of the 4WARD project // Second NetInf Architecture Description, 2010. – URL: http://www.4ward-project.eu/index.php?s=filen downloadn&id=70 (дата обращения: 17.11.2022).

5. Conceptual Architecture of PSIRP Including SubcomponentDescriptions, Public Deliverable (D2.2) of the PSIRP project. – URL: http://www.psirp.org/files/Deliverables/FP7-INFSO-ICT-216173-PSIRP-D2.2 ConceptualArchitecture v1.1.pdf (дата обращения: 10.07.2022).

6. Ahmed R., Boutaba R. Distributed Pattern Matching: A Key to Flexible and Efficient P2P Search // IEEE Journal on Selected Areasin Communications (JSAC) issue on Peer-to-Peer Communications and Applications, vol. 25, no. 1, 2007. – p. 73-83.

UDC 316.776

N.V. Patsei, PhD stud. G. Jaber
(BSTU, Minsk)

SEMANTIC INFORMATION-CENTRIC NETWORKING MODELING RESULTS

To compare Semantic Information-Centeric Networking (SICN) with other Name Data Networking (NDN) projects it was built a Python model under some assumptions [1]. Model has following components:

- *Publisher*: main content source;
- *Subscriber*: data content user;
- *Search engine*: make data translation from informal to formal form;
- *DNS*: used to find IP of data source;
- *Cache*: saving data.

To reduce model complexity was held the following notations and assumptions: u – is the number of users ($u = 10$); n - publisher depth (it is defined as number of extended branches from root of tree from a subscriber to data source, n is variable); e - search engine depth (defined as number of branches from root of tree from a subscriber to search engine, supposed that $e = n$); d - *DNS* depth (is defined as number of branches from root of tree from a subscriber to *DNS*, $d = n/2$); c – cache depth (defined as number of branches from root of tree from a subscriber to cache, supposed that $c = n/2$); s – sharing coefficient (is defined as the ratio of shared of links by subscribers to total links. $s = 0.25$); r – sharing factor (is defined as the utilization factor from sharing paths between subscribers, $r = 1+ s (u -1)$); L – total number of extended branches for each subscriber to data source $L = 2^{(n+1)} – 2$ (is supposed that each node has two branches).

We computed three different metrics over the six schemas: IP, DONA, PURSUIT, CBCB, KBN, and SICN, which characteristics are presented in table.

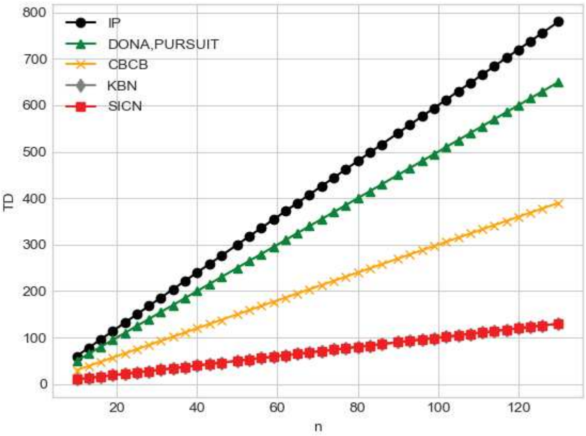
Metrics used in the empirical results are: *Time delay* (*TD*), defined as the minimum number of hops starting from requesting data by a subscriber till receiving the requested data per subscriber assuming that there is no queuing for subscribers and it is constant for subscribers, *TD* € *N*, *N* is a nature number, [0, ∞ [; *Flooding or traffic* (*F*) - is total number of links occupied starting from requesting data till receiving it, *F* € *N*, [0, ∞ [; *Efficiency Reuse* factor for data (*ER*) - represents the ratio of the number of interests of a certain data to the data travel distance.

Table – ICN project characteristics

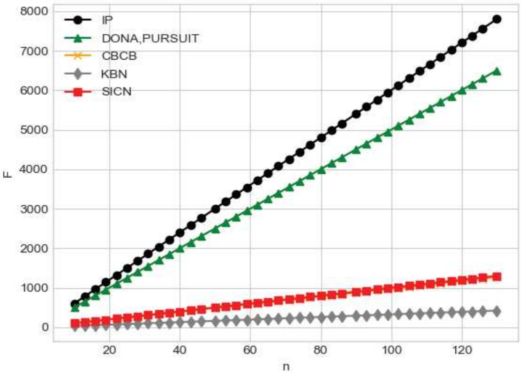
ICN project	*Routing approach*	*Naming structure*	*Routing*	*Caching*	*Abstract Level*
DONA	name resolution	flat naming <p:l> self-certifying	pull	on-path off-path	data
CBCB	name based routing	set of paired attribute value don’t ensure uniqueness	push–pull	on-path off-path	information
NDN	name based routing	hierarchical human friendly	pull	in-network caching	data
PURSUIT	name resolution	flat naming <p:l>	pull	off-path on-path	data
Netinf	name resolution	flat <p: l>	pull	on-path off-path	data
CCN	name based routing	hierarchical, human friendly	pull-push	on-path off-path	information
KBN	name based routing	set of keywords ontological categorization	pull-push	on-path of-path	knowledge
SICN	name based routing	human friendly, hierarchal	pull	on path	knowledge

The content transmitted is knowledge. It is the highest abstract level of the content and is *R4* request type where subscribers ask for knowledge from any publisher [2]. SICN and KBN show better results than other projects in terms of *TD* (pic.1). All other schemas need search engine to understand the knowledge level content.

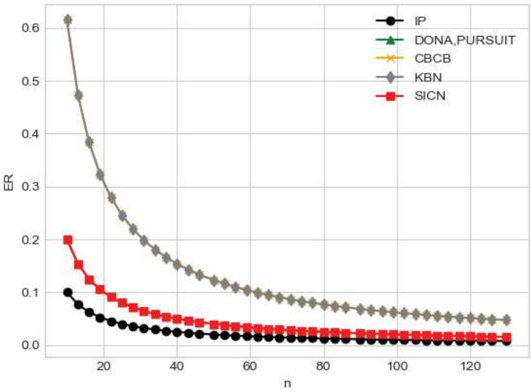
In terms of *F* and *ER*, KBN outperforms SICN as the first benefits from shared paths (pic. 2-3). IP shows less performance in the three metrics as a result for depending on the publisher as a source of data and not caching it.



Pictogram 1 – Time Delay vs. Number of Links to the Data Source



Pictogram 2 – Flooding vs. Number of Links to the Data Source



Pictogram 3 – Efficiency of Data vs. Number of Links to the Data Source

An important feature of SICN is that it is backward compatible, i.e. it would work over IP. Empirical results were tested with knowledge on three different metrics over six schemas. SICN and KBN show better results than other schemas in terms of *TD*. In terms of *F* and *ER*, KBN outperforms SICN.

REFERENCES

1. G. Jaber, N.V. Patsei 3D-ROUTING TABLE ALGORITHMS IN SEMANTIC INFORMATION CENTRIC NETWORKING // Информа-

ционные технологии : материалы 84-й науч.-техн. конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием), Минск, 4-14 февраля 2020 года [Электронный ресурс] / отв. за издание И.В. Войтов; УО БГТУ. – Минск : БГТУ, 2020. – с. 79-80.

2. 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 : April 30, 2020, Vilnius, Lithuania, IEEE -2020. P.93-97.

УДК 004.432.2

Ст. преп. А.С. Наркевич
(БГТУ, г. Минск)

C++20: МОДУЛИ

В 2022 году языком программирования года по версии TIOBE стал C++. На данный момент действует стандарт языка C++ 20. Стандарт C++23 официально еще не утвержден, но его уже можно опробовать на компиляторах GCC, Clang, MSVC [1].

Одним из главных нововведений стандарта C++20 являются модули, которые реализованы начиная с Visual Studio 2022 версии 17.1.

До модулей в C++ использовались заголовки – отдельные текстовые файлы с расширением .h или .hpp, при подключении которых содержимое файла копировалось в место их включения в программу. Возможные проблемы:

– неочевидный побочный эффект включения заголовочных файлов (в зависимости от порядка расположения два включаемых фрагмента могут влиять друг на друга);

– нарушение правила одного определения (one definition rule):

– функция или класс могут включаться в разные файлы .cpp, разные единицы трансляции, что может привести к нарушению правила одного определения;

– включения из заголовочного файла зависят от макросов, которые могут быть переопределены в момент включения;

– медленная компиляция: возникает, когда один и тот же заголовок целиком включается в разные единицы трансляции, компилятор компилирует его каждый раз;