УДК 316.776

G. Jaber, N. V. Patsei
Belarusian State Technological University

SEMANTIC BASED INFORMATION-CENTRIC NETWORKING OUTING ALGORITHMS

The article addressed the problem of routing in the field of Information-Centric Networking where a new semantic-based scheme is proposed to solve the obstacles facing IP networks. The paper represents the structure of routing tables for Semantic Information-Centric Networking (SICN) 3D-addressing: Semantic-Pub ID, Geo-Pub ID, Semantic-Geo. Four different scenarios based on information content type are used. Three algorithms for the 3D-address routing table and two for the cache ae presented: Router Record Update Algorithm, Cache TTL Update algorithm, Geo-Pub ID Garbage Collector algorithm, Semantic Garbage Collector algorithm, Cache Garbage Collector algorithm. This algorithms used for adding, removing and merging records in routing tables and based on Time to Live parameter. The article described semantic, geographical and publisher ID matching. Based on the developed algorithms, modeling was performed for four scenarios of network operation by metric Time Delay, Flooding or Traffic, and Efficiency Reuse factor for data. Compared with other ICN projects. The effectiveness of SICN and it's routing algorithms was shown.

Key words: algorithms, information-centric networks, semantic, routers table, address, garbage collection, cache, time to leave.

For citation: Jaber G., Patsei N. V., Semantic based Information-Centric Networking routing algorithms. *Proceedings of BSTU, issue 3, Physics and Mathematics. Informatics*, 2022, no. 2 (260), pp. 108–113.

Г. Джабер, Н.В. Пацей
Белорусский Государственный Технологический Университет

АЛГОРИТМЫ МАРШРУТИЗАЦИИ ИНФОРМАЦИОННО-ОРИЕНТИРВАННОЙ СЕТИ, ОСНОВАННОЙ НА СЕМАНТИКЕ

В статье рассматриваются вопросы маршрутизации данных в информационно-ориентированной сети, в которой была предложена новая схема именования, основанная на семантике для решения существующих проблем передачи информации в IP сетях. В работе изложены структуры таблиц маршрутизации сематической информационно-ориентированной сети с 3D-адресацией, названные: Semantic-Pub ID, Geo-Pub ID, Semantic-Geo. Они используются для различных сценариев в зависимости от типа передаваемого сетевого контента. Представлены алгоритмы 3D-адресации в таблицах маршрутизации и алгоритм кеширования: Router Record Update Algorithm, Cache TTL Update algorithm, Geo-Pub ID Garbage Collector algorithm, Semantic Garbage Collector algorithm, Cache Garbage Collector algorithm. Данные алгоритмы используются для добавления, удаления и объединения записей в таблицах на основе параметра Time to Live. В статье описывается поиск на основе сематического, географического и уникального идентификатора владельца. На основе разработанных алгоритмов было выполнено моделирование производительности сети для четырех сценариев сетевого взаимодействия. Производительность оценивалась на основе следующих метрик: Time Delay, Flooding (Traffic) и фактора Efficiency Reuse. Проведено сравнение с другими проектами информационного-ориентированных сетей. Показана эффективность алгоритмов маршрутизации семантической сети.

Ключевые слова: алгоритмы, информационно-ориентированные сети, семантика, таблицы маршрутизации, адрес, сборка мусора, кэш, время жизни.

Для цитирования: Jaber G., Patsei N. V., Semantic based Information-Centric Networking routing algorithms // Труды БГТУ. Сер. 3, Физика-математические науки и информатика. 2022. № 2 (260). С. 108–113.

Introduction. Every year the number of Internet users and the use (volume) of content is constantly increased. There is a growing need for security and support for the mobility of information sources. To solve the problem of Internet network name resolution, *Content Distribution Networks* (*CDN*) [1], peer-to-peer file exchange systems (*BitTorrent*) have been proposed for extracting content from multiple sources, replicating and distributing content. These mechanisms helped improve access to content, but did not improve the underlying network topology and therefore did not optimize network delivery performance.

The search for new approaches to organizing architectures led to the creation of *Information-Centric Networking* (*ICN*) [1]. For *ICN*, location-independent naming and name-based routing are the main concept. Content in the *ICN* is named directly, published, copied to the cache, and finally queried by name using a lookup or fetch primitive. The process of data searching through an arbitrary broadcast of a request, and the result will be an information object from any host that has a copy of the data.

Based on the main principles of *ICN*, in the last decade, data network projects have been developed that differ in routing algorithms, naming schemes, and deployment models: Data-Oriented Network Architecture (DONA, Berkeley University), Publish-Subscribe Internet Technology (PURSUIT), Publish-Subscribe Internet Routing Paradigm (PSIRP), Scalable and Adaptive Internet solutions (SAIL), 4WARD, Content Mediator architecture for content-aware networks (COMET), CONVERGENCE, Named Data Networking (NDN), Content Centric Networking (CCN), Mobility First (NSF Mobility First project, online), ANR Connect, Knowledge-Based Networking (KBN), Network of Information (NetInf), Context-Aware Green ICN Model (CAGIM).

There are four types of object naming in *ICN* – hierarchical (used in CCN, NDN), flat (used in NetInf, MobilityFirst and DONA project), attribute-based and hybrid [2, 3]. Most of the considered *ICN* designs are not suitable for all types of information object data transfer between publishers/subscribers. That's why *Semantic Information-Centric Networking* (*SICN*) was proposed [4]. It is *ICN*-based architecture with its own naming scheme which included three types of addresses: semantic, geographic and publisher [4–8].

It was designed *SICN* header format and made the classification of *SICN* data and requests (named: *R1*, *R2*, *R3*, *R4*), according to it, into four types based on publishers/subscribers' number and frequency of data object used [8].

There were described four communication scenarios. *Scenario 1* (type A): communication components are requesting any data content from specific publisher. *Scenario 2* (type B): components requesting specific data content from a specific publisher. *Scenario 3* (type C) in which request specific data content from any publisher. *Scenario 4* (type D) is the case when requested information with any data content is accessing from any publisher [4–8].

The components comprising *SICN* model (fig. 1) are as follows: subscriber, publisher, router, cache server. When a subscriber interested in specific information it will send a pull message: *Interest Request Message* (*IRM*) to its gateway or router. This *IRM* would be one of the four types of subscriber requests.

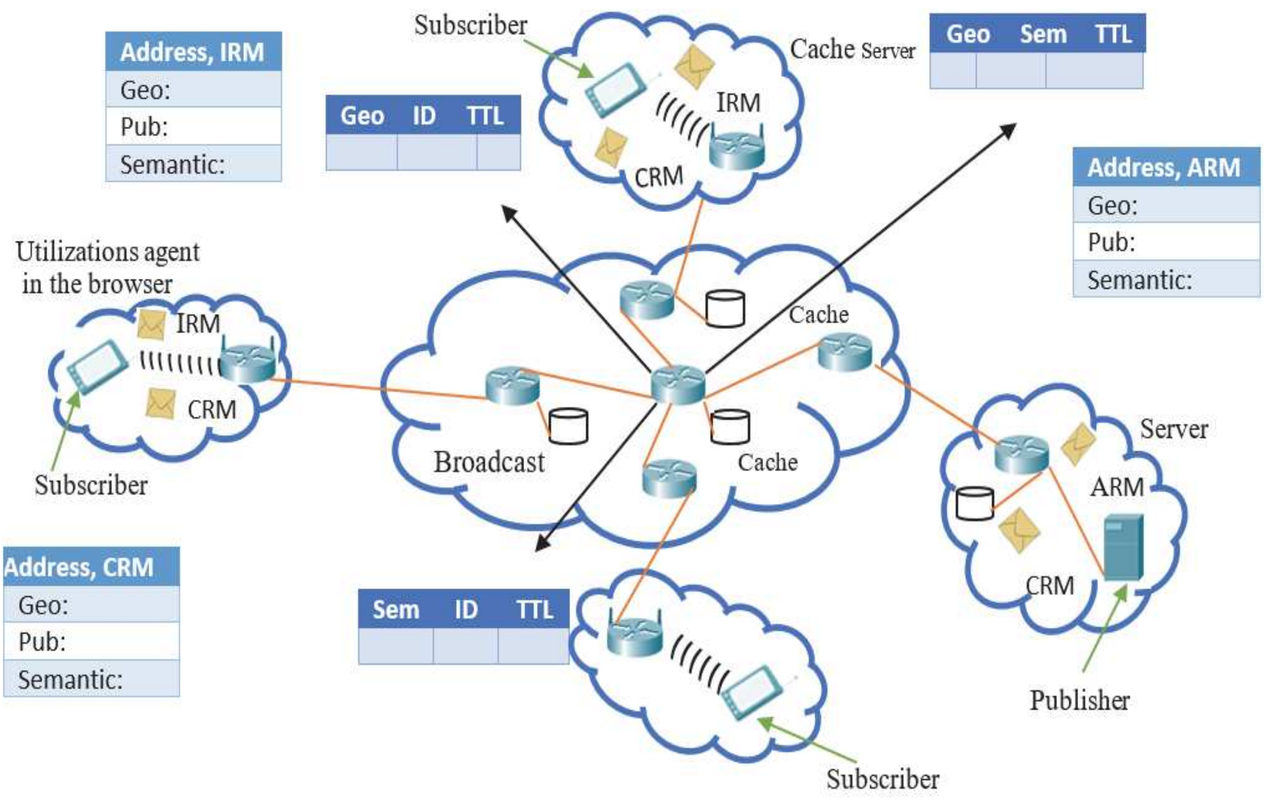

Fig. 1. SICN communication model

Then *Address Reply Message (ARM)* send from the publisher routers or any in path router holding the needed content towards the subscriber requesting certain content by *IRM*. *ARM* holds the addresses. It updates the 3D-address routing tables as follows: when records are not found in table it adds them; if records are found, it increases *TTL* (*Time to Live*) parameter in table for the record.

CRM is the *Content Request Message* that contains the requested data and addresses. It is sent from the publisher to the subscriber.

Theoretical base. According to the *SICN* architecture routers have three tables where three dimensions combined, named 3D-tables. These three address dimensions will allow the matching between publisher and subscriber based on naming scheme that includes any of following: publisher ID, semantic or geographical addresses. So, the subscriber having one of the three address dimensions can find a match to the other two address dimensions using the proposing routing tables.

Each table includes two parts. The first part is the address part (may contain publisher ID, geographical or semantic addresses) that names the data and are learnt or defined from publisher advertisement. The address part consists of semantic and publisher ID addresses in Semantic-Pub ID table, and of publisher ID and geographical addresses in Geo-Pub ID table, and of semantic addresses and geographical addresses, in Geo-Semantic table (fig. 2–4).

The second part of each table is the orientation part that directs the data toward the subscriber, and are learnt from subscriber interest message. The *TTL* represents *Time to Live* in the cache. The *Interface* is the input output ports, which connects network nodes. *Hops number* shows the cost to reach data in terms of hops number.

The first table matches Semantic to Publisher ID address (fig. 2).

Semantic address	Publisher ID address	*TTL*	Interface	Hops number

Fig. 2. Routing table header for Semantic-Pub ID maps Semantic addresses to Publisher ID address

The second table matches Publisher ID address to Geographical address (fig. 3).

PublisherID add.	Geographical address	*TTL*	Interface	Hops number

Fig. 3. Routing table header for Geo-Pub ID maps Publisher Unique address to Geographical address

The third table matches Semantic address to Geographical address (fig. 4).

Semantic address	Geographical address	TTL	Interface	Hops number

Fig. 4. Routing table header for Semantic-Geo maps Semantic addresses to Geographical address

Updating. Let's consider the updating records table processes. It works depends of *TTL* and the following value as shown in table.

***TTL* types**

TTL type	Description
TTL^T	Time to Live for the records in the table
$TTL^T_{th\ \min}$	Minimum threshold to live for record in table. Role: minimum value of TTL^T of the record to stay in 3D-table. If $TTL^T < TTL^T_{th\ \min}$ then the garbage collector will remove the record.
$TTL^T_{add\ \mathrm{def}}$	The value of the default TTL^T that the new added record to the table (equal β)
$TTL^T_{th\ \max}$	Maximum there hold time to live forward in the table: Role: the needed value of TTL^T of the record during addition to be added to the cache.

Adding records to the tables. When a subscriber sends *Interest Request Message* (*IRM*) and it reaches the router, the router will search its tables for an entry match. There are three cases:

Case 1: If there is no positive match between *IRM* and any addressing table in the router, the router will broadcast the *IRM* to the network from all its interfaces in a spanning tree technique to avoid loops. When *IRM* reaches the match in any router caching the content or even in the original publisher cache, the latest will send a reply message to all requested interfaces, which sent *IRM* to it containing the data with the three publisher addressing dimensions. The *Content Reply Message* (*CRM*) will allow updating all the routing tables across the requested paths towards the subscriber who sent the *IRM* for the first time. Thus, when there are no previous records in the tables, it will be created with a default *TTL*.

Case 2: If positive naming matches occurs between *IRM* and routing table and the router already caches the content, *TTL* will be updated (increased) in the routing table and calculated based on the number of subscribers interested in the address and the number of usages of this address. Then the *Address Reply Message* (*ARM*) will be sent to the subscriber with the three addresses dimensions without the content to allow the subscriber to choose.

Case 3: If positive naming matches occurs between *IRM* and routing table but the content is not cached locally in the router, the router will forward the message toward the nearest publisher which could be a relay cache or the original publisher directly from the interface connected towards them since their ID or IP are recorded in the table. The latest will send the *ARM* with its three addressing dimensions. This case applies when the subscriber making a video call or asks for a video from certain publisher as YouTube.

After big count of requests, the *TTL* will reach a threshold defined by the administrator TTL_{th} allowing to caching the data and as *TTL* increases caching time will also increase.

In all cases, the interfaces in the tables will be learnt from the interface that passed the reply message.

Removing records from the tables. *TTL* will decay automatically with a ratio γ defined by the administrator. If *TTL* reaches the minimum threshold $TTL^{T}_{th\ min}$, which means there no interests from subscriber exists for a long time, then the record will be removed from the table and from the cache if it exists. It allows the tables to be always dynamic and up to date.

Considering the case of subscriber message coming from interface matches the advertised naming of the publisher, the router will automatically increase the *TTL* of this interface if the match of the data already increases. However, if it was not existed, the source interface of the subscriber message will be added to the publisher record in the routing table and then increase *TTL*.

Merging records in tables. Merging is the process of generalize the addresses in the routing table in order to reduce its size.

In semantic address part the specific relations will be merged into the general relations. In publisher ID address the table will keep the most general publisher ID depending on the number of requests. In geographical address part the merge will take place on the network. If we have any super network and the subnetwork on the same interface, super network will be considered and the sub network will be discarded.

Matching. Matching is the process of comparing the addresses of the *IRM* to the addresses existed in the routing tables and find the best match based on the longest prefix match.

Matching occurs on three levels based on the *IRM* address content: semantic matching, geographical matching, and publisher ID matching.

Semantic matching. Each semantic address is composed of a set of *Object-Relation-Object* (*ORO*) chains. When an *IRM* holding a semantic address is requested, the router will search for the *ORO* chains of this semantic address in the Geo-Semantic and Semantic-Pub ID routing tables respecting the logical relations (or/xor/and/in/super) determined by the user. Knowing that in semantic matching the router chooses from its tables only *ORO* chains from the same information source (Publisher ID, Geographical IP). In a brief, the semantic matching is the matching by all chains for the same source.

Geographical matching. When an *IRM* holding a geographical address is requested, the router will search for the longest prefix match in the geo-semantic and Geo-Pub ID tables to route *IRM* to the requested information source. If the requested *IP* is not found in the last router because the mobility of the user, this router will multicast the prefix to all the nearest neighbor subnets with fixing the suffix which is based on EUI64 address assignment.

Publisher ID matching. When an *IRM* holding a Publisher ID is requested, the router will search for this publisher in the Semantic-Pub ID and Geo-Pub ID tables. When the router finds a positive match respecting logical relation, it will route the *IRM* toward the source of information using the geographical address.

3D-routing table algorithms. For previously described processes of add new records, update *TTL* for the record, add records to cache when *TTL* reaches threshold, matching, garbage collection for removing records there are needed algorithms.

Router Record Update (RRU) Algorithm. When router receives new *ARM* and the record is not already existing in it, *Router Record Update algorithm* adds the new record to the tables with a default *TTL* (TTL^{T}_{i}.) However, if the record is already existing in the table, *RRU* adds its TTL^{T}_{i} by a unit (α*T*) till reaching the minimum threshold. When *TTL* of the record reaches the minimum threshold, *RRU* send the data to the cache after receiving the next *CRM*. Thus, the functions of *RRU* are: add new record with *TTL* add; update *TTL* by increasing it by one unit after each *ARM*; add record to cache when TTL^{T}_{i} reaches $TTL^{T}_{th\ \min}$.

Algorithm 1: *Router Record Update (RRU)*

Start
  Receive *ARMi*
  If (*ARMi*) exist then
    $TTL^{T}_{i} = TTL^{T}_{i} + \alpha^{T}$
  Else
    add address (*ARMi*) to table
  End If
  If $TTL^{T}_{i} > TTL^{T}_{th\ \min}$ then
    add next (*CRM_i*) to cache
  End If
End

Geo-Pub ID Garbage Collector (GPIGC) Algorithm. *Geo-Pub ID Garbage Collector* (*GPIGC*) decreases TTL^{T}_{i} for each record in the Geo-Pub ID table by a constant unit (γ*T*) continuously and checks it if its TTL^{T}_{i} is less than the minimum threshold to remove it from the table. Thus, *GPICG* role is to decrease TTL^T and to delete relation with low weight (less than min-threshold) in the Geo-Pub ID table performing scalability to it.

Algorithm 2: *Geo-Pub ID Garbage Collector (GPIGC)*

1. Start
2. $Timer = Time + 1$
3. $TTL^T = TTL^T - \gamma_T$
4. If $TTL^T < TTL^T_{th\,\min}$ then
5. remove record from table
6. End If
7. End

Semantic Garbage Collector (SGC) Algorithm decreases TTL_i^T for each record in the Semantic-Geo and Semantic-Pub ID tables by a constant unit (γT) continuously and checks it if its TTL_i^T is less than the minimum threshold ($TTL^T_{th\,\min}$) or if the weight W of the record is less than the minimum threshold weight ($W_{th\,\min}$) to remove it from the table. Thus, *GPICG* role is to decrease $TTL^T$and delete relations with low weight (less than min-threshold) or less than $TTL^T_{th\,\min}$ in the two tables: Semantic-Geo and Semantic-Pub ID.

Algorithm 3: *Semantic Garbage Collector (SGC)*

1. Start
2. $Timer = Time + 1$
3. $TTL = TTL^T - \gamma c$
4. If $TTL^T < TTL^T_{th\,\min}$ or $W < W_{th\,\min}$ then
5. remove record from the table
6. Else go to 2
7. End If
8. End

Cache algorithms. Caching on-path (in-network) is considered to be a basic architectural component of an ICN architecture. It may be used to provide a Quality-of-Service (QoS) experience to users, reduce the overall network traffic, prevent network congestion and Denial-of-Service (DoS) attacks and increase availability

SICN uses On-path caching strategy that reduces download time, server workload, and network traffic resulting in a less response time for data requests and reduces distance of data path to reach subscriber. In the proposed scheme, on path cach use threshold where two methods: Garbage Collector (GC) algorithm and *TTL* time decay function to reduce *TTL* and consequently reducing the amount of data.

Cache TTL Update (CTU) Algorithm. Cache *TTL* Update algorithm adds the TTL^C of each record in the cache existed in the cache by a constant unit (αc) after receiving a new *CRM*.

Algorithm 4: *Cache TTL Update (CTU)*

1. Start
2. Receive *IRM2i*
3. If exist (*CRMi*) then
4. $TTL^C = TTL^C + \alpha c$
5. End If
6. End

Cache Garbage Collector (CGC) Algorithm decreases TTL^C of each record in the cache of by decay time unit γ. *GPICG* checks also TTL^C each record in the cache to remove it when reaches the minimum cache threshold ($TTL^C_{\min}$). Thus, *CGC* role is to decrease TTL^C continuously and to control the size of the cache by removing records with low *TTL* (less than $TTL^C_{th\,\min}$).

Algorithm 5: *Cache Garbage Collector (CGC)*

1. Start
2. $Timer = Time$
3. $TTL^C = TTL^C - \gamma C$
4. If $TTL^C < TTL^C_{\min}$ then
5. remove record from cache
6. End If
7. End

Conclusion There were presented three algorithms for the 3D-address routing table and two for the cache. It includes *Router Record Update Algorithm, Cache TTL Update algorithm*, *Matching algorithms* and *Semantic table update learning algorithm.*

To compare between the different ICN projects, we build a Python model under some assumptions (users number, publisher, DNS and cache depth, sharing coefficient and factor). Empirical results were tested over four scenarios: non-cached data level; cached data level; information level; knowledge level. It was analyzed three metrics are: *Time Delay (TD)*, *Flooding* or *Traffic (F)*, and *Efficiency Reuse (ER) factor* for data.

In *Scenario 1 SICN* obtains much better performance of *SICN* compares to other schemas in terms of flooding parameter.

In *Scenario 2* KBN, CBCB, and *SICN* have the lowest *TD* (for 120 users in 2,7 times lower) . In terms of time delay, KBN is the lowest followed by *SICN*.

In *Scenario 3 TD* is the lowest in KBN, CBCB, and *SICN* schemas, but *F* in *SICN* exceeds *F* in KBN and CBCB.

In *Scenario 4 SICN* and KBN show better results than other schemas in terms of *TD*. In terms of *F* and *ER*, KBN outperforms *SICN*.

A significant advantage of using the proposed *SICN* model and the described algorithms are: compensation search engine and DNS server, reducing traffic flow, ensure scalability (information based is the most scalable and on-line cache maintain scalability), maintain mobility.

References

1. Kutscher D. Information-centric networking (ICN) research challenges. *Research Report*, 2016, pp. 1–32.

2. Garcia-Luna-Aceves J. J. A Comparison of Name-Based Content Routing Protocols. *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems,* 2015, pp. 537–542.

3. Bouk S. H., Ahmed S. H., Kim D. Hierarchical and hash-based naming scheme for vehicular information centric networks. *IEEE International Conference on Connected Vehicles and Expo (ICCVE).* IEEE, 2014, pp. 765–766.

4. Patsei N. V., Jaber G. Semantic base addressing strategies for Information-Centric Networking. *Problems of Information Technology: Materials of the III All-Ukrainian Science-and-Technology Conference.* Poltava, 2019, p. 103.

5. Jaber G., Patsei N. V., Rahal F. Semantic information-centric networking naming schema. *Trudy BGTU* [Proceedings of BSTU], issue 3, Physics and Mathematics. Informatics, 2020, no. 1 (230), pp. 69–73.

6. Patsei N., Jaber G. Routing Schema for Information-Centric Networking. *11th International Conference NEET.* Publ., Lublin, Lublin University of Technology, 2019, p. 29.

7. Jaber G., Patsei N. V. Information Centric Networking for web-based content distribution and manipulation. *Trudy BGTU* [Proceedings of BSTU], issue 3, Physics and Mathematics. Informatics, 2017, no. 2, pp. 88–91.

8. Jaber G., Patsei N., Rahal F., Abboud A. Naming and Routing Scheme for Data Content Objects in Information-Centric Network. *2020 Open Conference of Electrical, Electronic and Information Sciences (eStream): Proceedings of the Conference*. Vilnius, Lithuania, 2020, pp. 93–97.

Список литературы

1. Kutscher D. Information-centric networking (ICN) research challenges. *Research Report*. 2016. P. 1–32.

2. Garcia-Luna-Aceves J. J. A Comparison of Name-Based Content Routing Protocols. *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*. 2015. P. 537–542.

3. Bouk S. H., Ahmed S. H., Kim D. Hierarchical and hash-based naming scheme for vehicular information centric networks. *IEEE International Conference on Connected Vehicles and Expo (ICCVE).* IEEE, 2014. P. 765–766.

4. Patsei N. V., Jaber G. Semantic base addressing strategies for Information-Centric Networking. *Problems of Information Technology: Materials of the III All-Ukrainian Science-and-Technology Conference.* Poltava, 2019, p. 103.

5. Jaber G., Patsei N. V., Rahal F. Semantic information-centric networking naming schema // Труды БГТУ. Сер. 3, Физика-математические науки и информатика. 2020. № 1 (230). С. 69–73.

6. Patsei N., Jaber G. Routing Schema for Information-Centric Networking. *11th International Conference NEET. Lublin University of Technology*. Lublin, 2019. P. 29.

7. Jaber G., Patsei N. V. Information Centric Networking for web-based content distribution and manipulation // Труды БГТУ. Сер. 3, Физика-математические науки и информатика. 2017. № 2. С. 88–91.

8. Jaber G., Patsei N., Rahal F., Abboud A. Naming and Routing Scheme for Data Content Objects in Information-Centric Network. *2020 Open Conference of Electrical, Electronic and Information Sciences (eStream): Proceedings of the Conference*. Vilnius, Lithuania, 2020. P. 93–97.

Information about the authors

Jaber Ghassan – PhD student, the Department of Software Engineering. Belarusian State Technological University (13a, Sverdlova str., 220006, Minsk, Republic of Belarus). E-mail: ghassanjaber@hotmail.com

Patsei Natallia Vladimirovna − PhD (Engineering), Associate Professor, Head of the Department of Software Engineering. Belarusian State Technological University (13a, Sverdlova st., 220006, Minsk, Republic of Belarus). E-mail: n.patsei@belstu.by

Информация об авторах

Гассан Джабер – аспирант кафедры программной инженерии. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: ghassanjaber@hotmail.com

Пацей Наталья Владимировна – кандидат технических наук, доцент, заведующий кафедрой программной инженерии. Белорусский государственный технологический университет (220006, г. Минск, ул. Свердлова, 13а, Республика Беларусь). E-mail: n.patsei@belstu.by

Received 25.04.2022