

**本 科 毕 业 设 计（论文）**



**题目: 无线路由器延迟测量与优化**

**姓 名 孟媛**

**学 院 信息与通信工程学院**

**专 业 电子信息工程**

**班 级 2012211102**

**学 号 2012212724**

**班内序号 31**

**指导教师 裴丹**

**2016年 5 月**

**无线路由器延迟测量与优化**

**摘 要**

当前无线网络已经日渐成为我们生活中不可缺少的一部分，同时无线网络延迟对于手机应用的用户体验有着至关重要的作用。然而，对于无线网络延迟目前大部分仍处于未知状态。在这篇论文中，我们首先提出一种有效的方法来对网络往返延迟（即RTT）进行分解。然后基于我们在国内某高校部署的47台无线路由器在两个月内采集的无线延迟及无线参数信息，我们将提出一种对于无线延迟的系统研究方法。我们发现，无线延迟可以成为网络往返延迟最薄弱的环节：有50%的TCP包的无线延迟超过20ms，有10%的TCP包的无线延迟超过100ms。并且对于超过一半的网络延迟来说，无线延迟的占比将会超过60%。为了更好的理解无线延迟，并对其进行故障定位，优化分析，我们训练了一种决策树的模型。基于此模型，我们对一实际场景进行了优化，并将无线延迟的中值从50ms减少到10ms，即降低80%。同时将另一实际场景的最大延迟，从250ms降低到50ms。同时，我们利用我们测量的延迟作为分布式的针对独立的无线路由器的信道调整的参数，最终我们可以将无线延迟的中位数降低5倍。

**关键词** 无线延迟 决策树 故障定位 优化分析

**Measurement and Optimization of the WiFi Latency**

**ABSTRACT**

As mobile Internet is now indispensable in our daily lives, WiFi’s latency performance has become critical to mobile applications’ quality of experience. Unfortunately, WiFi hop latency in the wild remains largely unknown. In this paper, we first propose an effective approach to break down the round trip network latency. Then we provide the first systematic study on WiFi hop latency in the wild based on the latency and WiFi factors collected from 47 APs on T university campus for two months. We observe that WiFi hop can be the weakest link in the round trip network latency: more than 50% (10%) of TCP packets suffer from WiFi hop latency larger than 20ms (100ms), and WiFi hop latency occupies more than 60% in more than half of the round trip network latency. To help understand, troubleshoot, and optimize WiFi hop latency for WiFi APs in general, we train a decision tree model. Based on the model’s output, we are able to reduce the median latency by 80% from 50ms to 10ms in one real case, and reduce the maximum latency from 250ms to 50ms in another real case. Using the latency we measured as input, distributed channel selection on individual Aps reduces the median WiFi hop delay by up to 5×.

**KEY WORDS** wifi latency decision tree troubleshoot optimazation

目 录

第一章 无线延迟研究系统介绍 1

1.1 研究背景 1

1.2 研究方法介绍 1

第二章 网络延迟分解 2

2.1 往返网络延迟中的时间戳 2

2.2 数据采集平台及算法 4

2.2.1 数据采集平台 4

2.2.2 数据采集介绍 4

2.2.3 抓包 4

2.3 分解实际场景中TCP三次握手过程的网络延迟 4

2.3.1 有线延迟 5

2.3.2 下行无线延迟 5

2.3.3 上行无线延迟 5

2.4 分解实际场景中TCP数据包的网络延迟 6

第三章 网络延迟测量结果 6

3.1 实际场景中的无线延迟 6

3.2 往返网络延迟各部分的累积分布 7

第四章 无线参数分析 8

4.1 无线参数统计 8

4.2 无线参数的累积分布函数 9

4.3 无线参数及无线延迟相关性的初步展示 9

4.4 量化的相关性 10

4.4.1 相关系数 10

4.4.2 相对信息增益 11

4.5 无线参数分析总结 11

第五章 无线延迟模型 11

5.1 问题陈述 12

5.2 模型训练 12

5.3 决策树模型介绍 12

5.3.1 决策树的优点 13

5.3.2 决策树的缺点 13

5.3.3 数据预处理 13

5.4 评估 14

第六章 基于预测模型的优化方案 15

6.1 优化的三个步骤 15

6.2 调整无线路由器的位置 16

6.3 信道选择 17

第七章 优化的信道选择算法 18

7.1 最小拥塞信道选择算法 18

7.2 实验场景介绍 18

7.3 实验结果及分析 19

7.4 关于信道调整的补充说明 20

第八章 相关工作 20

第九章 总结 21

参考文献 22

致 谢 24

附 录 25

# 

# 第一章 无线延迟研究系统介绍

## 1.1 研究背景

当前无线网络已经日渐成为我们生活中不可缺少的一部分。并且无线网络延迟对于手机应用的用户体验有着至关重要的作用，原因如下：首先无线路由器已经成为用户接入互联网的主要方式。在2013年，对于55%的网络流量，WIFI成为他们的最后一跳[1]。其次，用户要求更短的移动应用响应时间，如果页面载入时间超过3秒钟，就会有40%的用户放弃此页面的浏览[2]。它同时也表明有47%的用户希望页面加载时间小于2秒钟。另外，当前大部分移动应用所采用的HTTP协议是对于网络延迟十分敏感的，网络延迟每10毫秒的增加，就会造成页面载入时间1000毫秒的升高[3]。并且，无线干扰一旦出现，就会造成无线延迟的激增。因此，综合以上数据[2][3]，为了满足用户需求，无线延迟应控制在20毫秒至30毫秒以下。

尽管无线延迟的影响极大，目前对于无线延迟的研究大部分，仍然处于位置状态，这个可能是由于有效测量方法的缺失。在这篇论文中，我们将提出一个系统的研究方法，来回答以下三个有关于现实场景中无线延迟及无线参数的重要问题：

现实场景中的无线延迟是什么样子的？

哪些无线参数对于无线延迟的影响最大？

我们应当怎么优化现实场景中的无线延迟？

## 1.2 研究方法介绍

首先我们提出一种叫做wing的方法，它能够有效的测量分解TCP包的网络往返延迟（值得注意的是，这里我们提到的网络往返延迟将忽略服务器处理时间，具体细节将在下一章介绍）。这里我们将其分解为上行无线延迟，有线延迟，及下行无线延迟。我们的方法能有精确的测量所有TCP包的下行无线延迟，并且用精确测量的TCP三次握手包的上行无线延迟和有线延迟来近似的估计其他TCP包的上行无线延迟和有线延迟。此方法不需要用户端的任何协助，并且能够部署在任何一台可装载OpenWrt系统的无线路由器中，而不需要无线路由器芯片厂商的支持。

其次，本课题展现了在实际场景下对于无线延迟的研究。我们将47台配置了OpenWrt系统及wing方法的无线路由器部署于国内某高校学生宿舍中，作为他们的主要联网接入点。我们在2016年2月份至4月份两个月内，采集了无线延迟及无线参数（如信号接入强度，重传比率，信道占用率等）共约2T的原始数据。我们观察到对于这些无线路由器来说，有超过50%TCP包的无线延迟超过20毫秒，有超过10%的TCP包的无线延迟超过100毫秒。令人惊讶的是，超过一半的流量中，无线延迟都占到往返总延迟的60%。无线路由器，作为当前大部分网络的最后一跳，在大部分情况下，已经成为了网络延迟中最薄弱的环节。

第三，利用之前采集到的大量的无线参数数据，我们训练了一个普试的决策树模型，这个决策树模型能够对部署了本课题中方法的无线路由器及其他路由器提出优化建议。1）这个直观的决策树模型能够表明哪种无线参数对于无线延迟的影响最大。例如我们观察到，当信道占用率超过55%的时候，就会有超过67%的可能性出现无法忍受的无线延迟。2）在一个实际的例子中，我们利用我们普试的决策树模型，根据无线参数来对TCP包进行分类。分类的结果表明造成高无线延迟的主要原因是高信道占用率。当我们把无线信道调整到信道占用率低了8%的信道上后，此无线路由器的最高无线延迟从250毫秒降低到50毫秒。3）在另外一个实际的例子中，分类的结果说明，客户端的接收信号强度是高延迟的问题所在。重新放置无线路由器的位置，能够将无线延迟的中位数从50毫秒降低到10毫秒，降低了80%。

第四，利用我们在前面提到的延迟测量方法，我们对于当前最为常用的最小拥塞信道选择（Least Congestion Channel Search）算法进行了优化，我们直接将我们测量的延迟结果作为最小拥塞信道选择算法的输入参数，而不是其常用的信道利用率。最终在我们针对每个独立无线路由器选择算法中，我们将无线路由器的最后一跳的延迟降低了5倍。并且考虑到切换信达可能对于用户造成的影响，我们采用了802.11n中提出的信道调整方法，能够在对于用户影响最小的情况下，调整信道。

# 第二章 网络延迟分解

在这一章，我们首先将利用TCP包的生存周期来系统的介绍往返网络延迟（忽略服务器处理时延）。其次，我们将介绍我们的数据收集平台和算法。最后我们将描述我们利用TCP三次握手包及TCP数据包，将往返网络延迟分解为三个不同部分的方法。

## 2.1 往返网络延迟中的时间戳

为了研究往返网络延迟（忽略服务器处理时延RTNL=RTT-S），在本课题中将研究一个由客户端产生的包p，及在一个往返通信中，一个由服务端产生的响应包q的生存周期，并最终抽离出延迟部分。本课题重点研究TCP包，因为TCP包是当前主要的网络传输协议[4][5]。并且对于网络延迟十分敏感的，最常用的移动应用网络协议——HTTP协议，也是基于TCP协议的[3]。现在我们来看一下TCP三次握手包的生存周期，来系统的分析整个过程中每部分产生的网络延迟。



图 1-1 TCP三次握手包往返延迟示例

图1-1中展示了TCP三次握手过程中，包在往返通信过程中的生存状态，以及它们在这个过程中经历的网络延迟。在这个例子中，p是TCP的SYN包，q是TCP的SYN-ACK包。

在最开始，在时间时刻，包p由用户设备（UE）生成。经过了发送排队等待时间，信道避让时间，发送时间以及可能的重传时间后，包p最后在时间到达无线路由器的无线网卡。这段从到的时间，这里我们定义为无线上行延迟（UL）。

包p一到达路由器的无线网卡（wireless NIC），它就会在无线路由器内部被处理，然后被交付到无线路由器的有线网卡（wired NIC）。我们这里用来标记包p到达无线路由器有线网卡的时间。那么-就是无线路由器内部处理延迟（PL）。在这之后，无线路由器将等到在有线网卡收到响应包q。这段过程中的时间--S我们标记为有线处理延迟（WL）。S这里是指服务器端的处理时延。有线延迟主要包含了数据包从无线路由器到服务器端的有线网络延迟。

等到包q到达无线路由器的有线网卡，它就会经过无线路由器内部处理然后被交付到无线路由器的有线网卡。这里我们用来标记包q到达无线路由器无线网卡，并开始排队等待被发出的时间。类似的这里的-就是无线路由器的内部处理延迟（PL）。当包q到达无线路由器的无线网卡后，它一样会经历一段排队等待时间，信道避让时间，发送时间及重传时间。在这之后，无线路由器才会成功收到由用户发过来的MAC层的ACK包。这里我们用来标记包q到达用户端的时间，并且用来标记无线路由器收到的由用户发来的q的MAC层ACK的包的时间。类似上行无线延迟，从到的这段时间我们标记为无线下行延迟（DL）。到此，包p，包q已经完成了它们的一次完整的往返通信过程。从到的这段时间我们定义为往返网络延迟（RTT），这里我们研究忽略服务器处理时延后即为

RTNL=RTT-S。

## 2.2 数据采集平台及算法

### 2.2.1 数据采集平台

我们的数据是从实际部署与国内某高校的路由器中收集得到的，这个高校有42000个学生及11000位教职工。我们共免费发放了47台无线路由器给此学校的学生，并让他们使用此路由器作为他们无线上网的主要方式。我们所有部署的无线路由器都是NETGEAR N750 WiFi Dual Band Gigabit Router Premium Edition (WNDR4300)。此路由器拥有560MHz AMD CPU，128MB 内存, 128MB 闪存, 4个1Gbps 局域网口, 一个1Gbps广域网网口以及两个Atheros无线接口(一个支持 b/g/n and 一个支持 a/b/g/n).

在部署的47台无线路由器中，有44台部署于学生宿舍，剩余的部署于研究生实验室。在不同的学生宿舍楼里，每层有20——40间宿舍，每间宿舍的大小为20平方米。

### 2.2.2 数据采集介绍

我们在每一台装载了OpenWrt ( Barrier Breaker [6] version)系统的无线路由器上，植入了我们的轻量级测量程序。本课题中所采用的测量方法可以直接被部署于任何一个支持OpenWrt系统的无线路由器中，而不需要来自设备厂商的任何支持。我们的程序将持续的采集一下数据，1）由实际流量中提取的数据包的相关信息，2）可以反映无线环境情况的无线参数信息。

### 2.2.3 抓包

我们用了基本的libpcap[7]库来抓取实际通过路由器的数据包。在本课题中，我们将TCP包作为我们主要的研究对象，因为它是大部分网络应用所依赖的网络协议[4][5]。不像UDP，TCP协议的可靠传输（如对于SYN包的立即的ACK包）能够帮助我们更好，更准确的在整个往返网络时间中抽离出网络延迟，而避免和上层应用延迟重叠。处于隐私及负载的考虑，我们只抽离每个TCP包的以下域：IP地址，序列号，确认号。所有的IP地址都是匿名的，而不能被追溯到某个特定的用户。序列号和确认号被用来重建整个TCP的过程。我们也在无线路由器系统内记录了每个TCP包到达或者离开某个网卡的时间，为了进一步提取网络延迟。在整个两个月的测量中，我们在活跃的47台无线路由器中共采集了2T的数据。我们利用rsync同步工具[8]，每两分钟将原始数据压缩同步到服务器中。

## 2.3 分解实际场景中TCP三次握手过程的网络延迟

在展示了对于TCP三次握手过程的剖析后，我们将往返延迟分解成为几个独立的部分——上行无线延迟，下行无线延迟，有线延迟。然而，现实中的世界挑战是我们怎么来测量这些延迟，因为目前我们能够直接测量的只有图1-1中用椭圆标注的时间戳。下面我们将给出说明:

### 2.3.1 有线延迟

有线延迟（WL）= --S。有线延迟利用时间戳，可以直接计算得到，这两个时间戳都可以直接在包经过无线路由器有线网卡的时候被测量到。另外对于三次握手包来说，服务器处理时延可以忽略不计，因为对于TCP的SYN包来说，服务器的处理时间是极短的。因此我们用TCP三次握手包的-来近似估计有线延迟。

### 2.3.2 下行无线延迟

下行无线延迟（DL）= -。这里本课题中第一个难点在于如何测量及。可以通过无线路由器被直接测量。一旦MAC层的q的确认包到达无线路由器的无线网卡，无线网卡的驱动就会调用一个函数来报告包q已经被成功发出，并且包q将会被设置在无线网卡上的抓包程序抓取，并记录时间戳，这个时间即为。这里我们可以用近似的估计，原因如下：1）MAC层的ACK包在用户端不会排队等待，并且这个包的信道避让时间可以忽略不计，因为它相比其他包来说有着更高的优先级[9]。2）MAC层ACK包的传输时间可以忽略因为这个包很小，其传输时间也就是≈19us。3）MAC层ACK包出错丢失的情况是极少可以忽略的[10]。对于，我们可以用来进行估计。实际的测量结果表明在和之间的时间差，也就是处理延迟（PL），和下行无线延迟相比，是极小并且可以忽略的。类似的，我们用来估计，来估计。

有线包及其对应的无线加密包的匹配：

在本课题中我们遇到的另一个困难是，我们是从无线网卡获取的时间戳，然而我们的时间戳是从有线网卡截取的，我们如何能够得到属于同一个包的匹配的，对？SHAMAN [11]可以通过逐个匹配包的内容来合并无线流量和有线流量。然而，这对实际中的测量工作值不现实的，因为大部分的实际中的无线路由器（尤其是家用无线路由器）都会利用无线网络名称（SSID）的密码来加密发出的包。这个加密使得，从无线网卡获取的无线包是不可辨认的，以至于其无法与有线包进行匹配。为了解决这个困难，我们对装载了OpenWrt的无线路由器的驱动进行了小的修改，我们将在无线路由器内核时间戳存入的一个包结构中，它将在这个包结构的中保存，直到802.11 规定的MAC层的ACK包在被收到。此时，这个包也标记了时间戳的包，将会被交付给libpcap程序，然后我们将在MAC层的ACK包中重新获得时间戳。

### 2.3.3 上行无线延迟

准确的开始于用户端的上行无线延迟为-。是用户准备发送包p（即SYN包）的时刻。然而，在没有用户端协助的情况下，我们并不能获得。因此我们不能直接测量准确的上行无线延迟-。但是，我们可以用下一个上行包TCP ACK包的上行延迟-来估计上行无线延迟-。这里我们基于两个假设：首先，我们假设在到这段时间的无线环境是相似的。这是因为从到是一个相对较短的时间间隔，无线环境的改变很小。另外，我们假设用户端将在收到TCP SYN-ACK包后立即发送TCP ACK包，而没有任何的延迟，这个是基于延迟ACK机制的[12]，并且是在实际中一直在使用的。类似于下行无线延迟，我们用来近似的估计，然后我们得到的上行无线延迟约为-。

## 2.4 分解实际场景中TCP数据包的网络延迟

利用本课题中所提出的方法，我们能够准确的测量TCP数据包的下行无线延迟。对于数据包，下行无线延迟近似为-，并且和都可以直接在无线路由器上被准确的测量到，这个和TCP三次握手包是相同的。

在我们的定义中，对于TCP数据包有线延迟，是在有线网络上产生的往返延迟，即从无线路由器到服务器端的延迟，其并不包含服务器处理延迟。因为有线端的延迟和拥堵状态是相对比较稳定的，我们认为有线延迟主要是由无线路由器到服务器的传播延迟决定的，因此利用TCP三次握手包的有线延迟来估计TCP数据包的有线延迟是相对比较合理的做法。

在没有用户端协助的情况下，精确计算数据包的上行无线延迟是一个比较大的挑战。然而，我们可以利用这个之前的最近的TCP三次握手包的上行无线延迟来粗略的作为数据包上行无线延迟低值。这个方法会低估TCP数据包的上行无线延迟，原因如下：数据包的大小可能会比三次握手包的ACK包大，因此它们更加容易受到周围路由器的干扰，因为它们的接收时间比ACK包要长，就会有一个更长的上行无线延迟。TCP数据包的上行无线延迟，开始于TCP将数据包交付给操作系统，然后交付给802.11 MAC层，因此上行无线延迟不包含在客户端由于延迟的ACK而造成的上行无线延迟。

# 第三章 网络延迟测量结果

在这一章中，我们将采用我们在前两章中提到的方法，在我们部署的路由器中，来测量网络往返延迟。首先我们利用累积分布函数（Cumulative Distribution Function）来观察实际场景中的无线网络延迟是什么样子的。然后我们将网络往返延迟分解为有线延迟，上行无线延迟，下行无限延迟。我们发现，确实存在有无线路由器，它无线延迟在整个网络延迟中起主导作用。在这篇论文中，我们所有的测量都是在2.4GHz频段进行的，这个频段的用户比5GHz要高很多。当然，我们的测量和分析方法也是使用于5GHz频段的。

## 3.1 实际场景中的无线延迟

根据地点的不同，我们将无线路由器分为两类：实验室无线路由器和宿舍无线路由器。根据[13]的测量结果，我们将繁忙时间段（rush hour）定义为一天中企业无线局域网中流量最大的时间，也就是说，在这段时间里面，我们的无线路由器更加容易受到周围的干扰。而其余的时间段我们定义为正常时间段（normal hours）。对于实验室来说，我们设置繁忙时间为16：00到17：00，而宿舍的繁忙时间段为23：00到24：00。值得一提的是，我们这里提到的实验室是计算机系的研究生实验室，由研究生和教授共同使用。这个实验室没有确切的关闭时间，一般使用者会在17：00后离开吃晚餐，而很少会有人再回来继续工作。但是在他们离开之前，他们会更容易因为娱乐而产生大量的流量，例如看电影。根据[13]中提到的结果，实验室中在繁忙时间段产生的流量是18：00到19：00的3.5倍，是10：00到11：00的1.7倍。

根据这些分类条件，我们定义了4种没有重叠的分类：实验室繁忙时间段，实验室正常时间段，宿舍繁忙时间段，宿舍正常时间段。图3-1中展示了基于我们采集的所有数据的，这个4种分类的累积分布。正如预期的，不管是实验室还是学生宿舍，繁忙时间段的无线网络延迟都要大于正常时间段。另外，我们发现实验室的无线网络延迟要大于学生宿舍的无线网络延迟，这个是因为实验室周围的无线网络环境更差，关于这一点我们将在下一章解释。而且我们观察到对于实验室无线路由器来说，有超过50%的TCP包的无线延迟超过20毫秒，有超过10%的TCP包的无线延迟超过100毫秒。宿舍中的无线路由器有这个一个很长的高延迟部分，有10%的包的延迟超过30ms。



图 3-1 4种情况下无线延迟分布

## 3.2 往返网络延迟各部分的累积分布

利用前面提到的方法，我们将往返网络延迟分为上行无线延迟，有线延迟，下行无线延迟。图3-2中，我们展示了基于我们测量的数据绘制的无线延迟比率。这里我们将有线延迟分为国内服务器和国外服务器两类。对于国外的服务器，由于距离更长，有线延迟将更长（传播延迟更长）。另外，我们通过计算我们的有线包的数目，我们发现有98%的数据是来自或者发给国内的服务器的。通过图3-2，我们可以看到，有将近50%的数据包，在国内的服务器的分类下，无线延迟的比率占总的往返延迟的60%，这也就意味着，无线路由器，作为网络传输的最后一跳，已经在大部分情况下成为了最薄弱的环节。



图 3-2 无线延迟占比分布图

# 第四章 无线参数分析

在上一章中我们对于往返网络延迟进行了分解，我们对现实场景中的无线延迟有了一个整体的概观。这一章，我们将主要关注无线参数以及他们和无线延迟之间的相互关系。我们将会回答以下几个问题：

这些无线参数的分布是什么样子的？

无线参数和无线延迟间有着怎样的相关性关系？

那些无线参数对于无线延迟的影响最大？

这里我们首先要介绍我们如何将这些无线参数信息在我们采集的数据中提取出来。然后我们将要计算每种无线参数的累积分布函数，来看一下实验室无线路由器，宿舍无线路由器，以及它们在繁忙时间段及平常时间段的差别。接着，我们要直观的去看一下这些无线参数是否和无线延迟存在着线性关系。最终，我们将利用Kendall相关性系数以及相对信息增益（information gain）来评价这些无线参数的重要性。值得提到的是，本论文中所有的测量结果都是基于我们在上一章中提出的采集的原始数据。但是我们在接下来的章节中提到的测量和优化方法也适用于其他的数据库。

## 4.1 无线参数统计

表 4-1 无线参数列表

|  |  |  |  |
| --- | --- | --- | --- |
| 缩写 | 无线参数 | 描述 | 生成方式 |
| AU | airtime utilization | 所有流量占用无线信道的时间（%） | iw info |
| Q | queue length snapshot | 在硬件设备中包的排队数目 | debugfs |
| RR | retry ratio | IEEE 802.11 MAC层中包的重传比率（%） | iw info |
| RSSI | RSSI | 连接无线路由器用户的接收信号强度（dbm） | iw info |
| Ttx | transmitting throughput | 用户每10秒发送的比特数（byte） | ifconfig info |
| Trx | receiving throughput | 用户每10秒接收的比特数（byte） | ifconfig info |
| RPR | receiving physical rate | 路由器接收的物理速率（Mbps） | iw info |
| TPR | transmitting physical rate | 路由器发送的物理速率（Mbps） | iw info |

我们在表4-1中总结了我们所有在无线路由器上采集到的无线参数信息。本身OpenWrt操作系统提供了内部命令（列于表4-1的第四列），可以使我们采集多种无线参数信息，这些无线参数能够有力的反映当前的无线网络环境。这里我们将所有这些采集到的无线参数粗略的分为两类：有关无线路由器的参数信息（表4-1的上部）和有关无线路由器及用户的参数信息（表4-1的下部）。[14]中曾经用到表4-1中列出的部分参数来估计吞吐率。后面我们也将利用这其中的一些无线参数来训练我们的决策树模型。

为了避免对无线路由器造成过高的负载，我们实际中每10秒将采集一次无线参数。因此这里我们的发送吞吐率及接受吞吐率主要是反映了在这10秒内无线路由器用户的流量需求。考虑到可能对于无线路由器造成的高负载，更加频繁的采集发送吞吐率和接受吞吐率是没有意义的。对于剩余的这些参数，我们的采样间隔从20毫秒到10秒不等，错误范围保持在3%以内。因此我们每10秒的采样间隔是平衡负载和准确性后一个相对比较合理的选择。

## 4.2 无线参数的累积分布函数

在上一章中，我们展示了无线延迟在不同的时间及空间条件下的累积分布图，如宿舍繁忙时间，宿舍正常时间，实验室繁忙时间，实验室正常时间。我们也计算了在这四个互相不重叠的分类中，我们在图3-1中列出的无线参数的累积分布函数。我们想要借此来解释为什么在实验室中的无线延迟要大于宿舍。我们观察到，对于实验室无线路由器，信道占用率的中值是0.45，要大于在宿舍无线路由器的信道占用率的中值，0.42。我们也计算了在实验室和宿舍周围的环境中的无线路由器的数目，实验室周围有120台无线路由器，宿舍周围有24台。周围环境中的无线路由器的数目表示，对于实验室的无线路由器来说，它有更大的可能性遭受到来自周围无线路由器载波侦听及隐藏终端的干扰。除此之外，实验室无线路由器的接收信号强度的中值是-55dbm，要小于宿舍无线路由器的这个值-46dbm。接收信号强度表明的是连接此无线路由器的用户的信号强度。总的来说，实验室的无线路由器有这个更高的概率遭受到周围严重干扰，并且其有着较低的接收信号强度的问题，这是造成实验室无线路由器有着更高的无线延迟的主要原因。

## 4.3 无线参数及无线延迟相关性的初步展示



图 4-1 无线参数与无线延迟相关分布图

我们绘制了图4-1来解释无线参数对于无线延迟的影响。Y轴代表了所对应的上行无线延迟或下行无线延迟，每个小的图像中的X轴表示每个一个特定的无线参数的取值。我们根据不同的无线参数特性对于不同的无线参数，分了不同的区间进行了聚合：发送物理速率，接收物理速率是10Mbps，接收信号强度是10dbm，重传比率，信道占用率是0.1，排队长度是10，发送物理速率，接收物理速率是20Kbps。我们用以下的统计数据来展示每个集合中的信息：均值，50分位数，90分位数。我们绘制了下行无线延迟和信道占用率，重传比率，排队长度，发送物理速率，发送吞吐率之间的关系图，因为这些无线参数主要影响下行无线延迟。类似的我们绘制了上行无线延迟和接收信号强度，接收物理速率及接收吞吐率的关系，因为他们主要影响上行无线延迟。从图4-1，我们可以将这些无线参数分为两类：（1）与无线延迟有着明显关系的无线参数（信道占用率，重传比率，排队长度，发送物理速率，接收物理速率，接收信号强度），包含正相关和负相关。（2）与无线延迟无明显关系的无线参数（发送吞吐率，接收吞吐率）。

## 4.4 量化的相关性

这部分，我们将主要研究Kendall相关系数和信息增益，这两个都曾在[15]中被使用。通过量化无线参数和无线延迟之间的相关性关系，我们可以为下一章的决策树模型获取更多的中间信息。同时信息增益能够帮助我们在下一章的决策树模型中挑选参数。

### 4.4.1 相关系数

表 4-2 无线参数与延迟相关性列表

|  |  |  |
| --- | --- | --- |
| 无线参数 | Kendall相关系数 | 相对信息增益 |
| 信道利用率（AU） | 0.86 | 0.05 |
| 接收信号强度（RSSI） | -0.5 | 0.06 |
| 重传比率（RR） | 0.4 | 0.08 |
| 发送物理速率（TPR） | -0.3 | 0.11 |
| 接收物理速率（RPR） | -0.2 | 0.09 |
| 接收吞吐率（Trx） | -0.17 | 0.01 |
| 排队长度（Q） | 0.15 | 0.007 |
| 发送吞吐率（Trx） | -0.006 | 0.02 |

为了对比不同的无线参数，我们利用Kendall相关系数作为我们的衡量标准。这是因为Kendall相关系数不需要数据遵循某种特定的分布，例如高斯分布。在我们对这些无线参数根据上面提到的间隔进行聚合后，我们计算了分入每个集合内的无线参数的相关系数的平均值。然后我们对于每个无线路由器及用户对，我们将会计算上行无线延迟或下行无线延迟和不同的无线参数之间的Kendall相关系数。这个Kendall的中值展示在表4-2的第二列。我们可以看到信道利用率，接收信号强度，重传比率在相关系数中排到前三名。

### 4.4.2 相对信息增益

[15]中指出，基于相关性的分析不能很好的体现非单一的相关性，例如前面展示的发送吞吐率，接收信号强度，接收吞吐率，排队长度和接收物理速率。下面我们将利用相对信息增益来解决这个问题。相对信息增益是用来表明在我们已知X的情况下对于预测Y有多大的帮助，换言之，即为在我们已知X的情况下，对于Y的不确定性的减少有多大的帮助。我们首先将下行无线延迟或上行无线延迟以10毫秒为间隔进行离散化来得到Y。对于每一个每一个无线参数来计算相对信息增益。我们将对不同的无线路由器重复这个过程，这些值的中值将被展示在表4-2的第三列。表4-2说明发送物理速率有着最大的相对信息增益。基于对相对信息增益的定义，这里的相对信息增益代表了在已知某个无线参数后，无线延迟的不确定性降低了多少。相对信息增益的范围是0到1，当它为0的时候说明无线延迟和无线参数之间是完全独立的。当相对信息增益大于0的时候，表明此无线参数对于预测无线延迟是有帮助的。[16]中列出了发送物理速率变化的三个原因：（1）接收信号强度的改变。（2）由于干扰而造成的丢包，这和我们研究的重传比率和信道占用率有关。（3）为了评价无线信道质量而采用的速率采样算法。这也就说明发送物理速率是和接收信号强度，重传比率和信道占用率是相关的，并且这也是一个综合考虑接收信号强度，重传比率和信道占用率的参数。这同时也解释了为什么发送物理速率在表4-2中体现了最高的相对信息增益。而在下一章，在决策树种结合其他的参数考虑后，这个参数将被进一步放大。我们从表4-2中获得的另一个信息就是虽然信道占用率有着最高的Kendall相关系数，但是它却不是预测下行无线延迟的最好参数。这个就可以用来解释为什么在图4-1（a）中只有在信道占用率大于0.5时，下行无线延迟才会增大。而在信道占用率小于0.5时，下行无线延迟却没有什么变化。除此之外，发送吞吐率和接收吞吐率都和上行无线延迟或下行无线延迟有很小的相关性。

## 4.5 无线参数分析总结

这里我们将总结我们通过以上分析得到的主要结论：

1）发送物理速率在预测无线延迟时有着最高的信息增益。

2）只有信道占用率超过一定的阈值的时候，无线延迟才会受到影响。在我们的测量中，这个阈值是0.5

3）不管是从Kendall相关系数还是从信息增益来看，用户在每10秒内的流量需求都和无线延迟没有明显关系。

# 第五章 无线延迟模型

由于速率适应算法[16]，发送物理速率是依赖于信道占用率，接收物理速率，重传比率相关的，同时它也是对当前信道状况的一个很好的反映。但是对于大部分的无线参数，如排队长度、发送吞吐率、接收吞吐率，他们和无线延迟之间的关系都是非线性的。那我们如何用这些无线参数来预测无线延迟呢？我们要用多少的无线参数就能够准确的预测无线延迟？为了解决这些挑战，在这一章中，我们将利用机器学习来建立无线参数和无线延迟之间的模型。我们首先会提出对于我们想要解决问题的一个严谨的定义。然后我们将介绍我们用来解决这个问题的机器学习的方法。最后我们将利用10折交叉确认法来评价我们的方法的准确性。

## 5.1 问题陈述

对于家用无线路由器的使用者来说，利用无线参数来预测无线延迟来帮助他们对无线路由器来进行优化或者重新配置是十分重要的。然而，由于不同的无线参数之间内部依赖，又相互影响，所以要做这件事的挑战很大。我们想要综合的运用我们可以在家用无线路由器中获取的无线参数来预测下行无线延迟，上行无线延迟及双向无线延迟。因此我们给出的问题陈述如下：给出一系列的无线参数，预测无线延迟（包含上行延迟和下行延迟）。

## 5.2 模型训练

考虑到无线延迟和表4-1中无线参数之间的复杂关系，我们选择用机器学习作为我们的工具来解决我们在上面提出的问题。这里十分关键的就是选择一个格式的机器学习算法。一个适用于我们问题的算法应该符合以下特性：

训练出的模型应该有足够的准确性来利用无线参数预测无线延迟。

训练出的模型应该易解读，并且能够为进一步的优化提供指导。

这里我们将我们的问题建立为一个二分类的问题。我们利用一个阈值将无线延迟分为快（fast）和慢（slow）两类。每20毫秒至30毫秒的最后一公里的延迟将会导致页面加载时间2秒钟的增大[3]，这同时也会导致收益的降低[2][17]。根据以上的数据我们进行了封底的运算，无线延迟应当保持在20毫秒到30毫秒这个阈值以下，来满足用户的需求。因此我们将双向延迟的阈值设为25毫秒，单向延迟（上行无线延迟或下行无线延迟）的阈值设为12.5毫秒。

在经过一系列适用性实验后，我们选择决策树作为我们的分类方法（利用的python的scikit-learn包[18]）。在我们的实验中，决策树的效果比贝叶斯回归预测和线性回归预测都要好。我们分析，由于我们的数据中各个无线参数相互依赖，所以贝叶斯回归预测和线性回归预测的效果并不好。另外我们的无线参数和无线延迟之间的关系也都不是线性的。然而，决策树模型对于我们提供的参数内部的依赖性以及参数和无线延迟之间的相关性都没有做任何的假定。

## 5.3 决策树模型介绍

决策树（Decision Trees ，DTs）是一种无监督的学习方法，用于分类和回归。它通过学习数据的相关参数，对数据中蕴含的决策规则建模，以预测目标变量的值。某些情况，例如下面的例子，决策树通过学习模拟一个包含一系列是否判断的正弦曲线。树越深，决策树的规则和拟合越复杂。

### 5.3.1 决策树的优点

易于理解和解释。数可以可视化。

几乎不需要数据预处理。其他方法经常需要数据标准化，创建虚拟变量和删除缺失值。决策树还不支持缺失值。

使用树的花费（例如预测数据）是训练数据点(data points)数量的对数。

可以同时处理数值变量和分类变量。其他方法大都适用于分析一种变量的集合。

可以处理多值输出变量问题。

使用白盒模型。如果一个情况被观察到，使用逻辑判断容易表示这种规则。相反，如果是黑盒模型（例如人工神经网络），结果会非常难解释。

可以使用统计检验检验模型。这样做被认为是提高模型的可行度。

即使对真实模型来说，假设无效的情况下，也可以较好的适用。

### 5.3.2 决策树的缺点

决策树学习可能创建一个过于复杂的树，并不能很好的预测数据。也就是过拟合。修剪机制（现在不支持），设置一个叶子节点需要的最小样本数量，或者数的最大深度，可以避免过拟合。

决策树可能是不稳定的，因为即使非常小的变异，可能会产生一颗完全不同的树。

学习一颗最优的决策树是一个NP-完全问题。因此，传统决策树算法基于启发式算法，例如贪婪算法，即每个节点创建最优决策。这些算法不能产生一个全家最优的决策树。对样本和特征随机抽样可以降低整体效果偏差。

概念难以学习，因为决策树没有很好的解释他们，例如，XOR, parity or multiplexer problems。

如果某些分类占优势，决策树将会创建一棵有偏差的树。因此，建议在训练之前，先抽样使样本均衡。

### 5.3.3 数据预处理

因为sklearn所实现的决策树模型是一个二分树模型，它是在每个节点，选定一个特定的阈值，然后对于大于这个阈值或者小于这个阈值的数据进行分类，也就是说它只能根据连续的数值特征进行准确分类。而对于分类属性，就需要进行二维数据的匹配。

同时我们在训练模型的时候，也遇到了正例过多，而导致正负例训练结果准确度相差较大的情况。对于这种情况，我们采取了两种解决办法。首先，为了避免抽样而导致的优势分类的样本的丢失，我们对于小类的样本进行了，重复增加的方法。我们对于小的样本进行随机的采样，然后进行复制，而使其样本数量与大类样本差别不至于过低。另外我们在进行决策树决策时，我们调整了小样本的权重，而使得小类样本在决策中起到更加重要的作用。

## 5.4 评估

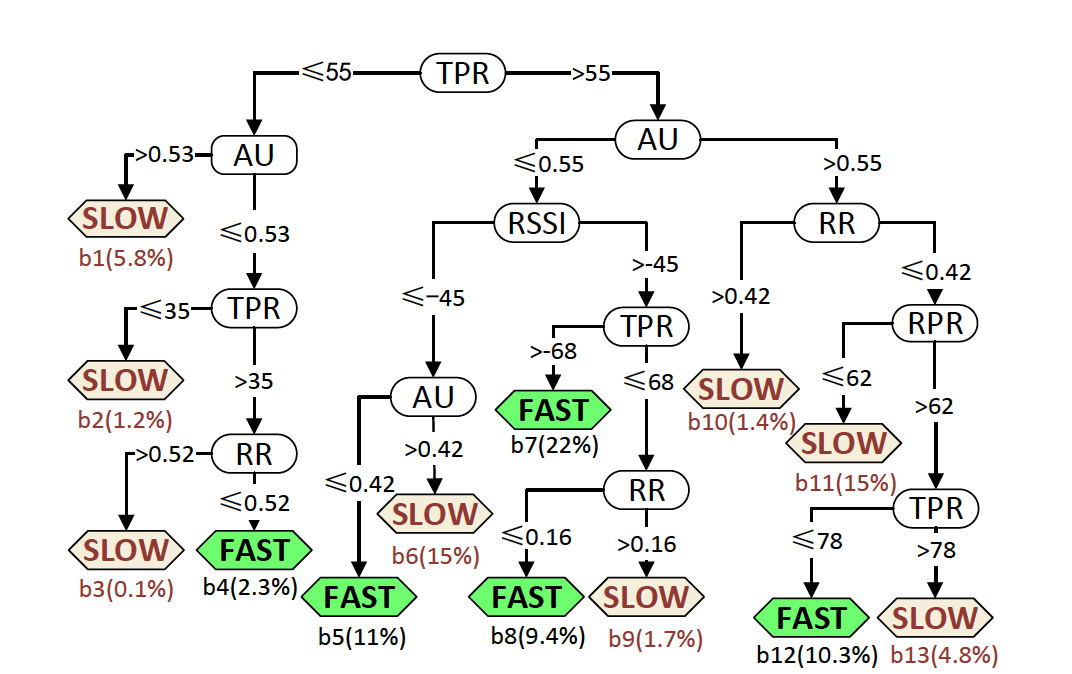


图 5-1 决策树模型

我们采用10折交叉方法来评价决策树的准确性。我们随机的将我们整体的数据集分为两个部分：90%的数据用来训练决策树的模型，剩余的10%的数据我们用来对此模型进行评价。对于评价此模型，我们将首先用一个阈值来将我们测量的延迟分为两类——快（FAST）和慢（SLOW）。我们用这些作为真实的数据，来评价我们训练的模型分类得到的结果。我们将用分类的接收特性值（Receiver Operating Characteristic）来评价这个两个模型。这个包括准确性（abbreviated缩写为ACC），真阳率(true positive rate缩写为TP)以及假阳率（false positive rate缩写为FP）。这个结果总结在表5-1中，这里我们对上行无线延迟，下行无线延迟及双向延迟都进行了分类。

表 5-1 决策树准确度结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法 | 延迟类型 | 准确度 | 真阳率 | 假阳率 |
| 决策树 | 下行无线延迟 | 0.78 | 0.76 | 0.24 |
| 上行无线延迟 | 0.68 | 0.67 | 0.27 |
| 上下行无线延迟和 | 0.77 | 0.79 | 0.31 |

决策树模型：

这里我们在训练决策树模型的时候，没有将排队长度，发送吞吐率，接收吞吐率加入我们的训练参数中，因为他们的信息增益很低。初次之外，我们将决策树的最大深度设为15，最小的叶节点大小为100。对于下行无线延迟分类，目前我们的决策树模型的准确率是78%。

而在使用随机森林的时候，这个准确率将会被提升，因为随机森林是对多棵树进行表决选择的。在我们的实验中，在使用随机森林的时候，我们树的个数是200，准确率可以达到81%。随着随机森林树的个数的增加及深度的增大，准确率和真阳率都会进一步的提升。但是我们并不建议去做这件事，因为如果这样的话，树的数目以及每棵树都会很大，我们将无法对这个模型有直观的感受。为了给无线路由器的使用者有更加直观的感受，我们在图5-1中绘制了减枝后的决策树模型。我们利用这个模型将下行无线延迟分为快（FAST）和慢（SLOW）两类。括号里面的数字代表了利用我们采集的两个月的数据，将会有多少数据落入每个分支内。通过追溯从根节点到某个特定叶节点的路径，我们就可以知道在某个特定的分类下发生了什么。这里要说明的是，在scikit-learn包里面所实现的决策树算法是一个贪心算法[18]，例如在每个结点都会基于当前节点进行最优选择。在图5-1中发送物理速率被选择作为决策树的根节点，因为发送物理速率在表4-2中有着最大的信息增益。而在决定其他层的节点的时候，是结合了从从此结点到根节点所有的信息增益的结果。用结合的信息增益来挑选节点的参数时，是和在表4-2中计算对单一的参数计算信息增益是不同的。这也就解释了为什么在表4-2中，信道占用率的信息增益只排名第五，而其却在决策树模型中排在第二层。换句话说，信道占用率和发送物理速率结合在一起的作为分类节点效果要比其他的好。

# 第六章 基于预测模型的优化方案

在第五章中，我们利用我们采集的数据集训练了一个普试的决策树模型。如图5-1中的决策树模型不止可以被应用于我们的无线路由器。它可以帮助我们对于实际场景中的无线路由器对无线延迟问题的理解，故障定位及优化。对于一个特定的无线路由器，我们需要将它的无线参数对应到我们在图5-1中的决策树模型中，然后找到它从根节点到叶节点的路径。这里我们将重点提出两个实例来证明，利用我们这个模型来进行优化的效果。

## 6.1 优化的三个步骤

这里我们将分三个步骤利用我们已经训练好的决策树的普试模型来对我们的无线延迟进行优化。首先我们将在特定的时间段内，对于我们想要优化的无线路由器收集包含相关无线参数的包。然后我们将根据这些包，利用图5-1中的决策树模型进行分类，并记录那些分到慢（SLOW）节点的路径，并从中选择出现概率最大的一条。那么这条路径上所有涉及到的无线参数，就形成了我们待优化的参数集。这里我们将不需要对每个包来讲我们测量的无线延迟和预测结果进行对比。最终，每个无线路由器的使用者都可以尝试对这个参数集的参数进行调整，例如增大发送信号强度，调整信道，更改无线路由器的位置。对于大部分的情况，只进行一轮的三步优化即可，可能对于个别的情况，可能要进行多次这样的三步优化就可以达到一个很好的优化结果。

这个三步优化可以给无线路由器的使用者其无线路由器的可优化参数及优先级的列表。没有这个决策树的模型，无线路由器的使用者对于如何调整他们的无线参数完全是没有概念的。而且我们的模型可以在实际采取一定的措施前，就对可能造成的延迟的改变进行预测。

## 6.2 调整无线路由器的位置

这个无线路由器的使用者曾经抱怨这个无线路由器的延迟过长，而且采用了多种办法来减小延迟（如增大无线路由器的发送信号强度，调整到不同的信道）,但是都没有对无线延迟有很好的改善。我们的决策树模型就帮助他解决了这个问题。首先我们对这个无线路由器采集了24个小时的数据包及无线参数数据，然后使用我们在图5-1中的模型进行分类。我们把包落入每个分支的比例列在表6-1的第二行中，这个表格说明，58%的时间内，这个无线路由器都会进入SLOW的分类也就是，分支b6。这也就意味着这个无线路由器的发送物理速率大于55Mbps，信道利用率小于0.55，接收信号强度小于-45dbm。

我们不能手动的调整发送物理速率，因为它只能在设备商的芯片里调整。而接收信号强度表明了用户接收的信号强度，它是由用户的发送强度以及用户到无线路由器的距离决定的。增大无线路由器的发送信号强度并不能提升接收信号强度。我们发现，虽然用户之间的距离很近，但是他们距离无线路由器的距离很远，无线路由器是在另外的一个房间里面。因为用户只能待在一个特定的空间内，因此我们调整了无线路由器放置的位置，使他能够更加的接近用户，调整后的结果展示在表6-1的第三行中。这次大部分的包都落入了快（FAST）的分类中，也就是分支b7。我们在图6-1中展示了调整前后的下行无线延迟，可以看到无线延迟的中值从50毫秒降低到10毫秒，有了5倍的提升。这里要说明的是，不管是这个例子还是下一个，我们都对调整前后的下行无线延迟进行了一个星期的测量，来消除由于流量不同而可能造成的影响。

表 6-1 无线路由器位置调整性能对比

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分支 | b1 | b2 | b5 | b6 | b7 | b10 | b11 | b12 | b13 |
| 优化前 | 0.1 | 0.8 | 3 | 58 | 0 | 6.7 | 13.4 | 4.2 | 13.8 |
| 优化后 | 0 | 0 | 0 | 0.5 | 61.2 | 0.4 | 3.5 | 0.4 | 34 |



图 6-1 无线路由器位置调整前后下行无线延迟分布图

## 6.3 信道选择

一个宿舍无线路由器的拥有者抱怨他在每天的23点到24点经常会遇到高延迟的情况，这个也是宿舍的繁忙时间段。我们将在23点到24点采集的包的无线参数灌入决策树模型,然后我们将落入每个分支的包的比例列在表6-2的表格里。从表6-2的第二行，我们可以看到有82%的情况下会被分到慢（SLOW）的分支里面也就是分支11。也就说明当时是发送的物理速率大于55Mbps，信道占用率大于0.55，重传比例小于0.42以及接收的物理速率小于62Mbps。

表 6-2 无线路由器信道调整性能对比

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 分支 | b1 | b6 | b7 | b10 | b11 | b12 |
| 优化前 | 3 | 0 | 0 | 0 | 82 | 15 |
| 优化后 | 0 | 12.6 | 82 | 0.9 | 4.5 | 0 |

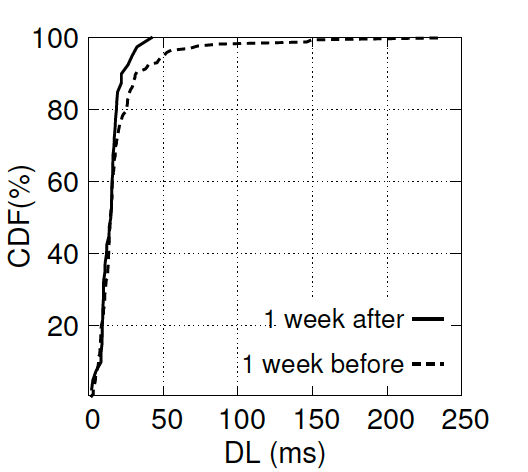


图 6-2 无线路由器信道调整前后下行无线延迟分布

在检测了这个无线路由器周围的信道后，我们发现这个无线路由器之前一直在信道6，而信道6的平均信道占用率是0.6。而信道1的平均信道占用率是0.52。当我们将这个无线路由器的信道调整到信道1后，信道1的平均信道占用率应在0.55以下。并且回想前面的结果（图4-1），当信道占用率大于0.5后，它就能极大的开始影响无线延迟了。因此我们将我们的无线信道调整到信道1，然后从23点到24点数据的分类结果就在表6-2的第三行显示。现在我们可以看到有82%的包会进入快（FAST）的分类中，也就是分支b7。同样的，我们也绘制了在调整前后，信道占用率和下行无线延迟的累积分布函数，结果在图6-2中展示。从图6-2中，我们可以看到在信道调整后，在繁忙时间段的最大无线延迟从250毫秒降低到50毫秒。这个改善的结果确认了我们的优化方法的有效性。

# 第七章 优化的信道选择算法

## 7.1 最小拥塞信道选择算法

在受到干扰的环境下，信道调整是最有效也是最简单的优化无线路由器性能的方法[19]。在实际的操作中，最小拥塞信道选择算法（Least Congestion Channel Search）是当前基于单个无线路由器，最为常用的无线信道选择算法[20]。其算法执行的核心如图7-1所示。由于在mac 802.11协议所规定的2.4GHz的无线信道中，只有1、6、11三个信道互不干扰，因此最小信道拥塞选择算法，就是将无线路由器的无线信道依次调整到1、6、11三个信道，并且同时采集它的无线参数，并根据这些无线参数，选择一个性能最好的信道，然后将无线信道调整到这个信道上。经过一定的时间间隔后，再次重复之前的过程。总的来说最小信道拥塞选择算法就是一个轮训的算法。基于这个算法最初的目的——减少无线信道之间的拥塞，因此这个算法最常用的选择最优无线信道的无线参数是信道利用率。而基于之前的结果，我们可以看到在信道利用率小于0.5时，无线延迟基本没有变化，也就是说，最小的无线信道利用率并不代表着最小的无线延迟。而很显然的是，对于用户来说，他所关注的重点并不在于信道是否拥塞，而是当前的无线延迟是否是最低的。因此我们想要采用我们所测量的无线延迟结果直接作为最小拥塞信道选择算法的输入参数。以下我们将说明，我们使用wing的无线延迟测量结果能够更好的优化无线路由器的无线延迟。

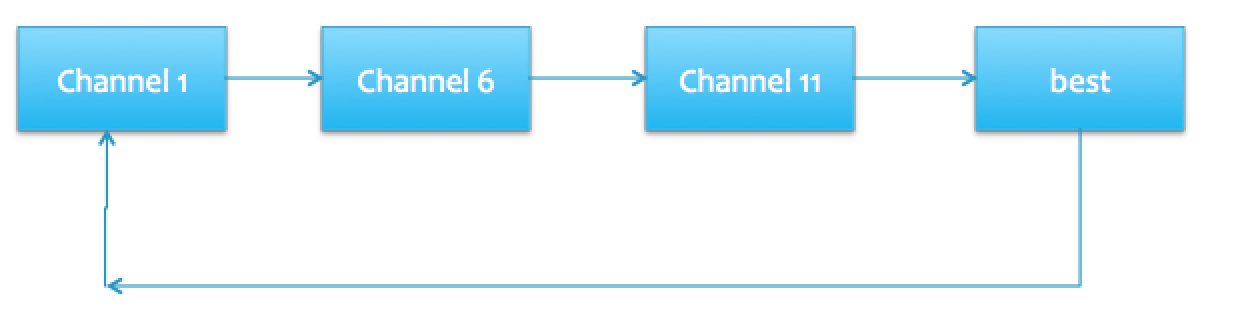


图 7-1 最小拥塞信道选择算法

## 7.2 实验场景介绍

为了实现对于这两种不同参数的公平评价，我们采用了背靠背，同本地流量，同干扰流量的实验方法。我们在我们部署的无线路由器中，选择了7个最为活跃的无线路由器（分布在不同的学生宿舍中），并且记录了他们一周的流量行为。另外我们利用树莓派及两台无线路由器，实现了对于这个无线流量的重放。实验拓扑如图7-2所示。我们将利用在树莓派上的iperf工具，来对我们的无线流量进行重放，并且同时在无线路由器1上我们将执行基于延迟的最小拥塞信道调整算法，在无线路由器2上我们将执行基于信道利用率的最小拥塞信道调整算法。在我们的实验中，我们检测每个信道的时间间隔是5分钟，我们保持在最好信道的时间是30分钟。通过这种的拓扑及实验方法，我们可以保证每个无线路由器所受到周围无线路由器的无线干扰是相同的。并且由于他们之间的无线流量相同，因此，他们互相之间所产生的无线干扰也是相同的。并且为了公平起见，我们将在这个过程中，针对每种算法，收集我们由ping获取的无线延迟结果，并将其作为我们评价两种算法的参数。

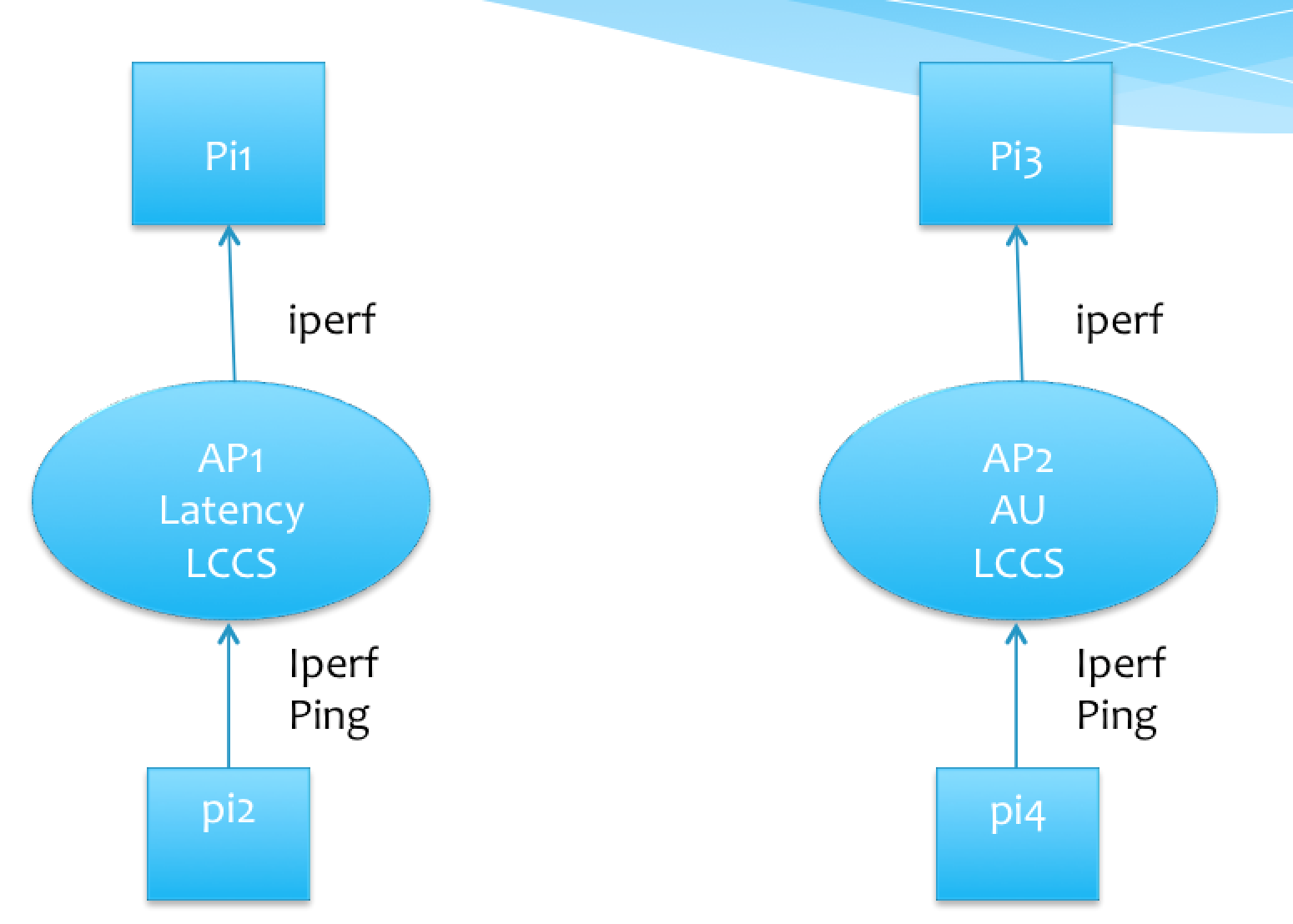
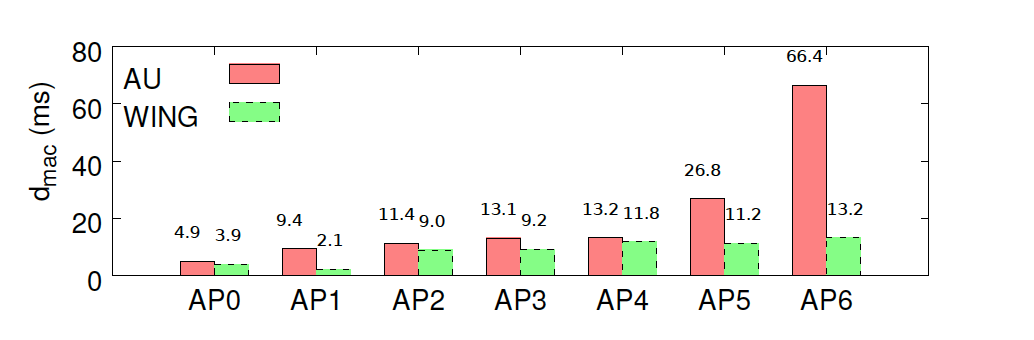


图 7-2 背靠背实验拓扑

## 7.3 实验结果及分析

图7-3展示了对于不同的算法（也就是对于最小拥塞信道选择算法使用不同的输入参数）我们所测量的无线延迟。我们可以看到对于所有的测试路由器，我们利用无线延迟作为信道选择参数最终的无线延迟，都要小于利用信道利用率最终的无线延迟。其中最大的延迟降低出现在无线路由器6，一共有着5倍的延迟降低。

 图 7-3 背靠背实验结果展示

由以上的结果可以看出，信道利用率只是对于当前无线信道占空比的一个反映，它可能对于无线延迟有着一定的影响，但是他并不能直接的反映无线延迟。并且信道利用率的采样是在一段时间内平均的结果，却不能直接的说明，高的信道利用率是否对于当前本机的数据包产生了影响。这里我们可以举一个极端的例子，如果在某无线路由器周围有一个高吞吐率的无线路由器正在发包，而其流量正好是和我们的无线路由器的流量错开，那么在这个一段时间内的信道利用率一定是一个很高的值，但是却对当前我们的无线流量并没有任何的影响。而我们在本文章中使用的无线延迟，是对每一个数据包的采样，由此可以规避对于大的延迟包的平均效应，而更加的容易识别出这其中可能存在的高延迟，从而选择一个更好的信道。

## 7.4 关于信道调整的补充说明

鉴于以上我们提供的优化方法，这里我们需要对于用户的无线信道进行相对频繁的调整。首先为了不影响用户服务的使用，我们在用户的流量高峰期是不会对无线信道进行调整的。其次我们采用了mac 802.11n协议中所规定的，使用户与无线路由器协同调整信道的方法，而规避了由信道调整可能带来的用户断连。

在之前的方法中，我们采用openwrt 内部的uci wifi命令进行信道切换，这将会重启无线模块，这就会使用户收到来自无线路由器的取消认证的包，也就是告诉用户：此路由器在这段时间内无法连接。然后用户就会重新的对周围的无线路由器进行扫描，选择一个最好的无线路由器进行连接，并且需要向新的路由器发送认证连接的请求。这个过程不但可能会导致用户的断连，而且可能会使用户在这个时候就直接连接了一个新的无线路由器，而放弃了原来的无线路由器。因此整个这个过程是得不偿失的。

而在mac 802.11n中所规定的无线路由器调整方法中，在无线路由器调整信道之前，它会向用户发送探测包，来告诉用户它要调整信道了，然后在约0.9秒的倒计时后，无线路由器开始调整信道，同样的用户也会调整信道，这一切的行为都是发生在mac层的，而不会对上层的应用产生影响。经过我们的抓包实验后，也证实了这种方式，可以保证TCP正常的通信过程。

目前的OpenWrt内部的hostapd模块已经集成了这个功能，我们只需要使用hostapd\_cli就可以完成以上的信道切换过程。

# 第八章 相关工作

我们的测量无线路由器延迟的算法是受到[17]的激发，并且和[21]有一定的相似性。我们对于无线延迟测量的结果和[21]具有一致性。我们提升在于我们无线延迟进行了分解，分为了上行无线延迟和下行无线延迟。另外我们用决策树模型来预测无线延迟，极大的帮助了我们的优化工作。

无线测量：SHAMAN[11]通过抓包和合并在有线及无线端的包，来测量下行无线延迟。当我们的家用无线路由器对于其进行了加密后，这个方法就是不适用的。我们通过对于OpenWrt驱动的一些修改，得到了一个比较通用的方法来测量延迟。而[22]中，通过抓取通过学校网关交换机的包来测量无线延迟。但还是它不能针对某一个无线路由器-用户对来测量延迟。[14][16][23]将无线路由器部署在不同的地方，来进行激进的测量，但是他们缺少对于无线延迟的分析研究。

优化指导：[15][24]采用了机器学习的方法来研究移动或网页的视频服务质量和蜂窝网络之间的关系，这个和演技无线环境中的无线参数是不同的。

# 第九章 总结

基于在现实场景中部署的47台无线路由器的测量结果，在此篇论文中，我们了解到真实世界的无线延迟是什么样子的。并且这篇论文提出了一个在不需要用户端和芯片厂商支持的方法，来分解网络往返延迟。另外我们提出了决策树的模型来帮助我们对无线延迟的理解、故障定位及优化。并且利用我们测量的无线延迟对于信道选择算法进行了优化。

我们的测量结果显示，无线，作为当前网络重要的最后一跳，已经成为当前往返网络中最薄弱的一环。这也说明我们急切的需要更大的经历去研究无线延迟。这篇论文中，我们采取了一套系统的研究方法去研究实际场景中的无线延迟。我们相信，这是对我们进一步研究无线延迟的十分有意义的一步。

# 

# 参考文献

[1] Cisco. Cisco visual networking index: Forecast and methodology, 2013–2018. CISCO White paper, pages 2013–2018, 2013.

[2] Akamai study. http://goo.gl/2pwozG.

[3] S. Sundaresan, N. Feamster, R. Teixeira, et al. Measuring and mitigating web performance bottlenecks in broadband access networks. In ACM Internet Measurement Conference, 2013.

[4] EXFO. http://goo.gl/OMGKnE. Accessed: 2016-01-20.

[5] K. Thompson, G. J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. Network, IEEE, 1997.

[6] Barrier Breaker. https://openwrt.org/. Accessed: 2016-01-20.

[7] Libpcap. http://en.wikipedia.org/wiki/Pcap. Accessed: 2016-01-20.

[8] Rsync. https://rsync.samba.org/. Accessed: 2016-01-20.

[9] Mustafa Ergen. Ieee 802.11 tutorial. University of California Berkeley,70, 2002.

[10] Z. Tang, Z. Yang, and J. He, et al. Impact of bit errors on the performance of dcf for wireless lan. In ICCCAS. IEEE, 2002.

[11] Y.-C. Cheng, M. Afanasyev, and P. Verkaik, et al. Automating cross-layer diagnosis of enterprise wireless networks, volume 37. ACM, 2007.

[12] Robert Braden. Rfc-1122: Requirements for internet hosts. Request for Comments, pages 356–363, 1989.

[13] K. Sui, Y. Zhao, and D. Pei, et al. How bad are the rogues impact on enterprise 802.11 network performance? In INFOCOM, 2015.

[14] A. Patro, S. Govindan, and S. Banerjee. Observing home wireless experience through wifi aps. In MobiCom, pages 339–350. ACM, 2013.

[15] F. Dobrian, V. Sekar, and A. Awan, et al. Understanding the impact of video quality on user engagement. SIGCOMM, 41(4):362–373, 2011.

[16] P. Ioannis, L. Henrik, and S. Augustin, et al. Characterizing home wireless performance: The gateway view. In INFOCOM, 2015.

[17] Y. Chen, R. Mahajan, and B. Sridharan, et al. A provider-side view of web search response time. SIGCOMM, 43(4):243–254, 2013.

[18] F. Pedregosa, G. Varoquaux, and A. Gramfort, et al. Scikit-learn: Machine learning in python. The Journal of Machine Learning Research, 12:2825–2830, 2011.

[19] Ashish Patro and Suman Banerjee. Coap: A software-defined approach for managing residential wireless gateways.

[20] J. Geier. Assigning 802.11b access point channels. In WiFi Planet , 2004.

[21] S. Sundaresan, N. Feamster, and R. Teixeira. Measuring the performance of user traffic in home wireless networks. In Passive and Active Measurement, pages 305–317. Springer, 2015.

[22] X. Chen, R. Jin, and K. Suh, et al. Network performance of smart mobile handhelds in a university campus wifi network. In IMC, pages 315–328. ACM, 2012.

[23] S. Grover, M. S. Park, and S. Sundaresan, et al. Peeking behind the nat: an empirical study of home

networks. In IMC, pages 377–390. ACM, 2013.

[24] A. Balachandran, V. Aggarwal, and E. Halepovic, et al. Modeling web quality-of-experience on cellular networks. In MobiCom, pages 213–224. ACM, 2014.

# 

# 致 谢

毕业论文暂告收尾这也意味着我在北京邮电大学的四年的学习生活既将结束。回首既往，这一段宝贵的时光能于这样的校园之中能在众多学富五车、才华横溢的老师们的熏陶下度过实是荣幸之极。在这四年的时间里我在学习上和思想上都受益非浅。这除了自身努力外与各位老师、同学和朋友的关心、支持和鼓励是分不开的。

在本论文的写作过程中我的导师——清华大学计算机系裴丹老师倾注了大量的心血，从选题到开题报告从写作提纲到一遍又一遍地指出每稿中的具体问题严格把关循循善诱，在此我表示衷心感谢。并且整个论文完成过程中也离不开裴昶华学长的悉心指导，使得我在完成毕业论文的这将近一年的时间内，各项能力都有了质的飞跃，这将为我未来5年的博士生活打下良好的基础。

虽即将离开大学校园，但是我会不忘初心，努力学习，为社会贡献自己的力量。

# 

# 附 录