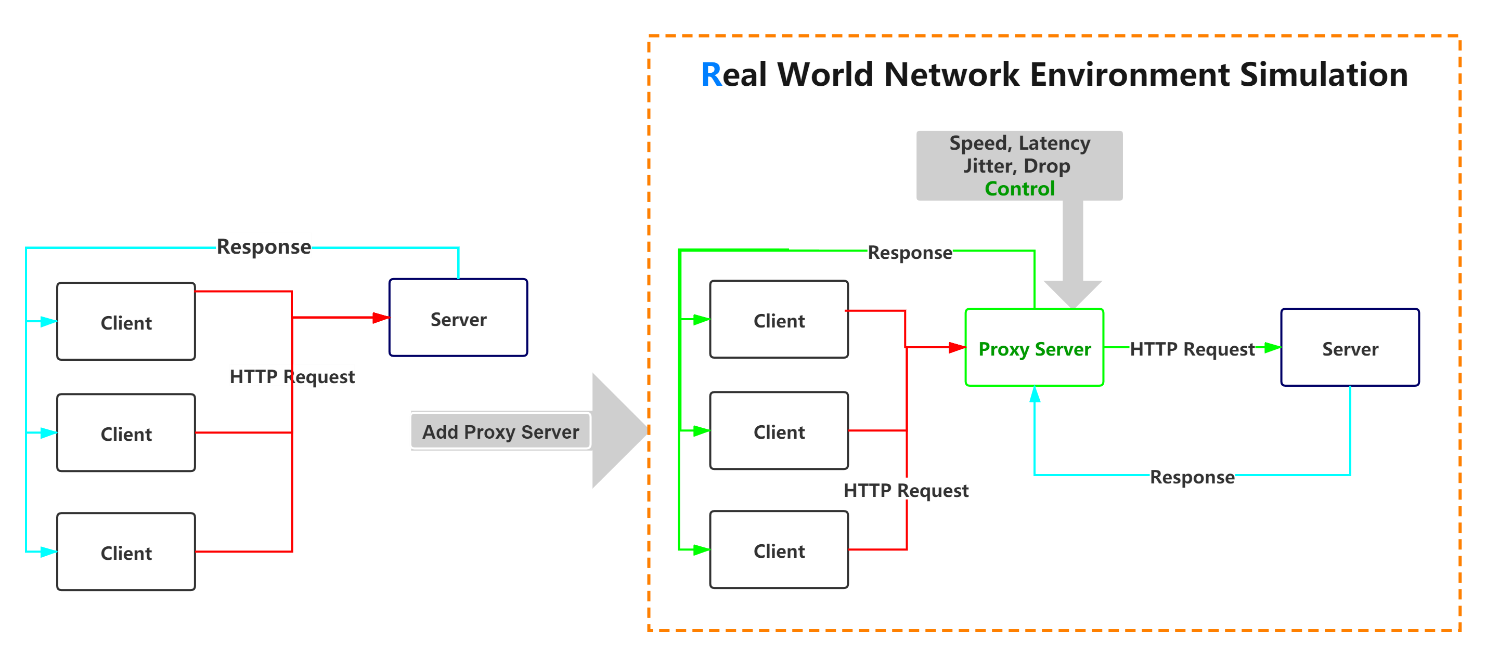
**Introduction:**

一个典型的Web应用与用户间的交互可以分为四个部分：一、用户（User）通过浏览器发送一个HTTP请求；二、服务器（Server）收到请求，生成一个HTML文档；三、服务器（Server）把HTML文档作为HTTP响应的Body发送给浏览器；四、浏览器收到HTTP响应，从HTTP Body取出HTML文档并显示给用户。

所以，简单的Web应用就是先把HTML用文件保存好，用一个HTTP服务器接收用户请求，从本地读取文件，生成HTML文档并返回给客户端。这就是Apache、Nginx、Lighttpd等一些常见的静态服务器的实现原理，本项目的服务器属于此类静态服务器。在现实中，静态服务器传输文档时常会遇到网络拥塞，服务器传输速率将会受到限制。并且，现实不稳定的网络环境将时不时产生网络抖动Jitter，导致传输速率跳变，甚至出现断网。为了证实我们提出的新型ABR算法可以适应现有的网络环境，我们需要模拟多种不同的网络状况，以确定新型ABR算法能针对不同的环境进行较好自适应调节，使得QOE指标维持在一个相对较高的水平。

**总体思路：**

在Server端进行模拟需要改变服务器内置代码才能达到目的，需要对服务器的框架和主体代码做出较大改动。而我们通过增设一个代理服务器，使得原先客户端向服务器直接的请求变成向代理服务器发送请求，通过代理服务器再向原服务器发送原请求。代理服务器（Proxy Server）内置响应延迟时长（Response Latency）、传输速率上限（Transmission Speed）、网络抖动概率（Jitter）、断网概率（Disconnetion）等参数，以模拟各种典型的网络状况。代理服务器接受到原服务器的HTTP相应之后，再向客户端发送响应报文，从而在客户端达到模拟各种真实网络状况的目的。



图片一： 模拟各种现实网络环境的总体思路

**Features:**

**使用Tornado Web Application开发框架**

搭建Web 应用程序常用的有三大框架：Django、 Flask 以及 Tornado。三个框架各有特点，适合于不同场景的Web 应用程序开发。其中Django是Python 界最全能的 web 开发框架，各种Web功能完备，架构稳定，可维护性强。Django完备的功能和普适性的框架适合于快速开发大型的项目。Django 的缺点就是处理速度慢，吞吐量小，其原因主要出在 Django 与数据库的交互上（这点可以通过 Celery 等工具解决）。Flask适合于开发轻量级的Web 应用程序，是微框架的典范。Flask拥有简洁的接口和高度的灵活性，小型项目开发者可以使用Flask自由开发Web框架和主体内容。然而这个灵活性也是一把双刃剑，经验不足的开发人员使用Flask框架容易写出低质量的代码，是得Web应用程序稳定性较差、难以维护。

Django 和 Flask 均是同步框架，在处理并发性请求时性能受限，不能很好的适应高吞吐量的应用场景。而Tornado天生自带异步、非阻塞特性，在IO密集型应用和多任务处理上占据着绝对性的优势。Tornado良好的异步性能使得其搭建的Web Application在面对高并发的请求时能保证正常的吞吐量，适合于本项目的应用场景。

协程（Coroutine）与异步框架

在实际服务器和客户端交互的过程中，

**双模式**

* Simple mode 是默认模式。可以使用 python config.py mode simple 设置。进入 simple mode 后，用户可以设置：

1. Latency (ms) : HTTP 链接首字节传输的延迟；

2.speed (bit/s) ：所有通过代理服务器的HTTP 链接总速率限制。

* Advanced Mode 是高级模式，在此模式下网络模拟器会引入随机的网络延迟、传输速率限制和抖动、断网等网络情况，可通过python config.py mode advanced设置。进入Advanced Mode后，用户可以设置以下模拟器参数：

1.latency-(min|max): HTTP 链接首字节传输延迟的上限和下限，默认情况下网络模拟器设置为latency-min = latency-max = 1000ms；

2. speed-(min|max): 所有通过 chaos-proxy 的 HTTP 链接总速率限制上限和下限。通过设置速率的抖动上下限，可以模拟网络的不同稳定情况；

3. jitter-prob：网络抖动发生的概率（0-1 之间）；jitter-(min|max)：网络抖动的时间长度（以毫秒表示）。通过设置jitter参数，模拟器可以模拟网络抖动的情况和发生抖动的概率。

4.reset-enable：是否启用断网模式(默认: false 启用: true)；reset-prob：在用户启用了断网模式后，断网发生的概率。网络模拟器通过主动断开与客户端的HTTP连接，来模拟服务器宕机的情况。

用户可通过组合多种参数的不同取值，来模拟复杂的网络环境（如传输速率跳变、响应延迟时长跳变、随机网络抖动等）。在测试过程中，我们先是使用simple mode 对网络的基本环境进行了模拟。我们分别设置延迟时长（Response Latency）、传输速率上限（Transmission Speed）为(300ms，5000bit/s), (500ms，3500bit/s), (1000ms，2000bit/s), (2000ms，800bit/s) ，在服务器对应模拟网络状态下测试多个客户端并发请求不同字节客户端平均接收速率VClient（详细实验测试结果见附件）。