One-to-One and One-to-Many Node-Disjoint Routing Algorithms for RTCC Networks

Saeedeh Bakhshi[1,2], Marzieh Bakhshi[1] and Hamid Sarbazi-Azad[1,2]

[1]Sharif University of Technology, Tehran, Iran
[2]IPM School of Computer Science, Tehran, Iran
E-mail : bakhshi@ ipm.ir, m_bakhshi@ce.sharif.edu, azad@ ipm.ir

ABSTRACT

Recently, routing with disjoint paths has received much attention because disjoint paths have the advantages of efficiency and fault tolerance. On the other hand, RTCC network was studied, due to its favorable properties such as high degree of stability and resilience. In this paper we studied the one-to-one and one-to-many disjoint routings for RTCC network. The connectivity of a network is an important measure of fault tolerance, while the diameter represents the worst-case transmission delay between two arbitrary nodes. The wide diameter, fault diameter and Rabin number, which take connectivity, faults and parallel paths into consideration, are three generalizations of the diameter. We obtained these properties of RTCC network and by using them we showed that in the faulty situations, this network act very well in comparison with the situations free of any fault.

Keywords: *node-disjoint routing, RTCC network, one-to-one routing, one-to-many routing*

1. INTRODUCTION

The Recursive Transpose-Connected Cycles (*RTCC*) Network is a class of recursively scalable networks, denoted by *RTCC(C, L)*, that is constructed hierarchically by grouping basic cycle modules. Any *C*-node cycle can serve as the basic modules.

With continuous increases in network size, routing in networks with faulty nodes has become unavoidable. Routing through node-disjoint paths in interconnection networks can not only provide alternative routes to tolerate faulty nodes but also avoid communication bottlenecks. Moreover, routing through node-disjoint paths can speed up the transmission time by distributing data among disjoint paths. Thus, the study of disjoint paths connecting any two nodes can be useful for increasing the reliability of interconnection networks, as well as transmission efficiency. A larger number of disjoint paths is more desirable because of less vulnerability to disconnection. The study of node-disjoint paths varies according to the number of source and destination nodes. There are three well-known paradigms: *one-to-one routing* that constructs the maximum number of node-disjoint paths in the network between two given nodes, *one-to-many routing* that constructs node-disjoint paths in the network from a given node to a given set of nodes, and *many-to-many routing* that constructs node-disjoint paths between a given set of nodes. Using these paradigms, node-disjoint paths have been extensively studied on some networks [9, 10, 11].

In this paper, we studied the fault tolerance properties and node-disjoint algorithms in RTCC network. In section 2, we formally define the RTCC topology and a number of useful notations and definitions to be used in subsequent sections. In section 3, we obtained the connectivity and fault-diameter of the network and proposed an algorithm for one-to-one node-disjoint paths in RTCC network. The wide-diameter of the network was obtained in this section, too. Section 4, studies the one-to-many node-disjoint routing algorithm and the Rabin-number of RTCC network. Finally, section 5 concludes this study.

2. DEFINITIONS AND PRELIMINARIES

It is natural to model interconnection networks with graphs that have nodes representing processing units and communication units (switch) connected with edges representing data streams between the nodes. The RTCC network [8] is a class of recursively scalable networks denoted as *RTCC*(*C*, *L*) that is constructed by hierarchically grouping basic modules each a *C*-node cycle graph. *RTCC* (*C*, *L*) consists of *C RTCC* (*C*, *L*–1)'s each of them could be considered as a super-node. These *RTCC* (*C*, *L*–1)'s are connected as a complete graph.

Definition 1. A C-node cycle consists of a set of nodes {0, 1, 2, 3, …, C-1} and set of edges $\{e_0, e_1, e_2, ..., e_{C-1}\}$ such that $e_i = (i , i+1 \bmod C)$.

Definition 2. The definition of the *RTCC*(C, L) is based on a C-node cycle. We name all the nodes in this cycle, '*Extern nodes*' or '*Open nodes*'. An *RTCC*(C, 2), consists of a number of C discrete *RTCC*(C, 1) networks, or C-node cycles, numbered 0 to C-1. Each external node *i* of each C-node cycle *j* is connected to node *j* of C-node cycle *i*. It is obvious that a node whose number is equal to the number of the *RTCC*(C, 2) in which it resides, is not directly connected to any other cycle, and is of node degree one less than other nodes in the network. There is one such node in each *RTCC*(C, 1) used to construct an *RTCC*(C, 2), and thus a total of C such nodes. We name these nodes as the external nodes of the *RTCC*(C, 2), the number of each one being equal to the number of the *RTCC*(C, 1) to which it belongs. In a similar manner, the *RTCC*(C, 3) can be defined as C discrete *RTCC*(C, 2) networks that are connected in such a way that each external node *i* in *RTCC*(C, 2) number *j* is connected to external node *j* in *RTCC*(C, 2) number *i*. Once again, a node whose number is equal to the address of the *RTCC*(C, 2) in which it resides, is not directly connected to any other cycle, and is of node degree one less than other nodes in the network. We name these nodes, of which there is a total of C, the external nodes of the *RTCC*(C, 3). Higher level *RTCC* networks can be defined in a similar manner. In Fig 1, the connection between nodes of an *RTCC* (4, 2), *RTCC* (5, 3) and an *RTCC* (6, 1) are displayed.

The node set of an *RTCC*(C, L) can be expressed as $\{(a_L a_{L-1} \ldots a_1) \mid a_i \in C, 1 \le i \le L\}$ where C = {0, 1, ..., C – 1}. Therefore, the number of nodes of an *RTCC*(*C*, L) is equal to $|RTCC(C, L)|_v = C \times |RTCC(C, L\text{-}1)|_v = C^L$. On the other hand, the number of edges of an *RTCC*(*C*, *L*) is equal to $|RTCC\ (C, L)|_c = C \times |RTCC(C, L\text{-}1)|_e + C(C\text{-}1)/2 = (3C^L - C)/2$. The degree of extern nodes of an *RTCC*(C, L) is 2 and the degree of all other nodes is 3. Therefore, the degree of the network is fixed and equal to 3.

$(a_L a_{L-1} \ldots a_1)$ are: $(a_L a_{L-1} \ldots (a_1\text{-}1)_{\bmod C})$ and $(a_L a_{L-1} \ldots (a_1+1)_{\bmod C})$ in the same lowest-level sub-graph, which we refer to as *sister* nodes, and $(a_L a_{L-1} \ldots a_{j+2} a (a_{j+1})^j)$, $1 \le j \le L\text{-}1$, where $a_1 = a_2 = \ldots = a_j = a \ne a_{j+1}$, corresponding to the connection between the external nodes of level-*j* sub-graphs. Notation $(a)^j$ denotes *j* consecutive *a*'s. We refer to this adjacent node as a *cousin* node. It is obvious that a node whose address is of the form $(a_x)^L$, i.e. the address of all sub-graphs to which the node belongs to are the same, is of no cousin. These are the same nodes we refer to as *extern node*.

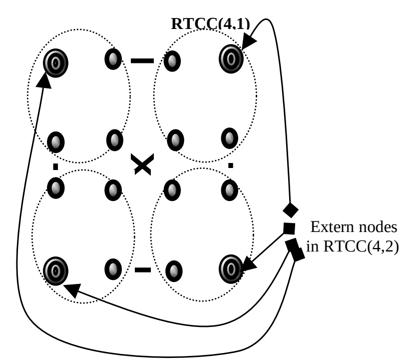

Fig. 1: The RTCC (6, 1), RTCC (4, 2), and RTCC (5, 3) networks

there does not exist a set of $k - 1$ vertices whose removal disconnects the graph. The connectivity of G is shown with $\kappa(G)$.

Definition 4. If G is a connected graph and u and v are two of its nodes, A (u, v)-*container* in G [4], denoted by $C(u, v)$ is a set of node-disjoint paths between u and v. The number of paths in $C(u, v)$ is called the *width* of $C(u, v)$, denoted by $w(C(u, v))$.

Definition 5. The maximal length of paths in $C(u, v)$ is called the *length* of $C(u, v)$, denoted by $l(C(u, v))$. A $C(u, v)$ is the *best* if its length is minimum. We use $C_x(u, v)$ to denote a $C(u, v)$ with width x, and $C_x^*(u, v)$ to denote a best $C_x(u, v)$, where $x \geq 1$. The *x-wide distance* between u and v is defined as $l(C_x^*(u, v))$, where $w(C_x^*(u, v)) = x$ and $x \geq 1$ [4].

Definition 6. The *x-wide diameter* of G, denoted by $wd_x(G)$, is defined as the maximum x-wide distance between two arbitrary nodes of G[4].

Definition 7. The $(k - 1)$-fault diameter $FD_{k-1}(G)$ of a k-connected graph G is the maximum diameter of $G - F$ for any $F \subset V(G)$ with $|F| < k$.

Definition 8. The *z-Rabin number* of G, denoted by $rn_z(G)$, is defined as the the minimal l so that for any z+1 distinct nodes $u, v_1, v_2, \cdots, v_z$, there exist z disjoint paths of lengths at most l from u to $v_1, v_2, \cdots, v_z$ respectively, where $z \geq 1$ [12].

Definition 9. The *strong z-Rabin number* of G, denoted by $RN_z(G)$, has the same definition as the z-Rabin number, except that $v_1, v_2, \cdots, v_z$ are not necessarily distinct [5].

Compared with diameter, wide diameter measures the maximal length of all best containers. Fault diameter estimates the maximal increment of diameter when there are node faults. When the multicasting problem is concerned, Rabin number estimates the minimal transmission delay with maximal parallelism. According to Menger's theorem [2], there are $\kappa(G)$ disjoint paths between every two distint nodes of G. Moreover, there are $\kappa(G)$ disjoint paths from one node to other $\kappa(G)$ distinct nodes in G. Hence $wd_x(G)$, $fd_y(G)$ and $rn_z(G)$ are defined for $x \leq \kappa(G)$, $y < \kappa(G)$ and $z \leq \kappa(G)$.

3. ONE-TO-ONE NODE-DISJOINT ROUTING ALGORITHM

In this section we obtain the fault diameter and wide diameter of RTCC network and propose a routing algorithm for one-to-one node-disjoint paths of it.

Lemma 1. [1] The node connectivity of *RTCC* (C, L) is 2.

Lemma 2. The edge connectivity of *RTCC* (C, L) is 2.

Proof. Since the minimum vertex degree of *RTCC* (C, L) is 2, it will be its edge connectivity, too.

Lemma 3. [1] The diameter of *RTCC* (C, L) is

$$2^{L-1}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$$

First we want to check the diameter of the network when $d - 1$ nodes are faulty. We first give a lower bound on the fault diameter of *RTCC* (C, L).

Theorem 1. [1]

$$FD_2\left(RTCC\left(C, L\right)\right) \leq 2^L + 2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$$

See Fig. 2 for an example of fault diameter in *RTCC* (5, 2).

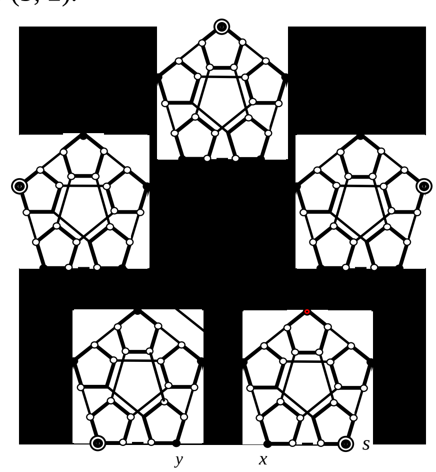

Fig. 2: Construction of the shortest path between two maximum distance nodes in faulty *RTCC* (5, 2) network

By the definition of WD_x (G), we have WD_x (G) = max $\{l(C_x^*(u, v)) \mid$ for all pairs of nodes u, v in $G\}$

Lemma 4. There exist two node-disjoint paths from an arbitrary node to two of extern nodes of *RTCC* (*C*, *L*) with length at most $2^{L-1}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$, which equals its diameter.

Proof. We use induction to prove the lemma. For the case of $L = 1$, *RTCC* (*C*, *L*) is a cycle with *d* nodes and there are two node-disjoint paths between each pair of nodes in a cycle with length at most $\lceil C/2 \rceil$. We assume that the claim is true for *RTCC* (*C*, *L* – 1), it is enough to show that it works for *RTCC* (*C*, *L*). Assume you have a *RTCC* (*C*, *L*) network with the source node $S = (s_L s_{L-1} \ldots s_1)$ and the destination extern nodes $T_1 = \left((t_1)^L\right)$ and $T_2 = \left((t_2)^L\right)$. From the above assumption, we know that we have two node-disjoint paths in $s_i th\, RTCC(C, L-1)$-subgraph from $S = (s_L s_{L-1} \ldots s_1)$ to $X_i = \left(s_L t_i^{L-1}\right) : 1 \le i \le 2$. Up to now, we have at least two node-disjoint paths of length at most $2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$ from $S = (s_L s_{L-1} \ldots s_1)$ to nodes $X_i = \left(s_L t_i^{L-1}\right) : 1 \le i \le 2$. Now, we can route each of the paths from $X_i = \left(s_L t_i^{L-1}\right) : 1 \le i \le 2$ to $Y_i = \left(t_i s_L^{L-1}\right) : 1 \le i \le 2$ by using an edge, and then we have each of the paths in a distinct *RTCC* (*C*, *L* – 1)-subgraph, and we can route it to the desired extern node by traversing at most *D*(*RTCC* (*C*, *L*–1)) edges. Therefore, the length of this path is at most

$$2\left(2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1\right) + 1 = 2^{L-1}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$$

and the proof is complete. (See Fig. 3)

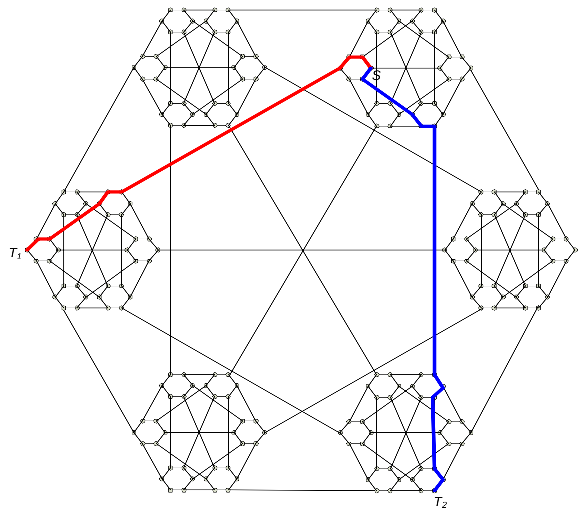

Fig. 3: Construction of two node-disjoint paths from S to two extern node destinations of *RTCC*(6,3)

Lemma 5. $w\left(C(u,v)\right) = 2$ where $\forall u, v \in V\left(RTCC(C, L)\right)$

Proof. According to the Menger's theorem [2], there are $\kappa(G)$ node disjoint paths from one node to another, where $\kappa(G)$ is the node-connectivity of *G*. Since, from lemma 1, we know that the node-connectivity of v is two, the claim results.

Theorem 2. $l\left(C_2^*(u,v)\right) \le 3.2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$ for every two distinct nodes *u* and *v* of $RTCC(C, L)$

Proof. To prove this theorem, it suffices to show two disjoint paths between *u* and *v* whose lengths are shorter than 3.2^{t-1} – 1. Suppose that $u = (u_L u_{L-1} \ldots u_1)$ and $v = (v_L v_{L-1} \ldots v_1)$. We know that there are two disjoint paths between an arbitrary pair of nodes in the cycle of *C* nodes. If $u_L u_{L-1} \ldots u_1 = v_L v_{L-1} \ldots v_1$ for $i > 1$, the problem of finding the node-disjoint paths in *RTCC* (*C*, *L*) converts to the problem of finding the node-disjoint paths in *RTCC* (*C*, *L* – 1), which is the same as one that will be discussed here. Otherwise, either *u* and *v* belongs to the same cycle and there exist two disjoint paths between them with length at most $\lceil C/2 \rceil$ or $u_L \neq v_L$.

Therefore, it is enough to compute the maximum length of two node-disjoint paths from *u* to *v*, in different *RTCC* (*C*, *L*–1)-subgraphs. From lemma 4, we know that we can construct node disjoint paths of length at most $2^{L-1}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$ to all extern nodes in every *RTCC* (*C*, *L*–1) subgraphs (i.e. L-frontiers of *RTCC* (*C*, *L*). Consider the path from $u = (u_L u_{L-1} ... u_1)$ to two frontiers of the form $w = \left(w_L {v_L}^{L-1}\right) : 1 \leq w_L \leq C, w_L \neq v_L$ of length at most $2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$, we can continue these paths by traversing one edge in each, going to $w = \left(v_L {w_L}^{L-1}\right) : 1 \leq w_L \leq C, w_L \neq v_L$. Nodes of the form $w = \left(v_L {w_L}^{L-1}\right) : 1 \leq w_L \leq C, w_L \neq v_L$ are (*L* – 1) frontier nodes of $v_L th$ *RTCC* (*C*, *L* – 1) subgraph. Therefore there are two node-disjoint paths from $v = (v_L v_{L-1} ... v_1)$ to these (*L* – 1)-frontiers of length at most $2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$. Hence the length of all two node-disjoint paths from *u* to *v* in *RTCC* (*C*, *L*) would be

$$3\left(2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1\right) + 2 = 3.2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$$

and the theorem follows. Fig. 4 shows five node-disjoint paths built according to the proof of theorem in *RTCC* (6, 3).

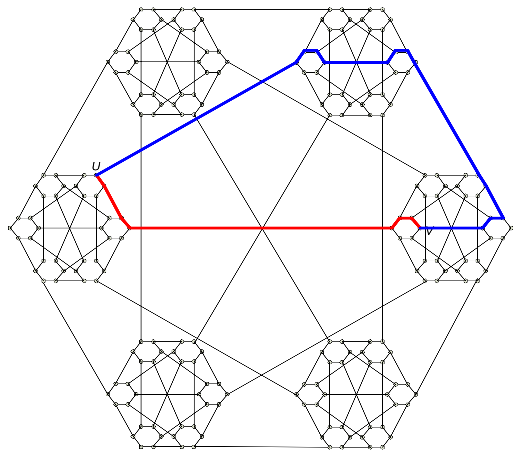

Fig. 4: constructing two node-disjoint paths of according to the proof of theorem in *RTCC* (6, 3)

Now, we can propose the one-to-one routing algorithm in *RTCC* (*C*, *L*) where the paths are node-disjoint.

Algorithm *RTCC(C,L)*-Disjoint Routing

Input: Shortest path routing algorithm in *RTCC* (*C*, *i*) $R_i : 1 \leq i \leq L-1$, in source vertex $u = (u_L u_{L-1} ... u_1)$ destination vertex $v = (v_L v_{L-1} ... v_1)$.

Output: two disjoint paths $P_i : i = 1, 2$ connecting *u* to *v*.

Begin

1. i := L;

2. while ($u_i = v_i$) do

2.1 i := i – 1

3. Route P_1 from $u = (u_L u_{L-1} ... u_1)$ to $x = \left(u_L u_{L-1} ... u_i \left(v_i\right)^{i-1}\right)$ using the routing R_{i-1}

4. Route P_1 from $x = \left(u_L u_{L-1} ... u_i \left(v_i\right)^{i-1}\right)$ to $x' = \left(u_L u_{L-1} ... v_i \left(u_i\right)^{i-1}\right)$ by traversing an edge

5. Route P_1 from $x' = \left(u_L u_{L-1} ... v_i \left(u_i\right)^{i-1}\right)$ to $v = \left(v_L v_{L-1} ... v_1\right)$ using the routing R_{i-1}.

6. Route P_2 from $u = \left(u_L u_{L-1} ... u_1\right)$ to $x = \left(u_L u_{L-1} ... u_i \left(x_i\right)^{i-1}\right) : x_i \neq v_i, u_i$ using the routing R_{i-1}.

7. Route P_2 from $x = \left(u_L u_{L-1} ... u_i \left(x_i\right)^{i-1}\right) : x_i \neq v_i, u_i$ to $x' = \left(u_L u_{L-1} ... x_i \left(u_i\right)^{i-1}\right)$ by traversing an edge

8. Route P_2 from $x' = \left(u_L u_{L-1} ... x_i \left(u_i\right)^{i-1}\right)$ to $y = \left(u_L u_{L-1} ... x_i \left(v_i\right)^{i-1}\right)$ using the routing R_{i-1}.

9. Route P_2 from $y = \left(u_L u_{L-1} ... x_i \left(v_i\right)^{i-1}\right)$ to $y' = \left(u_L u_{L-1} ... v_i \left(x_i\right)^{i-1}\right)$ by traversing an edge.

10. Route P_2 from $y' = \left(u_L u_{L-1} ... v_i \left(x_i\right)^{i-1}\right)$ to $v = \left(v_L v_{L-1} ... v_1\right)$ using the routing R_{i-1}

End

4. ONE-TO-MANY NODE-DISJOINT ROUTING ALGORITHM

In this section, we study an algorithm that presents two node-disjoint paths from an arbitrary node in *RTCC* (*C*, *L*) to two distinct target nodes. This algorithm is a recursive one and is based on the routing algorithm for unicast and multicast in $RTCC(C,i): 1 \leq i \leq L-1$.

Algorithm Disjoint Multicast Routing (s, T_1, T_2, C, L)

Input: Shortest path routing algorithm in *RTCC* (*C*, *i*) : $R_i :: 1 \leq i \leq L-1$, in source vertex $s = \left(s_L s_{L-1} ... s_1\right)$, destination vertices $T^1 = \left(t_L^1, t_{L-1}^1, ..., t_1^1\right)$ and $T^2 = \left(t_L^2, t_{L-1}^2, ..., t_1^2\right)$.

Output: two disjoint paths connecting *s* to T_1 and T_2

Begin

1. i := L, j:=L;
2. while ($s_i = t_i^1$) do
   2.1 i := i – 1
3. while ($s_j = t_j^2$) do
   3.1 j := j – 1
4. if (j < i) then
   4.1 Disjoint Multicast Routing $\left(s_L s_{L-1} ... s_j \left(t_j^1\right)^{j-1}\right)$, $\left(s_L s_{L-1} ... s_j \left(t_j^2\right)^{j-1}\right)$, C, i)
   4.2 Route P_2 from $\left(s_L s_{L-1} ... s_j \left(t_j^2\right)^{j-1}\right)$ to $\left(s_L s_{L-1} ... t_j^2 \left(s_j\right)^{j-1}\right)$ by traversing an edge
   4.3 Route P_2 from $\left(s_L s_{L-1} ... t_j^2 \left(s_j\right)^{j-1}\right)$ to $T^2 = \left(t_L^2, t_{L-1}^2, ..., t_1^2\right)$ using the routing R_{j-1}
   4.4 Route P_1 from $\left(s_L s_{L-1} ... s_j \left(t_j^1\right)^{j-1}\right)$ to $\left(s_L s_{L-1} ... s_i \left(t_i^1\right)^{i-1}\right)$ using the routing R_{i-1}
   4.5 Route P_1 from $\left(s_L s_{L-1} ... s_i \left(t_i^1\right)^{i-1}\right)$ to $\left(s_L s_{L-1} ... t_i^1 \left(s_i\right)^{i-1}\right)$ by traversing an edge
   4.6 Route P_1 from $\left(s_L s_{L-1} ... t_i^1 \left(s_i\right)^{i-1}\right)$ to $T^1 = \left(t_L^1, t_{L-1}^1, ..., t_1^1\right)$ using the routing R_{i-1}
5. else
   5.1 Disjoint Multicast Routing (s, $\left(s_L s_{L-1} ... s_i \left(t_i^1\right)^{i-1}\right)$, $\left(s_L s_{L-1} ... s_i \left(t_i^2\right)^{i-1}\right)$, C, i)
   5.2 Route P_1 from $\left(s_L s_{L-1} ... s_i \left(t_i^1\right)^{i-1}\right)$ to $\left(s_L s_{L-1} ... t_i^1 \left(s_i\right)^{i-1}\right)$ by traversing an edge

5.3 Route P_1 from $\left(s_L s_{L-1} \ldots t_i^1 (s_i)\right)^{i-1}$ to $T^1 = \left(t_L^1, t_{L-1}^1, \ldots, t_1^1\right)$ using the routing R_{i-1}

5.4 Route P_2 from $\left(s_L s_{L-1} \ldots s_i \left(t_i^2\right)^{i-1}\right)$ to $\left(s_L s_{L-1} \ldots s_j \left(t_j^2\right)^{i-1}\right)$ using the routing R_{j-1}

5.5 Route P_2 from $\left(s_L s_{L-1} \ldots s_j \left(t_j^2\right)^{i-1}\right)$ to $\left(s_L s_{L-1} \ldots t_j^2 (s_j)\right)^{i-1}$ by traversing an edge

5.6 Route P_2 from $\left(s_L s_{L-1} \ldots t_j^2 (s_j)\right)^{i-1}$ to $T^2 = \left(t_L^2, t_{L-1}^2, \ldots, t_1^2\right)$ using the routing R_{j-1}

End

Theorem 4. $rn_2\left(RTCC(C,L)\right)$

$$= RN_2\left(RTCC(C,L)\right) = 3.2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$$

Proof. According to the above algorithm, we can route each path by traversing at most the source *RTCC* (*C*, *L*–1)-subgraph, one intermediate *RTCC* (*C*, *L*–1)-subgraph and the last *RTCC* (*C*, *L*–1)-subgraph. Since routing in *RTCC* (*C*, *L*–1)-subgraphs are like that of theorem 3, these paths have the maximum length of $3.2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$. The above reasoning is true in the situation that some destination nodes are not distinct, so the proof gets complete.

Fig. 5 shows an example of constructing multicast routing in a *WK* (6, 3) by the use of mentioned algorithm.

Since, we obtained the generalized diameters of RTCC networks with the size of $3.2^{L-2}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$, now we can have a comparison with the original diameter of the network

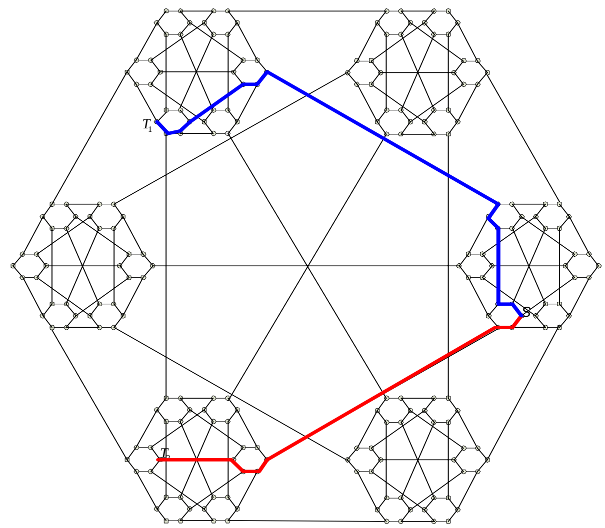

Fig. 5: Constructing multicast node-disjoint routing in a *RTCC* (6, 3)

$2^{L-1}\left(\left\lfloor \frac{C}{2} \right\rfloor + 1\right) - 1$. While the RTCC network has a modular topology with variables *C* and *L*, the length of various disjoint paths and fault diameter of the network depends on none of them, but on a constant factor. Therefore, the generalized diameters of RTCC network, shows its good resilience in faulty situations.

5. CONCLUSION

In this paper, we proposed algorithms for one-to-one and one-to-many node-disjoint routing algorithms for RTCC network. We computed the fault diameter, the wide diameter and rabin number of the network and concluded in a constant increase of them according to the original diameter. Therefore, the RTCC network provides good characteristics in implementing various node-disjoint paths and the diameter of the network according to the disjoint paths is not much more than the original diameter, which proves the strong fault tolerance properties of such networks.

REFERENCES

1. J. A. Bondy and U. S. R. Murty, “Graph Theory with Applications,” North-Holland, NY, 1976.
2. F. Cao, D. Z. Du, D. F. Hsu, and S. H. Teng, “Fault

tolerance properties of pyramid networks," IEEE Transactions on Computers, vol. 48, no. 1, pp. 88-93, 1999.

3. D. F. Hsu, "On container width and length in graphs, groups, and networks," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science, vol. E77-A, no. 4, pp.668-680, 1994.
4. S. C. Liaw and G. J. Chang, "Generalized diameters and Rabin numbers of networks," Journal of Combinatorial Optimization, vol. 2, no. 4, pp. 371-384, 1999.
5. R. Fernandes, D.K. Friesen, and A. Kanevsky, "Efficient Routing and Broadcasting in Recursive Interconnection Networks," Proc. 23rd Int'l Conf. Parallel Processing. Volume 1: Architecture, pp. 51-58,1994.
6. C. C. Chen and J. Chen, "Nearly Optimal One-to-Many Parallel Routing in Star Networks," IEEE Trans. Parallel, Distrib. Syst., vol. 8, pp. 1196-1202, 1997.
7. M. Dietzfelbinger, S. Madhavapeddy, and I. H. Sudborough, "Three Disjoint Path Paradigms in Star Networks," Proc. 3nd IEEE Symposium on Parallel and Distributed Processing, Dallas, TX, pp. 400-406, 1991.
8. Q.-P. Gu and S. Peng, "An Efficient Algorithm for *k*-Pairwise Disjoint Paths in Star Graphs," Information Processing Letters, vol. 67, pp. 283-287, 1998.
9. D. R. Duh and G. H. Chen, "On the Rabin number problem," Networks, vol. 30, no. 3, pp. 219- 230, 1997.
10. M. Hoseiny Farahabady, H. Sarbazi-Azad, "The Recursive Transpose-Connected Cycles (*RTCC*) Interconnection Network for Multiprocessors", *ACM SAC'05*, (March 2005), Santa Fe, New Mexico, USA.
11. M. Hoseiny Farahabady, H. Sarbazi-Azad, "The RTCC-pyramid: a versatile pyramid network", High-Performance Computing in Asia-Pacific Region, 2005. Proceedings. Volume, Issue , 30 Nov.-3 Dec. 2005.