标题：构建WiFi环境下Web用户体验与网络性能的关系模型

引言

研究背景

随着智能手机和平板电脑等智能设备的普及，越来越多的设备只能通过无线保真技术（Wireless Fidelity，以下简称Wi-Fi）或蜂窝网络技术等无线方式接入互联网。同时，因为无线上网的便捷性，越来越多的用户也选择采用无线方式而非传统的有线方式上网。据统计，在2013年整个互联网流量的55%是通过Wi-Fi接入网络的。（I Cisco. Cisco visual networking index: Forecast and methodology, 2013–2018. CISCO White paper, pages 2013–2018, 2013.）不仅如此，支持Wi-Fi的设备从2012年到2017年，预计从50亿台增加至200亿台。（ABI Research. https://www.abiresearch.co m/press/over-5-billion-wireless-con nectivity-chips-will-sh.）这些都表明Wi-Fi已经成为当今互联网最主要的网络接入方式。

相对于传统的有线网络，Wi-Fi网络的环境更加复杂，其服务受到更多因素的影响，服务的稳定性也更加难以保证。在Wi-Fi普及的潮流下，为了给人们提供更方便的上网服务，很多商家都在公共场所布置了大量的无线接入点（Access Point，简称为AP），包括教室、商场、餐厅等。如今的家庭中也会有一个甚至多只多个无线接入点。在给用户提供了方便的上网接入同时，这些无线接入点的相互干扰，可能会影响Wi-Fi的服务性能，给用户带来难以预测的影响。然而由于影响无线网络的因素众多，即使出现上网的性能不能让人满意的问题，用户也难以找到其中的原因。

在互联网众多的服务中，基于HTTP的Web网页浏览服务着数以十亿计的用户，是互联网中最重要也是最基础的应用。（[6] Wang, Xiao Sophia, et al. "Demystifying Page Load Performance with WProf." NSDI. 2013.）网页浏览对于网络性能比较敏感，当网络性能不好时，经常出现网页加载缓慢甚至网页无法加载的问题。这些问题会严重影响浏览网页的用户的体验。用户体验与用户对一个服务提供商的忠诚度至关重要，直接影响着服务提供商可能获得的利润。例如一个用户进行网购时他发现网站无法及时显示他查询的物品或者他进入购物车后迟迟没有反应，他很可能更换到另一个网购网站完成购物，这对于最初的购物网站就是一个损失。统计显示平均增加500毫秒的延迟会导致Bing搜索的利润减少1.2%（S. Souders. Velocity and the bottom line. http://radar.oreilly.com/2009/07/velocity-making-your-site-fast.html, July 2009.），平均增加400毫秒的延迟会导致Google搜索的利润减少0.74%（J. Brutlag. Speed matters for Google Web search. http://services.google.com/fh/files/blogs/google\_delayexp.pdf, June 2009.）。由此可见，Web网页浏览的用户体验关系重大。

虽然Web用户体验非常重要，但是想要简单地获得反应Web用户体验的评价标准却非常困难。与用户相关的信息最主要记录在客户端上，而用户与设备之间的交互更是在客户端才能直接获得。如果希望在客户端上抓取并记录用户的信息用于分析，必须提前让用户安装相应软件，这给大规模部署带来了困难。这些困难导致网络运营商和服务提供商难以有针对性地提升用户体验。

综上所述，Wi-Fi环境下的Web浏览是当今互联网的主要服务，而其用户体验有着巨大的价值和意义。然而，目前缺乏一种有效的在非客户端上测量WiFi环境下Web用户体验的方式。另外，Wi-Fi的多个因素均可能影响用户体验，不同因素对Web用户体验的影响程度也有待研究。本文拟采用决策树和朴素贝叶斯分类基于无线接收点采集的信息构建Wi-Fi参数与用户体验的关系模型，在这个模型的基础上指出对用户体验影响关系较大的Wi-Fi参数，

并给出提升用户体验的指导意见。

相关工作

无线接入点在家庭网络中扮演着一个重要的网络入口的角色，在上面有可能获得更加详细的关于用户设备及用户网页浏览情况的信息，而这些信息是在NAT之后就难以分辨的。基于此，BISmark团队在OpenWRT路由器上部署了Mirage，用于更加精确地监测在NAT之后的家庭网络的性能变化（Measuring and Mitigating Web Performance Bottlenecks in Broadband Access Networks）。BISmark团队在全球超过30个国家部署了测量路由器。他们的研究结果显示接入延迟（Last-mile latency）在整个互联网的延迟中占有很大比例，而接入延迟增加十几毫秒就可能导致网页加载时间增加几百毫秒。他们的实验结果同时显示通过对DNS和TCP连接进行缓存可以有效介绍网页加载时间。

传统研究Web用户体验的方法主要依赖于在客户端安装软件记录用户的信息。WProf（Demystifying Page Load Performance with WProf）是一个植入于浏览器中的插件，目标是分析网页加载时间的不同组成部分。通过监测浏览器的行为，WProf绘制了访问一个网页时网页对象加载、HTML解析、JavaScript与CSS评估和DOM解析的依赖关系图，并且绘制了相应时间的关键路径。关键路径上的节点是构成网页加载时间的不可缺少的部分。研究结果显示网页中的计算部分占据了关键路径时间的大约35%，同时对JavaScript的异步调用以为会阻塞HTML的解析也会显著增加网页的加载时间。

Athula Balachandran（Developing a Predictive Model of Quality of Experience for Internet Video）首次引入机器学习的方法分析网络性能参数和观看视频的用户体验的关系。因为网络参数之间的关联性很强，传统的分析单个网络参数与用户体验的关系难以找到根本的影响因素。他们考虑了视频类型、视频缓冲比、视频缓冲次数、视频平均传输编码率等因素，与用户体验指标——用户观看视频的持续时间进行关联，利用决策树构建了关系模型，并且提出了普遍的用于预测用户体验的框架。

也有工作研究了其他网络类型中网络性能参数与Web用户体验的关系。AT&T团队构建了蜂窝网络中网络参数与Web用户体验的模型。他们在基站上分别采集网络和用户浏览两方面的信息。前者包括基站信号强度、服务用户个数、上下行带宽和不同频率间的切换次数的因素。后者包括网页请求的URL和用户设备的MAC地址等。利用机器学习方法，他们根据信息计算出用户浏览网页的长度、放弃网页的次数和网页未完全加载百分比作为衡量用户体验的标准。在此基础上他们用机器学习的方法构建网络性能参数与用户体验标准的关系。最终，他们发现用户设备在不同频率间的切换对于用户体验有最大的影响，因为频率间的切换会导致网络中断长达几秒钟的时间，接下来影响较大的因素是信噪比。与预期相反，上下行带宽对用户体验影响不大，可能原因是网页浏览对带宽的需求较低。

主要工作

本文的工作主要分为以下几个部分：

1. 实现基于OpenWRT路由器的测量工具：

在路由器上抓取网络相关参数

为了构建网络性能与用户体验的关系，首先需要获得在用户使用无线网络时的路由器参数，包括时间戳、信道占空比、上下行带宽、物理层上下行速率、MAC层数据包重传比例、路由器当前连接设备数和每个设备的信号强度等。获得的数据传输到服务器上用于进一步分析。

在路由器上抓取用户访问信息

目前的Web应用基本上基于HTTP完成。为了了解用户的网页浏览状态，需要在路由器上抓取HTTP数据包记录。具体包括HTTP中GET类型数据包的GET、Host、User-Agent和Referer域，以及设备的MAC地址和时间戳。获得的数据同样传输到服务器上用于进一步分析。

1. 实现用户体验的分析工具

构建网站的HTTP请求的下载链

路由器上抓取的HTTP的GET数据包不仅来源于用户的网页浏览，也可能来源于其他Web应用。为了研究用户的网页浏览体验，首先要过滤掉其他来源的数据包。因为在路由器上抓取的记录是以数据包为单位的，在过滤掉无关数据之后，还需要在服务器上利用抓取的数据包的关系，恢复出用户对网站的完整访问流程。

区分用户请求与网页嵌入对象的自动请求

网页浏览的HTTP的GET请求来源分为两种。一种来自于用户点击一个超链接或者在浏览器中输入一个网址导致的主动请求。另一种来源于浏览器解析网页时，针对嵌入的对象如图片、JavaScript和CSS等自动发出的请求。前者与用户网页浏览时的投入程度关联紧密即与用户体验直接相关。后者主要用于呈现网页的内容，并不直接反应用户体验。因此，需要在请求中对两者进行区分。

构建网络参数与用户体验的关系模型

在区分出用户请求之后，将其作为衡量用户体验的标准。把用户请求与对应的Wi-Fi网络参数匹配，分析网络参数对其不同的影响程度。在此基础上挑选出具有主要影响力的网络参数用于构建关系模型，并且基于模型提出提高用户体验的建议。

1. 为了能够将带有测量工具的路由器分发开展实际测量，首先需要完成测量数据与服务器间的传输工作。然后，需要控制测量工具的计算开销。在实际部署测量之后，需要根据收集到的数据进行分析，进一步完善测量程序。

论文结构

本文第一章引言介绍了相应的研究背景、已有的工作和文章的主要工作内容。

本文第二章主要介绍不同的测量Web用户体验的方法。

本文第三章主要介绍系统的架构及实验的部署。

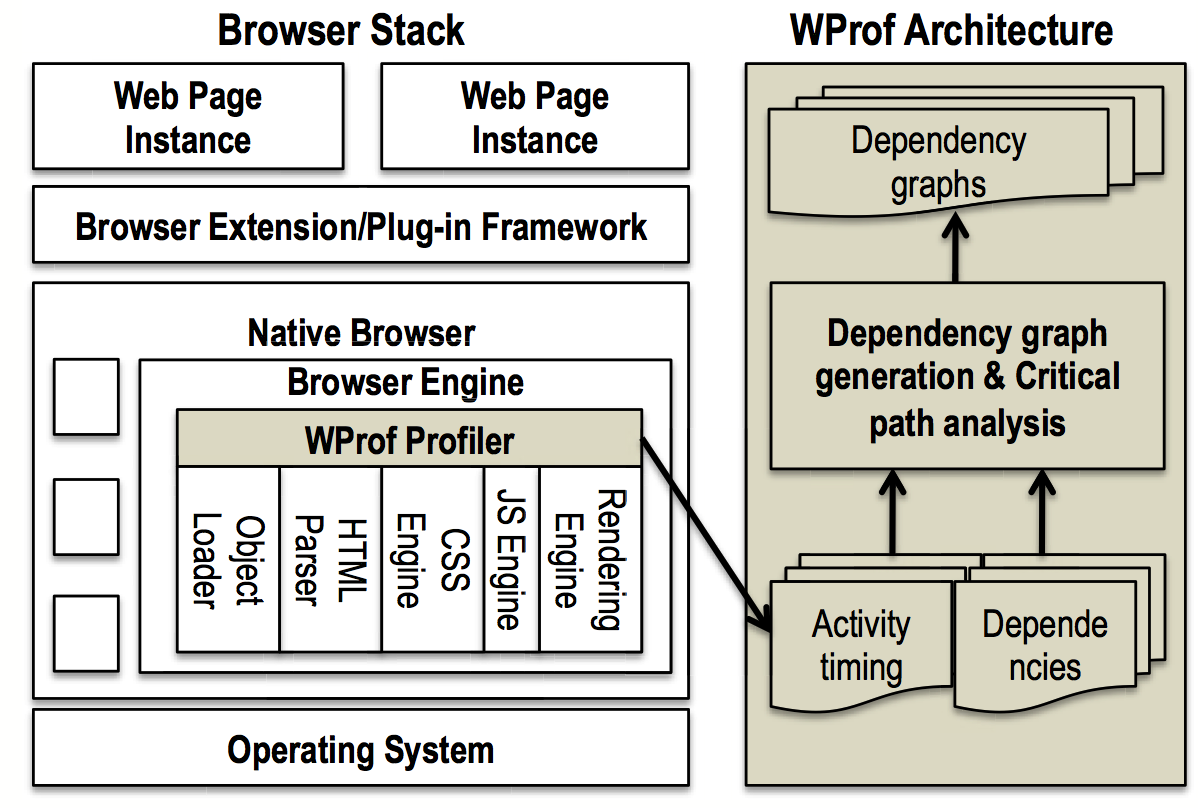
本文第四章主要介绍模型的构建以及数据的分析。

本文第五章对文章进行总结，探讨实验中遇到的问题以及未来的后续工作。

第2章Web用户体验的测量

本文的主要目的是构建Wi-Fi网络性能与Web用户体验的关系模型，准确有效地测量Web用户体验是构建模型的前提与基础。测量的方式也决定了需要收集的数据和实验框架的构建。

本章节首先根据测量方式的分类具体介绍了几种不同的测量方式，然后详细介绍了实际采用的一种适于大规模部署的基于路由器被动测量的方法。

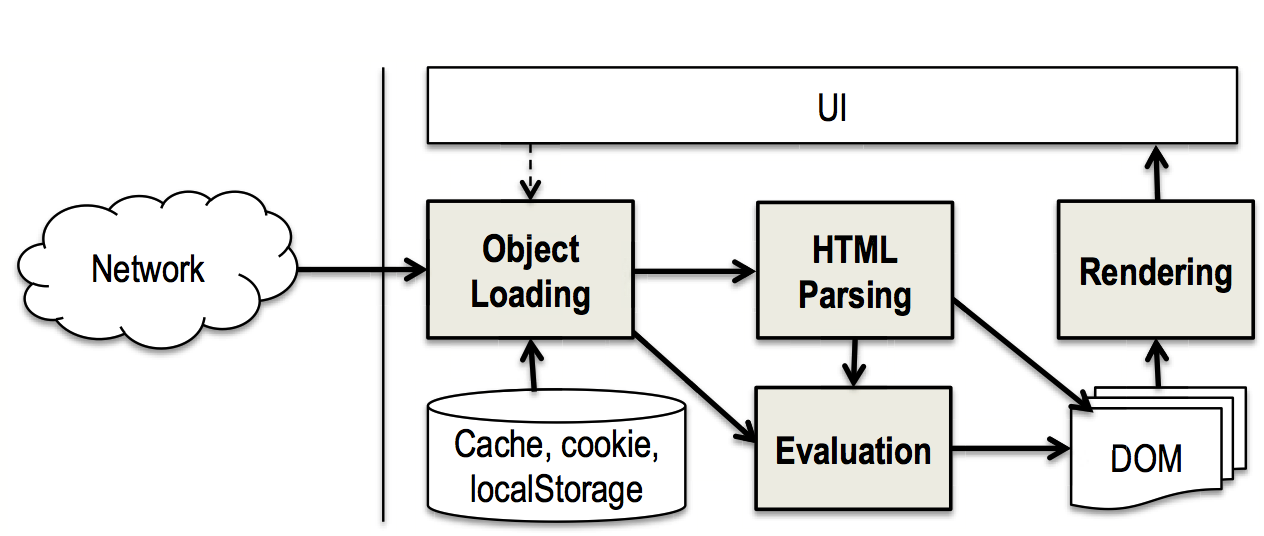


主动实验者实验

一些关于用户体验的实验邀请实验者到实验室中完成规定的任务，通过实验后的问卷调查了解用户体验。DF Galletta请196名学生在实验室环境下完成对三个网站的搜索任务。三个网站保存在光盘上，用以严格控制访问时间。在用户完成访问不同的访问任务时，电脑加入人工设置的延迟时间模拟不同网络性能。在用户完成实验后，他们被要求填写问卷评估自己完成任务的目的性和感受。主动实验者实验的优点在于调查问卷获得的用户反馈最真实直接地反应用户体验，是最准确的标准。然而这种实验方式缺点也非常明显，实验需要在实验室环境中才能完成，因此地点受到限制，导致被试者的普遍性受到限制。除此之外，每一次的实验代价非常大，持续时间却不能太长，不可能进行大规模实验。

客户端上的测量

目前常用的一种测量方法是在用户的设备即客户端上安装专门的程序捕捉用户信息或者用户与设备直接的交互情况以此来推测用户体验。Xiao Sopiha Wang将WProf，一个网页加载分析插件，植入于不同种类的浏览器的引擎中。该插件可以将浏览器加载网页的每一个事件记录下来并根据发生时间和相互影响构建它们的依赖关系。在客户端上进行测量的优势是可以获得客户端上独有的数据，在该种情形下比如网页的渲染时间和浏览器对JavaScript的计算时间。这些因素构成了网页加载时间的重要部分，直接影响着用户体验，而且这些数据只能在客户端被获取。在客户端进行测量的最大的缺陷是需要让用户额外安装测量软件，这种不便使得该种测量方法难以被大规模部署，也就难以适用于广泛地获得大量数据进行实验。另外一个缺点是在用户安装测量软件后，用户对隐私被泄露的担心可能改变他们使用网络的习惯，不能最真实地反应实际用户使用情况。



服务器端的测量

中间节点的测量

被动测量

基于中间节点的测量

鉴于本文探讨的是构建Wi-Fi环境下网络性能与Web用户体验的关系，我们将中间节点限制在普通类型的家用路由器上。路由器作为室内无线上网的主要接口，通常是室内无线上网所有流量经过的通道。同时，路由器作为接入互联网的最后一跳，它与用户设备直接相连，在路由器上收集数据使得我们能够将不同的记录针对不同设备或用户区分开，给后续研究奠定基础。从路由器之后这些数据包被汇聚在一起，弱化了每个用户独有的特征。因此，路由器是部署Wi-Fi性能测量工具的最佳选择。在路由器上，采集到的Wi-Fi参数主要分为两类。一类是路由器的总的工作状态记录，例如路由器累计的工作时间、累计传输数据的时间、累计接受数据的时间等。另一类数据是与单个设备相关的，包括当前连接的设备个数、设备MAC地址、每个设备传输的数据量、每个设备接受的数据量和每个设备的信号强度等。

路由器虽然是理想的测量Wi-Fi性能参数的地点，在测量用户体验时却不够直接。因为用户体验是用户层面的一种衡量标准，更加宏观，而路由器上只能获得包级别的底层数据。这就需要一个机制能够从底层的数据包记录中恢复用户体验的相关数据。

我们选择的衡量Web用户体验的指标是会话长度（Session Length）（引用！！！）。会话长度指用户进入一个网站后，从该网站及相关链接中手动打开的网页个数。会话长度表示了用户的投入程度。当网络性能良好时，网页打开的过程非常流畅，用户体验良好，因此会使用网页直到达到最初的目标。当网络性能不好时，网页的加载过程可能非常缓慢，有可能出现网页部分加载失败甚至整个网页无法加载的情况，这时用户的浏览体验不好就会关闭网页，导致会话长度很短。因此会话长度可以有效衡量Web用户体验。

接下来需要利用包级别的数据推测会话长度。用户手动打开网页反映在数据包记录中就是一个针对网页地址的HTTP的GET请求。然而路由器上记录的HTTP的GET请求并不仅仅来自于用户手动点击网页的访问。当浏览器解析网页的HTML文件时，会对其中的嵌入对象比如图片、JavaScript和CSS发出HTTP的GET请求，下载内容完成网页的加载，这些请求占据了GET请求的一大部分。现在主要问题转变为如何区分用户主动点击发出的请求和网页嵌入对象自动发出的请求。一种传统做法是（引用！！！）利用请求的间隔时间（idle time）。这种方法做出了以下假设，因为对嵌入对象的请求是在浏览器解析网页时自动发出的，时间间隔非常短，请求一般都在网页加载时发出。相反，用户的点击表示打开新的页面，一般在浏览原有网页之后，相隔时间较长。因此，通过对比两次相邻链接之间的时间差，如果大于预定阈值就可以认为这个请求是用户点击产生的，反之就是浏览器发出的。

然而，如今网页的复杂度逐渐增加，需要动态加载的内容越来越多，加载时间的跨度也逐渐增加，利用一个固定的阈值已经难以区分网页嵌入对象自动发出的请求与用户点击的请求。本文采用了一种更加有效的基于文本分类的用户点击识别方法。观察从一个网站中发出的所有请求，可以发现请求的URL有明显的区分。以新浪网为例，www.sina.com.cn和sports.sina.com.cn很可能是用户点击的网站链接。而对sax.sina.com.cn或d1.sina.com.cn的请求很大可能性来源于对嵌入对象的请求。基于上述特征，文本分类的方法可以应用于区分用户点击去嵌入对象。我们分解URL的中间部分以及文件类型作为特征。以下面这个URL为例http://www.cs.tsinghua.edu.cn/publish/cs/index.html，特征集为{cs, tsinghua, edu, cn, publish, html}。分类方法上，我们选用了在文本分类中效果很好的朴素贝叶斯方法。为了减少人工标注训练集的工作量，我们采用了一种自动标注的方法。将从一个网站开始的全部请求按时间排序，取出前10秒的部分。如前所述，因为人在每次手动点击之间的时间是浏览网页，相对较长，我们可以认为10秒这个区间内只有第一次请求为用户点击发出的，其他请求都为浏览器自动发出的。我们可以使用这样的方法自动生成训练集。获得足够的训练集之后，用这些训练集训练朴素贝叶斯模型，然后用训练后的模型对所有请求数据进行处理，获得仅包含用户点击的列表。

综上所述，为了保证测量实验可以被普遍的部署而不限制在具体的实验环境中，我们决定采用放置在中间节点即家用路由器上的测量方法。通过朴素贝叶斯法对用户的访问记录进行分析，识别其中来自于用户点击的请求。对于一个网站的请求个数即该网站的会话长度，表示着用户访问该网站的体验程度。

（）（我也要分别写HTTP数据的获得和Wi-Fi参数的获得）（我也要写匹配算法，非常朴素）（）

第三章 系统的架构及实验部署

本文实现了一个部署在路由器上的轻量级的测量工具，分别记录路由器与设备之间的Wi-Fi网络性能以及与用户访问网页相关的记录。除此之外，本文还在服务器端实现了一串工具管道完成两者之间关系模型的分析与构建。本章将具体介绍系统的架构设计以及具体实现，描述系统捕捉数据的规格以及功能实现的算法。

3.1系统架构图

整个系统主要由三个部分组成，分别是：路由器端数据包的抓取、服务器端数据的分析和结果的输出。首先，我们在路由器端的无线网卡上抓取数据包，然后根据需求对数据包进行处理，保留有用的部分。每隔一段时间，系统会自动地把保留下的数据包记录发送到服务器端。最后，我们在服务器端将这些数据汇聚起来。先通过算法将用户请求重建起来，然后通过匹配算法将用户访问记录与Wi-Fi网络参数对应起来，进而分析两者之间的关系。

系统流程如图所示（我要图片！！）

【对图片中的内容做一些说明】

3.2数据的获取

3.2.1用户访问相关数据的抓取

在本系统中，我们利用libpcap（引用？）在wlan接口上抓包。wlan为路由器的无线网卡接口。因为我们研究的对象是Wi-Fi网络性能，在wlan上抓包可以避免抓到来自通过网线与路由器直接连接的设备的数据包，避免了对实验结果的影响。用户的Web访问是基于HTTP的GET请求，因此HTTP的GET请求就是我们的研究对象，而HTTP是基于TCP的。所以我们利用libpcap对每个TCP数据包的负载进行检查，如果以GET开头，则说明该TCP数据包实际是HTTP的GET请求。TCP的帧结构如下图所示（图！！！）HTTP的GET内容出于TCP包的负载部分，结构如下（图！！！）。当我们发现一个TCP包的内容是HTTP的GET时，我们利用正则表达式匹配，将GET请求中的GET、Host、User-Agent和Referer四个域提取出来，每个域最多保留256字节，保留在我们的记录结构中。同时，我们还会保留TCP帧中的源IP地址、目的IP地址、源MAC地址和目的MAC地址，并且打上系统的时间戳。整个记录的格式如下所示（图！！！）

GET域是请求资源的地址，不包括域名。Host域为GET请求资源的主机名称。User-Agent代表GET请求的发出者。Referer表示当前GET请求的来源地址。

3.2.2Wi-Fi网络参数的抓取

xxxx

3.2.3网站请求链的重建

为了分析用户访问网站的用户体验，我们需要解决如下几个问题。首先，在路由器上抓取的数据是包级别的，而用户体验访问的会话是宏观的，需要从包级别的数据中重建出宏观的用户对网站的访问。其次，一个路由器上可能同时连接了多个设备，我们需要对每个设备进行区分才能准确匹配相应的网络参数。最后，即使一个路由器上只连接了一个设备，用户可能使用不同浏览器访问网络，或者用户电脑上其他Web应用发出的请求也会对分析造成影响。这里，我们提出一个算法从包级别的数据恢复单一用户单一浏览器发出的请求过程。

func reconstruct\_visit\_record(lines)

for line in lines:

if User\_Agent is not valid:

continue

if Referer field is empty:

session = Session()

new\_ref = Host + GET

session.MAC = MAC

session.url\_list.append(new\_ref)

session.timestamp\_list.append(timestamp)

session\_list.append(session)

record\_dict[MAC][User\_Agent][new\_ref] = session

else:

ref = Referer[7:]

if ref not in record\_dict[MAC][User\_Agent]:

continue

session = record\_dict[MAC][User\_Agent][ref]

new\_ref = Host + GET

session.url\_list.append(new\_ref)

session.timestamp\_list.append(timestamp)

record\_dict[MAC][User\_Agent][new\_ref] = session

该算法的核心是依赖于GET的Referer域。如前所述，Referer标记着该请求的来源地址。用户在网站A中手动点击打开超链接发出的请求或者浏览器对网页嵌入对象的请求的Referer域都为网站A。利用Referer，我们就可以将单个的GET请求恢复成一个下载链。然而，由于我们需要达到单个用户单个浏览器的精度，还需要做一些额外的处理。该算法的核心数据结构是一个三级索引的字典，关键字分别为请求源MAC地址、User\_Agent和Referer地址，取得的值是一个用户访问会话。利用TCP包中的源MAC地址可以对连接在路由器上的设备进行区分。在保证记录来自于同一个设备后，User\_Agent可以保证请求从同一个浏览器发出，避免不同浏览器访问同一网站造成的混淆。最后Referer保证这些请求属于同一个会话。算法逐行读取包级别的GET请求记录。首先针对User\_Agent进行判断，如果请求发出者不是浏览器，那么我们就忽略该条记录。然后根据Referer域进行判断，如果Referer域为空，说明该请求将开始一个新的会话。我们创建一个新的会话类，把当前的记录写入会话类中。对于每一个会话，我们都需要知道哪些请求可能从它发出的，即Referer中的哪些地址属于这个会话。对于当前的新会话，我们把它的Host和GET两个域的字符串拼接起来，即得到该请求的完整地址。将完整地址结合TCP的源MAC地址和GET的User\_Agent三者作为关键字加入字典，则从当前网站发出的请求可以利用字典找到对应的会话。如果Referer不为空，则使用上述三个关键字在字典中寻找。如果找到会话，则把最新的请求的完整地址加入会话的下载记录中，并且指向目标会话。如果没有找到，原因可能是路由器没有完整记录所有的数据包，丢失了创建改会话的数据包，那么对于该种记录，我们不予考虑。

在将所有请求按会话归类存储在字典之后，程序会根据会话为单位遍历记录，按时间顺序输出一个会话的所有请求。输出格式如下（小表格，说明每个域的内容）timestamp，url。

时间戳是路由器收到该请求数据包时记录的时间，URL是该请求的资源地址。对于重建后的请求链使用在第二章中提到的方法进行分析即可得到用户的访问会话长度记录，我们以如下格式记录session length，time，site，request length，device MAC，AP MAC。Session length表示用户访问网站的会话长度。time里面包含这个会话的起止时间戳，单位为秒。site表示该会话初始访问的网站。request length表示用户恢复该会话的原始记录中包含的请求个数。Device MAC是用户设备的MAC地址。AP MAC是用户连接的路由器的MAC地址。

3.2.4数据的匹配

在恢复用户的访问会话之后，需要把用户访问过程与该过程发生时的Wi-Fi参数匹配起来，从而分析两者之间的关系。用户请求的时间戳来自于路由器抓取到GET数据包时记录下的系统时间，而Wi-Fi参数则是固定地每10秒采集一次，两者不是精确匹配的。因此，我们决定对每次用户访问会话寻找能够覆盖它的Wi-Fi参数记录区间作为近似。用户会话的长度记录如3.2.3所示。对应的Wi-Fi记录分为两类，一类是路由器工作信道的数据，另一类是路由器与客户端设备时间的数据。前者的记录格式如下：Timestamp, Active, Busy, Receive Time, Transmit Time。Timestamp是记录该组数据的时间戳。Active是路由器从开机启动开始累计的时间。Busy是路由器从开机启动开始累计的繁忙时间，包括传输数据、接收数据以及等待其他造成干扰的路由器传输数据。Receive Time是路由器自启动之后累计接收各客户端数据的时间。Transmit Time是路由器自启动之后累计向客户端传输数据的时间。后者的记录格式如下：Timestamp，Device MAC，Receive Bytes，Transmit Bytes，Send Packets，Resend Packets，Signal Strength，Transmit Phyrate，Receive Phyrate。Timestamp是记录该组数据的时间戳。Device MAC是客户端设备的MAC地址。Receive Bytes是路由器自启动之后累计接收该客户端的字节数。Transmit Bytes是路由器累计给该客户端发送的字节数。Send Packets代表路由器累计给该客户端发送的包的个数。Resend Packets代表路由器累计给该客户端重传的包的个数。Signal Strength代表该客户端接受到的路由器的信号强度，单位为dB。Transmit Phyrate代表路由器给客户端发送数据包时物理层的发送速率。Receive Phyrate代表路由器物理层接受客户端发送的数据的速率。利用时间戳进行匹配的算法伪代码如下：

def find\_records(file\_name, start\_time, end\_time, device\_MAC, AP\_MAC):

file\_path = '../wifidata/%s/%s' % (AP\_MAC, file\_name)

f = open(file\_path, 'r')

lines = f.readlines()

f.close()

# start\_time and end\_time are string

start\_index = bisect(lines, start\_time)

end\_index = bisect(lines, end\_time)

return lines[start\_index - 1 : end\_index + 1]

因为路由器两类Wi-Fi参数记录的时间戳含义相同，因此可以使用相同的算法进行匹配。对于每一个用户会话记录，分别获得它的开始和终止时间戳。利用二分查找，在Wi-Fi参数记录中分别找到不超过起始和终止时间戳的最大时间。然后取这两个时间中的所有Wi-Fi记录作为与查询会话的匹配结果。

本章小结

本章对实验系统的架构进行了介绍，特别详细介绍了在路由器上的数据抓取部分，给出了相应的算法代码以及每个记录参数的具体含义。在路由器上抓取的数据是分析并构建模型的来源，也是整个实验的基石。