



- (51) **International Patent Classification:**
H04L 12/745 (2013.01) *H04L 12/743* (2013.01)
- (21) **International Application Number:**
PCT/US2015/040934
- (22) **International Filing Date:**
17 July 2015 (17.07.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventor:** WICKERAAD, John A.; 8000 Foothills Blvd., Roseville, California 95747 (US).
- (74) **Agents:** PINKSTON, Caroline et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (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, 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.

- (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, ST, 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).

Declarations under Rule 4.17:

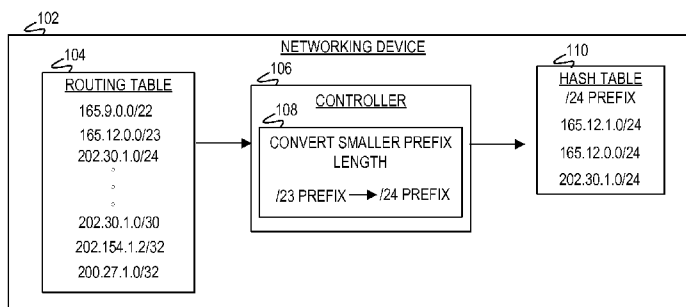
- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

Published:

- with international search report (Art. 21(3))

(54) **Title:** COMBINING PREFIX LENGTHS INTO A HASH TABLE

FIG. 1



(57) **Abstract:** Examples herein disclose a smaller prefix length and a greater prefix length which are identified from a routing table of various prefix lengths. The smaller prefix length is converted into the greater prefix length. The converted prefix length and the greater prefix length are combined into a hash table.

COMBINING PREFIX LENGTHS INTO A HASH TABLE

BACKGROUND

[0001] A routing table is a data table on a networking device which lists the routes and appropriate destinations for a data packet. The networking device uses the routing table to perform an address prefix lookup operation to determine the next appropriate destination.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] In the accompanying drawings, like numerals refer to like components or blocks. The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram of an example networking device to build a hash table by converting a smaller prefix length from a routing table into a greater prefix length in accordance with the present disclosure;

[0004] FIG. 2A is an example networking device to build multiple hash tables by converting smaller prefix lengths identified from a routing table into greater prefix lengths as in accordance with the present disclosure;

[0005] FIG. 2B is a data table of an example index of bitmap values to convert a smaller prefix length into a greater prefix length in accordance with the present disclosure;

[0006] FIG. 3 is a flowchart of an example method to combine a converted smaller prefix length and a greater prefix length into a hash table in accordance with the present disclosure;

[0007] FIG. 4 is a flowchart of an example method to build a hash table for storage in a random access memory (RAM) as in accordance with the present disclosure;

[0008] FIG. 5 is a block diagram of an example computing device with a processing resource to execute instructions in a machine-readable storage medium building a hash table in accordance with the present disclosure; and

[0009] FIG. 6 is a block diagram of an example computing device with a processing resource to execute instructions in a machine-readable storage medium for building a hash table in accordance with the present disclosure.

DETAILED DESCRIPTION

[0010] The growth of the Internet has led to an increased demand for a networking device to efficiently perform an Internet Protocol (IP) address prefix lookup. The networking device references a routing table to decide the next hop for each data packet. As such, the routing table includes various prefix lengths corresponding to different destination subnets. These routing tables have grown in size to accommodate the various prefix lengths and routing destinations.

[0011] The routing table such as a best match prefix (BMP), also known as longest prefix match (LPM) routing table may include few address entries at particular prefix lengths. A hash table may be implemented to dedicate to each prefix length; however this may not make good use of resources. A ternary content addressable memory (TCAM) can be used to hold the prefix lengths of few entries; however placing the prefix lengths in the TCAM may take much power and memory. For example, for each cell of data the TCAM would store the full prefix including the full routing address and masking bits to mask the proper prefix length for lookup operations. Using the additional masking bits may increase the additional amount of memory for the TCAM to store an entry. Additionally, the search operation performed within the TCAM results in increased power as each entry is compared to locate the appropriate entry.

[0012] To address the above-referenced issues, some examples provide a mechanism to optimize a routing table and/or hash table. The examples break up a large routing table into smaller hash tables of varying lengths. The examples reference the routing table to identify a smaller prefix length and a greater prefix length. The smaller prefix length is converted into a length of bits equivalent to the greater prefix length. Converting the smaller prefix length allows the combination between the converted prefix length and the greater prefix length into a single hash table of a unified prefix length. Creating the single hash table of unified prefix length produces fewer hash tables used to implement the routing table and reduces the amount of prefix lengths in the routing table. This allows the networking device to perform more efficient lookup operations.

[0013] In another example, the smaller prefix length is converted by extending the smaller prefix length into the number of bits equivalent to the greater prefix length. Extending the smaller prefix length enables the smaller prefix to fit the greater fix length to provide the unified hash table.

[0014] The unified hash table may be placed in a random access memory (RAM) as opposed to the TCAM which would takes more space for the additional functional bits per cell. The

unified hash table may be placed in a random access memory (RAM) as opposed to the TCAM which would takes more space for the additional functional bits per cell. For example, the TCAM is a type of memory which consists of a bit comparator and two memory elements, one memory element for storing data and the other memory element for storing the compare mask. Storing the hash table in the TCAM increase the memory space, thus placing the hash table in RAM saves power and memory as RAM uses less power than the TCAM.

[0015] Thus some of the examples disclosed herein optimizes a number of address entries in a hash table. The examples break up a large routing table which takes much power and memory into smaller hash tables of specific lengths.

[0016] FIG. 1 is a block diagram of an example networking device 102 to build a hash table 110 of unified prefix length. The networking device 102 includes a controller 106 to use a routing table 104 for identifying a smaller prefix length and a greater prefix length. The controller 106 converts the smaller prefix length at module 108 to build the hash table 110. FIG. 1 illustrates a networking system in which networked computing devices, such as the networking device 102 may exchange data in the form of data packets. These networked computing devices establish data connections in the form on networking links to route and/or forward the data packets. As such, implementations of the networking system include, by way of example, a telecommunications network, Internet, Ethernet, wide area network (WAN), local area network (LAN), optic cable network, virtual network or other type of networking system to route data packets, accordingly.

[0017] The networking device 102 is a networking component that receives incoming data packets and performs a lookup operations to route the incoming data packets to the appropriate destinations. Implementations of the networking device 102 include, by way of example, a networking switch, router, virtual networking component, or other type of networking device 102 which routes data packets to the appropriate destinations.

[0018] The routing table 104 is a data table obtained by the networking device 102 that specifies a route to particular network destinations. As such, the networking device 102 references the routing table to perform an Internet Protocol (IP) address lookup to determine the destination address of an incoming data packet. The routing table 104 includes various prefix lengths (e.g., /22, /23, /24, /30, /32). The various prefix lengths correspond to each destination subnet so the networking device 102 may determine the next hop for the data packets. The prefix lengths

represent the number of bits set in a subnet mask. For example, if the subnet mask is 255.255.255.0, there are 24 1's in the binary version, so the prefix length is 24 bits.

[0019] Referencing the routing table 104, the controller 106 identifies the smaller prefix length and the greater prefix length. The smaller prefix length includes the number of bits less than the greater prefix length. The greater prefix length provides the guidance on the unified prefix length of the hash table 110. The smaller prefix length and the greater prefix length may be identified by determining which prefix lengths are adjacent to each other in the routing table 104. For example, a prefix length of /23 is adjacent to the prefix length of /24 within the routing table 104.

[0020] The controller 106 converts the smaller prefix length into a number of bits equivalent to the greater prefix length at module 108. For example as illustrated in FIG. 1, the /23 prefix length may be converted into a /24 prefix length by extending the prefix lengths by a number of bits to achieve an equivalent length. Converting the smaller prefix length enables the greater prefix length and the converted smaller prefix length to be combined into the hash table 110 to provide the unified prefix length. Implementations of converting the smaller prefix length into the greater prefix length may be explained in detail in the next figure. The module 108 may include, by way of example, instructions (e.g., stored on a machine-readable medium) that, when executed (e.g., by the controller 106) implement the functionality of module 108. Alternatively, or in addition, the module 108 may include electronic circuitry (i.e., hardware) that implements the functionality of module 108. The controller 106 may include, by way of example, a microcontroller, integrated circuit, processing device, semiconductor, circuit, or other type of hardware component for converting the smaller prefix length into the greater prefix length.

[0021] The hash table 110 is a data structure in which the controller may perform the lookup operation to determine an IP address of where to route the incoming packet(s). The hash table 110 includes prefix lengths and the corresponding addresses from the converted smaller prefix length and the identified greater prefix length. Combining both the converted smaller prefix length and the identified greater prefix length provides a unified prefix length. For example, the hash table 110 includes the unified count of /24 prefix length. As such, the /23 prefix length may be converted to the /24 prefix length for placement in the hash table 110.

[0022] FIGS. 2A-2B illustrate an example networking device and a prefix data representation of a converted smaller prefix length. Not all of the depicted components may be required, however, implementations may include additional components not shown in the figures.

Variations in the arrangement and type of the components may be made without departing from the scope of the claims as set forth herein.

[0023] FIG. 2A is an example networking device 202 to build multiple hash tables 210 and 212. The hash tables 210 and 212 are each of a different unified prefix length. For example, the hash table 210 includes the /24 unified prefix length while the hash table 212 includes the /32 unified prefix length. The hash tables 210 and 212 are created by a controller 206 identifying smaller prefix lengths and greater prefix lengths from a routing table 204. The controller 206 converts the smaller prefix lengths at module 208 into an equivalent number of bits as the greater prefix lengths. Upon converting the smaller prefix lengths, the controller 206 combines the converted smaller prefix lengths with the identified greater prefix lengths to create the respective hash tables 210 and 212. These hash tables 210 and 212 are placed in RAM 214.

[0024] The controller 206 converts the smaller prefix lengths into a length of bits equivalent to the identified greater prefix lengths. For example, the /22 and /23 are converted into the /24 prefix length, while the /30 prefix length is converted into the /32 prefix length. The smaller prefix length(s) are identified as being those prefix lengths smaller (e.g., smaller number of set bits in an address) than the greater prefix lengths. In one implementation, the greater prefix lengths are initially identified, while the smaller prefix lengths are those lengths adjacent to the greater prefix lengths. For example, the greater prefix length in the routing table 104 includes the /24 prefix length while the smaller prefix lengths include the /22 and /23 prefix lengths. In FIG. 2A, the controller 206 converts the /22 and /23 prefix lengths into the /24 prefix length while the controller 206 also converts the /30 prefix length into the /32 prefix length. Converting the smaller prefix lengths includes adding the number of bits to make the length of the smaller prefix adjacent to the length of the greater prefix. This is illustrated in FIG. 2A.

[0025] The hash table 210 includes the converted smaller prefix lengths 216 and 218 and the identified greater prefix length 220. In the example of hash table 210, the identified greater prefix length includes a /24 count, thus the smaller prefix lengths of /22 and /23 are converted into a number of bits equivalent to the /24 count. These lengths and corresponding address entries are combined into the hash table 210 to create the unified prefix length of /24. The converted prefix length 216 represents the situation of where the original /22 smaller prefix length is converted into the /24 greater prefix length. The converted prefix length 218 represents the original /23 smaller prefix length as

converted into the /24 greater prefix length. The greater prefix length 220 is not converted from the routing table 204 prior to placement in the hash table 210.

[0026] The hash table 212 includes the converted smaller prefix length 222 and the identified greater prefix length 226. In the example of hash table 212, the identified greater prefix length includes the /32 count, thus the smaller prefix lengths as from the routing table 204 include the /30 prefix length. This smaller prefix length (/30) is converted into the number of bits equivalent to the /32 count. These converted lengths and corresponding address entries are combined into the hash table 212 to create the unified prefix length of /32. The converted prefix length 222 represents the original /30 smaller prefix length which is converted into the /32 greater prefix length. The greater prefix length 226 is not converted from the routing table 204 prior to placement into the hash table 212.

[0027] FIG. 2B is database representing an example bitmap index of values corresponding to a smaller prefix length 228. The smaller prefix length 228 is converted into a greater prefix length by extending the index value by a bit to achieve a number of bits equivalent to the greater prefix length. The equivalent number of bits to the greater prefix length is achieved by converting the smaller prefix lengths 230 and 232. The conversion of the smaller prefix lengths 230 and 232 are produced by extending the bits with a binary value representing two conditions. For example, the smaller prefix length of /23 is converted into the greater prefix length of /24. The networking device 202 adds the bit of 0 or 1 in the index place value of /24. The converted smaller prefix lengths 230 and 232 are added with the greater prefix length of /24 to build the unified hash table of /24 prefix length. Increasing the number of bits also increases the number of states or combinations of the converted prefix length. In this example, by adding the single bit of 0 or 1, represents two different states of the converted prefix length. Thus, adding two bits of 00, 01, 10, and 01 represents the four different combinations of the converted smaller prefix length.

[0028] Referring now to FIGS. 3 and 4, flowcharts are illustrated in accordance with various examples of the present disclosure. The flowcharts represent processes that may be utilized in conjunction with various systems and devices as discussed with reference to the preceding figures. While illustrated in a particular order, the flowcharts are not intended to be so limited. Rather, it is expressly contemplated that various processes may occur in different orders and/or simultaneously with other processes than those illustrated.

[0029] FIG. 3 illustrates a flowchart of an example method to optimize a hash table by combining a converted smaller prefix length and a greater prefix length together. A networking device receives a routing table of various prefix lengths. Using this routing table, the networking device identifies a smaller prefix length and a greater prefix length. The smaller prefix length is converted into an equivalent amount of bits as the greater prefix length. The converted smaller prefix length and the identified prefix length are combined into the hash table. Converting the smaller prefix length creates uniformity in which the smaller prefix length has the equivalent amount of bits to the identified greater prefix length. In discussing FIG. 3, references may be made to the components in FIGS. 1-2B to provide contextual examples. In one implementation, the networking device 102 executes operations 302-306 build the hash table. Although FIG. 3 is described as implemented by the networking device 102, it may be executed on other suitable components. For example, FIG. 3 may be implemented in the form of executable instructions on a machine-readable storage medium 504 and 604 as in FIGS. 5-6.

[0030] At operation 302, the networking device identifies the smaller prefix length and the greater prefix length from the routing table. The routing table is used by the networking device to perform lookup function in which to determine a location (e.g., next hop information) for routing incoming data packets. The routing table may be obtained from a nearby server listing the various address prefix lengths. The various prefix lengths represent a number of bits to designate an Internet Protocol (IP) address. The IP address is a 32 bit value; as such the routing table entry with a prefix length represents a set of IP address that matches the upper portion of the prefix length. As such, the routing table may include an abundant amount of specific prefix lengths. Accordingly, a lookup operation may become more efficient if particular prefix addresses were separated out from the routing table into the hash table. Identifying the smaller prefix length and the greater prefix length from the routing table enables the networking device to separate these address lengths to build the hash table. The networking device may initially identify the greater prefix length to determine which prefix lengths may be less than the greater prefix length for identification of the smaller prefix length.

[0031] The smaller prefix length represents a smaller number of bits in an address compared to the greater prefix length. In one implementation, the smaller prefix length is a lesser number of bits than the greater prefix length. For example, the smaller prefix length of <16 (less than 16 bits) may be combined into the greater prefix length of $/16$. In other implementations, the smaller prefix length

may be an adjacent prefix length to the greater prefix length in the routing table. For example, a smaller prefix length of /15 may be combined with a greater prefix length of /16.

[0032] At operation 304, the networking device converts the smaller prefix length into the greater prefix length. The networking device converts the smaller prefix length into a length of bits equivalent to the greater prefix length. Converting the prefix length, provides a unified prefix length to combine both identified prefix lengths together into the hash table. Additionally, this conversion allows entries of different prefix lengths to be combined into the same hash table.

[0033] At operation 306, the networking device combines both the converted smaller prefix length and the greater prefix length identified at operation 302 into the hash table. Combining the converted smaller prefix length and the greater prefix length includes combining the corresponding address entries from the routing table into the hash table.

[0034] FIG. 4 is a flowchart of an example method, executable by a networking device, to combine a converted smaller prefix length and a greater prefix length into a unified prefix length hash table. The networking device converts the smaller prefix length by extending the smaller prefix length by a number of bits to produce an equivalent prefix length to the greater prefix length. Upon converting the smaller prefix length, the networking device identifies an overlapping address entry for removal. Upon removing the overlapping address entry, the networking device places the hash table in RAM. In discussing FIG. 4, references may be made to the components in FIGS. 1-2B to provide contextual examples. In one implementation, the networking device 102 executes operations 402-420 to build the hash table. Although FIG. 4 is described as implemented by the networking device 102, it may be executed on other suitable components. For example, FIG. 4 may be implemented in the form of executable instructions on a machine-readable storage medium 504 and 604 as in FIGS. 5-6.

[0035] At operation 402, the networking device identifies the smaller prefix length and the greater prefix length from the routing table. As such, the networking device may proceed to operations 404-408 to identify the prefix lengths.

[0036] At operations 404-408, the networking device identifies adjacent prefix lengths from the routing table. The adjacent prefix lengths include the smaller prefix length and the greater prefix length. The greater prefix length represents the unified prefix length in which to create the hash table. The smaller prefix length is the length(s) which may be converted to obtain the unified prefix length.

[0037] At operation 410, the networking device converts the smaller prefix length into the greater prefix length. The networking device may extend the smaller prefix length by a number of bits to produce a bit length equivalent to the greater prefix length as at operation 412.

[0038] At operation 412, the networking device extends the smaller prefix length by the bit(s). The networking device adds the bit(s) to extend the smaller prefix length to fit the hash table of the greater prefix length. The extender bit(s) represent the different conditions of the smaller prefix length. As such, each condition of the smaller prefix length is captured to include as part of the hash table. For example, if a single bit is added (0 or 1) this represents two different conditions. Thus, both of these conditions are represented for combination in the hash table. In another example, if two bits are added (00, 01, 10, or 11), this represents four different conditions or combinations of the extender bits. As such, each of the four conditions are added to the smaller prefix length to produce four different combinations of the smaller prefix length.

[0039] At operation 414, the networking device combines the address entries corresponding to the converted prefix length and the greater prefix length to build the hash table. Combining the address entries, provides the hash table of the unified prefix length.

[0040] At operations 416-418, the networking device identifies the overlapping address between the converted smaller prefix length and the greater prefix length for removal. Upon extending the smaller prefix length by the number of bits, one of the extended smaller prefix length addresses may overlap with one of the greater prefix length addresses. As such, including multiple address may make the routing destinations incorrect. In this implementation, there may be a correct overlapping address entry to keep and another overlapping address entry to remove. The duplicate address entry after combining the prefix lengths are resolved by keeping the address entry that was originally the longer prefix. The original longer prefix is that address entry which was identified in connection with the greater prefix length from the routing table. Thus, the networking device compares the extended smaller prefix length addresses to the greater prefix length address to identify those address entries which may be repetitive. As such, the networking device proceeds to remove these repetitive or overlapping address from the hash table.

[0041] At operation 420, the unified prefix length hash table is stored in RAM. Storing the hash table in RAM enables the networking device to readily perform the lookup operations on the hash table.

[0042] FIG. 5 is a block diagram of computing device 500 with a processing resource 502 to execute instructions 506-514 within a machine-readable storage medium 504. Specifically, the computing device 500 with the processing resource 502 is to convert a smaller prefix length for combining with a greater prefix length into a hash table. Although the computing device 500 includes processing resource 502 and machine-readable storage medium 504, it may also include other components that would be suitable to one skilled in the art. For example, the computing device 500 may include the controller 106 as in FIG. 1. The computing device 500 is an electronic device with the processing resource 502 capable of executing instructions 506-514, and as such embodiments of the computing device 500 include a mobile device, client device, personal computer, desktop computer, laptop, tablet, video game console, or other type of electronic device capable of executing instructions 506-514. The instructions 506-514 may be implemented as methods, functions, operations, and other processes implemented as machine-readable instructions stored on the storage medium 504, which may be non-transitory, such as hardware storage devices (e.g., random access memory (RAM), read only memory (ROM), erasable programmable ROM, electrically erasable ROM, hard drives, and flash memory).

[0043] The processing resource 502 may fetch, decode, and execute instructions 506-514 to combine the converted smaller prefix length and the greater prefix length into the hash table for optimization. Specifically, the processing resource 502 executes instructions 506-514 to: identify the smaller prefix length and the greater prefix length from a routing table; convert the smaller prefix length into a number of bits equivalent to the greater prefix length; and combine the converted smaller prefix length and the identified greater prefix length into the hash table together.

[0044] The machine-readable storage medium 504 includes instructions 506-514 for the processing resource 502 to fetch, decode, and execute. In another embodiment, the machine-readable storage medium 504 may be an electronic, magnetic, optical, memory, storage, flash-drive, or other physical device that contains or stores executable instructions. Thus, the machine-readable storage medium 504 may include, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a memory cache, network storage, a Compact Disc Read Only Memory (CDROM) and the like. As such, the machine-readable storage medium 504 may include an application and/or firmware which can be utilized independently and/or in conjunction with the processing resource 502 to fetch, decode, and/or execute instructions of the machine-readable storage medium 504. The application and/or firmware may be stored on the

machine-readable storage medium 504 and/or stored on another location of the computing device 500.

[0045] FIG. 6 is a block diagram of computing device 600 with a processing resource 602 to execute instructions 606-622 within a machine-readable storage medium 604. Specifically, the computing device 600 with the processing resource 602 extends a smaller prefix length and combines the extended smaller prefix length with a greater prefix length into a hash table. The extended smaller prefix length and the greater prefix length are processed to remove an overlapping entry address. Upon removal of the overlapping entry address, the hash table is stored in RAM. Although the computing device 600 includes processing resource 602 and machine-readable storage medium 604, it may also include other components that would be suitable to one skilled in the art. For example, the computing device 600 may include the controller 106 as in FIG. 1. The computing device 600 is an electronic device with the processing resource 602 capable of executing instructions 606-622, and as such embodiments of the computing device 600 include a mobile device, client device, personal computer, desktop computer, laptop, tablet, video game console, or other type of electronic device capable of executing instructions 606-622. The instructions 606-622 may be implemented as methods, functions, operations, and other processes implemented as machine-readable instructions stored on the storage medium 604, which may be non-transitory, such as hardware storage devices (e.g., random access memory (RAM), read only memory (ROM), erasable programmable ROM, electrically erasable ROM, hard drives, and flash memory).

[0046] The processing resource 602 may fetch, decode, and execute instructions 606-622 to convert the smaller prefix length through extension of a bit. The converted smaller prefix length is compared against the greater prefix length to identify the overlapping address entry for removal. Upon removing the overlapping address entry, the converted prefix length and the greater prefix length are combined into the hash table for storage within the RAM. Specifically, the processing resource 602 executes instructions 606-622 to: identify a smaller prefix length and a greater prefix length from the routing table of various prefix lengths; the identified smaller prefix length is converted by extending a bit to achieve an equivalent number of bits to the identified greater prefix length; the converted smaller prefix length and the greater prefix length are compared to each other to identify an overlapping address entry; upon identification of the overlapping address entry, the overlapping address is removed; the converted smaller prefix length and the greater prefix length are combined

into the hash table sans the overlapping address entry; upon the combination of the hash table, the hash table is placed in the RAM for storage.

[0047] The machine-readable storage medium 604 includes instructions 606-622 for the processing resource 602 to fetch, decode, and execute. In another embodiment, the machine-readable storage medium 604 may be an electronic, magnetic, optical, memory, storage, flash-drive, or other physical device that contains or stores executable instructions. Thus, the machine-readable storage medium 604 may include, for example, Random Access Memory (RAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), a storage drive, a memory cache, network storage, a Compact Disc Read Only Memory (CDROM) and the like. As such, the machine-readable storage medium 604 may include an application and/or firmware which can be utilized independently and/or in conjunction with the processing resource 602 to fetch, decode, and/or execute instructions of the machine-readable storage medium 604. The application and/or firmware may be stored on the machine-readable storage medium 604 and/or stored on another location of the computing device 600.

[0048] Although certain embodiments have been illustrated and described herein, it will be greatly appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent embodiments or implementations calculated to achieve the same purposes may be substituted for the embodiments shown and described without departing from the scope of this disclosure. Those with skill in the art will readily appreciate that embodiments may be implemented in a variety of ways. This application is intended to cover adaptations or variations of the embodiments discussed herein. Therefore, it is manifestly intended that embodiments be limited only by the claims and equivalents thereof.

CLAIMS

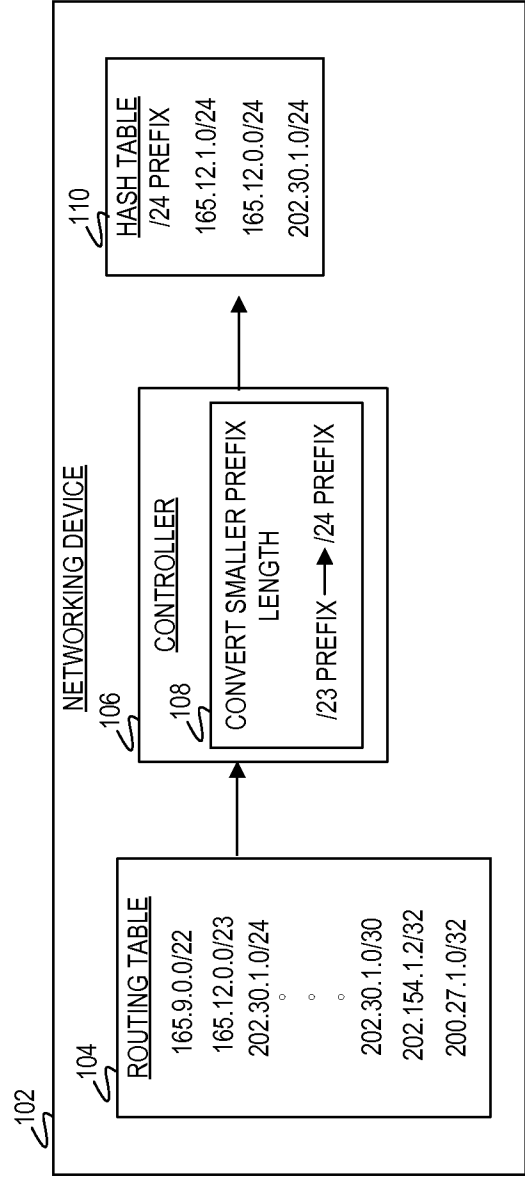
I claim:

1. A non-transitory machine-readable storage medium comprising instructions that when executed by a processing resource cause a networking device to:
 - identify a smaller prefix length and a greater prefix length from a routing table of various prefix lengths;
 - convert the smaller prefix length into the greater prefix length; and
 - combine the converted smaller prefix length and the greater prefix length into a hash table.
2. The non-transitory machine-readable storage medium of claim 1 wherein to convert the smaller prefix length into the greater prefix length comprises instructions that when executed by the processing resource causes the networking device to:
 - extend the smaller prefix length by a number of bits to produce a length of bits equivalent to the greater prefix length.
3. The non-transitory machine-readable storage medium of claim 1 wherein to combine the converted smaller prefix length and the greater prefix length comprises instructions that when executed by the processing resource causes the networking device to:
 - identify an overlapping prefix address between the converted smaller prefix length and the greater prefix length; and
 - remove the overlapping prefix address from the hash table.
4. The non-transitory machine-readable storage medium of claim 1 wherein the smaller prefix length and the greater prefix length include lengths that are adjacent to each other.
5. The non-transitory machine-readable storage medium of claim 1 comprising instructions that when executed by the processing resource causes the networking device to:
 - store the hash table in a random access memory (RAM) of the networking device.

6. The non-transitory machine-readable storage medium of claim 1 wherein to convert the smaller prefix length into the greater prefix length comprises instructions that when executed by the processing resource causes the networking device to:
 - extend the smaller prefix length by a bit to produce the greater prefix length.
7. A method, executable by a networking device, the method comprising:
 - identifying a smaller prefix length and a greater prefix length from a routing table of various prefix lengths;
 - converting the smaller prefix length into the greater prefix length; and
 - combining the converted smaller prefix length and the greater prefix length into a hash table.
8. The method of claim 7 wherein converting the smaller prefix length into the greater prefix length comprises:
 - extending the smaller prefix length by a number of bits to produce a length equivalent to the greater prefix length.
9. The method of claim 7 wherein identifying the smaller prefix length and the greater prefix from the routing table of various prefix lengths comprises:
 - obtaining the routing table;
 - identifying adjacent prefix lengths from the routing table; and
 - determining which of the adjacent prefix lengths is the smaller prefix length and the greater prefix length.
10. The method of claim 7 wherein combining the converted smaller prefix length and the greater prefix length into the hash table comprises:
 - identifying an overlapping prefix address between the converted smaller prefix length and the greater prefix length; and
 - removing the overlapping prefix address prior to combination into the hash table.

11. The method of claim 7 comprising:
storing the hash table in a random access memory (RAM) of the networking device.
12. A networking device comprising:
a routing table of various prefix lengths;
a networking controller to:
 convert a smaller prefix length identified from the routing table into a greater prefix length; and
 build a hash table by combining the converted smaller prefix length and the greater prefix length from the routing table.
13. The networking device of claim 12 comprising:
the hash table comprising the converted smaller prefix length and the greater prefix length;
and
a random access memory (RAM) to store the hash table.
14. The networking device of claim 12 wherein to convert the smaller prefix length into the greater prefix length the networking controller is to:
extend the smaller prefix length by a bit to produce the greater prefix length.
15. The networking device of claim 12 wherein to build the hash table the networking controller is to:
combine the converted smaller prefix length and the greater prefix length from the routing table.

FIG. 1



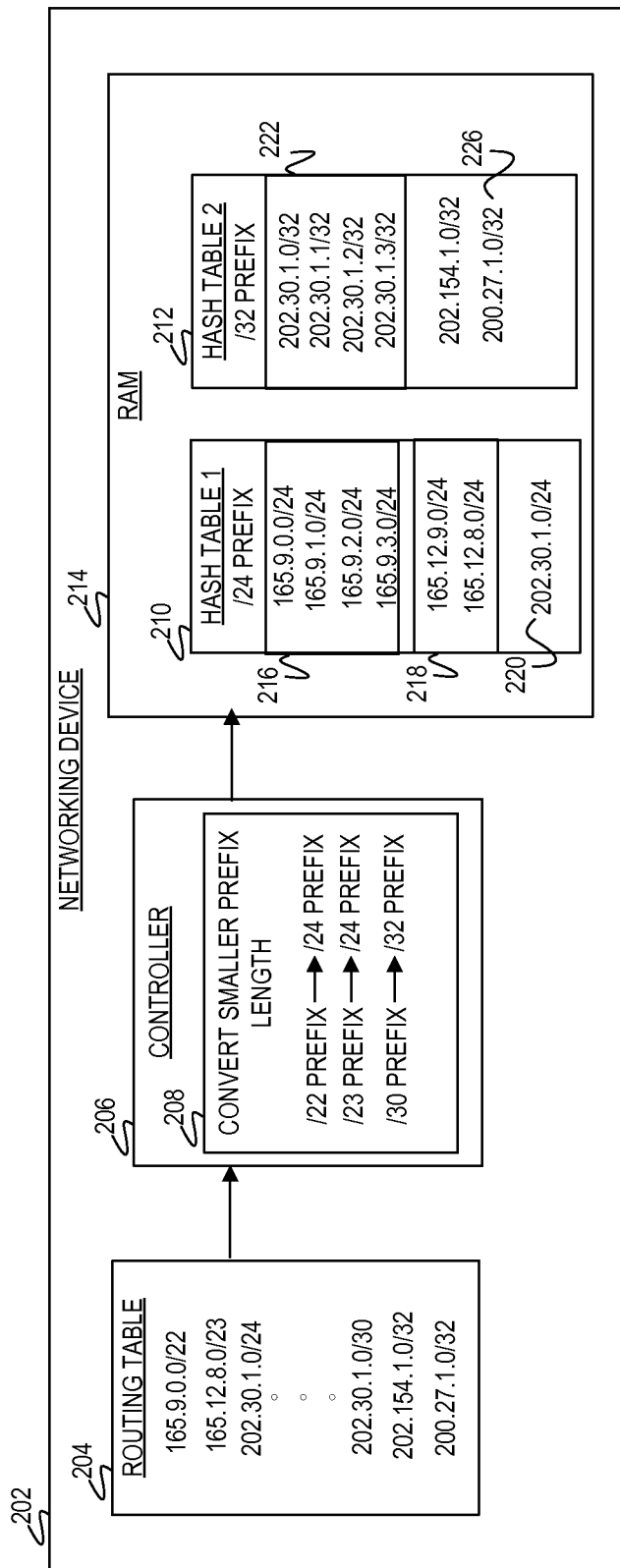


FIG. 2A

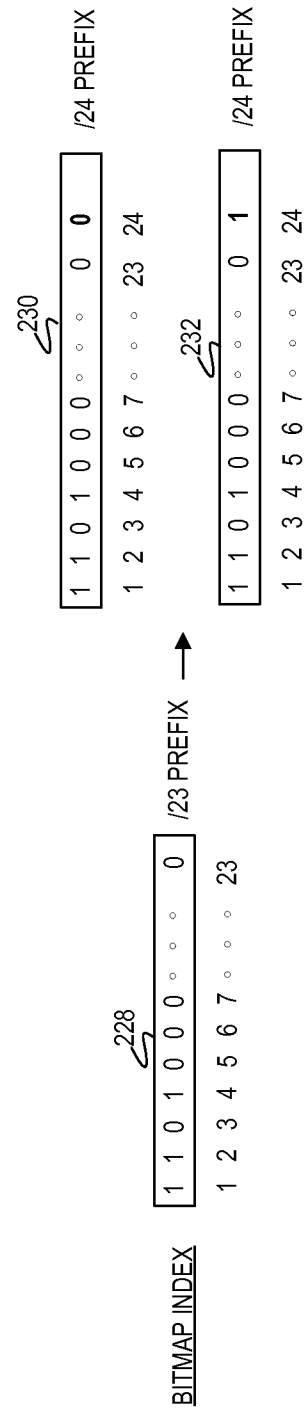
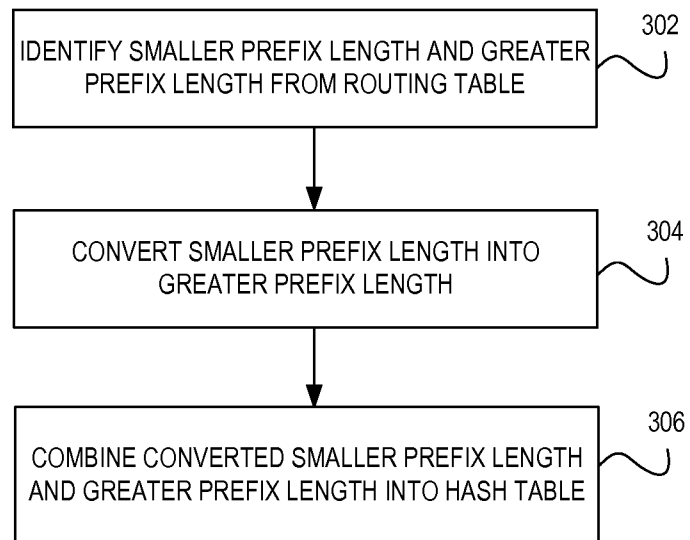


FIG. 2B

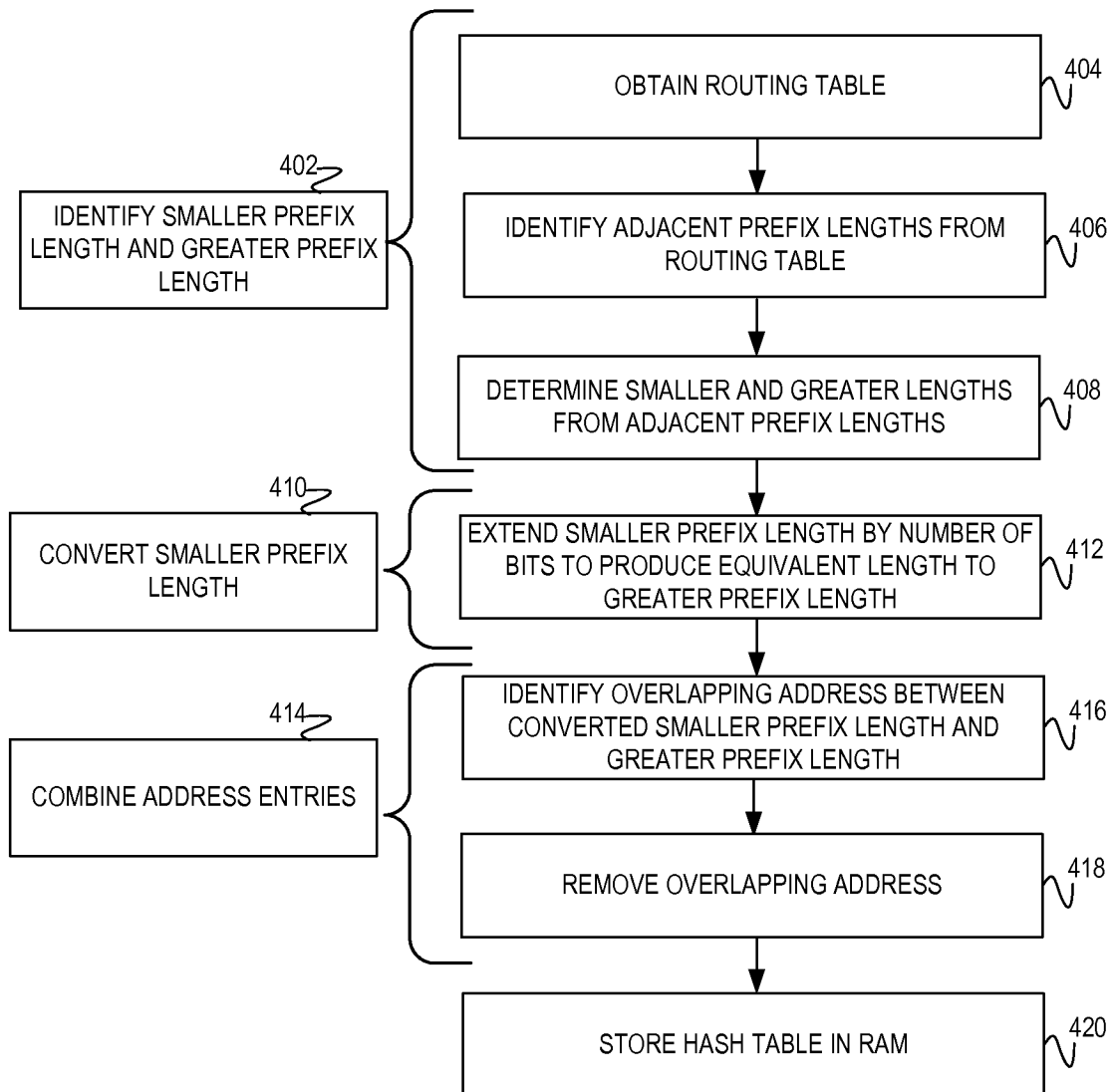
3/6

FIG. 3



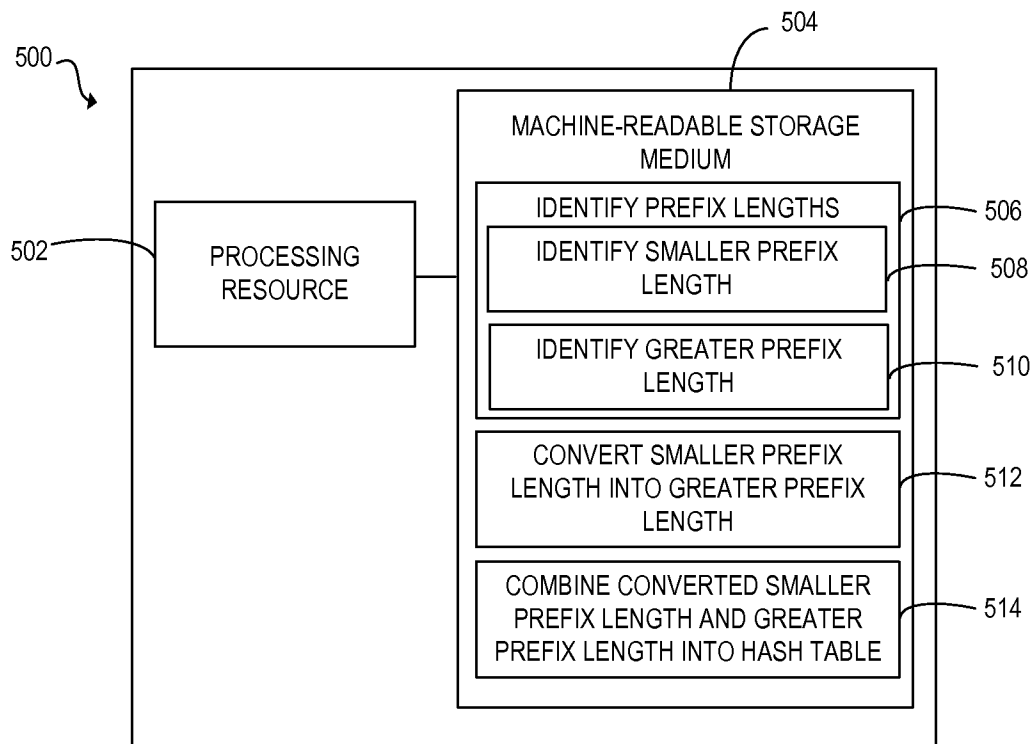
4/6

FIG. 4



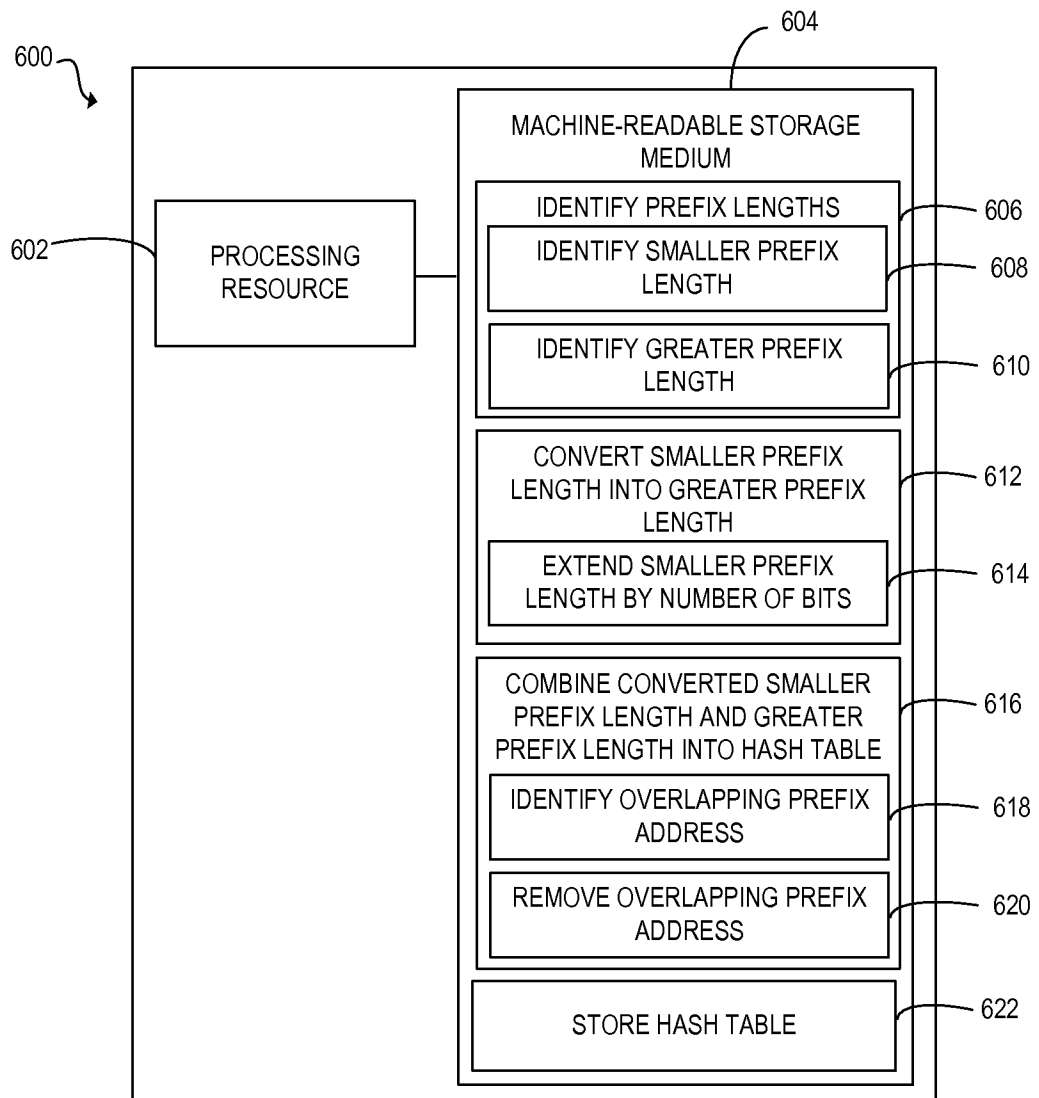
5/6

FIG. 5



6/6

FIG. 6



A. CLASSIFICATION OF SUBJECT MATTER**H04L 12/745(2013.01)i, H04L 12/743(2013.01)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 12/745; G06F 17/30; G06F 7/00; G06F 12/00; H04W 4/00; H04L 12/28; H04L 12/56; H04L 12/743

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: routing table, prefix length, small, convert, extend, overlap, remove, hash table, RAM

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 7398278 B2 (SRIHARI CADAMBI et al.) 08 July 2008 See column 3, line 12 - column 5, line 30; claim 1; and figures 1-6.	1-15
Y	US 2011-0170531 A1 (CHAN WAH NG et al.) 14 July 2011 See paragraphs [0037]-[0163]; claim 12; and figure 7.	1-15
Y	US 2008-0112413 A1 (FONG PONG) 15 May 2008 See paragraph [0029]; and figure 2.	5, 11, 13
A	US 2007-0136331 A1 (JAHANGIR HASAN et al.) 14 June 2007 See paragraphs [0085]-[0094]; and figures 5-6.	1-15
A	WO 2010-145295 A1 (ZTE CORPORATION) 23 December 2010 See page 3, lines 19-24; and figure 1.	1-15



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

28 March 2016 (28.03.2016)

Date of mailing of the international search report

14 April 2016 (14.04.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

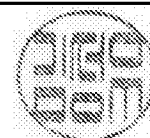
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/040934

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7398278 B2	08/07/2008	JP 2006-246489 A US 2006-0200581 A1	14/09/2006 07/09/2006
US 2011-0170531 A1	14/07/2011	CN 102165839 A EP 2341753 A1 WO 2010-035464 A1	24/08/2011 06/07/2011 01/04/2010
US 2008-0112413 A1	15/05/2008	US 2008-0112412 A1 US 7852851 B2 US 7885268 B2	15/05/2008 14/12/2010 08/02/2011
US 2007-0136331 A1	14/06/2007	US 7653670 B2	26/01/2010
WO 2010-145295 A1	23/12/2010	CN 101692653 A CN 101692653 B	07/04/2010 04/04/2012