euronog
european network operators' group

Building highly resilient services using IP Anycast

Łukasz Bromirski
lbromirski@cisco.com

Agenda

- IP Anycast – what is this?
- Exploiting anycast (for fun)
- DNS considerations
- HTTP considerations
- Q&A

IP Anycast

IP Anycast history

It's not new, but still it's good ☺

- Discovery of the services – SNTPv4 (RFC 2030) and NTP "manycast" (RFC 4330)
- Original IPv6 transition mechanism (RFC 2893) and then 6to4 (RFC 3068)
- Multicast RP use of anycast – MSDP and PIM (RFC 4610)
- For IPv6, first RFCs (RFC 1884, 2373 and 3513) were forbidding to use anycast address as a source address; it was taken care of in the RFC 4291
- DNS use of anycast (RFC 3258) mentioned the "shared unicast"

IP Anycast

- Simple, efficient, widely deployed ☺
- L3 agnostic

  IPv4 and IPv6 both work

  there's a 'anycast' in the RFC specifications, but most of the scaling techniques use a 'shared unicast' anycast approach

IP Anycast

- It is universal, as the method can be combined or used in place of:

  load-balancing at L3/L4

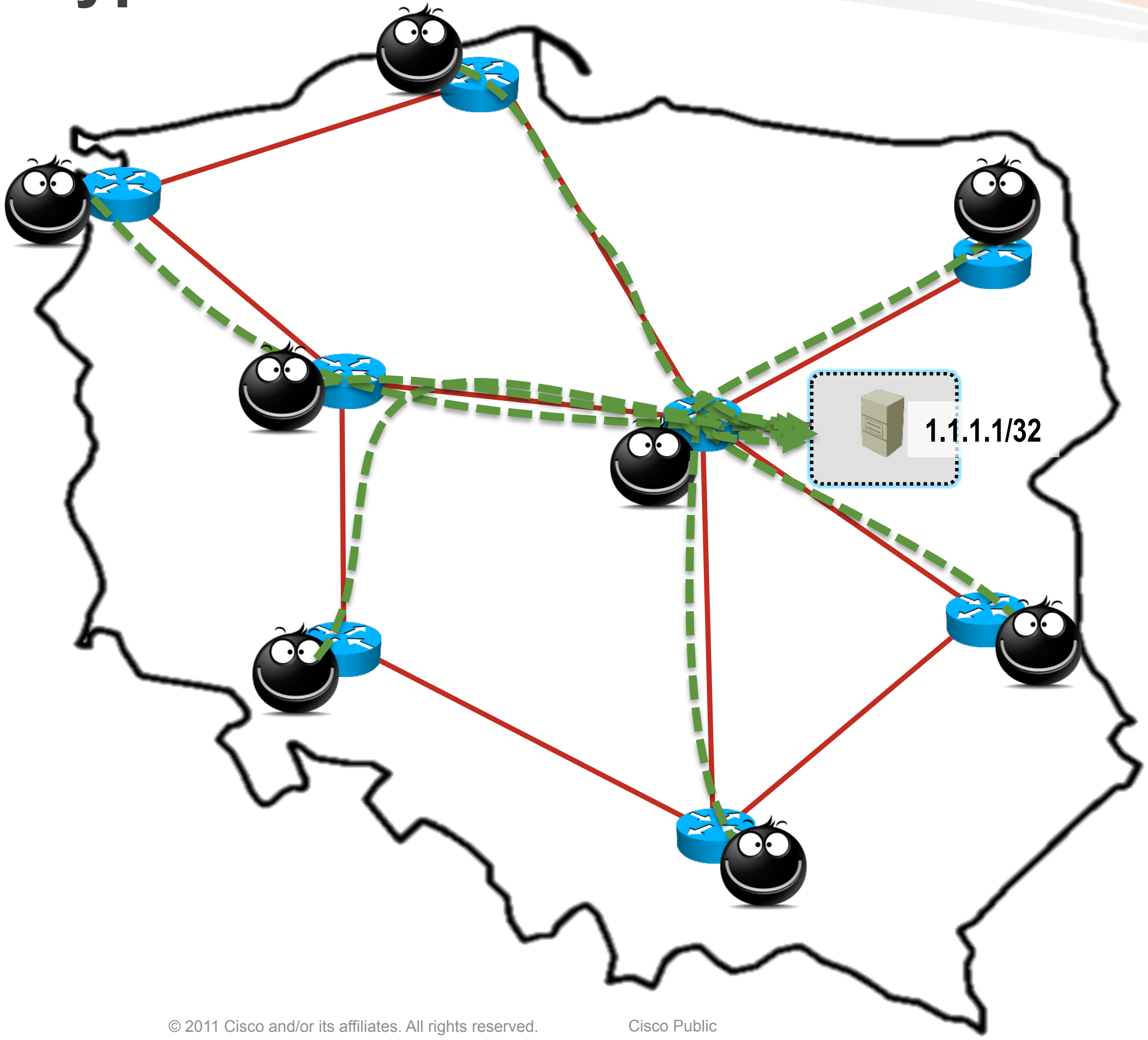
  adding new IP servers in the usual way

  contrary to popular belief, IP Anycast can also be used to scale TCP services, not only UDP ones*

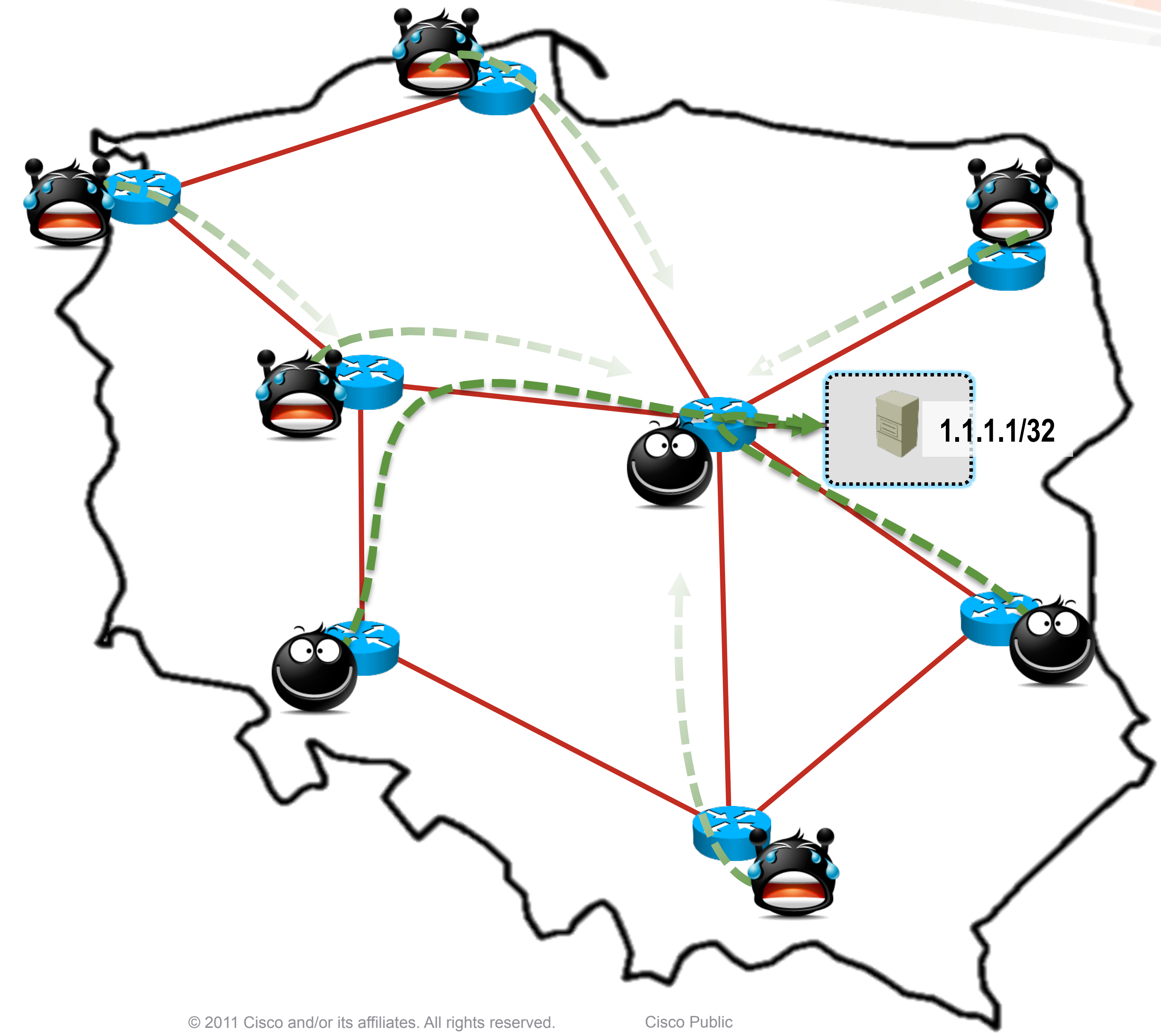
* more on this later, and please remember YMMV

Exploiting IP anycast (for fun)

The typical use scenario

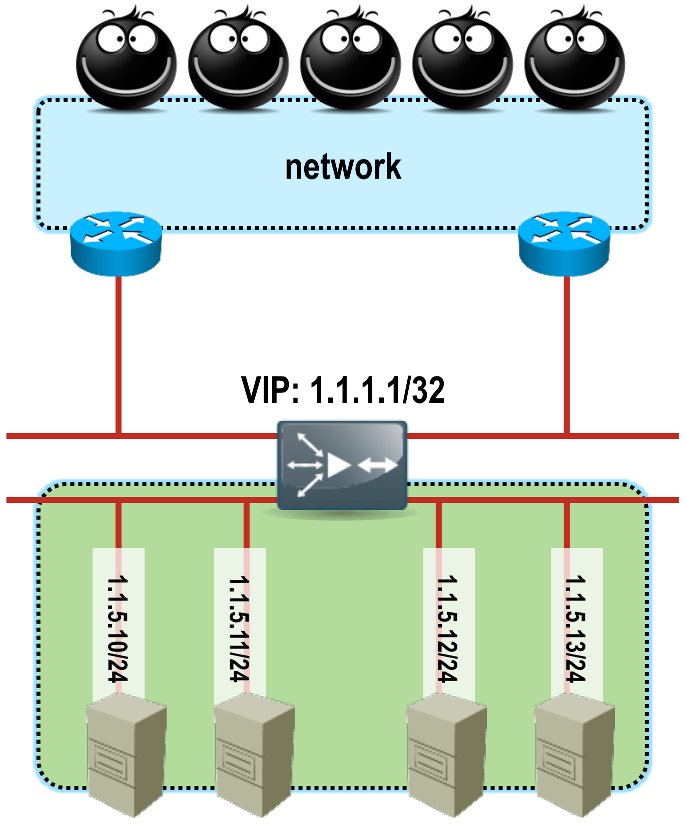


Server becomes overloaded

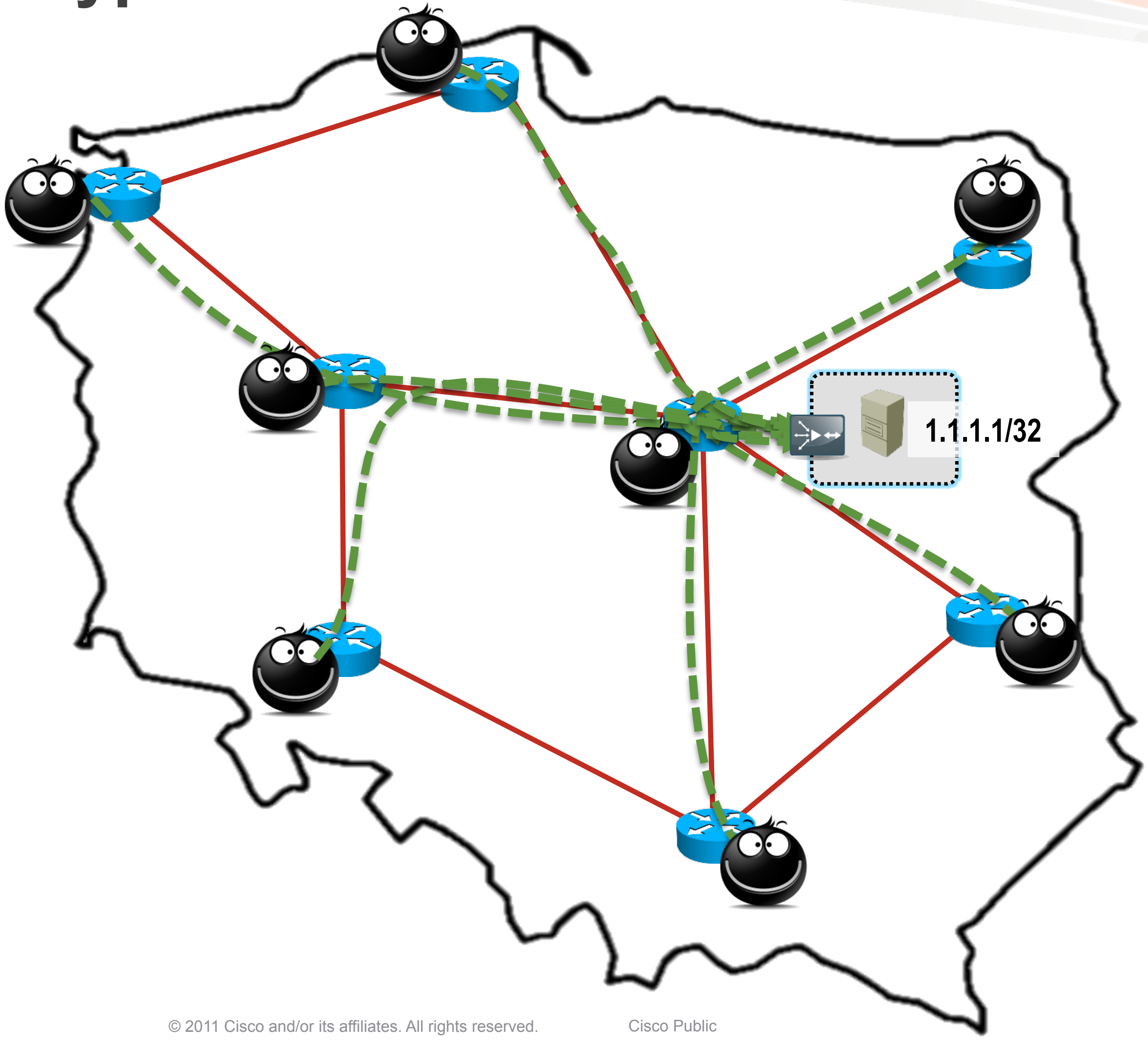


First solution: grow horizontally

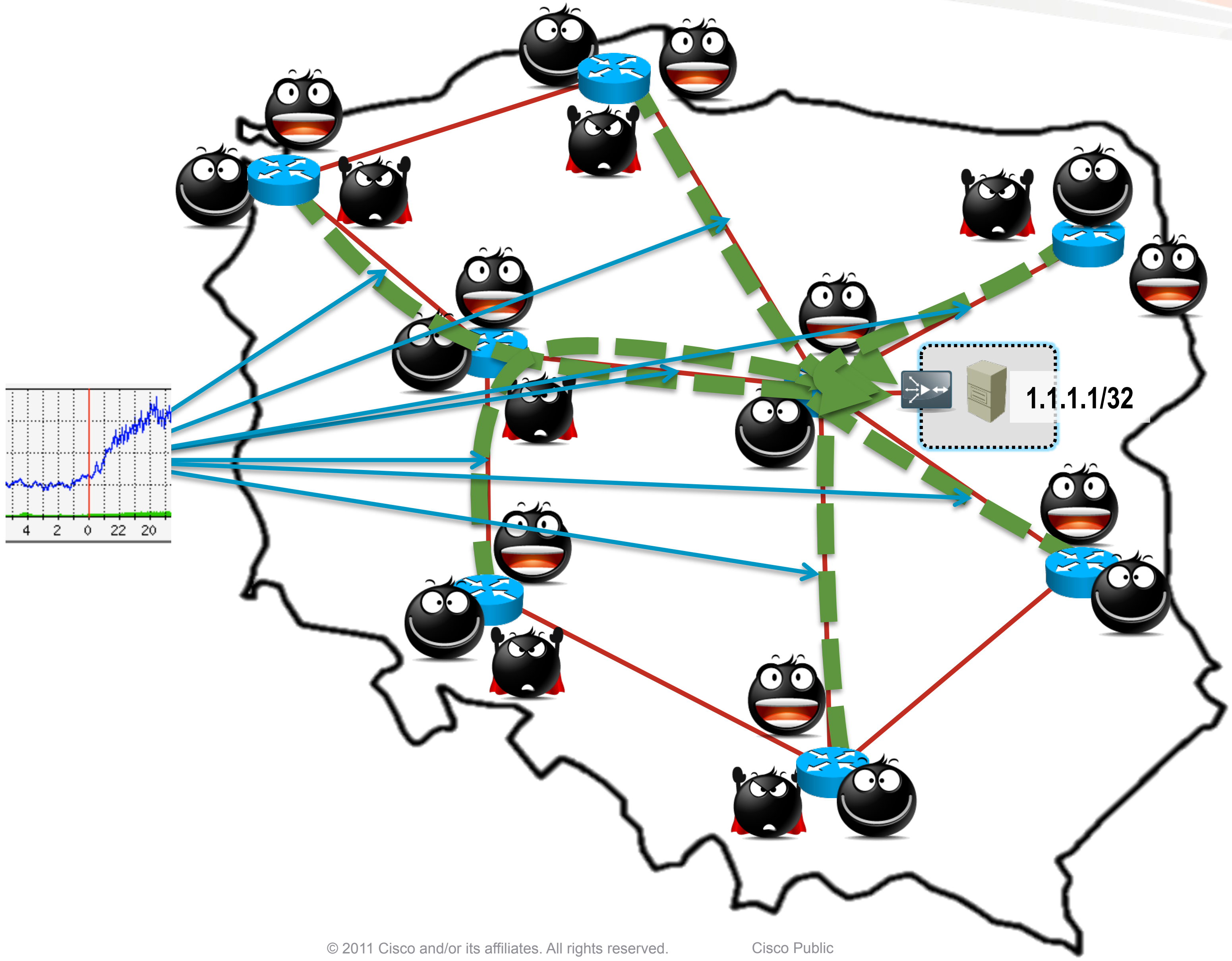
"Install a load balancer and add servers"



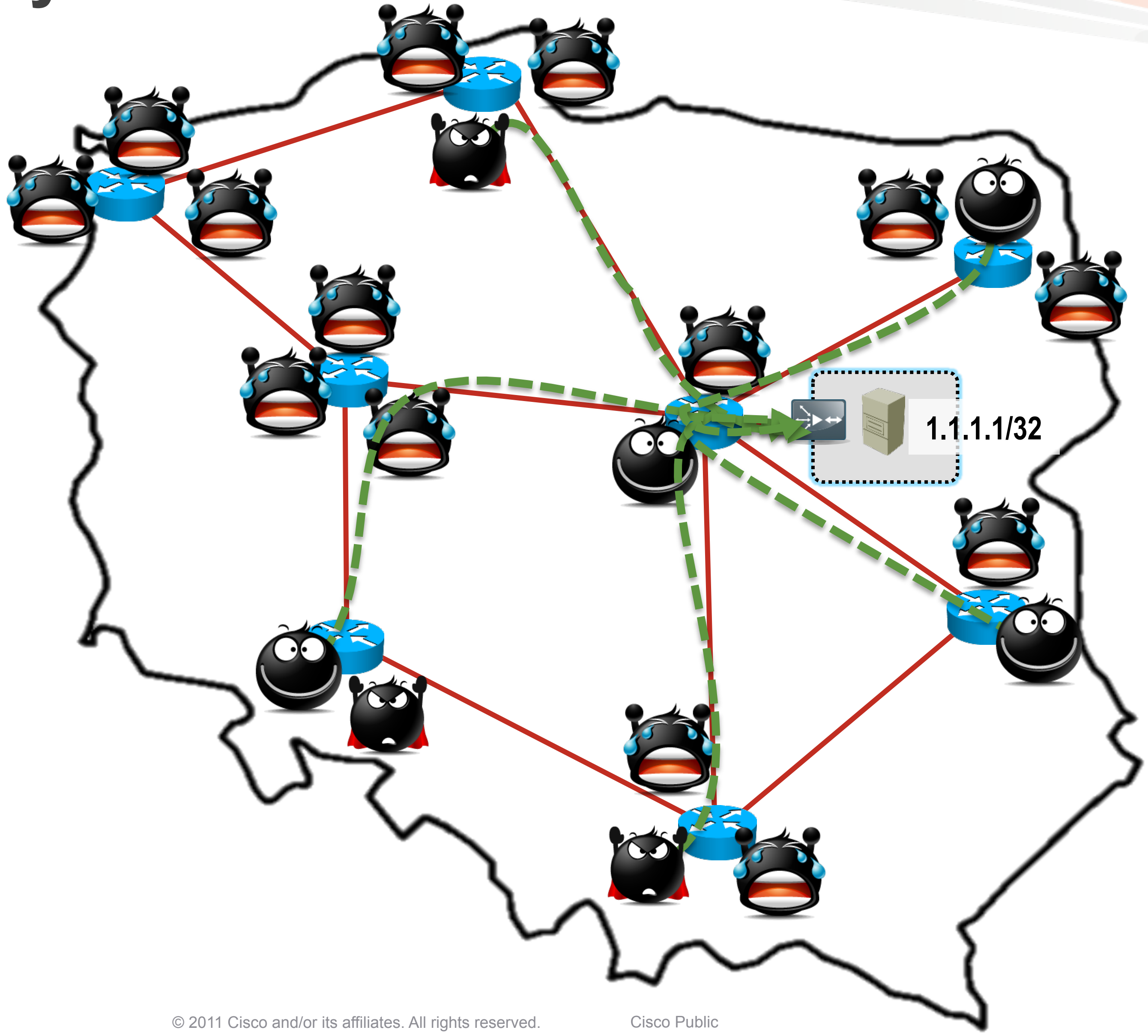
The typical use scenario with LB



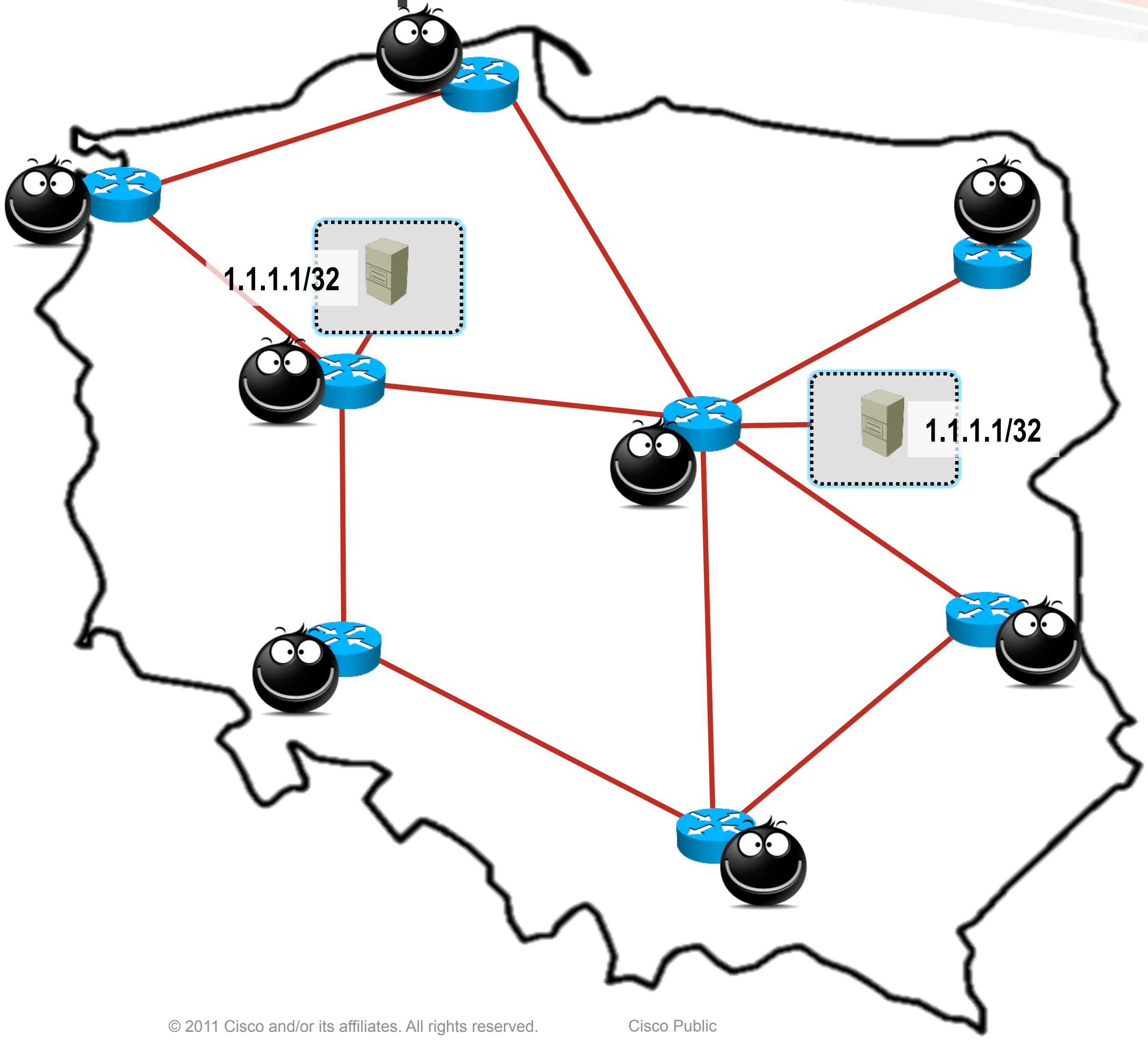
The links are overloaded



Only some users will receive service…



What if we'd replicate the IP?



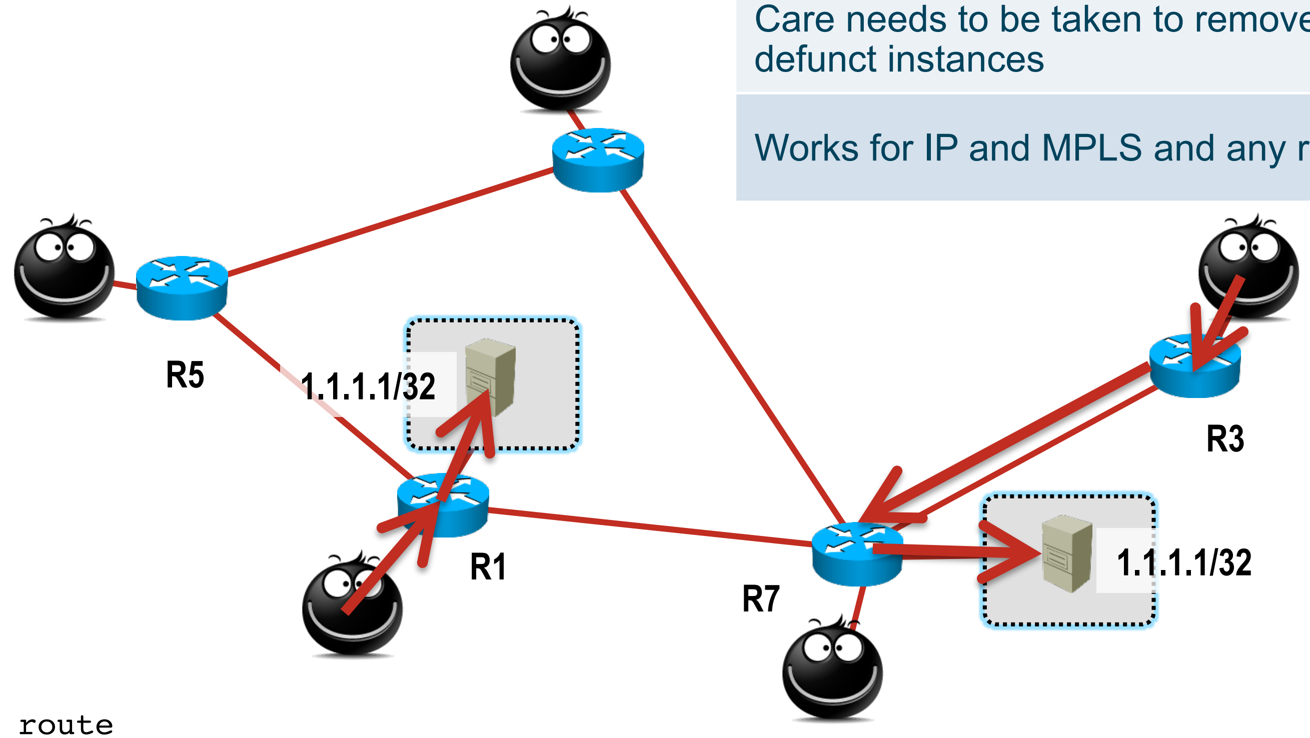
Replicated service

Routing will choose the least cost path

Each host will be directed towards the nearest service instance

Care needs to be taken to remove from routing table defunct instances

Works for IP and MPLS and any routing protocol



```
R1> show ip route
I    1.1.1.1/32 [115/10] via 192.168.11.6
```

```
R3> show ip route
I    1.1.1.1/32 [115/10] via 192.168.73.7
```

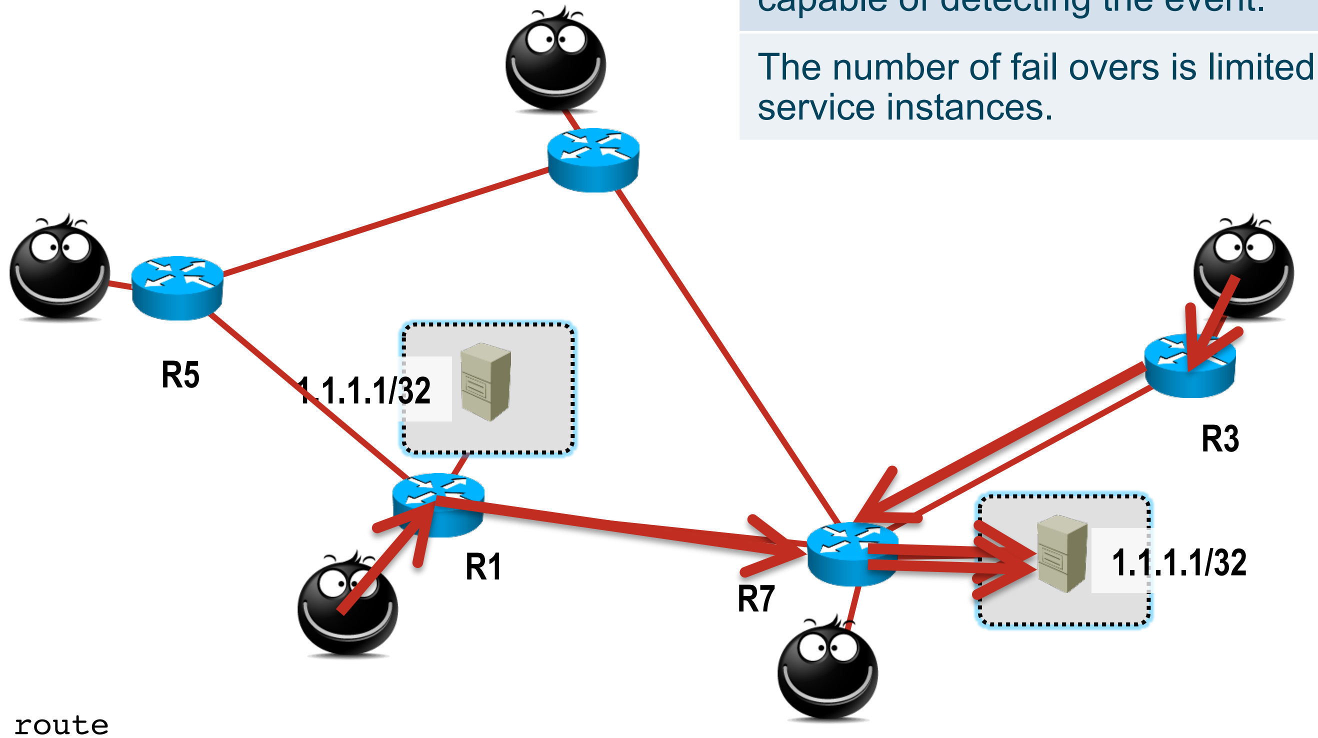
```
R7> show ip route
I    1.1.1.1/32 [115/10] via 192.168.77.6
```

Replicated service

Replication will provide HA for instance

Death of service will be propagated by IGP, if IGP is capable of detecting the event.

The number of fail overs is limited to the number of service instances.



```
R1> show ip route
I    1.1.1.1/32 [115/10] via 192.168.17.7

R7> show ip route
I    1.1.1.1/32 [115/10] via 192.168.77.6
```

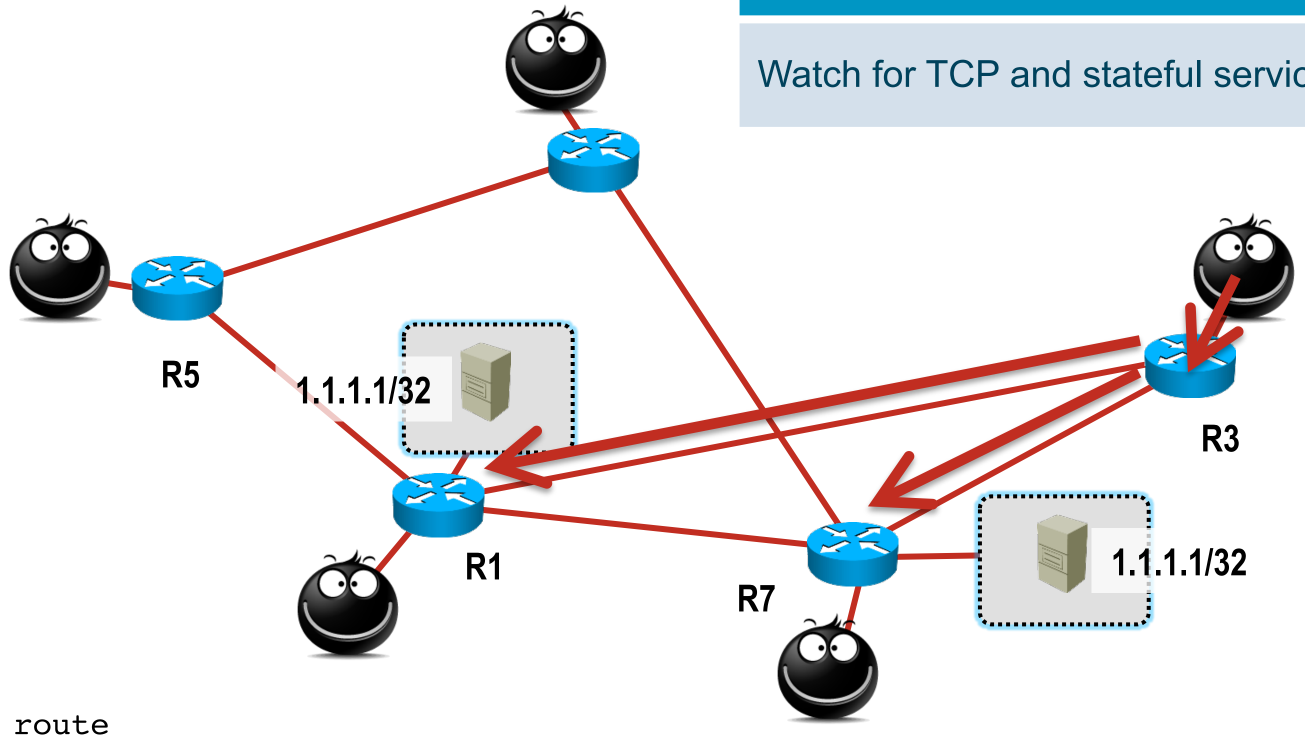
```
R3> show ip route
I    1.1.1.1/32 [115/10] via 192.168.73.7

R7> show ip route
I    1.1.1.1/32 [115/10] via 192.168.77.6
```

Replicated service

Access load balanced at L3/L4

Watch for TCP and stateful services in general*



```
R3> show ip route
I    1.1.1.1/32 [115/10] via 192.168.13.1
                         via 192.168.37.7
```

IP Anycast for DNS

Problems with traditional setup

- The lack of following of best practices:
  - DNS servers are usually installed in one place (from network topology point of view), which makes them an easy target and increases the risk of the surrounding infrastructure
    - hitting the DNS service along with the links to it was never easier
  - Even if DNS servers are in different parts of the network, they attract different traffic – destined to a different IPs
    - you can still DDoS 2-3 destinations easily by just checking what are current NS records of the domain/company and pointing your virtual guns to them

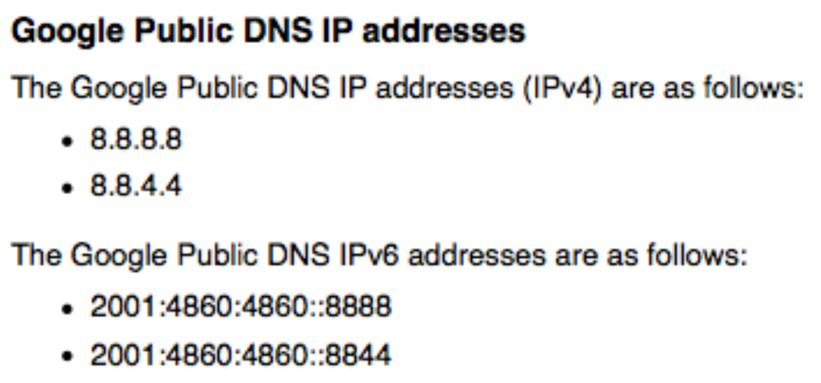
Problems with traditional setup

- Scalability issues:
  - modern dns clients (resolvers) working in the OSes are usually limited to 1-2 DNS servers, and even with that, they tend to use them sequentially
    - you can sometimes add more, usually doesn't make any difference*
    - you may choose to serve different DNS IPs in different parts of your network – possible, but hard to manage
  - you may add the local load-balancing by introducting specialized appliance – software or hardware
    - this brings additional complexity

Who is doing it?

- "Almost" everybody! ☺

OpenDNS®



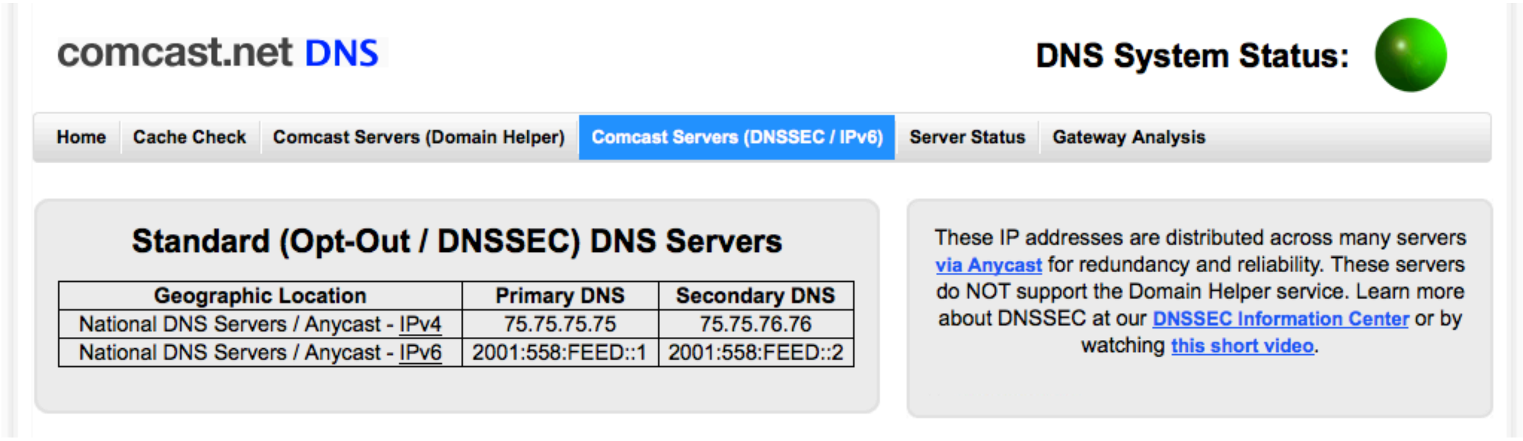
Google Public DNS IP addresses

The Google Public DNS IP addresses (IPv4) are as follows:

- 8.8.8.8
- 8.8.4.4

The Google Public DNS IPv6 addresses are as follows:

- 2001:4860:4860::8888
- 2001:4860:4860::8844



comcast.net DNS

DNS System Status: ●

Home
Cache Check
Comcast Servers (Domain Helper)
Comcast Servers (DNSSEC / IPv6)
Server Status
Gateway Analysis

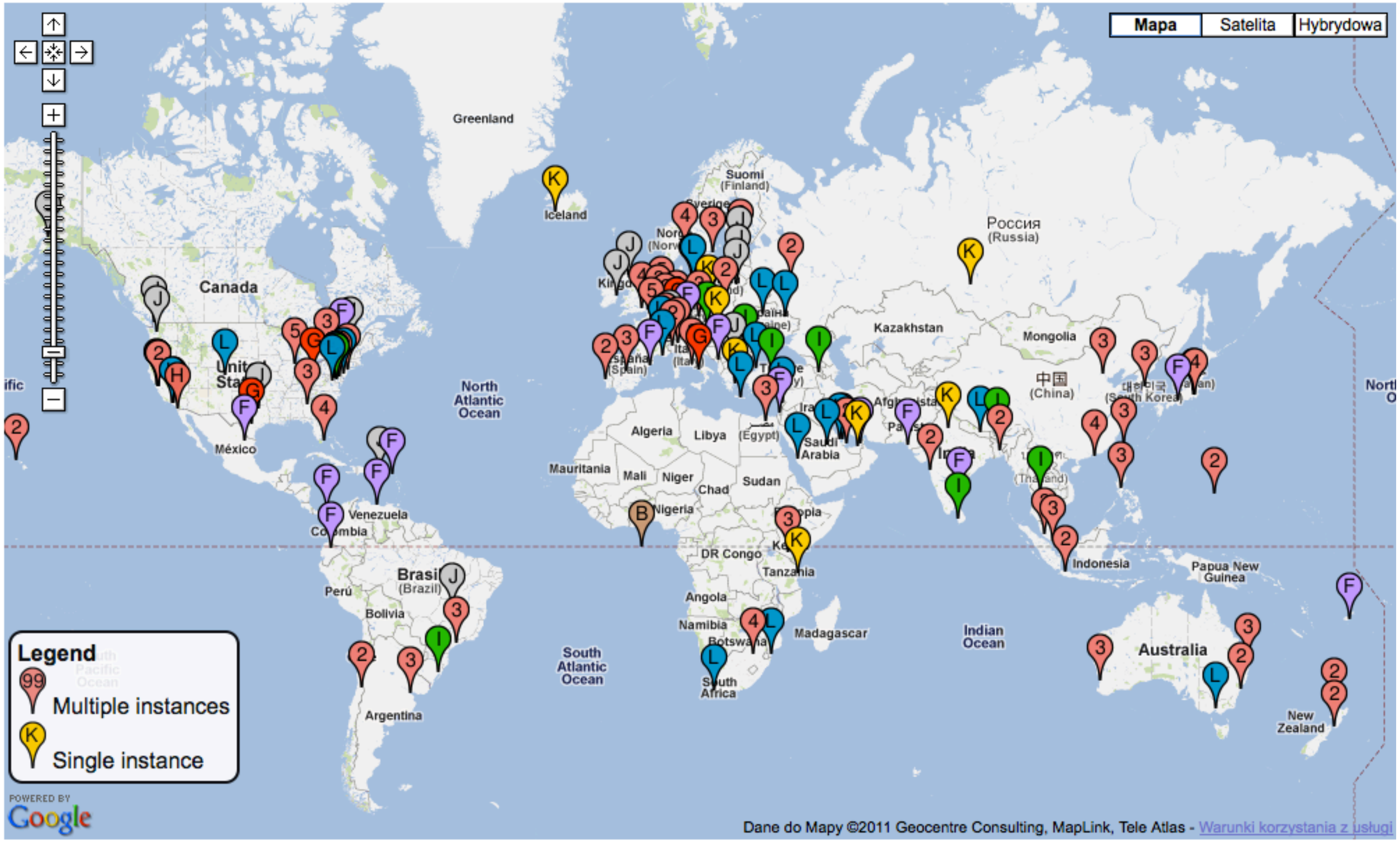
Standard (Opt-Out / DNSSEC) DNS Servers

Geographic Location	Primary DNS	Secondary DNS
National DNS Servers / Anycast - IPv4	75.75.75.75	75.75.76.76
National DNS Servers / Anycast - IPv6	2001:558:FEED::1	2001:558:FEED::2

These IP addresses are distributed across many servers via Anycast for redundancy and reliability. These servers do NOT support the Domain Helper service. Learn more about DNSSEC at our DNSSEC Information Center or by watching this short video.

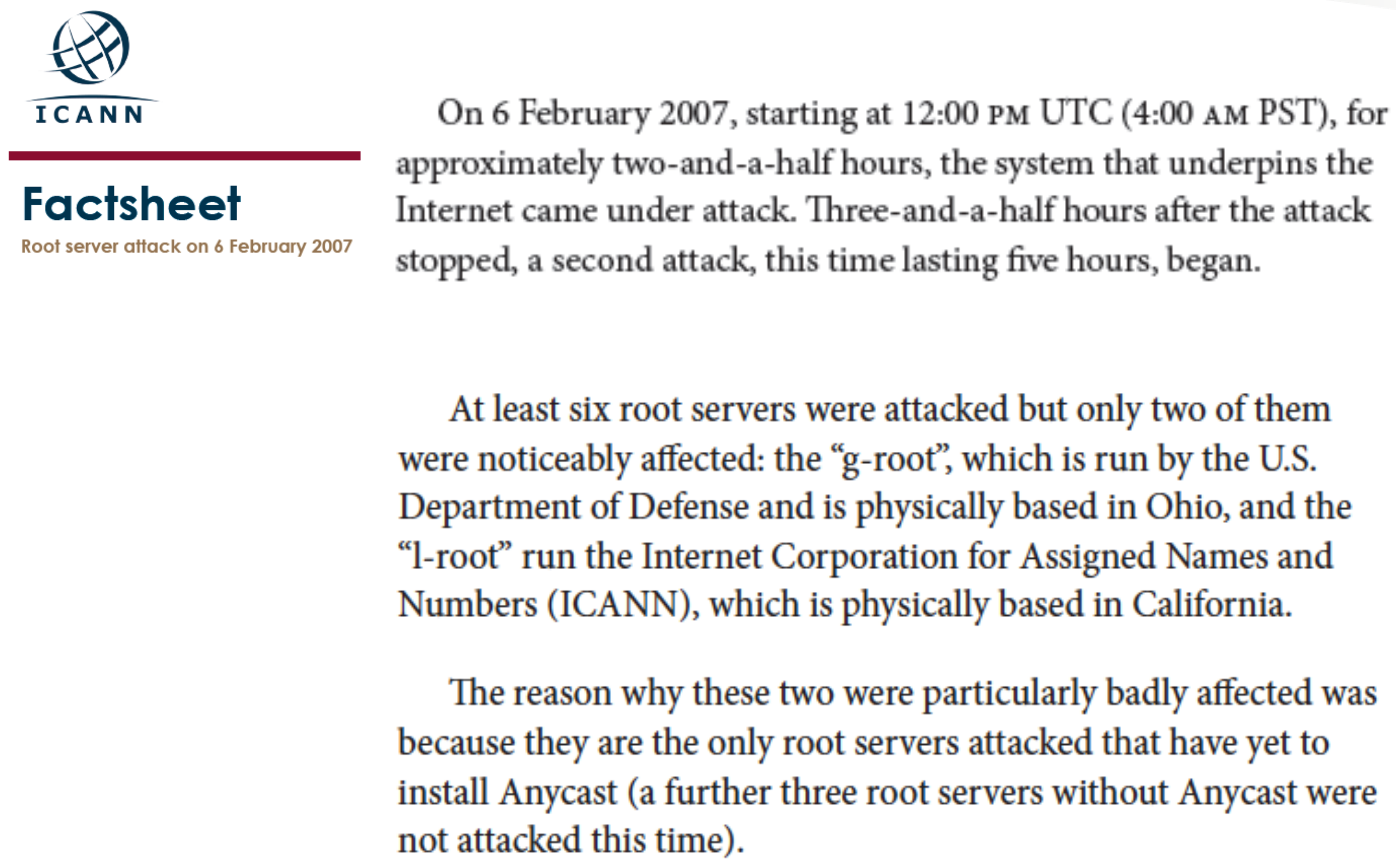
* CAIDA maps for root DNS: http://www.caida.org/research/dns/influence-map/

Root DNS anycast setup map



http://www.root-servers.org/

Root DNS anycast – does it work?



ICANN

Factsheet

Root server attack on 6 February 2007

On 6 February 2007, starting at 12:00 PM UTC (4:00 AM PST), for approximately two-and-a-half hours, the system that underpins the Internet came under attack. Three-and-a-half hours after the attack stopped, a second attack, this time lasting five hours, began.

At least six root servers were attacked but only two of them were noticeably affected: the "g-root", which is run by the U.S. Department of Defense and is physically based in Ohio, and the "l-root" run the Internet Corporation for Assigned Names and Numbers (ICANN), which is physically based in California.

The reason why these two were particularly badly affected was because they are the only root servers attacked that have yet to install Anycast (a further three root servers without Anycast were not attacked this time).

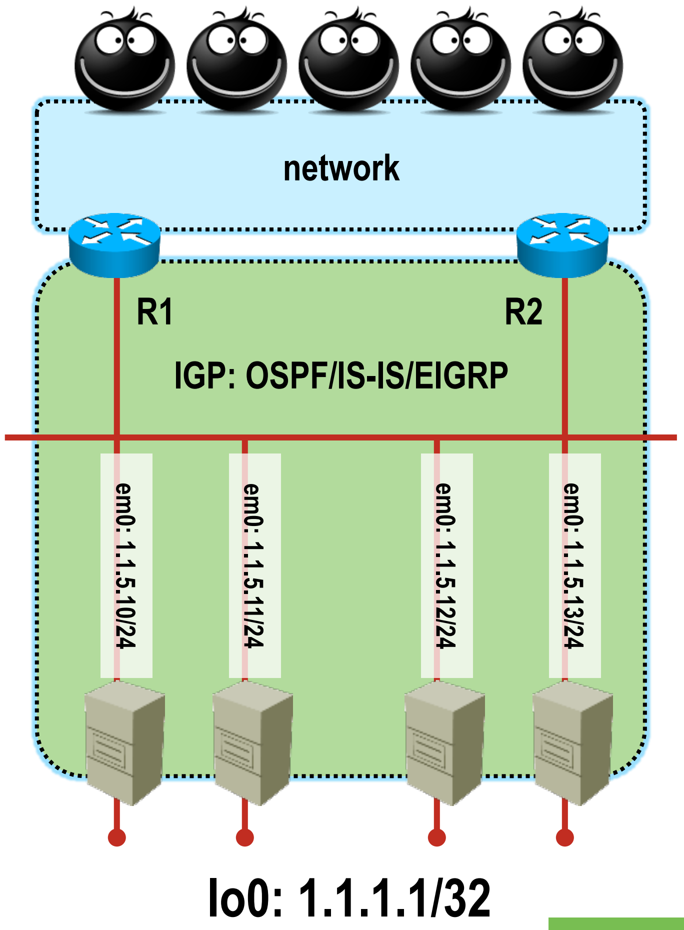
DNS IP anycast solution

In details

R1 route table

```
R1# sh ip route

*> 1.1.1.1/32
      via 1.1.5.10
      via 1.1.5.11
      via 1.1.5.12
      via 1.1.5.13
```



R2 route table

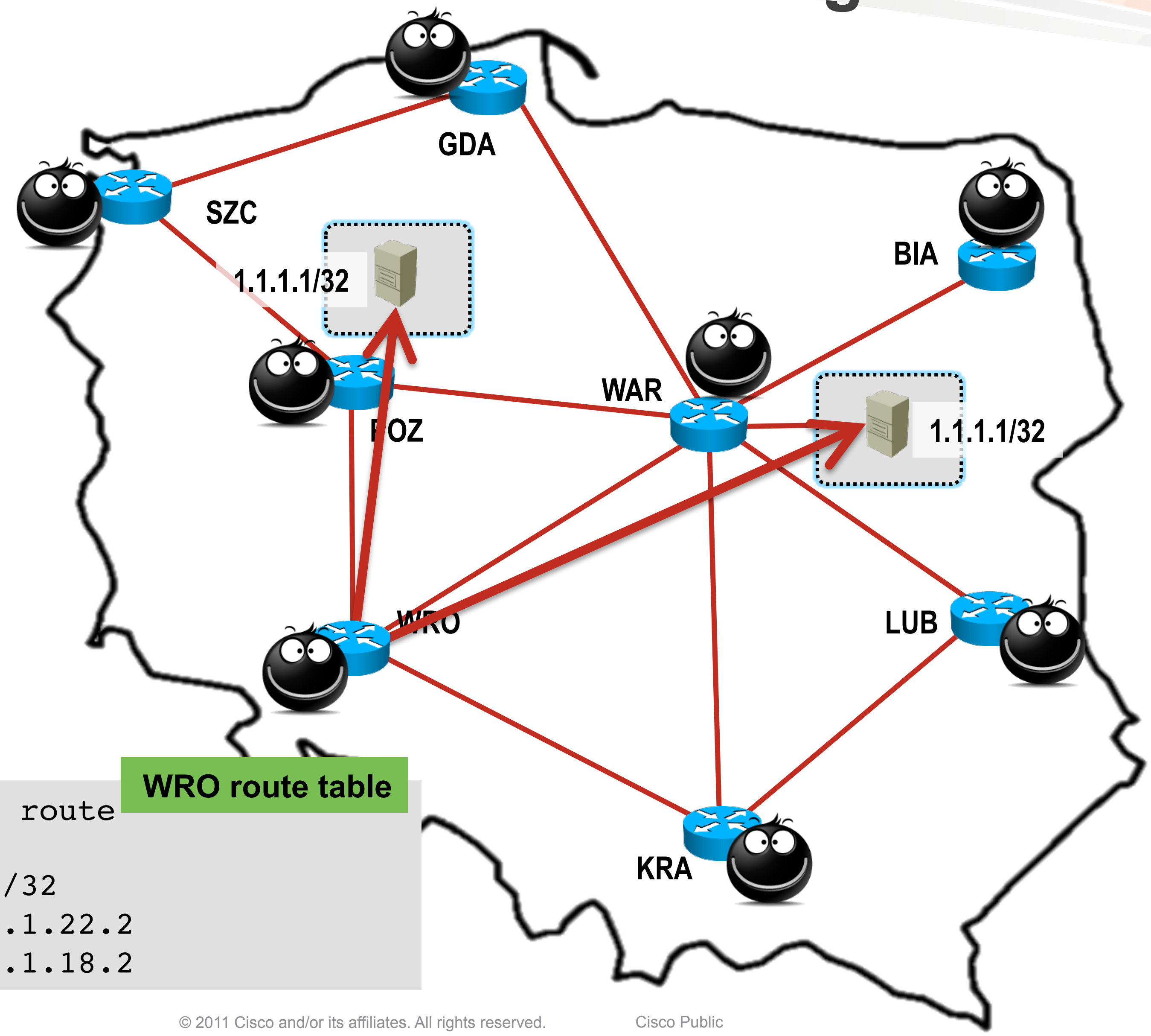
```
R2# sh ip route

*> 1.1.1.1/32
      via 1.1.5.10
      via 1.1.5.11
      via 1.1.5.12
      via 1.1.5.13
```

name query for anycasted name server

```
;; ANSWER SECTION:
ns1.anycasted.com.        86400    IN A     1.1.1.1
```

Network-wide load balancing



WRO route table

```
WRO# sh ip route

*> 1.1.1.1/32
     via 1.1.22.2
     via 1.1.18.2
```

DNS IP anycast solution

DNS system configuration - named

- named has to be carefully configured

  we answer queries for anycasted address, assigned to the loopback0 interface (1.1.1.1/32 & 2001:db8::100:100)

  we do our own queries and zone transfers from physical interface (or from separate loopback with separate address)

NS1-1 named.conf

```
options {
  listen-on { 1.1.1.1; };
  listen-on-v6 { 2001:db8::100:100; };
  query-source address 1.1.5.10;
  query-source-v6 2001:db8::34:254;
  transfer-source 1.1.5.10;
  transfer-source-v6 2001:db8::34:254;
};
```

NS1-2 named.conf

```
options {
  listen-on { 1.1.1.1; };
  listen-on-v6 { 2001:db8::100:100; };
  query-source address 1.1.5.11;
  query-source-v6 2001:db8::34:254;
  transfer-source 1.1.5.11;
  transfer-source-v6 2001:db8::34:254;
};
```

DNS IP anycast solution

DNS system configuration – quagga/OpenOSPFd

- Quagga should be set up with the IGP used in the AS

quagga.conf

```
interface em0
  ip address 1.1.5.11/24
!
interface lo0
  ip address 1.1.1.1/32
```

ospfd.conf

```
router ospf
  ospf router-id 1.1.5.11
  network 1.1.5.0/24 area 10
  redistribute connected route-map PF-DNS
!
route-map PF-DNS permit 10
  match ip address prefix-list dns-ospf
!
ip prefix-list dns-ospf permit 1.1.1.1/32
```

IP Anycast for HTTP

Scaling TCP services in general

- The topology changes should be limited
  - topology change means breaking TCP session, working around it requires either:
    - state synchronization (painful, "heavy", complex)
    - teaching enduser or middlebox (CPE) to accept it (works, proved to work in real cases*)
- If ECMP paths exist between end nodes, it may lead to problems
  - YMMV
  - may be worked around with simple tweaks to network logic topology (changing cost of links, etc)

* Case study on TCP streaming for IP anycast services, NANOG #37

HTTP anycast service

- If the topology changes during the session in established phase, it will break
- If the topology changes between client requests, some problems may happen

  i.e. cookies out of sync, causing login/password protected sites to bounce out client

  may be worked around with the backend synchronization (using different, non-anycasted set of addresses)

References

- Study on using TCP with anycast

  Anycast-aware transport for content delivery networks

  http://dl.acm.org/citation.cfm?id=1526750

- Operational experience with TCP and Anycast

  NANOG #37

  http://www.nanog.org/meetings/nanog37/abstracts.php

- BCP 126 / RFC 4786

  Operation of Anycast Services

  http://tools.ietf.org/html/rfc4786

- Architectural Considerations of IP Anycast

  http://tools.ietf.org/html/draft-mcpherson-anycast-arch-implications-00

Q&A

Thank you.

Łukasz Bromirski
lbromirski@cisco.com