Deep dive into Kubernetes Networking

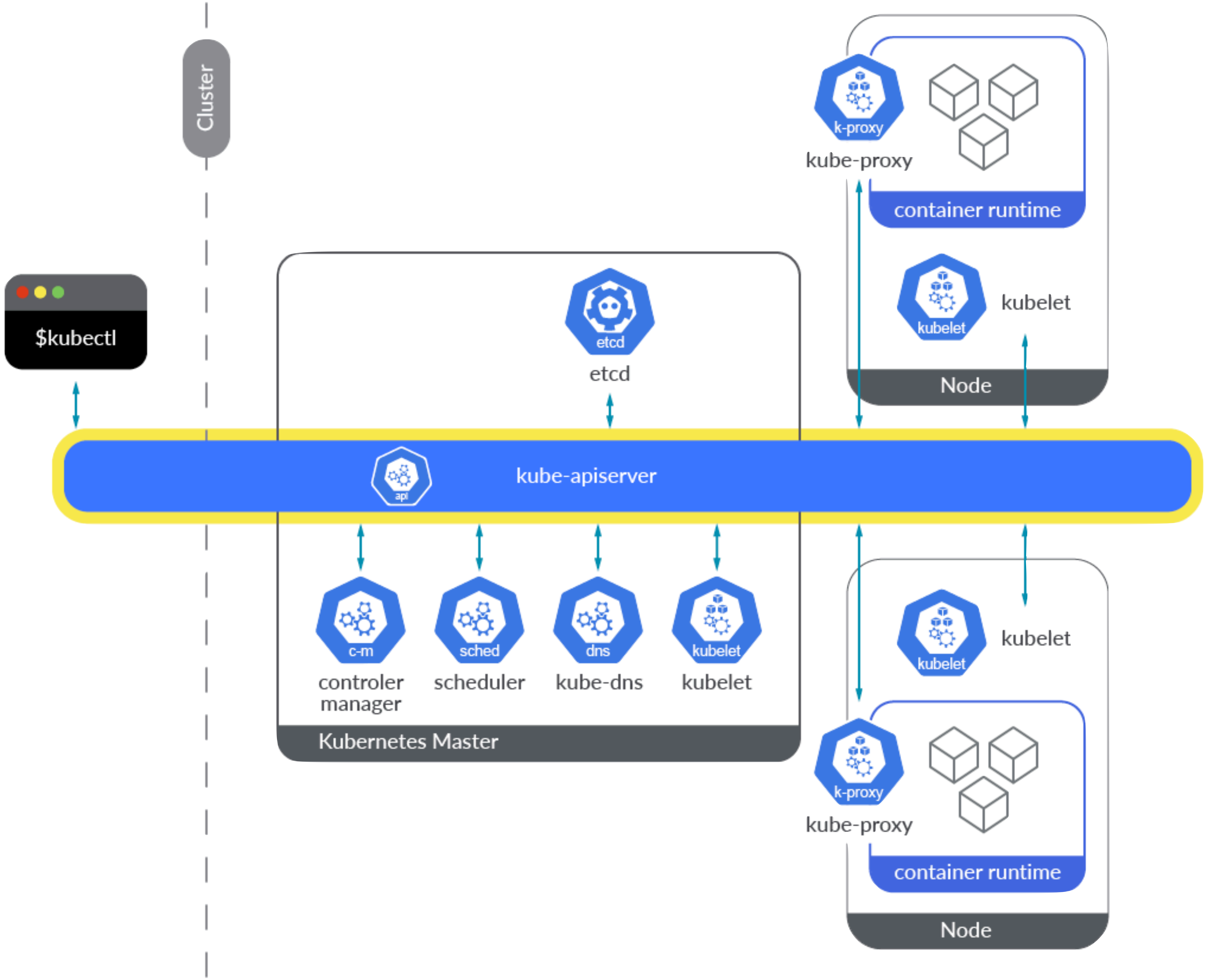
Rajat Khanna

SDE III @ CommerceIQ

legalimpurity rajatkhanna08 legalimpurity

Prerequisites

Kubernetes API Server

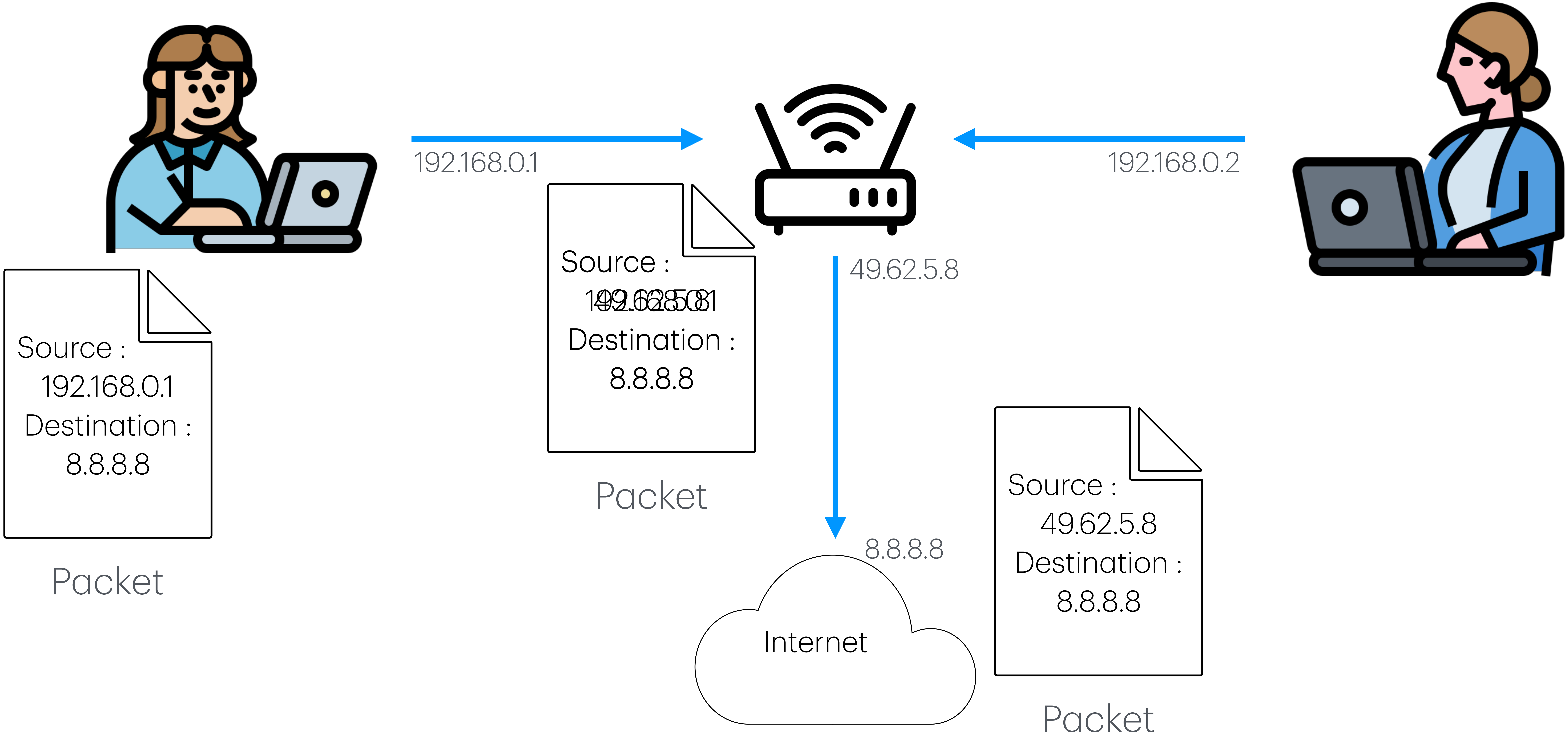


Controllers

```
while true:
  X = currentState()
  Y = desiredState()

  if X == Y:
    return  # Do nothing
  else:
    do(tasks to get to Y)
```

What is NAT?



Source :
192.168.0.1
Destination :
8.8.8.8

Packet

Source :
192.168.0.1 49.62.5.8
Destination :
8.8.8.8

Packet

Source :
49.62.5.8
Destination :
8.8.8.8

Packet

Kubernetes Networking Model

Requirements for any networking implementation

- All Pods can communicate with all other Pods without NAT
- All nodes can communicate with all Pods without NAT
- The IP that a Pod sees itself as is the same IP that others see it as.

Types of Networking in Kubernetes

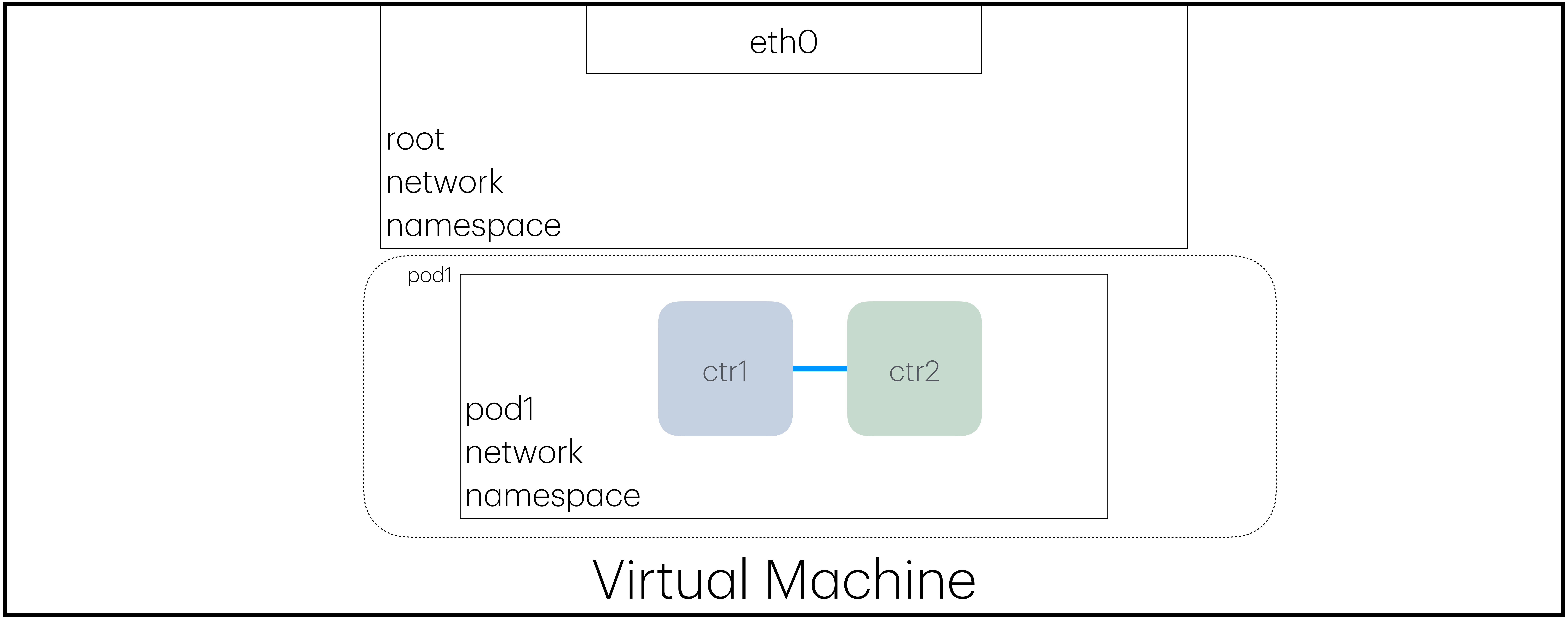
- Container to Container Networking
- Pod to Pod Networking
  - Same Node
  - Different Nodes
- Pod to Service Networking
- Pod to Internet Networking (Egress)
- Internet to Pod Networking (Ingress)
  - Layer 4 Load Balancer
  - ~~Layer 7 Ingress Controller~~

Container to Container Networking

Container to Container Networking

Life of a Packet

src	ctr1
dest	ctr2

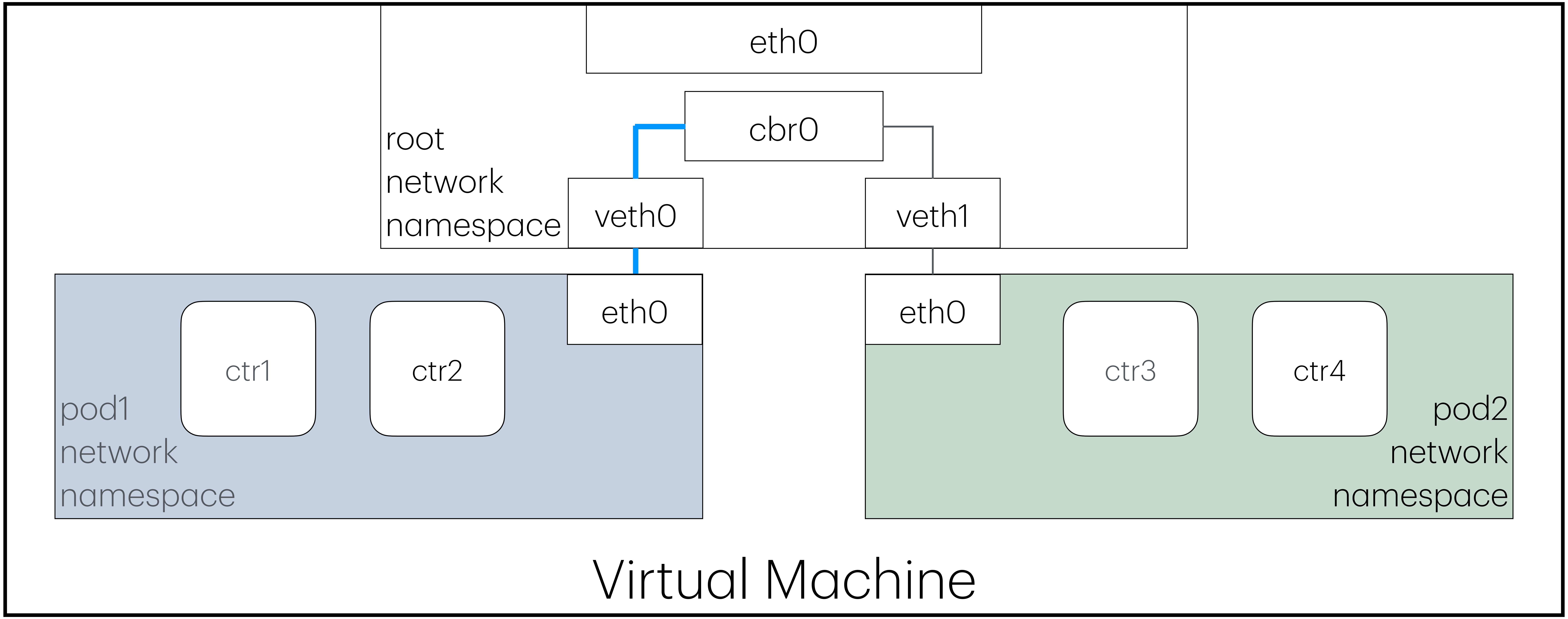


Pod to Pod Networking

Pod to Pod Networking

Life of a Packet: Same Node

src	pod1
dest	pod2

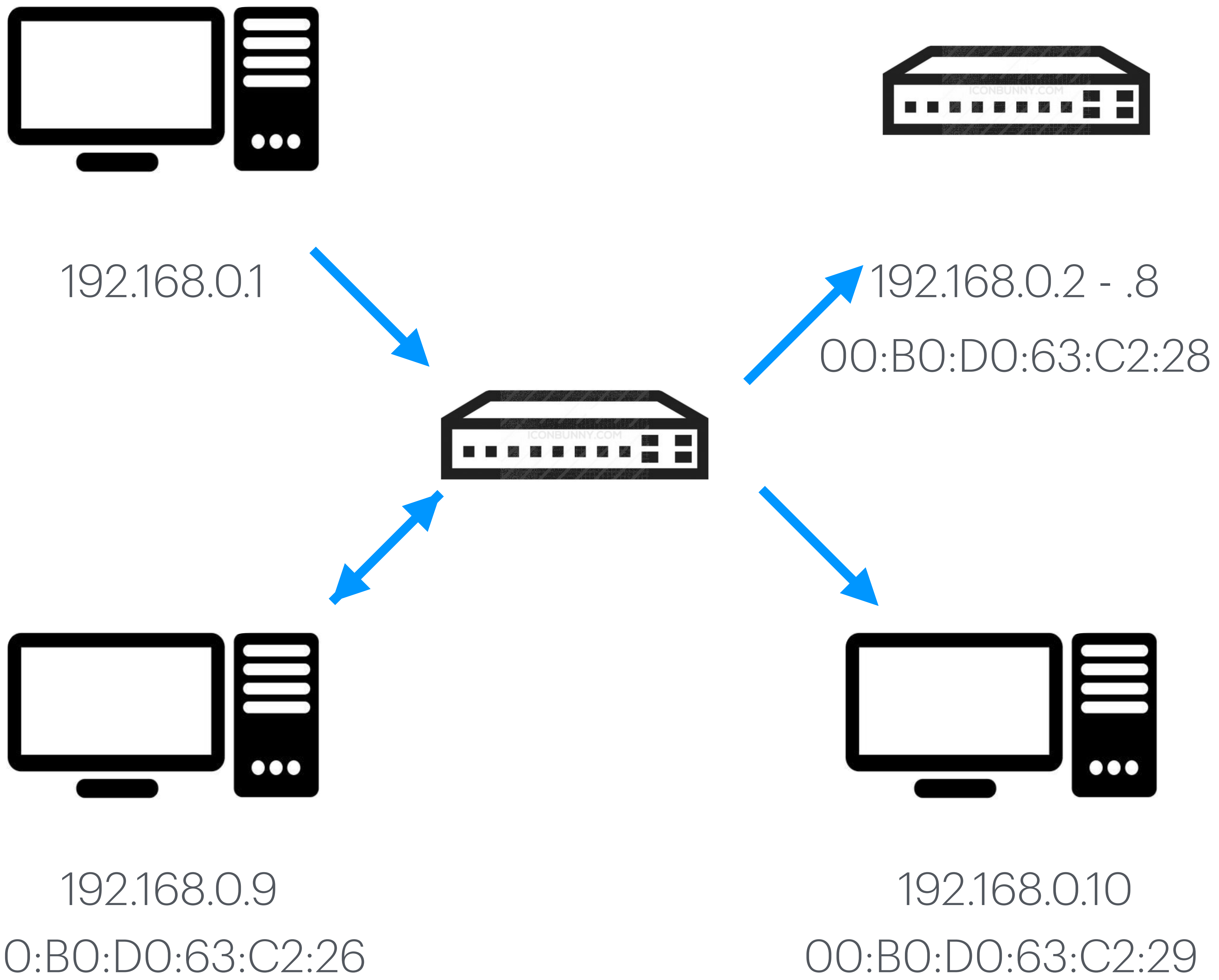


Pod to Pod Networking

Address Resolution Protocol

ARP Lookup Table

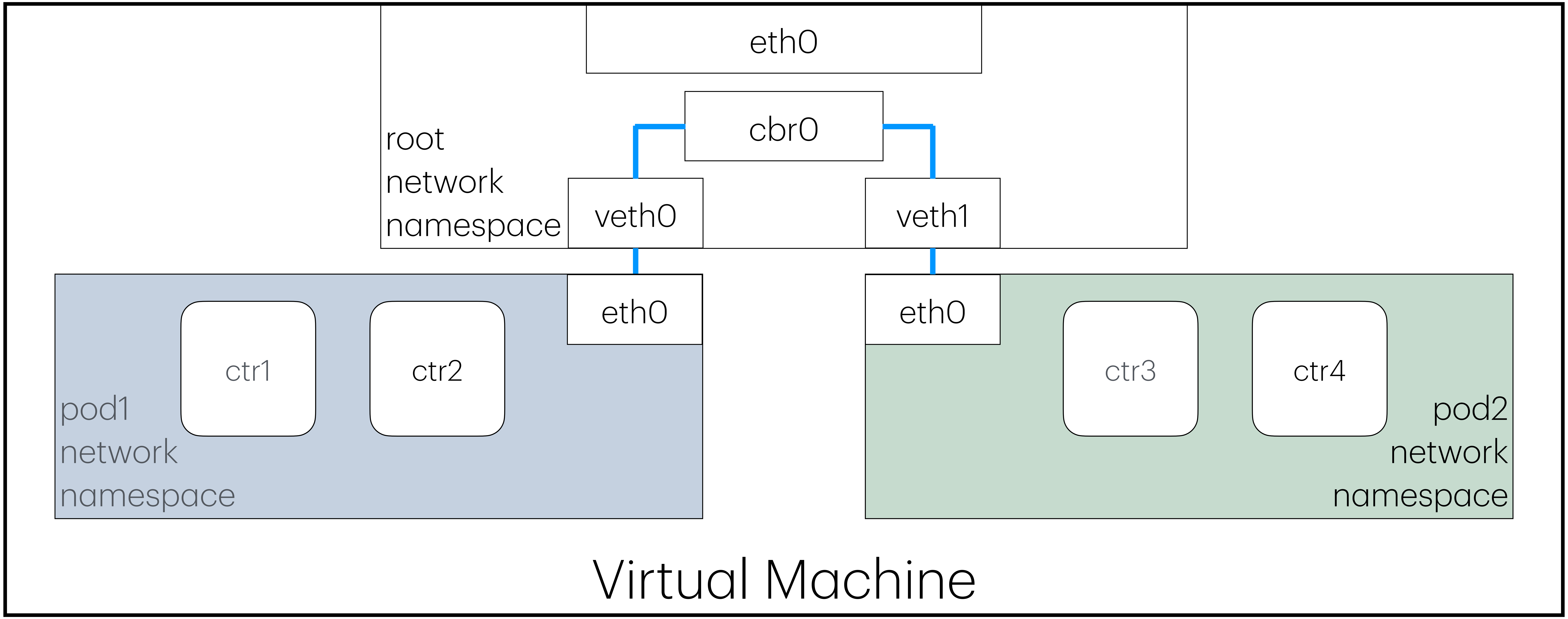
IP	MAC ADDRESS
192.168.0.9	00:B0:D0:63:C2:26
192.168.0.2	00:B0:D0:63:C2:28
192.168.0.10	00:B0:D0:63:C2:29



Pod to Pod Networking

Life of a Packet: Same Node

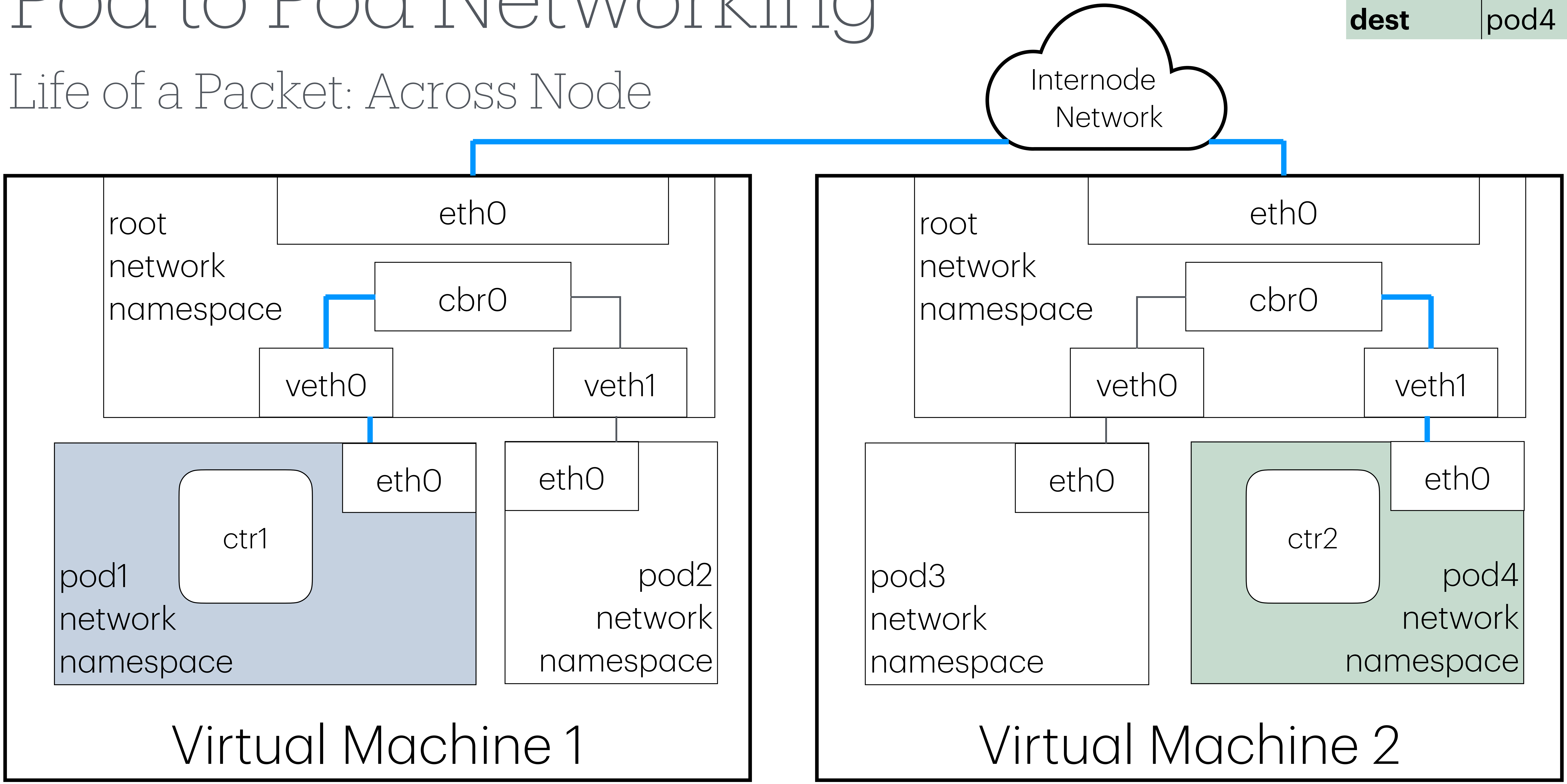
src	pod1
dest	pod2



Pod to Pod Networking

Life of a Packet: Across Node

src	pod1
dest	pod4



K8s service

- A Kubernetes service allows you to track a set of Pod Ip addresses that are dynamically changing over time.

- Services act as an abstraction over Pods and assign a single virtual IP address to a group of Pod IP addresses.

- Any traffic addressed to the virtual IP of the Service will be routed to the set of Pods that are associated with the virtual IP.

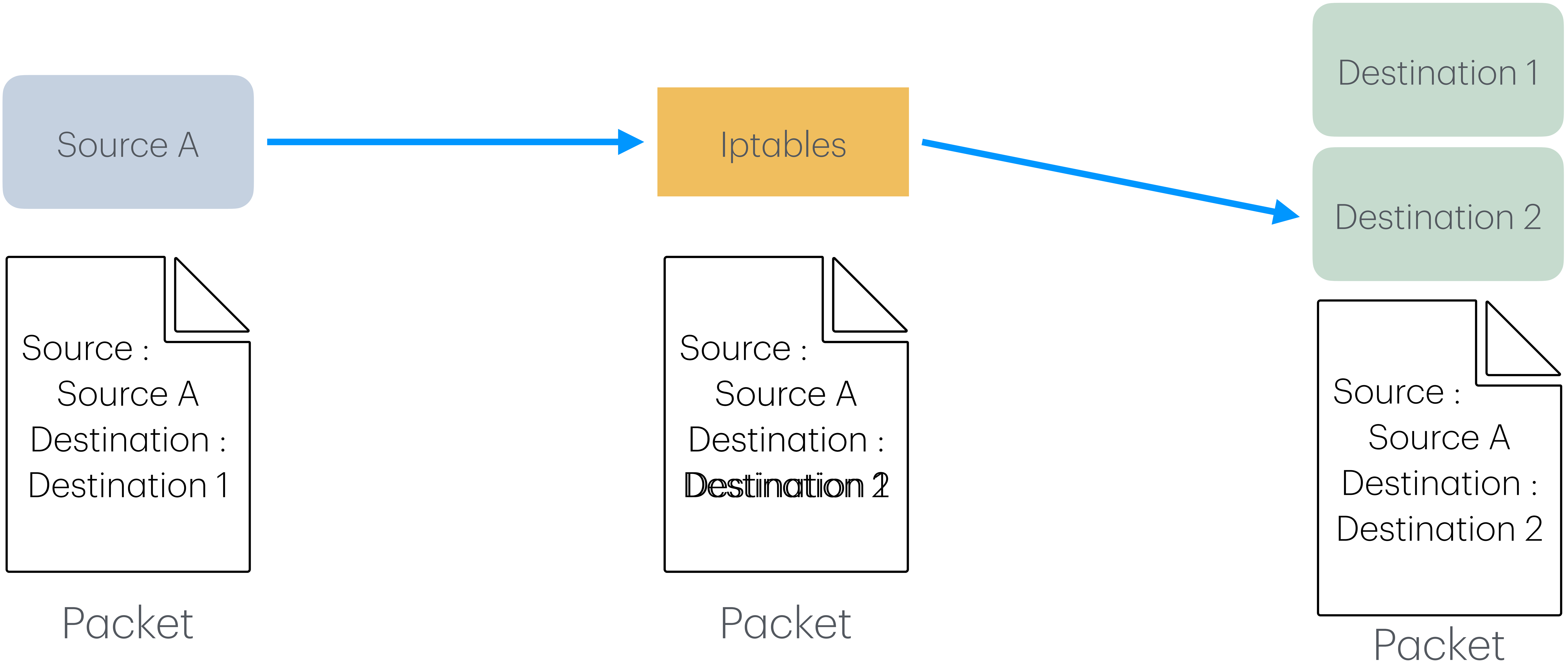
- This allows the set of Pods associated with a Service to change at any time — clients only need to know the Service’s virtual IP, which does not change.

netfilter

- Networking framework build in linux.
- Allows networking-related operations in the form of customised handlers.
- Functions:
  - Packet Filtering
  - Network Address Translation
  - Port Translation

iptables

- iptables is a user-space program providing a table-based system for defining rules for manipulating and transforming packets using the netfilter framework.



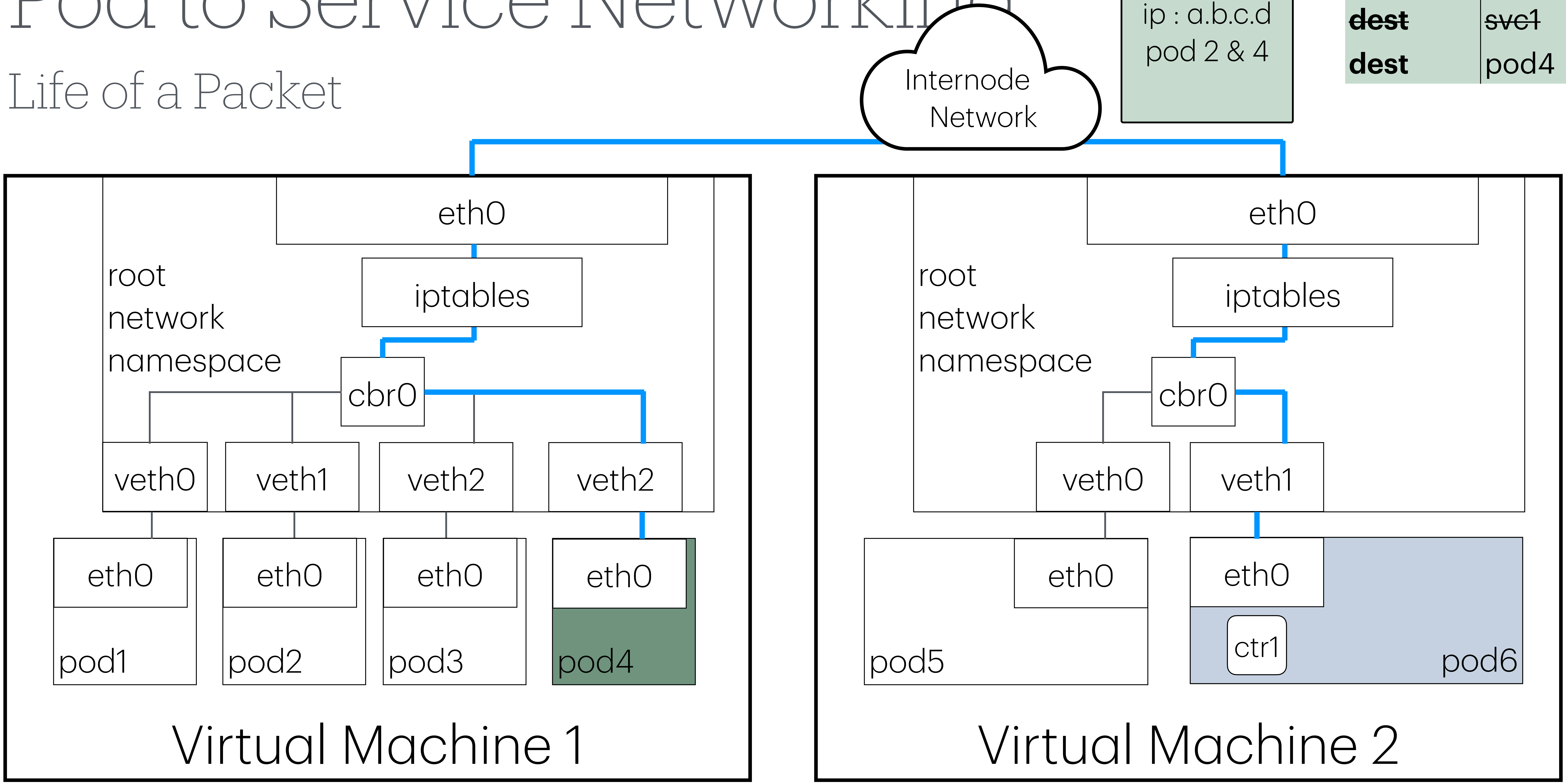
Pod to Service Networking

Pod to Service Networking

Life of a Packet

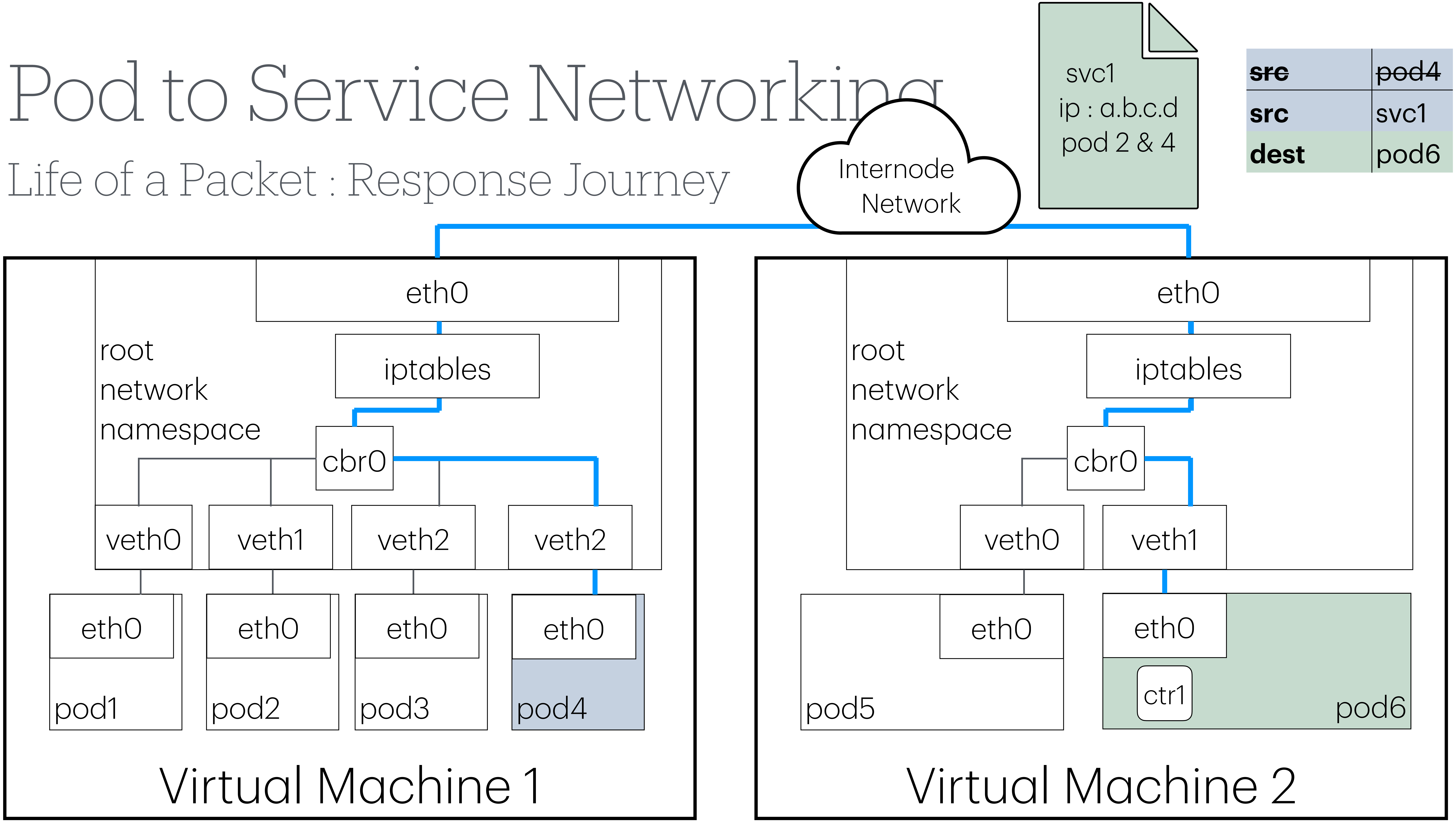
svc1
ip : a.b.c.d
pod 2 & 4

src	pod6
~~**dest**~~	~~svc1~~
dest	pod4



Pod to Service Networking

Life of a Packet : Response Journey



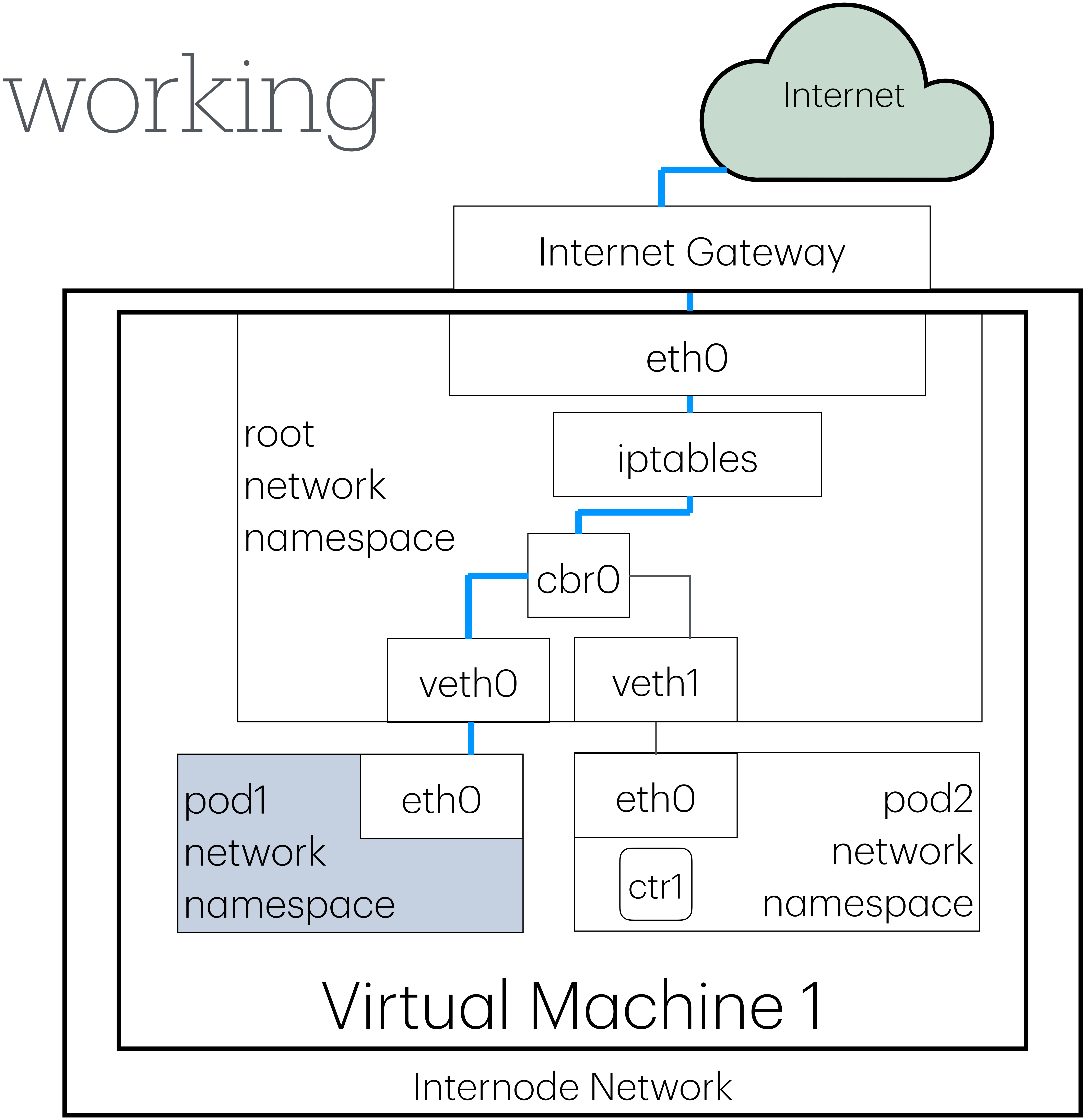
~~**src**~~	~~pod4~~
src	svc1
dest	pod6

Pod to Internet Networking

Pod to Internet Networking

Life of a Packet: Egress

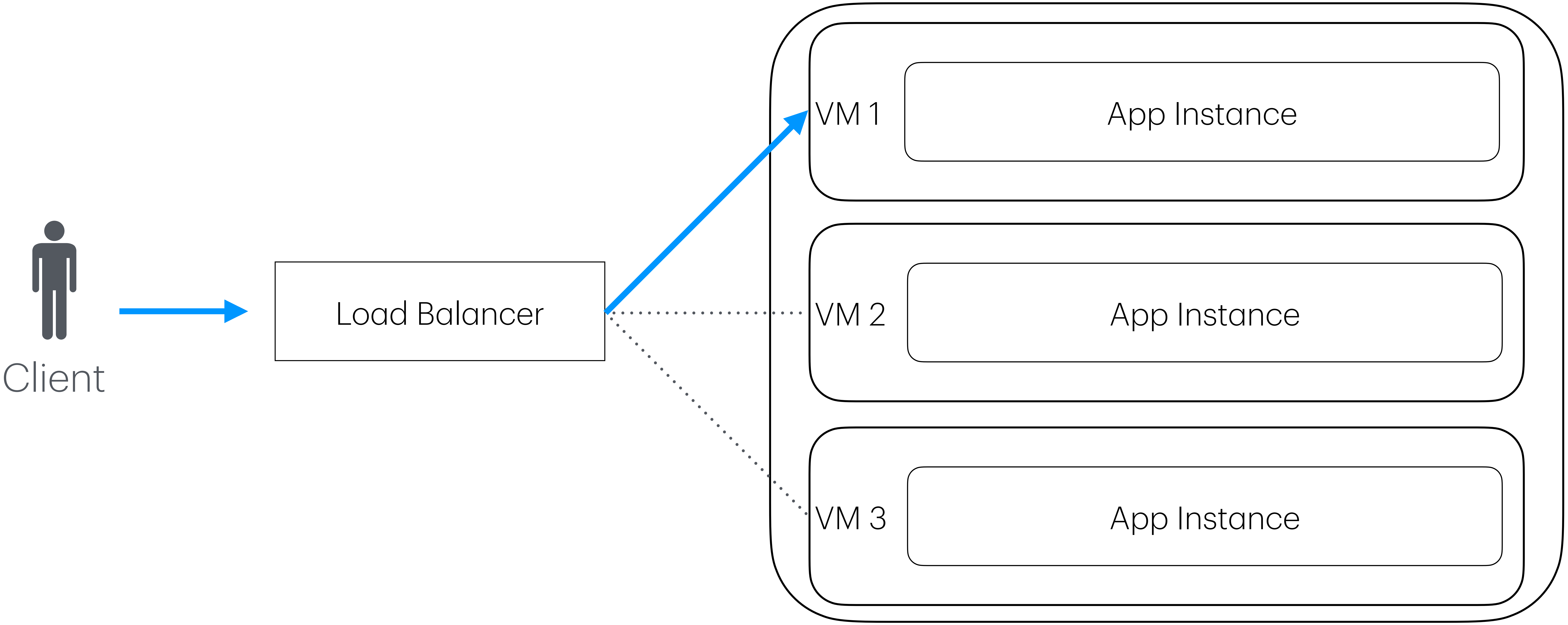
~~**src**~~	~~pod1~~
~~**src**~~	~~vm1 ip~~
src	internet gw ip
dest	internet



Internet to Pod Networking

Internet to Pod Networking

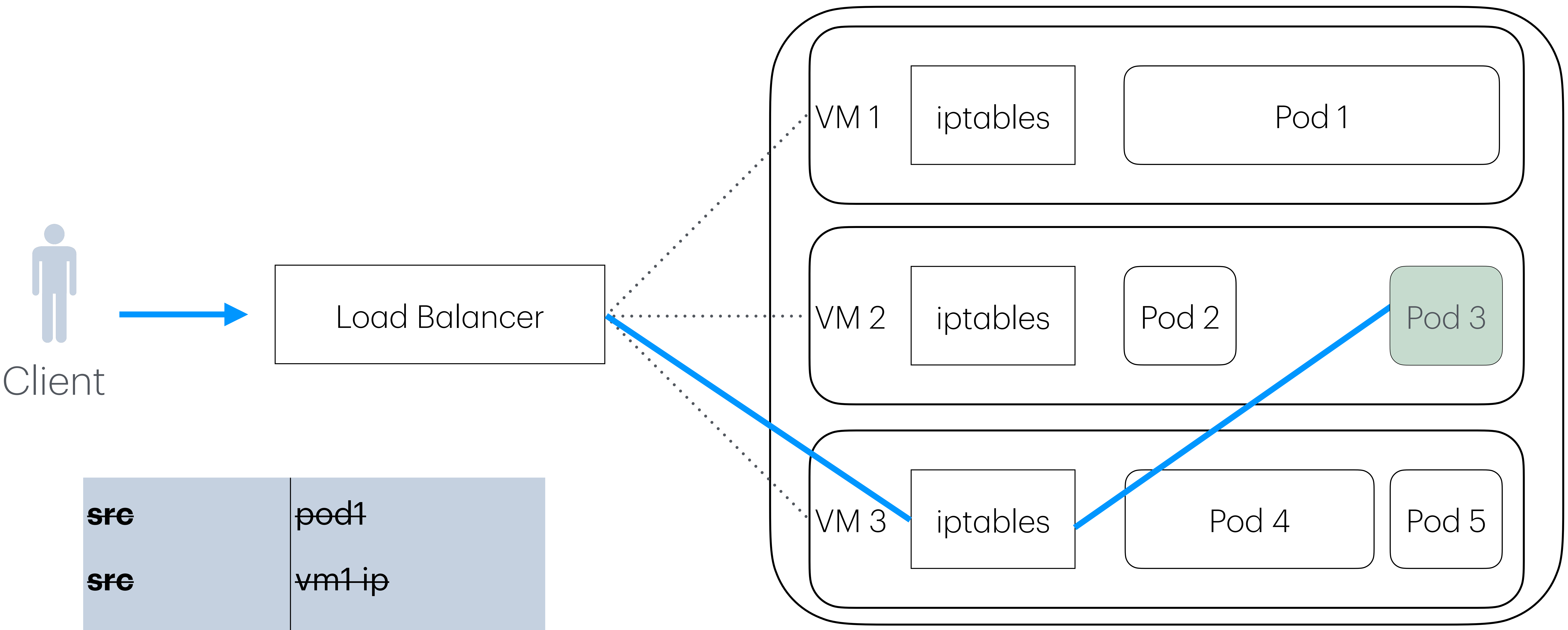
Life of a Packet: Traditional Servers



Internet to Pod Networking

Life of a Packet: Ingress (Layer 4 Load Balancer)

src	client
dest	pod3



~~**src**~~	~~pod1~~
~~**src**~~	~~vm1 ip~~
src	internet gw ip

Networking Types Summarised

Container to Container Networking	Through localhost in the same network namespace
Pod to Pod Networking	
Same Node	Through Bridge
Different Nodes	Through Internode network
Pod to Service Networking	Through iptables
Pod to Internet Networking (Egress)	Through Internet Gateway
Internet to Pod Networking (Ingress)	
Layer 4 Load Balancer	Through NodePort
~~Layer 7 Ingress Controller~~	

Thank You!