connected to a "parent" communication node and to a number communication nodes, and wherein each communication node comprises:

means for receiving a message including information defining a message type and a message origin identifier;

a memory, storing:

data identifying respective devices associated with the "child" communication nodes;

data identifying a device associated with the "parent" communication node; and

a register relating each message type to a number of message destinations selected from the "child" communication nodes and the "parent" communication node;

and processor means operable to:

determine the message type of a received message;

determine one or more destinations for the received message, on the basis of the determined message type and the correlation stored in the register; and

forward the received message to the determined destination or destinations;

and wherein the processor is further operable to:

receive from a device associated with a child node an indication of a message type and message origin identifier;

store the specified message type and origin identifier in its register;

determine, from the message origin identifier, if a message of the specified type and origin will be received from a device associated with a child node; and

i. if it is determined that the message will be received from a child node, send a message to that child node to store the specified message type and origin identifier in a memory, in association with the address of the node; and

ii. if it is determined that the message will not be re
a child node, send a message to the parent node to store the specified message type and origin identifier in a memory, in association with the address of the node.

A second aspect of the invention provides system for transmitting a message from a sender to one or more recipients via at least one communication node, wherein each communication node has a "parent" communication port and to a number of "child" communication ports, and wherein each communication node comprises:

means for receiving a message including information defining a message type, and a message origin identifier;

a memory, comprising:

a first register storing:

data identifying one or more devices associated with respective "child" communication ports;

data identifying one or more device associated with the "parent" communication port; and

a second register storing:

data relating each message type to a number of message destinations selected from the devices associated with the "child" communication nodes and with the "parent" communication node;

and processor means operable to:

determine the message type of a received message;

determine one or more destinations for the received message, on the basis of the determined message type and the correlation stored in the second register; and

forward the received message to the determined destination or destinations.

In an advantageous embodiment, the message includes information c response requirement, and the processor is further operable to:

receive at a child port an indication of a message type and message origin identifier;

store the specified message type and origin identifier in its second register;

determine, from the message origin identifier, if a message of the specified type and origin will be received at a child port; and

iii. if it is determined that the message will be received at a child port, send a message to the device associated with that child port to store the specified message type and origin identifier in its memory, in association with the address of the node; and

iv. if it is determined that the message will not be received at a child port, send a message to the device associated with the parent node to store the specified message type and origin identifier in its memory, in association with the address of the node;

and wherein the processor is further operable, when it is determined that a received message requires a reply, to:

v. determine which of the child and parent ports the message was received at;

vi. extract from the message the message origin identifier; and

vii. add to the first register the message origin identifier in relation to the determined port.

Other aspects of the invention concern methods of transmitting messages over a communications network, and communication nodes for use in hierarchical communication systems.

Brief Description of the Drawings

An embodiment of the invention will now be described in detail with reference to the accompanying drawings, in which:

a particular area are all linked to a communications node A associate area, and the nodes associated with each area are linked to the communications node HO associated with the Head Office. Further strata in this hierarchical organisation are, of course, possible. For example, an "area" may represent a geographical area of a state, and casinos in a particular state or country may be all linked to a "country" communications node, with all countries in a region being linked to a "region" communications node. The "region" nodes may be linked to a "global headquarters" communication node.

Devices in the communication system are of four general types. Devices which produce originating messages are referred to as "message generators", and are illustrated in Figure 1 with the notation MG. These devices detect events, such as placing of a bet, arrival of a player at a table, dealing of a hand of cards at the table, outcome of a spin of a roulette wheel at the table, etc and generate a message indicative of the event.

Devices which receive messages produced by the message generators and produce an output such as a display on the basis of the received messages are referred to as "message consumers" and are illustrated in Figure 1 with the notation D. In the case of a roulette table, the message consumer device may for example be a display on the table indicating bets placed at the table, or a display confirming the outcome of a spin of the roulette wheel.

It will be appreciated that there may be devices which are both message generators and message consumers, and such devices will have the functionality of a both message generator device MG and a message consumer device D.

Communication nodes at the intermediate levels of the hierarchical system are all essentially identical in function, and each intermediate node is connected to one "parent" node at the next higher level in the hierarchy, and to one or more "child" nodes at the next lower level in the hierarchy. The intermediate node may receive

messages either from message generators directly, or from other con
nodes (child or parent) to which they are connected. For example, a "table" node may receive messages from message generators associated with the table node, or may receive messages from the "site" communications node associated with the casino in which the table is located.

The "head office" communications node HO receives messages from the "area" communications nodes as its "child" nodes, but has no "parent" node. The "head office" node may be the destination of messages, and it may also function to redistribute messages from one "area" communications node to one or more other "area" nodes. The "head office" node may also include the functionality of a message generator.

It will be understood that although the hierarchical organisation of the communications system may reflect the physical locations of the various devices, if the communications network can provide the appropriate communications path, then a communications node at any particular physical location may have its parent node at any other location, and may also have child nodes at different locations. For example, a node associated with a gaming table in a first casino may have as one of its child nodes a display device or a message generator situated at another casino. Likewise, a node associated with a gaming table in a first casino may have as its parent node a communications node in a location other than the first casino.

Figure 2 schematically illustrates the hierarchical structure of the connections between the display devices, message generators and communication nodes of the network such as that shown in Figure 1. In this example, a "head office" node HO is the parent node of two "area" nodes A1 and A2. Each area node A1, A2 is the parent of two "site" nodes, S11 and S12 being the site nodes which are child nodes of area node A1, and site nodes S21 and S22 being the child nodes of area node A2.

embodiments, have more than one trigger sensor 31, each trigger sens arranged to detect a particular event associated with a corresponding message type.

In operation, the trigger sensor 31 detects an event and sends a signal to the processor 32 indicating detection of the event and optionally also other relevant data concerning the event. The processor 32 retrieves from the memory the message type associated with that event, and the message generator 34 generates a message identifying the message type, the origin of the message, and optionally also containing the further relevant data as the message content. For example, the message may be message type M1 indicating that a bet has been placed, the origin address MG1112 indicating where the bet was placed, and the content of the message may be that a bet of a predetermined amount has been placed by a particular player or on a particular number. The assembled message is then sent, by the message sender 35, to the parent node to which the message generator 30 is connected, via an output connection 35a.

Structure of a message

Figure 4 illustrates the data structure of a typical message. The message contains a header part 41, which contains the origin address identifier 42 of the message generator which generated the message and a message type identifier 43. The message content 44 will contain the additional detail, if any. For example, if the "Message Type" data indicates that the message is of type M1 indicating that a bet has been placed, the message content 44 may identify the amount of the bet, the player, the table being played and the number or outcome on which the bet has been placed, for example "10 dollars placed by player in seat 1 at table 2 in Casino 3 on roulette number 17". Alternatively, if the message is simply to indicate that a player has taken a seat at the table, in order to enable another device on the network to keep a count of how many players are present at that table, the message type data and the message origin may be sufficient to uniquely

alarm. Processor 63 is also linked to a memory 65 in which ar identifier 66 which uniquely identifies the message consumer, a registration routine 67 which is executed when the consumer is connected to its parent node, and a look-up table 68 which identifies a number of message types and message origins, and corresponding outputs for each combination of message type and origin. It will be understood that, if the "message type" information includes data identifying the origin of a particular message of this type, then the look-up table 68 will only require to store information relating to message type and corresponding output.

The unique address of the parent node to which the message consumer 60 is connected is also stored in the memory 65, at 69.

In operation, an incoming message is received via the input port 61 of the message consumer 60 and interrogated by the message interrogator 62, which identifies the message type and message origin from the message header data. Processor 63 then retrieves from the look-up table 68 in memory 65 the appropriate action to be taken by the output device in response to receipt of that message, and causes the required output to be given by the output device 64.

Setting up a message generator

The initial set-up of the message generator consists in firstly configuring the trigger sensor so that it can detect the event or events in respect of which the message generator is intended to generate messages. This may be done by temporarily connecting an external input device 100 such as a portable computer to the message generator. The trigger sensor 31 may be configured to receive an output, for example from a ball detector of a roulette wheel, so that the trigger sensor is notified when the ball has entered a pocket in the roulette wheel and can generate a corresponding output. Alternatively, the trigger sensor may receive an output from a pressure sensor on a card or roulette table which detects the placing

of a gaming chip on the table, so that the trigger sensor can generate a signal to indicate that a bet has been placed.

Next, data may be input from the input device 100 to the look-up table 37 in memory 33 identifying the events which the trigger sensor will detect, and the message types which are to be produced by the message generator in response to the detection of those events. The external input device 100 may also input into the memory 33 a network address such as "MG1112" or other identifying data relating to the message generator device.

Connecting a message producer to a node

The registration routine for connecting a message generator such as message generator MG1112 to a node such as T111 will now be described with reference to Figure 7.

A physical or other communication connection between the message sender 35 of the message generator 30 and the message interrogator 57 of the node 50 is established via the output port 35a of the message generator, for example by connecting a cable or by establishing a wireless or infrared connection.

When communication is established, the communication node 50 sends a message to the message generator 30 identifying the address T111 of the communication node 50, which address the processor 32 of the message generator stores in its memory 39 as its "parent address".

Message generator 30 sends a message to the node 50, identifying the address or other unique identifier MG1112 of the message generator, and requesting that the address or identifier of the message generator is added in the list of child nodes in the register 53 of the node 50. The node 50 receives this message, and the processor 51 as part of the registration routine identifies the child port at which

the message arrived, and adds the identifier MG1112 to the list associated with that child port in the register 53. In the illustrated embodiment, the message generator 30 is connected to child port C1 of the table node 50, and thus in the register 53 the identifier MG 1112 is recorded against the port C1.

The message generator 30 also sends information identifying the types of message M1, M2....Mn which the message generator can produce. In this example, the message generator 30 is able to produce message type MI indicating that a bet has been placed, and message type M2 indicating that a player has taken a seat at the table. In this example, message type M1, M2 includes data identifying the message generator MG1112 as the origin of the message, and thus message type MI indicates a message from the message generator MG 1112 indicating that a bet has been placed. Alternatively, the origin information may be added to the message header independently of the message type data. The processor 51 of the node 50 checks whether each message type is already present in its look-up table 55, and if it is not then the processor 51 then adds the message type to its look-up table 55. At this stage in the registration process, the look-up table 55 includes only identifiers of message type/origin M1, M2, etc, but does not yet include any addresses to which these message types should be sent. The addition of this information to the look-up table 55 comes later in the process of setting up the network.

When all of the message types M1, M2, etc, that the message generator 30 can produce have been added to the look-up table 55, and the address of the message generator MG1112 has been added to the list of child nodes in the register 53, then the registration routine in the node 50 is complete. The registration routine of the message generator 30 is complete when all this information has been sent to the parent node 50, and the address of the parent node T111 has been recorded in the memory 33 of the message generator.

The input device 100 may also be used to store in the memory 65 an 66, such as "D1111", for the message consumer to identify it to other network devices, in particular to its parent node. The appropriate connections to display devices etc are then made from the output device so that the required actions can be performed on receipt of their respective messages.

Connecting a message consumer to a node

The registration routine for connecting a message consumer such as D1111 to a node such as T111 will now be described with reference to Figure 8.

To connect the message consumer 60 (for example D1111) to a node 50 of the communications network, a physical or other communication connection between the message interrogator 62 of the message consumer 60 and the message sender 58 of the node 50 is established via the input port 61 of message consumer 60, for example by connecting a cable or by establishing a wireless or infrared connection. In this example, the message generator MG 1112 has already been connected to the child port C1 of the node 50, and thus the message consumer 60 is connected to the child port C2 of the node 50.

When communication is established, the communication node 50 sends a message to the message consumer 60 identifying the address of the communication node 50 (for example T111), which the processor 63 of the message consumer stores in its memory 65 as its "parent address" 69.

The message consumer 60 sends its address identification 66 (in this case D1111) to the node 50, and requests to be added to the list of child nodes in the register 53 of the node 50. Processor 51 of the node 50 then identifies which child port the device is connected to (in this case C2), adds the address ID to the list of devices associated with that child port in the register 53 of its memory 52. In this example, message generator MG 1112 is already present in the list, associated

with child port C1. The message consumer D1111 is entered in the associated with child port C2.

In the look-up table 68 of the message consumer 60, the message types for which an output from the message consumer 60 is required are listed. In the registration routine, each of these message types is sent to the node 50. In the example seen in Figure 6, the message consumer 60 has been set up so that in response to a message type M1 from an origin address MG111, the consumer performs an output identified as "Output 1". Likewise, in response to a message type M2 from an origin address MG1122, the consumer performs an output identified as "Output 2".

When the message type information is received at node 50, the processor 51 firstly determines whether or not that message type (i.e. that combination of message type and origin) is already present in its look-up table 55. If the message type is already present in the look-up table 55, then the address ID of the message consumer D1111 is added to the look-up table 55 in relation to that message type.

The processor 51 then determines whether the message type has an origin ID address that matches an address stored in the register 53 of the node 50 and associated with one of the child ports C1, C2 of the node 50, i.e. determines whether a message required to be sent to the message consumer 60 in fact originates at another child node of the communications node 50.

If the message type originates from one of the child nodes of the node 50, such as MG1112, then the message type will already be present in the look-up table 55 (having been stored there during the registration process for the message generator) and the registration routine will only require the address of the message consumer D1111 to be added to the look-up table 55 in relation to that message type.

In the example of Figure 8, the message consumer D1111 requires messages of type M2 from the message generator MG 1122. Since message type M2 is already present in the look-up table 55 of table node 50 following the registration process of message generator MG 1122 described in relation to Figure 7, the address of the message consumer D1111 is added in the look-up table in association with message type M2.

If the message type such as M1, M3 or M4 is not already present in the look-up table 55 indicating that this message type does not originate with a child node of this parent node 50, the processor 51 makes a new entry in the look-up table 55, recording the message type (for example M1 originating from MG111), and the address D1111 of the message consumer 60 to which it should be sent. Additionally, the processor 51 of the node 50 sends a message to its parent node (in this case node S11, as seen in Figure 2) requesting that the message type M1 originating from MG111 with the destination address T111 of node 50 be added to the look-up table 55 in the parent node S11. Likewise, message types M3 originating from MG2211 and M4 originating from M1212 are added to the look-up table 55 in the parent node S11, each associated with the address T111 of node 50.

The registration process continues at the parent node S11, where the processor determines whether or not each message type/origin received from node T111 is already present in the look-up table of the node S11. If it is not, then the processor adds the message type/origin to the lookup table 55 of node S11, in association with the address T111 of the node 50. The processor of the parent node S11 then determines whether the origin address of this message type is associated with one of its child ports, and if it is not the node S11 sends a message to its parent node (in this case node A1, as seen in Figure 2) requesting that the message type/origin be added to the look-up table 55 of the parent node A1, in association with the address S11.

message destination from its LUT 55 to site node S11. For each comb node S11 first determines whether the message type is already in its LUT 55.

If the message type is not already in the LUT 55 of site node S11, then the message type is added.

If the message type is already in the LUT of site node S11, then the site node determines from the origin address information associated with that message type whether the message type will reach site node S11 from an address on the list of child nodes of node S11, i.e. checks whether the origin address is "local" to the node S11. If the origin address is local, then the site node adds the destination address to its LUT in relation to this message type.

If the site node determines from the origin address information that this message type will not arrive from a child node of the site node S11, the site node sends a message to its parent node (the area node A1) requesting that this message type be added to the LUT of the parent node A1, with the destination address set as site node S11.

The processing moves on to the next message type found in the LUT of table node T1.

Figure 10 is a flowchart illustrating the process steps taken at a parent node N when connecting a child node N-1 to the parent node N.

At step 101, the parent node receives an identifier from the child node, and at step 102 the identifier is added to the list of child nodes held in the register 53 of the parent node N.

The parent node N then sends its identifier to the child node N-1, and the child node records the identifier as the address of its parent node in its register 53.

The parent node then interrogates the child node at step 112 to determ any other message types are stored in its LUT, and if so requests the child node to send the next one whereupon the processing resumes at step 104.

When the child node has notified the parent node of all message types stored in the LUT of the child node, the processing ends.

Each time a child node has an entry added to its LUT, or has an existing entry in its LUT altered, the relevant entry is notified to its parent node which then carries out processing steps 104 to 112 of Figure 10.

With reference to Figure 11, an exemplary part of a network will now be described. In the network shown in Figure 12, a site node S1 has two table nodes T1 and T2 connected to it as "children". Table node T1 has connected to it a first message generator MG1 and a first display unit D1. Table node T2 has connected to it a second message generator MG2.

The first message generator MG1 produces a message type M1.1 in response to a first event E1, and produces a message type M1.2 in response to a second event E2.

The second message generator MG2 produces a message type M2.1 in response to a third event E3, and produces a message type M2.2 in response to a fourth event E4.

Display device D1 is required to produce an output action A1 in response to a message of type M1.1 from message generator MG1, and to produce an output action A2 in response to a message of type M2.2 from message generator MG2.

When the first message generator MG1 is connected to child port C1 of table node T1, the message generator MG1 stores its parent address T1 and requests table

node T1 to add MG1 to its list of child nodes. MG1 then sends its m information to T1, and T1 adds to its lookup table the message types M1.1 and M1.2, at this stage without any corresponding "send to" addresses.

Display device D1 is then connected to child port C2 of table node T1, and T1 sends its address to D1 for storing as its parent address. D1 sends its address identification to T1 and requests to be added to the list of child nodes of T1. D1 then sends to T1 the message type information M1.1 and M2.2 identifying the messages which are to be sent to D1 for action.

When D1 sends message type M1.1, T1 determines that message type M1.1 is already present in the LUT of T1, T1 determines from the origin address information relating to message type M1.1 that this message type will be received at T1 from its child node MG1, and that the destination address D1 is also on its list of child nodes, and simply adds the address D1 in relation to message type M1.1 in its LUT. T1 may also send a message to child node MG1 indicating that an interest in message type M1.1 has been registered (because a "send to" address D1 is recorded in the LUT in relation to message type M1.1). Message generator MG1 may be configured so that when event E1 is detected, message M1.1 is only sent if message generator MG1 has been notified of the registration of an interest in message type M1.1, i.e the message is only sent if T1 has notified MG1 that there is a "send to" address (D1) recorded in the LUT of T1 in relation to message type M1.1.

D1 then sends message type M2.2 to T1. This message type is not already present in the LUT of T1, so T1 then adds the message type M2.2 and a corresponding destination address D1 to its LUT. T1 then determines from the origin address information relating to message type M2.2 whether this message type will be received at T1 from a child of T1. Finding that it will not, T1 sends a message to S1 requesting that message type M2.2 be added to the LUT of S1, with the corresponding "send to" destination address T1.

In operation, occurrence of event E4 causes message generator MG message of the message type M2.2 to its table node T2. Table node T2 receives the message, retrieves the address S1 from its LUT and forwards the message on to site node S1 because the LUT of T2 has S1 as the destination address for message type M2.2.

The message arrives at site node S1, which redirects it to table node T1 on the basis of the "send to" destination address held in the LUT of S1. Likewise, when the message is received at table node T1, the message is forwarded to display device D1 on the basis of the "send to" destination address D1 held in the LUT of T1 for message type M2.2. Receiving message type M2.2, display device D1 performs the action A2 which may be to display an indicator lamp or play a video or make another output.

Likewise, when event E1 is detected by message generator MG1, then message type M1.1 is sent to table node T1, whence it is redirected to display device D1 on the basis of the "send to" destination address D1 held in the LUT of T1 for message type M1.1. Display device D1 performs action A1 in response to receipt of the message type M1.1.

Messages requiring responses

The above description relates to messages which are sent out from message generators, but do not require responses from their recipients. The invention foresees that the communications network may also deal with messages which require the terminal receiving the message to send a response back to the originator of the message.

In the case of such messages, the message header includes data indicating that a response is required, in addition to data indicating the origin of the message, and the nature of the response required. With reference to Figure 11, an example will

now be described in which a message of type M2.2 is sent fro generator MG2 to message consumer D1, requiring a response to be sent from message consumer D1 back to message generator MG2.

Message generator MG2 assembles the required message, placing in the message header the source address MG2, a "response required" flag, and message type information M2.2. The message is then sent via the parent port P of message generator MG2, and arrives at the child port C1 of node T2.

The processor of T2 retrieves from the message header the address and information relating to the source of the message, and determines that a reply is required by detecting the "response required" flag. The port C1 on which the message arrived is also identified. The node T2 then adds the identification of the source of the message (MG2) to the entry relating to port C1 list in its registry 53 which maintains a relationship between the ports of the node T2 and devices associated with each of those ports. The list of devices and respective message arrival ports is updated, irrespective of whether the arrival port is a child port or the parent port.

The message is then forwarded in accordance with the "send to" address for that message type recorded in the look-up table of the node T2, which in this case sends the message from parent port P of node T2 to the C2 child port of node S1.

The processor of S1 retrieves from the message header the address and information relating to the source of the message, and determines that a reply is required by detecting the "response required" flag. The port C2 on which the message arrived is also identified. The node S1 then adds the identification of the source of the message MG2 against the port C2 on the list in its registry 53 which maintains a relationship between the ports of the node S1 and devices associated with each of those ports. The node S1 then forwards the message via its port C1 to node T1, in accordance with the "send to" address recorded in its look-up table.

player, and as a message consumer. For example, each player positio
a display screen which shows information relating to the particular player, such as his remaining account balance, the wager he has currently placed, the wagers of other players at the table, and optionally other information such as the results of the last ten spins of the roulette wheel.

The roulette wheel 121, the player positions 123 and the display 122 are all linked to the table communications node 124, so that gaming software, for example resident at the table computer 125 linked to the table node 124, can control operation of the roulette wheel 121 and resolve wagers placed by the players when each spin of the wheel results in the ball settling in a pocket. The table computer at 125 may be linked to more than one table node 124, and thus may control more than one roulette table 120.

When setting up the communication system, the display 122 is set up as a message consumer which receives messages from the player positions 123a, 123b etc in order to display each player's wager for the current game. Table node 124 has a look-up table in which message types of originating from the player positions and including wager information are listed, with the display 122 as their destination address. The display 122, in response to receiving those messages displays images corresponding to the wagers placed by the players. The player's wager messages may also be sent to the table computer 125, by adding the network address of the table computer 125 to the LUT in the table node 124 in relation to these messages.

The roulette wheel 121 is configured as a message generator, and has a ball sensor for generating messages signifying that the ball has settled in a particular pocket of the roulette wheel. These messages may be routed through the table node 124 to the table computer 125, where they are compared with wager information received from the player positions in order to determine how each player's wager should be resolved.

It will be understood that the wager messages from player position already have an entry or entries in the LUT of table node 124A, so that the wager messages can be sent to the table computer 125. The table node 124A simply adds a further "send to" address in the LUT for each of the required messages so that they are additionally forwarded to the casino node S.

Likewise, if a player in a first casino wishes to see the wagers being placed by an acquaintance in another casino, he may make an entry in the LUT of his player terminal 123a, which will alter the LUT in the table node 124 and in the casino node S. The casino node S at the first casino will then send a message to the next node up in the hierarchy, the area node, which will then add these message types to its LUT with a destination address of the casino node S. The area node will then identify the source of the messages required and determine whether the messages will reach the area node from a child of the area node (i.e. from another casino within the area). If it is determined that they will, the area node notifies that child node (i.e. the other casino) to direct messages of this type to the area node. The casino node at the other casino determines which table node it expects the messages to arrive from and notifies that table node to add the casino node to its LUT in respect of these messages. The table node will already have an entry in its LUT in respect of these messages from the player position and thus simply adds the casino node address as a further destination address in its LUT.

When these operations are completed the relevant messages from the player position at the other casino will be sent to its table node, forwarded to the casino node of the other casino, forwarded to the area node and then returned to the casino node at the first casino and then to the table node of the interested player and eventually to the interested player's position.

It will thus be understood that by connecting a message consumer device to any node in the system and specifying the type and source of messages which that consumer device shall receive, the necessary modifications to the LUTs of

communications nodes are made, firstly moving up the hierarchical s
node is reached which is common to both the message consumer and the message origin, and then moving down from this common node to the message origin node or device. In each case, the message passes up the hierarchy from the message generator only as far as it needs to in order to be able to come back down to the relevant message consumer.

connected to a "parent" communication node and to a number of "child" communication nodes, and wherein each communication node comprises:

means for receiving a message including information defining a message type and a message origin identifier;

a memory, storing:

data identifying respective devices associated with the "child" communication nodes;

data identifying a device associated with the "parent" communication node; and

a register relating each message type to a number of message destinations selected from the "child" communication nodes and the "parent" communication node;

and processor means operable to:

determine the message type of a received message;

determine one or more destinations for the received message, on the basis of the determined message type and the correlation stored in the register; and

forward the received message to the determined destination or destinations;

and wherein the processor is further operable to:

receive from a device associated with a child node an indication of a message type and message origin identifier;

store the specified message type and origin identifier in its register;

determine, from the message origin identifier, if a message of the specified type and origin will be received from a device associated with a child node; and

i. if it is determined that the message will be received from a child node, send a message to that child node to store the specified message type and origin identifier in a memory, in association with the address of the node; and

ii. if it is determined that the message will not be received from a child node, send a message to the parent node to store the specified message type and origin identifier in a memory, in association with the address of the node.

A second aspect of the invention provides system for transmitting a message from a sender to one or more recipients via at least one communication node, wherein each communication node has a "parent" communication port and to a number of "child" communication ports, and wherein each communication node comprises:

means for receiving a message including information defining a message type, and a message origin identifier;

a memory, comprising:

a first register storing:

data identifying one or more devices associated with respective "child" communication ports;

data identifying one or more device associated with the "parent" communication port; and

a second register storing:

data relating each message type to a number of message destinations selected from the devices associated with the "child" communication nodes and with the "parent" communication node;

and processor means operable to:

determine the message type of a received message;

determine one or more destinations for the received message, on the basis of the determined message type and the correlation stored in the second register; and

forward the received message to the determined destination or destinations.

In an advantageous embodiment, the message includes information concerning a response requirement, and the processor is further operable to:

receive at a child port an indication of a message type and message origin identifier;

store the specified message type and origin identifier in its second register;

determine, from the message origin identifier, if a message of the specified type and origin will be received at a child port; and

iii. if it is determined that the message will be received at a child port, send a message to the device associated with that child port to store the specified message type and origin identifier in its memory, in association with the address of the node; and

iv. if it is determined that the message will not be received at a child port, send a message to the device associated with the parent node to store the specified message type and origin identifier in its memory, in association with the address of the node;

and wherein the processor is further operable, when it is determined that a received message requires a reply, to:

v. determine which of the child and parent ports the message was received at;

vi. extract from the message the message origin identifier; and

vii. add to the first register the message origin identifier in relation to the determined port.

Other aspects of the invention concern methods of transmitting messages over a communications network, and communication nodes for use in hierarchical communication systems.

Brief Description of the Drawings

An embodiment of the invention will now be described in detail with reference to the accompanying drawings, in which:

a particular area are all linked to a communications node A associated with that area, and the nodes associated with each area are linked to the communications node HO associated with the Head Office. Further strata in this hierarchical organisation are, of course, possible. For example, an "area" may represent a geographical area of a state, and casinos in a particular state or country may be all linked to a "country" communications node, with all countries in a region being linked to a "region" communications node. The "region" nodes may be linked to a "global headquarters" communication node.

Devices in the communication system are of four general types. Devices which produce originating messages are referred to as "message generators", and are illustrated in Figure 1 with the notation MG. These devices detect events, such as placing of a bet, arrival of a player at a table, dealing of a hand of cards at the table, outcome of a spin of a roulette wheel at the table, etc and generate a message indicative of the event.

Devices which receive messages produced by the message generators and produce an output such as a display on the basis of the received messages are referred to as "message consumers" and are illustrated in Figure 1 with the notation D. In the case of a roulette table, the message consumer device may for example be a display on the table indicating bets placed at the table, or a display confirming the outcome of a spin of the roulette wheel.

It will be appreciated that there may be devices which are both message generators and message consumers, and such devices will have the functionality of a both message generator device MG and a message consumer device D.

Communication nodes at the intermediate levels of the hierarchical system are all essentially identical in function, and each intermediate node is connected to one "parent" node at the next higher level in the hierarchy, and to one or more "child" nodes at the next lower level in the hierarchy. The intermediate node may receive

messages either from message generators directly, or from other communication nodes (child or parent) to which they are connected. For example, a "table" node may receive messages from message generators associated with the table node, or may receive messages from the "site" communications node associated with the casino in which the table is located.

The "head office" communications node HO receives messages from the "area" communications nodes as its "child" nodes, but has no "parent" node. The "head office" node may be the destination of messages, and it may also function to redistribute messages from one "area" communications node to one or more other "area" nodes. The "head office" node may also include the functionality of a message generator.

It will be understood that although the hierarchical organisation of the communications system may reflect the physical locations of the various devices, if the communications network can provide the appropriate communications path, then a communications node at any particular physical location may have its parent node at any other location, and may also have child nodes at different locations. For example, a node associated with a gaming table in a first casino may have as one of its child nodes a display device or a message generator situated at another casino. Likewise, a node associated with a gaming table in a first casino may have as its parent node a communications node in a location other than the first casino.

Figure 2 schematically illustrates the hierarchical structure of the connections between the display devices, message generators and communication nodes of the network such as that shown in Figure 1. In this example, a "head office" node HO is the parent node of two "area" nodes A1 and A2. Each area node A1, A2 is the parent of two "site" nodes, S11 and S12 being the site nodes which are child nodes of area node A1, and site nodes S21 and S22 being the child nodes of area node A2.

embodiments, have more than one trigger sensor 31, each trigger sensor 31 being arranged to detect a particular event associated with a corresponding message type.

In operation, the trigger sensor 31 detects an event and sends a signal to the processor 32 indicating detection of the event and optionally also other relevant data concerning the event. The processor 32 retrieves from the memory the message type associated with that event, and the message generator 34 generates a message identifying the message type, the origin of the message, and optionally also containing the further relevant data as the message content. For example, the message may be message type M1 indicating that a bet has been placed, the origin address MG1112 indicating where the bet was placed, and the content of the message may be that a bet of a predetermined amount has been placed by a particular player or on a particular number. The assembled message is then sent, by the message sender 35, to the parent node to which the message generator 30 is connected, via an output connection 35a.

Structure of a message

Figure 4 illustrates the data structure of a typical message. The message contains a header part 41, which contains the origin address identifier 42 of the message generator which generated the message and a message type identifier 43. The message content 44 will contain the additional detail, if any. For example, if the "Message Type" data indicates that the message is of type M1 indicating that a bet has been placed, the message content 44 may identify the amount of the bet, the player, the table being played and the number or outcome on which the bet has been placed, for example "10 dollars placed by player in seat 1 at table 2 in Casino 3 on roulette number 17". Alternatively, if the message is simply to indicate that a player has taken a seat at the table, in order to enable another device on the network to keep a count of how many players are present at that table, the message type data and the message origin may be sufficient to uniquely

alarm. Processor 63 is also linked to a memory 65 in which are stored an identifier 66 which uniquely identifies the message consumer, a registration routine 67 which is executed when the consumer is connected to its parent node, and a look-up table 68 which identifies a number of message types and message origins, and corresponding outputs for each combination of message type and origin. It will be understood that, if the "message type" information includes data identifying the origin of a particular message of this type, then the look-up table 68 will only require to store information relating to message type and corresponding output.

The unique address of the parent node to which the message consumer 60 is connected is also stored in the memory 65, at 69.

In operation, an incoming message is received via the input port 61 of the message consumer 60 and interrogated by the message interrogator 62, which identifies the message type and message origin from the message header data. Processor 63 then retrieves from the look-up table 68 in memory 65 the appropriate action to be taken by the output device in response to receipt of that message, and causes the required output to be given by the output device 64.

Setting up a message generator

The initial set-up of the message generator consists in firstly configuring the trigger sensor so that it can detect the event or events in respect of which the message generator is intended to generate messages. This may be done by temporarily connecting an external input device 100 such as a portable computer to the message generator. The trigger sensor 31 may be configured to receive an output, for example from a ball detector of a roulette wheel, so that the trigger sensor is notified when the ball has entered a pocket in the roulette wheel and can generate a corresponding output. Alternatively, the trigger sensor may receive an output from a pressure sensor on a card or roulette table which detects the placing

of a gaming chip on the table, so that the trigger sensor can generate an output to indicate that a bet has been placed.

Next, data may be input from the input device 100 to the look-up table 37 in memory 33 identifying the events which the trigger sensor will detect, and the message types which are to be produced by the message generator in response to the detection of those events. The external input device 100 may also input into the memory 33 a network address such as "MG1112" or other identifying data relating to the message generator device.

Connecting a message producer to a node

The registration routine for connecting a message generator such as message generator MG1112 to a node such as T111 will now be described with reference to Figure 7.

A physical or other communication connection between the message sender 35 of the message generator 30 and the message interrogator 57 of the node 50 is established via the output port 35a of the message generator, for example by connecting a cable or by establishing a wireless or infrared connection.

When communication is established, the communication node 50 sends a message to the message generator 30 identifying the address T111 of the communication node 50, which address the processor 32 of the message generator stores in its memory 39 as its "parent address".

Message generator 30 sends a message to the node 50, identifying the address or other unique identifier MG1112 of the message generator, and requesting that the address or identifier of the message generator is added in the list of child nodes in the register 53 of the node 50. The node 50 receives this message, and the processor 51 as part of the registration routine identifies the child port at which

the message arrived, and adds the identifier MG1112 to the list of devices associated with that child port in the register 53. In the illustrated embodiment, the message generator 30 is connected to child port C1 of the table node 50, and thus in the register 53 the identifier MG 1112 is recorded against the port C1.

The message generator 30 also sends information identifying the types of message M1, M2....Mn which the message generator can produce. In this example, the message generator 30 is able to produce message type MI indicating that a bet has been placed, and message type M2 indicating that a player has taken a seat at the table. In this example, message type M1, M2 includes data identifying the message generator MG1112 as the origin of the message, and thus message type MI indicates a message from the message generator MG 1112 indicating that a bet has been placed. Alternatively, the origin information may be added to the message header independently of the message type data. The processor 51 of the node 50 checks whether each message type is already present in its look-up table 55, and if it is not then the processor 51 then adds the message type to its look-up table 55. At this stage in the registration process, the look-up table 55 includes only identifiers of message type/origin M1, M2, etc, but does not yet include any addresses to which these message types should be sent. The addition of this information to the look-up table 55 comes later in the process of setting up the network.

When all of the message types M1, M2, etc, that the message generator 30 can produce have been added to the look-up table 55, and the address of the message generator MG1112 has been added to the list of child nodes in the register 53, then the registration routine in the node 50 is complete. The registration routine of the message generator 30 is complete when all this information has been sent to the parent node 50, and the address of the parent node T111 has been recorded in the memory 33 of the message generator.

The input device 100 may also be used to store in the memory 65 an address ID 66, such as "D1111", for the message consumer to identify it to other network devices, in particular to its parent node. The appropriate connections to display devices etc are then made from the output device so that the required actions can be performed on receipt of their respective messages.

Connecting a message consumer to a node

The registration routine for connecting a message consumer such as D1111 to a node such as T111 will now be described with reference to Figure 8.

To connect the message consumer 60 (for example D1111) to a node 50 of the communications network, a physical or other communication connection between the message interrogator 62 of the message consumer 60 and the message sender 58 of the node 50 is established via the input port 61 of message consumer 60, for example by connecting a cable or by establishing a wireless or infrared connection. In this example, the message generator MG 1112 has already been connected to the child port C1 of the node 50, and thus the message consumer 60 is connected to the child port C2 of the node 50.

When communication is established, the communication node 50 sends a message to the message consumer 60 identifying the address of the communication node 50 (for example T111), which the processor 63 of the message consumer stores in its memory 65 as its "parent address" 69.

The message consumer 60 sends its address identification 66 (in this case D1111) to the node 50, and requests to be added to the list of child nodes in the register 53 of the node 50. Processor 51 of the node 50 then identifies which child port the device is connected to (in this case C2), adds the address ID to the list of devices associated with that child port in the register 53 of its memory 52. In this example, message generator MG 1112 is already present in the list, associated

with child port C1. The message consumer D1111 is entered in the register 53 associated with child port C2.

In the look-up table 68 of the message consumer 60, the message types for which an output from the message consumer 60 is required are listed. In the registration routine, each of these message types is sent to the node 50. In the example seen in Figure 6, the message consumer 60 has been set up so that in response to a message type M1 from an origin address MG111, the consumer performs an output identified as "Output 1". Likewise, in response to a message type M2 from an origin address MG1122, the consumer performs an output identified as "Output 2".

When the message type information is received at node 50, the processor 51 firstly determines whether or not that message type (i.e. that combination of message type and origin) is already present in its look-up table 55. If the message type is already present in the look-up table 55, then the address ID of the message consumer D1111 is added to the look-up table 55 in relation to that message type.

The processor 51 then determines whether the message type has an origin ID address that matches an address stored in the register 53 of the node 50 and associated with one of the child ports C1, C2 of the node 50, i.e. determines whether a message required to be sent to the message consumer 60 in fact originates at another child node of the communications node 50.

If the message type originates from one of the child nodes of the node 50, such as MG1112, then the message type will already be present in the look-up table 55 (having been stored there during the registration process for the message generator) and the registration routine will only require the address of the message consumer D1111 to be added to the look-up table 55 in relation to that message type.

In the example of Figure 8, the message consumer D1111 requires to receive messages of type M2 from the message generator MG 1122. Since message type M2 is already present in the look-up table 55 of table node 50 following the registration process of message generator MG 1122 described in relation to Figure 7, the address of the message consumer D1111 is added in the look-up table in association with message type M2.

If the message type such as M1, M3 or M4 is not already present in the look-up table 55 indicating that this message type does not originate with a child node of this parent node 50, the processor 51 makes a new entry in the look-up table 55, recording the message type (for example M1 originating from MG111), and the address D1111 of the message consumer 60 to which it should be sent. Additionally, the processor 51 of the node 50 sends a message to its parent node (in this case node S11, as seen in Figure 2) requesting that the message type M1 originating from MG111 with the destination address T111 of node 50 be added to the look-up table 55 in the parent node S11. Likewise, message types M3 originating from MG2211 and M4 originating from M1212 are added to the look-up table 55 in the parent node S11, each associated with the address T111 of node 50.

The registration process continues at the parent node S11, where the processor determines whether or not each message type/origin received from node T111 is already present in the look-up table of the node S11. If it is not, then the processor adds the message type/origin to the lookup table 55 of node S11, in association with the address T111 of the node 50. The processor of the parent node S11 then determines whether the origin address of this message type is associated with one of its child ports, and if it is not the node S11 sends a message to its parent node (in this case node A1, as seen in Figure 2) requesting that the message type/origin be added to the look-up table 55 of the parent node A1, in association with the address S11.

message destination from its LUT 55 to site node S11. For each combination, site node S11 first determines whether the message type is already in its LUT 55.

If the message type is not already in the LUT 55 of site node S11, then the message type is added.

If the message type is already in the LUT of site node S11, then the site node determines from the origin address information associated with that message type whether the message type will reach site node S11 from an address on the list of child nodes of node S11, i.e. checks whether the origin address is "local" to the node S11. If the origin address is local, then the site node adds the destination address to its LUT in relation to this message type.

If the site node determines from the origin address information that this message type will not arrive from a child node of the site node S11, the site node sends a message to its parent node (the area node A1) requesting that this message type be added to the LUT of the parent node A1, with the destination address set as site node S11.

The processing moves on to the next message type found in the LUT of table node T1.

Figure 10 is a flowchart illustrating the process steps taken at a parent node N when connecting a child node N-1 to the parent node N.

At step 101, the parent node receives an identifier from the child node, and at step 102 the identifier is added to the list of child nodes held in the register 53 of the parent node N.

The parent node N then sends its identifier to the child node N-1, and the child node records the identifier as the address of its parent node in its register 53.

The parent node then interrogates the child node at step 112 to determine whether any other message types are stored in its LUT, and if so requests the child node to send the next one whereupon the processing resumes at step 104.

When the child node has notified the parent node of all message types stored in the LUT of the child node, the processing ends.

Each time a child node has an entry added to its LUT, or has an existing entry in its LUT altered, the relevant entry is notified to its parent node which then carries out processing steps 104 to 112 of Figure 10.

With reference to Figure 11, an exemplary part of a network will now be described. In the network shown in Figure 12, a site node S1 has two table nodes T1 and T2 connected to it as "children". Table node T1 has connected to it a first message generator MG1 and a first display unit D1. Table node T2 has connected to it a second message generator MG2.

The first message generator MG1 produces a message type M1.1 in response to a first event E1, and produces a message type M1.2 in response to a second event E2.

The second message generator MG2 produces a message type M2.1 in response to a third event E3, and produces a message type M2.2 in response to a fourth event E4.

Display device D1 is required to produce an output action A1 in response to a message of type M1.1 from message generator MG1, and to produce an output action A2 in response to a message of type M2.2 from message generator MG2.

When the first message generator MG1 is connected to child port C1 of table node T1, the message generator MG1 stores its parent address T1 and requests table

node T1 to add MG1 to its list of child nodes. MG1 then sends its message type information to T1, and T1 adds to its lookup table the message types M1.1 and M1.2, at this stage without any corresponding "send to" addresses.

Display device D1 is then connected to child port C2 of table node T1, and T1 sends its address to D1 for storing as its parent address. D1 sends its address identification to T1 and requests to be added to the list of child nodes of T1. D1 then sends to T1 the message type information M1.1 and M2.2 identifying the messages which are to be sent to D1 for action.

When D1 sends message type M1.1, T1 determines that message type M1.1 is already present in the LUT of T1, T1 determines from the origin address information relating to message type M1.1 that this message type will be received at T1 from its child node MG1, and that the destination address D1 is also on its list of child nodes, and simply adds the address D1 in relation to message type M1.1 in its LUT. T1 may also send a message to child node MG1 indicating that an interest in message type M1.1 has been registered (because a "send to" address D1 is recorded in the LUT in relation to message type M1.1). Message generator MG1 may be configured so that when event E1 is detected, message M1.1 is only sent if message generator MG1 has been notified of the registration of an interest in message type M1.1, i.e the message is only sent if T1 has notified MG1 that there is a "send to" address (D1) recorded in the LUT of T1 in relation to message type M1.1.

D1 then sends message type M2.2 to T1. This message type is not already present in the LUT of T1, so T1 then adds the message type M2.2 and a corresponding destination address D1 to its LUT. T1 then determines from the origin address information relating to message type M2.2 whether this message type will be received at T1 from a child of T1. Finding that it will not, T1 sends a message to S1 requesting that message type M2.2 be added to the LUT of S1, with the corresponding "send to" destination address T1.

In operation, occurrence of event E4 causes message generator MG2 to send a message of the message type M2.2 to its table node T2. Table node T2 receives the message, retrieves the address S1 from its LUT and forwards the message on to site node S1 because the LUT of T2 has S1 as the destination address for message type M2.2.

The message arrives at site node S1, which redirects it to table node T1 on the basis of the "send to" destination address held in the LUT of S1. Likewise, when the message is received at table node T1, the message is forwarded to display device D1 on the basis of the "send to" destination address D1 held in the LUT of T1 for message type M2.2. Receiving message type M2.2, display device D1 performs the action A2 which may be to display an indicator lamp or play a video or make another output.

Likewise, when event E1 is detected by message generator MG1, then message type M1.1 is sent to table node T1, whence it is redirected to display device D1 on the basis of the "send to" destination address D1 held in the LUT of T1 for message type M1.1. Display device D1 performs action A1 in response to receipt of the message type M1.1.

Messages requiring responses

The above description relates to messages which are sent out from message generators, but do not require responses from their recipients. The invention foresees that the communications network may also deal with messages which require the terminal receiving the message to send a response back to the originator of the message.

In the case of such messages, the message header includes data indicating that a response is required, in addition to data indicating the origin of the message, and the nature of the response required. With reference to Figure 11, an example will

now be described in which a message of type M2.2 is sent from message generator MG2 to message consumer D1, requiring a response to be sent from message consumer D1 back to message generator MG2.

Message generator MG2 assembles the required message, placing in the message header the source address MG2, a "response required" flag, and message type information M2.2. The message is then sent via the parent port P of message generator MG2, and arrives at the child port C1 of node T2.

The processor of T2 retrieves from the message header the address and information relating to the source of the message, and determines that a reply is required by detecting the "response required" flag. The port C1 on which the message arrived is also identified. The node T2 then adds the identification of the source of the message (MG2) to the entry relating to port C1 list in its registry 53 which maintains a relationship between the ports of the node T2 and devices associated with each of those ports. The list of devices and respective message arrival ports is updated, irrespective of whether the arrival port is a child port or the parent port.

The message is then forwarded in accordance with the "send to" address for that message type recorded in the look-up table of the node T2, which in this case sends the message from parent port P of node T2 to the C2 child port of node S1.

The processor of S1 retrieves from the message header the address and information relating to the source of the message, and determines that a reply is required by detecting the "response required" flag. The port C2 on which the message arrived is also identified. The node S1 then adds the identification of the source of the message MG2 against the port C2 on the list in its registry 53 which maintains a relationship between the ports of the node S1 and devices associated with each of those ports. The node S1 then forwards the message via its port C1 to node T1, in accordance with the "send to" address recorded in its look-up table.

player, and as a message consumer. For example, each player position may have a display screen which shows information relating to the particular player, such as his remaining account balance, the wager he has currently placed, the wagers of other players at the table, and optionally other information such as the results of the last ten spins of the roulette wheel.

The roulette wheel 121, the player positions 123 and the display 122 are all linked to the table communications node 124, so that gaming software, for example resident at the table computer 125 linked to the table node 124, can control operation of the roulette wheel 121 and resolve wagers placed by the players when each spin of the wheel results in the ball settling in a pocket. The table computer at 125 may be linked to more than one table node 124, and thus may control more than one roulette table 120.

When setting up the communication system, the display 122 is set up as a message consumer which receives messages from the player positions 123a, 123b etc in order to display each player's wager for the current game. Table node 124 has a look-up table in which message types of originating from the player positions and including wager information are listed, with the display 122 as their destination address. The display 122, in response to receiving those messages displays images corresponding to the wagers placed by the players. The player's wager messages may also be sent to the table computer 125, by adding the network address of the table computer 125 to the LUT in the table node 124 in relation to these messages.

The roulette wheel 121 is configured as a message generator, and has a ball sensor for generating messages signifying that the ball has settled in a particular pocket of the roulette wheel. These messages may be routed through the table node 124 to the table computer 125, where they are compared with wager information received from the player positions in order to determine how each player's wager should be resolved.

It will be understood that the wager messages from player position 123d will already have an entry or entries in the LUT of table node 124A, so that the wager messages can be sent to the table computer 125. The table node 124A simply adds a further "send to" address in the LUT for each of the required messages so that they are additionally forwarded to the casino node S.

Likewise, if a player in a first casino wishes to see the wagers being placed by an acquaintance in another casino, he may make an entry in the LUT of his player terminal 123a, which will alter the LUT in the table node 124 and in the casino node S. The casino node S at the first casino will then send a message to the next node up in the hierarchy, the area node, which will then add these message types to its LUT with a destination address of the casino node S. The area node will then identify the source of the messages required and determine whether the messages will reach the area node from a child of the area node (i.e. from another casino within the area). If it is determined that they will, the area node notifies that child node (i.e. the other casino) to direct messages of this type to the area node. The casino node at the other casino determines which table node it expects the messages to arrive from and notifies that table node to add the casino node to its LUT in respect of these messages. The table node will already have an entry in its LUT in respect of these messages from the player position and thus simply adds the casino node address as a further destination address in its LUT.

When these operations are completed the relevant messages from the player position at the other casino will be sent to its table node, forwarded to the casino node of the other casino, forwarded to the area node and then returned to the casino node at the first casino and then to the table node of the interested player and eventually to the interested player's position.

It will thus be understood that by connecting a message consumer device to any node in the system and specifying the type and source of messages which that consumer device shall receive, the necessary modifications to the LUTs of

communications nodes are made, firstly moving up the hierarchical strata until a node is reached which is common to both the message consumer and the message origin, and then moving down from this common node to the message origin node or device. In each case, the message passes up the hierarchy from the message generator only as far as it needs to in order to be able to come back down to the relevant message consumer.