RAIT - Poor man’s SDWAN

by Nick Cao

什么是~~沙雕网~~SDWAN

首先他是个WAN，然后它SD

什么是SDWAN

Software defined wide area network (SD-WAN) is a type of computer network that enables bonding of multiple internet access resources – such as DSL, cable, cellular or any other IP transport – to provide reliable high throughput data channels.

ref: What is SD WAN? Software defined WAN (SDWAN) explained – GFI

Scope of the problem

IP transport: tunnel over clearnet

Bonding of resource: out of scope, why not MPTCP

provide reliable high throughput data channels: dynamic routing

那我怎么搞一个

- Zerotier ZeroTier – Global Area Networking
- Tinc Tinc VPN
- Weave Net Weave Net: Network Containers Across Environments
- Flannel coreos/flannel: flannel is a network fabric for containers, designed for Kubernetes

However……

- 大多有着糟糕的性能
- 甚至有自己的关不掉的IPAM
- 还可以硬依赖Docker
- 更别提诡异的选路

他们干的太多了！

Do One Thing and Do It Well

Step 1: Link Local Connectivity (IP transport)

- VXLAN
- GRE
- IPIP
- GRETAP
- GENEVE

Pros and Cons

Pro

1. standardized protocol ensures interoperability
2. BGP EPVN and other existing control plane eases deployment

Cons

1. the unneeded ethernet header adds to overhead
2. protocols other than TCP and UDP may have issue with middle boxes

But we have: wireguard

1. operates on layer 3
2. UDP encapsulated
3. built in roaming
4. formally verified cryptography, protocol and implementation
5. available natively in FreeBSD, OpenBSD and Linux

(it turns out to be a bad decision latter though)

Step 2: Site Local Connectivity (Routing Protocols)

- RIP
- BGP
- OSPF
- ISIS
- EIGRP
- OpenFabric

Pros and Cons

Pros

1. S T A N D A R D（

Cons

1. mostly with static cost/metric
2. heavy implementation not suitable for restricted environments

Still we have: babel [Babel — a loop-avoiding distance-vector routing protocol]

- optimized for wireless network or tunnels
- updates link cost based on the RTT
- and even other metrics
- source specific routing !
- have a stub implementation for embeded system
- https://grafana.nichi.co/d/6td87mzGz/node-metrics?viewPanel=12&orgId=1

(I know, but the reference implementation sucks)

~~Step 3: Global Connectivity~~

~~首先去RIPE注册一下ASN~~

~~然后Vultr全区开满~~

~~APNIC同款Anycast网络有子~~

However……

Wireguard is broken, as intended.

AlllowedIPs is nothing but a routing table

And babeld requires multicast

解决办法

那我们创建114514个interface，每个peer一人一个不就好了

你说interface数量上限，不存在的（我试过了）

```
# for i in {1..1000000}; do ip l add $RANDOM$RANDOM type dummy; done
```

```
# ip l | wc -l
```

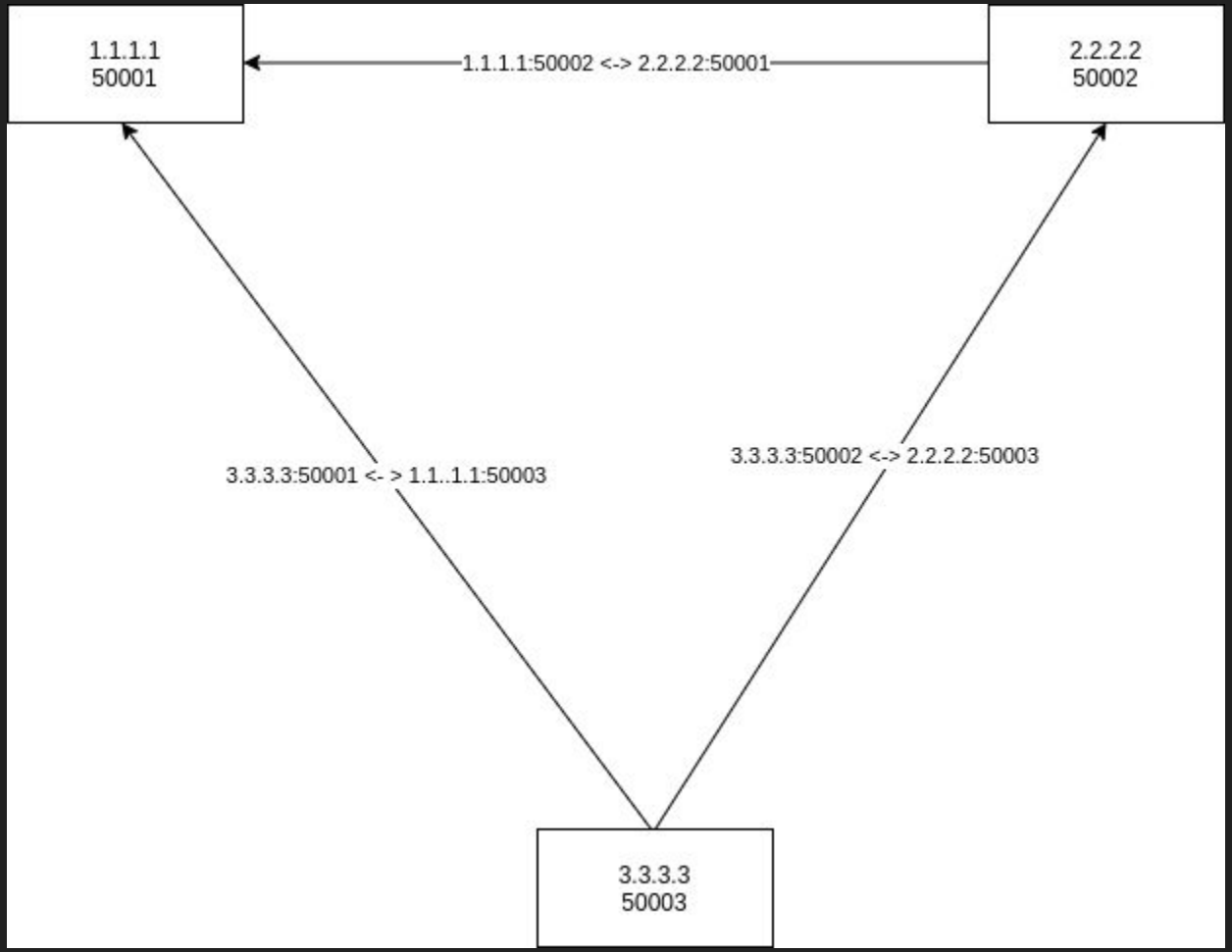
70604

至于端口，我们等下再说

关于端口

端口数量：1025-65535

端口分配：如何避免冲突？SendPort！


1.1.1.1
50001
1.1.1.1:50002 <-> 2.2.2.2:50001
2.2.2.2
50002
3.3.3.3:50001 <- > 1.1..1.1:50003
3.3.3.3:50002 <-> 2.2.2.2:50003
3.3.3.3
50003

RAIT NickCao / RAIT · GitLab

（图片太大了塞不下）

https://pb.nichi.co/3f7b343d-5284-466c-b246-35133fd3594d

What’s more

如何利用好多个上游？

https://github.com/FireflyTang/linux-wireguard-bind

What’s more

如何省去这一打interface？

AF_WIREGUARD and mapped ethernet address

Babeld

random-id true
export-table 254
local-path-readwrite /run/babeld.ctl

default type tunnel link-quality true split-horizon false rxcost 32 hello-interval 20
default max-rtt-penalty 1024 rtt-max 1024

interface foo

redistribute ip 2a0c:b641:69c::/48 ge 64 le 64 allow
redistribute ip ::/0 le 0 src-prefix 2a0c:b641:69c::/48 metric 4096
redistribute local deny

See it in action

2a0c:b641:69c:99cc::1

Thanks for watching!