



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

基于 WIFI Fingerprint 的防作弊签到系统

计算机网络

CS339

卿云帆 516020910073

潘家琦 519030910243

李昕然 519030910121

摘 要:

本小组针对现有签到系统的不足,同时运用《计算机网络》所学相关知识,创新地开发出一套基于 WIFI Fingerprint 的防作弊签到系统。该系统主要通过收集学生端和教师端的 WIFI 列表信息,并进行相似度比对,进而判断签到成功与否。本小组力求该系统同时具备方便快捷与防作弊两种优势。

目 录

一、项目动机

二、实现过程

- 1、确定方案
- 2、确定工程框架
- 3、实现细节
- 4、测试表现

三、优势、不足与展望

四、人员分工、主要贡献、项目时间安排

五、总结与收获

六、致谢

一、项目动机

签到，作为学生没有缺席的一种证明，是考察学生上课态度的一种方式，也是平时成绩的重要组成部分。当前学校普遍采用的签到方式有扫码签到，签名签到，老师或助教点名签到等几种。其中，点名和签名签到较为耗时，扫码签到方便快捷但容易作弊，只要同学们扫到码，无论在教室还是宿舍，都能进行签到。总之，现有的签到方式普遍都无法同时满足方便快捷与防止作弊代签两方面的同时保证。

因此，本小组从实用性角度出发，同时结合了计算机网络相关知识，学以致用，致力于开发出一种既方便快捷，又能防止作弊代签的签到系统。在巩固计算机网络知识的同时，亦能服务于实际生活学习，同时体会到小组共同开发研究的乐趣。

二、实现过程

1. 确定方案

为了实现这样一个签到系统，我们小组的第一步是思考一个合理的方案。在实际过程中，我们思考过如下几种不同的方案：

1.1 定位签到法（否决）

利用手机 GPS 定位来确定学生的位置，调用该位置信息，匹配对应的教室或老师位置，检测是否一致，从而判定学生是否到场上课。

方案问题：定位能够被较轻松地模拟。上网搜索模拟定位软件，可以轻松地搜索到各种各样的模拟定位软件，那么学生就可以简单地通过下载现有的模拟定位软件，修改目前的位置信息，从而达到作弊签到的目的，且付出的代价相对较低。那么该系统的防作弊目的就难以实现。

1.2 局域网通信法（否决）

建立学生端和教师端，利用同一个 WIFI 下的局域网通信，自动在学生端与教师端之间发送信息，从而确认学生与老师在同一个 WIFI 下，进而间接证明学生已经在教室上课。

方案问题：同一个 WIFI 下的局域网通信在通常情况下很有效，但是在学校的网络环境下，通过测试我们发现，设备获得的动态 IP 地址具有一定的持久性，在一定范围内，不随地理位置变化而变化，因此无法用来确定通信双方在相近的地点，也就没法用来确定学生是否身处教室以实现签到。

1.3 检测特定 WIFI 法（否决）

建立学生端和教师端，教师端自动开启热点，学生端通过检测 WIFI 列表，寻找能否检测到此 WIFI，从而判断学生是否在教室。其实这种办法与一些公司的 WIFI 打卡有些类似。

方案问题：，可以找到“WIFI 模拟器”等软件，该类软件能够模拟出手机“已连上”某个自定义 WIFI，用户输入 WIFI 名称，SSID 信息，即可模拟手机连上该 WIFI。因此，学生也可以在不花费太多努力的情况下，通过这类软件进行作弊签到。另外，教师端自动开启热点时，会中断教师端手机的 WIFI 连接，在某些情况下，会造成网络环境的变化，从而导致一些麻烦，

这与方便快捷又有所违背。所以该方案也被否决。

1.4 WIFI Fingerprint 比照法（最后采纳选用）

建立学生端和教师端，在签到时，收集教师端检测到的 WIFI Fingerprint 和学生端检测到的 WIFI Fingerprint 信息，通过比较双方 WIFI 的 Fingerprint 信息，做出一个相似度判断。根据相似度判断，来决定学生和老师是否位于同一间教室或同一处地方，从而决定签到成功与否。同时，我们不需要做到精确的室内定位，只需达到大致一间教室的精确度即可，这一一定程度上提高了项目鲁棒性。

由于目前搜索到的现有的各种 WIFI 模拟器并没有提供同时模拟大量 WIFI 的功能，同时也不提供 WIFI 强度的模拟，因此在这方面是难以作弊的。另外，其他防作弊的手段后续也会提到。

2. 确定工程框架

2.1 采用前后端模型：

搭建前端后端，最后做成较完整，可实际使用的工程项目。

起初，前端我们想通过微信小程序来实现。主要原因是微信小程序具有较强的可移植性，同时有较好的 API 接口支持，对于没有编写软件基础的我们来说会相对容易，好上手，代码量也会比较小一点。

但后续该计划被否决，原因是微信小程序有许多的限制：

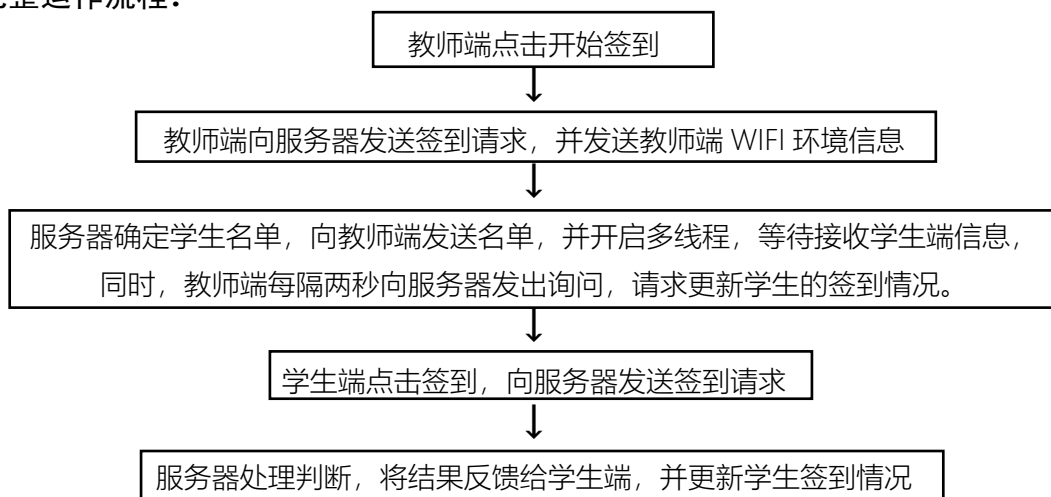
1、无法很自由地获取各种 WIFI 列表相关的信息，只能通过调用一个函数获取固定格式固定内容的信息，不能很好满足我们的需求，在获取手机唯一标识码的方面，也表现得无能为力。

2、在与服务器的通信方面，使用微信小程序通过 UDP 或 TCP 协议进行通信时，必须使用已备案的域名，对于课程项目，可能显得代价过高。

因此，我们最后决定直接编写 Android APP 作为前端实现，包括学生端和教师端。

而后端在云服务器搭建服务器。处理前端发送来的数据，进行校验码判断以及 WIFI 相似度比对，得出结果，并反馈给前端（教师端）。

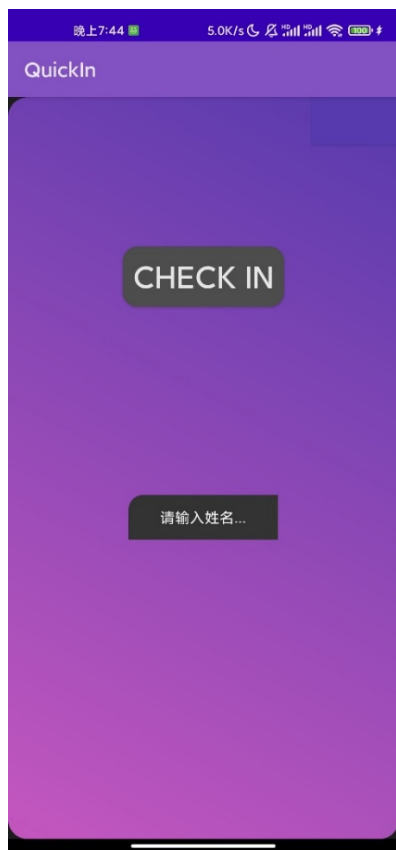
工程完整运作流程：



3. 细节

3.1 前端：

3.1.1 APP 界面展示：



(图一)



(图二)

3.1.2 功能说明以及运作流程：

图一，是学生端界面。请输入姓名处由学生输入姓名。后续考虑关联 jaccount 账号功能，即在登录后直接关联学生信息，而无需进行额外的输入。

点击 CHECK IN 进行签到，自动向服务器发送学生端的 WIFI Fingerprint 信息。同时，调取该手机唯一标识码信息，并对标识码加密，将加密后的结果和其它信息一起发送至服务器，

关于标识码及加密细节将在后文 “防作弊方面的细化” 中进行详细说明。服务器接收并做出判断后会返回相关信息，提示学生签到成功与否。

图二，是教师端界面。长按开始签到即可向服务器发送签到请求，同时自动发送教师端的 WIFI Fingerprint 信息。服务器会实时更新学生的签到情况，教师端每隔两秒向服务器发出询问，请求回复，更新教师端学生的签到情况。这里采取教师端定时询问服务器获得回复的方式，是考虑到对教师端网络状态可能不具有公网 IP，很难由服务器主动唤起 UDP 通信，向教师端发送信息。

教师端可以通过点击 “已签到”，“存疑”，“未成功签到” 以更换页面，来查看学生的实时签到情况。同时在存疑签到和未成功签到界面，教师可通过手动勾选的方式进行学生签到信息的更新。

3.2 后端:

我们使用原生 UDP 建立三次握手, 首先由教师端发送 invoke 指令开启一个签到, 服务器返回确认并附加学生名单以节省握手次数, 教师端收到 ack 后发送 Wifi Fingerprint。服务器接收后会创建多个学生线程与一个监听线程, 并维持共计十分钟, 在 10 分钟后都将强制关闭, 以限制允许签到时间为 10 分钟。

学生线程多并发地接收学生签到信息, 这里包含的信息有学生姓名、设备唯一标识码与 WIFI Fingerprint。首先服务器会搜索学生姓名, 失败则返回姓名错误, 成功则查看服务器本地存储的该同学的唯一标识码。

唯一标识码的写入由学生首次签到确认并进行持久化存储。后续签到中, 只需对比本地和此次接收标识码, 并且和其它标识码进行重复校验, 如比对和校验通过则说明该设备是合法的, 继续进行下一步校验; 若比对失败, 则说明此时签到使用非常用设备, 认为签到存疑, 直接向教师端报告存疑签到, 以等待教师手动进行更新处理。WIFI 列表比较函数有三种返回值 0, -1, 1:

- a. 1 表示定位地点与教师在同一教室内, 签到成功;
- b. 0 表示定位地点位于教室附近, 属于存疑情况;
- c. -1 表示定位地点距离教室较远, 签到失败。

监听线程总是在等待接收教师端的询问申请, 一旦收到询问则向其发送当前签到成功与签到存疑的学生名单。

3.3 WIFI 相似度比对原理:

主要根据检测到的 WIFI Fingerprint 中, SSID 和信号强度两个维度的进行相似度判断。

设置匹配度, 根据 SSID 匹配度以及强度的匹配度进行匹配度的调整, 通过一系列匹配度判断过程, 得到最终的匹配度得分, 若得分超过一定值, 则认为成功签到, 若得分低于一定值, 则认为签到失败, 若得分介于这两者之间, 则认为签到存疑。这些结果在教师端均可查看到。对模糊范围的设定也进一步增加了项目的鲁棒性。

其中, 具体参数的确认是根据小组的实地进行测试, 通过人工的“反向误差传播”进行调整。调整结束后, 对 206 个数据点进行了共 824 组的测试, 最后的调整结果保证了在测试集上有 99.5%左右的准确率, 也表现出了较好的泛化性。

具体的得分判断过程是人工建立的决策树, 具体方式如下:

为了选取可靠有效的数据, 我们先根据强度从小到大排序, 然后选择强度小于等于 65 的 WiFi 来比对, 但考虑到某些情况下, 强度小于等于 65 的 WiFi 数目可能太少, 我们最终选取 max (小于等于 65 的 WiFi 数目, 13, 最大 WiFi 数量)。可参考如下代码。

```

int r1=0,r2=0;

while(s1[r1].s!="--" || s1[r1].q!=0)r1++;
while(s2[r2].s!="--" || s2[r2].q!=0)r2++;
r1--;r2--;
int r11=r1,r22=r2;

sort(s1,s1+r1+1,cmp);
sort(s2,s2+r2+1,cmp);

r1=0;r2=0;
while(s1[r1].q<=65&&(s1[r1].s!="--" || s1[r1].q!=0))r1++;
while(s2[r2].q<=65&&(s2[r2].s!="--" || s2[r2].q!=0))r2++;
r1--;r2--;

if(r1<13)r1=min(13,r11);
if(r2<13)r2=min(13,r22);

```

得到需要比对的 WiFi 之后，我们分别对教师端和学生端进行两轮循环，若匹配到相同的 WiFi ssid，则对比两者间的强度差值，根据强度差值加不同得分，最后再除以数目，得到平均值，可参考如下代码，

```

for(i=0;i<=r1;i++){
    for(int j=0;j<=r22;j++){
        if(s1[i].s==s2[j].s){
            if(abs(s1[i].q-s2[j].q)<4)score+=300;
            else if(abs(s1[i].q-s2[j].q)<15)score+=150;
            else if(abs(s1[i].q-s2[j].q)<20)score+=100;
            else score+=30;
            break;
        }
    }
}
score/=max((r1+1),1);

double score2=0;
for(i=0;i<=r2;i++){
    for(int j=0;j<=r11;j++){
        if(s2[i].s==s1[j].s){
            if(abs(s2[i].q-s1[j].q)<4)score+=300;
            else if(abs(s2[i].q-s1[j].q)<15)score+=150;
            else if(abs(s2[i].q-s1[j].q)<20)score+=100;
            else score+=30;
            break;
        }
    }
}
score2/=max(1,(r2+1));

score+=score2;

```

最后进行范围判断

```
int ans;  
if(score>=600)ans=1;  
if(score>=200&&score<600)ans=0;  
if(score<200)ans=-1;  
return ans;
```

3.4 防作弊方面的细化：

在上文，我们已经部分说明了想要通过 WIFI Fingerprint 的方式来进行作弊是较为困难的，有较高的作弊门槛。同时，我们对其它可能作弊方式也进行了综合考虑：

- 1) 针对 “代签” 作弊，即使用同一部设备进行多次不同用户签到。

我们做出的防作弊措施是，实施“唯一标识码”匹配。利用安卓提供的 UUID 相关组件，在学生端中生成并持久化存储设备唯一标识码，在签到时将标识码经过加密发送给服务器，服务器进行接收码与本地信息的比对，从而实现 “1 部手机，1 名学生，1 次成功签到” 的一一对应。

并且，安卓提供的 UUID 在 APP 被卸载后仍然存在，只能通过恢复出厂设置来进行删除，具有较好的持久性。

- 2) 针对 “抓包” 的作弊方式，即去教室进行签到一次并进行抓包，再后续持续发送该抓包数据，以完成作弊签到。

考虑到 wifi 的强度变化是难以确认的，因此可能出现某段时间的 wifi 强度信息较为稳定，因此做了额外的加密，以时间作为密钥，加密唯一标识码，将其作为密文进行传输，以进行校验。

学生端调取当前的日期作为密钥，设计了一个加密和解密算法，主要是截取唯一标识码的片段，并利用月份日期的四位数字进行一一对应的加密，通过对截取信息每一位的 ASCII 编码进行后移操作，即可得到新的四位标识码，从而实现简单高效的加密。

服务器收到后再根据当前的时间信息进行解密，还原标识码并进行匹配判断，这里采取了粗粒度的时间信息，保证了信息有较长的有效时间。

这样一来，以抓包重发方式作弊也会导致唯一标识码的不匹配，从而导致签到失败，并提醒老师该名同学很可能存在问题，需要进行额外的确认。

需要强调的是，我们的加密并不旨在增强信息的安全性，而是为了给 Fingerprint 设定有效时间限制，从而提高作弊的门槛。

4. 测试表现

我们总共收集了 206 个数据点，有四组教室的数据，每组的 30 个点。另外，在教室周围和其他地区随机收集有 80 个左右的数据点进行测试。

一共进行 4 轮测试。每轮测试时，选取测试教室讲台处收集的 WIFI 信号作为教师端，与其他所有数据点进行相似度比较，得出签到成功/存疑/失败的结果，并与实际情况进行对比。从而统计出准确率及其他效果表现。

测试数据集详见 TestResultData 文件夹中的 dataset.txt，几个教师的具体测试输出在对应的其它文件。

第一轮测试：东中 304 教室讲台作为教师端

教室内数据点数量:			教室外数据点数量:		
30			176		
实际测试中不同结果的数量:					
成功	存疑	失败	成功	存疑	失败
30	0	0	0	0	176

第二轮测试：下院 114 教室讲台作为教师端

教室内数据点数量:			教室外数据点数量:		
30			176		
实际测试中不同结果的数量:					
成功	存疑	失败	成功	存疑	失败
29	0	1	0	1	175

第三轮测试：东上 409 教室讲台作为教师端

教室内数据点数量:			教室外数据点数量:		
30			176		
实际测试中不同结果的数量:					
成功	存疑	失败	成功	存疑	失败
27	3	0	2	0	174

第四轮测试：东下 302 教室讲台作为教师端

教室内数据点数量:			教室外数据点数量:		
41			165		
实际测试中不同结果的数量:					
成功	存疑	失败	成功	存疑	失败
41	0	0	1	5	159

根据上述测试结果，4 轮总共测试了 824 组对比数据点，准确率达到 99.5%（不计存疑组）存疑率为 1.1%。可以看到，测试结果以及满足签到系统的基本准确率要求。

三、优势、不足与展望

优点：

1. 实现了一个较为完整、能实际运用的工程项目。在基本知识的基础上，进行了工程化的应用。开发了相关 app，实现了前端后端的搭建，基本达到实际使用要求。
2. 创新的使用了 WIFI 信号强度作为判断依据。

现有的常见打卡签到方案，很少有用到信号强度作为判断依据，我们在项目中使用了信号强度作为判断依据，在防作弊程度上能更高的门槛。

3. 实现了简单但较为完整的网络通信方案。

在这个项目中，前端后端需要频繁的网络通信。我们基于原生 udp 协议加入停等等方式实现了前后端简单稳定的通信。巩固了在课程上所学的知识，并且学以致用，加深了对网络通信的理解与认识。

不足与展望：

1. 在 WIFI 相似度判断中，目前使用的还是人工来决定判断依据，后续考虑使用机器学习一类的算法，采用诸如神经网络等算法，并采集到更多维度的数据，可能会在效果上有较好的提升。
2. 移动端操作系统对一些敏感行为会有限制。在此项目的扫描 WIFI 列表这一操作中，并不是每次扫描都能获取到当前实时的 WIFI 列表信息，会有几十秒的限制，若两次扫描间隔时间很短，会无法获取最新的列表信息。也是基于此，我们设定了存疑签到和教师端手动确认以解决类似的特殊情况。
3. 除开学校的环境，该签到系统在商场、公司等其他情况下，亦可以进行进一步地扩展应用。当然，虽然从原理上来说是可以推广的，但若要进行进一步拓展，需要更多地测试来验证。
4. 服务器端采用 C++这类偏于系统的语言可能过于繁琐，尤其是在字符编码等问题上，C++对项目造成了较多的困扰，后续可考虑采取 Python 或 Golang 等支持较好的语言进行实现。

四、人员分工、主要贡献、项目时间安排

卿云帆：

1. 完成前端主体开发，学生端、教师端的 app 代码编写。
2. 掌控好项目开发的进度，进行项目管理。
3. 制作汇报 ppt。

潘家琦：

1. 完成后端主体开发，包括通信框架和整体的签到逻辑。
2. 处理和签到信息相关的文件流系统。
3. 实现系统测试和服务器维护。

李昕然：

1. 实现 WIFI 列表相似度比对程序
2. 实现对标识码加密解密程序
3. 撰写项目报告

项目时间安排：

第 2、3 周：确定项目方向

第 4、5 周：思考并确定方案

第 6、7、8、9 周：一步步实现工程。先搭建服务器，再做好分工，同时进行前端、后端、核心代码的开发

第 10、11、12 周：测试效果，并修改完善

第 13 周：撰写报告、制作 ppt、最终调试

五、总结与收获

《计算机网络》的课程安排较为合理，给予了我们较高的自由度和较为充分的时间去一步步构思和实现项目。能够在实践中巩固课程上学习到的基础理论知识，也能去尝试不同的方案，允许我们有一定的试错成本，从而更好的推进项目。

我们小组基本上从第三周就开始了大作业的初步工作，约定一每周组会的形式进行汇报，这保证了每个小组成员在每一周都能有所进展。这样的方式保证了项目进展能够稳定的推进，从而能够实现一个较大代码量的工程项目，也在最后实现了较为满意的结果。

除了技术、知识层面的收获外，我们还有一些其他方面的感想，主要是开展项目时的注意点：

1. 前期的方案构思是相当重要的。正所谓“磨刀不误砍柴工”，方案构思很大程度上决定了整体项目的效果和实现的难易程度，多花一些时间在方案的构思上，往往能使得后面的工作事半功倍，减少试错成本，这方面也格外感谢孔老师对我们相关问题的详细解答，也为我们提供了很好的方向指导。
2. 明确而合理的分工相当重要。适当的对项目进行解耦，从而明确具体任务，以便于规划方案，同时每个成员有自己负责的部分，能够较快的进行调试和修改代码，使进展更加有效率。
3. 明确核心任务和任务顺序很重要。在一个工程项目中，或是在一个项目的每个小阶段中，都会有一些核心的任务，分配较多的时间精力优先解决核心任务，可以更有效率的推进进度，同时也能提高整个项目小组成员的信心。另外，需要注意的是任务之间的制约顺序，对于有些牵制到其他任务的前置工作，要尽快地，高效地去解决，以便更好地推进项目进展。

六、致谢

感谢孔令和老师和助教对项目全程提供的支持和帮助，以及对计算机网络基础知识的指导和答疑。