FAST-ANT 2.0可重构网络测试与分析套件

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本号 | 修改人 | 修改时间 | 备注 |
| 2.0 | 蒋越 | 2019.05.23 | FAST-ANT 2.0初步设计 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

FAST-ANT（Agile Network Tester）作为一款基于FPGA/CPU的轻量级网络测试分析器，支持针对吞吐、延迟等相关指标的精确测量。其功能可定制、轻量级、低成本的特点，适合中小型研究机构进行网络性能与功能的测试与分析。

FAST-ANT 1.0将支持的基本功能有：

1. 测量设备对于不同大小、不同协议类型报文的精确/粗略时延；
2. 测量设备对于不同大小、不同协议类型报文的吞吐率；
3. 测量设备对于不同大小、不同协议类型报文的丢包率；
4. 数据统计功能。

但随着实际应用场景复杂化、多样化，FAST-ANT 1.0所提供的测试功能已经难以满足针对新的协议或功能的全面测试，如：并发的多条多类型流、流量回放、更高发包速率等。因此，基于上述现状，我们将设计并开发FAST-ANT 2.0来满足新的功能需求和性能指标。

FAST-ANT 2.0将基于FAST-ANT 1.0的基础上，优化重构软硬件设计实现，并在OpenBox-S4上先实现原型系统，后移植至NUDT-NSG团队自主开发的40G NP上。

# 1 引言

FAST-ANT 1.0采用FPGA/CPU的软硬件协同设计思想，充分发挥软件在灵活可定制和硬件在高性能精度上的优势，支持针对吞吐、延迟等指标的高精度测量。目前，FAST-ANT 1.0被广泛应用于测量网络设备（如：交换机）的转发速率和处理延迟，以及构造背景流量。FAST-ANT 1.0已经成功支撑不同设备、算法、协议的测试和验证工作，具有较高的应用价值和意义。

如图1所示，在FAST-ANT 1.0的硬件设计中，我们在FAST 2.0的五级流水线基础上，扩充两个功能模块PGM（报文生成模块）和SCM（统计收集模块）构成七级单流水线，提供了通用的高性能精度的报文发送和收集功能。



图1 FAST-ANT 1.0硬件设计

如图2所示，在软件设计中，在FAST API的基础上，向用户封装底层实现细节，支持用户通过配置相关测试参数，如：发送速率、发送端口等，来控制流量发送的模式；如：报文大小、内容等，来构造不同类型的流量；如：捕获报文类型，来收集统计特定类型报文的相关测试数据信息。

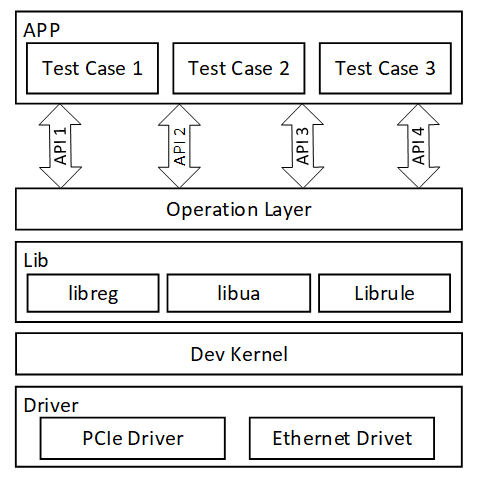


图2 FAST-ANT 1.0软件设计

在FAST-ANT 1.0中，由于采用的是单流水线的设计思想，因此仅支持发送单条特定类型的流量，并只能对指定类型的报文进行简单的数据统计。但随着测试场景的复杂化、待测协议和功能多样化，FAST-ANT 1.0已经难以满足新的测试需求。

首先，难以模拟复杂的测试场景，无法同时支持多条不同类型的流量并发存在。并且，流量发送模式单一，仅支持对发送速率的控制；

其次，对于流量的捕获统计只是简单地统计某一指定类型的报文个数，缺乏对捕获报文的保存来支持后续的流量重放；

最后，FAST-ANT 1.0仅支持1Gbps的报文发送速率，在性能方面有所限制，需要进一步提高整体性能。

因此，基于以上分析，我们将开发FAST-ANT 2.0，以更好地支持满足不同测试功能和测试指标。

# 2 测试功能

## 2.1 并发的多条多类型流量

FAST-ANT 2.0扩充的第一个功能是支持并发的多条多类型流量。FAST-ANT在测试中常常被用于构造测试环境的背景流量，例如：模拟流量爆发，产生一条高速率流量来测试网络转发设备的丢包率、最大吞吐或时延等，如图3所示。



图3 针对网络设备的单/双端口测试

但是在一些复杂的测试场景中，例如：TSN，需要产生复杂的背景流量，这样的背景流量中可能包含了不同优先级和不同类型的流。这些背景流量可以从单一的端口发出，也可以分别从不同的端口发出，如图4所示。



图4 复杂背景流测试场景

该功能的思路是：用户可以在Linux端通过构造不同的测试报文（报文类型、大小），配置报文发送的相关参数（速率）以及设定报文发送模式（乒乓报文、周期报文），同时指定不同类型的流的发送端口。在FPGA上根据Linux端用户设定来实现流量的精确、高性能发送。

## 2.2 流量回放

在FAST-ANT 1.0中，对于捕获到的报文，我们仅统计报文个数、吞吐量以及计算指定类型报文的延迟。当获取了以上统计数据后，捕获到的报文将被丢弃。但在某些测试场景中，例如：在相同测试场景中对比不同算法的效果或不同设备的性能，需要保存某一时间段内所捕获到的所有报文并再次使用；或是在测试结束后，对某一时段产生的流量进行进一步分析来获取网络的某一时段状态。

因此，在FAST-ANT 2.0中，我们将支持流量回放。该功能的思路是：用户需要在Linux端设定捕获报文类型、捕获时间段或捕获报文个数；在FPGA中根据设定的捕获参数，将捕获的报文存储在硬件上，允许用户在Linux端读取存储的报文生成可视化结果并用于后续的分析。

## 2.3 FAST-ANT 1.0测试功能

FAST-ANT 2.0将继续支持FAST-ANT 1.0中所提供的测试功能，包括：时延、吞吐量等相关测试功能和内容。

# 3 总体架构

FAST-ANT 2.0基于FAST平台设计开发，分别在软硬件层面借助了FAST所提供的数据结构、接口函数以及FPGA OS和UM中的部分代码（根据扩充功能去掉UM流水线中部分不需要的功能模块）。原型系统在OpenBox-S4上进行开发。

在分析上述功能后，我们发现不同功能之间的相关性较低，代码可复用部分较少，在同一硬件流水线上实现所有功能难度较大。因此，我们采用软件端统一化、硬件端可重构化的设计思想。

在软件端的设计开发中，基于FAST提供的接口函数，用户通过在图形界面上选取测试功能并配置相应的测试参数。配置信息通过接口函数配置到硬件FPGA上。底层的具体实现细节对用户屏蔽，用户只需要在图形化界面上进行操作即可获取预期的测试功能和结果。软件端在标准的Linux机器上开发，可移植到任意Linux机器上运行。

在硬件端的设计开发中，针对每一个功能生成对应的二进制烧录文件。根据软件端用户给出的具体需求，通过脚本文件自动烧入相应功能所对应的烧录文件，从而动态重构硬件部分。采用这样的设计方式可以充分发挥硬件性能，资源利用最大化。同时，不同功能之间不存在影响干扰，为将来扩充新的功能提供便利。

本章首先介绍FAST-ANT 2.0的软件架构设计。接着，根据不同的功能给出相对应的硬件设计。

## 3.1 FAST-ANT 2.0软件架构

FAST-ANT 2.0软件端用于接收用户在可视化界面上选定的测试功能和配置的测试参数，并在测试完成后以图标和数据文字相结合的方式展示相对应的测试结果。其整体结构如图5所示。



图5 FAST-ANT 2.0软件架构图

FAST-ANT 2.0的软件端部分基于FAST库所提供的API和数据结构，通过Linux NIC同硬件中LCM模块进行交互来完成测试模式和参数的配置工作。在继承了FAST-ANT 1.0的基础上，将软件端从ARM移植至标准Linux机器上，并增加可视化界面，简化测试仪的使用流程，提升用户体验。

首先，用户在提供的FAST-ANT GUI界面中选定测试功能并配置相对应的测试参数。GUI中的输入值将传递到FAST-ANT OS层，并映射到对应的FAST数据结构中；接着，将调用相应的FAST API和库，最终启动相应功能的脚本重构FPGA并将参数配置到对应的FPGA中的寄存器内；最后，FAST-ANT将同样通过调用相应的FAST API读取相关FPGA内寄存器的值，并在FAST-ANT OS层经过处理后以图标和数据文字的形式反馈至GUI界面，形成最终的测试结果的展示。

## 3.2 FAST-ANT 2.0硬件架构

FAST-ANT 2.0的硬件部分基于FAST流水线进行开发，但是同传统的FAST流水线有所不同。结合FAST-ANT 2.0所支持的新功能以及FAST-ANT 1.0已有的功能，我们去掉了原有五级流水线中的GKE和GME模块。同时，根据新的功能需求，不仅仅只实现单一流水线，并开发新的功能硬件功能模块。

考虑到在FAST-ANT 2.0中采用硬件可重构的设计思想，针对不同功能将开发插入不同功能模块的扩展流水线，并形成对应的硬件烧录文件。具体内容将在3.2.1和3.2.2节中进行介绍。

## 3.2.1 并发的多条多类型流量

支持并发的多条多类型流量功能的硬件设计图如图6所示。



图6 支持并发的多条多类型流量功能的硬件设计图

在FPGA中，我们实现并行的多条流水线，每一个流水线仅保留FAST-ANT 1.0中的PGM模块，从而支持多条流水线同时发送报文。软件端构造的报文和配置的发送参数将通过LCM模块分发到不同流水线中的PGM模块内，写入对应的RAM，修改相应的寄存器值。

在目前的FPGA OS中，已经实现指定映射端口。因此，我们可以指定不同流水线映射至不同的端口，并且实现一个DMUX来支持用户的自定义配置端口-流水线映射关系。这样，FAST-ANT 2.0既可以支持多个端口同时生成不同模式的不同类型的流量，每条流量之间互不干扰（每一条流水线对应于一个端口）；也可以支持生成一条混杂流（多条流水线映射至同一端口）。

该硬件架构的核心在于LCM中自定义控制数据报文格式和DMUX实现N到N的映射。

## 3.2.2 流量回放

在FAST-ANT 2.0中，为了支持流量回放功能，我们在SCM模块后插入UDC模块来实现，如图7所示。



图7 支持流量回放功能的硬件设计图

SCM对指定类型报文进行计数统计和延迟计算后，将所有报文传递到UDC模块。UDC模块根据用户配置的参数（报文类型、报文数量或捕获时长），将满足条件的报文存储到DDR 3，方便在测试结束后软件端获取相关报文。

SCM和UDC的参数配置均通过LCM模块。LCM模块将软件端下发的配置信息送至对应模块，修改相应寄存器值。在测试结束后，读取DDR 3，获得存储报文（这一过程对于读取速度要求不高）。

核心在于LCM中自定义控制数据报文格式以及UDC和DDR 3之间的通信。

# 4 FAST-ANT 2.0详细设计

# 5 进一步工作与开发计划

## 5.1 进一步工作

1. 在OpenBox-S4上实现FAST-ANT 2.0原型系统后，移植至40G NP并测试相关功能是否正常、相关指标是否满足；
2. 在40G NP上进行硬件功能模块的细化，充分发挥不同FPGA的优势，将高性能和精度相关的功能模块移植到Altera FPGA上，将发送控制模块移植到Xilinx FPGA上。

## 5.2 开发计划

1. 项目人员

孙志刚，全巍，徐东来，杨翔瑞，蒋越，吴尚明。

1. 项目分工
2. 设计文档指导：

孙志刚，全巍，杨翔瑞；

1. 项目开发指导：

徐东来，杨翔瑞；

1. 软件开发：

蒋越（软件配置硬件寄存器、FPGA动态重构脚本、读取存储在硬件上报文）

杨翔瑞（GUI）

1. 硬件开发：

蒋越（LCM模块设计开发、SCM和UDC模块设计开发、软硬件通信）

吴尚明（流水线并行、DMUX模块设计开发、测试仿真）

1. 进度安排

|  |  |  |
| --- | --- | --- |
| 日期 | 工作内容 | 参与人员 |
| 2019.05.27 | 针对设计文档2.0进行讨论修改。 | 孙志刚，全巍，徐东来，杨翔瑞，蒋越，吴尚明 |
| 2019.05.28  ~  2019.05.30 | 完成详细设计部分。  软件部分完成核心数据结构、函数设计；  硬件部分完成核心功能模块接口信号和状态机设计。 | 蒋越，吴尚明 |
| 2019.05.31 | 针对设计文档2.1进行讨论修改。 | 孙志刚，全巍，徐东来，杨翔瑞，蒋越，吴尚明 |
| 2019.06.01  ~  2019.06.02 | 完善软件和硬件详细设计部分。 | 蒋越，吴尚明 |
| 2019.06.03  ~  2019.06.16 | 开发硬件功能模块并进行仿真测试。 | 蒋越，吴尚明 |
| 2019.06.17  ~  2019.07.05 | 开发软件部分并测试；  上板联合调试软硬件代码。 | 蒋越，吴尚明，杨翔瑞，徐东来 |