



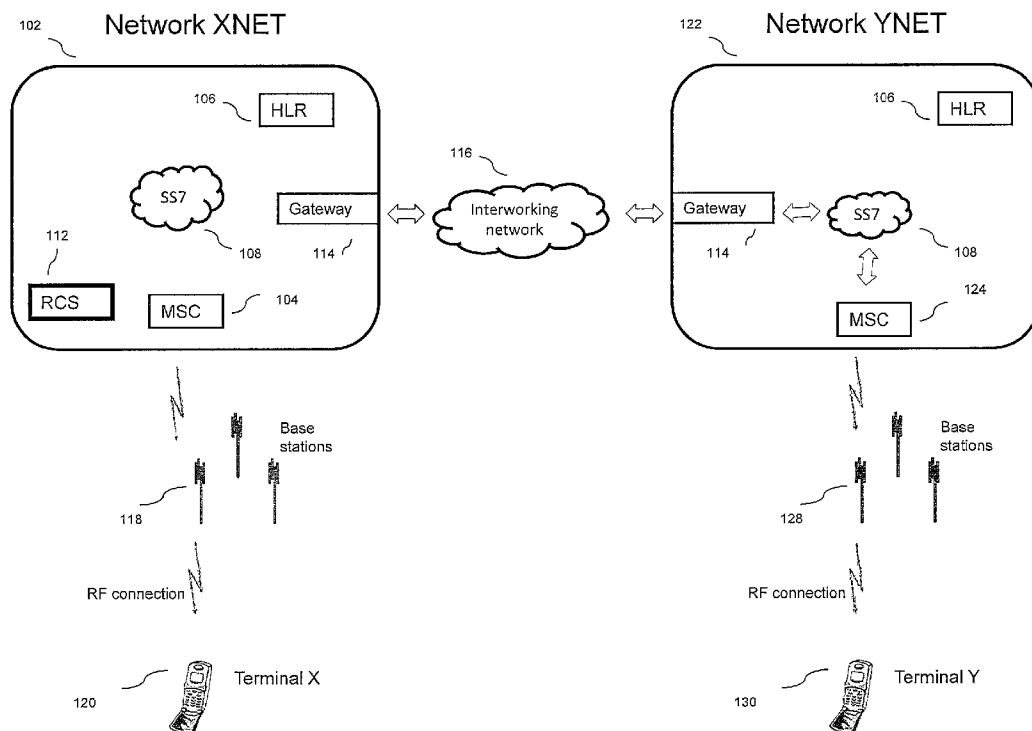
US 20110230196A1

(19) **United States**(12) **Patent Application Publication****Tal et al.**(10) **Pub. No.: US 2011/0230196 A1**(43) **Pub. Date: Sep. 22, 2011**(54) **DROPPED CALL RECONNECTION SYSTEM WITH INTER-NETWORK CAPABILITIES**(52) **U.S. Cl. 455/450**(75) Inventors: **Oren Tal, Tel-Aviv (IL); Reuven Tal, Tel-Aviv (IL)**(73) Assignee: **On-Q Telecom Systems Co., Inc.,**
Cheyenne, WY (US)(21) Appl. No.: **12/839,161**(22) Filed: **Jul. 19, 2010****Related U.S. Application Data**

(60) Provisional application No. 61/315,076, filed on Mar. 18, 2010.

Publication Classification(51) **Int. Cl.**
H04W 72/00 (2009.01)(57) **ABSTRACT**

Reconnecting a dropped call between two parties by detecting an unintentional disconnection of one connection of the call using a system connected to a network associated with at least one of the parties to the call. A detection task receives the number identifications of the two parties; receives a release message associated with the call; and tests a cause-indicator in the received release message to determine a match to a prescribed criterion. In response to a match, a portion of the set of resources allocated to the call by the entities in a network of the party that was not unintentionally disconnected from the network are preserved while permitting release of all other resources allocated to the call. A reconnection task executes to ultimately reconnect the dropped party using at least the preserved portion of the set of resources allocated to the original call. A system is also described.



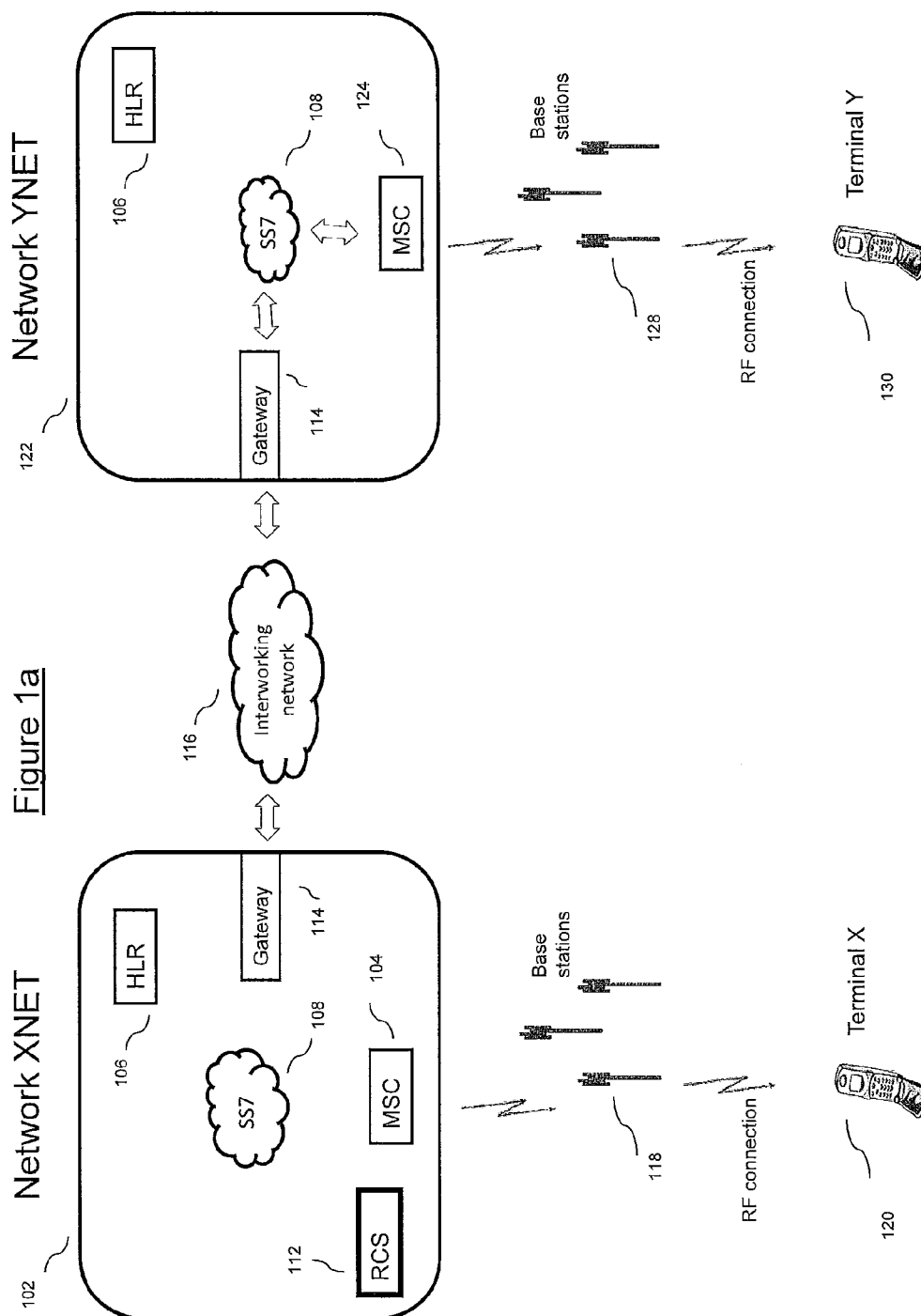


Figure 1b

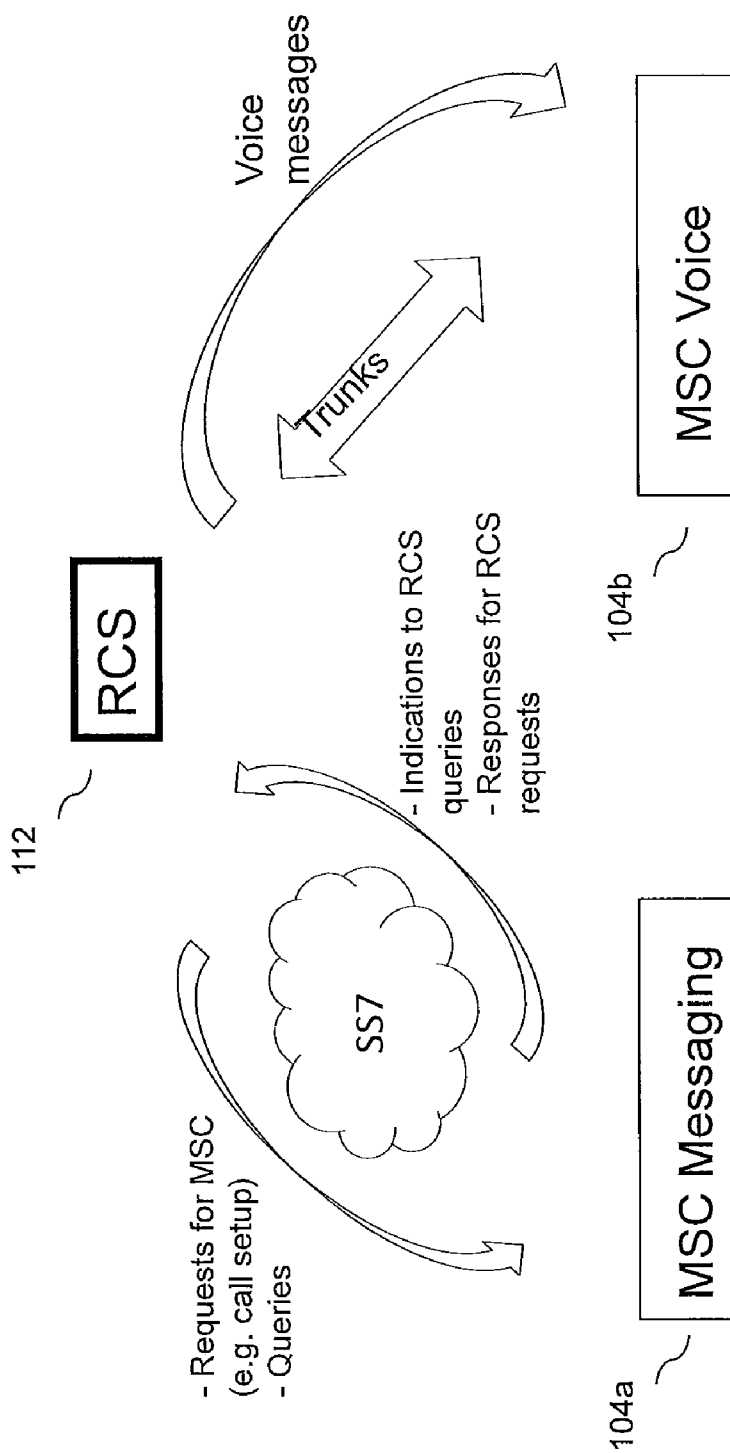


Figure 2

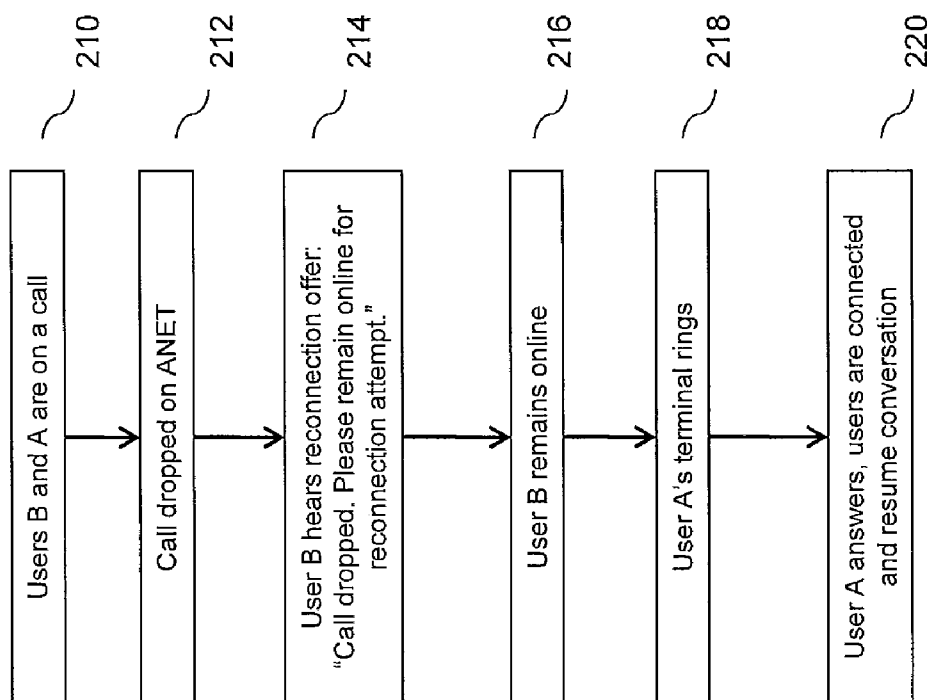
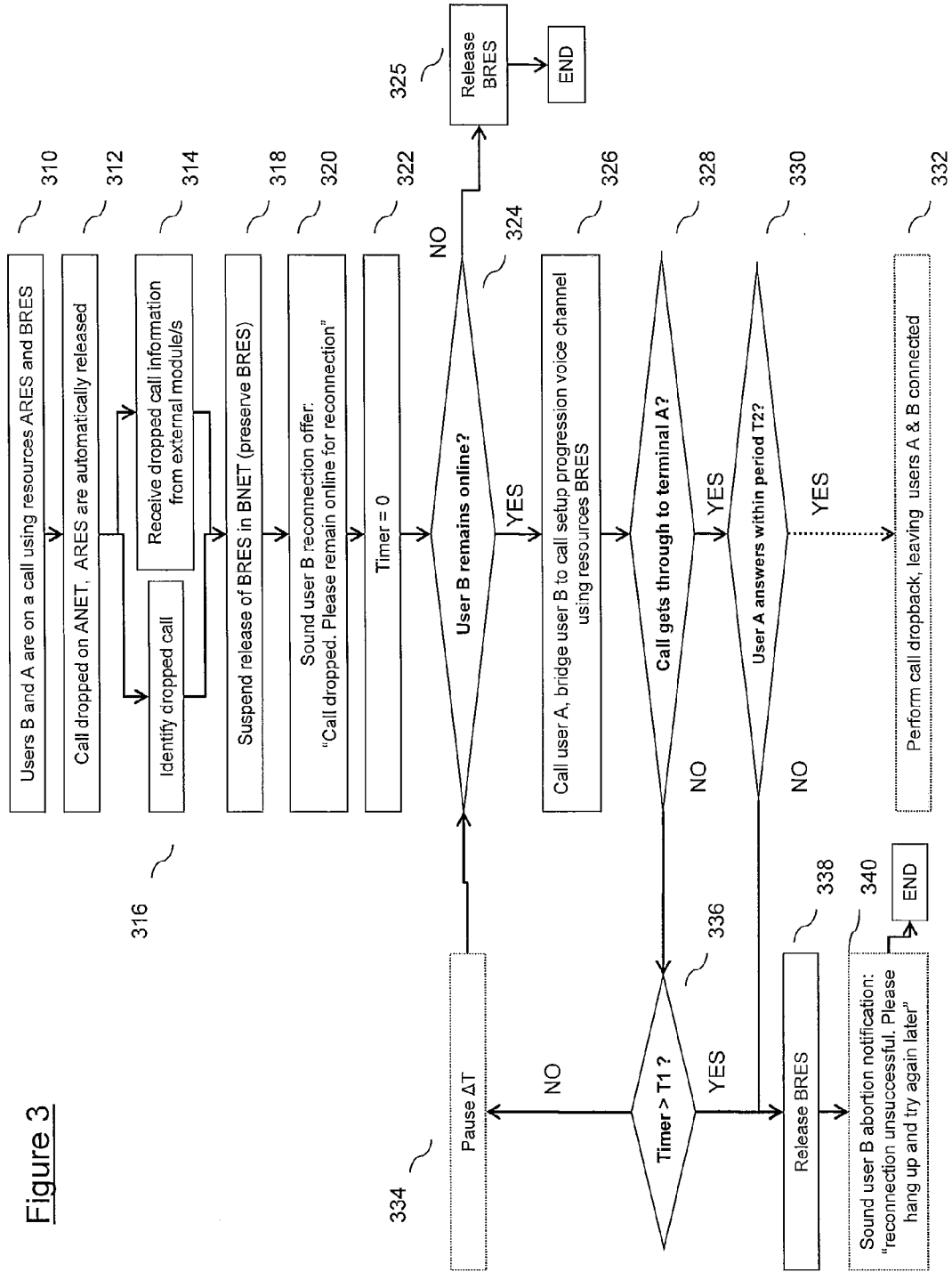


Figure 3



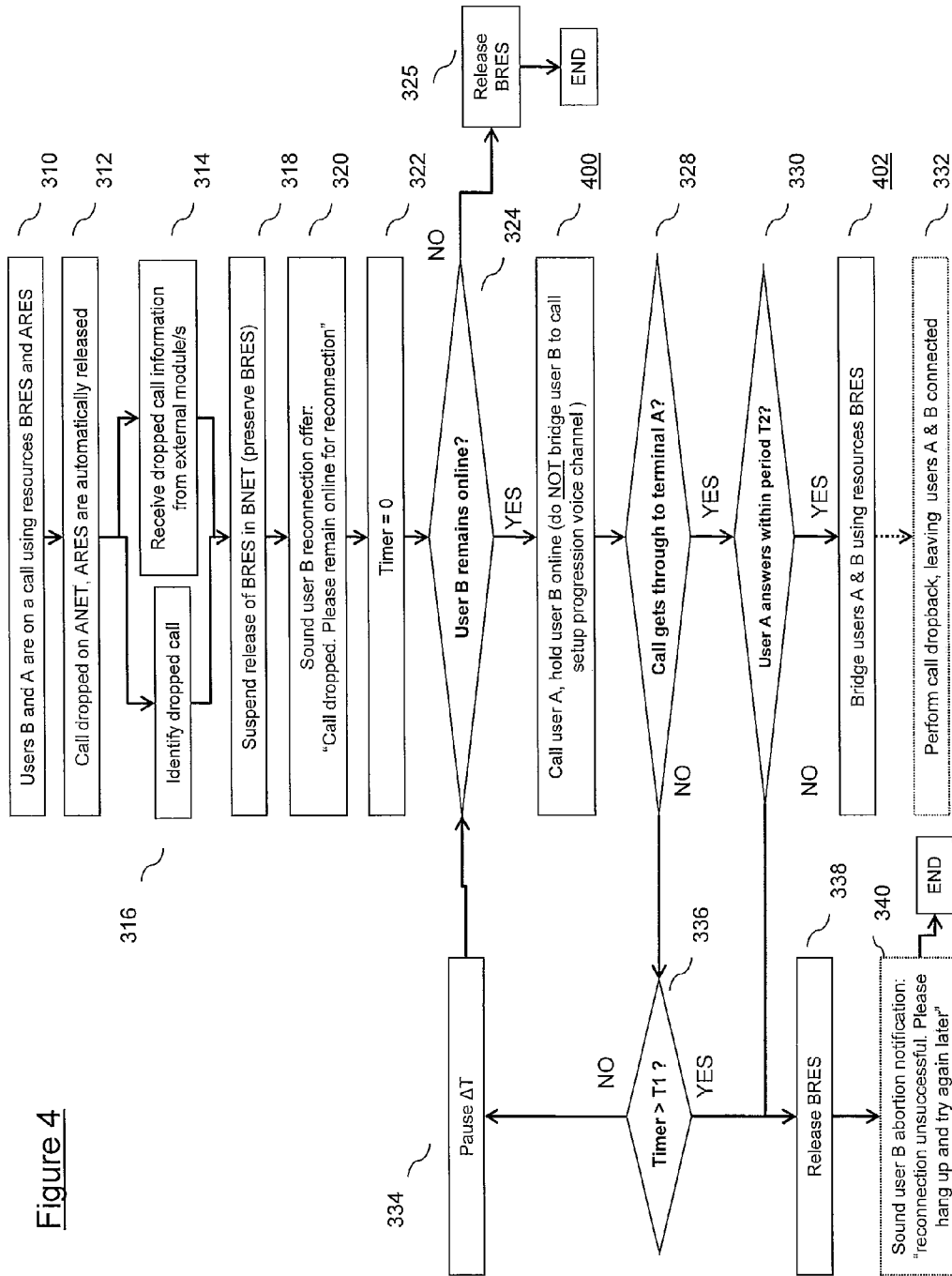


Figure 5

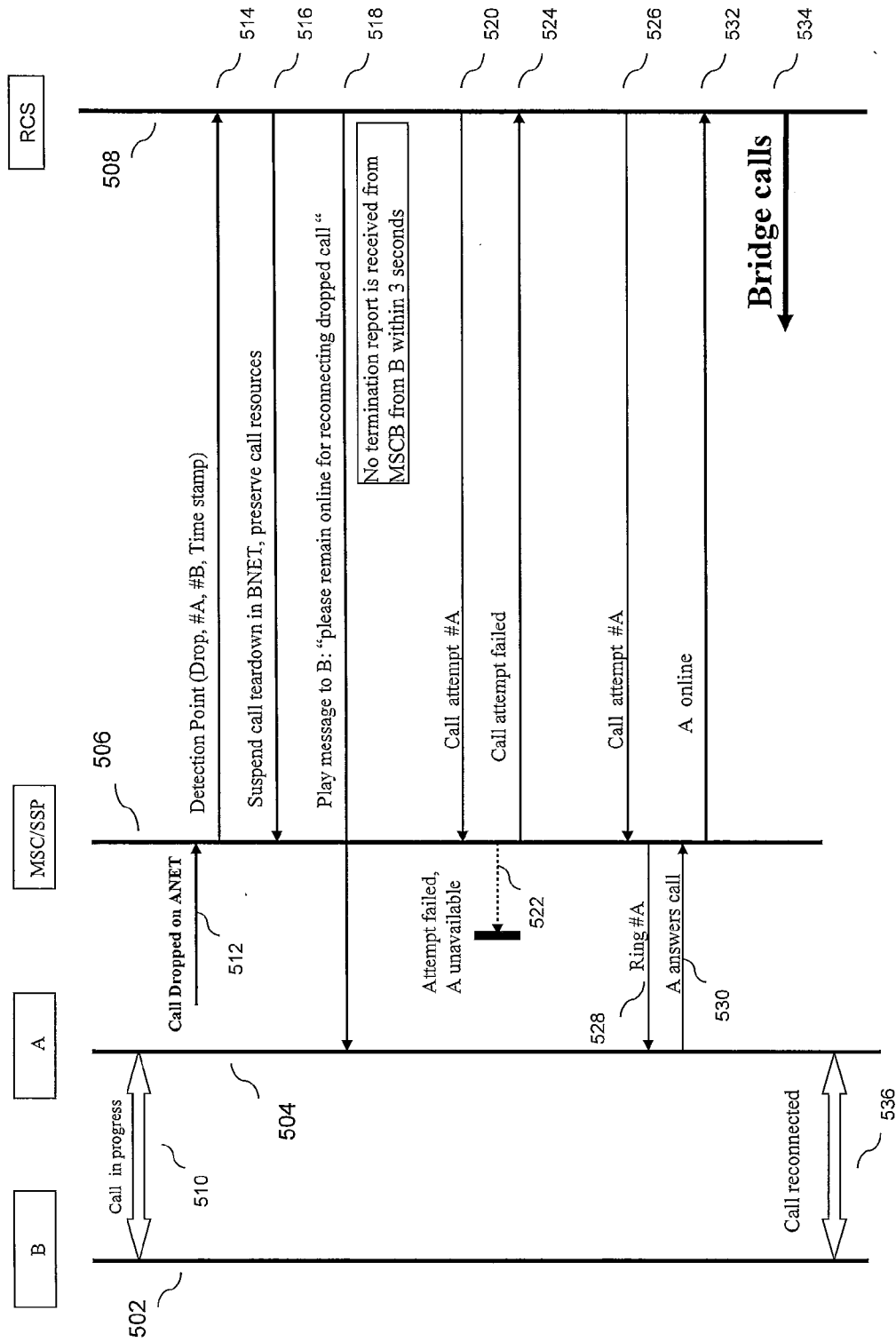


Figure 6

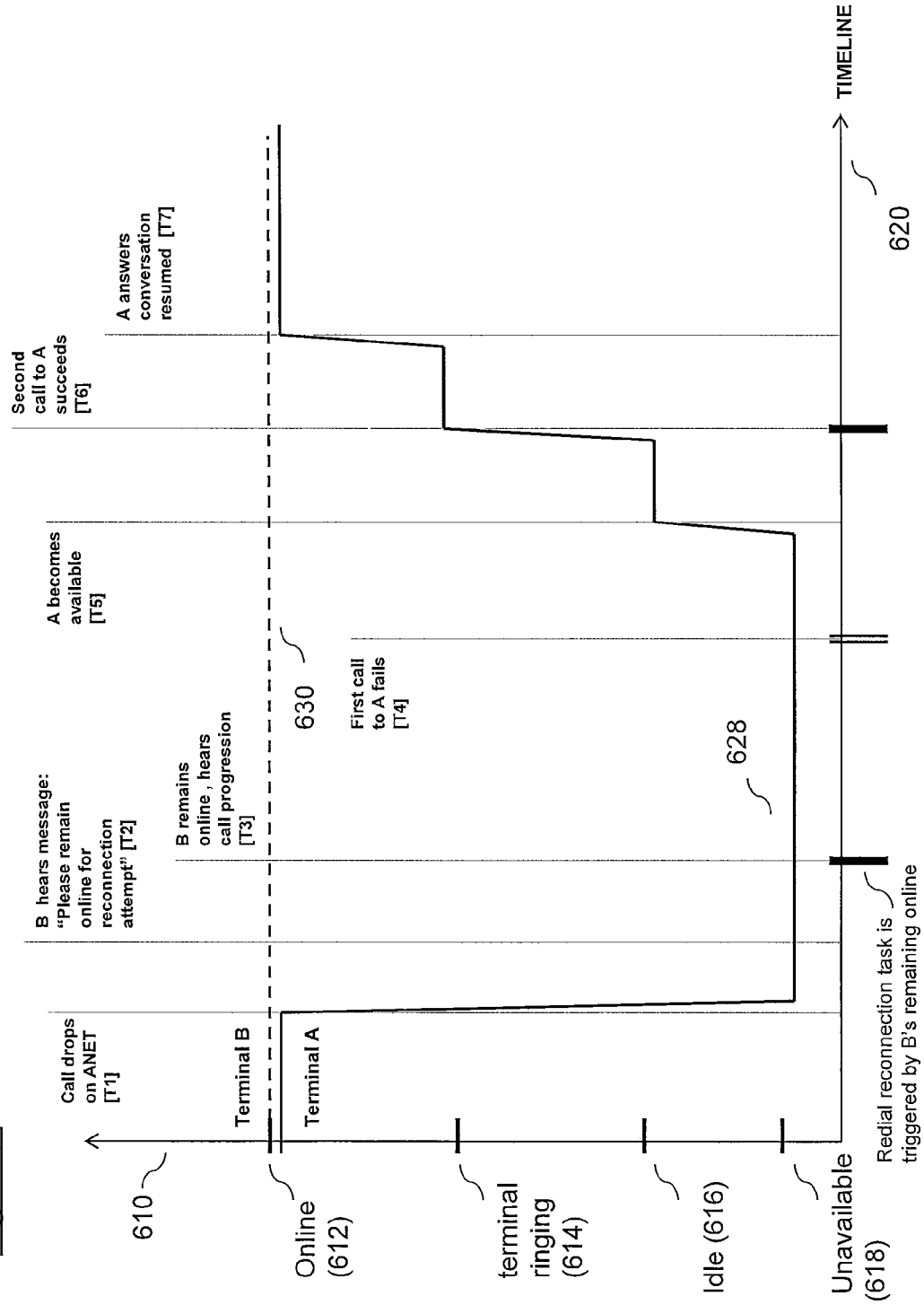
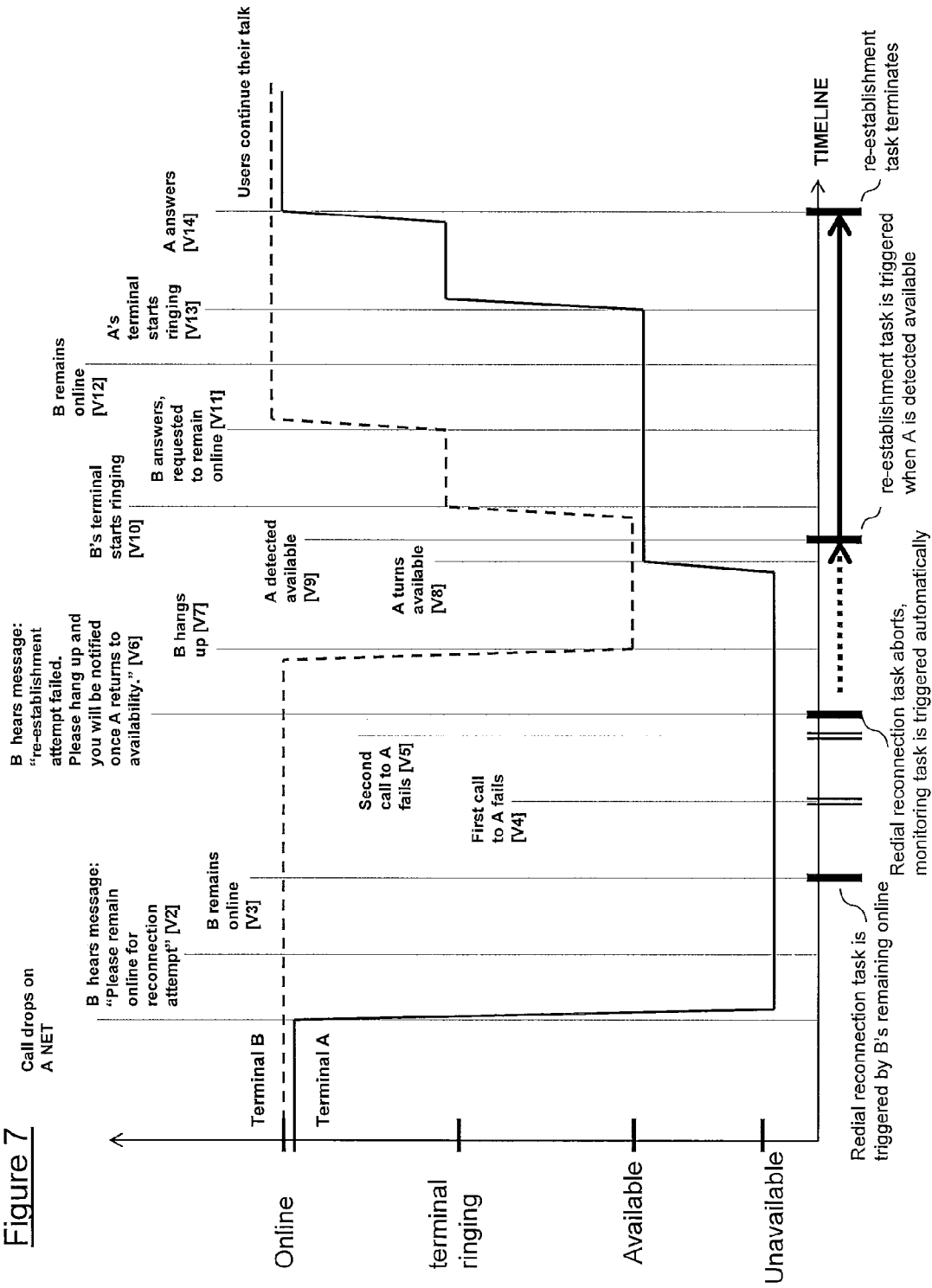


Figure 7



DROPPED CALL RECONNECTION SYSTEM WITH INTER-NETWORK CAPABILITIES

[0001] This application claims the benefit of priority from U.S. Provisional Application Ser. No. 61/315,076, filed on Mar. 18, 2010, entitled “Inter-Network Reconnection System,” which is hereby incorporated by reference as if set forth in its entirety herein.

FIELD OF THE INVENTION

[0002] The present invention relates to communication systems and, more particularly, relates to dropped-call reconnection systems and methods configured to reconnect communications between a caller and a call destination, including among subscribers of potentially different networks.

BACKGROUND OF THE INVENTION

Terminology

[0003] Modern mobile communication systems include a vast variety of technologies and standards such as cellular systems, WiFi, WiMax, and more. Even within the cellular systems family different standards exist such as GSM, CDMA, TDMA, IDEN. These systems and technologies are generally used to transfer two types of content—voice and data. The introduction of technologies such as voice over internet protocol (VoIP), in which both voice and data are transferred in packets, contributes to the delineation of voice and data transfer.

[0004] A basic element in every communication network is a dynamic database holding the current attributes of each terminal in the network, such as its availability, its location, its settings and its preferences. This database may be located in one physical machine or alternatively distributed across the network. In traditional cellular networks, HLR (Home Location Register) is the database holding the attributes of the network’s own subscribers. In advanced communication networks, other types of databases and architectures may be utilized for keeping track of dynamic attributes of users that are registered or even just recognized by the network. For brevity and clearness, the descriptions hereinafter will use the traditional term “HLR” but it is to be understood that this term represents the much broader entities as described herein.

[0005] In addition, the convergence of the aforementioned technologies sometimes allows interoperability between technologies—for example, one side of a call can be e.g., a cellular terminal and the other e.g., a WiFi terminal. To make things more complicated, “hybrid” devices have been introduced that enable communicating between two mobile devices supporting two or more technologies, with an ability to transfer between communication technologies even in the course of a call. Therefore, what has traditionally been referred to as a “call” between two or more users using a wireline and/or mobile devices can presently be referred to as the much broader term of “communication session”. Accordingly, the traditional act of “dialing” can now be referred to as the much broader “attempt to set-up a communication session”. For brevity and clearness, the descriptions hereinafter will use the traditional terms such as “dial”, “call”, “calling”, “caller”, “destination”, but it is to be understood that these terms represent the much broader entities as described herein.

[0006] According to the same line of logic, the traditional SMS message that was until not long ago an essentially sole

method of transferring data between mobile devices has evolved into a vast variety of messages such as Multi-Media Messages (MMS) and instant messages (IM). In another example, data is delivered by a call that goes out to user and plays a recorded audio sign or statement. For brevity, the descriptions that follow will use the term “SMS” and this term is to be understood as a message in its broadest sense as elaborated herein, including the evolved incarnations noted above and that generally are part of a message service, whether or not an “SMS” in the traditional sense. Messages sent within certain types of messaging systems can be configured to trigger, when delivered at recipient’s mobile device, a report informing the delivery and sometimes the time of delivery, which is sent back to the entity that had sent the message (human or other). This type of report will be hereinafter referred to as a “delivery report”.

[0007] Hundreds of mobile operators exist around the world. Each operator operates its own communication equipment and infrastructure for supporting mobile communication of its subscribers. For a single operator, all these will be collectively referred to as a “network” entity. Noticeably, a network can combine two or more technologies, such as a cellular network combining CDMA and IDEN technologies. A call between users of two different operators will be hereinafter referred to as an “internetwork” call. Industry standards allow standard actions such as a call, an SMS and an instant message to be performed between users of different operators, or “interoperator”.

[0008] For a connection to be established between two terminals, a “call setup” procedure has to take place. This typically commences with first user initiating a “call setup request” on his/her terminal. This request propagates from user’s terminal to the serving network, and from there on to second user’s serving network and, if possible, to second user’s terminal. Second user can then decide whether to approve the call setup. In case the second user approves, each network allocates certain resources at different network entities, e.g., switches and base stations. These resources support the connections that constitute the call, such as voice trunks and signaling resources. When the resource allocation procedure is successful in both networks then a call is established and users can commence the conversation. Notably, in each network, the resources allocated for a call are deducted from pools of respective resources available at the network, reducing the number of other connections that can be supported at any given instance. When a call terminates, at both networks a “call tear-down” procedure takes place which releases these resources to their respective pools.

Dropped Calls

[0009] Wireless communication systems are well known in which mobile units can initiate or receive calls while roaming between different radio frequency (RF) coverage areas (sometimes referred to as “cells”). The mobile units communicate via RF resources with base stations distributed among the cells, which base stations are controlled by one or more mobile switching centers (MSCs). The MSCs provide control signaling for the call and connect the mobile unit to other participating endpoints, which may comprise other mobile units or wireline units.

[0010] In general, a call between two users in a communication network is constituted by a plurality of connections between different network entities, such as connections between the users and their serving base stations and connec-

tions between base stations and their respective mobile switching centers (MSCs). If one of these connections is impaired during an active call, the call might be unintentionally-disconnected. A major cause for call disconnections is a loss or a severe degradation in one of the connections between users and base stations or base stations and switches. Occasionally, mobile units can encounter service interruption(s) during a call, for example, upon entering a tunnel or reaching a fringe RF coverage area or due to a handoff error, causing the mobile unit to become unintentionally disconnected from the serving network and terminating the call. This occurrence is hereinafter referred to as a dropped call. The user whose mobile unit has been unintentionally disconnected from his/her serving network will be hereinafter referred to "user A" and the other at least one user shall be hereinafter referred to as "user B".

[0011] Dropped calls are a major cause of dissatisfaction of cellular phone users and may ultimately cause the unsatisfied user to terminate his/her subscription with the network, or in other terms, churn. A method for reconnecting an unintentionally-terminated call is therefore highly desirable from the point of view of the user as well as of the network carrier. After a call has dropped, or "unintentionally disconnected", in many cases it would be much desired to resume the conversation that had been disrupted.

PRIOR ART

[0012] The references below teach various methods for reconnecting a dropped call. These methods attempt to enable users that have experienced a dropped call to resume their conversation. In these methods, the party that was not disconnected (hereinafter referred to as user B, subscriber of network BNET) remains online waiting for the disconnected party (hereinafter referred to as user A, subscriber of network ANET) to rejoin the connection. Some methods are transparent to users, i.e., users experience only a short interruption in the voice channel and then continue their respective calls, while other methods provide user B a voice notification prompting him/her to remain online for user A to rejoin the call.

[0013] U.S. Pat. No. 6,215,782 to Buskens at al. teaches a method for reconnecting calls affected by a loss of synchronization. After a call is disconnected, a time-out sequence is initiated during which reconnection attempts are performed. The time-out sequence can be incremented a number of times before reconnection attempts are terminated and the call is released. In a further embodiment, a mobile switching center and a serving base station may selectively determine whether reconnection attempts are to be made.

[0014] U.S. Pat. No. 6,343,216 to Kim at al. teaches an automatic dropped call reconnection method in a communication system. According to this method, the base station (BS) serving the terminal that has been disconnected informs the serving mobile switching center (MSC) of the service impediment, the MSC sends a reconnection paging request to a group of base stations and/or another MSC, and the latter attempts reconnecting with the disconnected terminal.

[0015] U.S. Pat. No. 7,130,619 to Florkey et al. teaches a method and system for whereby a mobile unit may initiate reconnection of an interrupted call by sending a mobile-originated reconnect (MORC) message into the network; or, conversely, the mobile unit may affirmatively decline a reconnect attempt by choosing not to send a MORC message (or, alternatively, by sending a reconnect decline message) within

a designated waiting period. Responsive to the MORC message (or absence of MORC message), the network will conditionally attempt (or not attempt) to reconnect the call. In such manner, a method and system provide for a mobile unit affirmatively requesting or declining a reconnect attempt of a call, rather than relying on a network-initiated reconnect that may or may not occur; and for the network to reconnect a call without network-initiated pages.

[0016] The above references teach various methods for reconnecting a dropped call. These methods attempt to enable users that have experienced a dropped call to resume their conversation. In these methods, user B remains online waiting for user A to rejoin the connection. Some methods are transparent to users, i.e., users only experience a short interruption in the voice channel and then continue their call, while other methods provide user B a voice notification prompting him/her to remain online for user A to rejoin the call. These services will be hereinafter referred to as "call reconnection" services. Notably, all of these systems attempt to re-establish or reconnect connections or channels between two entities located within a common network in which they are installed, such as connection between a terminal and a base-station or a terminal and an MSC. In particular, none of these systems can reach out beyond the limits of the particular network in which it is installed since it cannot influence the operation of such entities in other networks.

[0017] One deficiency of the state-of-the-art call reconnection systems described above pertains to their inability to reconnect inter-network dropped calls in which a reconnection system is installed in BNET (user B's network) but not in ANET (user A's network). As noted above, state-of-the-art reconnection systems installed in BNET cannot reconnect a call dropped in ANET.

[0018] A second deficiency pertains generally to the implementation of these systems in existing networks. Implementing state-of-the-art systems in existing networks requires extensive network installations and modifications, specifically, on base stations controllers (BSCs) and sometimes on MSCs and terminals. For example, in U.S. Pat. No. 7,130,619 to Florkey et al. it is apparent that the terminal and base stations must be provided with dedicated modules executing the MORC procedure. Implementing communication protocols on BSC stations and/or terminals requires installation of software and/or hardware modules within these entities. For BSCs, this involves extensive activities such as integration, testing and more. For terminals, this requires a dedicated module to be installed. Practically viewed, the large number of BSCs in a network and the vast variety of handset manufacturers and models in use present significant hurdles to any wide implementation of such systems on existing networks.

[0019] A different approach for handling dropped calls is taught in co-pending U.S. application Ser. No. TBA [Attorney Docket No. 3267/0851-US1], filed on even-date herewith, titled "Dropped Call Re-establishment System with Inter-Network Capabilities," assigned to the present assignee, and which is hereby incorporated by reference as if set forth in its entirety herein. This patent application teaches a method and system for re-establishing dropped calls between possibly different networks. In contrast to the reconnection methods that preserve some of the dropped call's resources and use them for the reconnected call, the re-establishment methods taught in this application allow the release of all resources associated with the dropped call in both networks and establish a new independent call. Effectively, this means that user

B must be brought offline (terminal “idle”) before the system commences its re-establishment attempts.

[0020] The re-establishment method as described in the aforementioned “Dropped Call Re-establishment System with Inter-Network Capabilities” application can be viewed as a general method that can handle all types of dropped calls, irrespective of where they dropped, their cause and the time it takes until conditions allow a further connection. However, in certain situations there is a high probability that conditions will enable users A and B to connect shortly after the drop. Further, in certain situations user B can be assumed to be more tolerant to failed reconnection attempts. In these situations, it may be desirable and beneficial to keep user B online for a reconnection attempt instead of (or even prior to) allowing him/her to hang up.

[0021] Accordingly, it would be an improvement in the art to facilitate the reconnection of dropped calls between subscribers of possibly different operators. The system, when installed in a network, will support reconnection of dropped calls in which at least one user is a user of that network. Advantageously and optionally, the system will attempt to reconnect calls selectively, based on their chance of being immediately reconnected. Advantageously and optionally, a reconnection will take place in the shortest time possible. Advantageously and optionally, when user remains online user is informed of the service flow status to the maximum extent. Advantageously and optionally, system implementation in an existing network will not require extensive installations and modifications. The present invention is directed to addressing one or more of these needs, either as a system or a method.

SUMMARY OF THE INVENTION

[0022] According to one aspect of the present invention, a method for reconnecting a dropped call between two parties is described. The call is in a communication system and is dropped as a result of a loss of one of a plurality of connections which constitute the call between the two parties. The call is supported by a set of resources allocated to the call by the entities of both parties’ networks, and the method comprises, in part, detecting that one connection of the call has been unintentionally-disconnected. Such detection is made using a reconnection system (RCS) connected to a network associated with at least one of the parties to the call, the RCS having a detection task module executing a detection task on a processor thereof. The detection task is operative to: receive the number identifications of the two parties; receive a release message associated with the call; and test a cause-indicator in the received release message in a machine executing the RCS to determine a match to a prescribed criterion. In part, the method responds to the match by preserving a portion of the set of resources allocated to the call by the entities in a network of user B—the party that was not unintentionally disconnected from the network, while permitting release of all other resources allocated to the call. In part, the method responds to the match by executing a reconnection task within the RCS using the reconnection task module. The reconnection task executes to initiate a call set-up request with user A—the party that has been unintentionally disconnected from the network, and to repeat the call-set up request for a pre-defined time limit until user A’s terminal rings. Upon A answering the call, a reconnection is made using at least the preserved portion of the set of resources allocated to the call in the network of user B.

[0023] According to another aspect of the invention, a reconnection system (RCS) is described that reconnects a dropped call between a user A (who has been disconnected from a communication network) and a user B (the other party to the call). The disconnection is a result of a loss of at least one of a plurality of connections between user A and user B which together constitute the call. The RCS is connected to the communication network and is associated with at least one of user A and user B. The call is supported by a set of resources allocated to the call by respective switching networks associated with each of user A and user B. A database and a computer having a memory and a processor for executing modules therein are provided, wherein the computer is in communication with the communication network. A detection task module is loaded into the memory of the computer and executes at least one detection task in the processor. The detection task configures the processor to cause the computer to: load into the memory number identifications of user A and user B which are provided by the communication network; load into the memory a release message associated with the call which is provided by the communication network; and test a cause-indicator in the release message to determine a match to a prescribed criterion stored in the database. A call-teardown-suspension-task module is loaded into the memory of the computer and executes at least one call-teardown-suspension-task in the processor in response to the test by the detection task returning the match. The call-teardown-suspension-task configures the processor to cause the computer to preserve a portion of the set of resources allocated to the call by the switching network of user B while permitting release of all other resources allocated to the call. A reconnection task module is loaded into the memory of the computer that executes at least one reconnection task in the processor in response to the test by the detection task returning the match. A task manager is loaded into the memory of the computer and executes in the processor so as to configure the processor to monitor the execution of up to a multiplicity of detection tasks, call-teardown-suspension-tasks and reconnection tasks.

[0024] In one variation, a system as generally described above has the reconnection task configuring the processor to cause the computer to: initiate a call set-up request with user A; monitor whether user A’s terminal has started to ring based on data provided by the communication network; and if user A’s terminal has started to ring within a pre-defined time limit, then monitor whether user A answers the call-set up request based on data provided by the communication network and, if user A answers the call, then bridging user A and user B using at least the portion of the set of resources allocated to the call in the network of user B.

[0025] In another variation, a system as generally described above has the reconnection task configuring the processor to cause the computer to: initiate a call set-up request with user A that establishes a call setup voice channel; and bridge user B to the call setup voice channel using at least the portion of the set of resources allocated to the call in the network of user B.

[0026] These and other aspects, features, and advantages of the present invention, some of which are detailed in the claims attached hereto, can be further appreciated from the following discussion of certain embodiments of the invention taken together with the drawing figures that illustrate the embodiments thereof.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

[0027] FIG. 1a provides a block-diagram description of an RCS installed in a communication system and serving it.

[0028] FIG. 1*b* provides a schematic description the connectivity of the RCS system with several important external modules.

[0029] FIG. 2 provides a schematic description of the user experience after each occurrence of a dropped call.

[0030] FIG. 3 provides a schematic description of the stages performed by the RCS after each occurrence of a dropped call.

[0031] FIG. 4 provides a schematic description of the stages performed by the RCS after each occurrence of a dropped call in another embodiment.

[0032] FIG. 5 provides a schematic description of the signaling traffic involved in the process of reconnecting users A and B following a dropped call, according to the embodiment described in FIG. 4.

[0033] FIG. 6 provides a timeline perspective of a reconnection embodiment according to the present invention.

[0034] FIG. 7 provides a timeline perspective of an embodiment combining the present invention and a procedure taught in the co-pending “Dropped Call Re-establishment System with Inter-Network Capabilities” referenced herein above.

DETAILED DESCRIPTION OF CERTAIN EMBODIMENTS OF THE INVENTION

1) General

[0035] a) Introduction and Overview

[0036] The present invention addresses these needs, in part, by providing a system for reconnecting dropped calls. The system, when installed in a network, supports reconnection of dropped calls in which at least one user is a user of the network. The system subject of the present invention is generally denoted as RCS—Reconnection System. It should be understood that that term is only a convenient abbreviation

[0037] In accordance with one aspect of the invention, a system and method for facilitating reconnection of dropped calls is provided. After a call is detected as having dropped user A, the RCS keeps user B online by reporting the drop event and announcing that call reconnection is feasible. If user B remains online, the RCS performs a sequence of repeated call attempts for a pre-determined period until user A's terminal rings and user A answers the call. If user A answers within the pre-determined time period, he/she is bridged to user B. If not, the RCS preferably announces user B a voice message reporting the task failure. In one variation of this embodiment, while user B is waiting online, he/she is bridged to a “call setup voice channel”, through which he/she can hear about each call attempt performed by the RCS including the sound of the attempt itself, e.g., whether it encounters “user unavailable”, “user busy” or the ringback tone of user A's terminal. This provides user B a sense of orientation or awareness as to the status of the reconnection attempt and the chance of it materializing. In a second embodiment, user B does not have the call setup progression announced, but rather is held online (possibly hearing music) and is bridged to user A only once user A answers. In a similar embodiment, user B is held online until user A's terminal starts to ring and is then bridged to the “call setup voice channel”, i.e. hearing the ringback tones of user A's terminal.

[0038] The RCS service according to the present invention can provide inter-network capabilities if at least one of networks ANET and BNET is RCS-installed. Moreover, in a preferred embodiment, the user-experience of users A and B is independent of where the RCS is installed (ANET or

BNET), i.e., either user can have the same quality reconnection experience as long as an RCS service is on either ANET or BNET.

[0039] The call reconnection procedure according to the present invention can be appreciated as being superior in several aspects to the existing call reconnection systems described above.

[0040] First, in contrast to any call reconnection system described above in which calls are only reconnected if dropped on the network in which the system is installed, the RCS service reconnects inter-network calls that dropped for users outside the network in which it is installed (provided only that one party to the call is a subscriber of the RCS-installed network). This means that it is sufficient to have one party of a call as a subscriber to an RCS-installed network for a dropped call to be reconnected, regardless of where the call dropped.

[0041] The second aspect that makes the RCS according to the present invention superior over all state-of-the-art methods is its ease of implementation in existing networks (i.e., in networks whose MSCs, BSCs and terminals have not been originally designed to support dropped call reconnection functionality) as compared to the complexity of implementing all reconnection systems in the above-described existing networks. For the reasons noted above (see discussion of the Florkey et al. patent), implementation of known systems on existing networks requires extensive network installations and modifications, specifically on base stations controllers and sometimes on MSCs and terminals.

[0042] For implementing the RCS according to the present invention, however, there is absolutely no need for installation of any RCS modules or other component on the BSCs or on terminals. The RCS is essentially transparent to the BSCs and terminals, and in certain embodiments even to the MSCs. In other words, there is no functionality required from these entities additional to their existing (standard) functionality. This is viewed as a major advantage in implementing the RCS according to the present invention on existing networks.

[0043] An RCS according to the invention can be implemented on a computer having a processor, a memory, and a database for executing modules therein, the computer being in communication with either ANET or BNET. As will be elaborated hereinafter, the RCS can be implemented in various ways, such as code executable on an MSC or on a different processor (external to the MSC). For each dropped call, the RCS service flow can be roughly divided into three stages: (a) detecting the dropped call and determining its cause, (b) suspending call teardown in BNET, and (c) carrying out the reconnection. These stages are described under the headings that follow.

[0044] b) Detection of the Drop and Determination of the Cause

[0045] For reconnection of a dropped call, the very event of unintentional call termination (sometimes referred to herein as unintentional call disconnection) has first to be detected. Detecting that a call between user A and user B has dropped can be performed as a task as in the RCS which is connected to a network associated with at least one of the parties to the call. The RCS has a detection task module executing a detection task on a processor thereof and operative to receive the number identification of both users, receive a release message associated with the call attempt, and test the cause-indicator in the received release message to determine a match to a prescribed criterion.

[0046] Such detection is performed as a task as part of the functions of the RCS which is connected to a network associated with at least one of the parties to the call. The RCS routinely receives release messages from the MSC it is associated with, including information originating from the MSC of the network in which the call has been terminated or directed to. Release messages typically progress on the signaling path between MSCs, and provide indications from one MSC to another that, e.g., an active call has been terminated or that a call set-up attempt has been unsuccessful.

[0047] Release messages typically include one or more cause-indicators, or clearing codes. These indicators traditionally serve as a useful debugging tool for tracking and analyzing network malfunctions and singularities. One of the focal points of the present invention is exploiting these indicators for detecting dropped calls and incomplete call attempts. An example of the cause indicator codes is the ISDN Event Cause Codes, also known as Bellcore Standard ISDN Cause Codes or SS7 Cause Codes. These codes are associated with the Q.931 Standard.

[0048] As an example, Cause No. 16 signifies normal call clearing. This cause indicates that the call is being cleared because one of the users involved in the call has requested that the call be cleared, i.e. has pressed END. Obviously, when this cause is received during an active call a triggering of the reconnection task is unnecessary.

[0049] In a further example, Cause No. 38 signifies that a network is out of order. This cause indicates that a user's network is not functioning correctly and that this condition is likely to last a relatively long period of time. In this case, triggering the reconnection task is not likely to be successful. In a further example, Cause No. 41 signifies temporary failure. This cause indicates that the network is not functioning correctly and that the condition is not likely to last a long period of time. In this case, activating the reconnection task may be desired.

[0050] In a further example, Cause No. 54 signifies that called party has barred incoming calls. In this case, it is generally useless to activate the reconnection task since no incoming connection can be made to called party.

[0051] As such, a cause code generated by any call can be tested by code executing to implement this embodiment against values representing different causes for the call termination to determine whether the call termination has been intentional or unintentional. The detection task in the RCS in fact can parse the release message in a machine executing the RCS to obtain the cause-indicator, if necessary to obtain it. The detection task further tests the cause-indicator parsed from the release message to determine if it corresponds to a prescribed criterion, i.e. a cause-indicator indicative of unintentional call termination.

[0052] In certain implementations, it may be desired to allow selective triggering of the reconnection tasks, e.g., for different users. In these cases, an additional task can be implemented to access a database which contains data pertaining to users' preferences, e.g., whether they wish to be offered the reconnection service or not.

[0053] A considerable subset of dropped calls is the outcome of communication failures between the terminal and the base station. Automatic detection of a dropped call in a Base Station and transmitting it to an MSC is well known to those familiar in the art of cellular communication. For example, U.S. Pat. No. 6,343,216 by Kim et al. details the detection of the service impediment by the Base Station and trans-

mitting the information to the Mobile Switching Center. Those, as well as other possible detection methods, could be also used in conjunction with the embodiments of the present invention.

[0054] c) Suspending Call Teardown in BNET

[0055] A core aspect of the present invention concerns keeping user B online after a call has dropped on ANET. In normal conditions, following such a drop a release message issued from ANET would, upon reaching BNET, trigger the release of the resources in BNET that had been allocated to that call, in what is referred to as a "call teardown" procedure (the mirror image of a call setup procedure). Keeping user B online is done by suspending the call teardown procedure in network B. For this suspension to take place, at least a portion of the set of resources allocated to the call in BNET must be preserved. In one example, the voice trunk allocated to user B in dropped call is preserved and later utilized in the reconnection process. This portion of the set of resources preserved by the RCS in NET shall be hereinafter referred to as BRES. The terms "preserving a portion of resources in BNET", "suspension of call teardown in BNET" and "keeping user B online" will be hereinafter used interchangeably.

[0056] The RCS utilizes a call teardown suspension task module executing a call teardown suspension task on a processor thereof and operative to suspend call teardown in BNET by preserving a portion of resources allocated to the call. This suspension may include suspension of notification to one or more entities in BNET that the call has been determined as terminated in ANET. In ANET, however, the RCS permits the release of all resources allocated to the call.

[0057] Notably, there are differences in the way an RCS suspends call teardown when installed in ANET and the way it suspends call teardown when installed in BNET. These methods are described hereinafter.

[0058] According to one embodiment of the present invention, the RCS reconnects calls that have been dropped in the network in which it is installed (i.e. RCS is installed in ANET). Normally, when a call is dropped in ANET, a release message is sent from ANET to BNET informing that that the call has terminated. According to this embodiment, the release message from ANET is suspended short of being sent to BNET, so as to preserve user B's resources allocated to the call and suspend call teardown in BNET. Stopping the message can be implemented in several ways depending on the way the RCS is implemented in BNET, as described hereinafter.

[0059] One option is to have an RCS module implemented as code executing on MSCA (that is, the MSC of user A's network). This module can directly configure the MSCA to suspend the release message short of being issued to BNET, based on pre-set criteria according to an aspect of the present invention. Another option is to have an RCS module implemented as code executing on a different processor, such as a server associated with the MSCA allowing it to register to DISCONNECT events in the MSCA, i.e. to be invoked by MSCA whenever such events occur.

[0060] In a variation, the RCS "hooks on" to each call that is supported by the MSCA, e.g., as a third party in a conference call connected through user B. When the call drops on user A, since the RCS is still present as a party in the conference call, the release message sent by MSCA to MSCB will not trigger call teardown in BNET. Since the RCS is already hooked to a voice channel with user B, it can use the portion of the set of call resources, e.g., the voice trunk allocated to

user B, to announce to the reconnection offer and continue the reconnection process thereafter. As explained in the aforementioned “Dropped Call Re-establishment System with Inter-Network Capabilities” application, the notification and such other messaging can be implemented using a second communication channel in situations in which user B is using a “hybrid” constructed device.

[0061] According to another embodiment of the present invention, the RCS reconnects calls that have been dropped in the network other than the one in which it has been installed (i.e., when the RCS is installed in BNET). According to this embodiment, the RCS prevents the release message issued by MSCA from triggering call teardown in BNET. Preventing teardown can be implemented in several ways depending on the way the RCS is implemented, as described hereinafter.

[0062] One option is to have an RCS module implemented as code executing on MSCB. This module can directly configure the MSCB to suspend call teardown based on pre-set criteria according to an aspect of the present invention rather than performing an automatic release of resources. Another option is to have an RCS module implemented as code executing on a different processor, with connectivity to MSCB. In a variation of the foregoing, an RCS server which is associated with either ANET or BNET is notified of the setup of every ingoing and outgoing call via an INAP (Intelligent Network Application Part) IDP (Initial Detection Point) message. In response to such IDP message, the RCS is configured to always indicate to connect the call, and to register to DISCONNECT events. When this call is released, both MSCs involved generate a DISCONNECT event. This event invokes the RCS detection task which tests the cause indicator to determine a match to pre-defined criteria indicating a drop. If a match is found, call teardown is suspended by the RCS and the reconnection task is invoked. Otherwise, the call is released normally.

[0063] In a variation, the RCS “hook on” to each call that is supported by the MSCB, e.g., as a third party in a conference call connected through user B. When the call drops user A, since the RCS is still present as a party in the conference call the release message sent by MSCA to MSCB will not trigger call teardown in BNET. Since the RCS is already hooked to a voice channel with user B, it can proceed directly with announcing to user B a reconnection offer and continue the reconnection process thereafter.

[0064] Whichever approach is chosen for implementation, if the reconnection attempt does not succeed for a predefined period, the RCS acts to resume the call teardown in MSCB using conventional steps. It should be understood that the action of “preserving resources” by the RCS does not necessarily indicate that the preserved resources are in fact stored within the RCS or allocated to it or at any other way are within its reach. Rather, it only indicates that the RCS performs the suspension of their release. These resources can be, in fact, allocated or stored beyond the reach of the RCS, such as in a different network than the one in which the RCS is installed.

[0065] d) Carrying Out the Reconnection

[0066] The RCS has a reconnection task module executing a reconnection task on a processor thereof and is operative to carry out the reconnection. The reconnection task module is implemented either in ANET or in BNET, or in an external module connected to one of these networks. In any of these implementations, the reconnection task module performs the following steps, which are standard actions in a cellular network well known to a person skilled in the art.

[0067] After having kept user B online the RCS utilizes a standard IVR module to announce to user B that call reconnection is feasible and verify that he/she has remained online. If user B has indeed remained online, the RCS carries on to initiate a call set-up attempt to user A. Several options exist for proceeding.

[0068] In one implementation, the RCS initiates a call setup attempt to user A which includes establishing a call setup voice channel to user A’s terminal, i.e. a voice channel that is heard in the caller’s terminal just after a standard call dial to user A takes place. The RCS then bridges user B to the call set-up channel. From user B’s perspective, he/she is hearing what he/she would be hearing when dialing user A manually. If user A answers, the users’ voice channels are already bridged and they can resume their conversation. If for any reason the reconnection attempts to user A have failed, the RCS sounds a voice message reporting the task failure. This embodiment provides user B a sense of orientation of the progression of reconnection task and the chance of it materializing.

[0069] Bridging user B to user A or to the call setup voice channel may be performed just as it would be done in commencing a standard conference call involving three human users. A first user (corresponding to the RCS) and a second user (corresponding to user B) are on an active call, and try to add a third user (corresponding to user A) to the session. When the first user (RCS) calls the third user (user A), it is up to the first user to decide when to bring online the second user (user B)—whether at the call setup stage or only after the third user has answered. In any case, after both the third user and the second user are brought online, a three-way session takes place, i.e. all three can hear each other.

[0070] In a second implementation, user B is held online and is bridged to user A only once user A answers the call. Here too, if for any reason the reconnection attempts to user A have failed, the RCS sounds a voice message reporting the task failure.

[0071] After users A and B are online, the RCS can preferably perform “call dropback”, which is a technique well-known in the art for “stepping out” of a three-way connection and leaving the other two users connected to each other. The call dropback method is favorable since it releases RCS resources very soon after the call is commenced instead of keeping them committed during the call.

[0072] In cases in which users A and B are of the same RCS-installed network, the implementation is a particular case of any of the above-described inter-network implementations. However, when ANET and BNET are distinct RCS-installed networks, co-ordination is required between the two RCS systems in order to avoid a situation in which both systems would independently attempt to reconnect the dropped call.

[0073] e) Technical Implementation

[0074] FIG. 1a provides a schematic description of the high-level architecture of the cellular system connecting (RCS-installed) terminal X and non-RCS installed terminal Y. Network XNET (102) serving user X consists (among others) of the following modules and subsystems: an HLR (106), several mobile switches (MSC), and a gateway switch (114). A RCS module (112) is introduced according to the present invention (112). These entities are physically interconnected and communicate with each other via SS7 signaling (108). The MSC serving terminal X communicates with the serving mobile base station (118) serving terminal X. The

mobile base station communicates with terminal X by RF communication. XNET is connected by a gateway switch (114) to an internetworking network (116) through which it is connected to YNET (122) which is serving terminal Y. YNET consists of basically the same elements as XNET apart from the RCS which is not present in this network. The MSC serving terminal Y (124) communicates with the mobile base station (128) which communicates with terminal Y (130).

[0075] The arrangement of FIG. 1a, therefore, provides a description of the most general case when a communication session is to be made between a network which is provided with a RCS and a network which is not provided such system. As will be appreciated by persons of ordinary skill in the art, the blocks illustrated in FIG. 1b (and throughout the figures) can be implemented as modules that comprise code that, upon execution, configure a processor to perform the functions and interactions described herein.

[0076] FIG. 1b provides a schematic description the connectivity of the RCS system with several important external modules, in the most general case. The RCS communication with the MSC signaling side (104a) is handled in ISUP/INAP via SS7 network, whereas different queries and requests are sent from the RCS to the MSC and indications and responses are sent back from the MSC to the RCS. Additionally, the RCS uses voice trunks to convey to the MSC voice side (104b) messages to be announced to users.

[0077] Preferably, RCS signaling is based upon the INAP protocol (the Intelligent Network Application Part (INAP) is a signaling protocol used in the Intelligent network architecture). Preferably, for CDMA/TMDA installations popular in North America, the RCS will use the WIN version (Wireless Intelligent Network). The INAP/WIN protocols will be used to monitor calls as they progress through the network, identify calls which have been dropped, and trigger the RCS's unique functionally if a dropped call has been detected.

[0078] In another example, a protocol for GSM is called MAP (Mobile Application part). IS41 and MAP will be used for handling handoffs, SMS deliveries. ISUP (ISDN) user part will be used by Intelligent peripherals (IP) in order to set up or release calls, as well as play voice prompts.

[0079] As for the specific hardware used for implementing the RCS, various configurations can be used depending heavily on the system requirements and the specific characteristics of the network in which it is implemented. By way of example, two such configurations are hereinafter described. (a) A series of stand-alone servers, e.g., HP Proliant 380G6 or equivalent Servers, which are NEBS compliant and provide High Performance in a 2U or 4U 19" rack-mount packaging. Each such server is fitted with either AudioCodes IPM260 8E1 cards for dealing with ISUP and Voice (i.e. prompts), or Ulticom SIGNALWARE boards and protocol stack, for INAP/WIN and TCAP. The actual number of cards depends on system load, realtime requirements, site characteristics, system design and performance. (b) A system based on a 19" Card cage that includes CPU blades. This form factor is often called COMPACT PCI, and is provide by vendors as e.g., HP, PTI and many others. The system also consists of AudioCodes IPM260 16E1 BLADE, or the PTI 24E1. Additionally, Ulticom SIGNALWARE boards and protocol stack, for INAP/WIN and TCAP. Configuration (b) is far more useful for large single sites, and is also preferable for supporting high system performance.

[0080] For each dropped call, the RCS service is executed by several tasks, such as detection of dropped calls, system

dialing, bridging users, etc. In systems which operate at highly dynamic environments it is generally advantageous to implement an architecture in which a task manager monitors and controls the execution this multitude of tasks. The task manager will, e.g., spawn tasks according to predefined events, such as spawning detection tasks in the events of receiving terminated-call data from an MSC. In another example, the task manager utilizes a timer module to control the termination or abortion of outdated tasks. The task manager can also be utilized to control the RCS performance at times of enhanced demand for call reconnections, such as in peak hours of network usage. The task manager can handle such situations by different schemes of dynamic resource allocations.

[0081] Each of the modules and other components that comprise the RCS, whether hardware or code, can be integrated within the MSC of a telephone service provider. Integration of the RCS can include co-location of the equipment or even installation of one or more of the modules on a server that operates the MSC or that is on the same local network as the MSC, for example, on the same LAN segment that has the MSC.

[0082] f) Concluding Comments Regarding Overview

[0083] In both of the reconnection embodiments described above, neither user actually dials since one user remains online and the other answers an incoming call. This characteristic enables flexibility in billing and allows various billing schemes for the reconnected call.

[0084] Outgoing calls from the RCS to user A must detect with good reliability the outcome of the dialing to user A, such as being answered by a person, a fax, an answering machine, getting a busy signal or a Ring/No Answer signal. For that end, at least two types of technological methods can be utilized: (a) Legacy predictive-dialer technology, which monitors and analyzes the voice channel traffic to differentiate between audio signals typical of, e.g., faxes, answering machines and an actual person. (b) Technology for monitoring the signaling traffic on the SS7, i.e., detecting a response message from dialed user's MSC propagating on the signaling channel. In either case, if an actual person answers the call, the RCS is typically informed of it shortly after the occurrence. The signaling monitoring method typically enables a more robust detection as well as more precise information as to the reason call attempt has failed, which can be used for optimizing dialing schemes. This technique is taught in a co-pending U.S. application Ser. No. TBA [Attorney Docket No. 3267/0851-US3], filed on even-date herewith, entitled "Availability Notification System with Inter-Network Capabilities", which is hereby incorporated by reference in its entirety.

[0085] A possible feature in all reconnection methods is that when calling user A, the RCS configures the CallerID of the call initiated to user A to be the number of user B. Alternatively, an alphanumeric string can be used to indicate to user A that an RCS-initiated call is coming in. The advantage of this feature is to provide user A a chance to decline the reconnection offer even before answering the call.

[0086] An advantageous feature in the present invention is conditioning the activation of the reconnection task on data representative of an approval from user A and/or user B. Querying a database storing such data can be performed before or after a detection task has been executed. The database can include a subscriber table storing subscriber data of users A and/or B.

[0087] Notably, the present invention can also be utilized to reconnect dropped calls that occur between wireline terminals and wireless terminals, typically when the wireline terminal is user B.

2) Detailed Discussion of Certain Components and Embodiments

[0088] The system for reconnecting dropped calls can be installed in a cellular network, e.g., as a dedicated server or as a module added to existing servers used for performing other tasks. For brevity, this system will be hereafter referred to as RCS—Reconnection System.

[0089] FIG. 2 provides a schematic description of the user experience after each occurrence of a dropped call. Users A and B are on a call (step 210) when the call is dropped in ANET (step 212). User B then hears a reconnection offer “Call dropped. Please remain online for reconnection attempt” (step 214). If user B remains online (step 216), and if user A is back to availability by that time, user A’s terminal rings (step 218). When user A answers, users are connected and can resume their conversation.

[0090] FIG. 3 provides a schematic description of the stages performed by the RCS after each occurrence of a dropped call. The sequence of steps is hereby elaborated.

[0091] Users A and B are on a call (step 310) using network resources ARES and BRES in ANET and BNET respectively. When the call is dropped on ANET (step 312), ARES are automatically released by ANET (step 312). According to the present invention, identification of the dropped call can be performed either by an external entity and then streamed to the RCS system (step 314) and/or directly by the RCS system itself (step 316). At either case, the dropped call data such as users’ number identities are input to the RCS.

[0092] The RCS then issues a request to the MSC serving user B in BNET to suspend the release of BRES in BNET (step 318). This means that the standard call teardown process of the dropped call (which is normally triggered by the drop in ANET), including the release of BRES, is suspended, and user B is left online, i.e. connected to the voice channel. Next, the RCS announces to user B (through same voice channel) a reconnection offer (step 320) and resets Timer to zero (step 322). Then the RCS tests whether user B has remained online (condition 324). If not, BRES are released (step 325) and the reconnection task is aborted. If user B has remained online, a call is initiated by the RCS to user A and user B is bridged (using preserved resources BRES) to the “call setup voice channel” (step 326). This means that user B hears the exact same sounds that he/she would hear when making an ordinary call attempt to user A. Then, a check is made whether the call got through to terminal A (condition 328).

[0093] If the call did get through to terminal A, then a check is made whether user A has answered the call within a pre-set time period T2 of the phone starting to ring (condition 330). If A has answered within T2, then users A and B can hear each other and resume the conversation (step 332). Advantageously, the RCS then performs “call dropback”, leaving users A and B online and terminating the task. If user A has not answered within T2, resources BRES are released (step 338) and the reconnection task aborts, advantageously notifying user B of the abortion (step 340).

[0094] If the call didn’t get through to terminal A a check is made whether Timer has passed a pre-set time limit T1 (condition 336). If YES, resources BRES are released (step 338) and the reconnection task aborts, advantageously notifying

user B of the abortion (step 340). If Timer hasn’t yet reached predefined time limit T1, after a possible pause of a predefined time period ΔT (step 334) the system reverts back to step 324.

[0095] In this embodiment, since user B is bridged to the call setup channel he/she is at all times informed of the status of the reconnection process, e.g., such as when the RCS encounters a BUSY signal, a Ring/No answer situation, or a “user unavailable” message. This provides user a sense of control on the reconnection process.

[0096] Such conditions can be processed by code executing in at least one module or in a hardware component connected to the telephony system and constructed in accordance with the invention that implements the logical tests and initiates actions or further tests in response to the results

[0097] In the described embodiment, the stage of detecting the dropped calls and screening them from all other intentionally-disconnected calls is done at the MSC/SSP or at lower-level network modules. The call records received at the RCS (step 314) are only of dropped calls. However, other methods may be applied for screening dropped calls. In one example, all terminated calls are streamed to the RCS and the RCS detects which of them are dropped calls. The latter method is superior to the former one in the aspect that it does not require changes to core network elements. However, the latter method is inferior to the former in the aspect that it requires hauling and processing of considerable amounts of data.

[0098] FIG. 4 provides a schematic description of the stages performed by the RCS after each occurrence of a dropped call in another embodiment. The sequence of steps is hereby elaborated.

[0099] This embodiment is similar to the one described in FIG. 3. The only difference is that in this embodiment, after user B remains online he/she is not immediately bridged to the “call setup voice channel”, but rather held online (e.g. hearing music, step 400) and only bridged if and when user A answers the call (step 402). The advantage of this embodiment is that user B is provided a friendly and smooth user experience in which he/she is not bothered by the different tones and beeps that are heard on the call setup voice channel in the course of the reconnection process—such as ringback tones, busy tones, and dial tones.

[0100] FIG. 5 provides a schematic description of the signaling traffic involved in the process of reconnecting users A and B following a dropped call, according to the service flow as described in FIG. 4. User A’s terminal is represented by vertical line 502, and so is user B’s terminal (504). The switching entities MSC/SSP (Mobile Switching Center/Service Switching Point) in both A and B’s serving networks are collectively represented by vertical line 506. The assumption behind this aggregation is that these entities communicate between themselves continuously and share all relevant data so that data arriving at one entity can be assumed to be present at the second within a short time. Finally, the RCS itself is represented by vertical line 508. Users A and B are on a call (step 510) when the call is dropped on ANET. The occurrence is notified to the MSC/SSP of BNET (step 512) and then a report is sent from the MSC/SSP to the RCS (step 514) with the following data: a call drop has occurred, which serving network dropped the call, A’s number, B’s number, and the time of the drop.

[0101] The RCS then issues a request to BNET to suspend call teardown in BNET, i.e. preserve BRES which were used to support the dropped call (step 516). Then, the RCS

announces to user B a message offering him/her to remain online for the reconnection service (step 518). User B remains online, and the RCS is implicitly notified of it when for several seconds (e.g. 3) there is no call termination report received from terminal B. Then, the RCS attempts to set up a call with user A. A first call setup attempt (step 520) fails when the MSC cannot reach terminal A (step 522) and a report is sent back to the RCS (step 524). A second call setup attempt to A (step 526) succeeds and a call request is forwarded to user A's terminal (step 528). When user A answers the call a report goes back to the MSC (step 530), which sends a further report to the RCS that user A has answered and is now online (step 532). The RCS then bridges user A and user B (step 534), e.g., as in a "conference call" with three parties (users A and B and the RCS), and then preferably performs a "call dropback" in order to leave only A and B online (step 536).

[0102] In certain special cases, the redial sequence could be extended for a longer period than the normal reconnection attempts. This could apply for a small fraction of the dropped calls, such as in the case of dropped emergency calls. In this respect, one could apply the teachings of US Patent Application 20070121852 to Taylor et al. titled "Method and System for User Prioritization within Telecommunication Services and in Particular within Call Completion Services", which is hereby incorporated by reference in its entirety. In US Patent Application 20070121852, a concept is suggested for prioritizing subscribers in telecommunication systems by providing these subscribers preferred allocation of network resources. Various criteria of prioritization could apply, including the ones mentioned above.

[0103] FIG. 6 provides a timeline perspective of a reconnection embodiment according to the present invention. The vertical axis (610) represents four states that a terminal can be in—unavailable (618), idle (i.e. available and not in use, 616), ringing state (614) and online (i.e. on an active call, 612). The horizontal axis (620) is a time axis. Terminal B is represented by a dotted line (630), and terminal A is represented by a continuous line (628). After a call drops on user A (T1), the reconnection task according to the present invention is triggered. User B is immediately announced a message advising him/her to remain online for a reconnection attempt (T2). After user B remains online (T3), a first call attempt is initiated automatically by the RCS to user A (T4) which turns out to be futile since user A is unavailable. User B remains online, and shortly after user A returns to availability. The second automatic call attempt by the RCS rings user A's phone (T6), and when user A answers (T7) the call is reconnected and users A and B can resume their conversation.

[0104] FIG. 7 provides a timeline perspective of a combined embodiment, in which first a reconnection task according to the present invention is tried, and when unsuccessful, a call re-establishing task as taught in a co-pending U.S. application Ser. No. TBA [Attorney Docket No. 3267/0851-US1], filed on even-date herewith, entitled "Dropped Call Re-establishment System with Inter-Network Capabilities", is automatically triggered.

[0105] After a call drops on user A (V1), the reconnection embodiment according to the present invention is triggered. User B is immediately announced a message advising him/her to remain online for a reconnection attempt (V2). After user B remains online (V3), two consecutive call attempts are made by the RCS to user A (V4,V5) which turn out to be futile. Once the second call attempt fails, user B is announced

a message "reconnection attempt failed. Please hang up and you will be notified once user A returns to availability." (V7). Thereafter, the HLR-based re-establishment task according to co-pending U.S. application Ser. No. TBA [Attorney Docket No. 3267/0851-US1] is automatically triggered: user A's availability is monitored and once he/she turns available (V8) and detected as such (V9), user B is rung. Once user B answers, he/she is requested to remain online (V11). If user B remains online (V12) user A is called (V13) and when user A answers (V14), the call is re-established and users A and B can resume their conversation.

[0106] While the invention has been described in connection with a certain embodiment thereof, the invention is not limited to the described embodiments but rather is more broadly defined by the recitations in the claims below and equivalents thereof.

We claim:

1. A method for reconnecting a dropped call between two parties in a communication system as a result of a loss of one of a plurality of connections therebetween which constitute the call, wherein the call is supported by a set of resources allocated to the call by the entities of both parties' networks, comprising the steps of:

a. detecting that one connection of the call has been unintentionally-disconnected using a reconnection system (RCS) connected to a network associated with at least one of the parties to the call, the RCS having a detection task module executing a detection task on a processor thereof and operative to:

1. receive the number identifications of the two parties;
2. receive a release message associated with the call, said release message originating in an MSC of the network that initiated the call termination, the release message including data that identifies the parties to the call and includes one or more cause-indicators; and
3. test the cause-indicator in the received release message in a machine executing the RCS to determine a match to a prescribed criterion; and

b. in the event of the match, the RCS performing the step of:

1. preserving a portion of the set of resources allocated to the call by the entities in a network of user B, who is the party which was not unintentionally disconnected from the network, while permitting release of all other resources allocated to the call; and
2. executing a reconnection task within the RCS using the reconnection task module,

wherein the reconnection task executes to initiate a call set-up request with user A, who is the party that has been unintentionally disconnected from the network, and to repeat the call-set up request for a predefined time limit until user A's terminal rings, and

wherein, upon A answering the call, a reconnection is made using at least the preserved portion of the set of resources allocated to the call in the network of user B.

2. A method as in claim 1, wherein if user A's terminal starts ringing within the predefined time limit and user A answers the call, the reconnection task executes to bridge user A and user B using at least the preserved portion of the set of resources allocated to the call in the network of user B.

3. A method as in claim 1, wherein the reconnection task executes to bridge user B to a call setup voice channel to user A after it is established, and, if user A answers the call, to keep

user B bridged to user A using at least the preserved portion of the set of resources allocated to the call in the network of user B.

4. A method as in claim 1, wherein the preserving step comprises suspending any notification to the entities comprising the network of user B that the call has been determined as terminated in the MSC serving user A.

5. A method as in claim 1, including the additional steps of:

- a. testing for any release message from the MSC serving user B to determine whether user B has hung up the call; and
- b. if user B remains on the call, sounding a voice message to user B advising to remain online.

6. A method as in claim 1, wherein user A and user B are subscribers of the same network.

7. A method as in claim 1, wherein user A and user B are subscribers of different networks.

8. A reconnection system (RCS) for reconnecting a dropped call between a user A who has been disconnected from a communication network and a user B who is the other party to the call, the disconnection being a result of a loss of at least one of a plurality of connections between user A and user B which together constitute the call, wherein the RCS is connected to the communication network and is associated with at least one of user A and user B, and wherein the call is supported by a set of resources allocated to the call by respective switching networks associated with each of user A and user B, comprising:

- (a) a computer having a memory and a processor for executing modules therein, the computer being in communication with the communication network;
- (b) a database;
- (c) a detection task module loaded into the memory of the computer and executing at least one detection task in the processor, the detection task configuring the processor to cause the computer to:
 - i. load into the memory number identifications of user A and user B which are provided by the communication network;
 - ii. load into the memory a release message associated with the call which is provided by the communication network, the release message including one or more cause-indicators; and
 - iii. test the cause-indicator in the release message to determine a match to a prescribed criterion stored in the database;
- (d) a call-teardown-suspension-task module loaded into the memory of the computer and executing at least one call-teardown-suspension-task in the processor in response to the test by the detection task returning the match, the call-teardown-suspension-task configuring the processor to cause the computer to preserve a portion of the set of resources allocated to the call by the switching network of user B while permitting release of all other resources allocated to the call;
- (e) a reconnection task module loaded into the memory of the computer and executing at least one reconnection task in the processor in response to the test by the detection task returning the match, the reconnection task configuring the processor to cause the computer to:
 - i. initiate a call set-up request with user A;
 - ii. monitor whether user A's terminal has started to ring based on data provided by the communication network; and

- iii. if user A's terminal has started to ring within a first pre-defined time limit, then monitor whether user A answers the call-set up request based on data provided by the communication network and, if user A answers the call within a second pre-defined time limit, then bridging user A and user B using at least the portion of the set of resources allocated to the call in the network of user B;

- (f) a task manager loaded into the memory of the computer and executing in the processor so as to configure the processor to monitor the execution of up to a multiplicity of detection tasks, call-teardown-suspension-tasks and reconnection tasks.

9. A system as in claim 8, wherein the task manager further configures the processor to launch at least one of the call-teardown-suspension-task and the reconnection task on the condition that the RCS has access to data representative of an approval from at least one of user A and user B to execute the reconnection task.

10. A system as in claim 8, wherein the task manager tests whether the database contains data representative of the approval of at least one of user A and user B prior to the detection task being executed.

11. A system as in claim 8, wherein the database includes a subscriber table and wherein at least one of user A and user B have subscriber data stored in the subscriber table.

12. A system as in claim 8, wherein the RCS is part of a conference call with user B and wherein the reconnection task further configures the processor to cause the computer to announce that the reconnection is feasible using the portion of the set of resources that has been preserved from the call.

13. A system as in claim 8, wherein preservation of the portion of the set of resources by the call-teardown-suspension-task comprises configuring the processor to suspend any notification to the entities comprising the network of user B that the call has been determined as terminated in the MSC serving user A.

14. A system as in claim 8, wherein the switching network associated with at least one of user A and user B is a mobile switching center.

15. A reconnection system (RCS) for reconnecting a dropped call between a user A who has been disconnected from a communication network and a user B who is the other party to the call, the disconnection being a result of a loss of at least one of a plurality of connections between user A and user B which together constitute the call, wherein the RCS is connected to the communication network and is associated with at least one of user A and user B, and wherein the call is supported by a set of resources allocated to the call by respective switching networks associated with each of user A and user B, comprising:

- (a) a computer having a memory and a processor for executing modules therein, the computer being in communication with the communication network;
- (b) a database;
- (c) a detection task module loaded into the memory of the computer and executing at least one detection task in the processor, the detection task configuring the processor to cause the computer to:
 - i. load into the memory number identifications of user A and user B which are provided by the communication network;

- ii. load into the memory a release message associated with the call which is provided by the communication network, the release message including one or more cause-indicators; and
- iii. test the cause-indicator in the release message to determine a match to a prescribed criterion stored in the database;
- (d) a call-teardown-suspension-task module loaded into the memory of the computer and executing at least one call-teardown-suspension-task in the processor in response to the test by the detection task returning the match, the call-teardown-suspension-task configuring the processor to cause the computer to preserve a portion of the set of resources allocated to the call by the switching network of user B while permitting release of all other resources allocated to the call;
- (e) a reconnection task module loaded into the memory of the computer and executing at least one reconnection task in the processor in response to the test by the detection task returning the match, the reconnection task configuring the processor to cause the computer to:
 - i. initiate a call set-up request with user A that establishes a call setup voice channel; and
 - ii. bridge user B to the call setup voice channel using at least the portion of the set of resources allocated to the call in the network of user B; and
- (f) a task manager loaded into the memory of the computer and executing in the processor so as to configure the processor to monitor the execution of up to a multiplicity of detection tasks, call-teardown-suspension-tasks and reconnection tasks.

16. A system as in claim **15**, wherein the task manager further configures the processor to launch at least one of the call-teardown-suspension-task and the reconnection task on the condition that the RCS has access to data representative of an approval from at least one of user A and user B to execute the reconnection task.

17. A system as in claim **15**, wherein the task manager tests whether the database contains data representative of the approval of at least one of user A and user B prior to the detection task being executed.

18. A system as in claim **15**, wherein the database includes a subscriber table and wherein at least one of user A and user B have subscriber data stored in the subscriber table.

19. A system as in claim **15**, wherein the RCS is part of a conference call with user B and wherein the reconnection task further configures the processor to cause the computer to announce that the reconnection is feasible using the portion of the set of resources that has been preserved from the call.

20. A system as in claim **15**, wherein preservation of the portion of the set of resources by the call-teardown-suspension-task comprises configuring the processor to suspend any notification to the entities comprising the network of user B that the call has been determined as terminated in the MSC serving user A.

21. A system as in claim **15**, wherein the switching network associated with at least one of user A and user B is a mobile switching center.

* * * * *