RFC 2181 Ranking data and referrals/glue importance

--- new resolver algorithm proposal ---

Kazunori Fujiwara
fujiwara@jprs.co.jp
Japan Registry Services Co., Ltd (JPRS)

DNS-OARC Workshop 2016/10/16

Last update: 2016/10/16 0810 UTC

Presentation summary

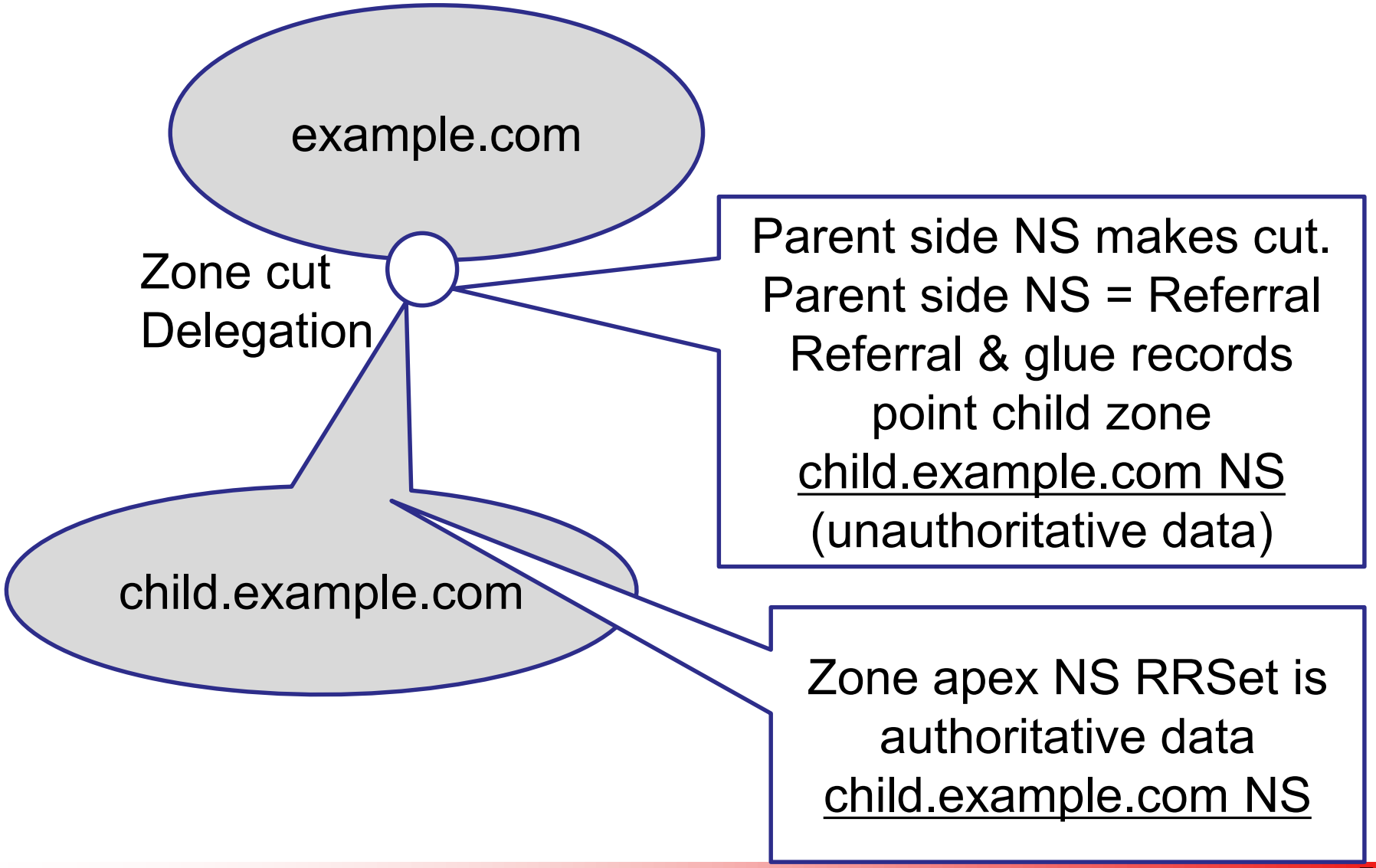
- Parent side NS RRSet (=referrals) creates zone ‘cut’ and ‘new zone’
  - Parent side NS RRSet and glue records are all information to access servers for child zone
- However, parent side NS RRSet may be overwritten by child zone apex NS RRSet
  - Glue records are also overwritten by authoritative data
- Proposal: (simplified) new resolver algorithm
  - Only use referral + glue records (+ additional name resolution for out-of-bailiwick name server name) to iterate
  - Resolvers answer authoritative data only
  - Update RFC 1034 and RFC 2181

Sources of definitions

- RFC 1034 DOMAIN NAMES – CONCEPT and FACILITIES
- RFC 1035 DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION
- RFC 2181 Clarifications to the DNS Specification
- RFC 7719 DNS Terminology
  - Because RFC 1034 and RFC 1035 did not define DNS terminologies well

Current definition: referrals and glue records

Referrals and glue records



(Zone) “cuts”

- "cuts" in the name space can be made between any two adjacent nodes. After all cuts are made, each group of connected name space is a separate zone. The zone is said to be authoritative for all names in the connected region.(Quoted from RFC 1034, Section 4.2)
  - “cuts” makes new zone

Who makes “cut” ?

- “The RRs that describe *cuts around the bottom of the zone are NS RRs that name the servers for the subzones*. Since the cuts are between nodes, these RRs are NOT part of the authoritative data of the zone, and should be exactly the same as the corresponding RRs in the top node of the subzone.” (Quoted from RFC 1034, section 4.2.1)
  - Parent side NS RRSet makes “cut”. Parent side NS RRSet is not authoritative data.
  - Authoritative data at “cut” is in the subzone
  - Parent side NS RRSet and child zone apex NS RRSet should be the same

Parent side NS RRSet

- “That is, parent zones have all the information needed to access servers for their children zones.” (Quoted from RFC 1034, section 4.2.1)

→ Parent side NS RRSet and glue records are all information to access servers for child zone

- “The simplest mode for the client is recursive, since in this mode the name server acts in the role of a resolver and returns either an error or the answer, but never referrals.” (Quoted from RFC 1034, Section 4.3.1)

→ Parent side NS RRSet is “referral”.

“referral” must not be used as name resolution result

Delegation

- “The process by which a separate zone is created in the name space beneath the apex of a given domain. Delegation happens when an NS RRset is added in the parent zone for the child origin.” Quoted from RFC 7719
- “This situation typically occurs when the glue address RRs have a smaller TTL than the NS RRs marking delegation,” Quoted from RFC 1035, Section 7.2

→ Parent side NS RRSet makes delegation (zone cut)（a new zone)

Referral

- “Resolvers must be able to access at least one name server and use that name server's information to answer a query directly, or pursue the query using referrals to other name servers.” (Quoted from RFC 1034, Section 2.4)

→ Referrals are used to pursue queries to other name servers (not child zone apex NS RRSet)

(Referrals are parent side NS RRSets)

Zone Cuts defined in RFC 2181

- “The existence of a zone cut is indicated in the parent zone by the existence of NS records specifying the origin of the child zone.” (Quoted from RFC 2181, Section 6)

Summary of Referrals defined in RFC 1034 and 1035

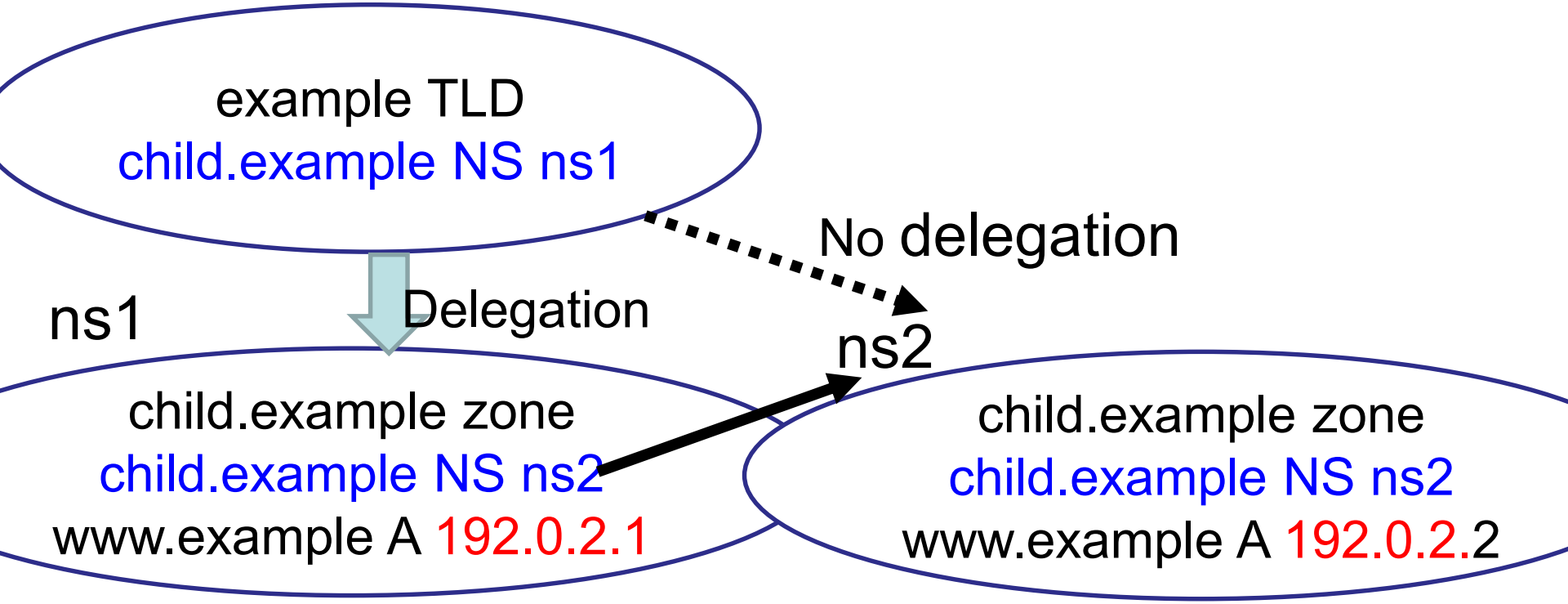
- Parent side NS RRSet is referral (in RFC 1034 and 1035)
- Parent side NS RRSet makes zone “cut” (delegation) and a new zone
- Referrals and glue records are all information to access servers for child zones
  - More important than zone apex NS to iterate
- “Referral” and glue records are not authoritative data
- Authoritative data around “cut” is the zone apex NS RRSet in the child zone

Current resolver algorithm

Resolver algorithm

- Described in RFC 1034 section 5.3.3
- “2. Find the best servers to ask.”
- “3. Send them queries until one returns a response.”
- “4. Analyze the response, either:”
- “4b. if the response contains a better delegation to other servers, cache the delegation information, and go to step 2.”
  - This procedure caches referrals and glue
- “4a. if the response answers the question or contains a name error, cache the data as well as returning it back to the client.
  - This procedure caches authoritative data
  - NS RRSet at a zone apex may be cached (It may overwrite referrals and glue)

Parent NS and child NS mismatch



- First resolve: “www.example A” returns 192.0.2.1
  with “example NS ns1”
- Next, if a stub resolver resolve “child.example NS”, then child.example NS is overwritten by “child.example NS ns2”
- Second resolve: www.example A”returns 192.0.2.2
- It’s unpredictable and unstable

RFC 2181 5.4.1 Ranking data

- “When considering whether to accept an RRSet in a reply, or retain an RRSet already in its cache instead, a server should consider the relative likely trustworthiness of the various data. An authoritative answer from a reply should replace cached data that had been obtained from additional information in an earlier reply.”
- “Trustworthiness shall be, in order from most to least: “ (next slide)

RFC 2181 Ranking (1)

1. Data from a primary zone file, other than glue data,
2. Data from a zone transfer, other than glue,
3. The authoritative data included in the answer section of an authoritative reply.
4. Data from the authority section of an authoritative answer,
5. Glue from a primary zone, or glue from a zone transfer,
6. (6a) Data from the answer section of a non-authoritative answer, and (6b) non-authoritative data from the answer section of authoritative answers,
7. (7a) Additional information from an authoritative answer,

(7b) Data from the authority section of a non-authoritative answer, (7c)Additional information from non-authoritative answers.

Auth server,referral,glue,answer,cached?,additional, attached NS

RFC 2181 Ranking (2)

- Remove authoritative server data, and rewrite
- Ranking in resolver
  - 3. Authoritative answer that matches a query
  - 4. NS RRSet attached in “3. Answer”
  - 6a. Answer from cache : (miss configuration)
  - 6b. non-authoritative data from the answer section of authoritative answers
    - RRs that is synthesized by CNAME RR
    - CNAME generated by DNAME
  - 7a. Additional information from an authoritative answer
    - A/AAAA RRs that matches MX EXCHANGE field
  - 7b. Referrals
  - 7c. Glue records

Problems of ranking data

- Referrals and glue records are overwritten by “3. Authoritative answer that matches a query” and “4. NS RRSet attached in 3 authoritative answer”.
  - not always occur
  - will break “Referrals and glue records are information to access servers for child zones”
  - Some implementations have complicated code to handle “4. NS RRSet attached in authority section”
  - After the overwrite, name resolution results may be changed
- RFC 2181 is deemed all data (authoritative, non-authoritative, referrals and glue) is merged into one
  - It may be used as an answer from resolvers
  - Non-authoritative data, referrals and glue records SHOULD NOT be used as answers
  - Recent implementations answer authoritative data only

Parent NS and child NS mismatch again

example TLD

child.example NS ns1

Delegation

ns1

child.example zone
child.example NS ns2
www.example A 192.0.2.1

(1) Query: www.example A
Answer:
Authority: child.example NS ns1

7b: Data from the authority section of a non-authoritative answer

(2) Query: www.example A
Answer: www.example A 192.0.2.1
Authority: child.example NS ns2

4. Data from the authority section of an authoritative answer

Resolver cache:
(1)child.example NS ns1

Overwritten by (2)
(2) child.example NS ns2
www.example A 10.0.0.1

Problem happened by overwrite

- CVE-2012-1033: Ghost Domain Names: Revoked Yet Still Resolvable
  - https://kb.isc.org/article/AA-00691
  - BIND and other software affected by this behavior are so affected because of the inherent, longstanding design of the DNS protocol.
  - The attack used updates of NS RRSet attached in authoritative answer.
  - From BIND 9 Changes: 3282. [bug] Restrict the TTL of NS RRset to no more than that of the old NS RRset when replacing it.
    - DNS complexity increased, I think.

Summary of RFC 2181 Ranking data

- Ranking data seems to be written with an assumption that name servers (resolver + authoritative server) have single cache
  - Maybe BIND 8 ?
- Recent implementations have complicated functions
  - Answer authoritative data only
  - Careful check to accept responses from authority section and additional section

New resolver algorithm proposal

New resolver algorithm proposal

- Resolvers answer authoritative data only
  - 3. Authoritative answers that match queries
  - Don’t answer referrals and glue records
- Referrals and glue records are the only information to find name servers for child zones
  - And additional name resolution for out-of-bailiwick name server names
  - Don’t use zone apex NS RRSet for name resolution
  - Glue records are used for corresponding delegation only
- Preloaded zone file are treated as answers from authoritative servers
  - They are treated as static authoritative data, referrals, and glue records

Implementation proposal

- Separate caches
  - Authoritative data cache
  - Referral (+glue) cache
    - SLIST by RFC 1034
  - Needs further considerations for Negative caching, DNSSEC and DNAME

Resolver idea (1)

0. Preload zone files
   - Root hint is imported to referral cache
   - Priming updates the referral cache and authoritative cache
   - Authoritative data from preloaded zones are imported to authoritative data cache
   - Referrals and glue records from preloaded zones are imported to referral cache

Resolver idea (2) Resolving

1. Search QNAME/QTYPE from authoritative cache. If it's valid, answer it.
2. Search the closest encloser of QNAME from the referral cache

- It points referrals and glue records (or error)
- If referrals contain out-of-bailiwick name servers, resolve them

3. send QNAME/QTYPE queries to name servers of the referral

4. analyze the answer

- 4a. If it contains authoritative answer, copy it in authoritative cache and answer it
- 4b. If it contains referrals, copy it in referral cache, goto step 2
- Otherwise: Error handlings

Characteristics of proposal

- This proposal does not change resolver algorithm described in RFC 1034 section 5.3.3, except updates of referrals
- Separated authoritative data (possible to answer) and referrals (used to iterate DNS tree)
- No special order (trustworthiness ranking)
- More stability of name resolution
  - Results of traditional name resolution will flap if NS RRSets are different between the parent and the child
  - First time, referral is used
  - Second time, zone apex NS RRSet may be used
- This algorithm is similar to traditional algorithm when the cache is empty

Issues of proposal

- TTL control of zone apex NS RRSet does not work
  - However, overwrite of the referral does not occur always.
  - TTL control may increase queries to TLD
- Update standards track RFCs

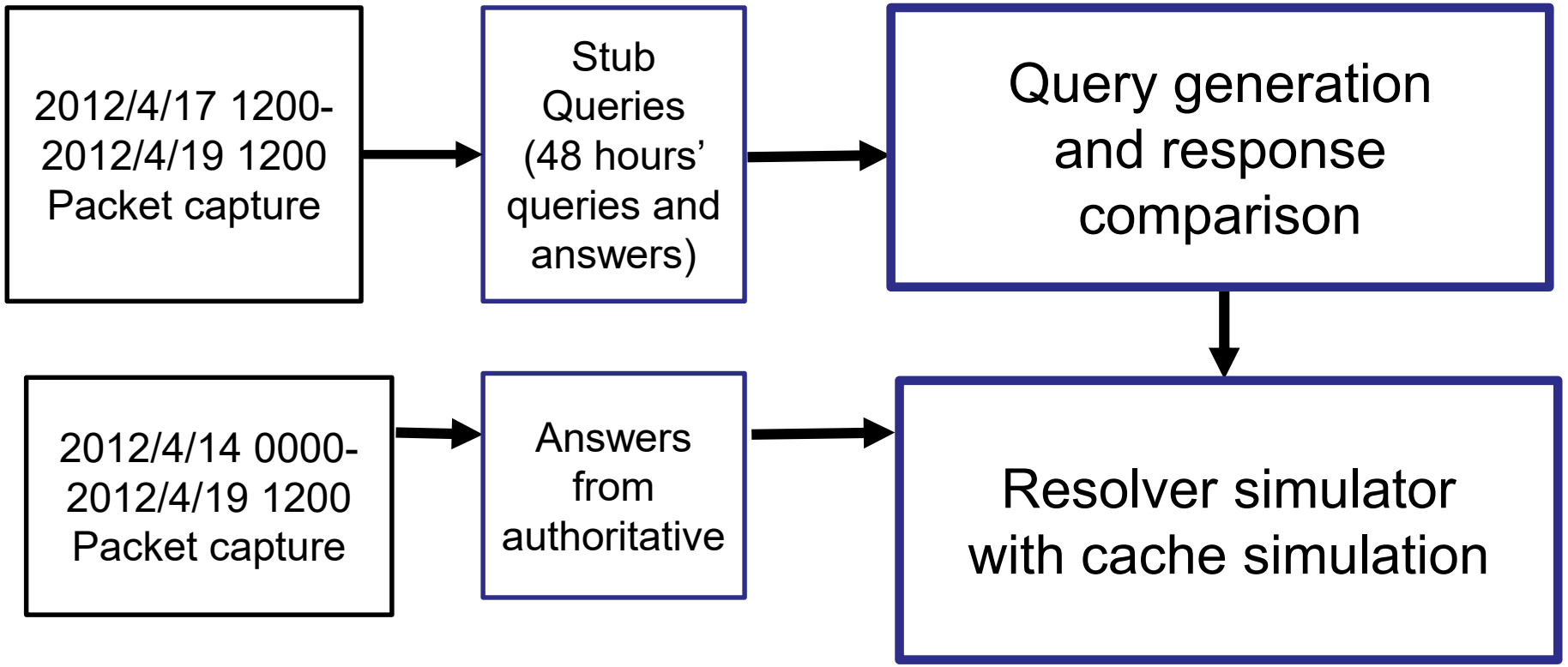
Considerations of root zone

- Root zone is special because it is not delegated
- Root hint and priming are exceptions
  - Because priming replaces preconfigured root hint by root zone apex NS RRSet (authoritative data)

Experiment of the proposal

- Simulation using captured packet
  - Read both answers from authoritative and stub queries
  - For all stub queries, it simulates resolvers (with caches) and output result
  - Compare responses and count authoritative queries
- Input: packet capture around BIND 9 resolver.
  - University of Tsukuba, April (14-)17-19, 2012
  - 28 million client queries / 48 hours (162 qps)
  - 8429 clients
  - 7.3 million answers from authoritative servers

Experiment Environment



Experiment Result (1)

- Do answers match with BIND 9 ?

Number of queries	**28,359,467**	
Matched answers (A,AAAA,PTR)	28,161,142	99.3%
Excuded types (except A,AAAA,PTR)	197,950	0.7%
Unmatched answers (A,AAAA,PTR)	375	0.001%

- Answers from CDN varied. They are merged and treated as matched answer.
- Unmatched answers (375) were answered from another server rewritten by zone apex NS RRSet
- Most of all answers matched with captured stub answers (BIND 9) except unmatched answers.
- Proposed resolver algorithm may work

Experiment Result (2)

- Number of queries to authoritative servers

	To root			To TLD	Auth
		Non-exist	exist		
Captured data (BIND 9)	118,360	12,579	105,781	687,365	6,524,070
Unbound + large cache	13,300	11,444	1,856	870,650	9,102,884
Simulation	13,257	12,234	1,023	350,873	5,542,808

- Proposed algorithm sends smallest number of queries to authoritative servers

Previous implementations

- I heard that some software implemented similar algorithm
  - Nominum
    - https://nominum.com/ghosts-in-the-dns-machine/
  - MaraDNS Deadwood

Effects to DNSSEC

- DNSSEC validates authoritative data
  - No changes to DNSSEC
- DNSSEC does not validate referrals
  - Referrals may be poisoned
    - However, authoritative data retrieved from induced servers are validated by DNSSEC
  - Referral signature may be possible
    - It may be used to validate referrals of unsigned delegations
    - It will be another proposal

Effects to qname minimisation

- No effects because answers from authoritative servers don’t change
- Referral cache and authoritative data cache separation will need small implementation changes

Conclusion and future plan

- RFC 2181 Ranking data may be obsoleted
- Proposed a new resolver algorithm
  - A data trust model without complicated ranking
  - Only use referral + glue records (+ additional name resolution for out-of-bailiwick name server name) to iterate
  - Resolvers return authoritative data only
- Evaluated the proposed algorithm
  - Seems to work in simulation level
- Future plans
  - Submit an internet-draft this month
    - Update RFC 1034 and RFC 2181
  - Do you have interests ?