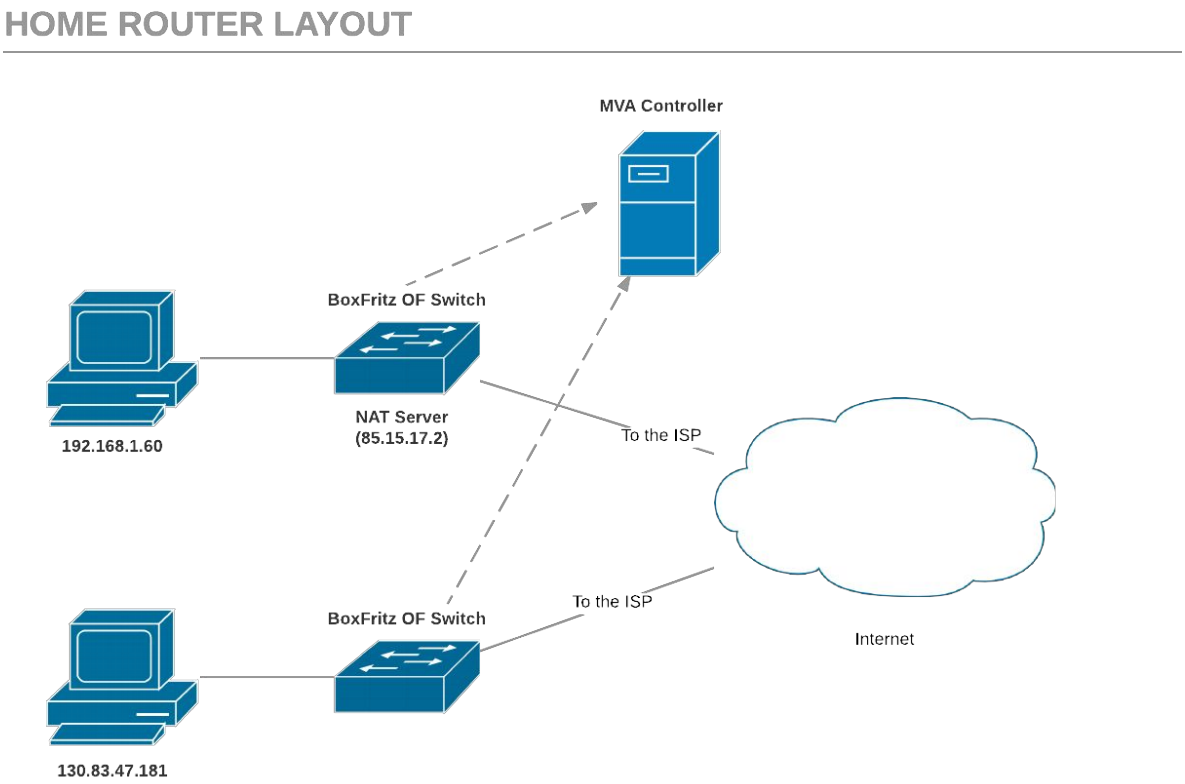
Solution 1:
The idea behind this solution is to forward each DNS request to the controller (and at the same time passing it through to get DNS response). The controller then checks if the query domain matches the list of blocked domains for that region (region can be detected by IP address). If this is the case, the controller picks a proxy from the list and installs a flow rule which tells the switch to match packets for the domain IP address, and rewrite the destination IP address with the proxy address.



Suppose that hosts in the picture above want to access an OTT provider which is blocked in their country. The host sitting behind NAT wants to access www.vimeo.com which is blocked in Iran and the other host wants to access www.hulu.com which is blocked in Germany.
Here is the rules installed on the upper switch:

Flow Table of the upper switch before a DNS request for vimeo

Incoming Port	Ethernet Type	IP Src	IP Dest	IP Proto.	Source port	Dest port	Action
*	0x0800	*	*	0x11	*	53	Packet-In; pass-through

After a DNS request for vimeo.com is forwarded to the controller, the controller goes into the application layer payload and checks if it is for a blocked OTT provider (the whole list). Then it checks region of source IP based on the lists like the one here, to see if this is blocked in the source country. Here: Iran. If this is the case, it picks a proxy server suitable for accessing vimeo.com from Iran, and installs a rule on that switch which says to rewrite destination IP address of all packets destined to vimeo.com (detected by IP address of course) with the proxy server's IP address. This way we ensure transparent redirection of all requests for vimeo.com to the proxy server.

Flow Table of the upper switch after a DNS request for vimeo

Incoming Port	Ethernet Type	IP Src	IP Dest	IP Proto.	Source port	Dest port	Action
*	0x0800	*	*	0x11	*	53	Packet-In; pass-through
*	0x0800	*	vimeo.com's IP address	0x06, 0x011	*	*	Rewrite dest IP to Proxy's address; forward

The same procedure happens for the lower switch.

This solution is not scalable as it forwards each DNS packet to the controller for further investigations. It also keeps the controller busy, even

when it has already installed a proper rule for requests to a block OTT, like Vimeo in this case.

Solution 2:
In this solution, in priori we install OF rules that redirect packets destined for blocked OTT services to the proxy servers according to the country. Switches are initialized with a simple rule which says forward all the incoming packets to the controller. The controller then decides based on source IP address location, which OTTs are blocked in the country the switch is deployed in. Then it installs rules that say rewrite packets destined for the blocked OTTs IP address with the appropriate proxy server.

Consider the previous layout. Here is the flow table for the lower switch (in this example, residing in Germany):

Initial Flow Table of the lower switch

Incoming Port	Ethernet Type	IP Src	IP Dest	IP Proto.	Source port	Dest port	Action
1, 2, 3	*	*	*	*	*	*	Packet-In

We assume that all BoxFritz switches have incoming ports numbered 1 to 3. That is, no switch has more than 3 incoming ports and no output port is numbered in this range.
This rule is here because at first, the controller has no idea in which country this switch has been installed and is operating. So it should get one packet and determine the locality by IP address.
After this, the controller will delete this rule from the switch and install as many rules as there are blocked OTTs in that country. Here: Germany. These rules say that rewrite destination IP address of packets destined for blocked OTTs with proxy server’s address.

Flow Table after the first packet goes to the controller

Incoming Port	Ethernet Type	IP Src	IP Dest	IP Proto.	Source port	Dest port	Action
*	0x0800	*	youtube.com's IP address	0x06, 0x011	*	*	Rewrite dest IP to Proxy's address; then pass through
*	0x0800	*	hulu.com's IP address	0x06, 0x011	*	*	Rewrite dest IP to Proxy's address; then pass through

We have assumed that only Youtube and Hulu are blocked in Germany for the sake of this example.

This solution is more scalable than the previous because controller is not involved for each domain name lookup. But it requires more space in TCAM as there would be many blocked OTTs, hence many OF rules. Hard timeout for rules can be deployed to renew proxy server's IP addresses when each rule expires. This comes handy when a proxy server dies in order to keep the traffic going.