



(51) International Patent Classification:
G06Q 50/32 (2012.01)

(21) International Application Number:
PCT/US2013/066543

(22) International Filing Date:
24 October 2013 (24.10.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/662,153 26 October 2012 (26.10.2012) US

(71) Applicant: **GOOGLE INC.** [US/US]; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(72) Inventors: **LIN, Liang**; c/o GOOGLE INC., 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **SUNDARARAMAN, Jaishankar**; c/o GOOGLE INC., 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **RAGHAVAN, Balaji**; c/o GOOGLE INC., 1600 Amphitheatre Parkway, Mountain View, California

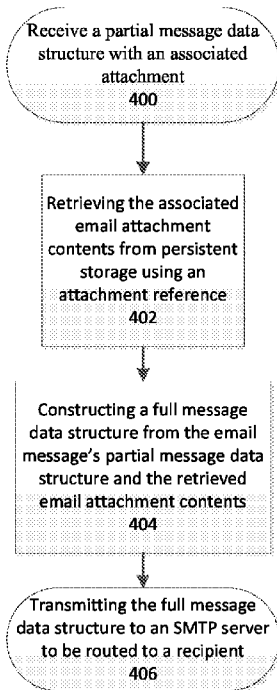
94043 (US). **CARPENTER, Wiltse**; c/o GOOGLE INC., 1600 Amphitheatre Parkway, Mountain View, California 94043 (US). **BUCY, John**; Google San Francisco, 345 Spear Street, Floors 2-4, San Francisco, California 94105 (US). **RENGASWAMY, Ram Kumar**; Google New York, 76 Ninth Avenue, 4th Floor, New York, New York 10011 (US).

(74) Agent: **CAMMARATA, Michael R.**; c/o Birch, Stewart, Kolasch & Birch, LLP, P.O. Box 747, Falls Church, Virginia 22040-0747 (US).

(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,

[Continued on next page]

(54) Title: IMPROVING ATTACHMENT HANDLING DURING EMAIL DELIVERY



(57) Abstract: A system and method for improving attachment handling during email delivery is disclosed. A partial message data structure stores all data associated with an email except the email's attachments' contents, although attachment contents may be stored in the partial message data structure if necessary (400, 402). When an email is sent to a recipient, a message router constructs the full email message including attachments from the partial message data structure and the stored attachment contents before sending the email to an SMTP server (404, 406). When an email is received from an SMTP server (600), a message router transmits the email to a message parser that parses any attachments from the email and stores the attachments in persistent storage (602). The message parser then sends a partial message data structure containing data associated with the email and references to the email's associated attachments to the message router (604). The message router transmits the partial message data structure to a server for delivery (606).

FIG. 4

WO 2014/066597 A1

TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

(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,

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

IMPROVING ATTACHMENT HANDLING DURING EMAIL DELIVERY

BACKGROUND

[0001] Electronic mail, or email, is a powerful communication tool that allows information to be exchanged nearly instantaneously. There are several email client applications that allow users to draft email messages through a user interface and send the messages to recipients. Some email messages may contain files as attachments that are delivered to a recipient along with the message. An email attachment may be any file type including a spreadsheet, a word processing document, or a photograph. One or more files can be attached to a given email message. Attachments can vary in size, typically ranging from a few bytes to twenty-five megabytes.

[0002] In a conventional system, data structures called full message data structures are created to organize, store, and manipulate email messages and their attachments. Typically, these data structures are in Multipurpose Internet Mail Extensions (MIME) format and are flat byte strings. Full message data structures include all information regarding an email including sender, recipient, and subject. These data structures also include all content of associated attachments as well as the attachments' metadata. Full message data structures can include information regarding the sender, the recipient, the message, MIME-Version, Content-ID, Content-Type, Content-Disposition, Content-Transfer-Encoding, and Encoded-Word. The Full message data structure may contain information per attachment including: Content-Type, Content-Transfer-Encoding, Content-ID (used for displaying an attachment in-line such as an advertisement or a picture), Content-Disposition, and the actual content of the attachment. A full message data structure representing an email is transmitted internally anytime the email and/or its attachments are modified or when the email is sent to a recipient. FIG. 1 illustrates an example of a full message data structure containing two image attachments (111a,b).

[0003] When a user clicks a send button, or initiates a send action in some other manner in an email client application, the email's full message data structure is routed through the sender's email client application's servers (202a), referred to as internal mail servers, to Simple Mail Transfer Protocol (SMTP) servers (204), and then to servers controlled by the email recipient's email client application (202b) as shown in FIG. 2.

[0004] For example, a user (200a) using an email client application may want to send a message to a recipient who uses a different email client application (200b) for email. The

sender may draft an email. Once an email has been drafted and the appropriate attachments attached, a user may click send, starting the process of routing the email to the recipient's email client's application servers. As discussed above, emails are typically relayed from an email client to a mail server via SMTP in MIME format.

[0005] In a conventional system, the full message data structure is transmitted internally in an email client application anytime there are changes to the email or when an email is transmitted to a recipient. The full message data structure size makes routing slower since sending a data structure which contains all information including attachment contents increases the latency of transmission. However, it is not always necessary to transmit or route the full message structure. Most internal manipulation of email messages does not require actual email attachment content information. When a user interacts with an email client application, the user often performs operations which merely require the subject and body of an email and do not require loading attachments. For example, users may read through old messages or forward an email to a friend. Attachments do not need to be loaded from a server in order to accomplish these tasks. Since the attachments are not necessary, there is no need to load attachments into the server especially when the server is serving a lot of users. Therefore, there should be a way to improve attachment handling so that emails can be routed to SMTP servers and delivered to email client application servers with reduced latency. Routing emails without their associated attachments also requires less network bandwidth since huge email data structures will not be sent across the network. Additionally, since attachments are often large, not storing these attachments in memory frees up the memory for other uses. Finally, less CPU time is required to process emails without their large attachments.

SUMMARY

[0006] This specification describes technologies relating to email attachment handling in general, and specifically to methods and systems for improving attachment handling during email delivery and receipt.

[0007] In general, one aspect of the subject matter described in this specification can be embodied in a system and method for improving email attachment sending flow. An exemplary system includes: one or more processing devices and one or more storage devices storing instructions that, when executed by the one or more processing devices, cause the one or more processing devices to: receive a partial message data structure for an email with an

associated attachment; retrieve the associated email attachment contents from persistent storage using an attachment reference; construct a full message data structure from the email's partial message data structure and the retrieved email attachment contents; and transmit the full message data structure to an SMTP server to be routed to a recipient. An exemplary method includes: receiving a partial message data structure for an email with an associated attachment; retrieving the associated email attachment contents from persistent storage using an attachment reference; constructing a full message data structure from the email's partial message data structure and the retrieved email attachment contents; and transmitting the full message data structure to an SMTP server to be routed to a recipient. Another aspect of the subject matter described in this specification can be embodied in a system and method for improving email attachment handling delivery. An exemplary method includes: receiving an email message that contains an attachment; transmitting the email to a message parser which parses the attachment from the email and stores the attachment in persistent storage; receiving a partial message data structure from the message parser which includes email data and a reference to the attachment; and transmitting the partial message data structure to a server for delivery.

[0008] These and other embodiments can optionally include one or more of the following features: (1) an attachment reference may be obtained from the partial message data structure; (2) the attachment reference may be obtained from client cache; (3) the partial message may be stored in Protocol Buffer format; (4) the partial message may be stored in XML format; and (5) the partial message may be stored as an object in a programming language.

[0009] The details of one or more embodiments of the invention are set forth in the accompanying drawings which are given by way of illustration only, and the description below. Other features, aspects, and advantages of the invention will become apparent from the description, the drawings, and the claims. Like reference numbers and designations in the various drawings indicate like elements.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a block diagram illustrating a full message data structure.

[0011] FIG. 2 is a block diagram illustrating a conventional email delivery path.

[0012] FIG. 3 is a block diagram illustrating a partial message data structure for the same email that is represented by a full message data structure in FIG. 1.

[0013] FIG. 4 is a flow diagram of an exemplary method for improving email attachment sending flow.

[0014] FIG. 5 is a block diagram illustrating an exemplary email delivery path.

[0015] FIG. 6 is a flow diagram of an exemplary method for improving email attachment handling delivery.

[0016] FIG. 7 is a block diagram illustrating an exemplary email delivery system.

[0017] FIG. 8 is a block diagram illustrating an exemplary computing device.

DETAILED DESCRIPTION

[0018] Attachment contents often do not need to be transmitted through an email client application system until the email with which the attachment is associated is transmitted to a recipient. According to an exemplary embodiment, a data structure may be used to send details about email messages and their attachments throughout an email client application system without sending actual attachment contents until messages are ready to be sent to the SMTP server for delivery. This data structure is called a partial message data structure and may be created and stored as serialized structured data such as a protocol buffer. Protocol buffers are automated mechanisms for serializing structured data. Partial message data structures may also be in XML format. Additionally, partial message data structures may be stored as objects in programming languages such as Java, C++, or Python. Since the partial message data structure may include a reference to attachment contents instead of the attachment contents themselves and may therefore not be as large as a full message data structure, an email represented by a partial message data structure is more efficient for processing than the same email represented by a full message data structure.

[0019] In an exemplary embodiment when an email is created, a partial message data structure may be constructed to represent the email. This partial message data structure may organize and store all parts of an email except the contents of any associated attachments. Although a partial message data structure may store attachment contents if necessary, the data structure may contain only references to attachment contents. Instead of storing attachments in the partial data structure, attachment contents may be stored in remote persistent storage. FIG. 3 represents a partial message data structure for the same email that was represented by

a full message data structure in FIG. 1. In FIG. 3, 300a and 300b are attachment contents that are stored in remote persistent storage.

[0020] In some embodiments, a partial message data structure may be transmitted internally in an email client application anytime there are changes to the email or when an email is transmitted to a recipient. A partial message data structure's size makes routing an email faster than routing the same email represented by a full message data structure since a data structure which does not include attachment contents is smaller than a data structure that does contain attachment contents.

[0021] When a user clicks a send button, or initiates a send action in some other manner in an email client application, the email's partial message data structure is routed through the sender's email client application's servers. However, according to an exemplary embodiment, before being sent to a SMTP server, the partial message data structure is sent to an additional server or component that routes internal emails called a "message router." An exemplary method, as depicted by FIG. 4, begins with the message router (504) receiving a partial message data structure for an email with an associated attachment (400). The message router retrieves the associated email attachment contents from persistent storage using attachment references (402). References to attachments may be stored in the partial message data structure or in the client cache. When a user uploads attachments from an email client application user interface, the attachments may be sent to a storage system with their references returned to the client cache. If attachments are forwarded or autosaved, the attachments may already be in the remote persistent storage and their references may be stored in the partial message data structure of the attachments' corresponding email.

[0022] When a full message data structure needs to be constructed, attachments may be retrieved by the message router from remote persistent storage or the server cache, depending on where they are currently stored. The message router then constructs a full message data structure from the partial message data structure and the references to associated attachments which the partial message data structure may contain (404). The email's full message data structure may then be sent to Simple Mail Transfer Protocol (SMTP) servers, and to servers controlled by the email recipient's email client application as shown in FIG. 5 (406).

[0023] In an exemplary system, not only is data flow and transmission of attachments improved by use of partial message data structures, but email delivery is improved as well.

An exemplary embodiment receives an email with an associated attachment from an SMTP server via a message router as shown in FIG. 6. When a message router (705) receives an email from an SMTP server, the message router (705) routes the email to a message parser (707) (600, 602). In the message parser (707), a partial message data structure may be created from the email by removing any associated attachments from the email (602). The attachments may be sent to persistent storage (711) with references to the attachments being returned and stored in the persistent message data structure. The partial message data structure may then be transmitted back to the message router which then routes the data structure to other servers (709) for delivery as shown in FIG. 7 (606).

[0024] FIG. 8 is a block diagram illustrating an example computing device (800) that is arranged for improving attachment handling during email delivery and receipt. In a very basic configuration (801), the computing device (800) typically includes one or more processors (810) and system memory (820). A memory bus (830) can be used for communicating between the processor (810) and the system memory (820).

[0025] Depending on the desired configuration, the processor (810) can be of any type including but not limited to a microprocessor (μ P), a microcontroller (μ C), a digital signal processor (DSP), or any combination thereof. The processor (810) can include one or more levels of caching, such as a level one cache (811) and a level two cache (812), a processor core (813), and registers (814). The processor core (813) can include an arithmetic logic unit (ALU), a floating point unit (FPU), a digital signal processing core (DSP Core), or any combination thereof. A memory controller (816) can also be used with the processor (810), or in some implementations the memory controller (815) can be an internal part of the processor (810).

[0026] Depending on the desired configuration, the system memory (820) can be of any type including but not limited to volatile memory (such as RAM), non-volatile memory (such as ROM, flash memory, etc.) or any combination thereof. System memory (820) typically includes an operating system (821), one or more applications (822), and program data (824). The application (822) may include a method for improving attachment handling. Program Data (824) includes storing instructions that, when executed by the one or more processing devices, implement a system and method for improving attachment handling during email delivery and receipt. (823). In some embodiments, the application (822) can be arranged to operate with program data (824) on an operating system (821).

[0027] The computing device (800) can have additional features or functionality, and additional interfaces to facilitate communications between the basic configuration (801) and any required devices and interfaces.

[0028] System memory (820) is an example of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device 800. Any such computer storage media can be part of the device (800).

[0029] The computing device (800) can be implemented as a portion of a small-form factor portable (or mobile) electronic device such as a cell phone, a personal data assistant (PDA), a personal media player device, a wireless web-watch device, a personal headset device, an application-specific device, or a hybrid device that include any of the above functions. The computing device (800) can also be implemented as a personal computer including both laptop computer and non-laptop computer configurations.

[0030] The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. In one embodiment, several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, can be equivalently implemented in integrated circuits, as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of

skill in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies regardless of the particular type of non-transitory signal bearing medium used to actually carry out the distribution. Examples of a non-transitory signal bearing medium include, but are not limited to, the following: a recordable type medium such as a floppy disk, a hard disk drive, a Compact Disc (CD), a Digital Video Disk (DVD), a digital tape, a computer memory, etc.; and a transmission type medium such as a digital and/or an analog communication medium. (e.g., a fiber optic cable, a waveguide, a wired communications link, a wireless communication link, etc.)

[0031] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

[0032] Thus, particular embodiments of the subject matter have been described. Other embodiments are within the scope of the following claims. In some cases, the actions recited in the claims can be performed in a different order and still achieve desirable results. In addition, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.

What is claimed is:

1. A computer-implemented method for improving email attachment sending flow comprising:
 - receiving a partial message data structure for an email with an associated attachment(400);
 - retrieving the associated email attachment contents from persistent storage using an attachment reference (402);
 - constructing a full message data structure from the email's partial message data structure and the retrieved email attachment contents (404); and
 - transmitting the full message data structure to an SMTP server to be routed to a recipient (406).
2. The method of claim 1, wherein an attachment reference is obtained from the partial message data structure.
3. The method of claim 1, wherein the attachment reference is obtained from the client cache.
4. The method of claim 1, wherein the partial message is stored in Protocol Buffer format.
5. The method of claim 1, wherein the partial message is stored in XML format.
6. The method of claim 1, wherein the partial message is stored as an object in a programming language.
7. A computer-implemented method for improving email attachment handling delivery comprising:
 - receiving an email message that contains an attachment(600);
 - transmitting the email to a message parser which parses the attachment from the email and stores the attachment in persistent storage (602);

receiving a partial message data structure from the message parser which includes email data and a reference to the attachment (604); and

transmitting the partial message data structure to a server for delivery (606).

8. The method of claim 7, wherein an attachment reference is obtained from the partial message data structure.

9. The method of claim 7, wherein the attachment reference is obtained from the client cache.

10. The method of claim 7, wherein the partial message is stored in Protocol Buffer format.

11. A system for improving email attachment sending flow, the system comprising:

one or more processing devices; and

one or more storage devices storing instructions that, when executed by the one or more processing devices, cause the one or more processing devices to:

receive a partial message data structure for an email with an associated attachment (400);

retrieve the associated email attachment contents from persistent storage using an attachment reference (402);

construct a full message data structure from the email's partial message data structure and the retrieved email attachment contents (404); and

transmit the full message data structure to an SMTP server to be routed to a recipient (406).

12. The system of claim 11, wherein an attachment reference is obtained from the partial message data structure.

13. The system of claim 11, wherein the attachment reference is obtained from the client cache.

14. The system of claim 11, wherein the partial message is stored in Protocol Buffer format.
15. The system of claim 11, wherein the partial message is stored in XML format.
16. The system of claim 11, wherein the partial message is stored as an object in a programming language.
17. A non-transitory computer readable medium having stored therein computer executable code that causes one or more processors to execute the steps of:
 - receiving a partial message data structure for an email with an associated attachment (400);
 - retrieving the associated email attachment contents from persistent storage using an attachment reference (402);
 - constructing a full message data structure from the email's partial message data structure and the retrieved attachment contents (404);
 - transmitting the full message data structure to an SMTP server to be routed to a recipient (406).
18. The non-transitory medium of claim 17, wherein an attachment reference is obtained from the partial message data structure.
19. The non-transitory medium of claim 17, wherein the attachment reference is obtained from the client cache.
20. The non-transitory medium of claim 17, wherein the partial message is stored in Protocol Buffer format.

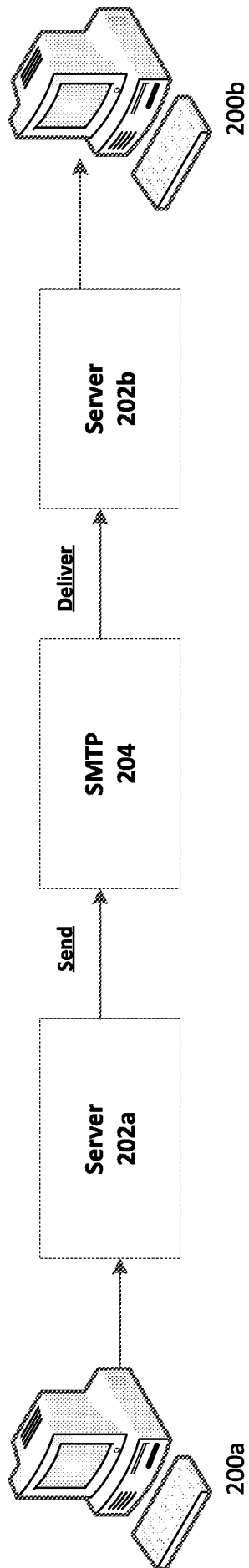


FIG. 2

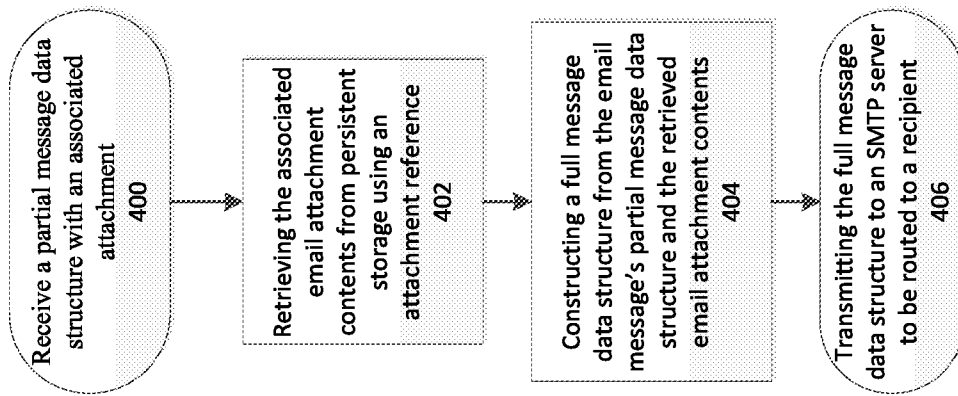


FIG. 4

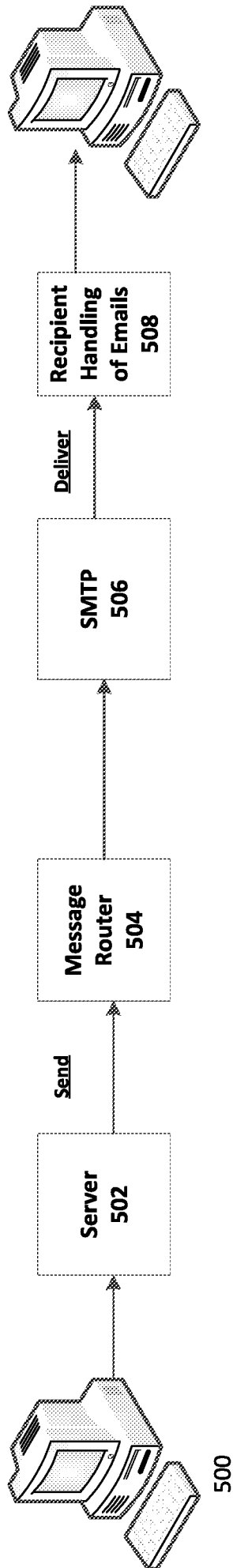


FIG. 5

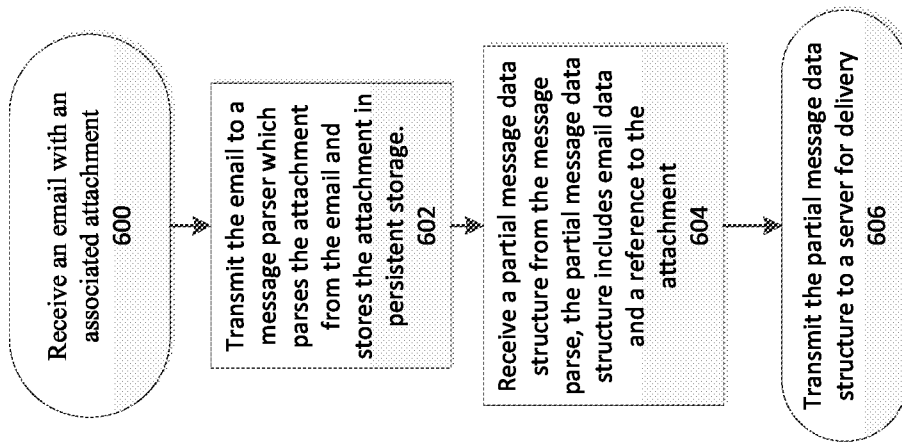


FIG. 6

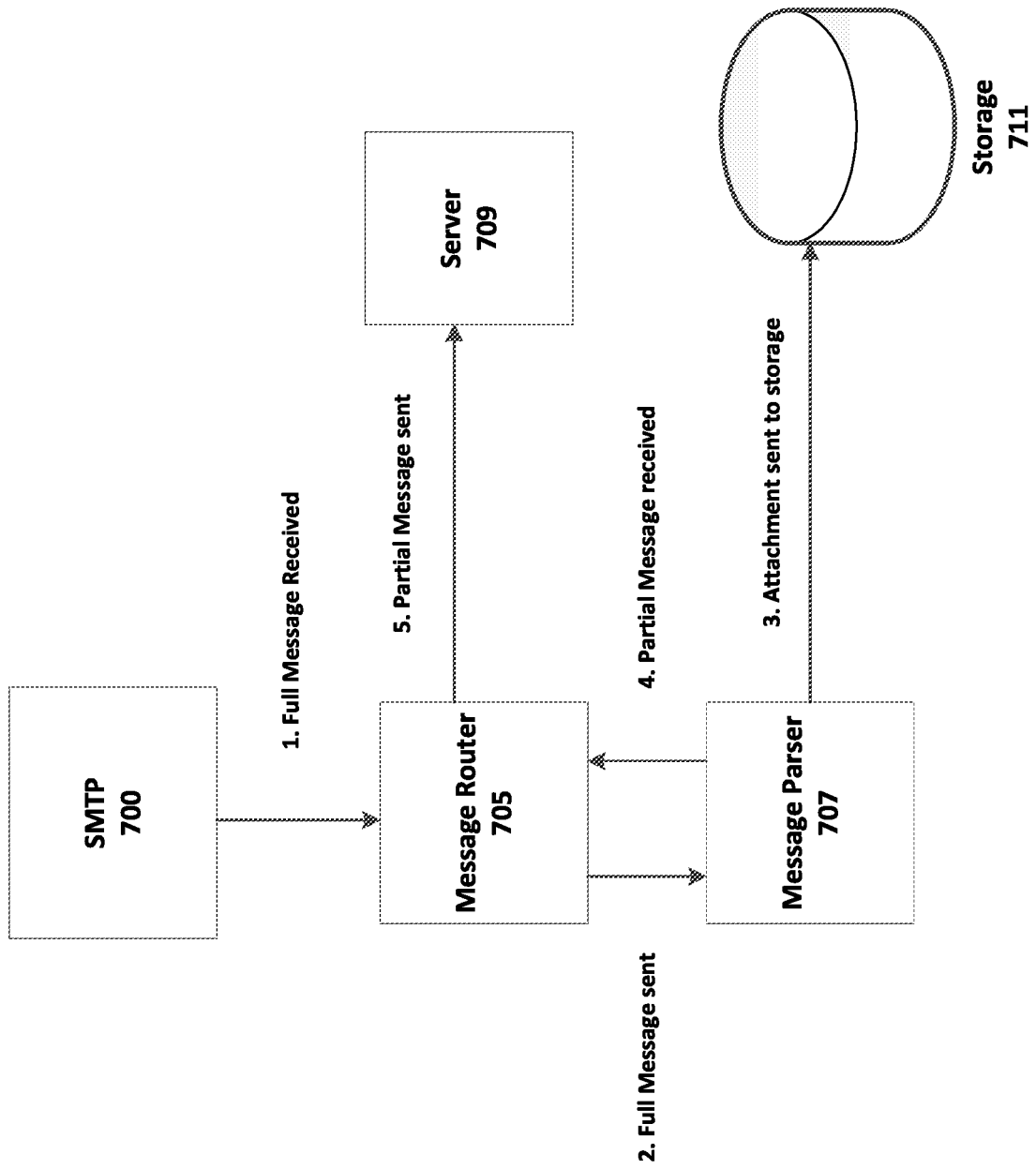


FIG. 7

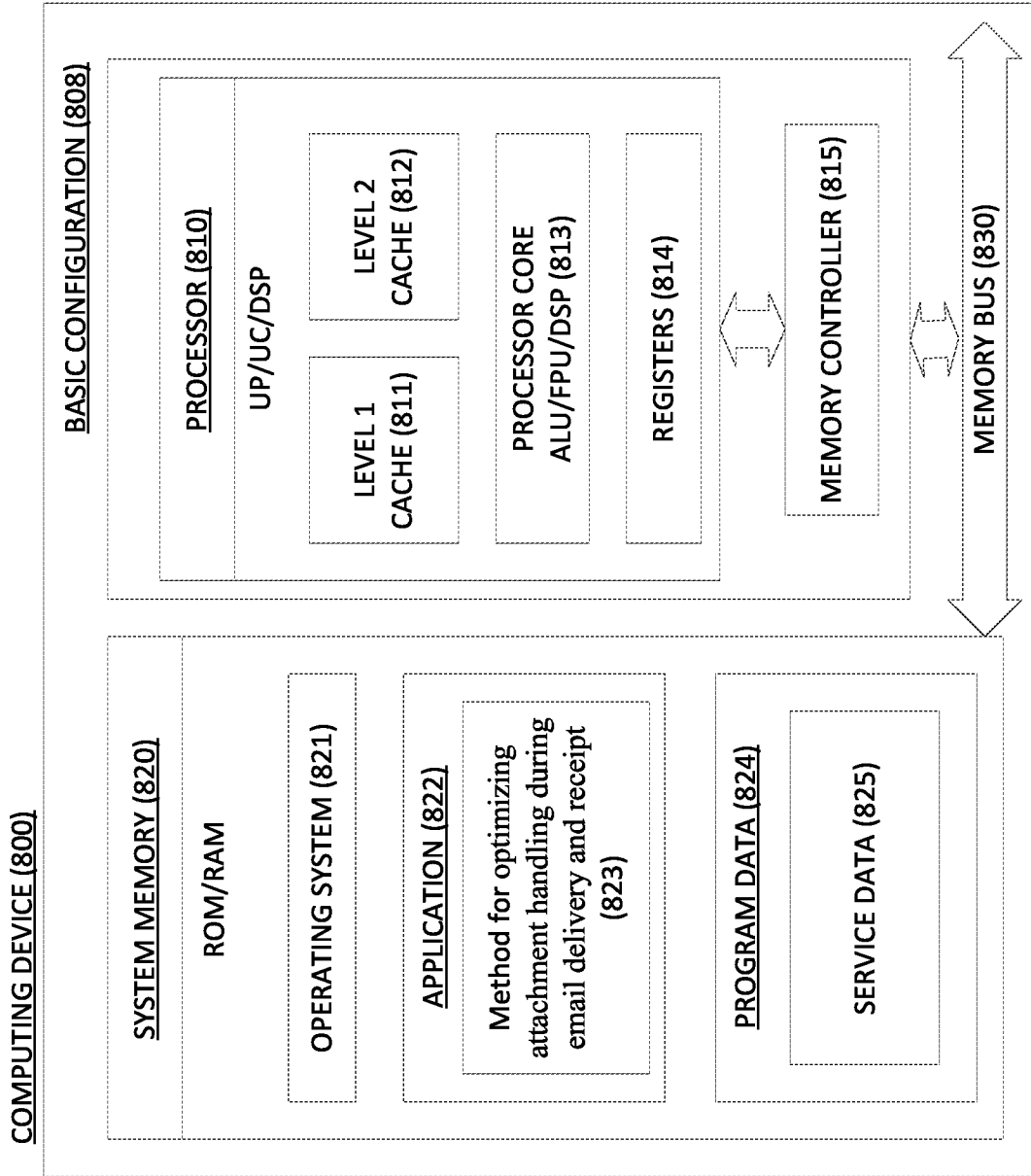


FIG. 8

A. CLASSIFICATION OF SUBJECT MATTER**G06Q 50/32(2012.01)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)

G06Q 50/32; H04L 12/58; H04N 1/00; G06F 13/00

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & keywords: email delivery, attachment content, attachment reference, SMTP

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 2009-188805 A (FUJITSU LTD.) 20 August 2009 See abstract, paragraphs [0006], [0022], [0031], [0035], [0037], [0042]-[0043], [0045], claim 1 and figures 2, 8.	1-4, 7-14, 17-20
Y		5-6, 15-16
Y	KR 10-1020981 B1 (TRIPOD INC.) 09 March 2011 See abstract, paragraphs [0027], [0032]-[0035] and claim 1.	5-6, 15-16
A		1-4, 7-14, 17-20
A	US 2007-0263259 A1 (YOSHIMURA, SHIN) 15 November 2007 See abstract, paragraphs [0035]-[0038], claim 1 and figures 1-3.	1-20
A	JP 2008-065742 A (NIPPON YUNISHISU K.K.) 21 March 2008 See abstract, paragraphs [0013]-[0019], claims 1, 5-6 and figure 2.	1-20
A	JP 2003-178011 A (CASIO COMPUTER CO., LTD.) 27 June 2003 See abstract, paragraphs [0051]-[0055], claims 1-4 and figure 9.	1-20

 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

"E" earlier application or patent but published on or after the international filing date

"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)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search

13 February 2014 (13.02.2014)

Date of mailing of the international search report

14 February 2014 (14.02.2014)

Name and mailing address of the ISA/KR

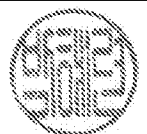
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,
302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

PARK, Hye Lyun

Telephone No. +82-42-481-3463



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/066543

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
JP 2009-188805 A	20/08/2009	None	
KR 10-1020981 B1	09/03/2011	None	
US 2007-0263259 A1	15/11/2007	CN 101036130 A EP 1811390 A1 EP 1811390 A4 IL 181613 D0 JP 2006-119738 A JP 4789100 B2 KR 10-2007-0067681 A WO 2006-043495 A1	12/09/2007 25/07/2007 02/06/2010 04/07/2007 11/05/2006 05/10/2011 28/06/2007 27/04/2006
JP 2008-065742 A	21/03/2008	None	
JP 2003-178011 A	27/06/2003	None	