

# 4over6 隧道协议实验报告

马 也 2013011365 计 34

沈哲言 2013011371 计 34

May 19, 2016

## 1 实验目的

- 掌握 Android 下应用程序开发环境的搭建和使用
- 掌握 IPv4 over IPv6 隧道的工作原理

## 2 实验原理

### 2.1 4over6 隧道模式原理

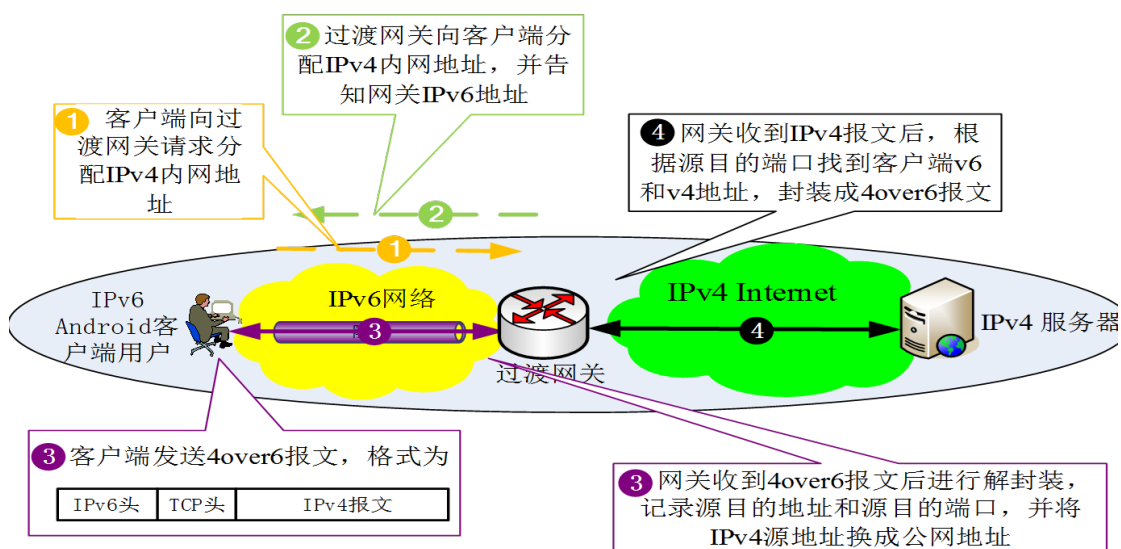


Figure 2.1: 4over6 隧道模式原理图

IPv4 over IPv6 隧道模式是指，将 IPv4 的报文装载到 IPv6 网络的数据段，通过隧道模式进行传递，这样就可以使用 IPv6 的网络来访问 IPv4 的资源。主要流程如下：

安卓客户端向过渡网关请求分配 IPv4 内网地址；过渡网关分配 IPv4 内网地址，并提供对应的 IPv6 网络地址；接着，安卓客户端发送 4over6 报文，过渡网关接受报文并进行分析，得到源地址和目的地址，将 IPv4 地址转化为公网地址，发送到公网之中；当过渡网关收到公网的 IPv4 报文之后，根据记录好的映射关系，重新封装成 4over6 报文，发给对应的内网用户，完成数据的转发和接受。

在以上流程中，用户处于 IPv6 的网络环境之中，过渡网关横跨 IPv4 和 IPv6，并在其中提供 IPv4 和 IPv6 地址的转换和分发。在本实验中，过渡网关部分的代码已经完成，我们需要完成用户端的代码，以实现 IPv4 over IPv6 隧道协议。

## 2.2 安卓 VPN Service 原理

在本实验中，我们采用了 VPN Service 作为基本框架来实现 IPv4 over IPv6 隧道协议。VPN Service 是安卓提供的一套 API 接口，方便编程人员创建 VPN 服务。当打开该服务之后，安卓系统将所有应用程序发送的 IP 包都根据 iptables，使用 NAT 转发到 TUN 虚拟网络设备上，其端口为 tun0。当打开 VPN Service 之后，我们可以获取 tun0 的文件描述符，这样就可以读取或者写入数据以实现发送或者接收数据。

也就是说，在打开 VPN Service 之后，我们可以监控系统所有的网络进程，并从 tun 中获得系统中所有 IP 包。这样，我们就可以将 IP 包封装在 IPv6 报文之中，使用隧道协议将报文发送到服务器，实现数据的发送；当我们收到服务器反馈的 IPv6 报文后，我们将其进行拆解，得到真正的 IPv4 包，再写入到 tun 文件之中，以实现数据的接收。

## 2.3 JNI 与 NDK

在本次实验中，由于经常要和以字节为单位的数据打交道，并且需要精细地管理内存，因此我们采用 C 来实现 IPv4 over IPv6 的核心功能。要在以 Java 为语言的安卓环境中使用 C 来进行编程，我们需要使用 JNI 和 NDK。

Java Native Interface (JNI) 规定了一套 Java 的原生接口，规定了 Java 和其他语言交互或者是在其他平台下运行时的接口，让我们可以使用其他语言（如 C、C++、汇编等）访问 Java 中的类和对象。而 Native Development Kit (NDK) 是 Google 提供的一套方便开发者的开发套装 (SDK)，方便编程人员将 C 或者 C++ 程序打包并加

入到 APK 之中，供安卓程序使用。

## 3 实验设计与内容

### 3.1 程序整体设计流程

本次实验的编码主要分为两个部分，Java 实现的安卓客户端（前端）和 C 实现的收发进程（后端），前端和后端之间通过管道（named pipe）来进行连接并传递数据。实验整体流程如下：

```
// TODO
```

### 3.2 前端工作流程

### 3.3 后端工作流程

### 3.4 前端与后端的交互和通讯

## 4 实验结果与分析

## 5 实验中遇到的问题

## 6 实验心得与体会

## 7 对实验的意见和建议