Accelerate Host Network Traffic

Hongliang Liu

What We Are Trying to Solve

Accelerate the traffic below between antrea-gw0 and other interfaces traversing host networking stack.

Traffic Type	Traffic Mode	Host Networking Path	Rely on Host Networking
Pod-to-Pod	noEncap	antrea-gw0 <-> transport interface	No
Pod-to-Service	noEncap	antrea-gw0 <-> transport interface	No
External-to-LoadBalancerIP / ExternalIP	encap/noEncap/ Hybrid	Default route interface <-> antrea-gw0	No
External-to-NodePort	encap/noEncap/ Hybrid	NodePort interface <-> antrea-gw0	DNAT
External-to-Service	noEncap	antrea-gw0 <-> transport interface	SNAT

Solution Based on Traffic Control

Traffic Type	Traffic Mode	Host Networking Path	Supported
Pod-to-Pod	noEncap	antrea-gw0 <-> transport interface	Yes
Pod-to-Service	noEncap	antrea-gw0 <-> transport interface	Yes
External-to-LoadBalancerIP / ExternalIP	encap/noEncap/ Hybrid	Default route interface <-> antrea-gw0	No
External-to-NodePort	encap/noEncap/ Hybrid	NodePort interface <-> antrea-gw0	No
External-to-Service	noEncap	antrea-gw0 <-> transport interface	No

Solution Based on Traffic Control

OVS Pipeline Changes

Local Pod CIDR

Remote Pod CIDR

table=**L3Forwarding**, priority=211,ip,**nw_src=10.10.0.0/24**,**nw_dst=10.10.1.0/24**
actions=**set_field:0a:00:00:00:00:01->eth_src**,**set_field:00:00:10:10:01:01->eth_dst**,set_field:0x20/0xf0->reg0,goto_table:L3DecTTL

Set to local Node transport interface' MAC

Set to remote Node transport interface' MAC

table=L2ForwardingCalc, priority=200,**dl_dst=00:00:10:10:01:01 actions=set_field:0x8001->reg1**,set_field:0x200000/0x600000->reg0,goto_table:ConntrackCommit

Set output port to antrea-gw0 if the destination MAC is the remote Node transport interface' MAC

Solution Based on Traffic Control

TC on eth0 (assumed it is the transport interface)

```
filter protocol ip pref 40010 flower chain 0
filter protocol ip pref 40010 flower chain 0 handle 0x1
  eth_type ipv4
  dst_ip 10.10.1.0/24
  src_ip 10.10.0.0/24
  not_in_hw
        action order 1: mirred (Egress Redirect to device antrea-gw0) stolen
        index 3 ref 1 bind 1
```

Local Pod CIDR

Remote Pod CIDR

Redirect traffic to antrea-gw0 egress, Bypassing host network stack.

TC on antrea-gw0

```
filter protocol ip pref 40010 flower chain 0
filter protocol ip pref 40010 flower chain 0 handle 0x1
  eth_type ipv4
  dst_ip 10.10.0.0/24
  src_ip 10.10.1.0/24
  not_in_hw
        action order 1: mirred (Egress Redirect to device eth0) stolen
        index 3 ref 1 bind 1
```

Remote Pod CIDR

Local Pod CIDR

Redirect traffic to eth0 egress, Bypassing host network stack.

Solution Based on Nftables Flowtable

Traffic Type	Traffic Mode	Host Networking Path	Supported
Pod-to-Pod	noEncap	antrea-gw0 <-> transport interface	Yes
Pod-to-Service	noEncap	antrea-gw0 <-> transport interface	Yes
External-to-LoadBalancerIP / ExternalIP	encap/noEncap/ Hybrid	Default route interface <-> antrea-gw0	Yes
External-to-NodePort	encap/noEncap/ Hybrid	NodePort interface <-> antrea-gw0	Yes
External-to-Service	noEncap	antrea-gw0 <-> transport interface	Yes

More types of traffic could be supported, e.g., Node-to-remote Pod CIDR.

Solution Based on Nftables Flowtable

```
# Create a new table for Antrea
nft add table ip antrea

# Define a flowtable named "fastpath".
# Add all interfaces that may carry accelerated connections.
# For Service traffic (e.g. NodePort), packets may arrive on any interface, so include them all.
nft add flowtable ip antrea fastpath \
  { hook ingress priority 0\; devices = { antrea-gw0, eth0, eth1, eth2 }\; }

# Create a forward chain to attach acceleration rules.
nft add chain ip antrea forward { type filter hook forward priority 0\; }

# Accelerate Pod-to-Pod / Pod-to-Service traffic.
nft add rule ip antrea forward iif "antrea-gw0" ip saddr 10.10.0.0/24 oif "eth0" ip daddr 10.10.1.0/24 flow add @fastpath

# Accelerate External-to-Pod traffic before SNAT.
nft add rule ip antrea forward iif "antrea-gw0" ip saddr 169.254.0.253 oif "eth0" ip daddr 10.10.1.0/24 flow add @fastpath

# Accelerate External-to-NodePort traffic (via antrea-gw0) after DNAT.
nft add rule ip antrea forward oif "antrea-gw0" ip daddr 169.254.0.252 flow add @fastpath

# Accelerate External-to-LoadBalancerIP / ExternalIP traffic (via antrea-gw0).
nft add rule ip antrea forward oif "antrea-gw0" ip daddr 172.18.0.151 flow add @fastpath
```

Local Pod CIDR

Remote Pod CIDR

Hairpin SNAT IP

Remote Pod CIDR

NodePort DNAT IP

LoadBalancer Ingress IP/ ExternalIP

Test Results

Pod-to-Pod

	TCP_STREAM	TCP_RR	TCP_CRR
Default	16049.15	18208.75	7259.13
Traffc Control	17968.28 **(+20%)**	20589.19 **(+14%)**	10269.33 **(+41%)**
Flowtable	20986.28 **(+31%)**	21133.65 **(+16%)**	7691.5 **(+6%)**

Test Results

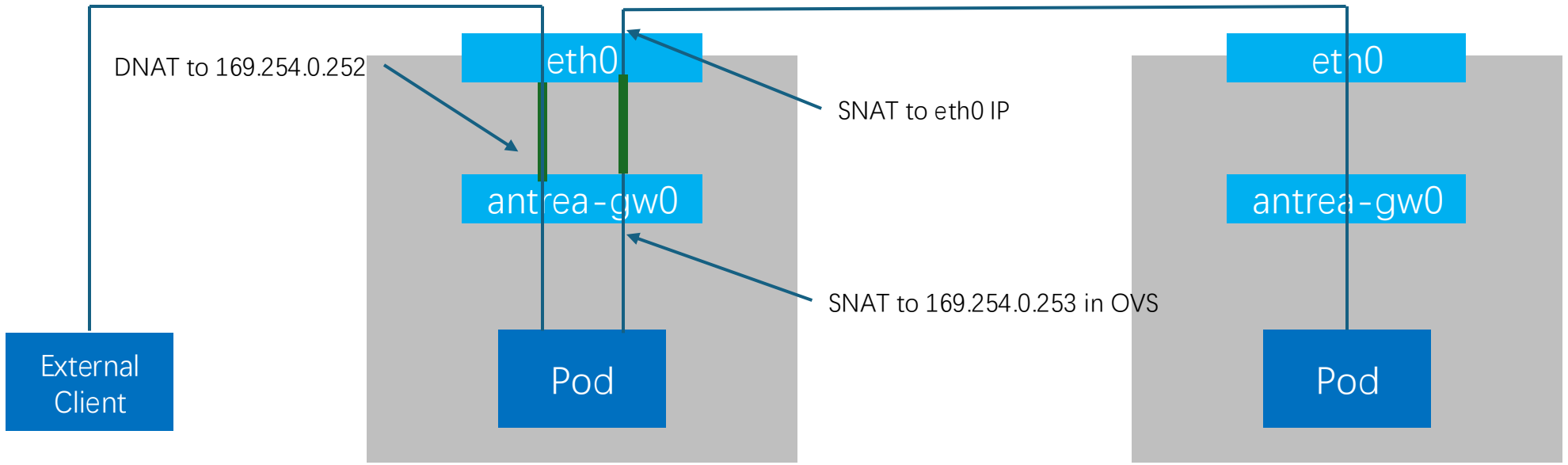
Pod-to-Service

	TCP_STREAM	TCP_RR	TCP_CRR
Default	12993.01	17842.92	7245.25
Traffc Control	18272.46 **(+40%)**	20442.56 **(+15%)**	9701.89 **(+34%)**
Flowtable	19876.21 **(+53%)**	20985.76 **(+18%)**	8085.41 **(+16%)**

Test Results

External-to-NodePort

	TCP_STREAM	TCP_RR	TCP_CRR
Default	12648.59	17956.66	7315.98
Flowtable	16419.67 **(+30%)**	18589.67 **(+4%)**	6837.70 **(-7%)**



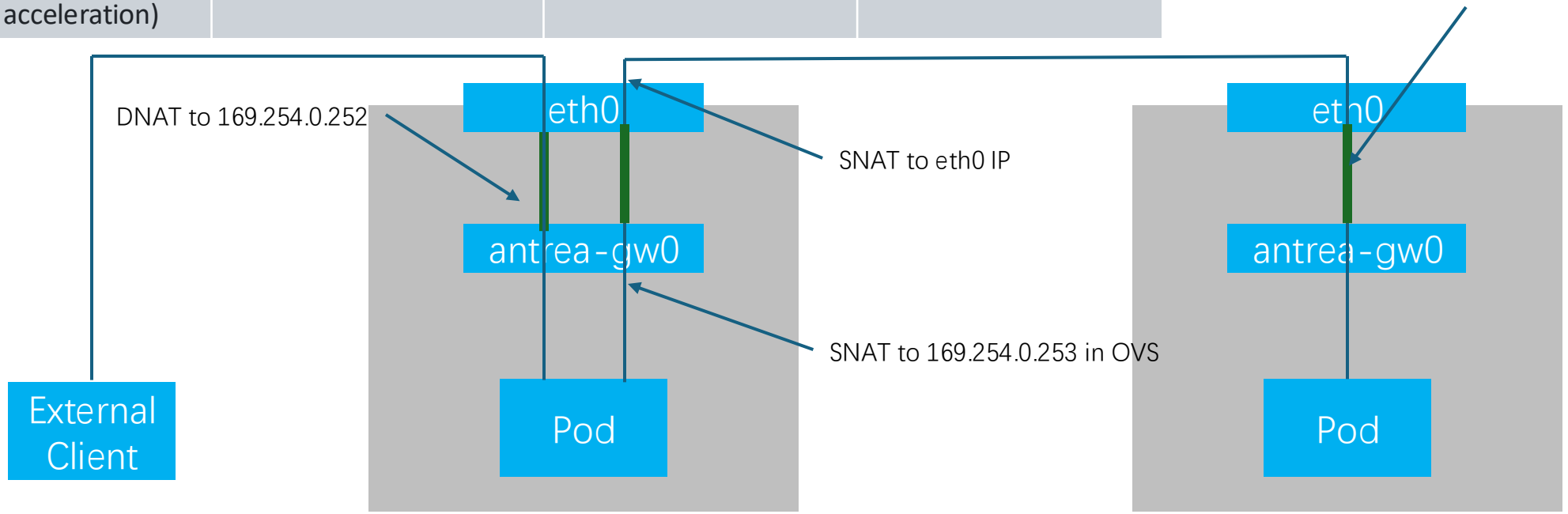
Test Results

External-to-NodePort

	TCP_STREAM	TCP_RR	TCP_CRR
Default	12648.59	17956.66	7315.98
Flowtable	16419.67 **(+30%)**	18589.67 **(+4%)**	6837.70 **(-7%)**
Flowtable (full path acceleration)	18558.89 **(+47%)**	19234.2 **(+7%)**	7052.20 **(-4%)**

Accelerate traffic with the nft rule:

```
nft add rule ip antrea forward
  # Remote transport IP
  ip saddr 172.18.0.2 \
  # Local Pod CIDR
  ip daddr 10.10.1.0/24 \
  flow add @fastpath counter
```



Test Results

	Traffic Type	TCP_STREAM	TCP_RR	TCP_CRR
Default	Pod-to-Pod	16049.15	18208.75	7259.13
Traffc Control	Pod-to-Pod	17968.28 **(+20%)**	20589.19 **(+14%)**	10269.33 **(+41%)**
Flowtable	Pod-to-Pod	20986.28 **(+31%)**	21133.65 **(+16%)**	7691.5 **(+6%)**
Default	Pod-to-Service	12993.01	17842.92	7245.25
Traffc Control	Pod-to-Service	18272.46 **(+40%)**	20442.56 **(+15%)**	9701.89 **(+34%)**
Flowtable	Pod-to-Service	19876.21 **(+53%)**	20985.76 **(+18%)**	8085.41 **(+16%)**
Default	External-to-NodePort	12648.59	17956.66	7315.98
Flowtable	External-to-NodePort	16419.67 **(+30%)**	18589.67 **(+4%)**	6837.70 **(-7%)**
Flowtable(full path)	External-to-NodePort	18558.89 **(+47%)**	19234.20 **(+7%)**	7052.20 **(-4%)**

Traffic Control vs. Nftables Flowtable

Traffic Control

- Better for **TCP_CRR** (short-lived connections).
- **Bypasses the host networking stack** completely for all packets.
- **Stateless**: requires hacking source/destination MAC addresses.
- Limited scope: only accelerates **Pod-to-Pod / Pod-to-Service** traffic; cannot handle flows that depend on host networking stack (e.g., NAT).

Nftables Flowtable

- Better for **TCP_STREAM / TCP_RR** (long-lived or continuous flows).
- First 1–2 packets traverse full stack; subsequent packets **bypass most hooks** (prerouting, forwarding, postrouting).
- **Stateful**: no need to rewrite MACs; accelerated flows visible in **conntrack**.
- Flexible: can accelerate **all traffic types**, including those relying on host networking (e.g., SNAT/DNAT).

Nftables Flowtable is well integrated into the networking stack, suitable for general use cases and complex forwarding requirements.