DNS Security and Resiliency

WACREN, DNS/DNSSEC Regional Workshop

Ouagadougou, 10-14 October 2016

Threats to DNS

- Server crashes
- Server compromise
- Denial of service attacks
- Amplification attacks
- Cache poisoning
- Targeted host attacks using zone information
- Information exposure
- Etc..

DNS security?

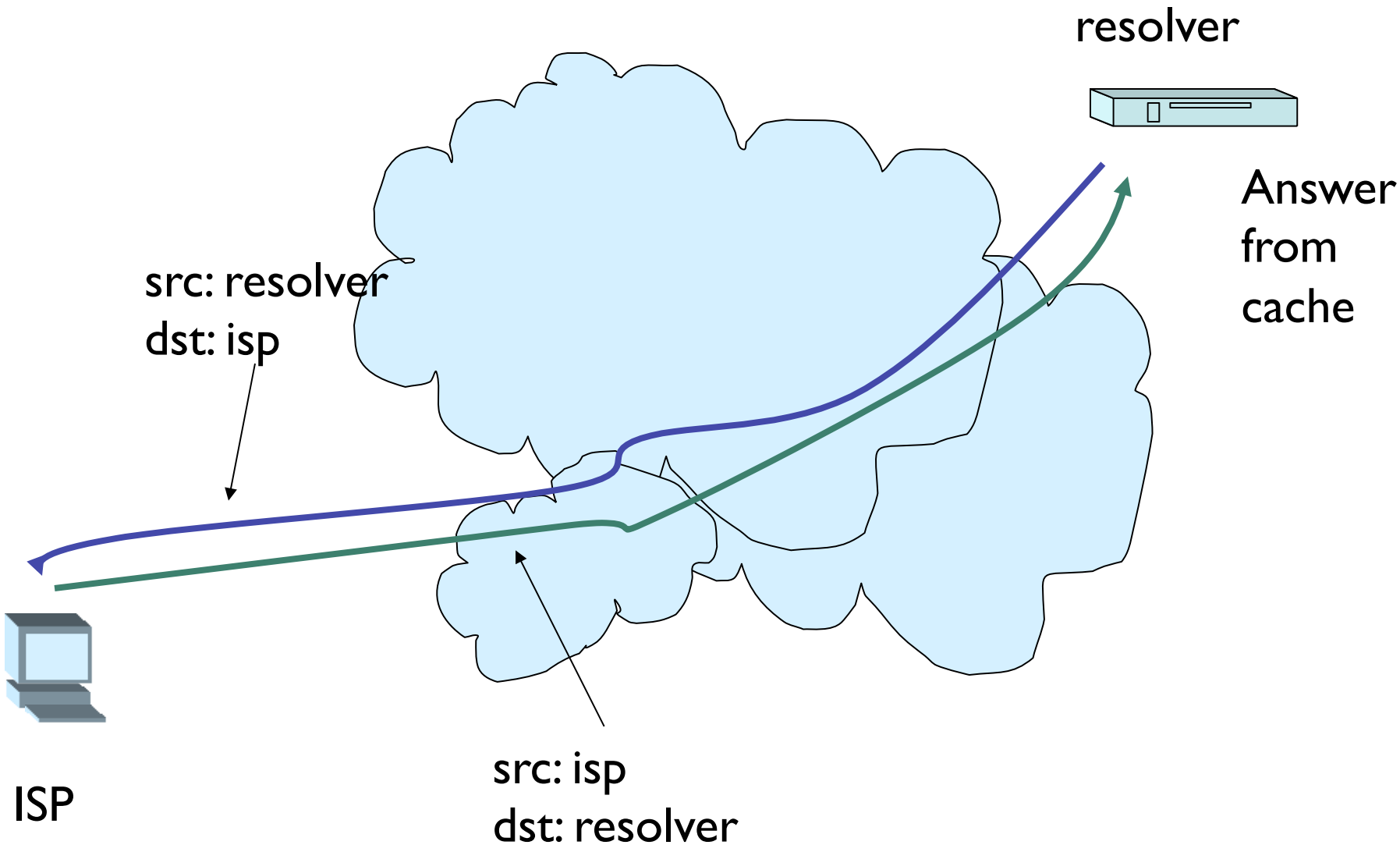
- There is more you need to think of when ‘securing’ your DNS services
  - Host security
  - Network security
  - Registry system security

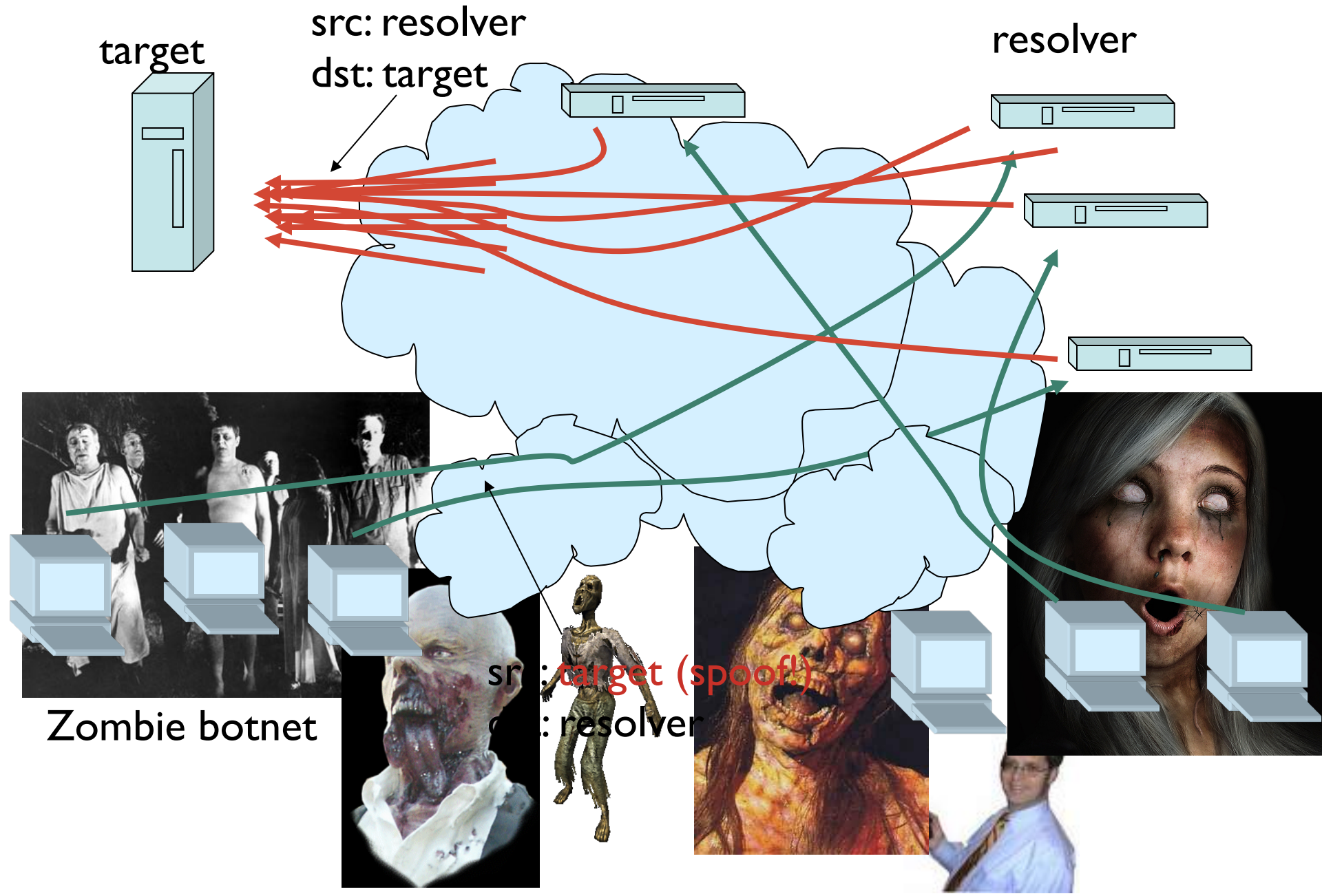
Network security

- Shield your registry
- Shield your nameservers
  - Port 53 (dns)
  - Port 22 (ssh)
  - Port 953 (rndc)
  - More ???

DDOS and the DNS

- Reflector attacks
  - Recently Open recursive servers used to amplify traffic
  - several Gbits/second traffic to critical infrastructure
  - Source addresses at DDOS target are valid, packet format valid

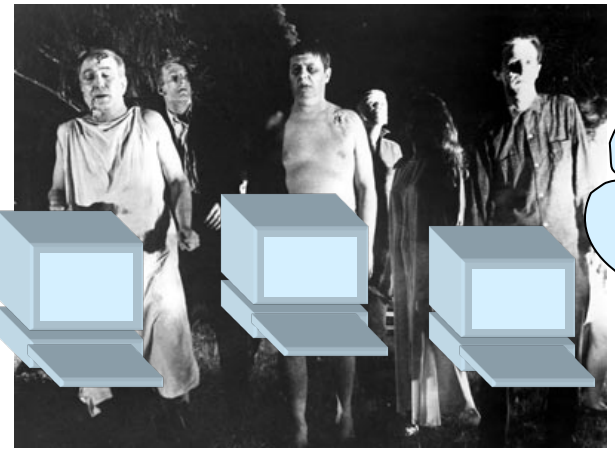

resolver
Answer
from
cache
src: resolver
dst: isp
ISP
src: isp
dst: resolver


target
src: resolver
dst: target
resolver
Zombie botnet
src: target (spoof!)
dst: resolver

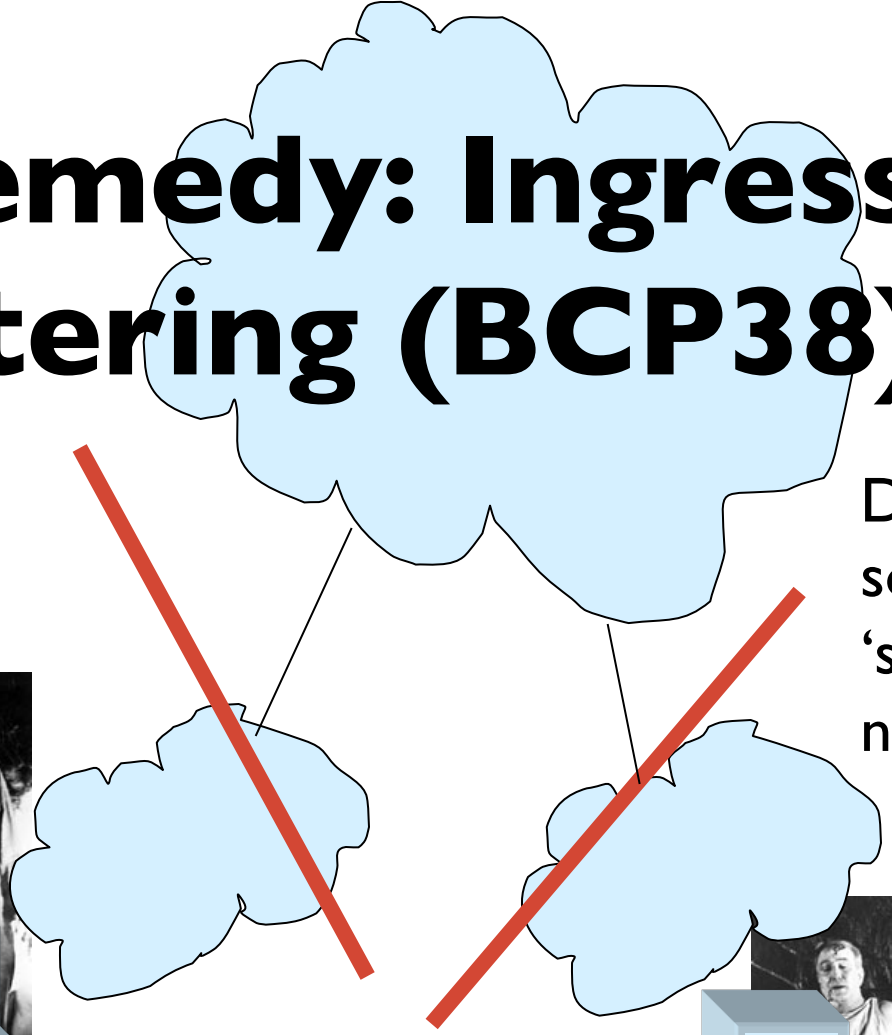
an UDP problem

- DNS has nice amplification characteristics
- ‘Closing open resolvers’ helps, but authoritative servers will do too
- You make the packets smaller? We’ll just wake up more zombies

Remedy: Ingress filtering (BCP38)



Zombie botnet



Drop packets if source address is ‘strange’ to the network

Repeat: BCP38

- http://www.**ietf.org**/rfc/rfc2827.txt(**BCP38)**

**"Network Ingress Filtering:
Defeating Denial of Service Attacks which employ IP Source Address Spoofing"**

- Deploy on your own networks
  - Act responsibly in the Public Space
- Require deployment by others
  - Part of procurement procedures

Host Security

- For all your name servers
  - Latest OS
  - CERT
  - Update regularly
  - Use tools like tripwire
  - Read your security logs

Other people security

- Some security mechanisms may cause you problems
- EDNS0: allows > 512 bytes packets
  - Needed for DNSSEC and IPv6
- Larger packets lead to UDP fragments
- UDP fragments are often blocked by firewalls.
  - And then those firewalls also block TCP
  - Or you block TCP (BAD)

Dangers of zone transfers

- Zone transfers meant to be used to distribute zones among authoritative servers
- Transfers are expensive operations in terms of resources
  - Could be used for DoS attack
- Having your whole zone makes hacker's life easier:
  - No need to scan your address space
  - Better understanding of your network

Authoritative vs. Recursive

Server Function	Information	Target audience
Authoritative	Your domains	The Internet
Recursive	All other domains	Your users

Separation of Duties

- Physically separating authoritative and recursive servers gives you:
  - Easier control
    - •Apply restrictions to what the servers can be used for, and by whom
  - Easier troubleshooting
    - • Consider what happens when a DNS-hosted customer moves their domain to another provider without telling you.

Authoritative – BIND options

```
options {
   version "9999.9.9”;
   allow-transfer { peers; };
   blackhole { attackers; };
   recursion no;
   allow-query { any; };
   …};
```

Authoritative – IP filters

- Can't really filter much here
  - Ports udp/53 and tcp/53 should be open to the world.
- Just don't run any other services
  - No web server, mail server, etc.
  - Keep it really simple

Authoritative - Location

- Locate your servers topologically and geographically dispersed
  - Establish a relationship with another operator, or
  - There are companies that provide secondary service
  - Ask for anycast, DNSSEC and IPv6 support!
  - See RFC 2182

Recursive – BIND options

```
options {
   version "9999.9.9”;
   recursive-clients 5000;
   allow-transfer { none; };
   blackhole { attackers; };
   allow-recursion { customers; };
   allow-query { customers; };
   dnssec-enable yes;
   dnssec-validation yes;
   …};
```

Recursive – IP filters

- udp/53 and tcp/53 open only to customers
  - Drop the packets early, don't bother the DNS daemon
  - Remember to filter IPv6 as well if you have v6 connectivity
  - Can be done simply with
    - iptables on Linux.
    - ipfw on FreeBSD

DNSSEC Validation

- The root is signed!
- Lot of names are signed (TLDS and others)
- Only true way to avoid cache poisoning
- Started with universities and research organizations, now large ISPs are joining
- Trust Anchor
  - https://data.iana.org/root-anchors/
  - Root KSK rollover process in progress

DNSSEC Validation

```
 options {
    dnssec-enable yes;
    dnssec-validation yes;
}


managed-keys {
"." initial-key 257 3 8 "AwEAAagAIKlVZrpC6Ia7gEzahOR
+9W29euxhJhVVLOyQbSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/
RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/
QZxkjf5/
Efucp2gaDX6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpzW5hO
A2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relSQageu
+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=";
};
```

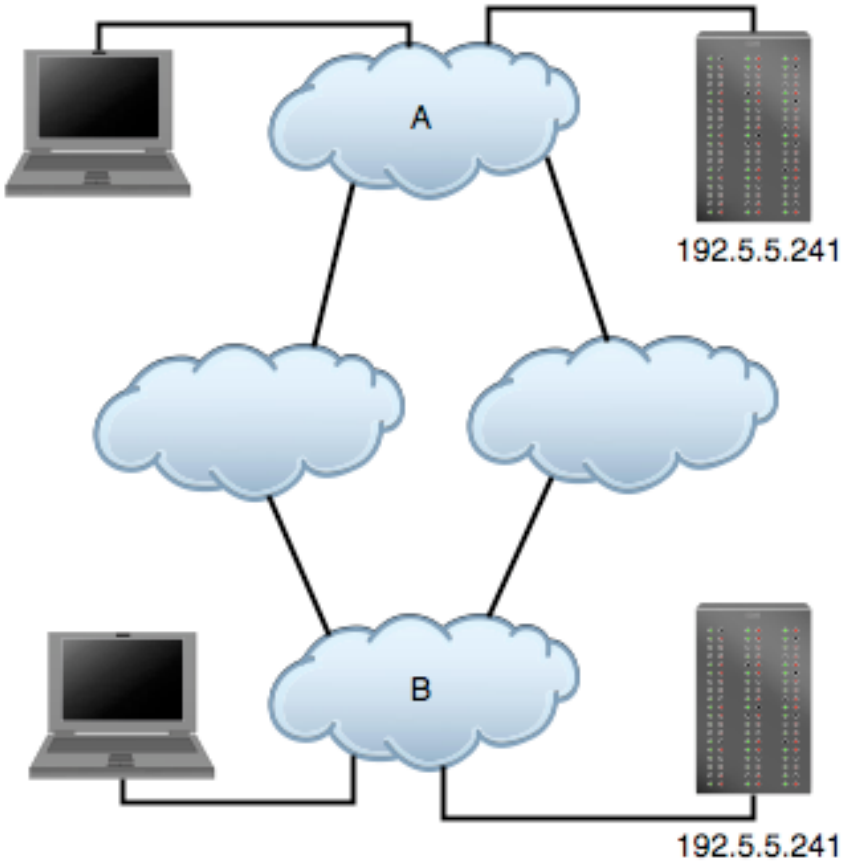
Client failover behavior

- Clients of authoritative servers (other recursive servers)
  - Fail over well using different NS records
- Clients of recursive servers (stub resolvers)
  - Do a very poor job at failing over
  - Users complain immediately
  - Services break due to timeouts

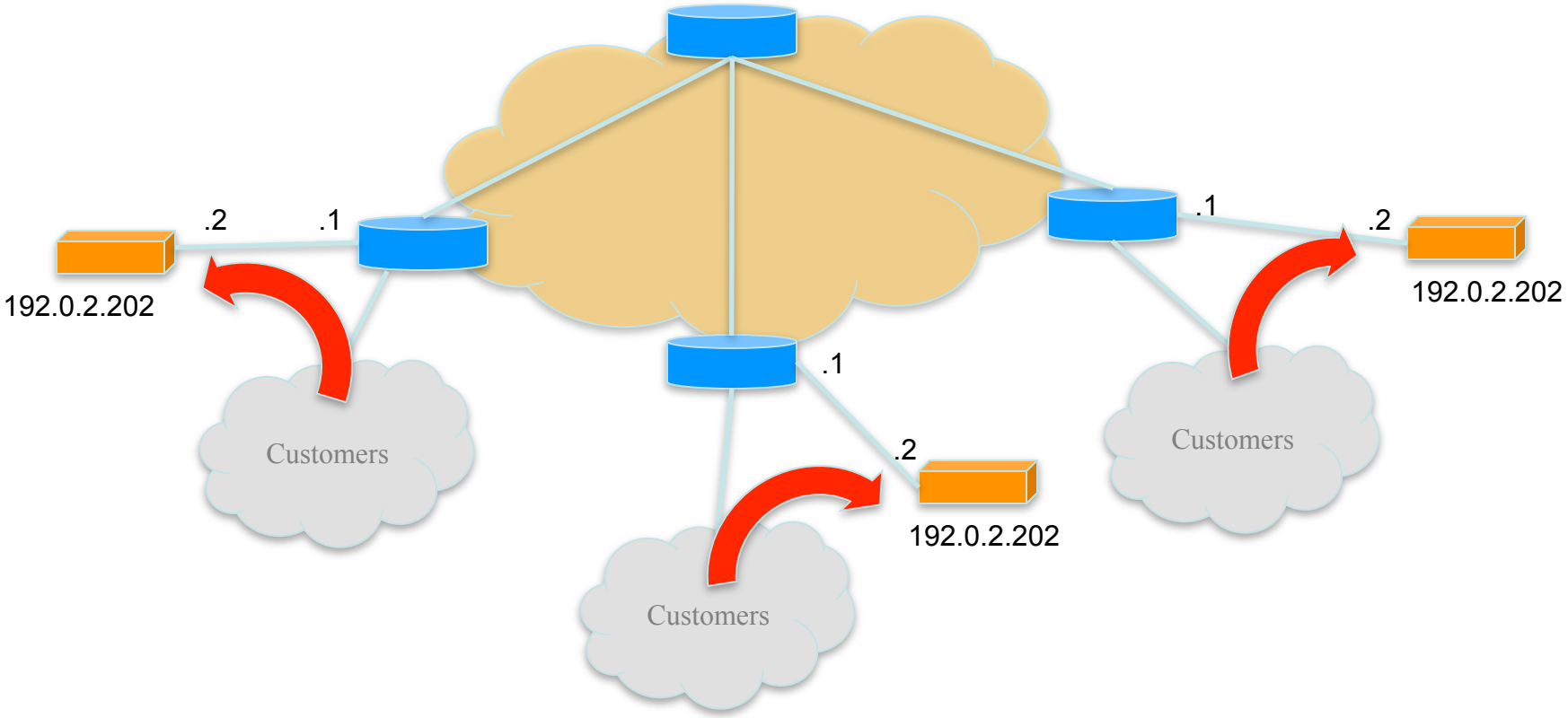
Anycast

- Routing trick in which the same IP address is announced by multiple routers so that a particular sender reaches the topologically nearest node that responds to that address
- Excellent solution to enhance DNS:
  - Load-balancing
  - Failover
  - DoS attack isolation
  - Cache poisoning isolation

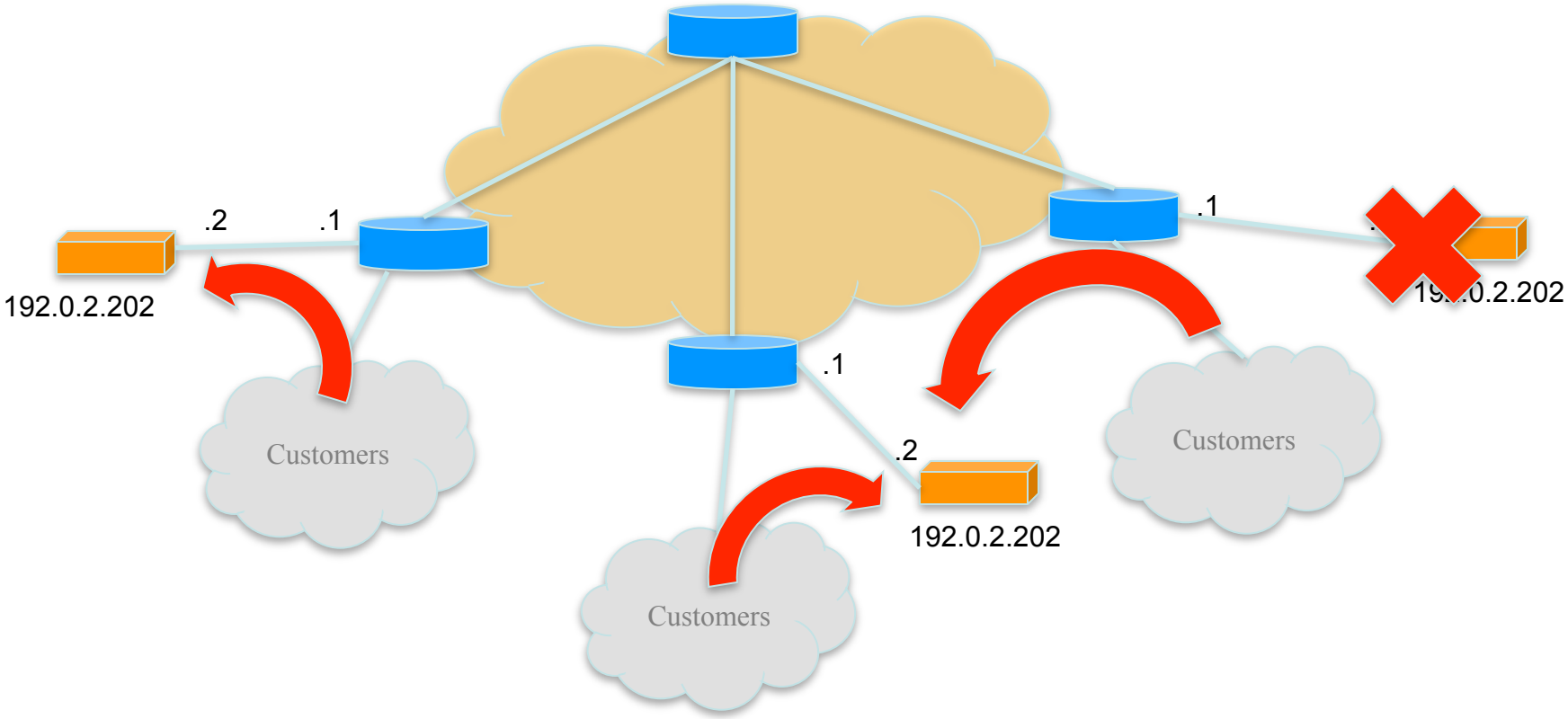
Anycast routing



Anycast Topology



Anycast Topology



Before a local F-root

```
[halibut:~]$ traceroute f.root-servers.net
traceroute to f.root-servers.net (192.5.5.241), 64 hops max, 40 byte packets
 1  router.cctld.or.ke (196.216.0.62)  1.945 ms  7.147 ms  1.165 ms
 2  196.216.66.5 (196.216.66.5)  44.967 ms  23.918 ms  12.420 ms
 3  217.21.112.4.swiftkenya.com (217.21.112.4)  5.141 ms  9.491 ms  5.791 ms
 4  193.220.225.5 (193.220.225.5)  8.919 ms  5.708 ms  5.898 ms
 5  no-nit-tn-7.taide.net (193.219.192.7)  538.820 ms  539.738 ms  550.056 ms
 6  no-nit-tn-5.taide.net (193.219.193.145)  540.073 ms  551.002 ms  536.818 ms
 7  pos5-1.gw3.osl2.alter.net (146.188.39.1)  535.738 ms  536.197 ms  534.790 ms
 8  so-3-0-0.xr2.osl2.alter.net (146.188.15.97)  535.701 ms  542.140 ms  543.969 ms
 9  so-4-2-0.tr1.stk2.alter.net (146.188.15.61)  541.221 ms  545.562 ms  544.435 ms
10  so-7-0-0.ir2.dca4.alter.net (146.188.11.226)  653.929 ms  652.082 ms  649.199 ms
11  so-1-0-0.il2.dca6.alter.net (146.188.13.45)  658.517 ms  652.177 ms  664.978 ms
12  0.so-0-2-0.tl2.sac1.alter.net (152.63.0.190)  887.784 ms  739.093 ms  717.126 ms
13  0.so-1-3-0.xl2.pao1.alter.net (152.63.48.181)  718.044 ms  720.835 ms  727.418 ms
14  pos1-0.xr2.pao1.alter.net (152.63.54.78)  717.283 ms  716.201 ms  714.212 ms
15  188.atm7-0.gw10.pao1.alter.net (152.63.53.21)  778.208 ms  731.906 ms  832.482 ms
16  isc-pao-gw.customer.alter.net (157.130.205.230)  717.801 ms  712.912 ms  712.718 m
17  f.root-servers.net (192.5.5.241)  743.804 ms  721.633 ms  746.818 ms
[halibut:~]$
```

After...

```
[halibut:~]$ traceroute f.root-servers.net
traceroute to f.root-servers.net (199.6.6.14), 64 hops max, 40 byte
packets
 1  router.cctld.or.ke (196.216.0.62)  244.241 ms  1.159 ms  1.099 ms
 2  196.216.66.5 (196.216.66.5)  8.678 ms  4.942 ms  31.862 ms
 3  80.240.202.54.swiftkenya.com (80.240.202.54)  22.455 ms  15.803
ms  14.864 ms
 4  198.32.143.125 (198.32.143.125)  40.770 ms  7.192 ms  7.786 ms
 5  f.root-servers.net (192.5.5.241)  10.906 ms  10.894 ms *
[halibut:~]$
```

Diversify OS and DNS software

- Consider running different DNS software (Bind, Unbound, NSD, etc.) on different OSs
  - Saves you from total disaster when you hit a bug, but…
  - Makes configuration management a bit more challenging

Periodic zone checks

- Periodically run checks for
  - Inconsistent, missing or bad data
  - Catching common misconfigurations
  - RFC 1912
- Check out dnscheck
  - https://github.com/dotse/dnscheck

Watch those logs

- Use a tool to analyze your DNS logs and alarm on important messages
  - Swatch, Tenshi, etc.
  - Look for:
    - Zone syntax errors
    - Transfer problems
    - DNSSEC validation errors
    - etc

Monitoring Availability – Nagios

- Use *check_dns* to make sure that the server is actually resolving
  - Don't just ping the server
- You can also use this to make sure that very important A records are there:
  - www, smtp, imap,…
- Make sure that your alarms will work despite DNS being down!

Monitoring Availability - Nagios

Service 'DNS' On Host 'ns1'

01-01-2010 00:00:00 to 11-07-2010 21:08:40
Duration: 310d 21h 8m 40s

[Availability report completed in 0 min 16 sec]

Service State Breakdowns:

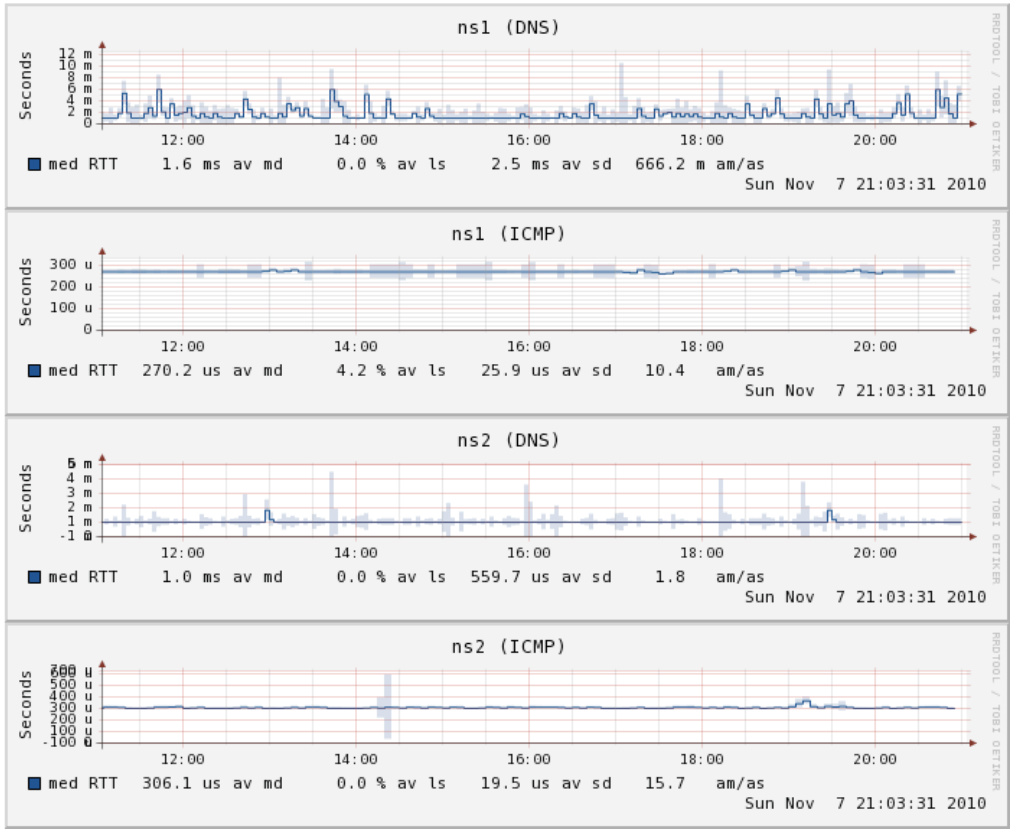
State	Type / Reason	Time	% Total Time	% Known Time
OK	Unscheduled	90d 22h 8m 40s	29.247%	100.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	**90d 22h 8m 40s**	**29.247%**	**100.000%**
WARNING	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	**0d 0h 0m 0s**	**0.000%**	**0.000%**
UNKNOWN	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	**0d 0h 0m 0s**	**0.000%**	**0.000%**
CRITICAL	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	**0d 0h 0m 0s**	**0.000%**	**0.000%**
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	219d 23h 0m 0s	70.753%	
	Total	219d 23h 0m 0s	70.753%	
All	Total	310d 21h 8m 40s	100.000%	100.000%

Monitoring Delay

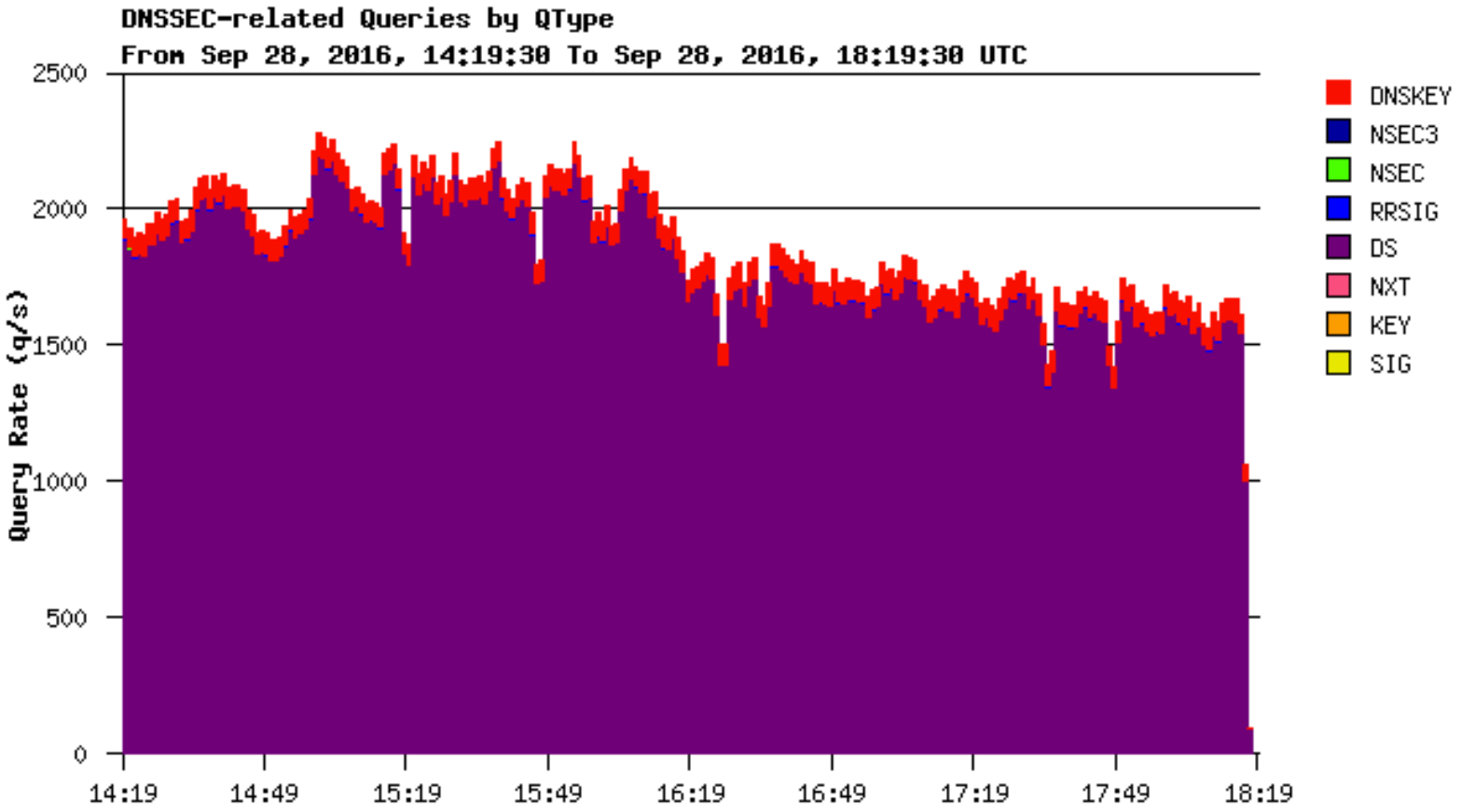
- Important to look at both
  - Network delay
  - DNS service delay

Monitoring Delay - Smokeping

Recursive



Query Statistics - DSC



Questions?