



- (51) **International Patent Classification:**
H04L 29/12 (2006.01)
- (21) **International Application Number:**
PCT/CN2014/080626
- (22) **International Filing Date:**
24 June 2014 (24.06.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
201310257163.3 25 June 2013 (25.06.2013) CN
- (71) **Applicant: TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED** [CN/CN]; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN).
- (72) **Inventors: ZHOU, Hongchang**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **YIN, Shengyu**; Room 403, East

Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN). **MAI, Shiyong**; Room 403, East Block 2, SEG Park, Zhenxing Road, Futian District, Shenzhen, Guangdong 518000 (CN).

(74) **Agent: ADVANCE CHINA IP LAW OFFICE**; Room 3901, No.85 Huacheng Avenue, Tianhe District, Guangzhou, Guangdong 510623 (CN).

(81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[Continued on next page]

(54) **Title:** DOMAIN NAME RESOLUTION METHOD, DOMAIN NAME SERVER, AND DOMAIN NAME ACCESS METHOD, SYSTEM AND TERMINAL

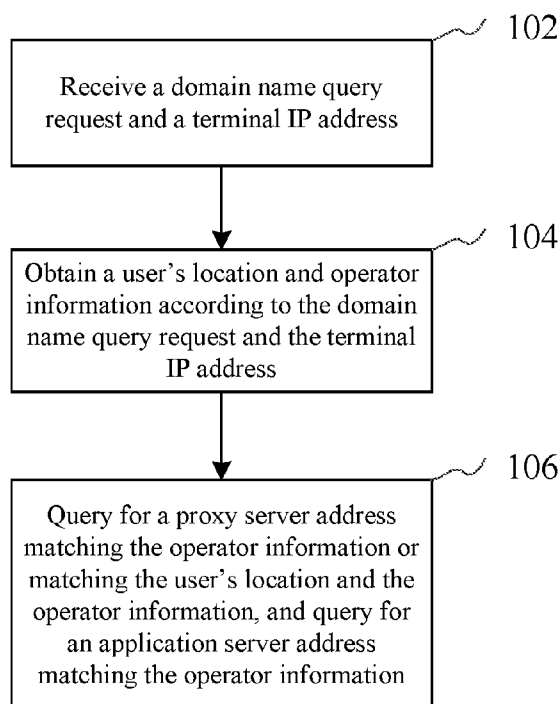


Fig. 1

(57) **Abstract:** A domain name resolution method is provided. The method includes: receiving a domain name query request and a terminal IP address; obtaining, according to the domain name query request and the terminal IP address, user's location and operator information; and querying for a proxy server address matching the operator information or matching the user's location and the operator information, and querying for an application server address matching the operator information. The method allows improving of domain name resolution speed and thus domain name access speed. In addition, a domain name server, domain name access method, system and terminal are provided.



(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

DOMAIN NAME RESOLUTION METHOD, DOMAIN NAME SERVER, AND DOMAIN NAME ACCESS METHOD, SYSTEM AND TERMINAL

CROSS REFERENCE TO RELATED APPLICATION

This application claims the priority benefit of Chinese Patent Application No. 201310257163.3, filed on June 25, 2013, the contents of which are incorporated by reference herein in their entirety for all purposes.

FIELD OF THE INVENTION

The present disclosure relates generally to the field of Internet technology, and more particularly to a domain name resolution method, domain name server, domain name access method, system and terminal, as well as non-transitory computer storage mediums respectively for performing domain name resolution and domain name access.

BACKGROUND OF THE INVENTION

Before accessing a domain name, domain name resolution is needed. In conventional techniques, the work of domain name resolution is normally done by the Domain Name System (DNS). DNS is a core service of the Internet, which can realize the mapping of domain name and IP (Internet protocol, the interconnection protocol between networks) address in a distributed database. Thus, it allows the users to access Internet more conveniently, without having to remember the machine-readable IP number strings.

However, it is found that at least the following problems exist in the prior art: the existing domain name system is a distributed database, and according to the existing domain name resolution methods, when a local domain name server cannot resolve a domain name, then it has to query for other domain name servers for resolution of the domain name by constant iteration until it is resolved. This will lead to a lot of interaction, thereby reducing the speed of domain name resolution, and further affecting the speed of accessing the domain names.

Therefore, heretofore unaddressed needs exist in the art to address the aforementioned deficiencies and inadequacies.

SUMMARY OF THE INVENTION

To address the aforementioned deficiencies and inadequacies, there is a need to provide a domain name resolution method, a domain name server, a domain name access method, and system and terminal thereof.

According to one aspect of the present disclosure, a domain name resolution method includes:

- receiving a domain name query request and a terminal IP address;

- obtaining, according to the domain name query request and the terminal IP address, user's location and operator information; and

- querying for a proxy server address matching the operator information or matching the user's location and the operator information, and querying for an

application server address matching the operator information.

According to another aspect of the present disclosure, a domain name server includes:

- a request and IP address receiving module, configured to receive a domain name query request and a terminal IP address;

- an attribute obtaining module, configured to obtain, according to the domain name query request and the terminal IP address, user's location and operator information;

- a proxy server address querying module, configured to query for a proxy server address matching the operator information or matching the user's location and the operator information; and

- an application server address querying module, configured to query for an application server address matching the operator information.

According to a further aspect of the present disclosure, a domain name access method includes:

- requesting a domain name server to query for a domain name, and receiving an application server address and a proxy server address returned by the domain name server;

- obtaining pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server;

- connecting to the application server corresponding to the application server address using the link with less network delay, and sending to the application server a data processing request.

According to another further aspect of the present disclosure, a terminal includes:

- a domain name resolution requesting module, configured to request a domain name server to query for a domain name, and to receive an application server address and a proxy server address returned by the domain name server;

- a link information obtaining module, configured to obtain pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server; and

- a data processing requesting module, configured to connect to the application server corresponding to the application server address using the link with less network delay, and to send to the application server a data processing request.

According still a further aspect of the disclosure, a domain name access system includes:

- a terminal, configured to send, to a domain name server, a domain name query request and a terminal IP address;

- a domain name server, configured to receive the domain name query request and the terminal IP address, to obtain user's location and operator information according to the domain name query request and the terminal IP address, to query for a proxy server address matching the operator information or matching the user's location and

the operator information, and to query for an application server address matching the operator information; and

a proxy server, configured to receive data packet sent by the terminal, to forward the data packet to the application server, to receive the data packet returned by the application server, and to return the data packet back to the terminal;

wherein the terminal is further configured to receive an application server address and a proxy server address returned by the domain name server, and to obtain pre-stored link information, wherein the link information contains comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server;

the terminal is further configured to connect to the application server corresponding to the application server address using the link with less network delay, and to send to the application server a data processing request.

According to still a further aspect of the disclosure, a non-transitory computer-readable storage medium for storing computer-executable instructions is provided. The computer-executable instructions are configured to execute a method for domain name resolution, wherein the method includes the steps of:

receiving a domain name query request and a terminal IP address;

obtaining, according to the domain name query request and the terminal IP address, user's location and operator information; and

querying for a proxy server address matching the operator information or matching the user's location and the operator information, and querying for an application server address matching the operator information.

According to still a further aspect of the disclosure, a non-transitory computer-readable storage medium for storing computer-executable instructions is provided. The computer-executable instructions are configured to execute a method for domain name access, wherein the method includes the steps of:

requesting a domain name server to query for a domain name, and receiving an application server address and a proxy server address returned by the domain name server;

obtaining pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server;

connecting to the application server corresponding to the application server address using the link with less network delay, and sending to the application server a data processing request.

By the above domain name resolution method and domain name server, the proxy server address and the application server address are queried based on the domain name query request and the terminal IP address. Compared with the prior art, the interaction process is reduced, thereby improving the resolution speed of domain names. By connecting to the application server using the link with less network delay, the above method, system and terminal improve the efficiency of the domain name access, and allow the terminal to receive the data processing result returned from the application server more quickly.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a flowchart showing a domain name resolution method in one embodiment of the present disclosure.

Fig. 2 is a flowchart showing Step 104 of Fig. 1.

Fig. 3 is a flowchart showing a domain name resolution method in another embodiment of the present disclosure.

Fig. 4 is an application scene of a domain name resolution method in one embodiment of the present disclosure.

Fig. 5 is a flowchart showing a domain name resolution method in yet another embodiment of the present disclosure.

Fig. 6 is a structural block diagram showing a domain name server in one embodiment of the present disclosure.

Fig. 7 is a structural block diagram showing an attribute obtaining module of Fig. 6.

Fig. 8 is a structural block diagram showing a domain name server in another embodiment of the present disclosure.

Fig. 9 is a flowchart showing a domain name access method in one embodiment of the present disclosure.

Fig. 10 is a flowchart showing a domain name access method in another embodiment of the present disclosure.

Fig. 11 is an application scene of a domain name access method in one embodiment of the present disclosure.

Fig. 12 is a structural block diagram showing a terminal in one embodiment of the present disclosure.

Fig. 13 is a structural block diagram showing a terminal in another embodiment of the present disclosure.

Fig. 14 is a structural block diagram showing a terminal in yet another embodiment of the present disclosure.

Fig. 15 is a structural block diagram showing a domain name access system in one embodiment of the present disclosure.

Fig. 16 is a structural block diagram showing a domain name access system in another embodiment of the present disclosure.

Fig. 17 is a structural block diagram showing a domain name access system in yet another embodiment of the present disclosure.

Fig. 18 is a structural schematic diagram showing a device for accessing domain name in one embodiment of the present disclosure.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following description of embodiments, reference is made to the accompanying drawings which form a part hereof, and in which it is shown by way of illustration specific embodiments of the disclosure that can be practiced. It is to be understood that other embodiments can be used and structural changes can be made without departing from the scope of the disclosed embodiments.

As shown in Fig. 1, a domain name resolution method is provided in accordance

with one embodiment. The method is illustrated by way of example of implementation in a variety of computers. The computers include but not limited to personal computer, supercomputer, mainframe computer or computer cluster, etc. The method includes the following steps.

Step 102, receiving a domain name query request and a terminal IP address.

In this embodiment, the domain name server can receive domain name query requests and terminal IP addresses sent by a variety of terminals, these terminals including but not limited to desktop computer, laptop portable computer, personal digital assistant, tablet PC, smart phone, e-book reader device, MP3 (Moving Picture Experts Group Audio Layer III) or MP4 (Moving Picture Experts Group Audio Layer IV) player, POS terminal and on-board computer, etc.

Step 104, obtaining, according to the domain name query request and the terminal IP address, user's location and operator information.

Domain name requested to query for is specified in the domain name query request. In addition, an operator can also be specified in the request, and the queried domain name and the specified operator can be obtained according to the request. A nationwide IP database, storing IP addresses of the whole country accurate to the location of the users and operators, can be configured in advance in the domain name server, wherein user's location is preferably the province that a user belongs to. Therefore, the user's location and operator corresponding to the terminal IP address can be obtained according to the terminal IP address.

In one embodiment, as shown in Fig. 2, Step S104 further includes:

Step 202, obtaining the corresponding user's location and the operator according to the terminal IP address.

In this embodiment, the corresponding user's location and the operator can be obtained according to the IP address stored in the IP library. a user's location can be the province that a user belongs to, and the operators may include China Unicom, China Telecom and China Mobile, etc.

Step 204, determining whether the specified operator is obtained from the domain name query request; if so, go to Step 206, otherwise go to Step 208.

Step 206, it is determined that the operator information contains information of the specified operator.

Step 208, it is determined that the operator information contains information of the operator obtained according to the terminal IP address.

In the embodiment, when sending a domain name query request by the terminal, the operator can be specified in the request. If the specified operator can be obtained from the request, the operator in the operator information will be the specified operator. If no specified operator can be obtained from the request, the operator in the operator information will be the operator obtained according to the terminal IP address.

Step 106, querying for a proxy server address matching the operator information or matching the user's location and the operator information, and querying for an application server address matching the operator information.

A proxy server address can be the IP address of a proxy server, and an application server address can be the IP address of an application server. A proxy

server can serve as an access point, by which the data processing request from the terminal can be forwarded to the application server. The application server, i.e., the server corresponding to the application program on the terminal, is configured to process the data processing request sent by the terminal, and to return the processing result to the terminal.

Specifically, in one embodiment in Step 106, it is possible to query for the proxy server address of which the operator is the same as that contained in the operator information, or to query for the proxy server address of which the location is the same as the user's location, and the operator is the same as that contained in the operator information. In the embodiment, a lot of proxy server addresses accurate to the user's province and the operator are pre-stored in the domain name server.

In one embodiment, in Step 106, application server addresses corresponding to the domain name requested to query for can be obtained; then, the application server address of which the operator is the same as that contained in the operator information is queried from the application server addresses corresponding to the domain name requested to query.

In the embodiment, the correspondence relations between domain name and application server address can be pre-stored in the domain name server, in which one domain name can correspond to multiple addresses accurate to the province and operator. Corresponding application server address can be obtained based on the domain name requested to query. Then, the application server address of which the operator is the same as that contained in the operator information is queried from the application server addresses. By obtaining in advance the application server addresses corresponding to the domain name, it is possible to limit the range of query to these addresses, thereby improving the query efficiency.

In this embodiment, upon receiving a domain name query request, the proxy server address and application server address are queried and returned back to the terminal. Based on the received proxy server address and the application server address, a data processing request can then be sent by the terminal to the corresponding proxy server or application server. Compared to the prior art, the interaction process is reduced, and the speed of the domain name resolution is improved.

As shown in Fig. 3, in another embodiment, a domain name resolution method is provided. The method includes the following steps.

Step 302, receiving a domain name query request and a terminal IP address.

Step 304, obtaining, according to the domain name query request and the terminal IP address, user's location and operator information.

Step 306, obtaining the application server address the speed test server address corresponding to the domain name requested to query. The speed test server address can be the IP address of the speed test server. In this embodiment, the speed test server and the application server are deployed in the same server room, that is, the server room where an application server is in is deployed with a corresponding test server. The correspondence relations between domain name and speed test server address can be pre-stored in the domain name server, in which one domain name can correspond to one speed test server addresses.

Step 308, querying for the proxy server address matching the operator information or matching the user's location and the operator information.

Step 310, querying for, from the application server addresses corresponding to the domain name requested to query for, the application server address of which the operator is the same as that contained in the operator information.

In the embodiment, the speed test server address obtained can be returned back to the terminal. The terminal can then access the corresponding speed test server via different links, compare the corresponding network delays to obtain a link with less delay, thereby determine the link for the next access to the application server.

Following is a specific application scene to illustrate the principle of the method according to an embodiment of the present disclosure. As shown in Fig. 4, the terminal 402 sends to multiple domain name servers 404 domain name query request. Preferably, the domain name query request can be sent to the domain name servers 404 of a same operator; since the results returned by all domain name servers are the same, the fastest result obtained by the domain name server 404 can be returned to the terminal 402. The domain name servers 404 receives the domain name query request and the terminal IP address sent by the terminal 402. As long as the application server address, the proxy server address and speed test server address are firstly queried by one of the domain name servers 404, these data will be returned to the terminal 402. Upon receiving the data, the terminal 402 can send to the corresponding application server the data processing request (connecting to the application server directly or through a proxy server) according to the link information, and obtain a link with less network delay by accessing the corresponding speed test server via different links.

As shown in Fig. 5, in one embodiment, a domain name resolution method is provided. The method includes the following steps.

Step 502, receiving a domain name query request and a terminal IP address.

Step 504, obtaining, according to the domain name query request and the terminal IP address, user's location and operator information.

In this embodiment, the domain name server obtains the corresponding user's location and the operator according to the terminal IP address. If the specified operator can be obtained from the domain name query request, then it is determined that the operator information contains information of the specified operator, otherwise it is determined that the operator information contains information of the operator corresponding to the terminal IP address.

Step 506, obtaining, according to the domain name, the corresponding application server address and the speed test server address.

Step 508, determining whether the domain name has been configured; if so, go to Step 512, otherwise go to Step 510.

Step 510, returning specific error codes.

Determining whether the domain name has been configured may include: determine whether it is possible to obtain the corresponding application server address and speed test server address according to the domain name; if no, return to the terminal the specific error codes indicating that the domain name is inaccessible.

Step 512, determining whether the operator is known; if so, go to Step 516,

otherwise go to Step 514.

Step 514, returning the speed test server address and all the addresses of the application servers.

Determining whether the operator is known may include: determine whether it is possible to obtain the operator according to the domain name query request and the terminal IP address; if not, the matching application server address cannot be queried from the application server addresses corresponding to the domain name; as a result, all the application server addresses and the speed test server address corresponding to the domain name are returned. Upon receiving the data, the terminal may randomly select an application server address, and send to the corresponding application server the data processing request.

Step 516, determining whether there is an address of proxy server of the same location and the same operator; if so, go to Step 524, otherwise go to Step 518.

In the embodiment, the domain name server queries whether there is an address of proxy server of which the location is the same as that of the user, and the operator is the same as that contained in the obtained operator information. For example, if the user's location is "Guangdong", and the operator contained in the obtained operator information is "China Mobile", then query an address having the location of "Guangdong" and the operator of "China Mobile" in all proxy server addresses; if no result is obtained, go to Step 518, otherwise go to Step 524.

Step 518, determining whether there are application server addresses of the same operator; if so, go to Step 522, otherwise go to Step 520.

Step 520, returning the speed test server address and all the application server addresses.

In case of no proxy server address of the same location and the same operator can be found, and no application server address of the same operator can be found—for example, operator contained in the operator information is "China Mobile", and there is no address corresponding to the operator of "China Mobile" among all the application server addresses at this time, the domain name server returns to the terminal the speed test server address and addresses of all the application servers. Upon receiving the data, it is possible to select randomly an application server address, and to send the data processing request to the corresponding application server.

Step 522, returning the speed test server address and the application server addresses of the same operator.

If there are multiple returned application servers of the same operator, the terminal may randomly select one, and send to its corresponding application server the data processing request.

Step 524, determining whether are application server addresses of the same operator; if so, go to Step 526, otherwise, go to Step 528.

Step 526, returning the speed test server address, proxy server addresses of the same location and the same operator, and the application server addresses of the same operator.

If there are multiple proxy servers of the same location and the same operator, it is possible to randomly select one. If there are multiple application servers of the

same operator, it is also possible to randomly select one.

Step 528, returning the speed test server address, all the application server addresses, and proxy server addresses of the same location and the same operator.

In the embodiment, when the proxy server address of which the location and operator are the same as that of the user is found, and the application server address of which the operator is the same as that of the user is found, the terminal can send a data processing request to the proxy server of the same location and the same operator, or to the application server of the same operator, thereby improving the efficiency of data processing.

As shown in Fig. 6, in one embodiment, a domain name server includes a request and IP address receiving module 602, an attribute obtaining module 604, a proxy server address querying module 606 and an application server address query module 608.

The request and IP address receiving module 602 is configured to receive a domain name query request and a terminal IP address.

The attribute obtaining module 604 is configured to obtain, according to the domain name query request and the terminal IP address, user's location and operator information.

The proxy server address querying module 606 is configured to query for a proxy server address matching the operator information or matching the user's location and the operator information.

The application server address query module 608 is configured to query for an application server address matching the operator information.

In one embodiment, as shown in Fig. 7, the attribute obtaining module 604 includes a terminal IP address processing module 604a, and an operator determining module 604b.

The terminal IP address processing module 604a is configured to obtain the corresponding user's location and operator based on the terminal IP address.

The operator determining module 604b is configured to determine whether a specified operator is obtained based on the domain name query request; if so, it is determined that the operator information contains information of a specified operator, otherwise it is determined that the operator information contains information of the operator obtained according to the terminal IP address.

In one embodiment, the proxy server address query module 606 is configured to query for the proxy server address of which the operator is the same as that contained in the operator information, or to query for the proxy server address of which the location is the same as that of the user, and the operator is the same as that contained in the operator information.

In one embodiment, as shown in Fig. 8, based on the embodiment of Fig. 1, the domain name server further includes a speed test server address obtaining module 610.

The speed test server address obtaining module 610 is configured to obtain the speed test server address corresponding to the domain name requested to query.

In one embodiment, the application server address querying module 610 is configured to obtain an application server address corresponding to the domain name

requested to query, and to query the application server addresses corresponding to the domain name requested to query for an application server address of which the operator is the same as that contained in the operator information.

In one embodiment, as shown in Fig. 9, a domain name access method is illustrated by way of example of implementation in a variety of terminals, including but not limited to desktop computer, laptop portable computer, personal digital assistant, tablet PC, smart phone, e-book reader device, MP3 (Moving Picture Experts Group Audio Layer III) or MP4 (Moving Picture Experts Group Audio Layer IV) player, POS terminal and on-board computer, etc. The method includes the following steps.

Step 902, requesting a domain name server to query for a domain name, and receiving an application server address and a proxy server address returned by the domain name server.

In the embodiment, the terminal can send to a domain name server a domain name query request and a terminal IP address. Upon receiving the domain name query request and the terminal IP address, the domain name server can obtain the user's location and the operator information based on the request and the address, so as to further query for the matching application server address and the proxy server address and return them to the terminal. The proxy server address found can be an address matching the operator information or matching the user's location and the operator information, and the application server address found can be an address matching the operator information. The detailed description about how the domain server query for the application server address and the proxy server address are already illustrated in the above, which will not go into detail here.

Step 904, obtaining pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server.

In the embodiment, the link information can be pre-stored in the terminal. The link information is a comparison result stored upon speed test of the previous access of the terminal to the same domain name. The so-called speed test refers to test of network delay between two links including direct connection to the application server and connection to the application server through the proxy server, which is performed to find out the link with less delay.

Step 906, connecting to the application server corresponding to the application server address using the link with less network delay, and sending to the application server a data processing request.

Regarding the link information, when the network delay of direct connection is smaller, then direct connect to the application server in accordance with the received application server address; when the network delay of connection through a proxy server is smaller, then connect to the application server in accordance with the received application server by the proxy server in accordance with the received proxy server address. That is, the data processing request is sent to the proxy server, and forwarded to the application server by the proxy server.

In the embodiment, by selecting the link with less network delay, the efficiency

of the domain name access is improved, such that the terminal can receive the processing result data returned from the application server more quickly.

In one embodiment, the above domain name access method further includes the steps of: accessing the application server corresponding to the application server address by two links respectively, and comparing the network delay of the two links, and update the link information according to the comparison result, the two links being direct connection to the application server corresponding to the application server address, and connection to the application server through the proxy server corresponding to the proxy server address.

In the embodiment, by comparing the corresponding network delay of the two links, the link with less delay can be obtained in this speed test, according to which the link information is updated. Thus, it is possible to connect the application server using the link with less delay when accessing the same domain name next time, thus improving the access speed.

In another embodiment, as shown in Fig. 10, the domain name access method includes the following steps.

Step 1002, sending to a domain name server a domain name query request, receiving an application server address, a proxy server address and a speed test server address returned by the domain name server.

In the embodiment, the terminal can request the domain name server to query for the domain name, and the domain name server can query for the application server address, the proxy server address and the speed test server address and return these addresses to the terminal, which is described above in detail and will not repeat here.

Step 1002, obtaining pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server.

In the embodiment, the link information corresponds to the stored domain names. The comparison result in the link information is obtained upon speed test of the previous access of the terminal to the same domain name.

Step 1006, connecting to the application server corresponding to the application server address using the link with less network delay, and sending to the application server a data processing request.

Step 1008, access the speed test server corresponding to the speed test server address by two links below respectively, compare the network delay of the two links, and update the link information according to the comparison result. The two links are direct connection to the speed test server corresponding to the application server address, and connection to the speed test server through the proxy server corresponding to the proxy server address.

In this embodiment, the speed test server and the application server are deployed in the same server room. Thus, considering the network delay, access to the speed test server and access to the application server are equivalent. Specifically, the terminal accesses the speed test server through different links. One link is direct connection, i.e., the terminal sends directly to the speed test server speed test data packet, and the

speed test server returns immediately speed test response packet upon receiving the speed test data packet. The other link is connection via the proxy server, i.e., the terminal sends to the proxy server speed test data packet; the proxy server forwards the speed test data packet to the speed test server; the speed test server returns immediately to the proxy server speed test response packet upon receiving the speed test data packet; and the proxy server forwards the speed test response packet to the terminal. By comparing, using the terminal, the time of each link from sending the speed test response packet to receiving the speed test response packet, the link with less network delay can be obtained.

Further, in order to reduce causal factors resulting to fluctuations in the network, the terminal can send and receive data packets in multiple rounds, and take the mean value of time for sending and receiving to obtain a link with smaller mean value. In the present embodiment, after obtaining a link with less network delay, the link information stored in the terminal may be updated correspond to the domain name, such that the link with less network delay can be used to connect the application server in the next access to the same domain.

In the embodiment, the terminal accesses the speed test server through different links. By comparing the corresponding network delay of the two links, the link with less delay can be obtained. Thus, when accessing the same domain name next time, the speed will be faster. Since the processing logic of the application server is usually transparent to the terminal, the network delay can be obtained more accurately by accessing the speed test server through different links, which helps to provide a link with less delay more accurately.

Following is a specific application scene to illustrate the domain name access method according to an embodiment of the present disclosure. As shown in Fig. 11, the terminal 1102 requests the domain name server 1104 to query for a domain name. Specifically, request the domain name server 1104 of the same operator to query for the domain name. Upon finding the matching proxy server address, the application server address and the speed test server address, the domain name server 1104 will return these addresses to the terminal 1102. The terminal 1102 determines the link for accessing the application server 1108 according to the link information stored locally. For example, in the link information, if the network delay of the link through the proxy server 1106 to the application server 1108 is less, then the terminal 1102 can connect the application server 1108 corresponding to the received application server address through the proxy server 1106 corresponding to the received proxy server address, and send to the proxy server 1106 data processing request. The proxy server 1106 receives the data processing request and forwards it to the application server 1108. The application server 1108 processes the data processing request, and returns the processing result to the proxy server 1106, which will then return the processing result to the terminal 1102.

In addition, the terminal 1102 also accesses the speed test server 1110 corresponding to the received speed test addresses through two links. The speed test server 1110 is deployed in the same room with the application server 1108. The two links are, respectively: the terminal 1102 directly accesses the speed test server 1110; the terminal 1102 accesses the speed test server 1110 through the proxy server 1106.

Comparing the corresponding network delay of the two links, the link with less delay can be obtained and the link information locally stored is updated.

As shown in Fig. 12, in one embodiment, a terminal includes a domain name resolution requesting module 1202, a link information obtaining module 1204, and a data processing requesting module 1206.

The domain name resolution requesting module 1202 is configured to request a domain name server to query for a domain name, and to receive an application server address and a proxy server address returned by the domain name server;

The link information obtaining module 1204 is configured to obtain pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server; and

The data processing requesting module 1206 is configured to connect to the application server corresponding to the application server address using the link with less network delay, and to send to the application server a data processing request.

In one embodiment, as shown in Fig. 13 and based on the embodiment of Fig. 12, the terminal further includes a first link information updating module 1208. The first link information updating module 1208 is configured to access the application server corresponding to the application server address respectively through two links, compare the corresponding network delay of the two links, and update the link information according to the comparison result, the two links being direct connection to the application server corresponding to the application server address, and connection to the application server through the proxy server corresponding to the proxy server address.

In another embodiment, the domain name resolution server 1202 is configured to receive the speed test server address returned from the domain name server. As shown in Fig. 14, based on the embodiment of Fig. 12, the terminal further includes a second link information updating module 1210.

The second link information updating module 1210 is configured to access the speed test server corresponding to the speed test server address respectively through two links, to compare the corresponding network delay of the two links, and to update the link information according to the comparison result, the two links being direct connection to the speed test server corresponding to the speed test server address, and connection to the speed test server through the proxy server corresponding to the proxy server address.

As shown Fig. 15, in one embodiment, a domain name access system includes a terminal 1302, a domain name server 1304, and a proxy server 1306.

The terminal 1302 is configured to send, to a domain name server 1304, a domain name query request and a terminal IP address.

The domain name server 1304 is configured to receive the domain name query request and the terminal IP address, to obtain a user's location and operator information according to the domain name query request and the terminal IP address, to query for a proxy server address matching the operator information or matching the user's location and the operator information, and to query for an application server address matching the operator information.

The proxy server 1306 is configured to receive data packet sent by the terminal 1302, to forward the data packet to the application server, to receive the data packet returned by the application server, and to return the data packet back to the terminal 1302.

In the embodiment, the terminal 1302 is further configured to receive an application server address and a proxy server address returned by the domain name server 1304, and to obtain pre-stored link information, wherein the link information contains comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server.

In the embodiment, the terminal 1302 is further configured to connect to the application server corresponding to the application server address using the link with less network delay, and to send to the application server a data processing request.

In one embodiment, as shown in Fig. 16 and based on the embodiment of Fig. 15, the domain name access system further includes an application server 1308.

The application server 1308 is configured to receive the user data packet sent by the terminal 1302 or the proxy server 1306, to process the user data packet, and to return the processed data packet to the terminal 1302 or the proxy server 1306.

In the embodiment, the application server 1308 is further configured to receive the speed test data packet sent by the terminal 1302 or the proxy server 1306, and to return the speed test response packet to the terminal 1302 or the proxy server 1306.

In one embodiment, as shown in Fig. 17 and based on the embodiment of Fig. 16, the domain name access system further includes a speed test server 1310.

The speed test server 1310 is deployed in the same room with the application server 1308, configured to receive the speed test data packet sent by the terminal 1302 or the proxy server 1306, and to immediately return the speed test response packet to the terminal 1302 or the proxy server 1306.

In the embodiment, the domain name server 1304 is further configured to obtain a speed test server address corresponding to the domain name requested to query. The terminal 1302 is further configured to receive the speed test server address returned by the domain name server 1304, to access the speed test server 1310 corresponding to the speed test server address by two links respectively, to compare the network delay of the two links, and to update the link information according to the comparison result, the two links being direct connection to the speed test server 1310 corresponding to the application server address, and connection to the speed test server through the proxy server 1306 corresponding to the proxy server address.

In one embodiment, the proxy server is further configured to receive the speed test data packet sent by the terminal 1302, to forward the speed test data packet to the speed test server 1310, to receive the speed test response packet returned by the speed test server 1310, and to return the speed test response packet to the terminal 1302.

It should be noted that for a person skilled in the art, partial or full process to realize the methods in the above embodiments can be accomplished by related hardware instructed by a computer program, the program can be stored in a computer readable storage medium and the program can include the process of the embodiments of the above methods. Wherein, the storage medium can be a disk, a

light disk, a Read-Only Memory or a Random Access Memory, etc.

Another domain name access means is provided according to embodiments of the present disclosure as shown in Fig. 18. For ease of clarification, only the parts relevant to the embodiments of the disclosure are illustrated. The specific technical details which were not disclosed may refer to the methods according to embodiments of the present disclosure. The terminal may include mobile phone, tablet PC, PDA (Personal Digital Assistant), POS (Point of Sales), on-board computer and any other terminal equipment. The following description takes mobile phone as an example.

Fig. 18 is a structural block diagram showing a part of a mobile phone relevant to the terminal provided according to various embodiments of the present disclosure. Referring to Fig. 18, the mobile phone includes: RF (Radio Frequency) circuit 1810, storage 1820, an input unit 1830, a display unit 1840, a sensor 1850, audio circuitry 1860, a WiFi (Wireless Fidelity) module 1870, a processor 1880, a power supply 1890 and other components. It will be appreciated by the skilled in the art that the structure of the mobile phone shown in Fig. 18 does not constitute a limitation, and the mobile phone may include more or fewer components than as shown, or a combination of some of the components, or different arrangement of components.

The various parts of the mobile phone will be described in detail with reference to Fig. 18.

The RF circuitry 1810 is configured to receive and send signals during transmission of messages or calls. Specifically, the downlink message of the base station is received and processed by the processor 1880; in addition, the designed uplink data is sent to the base station. Typically, the RF circuitry includes but not limited to, antenna, at least one amplifier, transceiver, a coupler, a low noise amplifier (LNA), and diplexer, etc. In addition, the RF circuitry 1810 can also communicate with network and other devices via wireless communication. The wireless communication can use any communication standard or protocol, including but not limited to, Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS), CDMA (CDMA), wideband CDMA (WCDMA), Long Term Evolution (LTE), email, Short Messaging Service (SMS) and the like.

The storage 1820 may store software programs and modules. By running software programs and modules stored in the storage 1820, various application functions of the mobile phone and data processing can be performed. The storage 1820 may include generally program storage area and data storage area, wherein the program storage area may store the operating system program, application program of least one function (e.g., sound playback, image playback, etc.); the data storage area may store data created during to the actual use of the phone (e.g., audio data, phone book, etc.). In addition, the memory 1820 may include high-speed random access memory, and non-volatile memory, such as at least one disk storage device, flash memory device, or other non-volatile solid-state memory devices.

The input unit 1830 may receive the inputted number or character information, and generate key signal input related to the user settings and function control of the phone. Specifically, the input unit 1830 may include a touch panel 1831 and other input devices 1832. The touch panel 1831, also known as a touch screen, can collect the user's touch operations on or near the panel (such as operations on or near the

touch panel 1831 using a finger, stylus, and any other appropriate objects or accessories), and drive the corresponding connection device according to a preset program. Optionally, the touch panel 1831 may include two portions: a touch detection device and a touch controller. The touch detection device may detect the touch position of the user, detect the signal caused by the touch operation, and send the signal to the touch controller. The touch controller may receive touch information from the touch detection device, convert it into contact coordinates and send the same as the processor 1880, and receive commands sent by the processor 1880 to execute. Further, a variety of other types of touch panels including resistive, capacitive, infrared and surface acoustic wave, etc may also be used. In addition to the touch panel 1831, the input unit 1830 may also include other input device 1832. Specifically, the other input device 1832 may be any one of the following including but not limited to: physical keyboard, function keys (e.g., volume control keys, key switches, etc.), trackball, mouse, and operating lever.

The display unit 1840 can display information inputted by the user or information to be provided to the user, and the phone's menu. The display unit 1840 may include a display panel 1841, which may optionally in the form of liquid crystal display (LCD), organic light emitting diode (OLED), etc. Further, the touch panel 1831 may cover the display panel 1841. When the touch panel 1831 detects the touch operation on or near the panel, a signal is transmitted to the processor 1880 to determine the type of touch event. The processor 1180 then provides corresponding visual output to the display panel 1841 according to the type of the touch event. Although in Fig. 18, the touch panel 1831 and the display panel 1841 are used as two separate components to realize the input and output function of the mobile phone, in some embodiments, the touch panel 1831 and the display panel 1841 can be integrated to realize the input and output function of the mobile phone.

The phone 1850 may further include at least one sensor 1850, such as light sensor, motion sensor and other sensors. Specifically, the light sensor may include an ambient light sensor and a proximity sensor, wherein the ambient light sensor can adjust the brightness of the display panel 1841 according to the brightness of the ambient light, and the proximity sensor can switch off the display panel 1841 and/or the backlight when the phone is moved close to the ear. As a movement sensor, an acceleration sensor can detect the acceleration in each direction (typically triaxial), and can the magnitude and direction of gravity when it is stationary, which can be used in applications for identifying the posture of the phone (such as horizontal and vertical screen switching, related games, magnetometer attitude calibration), and function to identify the vibration (such as pedometer, percussion) and the like. The mobile phone can be configured with other sensors, such as gyroscope, barometer, hygrometer, thermometer, and infrared sensor, etc, which will not go into detail here.

The audio circuit 1860, the speaker 1861, the microphone 1862 can provide audio interface between the user and the phone. The electrical signal, converted from the received audio data, may be transmitted by the audio circuit 1860 to the speaker 1861, and then converted by the speaker 1861 to sound signal and outputted. On the other hand, the collected sound signal is converted by the speaker 1861 into electric signal 1862, which is received by the audio circuitry 1860 and converted into audio

data, and then outputted to the processor 1880 for processing; the processed data is sent to for example another phone via the RF circuit 1810, or outputted to the storage 1820 for further processing.

WiFi is a short-range wireless transmission technology; via the WiFi module 1870, the phone can send and receive emails, browse the web and access streaming media, etc. It provides users with wireless broadband Internet access. Although Fig. 18 shows a WiFi module 1870, it is understood that it is not necessary for the mobile phone 1800 to have a WiFi module, which can be omitted without changing the nature of the invention.

The processor 1880 is the control center of the phone, which connects various parts of the phone using a variety of interfaces and wires, and achieves various functions and data processing of the phone by running or executing software programs and/or module stored in the storage 1820, so as to realize overall monitoring of the phone. Optionally, the processor 1880 may include one or more processing units; preferably, the processor 1880 may be integrated with an application processor and a modem processor, wherein processor is configured to process the operating system, the user interface and applications, while the modem processor mainly processes wireless communication. It is understood that the above modem processor may not be integrated into the processor 1880.

The phone also includes a power supply 1890 (e.g. a battery) for each component. Preferably, the power supply can logically coupled to the processor 1880 via the power management system, so as to realize the functions of charge, discharge and power management by the power management system.

Although not shown, the phone can also include a camera, a Bluetooth module, etc., which will not go into detail here.

According to various embodiments of the disclosure, the processor 1880 included in the terminal may have the function of: request the domain name server to query for the domain name, receive the application server address and the proxy server address returned by the domain name server; obtain the pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server; connect to the application server corresponding to the application server address using the link with less network delay, and send to the application server a data processing request.

In one embodiment, the processor 1880 may have the function of: access the application server corresponding to the application server address by two links respectively, compare the network delay of the two links, and update the link information according to the comparison result, the two links being direct connection to the application server corresponding to the application server address, and connection to the application server through the proxy server corresponding to the proxy server address.

In one embodiment, the processor 1880 may receive the speed test server address returned by the domain name server; access the speed test server corresponding to the speed test server address by two links respectively, compare the network delay of the two links, and update the link information according to the

comparison result, the two links being direct connection to the speed test server corresponding to the application server address, and connection to the speed test server through the proxy server corresponding to the proxy server address.

In addition, the present invention also provides a non-transitory computer-readable storage medium having computer-executable instructions for performing domain name resolution and a non-transitory computer-readable storage medium having computer-executable instructions for performing domain name access, wherein the steps included in the methods executed by the computer-executable instructions stored in the computer-readable storage mediums are the same as those described above, and will not go into details here.

The embodiments are chosen and described in order to explain the principles of the disclosure and their practical application so as to allow others skilled in the art to utilize the disclosure and various embodiments and with various modifications as are suited to the particular use contemplated. Alternative embodiments will become apparent to those skilled in the art to which the present disclosure pertains without departing from its spirit and scope. Accordingly, the scope of the present disclosure is defined by the appended claims rather than the foregoing description and the exemplary embodiments described therein.

CLAIMS

1. A domain name resolution method, comprising:
receiving a domain name query request and a terminal IP address;
obtaining, according to the domain name query request and the terminal IP address, user's location and operator information; and
querying for a proxy server address matching the operator information or matching the user's location and the operator information, and querying for an application server address matching the operator information.
2. The method of claim 1, wherein obtaining the user's location and operator information comprises:
obtaining the corresponding user's location and the operator according to the terminal IP address; and
determining whether a specified operator is obtained from the domain name query request; if so, it is determined that the operator information contains information of the specified operator; otherwise it is determined that the operator information contains information of the operator obtained according to the terminal IP address.
3. The method of claim 1 or 2, wherein querying for the proxy server address comprises:
querying for a proxy server address of which the operator is the same as that contained in the operator information; or
querying for a proxy server address of which the location is the same as the user's location, and the operator is the same as that contained in the operator information.
4. The method of claim 1, wherein the method comprises:
obtaining a speed test server address corresponding to the domain name requested to query.
5. The method of claim 1 or 4, wherein querying for an application server address comprises:
querying for an application server address corresponding to the domain name requested to query; and
querying the application server addresses corresponding to the domain name requested to query for an application server address of which the operator is the same as that contained in the operator information.
6. A domain name server, comprising:
a request and IP address receiving module, configured to receive a domain name query request and a terminal IP address;
an attribute obtaining module, configured to obtain, according to the domain name query request and the terminal IP address, user's location and operator

information;

a proxy server address querying module, configured to query for a proxy server address matching the operator information or matching the user's location and the operator information; and

an application server address querying module, configured to query for an application server address matching the operator information.

7. The server of claim 6, wherein the attribute obtaining module comprises:

a terminal IP address processing module, configured to obtain the corresponding user's location and operator based on the terminal IP address; and

an operator determining module, configured to determine whether a specified operator is obtained based on the domain name query request; if so, it is determined that the operator information contains information of a specified operator, otherwise it is determined that the operator information contains information of the operator obtained according to the terminal IP address.

8. The server of claim 6 or 7, wherein the proxy server address querying module is configured to query for a proxy server address of which the operator is the same as that contained in the operator information, or to query for the proxy server address of which the location is the same as that of the user, and the operator is the same as that contained in the operator information.

9. The server of claim 6, wherein the domain name server comprises: a speed test server address obtaining module, configured to obtain a speed test server address corresponding to the domain name requested to query.

10. The server of claim 6 or 9, wherein the application server address querying module is configured to obtain an application server address corresponding to the domain name requested to query, and to query the application server addresses corresponding to the domain name requested to query for an application server address of which the operator is the same as that contained in the operator information.

11. A domain name access method, comprising:

requesting a domain name server to query for a domain name, and receiving an application server address and a proxy server address returned by the domain name server;

obtaining pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server; and

connecting to the application server corresponding to the application server address using the link with less network delay, and sending to the application server a data processing request.

12. The method of claim 11, further comprising:

accessing the application server corresponding to the application server address by two links respectively, comparing the network delay of the two links, and updating the link information according to the comparison result, the two links being direct connection to the application server corresponding to the application server address, and connection to the application server through the proxy server corresponding to the proxy server address.

13. The method of claim 11, further comprising:

receiving the speed test server address returned from the domain name server; and

accessing the speed test server corresponding to the speed test server address by two links respectively, comparing the network delay of the two links, and updating the link information according to the comparison result, the two links being direct connection to the speed test server corresponding to the application server address, and connection to the speed test server through the proxy server corresponding to the proxy server address.

14. A terminal, comprising:

a domain name resolution requesting module, configured to request a domain name server to query for a domain name, and to receive an application server address and a proxy server address returned by the domain name server;

a link information obtaining module, configured to obtain pre-stored link information, the link information containing comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server; and

a data processing requesting module, configured to connect to the application server corresponding to the application server address using the link with less network delay, and to send to the application server a data processing request.

15. The terminal of claim 14, further comprising:

a first link information updating module, configured to access the application server corresponding to the application server address respectively through two links, comparing the corresponding network delay of the two links, and updating the link information according to the comparison result, the two links being direct connection to the application server corresponding to the application server address, and connection to the application server through the proxy server corresponding to the proxy server address.

16. The terminal of claim 14, wherein the domain name resolution requesting module is configured to receive a speed test server address returned by the domain name server;

the terminal further comprises:

a second link information updating module, configured to access the speed test server corresponding to the speed test server address respectively through two links,

to compare the corresponding network delay of the two links, and to update the link information according to the comparison result, the two links being direct connection to the speed test server corresponding to the speed test server address, and connection to the speed test server through the proxy server corresponding to the proxy server address.

17. A domain name access system, comprising:

- a terminal, configured to send, to a domain name server, a domain name query request and a terminal IP address;

- a domain name server, configured to receive the domain name query request and the terminal IP address, to obtain user's location and operator information according to the domain name query request and the terminal IP address, to query for a proxy server address matching the operator information or matching the user's location and the operator information, and to query for an application server address matching the operator information; and

- a proxy server, configured to receive data packet sent by the terminal, to forward the data packet to the application server, to receive the data packet returned by the application server, and to return the data packet back to the terminal;

- wherein the terminal is further configured to receive an application server address and a proxy server address returned by the domain name server, and to obtain pre-stored link information, wherein the link information contains comparison result of network delay of two links, the two links being direct connection to the application server and connection to the application server through the proxy server; and

- the terminal is further configured to connect to the application server corresponding to the application server address using the link with less network delay, and to send to the application server a data processing request.

18. The system of claim 17, further comprising:

- an application server, configured to receive user data packet sent by the terminal or the proxy server, to process the user data packet, and to return the processed data packet to the terminal or the proxy server; and

- the application server is further configured to receive speed test data packet sent by the terminal or the proxy server, and to return speed test response packet to the terminal or the proxy server.

19. The system of claim 17 or 18, further comprising:

- a speed test server, deployed in the same room with the application server, configured to receive the speed test data packet sent by the terminal or the proxy server, and to immediately return the speed test response packet to the terminal or the proxy server;

- wherein the domain name server is further configured to obtain a speed test server address corresponding to the domain name requested to query; and

- the terminal is further configured to receive the speed test server address returned by the domain name server, to access the speed test server corresponding to the speed test server address by two links respectively, to compare the network delay

of the two links, and to update the link information according to the comparison result, the two links being direct connection to the speed test server corresponding to the application server address, and connection to the speed test server through the proxy server corresponding to the proxy server address.

20. The system of claim 19, wherein the proxy server is further configured to receive the speed test data packet sent by the terminal, to forward the speed test data packet to the speed test server, to receive the speed test response packet returned by the speed test server, and to return the speed test response packet to the terminal.

21. A non-transitory computer-readable storage medium having computer-executable instructions for performing domain name resolution as claimed in any one of claims 1-5.

22. A non-transitory computer-readable storage medium having computer-executable instructions for performing domain name access as claimed in any one of claims 11-13.

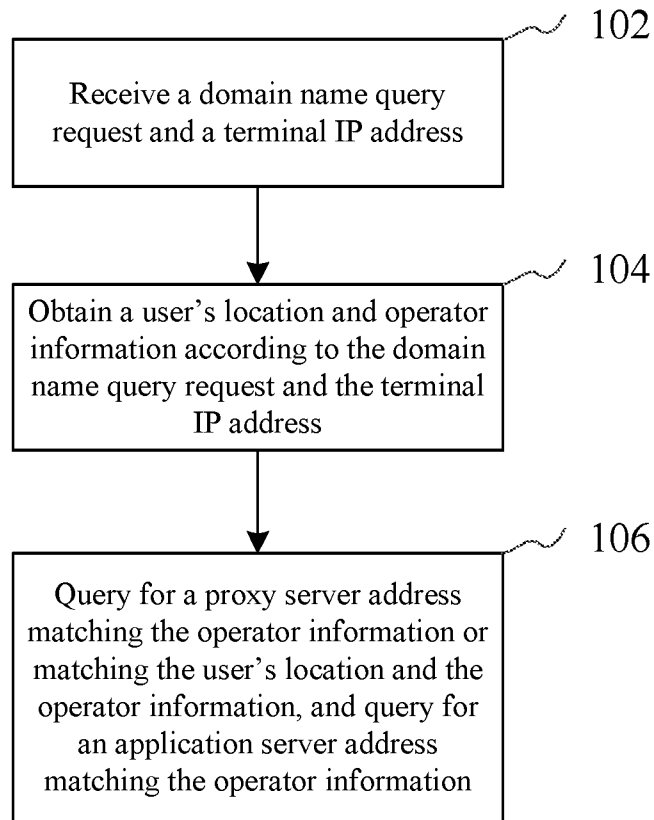


Fig. 1

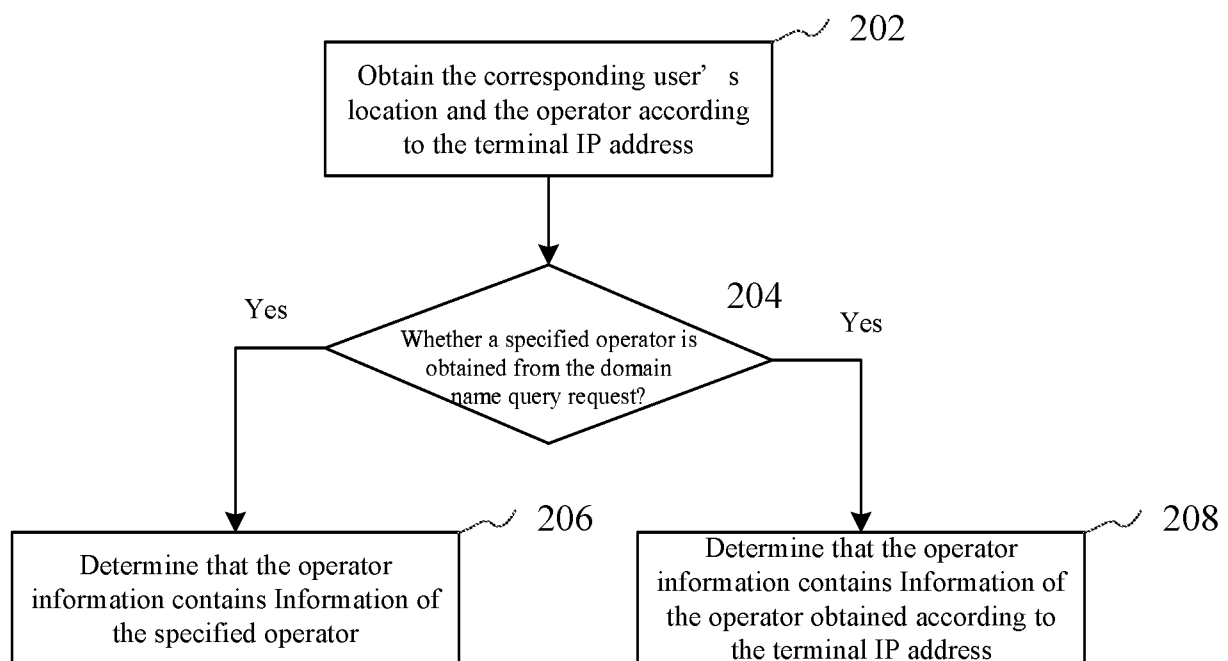


Fig. 2

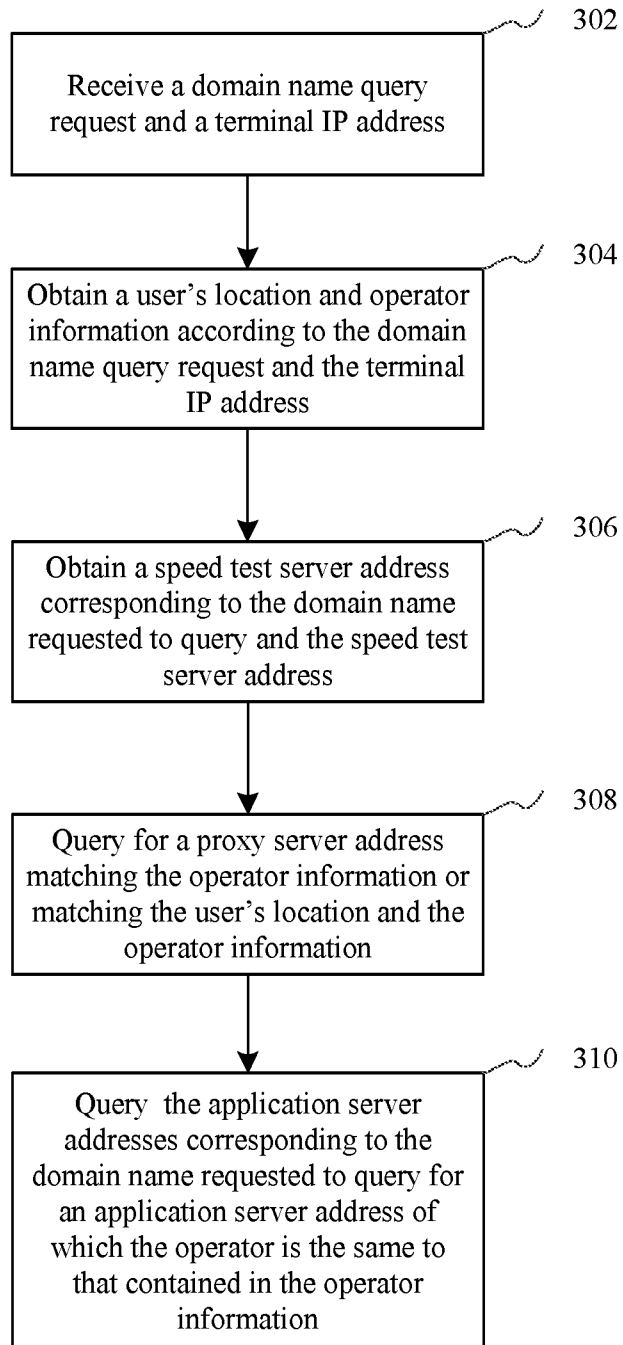


Fig. 3

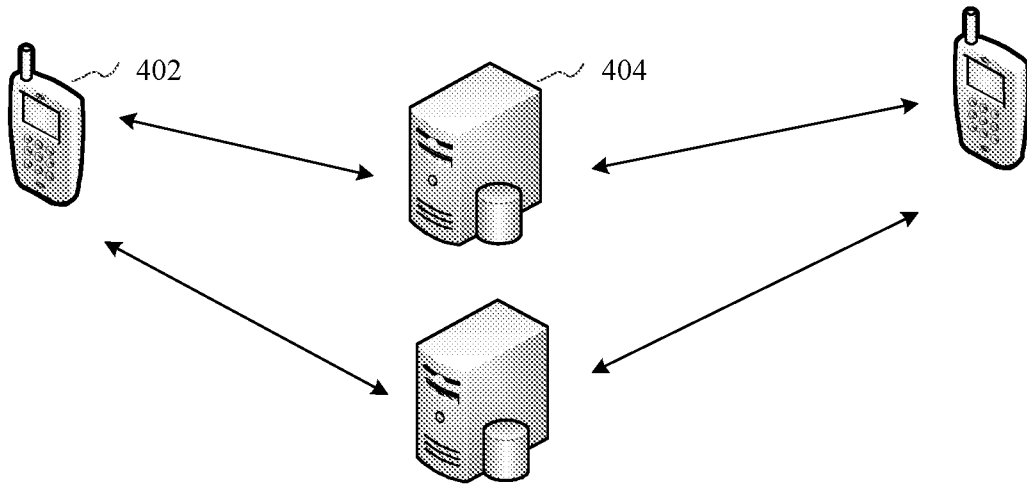


Fig. 4

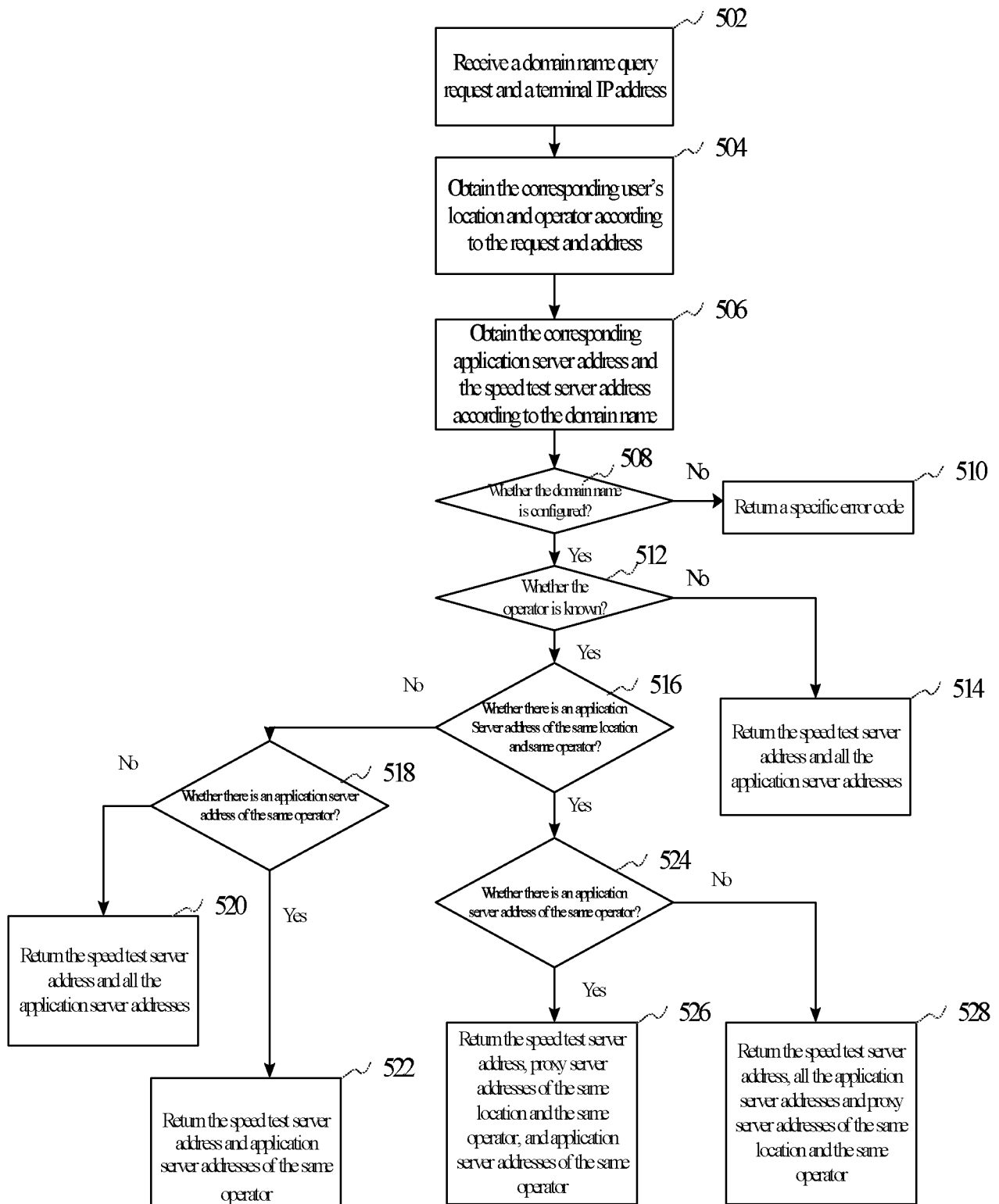


Fig. 5

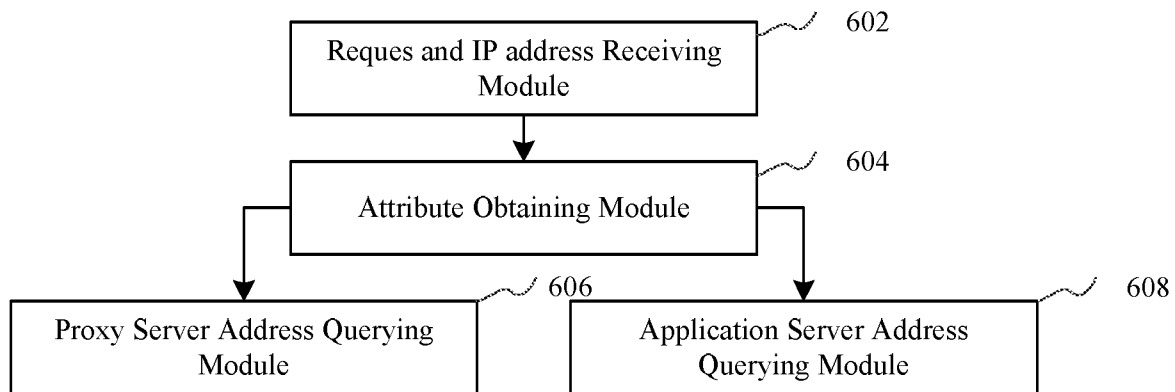


Fig. 6

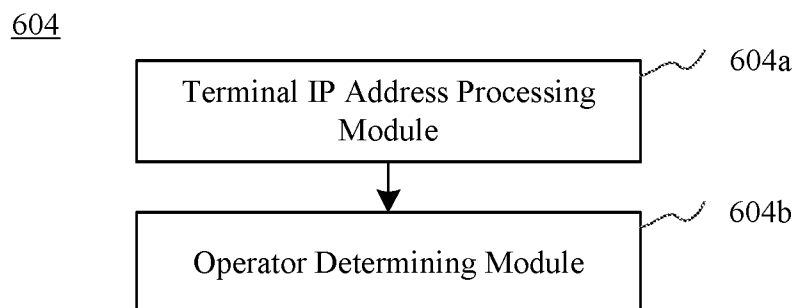


Fig. 7

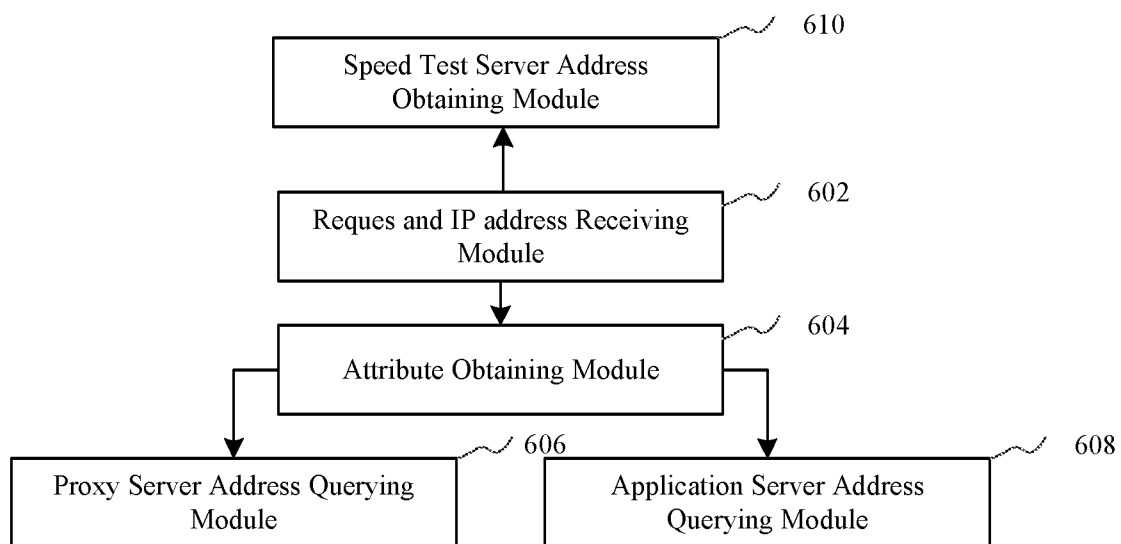


Fig. 8

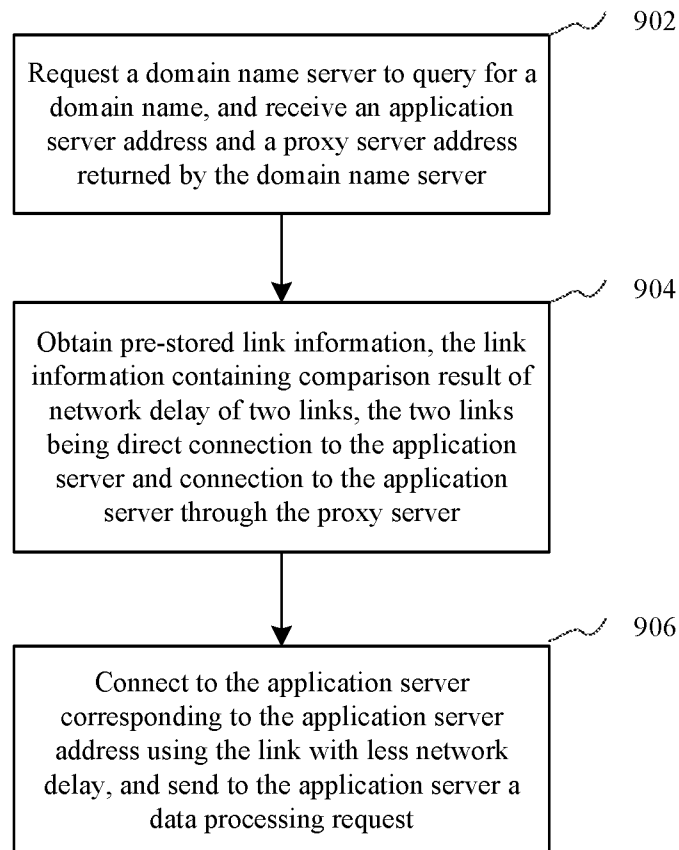


Fig. 9

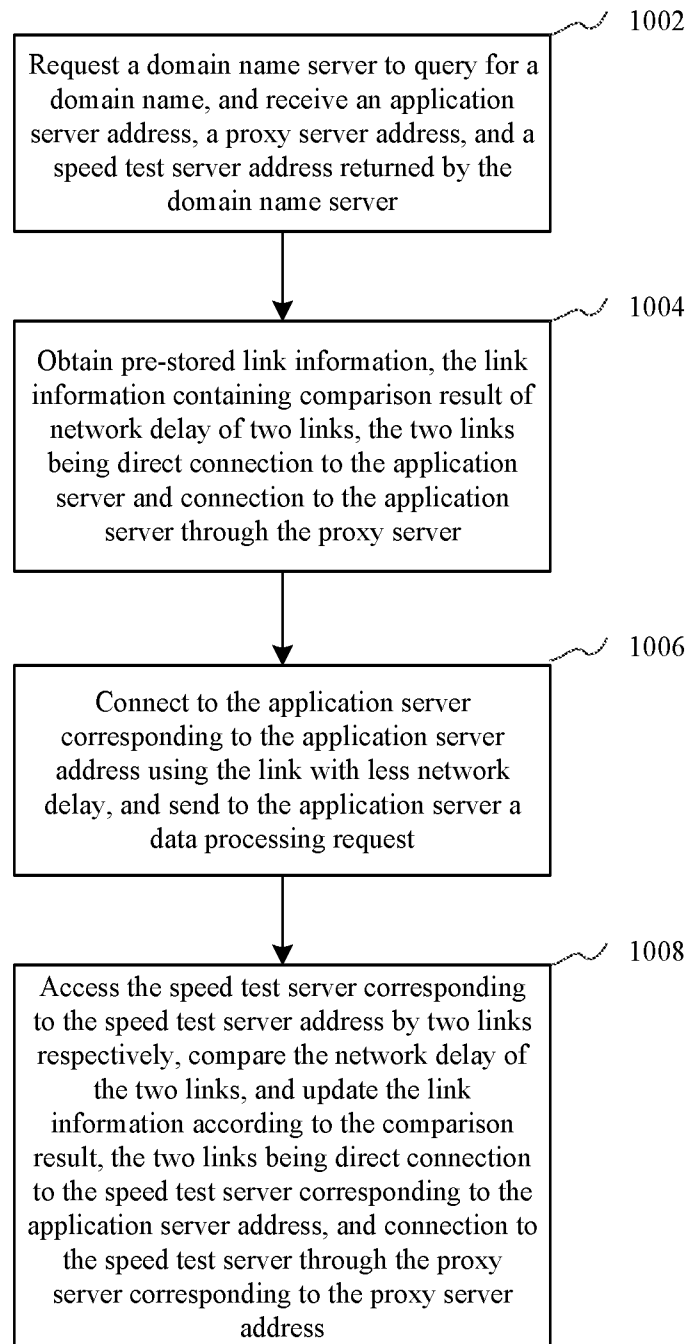


Fig. 10

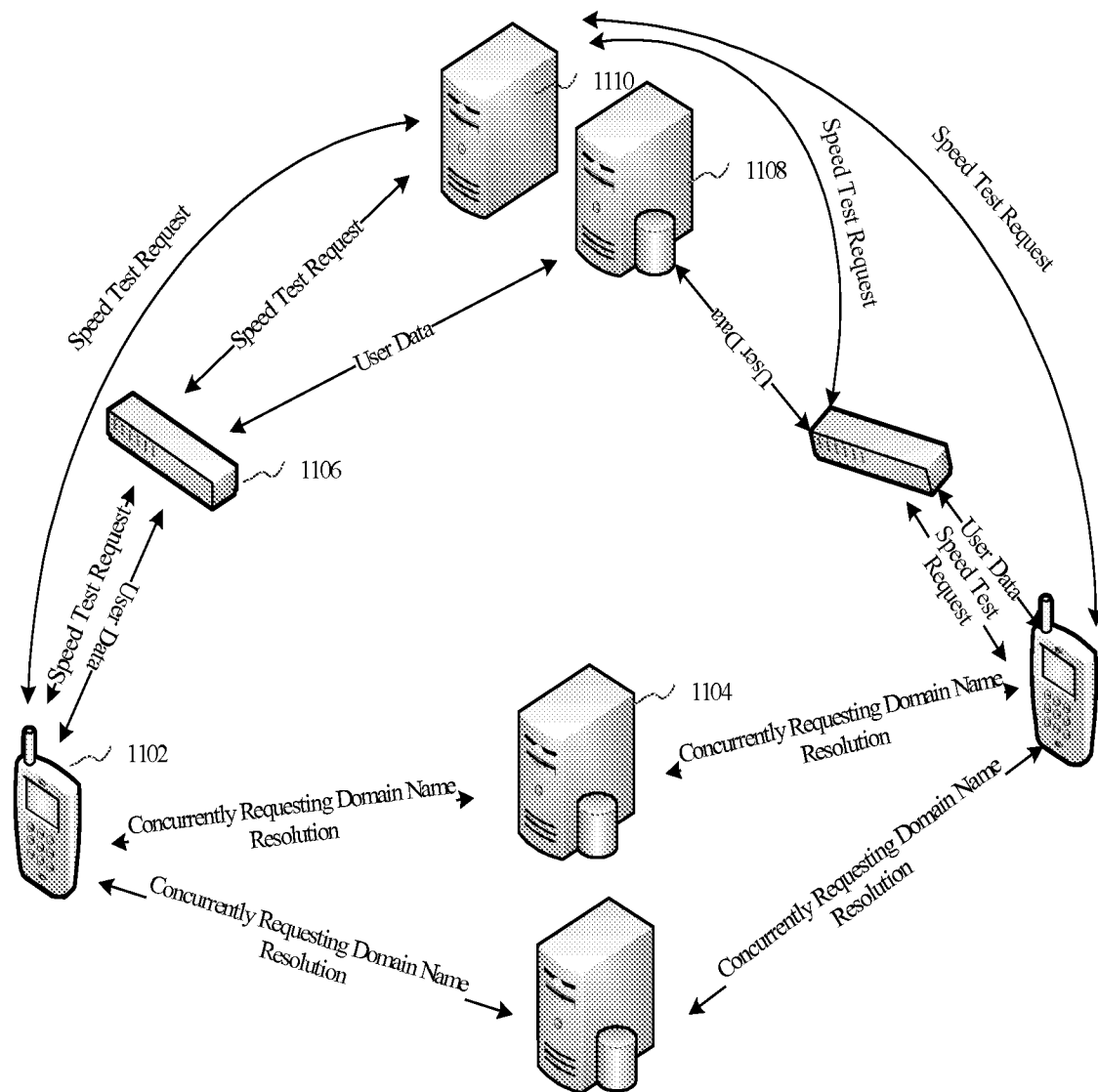


Fig. 11

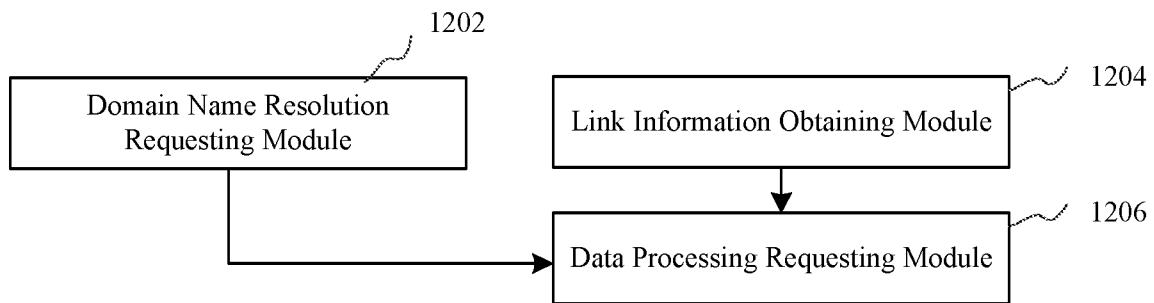


Fig. 12

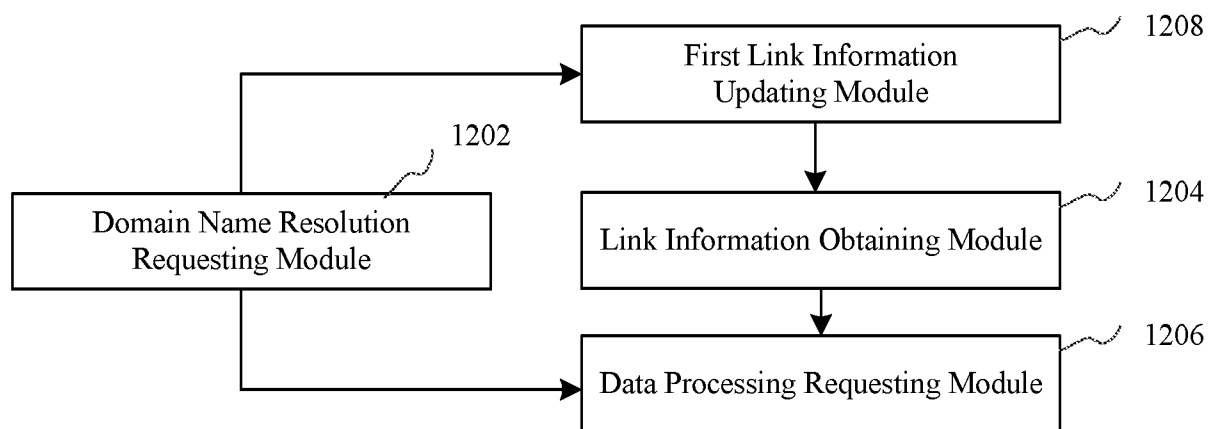


Fig. 13

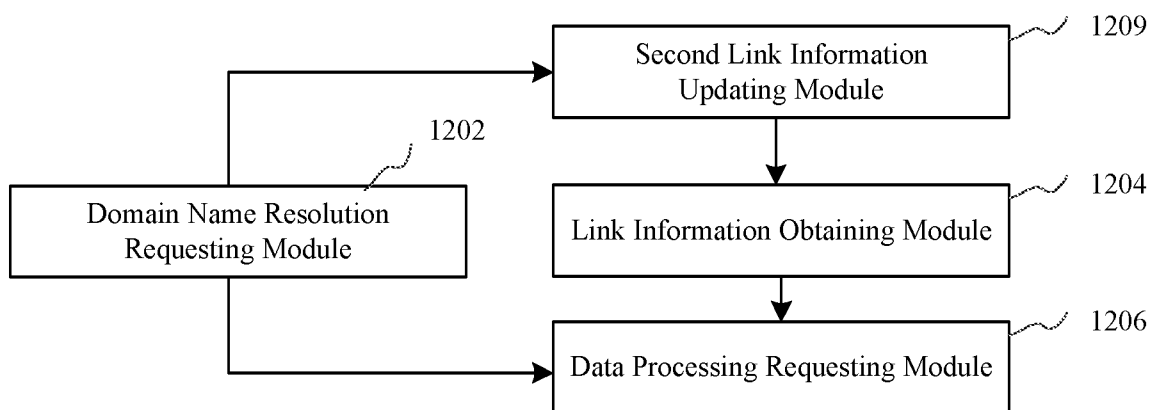


Fig. 14

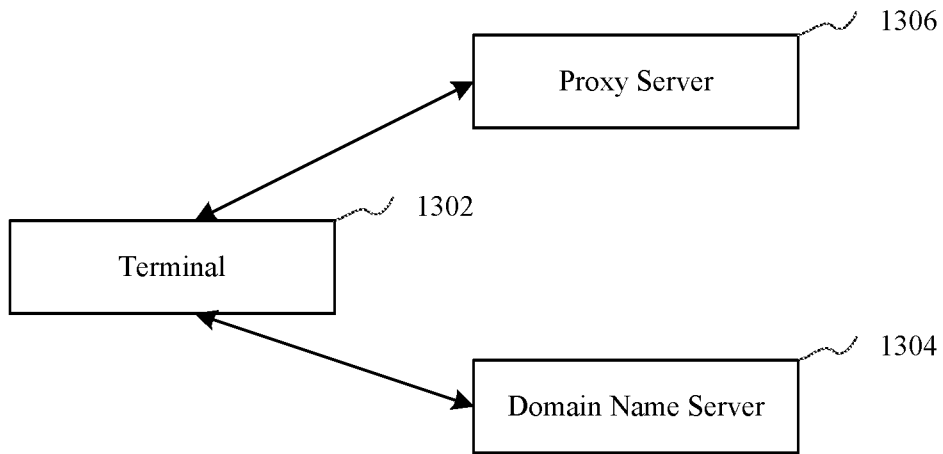


Fig. 15

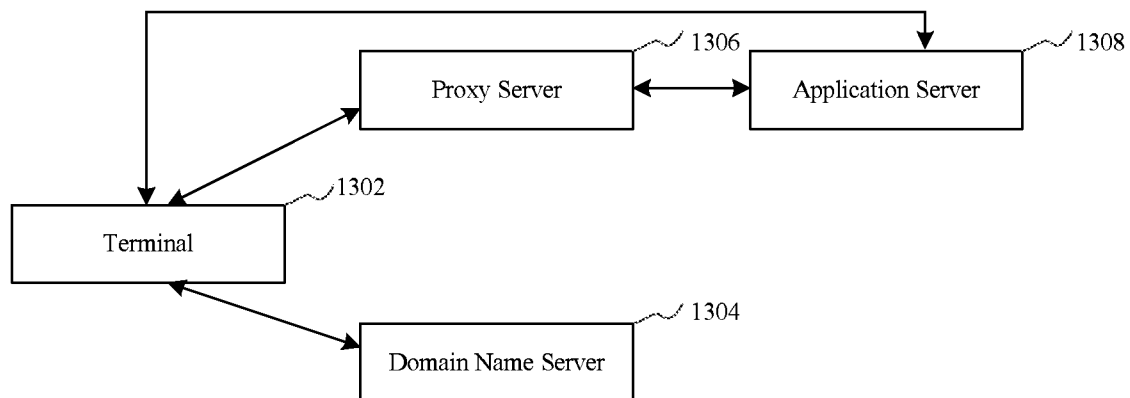


Fig. 16

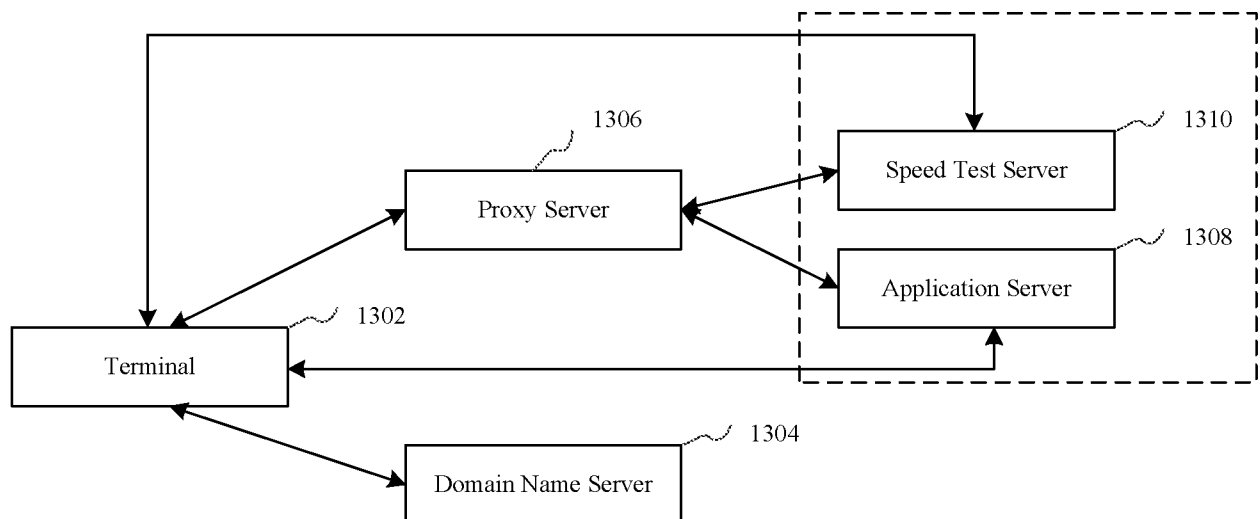


Fig. 17

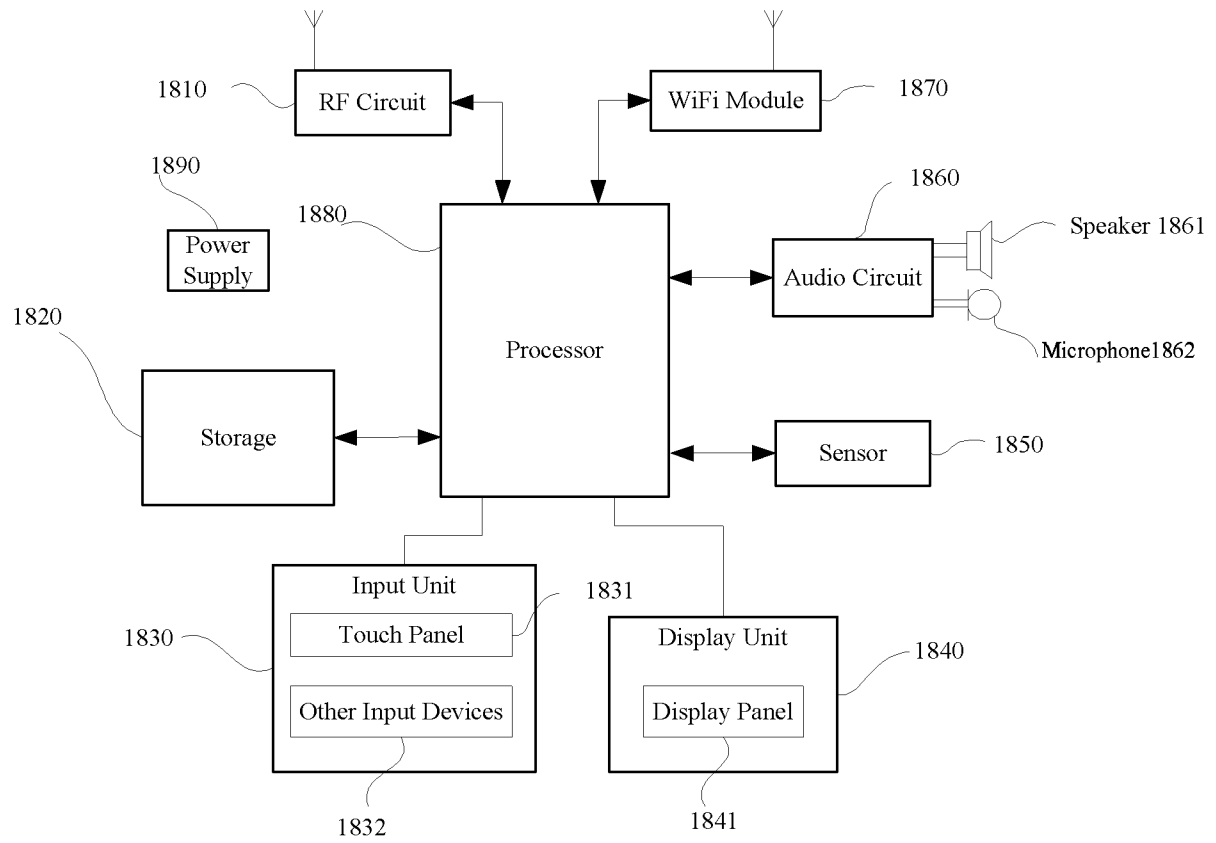


Fig. 18

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2014/080626

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/12(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L; G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, WPI, EPODOC: domain, name, resolution, URL, URI, web, address, application, content, multimedia, server, agent, proxy, link, line, route, latency, time, round, trip, fast, low, speed, location, provider, operator, ISP, DNS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 101764747 A (BEIJING 21VIANET CHINA, INC.) 30 June 2010 (2010-06-30) description, paragraphs [0061] to [0109], figures 1-7	1-10, 21
Y	CN 101764747 A (BEIJING 21VIANET CHINA, INC.) 30 June 2010 (2010-06-30) description, paragraphs [0061] to [0109], figures 1-7	17-20
X	US 6446121 B1 (CISCO TECHNOLOGY, INC.) 03 September 2002 (2002-09-03) description, column 5, line 60 to column 7, line 57, figures 4-6	11-16, 22
Y	US 6446121 B1 (CISCO TECHNOLOGY, INC.) 03 September 2002 (2002-09-03) description, column 5, line 60 to column 7, line 57, figures 4-6	17-20
A	CN 101420762 A (CHINA MOBILE COMMUNICATIONS CORPORATION) 29 April 2009 (2009-04-29) the whole document	1-22



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A”	document defining the general state of the art which is not considered to be of particular relevance	“T”	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“E”	earlier application or patent but published on or after the international filing date	“X”	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“L”	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“Y”	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“O”	document referring to an oral disclosure, use, exhibition or other means	“&”	document member of the same patent family
“P”	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

04 September 2014

Date of mailing of the international search report

24 September 2014

Name and mailing address of the ISA/

STATE INTELLECTUAL PROPERTY OFFICE OF THE
P.R.CHINA(ISA/CN)
6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088 China

Authorized officer

LU,Xia

Facsimile No. (86-10)62019451

Telephone No. (86-10)82245294

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2014/080626

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	101764747	A	30 June 2010	None			
US	6446121	B1	03 September 2002	US	6292832	B1	18 September 2001
				US	7143184	B1	28 November 2006
				US	7457877	B1	25 November 2008
CN	101420762	A	29 April 2009	EP	2205040	A1	07 July 2010
				KR	20100068457	A	23 June 2010
				US	2010303009	A1	02 December 2010
				RU	2010112648	A	10 October 2011
				JP	2010541508	A	24 December 2010
				WO	2009065335	A1	28 May 2009
				IN CHENP201001848		E	10 September 2010