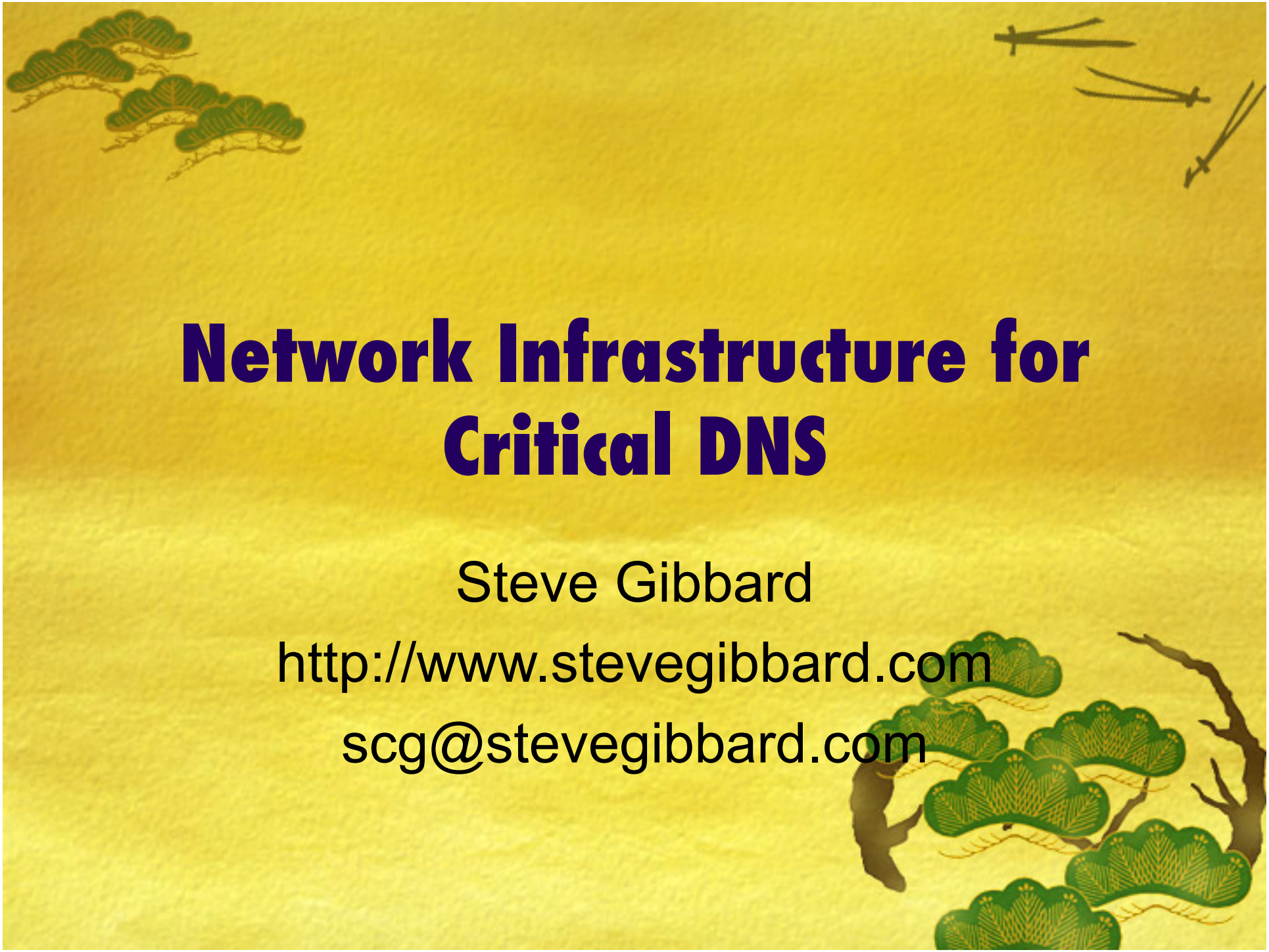


Network Infrastructure for Critical DNS

Steve Gibbard

http://www.stevegibbard.com

scg@stevegibbard.com

Introduction

- Mixing two talks:
  - Infrastructure Distribution
    - Where are DNS servers for ccTLDs?
  - DNS network architecture
    - Where and how should name servers be connected?
- Focusing on network infrastructure
  - Lots of important stuff happens on the servers too, but that's not my area.

DNS is critical infrastructure

- Without DNS, nothing else works.
- Authoritative DNS needs to be as reliable as the most reliable parts of the network.
- DNS is a hierarchy. For a domain name to work, its servers and those for all zones above it must be reachable.

Reliability is best close to authoritative servers

- There's less to break between the server and the user.
- Response times are faster.

gTLDs Focus Mostly on Core

Out of date .Com/.Net map (from March, 2007)

ccTLDs are location-based

- They're depended on by users in their countries.
- They may be used in neighboring/trading partner countries.
- People outside may not care much.
- It's somewhat obvious where they should be reliable.
- Local root servers are needed too.

Network partitions

- In a network partition, it's good if local stuff keeps working.
  - In satellite-connected regions, international connectivity breaks frequently.
  - Outages are rarer in fiber-connected regions, but last longer.
  - Local phone calls work without international connectivity. Local Internet should too.

DNS look-ups around the world

- Pakistan and .PK
  - Root look-ups handled locally, but ccTLD look-up are handled in the US.
  - Karachi has a root server.
  - .PK in UUNet and ev1Servers networks in US.
- Kenya and .KE
  - Root and TLD look-ups are handled locally.
  - Nairobi has multiple root servers.
  - .KE is hosted in Kenya and elsewhere.

Notable incidents

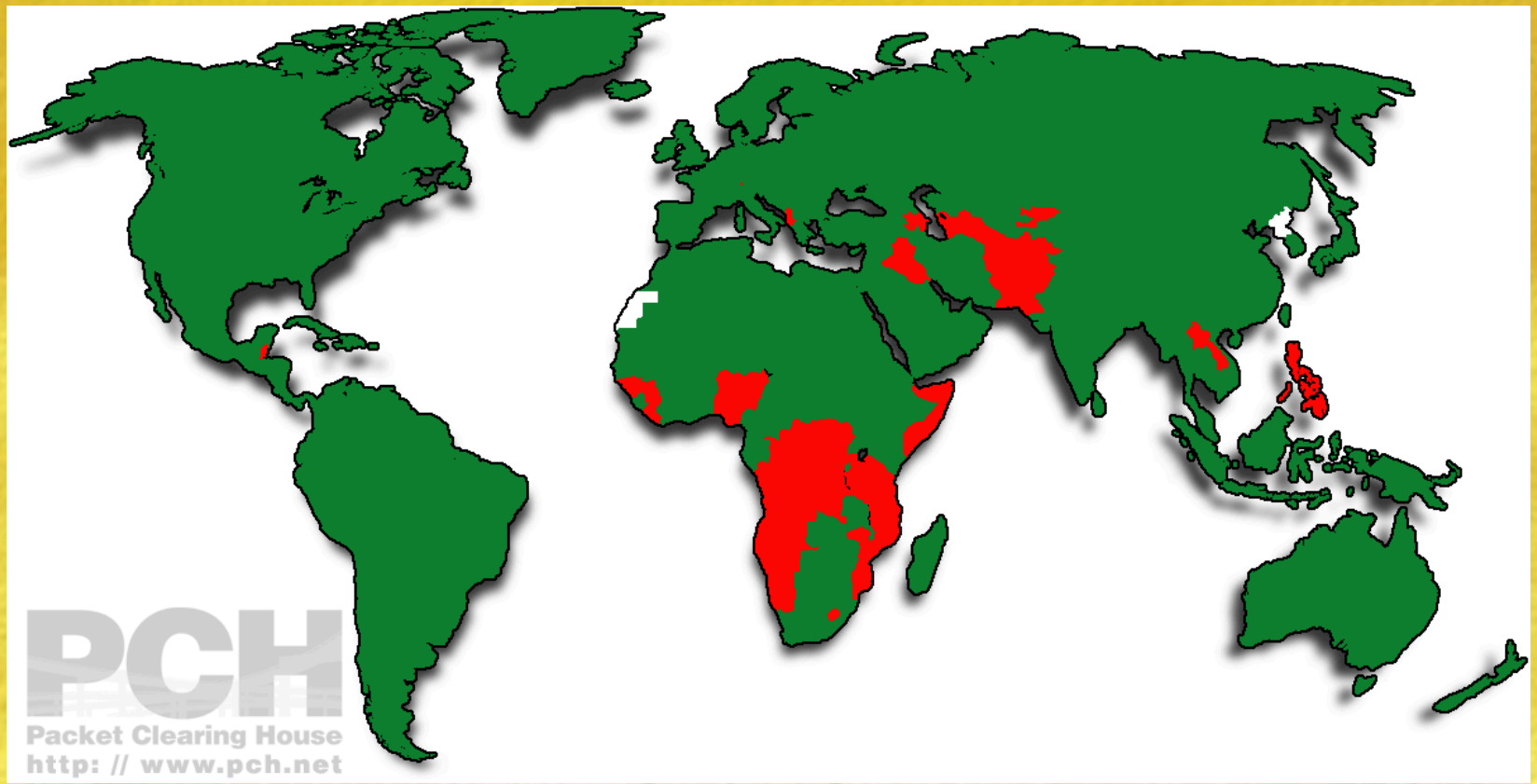
- Sri Lanka (2004)
  - International fiber was cut in Colombo harbor.
  - Press reports described an outage of “Internet and long distance phone service.”
  - ccTLD hosted locally, but no root server.
- Burma/Myanmar (2007)
  - International connectivity was cut off by the government.
  - Local connectivity kept working.
  - .MM worked inside but not outside.

Root Server Coverage

ccTLD Distribution:

- Just over 2/3 of ccTLDs are hosted in their own countries.
  - (but a lot of those that aren't are for really tiny countries).

Countries with local ccTLDs (green) old data



ccTLDs not hosted in core (old data)

- .AX -- Aland Islands
- .BB -- Barbados
- .BH -- Bahrain
- .CK -- Cook Islands
- .CN -- China
- .EC -- Ecuador
- .GF -- French Guiana
- .KW -- Kuwait
- .MP -- Northern Mariana Islands
- .MQ -- Martinique
- .NF -- Norfolk Island
- .PA -- Panama
- .PF -- French Polynesia
- .QA -- Qatar
- .SR -- Suriname9
- .TJ -- Tajikistan
- .YE -- Yemen
- .ZM -- Zambia
- List used to include .BD - Bangladesh -- Now fixed.

Building DNS infrastructure

- Goals
- How to build it
- Topology
- Redundancy

Goals

- Who are you trying to serve?
  - Local users?
  - Users in other local areas?
  - The rest of the Internet?
- Your region's topology:
  - Is everything well-connected, or a bunch of "islands?"
  - Servers in central location, or lots of places?

Whose infrastructure?

- Your own?
- Somebody else's?
  - Free global anycast services for ccTLDs provided by ISC, PCH, others
  - Several commercial anycast operators
  - Lots of free unicast options
  - Easy way to get large-scale global-build
- Mixture?
  - Your own servers in areas that matter most to you
  - Somebody else's global footprint

Where to put the servers

- In country
  - At a central location -- an exchange point?
  - One in each ISP?
  - At a common uplink location (like Miami for Latin America)?
- In the rest of the world:
  - At major Internet hubs?
  - At the other end of your ISPs' international links?

Unicast/anycast:

- This is mostly an issue of scale.
- For small numbers of servers, unicast works well.
- Anycast is required for larger numbers of servers.

Anycast topology – keeping traffic local

- Backbone engineers are often good at keeping local traffic local.
  - Use consistent peering/transit, hot potato routing.
  - Unicast operators don't need to think about this.
- Anycast DNS operators aren't so good at this.
  - Anycast looks like a backbone.
  - Plugging servers into random networks is done in pursuit of network diversity.
  - Networks send traffic to customers first, regardless of geography.

J-Root in Bay Area

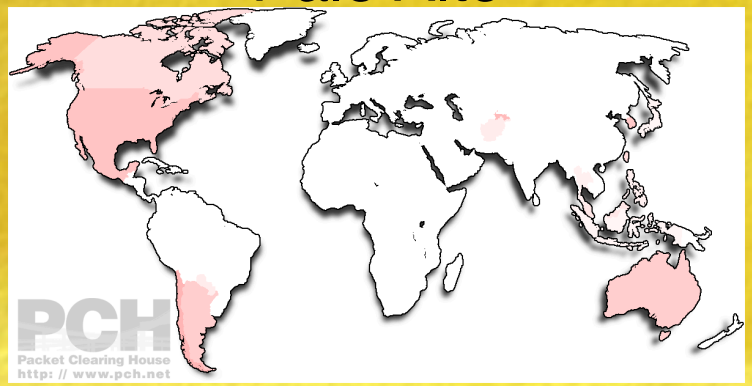
- There are local J-Root servers in Mountain View and San Francisco.
- Queries from 3 Bay Area hosts are responded to by:
  - jns2-kr
  - jluepe2-elmad1
  - jns2-elyyz

Anycast can keep traffic local

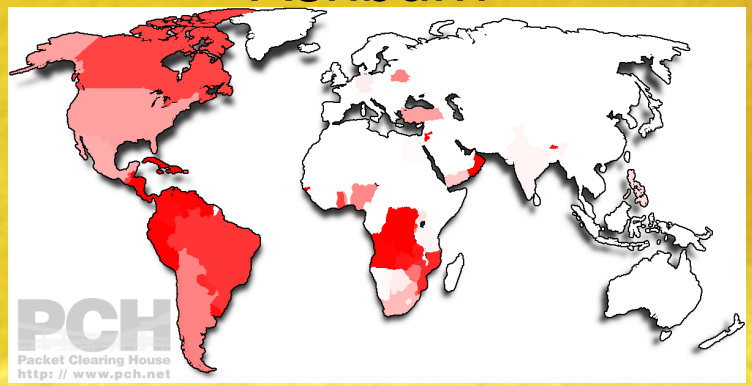
- If designed like a backbone.
- Consistent transit should be gotten from global ISPs.
- Peering only locations are good too, but peer with peers in all areas of overlap.
- No transit from non-global providers.:
  - Insist on being treated like a peer.

Queries with consistent transit

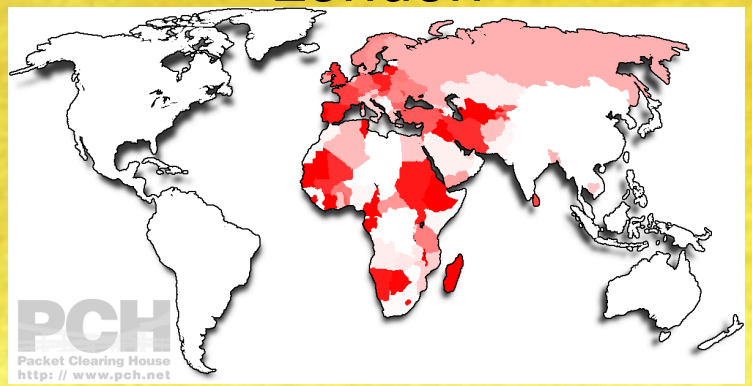
Palo Alto



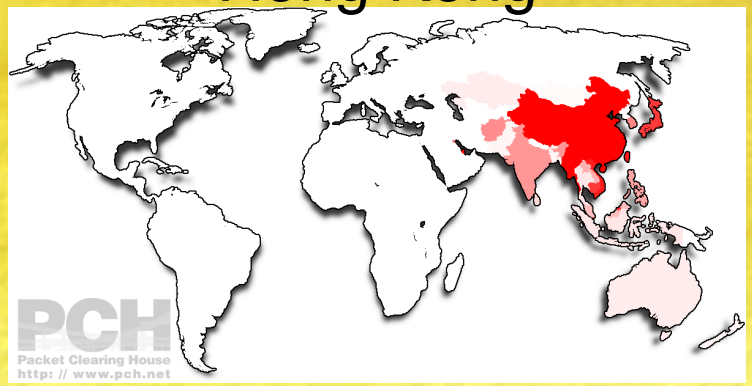
Ashburn



London



Hong Kong



Redundancy

- More servers are better than fewer, if they're manageable.
- There's no contradiction between using your own servers and outsourcing.
- Monitoring:
  - Check zone serial numbers on all servers frequently.
  - If using anycast, monitor individual unicast management addresses.

Further reading

- DNS infrastructure distribution
  - http://www.stevegibbard.com/dns-distribution-ipj.pdf
- Observations on anycast topology and performance.
  - http://www.stevegibbard.com/anycast-performance.pdf


Thanks!
Steve Gibbard
http://www.stevegibbard.com
scg@stevegibbard.com