Protection mechanism for the N2R Topological Routing Algorithm

Lopez, Jose Manuel Guterrez Lopez; Pedersen, Jens Myrup; Cuevas, Ruben; Madsen, Ole Brun

Published in:
2008 International Conference on HighPerformance Switching and Routing (HPSR 2008)

DOI (link to publication from Publisher):
10.1109/HSPR.2008.4734439

Publication date:
2008

Document Version
Early version, also known as pre-print

Protection mechanism for the N2R Topological Routing Algorithm

José Manuel Gutiérrez López*, Jens Myrup Pedersen*, Rubén Cuevas Rumín† and Ole Brun Madsen*

*Network and Security Department, Aalborg University, Denmark
Niels Jernes Vej 12 9220 Aalborg Ø
Emails: jgl@kom.aau.dk, jens@control.aau.dk, obm@control.aau.dk

†Departamento de Ingeniería Telemática Escuela Politécnica Superior Universidad Carlos III de Madrid
Email: rcuevas@it.uc3m.es

***Abstract*—The topological routing over N2R structures has previously been studied and implemented using different techniques. A first approach was achieved obtaining the best trade off between path length vs. path completion time for the shortest path between any pair of nodes. This paper introduces protection against failures by modifying the previous algorithm implementing the option of routing a packet using a second independent path. The goal is to prove that there is an easy and efficient method to route topologically a packet (in case of a failure) using an alternative path with no record at all of the original.**

***Keywords*— Topological routing, N2R, SQoS, Failure Protection**

I. Introduction

Topological routing is an alternative to traditional routing methods, based on tables. It allows for very fast restoration, and is particularly well suited for large-scale communication where table updates can be time consuming and introduces significant overheads [1]. Topological routing is understood as follows:

At a given address scheme, from any node any packet can be routed given only knowledge of the addresses of the current and the destination nodes, with no routing tables involved [1].

Related to this topological routing idea there are several studies about Small world networks (SWN) which demonstrate that with limited knowledge of the network, a greedy algorithm can construct short paths using only local information. Kleinberg's Small-world models [2] or the Watts and Strogatz Ring Model [3] are examples of the extensive literature which prove the existence of topological routing algorithms.

Topological routing has been successfully proposed for some regular structures (Grid and Honeycomb) as an alternative to the traditional table routing methods at [1], [4] and [5]. This table-free routing has also been proposed over N2R structures at [6] and [7].

The first approach concerning this issue was to implement an algorithm which could route a packet with no path information as routing tables or headers containing the complete path from any source to any destination nodes [7]. A second study proposed several algorithms to improve the results in terms of path distances and path completion time. The best solution found was named *"Balanced Algorithm (BA)"* which did not obtain the best results in path distances nor path completion time but those values had not a significant difference with the optimal [6]. The trade off between these two parameters was the best among the studied and therefore the one used to continue the work. The execution time of the *BA* is its best property. This algorithm is independent of the number of nodes of the network, the execution time at every node and any number of nodes is the always same. This method can be a good alternative for large scale networks where the routing table look up time increases with the size of the network.

N2R's are degree 3 topologies which allow to define three independent or disjoint paths. These structures, which are a subset of Petersen graphs [8], offer better properties than other degree 3 topologies such as Double Rings [9] or Degree Three Chordal Rings [10] and [11] in terms of average distances between any pair of nodes or diameters of the network. These properties allow to achieve better SQoS (Structural Quality of Service) [9] than other same degree regular topologies. Therefore, the potential of N2R structures makes them an attractive topic to focus on and this study will try to take advantage of these good structural properties combined will easy routing techniques to improve the delay on the transmissions.

The goal of this study is to find an improvement over the existing algorithm by being able to route the packets using an independent alternative path of the original. By independent or disjoint path is understood as no links nor nodes in common. When the two paths are disjoint, if an element of the shortest fails, it is guaranteed that the alternative path will be available since they will never have any element in common. This study only concerns a second independent path, but in future studies, the possibility of implementing a third path algorithm might be considered This improvement will give a protection against failures which will make the topological routing more feasible over N2R structures [12].

The topological routing protection mechanism must allow to define the alternative independent path in the same way as the original one. The packets are rerouted, in case of failure, using the current and destination nodes addresses applied to the topological properties of the N2R.

The current solutions to find the optimal second disjoint path are complex with long executions since they are based on a brute force algorithm. Therefore, to consider the topological routing as a feasible option for the future networks, the second path solution must be found in a simple and efficient way. The computational time can be minimized by reducing the complexity, but with the consequence of worse results in terms of path distances. The basic question of this study is how fast the nodes are able to make a decision to forward the packets, in case of failure, vs. the path distances obtained on the transmissions using topological routing.

The structure of the rest of the document is as follows. Section II treats the background, definitions and proper notation to understand the network structure under study and the previous algorithm. Section III introduces the modification of the previous algorithm to be able to route the packets using an alternative path. In Section IV the proposed algorithm is simulated and the results are exposed. Section V exposes the conclusions extracted from this paper. Finally, an appendix is presented an Section VI specific information is commented about problematic cases.

II. Background

The development of the topological routing over N2R structures has already some results to start with. Therefore this Section exposes the important ideas required to understand the whole concept of this paper. Subsections II-A, II-B and II-C introduce the basic properties of N2R structures, Balanced Algorithm and Protection respectively:

A. N2R Structure

N2R networks are a type of generalized Double Ring (DR) structure, where inner ring links do not interconnect physically neighbor nodes. The number of nodes in the N2R structure is any positive even integer larger than or equal to 6. These rings each contain the same number of nodes (p). Links in the outer ring and the links interconnecting the two rings can be described in the same way as the DR structure, but links in the inner ring are interconnecting node I_i and node $I_{(i+q)modp}$, where q is a positive integer. To avoid forming two separated networks in the inner ring, q must fulfill $gcd(p, q) = 1$ (Greatest Common Divisor), also q is evaluated from 1 to p/2 [13].

The addresses of the nodes are given in a certain way to make the algorithm as easy and fast as possible. The outer ring nodes addresses vary from *0* to *p-1* counterclockwise and the inner ring addresses vary from p to *2*p-1*. The relation between the outer and the inner nodes is in the way that the outer node X is connected to the inner node $X+p$.

This address system allows simple operations at the programming of the algorithm and the addresses of the neighbors are well known by every node.

Each node is connected to the neighbors with three links. In each of the cases there is a possibility of naming them as left, right and center *(L, R, C)*. Every node knows the address of the neighbors looking at the name of the link connected to. At a given address X of a outer node, to follow the link L means to reach the node $X+1$, link R means $X-1$ and link C means $X+p$. In the same way with the inner nodes, to follow the link L means to reach the node $X+q$, link R means $X-q$ and link C means $X-p$ [6]. Fig. 1 illustrates this idea.

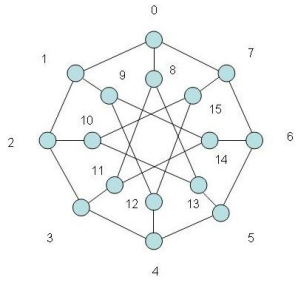


Fig. 1. N2R(8,3) notation

B. Balanced Algorithm (BA)

The *Balanced Algorithm* is named after previous studies and it was the best option found considering the trade off path distance vs. path completion time. This algorithm is executed by every node to find which link to use to forward the packets.

As a brief explanation, what the algorithm basically does is to calculate three potential distances to reach the destination with simple mathematical operations. These distances are based on the next three types of paths:

- Using the outer ring
- Using the inner ring clockwise
- Using the inner ring counterclockwise

To be able to find these values the only information that is required by the current node is the destination address. The current node address, N and q is assumed as implicit information at every node.

The shortest of these three possibilities is selected and the packet is forwarded using the link related with that option. At the next node the procedure starts all over again until the destination node is reached.

For further information and deep explanation see [6]. The solution proposed in this paper for a reliable topological routing is an extension of this algorithm. The next example illustrates the performance of the algorithm at a node. The example explains the procedure at the source node, but at the rest of the nodes is exactly the same:

The algorithm calculates the commented distances using the outer ring DT_{out} and using the inner ring DT_{in} and then the shortest one is selected to forward the packet using the corresponding link. These two variables can be split in two different terms: The number of hops of the path at any of the rings (D_{out} or D_{in}) and the number of hops to jump from one ring to the other (D_{rout} or D_{rin}). D_{out} must be in the correct format to work with (mod (p/2+1)), see [6]. Fig. 2 illustrates the terms calculated being N_S and N_D the source and destination addresses, and they must be converted to the correct format to work with as $N_S \ mod(p)$ and $N_D \ mod(p)$.

- N2R(8,3), p=8 ,q=3
- DT_{out}:

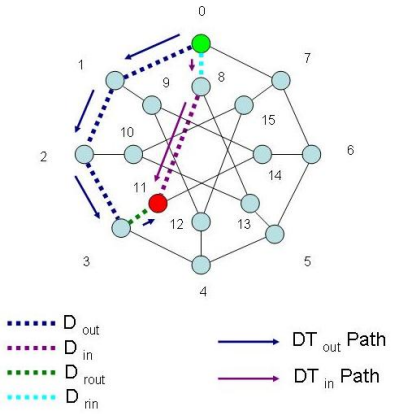


Fig. 2. Examples Node Decision

$N_S = 0$ and $N_D = 11$, $N_D\ mod(p)$=3
$D_{out} = |N_D\ mod(p) - N_S\ mod(p)| = 3 - 0 = 3$)
$D_{rout} = 1$ (Outer-Inner)
$DT_{out} = D_{out} + D_{rout} = 3 + 1 = 4$

- DT_{in}:
  $D_{in} = D_{out}/q$=3/3=1, integer. Only inner ring used.
  $D_{rin} = 1$ (Outer-Inner)
  $DT_{in} = D_{in} + D_{rin} = 1 + 1 = 2$
- $DT_{in} < DT_{out}$, therefore the link used is C (center). Start all over from the next node.

The key aspect of this algorithm is that the execution time at each node is constant and independent of the number of nodes. The table look-up time increases with the number of entries, therefore, for a large number of nodes this look up time can be unacceptably long. This algorithm solves this long delay problem for large networks.

C. Protection

An efficient routing algorithm must consider failure protection mechanisms. The option of automatically reroute packet when a failure occurs allows lower probability of loss of connectivity between any pair of nodes of the network which implies more reliable networks. The proposed algorithm addresses this automatic rerouting issue.

The proposed algorithm can be used in two different ways depending on the priority and volume of the traffic. For the regular traffic which has no high restrictions over the restoration latency in case that a failure occurs, the algorithm can be used for a *"Single path transmission (SPT)"* protection. The packets are sent just using one path (the original path in case of no failure and the alternative in case that the original is not available). This protection can be applied as well to high traffic volume since the cost of having redundancy at any moment by sending the same packet twice using two disjoint paths will consume too much network resources.

When using the *SPT* method, at a stabilized network (no failure) the *BA* routes normally the packets obtaining always at the end of communications the same path between any pair of nodes. The problem is when one element of the communication fails (node or link). Then the connectivity between that pair of nodes is lost. But if the source node is able to route the packet using a pre-configured different path than the failure one there is a protection against this kind of problem.

On the other hand, the traffic belonging to applications not tolerating any restoration latency, it is necessary to send data through two or more independent paths simultaneously, called *"Double path transmission (DPT)"* [1].

In this case the packets are duplicated and sent to the destination using both independent paths simultaneously. When this method is applied, there is no restoration latency in case of failure, allowing the prior packets to reach the destination at the required time. Having at any time information redundancy to support failures is more important than the resources spent for the protection.

The combination of these two methods can optimize the performance, response and resources of the network by selecting the best option for each kind of traffic.

III. Algorithm

In this Section the algorithm proposed is explained and the problems and difficulties found are deeply commented.

The mechanism proposed is based on the idea that the first links and last links of the original and alternative paths are different, the result is always two independent paths. The proof of this assumption will allow to define a simple protection mechanism which implies a fast execution by every node.

The difficulty of obtaining this second alternative is that there is no record at all of the original path. Hence, the procedure is to identify the mathematical properties of the N2R combined with the first path algorithm *(BA)* to obtain the guaranteed alternative independent path. This modification is based on the same distance calculation described in Subsection II-B, performing basic mathematical operations (sums and divisions) and comparisons easy to implement in any machine.

For this paper, in case of the *SPT*, it is assumed that the failure is already identified by the network and the source node knows, depending on the destination, which path to use (first or second). As further work, transition time might be studied, the time between the failure occurs and the network is stabilized (the nodes know exactly where is the problem and they can apply the second alternative if it is required).

To implement the algorithm there is a distinction between the source node and the rest of the middle nodes of the path.

Source Node: The BA algorithm is capable of calculating the distances using the different possibilities for a transmission. Therefore, at the source node it can be identified how the communication will be established just using simple mathematical operations. Then, if the kind of communication can be identified (using the outer ring, the inner or both), the last link used in the path just before reaching the destination can be also identified.

At this point the first link (this is exactly what the BA calculated before) and the last link used for the original path are identified. Hence, the proposal is to prove that an algorithm

can find an independent path just with the conditions of no coincidence of the first and the last link.

In the case of the *SPT* method and assuming that the failure was already identified, the source node knows that there is a problem with the original path for certain nodes. If the destination is one of those nodes, the second best link to forward the packet must be calculated. In the case of the *DPT* method, this operation is always realized.

To calculate that second best link to forward the packet, the algorithm calculates the distances using the other two links available to reach the destination. Then, the shortest possibility is identified and the last link of that potential path. If there is no coincidence between the last link of the original path and the alternative path, the packet will be forwarded using that option. But if there is a coincidence, the link to forward the packet will be the last available.

The information about the last link of the original path must be sent with the packet in order to use it at the middle nodes.

Middle nodes: In this case the procedure is the same for both protection methods. There are two possible links to use (the third one is the incoming link and the packet should never return to a previous node). Then, the algorithm calculates the shortest distance using the available links, and the last link for that potential path is identified as well.

If there is no coincidence between the last link identified and the last link of the original path (information implicit in the packet), the best link found is used. But if there is a coincidence the last available link is selected to forward the packet. This method is based on the idea that at some node a possible path will be found. The goal is to prove that there is a possibility of using this method and find a feasible solution at any situation. The procedure is the same at the next node until the destination is reached.

IV. Simulation

This Section treats the behavior of the algorithm and it is simulated to obtain the results for the original and the alternative paths. Subsections IV-A, IV-B and IV-C describe the simulation procedure and results:

A. Methods

The simulation was executed varying the value of p from 5 to 100 (200 nodes in total) and testing at each of these values all the possible configurations (possible q values). This simulation scenario is enough to identify the trend of the results as a function of the number of nodes in the network. Transmission from all the nodes to all the nodes are simulated at all the possible N2R configurations obtaining the distances of the original and alternative paths. These distances allow to calculate the average path distances and diameters for the analysis of the results. The execution time [1] of the algorithm at every node (time for the node to make a decision to forward the packet) is also calculated and compared with the *BA* algorithm execution time.

There is a variance on the results depending on different criteria to choose the optimal q value for each p. Hence, these q options are presented and the consequences for both paths are explained to make a decision. The three criteria are:

- **Previous studies optimal *q*,** (q_1): Values obtained as optimal for the original path using the *BA* [6].
- ***q* value optimizing the sum of the two paths diameters,** (q_2): The diameter gives the worst case possible in terms of distance. Hence, it is interesting to obtain results trying to optimize these diameters and its effect comparing them with the previous data obtained.
- ***q* value optimizing the alternative path diameter,** (q_3): These values are required in order to be able to compare the algorithm with the previous brute force based method to obtain the second path (criterion used for the study). In any case the differences over the q_2 option are not very significant (variance at less than 10% of the p values).

At the time of simulating the behavior of the algorithm, some problems were identified for very specific N2R configurations. These problems do not affect the optimal configurations of the network (optimal q) but to define a valid algorithm for all the possible configurations they should be mentioned. In Section VI the problems and the possible solutions are described in depth.

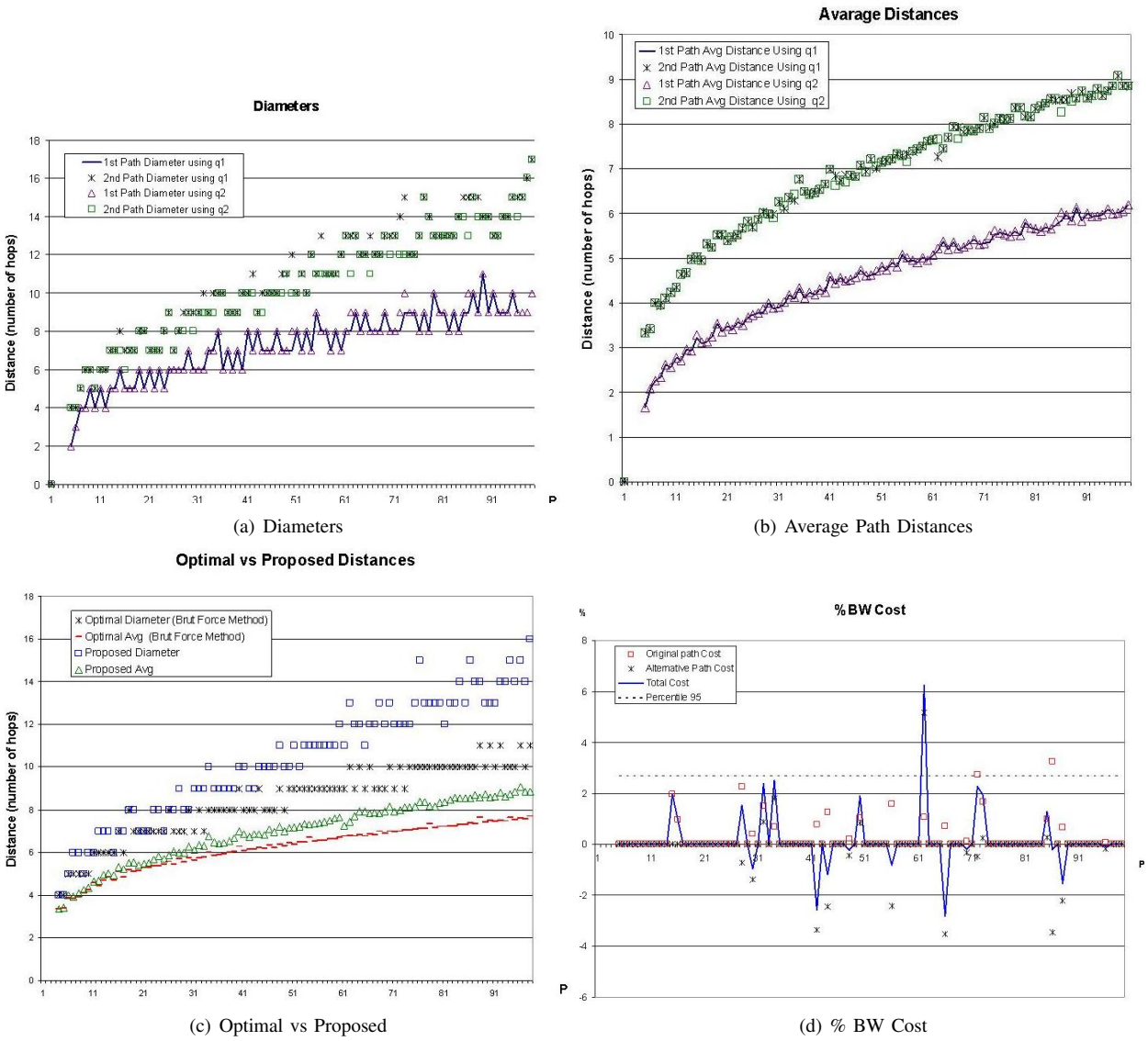
B. Results

The result of the simulation is presented in Fig. 3. Figs. 3(a) and 3(b) illustrate the comparison between the optimal q selecting options (q_1 and q_2) and Fig. 3(c) shows the difference between the proposed algorithm results (using q_3) and the optimal value obtained by a brute force based method. This brute force based method was basically to try all possible combinations of paths and then the best one was selected as pre-calculated paths. This method has some reductions complexity due to the symmetries of the N2R but it stills need high computational resources.

In terms of maximum distances (see Fig. 3(a)), the diameter of the alternative path obtained using q_1 (yellow line) has some peaks at several values of p. These peaks are improved using q_2 (green line). The alternative path diameter has been improved in 21% of the p values. To confirm the validity of this modification, the effect over the first path diameter has to be considered. Comparing the diameters obtained for the two cases (q_1, blue line, and q_2, red line) the diameters remain the same at 95% of the cases. At the rest the difference with the optimal is just of one hop.

To illustrate the effect of this improvement over the complete network (all possible communications) the average path distances are compared. Fig. 3(b) illustrates the values for both paths using the two q (q_1 and q_2) selecting criteria. As expected, the variance of the values for both paths is not very significant. The alternative path average distance (green line) using q_2, depending on the p value, is higher or lower than

[1]The execution time was obtained under the same conditions as the *BA*, since depending on the machine where these algorithms are executed the result will vary. It is assumed though that the proportion will be maintained. The machine used is a Genuine Intel(R) CPU T2050 @1.60 GHz (2 CPU) and 1GB of RAM. The Software use is PHP and MySQL



(a) Diameters

(b) Average Path Distances

(c) Optimal vs Proposed

(d) % BW Cost

Fig. 3. Simulation Graphs

the result using q_1 (yellow line), but still with no significant difference. The original path average distance using q_1 (red line) cannot be improved since the previous studies were focused on minimizing this value. Therefore, it can be assumed as optimal. Comparing it with the result of using q_2 (blue line) there is not a big difference in the cases that is not the same (no influence at 81% of the cases using q_2 over the original path average distance). To understand this comparison better the BW cost of selecting one configuration or the other is commented at Subsection IV-C.

To study the consequences, in terms of distance, of simplifying the path discovery method, Fig. 3(c) illustrates the results obtained with the proposed algorithm and the optimal solution using brute force based method related with the alternative path. The data used for the representation is the one obtained at the case of optimizing the second path diameter (q_3) since is the data available for the brute force based method. As commented before, the difference between using q_2 or q_3 are insignificant.

The result is not very close to the optimal solution in terms of average distances (yellow and red lines) the difference is acceptable assuming the facts proved at [6] which demonstrates that longer paths does not always imply a worst result if the simplicity is good enough to reach the destination faster than more complex algorithm. The problem comes in terms of diameters (blue lines), with the increment of p, the difference increases obtaining a significant variance on the longest possible path. This can be a problem at the time of guaranteeing certain levels of QoS for the applications since even though considering that the average value would meet the requirements, there would be some communications where the performance is worse than the permitted.

C. Results Analysis

The improvement of the alternative path by modifying the q value is not very relevant. In case of the *SPT* method, the use of that alternative path depends on many factors. Among others, the length of the links and the maintenance budget are very relevant factors. The probability failure of a link (cut in the fiber) is directly proportional to its length and the more time is spent in repairing that failure, more time the alternative path is used.

In case that the improvement is an option Fig. 3(d) illustrates the cost in terms of BW the of the selecting q_1 vs. q_2 and the "Percentile 95"[2] is represented as well to give a pseudo maximum for the total cost avoiding the peaks. This cost is assumed for networks under the same conditions (the same network capacity) and it is the same as the percentage of pps (packets per second) reduction.

The cost of implementing the improvement of the alternative path at 50% of the cases (with any variance between the q_1 and q_2 results) there is a not very significant reduction on the total pps in the communications or BW cost. The maximum given by the Percentile 95 is around 2.7%. At the other 50% of the cases the improvement involves a BW cost reduction which is the same as a pps increment in the communications, therefore a BW improvement. These values are assuming that the two paths are equally used, *DPT* method. In case of the *SPT* method the alternative path probably will be used a much smaller percentage of the time. The less time the alternative path is used the closer to the original path cost will be the total cost. In none of the cases this cost would be higher than 3%. Then the improvement could seem feasible using the *DPT* method, but with the *SPT* probably the alternative path usage time is too small to compensate the cost over the original path.

This first approach of protection method proves the possibility of a simple algorithm for protection using topological routing over N2R structures. There is still room for improvement due to the difference between the optimal brute force based distances (optimal) and the ones obtained with the proposed algorithm, see Fig. 3(c). The next step could be to improve the current algorithm to decrease the diameters of the second path. The improvement will add complexity to the algorithm, hence, an analysis of the trade off distance vs path completion time is a good idea to focus on as well.

The execution time of the algorithm at every node is still independent of the value of p. The *BA* study for the original path gives a constant execution time at any node of around 0.035 ms [6]. In case that a failure occurs, the execution time at any node increases due to the addition of some subroutines needed to deal with the problem. The constant execution time at any node in case of a failure of the original path is 0.078 ms, around double the time spent on the original path decision. The execution time is independent of the number of nodes in the network. Therefore, the algorithm is still a good solution for large networks when the table look-up time is too long due to the large number of entries in the routing tables.

[2]These values are calculated considering only the improved cases.

V. Conclusion

The protection problem when dealing with routing techniques is an important factor to consider. The protection solution must be as simple as possible to minimize the transmissions delay, but obtaining reasonable distances as well. Therefore, against other solutions which give better path distances results, an easier algorithm is proposed as a solution.

This algorithm can be improved to obtain shortest alternative independent paths but, as it was demonstrated at the original path study [6], the cost of improving the path length by adding complexity could be too high in terms of delay. The second path approach studied in this paper simplifies the previous methods using brute force.

The fact that if the first and the last links are different at the two paths, both paths are independent has been proved (only very specific exceptions where found not affecting the optimal configurations, commented at Section VI).

The algorithm can be applied to two kinds of failure protection, *SPT* and *DPT*. The use of each of the methods depends on the level of protection required and the traffic nature. Considering these two methods, the topological routing over N2R structures can be a possibility for multipurpose networks where all the different services and applications converge but having different QoS and availability requirements.

There is an interesting conclusion related to the value of q selected as optimal. Depending on the criteria: Optimize original path average length or diameter, optimize the same parameters of the alternative path or both at the same time. If the original optimal value of q (q_1) is modified to improve the alternative path parameters, there is an improvement at about 21% on the cases with no significant affection on the original path parameters. Beyond this optimization, the result cannot be really improved. Depending on the q selecting criteria a small percentage of the N2R configurations are improved by a small factor and at the same time others get worse by similar small factors. Hence, it can be assumed that there is not better results by modifying the q selecting criteria.

Another conclusion is related to the analysis of the difference between the optimal results and the proposed algorithm results. The difference is significant, especially the diameter values. This first approach was performed trying to implement a method as simple as possible to obtain an independent alternative path. The algorithm has to be improved in the way that better distances are obtained with no dramatic increment on the complexity. The execution time of the algorithm, at any node ,in case that a failure occurs is 0.078 ms which is around double the execution time for the algorithm to make a decision when there is no failure (0.035 ms [6]). The algorithm has allowed to take a big step going from a brute forced method to a mathematically based algorithm.

The modification of the *BA* algorithm for being able to route the packets using an alternative path, in case of failure of the original one, stills simple and easy due to the reuse of some data calculated for the original path. Therefore, there are not two different and independent algorithms joined

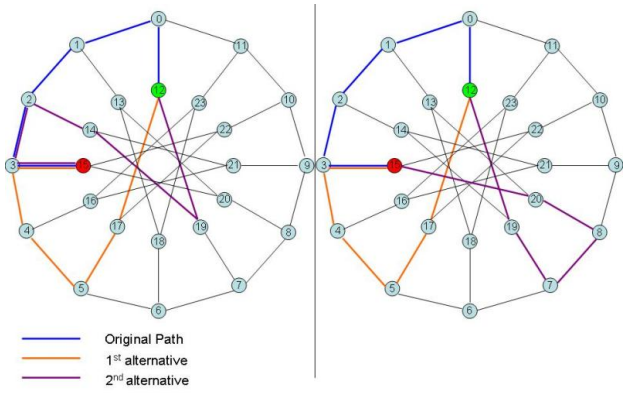


Fig. 4. Center Link Problem

together to obtain the two paths. There is no addition of complex sentences or long subroutines, the additional part of the algorithm keeps following the same basic operations as the previous algorithm (addition and subtraction of addresses and comparisons)

VI. Appendix: Exceptions

At the time of simulating the algorithm some specific problems were identified and the possible solutions are explained next:

Inner node - Inner node communications when the last links are always C (Center for all paths): The problem is that the last link for all the choices will be **C**, hence all the possibilities will not be independent of the original path. This problem is easily solved by adding some sentences to the algorithm to force the packet to go to the node that avoids the problem. Fig. 4 illustrates this idea. The left picture shows the distances calculated and how it would be the path using the three possible links to forward the packet for the source node. Following Fig 4 the final original path obtained with the *BA* algorithm is: *12-0-1-2-3-15*

Then the proposed algorithm will calculate two distances corresponding to the two links available to forward the packet using an alternative path. These calculations take place at the source node (12), both distances are 5 hops and related with the following final paths: *12-17-5-4-3-15* and *12-19-14-2-3-15*. None of these two options are valid since they are not independent of the original path.

The picture on the right shows how the packet is forced to go out of the inner ring to solve the problem. The procedure in this situations is to forward the packet to the next hop setting a flag in the packet to 1 so the middle node knows that it is mandatory to forward the packet to the outer ring . The resulting alternative path will be: *12-19-7-8-20-15*.

Unexpected Original Path: This problem occurs when $q \approx p/2$. The origin is that the distance calculated and the last link identified at the source node corresponds to a solution, but at the middle nodes the algorithm finds a better path to forward the packet. Hence, the last link identified at the source and the real last link used for the original path are different. The consequence is an error obtaining the alternative path. The solution is to add the information about the number of hops taken by the packet using the inner ring and then force the packet to travel using the path estimated at the source. This solution has the problem of a longer paths at the end of the transmission. This information has to be unavoidably added to the packet.

Node coincidence: This is a very specific problem. The error takes places at configurations with more than 100 nodes ($p > 50$) and exactly when the condition (1) is fulfilled.

$$0.17 < p/q - Int(p/q) < 0.3 \quad \& \quad p/q < 4 \tag{1}$$

The problem is that the result of the alternative path has two nodes and one link in common with the original path. This problem can be solved with conditions which just affect to this specific problem by reducing or increasing by one the number of hops using the inner ring but with the consequence of longer alternative paths.

References

[1] Pedersen, Jens Myrup ; Knudsen, Thomas Phillip ; Madsen, Ole Brun. “Topological Routing in Large-Scale Networks”. Proceedings of IEEE/CACT 2004, The 6th International Conference on Advanced Communication Technology. 2004. p. 911-916

[2] Chip Martel and Van Nguyen.“Analyzing Kleinberg’s (and other) small-world Models”, Annual ACM Symposium on Principles of Distributed Computing archive. St. John’s, Newfoundland, Canada, 2004. Pages: 179 - 188 ISBN:1-58113-802-4

[3] D. Watts and S. Strogatz. “Collective dynamics of small-world networks”. Nature, 393:440-442, 1998.

[4] J.M. Pedersen, A. Patel, T.P. Knudsen , O.B. Madsen “Applying 4-regular Grid Structures in Large-Scale Access Networks” Computer

[5] Ivan Stojmenavie, “Honeycomb Netwoks: Topological Properties and Communication Algorithm”, IEEE Transaction on Parallel and Distributed System, Vol8, NO. IO, October 1997.

[6] Jose M. Gutierrez, Ruben Cuevas, Jens M. Pedersen and Ole B. Madsen “Improving Topological Routing in N2R Networks”. CAAN´07, August 2007, Canada.

[7] Jens M. Pedersen, M.Tahir Riaz, Ole B. Madsen “A Simple, Efficient Routing Scheme for N2R Network Structures”, IT&T Annual Conference 2005 - Cork, Ireland . pp 69-80.

[8] R. Frucht, J. E Graver, M. E. Watkins, “The Groups of the generalized Petersen graphs”, Proceedings of the Cambridge Philosophical Society Vol.70 No. 2, September 1971.

[9] O.B. Madsen, T.P. Knudsen and J.M. Pedersen, “SQoS as the Base for Next Generation Global Infrastructure” IT & T 2003. Pag 17.

[10] J.M. Pedersen ; M. T. Riaz ; O.B. Madsen, “Distances in generalized Double Rings and Degree Three Chordal Rings”, Proceedings of IASTED PDCN 2005.

[11] A. Zabludowski, S. Bujnowski, B. Dubalski, “The Evaluation of Transmission Ability of 3rd Degree Chordal Rings with the Use of Adjacent Matrix”, Proc. of The 7th INFORMS telecommunications Conference, Miami, USA, March 2004.

[12] Jie Wu, ”A Fault-Tolerant and Deadlock-Free Routing Protocol in 2D Meshes Based on Odd-Even Turn Model”, IEEE Transactions on Computers. V01.52. No.9, September 2003

[13] T. Jørgensen, L.Pedersen and J.M. Pedersen“Reliability in single, double and N2R ring network structures” The International Conference on Communications in Computing, Las Vegas, Nevada, United States, June 2005