



北京邮电大学
Beijing University of Posts and Telecommunications

远程互操作性测试平台的设计与实现

论文标题

远程互操作性测试平台设计与实现

- 第一作者： 李思聪
- 第二作者： 黄小红(导师)





协议测试的重要性

随着网络应用的增加和QoS要求的提高，越来越多的网络协议出现在各个行业、各种场景当中。而协议的标准大多数都是使用自然语言描述，不同协议实现人员对于协议的理解，直接影响着协议的实现，甚至出现错误的实现。即便协议得到了正确的实现，也不能保证不同的实现彼此之间能够准确无误地通信。协议的错误实现或者错误的通信会导致基于协议的网络应用出现异常的错误而带来未知的后果。因此，为了提高协议的开发效率，促进标准化协议的实现，提高网络协议的可靠性和正确性，人们提出了协议测试这一概念：使用一种有效的方法来对协议实现进行判别，以验证协议实现与协议标准之间（一致性测试）、协议实现与协议实现之间的等价性（互操作性测试）。





传统互操作性测试局限性

在以往的实践中，为了对协议实现进行完备的测试，测试用户需要开发自己的测试系统或者求助于第三方测试，同时需要把测试系统和被测设备部署在同一本地网络内。这种限制一方面增加了测试的费用成本，另一方面也给测试方带来了诸如软件移植、人员出差等种种不便。我们中心之前进行过几次互操作性测试的活动，跟伦敦，巴黎那边合作，在这个过程中，我们的老师还有学生出差到国外，首先要跟各方的测试人员约定时间，并且要携带设备到一个会议室。然后，各方测试人员要摆出各种各样的设备，连接交换机进行一个局域网上的部署,再进行一对一的测试活动。这样的测试方式的成本是比较高的，而且需要测试各方高度的协调才能完成。





研究目标



平台架构与功能点



关键问题

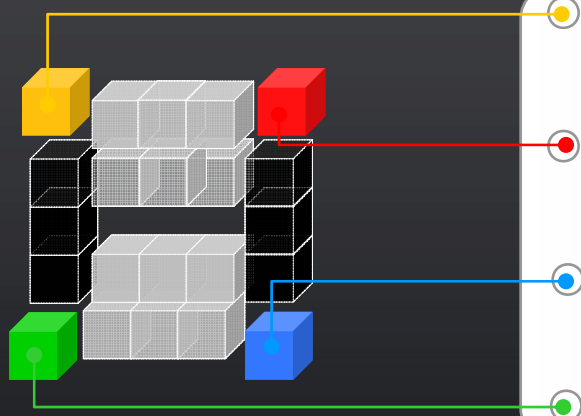


系统实现与验证



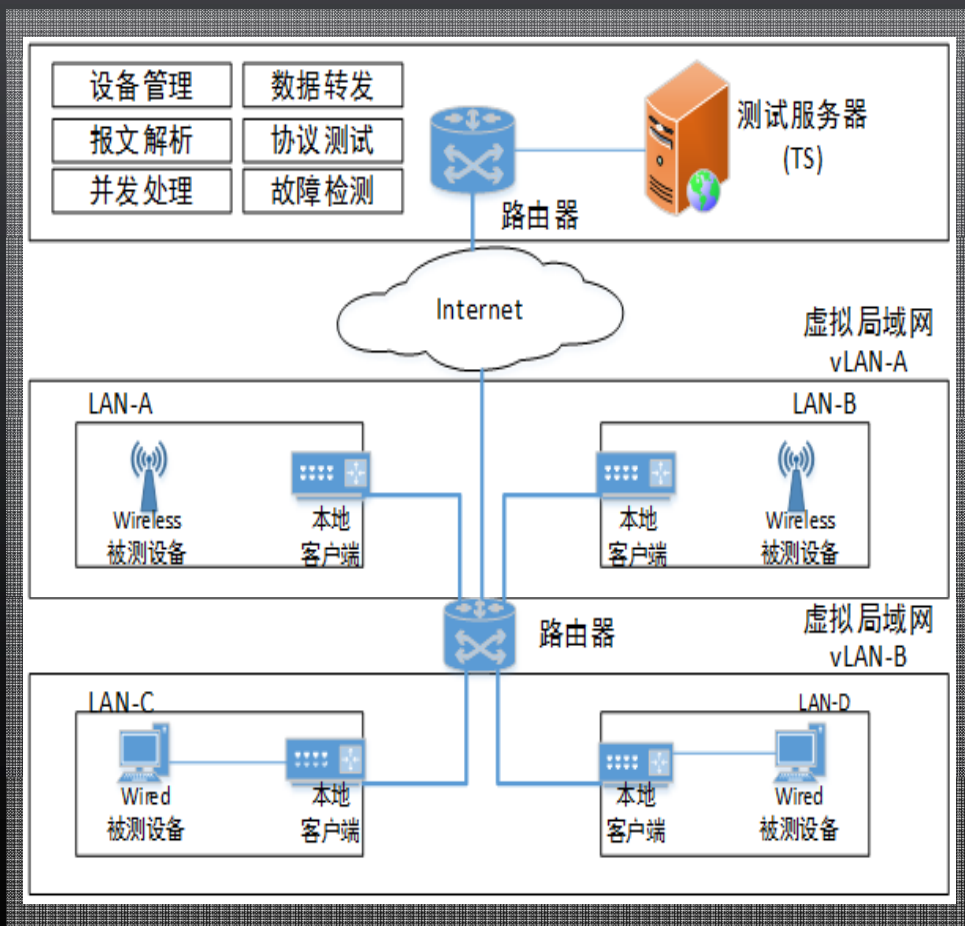
结论





- 提出一种远程互操作性测试平台架构
- 设计客户端和服务端相关功能
- 设计友好简便的交互界面
- 系统实现与验证





平台架构

如左图所示，存在多个本地局域网，它们分布在各自远程异地，每个局域网中都包含了被测设备以及平台客户端。客户端通过Internet连接到远程测试服务器上。通过远程服务器，可以将任意数量的局域网虚拟为一个本地局域网，使不同局域网中的被测设备好像处于同一本地网络之中。总的来说，远程服务器能够同时维护多个虚拟局域网，以满足多个测试同时进行的需求。





服务器端

服务器端的功能包括但不限于以下几个方面：

服务器端能集中管理注册过的设备，能实时查看在线设备。

服务器能管理每一次测试有哪些设备参与。

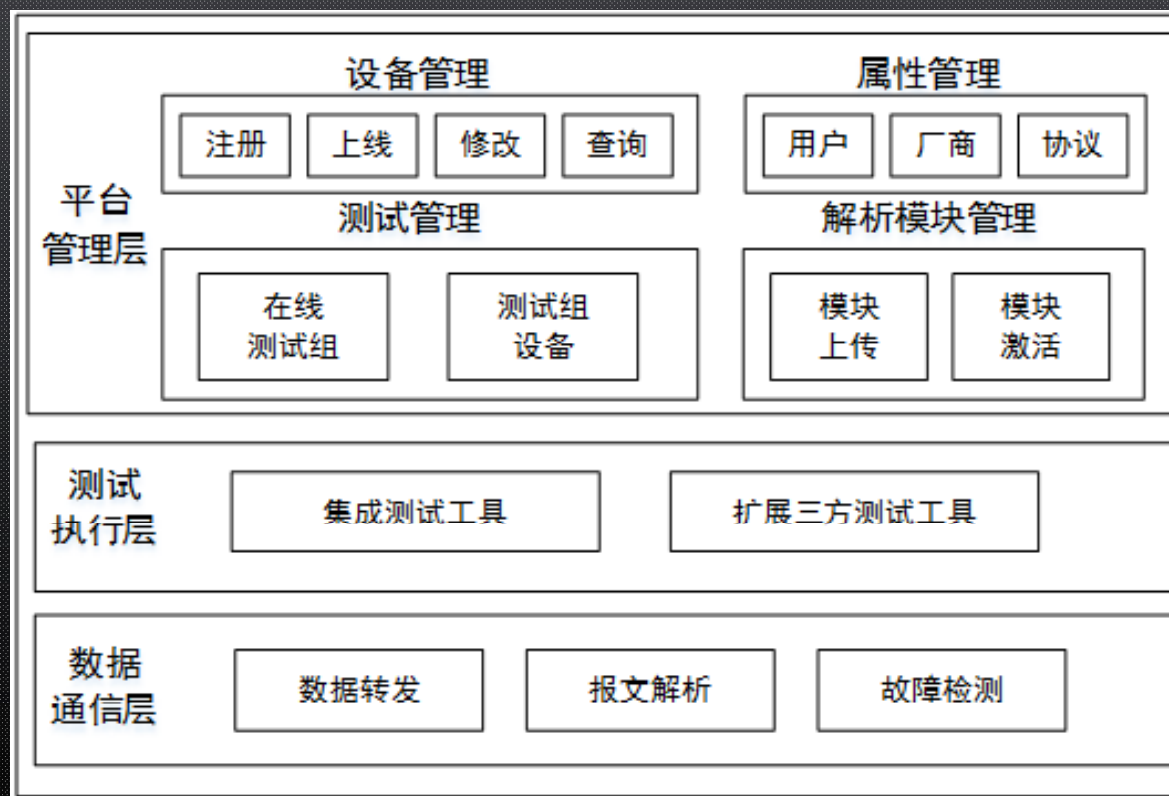
服务器能并发处理多组测试的进行，意味着服务器将以多线程提供服务。

服务器能实现数据的正确转发。

服务器能够存储所有的数据包，并对其进行正确的解析、分类、排序，以提供给客户端查询。



服务器端软件功能分层设计





客户端

客户端的功能应该包括但不限于以下几个方面：

- 客户端负责数据包的接收/发送,客户端相当于设备与服务器之间的中间件,维持两者的通信。
- 客户端能配置和注册相应设备,并将信息上传给服务器统一管理,以供其他设备使用。
- 客户端能查询所有可用的远程设备,并应该能预约和申请一到多个远程设备。
- 客户端能发起一次测试,当测试结束时,客户端能下载本次测试中整理、排序、解析过的数据包。





客户端软件功能分层设计





数据隔离及转发



当平台同时进行多组测试时（每组测试可能包括多台设备），服务器端必然涉及到多台设备的数据传输，数据报将汇总到服务器端并转发。如何区别数据包属于哪组测试的哪台设备，以及该数据包如何转发，是平台非常核心的一个模块。服务器端这一模块应该维护好测试组中的相关设备，并按照其关联关系正确转发数据包，同时存储数据包并整理出每组测试中数据包的源与目的地以及传输顺序。

通用性扩展

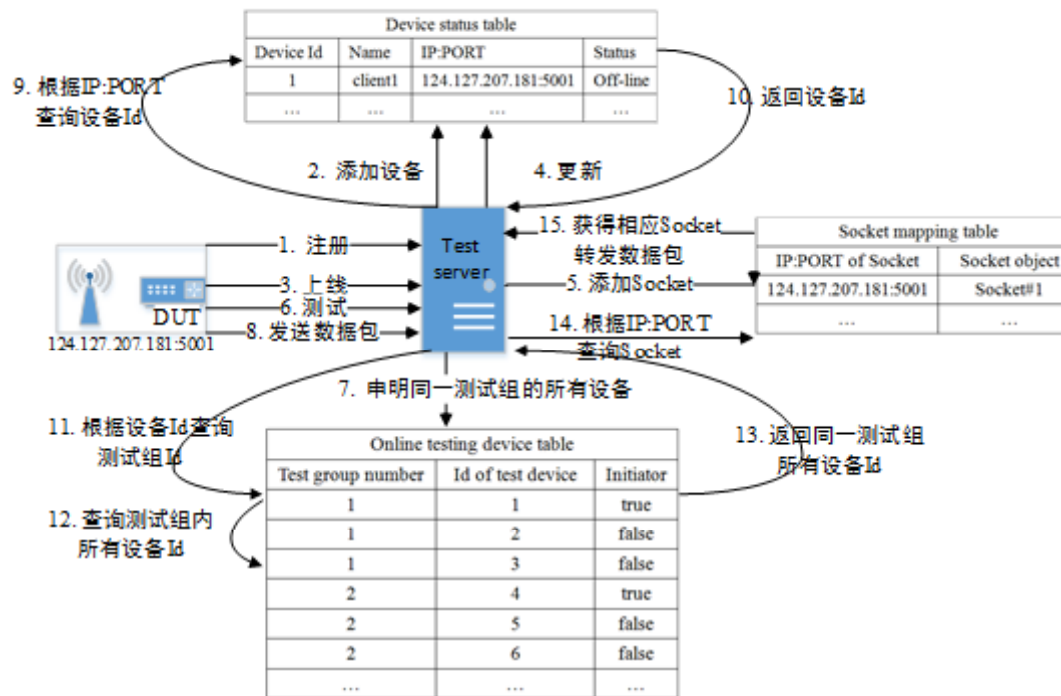


该平台面向的是不同厂商的设备和各种协议的测试。面对当前众多协议以及未来可能出现的协议，如何设计通用的数据解析模块是关系到平台通用性的重要方面，在平台的设计和实现时，要充分考虑其解析模块的可扩展性，最好面向接口编程，将解析模块设计成接口，针对某种协议有针对性的实现接口完成解析。保证当需要扩展某种协议的解析时，只需要开发插件，而不影响平台本身。

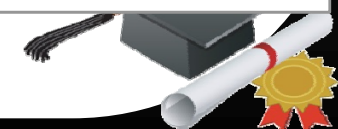
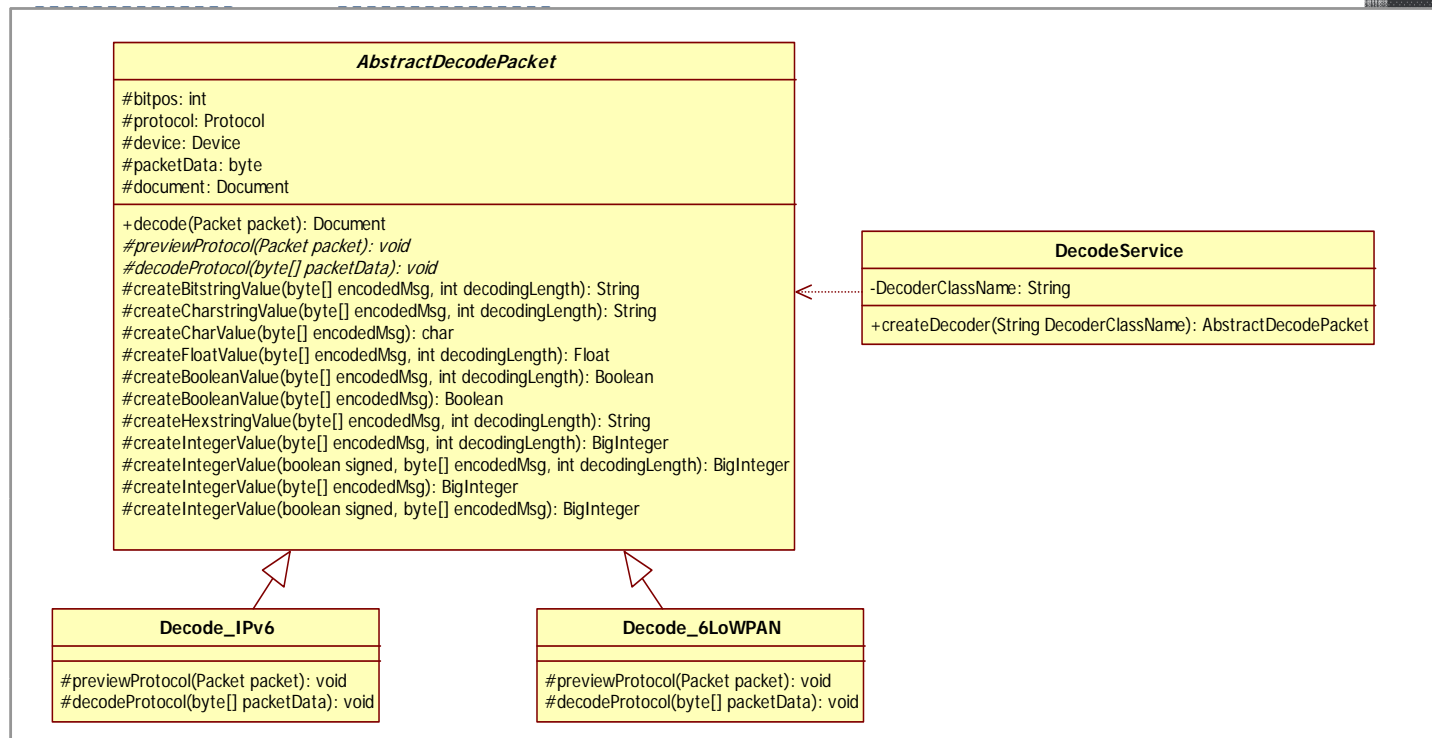




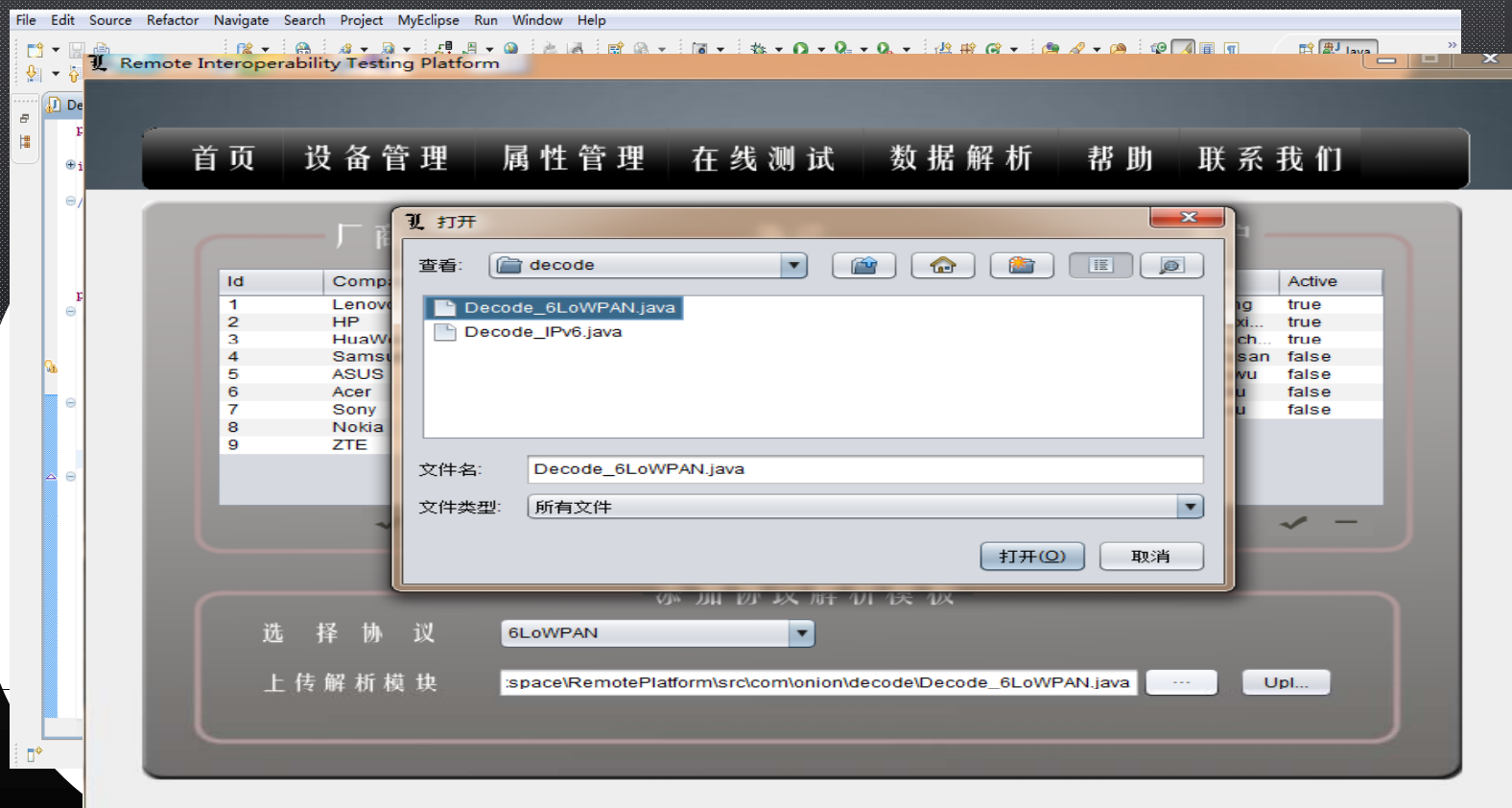
数据隔离及转发



通用性问题



6LoWPAN 协议解析插件



The screenshot displays the 'Remote Interoperability Testing Platform' web interface. At the top, a navigation bar includes links for '首页' (Home), '设备管理' (Device Management), '属性管理' (Attribute Management), '在线测试' (Online Test), '数据解析' (Data Parsing), '帮助' (Help), and '联系我们' (Contact Us). The main content area features a table of device manufacturers and a section for adding parsing plugins. A '打开' (Open) dialog box is open, showing the file 'Decode_6LoWPAN.java' selected in the 'decode' directory. The file name field contains 'Decode_6LoWPAN.java' and the file type is set to '所有文件' (All files). Below the dialog, the '选择协议' (Select Protocol) dropdown is set to '6LoWPAN', and the '上传解析模块' (Upload Parsing Module) field shows the file path: 'space\RemotePlatform\src\com\ionion\decode\Decode_6LoWPAN.java'.

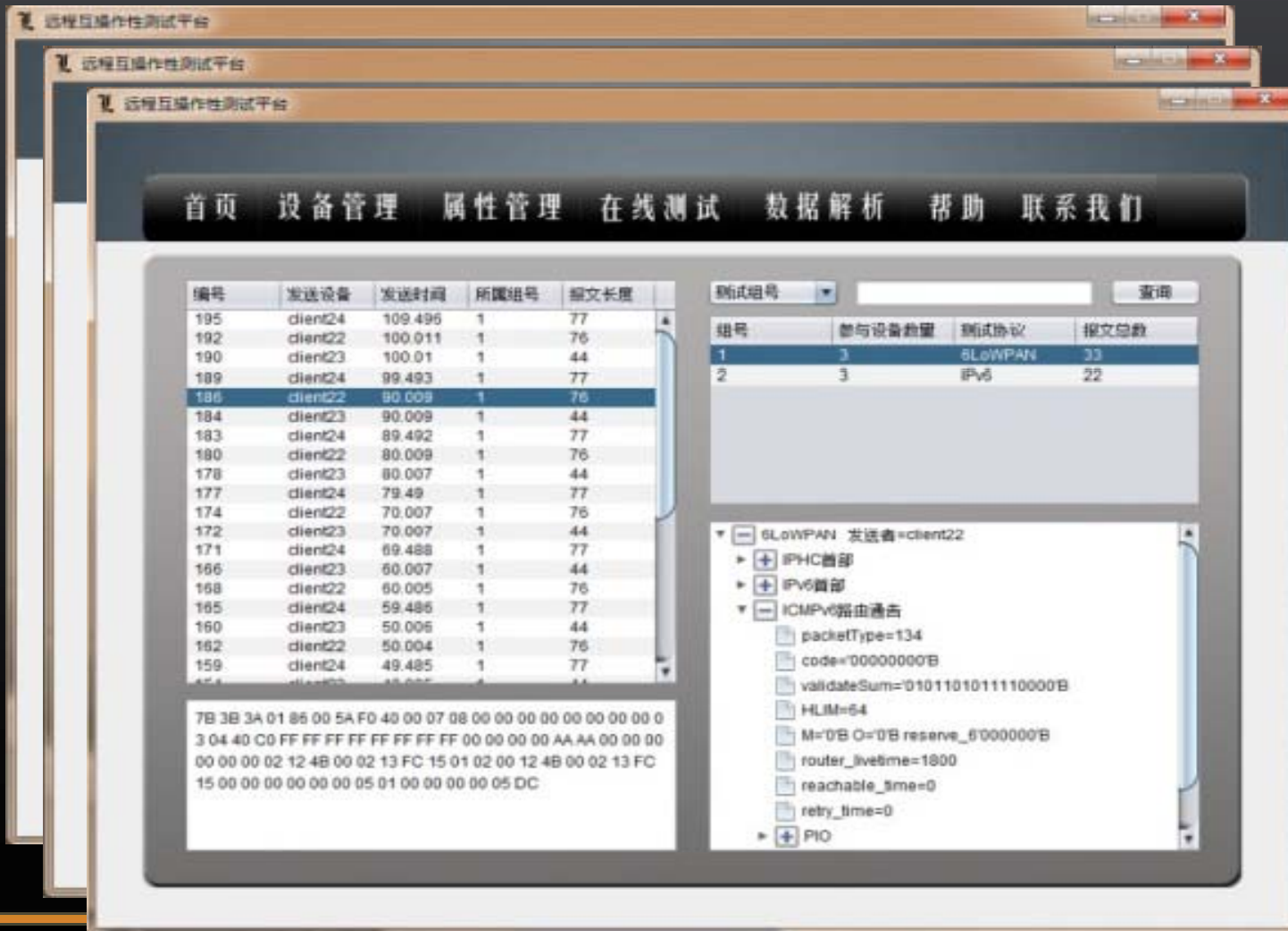
Id	Comp
1	Lenovo
2	HP
3	HuaWei
4	Samsung
5	ASUS
6	Acer
7	Sony
8	Nokia
9	ZTE

Active
true
true
true
false
false
false
false

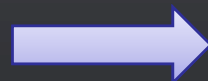
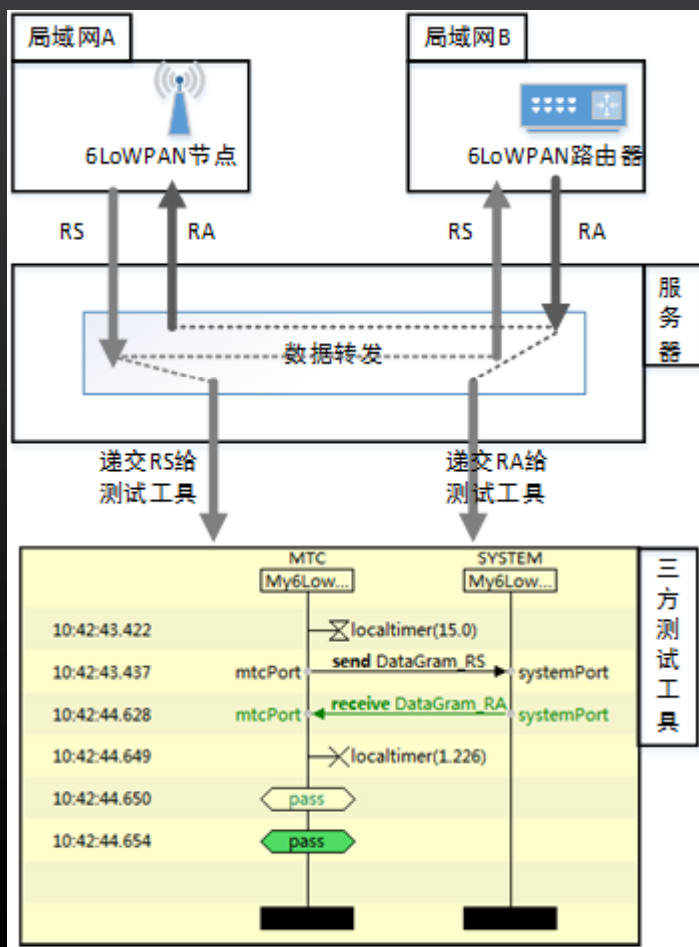
选择协议: 6LoWPAN

上传解析模块: space\RemotePlatform\src\com\ionion\decode\Decode_6LoWPAN.java

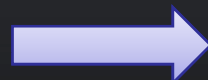
界面设计



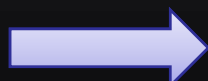
验证：远程测试6LoWPAN协议的路由发现功能点



远程客户端



服务器端



TTWorkBench
三方测试工具



本文创新性的提出了一种远程互操作性测试系统的架构，消除了传统协议测试必须局限在同一本地网络的局限性。文章针对远程测试系统的两个部分（服务器端和客户端）进行了分层设计与功能性设计。并且对于一些关键性的问题，如数据隔离、报文解析等进行了深入的研究并给出了解决方案。最后，我们实现了该远程测试平台的一个原型系统，并在该平台上对6LoWPAN协议进行了测试，验证了平台架构的正确性。

Thank You!

E-mail: lisc@buptnet.edu.cn

