(54) **Title:** METHOD AND APPARATUS FOR NODE REALIGNMENT IN A TELECOMMUNICATIONS NETWORK

Figure 1
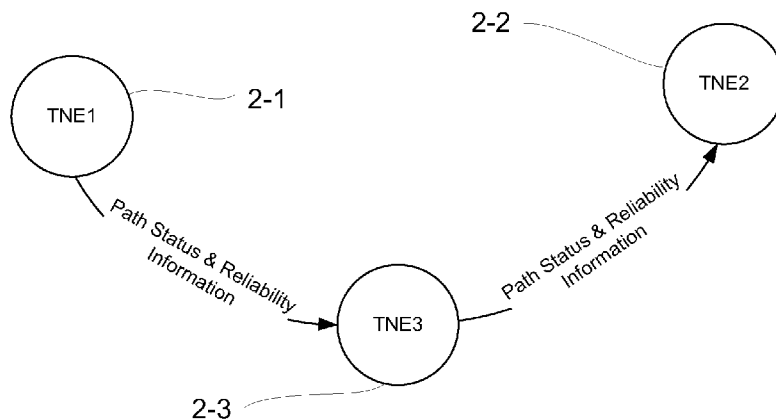
(57) **Abstract**: A method of realigning a node in a label switched network comprising a plurality of nodes and a node with processor, memory and a network interface, for carrying out the method. The method includes periodically maintaining backup path status information for the node restarting the node and re-establishing label switched paths with the other nodes using the backup status information. Communication with adjacent nodes is carried out in order to reconcile the path status information with respective path status information in the adjacent nodes in order to establish node realignment preferably judged against a threshold value for path reliability.

METHOD AND APPARATUS FOR NODE REALIGNMENT IN A TELECOMMUNICATIONS NETWORK

<u>Technical Field</u>

5

This invention relates to a method of realigning a node in a label switched network and to a node for use in a label switched network.

<u>Background</u>

10

A telecommunications network is typically made up of a plurality of nodes which are interconnected. Routes are created through the network by choosing a set of links between the nodes so that a route can enter the network at an ingress node, traverse the network by hopping between nodes via the links and exit the network at an egress node. Typically each node has knowledge of the routes traversing that node.

15

Periodically, it is necessary for nodes to be restarted. Prior to the restart, it is likely that one or more route will traverse the node and thus after a restart it is necessary for the node to recover the state information about routes traversing the node in order to successfully allow the one or more routes to be resurrected after the restart. This recovery of state is typically termed "realignment" of the node.

20

In the prior art, two approaches typically have been used to realign nodes following a restart. In the first approach, nodes carry out periodic backups of their state to some form of non-volatile memory and are then able to restore that state following a restart. In a traditional circuit-based domain this approach is called "hard state". In such a network (typically a time division multiplex (TDM) or dense wavelength division multiplex (DWDM) network) the circuit paths are generally quite static and do not vary over short periods of time. Thus periodic snapshots taken in the form of backups, generally work quite well because the routes in the network are unlikely to have changed in any significant way between backups. However, if significant changes to the network have been made, a very large manual effort is required to reconcile the local circuit status with the actual network status following a restart.

25

30

2

In other types of networks such as packet switched networks, for example Internet protocol (IP) or multi-protocol label switched (MPLS) networks, the routes change much more quickly than in traditional circuit-based networks. Accordingly a second approach to node restart has been made for this type of network, which involves a restarted node talking to neighbouring nodes in the network, i.e. nodes which are interconnected with the restarted node to gain state information about the routes in place before the restart. In the context of such networks this type of restart is known as a "graceful restart" and is described in Internet Request for Comments (RFC) 5063. This prior art graceful restart procedure assumes that only one node restarts at a time. However, if multiple restarts occur, the information available from neighbouring nodes as postulated in RFC 5063, is likely to be incomplete and the resurrection of previous routes will fail. Furthermore, if the network has fast changing routes, a backup and restore method will also fail because route changes since the last backup will often be too great and thus the restarted node will have an unacceptably outdated restarted state.

Thus there is a need for recovery of node state information to be possible in the event of multiple node restarts in a network having relatively rapid changes in routing.

It is an object of the present invention to overcome at least some of the problems of the prior art node restart approaches described above.

Summary

The present invention provides a method of realigning a node in a label switched network

Typically the network will have a plurality of nodes and the method includes periodically maintaining backup path status information for the node. In response to restarting the node, the label switched paths are re-established with the other nodes in the network using the backup status information. This is achieved by communicating with an adjacent node in order to establish a path reliability value for each path recorded in the backup status information using a reliability value from an adjacent node, in order to establish node realignment.

Advantageously, during restart, the node maintains a path reliability value for a path traversing the node. These path reliability values may be exchanged with adjacent nodes for a path traversing the adjacent nodes to establish a cumulative path reliability value for the path. In this way, the probability of the path being a valid path may be increased since the existence of the path is based upon multiple stored backups in the network rather than just the backup status information of the single node being restarted.

Preferably the cumulative path reliability value is compared with a path reliability threshold to determine whether the related path is reliable and thus may be considered to be a path which was in existence before the node restart. Advantageously the path reliability threshold is a function of the number of Hops in the related path.

The invention also provides a node for use in a label switched network comprising a plurality of nodes. The node includes a processor which is operable periodically to create a backup status record for the nodes. The node also includes a memory arranged to store the backup status record and a processor arranged to restart the node on command and the restore the node to a state defined by the backup status record. The processor is also arranged to calculate a path reliability value. The node has a network interface which is operable to receive path status information and path reliability values from adjacent nodes.

Following a restart, the node may receive a path reliability value from an adjacent node and the processor may modify the received path reliability value dependent on an internally held path reliability value for the same path. This modified path reliability value may then be passed to another adjacent node. The processor may be arranged to modify the path reliability value to establish a cumulative path reliability value for the path. The processor may further be arranged to compare the cumulative path reliability value with a path reliability threshold to determine whether the related path is reliable and thus may be properly set up following the restart. Preferably the processor is arranged to calculate the path reliability threshold as a function of the number of Hops in the related path.

The invention also provides a computer program product which, when executed on a computer, causes the computer to carry out the method of the invention.

Brief Description of the Drawings

4

Embodiments of the invention will now be described by way of example with reference to the drawings, in which:

5   Figure 1 is a schematic block diagram of several nodes in a network;

Figure 2 is a schematic block diagram of a node;

Figure 3 is a schematic diagram showing the process for establishing a route after node
10   restart;

Figure 4 is a schematic diagram of the process for deleting a route after node restart; and

Figure 5 is a flow chart of a restart procedure.
15

Detailed Description

The invention is described below in connection with generalised multi-protocol label
switching (GMPLS) but the technique is applicable to any network having multiple nodes
20   and routes set up through those nodes and in particular, to label-switched nodes..

Typically in such a network, a node or "network element" (NE) may be restarted due to a
software problem or normal maintenance. The restart of the control unit in the NE causes
re-initialisation of the control plane meaning that the control plane information about the
25   label switched paths (LSPs) in the Network Element (NE) may be lost. Typically the data
plane is unaffected.

Briefly, the NE stores periodic backup files the of path status, restores itself according to
this backup file and then communicates with adjacent nodes to test the validity and
30   currency of the backup file. This is described in more detail below.

In general terms, a particular LSP is considered to be valid when a significant number of
the NEs traversed by that LSP also have the same information relating to that LSP. Hence,
the more NEs that have recorded the existence of an LSP immediately before the NE

5

restart, the more likely it is that the LSP was indeed current before the restart and had not been removed.

Thus with reference to Figure 1, NE1, NE2 and NE3, 2-1, 2-2 and 2-3 are located in a GMPLS network and are able to communicate path status and reliability information between themselves. With reference also to Figure 2, each NE 2 has a processor 4, a memory 6 and a network interface 10. The processor is able to cooperate with the memory 6 to create a backup status record 8 and to communicate with the other NEs via the network interface 10.

During a restart, link management protocol (LMP) information is used in order to determine if an neighbouring node which is typically a node a single Hop away, is UP or DOWN. When the first control channel with an adjacent NE is up then the adjacency with that NE is considered to be UP and when the last control channel of an adjacent NE goes down, the adjacency with that NE is considered to be DOWN and it is likely that the neighbouring, or adjacent, NE has restarted.

In order to realign a neighbour, different Notify messages are used as explained below in connection with Figures 3 and 4.

The realignment procedure uses three new Notify messages:-

Notify_Add; **Notify_Upgrade** and **Notify_Delete**.

The new realignment procedure also uses the concept of an LSP **Degree of Reliability** which is explained in detail below.

<u>LSP Degree of Reliability</u>

An adjacent NE i.e. any NE which is traversed by a particular LSP, can be in two different states; namely **Aligned** or **Restarted**. Also information about an LSP can be **Reliable** or **Not_Reliable** and with a certain degree of reliability.

6

An adjacency is considered to be **Aligned** when if viewed from a particular NE, the adjacent NE has information about LSPs which is entirely consistent between the NEs.

When an adjacent NE is considered to be in the **Restarted** state, it will instead hold

5   information about LSPs which do not have an adequate or consistent degree of reliability compared with an adjacent NE. In the preferred embodiment, the degree of reliability for an LSP which is **Reliable** is zero whilst a **Not_Reliable** LSP can have a degree of reliability assuming any value between 1 and 2^8.

10   When the two sides of an adjacency, i.e. two NEs linked along an LSP, exchange the same LSP information and the state of the LSP has a degree of reliability indicated as reliable, then the state of that adjacency moves to **Aligned.**

In order to show how the procedure may be implemented, we define the following

15   parameters:

o   LSP degree of reliability (**DR**);

o   Received DR (**R_DR**): is the DR that a NE has received at its interface 10;

o   Propagated DR (**P_DR**): is the DR that a NE has propagated at its interface 10;

o   #_NE_Unaware: is the number of the NE belonging to the same LSP that is not able

20   to rebuild the LSP using its backup status record 8;

o   In order to decide when LSP information becomes **Reliable**, a threshold (**Th**) is also defined. The threshold preferably depends on the number of hops in the LSP, i.e. the number of links between NEs it must traverse, and on a configurable value.

To have an adjacency in the Aligned state all its LSP must have DR = 0.

25

At restart, the NE 2 rebuilds its database retrieving LSP's information from .rpp files from its persistent storage 6 or restored from a backup 8. Because this information is not reliable (it may be out-of-date) the degree of reliability for each LSP is assigned the value DR = 1.

30   When the DR reaches a threshold (Th) value the reliability for an LSP is moved to 0 (Reliable).

Assuming an LSP traverse #_Hops, the rule is

If DR >= Th *then* DR == 0

7

Where Th = #_Hops/X

X is a configurable value defined on network base.

As explained below, since the DR value gradually increments as the P_DR values propagate through a network, the division by the number of hops helps normalise the threshold value for different sized network. The value X, allows adjustment of the weighting given to the threshold value, Th; a lower value allowing an LSP to be considered Reliable with a larger number of NEs having information about the LSP in their backup data. The relevance of this will become more apparent later.

This procedure is repeated for each LSP in the rebuilt database. Every time there is an LSP "unknown" by an adjacent NE its DR, P_DR and R_DR are considered 'null'. Null values are considered less then 1 but higher then 0; so somewhere between Not_Reliable and Reliable.

Thus is pseudo-code we have:-

Procedure

NE Restarted (start):

o   When a NE restarts, it tries to recover information about all LSPs from .rrp. and sets DR for each LSP to 1 (Not_Reliable).

o   After this operation the NE tries to bring up the control channels with its neighbours.

o   For each neighbour with a running control channel the restarted node produces and sends a Notify_Update message with the indication that it is restarted and the list of LSPs it has been able to rebuild with the indication of their DR (DR=1).

o   P_DR is set to 1.

Each Aligned neighbouring NE (i.e. neighbours that haven't been re-started so have a completely reliable LSP database) receives a Notify_Update from the restarted NE

o   With respect to the adjacency from where the message is received the NE (Aligned) performs the following checks:

8

- LSP not present locally (DR=null) but reported into the Nofity_Update message (R_DR>0): Notify_Delete message is sent to the restarted node for this LSP because NE(Restarted) has an old LSP in its backup.

- LSP present locally (DR=0) but not reported in the Notify_Update message received

5   (R_DR=null): Notify_Add message is sent for this LSP with full info needed to rebuild the LSP because NE(Restarted) is missing a valid LSP in its backup.

- LSP present locally (DR=0) and also in the Notify_Update message (R_DR>0): Notify_Update message sent with DR=0 (Reliable) for this LSP i.e. NE Restarted has consistent information with NE(Aligned) for this LSP.

10   o   When all the LSPs locally present in the NE (Aligned) and reported into the Notify_Update message from its neighbour, have been checked, an indication of "ending procedure" is added to the last Notify message sent.

The NE (Restarted) must process the information received from its Aligned neighbouring

15   NEs.

Thus:-

NE Restarted (realigning):

   o   All its LSPs are Not_Reliable and have DR >0 (P_DR as well. DR is considered null

20   for LSPs that it is not aware).

   o   The NE Restarted starts to receive Notify message from its neighbours, and manages these messages in the following way:

- Notify_Add received: create the LSP reported (DR=0)

- Notify_Delete received: remove the LSP reported (DR=null)

25 -   Notify_Update received: set DR in the following way:

- *If* the information about the an LSP with [(R_DR > P_DR) *or* (P_DR is null *and* DR is null)] *then*:

- *If* the NE is Egress/Ingress for the LSP *then*:

- DR == R_DR

30 -   *If* the NE is Transit for the LSP *then*:

- DR = R_DR

- a Notify_Update message containing information about the LSP with P_DR = DR is propagated Downstream/Upstream

- P_DR == DR

9

- ▪ *If* (P_DR is null *and* DR is null) *then* : #_NE_Unaware ++

- ▪ *If* the information about the LSP with (P_DR is null *and* DR =1) *then*

- ▪ *If* the NE is Egress/Ingress for the LSP *then*:

- ▪ DR == R_DR + DR

5   ▪ a Notify_Update message containing information about the LSP with P_DR = DR is propagated Downstream/Upstream

- ▪ *If* the NE is Transit for the LSP *then*:

- ▪ DR = R_DR + DR

- ▪ a Notify_Update message containing information about the LSP with P_DR = DR is

10 propagated on both Downstream and Upstream

- ▪ P_DR == DR

- ▪ *When* DR >= Th *then*

- ▪ *If* the LSP traverses the Upstream NE *then* includes in the Notify_Add message the FullInfo about the LSP with DR = 0

15 ▪ *If* the LSP traverses the Downstream NE *then* includes in the Notify_Add message the FullInfo about the LSP with DR = 0

The state of the LSPs is the same for both the Upstream and Downstream side of the adjacency while the state of the adjacency depends on the state of all circuits it shares with

20 the neighbour.

This is explained in more detail with reference to Figures 3 and 4 which show examples of the process in action.

25 Firstly, with reference to Figure 3, all the NEs 2'-2 to 2'-7, have restarted and thus there are no Aligned nodes on which to rely. NEs 2'-2, 2'-4 and 2'-7 recover via .rpp, information about circuit **a**.

The DR values 50 are shown being incremented as messages pass between the NEs in the direction of the arrows. The R_DR values are labelled 52 and the P_DR values are

30 labelled 54. In this case, Th is set at 3.

It will be seen that at restart, NE 2'-7 only has DR=1 for route a. But NE 2'-2 and NE 2'-4 also have route a in their backup information and thus the cumulative DR increases as it propagates through these NEs.

When P_DR reaches NE7, 2'-7 the DR for route a at NE7 is finally incremented up to the Th threshold and thus this NE is able to issue a Notify_Add message with full setup information and thus allow all the other NEs in the LSP, to set the LSP as reliable (DR=0)

5   and set the route up.

In the discussion above, it has been assumed that DR for an LSP reaches the Th threshold. However, it is necessary to deal with the case in which the Th threshold is not reached for an LSP.

10   With reference to Figure 4, an example of the case of all the NEs of an LSP span being restarted but Th not being reached in which case a decision must be made whether to keep or delete an LSP.

The rule used by the NE in order to decide whether to keep or delete the LSP is the following:

15   If (LSP_Length - #_NE_Unaware) + DR < Th then the LSP can be removed

Where:

-       LSP_Length is the length of the LSP in hops, known a priori

-       #_NE_Unaware: is the number of the NEs belonging to the same LSP that are not able to rebuild the LSP thanks to the .rpp. When a NE restarted receives a Notify_Update

20   about an LSP unknown, the #_NE_Unaware is incremented by one;

-       DR is the degree of reliability of the LSP

-       Th is the threshold

All the NEs have restarted; NE1, 2"-1 and NE6 2"-6 recovers via .rpp, information about circuit a.

25   The first number on each arrow between NEs is the DR while the second one is the #_NE_Unaware

The boxes 56 represent the cumulative #_NE_Unaware.

The Th is again 3.

When the Notify_Update with DR = 1 and #_NE_Aware reach the NE 6 2"-6 we have:

-     LSP_Length: 6

-     #_NE_Aware: 5

5   -   DR: 1

-     Th: 3

So: (6 - 5) + 1 < 3 and thus it is possible to delete the LSP.

The realignment procedure is closed when all the LSPs a NE shares with a neighbour are

10   in the reliable state, that is, are all with DR = 0.

Figure 5 summarises the process in Flow-chart form.

The process starts, step 100 and the NE creates a backup, step 102. When the NE

15   restarts, step 104, the NE maintains a path reliability value for each path in the backup,
step 105 and begins to re-establish paths, step 106, by exchanging path reliability values
with adjacent nodes and deriving a cumulative reliability value for each path, step 108. The
cumulative path reliability value is compared with a threshold, step 110 and a decision
made whether to create or delete the path, step 112. When all the paths are deemed in a

20   reliable state, the process stops, step 114.

12

Claims

1.   A method of realigning a node in a label switched network comprising a plurality of nodes, the method comprising:

5   periodically maintaining backup path status information for the node;

in response to restarting of the node, re-establishing label switched paths with the other nodes using the backup status information, and

communicating with an adjacent node in order to establish a path reliability value for each path recorded in the backup status information using a reliability value from an adjacent

10   node, in order to establish node realignment.

2.   A method as claimed in claim 1 including the node exchanging the path reliability value with the adjacent node for a path traversing the adjacent nodes to establish a cumulative path reliability value for the path.

15

3.   A method as claimed in claim 2, including comparing a cumulative path reliability value with a path reliability threshold to determine whether the related path is reliable.

4.   A method as claimed in claim 3, wherein the path reliability threshold is a function

20   of the number of hops in the related path.

5.   A method as claimed in clam 3 or claim 4, wherein the comparison takes account of the number of hops in the path and the number of adjacent nodes which do not have the path recorded in their backup status information.

25

6.   A node (2) for use in a label switched network comprising a plurality of nodes, the node (2) including:

a processor (4) operable periodically to create a backup status record (8) for the node,

memory (6) arranged to store the backup status record (8),

30   the processor (4) being arranged to re-start the node (2) on command and to restore the node (2) to a state defined by the backup status record (8);

the processor (4) being further arranged to calculate a path reliability value; and the node (2) further including a network interface (10) operable to receive path reliability values from an adjacent node.

13

7.   A node (2) as claimed in claim 6, wherein the network interface (10) is operable to receive a path reliability value from an adjacent node, modify the received path reliability value dependant on an internally held path reliability value for the same path, and pass the modified path reliability value to another adjacent node.

8.   A node (2) as claimed in claim 7, wherein the processor (4) is arranged to modify the path reliability value to establish a cumulative path reliability value for the path.

9.   A node (2) as claimed in claim 8, wherein the processor is arranged to compare the cumulative path reliability value with a path reliability threshold to determine whether the related path is reliable.

10.   A node (2) as claimed in claim 9, wherein the processor is arranged to calculate the path reliability threshold as a function of the number of hops in the related path.

11.   A computer program product comprising computer executable code which when executed on a computer causes the computer to control a node to perform the method of any one of claims 1 to 5.
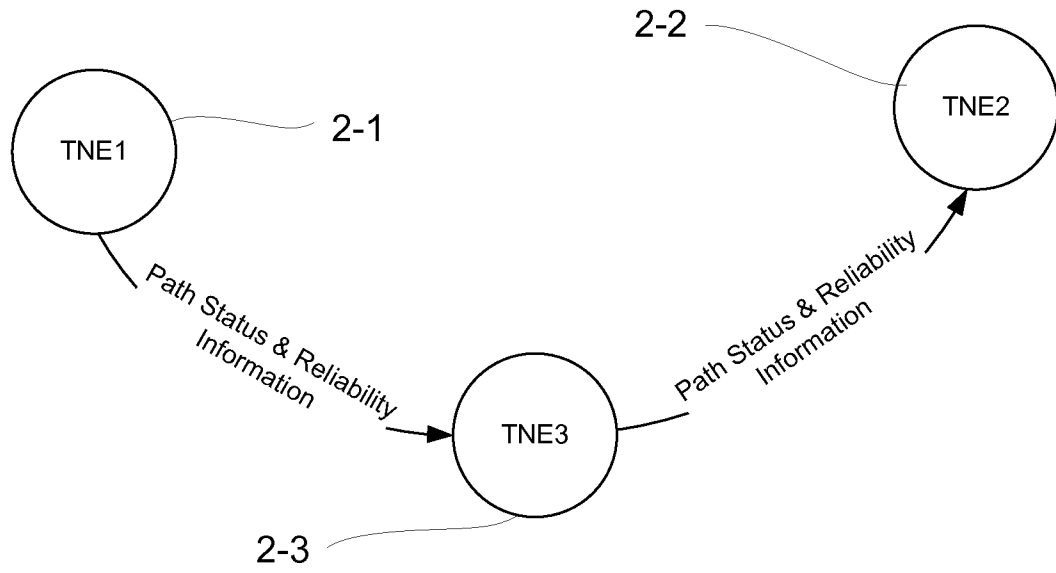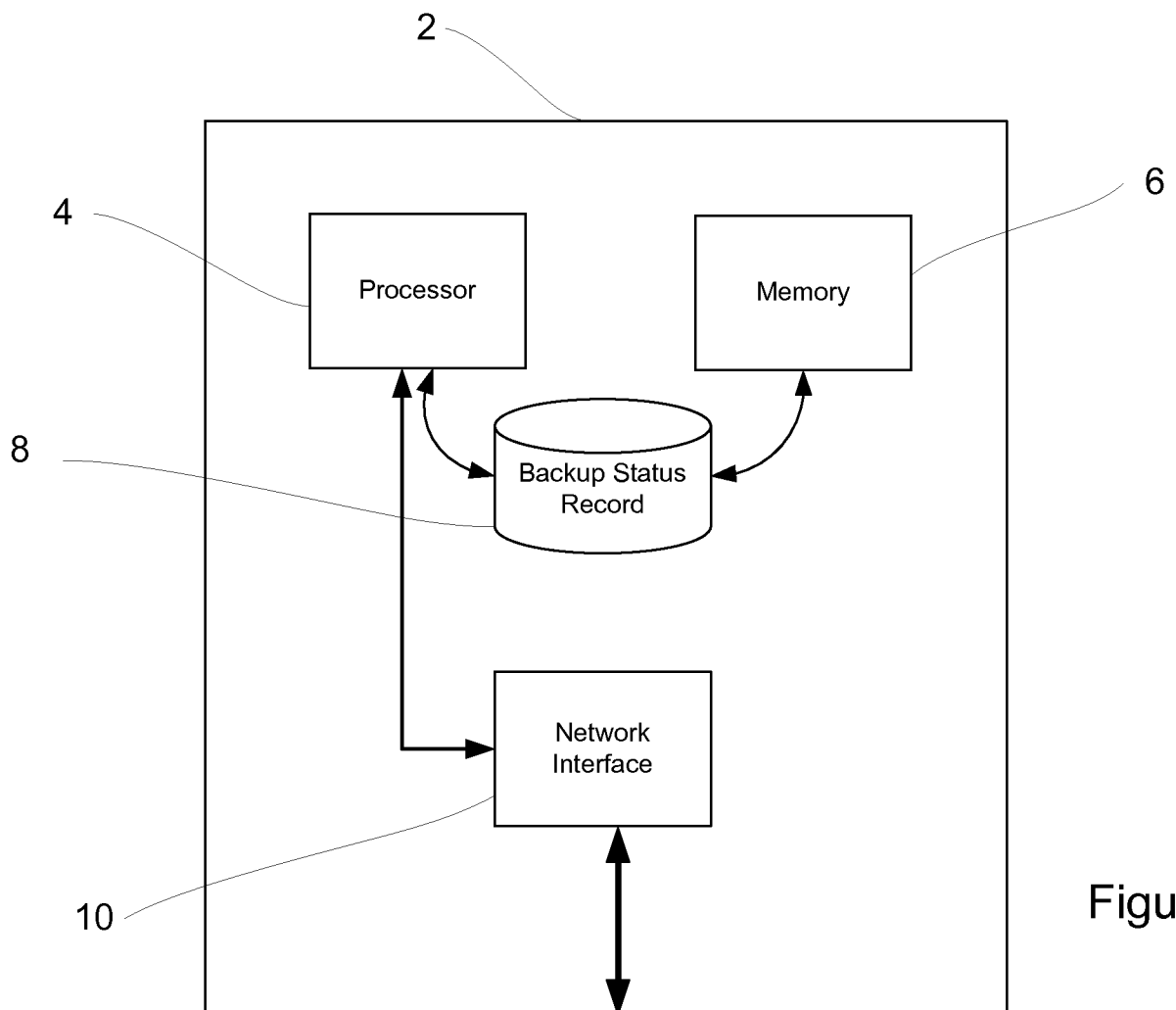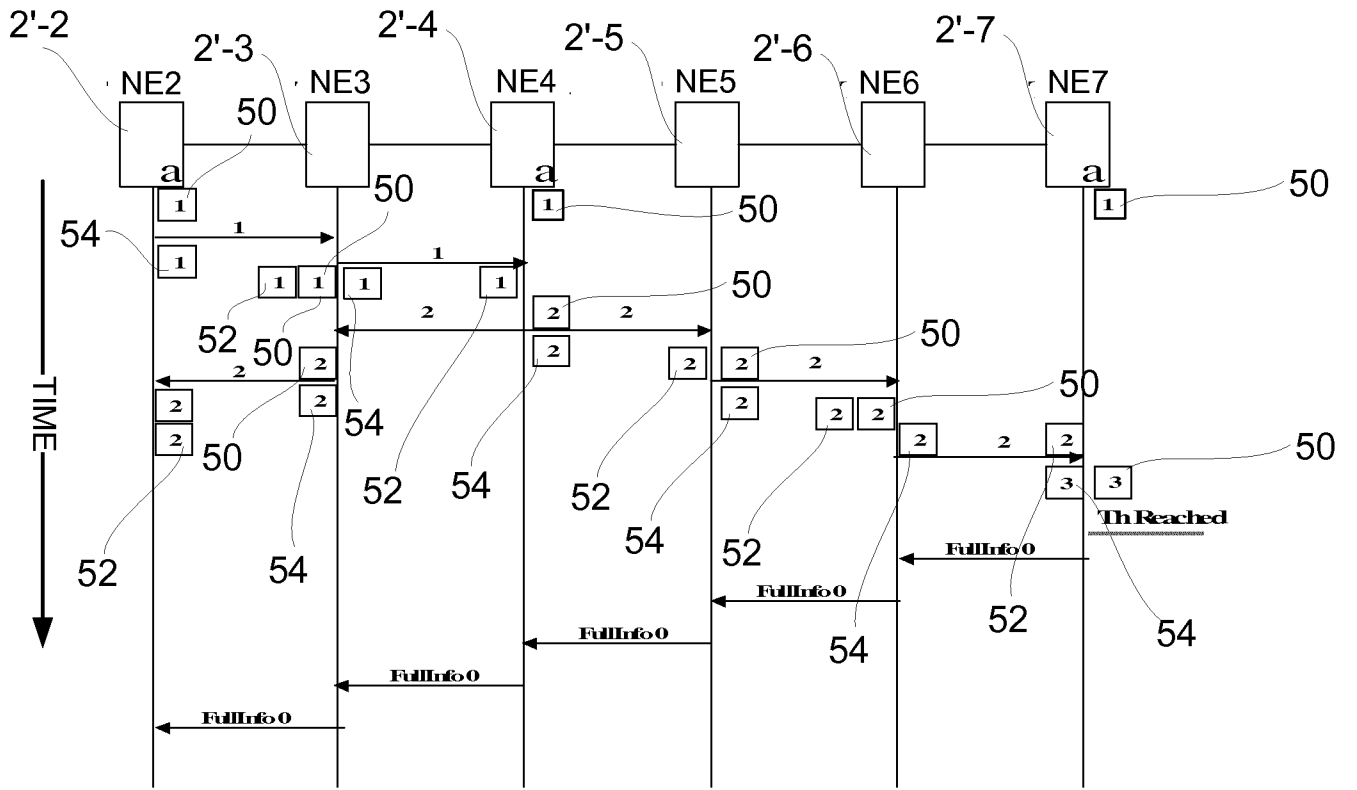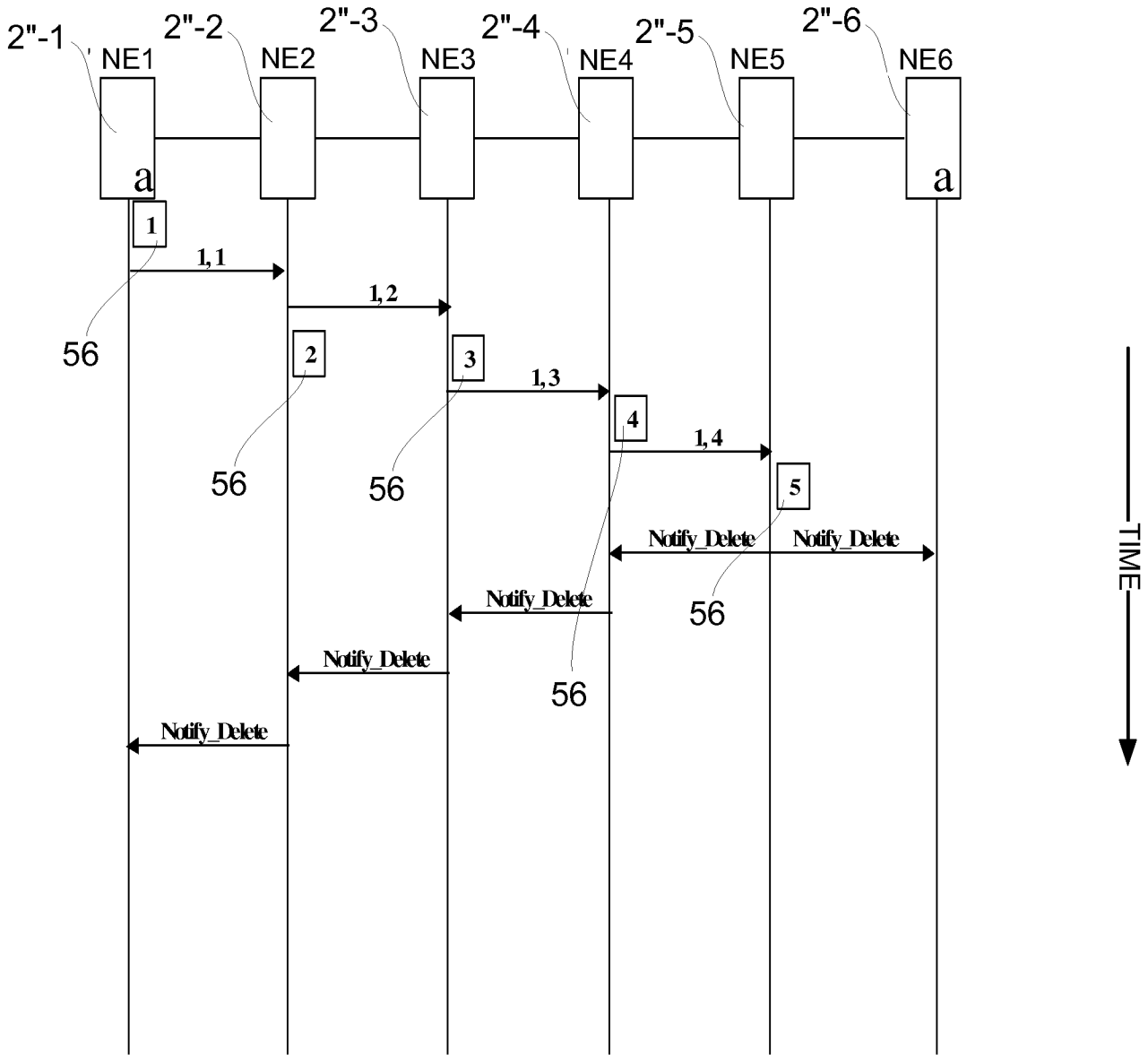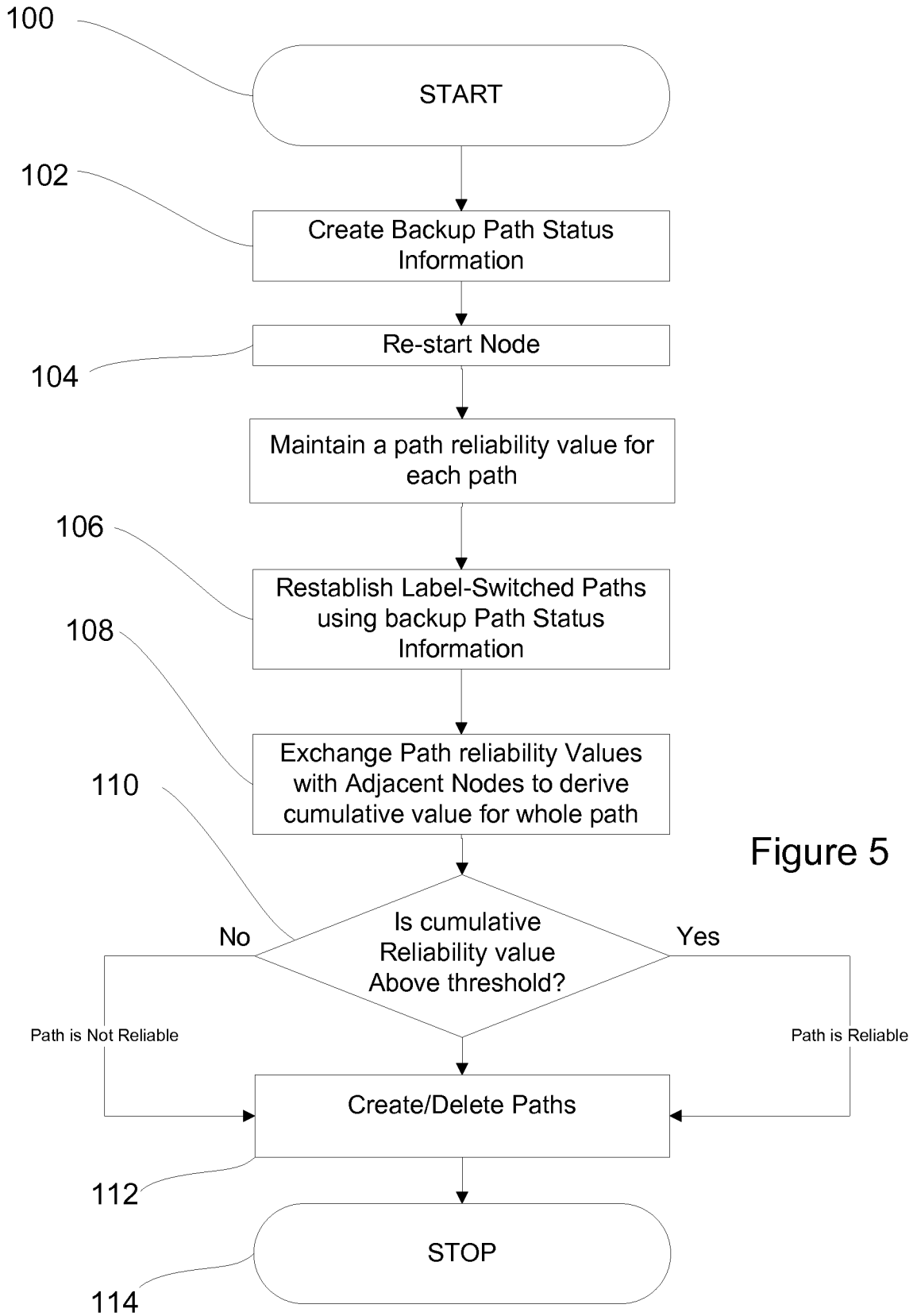
Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L12/24   H04L12/703   H04L12/723
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JING WU ET AL: "Recovery from control plane failures in the cr-ldp and o-uni signalling protocols", DESIGN OF RELIABLE COMMUNICATION NETWORKS, 2003. (DRCN 2003). PROCEEDI NGS. FOURTH INTERNATIONAL WORKSHOP ON OCT. 19-22, 2003, PISCATAWAY, NJ, USA,IEEE, 19 October 2003 (2003-10-19), pages 139-146, XP010688061, DOI: 10.1109/DRCN.2003.1275349 ISBN: 978-0-7803-8118-6	1,6,11
A	paragraph [00IV] - paragraph [000V] figure 4	2-5,7-10

-/--

[X] Further documents are listed in the continuation of Box C.   [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
23 May 2013	31/05/2013

| Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016 | Authorized officer Tous Fajardo, Juan |

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WU J ET AL: "Recovery from control plane failures in the LDP signalling protocol", OPTICAL SWITCHING AND NETWORKING, ELSEVIER, NL, vol. 2, no. 3, 1 November 2005 (2005-11-01), pages 148-162, XP027837964, ISSN: 1573-4277 [retrieved on 2005-11-01]	1,6,11
A	paragraph [0004] - paragraph [0005] figure 3 -----	2-5,7-10
A	NETWORKING WORKING GROUP RESHAD RAHMAN ET AL: "RSVP Graceful Restart Extensions; draft-rahman-rsvp-restart-extensions-00.txt", 20031001, 1 October 2003 (2003-10-01), XP015034196, ISSN: 0000-0004 page 3, line 17 - line 37 paragraph [0004] -----	1-11
A	JIANGWEILIAN(ZTE CORPORATION) FENGJUN(ZTE CORPORATION) CUIYING(ZTE CORPORATION) KONGYONG(ZTE CORPORATION) LUOJIAN(ZTE CORPORATION): "Mechanism of multiple adjacent nodes RSVP graceful restart Simultaneously; draft-jian-ccamp-multinodes-rsvp-restart-00.txt", 20051120, 20 November 2005 (2005-11-20), XP015042788, ISSN: 0000-0004 the whole document -----	1-11

1