IPv6 Regional Discussion
January 30, 2011

SoX
Southern Crossroads

DNS in IPv6

Joshua Hutchins

joshua.hutchins@gmail.com

Southern Crossroads

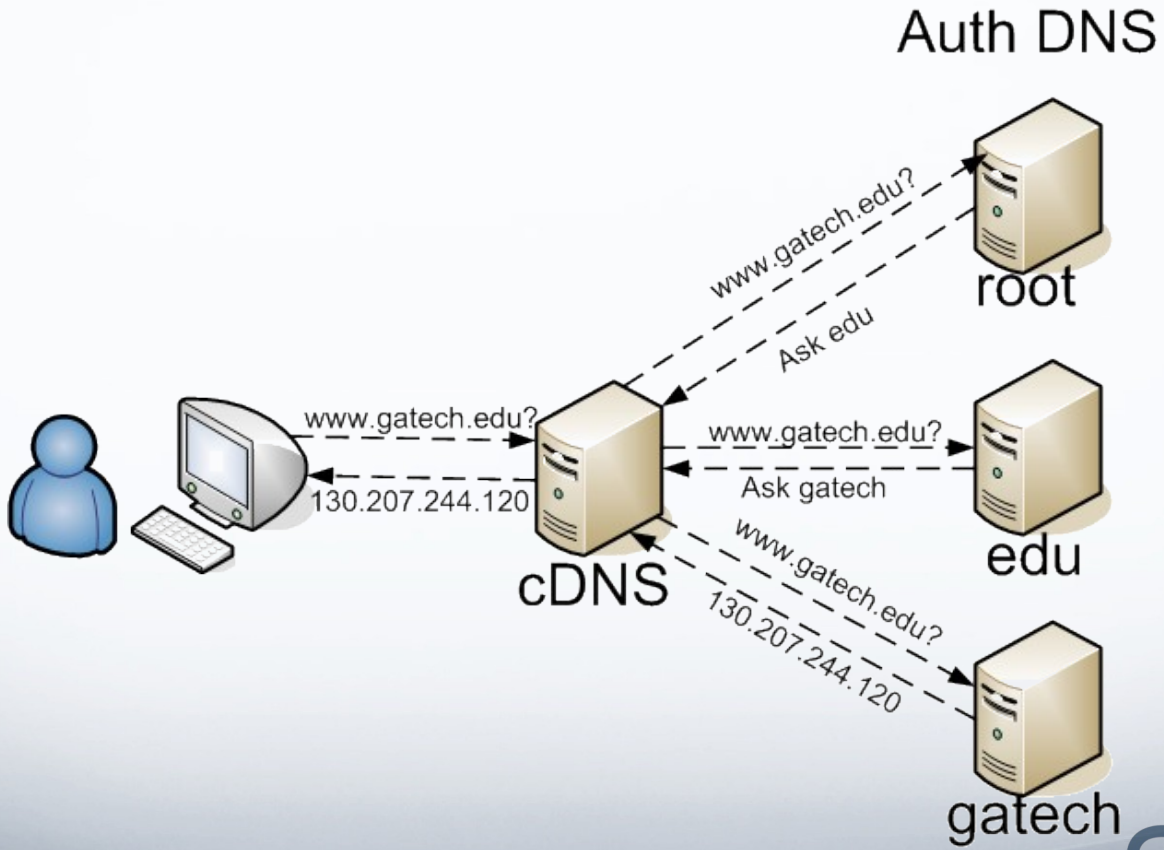
DNS - Short and Dirty

- Name to IP Translation
  - Name chaining
- IP to Name Translation (PTR)
- Simple, yet very important
- Need for speed and availability

Types of DNS

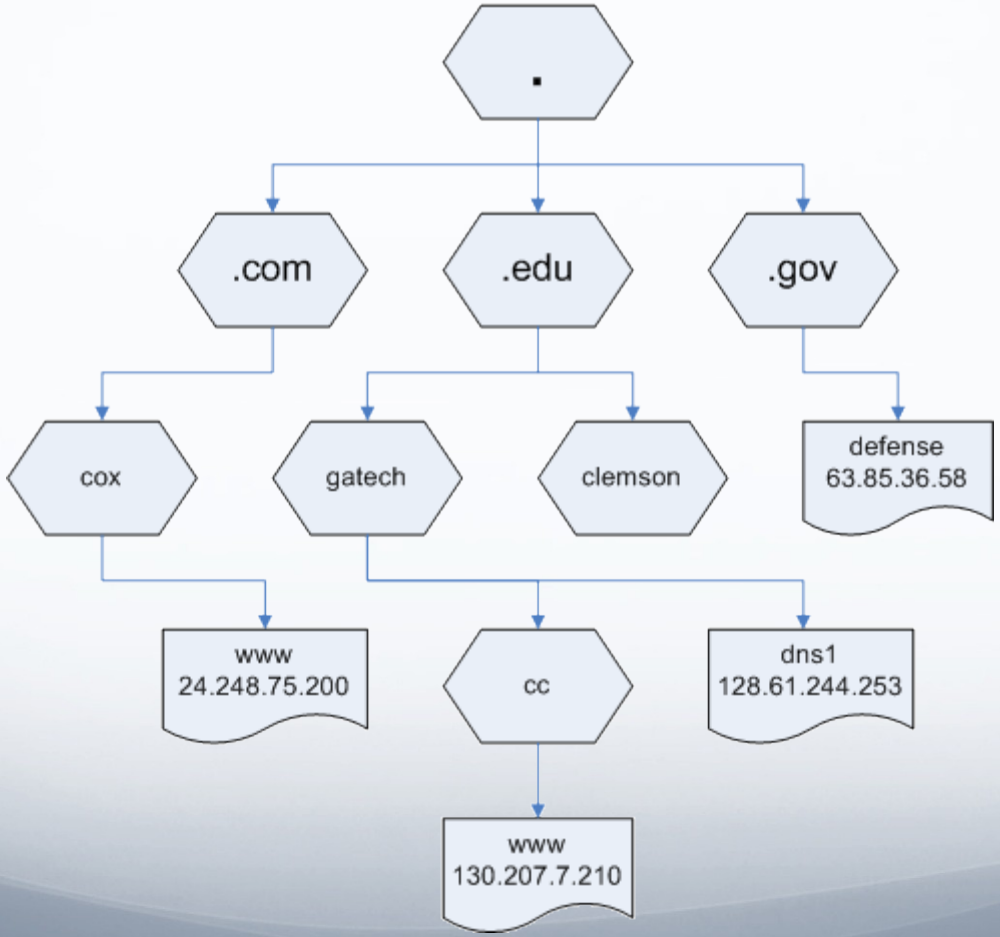
- Authoritative
  - Owns information
- Caching
  - Finds information
- Forwarder
  - Offloads caching responsibilities
  - Sends query directly to Auth server
- Internal vs. External

How does it work?



Southern Crossroads

How does it work?



Southern Crossroads

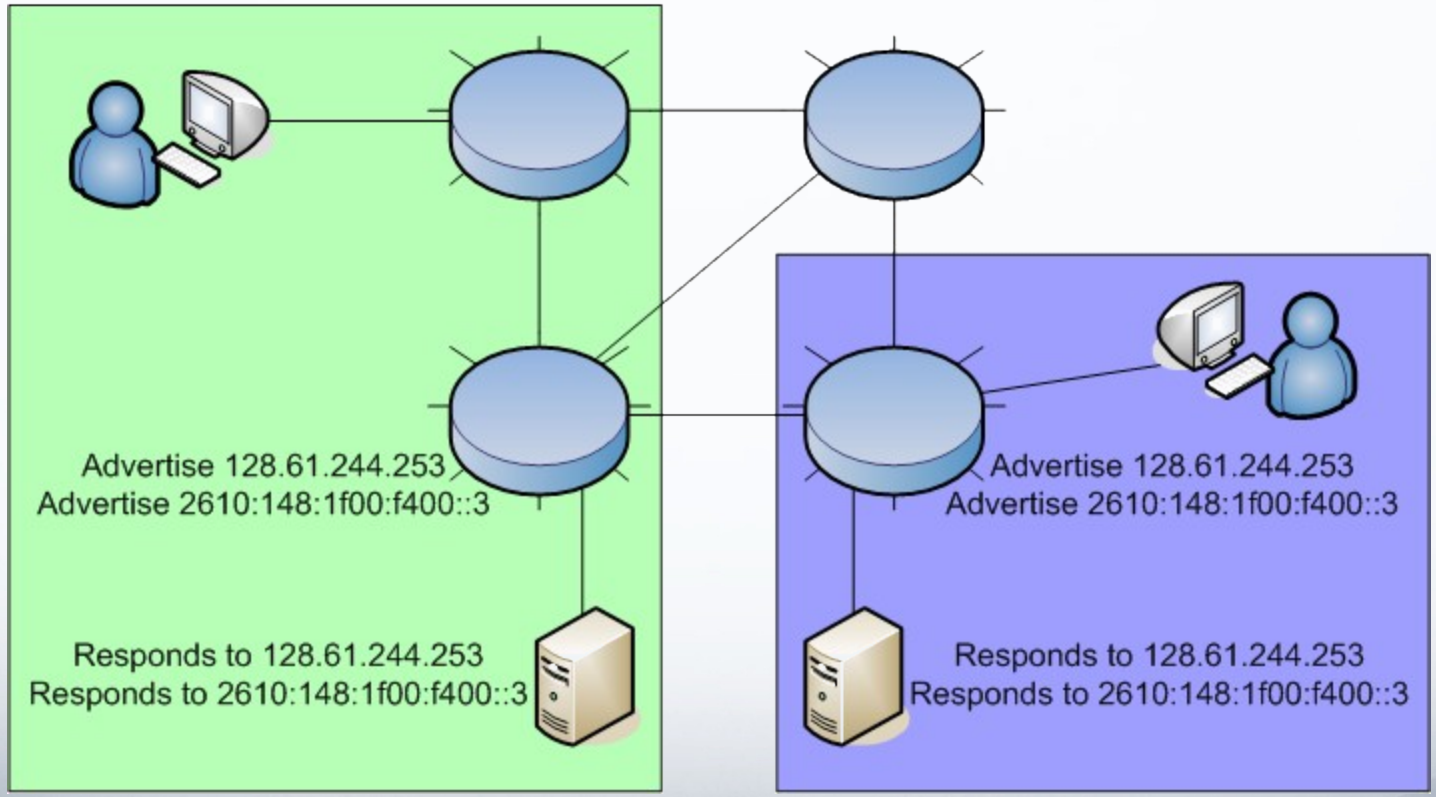
No Single Point of Failure

- Root Servers
  - 13 IPv4 IP's, 9 IPv6 IP's
  - 254 Sites worldwide
- gTLD's, ccTLD's, etc...
  - com, net, org, edu, gov, info, etc...
- Domains
  - gatech.edu, 8 Auth DNS, 3 locations

Anycast

- Multiple potential end points
- Nearest end point selected
- Fast failover; Route convergence
- Best for sessionless data
- Distribute load globally
- Excellent for DNS

Anycast in DNS



Anycast in DNS

- Most root servers use Anycast
  - D has 1 site in MD
  - J has 70 sites around the world
- Most gTLD/ccTLD's use Anycast
- Caching can also use Anycast
  - Google, Level 3, most ISP's

Delivery over IPv6
vs.
Delivery of IPv6 Data

- IPv6 Transport
  - Header of packet
- IPv6 Data
  - Payload of packet
- IPv6 DNS servers MUST be able to access IPv4 DNS servers

Delivery over IPv6

- Process
  - Give server v6 IP
  - Add "listen-on-v6" command

```
listen-on-v6 port 53 { any; };
```

- Static vs. SLAAC vs. DHCPv6

Delivery of v6 Data

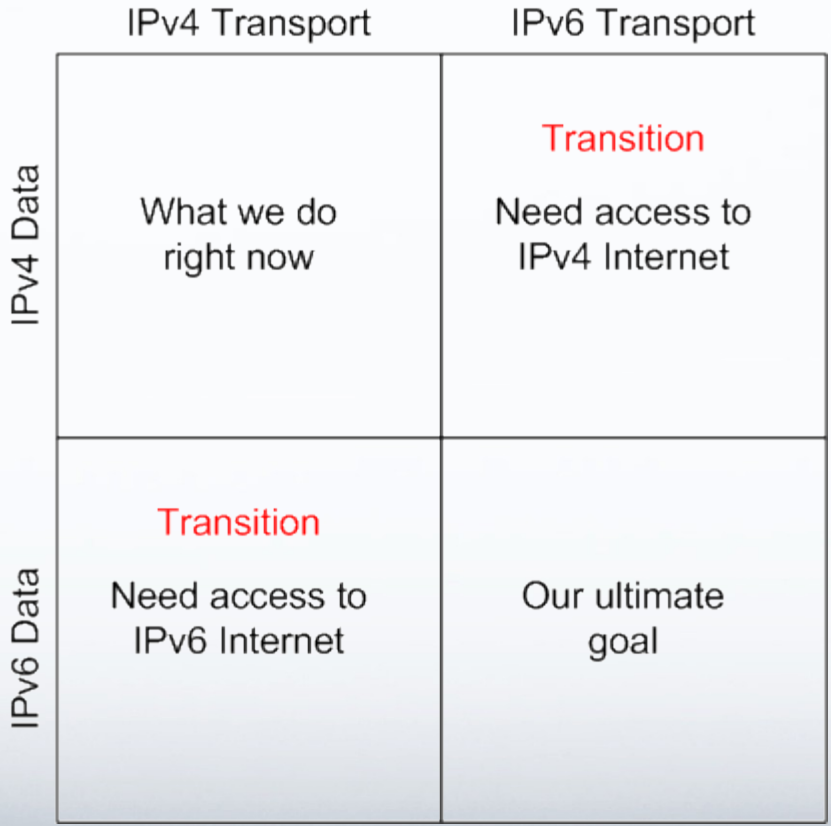
- AAAA Records

```
$ dig dns1.gatech.edu aaaa
;; ANSWER SECTION:
dns1.gatech.edu.  23669  IN  AAAA  2610:148:1f00:f400::3
;; SERVER: 8.8.8.8#53(8.8.8.8)
```

- v6 PTRs

```
$ dig -x 2610:148:1f00:f400::3
;; ANSWER SECTION:
3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.f.0.0.f.1.8.4.1.0.0.1.6.2.
ip6.arpa. 28800  IN  PTR  dns1.gatech.edu.
;; SERVER: 8.8.8.8#53(8.8.8.8)
```

Recap



Southern Crossroads

Dual-Stacking

- Dual-stacking a server
  - Second transport
- Dual-stacking a service
  - Second DNS name
- IPv6 is preferred

Reverse Delegation: IPv4

- Octet-level delegation
- Delegation occurs every 8 bits
  - /8, /16, and /24
  - Can require up to 128 delegations
- Smaller than /24 delegation

Octet Delegation: Example

- ARIN Assignment: 143.215.0.0/16
- Group needs /23
  - Assign 143.215.100.0/23
  - Delegate
    - 100.215.143.in-addr.arpa
    - 101.215.143.in-addr.arpa

Reverse Delegation: IPv6

- Nibble-level delegation
- Delegation occurs every 4 bits
  - Can require up to 8 delegations

Nibble Delegation: Example

- ARIN Assignment: 2610:148::/32
- Group 1 needs /44
  - Assign 2610:148:10::/44
  - Delegate 1.0.0.8.4.1.0.0.1.6.2.ip6.arpa
- Group 2 needs /47
  - Assign 2610:148:2a::/47
  - Delegate
    - a.2.0.0.8.4.1.0.0.1.6.2.ip6.arpa
    - b.2.0.0.8.4.1.0.0.1.6.2.ip6.arpa

But why PTR's?

- To see info in traceroute, ping, etc...
- SSH, FTP, etc... checking
- E-mail spam
  - Standard ISP users
  - Check forward against reverse
  - Scripted whitelist/blacklists

PTR in IPv4

- $GENERATE command in Bind
  - Generate /24 for 143.215.100.0

```
$ORIGIN 100.215.143.IN-ADDR.ARPA.
$GENERATE 0-255 $ PTR $.example.com

0.100.215.143.IN-ADDR.ARPA. PTR 0.example.com
1.100.215.143.IN-ADDR.ARPA. PTR 1.example.com
...
255.100.215.143.IN-ADDR.ARPA. PTR 255.example.com
```

$GENERATE in IPv6?

- Records in one /64 IPv6 network = 18,446,744,073,709,551,616
  - ~1.7 million petabytes on disk
  - 107 bytes per line, $.example.com

- How to deal with PTR in IPv6, then?

IPv6 PTR's

- Manage records individually
  - Most IP's won't have PTR
  - Impossible for dynamic addressing
  - Need for e-mail spam changes

IPv6 PTR's

- DDNS
  - Great for dynamic addressing
  - Potential for security problems
    - Best security with DHCPv6
    - Limit use with SLAAC
  - Still not 100% coverage

IPv6 PTR's

- On-the-fly / Wildcard PTR's
  - Poor support
    - PowerDNS + pipe backend + v6rev + custom scripts
  - Performance implications

- No "standard" solution yet

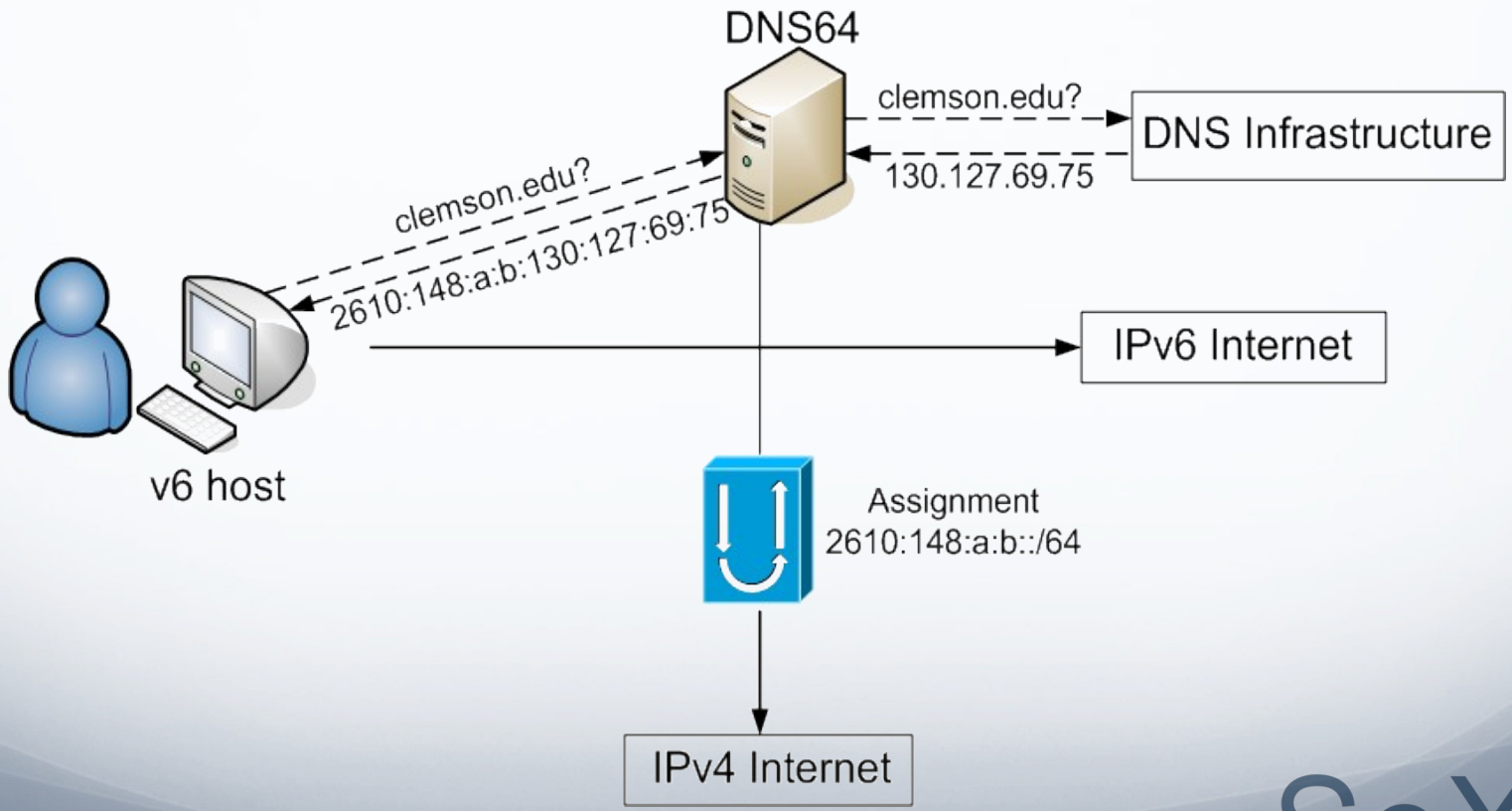
Transition Technologies

- Tunneling
  - Teredo, 6to4, 6in4, DSLite, etc...
- Translating
  - NAT444, NAT64
- Different methods for different levels of v6 deployment
- NAT64 requires DNS64 to work

NAT64 & DNS64

- v6 host trying to get to v4 server
- Host makes DNS query, gets A
- DNS responds with AAAA
  - Prefix to NAT device, A suffix
- Requires Views or second DNS
  - Views cause reduced performance

NAT64 Example



SoX™
Southern Crossroads

Performance

- IPv6
  - Larger headers + software inspection
  - Larger payloads
  - Transition technology
- DNSSEC
- Larger QPS

DHCPv6

- Stateful
- Most control of IP assignment
- DHCP Snooping + ARP Inspection
- Pass info to users
  - Set DNS servers
- Best security with DDNS

SLAAC

- Stateless
- No need to manage IP assignment
- Stop access via 802.1x
- Low security with DDNS
  - Hosts/users set own DNS

Stateless DHCPv6

- SLAAC + DHCPv6
- ND Flag at router
- Some options allowed, not all
  - DNS Servers