

# 西安邮电大学本科毕业设计（论文）开题报告

学生姓名	刘康	学号	04182117	专业班级	网络 1804
指导教师	李刚	题目	基于 Python 的 TChannel 网络协议		

选题目的（课题背景及意义、国内外研究情况）

**TChannel** 是用于 **RPC** 的网络复用和成帧协议，它以极高的性能支持无序响应，实现高性能转发路径，中间件可以快速做出转发决策；具有乱序响应的请求/响应模型，以便慢速请求不会在行首阻塞后续更快的请求；它具有将大型请求/响应模型分解为片段并逐步发送的能力；可选检验和；可用于在端点之间传输多种协议，例如 **HTTP** + **JSON** 和 **Thrift**。

随着信息时代的到来，人们之间的联系变得越来越紧密，而更依赖各种网络系统和应用，因此，计算机网络通信和远程控制相关技术变得越来越重要。网络远程控制相关技术恰恰解决了很多现实生活中由于距离太远而处理不了的问题，从而给各个行业带来创新性发展，也变得更高效<sup>[1]</sup>。**TChannel** 是用于 **RPC** 的网络复用和成帧协议而远程过程调用（**RPC**）是一种从远程计算机程序上请求一个服务器，而不需要了解上层网络技术的协议。**RPC** 使用的是 **C/S** 模式。首先，调用过程发送一个调用信息到服务过程，然后等待应答信息。在服务器端，过程保持睡眠状态到调用信息的到达。当一个调用信息到达，服务器获得过程参数和计算结果，发送应答信息，然后等待下一个调用信息。最后，调用过程接收应答信息，获得过程结果，调用执行继续进行，其采用同步调用通信方式<sup>[2]</sup>。但是，**RPC** 技术在设计之初并没有将缓存作为考虑目标，且这种基于调用的通信方式也为系统提供缓存机制带来了困难，因此 **RPC** 并不支持缓存机制<sup>[3]</sup>。

**TChannel** 中运用 **Thrift** 处理程序注册和序列化。**Thrift** 是 **Facebook** 开发并开源的，偏重于跨语言调用框架的远程通信协议，它有一个轻量级、独立于语言的软件堆栈，具有相关的 **RPC** 代码生成机制。**Thrift** 为数据传输、数据序列化和应用程序级处理提供了清晰的抽象<sup>[4]</sup>。

正是由于 **TChannel** 协议的这些优点，可以实现业务的智能路由。本课题期望使用 **Python** 语言来实现 **TChannel** 网络协议，方便用户使用 **TChannel** 协议实现高业务量的智能路由。

参考文献：

[1] 王俊恒, 李树文. 计算机网络远程控制技术及应用研究[J]. 电脑编程技巧与维护, 2021(11):157-158+161. DOI:10.16184/j.cnki.comprg.2021.11.059.

[2] 李德贤, 陆歌皓, 姚绍文. JSON-RPC 协议分析、扩展及其应用[J]. 中国科技论文在

线, 2008(02):125-130.

[3]王姜, 余萍, 曹春, 马晓星. 开放网络环境下的程序设计:从RPC到REST[J]. 计算机工程与应用, 2013, 49(17):30-37.

[4]刘江霞. 一种计算机远程通信协议的研究[J]. 电脑编程技巧与维护, 2021(11):159-161. DOI:10.16184/j.cnki.comprg.2021.11.060.

[5]Vinoski S. Convenience over correctness[J]. IEEE Internet Computing, 2008, 12(4):89-92.

[6]Sirapat Boonkrong. A more secure and efficient Andrew Secure RPC protocol[J]. Security and Communication Networks, 2014, 7(11):

[7]Gilles Muller, Eugen-Nicolae Volanschi, Renaud Marlet. Scaling up partial evaluation for optimizing the Sun commercial RPC protocol[J]. ACM SIGPLAN Notices, 1997, 32(12):

[8]胡喜明, 胡淼. 基于响应式的RPC系统设计[J]. 计算机工程与设计, 2021, 42(09):2408-2416. DOI:10.16208/j.issn1000-7024.2021.09.002.

[9]刘鹤. 数字化样机平台RPC通信中间件的设计与实现[D]. 西安电子科技大学, 2020. DOI:10.27389/d.cnki.gxadu.2020.003156.

[10]Shuai li, Xiaojun Huang. Dubbo's Serialization Protocol Extension and Optimization of Its RPC Protocol Thrift[C]//Proceedings of 2017 IEEE 8th International Conference on Software Engineering and Service Science., 2017:741-746.

前期基础(已学课程、掌握的工具, 资料积累、软硬件条件等)

#### 1. 已学课程

- (1) 数据结构
- (2) 网络程序设计
- (3) 计算机网络
- (4) web 开发技术
- (5) python 语言程序设计

#### 2. 掌握的工具

Python, Html, Css, Javascript, Node.js

#### 3. 软硬件条件      软件: vscode, python3.7 等    硬件: windos10

要研究和解决的问题（做什么）

1. 使用 Python 语言实现 TChannel 网络协议
2. 支持 TChannel 消息流的创建；
3. 实现 TChannel 帧格式的解析；
4. 实现消息分片功能；
5. 实现检验和功能。

工作思路和方案（怎么做）

### 1.工作思路

（1）进行需求分析和设计，理清课题要完成的任务。

TChannel 旨在通过为客户端和服务端提供一种协议来解决这些问题，该协议具有连接两者的智能路由网格（称为 Hyperbahn）。

再次考虑那些 SOA 问题：

服务发现：所有生产者和消费者都向路由网格注册自己。消费者只能通过他们的名字访问生产者；无需了解主机或端口。

容错性：路由网格跟踪故障率和 SLA 违规等内容。它可以智能地检测不健康的主机并将它们从可用主机池中删除。

请求跟踪：作为一等公民内置于协议中。

整合生产者和消费者之间的逻辑还允许将核心功能推送到整个 SOA，而无需应用程序更新库。

此外，在开发此协议和路由网格时，牢记以下目标：

该协议必须易于以多种语言实现，尤其是 Javascript、Python 和 Go。不想局限于一种语言。

异步行为是基本的。遵循传统的请求/响应模型，响应乱序；慢速请求不得阻塞行首的后续较快请求。

大的请求/响应可能/必须分解成片段以逐步发送（即支持“流式”请求）。

TChannel 可以与任意序列化方案一起使用，例如 JSON 和 Thrift，尽管 Thrift 因其 IDL、类型安全和验证保证而被首选。

路由网格需要高性能的转发路径。中介机构必须能够迅速做出转发决定。

请求/响应完整性的可选校验和。

（2）配置课题所需环境

（3）对涉及的技术所需要的环境以及编写代码所需要的技术进行系统的学习，同时回顾已学习过的内容

（4）开始代码的书写以及调试

（5）测试代码是否达到课题预期效果

### 2.技术方案

（1）创建一个名为 keyvalue 在其中工作的目录,在这个目录中，还将创建一个 keyvalue 模块，它需要根目录下的 aninit.py 和 a setup.py

（2）TChannel 有一些自定义行为，因此它不能使用 Apache Thrift 代码生成器生成的代码。相反，动态生成 Thrift 类型。

（3）要为应用程序提供服务，需要实例化一个 TChannel 实例，针对该实例注册处理程序

### 3.进度计划

（1）2021 年 11 月 20 日-2021 年 12 月 31 日 学习相关知识，掌握基本原理，撰写开题报告提交系统；

- (2) 2022 年 01 月 01 日-2022 年 01 月 20 日 熟悉所要开发的系统并进行系统分析;
- (3) 2022 年 01 月 21 日-2022 年 02 月 10 日 进行概要设计, 详细设计;
- (4) 2022 年 02 月 11 日-2022 年 04 月 14 日 根据设计进行环境搭建, 软件编码并实现, 进行中期检查;
- (5) 2022 年 04 月 15 日-2022 年 05 月 10 日 测试程序, 分析结果, 进行迭代, 进行第一次验收;
- (6) 2022 年 05 月 11 日-2022 年 06 月 12 日 终期验收, 论文撰写、答辩。

指导教师意见

签字:                      年    月    日