Biblioteka Informatyki
Szkół Wyższych

Information Systems Architecture and Technology

Networks Design and Analysis

Library of Informatics of University Level Schools

Series of editions under the auspices
of the Ministry of Science and Higher Education

The ISAT series is devoted to the publication of original research books in the areas of contemporary computer and management sciences. Its aim is to show research progress and efficiently disseminate current results in these fields in a commonly edited printed form. The topical scope of ISAT spans the wide spectrum of informatics and management systems problems from fundamental theoretical topics to the fresh and new coming issues and applications introducing future research and development challenges.

The Library is a sequel to the series of books including Multidisciplinary Digital Systems, Techniques and Methods of Distributed Data Processing, as well as Problems of Designing, Implementation and Exploitation of Data Bases from 1986 to 1990.

Wrocław University of Technology

Information Systems Architecture and Technology

Networks Design and Analysis

Editors
Adam Grzech
Leszek Borzemski
Jerzy Świątek
Zofia Wilimowska

Wrocław 2012

Publication partly supported by
Faculty of Computer Science and Management
Wrocław University of Technology

Project editor
Arkadiusz GÓRSKI

The book has been printed in the camera ready form

OFICYNA WYDAWNICZA POLITECHNIKI WROCŁAWSKIEJ
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław
http://www.oficwyd.pwr.wroc.pl;
e-mail: oficwyd@pwr.wroc.pl
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-703-0

CONTENTS

INTRODUCTION

The overall gain of contemporary proposed and deployed ICT (Information and Communication Technologies) applications is to explore and utilize new concepts, paradigms and architectures to increase the effectiveness of business processes and to propose applications of high societal value through making use of reappraised network architectures, services and technologies in large-scale application context. New functionalities of information systems are supported by new concepts to provide network services.

Most of today applications are so called *communication enabled applications*. Such an approach is representative for business processes that could be accelerated through communication networks. In such a case applications' functionalities depend on real-time networking capabilities together with network-oriented location, presence, proximity, and identity assurance functions. Available services are delivered based on implicit assumption that both domain- and network-specific services are available as callable within frameworks from which the application is composed. In gain to provide callable services, the domain- and network-specific services should be made virtual and component-like as well as representative for business process that could be accelerated through communications enablement.

Augmentation of the current networks architectures fulfill the assumptions of the *pervasive computing* paradigm where end-to-end services delivery is facilitated by a cloud of distributed networking devices and loosely coupled application modules. The key feature of such an approach is the user–centricity where the user does not invoke any particular applications or service nor even specifies where the application should be executed.

The abovementioned *pervasive computing* is strongly related to the concept of *service centric architectures* as a next generation service architecture needed to assure services flexibility, adaptability, security, content delivery mechanisms and management platforms facilitating mapping of users requirements onto the lower infrastructure layers to provide a broad range of services based on efficient service paradigms.

Observed trends in the contemporary ICT technologies and their applications may be distinguished as motivated by several important and perspective paradigms leading to service oriented systems with quality of services assured by intensive knowledge processing and communication enabled.

The book addresses subjects dealing with various methodological, technological and applications aspects of distributed information and communication systems, i.e., technologies, organization, application and management involved in gain to increase efficiency, resources utilization, flexibility, functionalities and quality of services offered by contemporary information and computer systems.

Chapters, selected and presented in the book, address a number of issues important and representative both for available information and communication technologies as well as information system users requirements and applications. Submissions, delivered within distinguished chapters, are strongly connected with issues being important for contemporary information processing, communication and data communication system.

The book is divided into two parts, which include fourteen chapters. The two parts contain chapters addressing – sometimes very particular – issues widely reported in research and technical papers and important for today's information and communication technologies and their implementations.

The parts have been completed from chapters addressing some extensively researched and recounted in the world literature important and actual issues of distributed information systems. The proposed decomposition of accepted set of chapters into parts is to compose units presenting methods, algorithm and tools for knowledge processing, information systems requirement analysis, service oriented systems, social network applications as well as modeling, analysis and optimization of networks infrastructures enabling content and context delivery of information.

The first part – **Knowledge and Social Networks Applications** – contains chapters addressing various issues related to knowledge bases, ontologies, semantic networks, social networks related to service oriented systems and resources management in such a systems. In chapters, included in this part, different aspects of knowledge and services description, processing, customization and deployment are considered. Selected chapters show advantages as well as shortcomings and difficulties in implementations of service oriented architecture and service oriented knowledge utilities paradigms and concepts.

The second part – **Content Aware Networks and Network Services** – is composed of chapters where some selected problems strongly connected with various quality of service delivery strategies for networked systems are considered. Problems and issues discussed in this part are related to networks where assurance of quality requires implementation of various traffic engineering concepts: access control, traffic shaping, flow and congestion control, content distribution, content and localization awareness, personalization, etc. Issues, elaborated in the presented chapters, show that new attempts, methods and algorithms are required in gain to obtain higher network resources utilization, increase quality of infrastructural as well as end-to-end network services and deploy effective information control mechanisms.

PART 1. KNOWLEDGE AND SOCIAL NETWORKS APPLICATIONS

In the **Chapter 1** some examples are presented demonstrate an important feature of ontologies, not only do they enable the formulation of knowledge, based on which instances of a certain class of problems can be solved, but they also allow to determine what the problem should satisfy in order to have a solution (i.e. the admissible knowledge base). The discussed and proposed approach to ontology, assuming existence of a layer of constraints among the instances, is related to the techniques of programming with constraints. Apart from this approach, there exists a variety of ontologies covering a whole range of descriptive logic.

Chapter 2 is devoted to discuss various aspects of knowledge fuzzification and it interpretation in the fuzzy sets algebra. For any defuzzification are defined rules describing axioms and conclusion according to rules. In particular a schema of algorithm of knowledge fuzzification and defuzzification in semantic networks is presented as implementable in some programming languages.

The next **Chapter 3** gains is to discuss formalized definition developed for architectural decisions representing service compositions which can make such decisions easier to comprehend and enable integration of existing architecture models. The introduced approach enables a number of the relations between architectural decisions to be defined, which can then be used to capture the decisions defining the structure of a service composition, the choice of composed services and the changes introduced during the evolution of such a service composition. These formalized relations can also be detected automatically, which enables the development of a tool support for decision-making and evolution documentation.

In the **Chapter 4** an approach, according to which the Web services interoperability and resulting composition schemes may be effectively used to create the network structures reflecting the patterns according to which the services interact, is proposed and discussed. It was shown how to create so-called networks of Web services which allow to effectively use the network structural analysis and optimization techniques to solve the network composition problems. The service network is created on the basis of the semantic bindings between the services in the repository joined with the actual patterns of the service usage resulting from composition queries. It was also presented how available techniques of dynamic network structure prediction and analysis may be useful to assess the future service usage and resource consumption of the service execution layer.

Chapter 5 aim is devoted to present early detection methods of social contagion. Authors present known, noteworthy idea of predicting social contagion development using set of individuals (sensors) in society which is based on fact that individuals in the center of network are more likely to be infected sooner, thus they could be used as a sensors that predict future state of whole society. In the chapter alternative algorithms of choosing sensors, without knowledge of social network structure – using only surveys among randomly chosen people are proposed, presented and compared.

The proposed alternative algorithms – as compared with the entire method – offer up to two – three times sooner detection of social contagion.

In the next **Chapter 6** an extension of service composition problem to the area of social networks and graph based service composition quality criterions is presented. The proposed service composition method is decomposed into three steps consisting of composite structure generation, semantic service discovery method and service plan optimization method. The goal of the service discovery is to find service candidates that fulfill functional requirements and the latter allows for optimal selection of services so that together they satisfy quality criterion, here based on social network graph measures. Presented approach is supported by examples from volleyball sport domain, denoted with domain ontology.

Chapter 7 aim is to propose the resource management linked to the notion of Service Level Agreement. Such an approach can bring the resources utilization closer to the ICT supported business processes aims. Unlike resource management oriented toward e.g. minimization of resources usage or processing time the proposed approach can incorporate costs or rewards, thus be directly connected with business objectives, i.e., can combine the flexibility with the possibility of reaching the business goal.

PART II. CONTENT AWARE NETWORKS AND NETWORK SERVICES

The **Chapter 8** deals with selected aspects of the Next Generation Network (NGN) concept, in particular NGN designing and dimensioning. The latter require proper traffic models, which should be efficient and also simple enough for practical applications. In the chapter such a traffic model for a single domain of NGN network based on the IP Multimedia Subsystem (IMS) concept is proposed, which allows to evaluate mean Call Set-up Delay (CSD) and mean Call Disengagement Delay (CDD), a subset of call processing performance parameters defined by International Telecommunication Union Telecommunication Standardization Sector (ITU-T). Using the model basic relationships between network parameters and call processing performance are investigated and presented. All obtained results are verified using simulations, which confirm correctness and usefulness of the proposed model.

In **Chapter 9** the problem of providing QoS for vulnerable services in IPv6 based networks by using self-managing network architecture is considered. To solve the abovementioned task a system cooperating with services and network nodes that fixes connection paths and guarantees minimal bandwidth requested by a network service is proposed. In case of change in a network topology, every path may be changed without loss of any packet. Important requirement of the system was to work with heterogeneous network, so the approach is independent of device vendors, medium types and connectivity technology. The system is composed of stream services, QoS aware middleware and the prototype system based on IPv6 QoS network architecture. In order to create connection between two services, these services negotiate with each other using middleware. After successful negotiations, middleware, on behalf of the services, requests network resources from network management system.

The **Chapter 10** reports results of evaluation of caching mechanisms that may be used in Content Aware Networking. Two algorithms has been evaluated and compared – LRU (Least Recently Used) and LFU (Least Frequently Used). The evaluation and comparison was prepared using open source simulator Omnet++ with an implementation of CCNx (Content Centric Network). The reported results are related to four different and representative topologies with different cache sizes. Parameters that were compared are: number of hops to nearest content, cache exchange ratio and cache hit ratio. Based on the collected results the LFU algorithm is better suited for usage in future Content Aware Networks similar to CCNx network.

In the **Chapter 11** a network storage service, as one of many services in IP network, is considered. In most cases in such a systems an IPSec protocol is commonly used to assure protection of transmitted data. The discussed problem is related to the main problem of the existing IPSec solutions which do not provide the throughput required for storage systems. The chapter presents quantitative and qualitative analysis and comparison of several options for IPSec implementations in network storage systems: location of crypto modules, mode of crypto implementation (software or hardware), mode of IPSec operation (transport or tunnel mode). Some different solutions to performance problem are analyzed: hardware cryptography accelerators, double implementation of IPSec, packet grouping and scheduling algorithms, lazy crypto approach.

The next **Chapter 12** addresses issues related to erasure coding and replication as a two popular and frequently applied methods providing high availability in distributed storage systems. The chapter presents an analytical formula for the expected value of the number of node departures until the first moment of data loss, assuming that the likelihood of a data loss in the system is greater or equal than some fixed value. It was shown that the resistance of a DHT (Distributed Hash Table) – based network to unexpected node failures for erasure coding approach and for replication.

The **Chapter 13** presents a method for modeling of multi-service switching networks with point-to-point selection and a system of overflow links. The concept of effective availability forms the basis for the adopted method for modeling. A particular attention in the article is also given to the way this parameter is determined for switching networks with overflow links. The results of the analytical calculations are compared with the results of the simulations for selected multi-service switching networks with overflow links and point-to-point selection. The study confirms high accuracy of the pro-posed method as well as the suitability of the application of the system of overflow links.

In the **Chapter 14** is devoted to discuss some selected issues related to the translation of the mathematical algorithms to programming languages which open paths to solving many complex problems. In the chapter the theoretical background for the inductive graphs is briefly presented and illustrated by examples of the classical as well as novel algorithms presented in the functional manner. The potential of the func-

tional approach in the further development of the graph algorithms utilized in the communication networks design theory is also discussed.

The **Chapter 15** presents the new Single Size Matching with Permanent Selection (SSMPS) algorithm for control crossbar switches with VOQ (Virtual Output Queuing). The proposed and presented algorithm provides high speed of working. This solution provides high efficiency of our algorithm with no additional calculations. For this reason, presented algorithm is easy to implement in hardware environment.

In the last **Chapter 16** it is shown how to calculate first-, second-, and third-order crosstalk stage-by-stage in the switching structure $\log_2 N-1$. Achieved results are compared with the traditional baseline network. The $\log_2 N-1$ structure gives better optical signal-to-crosstalk ratio for this same functionality and capacity of the switching fabric. It is also discussed how the optical signal goes through a switching network, through what kind and what number of an optical elements. There are also shown exact calculations of the number of passive and active optical elements. The number of such an elements is compared with traditional networks of the same capacity. The $\log_2 N-1$ network has in many cases fewer number of such an elements.

Wrocław, September 2012

Adam Grzech

PART 1

KNOWLEDGE AND SOCIAL NETWORKS APPLICATIONS

ontology, knowledge base admissibility, constraints

Grzegorz BOCEWICZ[*], Robert WÓJCIK[**],
Zbigniew BANASZAK[***]

KNOWLEDGE BASE ADMISSIBILITY: AN ONTOLOGY PERSPECTIVE

"Ontology" stands for the knowledge of the nature of being it encompasses the issues of conceptualization of entities and of the relations between them, and can be treated as kind of a knowledge representation. Since different representations can be associated with different problems the question we are facing with regards of knowledge base admissibility guaranteeing response to assumed set of queries.

1. ONTOLOGY

According to a common definition [1], [3], [6] ontology is a set of strictly defined terms (vocabulary) with regard to a specified area (domain, knowledge) accepted by the community related to that area. Gruber, who is often cited in the literature on this subject, defines it in a similar way: "ontology is a specification of a conceptualization". Based on this, another, different definition of ontology has been proposed – "it is a logical theory that yields formal, partial explanation of a conceptualization". It's also worth mentioning yet another definition here: "ontology is a set of precise descriptive sentences about a certain part of the world (termed an area of interest or a subject matter of ontology)".

[*] Dept. of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland, bocewicz@ie.tu.koszalin.pl

[**] Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wrocław, Poland, robert.wojcik@pwr.wroc.pl

[***] Dept. of Business Informatics, Warsaw University of Technology, Warsaw, Poland, Z.Banaszak@wz.pw.edu.pl

After all these definitions, it's worth asking what ontology really is. It is still difficult to answer this, but now it's easier to come up with a set of associations related to ontology: classification, hierarchy of terms, representation of an area of knowledge, etc. There is also hope that apart from classifications and associations, ontology will finally bestow upon us its practical utility (effectiveness).

In order to satisfy these expectations, for the time being, let's employ one, and this time more formal, definition of ontology proposed for the purposes of object-oriented constraint networks [8]:

$$\mathbb{O} = (O, Q, D, C), \tag{1}$$

where: O – a set of individuals or of classes of objects,
Q – a set of attributes of classes,
D – a set of domains of attributes,
C – a set of constraints.

This definition (1) enumerates six types of constraints related to assigning attributes to classes and attributes to domains, to the compatibility of classes, hierarchic relations, associative relations and functional constraints referring to the names of classes and attributes.

2. ILLUSTRATIVE EXAMPLES

The most significant concepts in ontology are classes, class attributes (Q and D of (1)) and their instances:

- Classes describe the basic concepts of a specified fragment of the world. Classes can contain subclasses , which in turn describe more detailed concepts, cf. Fig. 1. A subclass inherits the features of its parent class, termed superclass.

  Fig. 1 presents an example of a family of classes describing animals (a superclass). *Animals* were divided into subclasses: *Cows*, *Dogs*, *Rabbits*, *Cats*, with each of these subclasses corresponding to a certain group of animals and describing their characteristic features. Of course, each of these animals breathes, therefore, this feature is common, however, barking is characteristic only of the class of dogs. Someone could ask – why have we decided on such a classification? It is, of course, completely arbitrary. Obviously, those of us who deem themselves to be specialists can correct this by introducing new classes (new entities) or by proposing their own ontology with a new structure.

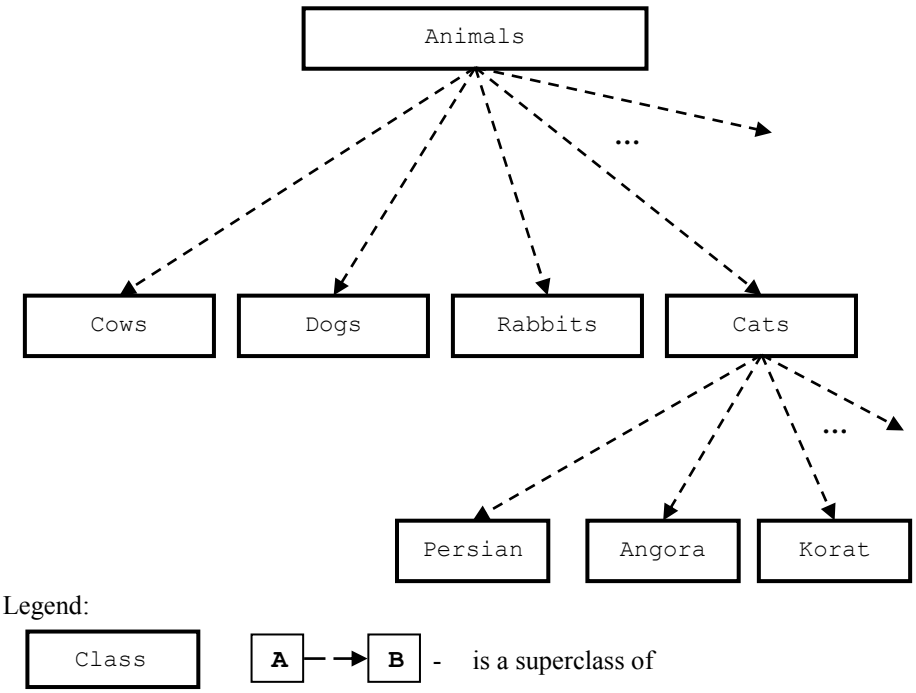

Fig. 1. Example of a class structure

- Attributes are the second significant element in the definition of ontology: they describe the characteristics of classes. Elements (instances) of classes can be different from one another. Although each human being belongs to the class of *Human Beings*, they are characterized by a set of features that make them unique. Examples of such attributes are: *name*, *age*, *colour*, *weight*, *DNA*, etc. Attributes assume values, such as: a string of symbols (e.g. *colour = 'white'*), numbers (*age* = 34), Booleans, etc. In particular, attribute values can themselves be instances (elements) of some other class, e.g. attribute Colour = white, where white is an instance of the class Colour, defined earlier.

  Relationships of this kind (the use of elements of one class in another) determine the relations between individual classes (such relations are denoted with arrows in Fig. 2)

  For instance, in Fig. 2 the class *Cowshed* has an attribute *contains*, whose values are instances of the class *Cow*; the number of instances (count) is denoted with k. A relation of this kind is to be understood as follows: each cowshed has a unique *name*, a *surface area* and inhabitants; there can be k *cows* in a cowshed. Cows are elements of a different class and each of them is characterized by a name and weight.

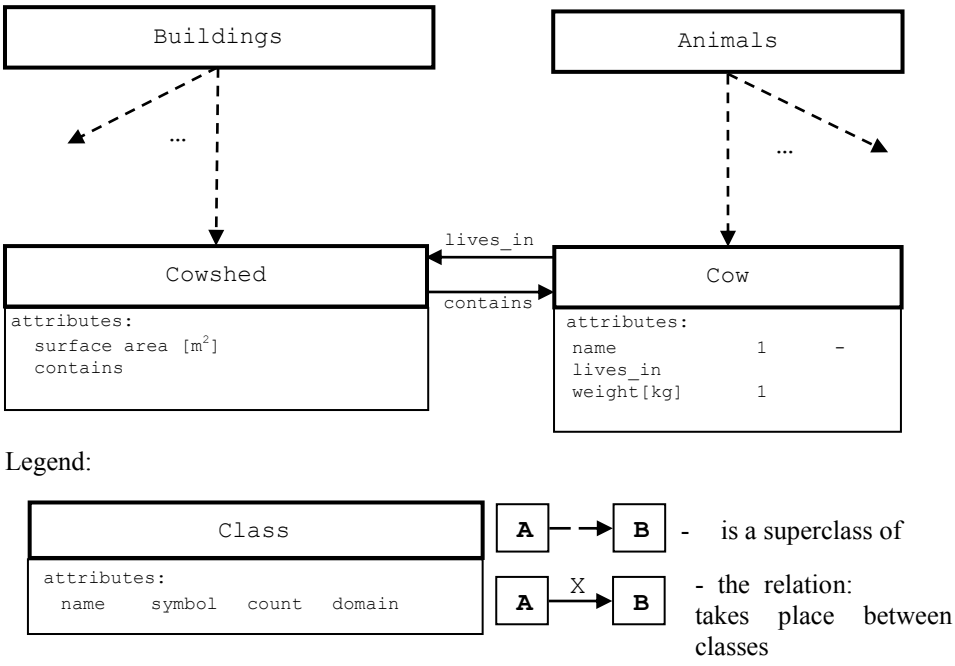

Fig. 2. Attributes of classes

- A family of classes, along with a hierarchy, attributes and relations, suffices to create frameworks (templates) from which specific knowledge bases can be obtained. Determining an element of a class by specifying the values of all its attributes is tantamount to creating an instance of this class. A set of instances is called a knowledge base.
  In the case of the ontology from Fig. 2, we can create instances of the class *Cowshed* and instances of the class *Cow*. Here, instances represent real buildings (in the case of *Cowshed*) or animals (*Cow*). How many cowsheds and cows (knowledge bases) can we have? We don't know this, and since there is no upper limit on their number, we can keep defining them *ad infinitum*. We know, however, that in order to create an instance, we must specify all the attribute values. This means that in order to instantiate *Cowshed no. 1*, we must create at least one instance of the class *Cow*. This requirement results from the relation between the classes *Cowshed* and *Cow*. In other words, in a cowshed, there must live cows (a set of instances of the class *Cow* must correspond to an instance of *Cowshed*). An example of a set of instances for ontologies in Fig. 2 is presented in Fig. 3.

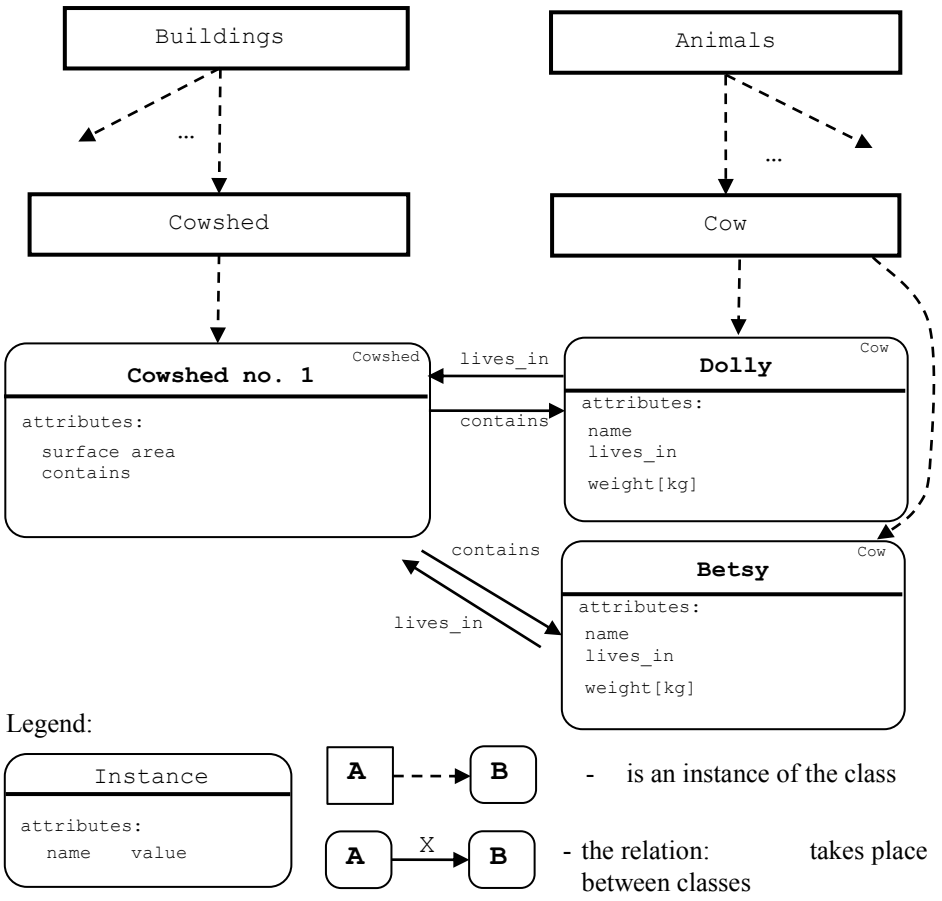

Fig. 3. Example of class instances

Thus, we got to know three basic elements that constitute ontology, which we can now employ to create more complex representations. But how are we to make any practical use of this entire ontology?

In turn, the ontology from Fig. 3 allows us to determine the location of a given animal.

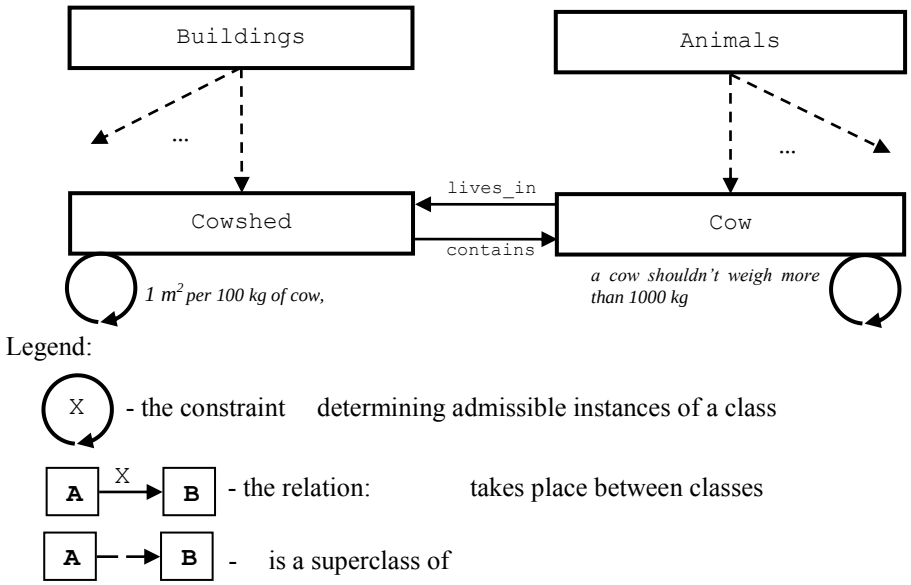

Fig. 4. Constraints for an example class structure

We must emphasize that in the presented formalism there are no constraints which would limit the number or type of created instances. This means that we cannot answer quantitative questions such as:

- How many cows will fit into *Cowshed no. 1* (Fig. 3)?
- What is the smallest cowshed (in terms of surface area) for a given number of cows?
- etc.

In order to be able to answer such queries, we must augment the description of a class by the addition of constraints. Constraints (i.e., the element C of (1)) determine relations (Boolean-algebraic) between the values of attributes of instances of a given class [2]. For instance, a cow shouldn't weigh more than 1000 kg, and for every 100 kg of cow, we must have 1 m^2 of area in a cowshed (Fig. 4) (of course, these values are once again arbitrary). The above constraints determine the set of instances which can be created based on the given ontology. Some examples are presented in Fig. 4 and in more detail in Fig. 5.

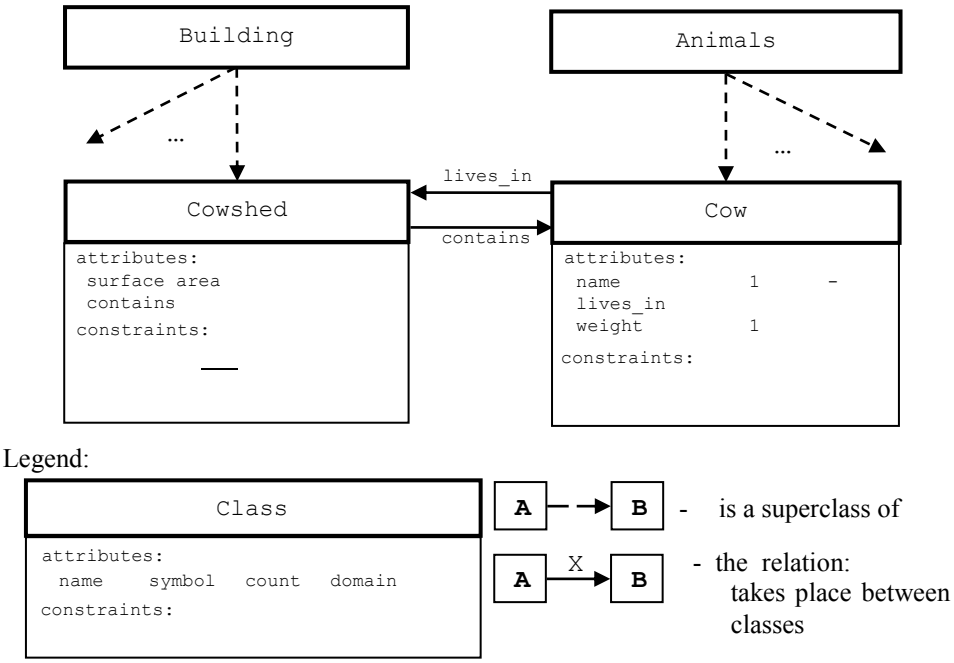

Fig. 5. Attributes and constraints for the class structure of interest

3. KNOWLEDGE BASE ADMISSIBILITY

The introduction of constraints means that not every knowledge base will be admissible in the framework of the defined ontology. An example of a knowledge base that doesn't satisfy the above constraints is presented in Fig. 6. And so, we can observe that, first of all, *Betsy* the cow is too heavy, and second of all, the cowshed at our disposal is too small for two cows. Conclusion: the presented knowledge base (Fig. 6) is inadmissible in the context of the proposed ontology.

Then, what knowledge bases are admissible in this context?

By answering this question, we also answer the questions posed before:

- How many cows can fit into *cowshed no. 1*, with an area of 20 m^2?
- How big a cowshed (in terms of surface area) should we have to fit in the given cows (Dolly – 300 kg, Betsy – 600 kg)?

In the first case, we create the instance *Cowshed no. 1* (Fig. 7) with an area of 20 m^2. The instances of the class *Cow* remain unknown. We don't know how many of them there are, but there must be at least as many as to satisfy all the constraints.

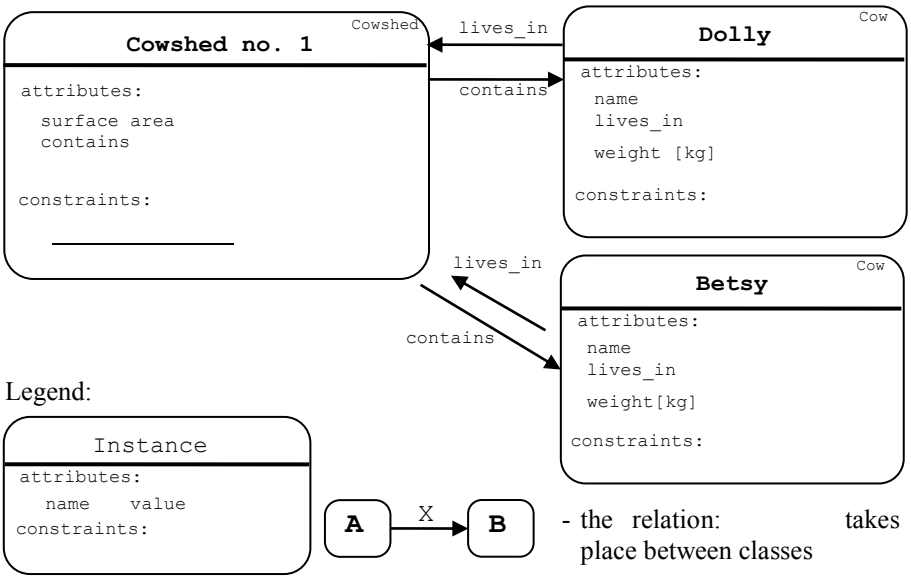

Fig. 6. Attributes and constraints for the class structure of interest

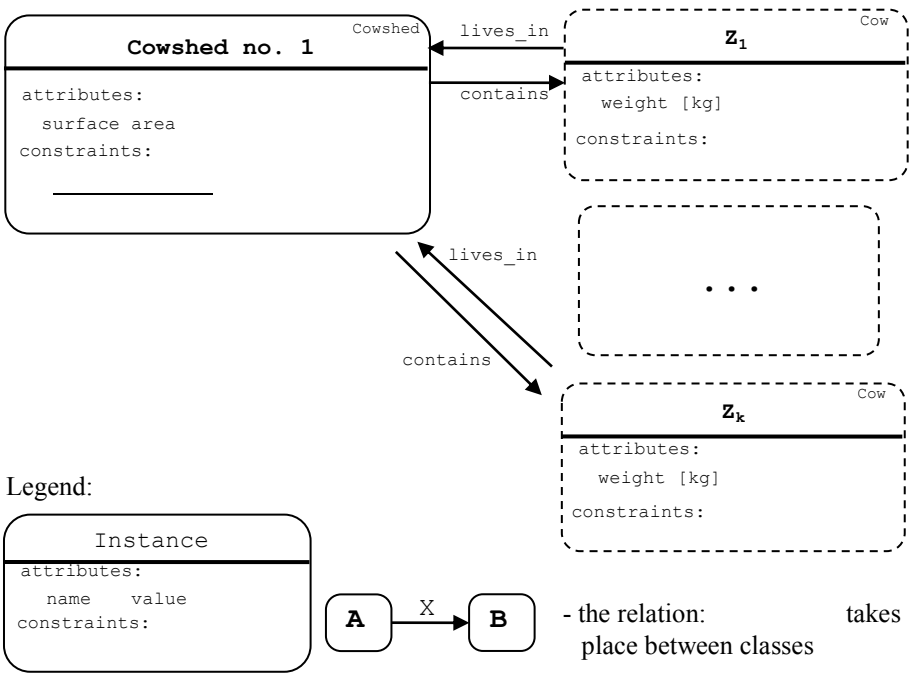

Fig. 7. Knowledge base for the question: how many cows will fit into *cowshed no. 1*?

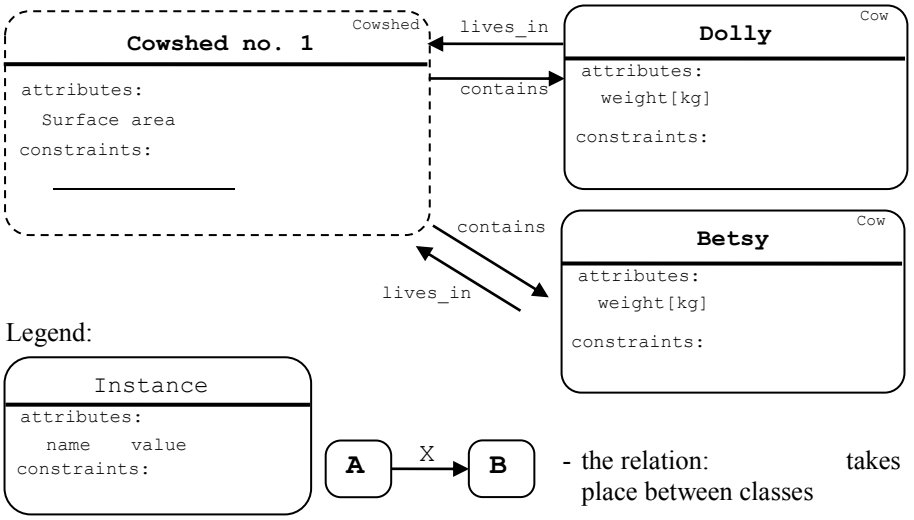

Fig. 8. Knowledge base for question 2

Neither cow can weigh too much and they must all fit into the cowshed. Here, the answer boils down to answering the following question – for the ontology of interest, is there any such admissible knowledge base that contains the instance *Cowshed no. 1* and a finite set of instances of the class *Cow* satisfying constraints C_1 and C_2?

In the second case, we seek to answer the question – for the ontology of interest, is there any such admissible knowledge base that contains the instances from Fig. 8? Thus, we have information about two cows: *Dolly* and *Betsy* (each of them weighs sufficiently much) and we ask whether it is possible to create an instance of the class *Cowshed*, whose area would be sufficient for both cows?

In both cases we were looking for instances, for which all the given constraints would be satisfied. This approach can be easily associated with constraint satisfaction problems. The partially filled knowledge bases that we propose (in one case we specified the cowshed, in another – the set of cows) is a form of a CSP. Therefore, if we have a suitable ontology, we can generate various versions of the CSP (depending on our needs), whose solution allows to answer the questions posed in its framework.

4. CONCLUSIONS

The presented examples demonstrate an important feature of ontologies, not only do they enable the formulation of knowledge, based on which we can solve instances of a certain class of problems, but they also allow to determine what the problem

should satisfy (e.g. what animals may live in a cowshed) in order to have a solution (i.e. the admissible knowledge base).

In conclusion, we should add that the proposed approach to ontology, assuming existence of a layer of constraints among the instances, is related to the techniques of programming with constraints. Apart from this approach, there exists a variety of ontologies covering a whole range of descriptive logic [4], [5], [7].

REFERENCES

[1] ALLEMANG D., HENDLER J., *Semantic Web for the Working Ontologist*, Second Edition: Effective Modeling in RDFS and OWL. Morgan Kaufman, Burlington, USA, 2011.

[2] BANASZAK Z., BOCEWICZ G., Decision Support Driven Models and Algorithms of Artificial Intelligence, Management Sciences Series, Warsaw University of Technology, Faculty of Management, 2011, 237 s.

[3] CZARNECKI, A., ORŁOWSKI, C., *Ontology Engineering Aspects in the Intelligent Systems Development*, Knowledge-Based and Intelligent Information and Engineering Systems, LNAI 6277, Springer, Berlin Heidelberg, 2010.

[4] CZARNECKI A., ORŁOWSKI C., *IT business standards as an ontology domain*, Lecture Notes in Computer Science, Vol. 6922, 2011, 582–591.

[5] DAVIES J., STUDER R., WARREN P. (Eds.), *Semantic Web Technologies*, Trends and Research in Ontology-based Systems, John Wiley & Sons, 2006.

[6] OWL 2 Web Ontology Language Document Overview, W3C Recommendation 27 October 2009, http://www.w3.org/TR/owl2-overview/

[7] KUSHTINA E., RÓŻEWSKI P., ZAIKINE O., *Ontological Model of the Conceptual Scheme Formation for Queuing System*, Management and Production Engineering Review, Vol. 1, No. 1, 2010, pp. 85-97.

[8] SMIRNOV A., LEVASHOVA T., SHILOV N., *Knowledge sharing in flexible production networks*: *a context-based approach*, International Journal of Automotive Technology and Management, Vol. 9, No.1, 2001, 87-109.

fuzzy Description Logic, algorithm of fuzzification,
algorithm of defuzzification, semantic network,
fuzzy knowledge base, ontology in semantic network

Anna BRYNIARSKA*

THE ALGORITHM OF KNOWLEDGE DEFUZZIFICATION IN SEMANTIC NETWORK

In the literature of the semantic networks the knowledge fuzzification has been precisely described by using Fuzzy Description Logic (*fuzzy*DL). In this article the knowledge fuzzification is identified with some interpretation function of *fuzzy*DL in the fuzzy sets algebra. For the fuzzification sets, fuzzy degrees and expressions of *fuzzy*DL is defined the defuzzification as some interpretation of this language in the set algebra. For any defuzzification are defined rules describing axioms and conclusion according to rules of some standard *fuzzy*DL. After this interpretation, in this paper is presented a schema of algorithm of knowledge fuzzification and defuzzification in semantic networks. This algorithm can be further used in implementation in some programming languages.

1. INTRODUCTION

The semantic network can be considered as indexed directed graph [9]. The nodes of this graph are assigned to the states of searching knowledge about some object. The edges of this graph are assigned to knowledge about the relationships between objects pointed by the nodes. Both nodes and edges are described by some terms indicating objects and relationships. The descriptions of nodes are *the individual names*, the descriptions of edges connecting only one node are *the concept names* and the descriptions of edges connecting two nodes are *the role names*. If we consider that t_1, t_2 are the individuals names and C is the name of the concept then it can be written that „ t_1 is C": „ t_1 is an instance of the concept C". Moreover R can be considered as the role name which means that „between objects t_1 and t_2 is a relationship which is an instance of the role R". The terminology of this semantic network consists of the individuals, concepts and roles descriptions. When relationship between the objects is represented by the semantic network then it is called *the assertion*. The assertion

* Opole University of Technology; Faculty of Electrical Engineering, Automatic Control and Informatics, ul. Sosnkowskiego 31, 45-272 Opole, Poland.

„ t_1 is C" is written „ t_1:C" and the assertion about the relationship R between objects names t_1 and t_2 is written „(t_1, t_2):R".

In the context of the semantic network research [6, 10–13], the representation of knowledge in the semantic network can be defined by the attribute language AL of the Description Logic DL [1]. Then knowledge is represented by two systems: the terminology called **TBox** and the set of assertion called **ABox**. The semantic network can be extended for edges which set relationships between concepts and roles. The description of these relationships, the concepts roles, are called axioms and the system which represent them is called **RBox**. Further is presented the syntax of language AL of fuzzy Description Logic *fuzzy*DL [2, 14] and based on articles [3, 4] the semantic of *fuzzy*DL as also the *fuzzy*DL interpretations which are the fuzzification and defuzzification.

2. SYNTAX OF FUZZYDL

The syntax of fuzzyDL is divided into three syntax of TBox, ABox and RBox.

2.1. SYNTAX OF TBOX

The follow names are included to the set of concepts and roles names:

The universal concept T (Top) and *the empty concept* $\perp$ (Bottom).

The universal concept includes all instances of concepts and the empty concept informs about no instance of concept.

Let C, D be names of concepts, R be the name of a role, and m be the modifier. Then complex concepts are:

$\neg C$ – *concept negation*; it means all instances of concepts which are not an instance of concept C;

$C \wedge D$ – *intersection of concepts C and D*; it means all instances of both concepts C and D;

$C \vee D$ – *union of concepts C and D*; it means all instances either of concept C or concept D;

$\exists R.C$ – *existential quantification*; it means all instances of concepts C which are in role R with at least once occurrence of the concept C;

$\forall R.C$ – *universal quantification*; it means all occurrence of concept C which is in role R with some occurrence of concept C;

$m(C)$ – *modification m of concept C*; it means the concept C which is modified by word m. For example m can occur as a word: very, more, the most or high, higher, the highest.

Concepts which are not complex are called *atomic*.

2.2. SYNTAX OF ABOX

For any concepts instances t_1, t_2, the concept name C and the role name R, *the assertions* are „t_1:C", „(t_1,t_2):R". We read them: t_1 is an instance of the concept C, the pair (t_1,t_2) is an instance of the role R.

For any concepts instances t_1, t_2, the concept name C and the role name R, *the assertions with membership degree* α are „$<t_1:C, \alpha>$", „$<(t_1,t_2):R, \alpha>$". We read them: t_1 is an instance with membership degree α of the concept C, the pair (t_1,t_2) is an instance with membership degree α of the role R.

2.3. SYNTAX OF RBOX

For any concepts names C, D, roles R_1, R_2 and assertions φ, ϕ, the axioms are:
$C \subseteq D$ – the concept C is the concept D,
$C=D$ – the concept C is identical with the concept D,
$R_1 \subseteq R_2$ – the role R_1 is the role R_2,
$R_1=R_2$ – the role R_1 is identical with the role R_2.
φ :- ϕ – Horn clause for the assertion φ, ϕ; we read: φ if ϕ.
$\square$ – empty assertion, if with no instances then φ :- $\square$ is read: assertion φ is a fact.

For any concepts names C, D, roles R_1, R_2 and assertions φ, ϕ, the axioms with membership degree α are: $<C \subseteq D, \alpha>$, $<C=D, \alpha>$, $<R_1 \subseteq R_2, \alpha>$, $<R_1=R_2, \alpha>$, $<\varphi$:- $\phi, \alpha>$.

3. FUZZIFICATION

Terms of *fuzzy*DL language are interpreted in chosen algebra of fuzzy sets: $\boldsymbol{F} = <F, \wedge^F, \vee^F, \neg^F, c^F, e^F, 0^F, 1^F, M, F_0>$, where for space $X \cup X \times X$, F is some family of fuzzy sets μ: $X \cup X \times X \rightarrow [0,1]$ which are described as follow.

For any fuzzy set μ there exist exactly two fuzzy sets μ_1:$X \rightarrow [0,1]$ and μ_2:$X \times X \rightarrow [0,1]$ that:

$$\mu(x) = \mu_1(x),\ x \in X \vee \mu_2(x),\ x \in X \times X \tag{1}$$

The family F is a set of all fuzzy sets in algebra $\boldsymbol{F}$, which only apply to mentioned bellow operations and relation, described by *t-norm, s-norm* the triangulation norms [8], conclusion, equality and modification. The operation $\wedge^F$ is intersection of fuzzy sets; $\vee^F$ is a sum; $\neg^F$ is a complement operation; c^F is a function c^F: $F \times F \rightarrow [0,1]$ called *the degree of containment* of fuzzy sets [8]; e^F is a function e^F: $F \times F \rightarrow [0,1]$

called *the degree of equality* of fuzzy sets [8]; the symbol 0^F is any fuzzy set with values 0; the symbol 1^F is any fuzzy set with values 1; M is a set of one-argument operation f:[0,1]→[0,1] called *the modification functions*; F_0 is a subset of F.

Let X is a set of all objects (data copies), which are part of the semantic network and $X\times X$ is a set of all ordered pairs of the set X. Then there can be described the function I which:

1. For the concept instances t assigns certain values $t^I \in X$ and for the pair instances (t_1, t_2) assigns pairs $(t_1^I, t_2^I) \in X\times X$. Most frequently concept instances are associated with data copies. These copies are considered by IT specialists as objects. Thus, the space $X\cup X\times X$ is a set of all data copies. For example specific words in a given location of the computer screen is an instance of data copy and the data copy is also specific relationship between data.

2. For the concept name C assigns fuzzy set C^I: $X\cup X\times X \rightarrow [0,1]$, that for any $x\in X$, $C^I(x)$ and for any $y\in X$, $C^I(x) = C^I((x, y)) = C^I(x, y)$.

3. For the role name R assigns fuzzy set R^I: $X\cup X\times X \rightarrow [0,1]$, equal 0 for arguments from X,

4. For the modifier m assigns a function m^I: $[0,1] \rightarrow [0,1]$, where $m^I \in M$,

5. For assertions and axioms E assigns some, described later number $E^I \in [0,1]$,

6. For the expression $<E, \alpha>$ - assertions and axioms E with membership degree α:- $<E, \alpha>^I = 1$ when $E^I \geq \alpha$, or $<E, \alpha>^I = 0$ otherwise.

3.1. SEMANTIC OF CONCEPTS (TBOX)

For any $x\in X$, concept names C, D, the role name R and the modifier m:

$$T^I(x) = 1 \tag{2}$$

$$\perp^I(x) = 0 \tag{3}$$

$$(\neg C)^I(x) = (\neg^F C^I)(x) \tag{4}$$

$$(C \wedge D)^I(x) = (C^I \wedge^F D^I)(x) \tag{5}$$

$$(C \vee D)^I(x) = (C^I \vee^F D^I)(x) \tag{6}$$

$$(\exists R.C)^I(x) = sup_{y\in X} \{(R^I \wedge^F C^I)(x, y)\} \tag{7}$$

$$(\forall R.C)^I(x) = inf_{y\in X} \{(\neg^F R^I \vee^F C^I)(x, y)\} \tag{8}$$

$$(m(C))^I(x) = m^I(C^I(x)) \tag{9}$$

3.2. SEMANTIC OF ASSERTIONS (ABOX)

For any instance t of concepts C, D and any instances t_1,t_2 of the role R:

$$(t{:}C)^I = C^I(t^I) \tag{10}$$

$$((t_1,t_2){:}R)^I = R^I(t_1{}^I, t_2{}^I) \tag{11}$$

3.3. SEMANTIC OF AXIOMS (RBOX)

For any concept names C, D, roles R_1, R_2 and assertions φ, ϕ:

$$(C \subseteq D)^I = c^F(C^I, D^I) \tag{12}$$

$$(R_1 \subseteq R_2)^I = c^F(R_1{}^I, R_2{}^I) \tag{13}$$

$$(C = D)^I = e^F(C^I, D^I) \tag{14}$$

$$(R_1 = R_2)^I = e^F(R_1{}^I, R_2{}^I) \tag{15}$$

$$(\varphi :\!- \phi)^I = max\{(1 - \phi^I) , \varphi^I\} \tag{16}$$

When the interpretation function I satisfies the condition (2)–(16), then it is called *fuzzification of fuzzyDL*. If after the fuzzification as the result there are only characteristic functions, then this interpretation is called exact. Then it is equivalent to the standard interpretation of description logic DL [1]. Let I is a fuzzification of language *fuzzyDL*, then the expression $<E, \alpha>$ is *satisfied in this interpretation* (what is written: $I \models <E, \alpha>$) iff $<E, \alpha>^I = 1$, meaning $E^I \geq \alpha$.

4. ONTOLOGY AND FUZZY KNOWLEDGE BASE

Let space $X \cup X \times X$ is a finite set of all considered data copies. The *ontology* is the specific description of these copies, defined as *Ont* = <*TBox, ABox, RBox*>, where: *TBox* is a finite set of terms describing concepts and roles; *ABox* is a finite set of assertions created from concepts and roles terms from the *TBox* set; *RBox* is a finite set of axioms, containing only terms from the *TBox* set. Moreover, expressions <E, α> can be included into ABox or RBox.

Consider the finite set *Fuz,* which corresponds to the set of all possible in practice realization of fuzzification accepted by some group of experts (agents). These experts

make expressions fuzzification in *fuzzy*DL. Then the set *Fuz* is called *the fuzzification space*. In the fuzzification process is made *the fuzzy knowledge base*: $K = <Fuz, V, Ont>$, where: V is a function called *the fuzzy confidence range,* which for concepts and roles assigns some sets of their fuzzification $I \in Fuz$ and *Fuz* is the fuzzification functions set.

Furthermore, all expressions $<E, \alpha>$ belonging to ABox or RBox are satisfied in some interpretation from the *Fuz* set. The expression $<E, \alpha>$ of *fuzzy*DL is *the fuzzy logic consequence of knowledge base K* (what is written: $K|- <E, \alpha>$) iff when it is satisfied in any fuzzy interpretation $I \in Fuz$.

We are looking for the family of subsets of the set $X \cup X \times X$, in which the description logic expression would be interpreted. Similar to the statistic where the confidence ranges are used, it is considered that most important is that all experts accept the membership degrees of objects of fuzzy set which is fuzzification of some concepts and roles belonging to the fuzzy knowledge base $K = <Fuz, V, Ont>$.

Furthermore, if for some fuzzification $I \in Fuz$, of the concept C or the role R, these degrees belong to one of these sets (17) or (18) of the fuzzy confidence range V accepted by all experts, then it can be assumed that elements belonging to this confidence range of the fuzzy set made some subset of space X or $X \times X$.

$$V(C) \subseteq \{\alpha: \text{for some instances } t \text{ of concept } C \text{ and some } I \in Fuz,\ \alpha = (t:C)^I\} \quad (17)$$

$$V(R) \subseteq \{\alpha: \text{for some instances } (t_1,t_2) \text{ of role } R \text{ and } I \in Fuz,\ \alpha = ((t_1,t_2):R)^I\} \quad (18)$$

Other words, experts consider the knowledge about fuzzy degree belonging to the fuzzy confidence range, as adequate knowledge within fuzzification with was made. That experts approach is a defuzzification of knowledge about elements belonging to the some subset space. Therefore designation of such subsets will be identified as knowledge defuzzification about objects belonging to the space X or $X \times X$.

5. FUZZIFICATION ALGORITHM IN DL LOGIC

In order to apply the general defuzzification method, the following task should be solved.

Input: set of copies of processed data $X \cup X \times X$; atomic concepts and roles; ontology $Ont = <TBox, ABox, RBox>$; the fuzzification functions set $Fuz = \{I_1, I_2, \ldots, I_n\}$; the fuzzy confidence range V. These data are collected in knowledge base.

Output: answer for question: if $<Fuz, V, Ont>$ is a fuzzy knowledge base?

The algorithm which answer this question is called *the fuzzification algorithm in DL logic*. After definition of fuzzy knowledge base, the algorithm can be formulated as follow:

1. Number data copies from *X* set;
2. Create an array of atomic concepts and roles described in TBox;
3. Create an array of assertion from ABox set;
4. Create an array of axioms from RBox set;
5. Create an array of membership function μ_E values, for expression E, described in step 2 and particular numbers of data copies, separate for all fuzzification functions;
6. Create an array of expressions <E, α> belonging to ABox or RBox;
7. Create an array of fuzzy degrees defined by function V for all expressions E described in step 2;
8. Check condition (17) and (18), for all expressions E described in step 2;
9. Check if expressions <E, α> are satisfied for some fuzzification from *Fuz* set*;*
10. If results of steps 8 and 9 is positive then <*Fuz, V, Ont*> is fuzzy knowledge base.

Algorithm's properties: procedures described in steps 1-7 create knowledge base and at the same time determined the system <*Fuz, V, Ont*>. It allows for application in standard programming language as well as application of classical data processing algorithms. If the knowledge base $K = <Fuz, V, Ont>$ would be recognized as fuzzy knowledge base, then the same procedures can be used in defuzzification algorithm.

6. DEFFUZIFICATION OF FUZZYDL

How *fuzzy*DL logic is interpreted in DL logic?

U. Straccia in paper [13] proposed interpretation in some algebra of optimally selected classes of fuzzy degrees. Since this interpretation does not refer to concepts and roles instances, it is not compatible with standard semantic of DL logic[1], it is proposed a different interpretation of *fuzzy*DL in DL logic.

The function $(.)^{Def}$ is called the defuzzification interpretation or defuzzification of the knowledge base $K = <Fuz, V, Ont>$, if for any concepts *C, D,* roles *R, R_1, R_2,* and concepts instances t, t_1, t_2, these formulas are true:

$$\bot^{Def} = \varnothing,\ \mathrm{T}^{Def} = X \tag{19}$$

If *C* is an atomic concept then:

$$C^{Def} = X_C, \text{ where } X_C \subseteq X, \tag{20}$$

$x \in X_C$ iff the fuzzification $I \in Fuz$ exists and $(t{:}C)^I \in V(C)$ and $x = t^{Def}$

$$R^{Def} = (X \times X)_R, \text{ where } (X \times X)_R \subseteq X \times X, \tag{21}$$

$(x, y) \in (X \times X)_R$ iff the fuzzification $I \in Fuz$ exists and $((t_1,t_2){:}R)^I \in V(R)$ and $x = t_1{}^{Def}$

If C is any concept then:

$$(\neg C)^{Def} = X \backslash C^{Def} \tag{22}$$

$$(C \vee D)^{Def} = C^{Def} \cup D^{Def} \tag{23}$$

$$(C \wedge D)^{Def} = C^{Def} \cap D^{Def} \tag{24}$$

$$(\exists R.C)^{Def} = \{x \in X\text{: exists } y \text{ such that } (x, y) \in R^{Def} \text{ and } y \in C^{Def}\} \tag{25}$$

$$(\forall R.C)^{Def} = \{x \in X\text{: for any } y, \text{ if } (x, y) \in R^{Def}, \text{ then } y \in C^{Def}\} \tag{26}$$

$$(t{:}C)^{Def} \text{ iff } t^{Def} \in C^{Def} \tag{27}$$

$$((t_1,t_2){:}R)^{Def} \text{ iff } (t_1{}^{Def}, t_2{}^{Def}) \in R^{Def} \tag{28}$$

$$(C \subseteq D)^{Def} \text{ iff } C^{Def} \subseteq D^{Def} \text{ and } (C = D)^{Def} \text{ iff } C^{Def} = D^{Def} \tag{29}$$

$$(R_1 \subseteq R_2)^{Def} \text{ iff } R_1{}^{Def} \subseteq R_2{}^{Def} \text{ and } (R_1 = R_2)^{Def} \text{ iff } R_1{}^{Def} = R_2{}^{Def} \tag{30}$$

$$(\varphi{:}\text{-}\ \phi)^{Def} \text{ iff } \varphi^{Def} \text{ if } \phi^{Def}, \text{ for assertions } \varphi, \phi \tag{31}$$

$$(\varphi :\text{-}\ \square)^{Def} \text{ iff } \varphi^{Def}, \text{ for assertion } \varphi \tag{32}$$

$$<E\ ,\ \alpha>^{Def} \text{ iff } E \text{ is satisfied in degree } \alpha, \textit{ if } K|\text{-}\ <E,\ \alpha>, \tag{33}$$

E is the assertion or axiom of knowledge base K

The axiom is called adequate if its defuzzification is true for any knowledge base K. From presented definition of *fuzzy*DL semantic and definition of DL semantic [1] apparent that all axioms satisfied in DL logic are also adequate in *fuzzy*DL.

The condition (33), when the defuzzification $(.)^{Def}$ is applied for the axiom E, enable to search for biggest degree α, such that E^{Def}, if $K|\text{-}\ <E,\ \alpha>$ [2]. This analysis is useful for verification by experts various options of describing the set *Fuz* and the function V of fuzzy confidence range.

7. CONCLUSION

It can be notice that proposed algorithm of *fuzzy*DL fuzzification allows to get fuzzy data for different fuzzification method presented in the literature: fuzzy control [8], fuzzy knowledge [6], optimization of fuzzification process [10–14], defuzzification process in production semantic networks [5]. According to presented definition of defuzzification these data can be used to create array of data obtained in defuzzification process. This new algorithm is called *the defuzzification algorithm in DL logic*.

Proposed algorithms can be used in implementation in programming language to produce software which would present fuzzification and defuzzification of knowledge in semantic networks. Moreover, this software would implement searching fuzzy knowledge in semantic network.

ACKNOWLEDMENT

Work co-financed by European Social Fund

REFERENCES

[1] BAADER, F., CALVANESE, D., MC GUINNESS, D., NARDI, D., PATEL-SCHNEIDER, P. (eds.), *The Description Logic Handbook. Theory, Implementation and Application,* Cambridge University Press, 2003.

[2] BOBILLO, F., STRACCIA, U., In: IEEE World Congress on Computational Intelligence *FuzzyDL: An Expressive Fuzzy Description Logic Reasoner.* Hong Kong, pp. 923–930, 2008.

[3] BRYNIARSKA, A., [in Polish] *Fuzzification and defuzzification of fuzzy knowledge in semantic networks,* In: Zeszyty Naukowe WETiI Politechniki Gdańskiej, Gdańsk, pp. 389–394, 2011.

[4] BRYNIARSKA, A., [in Polish] *Adequate defuzzification of fuzzy knowledge in semantic networks,* In: Proc. XIII International PhD Workshop OWD, Wisła, pp. 249–254, 2011.

[5] BRYNIARSKA A., [in Polish] *Fuzzification and defuzzification of knowledge in the semantic network of production systems*, PAR 4/2012, s. 98–104, Warszawa 2012

[6] BRYNIARSKA A. [in Polish] *Fuzzy knowledge search in the semantic network and its representation in data systems,* In.: Computer Technologies in Science, Technology and Education, Computer science in the age of XXI century, s.13–23, Radom 2012.

[7] GALANTUCCI, L. M., PERCOCO, G., SPINA, R., *Assembly and Disassembly by using Fuzzy Logic & Genetic Algorithms*, International Journal of Advanced Robotic Systems, Volume 1 Number 2, pp. 67–74, 2004.

[8] KACPRZYK, J., [in Polish] *Multistage fuzzy control,* WNT, Warszawa, 2001.

[9] KOWALSKI, R.A., *Logic for Problem Solving.* New York: North Holland, 1979.

[10] PAN, J. Z., STAMOU, G., STOILOS, G., THOMAS, E., *Expressive Querying over Fuzzy DL-Lite Ontologies.* In: Scalable Querying Services over Fuzzy Ontologies, 17th International World-Wide-Web Conference, Beijin, 2008.

[11] SIMOU, N., MAILIS, T., STOILOS, G., STAMOU, S., *Optimization Techniques for Fuzzy Description Logics.*, In: Proc. 23rd Int. Workshop on Description Logics (DL2010), Waterloo, Canada, pp. 244–254, 2010.

[12] SIMOU, N., STOILOS, G., TZOUVARAS, V., STAMOU, G., KOLLIAS, S., *Storing and Querying Fuzzy Knowledge in the Semantic Web,* In: Proc. 4th International Workshop on Uncertainty Reasoning for the Semantic Web Sunday 26th October, Karlsruhe, Germany, 2008.
[13] STRACCIA, U. *Transforming Fuzzy Description Logics into Classical Description Logics*. In Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA–04), 2004.
[14] STRACCIA, U. *A Fuzzy Description Logic*. In Proceedings of AAAI–98, 15th National Conference on Artificial Intelligence, pages 594–599, Madison, Wisconsin, 1998.

formalisation, BPMN, architectural decisions,
service composition, SOA systems

Szymon KIJAS, Andrzej ZALEWSKI *

FORMALISING ARCHITECTURAL DECISIONS FOR SERVICE COMPOSITION

Capturing architectural knowledge can considerably support system maintenance and evolution. However, architectural decision is an ambiguous and intrinsically complex concept. Developing its precise, formalised definition can make architectural decisions easier to comprehend and enable integration of existing architecture models with architectural decisions in a seamless way. However, such a formalisation cannot be achieved in general but only for chosen kind of architectural decisions and a chosen kind of software architecture. Such a strict definition has been developed for architectural decisions representing service compositions. The BPMN models have been chosen as a semantic domain. The introduced definition of architectural decision has enabled a number of the relations between architectural decisions to be defined, which can then be used to capture the decisions defining the structure of a service composition, the choice of composed services and the changes introduced during the evolution of such a service composition. These formalised relations can also be detected automatically, which enables the development of a tool support for decision-making and evolution documentation.

1. INTRODUCTION

Capturing architectural knowledge [4] with architectural decisions [1] can substantially facilitate software maintenance and evolution. However, the concept of architectural decision is rather broad and vague, and can represent a variety of qualities (e.g. design, its properties, general design assumptions) – compare Kruchten's classification into ontocrises, anticrises, pericrises and diacrisies [4]. Lack of a precise definition and textual representation [1], [2], [3], make architectural decision and the de-

* Institute of Control and Computation Engineering, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland.

rived concepts (e.g. the relations between architectural decisions) ambiguous, complex and difficult to comprehend [6].

We show that the notion of architectural decision can be precisely defined for a certain type of architectural decision (here: ontocrises) in the context of a concrete type of software architecture (service compositions of service oriented architecture) and its models (BPMN).

The semantics of an architectural decision is given by the structure of the corresponding BPMN [10] model of service composition. By precisely defining the notion of architectural decision, one can also define the relation between those decisions in an unambiguous way. These relations have been used both to capture the structure of service composition, as well as changes to service compositions, which are made during the evolution. The entire model is designed for capturing architectural knowledge concerning service compositions and the evolution of service compositions.

2. SERVICE COMPOSITIONS AS ARCHITECTURAL DECISIONS

We propose to represent service compositions as a superposition of architectural decisions, which define the structure of a composite service. Such an architectural decision is, in fact, a hierarchical, recursive architectural decision, i.e. more complex composition can be composed out of simpler ones, which can also be composed out of even simpler ones.

Definition 1. Service composition is a recursively composite "ontocrise" [4] (existential architectural decisions), which can be one of the following:

1. A trivial service composition, i.e. a simple process consisting of a single service invocation. Such a trivial decision concerns the choice of one service out of a number of available ones;
2. A sequential composition (fig. 1) – a series of service compositions ("sub-compositions"), which are executed one after another. It is represented by the relation *isSeriallyComposedOf*:

$$SC_i \; isSeriallyComposedOf \; \{SC_{i1}, SC_{i2}, \ldots, SC_{in}\}$$

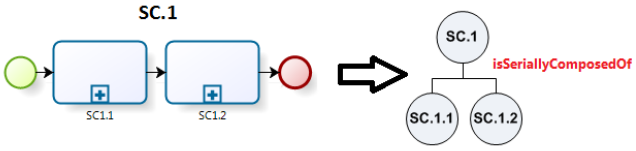

Fig. 1. Semi-formal meaning of the "isSeriallyComposedOf" relation

3. A parallel composition (fig. 2) – a number of "sub-compositions" executed simultaneously. It is represented by the "*isParallellyComposedOf*" relation.

$$SC_i \ isParallellyComposedOf \ \{SC_{i1}, SC_{i2}, \ldots, SC_{in}\}$$

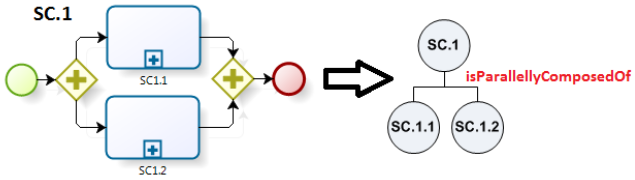

Fig. 2. Semi-formal meaning of the "isParallellyComposedOf" relation

4. A conditional composition (fig. 3) – one of the "sub-compositions" is chosen to be executed according to the fulfilment of a certain condition. It is represented by the "*isConditionallyComposedOf*" relation.

$$SC_i \ isConditionallyComposedOf \ \{SC_{i1}, SC_{i2}, \ldots, SC_{in}\}$$

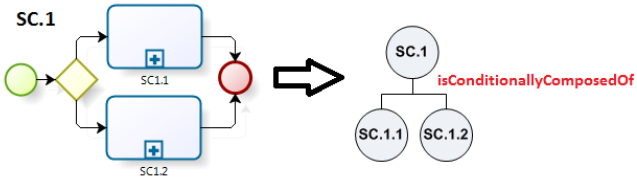

Fig. 3. Semi-formal meaning of the "isConditionallyComposedOf" relation

Every service composition can be captured as a recursive composition as defined above. For example – the service composition given in fig. 4 can be represented with the relations of definition 1 – see the service composition tree in fig. 5. The simplest sub-processes contain just a single task. More complex structures are created by applying the relations of fig. 5, i.e. serial and parallel process composition. Each of these compositions is an architectural decision. The actual service invocations are at the leaves of a composition tree.

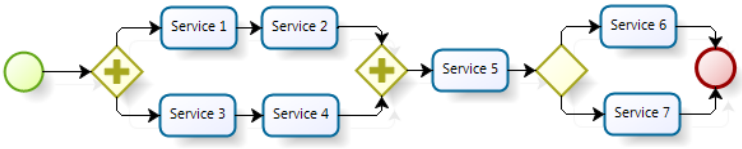

Fig. 4. Example of service composition model in BPMN

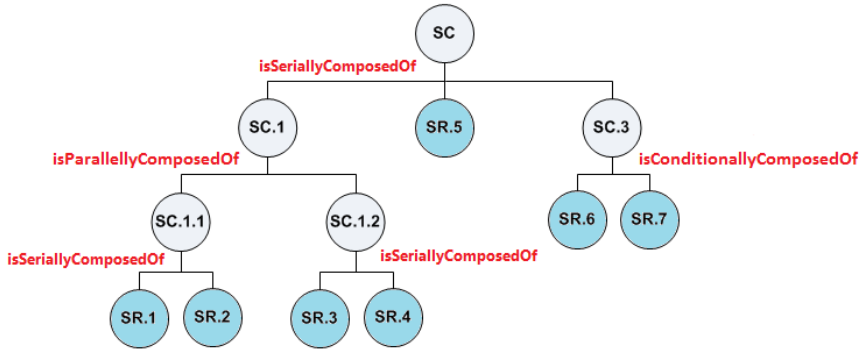

Fig. 5. Rules of service composition (for model in fig. 4)

As the compositions on all the levels and trivial compositions (service invocations) are, in fact, architectural decisions, they can be represented as in [1-3] – indicating considered options and choice rationale. The meaning of each such architectural decision is a BPMN model corresponding to a certain architectural decision.

3. RELATIONS BETWEEN ARCHITECTURAL DECISIONS OF A SINGLE SERVICE COMPOSITION

1. Is...composedof relation

There are three types of relations that are intended to indicate the way of service composition: "*isSeriallyComposedOf*", "*isParallellyComposedOf*" and "*isConditionallyComposedOf*". A description of these types of relations has been presented at the end of section "II".

2. Influences relation

Definition 2. The **"***Influences***"** relation represents a situation in which the input of one service-composition intersects with the output of some other composition, i.e. one composition is using data produced by the other one. An example illustrating this relation has been presented in fig. 6: the architectural service composition SC1 produces data consumed by the composition SC2.

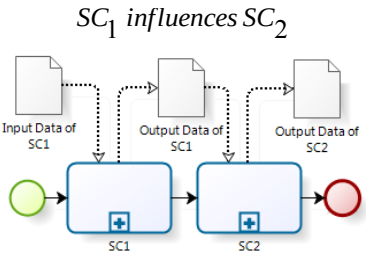

Fig. 6. BPMN model contained semi-formal meaning of "Influences" relations

4. EVOLUTIONARY RELATIONS BETWEEN ARCHITECTURAL DECISIONS

The evolution of service compositions is about making changes to service composition. As a result a new version of service composition and by the same a new version of architectural decision is created. The changes made to service composition can be represented with evolutionary relations, which include "isRefactoringOf", "isExtensionOf" and "isSimplificationOf" relations presented beneath.

The models presented in figures 7–9 have been used to illustrate the semantics of the evolutionary relations.

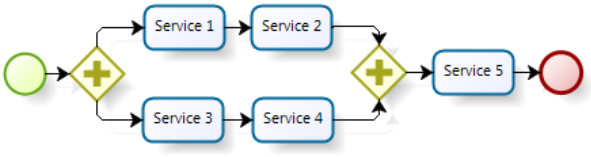

Fig. 7. Service composition model – model 1

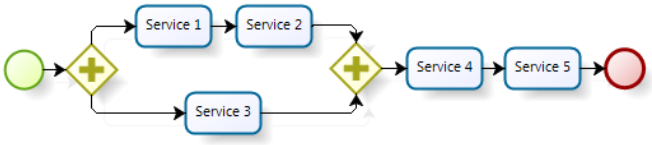

Fig. 8. Service composition model – model 2

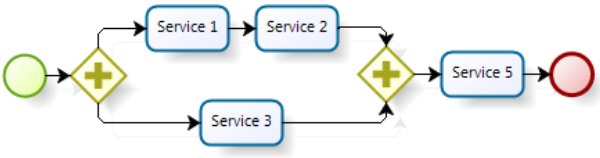

Fig. 9. Service composition model – model 3

1. IsRefactoringOf relation

Definition 3. The *"isRefactoringOf"* relation occurs between the initial architectural decision and its modified counterpart, when the modified composition does the same computing (i.e. invokes the same services) as the initial one, but in a different order.

"isRefactoringOf" means that the collection of services whose elements are the leaves of a service composition tree remain unchanged after the evolution step:

$$collectionOfServices(SC) = collectionOfServices(SC') \Rightarrow SC' \; isRefactoringOf \; SC$$

The architectural decisions representing the compositions of fig. 7 and fig. 8 are in the *"isRefactoringOf"* relation, which is illustrated in fig. 10:

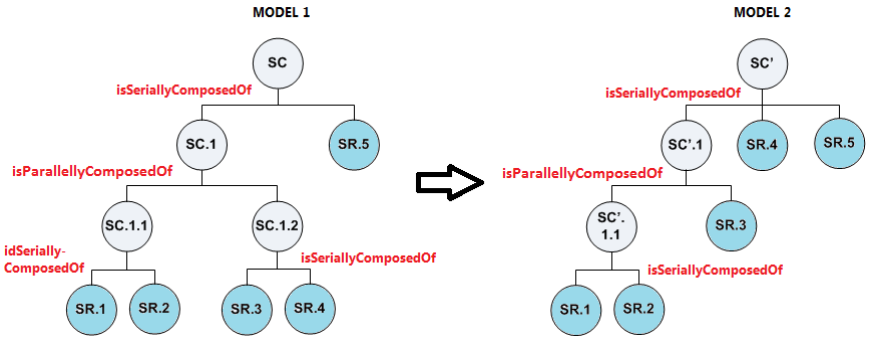

Fig. 10. Representation of the "isRefactoringOf" relation between model 1 and model 2 [the collections of the invoked services in case model 1 and model 2 are exactly the same]

"*isRefactoringOf*" is a symmetric relation, i.e.:

$$SC' \; isRefactoringOf \; SC \wedge SC \; isRefactoringOf \; SC' \equiv true$$

2. IsExtensionOf relation

Definition 4. The *"isExtensionOf"* relation occurs between the initial architectural decision and its modified counterpart, when the modified composition extends the computing made by the initial composition. Formally, this means that the difference between the collection of services invoked by the modified service composition (the leaves of a service composition tree) and the initial service composition is not an empty collection, i.e. the latter one is a strict sub-collection of the former:

$$collectionOfServices(SC') \setminus collectionOfServices(SC) \neq \mathrm{O} \Rightarrow SC' \; isExtensionOf \; SC$$

The service composition of fig. 7 extends the composition of fig. 9 – compare the corresponding composition trees in fig. 11.

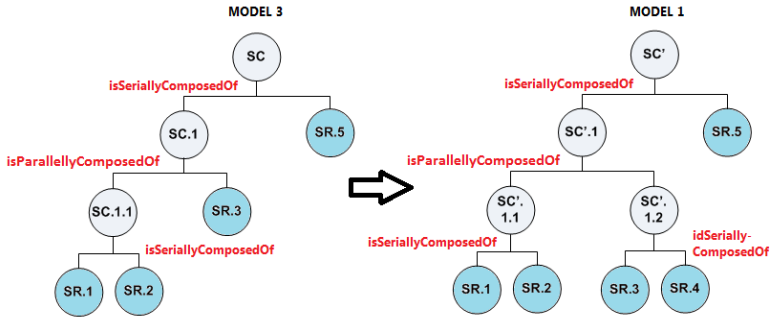

Fig. 11. Illustration of the "isExtensionOf" relation between model 3 and model 1. [the collection of services invoked in model 3 has been extended by "Service 4"]

3. IsSimplificationOf relation

Definition 5. The *"isSimplificationOf"* is an inversion of the "*isExtensionOf*" relation:

$$SC' \text{ isSimplificationOf } SC \Leftrightarrow SC \text{ isExtensionOf } SC'$$

Formally, this means that the collection of services invoked by the initial composition is a strict sub-collection of the collection of services invoked by the modified composition:

$$\text{collectionOfServices}(SC') \subset \text{collectionOfServices}(SC) \wedge$$
$$\text{collectionOfServices}(SC') \neq \text{collectionOfServices}(SC) \Rightarrow$$
$$SC' \text{ isSimplificationOf } SC$$

The composition of fig. 9 is a simplification of a composition of fig. 7.

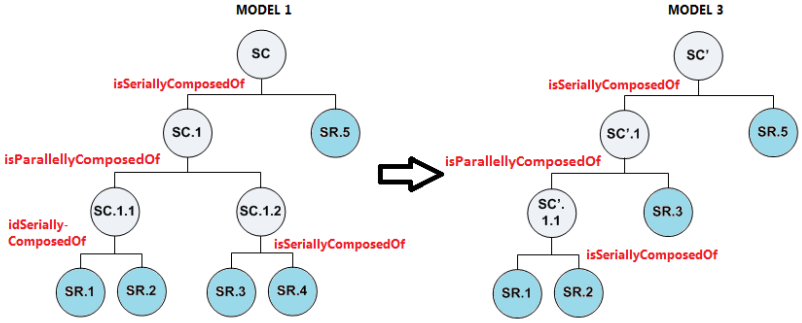

Fig. 12. Illustration of the "isSimplificationOf" relation between model 1 and model 3 [from the collection of services invoked in model 1 "Service 4" has been removed]

5. INTEGRITY CONSTRAINTS

The integrity constraints connected with the relations described in sections "III" and "IV" have been presented below.

1. Integrity constraint I

Architectural decision "SC_i" cannot be in "*influences*" or in any of the *"is...ComposedOf"* relations with itself.

$$\forall SC_i : \quad SC_i \text{ influences } SC_i \equiv \text{false}$$
$$\forall SC_i : \quad SC_i \text{ is...ComposedOf } \{SC_i\} \equiv \text{false}$$

2. Integrity constraint II

If two architectural decisions are in the "*Influences*" relation, then these cannot be in the *"is...ComposedOf"* relation, and vice versa. The "*influences*" relation can only be defined between architectural decisions that do not represent a composition and its sub-compositions.

$$\forall SC_i, \forall SC_j, SC_i \neq SC_j : \quad SC_i \text{ influences } SC_j \wedge SC_i \text{ is...ComposedOf } \{SC_j\} \equiv \text{false}$$
$$\forall SC_i, \forall SC_j, SC_i \neq SC_j : \quad SC_i \text{ influences } SC_j \vee SC_i \text{ is...ComposedOf } \{SC_j\} \equiv \text{true}$$

3. Integrity constraint III

The relations: *isRefactoringOf*, *isExtensionOf*, *isSimplificationOf* are mutually exclusive (any pair of service compositions "SC_i" and "SC'_i" can only be in one of these relations):

$$\forall SC_i, \forall SC'_i, SC_i \neq SC'_i : \quad SC'_i \; isRefactoringOf \; SC_i \vee$$
$$SC'_i \; isExtensionOf \; SC_i \vee SC'_i \; isSimplificationOf \; SC_i \equiv true$$
$$\forall SC_i, \forall SC'_i, SC_i \neq SC'_i : \quad SC'_i \; isRefactoringOf \; SC_i \wedge$$
$$SC'_i \; isExtensionOf \; SC_i \equiv false$$
$$\ldots$$
$$\forall SC_i, \forall SC'_i, SC_i \neq SC'_i : \quad SC'_i \; isExtensionOf \; SC_i \wedge$$
$$SC'_i \; isSimplificationOf \; SC_i \equiv false$$

6. RELATED WORK AND DISCUSSION

The definition of architectural decision presented in section "II" adopts a similar approach as in [1], i.e. architectural decisions are connected with certain parts of system's architecture (denoted in [1] as "design fragments"), which in our case are parts of service composition models in BPMN. Naturally, every decision can be supplemented with the information contained in textual representations presented in [2], [3], which enables the design rationale to be captured.

The semi-formal semantics have also been defined for the relations between architectural decisions. Thanks to that, the relations: Influences, isRefactoringOf, isExtensionOf, isSimplificationOf can even be identified automatically, or captured while developing a modified service composition during an evolution step. Let us observe that the already classical model for capturing architectural decisions and knowledge by Zimmerman et al. [5] has been founded on an informal definition of architectural decisions and relations between them, and so the relations had to be "manually" indicated by the knowledge engineers. The model of [5] was designed as a versatile tool, while our system is supposed to be a special purpose one (intended for service compositions).

7. CONCLUSION AND OUTLOOK

Semi-formal semantics of architectural decisions comprising service compositions have been defined. They prove that it is possible to formalise architectural decisions, though such a formalisation has to be targeted to a certain kind of architecture and its models. The formalised notions of the relations between architectural decisions make them strict and enable automated detection. The entire system, after completion, is supposed to be used to support the architectural decision and capturing architectural knowledge produced during the system's evolution.

The existing formal semantics of BPMN defined in [7], [8], [9] give way to further formalisation of our model, and to the development of automated tools supporting the analysis of service composition properties.

ACKNOWLEDGEMENT

This work was sponsored by the Polish Ministry of Science and Higher Education under grant number 5321/B/T02/2010/39.

REFERENCES

[1] JANSEN A., BOSCH J., *Software architecture as a set of architectural design decisions*, In Proceedings of the 5th Working IEEE/IFP Conference on Software Architecture, WICSA, 2005.

[2] TYREE J., ACKERMAN A., *Architecture decisions: Demystifying architecture*, IEEE Software, 22(19–27), 2005.

[3] HARRISON N. B., AVGERIOU P., ZDUN U., *Using Patterns to Capture Architectural Decisions*, IEEE Software, Volume. 24, Issue.4, pp. 38-45, July-Aug. 2007

[4] ALI BABAR M. et al., *Architecture knowledge management*, Theory and Practice, Springer-Verlag, Berlin Heidelberg (2009).

[5] ZIMMERMANN O., et al., *Managing architectural decision models with dependency rela-tions, integrity constraints, and production rules,* Journal of Systems and Software, vol. 82, no. 8, pp. 1249-1267, 2009.

[6] ZALEWSKI A., KIJAS S., *Architecture Decision-Making in Support of Complexity Control*, Lecture Notes in Computer Science, vol. 6285, pp. 501-504, Springer 2010.

[7] DIJKMAN R., DUMAS M., OUYANG C., *Semantics and analysis of business process models in bpmn,* Information and Software Technology, vol. 50, iss. 12, pp. 1281-1294, 2008

[8] RAEDTS I., PETKOVIC M., USENKO Y., VAN DER WERF J., GROOTE J., SOMERS L., *Trans-formation of BPMN models for behaviour analysis*, In Proc. of the 5th MSVVEIS, INSTICC Press, pp. 126-137, 2007

[9] TAKEMURA T., *Formal semantics and verifcation of BPMN transaction and compensa-tion*, In Proc. of the APSCC, IEEE, pp. 284-290, 2008

[10] Documents Associated with Business Process Model and Notation (BPMN), Version 2.0 Release date: January 2011, http://www.omg.org/spec/BPMN/2.0/, 2011.

Social Network, Network Analysis, Network of Services

Krzysztof JUSZCZYSZYN, Paweł STELMACH, Łukasz FALAS *

DYNAMIC NETWORKS OF SERVICES – THE EMERGING PATTERNS OF INTERACTION RESULTING FROM THE COMPOSITION OF WEB SERVICES

We propose an approach, according to which the Web services interoperability and resulting composition schemes may be effectively used to create the network structures reflecting the patterns according to which the services interact. We show how to create so-called networks of Web services which allow to effectively use the network structural analysis and optimization techniques to solve the network composition problems. The service network is created on the basis of the semantic bindings between the services in the repository joined with the actual patterns of the service usage resulting from composition queries. Next we show how available techniques of dynamic network structure prediction and analysis may help to assess the future service usage and resource consumption of the service execution layer. Our approach is illustrated by the real data gathered from the PlaTel platform, dedicated to the service management, provision, composition and execution.

1. INTRODUCTION

The rapid development of contemporary service systems, built in accordance with the SOA (Service-Oriented Architectures) paradigm triggers the development of various methods and algorithms devoted to the analysis of user activity, service usage and overall description of complex service systems [9][10]. Among them the first approach to graph based description of service repositories was proposed in [8]. In this work we extend this approach by demonstrating the application of graph structural analysis [5] to the networks of services and introducing a model of dynamic network of services. This allows us to apply and evaluate the existing link prediction methods to the evolving networks of services. A broad survey of link prediction methods is presented in [6]. It

* Faculty of Computer Science and Management, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland, e-mail: krzysztof.juszczyszyn@pwr.wroc.pl, pawel.stelmach@pwr.wroc.pl, lukasz.falas@pwr.wroc.pl

should be noted that most methods of the link prediction give rather poor results – the best predictors discussed in [2] can identify < 10% of emerging links.

Basing on our previous experience, which shows that the distribution of subgraphs in complex networks is statistically stable and typical for the considered network [4], we claim that it is possible to characterize the network structural changes by statistical data about the evolution of its subgraphs and show that this approach leads to especially good results in the case of service networks.

In the following sections we propose a method for the description of service repositories within a graph based model, then we show an example of structural analysis of such networks, which allows to infer the roles, importance and possible risk level of the services. We also present an example of dynamic network of Web services and propose an application of link prediction methods to infer the future service usage and evolution of service networks.

2. NETWORK OF WEB SERVICES

A Web service s_i typically has two sets of parameters: $s_{i,in}$ for SOAP request (as input) and $s_{i,out}$ for SOAP response (as output). When w is invoked with all input parameters $s_{i,in}$, it returns the output parameters, $s_{i,out}$. We assume that in order to invoke s_i, all input parameters in $s_{i,in}$ must be provided ($s_{i,in}$ are mandatory). In the case of composite services the input parameters for each of its atomic services are provided from two sources. First is the user input (which takes place when the user invokes a composite service, providing initial parameters) and we assume that these parameters are complete and adequately described. The second are the outputs of other atomic services taking part in the same execution plan of the composite service. This requires semantic compatibility between inputs and outputs of atomic services.

The information contained in the SSDL descriptions of Web services is sufficient to create the Network of Services (*NoS*) - a graph model representing all the semantic bindings between services within a given repository. The same concerns the standard approach – WSDL language [8].

The Network of Services is a tuple: $NoS = \langle S, E \rangle$ where:

- $S = \{ s_1 , s_2 , \ldots s_i \}$ is a set of services (stored and described in a repository)
- $E = \{ e_1 , e_2 , \ldots e_i \}$ a set of directed edges (relations) between the services from S, where:

$$(s_i, s_j) \in E \;\; iff \;\; s_{j,in} \subseteq s_{i,out}$$

In other words, the existence of a directed edge (s_i, s_j) in E may be interpreted as a fact that the execution of s_i provides a full set of input parameters needed to invoke s_j.

Having defined the *NoS* we may propose a general approach to the characterization of dynamic patterns of interaction between the Web services, which result from service composition and execution. Note that, the *NoS* represents *all* possible parameter transfers between semantically compatible services in a repository. As the composite services are being composed and executed, only a subset of them may be observed in a system within a given timeframe.

If we decide to observe all the parameter transfers between services in a time window of arbitrary length we may use the *NoS* approach and define the networks representing the service activity within this time window. This assumption leads to the definition of the Dynamic Network of Services (*DNoS*):

$$DNoS = \{ NoS^1, NoS^2, \ldots NoS^t \}$$

where $NoS^t = \langle S, E^t \rangle$, and $(s_i, s_j) \in E^t$ *iff* s_i was executed and provided input parameters for s_j during time window number t.

User queries trigger composition of complex services, which are then executed by the service engine. Thus, *DNoS* stores information about parameter interchange in a service system, which is time-dependent and has a graph representation.

We may notice that this approach is analogous to the representation of dynamic social networks, where the interactions between humans are stored as graphs and analysed on time-window basis [3][7]. In this case we may describe the DNoS model as a *social* network of Web services. We utilise this analogy to propose a Web service usage analysis methodology, which assumes the following steps:

For given service repository and associated service composition framework create the *NoS* model representing semantic service compatibility.

1. Apply structural; graph analysis methods to infer the properties of services.
2. Build DNoS – a series of networks representing the actual service usage.
3. Use link prediction methods to infer the future service usage and parameter flows.
4. Relate predictions to measurable consumption of system resources.

In order to illustrate the above concepts, in the next section we present the first experimental results obtained with the PlaTel service management framework.

3. PLATEL FRAMEWORK – EXPERIMENTAL SETUP

The illustrative example of the creation and analysis of the NoS model will be presented on the basis of repository of services belonging to the PlaTel (Platform for ICT solutions planning and monitoring) framework, supporting business processes in distributed ICT (Information and Communication Technologies) environment based on Service Oriented Architecture (SOA) paradigm [11]. The framework scope of

functionalities is divided into applications that cover the whole life cycle of business oriented ICT applications. The PlaTel approach to service description problem assumes the use of the native service description language, SSDL (Smart Service Description Language) which is proposed as a solution allowing simple description of composite service execution schemes, supporting functional and non-functional description of services. Its functionality includes that of the Web Service Description Language (WSDL), but offers important extensions. A definition of SSDL node types contains all basic data types which allow for the functional and non-functional description of a service, its execution requirements and the description of complex services with conditional execution of their atomic components.

Each of them is associated with the number of sub-nodes allowing for precise description of a service. An important part of the functional description of a SSDL node is *class* attribute, which contains semantic labels describing the input parameters of the service. The labels are taken from domain ontology and used during service composition and the construction of data flow inside a composite service. Thus, the service repositories in PlaTel store all information needed to create the *NoS* and *DNoS* models.

4. EXEMPLARY NETWORK OF SERVICES

For the first experiments on PleTel framework, the repository of services used to build service application for monitoring and property security domain was chosen.

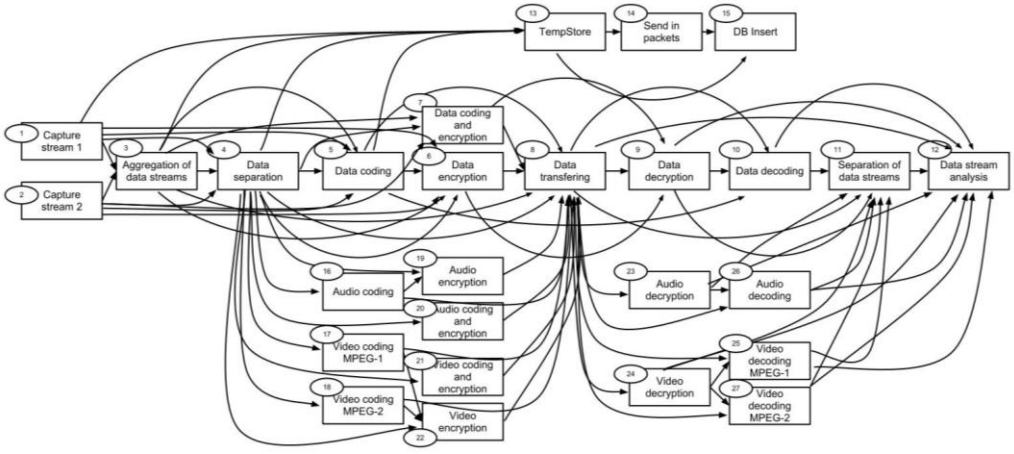

Fig.1. Network of Services for exemplary service repository.

The repository is relatively small and consists of 27 services, for which the *NoS* model was created, according to the definition given in the preceding section. Fig.1

presents the visualisation of the *NoS* , viewed as a directed graph with labelled nodes representing services.

The *NoS* was analysed using standard structural network analysis techniques which returned interesting results.

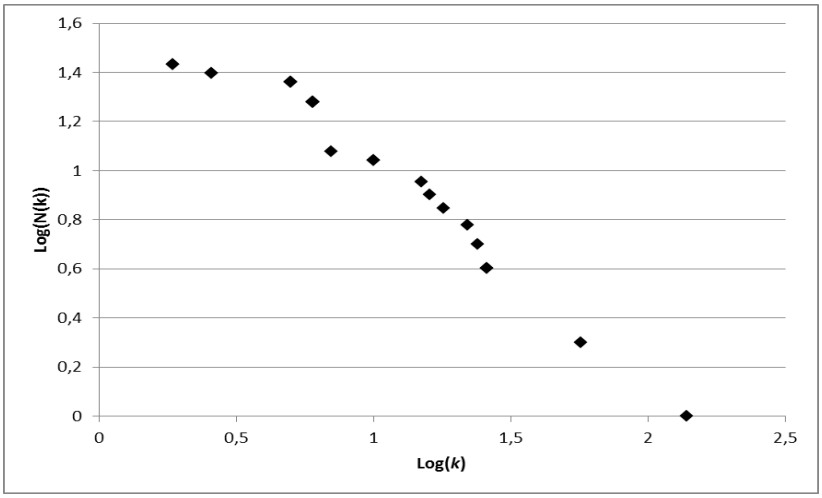

Fig. 2. Node degree (k) distribution for PlaTel service repository.

First of all the node degree distribution was checked – most of the complex networks existing in nature, from social to biological, economic and technology-based show scale free node degree distribution, following the power law [1], the same was confirmed for the *NoS* of PlaTel service repository. Fig. 2 presents the node degree distribution for PlaTel service repository (on log-log axis scale, *k* is the node degree, and *N(k)* – the number of nodes of the degree *k*).

This results confirms observations and conclusions presented in [8] for Web service repositories. The next step was the structural analysis of the network – inferring the node types from their connection patterns, calculating betweenness centralities (which correspond to the relative importance of the node in a graph) and node group analysis.

Fig. 3 presents the PlaTel *NoS* graph created in NetMiner 4.0 network analysis software, with three node groups detected by the standard CNM (Clauset, Newman and Moore) algorithm. We may note that the groups, however detected only on the basis of the graph structure contain the services with corresponding functionalities (coding and encryption – G1, decoding and decryption – G2, storing and stream processing – G3). This may suggest an effective strategy for categorization of Web services in large repositories, where services come from different providers.

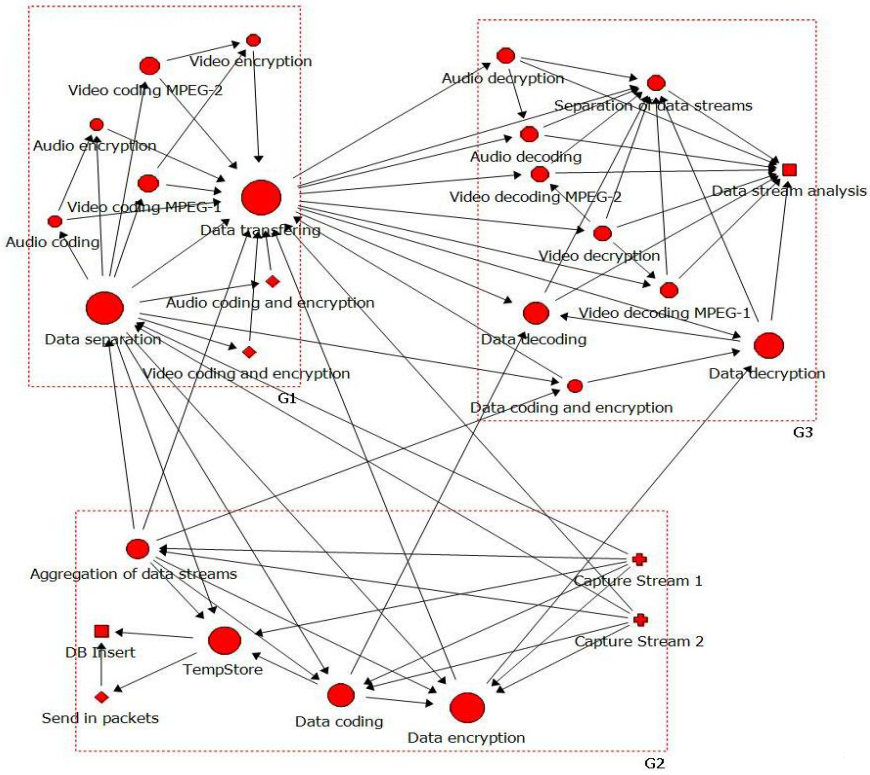

Fig.3. Network size for time windows of different size for WUT dataset.

The node types were detected using approach defined in [5]: *Isolate* nodes do not have any links. *Transmitter* (crosses on Fig.3) has only out links and no in links. *Receiver* (cross) node has only in links, while *Carrier* node (rhombus) has exactly one incoming and one outgoing link. *Ordinary* node (circle) does not fall in any of the above categories. The types allow to classify the nodes (services) in the context of their roles and the importance for the functionality of the service repository. dynamic structural patterns of the investigated network.

The detailed results concerning network nodes of the investigated *NoS* are presented in Table 1. The services *Data transferring* and *Data separation* were assigned the highest betweenness centrality which suggests their key role in the communication between the services in the repository which corresponds to the domain knowledge and typical usage of the PlaTel services.

We argue that the results of such analysis may be effectively used to select services which are important for given domain, and the structural *NoS* analysis may contribute to the risk and resource management in the service systems.

Tab.1. The node characteristics for the PlaTel NoS

	In Degree	Out Degree	Betweenn. Centrality	Node Type
Capture Stream 1	0	5	0	Transmitter
Capture Stream 2	0	5	0	Transmitter
Aggregation of data streams	2	6	0,006462	Ordinary
Data separation	3	11	0,038769	Ordinary
Data coding	4	3	0,006895	Ordinary
Data encryption	5	2	0,018998	Ordinary
Data coding and encryption	2	2	0,001305	Ordinary
Data transfering	12	8	0,166956	Ordinary
Data decryption	3	3	0,009916	Ordinary
Data decoding	3	2	0,006559	Ordinary
Separation of data streams	8	1	0,001972	Ordinary
Data stream analysis	8	0	0	Receiver
TempStore	4	2	0,015385	Ordinary
Send in packets	1	1	0	Carrier
DB Insert	2	0	0	Receiver
Audio coding	1	2	0	Ordinary
Video coding MPEG-1	1	2	0,003077	Ordinary
Video coding MPEG-2	1	2	0,003077	Ordinary
Audio encryption	2	1	0	Ordinary
Audio coding and encryption	1	1	0	Carrier
Video coding and encryption	1	1	0	Carrier
Video encryption	2	1	0	Ordinary
Audio decryption	1	3	0,001972	Ordinary
Video decryption	1	4	0,001972	Ordinary
Video decoding MPEG-1	2	2	0,001972	Ordinary
Audio decoding	2	2	0,001972	Ordinary
Video decoding MPEG-2	2	2	0,001972	Ordinary

5.LINK PREDICTION IN NETWORKS OF SERVICES

For the experiments with the *DNoS* a record of the actual service usage was needed. The dynamic network representing the actual service usage was created, then the link prediction methods were applied in order to assess the future service usage and the structure of the resulting *DNoS.* The experiments were carried on the PlaTel framework, with the following assumptions:

- 5 users took part in the experiment, and 9 types of queries (requirement graphs, representing the user demands for composite services) were invoked ~200 times.

- Queries were served by the PlaTel composer module, with *exact match* semantic filter (assuming exact correspondence between semantic description of requirements and the selected services).
- The resulting dataset (from here denoted as PlaTel dataset) was divided into 80 time windows, corresponding to the 80 *DNoS*s. First 30 were used to train the link prediction algorithms, the remaining 50 were used for verification.
- Prediction evaluation was performed according to the scheme proposed in [2].

The *DNoS*s created were highly dynamic. The number of links emerging and disappearing in the consecutive time windows varied frequently, which was quite different from the situation met in the case of dynamic social networks [12].

For the link prediction problem three algorithms were used: Preferential Attachment (PA), Common Neighbours (CN) and Triad Transition Matrix (TTM). First two are standard link predictors which assume the social-driven behaviour of network nodes: PA assumes the tendency of new links to be adjacent to network hubs, CA tries to connect nodes which have numerous common neighbours.

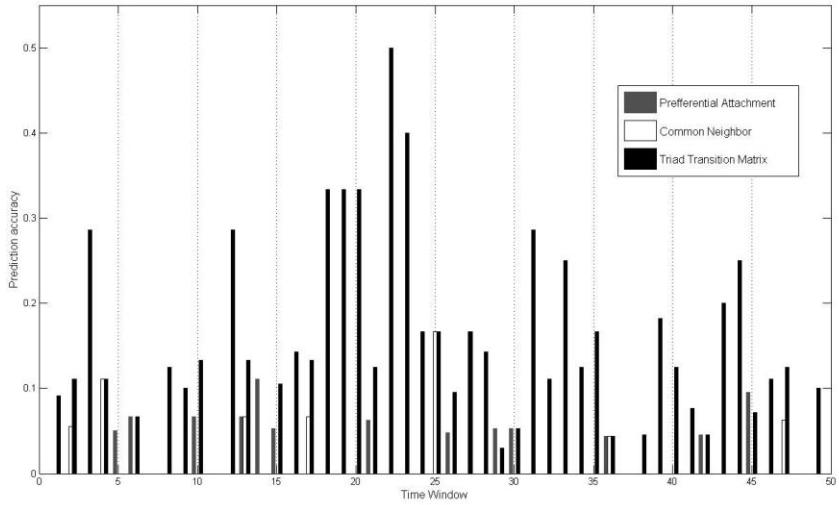

Fig.4. PlaTel dataset – prediction accuracy for PA, CN and TTM predictors.

This approaches have strong grounding in social science and were proven to be effective in the case of social networks. TTM is a novel, domain-independent method, first introduced in [12]. It is based on statistical description of changes in elementary network subgraphs – the triads of nodes. Despite the link prediction problem being hard (prediction accuracy for real-life complex networks are rarely better than 5%) it was shown to be very effective, especially for networks analysed in short time scales [12]. The results for all the above predictors used for the *DNoS*s of PlaTel dataset are

presented on the Fig.4. The average prediction accuracy was 1.3% for CA, 2.7% for PA and 18.7% for TTM. The TTM decisively outperforms the other predictors which leads to the conclusion, that the evolution of *DNoS* is not driven by social evolutionary schemes. The relative performance of CA and PA also confirms this conclusion – in most social network datasets CA outperforms PA. This also suggests, that we may expect similar phenomena for the majority of other link predictors available – most are derived from the observations of social phenomena applied to the complex networks.

Good results for TTM imply also that predictors using time series analysis, subgraph structure mining and network statistics will perform better in the case of dynamic networks of services. We may also note that for some windows all the predictors have zero accuracy. This is caused by the lack of user activity (queries) during these windows and suggests that a methodology for choosing window timespan is needed.

An important fact is also the significant reduction in the computational cost (for all predictors). This is caused by the reduction of possible link space in contrary to social networks, where one can expect n^2 possible links in a n-node network, in the case of *NoS* the complete link space is equal to the number of its links (note that only some of them occur in the *DNoS*). This, however had no influence on the performance of the predictors.

CONCLUSIONS AND FUTURE WORK

The presented approach is quite novel – the only one work suggesting the network approach to the description of service repositories is [8], however only the static approach to the service networks was presented there and no structural analysis or network evolution scenarios have followed. The concepts of NoS and DNoS open vast possibilities of applying various graph and network analysis techniques for the management and evolution discovery of complex service systems. The most attractive and practically important areas of future research are:

- Utilizing all the information stored in service description records (in WSDL and SSDL alike) for the creation of complex service networks.
- Broad analysis of link prediction methods in order to choose appropriate approaches to dynamic service networks.
- Establishing connections between structure prediction of service networks and resource consumption and allocation in service systems.
- Utilizing information about users (who submit composite service queries) during creation and analysis of service networks' models.

REFERENCES

[1] A.-L. BARABÁSI, The origin of bursts and heavy tails in humans dynamics, Nature 435, 207 (2005).
[2] D. LIEBEN-NOWELL, J.M. KLEINBERG, The link-prediction problem for social networks. JASIST (JASIS) 58(7), pp.1019–1031, 2007.
[3] D.BRAHA, Y. BAR-YAM, From Centrality to Temporary Fame: Dynamic Centrality in Complex Networks, Complexity, Vol. 12 (2), pp. 59–63, 2006.
[4] K. JUSZCZYSZYN, K. MUSIAL, P. KAZIENKO, B. GABRYS: Temporal Changes in Local Topology of an Email-Based Social Network. Computing and Informatics 28(6): 763–779, 2009.
[5] S. WASSERMAN, K. FAUST, Social network analysis: Methods and applications, Cambridge University Press, New York, 1994.
[6] L. GETOOR, C. P. DIEHL, Link mining: a survey, ACM SIGKDD Explorations Newslett., Vol. 7, pp. 3–12, 2005.
[7] Z. HUANG, D. K. J. LIN, The Time-Series Link Prediction Problem with Applications in Communication Surveillance,INFORMS Journal on Computing, Vol. 21, No. 2, pp. 286–303, 2009.
[8] S. OH, D. LEE, S. KUMARA, Effective Web Service Composition in Diverse and Large-Scale Service Networks, IEEE Trans. On Services Computing, Vol. 1, No. 1, 2008.
[9] Y. WANG, J. ZHANG, J.VASSILEVA, Effective Web Service Selection via Communities, Formed by Super-Agents, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 549–556, 2010.
[10] P. STELMACH, A. GRZECH, K. JUSZCZYSZYN, A model for automated service composition system in SOA environment. Technological innovation for sustainability : second IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2011, Portugal, November 21–23, 2011, Springer, s. 75–82.
[11] P. STELMACH, K. JUSZCZYSZYN, A. PRUSIEWICZ, P. ŚWIĄTEK, Service Composition in Knowledge-based SOA Systems. New Generation Computing, vol. 30, no 2&3 (2012).
[12] K. JUSZCZYSZYN, K. MUSIAL, M. BUDKA: Link Prediction Based on Subgraph Evolution in Dynamic Social Networks. SocialCom/PASSAT 2011, pp. 27–34.

social contagion, early detection,
social network, local hubs

Jakub PORZYCKI, Jarosław WĄS*

NOVEL ALGORITHMS OF SENSORS DETECTION IN SOCIAL NETWORK

Early detection of social contagion gives a chance for proper reaction. Regardless of its type – deadly disease or just a new trend in music - early knowledge of social contagion outbreak is very valuable. Nevertheless, current methods of social trends analysis give contemporaneous information. Nicholas Christakis and James Fowler in their paper "Social Network Sensors for Early Detection of Contagious Outbreaks"[2] present noteworthy idea of predicting social contagion development using set of individuals (sensors) in society. Presented method is based on fact that individuals in the center of network are more likely to be infected sooner, thus they could be used as a sensors that predict future state of whole society. The paper presents alternative algorithms of choosing sensors, without knowledge of social network structure – using only surveys among randomly chosen people. A few methods are discussed, some of them allow for two – three times sooner detection than Christakis-Fowler algorithm.

1. INTRODUCTION

1.1. SENSORS AND THEIR ALLOCATION IN NETWORK

Social contacts network is not homogeneous. One can observe individuals located more centrally than others - important nodes connected with many others. One can also observe a lot of nodes located peripherally. Figure 1 shows an example of social contact network with approximately 300 nodes. Two nodes are highlighted. Although they have similar number of connections, it is clear that in case of some social contagion spreading in this network, centrally located node B is more likely to be infected than node A. Following that thought, it could be concluded than an average central node will be infected earlier than average node of whole population.

* AGH University of Science and Technology, Al. Adama Mickiewicza 30, 30-059 Kraków

This phenomenon is caused by fact, that central nodes have large values of closeness centrality - defined as the sum of distances to all other nodes. Hence individual, which is relatively close to all other nodes in network, could easily be infected.

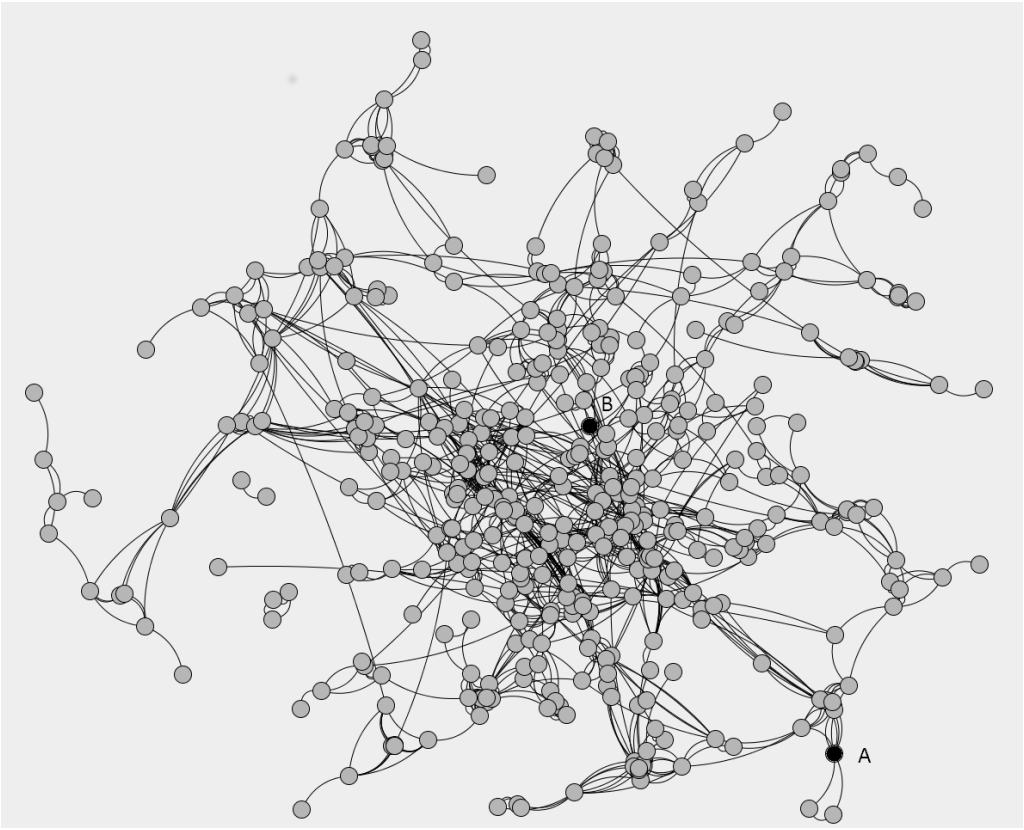

Figure 1. Different position in network

1.2. ADAPTATION CURVE

Adaptation curve is a chart that shows dependence of infected individual number from time. It can be plotted for any contagion spreading in society. At the beginning, very few people have contact with such innovation. Number of infected individuals grows slowly, due to the fact that total number of infected people is relatively small in comparison with population size. Simultaneously, a rapid outbreak occurs. Number of the infected quickly increases until the population is satiated. This mechanism produces classical S-curve of adaptation.

According to the hypothesis given in section 1.1, adaptation curve for sensors should be shifted in time comparing to the one representing whole population. Mentioned shift in time allows early detection. That situation is shown in figure 2. It should be stressed, that contagion outbreak among sensors occurs when whole population is still in slow growth phase.

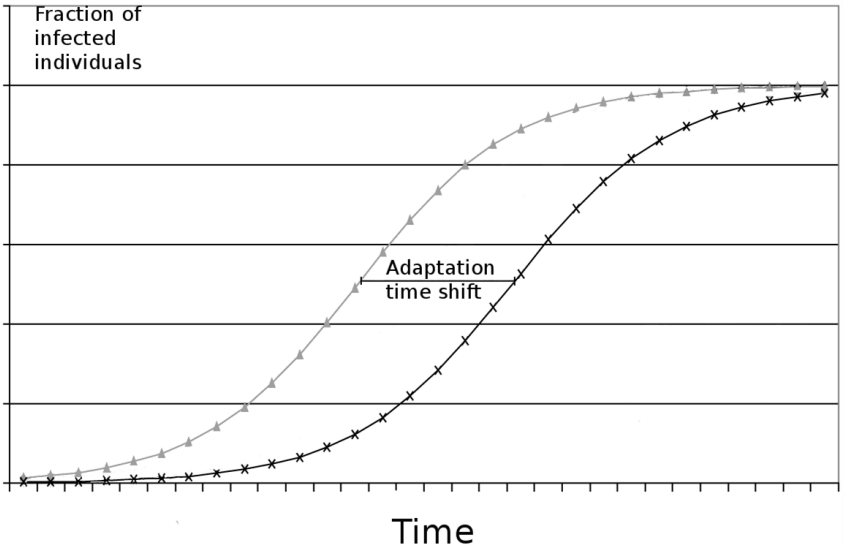

Figure 2. Hypothetic adaptation curves for whole population (grey), and sensors (grey)

1.3. SENSORS DETECTION PROBLEM

Despite of all the advantages of presented method, it has one weakness. In order to detect central nodes in graph (network) – its structure has to be known. However, mapping of social contact network is not an easy task to do. In most cases it is either non ethical or very expensive.

Without knowledge of network topology, none of classical algorithm for finding central nodes can be used. Considering measures like degree centrality, betweenness centrality, closeness centrality, page rank ... – every that needs global information. Thus in this case is useless. Proper algorithm has to operate only on local information.

2. ORIGINAL METHOD

Friendship paradox is a seeming inconsistency, firstly observed by sociologist Scott L. Feld in 1991[4]. The phenomenon is related to the fact that most people have fewer

friends than their friends. Easy explanation of that fact was given by Satoshi Kanzawa: "*You are more likely to be friends with someone who has more friends than with someone who has fewer friends. There are 12 people who have a friend who has 12 friends, but there is only one person who has a friend who has only one friend. And, of course, there is no one who has a friend who doesn't have any friend*" [6].

Formally, let us assume that social network is an undirected graph where set V of vertices corresponds with individuals in network and set E of edges corresponds with friendship relation between individuals. Here we assume that friendship is a symmetric relation. Therefore, if a node has d(v) edges connected, then the average number μ of friends of a random person in the graph is given by equation:

$$\mu = \frac{\sum_{v \in V} d(v)}{|V|} = \frac{2|E|}{|V|} \tag{1}$$

Average number μ_f of friends of a friend of a random person could be modeled by an edge in a graph (a pair of friends) and one endpoint of this edge (one of friend). Degree of selected endpoint can be expressed as:

$$\mu_f = \frac{\sum_{uW \in E} d(v)}{2|E|} = \mu + \frac{\sigma^2}{\mu} \tag{2}$$

where σ^2 is the variance of the degrees in graph [4]. Owing to the fact, that σ^2 as well as μ_f are positive, therefore $\mu_f \geq \mu$. Moreover, if degree distribution in graph is not uniform, then $\mu_f > \mu$ [2].

Christakis and Fowler proposed method of sensors detection based on friendship paradox. They noticed that reversed friendship paradox claims that random friend of a random individual has more friends than him. Therefore an average "random friends" are located more centrally than average individual. It is the property required from sensors. Moreover this algorithm does not require global information about network structure. One only needs a survey among group of randomly chosen individuals in which they point their random friend.

3. NOVEL METHODS

Method presented by Christakis and Fowler is easy and effective, however other methods should be considered. In the paper we discuss three other locally working algorithms of sensors detection:

- Iterative Friendship
- Friend With Most Friend
- Local Hubs

According to tests, described in section 5, none of the algorithms works worse than the original method, and some returns sensors that yield two - three time sooner detection.

3.1. ITERATIVE FRIENDSHIP

Iterative friendship algorithm was implemented in order to check whether iterative repeating of original method could bring any improvements. In this simple algorithm, random friends are asked about their random friends a few times iteratively. As a sensor can be chosen:

- random friend of random individuals

Other possible choices are:

- random friend of random friend of random individuals
- random friend of random friend of random friend of random individuals
- etc …

3.2. FRIEND WITH MOST FRIENDS

The second algorithm is based on an idea that if, on average, the most friends individual have the more central he is. Therefore an average friend with most friends of random individual should be located more centrally than average random friend of random individuals.

Only thing that have to be changed, comparing to the original algorithm, is a question to interviewed person. One has to ask for friend who have most friends rather than random friend. We assume that most people are able to point one of his friends with great amount of connections.

3.3. LOCAL HUBS

Local Hubs detection is the most complex case among the proposed algorithms. It could be interpreted as a combination of two previous algorithms or iterative version of *Friend With Most Friends* algorithm. The aim is to find a node that has more friends than any of his friends.

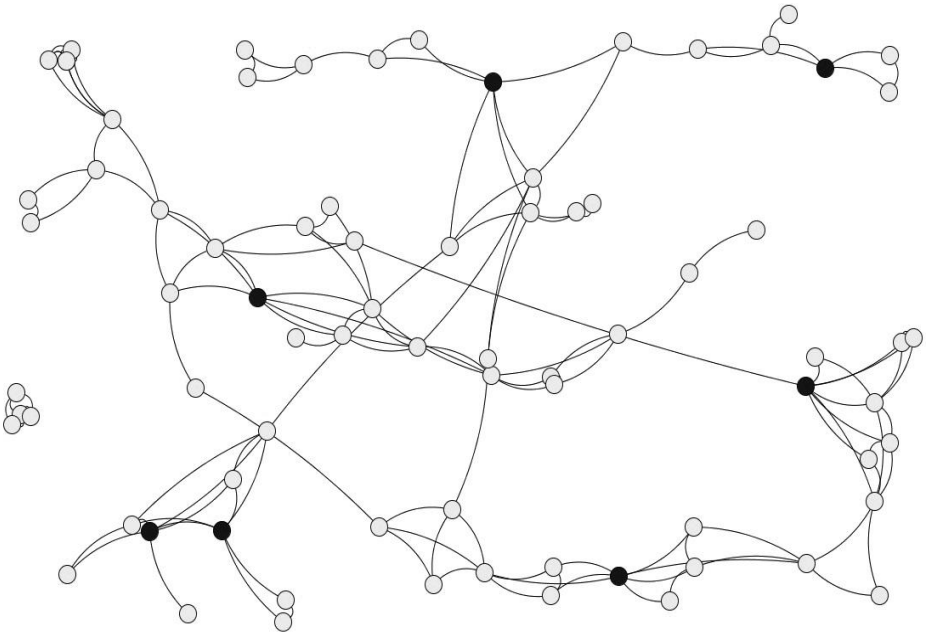

Fig. 3. Local hubs (black) found in small network

As it was mentioned before, the algorithm keeps iterating trough the network, and asks succeeding individuals about his friend with most friends. It is repeated until algorithms reach a node where it has already been. At this point from last two nodes, the one with more friends is chosen as a sensor.

In this method, noteworthy is a fact, that for a given structure and size of network, there is no point in manually choosing a desired number of sensors. Every other presented algorithm needs it. Due to the fact that number of hubs in network is relatively small, after finding some number of hubs, they start to repeat. If hubs repeat too often, for example ten times in a row, it means that vast majority of possible hubs was found. For a set of conditions presented in section 4.1 it usually was around 2% of population.

4. SIMULATION ENVIRONMENT

4.1. SOCIAL NETWORK

The new algorithms proposed, as well as the original method, were tested in a computer simulation of small (100 000 citizens) city. Social network was built based on census data for Poland in year 2002. Simulation reproduces household, workplace,

school and age structure in society. For any given individual (agent) in simulation his social contacts in household, job, school, among friends are represented. Agent belongs to different social groups such a household, a workplace \dots in different time of day.

Created network satisfies three main requirements for artificial social network:

- short mean distance between nodes
- power law node degree distribution
- high clustering coefficient

4.2. CONTAGION PARAMETERS

A typical virus was chosen as an example of social contagion. Individual starts to be infectious 1–2 days after he was infected and stays infectious until 5–7 day after infection. Contagion symptoms appear in 24–72 hour after infection and last 48 to 96 hours. When individuals stop being infectious and symptomatic, they becomes resistant. This is a typical SIR (Susceptible, Infected, Recovered) model of contagion.

In any time step (15 minutes), for every susceptible individual, infection probability is calculated. Infected individuals can change their state (infectious - non infectious, symptomatic - non symptomatic).

5. TESTS

In the figures below following curves styles was used:

- thin line – contagion spread in whole population
- bold line – contagion among sensors returned by original Christakis and Fowler algorithm
- dotted line – contagion among sensors returned by discussed algorithm

Every time unit on a chart corresponds with one time step in simulation. According to assumptions given in section 4.2 one time step represent 15 minutes. Therefore 500 units on chart represent slightly more than 5.2 day.

5.1. ITERATIVE FRIENDSHIP

First of presented algorithms does not bring any significant improvements in early detection. As it can be observed at figure 4 bold and dotted chart have almost the same shape. Regardless how many iterations were done, results are similar. With given parameters, discussed method as well as original algorithm result in 1 to 1.5 day sooner contagious detection.

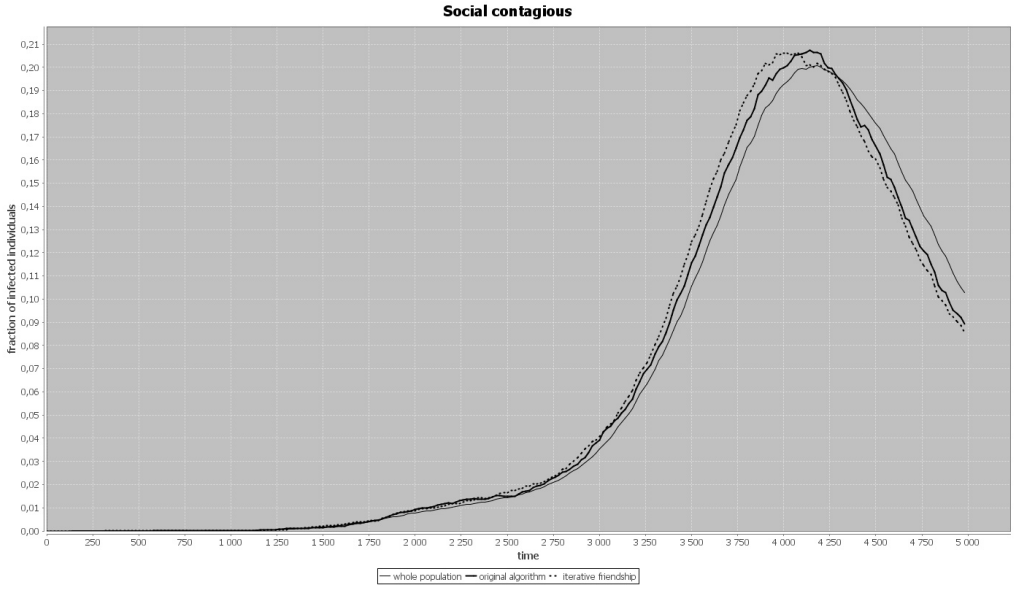

Fig. 4. Comparison of iterative friendship and original algorithm

5.2. FRIEND WITH MOST FRIENDS

Choosing friends with most friends instead of random friends, results in very significant changes of time shift. Figure 5 presents dotted plot shifted two - three times as far as the curve generated by original algorithm. That corresponds with time shift of approximately 3 days.

5.3. LOCAL HUBS

This algorithm returns individuals who are in the center of local part of social network. Therefore it was expected, that this method will result in the biggest time shift. Surprisingly, time shift resulting from this algorithm is very similar to the one granted from *Friend With Most Friend* algorithm. Figure 6 presents comparison of local hubs detection and the original algorithm. Time shift is similar to the one generated with use of *Friend With Most Friend* algorithm.

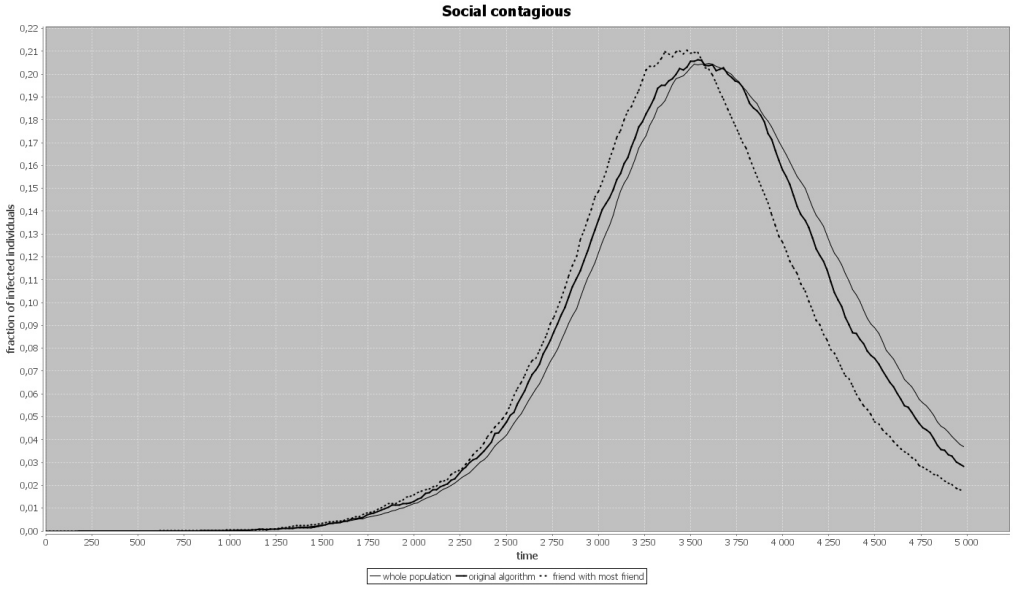

Fig. 5. Comparison of *Friend With Most Friend* and original algorithm

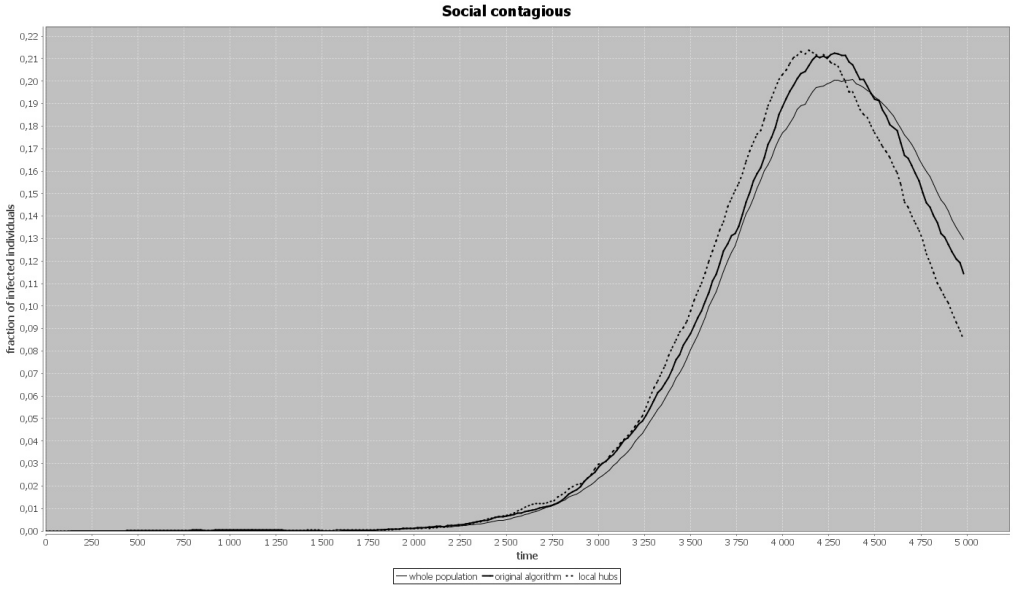

Fig. 6. Comparison of *Local Hubs* and original algorithm

6. DISCUSSION

During our research we found out a noteworthy possibility. Evaluation of original method done by Christakis and Fowler was very similar to results granted from *Friend With Most Friend* algorithm. To explain this phenomenon, a hypothesis that an average individual, when asked about his random friend, with great probability gives info about the one who has a lot of friends, was formulate.

To verify the hypothesis, a survey among 150 people was done. They were firstly asked to give a name of a random friend, then to give a name of their friend with most number of friends. Finally, they were asked if it was the same person. About 23% of them said yes to the last question. Many of the others claimed that friend given in first question also had more friends than average. Of course such a small survey is not enough to be a strong proof and this problem requires further research. However, it may indicate that in fact during evaluation using Christakis and Fowler method one gets sensors close to the sensors generated by *Friend With Most Friend* algorithm.

It is clear, that the sooner some social contagion is detected, the better it will be able to be prevented or taken advantage of. Proposed algorithms improve time shift of original Christakis - Fowler method, without losing its main advantage - possibility to execute without knowledge of social network structure.

Another important property of the original algorithm is a possibility to be executed as a series of surveys among population. In this area every proposed method is slightly harder to conduct than original.

However, some methods seem to be strictly better than others, final choice depends of amount of funds available as well as a type of network. In many cases, proposed algorithms give better results than original. We believe that the proposed methods can be useful tools in social contagion prediction.

REFERENCES

[1] BARABÁSI A., RÉKA A., *Statistical mechanics of complex networks* 2002, Reviews of modern physic No. 74, 48–92

[2] CHRISTAKIS N.A., FOWLER J.H., *Social Network Sensors for Early Detection of Contagious Outbreaks,* PLoS ONE, 2010, Vol. 10, No. 9, Public Library of Science,

[3] COHEN R., HAVLIN S., ben AVRAHAM D. *Efficient immunization strategies for computer networks and populations,* 2003, Phys. Rev. Lett

[4] FELD S.L., *Why your friends have more friends than you do,* American Journal of Sociology, 1991

[5] FRONCZAK A., FRONCZAK P. *Świat sieci złożonych,* 2009, Wydawnictwo Naukowe PWN

[6] KANAZAWA S., *Why your friends have more friends than you and why your girlfriend is a whore,* The Scientific Fundamentalist, 2009

service composition, QoS, social networks

Paweł STELMACH*, Łukasz FALAS*,
Krzysztof JUSZCZYSZYN*

AUTOMATED SERVICE COMPOSITION WITH SOCIAL GRAPH BASED QUALITY CRITERION

In this paper an extension of service composition problem to the area of social networks and graph based service composition quality criterions is presented. The proposed service composition method is decomposed into three steps consisting of composite structure generation, semantic service discovery method and service plan optimization method. The goal of the service discovery is to find service candidates that fulfill functional requirements and the latter allows for optimal selection of services so that together they satisfy quality criterion, here based on social network graph measures. Presented approach is supported by examples from volleyball sport domain, denoted with domain ontology. Also a web-based tool for service composition is presented. This tool allowed for implementation of the proposed approach but its service orientation allows for easy extension of composition algorithm as well as replacing its parts to fit various domain-specific problems.

1. INTRODUCTION

In the last ten years many researchers have contributed to the field of Service Oriented Architecture and to the Future Internet often also presented as the Internet of services and knowledge accompanying them. Service Oriented Architecture promises a flexible way to manage your IT assets via services – software components accessible by well-defined web protocols. A notion often described in literature is to offer compositions of basic, atomic services when no single service can fulfill user requirements. In that case various approaches to service composition are proposed [1], [2]. First were based on AI Planning, which, by presenting services as state-transforming operators, was trying to search through the space of such operators in order to find a series of operators transforming user-provided data into his goal. This concept is clear

* Wrocław University of Technology, Institute of Computer Science, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland, Pawel.Stelmach@pwr.wroc.pl, Lukasz.Falas@student.pwr.wroc.pl

in SWORD [3] but others like Synthy [4] or METEOR-S [5], despite adding valuable contributions in the form of abstract plan based composition, have also followed this notion in some way. This approach however had some disadvantages, especially an inability to generate more complex, closer to reality service execution plans introducing not only serial but also parallel structures. Also important was the limitation to provide true QoS-based plan generation algorithm. The service execution plans' ability to fulfill the user non-functional requirements was checked only after the composition was performed. Some current solutions [6] still follow the AI Planning approach extending it to solve large-scale problems or incorporate network-based solutions.

Service optimization approaches introduces QoS oriented service execution plan optimization, still focusing on QoS parameters like time of execution, cost of execution, availability and other typical non-functional properties that could be calculated through simple aggregations of service execution plan [7], [8], [9].

Besides the obvious non-functional parameters a good service composition should be done with respect to various Quality of Service (QoS) requirements [10], especially less trivial like security. The security evaluation process should be based on some formal prerequisites. The first problem is that the security measure does not have any specific unit. Also, security level has no objective grounding but it only in some way reflects the degree in which our expectation about security agree with reality. Security issues become crucial if we assume that complex processes are being realized by workflows of atomic services, which may have different security levels. In the case of security the composite service quality estimates should focus more on service connections than services themselves. In this context we can observe more approaches focusing on more complex QoS estimation methods even incorporating agent-base systems and aggregation of opinions on QoS parameters [11], [12].

In this work a social graph based QoS parameter is proposed along with the method for calculating its value. Here, services are perceived by their interactions in composite services and those interactions are modeled as interactions in the social network. A working example is presented in the sports domain, where we present specialized reservation services that allow for transferring of volleyball players. Each service can represent a specific volleyball player so the analogy to the social network is natural, however, a more abstract approach to social interactions between services could be applied to other domains.

The remainder of this paper is organized as follows: in Section 2 a general composition approach is introduced. Next section presents the social graph based criterion in more detail and then in Section 4 the application domain is described. In section 5 implementation aspects are discussed. The last section consists of summary and plans for future works.

2. SERVICE COMPOSITION

The service composition problem can be presented as transformation of user requirements into a composite service execution plan that fulfils them (Fig.1). Typically user will present both his functional and non-functional requirements, sometimes referred to as Service Level Agreement (SLA). To find a service that fulfils those requirements at the same time we propose a three-stage approach. First user functional requirements are given a structure of a graph. This graph of requirements is a good approximation of the general shape of the future composite service. Next a search for service candidates is performed. This goal of this stage is to find services that fulfil each of the requirements but may differ in non-functional parameters. This part is usually performed using semantic filters, which will not be discussed here in detail. Finally, from every service candidate for each functionality only one service is selected according to the aggregated QoS parameter value of the whole composite service. This last step, often called service plan optimization and the role of QoS requirement is described in more detail in the following subsections.

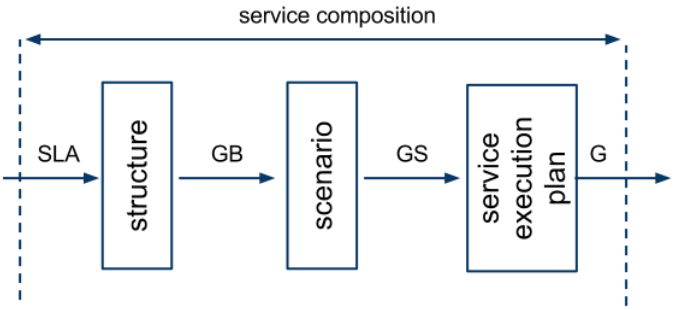

Fig. 1. Stages of service composition approach

3. QOS SOCIAL GRAPH BASED CRITERION

3.1. QOS CRITERION IN COMPOSITION

Most approaches to estimation of QoS parameters for the complete composite service base on recursive aggregation of parameters of serial and parallel structures of that service, replacing those structures with a service described by QoS parameters calculated according to a specific formula for the type of structure and type of QoS parameter (Fig. 2).

For instance, to calculate an execution time parameter of a composite service with a serial structure we add appropriate QoS parameters of each service in the series,

while in AND-type parallel structure we would choose the parameter with the maximum value and in OR-type parallel structure we would choose the minimal value.

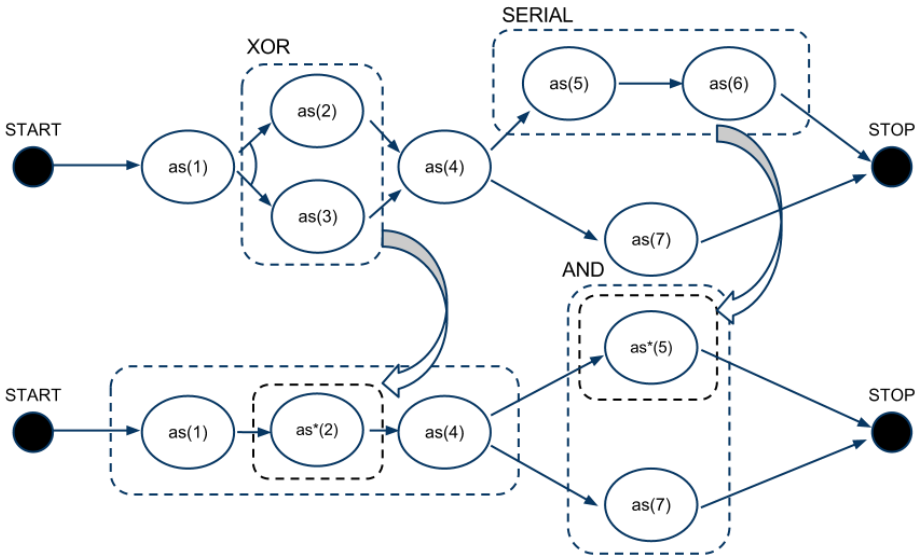

Fig. 2. A single step of composite service QoS parameters aggregation algorithm

Parameters like service cost, service availability and others were also presented with appropriate formulas for serial and parallel structures. However, this is not the only valid approach. As we have presented in [11], where we present a method for determining a security parameter from an agent-based system, more complex ways could be introduces both in a way that many agent-derived opinions on QoS parameters of the service could be taken into consideration and also that the node structure is not the only aspect to take into consideration. In the domain of security the links between services were even more valuable than the services themselves. The same notion is present in another QoS we discuss in this paper: social graph based QoS.

In the following subsection we present a social network based model of services and introduce a method for social QoS parameters calculation.

3.2. SOCIAL NETWORK MODEL

Sociologists are in agreement that a sense of significance or power is the foundation of social networks. There is a problem, however, how to measure this relevance. In literature this notion is frequently discussed with the notion of centrality in the group. This comes from the idea that an actor is not relevant as a unit but its value comes from the combined value of all actors in the network.

While the notion of power might be easily comprehensible in our working example of volleyball players, it is doubtful how well it could be extended to whole the domain of network services. Similarly as in social networks or volleyball teams the social interaction comes from the human and is limited to some organizational structures. Also in services domain it is the people who decide which services are often connected with other services and the input/output limitations are exactly the same as in human created groups – limitations that determine that some volleyball players are playing together more often (there are in one team – or in the field of services – they belong to one organization) and why some players even more often pass the ball to each other (the possible interactions are determined by roles in the team just as service interactions are limited by their input and output parameters). And so the service interactions are to some extent an analogue to human interactions because they come from those type interactions. Of course it would be beneficial to try and determine the overlaying network of service users and the interaction between this and service network, however, in this work we propose a simplified approach, leaving the former for the future works.

The proposed model of the social network describes:

– actors in the social network (here: services)

– relations among actors (these could be any links between services but we assume as a basis that any valid connection between services creates a link between services and any observed connection between services creates another link in a different type of relation; as a result we have at least two types of networks one overlaid on top of another)

– weight for each relation connecting two actors (this parameter could be used in many ways, however, we suggest to use it with the network of valid connections between services so it would simulate the layer of observed interactions, thus integrating both social network layers into one)

We propose two methods from literature that can be applied to social graph based QoS estimation of services:

– a method for calculating service centrality

– a method for calculating service relevance in the group of services

3.3. CENTRALITY

In this paper we will look at centrality through the following interpretations:

– closeness

– betweenness

In understanding centrality as closeness the actor is believed to be most important if he is closer to more actors than any other actor. Then a distance between two actors in the network is defined as a minimal number of relations between them weighted by appropriate weights of those relations:

$$d\ a_i, a_j = \sum_{r_k \in path(a_i,a_j)} \frac{1}{weight(r_k)} \quad (1)$$

where:
d – is a distance measure between two actors
path – is the minimal ordered set of relations that connect two actors
r_k is the *k-th* relation from the *path*
weight(r_k) is the weight of the *k-th* relation from the *path*
If between two actors there is no path that connects them then $d(a_i,a_i)$ is set to 0.

Example:
Based on Fig. 3 one can calculate that a distance between actors 1 and 4 is 7.5.

$$d\ a_1, a_4 = 5 + \frac{5}{2} = 7.5$$

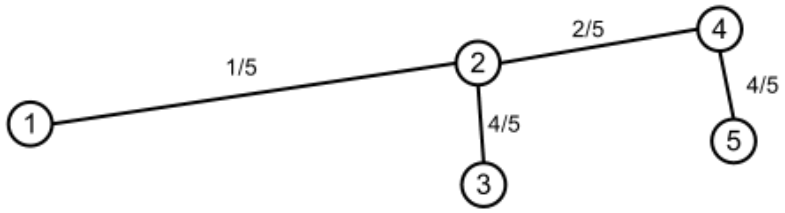

Fig. 3. A simple network of actors with 5 actors and weighted relations among them.

Closeness can be calculated also in another way, ie. by estimating a number of actors a given actor is connected to by 1 relation, by 2 relations, by 3 relations and so on and then generating the closeness parameter by aggregating those "distances" with appropriate weights (in this approach an actor that is connected to more actors directly than indirectly should have a better value of the closeness parameter).

Another interpretation of closeness is betweenness that is based on the notion of being central to the communication among other actors. Betweenness tells us how important is a particular actor for communication of all actors in a group. Betweenness is greater the more paths among other actors he is on (he is one of the actors in any of the relations in the *path* set). In a given network a method for determining betweennes of actor a_k is as follows:

1. *Determine paths path(a_i a_j) among every pair of actors in the network.*
2. *Calculate the number of paths a_k is a part of* (is one of the actors in any relation belonging to *path* set).
3. *Calculate betweennes for the actor a_k according to the formula:*

$$betweenness\ a_k = \frac{No.\ of\ paths\ with\ a_k}{No.\ of\ paths} \quad (2)$$

where:
No. of paths is number of paths determined in step 1
No. of paths with a_k is a number of paths calculated in step 2

4. *Normalize the betweennes parameter for each actor by dividing it by the value of the greatest betweenness among all actors.*

In service composition we use centrality-determining methods in the service selection step to determine this QoS parameter for each of the services.

Also in service plan optimization step we can use betweennes calculating method and calculate the parameter not for the whole social network but for the group of considered candidate services.

3.4. SIGNIFICANCE IN A GROUP

Another key aspect in discussion on social-based QoS parameters for service composition purposed is an ability to determine the significance or relevance of a service in a group of services. One of the parameters measured in social networks is their division into natural groups or cliques. Despite the fact that all users can interact with each other they often choose not to for unknown, intractable reasons. In the service domain this division into groups is exactly the same. One cannot predict service compositions only from the simple network of valid connections between services. Users prefer some services to other; providers promote all their services despite their functionality.

Thus, we would want to analyse if a specific actor in a defined group (not the whole network) acts on behalf of that group or at least is more inclined to act like that than other actor in considered group. We could interpret that criterion as a measure of experience of the group when using the observed connections of services or in more concrete domains when looking at volleyball players that played together and are likely to play well together in the future or in building industry when you look for a series of renovation tasks and you'd prefer all of them to be performed by the same people. In all those cases you'd like to use services that work well together.

What we want to optimize in this approach is the parameter determining some kind of cohesion of the group of actors (services). And a group is more cohesive if it has more or stronger relations in the group than relations outside of the group. An actor that is introduced as into a group of actors brings his relations. If those relations are already with the actors in the group than it is better that if he introduced more "foreign" relations to the actors outside. This is of course not a decisive element and an actor with foreign relations could be valid but in the domain of services it is just a notion that perhaps this service is better suited in another type of a composite service

and another candidate service with more in-group relations is a better match.

An individual significance QoS parameter for a service is defined as a difference between number of relations among services in a composite service and a number of relations outside of the considered composite service (in the network – this parameter has to be pre-calculated before composition).

On this basis the cohesion of the whole group (a QoS parameter for the composite service) is a sum of individual significances for each of the selected service candidates of the considered composite service.

4. APPLICATION IN VOLLEYBALL DOMAIN

The working example of service composition of services being interfaces for volleyball players and the task of service composition being in fact a task of volleyball team composition was influenced by our cooperation with actual sport managers willing to create distributed software for volleyball players management. In fact, the idea for having one registration service for each volleyball player and not one general service for registration came from the basic idea of Internet of Things and the assumption that each of us can have his own, strictly personal web site (or web application) that is not part of a greater monolithic application but at the same time can interact with other web based applications via web services. This approach allows for a centralized management of ones personal information and is in contrast to a person being a part of several or several hundred web portals, applications and social networks, which is highly distributed, uncontrolled and fragmented. A person controls his information and allows other services to access his data in a centralized manner with complete control over this process.

In this context having a repository of services for volleyball players registration, communication etc. is possible and in fact desirable. In this approach the analogy of social QoS parameters for services and social networks of people (volleyball players) are obvious but this example can be extended to other types of services because many of their interactions are in fact generated by people as have been stated already in this paper.

The data from social networks was used in service selection algorithms allowing manager to specify his requirements for particular players and the team. Requirements were both social based and also non-social based on objective qualities like height, agility, jump height or player efficiency derived from appropriate statistics. All QoS parameters and more was described and controlled by the domain ontology (that was also used in service selection stage).

5. IMPLEMENTATION

The composition tool was implemented in the Ruby on Rails web application framework, using mostly Ruby language and jQuery javascript framework for the front-end programming.

The composition method was implemented in a form of two composition services to obtain greater flexibility. Both service selection and service optimization are invoked through SOAP protocol, which is not natively supported by the Ruby on Rails framework, and it was necessary to include the fdv-actionwebservice gem for communication with SOAP-based web services, which provides mechanisms for automated web interface generation.

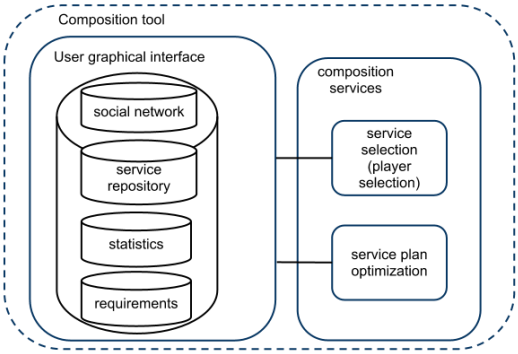

Fig. 4. Composition tool components

6. CONCLUSIONS

In this paper we have proposed and discussed an approach to social graph based QoS parameters estimation method both for describing single services and for the use in the service composition problem. For this purpose briefly service composition, the use of service plan optimization and QoS parameters aggregation in service plan optimization were discussed.

The presented approach was applied to the team sports domain on the example of volleyball team composition. Motivation for the application and appropriate player-service mapping in that domain were described. The validity of the notion to extend the social based QoS parameters to team sports services and services in general was discussed.

In future works more focus will be put on extending the model of social interactions among services to include the hidden user layer and to extract interactions on abstract service types and not only particular instances of services. With these modifi-

cations the suggested model will be used to propose a method for service scenario modification.

ACKNOWLEDGEMENT

The research presented in this paper is co-financed by the European Union as part of the European Social Fund.

REFERENCES

[1] JINGHAI R., XIAOMENG S., *A Survey of Automated Web Service Composition Methods*, Semantic Web Services and Web Process Composition, First International Workshop, SWSWPC 2004, San Diego, CA, USA, 43–54.

[2] MILANOVIC N., MALEK M., *Current Solutions for Web Service Composition*, IEEE Internet Computing 8(6), 2004, 51–59.

[3] PONNEKANTI S. R. AND FOX A.: *SWORD: A developer toolkit for Web service composition*. In Proceedings of the 11th World Wide Web Conference, Honolulu, HI, USA 2002

[4] AGARWAL V., CHAFLE G., DASGUPTA K., KARNIK N., KUMAR A., MITTAL S., SRIVASTAVA B., *Synthy: A system for end to end composition of web services*, Web Semantics: Science, Services and Agents on the World Wide Web In World Wide Web Conference 2005, Semantic Web Track, Vol. 3, No. 4., 2005, 311–339

[5] AGGARWAL R., VERMA K., MILLER J., MILNOR W., *Constraint Driven Web Service Composition in METEOR-S*, Proceedings of the 2004 IEEE International Conference on Services Computing, 2004, 23–30

[6] OH S., LEE D., KUMARA S., *Effective Web Service Composition in Diverse and Large-Scale Service Networks*, IEEE Trans. On Services Computing, Vol. 1, No. 1, 2008.

[7] JONG M. K., CHANG O. K., ICK-HYUN K., *Quality-of-service oriented web service composition algorithm and planning architecture*, The Journal of Systems and Software 81, 2008, 2079–2090

[8] JAEGER M. C., ROJEC-GOLDMANN G., MUHL G., *QoS aggregation in web service compositions*. In EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service 2005, 181–185.

[9] ARDAGNA D.,PERNICI B., *Global and local QoS constraints guarantee in web service selection.* In IEEE International Conference on Web Services (ICWS'05) 2005, 805–806.

[10] ANDERSON S., GRAU A., HUGHES C., *Specification and satisfaction of SLAs in service oriented architectures.* In 5th Annual DIRC Research Conference, 141–150, 2005.

[11] STELMACH P., JUSZCZYSZYN K., KOŁACZEK G., FALAS Ł., *Agent based approach to asynchronous security estimation of composite services in service oriented architecture*, In International Journal on Information Technologies & Security, № 3, 2011

[12] WANG Y., ZHANG J., VASSILEVA J., *Effective Web Service Selection via Communities, Formed by Super-Agents*, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 549–556, 2010.

Service Level Agreement,
resources management

Jan KWIATKOWSKI*, Grzegorz PAPKAŁA*

SLA DRIVEN RESOURCE MANAGEMENT FOR SOA APPLICATIONS

Managing the resources is a complex problem that has number of different solutions. The ideal one shall combine the flexibility with the possibility of reaching the business goals. This leads to the idea of combining the management of resources directly with the Service Level Agreements which are composed of the Service Level Objectives (SLO) objectives which one needs to be fulfill. Those shall be derived from the business objectives and requirements. The aim of the presented work is to propose the resource management linked to the notion of Service Level Agreement. Such an approach can bring the resources utilization closer to the business aims of the company. Unlike resource management oriented toward e.g. minimization of resources usage or processing time our approach can incorporate costs or rewards, thus be directly connected with aforementioned business objectives.

1. INTRODUCTION

Service Oriented Architecture is a concept that emerged in the end of last century and since then the idea is evolving changing the way software is developed, distributed and deployed. It is evident that services became the focal point for almost all software vendors [4], [5] offering solutions generally referred as being in "cloud".

The reason for change had at least three sources. First of all, as reported by Gartner average capacity usage of the servers oscillated around 10–15% in the beginning of 90's last century. Secondly increasing network capabilities and popularity allowed new ways of software distribution, last but not least was emerging technology in virtualization area.

* Institute of Informatics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, , 50-370 Wrocław, Poland.

Low resource utilization has been an issue already in the beginning of computer era. The answer then and now was similar – virtualization. As Gartner claims already almost 30% of all workloads running on x86 architecture are running on virtual machines [6]. Virtualization gave the simple, yet powerful, answer to the problem by introduction of a virtual layer between the physical hardware and software that uses it. This allowed administrators of data centers to break common rule of having one application on one server. Potential configuration issues have no longer been a problem increasing the utilization and leading to overall reduction of hardware requirements.

Service Oriented Architecture (SOA) is trying to resolve another problem appearing in IT world. Software composition and distribution in the traditional form, where applications are not flexible enough to follow rapidly changing needs of business had to be replaced with something more flexible. The idea to compose the business processes from services which are available either publicly or privately, mix and match them as needed, easily grant access to business partners seems like the aim of numerous efforts in software development which can be altogether called SOA. However simple the idea may appear its adoption is connected with lots of obstacles such as standardization, security issues, automation, services recognition, indexing and so on.

Nonetheless, although virtualization is already being used as a common and proven way to decrease the overall hardware needs and costs, still the hardware utilization is around 15-20% and storage utilization does not go above 60% [12]. Virtualization stopped at "low-hanging-fruit" and is not pushing forward. Mission critical services are used as before due to the easier maintenance, controlling and monitoring. What is more, reduced budgets and "do more with less" attitude made it much more complicated for real virtualization adaptation since - especially at the beginning - costs of implementation are higher than those of keeping everything as is.

The Quality of Service (QoS) has been an important issue since the beginning of the service oriented movement. The vision of self aware service, which can be mixed and matched easily to generate value for end customer could not exist without the means to ensure that provided complex service would be of a desired quality. Lack of proper SLA assurance prevents the PlaTel platform from proper handling of non-functional requirements through the whole stack of platform's modules.

The chapter briefly describes existing SLA frameworks that cover the whole stack the service is deployed in – from the infrastructure up to the business customer consuming the service. The structure of the chapter is as follows. The second section is devoted to the description of PlaTel project. Existing standards associated with the SLA their advantages and possible drawbacks are presented in the third section. Section four presents the proposition for implementation of the SLA Driven Resource Management in PlaTel-R that is a part of the PlaTel platform. Key elements of the proposition are: identification of actors and their roles, proposition of attributes to be subject of the agreements. Those are completed by the proposition of resource man-

agement based on established SLA's. Finally section 5 concludes the work and presents the future plans.

2. PLATEL-R – SERVICE EXECUTION ENVIRONMENT

PlaTel is a platform designed to support the composition, execution and monitoring of composite services based on Service Oriented Architecture paradigm. Among other applications building up the system PlaTel-R is devoted to execution of the atomic services on the available resources. To increase their utilization it exploits the capabilities offered by virtualization.

PlaTel-R has a modular structure with the number of modules. The architecture of PlaTel-R is presented in figure 1.

- Service Repository – storage for services descriptions,
- Transform – request analysis and routing,
- Matchmaker – finding match between request and service,
- Monitoring – check the status of servers and virtual machines,
- Manager – combine the work of modules together,
- Virtualizer – interface to connect with virtualization engine.
- Broker – handles user request and distributes them

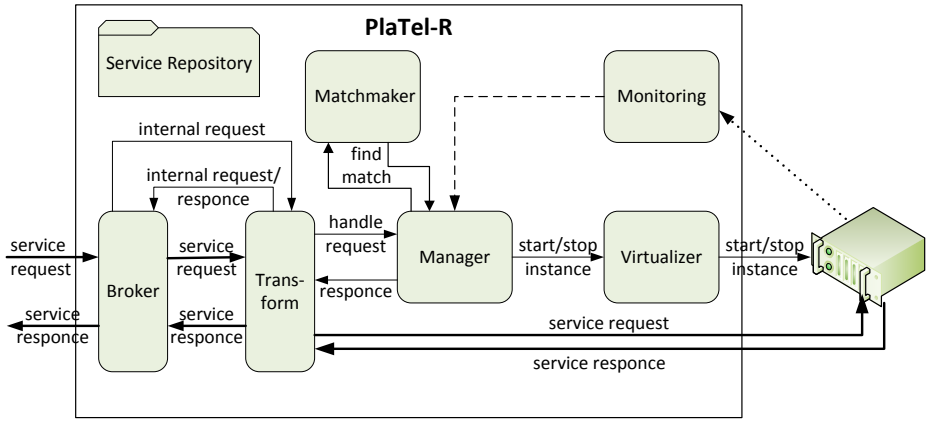

Fig. 1. The architecture of PlaTel-R

As one can easily note the system is currently missing the module to handle resource usage and provision services in a proper way with SLA taken into account. The aim of this article is to propose such a module and to describe the base functionalities it needs to cover.

3. SERVICE LEVEL AGREEMENT

Service Level Agreement is a term already well established in the computer society, yet the standards available for automatic handling of the agreements seem to be relatively simple. It is studied thoroughly in order to find a way to describe the terms in which some service is not only offered, but also the framework to establish such an agreement, negotiate, monitor and change it if needed. The term is strongly connected with the Quality of Service (QoS) that needs to be met by the provider to satisfy the needs of a client. In most cases SLA, although exists is only available on the top level between service provider and service customer [1].

There is a wide range of languages and efforts establishing some standards in that area. Their focus is placed on the non-functional requirements, which are supposed to be met while using the service. Corresponding functional parameters of the service are described using the Web Service Description Language. SLA descriptions are agnostic on the properties described, thus they can be used to describe practically any kind of service in any application domain.

General purpose of SLA means that the required parameters have same meaning by both parties involved in the contract and that the parameter is measurable. First condition implies some common understanding of parameters that may not be that easily shared among various clients and providers. Second, enforces the provider to be able to find a way to measure and monitor the service performance according to the given parameter.

3.1. SLA MANAGEMENT FRAMEWORKS

Building of the complex software nowadays in many cases comprises of reusing number of existing services. They are usually associated with some SLAs. In most cases deriving the low level SLA based on business oriented, non-functional parameters is a real challenge.

A lot of efforts have been already made in order to accomplish the overall full stack solution to assure SLA conformance throughout the whole life of a SLA starting from the SLA template preparation through negotiation, operation and finally archiving. On the other side lies the support for hierarchical enforcement and compliance to the SLA from the very top layer of service provider down to the infrastructures resources demanded to support it.

SLA@SOI is a framework that aims “to deliver and showcase an innovative open SLA Management Framework that provides holistic support for service level objectives – enabling an open, dynamic, SLA-aware market for European service providers” [1]. The framework is built top down, where the top is satisfaction of some hypo-

thetical business. The hierarchy then goes down to the infrastructure that is able to dynamically respond to those business requirements.

The framework goes well beyond what is offered by standards such as WS-Agreement. Although the description of SLA has its roots in that standard, SLA@SOI propose new more general model that abstracts from the web-service and uses more generic service as well as abstracts from the XML as representation thus allowing to describe any service using any language [2],[3]. It emphasizes the importance of model driven development and uses Palladio Component Model to evaluate the performance and reliability of the service before starting to offer the service. This makes initial SLA templates much reliable than those built only based on some assumptions or guesses made by template designer.

3.2. SLA LIFECYCLE

SLA lifecycle is tightly connected with the overall service lifecycle. It is important to note that it does not begin to be an important issue when the service is already finished and is about to be served but rather should be an issue since the beginning of development. As mentioned above usage of software model such as Palladio Component Model allows running the simulations how the system will behave without the need to employ real data and real customers thus enabling higher quality since the very beginning of the service offering.

Table 1 presents the most important actions related to the SLA in overall service lifecycle. The lifecycle although presented in tabular form has a continuous nature and last phase: *decommissioning* shall generate the output that can be an input for next *design and development* phase.

Table 1. Service lifecycle with SLA related actions (based on [1])

	Design & Development	**Offering**	**Negotiation**	**Provision-ing**	**Opera-tions**	**Decom-missioning**
Customer			Requirements			
Service Provider		SLA Template	SLA		Adjust, monitor	Archive SLA/SLA(T)
Software Provider	Service model					
Infrastructure Provider			Resources reservation	Sub SLA	Adjust, ensure QoS	

Service lifecycle starts with *design and development* phase thus shall not only concentrate on generating high quality product, but also preparing the model that can serve to built SLA templates with realistic parameters. Furthermore in negotiation phase such model can be used for simulation and giving answer to question whether the service is able to fulfil certain requirements or not. SLA templates serve for customer to easily specify requirements he/she needs from the service. They basically contain the set of parameters the provider is willing to fulfil. This can be e.g. number of users who are allowed to use certain service, response time guaranteed etc.

Based on the template customer can request certain values (if allowed in the template) and those need to be translated to requirements embraced lower layer, in this case PlaTel-R. During the *negotiation* phase customer's requirements need to be accepted on each of the levels of the platform as well as translated into some resource amount that can be reserved and used while operating.

After successful *negotiation* the infrastructure provider should have work on sub SLA that is connected with the parameters it can monitor and understand, thus *provision* the service accordingly. *Provisioning* itself may or may not require starting new instances of the service. It can be sufficient to work with the instances already available without degrading the QoS offered to the customers.

After *provisioning* the service is operable and accessible for the customer. According to the accepted SLA the service shall be monitored and the QoS ensured. The precision in which granted resources correspond to SLA objectives can be measured as well and such information should be kept while decommissioning ensuring the knowledge about service behaviour is not lost. Later it can be helpful while generating new templates for improved services.

3.3. SLA STANDARDS

The most used standards for describing the SLA are IBM's Web Service Level Agreement (WSLA) and WS-Agreement from Open Grid Forum. The aforementioned SLA* seems to be more flexible and general, yet it is not as simple as the two XML based ones. WSLA has the advantage of describing the Service Level Objectives (SLO) on the operation level, thus can be more fine grained than the WSLA. On the other hand WS-Agreement contains simple negotiation schema and is a subject of numerous extensions proposed by various authors [7], [10], [11]. Those include negotiation protocol extensions [7], [10] and semantics to the standard [7], [11], mainly.

WSLA did not gain such a high acceptance as WS-Agreement [7], yet it seems to be more directed toward web services unlike WS-Agreement, which has numerous implementations mainly, connected with the grid computing. WSLA standard has not been a subject of some updates since year 2003 which may indicate it is not going to be widely accepted or used.

Both standards are connected with similar lifecycle and contain same sections. Those are: description of the parties involved in the contract, description of the service being the subject of the SLA and their mutual obligations (from WSLA) or guarantee terms (from WS-Agreement).

4. THE POSSIBLE PLATEL-R EXTENSIONS

As there is no perfect and the only correct solution to integrate SLA into resources management, few requirements have been stated in order to narrow down and describe the one that suits best the condition PlaTel-R is operating in. Operating of the PlaTel-R one can assume existence of two basic scenarios:

- long running tasks with SLA generated on demand while the request is supposed to be sent and only for this particular request,
- generation of SLA for the customer who is expected to reuse the service in the constant manner, thus generating a stream of similar requests.

Each of the scenarios described above requires different handling in real live usage. As for the first one there is in most cases no way to estimate the resources or time needed to complete the request. This indicates that the SLA shall be designed to guarantee the resources for the customer. Such approach is similar to the job realization scenarios in grid environments where the key issue is proper scheduling of the jobs. In such situation there is no place for renegotiation of the SLA and possibility to reuse the agreement is limited.

The second scenario yields more possibilities concerning (re)negotiation of terms or reuse of the agreement. In this case it is common for example to deal with the agreement that states certain response time for request realization with the upper limit for number of requests incoming in the specified time period. Such agreement requires the knowledge about resource usage per request and its influence on the realization time. Based on the description of SLA lifecycle, standards and reference implementations of the SLA management frameworks, following extensions need to be added to PlaTel-R to increase its performance (better resource utilization):

- SLA contracts manager – responsible for SLA negotiation, advertisement of available templates, reacting on detected SLA violations,
- SLA monitoring agent – responsible for monitoring of the metrics pointed in a given SLA,
- SLA driven resource manager – responsible for proper resolution of the required parameters from agreement to the resources assigned to the service (provisioning).

4.1. SLA CONTRACTS MANAGER

In [9] authors summarize service attributes from various studies, which can be present in the agreement. They list following attributes: run time, reputation, uptime, response time, negotiation, cost, reliability, problem resolution and maintenance. From those the manager shall support cost, reliability, response time and uptime.

Apart from proper handling of the attributes which shall be combined with appropriate agreed units, time periods, levels etc. depending on the attribute contract manager shall implement an API allowing third parties to negotiate the agreement. The agreement itself when considering only the subset of attributes mentioned above can be understood as a tuple (based on [9]) $A_S = (C_S,R_S,Rt_S,Ut_S)$, where A is an agreement of service S, C its is cost, R – reliability, Rt – response time, and Ut – uptime.

Such a definition is oversimplified and does not convey the information about agreed thresholds for the attributes, their relation (whether desired value shall be greater or less than the given one) or priority. Nonetheless this approach is suitable for initial implementation which can assume to accept only exact matches of requested attributes and provided ones, thus rejecting all that are not equal.

4.2. SLA MONITORING AGENT

The very important feature of the attributes available in SLA is their measurability. Monitoring agent shall accompany any currently active agreement according to the specification from the SLA. The frequency of measures or agreed values of attributes are present in the contract thus the agent shall simply check if the operations are done accordingly. Any violation of the SLA shall be reported back to the SLA contract manager that can undertake appropriate actions (notifying customer or administrator via email, requesting more resources or renegotiation of the contract if possible).

4.3. SLA AWARE RESOURCE MANAGER

Resource manager is the core module that makes the overall idea work. Its key responsibility is to manage resources seeking two aims, which can – and in most cases would be – mutually exclusive. First of all the manager shall provision the instances of services with proper resources to ensure the fulfilment of conditions agreed in the SLA. Secondly it shall increase the utilization of the available resources, so that overall capacities are used to the highest possible degree. Taking into account traffic fluctuations, peeks in the incoming requests and so on it is generally considered 40% of average resource usage as a good result and 50% being the “Holy Grail” [13].

Managing the resources for SLA can be described in three distinct steps: provisioning of resources, adjusting and freeing the resources. As it was already mentioned

provisioning of the resources is connected with translating the SLO described using the attributes such as response time to resources such as CPU or memory. This can however be done the other way around and launched service instance can be described using the SLA terms to make matching easier.

Such an instance can then be described as the tuple just like the agreement itself. However to keep the QoS the description has to be extended at least by the number describing the amount of requests which can be processed at once without quality degradation, thus leading to following: $I_S = (C_S,R_I,Rt_{SI},Ut_I, Q_I)$, where I_S is an instance of service S, C_S is a cost of request to service S, R_I is a reliability of instance I, Rt_{SI} is a response time of instance of service S, Ut_I is a uptime of instance I and Q_I is concurrent request quota.

This approach hides from the description resources really assigned to the instance, yet they can as well be included in the tuple. The proper amount of resources to be assigned can be obtained in few ways. The software model can be used to simulate the usage of the service generating realistic results which can be used to properly adjust resources levels to offered SLA template. Alternatively software provider or expert can set the relation between e.g. response time and assigned CPU frequency explicitly. Furthermore each of those approaches should be followed by the adjustments made on the base of real execution data.

The list of parameters used allowed for SLAs is shorten for the sake of simplicity, but in real environment there are no limits in using the attributes which are not mentioned here. For example one can think of a requirements concerning security or localization. They can be relatively easy matched if servers on which the services are deployed have proper description. Nonetheless such parameters are hard to be measured especially in an ongoing fashion, thus their usage has to be well grounded.

5. CONCLUSIONS

Service Level Agreement is one of the pieces that makes SOA work the way it is expected to. It assures meeting the quality that is required by the customer. Adaptation of SLA in PlaTel would make its performance more robust and ensures that composed services would execute in the expected way. From the point of view of the execution environment (PlaTel-R) SLA used even internally makes resources management more reasonable. Offered and agreed upon parameters can serve as a base for provisioning of service instances.

The chapter presents some of the most important issues related to the SLA, available standards to describe them, existing frameworks and current works in this field. Proposed extension to PlaTel-R is based on this review and aims in providing some

viewpoint on order to find the best solution applicable in this project. Ultimately leading to the self-aware system able to react on changing conditions.

Further studies will be devoted to the proposed extensions. They will focus on preparing the negotiation scheme, starting from the simple exact matching and further improvements toward dynamic negotiation with parameter adjusting according to the system capabilities in certain time. Subject of SLA renegotiation would be also an important issue to be resolved. Most of the efforts will be directed to the resource management that will need to be extended by finding the algorithms to learn real mapping between resource usage and SLA attributes, making decision based on them and last but no least distribute the instances among available servers.

ACKNOWLEDGEMENTS

The research presented in this paper has been partially supported by the European Union within the European Regional Development Fund program no. POIG.01.03.01-00-008/08.

REFERENCES

[1] SLA@SOI, Reference *Architecture for an SLA Management Framework*, 2011, available at http://sla-at-soi.eu/.
[2] KEARNEY K.T., TORELLI F., *SLA Model*, 2011, available at http://sla-at-soi.eu/.
[3] KEARNEY, K.T.; TORELLI, F.; KOTSOKALIS, C.; *SLA: An abstract syntax for Service Level Agreements*, 11th IEEE/ACM International Conference on Grid Computing, 2010, pp.217–224.
[4] THOMAS E., *Service-Oriented Architecture: Concepts*, Technology and Design, Prentice Hall 2005.
[5] DEREK T., HAMILTON A. JR., MACDONALD R., SANDERS J., *Supporting a service oriented architecture*, Proceedings of the 2008 Spring simulation Multiconference, SpringSim '08, San Diego, USA, 2008, pp. 325–334..
[6] PERILLI A, *Hyper-v is underperforming says gartner*, available at, http://itknowledgehub.com/networking-infrastructure/hyper-v-is-underperforming-says-gartner
[7] OLIVER W., SEIDEL J., WIEDER P., ZIEGLER W., 2008, *Using SLA for resource management and scheduling - a survey*, Grid Middleware and Services, Springer US, 2008, pp. 335–347.
[8] AIELLO M., FRANKOVA G., MALFATTI D., *Semantics and Extensions of WS-Agreemen,* Journal of Software, 2006.
[9] BLAKE M., CUMMINGS D., *Workflow Composition of Service Level Agreements for web services*, IEEE International Conference on Services Computing (SCC 2007).
[10] DI MODICA G., TOMARCHIO O., VITA L., *Dynamic SLA's management in service oriented environments,* Journal of Systems and Software, 2009, Elsevier Inc.
[11] BADIA R., EJARQUE J., GOIRI I., GUITART J., JULIA F., DE PALOL M., TORRES J., *SLA-Driven Semantically-Enhanced Dynamic Resource Allocator for Virtualized Service Providers*, IEEE Fourth International Conference on eScience, 2008, pp.8–15.
[12] SARGEANT P., *Data center transformation: How mature is your it?*, available at http://www.gartner.com/it/content/1282000/1282013/data_centre_transformation_phil_sargeant_17feb2010.pdf
[13] VOGELS W., *Beyond Server Consolidation,* Queue, 2008.

PART 2

CONTENT AWARE NETWORKS AND NETWORK SERVICES

IMS, NGN, call processing performance,
Grade of Service, GoS, traffic model

Sylwester KACZMAREK*, Maciej SAC*

TRAFFIC MODEL FOR EVALUATION OF CALL PROCESSING PERFORMANCE PARAMETERS IN IMS-BASED NGN

In the modern world requirements for accurate and fast information distribution are becoming more and more important, which creates a strong necessity for appropriate telecommunication network architecture. Proposition of such an architecture is the Next Generation Network (NGN) concept, which in order to guarantee Quality of Service (QoS), should be correctly designed and dimensioned. For this reason proper traffic models must be proposed, which should be efficient and also simple enough for practical applications. In the paper such a traffic model of a single domain of NGN architecture based on the IP Multimedia Subsystem (IMS) concept is proposed, which allows to evaluate mean Call Set-up Delay (CSD) and mean Call Disengagement Delay (CDD), a subset of call processing performance parameters defined by International Telecommunication Union Telecommunication Standardization Sector (ITU-T). Using the model basic relationships between network parameters and call processing performance are investigated and presented. All obtained results are verified using simulations, which confirm correctness and usefulness of the proposed model.

1. INTRODUCTION

For the last years telecommunication, Internet and media organizations have focused on standardizing and implementing one common architecture delivering multimedia services called the Next Generation Network (NGN). According to the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) definition, NGN is a packet network with independent service and transport stratums providing various services with Quality of Service (QoS) guarantees [1]. Currently it is assumed that NGN service stratum is based on the 3rd Generation Partnership Project (3GPP) IP Multimedia Subsystem (IMS) [2] developed as a platform to provide

* Department of Teleinformation Networks, Faculty of Electronics, Telecommunications and Informatics, Gdańsk University of Technology, 11/12 Gabriela Narutowicza Street, 80-233 Gdańsk, Poland.

multimedia services to 3G mobile users. IMS elements handle user requests and utilize mainly SIP [3] and Diameter [4] communication protocols defined by Internet Engineering Task Force (IETF).

Satisfying quality guarantees requires appropriate design of Next Generation Network. Methods and mechanisms of traffic engineering must be applied to dimension technology-dependent transport stratum as well as to calculate required processing power and parameters of service stratum servers. This involves proposition and application of proper traffic models, which should be possibly simple and well describe the operation of NGN elements. From many parameters related to quality very important to evaluate using traffic models are call processing performance metrics [5,6] such as Call Set-up Delay (*CSD*) and Call Disengagement Delay (*CDD*). These metrics were formerly known as Grade of Service (GoS) parameters.

Performed review of the current standardization and research concerning IMS-based NGN (in the next part of the paper also abbreviated as IMS/NGN) traffic engineering [7] indicated that standardization organizations do not consider this area in their work and, apart from that, available traffic models do not fully include characteristics of IMS/NGN networks. There exist models describing different network architectures applicable to NGN transport stratum [8–10], however, they do not explicitly take into account resource and admission control elements [11] proposed by standardization bodies. Similar situation takes place in the case of NGN service stratum, for which are no specific models, and only SIP and IMS models [12-15] can be found with no standardized resource control mechanisms.

As a result of the review, we decided to propose our own traffic model, which allows to evaluate mean Call Set-up Delay and mean Call Disengagement Delay in a single domain of IMS-based NGN architecture with fine precision and is not excessively complicated. The model is presented in this paper, which is organized as follows. Architecture of IMS-based ITU-T NGN solution is introduced in section 2. Proposed analytical and simulation models are described in section 3. Section 4 is devoted to the results of performed IMS/NGN call processing performance investigations. Summary and future work concerning the proposed traffic model are presented in section 5.

2. IMS-BASED NGN

The IP Multimedia Subsystem concept [2] was introduced by 3GPP in 2002 as a part of proposed third generation mobile network architecture. The main elements of IMS are Call Session Control Functions (CSCFs), which are generally SIP servers: P-CSCF (Proxy-CSCF, first contact point for terminal), S-CSCF (Serving-CSCF, main server handling all sessions) and I-CSCF (Interrogating-CSCF, server handling mes-

sages from other domains). Key role is also performed by Home Subscriber Server (HSS) storing all data regarding user profiles, user location, authentication as well as authorization. For communication between functional elements of the IP Multimedia Subsystem architecture common and well defined protocols are used. SIP protocol is utilized for controlling multimedia sessions between users. For authentication, authorization, accounting as well as retrieving user profiles Diameter protocol is used. Elements concerning media are controlled using H.248 protocol. Detailed IMS architecture is described in [16].

The IMS concept is independent of the used transport network technology and can be included in other network architectures as a part of service stratum. Using such an approach ITU-T as well as ETSI TISPAN (European Telecommunications Standards Institute Telecommunications and Internet converged Services and Protocols for Advanced Networking) defined their NGN solutions [17,18], which are very similar. Next Generation Network architecture developed by ITU-T [17] is, however, more advanced and complex, especially in aspects of user mobility [19] and adopting particular transport technologies to transport stratum [20–22]. Therefore, this solution will be considered in the next part of the paper.

ITU-T NGN functional architecture (Fig. 1) was defined in 2006 and consists of transport stratum, service stratum and applications. The NGN network cooperates with various Customer Premises Equipments – CPEs (NGN terminals, legacy PSTN/ISDN terminals, customer networks) as well as network types (NGN networks, IP non-NGN networks, PSTN/ISDN networks) and is fully manageable through Management Functions.

NGN service stratum cooperating with applications delivering services to users includes Application Support Functions and Service Support Functions (ASF&SSF) as well as Service Control and Content Delivery Functions (SC&CDF). ASF&SSF elements include functions such as the gateway, registration, authentication and authorization at the application level and combined with SC&CDF units provide CPEs and applications with requested services. SC&CDF elements include Service User Profile Functions, SUPF, which are the equivalent of the HSS from the IMS concept, and service components. The most important service component is IP Multimedia Service Component containing IMS elements [23] and providing NGN terminals with multimedia as well as traditional PSTN/ISDN services.

ITU-T Next Generation Network transport stratum provides IP connectivity services to NGN users and is under the control of Transport Control Functions, including Network Attachment Control Functions (NACF), Resource and Admission Control Functions (RACF) and Mobility Management and Control Functions (MMCF). No assumptions are made about the technologies forming Transport Functions, which provide connectivity for all NGN elements. The NACF element offers mechanisms essential during connecting CPE to an access network such as dynamic provisioning of IP addresses and other parameters, authentication, authorization and location man-

agement. The information necessary for proper NACF operation is stored in Transport User Profile Functions (TUPF). The MMCF unit provides functions for the support of IP-based mobility in the transport stratum, which is considered as one of the NGN services. This element is not dependent on the used access network technology and supports handover across different technologies.

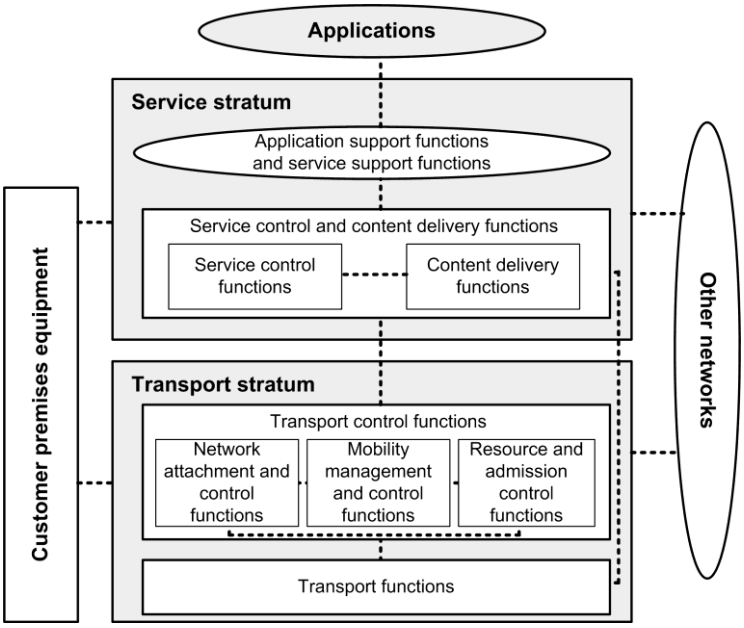

Fig. 1. ITU-T NGN Release 2 architecture [17]

The RACF element [11] is responsible for admission control procedures as well as resource allocation in the transport stratum. It acts as the arbitrator between Service Control Functions (SCF) and Transport Functions for QoS (Fig. 1). The final decision regarding the requested resources is based on transport subscription information, Service Level Agreements (SLAs), network policy rules, service priority, and transport resource status and utilization information. RACF provides an abstract view of transport network infrastructure to SCF and makes service stratum functions agnostic to the details of transport facilities. The architecture of RACF with division into two decision elements – transport technology independent PD-FE (Policy Decision Functional Entity) and transport technology dependent TRC-FE (Transport Resource Control Functional Entity) – allows to efficiently perform its tasks.

In the ITU-T NGN architecture there are two supported resource control modes: push mode and pull mode (Fig. 2) [11]. Push mode, a target mode for NGN, is utilized for CPEs (Customer Premises Equipments) which have QoS negotiation capability at service stratum (using e.g. SIP and SDP protocols and their extensions) or do not have

such a capability. The service request containing or not information about the demanded resource amount is sent to SCF (1) where this information is extracted or determined and transmitted to RACF (2) where the decision about the resource demand is made and demanded resources are allocated (3).

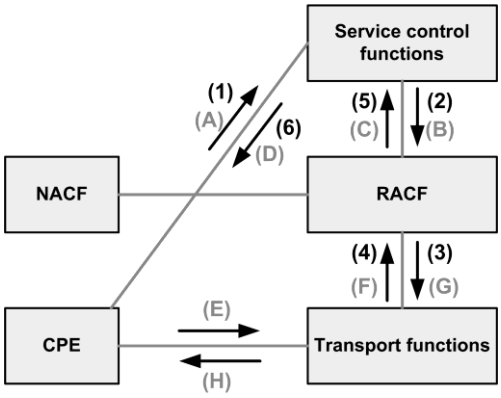

Fig. 2. RACF resource control modes: push mode (black numbers) and pull mode (gray letters) [11]

Pull mode, supported for compatibility with current transport technologies, is used for CPEs having QoS negotiation capability at transport stratum (using e.g. RSVP protocol). The resource request for the demanded service may optionally be preceded by sending a message containing or not service level description of QoS requirements to SCF (A) where the information about these requirements is extracted or determined and send to RACF for authorization (B). As a result, an authorization token, which can be used to bind service request at service and transport stratums, is returned to SCF (C) and CPE (D). After that, CPE generates the resource request to the transport stratum elements (E), which is further passed to RACF (F) and after policy decision resources are allocated (G).

3. TRAFFIC MODEL OF IMS-BASED NGN

The first preliminary concept of an analytical traffic model of a single domain of IMS/NGN was presented in [24] without carrying call processing performance investigations. The model described in this section is a thoroughly extended version of that work and includes among others detailed calculations of communication times. Apart from the analytical model, a proper simulation environment was also implemented, which allowed to verify the proposed extended analytical approach.

Network model proposed for a single domain of ITU-T NGN architecture [17,23] is presented in Fig. 3. Assumed call set-up and disengagement scenarios are depicted

in Fig. 4 and Fig. 5 [11,23–26]. In the domain users sending call set-up and disengagement requests, which do not involve utilization of application servers, are registered. User requests are sent by their terminals UE1 and UE2 to Proxy-CSCF (P-CSCF) server and consecutively to Serving-CSCF (S-CSCF) server, which stores local copies of handled user profiles. From S-CSCF requests are forwarded again to P-CSCF, which makes decision about handling or not user call set-up request based on the information exchange with transport stratum represented by the RACF element. In the case of call disengagement request P-CSCF sends the request to RACF, which releases transport resources associated with the call. After receiving a response from RACF, P-CSCF informs the terminal initiating the disengagement procedure about its result.

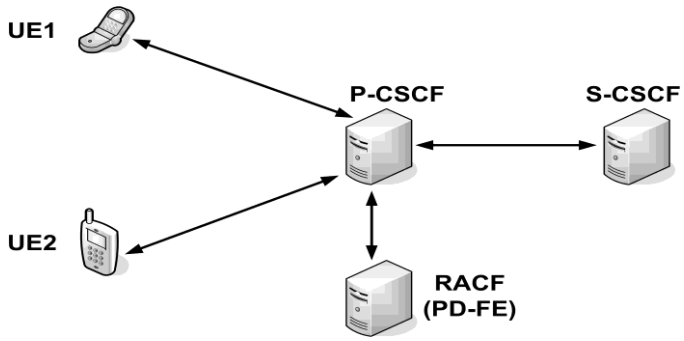

Fig. 3. Model of a single domain of IMS/NGN [17,23]

It is assumed in the model that message loss probability is negligible and, thus, there are no message retransmissions and each request is properly handled. We also assume that sets of audio/video codecs supported by UE1 and UE2 are compatible and no audio announcements are played during the call, which excludes involvement of the MRFC (Media Resource Function Controller) element [16].

Communication of P-CSCF with RACF control element is, however, included and performed using Rs interface and Diameter protocol with no assumptions concerning the transport network technology, which allows to model the behavior of any technology. It is assumed that RACF controls resources using push mode (Fig. 2) and performs a two-stage resource reservation procedure using AAR and AAA messages (Fig. 4) [25,26]: initial reservation (RR-Mode = 1) and final commitment (RR-Mode = 3). Release of resources associated with the disengaged call (Fig. 5) is executed in one stage using Diameter STR and STA messages.

Based on the available ITU-T standards [5,6] and paper [13] concerning call processing performance in Voice over Internet Protocol (VoIP) networks as well as the described network model (Fig. 3) and communication scenarios (Fig. 4 and Fig. 5) we can determine parameters which should be evaluated by the traffic model proposed in

the next part of this section. These parameters are mean values of Call Set-up Delay (*CSD*) and Call Disengagement Delay (*CDD*), which can be defined as follows using the $t_1 \div t_{10}$ times illustrated in Fig. 4 and Fig. 5.

$$CSD = (t_2 - t_1) + (t_4 - t_3) + (t_6 - t_5) \tag{1}$$

$$CDD = (t_8 - t_7) + (t_{10} - t_9) \tag{2}$$

Particular parts of formulas for *CSD* (1) and *CDD* (2) concern consecutive stages of call set-up and disengagement. It is important that equations (1) and (2) do not take into account behavior of a destination user (the time of answering an incoming call) and message processing times in terminals.

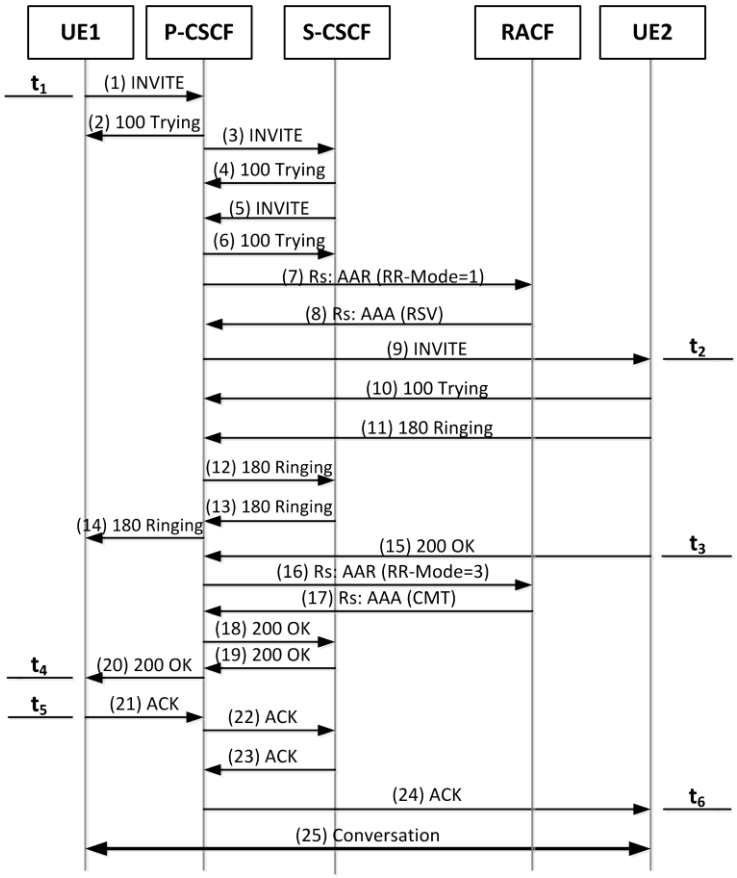

Fig. 4. Call set-up scenario and definition of $t_1 \div t_6$ times necessary to calculate *CSD*

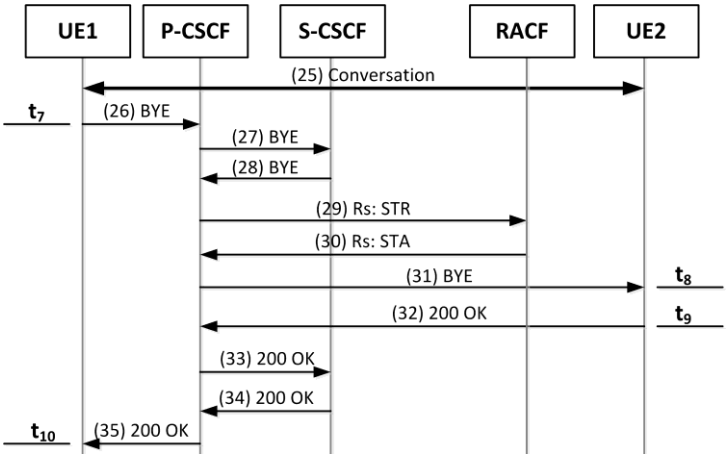

Fig. 5. Call disengagement scenario and definition of $t_7 \div t_{10}$ times necessary to calculate *CDD*

Structure of the traffic model proposed for the network architecture described in Fig. 3 and the communication scenarios presented in Fig. 4 and Fig. 5 is illustrated in Fig. 6. P-CSCF, S-CSCF and RACF elements correspond to the elements of the network model in Fig. 3. UE1 and UE2 units represent many user terminals performing call set-up (Fig. 4) and disengagement (Fig. 5) scenarios. Intervals between aggregated call set-up requests (SIP INVITE messages) arrivals are given by an exponential distribution with defined intensity λ_{INV}. Although message arrivals to CSCF servers in the model (Fig. 6) do not generally follow a Poisson process as in the case of INVITE requests, we assume that message inter-arrival times can be approximately described using an exponential distribution and operation of CSCF processors can be modeled using M/G/1 queuing systems [27]. In the next section we will demonstrate that such an approximate analysis gives in most cases satisfactory results.

Delay concerning transport stratum resource reservation and release introduced by RACF is described in the model as a random variable with any probability density. $K_{a\text{-}b}$ blocks (a, b = 1, 2, X, U) represent communication times between particular elements of the network, where the first letter (a) corresponds to the source element and the second (b) – to the destination element. For example, $K_{1\text{-X}}$ block represents the communication time between the processor 1 (P-CSCF) and X element (RACF). Communication times include propagation times, message transmission times as well as buffering messages in queues before transmission if communication links are busy.

The aim of the proposed model is to evaluate mean Call Set-up Delay and mean Call Disengagement Delay in a single domain of IMS-based NGN. For this reason the following input variables are defined:

- call set-up request (SIP INVITE message) intensity, λ_{INV},
- time of processing SIP INVITE message by P-CSCF, T_{INV1},
- time of processing SIP INVITE message by S-CSCF, T_{INV2},

– a_k factors ($k = 1, 2, \ldots, 8$) determining times of processing other SIP and Diameter messages by CSCF servers

$$\begin{gathered} T_{TRi} = a_1 \cdot T_{INVi},\ T_{RINGi} = a_2 \cdot T_{INVi},\ T_{OKi} = a_3 \cdot T_{INVi}, \\ T_{ACKi} = a_4 \cdot T_{INVi},\ T_{BYEi} = a_5 \cdot T_{INVi},\ T_{OKBYEi} = a_6 \cdot T_{INVi}, \\ T_{AAAi} = a_7 \cdot T_{INVi},\ T_{STAi} = a_8 \cdot T_{INVi},\quad i = 1, 2 \end{gathered} \tag{3}$$

– time of processing messages by RACF described by a random variable T_X,
– lengths of optical links, bandwidth available on optical links, lengths of messages transmitted over optical links – values necessary to calculate communication times $K_{a\text{-}b}$.

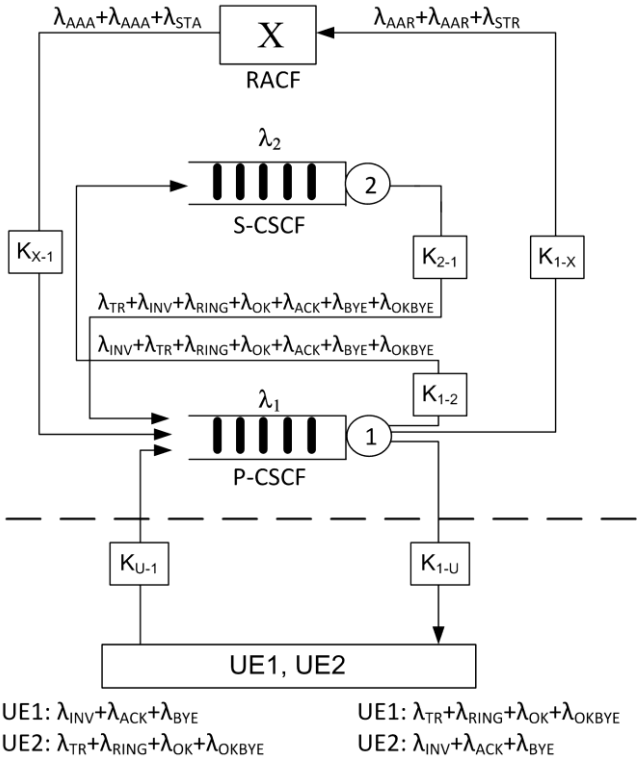

Fig. 6. Structure of the proposed traffic model with intensities of messages from scenarios presented in Fig. 4 and Fig. 5

In the proposed traffic model *CSD* and *CDD* values consist of several components including message processing times by P-CSCF, S-CSCF and RACF, message waiting times in P-CSCF and S-CSCF queues as well as communication times. In order to calculate mean Call Set-up Delay and mean Call Disengagement Delay

$$E(CSD) = E(t_2 - t_1) + E(t_4 - t_3) + E(t_6 - t_5) \tag{4}$$

$$E(CDD) = E(t_8 - t_7) + E(t_{10} - t_9) \tag{5}$$

mean values of these components have to be computed.

In our case processing times for particular messages in P-CSCF and S-CSCF are deterministic and can be easily obtained using (3). For RACF full message processing time description is given by the T_X input variable. Mean message waiting times in the queues of CSCF servers can be estimated using formulas for M/G/1 queuing system [27]. For this reason the following parameters are necessary:

- Intensities of messages sent to processor 1 (P-CSCF), λ_1, and 2 (S-CSCF), λ_2, which can be determined using Fig. 4, Fig. 5 and Fig. 6

$$\begin{aligned}\lambda_1 = (\lambda_{INV}+\lambda_{ACK}+\lambda_{BYE}) + (\lambda_{TR}+\lambda_{RING}+\lambda_{OK}+\lambda_{OKBYE}) + \\ + (\lambda_{TR}+\lambda_{INV}+\lambda_{RING}+\lambda_{OK}+\lambda_{ACK}+\lambda_{BYE}+\lambda_{OKBYE}) + + (\lambda_{AAA}+\lambda_{AAA}+\lambda_{STA})\end{aligned} \tag{6}$$

$$\lambda_2 = \lambda_{INV}+\lambda_{TR}+\lambda_{RING}+\lambda_{OK}+\lambda_{ACK}+\lambda_{BYE}+\lambda_{OKBYE} \tag{7}$$

  Since all elementary intensities in (6) and (7) are equal to λ_{INV}, these formulas can be simplified to

$$\lambda_1 = 17\,\lambda_{INV} \tag{8}$$

$$\lambda_2 = 7\,\lambda_{INV} \tag{9}$$

- Mean message processing time and variance of message processing time for P-CSCF and S-CSCF. These values can be easily calculated after determining the set of messages processed by particular CSCF servers, which can be done through analysis of Fig. 4, Fig. 5 and Fig. 6.

The last unknown parts of $E(CSD)$ (4) and $E(CDD)$ (5) are mean communication times, which consist of propagation times, message transmission times as well as message buffering delays before sending them through busy links. Propagation time is a constant value dependent only on the distance between network elements and assuming optical links is equal to 5μs/km. Transmission time includes a fixed time necessary to send a message, which can be calculated by the message length division by the link bandwidth. Mean message buffering delay before sending through the link can be approximately evaluated using M/G/1 model [27], analogically to the calculations of message waiting time in the queues of CSCF servers. All parameters necessary for such computations can be derived from Fig. 4, Fig. 5 and Fig. 6.

In order to verify the correctness of the analytical model described in the previous part of the section, an appropriate simulation model was implemented as well [28].

For a simulation framework OMNeT++ 4.2 [29] was chosen due to its high scalability, performance and open source character [30,31]. Elements of the network architecture illustrated in Fig. 3 performing the scenarios described in Fig. 4 and Fig. 5 were implemented as modules in C++ programming language. The operation of IMS/NGN architecture as well as call set-up and disengagement scenarios were precisely implemented in the simulation model without previously described approximations used in the analytical model. The following parameters were determined to make the simulation as realistic as possible and obtained results reliable (some assumptions are based on test simulation series):

- total simulation time: 2500 s,
- warm-up period: 1500 s,
- 5 measurement periods,
- 0.95 confidence level,
- time intervals between aggregated call set-up requests (SIP INVITE messages) are described by an exponential distribution with given request intensity λ_{INV},
- call duration time is determined by an exponential distribution with mean of 180 s,
- UE1 and UE2 elements represent many user terminals processing messages in nonzero time; SIP INVITE message processing time in UE1 and UE2 is given by a uniform distribution with values from 1 ms to 5 ms; times of processing other messages are related to this time as in (3)
- answer time (the time between sending 180 Ringing and 200 OK message by the destination user terminal) is described by a uniform distribution with values from 2 s to 8 s,
- SIP 100 Trying message is sent after 10% of SIP INVITE message processing time.

4. RESULTS

In the section we present results of call processing performance investigations in a single domain of IMS/NGN architecture obtained using the analytical and simulation model described in section 3. Results demonstrated in the next part of the paper were achieved using the data sets presented in Tab. 1. Additionally, the following values of the a_k factors (3) were assumed:

$$a_1=0.2,\ a_2=0.2,\ a_3=0.6,\ a_4=0.3,\ a_5=0.6,\ a_6=0.3,\ a_7=0.6,\ a_8=0.6 \tag{10}$$

For calculation of communication times between network elements message lengths presented in Tab. 2 were utilized. SIP links were modeled as M/G/1 queuing

systems. Due to the fact that lengths of particular Diameter messages are not known and only mean length is assumed, mean message buffering delay before sending through the Diameter link was approximately estimated using M/M/1 queuing model.

Table 1. Input data sets

Data set	λ_{INV} [1/s]	T_{INV1} [ms]	T_{INV2} [ms]	T_X [ms]	Link parameters
1a	5–250	0.5	0.5	3	0 km
1b	5–250	0.5	0.5	40	0 km
2a	40	0.1–3	0.1–3	3	0 km
2b	40	0.1–3	0.1–3	40	0 km
3a	100	0.5	0.5	3	0–1000 km, 10 Mb/s
3b	100	0.5	0.5	3	0–1000 km, 100 Mb/s
4a	220	0.5	0.5	3	0–1000 km, 10 Mb/s
4b	220	0.5	0.5	3	0–1000 km, 100 Mb/s

Table 2. Message lengths [15]

Message	Length in bytes
SIP INVITE	930
SIP 100 TRYING	450
SIP 180 RINGING	450
SIP 200 OK (answer to INVITE)	990
SIP ACK	630
SIP BYE	510
SIP 200 OK (answer to BYE)	500
Diameter message	750 (mean length)

Results presented in Fig. 7 demonstrate that $E(CSD)$ and $E(CDD)$ depend on call set-up request (SIP INVITE message) intensity, however this dependence is not so strong as the influence of RACF message processing time (T_X), which is related to transport network complexity and technology. Due to such an influence as well as the fact that during call set-up scenario (Fig. 4) P-CSCF communicates with RACF two times and during call disengagement (Fig. 5) – only once, mean *CSD* is approximately two times larger than mean *CDD*.

In Fig. 8 influence of CSCF servers processing power (time of processing SIP INVITE message) on mean Call Set-up Delay and mean Call Disengagement Delay can be observed. Higher T_{INV} values result in larger mean *CSD* and mean *CDD*. As in the case of Fig. 7, two data sets with different RACF message processing times are considered.

It is important that in Fig. 7 and Fig. 8 differences between calculations using analytical model and simulations can be observed for high load, where simulated mean *CSD* and mean *CDD* are larger than calculated. These differences, however, occur for

conditions avoided in practice. Moreover, as it can be observed in Fig. 7 and Fig. 8, with larger RACF message processing times (data sets 1b and 2b) and, thus larger $E(CSD)$ and $E(CDD)$, simulated results are closer to calculated.

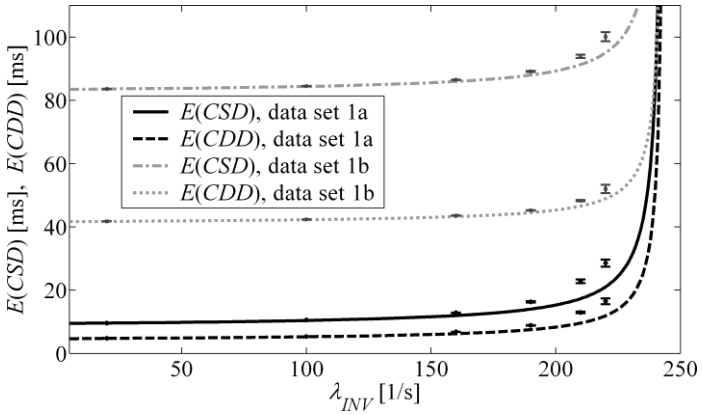

Fig. 7. Mean Call Set-up Delay and mean Call Disengagement Delay versus call set-up request intensity

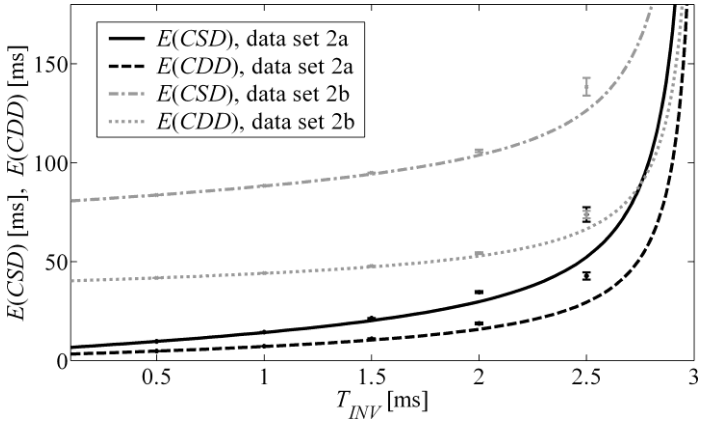

Fig. 8. Mean Call Set-up Delay and mean Call Disengagement Delay versus SIP INVITE message processing time by CSCF servers ($T_{INV} = T_{INV1} = T_{INV2}$)

Results presented in Fig. 7 and Fig. 8 are obtained based on the assumption that communication times are equal to zero, which means that all network elements illustrated in Fig. 3 are in the same place. The influence of non-zero distances between elements (lengths of optical links) on mean *CSD* and mean *CDD* is demonstrated in Fig. 9 and Fig. 10. For simplification of calculations and simulations it is assumed that all links between elements have the same length and bandwidth.

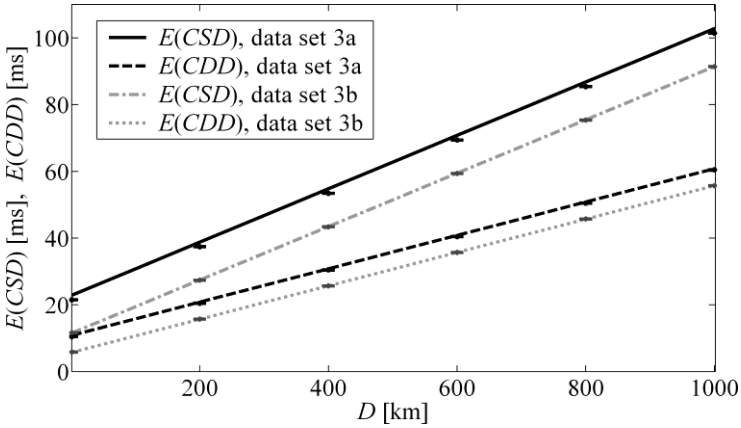

Fig. 9. Mean Call Set-up Delay and mean Call Disengagement Delay versus distance between network elements (length of optical links); data sets 3a and 3b (Tab. 1)

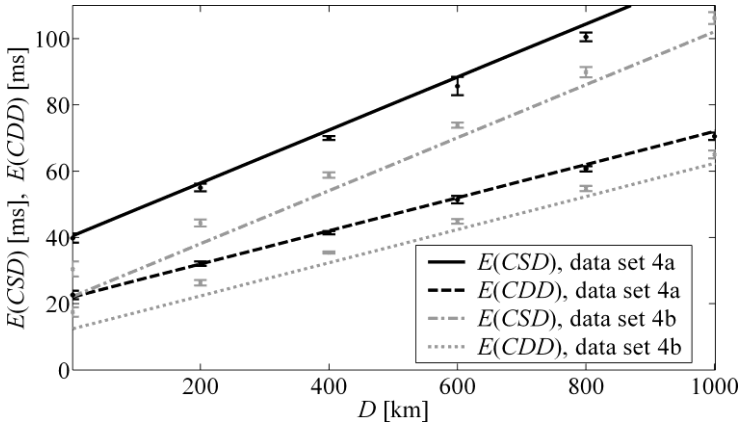

Fig. 10. Mean Call Set-up Delay and mean Call Disengagement Delay versus distance between network elements (length of optical links); data sets 4a and 4b (Tab. 1)

In Fig. 9 and Fig. 10 it can be noticed that mean Call Set-up Delay and mean Call Disengagement Delay increase linearly with length of optical links due to the distance dependent propagation time. It can also be observed that bandwidth available on links has visible impact on *E*(*CSD*) and *E*(*CDD*). For 10 Mb/s (data sets 3a and 4a) times of sending particular messages as well as times of buffering messages in queues are higher than for 100 Mb/s (data sets 3b and 4b). This results in higher mean *CSD* and mean *CDD* values. Performed experiments, however, demonstrate that 10 Mb/s link bandwidth is sufficient to carry signalling traffic regarding call set-up and disengagement even for high call set-up request intensities. Research regarding 1000 Mb/s link bandwidth was also carried out. As obtained results only slightly differ from these for 100 Mb/s, they are not marked in Fig. 9 and Fig. 10. In Fig. 10 higher call set-up re-

quest intensity is considered comparing to Fig. 9. This allows to observe differences in calculated and simulated results for high load (Fig. 10).

5. CONCLUSIONS

In the paper a traffic model of a single domain of ITU-T IMS-based NGN is proposed. The model allows to evaluate mean Call Set-up Delay and mean Call Disengagement Delay, which are a subset of call processing performance parameters formerly known as Grade of Service metrics.

Results obtained using the proposed analytical model were successfully verified by simulations, which proved that even for such complicated architectures like IMS/NGN for typically used parameters approximate network analysis using simple M/G/1 and M/M/1 queuing models gives sufficient results and can have practical applications. Differences between $E(CSD)$ and $E(CDD)$ for calculations and simulations are noticeable for high load, however, such conditions are avoided in practice. Furthermore, it can be observed that for higher RACF message processing times as well as lengths of optical links, these differences become less significant.

Although from practical point of view obtained analytical and simulation call processing performance results are very similar, for scientific reasons differences between them occurring for high load will be further examined and more proper queuing models for CSCF servers and optical links will be investigated. For this reason we are going to apply G/G/1 queuing systems, in which message inter-arrival distributions will be chosen based on the measurements taken in simulation environment. Apart from that, we are planning to develop our model in order to carry out call processing performance research in a multi-domain IMS/NGN architecture.

REFERENCES

[1] ITU-T Rec. Y.2001, *General overview of NGN*, Dec. 2004.
[2] 3GPP TS 23.228, *IP Multimedia Subsystem (IMS); Stage 2 (Release 11)*, v11.0.0, Mar. 2011.
[3] ROSENBERG J., et al., *SIP: Session Initiation Protocol*, IETF RFC 3261, Jun. 2002.
[4] CALHOUN P., LOUGHNEY J., GUTTMAN E., ZORN G., ARKKO J., *Diameter Base Protocol*, IETF RFC 3588, Sept. 2003.
[5] ITU-T Rec. Y.1530, *Call processing performance for voice service in hybrid IP networks*, Nov. 2007.
[6] ITU-T Rec. Y.1531, *SIP-based call processing performance*, Nov. 2007.
[7] KACZMAREK S., SAC M., *Traffic modeling in IMS-based NGN networks*, Gdańsk University of Technology Faculty of Electronics, Telecommunications and Informatics Annals, vol. 1, no 9, 2011, 457–464.
[8] LIN N., QI H., *A QoS model of Next Generation Network based on MPLS*, 2007 IFIP International Conference on Network and Parallel Computing, Liaoning, Sept. 18–21 2007.
[9] CHO I. K., OKAMURA K., *A centralized resource and admission control scheme for NGN core networks*, International Conference on Information Networking, ICOIN 2009, Chiang Mai, Jan. 21–24 2009.

[10] JOUNG J., SONG J., LEE S., *Flow-based QoS management architectures for the Next Generation Network*, ETRI Journal, vol.30, no.2, Apr. 2008, 238–248.
[11] ITU-T Rec. Y.2111, *Resource and admission control functions in next generation networks*, Nov. 2008.
[12] GUTKOWSKI P. S., KACZMAREK S., *Service time distribution influence on end-to-end call setup delay calculation in networks with Session Initiation Protocol*, First European Teletraffic Seminar, Poznań, February 14-16 2011, 37–42.
[13] GUTKOWSKI P. S., KACZMAREK S, *The model of end-to-end call setup time calculation for Session Initiation Protocol*, Bulletin of the Polish Academy of Sciences. Technical Sciences, vol. 60, no. 1, Jan. 2012, 95–101.
[14]HERNANDEZ A., ÁLVAREZ-CAMPANA M., VÁZQUEZ HADDADZADEH E., *Quality of Service in the IP Multimedia Subsystem*, The 5th COST 290 Management Committee Meeting, Delft, Feb. 9–10 2006.
[15] ABHAYAWARDHANA V. S., BABBAGE R., *A traffic model for the IP Multimedia Subsystem (IMS)*, IEEE 65th Vehicular Technology Conference, VTC2007-Spring, Dublin, Apr. 2007.
[16] 3GPP TS 23.002, *Network architecture (Release 10)*, v10.2.0, Mar. 2011.
[17] ITU-T Rec. Y.2012, *Functional requirements and architecture of next generation networks*, Apr. 2010.
[18] ETSI Standard ES 282 001, *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); NGN functional architecture*, v3.4.1, Sep. 2009.
[19] ITU-T Rec. Y.2018, *Mobility management and control framework and architecture within the NGN transport stratum*, Sep. 2009.
[20] ITU-T Rec. Y.2113, *Ethernet QoS control for next generation networks*, Jan. 2009.
[21] ITU-T Rec. Y.2121, *Requirements for the support of flow state aware transport technology in an NGN*, Jan. 2008.
[22] ITU-T Rec. Y.2175, *Centralized RACF architecture for MPLS core networks*, Nov. 2008.
[23] ITU-T Rec. Y.2021, *IMS for next generation networks*, Sep. 2006.
[24] KACZMAREK S., SAC M., *Traffic engineering aspects in IMS-based NGN networks* (*Zagadnienia inżynierii ruchu w sieciach NGN bazujących na IMS*), accepted for publication as a chapter in Teleinformatics library, vol. 6. Internet 2011 (Biblioteka teleinformatyczna, t. 6. Internet 2011), Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2012 (in Polish).
[25] ITU-T Rec. Q.3301.1, *Resource control protocol no. 1, version 2 – Protocol at the Rs interface between service control entities and the policy decision physical entity*, Jun. 2010.
[26] PIRHADI M., SAFAVI HEMAMI S. M., KHADEMZADEH A., *Resource and admission control architecture and QoS signaling scenarios in next generation networks*, World Applied Sciences Journal 7 (Special Issue of Computer & IT), 2009, 87–97.
[27] COOPER R. B., *Introduction to queueing theory*, Second Edition, Elsevier North Holland. Inc., New York, 1981.
[28] KACZMAREK S., KASZUBA M., SAC M., *Simulation model for assessment of IMS-based NGN call processing performance*, ICT Young 2012, Gdańsk, Poland, May 26–27 2012, 485–492.
[29] *OMNeT++ Network Simulation Framework*, www.omnetpp.org, Jun. 10 2012.
[30] WEINGARTNER E., VOM LEHN H., WEHRLE K., *A performance comparison of recent network simulators*, IEEE International Conference on Communications. ICC '09, Dresden, Jun. 14–18 2009.
[31] XIAODONG XIAN, WEIREN SHI, HE HUANG, *Comparison of OMNET++ and other simulator for WSN simulation*, 3rd IEEE Conference on Industrial Electronics and Applications 2008. ICIEA 2008, Singapore, Jun. 3–5 2008.

IPv6 QoS emulation, self-managing networks,
content aware networks, Execution Management,
Universal Communication Platform, SmartFit,

Damian PETRECKI, Bartłomiej DABIŃSKI,
Paweł ŚWIĄTEK*

PROTOTYPE OF SELF-MANAGING CONTENT AWARE NETWORK SYSTEM FOCUSED ON QOS ASSURANCE

Abstract: In this paper we consider the problem of providing QoS for vulnerable services in IPv6 based networks by using self-managing network architecture. We propose the system cooperating with services and network nodes that fixes connection paths and guarantees minimal bandwidth requested by a network service. In case of change in a network topology, every path may be changed without loss of any packet. Important requirement of the system was to work with heterogeneous network, so our approach is independent of device vendors, medium types and connectivity technology. The system is composed of stream services, QoS aware middleware and the prototype system based on IPv6 QoS network architecture that was proposed by Future Internet Engineering Project. In our system in order to create connection between two services, these services negotiate with each other using middleware. After successful negotiations, middleware, on behalf of the services, requests network resources from network management system. In this paper we also paid attention to description of the architecture in details and briefly discussed the system performance.

1. INTRODUCTION

During Self Managing Network Summit 2005 Jonas Svensson proposed requirements that should be met by a self managing network. Self-deploying and self-cleaning mean the ability to quickly add new devices to the network topology. Self-configuring and self-adapting base on the automatic SSID naming, automatic channels allocate and automatic IP addresses assigning if required. Self-optimizing consist in testing data flows in the network and proposing hardware changes that ensure fixed quality of services. Self-protecting is formed by default enabled security protocols (firewall, encryption, WPA2,

* Institute of Computer Science, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

etc.). Self-monitoring is traffic monitoring in order to detect errors. Self-diagnosing is permanent verification of connections and devices parameters to immediate detect or predict hardware failures. Self-healing bases on automatic establishing of alternative paths for broken connections. Prevention more than cure suggests that it is easier to avoid failures than repair it automatically. These terms are well described in [4].

In the presented system some of these requirements are attenuated but content awareness is added. Adding a device to the network only requires basic device configuration and inserting the device to the database. The security mechanisms are disabled because of laboratory usage of the network. Furthermore, paths for each connection are fixed after connection parameters negotiations for application scenarios.

In our research we modeled ourselves after Parallel Internet IPv6 QoS designed within the project Future Internet Engineering [3]. It's the content aware network focused on: data type awareness (stream, packet etc.), information contents awareness (use scenarios of applications), and user awareness (differentiation of individual users). Moreover, the network is designed to handle large amount of small pieces of data, e.g. transmitted from sensors within Body Area Network to e-health application. While Parallel Internet IPv6 QoS abandoned DiffServ in favor of non-standard solutions, our work is based on existing techniques and standards. We used e.g. IPv6-in-IPv6 tunneling, VLANs and HTB mechanisms for traffic engineering. According to self-organizing networks principle, these elements are automatically configured in response to request from higher level of system.

2. PROTOTYPE STRUCTURE

The authors assumed multi-layer architecture as is shown at figure 1. The highest level is an user application. This role plays the application e-health SmartFit that supports sportsmen's training [7]. This application processes data from sensors (that a sportsmen has on his body) along with data from external sensors (like video camera) and pass critical information through a server to a trainer. SmartFit handles also communication in the opposite direction (messages from a trainer) or can even aid a trainer in some cases (e.g. SmartFit is able to tell sportsmen to run faster if his pulse is too low). The application supports emergency scenarios like a fall of sportsman detection what is important because requires immediate rebuilding of connections graph in order to start emergency procedures. From network designer's point of view SmartFit is a distributed in many computers application with high and dynamic changing network requirements.

The middle layer is responsible for connection parameters negotiation and passing requests to a network resources management system. This role is played by Universal Communication Platform (UCP), which is compatible with SmartFit. UCP is made up of modules compiled along with an application, the UCP server and an XML-RPC

server. Services that work in the network do not know their locations but can communicate with the UCP server. The UCP server receives application's first request and indicates the second part of the establishing connection. The next step is negotiation of connection parameters. UCP recognizes services, identifies their type (packet, stream, emergency) and demanded bandwidth. After negotiation UCP send a request to the management system, receives a respond and mediate in the connection, which means that the application is network independent because data transmission is based on function calling of the UCP module. More about that you can find in [2].

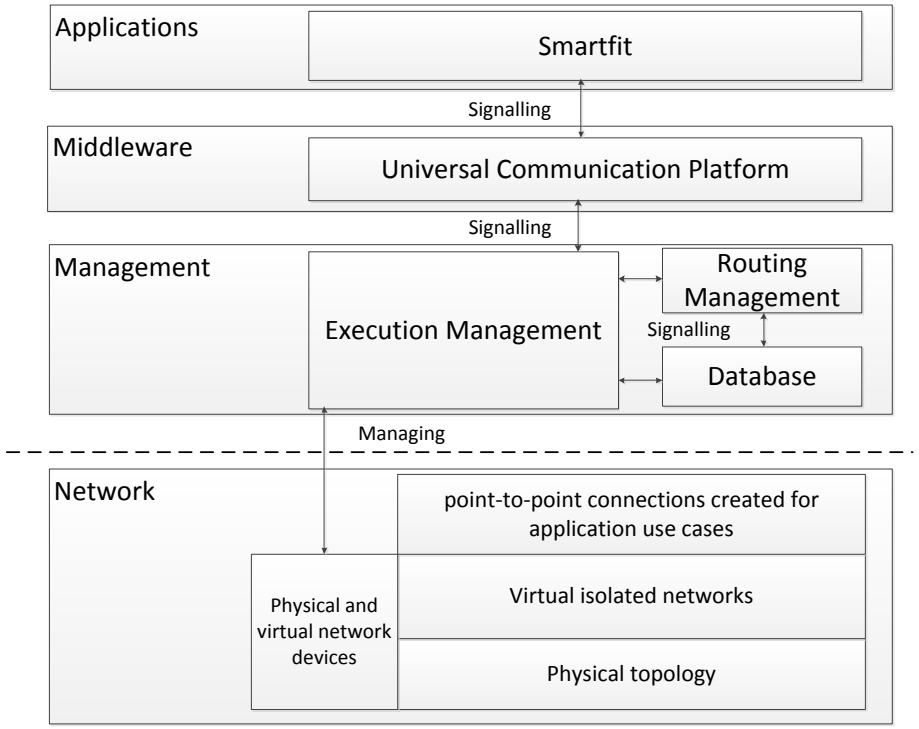

Fig. 1. System architecture

The management system is made up of three parts. Its main element is Execution Management (EM), which will be described later. The first supporting module is a database that stores current state of the network (nodes, physical and virtual connections etc.) and information necessary to system operating (predefined network devices, administrators accounts etc.). The next module is called Routing Management (RM) or QoS module and is responsible for calculating routes for new and modified connections. RM receives a request to calculate routes for new connections from EM, downloads via a database interface current network state (XML file) and calculates routes

that ensure QoS for new and existing connections with known parameters. Then RM returns the result of its working to EM. The structure of the management system is presented at figure 2.

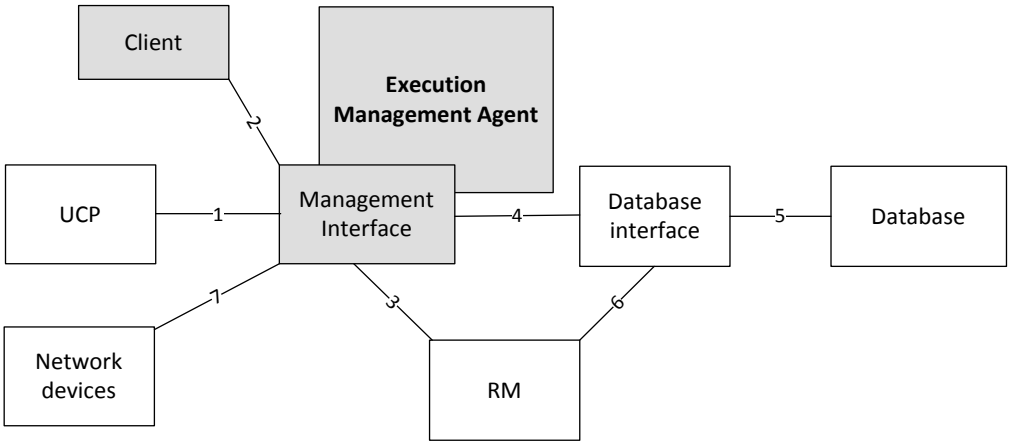

Fig. 2. Structure of the management system.

Hatched fields in this diagram represent the contribution of a team to which belong authors of this paper.

3. EXECUTION MANAGEMENT

EM is the heart of presented self-managing network. Originally, its role was to execute commands on remote hosts but in the course of time its role in the system became more and more important. EM performs two main functions.

The first function is to enable an insight into the state of the network and to enable remote network management via a web client application. This function makes way for adding to the database new nodes and connections. There is also ability to modify virtual networks. Nodes and topology of the virtual networks are modified with automatic configuring necessary network nodes. The web application supports two types of users: a network administrator and a Parallel Internet administrator. The former manages the whole network and is allowed to manipulate each of network resources. The latter manages a part of the network that forms a logical network. The network division and virtualization will be discussed later.

Second and more important role of EM is to configure network – establish connections for applications within this network. EM waits for requests (from UCP) of establishing, modifying or removing a logical network (in case of modification request, the author of a notification may be also RM). After receiving a request, EM starts a procedure of handling it. Firstly, there is assigning a number of logical networks along with new IP addresses and simultaneously the request of route calculating is sent to QoS module. After data collecting there is a stage for multi-threaded scripts creating. These scripts fall in two categories. In first category there are scripts that establish virtual connection and limit resources. The scripts of second category set routing for new logical network within the superior network. After scripts are prepared, there is launched a process responsible for simultaneously execution of commands on network nodes. In case of all operation succeeded, new connections are added to connections graph. In case of an error occurred (e.g. physical connection failure) the information about error is added to the response. After all actions are completed the response is sent and application's modules are able to communicate within their own, isolated, logical network.

The main advantage of the presented approach is transparency for higher levels. A sportsman runs the mobile application, all the operation described above are executed and now trainer is able to look into progress of the sportsman on a website. In middleware, after finishing negotiations, a request is sent and a response is received – independent on the network state and physical structure.

4. VIRTUALIZATION

The presented solution bases on two types of virtualization. First type is hardware virtualization and second is network connections virtualization. Both types are combined in the multi-level network resources virtualization system that is transparent to an end user and enable self-control network mechanisms. Presented solution was based on [6].

4.1. NETWORK NODES VIRTUALIZATION

The physical topology consists of routers, switches and computers. In details, for testing we used devices: MikroTik RouterBoard 800, Juniper EX4200 and Linux Debian Squeeze. MikroTik RB 800 enables user to create virtual routers (this function is called MetaROUTER) but current version of MikroTik operating system RoueOS 5.17 does not support policy routing for IPv6, which is crucial function for routers handling IPv6-in-IPv6 tunnels. Juniper supports creating logical nodes but not in EX-switches series although EX switches are L3 capable. Therefore, routers role were played by Debian Squeeze computers connected by Juniper EX. An end user may use

any device and any operating system supporting IPv6. Because our approach has very few requirements for end user devices, it may be even a smartphone that fully supports IPv6.

For virtualization on Debian nodes we used XEN 4.1 environment. Using XEN we created virtual routers and some end nodes. Network traffic is passed to appropriate virtual router using VLAN tags. Communication within the virtual network formed by virtual logical routers was ensured by OSPFv3 and static routes.

4.2. CONNECTIONS VIRTUALIZATION

The zero level of connection virtualization is physical connection. In devices we used to tests, there were gigabit ethernet connections.

The first virtualization level corresponds to Parallel Internet of Future Internet Engineering system. In the presented case the first level is accomplished by VLANs, which isolate fragments of network. Bandwidth guarantees for virtual connections of first level are assured by traffic limits per VLAN which are configured on Debian routers that host virtual routers. In figure 3 there are three such networks and only device common for these networks is Execution Management server. Each of the networks has different end nodes although the nodes may be running on same physical machine. Each network has also its own backbone that consists of virtual routers. In each network there should be a separate instance of UCP server and applications running in a particular network are not aware of existence of other networks. Within logical networks there is used dynamic routing protocol OSPFv3.

The important matter is connection bandwidth limiting. We had to use bandwidth limits for VLANs that the sum of these limits does not equal 100% of physical connection bandwidth. Otherwise there would be no free bandwidth for OSPFv3 and management packets then EM would have problems with accessing devices.

Second level of virtualization is formed by end-to-end connections (IPv6-in-IPv6 tunnels) for applications. For each scenario there is formed a graph of such connections. Each connection is able to work only within a VLAN in which operates an application that requests resources. Again, the network resources limiting is important. Tunnel IPv6-in-IPv6 must not use the whole bandwidth for VLAN because of routing OSPFv3, communication between EM and virtual nodes and UCP signalization. This type of traffic is also limited to value that depends on network size, in our case we assign 1 Mb/s. The limit concerns all the traffic in the virtual network (of the first level of virtualization) that is not IPv6-in-IPv6 encapsulated traffic. As we stated before, we do not require from end nodes supporting advanced network protocols and in particular ability to handle IPv6-in-IPv6 tunnels. This function is performed by their gateways, which are routers managed by our system. If an end device is in external, unmanaged network (e.g. GSM) its gateway is a first router in managed network. Packets are routed into a tunnel using policy routing mechanism, which select appropriate

packets using IP addresses, source ports and destination ports. The tunnel structure is presented on figure 4.

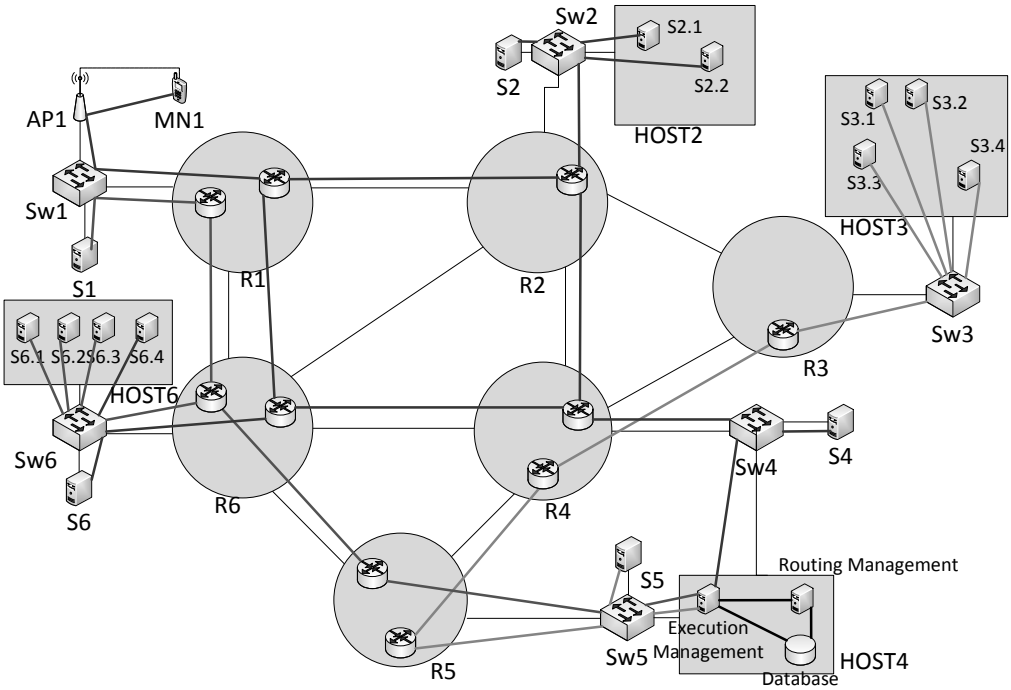

Fig. 3. Example network topology with isolated wirtual networks

Each tunnel has limited bandwidth (in both directions) on its ends. The path of a tunnel in one direction may be different than the path in the other direction. The paths depends on RM decision, which depends on free resources in the tunnel creation moment.

There are two situations, in which modification of existing tunnel is possible. The first situation is when QoS module takes decision to release a part of resources already assigned to a connection in order to assign resources to another connection. The second situation is when application requests modification of connection parameters (e.g. user enables video while a teleconference). In both cases following actions are taken. Firstly, current connection is removed from the database. In that the old connection will not be considered in calculations of the new connection. Next steps are identical as in the case of establishing new connection. When new tunnel is created, traffic is redirect to the new tunnel. When it is certain that all the packets sent to the old tunnel reached their destination, the old tunnel is removed and the new connection is added to the database. In case of error all the changes are rolled back and the old connection entry is added to the database. This approach ensures QoS and constant flow of

packets while tunnel switching. An application does not see any change because only nuisance is the ability that several packets will be reordered and this problem is mitigated by TCP mechanisms.

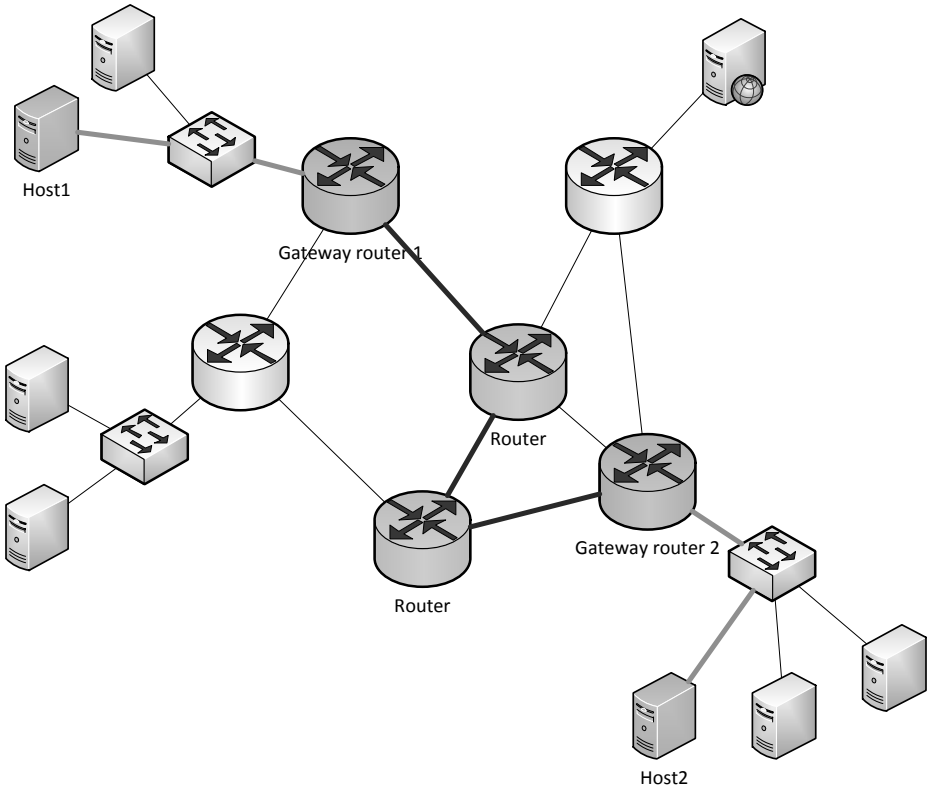

Fig. 4. IPv6-in-IPv6 tunnel with static routes sample

5. PERFORMANCE

The presented system should comply with ITU-T recommendation, which states that connection establishing can not last longer than 9 seconds. We attempted to establish many connections for an application. The connections form multiple connections graphs (virtual networks required by application). In this environment we got average connection establishing time of 7 seconds, while lowest result was 3 seconds. This is minimum time determined by the time of executing commands on single network node. We chose synchronic method to send commands. In this way we guarantee that commands were executed successfully and a new connection works correctly. The

time does not depend considerably on the number of nodes in the network and the number of nodes in a path due to multi-threaded sending configuration scripts. The only factor that may make working of the system slower is a flush of tasks coming before previous tasks are finished. New request has to wait because modifying the database is prohibited while another connection is establishing at the moment.

6. CONCLUSION

It seems to be impossible to design and deploy network system that is capable of self-managing in terms of postulates from Introduction. It is impossible to monitor network traffic in large enough measure to forecast a node failure or a connection failure without influencing network traffic. It is also impossible to implement security mechanisms that are effective and do not disturb an user to work. The next problem is necessity to use existing solutions combined with IPv6. The protocol IPv6 is not fully supported by many vendors of network equipment.

On the other hand, it is worth to mention that content and application awareness lets abandon permanent monitoring of network traffic and enables easier network management. When an administrator knows what kind of traffic is transmitted, he may manage this traffic and isolate particular types of traffic what is easier than in case of random network flows. We present mechanisms that do not need human action for this activity. The presented system of signalization, negotiation, establishing, modifying and removing connections is sufficient for everyday system operating. An intervention of the network administrator is required only in case the physical network was modified or hardware failure.

The layered structure of the system let make working of lower layers transparent for higher layers. Then higher layers are independent on lower layers implementation. In this way the application SmartFit may work either in a system controlled by Execution Management or in the Parallel Internet IPv6 QoS. The layered structure of virtualized networks enables to constrain network awareness of the application, which knows only its own logical network. Transmission of non-signalizing traffic within controlled tunnels ensures QoS for each connection.

The presented testing network environment is operational and may act as a reliable platform for testing user applications. The alternative solutions are Future Internet Engineering IPv6 QoS and OpenFlow with an intelligent controller capable of responding to resources requests. Both solutions offer higher throughput than the presented approach but the former requires expensive hardware (like NetFPGA, EZ Appliance) and the latter requires a driver cooperating with applications or middleware (and the list of devices supporting OpenFlow is currently short).

To sum up, we successfully designed and deployed a self-managing network that is ready to everyday operation and requires manual action only in special cases. The network is content aware and bases on existing, well known and widely supported standards and network protocols.

Current works are focused on replacing the external database module. Our implementation of a database will eliminate some problems and improve effectiveness of the system. Except that, we are working out new functions of client web application and we are performing accurate tests of the system in variety of environments. Furthermore, authors prepare to release Release Candidate version of Execution Management with QoS module.

REFERENCES

[1] BURAKOWSKI W., TARASIUK H., BĘBEN A. *Future Internet architecture based on virtualization and co-existence of different data and control planes,* 2011.

[2] ŚWIĄTEK P. R, RYGIELSKI P.: *Universal comunication platform for QoS-aware delivery of complex services. Proceedings of the VIth International Scientific and Technical Conference*, CSIT 2011, Lviv, Ukraine, 16–19 November 2011, 136–139

[3] TARASIUK H. et al., *Architecture and mechanisms of Parallel Internet IPv6 QoS*, Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne. 2011, vol. 84, no. 8/9, 944–954 (in polish)

[4] FARRICKER J., LABOVITZ C., PANDEY S., SVENSSON K. J., *Panel Discussion on Self-Management - What Does it Mean & Can it be Effective?*, Self-Managing Networks Summit 2005

[5] GĄSIOR D., DRWAL M., *Decentralized Algorithm for Joint Data Placement and Rate Allocation in Content-Aware Networks*, Computer Networks 2012

[6] CHUDZIK K., KWIATKOWSKI J., NOWAK K., *Virtual Networks with the IPv6 addressing in the Xen Virtualization Environment*, Computer Networks 2012

[7] KLUKOWKI P., PETRECKI D., ŚWIĄTEK P., GAWOR D. *Prototype of remote monitoring human vital signs in IPv6 network with QoS querantee system,* ICT Young 2012, Gdańsk 2012 (in polish)

Content Aware Networking, Content Centric Networks,
Cache, Future Internet

Remigiusz SAMBORSKI*

DATA CACHING IN CONTENT AWARE NETWORKS – LRU, LFU EVALUATION

This document provides an evaluation of caching mechanisms that may be used in Content Aware Networking.

Two algorithms has been evaluated: LRU (least recently used), LFU (least frequently used). The evaluation was made in an open source simulator Omnet++ with an implementation of CCNx network.

Research results include measurements in 4 topologies with different cache sizes. Parameters that were compared are: number of hops to nearest content, cache exchange ratio, cache hit ratio. The best results for these parameters were achieved with LFU algorithm, therefor it is better suited for usage in future Content Aware Networks similar to CCNx network.

1. INTRODUCTION

1.1. CONTENT CENTRIC NETWORK

Content Centric Network is a project created and supervised by Van Jacobson at Palo Alto Research Center, USA [1]. CCN is one of available prototypes of Content Aware Networks – a new way to look at networking. The main difference is to identify content (what) instead of network location (where). CAN approach requires content to be signed and versioned. The content is decoupled from it's location and way of transportation – it may be delivered to end user via Ethernet, IPv4 , IPv6, Pen-drive or any other mean of data transmission. This separation has many benefits for users and developers. Other assumptions for the new generation of network are: built-in data

* Wrocław University of Technology, Faculty of Computer Science and Management, 50-370 Wroclaw, Wybrzeże Wyspiańskiego 27.

dissemination to allow better bandwidth utilization and efficiently use devices potential; support for mobility and augmented Internet.

Figure 1 shows differences between IP and CCN protocol stacks. CCN introduces two new layers: strategy and security. The strategy layer is responsible for making choices to optimize usage of multiple conductivities in layer 2 under changing conditions.

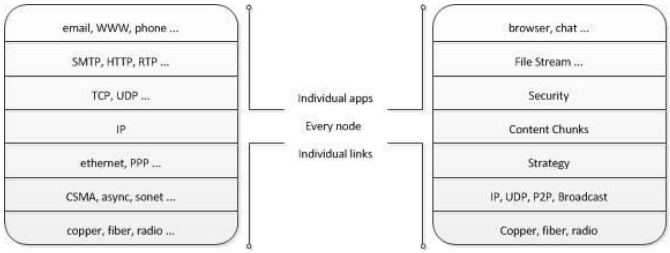

Fig. 1. IP and CCN protocol stacks [2]

Content in CCN is divided into content chunks, which are shared between hosts and are considered an universal agreement in this architecture – similar to an IP packet in IP stack. Communication is based on two types of packets: Interest and Data (figure 2). Interest packets can be called requests for data. These packets are broadcasted by the user, who demands content. An Interest packet is routed by name until it reaches a node, which has the demanded data. A Data packet returns using the same path used by an Interest packet to the user. CCN's architecture allows Interest packets to be send by multiple interfaces (called faces for distinction between IP protocols stack) simultaneously.

CCN namespace is human readable. This architecture uses URI like scheme for naming data. Routing is based on names and longest-match look-up. Data names may represent local context, for instance: /ThisRoom/projector, /Local/Friends etc.

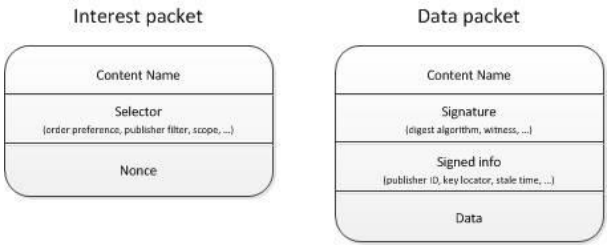

Fig. 2. Interest and Data packets in CCN [2]

Names create a hierarchical tree, which has these advantages among other: routing protocols from IP architecture may be adapted to work with CCN; trust system (key distribution) may be based on this hierarchy; content doesn't need to be registered. To keep content names human-readable they cannot be self-signing. In CCN the data and name signature is located in Data packet (figure 2).

Basic packet handling in a CCN node is very similar to an IP node. First the packet reaches a face, next a longest-match lookup is made to decide what should be done with the packet. CCN node's engine is presented in figure 3.

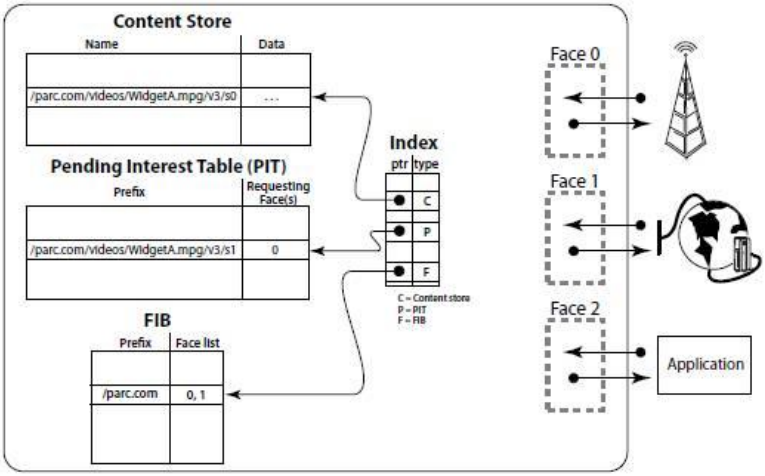

Fig. 3. CCN node's engine

CCN node contains 3 main tables:

Content store – content buffer, caches the data packets, so they may be used when requested from other nodes. In this article we will be testing network behavior with different sizes of this buffer.

PIT – Pending Interest Table contains information on recent content queries. If a new interest arrives and another query for the same content was recently sent than the face of the incoming packet is written in PIT. This table is responsible for Data packets traveling in reverse direction to Interest packets.

FIB – Forwarding Information Base, similar to a routing table used in IP nodes. It contains information for the longest-match lookup bundled with face identifiers. The main difference from IP is that it may connect many faces with on content name prefix – this tells the node that it should send Interest packets to more than one output face.

Figure 4 presents a block diagram of engine at work.

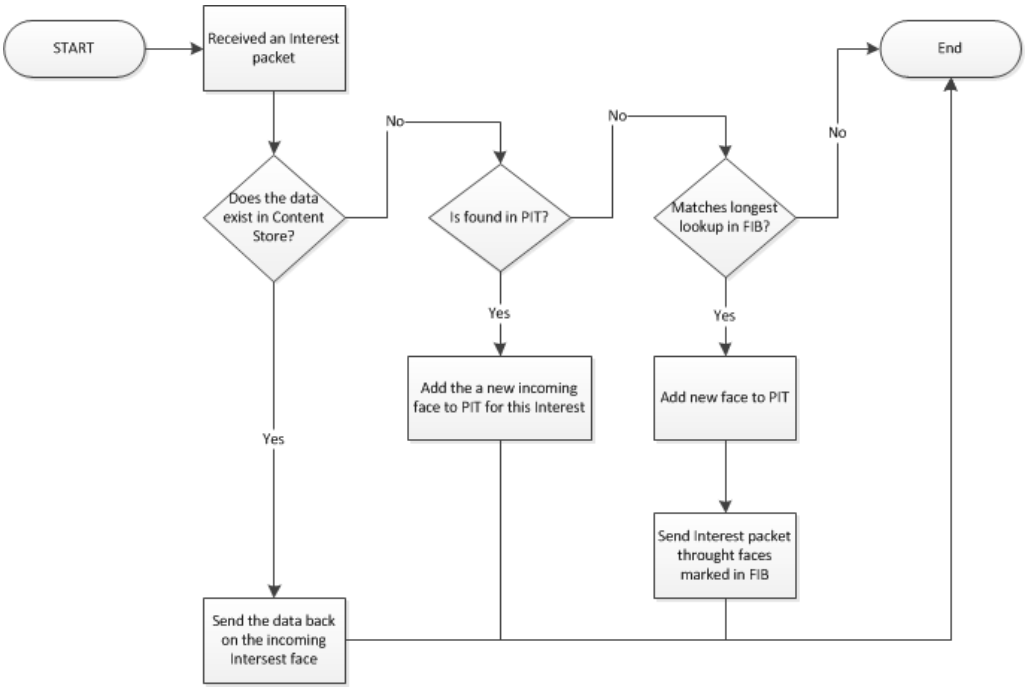

Fig. 4. CCN routing engine diagram

1.2. SIMULATION ENVIRONMENT

Based on CCNx documentation and prototypes code a simulation environment was developed in an opensource network simulation framework – Omnet++ [5]. The environment is freely available for further development at GitHub [6]. It was written to allow simulation of a CAN network similar to CCNx. The environment emulates Interest and Data packets travel through a network of nodes. A single node consists of 4 main objects represented by C classes: routing, application, contentStore and queue (one for each face – a connection to another node). This model is showed on figure 5.

The most advanced and crucial element of the model is the routing class, which is responsible for making decisions based on incoming packets. It realizes in practise the routing engine described in previous chapter. Application class is simulating user or application requests for data (generating new Interests) and servicing the received content (Data packets).

contentStore is responsible for the caching mechanisms. In scenarios presented in this article we have been using two different classes (LFUContentStore and LRUContentStore) to represent algorithms that has been tested. Currently the simulator allows

to use only one method in the whole network at a time. In the future contentStore might be treated as an abstraction class with LFU and LRU classes being it's implementations.

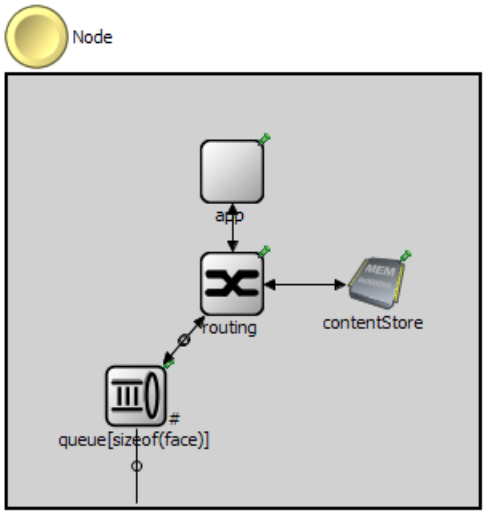

Fig. 5. OmnetCAN's node model

Queues are standard Omnet++ L2Queue classes, which provide connections to different nodes without packet loss.

Networks topologies may be constructed by interconnecting the described nodes. Every node is autonomous and makes it's own decisions based on it's own PIT, FIB and contentStorage. The most important simulation parameters may be set in a config file. These paramaters may be set globally or per node. This gives users a huge amount of flexibility in testing different configurations.

Some simplifications were made to shorten the development time, which should not have a major influence on study results:

- content is identified by integer numbers called contentIds instead of content names, therefore there is no longest-match lookup mechanism implemented
- each node may contain only one content with one contentId
- nodes search for random content in boundaries set in the configuration file
- network datarate (bandwidth), delays and packet sizes are set globally for the time of one simulation
- packets contain no real data

Simulation results were collected using Omnet++ builtin mechanisms such as signals and statistics.

1.3. CACHING ALGORITHMS

Caching is a widely used practice in all types of networks. Normally when users are requesting data it has to be downloaded from the source node. High frequency of requests leads to congestions and long response times. Content caching in IP is being done in special servers called proxies or in an overlay network of CDN servers. Not every node may participate in caching, because of protocol limitations. Main reason for this is that nodes transmitting data identify it by source and destination hosts and are not aware of the content. These limitations doesn't exist CAN networks, where content identification is the basic assumption. Therefore every node, which is participating in transmission may also participate in caching and data dissemination. It may use it's free resources to improve overall network efficiency.

The amount of content available in current networks is enormous. It's not possible to cache all of the content in any node of the network. There will always be more content available and sent through each node than its capacity to store it. For the cache to work efficiently there has to be a replacement policy, which clears data to make place for new content.

In this research different cache memory replacement algorithms has been tested in CCNx simulation environment. The most popular policies are:

- LRU – Least Recently Used element is removed
- LFU – Least Frequently Used element is removed
- MRU – Most Recently Used element is removed
- RR (Random Replacement) – random element is removed

According to [2] cache (content store) is the first place that is being checked before PIT and FIB tables, this leads to a conclusion that the efficiency of replacement algorithms should be similar to traditional IP networks. Therefore we have chosen LRU and LFU as the most popular for further research.

2. SIMULATION

2.1. TOPOLOGIES AND PARAMETERS

Simulations were executed in three different topologies:

- Ring – ring with 40 nodes (figure 6)
- Extended tree – binary tree with 40 nodes (figure 7)
- Irregular – irregular topology with 40 nodes (figure 8)

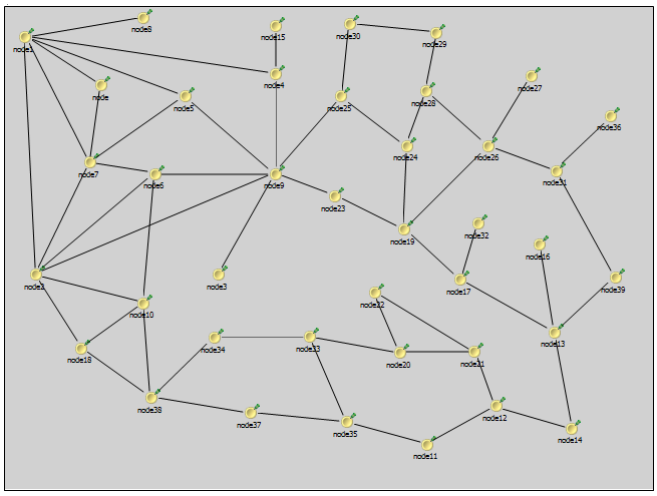

Fig. 6. Irregular topology

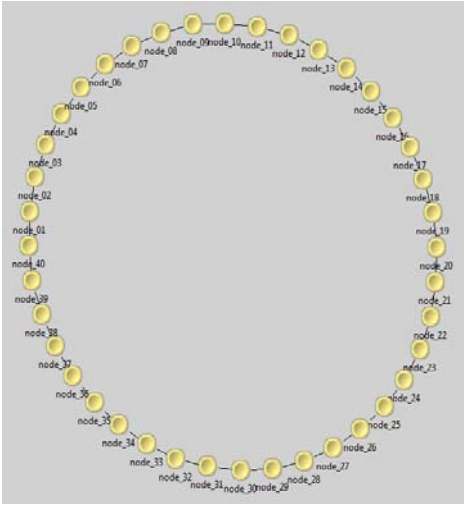

Fig. 7. Ring topology

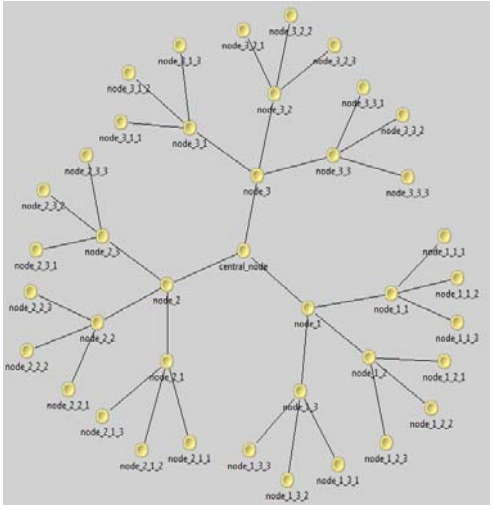

Fig. 8. Extended Tree topology

In each topology content has been evenly distributed over all available nodes (each node provides a different content ID). The distribution was the same for all measurements in the same topology. Requests for content were generated from each node in the network and included random content ID.

Research was carried out with cache sizes of: 50%, 25%, 15%, 10% and 5% of total amount of content available in the network. For instance: networks of 40 nodes contained 40 different content ids, tested cache sizes were: 20, 10, 6, 4 and 2.

All measurements were done with a time limit of 50 simulation seconds or more to get ensure proper results. These parameters were measured and compared:

- **hopToContent** – mean number of hops (visited nodes) made by an Interest packet before getting to a node containing cached content
- **exchangeRatio** – mean percentage of cache writes that required old data to be removed
- **hitRatio** – mean percentage of Interest packets that were handled by data contained in cache

2.2. RESULTS – LFU

Tables 1 to 3 present results for different topologies with Least Frequently Used replacement policy.

Table 1: Ring topology

cache size	50% (20)	25% (10)	15% (6)	10% (4)	5% (2)
hopToContent	1,94	3,92	6,80	8,94	13,86
exchangeRatio	80,85%	91,31%	96,20%	97,72%	99,14%
hitRatio	35,28%	18,83%	9,47%	6,08%	2,48%

Table 2. Extended Tree topology

cache size	50% (20)	25% (10)	15% (6)	10% (4)	5% (2)
hopToContent	2,02	3,44	3,77	3,90	4,07
exchangeRatio	49,07%	62,69%	72,66%	79,14%	87,39%
hitRatio	46,02%	26,52%	15,59%	9,75%	4,11%

Table 3. Irregular topology

cache size	50% (20)	25% (10)	15% (6)	10% (4)	5% (2)
hopToContent	2,82	4,57	5,31	5,60	5,60
exchangeRatio	47,15%	59,08%	65,04%	66,56%	66,89%
hitRatio	43,91%	25,56%	17,10%	9,93%	4,86%

Figures 9 to 11 present comparisons of parameters between different topologies.

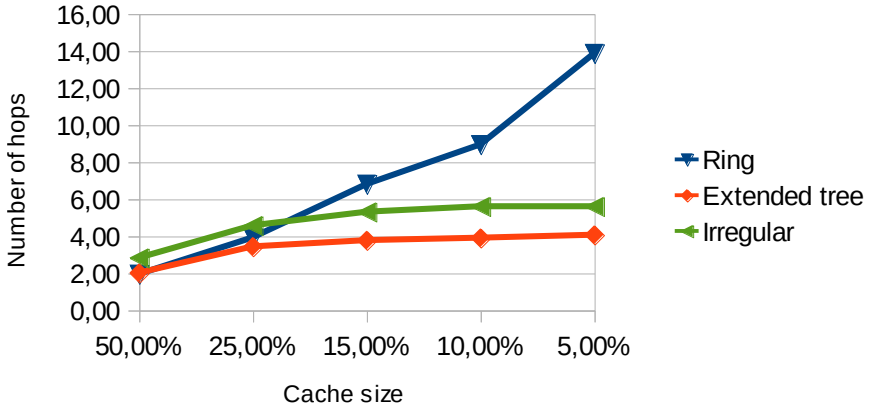

Fig. 9. LFU, hopToContent parameter with different cache sizes and topologies

As presented on figure 9 the mean value of hops stays on a low level for all topologies except from ring. In extended tree and irregular topologies the increase of number of hops is a logarithmic growth, which is a very good characteristic. The ring topology show an exponential growth due to it's nature – when the cache is smaller the packet has to travel through more nodes to reach it's destination.

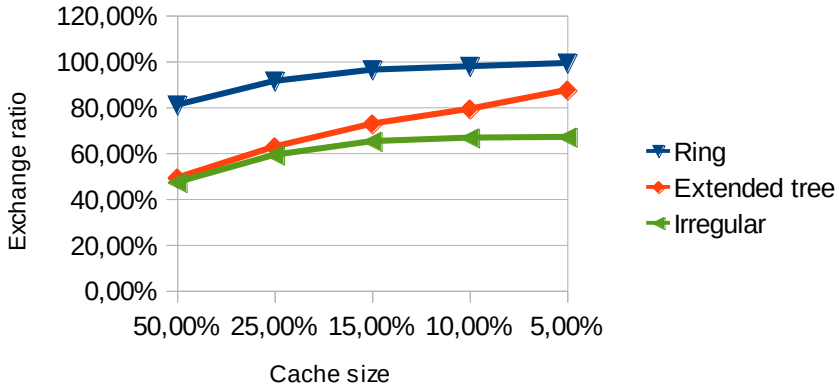

Fig. 10. LFU, exchangeRatio parameter with different cache sizes and topologies

Exchange ratio is increasing, when cache is getting smaller – this is a normal and predictable behavior. For irregular and tree topologies the values are below 50% at cache size of 20 (50% of all content available). The growth is logarithmic in the irregular network and almost linear in the tree topology. The ring topology has more exchanges in the cache regardless it’s size.

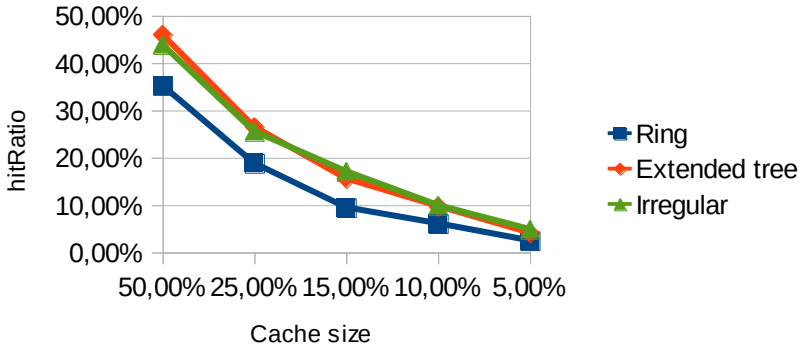

Fig. 11. LFU, hitRatio parameter with different cache sizes and topologies

Figure 11 shows that hitRatio is getting worse, when the cache size is being decreased. We may observe an exponential decay of this parameter in all topologies. Again ring has the worst values. This confirms that the ring topology requires bigger caches than other tested topologies to keep the same parameters.

2.3. RESULTS – LRU

Tables 4 to 6 present results for different topologies with Least Recently Used replacement policy.

Table 4. Extended tree topology

cache size	**50% (20)**	**25% (10)**	**15% (6)**	**10% (4)**	**5% (2)**
hopToContent	3,23	3,75	3,93	4,03	4,12
exchangeRatio	50,38%	62,21%	69,73%	75,20%	84,79%
hitRatio	34,83%	17,44%	10,63%	7,16%	3,51%

Table 5. Ring topology

cache size	**50% (20)**	**25% (10)**	**15% (6)**	**10% (4)**	**5% (2)**
hopToContent	4,49	11,12	14,16	15,74	17,16
exchangeRatio	92,74%	98,23%	99,13%	99,46%	99,69%
hitRatio	16,32%	4,82%	2,60%	1,63%	0,78%

Figures 12 to 14 present comparisons of parameters between different topologies.

Table 6. Irregular topology

cache size	50% (20)	25% (10)	15% (6)	10% (4)	5% (2)
hopToContent	4,23	4,97	5,28	5,42	5,55
exchangeRatio	48,45%	50,98%	52,05%	53,19%	57,47%
hitRatio	21,73%	9,93%	5,99%	3,88%	2,05%

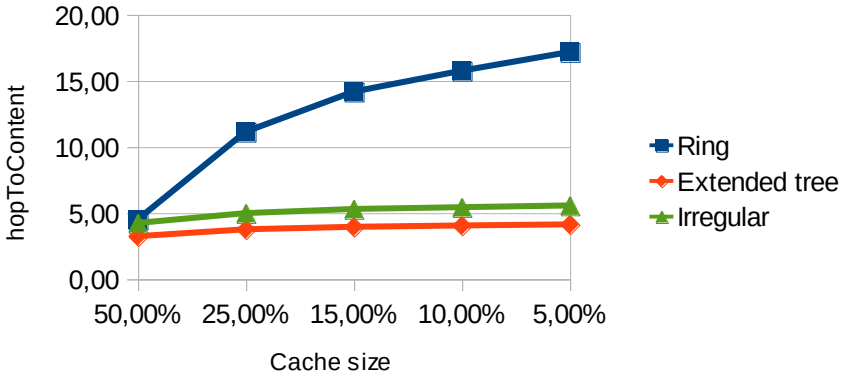

Fig. 12. LRU, hopToContent parameter with different cache sizes and topologies

Figure 13 shows the behavior of LRU algorithm in different topologies. Number of hops increases with cache size decrease. All topologies show a logarithmic growth of the value with cache size decrease.

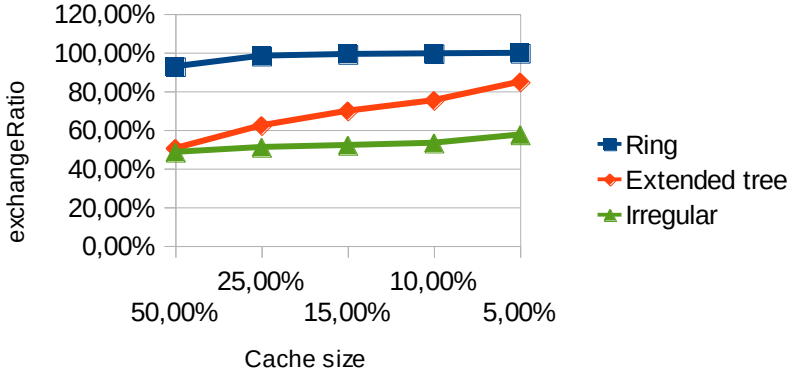

Fig. 13. LRU, exchangeRatio parameter with different cache sizes and topologies

Extended tree and irregular topologies have a very good exchange ratio of approximately 50% for cache size of 50%. Due to to cache decrease the exchange ratio is getting higher. We may observe a very good value of below 60% for irregular network

with cache size of 5% (2 contents). Ring topology has a high value of exchange ratio of over 90% for all cache sizes used.

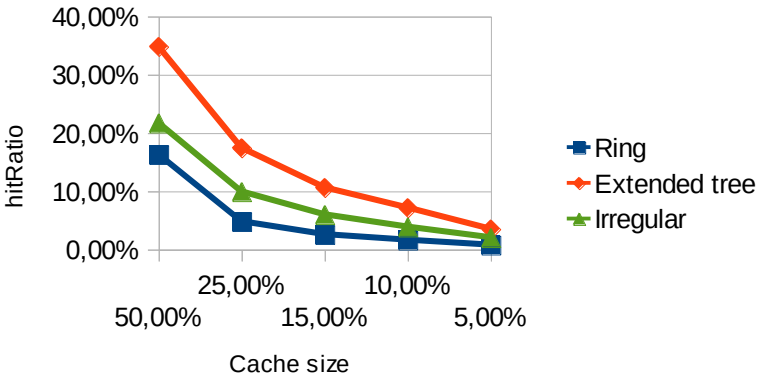

Fig. 14. LRU, hitRatio parameter with different cache sizes and topologies

Figure 14 shows the behavior of hitRatio parameter in different topologies and cache sizes. We may observe an exponential decay for all topologies with the cache size decrease. The best values are observed in the extended tree topology.

2.4. COMPARISON

Figures below show a comparison of all measured parameters in different topologies.

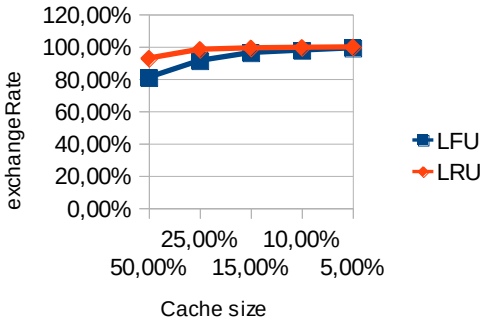

Fig. 15. exchangeRate in ring topology

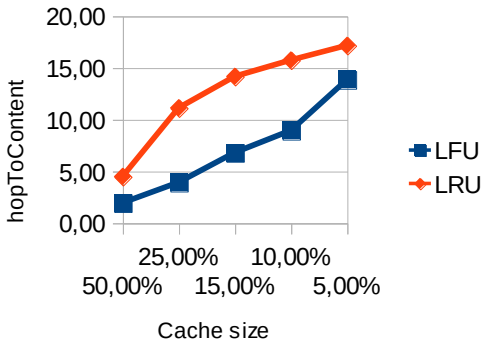

Fig. 16. hopToContent in ring topology

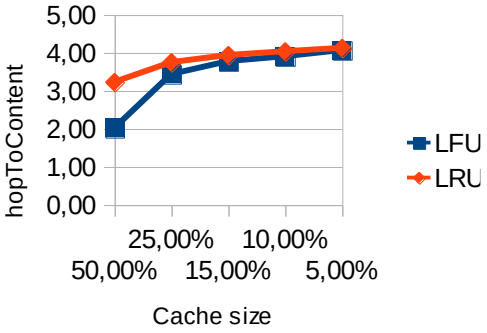

Fig. 17. hopToContent in extended tree topology

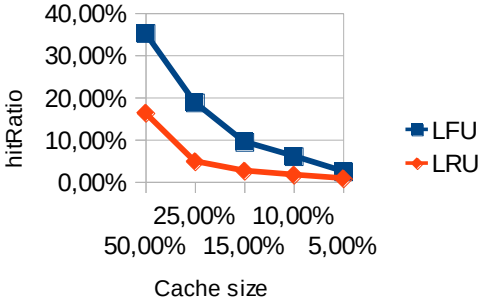

Fig. 18. hitRatio in ring topology

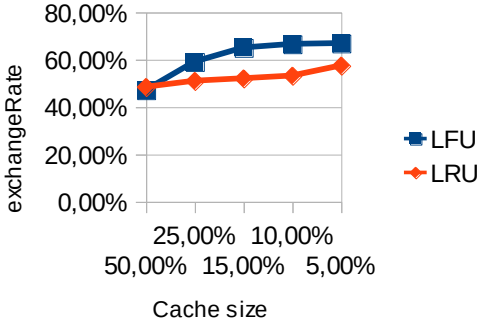

Fig. 19. exchangeRate in irregular topology

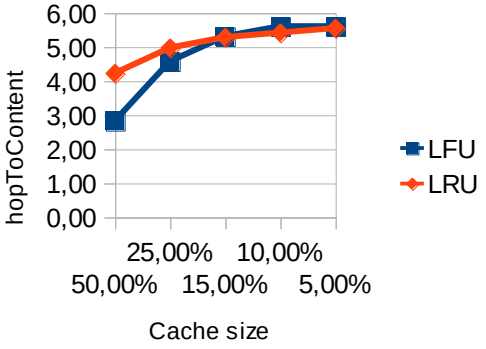

Fig. 20. hopToContent in irregular topology

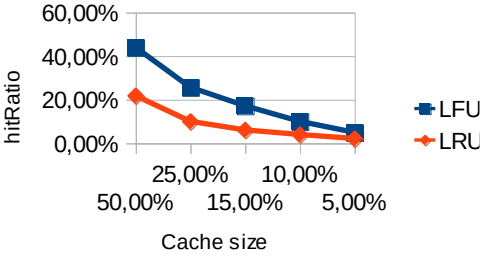

Fig. 21. hitRatio in irregular topology

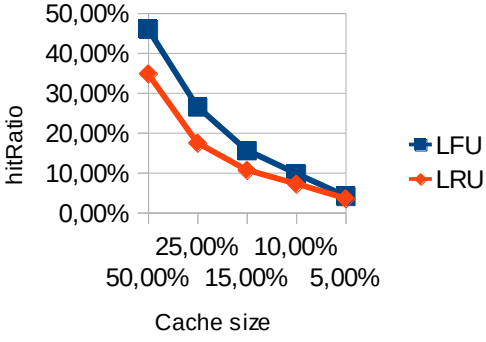

Fig. 22. hitRatio in extended tree topology

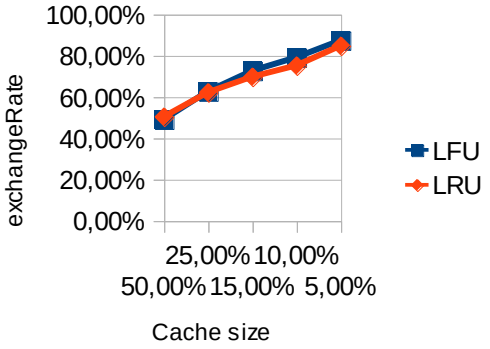

Fig. 23. exchangeRate in extended tree topology

We observed a better behavior of the LFU algorithm over LRU in almost all parameters and topologies. LRU has a better (smaller) values of exchangeRate parameter in two network topologies: extended tree and irregular. This hover doesn't lead to a higher hitRatio or lower value of hops to content parameter.

3. SUMMARY

After conducting research described in this article we may conclude that Least Frequently Used caching algorithm is better suited for CCN network prototypes than Least Recently Used algorithm. The only parameter, which was observed to be better in LRU was the exchangeRate. Nevertheless it didn't lead to a shorter path to content nor higher cache hit ratio. Taking into account that currently used devices may provide users with a very high rate of cache elements swapping, this parameter is not crucial. The most important parameters for modern caches are hitRatio and shorter path to content, which were obtained with LFU.

The environment, which was prepared in Omnet++ may be used to create various other simulations of CCN network behavior. This may lead to other interesting and promising studies of this future network prototype.

ACKNOWLEDGMENT

I would like to thank my students Ewa Kaczmarek and Joanna Kaczmarek, who have helped me in conducting of the above studies. I would also like to thank Professor Adam Grzech for his insights and help with this article preparation.

REFERENCES

[1] Project CCNx™. http://www.ccnx.org, Sep. 2009.
[2] JACOBSON V., SMETTERS D.K., THORNTON J.D., PLASS M.F., BRIGGS N.H., BRAYNARD R.L., *Networking Named Content*, CoNEXT 2009 Conference.
[3] JACOBSON V., *A New Way to look at Networking*, Google Conference, 30.09.2006.
[4] CCNx™ project group, *CCNx Documentation,* 2011.
[5] OMNeT++ Network Simulation Framework, http://www.omnetpp.org
[6] OmnetCAN – a CCNx simulator for Omnet++ by Remigiusz Samborski, https://github.com/rsamborski/OmnetCAN
[7] .JANG M.-W, Lee B.-J., *Content-Centric Networking, The Vision and an Early Experience*, Samsung Advanced Institute of Technology Communication & Networking Group 2011. 1. 19.
[8] KOURLAS T., *The Evolution of Networks beyond IP*, IEC, March 2007.

computer networks, data security,
IP storage, IPSec, iSCSI.

Tomasz BILSKI*

NETWORK STORAGE SYSTEMS WITH IPSEC IMPLEMENTATIONS

Network storage service is one of many network services. It is a system in which storage media are accessed by users with a use of IP network. The service has some distinctive features. For transmitted data protection IPSec protocol is commonly used. The main problem is the existing IPSec solutions do not provide the throughput required for storage systems (encryption/decryption processes, especially in software mode, trigger significant performance degradation up to 75%), while new high-speed hardware IPSec processors that are designed for VPN traffic will not work efficiently with data storage traffic. Another drawback of hardware crypto implementation is lack of flexibility. The chapter presents distinctive features of network storage services and some remarks on negative aspects of IPSEC usage in storage applications. The main part is dedicated to some different solutions to IPSec usage: hardware cryptography accelerators, double implementation of IPSec, packet grouping and scheduling algorithms, lazy crypto approach.

1. INTRODUCTION

Network storage system is a system in which storage media are accessed by users/applications with a use of local or wide area network. It is assumed that data are accessed without needing to alter storage applications – user (or application) may be unaware that data are remote not local. Network storage systems are becoming common. They have many applications, among them: data protection (backup and recovery), tier-2 storage level, replication. If network storage system is based on public IP network problems with data security should be solved. Evaluating and selecting the security solution for a given system we have to analyze many intricate criterions: se-

* Poznań University of Technology, Pl. Marii Skłodowskiej-Curie 5, 60-965 Poznań, Poland.

curity level, performance, key management, cost. Transmitted data are usually protected with a use of security mechanism at transport (e.g. Secure Socket Layer) or internetwork layer (IPSec). It may seem that data security problems may be solved by standard methods for network protection. IPSec is more often used in network storage systems based on iSCSI (Internet Small Computer System Interface) or FC (Fibre Channel) application layer protocols. Nevertheless, features of storage system traffic as well as security requirements of the systems are unusual. IP storage security problems cannot be solved with the same methods and tools that are dedicated to protect such common services as email, VoIP, WWW.

The security requirements of IP storage system involve three common aspects and two sets of threats. The security aspects are well known: confidentiality, integrity and availability. The first set of threats is related to communication channel, the second – to storage media. The security requirements should be preserved while data are transmitted and stored. Data integrity and availability are always required. Confidentiality is obligatory in the case of sensitive data.

2. DISTINCTIVE FEATURES OF NETWORK STORAGE SERVICE

Contemporary computer networks are used for many different services such as email, voice transmission, WWW, FTP, social networks and so on. Network storage service is one of them. The service has some distinctive features. The uniqueness of the transfer characteristics as well as distinctive security problems has significant impact on system performance and security.

First of all storage service has distinguishing characteristics of datagram flows:

- gigabyte level of volume of data,
- relatively great number of large datagrams,
- long-lived sessions between endpoints,
- intensive handshaking processes related to iSCSI (Internet SCSI) session management.

It is well known fact that common TCP/IP protocols are not optimized for given above traffic. They were designed many years ago for completely different purposes. Common solution in network storage devices is hardware TCP/IP implementation. Some network storage systems use TOE (TCP/IP Offload Engines) for low-latency processing. From the security point of view it is a drawback because IPSec acceleration hardware must incorporate the full range of IPSec functions and this is hardly possible with TOE.

Threats in network storage systems are similar to those related to such services as email and VoIP. Nevertheless the security problems are exclusive. In the case of these

common services as a rule confidential information (e.g. confidential email) is transmitted just once and exact time of transmission is usually hard to predict. On the other hand in network storage system a single piece of confidential data may be transmitted many times from client to storage server and back. Furthermore, if remote storage is used as backup then transmission point in time may be predicted – some backup services transmit data on a regular basis. So, eavesdropping and session hijacking risk levels are higher. Another network storage distinctive security issue is related to data sharing scheme. In remote storage systems we may encounter two categories of data: private and shareable. If data are private then protection methods may be unique to particular data owner. But, if data are shareable then security solutions must work in such a way that each client is able to access (read) data that were written and eventually encrypted by another client.

In general, it must be assumed that the existing tools and methods for securing common services (WWW, email...) are not suitable for network storage data protection. Dedicated protection methods and tools are necessary.

3. ISCSI STACK OF PROTOCOLS

Common solution to network storage service is Internet SCSI (iSCSI) application layer protocol. iSCSI protocol is frequently used in LANs and WANs. It enables block level operations on distant storage devices running over TCP/IP transport. At client side, SCSI commands, status and optionally data blocks are placed in iSCSI messages which are transferred by TCP/IP. At the server side, command or data are retrieved from iSCSI message and moved to storage device.

iSCSI devices are globally identified with a use of names and addresses. Names are defined according to iSNS (Internet Storage Name Service), addresses are built according to URL (Uniform Resource Locator) specification.

Since SCSI protocol and SCSI bus are commonly used to connect the components inside a computer, SCSI does not involve security (especially confidentiality protection) mechanisms. And iSCSI inherits this weakness. In consequence, network communication channel should be protected by security protocols in lower layers of TCP/IP stack. Usually IPSec is chosen as the solution to iSCSI system confidentiality protection problem (table 1).

Decision to use IPSec is just an initial step. Several alternatives related to location of crypto modules, operation mode (transport or tunnel mode) and nature of IPSec implementation (software or hardware) should be evaluated [1].

Table. 1. Common stack of protocols in network storage systems

Network layer	Protocol
Application	iSCSI
Transport	TCP
Network	IP+IPSec
Link	Ethernet

Servers to SAN (Storage Area Network) or SAN to SAN connections are usually secured with IPSec in tunnel mode. The mode encrypts datagrams completely, including the address fields. For transfer on an external network the encrypted datagram is included in another IP datagram, which contains no information about the real sender and recipient, but only firewall addresses. Thus a completely secure tunnel connection between host and remote storage site is created.

3. NEGATIVE ASPECTS OF IPSEC USAGE

IPSec is commonly used in IP networks, nevertheless it has significant, harmful impact on hosts as well as on network services. The impact is particularly noticeable in the case of software encryption implementations and in the case of mass data volume transmissions characteristic for network storage systems. Main quantitative, negative aspects of IPSec usage are as follows:

- network throughput deterioration,
- datagram delay increase,
- fluctuations of temporary throughput due to fluctuations of temporary central processing unit (CPU) load,
- significant CPU overload during massive data encryption/decryption.

Many research projects have been already completed in order to measure the impact. For example, Chaitanya et al. [6] demonstrated that on a representative IPSec implementation (Linux kernel 2.6), point-to-point throughput deterioration may reach up to 75%. In further research, Ferrante et al. [8] demonstrated the CPU usage during IPSec (with AES (Advanced Encryption Standard)) processing may reach up to 100%. Such CPU load is unacceptable in many cases.

Furthermore, new (not related to quantitative factors) problems of IPSec usage are common. Widespread problem, existing in many Internet services is incompatibility between IPSec and NAT (Network Address Translation) systems. Router performing NAT changes the IP addresses and port numbers of the datagram. It should recalculate the checksum of the TCP or UDP header. However with ESP (Encapsulated Security Payload) encryption, the NAT device cannot update the checksum since the TCP or UDP headers are encrypted with ESP.

IPSec – NAT incompatibility is well known issue, nevertheless IPSec is incompatible with some other network tools. An example is PEP (Performance Enhancing Proxy) [5].

Next problem is an effect of the additional overhead associated with adding new protocol headers (Authentication Header, Encapsulation Security Payload Header) to each protected datagram. This increase (at least 44 bytes) in datagram size[1] may cause that the maximum transmission unit (MTU) for a given network will be exceeded by inflated datagrams. Improperly configured routers may simply refuse to fragment or forward such datagrams. The same problem of exceeding MTU value may occur during IKE (Internet Key Exchange) phase of IPSec transmission. Large certificates (chains of certificates) or large certificate requests transmitted with a use of UDP may be bigger than MTU value. Blocking the fragmented UDP datagrams can lead to IKE failures that are very difficult to identify.

Mentioned above problems are solved with a use of some different methods evaluated in the next part of the chapter.

4. OVERCOMING IPSEC DRAWBACKS

4.1. PERFORMANCE PROBLEM SOLUTIONS

In most cases software performance problems are solved by replacing software with hardware implementations of the algorithm and/or by optimizing the algorithm. The rule is true in this case. In order to improve IPSec performance following methods were presented and evaluated:

- hardware cryptography accelerators,
- double implementation of IPSec with packet scheduling,
- lazy crypto approach.

Hardware implementations may use several technologies:

- NSP (Network Security Processor),
- ASIC (Application Specific Integrated Circuit),
- FPGA (Field Programmable Gate Array).

The advantage of hardware implementation is obvious. It has been shown [4], that for symmetric-key algorithms, performance increase of 4000% over software implementation on modern CPU may be achieved, simultaneously decreasing overhead of the general purpose processor.

---

[1] It must be noted that, in order to achieve relatively high throughput of network storage, datagrams should have as large as possible size.

There are several configurations of cryptography accelerators working for transmission purposes:

- Off-line,
- In-line,
- With single coprocessor,
- With mixed software+hardware implementations,
- With multiple coprocessors.

First, off-line configuration (known also as look-aside) means that IPSec encryption accelerator is placed outside the main data path in network interface card. The accelerator acts as co-processor to ordinary network processor. All the functions of managing full IPSec protocol support, including packet processing, link layer adaptations and security association handling, are performed by the network processor. The approach places significant load on the network processor and requires a high bandwidth bus for data transfer between the network processor and off-line IPSec accelerator [13].

The second, in-line architecture (known also flow-through architecture) is more efficient. It is assumed that in this case each byte of data to be transmitted flows through the accelerator. So host machine processing units may be completely unaware of IPSec existence. Several products (single chips handling the entire IPSec protocols) working this way are available commercially (e.g. family of Hifn products with Hifn HiPP III 4300/4450 Storage Security Processor as an example[2], Cavium Nitrox 3570[3], Intel 82576 Gigabit Ethernet Controller[4]). The throughput of such security processors reaches tens of gigabits per second.

An interesting mode of operation has been proposed by Amiri [2]. Amiri suggests twofold IPSec implementation, in which encryption system uses two parallel implementations of IPSec (one software and one hardware). The choice which implementation should be used in a given case is done by special scheduling algorithm.

Encryption may be performed by single coprocessor or by many coprocessors. In multiple coprocessor architectures a problem of optimal distributing datagrams among set of coprocessors should be solved. Scheduling algorithms for such systems have been proposed e.g. by Castanier [7] and Taddeo [15]. The scheduling algorithms utilize IPSec processing features: approximate hardware processing time of symmetric cryptography algorithm is known in advance and no data dependency exists between subsequent datagrams – according to IPSec specification each datagram is encrypted on its own and each datagram must carry any data required for its decryption.

Several shortcomings of the accelerators should be considered. First of all, fast Gigabit ASIC IPSec implementations are very expensive and may multiply total cost

[2] http://www.htmldatasheet.com/hifn/4350.htm
[3] http://www.cavium.com/processor_security.html
[4] http://www.intel.com/content/www/us/en/ethernet-controllers/82576-gbe-controller-brief.html

of network interface card (NIC) – vendors may not integrate accelerators with NIC in order to keep the prices low. Second problem, reported in some accelerators and some operating systems, is related to software drivers. Cryptographic libraries cannot access the accelerators without a special driver. For example, until recently there was no generic driver to access crypto libraries in the Linux kernel.

Another problem that should be mentioned here is the lack of flexibility, particularly in the case of accelerators built in NSP and ASIC technology. In general, the chips are not reprogrammable so it is not possible to substitute broken encryption algorithms – IPSec specification assumes that broken algorithms are replaced by new ones. Achieving simultaneously some level of flexibility and performance at level higher than in software is possible with a use of modern FPGA platforms [13]. A new feature of FPGA platforms, is partial reconfiguration mode of operation in which part of the chip is reconfigured while the remaining part is operational. But, due to lower internal clock speeds, throughput of FPGA chips is much smaller than in ASIC chips [14], [16].

Classic configuration of network storage is based on two separate encryption systems. IPSec is a solution to confidentiality problem during transmission. Another encryption system is necessary to protect data stored on remote media. A question is if two independent cryptography systems are always necessary [3]. Chaitanya et. al [6] propose a solution to IPSec performance problem based on modified IPSec protocol. In order to improve performance data encryption/decryption process at storage site (medium) is eliminated without decreasing confidentiality level. Data at rest are protected with a use of the same IPSec encryption that was used for communication confidentiality. Data sent to remote disk are encrypted before transmission to the disk at the client site with a use of modified IPSec procedure. Encrypted data are transferred and written (without decryption at the receiver side of the communication channel) to remote disk. So, data at rest are protected by IPSec encryption. During read process encrypted data from disk are read and transferred to client. IPSec decryption is performed by client after receiving data from remote storage site. The method is called "lazy crypto".

4.2. NAT PROBLEM SOLUTIONS

NAT problems are common. Standard techniques for solving the problems are NAT Traversal systems such as: SOCKS, Universal Plug and Play (UPnP), Session Traversal Utilities for NAT (STUN), UDP hole punching.

In the case of IPSec NAT-T (NAT Traversal) system is frequently used to traverse NAT [11]. NAT-T is based on additional encapsulation of datagrams. An original IPSec protected datagram is encapsulated in extra IP datagram and UDP segment (UDP with source and destination port numbers equal to 4500). Additional, non-encrypted UDP header and special marker (Non IKE marker) are inserted between the

ESP portion of the datagram and the original IP (copied and subsequently changed) header (fig. 1). In consequence any NAT device located in communication channel can access and change the IP address and port number information in additional, plain UDP header.

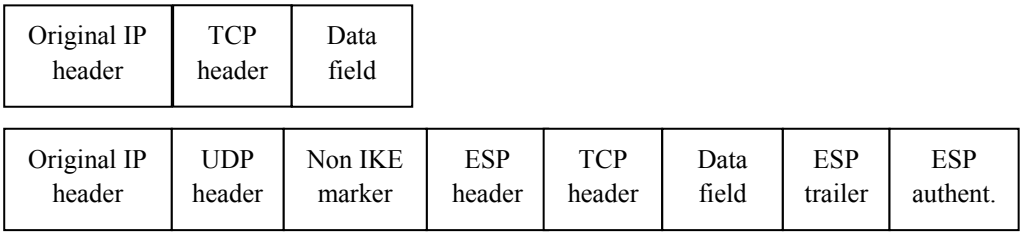

Fig. 1. IPSec datagram before and after NAT-T encapsulation

NAT-T is supported by main network devices and software producers. For example the feature is available in Microsoft operating systems starting from XP with SP2 [10]. Nevertheless Microsoft do not recommend NAT-T for Windows deployments with VPN servers. According to Microsoft tip [12]: "*When a server is behind a network address translator, and the server uses IPSec NAT-T, unintended side effects may occur because of the way that network address translators translate network traffic*".

NAT-T is also supported by Openswan[5], which is a complete IPsec implementation for Linux 2.0, 2.2, 2.4 and 2.6 kernels.

4.3. MTU PROBLEM SOLUTIONS

Common solution to archetypal MTU problem is datagram fragmentation performed at internetwork layer. Datagrams are fragmented and sent as a series of smaller packets. The same method is used here (for IKE MTU problem). At the initiator site the original IKE datagram is checked for size against minimum possible MTU (576 bytes). If the datagram is larger than 576 bytes, then it is split into a series of smaller datagrams. Each fragment is sent as an individual IKE datagram with IKE header. It is protected according to negotiation at the start of the IKE phase. The problem with the technique is the method is not universally supported by all systems (e.g. it is supported by Cisco but only Cisco IOS Software Release 12.4(15)T7 and later versions [9]).

NAT and MTU problems may also be solved by replacing IPv4 with IPv6. After full IPv6 introduction, NAT will be removed and minimum possible MTU value will be enlarged. In IPv6 networks, all links must handle a datagram size of at least 1280 bytes.

[5] http://lists.openswan.org

CONCLUSION

Network storage security is distinguishing and complex problem. The problem may not be solved by standard tools and methods used in other network services. In order to protect mass volume of transmitted data IPSec protocol should be used. Unfortunately, due to features of network storage service, common IPSec implementations are not committed to protect remote storage services.

The chapter is a short, original survey of issues related to network storage systems with a special emphasis to data security. In the chapter we have shortly discussed distinguishing features of network storage systems. Negative aspects of IPSec usage have been presented. Part 4 of the chapter is dedicated to solutions of some problems: problems with performance, problems with NAT and MTU.

In the special case of network storage, in order to maintain high level of performance, IPSec should be implemented in hardware with dedicated scheduling algorithms and with extra solutions to such drawbacks as NAT and MTU problems. Analysis of many different configurations and modes of operation of IPSec in network storage systems becomes a large research area.

ACKNOWLEDGEMENT

The research project is scheduled for years 2010–2013 and supported by scientific grant from the polish Ministry of Science and Higher Education.

REFERENCES

[1] ABOBA B., TSENG J., WALKER J., RANGAN J., TRAVOSTINO F., *Securing Block Storage Protocols over IP*, RFC 3723, IETF, 2004.
[2] AMIRI R., *IPSec*, 2007 [Online]: http://www.slideshare.net/Franklin72/ipsec-protocol.
[3] BILSKI T., *Data storage and transmission convergence concept*, E. Kozan (ed.) Proceedings of the Fifth Asia Pacific Industrial Engineering and Management Systems Conference, Queensland University of Technology, 2004, 14.8.1–14.8.16.
[4] BIRMAN M., *Accelerating Cryptography in Hardware*. HOTCHIPS Symposium on High Performance Chips, 1998 [Online]: HTTP://WWW.HOTCHIPS.ORG/ARCHIVES/HC10/2-MON/HC10.S4/HC10.4.2.PDF.
[5] BORDER J., KOJO M., GRINER J., MONTENEGRO G., SHELBY Z., *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, RFC 3135, IETF, 2001.
[6] CHAITANYA S., BUTLER K., SIVASUBRAMANIAM A., MCDANIEL P., VILAYANNURET M., *Design, Implementation and Evaluation of Security in iSCSI Based Network Storage Systems*, Storage SS'06, ACM, October 30, 2006, 17–28. [Online]: http://www.patrickmcdaniel.org/pubs/storss06.pdf.
[7] CASTANIER A. FERRANTE A., PIURI V., *Packet Scheduling Algorithm for IPSec MultiAccelerator Based Systems*, Proceedings of the 15th IEEE International Conference on Application-Specific Systems, Architectures and Processors (ASAP '04).

[8] FERRANTE A., PIURI V., OWEN J., *IPSec Hardware Resource Requirements Evaluation*, Proceedings of 1st Euro-NGI Conference on Next Generation Internet Networks – Traffic Engineering, Rome 2005, 240–246.

[9] *Fragmentation of IKE Packets*, Cisco Systems Inc., 2008, [Online]: http://www.cisco.com/en/US/docs/ios/sec_secure_connectivity/configuration/guide/sec_fragment_ike_pack.pdf.

[10] *IPSec NAT-T is not recommended for Windows Server 2003 computers that are behind network address translators*, Microsoft, 2006, [Online]: http://support.microsoft.com/kb/885348/en-us.

[11] KIVINEN T., SWANDER B., HUTTUNEN A., VOLPE V., *Negotiation of NAT-Traversal in the IKE*, RFC 3947, IETF, 2005.

[12] *L2TP/IPsec NAT-T update for Windows XP and Windows 2000*, Microsoft, 2012, [Online]: http://support.microsoft.com/kb/818043/en-us.

[13] LU J., LOCKWOOD J., *IPSec Implementation on Xilinx Virtex-II Pro FPGA and Its Application*, Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), Workshop 3, Vol. 04.

[14] SARAVANAN P. et. al., *High-Throughput ASIC Implementation of Configurable Advanced Encryption Standard (AES) Processor*, IJCA Special Issue on "Network Security and Cryptography" NSC, 2011, [Online]: http://research.ijcaonline.org/nsc/number3/SPE028T.pdf.

[15] TADDEO, A. V., FERRANTE A., *Scheduling Small Packets in IPSec Multiaccelerator Based Systems*, in Journal of Communications (JCM). Academy Publisher, Mar. 2007, Vol. 2, No. 2, 53–60.

[16] WANG H., BAI G., CHEN H., *A Gbps IPSec SSL Security Processor Design and Implementation in an FPGA Prototyping Platform*, Journal of Signal Processing Systems, Springer Verlag, Vol. 58, 2010, 311–324.

peer-to-peer networks, replication of documents,
erasure codes, asymptotic, inverse series

Karol MARCHWICKI*

ON ERASURE CODING AND REPLICATION IN PEER-TO-PEER SYSTEMS

Erasure coding and replication are two popular methods providing high availability in distributed storage systems. There are many papers with comprehensive studies on both techniques. Some of them compare those methods in terms of the storage overhead needed to achieve given level of availability and measured by so called effective redundancy factor, another evaluate the bandwidth required to restore partially lost redundant data. In our work we use a slightly different approach. We derive the analytical formula for the expected value of the number of node departures until the first moment of data loss, assuming that the likelihood of a data loss in the system is greater or equal than some fixed value. We show the resistance of a DHT-based network to unexpected node failures for erasure coding approach and for replication.

1. INTRODUCTION

The methods of redundancy in distributed storage systems are used to provide data access to the resources stored at nodes which are not always available and may sometimes temporarily or permanently fail. To protect against losing some portion of information and to maximise the data availability, such information has to be reproduced to other places. There are two main methods which are used for this purpose - erasure coding and replication. Many papers describe those two techniques. Some of them compare those methods in terms of the storage overhead needed to achieve given level of availability and measured by so called effective redundancy factor, another evaluate the bandwidth required to restore partially lost redundant data. In this paper we try to answer what is the resistance of the system in terms of the number of nodes that need to unexpectedly departure from the system to lose the first portion of infor-

* Institute of Mathematics and Computer Science, Wrocław University of Technology, Poland, ul. Janiszewskiego 14 a.

mation. We do not describe the implementation of redundancy methods. We evaluate the analytical formula for the expected number of node departures until the first moment of data loss. Finally we compare those results with the numerical experiment. We first consider erasure coding approach and then we compare those results with previous ones for replication [3].

Comparison of erasure codes and replication in terms of the mean time to failure has been presented in [15]. Authors analyse bandwidth and storage needed to provide given durability of a system. They indicate that erasure coding guarantees longer node lifetime, less bandwidth and less storage overhead than the whole file replication. In [8] authors present a replication algorithm which takes into account system durability and node availability. They also indicate the problem of the temporary failure which could be critical to system efficiency when considered in the same way as the permanent failure. In [11] authors point out to situations in which the whole file replication is preferred. They use asymptotic analysis to compute the optimal value of the number of blocks a file is divided into, to achieve the highest possible file availability. They also analyse the cost of reassembling file in erasure coding approach. In [13] authors argue that the whole file replication is in many cases a better solution than the erasure coding. They state that gains from coding are dependent on the characteristics of the nodes and that advantages of using this method are limited. Authors also consider the bandwidth needed to reconstruct data. In [4] authors state the question how to generate encoded fragments in a distributed way providing the lowest possible bandwidth during transmission. They point out that the dynamics of a network plays an important role when choosing the right strategy and they identify the trade-off between the storage and the bandwidth. In [1] authors analyse two main factors in redundancy schemes - file availability and replication factor. They compare two methods of redundancy by developing a probabilistic model for reasoning about the storage overhead with the given level of availability. Moreover they consider various host availability distributions and point out to the problem of using erasure coding schema when hosts failures are persistent. Finally they formulate the trade-off for implementing redundancy strategies in Peer-to-Peer system.

To the best of our knowledge no previous paper directly addresses the measurements of the number of nodes which need to be deleted to obtain the first data loss, what we do analyse in this work. The main advantage of this analysis is the fact that it can be helpful when determining the reasonable frequency of reconstructing partially lost fragments or files. To compare erasure codes with the whole file replication we use a redundancy strategy which spreads document copies at symmetric places in a DHT structure [3]. This approach can be called The Buddy Placement Policy or The Symmetric Replication Scheme (SRS). The decision whether to choose this approach is dictated by its resistance to unexpected node departures [2, 6, 7, 10, 12].

The paper is organised as follows. In Section 2 we present the basic notation and facts. Section 3 describes the mathematical model for the problem we analyse in section 4. Section 5 contains main theorems and results for erasure coding and its comparison with previous results for the whole file replication redundancy scheme. Section 6 presents simulation results and finally section 7 concludes our work.

2. NOTATION

In this section we describe basic notation and facts. We assume that node availabilities are independent and identically distributed. We also assume that the rate of erasure coding is constant (we consider only so called deterministic codes schemes). Moreover we assume that each node has the same replication factor. The probability that a single object is available is denoted by p, so the probability that the object is not available is equal to $1-p$. We also assume that each data item is stored at one cluster. Cluster capacity is denoted by a .The number of clusters is denoted by n and the number of nodes in the system is denoted by N. Detailed description of those parameters is described in the next section.

2.1. ERASURE CODES

We fix two parameters s and r .Each data item is divided into s fragments and then is encoded into $a = r + s$ fragments that can tolerate r failures. We call $r/(r+s)$ the *rate of encoding*. The ratio $(r+s)/s$ is called the *effective redundancy factor* and describes the storage overhead.

2.2. REPLICATION

We fix the parameter a which describes the cluster capacity and the number of replicas. The *effective redundancy factor* is equal to a.

3. MODEL DESCRIPTION

In this section we show the mathematical description of our model. We use the method of placing documents at symmetric places in a DHT structure which guarantees the highest possible resistance to the data loss. This solution was proposed and investigated in [6] and [7] where it was applied for the whole-file replication scheme.

It is called *Symmetric Replication Schema* (SRS).

Let us recall that the virtual space of a DHT-based Peer-To-Peer system such as Chord is the set $V = \{0,1\}^{160}$. Let us fix a hash function h with values in V and a number a.

3.1. ERASURE CODES

Document δ is divided into a data chunks which are stored at places

$$(h(\delta) + \lfloor (i/a) \cdot 2^{160} \rfloor) \mod 2^{160} : i = 0, \ldots, a-1 \tag{1}$$

3.2. REPLICATION

Similar idea of placing documents at symmetric places in a DHT ring structure can be applied to the whole file replication technique. In this case instead of data chunks we put the whole file replicas and their placement is described by equation (1).

4. FIRST MOMENT OF DATA LOSS

We assume that the node failures i.e. the unexpected node departures occur at random. Let $N = |\Omega|$ denotes the number of nodes and let us suppose that $a \mid N$. Let $n = N/a$. Let $\{U_i, i = 1, \ldots, n\}$ be a fixed partition of the set Ω into disjoint clusters of cardinality a.

4.1. ERASURE CODES

Let Y_j for $j = 1, ..., n$ denotes the random variable indicating whether we have or not object available at fixed $j-$th cluster and let Y denotes the random variable indicating whether we have or not objects available in all clusters. The probability that the object stored at fixed cluster of size a is available (see [1]) is given by $\sum_{i=s}^{a} \binom{a}{i} p^i (1-p)^{a-i}$ what can be expressed as $1 - \sum_{i=0}^{s-1} \binom{a}{i} p^{a-i} (1-p)^i$. Let us denote $\sum_{i=0}^{s-1} \binom{a}{i} p^{a-i} (1-p)^i$ by $f_{a,s}(p)$. Then $\Pr[Y = 1] = (1 - f_{a,s}(p))^n$. Let $Z_{N,a}$ denotes the random variable which describe the number of collisions. The random

ble $Z_{N,a}$ has a Binomial distribution with parameters $N = na$ and p. We have $E[Z_{N,a}] = N \cdot p$. Values of n and a are known and fixed, so to estimate the expected value of the random variable $Z_{N,a}$ we need to find a threshold value for p. To find a threshold for p we can bound the availability of all objects at all clusters by some fixed probability c and solve with respect to p the following inequality

$$(1 - f_{a,s}(p))^n \geq c \tag{2}$$

To estimate the first moment of data loss measured in the number of nodes that need to be deleted to lose the first portion of information we can fix a probability of losing all objects and apply it to $\mathrm{E}[Z_{N,a}]$. The average case would be for $c = 1/2$.

4.2. REPLICATION

The random variable which describes the first moment of data loss equals $K_a^{(n)} = \min\{k : (\exists i)(U_i \subseteq \{\omega_1, \ldots, \omega_k\})\}$. For further information see [3].

5. ANALYTICAL FORMULA

In this section we derive the analytical formula for the expected value of the number of node departures until the first moment of data loss, assuming that the likelihood of a data loss in the system is greater or equal than some fixed value. The main goal of the first subsection is to solve the inequality (2) from previous section with respect to p. The expected value of the first moment of data loss for the whole file replication approach was analysed in [3] and we recall the results from this work.

5.1. ERASURE CODES

We would like to solve the inequality (2) with respect to p. We have

$$1 - (c)^{\frac{1}{n}} \geq \sum_{i=0}^{s-1} \binom{a}{i} p^{a-i} (1-p)^i \tag{3}$$

Let us denote the left hand side of this equation by y. To find a threshold for p we can solve the corresponding equation (instead of inequality (3))

$$y = f_{a,s}(p) \tag{4}$$

In this equation variable p is implicit into a power series with a finite number of terms. We need to represent p as a function dependent on y. It can be done by reversing the series so that we would have $p = g_{(a,s)}(y)$ where $g_{(a,s)}(y)$ is the function we are looking for, describing the inverse series. The series is in the form $y = c_{r+1}p^{r+1} + c_{r+2}p^{r+2} + \ldots$, $p = \left(\frac{1}{c_{r+1}} \cdot y - \frac{c_{r+2}}{c_{r+1}} p^{r+2} - \ldots \right)^{\frac{1}{r+1}}$ so the inverse series should be in the form $p = a_1 \cdot y^{\frac{1}{r+1}} + a_2 y^{\frac{2}{r+1}} + \ldots$. We will first solve a modified equation in which the inverse series has positive natural powers. Denoting $x = y^{\frac{1}{r+1}}$ we have

$$x = \left(\sum_{i=0}^{s-1} \binom{a}{i} p^{a-i} (1-p)^i \right)^{\frac{1}{r+1}} \tag{5}$$

To solve this equation we will use The Lagrange Inversion Formula ([5, 14]).

THEOREM 1(Lagrange Inversion Formula – LIF) Let us assume that ϕ is analytic in 0 and that $\phi(0) \neq 0$. Then the equation $y(t) = t \cdot \phi(y(t))$ has an analytic solution around 0 and

$$[z^k]y(z) = \frac{1}{k}[u^{k-1}](\phi(u))^k$$

To solve the inverse series we would use the following corollary which is a consequence of applying LIF to the function $\frac{p}{f(p)}$.

COROLLARY1. If the function is in the form $\phi(p) = \frac{p}{f(p)}$ and $\phi(p)$ fulfils the assumptions of LIF then

$$[x^k]f^{-1}(x) = \frac{1}{k}[p^{k-1}]\left(\frac{p}{f(p)} \right)^k$$

We have $\phi(p) = \dfrac{p}{(f_{a,s}(p))^{\frac{1}{a-s+1}}}$ and $lim_{p\to 0}\phi(p) \neq 0$. According to the LIF the coefficients of the inverse series will be in the form

$$[x^k]f^{-1}(x) = \frac{1}{k}[p^{k-1}]\left(\frac{(f_{a,s}(p))^{\frac{1}{a-s+1}}}{p}\right)^{-k} = \frac{1}{k}[p^{k-1}]\left(\frac{f_{a,s}(p)}{p^{a-s+1}}\right)^{\frac{-k}{a-s+1}} \tag{6}$$

To raise the series to a given power we can use the recursive formula from [9] (chapter 4.7 p.526, formula (9)). To use this formula we need to norm our series so that the first term is equal to 1. First two coefficients of the inverse series are

$$\binom{a}{s-1}^{\frac{-1}{a-s+1}}, \binom{a}{s-1}^{\frac{-2}{a-s+1}} \frac{s-1}{a-s+2} \tag{7}$$

The final result for p is then

$$p \leq \binom{a}{s-1}^{\frac{-1}{a-s+1}} y^{\frac{1}{a-s+1}} + \binom{a}{s-1}^{\frac{-2}{a-s+1}} \frac{s-1}{a-s+2} y^{\frac{2}{a-s+1}} + O\left(y^{\frac{3}{a-s+1}}\right) \tag{8}$$

where $y = 1-(c)^{\frac{1}{n}} \approx \left(-\dfrac{\ln c}{n}\right)$. From this we obtain the bound for $E[Z_{N,a}]$

$$N\cdot\binom{a}{r+1}^{\frac{-1}{r+1}} y^{\frac{1}{r+1}} + N\binom{a}{r+1}^{\frac{-2}{r+1}} \frac{s-1}{a-s+2} y^{\frac{2}{r+1}} + O\left(N\cdot y^{\frac{3}{r+1}}\right) \tag{9}$$

5.2. REPLICATION

The formula for the replication was analysed in [3]. The final result describing the first moment of data loss can be summarised by the following asymptotic

$$E[K_{N,a}] = a^{\frac{1}{a}}\Gamma\left(1+\frac{1}{a}\right)N^{1-\frac{1}{a}} + O\left(N^{-\frac{1}{a}}\right) \tag{10}$$

N denotes the number of nodes, a is the replication factor and $K_{N,a}$ is a random variable describing the first moment of data loss.

6. EXPERIMENTAL RESULTS

In this section we present results from the simulation. We first show the evaluation for erasure codes model and then for the corresponding whole-file replication model. We assume in our model that a divides N . In the DHT structure the number of nodes N may not be divisible by a and the nodes may not divide their virtual space into segments of equal size. This may not directly correspond to our model (see [3]).

6.1. ERASURE CODES

Numerical experiments show that the first moment of collision in the erasure codes model can be upper bounded by the function $E[Z_{N,a}]$ when applying $c = 3/4$ and lower bounded by the same function when applying $c = 1/4$. The average value of the first moment of collision oscillates around $E[Z_{N,a}]$ for $c = 1/2$ (see figure 1) when the number of experiments is large enough (i.e. ≈ 100).

6.2. REPLICATION

Numerical experiments show that the function $E[K_{N,a}]$ is the upper bound for the first moment of collision in the whole file replication model and that $1/2E[K_{N,a}]$ is the corresponding lower bound. The average value of the first moment of collision oscillates around $3/4E[K_{N,a}]$ (see figure 2) when the number of experiments is large enough (i.e. ≈ 100).

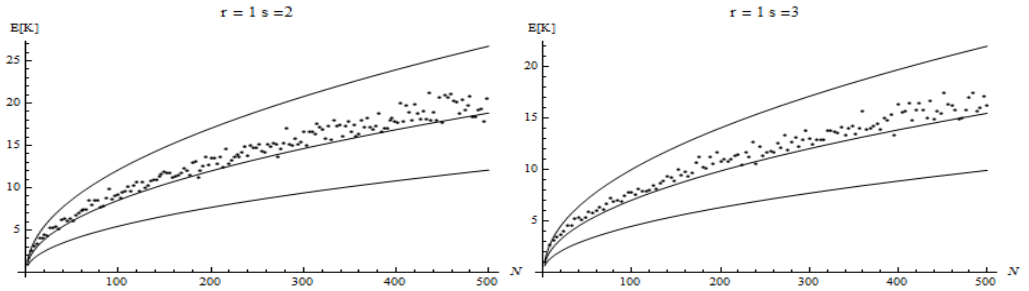

Fig. 1. Erasure codes redundancy scheme with (r, s) equal to(1,2) and (1,3). Expected number of nodes that need to be deleted (denoted by $E[K]$) to lose the first portion of information for selected initial number of nodes (denoted by N). Black dots which represent numerical results are situated between $c = 3/4$ and $c = 1/4$ applied to the function $E[Z_{N,a}]$ known from our theoretical considerations (solid lines).

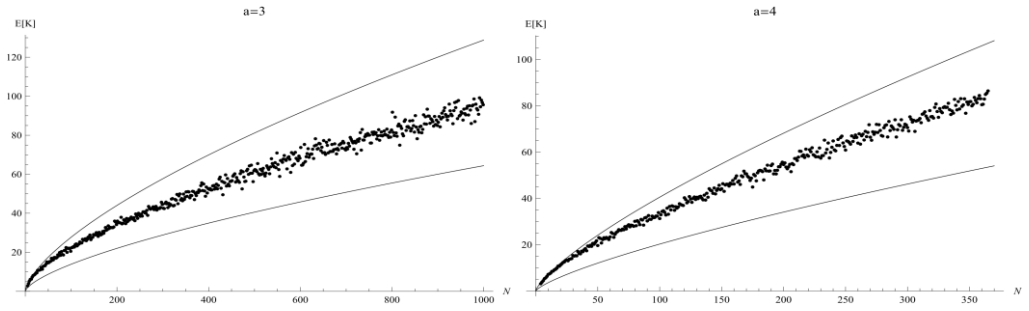

Fig. 2. Replication redundancy scheme with $a = 3/4$ replicas. Expected number of nodes that need to be deleted (denoted by $E[K]$) to lose the first portion of information for selected initial number of nodes (denoted by N). Black dots which represent numerical results are situated between $E[K_{N,a}]$ and $1/2E[K_{N,a}]$ which are known from the previous section (solid lines).

6.3. COMPARISON

To compare the erasure codes model with the whole file replication model we need to analyse the storage overhead measured by the effective redundancy factor. Let us recall that in the erasure codes model this factor is equal to $r + s / s$ and in the whole file replication model it is equal to the number of replicas a. Let us take the whole file replication model with a replicas. When we fix the value s, the same redundancy factor can be achieved by using erasure codes with s and $r = (a-1)s$ parameters. To compare the results we can take as an example three different models: (a) 2 replication, (b) (2,2) erasure codes and (c) (3,3) erasure codes. The best resistance to unex-

pected node departures is in case when the value of $E[K]$ is the largest one. It corresponds with the largest number of nodes to be deleted to have the first information loss. With parameters provided above the most resistant is (3,3) erasure codes model. One may notice that in (3,3) case the number of data chunks s is greater than in (2,2) case. Of course the large number of data chunks requires more bandwidth to restore partially lost data but it may be worthy to consider large values (see [11]). The results of the experiment are summarised in the figure 3.

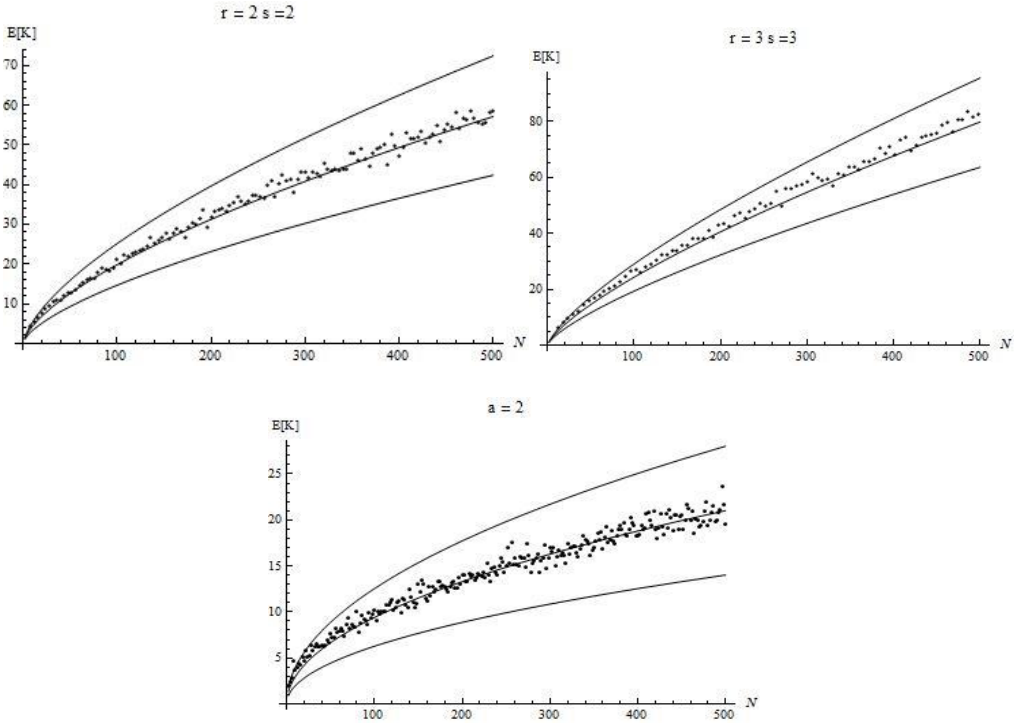

Fig. 3. Three different models: (a) (2,2) erasure codes, (b)(3,3) erasure codes and (c) 2 replication. Expected number of nodes that need to be deleted (denoted by E[K]) to lose the first portion of information for selected initial number of nodes (denoted by *N*). Black dots which represent numerical results are situated between previously calculated bounds (solid lines) known from our theoretical considerations

7. CONCLUSIONS

Results from our work show that the expected value of the first moment of collision in the system based on symmetric replication scheme in erasure codes model can

be upper bounded by c = 1/4 and lower bounded by c = 3/4 applied to the following function

$$E[Z_{N,a}] = N \cdot \binom{a}{r+1}^{\frac{-1}{r+1}} y^{\frac{1}{r+1}} + O\left(N \cdot y^{\frac{2}{r+1}}\right) \tag{11}$$

where N denotes the number of nodes, s is the number of fragments a file is divided into, a is the cluster capacity and c is the fixed probability of a data loss. The formula describes the expected value of the number of nodes that need to be deleted to have some fixed probability of information loss.

Similar result, although based on slightly different analysis was formulated in [3] for the whole-file replication model and it tells that the expected value of the first collision (i.e. data loss) in a DHT-based system can be upper bounded by $E[K_{N,a}]$ and lower bounded by $1/2E[K_{N,a}]$ where

$$E[K_{N,a}] = a^{\frac{1}{a}}\Gamma\left(1+\frac{1}{a}\right)N^{1-\frac{1}{a}} + O\left(N^{-\frac{1}{a}}\right) \tag{12}$$

N denotes the number of nodes and a is the replication factor.

Our considerations correspond to the Buddy Placement Policy. Similar model is presented in [2], however it does not treat about the first moment of data loss. Our solution can be applicable to all Peer-to-Peer systems which are based on DHT and which use The Buddy Placement Policy. The main advantage of this approach is the fact that it can be helpful to determine the reasonable frequency of reassembling data items. We plan to extend our results to such analysis. Moreover, the hypotheses that for every $a \geq 2$ the expected value of the first moment of data loss is close to $3/4E[K_{N,a}]$ in the whole file replication model and close to $E[Z_{N,a}]$ for $c = 1/2$ in the erasure codes model need detailed explanation.

ACKNOWLEDGEMENT

Work supported by grant nr 2011/B10041 of the Institute of Mathematics and Computers Science of the Wrocław University of Technology.

REFERENCES

[1] BHAGAWAN R., MOORE D., SAVAGE S., VOELKER G.M., *Replication strategies for highly available peer-to-peer storage,* In: Future directions in distributed computing , Schiper A., Shvartsman A.A., Weatherspoon H., Zhao B.Y. (Eds.), Berlin, Heidelberg, Springer-Verlag, 2003, 153–158.

[2] CARON S., GIROIRE F., MAZAURIC D., MONTEIRO J., PERENNES S., *Data life time for different placement policies in p2p storage systems*, Springer-Verlag LNCS, 6265:75–88, 2010.

[3] CICHOŃ J., KAPELKO R., MARCHWICKI K., *Brief announcement: A note on replication of documents*, Springer-Verlag LNCS, 6976:439–440, 2011.

[4] DIMAKIS R.G., GODFREY P.B., WU Y., WAINWRIGHT M.O., RAMCH K., *Network coding for distributed storage system*, In Proc. of IEEE INFOCOM, 2011.

[5] FLAJOLET.P, SEDGEWICK R., *Analytic Combinatorics*, Cambridge University Press, 2009.

[6] GHODSI A., *Distributed k-ary System:Algorithms for Distributed Hash Tables*, PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2006.

[7] GHODSI A., ALIMA S., HARIDI S., *Symetric replication for structured peer-to-peer systems*, DBISP2P, 2005.

[8] GON CHUN B., DABEK F., HAEBERLEN A., SIT E., WEATHERSPOON H.,KAASHOEK M.F., KUBIATOWICZ J. MORRIS R., *Efficient replica maintenance for distributed storage systems*, In: Proc. of NSDI, 2011, 45–58.

[9] KNUTH D.E., *The art of computer programming.* Vol. 2: *Seminumerical Algorithms*, Addison-Wesley, 1997.

[10] KTARI S., ZOUBERT M., HECKER.A.,LABIOD H., *Symetric replication for efficient flooding in dhts*, ACM, 978-1-60558-073-9/08/05, 441-442, 2011.

[11] LIN W.K., CHIU D.M., LEE Y.B., *Erasure code replication revisted*, Peer-To-Peer Computing, IEEE, 90–97, 2004.

[12] PITOURA., NTARMOS N., TIRANTAFILLOU P., *Replication, Load balancing and efficient range query processing in dhts*, Springer-Verlag LNCS, 3896:131–148, 2006.

[13] RODRIGUES R., LISKOV B., *High availability in dhts: Erasure coding vs. replication*, In Peer-To-Peer Systems 4th International Workshop IPTPS 2005, Ithaca, New York, 2005.

[14] SZPANKOWSKI W., *Average Case Analysis of Algorithms on Sequences*, Wiley, 2001.

[15] WEATHERSPOON H., KUBIATOWICZ J., *Erasure coding vs. Replication: A quantitive comparison*, In: Proc. of the First International Workshop on Peer-to-Peer Systems IPTPS 2002, Cambridge, Massachusetts, 2002.

switching networks, multi-service systems
traffic overflow, traffic engineering

Mariusz GŁĄBOWSKI*
Michał Dominik STASIAK*

SWITCHING NETWORKS WITH OVERFLOW LINKS AND POINT-TO-POINT SELECTION

This article presents a method for modeling of multi-service switching networks with point-to-point selection and a system of overflow links. The concept of effective availability forms the basis for the adopted method for modeling. A particular attention in the article is also given to the way this parameter is determined for switching networks with overflow links. The results of the analytical calculations are compared with the results of the simulations for selected multi-service switching networks with overflow links and point-to-point selection. The study confirms high accuracy of the proposed method as well as the suitability of the application of the system of overflow links.

1. INTRODUCTION

Parameters of modern communications networks depend on the effectiveness of switching devices in network nodes. The basis for the operation of these devices is formed by switching networks. One can distinguish blocking networks and non-blocking networks [1-3]. On account of costs involved, blocking networks are used in most devices. The application of such networks, however, is followed by a loss of part of traffic due to the occurrence of the internal blocking. There is a number of ways to reduce this phenomenon, such as the application of overflow links [4], call repacking, dynamic routing and rearrangement [2, 3], among others. Techniques based on repacking and rearrangements do not interfere with the physical structure of the network. The decrease in the internal blocking results from the application of complex algorithms for setting up connection. A major shortcoming of this approach, however, is the increased load in control devices. Any solution based on the application of overflow links is accompanied with a change in the structure of the network following the intro-

* Chair of Communications and Computer Networks, Poznań University of Technology.

duction of additional links between switches of a given stage (section). For the first time, overflow links were introduced in Pentaconta switching system used from 1960 to the 1980s [4]. In Pentaconta switching networks, overflow links were applied in the first stage. As a result, a several percent decrease in the internal blocking probability was possible to be obtained [5]. The possibility of the introduction of overflow links has been also considered in digital switching networks [6–8].

Modern network devices utilize networks that carry multi-service traffic. The simulation experiments that have been hitherto conducted [9, 10] indicate that the application of overflow links in these links results in a substantial increase in their traffic effectiveness effected by the reduction of the internal blocking. [11] proposes a modification of the PGBMT method (Point-to-Group Blocking with Multi-rate Traffic [12]) to model multi-service networks with overflow links and point-to-point selection. In [13], a method for modeling networks with overflow links that are offered Engset traffic is presented.

The present article attempts to define point-to-point blocking in the multi-service switching network with a system of overflow links. The model assumes that the capacity of overflow links is very high. Such an approach results from the simulation studies presented in [9–-11] in which it was observed that a two-fold increase in the capacity of overflow links – as compared to the capacity of inter-stage links – led to a virtual elimination of the phenomenon of internal blocking. The article is structured as follows: Section 2 describes the structure and the operation of a three-stage Clos network with overflow links. Section 3 proposes a model of the limited-availability group that serves as basis for calculations of the external blocking probability in the switching network. Section 4 discusses the PPD method (Point-to-point blocking Direct method)) [14-16] that forms a basis for the adopted method to model switching networks. Section 5 presents methods for determining effective availability in networks with overflow links. Section 6 discusses the results of the calculations and the simulations for selected switching networks. Finally, Section 6 sums up the article.

2. SWITCHING NETWORK WITH OVERFLOW LINKS

Figure 1 shows the Clos switching network. The network structure is controlled by k symmetrical switches of k x k links in each of its stages. Each link has a capacity of f BBU (Basic Bandwidth Unit [17]). The network is offered multrate traffic that is composed of several different classes of call streams. A call of each class requires a different number of BBUs to set up a connection. The call stream of each class is a Poisson stream. The adoption is that the network employs point-to-point selection. The outputs of the switching networks are grouped into directions in such a way that each i-th output of each of the switches of the last section belongs to the i-th output direction.

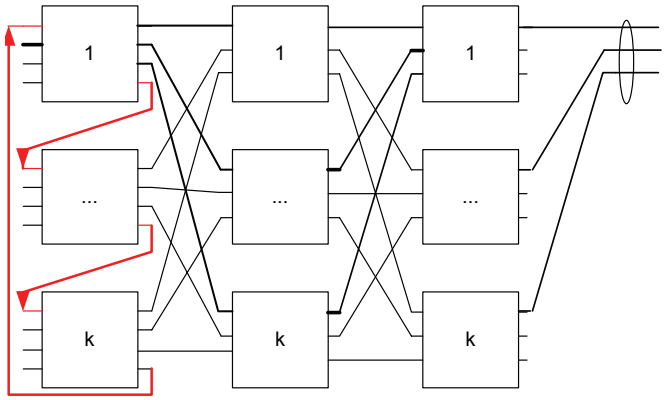

Fig. 1. Clos switching network with a system of overflow links in the first stage

In point-to-point selection, in the case of a network without overflow links, when a call arrives at the input of the switch of the first stage, the control algorithm chooses the switch of the last stage that has an available free link in the demanded direction. Then, the control algorithm tries to set up a connection between the selected switches of the first and the last stages. If it is not possible, the algorithm rejects the call due to the internal blocking. If all the links of the output link of a given direction are busy, the control algorithm rejects the call due to the external blocking.

In a three-stage Clos network, overflow links can be introduced in a number of ways [9]. In this article it is adopted that the overflow links system is applied to the first stage in such a way that the overflow links connect the additional output of a given switch with the additional input of a neighboring switch of the same stage (Fig. 1). The output of the last switch is connected with the additional input of the first switch. The results of the simulation experiments [9-10] show unequivocally that the highest decrease in the internal blocking is the result of the application of the overflow system in the first stage of the switching network. In addition, the application of overflow links with the capacity that is two-fold higher than the capacity of inter-stage links stabilizes the internal blocking at the level of insignificant values (at least, by the order of lower values). The network becomes thus a quasi-non-blocking network.

3. LIMITED-AVAILABILITY GROUP

The limited-availability group (LAG) [18], [19] is a set of separated links to which multirate traffic is offered. The group can model k output links of the switching network, each with the capacity of f BBUs, that belong to one direction. The occupancy distribution in LAG can be determined on the basis of the following recurrence formula:

$$n[P_n]_V = \sum_{i=1}^{M} A_i\, t_i\, \zeta_i(n-t_i)\left[P_{n-t_i}\right]_V, \tag{1}$$

where:

- $-\left[P_n\right]_V$ – occupancy probability of n BBUs in a group with the capacity V BBU ($V = kf$),
- $-M$ – the number of traffic classes offered to the output group,
- $-A_i$ – traffic intensity of traffic of class i offered to the output group,
- $-t_i$ – the number of BBUs necessary to set up a connection of class i,
- $-\varsigma_i(n)$ – the conditional stage passage probability for stream of class i.

In the LAG model, the conditional stage passage probability is determined on the basis of the following combinatorial formula:

$$\varsigma_i(n) = \frac{F(V-n,k,f,0) - F(V-n,k,t_i-1,0)}{F(V-n,k,f,0)}, \tag{2}$$

where $F(x, k, f, t)$ is the number of arrangements $x=V-n$ of free BBUs in k separated links, each with the capacity f BBUs, with the assumption that initially t free BBUs have been arranged in each link:

$$F(x,k,f,t) = \sum_{i=0}^{\left\lfloor \frac{x-kt}{f-t+1} \right\rfloor} (-1)^i \binom{k}{i} \binom{x-k(t-1)-1-i(f-t+1)}{k-1}, \tag{3}$$

Having determined the occupancy distribution in LAG, it is possible to determine the blocking probability for the traffic stream of class i in LAG on the basis of the following formula:

$$E_{\mathrm{WOD}}(i) = \sum_{n=k(f-t_i+1)}^{V} \left[P_n\right]_V \left[1-\zeta_i(n)\right]. \tag{4}$$

4. THE PPD METHOD

One of the possibilities to evaluate the blocking probability in a multi-service switching network with point-to-point selection is to apply the PPD method (Point-to-point blocking Direct method) [14-16]. The idea behind the method is based on a direct application of the value of the effective availability parameter as the measure for the internal blocking probability [14].

In accordance with the PPD method, the blocking probability in a 3-stage multi-service switching network can be determined in the PPBMT method by the following equation:

$$E_c(i) = E_z(i) + E_w(i)[1 - E_z(i)], \tag{5}$$

where $E_c(i)$ is the total blocking probability, whereas $E_z(i)$ and $E_w(i)$ are external and internal blocking probabilities for calls of class i:

$$E_z(i) = E_{\mathrm{WOD}}(i) = \sum_{n=k(f-t_i+1)}^{V} \left[P_n\right]_V \left[1 - \zeta_i(n)\right], \tag{6}$$

$$E_w(i) = \frac{V - d(i)}{V}. \tag{7}$$

In Formula (7), the parameter $d(i)$ is the effective availability for calls of class i. The effective availability determines the number of available switches of the third stage for one switch of the first stage, i.e., these switches of the third stage with which a call of class i can be set up from a given switch of the first stage. In line with the assumptions of the PPD method, to determine the effective availability we have to construct the so-called equivalent network, i.e., a single-service network with identical topological structure as the multi-service network, and with the capacity of inter-stage links equal to a single BBU. The method for a determination of the effective availability will be presented in the next charter.

5. EFFECTIVE AVAILABILITY OF THE SWITCHING NETWORK

According to [14-16], the effective availability of a three-stage network can be expressed with the following formula:

$$d(i) = [1 - \pi(i)]k + \pi(i)\eta Y_1(i) + \pi(i)[k - \eta Y_1(i)]\, e(i)\, \sigma_3(i), \tag{8}$$

where:

- $d(i)$ – effective availability for stream of class i in the equivalent network.
- $\pi(i)$ – probability of direct non-availability of the switch of the last stage for calls of class i, evaluation of this parameter is based on the channel graph of the equivalent switching network and can be calculated by the Lee method [20],

- k – total number of outgoing links in considered direction,
- η – such a portion of the average fictitious traffic of the switch of the first stage which is carried by the direction in question. If we assume that the traffic is uniformly distributed between k directions, we obtain:

$$\eta = 1 / k , \tag{9}$$

- $\sigma_3(i)$ – secondary availability coefficient, which is the probability of an event that the connection path of class i passes through directly available switches of intermediate stages [43],
- $e(i)$ – the fictitious load carried by one link of the equivalent switching network,
- $Y_1(i)$ – average fictitious traffic of class i carried by the switch of the first stage:

$$Y_1(i) = k\, e(i) . \tag{10}$$

The parameter $e(i)$ in Formula (8) is the load of the inter-stage (output) link of the equivalent switching network for traffic of class i. It is adopted [12] that this load is equal to the blocking probability for calls of class i in a real inter-stage link with the capacity of f BBU. The blocking probability and the occupancy distribution in the link with the capacity f can be determined on the basis of the Kaufman-Roberts distribution [21, 22]:

$$n[P_n]_f = \sum_{i=1}^{M} a_i t_i [P_{n-t_i}]_f , \tag{11}$$

$$e(i) = \sum_{n=f-t_i+1}^{f} P[n]_f , \tag{12}$$

where a_i is traffic offered to a single link of the switching network:

$$a_i = A_i / k . \tag{13}$$

The parameter $\pi(i)$ is the probability of direct unavailability that determines that a given switch of the second stage is not available for a given switch of the first stage. The parameter can be determined by Lee method [20] on the basis of the equivalent graph of the three-stage network.

The probability graph of the three-stage Clos network without overflow links is presented in Fig. 2a. This graph shows all connection paths between one switch of the first stage (Switch A) and one switch of the third stage (Switch X). The introduction of an inter-stage link between two switches of the first stage is followed by an increase in the number of connection paths in the graph (Fig. 2b), since the switches of the second stage become available for a given switch A of the first stage through inter-stage links outgoing from Switch A (i.e., links A_1, A_2, … ,A_k) and through the links outgoing from Switch B (i.e., links B_1, B_2, …, B_k), with which Switch A is connected by the overflow link AB. Assuming that the overflow link is lossless, the graph presented in Fig. 2b can be transformed to the form shown in Fig. 2c. Thus, on the basis of the graph presented in Fig. 2, the probability $\pi(i)$ can be determined by the following formulas:

– for network without overflow: $$\pi(i) = \left\{1-[1-e(i)]^2\right\}^k, \tag{14}$$

– for network with overflow: $$\pi(i) = \left\{1-[1-e^2(i)][1-e(i)]\right\}^k. \tag{15}$$

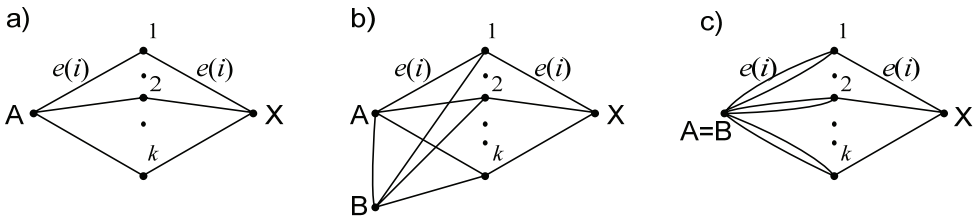

Fig. 2. Clos switching network graphs a) network without overflow links, b) network with overflow links c) network with lossless overflow links

The secondary availability coefficient for three-stage Clos networks (determined by the graphs in Fig. 2) can be defined as a ratio of directly available switches of the second stage $d_{b,2}(i)$ to all k switches of the second stage [12]:

$$\sigma_3(i) = \frac{d_{b,2}(i)}{k}. \tag{16}$$

The number of directly available switches of the second stage is determined on the basis of the probability of direct unavailability of the two-stage switching network:

$$d_{b,2}(i) = k\left[1 - \pi_2(i)\right]. \tag{17}$$

For two-stage switching networks constructed of two first stages of the considered three-stage Clos network, probability graphs for the system without overflow links and the system with lossless overflow links are presented in Fig. 3.

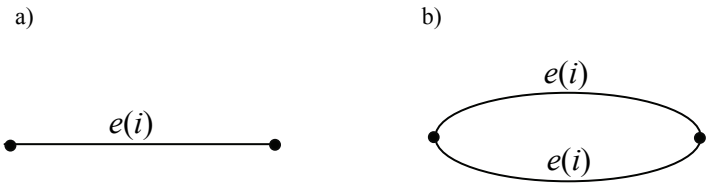

Fig. 3. A graph of the 2-stage network a) without overflow links, b) with lossless overflow links

On the basis of the graphs presented in Fig. 3 and Formulas (16) and (17), we can eventually obtain formulas for the evaluation of the secondary availability coefficient in the three-stage Clos switching network with and without the overflow links system:

– for network without overflow system: $\sigma_3(i) = 1 - e(i)$, (18)

– for network with overflow system: $\sigma_3(i) = 1 - e^2(i)$. (19)

The parameters $\pi(i)$ and $\sigma_3(i)$, determined on the basis of (14), (15), (18), (19), form a basis for a determination of the effective availability in the network with and without overflow links with the application of (8). By having the effective availability parameter for calls of individual traffic classes we are position to determine the internal blocking probability (7) and external blocking probability (6) in the switching network, with and without overflow links, that services multiservice traffic.

6. NUMERICAL RESULTS

Both simulation experiments and analytical calculations were conducted for the 3-stage Clos network. The network consists of 4 symmetrical switches 4x4 in each of

the stage. The capacity of the links in the network is 30 BBUs. The overflow links have been introduced to the first stage of the network. The network is offered multi-rate traffic that consists of 3 traffic classes that require 10 BBUs, 5 BBUs and 2 BBUs, respectively. Traffic of all classes is Erlang traffic and is offered in the following proportions: A_1: A_2: A_3 = 1:1:1. Point-to-point selection was used in the network to set up connections. Simulation experiments were conducted with the help of a dedicated digital simulator based on the event scheduling approach method [10]. In the simulation experiments, the determined 95% confidence interval was evaluated on the basis of the t-Student distribution for 10 series, each consisting of 100,000 calls of the oldest class in each of the series.

Figure 3 shows the results of the calculations and the simulations of the total blocking probability in the switching network with overflow links. The results are presented in relation to the average traffic offered to one BBU unit. Figure 4 presents a percentage decrease in the value of the internal blocking probability for each traffic class for the case of the analytical calculations (Fig. 4a) and for the case of the simulation experiments (Fig. 4b). The nature of the changes in the internal blocking probability is the same for both the analytical and calculations and the simulation experiments. This means that the modified PPD method makes it possible to evaluate changes in the internal blocking probability in the switching network with overflow links. The results of the study indicate a significant decrease in the internal blocking probability after the introduction of the system of overflow links. The results of the analytical calculations are placed above the results obtained in the course of the experiments, but the error in the calculations is acceptable from the engineering perspective. The proposed method can be, therefore, used to solve practical issues, i.e., analysis, designing and optimization of multi-service switching networks with the system of overflow links.

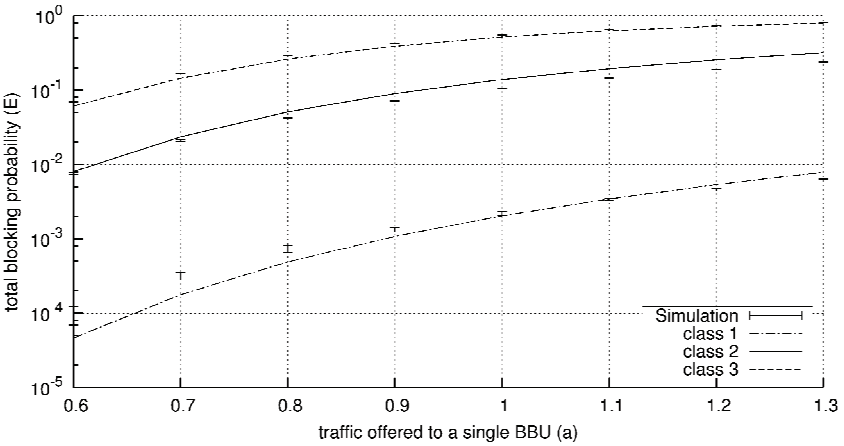

Fig. 3. Total blocking probability in the switching networks with overflow links

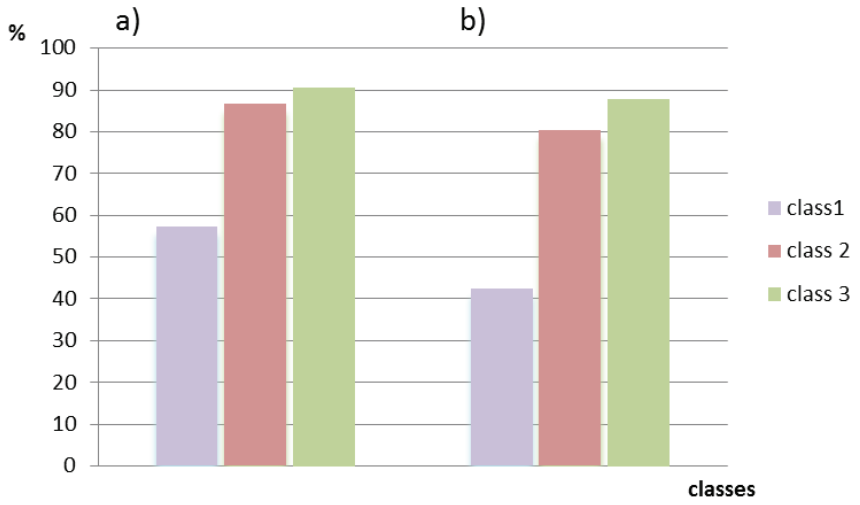

Fig. 4. Percentage decrease in the value of the internal blocking probability; a) results of the analytical calculations, b) simulation results

7. CONCLUSIONS

The present article shows that the PPD method, originally designed for modeling switching networks with multi-service traffic and point-to-point selection, can be also used – after an appropriate modification of the structure of the probability graph – to model switching networks with overflow links. The results of the simulation confirm high accuracy of the proposed method. The results show that a connection of neighboring switches of the first stage by an overflow link results in a significant decrease in the value of the internal blocking probability in the switching network. The conducted study has thus indicated high effectiveness in the operation of the system of overflow links in multi-service switching networks with point-to-point selection.

REFERENCES

[1] CLOS C., *A study of non- blocking switching networks,* Bell System Technical Journal, Vol. 32, No. 2, 1953, 406–424.
[2] KABACIŃSKI W., *Nonblocking Electronic and Photonic Switching Fabrics*, Springer, 2005.
[3] JAJSZCZYK A., *Wstęp do telekomunikacji*, WNT, Warszawa, 1998.
[4] FORTET R. (Ed.), *Calcul d'orange, Systeme Pentaconta*, L.M.T, Paris, 1961.
[5] STASIAK M., *Computation of the probabilisty of losses in commutation systems with mutual aid selector,* Rozprawy Elektrotechniczne, Vol. XXXII, No. 3, 1986, 961–977.
[6] INOSE H., SAITO T., KATO M., *Three-stage time-division switching junctor as alternate route*. Electronics letters, Vol. 2, No. 5, 1966, 78–84.

[7] KATZSCHNER L., LORCHER W., WEISSCHUH H., *On an experimental Local PCM Switching Network,* Proc. International Seminar o Integrated System for Speech, Video and Data Communication, Zurich, 1972, 61–68.
[8] ERSHOVA E., ERSHOV V., *Cifrowyje sistiemy raspriedielenia informacji*, Radio i Swiaz, 1983.
[9] STASIAK M.D., ZWIERZYKOWSKI P., *Performance Study in Multi-rate Switching Networks with Additional Inter-stage Links*, Proc. The Seventh Advanced International Conference on Telecommunications (AICT 2011), St. Marteen, Holland, 2011.
[10] STASIAK M.D., ZWIERZYKOWSKI P., *Multi-service switching networks witch overflow links*, Image Processing and communications, Vol. 15, No. 2, 2010, 61–71.
[11] GŁĄBOWSKI M., STASIAK M.D., *Internal Blocking Probability Calculation in Switching Networks with Additional Inter-Stage Links*, [in]: Information Systems Architecture and Technology, vol. Service Oriented Networked Systems, Wydawnictwo Politechniki Wrocławskiej, Wrocław, 2011, 279–288.
[12] STASIAK M., *Combinatorial considerations for switching systems carrying multi-channel traffic streams.* Annals of Telecommunications, Vol. 51, No. 11–12, 1996, 611–625.
[13] GŁĄBOWSKI M., STASIAK M.D., *Internal Blocking Probability Calculation in Switching Networks with Additional Inter-Stage Links and Engset Traffic.* Proc. 8th IEEE, IET Int. Symposium on Communication Systems, Networks and Digital Signal Processing, Poznań, 2012 (accepted for publication).
[14] STASIAK M.: *Blocage interne point a point dans les reseaux de connexion*, Annals of Telecommunications., vol. 43, No. 9–10, 1988, 561–575.
[15] GŁĄBOWSKI M., STASIAK M.,: *Point-to-point blocking probability in switching networks with reservation*, Proc 16th International Teletraffic Congress, Edinburgh, UK, 1999, Elsevier, Vol. 3a, 518–528
[16] GŁĄBOWSKI M., STASIAK M.: *Multi-service Switching Networks with Point-to-Group Selection and Several Attempts of Setting up a Connection*, [in:] Performance Modelling and Analysis for Heterogeneous Networks, River Publishers, Montreal, 2009, 3–26.
[17] ROBERTS J., MOCCI V., VIRTAMO I. (Eds.), *Broadband Network Teletraffic, Final Report of Action COST 242*, Berlin, Commission of the European Communities, Springer, 1996.
[18] STASIAK M., *Blocking probability in limited-availability group carrying mixture of different multichannel traffic streams*, Annals of Telecommunications, Vol. 48, No. 2, 1993, 71–76.
[19] GŁĄBOWSKI M., STASIAK M.: *Multi-rate model of the group of separated transmission links of various capacities*. Lecture Notes in Computer Science**,** vol. 3124, 2004, 1101–1106.
[20] LEE C., *Analysis of switching networks*, Bell Systems Technical Journal, Vol. 34, No. 6, 1955, 1287–1315.
[21] KAUFMAN J., *Blocking in a shared resource environment*, IEEE Transactions on Communications, Vol. 29, No. 10, 1981, 1474–1481.
[22] ROBERTS J., *A service system with heterogeneous user requirements-application to multi-service telecommunications systems*, [in]: Proceedings of Performance of Data Communications Systems and their Applications, Pujolle G. (Ed.), Elsevier, 1981, 423–431.

network modeling, QoS routing, functional programming

Krzysztof STACHOWIAK*

THE APPLICATION OF THE INDUCTIVE GRAPH MODEL FOR THE MODERN NETWORKS

The classical approach to model a graph has proven its applicability since the XVIII century, when it was first introduced by Leonhard Euler. The emergence of the computers brought the translation of the mathematical algorithms to programming languages which opened paths to solving many complex problems. However years of the programming languages study revealed several weaknesses of the original imperative approach leading to the emergence of the functional programming languages. The functional programming allows describing computation in a safer and terser way.

The foundation of the functional approach to the graph theory is an innovative graph modeling, which represents a departure from the set oriented reasoning. It concentrates on recursive (inductive) structures which enables simpler reasoning about the known algorithms, provides a clean notation and possibly alternative ways of inventing new graph algorithms.

The paper briefly presents the theoretical background for the inductive graphs and explains the transition from the classical to the functional graph algorithms approach. The haskell programming language with a background of the Haskell Graph Library is used as a basis for the notation.

1. INTRODUCTION

The graph theory has a well grounded position in many different scientific disciplines. For centuries the mathematical model has not changed and was simple and robust [1]. Graphs were described in terms of the set theory as two sets: one containing the nodes and another one containing the edges. The graph algorithms have been presented as sequences of operations that change the state of certain mathematical structures such as the aforementioned sets as well as other, algorithm specific constructs. The von Neuman's architecture fits the imperative scheme of the classical

* Poznań University of Technology, Chair of Communication and Computer Networks, ul. Polanka 3, 60-965 Poznań, e-mail: krzysiek.stachowiak@gmail.com

algorithms definition well and thus the programming languages that promptly followed the emergence of the hardware took such a form [2].

For a long time the non-imperative alternatives remained overshadowed. However the hardware's capabilities have been constantly increasing and the software was enabled to scale up freely even beyond the limits of maintainability. The problem of providing robust software at a rapidly growing scale revealed certain problems of the otherwise popular imperative paradigms. The classical programs become hard to reason about when it comes to assessing their correctness especially when the concurrent programming gains popularity [3]. The graph computations can particularly benefit from the fresh mindset even though most of the graph algorithms – both the fundamental [4][5][6] as well as the modern ones [7], [8], [9] – are given in the classical, imperative fashion. The regularity of the structures that are analysed (i.e. Graphs, trees, paths, etc.) provides clean algebraic ways of representation and thus makes them conveniently translatable into the functional domain. The main difficulty in such an endeavour is such a choice of the data structures that enables formulating semantics strong enough to fully take advantage of the functional paradigm.

In order to present the advantages of the non-imperative approach the Haskell language has been chosen. It is a very powerful and expressive language with a syntax that is very similar to purely mathematical formulas which increases its readability in the context of a scientific paper. As an implementation example the FGL library has been chosen [10]. It provides a very interesting and innovative graph model, maintaining good performance and source code terseness [11].

The only downside of the FGL is its simplicity which renders it infeasible for modern graph computation as it only supports a single numerical label per graph edge whereas multiple metrics are usually considered in QoS (Quality of Service) oriented research. Therefore the authors' contribution is a proposition of an extension to the FGL path finding model that enables generic and flexible handling of multiple optimization criteria.

2. THE GRAPH REPRESENTATION APPROACHES

2.1. THE CLASSICAL MODEL

In the first works about the graph theory [4], [5], [6] as well as in the most recent ones [7][8][9], an approach is taken that is strongly rooted in the set theory. The graphs are represented as a finite set of nodes $v \in V$ and the set of edges are defined as pairs of nodes $E = \{(u, v) : u, v \in V\}$. Also additional structures are defined such as maps from the edges to their costs or the subsets of the visited nodes, etc. Building of the partial results for instance also requires additional bookkeeping.

Translating such a model into the functional domain may be performed in many different ways and it turns out that many paths have been explored on the way to obtaining a convenient functional model.

2.2. FUNCTIONAL MODELS

The mutable state, which is seemingly the essence of the graph algorithms does not have a clean representation in purely functional languages.

One of the simplest ways of fitting it into non-imperative program is to thread the structures representing the state trough the function calls so that they can operate on the state they obtained as the input and pass it over by returning the modified version along with the normal function's result [12]. While simple, this solution is rarely elegant as it pollutes the functions' interfaces thus making them inconvenient to use and complicating their composability.

In order to solve this problem, monadic composition may be introduced that is more elegant, but breaks the purity of the program in the scope of its operation [13] thus disabling all the advantages that may be gained from the functional approach. Also like the aforementioned solution it distorts the functions' interfaces.

An interesting alternative has been proposed in [11] which represents a strong departure from the classical approaches replacing them with tools that are natural for the functional programming. It builds up from the recursive model of processing trees or linked lists, which are fundamental functional operations, and proposes a similar way of processing complex topological structures that is not only clean but directly usable for implementing graph algorithms.

3. INDUCTIVE GRAPHS AND THEIR ANALYSIS APPROACHES

3.1. GRAPH CONSTRUCTION

The foundation of the inductive graph model – the heart of the FGL library – is the construction of the graph. Instead of using the node and edge sets, the graph is represented recursively (inductively) as either an empty graph (with no edges or nodes) or a graph extended with a node and its predecessors (connected by the incoming edges) and successors (connected with the outgoing edges). The extending element is called context and takes the following form: (*predecessors, node_id, node_label, successors*), where the predecessors and successors reflect the incoming and outgoing edges respectively, with the following pairs: (*edge_label, connected_node*). This way, starting from the empty graph, and adding the nodes' contexts one by one any graph may be built [14]. Other, more convenient ways are also provided by the library (such as

converting the canonical node and edge lists into an inductive graph), but they are all equivalent and equally valid. This representation may give an impression of being identical to list definition (lists are constructed from an empty list by adding elements to them one by one), however it poses numerous additional restrictions that mirror the actual differences between a graph and a list.

It has been proven that any graph may be constructed by appending the edges in this fashion in an arbitrary order [14]. The same may be said about a decomposition: the graph may be decomposed by taking away nodes' contexts one by one in any order. This mechanism is in a direct parallel to the decomposition of a list by removing the first element until the list is empty.

3.2. GRAPH DECOMPOSITION

The recursive scheme of the graph decomposition is the foundation of the inductive graphs model. The FGL library provides a basic decomposition scheme, by the means of the function *match*. The function attempts extracting a context of the node of a given identifier (the *matchAny* alternative extracts arbitrary contexts) resulting in the context and the rest of the graph remaining after the extraction. They are both very useful for the algorithms consisting in traversing a graph when visiting each node only once is essential. For the ease of the presentation the function *match* which results in a context *c* and the remainder of a graph *g* will from now on be denoted as *c & g* expression.

To give an example, let's consider a *map* function that operates on a complex object by applying an arbitrary function to all of its elements and producing a transformed complex object as a result. Let us begin with an example for lists which is easier to comprehend and at the same time almost identical to the graph counterpart:

```
map :: (a → b) [a] → [b]
map _ [] = []
map f (head:tail) = f head : map f tail
```

For graphs the function looks the following way:

```
gmap :: (Ctx a b → Ctx c d) → Gph a b → Gph c d
gmap _ Empty = Empty
gmap f (c & g) = f c & gmap f g
```

One will notice a strong parallel between the list decomposition with the *cons* operator (*head:tail*) and the match decomposition represented here with the & active pattern [15].

4. THE IMPLEMENTATION DETAILS

In the imperative programming there are two basic ways of representing graphs. Adjacency lists assign a list of all outgoing neighbours to each node. Each of such mapping is associated with a cost of the corresponding edge. An alternative – the incidence matrix – consists of a two dimensional array storing costs of the edges between all possible node pairs, conventionally putting infinity or an equivalent wherever an edge between a given pair of nodes does not exist. The incidence matrix provides a faster lookup of specific edges O(1) compared to O(n) for adjacency lists, but a slower access to a list of a node's neighbours O(n) compared to O(1). However the incidence matrix takes up much more of memory for all but very dense graphs.

The direct functional counterparts of the above aren't efficient enough. Therefore different structures are used in the non-imperative programs. Care must be taken when choosing the low level constructs as even in case of the imperative implementations a proper choice of the data structures is crucial as e.g. the Dijkstra's algorithm, which originally presented the complexity of O(n2) may be optimized to the running time of O(m log n) if the linear search is replaced with a more efficient tool. The creator of the FGL library put a lot of research into selecting the proper data structures which provide a performance comparable to the imperative alternatives [11].

The so called term based implementation of the graphs (such as the one shown in the Gph type) provides a valuable property of persistence, i.e. obtaining a modified version of a graph retains the original copy. It is however very inefficient when implemented directly. Therefore for the Haskell implementation a search-tree based implementation was chosen, which provides fast lookup of the elements, efficient insertion and is still compatible with the concept of the graph construction and decomposition presented in the section 3 [11].

5. FUNCTIONAL GRAPH ALGORITHMS PRESENTATION

In order to present the benefits from assuming the inductive graph model and a non-imperative algorithm definition and an its original extension will be presented. The classical Dijkstra's algorithm [4] that is an integral part of the FGL library will be assumed as the basis for the extension.

The main drawback of the FGL's implementation of the Dijkstra's algorithm is that only a single metric of a numeric type is allowed for the edge labelling and therefore this representation is not always suitable for modern telecommunications as the modern problems often consist in optimizing multiple criteria. In case of algorithms that aggregate the metrics in a linear manner, the costs may be computed before the path finding phase and a single metric graph may be then passed to the solving func-

tion. However in case of nonlinear definitions of the metrics' aggregation one cannot precompute the aggregated costs as the aggregated cost of a path doesn't equal to the sum of the aggregations of the edges' costs. The multiple labels must therefore be stored throughout the computations and summed independently only to be aggregated at the point when it is necessary. The FGL is actually operating on an abstract type for the cost manipulation, however it requires the client to provide an instance for a Real data type. This implies the necessity of providing instances of our type for classes such as Num in which case most of the required functions such as *abs*, *signum* or operator * are unreasonable for any type of a complex metric such as a list of the partial costs. The heap implementation enforces providing an instance of the *Ord* type class which requires providing operation for numerous different comparisons, whereas only the *less-than* operation is really required by the code.

Therefore an alternated implementation of the FGL's Dijkstra's algorithm will be presented, accompanied by a modified implementation of the heap structure used in the library. This generalized model may serve an arbitrary way of aggregation of the metrics along the path as well as combining them into a single value wherever the Dijkstra's algorithm performs a classical label comparison.

5.1. DIJKSTRA'S ALGORITHM IMPLEMENTATION

The basic structures besides the graph that will be used are the labeled path which is a sequence of labeled nodes and labeled root tree which is a list of labeled paths rooted in the source node:

```
type LNode a  = (node, a)
type LPath a  = [LNode a]
type LRTree a = [LPath a],
```

where the a type is of the *Real* type class.

The complete implementation of the Dijkstra's algorithm is as follows:

```
expand :: Real b => b → LPath b → Ctx a b → [Heap (LPath b)]
expand d p (_, _, _, s) = map (\lv) → unitHeap((v, l+d):p)) s

dkstr :: Real → b => Heap (LPath b) → Gph a b → LRTree b
dkstr h g | isEmptyHeap h || isEmpty g = []
dkstr (p@((v,d):_)<h)(c&ᵛg)= p:dkstr(mrgAll(h:expand d p c)) g
dkstr (_<h) g= dkstr h g
```

It is worth noting that $\prec$, & and $\&^v$ are active patterns which are not available in Haskell, but their real Haskell equivalents are explained in [11]. The $\&^v$ operator means extracting a context of a given node or a failure at matching, should such node not exist in the graph. The $\prec$ operator means extracting the top element from the heap.

The function accepts two arguments: the initial heap of the nodes to be visited and the graph to be searched. The first equation of the *dkstr* function definition handles a case when either the heap or the graph that are passed is empty. This means the end of the computation and the empty list that is returned finishes the recursion. Several more passes may be performed for the remaining graph nodes or heap elements (should one of the structures still contain elements), but they will all be caught by the same equation and the function will terminate eventually performing no additional work. The second equation handles a situation when a front node of the path picked from the top of the heap (the $(p@((v, d):\_)\prec h)$ expression) can be found in the current part of the graph (the $(c\&^v g)$ expression). In such case the path picked from the heap is prepended to the result of the rest of the computation which is denoted as the recursive call to the *dkstr* function. The recursive call is passed the graph after the extraction of the currently considered node and a heap built out of the rest of the heap after extracting the currently considered top element. The neighbours to be visited are determined with the *expand* function which also computes the accumulated costs of reaching them. Note that the expansion may lead to visiting the nodes that have already been visited, however their presence in the heap presents no problem. When they're encountered, the graph pattern matching will fail due to the fact that in the subsequent recursion levels we only take into account the decomposed part of the graph which no longer contains any of the already visited nodes.

5.2. MULTICRITERIAL EXTENSION

In order to increase the robustness of the algorithm definition the requirement of the metric type being of the *Real* type class will be replaced with an entirely new type class that we will call CmpMetr which stands for "comparable metric". The type class definition is the following:

```
class CmpMetr a where
cm_less_than :: a → a → Bool
cm_expand :: a → a → a
```

This type class states that the metrics must be comparable as well as expandable. The comparison is performed within the heap and the expansion is performed by the aforementioned *expand* function. The *less-than* comparison is performed within the heap. With the requirements defined this way we may provide a more terse definition

of the metric type as we only define the essential operations for them. The proposed mechanics would also serve nonlinear or threshold aggregations, but as a simple example a linear combination metric shall be demonstrated.

We wish to be able to assign multiple real valued costs to each edge:

$$c(e) = (c_0, c_2, ...), e \in E$$

In order to combine them into a single value for the comparison we use the linear combination of the costs with a predefined set of weights:

$$w = (w_0, w_1, ...)$$

A cost vector for any subgraph $sg_{e0, e2, ...} = \{e_0, e_1, ...\}$ consisting of multiple edges, where is defined as follows:

$$c(sg) = \{ \Sigma c(e_i)[0], \Sigma c(e_i)[1], ...\},$$

and an aggregated cost used for the comparison is the following:

$$c_{aggr,w}(c) = \Sigma c_i \cdot w_i$$

Let us present the Haskell code for defining a type that could define such a metric. We must start with defining a data type storing the list of costs and the list of the weights:

```
data LinCmb = LinCmb [Double] [Double]
```

Next we must provide an instance of the CmpMetr type class:

```
instance CmpMetr LinCmb where
     cm_less_than (LinCmb ams aws) (LinCmb bms bws) =
          (aggr ams aws) < (aggr bms aws)
     where aggr ms ws = sum $ zipWith (*) ms ws
     cm_expand (LinCmb ams aws) (LinCmb bms bws) =
          LinCmb (zipWith (+) ams bms) aws
```

Once this code is provided, two pieces of code in the original FGL library must be changed. The comparison of the heap elements and the expansion of the metrics in the Dijkstra's algorithm. Therefore we modify the *expand* function defined above to take the following form:

```
expand :: CmpMetr b => b → LPath b → Ctx a b → [Heap (LPath b)]
    expand d p (_, _, _, s) =
        map (\lv) → unitHeap((v, l `cm_expand` d):p)) s
```

The heap merging function is the kernel of the heap's functionality (all other comparisons are done through the function), therefore we modify its original FGL's implementation into the following, modified form:

```
merge :: CmpMetr a => Heap a b -> Heap a b -> Heap a b
   merge h Empty = h
   merge Empty h = h
   merge h@(Node key1 val1 hs) h'@(Node key2 val2 hs')
       | key1 `cm_less_than` key2 = Node key1 val1 (h':hs)
       | otherwise = Node key2 val2 (h:hs')
```

Once the changes are made we may perform path finding for graphs described with more than one metric per edge. We may also provide more types compliant with the *CmpMetr* type class in order to create many different algorithms. Such a modified version of the Dijkstra's algorithm is often a base for building modern routing algorithms [7], [9].

6. CONCLUSIONS

The functional programming paradigm proves to be very useful for certain types of computations. The closer to maths is the domain, the more we gain from using the non-imperative constructs. The graph theory and the graph algorithms are presented by very regular structures that are relatively easy to represent in simple mathematical terms. It is therefore very natural to apply purely functional techniques in solving topological problems.

Studies reveal that the translation of imperative algorithms into the functional world may sometimes be done in a rather uninspired way by utilizing the imperative mechanisms of functional languages such as monads. Whereas it is advantageous in situations where no alternative exists (e.g. real world I/O can't be represented in a purely functional manner), it may pose a pitfall of retrieving a functionally equivalent implementation with an awkward syntax and absolutely no additional benefits. The FGL library proves that the pursue of purity may lead to very valuable conclusions and that the functional programming is not just a world of highly restricted bizarre languages, but also an alternative mindset that may broaden the spectrum of the graph algorithms research.

The benefits to the programming itself are a lot clearer. The functional programs are short and easy to reason about in a formal way. Besides the facilities such as the unit tests, proofs may be stated about purely functional programs with relative ease. The unit tests also benefit from the purity thanks to the referential transparency, i.e. the property that guarantees that for a given input the output will always be the same, which means that no global state is modified during the function execution.

Further research of the possible advantages of the non-imperative approach is planned as the field is extremely promising in terms of the ease of implementing and testing new algorithms thanks to both: the terseness of the Haskell language and the ease of writing robust less error prone experimental implementations.

REFERENCES

[1] BIGGS N., LLOYD E., WILSON R., Graph Theory, 1736–1936, Oxford University Press, 1986.
[2] LYNN, M. STUART, *Communications of the ACM*, New York, NY, USA,ACM, Vol. 12, No. 11, Nov. 1969.
[3] ARMSTRONG J., *Programming Erlang: Software for a Concurrent World*, Pragmatic Bookshelf, 2007.
[4] DIJKSTRA E. W., *A note on two problems in connexion with graphs*, Numerische Matematik, Vol. 1 1959, 269–271.
[5] BELLMAN R., *On a Routing Problem*, Quarterly of Applied Mathematics, 1958, Vol.16, 87–90.
[6] FLOYD R. W., *Algorithm 97: Shortest path*, Communications of the ACM, New York, NY, USA, Vol. 5, No. 6, 1962.
[7] FENG G., *Nonlinear Lagrange relaxation based QoS routing revisited,* Consumer Communications and Networking Conference, 2005, 504–509.
[8] KOU L., MARKOWSKY G., BERMAN L., *A Fast Algorithm for Steiner Trees,* Acta Informatica 1981, Vol. 15, 141–145.
[9] STACHOWIAK K., WEISSENBERG J., ZWIERZYKOWSKI P., *Lagrangian relaxation in the multicriterial routing*, AFRICON, 2011, 1–6.
[10] ERWIG M., *FGL/Haskell – A Functional Graph Library User Guide*, Praktische Informatik IV, Apr. 2000.
[11] ERWIG M., *Inductive graphs and functional graph algorithms*, Journal of Functional Programming, Sep. 2001, 467–492.
[12] BURTON F. W., YANG H.-K., *Manipulating Multilinked Data Structures in a Pure Functional Language*, Software – Practice and Experience 1990, Vol. 20, No. 11, 205–206.
[13] KINK D. J., LAUNCHBURY J., *Structuring Depth-First Search Algorithms in Haskell*, 22nd ACM Symposium on Principles of Programming Languages, 1995, 344–354.
[14] ERWIG M., *Functional Programming with Graphs*, 2nd ACM International Conference on Functional Programming 1997, 52–65.
[15] ERWIG M. *Active Patterns*, 8th International Workshop on Implementation of Functional Languages 1996, 21–40.

switching fabric, scheduling algorithm,
connection pattern, empty and non-empty connections

Grzegorz DANILEWICZ, Marcin DZIUBA*

THE NEW SSMPS ALGORITHM FOR VOQ SWITCHES

In this paper we present the new Single Size Matching with Permanent Selection (SSMPS) algorithm for control crossbar switches with VOQ. Our algorithm provides high speed of working. If at least two connections between inputs and outputs are established with no packets to be send, our algorithm tries to finds better connection patterns. This solution provides high efficiency of our algorithm with no additional calculations. For this reason, presented algorithm is easy to implement in hardware.

1. INTRODUCTION

Advancement in telecommunication technology has provided high speed data transfer with using cost effective methods. This methods are optical technology and the new generation of packet switches. The most important architecture for packet switches design is crossbar. The reason why they are so popular is the easy way to implementation and possibility of using large number of inputs and outputs. In the same time high speed buffering systems should be provided. The combination of these elements (crossbar architecture and buffering system)gives the good prospering system which can be used to design the new generation of routers. In literature can be found a few kinds of buffering systems and they can be divided into three main groups: input, output and combined buffering systems. When buffers are placed at the inputs to the switching fabric, then we have input buffering systems. Output buffering systems are realized by placed buffers on each switching fabric outputs. When buffers are placed inside the switching fabric, then we have architecture called Input Queued (IQ) switches. There are many possibilities for combining different types of buffers.

* Chair of Communication and Computer Networks, Poznan University of Technology, ul. Polanka 3, 60-965 Poznań, Poland.

For this reason we have Combined Input and Crosspoint-Buffered (CICB), Combined Input and Output-Queued (CIOQ) and many other such buffering systems.These architectures are known to have better performance, than presented in this articleswitch architecture. In the other side, these architectures are more complicated in implementation, that presented in this paper, switch architecture.Algorithms used to control CICB or CIOQ are complicated in implementation and consumed more hardware resources than input buffering system.

In this paper we present switch architecture with input buffering system. This architecture is easy to implementation but this kind of switches can provided only 58,6% throughput. The reason for the low throughput is the Head of Line (HOL) blocking effect. One of the techniques, used to eliminate this unwanted phenomenon is Virtual Output Queuing (VOQ). Virtual Output Queues were first used and proposed in [1], [20]. A scheduling algorithm is used to configure our switch and find the most optimal connections between inputs and outputs. Optimal means that in one time slot (basic time unit) we want to send the greatest number of packets. Our results have shown that 100% throughput can be achieved for high traffic load. In this article we present SSMPS algorithm which is compared with another scheduling algorithms by using computer simulations. We have compared: iSLIP algorithm which is presented in [15], Parallel Iterative Matching (PIM) [1], Iterative Round-Robin Matching (iRRM) [16], Maximal Matching with Random Selection (MMRS) [3], [4], [5], Maximal Matching with Round-Robin Selection (MMRRS) [3], [4], [5], Random [8] and Permanent [8].

This paper is organize as follows. In the second section switch architecture is presented. Next chapter is dedicated to the new algorithm description. In the fourth section simulation results are presented. At the end some conclusion are given.

2. SWICH ARCHITECTURE

Switch architecture which was used in our research is shown in Figure 1 [2].

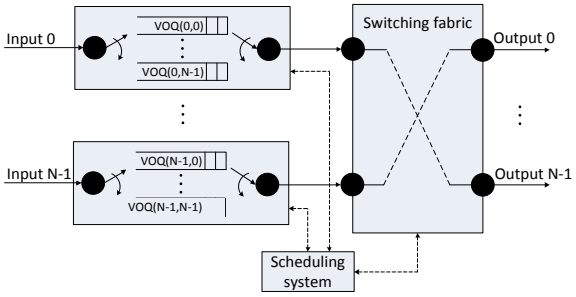

Fig. 1. Switch architecture

From Figure 1 it can be observed that this switch consists of input buffers and outputs. Between these elements switching fabric is placed. Switching fabric is the place where connections between inputs and outputs are established. Each input buffer is divided into N independent VOQs. The total number of virtual output queues depends on number of inputs and outputs. In a symmetrical switching fabric, which is the object of our research, number of inputs and outputs are equal. Under these assumptions, total number of VOQs is N. From Figure 1 can be seen that each virtual queue is denoted by VOQ (i, j) where i is the input port number and j is the output port number. We assumed that: $0 \leq i \leq N-1$ and $0 \leq j \leq N-1$.

The most important element, in presented switch architecture, is centralized scheduling mechanism. In this module the new algorithm is implemented. This algorithm is responsible for configures the fabric switching during each time slot and decides which inputs will be connected to which output. Scheduling system module is responsible for collecting data about VOQs condition. It means that scheduling system has information about number of packets waiting in queues to be send through the switch. This information are used to make decision about connections in switching fabric by algorithm. Similar solutions were presented in [12], [13], [21]. More details about algorithm are presented in the next chapter.

3. SCHEDULING ALGORITHM

Presented algorithm based on permanent connection between inputs and outputs. Connection patterns in 4x4 switch, in each time slot are presented in Figure 2.

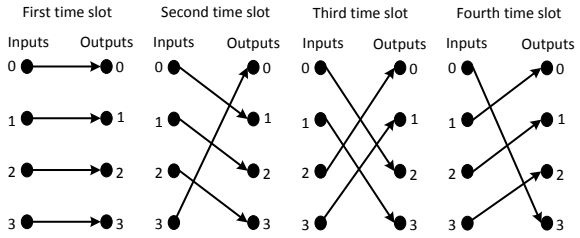

Fig. 2. Permanent connections pattern

It can be observed that in each time slot, we have different connection pattern. Total number of connection patterns is N, where N is a number of inputs and outputs. As mentioned, the most important element in presented switch architecture is scheduling system briefly called scheduler. This element is responsible for storing information about number of packets waiting in each virtual output queues. Based on this information Matrix of Queue Length has been created. Matrix is the simplest way to store

this kind of information and quick access to the information is provided. For example from Figure 3 can be seen how the MQL matrix has been complemented.

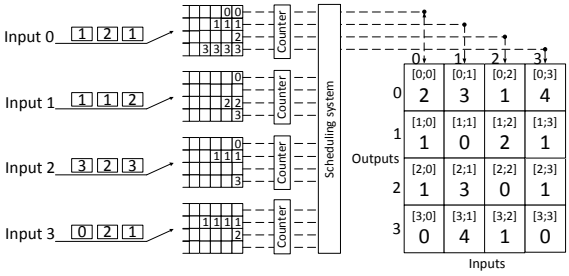

Fig. 3. MQL matrix

In presented 4×4 switch, there are four input buffers each consists of four VOQ because in our switch architecture we have four outputs (each VOQ is dedicated to a separated output). This is equivalent to the matrix structure, which is organized in four rows and four columns.Packets are counted by counters placed at each input buffer. Presented matrix is filled by rows. It means that first filled row is identified as 0. Second is 1 and so on, until we get to the last row. Each item in the matrix has a unique identification. In the general case, each position can be marked as [*i*;*j*], where *i* is the input number and *j* is the output number. Each position in our MQL matrix, reflecting situation in each VOQ. For example,position [0;0] in matrix is adequate to the VOQ(0;0). Going further in our discussion, position [0;0] in matrix is filled by 2 because in VOQ(0,0) there are two packets waiting to be send through the switch from input 0 to the output 0 (*i*=0, *j*=0). Next item in MQL matrix, marked as [0;1] is filled by 3, which means that in VOQ (0,1) three packets are waiting to be send. In conclusion can be said that row sum is a total number of packets, waiting in each input buffer. The column is the number of packets destined to the suitable output.

After matrix filling operation, algorithm marks all positions in MQL matrix according to permanent connections in the first time slot. Connection patterns for each time slot in 4×4 switch is shown in Figure 2.

Connections in first time slot in switching fabric and MQL matrixis shown in Figure 4. All connections in presented switch is marked by arrows. Connections between inputs and outputs are classified into two groups. First one of them is empty connections group which contains all connections where there is no packets to be send. This kind of group is marked by dotted arrows in switching fabric and by dotted ellipses in MQL matrix (Figure 4). Second group is non-empty connections group, where belong connections with packets to be sent through the switch. They are labeled by solid arrow in switching fabric and solid ellipse in MQL matrix (Figure 4).

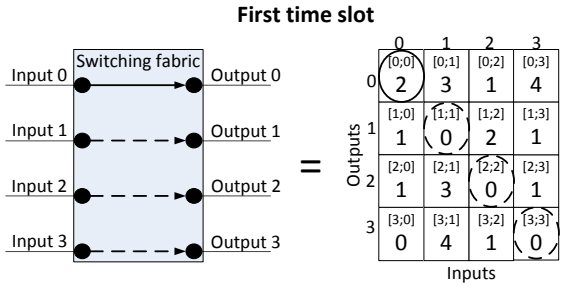

Fig. 4. Connection pattern in first time slot in switching fabric and MQL matrix

As can be seen from Figure 4, there are three empty connections. When in the current connections pattern is more than two empty connections, our algorithm tries to find better connection pattern. Better means that algorithm eliminates connections where are no packets to be send. The point is to send the biggest number of packets in one time slot. In the same time, equitable access to the output should be provide. In this case, permanent connection patterns provide it. The way how algorithm replaces connections is very quick and easy. In first step all non-empty connections are found and protected (in presented example it is connection from input 0 to output 0, position [0;0] in MQL matrix). Then all outputs and all inputs which are not available are eliminated. In this case it is row 0 and column 0 (Figure 5).

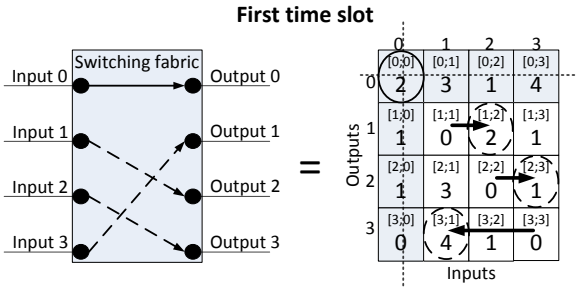

Fig. 5. The new connection pattern in switching fabric and MQL matrix

After this, first empty connection is replaced on the next available. The same situation is with the rest of empty connections. In presented example we finally received four non-empty connections instead of one non-empty and three empty. Algorithm does not take into the account what kind of the new connection is selected. It can be non-empty or empty connection. What is certain, that we do not receive worst connection patterns. Strong point of this solution is no additional calculations performed by algorithm. Calculations may take extra time and hardware resources. Finding a new connection patterns, in a short time, is the most important task for switching algorithms. Our algorithm is easy to implementation and finds new connection patterns in simple

way. Final connection pattern with scheduling system and MQL matrix is shown in Figure 6.

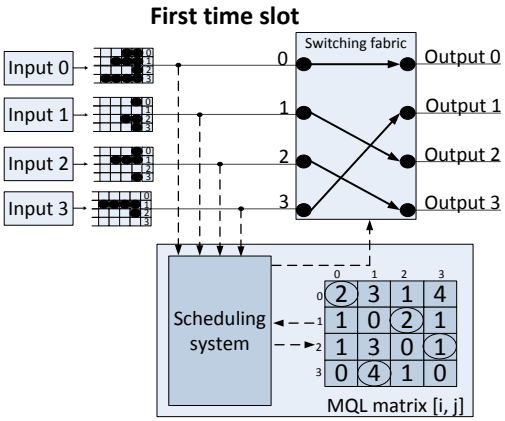

Fig. 6. Final connection pattern

In conclusion the new algorithm is worked base on few steps presented belowinProcedure 1. Input data are: switch_size – number of switch inputs/outputs and connection_pattern[output_*n*, … , output_*switch_size*-1] which hasarray form and on the next positions array consists of number of output. For example, in first time slot, for 4×4 switch, array has a form: connection_pattern[0, 1, 2, 3].

PROCEDURE 1:

```
Input data:switch_size, connection_pattern[output_n, … ,output_switch_size-1]
Begin
    empty_connection = 0;
    fori =0 toi≤ switch_sizedo
            if [i; connection_pattern[i]] = 0;
              empty_connection+1;
            endif;
        i+1;
    endfor
if (empty_connection>2)
    fori =0 toi≤ switch_sizedo
            if [i;connection_pattern[i]] = 0;
              connection_pattern[i] = connection_pattern[i] +1;
            endif;
        i+1;
        endfor;
```

4. SIMULATION RESULTS

We compared our algorithm with other by using computer simulations. Simulations were performed for different switches sizes: 4×4, 8×8, and 16×16. All of these

switches were equipped with VOQs of infinity size. Compared algorithms were simulated under Bernoulli packet arrivals [6], [9], [10], [19].It was assumed that one packet may occupied one time slot. Probability of packet arrivals at the input, in the given time slot is p, where $p \in (0<p\leq 1)$. Packets directed to particular output were distributed uniformly. Presented simulations results are shown as mean values ten independent simulation processes. Iterations is 500 000 where 30 000 iterations are required for obtaining convergence in the simulation environment. In our research, two parameters were compared.

1) Efficiency

Efficiency q was calculated according to equation 1.

$$q = \frac{{}_n a_n}{{}_n b_n}$$

where:

n – time slot number,

a_n– number of packets passed in n time slot through the switch,

b_n– number of packets which can be sent in n time slot through the switch.

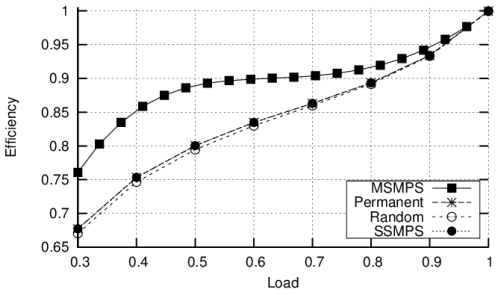

Fig. 7. The Efficiency in 4×4 switch for Bernoulli packet arrivals withdestination uniformly distributed over all outputs

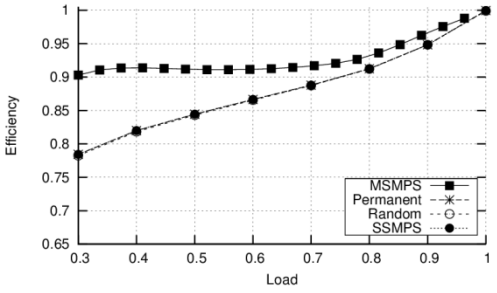

Fig. 8. The Efficiency in 8×8 switch for Bernoulli packet arrivals withdestination uniformly distributed over all outputs

The efficiency in 4×4, 8×8 and 16×16 switches for Bernoulli packet arrivals are shown in Figure 7, Figure 8 and Figure 9. It can be observed when the traffic load is growing, the efficiency also grows for Random MSMPS, Permanent and SSMPS algorithms (Figure 7 and 8). For the small switches sizes, SSMPS algorithm achieved better results than Random algorithm (Figure 7). The best efficiency achieved MSMPS algorithm. From Figure 8 and 9 can be observed that SSMPS and Permanent algorithms achieved similar results for high load. The reason is that more non-empty connections are executed and SSMPS algorithm does not need to change connection patterns. For high traffic load, more packets are waiting in VOQs to be sent through the switch.

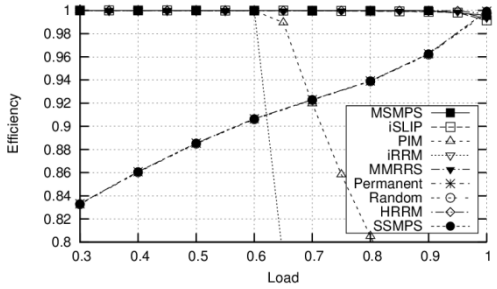

Fig. 9. The Efficiency in 16×16 switch for Bernoulli packet arrivals with destination uniformly distributed over all outputs

From Figure 9 can be seen that above 63% load, SSMPS algorithm achieved better results than iRRM algorithm and above 70% load achieved better results than PIM algorithm. Efficiency for PIM andiRRM algorithms rapidly decrease above 60% load. The reason of PIM and iRRMalgorithms behavior is the result of arbiters synchronization.

2) Mean Time Delay (MTD) [7].

$$MTD = \frac{{}_n(t_{out} - t_{in})}{k}$$

where:

n – time slots number,
t_{in} – time a packet arrive to the VOQ,
t_{out} – time when the same packet is transferred by the switch,
k – number of packets.

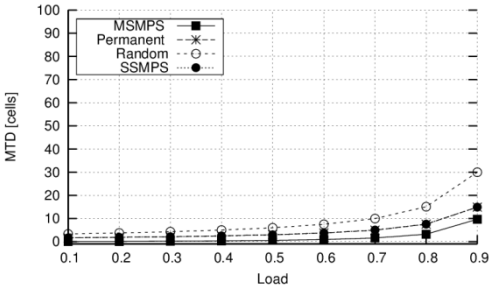

Fig. 10.MTD in 4×4 switch for Bernoulli packet arrivals with destinationuniformly distributed over all outputs

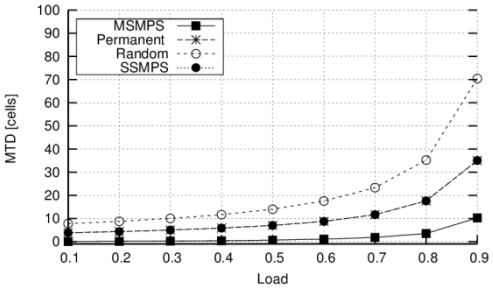

Fig. 11.MTD in 8×8 switch for Bernoulli packet arrivals with destinationuniformly distributed over all outputs

The efficiency in 4×4 and 8×8 switches is plotted in Figure 10 and Figure 11. It can be observed that when the traffic increasing, the MTD also become higher. For the low

traffic load, SSMPS algorithm achieved lower MTD than Random algorithm. Permanent and SSMPS algorithms achieved the same results for all presented switches sizes.

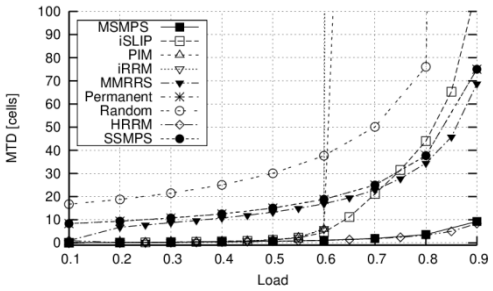

Fig. 12.MTD in 16×16 switch for Bernoulli packet arrivals
with destinationuniformly distributedover all outputs

From Figure 12 can be observed that above 60% load, SSMPS algorithm achieved lower MTD than PIM and iRRM algorithms. MTD for iSLIP algorithm rapidly increase above 60% and more than 75% achieved worse results than SSMPS algorithm. MTD for our algorithm grows systematically but not rapidly like in others compared algorithms. SSMPS algorithm achieved 75 cells (time slots) delay for 90% load.

5. CONCLUSION AND FURTHER WORKS

In this article we have presented SSMPS scheduling algorithm which is used to configure switches with VOQs. Objective of our research was simulate SSMPS algorithm under Bernoulli packet arrivals and compare results with another algorithms known from literature. Computer simulations was performed for constant packet length (packet may occupied one time slot), different switches sizes and for different traffic load. Packets were distributed uniformly. Gathered results show that our algorithm achieved the same results like the rest of compared algorithm and for some cases, SSMPS algorithm achieve better efficiency and lower MTD. Presented algorithm works very fast and does not need to do complicated calculations. For this reason SSMPS is easy to implementation and we would like to try implement it in Field Programming Gate Array (FPGA) matrixes [14].

In an future works we want to present SSMPS algorithm under different traffic and distribution models. It would be interesting to execute simulation under bursty [6] packet arrivals with uniformly and non-uniformly packets distribution. In the next research we want to use bigger switches sizes. We also want to improve efficiency for high traffic load. These results will be presented in the next article.

REFERENCES

[1] ANDERSON T. and et al., *High-speed switch scheduling for local-areanetworks*, ACM Transactions on Computer Systems, Vol. 11, No. 4, pp. 319–352, November 1993.

[2] BARANOWSKA A., KABACIŃSKI W., *Hierarchiczny Algorytm Planowania Przesyłania Pakietów Dla Przełącznika z VOQ,* Poznańskie Warsztaty Telekomunikacyjne (PWT), Poznań, 2004.

[3] BARANOWSKAA., KABACIŃSKIW., *The New Packet Scheduling Algorithms for VOQ Switches* ICT 2004, LNCS 3124, pp. 711–716, 2004.

[4] BARANOWSKAA., KABACIŃSKIW., *MMRS and MMRRS Packet Scheduling Algorithms for VOQ Switches*, MMB PCTS 2004, pp. 359–368, September 2004.

[5] BARANOWSKAA., KABACIŃSKIW., *Evaluation of MMRS and MMRRS Packet Scheduling Algorithms for VOQ Switches under Bursty Packet Arrivals*, High Performance Switching and Routing (HPSR), pp. 327–331, May 2005.

[6] CHAOJONATHAN H., LIUB., *High Performance Switches and Routers*, ISBN-13:978-0-470-05367-6, John Wiley and Sons, pp. 195–197, New Jersey, 2007.

[7] DANILEWICZG., DZIUBAM., *The New MSMPS Packet Scheduling Algorithm for VOQ Switches,* 8th IEEE, IET International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP), Poznań, 2012.

[8] DZIUBA M., *Comparison of Packet Scheduling Algorithms for VOQ Switches,* Poznańskie Warsztaty Telekomunikacyjne (PWT), Poznań, 2011.

[9] GIACCONEP., PRABHAKARB., and SHAH D., *Randomized Scheduling Algorithms for High-Aggregate Bandwidth Switches,* IEEE L Select. Areas Cam, vol. 21, pp. 5e559, May 2003.

[10] GIACCONEP., SHAHD., and PRABHAKARS., *An Implementable Parallel Scheduler for Input-Queued Switches*, IEEE Micro, Vol. 22, No. 1, pp. 19–25, 2002.

[11] KAROL M. J., HLUCHYJ M., MORGAN S. *Input vs output queuing on a space-division packet switch*, proceeding of the BLOBCOM, Huston, 1986, pp. 659–665.

[12] LaMAIRER. O.,SERPANOSD. N., *Two-Dimensional Round-Robin Scheduler for Packet Switches with Multiple Input Queues*, Journal IEEE/ACM Transactions on Networking (TON), Vol. 2 Issue 5, October 1994.

[13] LIUN.H., KWAN L. YEUNG, DEREK C. PAO, *Scheduling Algorithms for Input-queued Switches with Virtual Output Queueing*, EEE International Conference on Communications Proceedings, Helsinki, Finland, 11–14 June 2001, Vol. 7, pp. 2038–2042.

[14] MAJEWSKI J., ZBYSIŃSKI P., *Układy FPGA w przykładach*, BN 978-8360233-23-8,BTC, Warszawa 2007.

[15] McKEOWNN., *The iSLIP Scheduling Algorithm for Input-Queued Switches*, IEEE/ACM Trans. on Networking, Vol. 7, pp. 188–200, April 1999.

[16] McKEOWNN., VARAIYAP., and WARLANDJ., *Scheduling cells In an Input-Queued Switch*, IEE Electronics Letters, pp. 2174–2175, 1993.

[17] ROJAS-CESSAR., OKI E., ZHIGANG JING, CHAO JONATHAN H., *CIXB-1: Combined Input-One-cell-Crosspoint Buffered Switch*, High Performance Switching and Routing (HPSR), 2001.

[18] ROJAS-CESSAR., *High-Performance Round-Robin Arbitration Schemes for Input-Crosspoint Buffered Switches*, High Performance Switchingand Routing (HPSR), 2004.

[19] SHAHD.,GIACCONEAND P. and PRABHAKARB., *Efficent Randomized Algorithms for Input-Queued Switch Scheduling*, Prm. HOT1 H, Vol. 22, pp. 10–18, Jan. 2002.

[20] TAMIR Y. and FRAZIERG., *High performance multiqueue buffers for VLSI communication switches,* Proc. 15th Annu. Symp. Comput. Arch., pp. 343–354, June 1988.

[21] YU H., RUEPPS., BERGERM. S., *A Novel Round-Robin Based Multicast Scheduling Algorithm for 100 Gogabit Ethernet Switches*, IEEE Conference INFOCOM, San Diego USA, 2010.

optical switching networks, crosstalk,
OSXR, baseline-based switching networks,
$\log_2 N$*–1 switching networks,*

Remigiusz RAJEWSKI*

QUALITY OF OPTICAL CONNECTIONS IN THE LOG$_2N$–1 SWITCHING NETWORK

In this article, it is shown how to calculate first-, second-, and third-order crosstalk stage-by-stage in the $\log_2 N - 1$ switching structure. Achieved results are compared with the traditional baseline network. The $\log_2 N - 1$ structure gives better optical signal-to-crosstalk ratio for this same functionality and capacity of the switching fabric. It is also discussed how the optical signal goes through a switching network, through what kind and what number of an optical elements. There are also shown exact calculations of the number of passive and active optical elements. The number of such an elements is compared with traditional networks of the same capacity. The $\log_2 N - 1$ network has in many cases fewer number of such an elements.

1. INTRODUCTION

In the optical switching theory very often baseline switching fabric [1] is used. This network is also called the $\log_2 N$ switching network [2], where N denotes the capacity of such a structure. This architecture is build only from symmetrical optical switching elements (OSEs) of size 2×2. In a general case, OSE can has size of $d \times d$ and this structure is then called the $\log_d N$ switching network [3]. Such a baseline structure was later also extended to another networks of different functionality [4], [5], and [6]. More structures and their variants was described in [7], [8], and [9]. In [10] it was proposed another architecture, the $\log_2 N - 1$ switching fabric, which is better for almost all capacities of the switching network than the baseline architecture of the same functionality and capacity. The $\log_2 N - 1$ network is, however, build from both asymmetrical and symmetrical OSEs. Details about how this architecture looks for

* Chair of Communication and Computer Networks, Faculty of Electronic and Telecommunications, Poznan University of Technology, 3 Polanka St, 60-965 Poznań, Poland.

particular capacities and how it is extended to the switching network of greater capacity are described in details in [10].

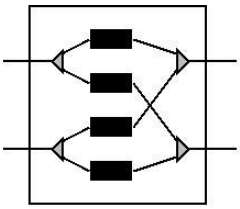

Fig. 1. Optical switching element of size 2×2

Different optical switching network structures are be build from passive and active optical elements. To compare these structures between themselves the number of such an optical elements should be counted in each architecture. As an active optical element it could be used semiconductor optical amplifier (SOA) [11], [12], and as a passive optical element it could be used splitter or combiner at the input or output side of an OSE, respectively. In Fig. 1 it could be seen optical switching element of size 2×2 which is build from 4 SOAs (rectangles filled in the black color), 2 splitters of size 2×2 (triangles filled in the gray color at the left side of OSE) and 2 combiners of size 2×2 (triangles filled in the grey color at the right side of OSE). This structure which consists from fewer number of SOAs, splitters, and combiners is cheaper solution (is better).

2. ARCHITECTURE

SOA-based optical switching network has strength because of the quite large bandwidth served by SOAs [11], [12]. What's more, gain of such an amplifier can be settled directly to compensate all losses which occurs in one OSE. These losses can come from passive optical elements like splitters or combiners. Using SOA an optical signal, which appears at the input in the OSE, can be switched inside optical switching elements to the proper output. In this situation amplifier works like ON-OFF switch. When this switch is in ON state optical signal can goes through it (SOA is running and signal is then amplified). When this switch is in OFF state optical signal cannot goes through it (SOA is turned off and signal cannot be sent through it or SOA is running and signal is attenuated).

General, a switching architecture is build from greater number of OSEs which constitutes stage or even stages. The right number of optical switching elements depends strongly of kind of a switching network. There were described and introduced many networks [7], [8], and [9], however, one of the best known and very often used in opti-

cal switching is baseline network [1]. Recently, in [10] it was introduced a new architecture which is even cheaper (therefore, it is better choice), where the cost of a whole switching network is expressed as the number of active and passive optical elements.

3. CONNECTION QUALITY

Quality of an any connection in the switching network depends strongly on architecture itself and, of course, of the technology in which such a structure is done. In this article all switching networks are build in this same technology (SOAs, splitters, and combiners – it allows to compare these structures between each other), and the main focus in put on the structure itself.

It could be assumed that at the input side of the switching network there are optical signals of powers P_1, P_2, P_3, …, P_N for inputs 1, 2, 3, …, N, respectively, where N denotes the number of inputs in the whole switching structure. To simplify, it can be assumed that each of such a signals has optical power P_{in}. The optical signal, which

Table 1. Stage-by-stage crosstalk in the $\log_2 8$ optical switching fabric

OSE	output	after stage s_1	after stage s_2	after stage s_3
1	1	$P_1 + mP_2$	$P_1 + m(P_2+P_3) + m^2P_4$	$P_1 + m(P_2+P_3+P_7) + m^2(P_4+P_6+P_8) + m^3P_5$
	2	$P_2 + mP_1$	$P_3 + m(P_1+P_4) + m^2P_2$	$P_7 + m(P_1+P_6+P_8) + m^2(P_2+P_3+P_5) + m^3P_4$
2	1	$P_3 + mP_4$	$P_7 + m(P_6+P_8) + m^2P_5$	$P_6 + m(P_3+P_5+P_7) + m^2(P_1+P_4+P_8) + m^3P_2$
	2	$P_4 + mP_3$	$P_6 + m(P_5+P_7) + m^2P_8$	$P_3 + m(P_1+P_4+P_6) + m^2(P_2+P_5+P_7) + m^3P_8$
3	1	$P_6 + mP_5$	$P_2 + m(P_1+P_4) + m^2P_3$	$P_2 + m(P_1+P_4+P_5) + m^2(P_3+P_6+P_8) + m^3P_7$
	2	$P_5 + mP_6$	$P_4 + m(P_2+P_3) + m^2P_1$	$P_5 + m(P_2+P_6+P_8) + m^2(P_1+P_4+P_7) + m^3P_3$
4	1	$P_7 + mP_8$	$P_5 + m(P_6+P_8) + m^2P_7$	$P_4 + m(P_2+P_3+P_8) + m^2(P_1+P_5+P_7) + m^3P_6$
	2	$P_8 + mP_7$	$P_8 + m(P_5+P_7) + m^2P_6$	$P_8 + m(P_4+P_5+P_7) + m^2(P_2+P_3+P_6) + m^3P_1$

Table 2. Stage-by-stage crosstalk in the $\log_2 8$-1 optical switching fabric

OSE	output	after stage s_1	after stage s_2
1	1	$P_1 + m(P_2+P_3)$	$P_1 + m(P_2+P_3+P_4+P_6) + m^2(P_5+P_7+P_8)$
	2	$P_2 + m(P_1+P_3)$	$P_4 + m(P_1+P_5+P_6) + m^2(P_2+P_3+P_7+P_8)$
	3	$P_3 + m(P_1+P_2)$	$P_6 + m(P_1+P_4+P_7+P_8) + m^2(P_2+P_3+P_5)$
2	1	$P_4 + mP_5$	$P_2 + m(P_1+P_3+P_7) + m^2(P_4+P_5+P_6+P_8)$
	2	$m(P_4+P_5)$	$P_7 + m(P_2+P_6+P_8) + m^2(P_1+P_3+P_4+P_5)$
	3	$P_5 + mP_4$	**not present**
3	1	$P_6 + m(P_7+P_8)$	$P_3 + m(P_1+P_2+P_5+P_8) + m^2(P_4+P_6+P_7)$
	2	$P_7 + m(P_6+P_8)$	$P_5 + m(P_3+P_4+P_8) + m^2(P_1+P_2+P_6+P_7)$
	3	$P_8 + m(P_6+P_7)$	$P_8 + m(P_3+P_5+P_6+P_7) + m^2(P_1+P_2+P_4)$

appears at the input of the switching fabric, goes through an OSE and then at the output side of OSE it has power P_{out}. Because the OSE is build from SOAs, which compensates all losses in OSE (see section 2), it can be assumed that the power of the output signal is equal to the power of the input signal, that's why $P_{out} \approx P_{in}$. At another outputs, in this same OSE, it can appears noise denoted by $P_{noise} = mP_{in}$ (it is true only when one connection is established through this optical switching element). It allows to assign optical signal to crosstalk ratio OSXR [13], [14], [15] for the overall case at the output of one OSE:

$$OSXR\ network\ = 10\log_{10}\frac{P_{in}}{P_{noise}} \tag{1}$$

Generally, interstice between P_{in} and P_{noise} is $m = 0.01$. In such a situation equation (1) gives

$$OSXR\ network\ = 10\log_{10}\frac{P_{in}}{P_{noise}} = 10\log_{10}\frac{1}{m} = X \tag{2}$$

and then $X = 20dB$. Of course, if there is more than one connection set up through one OSE, the noise generated by the other connections influences to the first connection. The worst case (the maximum number of connections which can be set up in one optical switching element) at each OSE's output in the baseline and the $\log_2 N - 1$ networks of capacity $N = 8$ are shown in Table 1 and Table 2, respectively. For these same two type of networks and capacity $N = 16$ results are shown in Table 3 and Table 4, respectively. It is obvious that more stages in the switching architecture are the worst signal at the output of the whole switching network is. Serving this same functionality and for this same capacity the $\log_2 N - 1$ architecture has one stage less than a baseline switching network – it reflects from the way this structure is constructed [10]. It results also in OSXR (it can be clearly seen in Table 5).

As it can be seen in Table 1 in the worst case the first-, the second-, and the third-order crosstalk for the baseline network of capacity $N = 8$ is $3P$, $3P$, and P respectively. In turn, for the $\log_2 N - 1$ switching fabric of this same capacity in the worst case the first-, the second, and the third-order crosstalk is $4P$ (or $3P$ for some outputs), $3P$ (or $4P$ for some outputs), and 0, respectively (see Table 2 for details). It means, that the $\log_2 N - 1$ network has greater (or exactly the same) first-order crosstalk, exactly the same (or greater) second-order crosstalk, and has no third-order crosstalk. For capacity $N = 16$ the baseline network has $4P$, $6P$, and $4P$ the first-, the second, and the third-order crosstalk, respectively. In turn, the $\log_2 N - 1$ switching fabric of capacity $N = 16$ has $4P$, $5P$, and $2P$ the first-, the second-, and the third-order crosstalk, respectively. It means, that the $\log_2 N - 1$ structure has always equal or even smaller crosstalk (for an different order) than baseline network of the same capacity.

Table 3. Stage-by-stage crosstalk in the $\log_2 16$ optical switching fabric

OSE	output	after stage s_1	after stage s_2	after stage s_3	after stage s_4
1	1	$P_1 + mP_2$	$P_1 + m(P_2+P_4) + m^2P_3$	$P_1 + m(P_2+P_4+P_5) + m^2(P_3+P_6+P_7) + m^3P_8$	$P_1 + m(P_2+P_4+P_5+P_{11}) + m^2(P_3+P_6+P_7+P_{10}+P_{12}+P_{13}) + m^3(P_8+P_9+P_{14}+P_{15}) + m^4P_{16}$
	2	$P_2 + mP_1$	$P_4 + m(P_1+P_3) + m^2P_2$	$P_5 + m(P_1+P_6+P_7) + m^2(P_2+P_4+P_8) + m^3P_3$	$P_{11} + m(P_1+P_{10}+P_{12}+P_{13}) + m^2(P_2+P_4+P_5+P_9+P_{14}+P_{15}) + m^3(P_3+P_6+P_7+P_{16}) + m^4P_8$
2	1	$P_4 + mP_3$	$P_5 + m(P_6+P_7) + m^2P_8$	$P_{11} + m(P_{10}+P_{12}+P_{13}) + m^2(P_9+P_{14}+P_{15}) + m^3P_{16}$	$P_{13} + m(P_5+P_{11}+P_{14}+P_{15}) + m^2(P_1+P_6+P_7+P_{10}+P_{12}+P_{16}) + m^3(P_2+P_4+P_8+P_9) + m^4P_3$
	2	$P_3 + mP_4$	$P_7 + m(P_5+P_8) + m^2P_6$	$P_{13} + m(P_{11}+P_{14}+P_{15}) + m^2(P_{10}+P_{12}+P_{16}) + m^3P_9$	$P_5 + m(P_1+P_6+P_7+P_{13}) + m^2(P_2+P_4+P_8+P_{11}+P_{14}+P_{15}) + m^3(P_3+P_{10}+P_{12}+P_{16}) + m^4P_9$
3	1	$P_5 + mP_6$	$P_{11} + m(P_{10}+P_{12}) + m^2P_9$	$P_7 + m(P_4+P_5+P_8) + m^2(P_1+P_3+P_6) + m^3P_2$	$P_{10} + m(P_7+P_9+P_{11}+P_{15}) + m^2(P_4+P_5+P_8+P_{12}+P_{13}+P_{16}) + m^3(P_1+P_3+P_6+P_{14}) + m^4P_2$
	2	$P_6 + mP_5$	$P_{10} + m(P_9+P_{11}) + m^2P_{12}$	$P_4 + m(P_1+P_3+P_7) + m^2(P_2+P_5+P_8) + m^3P_6$	$P_7 + m(P_4+P_5+P_8+P_{10}) + m^2(P_1+P_3+P_6+P_9+P_{11}+P_{15}) + m^3(P_2+P_{12}+P_{13}+P_{16}) + m^4P_{14}$
4	1	$P_7 + mP_8$	$P_{13} + m(P_{14}+P_{15}) + m^2P_{16}$	$P_{10} + m(P_9+P_{11}+P_{15}) + m^2(P_{12}+P_{13}+P_{16}) + m^3P_{14}$	$P_4 + m(P_1+P_3+P_7+P_{15}) + m^2(P_2+P_5+P_8+P_{10}+P_{13}+P_{16}) + m^3(P_6+P_9+P_{11}+P_{14}) + m^4P_{12}$
	2	$P_8 + mP_7$	$P_{15} + m(P_{13}+P_{16}) + m^2P_{14}$	$P_{15} + m(P_{10}+P_{13}+P_{16}) + m^2(P_9+P_{11}+P_{14}) + m^3P_{12}$	$P_{15} + m(P_4+P_{10}+P_{13}+P_{16}) + m^2(P_1+P_3+P_7+P_9+P_{11}+P_{14}) + m^3(P_2+P_5+P_8+P_{12}) + m^4P_6$
5	1	$P_{10} + mP_9$	$P_3 + m(P_2+P_4) + m^2P_1$	$P_6 + m(P_3+P_5+P_8) + m^2(P_2+P_4+P_7) + m^3P_1$	$P_6 + m(P_3+P_5+P_8+P_{14}) + m^2(P_2+P_4+P_7+P_9+P_{13}+P_{16}) + m^3(P_1+P_{10}+P_{12}+P_{15}) + m^4P_{11}$
	2	$P_9 + mP_{10}$	$P_2 + m(P_1+P_3) + m^2P_4$	$P_3 + m(P_2+P_4+P_6) + m^2(P_1+P_5+P_8) + m^3P_7$	$P_{14} + m(P_6+P_9+P_{13}+P_{16}) + m^2(P_3+P_5+P_8+P_{10}+P_{12}+P_{15}) + m^3(P_2+P_4+P_7+P_{11}) + m^4P_1$
6	1	$P_{11} + mP_{12}$	$P_6 + m(P_5+P_8) + m^2P_7$	$P_{14} + m(P_9+P_{13}+P_{16}) + m^2(P_{10}+P_{12}+P_{15}) + m^3P_{11}$	$P_9 + m(P_3+P_{10}+P_{12}+P_{14}) + m^2(P_2+P_4+P_6+P_{11}+P_{13}+P_{16}) + m^3(P_1+P_5+P_8+P_{15}) + m^4P_7$
	2	$P_{12} + mP_{11}$	$P_8 + m(P_6+P_7) + m^2P_5$	$P_9 + m(P_{10}+P_{12}+P_{14}) + m^2(P_{11}+P_{13}+P_{16}) + m^3P_{15}$	$P_3 + m(P_2+P_4+P_6+P_9) + m^2(P_1+P_5+P_8+P_{10}+P_{12}+P_{14}) + m^3(P_7+P_{11}+P_{13}+P_{16}) + m^4P_{15}$
7	1	$P_{13} + mP_{14}$	$P_9 + m(P_{10}+P_{12}) + m^2P_{11}$	$P_2 + m(P_1+P_3+P_8) + m^2(P_4+P_6+P_7) + m^3P_5$	$P_2 + m(P_1+P_3+P_8+P_{16}) + m^2(P_4+P_6+P_7+P_{12}+P_{14}+P_{15}) + m^3(P_5+P_9+P_{11}+P_{13}) + m^4P_{10}$
	2	$P_{14} + mP_{13}$	$P_{12} + m(P_9+P_{11}) + m^2P_{10}$	$P_8 + m(P_2+P_6+P_7) + m^2(P_1+P_3+P_5) + m^3P_4$	$P_{16} + m(P_2+P_{12}+P_{14}+P_{15}) + m^2(P_1+P_3+P_8+P_9+P_{11}+P_{13}) + m^3(P_4+P_6+P_7+P_{10}) + m^4P_5$
8	1	$P_{15} + mP_{16}$	$P_{14} + m(P_{13}+P_{16}) + m^2P_{15}$	$P_{16} + m(P_{12}+P_{14}+P_{15}) + m^2(P_9+P_{11}+P_{13}) + m^3P_{10}$	$P_8 + m(P_2+P_6+P_7+P_{12}) + m^2(P_1+P_3+P_5+P_9+P_{11}+P_{16}) + m^3(P_4+P_{10}+P_{14}+P_{15}) + m^4P_{13}$
	2	$P_{16} + mP_{15}$	$P_{16} + m(P_{14}+P_{15}) + m^2P_{13}$	$P_{12} + m(P_9+P_{11}+P_{16}) + m^2(P_{10}+P_{14}+P_{15}) + m^3P_{13}$	$P_{12} + m(P_8+P_9+P_{11}+P_{16}) + m^2(P_2+P_6+P_7+P_{10}+P_{14}+P_{15}) + m^3(P_1+P_3+P_5+P_{13}) + m^4P_4$

The fourth-order crosstalk is just omitted because it's constitutes only a very small part of the output optical signal. Very often, however, the third-order (and sometimes also second-order) crosstalk is omitted too because it is much more smaller (order of magnitude) in relation to useful optical signal.

The OSXR at the output from a switching network is equal to the OSXR which appears at the OSE's output from the last stage in the switching fabric. In similar way equation (1) gives an OSXR for the optical signal measured in any place in a switching fabric – it should be noted that the right value of P_{in} and P_{noise} must be used. Power of an useful optical signal, which represents some connection in a switching structure, and optical noise could be found in Table 1, Table 2, Table 3, and Table 4.

In the Benes network the optical signal-to-crosstalk ration which appears at the output of a network is [13], [14]:

Table 4. Stage-by-stage crosstalk in the $\log_2 16$-1 optical switching fabric

OSE	output	after stage s_1	after stage s_2	after stage s_3
1	1	$P_1 + mP_2$	$P_1 + m(P_2+P_4+P_6) + m^2(P_5+P_7)$	$P_1 + m(P_2+P_4+P_6+P_9) + m^2(P_5+P_7+P_{10}+P_{12}+P_{14}) + m^3(P_{13}+P_{16})$
	2	$P_2 + mP_1$	$P_4 + m(P_1+P_5+P_6) + m^2(P_2+P_7)$	**not used**
	3	**not used**	$P_6 + m(P_1+P_4+P_7) + m^2(P_2+P_5)$	$P_9 + m(P_1+P_{10}+P_{12}+P_{14}) + m^2(P_2+P_4+P_6+P_{13}+P_{16}) + m^3(P_5+P_7)$
2	1	$P_4 + mP_5$	**not present**	$P_4 + m(P_1+P_5+P_6+P_{14}) + m^2(P_2+P_7+P_9+P_{12}+P_{16}) + m^3(P_{10}+P_{13})$
	2	$P_5 + mP_4$		$P_{14} + m(P_4+P_9+P_{12}+P_{16}) + m^2(P_1+P_5+P_6+P_{10}+P_{13}) + m^3(P_2+P_7)$
	3	**not present**		**not present**
3	1	$P_6 + mP_7$	$P_9 + m(P_{10}+P_{12}+P_{14}) + m^2(P_{13}+P_{16})$	$P_6 + m(P_1+P_4+P_7+P_{12}) + m^2(P_2+P_5+P_9+P_{13}+P_{14}) + m^3(P_{10}+P_{16})$
	2	$P_7 + mP_6$	$P_{14} + m(P_9+P_{12}+P_{16}) + m^2(P_{10}+P_{13})$	**not used**
	3	**not used**	$P_{12} + m(P_9+P_{13}+P_{14}) + m^2(P_{10}+P_{16})$	$P_{12} + m(P_6+P_9+P_{13}+P_{14}) + m^2(P_1+P_4+P_7+P_{10}+P_{16}) + m^3(P_2+P_5)$
4	1	$P_9 + mP_{10}$	$P_2 + m(P_1+P_5+P_7) + m^2(P_4+P_6)$	$P_2 + m(P_1+P_5+P_7+P_{10}) + m^2(P_4+P_6+P_9+P_{13}+P_{16}) + m^3(P_{12}+P_{14})$
	2	$P_{10} + mP_9$	$P_5 + m(P_2+P_4+P_7) + m^2(P_1+P_6)$	**not used**
	3	**not used**	$P_7 + m(P_2+P_5+P_6) + m^2(P_1+P_4)$	$P_{10} + m(P_2+P_9+P_{13}+P_{16}) + m^2(P_1+P_5+P_7+P_{12}+P_{14}) + m^3(P_4+P_6)$
5	1	$P_{12} + mP_{13}$	**not present**	$P_5 + m(P_2+P_4+P_7+P_{16}) + m^2(P_1+P_6+P_{10}+P_{13}+P_{14}) + m^3(P_9+P_{12})$
	2	$P_{13} + mP_{12}$		$P_{16} + m(P_5+P_{10}+P_{13}+P_{14}) + m^2(P_2+P_4+P_7+P_9+P_{12}) + m^3(P_1+P_6)$
	3	**not present**		**not present**
6	1	$P_{14} + mP_{16}$	$P_{10} + m(P_9+P_{13}+P_{16}) + m^2(P_{12}+P_{14})$	$P_7 + m(P_2+P_5+P_6+P_{13}) + m^2(P_1+P_4+P_{10}+P_{12}+P_{16}) + m^3(P_9+P_{14})$
	2	**not used**	$P_{16} + m(P_{10}+P_{13}+P_{14}) + m^2(P_9+P_{12})$	**not used**
	3	$P_{16} + mP_{14}$	$P_{13} + m(P_{10}+P_{12}+P_{16}) + m^2(P_9+P_{14})$	$P_{13} + m(P_7+P_{10}+P_{12}+P_{16}) + m^2(P_2+P_5+P_6+P_9+P_{14}) + m^3(P_1+P_4)$

$$OSXR\ Benes\ =\ X\ -10\log_{10}\ 2\log_2 N-1 \tag{3}$$

Table 5. The OSXR, the number of stages, passive and active optical elements through which goes connection in the worst (best) case

N	**network**	**stages**	**1×2 splitters**	**2×1 combiners**	**SOAs**	**OSXR [dB]**
8	Benes	5	5	5	5	13.0103
	log_2N	3	3	3	3	15.2288
	log_2N-1	2	4 (3)	4 (3)	2	13.9794
16	Benes	7	7	7	7	11.5490
	log_2N	4	4	4	4	13.9794
	log_2N-1	3	5 (4)	5 (4)	3	13.9794
32	Benes	9	9	9	9	10.4576
	log_2N	5	5	5	5	13.0103
	log_2N-1	4	6 (5)	6 (5)	4	13.9794
64	Benes	11	11	11	11	9.5861
	log_2N	6	6	6	6	12.2185
	log_2N-1	5	7 (6)	7 (6)	5	13.0103
128	Benes	13	13	13	13	8.8606
	log_2N	7	7	7	7	11.5490
	log_2N-1	6	8 (7)	8 (7)	6	12.2185
256	Benes	15	15	15	15	8.2391
	log_2N	8	8	8	8	10.9691
	log_2N-1	7	9 (8)	9 (8)	7	11.5490
512	Benes	17	17	17	17	7.6955
	log_2N	9	9	9	9	10.4576
	log_2N-1	8	10 (9)	10 (9)	8	10.9691
1024	Benes	19	19	19	19	7.2125
	log_2N	10	10	10	10	10.0000
	log_2N-1	9	11 (10)	11 (10)	9	10.4576
2048	Benes	21	21	21	21	6.7778
	log_2N	11	11	11	11	9.5861
	log_2N-1	10	12 (11)	12 (11)	10	10.0000
4096	Benes	23	23	23	23	6.3827
	log_2N	12	12	12	12	9.2082
	log_2N-1	11	13 (12)	13 (12)	11	9.5861
8192	Benes	25	25	25	25	6.0206
	log_2N	13	13	13	13	8.8606
	log_2N-1	12	14 (13)	14 (13)	12	9.2082

For the baseline network of capacity N the OSXR at the output of the switching network is given by the following expression:

$$OSXR\ \log_2 N\ =\ X\ -10\log_{10}(\log_2 N). \tag{4}$$

In the $\log_2 N-1$ switching network there are two cases for OSXR given by the following expression:

$$OSXR\ \log_2 N-1\ =\ \begin{matrix} X\ -10\log_{10} 4 & for\ N=8,16; \\ X\ -10\log_{10}(\log_2 N-1) & for\ N\geq 32. \end{matrix} \tag{5}$$

These two cases reflect strictly from the characteristic architecture of the $\log_2 N-1$ switching fabric [10].

In Table 5 and Figure 2 it can be found comparison of an OSXR for different structures and capacities. The $\log_2 N-1$ switching fabric for capacity greater or equal to 16 has always better useful output optical signal power than other structures. What's more, all connections go through one active optical element less.

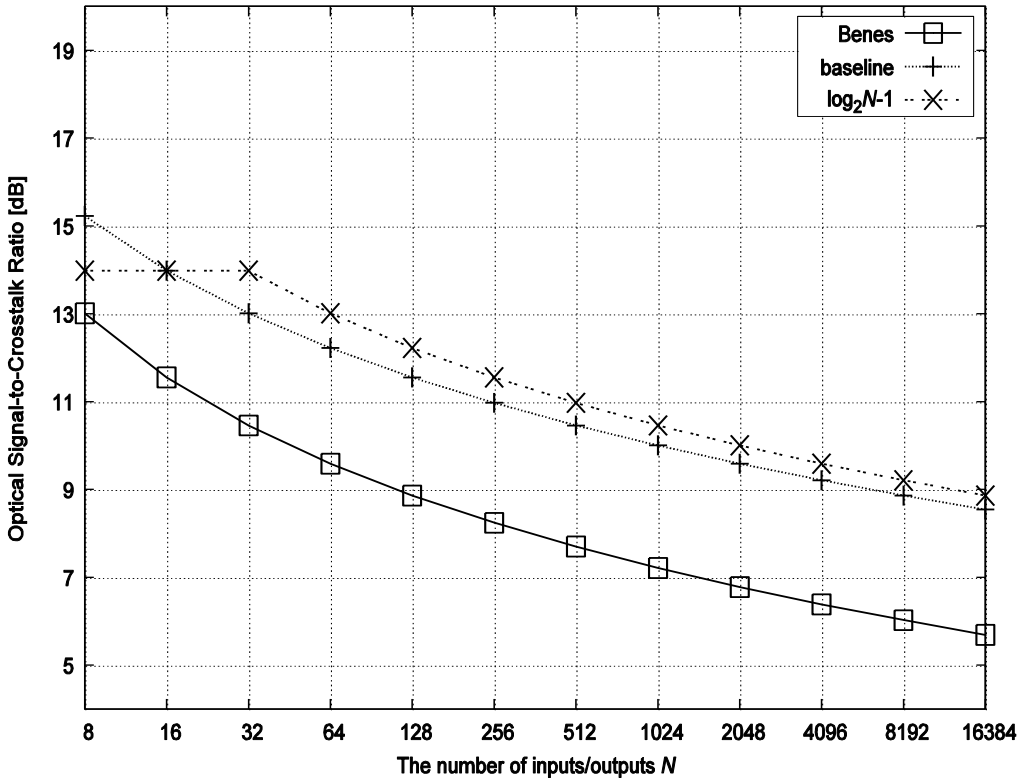

Fig. 2. Optical Signal-to-Crosstalk Ratio

3. CONCLUSION

In this paper the crosstalk and its influence to the optical signal was described. It was considered first-, second-, and third-order crosstalk in the baseline and the $\log_2 N - 1$ switching networks. It was also calculated and compared OSXR for two mentioned above networks and for the Benes network, too. In results, the $\log_2 N - 1$ switching fabric, proposed in [10], gives better OSXR and the optical signal goes through fewer number of active optical elements than in the baseline network of the same capacity and functionality.

REFERENCES

[1] GOKE L. R., LIPOVSKI G. J., *Banyan networks for partitioning multiprocessor system*, In: Proceedings ISCA, 1973, 21-28.
[2] LEA C.-T., *Multi-$log_2 N$ networks and their applications in high-speed electronic and photonic switching systems*, IEEE Transactions on Communications, Vol. 38, 1990, 1740-1749.
[3] LEA C.-T., *Buffered or unbuffered: a case study based on $log_2(N, e, p)$ networks*, IEEE Transactions on Communications, Vol. 44, No. 1, 1996, 105-113.
[4] MELEN R., TURNER J. S., *Nonblocking multirate networks*, SIAM Journal on Computing, Vol. 18, No. 2, 1989, 301-313.
[5] CHEUNG S.-P., ROSS K. W., *On nonblocking multirate interconnection networks*, SIAM Journal on Computing, Vol. 20, No. 4, 1991, 726-736.
[6] MELEN R., TURNER J. S., *Nonblocking multirate distribution networks*, In: Proceedings IEEE INFOCOM, Vol. 3, 1990, 1234-1241.
[7] PATTAVINA A., *Switching Theory – Architectures and performance in broadband ATM networks*, England: John Wiley & Sons, 1998.
[8] HWANG F. K., *The mathematical theory of nonblocking switching networks*, 2nd edition, Singapore: World Scientific, 2004.
[9] KABACINSKI W., *Nonblocking electronic and photonic switching fabrics*, Boston/London: Kluwer Academic Publisher, 2005.
[10] DANILEWICZ G., KABACINSKI W., RAJEWSKI R., *The $log_2 N - 1$ optical switching fabrics*, IEEE Transactions on Communications, Vol. 59, No. 1, 2011, 213-225.
[11] CONNELLY M. J., *Semiconductor optical amplifiers*, Kluwer Academic Publisher, 2004.
[12] EL-BAWAB T. S., *Optical switching*, Springer, 2006.
[13] LU C.-C., THOMPSON R. A., *The double-layer network architecture for photonic switching*, IEEE Transactions on Lightwave Technology, Vol. 12, No. 8, 1994, 1482-1489.
[14] KABACINSKI W., *Modified dilated Benes networks for photonic switching*, IEEE Transactions on Communications, Vol. 47, No. 8, 1999, 1253-1259.
[15] DANILEWICZ G., KABACINSKI W., ZAL M., *Reduced Banyan-type multiplane rearrangeable switching networks*, IEEE Communications Letters, Vol. 12, No. 12, 2008.

BIBLIOTEKA INFORMATYKI SZKÓŁ WYŻSZYCH

Information Systems Architecture and Technology. Web Information Systems: Models, Concepts & Challenges, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Information Systems and Computer Communication Networks, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Models of the Organisations Risk Management, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2008

Information Systems Architecture and Technology. Designing, Development and Implementation of Information Systems, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Model Based Decisions, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2008

Information Systems Architecture and Technology. Advances in Web-Age Information Systems, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2009

Information Systems Architecture and Technology. Service Oriented Distributed Systems: Concepts and Infrastructure, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2009

Information Systems Architecture and Technology. Systems Analysis in Decision Aided Problems, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2009

Information Systems Architecture and Technology. IT Technologies in Knowledge Oriented Management Process, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2009

Information Systems Architecture and Technology. New Developments in Web-Age Information Systems, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2010

Information Systems Architecture and Technology. Networks and Networks Services', pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2010

Information Systems Architecture and Technology. System Analysis Approach to the Design, Control and Decision Support, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2010

Information Systems Architecture and Technology. IT TModels in Management Process, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2010

Information Systems Architecture and Technology. Web Information Systems Engineering, Knowledge Discovery and Hybrid Computing, pod redakcją Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2011

Information Systems Architecture and Technology. Service Oriented Networked Systems, pod redakcją Adama GRZECHA, Leszka BORZEMSKIEGO, Jerzego ŚWIĄTKA, Zofii WILIMOWSKIEJ, Wrocław 2011

Information Systems Architecture and Technology. System Analysis Approach to the Design, Control and Decision Support, pod redakcją Jerzego ŚWIĄTKA, Leszka BORZEMSKIEGO, Adama GRZECHA, Zofii WILIMOWSKIEJ, Wrocław 2011

Information Systems Architecture and Technology. Information as the Intangible Assets and Company Value Source, pod redakcją Zofii WILIMOWSKIEJ, Leszka BORZEMSKIEGO, Adama GRZECHA, Jerzego ŚWIĄTKA, Wrocław 2011

Wydawnictwa Politechniki Wrocławskiej
są do nabycia w księgarni „Tech”
plac Grunwaldzki 13, 50-377 Wrocław
budynek D-1 PWr., tel. 71 320 29 35
Prowadzimy sprzedaż wysyłkową
zamawianie.ksiazek@pwr.wroc.pl

ISBN 978-83-7493-703-0