Building an SD-WAN solution based on Wireguard tunnels

Robin Douine
Context
Target
Expected benefits

- Reliability
- Deployment time
- Consumption-based pricing model
Technical objectives

• Support of several hundreds of Gbps
• Change routing dynamically based on the IP transit state
• Use of standard components
• Use of commodity hardware
Routingctl
Gatewan

- Using Serf to build a secure mesh with the remote Gatewans
- Exchanging the local prefixes with the remote Gatewans
- Creating Wireguard tunnels with the remote Gatewans for each IP transit provider
- Collecting metrics via tunnels probing (loss, latency, jitter)

https://www.serf.io
https://www.wireguard.com
Gatewan

• Using Serf to build a secure mesh with the remote Gatewans
• Exchanging the local prefixes with the remote Gatewans
• Creating Wireguard tunnels with the remote Gatewans on each IP transit provider
• Collecting metrics via tunnels probing (loss, latency, jitter)

https://www.serf.io
https://www.wireguard.com
id: 3755b299-afe6-4d63-bfe8-8181d44013de
 networks: 10.1.11.0/24, 10.176.0.0/14, 10.88.0.0/14, fd05::/16
meta:
  datacenter: sg1
  sdwan_local_transit: lumen
  sdwan_peer_transit: telstra
usage: capacity: 1.0 Gbps  rx_bps: 1.6 Mbps  tx_bps: 339 kbps
probe:
  source_addr: 10.13.76.135
  target_endpoint: 10.13.76.134:7117
  measurements: 10
  loss: 0.0%
  latency: 224.36043ms
  jitter: 186.54μs
  reorder: 0.0%
peer: gatewan09-sg1
state: Established
target_datacenter: sg1
interface: sdwan52
started_at: Mon, 01 Jan 0001 00:00:00 UTC
local_info:
  public_endpoint: [2620:100:a006::c]:6152
  local_address_v4: 10.13.76.135
  local_address_v6: fd05:1:0:2:5::87
  public_key: 92777e8837d9d9090adabd80e192dc6c225f200aa36428f069388b4fee60b6b484f
  echo_port: 7054
peer_info:
  public_endpoint: [2406:2600:a::1]:6215
  local_address_v4: 10.13.76.134
  local_address_v6: fd05:1:0:2:5::86
  public_key: d99d959796c9dc5b66c4e4b9cf5419c4098336ed2855b4f1715be9547a20d0f06
  echo_port: 7117
Gatewan

```
- id: 3755b299-afe6-4d63-bfe8-8181d4013de
  networks: 10.1.11.0/24, 10.176.0.0/14, 10.88.0.0/14, fd05::/16
  meta:
    datacenter: sg1
    sdwan_local_transit: lumen
    sdwan_peer_transit: telstra
  usage:
    capacity: 1.0 Gbps
    rx_bps: 1.6 Mbps
    tx_bps: 339 kbps
  probe:
    source_addr: 10.13.76.135
    target_endpoint: 10.13.76.134:7117
    measurements: 10
  loss: 0.0%
  latency: 224.36043ms
  jitter: 186.54µs
  reorder: 0.0%
  peer: gatewan09-sg1
  state: Established
  target_datacenter: sg1
  interface: sdwan52
  started_at: Mon, 01 Jan 0001 00:00:00 UTC
  local_info:
    public_endpoint: [2620:100:a006::c]:6152
    local_address_v4: 10.13.76.135
    local_address_v6: fd05:1:0:2::87
    public_key: 92777e8837d9090adadb80e192dc6c225f200aa36428f06938b4fee60bbcc484f
    echo_port: 7054
  peer_info:
    public_endpoint: [2406:2600:a::1]:6215
    local_address_v4: 10.13.76.134
    local_address_v6: fd05:1:0:2::86
    public_key: d99d959796c9dc5b66c4e4b9c5419c4098336ed2855b4f1715be9547a20df06
    echo_port: 7117
```
• How do we send the traffic to the Gateway?
• Use of VXLAN to encapsulate the traffic
Controller

- Calculating a tunnel preference according to a policy
- The policy uses the metrics gathered by the Gatewans
- Choosing tunnels until reach the required capacity
- Building orders and sending them to the bgpctl and the Gatewans

https://www.serf.io/
Bgpctl

• Provide dynamically the local prefixes to the controller
• Translate the controller's orders to BGP EVPN Route Type 5 (IP Prefix route)
• Based on GoBGP

https://osrg.github.io/gobgp/
Bgpctl

- Provide dynamically the local prefixes to the controller
- Translate the controller's orders to BGP EVPN Route Type 5 (IP Prefix route)
- Based on GoBGP

https://osrg.github.io/gobgp/
SD-WAN
The journey of the packets
The journey of the packets
The journey of the packets
The journey of the packets
The journey of the packets
The journey of the packets
The journey of the packets
Drawbacks

• Debugging is complex
• Tricky bootstrapping (chicken-and-egg problem)
• Vulnerable to DDoS
Next steps

• Improve the performance per tunnel
• Adding more parameters in the controller choice (e.g. IP transit interface usage)
Conclusion
Thank you