System Administration

Week 06, Segment 2

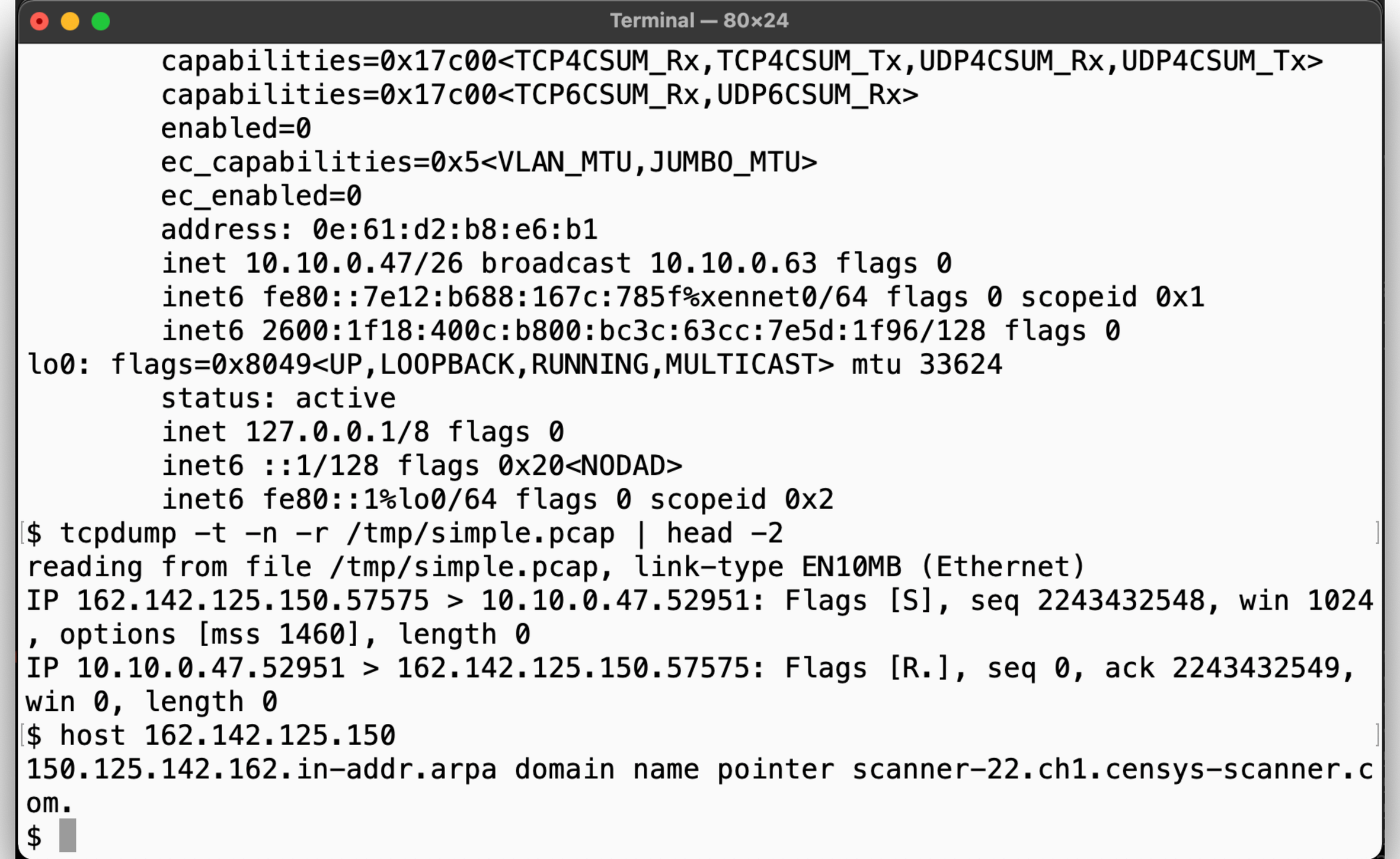
Networking II: A Simple Request II

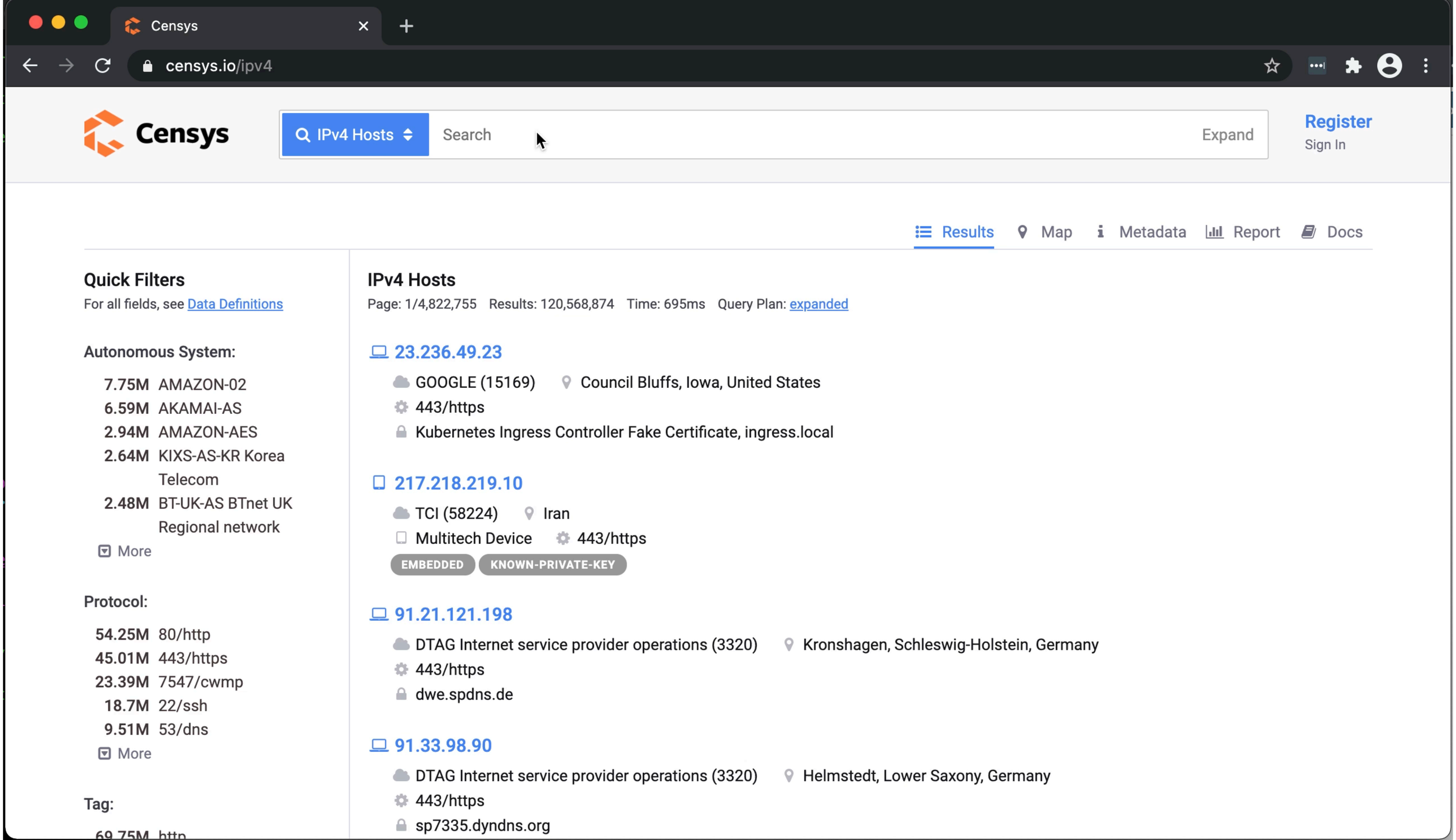
Department of Computer Science
Stevens Institute of Technology

Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/615/

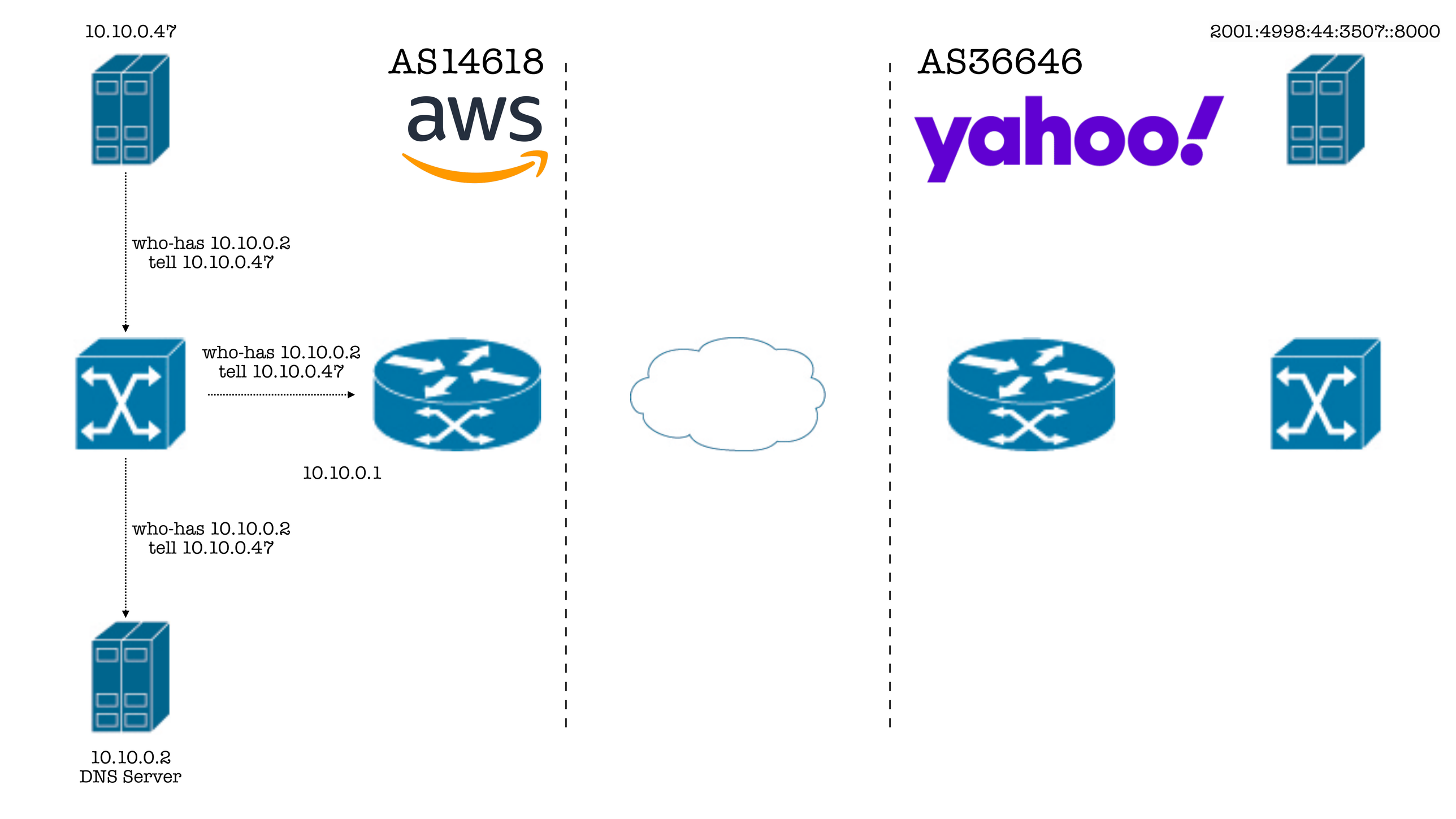
Terminal — 80×24

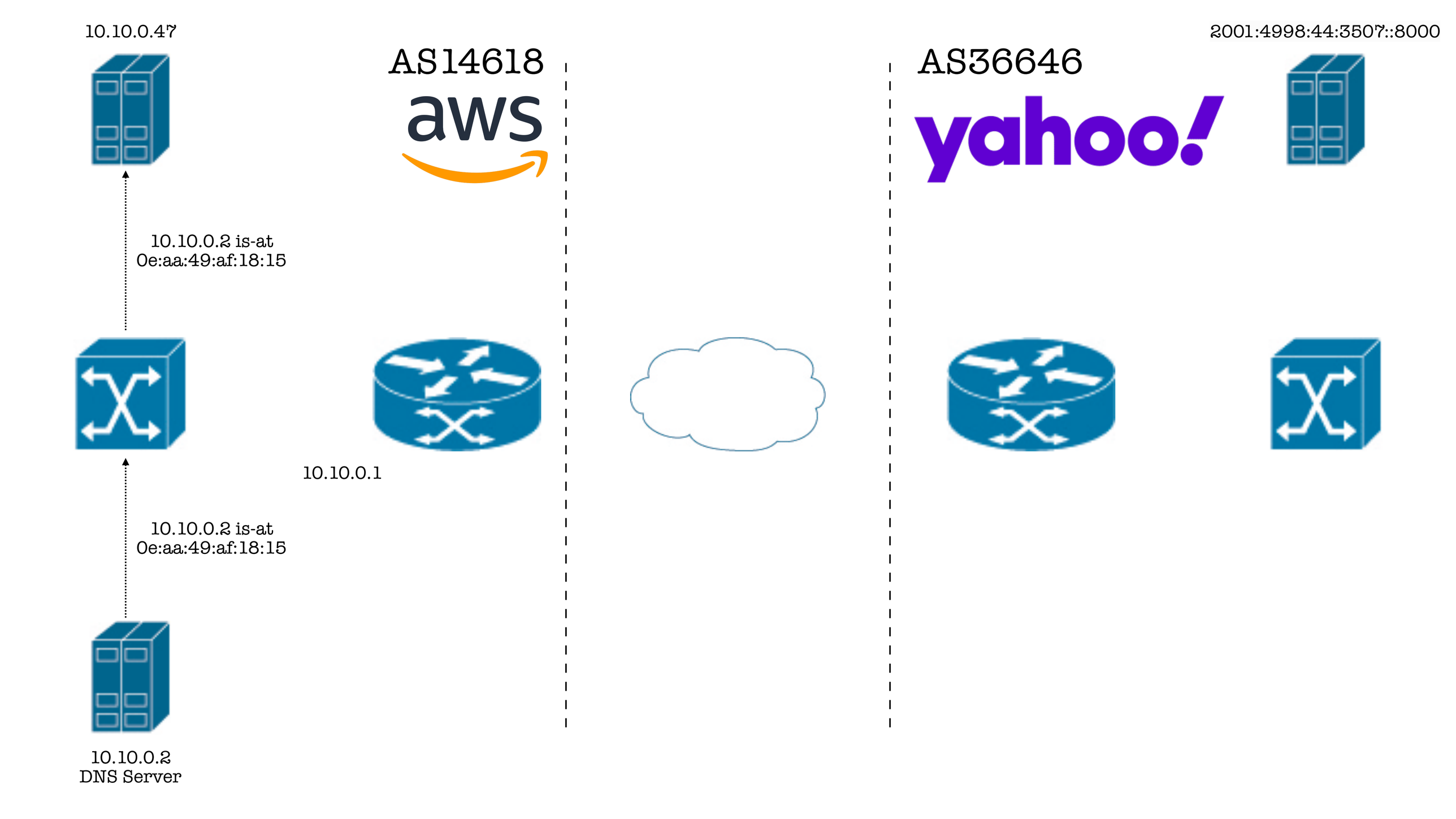
```
# tcpdump -w /tmp/simple.pcap port not 22 >/dev/null 2>&1 &
# arp -d -a
# ktrace -i telnet -K www.yahoo.com 80
Trying 2001:4998:44:3507::8000...
Connected to new-fp-shed.wg1.b.yahoo.com.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.0 200 OK
Date: Wed, 10 Mar 2021 23:17:18 GMT
Server: ATS
Cache-Control: no-store, no-cache, max-age=0, private
Content-Type: text/html
Content-Language: en
Expires: -1
X-Frame-Options: SAMEORIGIN
Content-Length: 12

Connection closed by foreign host.
# fg
tcpdump -w /tmp/simple.pcap port not 22 >/dev/null 2>&1 &
^C#
```

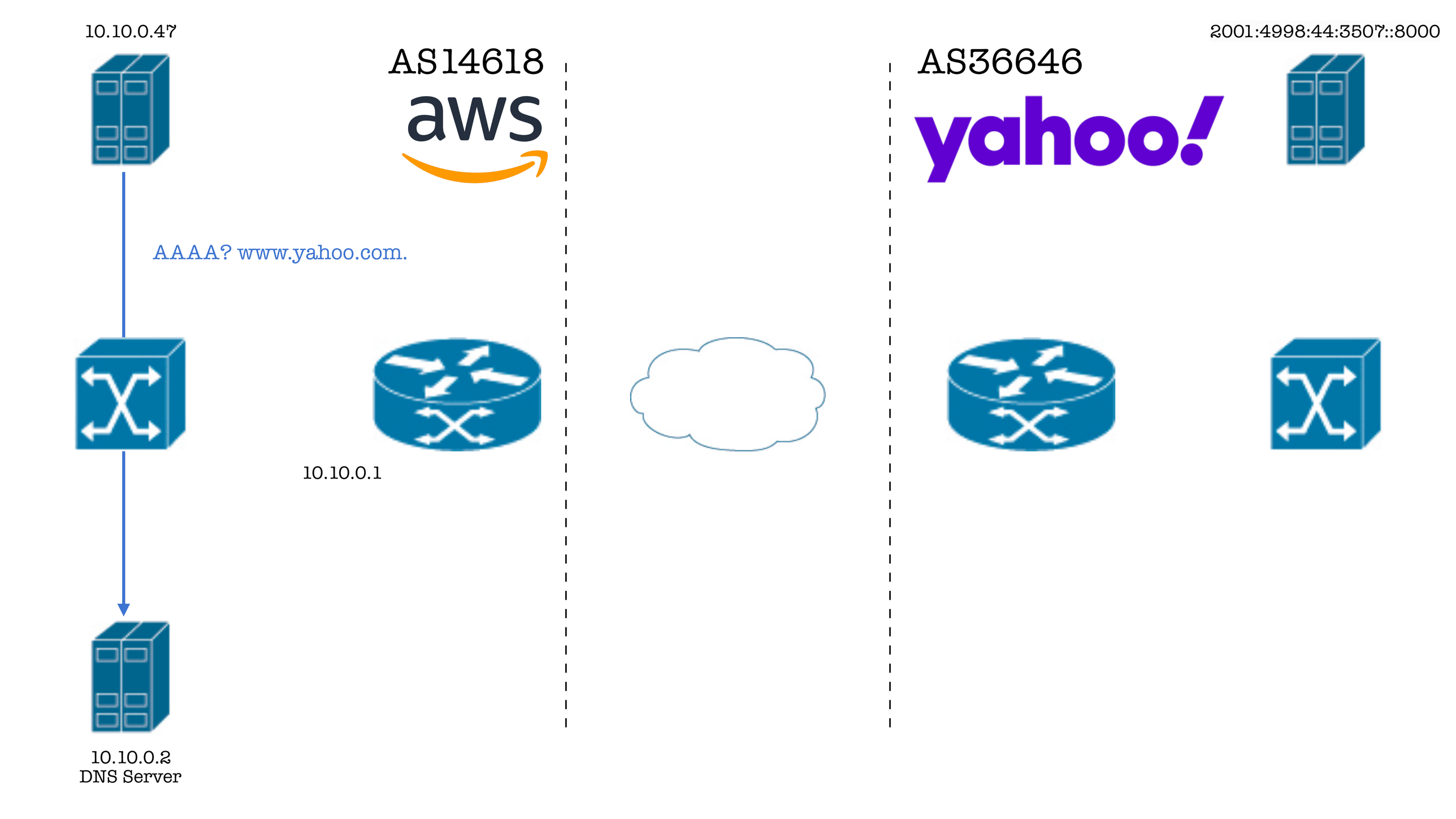

Terminal — 80×24
capabilities=0x17c00<TCP4CSUM_Rx,TCP4CSUM_Tx,UDP4CSUM_Rx,UDP4CSUM_Tx>
capabilities=0x17c00<TCP6CSUM_Rx,UDP6CSUM_Rx>
enabled=0
ec_capabilities=0x5<VLAN_MTU,JUMBO_MTU>
ec_enabled=0
address: 0e:61:d2:b8:e6:b1
inet 10.10.0.47/26 broadcast 10.10.0.63 flags 0
inet6 fe80::7e12:b688:167c:785f%xennet0/64 flags 0 scopeid 0x1
inet6 2600:1f18:400c:b800:bc3c:63cc:7e5d:1f96/128 flags 0
lo0: flags=0x8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33624
status: active
inet 127.0.0.1/8 flags 0
inet6 ::1/128 flags 0x20<NODAD>
inet6 fe80::1%lo0/64 flags 0 scopeid 0x2
$ tcpdump -t -n -r /tmp/simple.pcap | head -2
reading from file /tmp/simple.pcap, link-type EN10MB (Ethernet)
IP 162.142.125.150.57575 > 10.10.0.47.52951: Flags [S], seq 2243432548, win 1024
, options [mss 1460], length 0
IP 10.10.0.47.52951 > 162.142.125.150.57575: Flags [R.], seq 0, ack 2243432549,
win 0, length 0
$ host 162.142.125.150
150.125.142.162.in-addr.arpa domain name pointer scanner-22.ch1.censys-scanner.c
om.
$

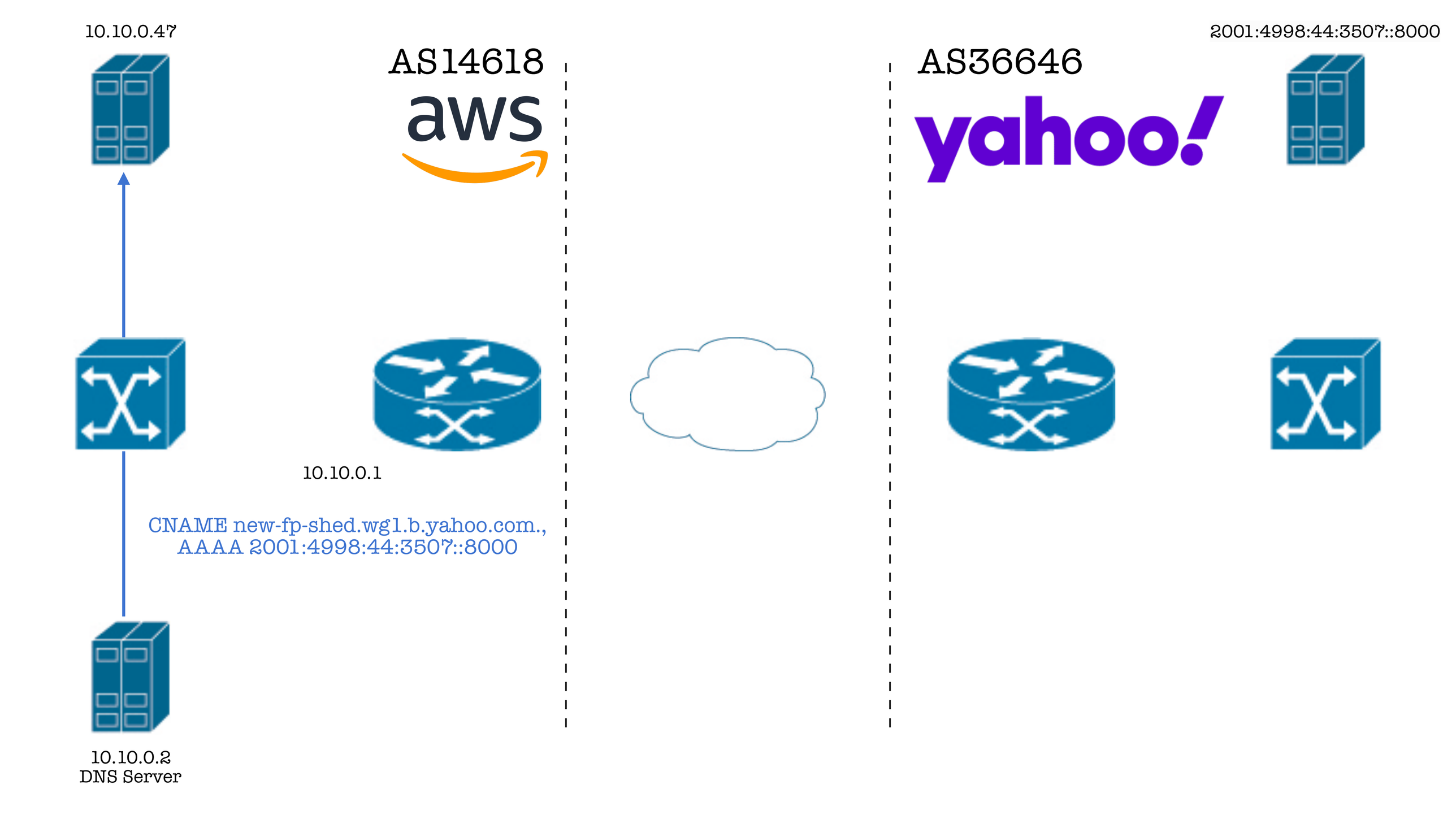

Censys
censys.io/ipv4
Censys
IPv4 Hosts
Search
Expand
Register
Sign In
Results
Map
Metadata
Report
Docs
Quick Filters
For all fields, see Data Definitions
Autonomous System:
7.75M AMAZON-02
6.59M AKAMAI-AS
2.94M AMAZON-AES
2.64M KIXS-AS-KR Korea Telecom
2.48M BT-UK-AS BTnet UK Regional network
More
Protocol:
54.25M 80/http
45.01M 443/https
23.39M 7547/cwmp
18.7M 22/ssh
9.51M 53/dns
More
Tag:
69.75M http
IPv4 Hosts
Page: 1/4,822,755 Results: 120,568,874 Time: 695ms Query Plan: expanded
23.236.49.23
GOOGLE (15169) Council Bluffs, Iowa, United States
443/https
Kubernetes Ingress Controller Fake Certificate, ingress.local
217.218.219.10
TCI (58224) Iran
Multitech Device 443/https
EMBEDDED
KNOWN-PRIVATE-KEY
91.21.121.198
DTAG Internet service provider operations (3320) Kronshagen, Schleswig-Holstein, Germany
443/https
dwe.spdns.de
91.33.98.90
DTAG Internet service provider operations (3320) Helmstedt, Lower Saxony, Germany
443/https
sp7335.dyndns.org

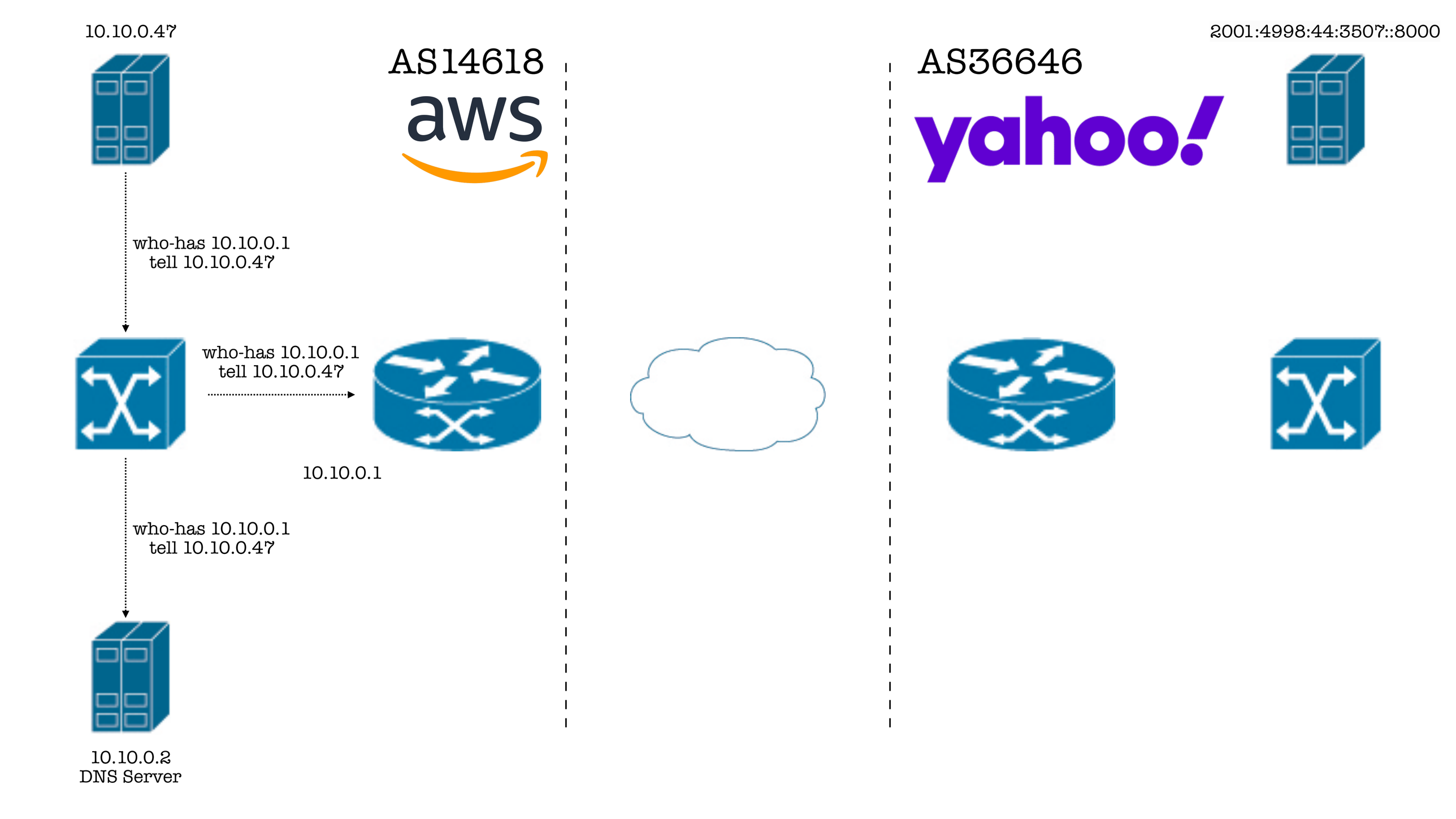

Terminal — 80×24
capabilities=0x17c00<TCP4CSUM_Rx,TCP4CSUM_Tx,UDP4CSUM_Rx,UDP4CSUM_Tx>
capabilities=0x17c00<TCP6CSUM_Rx,UDP6CSUM_Rx>
enabled=0
ec_capabilities=0x5<VLAN_MTU,JUMBO_MTU>
ec_enabled=0
address: 0e:61:d2:b8:e6:b1
inet 10.10.0.47/26 broadcast 10.10.0.63 flags 0
inet6 fe80::7e12:b688:167c:785f%xennet0/64 flags 0 scopeid 0x1
inet6 2600:1f18:400c:b800:bc3c:63cc:7e5d:1f96/128 flags 0
lo0: flags=0x8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33624
status: active
inet 127.0.0.1/8 flags 0
inet6 ::1/128 flags 0x20<NODAD>
inet6 fe80::1%lo0/64 flags 0 scopeid 0x2
$ tcpdump -t -n -r /tmp/simple.pcap | head -2
reading from file /tmp/simple.pcap, link-type EN10MB (Ethernet)
IP 162.142.125.150.57575 > 10.10.0.47.52951: Flags [S], seq 2243432548, win 1024
, options [mss 1460], length 0
IP 10.10.0.47.52951 > 162.142.125.150.57575: Flags [R.], seq 0, ack 2243432549,
win 0, length 0
$ host 162.142.125.150
150.125.142.162.in-addr.arpa domain name pointer scanner-22.ch1.censys-scanner.c
om.
$

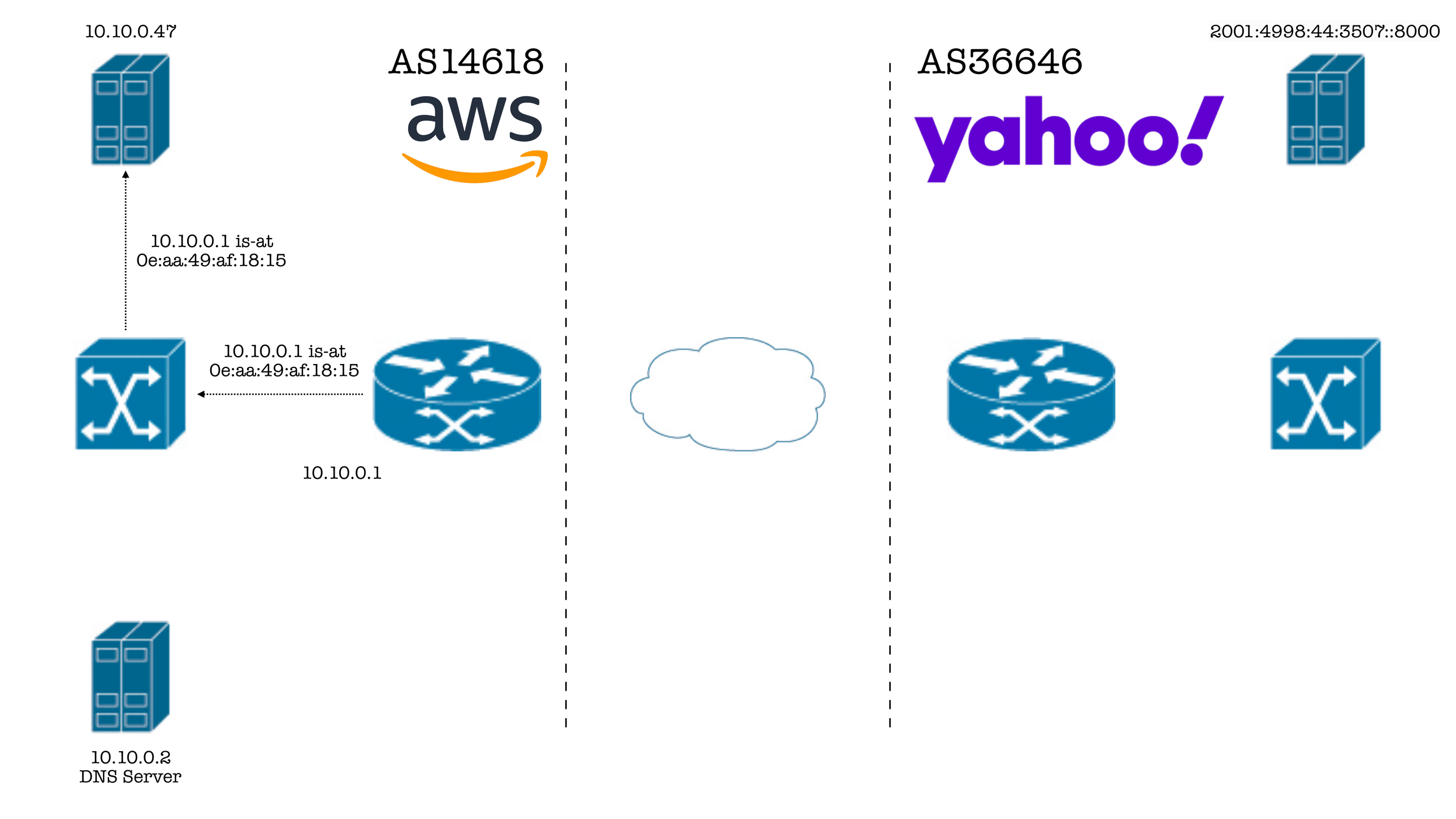

10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
who-has 10.10.0.2
tell 10.10.0.47
who-has 10.10.0.2
tell 10.10.0.47
10.10.0.1
who-has 10.10.0.2
tell 10.10.0.47
10.10.0.2
DNS Server

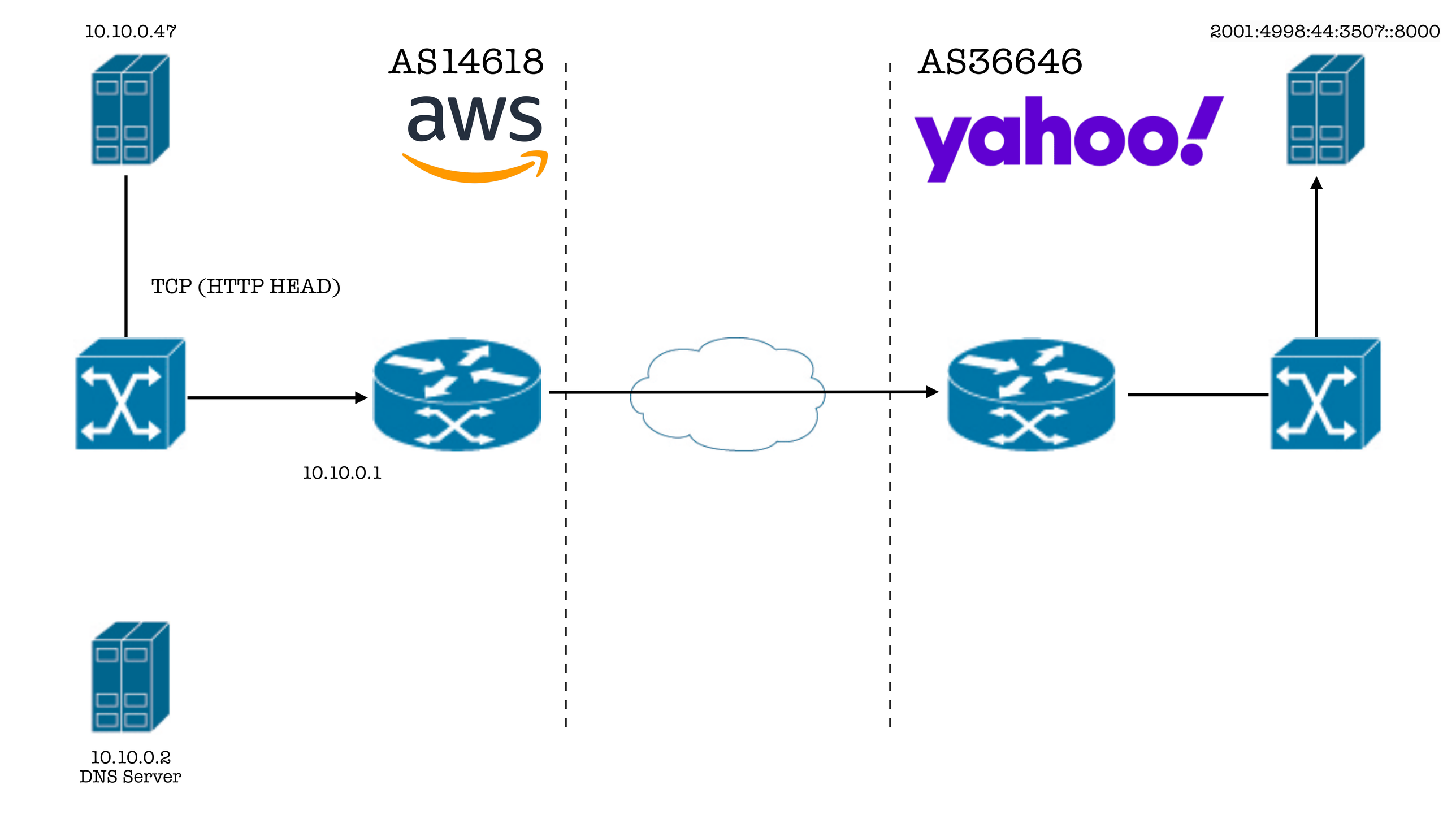

10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
10.10.0.2 is-at
0e:aa:49:af:18:15
10.10.0.1
10.10.0.2 is-at
0e:aa:49:af:18:15
10.10.0.2
DNS Server

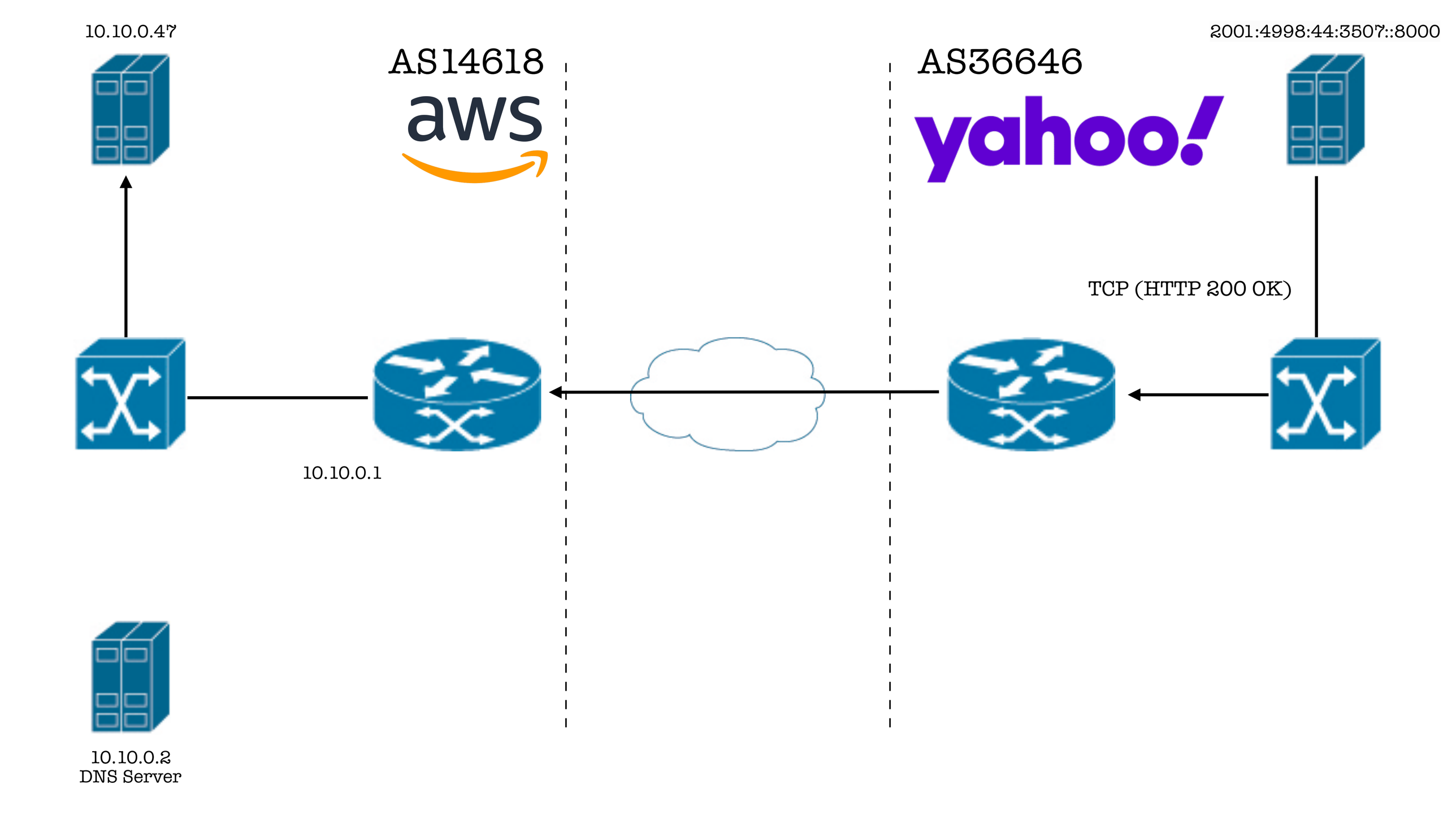

10.10.0.47
AS14618
aws
2001:4998:44:3507::8000
AS36646
yahoo!
AAAA? www.yahoo.com.
10.10.0.1
10.10.0.2
DNS Server

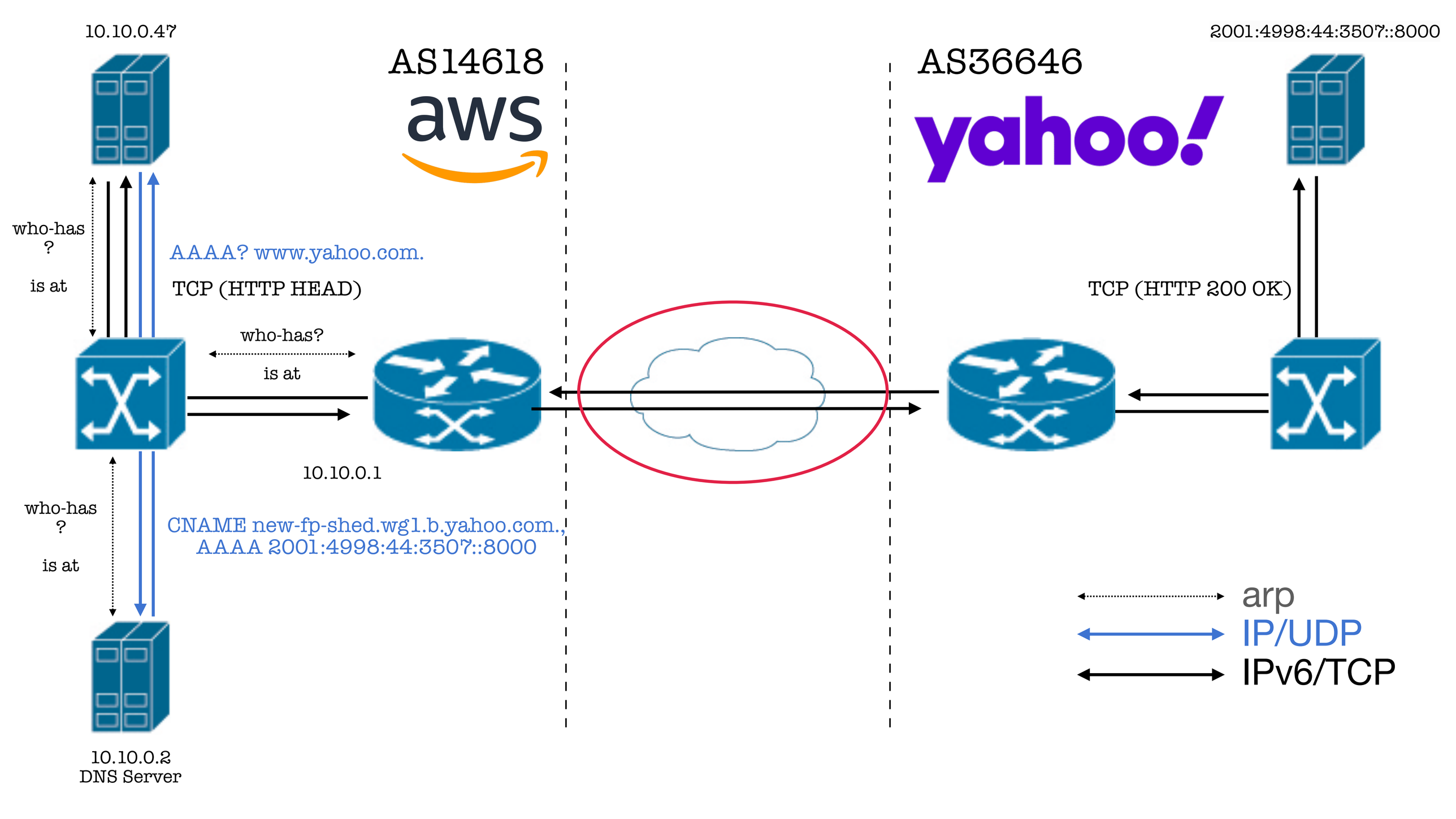

10.10.0.47
AS14618
aws
2001:4998:44:3507::8000
AS36646
yahoo!
10.10.0.1
CNAME new-fp-shed.wg1.b.yahoo.com.,
AAAA 2001:4998:44:3507::8000
10.10.0.2
DNS Server


10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
who-has 10.10.0.1
tell 10.10.0.47
who-has 10.10.0.1
tell 10.10.0.47
10.10.0.1
who-has 10.10.0.1
tell 10.10.0.47
10.10.0.2
DNS Server


10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
10.10.0.1 is-at
0e:aa:49:af:18:15
10.10.0.1 is-at
0e:aa:49:af:18:15
10.10.0.1
10.10.0.2
DNS Server


10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
TCP (HTTP HEAD)
10.10.0.1
10.10.0.2
DNS Server


10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
TCP (HTTP 200 OK)
10.10.0.1
10.10.0.2
DNS Server


10.10.0.47
AS14618
aws
AS36646
yahoo!
2001:4998:44:3507::8000
who-has
?
is at
AAAA? www.yahoo.com.
TCP (HTTP HEAD)
who-has?
is at
TCP (HTTP 200 OK)
10.10.0.1
who-has
?
is at
CNAME new-fp-shed.wg1.b.yahoo.com.,
AAAA 2001:4998:44:3507::8000
arp
IP/UDP
IPv6/TCP
10.10.0.2
DNS Server

TCP/IP Basics: Protocol Layers

Even simple examples cross multiple layers and protocols:

4. Hypertext Transfer Protocol (RFC 2616)
   Domain Name System (various RFCs)
3. Transmission Control Protocol (RFC 793, `tcp(4)`)
   User Datagram Protocol (RFC 768; `udp(4)`)
2. Internet Protocol (RFC 791; `ip(4)`)
   Internet Protocol, Version 6 (RFC 8200; `ip6(4)`)
1. Address Resolution Protocol (RFC 826; `arp(4)`)

	Layer	Function
4	Application Layer	End-User Application Programs
3	Transport Layer	Delivery of data to applications
2	Network Layer	Basic communication, addressing, and routing
1	Link Layer Physical Layer	Network Hardware and device drivers Cable or physical

Exercises

Inspect our tcpdump output in detail. You should notice (at least) two things:

- We observe ARP requests from/to the default router *before* we talk to our DNS server — why is that?
- Look at the ARP replies from the DNS server and the default router. What can you deduce about the layer 2 network our instance is on from them?

Coming up: a few more protocol examples