本文作者：牛国柱
Blog：http://iamniu.com 微博：http://weibo.com/niuguozhu

互联网精准广告定向技术

一切你该了解的知识总结与整理

原文作者为资深互联网广告行业从业者，经过一段长时间的积累，从基础知识、技术应用以及在线广告作弊手段整理和反作弊方法三大角度，共计 16 篇系列专文，对于精准广告定向技术的介绍进行了一个全面且切实的总结与全景介绍。

以下为原文：

截止今天，关于精准广告定向技术的介绍已经全部写完。介绍的写作初衷是总结自己的知识，将知识从片段的、隐形的转化为可以向别人讲述、能够给人帮助的。在总结的过程中自己也提升了很多，同时希望这些内容能够切实的给刚进入这个行业的同学们以帮助。

一、基础知识篇：

Http Header 之 User-Agent

User Agent 中文名为用户代理，是 Http 协议中的一部分，属于头域的组成部分，User Agent 也简称 UA。它是一个特殊字符串头，是一种向访问网站提供你所使用的浏览器类型及版本、操作系统及版本、浏览器内核、等信息的标识。通过这个标识，用户所访问的网站可以显示不同的排版从而为用户提供更好的体验或者进行信息统计；例如用手机访问谷歌和电脑访问是不一样的，这些是谷歌根据访问者的 UA 来判断的。UA 可以进行伪装。

浏览器的 UA 字串的标准格式：浏览器标识 (操作系统标识; 加密等级标识; 浏览器语言) 渲染引擎标识版本信息。但各个浏览器有所不同。

字串说明：

1、浏览器标识
出于兼容及推广等目的，很多浏览器的标识相同，因此浏览器标识并不能说明浏览器的真实版本，真实版本信息在 UA 字串尾部可以找到。

2、操作系统标识

平台	标识	备注
FreeBSD	X11; FreeBSD (version no.) i386	
	X11; FreeBSD (version no.) AMD64	
Linux	X11; Linux ppc	
	X11; Linux ppc64	
	X11; Linux i686	
	X11; Linux x86_64	
Mac	Macintosh; PPC Mac OS X	
	Macintosh; Intel Mac OS X	
Solaris	X11; SunOS i86pc	
	X11; SunOS sun4u	
Windows	Windows NT 6.1	对应操作系统 Windows 7
	Windows NT 6.0	对应操作系统 Windows vista
	Windows NT 5.2	对应操作系统 Windows 2003
	Windows NT 5.1	对应操作系统 Windows xp
	Windows NT 5.0	对应操作系统 Windows 2000
	Windows ME	
	Windows 98	

3、加密等级标识

N: 表示无安全加密

I: 表示弱安全加密

U: 表示强安全加密

4、浏览器语言

在首选项 > 常规 > 语言中指定的语言

5、渲染引擎

显示浏览器使用的主流渲染引擎有：Gecko、WebKit、KHTML、Presto、Trident、Tasman 等，格式为：渲染引擎/版本信息

6、版本信息

显示浏览器的真实版本信息，格式为：浏览器/版本信息

注：

1、在广告定向设定中，浏览器定向和操作系统定向均是针对 User-Agent 中的信息进行定向。

2、欲了解更多的 User-Agent 信息，请参考 User-agent 字串史。

浏览器 User-Agent 的详细信息

PC 端：

safari 5.1 – MAC
User-Agent:Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us)
AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50

safari 5.1 – Windows
User-Agent:Mozilla/5.0 (Windows; U; Windows NT 6.1; en-us)
AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50

IE 9.0
User-Agent:Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;

IE 8.0
User-Agent:Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; Trident/4.0)

IE 7.0
User-Agent:Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0)

IE 6.0
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

Firefox 4.0.1 – MAC
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:2.0.1)
Gecko/20100101 Firefox/4.0.1

Firefox 4.0.1 – Windows
User-Agent:Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101
Firefox/4.0.1

Opera 11.11 – MAC
User-Agent:Opera/9.80 (Macintosh; Intel Mac OS X 10.6.8; U; en) Presto/2.8.131 Version/11.11

Opera 11.11 – Windows
User-Agent:Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131 Version/11.11

Chrome 17.0 – MAC
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.56 Safari/535.11

傲游（Maxthon）
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0)

腾讯 TT
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; TencentTraveler 4.0)

世界之窗（The World） 2.x
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)

世界之窗（The World） 3.x
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The World)

搜狗浏览器 1.x
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SE 2.X MetaSr 1.0; SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr 1.0)

360 浏览器
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; 360SE)
Avant
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Avant Browser)

Green Browser
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)

移动设备端：

safari iOS 4.33 – iPhone
User-Agent:Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5

safari iOS 4.33 – iPod Touch
User-Agent:Mozilla/5.0 (iPod; U; CPU iPhone OS 4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5

safari iOS 4.33 – iPad
User-Agent:Mozilla/5.0 (iPad; U; CPU OS 4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5

Android N1
User-Agent: Mozilla/5.0 (Linux; U; Android 2.3.7; en-us; Nexus One Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1

Android QQ 浏览器 For android
User-Agent: MQQBrowser/26 Mozilla/5.0 (Linux; U; Android 2.3.7; zh-cn; MB200 Build/GRJ22; CyanogenMod-7) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1

Android Opera Mobile
User-Agent: Opera/9.80 (Android 2.3.4; Linux; Opera Mobi/build-1107180945; U; en-GB) Presto/2.8.149 Version/11.10

Android Pad Moto Xoom
User-Agent: Mozilla/5.0 (Linux; U; Android 3.0; en-us; Xoom Build/HRI39) AppleWebKit/534.13 (KHTML, like Gecko) Version/4.0 Safari/534.13

BlackBerry
User-Agent: Mozilla/5.0 (BlackBerry; U; BlackBerry 9800; en) AppleWebKit/534.1+ (KHTML, like Gecko) Version/6.0.0.337 Mobile Safari/534.1+

WebOS HP Touchpad
User-Agent: Mozilla/5.0 (hp-tablet; Linux; hpwOS/3.0.0; U; en-US) AppleWebKit/534.6 (KHTML, like Gecko) wOSBrowser/233.70 Safari/534.6 TouchPad/1.0

Nokia N97
User-Agent: Mozilla/5.0 (SymbianOS/9.4; Series60/5.0 NokiaN97-1/20.0.019; Profile/MIDP-2.1 Configuration/CLDC-1.1) AppleWebKit/525 (KHTML, like Gecko) BrowserNG/7.1.18124

Windows Phone Mango
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0; HTC; Titan)

UC 无
User-Agent: UCWEB7.0.2.37/28/999

UC 标准
User-Agent: NOKIA5700/ UCWEB7.0.2.37/28/999

UCOpenwave
User-Agent: Openwave/ UCWEB7.0.2.37/28/999

UC Opera
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;) Opera/UCWEB7.0.2.37/28/999

用户追踪之基础技术——Cookie

前言

Cookie 是如此的重要，以至于我们后面要讲到的回头客定向、访客频次定向、用户定向等等都需要基于此技术才可以实现，并且我们日常工作中所能见到的第三方监测工具如 doubleclick、99click、秒针等也都要利用 cookie 技术，网站分析工具如 GA、百度统计、CNZZ 等也需要利用 cookie。如果没有 Cookie，互联网广告市场将受到巨大打击，尤其对于目前我们谈论的精准广告而言。如果没有 Cookie，网站分析也不从做起，遑论优化了。

Cookie 是什么

Cookie 在英文中是小甜品的意思，但在计算机语言中，Cookie 指的是当你浏览某网站时，网站存储在你电脑上的一个小文本文件，伴随着用户请求和页面在 Web 服务器和浏览器之间传递。它记录了你的用户 ID，密码、浏览过的网页、停留的时间等信息，用于用户身份的辨别。Cookie 通常是以 user@domain 格式命名的，user 是你的本地用户名，domain 是所访问的网站的域名。

为什么要 Cookie

因为 HTTP 协议是无状态的，对于一个浏览器发出的请求，服务器无法区分是不是同一个来源，无法知道上一次用户做了什么。所以，需要额外的数据用于维护会话。Cookie 正是这样的一段随 HTTP 请求一起被传递的额外数据，用于维护浏览器和服务器的会话。我们可以想象一个场景，你没有登录京东时在京东上购物，选择了 3 件

商品放入购物车，在结算时，京东为什么还能知道这三件商品是什么？没错，是Cookie！

Cookie 的传递流程

Cookie 利用网页代码中的 HTTP 头信息，伴随着用户请求和页面在 Web 服务器和浏览器之间传递。例如：当你在浏览器地址栏中键入了 Amazon 的 URL，浏览器会向 Amazon 发送一个读取网页的请求，并将结果在显示器上显示。在发送之前，该网页在你的电脑上寻找 Amazon 网站设置的 Cookie 文件，如果找到，浏览器会把 Cookie 文件中的数据连同前面输入的 URL 一同发送到 Amazon 服务器。服务器收到 Cookie 数据，就会在他的数据库中检索你的 ID，你的购物记录、个人喜好等信息，并记录下新的内容，增加到数据库和 Cookie 文件中去。如果没有检测到 Cookie 或者你的 Cookie 信息与数据库中的信息不符合，则说明你是第一次浏览该网站，服务器的 CGI 程序将为你创建新的 ID 信息，并保存到数据库中。（此例子来源于百度百科——Cookie）

关于 Cookie 的一些知识点

1、Cookie 是基于浏览器的，因此当电脑上安装多个浏览器时，服务器会生成多个 Cookie。虽然是同一个人，但服务器是识别为多个用户。

2、Cookie 是基于浏览器的，因此当同一台电脑有多个人使用时，服务器也只会生成一个 Cookie。虽然是多个人，但服务器会认为是一个用户。补充：在多个人均登录账户时，服务器可以以账户为区分，为每个账户生成单独的 cookie，比如多人用同一台电脑登录新浪微博。（感谢数据挖掘_PHP 的指正）

3、Cookie 是无法跨设备进行设置的。比如我们在单位和家里分别使用两台电脑，即使我们使用同一种同一版本的浏览器，我们还是生成了两个 Cookie，服务器会认为是两个用户。（PS：现在有些浏览器可以同步数据，比如 Chrome、Friefox，可以避免这种问题）

请注意：以上所说的 Cooke 指的全部是 Http Cookie。有一种 Cookie——Flash Cookie，可以解决多浏览器的问题。

关于 Flash Cookie

FlashCookie 是由 FlashPlayer 控制的客户端共享存储技术，鉴于目前 Flash 技术的普遍性，几乎所有的网站都采用，所以具有同 Http Cookie 一样的作用。在技术上，通过使用 JavaScript 与 ActionScript 可以将 Http Cookie 和 Flash Cookie 进行互通。

Flash cookie 的优势在于：

1、跨浏览器

不管用户的计算机上安装了多少个浏览器或者浏览器的不同版本，使用 Flash Cookie 能够使所有的浏览器共用一个 Cookie。

2、不易删除

所有的浏览器均提供了清除 Http Cookie 的快捷方式，但 Flash Cookie 并没有此种方式，并且其保存位置非常隐蔽，网民难以删除。

3、容量更大

Flash Cookie 可以容纳最多 100 千字节的数据，而一个标准的 HTTP Cookie 只有 4 千字节。

作为网络广告行业的销售人员，了解以上知识就已经绰绰有余了。如果想了解更多，可以接着往下看。

Cookie 的数量

1、大多数浏览器支持最大为 4096 字节的 Cookie。因此最好用 Cookie 来存储用户 ID 之类的标识符，用户的详细信息则通过用户 ID 从数据库或其他数据源中读取。

2、浏览器还限制站点可以在用户计算机上存储的 Cookie 的数量。大多数浏览器只允许每个站点存储 20 个 Cookie；当存储更多 Cookie 时，最旧的 Cookie 便会被丢弃。有些浏览器还会对它们将接受的来自所有站点的 Cookie 总数作出绝对限制，通常为 300 个。

Cookie 的失效时间

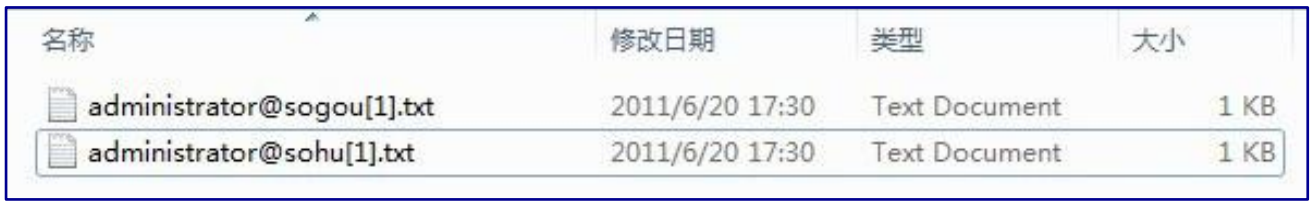
1、浏览器的 Cookie 设置会决定是否保存 Cookie 数据。如果浏览器不允许 Cookie 保存，则关掉浏览器后，这些数据就消失。

2、如果浏览器允许保存 Cookie，那么 Cookie 的时间由服务器的设置决定。Cookie 有一个 Expires（有效期）属性，这个属性决定了 Cookie 的保存时间，服务器可以通过设定 Expires 字段的数值，来改变 Cookie 的保存时间。如果不设置该属性，那么 Cookie 只在浏览网页期间有效，关闭浏览器，这些 Cookie 自动消失，绝大多数网站

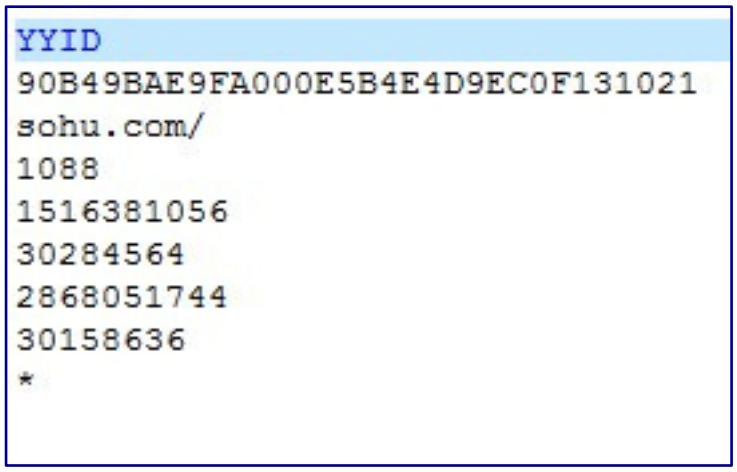
属于这种情况。通常情况下，Cookie 包含 Server、Expires、Name、value 这几个字段，其中对服务器有用的只是 Name 和 value 字段，Expires 等字段的内容仅仅是为了告诉浏览器如何处理这些 Cookies。

Cookie 的样例

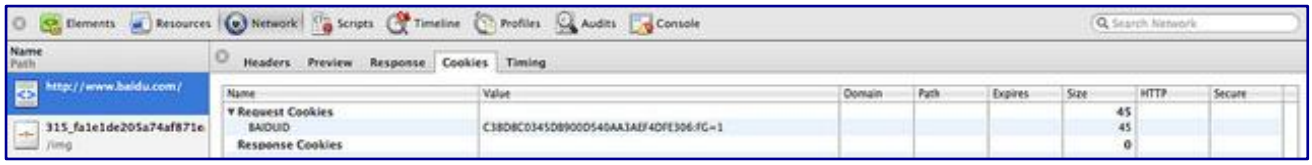
1、Cookie 的名称



2、Cookie 的内容



3、从页面代码监测工具看 Cookie



Cookie 的位置

1、Http Cookie 的位置

Windows 9X 系统 C:WindowsCookies

Windows NT/2000/XP 系统 C:\Documents and Settings\用户名\Cookies

win7 系统 C:\Users*\AppData\Roaming\Microsoft\Windows\Cookies*

OS X 系统 ~/Users/用户名/Library/Cookies

2、Flash Cookie 的位置

非 Win7 系统
C:\Documents and Settings\[username 你的用户名]\Application Data\Macromedia\Flash Player\#SharedObjects
Win7
C:\Users\[username 你的用户名]\Application Data\Macromedia\Flash Player
其中：Users 可能显示为“用户”
OS X 系统
~/Users/用户名/Library/Preferences/Macromedia/Flash Player/#SharedObjects
~/Users/用户名/Library/Preferences/Macromedia/Flash Player/macromedia.com/support/flashplayer/sys/

第一方 Cookie 和第三方 Cookie

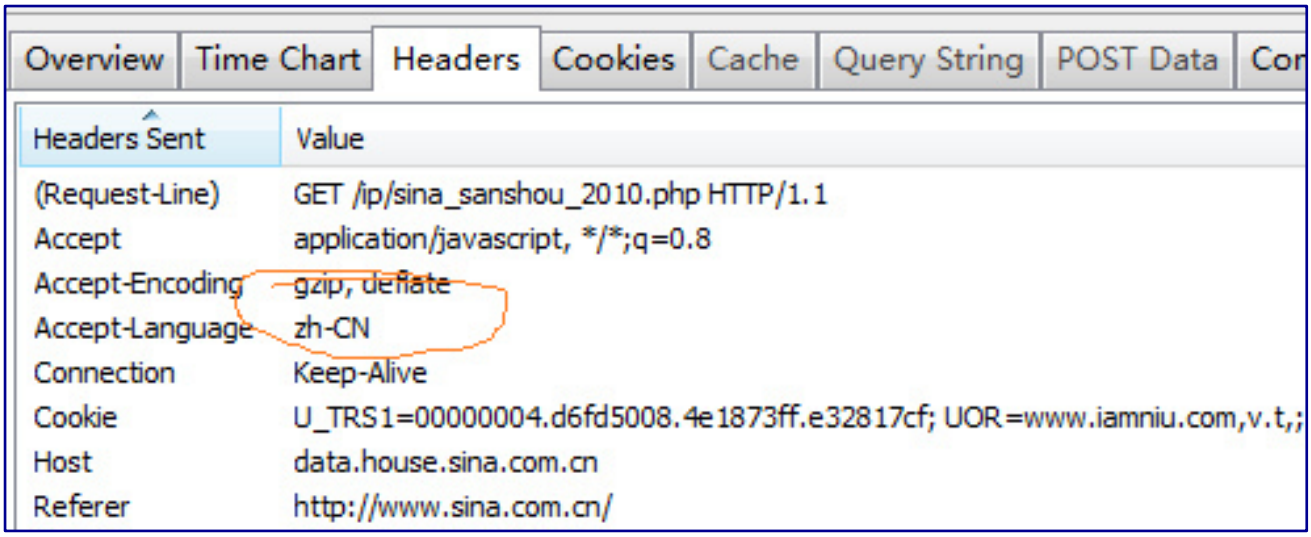
大多数的第三方监测工具和网站分析工具都会采用第三方 Cookie。所谓第一方和第三方的说法，是用来确定 Cookie 的归属的，这个归属是指 Cookie 中记录的域（domain）。第一方和第三方的唯一区别只是：Cookie 中的域名是否和被访问网站的域一样，是就是第一方，否就是第三方。举个例子：如果你访问网站 www.chinawebanalytics.cn 的时候，网站在你的电脑上设置了一个 Cookie，里面的记录的域名也是 www.chinawebanalytics.cn，那么这个 Cookie 就是第一方的，归你访问的网站 www.chinawebanalytics.cn 所有。而如果你访问网站 www.chinawebanalytics.cn 时，在你的计算机中设置的 Cookie 的域名是 www.abc.com，那么这个 Cookie 就是第三方 Cookie，归 www.abc.com 所有。

第一方 Cookie 并不一定需要由某个网站自己的服务器给自己建立，别的网站也能为它建立；而且，第一方 Cookie 也不一定是能由某个网站自己读取的，它完全可能由第三方读取。（以上内容和例子来自于捍卫 Cookie——没有 Cookie，我们什么都没有了）

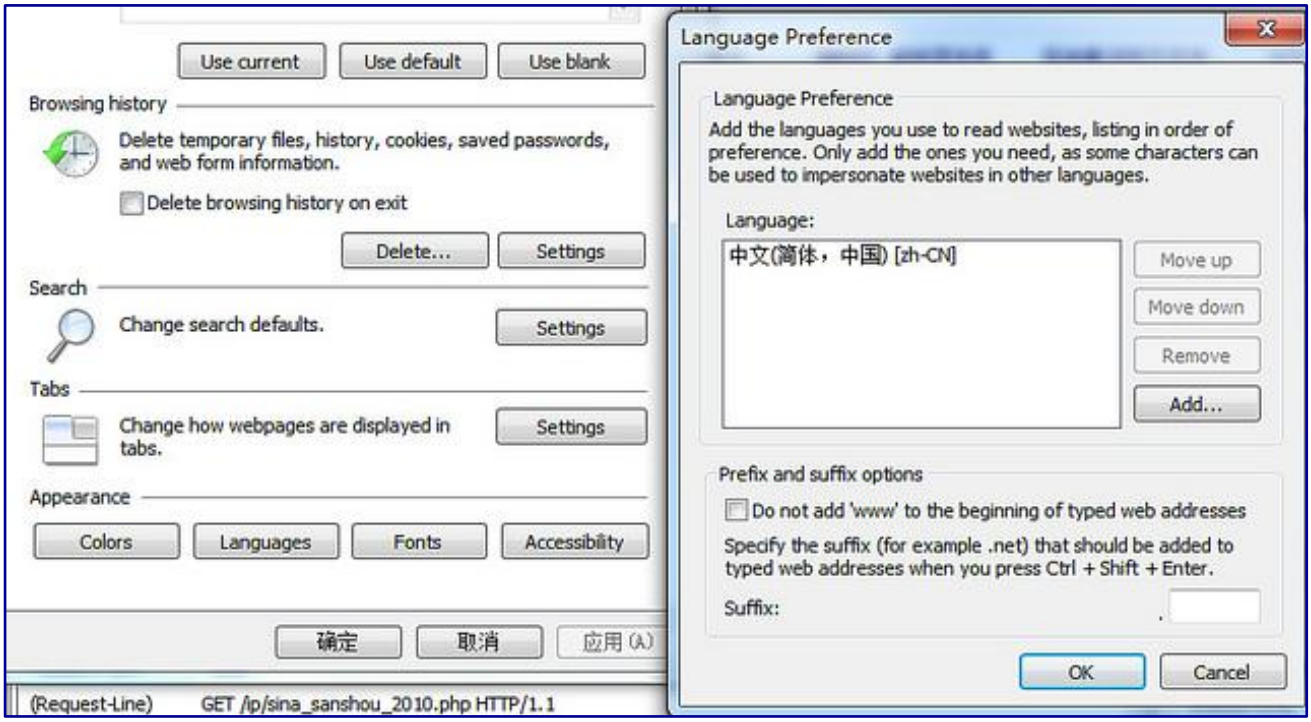
二、定向技术介绍：

语言定向

1、语言的来源
简单理解，语言指的是用户的浏览器语言，是从浏览器的 Http Header 的 Accept-Language 的字段来的。



2、浏览器的 Accept-Language 是由浏览器的语言设置所决定的。



3、浏览器的默认语言设置和浏览器语言无关，默认继承操作系统的语言。

当然，我们可以通过设置修改浏览器的语言设置，如下图，将浏览器语言设置为德文：

从下图中我们可以看到，Http Header 中的 Accept-Language 已经变为了 de（德文）：

浏览器定向

浏览器定向同样需要依赖于各个浏览器在打开页面时所传输的 Http header 信息中的 User-Agent。

我们来了解 User-Agent 中浏览器及版本识别的方法：

一、浏览器的使用率说明：

浏览器类型	2012年01月使用率	2012年01月占有率
Internet Explorer	54.91%	55.14%
奇虎360旗下浏览器	27.99%	27.32%
搜狗高速浏览器	7.73%	9.04%
Chrome	2.42%	1.82%
Safari	2.38%	2.52%
腾讯旗下浏览器	1.43%	1.79%
火狐	1.41%	0.77%
Theworld	0.80%	0.73%
傲游	0.74%	0.76%
Opera	0.16%	0.11%

——数据来源于 CNZZ 数据中心

我们针对以上的浏览器进行说明，另外再针对移动设备上的几款浏览器进行说明。

二、浏览器识别

1、IE 浏览器（以 IE 9.0 为例）

PC 端：User-Agent:Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0;
移动设备：User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0; HTC; Titan)

由于遨游、世界之窗、360 浏览器、腾讯浏览器以及搜狗浏览器、Avant、Green Browser 均采用 IE 的内核，因此 IE 浏览器判断的标准是”MSIE“字段，MSIE 字段后面的数字为版本号，但同时还需要判断不包含”Maxthon“、”The world“、”360SE“、”TencentTraveler“、”SE“、”Avant“等字段（Green Browser 没有明显标识）。移动设备还需要判断 IEMobile+版本号。

2、360 浏览器

PC 端：User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; InfoPath.2; .NET4.0C; .NET4.0E; .NET CLR 2.0.50727; 360SE)
移动设备：暂无

360 浏览器的判断标准是”360SE”字段，没有版本表示。

3、搜狗浏览器

PC 端：User-Agent:Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; SE 2.X MetaSr 1.0; SE 2.X MetaSr 1.0; .NET CLR 2.0.50727; SE 2.X MetaSr 1.0)
移动设备：暂无

搜狗浏览器的判断标准是”SE“、”MetaSr“字段，版本号为 SE 后面的数字。

4、Chrome

PC 端：Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_0) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.56 Safari/535.11
移动设备：User-Agent: Mozilla/5.0 (Linux; U; Android 2.2.1; zh-cn; HTC_Wildfire_A3333 Build/FRG83D) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1

PC 端 chrome 浏览器的判断标准是 chrome 字段，chrome 后面的数字为版本号；移动端的 chrome 浏览器判断”android“、”linux“、”mobile safari“等字段，version 后面的数字为版本号。

5、Safari

PC 端：User-Agent:Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_8; en-us) AppleWebKit/534.50 (KHTML, like Gecko) Version/5.1 Safari/534.50

移动设备：User-Agent:Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_3_3 like Mac OS X; en-us) AppleWebKit/533.17.9 (KHTML, like Gecko) Version/5.0.2 Mobile/8J2 Safari/6533.18.5

由于 Chrome 及 Nokia’ s Series 60 browser 也使用 WebKit 内核，因此 Safari 浏览器的判断必须是：包含 safari 字段，同时不包含 chrome 等信息，确定后”version/“后面的数字即为版本号。在以上条件下包含 Mobile 字段的即为移动设备上的 Safari 浏览器。

6、腾讯浏览器

PC 端：User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; TencentTraveler 4.0; .NET CLR 2.0.50727)
移动设备：User-Agent: MQQBrowser/26 Mozilla/5.0 (Linux; U; Android 2.3.7; zh-cn; MB200 Build/GRJ22; CyanogenMod-7) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1

腾讯浏览器的判断标准是”TencentTraveler“或者”QQBrowser“，TencentTraveler 或 QQBrowser 后面的数字为版本号。

7、Firefox

PC 端：User-Agent:Mozilla/5.0 (Windows NT 6.1; rv:2.0.1) Gecko/20100101 Firefox/4.0.1
移动设备：User-Agent: Mozilla/5.0 (Androdi; Linux armv7l; rv:5.0) Gecko/ Firefox/5.0 fennec/5.0

Firefox 的判断标准是 Firefox 字段，firefox 后面的数字为版本号。

8、The world

PC 端：User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The World)

移动设备：暂无

Theworld 浏览器的判断标准是”The world“字段，没有标示版本号。需要注意的是：The world 2.x 版本的 User-Agent 中没有”The world“的字段。

9、遨游

PC 端：User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Maxthon 2.0)
移动设备：暂无

遨游浏览器的判断标准是”Maxthon “，Maxthon 后面的数字为版本号。

10、Opera

PC 端：User-Agent:Opera/9.80 (Windows NT 6.1; U; en) Presto/2.8.131 Version/11.11
移动设备：User-Agent: Opera/9.80 (Android 2.3.4; Linux; Opera mobi/adr-1107051709; U; zh-cn) Presto/2.8.149 Version/11.10

opera 浏览器的判断标准是 opera 字段，opera 字段后面的数字为版本号。

11、UC 浏览器

UC Web 有多种模式浏览方式，对应的 User-Agent 为：

UC 无
User-Agent: UCWEB7.0.2.37/28/999

UC 标准
User-Agent: NOKIA5700/ UCWEB7.0.2.37/28/999

UCOpenwave

User-Agent: Openwave/ UCWEB7.0.2.37/28/999

UC Opera
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0;) Opera/UCWEB7.0.2.37/28/999

UC 浏览器的判断标准是”UCWEB“字段，UCWEB 后面的数字为版本号。

操作系统定向

操作系统定向依赖于各个浏览器在打开页面时所传输的 http header 信息中的 User-Agent。

我们来了解 User-Agent 中的不同操作系统的识别方法。

PC 端：

平台	标识	备注
FreeBSD	X11; FreeBSD (version no.) i386	
	X11; FreeBSD (version no.) AMD64	
Linux	X11; Linux ppc	
	X11; Linux ppc64	
	X11; Linux i686	
	X11; Linux x86_64	
Mac	Macintosh; PPC Mac OS X	
	Macintosh; Intel Mac OS X	
Solaris	X11; SunOS i86pc	
	X11; SunOS sun4u	
Windows	Windows NT 6.1	对应操作系统 Windows 7
	Windows NT 6.0	对应操作系统 Windows vista
	Windows NT 5.2	对应操作系统 Windows 2003
	Windows NT 5.1	对应操作系统 Windows xp
	Windows NT 5.0	对应操作系统 Windows 2000
	Windows ME	
	Windows 98	

移动设备端：

平台	标识	备注
iOS	iPhone OS 4_3_3	iPhone
	iPhone OS 4_3_3	iPod
	OS 4_3_3	iPad，注意仅是OS，没有i
Android	Android 2.3.7	后面数字为版本号
WebOs	hpwOS/3.0.0	
SymbianOS	SymbianOS/9.4	
Windows Phone OS	Windows Phone OS 7.5	
BlackBerry	BlackBerry	

地域定向

地域定向依赖于对 IP 地址的识别，而 IP 协议是互联网的基础协议，因此从网络诞生的第一天起，地域定向就可以被使用了。

欲详细了解 IP 协议，请查看百度百科——TCP/IP 协议。有关 IP 地址的详细信息，请查看百度百科——IP。

通俗来讲，IP 地址就是互联网上的门牌号，接入互联网的所有主机就是我们的一个个住所，其中有个人的，有单位的。个人住所一家一个门牌号，单位的多家公用一个门牌号，由于规划的原因，有的住所会有多个门牌号，也是规划的原因，门牌号有时会发生变化。IP 地址也有此特点，一台主机可以具有多个 IP 地址，而多台主机也可以公用一个 IP 地址。

现实中，不管如何规划，通过门牌号我们能找到我们要找的住所，也能清楚住所所在的具体位置。同样，在网络中，通过 IP 地址我们也能定位到我们所需要找的主机，并且清楚知道主机所在的地理位置。这样我们就能进行广告的地域定向了。

从技术层面讲，地域定向的工作逻辑是：
当一个请求发送给服务器时，服务器根据配置（以 Apache 为例，在 Apache Httpd 中进行配置）记录下请求的相关数据，组成日志文件，日志基本会包括请求时间、请求 IP、请求的 URL、请求的 Reffer、请求的 User-Agent 以及其他信息，将收集到的 IP 地址与已有的 IP 数据库进行比对，即可以确定请求者的地理位置了，比如山西省太原市。当然，请求的 IP 信息从 http 协议中可以获取，不是必须依赖于日志。（感谢 数据挖掘_PHP 的指正）

国内目前免费的 IP 库有 QQ IP 数据库 纯真版，即我们通常所说的纯真 IP 库，收集了包括中国电信、中国网通、长城宽带、网通宽带、聚友宽带等 ISP 的最新准确 IP 地址数据，包括最全的网吧数据。IP 数据库每 5 天更新一次，企业可以在此基础上修正后使用。

目前的地域定向更多的是针对省份以及地级城市的定向，针对县级市或者区级的定向基本上都十分不准确。

回头客定向

随着电商网站的火爆，从 2010 年开始，互联网广告行业出现了一种定向方式——回头客定向。回头客定向是随着精准理念的发展而提出来的。顾名思义，回头客定向是指针对到达过广告主网站的某一个点的用户或者发生过某一个行为的用户进行定向。从概念中，我们可以发现回头客定向的三个基本点：**1、到达过；2、某一个点或某个行为；3、定向投放。这三点也是回头客定向和人群定向的区别之处。**

从营销的角度讲，针对不同到达深度的用户或者不同行为的用户，我们需要采取的营销策略可能会有不同。我们以电商网站的购物流程来举例子。电商网站的购物流程分为以下几个步骤：

1、针对浏览过商品的人，我们应该分析他的浏览记录，发现他感兴趣的商品，然后通过广告将他感兴趣的商品推送到他的面前（如果要做到非常完美，针对每个用户有不同的广告显示，需要有哪些条件？大家可以评论，我们一起交流）。
2、针对已经将商品加入购物车的人，此时可能更重要的是给他一张电子优惠券，以促进其下单。
3、针对到达过注册或者登录界面，但未完成注册和登录的人，给他一个商品即将售罄或者即将涨价的倒计时更能促进其回来下单。
4、针对到过填写配送地址页面但没有提交订单的人，提示免邮递费用或者直接告诉他“你还差一步就将完成订单”，可能会是一个好的方法。
5、已经提交订单的人，是我们的老客户了，此时应该推荐关联的商品信息，以促进其二次消费。

所以，进行回头客定向的投放，一定是要有以下三个步骤的：

1、设置回头客人群的监测。支持回头客定向的系统必须能够支持对各个点的监测，因此提取监测代码在此是必须的。好的系统可以利用一个监测代码，通过数据分析得出不同监测点的回头客（大家说如何做到？）；差的系统就提供不同监测点的设置功能，每个监测点提取不同的监测代码。

2、整理针对各个监测点用户的独特营销诉求。制作针对不同回头客的不同创意。

3、利用投放系统，对回头客进行定向的广告投放。

一般来讲，定向越准确，能得到的量就会越少，因此，在做回头客定向时，不应该再选择媒体进行投放。从另一个角度理解，回头客定向已经是最领先精准的目标用户定向了，此时媒介选择的意义也大大弱化了。

以上所说的是纯正意义上的回头客定向，鉴于回头客定向受人欢迎的精准的概念和可怜的流量，有些人或公司权衡后会将回头客定义的非常广泛，比如到过网站的人、点过广告的人、看过广告的人都算作回头客，这只是又一次的中国特色而已。这种事情多了，反而于精准广告市场的发展不利。

人群定向

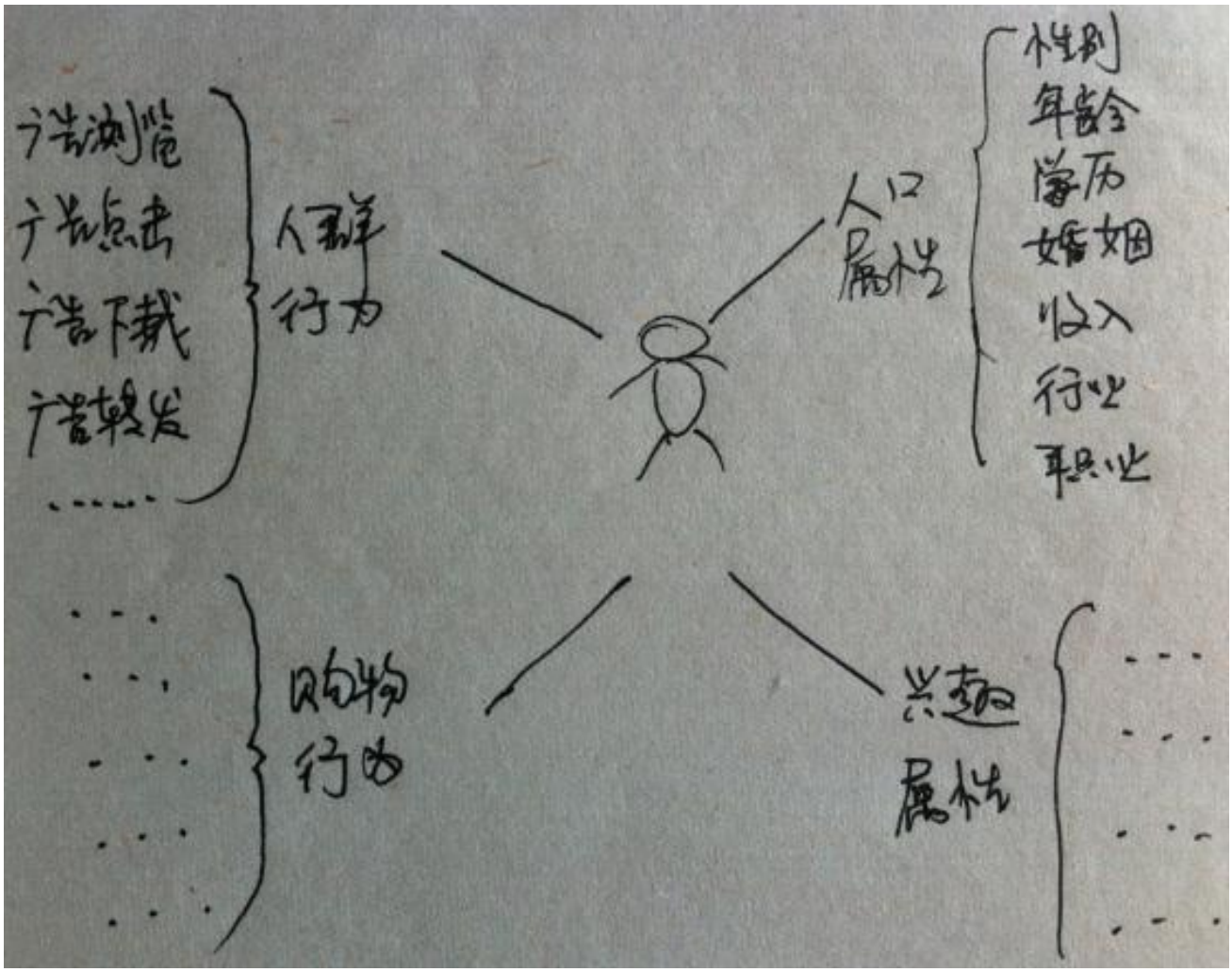
人群定向其实就是目标人群定向，在营销学中，产品定位以及人群细分是非常重要的理念，这种理念也已经得到了市场的认可，因此每一种产品在设计、生产之初就已经确定了自己的目标人群。从我们的广告投放、市场宣传来讲，一定是希望能给对目标人群进行，花费在目标人群之外的推广都是浪费的。

但在以往的媒介中，想要完全的识别用户，以确定是否目标人群并不是容易的事情，甚至从理论上说是完全做不到的，只能通过不同的媒介手段去尽量的靠近目标人群（电视、广播、杂志都是如何确定自己的受众的呢？有人讨论嘛？）。但即使这样，也产生了一句广告界最著名的话语——我知道广告费浪费了一半，但我不知道到底是哪一半。

在互联网时代，通过技术的力量，可以无限的接近、近乎准确的判断每一个人的属性，从而为广告主目标群体定向服务。但是，互联网也只是无限的接近，而不是确切的能

标示出个人的属性。目前，最接近的应该是类似于罗维邓白氏之类公司的数据（顺便说一句，央视 315 晚会的曝光，对罗维邓白氏公司只能是免费的广告，而不是打击）。

言归正传，我们来说说互联网的人群定向。互联网公司通常讲的人群定向并不单单包括人口的自然属性（demographic），还包括人群兴趣（interest）、人群行为（behavior）、购物行为（purchasing）。



注：此处我们说的人群行为指的是对广告的行为，比如浏览广告，点击广告以及转发、下载广告等交互行为。目前市场上经常有一些公司标榜行为定向，但让其展开一说，就只是说对用户的浏览行为进行定向，非常正确、毫无破绽的说法，但细问却还是这一句。这只能说明这种公司忽悠而无真章的事实（大家说说为什么能说明？）。

对于真正提供定向的公司，不管各个公司都提供什么样的人群定向，以上所说的 4 类属性或行为都是基于 cookies 技术，通过对用户长期的互联网浏览行为数据进行分析所得出的。由于各公司的资源优势不同，因此目前没有一个公司能够建立健全的数据。

自然属性（demographic）

自然属性包括性别、年龄、学历、地域、婚姻状况、家庭状况（是否有小孩，小孩年龄等）、收入（个人收入、家庭收入）、行业、职业等信息。单纯通过互联网浏览行为并不能分析到如此全面且准确的信息，目前还主要以找到真实的样本进行建模分析为主。自然属性数据以艾瑞的数据最为准确。

人群兴趣（interest）

人群兴趣在每个公司会有不同的认知。目前，兴趣数据属悠易最好，悠易的数据是公开的，可以通过悠易受众引擎查看。

人群行为（behavior）

上面注解所说的人群行为仅仅是行为中的一种，如果有搜索引擎的资源，则可以加入搜索行为的监测（如百度的搜客定向——对在百度搜索过已添加关键词的人，在其浏览指定的投放网站时投放客户推广组下的创意。）；如果有微博数据，则可以加入关注与被关注的行为（新浪有此打算吗？），因此人群行为各公司的定义差异是最大的。

购物行为（purchasing）

购物行为指的是作为消费者角色，互联网用户的消费数据。毋庸置疑，购物数据如果淘宝是第二，也没人可以自称第一。

在广告系统中，用户的所有属性或行为应该是可以进行自由组合设定的。但以上所有的属性或行为就可以全方面的了解用户了吗？并不是！这是一个发散性的命题，每个人会有不同的见解。比如我们还可以加入用户的设备（PC、Pad、移动设备等），通过用户上网通道来描述用户。还有其他的角度吗，大家留言讨论吧！

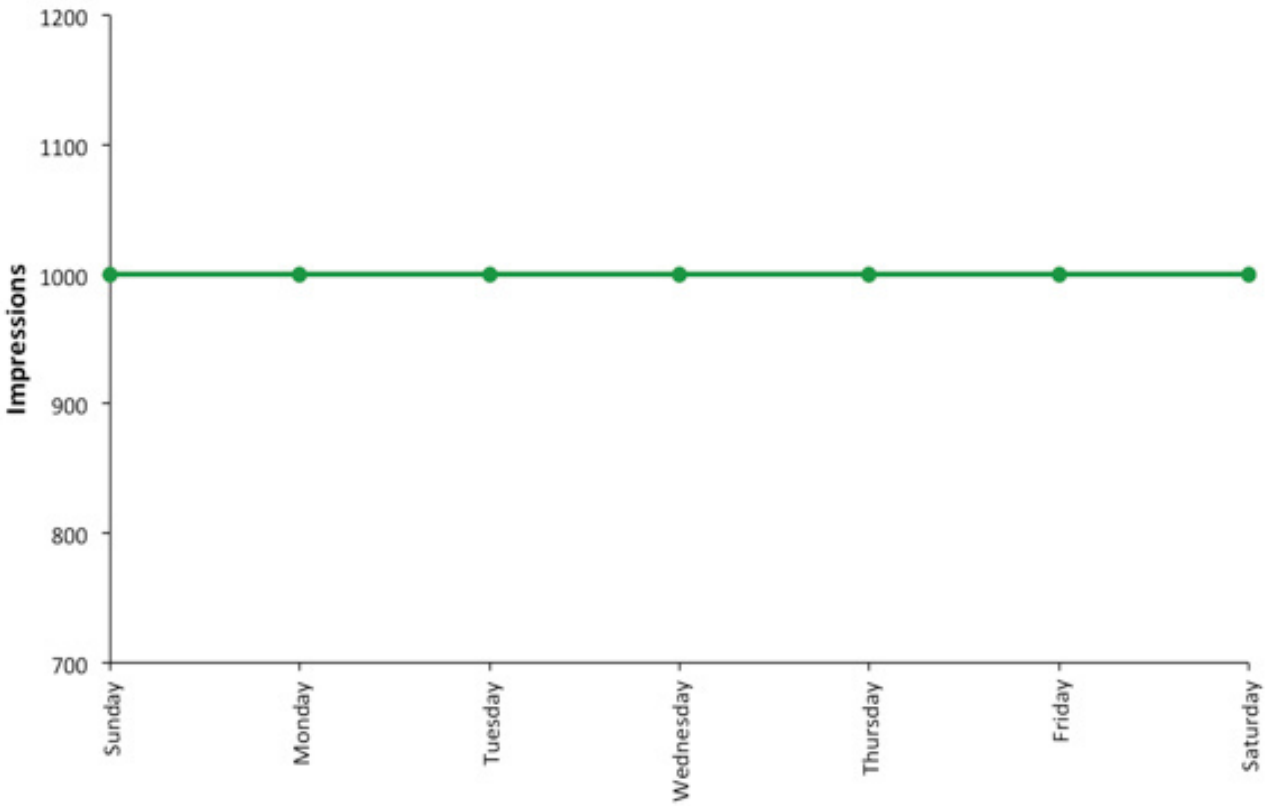
并发次数

在按天售卖或者按时间售卖的时代，是不需要考虑并发次数的。只是在按照展现次数（CPM）售卖的时候，我们才有可能需要考虑广告并发的设置。

在按照 CPM（何为 CPM）售卖时，广告投放的速度可以有两种——尽快投放和匀速投放。尽快投放很好理解，就是尽快投放完规定的量。匀速投放就是在规定的时间内均匀的投放完规定的量。举个例子，一天之内投放 1000 个 CPM，选择尽快投放就意

味着广告在第 x 小时投放完毕，那么(24-x)的时间内就不会再看到广告；而匀速投放意味着我们需要在第 23 小时 59 分时还看到广告。这个如何做到呢？此时就需要利用并发次数的设定了。

并发次数指的是广告某个时间周期内播放的次数，其目的是为了保证广告的匀速投放。并发次数的计算方法为：广告投放量/投放时长。注意：此处的时长根据需要，可以按照秒、分、刻等单位来计算。并发次数的规则需要广告投放核心的支持，当在规定的时长内，广告未达到并发次数时，广告可以展现。达到设置次数后，则不予以展现。



一个思考题：*如果一个广告一天内要求投放 1000CPM，而媒体的 PV 一天正好是 1000CPM，那么尽快投放是否能够跑完广告的规定量？匀速投放是否能够跑完广告规定的量？如果跑不完，我们需要怎么做，才可以跑完？*

时段定向

每一个广告活动，每一次宣传活动，都会有周期的设定。在一个投放活动被制订出来后，在每种媒介、每个媒体上的投放周期就已经确定了。电视、广播、报纸杂志是以节目的

播放时间、广告顺序以及报刊杂志的期数来决定投放的周期的。互联网广告则以开始日期、结束日期以及投放时段来决定投放周期的（需要注意的是：**投放时间是以服务器的时间为准的**）。

说明：互联网的时间使用的是 UTC 时间体系，北京时间=UTC+8。（关于 UTC 时间和 GMT 时间以及北京时间的关系）

问题：*在广告系统设计时，怎样设计可以使用户方便快捷的设置每天不同的投放时段？*

网页定向

网页定向指的是针对特定的 URL 进行定向，使广告投放在指定的 URL 上。网页定向是互联网广告定向中不常使用的定向。

网页定向最核心的技术有两个：

1、如何获取当前页的 URL，注意是当前页非 Http Header 中的 Reffer。当前页 URL 需要通过加在页面上的 JS 代码获得，设计时需要考虑到如果 JS 代码被放在 iFrame 中的情况，甚至会被放置到好几层嵌套的 iFrame 上（这样放置代码的媒体更多为了作弊，可以参见在线广告作弊手段一览【见下】）。

2、广告系统在定向设置时需要考虑到 URL 匹配问题。左匹配、右匹配、包含、不包含、通配符等。匹配规则需要在广告投放核心进行处理。

访客频次

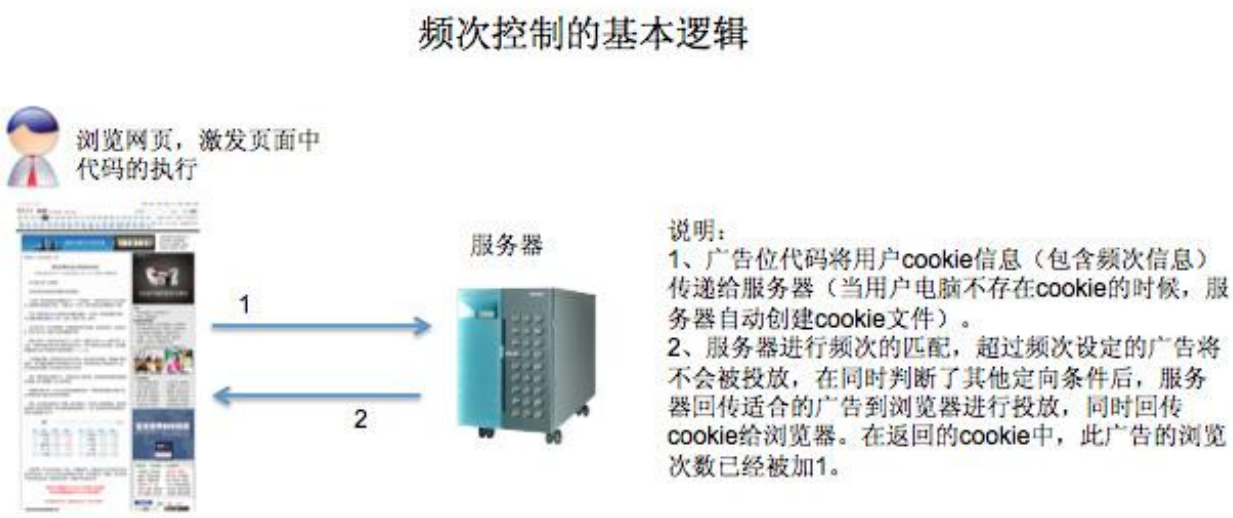
频次是广告投放中一个非常重要的概念。网络广告的频次和其他媒介投放时的频次概念是一致的。

频次是指个人或家庭接触广告信息的次数。在传统的电视媒介中，我们不能准确的控制每一个人接触广告信息的次数，只能是通过总收视点除以到达率计算得出。但是在

网络广告中，一个人可以接触广告信息的最高频次是可以严格控制的，实现严格控制的基础技术也是 cookie，可见 cookie 对于互联网广告精准投放的重要性。

在网络广告的投放中，频次的控制对象比其他媒介更广泛，频次可以控制广告的浏览、点击、完整浏览，甚至是广告的转发、下载等其他的行为，因此互联网的频次指的是访客与广告发生互动的最高次数，而互动的行为设定则需要能够在广告系统中进行设置。当然经常还是对广告的浏览进行频次设置，我们也以此举例。

网络广告频次控制的原理非常简单。当用户通过浏览器访问页面时，会请求放置在页面的广告位代码，广告位代码和服务器进行交互，广告位代码将用户的 cookie 信息（包含对广告的访问次数）传给服务器（如果没有 cookie，服务器会生成一个），服务器进行频次的匹配，超过频次设定的广告将不会被投放，在同时判断了其他定向条件后，服务器回传适合的广告到浏览器进行投放，在返回信息的同时，还会将用户 cookie 上此广告的浏览次数加 1。通过这种方式，网络广告实现了精确的频次控制。



广告投放中，并不是频次越高越好，过少的接触不会在接触的用户心中产生印象，过多的接触反而会使接触的用户产生不快，厌恶。1972 年，美国心理学家赫尔伯特.克鲁格曼经过研究，确立了消费者接触广告三次的心理学关系：第一次好奇：“这是什么？”第二次是认识：“干什么用的？”第三次是判断：“对广告产生什么印象？”。当然，因为产品、市场、品牌、竞争、创意以及媒体等不同，在频次设置上也会有所不同，不过，对广告的有效接触频次限定一般都是以 3 次为底限的。

为了了解广告的投放效果，在报表中，广告系统一般会提供平均接触频次、频次分布图。

讨论：频次分布图是什么样子？设计时需要注意什么？

关键词定向

我们所讲的关键词定向实际上就是 Google AdWords 中的内容相关广告（Contextual）。

关键词定向实现必须具备以下能力：

抓取网页内容并进行分析的能力

分析时需要考虑到页面的结构、html 标签、链接等影响，对页面的正文进行分析，得到最恰当的一些关键词来描述页面所表达的内容。关键词定向是否有效的瓶颈即在于此。

需要注意的是，由于实时快速分析页面的要求非常高，当页面足够多的时候，系统执行效率会非常的低下，因此必须具有提前抓取有可能出现广告页面的能力。

当然，实时快速分析同样重要。

广告系统中设置广告投放关键词的能力

需要能够确保操作人员可以方便快捷的在系统中进行关键词的设置（正向选择、反向排除），如果能够提供对之前投放的关键词效果分析及推荐更好。

投放核心快速匹配投放能力

将 1 的分析结果和 2 的投放设置进行快速匹配并进行投放，这是最根本的要求。

关键词定向的效果：

Targeting type	Ad requests	Coverage	Clicks	Ad request CTR	CPC	Ad request RPM	Estimated earnings ↓
Interest-based	765	100.00%	15	1.96%	$0.13	$2.47	$1.89
Contextual	793	100.00%	19	2.40%	$0.10	$2.28	$1.81
(Unmatched ad requests)	27	0.00%	0	0.00%	—	$0.00	$0.00
Averages	528	—	11	—	—	—	$1.23
Totals	1,585	98.30%	34	2.15%	$0.11	$2.33	$3.70

三、网络广告反作弊：

在线广告作弊手段一览

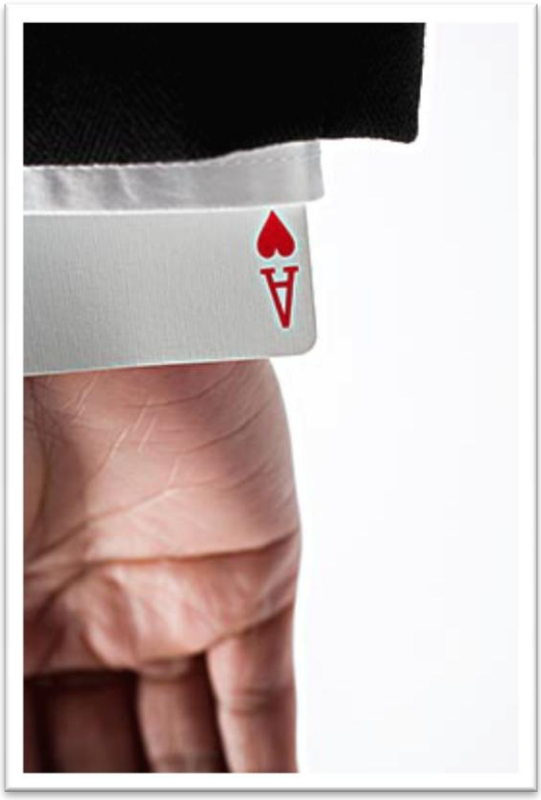
以下文章的内容转自牛人扫地老僧的 blog，原文链接地址为 URL: http://www.doyj.com/2011/11/11/在线广告作弊手段一览/。以下为正文：

这里提到的在线广告作弊是指媒体为了刷广告流量而进行的作弊。他们的作弊手段很多， 这里介绍常见的几种。

iframe 是广告作弊最常用的技巧，就是在自己的网页上嵌入 iframe, 大小为 0×0 或 1×1，也就是用户不可见。通过 iframe 打开其他页面，在用户看不见的情况下刷流量。别看 iframe 简单，里面花样很多。

1、页面内嵌入本站页面的 iframe

iframe 打开和当前页一样的页面地址，或本站的其他页面。 这样用户的一个浏览行为，很轻松就从 1 个 pv 翻倍变成 2 个 pv。如果嵌入 iframe 多点， 就能翻 3 倍，4 倍…。但使用这个方法很容易被发现，广告投放方，通过分析 UV，独立 IP 等很容易就发现异常。 这是很老的方法，不过还是有些网站乐此不疲。



2、两个站点间互相嵌入对方站点页面的 iframe

这是比较巧妙的作弊技巧，UV，独立 IP 等分析方法是不能发现异常的。

3、双层 iframe

作弊的 iframe 为了不让人看见，大小只有 0×0 或 1×1，但有些在线广告在显示时会判断浏览窗口大小，如果太小可能就不能显示。这时有些网站就采用了双层 iframe 技术来刷广告流量。 第一层 1×1 大小的 iframe 中又嵌入一个 iframe，这个第二层 iframe 是正常浏览窗口大小，广告代码很难发现异常。

这种作弊方式使用巧妙的，会让主页面和两个 iframe 使用三个不同的域名，这样因为跨域的问题， 里面的 js 不可能得到最外层真正的页面地址， 想抓证据都抓不到。

4、IP 屏蔽

有些站点在进行作弊时，会屏蔽北京，上海等大城市的访问，你从这些地区访问时，看不到他们的作弊代码，一切正常。等换用其他地方的代理访问时，你在他们页面里就能看到作弊用 iframe 代码。 这是因为很多 IT，在线广告公司都在这些大城市，这种屏蔽让他们的作弊手段很难被同业发现。

5、购买垃圾流量

现在来自 iframe，木马的垃圾流量都是明码标价在卖的，可以用这些流量来刷页面，刷广告。这种也比较难以发现。

网站去刷流量目的往往比较复杂，一是刷广告流量，赚广告商和广告主的钱，二是为了 alexa 之类的排名，也有是为了给投资人看所谓的“业绩”。

上面谈的基本都是 CPM 广告方式的作弊，下面说说其他的。

6、CPC 作弊

CPC 作弊其实是很简单的，只要用 iframe 打开点击链接即可。

7、CPA 作弊

有些网站广告按 CPA 结算，比如注册人数等。 这种情况下，有的公司会做专门的自动注册机，保证你的注册人数疯狂上涨。

8、CPS 作弊

很多人感觉 CPS 方式是不可能作弊的，其实这也是可以的。 卖个关子，下次和大家介绍。

其实大多数作弊方法通过耐心细致的分析，不难发现端倪。 但中国在线广告的作弊还是相当严重，形成这种现象的原因是蛮有意思的，和多人都探讨过， 有时间另写个博文分析。[转载正文结束]

扫地老僧一直走在高精尖道路上，因此以上提到的都是比较巧妙以及难以发现的作弊方式，经常是一些流量比较大的媒体使用。不过目前还有大量的小流量媒体采用更原始的方式，那就是强制刷新页面，在用户看页面时，每个几秒页面就刷新一次。

CPC、CPA、CPS 广告的作弊方法

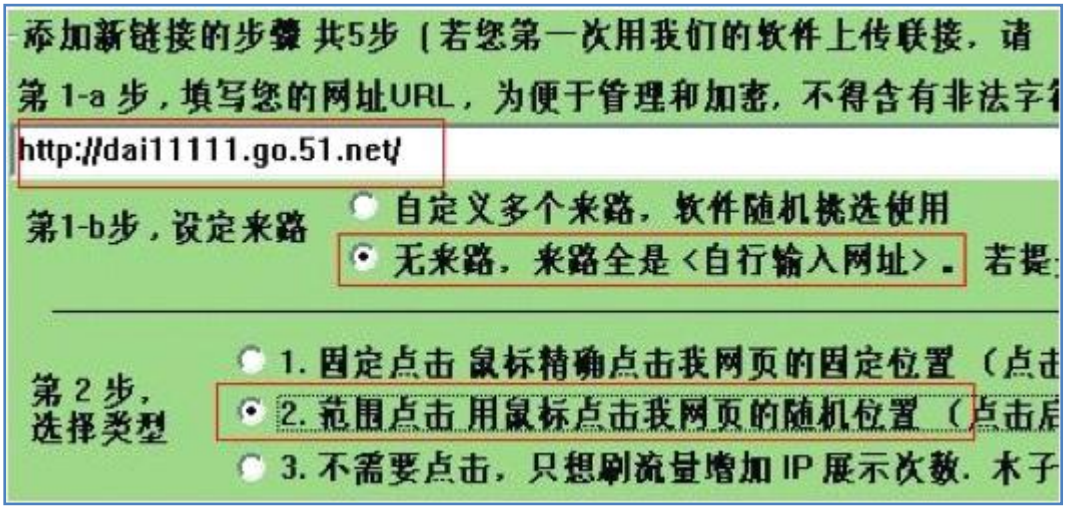
在之前已经介绍过网站对广告流量即广告的 CPM 作弊的方法《在线广告作弊手段一览》，也介绍了实际工作中通过异常现象进而分析日志找到作弊的数据《如何从日志中发现广告被作弊？》，今天我们介绍一下媒体对 CPC、CPA、CPS 广告可能进行的作弊方法。分析作弊方法是为了我们更好的了解对手（应该不至于这么仇恨），为了更好的防作弊，千万不要抱着“我也要作弊”的想法来了解。

CPC 作弊

总的而言，CPC 作弊容易，但也是容易反查出来的。
目前国内最常用的点击软件有：木子联盟、先锋专家。我们通过木子软件的设置截图来了解点击作弊软件的功能。

链接管理

链接设置：从截图可以看到，软件可以设置来路和鼠标的点击位置



点击区域的设置：比许多广告系统的易用性都好

CPA 作弊

CPA 计费会因为客户对 A 的定义不同而产生多种情况，比如 A 有可能是注册，有可能是安装软件，甚至可能是注册了以后进行了登录、安装软件后进行了操作等等。
以软件为推广例，一般来说，广告主会在广告展现到软件被使用的各个关键点设置监测。一个软件安装并使用的流程大概为：
第一步，广告物料展现
第二步，点击进入广告着陆页
第三步，点击软件下载安装
第四步，软件安装
第五步，软件使用
第六步，软件卸载
广告主在每个关键点会如何监测，采用哪些监测技术，大家可以提出自己的想法，一起讨论。
如何广告主严格按照上面的第一到第五步作为 A 的考核标准，那么还能有什么作弊方法呢？如果只有第一到第三步，可以利用注册机进行注册并模拟下载，但是很少有注册机可以做到第四第五步的。但是还有一个方法可以做到，那就是利用安装在用户电脑里面的木马程序。通过木马程序暗地里走完整个安装使用流程。

那对这种作弊我们有办法去识别吗？肯定会有！任何作弊肯定会有漏洞存在。我们可以想象到，如果软件是用户主动安装，那么一定会存在特有的使用规律，而这种使用规律很难模拟，因为谁也不会知道某个软件的用户使用规律是怎么样的。关于这方面的分析，可以看之前的文章《渠道商用假量冒充真实用户：开发者求给条活路》。

CPS 作弊

随着现在 CPS 联盟平台的兴旺以及大型电商自建 CPS 平台的发展，越来越多的小站长加入到了 CPS 的联盟里面。CPS 对站长来说确实是非常苛刻的计费方式，完全抹去了小站点对产品品牌曝光的贡献。但上有政策下有对策，在实际发展中，小站长也有了自己的作弊方法。

Cookie Stuffing

Cookie Stuffing 指的是通过网站上安装的特别的程序,把特定的 Cookie 植入访问者的电脑里，当电脑的使用者，去一些特定的网站买东西的时候，你就有了提成。Cookie Stuffing 的存在是因为广告联盟用 Cookie 来跟踪业绩。Cookie 产生后被存放在用户的浏览器中，当用户在广告主网站上产生任何的消费行为时，因为这个 Cookie 的存在，广告主会认定此次销售是你带过去的，因此，你可以获得提成。关于 Cookie 的知识，请参考《用户追踪之基础技术——Cookie》。

Cookie Stuffing 技术有三种常见方法：
1、图片
基本代码是：
只要访问者访问到了放有此代码的这个页面,你的 affiliate cookie 就会强制植入他们的电脑里。
2、iFrame
基本代码是：<iframe src=“” width=“1”height=”1” scrolling=“auto” frameborder=“0”></iframe>
其原理是将一个很小的 iframe（0×0 或者 1×1）放在自己的 affiliate 网站中。当用户浏览自己的 affiliate 网站时，将 cookie 存入用户的电脑。
3、.htaccess
基本代码是：RewriteEngine on

RewriteRule affiliate.jpg http://affiliatelink.你的域名.com/
[L,R=301]
affiliate.jpg 是你要用在网站上的图片，比如 http://www.你的域名.com/affiliate.jpg。当访问者试图读取这个图片时，其实你的 affiliate cookie 已植入他们的电脑里。
Cookie Stuffing 是小站长为了扩大点收入所采用的方法，不会一夜暴富，也不会影响其他站长。但有一种方法，却是典型的损人利己，是在剥夺其他人好不容易获得的收入。这种方法就是 Http 劫持。

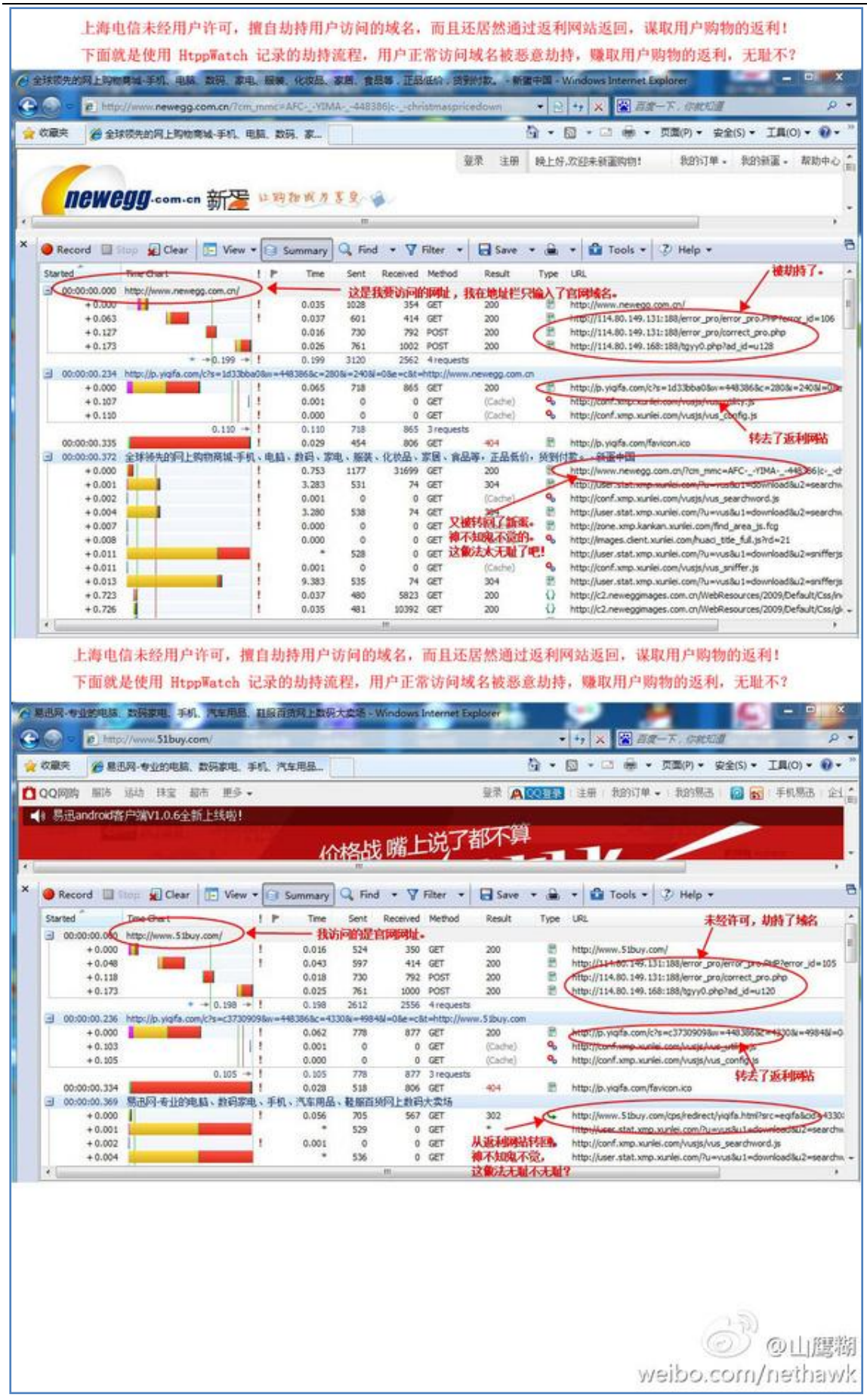
Http 劫持

对 CPS 模式的广告平台来说，会通过为每一个推广渠道创建唯一的 pid 来分辨不同渠道的效果。而 Http 劫持表现为不管用户从哪一个渠道、哪一条推广链接进入，pid 都会被替换为特定的 pid，剥夺了原本应该获得收益的渠道的收益。
例如淘宝客的推广，关于 pid 被规模性劫持的文章可以参考：
http://club.alimama.com/read-htm-tid-1554590.html
并非所有的基于 Http 的劫持都发生在电信运营商身上，只要能掌握用户互联网入口的人都可以做到。比如前段时间爆出的 360 浏览器自动替换淘宝客 pid 的事情，就是基于浏览器修改用户正常的推广渠道 pid。当然，和 360 相比，电信运营商还可以做到 DNS 劫持。

DNS 劫持

DNS 劫持相比 Http 劫持替换正常渠道 pid 的做法，还算只是利己、未损人。一般做法是将用户正常的访问进行多次跳转，跳转到返利站点推广渠道或者淘宝客的推广渠

道。比如微博上的一个截屏：


上海电信未经用户许可，擅自劫持用户访问的域名，而且还居然通过返利网站返回，谋取用户购物的返利！
下面就是使用 HttpWatch 记录的劫持流程，用户正常访问域名被恶意劫持，赚取用户购物的返利，无耻不？
这是我要访问的网址，我在地址栏只输入了官网域名。
被劫持了。
转去了返利网站
又被转回了新蛋。
神不知鬼不觉的。
这做法太无耻了吧！
上海电信未经用户许可，擅自劫持用户访问的域名，而且还居然通过返利网站返回，谋取用户购物的返利！
下面就是使用 HttpWatch 记录的劫持流程，用户正常访问域名被恶意劫持，赚取用户购物的返利，无耻不？
我访问的是官网网址。
未经许可，劫持了域名
转去了返利网站
从返利网站转回。
神不知鬼不觉，
这做法无耻不无耻？
@山鹰翔
weibo.com/nethawk

以上是本文的所有内容，对这些内容有哪些问题？留言一起讨论吧！CPC、CPA、CPA 广告的作弊方法肯定不至于此，对这方面有研究的朋友也可以赐教，让大家都能提高。

（完）