ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20170515150608
Original scientific paper

Dynamic Server Selection by Using a Client Side Composite DNS-Metric

Dražen TOMIĆ, Drago ŽAGAR

Abstract: Dynamic Server Selection (DSS) is a new DNS method for the optimal server selection of a multiple available network service. The method allows dynamic selection of a server on the client side based on the information of the server load and its network topological distance from the client. The server selection is based on the calculations of a composite DNS-metric in which servers, whose IP addresses are sent in a DNS response, are ranked from the optimal to the least suitable. Calculation parameters are server response time, which the client measures for each server independently, and the server load, which is specified by the server administrator. The DSS method has the lowest overall network service response time in comparison with the other four observed methods (Geographical, Hops, Random and RTT) which, in measurements done in a real time environment, have longer response time from 8.5% to 26.8% compared to DSS.

Keywords: computer networks; domain name system; dynamic server selection; network servers

1 INTRODUCTION

Due to the increasing importance of the high availability of network services that servers provide to its clients and increased server load, it is common to establish a highly available network service by using two or more servers, located on the same or different locations, and connected to the internet with a single or multiple communication link [1, 2]. By using multiple servers to perform the same network function, it is possible to [3, 4]:

- distribute the load between servers based on the current load of the individual server and the load of individual communication link of the server (server and link load balancing);
- enable access to network services in the case of unavailability of a particular server, caused by server failure or the failure of the network path between the client and the server (network and server failure failover).

Thereby placing servers on different physical locations and/or different access communication links, allows the optimization of the network traffic between the client and the servers in a manner of selecting the optimal server for each client [5, 6]. If in such cases redundant servers have different network response times, because of different performance and/or load of a server and/or a communication link, the client can access the server with the shortest network distance and/or lowest loaded, that is select the server with fastest response time to the request.

To determine the server with the fastest respond to a client's request, it is necessary to consider server response time, a parameter of the computer network, and server load, the parameter of the server's network service. Since these are dynamic, time varying parameters, it is necessary to have their most accurate and the most recent values and an algorithm that, based on the available data of network/server response time and server load, will determine which server is the best for delivering a network service for each individual client.

The goal of developing a method for a client side dynamic server selection of a multiple available network service by using a composite DNS-metric in the existing DNS [7] is to provide:

- For clients: faster network service response time and faster data transfer, therefore greater customer satisfaction with the network service
- For network service providers: improved solution for server unavailability occurrence and enhanced management of the server and internet links load
- For basic network infrastructure/ISP's: faster data throughput and reduced possibility of network congestion

In this paper DSS (Dynamic Server Selection), a new client side method for dynamic server selection of a multiple available network service is proposed. The server selection process is based on the calculations of a composite DNS-metric for address DNS resource records returned to the client. Composite DNS-metric can include and combine the information of the particular server load and its network topological distance from the client. The DSS method does not exclude any of the existing methods for server selection and can be used as a standalone or an additional method for optimal server selection on the client side, in particular as an extension of server side methods by introducing the client's view into the final selection process.

The paper is organized as follows: after the introduction, in Chapter 2 the previously related work is shown chronologically. The third chapter describes DSS, a new client side method for dynamic server selection of a multiple available network service. Chapter 4 contains implementation of the DSS method and analysis of the method effectiveness. Chapter 5 concludes the work.

2 RELATED WORK

In [8], a dynamic server selection is introduced for the first time and the basic metrics for measuring network distance are defined: the number of hops (static metrics) and the packages round trip time – RTT (dynamic metric). The dominance of the dynamic methods compared to static methods is confirmed. In [9] the goal of the dynamic server selection is defined: delivery of service in the shortest possible time. The main reasons for using replicated network services are server load and network delay because of slow, loaded or long distances paths.

In [10], it is determined that when selecting the closest server, its geographic proximity does not necessarily reflect its network proximity as well as the least loaded server, and using the DNS round robin method is not sufficiently precise.

In [11] it is shown that collecting additional servers and network paths data before selecting servers significantly improves service response time. The metric used is multiple measuring of RTT. It is proposed to introduce server load as a new additional metric. The introduction of DNS queries preprocessor related to DNS that works as a DNS proxy and can operate in both the client and the server side was proposed in [12] to solve the problem of request distribution between redundant servers. Further in the same study, in [13] the necessity of introducing a method of selecting the server on the client side was indicated, wherein the possible problem is collecting the necessary server parameters.

In [14] it is concluded that DNS is a simple method for directing clients to the closest server that does not require changes in existing network protocols wherein the existing problem is collecting, combining and delivering the necessary server parameters. In [15] a DNS based scheme of the server selection in CDN (Content Delivery/ Distribution Network) was introduced where the basic requirements are sustainability and simplicity.

Dynamic client server selection method based on QoS (Quality of Service) is shown in [16]. It is said that the selection on the client side has the problem of determining the server load and the best server selection from the client's perspective can be done using QoS.

In [17] it is shown that the appropriate server should be selected considering the estimated location, measured RTT and advertised server load. In [18], the authors are developing a tool that researchers can use for their own metrics exploration and for determining network proximity and algorithms for routing queries.

In [19], an empirical assessment of five client-based policies for the selection of web services is created. It was concluded that features of the local client environment can have a significant impact on some of the policies. In [20], the resolution of DNS queries in CDN by defining the "CDN DNS Request Routing" strategies is analysed. A network infrastructure solution for server selection in which routers can select different paths from clients to servers is proposed in [21].

According to the authors in [22], by using a multihomed networks Internet service may be available by multiple network paths and it would be possible to improve the sensitivity of the servers to network failure and increase system performance.

The impact of the DNS TTL value to the DNS server load is analysed in [23]. In [24] the introduction of a proxy DNS on the client side is proposed which, for the CNAME (Canonical Name) DNS responses, directly send queries to the authoritative DNS servers of the CDNs so the CDNs know the location of the client and can modify the DNS response.

In [25] and [26] it is stated that the CDNs direct the clients to their servers based on unreliable information of the client's location and the network conditions between the client and the server. They propose a method in which the client must use the ISP's DNS server that forwards authoritative answers to the server that ranks them.

In [27] a probabilistic video chunk scheduling approach for HTTP protocol from multiple servers in parallel is proposed, considering heterogeneous and time-varying bandwidth of multiple servers. In [28] new protocol is designed to improve the user experience by providing better fairness, efficiency and stability in the context of multi-server HTTP adaptive streaming. In [29] dynamic server selection strategy is proposed that enables the streaming client to select the optimal video delivery server and allows any adaptation algorithm to be plugged into it.

This paper proposes a new method for the optimal server selection of a multiple available network service that is not specific to one type of network service, which takes into account the current server load and the client's topological network distance instead of the geographic distance between the server and the client. Furthermore, a new method does not approximate the client's network distance based on the client's IP address, and does not require the creation of special infrastructure or the use of specialized servers. The method does not require constant monitoring of network services, does not have complex network requirements for implementation and does not require changes to basic network infrastructure.

The DSS method solves the problem of delivery and combining parameters for dynamic server selection process on a client side by combining two dynamic parameters, one from the server side (server load) and one from the client side (network distance represented by RTT) in the form of a composite DNS-metric controlled by the service/server administrator. The DSS method can be implemented quickly and easily by using an existing DNS infrastructure for DSS data transfer to the client and can use DNS functionality for dynamic parameters updates.

3 DYNAMIC SERVER SELECTION (DSS)

The DSS method allows the client to independently, according to the information of server's IP addresses for the requested network service, which is given as a response to the DNS request, determine which servers are currently available, load factor of each server and its network topological distance from the client. This data, along with rules specified by the administrator of the network service, allow client to select the most appropriate server through the process of calculating the composite DNS-metrics for delivered A/AAAA DNS resource records (RR), where A is IPv4 and AAAA is IPv6 IP address. That way, the two dynamic parameters, one from the client side (network distance represented by RTT) [9, 11] and one from the server side (server load) [12] can be combined together. That solves the problem of introducing the method of selecting the server on the client side, wherein the existing problem is collecting, combining and delivery of the necessary parameters [14].

The method introduces a new type of DNS RR: RR TYPE DSS in the DNS class CLASS (IN)

DSS RR is an optional RR in the zone database of the authoritative DNS server. It allows description of the parameters such as load of a particular server (defined by A/AAAA RR), parameters for network distance determination and the parameters for calculating the composite DNS-metric of the A/AAAA RR. DSS RR is used in additional section of DNS response and has a standard structure of an additional section RR:

Name Type Class TTL A/AAAA Priority Load Impact Request Interval Protocol Port Time Refresh Timeout Flags

Field: *Type:* Description:

NAME: *u_int16* requested DNS name (QNAME)

TYPE: *u_int16* specifies the type of the resource

CLASS: *u_int16* identifies a protocol family or instance of a protocol, for DSS RR has standard value IN

TTL: *u_int32* describes how long a RR can be cached before it should be discarded

RDLENGHT: *u_int16* field that defines the length of the RDATA RR in octets

RDATA: contains DSS RR data and has the following format:

Field: *Type:* Description:

A/AAAA: *u_int16* Value of A/AAAA RR of the server for which the DSS record is defined. In the database of the authoritative DNS server a 32-bit IP address is entered for IPv4 or a 128-bit IP address for IPv6. During the formatting response to the DNS query, it is replaced by a 16-bit pointer to IP address in A/AAAA RR, in the same form of NAME field pointer

PRIORITY: *u_int8* server priority factor, allows the distinguishing between primary and secondary servers. The value 0 represents highest priority and the value 255 represents lowest priority

LOAD: *u_int8* server load factor, allows the description of the server load within the priority class. The value 0 represents lowest and the value 255 represents highest server load

IMPACT: *u_int8* composite DNS-metric factor that is used in metric calculation to determine the impact of server's network distance on the composite metric calculation. The value 0 means the least impact (network distance is not considered) and the value of 255 indicates the biggest impact. It can also be used to prioritize certain server's links

REQUEST: *u_int8* define the number of requests that the client sends to the server for testing the network distance

INTERVAL: *u_int16* time in milliseconds between sending two consecutive requests for testing the network distance

PROTOCOL: *u_int16* Internet protocol type, has the standard value 6 for TCP and 17 for UDP

PORT: *u_int16* PROTOCOL's port that contains the service for network distance testing

TIME: *u_int16* time in milliseconds from the sending of the first packet to test network distances up to the start of DSS procedures for calculation of composite DNS-metric

REFRESH: *u_int16* time in milliseconds from the start of the DNS-metric calculation up to the restart of the DSS calculation in order to refresh the DNS-metric

TIMEOUT: *u_int16* time in milliseconds from sending the first packets to test network distance up to declaring the server unavailable

FLAGS: *u_int16* field for marks, 8 MSB bits for development, 8 LSB bits for the mark of the DSS version, the initial version is 0

The DSS method functionality is described for authoritative and non-authoritative DNS servers and DNS clients.

3.1 Authoritative DNS Server

The rules of implementing DSS RRs into an authoritative DNS server:

- Authoritative DNS servers implement DSS RRs only for A/AAAA RRs and do not use them for CNAME RRs. Each A/AAAA RR can have one or more DSS RRs but only one RR for the same value PROTOCOL/PORT. If there are more DSS RRs defined for the same domain name, the last defined RR is always used, unless there is an explicit request for a specific DSS RR defined by the value PROTOCOL/PORT. If there is at least one DSS RR existing for a domain name, the value for the last defined DSS RRs is used on all the A/AAAA RRs of that network name, for which the DSS RR is not explicitly defined. When there is no DSS RRs defined for a domain name, then DSS method is considered not implemented for that domain name
- For DSS RRs the same rules are applied as for the other DNS RRs. For the RRs values (fields) that are not specified, values defined in the first previous NAME record are applied. The first DSS RR for each domain name must have all DSS fields defined, there is no predefined values for DSS parameters (fields)
- The authoritative DNS server will enter all DSS fields in an additional section in reply to the requested A/AAAA RR and will fill in all fields considering the entry order in the database. The DSS algorithm on the client side will enforce all the rules for multiple RRs and RRs that are not defined.

An example of implementation of the DSS RRs in the authoritative server is shown in Fig. 1. Explanation of the DNS RRs of the authoritative DNS server:

The domain name "server.example.com." has 4 defined IPv4 addresses of network servers: 192.0.2.10, 198.51.100.20, 203.0.113.30 and 203.0.113.40.

For the domain name "server.example.com." 4 DSS RRs are defined, 3 to TCP port 80 (HTTP service) and 1 for TCP port 25 (SMTP service) with TTL values that are identical to the corresponding A RRs. In the first DSS RR, all the DSS fields are defined. There is no defined DSS RRs for the IP address 198.51.100.20, but as this IP is in the A RRs of the domain name "server", the last defined DSS RR is used for it: "DSS 192.0.2.10 0 0 0 4 10 TCP SMTP 3000 10 5000 0". DSS RRs for the IP addresses 203.0.113.30 and 203.0.113.40 don't have all DSS parameters defined, thereby the parameters of the first DSS RRs are used, respectively the values "0 4 5 TCP HTTP 2000 5 5000 0" for the IP address 203.0.113.30 and the values "4 5 TCP HTTP 2000 5 5000 0" for the IP 203.0.113.40.

The total size of the DNS response to the A RR request for "server.example.com.", where the response has four A RRs and 4 DSS RRs, is 12 + 24 + 64 + 128 = 228 bytes which is significantly less than the recommended maximum size of DNS UDP datagram of 512 bytes. When the A/AAAA field would use a 32-bit IP address instead of a 16-bit pointer, the overall size of the DNS response from the example would be 228 + 4 × 2 = 236 bytes. The DSS method in its basic design supports the IPv6 protocol as information about an IP address, regardless it is IPv4 or IPv6 address, is delivered as a pointer to the appropriate A/AAAA RR in the response section. The size of the UDP datagram containing the DSS's DNS response to the AAAA query is increased by 12 bytes per AAAA RR, what is the difference in size between the IPv4 and IPv6 addresses.

3.2 Non-Authoritative DNS Server

Non-authoritative, usually recursive or forwarding DNS servers should forward the DSS RRs in the same form as they received them: for A/AAAA RRs they should forward all the associated DSS RRs with unchanged content (except TTL field) in the same order they received them in the DNS response from the authoritative or some other non-authoritative DNS server. Non-authoritative server that received DSS RRs from other non-authoritative server does not require an authoritative response and can insert the DSS RRs in a cache considering the TTL value of the DSS RRs.

3.3 DNS Client (Resolver)

A DNS client that receives a DNS response containing DSS RR and supports the DSS method calculates the DNS-metric in the following way:

- Checks whether all A/AAAA RRs have the corresponding DSS RR. If an A/AAAA RR has no associated A/AAAA RR, the data of the last DSS RR obtained in a DNS response is applied on it
- Checks whether an A/AAAA RR has multiple DSS RRs. If there are multiple DSS RRs, the last DSS RR is used unless the DNS client has the information for which PROTOCOL/PORT A/AAAA RR is required, in which case is selected the appropriate DSS RR. For the full functionality of the DSS method, which supports multiple DSS RRs for the same A/AAAA RR, it is necessary to provide the information to the DSS client for which service the DNS query is requested
- Checks whether the field PRIORITY has the same value in all DSS RRs. If the field PRIORITY has different values, it processes only DSS RRs with the highest priority (lowest value of the field PRIORITY)
- (a) Starts measuring time and to all A/AAAA IP addresses from the previous step sends queries every INTERVAL milliseconds, whose number is defined by the REQUEST parameter, to determine the availability of the network service on the defined PROTOCOL/PORT
- After the expiration of the TIME period, checks the received responses (RTT values):
- If a response was received from a minimum of one A/AAAA IP address, the DNS client initiates the procedure of calculating the DNS-metric using medium RTT value as RESPONSE parameter (in milliseconds) in the calculation of the metric. After calculating the DNS metric, the IP addresses are sorted in the order of the IP address with the lowest metric to IP addresses with the highest composite DNS-metric. With the IP addresses obtained after calculating the metric, the DNS client continues to work as with classic DNS responses respecting the order of sorted IP addresses
- If there is no received response from the A/AAAA IP addresses to which the request was sent, after expiry of the REFRESH period, the algorithm returns to point (a), thus increasing the period to twice the value of the previous TIME period (TIME = TIME * 2, binary exponential back off). The maximum number of iterations is two, after which the DSS RRs with the next lower priority are processed (with the next higher value of the field PRIORITY). If even after checking the RRs with the lowest priority there is no received responses, the DNS client continues to work with DNS query as with a classic DNS query without the DSS support. For servers that do not respond in the TIMEOUT period, the associated DSS RRs are not considered until the end of the TTL time

Results of the DSS algorithms can be stored in the DNS client's cache, considering the TTL value of A/AAAA and DSS RRs (it is recommended that both have the same TTL value) and specifying the PROTOCOL/PORT for which the calculation is made.

3.3.1 Calculating the Composite DNS-Metric

After receiving DSS RRs and finishing measuring RESPONSE parameter, the client starts to calculate the composite DNS-metric for every server's IP address. Calculation shown in Eq. (1) is based on DSS RRs data returned by DNS server. The metric (METRIC) for the i^{th} DSS RR from a total of N DSS RRs for which a response to a query is received is calculated as follows:

```
$ORIGIN example.com.
@                    SOA server.example.com. postmaster.example.com. (
                     2016010101 3600 3600 604800 3600 )
                     NS   ns1.example.com.
                     NS   ns2.example.com.
; A resource records
server    2h    IN   A    192.0.2.10
                     A    198.51.100.20
                     A    203.0.113.30
                     A    203.0.113.40
; DSS resource records
server    2h    IN   DSS  192.0.2.10     0   0    0    4   5    TCP   HTTP   2000 5    5000 0
                     DSS  203.0.113.30   0   127
                     DSS  203.0.113.40   0   100  255
                     DSS  192.0.2.10     0   0    0    4   10   TCP   SMTP   3000 10   5000 0
```

Figure 1 Example of DSS method implementation

$$METRIC_{(i)} = \\ = \begin{cases} \dfrac{LOAD_{(i)}}{\max\limits_{j\in N} LOAD_{(j)}} + \dfrac{IMPACT_{(i)}}{255} \times \dfrac{RESPONSE_{(i)}}{\max\limits_{j\in N} RESPONSE_{(j)}}, \\ \forall i \in \{1,..., N\} \forall \max\limits_{j\in N} LOAD_{(j)} > 0, \\ \\ \dfrac{IMPACT_{(i)}}{255} \times \dfrac{RESPONSE_{(i)}}{\max\limits_{j\in N} RESPONSE_{(j)}}, \\ \forall i \in \{1,..., N\} \forall \max\limits_{j\in N} LOAD_{(j)} > 0. \end{cases} \qquad (1)$$

The maximum value of a DNS metric for some IP address ($METRIC_{(i)}$) is 2, which can be obtained as a sum of the maximum metric of the server load $LOAD_{(i)} = \max\limits_{j\in N} LOAD_{(j)}$ and the maximum metric of server response time $RESPONSE_{(i)} = \max\limits_{j\in N} RESPONSE_{(j)}$, along with the maximum value of impact of response time on the calculation of the composite metric ($IMPACT_{(i)}$ = 255). The minimum value of the DNS metric is 0, for the minimum value of the server load metric ($LOAD_{(i)}$ = 0) and minimum value of impact of response time on the calculation of the composite metric ($IMPACT_{(i)}$ = 0).

Fig. 2 shows the impact of changing the parameter RESPONSE, which is changing from the starting value 0 ms to the final value 30 ms, and the parameter IMPACT, which is changing from his minimal possible value of 0 to his maximum value of 255, on the calculation of the DNS metric. Parameter maxLOAD has a value of 0 so the server load is not considered.

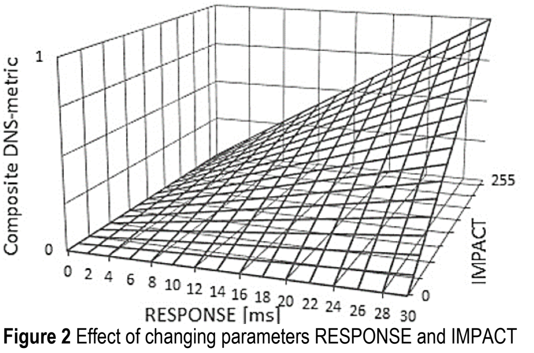

Figure 2 Effect of changing parameters RESPONSE and IMPACT

4 IMPLEMENTATION AND ANALYSIS OF THE EFFECTIVENESS OF THE DSS METHOD

The DSS method is primary developed for small to medium scale network services with dynamic, on demand, unique generated content based on client requests to network service. In situations where a client can choose a particular server or server cluster (represented to the client as a single server with one IP address), the DSS returns to the client two or more, but limited number of unicast and/or anycast IP addresses of servers with required network service for optimal server selection on the client side. Thereby servers and server clusters typically have different topological distances to the client and/or different server loads and some implementations of the DSS method would require additionally exposed IP addresses to the clients for better server granularity.

To analyse the efficiency of the DSS method, practical measurements were made in real conditions, using six ISPs that make up their own autonomous systems on the client side and the two servers connected to the Internet by two independent communication links each. According to Fig 3, the server server1.example.com is available over two independent internet links and has a public IP address on each of them: ISP 1: 8/8 Mbps, IP: IP_1_1 and ISP 2: 10/10 Mbps, IP: IP_2_1.

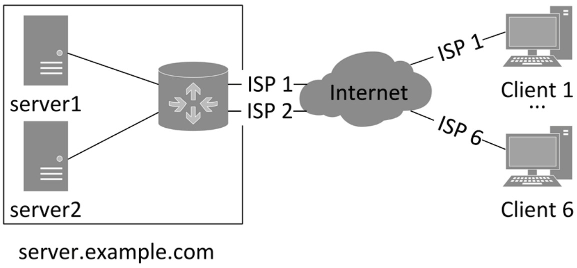

Figure 3 Test environment

The server server2.example.com is available over two public IP addresses, ISP 1: IP: IP_1_2 I ISP 2 IP: IP_2_2. Selection of server and internet link is done on the client side by selecting the destination IP address of the server (IP_1_1, IP_2_2, IP_1_2 or IP_2_2). Selection of the outgoing internet link is done on the server router by using policy-based routing based on the source IP address of the server.

Servers (OS Windows Server 2008 R2 x64) have an active HTTP service on TCP port 80 and enabled ICMP protocol. For servers server1.example.com and server2.example.com in the DNS zone example.com are defined 4 A RRs for FQDN (fully qualified domain name) server.example.com. with IP addresses IP_1_1, IP_2_2, IP_1_2 IP_2_2.

Measurements have been made for accessing the servers server.example.com, connected by two internet links, from six clients belonging to six different autonomous systems. Clients are geographically located within a radius of 3 kilometres from the servers server.example.com, thus excluding geopositioning, which is usually defined on a country, region or city level.

Measurements on clients (Windows 7 x86/x64) collected the following data:

- the number of hops from the host to the server (using the tool Traceroute)
- three RTT measurements with the 8 ICMP packets, with packet size of 16, 32 and 64 bytes respectively and sending interval of 10 ms
- transfer time for a file of the size of 256 KB, 1MB and 4MB respectively and response time of a web page that contains demanding CPU calculation, using both internet links and first low then high loaded server.

4.1 Impact of the Server Load (LOAD Parameter) on the Service Response Time

Since the server load can be a parameter in calculating the composite DNS-metrics using the LOAD parameter, the influence of the server load on service response time is observed. LOAD parameter allows the description of the server load within the priority class. The value 0 represents the lowest and the value 255 represents the highest server load. Parameter is defined by the server administrator and depends on the type of network service. It can be based on the CPU load, disk IO load, memory usage, network load, etc. or any combination of server's performance parameters.

The response time for the HTTP service of the server1.example.com is measured, available on the IP addresses IP_1_1 and IP_2_1, during low and high server load where the LOAD parameter for server load is based on the CPU load, other system resources in all measurements are unburdened. The CPU load is in all measurements controlled. It has measured the response time difference of requests to the HTTP service in two cases:

- downloading a file the size of 256 KB, 1 MB and 4 MB by using HTTP service from the server server1.example.com with very low server's processor usage (average CPU load 1%) and with full loaded processor (average CPU load 100%), shown in Fig. 4.
- loading a web page containing mathematical calculation by using the HTTP service from the server server1.example.com with a low (1%) and then high (100%) loaded processor, shown on Fig. 5.

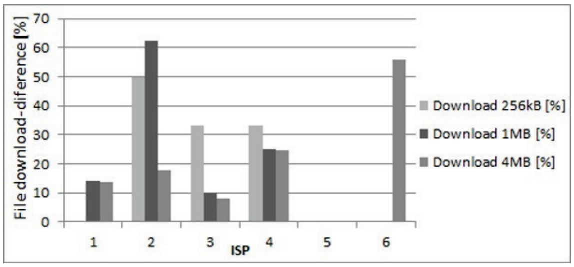

Figure 4 File download: difference high – low loaded server for IP_2_1

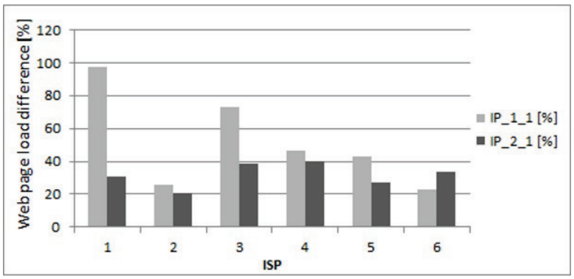

Figure 5 Loading a web page with CPU calculation, difference high – low loaded server

Measurements results show a significant impact of the processor load on service response time. The average increase of the response time for downloading a file from the IP address IP_1_1 is 23.9% with a value range from 0% do 110%, while the average increase of the response time for downloading from the IP_2_1 is 19.4% with a value range from 0% to 62.5%.

Measurements results of loading a web page containing mathematical calculation, which is based on intensive CPU usage, show a significant average increase of service response time in relation to the increase in response time for downloading a file. The average increased response time of loading a web page with a mathematical calculation from the IP address IP_1_1 is 52.4%, with the value range from 22.8% to 97.5%, while the average increased response time for the IP address IP_2_1 is 31.9%, with the value range from 21% to 39.7%.

In both cases the impact of the server load, in this particular case the processor load, on the response time was confirmed thus also confirms the necessity of the LOAD parameter in the calculation of the DNS-metric.

4.2 Impact of the Network Response Time (RESPONSE Parameter) on the Service Response Time

To determine the impact of the network response time (parameter RESPONSE) on the service response time, it is observed difference of the RTT to the server server1.example.com when it is accessed over the internet links ISP 2 (IP_2_1) and ISP 1 (IP_1_1) and the difference of the service response time for downloading a file of the size 256 kB, 1 MB and 4 Mb for a low and a high loaded server, as shown on Fig. 6 and Fig. 7.

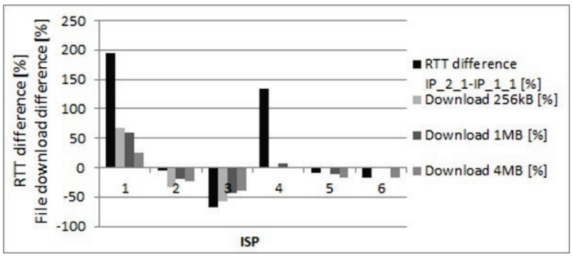

Figure 6 File download: difference IP_2_1-IP_1_1 for low loaded servers

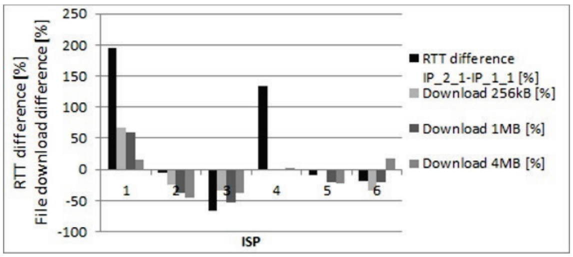

Figure 7 File download: difference IP_2_1-IP_1_1 for high loaded servers

Measurements results show strong relationship of the network response time to the service response time. A server that has a shorter RTT responds to request quickly, which means that the client can download the file faster from a server to which has a shorter RTT. The results show the same trend of RTT influence on service response time when accessing both low and high loaded servers. Measurements confirmed the impact of the network response time on the service response time thus also necessity of the RESPONSE parameter in the calculation of the DNS-metric.

4.3 Determining the Optimal Server to Access a Multiple Available Network Service

To determine the optimal server to access a multiple available network service using proposed DSS method, the server selection results by using the DSS method are compared with four common methods of server selection: Geographical - static, based on geographical distance; Hops - static, based on the number of hops; Random - dynamic, based on a random server selection (round robin DNS) and RTT - dynamic, based on RTT measurements.

The results for other four observed methods are calculated from measurements result of the DSS method.

The DSS method is implemented in a manner that the LOAD parameter to the server server1.example.com is set to the value 0 (minimum loaded server), while the server server2.example.com LOAD parameter value is set to 255 (maximum loaded server), according to the actual server workload. The IMPACT parameter is set to a maximum value of 255 (maximum impact of the RESPONSE parameter) for both servers.

The measurements results and the calculations for all five methods, shown in Tab. 1, show that in the case of access multiple available network service, available on two servers accessible by two internet links each, the Hops method is least favourable because it only gave the optimal IP address in 1/3 of the cases (in two out of six cases). The Random, Geographical and RTT methods all gave the optimal IP addresses to the multiple available network service only in 50% of the cases. The DSS method gave the optimal IP address in all 6/6 cases, with a 100% accuracy.

Table 1 Measurement results for two servers on two internet links each

	ISP 1	ISP 2	ISP 3	ISP 4	ISP 5	ISP 6
Number of hops for IP_1_1 and IP_1_2 (ISP 1)	10	8	19	5	11	8
Number of hops for IP_2_1 and IP_2_2 (ISP 2)	11	4	11	12	12	9
RTT for IP_1_1 and IP_1_2 (ms) (average)	21	17	85	23	11	11
RTT for IP_2_1 and IP_2_2 (ms) (average)	62	16	28	54	10	9
Download 1MB from IP_1_1 (low CPU load) (s)	2,2	1	1,8	1,5	0,9	1
Download 1MB from IP_2_1 (low CPU load) (s)	3,5	0,8	1	1,6	0,8	0,8
Download 1MB from IP_1_2 (high CPU load) (s)	2,5	2,1	2,3	2	1	1
Download 1MB from IP_2_2 (high CPU load) (s)	4	1,3	1,1	2	0,8	0,8
The shortest time for transfer of 1MB file (s)	2,2	0,8	1	1,5	0,8	0,8
Optimal server (the shortest transfer of 1MB file)	IP_1_1	IP_2_1	IP_2_1	IP_1_1	IP_2_1	IP_2_1
Composite DNS-metric for IP_1_1	0,55	0,47619	0,78261	0,75	0,9	1
Composite DNS-metric for IP_2_1	0,875	0,38095	0,43478	0,8	0,8	0,8
Composite DNS-metric for IP_1_2	1,625	2	2	2	2	2
Composite DNS-metric for IP_2_2	2	1,61905	1,47826	2	1,8	1,8
The smallest composite DNS-metrics	0,55	0,38095	0,43478	0,75	0,8	0,8
DSS	IP_1_1	IP_2_1	IP_2_1	IP_1_1	IP_2_1	IP_2_1
Geographical (same as Random)	IP_1_1	IP_2_1	IP_1_2	IP_2_2	IP_1_1	IP_2_1
Hops	IP_1_1	IP_2_1	IP_2_2	IP_1_2	IP_1_1	IP_1_2
Random	IP_1_1	IP_2_1	IP_1_2	IP_2_2	IP_1_1	IP_2_1
RTT	IP_1_1	IP_2_1	IP_2_2	IP_1_2	IP_2_1	IP_2_2

In the process of determining the optimal IP address for a multiple A RRs a conventional round robin DNS mechanism of A RRs sorting is assumed as well as sequential DNS queries from ISP1 to ISP 6 for Hops, RTT and Random methods.

Tab. 2 shows the summary time needed to download a 1MB file for all 6 ISPs for 5 observed methods. The RTT method has the least increase in download time in relation to the DSS method, which is 8.5%. The Hops method extends the total time of file transfer by 12.7%. Geographical/Random method extends the total file download time by 26.8% compared to the DSS method. Because of same geographic location for all six clients Geographic method turned into Random.

Table 2 The summary time needed to download a 1MB file for all 6 ISPs

Method	Download time (s)	Download time increasing (%)
DSS	7,1	-
Geographical	9,0	26,8
Hops	8,0	12,7
Random	9,0	26,8
RTT	7,7	8,5

5 CONCLUSION

User demands for a high availability and as fast as possible network service response time require redundant servers for network service, wherein the servers are very often physically distributed. Thus the access to a particular server can be based on the current load of the server and/or the communication link and allow the client to select the server with fastest service response time.

The DSS method for selecting the optimal server of multiple available network service allows a dynamic selection of servers based on the information of the server load and its network topological distance to the client. The DSS method adds a new DNS RR, size of 32 bytes for each server, in the additional section of the DNS message. The server selection is based on a calculation of the composite DNS-metric on the client side where the servers are ranked by using the network response time parameter, which the client is measuring for each server individually, and the server load parameter, that is specified by the server/service administrator and forwarded to the client, together with the rules for the calculation of the DNS-metric.

In this article the need for a new method for a client side dynamic server selection of multiple available network service has been established. Measurement results confirmed the effects of network response time and server load on the service response time. The DSS method showed shortening of service response time in relation to other four observed methods, which in examined case, with two servers on the two links each, is from 8.5% to 26.8%.

The DSS method can be used as only one or as an additional method for optimal server selection. It does not exclude the use of any of existing methods, it combines two dynamic parameters, one from the server side (server load) and one from the client side (network distance represented by RTT) in the form of a single, composite DNS-metric controlled by the service/server administrator. DSS method can be implemented quickly and easily by using an existing DNS infrastructure for transfer data to the client and can use existing DNS functionality for dynamic parameters updates.

Future work will include development of an analytical model for calculating the response time of the network service for determining the efficiency index of the DSS method.

6 REFERENCES

[1] Sonderegger, J., Blomberg, O., Milne, K., & Palislamovic, S. (2009). *JUNOS High Availability*. O'Reilly Media Inc.

[2] Hewlett-Packard WAN Design Guide The Lower Layers. ProCurve Networking by HP, 2005.

[3] Cisco Systems Computer networks, Domain Name System, Network servers. Sec. Edition, Cisco Press, 1999.

[4] Cisco Systems CCNA Routing and Switching curriculum. Cisco Networking Academy Program, 2013.

[5] Wen, Y. (2001). *Enterprise IP LAN/WAN Design*. Version 1.1, TAOS.

[6] Cahn, R. S. (1998). *Wide Area Network Design: Concepts and Tools for Optimization*. Kaufmann Publishers.

[7] RFC 1034: Domain Names - Concepts and Facilities

[8] Crovella, M. E. & Carter, R. L. (1995). *Dynamic Server Selection in the Internet*. Technical Report, Boston University. https://doi.org/10.1109/HPCS.1995.662022

[9] Carter, R. L. & Crovella, M. E. (1997). Server Selection using Dynamic Path Characterization in Wide-Area Networks. *INFOCOM 1997 Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings* / IEEE Vol. 3, 1014-1021. https://doi.org/10.1109/INFCOM.1997.631117

[10] Fei, Z., Bhattacharjee, S., Zegura, E. W., & Ammar, M. H. (1998). A novel server selection technique for improving the response time of a replicated service. *INFOCOM 1998 Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings* / IEEE Vol. 2, 783-791. https://doi.org/10.1109/INFCOM.1998.665101

[11] Carter, R. L. & Crovella, M. E. (1999). *On the network impact of dynamic server selection*. Computer Networks 31, Elsevier Science B.V., 2529-2558. https://doi.org/10.1016/S1389-1286(99)00119-X

[12] Shimokawa, T., Yoshida, N., & Ushijima, K. (2000). Flexible server selection using DNS. *ICDCS Workshop on Internet*, A76-A81.

[13] Shimokawa, T., Yoshida, N., & Ushijima, K. (2000). DNS-based Mechanism for Policy-added Server Selection. *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR*.

[14] Shaikh, A., Tewari, R., & Agrawal, M. (2001). On the Effectiveness of DNS-based Server Selection. *INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings* / IEEE Vol. 3. https://doi.org/10.1109/INFCOM.2001.916678

[15] Pan, J., Hou, Y. T., & Li, B. (2003). An overview of DNS-based server selections in content distribution networks. *Computer Networks, 43*, 695-711. https://doi.org/10.1016/S1389-1286(03)00293-7

[16] Mase, K., Kuribayashi, T., & Tsuno, A. (2004). A Dynamic Server Selection Method Using QoS Statistics. *Electronics and Communications in Japan, Part 1, 87*(7), Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J86-B, No. 3(2003), pp. 499-510. https://doi.org/10.1002/ecja.10168

[17] Cai, L., Ye, J., Pan, J., Shen, X. (Sherman), & Mark, J. W. (2006). Dynamic server selection using fuzzy inference in content distribution networks. *Computer Communications, 29*, 1026-1038. https://doi.org/10.1016/j.comcom.2005.06.001

[18] Alzoubi, H. A., Rabinovich, M., Spatscheck, O. (2007). MyXDNS: A Request Routing DNS Server with Decoupled Server Selection. *Proceedings of the 16th international conference on World Wide Web*, 351-360. https://doi.org/10.1145/1242572.1242620

[19] Mendonc, N. C., Silva, J. A. F., & Anido, R. O. (2008). Client-side selection of replicated web services: An empirical assessment. *The Journal of Systems and Software, 81*, 1346-1363. https://doi.org/10.1016/j.jss.2007.11.002

[20] Bhardwaj, S. K. & Malhotra, J. S. (2011). Simulation and comparison of hashing techniques in CDN DNS. *International Journal of Engineering Science & Technology, 3*(4).

[21] Nicholes, M. O., Chuah, C., Wub, S. F., & Mukherjee, B. (2011). Inter-domain collaborative routing (IDCR): Server selection for optimal client performance. *Computer Communications, 34*, 1798-1809. https://doi.org/10.1016/j.comcom.2011.04.002

[22] Jin, Y., Yamai, N., Okayama, K., & Nakamura, M. (2012). An Adaptive Route Selection Mechanism per Connection Based on Multipath DNS Round Trip Time on Multihomed Networks. *Journal of Information Processing, 20*(2), 386-395. https://doi.org/10.2197/ipsjjip.20.386

[23] Mackus, T., Simonaitis, T., & Tamulioniene, D. (2012). Adaptive TTL based Approach to Balance DNS Server Load. *Electronics and Electrical Engineering, 117*(1). https://doi.org/10.5755/j01.eee.117.1.1058

[24] Otto, J. S., Sánchez, M. A., Rula, J. P., Stein, T., & Bustamante, F. E. (2012). Namehelp: Intelligent Client-Side DNS Resolution. *ACM SIGCOMM Computer Communication Review Special Issue, 42*(4), 287-288. https://doi.org/10.1145/2377677.2377734

[25] Poese, I., Frank, B., Ager, B., Smaragdakis, G., Uhlig, S., & Feldmann, A. (2012). Improving Content Delivery with PaDIS. *Internet Computing, IEEE, 16*(3), 46-52. https://doi.org/10.1109/MIC.2011.105

[26] Frank, B. (2014). Dynamic content delivery infrastructure deployment using network cloud resources. Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, *Doctoral Thesis*.

[27] Liu, L., Zhou, C., Zhang, X., Guo, Z., & Li, C. (2014). Probabilistic chunk scheduling approach in parallel multiple-server dash. *IEEE Visual Communications and Image Processing Conference*, 5-8. https://doi.org/10.1109/VCIP.2014.7051490

[28] Zhang, S., Li, B., & Li, B. (2015). Presto: Towards fair and efficient http adaptive streaming from multiple servers. *IEEE International Conference on Communications*, *IEEE Xplore*, 6849-6854. https://doi.org/10.1109/ICC.2015.7249417

[29] Bouten, N., Claeys, M., Van Poecke, B., Latrey, S., & De Turck, F. (2016). Dynamic Server Selection Strategy for Multi-server HTTP Adaptive Streaming Services. *12th International Conference on Network and Service Management*, Montreal, 201-209. https://doi.org/10.1109/CNSM.2016.7818403

Contact information:

Dražen TOMIĆ, PhD student
Faculty of Electrical Engineering, Computer Science and Information
Technology in Osijek
Kneza Trpimira 2b, 31000 Osijek, Croatia
E-mail: drazen.tomic@etfos.hr

Drago ŽAGAR, PhD, Full Professor
Faculty of Electrical Engineering, Computer Science and Information
Technology in Osijek
Kneza Trpimira 2b, 31000 Osijek, Croatia
E-mail: drago.zagar@ferit.hr