#RP30

Automated Deployment and Scaling of Named Data Networks in Cloud Environments

Sean Liao
Supervisor: Zhiming Zhao

Next 20 mins...

Some academic(s) arrive to tell us
that (once again) they have Fixed the Internet,
and (once again) it runs on top of the current actually-working internet,
and (once again) if you sign up you can communicate with as many as twelve other computers.

n-gate.com *in reference to SCION*

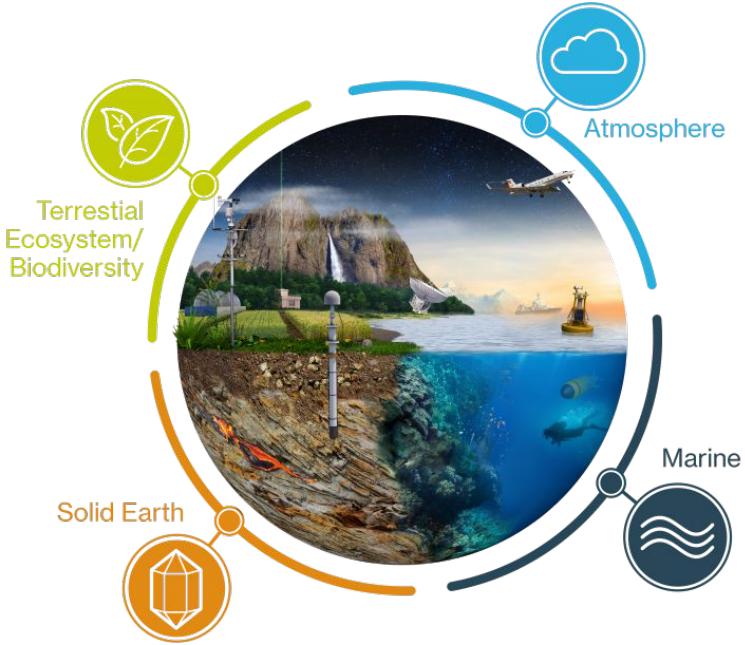
Introduction

ENVRI-FAIR

ENVironmental Research Infrastructures (ENVRI)

connection to

European Open Science Cloud (EOSC)



Single Domain

AnaEE
DiSSCO
ELIXIR
EMPHASIS
INTERACT

EPOS

EUROFLEETS
EURO-ARGO
JERICO-RI
SEADATANET

ACTRIS
ARISE
EISCAT_3D
EUFAR
EUROCHAMP 2020
HEMERA
IAGOS

Multi Domain

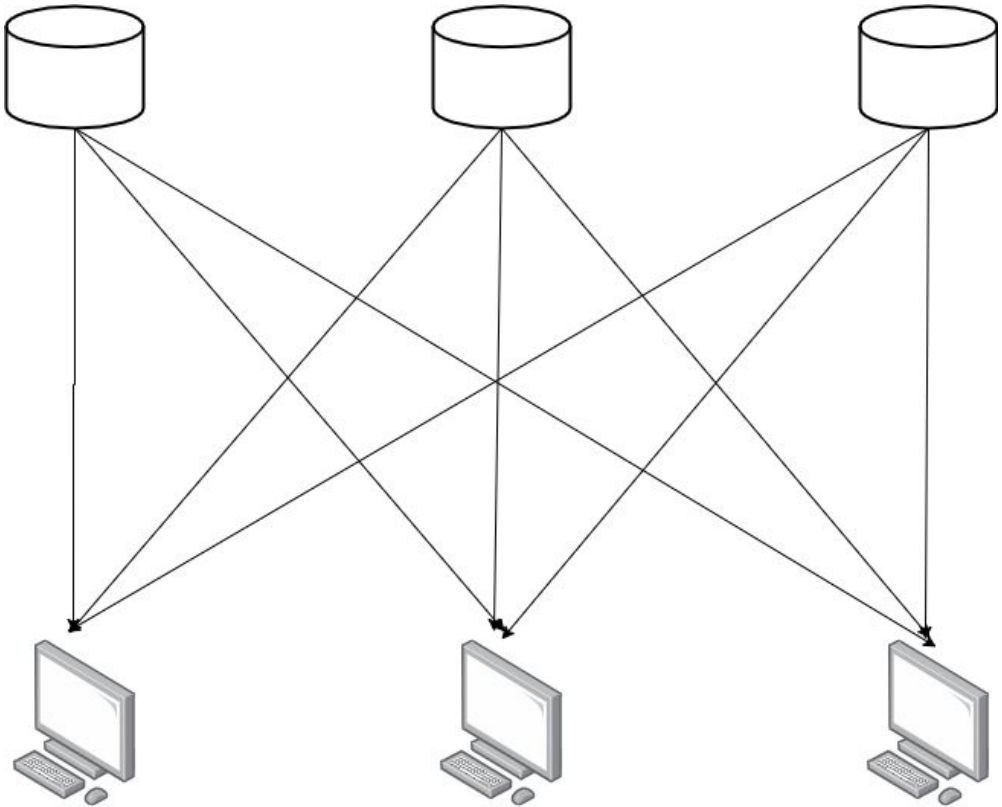
EuroGOOS
ICOS
IS-ENES
SIOS

AQUACOSM
DANUBIUS
eLTER
EMBRC
LifeWatch

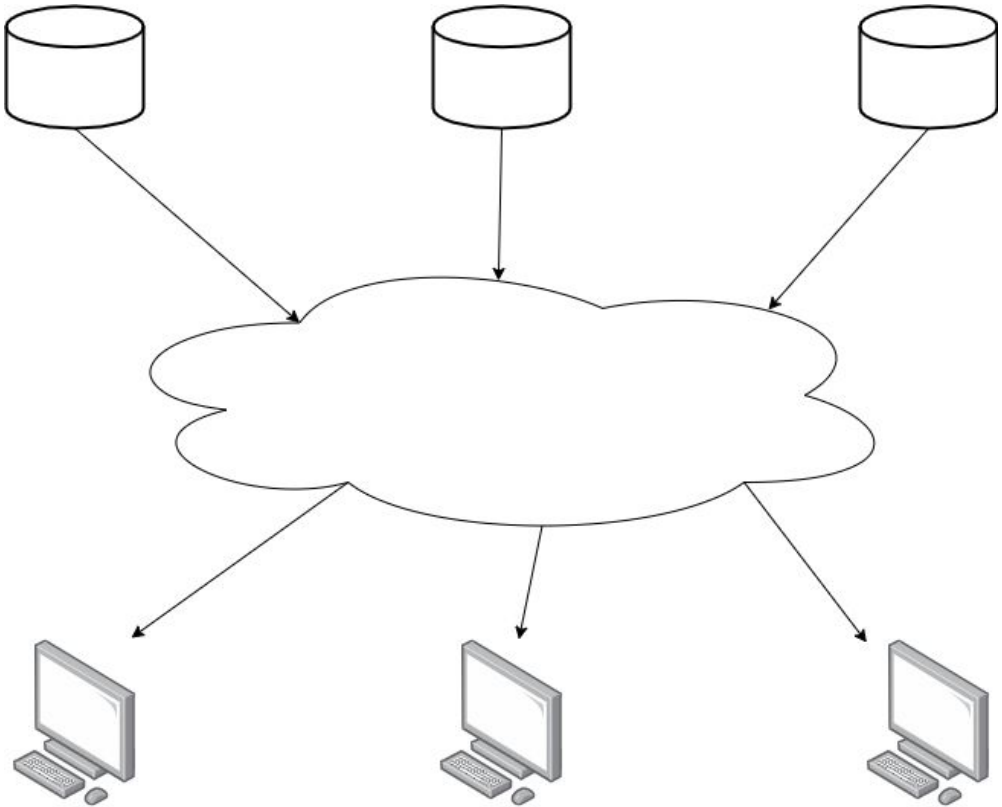
EMSO

https://envri.eu/about-envri-fair/

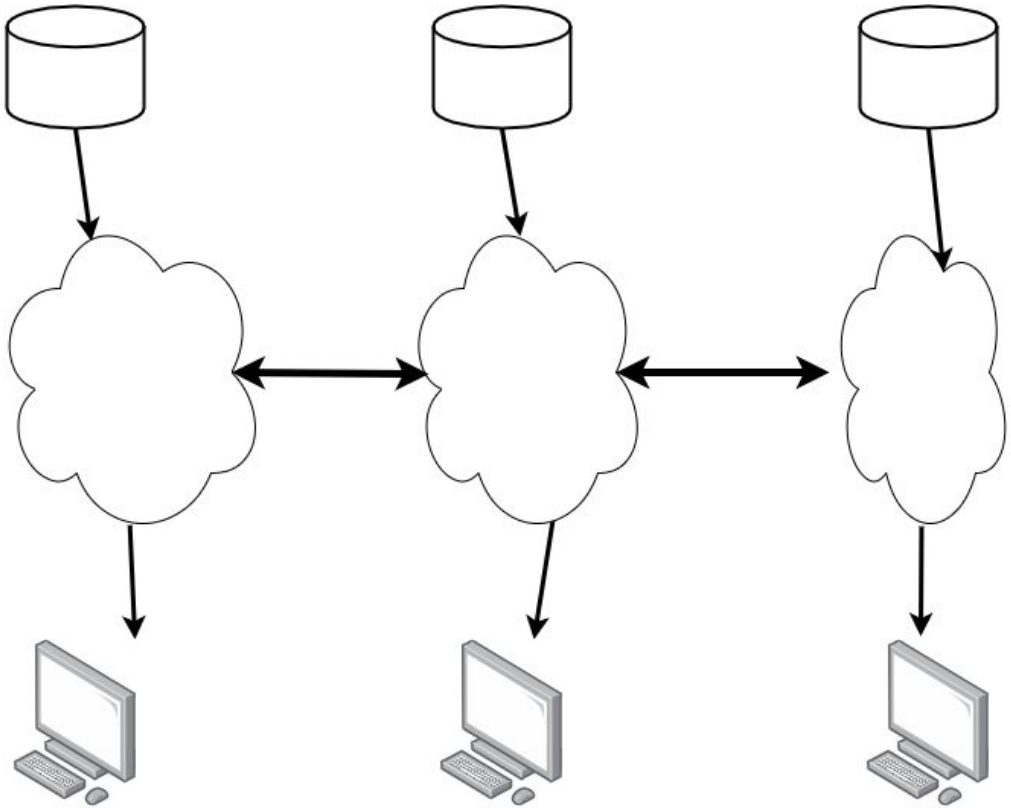
Data distribution



Content Distribution Network



Federated Clouds



Named Data Networking (NDN)

Named Data Networking

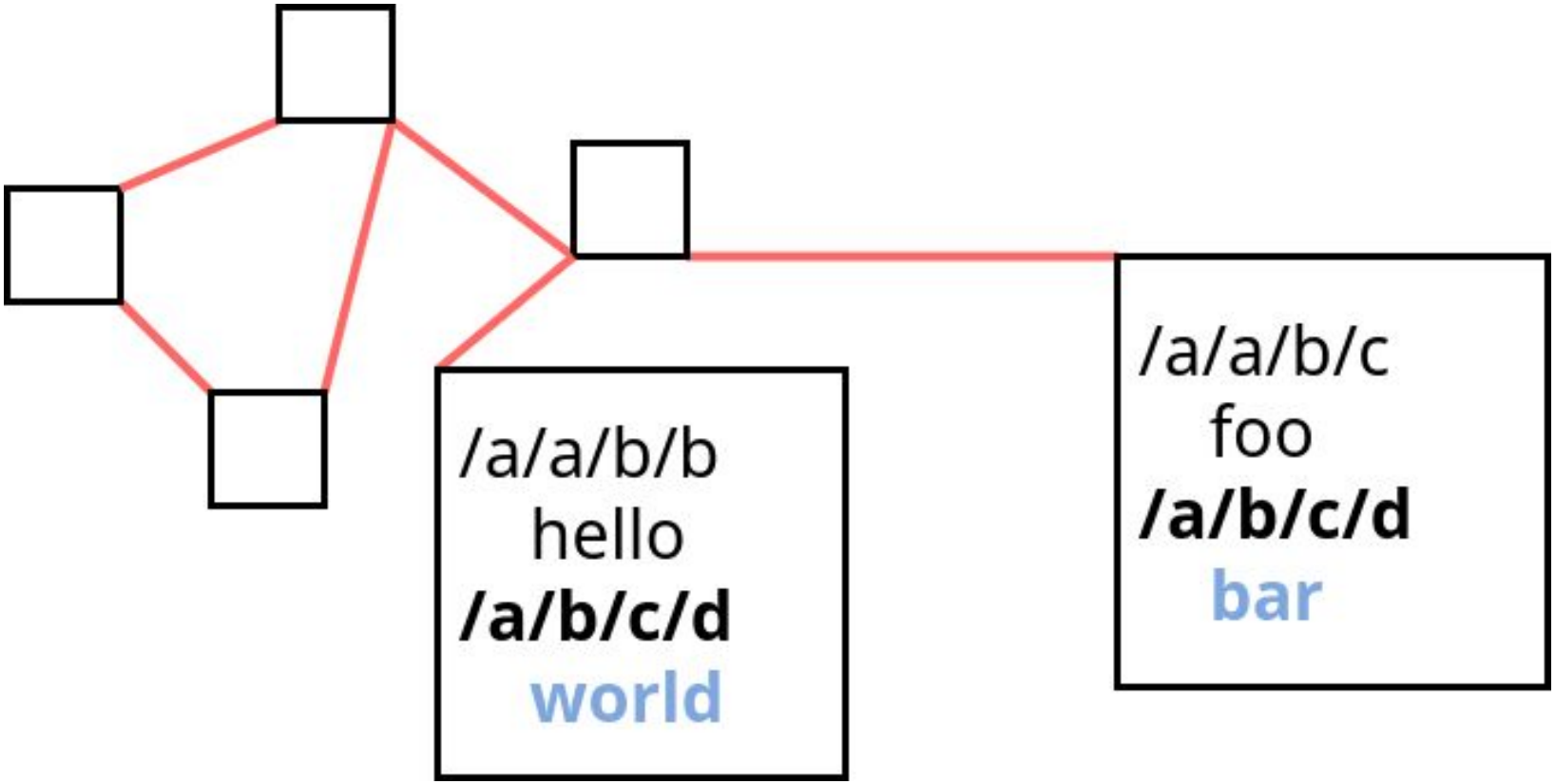
NSF Future Internet Architecture Program

Information distribution network

Potential benefits:

- content caching
- network level security of data

1.2.3.4/a/b/c/d



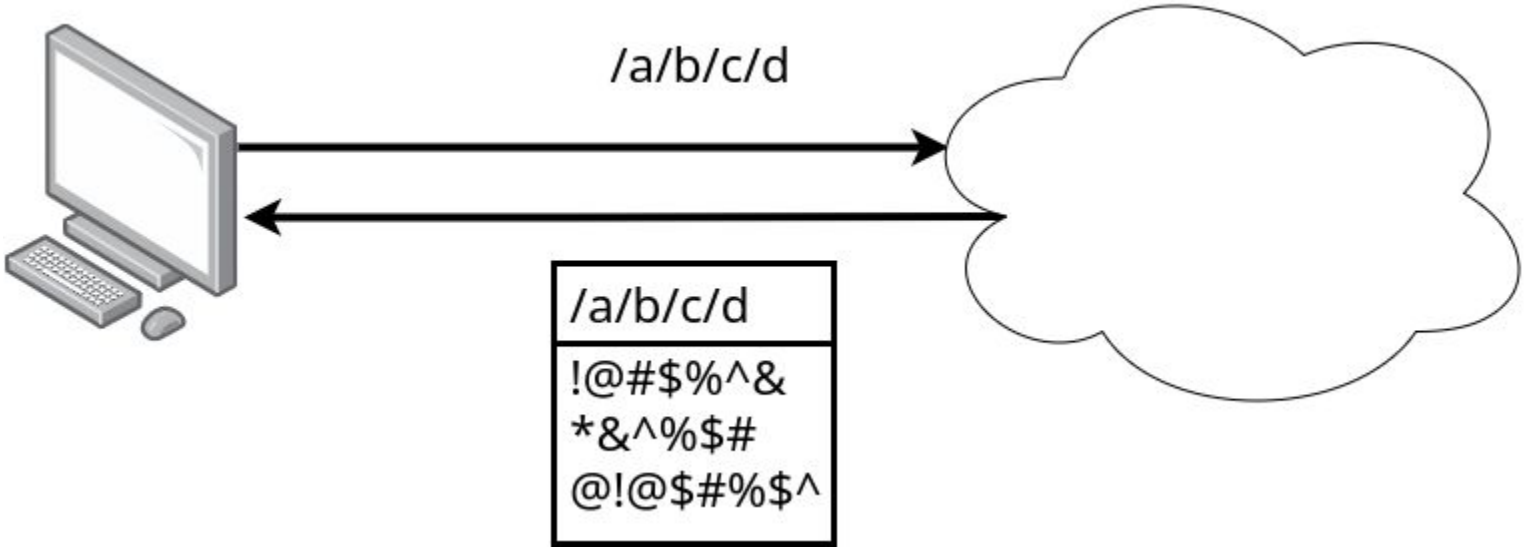
Route directly to data

IPv4: 1.2.3.4

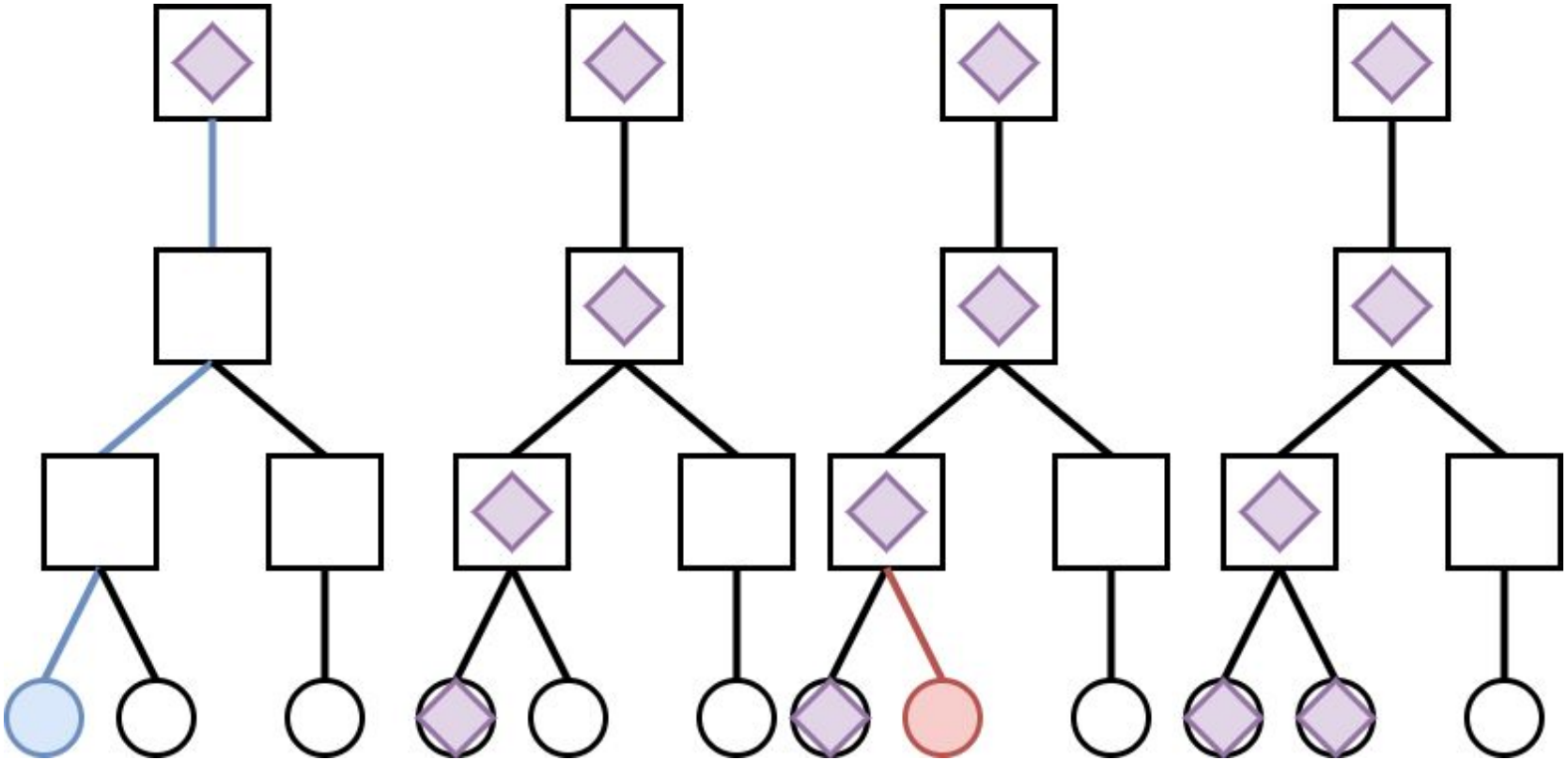
IPv6: 1:2:3:4:5:6:7:8

NDN: /arbitrary/strings/infinite/address/space

1-1 Request-Response (Interest-Data)



In network caching



Run it

In the Cloud

Overlay over IP

Simplify deployment

Scalable

Existing Tools

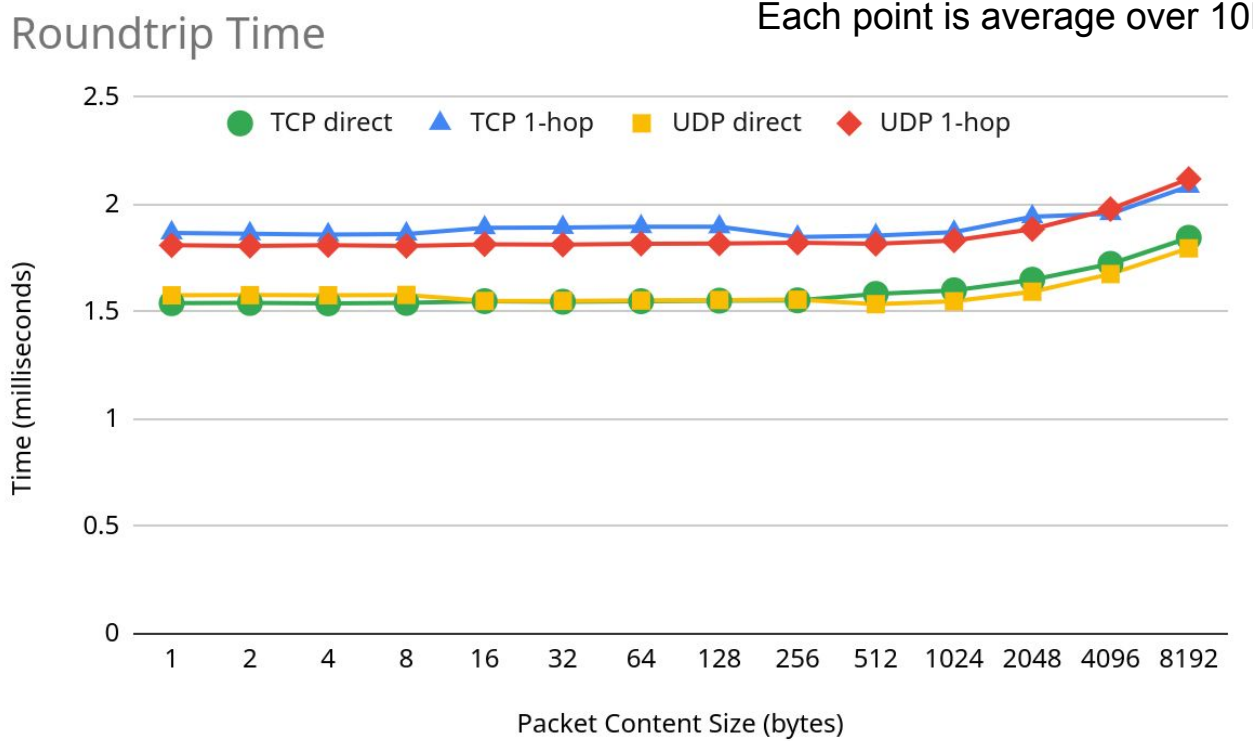
Router: NFD

Link State Routing: NLSR
Dynamic route updates on static network

Connectivity: FCH
Find closest hub/gateway

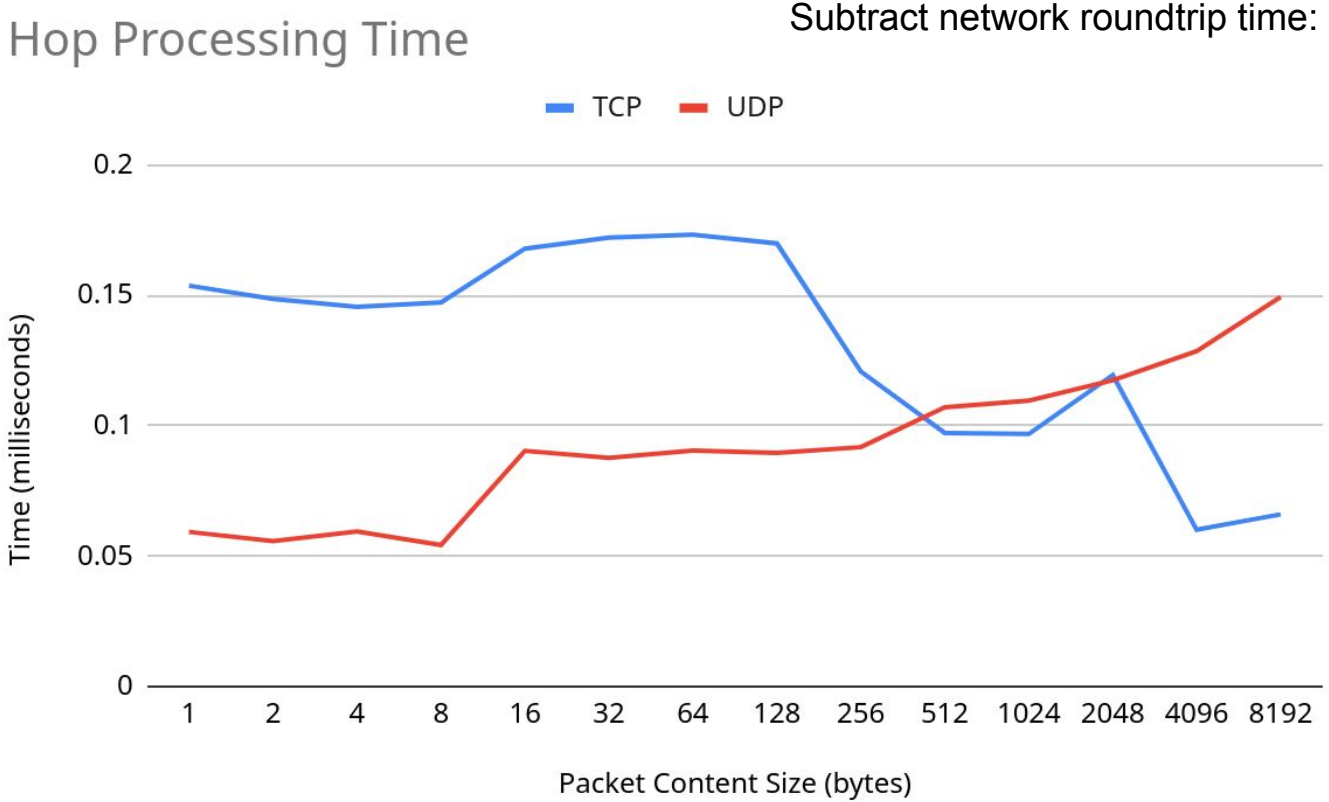
Overlay over TCP / UDP

Each point is average over 10K requests



Processing Overhead

Subtract network roundtrip time: 0.174ms

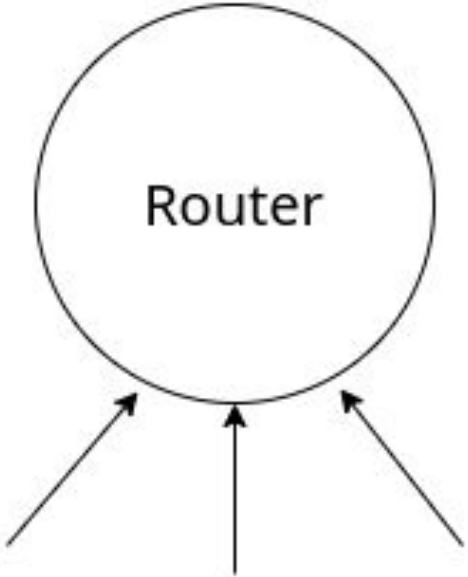


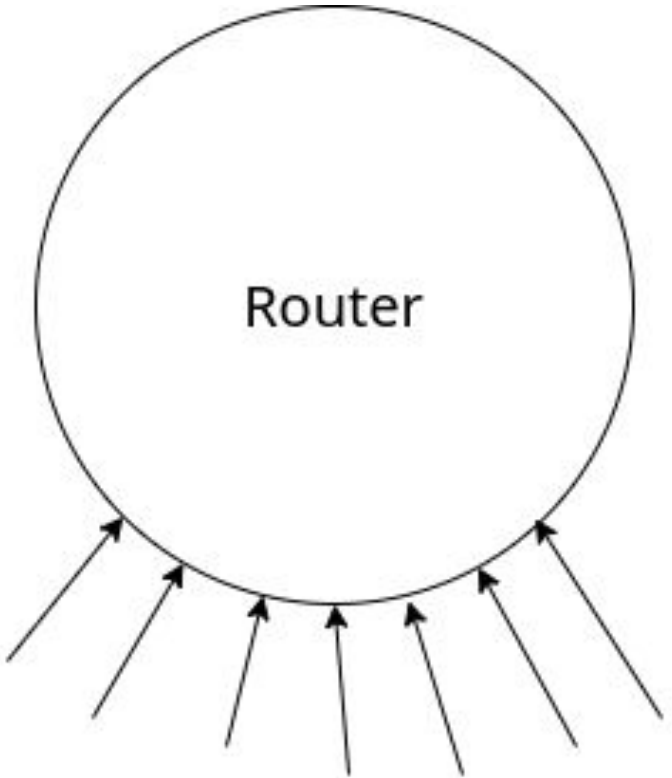
Testing notes

Performance severely degrades with 50000+ cached / in-flight requests

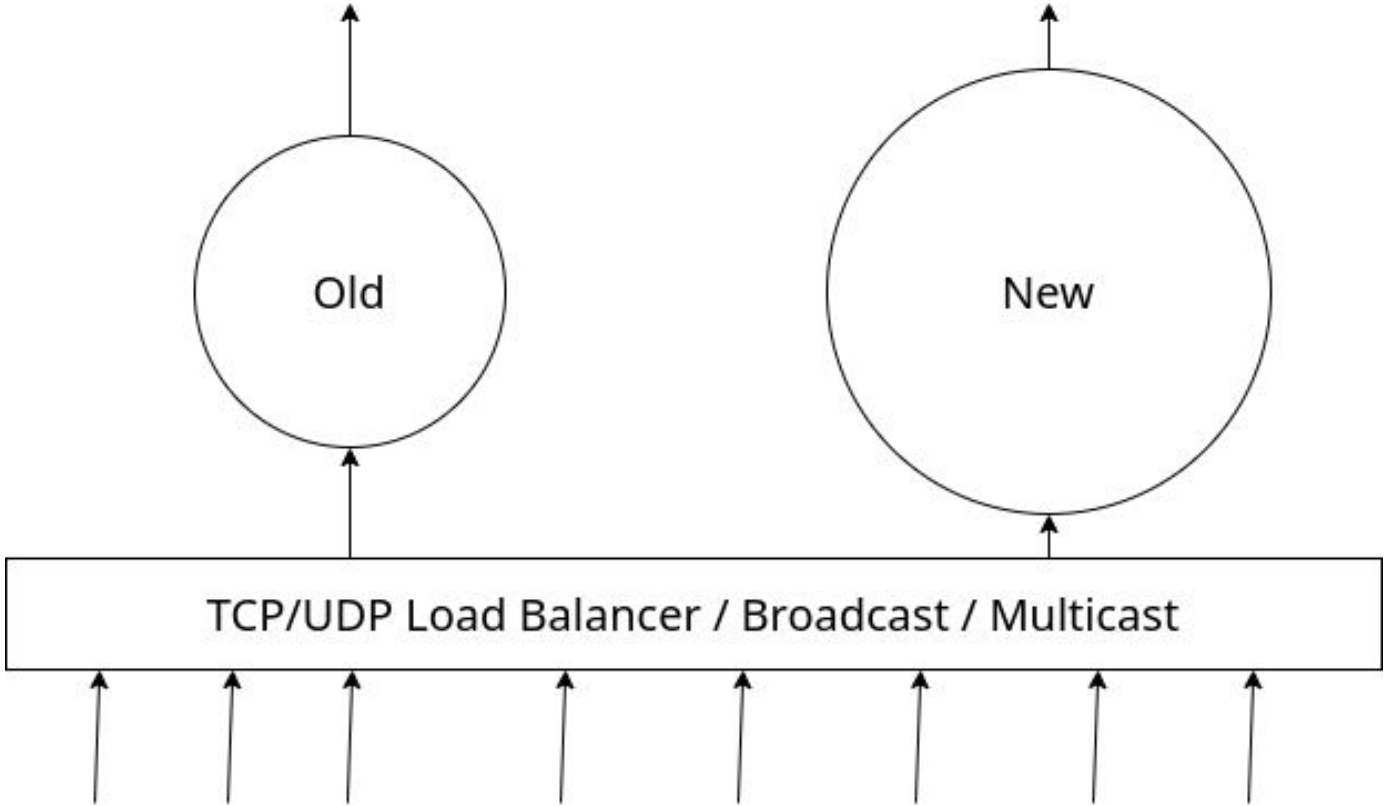
Scaling Up

Problem: Growing a node

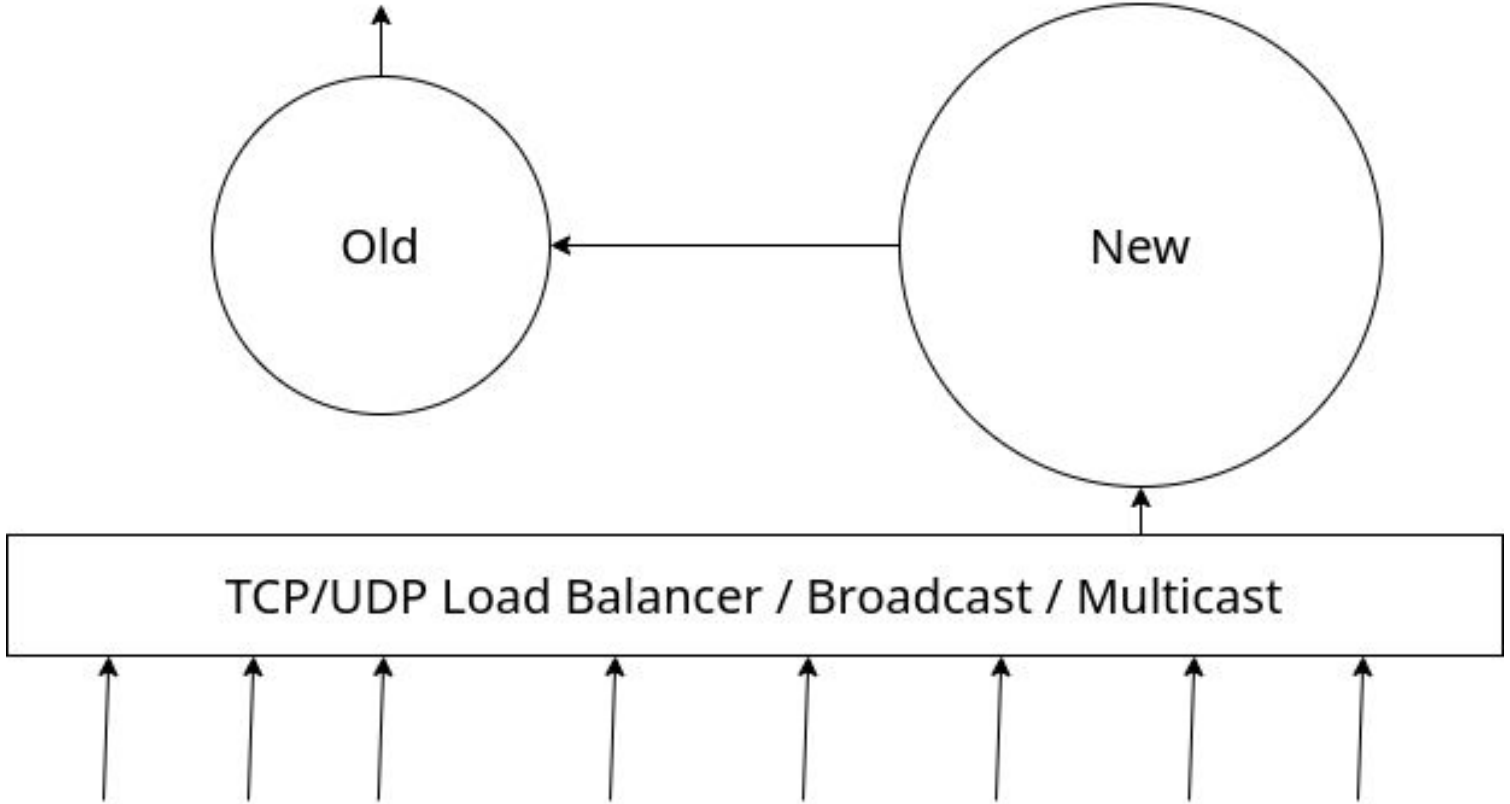




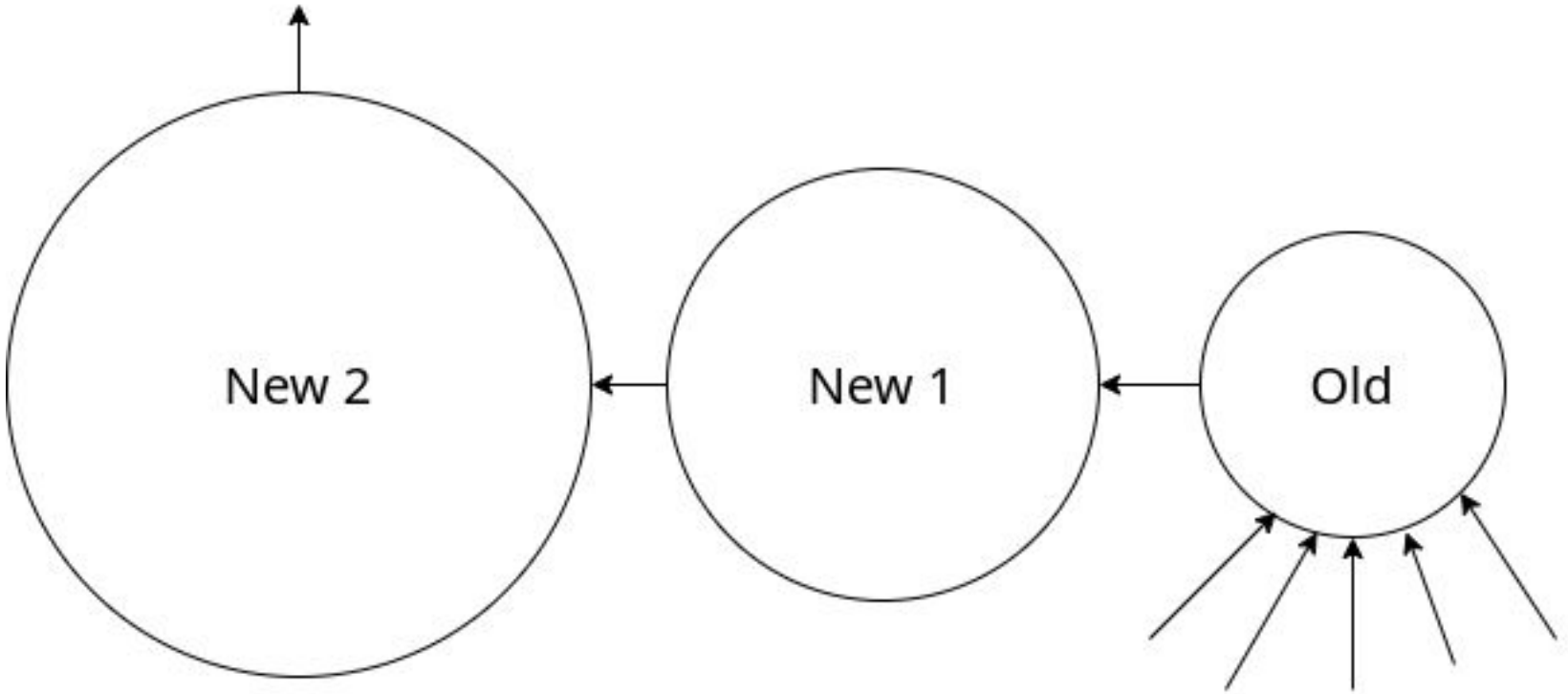
Replace



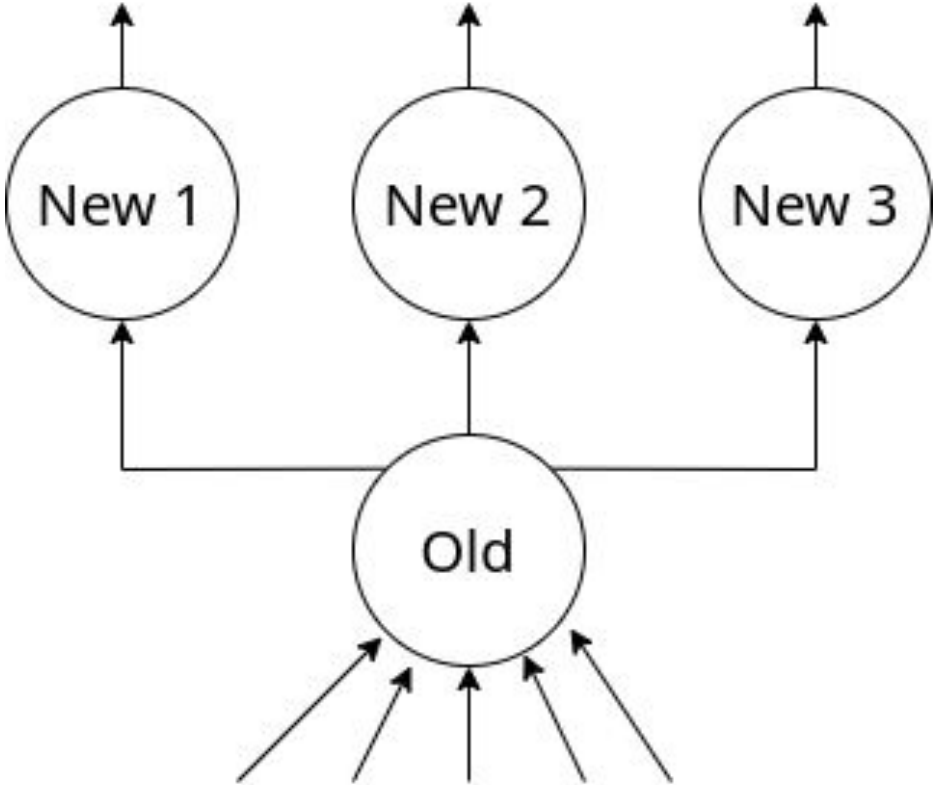
New Node in Front



New Node Behind (Chain)

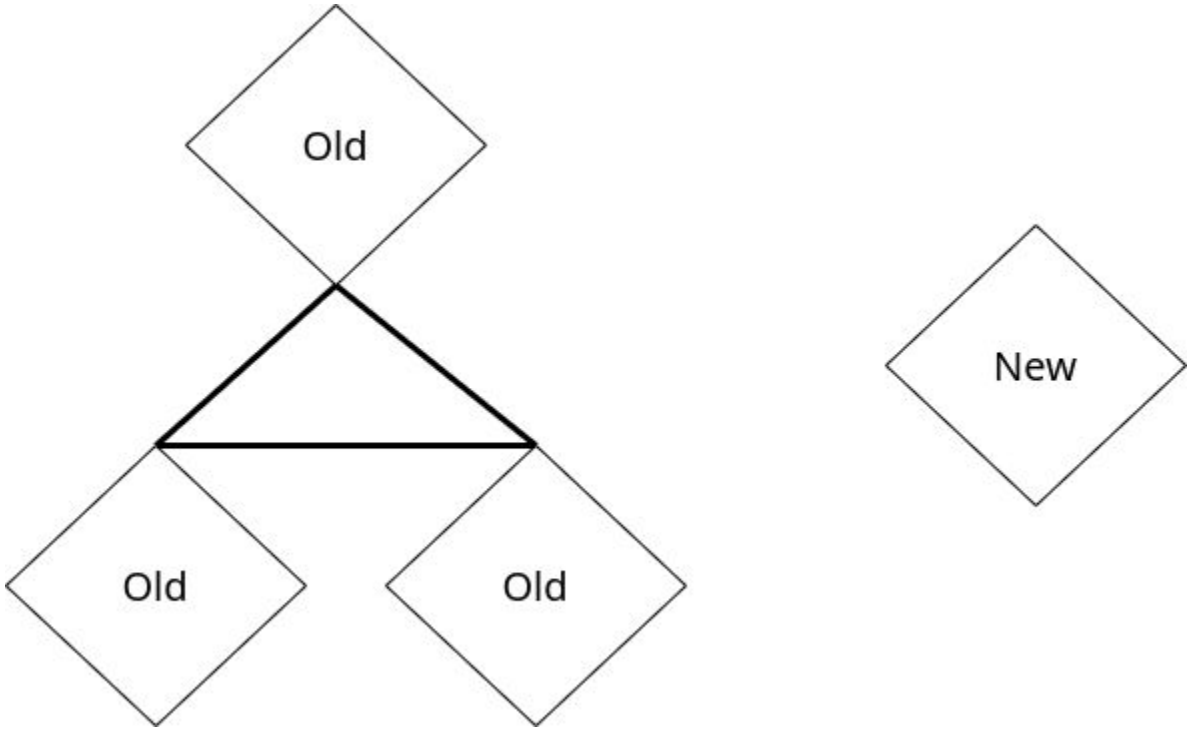


New Node Behind (Load Balancing)

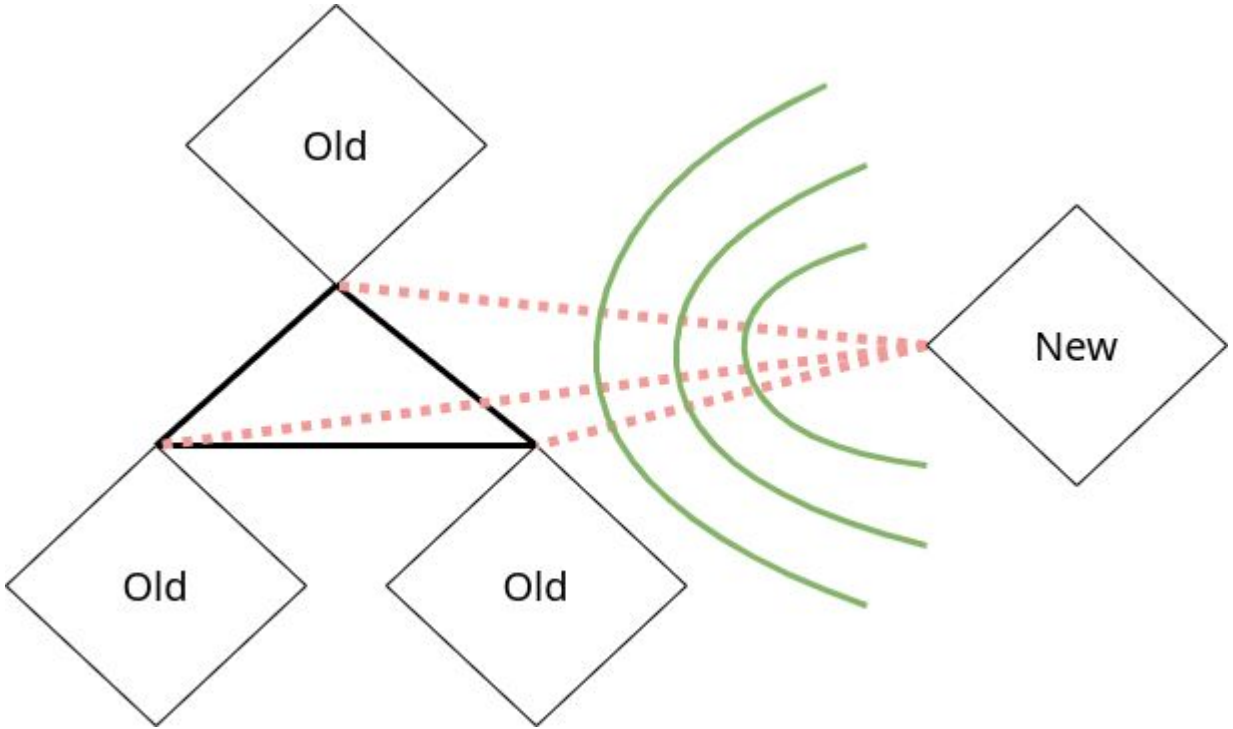


Growing the Network

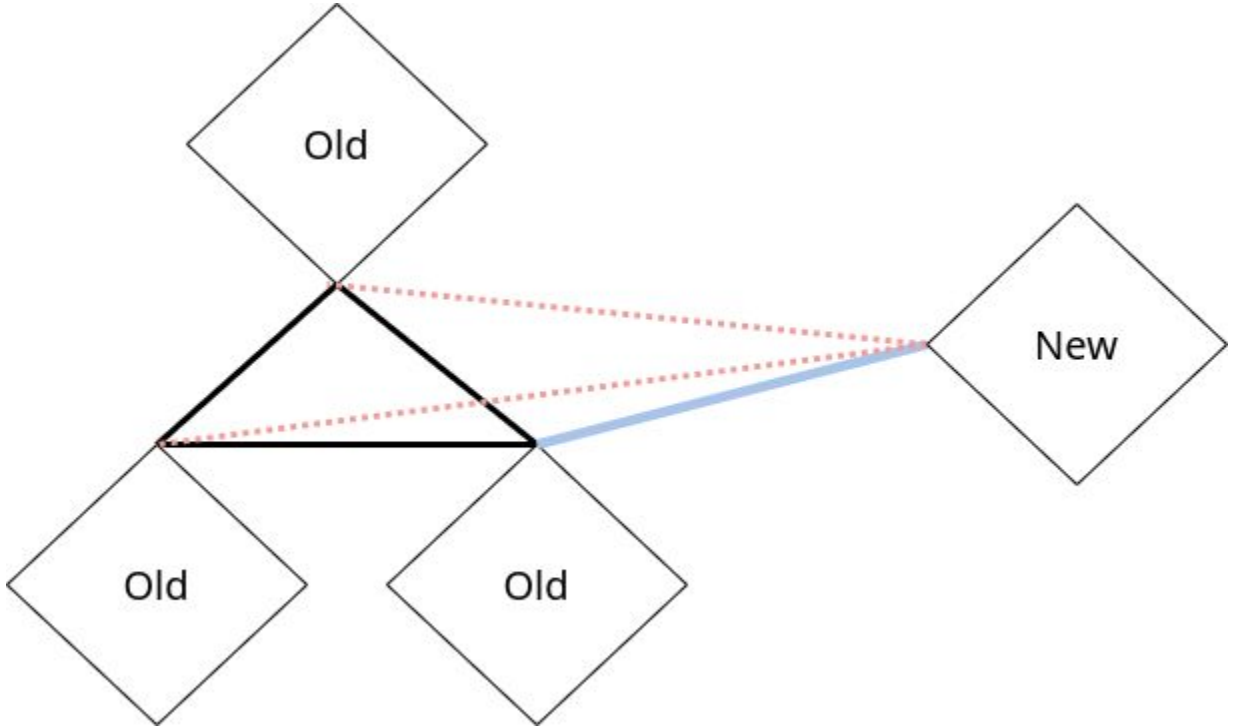
Problem: New Node



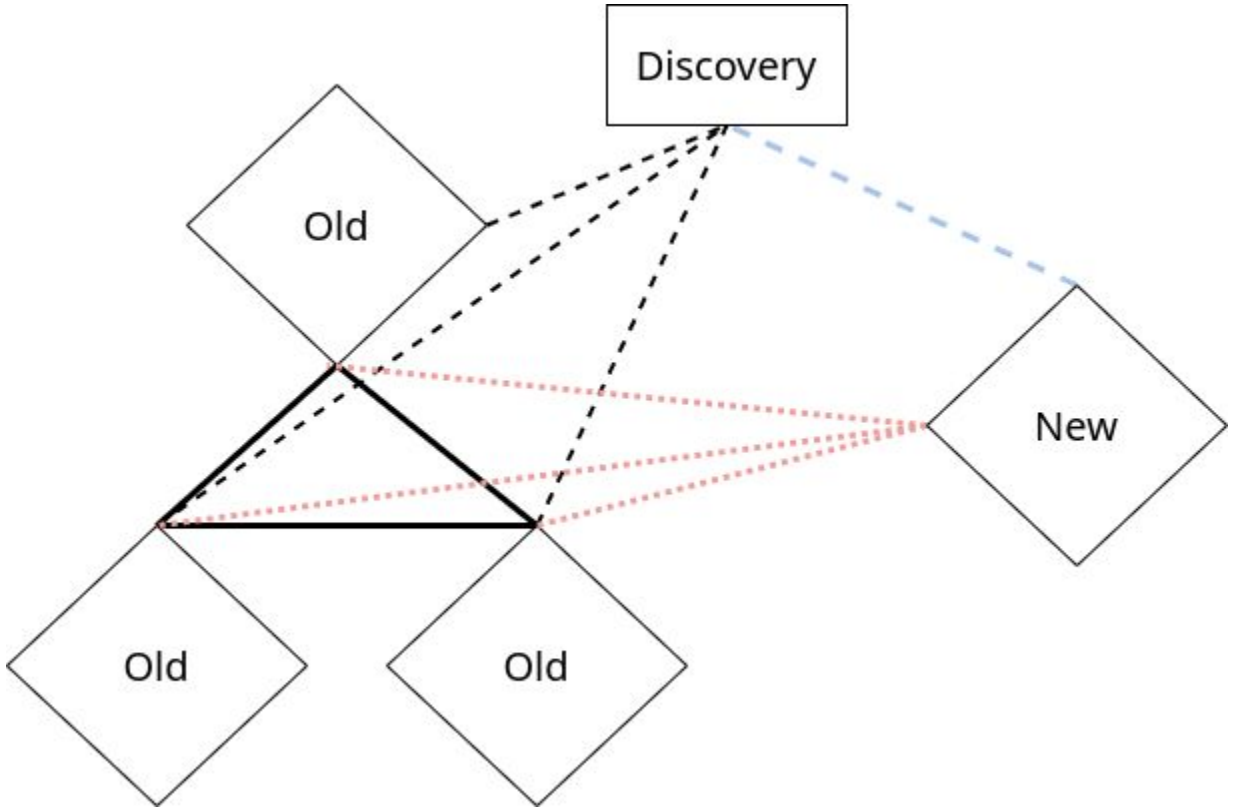
Broadcast / Multicast



Bootstrap Gossip

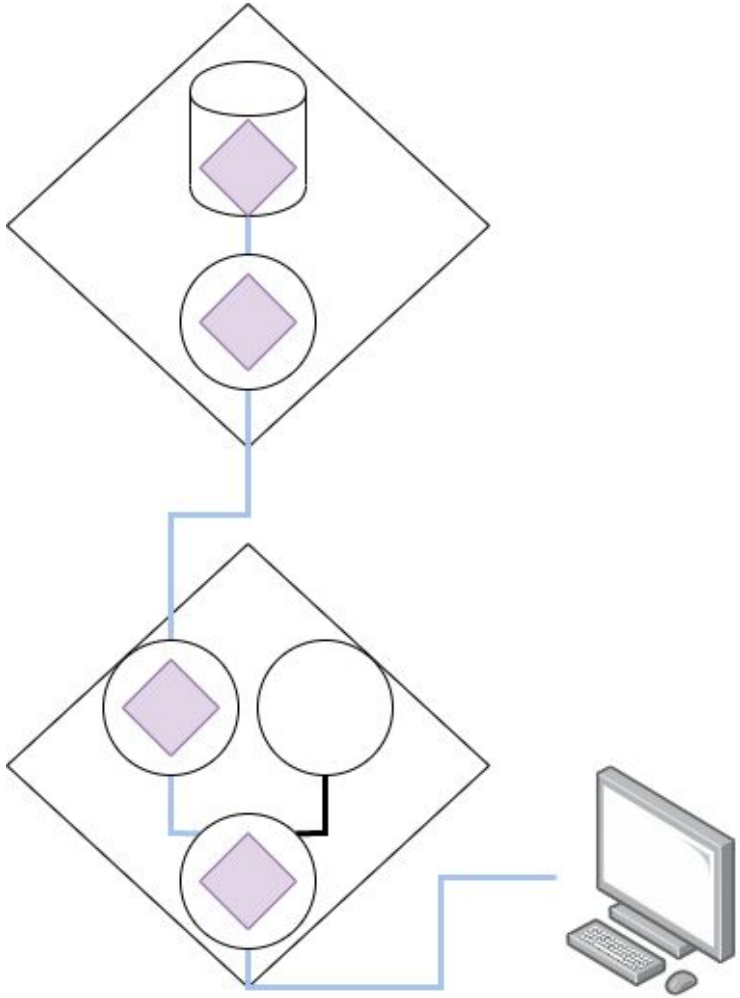


Central Discovery Server

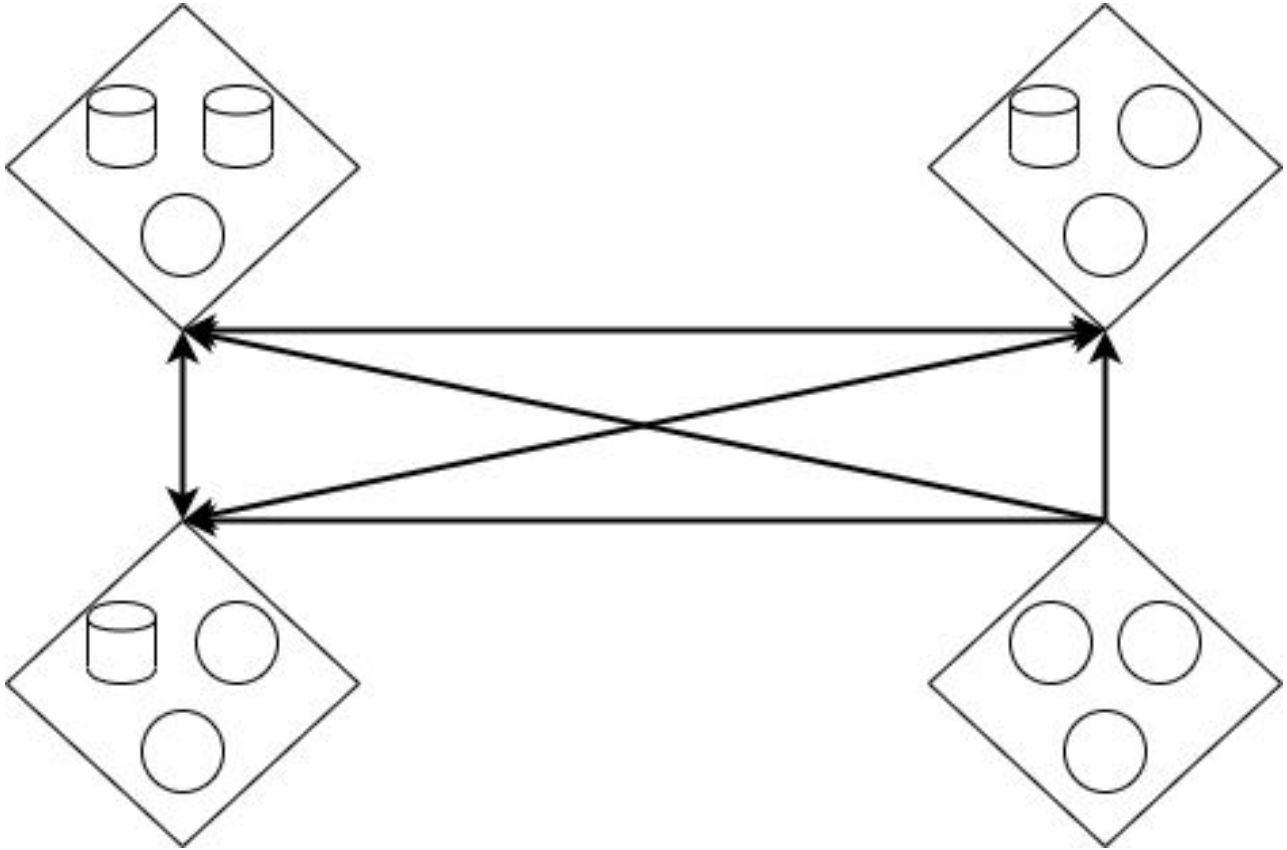


Network Architecture

3 Layer Cache

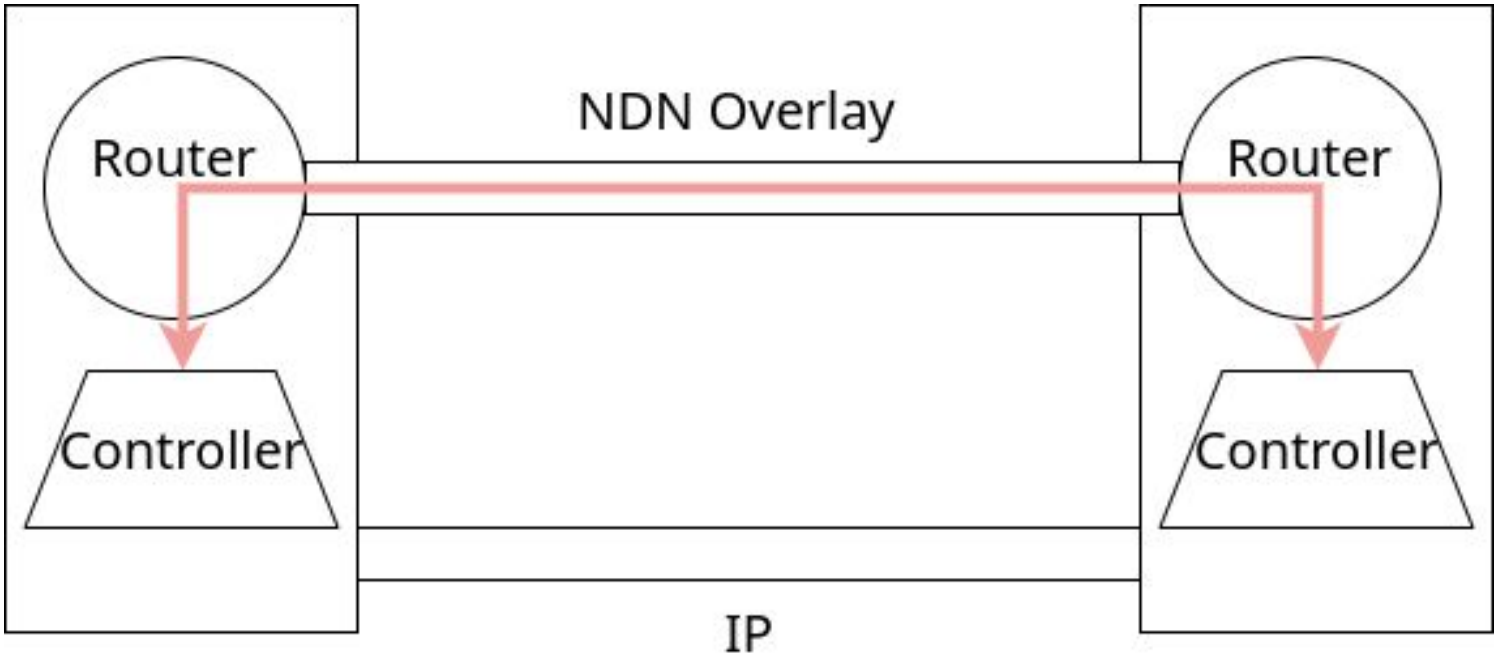


Mesh Network

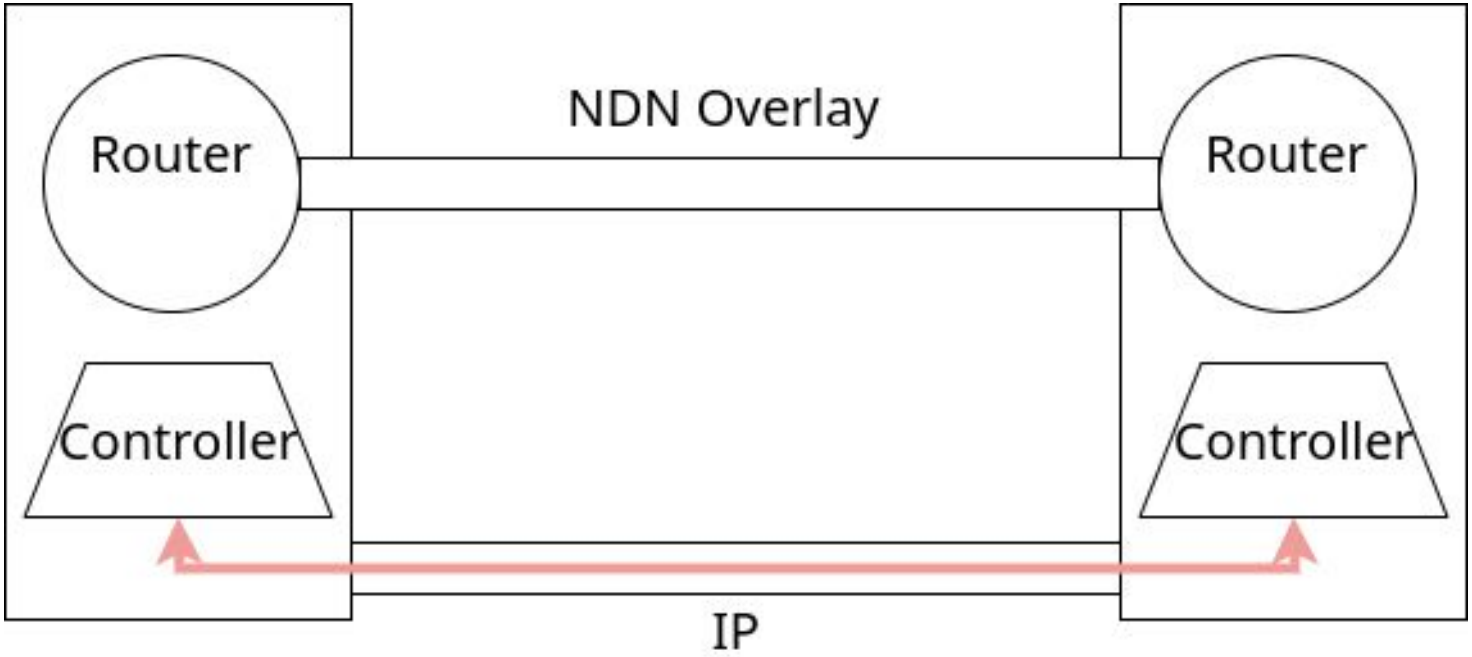


Automation

In-band Management



Out of Band Management



Proof of Concept

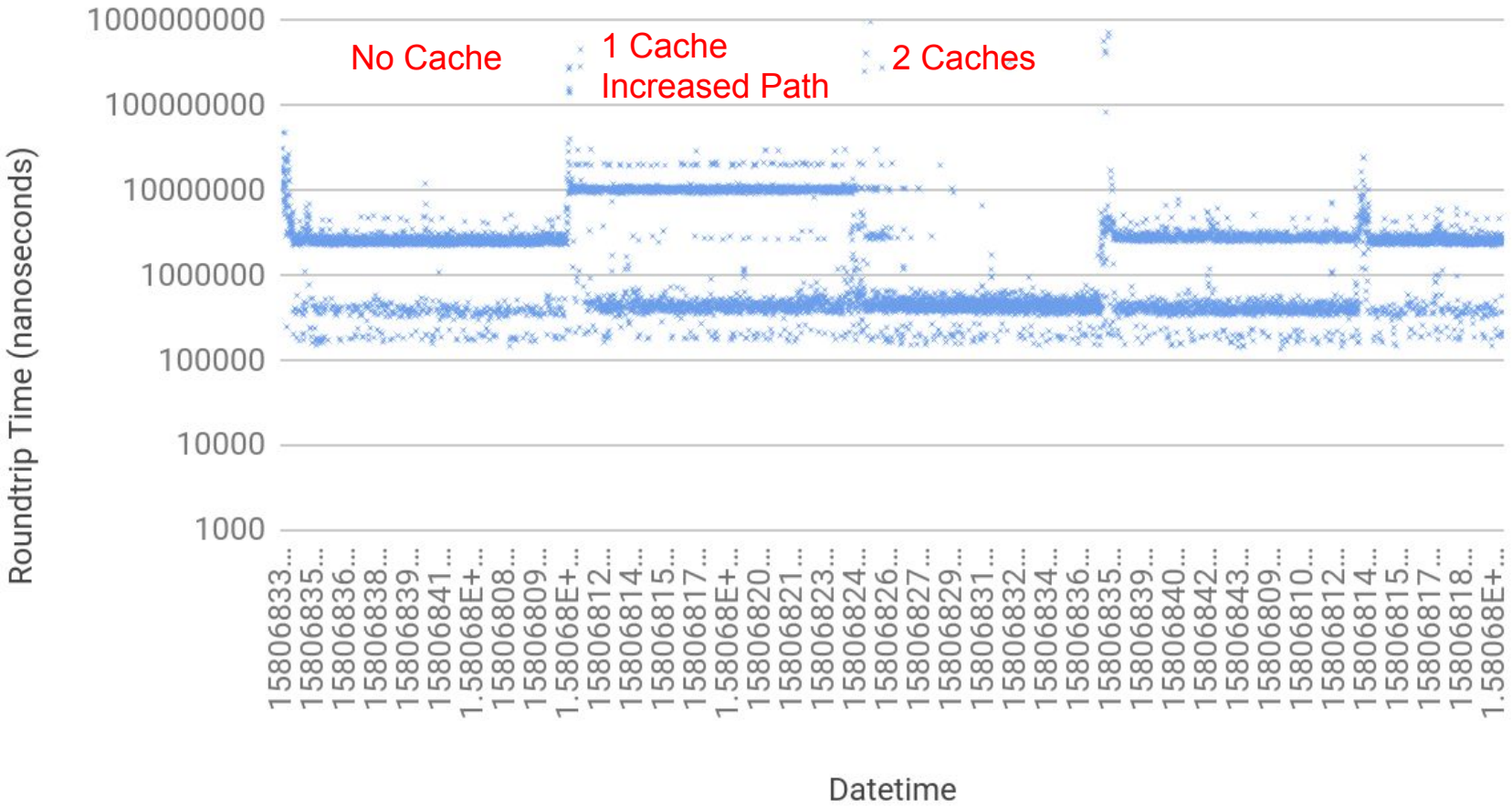
Necessary Configuration

Discovery Server
- Write down its address / Give it a preconfigured address

Load Balancer
- Address of Discovery Server
- Write down its address / Give it a preconfigured address

Caching Server
- Address of Load Balancer

Roundtrip Time



Proof of Concept Performance

Scraping state of router through CLI (on a timer)

TCP connections propagate routes and updates

Coarse grained partitioning of routes

Conclusion

Does it work?

Reuse existing router in a load balancing configuration

Minimal configuration, self connecting network

Lessons Learned

Now

Rapidly evolving research testbed for new ideas

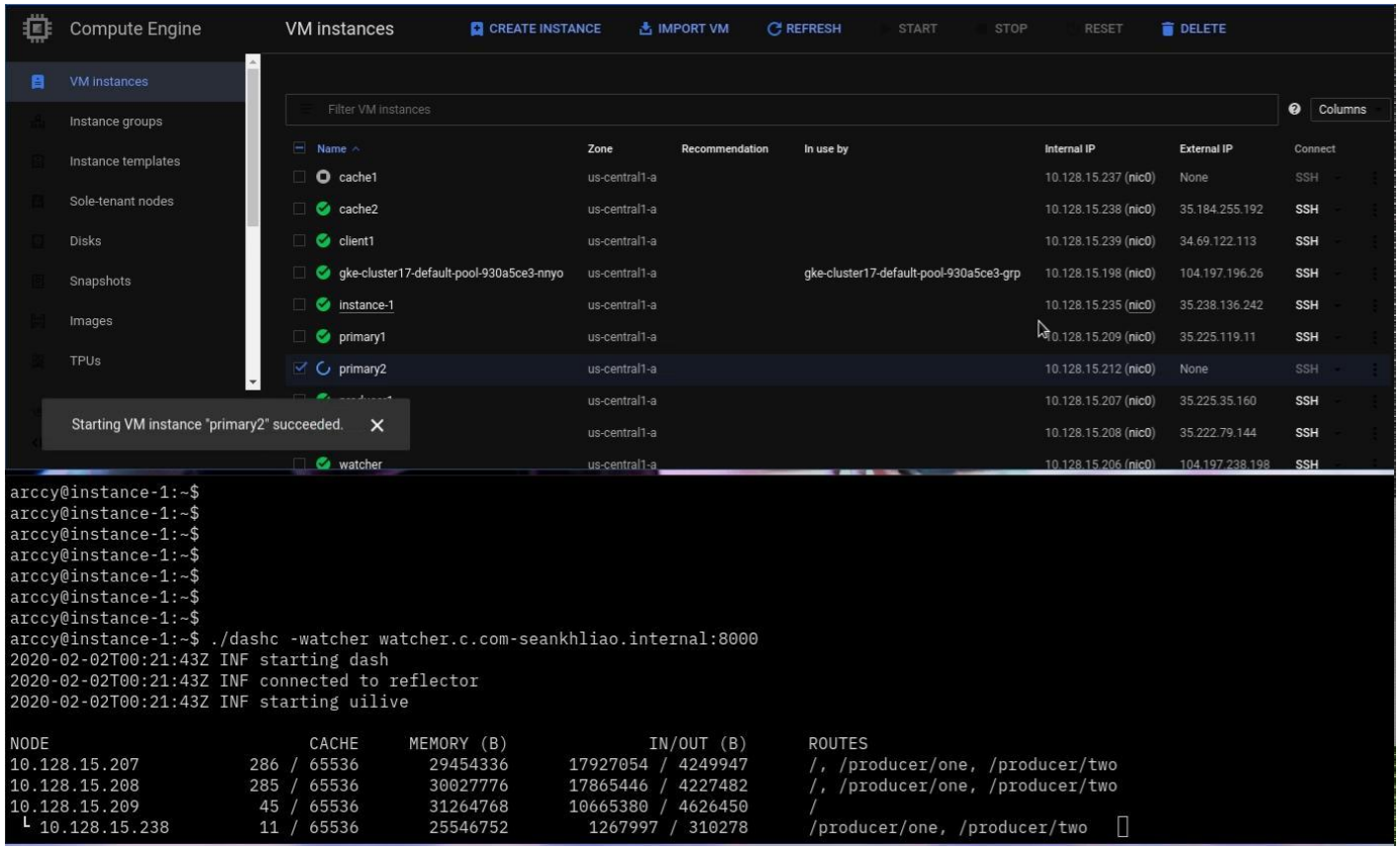
Things break or are not optimized

Future

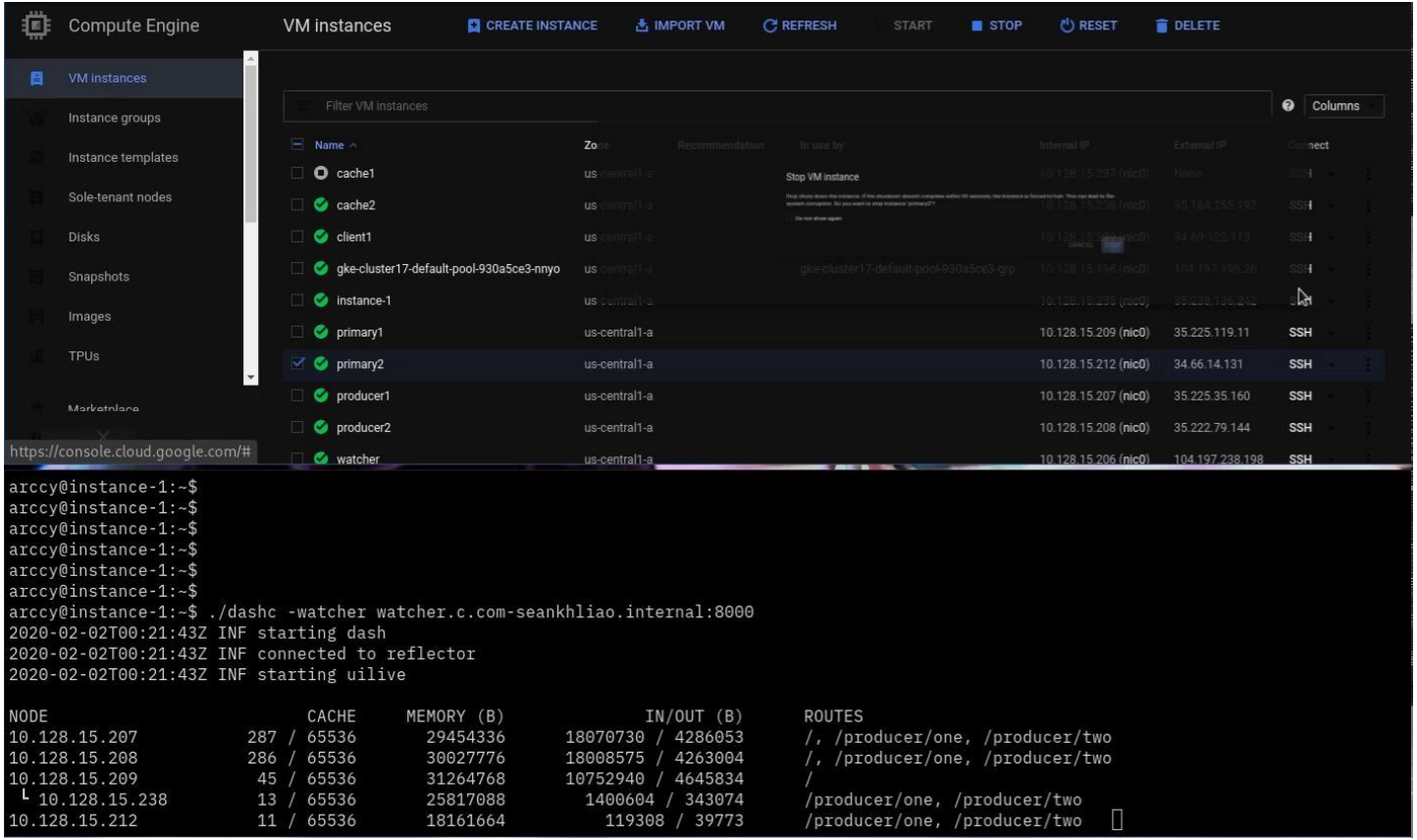
Some new network based on these ideas

Or maybe not (IPv6…)

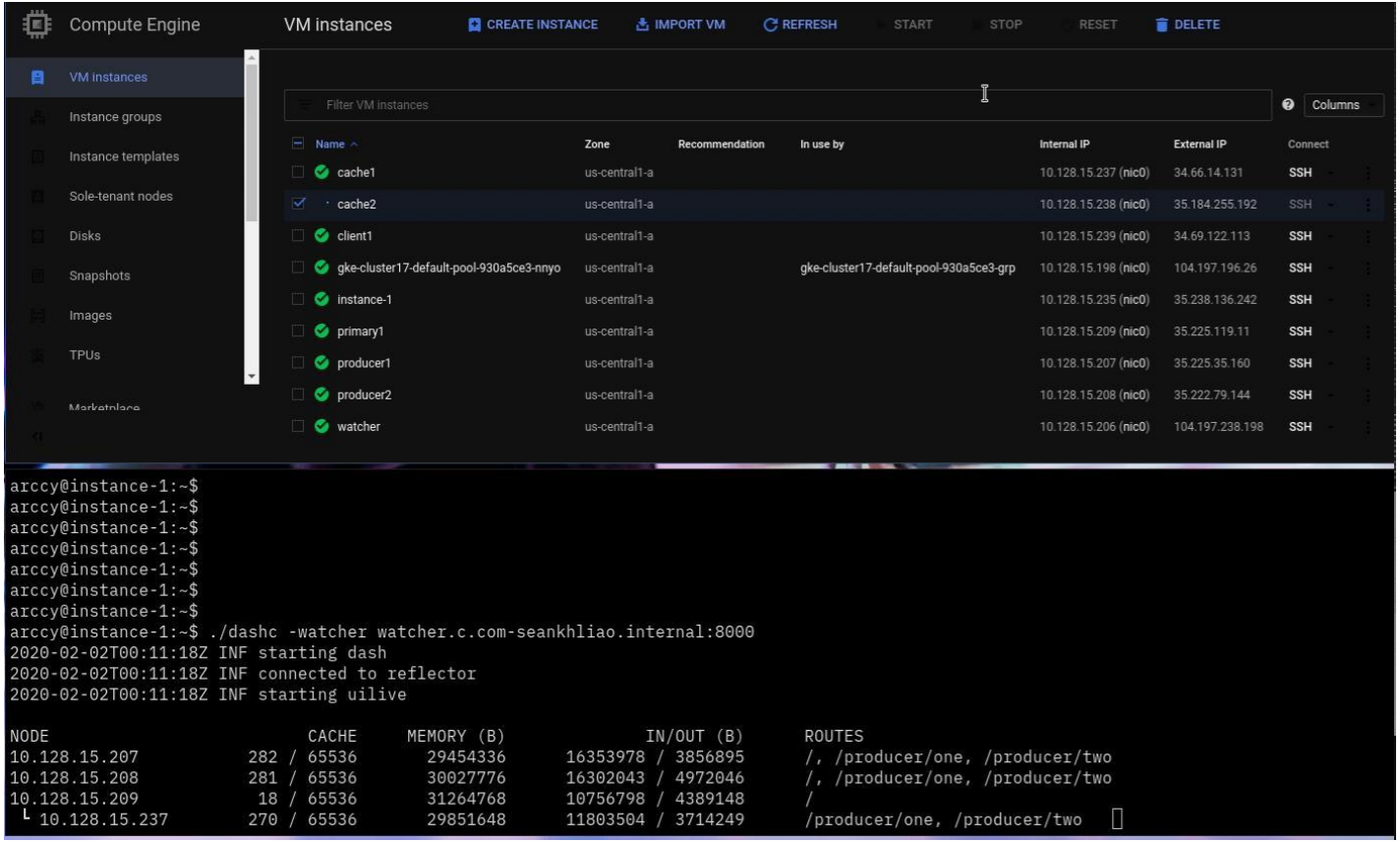
Demo: add load balancer (40s)



Demo: remove load balancer (16s)



Demo: add Cache (44s)



Demo: remove Cache (17s)