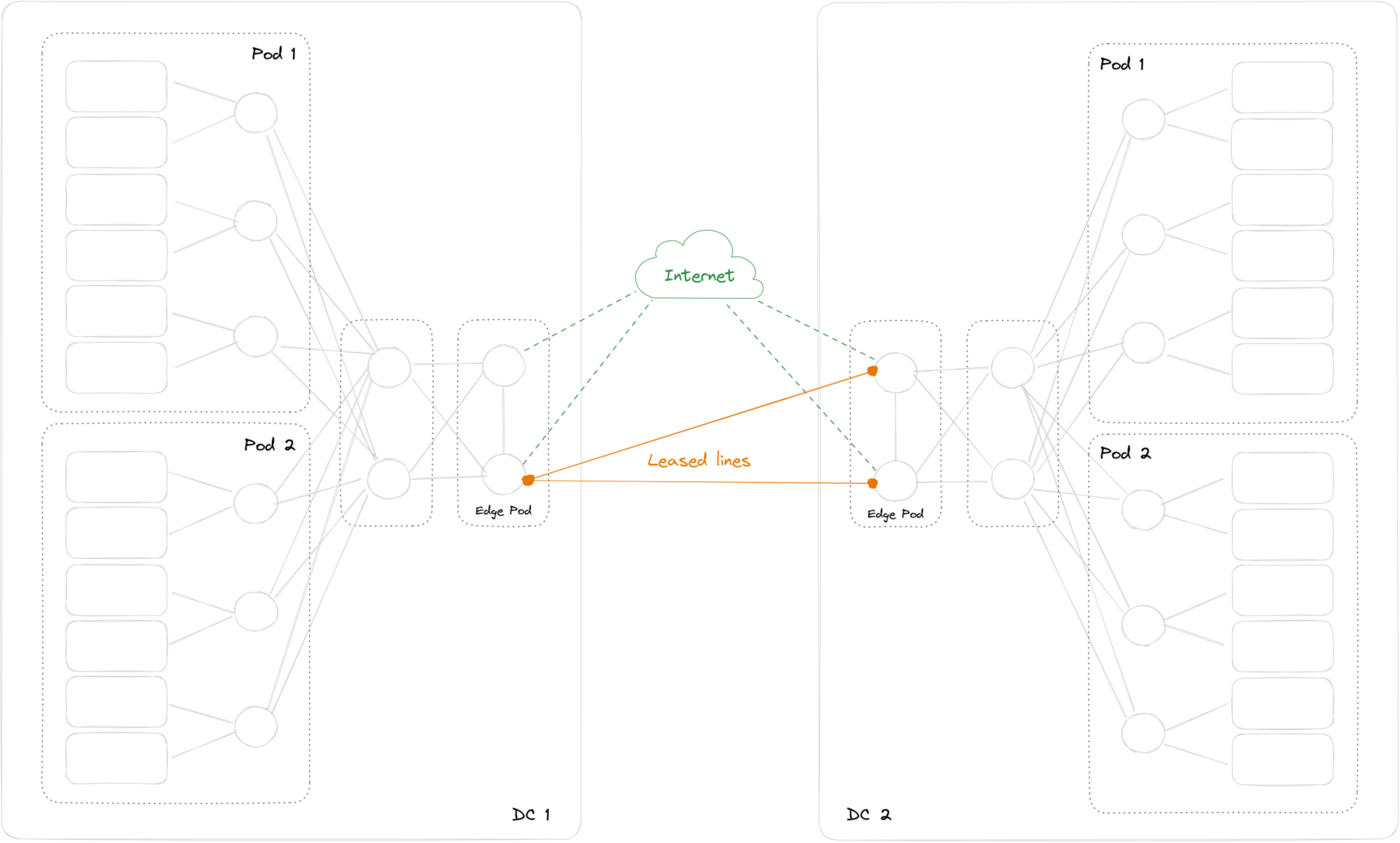
CRITEO

Building an SD-WAN solution based on Wireguard tunnels

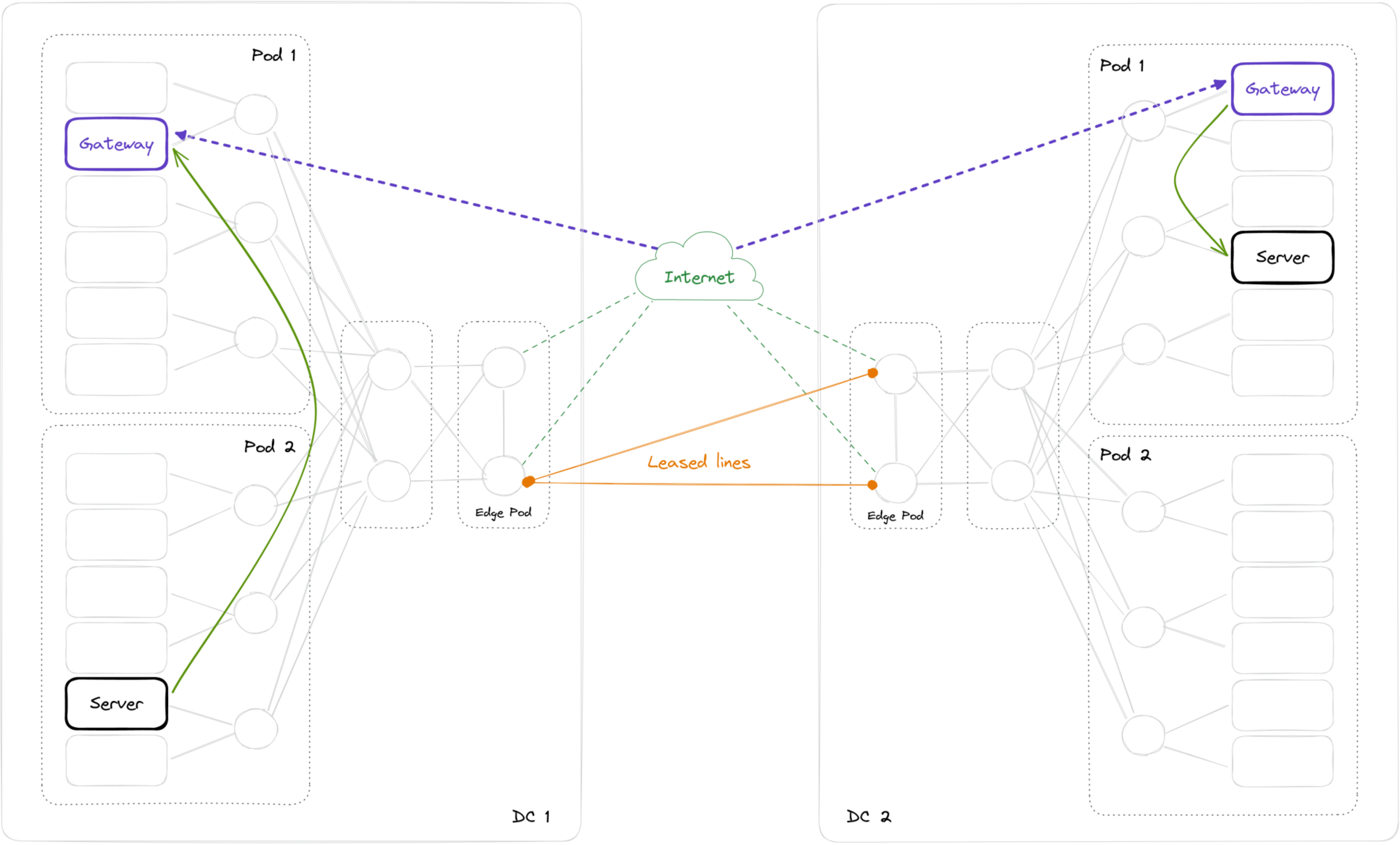
Robin Douine

Context



CRITEO

Target


Pod 1
Gateway
Pod 2
Server
Edge Pod
DC 1
Internet
Leased lines
Pod 1
Gateway
Server
Edge Pod
Pod 2
DC 2

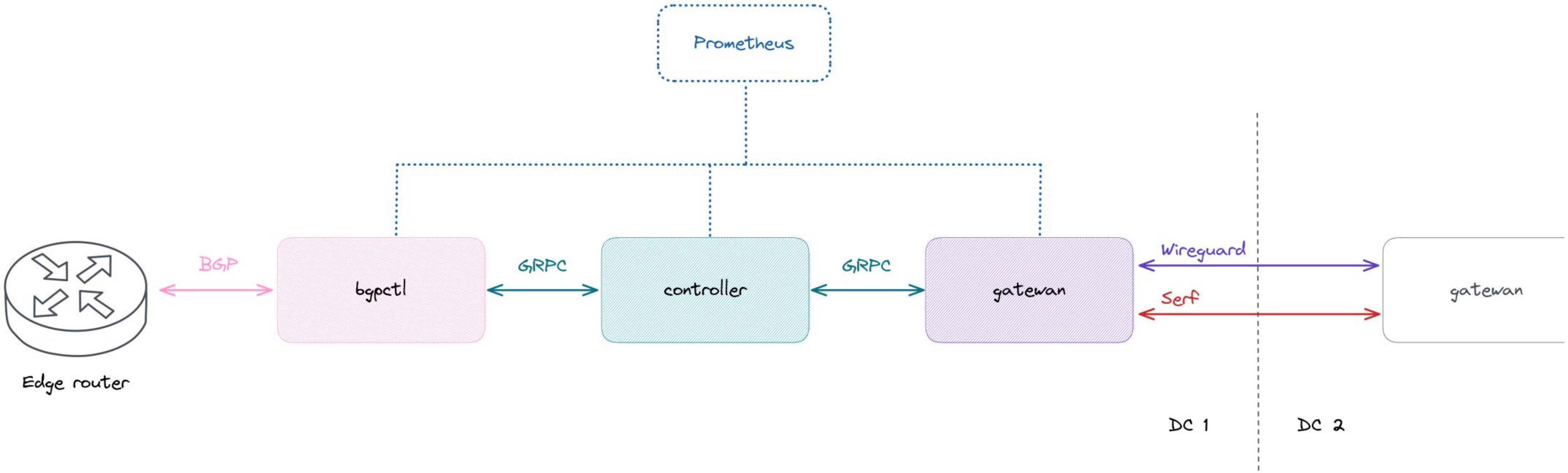
Expected benefits

- Reliability
- Deployment time
- Consumption-based pricing model

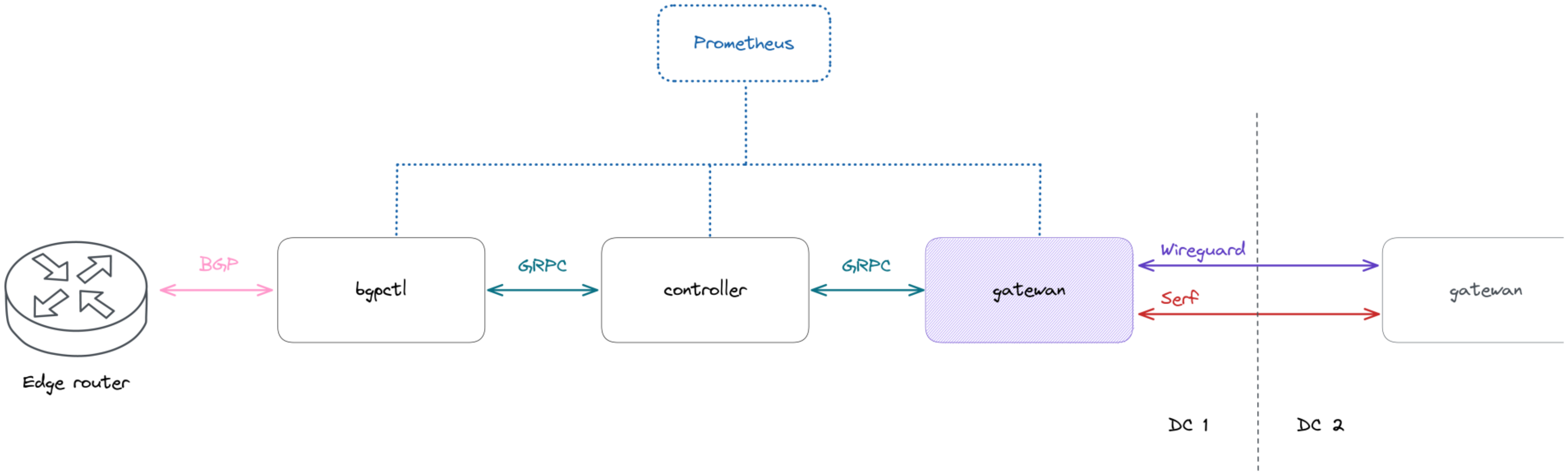
Technical objectives

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

Routingctl


Prometheus
Edge router
BGP
bgpctl
GRPC
controller
GRPC
gatewan
Wireguard
Serf
gatewan
DC 1
DC 2

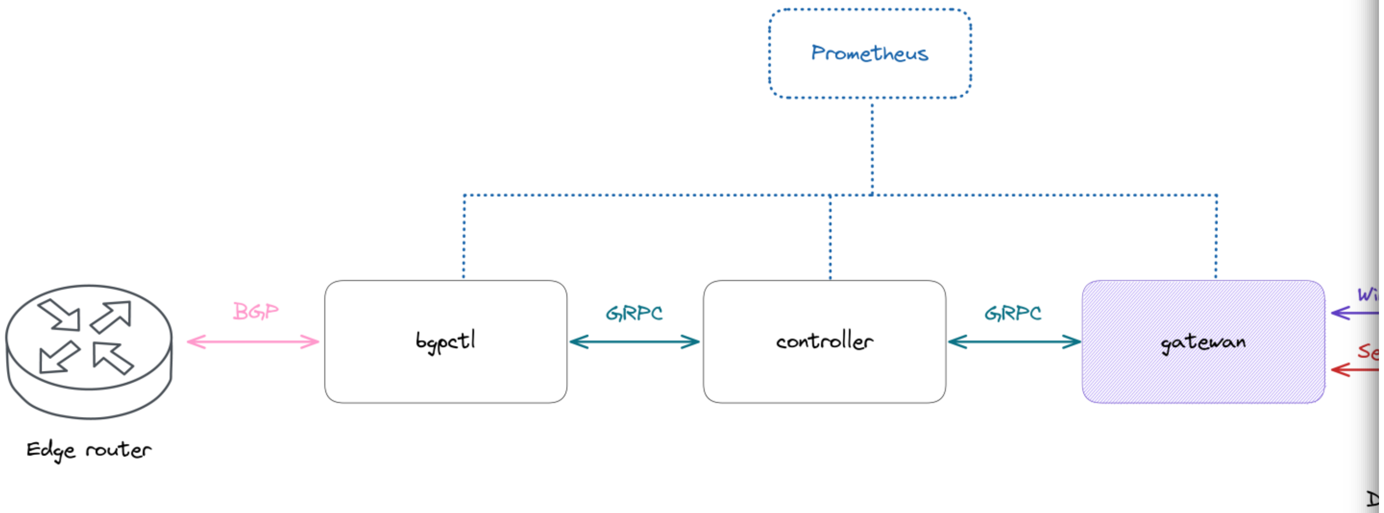
Gatewan

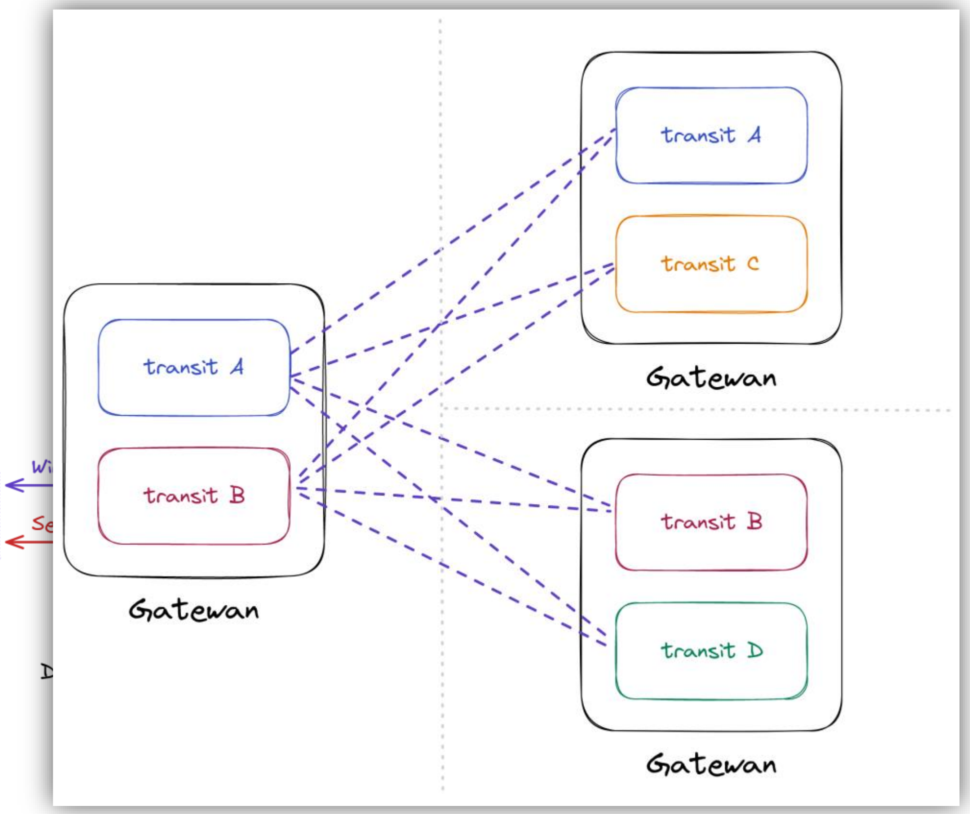


- Using Serf to build a secure mesh with the remote Gatewans
- Exchanging the local prefixes with the remote Gatewans
- Creating Wireguard tunnels with the remote Gatewans for each IP transit provider
- Collecting metrics via tunnels probing (loss, latency, jitter)

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

Gatewan

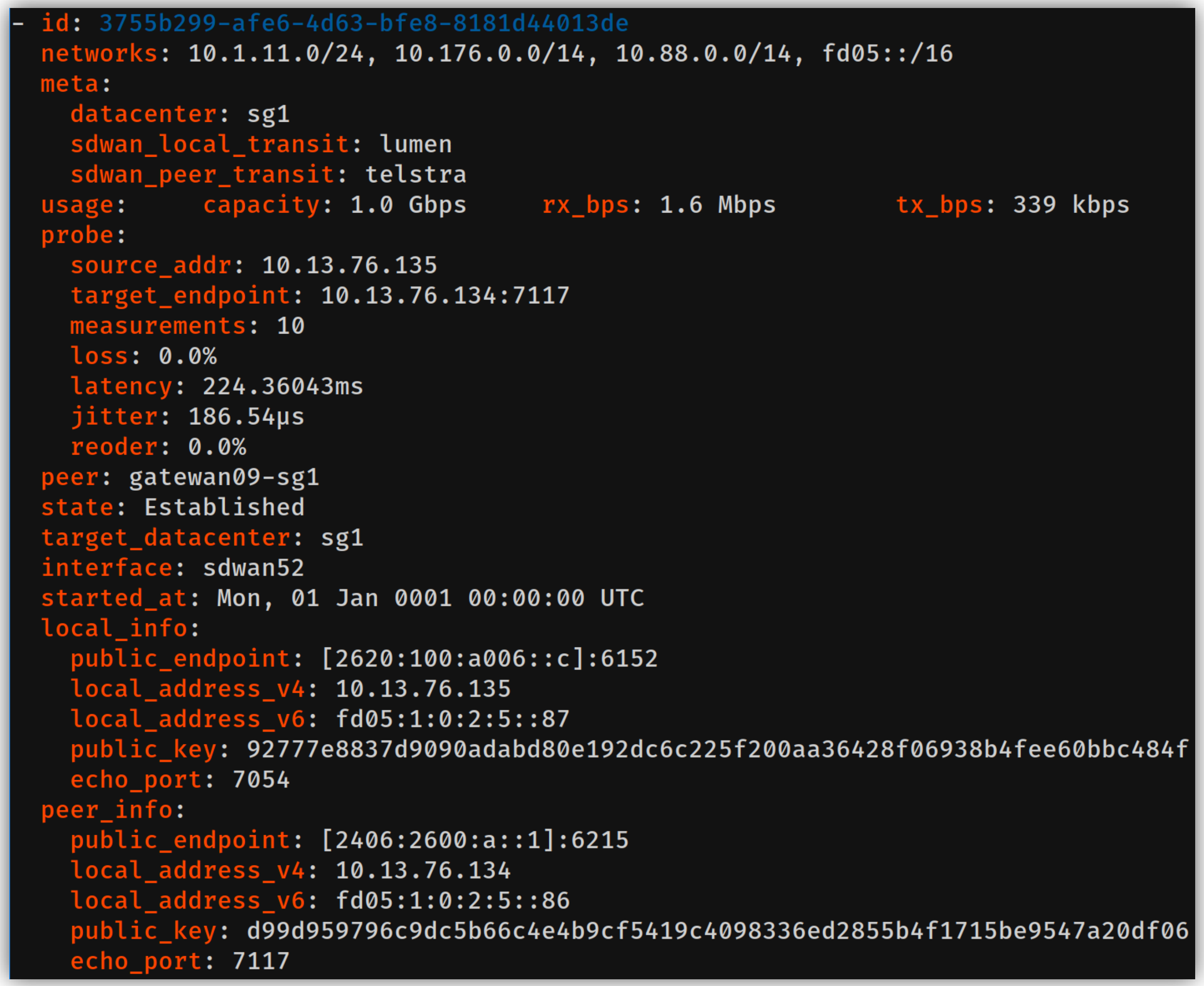




- Using Serf to build a secure mesh with the remote Gatewans
- Exchanging the local prefixes with the remote Gatewans
- Creating Wireguard tunnels with the remote Gatewans on each IP transit provider
- Collecting metrics via tunnels probing (loss, latency, jitter)

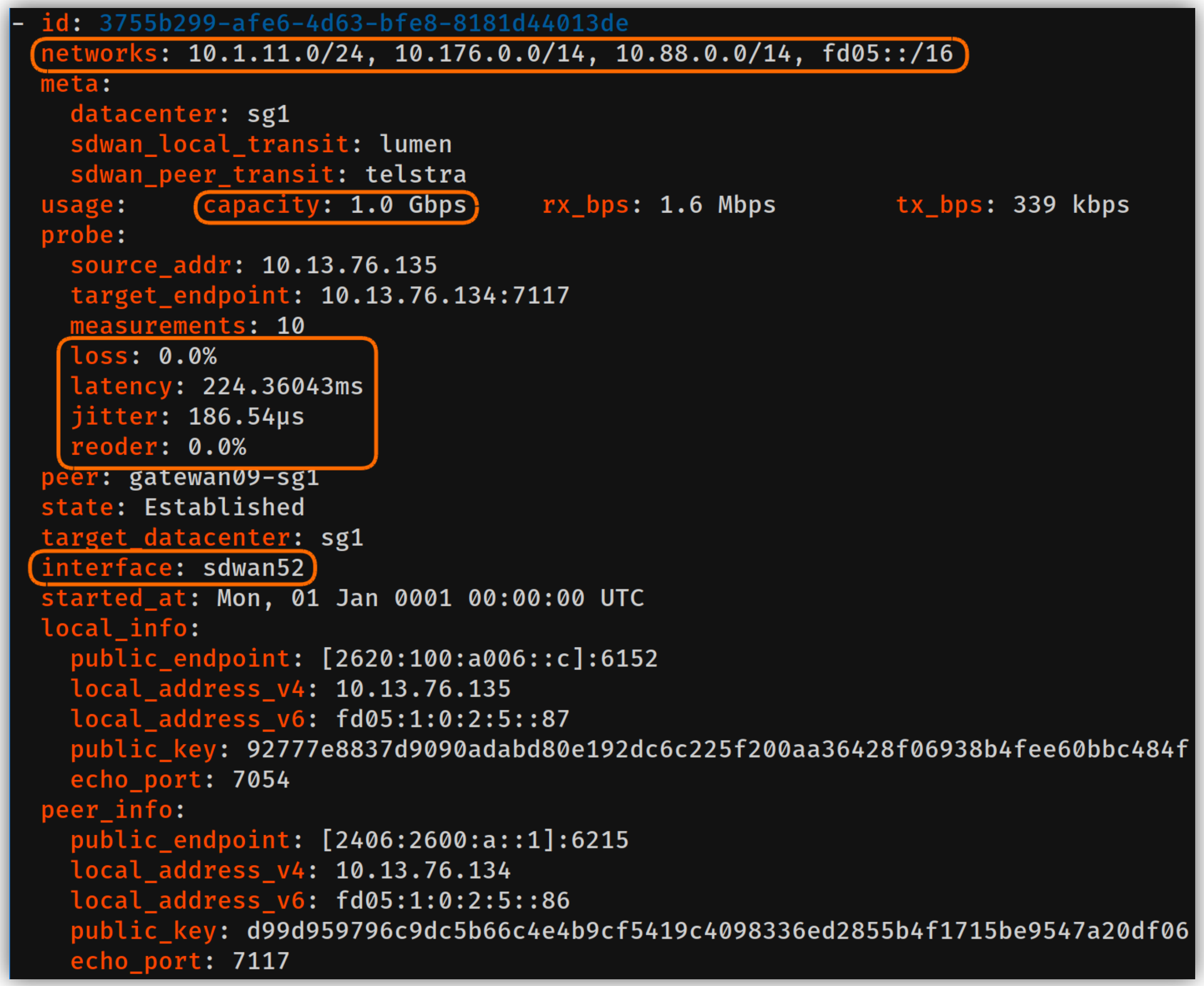
https://www.serf.io
https://www.wireguard.com

Gatewan



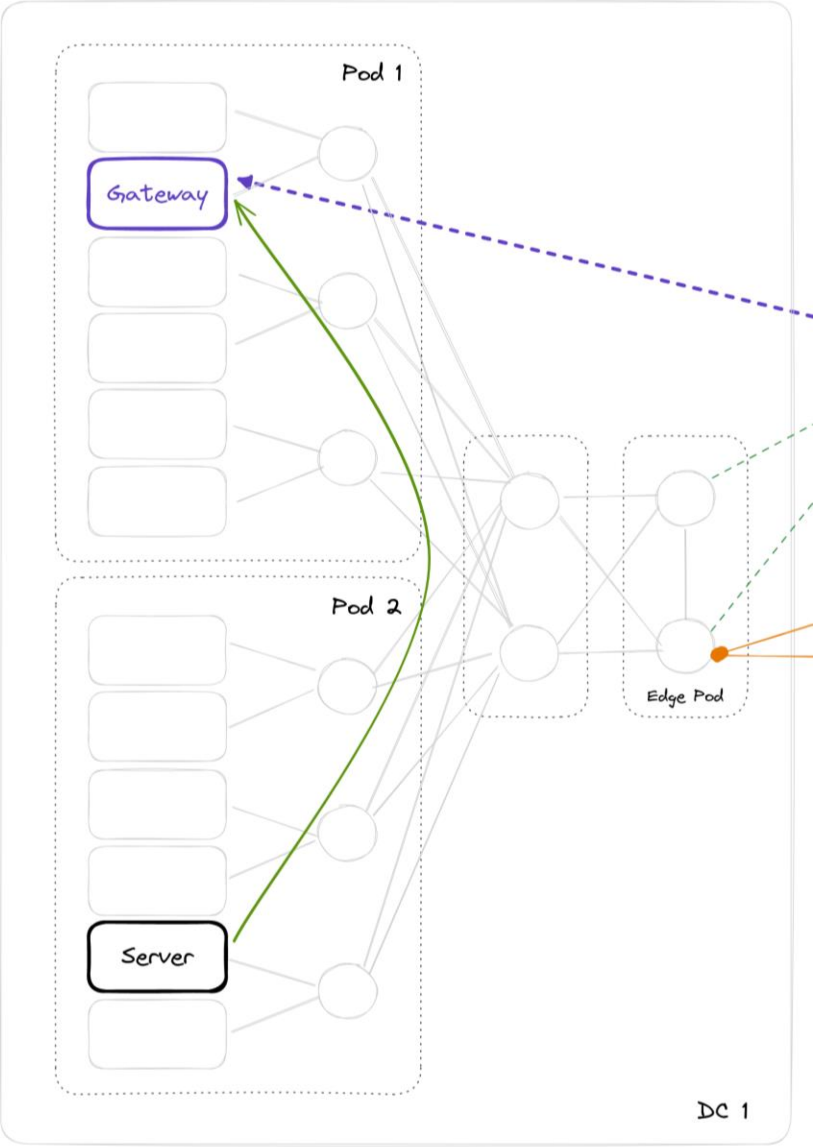
```
- id: 3755b299-afe6-4d63-bfe8-8181d44013de
  networks: 10.1.11.0/24, 10.176.0.0/14, 10.88.0.0/14, fd05::/16
  meta:
    datacenter: sg1
    sdwan_local_transit: lumen
    sdwan_peer_transit: telstra
  usage:     capacity: 1.0 Gbps     rx_bps: 1.6 Mbps        tx_bps: 339 kbps
  probe:
    source_addr: 10.13.76.135
    target_endpoint: 10.13.76.134:7117
    measurements: 10
    loss: 0.0%
    latency: 224.36043ms
    jitter: 186.54µs
    reoder: 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: 92777e8837d9090adabd80e192dc6c225f200aa36428f06938b4fee60bbc484f
    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: d99d959796c9dc5b66c4e4b9cf5419c4098336ed2855b4f1715be9547a20df06
    echo_port: 7117
```

Gatewan



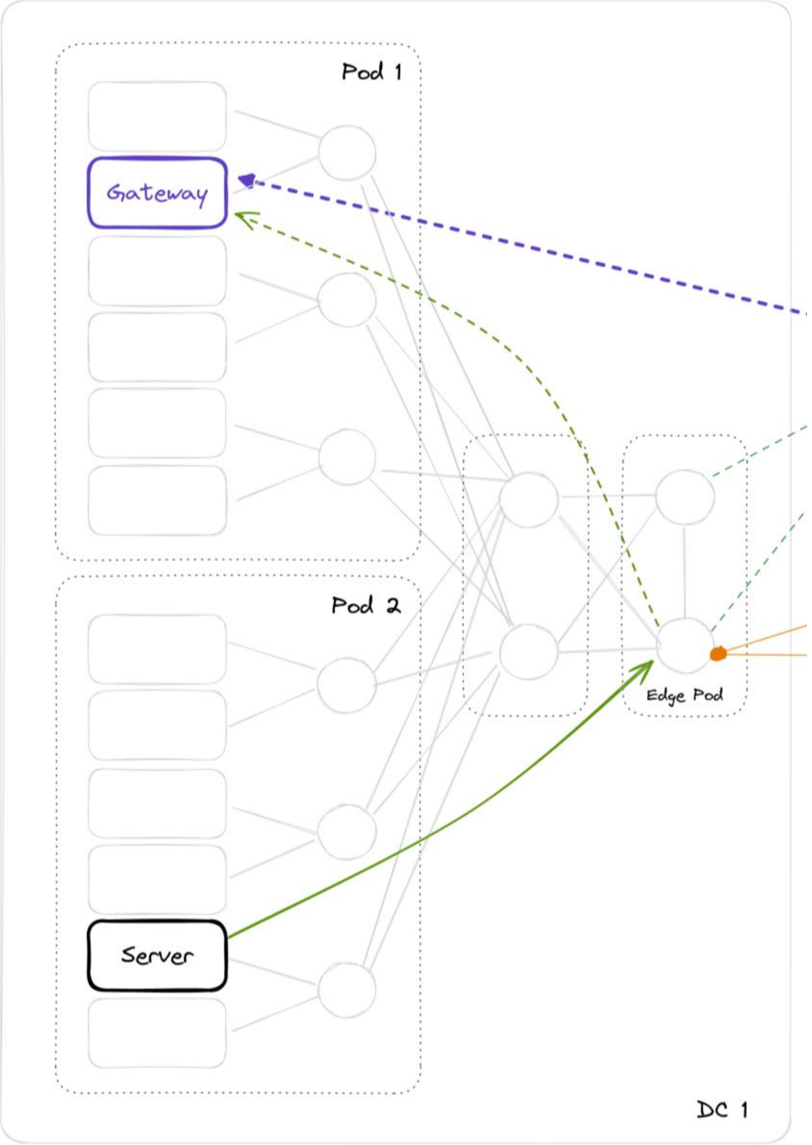
```
- id: 3755b299-afe6-4d63-bfe8-8181d44013de
  networks: 10.1.11.0/24, 10.176.0.0/14, 10.88.0.0/14, fd05::/16
  meta:
    datacenter: sg1
    sdwan_local_transit: lumen
    sdwan_peer_transit: telstra
  usage:     capacity: 1.0 Gbps      rx_bps: 1.6 Mbps         tx_bps: 339 kbps
  probe:
    source_addr: 10.13.76.135
    target_endpoint: 10.13.76.134:7117
    measurements: 10
    loss: 0.0%
    latency: 224.36043ms
    jitter: 186.54µs
    reoder: 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: 92777e8837d9090adabd80e192dc6c225f200aa36428f06938b4fee60bbc484f
    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: d99d959796c9dc5b66c4e4b9cf5419c4098336ed2855b4f1715be9547a20df06
    echo_port: 7117
```

Gatewan



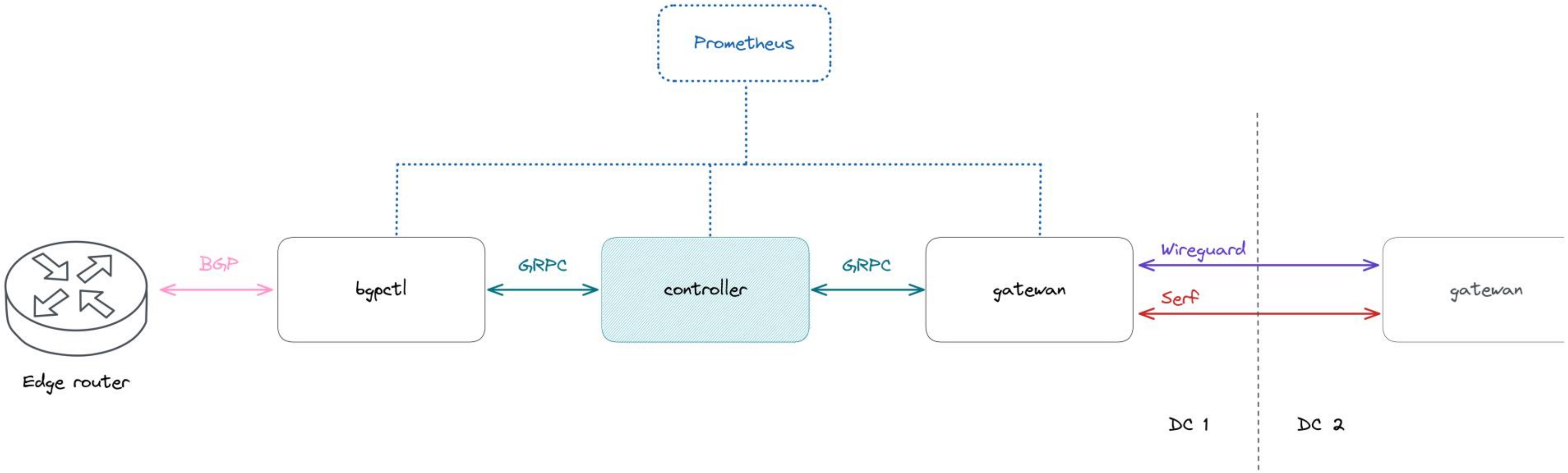
- How do we send the traffic to the Gatewan?

Gatewan



- Use of VXLAN to encapsulate the traffic

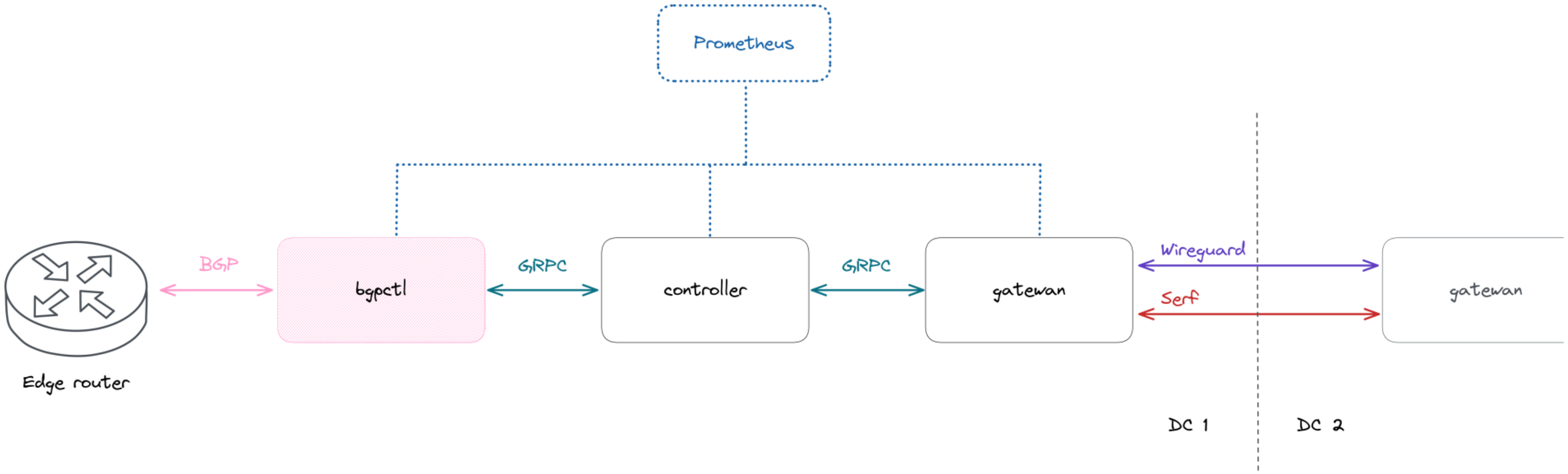
Controller



- 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/

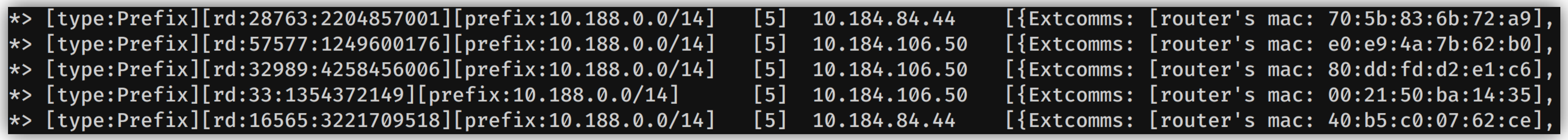
Bgpctl



- 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

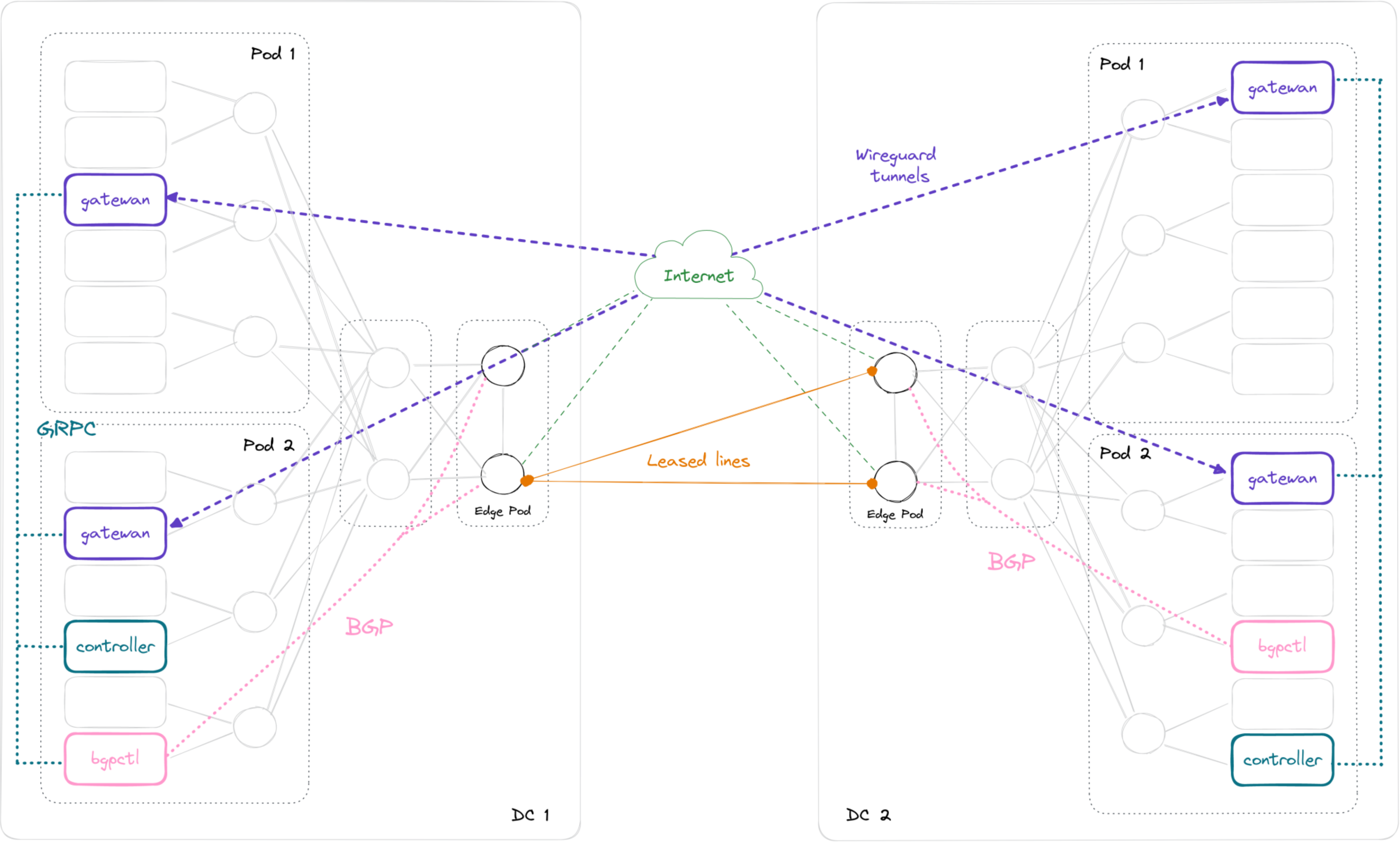


```
*> [type:Prefix][rd:28763:2204857001][prefix:10.188.0.0/14]   [5]  10.184.84.44    [{Extcomms: [router's mac: 70:5b:83:6b:72:a9],
*> [type:Prefix][rd:57577:1249600176][prefix:10.188.0.0/14]   [5]  10.184.106.50   [{Extcomms: [router's mac: e0:e9:4a:7b:62:b0],
*> [type:Prefix][rd:32989:4258456006][prefix:10.188.0.0/14]   [5]  10.184.106.50   [{Extcomms: [router's mac: 80:dd:fd:d2:e1:c6],
*> [type:Prefix][rd:33:1354372149][prefix:10.188.0.0/14]      [5]  10.184.106.50   [{Extcomms: [router's mac: 00:21:50:ba:14:35],
*> [type:Prefix][rd:16565:3221709518][prefix:10.188.0.0/14]   [5]  10.184.84.44    [{Extcomms: [router's mac: 40:b5:c0:07:62:ce],
```

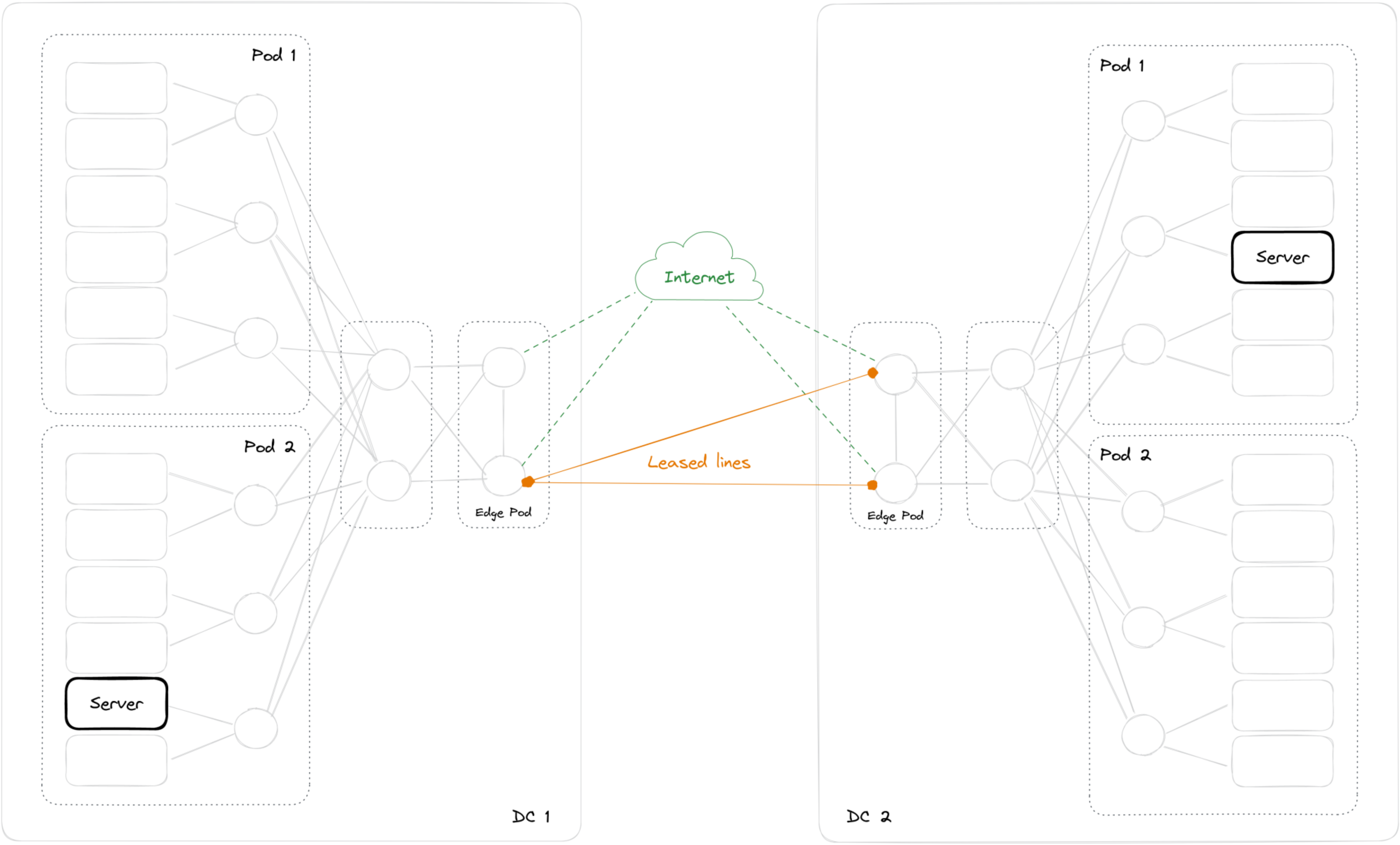
- Provide dynamically the local prefixes to the controller
- Translate the controller's orders to BGP EVPN Route Type 5 (IP Prefix route)
- Based on GoBGP

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

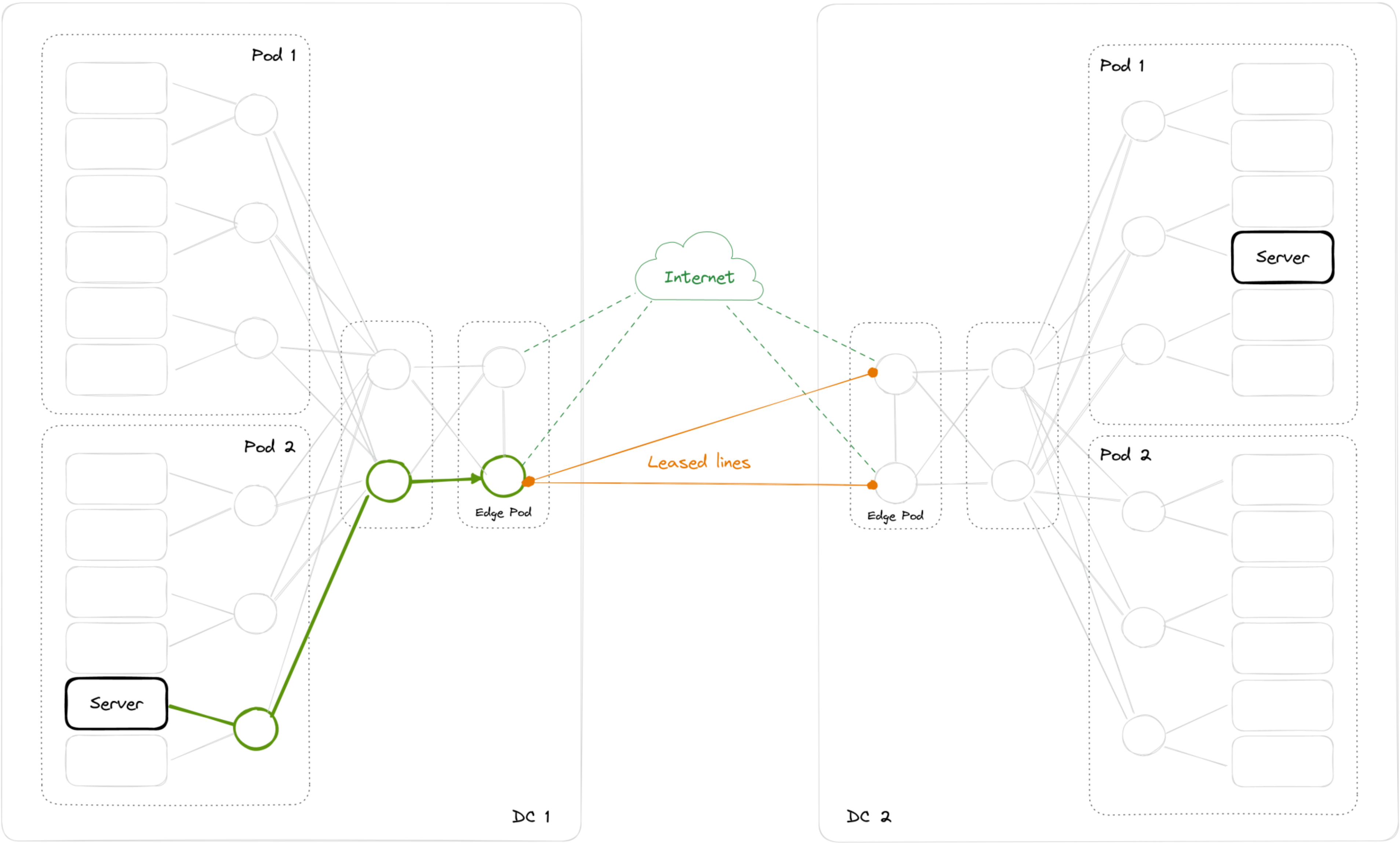
SD-WAN


Pod 1
gatewan
GRPC
Pod 2
gatewan
controller
bgpctl
BGP
Edge Pod
DC 1
Internet
Wireguard tunnels
Leased lines
Edge Pod
BGP
DC 2
Pod 1
gatewan
Pod 2
gatewan
bgpctl
controller

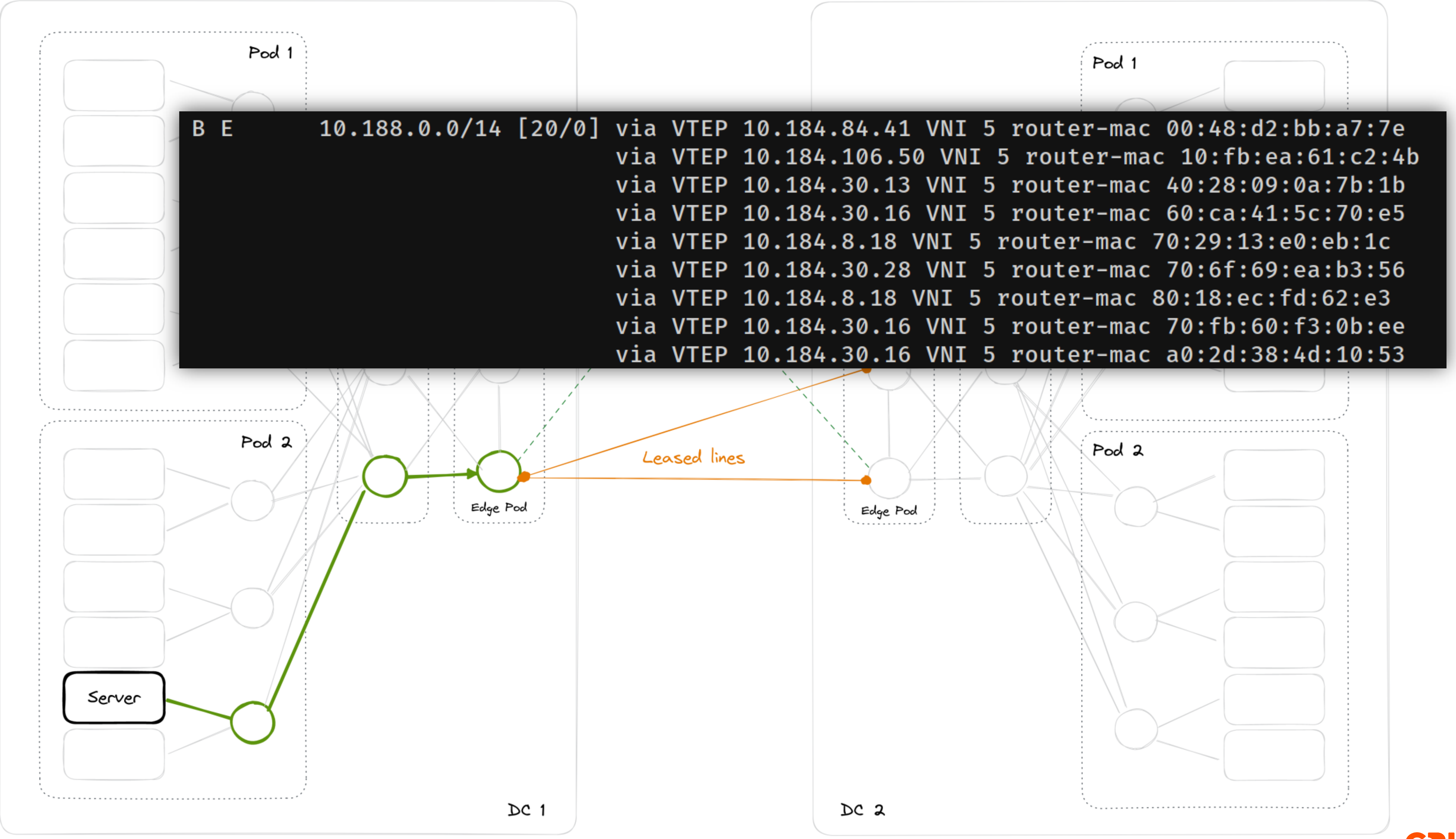
The journey of the packets



The journey of the packets

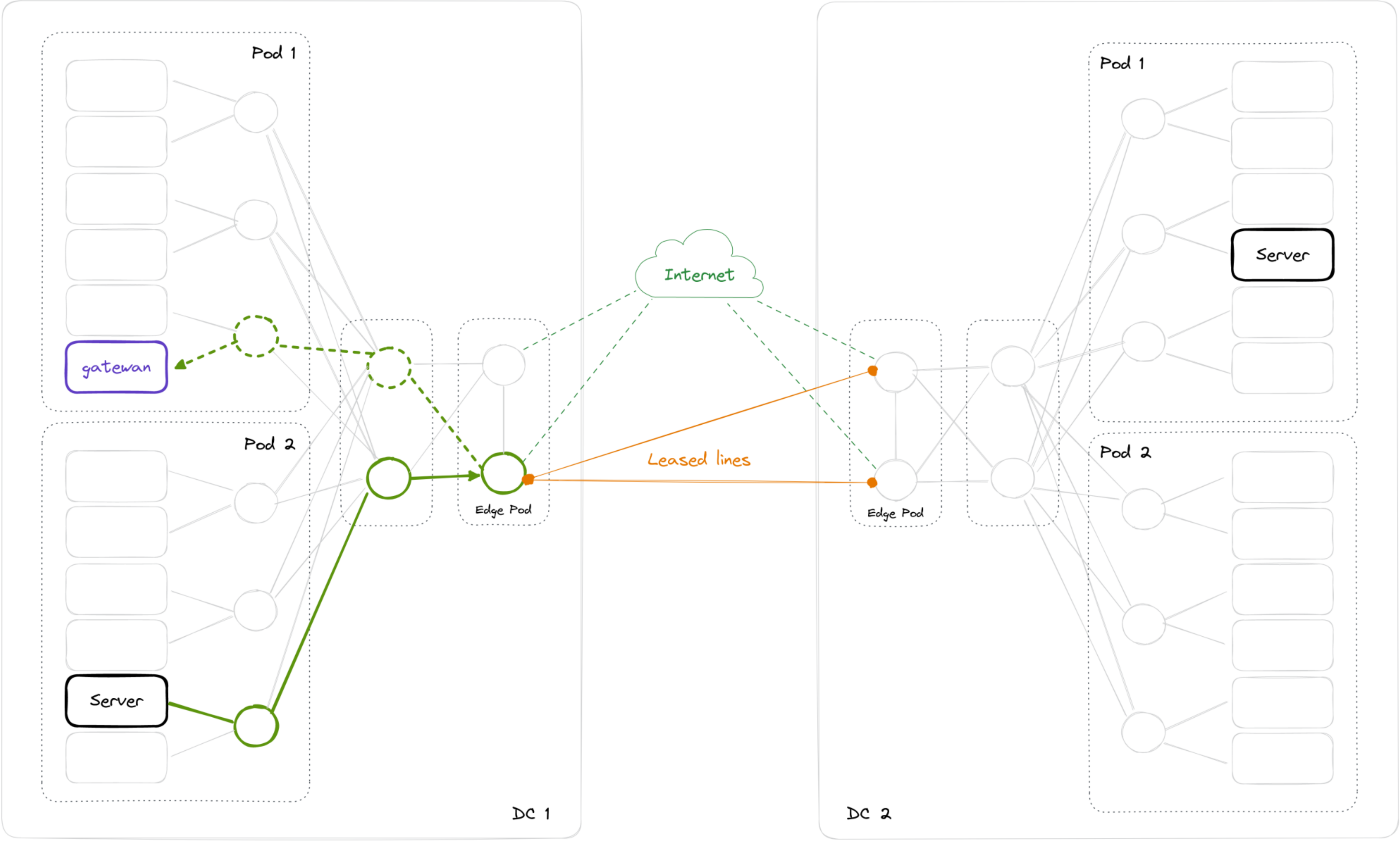


The journey of the packets



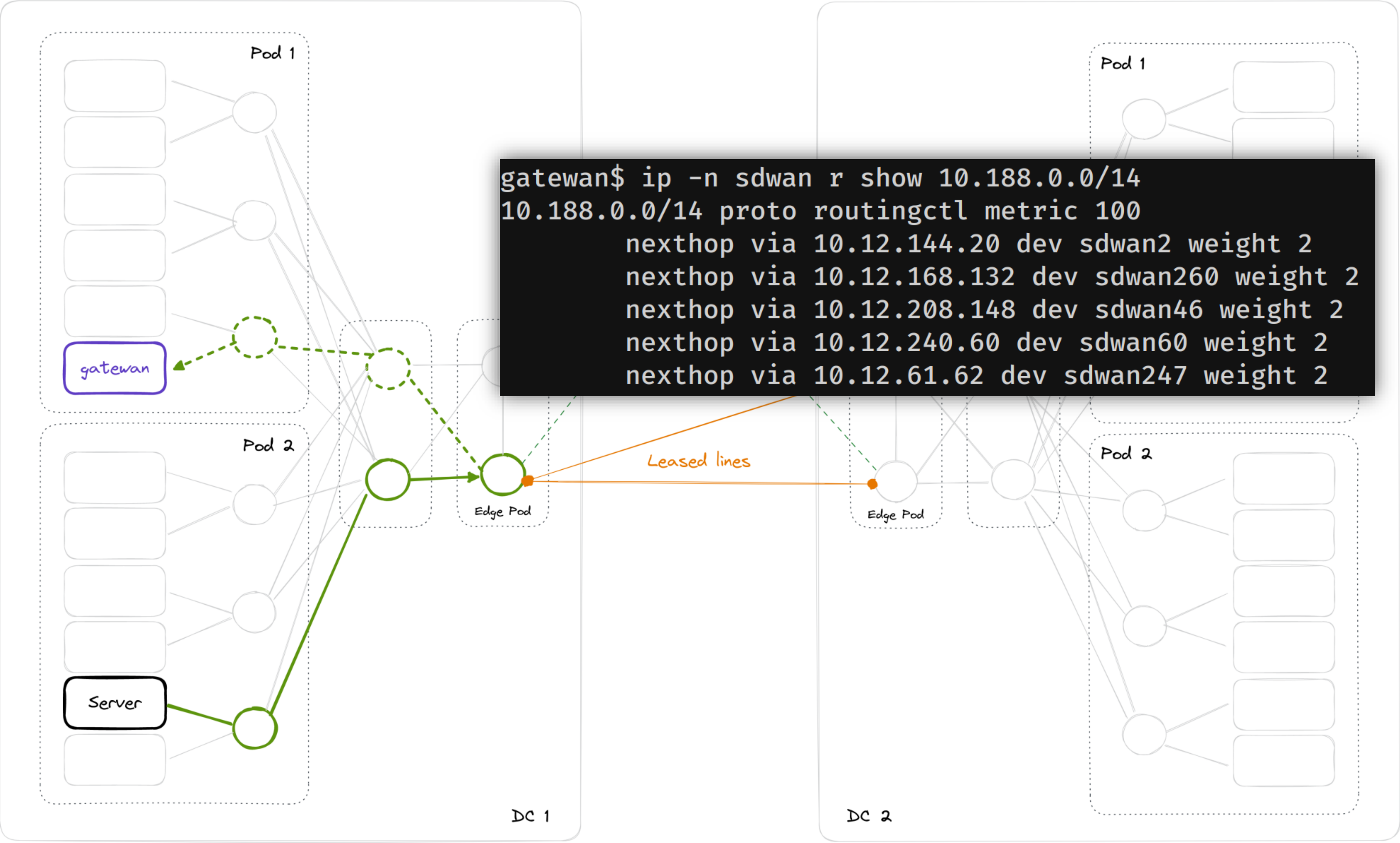
```
B E       10.188.0.0/14 [20/0] via VTEP 10.184.84.41 VNI 5 router-mac 00:48:d2:bb:a7:7e
                               via VTEP 10.184.106.50 VNI 5 router-mac 10:fb:ea:61:c2:4b
                               via VTEP 10.184.30.13 VNI 5 router-mac 40:28:09:0a:7b:1b
                               via VTEP 10.184.30.16 VNI 5 router-mac 60:ca:41:5c:70:e5
                               via VTEP 10.184.8.18 VNI 5 router-mac 70:29:13:e0:eb:1c
                               via VTEP 10.184.30.28 VNI 5 router-mac 70:6f:69:ea:b3:56
                               via VTEP 10.184.8.18 VNI 5 router-mac 80:18:ec:fd:62:e3
                               via VTEP 10.184.30.16 VNI 5 router-mac 70:fb:60:f3:0b:ee
                               via VTEP 10.184.30.16 VNI 5 router-mac a0:2d:38:4d:10:53
```

The journey of the packets



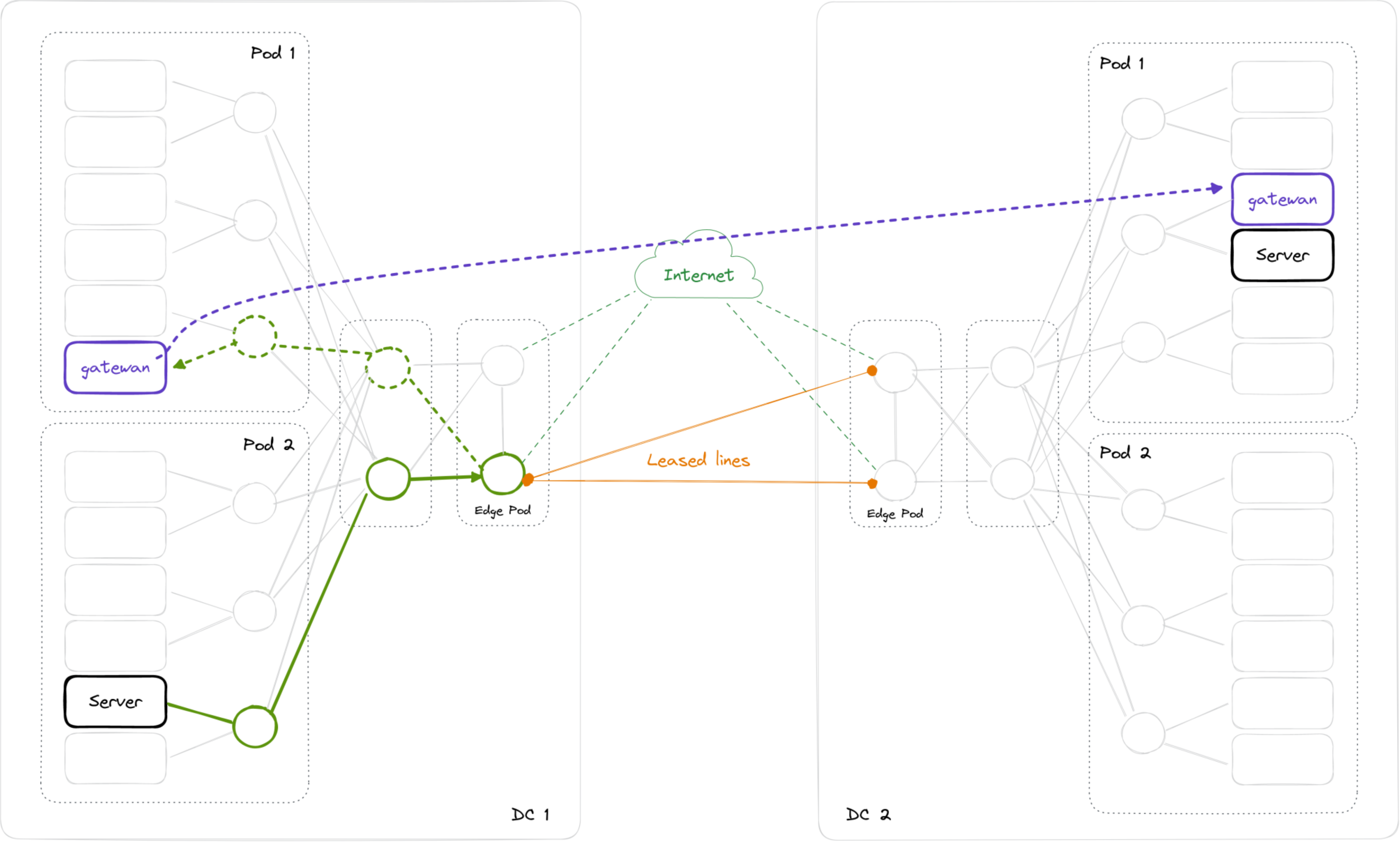
CRITEO

The journey of the packets

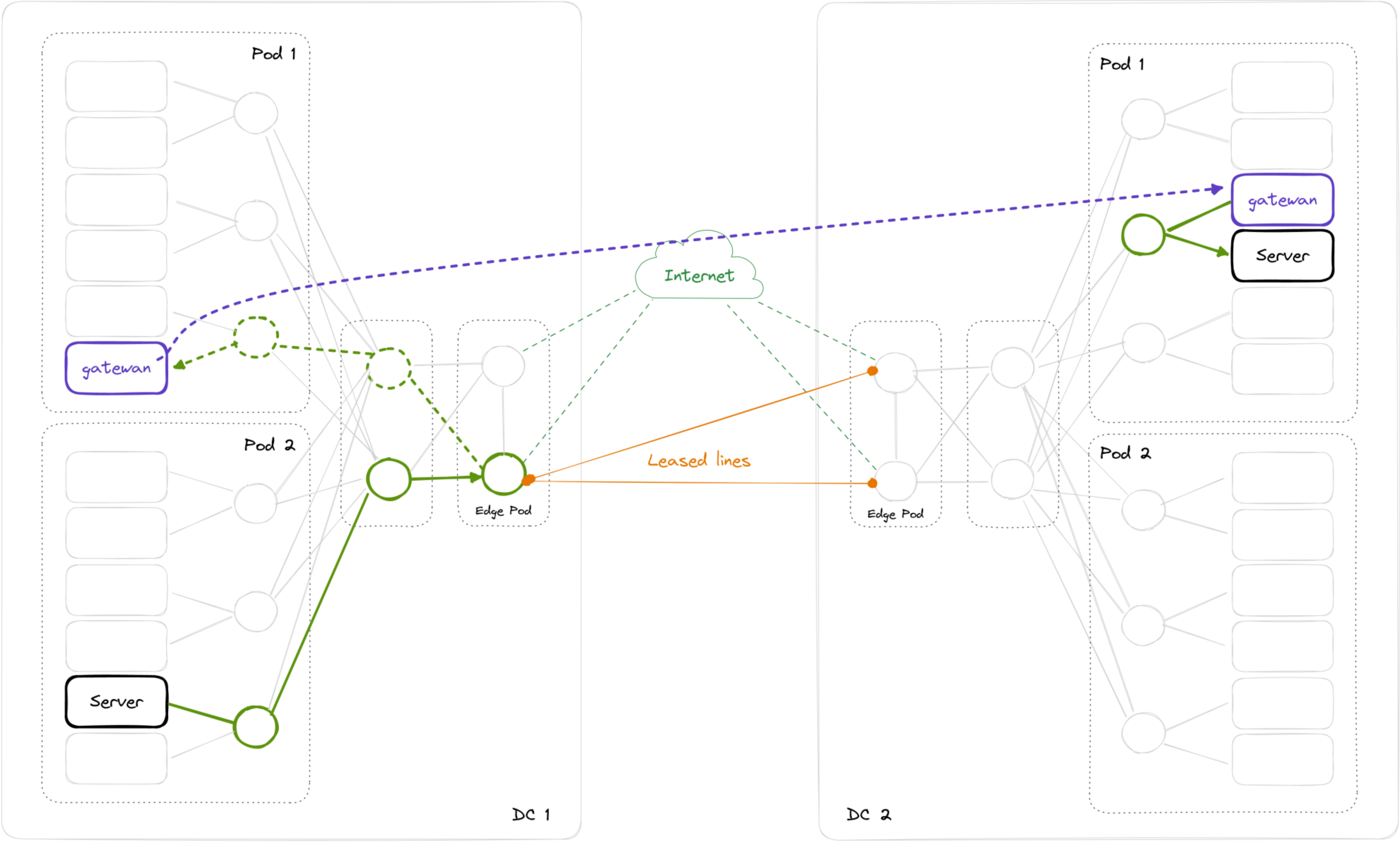


```
gatewan$ ip -n sdwan r show 10.188.0.0/14
10.188.0.0/14 proto routingctl metric 100
        nexthop via 10.12.144.20 dev sdwan2 weight 2
        nexthop via 10.12.168.132 dev sdwan260 weight 2
        nexthop via 10.12.208.148 dev sdwan46 weight 2
        nexthop via 10.12.240.60 dev sdwan60 weight 2
        nexthop via 10.12.61.62 dev sdwan247 weight 2
```

The journey of the packets



The journey of the packets



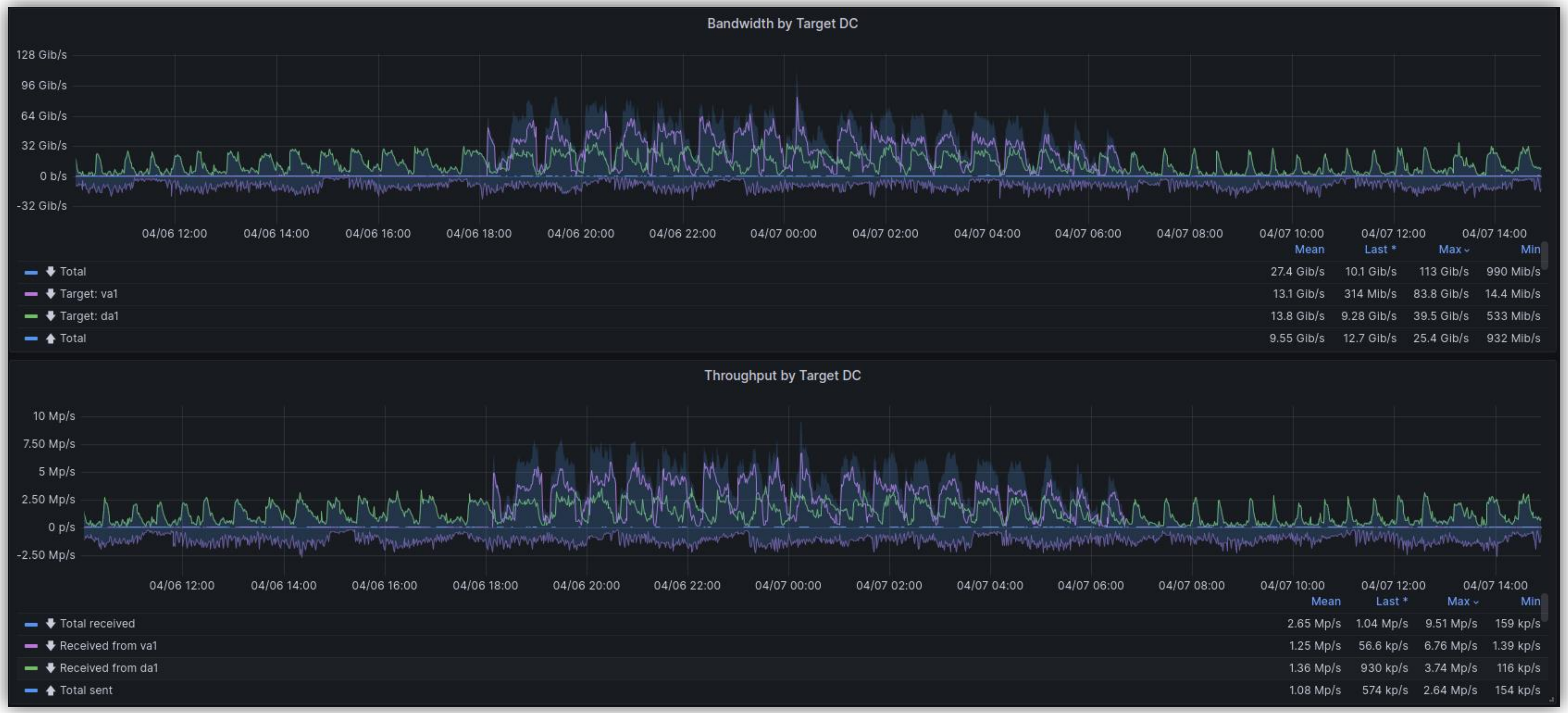
Drawbacks

- 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



Bandwidth by Target DC

	Mean	Last *	Max	Min
⬇ Total	27.4 Gib/s	10.1 Gib/s	113 Gib/s	990 Mib/s
⬇ Target: va1	13.1 Gib/s	314 Mib/s	83.8 Gib/s	14.4 Mib/s
⬇ Target: da1	13.8 Gib/s	9.28 Gib/s	39.5 Gib/s	533 Mib/s
⬆ Total	9.55 Gib/s	12.7 Gib/s	25.4 Gib/s	932 Mib/s

Throughput by Target DC

	Mean	Last *	Max	Min
⬇ Total received	2.65 Mp/s	1.04 Mp/s	9.51 Mp/s	159 kp/s
⬇ Received from va1	1.25 Mp/s	56.6 kp/s	6.76 Mp/s	1.39 kp/s
⬇ Received from da1	1.36 Mp/s	930 kp/s	3.74 Mp/s	116 kp/s
⬆ Total sent	1.08 Mp/s	574 kp/s	2.64 Mp/s	154 kp/s

CRITEO

Thank you