



US009270844B2

(12) **United States Patent**
Takano

(10) **Patent No.:** **US 9,270,844 B2**
(45) **Date of Patent:** **Feb. 23, 2016**

(54) **IMAGE PROCESSING APPARATUS,
CONTROL METHOD, AND STORAGE
MEDIUM THAT COMPLEMENT A DOMAIN
TO AN ADDRESS DATA ITEM WITH NO
DOMAIN NAME**

USPC 709/221, 227, 230, 203; 370/225, 389
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,784,560 A * 7/1998 Kingdon G06F 21/604
709/201
8,045,485 B2 * 10/2011 Shima et al. 370/255
8,543,107 B1 * 9/2013 Bertz H04L 51/38
370/238
2002/0035624 A1 * 3/2002 Kim H04L 29/12066
709/222

(Continued)

FOREIGN PATENT DOCUMENTS

CN 1701587 A 11/2005
EP 1 763 216 A2 3/2007

(Continued)

OTHER PUBLICATIONS

Mockapetris. "RFC 883—Domain Names: Implementation Specifi-
cation." IETF Tools. Nov. 1983. Web. <[http://tools.ietf.org/html/
rfc883](http://tools.ietf.org/html/rfc883)>.*

(Continued)

Primary Examiner — Imad Hussain

(74) *Attorney, Agent, or Firm* — Rossi, Kimms & McDowell
LLP

(57) **ABSTRACT**

An image processing apparatus functions as a server. When a
reference request is received from a client or a user's instruc-
tion for transmitting such address data to a client is received,
and the address data to be transmitted has no domain name
described therein, the apparatus complements the address
data with a domain name, and then sends the complemented
address data to the client. This makes it possible to export
address data correctly to a designated destination, and to
reduce failure of authentication.

10 Claims, 19 Drawing Sheets

(75) Inventor: **Shinichi Takano**, Kawasaki (JP)

(73) Assignee: **CANON KABUSHIKI KAISHA**,
Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 974 days.

(21) Appl. No.: **12/717,747**

(22) Filed: **Mar. 4, 2010**

(65) **Prior Publication Data**

US 2010/0228820 A1 Sep. 9, 2010

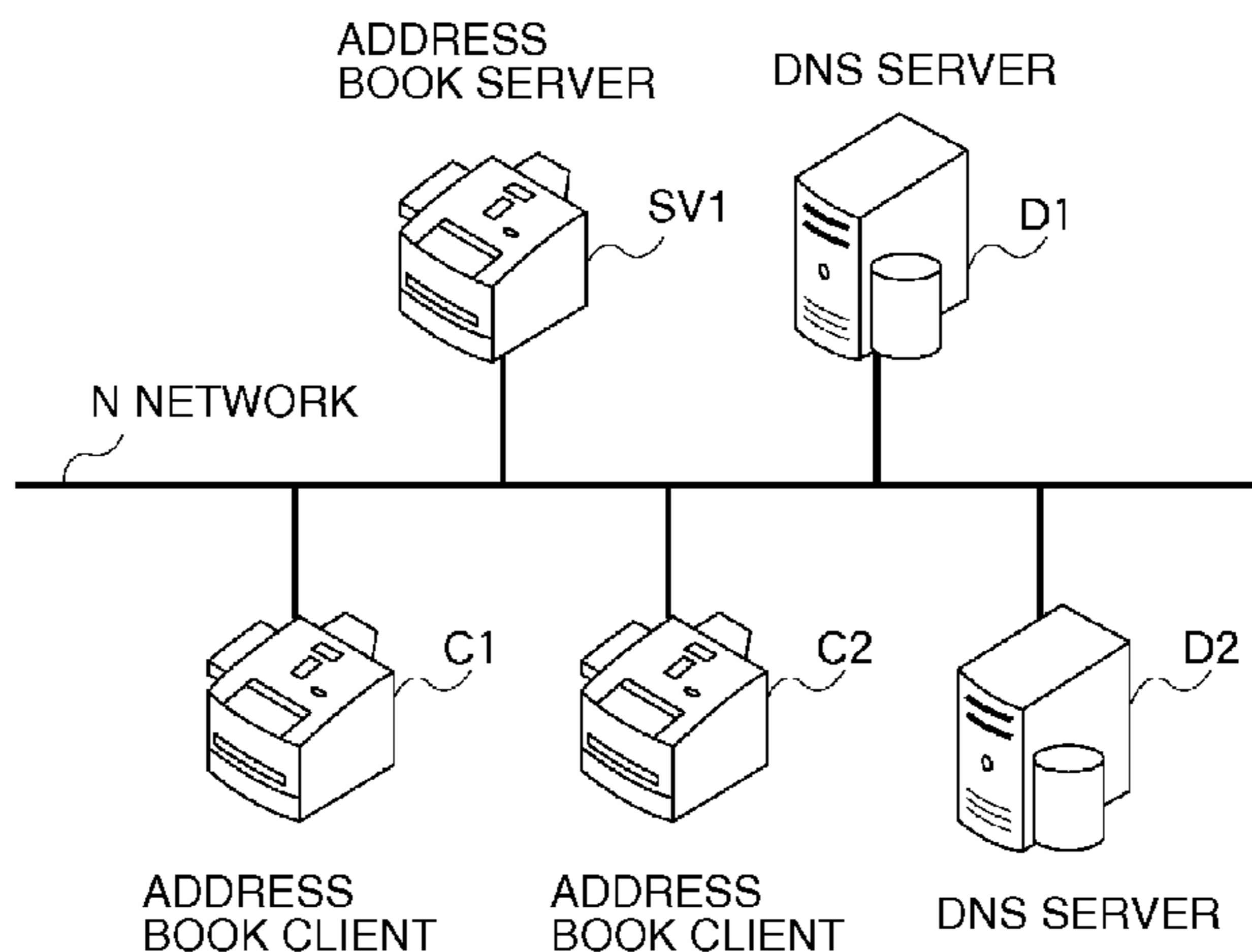
(30) **Foreign Application Priority Data**

Mar. 4, 2009 (JP) 2009-050755
Mar. 1, 2010 (JP) 2010-044499

(51) **Int. Cl.**
G06F 15/16 (2006.01)
H04N 1/00 (2006.01)
H04L 29/12 (2006.01)

(52) **U.S. Cl.**
CPC **H04N 1/00204** (2013.01); **H04L 61/15**
(2013.01); **H04L 61/1594** (2013.01); **H04L**
61/303 (2013.01); **H04N 1/00244** (2013.01);
H04N 1/00347 (2013.01); **H04L 29/12066**
(2013.01); **H04L 61/1511** (2013.01); **H04N**
2201/0031 (2013.01); **H04N 2201/0039**
(2013.01); **H04N 2201/3208** (2013.01); **H04N**
2201/3278 (2013.01)

(58) **Field of Classification Search**
CPC H04L 61/15



(56)

References Cited

U.S. PATENT DOCUMENTS

2002/0087743 A1* 7/2002 Givoly G06F 17/30286
719/330
2002/0191211 A1* 12/2002 Miller G06F 3/1204
358/1.15
2003/0055979 A1* 3/2003 Cooley 709/227
2003/0126241 A1* 7/2003 Nagura et al. 709/221
2003/0145089 A1* 7/2003 Edwards H04L 29/06
709/228
2003/0163737 A1* 8/2003 Roskind G06F 21/31
726/6
2004/0047349 A1* 3/2004 Fujita et al. 370/389
2004/0153455 A1 8/2004 Bhogal et al.
2004/0220970 A1* 11/2004 Wilson G06F 3/1204
2004/0263898 A1* 12/2004 Ferlitsch G06F 3/1207
358/1.15
2004/0264389 A1* 12/2004 Abdo H04L 29/12113
370/255
2005/0086377 A1* 4/2005 Aso H04L 29/12066
709/245
2007/0081399 A1 4/2007 Kishimoto
2008/0301313 A1* 12/2008 Bhogal et al. 709/230
2009/0182821 A1* 7/2009 Allen H04L 63/10
709/206
2010/0005146 A1* 1/2010 Drako G06Q 10/00
709/206
2010/0005191 A1* 1/2010 Drako G06F 15/16
709/238

FOREIGN PATENT DOCUMENTS

JP 2004-165877 A 6/2004
JP 2005-094676 A 4/2005
KR 10-2005-0103192 A 10/2005

OTHER PUBLICATIONS

Extended European Search Report issued in European Patent Application No. 10155389.9 dated Jun. 29, 2010.
Mockapetris P. ,ISI, "Domain Names-Implementation and Specification" Network Working Group Request for Comments # 883, IETF Standard, Internet Engineering Task Force, IETF, Nov. 1, 1983, XP015006873, pp. 1-76. Cited in EESR issued in corresponding EP 10155389.9 dated Jun. 29, 2010; see NPL cite No. 1.
QAS: "Address database correction software," Jul. 31, 2008, XP002586763, retrieved from the Internet on Jun. 11, 2010: <http://web.archive.org/web/20080731051156/http://www.qas.com/address-correction-software.htm>; Sections "Address database correction software" and "QAS Pro Web address validation software" included, pp. 1-3. Cited in EESR issued in corresponding EP 10155389.9 dated Jun. 29, 2010; see NPL cite No. 1.
Chinese Office Action issued in counterpart Chinese application No. CN201010126858.4, dated Aug. 16, 2012. English translation provided.
Korean Office Action. Mail date Sep. 26, 2012.

* cited by examiner

FIG.1

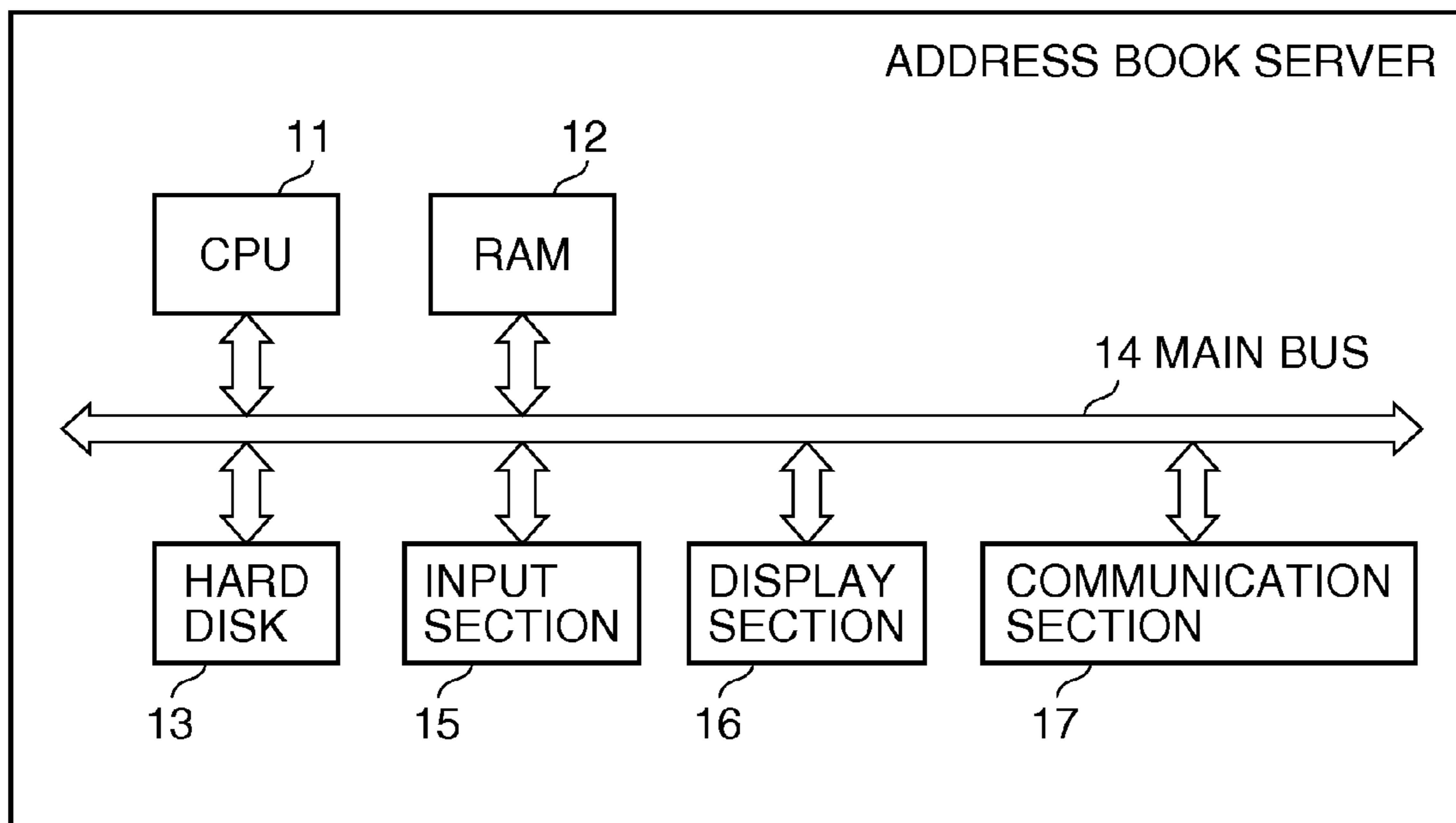


FIG.2

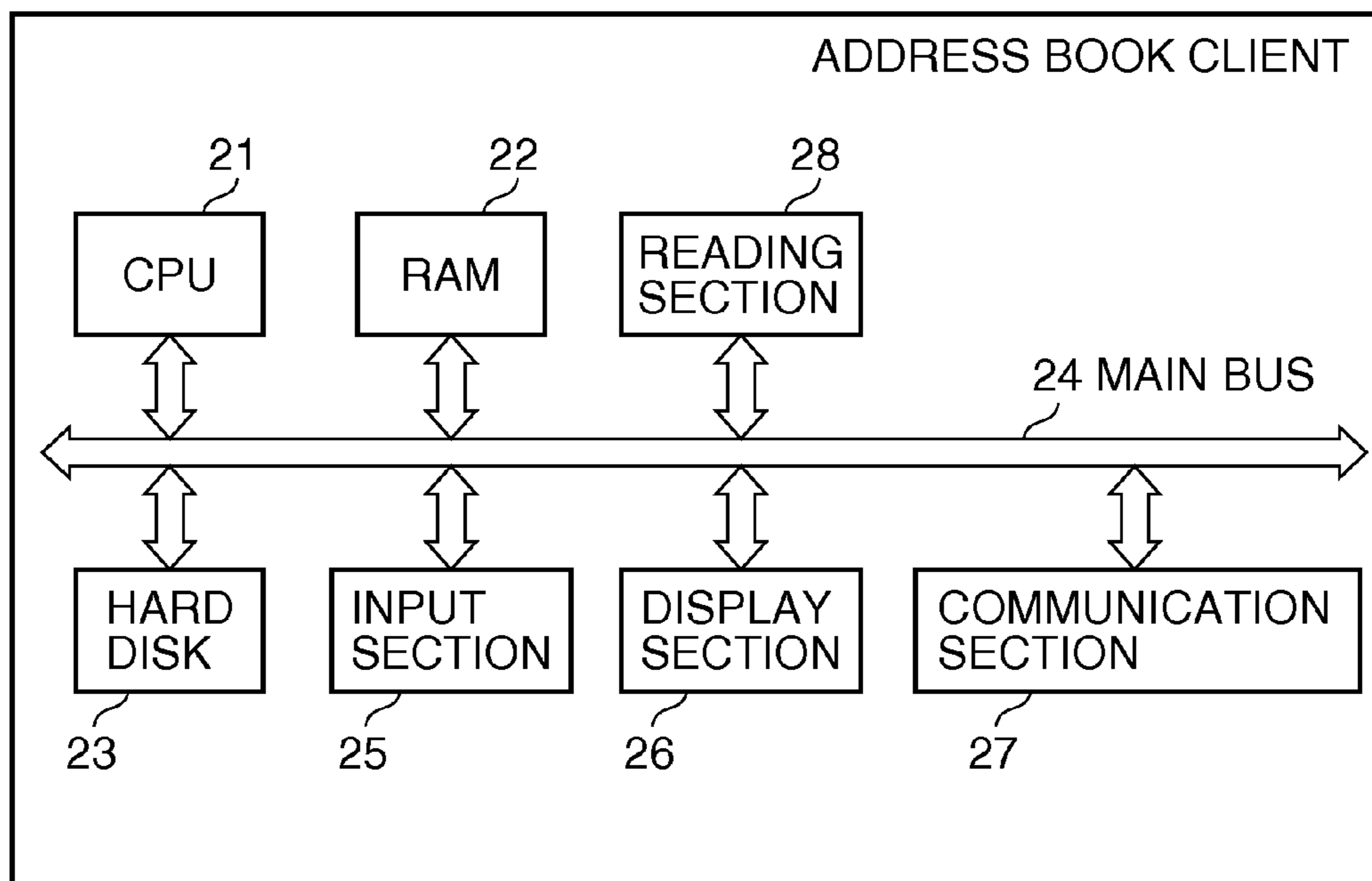


FIG.3

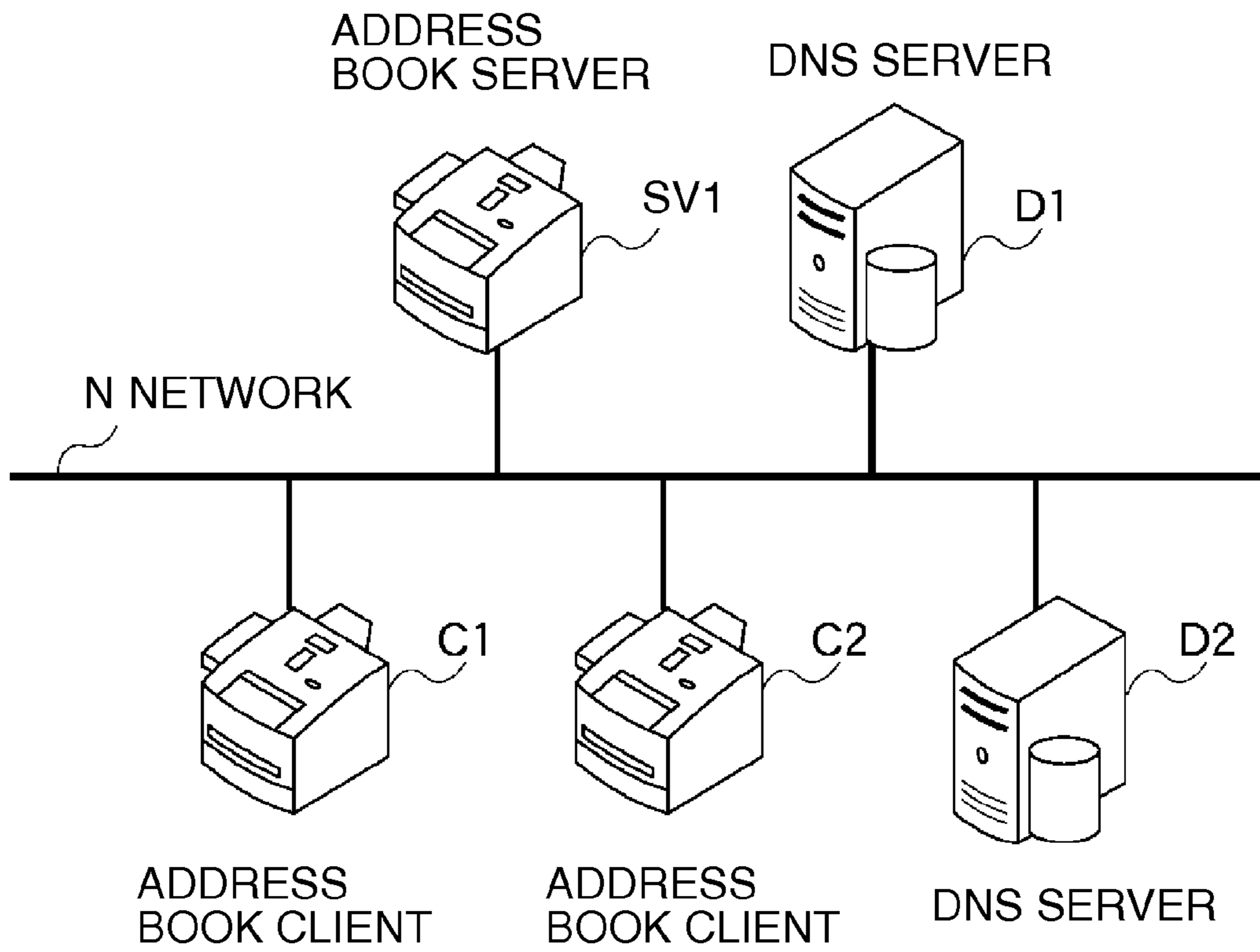


FIG.4

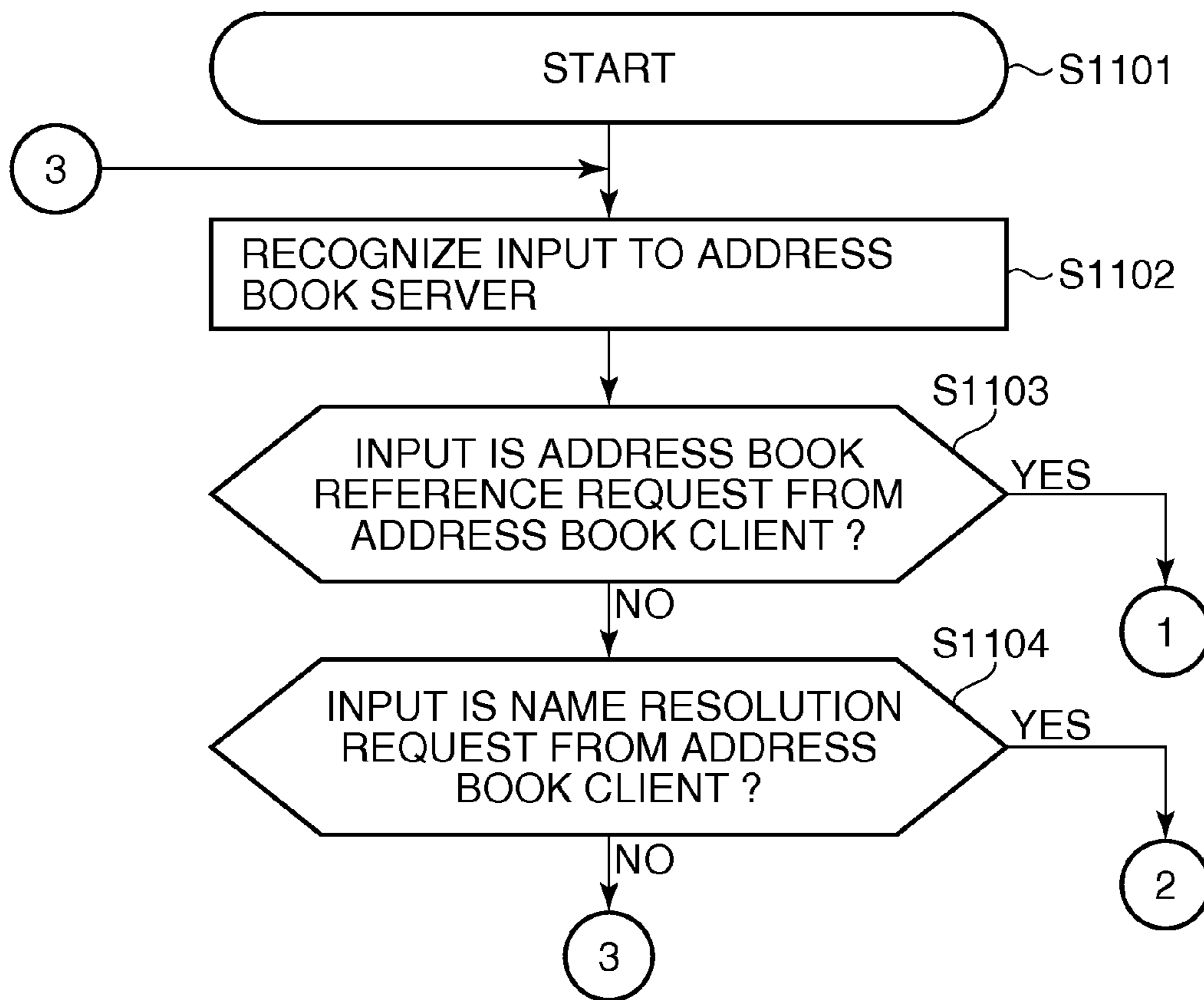


FIG. 5

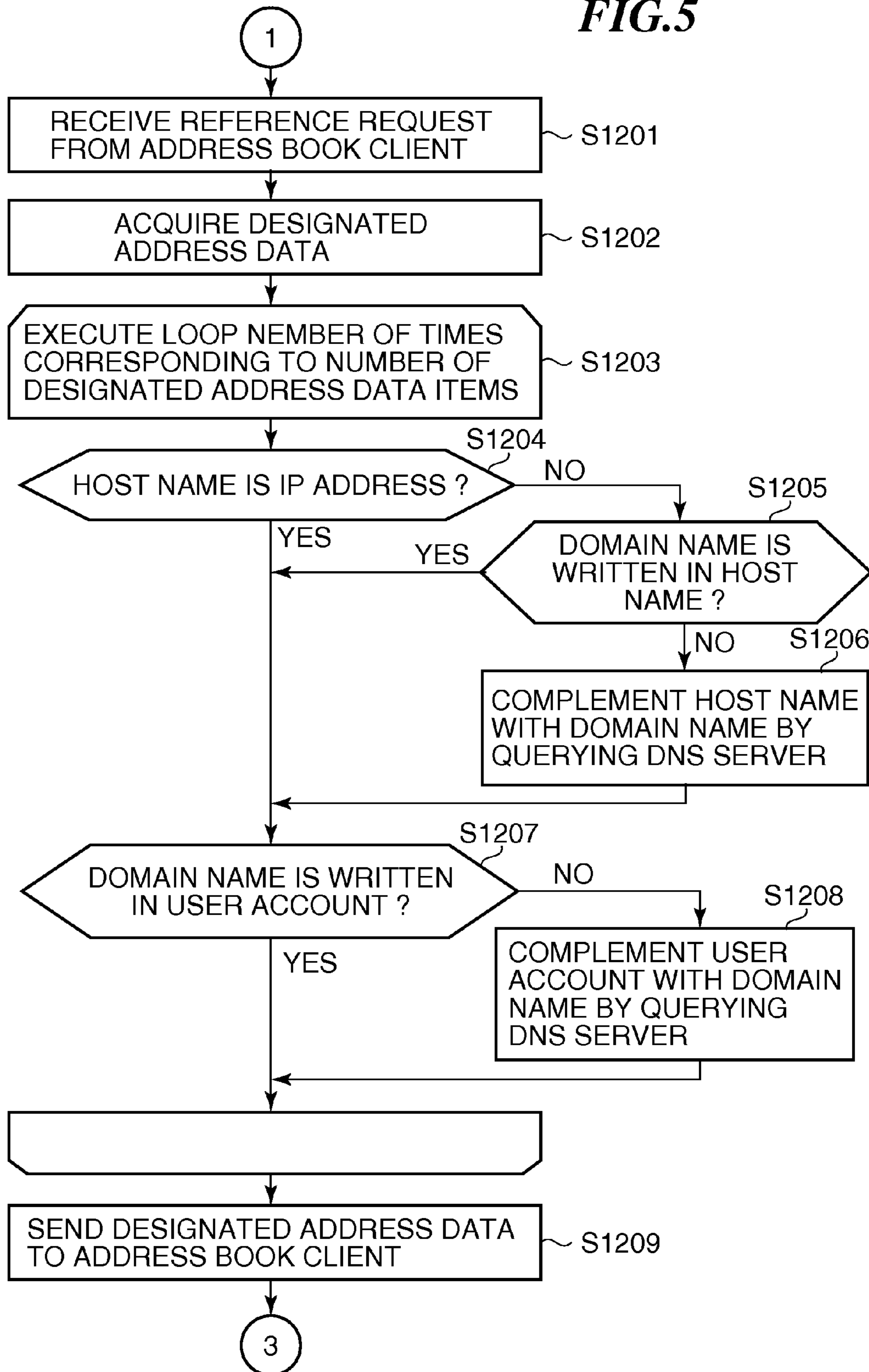


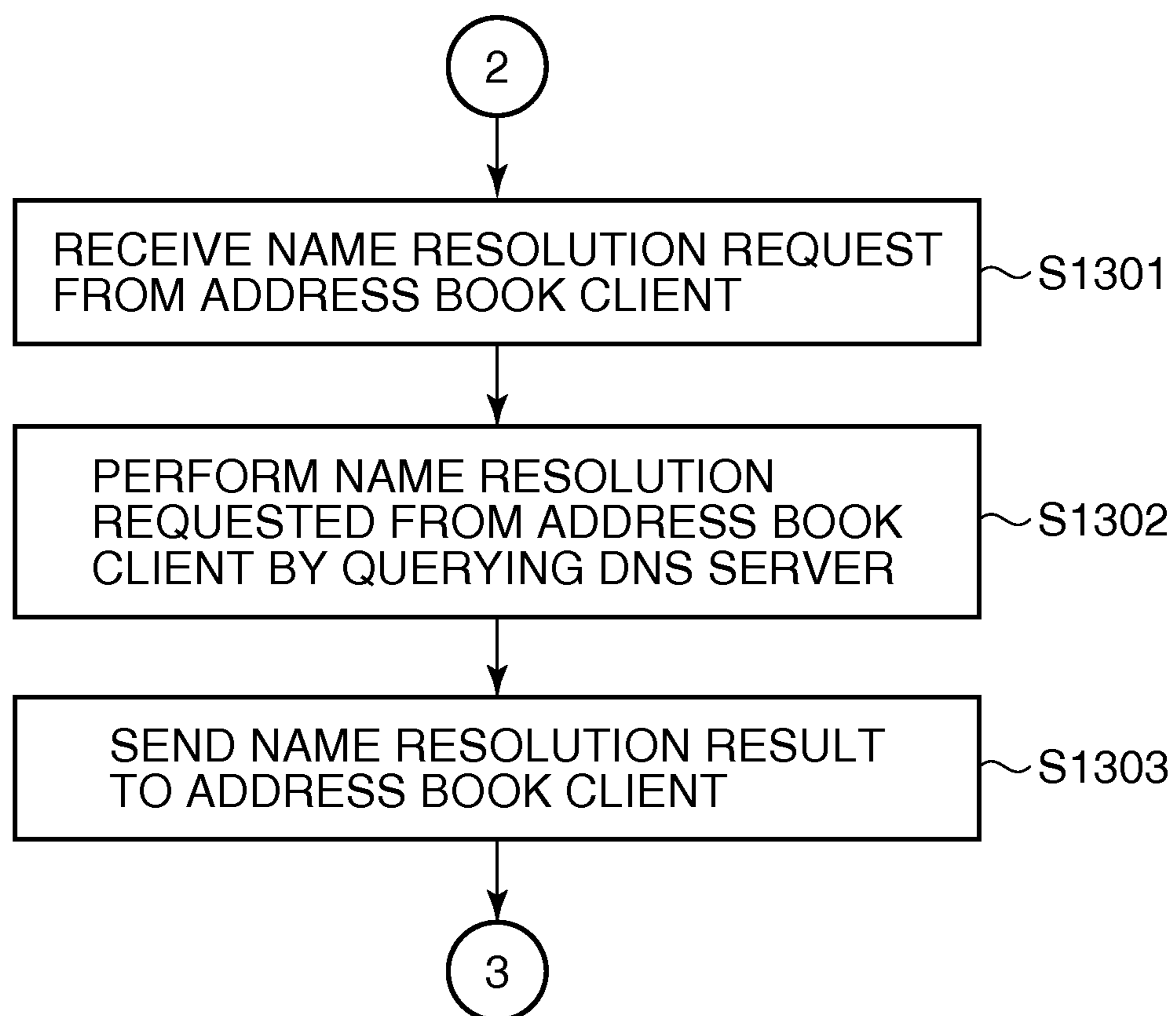
FIG. 6

FIG. 7

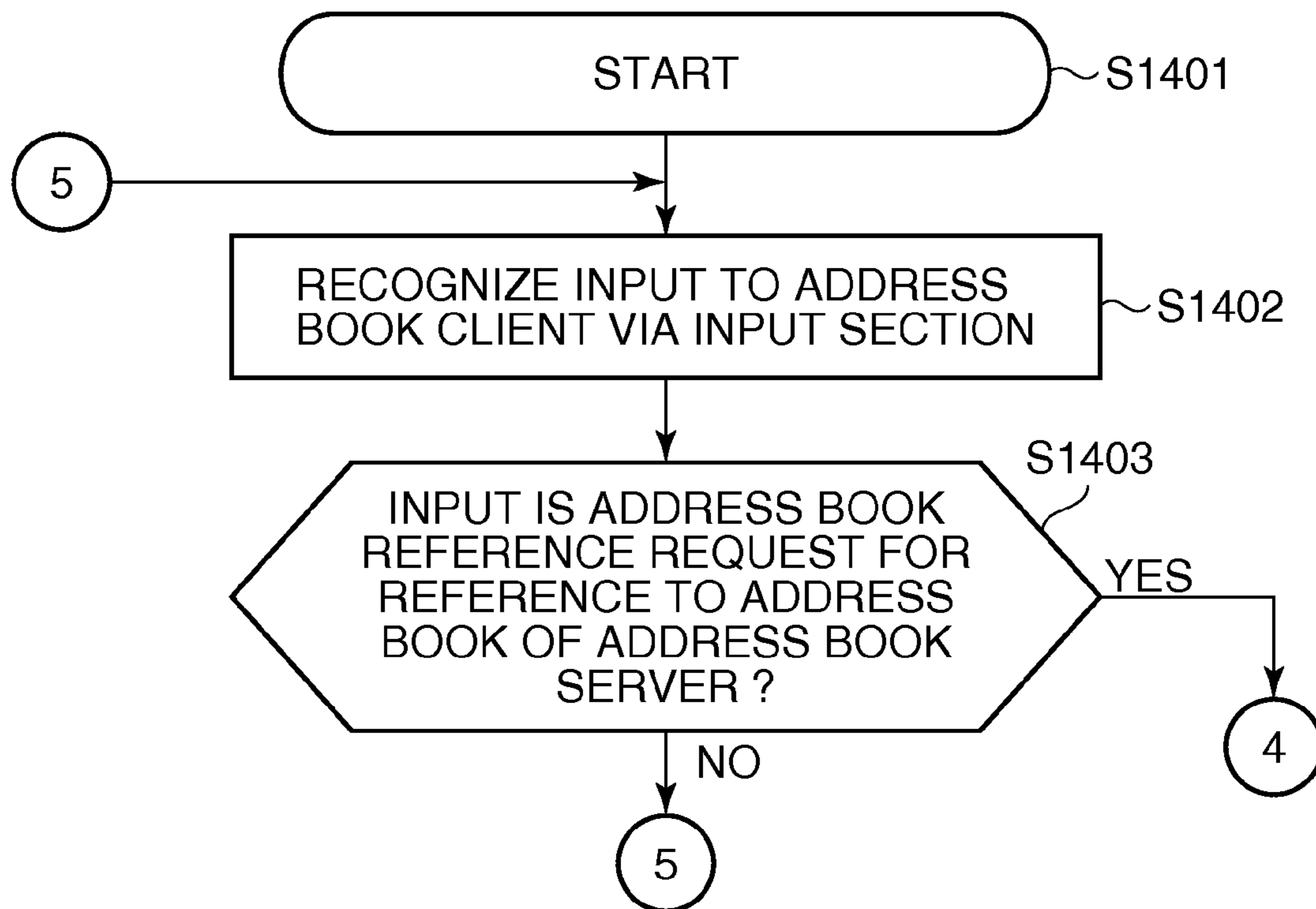


FIG.8

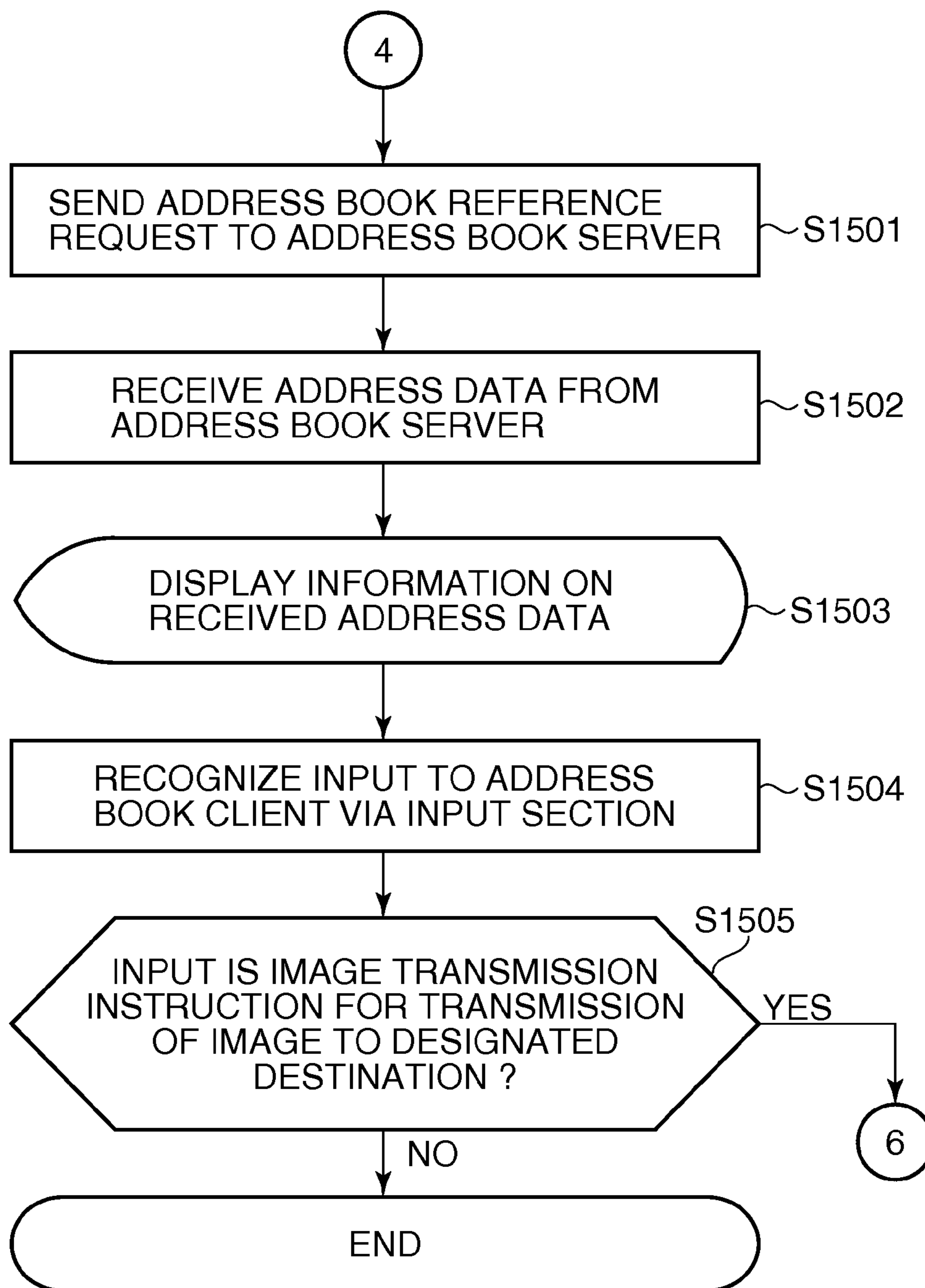


FIG. 9

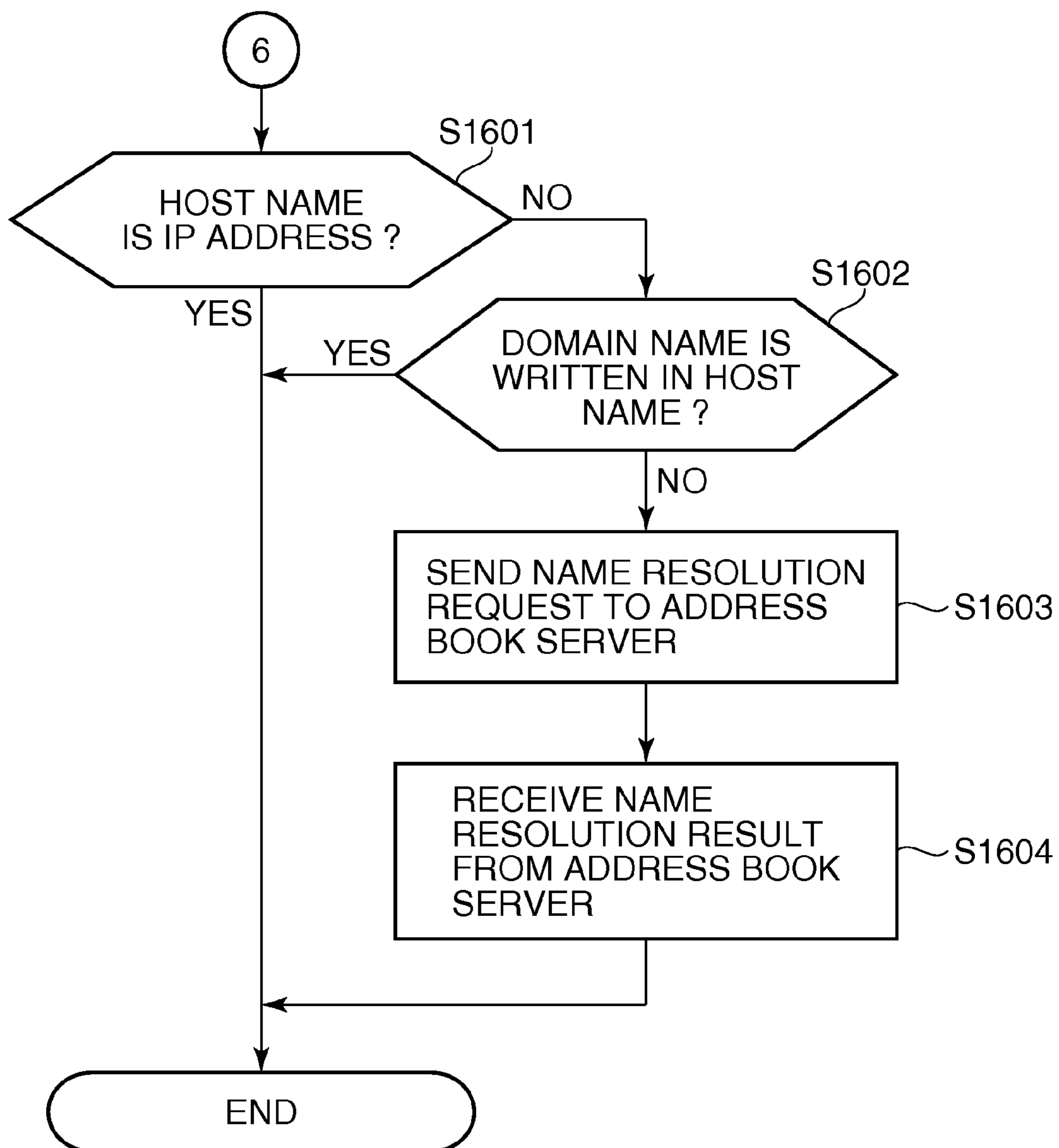


FIG. 10

ADDRESS BOOK

NAME	FILE SERVER 1
HOST NAME	<u>fileserver1.aaa.bbb.co.jp</u>
USER NAME	<u>Tanaka.aaa.bbb.co.jp</u>
PASSWORD	* * * * *

CLOSE

HOST NAME AND USER NAME ARE COMPLEMENTED WITH DOMAIN NAMES

Detailed description of FIG. 10: The diagram shows a rectangular window titled 'ADDRESS BOOK'. Inside, there are four input fields stacked vertically. The first field is labeled 'NAME' and contains the text 'FILE SERVER 1'. The second field is labeled 'HOST NAME' and contains the text 'fileserver1.aaa.bbb.co.jp', with a horizontal line underlining the entire text. The third field is labeled 'USER NAME' and contains the text 'Tanaka.aaa.bbb.co.jp', with a horizontal line underlining the entire text. The fourth field is labeled 'PASSWORD' and contains ten asterisks '* * * * *'. To the right of these fields is a rectangular button labeled 'CLOSE'. Above the 'HOST NAME' and 'USER NAME' fields, there is a vertical text label: 'HOST NAME AND USER NAME ARE COMPLEMENTED WITH DOMAIN NAMES'. Two arrows originate from this label: one points to the underlined text in the 'HOST NAME' field, and the other points to the underlined text in the 'USER NAME' field.

FIG. 11

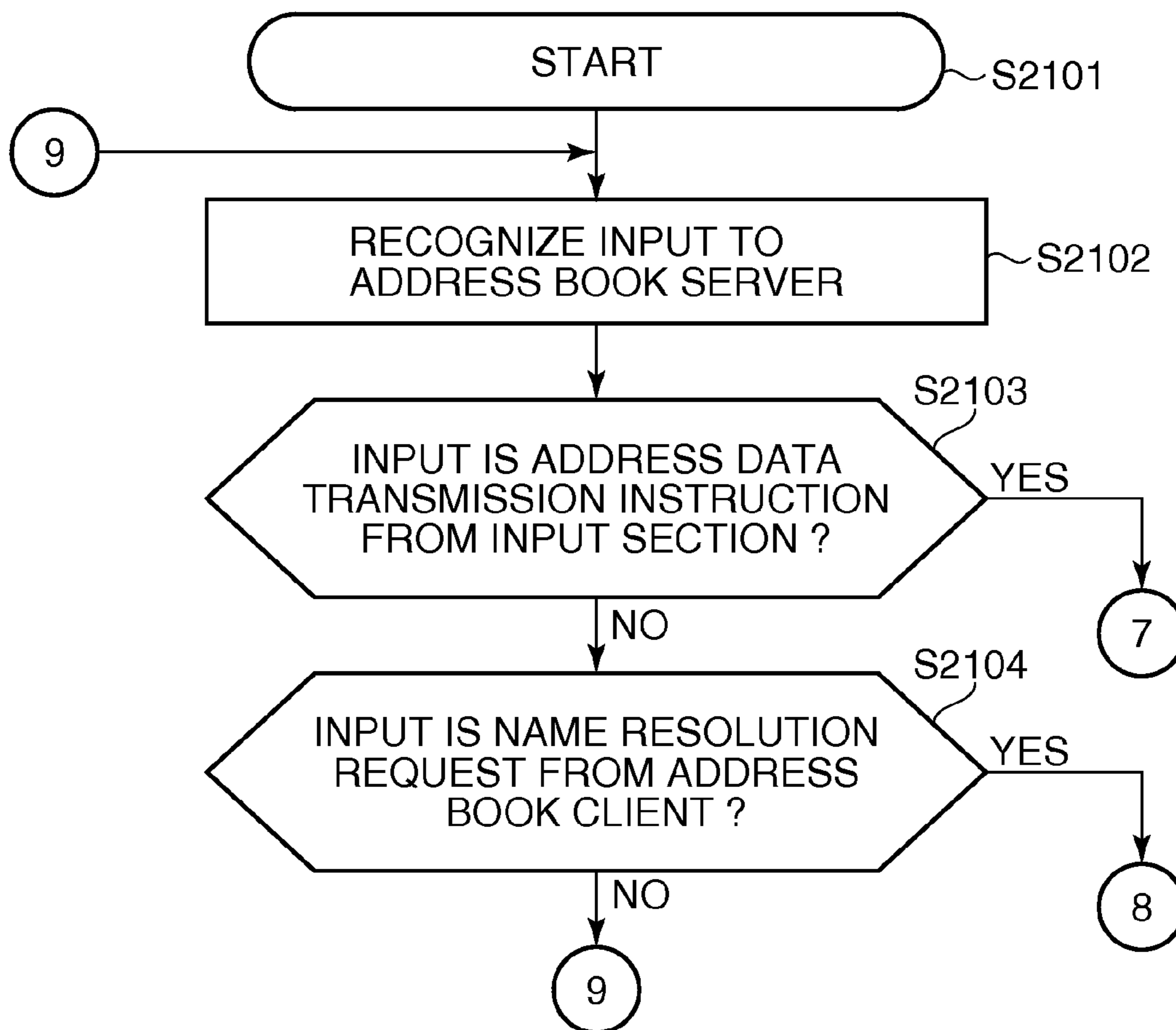


FIG.12

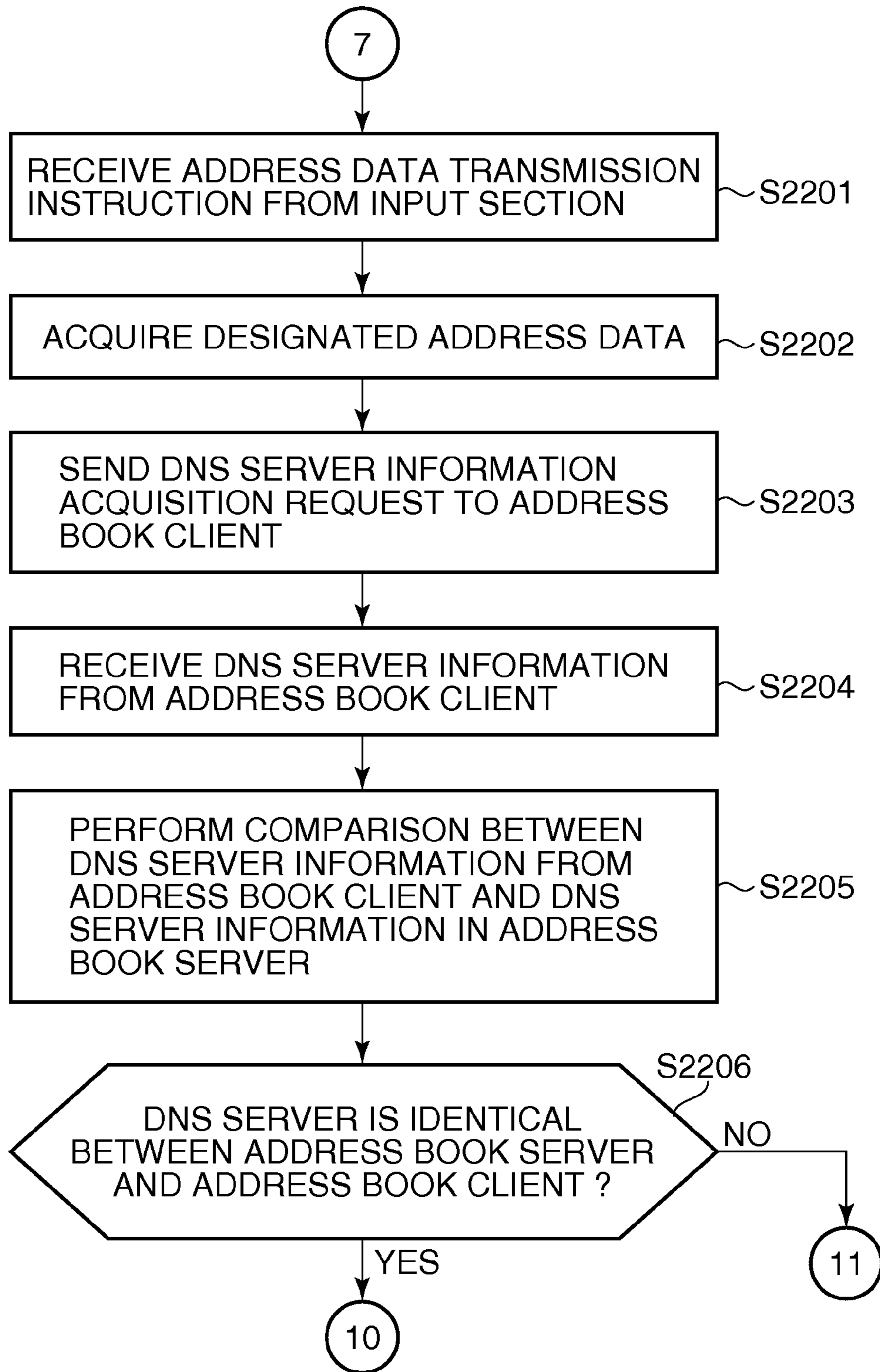


FIG.13

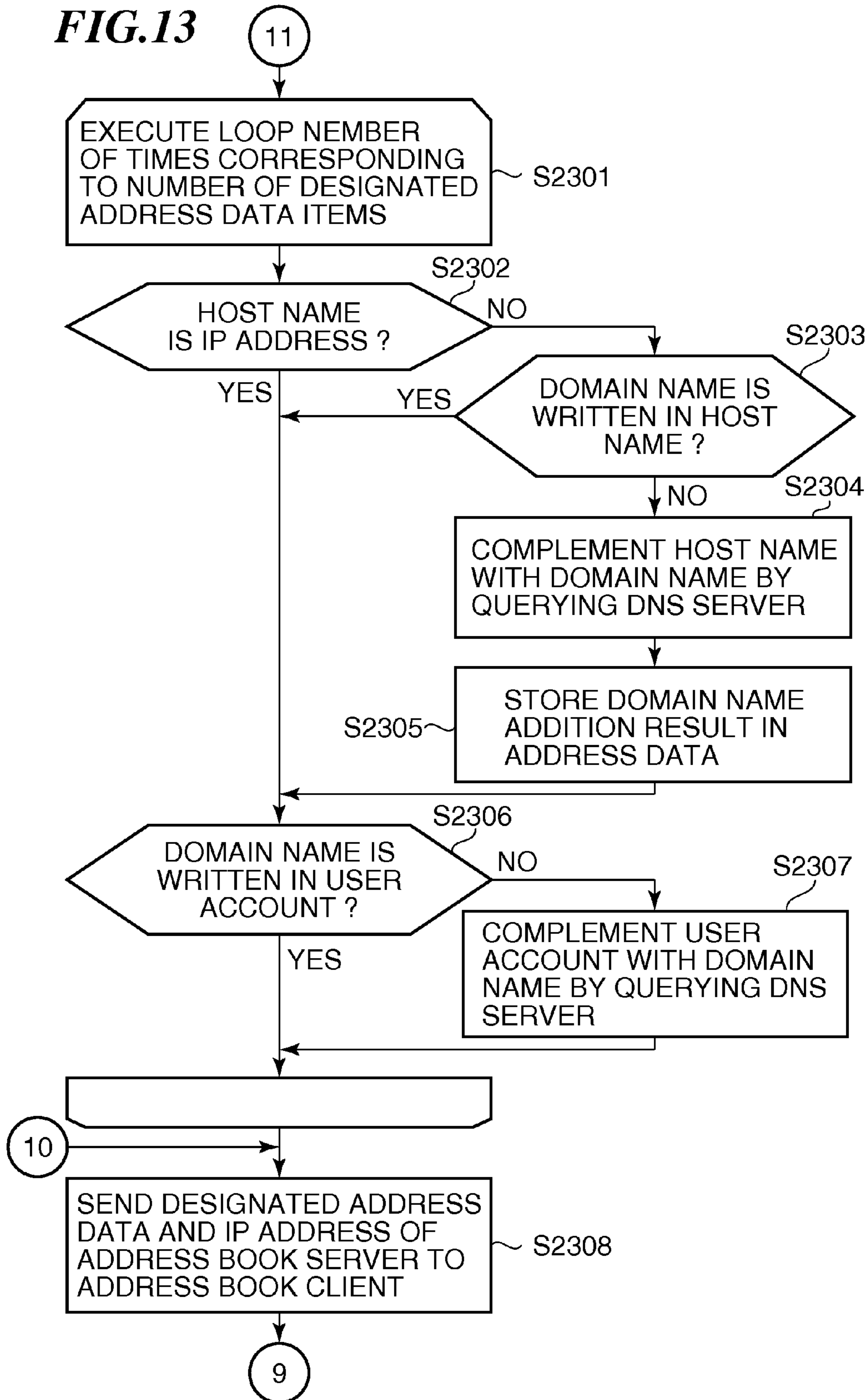


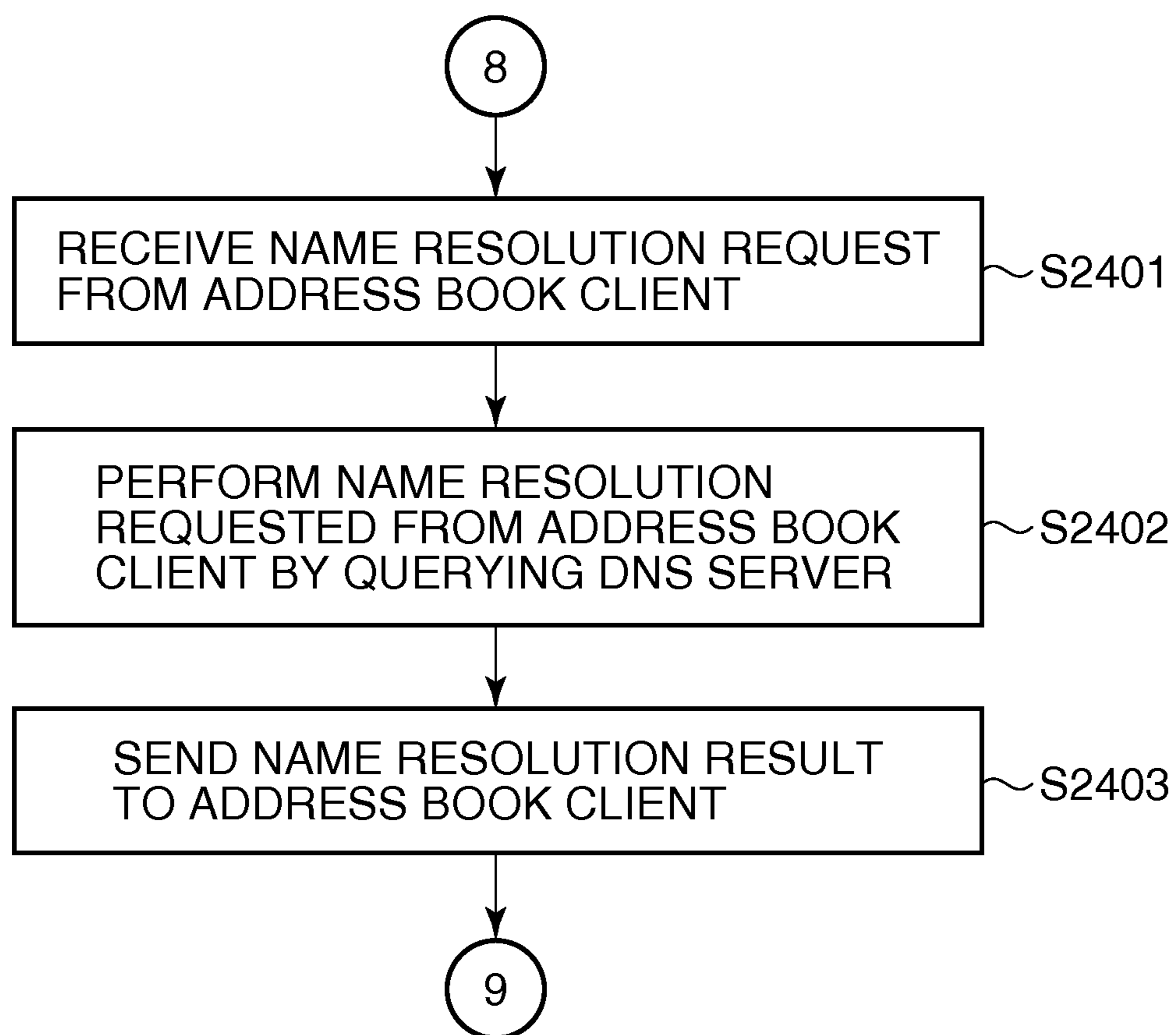
FIG. 14

FIG.15

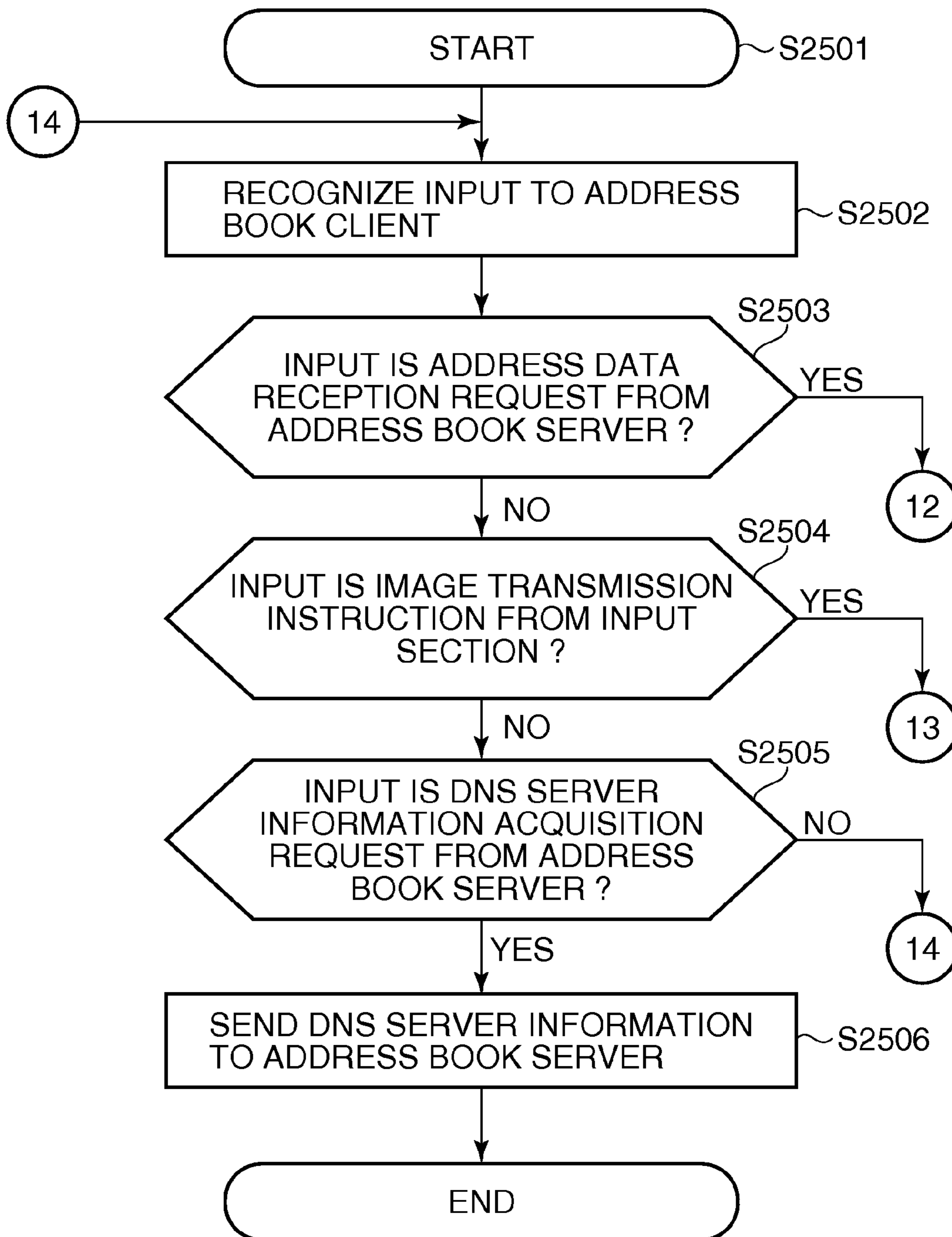


FIG.16

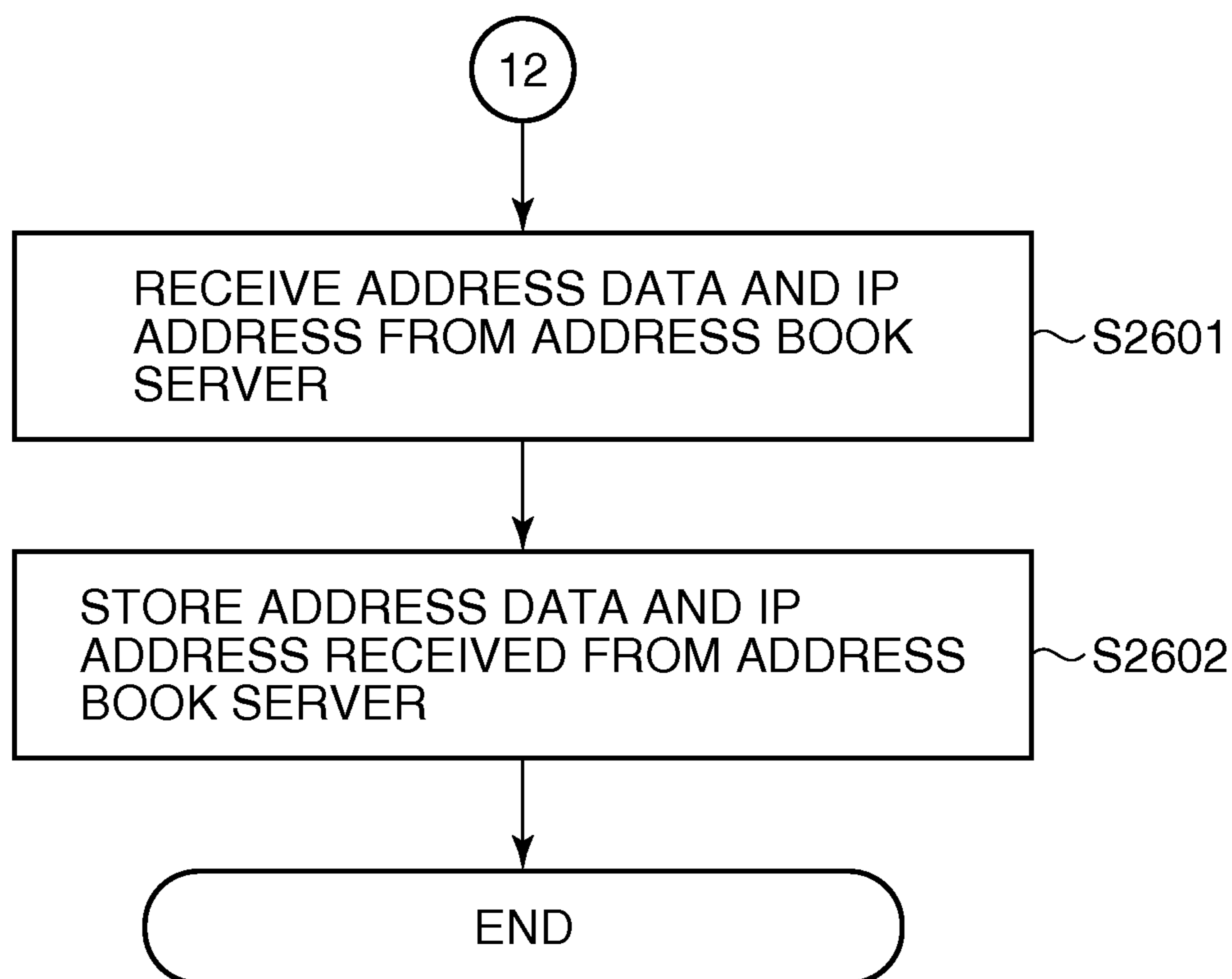


FIG.17

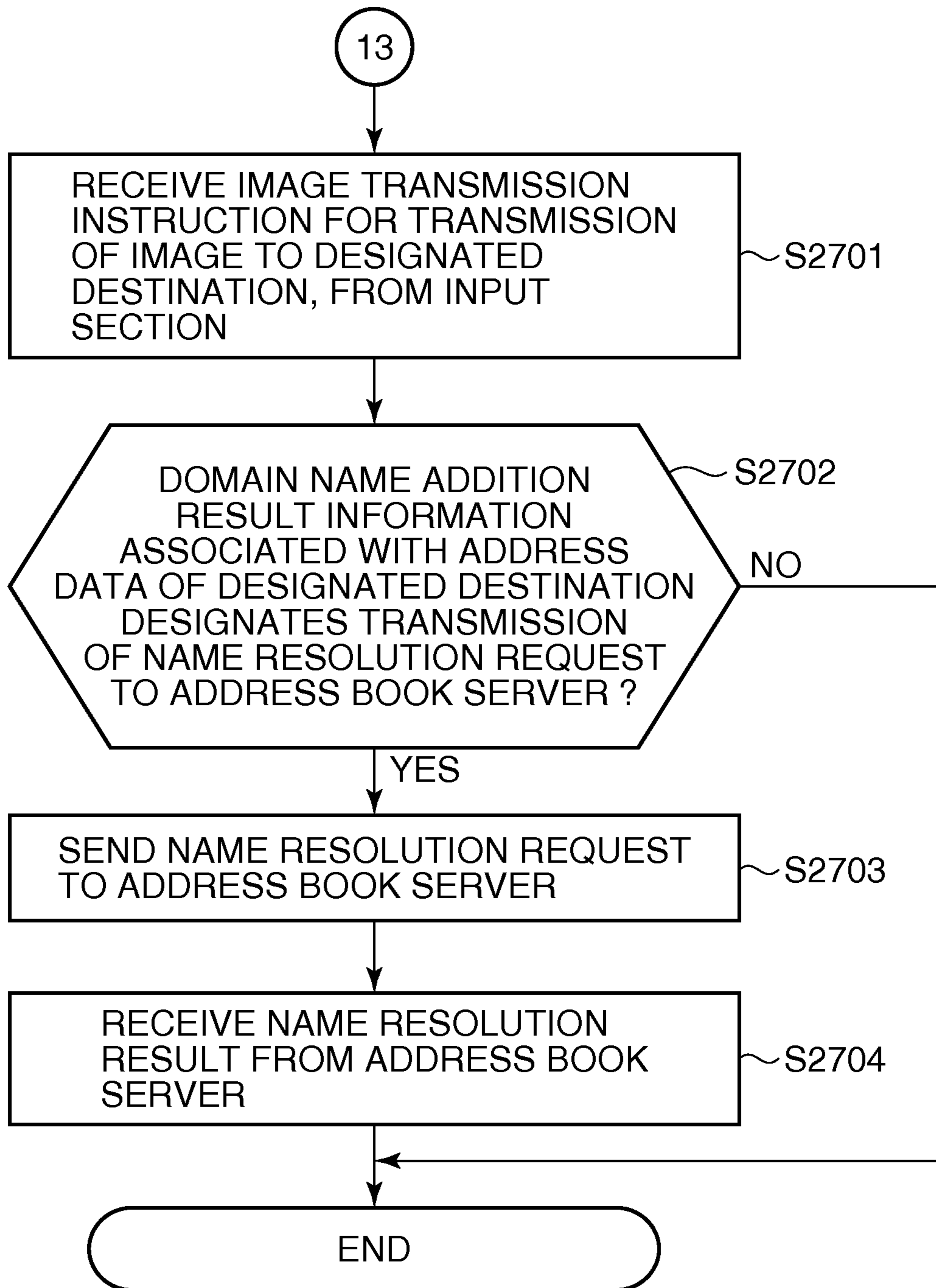


FIG. 18

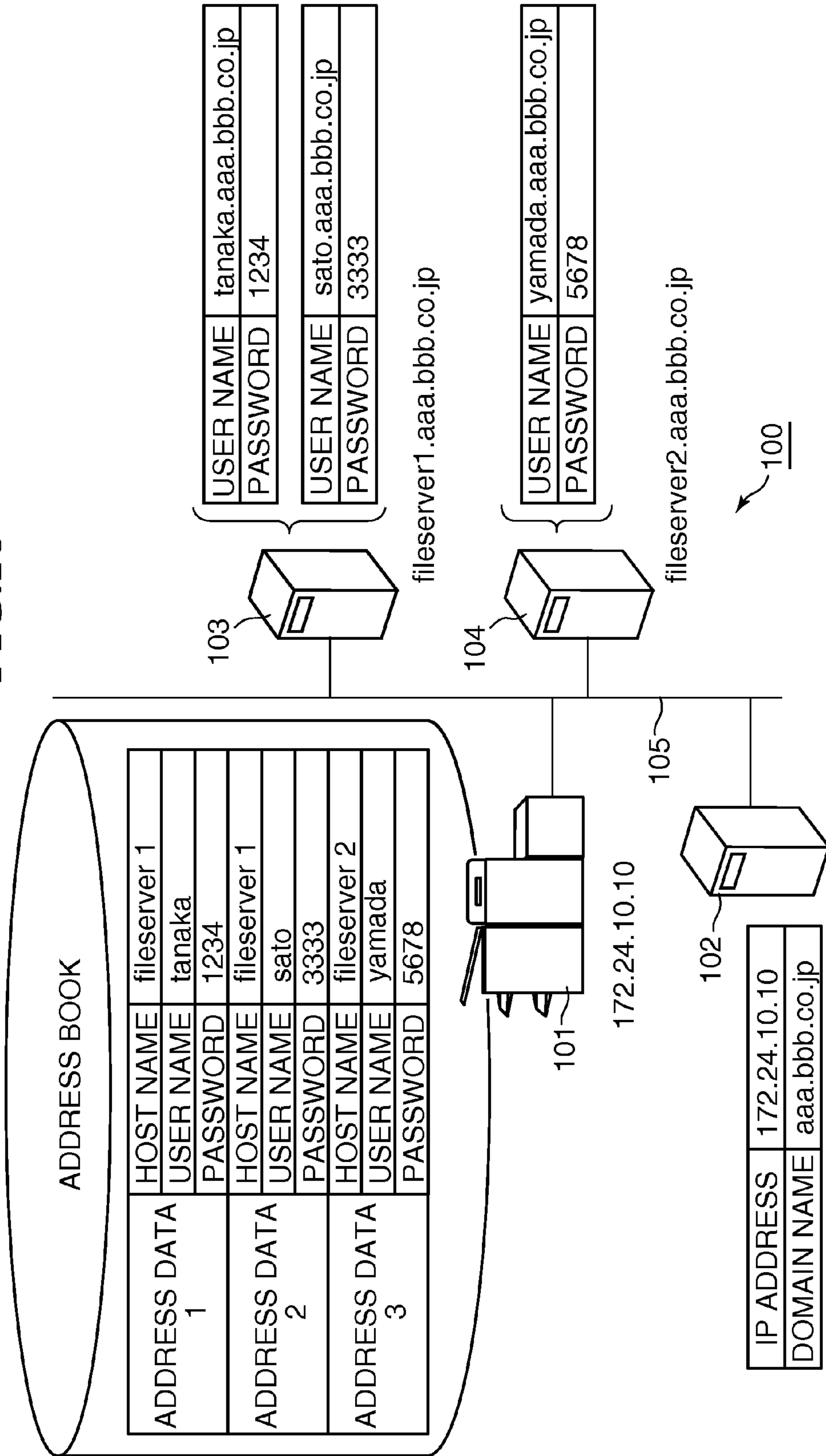
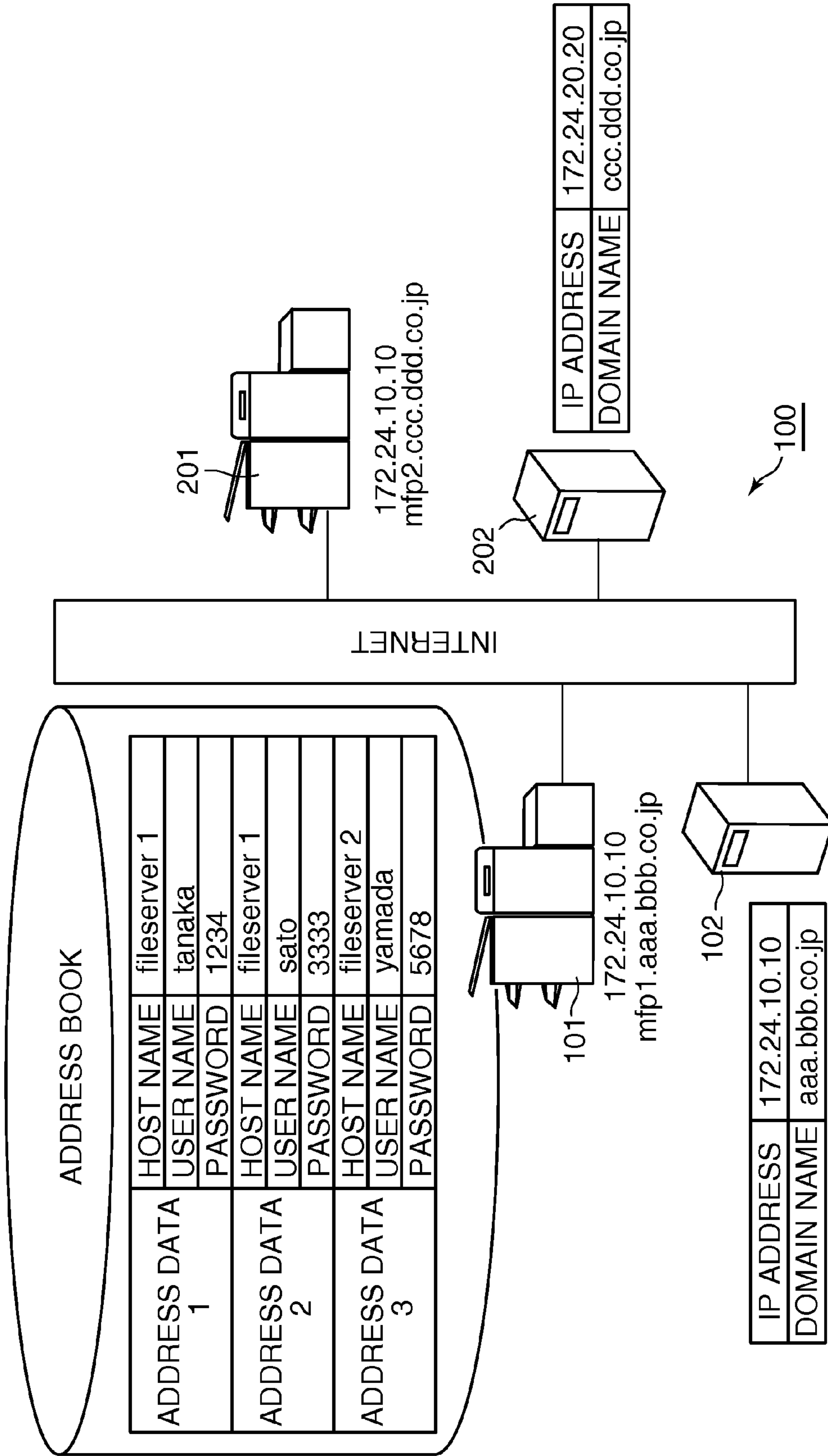


FIG. 19



**IMAGE PROCESSING APPARATUS,
CONTROL METHOD, AND STORAGE
MEDIUM THAT COMPLEMENT A DOMAIN
TO AN ADDRESS DATA ITEM WITH NO
DOMAIN NAME**

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to an image processing apparatus which functions as a server communicable with a client image processing apparatus via a network, and a method of controlling the image processing apparatus.

2. Description of the Related Art

Conventionally, there has been disclosed a technique in which e.g. two MFPs (Multiple Function Peripherals) are assigned functions of a server and functions of a client, respectively, and the client MFP acquires and utilizes address data acquirable from or stored in the server MFP. This kind of technique enables a client to perform transmission using address data (data storing the address of a transmission destination to which the client intends to perform transmission) acquired from a server (see e.g. Japanese Patent Laid-Open Publication No. 2005-94676).

On the other hand, there has been disclosed a technique related to name resolution (method of looking up the IP address of a host based on the associated host name). A description will be given of a system for performing name resolution, by taking a DNS (Domain Name System) as an example. The DNS is a system which associates host names with IP addresses on the Internet. In the DNS, it is possible to look up a host name based on an associated IP address, and vice versa.

For example, a name resolution unit (e.g. a DNS server) is set in each of a server MFP and a client MFP, and the name resolution is performed using the set name resolution units. In this method, when a domain name is not described in address data designating a transmission destination, the name resolution is generally performed after the address data is automatically complemented with a domain name by one of the name resolution units. Further, a method is also proposed in which the address data is configured to store not only a destination address, but also a user account and a password, and authentication is performed based on the user account and the password at a transmission destination (receiving end).

FIG. 18 shows parts of a system which uses “reverse DNS” to obtain a host name based on an IP address. The system 100 comprises an MFP 101, a DNS server 102 and first and second file servers 103 and 104. The MFP 101, DNS server 102 and the file servers 103 and 104 are interconnected via a network 105 such as a Local Area Network (LAN).

The MFP 101 stores an address book having a plurality of entries corresponding to different respective users of the system. Each entry has a host name, user name and password. For example, the first entry in the address book, “address data 1”, has “fileserv1” as the host name, “tanaka” as the user name and “1234” as the password.

The host name identifies the name of a host assigned to the first file server 103. The full host name for the first file server 103 is “fileserv1.aaa.bbb.co.jp” but, as shown in FIG. 18, the address data 1 only specifies “fileserv1”, i.e. the domain name “aaa.bbb.co.jp” is omitted. Similarly, the full user name for the user “tanaka” is “tanaka.aaa.bbb.co.jp” but the address data 1 only specifies “tanaka”. It is possible to omit the domain name in this case because the MFP 101 and the file servers 103 and 104 belong to the same network, i.e. they share the same domain name.

A user who wishes to transmit image data from the MFP 101 to another user or to himself, selects the relevant address-book entry, for example the address data 1 for the user “tanaka”. In this case, the image data should be transmitted to a directory corresponding to “tanaka” of the file server 103 having the address “fileserv1.aaa.bbb.co.jp”. To obtain this address the MFP 101 obtains the domain name “aaa.bbb.co.jp” from the DNS server 102 based on the IP address of MFP 101 which is “172.24.10.10” and adds this domain name to “fileserv1” and “tanaka”. As a result, the complemented address data “fileserv1.aaa.bbb.co.jp” and “tanaka.aaa.bbb.co.jp” can be used as the destination of the image data.

FIG. 19 shows another system having, in addition to the MFP 101 and DNS server 102 of FIG. 18, a further MFP 201 and a further DNS server 202. The MFP 101 uses the DNS server 102 to obtain a domain name based on an IP address, whereas the further MFP 201 uses the further DNS server 202 to obtain a domain name based on an IP address. The MFP 101 and DNS server 102 are connected to the MFP 201 and DNS server 202 via the Internet. The domain name of the MFP 201 is “ccc.ddd.co.jp”, which is different from the domain name “aaa.bbb.co.jp” of the MFP 101.

In the system of FIG. 19, a problem occurs when the address book of the MFP 101 is exported to the MFP 201 which uses the DNS server 202 different from the DNS server 102 used by the MFP 101. This problem arises because the address book of the MFP 101 does not include domain names. If a user of MFP 102 selects the exported address data 1, in order to transmit image data from the MFP 102 to the user “tanaka”, the domain name “ccc.ddd.co.jp” is added to the host name and the user name for “tanaka”. As a result, an incorrect host name “fileserv1.ccc.ddd.co.jp” and an incorrect user name “tanaka.ccc.ddd.co.jp” are used to transmit the image data, and a transmission error (destination unknown) occurs.

Thus, according to the above-described conventional technique, in the system comprised of the server MFP (MFP 101) that makes address data acquirable or stores the same and the client MFP (MFP 201) that performs transmission using the address data, if the server MFP and the client MFP have different name resolution units (e.g. DNS servers 102 and 202) set therefor, the following problems occur: When address data is exchanged in a state where an associated domain name is not described, there is a possibility that authentication or transmission might fail due to complementary addition of a wrong domain name. Further, it is difficult for a user who intends to perform transmission using address data to check whether or not a domain name is described in the address data.

SUMMARY OF THE INVENTION

The present invention provides an image processing apparatus as a server which, when a reference request for referring to address data having no domain name described therein is received from a client or a user’s instruction for transmitting such address data to a client, complements the address data with a domain name, and then sends the complemented address data to the client.

In a first aspect of the present invention, there is provided an image processing apparatus operable as a server to communicate with a client image processing apparatus via a network, comprising a storage unit configured to store an address book containing a plurality of address data items including one or more address data items in which no domain name is described, a determination unit configured to determine

3

whether or not a domain name is described in a designated one of the stored address data items, a complementing unit operable when the determination unit determines that a domain name is not described in the designated address data item, to complement the designated address data item with a domain name, and a transmission unit configured to transmit the address data item, complemented with the domain name by the complementing unit, to the client image processing apparatus.

In a second aspect of the present invention, there is provided a method of controlling a first image processing apparatus that is operable as a server to communicate with a client image processing apparatus via a network, and that stores an address book containing a plurality of address data items including one or more address data items in which no domain name is described, comprising determining whether or not a domain name is described in a designated one of the address data items stored by the first image processing apparatus, complementing, when it is determined that a domain name is not described in the designated address data item, the designated address data item with a domain name, and transmitting the address data item, complemented with the domain name, to the client image processing apparatus.

In a third aspect of the present invention, there is provided a computer-readable storage medium storing a computer-readable program for causing a computer to execute a method of controlling a first image processing apparatus that is operable as a server to communicate with a client image processing apparatus via a network, and that stores an address book containing a plurality of address data items including one or more address data items in which no domain name is described, wherein the method comprises determining whether or not a domain name is described in a designated one of the address data items stored by the first image processing apparatus, complementing, when it is determined that a domain name is not described in the designated address data item, the designated address data item with a domain name, and transmitting the address data item, complemented with the domain name, to the client image processing apparatus.

According to the present invention, when a reference request for referring to address data having no domain name described therein is received, it is possible to complement the address data with a domain name and then transmit the same. This makes it possible e.g. to send address data correctly to a designated destination, and to reduce failure of authentication.

Further features of the present invention will become apparent from the following description of exemplary embodiments with reference to the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a server image processing apparatus according to a first embodiment of the present invention.

FIG. 2 is a block diagram of a client image processing apparatus.

FIG. 3 is a schematic diagram of a whole image processing system.

FIG. 4 is a flowchart of a server process executed by the server image processing apparatus.

FIG. 5 is a continuation of FIG. 4.

FIG. 6 is a continuation of FIG. 4.

FIG. 7 is a flowchart of a client process executed by the client image processing apparatus.

FIG. 8 is a continuation of FIG. 7.

FIG. 9 is a continuation of FIG. 8.

4

FIG. 10 is a view of an example of an address book reference screen displayed on a display section of the client image processing apparatus.

FIG. 11 is a flowchart of a server process executed by a server image processing apparatus according to a second embodiment of the present invention.

FIG. 12 is a continuation of FIG. 11.

FIG. 13 is a continuation of FIG. 12.

FIG. 14 is a continuation of FIG. 11.

FIG. 15 is a flowchart of a client process executed by a client image processing apparatus.

FIG. 16 is a continuation of FIG. 15.

FIG. 17 is a continuation of FIG. 15.

FIG. 18 is a schematic diagram showing parts of a conventional image processing system.

FIG. 19 is a schematic diagram showing parts of another conventional image processing system.

DETAILED DESCRIPTION OF THE EMBODIMENTS

The present invention will now be described in detail below with reference to the accompanying drawings showing embodiments thereof.

FIG. 1 is a block diagram of a server image processing apparatus (hereinafter simply referred to as "the address book server") according to a first embodiment of the present invention.

Referring to FIG. 1, the address book server has a function of transmitting address data and a name resolution result to an address book client in response to an address book reference request and a name resolution request respectively sent from the address book client. The address book server is comprised of a CPU 11, a RAM 12, a hard disk 13, a main bus 14, an input section 15, a display section 16, and a communication section 17. The CPU 11 not only controls the units connected to the main bus 14, but also executes a server process shown in FIGS. 4 to 6, based on a program for implementing a control method according to the present invention.

The RAM 12 provides a work area and a data storage area for the CPU 11. The hard disk 13 stores the program for implementing the control method of the present invention, various kinds of data (including an address book storing a plurality of address data items), various settings, and so forth. The hard disk 13 may be replaced by an NVRAM or the like. The communication section 17 communicates with external apparatuses via a network. In the present embodiment, processes in the address book server are executed by the CPU 11 controlling the RAM 12, the hard disk 13, and the communication section 17 via the main bus 14, unless otherwise specified.

FIG. 2 is a block diagram of a client image processing apparatus (hereinafter simply referred to as "the address book client").

Referring to FIG. 2, the address book client comprises a CPU 21, a RAM 22, a hard disk 23, a main bus 24, an input section 25, a display section 26, a communication section 27, and a reading section 28. The CPU 21 not only controls the units connected to the main bus 24, but also executes a client process shown in FIGS. 7 to 9, according to a program. The RAM 22 provides a work area and a data storage area for the CPU 21. The hard disk 23 stores the program and various settings. The hard disk 23 may be replaced by an NVRAM or the like.

The input section 25 is used by users (including an operator and an installer) for inputting commands. The display section 26 displays various screens. The communication section 27

5

communicates with external apparatuses via the network. The reading section 28 reads an image as electronic data from an original such as a sheet original. In the present embodiment, processes in the address book client are executed by the CPU 21 controlling the RAM 22, the hard disk 23, the input section 25, the display section 26, the communication section 27, and the reading section 28 via the main bus 24, unless otherwise specified.

FIG. 3 is a schematic diagram of parts of an image processing system.

As shown in FIG. 3, the image processing system is comprised of an address book client C1, an address book client C2, an address book server SV1, a DNS server D1, and a DNS server D2, which are interconnected by a network N (including the Internet). In the illustrated example described hereafter, it is assumed that the DNS server D1 is set in the address book server SV1 as the DNS server to be queried for complementary addition of a domain name (i.e. the IP address of the DNS server D1 is stored in the address book server SV1), whereas the DNS server D2 is set in each of the address book clients C1 and C2 as the DNS server to be queried for complementary addition of a domain name (i.e. the IP address of the DNS server D2 is stored in each of the address book clients C1 and C2).

Although in the present embodiment, the DNS (Domain Name System) servers D1 and D2 are employed as means for performing name resolution, this is not limitative, and any other means, such as WINS (Windows (registered trademark) Internet Naming Service), for performing name resolution may be employed.

Next, the operation of the image processing system of the present embodiment, which is configured as above, will be described in detail with reference to FIGS. 4 to 10.

FIG. 4 is a flowchart of a server process executed by the address book server SV1. FIGS. 5 and 6 are continuations of FIG. 4. The present server process is executed by the CPU 11 of the address book server SV1 according to the program stored in the hard disk 13 (or the RAM 12).

Referring to FIG. 4, the CPU 11 of the address book server SV1 starts the present server process (step S1101) and recognizes an input received from an external apparatus via the communication section 17 (step S1102). Inputs from external apparatuses include an address book reference request and a name resolution request from an address book client, for example. The CPU 11 determines whether or not the input is an address book reference request from an address book client (step S1103) or whether or not the input is a name resolution request from an address book client (step S1104).

First, referring to FIG. 5, a description will be given of a case where it is determined in the step S1103 that the input is an address book reference request from an address book client. The CPU 11 of the address book server SV1 receives the address book reference request from the address book client via the communication section 17 (step S1201). Then, the CPU 11 acquires address data designated by the address book reference request from the hard disk 13 and stores the acquired address data in the RAM 12 (step S1202). Then, the CPU 11 performs the following processing on all the designated address data items (loop starting from a step S1203).

First, the CPU 11 determines whether or not a host name specified by a designated address data item is an IP address (step S1204). If the CPU 11 determines that the host name is an IP address, the process proceeds to a step S1207. If the host name is not an IP address, the CPU 11 determines whether or not a domain name is included (described) in the host name (step S1205).

6

If the CPU 11 determines that a domain name is included (described) in the host name, the process proceeds to the step S1207, whereas if not, the CPU 11 complements the host name with a domain name by querying the DNS server D1 via the communication section 17 (step S1206), i.e. the address data item is formed into address data in FQDN (Fully Qualified Domain Name) format. Then, the process proceeds to the step 1207.

Then, the CPU 11 determines whether or not a domain name is included in a user account (user name) by the designated address data item (step S1207). If a domain name is included in the user account, the process returns to the step S1203 to execute the present loop again for the next address data item designated in the address book reference request. Thus the loop is executed a number of times equal to the number of the designated address data items. If a domain name is not included in the user account, the CPU 11 complements the user account with a domain name by querying the DNS server D1 via the communication section 17 (step S1208), followed by the process returning to the step S1203.

When the process exits from the loop starting from the step S1203, the CPU 11 sends the designated address data, via the communication section 17, to the address book client that transmitted the address book reference request (step S1209), followed by the process returning to the step S1102.

The determination in the step S1204 as to whether or not the host name is an IP address is performed by determining whether or not the host name is formed by a combination of numerals, a dot, numerals, a dot, numerals, Further, the determination in the step S1205 as to whether or not a domain name is written in a host name is performed by determining whether or not the host name is formed by a combination of alphanumeric characters, a dot, alphanumeric characters, a dot, alphanumeric characters, Further, the determination in the step S1207 as to whether or not a domain name is written in a user account is performed by determining whether or not the user account is formed by a combination of alphanumeric characters, a dot, alphanumeric characters, a dot, alphanumeric characters,

Next, a detailed description will be given of the processing executed in the steps S1206 and S1208 for querying the DNS server D1 to complementarily add a domain name. Each of the steps S1206 and S1208 may be executed as follows: After a first query is made to the DNS server D1, a domain name is cached in a storage area of the RAM 12 or the like, and from the next execution of the loop, the cached domain name is acquired without making a query to the DNS server D1, and used for complementary addition of a domain name. This can be done when one or more address data items, in which a domain name is not described, in the address book of SV1 have the same domain name. In this case, the cached domain name is deleted after completion of check on all the designated address data items.

When a domain name cannot be complementarily added, for example the DNS server D1 is temporally down, in the step S1206 or S1208, the following processing method may be employed: The CPU 11 adds information indicative of failure of complementary addition of a domain name in association with the address data, as complementing result information, and sends the address data to the address book client via the communication section 17. Then, the address book client selectively determines which DNS server (the DNS server D2 or the address book server SV1) should execute name resolution based on the complementing result information. The complementing result information may be stored e.g. in the storage area of the RAM 12 separately from address data and be transmitted.

Next, referring to FIG. 6, a description will be given of a case where it is determined in the step S1104 that the input is a name resolution request from the address book client. The CPU 11 of the address book server SV1 receives the name resolution request from the address book client via the communication section 17 (step S1301).

Then, the CPU 11 makes a query to the DNS server D1 set in the address book server SV1 via the communication section 17, whereby the CPU 11 performs, using the DNS server D1, name resolution on a host name (a transmission destination to which an image is to be sent from the address book client) received from the address book client (step S1302). Next, the CPU 11 sends a name resolution result to the address book client via the communication section 17 (step S1303), followed by the process returning to the step S1102.

FIG. 7 is a flowchart of a client process executed by the address book client C1. FIG. 8 is a continuation of FIG. 7, and FIG. 9 is a continuation of FIG. 8. The present client process is executed by the CPU 21 of the address book client C1 according to the program stored in the hard disk 23 (or the RAM 22). The operation of the address book client C2 is identical to that of the address book client C1, and therefore description thereof is omitted.

Referring first to FIG. 7, the CPU 21 of the address book client C1 starts the present client process (step S1401) and recognizes a user input from the input section 25 (step S1402). If the input is an address book reference request for reference to the address book of the address book server SV1 (YES to a step S1403), the process proceeds to a step S1501 in FIG. 8, wherein the CPU 21 sends a reference request for reference to designated address data to the address book server SV1 via the communication section 27.

Then, the CPU 21 awaits transmission of the address data from the address book server SV1 and receives the address data via the communication section 27 (step S1502). The CPU 21 displays an address book reference screen (see FIG. 10) for reference to the received address data on the display section 26 (step S1503). On the address book reference screen shown in FIG. 10 are displayed an address book server name (File Server 1 in the illustrated example), a host name, a user name, and a password. The purpose of displaying the address book reference screen is to enable a user to check the host name, user name, and/or password to be used to transmit an image. FIG. 10 illustrates an example in which the host name and the user name are complemented with respective domain names.

Then, the CPU 21 recognizes a user input from the input section 25 (step S1504). If the input is not an image transmission instruction for image transmission to a designated destination (NO to a step S1505), the present process is immediately terminated. If the input is an image transmission instruction for image transmission to a designated destination (YES to a step S1505), the process proceeds to a step S1601 in FIG. 9, wherein first, the CPU 21 determines whether or not a host name stored in the designated address data is an IP address. If the host name is an IP address, the present process is immediately terminated. If the host name is not an IP address, it is determined whether or not the host name has a domain name written therein (step S1602).

If the host name has a domain name written therein, the present process is immediately terminated. If the host name does not have a domain name written therein, the CPU 21 sends a name resolution request to the address book server SV1 via the communication section 27 (step S1603). The CPU 21 awaits transmission of a name resolution result from the address book server SV1 and receives the name resolution

result via the communication section 27 (step S1604), followed by terminating the present process.

In the processing executed in the step S1501 for sending an address book reference request to an address book server, information (IP address or the like) on the DNS server D2 set in the address book client C1 may be transmitted along with the address book reference request. This makes it possible to determine whether or not the address book server should perform complementary addition of a domain name. That is, the address book server SV1 performs complementary addition of a domain name if the DNS server set in the address book client C1 is different from the DNS server set in the address book server SV1.

Further, when it is determined in the step S1505 that the input is an image transmission instruction for image transmission to the designated destination and when the address data received from the address book server contains complementing result information, by consulting the complementing result information, a name resolution request may be sent to the address book server (step S1603) without performing host name check (steps S1601 and S1602).

Furthermore, when the address data contains the complementing result information, the address book client C1 may directly send a name resolution request to the DNS server D1 set in the address book server in place of sending a name resolution request to the address book server. In this case, the complementing result information sent from the address book server contains the IP address of the DNS server D1 set in the address book server.

As described above in detail, according to the present embodiment, when address data is not fully descriptive, e.g. due to absence of a domain name, the address data is complemented with a domain name to make the same fully descriptive, and is then transmitted. This makes it possible to send (export) address data correctly to a designated destination, and to reduce failure of authentication.

Next, a description will be given of a second embodiment of the present invention. The present embodiment is identical in hardware configuration to the above-described first embodiment, and hence while omitting duplicate illustration and description of the hardware configuration, the following description will be given using the same reference numerals denoting the corresponding elements thereof. The present embodiment is basically different from the first embodiment in the following points:

A user inputs an address data transmission instruction for sending designated address data to an address book client, from the input section of the address book server, whereby the address book server operates in response to the address data transmission instruction (step S2103).

An address book server acquires information on a name resolution unit (DNS server) set in an address book client, to thereby determine whether or not to complementarily add a domain name (steps S2203 to S2206).

The address book server stores a domain name addition result as domain name addition result information in address data (step S2305).

The address book server sends information on the address book server to the address book client (step S2308).

The address book client sends the information on the name resolution unit set therein to the address book server (step S2506).

The address book client determines a name resolution request destination based on domain name addition result information stored in the address data (step S2702).

And, the address book client receives an IP address of the address book server from the address book server and stores the same (steps S2601 and S2602).

FIG. 11 is a flowchart of a server process executed by the address book server. FIG. 12 is a continuation of FIG. 11. FIG. 13 is a continuation of FIG. 12. FIG. 14 is a continuation of FIG. 11. The present server process is executed by the CPU 11 of the address book server SV1 according to a program stored in the hard disk 13 (or the RAM 12).

Referring first to FIG. 11, the CPU 11 of the address book server SV1 starts the present server process (step S2101) and recognizes a user input from the input section 15 or an input received from an external apparatus via the communication section 17 (step S2102). Inputs from the input section 15 and the communication section 17 include an address data transmission instruction from the input section 15 and a name resolution request received from an address book client via the communication section 17, for example. The address data transmission instruction is input via the input section 15 that is for example operation panel, by a user of the address book server SV1. The CPU 11 determines whether or not the input is an address data transmission instruction from the input section 15 (step S2103) or whether or not the input is a name resolution request from an address book client (S2104).

First, referring to FIG. 12, a description will be given of a case where it is determined in the step S2103 that the input is an address data transmission instruction from the input section 15. The CPU 11 of the address book server SV1 receives the address data transmission instruction from the input section 15 (step S2201). Then, the CPU 11 acquires address data designated in the address data transmission instruction from the hard disk 13 and stores the acquired address data in the RAM 12 (step S2202).

Then, the CPU 11 sends to an address book client designated in the address data transmission instruction an IP address acquisition request for the IP address of a DNS server set in the address book client (i.e. the DNS server D2) via the communication section 17 (step S2203). Further, the CPU 11 receives the IP address of the DNS server D2 set in the address book client and sent back therefrom in response to the IP address acquisition request, via the communication section 17 (step S2204).

Then, the CPU 11 performs comparison between the received IP address of the DNS server D2 set in the address book client and the IP address of the DNS server D1 set in the address book server SV1 (step S2205). Further, the CPU 11 determines, based on determining conditions, whether or not the DNS server of the address book server SV1 and the DNS server of the address book client are identical, to thereby determine whether or not to complementarily add a domain name (step S2206). The determining conditions may be "YES if the IP addresses of all the DNS servers are identical" or "YES if at least one of the DNS servers set in the address book client is identical to one set in the address book server SV1", for example.

If the DNS server of the address book server and the DNS server of the address book client are identical (the IP addresses of the two DNS servers are identical), the process proceeds to a step S2308 in FIG. 13, wherein the CPU 11 sends the designated address data and the IP address of the address book server SV1 to the address book client via the communication section 17 without complementing the designated address data with a domain name, followed by the process returning to the step S2102.

If the DNS server of the address book server and the DNS server of the address book client are not identical (the IP addresses of the two DNS servers are different from each

other), the process proceeds to a step S2301 in FIG. 13, wherein the CPU 11 performs the following processing on all the designated address data items (loop starting from the step S2301). A user of the address book server SV1 designates at least one address data item to be transmitted (exported). The designated address data items can be one address data item of the address book or all of the address data items of the address book.

First, the CPU 11 determines whether or not a host name included in a designated address data item is an IP address (step S2302). If it is determined that the host name is an IP address, the CPU 11 executes a step S2306, referred to hereinafter. If the host name is not an IP address, the CPU 11 determines whether or not a domain name is written in the host name (step S2303). If a domain name is written in the host name, the CPU 11 executes the step S2306.

If it is determined that a domain name is not written in the host name, the CPU 11 complements the host name with a domain name by querying the DNS server D1 via the communication section 17 (step S2304), i.e. the address data item is formed into address data in FQDN format. Then, the CPU 11 stores the result of the complementary addition of the domain name as domain name addition result information in the address data (step S2305). In the domain name addition result information, information indicating "Domain name addition is successful" or "Domain name addition is unsuccessful, and therefore a name resolution request should be sent to the address book server" is stored as a flag or the like. The domain name addition result information may be stored e.g. in the RAM 12 separately from the address data.

Then, the CPU 11 determines whether or not a domain name is written in a user account (step S2306). If a domain name is written in the user account, the process returns to the step S2301. If a domain name is not written in the user account, the CPU 11 complements the user account with a domain name by querying the DNS server D1 via the communication section 17 (step S2307), followed by the process returning to the step S2301 execute the present loop a number of times corresponding to the number of the designated address data items.

When the process exits from the loop starting from the step S2301, the CPU 11 sends the address data and the IP address of the address book server SV1 via the communication section 17 to the address book client that transmitted the IP address of the DNS server D2 (step S2308), followed by the process returning to the step S2102.

Next, a detailed description will be given of the processing executed in the steps S2304 and S2307 for querying the DNS server D1 to complementarily add a domain name. The steps S2304 and S2307 may be executed as follows: After a first query is made to the DNS server D1, a domain name is cached in a storage area of the RAM 12 or the like, and from the next execution of the loop, the cached domain name is acquired without making a query to the DNS server D1 and is used for complementary addition of a domain name. In this case, the cached domain name is deleted after check on all the designated address data items is completed.

Next, referring to FIG. 14, a description will be given of a case where it is determined in the step S2104 that the input is a name resolution request from an address book client. The CPU 11 receives the name resolution request from the address book client via the communication section 17 (step S2401). Then, the CPU 11 makes a query to the DNS server D1 set in the address book server SV1 via the communication section 17, whereby the CPU 11 performs, using the DNS server D1, name resolution on a host name received from the address book client (step S2402). Next, the CPU 11 sends a name

11

resolution result to the address book client via the communication section 17 (step S2403), followed by the process returning to the step S2102.

FIG. 15 is a flowchart of a client process executed by the address book client C1. FIGS. 16 and 17 are continuations of FIG. 15. The present client process is executed by the CPU 21 of the address book client C1 according to a program stored in the hard disk 23 (or the RAM 22). The operation of the address book client C2 is identical to that of the address book client C1, and therefore description thereof is omitted.

Referring first to FIG. 15, the CPU 21 of the address book client C1 starts the present client process (step S2501) and recognizes an input to the address book client C1 (step S2502). In the following, a description will be given of each of a case where the input is an address data reception request from the address book server, a case where the input is an image transmission instruction from the input section 25, and a case where the input is a DNS server information acquisition request from the address book server.

First, a description will be given of the case where the input is an address data reception request from the address book server (YES to a step S2503). In this case, referring to FIG. 16, the CPU 21 receives address data that is designated at the address book server and the IP address of the address book server from the address book server via the communication section 27 (step S2601). Then, the CPU 21 stores the address data and the IP address of the address book server received from the address book server, in the hard disk 23 in association with each other (step S2602), followed by terminating the present process.

Next, a description will be given of the case where the input is an image transmission instruction from the input section 25 (YES to a step S2504). In this case, referring to FIG. 17, the CPU 21 receives the image transmission instruction for sending image data to a destination designated in the image transmission instruction (step S2701) from the input section 25. Then, the CPU 21 performs determination as to the contents of domain name addition result information associated with the address data of the designated destination (step S2702). The domain name addition result information is included in the received address data (step 2305).

If the domain name addition result information does not designate transmission of a name resolution request to the address book server, the present process is immediately terminated. If the domain name addition result information designates the transmission of the name resolution request, the CPU 21 sends the name resolution request to the address book server via the communication section 27 (step S2703). Then, the CPU 21 receives a name resolution result from the address book server via the communication section 27 (step S2704), followed by terminating the present process.

Next, referring to FIG. 15, a description will be given of the case where the input is a DNS server information acquisition request from the address book server (YES to a step S2505). In this case, the CPU 21 sends information (including an IP address) on the DNS server D2 set in the address book client C1 to the address book server via the communication section 27 (step S2506). Various methods have been conventionally proposed as transmission methods, but they are not directly related to the present invention, and hence, description thereof is omitted.

As described above, according to the present embodiment, when address data is not fully descriptive e.g. due to absence of a domain name, the address data is complemented with a domain name to make the same fully descriptive, and is then transmitted. This makes it possible to send address data cor-

12

rectly to a designated destination, to reduce failure of authentication, and to check information more correctly at a receiving end.

According to the second embodiment, it is determined whether or not a DNS server set in an address book client and a DNS server set in the address book server are identical, and then processing is executed differently depending on the determination. However, for example, the address book server may store, in advance, identifier information for identifying an address book client in which is set the same DNS server as that set in the address book server. In this case, when an address data transmission instruction for sending address data to the address book client in which is set the same DNS server as that set in the address book server is received via the input section, processing for complementarily adding a domain name is not carried out.

Further, in a case where it is possible to set a plurality of DNS servers in a single address book client, if at least one of the DNS servers set in the address book client is identical to the DNS server set in the address book server, the address book client may be regarded as having the same DNS server set therein.

Aspects of the present invention can also be realized by a computer of a system or apparatus (or devices such as a CPU or MPU) that reads out and executes a program recorded on a memory device to perform the functions of the above-described embodiments, and by a method, the steps of which are performed by a computer of a system or apparatus by, for example, reading out and executing a program recorded on a memory device to perform the functions of the above-described embodiments. For this purpose, the program is provided to the computer for example via a network or from a recording medium of various types serving as the memory device (e.g., computer-readable medium).

While the present invention has been described with reference to exemplary embodiments, it is to be understood that the invention is not limited to the disclosed exemplary embodiments. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

This application claims the benefit of Japanese Patent Application No. 2009-050755, filed Mar. 4, 2009, and Japanese Patent Application No. 2010-044499, filed Mar. 1, 2010, which are hereby incorporated by reference herein in its entirety.

What is claimed is:

1. An image processing apparatus communicable with a client image processing apparatus via a network, the image processing apparatus comprising:

a storage device storing an address book containing at least one address data item in which a host name is described but no domain name is described, each of the at least one address data item being usable as a transmission destination to which an image is transmitted;

a controller configured to:

receive a request of the address book from the client image processing apparatus, the request being for requesting the image processing apparatus to transmit address data items contained therein, to the client image processing apparatus;

acquire from the client image processing apparatus an IP address of a DNS server set in the client image processing apparatus;

compare between the acquired IP address and an IP address of a DNS server set in the image processing apparatus;

13

determine, based on a result of the comparison, whether or not to add a domain name of the image processing apparatus;

add, based on a result of the determination, the domain name of the image processing apparatus to the at least one address data item in which the host name is described but no domain name is described, when the request of the address book is received from the client image processing apparatus; and

transmit the at least one address data item, to which the domain name of the image processing apparatus is added, to the client image processing apparatus.

2. The image processing apparatus according to claim 1, wherein the controller is configured to add the domain name of the image processing apparatus to the at least one address data item in which the host name is described but no domain name is described by querying a DNS server via the network, to thereby form the address data item into address data in Fully Qualified Domain Name format.

3. The image processing apparatus according to claim 1, wherein the controller is configured to:

when a name resolution request is received from the client image processing apparatus, perform name resolution on a transmission destination to which an image is to be sent from the client image processing apparatus; and send a result of the name resolution to the client image processing apparatus.

4. The image processing apparatus according to claim 3, wherein the controller performs the name resolution using a DNS server by querying the DNS server via the network.

5. The image processing apparatus according to claim 1, wherein the controller is configured to:

store a result of the addition of the domain name in the at least one address data item, as complementing result information; and

send the at least one address data item, to which the domain name of the image processing apparatus is added, and with the complementing result information, to the client image processing apparatus.

6. The image processing apparatus according to claim 1, wherein the controller is configured to:

when the IP addresses of the two DNS servers are identical, not add the domain name of the image processing apparatus, and

when the IP addresses of the two DNS servers are different from each other, add the domain name.

7. The image processing apparatus according to claim 1, wherein each of the address data items is designated by an address data transmission instruction, the address data transmission instruction being inputtable to the claimed image processing apparatus by a user thereof to cause the claimed image processing apparatus to transmit the respective address data item to the client image processing apparatus.

8. A method of controlling a first image processing apparatus communicable with a client image processing apparatus via a network, and that stores an address book containing at least one address data item in which a host name is described but no domain name is described, each of the at least one address data item being usable as a transmission destination to which an image is transmitted, the method comprising the steps of:

14

receiving a request of the address book from the client image processing apparatus, the request being for requesting the image processing apparatus to transmit address data items contained therein, to the client image processing apparatus;

acquiring from the client image processing apparatus an IP address of a DNS server set in the client image processing apparatus;

comparing between the acquired IP address and an IP address of a DNS server set in the image processing apparatus;

determining, based on a result of the comparing step, whether or not to add a domain name of the image processing apparatus;

add, based on a result of the determining step, a domain name of the image processing apparatus to the at least one address data item in which the host name is described by not domain name is described, when the request of the address book is received in the receiving step; and

transmitting the at least one address data item, to which the domain name of the image processing is added, to the client image processing apparatus.

9. A non-transitory computer-readable storage medium storing a computer-readable program executable by a computer to execute a method of controlling a first image processing apparatus that is communicable with a client image processing apparatus via a network, and that stores an address book containing at least one address data item in which a host name is described but no domain name is described, each of the at least one address data item being usable as a transmission destination to which an image is transmitted, the method comprising the steps of:

receiving a request of the address book from the client image processing apparatus, the request being for requesting the image processing apparatus to transmit address data items contained therein, to the client image processing apparatus;

acquiring from the client image processing apparatus an IP address of a DNS server set in the client image processing apparatus;

comparing between the acquired IP address and an IP address of a DNS server set in the image processing apparatus;

determining, based on a result of the comparing step, whether or not to add a domain name of the image processing apparatus;

add, based on a result of the determining step, a domain name of the image processing apparatus to the at least one address data item in which the host name is described by not domain name is described, when the request of the address book is received in the receiving step; and

transmitting the at least one address data item, to which the domain name of the image processing is added, to the client image processing apparatus.

10. The image processing apparatus according to claim 1, wherein the controller is configured to receive another request of the address book from another client image processing apparatus belonging to another domain.

* * * * *