

<https://github.com/ProtonVPN/android-app/issues/37>
“[feature request] allow ip range in split tunneling #37”

In case it is of help to Proton. Very roughly, a possibility for allowing the use of sub nets is to...

1. Change editIP edit text in layout dialog_split_tunneling.xml to allow users to specify both IP addresses and sub nets in CIDR format (eg 192.168.1.0/24).
2. Change editIP.addTextChangedListener in class IpDialog so that the listener verifies that the input (editable.toString()) is a valid address *or* sub net, not just an individual address.
3. Change class UserData: 1) add a new attribute of type List just like splitTunnelIpAddresses, lets call it splitTunnelIps, to hold the addresses *and* sub nets the user introduces via IpDialog. 2) Change current splitTunnelIpAddresses attribute so that now it is the result of the explosion of splitTunnelIps, ie it represents all the individual Ip addresses that derive from the addresses and sub nets in splitTunnelIps.
4. Change class IpAdapter so that the data displayed by the recycler view be not UserData.splitTunnelIpAddresses, but the new attribute, UserData.splitTunnelIps.

This would make it possible for users to work with individual addresses and sub nets while requiring code changes only in the split tunneling feature. Still, it may be possible to use sub nets in the VPNProfile classes of IKEv2 and OpenVPN (the classes that take the excluded Ip addresses from UserData so that they are actually excluded from the VPN). Indeed, according to its documentation, IKEv2 seems to support sub nets in the profile [directly](#).

Additionally, some new decisions would have to be made with regards to Ip address/sub net inputs. If, presently, a valid input is a correctly formatted Ip address that is not already excluded, with this change, a valid input may be: a correctly formatted Ip address that is not already excluded (by means of the same address, *or* a sub net), *and* a correctly formatted sub net that does not overlap with another, already excluded, sub net. Additionally, if a sub net includes, in their entirety, an already excluded address or sub net, a dialog could be presented to the user: is it Okay to delete those addresses and sub nets, to add the new sub net? This would ensure that the exclusion list does not contain overlaps.

By the way, for doing Ip address and range validations as mentioned above and in points 1, 2, library [IPAddress](#) could help. I wrote a small [class](#) that makes use of it to make some of those validations, just to provide a basic sample.