

1/51

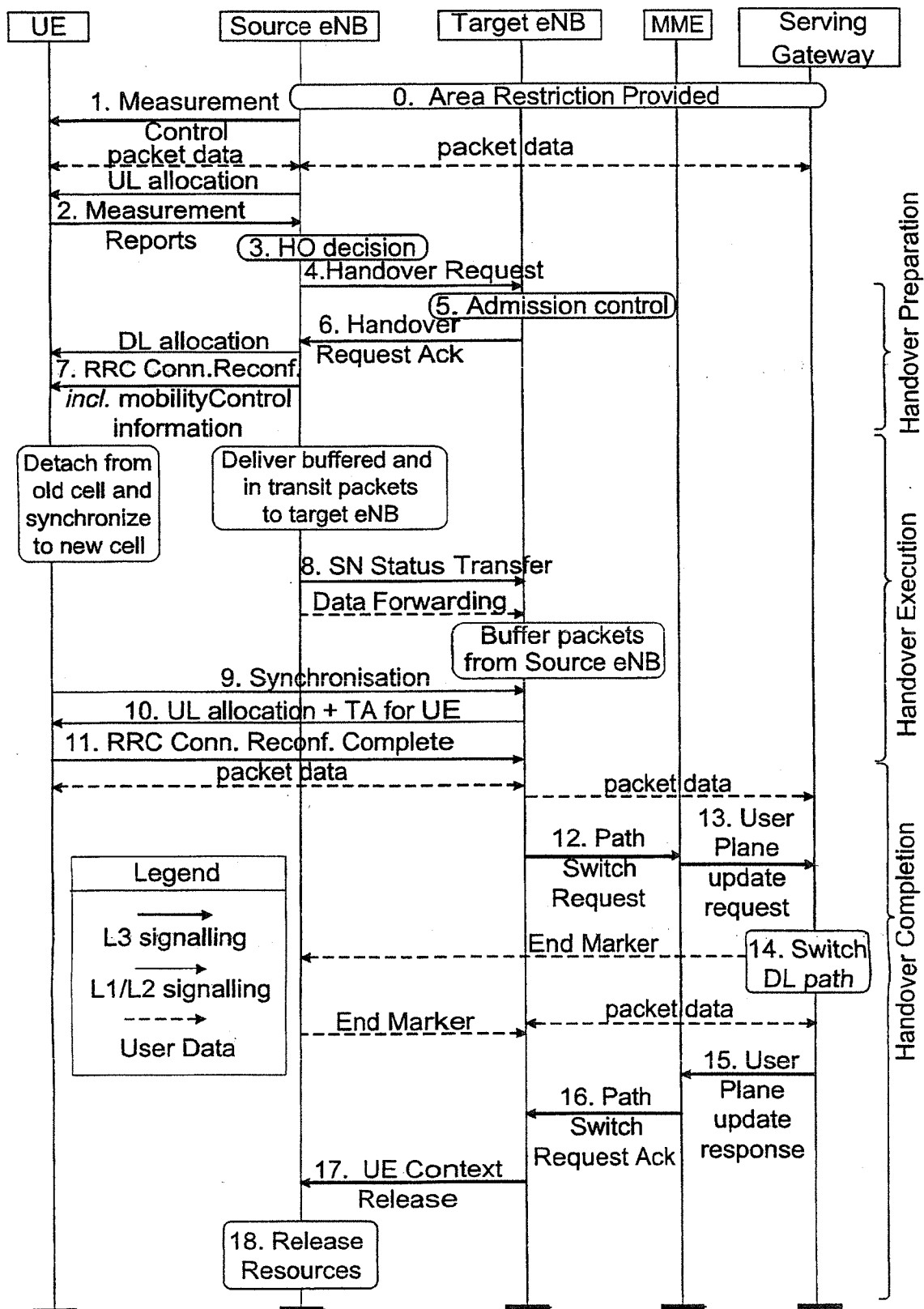
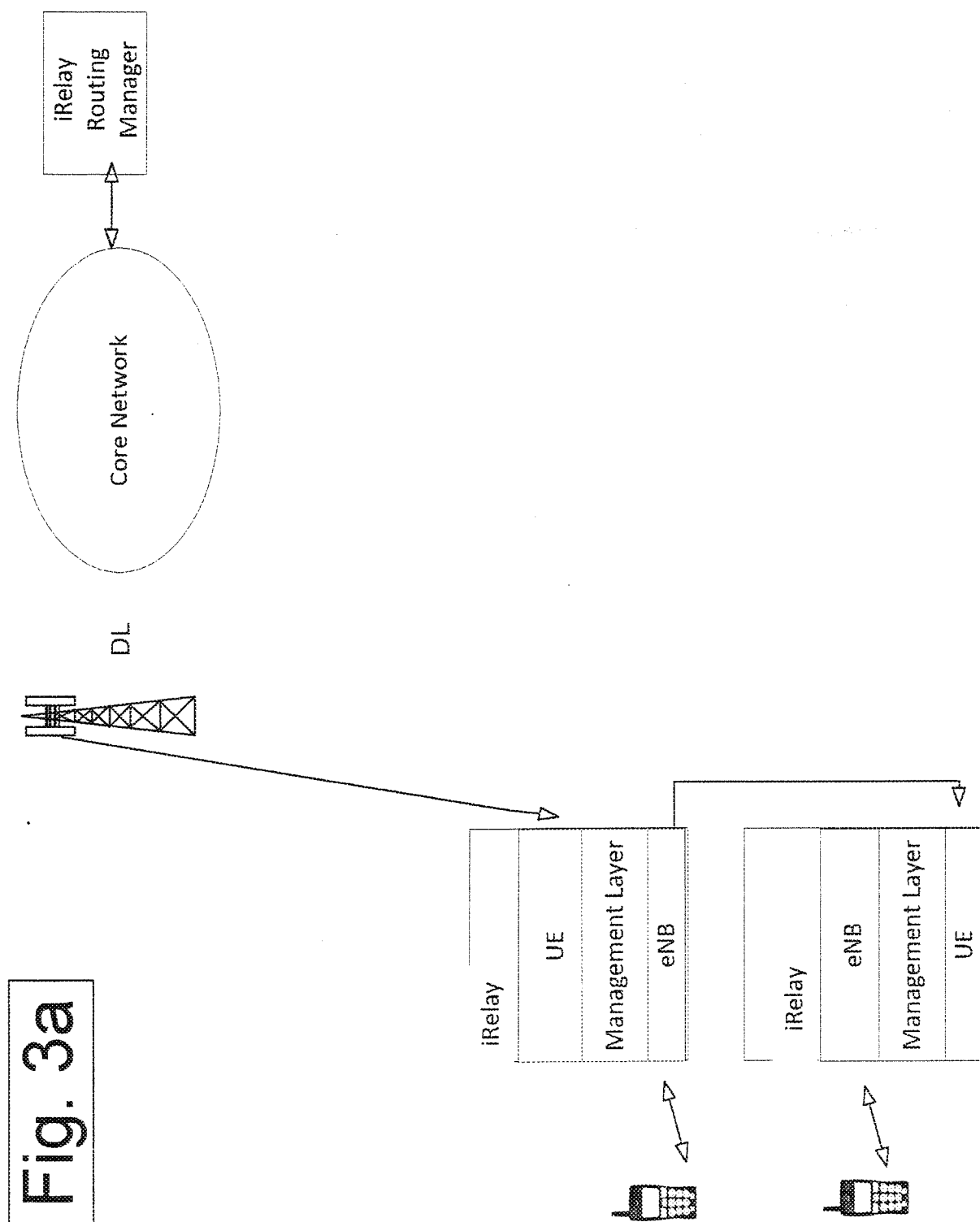


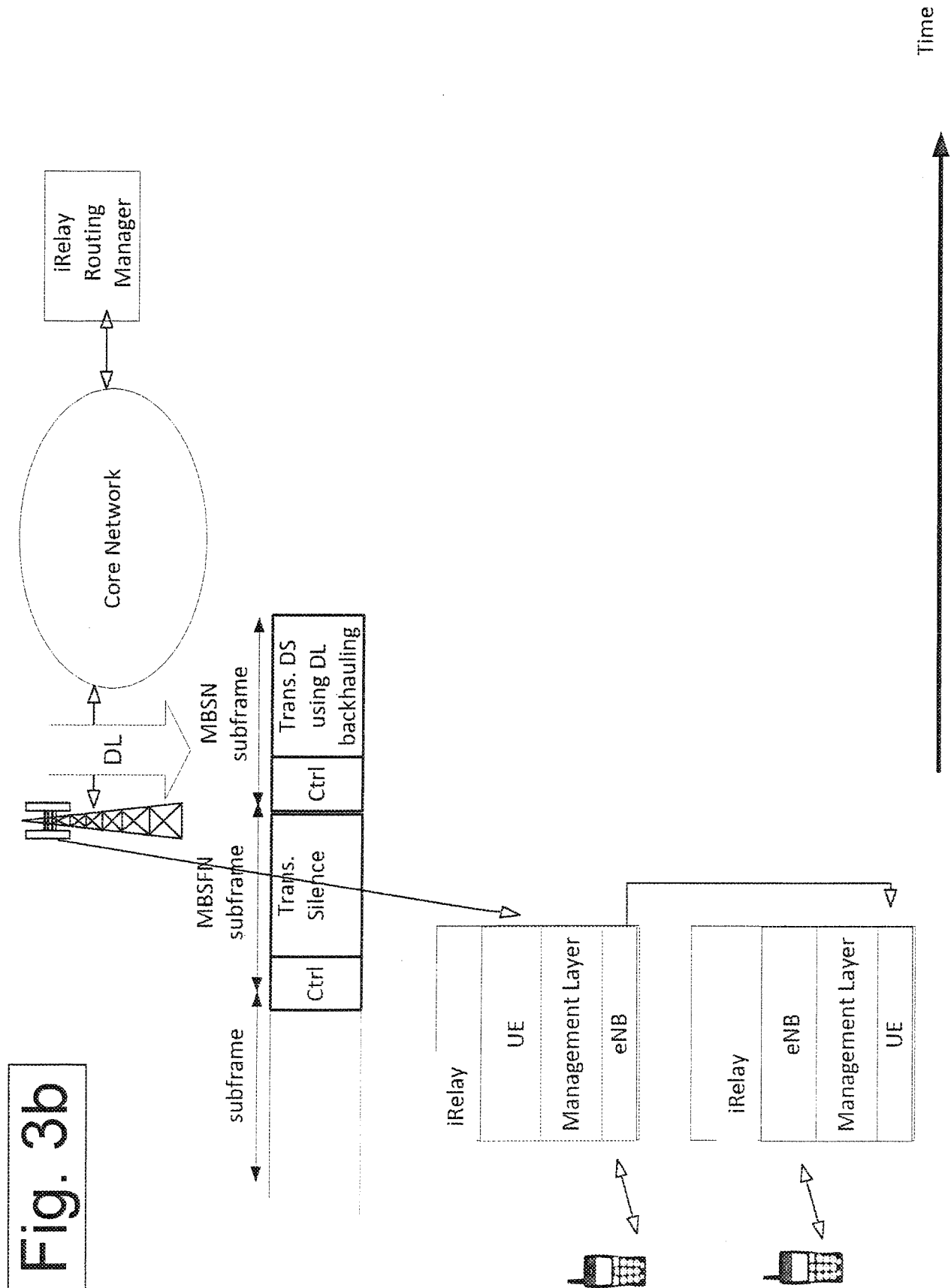
Fig. 1 (PRIOR ART)

FDD	0	1	2	3	4	5	6	7	8	9
	X	MBSFN Enabled	MBSFN Enabled	MBSFN Enabled	X	X	MBSFN Enabled	MBSFN Enabled	MBSFN Enabled	X
TDD	0	1	2	3	4	5	6	7	8	9
	X	X	X	MBSFN Enabled	MBSFN Enabled	X	X	MBSFN Enabled	MBSFN Enabled	X

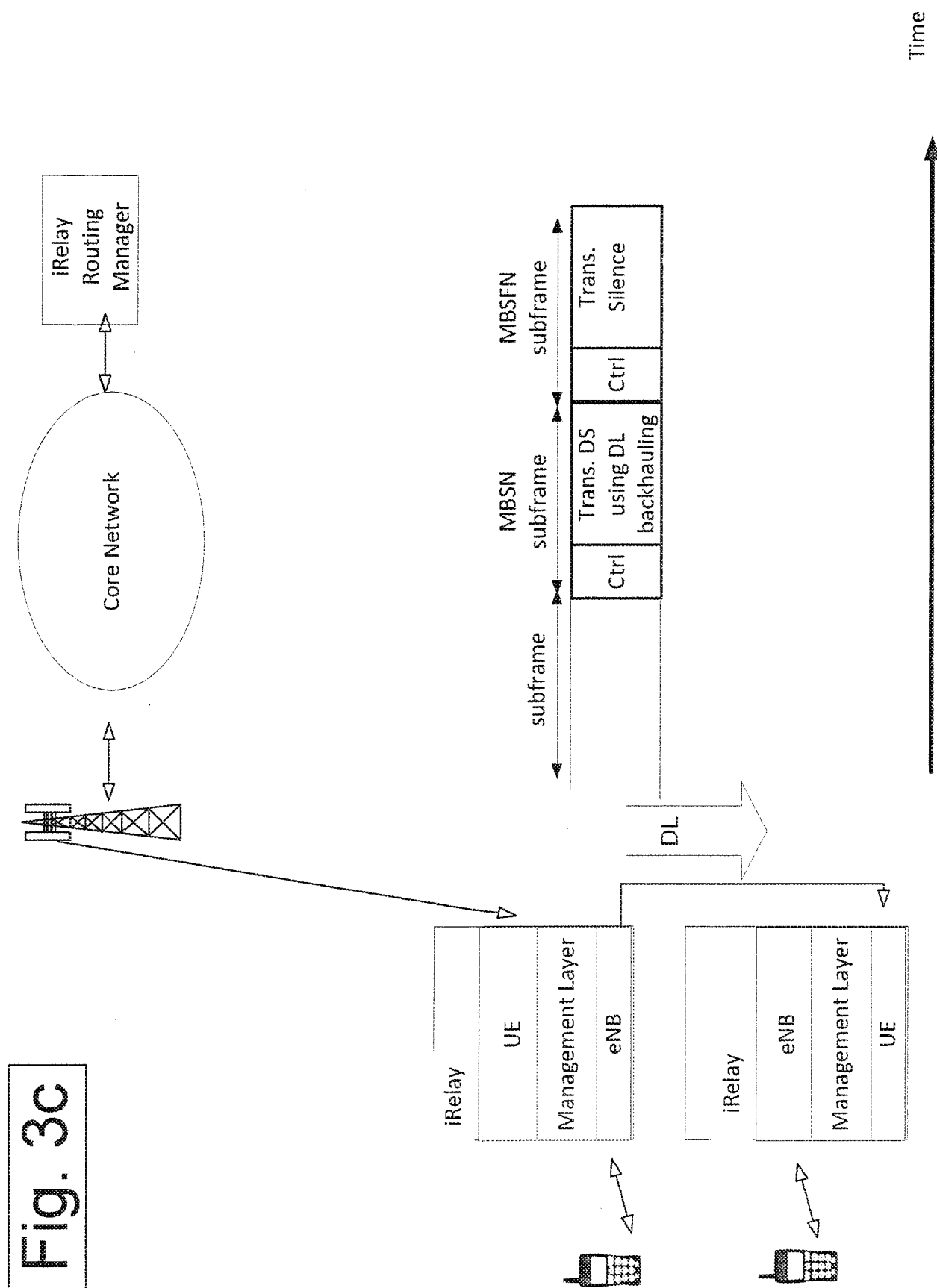
Fig. 2 (PRIOR ART)

3/51



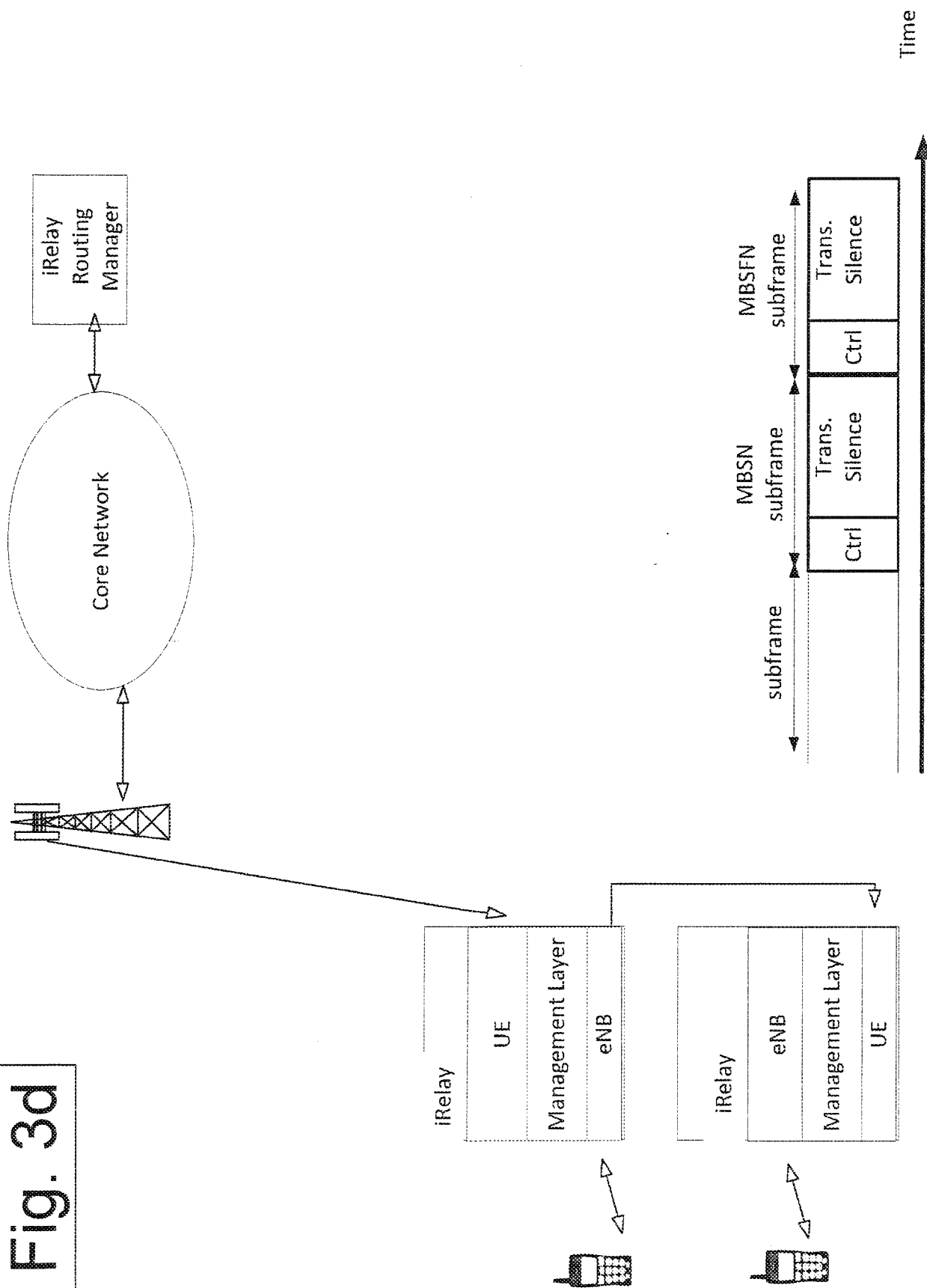


5/51



6/51

Fig. 3d



7/51

Fig. 4a

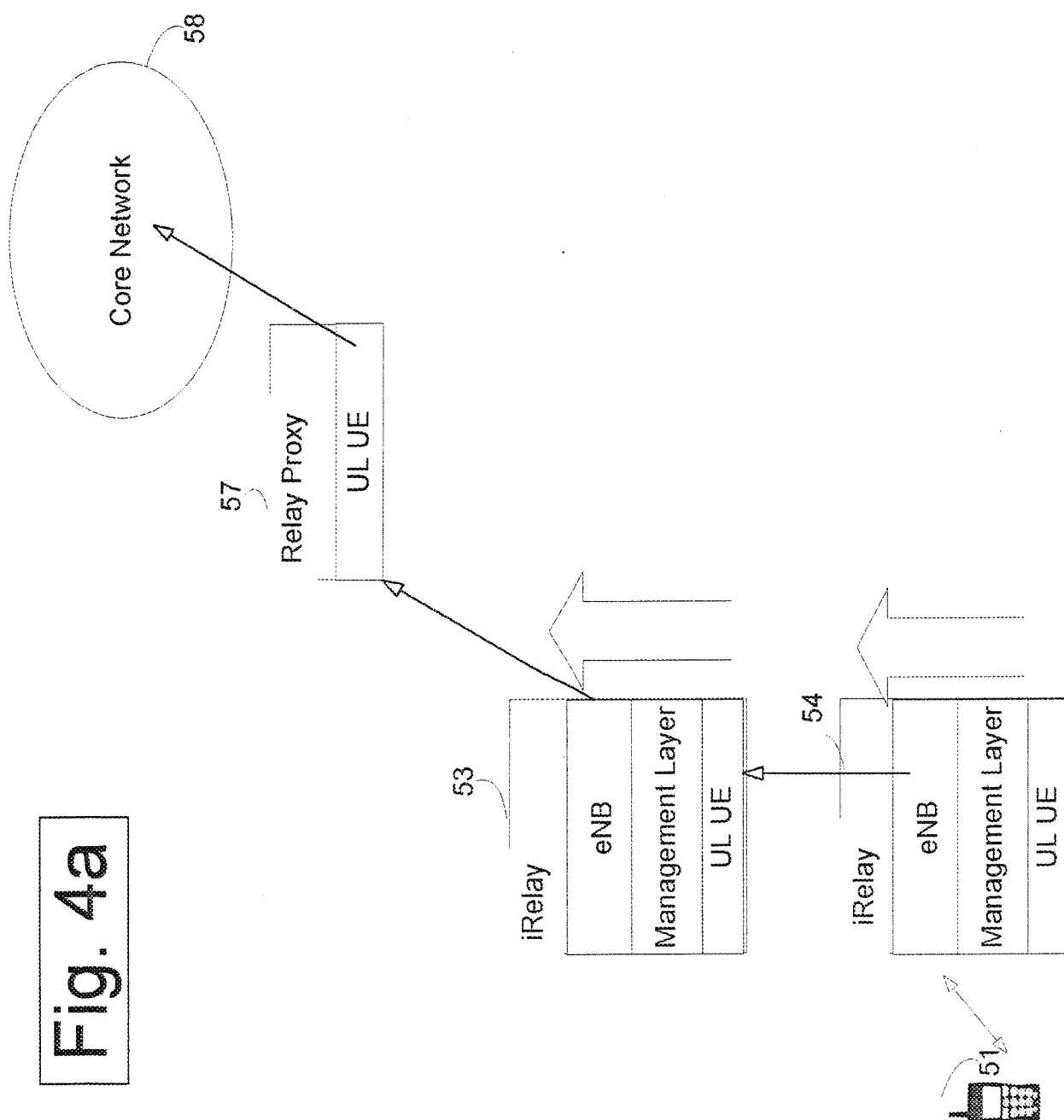


Fig. 4b

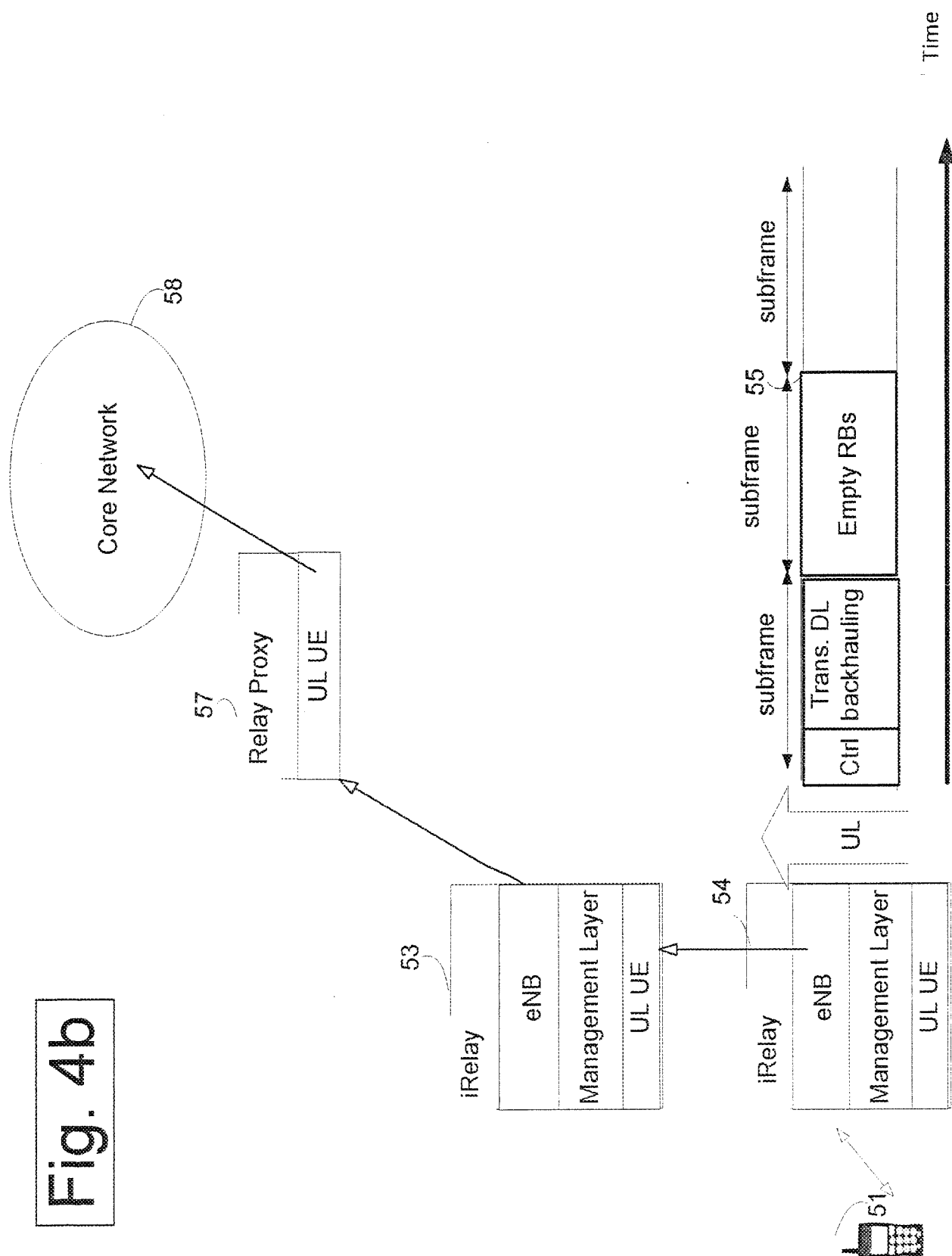
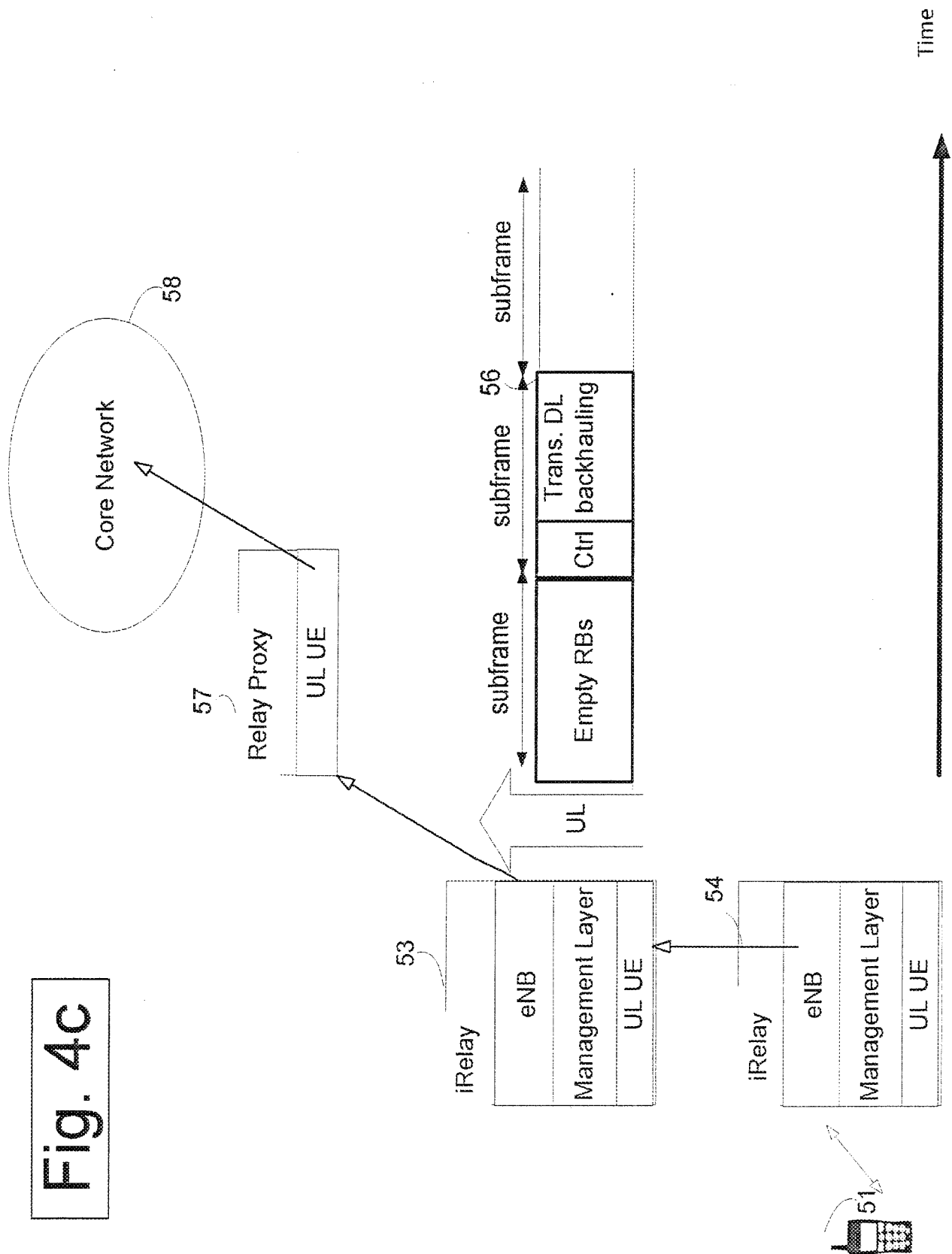
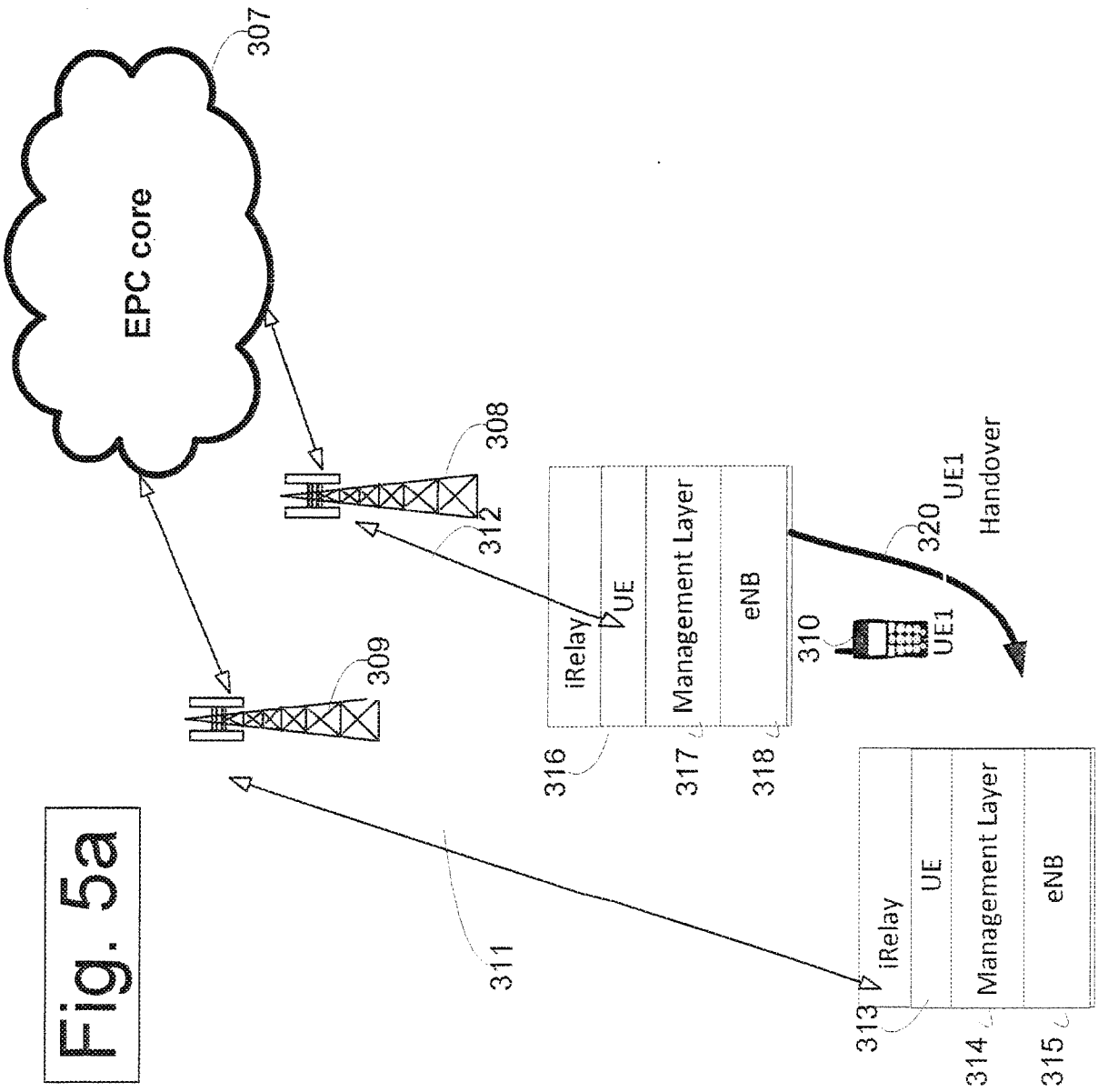
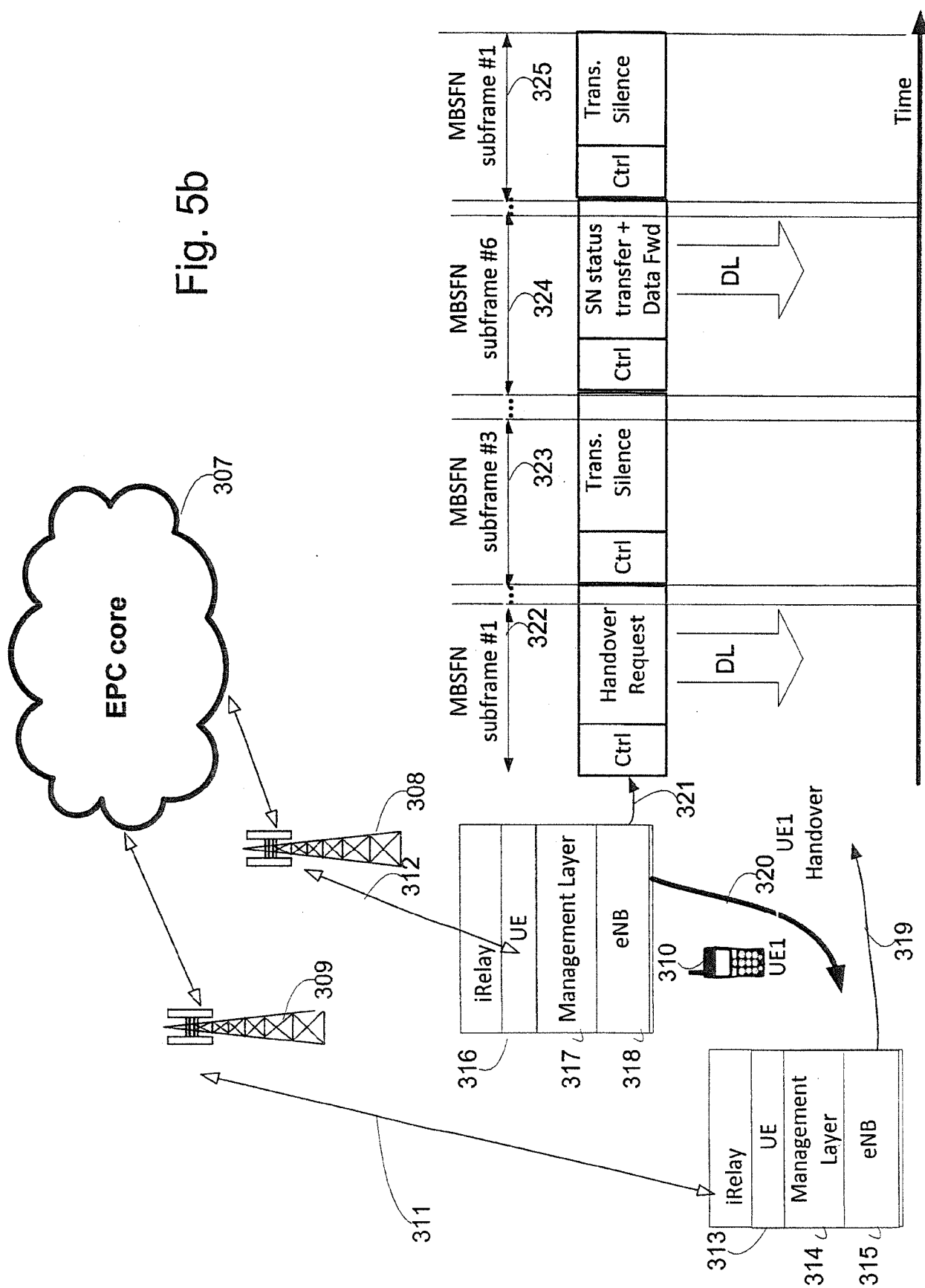


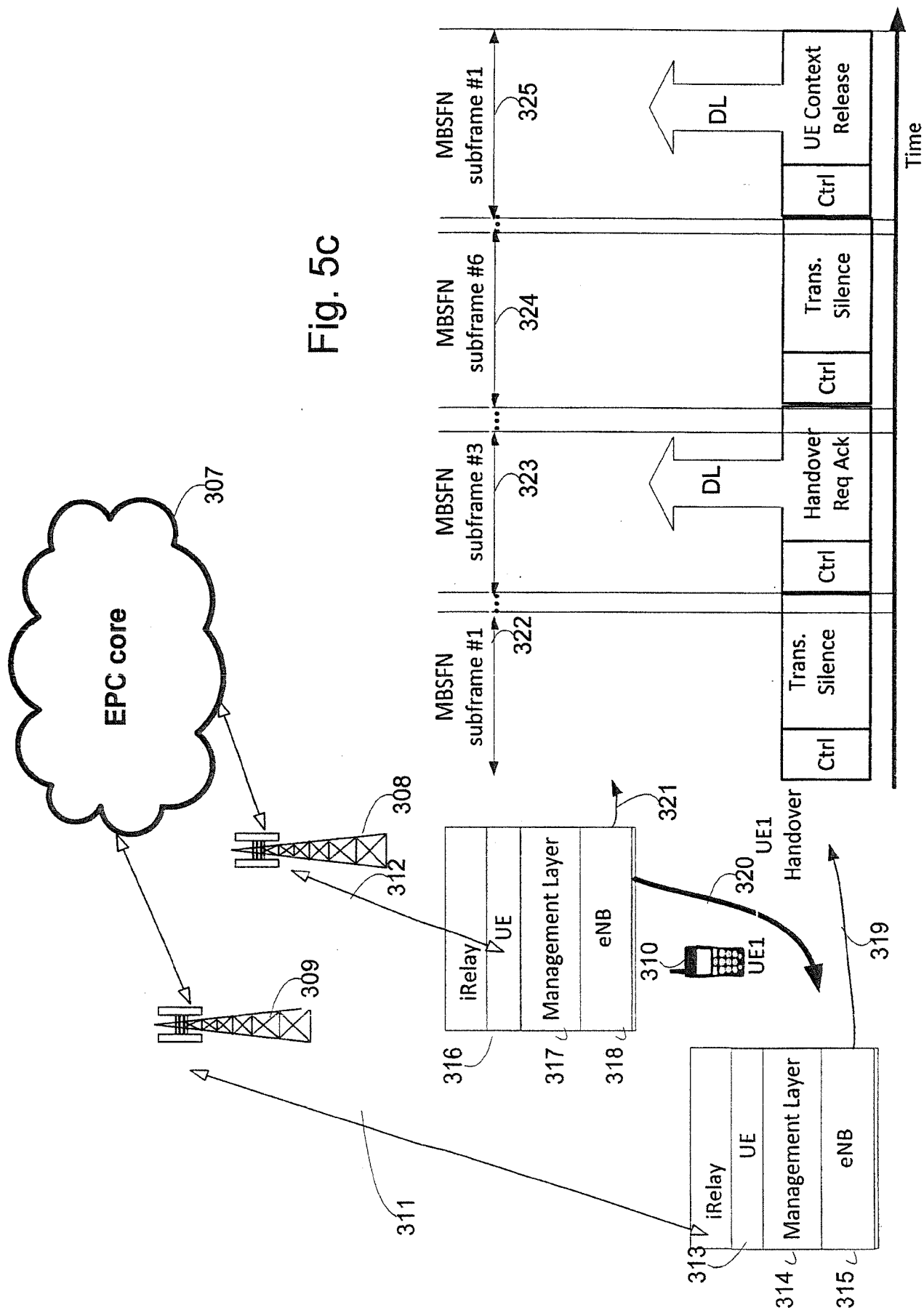
Fig. 4c

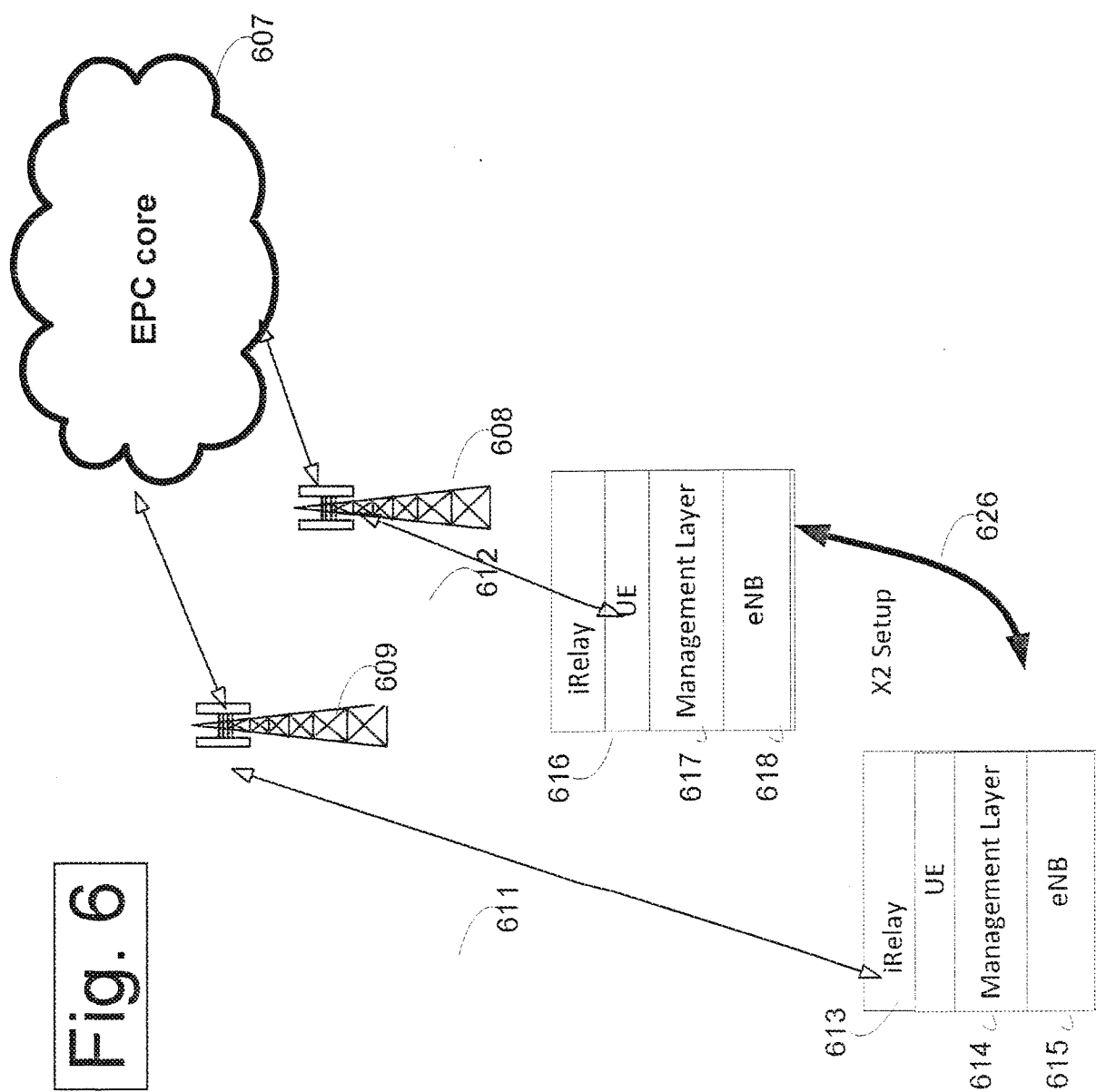


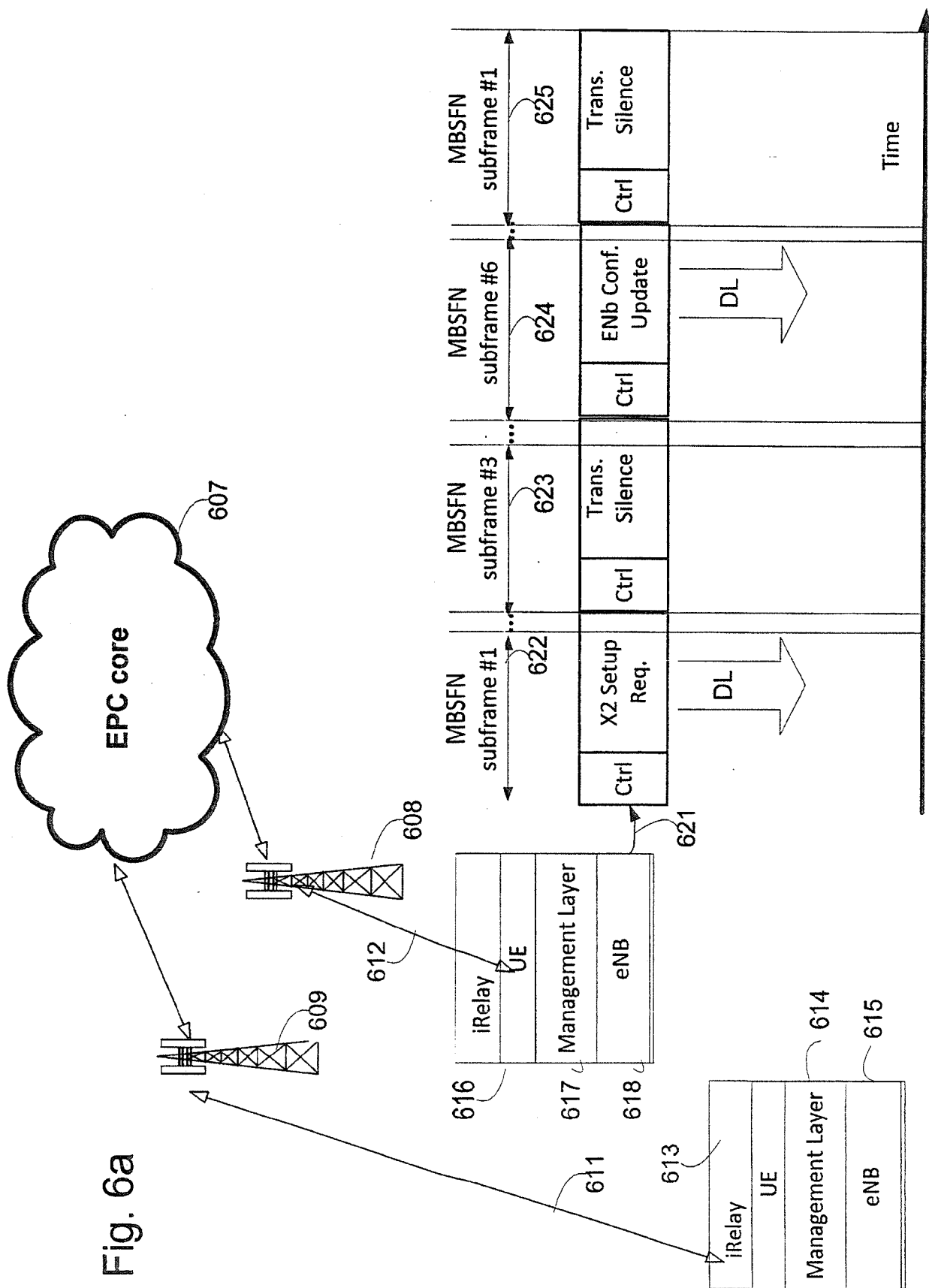


11/51









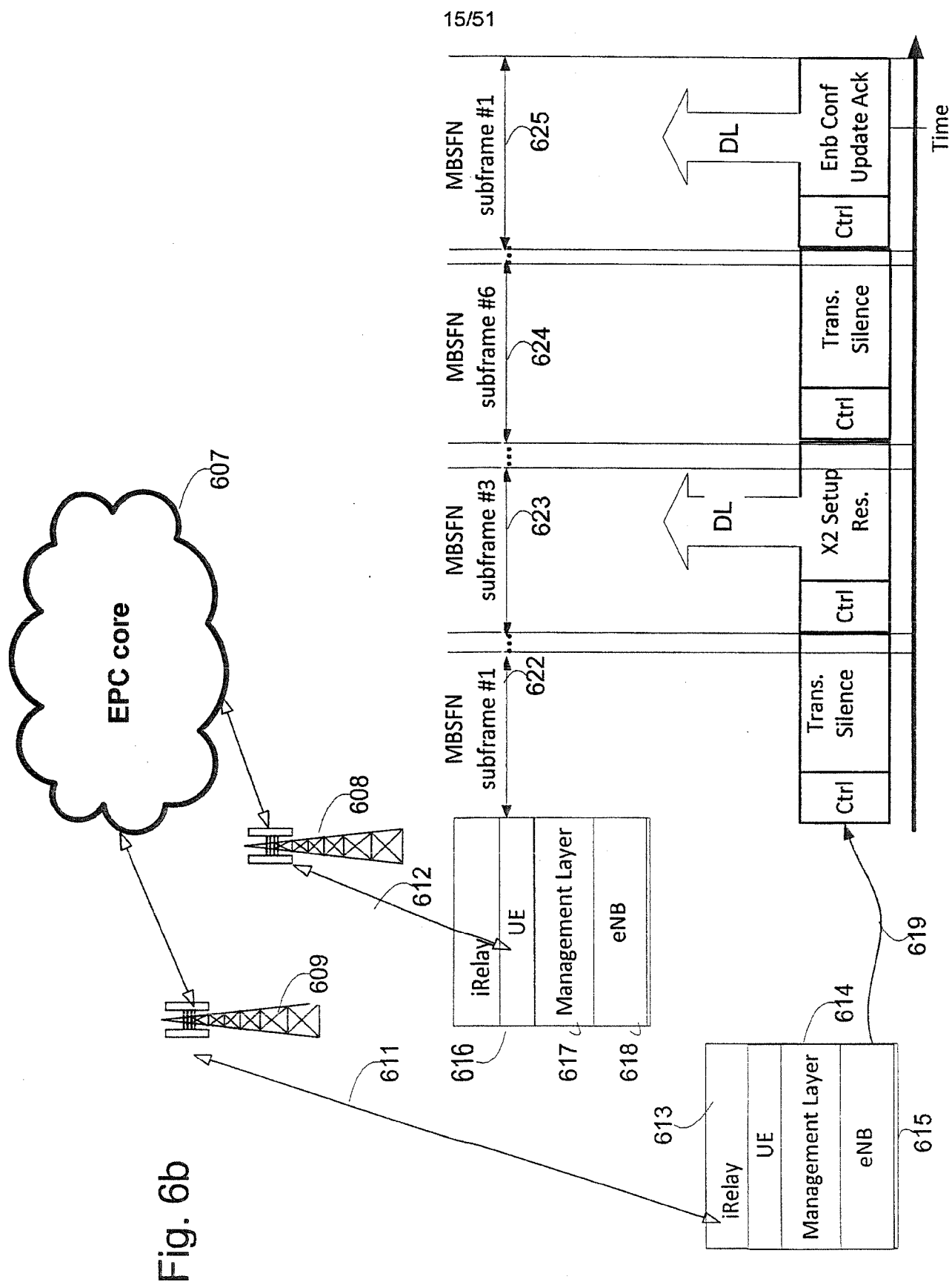


Fig. 7

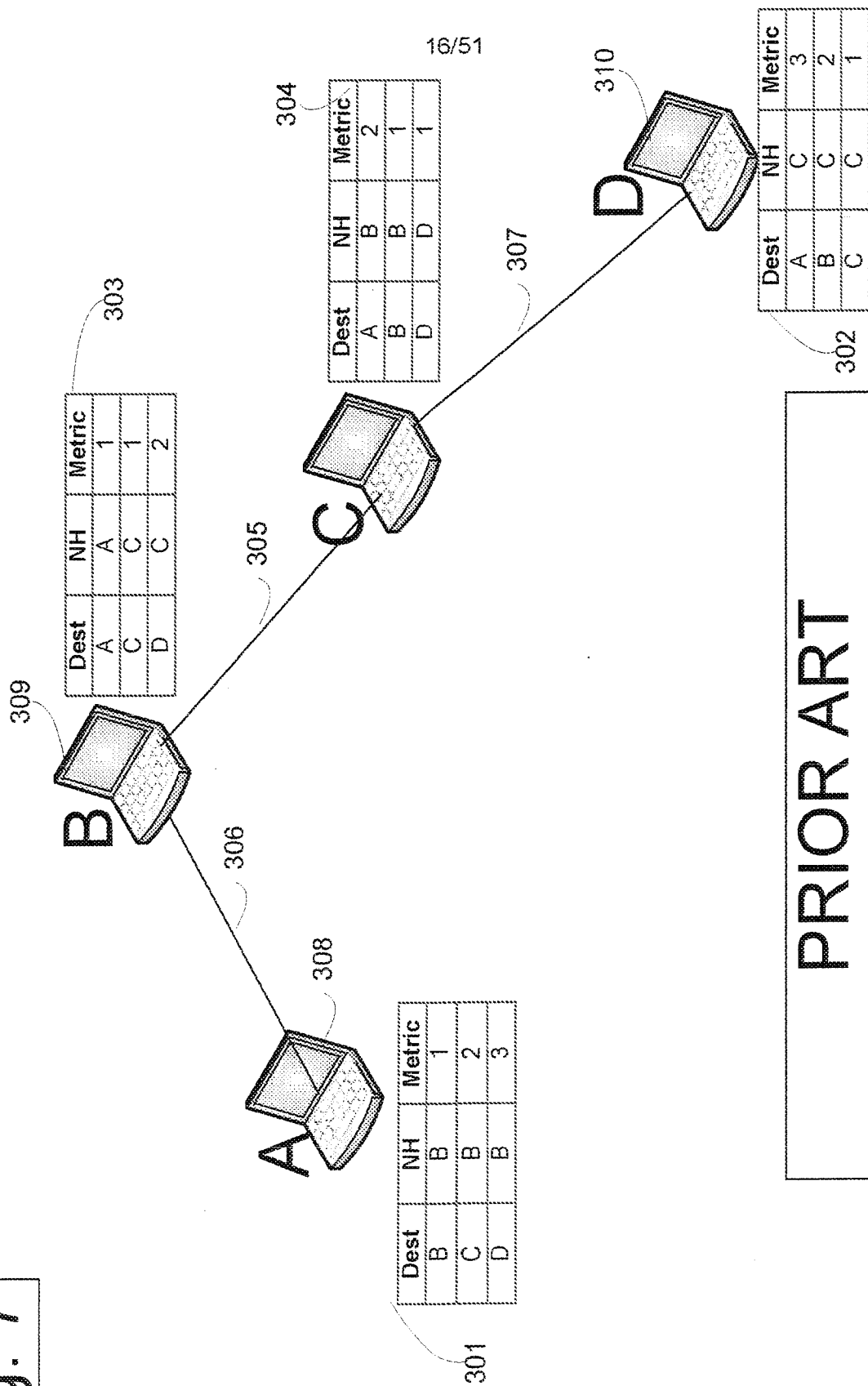


Fig. 8

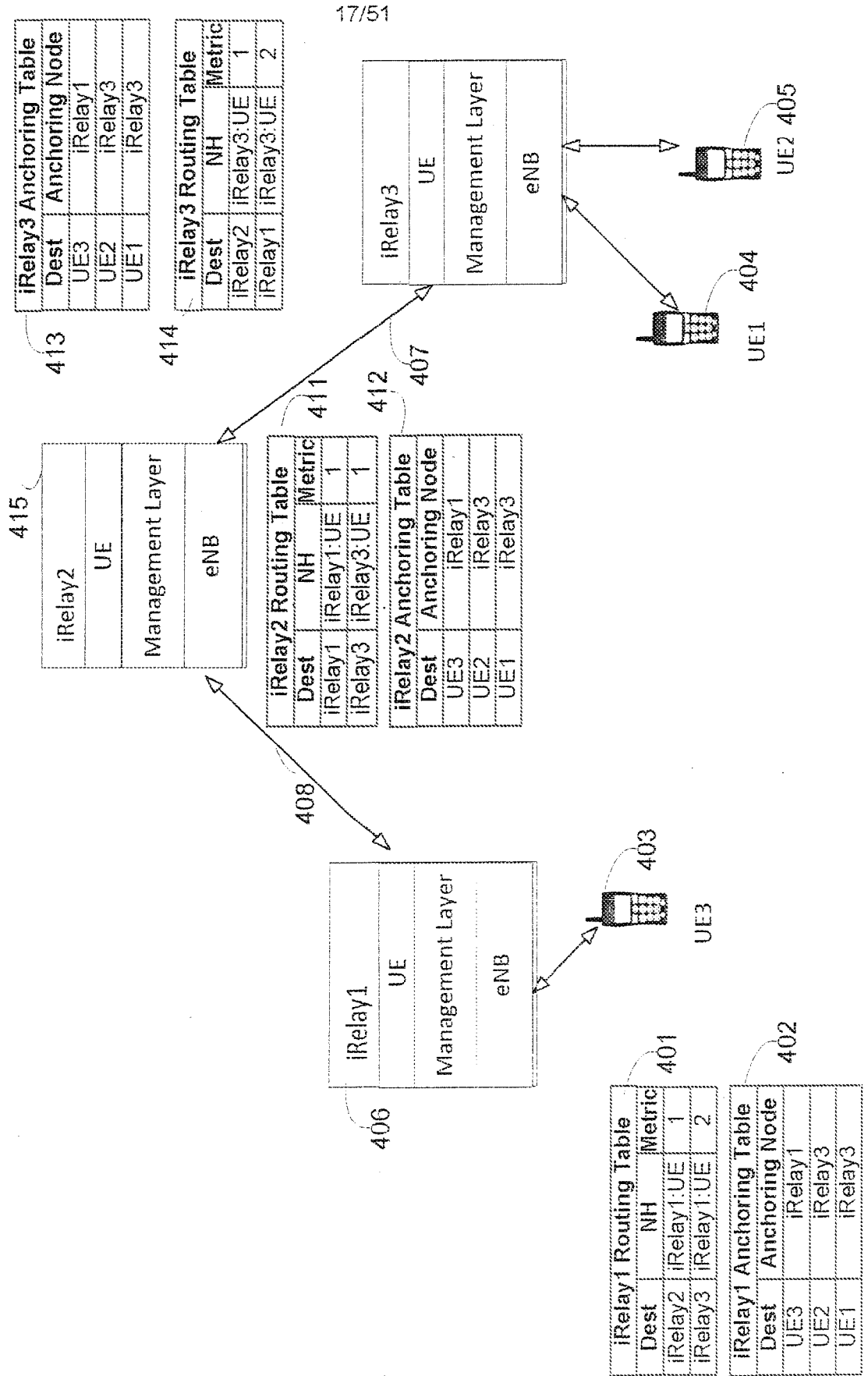


Fig. 8a

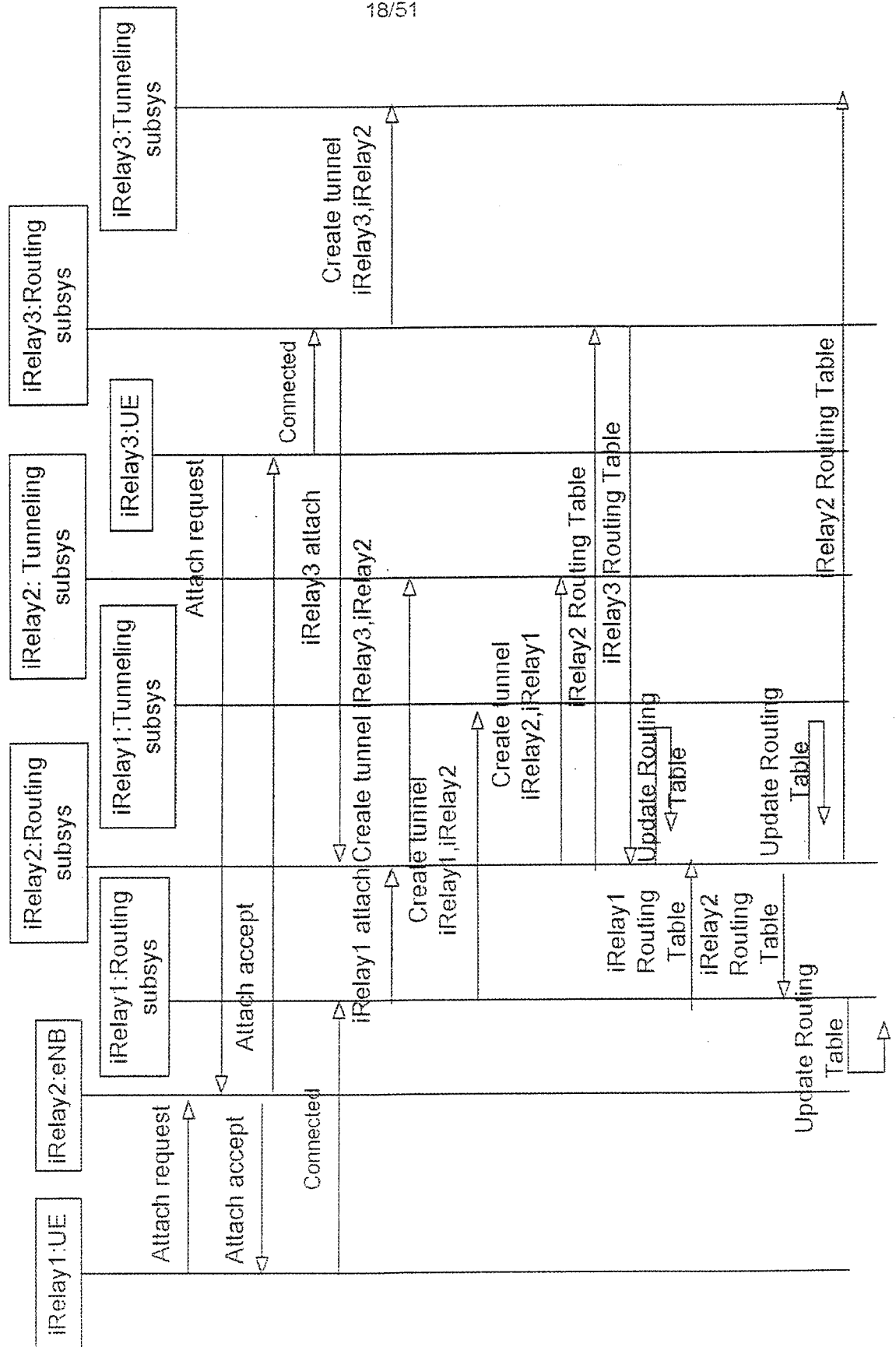
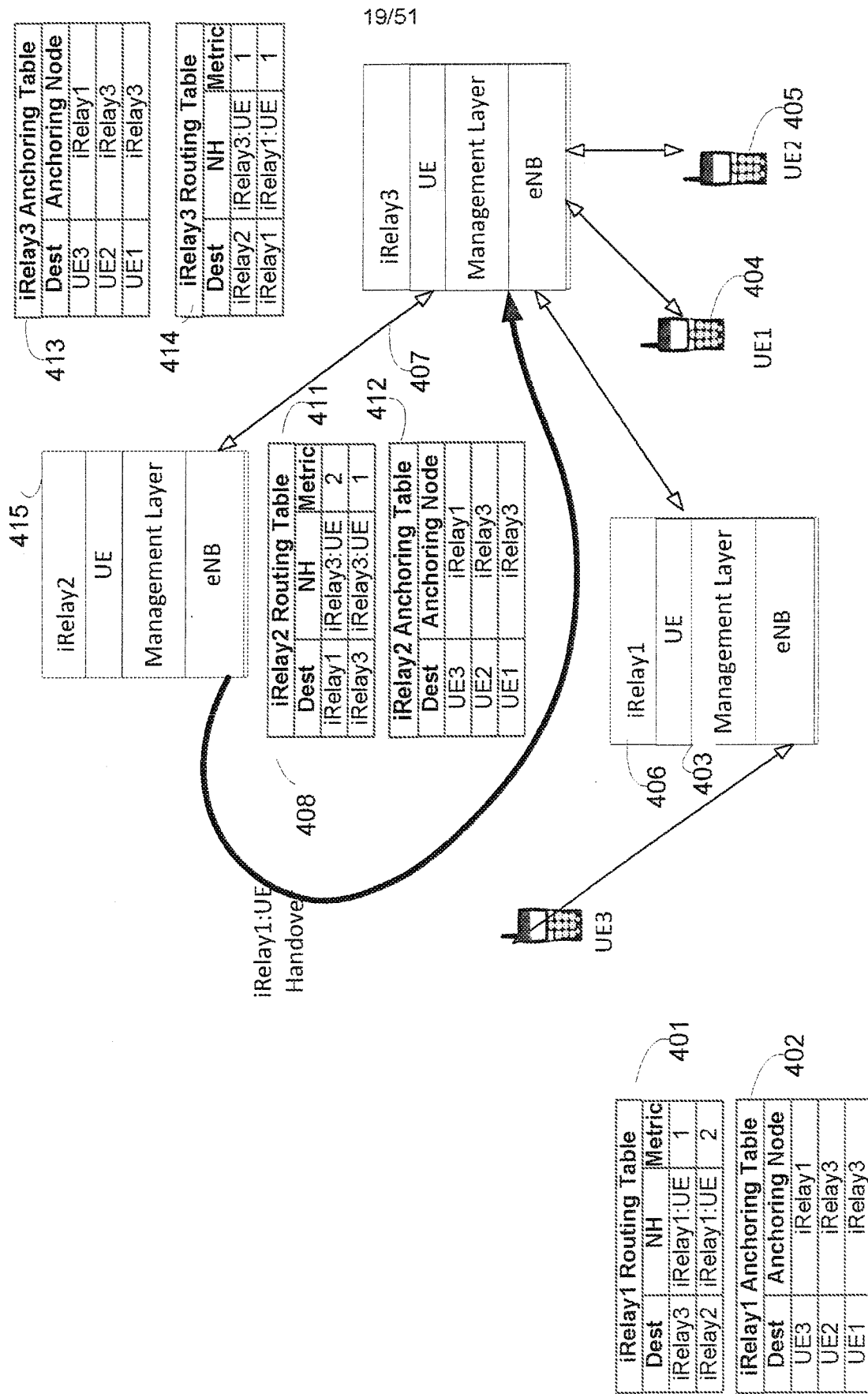


Fig. 8b



20/51

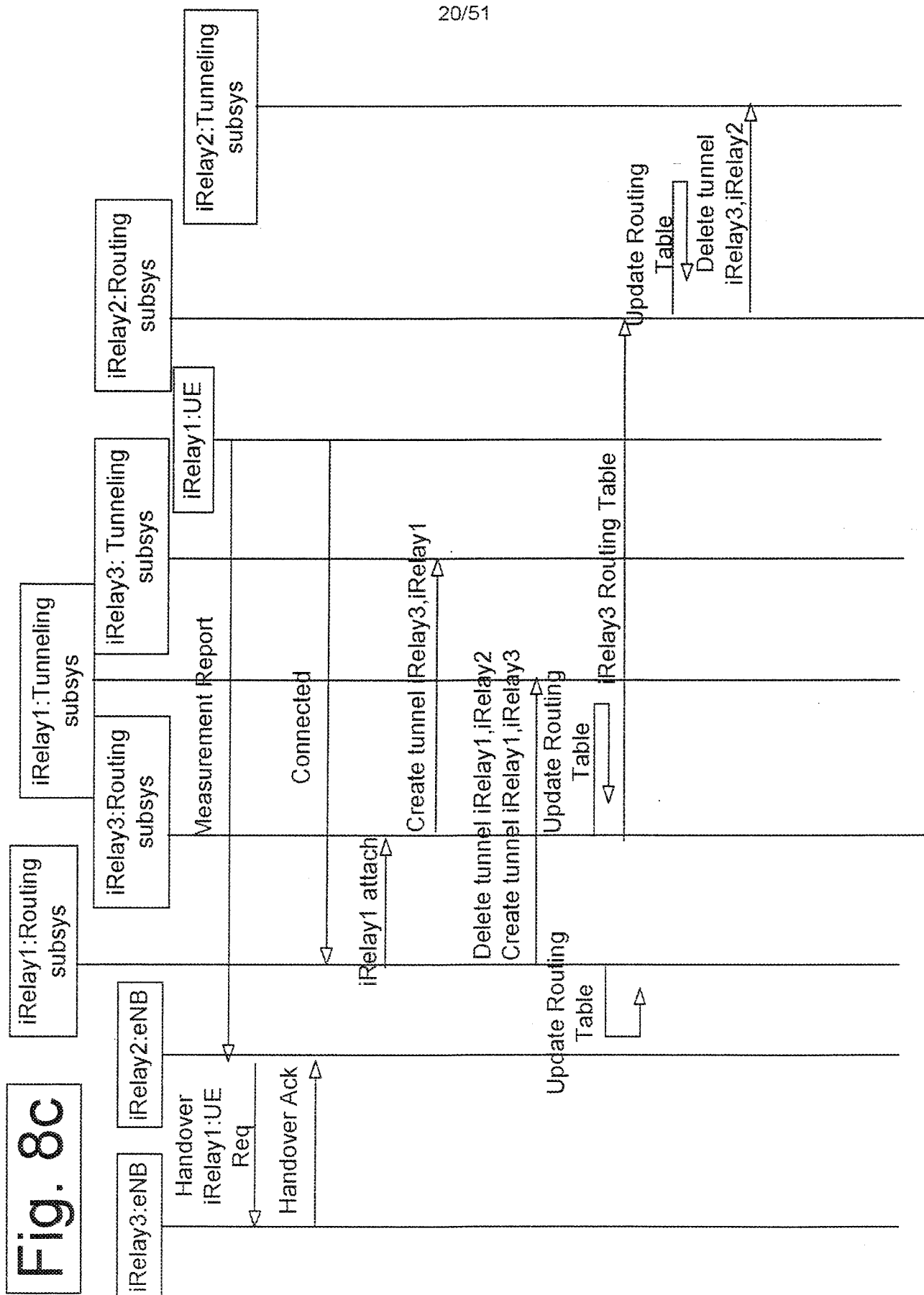


Fig. 9

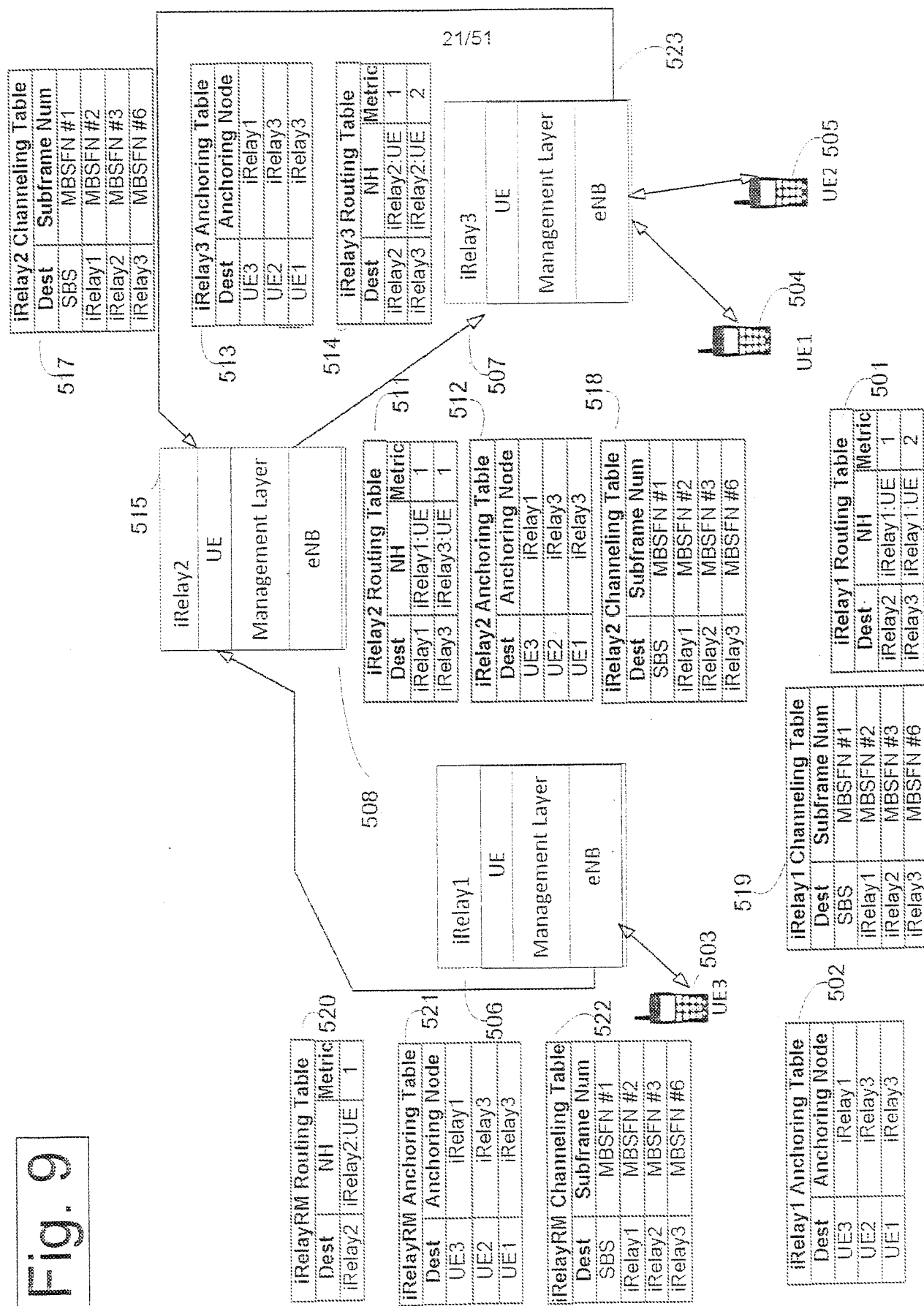
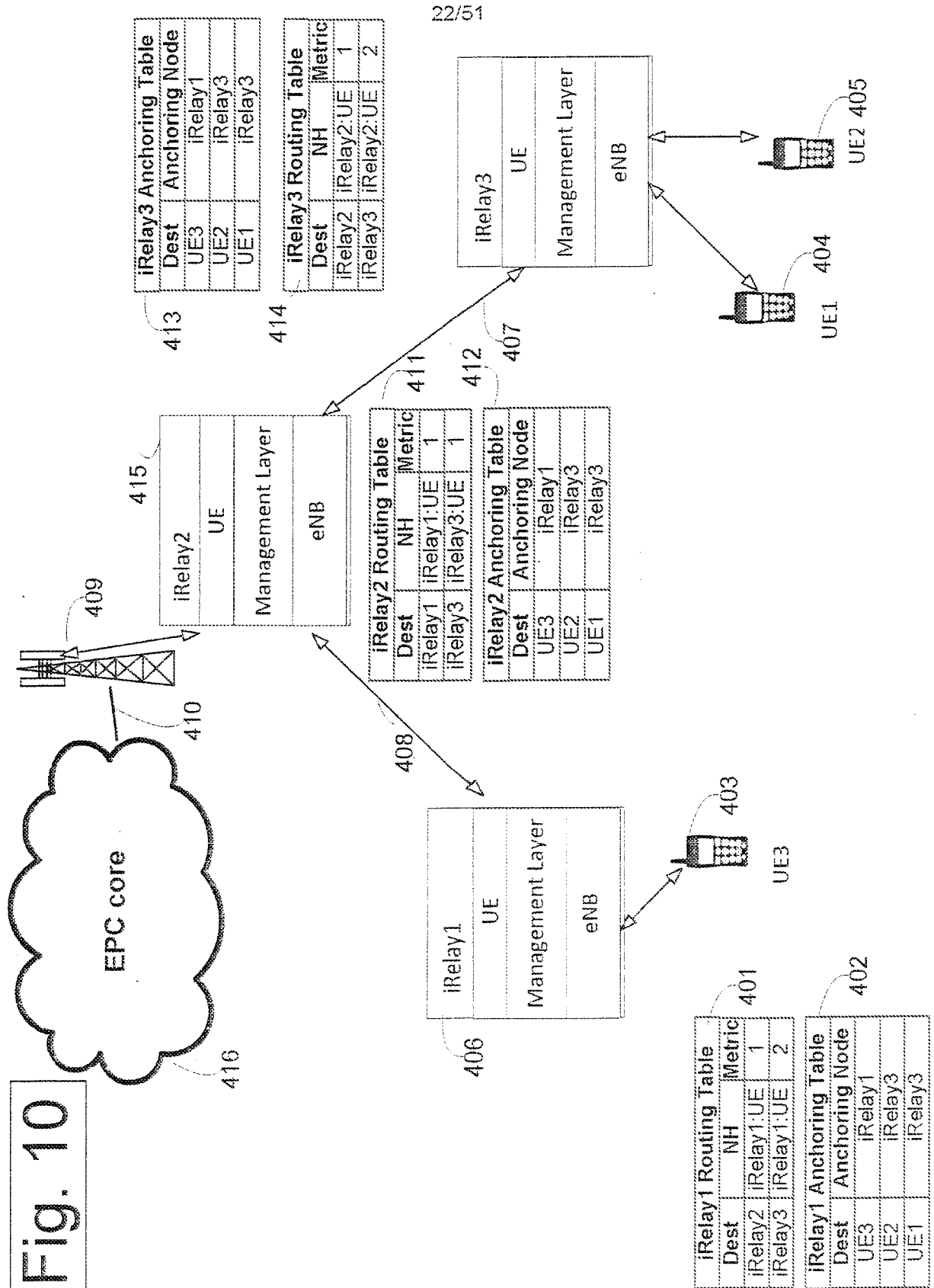
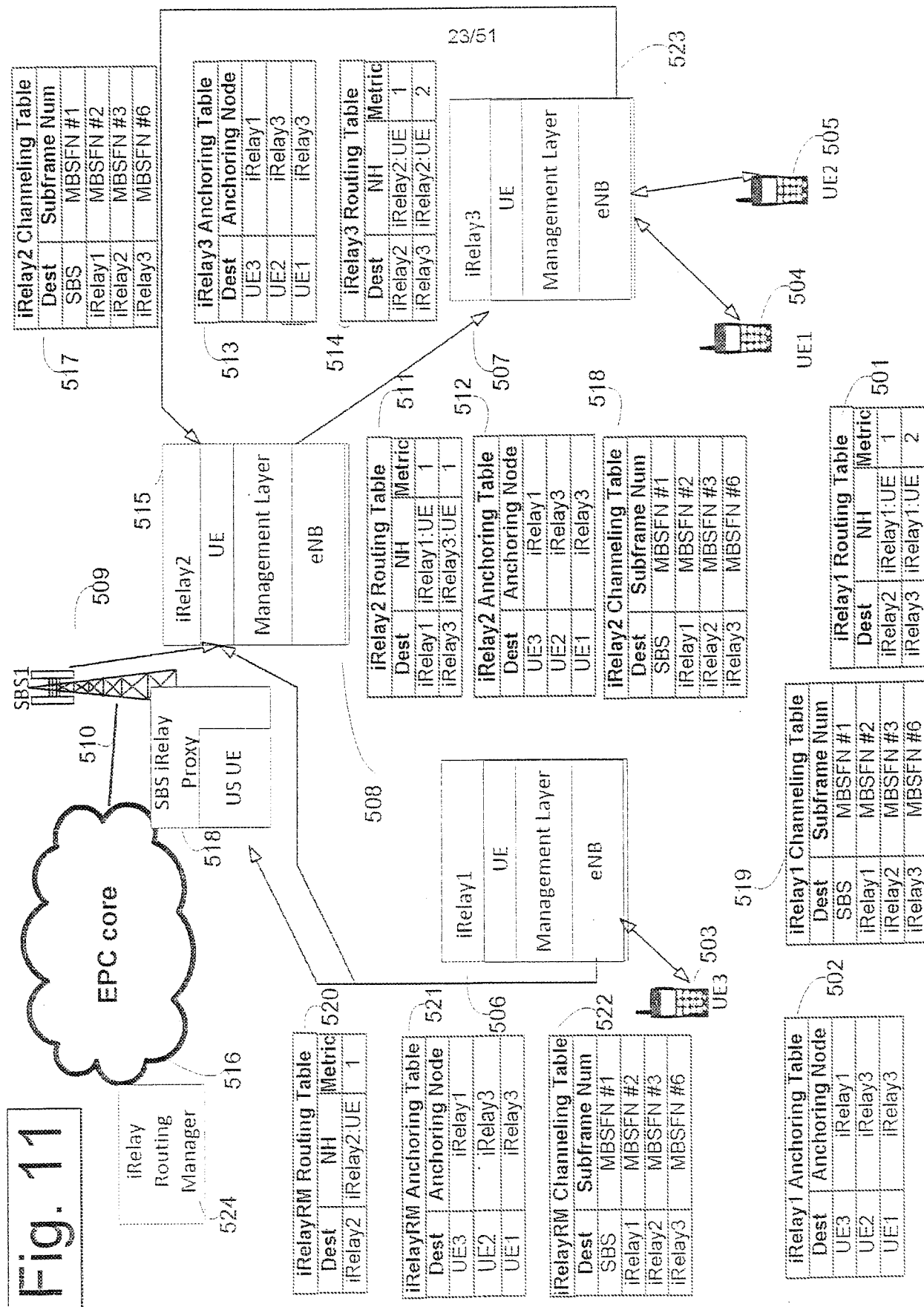
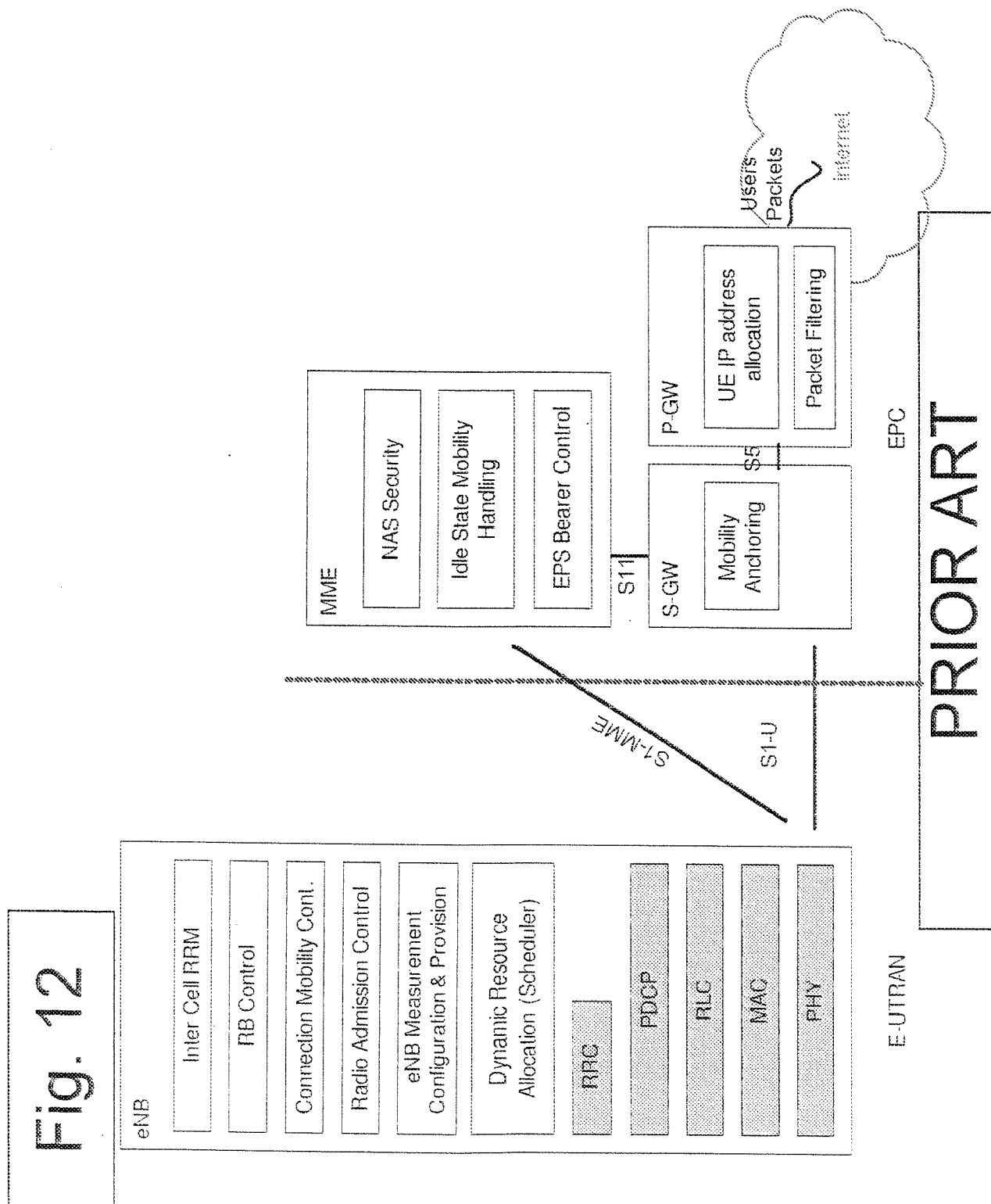


Fig. 10





24/51



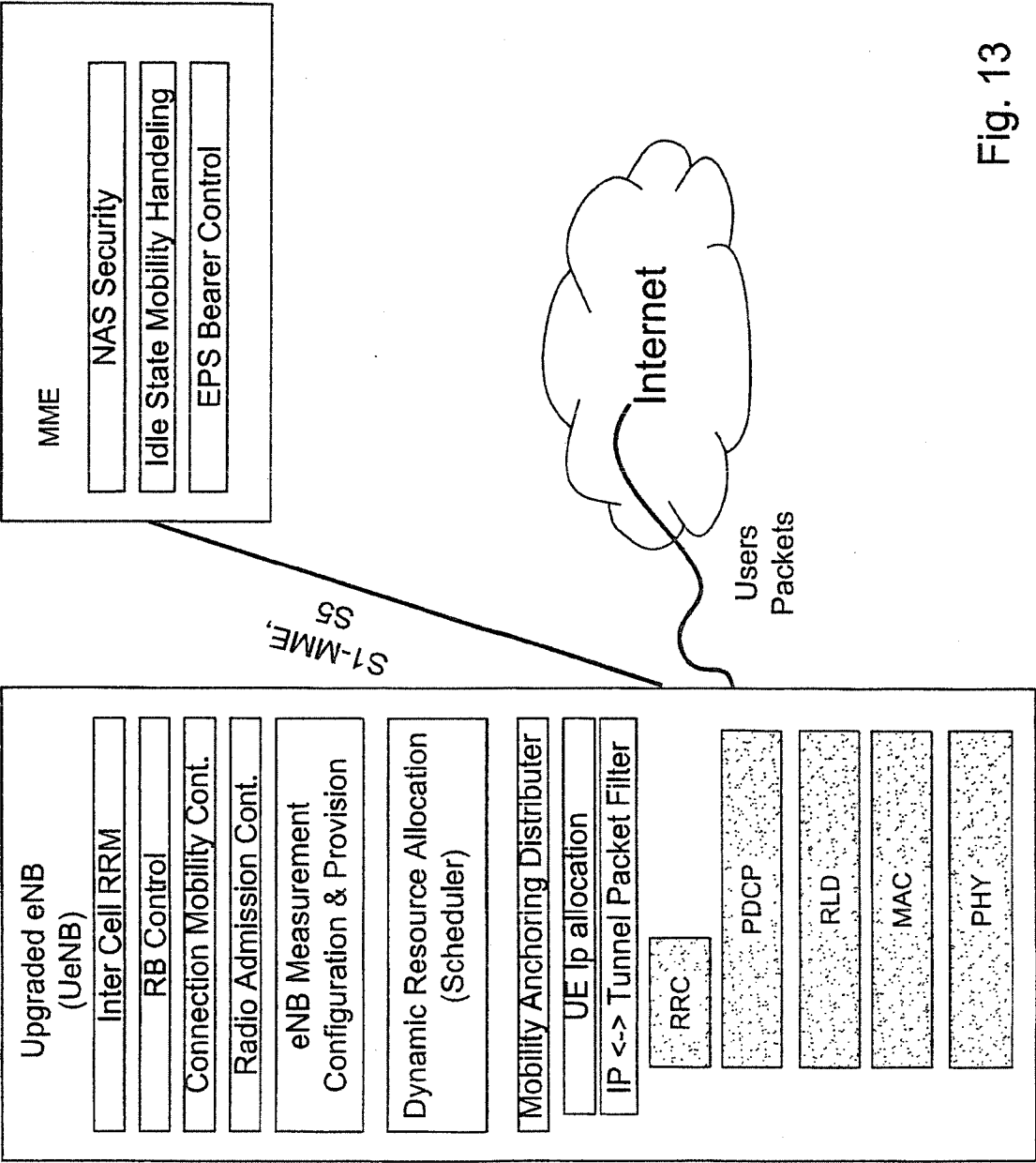


Fig. 13

26/51

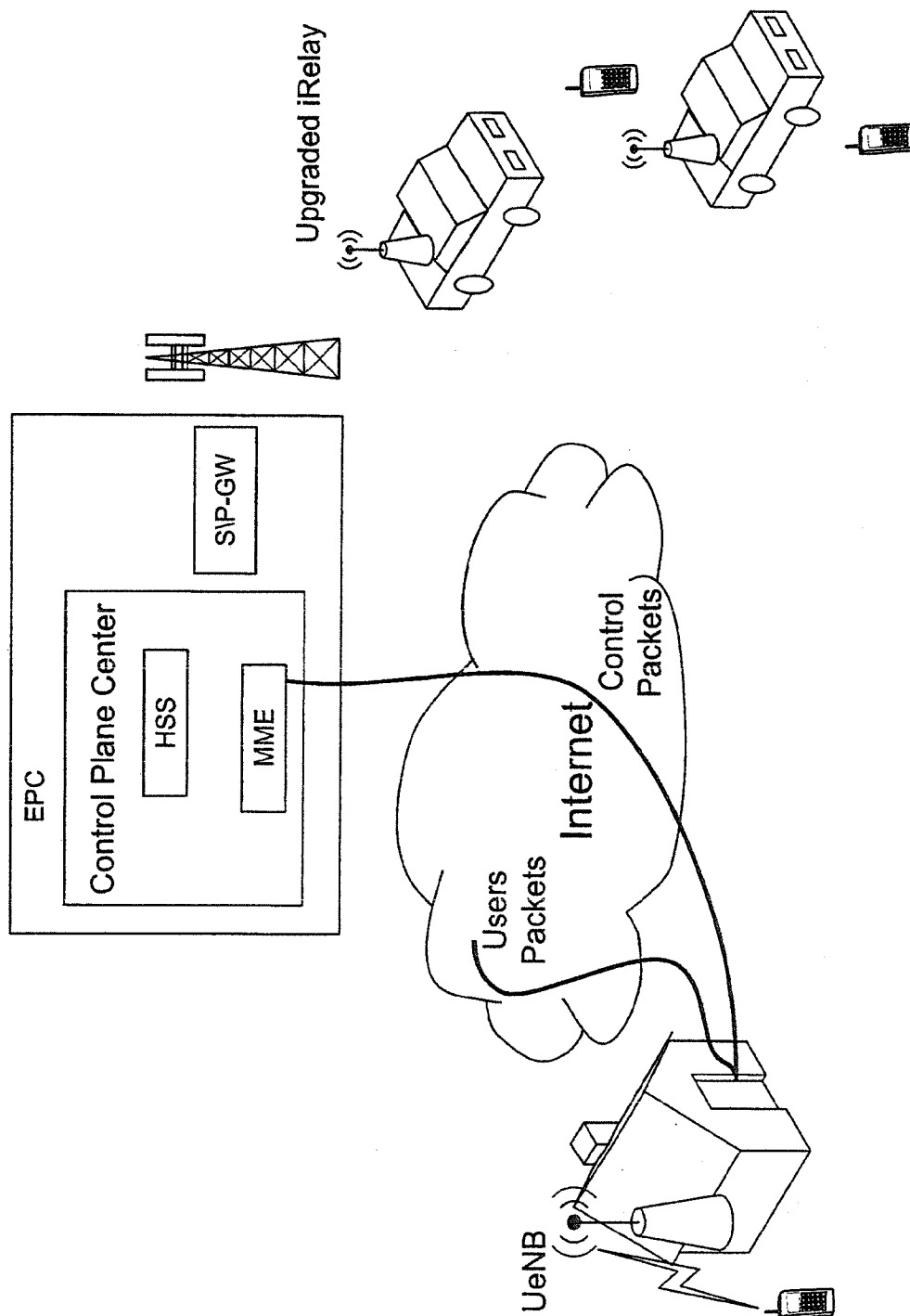


Fig. 14

Fig. 15

27/51

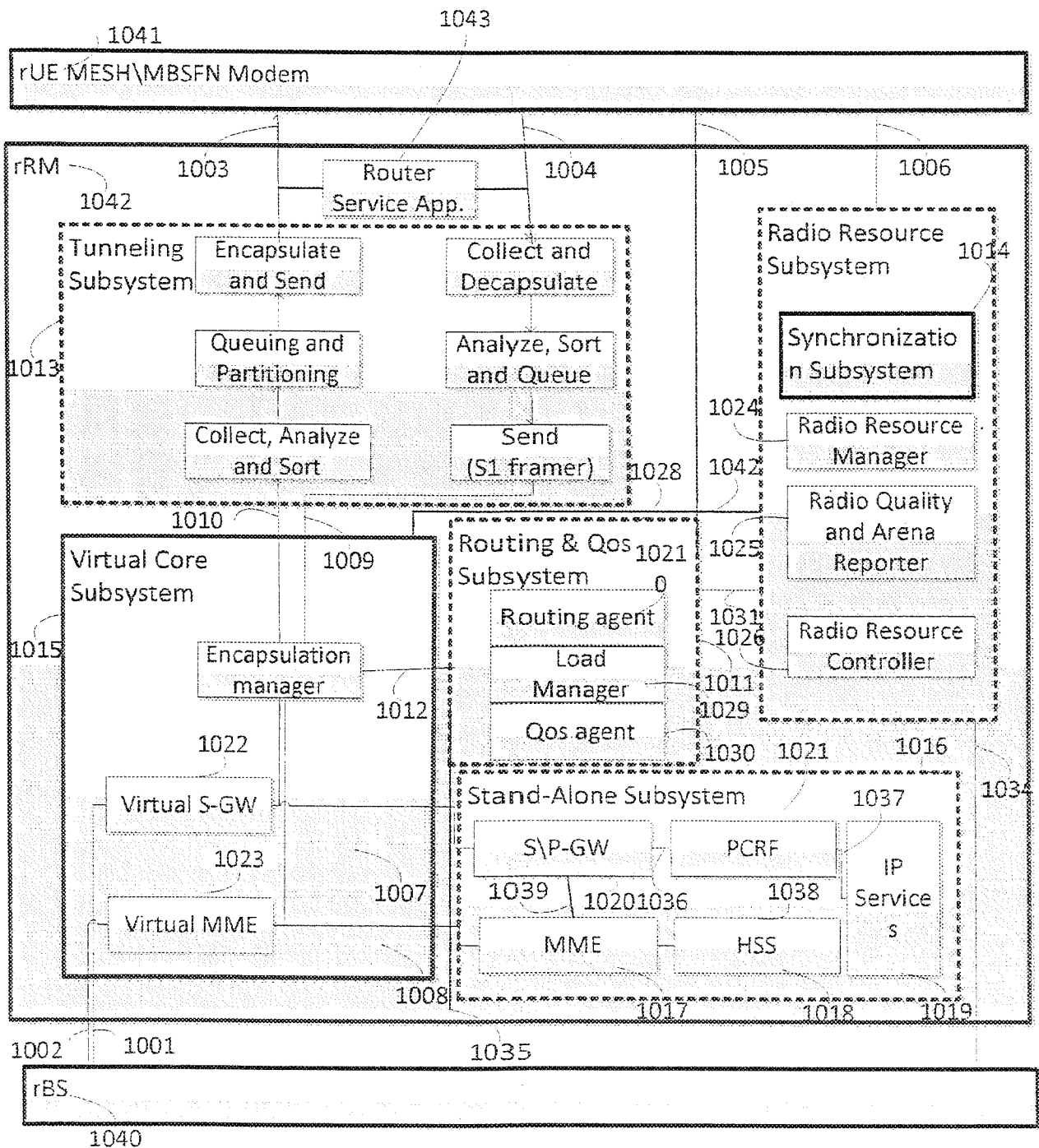
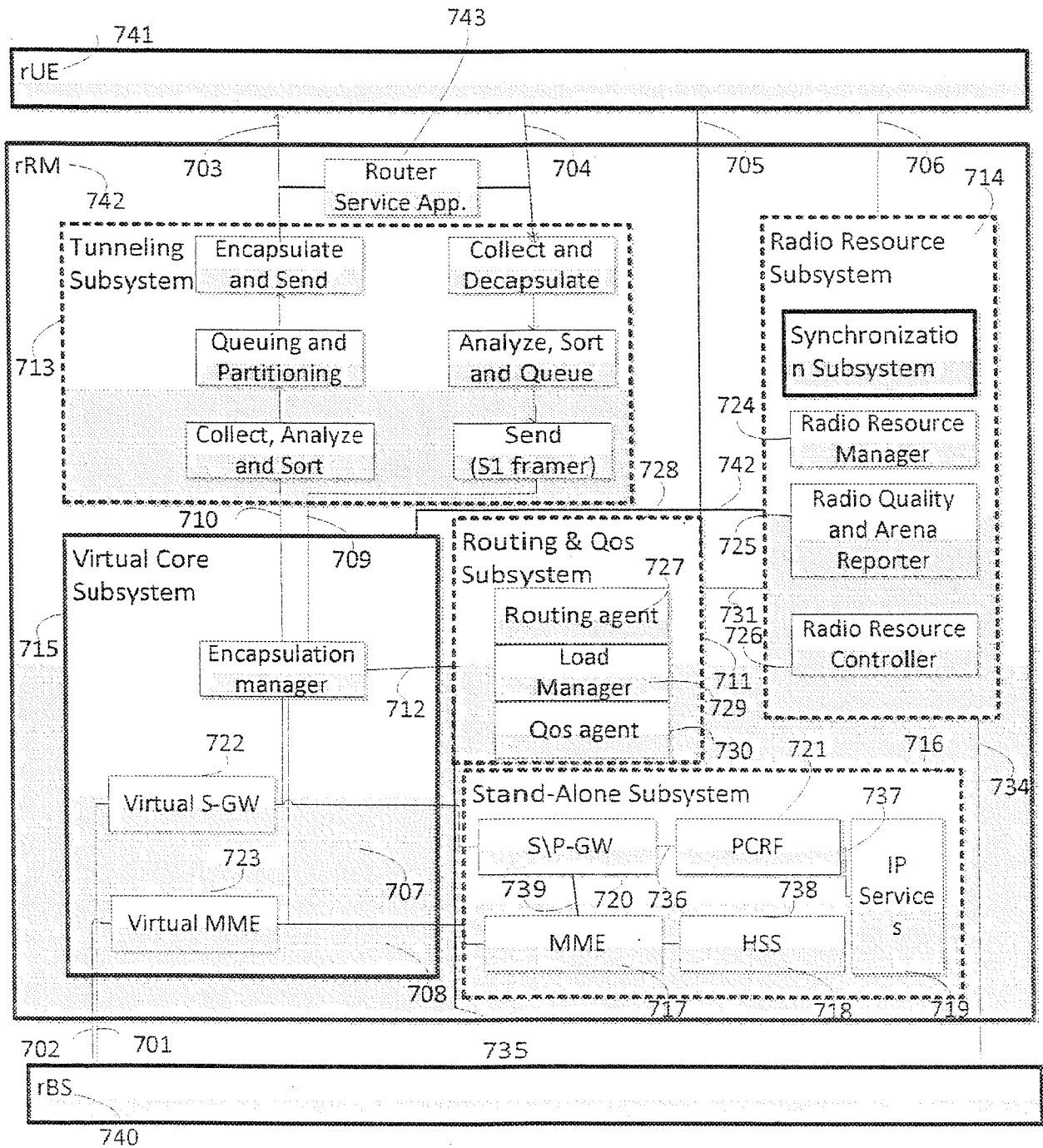
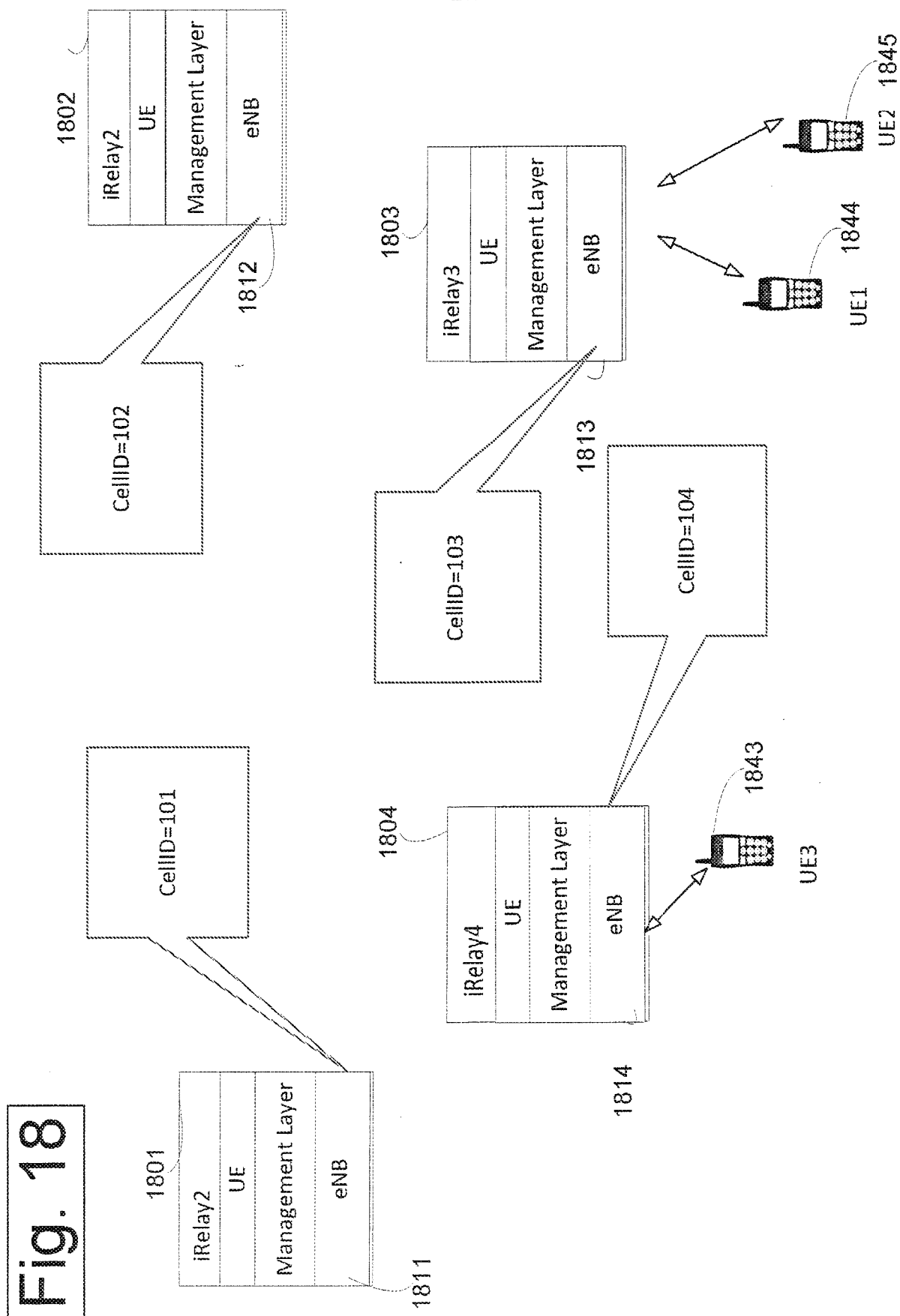


Fig. 16

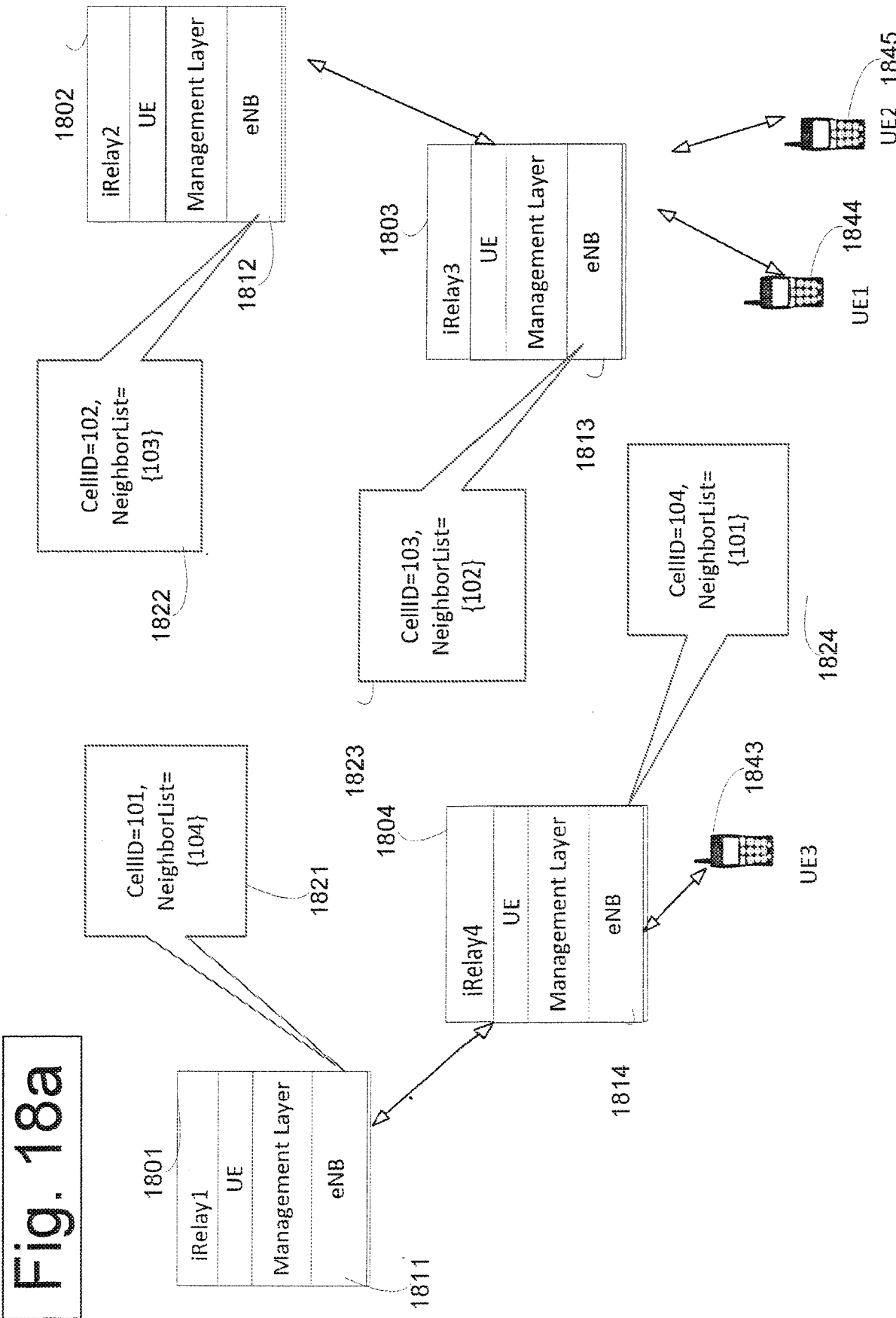
28/51



29/51



30/51



31/51

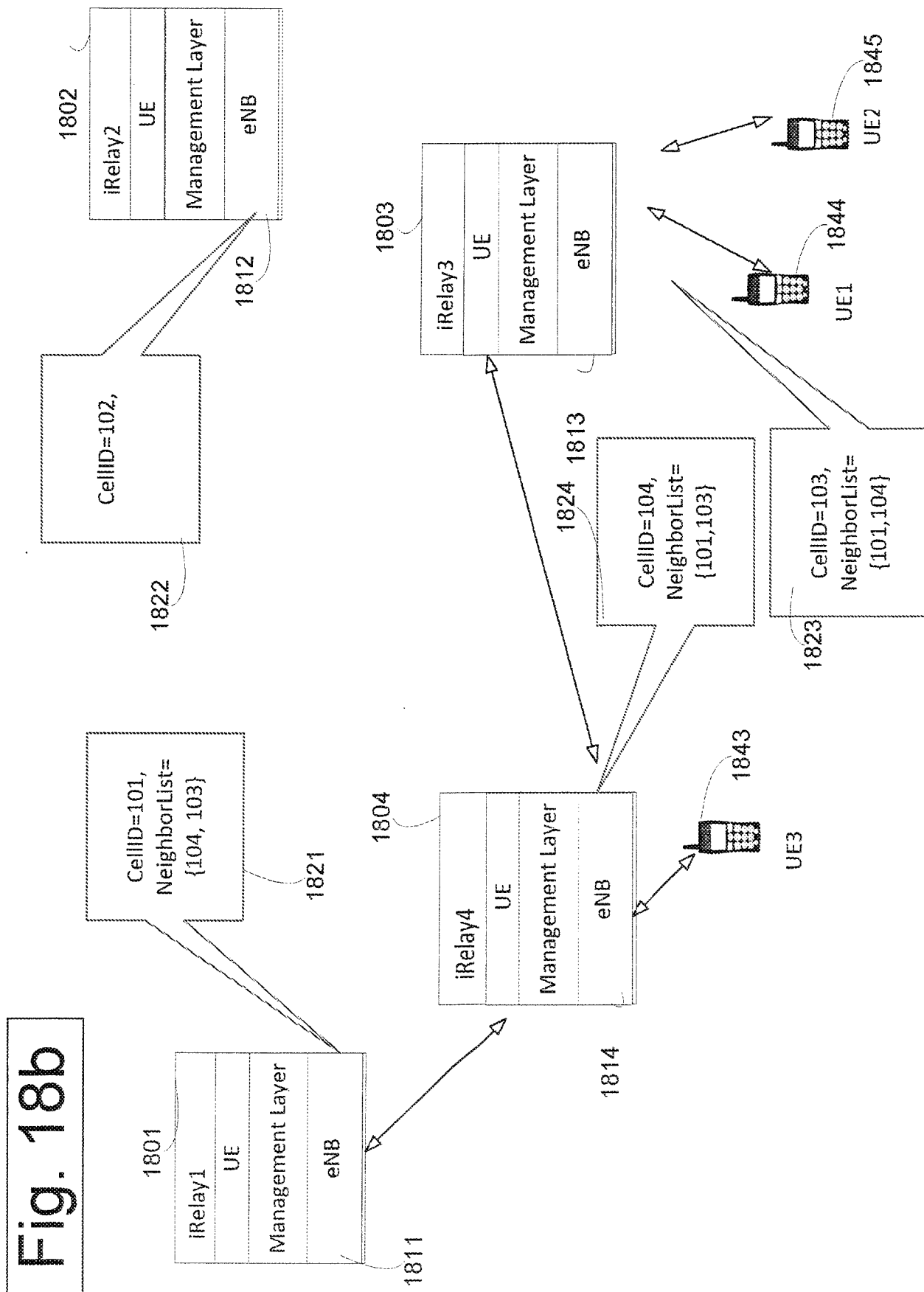


Fig. 18c

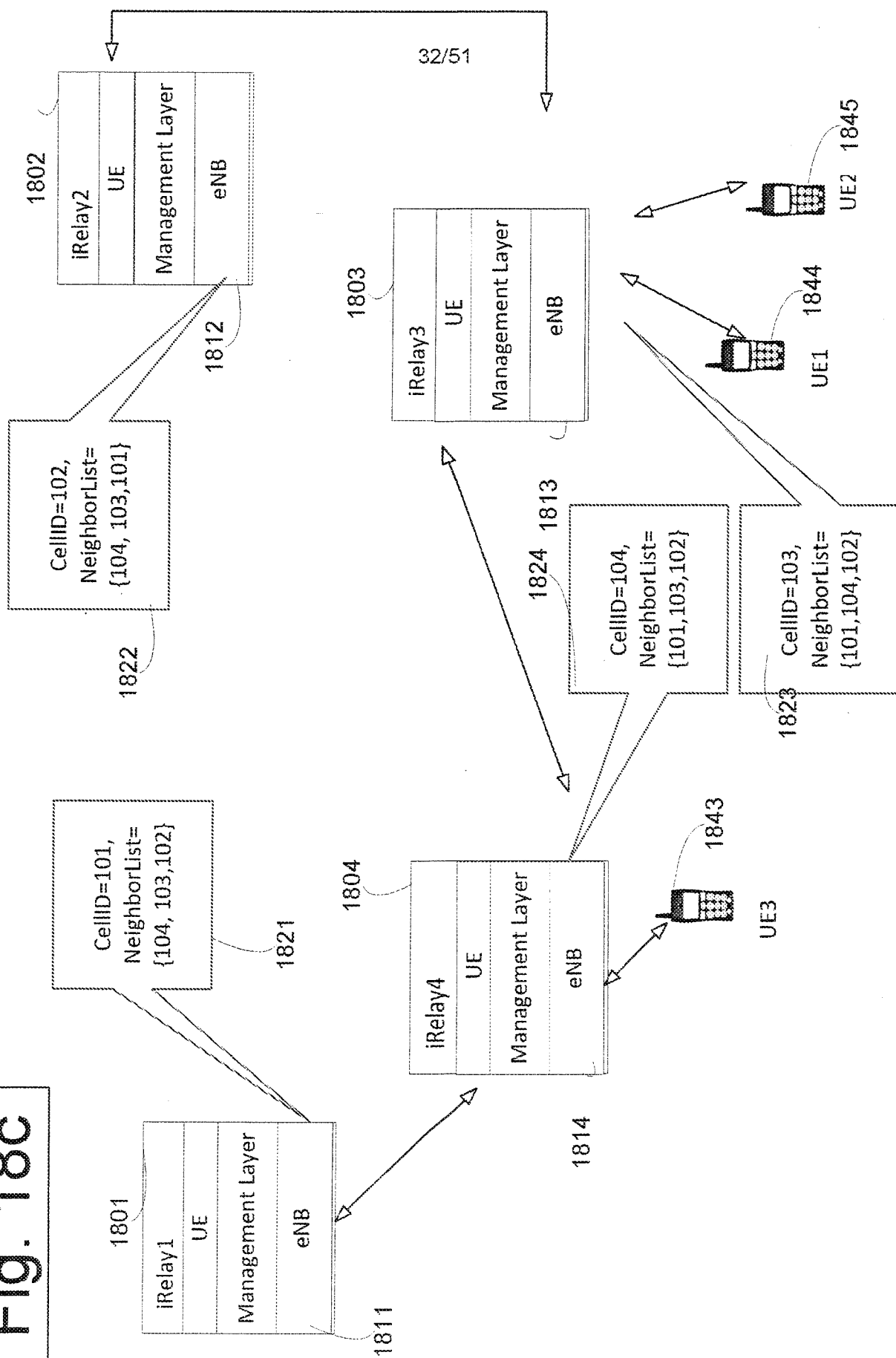
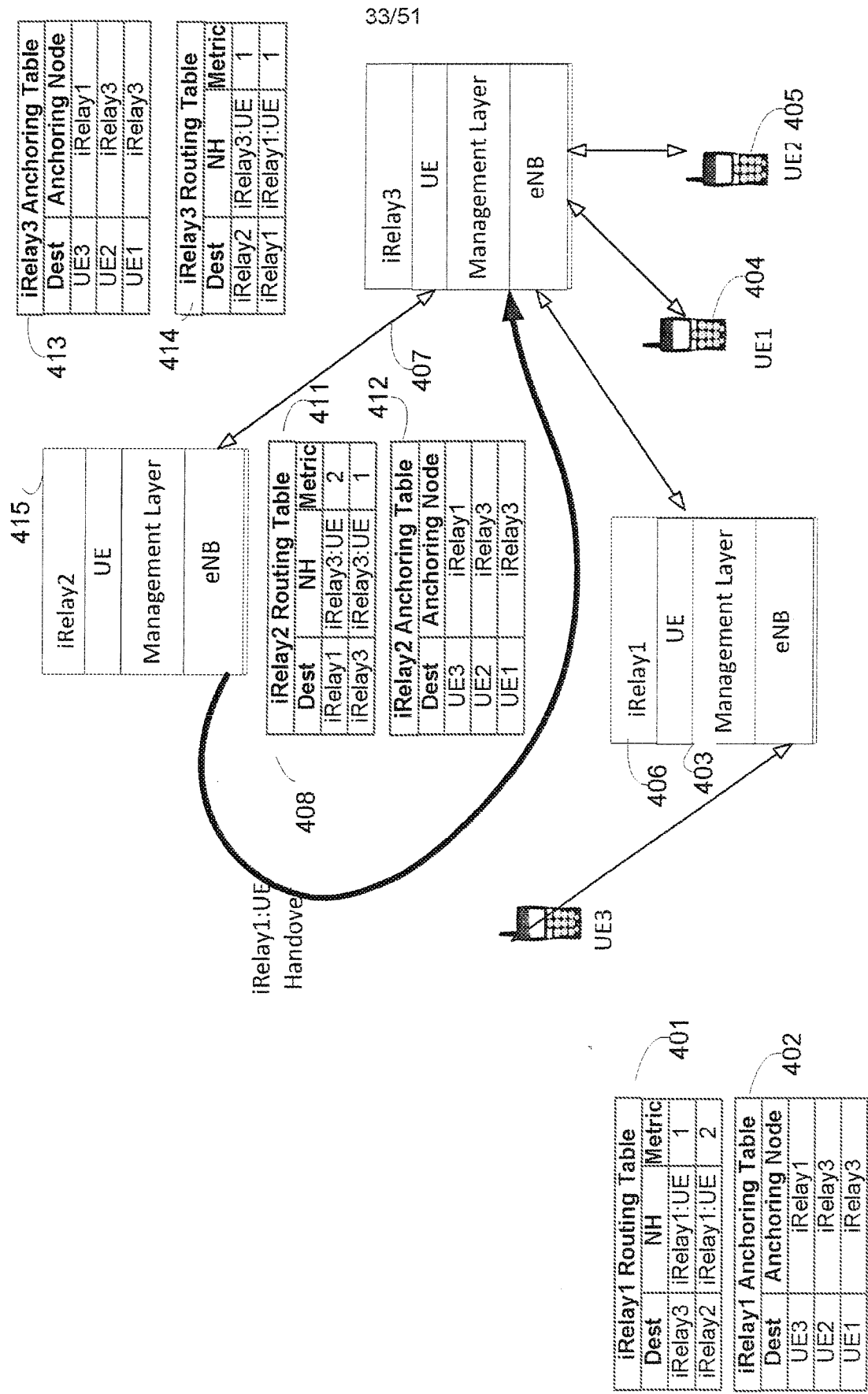
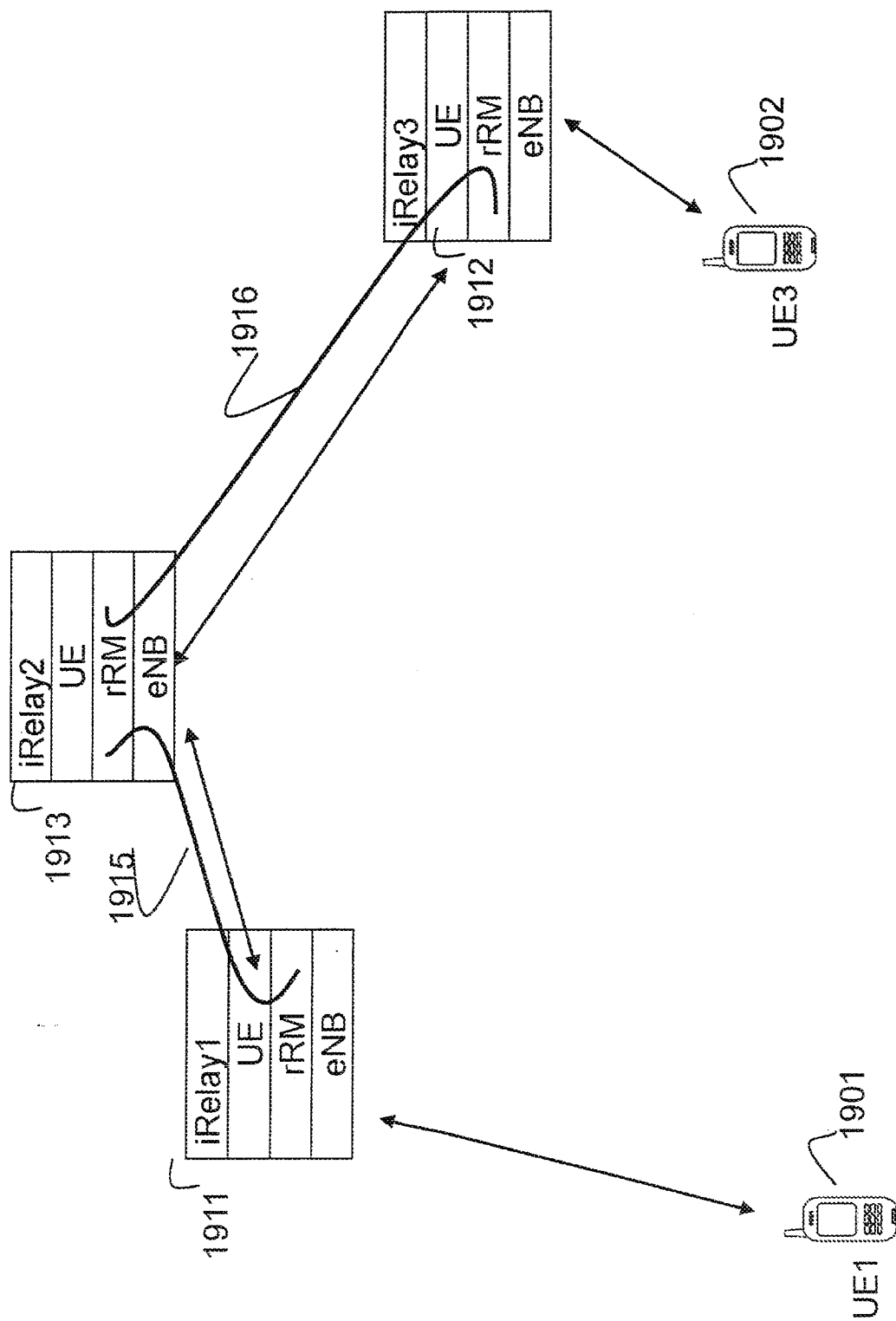


Fig. 19



34/51

Fig 19



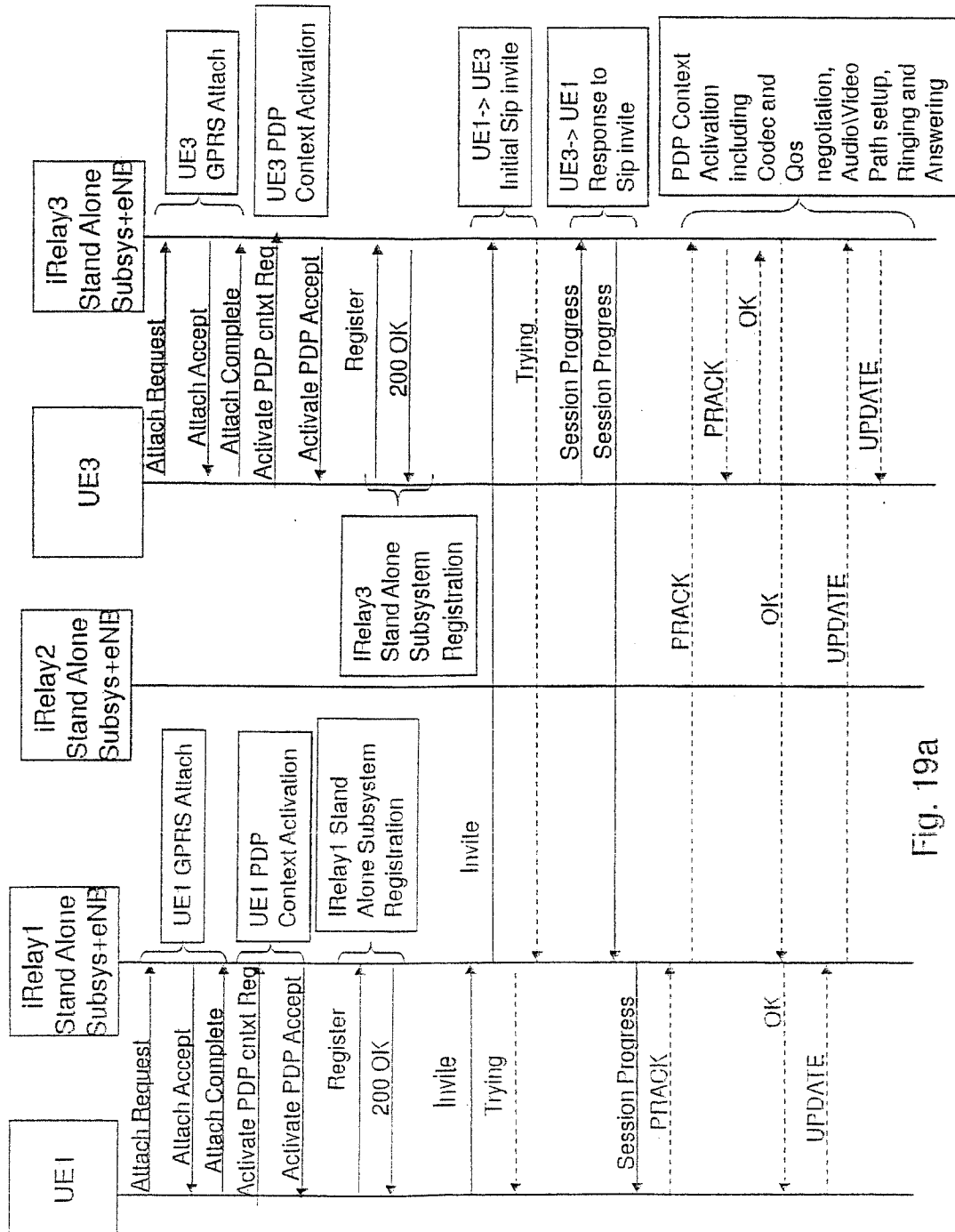


Fig. 19a

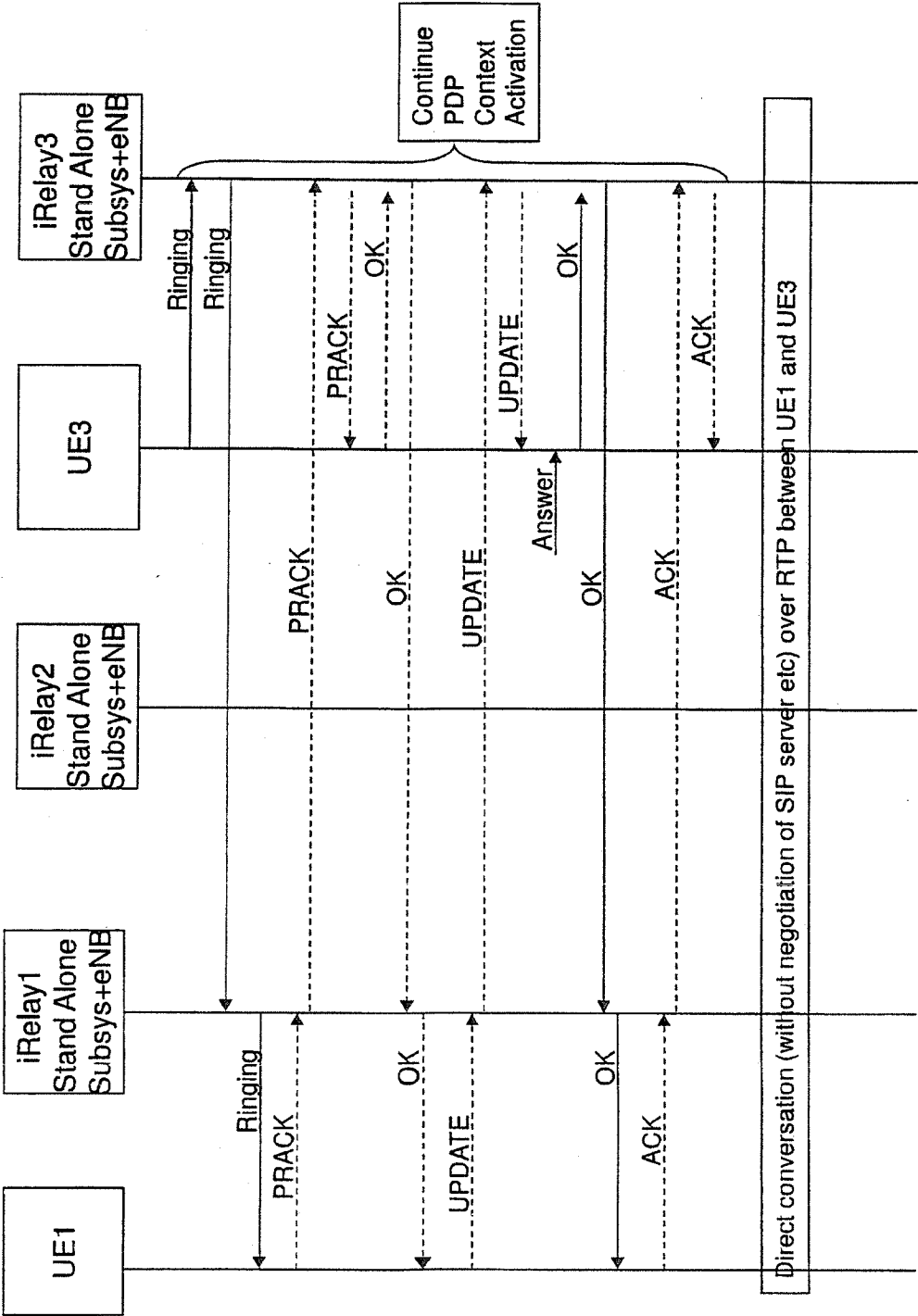


Fig. 19b

37/51

Fig 19c

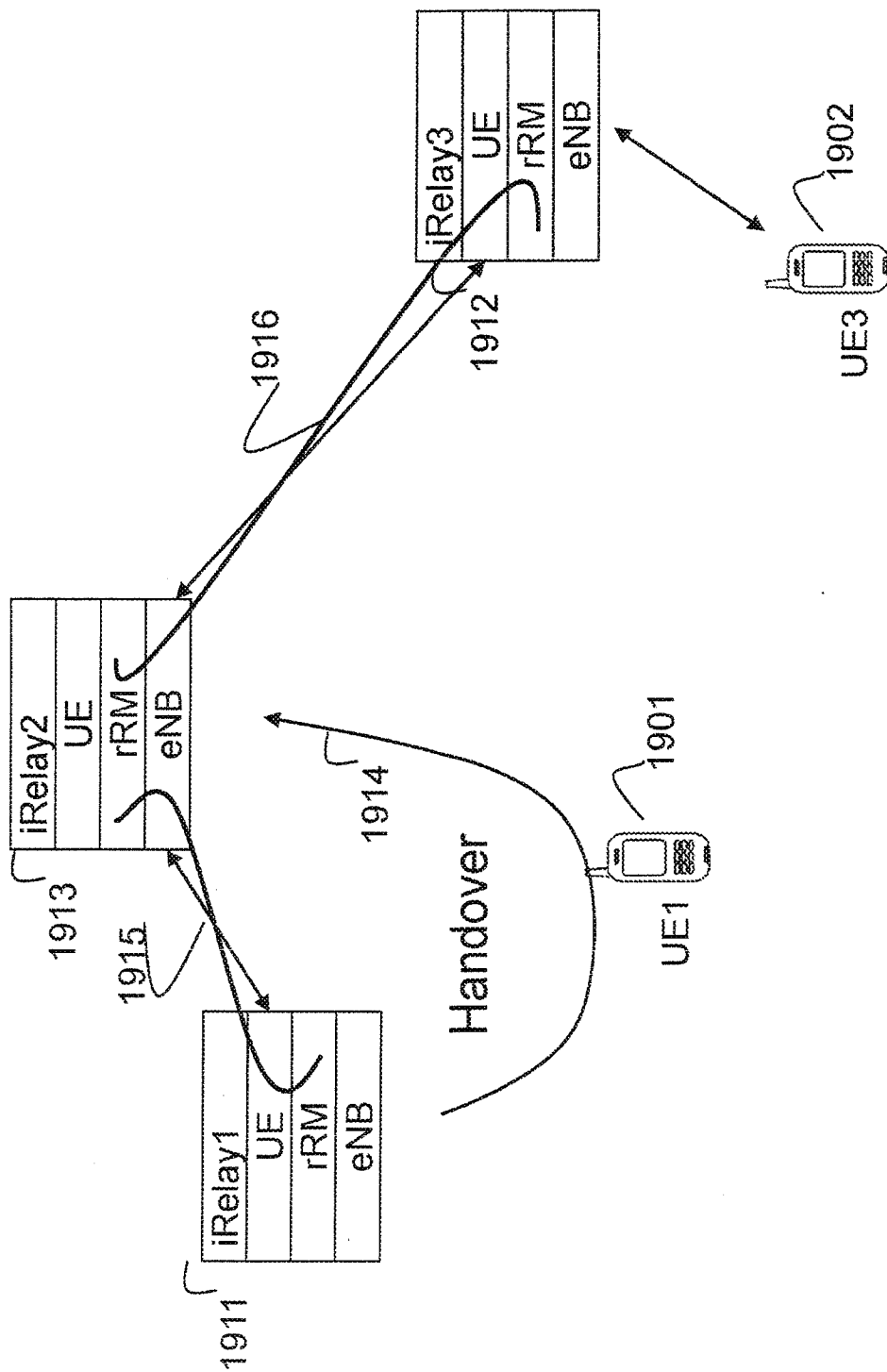
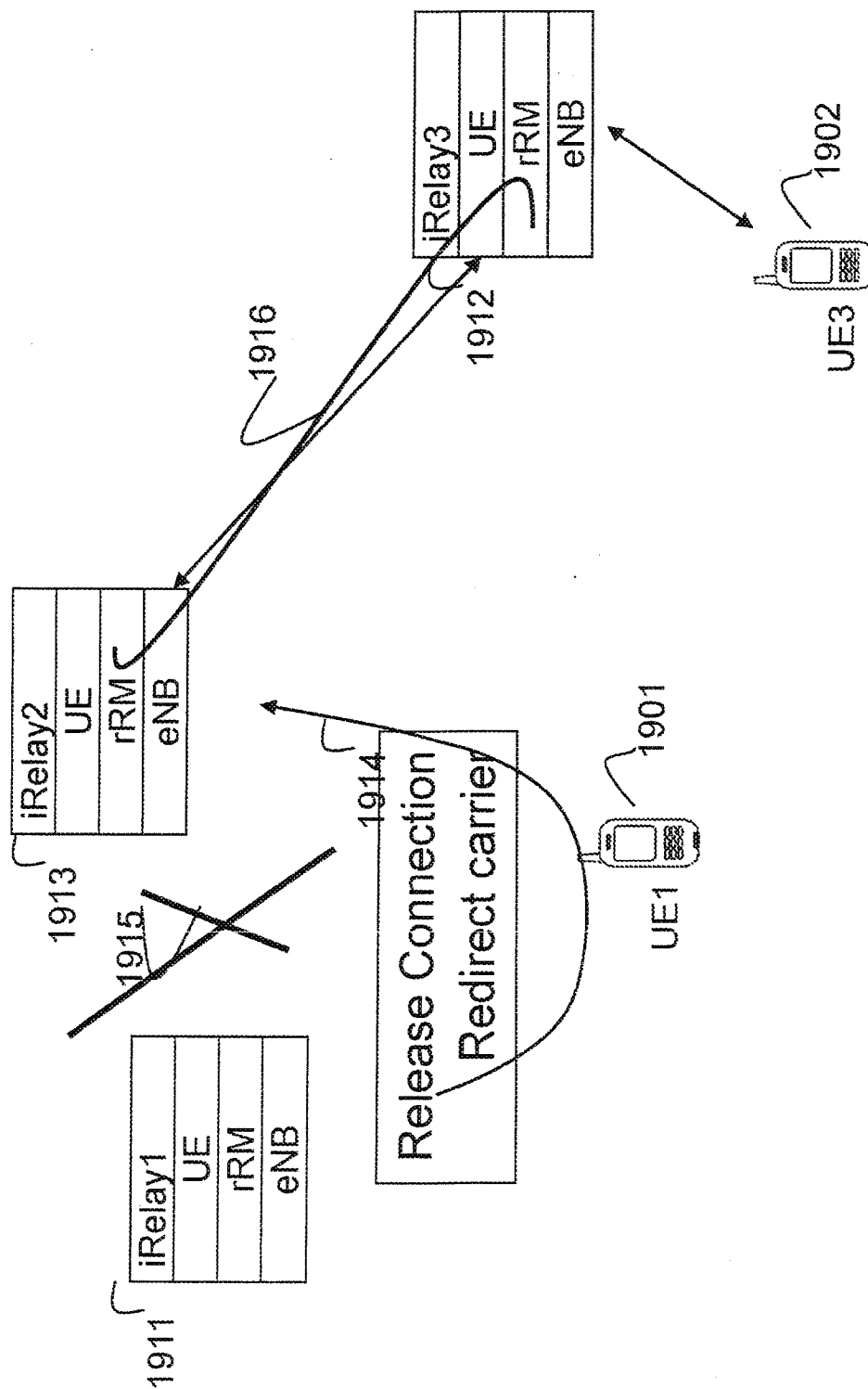


Fig 19e



39/51

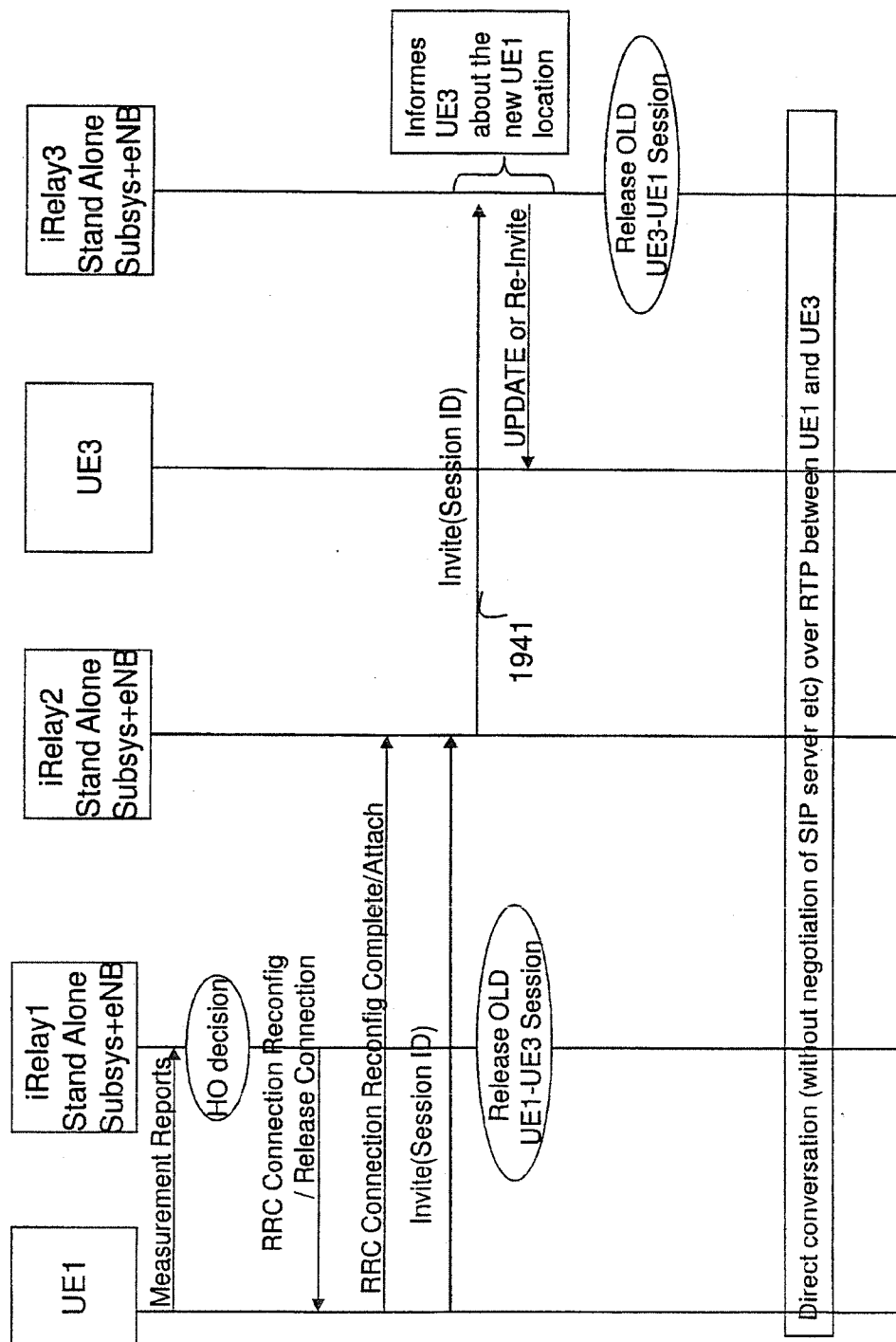


Fig. 19d

40/51

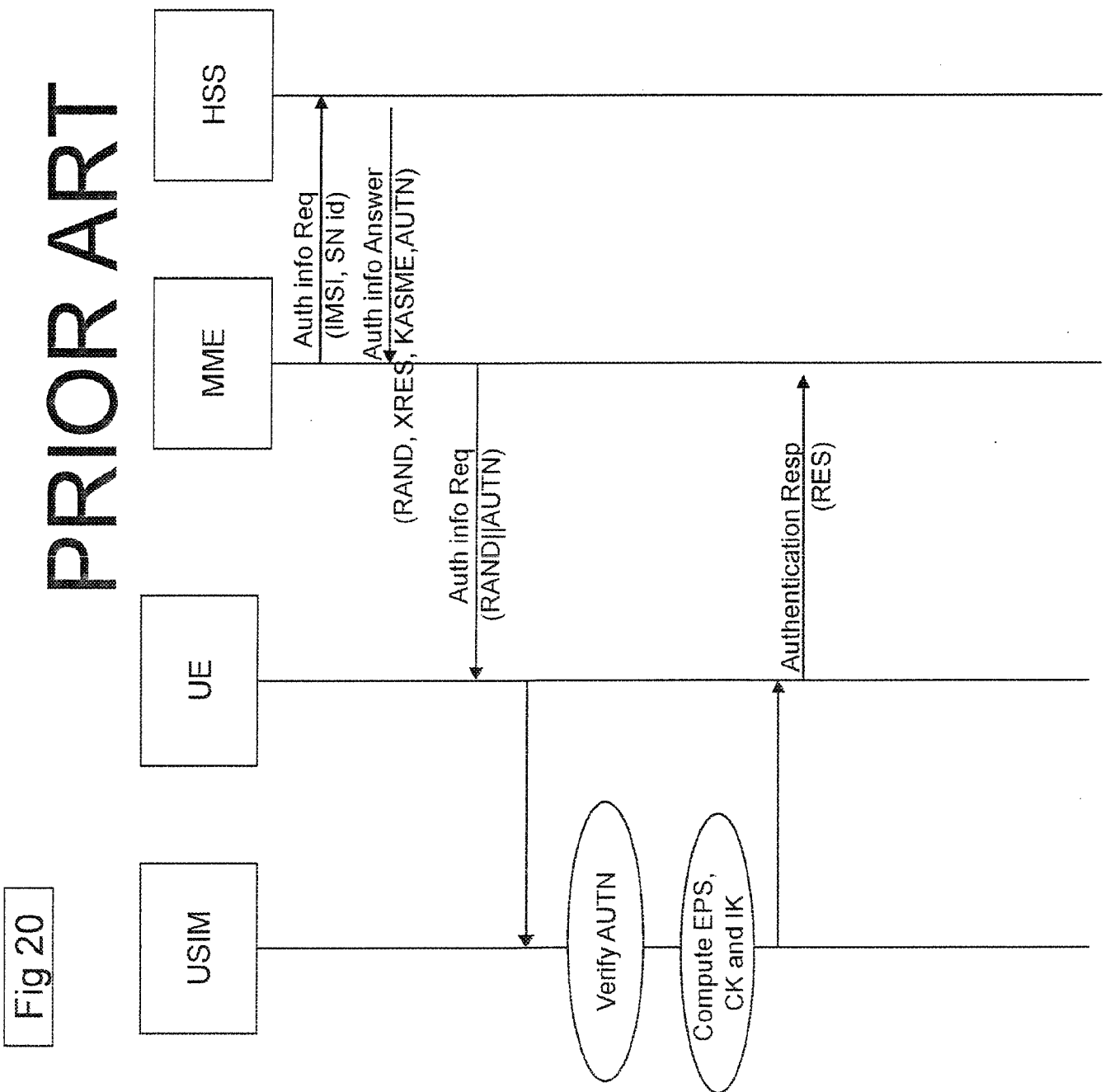


Fig 20a

PRIOR ART

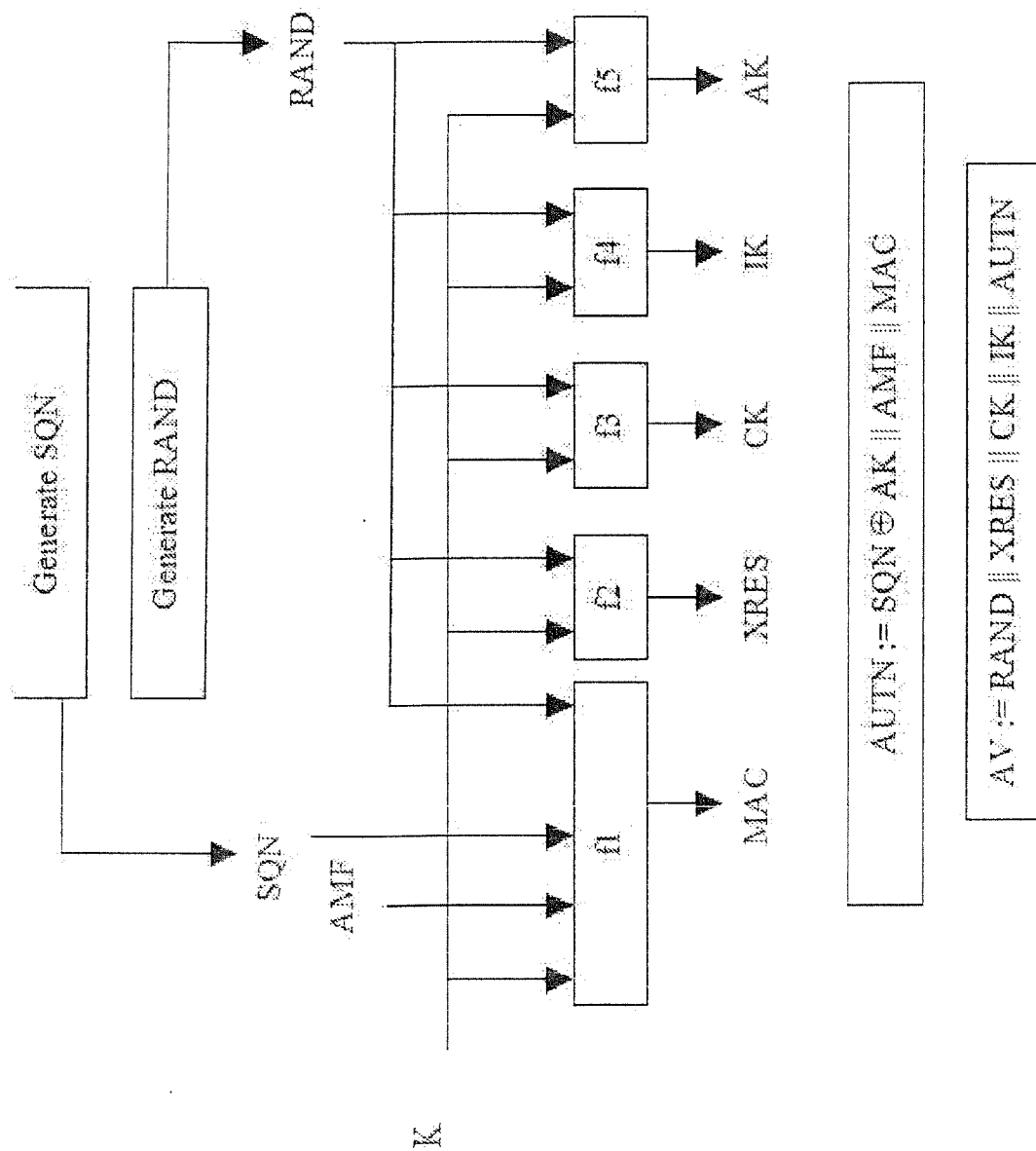
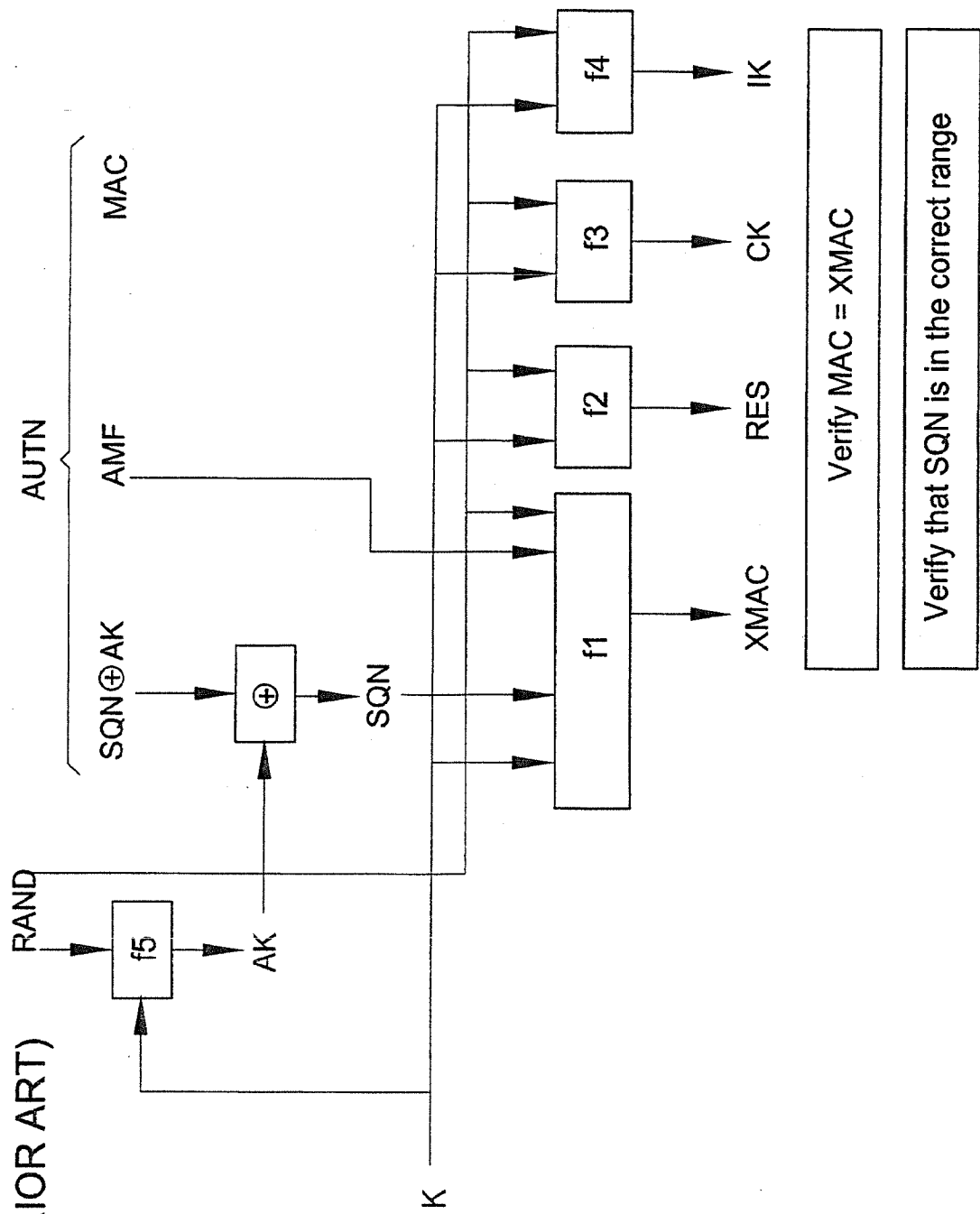
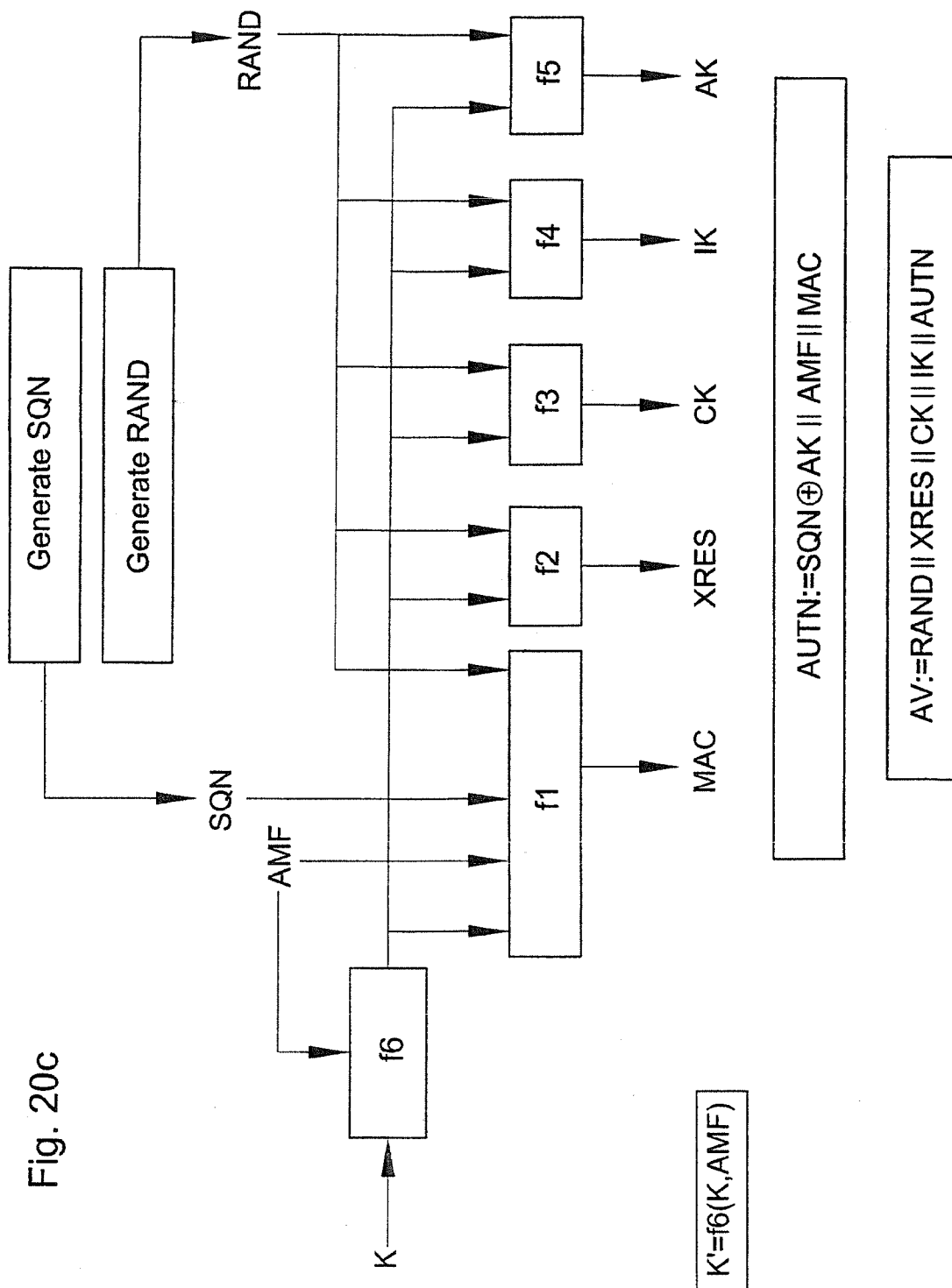


Fig. 20b (PRIOR ART)



43/51



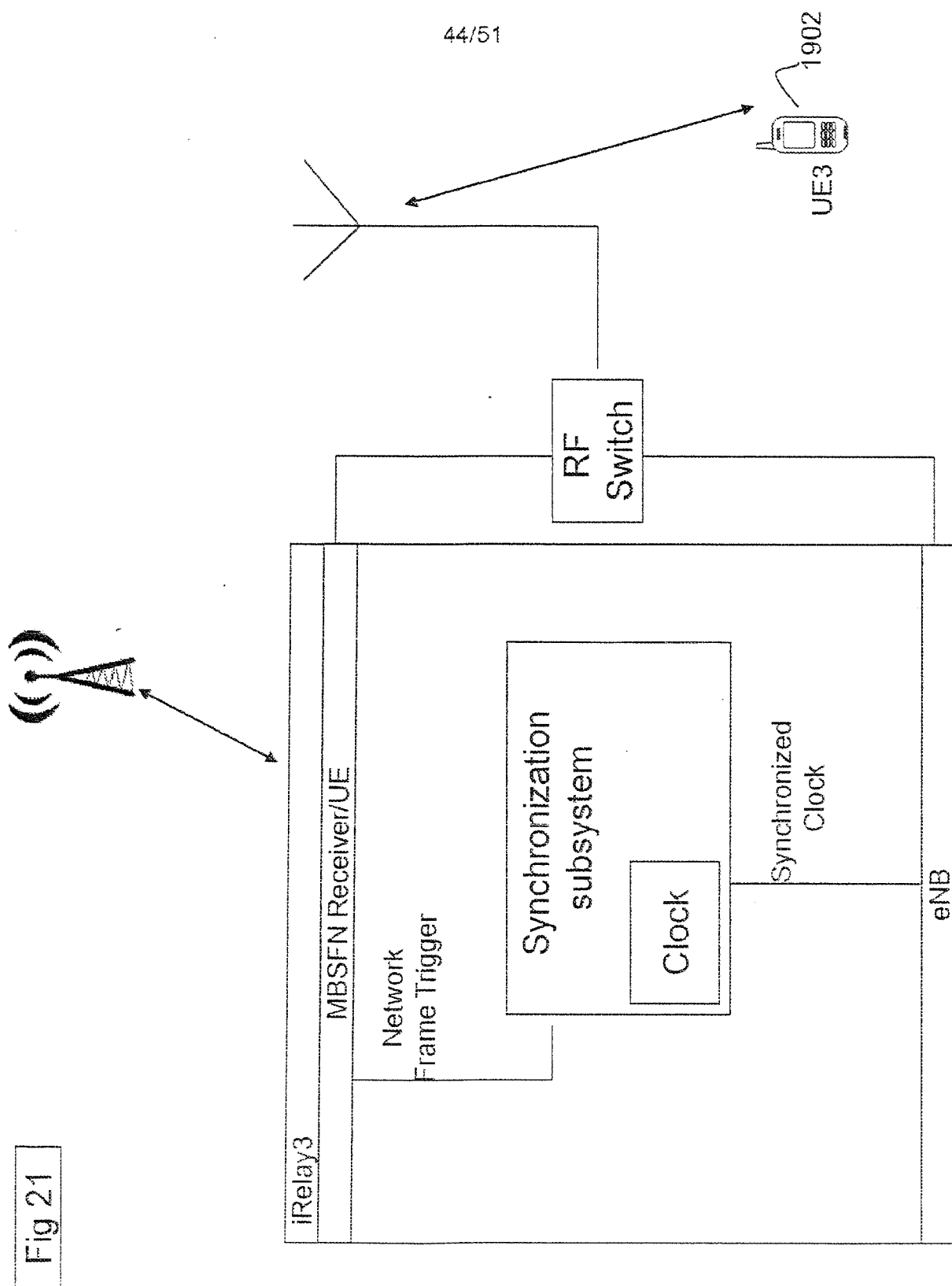


FIG. 17

2710: Each rUE scans for its neighboring cells and decodes its neighcellsList (see SIB4, IntraFreqNeighCellList in 3GPP TS 36.331) as published in the System information block and its cell ID.

2720: The rUE sends the list to the routing and Qos subsystem

2730: If the rBS published neighboring list is smaller than the decoded neighcellsList

Or published neighboring list is different than the decoded neighcellsList and current rBS cellID is smaller than the discovered base station then the routing agent triggers "handoff rue" methods that eventually will have the rUE attached to the new discovered base station and publishing its neighboring cells list; and

2740: The routing agent sends an update_neighbors message to the attached base station and all neighboring cells in the list, in order to add the collocated rBS to their neighboring list

FIG. 22

3010: Wait for send data queue is not empty

3020: listen for number of random() %
TOTAL_NUMBER_OF_SUBFRAMES

3030: if channel found busy, go to step 3010

3040: if not send all waiting data in the queue up to
MAXIMAL_ALLOWED_NUMBER_OF_SUBFRAMES_IN_
A_TRANSMISSION

3050: wait for ack from all the destinations in this transactions

3060: if ack received clear sent data from sending queue

3070: go to step 3010

46/51

Fig. 23:

1310: User A is connected to Server A (e.g. an application server providing images, maps or other special services) through core A,

1320: User A is handed over (BBM) due to RF considerations and is now connected to core B

1330: User A may re-register to Server A through Core B, or connect to a local replicated instance of Server A, Server B .

Fig. 24:

1410: User A is having a voip call to User B through core A and core B,

1420: Core A informs Core B that user A is being handed over to core C

1430: Core C invites User B from core B to hold an additional session so that the user experience will be uninterrupted communication.

1440: User A is handed over (MBB) due to RF considerations and is now connected to core C

1450: User A uses the additional session to continue his talk with user B, without dropping the session.

47/51

FIG. 25

1510: Wait for event (attach, update, disconnect)

1520: Switch event: If received attach message from client (local core = server) do method of Fig. 26a; If received attach message from mobile station functionality of relay (e.g. the modem has attached to a new BS and local core is serving as client) do method of Fig. 26B; If received update message do method of Fig. 26c; If received disconnect event from modem or socket do method of Fig. 26d

FIG. 26A

1605: receive "tunnel generation" message typically including routing table and metric from client

1610: Create ip tunnel e.g. GRE-standard tunnel; give tunnel a local address [address of self router] and remote address [address of client indicated in the attach message]

1615: Set up the ip tunnel

1620: Send ack message typ including local routing table + per-line time stamps + metrics to the client which sent the attach message

1625: dynamically build and distribute expanded routing table e.g. by using OSPF protocol or by performing method of Fig. 27

48/51

Fig. 26B

1630: receive attach message typically including routing table and metric e.g. from mobile station functionality, e.g. if modem has just attached to a base station

1635: Send "tunnel generation" message, typically including modem ip of server core to which it is desired to establish a tunnel, and routing table with per-line time stamp + metrics , to stationary core (e.g. EPC router – to functionality therewithin known to local client core to possess ip tunnel generation functionality)

1640: when ack message received from server core, set up ip tunnel and add address of the tunnel to local routing table

1645: dynamically build and distribute expanded routing table e.g. by using OSPF protocol or by performing method of Fig. 27

FIG. 26C

1650: receive update message including routing table typically with per-line time stamp + metrics

1655: Merge new routing table into current routing table; typically merge in only content whose time-stamp is newer than the relevant line in the routing table

1660: run Dijkstra like algorithm over current routing table + metrics to compute expanded routing table

1665: Update routing table +metrics accordingly and send updates message including the new routing table + timestamp to neighbors e.g. those nodes which are 1 hop away other than the neighbor which sent the update message

FIG. 26D

1670: receive disconnect event from client core (modem) or server core (socket) at remote end of ip tunnel linked to local mobile core

1675: Close local end of ip tunnel that has been linking local mobile core to modem or socket which sent disconnect event

1680: Remove all networks using ip tunnel's link (e.g. all lines in which next hop = that ip tunnel) from the routing table including metrics and send updates message including the new routing table + timestamp to the neighbors

FIG. 27

1710: Add the new link (tunnel) to local routing table typically with a time-stamp

1720: merge content (routing table with time stamps + metrics e.g.) of just-received (attach or ack) message into local routing table; typically merge in only content whose time-stamp is newer than the relevant line in the routing table

1730: run Dijkstra like algorithm over merged routing table + metrics to compute expanded routing table typically having more lines, each with time-stamp; each line of expanded routing table indicates how possible source-destination pair can communicate, including, when appropriate, via existing tunnels

1740: replace local routing table with expanded routing table

1750: send update message including expanded routing table to neighbors (e.g. all nodes one hop away other than neighbor which sent just-received message)

50/51

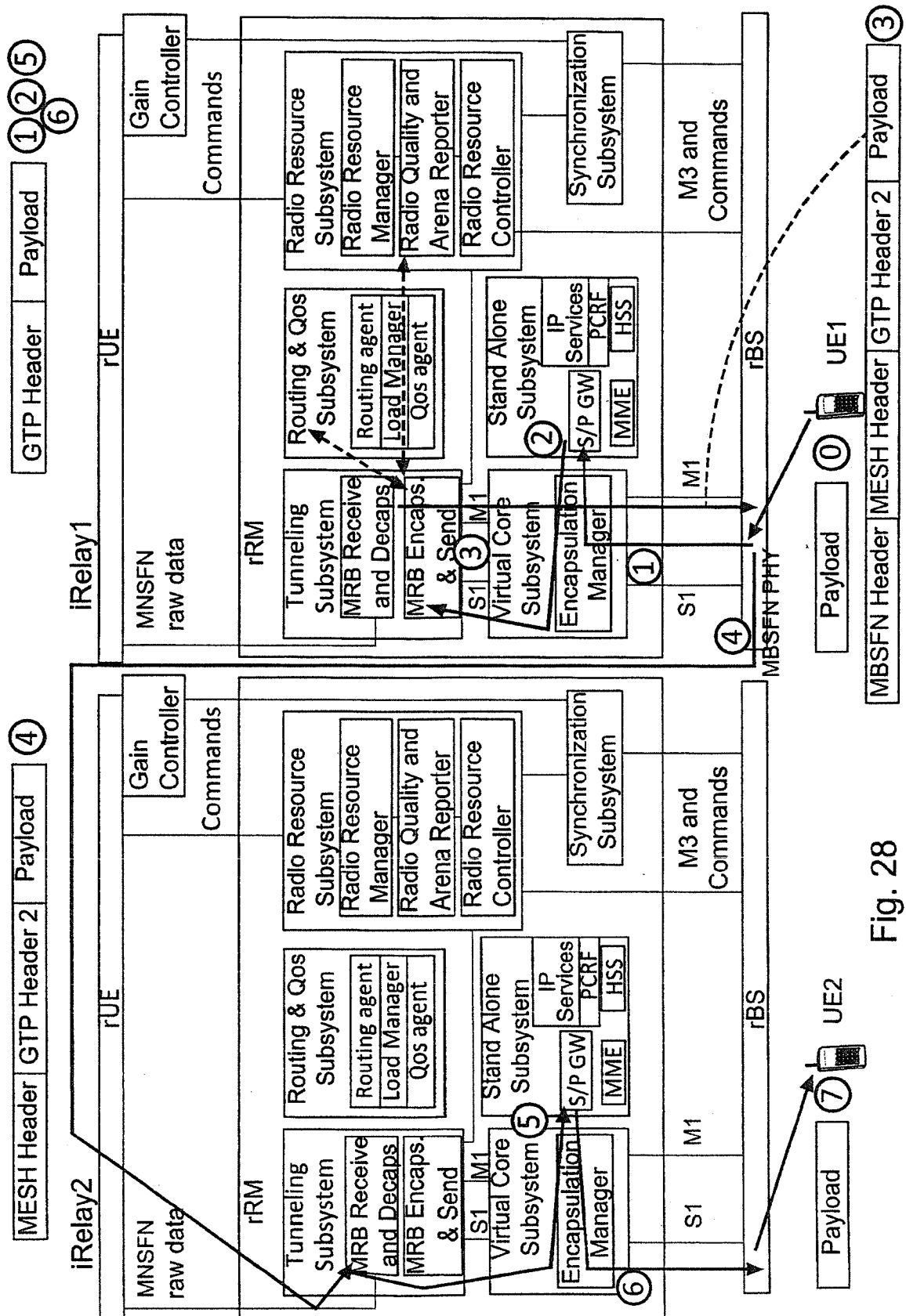


Fig. 28

51/51

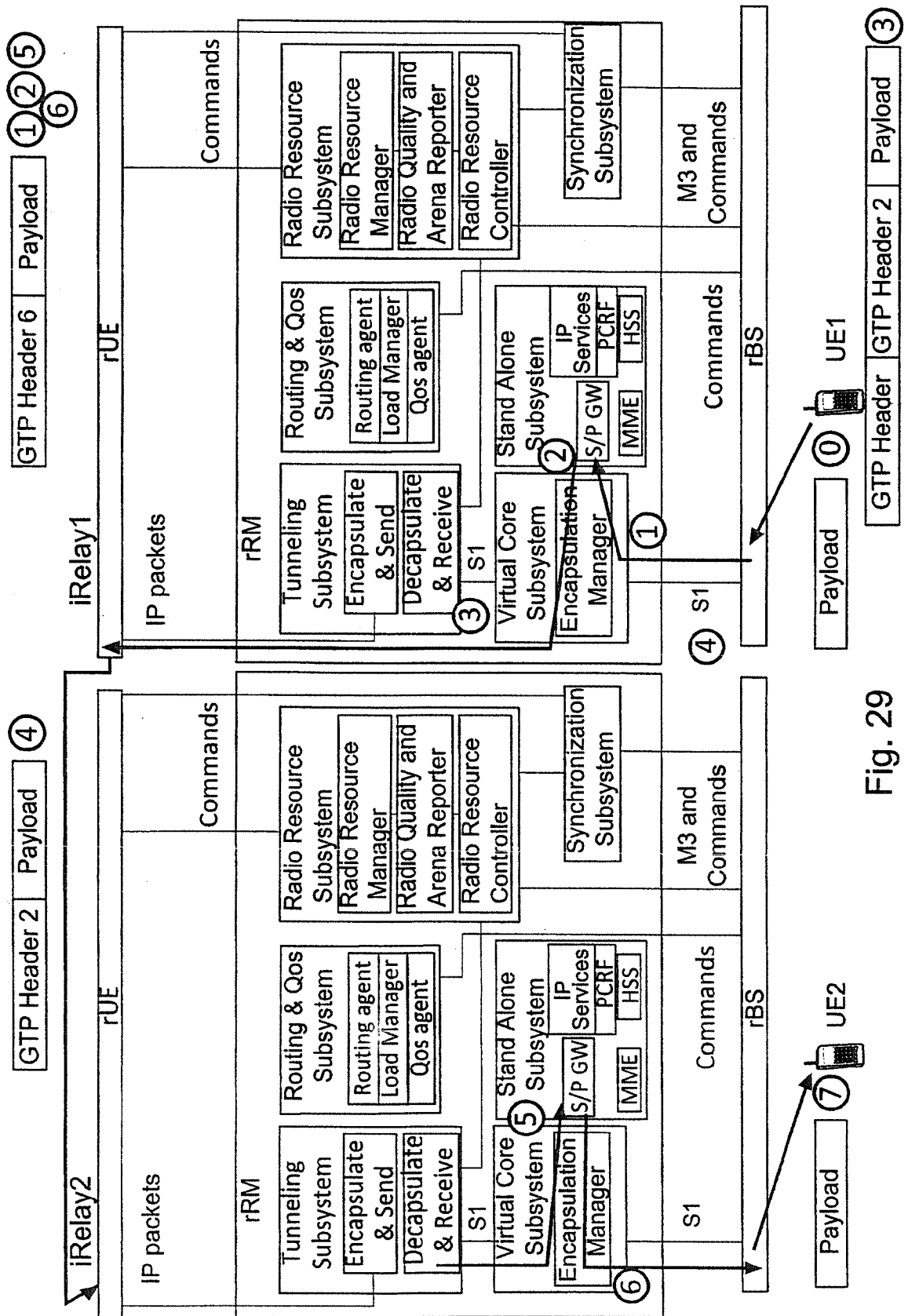


Fig. 29