/FAIL),测试过程不需要人工干预,极大地减少了测试资源的投入,提高了测试效率。

参考文献:

- [1] 聂长海,徐宝文,史亮.一种新的二水平多因素系统两两组合覆盖测试数据生成算法[J].计算机学报,2006,29(6):841-848.
- [2] 张德平,聂长海,徐宝文.基于 Markov 决策过程用交叉熵方法优化软件测试[J].软件学报,2008,19(10):2270-2279.
- [3] 曾敏,黄颖.基于自学习控制算法优化软件测试用例的研究[J].计算机工程与应用,2011,47(3):70-73.
- [4] 缪力.依赖性分析及其在软件测试中的应用[D].长沙:湖南大学,2005.
- [5] 杜庆峰,李娜.白盒测试基路径算法[J].计算机工程,2009(8):100-102,123.
- [6] 肖良,杨根兴,蔡立志.软件测试用例可复用性度量[J].计算机应用与软件,2010,27(6):46-49.
- [7] 施寅生,邓世伟,谷天阳.软件安全性测试方法与工具[J].计算机工程与设计,2008,29(1):27-29.

An Automated Testing Framework Based on TCL/TK

TANG Zan-yu¹, JIA Zhi-hong²

(1. School of Information Science and Engineering, Jishou University, Jishou 416000, Hunan China; 2. Overseas Projects Support Center, Chinese Railway 20th Bureau Group Corporation, Shanghai 201900, China)

Abstract: Based on automation testing theory and TCL/TK, a automated testing framework and system are designed and implemented. It is showed that, based on experiment, this automated testing system can automatically connect to tester, realize automatic configuration of DUTs equipment, and judge the configuration results, control tester sending and receiving data frames, make judgement on the testing results obtained and give the final test results.

Key words: software test; framework; automation; TCL/TK; function library

(责任编辑 易必武)