
/delivers/

# No Helpdesk for Light Switches

## The Unbearable Lightness of Being Everywhere

Joe Ableyjoe

@ableyjoe

DNS-OARC, Warsaw, 2014

# Technical Awareness of End-Users in Decline

Let us rejoice together in our collective lack of surprise

- a consequence of the continued mainstreaming of the Internet as a conduit for all things
  - refrigerators, thermostats, doorlocks, alarms
  - news, dictionaries, travel, television, phone, mail, everything
- most end-users of bathrooms don't understand plumbing, either

# Who do an ISP's Customers Call?

- if a service is broken, call the service operator (maybe)
- if a device is broken, call the shop (maybe)
- if the network is broken, call the ISP (maybe)
- get your teenage child to look at it
- if none of those things work, you're stuck
    - stop using whatever it is

# Outsourced Reliability

- if you're an established outfit with revenue, you can build massive infrastructure
    - expensive, difficult
- if you're a tiny start-up perhaps you can't afford the cost of a huge build-out, which is a shame because your idea depends on reliability, more than the big guys, even
    - rise of the data centre, rise of the cloud
    - compute, storage, operations, …. and DNS

# Wide-Scale Distribution of DNS Service

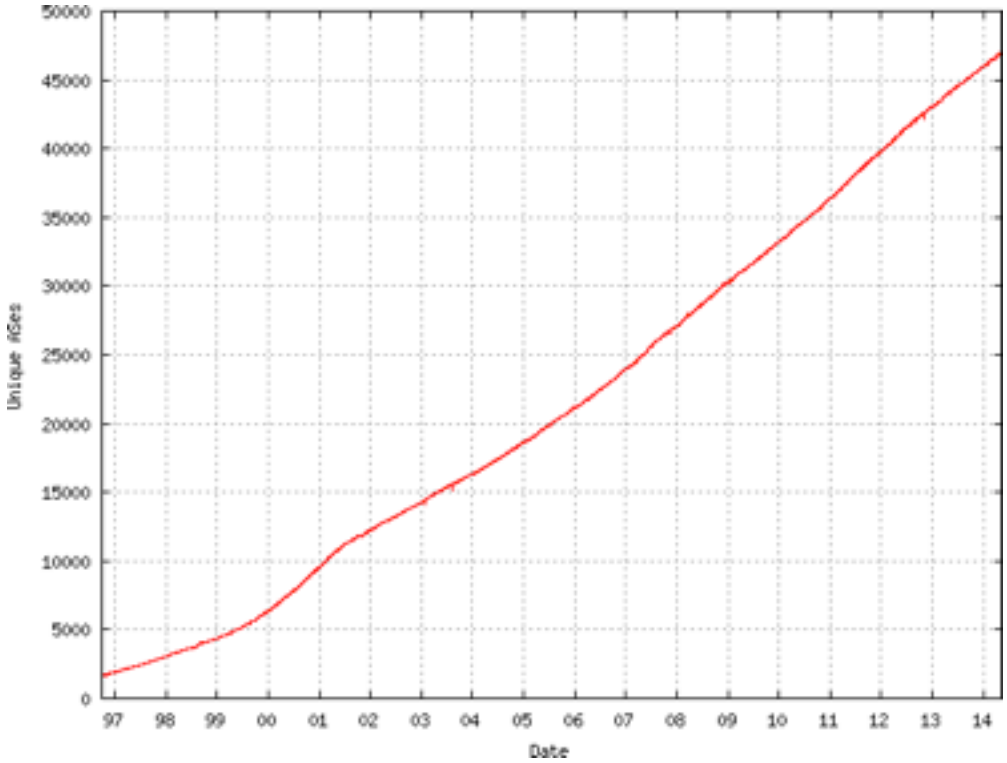
- People have been using anycast to distribute DNS service for a long time
  - authoritative DNS service, recursive DNS service
  - protocol is (often, usually) stateless
  - transactions are (often, usually) short-lived
  - largely unaffected by routing churn
    - probably, apparently

# What is local, and what is remote?

- Lots of Internet end-users follows a similar pattern
- If all your YouTube, FaceBook, Netflix works locally, then do you notice if remote things are broken?
- Who do you blame if your expectations are not met?
- The End-to-End Principal is vital and important
  - let’s assume it is dead
- What next?

# How much Anycast is Enough?

# How many ASes are there?



# Aim to Scale

- If we're going to build out Internet infrastructure on a grand scale, we may as well aim big
  - "at least one node per AS"
- Keep service local (keep attack flows on-net)
- Minimise the RTT
- Reduce the hardware requirements of an individual node to something that can be built for $2,000 and treated as an appliance

# Which ASes?

- All of them.
- OK, at least as many as want them.
  - reduced end-user support costs
    - increased customer performance
  - containment of attack flows on-net
    - this one is bigger than you might imagine

# Operational Implications

- If we don’t want to hire thousands more people, we need all these nodes to be heavily automated
  - self-service for network operators (renumbering, BGP, maintenance, etc)
  - ship direct from factory to site
  - installation and troubleshooting simplified to a level that would not challenge a small child
  - low-power appliance, two-post rack, no shelves, no rails

# Security Considerations

- Every node is installed in a hostile, remote network
- Starting point is to assume that Bad People are going to compromise the box immediately, if not sooner
  - no secrets on any node beyond those that are relevant to the node itself
  - regular, frequent, automatic bare-metal reinstalls
- careful thinking required

# Operational Management

- Patch and configuration management (plus associated test processes) completely automated
- Element monitoring (centralised and distributed) provisioned along with each node, so that the list of things to test is always up-to-date
    - careful thinking for escalation, to avoid the situation where a single problem causes 50,000 alarms per second, and the NOC shoot themselves

# Data Flow

- Many of these nodes will be in dark, cobwebby, poorly-connected parts of the Internet
    - need to be light on the network and extremely tolerant of congestion and partition
    - opportunistic peer-to-peer communications, allowing a swarm of nodes to exchange provisioning data, service data and the results of data analysis with each other
- We do not expect full centralisation of data to be possible, so we need to be able to distribute analysis to the edge

# Mobile Tenants

- We want applications on this new edge to be constrained, so they can't hurt anything that is already running there
  - breed operational confidence that services can be moved without planning
  - enable agile reactions to operational calamities
  - foster an expectation of resource planning
    - things that need more than is there will not start

# When Things Break

- Sometimes the right thing to do is withdraw service
- Other times the right thing to do is to sink the junk locally and protect the rest of the world
    - being able to distinguish between the two is a job for humans
    - Humans are expensive
    - RISE OF THE NETOPS ROBOT ARMY
        - It’s Dyn, not Cyberdyne. Honest.

# Add it together and what do we get?

- Massively-redundant, massively-distributed DNS service
  - more reliable, faster, shiny
  - ridiculously scaleable
- A new level of service intelligence
  - as many views of the global routing table as we have nodes
  - an additional dimension for assessing client reputation, significantly less prone to error from external topology changes

# Progress

- 20 nodes about to go live in Dyn-controlled environments
  - baseline performance
- Remote deployment starting in Q3/2014
  - Sprints of 200 units
  - Locations prioritised by customer demand, and by whomever buys us the most beer at DNS-OARC meetings

Dyn

SM