**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2022 Т. 14 № 4 С. 983–993**
DOI: 10.20537/2076-7633-2022-14-4-983-993

СПЕЦИАЛЬНЫЙ ВЫПУСК

УДК: 004.023

О подходе к разработке и валидации алгоритмов маршрутизации на разрывных сетях

А. А. Федоров[a], И. В. Сошилов[b], В. Н. Логинов[c]

Московский физико-технический институт,
Россия, 141701, Московская область, Долгопрудный, Институтский переулок, д. 9

E-mail: [a] fedorov@phystech.edu, [b] soshilov@phystech.edu, [c] loginov.vn@mipt.ru

Получено 25.12.2021, после доработки — 31.01.2022.
Принято к публикации 02.02.2022.

В данной статье рассматривается проблема централизованного планирования маршрутов передачи данных в сетях, устойчивых к задержкам и разрывам. Исходная проблема расширяется дополнительными требованиями к хранению узлов и процессу связи. Во-первых, предполагается, что связь между узлами графа устанавливается с помощью антенн. Во-вторых, предполагается, что каждый узел имеет хранилище конечной емкости. Существующие работы не рассматривают и не решают задачу с этими ограничениями. Предполагается, что заранее известны информация о сообщениях, подлежащих обработке, информация о конфигурации сети в указанные моменты времени, взятые с определенными периодами, информация о временных задержках для ориентации антенн для передачи данных и ограничения на объем хранения данных на каждом спутнике группировки. Два хорошо известных алгоритма — CGR и Earliest Delivery with All Queues — модифицированы для удовлетворения расширенных требований. Полученные алгоритмы решают задачу поиска оптимального маршрута в сети, устойчивой к разрывам, отдельно для каждого сообщения. Также рассматривается проблема валидации алгоритмов в условиях отсутствия тестовых данных. Предложены и апробированы возможные подходы к валидации, основанные на качественных предположениях, описаны результаты экспериментов. Проведен сравнительный анализ производительности двух алгоритмов решения задачи маршрутизации. Два алгоритма, названные RDTNAS-CG и RDTNAS-AQ, были разработаны на основе алгоритмов CGR и Earliest Delivery with All Queues соответственно. Оригинальные алгоритмы были значительно расширены и была разработана дополненная реализация. Валидационные эксперименты были проведены для проверки минимальных требований «качества» к правильности алгоритмов. Сравнительный анализ производительности двух алгоритмов показал, что алгоритм RDTNAS-AQ на несколько порядков быстрее, чем RDTNAS-CG.

Ключевые слова: DTN, алгоритмы на графах, комбинаторные алгоритмы, количественная валидация

COMPUTER RESEARCH AND MODELING
2022 VOL. 14 NO. 4 P. 983–993
DOI: 10.20537/2076-7633-2022-14-4-983-993

SPECIAL ISSUE

UDC: 004.023

Augmented data routing algorithms for satellite delay-tolerant networks. Development and validation

A. A. Fedorov[a], I. V. Soshilov[b], V. N. Loginov[c]

Moscow Institute of Physics and Technology,
9 Institute lane, Moscow, 141701, Russia

E-mail: [a] fedorov@phystech.edu, [b] soshilov@phystech.edu, [c] loginov.vn@mipt.ru

Received 25.12.2021, after completion — 31.01.2022.
Accepted for publication 02.02.2022.

The problem of centralized planning for data transmission routes in delay tolerant networks is considered. The original problem is extended with additional requirements to nodes storage and communication process. First, it is assumed that the connection between the nodes of the graph is established using antennas. Second, it is assumed that each node has a storage of finite capacity. The existing works do not consider these requirements. It is assumed that we have in advance information about messages to be processed, information about the network configuration at specified time points taken with a certain time periods, information on time delays for the orientation of the antennas for data transmission and restrictions on the amount of data storage on each satellite of the grouping. Two well-known algorithms — CGR and Earliest Delivery with All Queues are improved to satisfy the extended requirements. The obtained algorithms solve the optimal message routing problem separately for each message. The problem of validation of the algorithms under conditions of lack of test data is considered as well. Possible approaches to the validation based on qualitative conjectures are proposed and tested, and experiment results are described. A performance comparison of the two implementations of the problem solving algorithms is made. Two algorithms named RDTNAS-CG and RDTNAS-AQ have been developed based on the CGR and Earliest Delivery with All Queues algorithms, respectively. The original algorithms have been significantly expanded and an augmented implementation has been developed. Validation experiments were carried to check the minimum «quality» requirements for the correctness of the algorithms. Comparative analysis of the performance of the two algorithms showed that the RDTNAS-AQ algorithm is several orders of magnitude faster than RDTNAS-CG.

Keywords: delay tolerant networks, graph routing, combinatorial algorithms, qualitative validation

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 4, pp. 983–993.

Introduction

In this paper, we consider the problem of planning data transmission routes in delay tolerant networks (DTN) — ones that are resistant to interruptions [Fall, 2003].

Compared to traditional networks, which are represented as distance graphs with a function on the edges in the form of the connection speed between nodes, the following features can be distinguished in DTN:

1) the possibility of disconnection;
2) nonconstant data transfer rate;
3) the limited memory size of storages on the nodes.

These features are inherent, in particular, in space data transmission systems, including artificial satellites and ground stations. The discontinuity of the connections and the variability of the data transfer rate are caused by changes in the relative position of the satellites and their visibility conditions.

Due to the instability of graph connectivity caused by these circumstances, the application of traditional algorithms for finding the shortest path on graphs [Shortest path problem], particularly in the problem of route planning in space satellite systems [Fedorov, Soshilov, Loginov, 2020], faces great difficulties in practical implementation, although at the same time this task becomes increasingly relevant especially in relation to the rapidly developing space constellations for the global Internet and remote sensing of the Earth.

There are a number of works in this area devoted to the development of routing algorithms in DTN networks.

The work [Jain, Fall, Patra, 2004] formulated the problem of routing networks that allow delays, in which messages must move from one node to another through a time-varying connectivity graph, under the assumption that the dynamics of the connectivity changes are known in advance. The paper proposed an approach for evaluating routing algorithms in such networks, developed the algorithms themselves and evaluated their performance. The authors have found out that algorithms which had used the least amount of knowledge about network topology had shown weaker results than algorithms with more complete knowledge. As an algorithm that uses the most complete knowledge, the authors cite the linear programming algorithm. This algorithm optimizes routes for all messages at once, whereas the weaker algorithms optimize for each subsequent message without changing the result for the previous ones. It was shown that one of the main disadvantages of this algorithm is its high computational complexity.

The paper [Burleigh, 2010] solves a similar problem of forming a message delivery route from one node of a space satellite constellation to another. For this purpose the original Contact Graph Routing (CGR) algorithm was developed. Its main feature is that it runs on all nodes and works in real time: as soon as a node receives a data bundle, the algorithm chooses a node to redirect it to the next for the fastest delivery to the destination node. This takes into account the current location of all satellites at each time.

The considered algorithms are a good basis for solving the problem of finding the optimal data transmission path in a satellite constellation, but they do not take into account the following features that arise due to technical limitations on satellite equipment:

1) satellites have antennas which require some time to perform their orientation in space to establish communication with the next satellite in the data transmission route;
2) satellites have data storages with limited size each, which may lead to storage overflow and to disruptions in data transmission.

None of the algorithms considered in [Jain, Fall, Patra, 2004; Burleigh, 2010] takes into account these features. In addition, the CGR algorithm [Burleigh, 2010] is designed to work in real time on a multi-agent network, which does not correspond to the expected mode of operation of the required solution that is a central planner with full prior knowledge of the satellite configuration and potential connections.

Problem statement

In this paper, we consider the problem of finding a path that is optimal in terms of delivery time for each individual message on a network with nonpermanent connections. It is assumed that we have in advance information about messages to be processed, information about the network configuration at specified time points taken with certain time periods, information on time delays for the orientation of the antennas for data transmission and restrictions on the amount of data storage on each satellite of the grouping.

It is assumed that the connection between the nodes of the graph is established using antennas. One antenna can only communicate with the only antenna at the same time. It takes some time to reconfigure the antenna, this time — delay time — is fixed for each antenna.

Connections are specified for each pair of antennas of two different nodes on the graph. Connection represents a time interval in which information (messages, data bundles) may be transmitted between these nodes through these antennas. For each connection the following parameters are set: the delay time, the data transfer rate (Mb/s), and the cost of transmitting a bit of information (W/bit).

Connection means the ability to provide a contact between nodes. In this case, the list of links may include competing links, for example, using the same antenna at the same time. The algorithm being developed must ensure that at any given time each antenna was used to transmit information over no more than one link.

Since satellites have limited storage capacities, each node in the network can hold a limited amount of information at any given time. The amount of memory equal to the bundle size is allocated when bundle receiving starts and is reallocated after the bundle transmission to the next satellite ends.

It is known in advance on which node the message appears (the source node) and its size (in MB), as well as which nodes it must be delivered to — the target nodes.

Changes to the network infrastructure, specifically, nodes and their features, including storage capacity and antenna configuration and connections between nodes and their features, including delay and data transfer rate, are known in advance. Since the developed solution is supposed to be used at a planning stage rather than as a real-time simulator, in sake of approach simplicity changes to network infrastructure are required not to interfere with message delivery in time, i. e., no network infrastructure change is supposed to happen during processing of a message.

In the work, the following task is set: to build a delivery route for each message from a source node to one of the target nodes with minimal delivery time (difference between the receiving time at the target node and the appearance time at the source node), given the above constraints.

Materials and methods

As a result of the conducted research, two algorithms for solving the formulated problem were developed. They both are an upgrade of the previously considered CGR algorithm [Burleigh, 2010] and Earliest Delivery with All Queries [Jain, Fall, Patra, 2004], with the limitations on storage capacity and reconfiguring antennas delay.

This paper presents comparative results of the implementation of these algorithms for solving the task and develops an approach to their validation.

RDTNAS-AQ

The algorithm works with a classical graph, in which the nodes are objects that store and transmit messages, and the edges store information about connections between the corresponding nodes. The algorithm was named RDTNAS-AQ — Routing in a Delay-Tolerant Network with Antennae and Storages via All Queue. Dijkstra's algorithm (see Figure 1) is used as a basis.

```
Input: $G = (V, E)$, $s$, $T$, $w(e, t)$
Output: $L$
$Q \leftarrow \{V\}$
$L[s] \leftarrow 0$, $L[v] \leftarrow \infty\ \forall v \in V$ s.t. $v \neq s$.
while $Q \neq \{\}$ do
    Let $u \in Q$ be the node s.t. $L[u] = \min_{x \in Q} L(x)$
    $Q = Q \leftarrow \{u\}$
    for each edge $e \in E$, s.t. $e = (u, v)$ do
        if $L[v] > (L(u) + w(e, L[u] + T))$ then
            $L[v] \leftarrow L[u] + w(e, L[u] + T)$
        end if
    end for
end while
```

Figure 1. Dijkstra's algorithm

The function w sets the weight of the edge depending on time. In fact, it should answer the question: how long will the transmission take, considering gaps, delays and bandwidth changes. The value s is defined as the closest moment in time of message delivery. This value is determined by integrating the residual edge bandwidth function, whose prototype is w. Multiple edges are created between the nodes for each pair of antennas that can communicate. A piecewise-constant bandwidth function is associated with each edge. This function is represented as sorted time ranges and bandwidth values. When a message route is found, it is committed by making the necessary changes to the appropriate bandwidth functions.

Piecewise constant functions are also used to take into account the limitations on the operation of data stores and antenna directivity.

An orientation log is stored for each antenna. The orientation log consists of records with the following content: the ID of the satellite to which the antenna is directed, the start time and the end time. The log is filled in such a way that the records do not conflict and the delay required for antenna redirection is considered.

It is worth addressing the question of route lookup step by step. At each iteration, a vertex that has not been processed yet is selected, the distance to which is minimal, and its processing is performed. In the context of the problem, distance means the time by which a message can be delivered to a particular node in the graph. Vertex processing consists of updating the distance to other unprocessed vertices. Below the sequence of actions for updating the distance to other vertices is described in detail.

1. Assume that the processed node has identifier i and the node to which the distance is updated has identifier j.

2. Assume that $dist[k]$ is the moment of time at which the message reaches the node k. Assume that t is the earliest possible moment for starting the transmission. Initially $t = dist[i]$. The distance to the node at which the message appears initially is equal to the moment of the appearance time. The algorithm iterates all edges between nodes i and j.

3. The residual edge bandwidth function is used to pick such a time range $[t;\ t+dt]$ that the message gets delivered to node j at time $t + dt$. In case the time range cannot be picked, the algorithm skips the edge.

4. The sufficiency of residual storage capacity for the source node i and the receiver node j is ensured. For source node the time range requiring the ability to store the message is $[dist[i];\ t + dt]$. For the receiver node this time range is $[t;\ t+dt]$. In case the capacity is not sufficient, the earliest possible transmission time t is shifted at least by 1 unit and step 3 is repeated.

5. The possibility of directing the antennas of the source and the receiver nodes is ensured. Assume that the antenna redirection delay is equal to z. It is mandatory to ensure the following:
   - the source antenna is undirected in the time range $[t;\ t + dt]$;
   - the source antenna is either directed to the receiver node or undirected in the time range $[t - z;\ t + dt + z]$.

6. The possibility described in step 5 above should also be ensured for the receiver antenna.

7. In case either step 5 or 6 cannot be satisfied, the earliest possible transmission time t is shifted at least by 1 unit and step 3 is repeated.

8. $dist[j] = \min(dist[j],\ t + dt)$.

After one of the target nodes turns out to be the source node of the Dijkstra algorithm iteration, the algorithm stops and the route is committed: records are made in the antenna orientation logs, residual storage capacities and the residual edge bandwidths.

RDTNAS-CG

The algorithm described below is an interpretation of the CGR algorithm, from which Routing in a delay-tolerant network with antennas and storage via Contact graphs, the RDTNAS-CG algorithm, borrowed its name.

At its core, RDTNAS-CG is the use of Dijkstra’s algorithm [Cormen et al., 2006] on the contact graph.

Like CGR, RDTNAS-CG works with a contact graph represented as follows: the vertex (node) of the graph is a directed contact between two nodes of the space constellation. The attributes of the graph node are: the ID of the transmitting node, the ID of the receiving node, the start and end time of the contact between them (the beginning and end of a possible connection between them). An edge of the graph connects two nodes of the graph if the nodes are matched, that is, the receiving node of the first contact is the transmitting node in the second contact, and the start time of the first contact does not exceed the end time of the second contact. There can be several edges between two vertices of the contact graph. It is important to note that there are no nodes with completely matching attributes in the contact graph.

Each contact (graph node) is mapped to a set of connections between the corresponding satellite constellation nodes.

The contact graph is built for each individual data bundle (message). Based on the message attributes (the time of appearance on the satellite, ID of the satellite constellation node generating the data packet, IDs of the target satellite constellation nodes), additional graph vertices are constructed:

1) fictitious source: the identifier of the sending node is equal to the ID of the receiving node, and is equal to the ID of satellite constellation node generating a data packet; the start time is the time the bundle appears; the end time is set to infinity;

2) fictitious receiver: the identifier of the transmitting node is equal to the identifier of the receiving node and is set to an identifier that is not represented in the satellite constellation nodes (in our implementation it was set one more than the maximum existing identifier among all objects); the contact start time is set to minus infinity; the contact end time is set to infinity.

3) multiple «helping» receivers: transmitting node ID — ID of the target satellite constellation node for a given data bundle (one «helping» contact is created for each target node); receiving node ID is the fictitious receiver; start time — minus infinity; end time — infinity. Such receivers help to identify undelivered messages.

Herewith it is assumed that fictitious transmission through additional nodes of the graph occurs instantly and without any energy consumption.

Every contact graph node is assigned with a tentative distance value — the estimated arrival time of the data bundle to the corresponding vertices (by default, this value is set to infinity, except for a fictitious source, for them it is equal to the appearance time of the data bundle) or the total calculated delivery energy of the data bundle to the corresponding vertices (this value is set to infinity, except for a fictitious source, for them it is zero). The estimated arrival time is calculated as the ratio of the message size to the data transmission rate of the corresponding connection (rounded up), the total calculated energy is calculated as the product of the message volume and the bit transmission cost of the corresponding connection.

The Dijkstra algorithm is used to find the route that is optimal in terms of arrival time or calculated delivery energy. Antenna availability, delays and node storage capacity restrictions are handled in the same way as described above in RDTNAS-AQ algorithm.

At the end of the algorithm processing, the distance value for the fictitious receiver is equal to the shortest time it takes for the data bundle to reach the target satellite constellation node from the source node (the shortest time among all the target nodes), or, similarly, the lowest calculated delivery energy.

At the end of the RDTNAS-CG algorithm, the shortest path is restored and the target node with the lowest delivery time or energy is determined.

Validation of the algorithms

The problem considered here is a combinatorial type problem. For such problems, as a rule, there are no complete test data sets. Sufficient and necessary conditions for the correctness of such algorithms have to be formulated based on the features of the problem, as well as the requirements for the input data and the logic of the solution. Below are several possible algorithms validation techniques and an analysis of their application to both algorithms is outlined.

Analysis of the impact of message splitting on the result of the algorithms

Experiment description

The experiment consists of several independent measurements taken under slightly different conditions. At each measurement, except for the first one, messages which exceed the specified limitation for this measurement are split into messages of a smaller size.

For example, in case the initial message size equals 128 and the limitation equals 63, two messages of size 63 each and one message of size 2 are produced as the result of the split. There are 128 measurements in the experiment. The number of each measurement varies from 1 to 128 and corresponds to the limitation of the message part size to which the split occurs. Bandwidth for all connections is equal to 8. The bandwidth is chosen equal for all connections to clearly depict the dependence of transmission time on the remainder of integer division and highlight the importance of proper choice of scale for time and speed. Thus, we both iterate through message sizes several time higher than the bandwidth and those which are comparative and even less than the bandwidth.

The messages obtained as the result of the split are marked to refer back to the original messages. The statistics on the average message lifetime and actual message transmission time are averaged over the original unsplitted messages.

Consider an example of a group of three messages of size 63, 63 and 2, respectively, obtained as the result of splitting a message of size 128. For the average message lifetime the maximal value is taken as average original message lifetime, as well as for actual message transmission time.

Experimental data visualization

The data obtained as a result of the experiments is presented below in three graphs. A mark at the *X*-axis corresponds to a specific measurement in the experiment and indicates the limitation of the message size.

The *Y*-axis is the average message lifetime in the measurement, the average actual message transmission time in the measurement, and the ratio of undelivered messages in the measurement, respectively, at each of the three graphs.

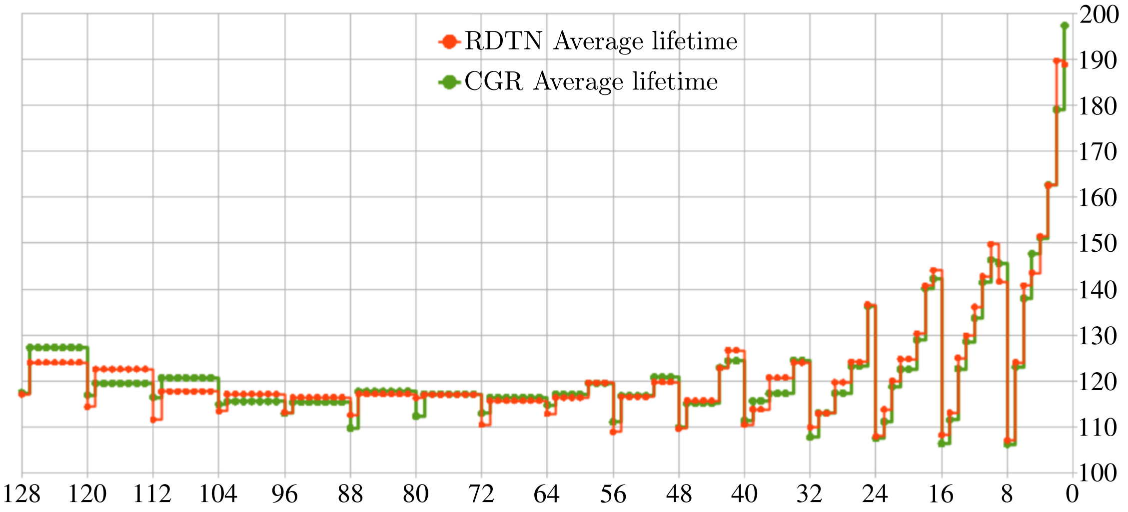

Figure 2. Average message lifetime

Legend for Figure 2.

1. *X* axis is the maximum message size; all larger messages are split.
2. *Y* axis is the message lifetime. If the message was split into parts, the maximum among the parts is taken and is considered the lifetime of the original message.
3. RDTN average lifetime is the average message lifetime when processed by the RDTNAS-AQ algorithm.
4. CGR average lifetime is the average message lifetime when processed by the RDTNAS-CG algorithm.

Legend for Figure 3.

1. *X* axis is the maximum message size; all larger messages are split.

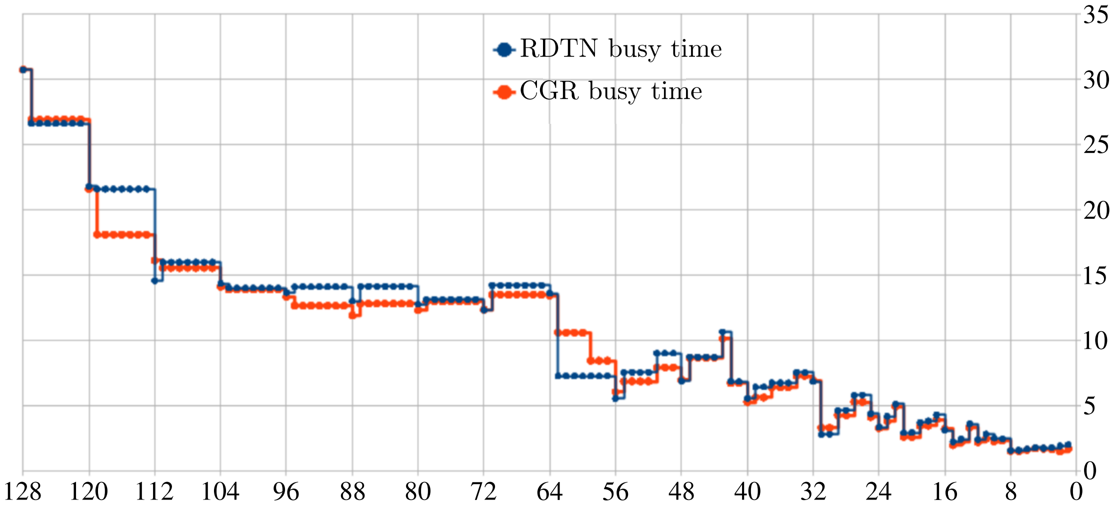

Figure 3. Average actual message transmission time

2. *Y* axis is the time of actual message transmission (excluding time spent in storages). If the message has been split into parts, the maximum is taken among the parts and is considered the time of the actual transmission of the original message.

3. RDTN busy timeis the average time of actual message transmission when processed by the RDTNAS-AQ algorithm.

4. CGR busy time is the average time of actual message transmission when processed by the RDTNAS-CG algorithm.

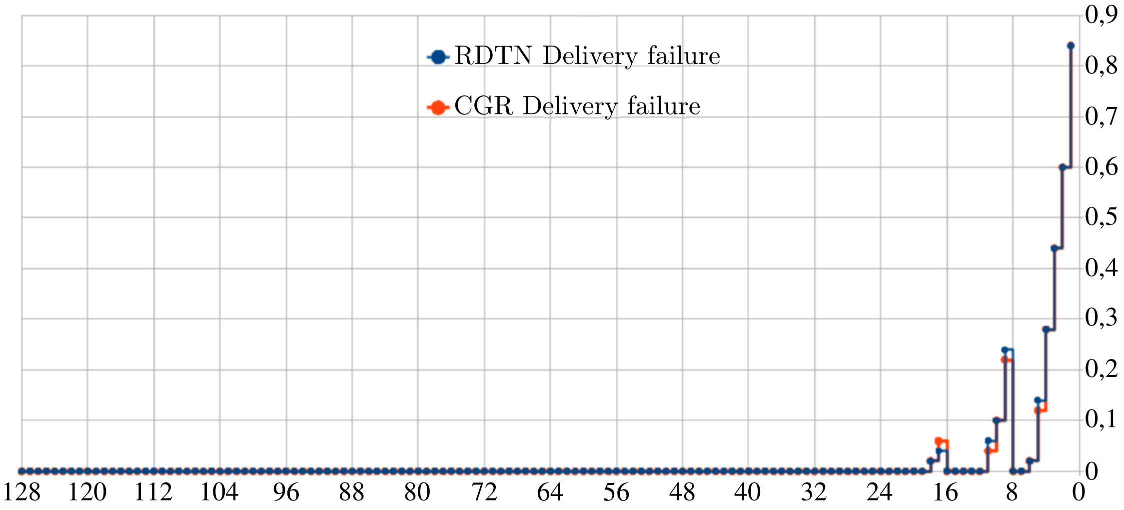

Figure 4. Ratio of undelivered messages (delivery failure)

Legend for Figure 4.

1. *X* axis is the maximum message size; all larger messages are split.

2. *Y* axis is the ratio of undelivered messages. If the message was split into parts and one of the parts was not delivered, then the entire message is considered undelivered.

3. RDTN delivery failure is the ratio of undelivered messages when processed by the RDTNAS-AQ algorithm.

4. CGR delivery failure is the ratio of undelivered messages when processed by the RDTNAS-CG algorithm.

Experiment results and interpretation

Both algorithms show similar results for the depicted metrics. The more fragmented the messages are, the less the actual transmission of the original message lasts (see Figure 3).

The message lifetime is not subject to a drastical decrease (see Figure 2), which can be explained by the sparseness of the graph. In other words, the message often waits for subsequent sending for a considerable amount of time.

It also makes no sense to take the message size less than typical transmission rate due to the fact that algorithms do not work with fractional time. Anomalies in average lifetime and delivery ratio appear when the message size limit reaches a value around 16 (see Figures 2, 4).

The graphs show that the average lifetime and delivery failure indicators begin to deviate greatly from the average values in the vicinity of the values when the message size limit approaches the typical transmission bandwidth. The network's ability to deliver messages is largely determined by the ratio of the average message size to the average bit rate. This parameter can be used for diagnostic purposes to determine network capabilities.

Analysis of the effect of scaling the problem on the result of the algorithms

This approach involves changing the input data in small increments and keeping track of the output changes, based on the assumption that in case the implementation of the algorithm and the algorithm itself are correct, then the output data should not change significantly, and if the output data changed greatly, then this may indicate the presence of incorrectness.

One of the implementations of such approach to algorithm validation is described below. Choose a coefficient k by which the following input characteristics are multiplied:

1) message size,

2) transmission bandwidth,

3) storage capacity.

Coefficient k is varied and the following output characteristics are tracked:

1) the proportion of changed message routes,

2) the average message delivery time (lifetime),

3) the proportion of undelivered messages.

The experiment showed that scaling up the input according to the conditions described above does not affect any of the tracked output characteristics, i. e., both algorithms behave as expected. The experiment was carried out with both zero and non-zero delay, which was not scaled.

Comparison of the algorithms on random data

To compare the performance of the algorithms, random test data was generated. Here are the characteristics of one of the tests and the comparison results.

A test was generated with 95 nodes and 4548 connections between them. It was required to build routes for 57 600 messages. Both algorithms found routes for 57 560 messages, and the undelivered messages are the same in both cases.

RDTNAS-AQ performance was 3000 times faster (average 2,5 seconds) than RDTNAS-CG (average 7300 seconds). RDTNAS-AQ found shorter (by 6 %) routes for 623 messages (1 % of the total number of messages). RDTNAS-CG found shorter (by 2,12 %) routes for 189 messages (0,3 % of the total number of messages).

Conclusion

This paper covers the problem of finding a path that is optimal in terms of delivery time for each individual message on a network with nonpermanent connections, with nodes characterized by finite storage capacity and antennas for establishing connections.

Two algorithms named RDTNAS-CG and RDTNAS-AQ have been developed based on the CGR and RDTN-AQ algorithms, respectively. The original algorithms have been significantly expanded and an augmented implementation has been developed.

Validation experiments were carried to check the minimum «quality» requirements for the correctness of the algorithms. Comparative analysis of the performance of the two algorithms showed that the RDTNAS-AQ algorithm is several orders of magnitude faster than RDTNAS-CG.

The purpose of further research is to develop methods for verifying algorithms in the absence of test data.

References

Burleigh S. Contact Graph Routing (draft-burleigh-dtnrg-cgr-01). — Laboratory, California Institute of Technology, 2010.

Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. — 2nd ed. — MIT Press and McGraw–Hill, 2006. — P. 595–601.

Fall K. A Delay-Tolerant Network Architecture for Challenged Internets // ACM SIGCOMM. — 2003.

Fedorov A. A., Soshilov I. V., Loginov V. N. Heuristic algorithms for finding routes for data transmission in satellite systems and their validation // Proceedings of MIPT. — 2020. — Vol. 12, No. 3 (47). — P. 64–77.

ION-DTN. — SourceForge.net. — https://sourceforge.net/projects/ion-dtn/

Jain S., Fall K., Patra R. Routing in a delay tolerant network // ACM SIGCOMM Computer Communication Review. — 2004. — Vol. 34, No. 4. — P. 145. — DOI: 10.1145/1030194.1015484

Shortest path problem. — Wikipedia. — https://en.wikipedia.org/wiki/Shortest_path_problem