Lightweight DNS for Multipurpose and Multifunctional Devices

Yogesh A. Wagh[1], Prashant A. Dhangar[2], Trupti C. Powar[3]

Amit Magar[4] P.B.Mali[5]

(Department of Computer Engineering, Smt.Kashibai Navale College Of Engg./ University Of Pune, India)

ABSTRACT: Internet is the combination of various devices. It is a medium of information transfer so it is used everywhere in the world. There are many DNS providers which offer DNS services to World Wide Web. This paper describes problems that occur during the use of DNS. Most of the DNS are working on the IPv4 addressing scheme as the use of Internet is increasing the size of IPv4 addressing is not enough hence we need to use IPv6 addressing to increase number of addressing port. This paper also covers lightweight DNS in terms of energy as well as time and study of various types of DNS.

Keywords: *DHCP, DNS, IP, RR, TCP.*

1. Introduction

Domain Name System [1] is used for the Internet and now a day's Internet is growing faster. Many users use DNS frequently because Domain names are easy to remember than the IP addresses .DNS is used instead of using directly IP address. Domain name is mapping of IP address. Domain names actually a simple English like words separated by dot ("."). DNS works on the application layer of standard Internet model. web-sites are identifying by name that is Domain Name. Browser requires more time to load web page as user type domain name instead of IP address and to get the ip address of these domain names request is sent to domain name server. There are two types of domain name authoritative server and recursive server which serves the request of users .DNS stores mapping of IP addresses in the form records. These records are going to change after some time referred as TTL which is basically hop count limit.

Whenever user travels from one base station to another then the requests for service provider is made repeatedly which introduces the locality problem thus the energy utilization is more. hence to overcome these problems need to design a lightweight DNS as the energy is function of time and space. Dynamic changes in domains are get updated continuously in the locally stored DNS records. The record which is available on DNS server we can simply copy and stored that in local DNS file.

1.1 Motivation

Today the world is of handheld devices. We need to use the Internet or network from anywhere, anytime, anything. So this type of system is calling as ubiquitous [1] system. The world is heavily depends on electronic devices (handheld) and they are closely related to use of energy, so the efficient use of energy is the main concern. Handheld devices are multipurpose and multi-functional. Hence the use of energy should uniformly distribute among the entire task performed by the device.

Currently working DNS are installing on remote machine. When client wants to use any site it initiates ask for address-name resolution. DNS server [2] generates the response. So it causes lots of energy waste during the process of request and response. Time requiring for this process based on locomotion of DNS server and client as well as network characteristic such as network topology, bandwidth, and traffic in network. The time required to above process varies a lot. DNS server fulfills the request of many clients simultaneously. If number of client’s requests to same address resolution then it creates duplicate copy of record it has been needed for each client so this creates redundancy of in DNS records. Space requirement is large than actual required memory. To overcome these drawbacks we need to avoid duplication of records. Energy is function of time and space so if requirement of time and space is less, then utilization of energy is less. Handheld devices become more compact and less power consumption. Memory requirement to store DNS records is more as we are using IPV4 [3]. IPV4 requires 32 bit of address mapping to DNS name. IPV4 is running out of limit its addressing capacity.

80% of requests are locally cached in the local DNS file and 20% of are not. So 80% of requests required less time than second. And 20-30% of non-cached look-ups take more than 1 second so out of all 4-6% requests take more than second[7].We expect popular servers to have large influence on DNS request. So we are going to save

Manuscript received December 5, 2013
Manuscript revised December 20, 2013

them locally. So that DNS mapping time can be reduced. Hence overall performance gets increased.

2. Related Work

Power DNS [4] is an authoritative as well as recursive server. Power DNS supports dynamic, plain zone and LDAP directories backend. It can immediately update the zone record because it stores mapping between domain names and IP addresses to the database. Reloading of records is very fast due to it does not uses any traditional loading methods. Hence it is very flexible. Power DNS is much optimized it takes 6 minutes to parse 113k small zones and start serving them.

DJB DNS is collection of server that handles different characteristic of DNS example caching authoritative server zone transfers. DJB DNS can easily be victim of Cache poisoning in relatively very less time. DJB DNS is an alternative to BIND server because it is easily configured than BIND. Less than 7000 instructions are used in Different Vulnerable attacks are possible on DJB DNS like birthday attack.

Open DNS [5] is largest independent DNS provider processing over 30 billion request per day. It is originally based on DJB DNS cache. Open DNS employed by 1 percent of worldwide network. It offers web content filtering and malware, botnet and phishing making protection to secure the schools, small business and enterprises networks. Open DNS is world's leading provider of DNS that makes k-12 schools network faster, reliable and more safer to use. Drawbacks of Open DNS are it requires fast response time.

Simple DNS plus [6] is used to speed up the Internet access and it also helps to host local DNS. It uses standard ASCII based text file (Zone file) to store the DNS records and this file is located in local directory. It provides user interface and user can easily add, update, delete operation .Simple DNS plus perform function as "DNS Resolver/Cache [7]" and "Authoritative DNS Server ".

KNOT DNS is open source DNS with high performance and full zone transfer. It is based on object oriented design. Knot DNS is another authoritative only and open source server. It is developed in object oriented programming approach. Knot has implemented lock free architecture making it more concurrent and faster in working. Knot design clearly separates the network and threading, query processing and DNS Library making it three tier architecture. It is compliance nearly all current standards. It minimizes look up time of each query with hash table its worst case O(1) look up time making query processing more faster. It minimizes energy and time required to look up operation which is very low is Knot DNS compared to another DNS. The main feature of Knot DNS is it supports runtime reconfiguration. KNOT DNS is currently under testing phase and many features like dynamic update, root zone support yet to be added.

BIND 9[8] are implemented by Berkely Internet Name Domain (BIND).It is available in different operating system like Unix , windows (XP,VISTA).It supports both IPv4 as well as IPv6[9] addresses. BIND 9 provides resolution services to local clients using a combination of a lightweight resolver library using the Lightweight resolver protocol which is based on simple UDP-based protocol. Bind 9 deal with zones. It requires large memory space for the purpose of cache. It also store older libbind resolver library. DNS Notify, Dynamic update etc. features are available in BIND 9.

Dnsmasq is a lightweight DNS server provides DNS service to LAN. It supports IPv6 addressing for DNS. Every machine in LAN has a particular name and this name and MAC address of that machine as DNS record entry in Dnsmasq DNS if LAN is connected through DHCP. All DNS records stored By Dnsmasq are not globally declared IP. It does not solve recursive query but it send this query to rec.

3. Related Work Evaluation

Recursive access control is using for load balancing. Load balancing used for distributing DNS balance among the host in the network. Knot DNS, Dnsmasq does not Support Recursive access control. Dnsmasq is partially authoritative server and does not offer recursive DNS services. Scope of Dnsmasq is very limited because it is going to store DNS records only for locally configured IP not globally configured IP. Djbdns partially supports IPv6 via generic records. Wildcard DNS technique is partially available in Djbdns. As we consider about Knot DNS it is not recursive. It consists of many zones but starting with it is very slow.And also it requires high memory storage for query processing.if we try to reduce the memory requirement then it may affect on performance. DDNS is the dynamic DNS which update the DNS after changing of IP address at run time. But it is absent in the knot DNS for updating of DNS we can use both static and dynamic IP but using simple DNS plus we can update by only dynamic IP. All DNS suffers from the Zone transfer problem.

4. Proposed Work

It is necessary to use information from web it must light-weight .Lightweight about when access made to any website (web document) instead the information stored in web infrastructure it get stored on DNS server by the service providers. The Internet service provider stores it in DNS records .So the access is making through only DNS resolver. By performing such a task, time is least because of the overhead is reduces by eliminating TCP handshaking and http get a response. Thus it is lightweight information provision using DNS. Once Information stored in the form of DNS record the access to this which done by the DNS resolver who satisfies the requester when it resolving DNS name, there is no need to use other web resources.

Consider sample example if user likes to know the population of certain place represented it by domain name"_populatn.place@1 .sample.com". First it queries a DNS request to the local DNS server , local DNS does not found information in its database , then it will communicate to information providers maintained authorized DNS server. Thus the client query will be satisfied.

There are certain assumptions regard to DNS for information providers are

1. Expressing information: Providers consider to express (organize) the information in accordance with DNS name space. In the sample example population denoted as _populatn.*.And the name space is hierarchical which represents the information of parent domain. Provider organizes the data on server in a way that depends on organization of parent domain information.

2. Information storage: How to store different kind of information in DNS server? DNS RR (Resource record with TTL value) used to store the contents.

3. Request and response information: User sends a DNS SRV query for retrieving information which is located at DNS server and response message used to return TXT RR.

4. Updating information: Dynamic updates of information done by DNS update queries (DNS Dynamic updates).

The function of Domain name system is specially working for mapping IP address to its related domain name. Today all over the world IPV4 addressing is used but in future there are lots of limitations because size of IPV4 is 32 bit. Hence we proposed system who works on IPV4 as well as IPV6. Since IPV6 is a "Lightweight Addressing Scheme" and it also provides 128 bit size. Here we have to use 6LowPAN for energy efficiency. Generally HTTP request is sent to authoritative and recursive server who serves DNS request to the host for particular domain name. Time required for above process is too much for that reason here we store domain name records to machine it. Only most frequently used sites get stored .For example- Google, Facebook, and yahoo.

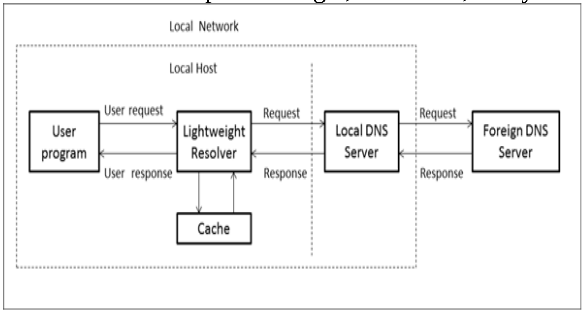


Fig. 1- Proposed Lightweight DNS architecture.

5. Implementation

The system has been developed mainly in three modules as given below: First part of project contains viewer module. This module contains all GUI required for user interface. Second part contains domain name resolver connecting module which map Domain name to IP address. The last module is for content filtering and access point.

PART 1

Viewer module is nothing but the user interface module which provides GUI to user with help of this user connect to internet service where user have to enter domain name of required web page in to the textbox available on the GUI called address bar. User can give new domain address by refreshing it.

PART 2:

Domain name resolver take input from viewer module which is specified in address bar as domain request .it resolves this domain name request to IP address using the local cache file if record is not available in the cache file then it send request to domain name server and send response to viewer module .simultaneously save this record in cache file.

PART 3:

In content filtering module we can block the web pages by using it. The basic idea behind it is we can block one word or one sentence according to that web page check this word or sentence and block this web page according to it. And another part is creating the access point means is a device that allows wireless devices to connect to other devices. So it allows connecting other devices to this device and accessing the cache file.

Web Content Filtering

Content filtering is technique in networking to ban or deny some information based on the contents of information. Sometime the user tries to access sites which are not allowed to them or parents want to restrict their children from adult contents of website. There is no option available on android devices to do so. We have designed an algorithm by which the contents of website will be filterable based on user need.

Algorithm for Content Filtering

1) Get address of web page user want to visit check to see it is in blacklisted list if not then proceed to step 2 else go to step 6.
2) Get html representation of the page store it in some string.
3) If page contents the RTA tag and user want to restrict adult contents then ban the block access to website then go to step 6 else proceed to next step.
4) Check page contents with user banned key words if page contents those words then ban the webpage. Enter it into blocked websites list (black list). If it doesn't contain then load the web page.
5) If user entered that site again then check it with blocked website addresses if it is present in list then don't load it. Display default page of banned site else Load the web pages.

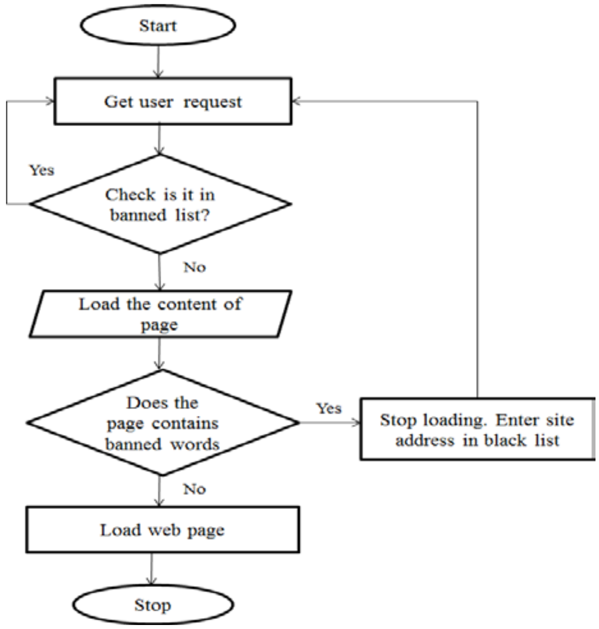

Start
Get user request
Yes
Check is it in banned list?
No
Load the content of page
Does the page contains banned words
Yes
Stop loading. Enter site address in black list
No
Load web page
Stop

Fig. 2 - Flowchart for content filter

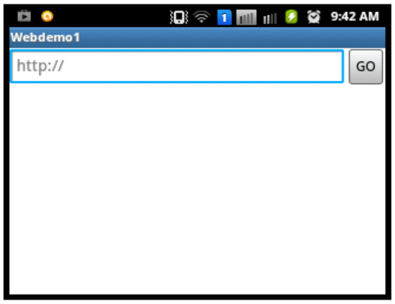


Fig. 3- Browser view

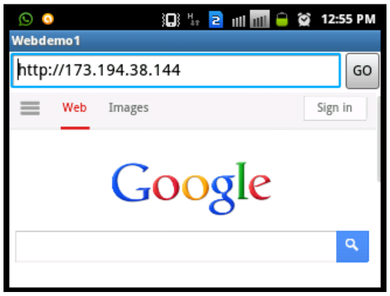
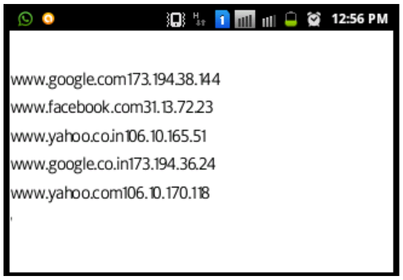
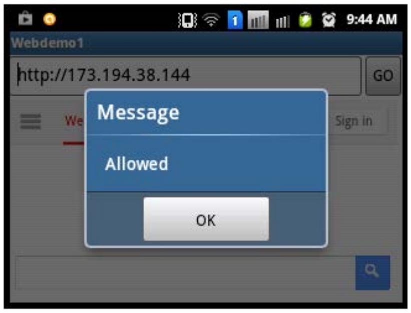


Fig. 4 -Mapping of Domain name to IP address



```
www.google.com173.194.38.144
www.facebook.com31.13.72.23
www.yahoo.co.in106.10.165.51
www.google.co.in173.194.36.24
www.yahoo.com106.10.170.118
```

Fig. 5 - Local cache File



Message

Allowed

Fig.6 -Content Filtering

6. Conclusion

In this paper, we have proposed the Light weight DNS for Handheld Devices. We have described how DNS actually work and what are the problems occurred in the old DNS in terms of memory, Energy, Time etc. And we have proposed a new DNS which is more efficient in terms of Energy, Time and space.

7. Future Work

As we are proposing DNS for handheld devices there is a future scope for it. The main issue regarding the security so now a day's security is important so we can add security in DNS so that unauthorized person cannot handle it and there is one more issue regarding handheld devices that are battery life. When we go through one base station to another base station then for hand-off required battery wastage is more so we can save the battery/ power so we will create a Green DNS.

References

[1] "*Domain name concepts and facilities*" RFC 1034 by P Mockapetris , November 1987

[2] "*Domain name implementation and specification*" RFC 1035 by P. Mockapetris , November 1987

[3] "Mobile IPv4 challenges/Response Extensions(Revised)" RFC 4721 by C. Perkins ,January 2007

[4] "The Power DNS name server" by Augie Schwer

[5] "Open DNS 2010 Report web content filtering and publish". http://www.OpenDNS.com

[6] Simple DNS Plus Version 5.1 Copyright © 1999-2008 JH Software ApS

[7] "The Contribution of DNS Lookup Costs to web to object retrieval" by Craig E. Wills,Hao,Shang.

[8] BIND 9 Administrator Reference Manual Internet System Consortium 950 Charter Street Redwood City, California USA http://www.isc.org/

[9] "*Internet Protocol version 6(IPv6) Addressing Architecture* " RFC 3513 by R. Hinden Nokia and S. Deering Cisco system , April 2003.