ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

PROGRESSIVE INFORMATION TECHNOLOGIES

ПРОГРЕССИВНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

UDC 681.5 (048)

DELAY TOLERANT NETWORKING SUPPORT FOR CREATION HIGH-ACCURACY MAGNETIC FIELD MAPS

Babeniuk G. M. – Post-graduate student, Department of Aeronavigation of National Aviation University, Kyiv, Ukraine.

ABSTRACT

Context. The main purpose of Correlation Extremal Navigation system is finding coordinates in case of absence of Global Positioning System signal and as a result high-accuracy maps as the main source of information for finding coordinates are very important. Magnetic field map as the main source of information can include errors values, as an example: not good enough equipment or human factor can cause error value of measurements.

Objective. In order to create high-accuracy maps given work proposes to improve the process of creating magnetic field maps. The given work represents delay tolerant networking as an additional approach for data transmission between magnetic observatory and magnetic station and its improvement.

Method. Improved Dijkstra's algorithm together with Ford-Fulkerson's algorithm for finding path with minimum capacity losses, earliest delivery time and maximum bit rate in case of overlapping contacts should be represented in the given work because nowadays, delay tolerant networking routing protocols do not take into account the overlap factor and resulting capacity losses and it leads to big problems

Results. For the first time will be presented algorithm that chooses the route that guarantees the minimum of capacity losses, earliest delivery time and maximum bit rate in the delay tolerant networking with overlapping contacts and increases the probability of successful data transmission between magnetic stations and magnetic observatories.

Conclusions. In order to perform high-accuracy measurement of magnetic field group of people allocate their equipment for magnetic field measurement in remote areas in order to avoid the influence of environment on measurements of magnetometer. Since magnitude of magnetic field can vary dependent on temperature, proximity to the ocean, latitude (diurnal variation of magnetic field) and magnetic storms magnetic station from time to time adjusts its measurements with a help of reference values of magnetic field (magnetic station sends request for reference values to magnetic observatory). The problem of the given approach is that remote areas usually are not covered by network (no Internet) and as a result the adjustment of measurements is impossible. In order to make adjustment of measurements possible and as a result improve accuracy of magnetic maps given work proposed the usage of Delay Tolerant Networking that delivers internet access to different areas around the world and represented its improvement to make its approach even better.The results are published for the first time.

KEYWORDS: Correlation Extremal Navigation system, Ford-Fulkerson's algorithm, magnetic maps, Dijkstra's algorithm.

ABBREVIATIONS

CENS is Correlation Extremal Navigation system;
DTN is Delay Tolerant Networking;
UAV is Unmanned Aerial Vehicle;
CGR is Contact Graph Routing protocol;
ION is Interplanetary Overlay Network;
LEO is low Earth orbit;
AOS is acquisition of-signal;
LOS is loss-of-signal.

DOI 10.15588/1607-3274-2021-1-15

NOMENCLATURE

x_i is a support time;

$\Pi_i(x_i)$ is a utility function of x_i;

$L_i(x_i)$ is a loss function of x_i;

st is a start time of acquisition of signal of the satellite;

end is a time when loss of signal of the satellite occurs;

$M = \{S1, S2, ..., S_m\}$ is a set of satellites;

$[a_i, b_i], i = 1, 2, .., m$ is a time window for satellites.

INTRODUCTION

CENS is based on the idea of comparison of current realization of maps with those saved in memory and by finding the maximal match between current and template maps it identifies the coordinates of UAV when GPS signal is absent.

Template maps can be of different types but in this work attention will be focused on magnetic field maps and their accuracy.

In order to increase the accuracy of magnetic field maps given work proposes the usage of DTN as an additional network that can guarantee the access to data of magnetic observatory in any area around the world.

OneWeb worldwide satellite broadband network is based on integration of LEO satellites and DTN. Given network can be perfectly used as support for creating high-accuracy magnetic field maps.

The minimum theoretical latency from ground to satellite of OneWeb worldwide satellite broadband network is about 1–4 milliseconds; a LEO satellite constellation also provides more system capacity by frequency reuse across its coverage; the data transmission rate is about 400Mbit/sec (100Mbit/sec have standard http protocols). The disadvantage of having a OneWeb satellite constellation of 700 satellites is cost. While the OneWeb satellites are physically much, much smaller—about 150kg (330lbs) per OneWeb satellite vs. 4000kg (9000lbs) [1] for a geosynchronous communications satellite-it costs a lot to build 900 satellites and launch 700 satellites into space (The extra 200 satellites will be held back on Earth until they're needed) [2].

Despite represented disadvantage given work highly recommends OneWeb satellite constellation as support for creating high-accuracy magnetic field maps [3].

Another aspect that is very important is that due to large number of LEO [4] satellites and large surface coverage the probability of overlapping contacts of the nodes increases and consequently more data losses happen. The cause why capacity loss of the channel occurs, from technical perspective, is due to the fact that single-antenna installation is incapable to maintain uninterrupted operations from several satellites simultaneously and support of multiple installations is expensive; ground station selects only one satellite at a point in time while the data of another satellite getting ignored.

In order to have high quality support for creating high-accuracy magnetic field maps in the given work the mentioned above problem has to be solved. Improved routing algorithm for finding path with minimum capacity losses, earliest delivery time and maximum bit rate in case of overlapping contacts should be represented in the given work because nowadays, delay tolerant networking routing protocols do not take into account the overlap factor and resulting capacity losses and it leads to big problems.

The object of study is the process of creation of high-accuracy magnetic field maps.

The subject of study is improvement of existing approaches for magnetic field maps creation.

The purpose of the work: usage of improved DTN as an additional approach for data transmission between magnetic observatory and magnetic station and its improvement.

1 PROBLEM STATEMENT

DTN is by default the best approach for delivering internet access in remote areas where standard Internet protocols cannot be used that's why the benefit of usage of the given approach as support for creating high-accuracy magnetic field maps is obvious.

Problem statement: the given work should represent the improvement for existing DTN thereby improving the accuracy of magnetic field maps i.e. should be developed an algorithm that aims to find out the optimal path that guarantees the minimum of network losses, earliest delivery time and maximum bit rate.

Let's suppose we have instance: m satellites, $[a_i, b_i], i = 1, 2, .., m$ for S_i reconfiguration time $r = 0$, $\Pi_i(x_i)$ for satellites $S_i, i = 1, 2, ..., m$ and $L_i(x_i)$ for satellites $S_i, i = 1, 2, ..., m$ in the network with overlapping contacts.

Objective: find path with maximum $\Pi_i(x_i)$ and minimum $L_i(x_i)$.

Based on the mentioned above picture the following rules for optimal path selection for default satellites $S1$ and $S2$ can be distinguished.

– if $\Pi_{s1}(x_{s1}) > \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) < L_{s2}(x_{s2})$, then $S1$ node is selected.

– if $\Pi_{s1}(x_{s1}) < \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) < L_{s2}(x_{s2})$, then $S1$ node is selected.

– if $\Pi_{s1}(x_{s1}) > \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) == L_{s2}(x_{s2})$, then $S1$ node is selected.

– if $\Pi_{s1}(x_{s1}) < \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) == L_{s2}(x_{s2})$, then $S2$ node is selected.

– if $\Pi_{s1}(x_{s1}) < \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) > L_{s2}(x_{s2})$, then $S2$ node is selected.

– if $\Pi_{s1}(x_{s1}) > \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) > L_{s2}(x_{s2})$, then $S2$ node is selected.

– if $\Pi_{s1}(x_{s1}) == \Pi_{s2}(x_{s2})\,\&\&L_{s1}(x_{s1}) == L_{s2}(x_{s2})$, then other features should be analyzed.

These are the basic rules presented in this work based on which the optimal path has to be selected.

2 REVIEW OF THE LITERATURE

CGR protocol as an implementation of DTN and basic functionality of which should be improved in order to solve the problem of overlapping contacts, proposed by S. Burleigh, follows a distributed approach: the next hop is determined by each DTN node on the path with a help of recomputing the best route to destination, as soon as a bundle is received. This routing procedure takes into

DOI 10.15588/1607-3274-2021-1-15

account that a global contact plan [5], comprising all forthcoming contacts, is timely distributed in advance in order to enable each node [6] to have an accurate understanding of the network [7].

Contact plan is the main source of information that provides information about network i.e. neighborhood nodes (the start and end time of their contacts) [8]. Routes can thus be calculated by each node on demand, based on extensive topological knowledge of the network (contact plan) [9]. This workflow where a contact plan can be initially determined by means of orbital propagators and communication models, then distributed to the network, and finally used by the DTN nodes to calculate routes to the required destinations. CGR can dynamically respond to changes in network topology and traffic demands. An early version of CGR was flight-validated in Deep Space [10] by National Aeronautics and Space Administration in 2008 and CGR has been estimated as the most studied routing solutions for space networking since then [11].

CGR was documented in 2009 and later updated in 2010 as an experimental Internet Draft.

By the end of the same year, Segui et al. presented earliest-arrival-time as a convenient monotonically decreasing optimization metric that avoids routing loops and enables the use of standard Dijkstra's algorithm for path selection. This enhancement was then presented in the official version of CGR, included in the Interplanetary Overlay Network DTN Stack developed by the Jet Propulsion Laboratory. Later, in 2014, authors observed the implementation of temporal route lists as a mean to minimize CGR executions. In the same year, Bezirgiannidis et al. proposed to monitor transmission queues within CGR as they increase the earliest transmission opportunity. the same paper, a complementary overbooking management innovation enables proactive reforwarding of bundles whose place in the outbound queue was presented by subsequent higher priority traffic. Both ETO and overbooking management are now presented in the official CGR version. Regarding congestion management, further extensions were provided as well. Most recently, in 2016, Burleigh et al. presented an opportunistic extension as a means of enlarging CGR applicability from deterministic space networks to opportunistic terrestrial networks. The latter contribution since it could, if successful; pave the way towards implementing space DTN advances on ground-based networks. At the time of this writing, CGR procedure is being formally presented as part of the Schedule-Aware Bundle Routing (SABR) procedure in a CCSDS Blue Book. Finally, this software becomes an important point of reference for the latest routing and forwarding mechanisms for space DTNs.

3 MATERIALS AND METHODS

In order to find out the optimal path by considering not only the shortest distance but also the network losses and to perform scheduling of bundles of CGR protocol, the following single processing steps should be examined (Fig. 1):

– Step 1. Initial configuration;
– Step 2. Selection of highest priority bundle;
– Step 3. Finding the max flow value in flow network;
– Step 4. Checking the size of bundle;
– Step 5. Calculation of optimal path;
– Step 6. Finding the max flow value in optimal path;
– Step 7. Updates in contact plan;
– Step 8. Finding the rest data.

In processing step № 1 "sender" should invoke some function in application service to send a unit of application data to a remote counterpart. The destination of the application data unit should be expressed as a bundle protocol endpoint identifier. The application data unit should be encapsulated in a bundle. Each bundle should have its own "sender-defined" priority and should be forwarded over selected path build based on the "sender-defined" contact plan. At the beginning program asks the user to enter the following data: ID of source node, ID of destination node, bundle itself (program reads data from a file whose name is input by the user), bundle priority and contact plan, represented as Json object. So to say, user provides information about transmission details. Data should be entered in specific order. If user wants to forward several (more than one, as an example: 2) bundles, user has to enter the ID of source node for first bundle and after for the second one then in the same order user should enter the ID of destination node for first bundle and after for second one and bundle priority for first bundle and after for second one, etc., the order is important because program associates the bundle with its index position and all the data belong to the bundle should have the same index position. Program retrieves bundle by highest priority (as was discussed bundle priority should has the same index position as bundle to which it belongs). After successful data forwarding, all the data by certain index position will be removed, in order to avoid infinite loop, after program will retrieve another bundle by highest priority (lowest value – highest priority) and everything will start again from the beginning. User input example: tvg_g10_s14.json -pay bundle_1.txt bundle_2.txt -s "FLOCK 1C-4" "FLOCK 1C-4" -e "gs18" "gs18" -pr 1 2

In processing step № 2 is stating that processing logic should be aware of the priority of bundles to guarantee reduction of transmission delays. Exist different priority levels: 2 = bulk, 1 = normal, 0 = expedited. In the proposed approach bundles of low-priority value should represent top priority. In processing step № 3 the max flow value in flow network should be calculated. In the proposed approach the Ford-Fulkerson's algorithm should be used in order to find max flow in the flow network to identify amount of data that can be transferred from one point to another. In order to truly understand the current state of the network – maximum number of bits that can be transferred from the source node to sink node (source node and sink node are user inputs) program uses Ford-Fulkerson's algorithm. Program has contact plan – user-defined (as an example: contact plan can be presented in form shown by contact plan updater) and implementation

© Babeniuk G. M., 2021
DOI 10.15588/1607-3274-2021-1-15

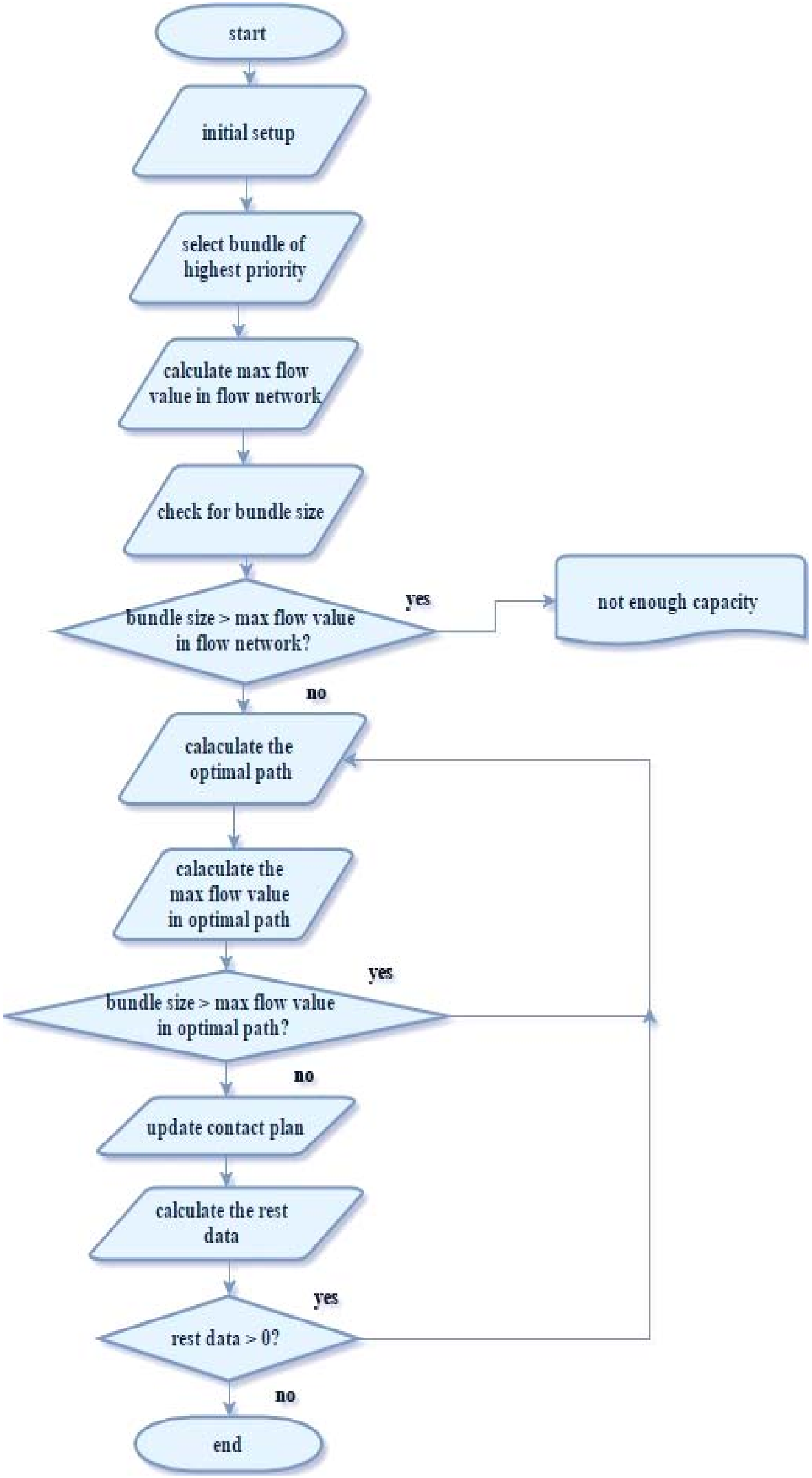


Figure 1 – Overall process flow that is triggered in case optimization approach is applied

of Ford-Fulkerson's algorithm represents a flow network based on the contact plan where every edge has a capacity. Capacity of an edge is the maximum number of bits that can be transferred through this edge and equals: end time of loss-of-signal (LOS) – start time of acquisition of-signal (AOS) * bit rate. The path between nodes is characterized by start time of acquisition of-signal (AOS) and end time of loss-of-signal (LOS). As was already discussed one node can has several start nodes. When start node links to neighboring node it requests the history of communications (earliest start time and earliest delivery time; with regard to the size of bundle) between neighboring node and other start nodes. The main advantage of this algorithm is that it shows how

© Babeniuk G. M., 2021
DOI 10.15588/1607-3274-2021-1-15

much "flow" can the network (all existing paths) process at a time for specified source node and sink node. After that, program compares the size of bundle and max flow value of the network, if the size of bundle is less than max flow value then program transfers an entire bundle completely otherwise bundle waits for updates in existing contact plan (or fragmentation is applied). As an example if the bundle size is 45 bits and max flow value is 25 bits then 45-25 = 20, so 20 bits should wait for updates in existing contact plan because current one has insufficient capacity.

In processing step №4 the proposed approach calculates the size of the bundle. A bundle itself plays an important role. In case the size of bundle is bigger than max flow value in flow network bundle can't be forwarded because current network has not enough capacity to forward that bundle otherwise the processing step №5 activates.

The network max flow value is value defined by Ford-Fulkerson's algorithm for specified source node and destination node; and if the contact plan has a lack of capacity for certain bundle (for certain destination of bundle) it doesn't mean that other bundles (other bundle endpoints) can't be transferred through the network. In our example was determined the current state of the network – max flow value for concrete bundle (for concrete destination of bundle) i.e. was determined that the size of bundle is bigger than max flow value and that the part of this bundle which can't be transferred should wait for updates in existing contact plan while other low priority bundles can be transferred. If the size of bundle is less than max flow value then program calculates the optimal path for this bundle – Dijkstra's algorithm.

In processing step №5 proposed approach calculates the optimal path. Our implementation of Dijkstra's algorithm gives top priority to the minimum cost path (minimum of capacity losses). As was already determined, the cause why capacity loss of the channel occurs, from technical perspective, is due to the fact that single-antenna installation is incapable of uninterrupted operation from several satellites simultaneously; ground station selects only one satellite at a point in time while the data of another satellite getting ignored. When paths costs are the same program starts checking their bit rates and distances. In such way program improves the performance of the network. Each node has parameters which characterize path to this node: the cost of path, distance, bit rate, earliest start time, earliest delivery time, name of parent node (predecessor), losses of the path, route flow and useless link. The cost of the node is the cost of path to this node.

In order to find out the optimal path that guarantees the minimum of network losses, earliest delivery time and maximum bit rate such components as optimal path finder, contact overlap finder and capacity loss calculator should be applied. First of all, the contact overlap finder extracts data from a contact plan. A contact is defined as an interval during which it is expected that data will be transmitted by DTN transmitting node and most or all of the transmitted data will be received by contact's receiving node. Knowing the contact plan of the whole network the overlapping contacts can be easily determined by contact overlap finder in the following way: Let first condition A means that start time of AOS and end time of LOS of the node $S1$ completely after start time of AOS and end time of LOS of the node $S2$ (Fig. 2).

|---- AOS, LOS of $S1$ -----|
|--- AOS, LOS of $S2$ -----|
(True if AOS ($S1$) > LOS ($S2$))

Figure 2 – Example 1. Non overlapping contacts

Let second condition B means that start time of AOS and end time of LOS of the node $S1$ is completely before start time of AOS and end time of LOS of the node $S2$ (Fig. 3).

|---- AOS, LOS of $S1$ -----|
|--- AOS, LOS of $S2$ -----|
(True if LOS ($S1$) < AOS ($S2$))

Figure 3 – Example 2. Non overlapping contacts

If neither $S1$ nor $S2$ is true – then overlap exists (Fig. 4).

AOS ($S1$) <= LOS ($S2$) and LOS ($S1$) >= AOS ($S2$)

Figure 4 – Example 3. Overlapping contacts

So using this statement contact overlap finder determines the presence of overlaps. But this is not enough because the goal is searching for paths with minimum capacity losses.

As was already determined completed bundle transmission or reception activity reduces the current capacity of the applicable throttle by the capacity consumption computed for that bundle. This reduction may cause the throttle's current capacity to become negative.

A satellite is said to be available for a specified interval of time, called its time window [12], if it is visible to the ground station during this interval; let we have: $[0, L], L > 0$ $M = \{S1, S2, ..., S_m\}$; a satellite is visible for a subinterval of the planning horizon: let $[st_i, end_i] \subseteq [0, L]$ be the visible for satellite $S_i, i = 1, 2, ..., m$ if $st_1 < st_2$ and $end_1 < end_2$ and $end_1 < st_2$ thus the overlap values between satellites $S1$ and $S2$ can be denoted in the following way:

$$Overlap_{S_1} = \frac{end_1 - st_2}{end_2 - st_2} \times 100, \quad (1)$$

$$Overlap_{S_2} = \frac{end_1 - st_2}{end_1 - st_1} \times 100 \quad (2)$$

© Babeniuk G. M., 2021
DOI 10.15588/1607-3274-2021-1-15

This overlap values present the value of capacity losses during overlap for overlapping nodes $S1$ and $S2$.

So capacity loss values for overlapping nodes provided by capacity loss calculator are used to find the path with minimum capacity losses. Thus, optimal path finder using information taken from contact overlap finder and capacity loss calculator calculates the optimal path. The base idea of the proposed approach is assignment of the highest priority to the path with maximum utility and minimum of capacity losses. In some cases utility value can be neglected, because in case of overlap, max flow value along the path where affected node locates reduces and it leads to severe network performance degradation (network losses), that's why should be avoided paths with overlapping contacts. This is actually the main reason why paths with minimum capacity losses and low utility are more preferable than paths with maximum of capacity losses and high utility. In case where paths don't have any overlapping contacts proposed approach selects path with earliest delivery time and maximum bit rate.

To find all possible routes in the contact plan, the load route list routine performs a series of Dijkstra's searches over a contact graph derived from the contact plan. Although complex algorithm, CGR is considered to be among the most mature strategies towards forwarding and routing in space DTNs. In this work, should be developed an algorithm that aims to find out the optimal path that guarantees the minimum of network losses, earliest delivery time and maximum bit rate and to perform scheduling of bundles in the network with overlapping contacts. The main idea of proposed approach can be presented by following example.

The main advantages of the proposed approach are following:

– Low scheduling overhead;
– Capacity losses control;
– Conceptually very simple.

The proposed approach is integrated into Dijkstra's algorithm to assure that the selected path is also the shortest one (has earliest delivery time).

After optimal path selection processing step № 6 activates. In the processing step № 6 proposed approach finds the max flow value in optimal path and compares this value with size of bundle. If size of bundle is bigger than max flow value then the following problem appears:

– proposed approach calculates the value of capacity losses with regard to the size of bundle, but in the case where some edge has a lack of bandwidth for transferring bundle of the given size then bundle cut-off happens and proposed approach can't calculate the cost of overlap for overlapping contacts with regard to the size of bundle anymore because the size of bundle was changed. In such situation proposed approach sets the size of bundle to the max flow value in optimal path and returns to step № 5 to calculate correct value of capacity losses for overlapping contacts. If size of bundle is less than max flow value in optimal path then proposed approach moves to processing step № 7. In the processing step № 7 proposed approach updates central contact plan. It should be noted that due to the application context of DTN, nodes are not able to synchronize their contact plans instantly (every node has an own contact plan independently of the contact plans of the other nodes). Our approach proposes central contact plan for all nodes. Before next bundle selection central contact plan should be updated because as was mentioned above completed bundle transmission activity and contact overlap reduce the current capacity of the applicable throttles by the capacity consumption computed for that bundle.

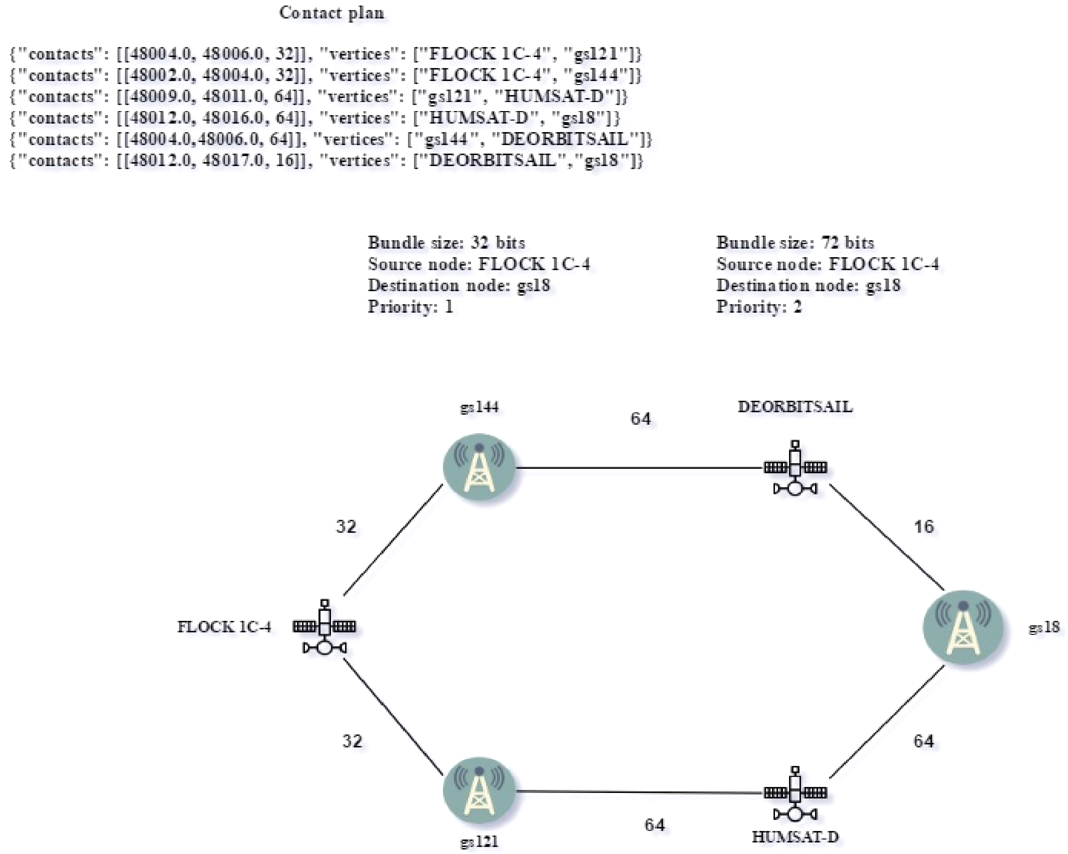
In processing step № 8 proposed approach finds the rest data. If size of bundle is bigger than optimal path max flow value then proposed approach performs the fragmentation. It is clear that in processing step № 8 the part of bundle that was smaller or equal to max flow value in optimal path was scheduled already, but if size of bundle is bigger than optimal path max flow value then proposed approach subtracts bundle size from max flow value in optimal path (fragmentation) and performs scheduling for the rest of bundle starting from processing step № 5. Thus, can be concluded that all algorithmic enhancements for network with overlapping contacts are configured to deliver bundles not only as early as possible but also with maximum favorable cost of performance. Proposed algorithm can be integrated into DTN protocol implementation software called "ION" (CGR - based protocols block) as far as its configuration environment is simulated in similar way and it provides solutions for problems arose in network with overlapping contacts.

4 EXPERIMENTS

In order to develop and test code Eclipse Integrated Development Envirinment was used, the code itself was written in Python.

The program implementation can be described with the following pictures (Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9, Fig. 10 and Fig. 11).

DOI 10.15588/1607-3274-2021-1-15



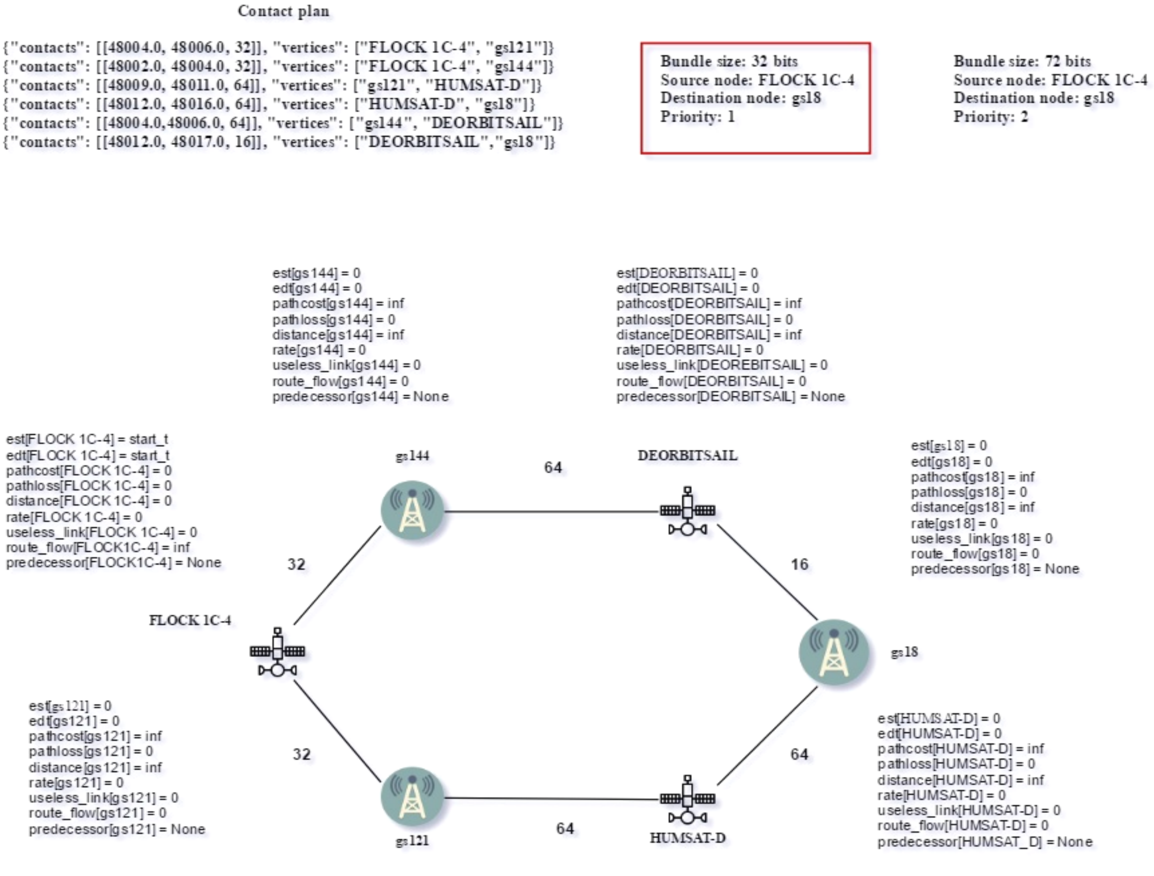
Contact plan

```
{"contacts": [[48004.0, 48006.0, 32]], "vertices": ["FLOCK 1C-4", "gs121"]}
{"contacts": [[48002.0, 48004.0, 32]], "vertices": ["FLOCK 1C-4", "gs144"]}
{"contacts": [[48009.0, 48011.0, 64]], "vertices": ["gs121", "HUMSAT-D"]}
{"contacts": [[48012.0, 48016.0, 64]], "vertices": ["HUMSAT-D", "gs18"]}
{"contacts": [[48004.0,48006.0, 64]], "vertices": ["gs144", "DEORBITSAIL"]}
{"contacts": [[48012.0, 48017.0, 16]], "vertices": ["DEORBITSAIL","gs18"]}
```

Bundle size: 32 bits
Source node: FLOCK 1C-4
Destination node: gs18
Priority: 1

Bundle size: 72 bits
Source node: FLOCK 1C-4
Destination node: gs18
Priority: 2

Figure 5 – Initial internal configuration



Contact plan

```
{"contacts": [[48004.0, 48006.0, 32]], "vertices": ["FLOCK 1C-4", "gs121"]}
{"contacts": [[48002.0, 48004.0, 32]], "vertices": ["FLOCK 1C-4", "gs144"]}
{"contacts": [[48009.0, 48011.0, 64]], "vertices": ["gs121", "HUMSAT-D"]}
{"contacts": [[48012.0, 48016.0, 64]], "vertices": ["HUMSAT-D", "gs18"]}
{"contacts": [[48004.0,48006.0, 64]], "vertices": ["gs144", "DEORBITSAIL"]}
{"contacts": [[48012.0, 48017.0, 16]], "vertices": ["DEORBITSAIL","gs18"]}
```

Bundle size: 32 bits
Source node: FLOCK 1C-4
Destination node: gs18
Priority: 1

Bundle size: 72 bits
Source node: FLOCK 1C-4
Destination node: gs18
Priority: 2

est[gs144] = 0
edt[gs144] = 0
pathcost[gs144] = inf
pathloss[gs144] = 0
distance[gs144] = inf
rate[gs144] = 0
useless_link[gs144] = 0
route_flow[gs144] = 0
predecessor[gs144] = None

est[DEORBITSAIL] = 0
edt[DEORBITSAIL] = 0
pathcost[DEORBITSAIL] = inf
pathloss[DEORBITSAIL] = 0
distance[DEORBITSAIL] = inf
rate[DEORBITSAIL] = 0
useless_link[DEOREBITSAIL] = 0
route_flow[DEORBITSAIL] = 0
predecessor[DEORBITSAIL] = None

est[FLOCK 1C-4] = start_t
edt[FLOCK 1C-4] = start_t
pathcost[FLOCK 1C-4] = 0
pathloss[FLOCK 1C-4] = 0
distance[FLOCK 1C-4] = 0
rate[FLOCK 1C-4] = 0
useless_link[FLOCK 1C-4] = 0
route_flow[FLOCK1C-4] = inf
predecessor[FLOCK1C-4] = None

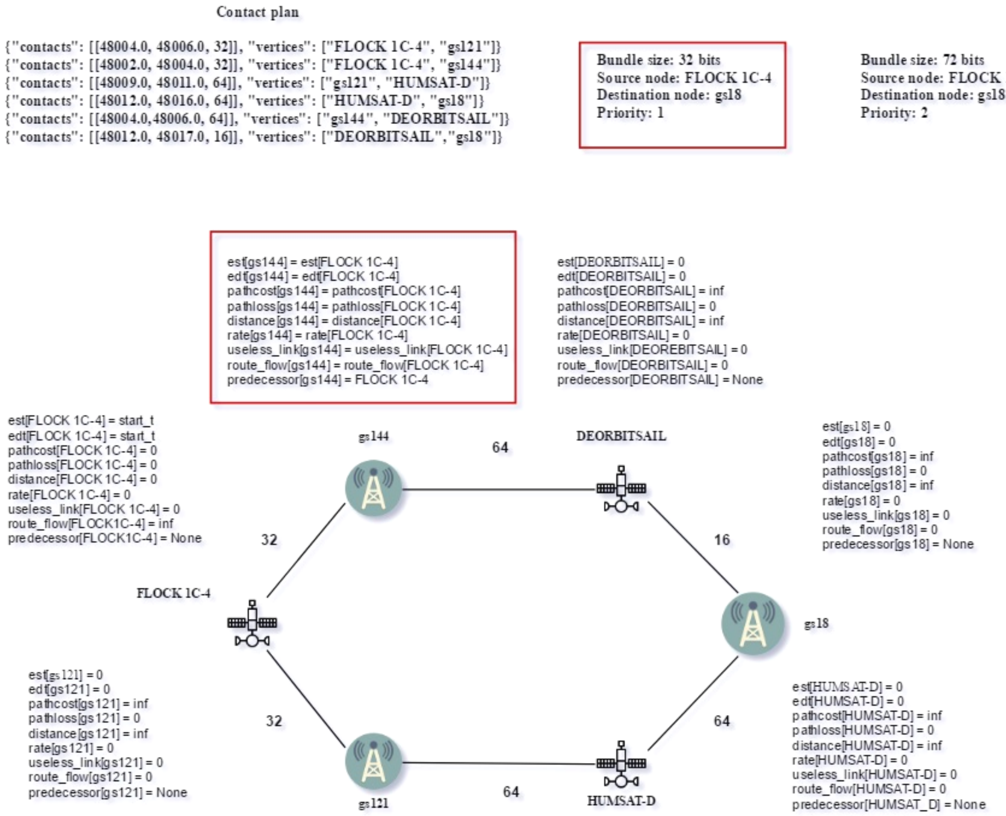
est[gs18] = 0
edt[gs18] = 0
pathcost[gs18] = inf
pathloss[gs18] = 0
distance[gs18] = inf
rate[gs18] = 0
useless_link[gs18] = 0
route_flow[gs18] = 0
predecessor[gs18] = None

est[gs121] = 0
edt[gs121] = 0
pathcost[gs121] = inf
pathloss[gs121] = 0
distance[gs121] = inf
rate[gs121] = 0
useless_link[gs121] = 0
route_flow[gs121] = 0
predecessor[gs121] = None

est[HUMSAT-D] = 0
edt[HUMSAT-D] = 0
pathcost[HUMSAT-D] = inf
pathloss[HUMSAT-D] = 0
distance[HUMSAT-D] = inf
rate[HUMSAT-D] = 0
useless_link[HUMSAT-D] = 0
route_flow[HUMSAT-D] = 0
predecessor[HUMSAT_D] = None

Figure 6 – Selection of the start node

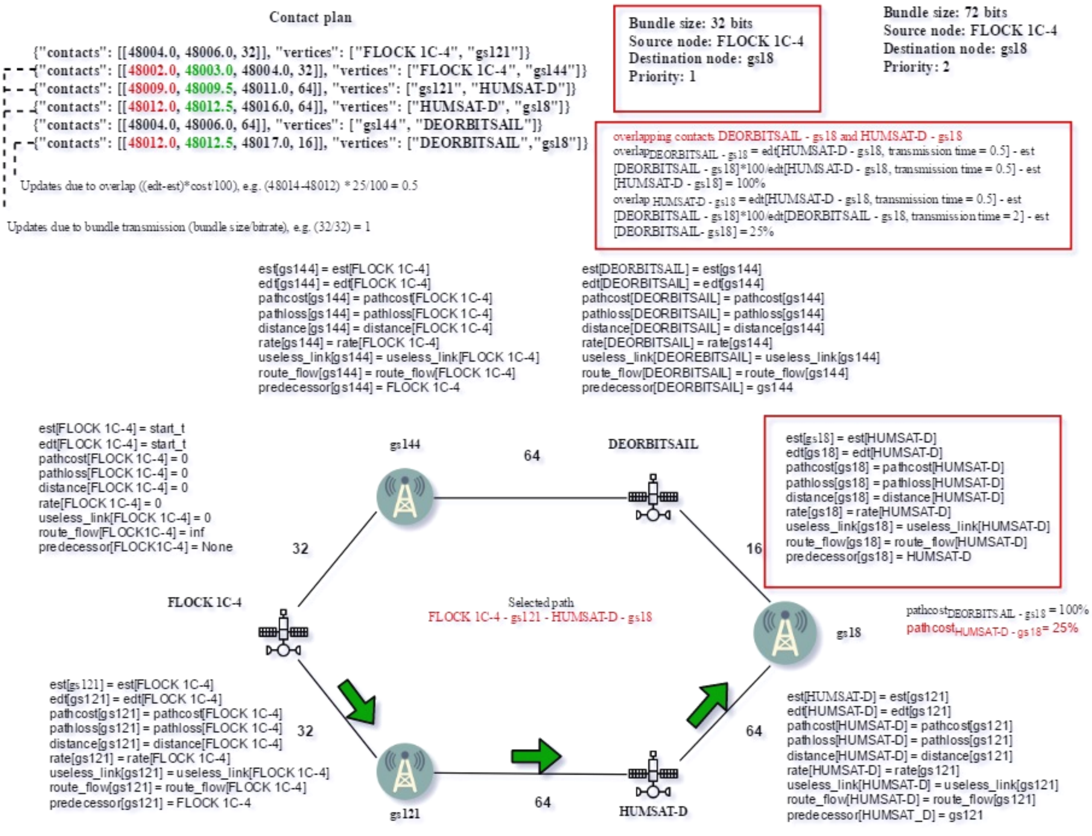
DOI 10.15588/1607-3274-2021-1-15



Contact plan

```
{"contacts": [[48004.0, 48006.0, 32]], "vertices": ["FLOCK 1C-4", "gs121"]}
{"contacts": [[48002.0, 48004.0, 32]], "vertices": ["FLOCK 1C-4", "gs144"]}
{"contacts": [[48009.0, 48011.0, 64]], "vertices": ["gs121", "HUMSAT-D"]}
{"contacts": [[48012.0, 48016.0, 64]], "vertices": ["HUMSAT-D", "gs18"]}
{"contacts": [[48004.0,48006.0, 64]], "vertices": ["gs144", "DEORBITSAIL"]}
{"contacts": [[48012.0, 48017.0, 16]], "vertices": ["DEORBITSAIL","gs18"]}
```

Figure 7 – Capacity losses calculation, optimal path selection and contact plan updates



Contact plan

```
{"contacts": [[48004.0, 48006.0, 32]], "vertices": ["FLOCK 1C-4", "gs121"]}
{"contacts": [[48002.0, 48003.0, 48004.0, 32]], "vertices": ["FLOCK 1C-4", "gs144"]}
{"contacts": [[48009.0, 48009.5, 48011.0, 64]], "vertices": ["gs121", "HUMSAT-D"]}
{"contacts": [[48012.0, 48012.5, 48016.0, 64]], "vertices": ["HUMSAT-D", "gs18"]}
{"contacts": [[48004.0, 48006.0, 64]], "vertices": ["gs144", "DEORBITSAIL"]}
{"contacts": [[48012.0, 48012.5, 48017.0, 16]], "vertices": ["DEORBITSAIL","gs18"]}
```

Figure 8 – Path selection for second bundle

DOI 10.15588/1607-3274-2021-1-15

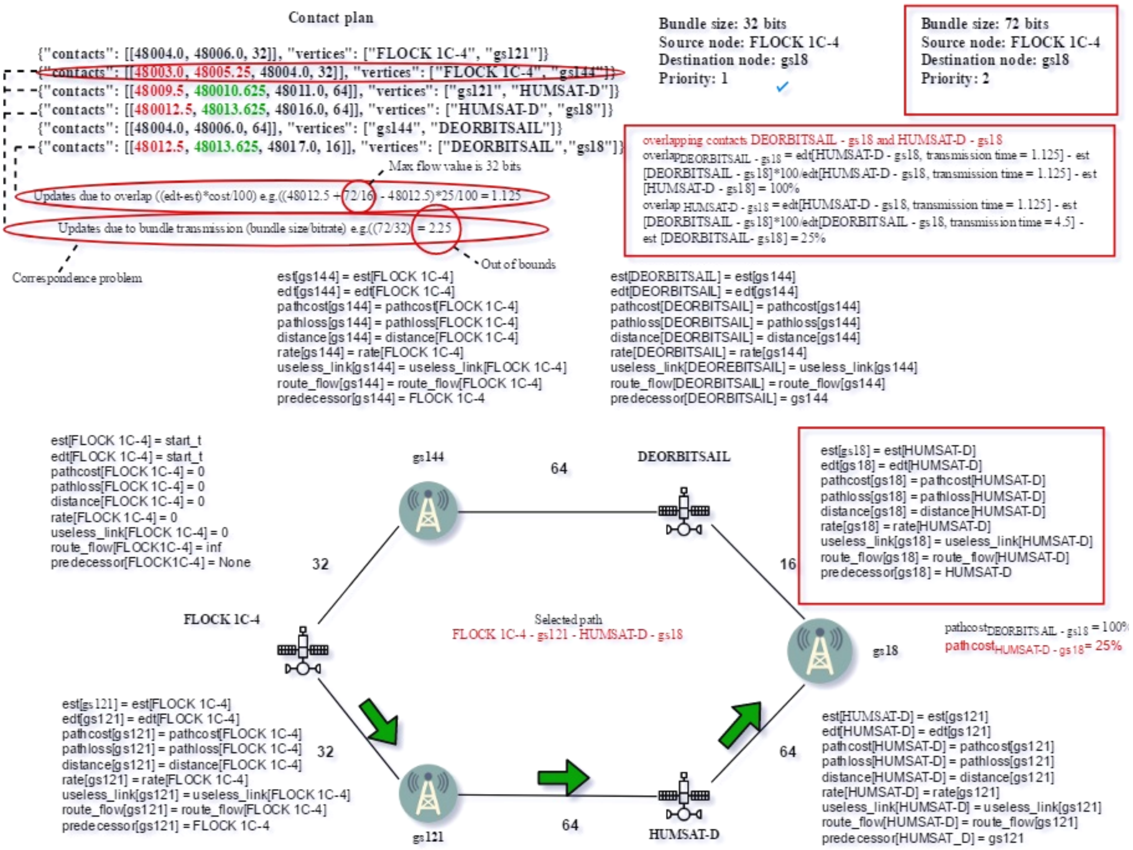


Figure 9 – Recalculation of the optimal path for second bundle

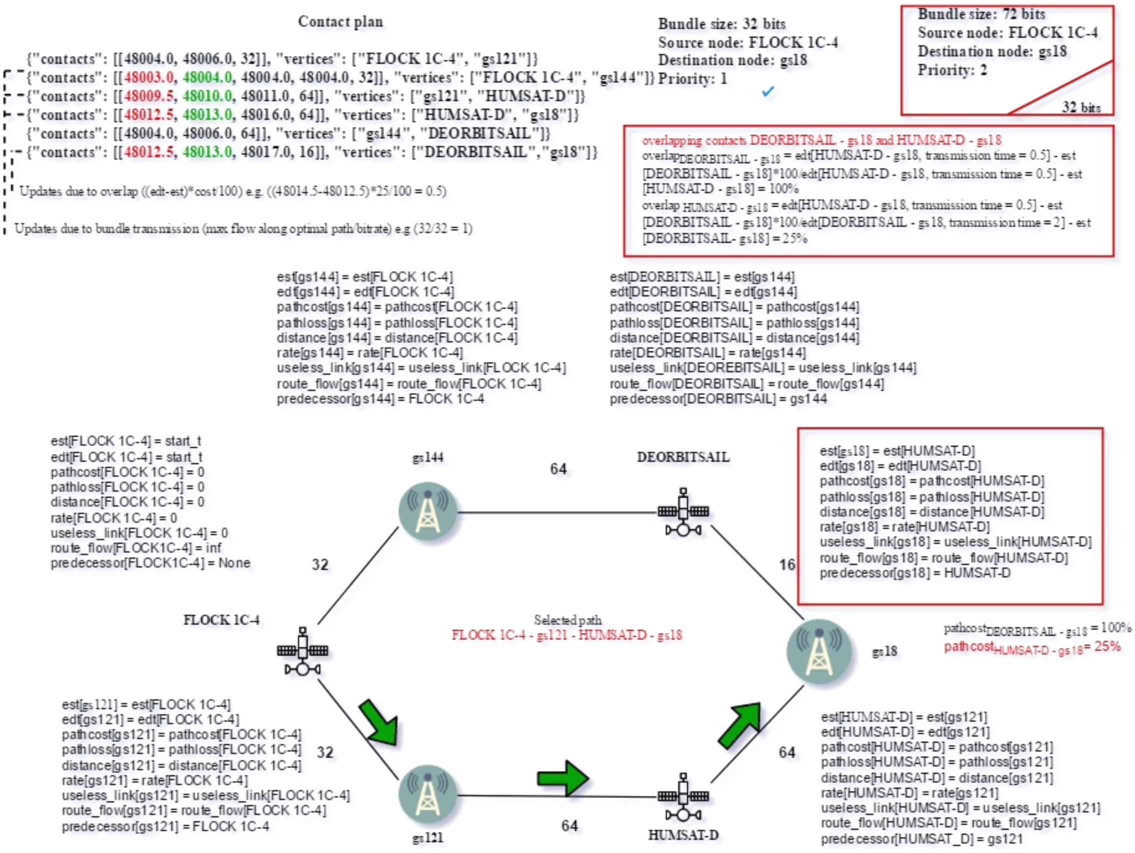
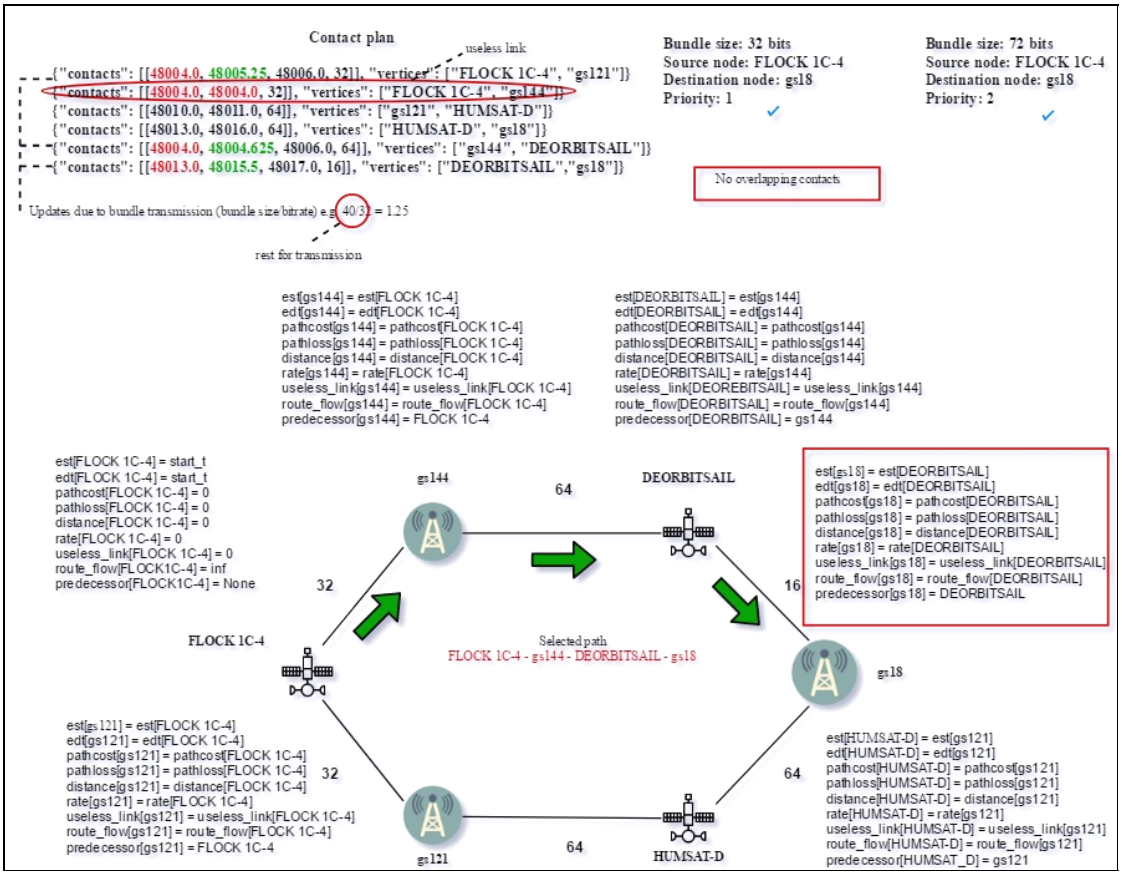


Figure 10 – Contact plan updates for second bundle

DOI 10.15588/1607-3274-2021-1-15



```
{"contacts": [[48004.0, 48005.25, 48006.0, 32]], "vertices": ["FLOCK 1C-4", "gs121"]}
{"contacts": [[48004.0, 48004.0, 32]], "vertices": ["FLOCK 1C-4", "gs144"]}
{"contacts": [[48010.0, 48011.0, 64]], "vertices": ["gs121", "HUMSAT-D"]}
{"contacts": [[48013.0, 48016.0, 64]], "vertices": ["HUMSAT-D", "gs18"]}
{"contacts": [[48004.0, 48004.625, 48006.0, 64]], "vertices": ["gs144", "DEORBITSAIL"]}
{"contacts": [[48013.0, 48015.5, 48017.0, 16]], "vertices": ["DEORBITSAIL","gs18"]}
```

Updates due to bundle transmission (bundle size/bitrate) e.g. 40/32 = 1.25

Bundle size: 32 bits
Source node: FLOCK 1C-4
Destination node: gs18
Priority: 1 ✓

Bundle size: 72 bits
Source node: FLOCK 1C-4
Destination node: gs18
Priority: 2 ✓

```
est[gs144] = est[FLOCK 1C-4]
edt[gs144] = edt[FLOCK 1C-4]
pathcost[gs144] = pathcost[FLOCK 1C-4]
pathloss[gs144] = pathloss[FLOCK 1C-4]
distance[gs144] = distance[FLOCK 1C-4]
rate[gs144] = rate[FLOCK 1C-4]
useless_link[gs144] = useless_link[FLOCK 1C-4]
route_flow[gs144] = route_flow[FLOCK 1C-4]
predecessor[gs144] = FLOCK 1C-4
```

```
est[DEORBITSAIL] = est[gs144]
edt[DEORBITSAIL] = edt[gs144]
pathcost[DEORBITSAIL] = pathcost[gs144]
pathloss[DEORBITSAIL] = pathloss[gs144]
distance[DEORBITSAIL] = distance[gs144]
rate[DEORBITSAIL] = rate[gs144]
useless_link[DEOREBITSAIL] = useless_link[gs144]
route_flow[DEORBITSAIL] = route_flow[gs144]
predecessor[DEORBITSAIL] = gs144
```

```
est[FLOCK 1C-4] = start_t
edt[FLOCK 1C-4] = start_t
pathcost[FLOCK 1C-4] = 0
pathloss[FLOCK 1C-4] = 0
distance[FLOCK 1C-4] = 0
rate[FLOCK 1C-4] = 0
useless_link[FLOCK 1C-4] = 0
route_flow[FLOCK1C-4] = inf
predecessor[FLOCK1C-4] = None
```

```
est[gs18] = est[DEORBITSAIL]
edt[gs18] = edt[DEORBITSAIL]
pathcost[gs18] = pathcost[DEORBITSAIL]
pathloss[gs18] = pathloss[DEORBITSAIL]
distance[gs18] = distance[DEORBITSAIL]
rate[gs18] = rate[DEORBITSAIL]
useless_link[gs18] = useless_link[DEORBITSAIL]
route_flow[gs18] = route_flow[DEORBITSAIL]
predecessor[gs18] = DEORBITSAIL
```

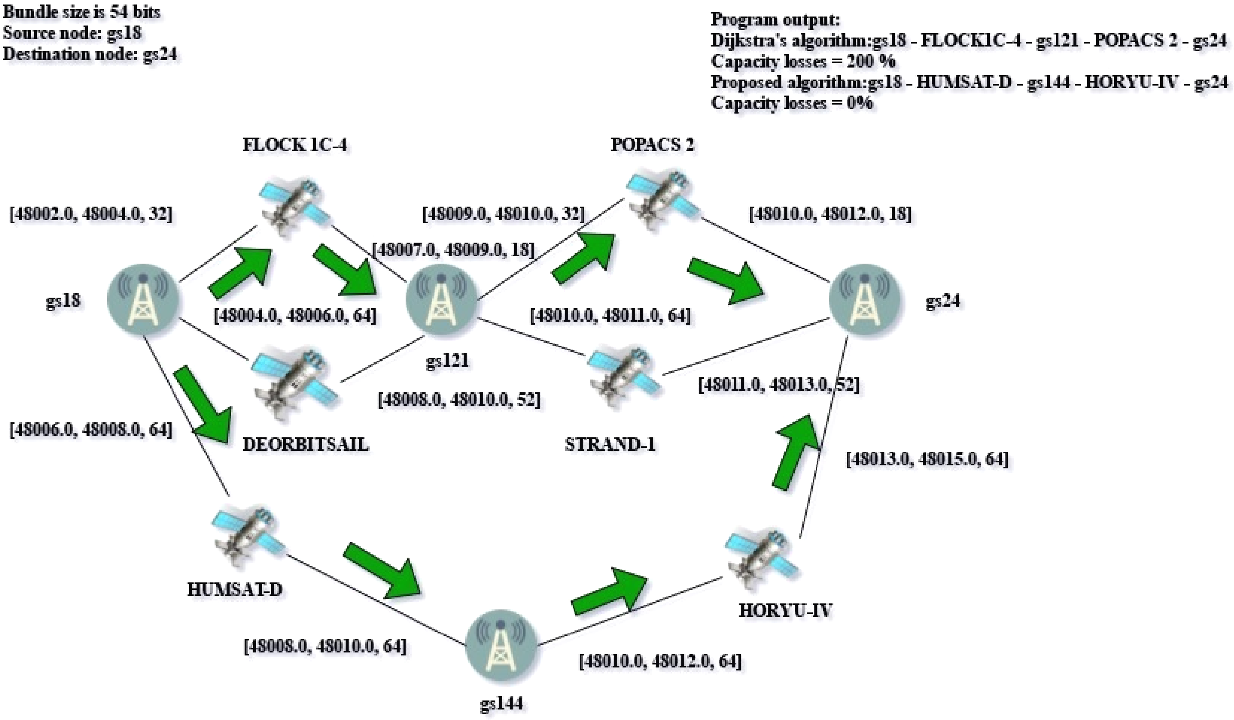
```
est[gs121] = est[FLOCK 1C-4]
edt[gs121] = edt[FLOCK 1C-4]
pathcost[gs121] = pathcost[FLOCK 1C-4]
pathloss[gs121] = pathloss[FLOCK 1C-4]
distance[gs121] = distance[FLOCK 1C-4]
rate[gs121] = rate[FLOCK 1C-4]
useless_link[gs121] = useless_link[FLOCK 1C-4]
route_flow[gs121] = route_flow[FLOCK 1C-4]
predecessor[gs121] = FLOCK 1C-4
```

```
est[HUMSAT-D] = est[gs121]
edt[HUMSAT-D] = edt[gs121]
pathcost[HUMSAT-D] = pathcost[gs121]
pathloss[HUMSAT-D] = pathloss[gs121]
distance[HUMSAT-D] = distance[gs121]
rate[HUMSAT-D] = rate[gs121]
useless_link[HUMSAT-D] = useless_link[gs121]
route_flow[HUMSAT-D] = route_flow[gs121]
predecessor[HUMSAT_D] = gs121
```

Figure 11 – Selection of optimal path for rest data

5 RESULTS

In this section experimental analysis of behavior of the proposed algorithm will be examined (Scenario 1, Scenario 2).

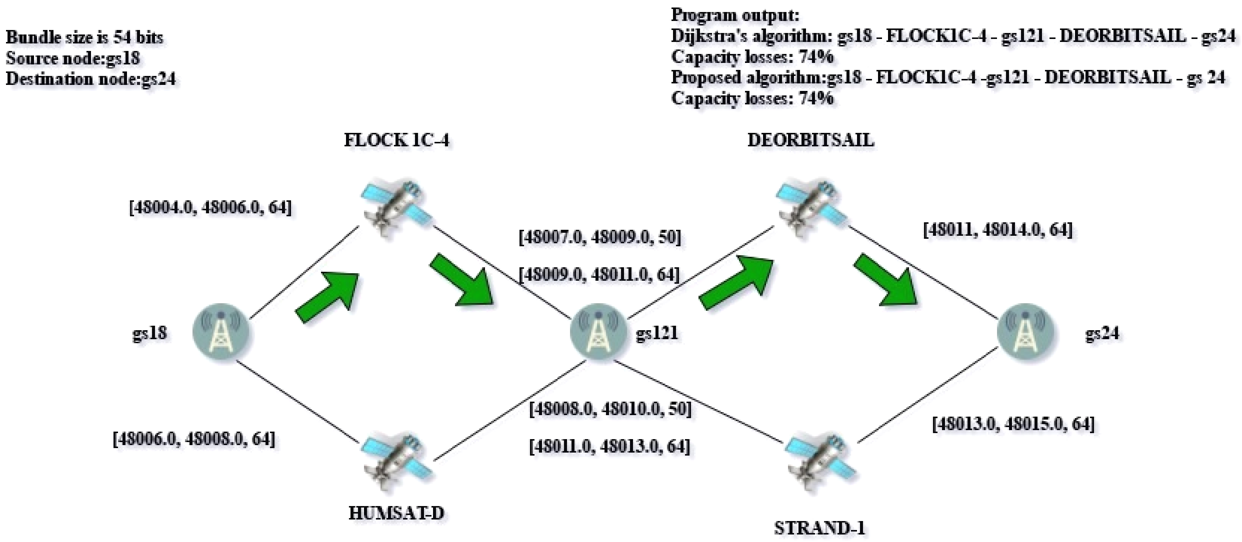


Bundle size is 54 bits
Source node: gs18
Destination node: gs24

```
Program output:
Dijkstra's algorithm:gs18 - FLOCK1C-4 - gs121 - POPACS 2 - gs24
Capacity losses = 200 %
Proposed algorithm:gs18 - HUMSAT-D - gs144 - HORYU-IV - gs24
Capacity losses = 0%
```

Scenario 1 – Network topology

© Babeniuk G. M., 2021
DOI 10.15588/1607-3274-2021-1-15



```
Bundle size is 54 bits
Source node:gs18
Destination node:gs24
```

```
Program output:
Dijkstra's algorithm: gs18 - FLOCK1C-4 - gs121 - DEORBITSAIL - gs24
Capacity losses: 74%
Proposed algorithm:gs18 - FLOCK1C-4 -gs121 - DEORBITSAIL - gs 24
Capacity losses: 74%
```

Scenario 2 – Network topology

In the first scenario the following network topology will be evaluated (Scenario 1).

Network architecture (data flow) based on the central contact plan (vertices are nodes and edges are node paths: contacts + vertices), thus given a graph which represents a flow network where every edge has a capacity. After simulation given topology program provides the following output:

– Dijkstra's algorithm:

[['gs18', 'FLOCK1C-4', 0], ['FLOCK1C-4', 'gs121', 100], ['gs121', 'POPACS 2', 100], ['POPACS 2', 'gs24', 200]].

– Proposed routing algorithm:

[['gs18', 'HUMSAT-D', 0], ['HUMSAT-D', 'gs144', 0], ['gs144', 'HORYU-IV', 0], ['HORYU-IV', 'gs24', 0]].

Dijkstra's algorithm selects the path with taking into account earliest delivery time and available link capacity and as a result the following disadvantages arise.

In Scenario 2 network architecture (data flow) based on the central contact plan (vertices are nodes and edges are node paths: contacts + vertices), thus given a graph which represents a flow network where every edge has a capacity. After simulation given topology program provides the following output:

– Dijkstra's algorithm:

[['gs18', 'FLOCK1C-4', 0], ['FLOCK1C-4', 'gs121', 74], ['gs121', 'DEORBITSAIL', 74], ['DEORBITSAIL', 'gs24', 74]].

– Proposed algorithm:

[['gs18', 'FLOCK1C-4', 0], ['FLOCK1C-4', 'gs121', 74], ['gs121', 'DEORBITSAIL', 74], ['DEORBITSAIL', 'gs24', 74]].

6 DISCUSSION

Regarding Scenario 1. As was already discussed DTN routing protocols (CGR-based protocols) does not take into account the overlap factor and resulting capacity losses. In presented example there are overlapping contacts such as: FLOCK1C-4 – gs121 & DEORBITSAIL – gs121 & POPACS 2 – gs24 & STRAND-1 – gs24 in gs18 – FLOCK1C-4 – gs121 – POPACS 2 – gs24 path.

As far as single-antenna installation is incapable to maintain uninterrupted operations from several satellites simultaneously and support of multiple installations is expensive; ground station selects only one satellite at a point in time while the data of another satellite getting ignored. Thus, capacity losses of gs18 – DEORBITSAIL – gs121 – STRAND-1 – gs24 path are ignored. In comparison with algorithm used in CGR-based protocols proposed algorithm selects another path by taken into account overlapping contacts. Selected path has slightly worse earliest delivery time but in given scenario given path is optimal one. Thus, can be concluded that path selected by proposed algorithm is optimal due to following reasons:

– Bundle is forwarded without losses (high bit rate).

– Transmission delay in comparison with Dijkstra's algorithm is negligible small.

Regarding Scenario 2. In the given example proposed algorithm selects the same path as Dijkstra's algorithm.

As was discussed when capacity losses of both paths are same then proposed algorithm selects path with minimum transmission delay and high bit rate.

In the given example capacity losses due to overlapping contacts such as: FLOCK1C-4 – gs121 & HUMSAT-D – gs121 of both paths are the same that's why proposed algorithm selected the same path as Dijkstra's algorithm. Thus, can be concluded that really proposed algorithm selects not only paths with minimum of capacity losses but also paths with minimum transmission delay and high bit rate.

7 CONCLUSIONS

Proposed optimization algorithm finds the path with minimum capacity losses, in the least amount of time with the highest delivery ratio relative to the available

DOI 10.15588/1607-3274-2021-1-15

data rates and performs bundles scheduling. This algorithm is based on improved network management model that includes: improved Dijkstra's algorithm, Ford-Fulkerson's algorithm, central contact plan proposed in this work. The proposed model allowed us to schedule bundles with minimum capacity losses. The major enhancements obtained from this algorithm were related to four main aspects. First, was allowed to use improved Contact Plan that provides up-to-date information only due to global knowledge of nodes (nodes update contact plan very often), this representation is more efficient than in CGR-based protocols. Second, was performed bundle scheduling using Ford-Fulkerson's algorithm by computing max flow in flow network, it allowed performing bundles fragmentation in cases where it was necessary to avoid unnecessary data losses.

Third, was computed the effectiveness of the paths based on their capacity losses, data rate, delivery time (distance). So to say, selected path fulfills the following requirements:

– Maximum flow;

The proposed principle of capacity losses calculation provides not only the path with minimum capacity losses but also the path of highest rate, maximum flow and minimum distance.

– Minimum bundle transmission time;

When paths costs are the same (capacity losses); program starts checking their bit rates and distances (Dijkstra's algorithm).

– Maximum data rate;

Priority-aware logic used in this work provides good performance (reduces transmission delay – rate control).

Finally, was computed the availability of the selected path by checking max flow value in selected route. It allowed performing bundle fragmentation in cases where it was necessary to avoid unnecessary data losses. Thus, can be concluded that proposed algorithm choose the route that guarantees the minimum of capacity losses, earliest delivery time and maximum bit rate in the network with overlapping contacts and increases the probability of successful data transmission between magnetic stations and magnetic observatories. To validate this work, the implementation of algorithm was integrated inside environment similar to CGR protocol and was shown that proposed algorithm can be efficiently used in the network with overlapping contacts.

The scientific novelty is this article for the first time was presented DTN as solution of CENS problems connected with accuracy of magnetic field maps. For the first time was presented an improvement of existing algorithms of data transmission of DTN.

As it was described shown now the system chooses the route with minimum of capacity losses, i.e the route with earliest delivery time and maximum bit rate; instead of choosing the route with 200% of looses it chooses the route with 0% of looses.

The practical significance is developed improvement of existing algorithms of data transmission of DTN in Python.

© Babeniuk G. M., 2021
DOI 10.15588/1607-3274-2021-1-15

Prospects for the further research are integration of developed software into real DTN protocol presented by National Aeronautics and Space Administration and integration of DTN into magnetic field maps creation software.

ACKNOWLEDGEMENTS

The work is supported by the scientific Phd work done in National Aviation University called "Cartographic support of CENS".

REFERENCES

1. Perdaen D., Armbruster D., Kempf K., Lefeber E. Controlling a re-entrant manufacturing line via the push-pull point, *International Journal of Production Research,* 2008, № 46 (16), pp. 4521–4536. DOI: 10.1080/00207540701258051.
2. Armbruster D., Ringhofer C., Jo T. J. Continuous models for production flows, *American Control Conference. Boston, MA, USA*, 2004, No. 5, pp. 4589–4594. DOI: 10.1109/ACC.2004.182675
3. Schmitz J. P., Beek D. A., Rooda J. E. Chaos in Discrete Production Systems? *Journal of Manufacturing Systems*, 2002, No. 21 (3), pp. 236–246. DOI: 10.1016/S0278-6125(02)80164-9
4. Bramson M. Stability of queueing networks, lecture notes in mathematics, *Journal of Probability Surveys,* 2008, No. 5, pp. 169–345. DOI: 10.1214/08-PS137. https://projecteuclid.org/euclid.ps/1220879338
5. Tian F., Willems S. P., Kempf K. G. An iterative approach to item-level tactical production and inventory planning, *International Journal of Production Economics*, 2011, No. 133, pp. 439–450. DOI: 10.1016/j.ijpe.2010.07.011
6. Chankov S., Hütt M., Bendul J. Influencing factors of synchronization in manufacturing systems, *International Journal of Production Research*, 2018, No. 56 (14), pp. 4781–4801. DOI: 10.1080/00207543.2017.1400707
7. Kempf K. Simulating semiconductor manufacturing systems: successes, failures and deep questions, *Proceedings of the 1996 Winter Simulation Conference, Institute of Electrical and Electronics Engineers. – Piscatawa.* New Jersey, 1996, pp. 3–11. DOI: 10.1109/WSC.1996.873254
8. Bitran G., Haas E., Hax A. Hierarchical production planning: a two-stage system, *Operations Research*, 1982, No. 30, pp. 232–251. DOI: 10.1287/opre.30.2.232
9. Pihnastyi O., Khodusov V. D. Optimal Control Problem for a Conveyor-Type Production Line, *Cybern. Syst. Anal.*, 2018, No. 54(5), pp. 744–753. DOI: 10.1007/s10559-018-0076-2
10. Pihnastyi O. M., Yemelianova D., Lysytsia D. Using pde model and system dynamics model for describing multi-operation production lines, *EasternEuropean Journal of Enterprise Technologies*, 2020, No. 4 (4(106)), pp. 54–60. DOI: 10.15587/17294061.2020.210750
11. Wonham W., Cai K., Rudie K. Supervisory Control of Discrete-Event Systems: A Brief History – 1980-2015, *20th IFAC World Congress IFAC-PapersOnLine*, 2017, No. 50(1), pp. 1791–1797. DOI: 10.1016/j.ifacol.2017.08.164
12. Gross D., Harris C. M. Fundamentals of Queueing Theory. New York, 1974, 490 p.

Received 15.11.2020.
Accepted 22.01.2021.

УДК 681.5 (048)

ВИКОРИСТАННЯ МЕРЕЖІ СТІЙКОЇ ДО РОЗРИВІВ ДЛЯ СТВОРЕННЯ ВИСОКОТОЧНИХ МАГНІТОМЕНТРИЧНИХ КАРТ

Бабенюк Г. М. – аспірантка, факультет аеронавігації Національного авіаційного університету, Київ, Україна.

АНОТАЦІЯ

Актуальність. Основним призначенням кореляційно-екстремальної навігаційної системи є пошук координат у разі відсутності сигналу глобальної системи позиціонування, і як результат високоточні карти як основне джерело інформації для пошуку координат дуже важливі. Магнітометрична карта як основне джерело інформації може включати похибки, на приклад: недостатньо добре обладнання або людський фактор може спричинити похибку вимірювань.

Мета роботи. З метою створення високоточних карт дана робота пропонує вдосконалити процес створення магнітометричних карт. Дана робота представляє мережу стійку до розривів як додатковий підхід до передачі даних між магнітною обсерваторією та магнітною станцією а також її вдосконалення.

Метод. Покращений алгоритм Дейкстрі разом з алгоритмом Форда-Фалкерсона що використовуються для пошуку шляху з мінімальними втратами даних, найшвидшого та з максимальною швидкістю передачі даних у разі перекриття контактів повинні бути представлені в даній роботі, оскільки в наш час протоколи мережі стійкої до розривів не враховують факт перекриття контактів і це призводить до великих проблем.

Результати. Вперше буде представлений алгоритм, який вибирає маршрут, що гарантує мінімум втрат даних є найшвидшим та має максимальну швидкістю передачі даних в мережах стійких до розривів з перекриттям контактів та збільшує ймовірність успішної передачі даних між магнітними станціями та магнітними обсерваторіями.

Висновки. Для проведення високоточних вимірювань магнітного поля група людей розміщує своє обладнання для вимірювання магнітного поля у віддалених районах, щоб уникнути вплив навколишнього середовища на показники магнітометра. Оскільки величина магнітного поля може змінюватися в залежності від температури, відстані до океану, широти (добової зміни магнітного поля) та магнітних бур, магнітна станція час від часу коригує свої виміри за допомогою еталонних значень магнітного поля (магнітна станція надсилає запит на еталонні значення до магнітної обсерваторії). Проблема даного підходу полягає в тому, що віддалені райони, як правило, не охоплені мережею (немає Інтернету), і в результаті коригування вимірювань є неможливим. Для того, щоб зробити можливим коригування вимірювань і, як результат, підвищити точність магнітних карт, в даній роботі пропонується використовувати мережу стійку до розривів, яка забезпечить доступ до Інтернету в різних регіонах світу та її вдосконалення, щоб зробити даний підхід ще кращим.

Отримані результати публікуються вперше.

КЛЮЧОВІ СЛОВА: кореляційно-екстремальна навігаційна система, алгоритм Форда-Фалкерсона, магнітометричні карти, алгоритм Дейкстрі.

УДК 681.5 (048)

ИСПОЛЬЗОВНИЕ СЕТИ УСТОЙЧИВОЙ К РАЗРЫВАМ ДЛЯ СОЗДАНИЯ ВЫСОКОТОЧНЫХ МАГНИТОМЕТРИЧЕСКИХ КАРТ

Бабенюк А. Н. – аспирантка, факультет аэронавигации Национального авиационного университета, Киев, Украина.

АННОТАЦИЯ

Актуальность. Основным назначением корреляционно-экстремальной навигационной системы является поиск координат при отсутствии сигналов глобальной системы позиционирования, и как результат высокоточные карты в качестве основных источников информации для поиска координат важны. Магнитометрическая карта как основной источник информации может включать погрешности, например: недостаточно хорошо оборудования или человеческий фактор может вызвать погрешность измерений.

Цель работы. С целью создания высокоточных карт данная работа предлагает усовершенствовать процесс создания магнитометрическая карт. Данная работа представляет сеть устойчивую к разрывам как дополнительный подход к передаче данных между магнитной обсерваторией и магнитной станцией а также ее усовершенствование.

Метод. Улучшенный алгоритм Дейкстры вместе с алгоритмом Форда-Фалкерсона что используются для поиска пути с минимальными потерями данных, быстрого и с максимальной скоростью передачи данных в случае перекрытия контактов должны быть представлены в данной работе, поскольку в наше время протоколы сети устойчивой к разрывам не учитывают факт перекрытия контактов и это приводит к большим проблемам.

Результаты. Впервые будет представлен алгоритм, который выбирает маршрут, что гарантирует минимум потерь данных является самым быстрым и имеет максимальную скорость передачи данных в сетях устойчивых к разрывам с перекрытием контактов и увеличивает вероятность успешной передачи данных между магнитными станциями и магнитными обсерваториями.

Выводы. Для проведения высокоточных измерений магнитного поля группа людей размещает свое оборудование для измерения магнитного поля в отдаленных районах, чтобы избежать влияние окружающей среды на показатели магнитометра. Поскольку величина магнитного поля может меняться в зависимости от температуры, расстояния до океана, широты (суточного изменения магнитного поля) и магнитных бурь, магнитная станция время от времени корректирует свои измерения с помощью эталонных значений магнитного поля (магнитная станция посылает запрос на эталонные значения к магнитной обсерватории). Проблема данного подхода заключается в том, что отдаленные районы,

DOI 10.15588/1607-3274-2021-1-15

как правило, не охвачены сетью (нет Интернета), и в результате корректировка измерений является невозможной. Для того, чтобы сделать возможным корректировку измерений и, как результат, повысить точность магнитных карт, в данной работе предлагается использовать сеть устойчивую к разрывам, которая обеспечит доступ в Интернету в различных регионах мира и ее усовершенствование, чтобы сделать данный подход еще лучшим. Полученные результаты публикуются впервые.

КЛЮЧЕВЫЕ СЛОВА: корреляционно-экстремальная навигационная система, алгоритм Форда-Фалкерсона, магнитометрические карты, алгоритм Дейкстри.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Perdaen D. Controlling a re-entrant manufacturing line via the push-pull point. / D. Perdaen, D. Armbruster, K. Kempf, E. Lefeber // International Journal of Production Research. – 2008. – № 46 (16). – P. 4521–4536. DOI: 10.1080/00207540701258051.
2. Armbruster D. Continuous models for production flows / D. Armbruster, C. Ringhofer, T. J. Jo // American Control Conference. Boston, MA, USA. – 2004. – № 5. – P. 4589–4594. DOI: 10.1109/ACC.2004.182675
3. Schmitz J. P. Chaos in Discrete Production Systems? / J. P. Schmitz, D. A. Beek, J. E. Rooda // Journal of Manufacturing Systems. – 2002. – №21 (3). – P. 236–246. DOI: 10.1016/S0278-6125(02)80164-9
4. Bramson M. Stability of queueing networks, lecture notes in mathematics / M. Bramson // Journal of Probability Surveys. – 2008. – № 5. – P. 169–345. DOI: 10.1214/08-PS137. https://projecteuclid.org/euclid.ps/1220879338
5. Tian F. An iterative approach to item-level tactical production and inventory planning / F. Tian, S. P. Willems, K. G. Kempf // International Journal of Production Economics, – 2011. – № 133. – P. 439–450. DOI: 10.1016/j.ijpe.2010.07.011
6. Chankov S. Influencing factors of synchronization in manufacturing systems / S. Chankov, M. Hütt, J. Bendul // International Journal of Production Research. – 2018. – № 56 (14). – P. 4781–4801. DOI: 10.1080/00207543.2017.1400707
7. Kempf K. Simulating semiconductor manufacturing systems: successes, failures and deep questions / K. Kempf // Proceedings of the 1996 Winter Simulation Conference, Institute of Electrical and Electronics Engineers. – Piscataway, New Jersey. – 1996. – P. 3–11. DOI: 10.1109/WSC.1996.873254
8. Bitran G. Hierarchical production planning: a two-stage system / G. Bitran, E. Haas, A. Hax // Operations Research. – 1982. – № 30. – P. 232–251. DOI: 10.1287/opre.30.2.232
9. Pihnastyi O. Optimal Control Problem for a Conveyor-Type Production Line / O. M. Pihnastyi, V. D. Khodusov // Cybern. Syst. Anal. – 2018. – № 54(5). – P.744–753. DOI: 10.1007/s10559-018-0076-2
10. Pihnastyi O. M. Using pde model and system dynamics model for describing multi-operation production lines / O. Pihnastyi, D. Yemelianova, D. Lysytsia // EasternEuropean Journal of Enterprise Technologies. – 2020. – №4 (4(106)). – P. 54–60. DOI: 10.15587/17294061.2020.210750
11. Wonham W. Supervisory Control of Discrete-Event Systems: A Brief History – 1980-2015 / W. Wonham, K. Cai, K. Rudie// 20th IFAC World Congress IFAC-PapersOnLine. –2017. – № 50(1). – P. 1791–1797. DOI: 10.1016/j.ifacol.2017.08.164
12. Gross D. Fundamentals of Queueing Theory / D. Gross, C. M. Harris. – New York, 1974. – 490 p.

DOI 10.15588/1607-3274-2021-1-15