Journal of Information Technology and Engineering
Vol. 2 No. 2 (December, 2017)

Advanced Routing Algorithms to Reduce Time Complexity in Mobile Corporative Network

Khalil AL-SHQEERAT
Department of Computer Science, Zarqa Private University, Zarqa 13110, Jordan
E-mail:kha_2000@hotmail.com

Received: 22nd March 2017   Revised: 14th July 2017   Accepted: 20th October 2017

Abstract: *The rise of the wireless communication paired with the rapid developments in networking technology where users need mobility. The included analysis of known routing algorithms aims to evaluate their performance in mobile networks. The suggested routing algorithm offered subsequent rerouting with minimum time complexity. This is achieved by creating tree for the shortest paths, which takes into account subscriber movement probability, allows minimizing number of "returns" in the delivery tree structure and thus reduces rerouting time, and ensures minimum complexity of rerouting during subscriber movement. Our algorithm is designed to perform the rerouting of user connections in fast and efficient manner and can be a new technique that speeds up routing, performs Quality of Service and provides a great solution for traffic engineering by controlling traffic flows in the network. Our proposed algorithm can be used for both single-address and multiple-address routing.*

Keywords: mobile network, routing algorithms, Dijkstra algorithm**.**

1. INTRODUCTION

Corporate network means computer network of large industrial associations, including banks, separate departments which are located on different distances from each other [1]. In virtue of specificity of industrial activities of large industrial associations, diversified and vast scope of functions is imposed on corporate network. As a result of this modern corporate networks are supposed to rely on both batch processing and dialog (interactive) application. Aiming to support different applications, the network is to include efficient program and technical means, including different channels transmission rates, interfaces etc.

Mobile subscriber systems are widely spread in modern corporate networks. It determines additional requirements and peculiarities of corporate networks building. Primarily it leads to complication of corporate networks forming procedure, which can be changed dynamically in the process of network functioning. Also routing procedures are becoming complicated.

Mobile networks involve mobile hosts and an underlying wired network consisting of base stations and intermediate routers. The mobile hosts move from one cell to another and register with the base station in-charge of the cell when they move into the cell (Figure 1). This process of transferring the control and responsibility for maintaining communication connectivity is called handoff [2]. The process of routing following the occurred handoff is called Rerouting [4, 5, 6, 7].

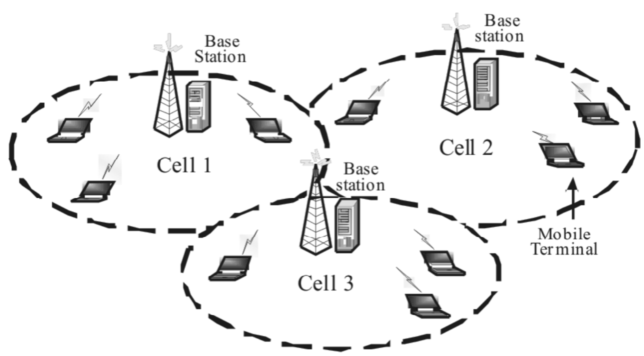


Figure 1

The information about subscriber system ownership to a cell is maintained in an access server that stores information about subscriber systems of one or more cells. In the first case, as a rule, access server function is served by the base station. In the second case a separate computer represents an access server. Thus the cells serviced by one access server form a cluster [8].

When relocating subscriber system from one cell into another cell, respective adjustment of the subscriber systems location tables is carried out in access servers. In wireless computer networks routing process is divided into intracluster routing and intercluster routing. In the first case the access server routes subscriber systems located inside a cluster. In the second case routing is carried out at the access server's level. For this purpose, each access server also contains information necessary for the routing subscriber systems located in other clusters.

2. BACKGROUND AND RELATED WORK

We will study routing algorithms with technique that is widely used in many forms because it is simple and easy to understand [3]. The idea is to build a graph of the subnet $G = (V, E)$, (where $V = \{v_i \mid i = 1,2,...,n\}$–the set of nodes (vertices) of the graph, $E = \{e_{i,j} \mid i = 1,2,..., n; j = 1,2,..., m\}$ –the set of edges (arcs) of the graph) with each node of the graph representing a router and each arc of the graph representing a communication line.

Each graph's edge $e_{i,j}$ is characterized by the weight $w_{i,j}$ which means transmission delay $t_{i,j}$ between network nodes v_i and v_j. In this case admissible value of transmission delay t_d between arbitrary networks nodes is introduced as restrictions, and parameter optimization is considered to be a total delay $T_{s,r}$ in the route $L_{s,r}$. In this case routing task is to find a route $L(v_s,v_r) = (e_{s,i}, ..., e_{j,r})$ between vertexes v_s and v_r for which the condition is being completed:

$$T_{s,r} = \min \sum_{e_{i,j} \in L_{s,r}} t_{i,j} \tag{1}$$

where: $T_{s,r}$–total delay on the path $L(v_s,v_r)$.

The following condition is considered as restrictions when inclusion edge $e_{i,j}$ in the path $L_{s,r}$: $C(i, j) > c_s$, where: $C(i, j)$–bandwidth of channel; c_s– required bandwidth for messages delivery on the path $L_{s,r}$. $t_d > t_m$, where: t_d– admissible delay for transmission packet; t_m–maximum delay of packet transfer between tree nodes. To choose a route between a given pair of the routers, the algorithm just finds the shortest bath between them on the graph. Several algorithms for computing the shortest path between two nodes of a graph are known [13]. This one is due to Dijkstra (1959). Each node is labeled (in parentheses) with its distance from the source node along the best known path. Initially, no path are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible bath from the source to the destination node, it is made permanent and never changed thereafter.

There is introduced a comparative evaluation of time complexity for routing algorithm in the work [14]. It is shown that Dijkstra algorithm possesses the smallest time complexity upon unloaded graphs.

The rerouting in this case involves establishing paths through the fixed route. The disadvantage of this scheme is that if the mobile hosts keep moving, the route used for communication may be inefficient as the scheme does not dynamically update the routes based on the location of the mobile hosts [2].

3. VIRTUAL STRUCTURE OF CORPORATIVE NETWORK

In accordance with bi-level structure of corporate network, graph $G = (V, E)$ can be presented in the form of graph $G_0 = (V_0, E_0)$, determining backbone sub network structure, and some variety of sub graphs $\{G_i = (V_i, E_i) | i = 1,…,n\}$, relevant to local sub networks of corporate networks. In general, for the vertex $v_s \in V_1$ of sub graph $G_1 = (V_1,E_1)$ and the vertex $v_r \in V_2$ of sub graph $G_2 = (V_2, E_2)$, and also extreme vertexes $v_k \in V_1$ ($\exists\, e_{k,i}$, $v_i \in V_0$) and $v_m \in V_2$ ($\exists\, e_{m,j}$, $v_j \in V_0$), route $L(v_s,v_r)$ is determined by the expression: $L(v_s,v_r) = L_1(v_s,v_k) + L_0(v_k,v_m) + L_2(v_m,v_r)$. Accordingly, the total delay $T_{s,r}$ of the path $L(v_s,v_r)$ is:

$$T_{s,r} = \sum_{e_{x,y} \in L_1(v_s,v_k)} t_{x,y} + \sum_{e_{i,j} \in L_0(v_k,v_m)} t_{i,j} + \sum_{e_{z,q} L_2(v_m,v_r)} t_{z,q} \tag{2}$$

Taking to the account that data transmission rate in local sub networks is higher than data transmission rate in global networks, as a rule.

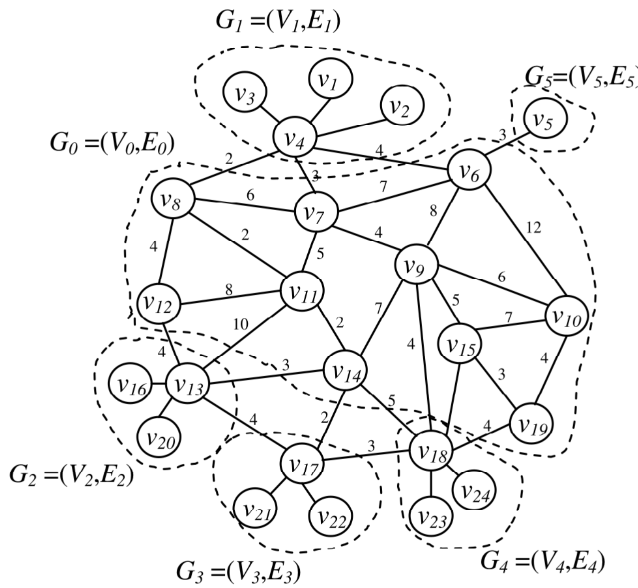


Figure 2

If to locate a source and an addressee in different local networks on considerable distance one from another, in this case $T_{s,r} \approx T_{k,m}$ and, accordingly:

$$T_{s,r} \approx \sum_{e_{i,j} \in L_{k,m}} t_{i,j} \tag{3}$$

Therefore, in this certain case main attention should be given to optimal path construction problem between extreme vertexes on the graph $G_0 = (V_0, E_0)$. Lets study a case in which AC_1 is located constantly about the vertex v_4, and mobile AC_2 is shifted along the path.

As it is shown in Figure 3, in accordance with edge weight paths are forming between extreme vertex v_4 of sub graph $G_1(V_1, E_1)$ and extreme vertexes v_{13}, v_{17} and v_{18}, accordingly sub graphs $G_2(V_2, E_2)$, $G_3(V_3, E_3)$ и $G_4(V_4, E_4)$.

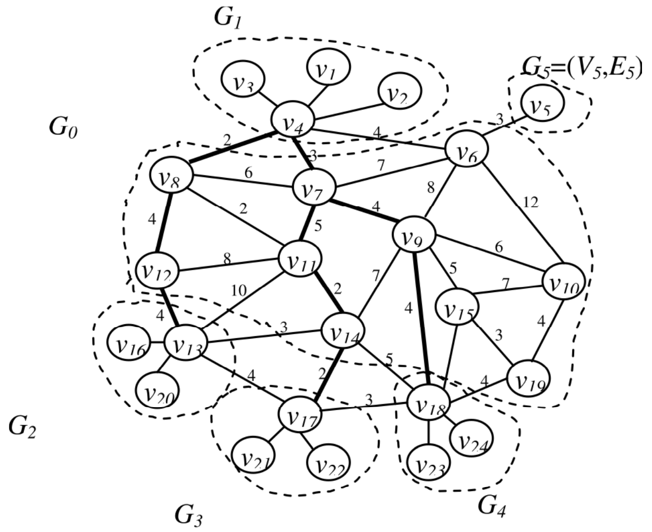


Figure 3

4. BASIC IDEA OF THE WORK

Small time complexity of rerouting procedure is considered as a basic requirement to the mobile network routing algorithm since paths calculation should be produced in real time. Routing algorithm should be simple enough and dynamic. With the purpose of reducing rerouting time, generation of feasible solutions tree representing a minimal covering tree is used instead of virtual paths of the information delivery route. For that we present data-transmission network as a graph.

This is assumed as a basis of the developed routing algorithm which creates a tree so that rerouting complexity is minimum during subscriber movement from one vertex to another, i.e. the total cost of the old (disconnected) and new (active) branch is minimum at movement from one branch of the delivery tree to another. Thus, the number of "returns" in the delivery tree structure is minimized at rerouting. Let's assume two methods of rerouting. The first method is to prolong the route; the second one is route reconfiguration. In the first case a time complexity of rerouting is minimal, but a threat of excessive prolongation and possibility for loops appearance in it is appeared. Path reconfiguration is characterized by the greater time complexity, but allows generating an optimum path.

4.1 The Path Prolongation

Let cover methods of paths forming at the expense of its prolongation. Lets consider that AC_2 is shifted along vertex $v_{13}- v_{17}- v_{18} -v_{19}-v_{10}$. So, in process of shifting AC_2 path $L_0(v_{13}, v_{10}) = 15$ will be forming. In this case compound path $L_1(v_4, v_{10}) = L_1(v_4, v_{13}) + L_0(v_{13}, v_{10})$ equals 25, and the shortest path $L_4(v_4, v_{10}) = 16$. At the same time using the path $L_3(v_4, v_{18})$ (Fig. 4) metrics to the vertex v_{13} of the path $L_3(v_4, v_{13}) = L_3(v_4, v_{18}) + L_0(v_{18}, v_{13})$ will equal 18, and to the uttermost vertex v_{10} of the path $L_3(v_4, v_{10}) = L_3(v_4, v_{18}) + L_0(v_{18}, v_{10})$ equals 19. It is far less than path length $L_1(v_4, v_{10}) = L_1(v_4, v_{13}) + L_0(v_{13}, v_{10})$.

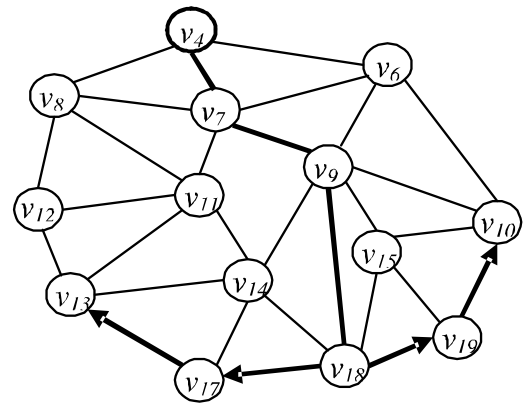


Figure 4

Thus, path forming in this case might result to location of such vertex v_i in the path $L_0(v_{13}, v_{10})$ in which the distance from the vertex v_4 to the uttermost vertex of the path $L_0(v_{13}, v_{10})$ is minimal.

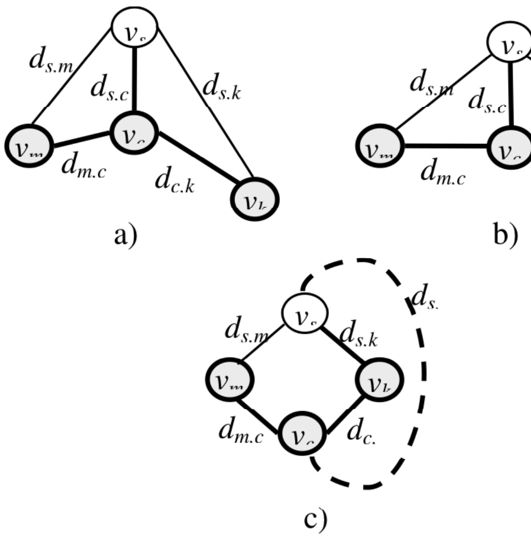
Generally task solution might be done by searching all possible variants; however such approach is not efficient if the amount of nods is rather big. In the context of the current work it is proposed rather quick algorithm directed to variant searching. For this reason we will define a task generally: On loaded graph $G(V, E)$ with set of edge's lengths $W=\{d_{i,j} | e_{i,j} \in E\}$ it is given a source vertex v_s and linear subgraph $G_0 = (V_0, E_0)$ with initial vertex v_m, terminal vertex v_k and set of edge's lengths $W_0=\{d_{i,j} | e_{i,j} \in E_0\}$. Subgraph $G_0 = (V_0, E_0)$ determine unambiguously the minimal path $L_0(v_m, v_k)$, length D_0 of which equals:

$$D_0 = \sum_{e_{i,j} \in E} d_{i,j} \qquad (4)$$

Then we will determine the vertex $v_c \in V_0$ in which the path $L_c(v_s, v_c)$ comes to the end. There is a graph represented in the Figure 4, source vertex of which is a vertex v_4, a set of vertexes $V_0=\{v_{13}, v_{17}, v_{18}, v_{19}, v_{10}\}$. The initial vertex is vertex v_{13}, the terminal vertex is v_{10}. The vertex of path connection is vertex v_{18}. In general this task might be formed in the following manner: To determine the vertex $v_c \in V_0$ the magnitude of which $D_c=D(v_s, v_c) + max\ (D(v_s, v_c), D(v_s, v_c))$ is minimal among all magnitudes $(D_i | v_i \in V_0)$.

Let's look at geometrics solution of this task:

As it is shown in the Figure 5 upon different lateral length relationships when deposit vertex v_c in the middle of the path $L_0(v_m, v_k)$ magnitude $D_c=D(v_s, v_c) + max\ (D(v_s, v_c), D(v_s, v_c))$ will be minimal among all magnitudes $(D_i | v_i \in V_0)$. As shown in the Figure 5a and 5b the magnitudes $D_c = d_{s,c} + d_{c,k}$, $D_m = d_{s,m} + d_{m,c} + d_{c,k}$ and $D_k = d_{s,k} + d_{c,k} + d_{m,c}$ are presented. Let's compare magnitude D_c with D_m. Since by definition $d_{s,c}$ is the shortest path between vertexes v_s and v_c, so the



a) $D(v_s,v_m) > D(v_s,v_c)$ и $D(v_s,v_k) > D(v_s,v_c)$.

b) $D(v_s,v_m) > D(v_s,v_c)$ и $D(v_s,v_k) < D(v_s,v_c)$.

c) $D(v_s,v_m) < D(v_s,v_c)$ и $D(v_s,v_k) < D(v_s,v_c)$

Figure 5

condition $d_{s,m} + d_{m,c} > d_{s,c}$ is fulfilled, accordingly, the following is fair: $D_m > D_c$. Let's compare magnitude D_c with D_k. Since by the definition $d_{s,c}$ is the shortest path between vertexes v_s and v_c, so the condition $d_{s,k} + d_{c,k} > d_{s,c}$, is fulfilled all the time, accordingly, the following is fair: $D_k > D_c$. Therefore, D_c is minimal among other magnitudes D_m and D_k, which was to be proved. There is a variant of graph is presented in the Figure 5c, which has $d_{s,c} > d_{s,m} + d_{m,c}$ и $d_{s,c} > d_{s,к} + d_{c,к}$. In this case $D_c > D_k$ and $D_c > D_m$. Therefore, the path $d_{s,c}$ is not the shortest path, and according to the condition the shortest path is chosen as a primarily path. In this case lets this path $L_c(v_s,v_c)$ be passing through the vertex v_k, i.e. $L_c(v_s,v_c) = e_{s,k} + e_{k,c}$. The length of this path equals $d_{s,m} + d_{m,c}$. In this case $D_k = D_c$ and $D_m > D_c$. So the magnitude D_c is minimal in case of the vertex v_c position in the middle of the path $L_0(v_m,v_k)$. The median vertex of the path is determined by the data: among other all vertices $v_i \in V_0$ the vertex v_c with the low value $\Delta_c = |D(v_c,v_m) - D(v_c,v_k)|$ is chosen. In the general way $\Delta_c \neq 0$, since the length of edges may differ.

Therefore, forming of message transmission tree lies in the following:

For the vertices $v_i \in V_0$, belonging to the path $L_0(v_m,v_k)$ the magnitude Δ_i is calculated.

Among the vertices $v_i \in V_0$ a vertex with minimal magnitude Δ_c is chosen.

The minimal path L (v_s, v_c) is determined between the source v_s and vertex v_c.

In the presence of vertices $v_i \in V_0$, $v_j \in V_0$ with $\Delta_i \cong \Delta_j$ as potential two groups of vertices are considered. In this case the vertex with minimal path length L (v_s,v_c) is chosen.

The routes $L_c(v_s,v_m) = L(v_s,v_c) + L(v_c,v_m)$ and $L_c(v_s,v_k) = L(v_s,v_c) + L(v_c,v_k)$ are forming through the vertex v_c.

As an example, we will watch the forming of the path in the graph, presented in the Figure 6. Let, as it was before, a vertex v_s will be a vertex v_4. AC_2 moves between vertices $v_{13} - v_{17} - v_{18} - v_{19} - v_{10}$, the length of this path is 15.

In the table below we can see the magnitude L_i^m of the distance between vertexes v_i and the outermost extreme vertex, also tabulated magnitude $L(v_s,v_i)$, Δ_i, D_i for all vertices $v_i \in V_0$ where:

Δ_i–Difference in distances between the defined vertex and the extreme vertex;

$L(v_s,v_i)$–minimal path between the source and the defined vertex;

D_i–diameter of the tree; D_{cp} – average magnitude $D_i k_i = (D_i - D_{min}) / D_{min}$

Table 1

	v_{13}	v_{17}	v_{18}	v_{19}	v_{10}
Δ_i	15	7	1	7	15
L_i^m	15	11	8	11	15
$L(v_s,v_i)$	10	12	11	15	13
D_i	25	23	19	26	28
$D_i - D_{cp}$	0.8	-1.2	-5.2	1.8	3.8
k_i	0.32	0.21	0	0.37	0.47

In this case the minimal magnitude $\Delta_c = 1$ corresponds the vertex v_{18}, for which the magnitude $D_{18} = 19$ is minimal among all D_i.

4.2 The Path Reconfiguration

We will study the reconfiguration of the virtual path on minimal length criteria. Let's suppose, that like in the previous case, AC_2 moves on the path $L_0(v_{13},v_{10})$ from the vertex v_{13} to the vertex$_{10}$. In this case, while moving AC_2 to the vertex v_{17} instead of the shortest path $L_1(v_4,v_{13})$, passing through the vertices v_8 and v_{12} a new shortest path $L_2(v_4,v_{17})$ will be formed, passing through the vertexes v_{14}, v_{11}, v_7. Eventually, an amount of routes corresponding to the amount of the route vertices $L_0(v_{13},v_{10})$ will be formed. In this case the amount of the paths equals 5 with the total amount of the edges 12. They are indicated with fat lines in the Figure 6. As it is shown in the Figure 6 a separate path may lead to the vertices or partly the routes might coincide, in this case on the graph the vertices are present with branching path. In this case while passing from the vertex v_{13} into the vertex v_{17} reconfiguration of the whole path takes place, as a result 3 edges are changed for 4 new edges, time complexity equals to 7.

Upon crossing from the vertex v_{17} into the vertex v_{18} the time complexity equals 5. Upon crossing from the vertex v_{18} into the vertex v_{19} the time complexity equals 3. Upon crossing from the vertex v_{19} into the vertex v_{10} the time complexity equals to 3. Therefore, as a result of the moving AC_2 on the path $L_0(v_{13},v_{10})$ the total time complexity equals 18.

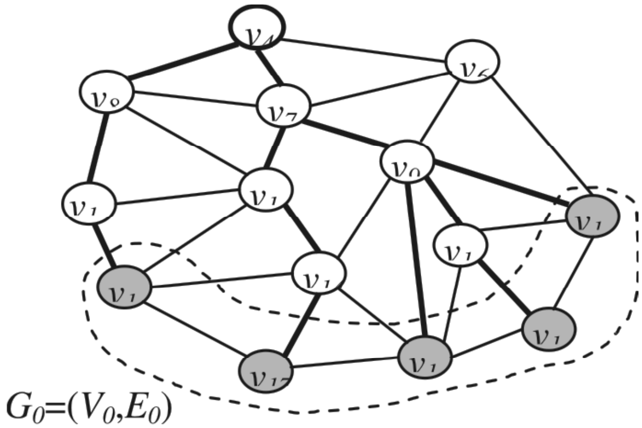


Figure 6

There is presented a tree structure with minimal time complexity reconfigurations in the Figure 7. In this case an amount of tree edges are diminished on 25% and equals 9.

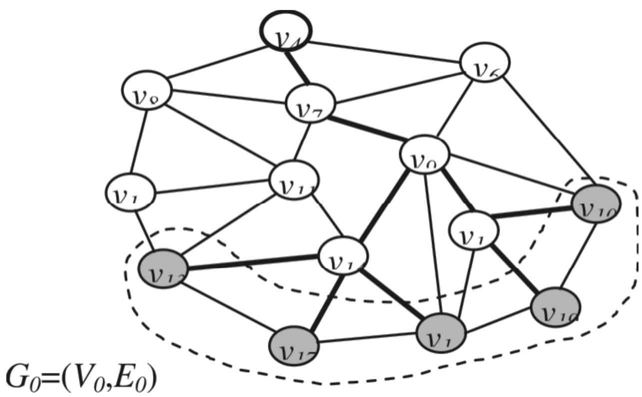


Figure 7

Branching Nodes are closed to the subgraph $G_0 = (V_0, E0)$ at most. It allows diminishing the time complexity of rerouting. In this case transition from the vertex v_{13} into the vertex v_{17} one edge is changed to a new edge, time complexity is 2. While transition from the vertex v_{17} into the vertex v_{18} the time complexity also is 2. While transition from the vertex v_{18} into the vertex v_{19} the time complexity is 4. While transition from the vertex v_{19} into the vertex v_{10} the time complexity is 2. Therefore, in consequence of transmission of AC_2 on the path $L_0 (v_{13}, v_{10})$ the total time complexity diminished in twice and equals 10. Therefore, the task of minimizing the time complexity of rerouting comes to the task of forming the minimal amount of branching nodes approximate to the vertices of subgraph $G_0 = (V_0, E_0)$.

This task refers to the task of forming the shortest tree on the great amount of the vertices $\{V_0, v_s\} \subset V$ of the primarily graph $G = (V, E)$. Having formed the tree, its vertex gets connected on the shortest path to the source vertex. In this case such vertex is the vertex $v_{9,}$ from which the shortest path is forming to the vertex v_4.

In general, while AC is moving between the local sub networks, presented by according sub graphs $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$, the initial path $L_0(v_k, v_m)$ in the graph $G_0 = (V_0, E_0)$ is changed to the new path $L_1(v_k, v_l)$. The boundary vertex v_i of the sub graph $G_2 = (V_2, E_2)$, is determined due to the condition $v_k \in V_2\,(\exists e_{k,i}, v_i \in V_0)$. Lets suppose, that the path $L_0(v_k, v_m)$ consists of set of edges $E_{L0}=\{e_{i,j} \in L_0(v_k, v_m)$, and a path $L_1(v_k, v_l)$ consists of a set of edges $E_{L1}=\{e_{i,j} \in L_1(v_k, v_l)$. As a result of AC moving some sets of edges $E_m \subseteq E_{L0}$ interchange to new sets of edges $E_l \subseteq E_{L1}$. Combination of this sets forms sets of edges $E_{s,}$ which are involved in the process of path reconfiguration. The power of set E_s characterizes the time complexity of the initial path reconfiguration procedure. Set E_s is determined on the basis of the following formulation:

$$E_s = (E_{L0} \cup E_{L1}) \setminus (E_{L0} \cup E_{L1}) \tag{5}$$

If the paths $L_0(v_k, v_m)$ and $L_1(v_k, v_l)$ do not coincide, then $E_{L0} \cap E_{L1} = \varnothing$. In this case the power of the set E_s is maximal. Forming the route $L_1(v_k, v_l)$ at the expense of prolongation of the initial route $L_0(v_k, v_m)$ power Q_S of set E_s is minimal. Lets suggest, that as a result of rerouting edge $e_{i,j}$ is adding. In this case set of edges $E_{L1}=\{E_{L0}, e_{i,j}\}$ can be presented as $E_{L1} = E_{L0} \cup \{e_{i,j}\}$. Therefore: $E_s = (E_{L0} \cup (E_{L0} \cup \{e_{i,j}\})) \setminus (E_{L0} \cap (E_{L0} \cup \{e_{i,j}\})) = (E_{L0} \cup e_{i,j}) \setminus E_{L0} = \{e_{i,j}\}$.

While AC is moving between some local networks set of paths are forming $\{L_i(v_k, v_j) \mid i = 0, \ldots, (n\text{-}1); j = p, \ldots q\}$ between the initial vertex v_k and the vertex of the sets $V_r = \{v_j \mid j = p, \ldots q\}$. In this case:

$$E_s = (E_{L0} \cup E_{L1} \cup \ldots \cup E_{L(n-1)}) \setminus (E_{L0} \cap E_{L1} \cap \ldots \cap E_{L(n-1)}) \tag{6}$$

For the evaluation of the path reconfiguration complexity we will use the following power factor:

$$k_s = \frac{Q_s}{\sum_{i=0}^{n-1} Q_{Li}} \tag{7}$$

Where: Q_{Li}–power of the set E_{Li}.

Magnitude of the power factor k_s is changing in the range (0-1). If $k_s \to 0$ the path reconfiguration will be minimal. If $k_s = 1$ the whole path is reformed completely. As it follows from the formula (7) $k_s \to 0$ if $Q_S \to 0$. In turn Q_S is determined by the amount of edges, forming the path between the vertices v_p and v_q. Magnitude Q_S is minimal only for the shortest path between v_p and v_q. Then we will form the shortest path from the initial vertex to the closest vertex belong to the set of information destination vertices. As a result, from the set of paths a spanning tree $ST_k = (V_k, Ek)$ will be formed with the initial vertex v_k and minimal magnitude k_s.

Therefore, the task of forming information delivery routes on the criteria of rerouting time complexity results to the task of forming the spanning tree with minimal number of edges. The solution of this task lies in the determination of the shortest path from the vertex v_k to the closest vertex $v_m \in V_r$.

The proposed algorithm description:

1. The minimal path L0 (vk, vm) from vertex vk to the closest vertex vm∈Vr is determined by Dijkstra's algorithm.
2. Vertex vm is eliminated from the Vr set and included in the EL0 set, i.e. vm ∉ Vr → vm ∈ EL0.
3. Among the remaining vertices of the Vr set, there is vertex vi ∈ Vr which the closest to one of vertices vj ∈ EL0. Supposing that path L0(vk, vm) is minimal, vertex vm will be as vj.
4. The path is formed as: L1(vk,vi) = L0(vk, vm) + em, i.
5. Vertex vi is eliminated from the Vr set and included in the EL1 set, i.e. vi ∉ Vr → vi ∈ EL1.
6. Among the remaining vertices of the Vr set there is vertex vi+1 ∈ Vr which the closest to one of vertices vj ∈ EL0 or vl ∈ EL1.
7. At ei, j > ei,l, path L1(vk, vi + 1) = L1(vk, vl)+ el,i + 1 otherwise path L0(vk, vi+1) = L0(vk, vm)+ ej,i + 1 is extended.
8. Vertex vi + 1 is eliminated from the Vr set and included in the EL1 or EL0 set.
9. At Vr ≠ Ø turn to point 6.
10. The process of delivery tree forming is complete.

In this case two paths L0(vk, vp) and L1(vk, vq) are formed from the vertex vk to the extreme vertexes vp and vq. In case when one of the vertexes vp or vq is appeared to be vm, only one path is formed L0(vk, vp) or L1(vk, vq). Therefore a suggested algorithm of delivery tree forming is characterized by the minimal time complexity of rerouting, however restriction on the transmission delay is neglected: td >tm. For this reason in points 3 and 6 on choosing the tree vertex, to which a new vertex connected, the condition $t_d > t_m$ is checking additionally. Fulfilling this condition the connection of the vertex is carried out. Otherwise as a candidate for the connection is chosen the following vertex of previously formed tree and so on and so forth. In better case, as without regard to the condition $t_d > t_m$, only one path is going to be formed, connecting all vertices of sets V_r with the vertex v_k. In the worst case, upon strict restrictions on transmission delay, delivery tree can be formed only upon following condition $(E_{L0} \cap E_{L1} \cap ... \cap E_{L(n-1)}) = \varnothing$. Here in reconfiguration task of virtual channel results to the complete rerouting. This result might be generalized in case of vertex v_s moves around. In this case the tree $ST_k(V_k, E_k)$ is accompanied with new vertices of set V_s Tree forming algorithm may be used and for case when the source (vertex v_s) is moves around. In this case within the framework of subgraph $G_0 = (V_0, E_0)$ set $V_s = \{v_z \mid z = k, ...h\}$ are formed, reflecting vertex v_s travel. As a result the tree $ST_k = (V_k, E_k)$ is accompanied with new vertices from set $V_s \subset V_0$. Vertices $v_z \in V_s$ connection is carried out concerning the vertex v_k in accordance with submitted algorithm before. Forming the route between network subscribers it is reasonable to regard a possibility of AC movement. The problem of restriction on delay in the process of delivery tree forming may be described by the following way:

- Connected graph G = (V, E), is given, where V – set of its vertices and E–set of its edges, and also two functions: cost c(i, j) for network usage (i, j) ∈ E and delay d(i, j) in the direction (i, j) ∈ E.
- To find a tree T = (VT, ET), where T ⊆ G connecting vertices vs and vi , i =1,.., n ∈ V so that $\sum_{(i,j)\in E_T} c(i,j)$ is minimized and $\forall i, i = 1, n$: D(vs,vi) ≤ ΔT, delay restriction, where D(vs,vi) = $\sum_{(i,j)} d(i,j)$ for all (i, j) on the path from vs to vi in the tree T.

5. TIME COMPLEXITY EVALUATION

Dijkstra algorithm time complexity is equal to [8]:

$$T_d = O\left(\frac{n \times k}{2} \times \log n\right) \tag{8}$$

Where n – the number of graph nodes; k – medium extent of nodes.

Generally, $k = \frac{2 \times l}{n}$ where l– the number of graph edges.

Dijkstra algorithm realized by proposed algorithm in 2 stages: At the first stage, the shortest path up to one of node of the subset $V_1 \subset V$ is searched. In our case the number of nodes of this subset V_1 is equal n_1. The number of nodes of other subset $V_0 = V/V_1$ is equal $(n-n_1)$.

As consistent with the formula (1), the time complexity of finding the shortest path in subset V/V_1 it is equal:

$$T_{d_0} = O\left(\frac{(n-n_1) \times k}{2} \times \log(n-n_1)\right) \tag{9}$$

At the second stage, the time complexity of finding a path inside subset V_1 with nodes n_1 is determined by:

$$T_{d1} = O\left(\frac{n_1 \times k}{2} \times \log n_1\right) \tag{10}$$

Thus total time complexity on the ground of formulas (9) and (10):

$$T_m = O\left(\frac{k}{2} \times \left(n \times \log_2(n-n_1) - n_1\left(\log\left(\frac{n}{n_1} - 1\right)\right)\right)\right) \tag{11}$$

Where n_1 - the number of nodes in the sub graph to which data is delivered.

5.1 The Expected Results

As it is shown in the Figure 8 the time complexity of Dijkstra algorithm is presented in upper line and the time complexity of proposed algorithm is presented in lower line. Here

$k = 4$, value of $n_1 = 20$, and value of n varies from 20 to 40. As it is shown on the Figure 6, with increasing the number of graph nodes, efficiency of the proposed algorithm is higher in comparison with Dijkstra algorithm.

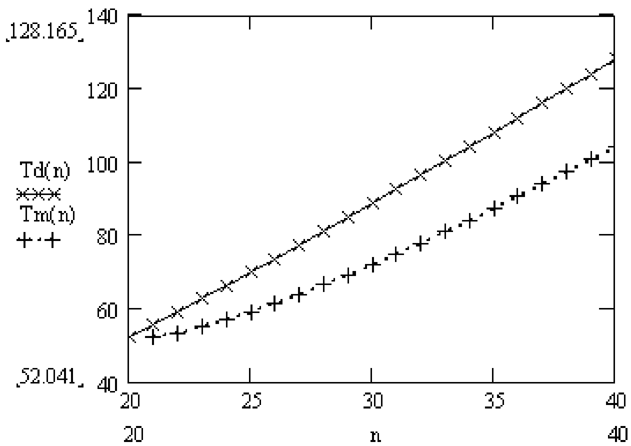


Figure 8

6. PROPOSED ALGORITHM BASED ON MOBILITY PREDICTION

Some value P_i (which can indicate probability of subscriber's connection to this vertex (node) while moving on the network) corresponds to each v_i vertex of the graph G (V, E). Some value $c_{i,j}$ (which indicates cost of data-transmission by the channel) corresponds to each edge $e_{i,j}$ of the graph G = (V, E). With subscriber moving across the network his next location is a priori determined by the P_i-probabilities, therefore the following problem solution is offered: to use the following general criterion of minimization $K_i = P_i/C_i$ at covering tree construction, where C_i - the complete cost of the path to the vertex i, and value P_i - probability of the subscriber movement to the vertex i. Value K_i is directly proportional to probability P_i and inversely proportional to the cost of path from a source to the vertex i, i.e. the preferable path of the possible paths set is that one whose criterion K_i is maximum.

The adaptive routing algorithm has been developed on the basis of the analysis the known ways of data transfer organization as well as taking into account the peculiarities of mobile corporative networks.

The path formation is carried out within the framework of the virtual subnet with the minimal number of edges, which is formed with consideration of possible mobile terminal's moving. This allows to reduce time and to increase the efficiency of routing procedure. The delivery tree formation within the framework of the virtual subnet is carried out similarly to previous algorithm , taking into account the probability of the mobile terminal's stay in this or that local subnet. It allows forming the information delivery path with the minimal time complexity of its reconfiguration. The values of re-routing time complexity

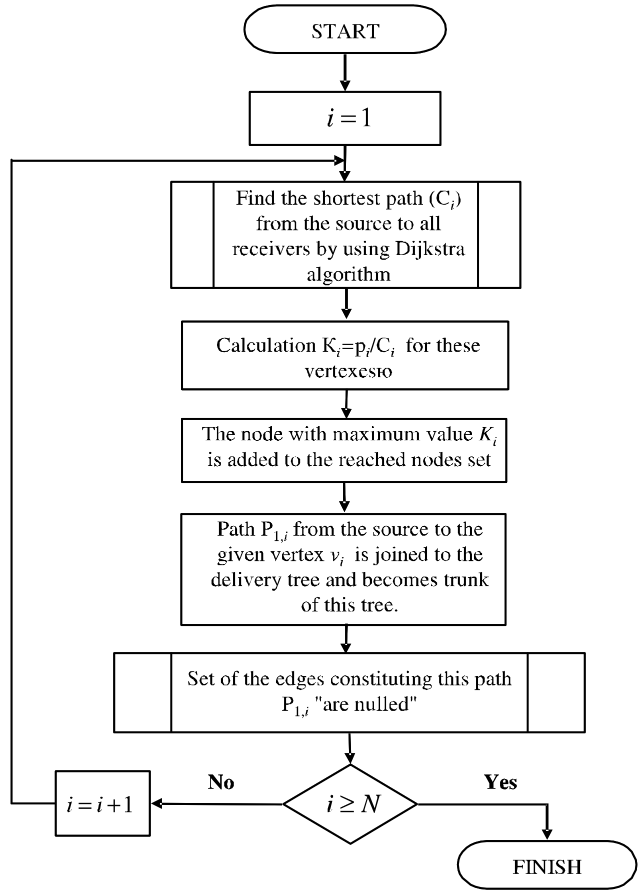


Figure 9

in case of different methods of information delivery tree building and different network topologies are shown in Fig. 10 and Fig. 11.

The fully connected structure is chosen as an initial one. Each new structure is formed from the previous one by means of eliminating the separate edges in the previous structure. The dependence of temporary complexity when building the delivery tree with two levels is shown in Figure 10. The dependence of temporary complexity when building the delivery tree with ten levels is shown in Figure 11.

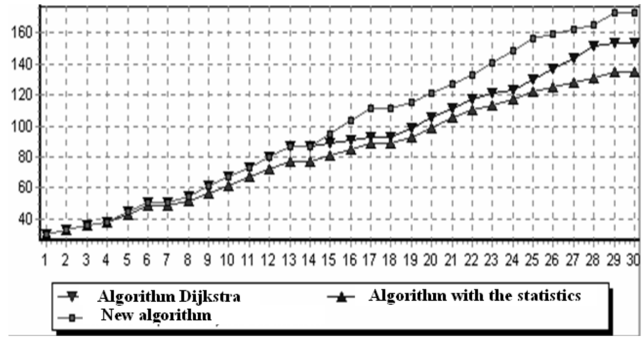


Figure 10

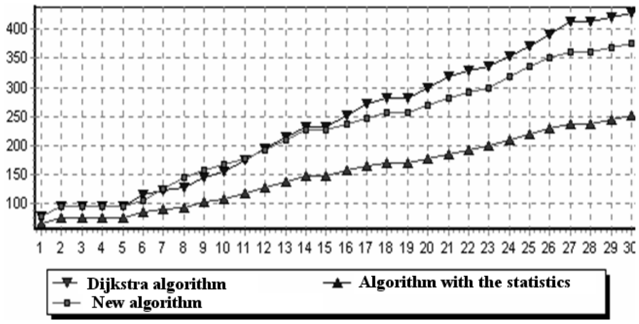


Figure 11

As follows from Fig. 10 and Fig. 11, when the number of levels increases the efficiency of the proposed method exceeds the efficiency of the method based on Dijkstra's algorithm. With the purpose of providing required level of network services it is proposed a combine method of rerouting, which combines advantages of route prolongation method with method's benefits on the basis of preset tree. Initially the algorithm forms extended path for providing nonbreaking transmission, with further path optimization. As a sufficient condition of route prolongation the following condition is considered:

$$t_d \geq \sum_{e_{x,y} \in L_1(v_s, v_k)} t_{x,y} + \sum_{e_{i,j} \in L_0(v_k, v_m)} t_{i,j} + \sum_{e_{z,q} \in L_2(v_m, v_r)} t_{z,q} + \sum_{e_{p,g} \in L_z(v_k, v_m)} t_{p,g}$$

where: $t_{p,g}$–time of cells transmission in the edge $e_{p,g}$ in the path $L_Z(v_k, v_m)$ of ÀÑ transmission between external rooters v_k and v_m related local sub-networks. Route optimization as well as its cycling trim is carried out by means of reconfiguration (reshuffle) previously added path locations. Based on the analysis of famous reshuffle scheme it is proposed to use a scheme of adaptive reshuffle which is a modification of nonbreaking reshuffling scheme on the basis of AC moving.

7. CONCLUSION

1. Most of known routing Algorithms like Dijkstra prove to be ineffective in mobile networks. In addition, along with the mobility increase control information needed for rerouting grows sharply too.
2. Suggested two methods of rerouting. The first method is to prolong the route; the second one is route reconfiguration. In the first case a time complexity of rerouting is minimal, but a threat of excessive prolongation and possibility for loops appearance in it is appeared.
3. Path reconfiguration is characterized by the greater time complexity, but allows generating an optimum path.
4. Proposed approach to creating tree of the shortest paths, which takes into account subscriber movement probability, allows minimizing number of "returns" in the delivery tree structure and thus reduces rerouting time.
5. Proposed algorithm can be used for both single-address and multiple-address routing.

REFERENCES

[1] Kulgen M., Encyclopedia: *Corporate Networks Technology*, Piter, 2000.

[2] Chandra N. Sekharan, and Gopal Racherla, A Distributed Rerouting Algorithm for Mobile-Mobile connections in Connection-Oriented Networks, *icccn, International Conference on Computer Communications and Networks*, October 1998.

[3] A. Tanenbaum, *Computer Networks*, fourth edition, Prentice Hall, 2003.

[4] B. Akyol and D. C. Cox, "Handling Mobility in a Wireless ATM Network," In *Proceedings of IEEE INFOCOM, San Francisco, CA, USA*, March 1996.

[5] G. Racherla, S. Radhakrishnan, and C. N. Sekharan, A Framework for Evaluation of Rerouting in Connection Oriented Mobile Networks. Technical Report, School of Computer Science, University of Oklahoma, 1997.

[6] S. Seshan, "Low Latency Handoff for Cellular Data Networks", PhD thesis, University of California, Berkeley, 1995.

[7] C. Toh, "The Design and Implementation of a Hybrid Handover Protocol for Multimedia Wireless Lans," *In Proceedings of ACM Mobicom*, Berkeley, CA, USA, November 1995.

[8] V. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Proc. *INFOCOM'97*, Apr. 1997, 9 pages.

[9] A. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Addison Wesley, 2000.

[10] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks", *ACM Mobile Networks and App. J.*, Special Issue on Routing in Mobile Communication Networks, Oct. 1996, pp. 183-97.

[11] Sherbo V. *Standard of Computer Networks*, Inter Couplings of Networks: the Quick Reference, M. KUDITS-MODE, 2000.

[12] Mingliang Jiang, Jinyang Li, Y. Tay, *Cluster Based Routing Protocol*, August 1999 IETF Draft, 27 pages. http://www.ietf.org/internet-drafts/draft-ietf-manet-cbrp-spec-01.txt.

[13] Sarrafzadeh M., "Wong C. K. Bottleneck Steiner Trees in the Plane",–*IEEE. Trans. on Computers*, 1992, Vol. 41, No. 3.

[14] Vijayan G. "Generalization of Min-Cut Partitioning to Tree Structures and Its Applications",–*IEEE. Trans. On Computers*, 1991, Vol. 40, No. 3.