Motivations and Terminology for DNSSEC Handover

Steve Crocker
Ólafur Guðmundsson
Andrew Sullivan
Shinkuro, Inc.

April 4, 2011

The Challenge: Ripple-free Transfers

- Move a signed zone
- Without disrupting validation

  and
- Without losing resolution.

Registrars vs DNS Operators

- Registrars handle registration
  - Convey NS and DS records to parent
  - Handle whois data

- DNS Operators provide name service

- Most registrars provide name service, and most registrants get their name service from registrars, but nonetheless, these are different.

Peeking Ahead

- Ripple-free transfers are indeed possible if…
- Losing Operator is cooperative AND
- Someone, either registrant or agent, has necessary access to registrar AND
- Timing constraints are observed

Yes, there's a key rollover

- We assume the DNS Operator is the one who signs the zone
- We assume the Old ("Losing") Operator and the New ("Gaining") Operator will use different keys
  - Private keys do not travel between operators
- Thus, the transfer process includes a rollover.

A Word About Caching Resolvers

- A strategy for ripple-free transfer has to work for essentially all resolvers
- But not all resolvers behave the same way
- And the differences do matter

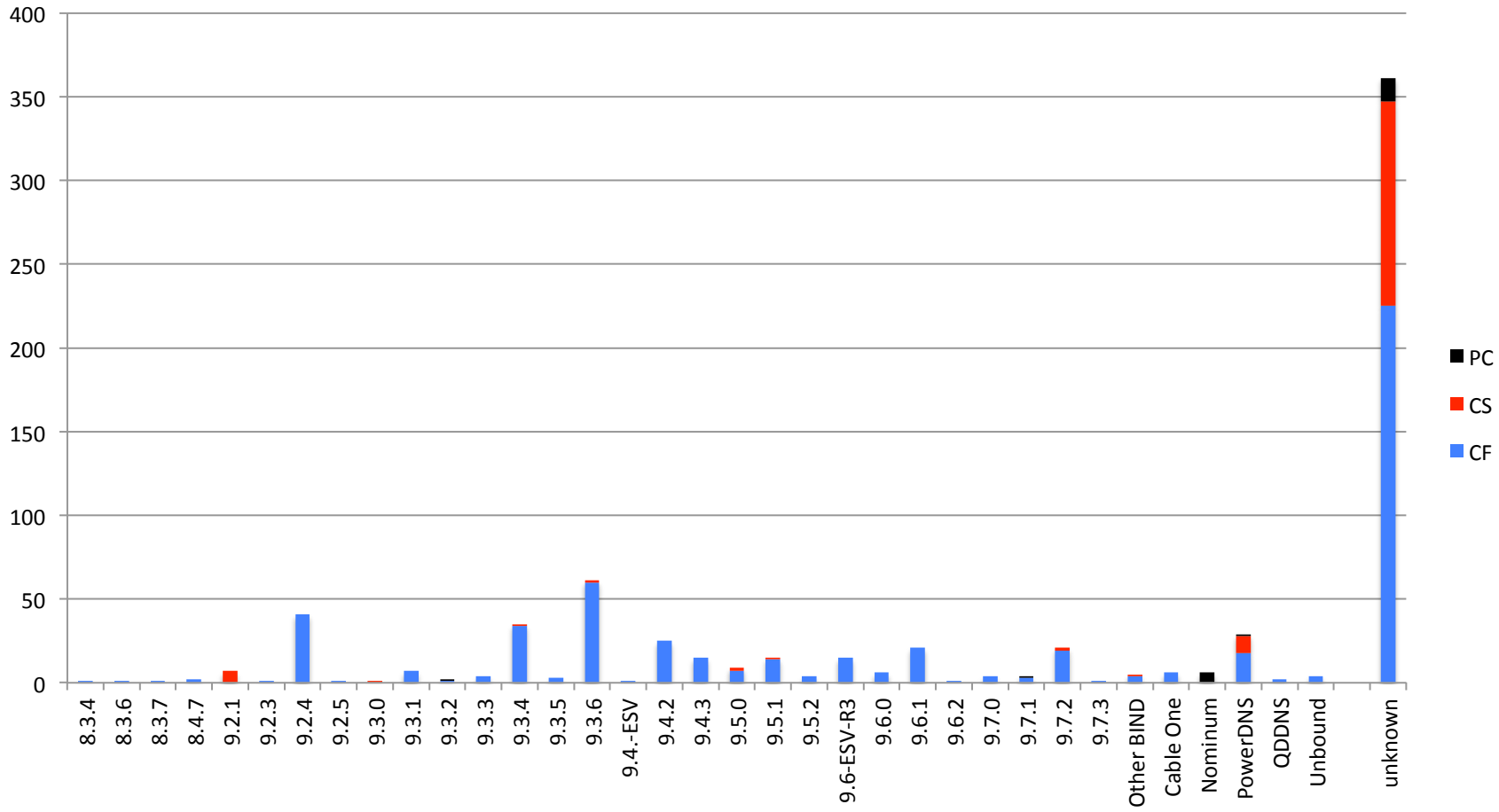
Classes of Caching Resolvers

- CF – Child Focused
  - Renews expired NS RRset from child
- CS – Child Focused, Sticky
  - Renews expired NS RRset from child
  - Opportunistically extends TTL if there's a related query
- PC – Parent Centric
  - Renew expired NS RRset from parent

Selected Totals: 18 Feb 2011 Survey

Version	CF	CS	PC	Total
8.3.4	1			1
8.3.6	1			1
9.7.3	1			1
Other BIND	4	1		5
Nominum			6	6
PowerDNS	18	10	1	29
Unbound	4			4
unknown	225	122	14	361
Grand Total	552	148	23	723
Percents	76.3%	20.5%	3.2%	

Graphic Summary



Notation used

- Lower case: contents from old operator
- UPPER case: contents from new operator
- k, K: Key Signing Keys
- z, Z: Zone Signing Keys
- n, N: Nameserver sets
- d, D: DS records pointing to k or K respectively
- r, R: DNS data
- r(z) : RRset signed by z, (from old operator)

Timing issues

- All waits are expressed as TTL of an RRset
- The timer starts once the LAST name server for that operator reflects the change
- When a rule has a MAX that covers TTLs from two operators (parent and child) the second party's TTL has the delay to perform the action added to the value
  - We assume parent will perform actions before child for simplicity reasons but in some cases the order does not matter.

Ripple Free DNSSEC preconditions

- Old operator
  - is DNSSEC capable
  - Is cooperative (Will add new Z on request)
- Parent
  - Will accept DS for a key not in DNSKEY
- New operator
  - Is DNSSEC capable
- No sharing of keys

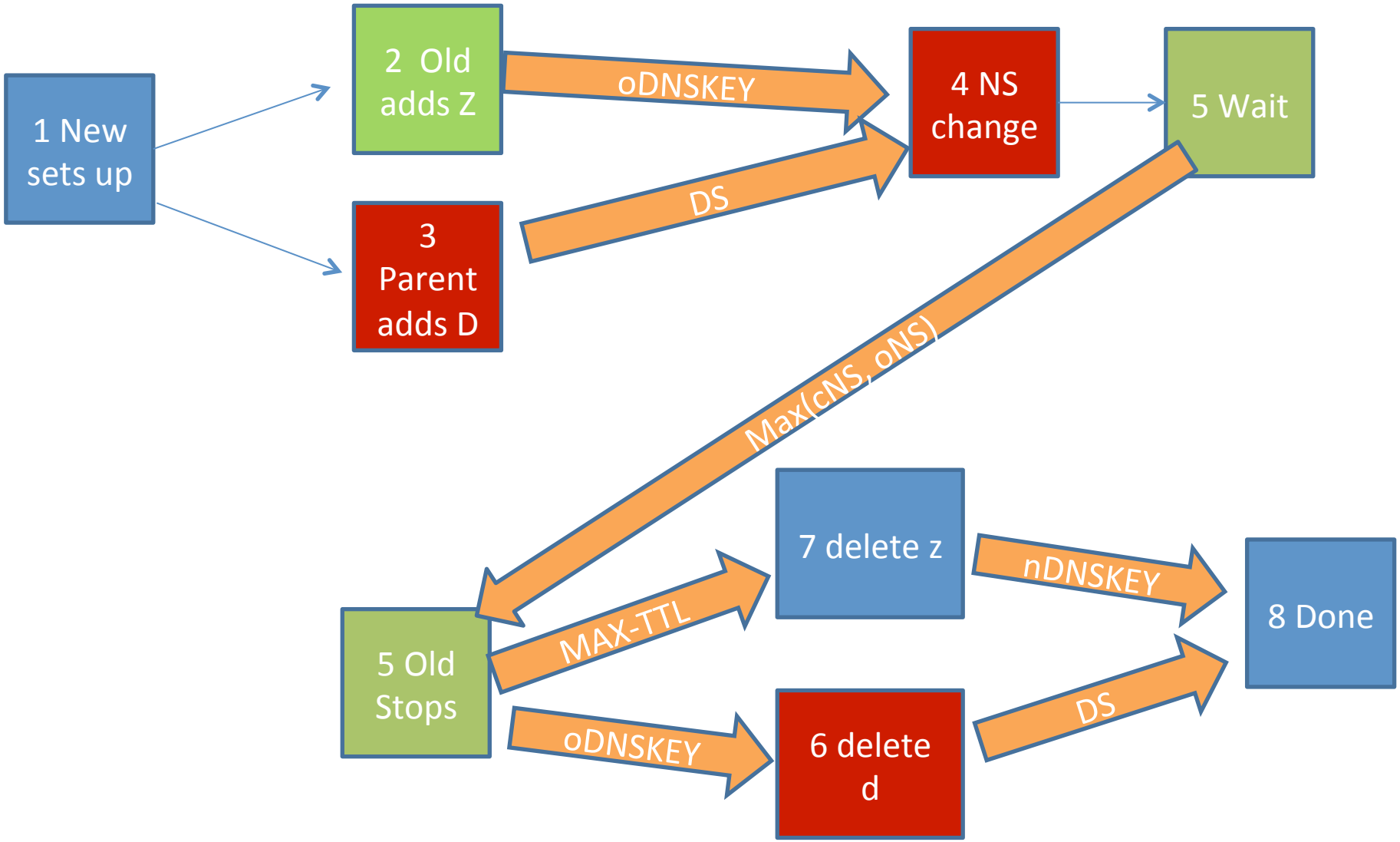
Ripple-free operator change

Actions

1. New brings up zone
2. Old adds Z to DNSKEY
   - **Essential for sticky resolvers**
3. Parent adds D to DS
4. Parent changes NS
   - Wait: MAX(parent NS, old child NS)
5. Old Phases out
6. Parent deletes d from DS
7. New deletes z from DNSKEY
8. Done

	Old	Par	New
0	kz,n,r(z)	n,d	
1	…	…	N,KZz, R(Z)
2	kzZ,n, r(z)	…	
3	…	n,dD	
4	…	N,dD	
5	X	…	
6		N,D	
7		…	N,KZ, R(Z)
8		…	

Ripple-free DNSSEC operator change



More to Come

- Precise timing analyses
  - More detailed diagrams and tables
- Effects of less safe sequences
- Advice re setting timing parameters
- Advice re future resolvers
- Clarity about roles: who tells registrar about new operator, what authority is required, etc.
- Potential for automation of transfers
  - It may be desirable to create a small protocol