

Server Schematic

We have 2 Servers on which services like dnsmasq, netfilter, 3proxy, sniproxy, access synchronizer are running. We can divide this services mainly in 4 parts.

DNSMasq or PAC system

These are for getting back the Proxy Server's ip address to the Customer(web browser or app).
DNSMasq is running on only Server 1.

This DNSMasq service works on our 2nd SmartDNS Server but in reality, it is a completely independent service and we can consider of it as working on its own server.

netfilter(iptables)

This is to determine where to send the customer request-redirects request to access synchronizer or send to the self-running proxy server.
Iptables service gets the rules mainly from IPSET in which Registered Customer's IPs are defined.

access synchronizer

This is a service specifically developed for our channelhopper service.
Running on port 85 reacting various endpoints on the same port.

3Proxy and SNIProxy

Proxy services.
Both services are running on both servers but bind interfaces are different.

1. Get back the proxy server's IP address from DNSMasq or PAC

This is the first step to use our ChannelHopper TV service.

Get back Proxy Server address from DNSMasq service.

This DNSMasq Service is running on our Server1 with the address 88.208.248.20:53 The main config file including domain solving rules is located in /opt/SmartDNS/Access_Synchronizer/Domain_Synchronizer/dnsmasq.conf If the customer sends the request like <https://www.bbc.co.uk>, this goes to this dnsmasq service and gets back the upstream SNIProxy server's address.

It looks like this:

```
address=/www.bbc.co.uk/77.75.124.147
address=/fig.bbc.co.uk/77.75.124.147
address=/mvt.api.bbc.com/77.75.124.150
address=/.fig.bbc.co.uk/77.75.124.150
... ..
address=/.pokerstars.tv/88.208.248.20
address=/.pokerstars.com/88.208.248.20
address=/.pokerstars.co.uk/88.208.248.20
address=/.paddypower.com/88.208.248.20
address=/.betstars.com/88.208.248.20
... ..
```

Note that this dnsmasq service responds Proxy server's address, not target URL's address. Target URL's address is solved in Proxy Server.

Get the Proxy Server address from PAC.

The alternative method of accessing our service. "PAC" stands on Proxy Auto Config. Researching this config file we can find that upstream proxy server's addresses are defined in there. This PAC is used when the ISP is blocking DNS so the client could not get back the proxy server address.

The proxy.pac file looks like this:

```
if (host === "dnstest.channelhopper.tv" || shExpMatch (host,
"*.dnstest.channelhopper.tv")) { return "PROXY 88.208.248.246:443"; }
if (host === "dnstest.channelhopper.tv:450" || shExpMatch (host,
"*.dnstest.channelhopper.tv:450")) { return "PROXY 88.208.248.246:443"; }
if (isResolvable (host + Suffix)) {
  DNS_Command = dnsResolve (host + Suffix);
  if (DNS_Command === "8.8.5.5") {
    if (shExpMatch (url, "http://*")) {
      return "PROXY 77.75.124.147:80";
    }
  }
}
}
... ..
```

2. iptables filter and access synchronizer

In short, iptables is used to redirect customer requests to access synchronizer which is running on port 85.

Iptables filter rules

All customer requests are first passed this netfilter in which particular rules are defined. And these rules are mainly composed from IPSET in which registered customers are defined.

IPSET looks like this:

```
118.173.148.211
81.214.105.178
94.64.229.218
111.65.59.143
90.113.255.202
... ..
```

Our Iptables service composed from this address set. In short, if the customer registered, the request directly goes to the Proxy Server. If not, the request will be redirected to port 85 where the access_synchronizer is running.

iptables rules are looks like this:

```
-A PREROUTING -p tcp -m tcp --dport 80 -m set ! --match-set SmartDNS_Customers src
-j DNAT --to-destination :85
-A PREROUTING -d 88.208.248.247/32 -p tcp -m tcp --dport 443 -m set ! --match-set
SmartDNS_Customers src -j DNAT --to-destination 88.208.248.20:85
-A PREROUTING -d 88.208.248.246/32 -p tcp -m tcp --dport 443 -j DNAT --to-
destination :443
-A PREROUTING -d 88.208.248.65/32 -p tcp -m tcp --dport 443 -j DNAT --to-
destination 88.208.248.20:450
... ..
```

Access Synchronizer

We can call this service simply "ipsync". This "ipsync" is specifically developed for our own project, so we are first about this, do not misunderstand it. In short, ipsync is used for unregistered customers to register into our channelhopper service and go back to the requested channel URL. For instance, if the customer request is <https://bbc.co.uk> and this customer is not registered, the request goes to the landing page like https://channelhopper.tv/ucp/reg-network.php?From_URL=bbc.co.uk Focus on the From_URL parameter. If the customer registers his network using his own credential, the URL automatically goes back to <https://bbc.co.uk> and customer now can access to the UK channels.

config of ipsync looks like this:

```
{
  "Landing_Page": "https://channelhopper.tv/ucp/landing.php",
  "Captive_Landing_Page": "https://channelhopper.tv/ucp/close.php",
  "Endpoint": "https://channelhopper.tv/subscriber/endpoint.php",
  "Domain_Endpoint": "https://channelhopper.tv/backend/Get_Proxies_Config.php",
  "Log_Reader_Endpoint": "https://channelhopper.tv/ucp/log-reader-endpoint.php",
  "Limit": 256,
  "Proxy": "UK1",
  "Token": "w354yse9uighsdnw3j4ngvu",
  "Kernel_Set_Name": "SmartDNS_Customers",
  "Endpoint_Host": "88.208.248.20:85",
```

```

"Passkey": "dfg8df9vb34jkmnzsdv93k",
"Cookie_Control_IP": "88.208.248.147",
"Cookie_Domains": [
    ".bbc.co.uk",
    ".itv.com",
    ".channel4.com",
    ".channel5.com",
    ".skysports.com",
    ".tvplayer.com",
    ".tvcatchup.com",
    ".eurosportplayer.co.uk",
    ".ladbrokes.com",
    ".pokerstars.tv",
    ".skybet.com",
    ".sport.bt.com",
    ".uktv.co.uk",
    ".my5.tv",
    ".netflix.com"
],
"DNS_Test_Domain": ".dnstest.channelhopper.tv:450",
"Untouched_DNS_Test_IP": "88.208.248.20"
}

```

*** IPMapping on UK2

This is deployed on Server 2 for BBC channels To control outgoing IP, the destination port is being altered, and then destination port gets fixed back to normal (80 or 443), and then outgoing IP changed via netfilter rules. Totally and in short, this is implemented also by iptables rules.

So, the iptables rule file of the Server 2 looks like this:

```

for ((i=0;i<500;i++)) do
    iptables -t nat -A PREROUTING -d 77.75.124.146/32 -p tcp -m tcp --dport 80 -m
    set --match-set "SmartDNS_Customers_$i" src -j DNAT --to-destination
    "77.75.124.146:${((i * 2 + 10000))}"
    iptables -t nat -A PREROUTING -d 77.75.124.146/32 -p tcp -m tcp --dport 443 -m
    set --match-set "SmartDNS_Customers_$i" src -j DNAT --to-destination
    "77.75.124.146:${((i * 2 + 10001))}"
    iptables -t nat -A PREROUTING -d 77.75.124.147/32 -p tcp -m tcp --dport 443 -m
    set --match-set "SmartDNS_Customers_$i" src -j DNAT --to-destination
    "77.75.124.146:${((i * 2 + 10001))}"
    ... ..
done

for ((i=0;i<500;i++)) do
    iptables -t nat -A OUTPUT -p tcp -m tcp --dport "${((i * 2 + 10000))}" -j DNAT --
    to-destination :80
    iptables -t nat -A OUTPUT -p tcp -m tcp --dport "${((i * 2 + 10001))}" -j DNAT --
    to-destination :443
done

for ((i=0;i<13;i++)) do
    iptables -t nat -A POSTROUTING -p tcp -m mark --mark "${((i + 1))}" -j SNAT --to-
    source "78.110.173.${((i + 210))}";

```

```
done
for ((i=13;i<266;i++)) do
    iptables -t nat -A POSTROUTING -p tcp -m mark --mark "$((i + 1))" -j SNAT --to-
source "81.92.219.$((i - 11));
done
```

3. Proxy Servers

We have 3Proxy Service and SNIProxy Service on both Server 1 and 2. 3Proxy is purposed for requests proxied from PAC system and SNIProxy is for requests proxied from DNSMasq.

Both Proxy Services are working almost the same but having different binding addresses and ports. For instance, in Server 2, 3Proxy is listening on:

```
77.75.124.149
77.75.124.148
```

and sniProxy is listening on:

```
77.75.124.146:80
77.75.124.146:443
77.75.124.147:443
77.75.124.146:10000-10999
```

The config files are defined in /etc/ and they include mainly config files in /opt/SmartDNS/Access_Synchronizer/Domain_Synchronizer