There is no Layer2 option in WireGuard

Last update: **07.06.2020 21:20**

As stated in the first sentence of “WireGuard: NextGeneration Kernel Network Tunnel” [1]

> WireGuard is a secure network tunnel, operating at layer 3, implemented...

All questions on StackExchange regarding WireGuard and bridging, broadcast traffic are answered pretty quickly: “It’s layer 3”. Period… Except maybe in the future multicast and IPv6 link local addresses, eventually.

Over the past years, I migrated lots of VPNs to WireGuard. Since it has found its way into the Linux Kernel (Premier Commit) it has become my first option to choose, when it comes to VPN.

Now in 99.5% of the cases, implementing VPN at layer 3 is the required solution. In .5% there is no way around at least a bit layer 2. Thinking of OpenVPN TAP interfaces I was looking for a way to close the gap. A good friend of mine, @kpanic, a pretty busy guy in the Freifunk FF3L Community gave me a good hint: “Ever thought about GRETAP?”.

GRE stands for *Generic Routing Encapsulation* and is defined in RFC 2784. GRE interfaces operate on layer 3. In Linux there is a thing called “gretap”. It’s a GRE tunnel based TAP interface. And because it is a TAP, it simulates a link layer device and therefore carrying Ethernet frames. I’ve not found good documentation about, but in the kernel source.

Compared to GRE, in GRETAP there is a “Inner Ethernet Header”. Keep in mind, this all adds up on our network overhead.

The case

Two sites, both with a private //24 IPv4 network, both Debian 10 boxes, installed on a APU2 based router with three gigabit ethernet NICs installed.

The plan is, that the third NIC (enp3s0) on router A gets bridged transparently to enp2s0 on router B. Giving anything plugged to enp3s0 on router A the full Layer 2 experience as directly connected to the Switch on Site B

On both sites: A WAN/Cable modem on enp1s0, network devices firewalled, behind a switch on enp2s0. Let’s call them Router A & B / Site A & B.\

- Site Network A: 192.168.178.0/24
- Site Network B: 192.168.92.0/24

WireGuard Config on Router A

```
[Interface]
PrivateKey = PrivateKeyRouterA
Address = 10.15.14.2/24

[Peer]
PublicKey = PublicKeyRouterB
AllowedIPs = 192.168.92.0/24,10.15.14.0/24
Endpoint = MyRemoteServer.com:4900
PersistentKeepalive = 15
```

WireguardConfig on Router B

```
[Interface]
Address = 10.15.14.1/24
ListenPort = 4900
PrivateKey = PrivatekeyRouterB

[Peer]
# APU2(Static Route)
PublicKey = PublicKeyRouterA
AllowedIPs = 10.15.14.2/32,192.168.178.0/24
```

How I think it could be solved

There are some prerequisites to be in place to make the bridge setup work

Ensured, IPv6 forwarding is enabled on both sites (This should be given on a router)

```
-> # cat /etc/sysctl.conf | grep net.ipv4.ip_forw
net.ipv4.ip_forward=1
```

bridge-utils installed on both sites

```
-> #  apt install bridge-utils
```

MSS clamping on both sides in FORWARD (This is the *quick** method of getting MSS right)

```
-> #  iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu;
```

and *br_netfilter* gets loaded on boot. If not, your iptables forward rules will not work for the resulting bridge.

```
-> # cat /etc/modules-load.d/brnetfilter.conf
br_netfilter
```

Time to create a bridge interface on both sides.

Router A: In */etc/network/interfaces* we create a bridge and connect enp3s0 to it. There is no IP configuration on enp3s0 nor on the bridge.

```
allow-hotplug enp3s0
iface enp3s0 inet manual

auto br0
# Bridge setup
iface br0 inet manual
    bridge_ports enp3s0
```

Router B: In */etc/network/interfaces* we create a bridge and connect enp2s0 to it on boot. The bridge is configured to static internal network 192.168.92.1/24 IP on enp2s0 the address set to manual with no address

```
allow-hotplug enp2s0
iface enp2s0 inet manual

auto br0
iface br0 inet static
    address 192.168.92.1
    netmask 255.255.255.0
    bridge_ports enp2s0
```

Append PostUp & Down scripts on wg0.conf at Router A

```
[Interface]
PrivateKey = PrivateKeyRouterA
Address = 10.15.14.2/24
PostUp = ip link add name gretap1 type gretap local 192.168.178.1 remote 192.168.92.1
PostUp = ip link set gretap1 up
PostUp = ip link set gretap1 master br0
PostDown = ip link del gretap1

[Peer]
# SiteB (Static Route)
PublicKey = PublicKeyRouterB
AllowedIPs = 192.168.92.0/24,10.15.14.0/24
Endpoint = MyRemoteServer.com:4900
PersistentKeepalive = 15
```

Append PostUp & Down scripts on wg0.conf at Router B

```
[Interface]
Address = 10.15.14.1/24
ListenPort = 4900
PrivateKey = PrivatekeyRouterB
PostUp = ip link add name gretap1 type gretap local 192.168.92.1 remote 192.168.178.1
PostUp = ip link set gretap1 up
```

```
PostUp = ip link set gretap1 master br0
PostDown = ip link del gretap1

[Peer]
# SiteA (Static Route)
PublicKey = PublicKeyRouterA
AllowedIPs = 10.15.14.2/32,192.168.178.0/24
```

Sources

- Title image: https://www.wireguard.com/img/wireguard.svg
- [1]Paper “WireGuard: NextGeneration Kernel Network Tunnel” Jason A. Donenfeld, https://www.wireguard.com/papers/wireguard.pdf