Identifying Influential SLD Authoritative Name Servers on the Internet

Haiyan Xu, Zhaoxin Zhang, Bing Han and Jianen Yan**

School of Cyberspace Science, Faculty of Computing, Harbin Institute of Technology, Harbin, China

DNS plays an important role on the Internet. The addressing of most applications depends on the proper operation of DNS. The root servers and the top-level domain servers are relied upon by many domains on the Internet, and their security affects the whole Internet. As a result, more attention has been paid to the security of servers at these two levels. However, the security of second-level domains and their servers also needs to be brought to the forefront. This paper focuses on showing the complex resolving dependencies and identifying influential name servers for second-level domains. We start by detecting domain name resolution paths and building up a name dependency graph. Then we construct domain name resolution networks of different numbers and sizes, which are connected by a certain number of domain name resolution graphs. On this basis, the network is analyzed from the perspective of complex network analysis, and a multi-indicators node importance evaluation method based on partial order is proposed to identify the influential name servers of the network. Once these name servers are not properly configured and fail or are compromised by DDoS attacks, it will cause resolution failure for a wide range of domain names.

Keywords: DNS, authoritative name servers, name dependency, complex networks, node importance, influential nodes, partial order

OPEN ACCESS

Edited by:
Shudong Li,
Guangzhou University, China

Reviewed by:
José-Fernán Martínez-Ortega,
Polytechnic University of Madrid,
Spain
Shitala Prasad,
*A*STAR Graduate Academy*
*(A*STAR), Singapore*

****Correspondence:***
Zhaoxin Zhang
heart@hit.edu.cn
Jianen Yan
yanjianen@hit.edu.cn

Specialty section:
This article was submitted to
Social Physics,
a section of the journal
Frontiers in Physics

Received: *31 August 2021*
Accepted: *25 October 2021*
Published: *24 November 2021*

Citation:
Xu H, Zhang Z, Han B and Yan J (2021)
Identifying Influential SLD Authoritative
Name Servers on the Internet.
Front. Phys. 9:768205.
doi: 10.3389/fphy.2021.768205

INTRODUCTION

The domain name system (DNS) provides the service of address resolution for most kinds of Internet applications, which transforms the more easily remembered domain name into the actual identification of hosts on the Internet - IP address, and vice versa. DNS is a globally distributed system. To deal with the scale problem, it deploys a large number of domain name servers, which are organized in a hierarchical structure and distributed all over the world. In general, there are three types of DNS servers in the hierarchy: root DNS server, top-level domain (TLD) DNS server, and authoritative name server below the top-level domain, as shown in **Figure 1**.

There is another kind of local DNS server, which is equivalent to a proxy. It is responsible for receiving domain name resolution requests from clients and forwarding them to DNS servers in the hierarchy. The root and TLD servers do not store the mapping information of specific domain names, and this information is stored in the authoritative name servers below the top level. The local DNS server can only obtain the IP address of the corresponding authoritative name server through DNS communication with the root and TLD server. Therefore, the importance of the root and TLD server is self-evident, and its security is also vital. Although there are many distributed denial of service (DDoS) attacks against these two-tier servers, their security has been concerned by many researchers. At the same time, their management and configuration are in charge of professional

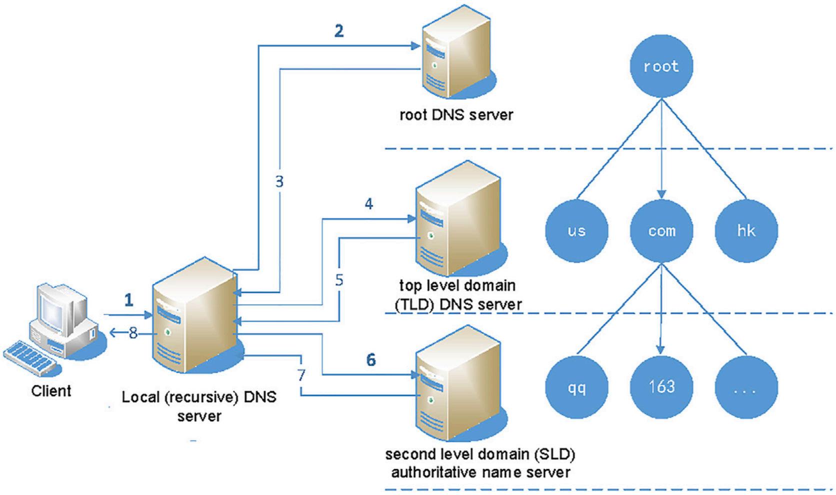


FIGURE 1 | DNS structure and domain name resolution flowchart.

organizations. However, some authoritative name servers at or below the second-level domain (SLD)are self-built, and some are entrusted to the DNS service provider or CDN service providers for trusteeship. There are great differences in operation, maintenance and security capability among service organizations, and the overall service capability is not high.

To ensure the availability of domain name resolution and the correctness of the resolution results, all DNS servers involved in the resolution need to work normally and stably. Even if the services of the root servers and the TLD servers are normal, the authoritative name servers below the TLD will bring risks to the domain name resolution. In recent years, some DDoS attacks have targeted DNS service providers' servers. For example, in October 2019, Amazon's DNS server suffered from a DDoS attack. The attacker sent a huge amount of junk network traffic to the target, resulting in the name resolution service being unable to access, which affected many websites and applications [1]. In October 2016, a DNS service provider encountered a large-scale DDoS attack, which made a large area of websites on the east coast of the United States inaccessible [2].

Moreover, according to the requirements of the DNS protocol specification [3, 4], most SLD administrators configure multiple name servers for domain zone to improve the performance of domain name resolution and distribute the servers in different regions to increase the reliability of domain name resolution. In addition, some large websites delegate resolution services directly to service providers or use services provided by CDNs. All these can make the resolution relationship of a domain very complicated, leading to associations between domains.

Therefore, we detected resolution paths of Alex [5] top one million domain names every month in 2020. According to the monthly survey, 86.18% of domain names involve more than two domains, and some domain name resolution relies on hundreds of name servers. When name resolution spans multiple domains, it will lead to name dependency and make the resolution process more complex. In addition, by analyzing these resolution dependencies, it is found that some SLD authoritative name servers (hereinafter referred to as name servers) provide resolution services for hundreds of domain names. Consequently, to identify influential SLD name servers, a domain name resolution network is built based on real data of a large number of domain name resolution paths data. The main contributions of this paper can be summarized as follows:

- The dependency relationships between domain names and SLD name servers are obtained by probing the resolving paths of domain names, and a name dependency graph model is proposed to express the resolution relationships of a domain name.
- The modeling of the domain name resolution network (DNRN) is proposed, which connects dependency graphs to construct a complex network. The DNRN represents the complex connection of many domain names on the resolution path, which is used to identify influential key domains and SLD name servers from the SLDs of DNS.
- A method for quantitative analysis of network nodes based on workload is proposed, which considers the actual process of workload transfer in the hierarchical resolution chain. The quantized weight of the node can be used as one of the indicators to describe the centrality of the node.
- A multi-indicators node importance evaluation method based on partial order is put forward combined with node workload weight and other classical networks centrality indicators, to identify influential servers from multiple perspectives.

The paper is organized as follows. *Related Work* gives an overview of the related work in this area. *Methods* introduces the methods of identifying influential SLD name servers on the Internet. *Experiments and Analysis* presents experiments and analysis by using the method described in *Methods*. *Conclusion* concludes our analysis.

RELATED WORK

In the aspect of DNS security, there is some significant research. Yehuda Afek et al. [6] et al. observed that the number of packets involved in a typical name resolution process is much larger than theoretically expected, which is mainly due to the extra resolving of name server IP address. Based on this vulnerability, the corresponding attack (NXNSAttack) was constructed, which can be used to launch DDoS attacks against any victims. This paper also measured the popularity of domains with out-of-bailiwick name servers, showing that most of the top one million popular websites have out-of-bailiwick name servers. As for out-of-bailiwick name servers, V. Rama Subramanian et al. [7] first raised this issue and proposed the concept of name dependency, which showed that a typical name depends on an average of 46 servers, while some names depend on hundreds of servers. In addition, it found that 30% of domain names can be hijacked by two servers, both of which have well-known security vulnerabilities. Casey Deccio et al. [8–10] found that more than half of the queries for a domain name were affected by the namespace beyond the control of the domain name owner. Fujiwara et al. [11] measured the growth of DNS traffic, and the results showed that 60% of DNS traffic was generated by out-of-bailiwick name servers.

In the aspect of DNS vulnerability research, Kröhnke et al. [12] studied the impact on the resolution of some domain names when the routers, name servers, and resolvers in the AS failed or the interconnection between ASs failed, which could identify the bottleneck and single point of failure in the network. Abhishta et al. [13] showed the impact of DDoS attacks against DNS service providers and discovered that the number of domain names specifically using a single DNS service provider is decreasing, and the trend of using multiple providers to disperse risk is on the rise.

The above papers studied that DNS faced many resiliency and security issues. In our previous work [14], the impact of the resolution dependence on the DNS was studied by constructing a name resolution network based on large-scale data from a macro perspective. The similarity between these two articles is that they use the same name dependency graph and the name resolution network model. The difference between these two articles is that this paper pays more attention to the optimization of the centrality algorithm as well as the identification of influential name servers from multiple angles, while the previous article analyzed the DNS vulnerability from the perspective of structure.

The research of complex networks is dedicated to finding macro-statistical characteristics and discovering the relationship between structure and function. Many of these studies focus on the survivability of complex networks. How to mine the key nodes of the network to prevent the network from intentional attacks has attracted the attention of many researchers. Linyuan Lü et al. classified and summarized the method of vital nodes identification in complex networks, as well as pointed out that the criteria of important nodes are diverse [15]. Therefore, it is impossible to find a general index that can best quantify the importance of nodes in each case. Sun Peng et al. [16] propose a community-based k-shell decomposition algorithm adapted to a network with a hierarchically ordered structure. This algorithm is superior to other algorithms on networks with community structures. Dong Zhihao [17] proposes a joint nomination strategy that can discover important nodes without global knowledge. This strategy can effectively identify key nodes only using local information and can be implemented in the real world such as the sudden outbreak of covid-19. Shang Qiuyan [18] proposes an effective distance gravitation model based on information fusion and multi-level processing to identify influential nodes. This method can comprehensively consider the global information and local information of complex networks, and use the effective distance to fuse static and dynamic information.

In addition, research on complex networks based on neural networks are also the focus in recent years. Veličković, et al. [19] proposed a graph attention network by introducing attention mechanisms in the propagation process. The attention mechanism assigns different attention coefficients to different neighbor nodes of a node, so that more important nodes can be found. Shudong Li et al. [20] Proposed a community detection algorithm based on a deep sparse autoencoder. In this paper, the unsupervised deep learning method is used to construct a deep sparse encoder, which can obtain a feature matrix with a stronger ability to express network features, and the algorithm can identify the community structure more accurately.

METHODS

Domain Name Resolution Data Acquisition

A large number of DNS servers are deployed in different parts of the world and organized hierarchically. DNS uses a delegation-based architecture for domain name resolution, in which clients follow a set of rules to resolve domain names through multiple name servers, starting with the root, then the TLD, and then the SLD authoritative name server. Following the delegate, a DNS query requires performing additional name resolution to obtain the address of the intermediate name server, and the resolution of each additional name depends on the delegate chain. In summary, the resolution process that follows the chain of delegation induces complex dependencies between name servers. On the Internet, the resolution of many domain names depends on a large number of name servers, and an extremely complex dependency relationship is formed between these domain names and name servers, which can be represented by a complex network.

We perform a real probe of the resolution paths of one million hotspot domain names to obtain the name servers involved in the resolution path of a domain name and the relationship between them. Specifically, we simulate the actual DNS resolution process without considering caching by sending DNS query packets to DNS servers at all levels and getting the address of the server to be queried next from the response packets until the A record for the domain name is obtained. The domains and servers involved in the path and the relationship between them are recorded throughout the process. The resolution path data of a domain name is recorded in XML format, as shown in **Figure 2**. It records all the domains and name servers involved in the resolution process, as well as all the dependencies of a domain name. Since we are mainly concerned with the resolution status below the TLD, the root and TLD servers from the path data are removed from the resolution path. In the probe, follow the following rules:

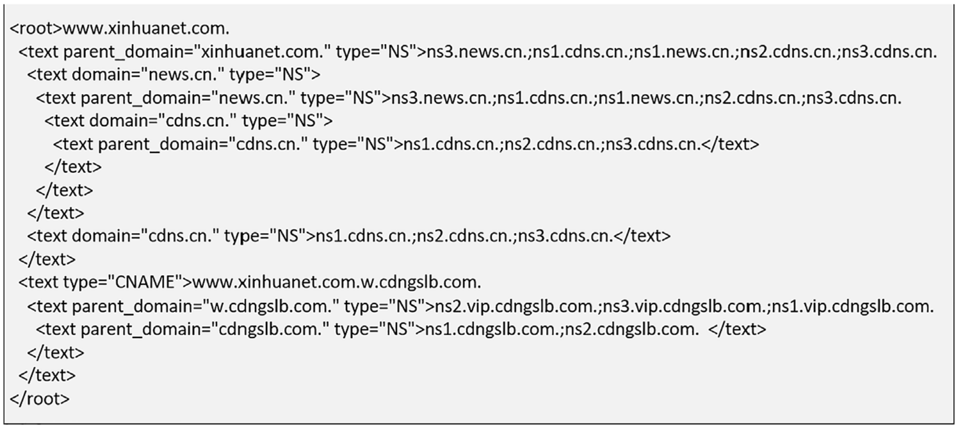
1) Stop detection when cycle dependency occurs. For example, the name server of the domain *A. com* is *ns.B.com*, and the name server of *B. com* is *ns.A.com*. According to the RFC [4], resolving domain *A. com* is needed to submit a DNS query to name server *ns.B.com*, then the DNS server software will first try to get the address of server *ns.B.com* by default. And in the process of resolving server *ns.B.com*, it is needed to query domain *B. com* and server *ns.A.com*, as shown in **Figure 3**. This will produce an endless loop. This is an incorrect way to configure a domain. This problem is discovered in some domains during the process of detecting the domain name resolution data.
2) Stop detection when the domain name servers are self-dependent. For example, the name server of *A. net* is *ns1.A.net*, which does not rely on other domains, so the detection process will stop.

Modeling the Domain Name Resolution Network

Name Dependency Graph Model Based on Resolution Path

To represent the dependencies on the domain name resolution path, a name dependency graph model is built to express the inner logic of domain name resolution. Based on the resolution path data recorded in the above XML file, the main domain name, parent domains, alias, and name servers are extracted as nodes, and the relation between nodes is extracted as edges to form a name dependency graph. A directed edge from a node *u* to a node *v* indicates that node *v* has a dependence on node *u*. According to the practical meaning of domain name resolution, the edges represent three different types of dependencies:

1) Parent domain dependency: The address mapping information of a subdomain is stored by the name servers of its parent domain, which is the basic specification of the DNS. For example, the domain name *www.abc.com* relies on the correct resolution of its upper level, namely the root domain, the TLD (*com*), and the SLD (*abc.com*). In this way, the domain node *www.abc.com* has a directed edge that points to its SLD node *abc. com* in its name dependency graph. Since this paper focuses on the resolution status below the TLD, the root and TLD servers are removed from the name dependency graph.
2) NS dependency: In the DNS specification [4], the address mapping information for domain names administrated by each domain zone is stored in their authoritative name servers (namely NS). Therefore, the dependency of a domain and its name servers is called NS dependency. Generally, a domain is configured with at least two NS servers, but it is also possible to configure multiple geographically dispersed NS servers, even those managed by other domains. The purpose is to improve resolution reliability, but it also brings the issue of resolution complexity. Nowadays, most popular sites rely on DNS or CDN service providers to support professional authoritative zone administration, which also leads to the phenomenon that some NS servers provide services for a large number of domain names. Once these NS servers fail or are attacked, it will affect resolution failures of a wide range of domain names. In this way, a directed edge from the domain node to its NS server node is formed in its name dependency graph.
3) Alias dependency: if an alias (namely Cname) is set for a domain name, the address of the Cname is needed to continue to resolve. Therefore, a domain name node and its alias can form an alias dependency. In this way, a directed edge from the domain name node to its alias node is formed in its name dependency graph.



```
<root>www.xinhuanet.com.
 <text parent_domain="xinhuanet.com." type="NS">ns3.news.cn.;ns1.cdns.cn.;ns1.news.cn.;ns2.cdns.cn.;ns3.cdns.cn.
  <text domain="news.cn." type="NS">
   <text parent_domain="news.cn." type="NS">ns3.news.cn.;ns1.cdns.cn.;ns1.news.cn.;ns2.cdns.cn.;ns3.cdns.cn.
    <text domain="cdns.cn." type="NS">
     <text parent_domain="cdns.cn." type="NS">ns1.cdns.cn.;ns2.cdns.cn.;ns3.cdns.cn.</text>
    </text>
   </text>
  </text>
  <text domain="cdns.cn." type="NS">ns1.cdns.cn.;ns2.cdns.cn.;ns3.cdns.cn.</text>
 </text>
 <text type="CNAME">www.xinhuanet.com.w.cdngslb.com.
  <text parent_domain="w.cdngslb.com." type="NS">ns2.vip.cdngslb.com.;ns3.vip.cdngslb.com.;ns1.vip.cdngslb.com.
   <text parent_domain="cdngslb.com." type="NS">ns1.cdngslb.com.;ns2.cdngslb.com.  </text>
  </text>
 </text>
</root>
```

FIGURE 2 | A XML format for resolution path data of a domain name.

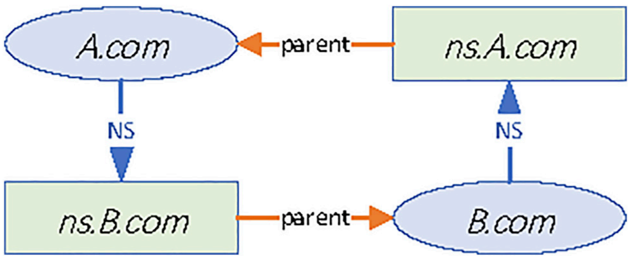


FIGURE 3 | Cycle dependency.

Figure 4 is the name dependency graph of *www.xinhuanet.com*. The ellipse box represents the domain node and the alias node, and the rectangular box represents the server node. The edges between the nodes are directed edges, expressing the above three dependencies. The labels on the edges indicate the type of dependencies.

Moreover, according to whether a NS server is a locally managed server, the NS dependency is further divided into Intra-domain and Inter-domain dependency.

1) Intra-domain dependency: If *v* is a NS server of domain *u*, and is administered by domain *u* itself, then the dependency between *u* and *v* is the Intra-domain dependency, such as the relationship between domain *cdngslb. com* and its server *ns1. cdngslb.com* in **Figure 4**. When the DNS resolver receives this type of resource record (RR), the server's IP address in the additional section of the DNS response packet is used to make the next query.
2) Inter-domain dependency: If *v* is a name server of domain *u*, and is administered by another domain, then the dependency between *u* and *v* is the Inter-domain dependency, such as the relationship between the domain *xinhuanet. com* and its server *ns1. cdns*.cn in **Figure 4**. In this case, there is no IP address of this server in the additional parts of the DNS response packet. So, the DNS resolvers will re-iterate to query the IP address of the out-of-bailiwick server. This leads to cross-domain resolution.

Because of the Inter-domain dependency, the resolution of a domain name involves more domains and more nameservers, which makes the administration and configuration of the SLD complicated.

Construction of Domain Name Resolution Network

Since some NS servers provide services for multiple domains, and the resolution of a domain name is associated with several different domains, a complex network is formed when connecting a certain number of name dependency graphs. This paper refers to this network as DNRN, which contains relationships between domains and name servers. **Figure 5** is an example of a domain name resolution network with 200 nodes, where the node types include domain names and aliases, domains, name servers.

Afterward, by using the statistical characteristics of complex networks, such as degree and aggregation coefficient, this paper analyzes the characteristics and structure of DNS networks and uses node importance analysis methods to identify the key name server nodes. These servers are highly dependent objects, which can affect the resolution of a large number of domain names. These are the focus of protection and should be paid attention to by domain zone administrators and service providers.

Multi-Indicators Node Importance Evaluation Based on Partial Order

In the research of complex networks, it is found that a small number of nodes play a key role [21, 22]. They have an irreplaceable role in network performance, and often determine the structure and function of the network. In the research on the identification of influential nodes of complex networks, the existing centrality algorithms of complex networks can be divided into the following categories according to their properties.

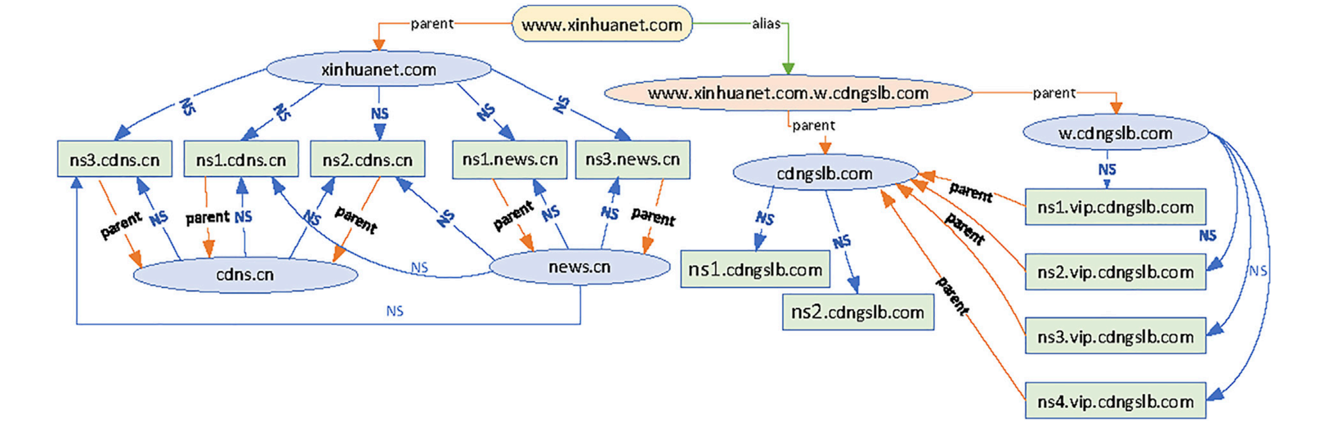


FIGURE 4 | An example of domain name dependency graph.

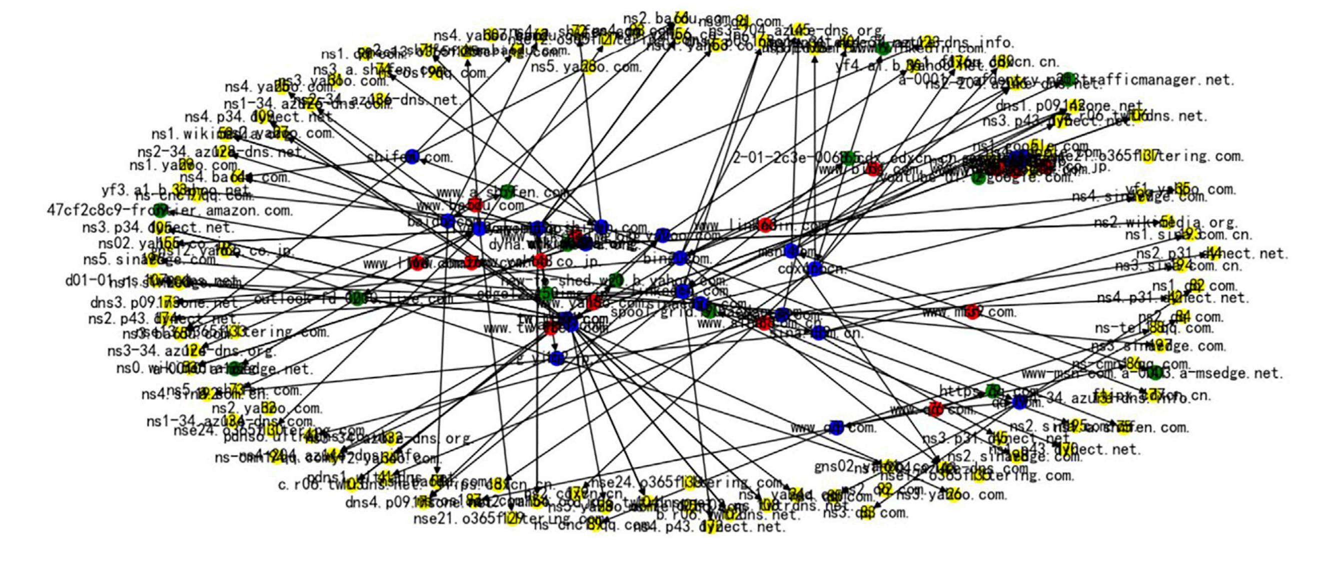


FIGURE 5 | An example of a domain name resolution network with 200 nodes.

Classical Centrality Algorithms for Complex Networks

Neighborhood-Based Method

This method is based on the number of neighboring nodes, e.g., degree centrality.

- In-degree centrality

The DNRN is a directed graph, so the node degree is divided into in-degree and out-degree. The in-degree of a node indicates the dependence of other nodes on it. For a network with *N* nodes, the maximum possible in-degree of each node is *N-1*, so the in-degree centrality of the node is obtained by normalizing *N-1*. For node *i,* its in-degree is k_i^{in}, then the in-degree centrality DC_i is:

$$\boldsymbol{DC_i = \frac{k_i^{in}}{N-1}} \tag{1}$$

Degree centrality metrics are simple, intuitive, and have low computational complexity. The disadvantage of degree centrality metrics is that it only considers the local information of the node, and does not explore the surrounding environment of the node (e.g., the location of the node in the network, higher-order neighbors, etc.) in more detail.

Distance-Based Method

It is based on the shortest distance associated with a node. Examples are betweenness centrality and closeness centrality.

- Closeness centrality

The average distance *d* from node *i* to all other nodes can be obtained by **formula (2)**, where d_{ij} is the shortest distance between node *i* and the other node *j*.

$$\boldsymbol{d_i = \frac{1}{N}\sum\nolimits_{j=1}^{N} d_{ij}} \tag{2}$$

The closeness centrality CC_i of node *i* is equal to the reciprocal of d_i, so the relative importance of node *i* in the network can also be expressed by the relative size of closeness centrality. The formula of CC_i is as follows:

$$\boldsymbol{CC_i = \frac{1}{d_i} = \frac{N}{\sum_{j=1}^{N} d_{ij}}} \tag{3}$$

If the shortest distance from a node to other nodes is very small, then the Closeness centrality CC_i of the node is high. This definition is more geometrically consistent with the concept of centrality than Degree centrality because the smallest average shortest distance to other nodes means that the node is geometrically centered in the graph.

- Betweenness centrality

Betweenness centrality stipulates that if a node appears on the shortest path of all node pairs in the network more frequently, then the node is more important. Its calculation formula is shown in **Eq. 4**, where n_{st} is the number of shortest paths between any node pair *s* and *t* that pass through node *i, and* g_{st} is the total number of shortest paths between any node pair *s* and *t.*

$$\boldsymbol{BC_i = \sum\nolimits_{s\neq i\neq t} \frac{n_{st}^{i}}{g_{st}}} \tag{4}$$

Betweenness centrality indicators perform well in the Internet protocol design, network optimization deployment, and network bottleneck detection. It reflects the influence of nodes on network flow, and through betweenness centrality can accurately identify important nodes with very large traffic in the network. The disadvantage of betweenness centrality is that the computational complexity, and especially in the big data environment, makes it restricted in practical applications.

Neighbor Importance-Based (Value Iteration) Method

The importance of a node depends on the importance of its neighboring nodes, such as the Eigenvector centrality and PageRank centrality. The computation of such algorithms can be performed by iterative algorithms.

- Eigenvector centrality

Eigenvector centrality argues that measuring the importance of a node by the number of neighboring nodes (the degree of the node) is too one-sided and should also focus on the importance of the neighboring nodes. Let EC_i be the importance of node *i*, then we have

$$\boldsymbol{EC_i = c\sum_{j=1}^{N} a_{ij}x_j} \tag{5}$$

where *c* is a constant, $A = (a_{ij})_{N\times N}$ is the adjacency matrix of the network, and $x = (x_{1,}\ x_{2,}\ \ldots\ ,\ x_n)^{\ T}$ is the eigenvector corresponding to the eigenvalue of matrix A. The eigenvector indicator focuses on the interaction between nodes and is ideal for analyzing information propagation problems.

- PageRank centrality

PageRank, also known as page ranking, is a page ranking algorithm used by Google in web search. The basic idea is that the importance of a web page lies not only in the number of web pages pointing to it but also in the quality of the web pages pointing to it.

First, initialize the PageRank value of all nodes, and make sure that the sum of the PageRank values of all nodes is 1. Then, an iterative computation is performed to divide the PageRank value of each node equally among the nodes it points to at every step. Let the out-degree of node j be k_j^{out} at step *n-1*, then each node pointed by node *j* will get a PageRank value $\frac{PR_j(n-1)}{k_j^{out}}$.

Node Location-Based Method

If a network has the characteristics of hierarchical structure, a node in the core layer of the network often has a high influence even if the degree of the node is small.

- Coreness centrality

The coreness of a node can indicate the depth of the node in the core. The concept of coreness is given by Batagelj [21]: given a graph, by removing all the nodes with degrees less than *k* and their corresponding edge, get a remaining sub-graph, called *k-core.* If a node belongs to *k-core* and does not belong to (*k+1*)-core, then it has a coreness of *k.* The coreness of a node is a global property based on the location of the network. Its low computational complexity makes it suitable for large-scale networks.

Node Weights Based on Domain Name Resolution Business Attributes

The domain name resolution network is a business network established based on the domain name resolution path. The nodes participating in domain name resolution are connected to the network through the resolution chain. Combining with the actual domain name resolution process, this paper proposes a centrality algorithm to quantify the weights of each node [14]. For a node *u*, its weight reflects the sum of all other nodes' dependence on *u* in the resolution process, that is, the node weights here represent the resolution load of the node. The algorithm starts from the domain name nodes and follows the resolution chain to traverse each node in the name dependency graph and calculate its weight. The main idea of the algorithm is as follows:

Set the initial value of all domain name nodes to one and the initial value of the remaining nodes to 0. Node weight calculation is an iterative accumulation process. In *Name dependency graph model based on resolution path*, we define three types of edges. According to different edge types, the accumulative way of weight will be different.

1) Parent domain dependency

When a directed edge points from a node *u* to its parent domain node, the weight of the parent domain node will accumulate the weight of *u*. For a parent domain *parent_u*, let *u* represent a subdomain that depends on *parent_u,* and $Weight_u$ represents the weight of *u*. Then the weight of the parent domain *parent_u* in the network is

$$\boldsymbol{Weight_{parent\_u}} = \sum\nolimits_{i\in N} \boldsymbol{Weight_{u_i}} \tag{6}$$

Where *N* is the number of subdomains that depend on the parent domain *parent_u*.

2) NS dependency

Multiple authoritative name servers can be deployed in a domain, and the domain can be resolved as long as one server is running properly. In this article, the load weight of a server node is equal to the weight of this domain divided by the number of name servers deployed in this domain, assuming that each name server has an equal chance of serving. If the server is set up as the authoritative name server for multiple domains, these values passed by each domain are accumulated as the weight for the server. Therefore, the weight of a name server in the network is

$$\boldsymbol{Weight_{name\_server}} = \sum\nolimits_{i\in N_{domains}} \left(\frac{\boldsymbol{Weight}_{\ \boldsymbol{domain\_zone_i}}}{\boldsymbol{N_{server}}} \right) \tag{7}$$

Where *name_server* is the name server node, *domain_zone*$_i$ is the domain where *name_server* is deployed, N_{server} is the number of all name servers in the domain *domain_zone*$_i$, and $N_{domains}$ are the number of domains where *name_server* is deployed.

3) Alias dependency

If a domain name has a resource record of *CNAME* type, i.e., alias, the alias will inherit its weight. Because the resolution of this domain name has been transferred to the resolution of the alias. There is no cumulative calculation for the weight of aliases, so the weight of an alias node is

$$\mathbf{Weight_{alias}} = \mathbf{Weight_u} \tag{8}$$

Where *alias* represents the *CNAME* of a domain name *u*.

Multi-Indicators Node Importance Evaluation

In addition, there are also some other node centrality methods. However, all the above methods have their limitations. If only one of the metrics is used to evaluate the importance of nodes, appears to be ignored the overall characteristics of the network, resulting in inaccurate evaluation results.

To make the evaluation results more objective and accurate, we use multiple indicators to identify influential nodes. In the multi-indicators evaluation, there may be conflicts among multiple indicators, and appropriate rules should be adopted to resolve this problem. One method is to use the weighting method for determining the node importance [22, 23], assigning weight values to each indicator in advance. But, this method has uncertainties and is a partial subjective approach.

Therefore, we choose a more objective way to sort nodes by using partial order relation [24], which outputs the ranks of the node. The nodes in the first rank are more dominant than other nodes in each indicator. According to such a method, nodes are categorized into different ranks. This ensures that the node evaluation is comprehensive and diverse. It can always guarantee that if a node is better than another node in all indicators, then the importance of this node must be higher than that of another node.

In our multi-indicators evaluation based on the partial order, the set of all node characteristics vectors forms a partial order set based on the dominant relationship and the strict partial order relationship. Each characteristic of the node comes from the classic centrality algorithm for complex networks.

The definitions of partial order and dominance are given as follows.

Definition 1. partial order: Let *R* be a relation on a set *A*. If *R* is self-reflexive, antisymmetric, and transitive, then *R* is the partial order relation of set *A*, referred to as partial order, denoted as "≤". For (*a*, *b*) ∈ *R*, it is denoted as *a* "≤" *b*.

For a given node *a* and *b*, $k_i\,(i = 1, 2 \ldots m)$ are the centrality indicators of the node, and *m* is the number of indicators. a_{k_i} denotes the value of node *a* on an indicator k_i.

Definition 2. dominance: Node *a* dominates node *b*, denoted as *a*<*b*, which is a strict partial order on *A*, and the following conditions shall be met:

a)

$$\forall \mathbf{i}:\ \left(\mathbf{a}_{\mathbf{k_i}}{}'' \le {}''\mathbf{b}_{\mathbf{k_i}}\right)$$

b)

$$\exists \mathbf{i}:\ \left(\mathbf{a}_{\mathbf{k_i}}{}'' < {}''\mathbf{b}_{\mathbf{k_i}}\right)$$

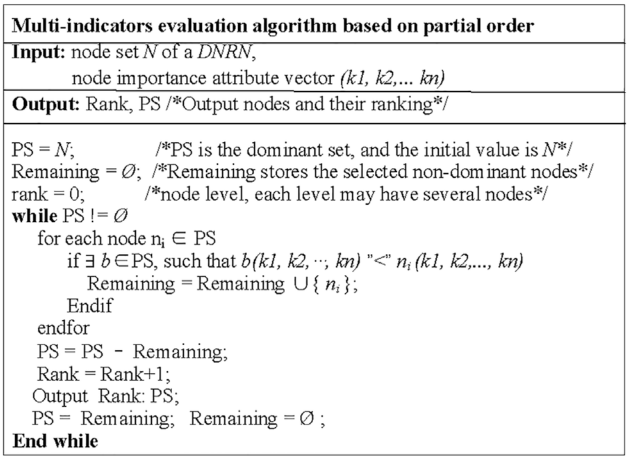
Definition 3. dominance set: For a given set *A*, the priority set (*PS*) contains all nodes that are not dominated by other nodes, such as

$$\boldsymbol{PS} = \left\{\boldsymbol{u} \middle| \exists \boldsymbol{a} \in \boldsymbol{A}\colon\ \boldsymbol{a}'' < {}''\boldsymbol{u}\right\}$$

A node is said to be dominant if each of its metrics is greater than the metrics of other nodes. These dominant nodes are removed from the node-set to be sorted and their node ranks are output. The comparison of the vectors continues until all nodes are output. This process is an iterative process to obtain equivalence classes, and the pseudo-code representation of the partial sorting algorithm is in **Figure 6**. Suppose each node has *n* attributes illustrated from different perspectives, denoted by vector *f* (k_1, k_2, . . . , k_n). In each iteration, the nodes that are dominant in each indicator are searched for to obtain the dominant set. Each iteration process outputs the rank and the nodes that belong to this rank.

Figure 7 is a simple example to illustrate the correctness of the algorithm. The input is vectors of seven nodes, and the vector contains four attributes, i.e. k_1, k_2, k_3, k_4. The output is a partial sequence of nodes. In the first iteration of the algorithm, each component of node "*3*" is greater than the other nodes, then "*3*" dominates the remaining nodes. Next, in each round, the dominant node is selected among the remaining nodes, until the remaining nodes (0, 2, 4) do not dominate the other nodes.

Under the partial ordering, there may be multiple nodes at the same rank. This algorithm is adapted to identify nodes that



```
Multi-indicators evaluation algorithm based on partial order
Input: node set N of a DNRN,
        node importance attribute vector (k1, k2,... kn)
Output: Rank, PS /*Output nodes and their ranking*/

PS = N;              /*PS is the dominant set, and the initial value is N*/
Remaining = Ø;  /*Remaining stores the selected non-dominant nodes*/
rank = 0;            /*node level, each level may have several nodes*/
while PS != Ø
    for each node $n_i$ ∈ PS
        if ∃ b ∈ PS, such that b(k1, k2, ··, kn) "<" $n_i$ (k1, k2,..., kn)
            Remaining = Remaining ∪ { $n_i$ };
        Endif
    endfor
    PS = PS - Remaining;
    Rank = Rank+1;
   Output Rank: PS;
   PS = Remaining;  Remaining = Ø ;
End while
```

FIGURE 6 | multi-indicators evaluation algorithm based on partial order.

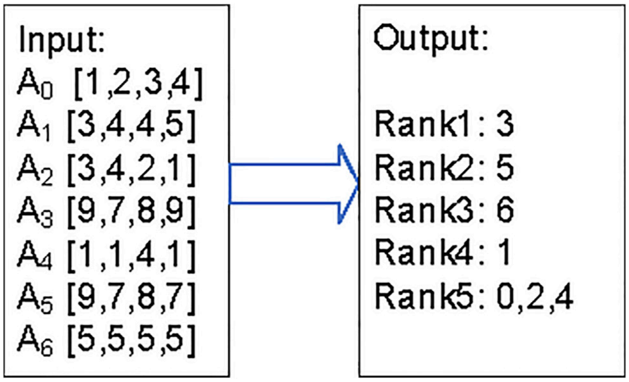


FIGURE 7 | a simple example to illustrate the multi-indicators evaluation algorithm.

perform well in all aspects. When the number of attributes in the vector is large, the number of ranks will be small, implying a higher number of nodes of the same rank, which affects the discrimination and identification of nodes. So, in practice, the selection of attributes is the key issue, and we need to select the relevant attributes according to the actual needs. It must be emphasized that if the selected indicators are different, the order relationship of nodes will be different.

EXPERIMENTS AND ANALYSIS

Test Data Set

We construct a resolution network DNRN for every 100,000 domain names selected in the order of the ranking of one million domain names, and a total of four networks (denoted as *N1, N2, N3, N4* respectively) have been constructed in this way. The domain name sets of these four networks are independent and have no intersection. The goal is to identify influential key nodes from a network. Of course, there is also the option to build a larger network. The number of 100,000 is chosen here as a compromise after considering experimental computing power. Moreover, the four networks are established for verification and comparison.

The DNRN is formed by the correlations among name dependency graphs of 100,000 domain names. In addition, the orphan domain name and small connected components are removed from it, and the maximum connected component is retained. Thus, *N1* to *N4* is a directed connected graph respectively.

The overall characteristics of these four networks, including the number of nodes, the number of edges, the degree-related features, the clustering coefficient, and the number of domain names contained in the network, are shown in **Table 1**. To compare the characteristics of each network, **Figure 8** shows a cross-sectional comparison of the data in **Table 1** for each feature, and it can be seen that the characteristics of the four networks are similar, as shown below:

1) In terms of the size of nodes and edges, *N1* is the largest. The other three networks are similar in scale and smaller than *N1.*
2) From the ratio of *M/N*, the four networks are extremely sparse.
3) The maximum degree of the four networks is above 1,000, but the average degree is around 8.
4) Nodes with the degree of one account for around 26% of the total number of nodes in *N1,* and the other three networks have roughly the same value, around 14%. This statistical value shows the number of edge nodes of the networks and allows us to foresee the overall structure of the network.
5) The average cluster coefficients (ACCs) of the four networks are 0.0016 and 0.026. ACC is used to characterize the robustness and redundancy of the network. If the ACC of a network is higher, the less chance it may be disconnected.

TABLE 1 | Characteristic parameters of *N1~N4.*

notation	***meaning***	***N1***	***N2***	***N3***	***N4***
N	number of nodes	237,848	210,526	210,092	212,385
M	number of edges	433,665	348,131	348,185	352,631
M/N	ratio of edges to nodes	2.206	1.654	1.657	1.660
max{d}	max degree	1,066	1,335	1,299	1,268
<d>	average degree	8.9	8.2	8.1	8.1
Pr [d = 1]	proportion of nodes with a degree of 1	26.07%	14.38%	14.20%	14.78%
<c>	average cluster coefficient	0.0026	0.0017	0.0021	0.0016
<domain names>	number of domain names	81,627	74,613	74,302	75,034

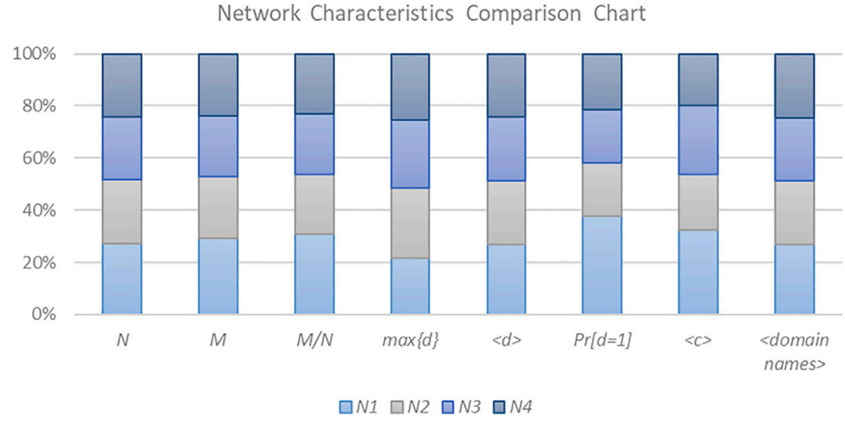


FIGURE 8 | Comparison of characteristic parameters of *N1-N4.*

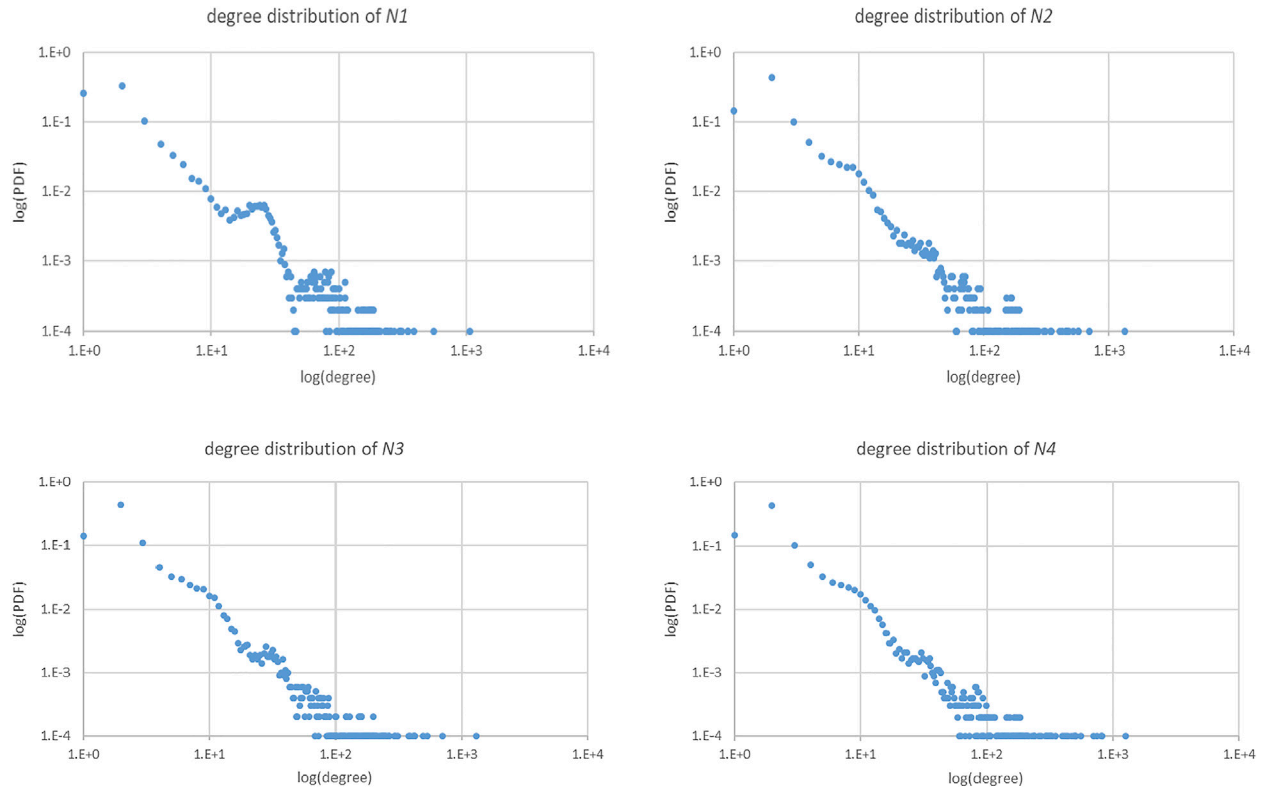


FIGURE 9 | Degree distribution of *N1~N4.*

Judging from the ACCs of the four networks, it also shows that network connections are relatively sparse.

6) *N1~N4* are the largest connected components in the networks formed by name dependency graphs of 100,000 domain names respectively. The number of domain names contained in network *N1* is 81,627, namely, the number of domain names included in the largest connected component is about 81.6% of the number of domain names included in the initial network construction. And *N2~N4* each contains nearly around 75%.

The general characteristics of the networks summarized above show that all four networks are sparse and extremely heterogeneous. In addition, degree distribution *P(k)* is the ratio of the number of nodes with degree value k to the total number of nodes, which is an important measure to understand the overall structure of the network. The degree distribution of the four DNRNs is shown in **Figure 9**. The abscissa of the distribution graph represents the degree k of the nodes, and the ordinate represents the proportion of the number of nodes with degree k in the total number of nodes. The graph is displayed using double logarithmic coordinates. From the shape of the distribution curve, it shows that the distribution trends of the four networks are similar. The distribution has no peaks and presents an approximate diagonal line. The long-tail feature on the side with a higher k indicates that a few nodes have a large number of connections. In summary, from the distribution graph and its analysis, the DNRN has approximate scale-free network characteristics. For networks with such attributes, the existence of some ultra-high connectivity nodes greatly reduces the robustness of the network.

Identifying Influential SLD Authoritative Name Servers

We identify influential SLD authoritative name servers on the Internet by building DNRNs and the multi-indicator evaluation

TABLE 2 | Four kinds of node centrality attributes and partial order rules.

No	Node Centrality attributes	Categories	Partial order rules
k1	In-degree	Neighborhood-based	If In-degree(a)≥ In-degree(b), then a≤b
k2	Weight	name resolution business attributes	If Weight(a)≥ Weight(b), then a≤b
k3	PageRank	Value Iteration-based	If Eigenvector(a)≥Eigenvector(b), then a≤b
k4	Closeness	Distance-based	If Closeness(a)≥ Closeness(b), then a≤b

TABLE 3 | The partial order results of the four indicators (k1, k2, k3, k4) in *N1*.

Partial_sort_rank (k1, k2, k3, k4)	Number of all nodes	Number of server nodes
0	4	0
1	5	0
2	17	9
3	19	8
4	32	21
5	20	9
6	36	23
7	37	25
8	18	10
9	63	50

algorithm is used to identify influential nodes. Therefore, it is necessary to calculate the characteristics of a node as an indicator vector and input to the algorithm. We comprehensively evaluate the nodes to select the influential nodes in four aspects: node connectivity, node business attributes, node neighbor importance, and network structure centrality. Therefore, we choose the four centrality attributes, and define them into four rules, as shown in **Table 2**. The reason for selecting these four indicators is to comprehensively consider the different angles represented by each indicator and the operating efficiency of the algorithm on large networks. It must be emphasized that if the selected indicators are different, the order relationship of nodes will be different.

We use the above four centrality indicators to sort the network nodes in the partial order. There are 421 levels in the sorted result. **Table 3** shows the number of all nodes and the number of server nodes in the top 10 ranks (There are four different types of nodes in the network, including the main domain name nodes, domain nodes, alias nodes, and server nodes). In the first five ranks, there are 36 server nodes, and **Table 4** shows the detailed information of these influential SLD name servers. They all belong to DNS service providers. From **Table 4**, we can see the classification statistics based on some keywords of server names, divided into six categories, which can be considered as the six companies to which the

TABLE 4 | The influential SLD name servers of *N1* in the first five ranks.

name_key_word	Number	server_name (Pronoun)	In_degree	Weight	PageRank	Closeness
akam/ akamaiedge	21	*A1.server*	30	280.3	1.5919E-04	1.3266E-02
		A2.server	182	825.8	8.9228E-04	1.3031E-02
		A3.server	14	278.2	1.5575E-04	1.3180E-02
		A4.server	164	816.5	8.8081E-04	1.2862E-02
		A5.server	155	279.6	1.7973E-04	1.2726E-02
		A6.server	1	201.7	1.3228E-04	1.3116E-02
		A7.server	1	201.7	1.3228E-04	1.3116E-02
		A8.server	1	201.7	1.3228E-04	1.3116E-02
		A9.server	1	201.7	1.3228E-04	1.3116E-02
		A10.server	1	201.7	1.3228E-04	1.3116E-02
		A11.server	1	201.7	1.3228E-04	1.3116E-02
		A12.server	1	201.7	1.3228E-04	1.3116E-02
		A13.server	1	201.7	1.3228E-04	1.3116E-02
		A14.server	1	201.7	1.3228E-04	1.3116E-02
		A15.server	140	279.4	1.7983E-04	1.2695E-02
		A16.server	1	201.7	1.3228E-04	1.3116E-02
		A17.server	1	201.7	1.3228E-04	1.3116E-02
		A18.server	1	201.7	1.3228E-04	1.3116E-02
		A20.server	1	201.7	1.3228E-04	1.3116E-02
		A21.server	129	810.8	8.6999E-04	1.2744E-02
		A22.server	107	810.8	8.7061E-04	1.2721E-02
Cloudflare	5	*C1.server*	1	0.4	9.7556E-03	4.4182E-02
		C2.server	1	0.4	9.7556E-03	4.4182E-02
		C3.server	1	0.4	9.7556E-03	4.4182E-02
		C4.server	1	0.4	9.7556E-03	4.4182E-02
		C5.server	1	0.4	9.7556E-03	4.4182E-02
Parklogic	3	*P1.server*	1	1,249.5	1.6076E-03	3.0102E-03
		P2.server	1,064	427.1	7.5897E-04	5.9480E-03
		P3.server	1,065	427.6	7.5985E-04	5.9536E-03
Cscdns	3	*CS1.server*	227	1,133.1	1.3240E-03	2.7459E-03
		CS2.server	348	305.0	9.6762E-04	2.3114E-03
		CS3.server	348	305.0	9.6762E-04	2.3114E-03
dnsv2	2	*DV1.server*	1	741.7	1.2498E-03	2.2557E-03
		DV2.server	2	1974.6	1.9538E-03	3.1912E-03
Dnspod	2	*DP1.server*	1,065	581.9	9.5876E-04	5.8287E-03
		DP2.server	1,065	581.9	9.5876E-04	5.8287E-03

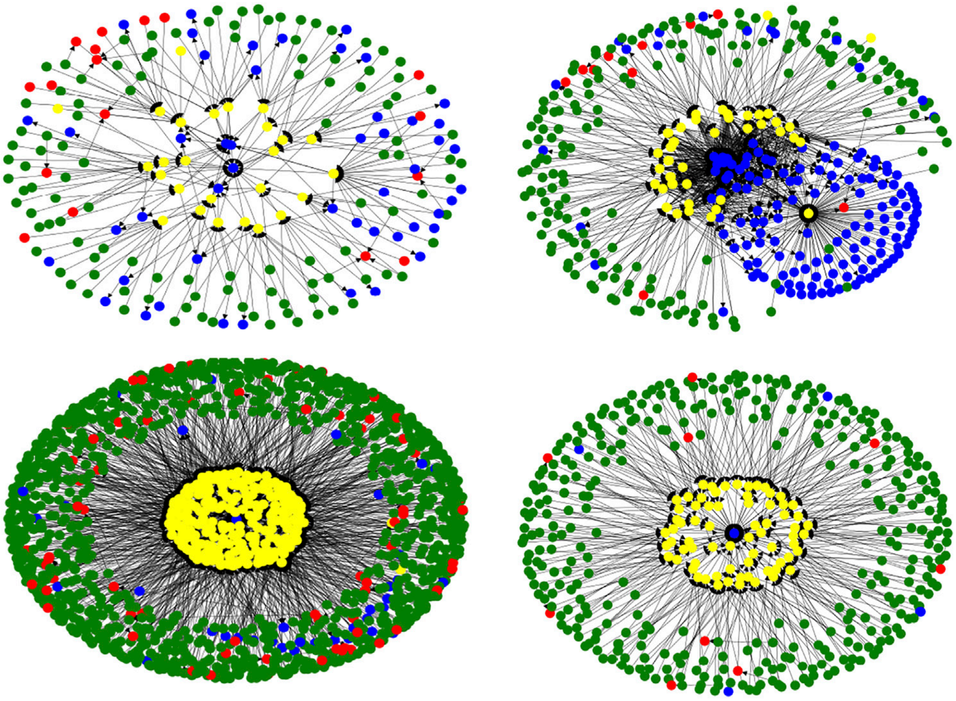


FIGURE 10 | network connections around four key name servers.

TABLE 5 | The partial order results of three sort combinations in *N1*.

Indicators	Number of ranks	Number of all nodes/server nodes in the top rank 3	Number of all nodes/server nodes in the top rank 5	Number of all nodes/server nodes in the top rank 10
k1, k2, k3, k4	421	26/9	77/38	251/155
k1, k2, k3	760	23/7	54/16	153/76
k1, k2	985	3/0	9/1	40/16

server belongs. The real names of the servers are hidden for security and privacy protection purposes. As can be seen from the four index values in the table, the results of the partial order sorting ensure the diversity of the nodes. The importance of such servers is self-evident, and their configuration and administration need to be highly valued. **Figure 10** shows the network connections around four key name servers respectively. For display purposes, some of the similar nodes with the same connections are merged in the figure. Different colors indicate different types of nodes: blue indicates server nodes, red indicates alias nodes, yellow indicates domain nodes, and green indicates primary domain name nodes. We can further develop easier-to-see visualization tools to zoom in on the connections around the core nodes so that administrators can easily check their configuration and security status.

This sort of ranking allows us to see the comprehensive evaluation of each node in the four aspects. It examines the comprehensive characteristics of the node and avoids the one-sidedness of a single indicator, but it also brings the disadvantage of weakening the advantage of the single feature of nodes.

The partial ordering of four indicators may have a coarser granularity, that is, there are multiple nodes at the same level. We can scale down the ranking metrics on this basis, leaving the more concerned node feature metrics to be ranked again. For comparison, several sets of ranking tests are conducted on the *N1* network. The ranking output and the number of top nodes are shown in **Table 5**. As the ranking index decreases, the number of output levels increases.

In practice, users can select multiple indicators that they consider important to sort the nodes in a partial order, and finally classify the nodes according to their ranks. The nodes ranked in the first few levels dominate other nodes in all indicators. If more indicators are selected, fewer ranks may be output, i.e., the nodes are roughly classified. On this basis, the number of indicators can be reduced and then re-ordered, so that each node can be better distinguished.

CONCLUSION

This paper studies the resolution name dependency of the SLD in DNS and concludes that the security and robustness of the SLD are as important as the root and TLD. There are also influential servers in the SLDs, and they serve many domain names. Failure or compromise of the key servers will affect the normal resolution of a large number of domain names. Based on the actual detected domain name resolution data, this paper constructs domain name resolution networks and uses a multi-indicators node importance evaluation method based on partial order to identify influential servers and domains. This method can combine multiple centrality indicators to evaluate the comprehensive characteristics of nodes. This article has built several networks and combinations of multiple indicators to verify the effectiveness of the method [25, 26].

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

This work is supported by the National Science and Technology Major Project of China (Nos. 2018YFB0804703) and the Natural Science Foundation of Shandong Province of China (Nos. ZR2020KF009).

ACKNOWLEDGMENTS

Thanks to the Network and Information Security Technology Research Center in Harbin Institute of Technology, Weihai for the technical support.

REFERENCES

1. The Register. Bezos DDoS'd: Amazon Web Services' DNS Systems Knackered by Hours-Long Cyber-Attack (2021). Available at: https://www.theregister.com/2019/10/22/aws_dns_ddos/ (Accessed August 20, 2021).
2. Threatpost. Mirai-Fueled IoT Botnet behind DDoS Attacks on DNS Providers (2021) Available at: https://threatpost.com/mirai-fueled-iot-botnet-behind-ddos-attacks-on-dns-providers/121475/ (Accessed August 20, 2021).
3. Mockapetris P. *Domain Names - Concepts and Facilities*. USA: IETF (1987). RFC 1034.
4. Mockapetris P. *Domain Names - Implementation and Specification*. USA: IETF (1987). RFC 1035.
5. Alex (2020). Available at: https://www.alexa.com/topsites (Accessed August 1, 2020).
6. Afek Y, Bremler-Barr A, Shafir L. NXNSAttack: Recursive DNS Inefficiencies and Vulnerabilities. In: Proceedings of the 29th USENIX Security Symposium; Aug 12–14, 2020. ELECTR NETWORK: USENIX Association (2020). p. 631–48.
7. Ramasubramanian V, Sirer EG. Perils of Transitive Trust in the Domain Name System. In: Proceedings of Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement; Oct 19-21, 2005; Berkeley, CA, USA. Berkeley: USENIX Association (2005). p. 379–84. doi:10.1145/1330107.1330152
8. Deccio C, Sedayao J, Mohapatra P. Measuring Availability in the Domain Name System. In: Proceedings of 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies; Mar 15-19,2010; San Diego, CA, USA. NY: IEEE (2010). p. 76–80. doi:10.1109/INFCOM.2010.5462270
9. Deccio C, Sedayao J, Mohapatra P. Quantifying DNS Namespace Influence. *Computer Networks* (2012) 56(2):780–94. doi:10.1016/j.comnet.2011.11.005
10. Deccio C, Chen C-C, Mohapatra P, Sedayao J, Kant K. Quality of Name Resolution in the Domain Name System. In: Proceedings of the 17th IEEE International Conference on Network Protocols; Oct 13-16, 2009; Plainsboro, NJ, USA. NY: IEEE (2009). p. 113–22. doi:10.1109/ICNP.2009.5339693
11. Fujiwara K, Sato A, Yoshida K. DNS Traffic Analysis: Issues of IPv6 and CDN. In: Proceedings of the 12th IEEE/IPSJ International Symposium on Applications and the Internet (SAINT)/IEEE Annual Signature Conference on Computer Software and Applications; Jul 16-20, 2012; Izmir, TURKEY. NY: IEEE (2012). p. 129–37. doi:10.1109/SAINT.2012.26
12. Lars K, Jansen J, Vranken H. Resilience of the Domain Name System: A Case Study of the .Nl-Domain. *Comp Networks* (2018) 139-5:136–50. doi:10.1016/j.comnet.2018.04.015
13. Abhishta RVR, Nieuwenhuis LJM. Measuring the Impact of a Successful DDoS Attack on the Customer Behaviour of Managed DNS Service Providers. *Comp Commun Rev* (2018) 48-5:70–6. doi:10.1145/3229598.3229599
14. Xu H, Zhang Z, Yan J, Ma X. Evaluating the Impact of Name Resolution Dependence on the DNS. *Security Commun Networks* (2019) 2019:1–12. doi:10.1155/2019/8565397
15. Lü L, Chen D, Ren X-L, Zhang Q-M, Zhang Y-C, Zhou T. Vital Nodes Identification in Complex Networks. *Phys Rep* (2016) 650:1–63. doi:10.1016/j.physrep.2016.06.007
16. Sun PG, Miao Q, Staab S. Community-based K-Shell Decomposition for Identifying Influential Spreaders. *Pattern Recognition* (2021) 120:108130. doi:10.1016/j.patcog.2021.108130
17. Dong Z, Chen Y, Tricco TS, Li C, Hu T. Hunting for Vital Nodes in Complex Networks Using Local Information. *Sci Rep* (2021) 11(1):11. doi:10.1038/s41598-021-88692-9
18. Shang Q, Deng Y, Cheong KH. Identifying Influential Nodes in Complex Networks: Effective Distance Gravity Model. *Inf Sci* (2021) 577. 162–79. doi:10.1016/j.ins.2021.01.053
19. Veličković P, Cucurull G, Casanova A, Romero A, Pietro L, Bengio Y. *Graph Attention Networks [Preprint]*. arXiv:1710.10903 (2017). Available at: https://arxiv.org/abs/1710.10903v3 (Accessed June 15, 2021).
20. Li S, Jiang L, Wu X, Han W, Zhao D, Wang Z. A Weighted Network Community Detection Algorithm Based on Deep Learning. *Appl Maths Comput* (2021) 401:126012. doi:10.1016/j.amc.2021.126012

21. Yang H, An S. Critical Nodes Identification in Complex Networks. *Symmetry* (2020) 12(1):123. doi:10.3390/sym12010123
22. Crucitti P, Latora V, Marchiori M, Rapisarda A. Error and Attack Tolerance of Complex Networks. *Physica A: Stat Mech its Appl* (2004) 340-1:388–94. doi:10.1016/j.physa.2004.04.031
23. Batagelj V, Zaversnik M. An O(m) Algorithm for Cores Decomposition of Networks. *Comp Sci* (2003) 1(6):34–7. doi:10.1007/BF01074693
24. Hui Y, Liu Z, Li Y. Key Nodes in Complex Networks Identified by Multi-Attribute Decision-Making Method. *Acta Phys Sin* (2013) 62(2):46–54. doi:10.7498/aps.62.020204
25. Tian B, Hu J, Deng Y. Identifying Influential Nodes in Complex Networks Based on AHP. *Physica A: Stat Mech its Appl* (2017) 479:422–36. doi:10.1016/j.physa.2017.02.085
26. Zheng B, Li D, Chen G, Du W, Wang J. *Ranking the Importance of Nodes of Complex Networks by the Equivalence Classes Approach*. arXiv:1211.5484 (2012) Available at: https://arxiv.org/abs/1211.5484 (Accessed June 15, 2021).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.