



(22) Date de dépôt/Filing Date: 2007/02/23

(41) Mise à la disp. pub./Open to Public Insp.: 2008/08/23

(45) Date de délivrance/Issue Date: 2012/05/15

(51) Cl.Int./Int.Cl. *H04L 12/16* (2006.01),
G06F 17/30 (2006.01), *H04L 7/00* (2006.01)

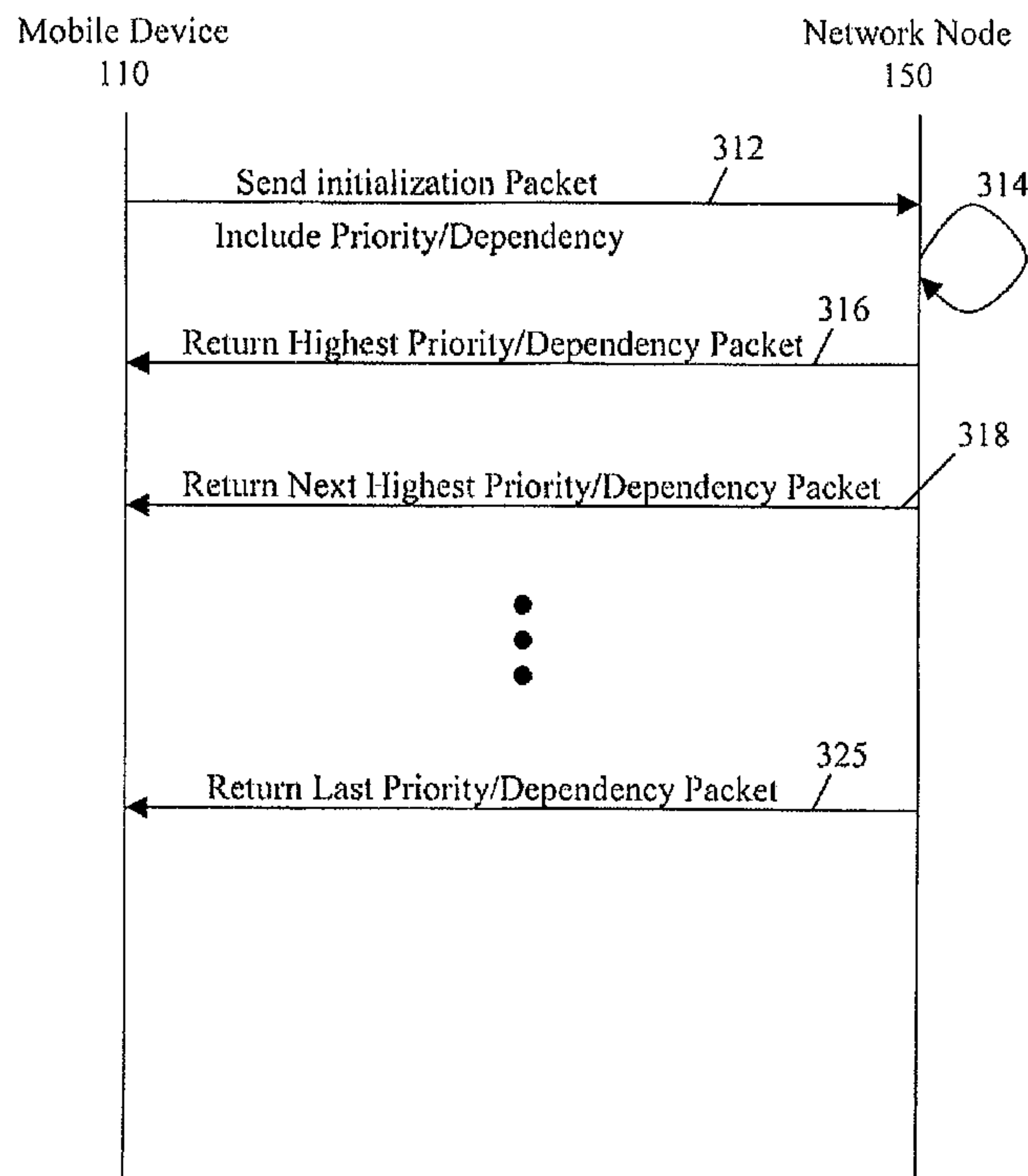
(72) Inventeur/Inventor:
GROUX, BRIAN, CA

(73) Propriétaire/Owner:
RESEARCH IN MOTION LIMITED, CA

(74) Agent: MOFFAT & CO.

(54) Titre : METHODE ET APPAREIL DE COMMANDE ELARGIE VIA UN RESEAU SANS FIL D'ENSEMBLES DE
DONNEES EN UTILISANT DES NIVEAUX DE PRIORITE ET DE DEPENDANCE

(54) Title: METHOD AND APPARATUS FOR EXTENDED ORDERING OVER A WIRELESS NETWORK OF
COLLECTIONS USING PRIORITY AND DEPENDENCY LEVELS



(57) **Abrégé/Abstract:**

A method and system for extended ordering of collections of data for synchronization over a wireless network, each collection having an existing priority level, the method having the steps of: adding, to each collection, a dependency field having a dependency value; exchanging the priority level and dependency value for each collection; and ordering synchronization of the collections based on priority levels and dependency values of the collections.

ABSTRACT

A method and system for extended ordering of collections of data for synchronization over a wireless network, each collection having an existing priority level, the method having the steps of: adding, to each collection, a dependency field having a dependency value; exchanging the priority level and dependency value for each collection; and ordering synchronization of the collections based on priority levels and dependency values of the collections.

METHOD AND APPARATUS FOR EXTENDED ORDERING OVER A WIRELESS NETWORK OF COLLECTIONS USING PRIORITY AND DEPENDENCY LEVELS

FIELD OF THE DISCLOSURE

[0001] The present disclosure relates to the synchronization of data between a mobile device and a network node and, in particular, to the ordering of the synchronization of data between the mobile device and network node.

BACKGROUND

[0002] A mobile device stores a number of databases or collections, hereinafter referred to as collections, that periodically need to be synchronized with a network node. Such a synchronization ensures that the collections stored on the network node are the same as those stored on the mobile device.

[0003] In order to ensure that synchronization occurs in a particular order, prior art solutions assign a priority number to each collection, where the priority number represents the order in which the collection is synchronized. In order to ensure simplicity, a limited number of priority levels are available in these prior solutions. Thus, each collection could be assigned, for example, a priority from 1-10. This priority ordinal could be sent to the network node as part of an initialization packet. The network node could then use the ordinal to order the returning packets to the device, thereby forcing the network node to perform the synchronization in a predetermined collections order.

[0004] The above solution, however, does not accommodate devices with a larger number of collections. Nor does it accommodate devices in which some collections become dependent on each other. By assigning various collections the same priority level, the network node can choose which collection to return first, creating a haphazard ordering system which could cause problems if a collection, which is dependent on another collection, is synchronized out of order.

SUMMARY

[0005] The present disclosure provides for a dependency field to be added to each priority level. In a preferred embodiment, the dependency field would be one byte, thereby allowing each priority level to be subdivided into dependency levels 1-255. However, in other cases where a larger number of collections are present on the mobile device, multiple bytes could be used. By adding a dependency level to a priority field, existing priority level schemes could be used on the mobile device without having to extend the priority scheme to an unmanageable number of priority levels. Further, dependencies could be established to ensure that collections are synchronized in the correct order.

[0006] The present disclosure therefore provides a method for extended ordering of collections of data for synchronization over a wireless network, each collection having an existing priority level, the method comprising the steps of: adding, to each collection, a dependency field having a dependency value; and exchanging, during synchronization, the priority level and dependency value for each collection, wherein synchronization is ordered based on priority levels and dependency values of the collections.

[0007] The present application further provides a method for extended ordering of collections of data for synchronization over a wireless network, each collection having an existing priority level, the method comprising the steps of: adding, to each collection, a dependency field having a dependency value; exchanging the priority level and dependency value for each collection; and ordering synchronization of the collections based on priority levels and dependency values of the collections.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The present application will be better understood with reference to the drawings in which:

Figure 1 is a block diagram illustrating a series of collections having a priority on the mobile device and a series of collections on the network node;

Figure 2 is a block diagram illustrating a series of collections having both priority and dependency on the mobile device and a series of collections on the network node;

Figure 3 is a sequence diagram showing a synchronization procedure between the mobile device and a network node;

Figure 4 illustrates an exemplary network containing both mobile device and a network node; and

Figure 5 shows a block diagram of an exemplary mobile device that can be used in association with the present method.

DETAILED DESCRIPTION

[0009] Reference is now made to the drawings. **Figure 1** illustrates a block diagram showing a mobile device **110** and a network node **150**. As will be appreciated by those skilled in the art, mobile device **110** could be any mobile device having collections that need to be synchronized with a network node. Such devices include, but are not limited to, cellular telephones, personal digital assistance, pagers, laptop computers and portable data devices.

[0010] An exemplary mobile device is illustrated with reference to **Figure 5** below.

[0011] As is seen in **Figure 1**, mobile device **110** includes a series of collections, labeled as a first collection **112**, a second collection **114**, a third collection **116** and collection **n 118**, where “n” is a positive integer. Each of collections **112**, **114**, **116** and **118** includes a priority associated therewith.

[0012] A network node **150** includes a first collection **152**, a second collection **154**, a third collection **156** and collection **n 158**, where n is a positive integer.

[0013] In operation, a first collection **112** needs to synchronize with a first collection **152**, a second collection **114** needs to synchronize with a second collection **154**, a third collection **116** needs to synchronize with a third collection **156** and so on, until collection n **118** synchronizes with collection n **158**. However, the order in which the collections synchronize may depend on various factors and thus the priority of each collection can be set.

[0014] In operation, when mobile device **110** connects to network node **150** through an interface **170**, a synchronization may occur. Interface **170** can be any connection between mobile device **110** and network node **150**, including an over-the-air connection such as a radio frequency cellular network connection or a WiFi connection, a short-range wireless connection such as Bluetooth™ or IrDA, or a physical connection such as a serial connection using a universal serial bus (USB) connection. Other possible connections would be known to those skilled in the art.

[0015] In order for synchronization to occur, when interface **170** is established between mobile device **110** and network node **150**, mobile device **110** sends network node **150** the priorities of each of its collections. Network node **150** can then order commands for the returning packets to mobile device **110**. As will be appreciated, commands include record level commands such as 'add' or 'delete', or could include collection level commands such as a hash function or a command to use a collection.

[0016] In an alternative embodiment, the actual packets may be ordered. This therefore forces the network node to initialize and perform the first synchronization in a predetermined database order.

[0017] As indicated above, the use of a priority by itself is however limiting. Specifically, if a limited number of priorities are available, then with a larger number of collections, two elements will necessarily need to share the same

priority level. Network node **150**, when receiving a series of collections and priority levels for each of those collections, will haphazardly arrange the collections of the same priority level when returning these to the mobile device **110**. As will be appreciated, this can create problems.

[0018] Reference is now made to **Figure 2**. In **Figure 2**, mobile device **110** communicates over interface **170** with network node **150**. In the embodiment of **Figure 2**, mobile device **110** includes a first collection **212**, a second collection **214**, a third collection **216**, and collection **n 218** where “n” is a positive integer. Similarly, network node **150** includes a first collection **252**, a second collection **254**, a third collection **256** and a collection **n 258**, where “n” is a positive integer.

[0019] As in the embodiment of **Figure 1**, in the embodiment of **Figure 2**, the mobile device **110** needs to synchronize first collection **212** with first collection **252**, second collection **214** with second collection **254**, third collection **216** with third collection **256** and so on until synchronizing collection **n 218** with collection **n 258**.

[0020] In the embodiment of **Figure 2**, collections **212**, **214**, **216** and **218** include both a priority and a dependency. The dependency field is added to the priority field in order to ensure that the priority field can stay the same for backwards compatibility purposes. In other words, the priority field remains the same in order that other databases or collections on the device expecting existing priority ordering still work. By subdividing the priority into various dependency levels, the ordering of collections at the same priority level can be accomplished correctly.

[0021] In a preferred embodiment, the dependency is a single byte added to the priority, thereby providing 255 dependency levels.

[0022] Reference is made to **Figure 3**. **Figure 3** shows exemplary communications between a mobile device **110** and a network node **150**. Once a

connection is made between a mobile device **110** and a network node **150**, the mobile device sends an initialization packet **312** to network node **150**. The initialization packet includes the priority and dependency of each collection within the mobile device **110**.

[0023] Network node **150** receives the packet **312** and, at **314**, sorts commands for the collections based on the priority and dependency levels received from mobile device **110**. Thereafter, network node **150** returns the highest priority and dependency packet in message **316**. Subsequently, network node **150** returns the next highest priority and dependency packet in message **318** and continues in this way until it returns the lowest priority and dependency packet in message **325**. In an alternative embodiment, the collections themselves could be sorted

[0024] The above allows for the growth of software to have a number of collections beyond the number of priority levels assigned to the mobile device. Further, it allows certain collections to be dependent upon other collections where the dependent collections can be put at the same priority level as the collections from which they depend and distinguish between the priority levels by assigning a dependency. This allows for synchronization to occur in the correct order.

[0025] The above is further illustrated with an example:

Example 1

[0026] A current mobile device may have 40 or 50 collections on the device that require synchronization. Two of these collections can include an auto-text correction, which allows a user to set an auto-correction for certain typographical errors that the user typically makes, and an options collection, which includes a list of options that the user sets.

[0027] One of the options in the options database is a “locale”. The locale sets the language on the device and the auto-text collection is therefore dependent on the options collection.

[0028] Since the options on the device are important to the device, the options database typically receives the top priority. However, this can create problems. Specifically, each time a locale is changed either by the user or through USB, serial, wireless or other interface restore, the auto-text collection is reset and repopulated with entries for the given locale.

[0029] The change in locale causes the auto-text collection to be reset. However, the auto-text collection has already started to be synchronized based on the initialization packet 312 from **Figure 3**. Thus, the auto-text collection would start synchronizing while a reset was being propagated through to the network. This results in a race condition and could cause loss of data or other problems on the device.

[0030] It is thus desirable to synchronize the auto-text prior to the synchronization of the options database. However, if the options database is already at a top priority, the only solution is to set the auto-text collection also to the top priority. This, however, does not necessarily solve the problem, since network node 150 from **Figure 1** could choose which priority level “one” synchronization to perform first. The selection is haphazard and therefore does not necessarily solve the problem of synchronizing a collection that depends from another collection first.

[0031] By augmenting each priority level with a dependency level, the synchronization can occur in the correct order. Specifically, if the options collection and the auto-text collection are both assigned priority level “one”, the auto-text could be assigned dependency level “one”, whereas the options collection could be assigned dependency level “two”.

[0032] When receiving the initialization packet **312** from **Figure 3** at network node **150**, the sort step **314** looks at both the priority and the dependency level and finds that the auto-text collection has a higher dependency than the options collection, and therefore the auto-text collection is synchronized first. The dependency therefore provides for the ordering of the collections at the same priority level.

[0033] In a further embodiment, not all dependencies will need to be utilized. Thus, as will be appreciated, non-contiguous dependencies could be used. Specifically, in the above example, the auto-text collection could be assigned a dependency level “5” and the options collection could be assigned a dependency level “10”, leaving levels 6 to 9 unused. Thereafter, if another collection is placed on mobile device **110** that is dependent on the options collection, it can be assigned a higher dependency than the options collection without having to rearrange all of the collections’ dependency levels. Thus, the new collection could be assigned a dependency level “7” if it is less important than the auto-text collection but needs to be synchronized prior to the options collection.

Exemplary Network

[0034] The above can be implemented on a variety of networks in which a network node and a mobile device need to synchronize with each other. One example is illustrated in **Figure 4**.

[0035] **Figure 4** illustrates an architectural overview for the synchronization of collections. Three mobile devices, **110a**, **110b** and **110c**, are illustrated.

[0036] Mobile device **110a** is a wide area network mobile device (WAN) which communicates with a cellular network **420**.

[0037] Mobile device **110b** is a wireless local area network (WLAN) mobile device which communicates with a data access port **422**.

[0038] Mobile device **110c** is a mobile device that is connected to a computer **424** through either physical or short-range wireless communications, such as a USB cable, Bluetooth [™] or IrDA, among others.

[0039] As will be appreciated by those skilled in the art, a mobile device 110 from figures 1 to 3 could implement one or more of the connection techniques of mobile devices **110a**, **110b** and **110c** from **Figure 4**. Thus, for example, a wide area network mobile device **110a** could also include means for connecting the mobile device to a computer as with mobile device **110c**. Further, a mobile device could include a wide area network connection (WAN connection) and a means to connect to a local computer all on one device. Other options would be apparent to those skilled in the art.

[0040] Mobile device **110a** may connect through cellular network **420** to provide voice or data. As will be appreciated, various cellular networks exist, including, but not limited to, GSM, GPRS, CDMA, UMTS, WCDMA, among others.

Data from mobile device **110a** proceeds through a relay **440**, preferably through a firewall **442** to a network node **150**.

[0041] Mobile device **110b** communicates over a data access point for a wireless local network **422**. Examples of WLAN technology include WiFi as one underlying technology of wireless local area networks.

[0042] As with a cellular connection, data is routed from data access point **422**, preferably through firewall **442** to network node **150**.

[0043] A mobile device **110c** can be connected to a local computer **424** through a number of means as described above. If local computer **424** includes an Internet connection, then data can be routed through the Internet, through optional firewall **442** to a network node **150** for synchronization of collections.

Exemplary Mobile Device

[0044] As will be appreciated, the above can be implemented on any mobile device in which a collection needs to be synchronized with a collection on a network node. One exemplary mobile device is described below with reference to **Figure 5**. This is not meant to be limiting, but is provided for illustrative purposes.

[0045] **Figure 5** is a block diagram illustrating a mobile device apt to be used with preferred embodiments of the apparatus and method of the present application. Mobile device **500** is preferably a two-way wireless communication device having at least voice and data communication capabilities. Mobile device **500** preferably has the capability to communicate with other computer systems on the Internet. Depending on the exact functionality provided, the wireless device may be referred to as a data messaging device, a two-way pager, a wireless e-mail device, a cellular telephone with data messaging capabilities, a wireless Internet appliance, or a data communication device, as examples.

[0046] Where mobile device **500** is enabled for two-way communication, it will incorporate a communication subsystem **511**, including both a receiver **512** and a transmitter **514**, as well as associated components such as one or more, preferably embedded or internal, antenna elements **516** and **518**, local oscillators (LOs) **513**, and a processing module such as a digital signal processor (DSP) **520**. As will be apparent to those skilled in the field of communications, the particular design of the communication subsystem **511** will be dependent upon the communication network in which the device is intended to operate.

[0047] Network access requirements will also vary depending upon the type of network **519**. In some CDMA networks, for example, network access is associated with a subscriber or user of mobile device **500**. A CDMA mobile device may require a removable user identity module (RUIM) or a subscriber identity module (SIM) card in order to operate on a CDMA network. The SIM/RUIM interface **544** is normally similar to a card-slot into which a SIM/RUIM card can be inserted and ejected like a diskette or PCMCIA card. The SIM/RUIM card can have approximately 64K of memory and hold many key configuration **551**, and other information **553** such as identification, and subscriber related information.

[0048] When required network registration or activation procedures have been completed, mobile device **500** may send and receive communication signals over the network **519**. As illustrated in **Figure 5**, network **519** can consist of multiple base stations communicating with the mobile device. For example, in a hybrid CDMA 1x EVDO system, a CDMA base station and an EVDO base station communicate with the mobile device and the mobile device is connected to both simultaneously. The EVDO and CDMA 1x base stations use different paging slots to communicate with the mobile device.

[0049] Signals received by antenna **516** through communication network **519** are input to receiver **512**, which may perform such common receiver functions as signal amplification, frequency down conversion, filtering, channel selection and the like, and in the example system shown in **Figure 5**, analog-to-digital (A/D) conversion. A/D conversion of a received signal allows more complex communication functions such as demodulation and decoding to be performed in the DSP **520**. In a similar manner, signals to be transmitted are processed, including modulation and encoding for example, by DSP **520** and input to transmitter **514** for digital-to analog-conversion, frequency-up-conversion, filtering, amplification and transmission over the communication network **519** via

antenna **518**. DSP **520** not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in receiver **512** and transmitter **514** may be adaptively controlled through automatic gain control algorithms implemented in DSP **520**.

[0050] Mobile device **500** preferably includes a microprocessor **538**, which controls the overall operation of the device. Communication functions, including at least data and voice communications, are performed through communication subsystem **511**. Microprocessor **538** also interacts with further device subsystems such as the display **522**, flash memory **524**, random access memory (RAM) **526**, auxiliary input/output (I/O) subsystems **528**, serial port **530**, one or more keyboards or keypads **532**, speaker **534**, microphone **536**, other communication subsystem **540** such as a short-range communications subsystem and any other device subsystems generally designated as **542**. Serial port **530** could include a USB port or other port known to those in the art.

[0051] Some of the subsystems shown in **Figure 5** perform communication-related functions, whereas other subsystems may provide “resident” or on-device functions. Notably, some subsystems, such as keyboard **532** and display **522**, for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list.

[0052] Operating system software used by the microprocessor **538** is preferably stored in a persistent store such as flash memory **524**, which may instead be a read-only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device applications, or parts thereof, may be temporarily loaded into a volatile memory such as RAM **526**. Received communication signals may also be stored in RAM **526**.

[0053] As shown, flash memory **524** can be segregated into different areas for both computer programs **558** and program data storage **550**, **552**, **554** and **556**. These different storage types indicate that each program can allocate a portion of flash memory **524** for their own data storage requirements. Various collections are also stored in flash memory **524**. Microprocessor **538**, in addition to its operating system functions, preferably enables execution of software applications on the mobile device. A predetermined set of applications that control basic operations, including at least data and voice communication applications for example, will normally be installed on mobile device **500** during manufacturing. Other applications could be installed subsequently or dynamically.

[0054] A preferred software application may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the user of the mobile device such as, but not limited to, e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores would be available on the mobile device to facilitate storage of PIM data items. Such PIM application would preferably have the ability to send and receive data items, via the wireless network **519**. In a preferred embodiment, the PIM data items are seamlessly integrated, synchronized and updated, via the wireless network **519**, with the mobile device user's corresponding data items stored or associated with a host computer system. Further applications may also be loaded onto the mobile device **500** through the network **519**, an auxiliary I/O subsystem **528**, serial port **530**, short-range communications subsystem **540** or any other suitable subsystem **542**, and installed by a user in the RAM **526** or preferably a non-volatile store (not shown) for execution by the microprocessor **538**. Such flexibility in application installation increases the functionality of the device and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the mobile device **500**.

[0055] In a data communication mode, a received signal such as a text message or web page download will be processed by the communication subsystem **511** and input to the microprocessor **538**, which preferably further processes the received signal for output to the display **522**, or alternatively to an auxiliary I/O device **528**.

[0056] A user of mobile device **500** may also compose data items such as email messages for example, using the keyboard **532**, which is preferably a complete alphanumeric keyboard or telephone-type keypad, in conjunction with the display **522** and possibly an auxiliary I/O device **528**. Such composed items may then be transmitted over a communication network through the communication subsystem **511**.

[0057] For voice communications, overall operation of mobile device **500** is similar, except that received signals would preferably be output to a speaker **534** and signals for transmission would be generated by a microphone **536**. Alternative voice or audio I/O subsystems, such as a voice message recording subsystem, may also be implemented on mobile device **500**. Although voice or audio signal output is preferably accomplished primarily through the speaker **534**, display **522** may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call-related information for example.

[0058] Serial port **530** in **Figure 5** would normally be implemented in a personal digital assistant (PDA)-type mobile device for which synchronization with a user's desktop computer (not shown) may be desirable, but is an optional device component. Such a port **530** would enable a user to set preferences through an external device or software application and would extend the capabilities of mobile device **500** by providing for information or software downloads to mobile device **500** other than through a wireless communication network. The alternate download path may for example be used to load an encryption key onto the

device through a direct and thus reliable and trusted connection to thereby enable secure device communication. As will be appreciated by those skilled in the art, serial port **530** can further be used to connect the mobile device to a computer to act as a modem.

[0059] Other communications subsystems **540**, such as a short-range communications subsystem, is a further optional component which may provide for communication between mobile device **500** and different systems or devices, which need not necessarily be similar devices. For example, the subsystem **540** may include an infrared device and associated circuits and components or a Bluetooth™ communication module to provide for communication with similarly enabled systems and devices.

[0060] The embodiments described herein are examples of structures, systems or methods having elements corresponding to elements of the techniques of this disclosure. This disclosure may enable those skilled in the art to make and use embodiments having alternative elements that likewise correspond to the elements or the techniques of this disclosure. The intended scope of the techniques of this disclosure thus includes other structures, systems or methods that do not defer from the techniques of the disclosure as described herein, and further includes other structures, systems or methods within substantial differences from the techniques of this disclosure as described herein.

CLAIMS

1. A method in a mobile device, the method comprising:
 - 5 assigning, by a processor, one or more dependency level values to collections of data, each collection of data having an existing priority level used in ordering the collections for synchronization, wherein the dependency level values are assigned to the collections of data for extended ordering of the collections within each existing priority level; and
 - 10 sending the existing priority level and dependency level value of each collection to a network node to initiate synchronization of the collections of data based on the extended ordering, wherein the extended ordering of collections that share a similar existing priority level is determined by the dependency level values.
- 15 2. The method of claim 1, wherein the dependency field is eight bits.
3. The method of claim 1 or 2, wherein the sending comprises transmitting an initialization packet from the mobile device, which includes the priority level
 - 20 and dependency level value of each collection.
4. The method of claim 1, wherein the sending step is performed between a mobile device and a network node in a wireless network.
- 25 5. The method of any one of claims 1 to 4, wherein the assigning step assigns a non-contiguous dependency level value from any previously added dependency level values to collections.
6. The method of any one of claims 1 to 5, wherein at least two of the
 - 30 collections of data share a same previously assigned priority level.

7. The method of claim 1, wherein the dependency level values are represented by a one-byte dependency field appended to a priority field.
8. The method of claim 1, wherein the synchronization of the collections are
5 ordered by the priority level first and by dependency level value second.
9. The method of claim 1, wherein the synchronization is performed between the mobile device and the network node over a wireless network.
- 10 10. A computer readable medium storing program software executable by a processor of a computing device for causing said computing device to perform the method of any one of claims 1 to 9.
11. A mobile device adapted for synchronizing collections of data, the device
15 comprising:
means for assigning one or more dependency level values to collections of data, each collection of data having an existing priority level used in ordering the collections for synchronization, wherein the dependency level values are assigned to the collections of data for extended ordering of the collections of data
20 within each existing priority level; and
a communication system for transmitting the existing priority level and dependency level value for each collection during synchronization to cause synchronization of the collections based on the extended ordering, wherein the extended ordering of collections that share a similar existing priority level is
25 determined by the dependency level values.
12. The mobile device of claim 11, wherein the dependency field is eight bits.
13. The mobile device of claim 11 or 12, wherein, during the sending
30 operation, the mobile device transmits an initialization packet which includes the priority level and dependency value of each collection.

14. The mobile device of any one of claims 11 to 13 wherein the means for assigning is further adapted to assign a non-contiguous dependency level value from any previously added dependency level values to collections.

5

15. The mobile device of any one of claims 11 to 14, wherein the dependency level values are represented by a one-byte dependency field appended to a priority field.

10

16. A method in a network node, the method comprising:

receiving, from a mobile device, priority levels and dependency level values associated with collections of data, wherein the priority level comprise existing priority levels in an existing priority scheme for ordering the collections for synchronization; and

15

sending synchronization commands for synchronizing the collections of data with the mobile device, based on an extended ordering of the collections, wherein the extended ordering of collections that share a similar priority level is determined by the dependency level values.

20

17. A network node comprising:

a communication subsystem configured for:

receiving, from a mobile device, priority levels and dependency level values associated with collections of data, wherein the priority level comprise existing priority levels in an existing priority scheme for ordering the collections for synchronization; and

25

sending synchronization commands, for synchronizing the collections of data with the mobile device, based on extended ordering of the collections, wherein the extended ordering of collections that share a similar priority level is determined by the dependency level values.

30

1/5

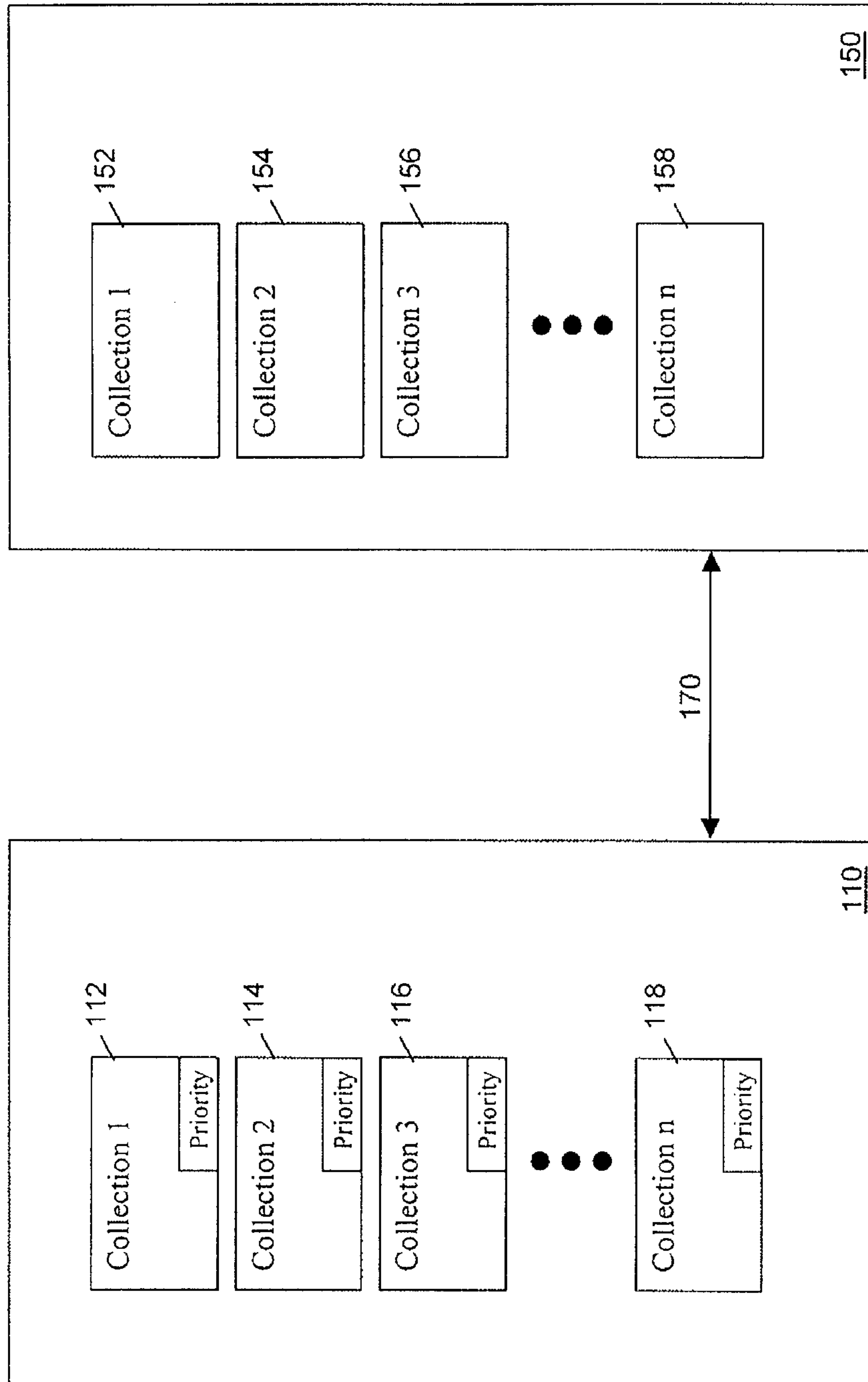


FIG. 1

2/5

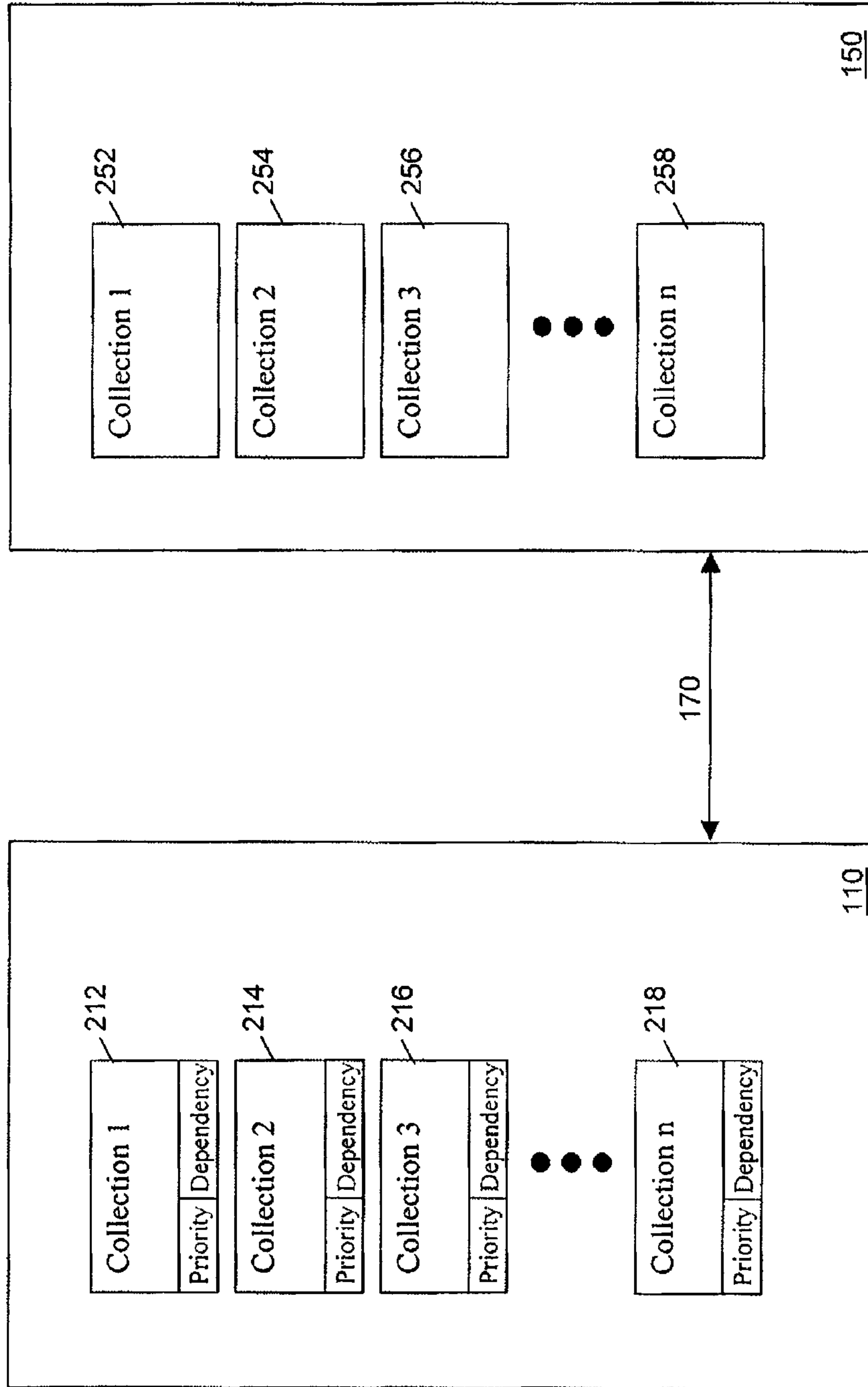


FIG. 2

3/5

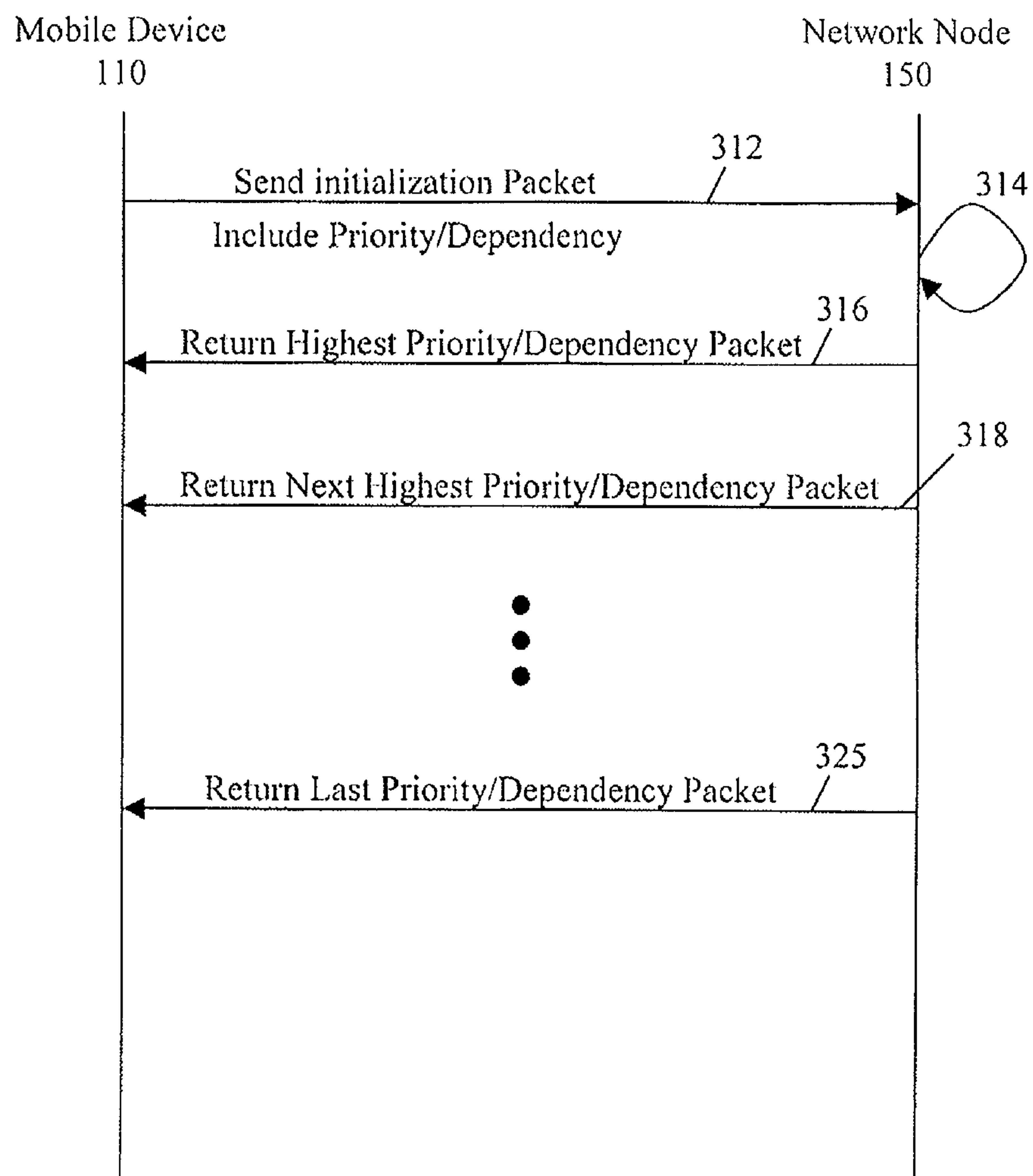


FIG. 3

4/5

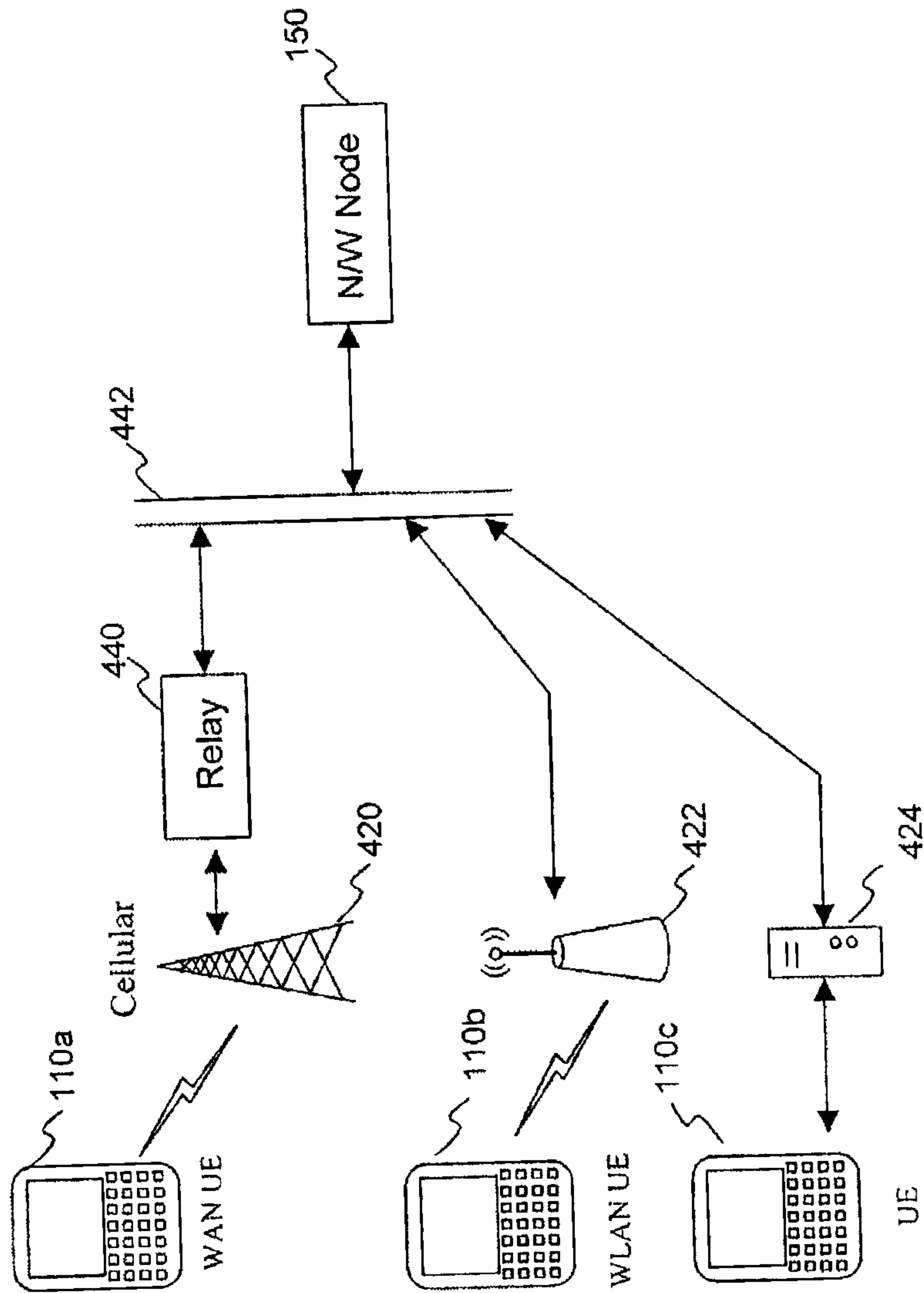


FIG. 4

5/5

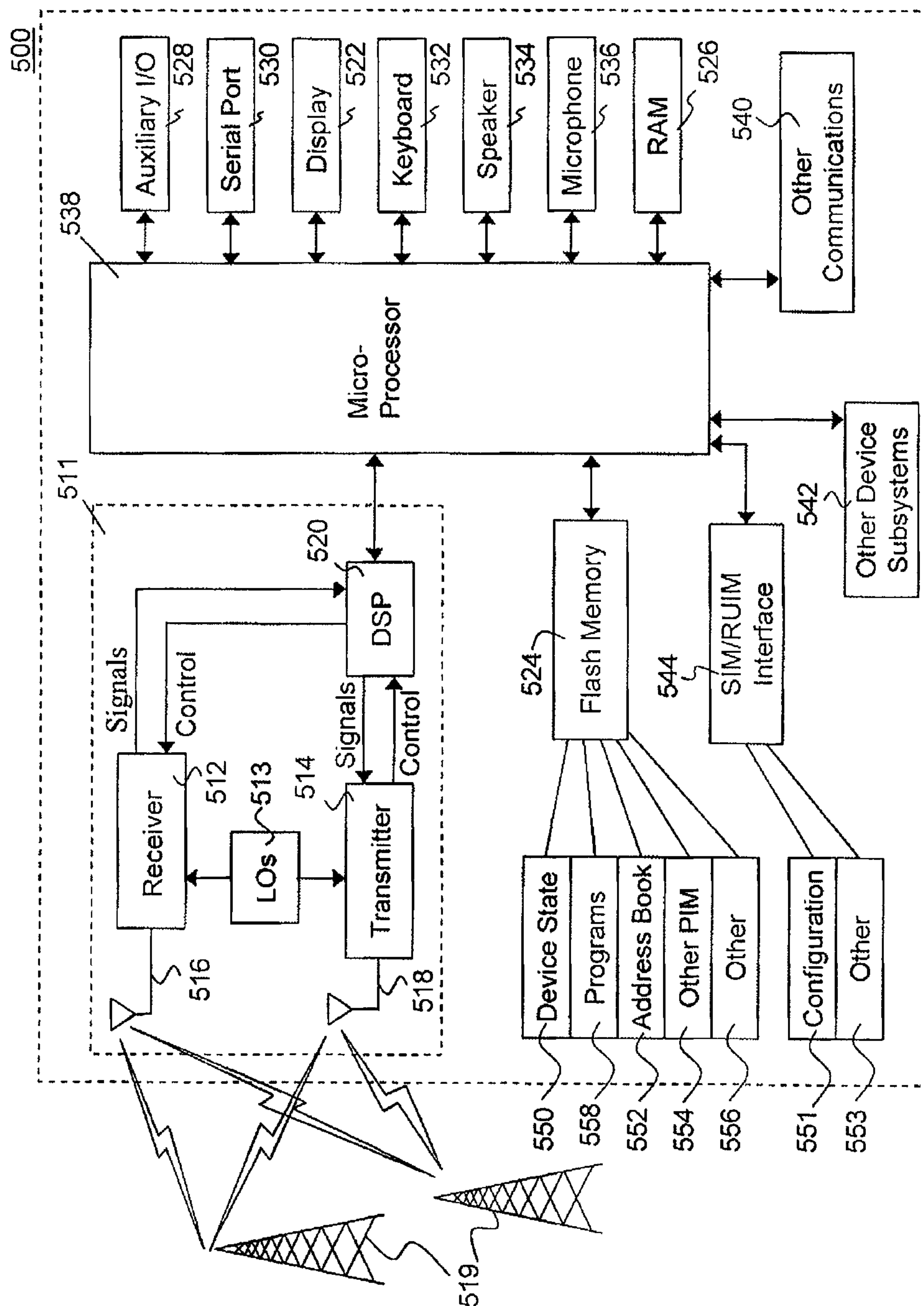


FIG. 5

Mobile Device
110

Network Node
150

