A Novel Data-Oriented Name Service

Hengkui Wu, Deyun Gao, Dong Yang and Hongke Zhang
National Engineering Lab for Next Generation Internet Interconnection Devices,
School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 10044, China
Email: {06111021, gaody, dyang, hkzhang}@bjtu.edu.cn

***Abstract*—The Internet has evolved from its original design. (1)The user cares about what they are looking for, not which machine provides the data or the service. The host-to-host Internet tends to become a data-oriented network. (2) In the pressures of commerce and security, the middleboxes, such as network address translators (NATs), firewalls and caching servers, become commonplace in current Internet. They improve the performance of the applications and security, but also violate the Internet layering and are difficult to maintain and configure.**

To adapt these changes, we proposed the novel Data-Oriented Name Service (DONS), which gives a clean-slate redesign of Internet naming and name resolution. It achieves (1) permanence, getting the resource using the persistent name; (2) semantic-free, improving the flexibility and functionality of the network; (3) middlebox integration, allowing and facilitating the deployment of middleboxes; (4) self-certifying, authenticating the source.

***Index Terms*—Network architecture, naming, name resolution, data-oriented.**

I. Introduction

The Internet design problem emerged in 1970s for the first time, and then emerged in 1990s. Now this problem has been a hot topic in IETF mailing lists again. Out of question, the Internet has gained a remarkable success. With the development of Internet, kinds of new applications emerge and Middleboxes, such as network address translators (NATs), firewalls and so on, become commonplcace. The Internet has evolved from its original incarnation. Some open questions arise.

Data-oriented or host-centric? Data retrieval and service access are the major usages of current Internet. With the development of Internet, the copy and the migrant of data become common. The user just cares about getting the resource, no matter who provides the resource. Based on DNS in current Internet, when a web page moves from one site to another, the hyper link fails, which is known as the broken link. HTTP redirection and dynamic DNS are used to minimize this problem, but they are not efficient enough.

Middleboxes violate Internet layering. The original Internet is based on the end-to-end principle. The core network forwards the packets without inspecting, filtering or modifying them. The service is carried out at the edge of the network. This open and transparent design makes it easy to deploy a new application in the Internet. It is an important factor to achieve today's remarkable success of the Internet. With the development, Internet has been a huge industry. Under the pressures of commerce and security, the Internet structure is changing. Middleboxes and kinds of policy accesses are now commonplace. It is the result of struggle among the user, the Internet Service Provider(ISP) and the society. Any genius design cannot avoid this struggle among these three. Middleboxes bring the benefit in commerce and security, but also violate the Internet layering.

In this paper, we propose the novel Data-Oriented Name Service (DONS), which gives a clean-slate redesign of Internet naming and name resolution. We take the following as the goals to design our novel name service.

- **Permanence.** Once given a name of the resource, the user can get it unless it does not exist in the network anymore. There is a counter example in current Internet using DNS. Every web page has a universal resource locator (URL). When the page moves from one web site to another web site, or from one file folder to another file folder in the same web site, we cannot get the service using the former URL. It is known as the "HTTP 440" error and also is called "broken links". Some mechanisms are taken to minimize this error, but is not efficient enough.
- **Semantic-free.** The resource name should be semantic free from the attributes, or else it will restrict the flexibility and functionality of the network. If the resource name is related to the host, when the resource moves from one host to another, the name is not valid any more. If the name is related to the service provider (such as the domain name, which may have many hosts), the same data provided by different providers will have different names. Without other mechanisms, the user cannot get the data from different providers at the same time (such as using P2P approaches) because of using different names.
- **Middleboxes integration.** The network entity can not only direct the resolution of the name to its own location, but also to the specified delegate. Data center in the Internet deploys a lot of middleboxes, such as NATs, firewalls and cache servers, to manage and improve the applications and the services. It is hard and not flexible to maintain and configure the middleboxes in current Internet. And the middleboxes do not work well under network churn. DONS integrates the middleboxes into the network using name resolution. It not only allows, but also facilitate the deployment of middleboxes.
- **Self-certifying.** If the received resource is different from the original one after the transfer, the user will find it. The current security work more focus on the security on

Manuscript received Aril 19, 2010; revised July 28, 2010; accepted August 31, 2010.

doi:10.4304/jcm.5.9.684-691

the channel from the source to the user. In DONS we use the name to authenticate the source. When the user gets the resource, he can use the data and the public key to get a hash value. Comparing the hash value and the name of the resource, we can authenticate the resource.

The rest of the paper is organized as follows. In section 2, we survey the related work. In section 3, we give the basic design of the DONS, which involves naming and name resolution. In section 4, we give more detailed design of the system architecture and some applications over DONS. In section 5, we analyze the feasibility of the system, in terms of requirements and the simulation. In section 6, we conclude the paper.

II. Related Work

DONA [12] gives a data-oriented and beyond network architecture, which involves a clean-slate redesign of Internet naming and name resolution. It supports three user-relevant properties, which are persistence, availability and authenticity. It is available that data and services should have high availability in terms of both reliability and low-latency. It is authenticity that users know that the data comes from the appropriate source, rather than from some spoofing adversary. It replaces DNS names with flat, self-certifying names, and replacing DNS name resolution with a name-based anycast primitive that lives above the IP layer. DONA uses the route-by-name paradigm for name resolution. DONA relies on resolution handlers (RHs) to do name resolution, which is organized in a tree-based data structure. We think anycast primitive in DONA can be replaced by a more efficient name resolution mechanism.

LNAI [2] gives three levels of name resolution: from user-level description to service identifiers; from service identifiers to endpoint identifiers; and from endpoint identifiers to IP addresses. The service identifiers are semantic-free in LNAI. It supports integrate middleboxes in the Internet architecture. It decouples transport and network layers to seamlessly accommodate mobility and multi-homing. It uses DHT approaches to resolve flat names scalably.

SFR [24] gives the design and implementation of semantic free referencing, a reference resolution infrastructure based on DHTs. It gives every web page in the Internet a persistent object reference, which is invariant even when the referenced object moves or replicated. The reference is semantic free to support the mobility and replication. It uses improved DHTs to resolve names and uses three kinds of TTL-based caching to reduce the latency, which are the relay cache, the recent record cache and the popular record cache of the node in the infrastructure. It achieves a version of Web that uses only SFR. It handles user-level naming outside the reference resolution service by enabling a competitive market for canonicalization services that map human readable names to semantic-free tags.

DOA [25] gives an extension to the Internet architecture, called the Delegation-Oriented Achitecture (DOA), which integrate the deployment of middleboxes, such as NATs, firewalls and tranparent caches. It involves a set of reference that are carried in packets and serve as persistant host identifiers and uses the DHT approach to resolve these references to delegates chosen by the referenced host. In DOA, every Internet entity has a unique network layer identifier and network elements do not violate the network layering anymore. It gives two application examples, network-extension boxes and network filters, in the architecture.

I3 [22] gives an overlay-based Internet Indirection Infrastructure (i3) that offers a rendezvous-based communication abstraction. It use a set of flat identifiers to abstract hosts' network locations and forwards the packets based on the identifiers. Many applications would benefit from the architecture of i3, such as multicast, anycast and host mobility, where the sending host no longer knows the identity of the receiving hosts(multicast and anycast) and the location of receiving host need not be fixed (mobility). A prototype is built based on the Chord [23] lookup protocol.

INS [1] gives a design and implementation of the Intentional Naming System (INS), a resource discovery and service location system for dynamic and mobile networks. It uses a attribute and value tree for naming and implements late binding mechanism to integrate name resolution. It supports service mobility that the client can get the service even if the service provider changes from one to another. It is easy to configure the INS, which reduces the configuration errors. The INS does not incorporate security mechanisms in the naming architecture.

And also, HIP [16] uses a flat, self-certifying identity for the host. The authors of [13] and [14] also propose self-certifying identities. URN [15, 20, 21] gives every entity in the network a global unique identity. CoDoNS [18]uses a DHT alternative to replace the DNS, which reduces the resolution latency and is more resilient to the DoS attack. IPNL [17] , UIP [7] and NUTSS [9] integrate middleboxes in the Internet. DTN [3, 5, 19], HTTP [6], IPNL and DONA achieve the name-based routing.

III. Basic Design

In this section, we give the basic design of DONS. It includes the naming and name resolution. We use a persistent certifying flat name for naming, and use a DHT approach to resolve the name.

A. Naming

Every entity should have a unique identifier in its layer. Internet is a layering network. Every entity in one layer of Internet should have a unique Identifier, and the identifier should not violate Internet layering. In DONS, we use SIDs to distinguish different data or services. The SID is a persistent, self-certifying and unique flat name. SID could be produced by hashing the data, which may be a video, a web site or a web page. The SID is produce by the special local proxy in a domain. We think the data or the service are provided by the local proxy. The local proxy provides authentications for service providers.

The data that the user has received includes <data, public key, signature>. When providing services, the local proxy hashes the data and gets the hash value. Through the hash

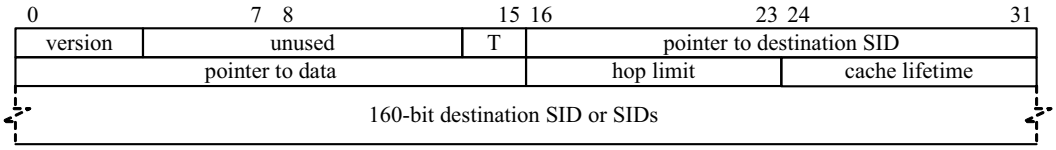

Fig. 1. The DONS message header format

TABLE I
SID MAPPING STRUCTURE

SID	0xd4f65d465f465df6adc...(160bit)
MAPPING	(Location1, Port1, TCP/UDP) (Location2, Port2, TCP/UDP) (Location3, Port3, TCP/UDP) ... (SID1, SID2, SID3, ...) Domain Name
METADATA	Path Name, Type of Service and so on
TTL	valid caching time of registered information

value and the local private key it gets the signature When getting the resource, the user uses public key to resolve the signature and gets the hash value of data. At the same time, the user hashes the data and gets another hash value. We will compare these two values. If they are same, we think the data is integrated. If they are different, we think the data is not integrated.

The SID is a hash value and is not readable for users. It is not for users but for applications. The user will provide keywords, user names or some other user descriptions. Some services, such as the search engines, will map the user description to the SID. The user deals with the user descriptions and the application fights for the SID. It is easier to deal with a uniform SID than a variable URL for the CPU.

B. Name Resolution

Table I illustrates the SID mapping structure. It mainly consists of SIDs, mapping information, meta-data and TTL values. The SID is an identifier of the resource. Mapping information is about the location of resource. SID can also be resolved into bunches of SIDs. It is used for integrating middleboxes. We will give detailed descriptions in the following chapter. SIDs can also be resolved into domain names. DONS is compatible of DNS. Meda-data is the path name, type of service of the resource or such information. The TTL value is the valid caching time of the given registered information. Through DONS, the SID is resolved into the network location of the resource. The Internet content provider registers the resource into DONS. Each resource has a global unique SID. The terminal application gets the network location and the meta-data of the resource from DONS using the SID.

DONS just maintains the meta-data of the resource. DONS is composed of distributed servers. It looks up the SID using some distributed algorithm, such as Chord, and stores the information using MySQL.

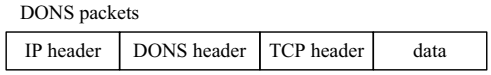

Fig. 2. DONS packets

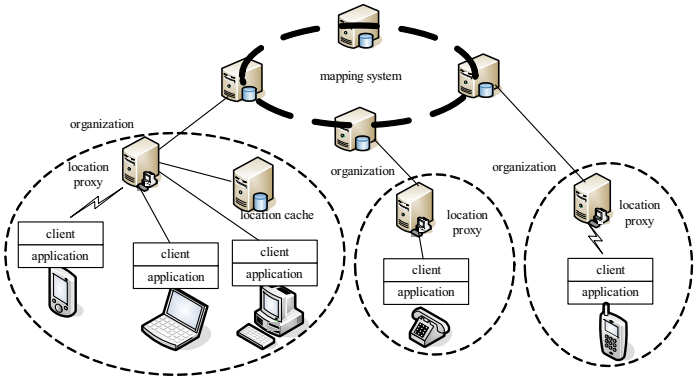

Fig. 3. DONS components

IV. SYSTEM ARCHITECTURE

In this section, we give the detailed design of the system architecture. It includes the DONS message header format, the components in DONS and some applications over DONS.

A. DONS message header format

The DONS data packets are shown in figure 2. The DONS-related content is inserted as a shim layer between the IP and transport headers. The transport header may be a TCP header or a UDP header. Here, we take the TCP header as a example. The DONS header format is shown in figure 1. A 2-bit flag T is used to determine the type of the name resolution. It may be an early resolving, or late resolving. The late resolving is composed of anycast late resolving and multicast late resolving. In this field, 00 or 01 presents early resolving; 10 presents anycast late resolving; 11 presents multicast late resolving. The destination SID is of variable length, so we use pointers to store the location of the destination SID. The destination SID or SIDs field stores the destination SID or a bunch of SIDs. The pointer to data field points to the content. The hop limit field decrements at each hop and limits the number of hops a message can traverse. The cache lifetime field determines the valid caching time of the given registered information.

B. DONS Components

The DONS components is shown in figure 3. The DONS is composed of the mapping system, the local proxy and the local cache server. The mapping system is used to map the SID to the network location of the resource. The local proxy provides

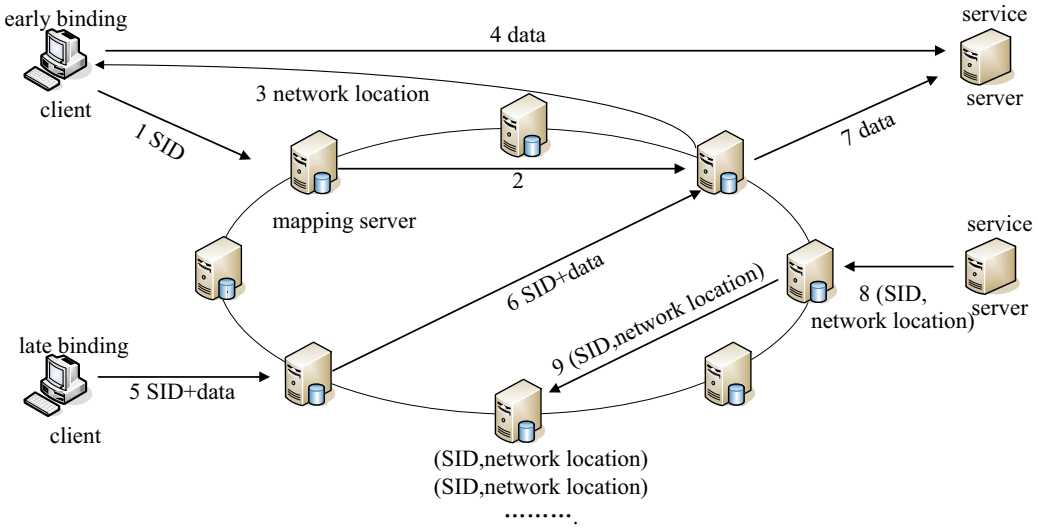

Fig. 4. The early resolving and late resolving

proxy service for the client lookup and registration. It is the delegation in its zone. The service can be considered provided by the local proxy. The local proxy should be responsible for the service providers in its zone. The local cache server is used to cache the lookup and register information. It includes two steps.

1) The register information is sent to the local proxy, and then sent to the mapping system;
2) The results of the lookup are cached in the local cache server.

The capacity of the server limits the cache content and time. The unpopular information will be deleted in time. There can be many local proxies with the common local cache server in one zone. The cache mechanism will improve the efficiency of the system. The clients could be PC, notebooks, mobile phones and sensor gateways. We should update the client software to support the system.

C. Web Application over DONS

The DNS system is host-centric, which resolve the host name of the service provider into the IP address. When a web page is moved from one web site to another or deleted from one web site, the original uniform resource locator (URL) is not valid anymore. The DONS system is date-oriented, which provides better-grained resolution. It supports the data movement and replication in different web sites. Every web page can have a unique and persist SID. We can get the SID by the search engine using keywords or by a email from your friend. The web browser sends SID to the mapping system and get the corresponding information of <IP address, port, path name>. Through these information, the client gets connection to the web server and views the web pages. When the web page moves to another web site, the corresponding information in the mapping system updates. The user can view the page through the new mapping information after the web page moves.

A SID could be a web site, a file folder, a web page or a picture in the web page. When the SIDs just focus on the web site, the mapping system is just like domain name system (DNS). DNS is a simple version of the mappings system.

D. SID-based routing over DONS

Name-based routing is divided into early resolving and late resolving. Source routing is considered an early resolving and routing based on forwarding table is considered a late resolving. In early resolving, the application gets the network location from the mapping system. Late resolving is divided into anycast late resolving and multicast late resolving, which forwards the messages through the mapping system. The process of early resolving is shown in left and upper part of figure 4.

1) The client forwards the SID query to the default mapping server;
2) The default server forward the SID query to the mapping server which maintain the response information;
3) A mapping servers response with the related network location information;
4) According to the network location information, the client connects to the server and gets the service.

The process of late resolving is shown in the left and lower part of figure 4.

5) The client forwards the packets with the SID and the data to the default mapping server;
6) According to the SID in the packets, the mapping server forwards the packets to one of the mapping servers;
7) The server, which maintains the network location information of the SID, gets the packets and then sends the packets to the right service provider.

The process of the service register is shown in the right part of figure 4.

8) The service provider send the <SID, network location> information to the default mapping server;
9) The mapping system forwards the register information to the right mapping server according to the SID and the routing algorithm (such as Chord).

In early resolving, the client first needs to resolve the SID and gets the network location information. Then according the network location, the client connects to the service provider. In late resoling, the data content and the SID are encapsulated in the packets at the same time. After getting the packets, the mapping server forwards the packets to the right mapping server or the right service provider according to the SID and the routing algorithm. The resolution of the domain name is an early resolving and DONS both supports the early resolving and the late resolving. In general, one SID may have multiple network locations, which means multiple servers provide the same service. The mapping server will select the service providers among them according the information of the latency, hops and so on. If we select the best server to provide the service, it is called anycast late resolving. If we select multiple servers to provide the service at the same time, it is called multicast late resolving. In anycast late resolving, if one of the server is failed, the mapping system will select the new service provider. The process of the service provide is not disconnected. In multicast late resolving, the servers provide the service at the same time. It reduces the bandwidth and the user can get the service in less time.

E. Middleboxes over DONS

The middleboxes, such as NATs, firewalls and CDN servers, play more and more important roles in the Internet. In many applications, the data should be sent firstly to the middleboxes to process. Such as, the wireless application protocol (WAP) gateway translates HTML web pages to WML for the wireless devices [4]. But in a certain time, middleboxes is considered that they disrupt the Internet layering. In DONS design, we not only allow, but also facilitate the deployment of middleboxes.

In figure 5, we assume that the user request is SID su, which is provided by the terminal B and processed by the server C.

1) The user A first looks up SID su through the mapping system, and gets the tuple <sc, sb>, which means the service is provided by B and is processed by C.
2) The user A looks up SID sc and gets the network location ac, such as the IP address of the server C.
3) A fills the packets with the source address aa and the destination address ac. And at the same time, the information <su, <sc, sb> and the data are encapsulated in the packets. The packets will be forwarded to the server C.
4) The server C gets the packets. It gets the information <su, <sc, sb> from the packets. It looks up sb through the mapping system and gets the network location of B is ab. It fills the new packets with the source address ac and the destination address ab. Also, the information <su, <sc, sb> and the data are encapsulated in the new packets. The packets will be send to the terminal B according to the destination address of the packets. Then, the service request is done.
5) According the process of the service request, the data provided by the terminal B will be forwarded to the server C. After processed, they will be sent to the terminal A finally.

It is the whole process of the service provide with the middleboxes. The DONS not only allows but also facilitates the deployment of the middleboxes. It keeps the openness and transparency properties of the Internet. In DONS, the user agent not only allows the packets are sent to its network location, but also supports the packets are sent to the proxy which is specified by the user and then sent to the user by the proxy. Such as, the user gets the e-mail after having it scanned for spam and viruses at the specified proxy.

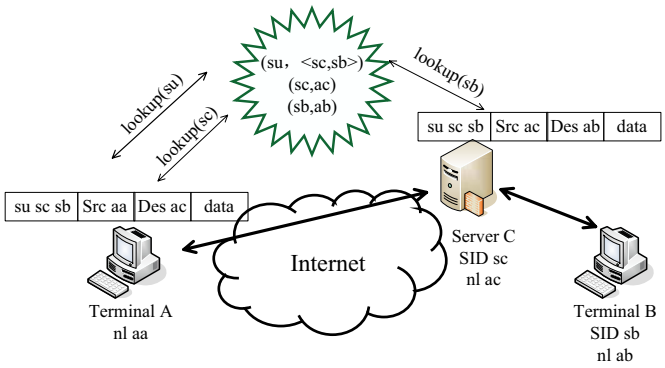

Fig. 5. Implementation of allowing and facilitating middleboxes

V. FEASIBILITY

In this section we analyze the feasibility of the system. We analyze the total number of the resource and the requirement of the system servers. We also present the performance of the mapping system through the simulation.

A. Requirements

The total number of web pages is more than 11.5 billions, about 10^{10} [10]. The increase of the number of web pages in recent 10 years is shown in figure 6. The number of the indexable web pages doubles in every 30 months.

We assume the total number of the data items is 10^{12}. The size of one mapping item is 428B, so the total size of all the data is 10^{12}*428B=256TB. If some service providers provide the same data or service, the number is smaller than this. We ignore this here.

We assume the the TTL value of every mapping item is 1 week, and then the total number of the mapping items which should be processed by the mapping system in every second is:

$$\frac{10^{12}}{1 * 7 * 24 * 3600} \approx 1.65 * 10^{6} \quad (1)$$

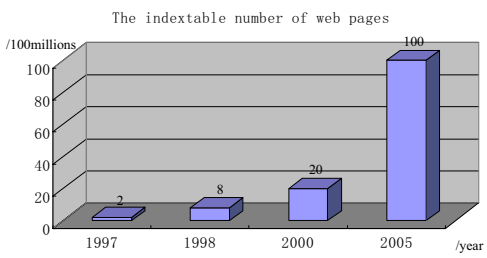

Fig. 6. The number of indexable web pages in recent 10 years

We assume that we give every mapping item a copy, and then the total number of the mapping items which should be processed by the mapping system in every second is 3.3 millions. If the hard disk size of every mapping server is 150GB, the total number of the mapping servers is:

$$\frac{256 * 2^{10}}{150} \approx 1740 \quad (2)$$

The total number of registration items which should be processed by the mapping in every second is about 2000. The size of every registration item is about 1KB, and the bandwidth of every mapping server for update is 15Mbps. If there is 20000 query packets for every second in a full Gbps link and the size of every query packet is about 400B, the query bandwidth is 60Mbps, which is six percents of the total full link loads.

B. Simulation

We use a modified version of p2psim [8], which is a free, multi-threaded, discrete event simulator to evaluate, investigate, and explore distributed protocols. The trace data is from Kingdata [11], Which measurements the latencies between a set of DNS servers. It can be used to evaluate other distributed systems including our DONS. The simulation runs for 72 hours.

The lookup latency with the maintenance overhead is shown in figure 7. There is 1740 nodes in the system. The average life time of the nodes is 3600 seconds. Every node sends one query in 600 seconds. The x axis is the live bandwidth of the node. It presents the total number of bytes sent by the node, which mainly includes the query and update traffic. The y axis is the lookup latency. The lookup latency and the maintenance overhead are contradictive, which cannot achieve the best at the same time. There should be a best trade-off point in this type of system.

Figure 8 shows the failed lookup rates with 256, 512 and 1740 nodes in the system. When the node bandwidth is lower than 6bytes/s, the failed lookup rate is high. While the failed lookup rate is low when the node bandwidth is higher than 6bytes/s. When the number of the node increases, the failed lookup rate increases little. The DONS is scalable to map the resource name to the resource location.

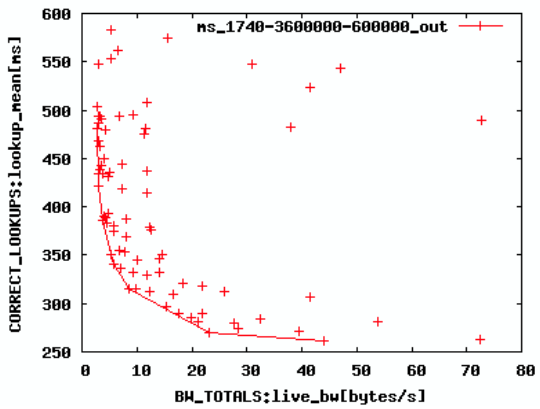

Fig. 7. The maintenance overhead with the lookup latency

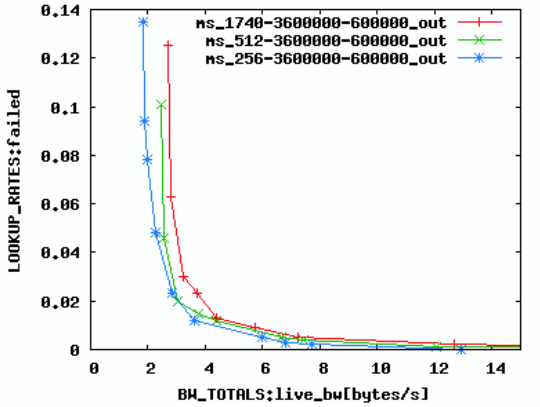

Fig. 8. The failed lookup rates with different node numbers

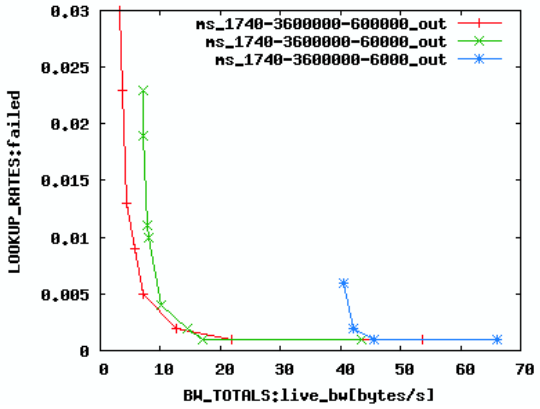

Fig. 9. The failed lookup rates with different lookup frequencies

The failed lookup rate and lookup latency vary according to different lookup frequencies, which are shown in figure 9 and figure 10 separately. In the figures, when the lookup frequency increase, the maintenance overhead increases. While the lookup latency and failed lookup rate increase little and maintain in low values.

Figure 11 shows that the lookup latency varies according to different node life times. When the node life time is long, the lookup latency is small and stable. Otherwise the lookup

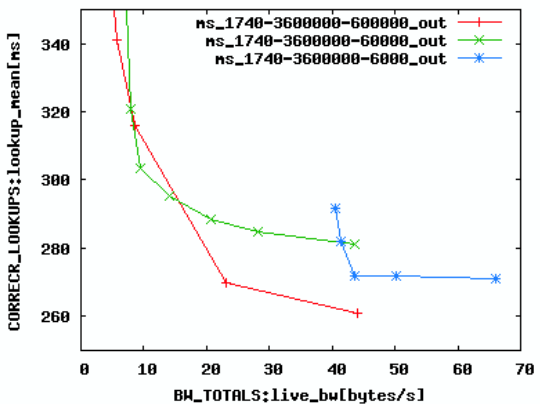

Fig. 10. The lookup latencies with different lookup frequencies

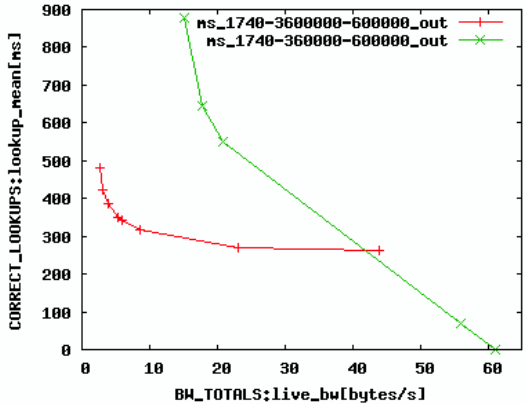

Fig. 11. The lookup latencies with different live bandwidth

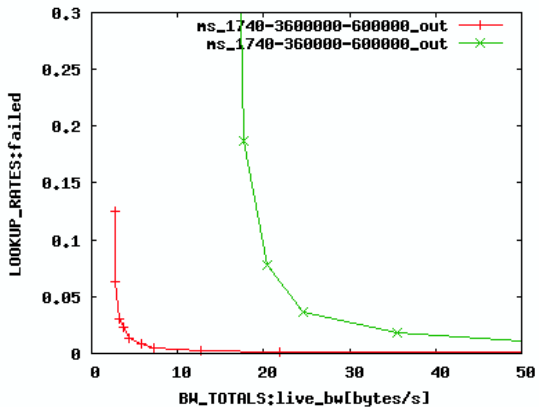

Fig. 12. The failed lookup rates with different life times

latency is big and varies in a large scale. Figure 12 shows that the failed lookup rate varies according to different node life times. When the node life time is short, the maintenance overhead of the system is huge and the failed lookup rate is high. When the node life time is long, the maintenance overhead maintains in a lower value and the value is stable relatively.

The lookup hops vs. different peer numbers is illustrated in figure 13. The lookup hops increase with the peer numbers logarithmically, which can be further reduced via caches. The maintenance overhead of the system vs. different peer numbers

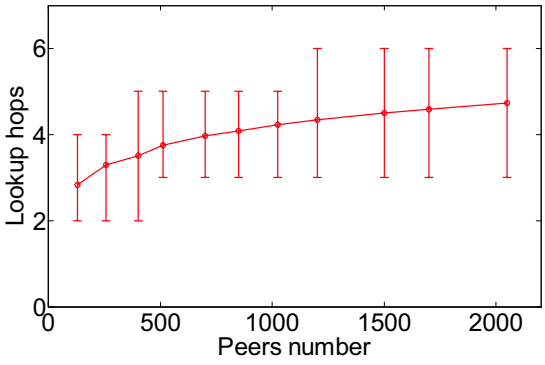

Fig. 13. The lookup hops vs. peer numbers

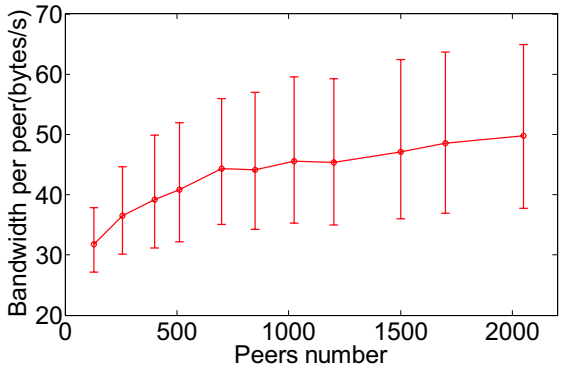

Fig. 14. The bandwidth usage vs. peer numbers

is illustrated in figure 14. The bandwidth usage increases with the peer numbers, which grows slower when the peer number is bigger. It is scalable, feasible and flexible for the lookup service of the internet.

VI. Conclusion

In this paper we propose the clean-slate redesign the Internet naming and name resolution. We decouple the service name from the host name and use semantic-free identifiers to name the resource, which improves the flexibility and functionality of the network. We use names to handle permanence so that once given a name of the resource, the user can get it at any time unless it does not exist in the network anymore. We use names to handle self-certifying so that if the received resource is different from the original one after the transfer, the user will find it. We use name resolutions to handle middleboxes integration so that the network entity can not only direct the resolution of the name to its own location, but also to the specified delegate.

There is also some work need to do in this type of system. DONS has a better-grained naming than host-based naming, but also takes more loads for the resolution. In such a data-oriented name service system, there will be plenty of data items in the network. Some more work need to do for improving the efficiency of the original DHT approaches or finding another method to provide more efficient resolution.

Acknowledgment

We thank the anonymous reviewers. This work is supported in part by the National Basic Research Program of China (”973 Program”) under Grant No.2007CB307101 and No.2007CB307106, in part by National Key Technology R&D Program under Contract No. 2008BAH37B03, in part by the Program of Introducing Talents of Discipline to Universities (111 Project) under Contract No. B08002, in part by the National Natural Science Foundation of China under Contract No. 60833002 and in part by the Fundamental Research Funds for the Central Universities under Contract No. 2009YJS016.

References

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, “The design and implementation of an intentional naming system,” in *Proceedings of the seventeenth*

ACM symposium on Operating systems principles. ACM New York, NY, USA, 1999, pp. 186–201.

[2] H. Balakrishnan, K. Lakshminarayanan, S. Ratnasamy, S. Shenker, and I. Stoica, "A layered naming architecture for the Internet," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM New York, NY, USA, 2004, pp. 343–352.

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant network architecture," *IETF Draft, draft-irtf-dtnrgarch-01.txt*, 2003.

[4] R. Cover, "WAP Wireless Markup Language Specification (WML)," *World Wide Web, http://www. oasis-open. org/cover/wap-wml. html*, 2001.

[5] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM New York, NY, USA, 2003, pp. 27–34.

[6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol-HTTP/1.1." RFC 2616, IETF, 1999.

[7] B. Ford, "Unmanaged Internet Protocol: taming the edge network management crisis," *Arxiv preprint cs/0603075*, 2006.

[8] T. Gil, F. Kaashoek, J. Li, R. Morris, and J. Stribling, "p2psim, a simulator for peer-to-peer protocols," 2003.

[9] S. Guha, Y. Takeda, and P. Francis, "NUTSS: A SIP-based approach to UDP and TCP network connectivity," in *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture*. ACM New York, NY, USA, 2004, pp. 43–48.

[10] A. Gulli and A. Signorini, "The indexable web is more than 11.5 billion pages," in *International World Wide Web Conference*. ACM New York, NY, USA, 2005, pp. 902–903.

[11] K. Gummadi, S. Saroiu, and S. Gfibble, "King: Estimating latency between arbitrary Internet end hosts," 2002.

[12] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM New York, NY, USA, 2007, pp. 181–192.

[13] D. Mazieres, M. Kaminsky, M. Kaashoek, and E. Witchel, "Separating key management from file system security," *ACM SIGOPS Operating Systems Review*, vol. 33, no. 5, pp. 124–139, 1999.

[14] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," RFC 4984, IETF, 2007.

[15] R. Moats, "URN syntax," RFC 2141, IETF, 1997.

[16] R. Moskowitz and P. Nikander, "Host Identity Protocol (HIP) Architecture," RFC 4423, IETF, 2006.

[17] P. Ramakrishna, "IPNL: A NAT-extended internet architecture," in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA, 2001.

[18] V. Ramasubramanian and E. Sirer, "The design and implementation of a next generation name service for the Internet," 2004.

[19] J. Scott, P. Hui, J. Crowcroft, and C. Diot, "Haggle: A networking architecture designed around mobile users," *IFIP WONS*, 2006.

[20] K. Sollins, "Architectural principles of uniform resource name resolution," RFC 2276, IETF, 1998.

[21] K. Sollins and L. Masinter, "Functional Requirements for Uniform Resource Names," RFC 1737, IETF, 1994.

[22] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proceedings of the 2002 SIGCOMM conference*, vol. 32, no. 4. ACM New York, NY, USA, 2002, pp. 73–86.

[23] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on networking*, vol. 11, no. 1, pp. 17–32, 2003.

[24] M. Walfish, H. Balakrishnan, and S. Shenker, "Untangling the Web from DNS," in *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation-Volume 1 table of contents*. USENIX Association Berkeley, CA, USA, 2004, pp. 17–17.

[25] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris, and S. Shenker, "Middleboxes no longer considered harmful," 2004.