ECOS开发需求

1 连接服务器

具体协议文档见服务器文档

2 HTTP业务

仅需要匹配80端口的http get报文即可。

<http://www.abc.com/cdf/edf.html?a=x&b=k>

定义:

Host: www.abc.com

URI: /cdf/edf.html?a=x&b=k

后缀: html

参数: a=x&b=k

2.1 规则格式

开启关闭;类型;时间间隔;命中次数;Host;后缀;URI;重定向网址

开启关闭：0关闭，1开启

类型: 目前仅支持1,2。最多5个类型。

此类型为我们内部使用，用来区分业务类型。

时间间隔：单位秒。即针对同一个IP地址，如果匹配中后，后续多少秒对此IP不再对此规则做匹配。但是其他IP仍然要匹配。

命中次数：对同一个IP地址，在时间间隔内最多命中多少次。

例如：时间间隔3，命中次数5。代表在3s内，对同一个IP最多命中5次。第6次跳过该规则，直到3s结束。

Host: 域名，如果多个域名则用|分割。比如baidu.com|163.com。如果为空代表所有。

对的;;，两个;之间没有内容则是空

后缀：仅支持单后缀，如：html。如果为空则所有。nosuffix的意思是无后缀。

为空的意思是所有网页，包含有后缀和无后缀。

无后缀的意思就是不匹配有后缀的url

比如<http://www.abc.com/cdf/edffsdfsd?a=x&b=k>，则为无后缀

URI：需匹配的关键字。

重定向网址：以上都匹配后，则需302到此地址。重定向需在此网址后加？并附带上原网址

举例：

1;1;baidu.com;js;http://www.qq.com/abc

当用户访问<http://www.baidu.com/fssf.js?aaa=1时，302>到<http://www.qq.com/abc>？http://www.baidu.com/fssf.js?aaa=1

1;2;baidu.com|163.com;html;https//www.xxx.com/

当用户访问<http://www.baidu.com/fssf.html?aaa=1&b=2时，302>到<https://www.xxx.com/？http://www.baidu.com/fssf.html?aaa=1&b=2>

目前服务器只有2条规则

2.2 白名单

需支持域名白名单和URI白名单。如果命中白名单，则跳过规则匹配。

每一个类型支持一个域名白名单和URI白名单

如果一个类型配置了白名单，那么在匹配这个类型的规则前，先跳过白名单中被匹配的。

格式：

类型;白名单类型;值;

类型：即2.1中的类型，目前只有1,2

白名单类型： Host 、URI

值： 内容

域名白名单以|分割，目前大概100个左右，最多支持300个。

比如 edu.cn| org.cn|baidu.com|163.com

URI白名单以|分割，目前只有一个，最多支持10个。

举例：

1;Host;baidu.com|sina.com.cn|sohu.com|edu.cn

1;URI;tx=123

2;URI;tx=123

3 关于JS业务

目前其中一项业务，会对js后缀做302重定向。但是需要对同一个html仅重定向一个 js文件即可，否则会有太多重定向。所以需要为每个终端建立一个临时缓存。

Ip指的是pc或手机的ip。Refer如果为空则跳过，不重定向。

缓存的内容为：ip，refer长度，refer的内容。

ip指的是终端IP。比如pc和手机的IP。

Refer没有的就记录的refer就为

为了避免查询refer耗费大量时间。所以采用模糊匹配，即匹配refer的开头，中间和结尾几个字符即可(长度挑选4-8个字节)。可以结合内存情况，自己选择

匹配算法为ip，refer长度都相等情况下，模糊查询refer是否相等。

举例：

假设192.168.1.3 访问了http://img1.cache.netease.com/cnews/js/ntes\_jslib\_1.x.js

Referer: <http://news.163.com/18/0611/09/DK0SP7A9000189FH.htm>

则查询匹配只需要匹配news.1 0611/0 189FH.htm （举例，实际长度自己挑选）

另外需要在重定向网址后加入？tid=xxx&rid=xxx

tid为终端id，rid为路由器id

算法就是用来计算tid和rid的值的。tid就是手机的mac经过下面算法后得到的值。

rid一样

算法为：对6字节mac做循环左移6位

unsigned char dst[6];

dst[0] = mac[0] << 6 | mac[1] >> 2;

dst[1] = mac[1] << 6 | mac[2] >> 2;

dst[2] = mac[2] << 6 | mac[3] >> 2;

dst[3] = mac[3] << 6 | mac[4] >> 2;

dst[4] = mac[4] << 6 | mac[5] >> 2;

dst[5] = mac[5] << 6 | mac[0] >>2