Sasa KLAMPFER[1], Joze MOHORKO[2], Zarko CUCEJ[2], Amor CHOWDHURY[1]

Margento Research & Development d.o.o., 2000 Maribor, Slovenia (1)
University of Maribor, Faculty of Electrical Engineering and Computer Science, 2000 Maribor, Slovenia (2)

Graph's theory approach for searching the shortest routing path in RIP protocol: a case study

***Abstract**. Routing is a problem domain with an infinite number of final-solutions. One of the possible approaches to solving such problems is using graph theory. This paper presents mathematical analysis methodologies based on circular graphs for solving a shortest path routing problem. The problem is focused on searching for the shortest path within a circular graph. Such a search coincides with the network routing problem domain. In this paper, we introduce in the detail all necessary parts needed to understand such an approach. This includes: definition of the routing problem domain, introduction to circular graphs and their usage, circular graph's properties, definition of walks through a circular graph, searching and determining the shortest path within a circular graph, etc. The state of the art routing methods, implemented in contemporary highly sophisticated routers, includes well-known weight-based algorithms and distance-vectors-based algorithms. The proposed solution can be placed between the two abovementioned methods. Each of these known methods strives for optimal results, but each of them also has its own deficiencies, which should be rectified with the proposed new method. This theoretically presented method is argued by a practical example and compared with the RIP (Routing Information Protocol) technique, where we look for the shortest path and possible walks through a specified circular graph.*

***Streszczenie.** W artykule zaprezentowano matematyczną analizę bazująca na teorii grafów do rozwiązania problemu poszukiwania najkrótszej ścieżki routingu. Przedstawiono problem routingu oraz grafy kołowe i ich użycie. (**Wykorzystanie teorii grafów do poszukiwania najkrótszej ścieżki routingu w protokole RIP**)*

Keywords: Circular graphs, shortest path, routing, graph diameter, walk through, CIGRP, connectivity matrix, network topology, symmetry, fully connected graph.
Słowa kluczowe: grafy kołowe, routing, topologia sieci

Introduction

Contemporary highly-sophisticated network applications represent a leading circumstance which dictates continuing development of new innovative and effective approaches for improving networks' performances. One of the most critical problems of the communication domain still remains the routing problem [1]. It has perceivable influence on packet delays, probability that packets will reach its destinations as well as on the link utilization between individual nodes. A routing algorithm belongs to the so-called NP complexity group [32], where an optimal solution for a specific problem could be found in polynomial time. We are usually satisfied with every good solution, but generally such a solution besides advantages also has its disadvantages. One of the solutions can be found with the routing method based on circular graphs, which will be introduced in this paper.

It is possible to find numerous articles related to circular graph theory [33] in many bibliographic databases. This theory being mostly used in the mathematical area [2], but also in other areas such as: multiprocessor systems [3], parallelism and redundancy identification [4], approach to recognizing Hamming properties [5], searching intersections of longest paths [6], wireless sensor networks [8] where spanning three graphs are mostly used [7], etc. In practice, network planners and innovators avoid network topologies that incorporate cycles, such as circular graphs definitely are. Such cyclic connections are usually undesired. Other routing protocols remove the cyclic or redundant connections from routing tables, RIP [9], for example. With this research, we make a step forward and present the main idea; how the circular graph theory can be useful in routing algorithms without fear that the traffic will start cycling. The concept introduced in the following chapters is brand new and very radical, because some criteria (symmetry, full connectedness and cyclic) must be fulfilled before such a concept can be implemented. The original contribution of this research is the idea about CIGRP (Circular Graph Routing Protocol), which will be based on circular graph theory and will be able of mutual operation with other routing protocols. In this paper, we show how to use circular graphs as a fundamental part of the routing protocol and at the same time the concept of integration of this idea into the existing protocols (RIP for example). An original contribution is also a divided mutual interoperability concept of deciding, which routing method will be in use at a specific moment considering actual network topology and nodes connectivity.

The reason why we propose a new routing algorithm based on circular graph topology is tightly connected with the topology simplifying aspect. With the help of the connectivity matrix, the algorithm is capable to find the shortest path between nodes u and v in a very simple way. This path is calculated from a set of all possible paths connecting nodes u and v by using a mathematical procedure presented within next subsections [35]. In this case, we do not need to calculate distance vectors based on Bellman-Ford algorithm [10], which is used in RIP. This algorithm is used in most routing protocols based on distance vectors, but has the following disadvantages:

- Network topology changes are not reflected fast enough, because they travel from one node to another. Such procedure continues till all nodes are informed about the change (our concept also has this disadvantage).
- Cycling to infinity which appears if one node becomes unreachable, whereas its neighbours increase distance estimation to infinity; during such a scenario packet cycling can appear (for our case, we present two methods of preventing those scenarios).

The benefit of our approach is limited only to network segments, which can be described with circular graph topology, where we can search for the shortest path in a simple way. In other cases the proposed method is excluded. This is also the reason, why we always combine CIGRP with another major routing protocol (in our case RIP).

In the second chapter, we want to acquaint the reader with the basis of the routing problem domain [1]. This chapter also introduces certain state of the art solutions implemented into modern routers. One of these is the routing algorithm based on distance vectors [11]. The third chapter presents the definitions of the basic circular graph properties including circular graph symmetry, circular graph diameter and connectedness between nodes within a specific circular graph. The same section also presents one of the possible examples of circular graph implementation. The fourth section presents the procedure of exploring available paths, connections, etc. We introduce a method which, within a set of all connections, searches for the optimal connections between communicating nodes of the circular graph structure. An optimal path is always composed with the best particular connections between individual nodes along the whole routing path. The fifth section presents the mathematical derivation of the equations with which the optimal path in a circular graph is calculated. The sixth section presents the cyclic structure searching procedure upon real network or network segment, meanwhile in seventh section is presented the proposal, how can be the proposed CIGRP concept included into existing routing protocols, such RIP is, etc. The eighth section concludes the paper.

Routing problem domain

The packet exchange problem can be generally described in the following manner; let the given network be constructed of n nodes, each of them containing zero or more packets, and let there be a well-known destination node for each packet. We want to find an appropriate routing algorithm procedure which will allow packet exchange according to the limitations arising from the topology and from the routing model. The simplest example represents a scenario where each packet has exactly one destination, and for each destination node, there is exactly one packet. Such a routing problem is called 'one-to-one'. But there also exists a 'many-to-one' routing problem, where many packets have the same destination target. The opposite of this is the so-called 'one-to-many' routing problem, where one packet has more than one destination addresses. In this case, the algorithm sends more copies of the same packet to different destinations. Another well-known routing problem is when resource packets appear during the routing procedure. In this case, we have to deal with simultaneous routing. An example of such a case is when packets are generated within the routing procedure and do not have defined destinations. The destination is then calculated within the routing procedure. Packets without defined destination addresses can be understood as an additional routing parameter, which serves as background information on where the new packets with a precisely defined destination address originate.

General routing problems is a group of routing problems, where the total number of packets is predefined and well-known before routing starts. These packets have predefined destinations. Such a problem can also be called a (N, p, k_1, k_2)-routing problem, where N represents the total number of packets at the start stored in p nodes. At the start, each node cannot have more than k_1 packets, and at the end each destination node cannot have more than k_2 packets stored. General routing problems usually disperse into several problems divided into different classes.

An example of 'many nodes–same message' represents a scenario where many nodes transmit or receive the same message within the routing sphere, which means that we have to deal with 'one-to-many' and 'many-to-one' routing problems. One-node transmitting is when one node transmits the same messages to other nodes within the network. With 'many-node' transmitting (broadcasting) each of the n nodes transmits one message to all other nodes. This is simultaneous n-multiply one-node transmitting. When such a scenario is used, n different messages appear within the network. 'One-node' collecting is the opposite operation of 'one-node' transmitting, and the 'multi-node' collecting procedure is the opposite of the multi-node transmitting. Permutation routing problems are also well-known. These are problems when exactly one packet is destined to node $\pi(x)$ and has its start point in each node x, where π represents the permutation of the node-set for a given network. Such a scenario is also called a group of permutation routing problems.

Introduction to circular graphs

As the name suggests, the k-circular graph is constructed of k repetitions of specified circular (cyclic) structure. This representation is, under specified conditions (more about these conditions later), completely correct because a 1-circular graph represents a cycle, while a 2-circular graph represents two interlaced cycles and a k-circular graph k interlaced cycles. Figures 1a and 1b illustrate the simplest 1-circular graph example $G(7;\pm1)$ represented in two different ways. The difference between both graphs is only in the node's arrangement on a flat surface; graphs 1a and 1b are isomorphic, because both represent a cyclic length of 7. From this point of view the human eye can detect these two graphs as being different, but ultimately both have one cycle. The nodes in Figure 1b are not directly connected with their neighbors (if we are looking at nodes arranged on a flat surface), but are connected with their neighbors' neighbors. If we rename the nodes in graph 1b (nodes are marked from 0 to 6 in the anti-clockwise direction), then we obtain the $G(7;\pm1)$ graph's isomorphic 1-circular graph $G(7;\pm2)$ (Figure 1c). In this graph node 0 is connected with node 2, node 1 with node 3, node 2 with node 4 and so on. Similarly, a graph based on seven nodes, where each node is connected to its neighbors' neighbor (depending on node arrangements on a flat surface), can be marked as $G(7;\pm3)$ (Figure 1d). The graphs shown in Figures 1a, 1c and 1d represents three different 1-circular graphs based on seven nodes. The fact is that graphs $G(7;\pm2)$ and $G(7;\pm3)$ are isomorphic in comparison to graph $G(7;\pm1)$, which is the reason why generalization of 1-circular graphs cannot be made. On $G(8;\pm2)$, it is noticeable that this graph bisects into two cycles of length 4 (Figure 1f). From this aspect, it is not isomorphic to cycle $G(8;\pm1)$. Graph $G(n;\pm h)$ is isomorphic to graph $G(n;\pm1)$ when and only when the largest common divider of numbers n and h is 1 ($gcd(n, h)=1$)). As with 1-circular graphs with cycle lengths n, we can similarly state this (a bit superficially) for two-cycle graphs. These represent graphs, which consist of two interlaced cycles. An example of two interlaced cycles is illustrated in Figure 1e, which shows the graph $G(7;\pm2,\pm3)$. We obtain this graph when connecting nodes with mutual connections from graphs $G(7;\pm2)$ and $G(7;\pm3)$. Graph $G(7;\pm2,\pm3)$ can be obtained from $G(7;\pm2)$ and $G(7;\pm3)$ if we draw the first over the second. Inaccuracies regarding the before-mentioned definition are reflected in the next case. Graph $(8;\pm2,\pm3)$ (Figure 1h) is constructed using three cycles – two cycles of length 4 obtained from the graph $G(8;\pm2)$ and one cycle of length 8 obtained from the graph $G(8;\pm3)$. Numbers 8 and 2 for $G(8;\pm2)$ are not co prime, which is the reason why the cycles are reduced. If we consider the abovementioned statement, the definition can be corrected, and written in a new form. A two-circular graph $G(n;\pm h_1,\pm h_2)$ consisting of $gcd(n,h_1)$ cycles of length $n/gcd(n,h_1)$, and $gcd(n,h_2)$ cycles of length $n/gcd(n, h_2)$. For an ordinary k-circular graph $G(n;\pm h_1,\dots,\pm h_k)$ we cannot state, that it is constructed of k cycles of length n, but we can say, that each $i \in \{1,2,\dots,k\}$ contributes $gcd(n, h_i)$ cycles of length $n/gcd(n, h_i)$.

In addition to the number of nodes n and connections' length h_i for $i \in \{1, 2,\dots, k\}$, the connection type is also very important. We distinguish between oriented and non-oriented connections. Considering the mentioned connections we can distinguish between oriented, non-oriented and part-oriented circular-graph types. The definitions for the first two types are evident from their names while, the third one has some connections oriented, while others are non-oriented. The individual node degrees (the number of connections a node has to the other nodes) are equal in all three cases, the difference only being in the function of an individual connection, which performs certain tasks in a circular graph. An oriented *k*-cycle graph has *k* input and *k* output connections for each node; meanwhile a non-oriented graph has *2k* input and *2k* output connections for each node. An oriented *k*-loop graph can be denoted as $G(n;h_1,\ldots,h_k)$, and a non-oriented as $G(n;\pm h_1, \ldots,\pm h_k)$ [35].

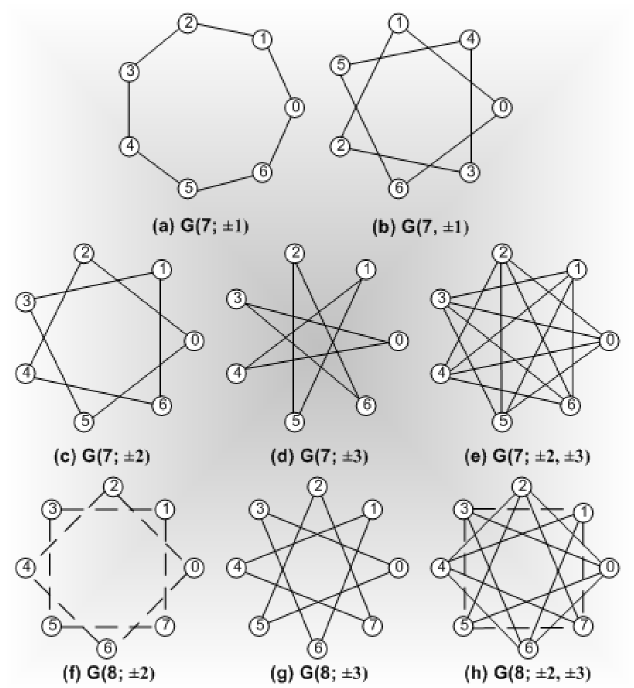

Fig. 1. Simplest circular graph examples G(7;±1), G(7;±2), G(7;±3), G(7;±2,±3), G(8;±3), G(8;±2, ±3)

Circular graphs' usage

A good network topology should have a small diameter, a small average distance between nodes, small nodes' degrees, and few links, which must be reliable, expandable, and symmetrical. Some of these listed properties mutually exclude each other, and an ideal topology actually does not exist. The topology that has certain advantages in some abilities has disadvantages in others. For example; a ring with oriented connections has a simple structure and can be easily implemented, but from other aspects it has a large diameter and is very unreliable, because the graph becomes disconnected if one connection or node falls out. A fully connected graph is the opposite in comparison to the ring. It is very reliable, but has many links and high individual node degrees. Fully connected topology implementations require complicated hardware equipment, especially in networks with a lot of nodes. In practice these topologies are rarely used. Circular graphs' topologies are placed somewhere between both mentioned extremes. They have small diameters, high reliability, they are expandable and symmetrical. For a node of degree 4, the graph's diameter is defined by equation $O\sqrt{n}$, where "*O*" represents the order of magnitude. Because of these ideal properties, circular graphs are often in use for modeling local and wide-area networks, and for linking schemes for processing units of multiprocessor systems. Nowadays, circular graphs are the best solution for FDDI-tokens, SILK and SONET/SDH rings, and in different parallel processing systems. Topological properties and properties connected with routing have been one of the main research subjects in the last few decades, but to-date there are still some questions unanswered. One of them is the question of time complexity for diameter calculation when *k* is bigger than 2.

Circular graph properties

Symmetry: The node symmetry [18] is a main circular graphs' property, which can be formally defined with the help of graph automorphism [3].

Definition (1): Let *A* be the bijective mapping of nodes set *V* of the *G(V,E)*. Mapping *A* represents the graph's *G(V,E)* automorphism, if for any nodes pair $u,v \in V$, for which $(u,v) \in E$, $(A(u),A(v)) \in E$ is true.

Graph's *G(V,E)* automorphism is each bijective mapping (permutation) on a set *V* which keeps the neighborhoods.

Example: For each $i \in \{0, 1,\ldots, n-1\}$ mapping $A_i:V \rightarrow V$, is defined as $A_i(v)=v+i$ (mod *n*), automorphism of the circular graph $G=G(n;h_1,\ldots, h_k)$, where *v+i* (mod *n*) means summing up module *n*. If nodes *u* and *v* in G are neighbors, then in G nodes *u+i* (mod *n*) and *v+i* (mod *n*) are also neighbors because from *u=v+h* (mod *n*) directly follows *u+i=v+i+h* (mod *n*). The modular arithmetic is described in [7].

Definition (2): Graph *G=G(V,E)* is vertex transitive, if for each node pair $u,v \in V$, such graph automorphism G exists that maps node *u* into a node *v*.

Theorem (1): Circular graphs are vertex transitive.
Proof (1): Let *u* and *v* be arbitrary nodes of a given circular graph $G(n;h_1,\ldots,h_k)$. If we mark *ω* with *ω=v−u* (mod *n*), where *ω=v−u* (mod *n*) represents subtraction according to module *n* [19], then mapping $A_\omega:x \rightarrow x+\omega$ (mod *n*) is automorphism and $A_\omega(u)=v$ is true.

Consequence (1): Let *u* and *v* be arbitrary points of the specific circular graph and let $e = \xi_{i1}\ \xi_{i2}\ \ldots\ \xi_{il}$ be the connection type sequence, which connects points *u* and *v*. Then we can say that *e* links points *0* and *ω = v − u (mod n)* lying on this graph.

Proof (2): Automorphism *A : x → x − u (mod n)* translates the node *u* into *0,* and the node *v* into *ω*. Such connection sequence is on the original graph presented as a link between *u* and *v*.

Following this consequence, the problem of searching for the shortest paths between all pairs, is simplified and translated into the problem of searching for the shortest paths between point *0* and all other available points on the circular graph. The graph separation symmetry could be defined similarly as the node symmetry. Such symmetry could be defined using the following approach: graph *G* is separation symmetric (arc-transitive) if for each pair of neighboring pairs *(u, v)* and *(x, y)* an automorphism which translates a point *u* into a point *x*, and point *v* into a point *y* exists. We have to know that circular graphs which are arc-transitive exist (for example, *G(8; ±1,±3))* but, in general, this is untrue.

Theorem (2): Circular graphs are generally not arc-transitive.

In order to prove this, we have to look at the *G(8;±1,±2)* graph. The link, connecting points *1* and *2,* lies on a cycle of length 8, while the link connecting points *1* and *3* lies on a cycle of length 4. Automorphism of graph *G*, for example, of the first connection, cannot translate it into another connection, and because of this graph *G* cannot be arc-transitive.

Connectivity: A circular graph, defined by parameters $n,h_1,\ldots,h_k$, has a connection [20, 22] when, and only when, the numbers $n, h_1,\ldots, h_k$ don't have a common factor, respectively when $gcd(n, h_1, h_2,\ldots, h_k)=1$ is valid. This thesis is also valid for non-oriented circular graphs. If $g=gcd(n, h_1, h_2,\ldots, h_k)\neq 1$, then we can reach from node *0* only those nodes whose mark/sign is a multiple number of *g*. In the opposite case, only when $g=1$, such as numbers $a'_0,a'_1,\ldots,a'_k$, exist that worth $a'_0n+a'_1h_1+a'_2h_2+\ldots+ a'_kh_k=1$. Such numbers can be calculated with an extended Evklids' algorithm. If for each $i=1,\ldots,n$ with a_i denoting the remainder when dividing the number a'_i by *n*, so that $a_i= mod(a'_i\ n)$, then a_i are natural numbers for which the next thesis is valid:

(1) $$a_1h_1 + a_2h_2 + \ldots + a_kh_k \equiv 1 \pmod n$$

From this we can conclude, in regard to this connection, that node 0 is connected with node 1 and, consequently, with all other nodes on the graph. $Gcd(n,h_1,h_2,\ldots,h_k)=1$ represents the conditions for graph connection. Namely, we can ascertain that each circular graph is correctly connected regarding links and nodes (graph stays connected if we remove less than *k* links or nodes, respectively), and that with a little more effort each *k*-circular graph can, regarding connectedness, become *2k*-connected. Even each two-circular graph with more than 5 nodes upon nodes can be 4-connected.

Diameter: A circular graph's diameter [20] depends directly on the numbers n and $h_1,\ldots, h_k$ with an order of magnitude $O(n)$. Along those graphs with a diameter *n,* graphs with much smaller diameters can also be found. The precise diameter values for circular graphs are unknown. This is also true for non-oriented graphs and *k*-circular graphs. Estimating its lower boundary *lb(n)* is also common. These estimations depend on links-orientation and on the number *k*. In all cases, they are in the order of magnitude $\theta\left(n^{1/k}\right)$.

The two-circular graph family is also well-known, with a diameter defined as *lb(n)*, but for *k*-circular graphs this diameter definition still remains unknown. Let's take a look at *lb(n)* for everyone abovementioned case.

Oriented two-circular graphs: Lower level for diameter of the oriented two-circular graph is defined as $lb(n)=\left\lceil\sqrt{3n}\right\rceil-2$. Equation introduces an infinity-graphs family where diameters are equal to *lb(n)* [14].

Non-oriented two-circular graphs: The precise lower level *lb(n)* for the diameters of a non-oriented two-circular graph is defined as $\left\lceil\left(-1+\sqrt{2n-1}\right)/2\right\rceil$ [21], because:

(2) $diam(G(n;\pm lb(n),\pm(lb(n) + 1)))=lb(n);\ \forall n$

Besides those mentioned, there are also other different two-way circular graph families with diameters equal to *lb(n)*. If for the given *n* the expression $diam(G(n;\pm 1,\pm h))=lb(n)$ is valid for *h*, then we can say that *n* is a suboptimal number. A set of suboptimal numbers consists of infinite elements, but for this specific case suboptimal numbers characterization is unknown [22].

Non-oriented k-circular graphs: The lowest diameter level for this graphs' family is defined by equation $(k!n)^{\frac{1}{k}}-0.5(k+1)$. Considering this equation, in reality an infinity-graphs family exists, which can achieve this diameter [23]. We can prove that there is an infinite number of *n*, for which such numbers as $h_2,h_3,\ldots,h_k$ exists where the diameter of the graph $G(n;1,h_2,\ldots,h_k)$ is lower or at most equal to $(((k-1)!c)^{1/k-1}+1/c)n^{1/k}-(k+2)/2$, for any $c\in Z\ -\{0\}$. Standard polynomial algorithms can be used for diameter calculations in ordinary graphs. Such algorithms have exponential time complexity. For general examples (for any *n* and any non-fixed *k*), circular graph diameter calculation is an NP-hard problem [22]. An *O(log(n))* algorithm for diameter calculation can be found for fixed *k* (*k* = 2), but the complexities of these problems, namely when k > 3, remain an open question [35].

Searching paths within circular graph

Let's take a look at the walk *s = (7,0)(0,3)(3,6)(6,8)* which represents an issue point. The set pair *(2, 2)* describes a walk between points *7* and *8* in the graph *G(9;±2,±3)*, and is derived from equation (3) which states that *7+2*2+2*3* is equal to *8*, summing according to module 9. This means, that the walk starts at point *7* and contains two connections of length 2, two connections of length 3, and it ends at point *8*. A similar equation can be written down for a general case. *k*-set $(x_1, x_2,\ldots,x_k)$ defines an equivalent class of walks between points *u* and *v* of the circular graph $G(n;\pm h_1,\ldots,\pm h_k)$ only when the following is true:

(3) $$\underset{\text{End point}}{8} = \overset{\text{Start point}}{7} + \underset{\text{Connection type }\varepsilon 1}{2} + \overset{\text{Connection type }\varepsilon 2}{3} + \underset{\text{Connection type }\varepsilon 2}{3} + \overset{\text{Connection type }\varepsilon 1}{2} \pmod 9$$

(4) $$u + x_1h_1 + x_2h_2 + \ldots + x_kh_k \equiv v \pmod n$$

An equivalent class-searching procedure between given points *u* and *v* within a circular graph is searching for a solution to congruent equation (4). This equation gives a result (with any *u* and *v*) only when $gcd(n, h_1,\ldots,h_k) = 1$ is true; if such a condition is fulfilled, we obtain an infinite number of solutions. The precise solution procedure of equation (4) will be described further in the article. We will see that Euclid's algorithm can be used to search for a solution in *O(log n)*. The condition for circular graph connectedness is that numbers *n* and h_1, $i\in I^+$ do not have a common factor. A graph is connected only when there is an existing walk between any pair of its points, i.e. only when a solution of the equation (4) exists. From equation (3) it can be seen that each walk between points *u* and *v* is also a walk between points *0* and $\omega = v - u \pmod n$. This is because each *k*-set $(x_1,x_2\ldots,x_k)$ that is a solution of equation (4) is also a solution of equation $0+x_1h_1+x_2h_2+\ldots x_kh_k=v-u \pmod n$. When considering this statement, we could translate the problem of the walk searching into the problem of searching for all walks between point 0 and all other points in the circular graph.

Shortest path searching procedure: A path in the circular graph is defined as a walk $u_0\ u_1\ u_2\ldots u_l$, where each of these enumerated nodes appears only once [24]. It is very important to know among all possible paths as to find, which is the shortest and which has the minimal number of links.

Definition: In each path, the shortest path between nodes *u* and *v* is that which has the shortest length between *u* and *v*. More than one possible shortest path can exist between two nodes. If the path, which contains $|x_i|$ connections of type $\varepsilon_{i \cdot sign(xi)}$ ($\forall i \in I^+$) is the shortest path between *u* and *v*, then each path, which is a member of the equivalent class $[x_1, x_2, \ldots, x_k]$, is also a shorter one. We can prove this by the following formalism.

Theorem: Let *u* and *v* be arbitrary nodes of a *k*-circular graph and let *k*-set of integer numbers $(x_1, x_2, \ldots x_k)$ be such a solution from equation (3), so that for each other solution $(y_1, y_2, \ldots, y_k)$ of this equation $\sum_{i \in I} |x_i| \le \sum_{i \in I} |y_i|$ is valid. Then each walk from equivalent class $[x_1, x_2, \ldots, x_k]$ represents a path on a circular graph.

Proof: Suppose there exists such a walk $s = u_0\, u_1\, u_2 \ldots u_l$ of equivalent *class* $[x_1, x_2, \ldots, x_k]$, which is not a path. Let u_i be the point of a walk *s* which has a repetition, and let *s'* be the sub-walk of the walk *s*, which starts and ends at a point u_i. For all elements of *k*-set $(z_1, z_2, \ldots, z_k)$ which represents sub-walk *s'*, worth $|z_i| \le |x_i|$ and $sign(z_i) = sign(x_i)$.
For each $i \in I^+$ is valid $|x_i - z_i| \le |x_i|$. Because *s'* represents a cycle, worth $u_i + z_1 h_1 + \ldots + z_k h_k = u_i \ (mod\ n)$, respectively $z_1 h_1 + \ldots + z_k h_k = 0 \ (mod\ n)$. Because of that $(x - z)$ is equal to $(x_1 - z_1, x_2 - z_2, \ldots, x_k - z_k)$ and at the same time represents a solution to equation (4), and worth $\sum_{i \in I} |x_i - z_i| \le \sum_{i \in I} |x_i|$, but this is contrary to proposition.

Consequence: Let *u* and *v* be the given nodes of the circular graph $G(n; \pm h_1 \ldots, \pm h_k)$ and let the *k*-set of integer numbers be such a solution from equation (3) that for each other solution $(y_1, y_2, \ldots, y_k)$ of this equation $\sum_{i \in I} |x_i| \le \sum_{i \in I} |y_i|$ will be valid. Each path of the equivalent class $[x_1, x_2, \ldots, x]$ is the possible shortest one between nodes *u* and *v*. The shortest path searching procedure is equivalent to searching for such a solution using a congruent equation (3), which minimizes the sum between all solutions of this equation,

$$|x_1| + |x_2| + \ldots + |x_k|. \qquad (5)$$

The fact that more than one equivalent path with the same length exists can be well used in the routing protocol development procedure, where the shortest paths are used between start and destination nodes. Multiple equivalent possible shortest paths mean more possibilities for solving eventual problems, which can appear during routing procedures (collisions, link loss, path loss, router loss, etc.). Furthermore, sometimes more than one equivalent class with possible shortest paths between two nodes for a given circular graph exists [35].

Cyclic structure searching method in practice

The original idea of how to find a circular graph topology in a specific network is tightly connected with the existing routing algorithms' data where RIP [9], OSPF [25], BGP [25], IS-IS [26], RIP2 [9], etc. belong. For better understanding, we should first give a short RIP working procedure description. RIP belongs to dynamic routing protocols family, meaning that it is capable of accommodating to dynamic network changes (adding, removing network elements, etc.). It is useful for routing within LAN or even WAN network segments. It uses the so-called hops' counter, which prevents traffic cycling in the network. However, this is in contradiction to our concept, where we are searching for cyclic connections, which are the basic elements for circular graph construction. RIP every 30 seconds sends full updates of network topology through the network. Such updates are a basis for updating individual routes, and at the same time every network unit (router) knows all other interconnections with other units in the network. Different routing protocols update his tables in different manners. The most popular way is to send the 'hello messages' (OSPF, for example). In this case, the network unit (router) sends a hello message into the network, and then waits for possible replies, which arrive from all its reachable neighbours. Such principle is periodically repeated through the whole network.

At this point, we can discuss the basis of the proposed concept of CIGRP routing protocol algorithm, which must be executed in three basic steps.

In the first step, from the individual network unit routing table, irrespective of used main routing algorithm, we can construct the connectivity matrix. The CIGRP supervisor sub-algorithm performs the routing table analysis, where searching for cycles. When the main routing table is changed, the connectivity matrix is also changed. Compared original RIP algorithm and algorithm CIGRP operating mutual interoperable with RIP, the CIGRP supervisor algorithm must be executed before the RIP's original 'reversal poison horizon' rule is executed. In the connectivity matrix, the algorithm searches for circular graphs as well their properties (connectivity, symmetry, etc.); if cycles do not exist or none of the listed cyclic structure properties are not fulfilled, then the algorithm returns a classical RIP's spanning tree graph [8].

In the second step, the CIGRP supervisor logic determines from the connectivity matrix in which regime the router will operate. When the supervisor logic finds the cycles and the definitions for cyclic structure is fulfilled, the router must inform all its neighbours of the temporary routing regime (for example, CIGRP mode). In opposite case, when cycles are not found in the connectivity matrix, the router must operate in one of the existing regimes (for example in RIP regime). From this aspect, the information on the operating regime is enough for the CIGRP operation mode, and must be forwarded to all neighbours. Routing tables are in RIP case refreshed every 30 seconds, meaning that the supervisor logic will also update the connectivity matrix on each router. Such matrix defines a graph topology. In cases, where there are no topology changes within the network, connectivity matrix and main routing table stay unchanged together with the routing regime.

In the third step, the CIGRP searches for all possible paths to a specific destination given by the temporary traffic (see section for searching shortest path). The algorithm must then calculate the shortest path of all possible paths, stored in the PP-Set (Possible Paths Set), where the CIGRP uses the mathematical procedure described in section (*Shortest path searching procedure*).

CIGRP must always operate with another routing protocol mechanism and this is also the main point of the proposed concept. Mutual operation is very important in cases where the circular topology does not exist. From this aspect, CIGRP presents only a supplement to other routing protocols. No matter which main routing protocol is combined with presented CIGRP concept, the main routing table of the main routing protocol is a common and fundamental part for both. Before routing can start, the routers must exchange the messages about the routing regime (CIGRP or RIP for example).

For easier understanding of how such concept works, we present the methodology of the first and the second step, described above, with two practical examples.

Example 1. CIGRP in the first step creates a local copy of the main routing table, before the RIP's redundancy removing algorithm is performed. In this case, we keep all possible connections. The CIGRP's algorithm for connectivity matrix creation can be then executed, as shown in Table 1.

Table 1. Connectivity matrix

	Node A	Node B	Node C	Node D
Node A	0	1	0	1
Node B	1	0	1	0
Node C	0	1	0	1
Node D	1	0	1	0

With the help of the connectivity matrix, we can construct the circular graph virtual scheme, which presents the order of units' connectivity, as presented in Figure 2. In Figure 2 we can see that unit 'A' has a connection with unit 'B' and vice versa, the 'B' has a connection with 'C' and vice versa, etc. In the connectivity matrix presented in Table 1 the circular graph G(4, ±1) exists. For this cyclic structure, all definitions are fulfilled, because the presented circular graph contains one cycle, and is at the same time symmetrical and fully connected. The procedure of calculating the shortest path can start at this point.

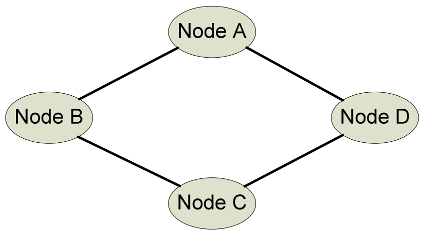

Fig. 2. Circular graph topology G(4, ±1) obtained from connectivity matrix

Example 2. For the circular graph structure G(7, ±3) (Figure 1) the connectivity matrix takes the form presented in Table 2. The connectivity matrix is obtained from the main routing table.

Table 2. Connectivity matrix for circular graph G(7, ±3)

Node	A	B	C	D	E	F	G
A	0	0	0	1	1	0	0
B	0	0	0	0	1	1	0
C	0	0	0	0	0	1	1
D	1	0	0	0	0	0	1
E	1	1	0	0	0	0	0
F	0	1	1	0	0	0	0
G	0	1	1	1	0	0	0

Proposal of CIGRP integration into the RIP protocol

At this point, we should answer a few questions, which might have occurred to the reader. For example: how does the RIP remove the cycles, how does the RIP calculate the fastest (not shortest as with CIGRP) path, etc. The answers can give a constructive comparison between the RIP protocol and the CIGRP concept.

How does the RIP eliminate the cycles?

The RIP protocol eliminates the cyclic connections which present the fundamental part of the CIGRP, as is described in the theoretical part in the beginning. Removing redundant connections from the RIP routing table is ensured by the supervisor RIP sub-algorithm. If from one source more than one way leads to a specific destination, the sub-algorithm compares the metrics of possible paths, and decides which path must be kept in the routing table and which must be removed (the CIGRP keeps all possible paths). The path with the better metrics must be kept. Such an arrangement prevents cycles and redundant connections within the network. Nevertheless, additional rules are also implemented into the RIP algorithm to prevent cycles. If a large part of the network falls out, routing tables are not updated fast enough and the metric counter is not effective, packets can start a cycling procedure within the network (undesirable scenario). Such scenario is prevented with the rules *split horizon* and *split poison horizon reversed* [27], so that packets cannot start the cycling procedure. In many cases, the rule *split poison horizon reversed* is in use, its main weakness being an increase of the update table sizes. Also, all paths through a specific node must be deleted if new messages do not arrive at its destination at a specific time (timeout).

How RIP calculates the fastest (not shortest) path?

RIP protocol fastest path calculation procedure is based on the distance vector algorithm. The distance vector contains information about direction, distance, and time estimation needed to transfer the packets to the specific destination node. Each node contains the distance-vectors table, which is continuously updated on each node. These tables are shared between all neighbors, and because the distance-vectors table is a sub-part of the main routing table the latter is also refreshed and updated.

RIP and CIGRP-RIP modified protocol comparison

Figure 3 presents the main mutual concept of routing regimes. RIP protocol and the proposed CIGRP routing protocol regime are described in previous section.

Figure 3 presents the main idea about how the CIGRP segment should be implemented into an existing RIP protocol. It would be able to use all the information on nodes' connectivity from the RIP routing table before the procedure of removing redundant connections starts. The dashed bordered frames present the main blocks of the proposed CIGRP algorithm, which are nested between RIP's (shading frames) main blocks. *Routing Table Manager* block belongs to RIP, and presents the basis for CIGRP implementation. All presented steps (first step, second step, and third step) are described in detail in previous section. The main difference between both methods is in the beginning phase, where the RIP main table local copy procedure starts before RIP rules for removing redundant connections executes their algorithms (this happens before the rule *split poison horizon reversed*). Such modification ensures that the local routing table copy keeps the cyclic connections, which are the basic part of CIGRP. Previous section describes how such mutual concept works. The small RIP protocol modification for mutual operation is needed only in the beginning of the first step and at the end of second step, where RIP must wait for CIGRP cyclic structure analysis. Waiting is needed, so that CIGRP supervisor logic can decide, which routing table will be forwarded to their neighbors and, which routing protocol will be in use at the specific moment.

We can notice that CIGRP concept contains a few steps more in comparison to RIP, but nowadays high performance processing devices are able to execute one or two simpler algorithms so fast that this will not present an obstacle in the future. The most important condition for mutual operation is the supervisor CIGRP logic which defines the operating regime (end of the second step).

The CIGRP shortest path searching concept is presented in section for shortest path calculation. With CIGRP introduction, we want to provide the possibility for each node to occupy the shortest path leading to a specific destination (RIP always sets the fastest path). At the same time, we want to keep the other routing protocols properties compatible with the CIGRP concept, such as:

- Main routing table creation based on routing tables' information exchange with other neighbor routers,
- reply capability to demands for updating routing tables,
- reliability, and
- capability of detecting network setting changes.

At this point, a very important question is how to prevent packets from cycling in the network, considering that we need them for CIGRP. Here many solutions are possible; we shall present two of them bellow, which should be enough for testing proposes.

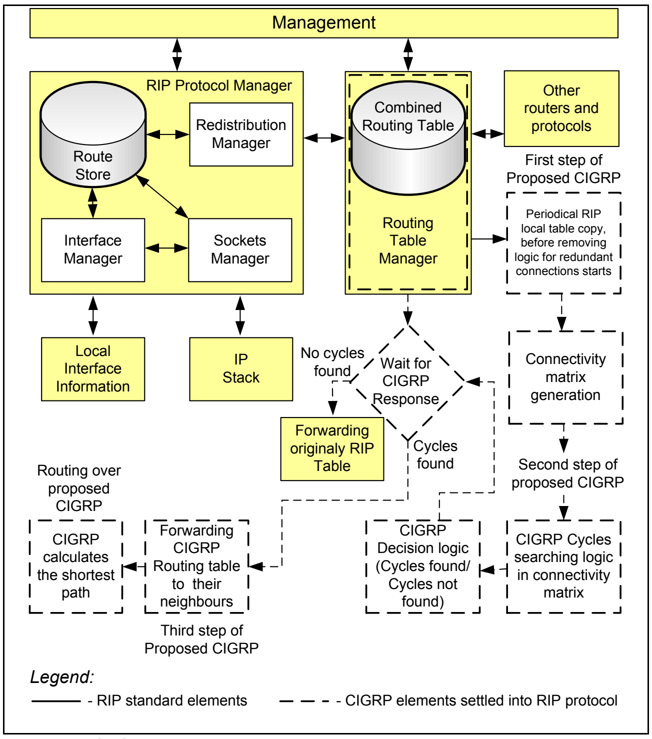

Fig. 3. CIGRP and RIP routing protocol mutual working scenario and RIP upgrading concept with CIGRP modules

Method 1. This is a less suitable solution, where an additional frame is added into the packet header, into which a unique identification number of each node is entered, when a packet goes through the router. The algorithm for packet mediation is responsible for header checking and comparison of node identification numbers entered into the new header. If the algorithm determines that temporary packet header already contains the passing router id number, it must immediately reject and destroy the packet. Such an approach will be useful only in cases where the traffic amount is small, because each record in the new header increases the packet size, and an additional header also means a higher packet length. That is contradicting to our thesis (as fast as possible).

Method 2. This is the more suitable solution, where the segment of the interior router memory is used for temporary packet identification numbers' storage. Each node temporarily remembers the identification of the packets the go through specific nodes. How long such identifications are stored is defined by parameter TTL. Packets that return to an already visited location must be immediately destroyed by a supervisor algorithm. This approach also has disadvantages (relatively big size memory reservation), but for first phase it is a good approach for researching and testing purposes.

Conclusion

With the introduction of a mathematical model, we have to provide the reader with an alternative solution which can be used in the routing procedure when searching for the shortest path within the network. This paper also presents the principle of how transfer paths can be presented by a mathematical model when network topology is translated into circular graph topology. With the help of circular graphs' mathematical laws, we can calculate a graph diameter, shortest paths between given nodes, and we are as well able to find possible walks through a circular graph, which are equivalent to paths in a real network. As we have mentioned in the beginning, existing methods [28] are not ideal and our method also has advantages, disadvantages and limitations. Our intention is to show the reader one of the possible methods when searching better paths within and between networks, especially in cases when we have to deal with time-sensitive and real-time applications.

At this point, it should be mentioned that the presented concept is only the main idea which has not yet been implemented and must first be implemented into a simulation tool, OPNET Modeler [29, 30, and 31], for example, then tested, estimated and evaluated over simulations. After implementation of the presented solution into a simulation environment we will be able to better say whether such concept is successful or not. For now a mathematical proof is enough to start researching in this direction.

The main aim of future research work is testing the proposed new routing concept (CIGRP) using simulation tool, such as OPNET Modeler. We have to find the proper solution comparing Method 1 and Method 2 described in above section. Simulation results will tell us if circular graph theory is useful on way, described in this paper.

REFERENCES

[1] Cisco Systems, "Internetworking Technologies Handbook – Routing Basics", Chapter 5 http://www.cisco.com/en/US/docs/internetworking/technology/handbook/Routing-Basics.pdf

[2] Janja Jerebic, Sandi Klavzar, "The strong isometric dimension of graphs of diameter two", April 7, 2003 http://www.imfm.si/preprinti/PDF/00876.pdf

[3] P. Potocnik, "Delovanje grup in simetrije grafov : doktorska disertacija", Univerza v Ljubljani, Fakulteta za matematiko in fiziko.

[4] Yuan-Shin Hwang, Joel Saltz, "Identifying Parallelism in Programs with Cyclic Graphs", Parallel Processing, 2000. Proceedings. 2000 International Conference, 21-24 Aug. 2000, Page(s):201 – 208

[5] IMRICH Wilfried, KLAVZAR Sandi, "On the complexity of recognizing Hamming graphs and related classes of graphs", *Eur. j. comb.*, 1996, let. 17, vol. 2/3, Page(s): 209-221

[6] KLAVZAR Sandi, PETKOVSEK Marko, "Graphs with nonempty intersection of longest paths", Ars comb., 1990, vol. 29, Page(s): 43-52

[7] Spanning tree graph theory, http://www.personal.kent.edu/~rmuhamma/GraphTheory/MyGraphTheory/trees.htm

[8] Karl Benkic, Marko Malajner, Peulic Aleksandar, Zarko Cucej, "Academic education Wireless Sensor Network: AeWSN", *50th International Symposium ELMAR-2008, 10-12 September 2008, Zadar, Croatia*

[9] Cisco Systems, "Internetworking Technologies Handbook – Routing Information Protocol", Chapter 47 http://users.freenet.am/~file/DownDB/CISCO_PDF/RoutingInformationProtocol_CISCO.pdf

[10] Wikipedia, “Bellman-Ford algorithm”, http://en.wikipedia.org/wiki/Bellman-Ford_algorithm

[11] Charles E. Perkins, Elizabeth M. Royer, “Ad-hoc On-Demand Distance Vector Routing”, Second IEEE Workshop on Mobile Computer Systems and Applications, New Orleans, Louisiana

[12] Permutation routing in hypercubic networks, Topics in parallel computing, CS1221, Feb 25, 1999 http://pages.cs.wisc.edu/~tvrdik/10/html/Section10.html

[13] Wikipedia, “Bijection”, *http*://en.wikipedia.org/wiki/Bijection

[14] J. Zerovnik and T. Pisanski (1993), “Computing the diameter in multiple-loop networks”, J. Algorithms, 14:226–243.

[15] Ming T. Liu, Sandra A. Mamrak, Jayshree Ramanathan, “The Distributed Double-Loop Computer Network (DDLCN)”, ACM Annual Conference/Annual Meeting, Proceedings of the ACM 1980 annual conference, Pages: 164 – 178, 1980

[16] G. Babic and T. L. Ming, “A performance study of the distributed loop computer network (DCLN)”, in Proc. Computer, Networking Symp., (National Bureau of Standards, Gaithersburg, MD, December 15, 1977), Pages: 66-76

[17] X. D. Hu, F. K. Hwang, “Cutting numbers for the forward loop backward hop network”, Discrete Applied Mathematics, Volume 48, Issue 2 (January 1994), Pages: 147 – 161

[18] ZEROVNIK Janez, “Szeged index of symmetric graphs”, J. chem. Inf. comput. sci., 1999, issue. 39, vol.1, Page(s): 77-80

[19] Wikipedia, “Modular arithmetic”, http://en.wikipedia.org/wiki/Modular_arithmetic

[20] D. S. Hirschberg and C. K. Wong, “Upper and lower bounds for graph-diameter problems with application to record allocation”, Journal of Combinatorial Theory, Series B, Volume 26, Issue 1

[21] F.K. Hwang (2003), “A survey on multi-loop networks”, Theoretical Computer Science, 299:107–121.

[22] J.Y. Cai, G. Havas, B. Mans, A. Nerurkar, J.P. Seifert, and I. Shparlinski (1999), “On routing in circulant graphs”, Proceedings of the International Conference Computing and Combinatorics, *COCOON’99, LNCS 1627*, str. 360–369, Tokyo, Japan, July 26–28.

[23] J.C. Bermond, F. Comellas, and D.F. Hsu (1995), “Distributed loop computer networks: A survey”, Journal of Parallel and Distributed Computing, 24:2–10.

[24] R.J. Wilson and J.J. Watkins (1997), “Uvod v teorijo grafov”, DMFA, Ljubljana.

[25] Cisco Systems, “Routing Protocols (BGP, OSPF), Cisco Active Network Abstraction 3.6.6 Technology Support and Information Model Reference Manual”, Chapter 3

[26] Sampo Saaristo, “Implementation of IS-IS Routing Protocol for IP versions 4 and 6”, Master of Science Thesis, Tampere University of Technology, Institute of Communications Engineering, April 2002 http://www.atm.tut.fi/faster/ipv6/isis_impl.pdf

[27] C. Hedrick, “RFC1058 - Routing Information Protocol”, Rutgers University, http://www.faqs.org/rfcs/rfc1058.html

[28] H. Jonathan Chao and Bin Liu (2007), “High Performance Switches and Routers”, John Wiley & Sons.

[29] Klampfer Sasa, “Networks simmulation in OPNET Modeler”, diploma degree thesis, University of Maribor, September 2007

[30] Klampfer Sasa, “Expert system for automatic analysis of tactical radio network properties”, master degree thesis, University of Maribor, June 2009

[31] J. Mohorko, M. Fras, “Mathematical model of IRIS replication mechanism for the simulation of tactical networks”, Computer Networks, May 2009, Vol. 53, Issue. 7, Pages: 1125- 1136

[32] Wikipedia, “NP-Hard problem”, http://en.wikipedia.org/wiki/NP-hard

[33] Sasa Klampfer, Joze Mohorko, Zarko Cucej, “Mathematical analysis of routing, based on circular graphs”, Software, Telecommunications and Computer Networks, 2008. SoftCOM 2008. 16th International Conference, 25-27 Sept. 2008 Page(s):389 – 393

[34] Shiou-Wen Lu and Ling-Yang Kung, “Cluster model for optimal double-loop connected network”, TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering, Issue 0, Part 30000, 19-21 Oct. 1993 Page(s):162 - 165 vol.3

[35] Tomaz Dobravec, Usmerjevalni algoritmi v omrezjih s topologijo kroznih grafov, Doktorska naloga, Fakulteta za racunalnistvo in informatyko, Univerza v Ljubljani, 2004

***Authors**: Msc. Sasa Klampfer, Margento Research & Development d.o.o., Gosposvetska cesta 84, 2000 Maribor, Slovenia, E-mail: sasa.klampfer@margento.com*