Network Fault Detection and Isolation

Richard Sheehan
Production Engineer

1 Network monitoring

2 Networks @Scale

3 NetNorad

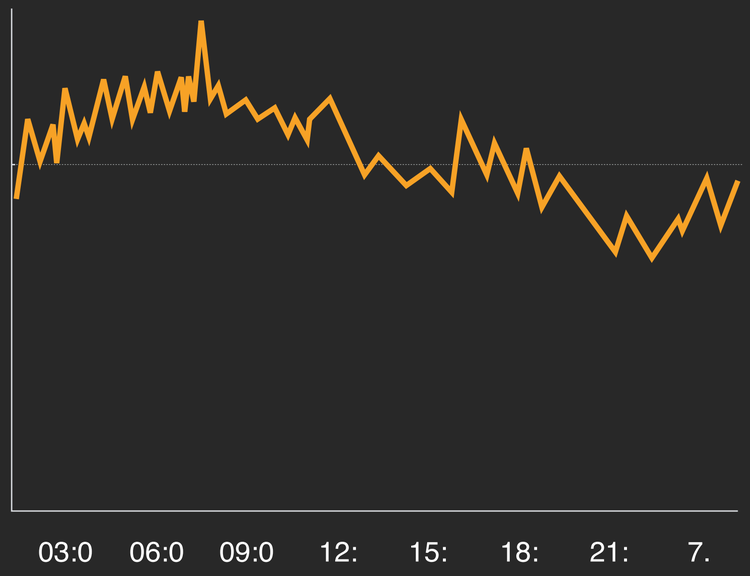
4 NFI


No Network Monitoring
People tell you when the network is broken!

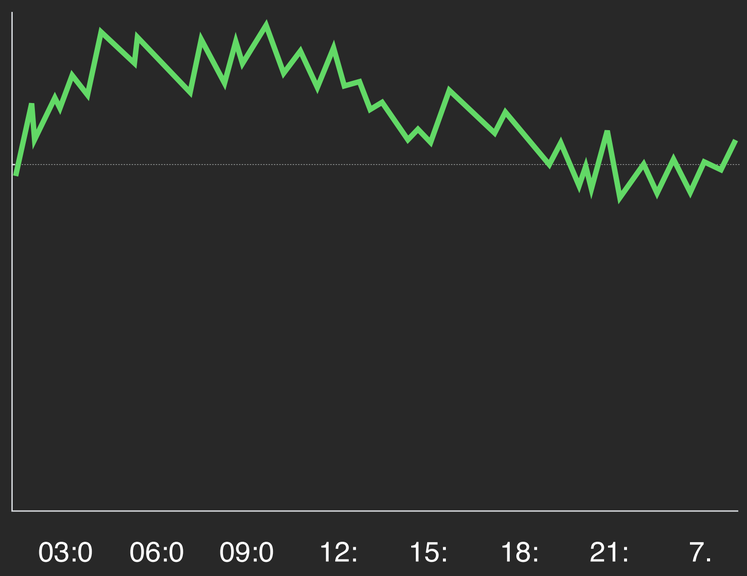

People tell you when they think the network is broken!
YOU investigate EVERYTHING, every time


m_id+'&or
02.0

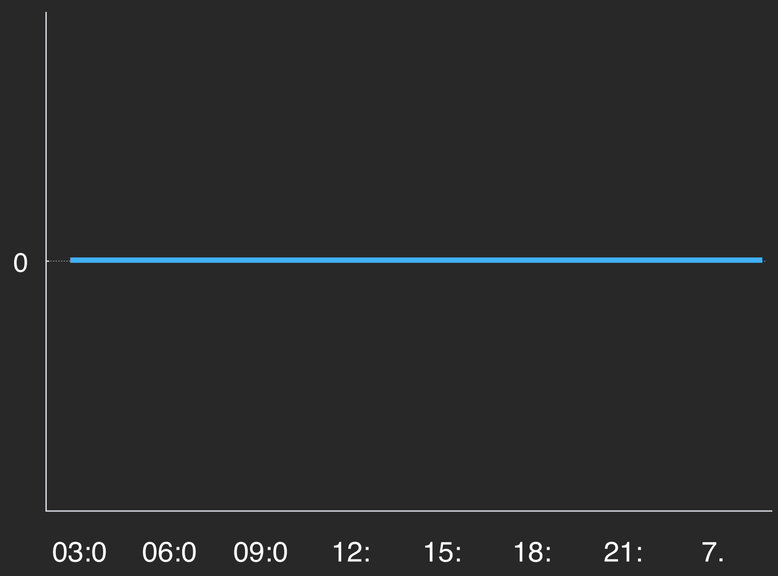
Output Utilization (bps)



Input Utilization (bps)

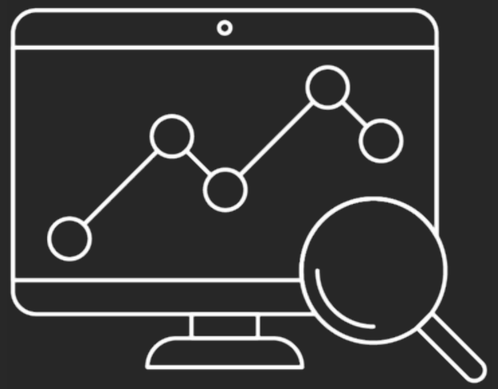


Errors



Detecting packet loss

Standard counters



SNMP

Non-standard counters

```
ssw001# show platform trident counters
Debug counters
            Description
T2Fabric19/0/1 RX - Non congestion discards
T2Fabric19/0/1 TX - IPV4 L3 unicast aged and dropped pkts
T2Fabric19/0/1 RX - Receive policy discard
T2Fabric19/0/1 TX - L2 multicast drop
T2Fabric19/0/1 RX - Tunnel error packets
T2Fabric19/0/1 TX - Invalid VLAN
T2Fabric19/0/1 RX - Receive VLAN drop
T2Fabric19/0/1 RX - Receive multicast drop
T2Fabric19/0/1 TX - Dropped because TTL counter
T2Fabric19/0/1 RX - Receive uRPF drop
T2Fabric19/0/1 TX - Packet dropped due to any condition
T2Fabric19/0/1 RX - IBP discard and CPB full
T2Fabric19/0/1 TX - Miss in VXLT table counter
```


RECOVERY: 78%


RECOVERY: YES


UPGRADE
INTERNET
Social Network
Social Media
Security System

Fundamental
Fatal Flaw

If you didn’t design hardware and software
You really shouldn’t trust it to tell you it’s ok.

1 Network monitoring

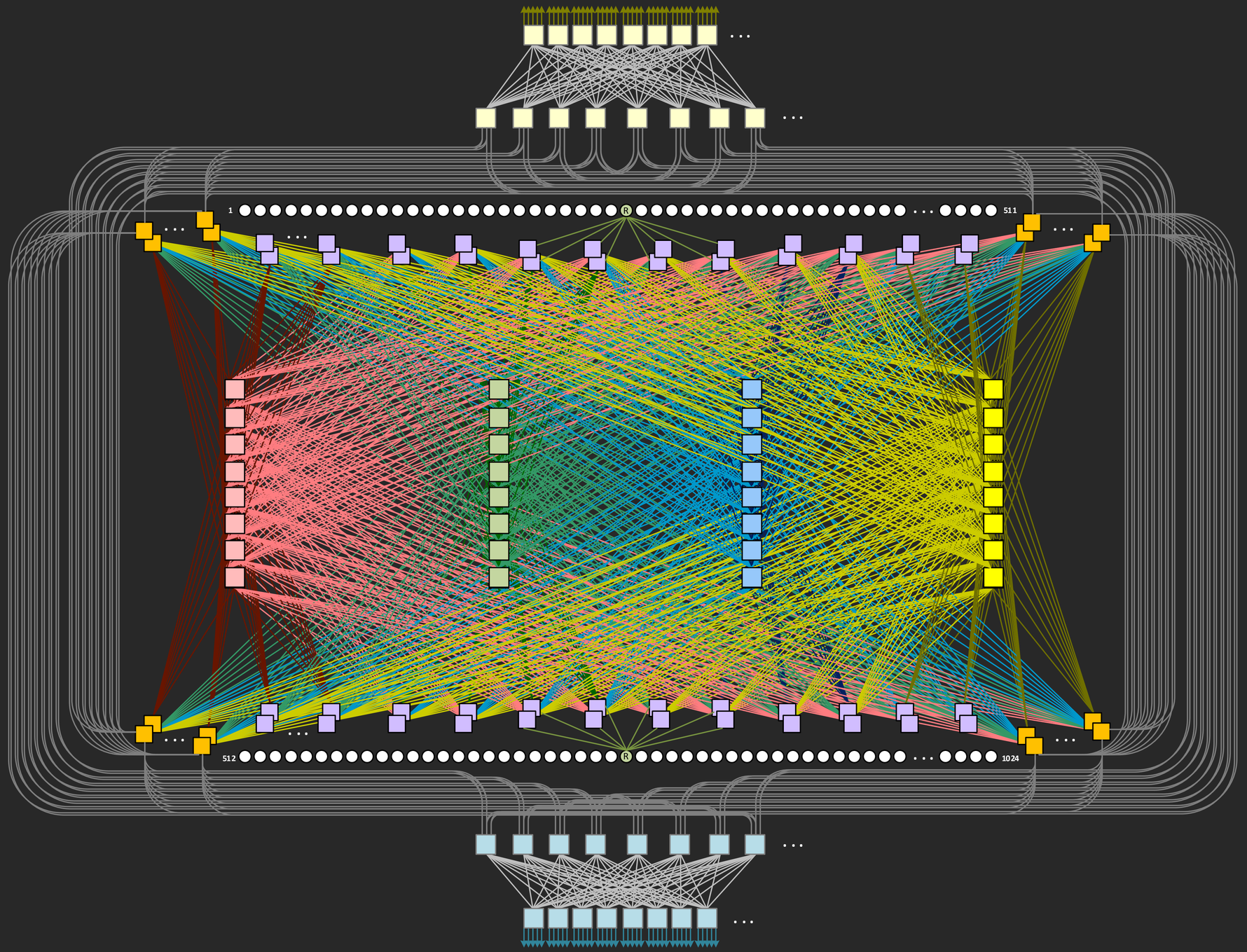
2 Networks @Scale

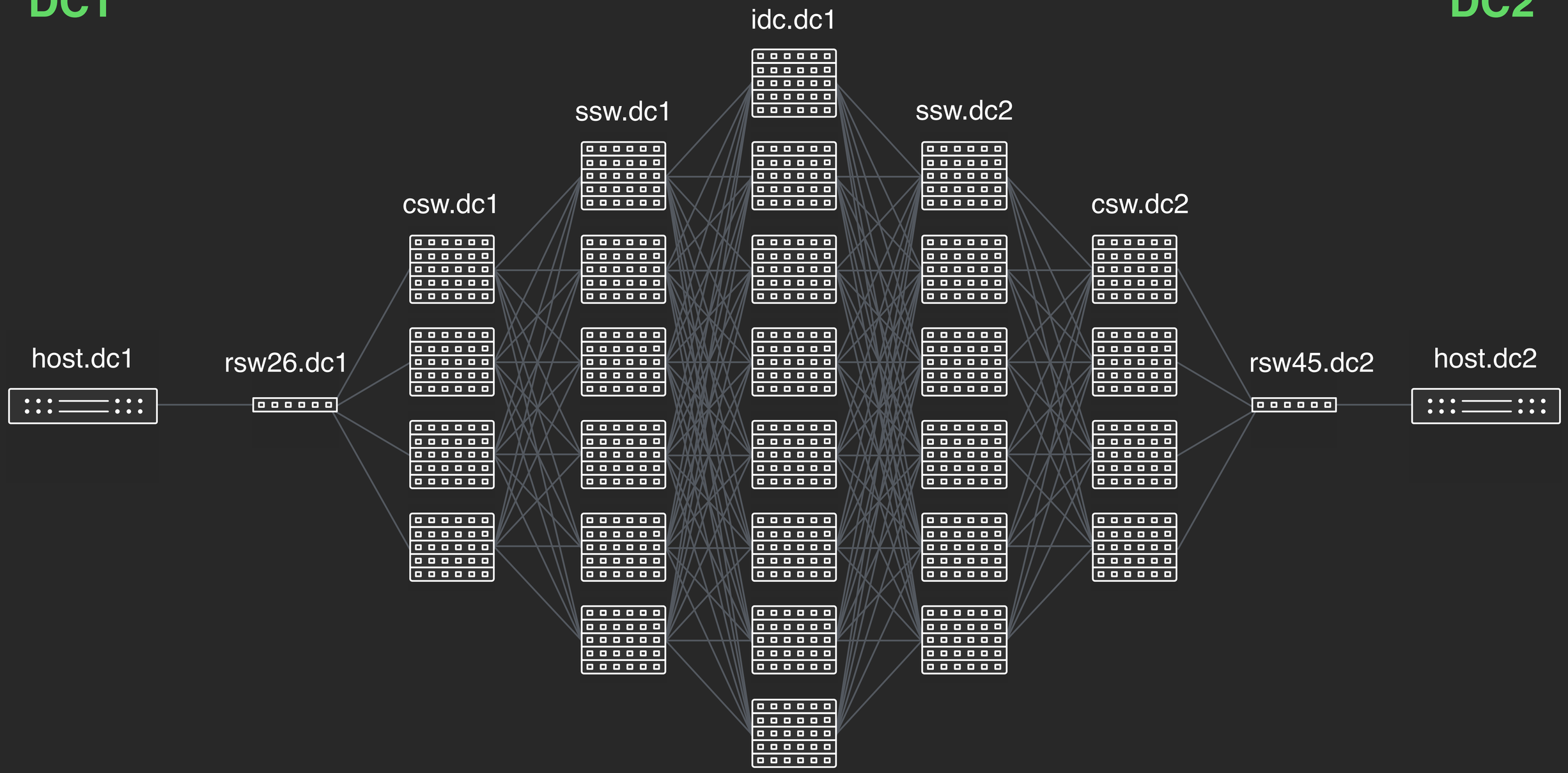
3 NetNorad

4 NFI

Data Center Networks

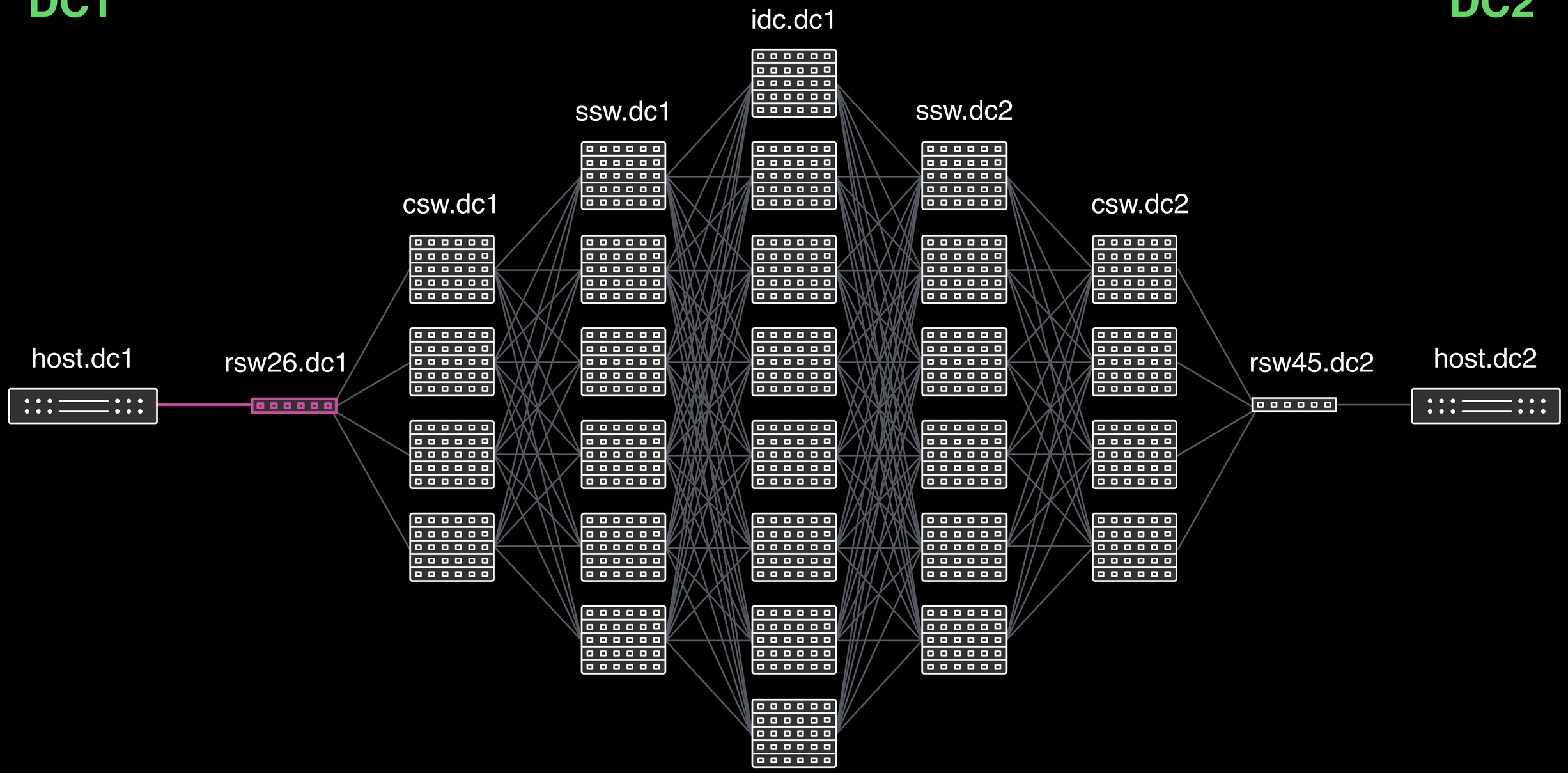
CLOS Fabric


1
511
512
1024
R
R

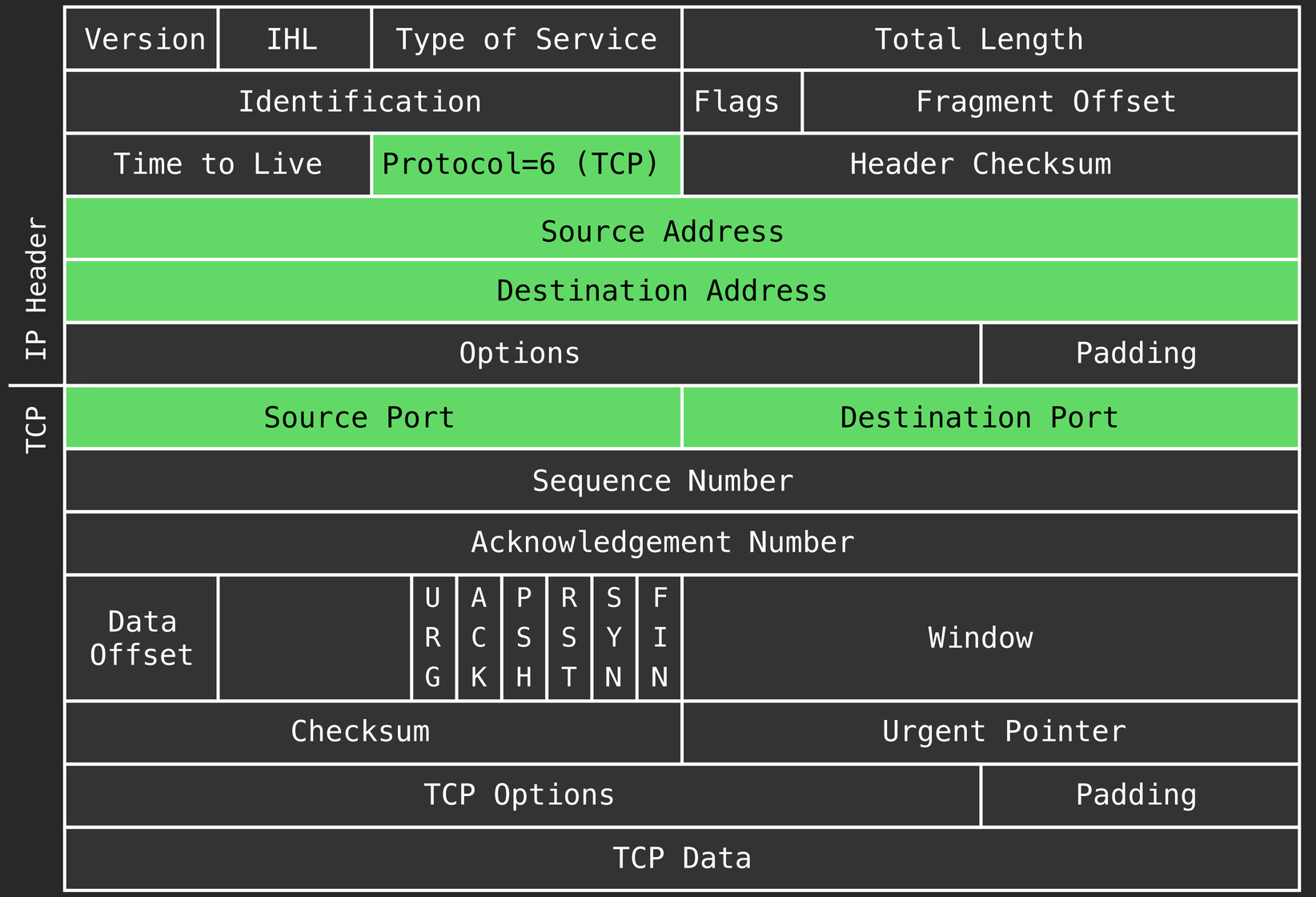

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

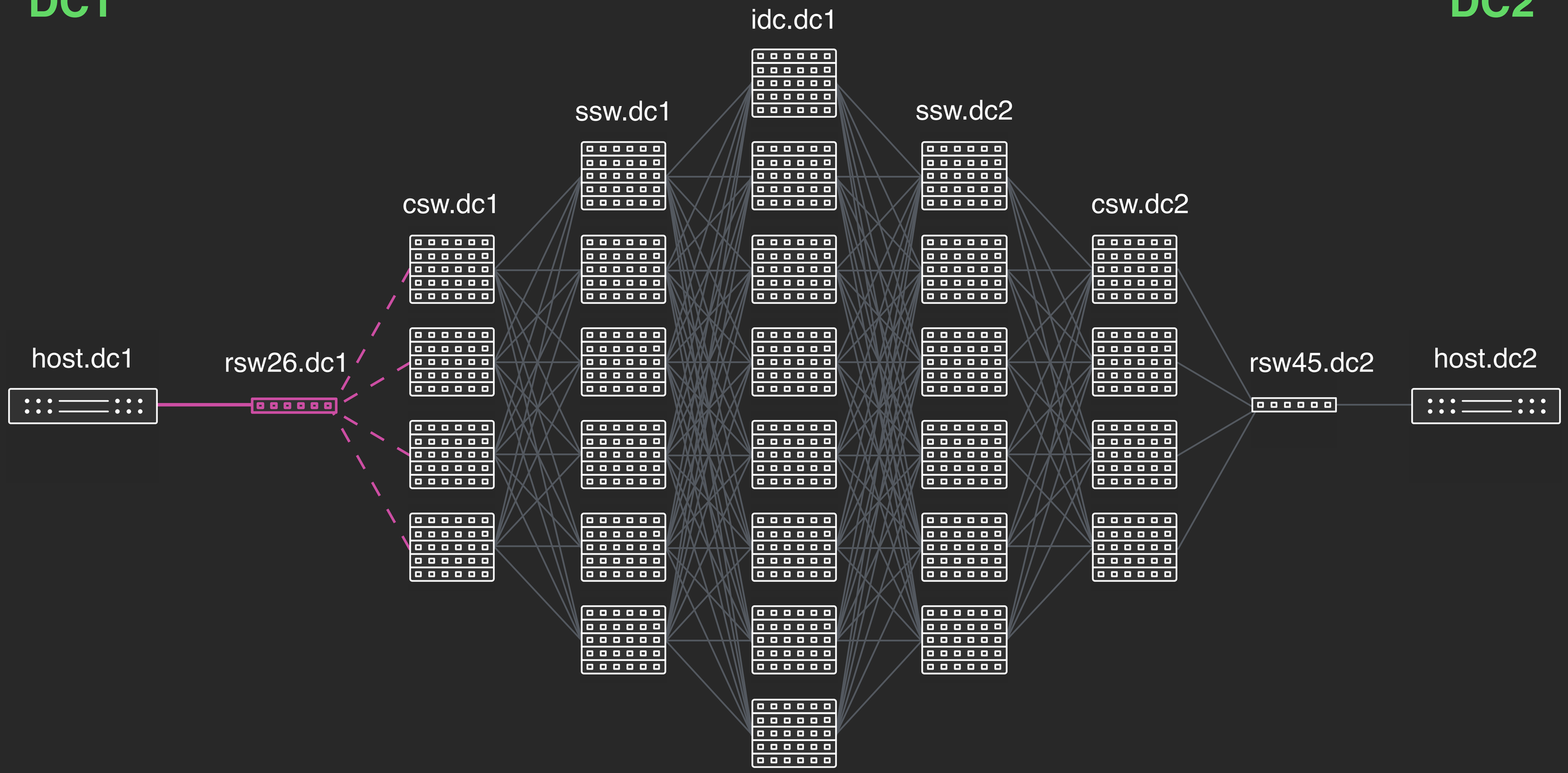
ECMP

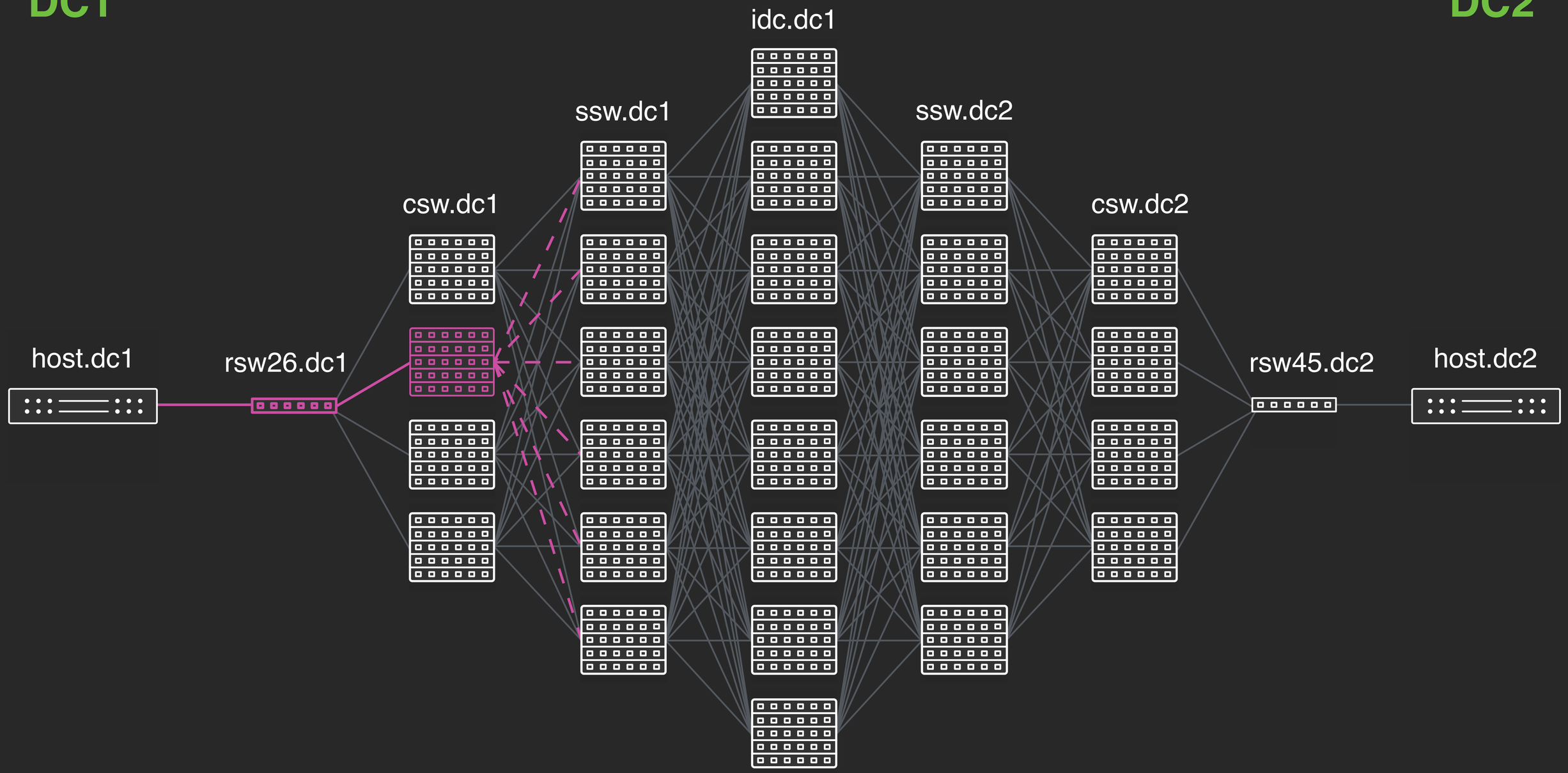
Equal-Cost Multi-Path

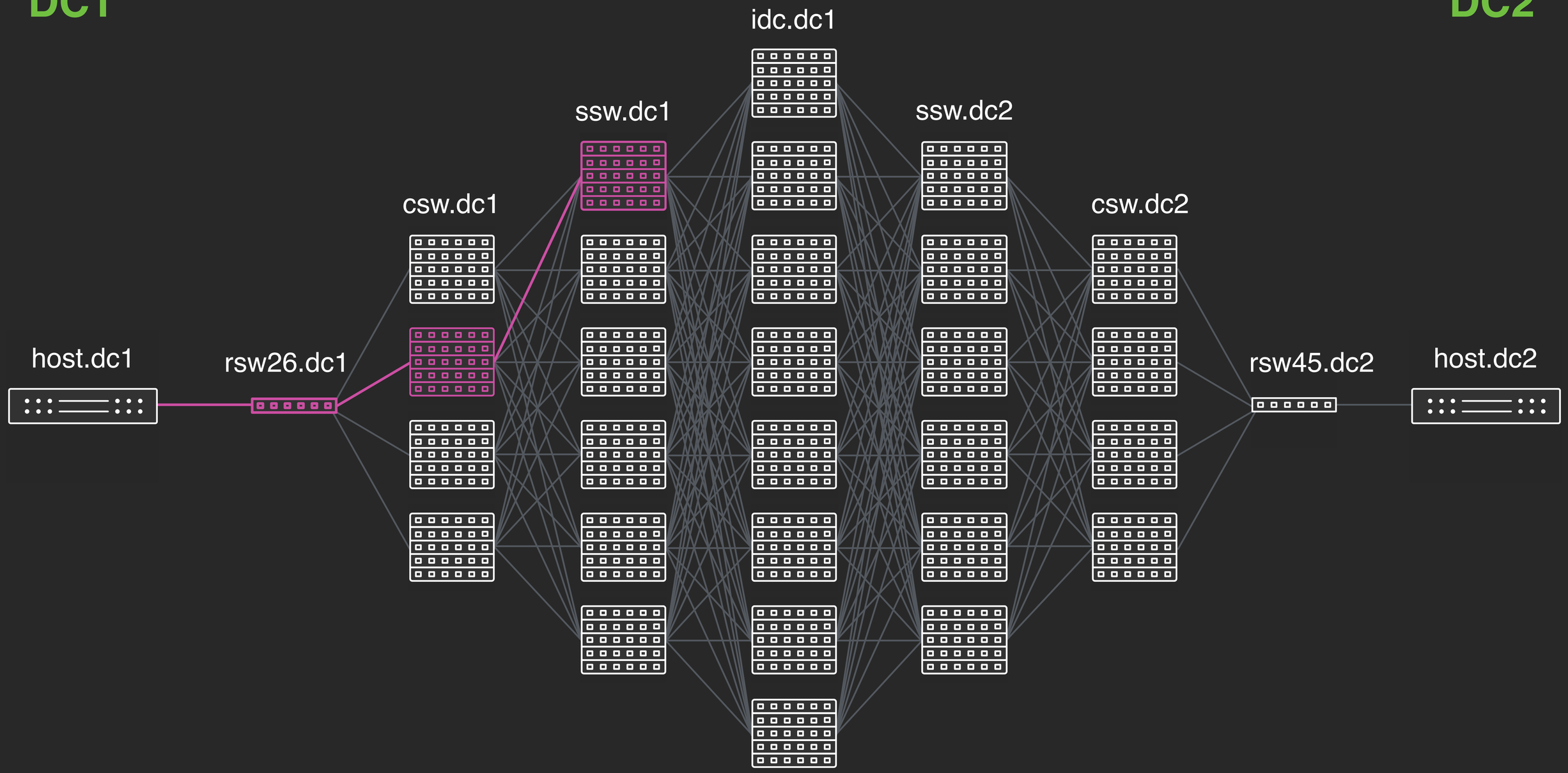

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

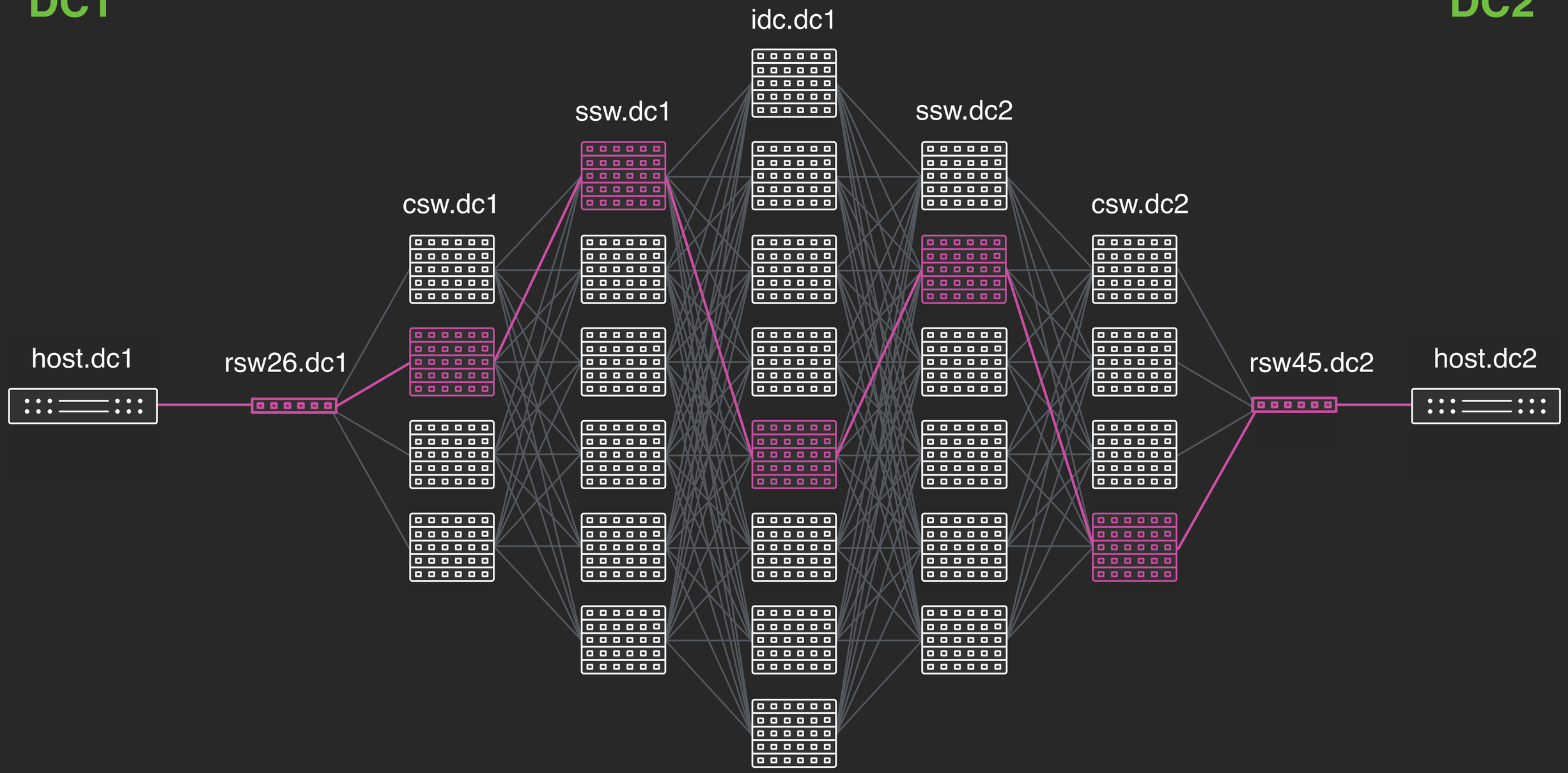
TCP/IP Packet

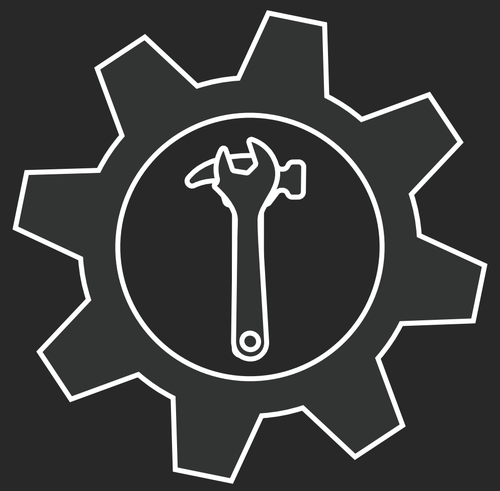



DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2


DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

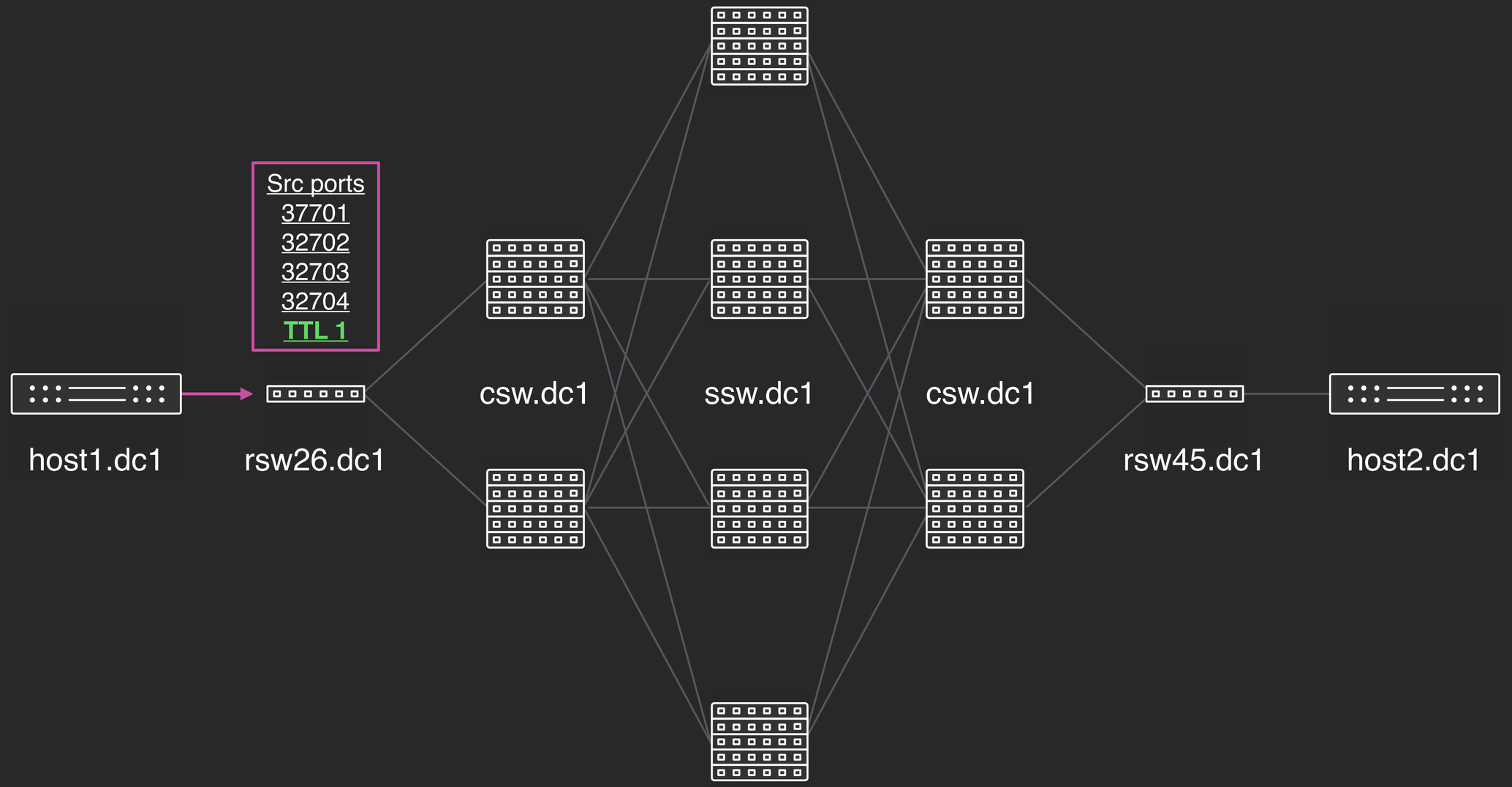

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

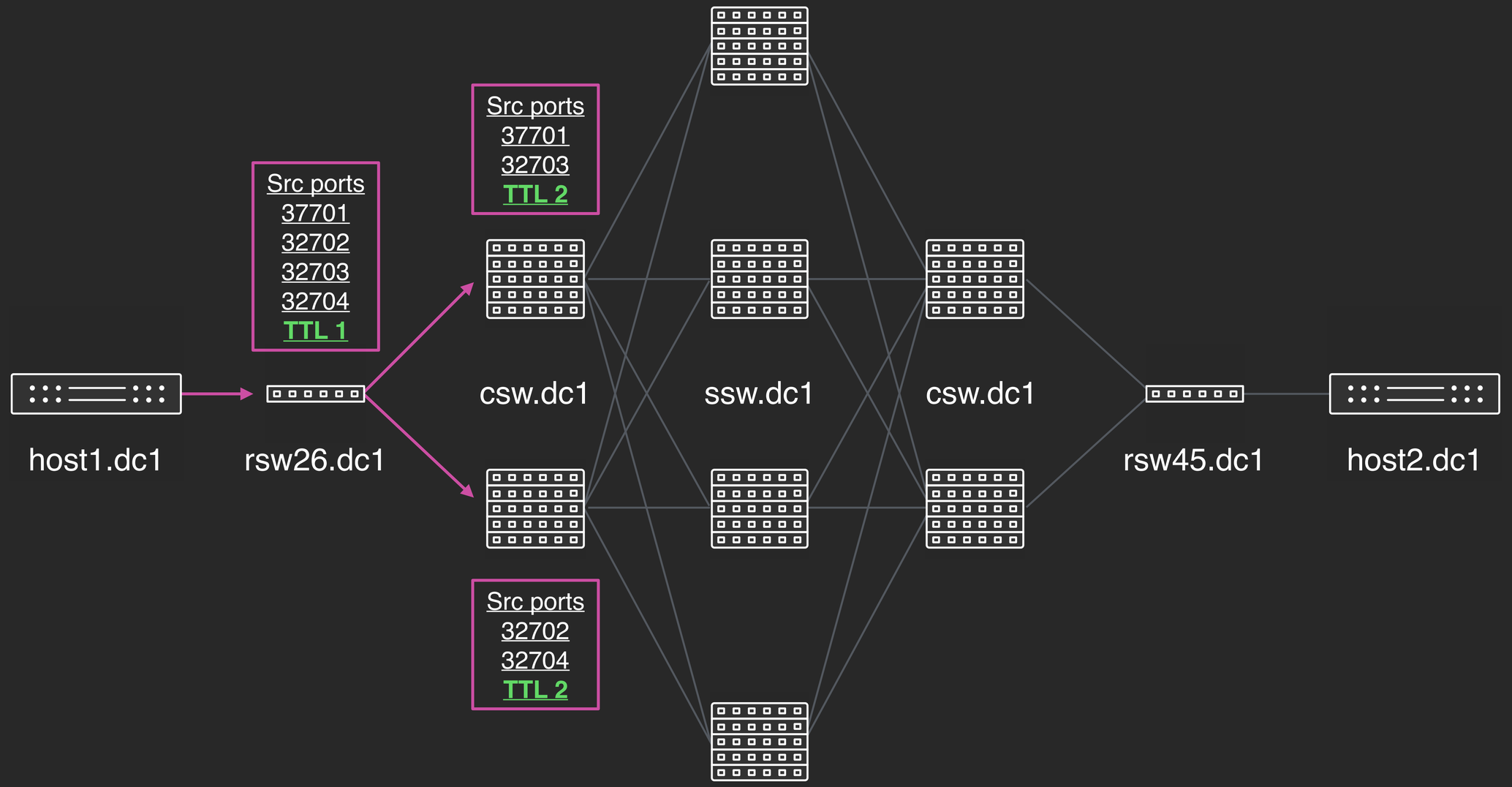

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

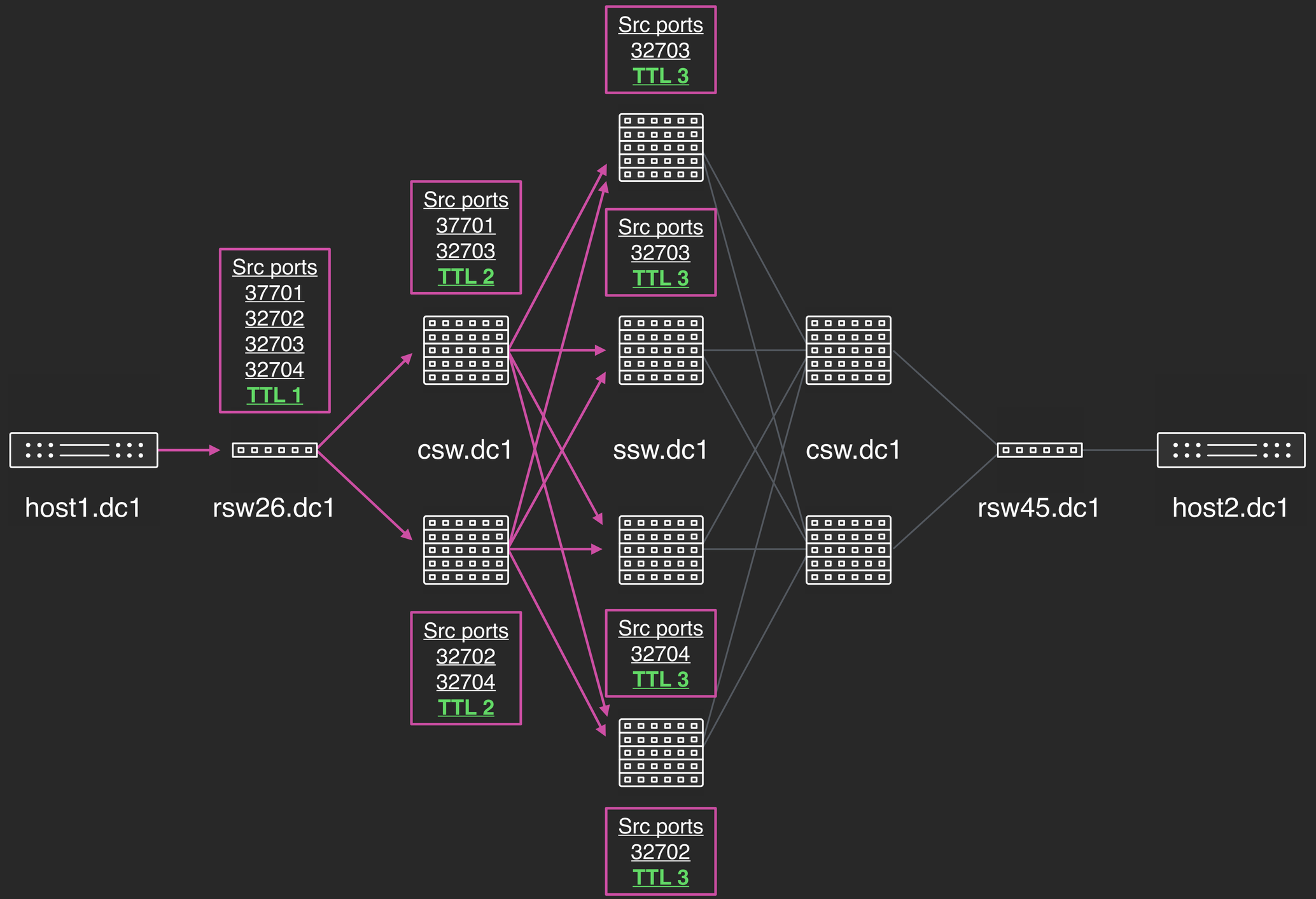


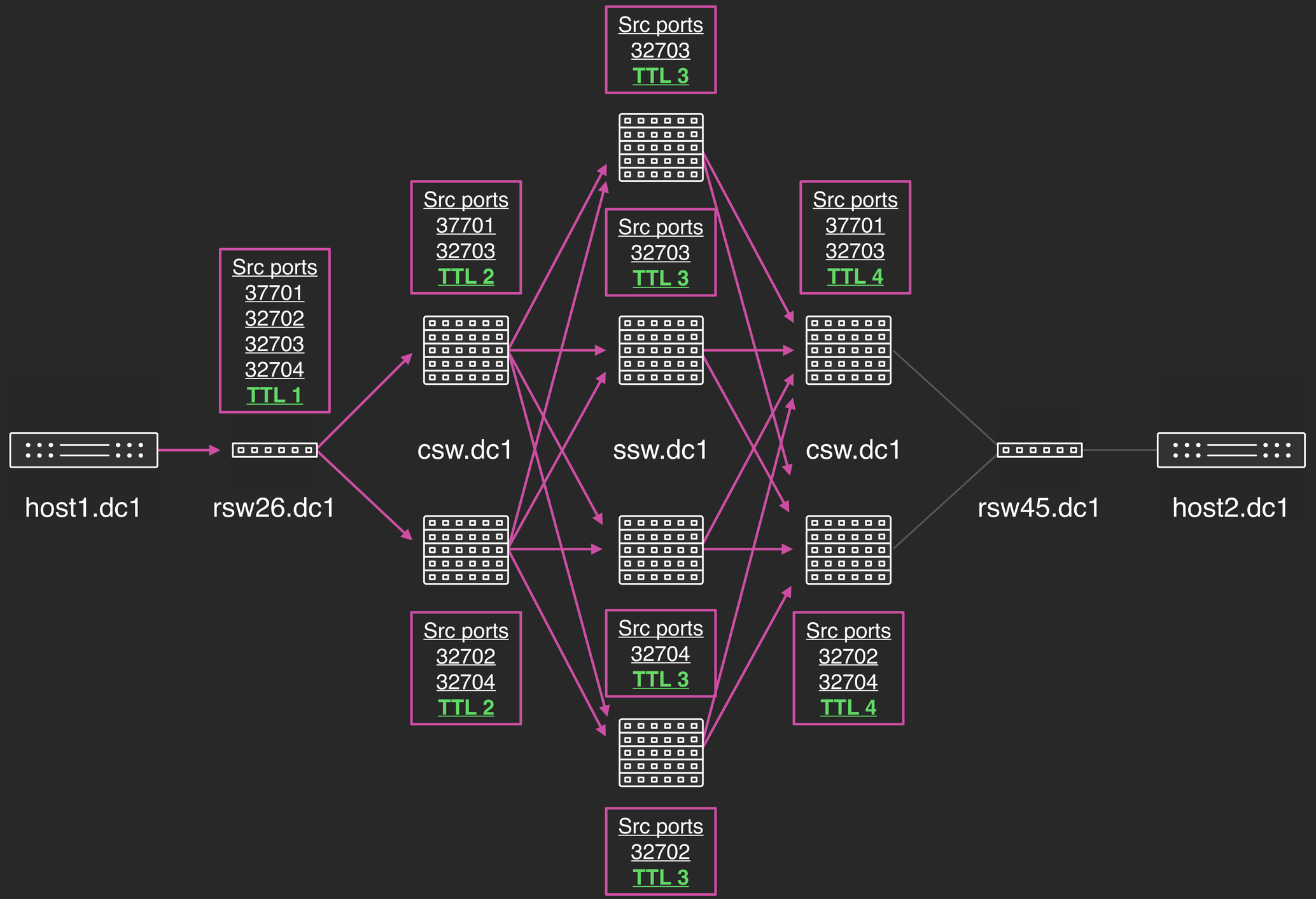
fbtracert

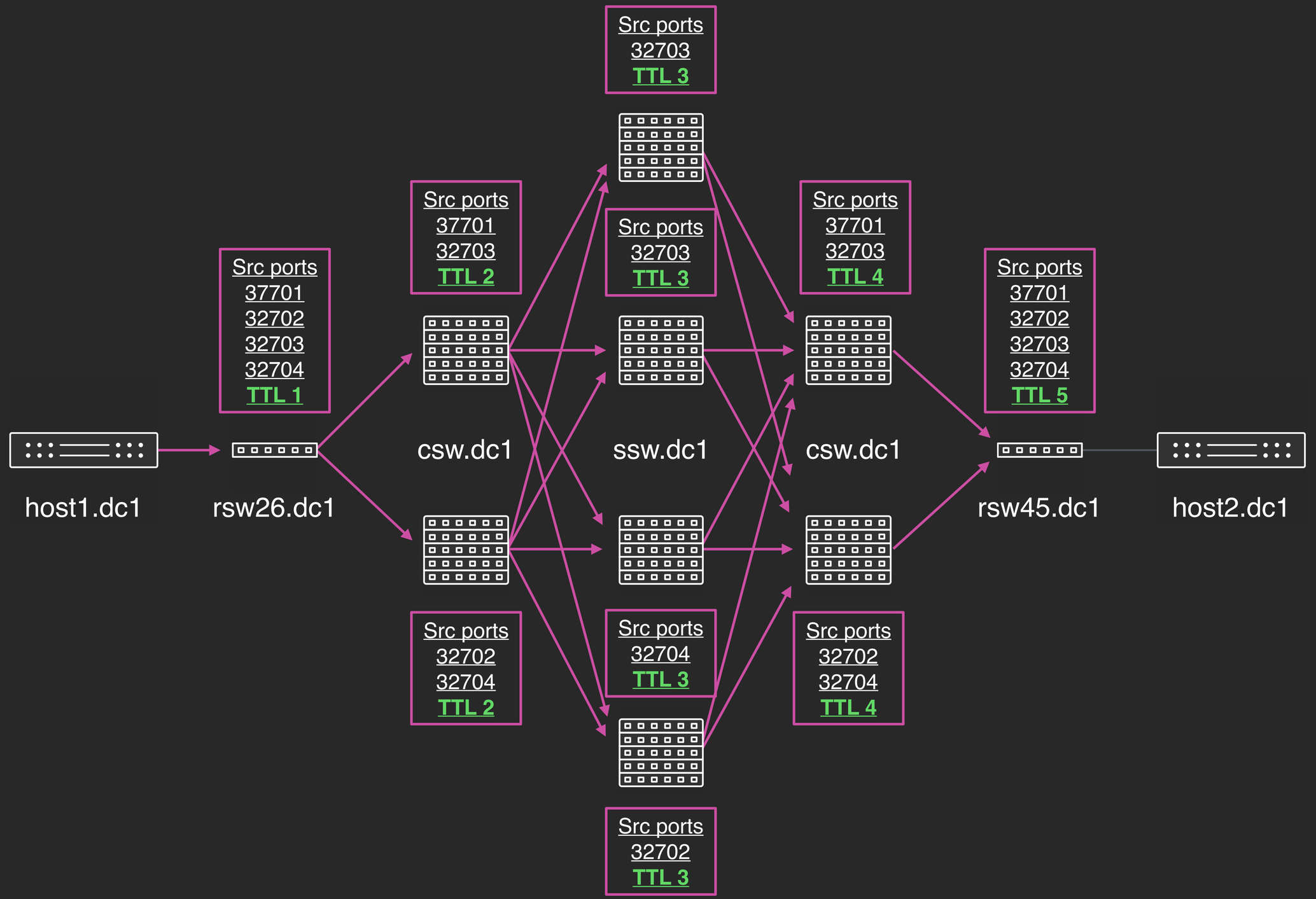
github.com/facebook/fbtracert

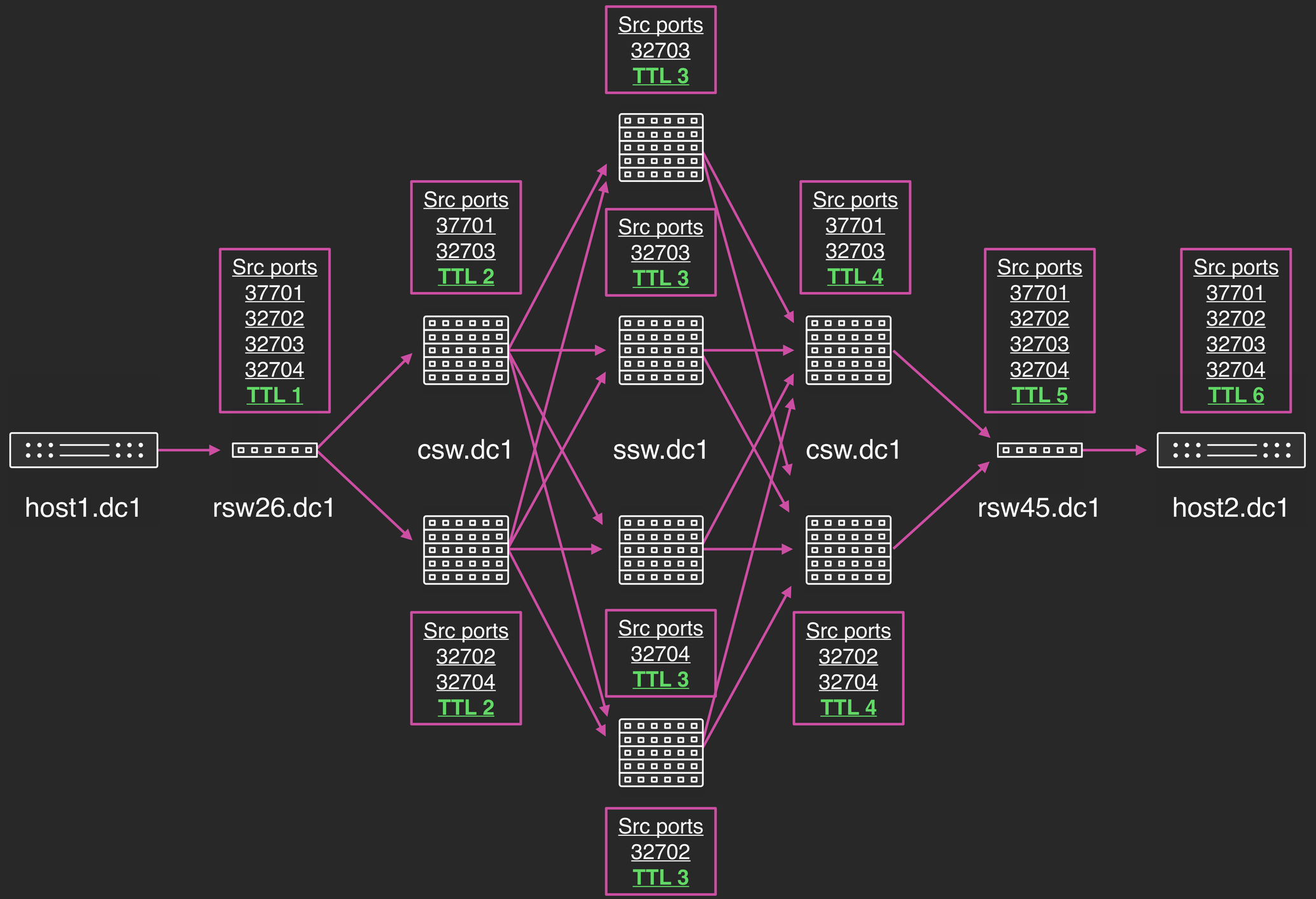

Src ports
37701
32702
32703
32704
TTL 1
host1.dc1
rsw26.dc1
csw.dc1
ssw.dc1
csw.dc1
rsw45.dc1
host2.dc1


Src ports
37701
32702
32703
32704
TTL 1
Src ports
37701
32703
TTL 2
Src ports
32702
32704
TTL 2
host1.dc1
rsw26.dc1
csw.dc1
ssw.dc1
csw.dc1
rsw45.dc1
host2.dc1


Src ports
32703
TTL 3
Src ports
37701
32703
TTL 2
Src ports
32703
TTL 3
Src ports
37701
32702
32703
32704
TTL 1
csw.dc1
ssw.dc1
csw.dc1
host1.dc1
rsw26.dc1
rsw45.dc1
host2.dc1
Src ports
32702
32704
TTL 2
Src ports
32704
TTL 3
Src ports
32702
TTL 3


Src ports
32703
TTL 3
Src ports
37701
32703
TTL 2
Src ports
32703
TTL 3
Src ports
37701
32703
TTL 4
Src ports
37701
32702
32703
32704
TTL 1
csw.dc1
ssw.dc1
csw.dc1
host1.dc1
rsw26.dc1
rsw45.dc1
host2.dc1
Src ports
32702
32704
TTL 2
Src ports
32704
TTL 3
Src ports
32702
32704
TTL 4
Src ports
32702
TTL 3


Src ports
32703
TTL 3
Src ports
37701
32703
TTL 2
Src ports
37701
32703
TTL 4
Src ports
37701
32702
32703
32704
TTL 1
Src ports
37701
32702
32703
32704
TTL 5
Src ports
32703
TTL 3
csw.dc1
ssw.dc1
csw.dc1
host1.dc1
rsw26.dc1
rsw45.dc1
host2.dc1
Src ports
32702
32704
TTL 2
Src ports
32704
TTL 3
Src ports
32702
32704
TTL 4
Src ports
32702
TTL 3


Src ports
32703
TTL 3
Src ports
37701
32703
TTL 2
Src ports
37701
32703
TTL 4
Src ports
37701
32702
32703
32704
TTL 1
Src ports
37701
32702
32703
32704
TTL 5
Src ports
37701
32702
32703
32704
TTL 6
Src ports
32703
TTL 3
csw.dc1
ssw.dc1
csw.dc1
host1.dc1
rsw26.dc1
rsw45.dc1
host2.dc1
Src ports
32702
32704
TTL 2
Src ports
32704
TTL 3
Src ports
32702
32704
TTL 4
Src ports
32702
TTL 3

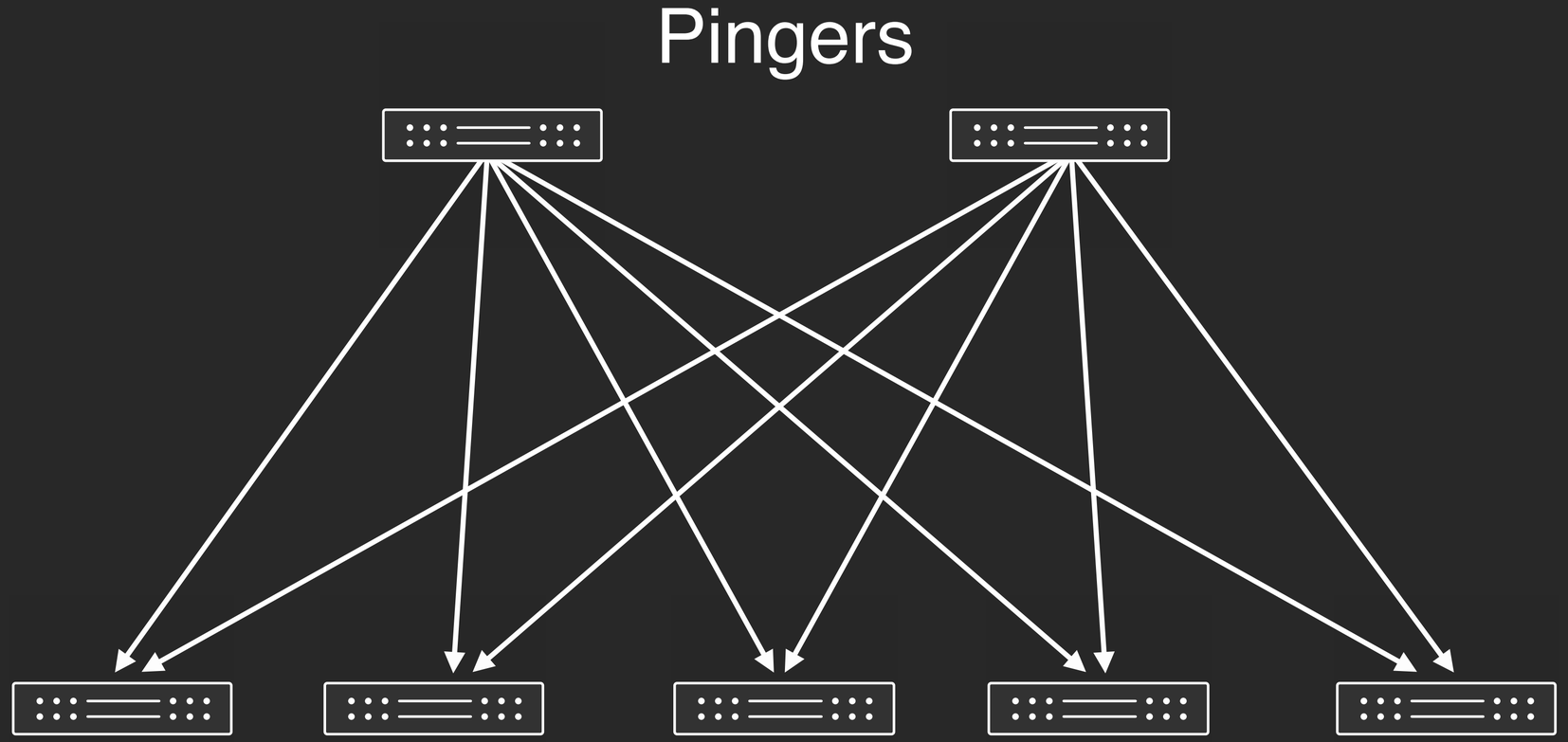
1 Network monitoring

2 Networks @Scale

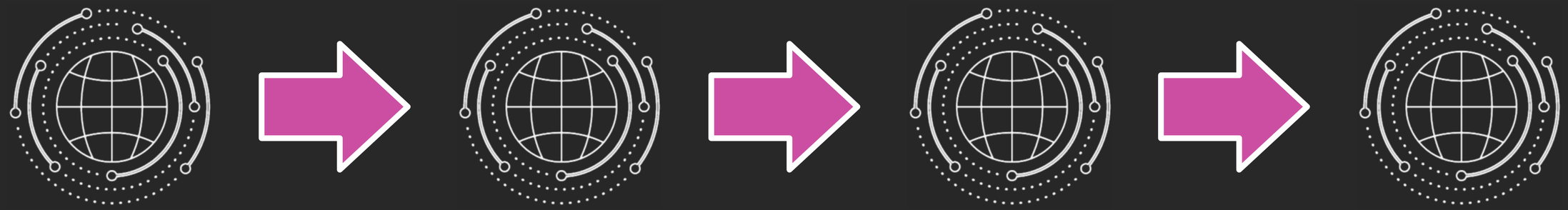
3 NetNorad

4 NFI

Massive pinging

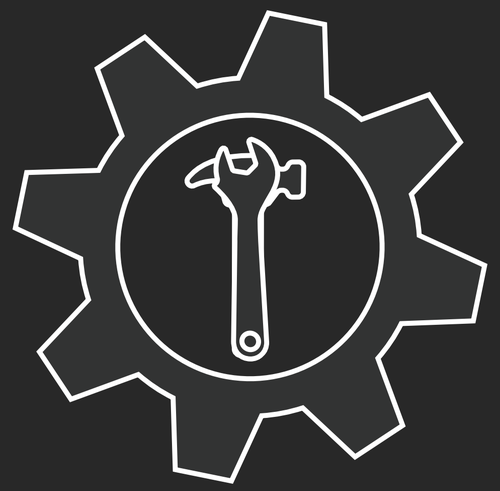


NetNORAD evolution



PING TCP ICMP UDP

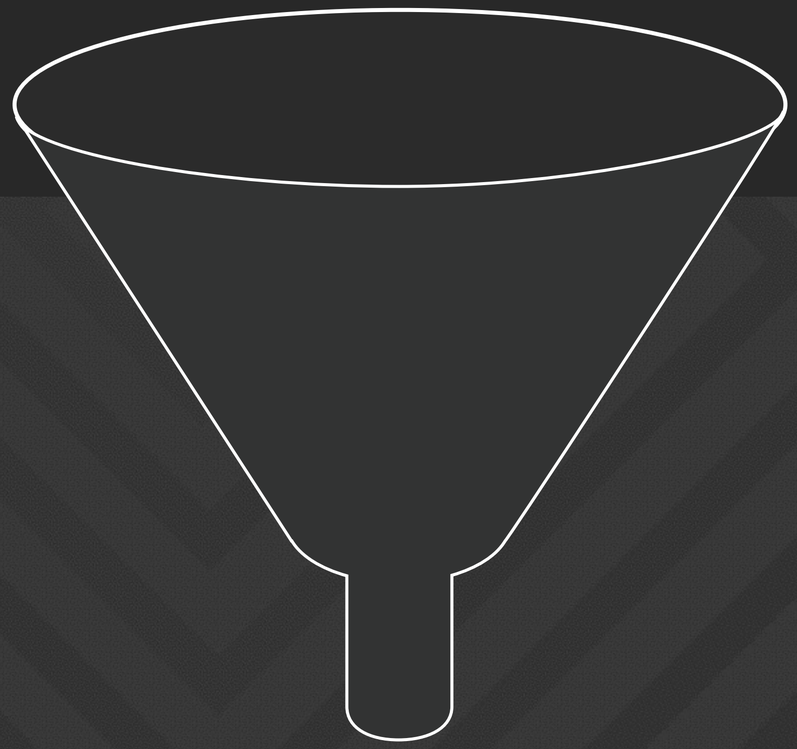
UDP Probe format

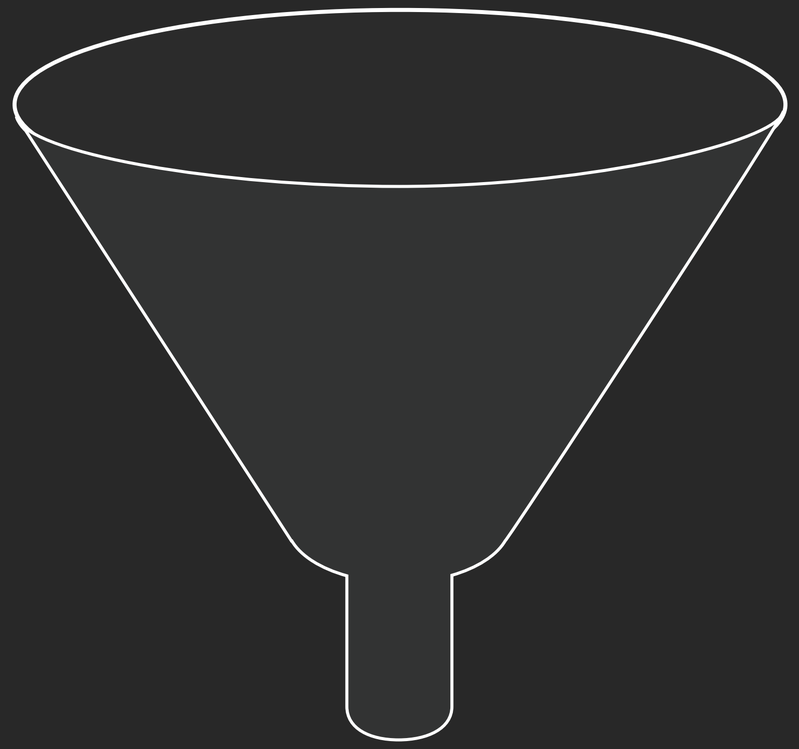


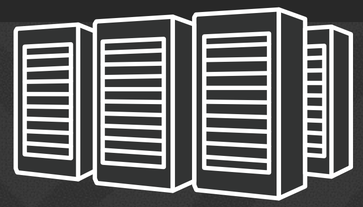
UdpPinger

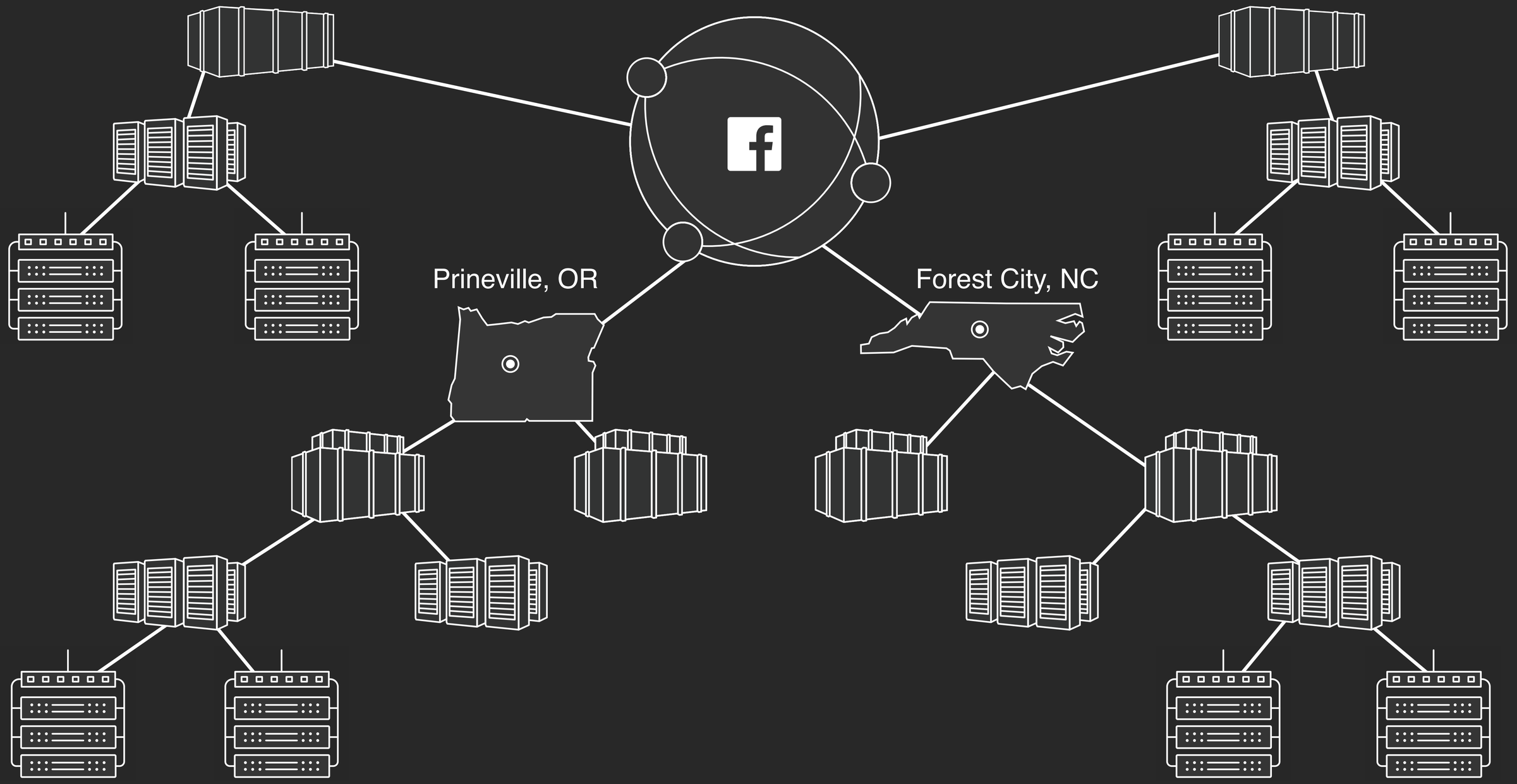
github.com/facebook/UdpPinger

How to ping and process data?

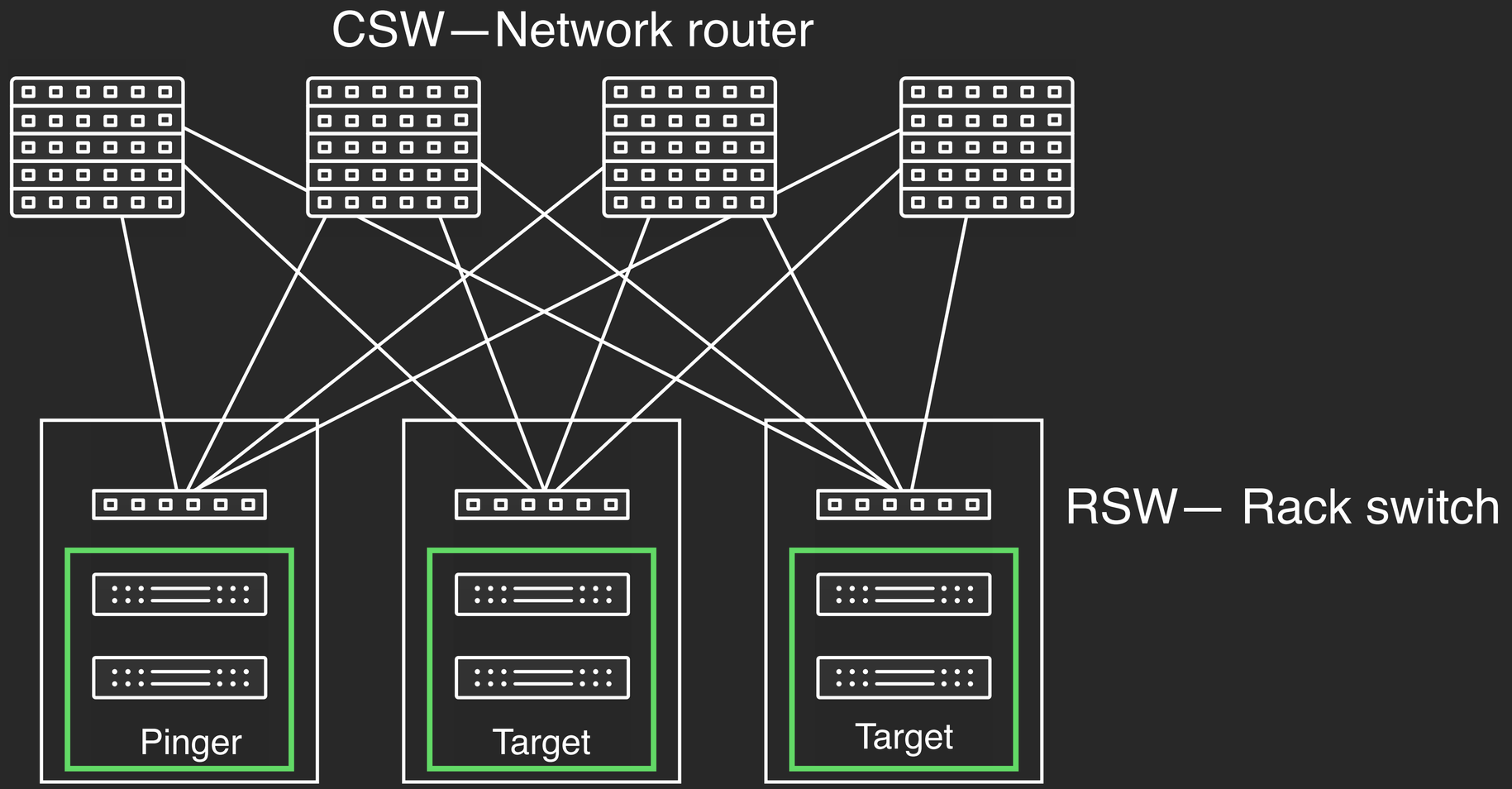


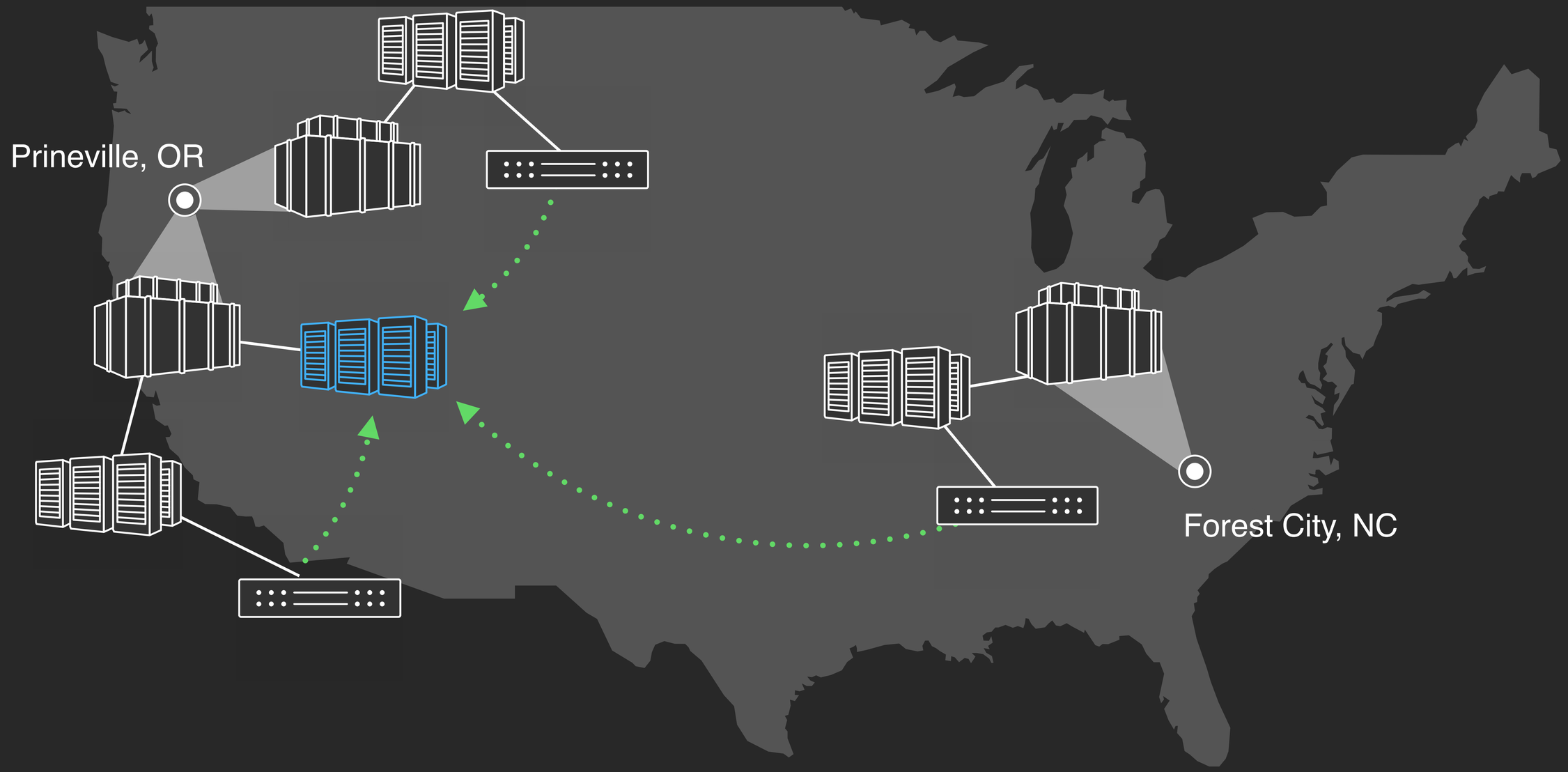


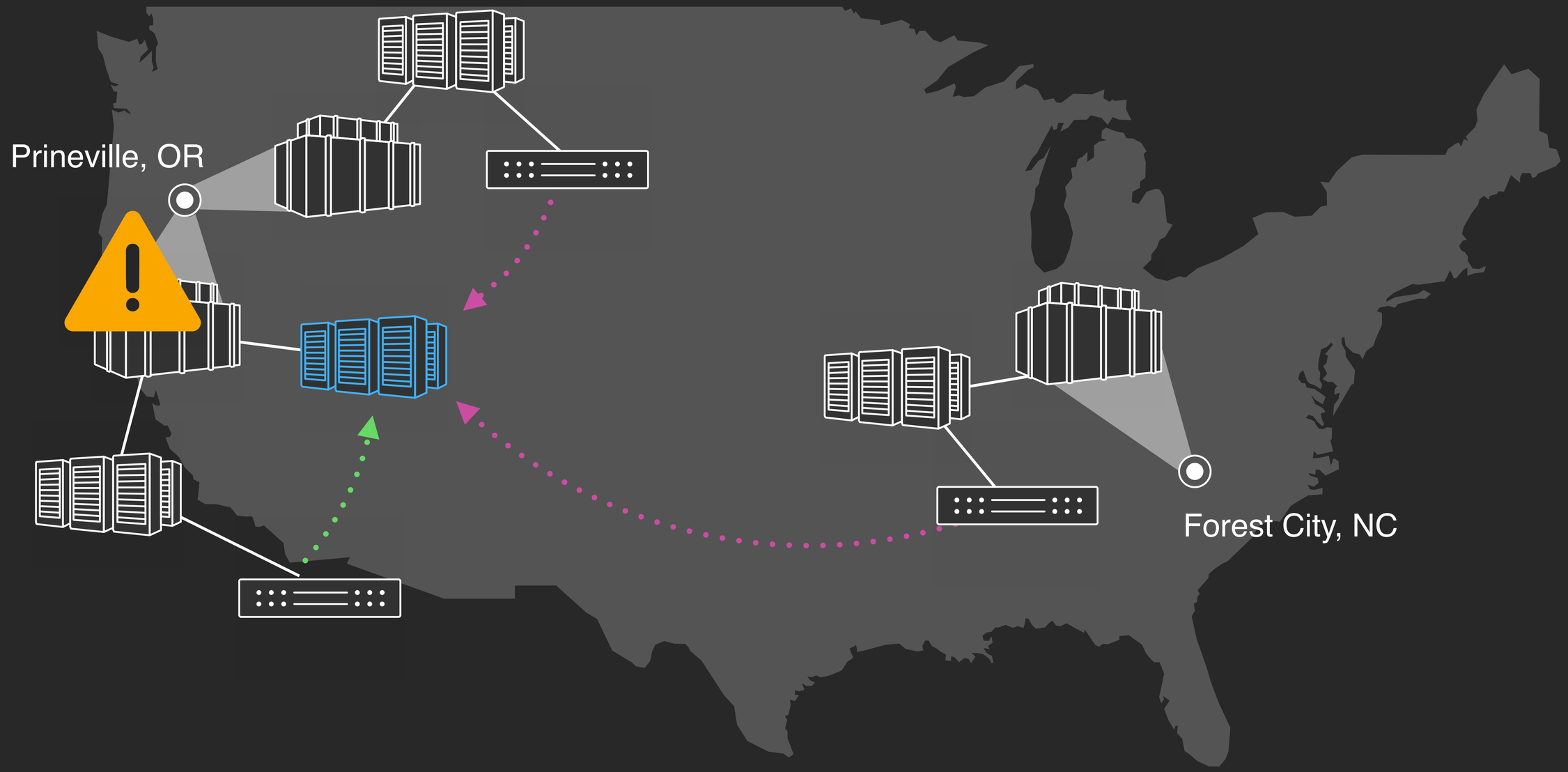



Prineville, OR
Forest City, NC

Pinging in a cluster




Prineville, OR
Forest City, NC


Prineville, OR
Forest City, NC

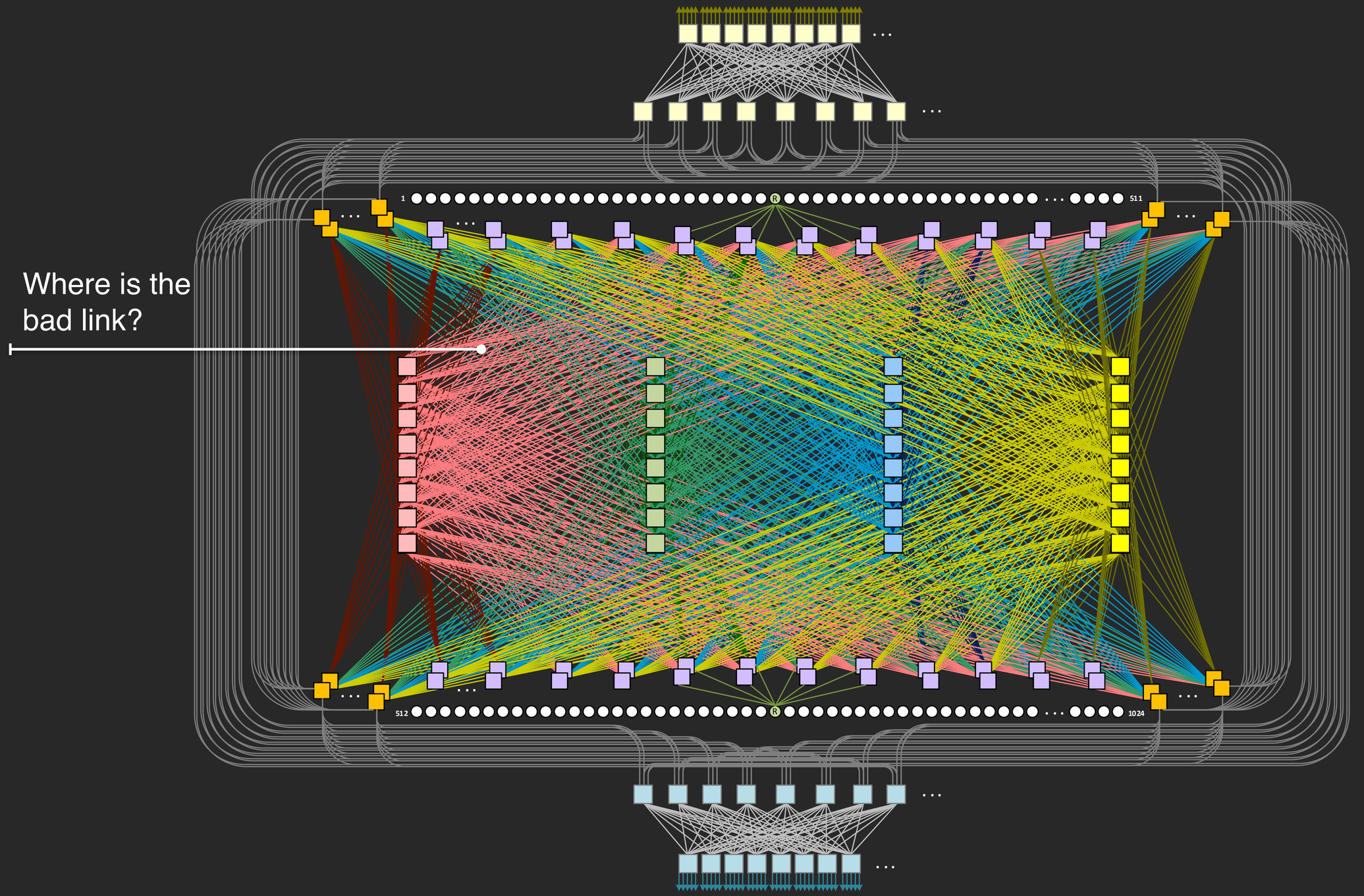
1 Network monitoring

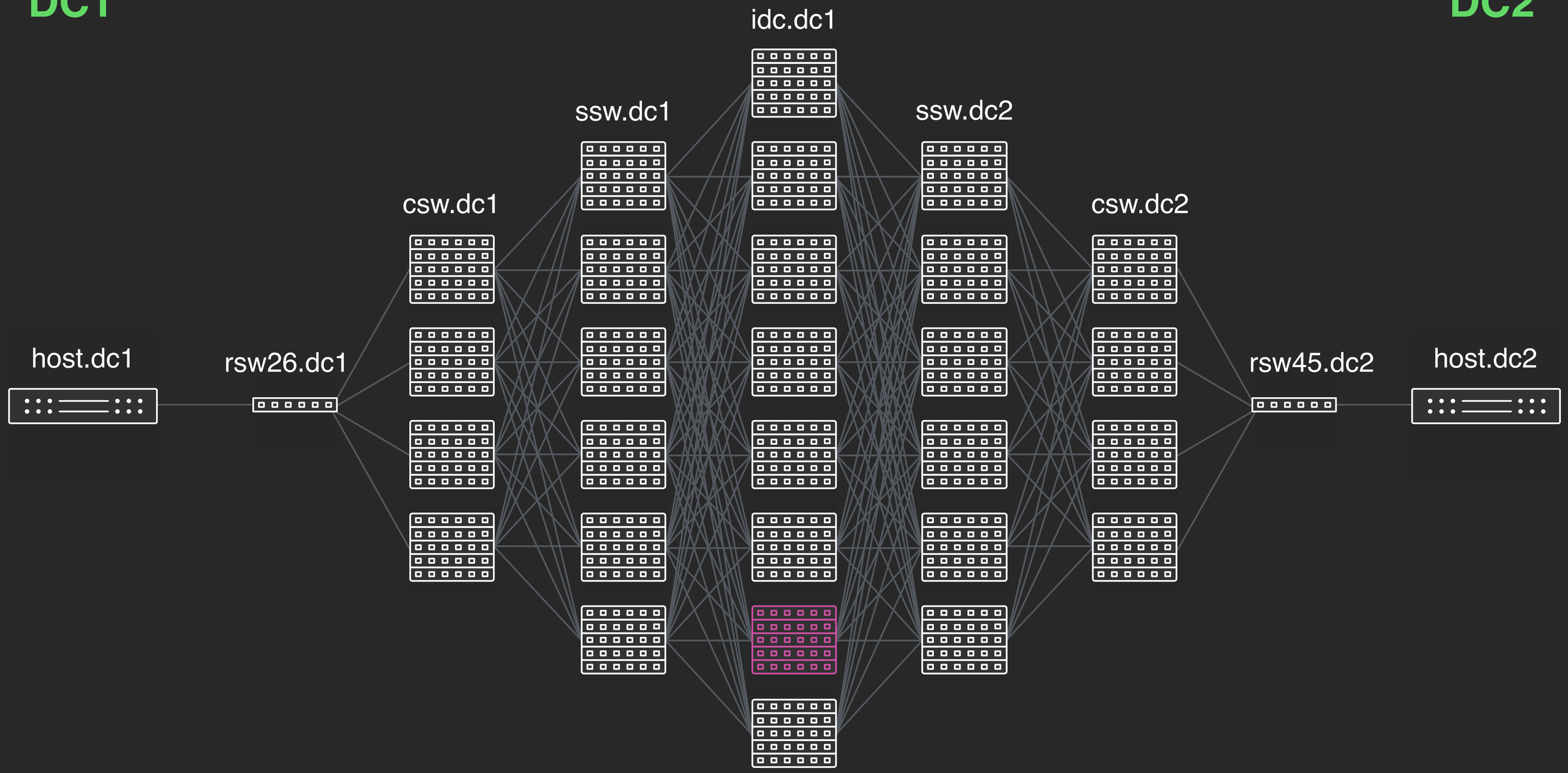
2 Network @Scale

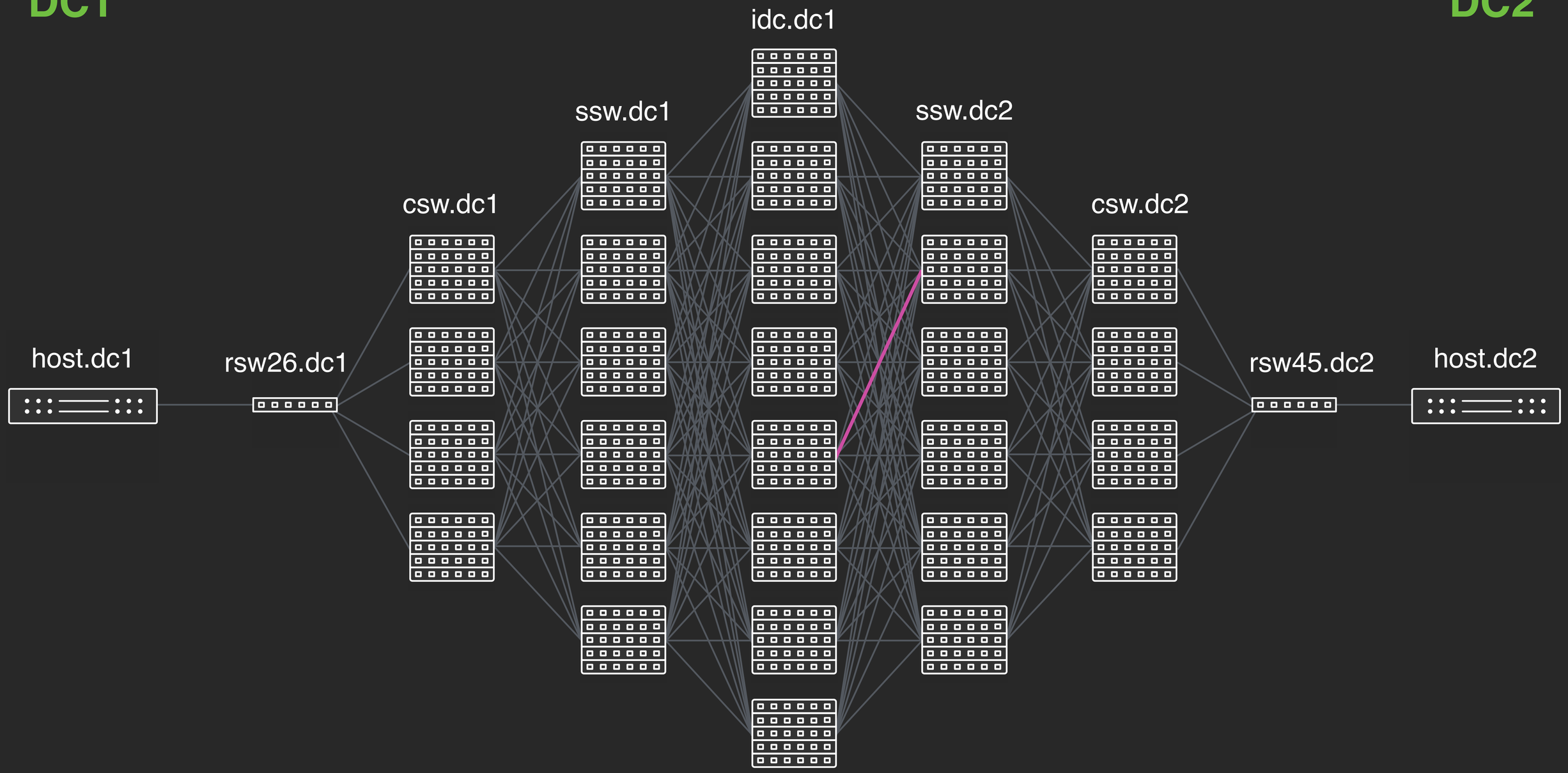
3 NetNorad

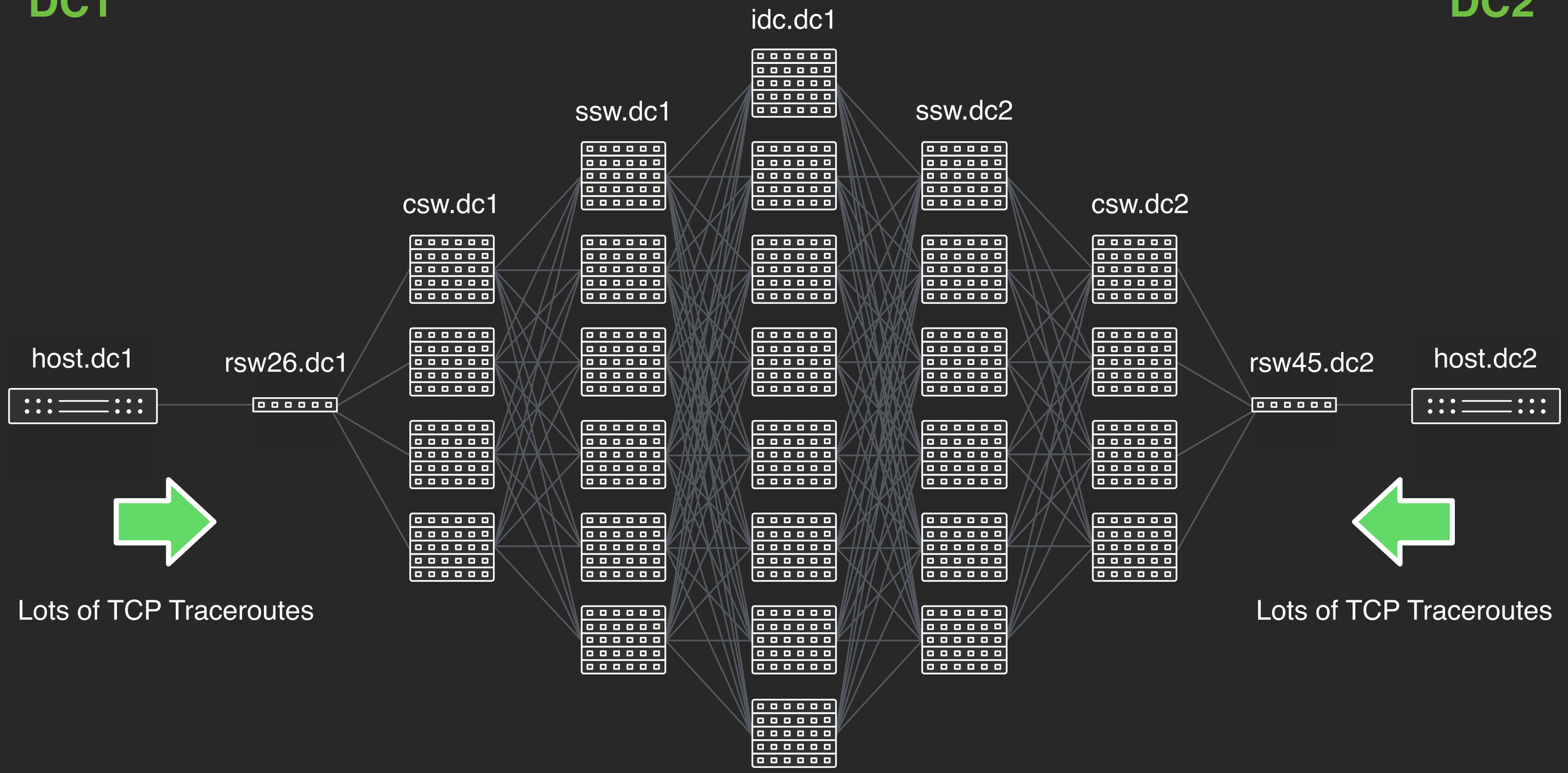
4 NFI

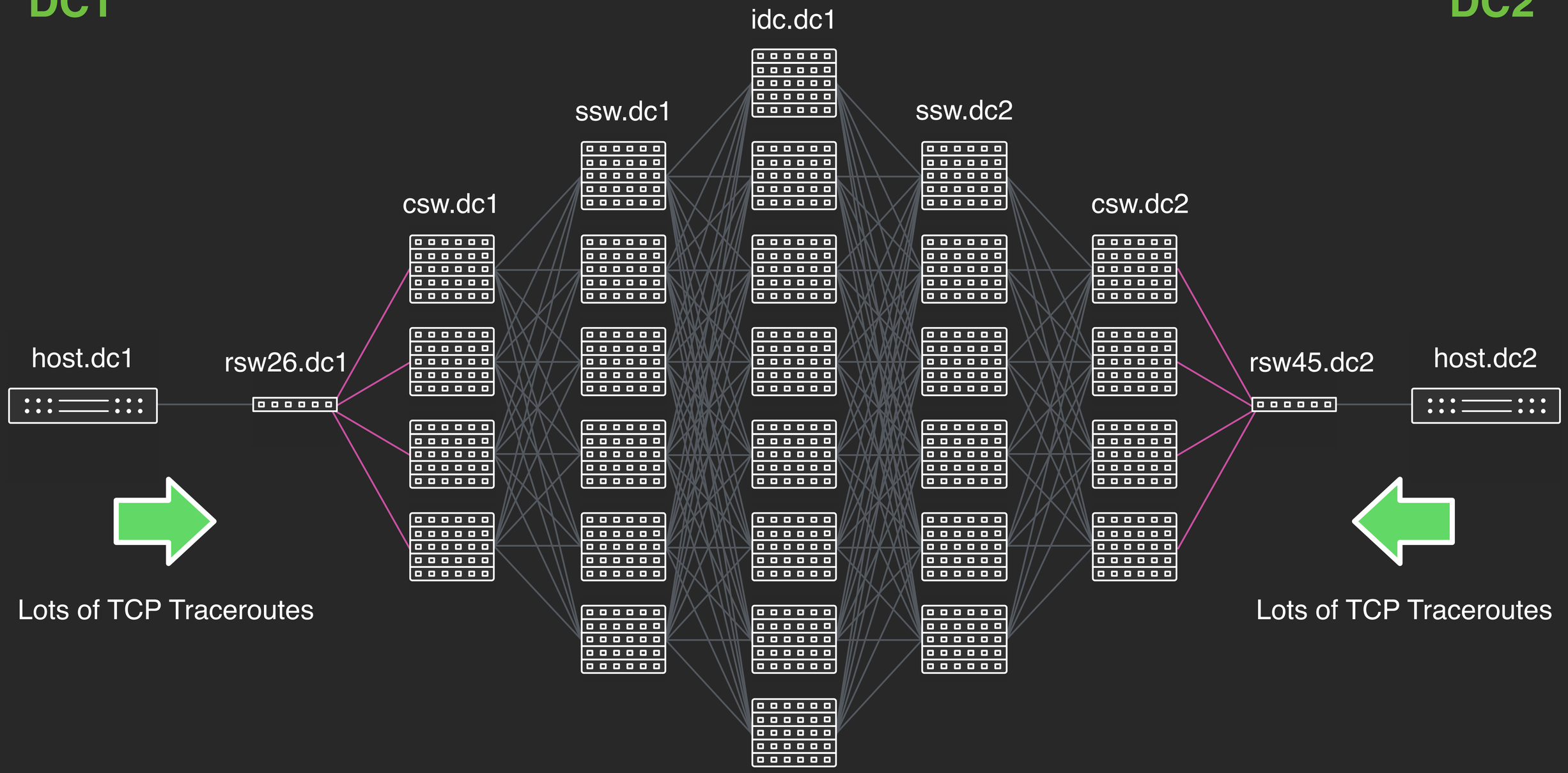
Alarming should be as accurate as possible with a clear path towards auto-remediation

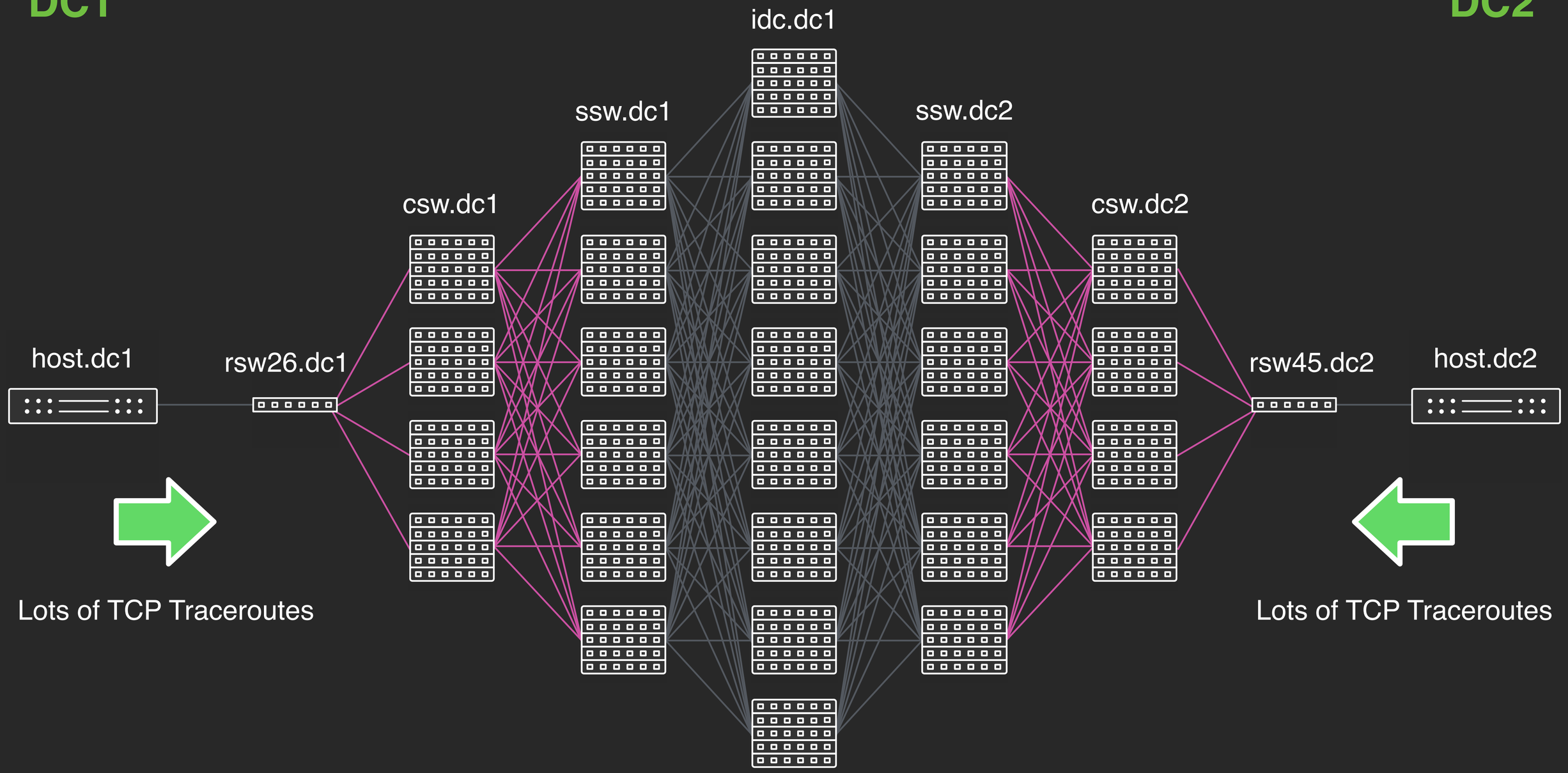

Where is the
bad link?
1
511
512
1024
R
R

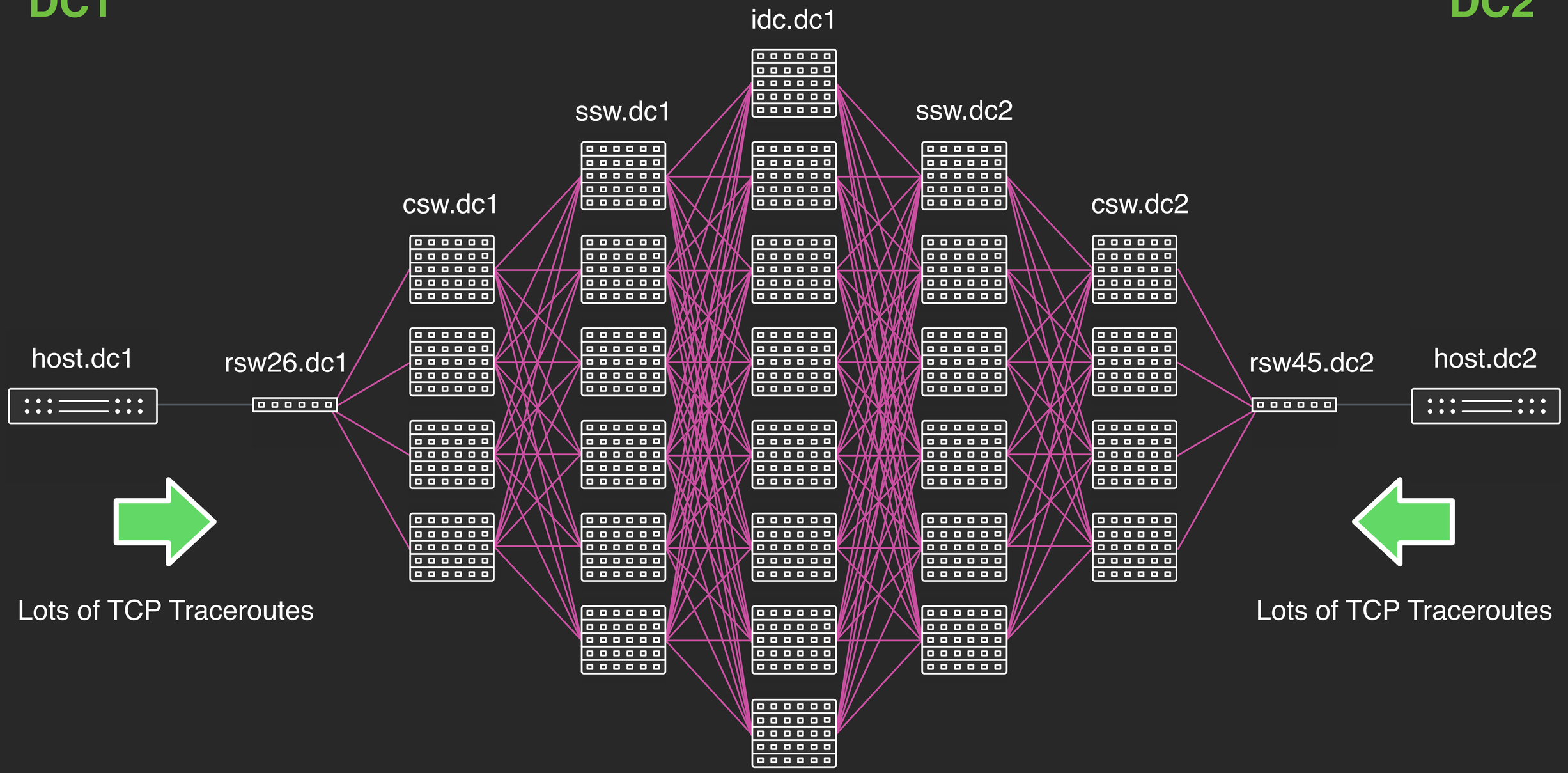

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2


DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

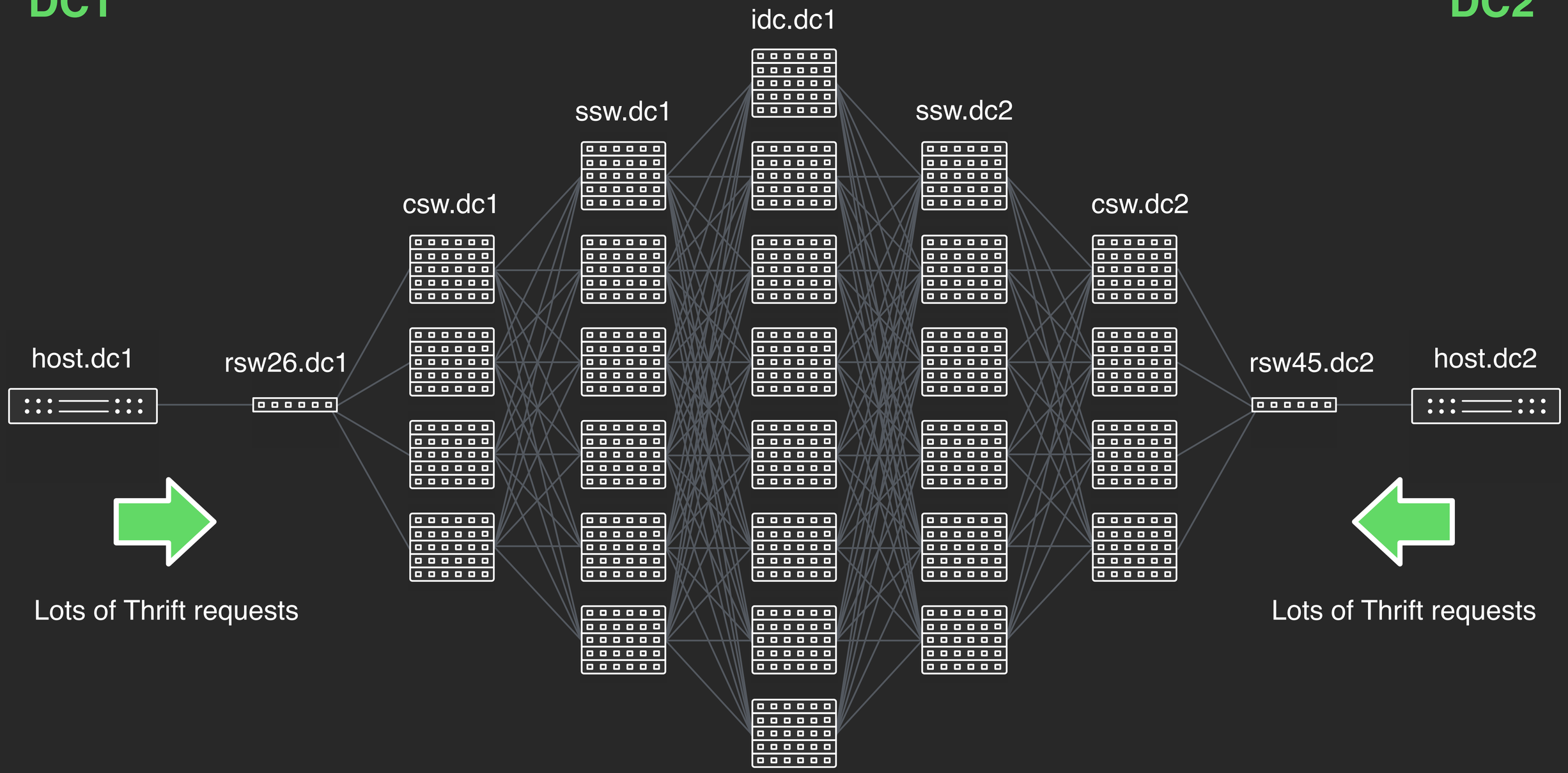

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2
Lots of TCP Traceroutes
Lots of TCP Traceroutes


DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2
Lots of TCP Traceroutes
Lots of TCP Traceroutes

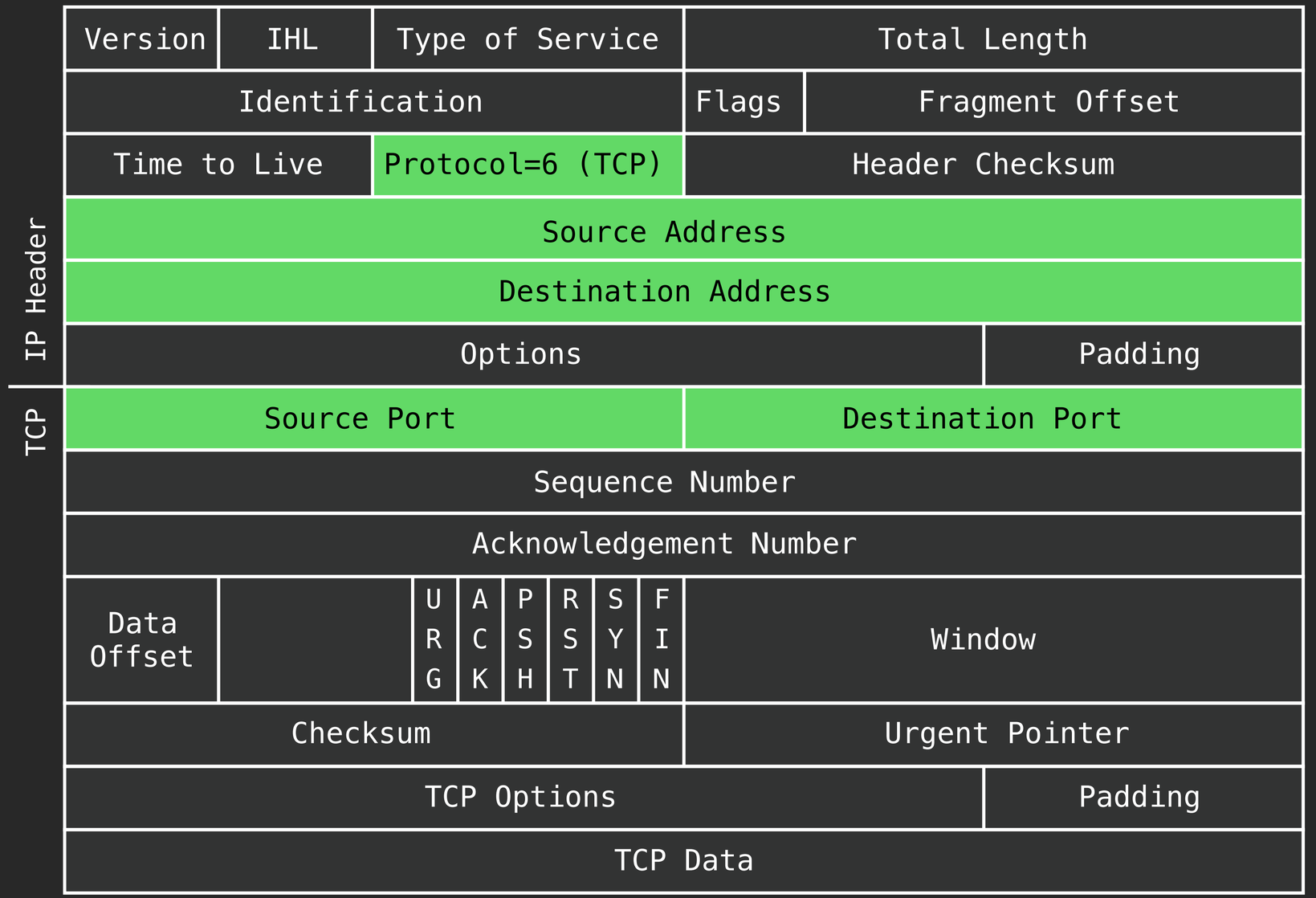

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2
Lots of TCP Traceroutes
Lots of TCP Traceroutes

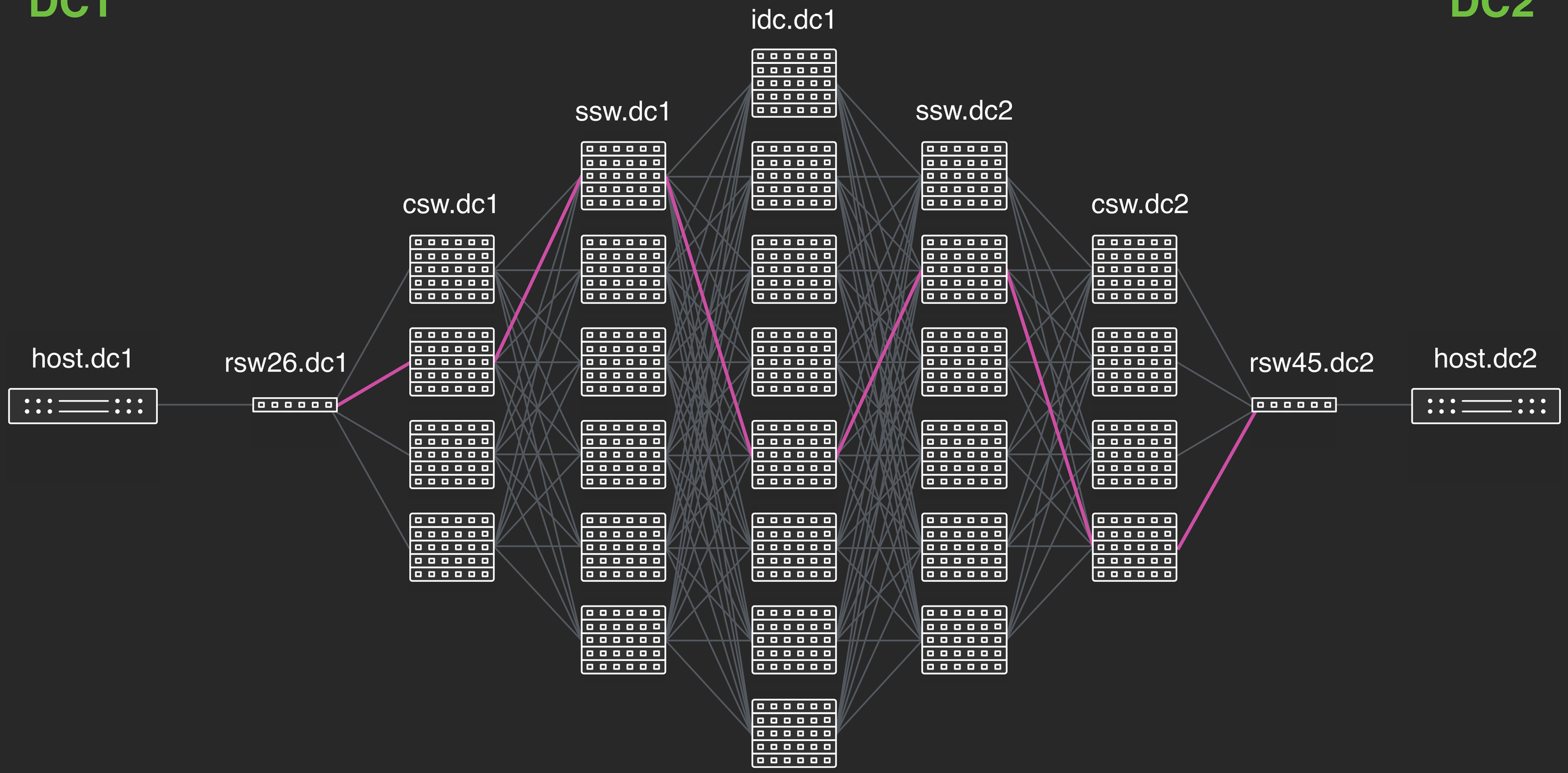

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2
Lots of TCP Traceroutes
Lots of TCP Traceroutes

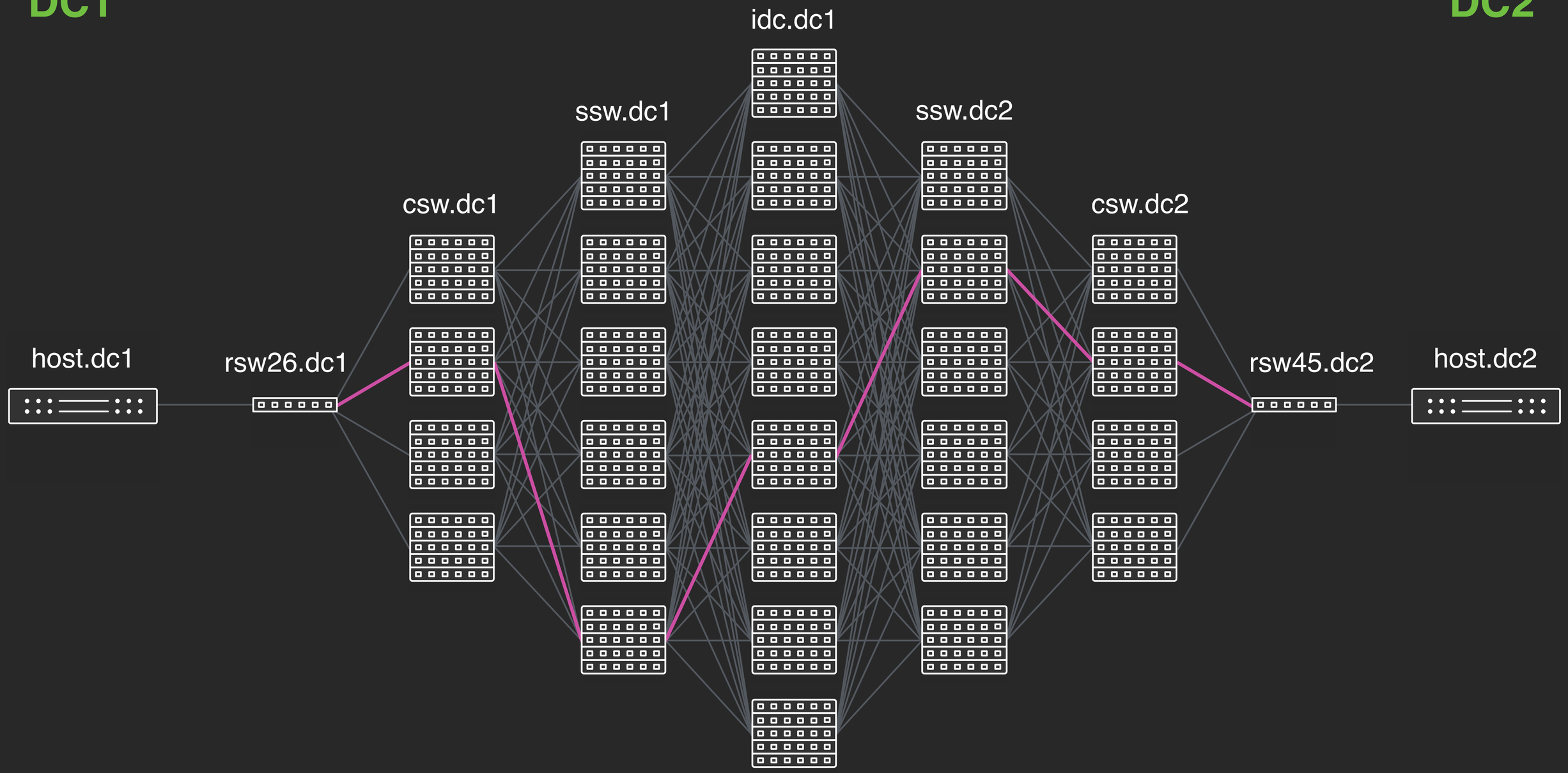
Src Port	Dst Port	Src IP	Dst IP	Links
50000	9999	10.0.0.10	10.99.99.99	e1.csw1.dc1 e5.ssw3.dc1 e19.idc6.dc1 e3.ssw4.dc2 e7.csw4.dc2
50001	9999	10.0.0.10	10.99.99.99	e1.csw4.dc1 e9.ssw2.dc1 e4.idc2.dc1 e8.ssw3.dc2 e7.csw3.dc2
50002	9999	10.0.0.10	10.99.99.99	e1.csw3.dc1 e5.ssw5.dc1 e19.idc6.dc1 e3.ssw4.dc2 e7.csw4.dc2

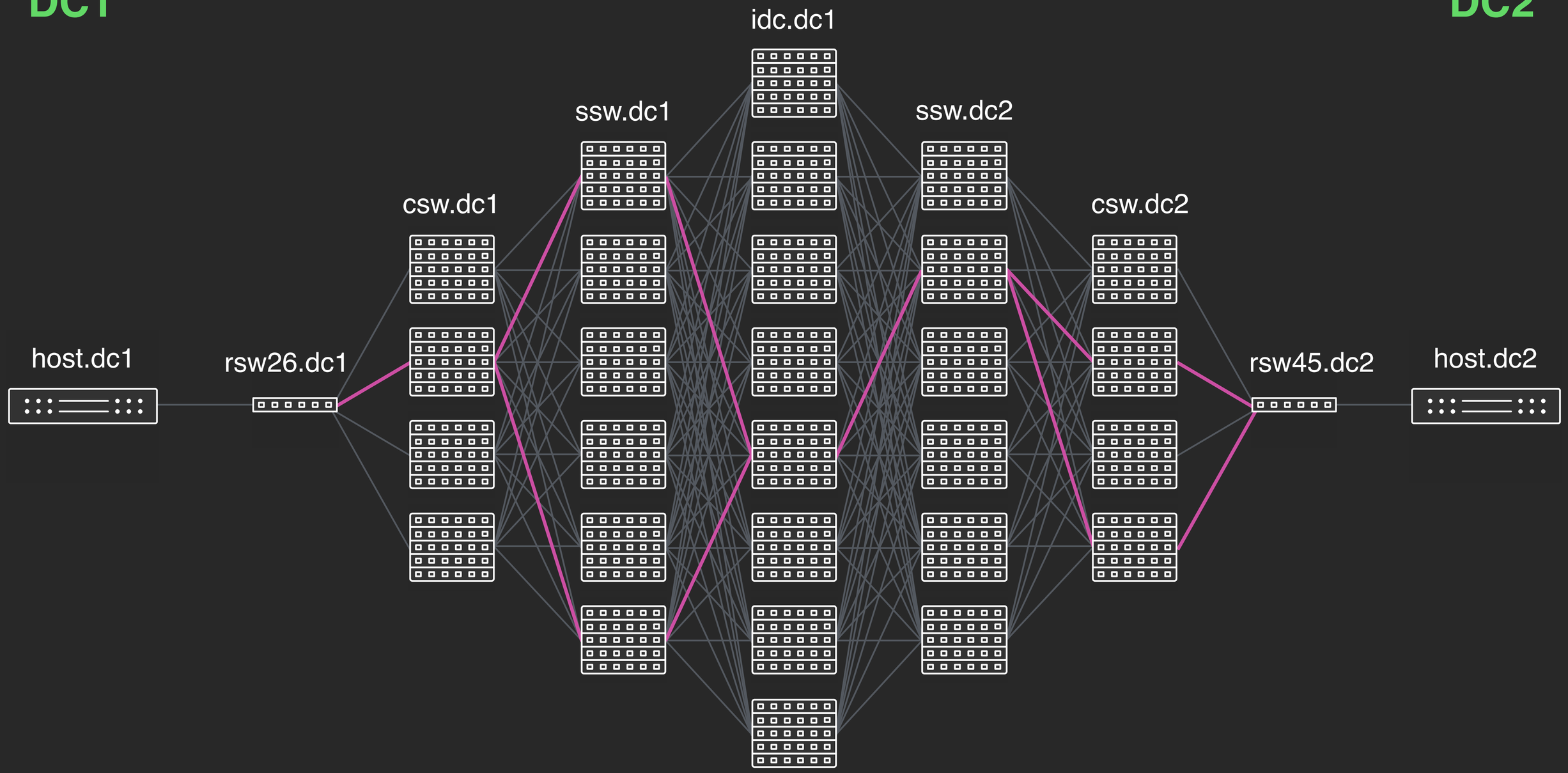

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2
Lots of Thrift requests
Lots of Thrift requests

TCP/IP Packet

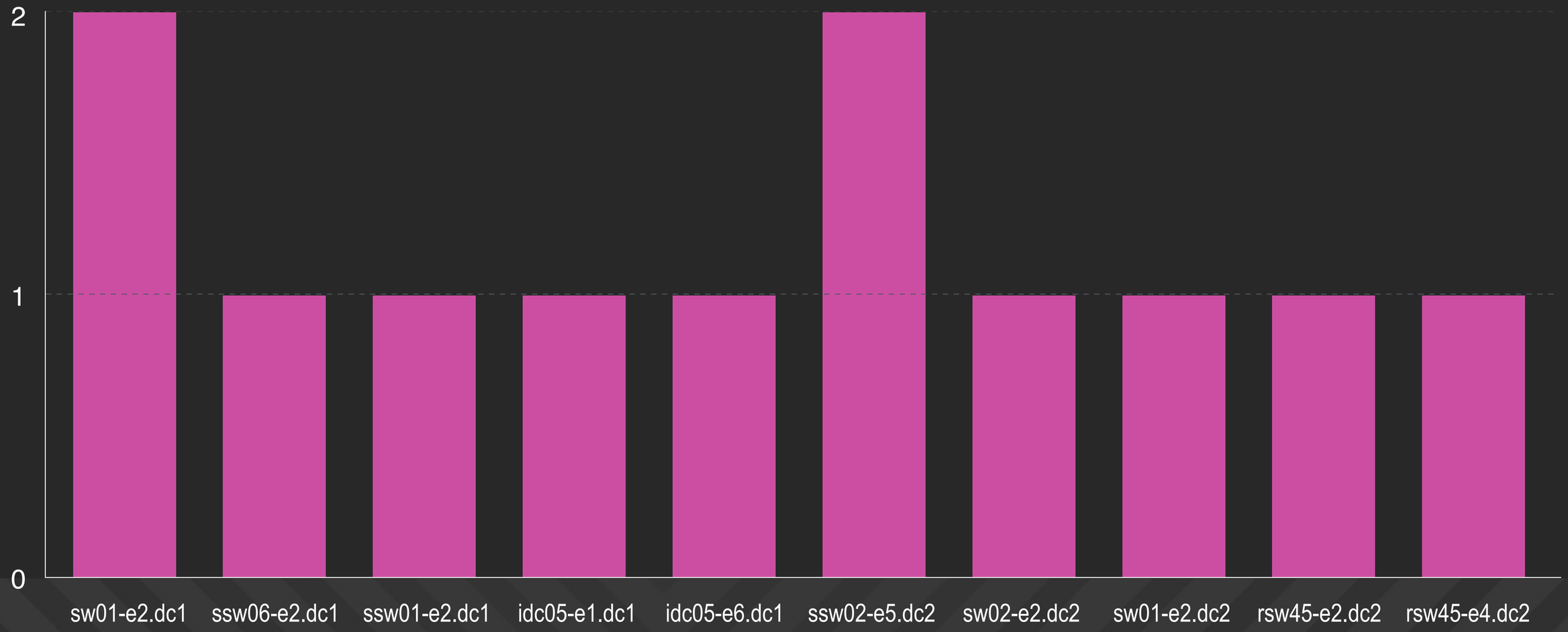


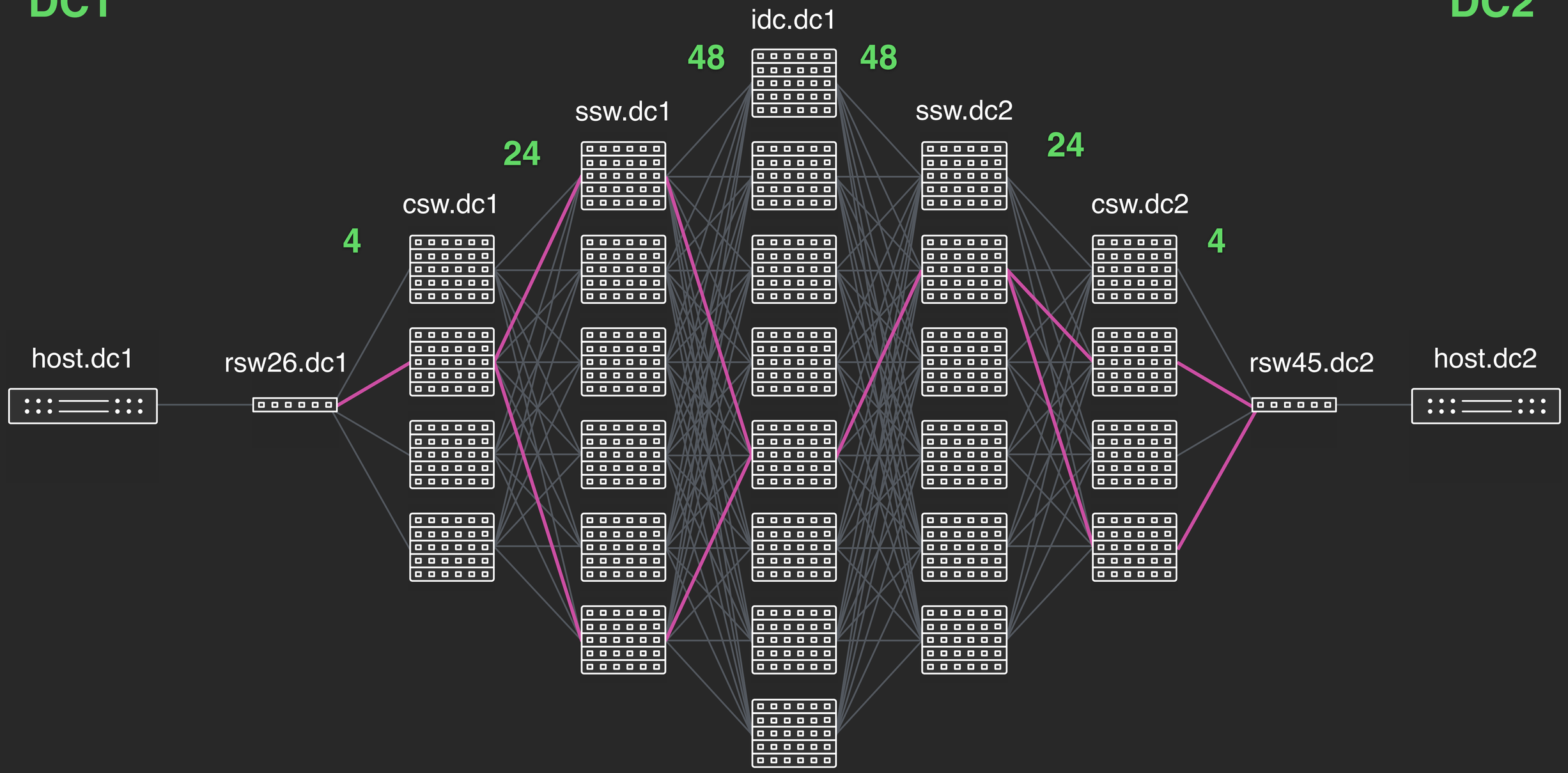

DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2


DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2


DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

Error count


2
1
0
sw01-e2.dc1
ssw06-e2.dc1
ssw01-e2.dc1
idc05-e1.dc1
idc05-e6.dc1
ssw02-e5.dc2
sw02-e2.dc2
sw01-e2.dc2
rsw45-e2.dc2
rsw45-e4.dc2


DC1
DC2
idc.dc1
48
48
ssw.dc1
ssw.dc2
24
24
csw.dc1
csw.dc2
4
4
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

Error Count for a Device

=

% of requests we lost

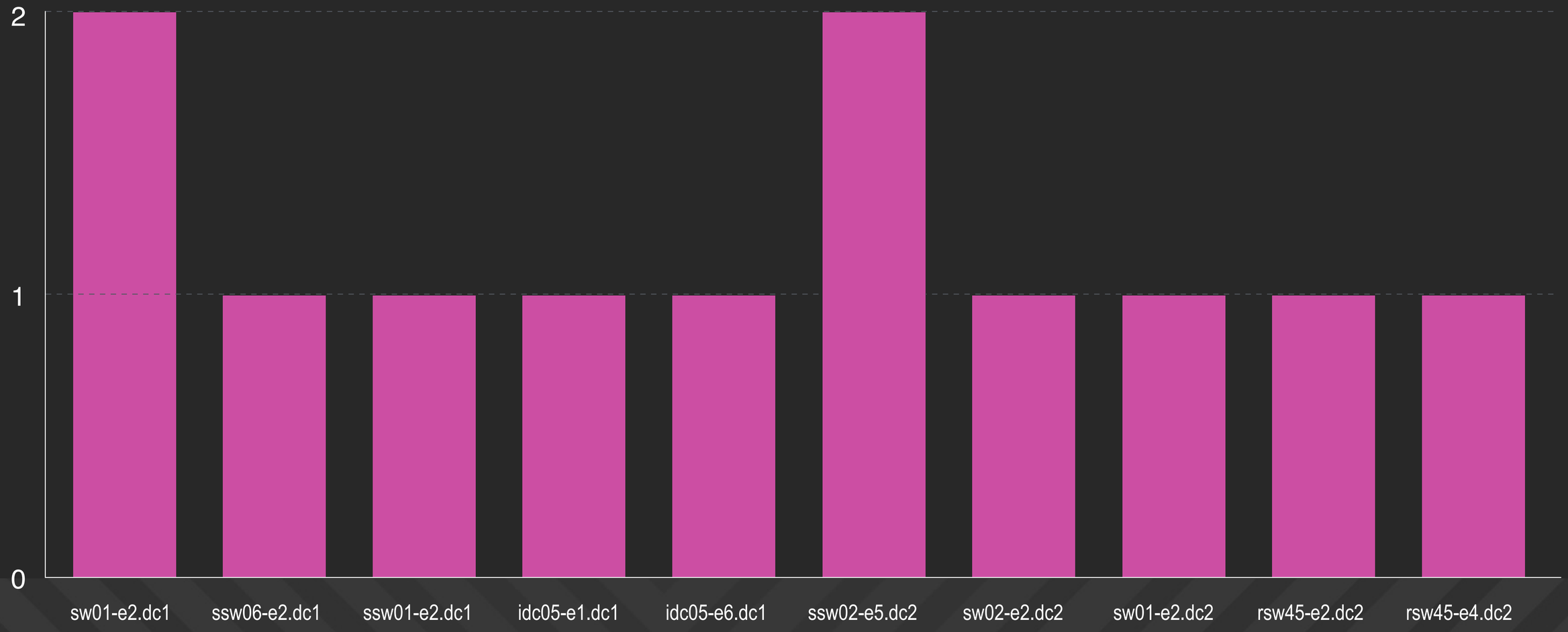
*

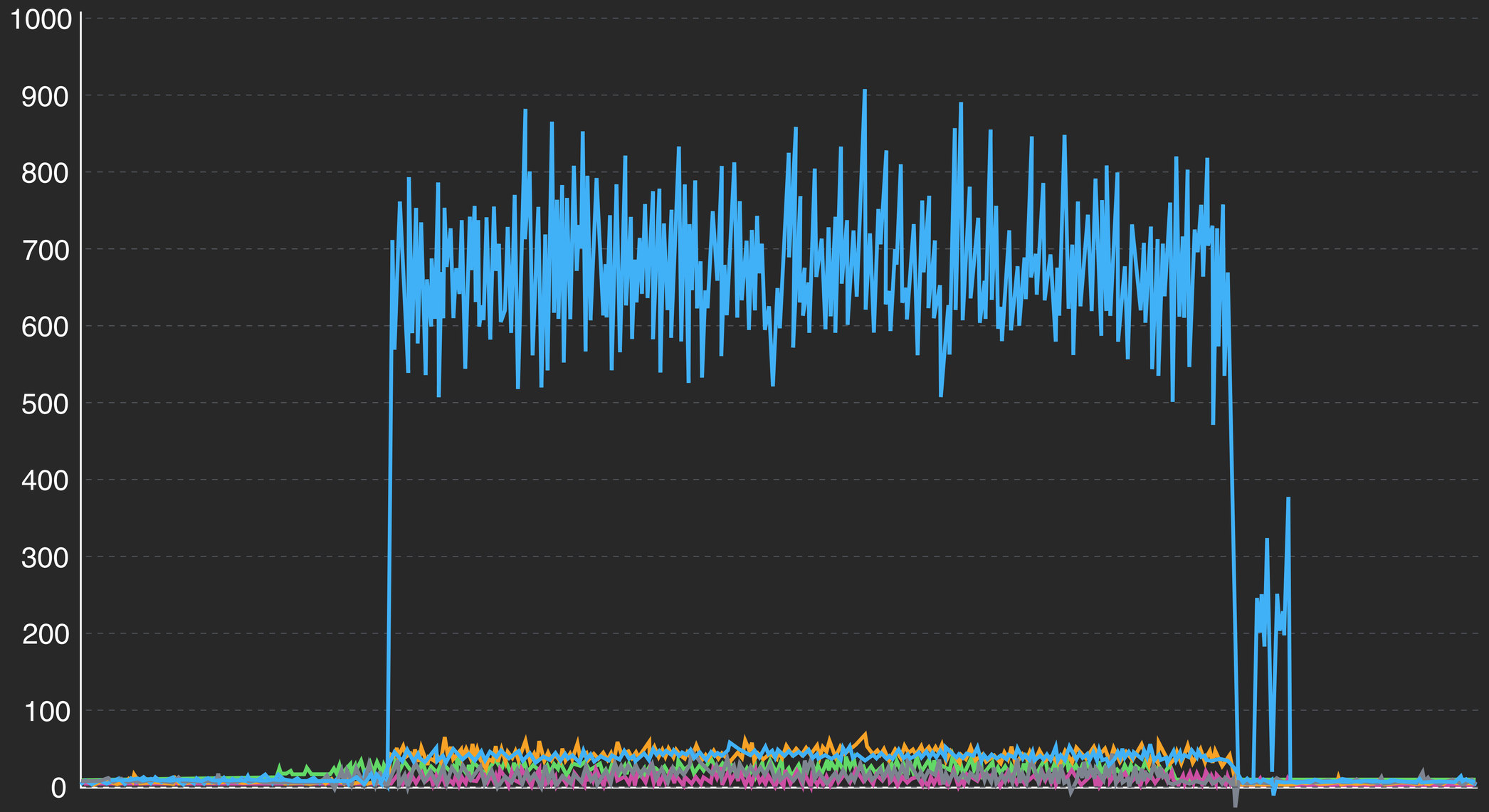
Number of ECMP links at that layer in the Network

*

% of lost requests this circuit was in

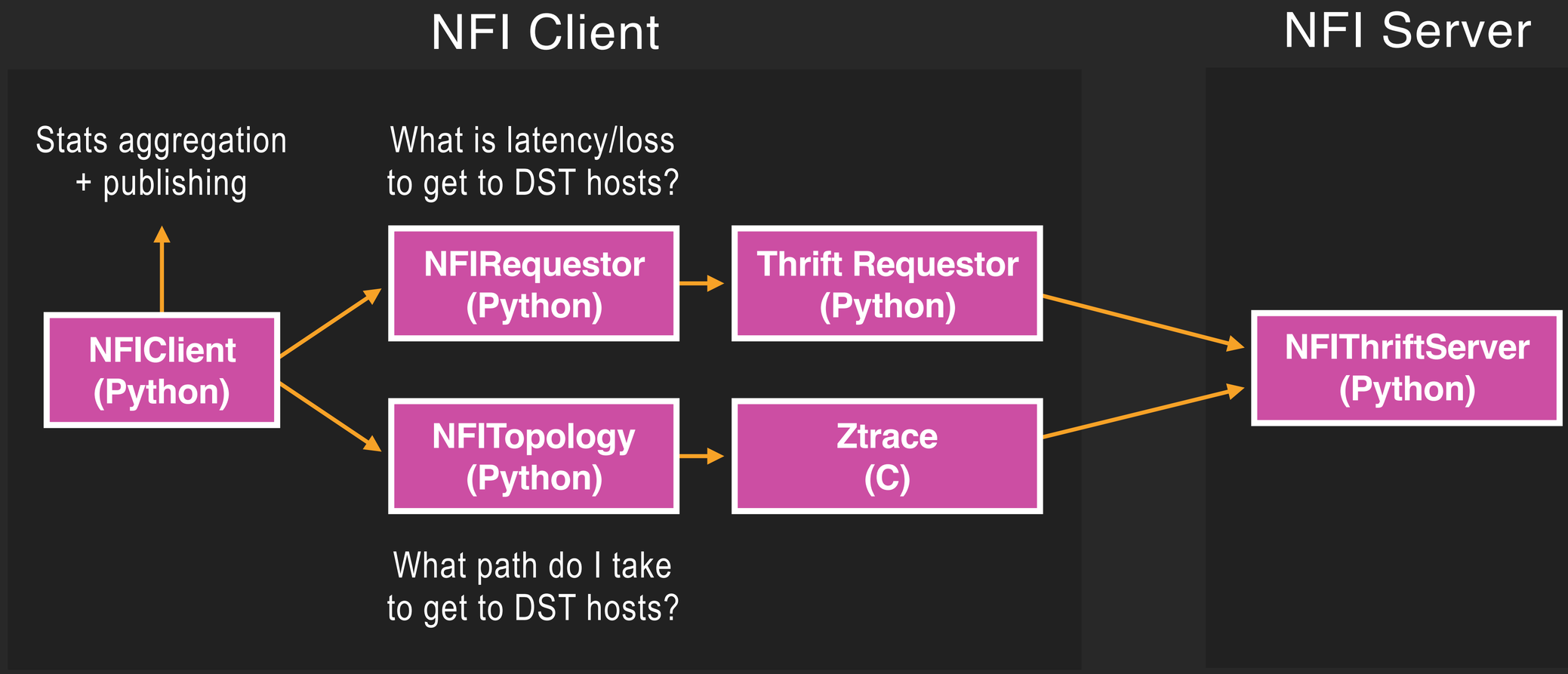
Error count


2
1
0
sw01-e2.dc1
ssw06-e2.dc1
ssw01-e2.dc1
idc05-e1.dc1
idc05-e6.dc1
ssw02-e5.dc2
sw02-e2.dc2
sw01-e2.dc2
rsw45-e2.dc2
rsw45-e4.dc2

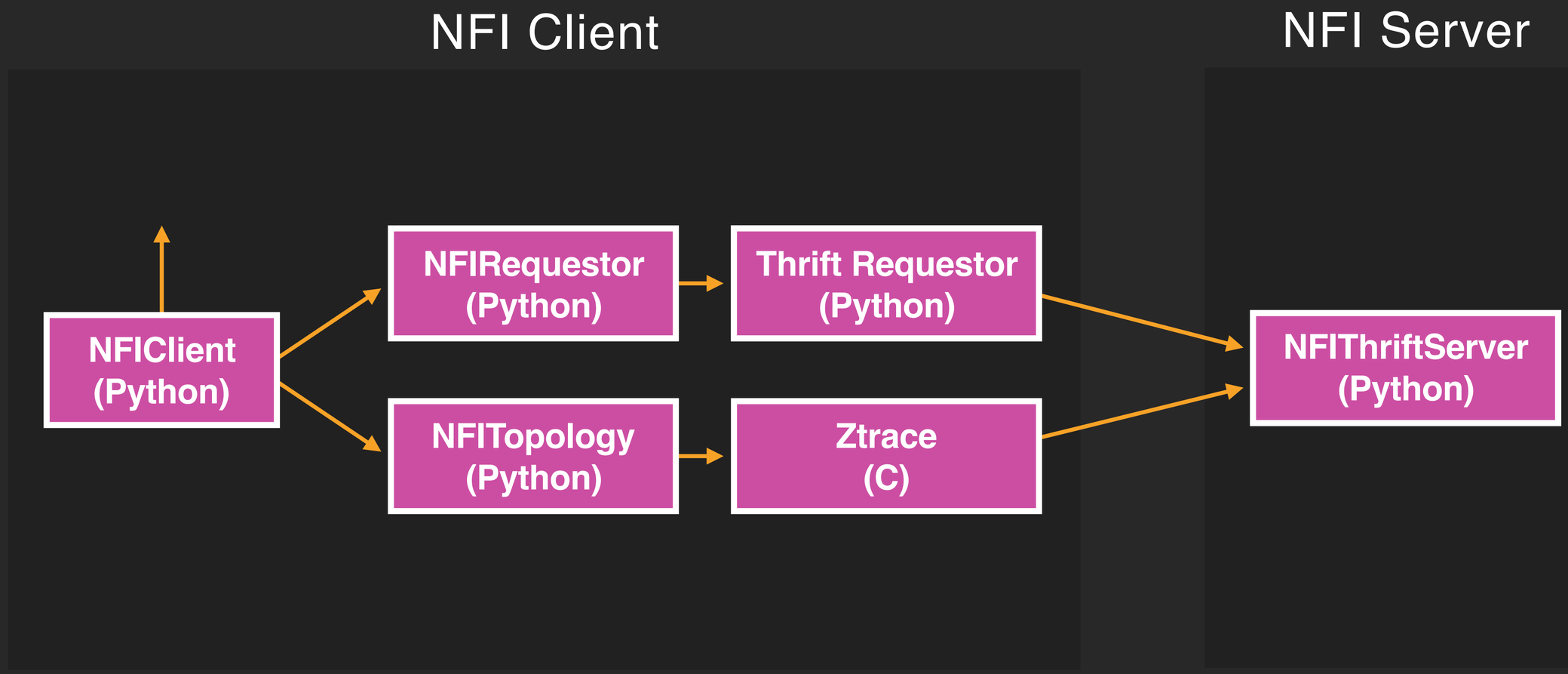

1000
900
800
700
600
500
400
300
200
100
0

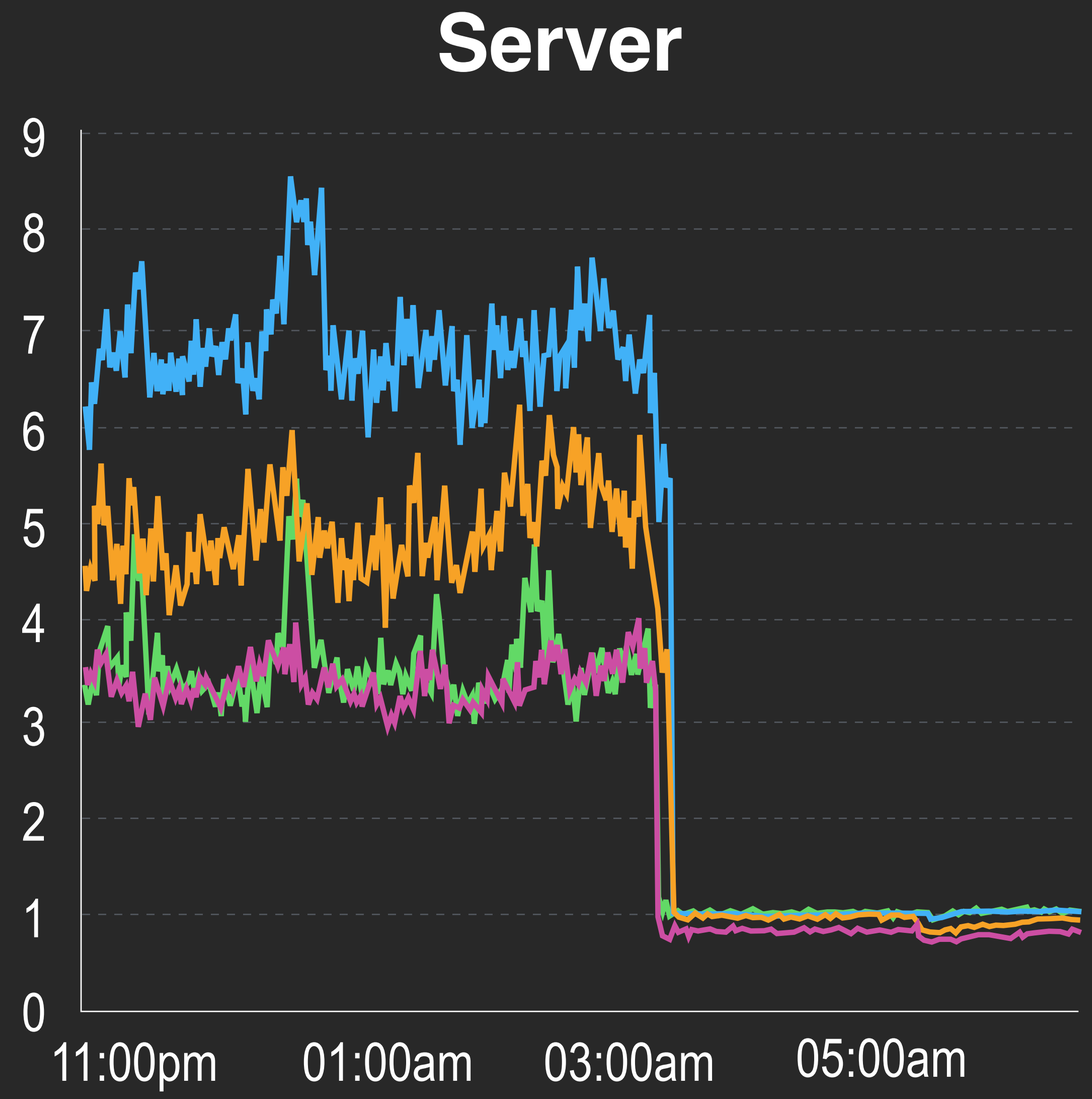
Done is better than perfect?

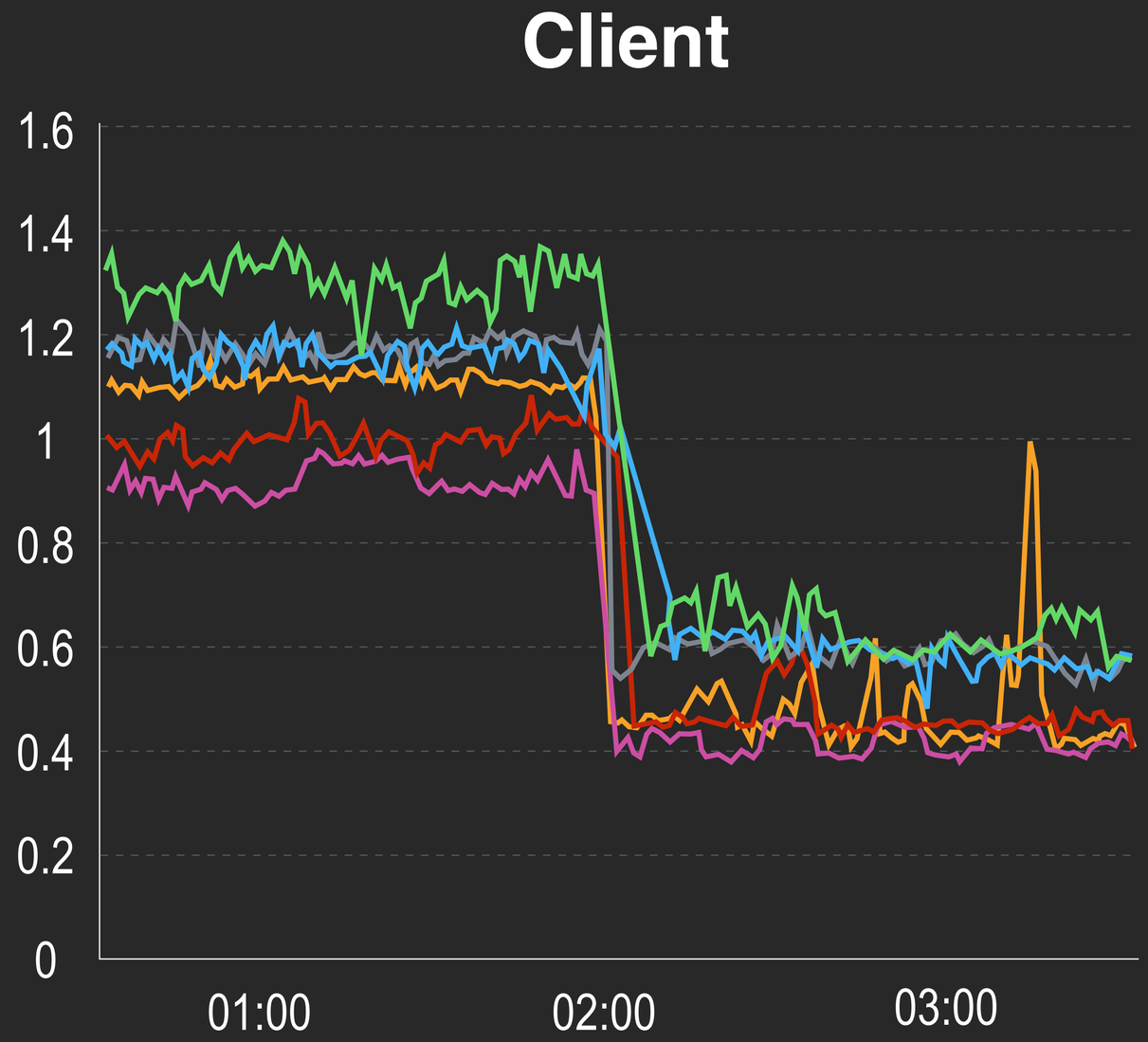
Modular components

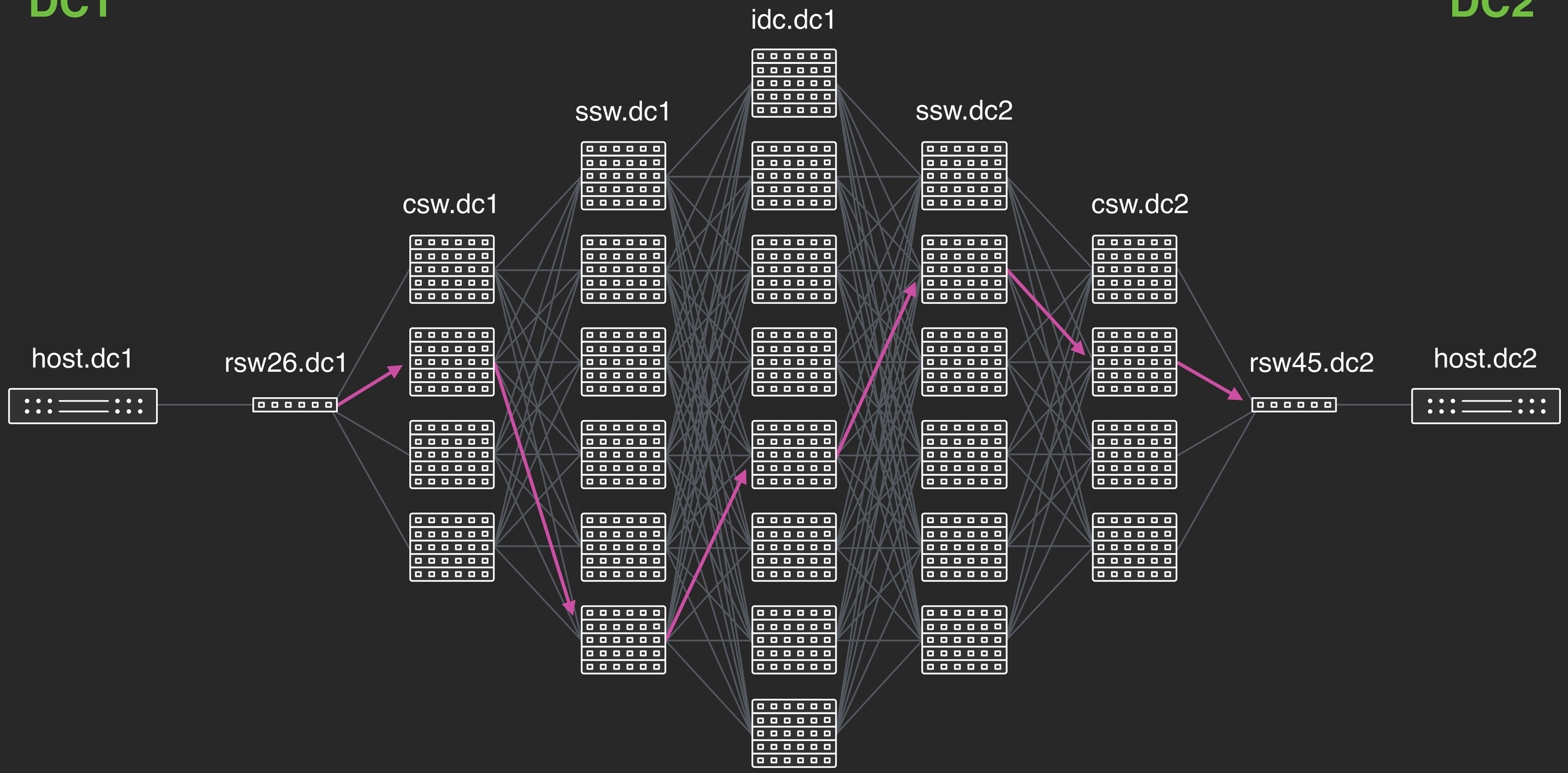


Software evolution

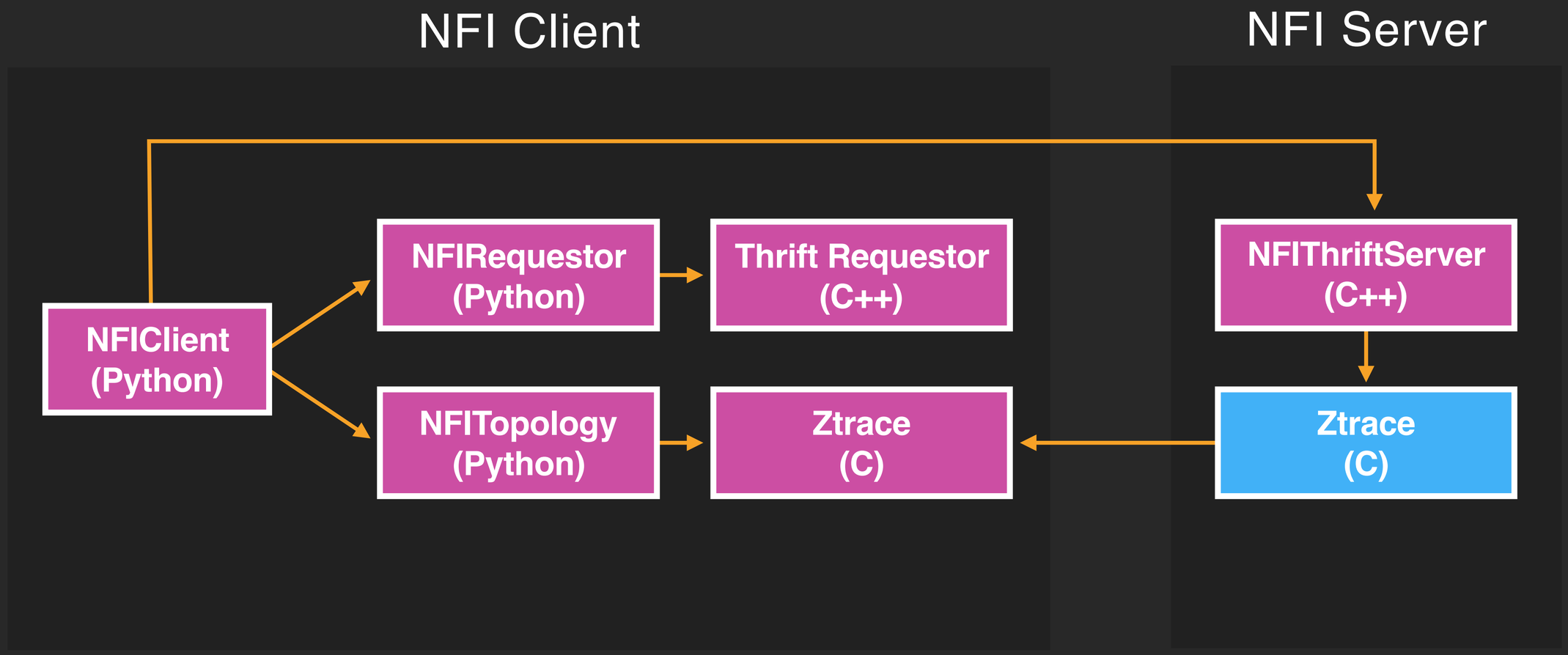


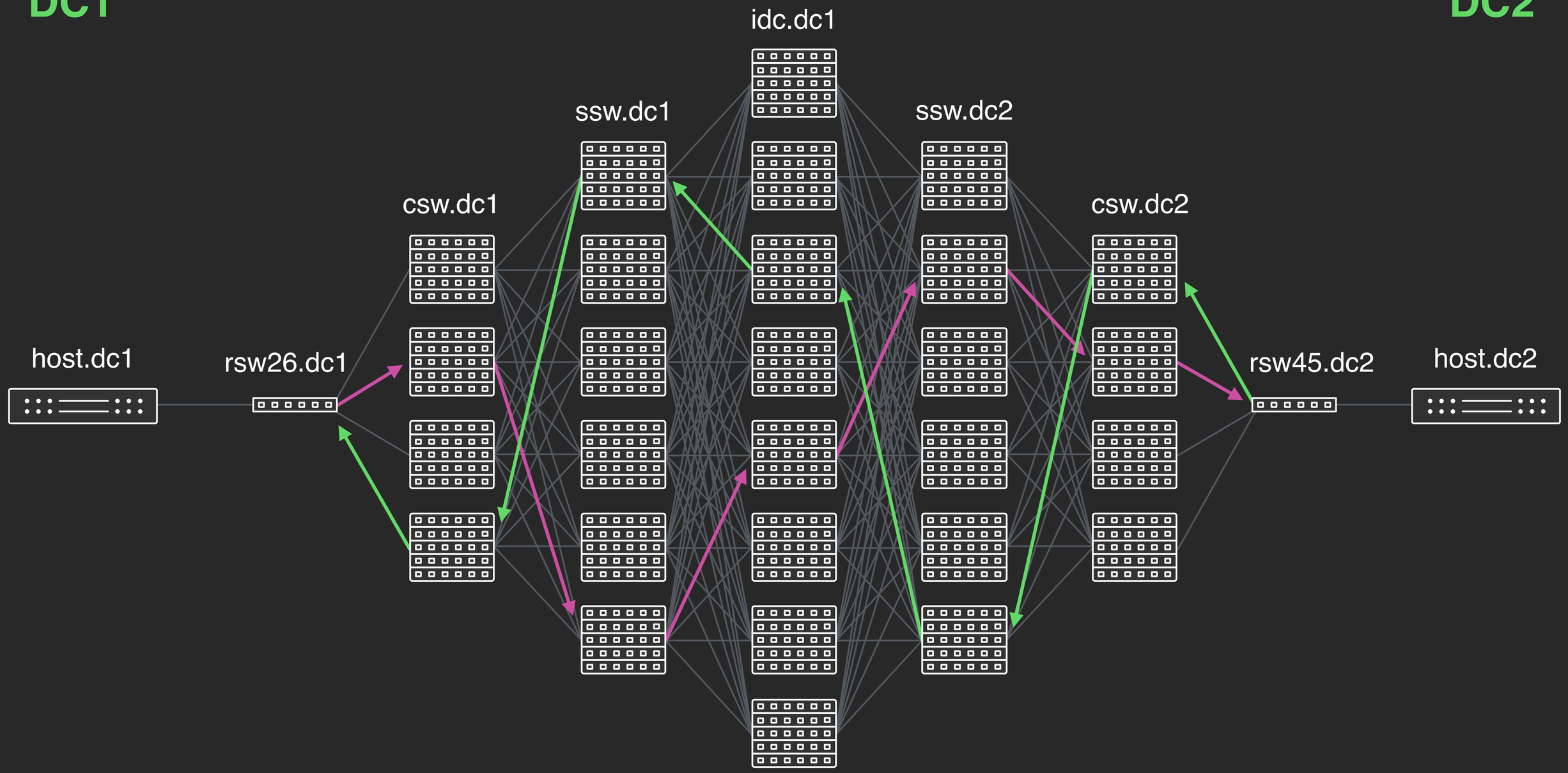

Server
9
8
7
6
5
4
3
2
1
0
11:00pm
01:00am
03:00am
05:00am


Client
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
01:00
02:00
03:00


DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

Reverse Traceroute

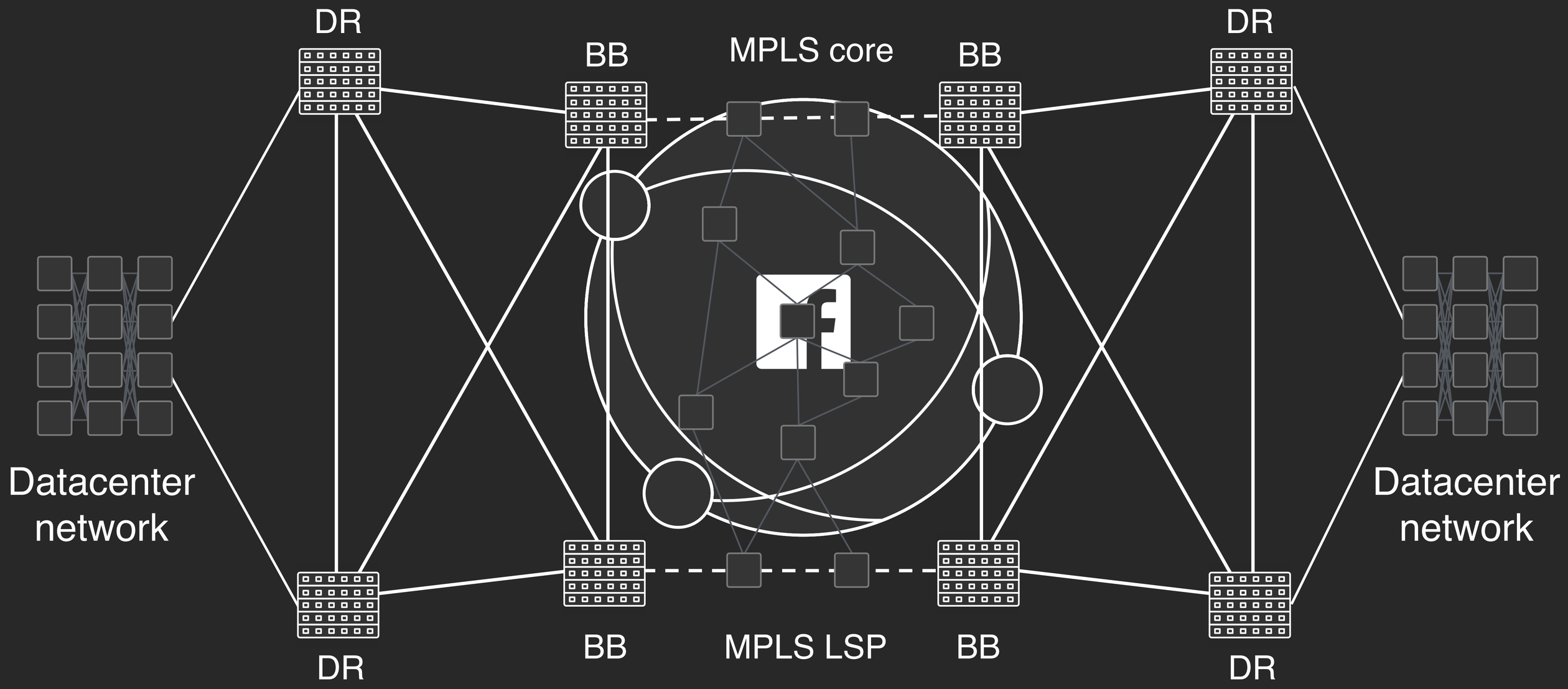



DC1
DC2
idc.dc1
ssw.dc1
ssw.dc2
csw.dc1
csw.dc2
host.dc1
rsw26.dc1
rsw45.dc2
host.dc2

What's next for NFI?

Running on Wedge

Backbone network




Distributed Systems Hate Packet-Loss