Optimizing DNS Authority Server Placement

Ning Kong, Guangqing Deng

IETF 90, DNSOP

Background

- DNS system is still in an expanding period
  - Universal deployment of DNSSEC
  - launch of new gTLDs
  - Increase of Internet users
  - Preventing security attacks such as DDOS
- DNS servers are deployed in a distributed manner
  - For better robustness and higher capacity.
    - E.g. 383 root servers are globally deployed on 218 DNS nodes

Problem Statement

- Do we need to assess the current DNS authority server placement scheme?
- If so, what should be the assessment metrics and methods?
- Further, do we need a model to guide the authority server placement for better performance and lower cost?

Use Cases

- Case 1. Geographically distributing all servers for new gTLDs.
- Case 2. Geographically distributing newly-added servers for current TLDs.
- Case 3. Readjusting service area of deployed servers for current TLDs.

Possible Solution (1/2)

- Possible assessment metrics
  - Processing capacity assessment: how quickly a DNS query is processed by one authority server? (processing latency)
    - The processing ability of each server;
    - The number of all servers;
    - The load balancing of all servers
  - Transport capacity assessment : how quickly a DNS query or response is delivered? (transport latency)
    - The network distance between DNS clients and servers.
    - Network congestion

Possible Solution (2/2)

- Possible assessment architecture
  - Input
    - Processing capacity related data
    - Transport capacity related data
    - Deployment cost related data
  - Processing
    - Optimization algorithm
      - High computational complexity: For instance, the computational complexity of selecting 50 locations from 1000 potential locations is about C(1000, 50) which is as large as 10^64!
  - Output
    - The optimized location of each DNS Node
    - The optimized capacity of each DNS Node
    - The optimized service area of each DNS Node

DNS Server Placement Model (1/3)

- Input
  - Processing capacity related data
    - The maximum processing capacity of each server.
    - The capacity of load balancer.
    - The geographical distribution of DNS queries.
  - Transport capacity related data
    - The RTT between potential locations of authority servers and clients.
    - The geographical distribution of DNS queries.
  - Deployment cost related data
    - The bandwidth, electricity, room rent and equipment deployment & maintenance price of each potential location.
    - The price of one authority server and load balancer.
    - The total financial budget.

DNS Server Placement Model (2/3)

- Processing
  - Optimization goal
    - Minimizing DNS query latency when the financial budget is given
      - Average query latency: best efficiency.
      - Maximum query latency: best fairness.
    - Or Minimizing financial cost when the DNS query latency is given?
  - Possible optimization algorithms
    - simulated-annealing Algorithm
    - Genetic Algorithm
    - Others? Or customized ones?

DNS Server Placement Model (3/3)

- Output
  - The optimized location of DNS Node
    - Selected from potential locations.
  - The optimized capacity of DNS Node
    - The number of servers.
    - The needed bandwidth, et al.
  - The optimized service area of each DNS Node
    - The set of clients should be served by each DNS Node.

Simple Example (1/5)

- Input data
  - From one network measurement where there are nearly 60,000 measurement clients and 500 measurement servers in the TOP 21 province (on Internet users) in China from Dec. 12 to 25, 2011.
    - Transport capacity related data
      - RTT between different provinces
      - DNS query rate of each provinces
    - Ignore the processing capacity and deployment cost related data
- Optimization goal
  - Minimizing Average query latency

Simple Example (2/5)

$$\min_{\{x_i\},\{c_{ij}\}} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\tau_{ij} * u_i * x_j * c_{ij} \right) \Big/ \sum_{i=1}^{N} u_i \right\}$$

$$st. \quad \sum_{j=1}^{N} x_j = M$$

$$for\ \forall i, \sum_{j=1}^{N} c_{ij} = 1$$

$$for\ \forall j, \sum_{i=1}^{N} c_{ij} \geq 1$$

- τ_{ij}: RTT between different area
- x_j: Indicate whether the location is selected
- c_{ij}: Indicate serving area
- u_i: Query NO. of different areas
- $\sum_{i=1}^{N} u_i$: Total query NO.
- M: The NO. of DNS node
- $\sum_{j=1}^{N} c_{ij} = 1$: One DNS user send its queries to one server
- $\sum_{i=1}^{N} c_{ij} \geq 1$: Each DNS node should serve at least one user area

Simple Example (3/5)

- Simulated-annealing based algorithm

Given a partition (*or clustering*) $\Omega^k = \{\Omega_1^k, \Omega_2^k, \Omega_3^k, ..., \Omega_M^k\}$ for area set $A = \{a_1, a_2, a_3, ..., a_N\}$

1 $if\ (T > T_{\min})$

2 *With probability of* $x\%$, *choose the cluster with highest average user latency and with probability of* $(1 - x\%)$, *randomly choose a cluster. Denote the finally selected cluster by* α.

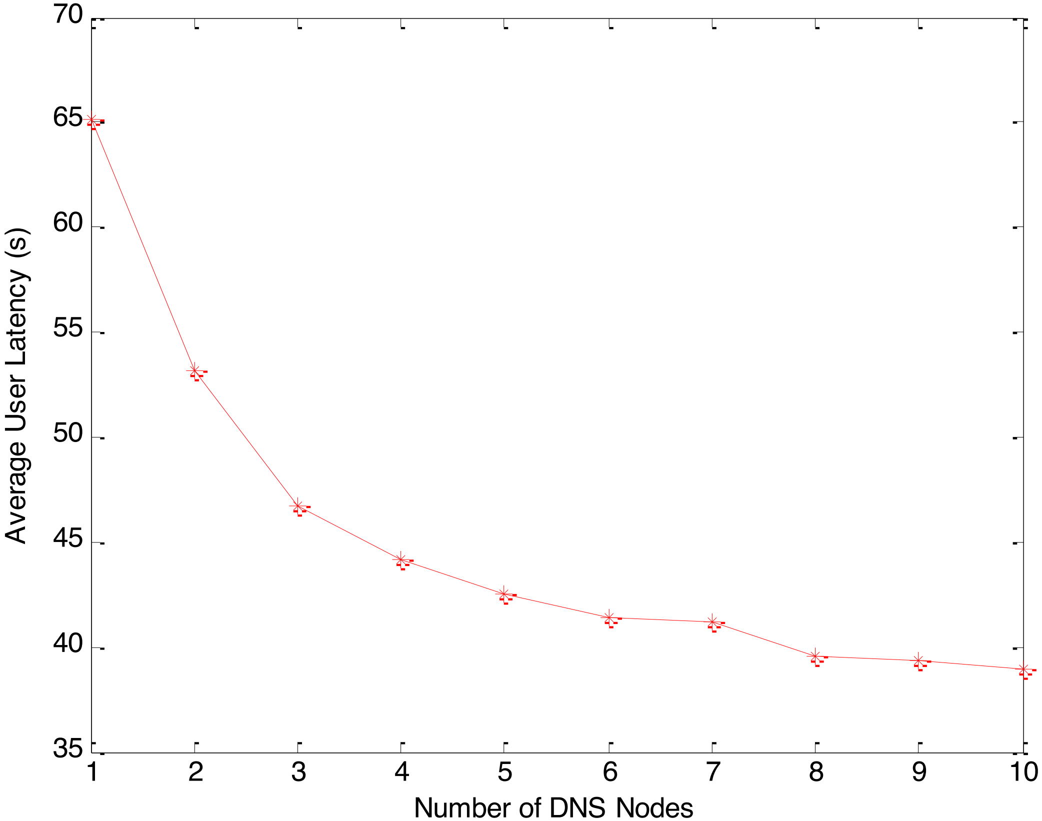
3 *With probability of* $y\%$, *choose the user area experiencing highest user latency in cluster* α *and with probability of* $(1 - y\%)$, *randomly choose an user area in cluster* α. *Assign the selected user area to another randomly chosen cluster and then the* $(k+1)th$ *partition* (Ω^{k+1}) *is formed*!

4 *if (average latency of* Ω^{k+1} (L_{k+1}) *is smaller than that of* $\Omega^k (L_k)$*)* *Replace* Ω^k *by* Ω^{k+1}
else Replace Ω^k *by* Ω^{k+1} *with probability of* $\exp(L_k - L_{k+1})$

5 $T = T * r\%$, *go to* 1.

Simple Example (4/5)

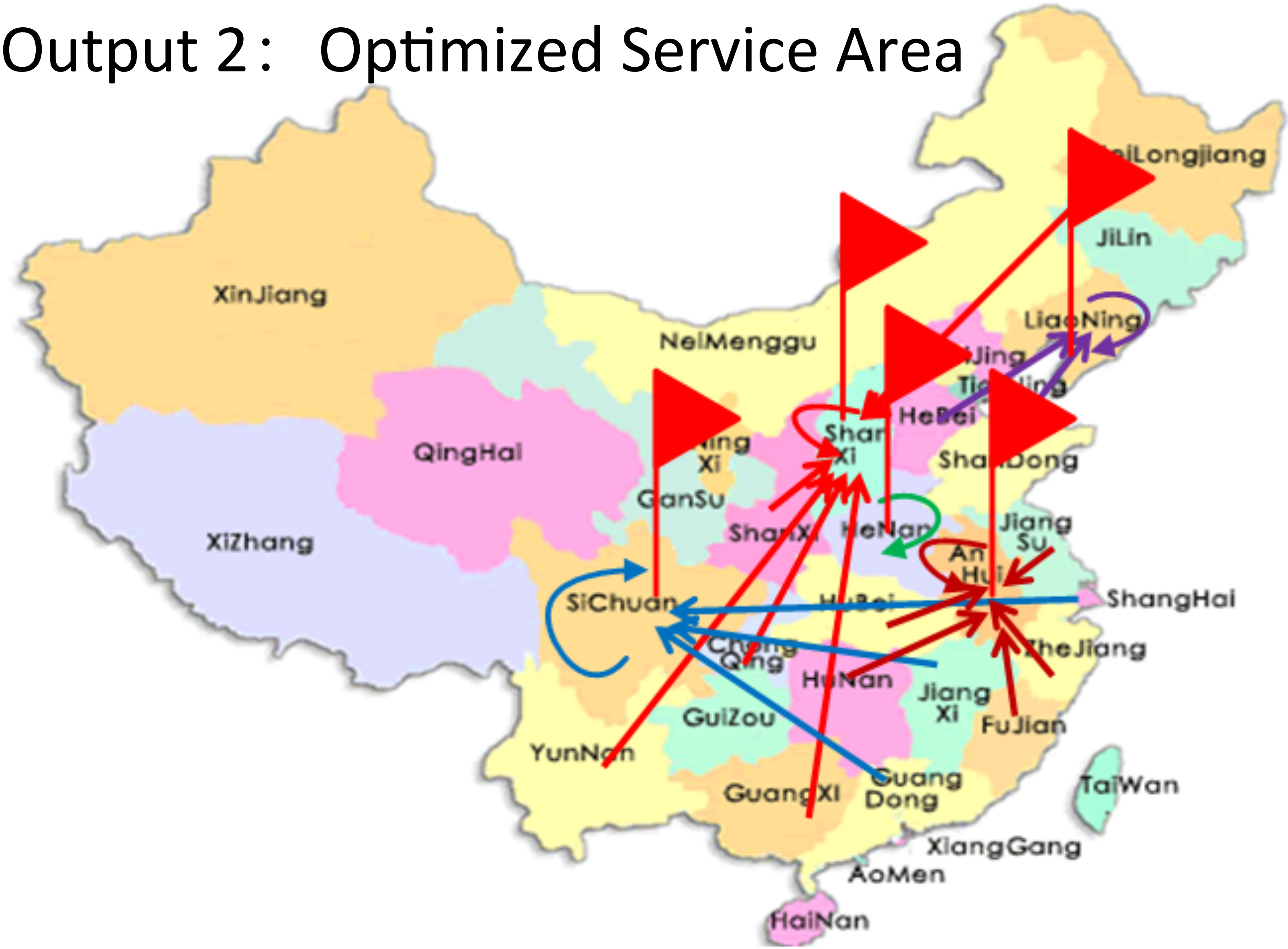
- Output 1： Optimized Locations



Node NO.	Selected location
1	Beijing
2	Beijing, Sichuan
3	Anhui, Liaoning, Sichuan
4	Anhui、Hebei、Henan、Sichuan
5	Anhui, Liaoning, Henan, Shanxi, Sichuan

Simple Example(5/5)

- Output 2： Optimized Service Area



Open Questions

- Is this topic in the scope of DNSOP WG?
- How many people interested in this work?
  - Document a draft
  - Review the draft

Any Comments?

Thanks!