Simulation Model for Application of the SDN Concept in IMS/NGN Network Transport Stratum

Sylwester Kaczmarek[1][0000-0003-2932-5610], Maciej Sac[1][0000-0002-6734-3046] and Jakub Adrych[2]

[1] Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Narutowicza 11/12, 80-233 Gdańsk, Poland
[2] ASH-TECH, Wiosenna 2, 86-065 Łochowo, Poland
`{kasyl,Maciej.Sac}@eti.pg.edu.pl, jakub.adrych99@gmail.com`

Abstract. The paper presents a simulation model allowing examination of cooperation between two currently used telecommunication networks concepts: IP Multimedia Subsystem/Next Generation Network (IMS/NGN) and Software-Defined Networking (SDN). Application of the SDN architecture elements in IMS/NGN networks will enable unified control and management of transport resources for various transport technologies and equipment manufacturers. However, such a cooperation is a new concept requiring verification, which is the aim of this paper. The structure of the modeled multidomain network and details about the simulator operation are described. Tests proving correctness of its operation are carried out. Selected research results regarding mean Call Set-up Delay and mean Call Disengagement Delay in the considered network are presented demonstrating that the cooperation between IMS/NGN and SDN is possible.

Keywords: SDN, NGN, IMS, simulation model, call processing performance.

1 Introduction

Growing demand for higher bit rates, development of the Internet of Things (IoT) concept, multimedia services and continuous progress in network architecture solutions (including 5G mobile network) require advanced research to guarantee Quality of Service (QoS) and efficiency of the proposed solutions.

The currently used Next Generation Network (NGN) [1] concept consists of service stratum with applications and transport stratum performing transmission and switching functions. NGN service stratum is based on the servers defined in the IP Multimedia Subsystem (IMS) [2] concept. Hence the name "IMS/NGN" is commonly used. In NGN transport stratum any technology supporting transmission of IP packets is allowed.

The concept of Software-Defined Networking (SDN) [3,4] was proposed primarily due to the diversity of telecommunications equipment related to transport stratum and the need to automate resource management and traffic control in this stratum. Recently, these features have become particularly important due to the Covid-19 pandemic, which

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

was associated with a huge amount of new and dynamically changing traffic (concerning distance learning, remote work, remote handling of official matters, etc.).

Due to these aspects, it is highly desirable to apply the SDN concept in existing telecommunications network architectures, such as the IMS/NGN network. However, the cooperation of the above-mentioned solutions has not been standardized or verified by the scientific community. This led the authors of this paper to address this problem. Due to complexity of the IMS/NGN network structure (especially for a multidomain network) and service scenarios as well as the desire to precisely describe the interaction of this network with the SDN concept, the authors decided to use the methods of computational science and develop a simulation model. The implemented simulator allows evaluation of mean Call Set-up Delay (mean CSD, $E(CSD)$) and mean Call Disengagement Delay (mean CDD, $E(CDD)$) for all types of successful call scenarios performed in a multidomain IMS/NGN network cooperating with the SDN concept. These parameters belong to a set of standardized Call Processing Performance (CPP) parameters [5,6] describing control performance in a telecommunications network, which is important for both network operators and users.

The rest of the paper is organized as follows. A review of the related work is provided in section 2. Section 3 describes the assumptions regarding the modeled network structure and service scenarios. It also contains details about the structure and operation of the implemented simulation model. Section 4 is devoted to functional tests of the simulator. It also provides selected obtained results. Section 5 concludes the paper.

2 Related Work

The aim of the performed review was to find other works related to the topic of this paper, i.e. regarding simulation models for the IMS/NGN network cooperating with the SDN concept. As already mentioned, such cooperation is not described in standardization documents. During the review, emphasis was placed on finding solutions that take into account both the NGN network service stratum based on the IMS concept and the transport stratum using the SDN architecture. Unfortunately, no such works were found, even when cooperation of pure IMS architecture with SDN was considered, without including NGN standards.

The review indicated that academic and industrial communities focus on practical verification of cooperation between IMS/NGN network and the SDN concept (testbeds). The available works can be divided into the following categories:

- proposals for network architecture and service scenarios, but without their implementation [7],
- practical implementations (testbeds) focusing on resource control using SDN (without taking into account the service stratum) [8],
- practical implementations (testbeds) of cooperation between pure IMS and SDN, without considering NGN standards [9-12].

A common feature of the above-mentioned works is the lack of consideration of ITU-T standards for NGN networks (e.g. related to the RACF resource control unit), as well

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

as the lack of access to the source code, which prevents further work on these projects. Therefore, the authors of this paper decided to address these issues and create their own testbed for IMS/NGN network using the SDN concept [13], which does not have the disadvantages of the other mentioned solutions.

The advantage of testbeds is the possibility of practical verification of selected issues related to network operation in a laboratory environment. However, the problem is testing more complex network structures (e.g. a multidomain network) and service scenarios. For this purpose, the best solutions are simulation models that are flexible in configuration and reflect phenomena occurring in real networks. Therefore, the implemented testbed [13] was used to confirm the assumed message exchange procedures and provide the data (among others message lengths and processing times) necessary to develop a simulation model for a multi-domain IMS/NGN network based on the SDN concept. This model is described in the next section.

3 Simulation Model

In this paper, a multidomain IMS/NGN network belonging to two operators is assumed (Fig. 1). Each operator has its own IMS/NGN domain, therefore, the terms domain and operator will be used interchangeably with a similar meaning. Each domain includes elements of service stratum and transport stratum. Their names are appended with domain numbers (1 and 2).

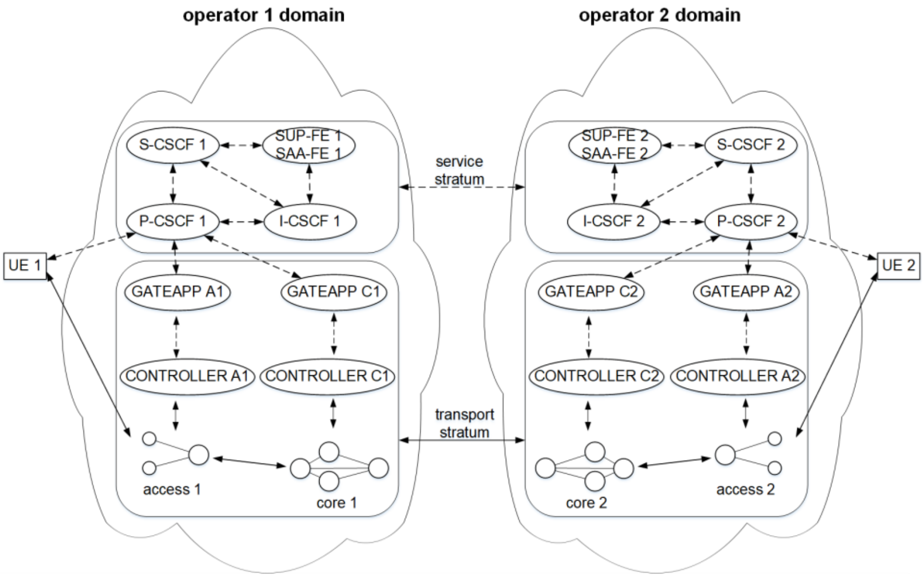


Fig. 1. Structure of the modeled multidomain IMS/NGN network utilizing the SDN concept. Physical connections are marked using solid lines with arrows, logical connections – dashed lines with arrows.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

Service stratum is responsible for controlling the process of delivering services to users via their terminals called User Equipment (UE). This stratum uses mainly SIP [14] and Diameter [15] communication protocols. It is based on the elements taken from the IMS concept, such as:

- Call Session Control Function (CSCF) servers: Proxy-CSCF (P-CSCF, they exchange messages with UEs), Serving-CSCF (S-CSCF, the main servers handling all calls) and Interrogating-CSCF (I-CSCF, servers used in multidomain calls),
- Service User Profile Functional Entity/Service Authentication and Authorization Functional Entity (SUP-FE/SAA-FE): the database storing information about user location and subscription of services; used for, among others, authentication, authorization and accounting.

The role of IMS/NGN transport stratum is to provide resources necessary for the services requested by the users. It is assumed that transport stratum of each operator is based on the SDN architecture and includes an access and a core network. The resources of these networks (programmable switches) are managed by separate SDN controllers. “A” and “C” letters, which are used in transport stratum elements’ names, indicate the network type, e.g. CONTROLLER A1 manages resources of access network in domain 1. The SDN concept unifies the protocol (the OpenFlow protocol [16]) used for resource control and management for different technologies and equipment. As a result, both classical packet networks and optical networks can be applied in IMS/NGN and managed in a unified way.

Consequently, the cooperation of IMS/NGN network and the SDN concept can bring many benefits. However, it has not been standardized. In the ITU-T standards for IMS/NGN networks [17], the P-CSCF server generates resource reservation and release requests to the transport resource control unit (called the Resource and Admission Control Function, RACF) using the Diameter protocol. In the SDN concept applications determine the required resource operations and use the API of a given controller, which adds, modifies or deletes entries in flow tables of programmable switches. To ensure IMS/NGN and SDN interoperability, it is therefore necessary to add an additional element (called the Gateway Application or simpler Gateapp) that translates messages generated by P-CSCF to these expected by the SDN controller and vice versa. It is important that SDN controller API is not standardized and depends on controller implementation. For the purpose of this work, one of the most popular API solutions was assumed (HTTP [18] REST API), which is used, e. g., in the ONOS controller [19].

Sixteen different service scenarios are assumed in the modeled network, including user registration as well as voice calls within one or two domains. Intra-domain calls may involve one or multiple access areas. The first case requires resource reservation only in the access network of a given domain, the second – also in the core network of a given domain. Inter-domain calls require resource reservation in the access and core networks of both domains. All the above-mentioned scenarios can be generated in domain 1 and 2. For voice calls, possibility of lack of resources in individual transport networks is taken into account, resulting in unsuccessful scenarios. Fig. 2 presents a message exchange scenario for a successful intra-operator call performed in domain 1 with both UEs connected to the same access areas.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

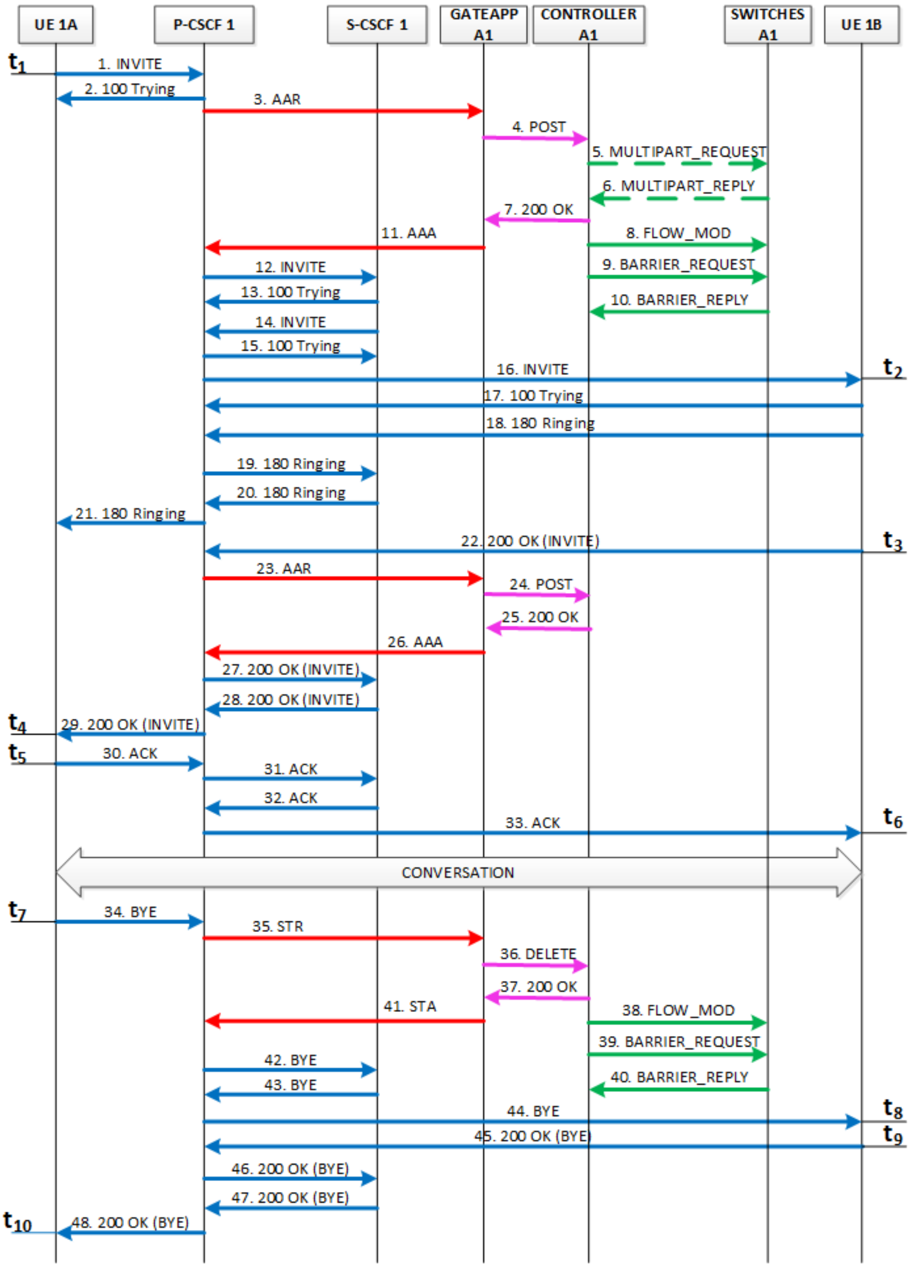


Fig. 2. Message exchange scenario for a successful intra-operator call performed in domain 1 with both UEs connected to the same access areas. Different colors are used for different signaling protocols: SIP – blue, Diameter – red, HTTP – pink, OpenFlow – green.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

To increase readability of this scenario, the user terminal generating call set-up request (UE 1A) and the one receiving this request (UE 1B) are represented by separate blocks. In Fig. 1, one block (UE 1) represents all terminals connected to the network of operator 1.

The scenario depicted in Fig. 2 starts with sending a call set-up request (message 1) from UE 1A to P-CSCF 1, which is confirmed by P-CSCF 1 (message 2). P-CSCF 1 sends an AAR message (message 3) to GATEAPP A1 to reserve transport resources of access 1 network for the requested call. This message is translated to HTTP POST request (message 4) and sent to CONTROLLER A1. If necessary, optional messages 5-6 are send by the SDN controller to all programmable switches on path (SWITCHES A1), to determine availability of resources for the requested service. If the controller already has the knowledge about the necessary resources, the scenario goes to messages 7-10. Message 7 is the confirmation about transport resource availability send to GATEAPP A1 and messages 8-10 perform proper changes in flow tables of all programmable switches on path. GATEAPP A1 translates message 7 to message 11 and sends it to P-CSCF 1. After that, the call set-up request is sent over the network (to S-CSCF 1 and P-CSCF 1 again) until it reaches UE 1B (messages 12-17). UE 1B rings (messages 18-21) and accepts the call (message 22). At this time messages 23-26 are used to confirm that the previously reserved resources will be used and traffic regarding the requested call will be sent. This step does not require communication between the SDN controller and programmable switches. Subsequently, the 200 OK (INVITE) message is sent over the network to UE 1A (messages 27-29), which confirms its receipt to UE 1B (messages 30-33). This starts a conversation between both end users, which is ended by UE 1A by sending a BYE message to P-CSCF 1 (message 34). This results in communication between P-CSCF 1, GATEAP A1, CONTROLLER A1 and SWITCHES A1 for releasing the resources allocated to the disengaged call (messages 35-41). After that the BYE message is sent from P-CSCF 1 to UE 1B (through S-CSCF 1 and P-CSCF 1 again; messages 42-44), which confirms call disengagement (messages 45-48).

A very important aspect are the t1-t10 times marked in Fig. 2. According to the ITU-T definitions [5,6], they can be used to calculate Call Set-up Delay (CSD) and Call Disengagement Delay (CDD):

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

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

Values of *CSD* and *CDD* concern one voice call. In our simulation model they are gathered separately for all types successful call scenarios performed in the modeled network and then averaged. The considered scenarios include:

- successful intra-operator calls performed in domain 1 and 2 with both UEs connected to the same access areas (scenarios b1 and b2),
- successful intra-operator calls performed in domain 1 and 2 with both UEs connected to different access areas (scenarios d1 and d2),
- successful inter-operator calls originated in domain 1 and 2 (scenarios f1 and f2).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

Message exchange for the b1 scenario is presented in Fig. 2. The b2 scenario is analogous but performed in domain 2. The d1 scenario involves multiple access areas. Thus, comparing to the b1 case, resource reservation and release in core network of operator 1 are necessary. These procedures are similar to those presented in Fig. 2 (messages 3-11 and 35-41). The most complex are successful inter-operator call scenarios (f1 and f2), requiring resource reservation in access and core transport networks in both domains. Due to lack of space they are not provided. Message flow for the f1 scenario for a multidomain IMS/NGN network without cooperation with SDN can be found in [20].

Structure of the implemented discrete-event simulation model software is presented in Fig. 3. The simulator was developed in the OMNeT++ simulation framework [21] by thoroughly extending our previous model regarding IMS/NGN service stratum [22]. This required adding new modules related to SDN-based transport stratum (GATEAPP, CONTROLLER, SWITCH) and including them in network operation logic.

To increase readability of Fig. 3, only elements of domain 1 and the “global” module common to the entire simulated network (providing global variables, collecting partial simulation results and calculating final results, providing diagnostic functions) are included. The structure of domain 2 is a mirror image of domain 1. Figure 3 includes two programmable switches for access network and core network, as this configuration will be further used in research (section 4).

Blue, red, pink, and green lines with arrows indicate communication between different network elements. The meaning of individual colors is the same as in Fig. 2. Black lines with arrows are used to indicate internal communication between simulation modules forming particular network elements. Italic font in Fig. 3 is used to denote the names of simple modules that perform basic operations defined in the C++ language. Simple modules are grouped into compound modules (bold font), which act as elements of the simulated network (Figs. 1-2). For each compound module, the name of the implemented network element and the name of the compound module itself (in brackets) are given. Individual compound modules (e.g. SWITCH) are used many times to implement the operation of various network elements, which is determined by the parameters provided during their initialization.

The implemented compound modules have a common structure. They consist of a “*_main” simple module and a set of “l2tran” simple modules. “*_main” simple modules constitute the main logic of the simulator. They are responsible for handling messages received from other modules and so-called selfmessages (generated by the same module and used to manage simulation events). As a result of handling, received messages may be delayed, their fields may be changed, and new messages may be generated depending on the implemented service scenarios. Most “*_main” simple modules implement a queue and a message processor. The exceptions are the “ue_main” and “supfe_saafe_main” modules, which respond to each received message with a certain delay (without queuing). “l2tran” simple modules participate in communication between elements of the modeled network. Their number in each compound module is equal to the number of other compound modules with which it communicates. The role of each “l2tran” simple module is to buffer messages outgoing from a given compound module if communication link is busy. Messages incoming to a given compound module pass through the “l2tran” modules transparently.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

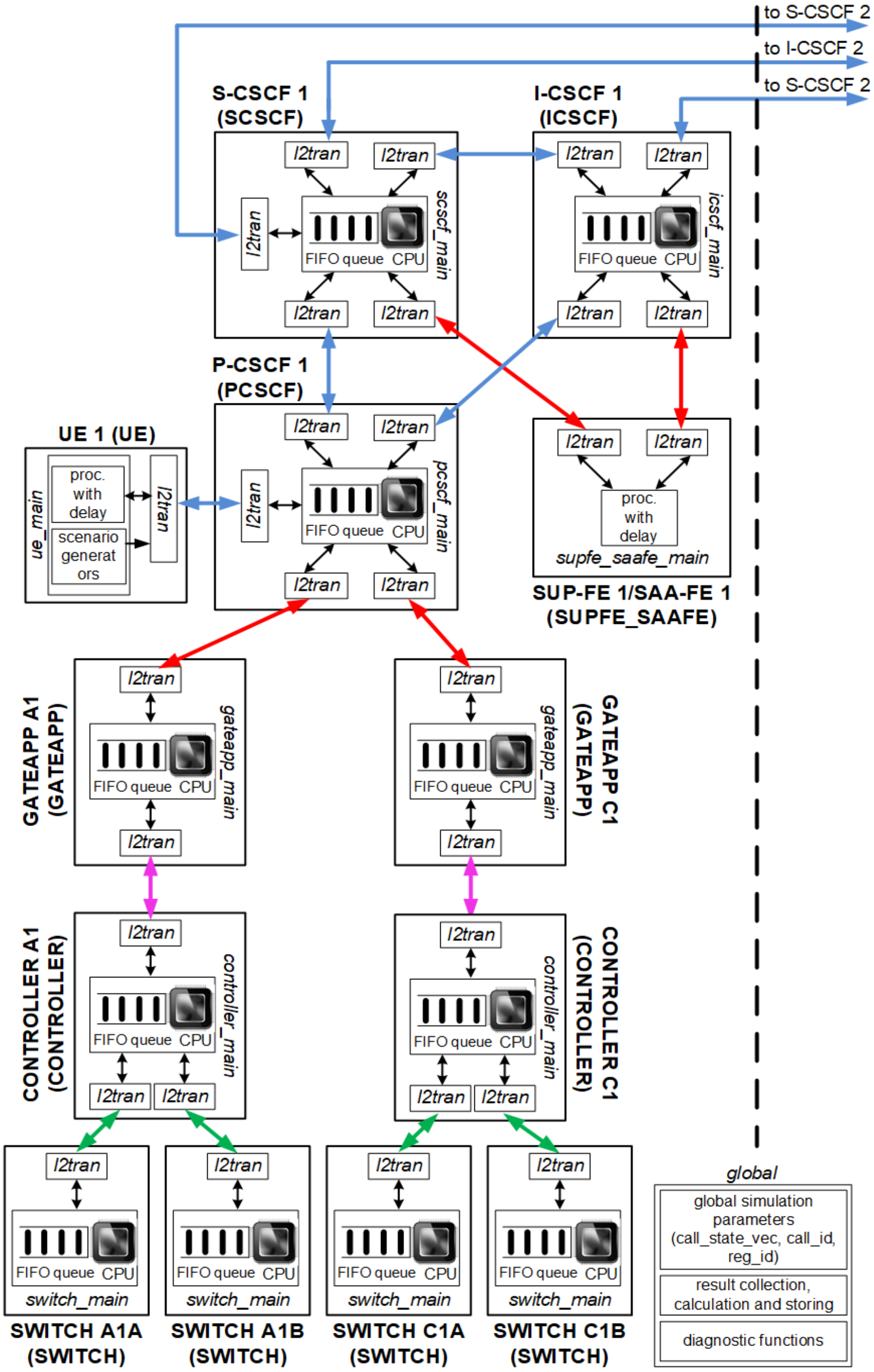


Fig. 3. Structure of the developed simulation model (domain 1).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

In the developed simulation model, messages are sent between the elements of the modeled network in accordance with service scenarios for IMS/NGN and SDN networks. The sent messages contain fields defined in standardization documents. Modules process messages (they among others include delays) similarly to elements of a real network. Therefore, the implemented simulator reflects the phenomena occurring in real network.

UE modules in particular domains (UE 1 and UE 2) are responsible for generating call set-up and registration requests. All exchanged messages, apart from the fields defined by the standards, contain fields for carrying information about the t_1-t_{10} times necessary for calculation of CSD (1) and CDD (2) parameters. When a message passes through subsequent network elements in accordance with the given scenario, these fields are appropriately filled with the values of current simulation time. When a given call is set-up and disengaged correctly (scenarios b1, b2, d1, d2, f1, f2), all t_1-t_{10} times are filled, making it possible to calculate CSD and CDD values for this call.

The final simulation results are mean CSD ($E(CSD)$) and mean CDD ($E(CDD)$) values obtained separately for all successful call scenarios along with corresponding confidence intervals. Scenarios' names are added as indexes to these output variables, e.g. $E(CSD)_{\mathrm{b1}}$. To obtain final results simulation time is divided into one warm-up period (related to the achievement of a steady state by the simulated network, it is not taken into account when analyzing the results) and a definable number of measurement periods. The Student's t-distribution is used to determine confidence intervals when processing measurement data obtained from the simulations. It is used in situations where the standard deviation of the population is unknown and estimated on the basis of measurements and the number of measurement periods is relatively small (below 10).

The simulator provides three possible conditions for ending simulations: exceeding a given simulation time, generating a given number of call set-up requests, reaching values of confidence intervals below a given threshold (this condition is checked periodically). All aspects of simulations are configurable using an *.ini file. The most important input variables of the implemented simulator are presented in Tab. 1.

Table 1. The most important input variables of the implemented simulator.

Variable name	Description	Default value
sim-time-limit	Maximum simulation duration time [s].	36000
warmup-period	Warp-up period duration time [s].	1250
call_num_max	Maximum number of generated calls.	1000000000
meas_per_num	Number of measurement periods.	5
conf_level	Confidence level.	0.95
conf_interv_max	Threshold value for confidence intervals, may be directly given in [s] or set in relation to the obtained mean values (in [%]).	5%
delay	Base delay value in [s] defined separately for all modeled network elements. For SUPFE_SAAFE compound modules it applies to all messages. For	0.001

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

	other modules it concerns a base message (processing time of other messages is proportional according to the ak variable, which is described below). The base messages are as follows: SIP INVITE for CSCF servers and UEs, Diameter AAR for GATEAPP modules, HTTP POST for CONTROLLER modules, FLOW_MOD for SWITCH modules.	
ak	Vector describing how long other messages are processed in particular modules in relation to the base messages. It contains 32 values.	Default values of ak and mess_length are not provided due space limitations
mess_length	Vector with message lengths in [B], containing 39 values.	
link_datarate	Link bandwidth in [bps], can be defined separately for all links.	50000000
link_length	Link length in [m], can be defined separately for all links.	200000
res_info_prob	Probability of controller having information about the resource state in programmable switches so that there is no need to send messages 5-6 from Fig. 2. This value can be defined separately for each controller.	0.7
res_unav_prob	Probability of resource unavailability in particular programmable switch. This value can be defined separately for each switch.	0
intrad_call_intensity	Intra-operator call set-up request intensity in [1/s], can be defined separately for each domain.	50
interd_call_intensity	Inter-operator call set-up request intensity in [1/s], can be defined separately for each domain.	50
registr_intensity	Registration request intensity in [1/s], can be defined separately for each domain.	50
multiple_access_areas _ratio	Ratio of intra-operator calls concerning multiple access areas to all generated intra-operator calls, can be defined separately for each domain.	0.5

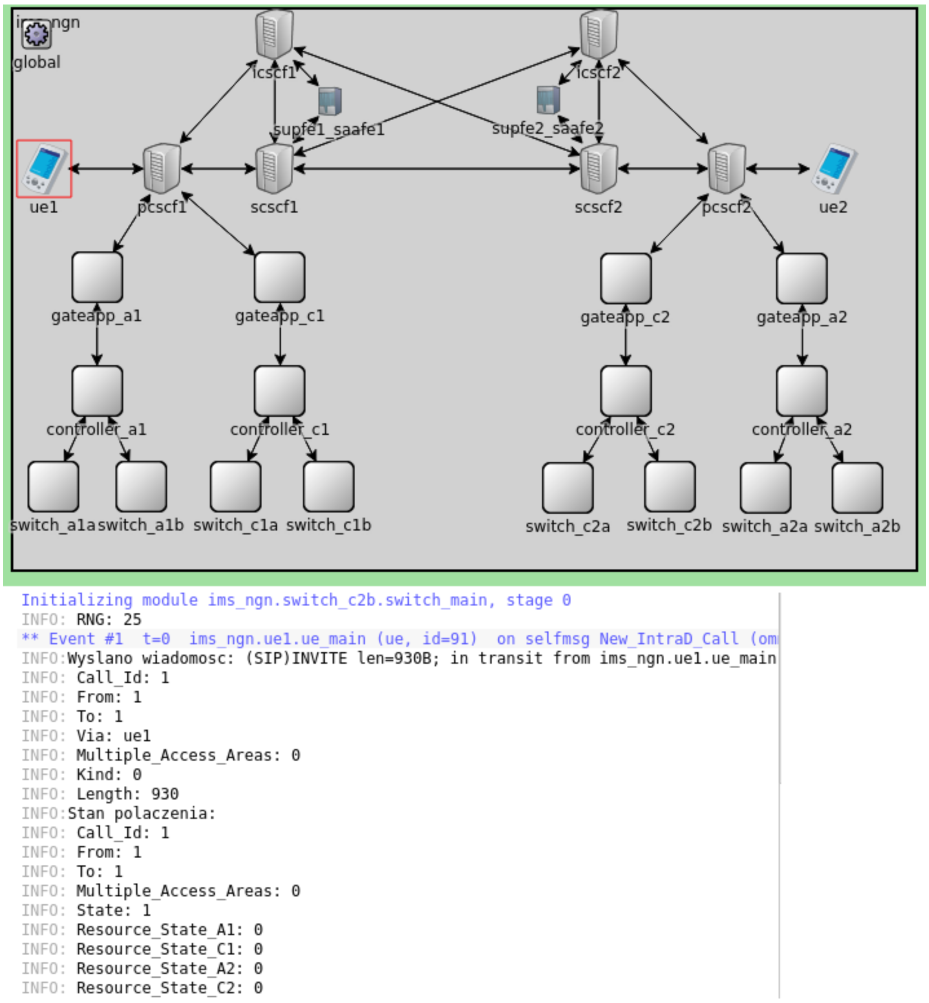
4 Tests and Results

Verification of correctness of the simulation model was carried out in several stages. The first tests were performed while extending the previous version of the simulator [22] to the current version supporting the SDN architecture. Each time a coherent piece of software was developed (for example, a function within a module), the written code

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

was carefully analyzed for errors and tested using different sets of input data. The results of the tested software fragments were displayed in the simulator console. If necessary, corrections were made to the code and tests were repeated until correct operation was achieved.

After implementing the target simulator software, final tests were performed by running simulations in graphical mode (Fig. 4) and analyzing simulations logs using the Sequence Chart Tool built into OMNeT++ (Fig. 5). These steps made it possible to check correctness of implementation of all service scenarios. For this reason the simulator was configured in such a way that particular scenarios did not overlap and could be analyzed separately.



```
Initializing module ims_ngn.switch_c2b.switch_main, stage 0
INFO: RNG: 25
** Event #1  t=0  ims_ngn.ue1.ue_main (ue, id=91)  on selfmsg New_IntraD_Call (om
INFO:Wyslano wiadomosc: (SIP)INVITE len=930B; in transit from ims_ngn.ue1.ue_main
INFO: Call_Id: 1
INFO: From: 1
INFO: To: 1
INFO: Via: ue1
INFO: Multiple_Access_Areas: 0
INFO: Kind: 0
INFO: Length: 930
INFO:Stan polaczenia:
INFO: Call_Id: 1
INFO: From: 1
INFO: To: 1
INFO: Multiple_Access_Areas: 0
INFO: State: 1
INFO: Resource_State_A1: 0
INFO: Resource_State_C1: 0
INFO: Resource_State_A2: 0
INFO: Resource_State_C2: 0
```

Fig. 4. Graphical simulation mode (a fragment of window) – structure of the simulated network (up) and events occurring during the simulation (down).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

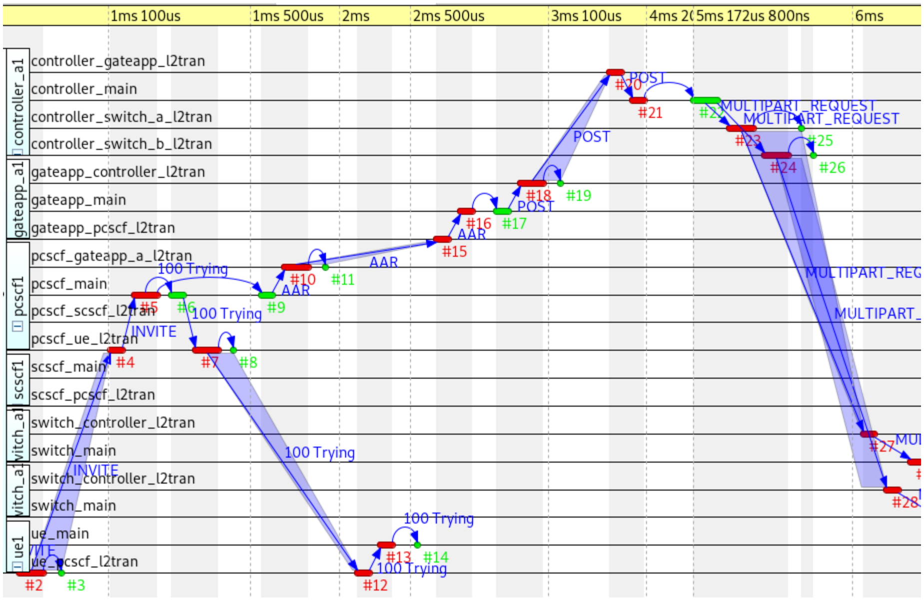


Fig. 5. Log from the performed simulation presented in graphical form using the Sequence Chart Tool (a fragment of window).

Graphical simulation mode (Fig. 4) contains, among others, a window visualizing the structure of the simulated network and communication between the modules, as well as a console displaying information about events occurring during the simulation. In addition to the information provided in the console by default, displaying contents of each sent and received message was implemented. Using these functionalities and running simulations "step-by-step", correctness of message passing through network elements was checked for all service scenarios along with message handling procedures (processing times, changes in the content of message fields). Particular attention was paid to completion of t_1-t_{10} times, which are used to calculate the *CSD* (1) and *CDD* (2) parameters and, consequently, the final simulation results.

The above-mentioned aspects were additionally verified using simulation logs and the Sequence Chart Tool [21]. The optionally recorded simulation log allows generating a diagram of message transitions between the simulator modules (Fig. 5), as well as a detailed analysis of events in text form. This was very useful for final confirmation of correct operation of the simulation model. As a result, it was demonstrated that the functionality of all network elements and all assumed service scenarios were correctly implemented.

In addition to the described functional tests, the paper includes selected results obtained using the developed simulator (Fig. 6) running in dedicated text mode (without graphical interface). The presented results concern the values of CPP parameters for the b1 and f2 scenarios ($E(CSD)$ and $E(CDD)$) versus intensity of call set-up requests generated in one domain λ_{sum} = intrad_call_intensity + interd_call_intensity. For each measuring point, the component intensities (intrad_call_intensity and interd_call_intensity) were equal to each other. The following values of the base delay (the "delay" input variable from Tab. 1) were used in the research: 0.5 ms for all CSCF servers and GATEAPP elements, 10 ms for all SUP-FE/SAA-FE units, 1ms for other network elements. The remaining simulation input parameters were set to their default values (Tab. 1).

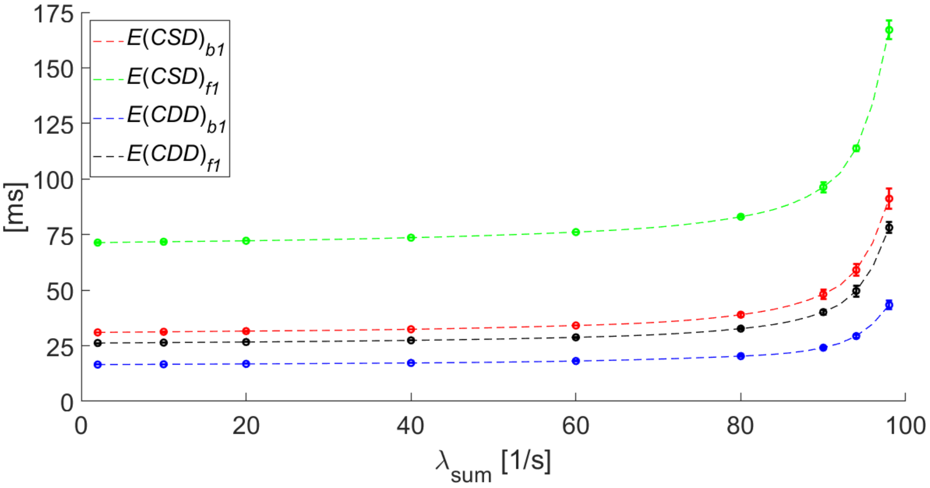


Fig. 6. $E(CSD)$ and $E(CDD)$ for the b1 and f1 scenarios versus intensity of call set-up requests generated in one domain (in the entire network, total intensity is twice larger). Dashed lines represent interpolation between the simulation results, which are points with confidence intervals.

It can be noticed that $E(CSD)$ and $E(CDD)$ times raise with increasing λ_{sum} values and the modeled network can handle up to 90 call set-up requests per second generated in each domain (above this value the network begins to be overloaded and may be unstable). Additionally, the values of CPP parameters for more complicated multidomain call scenarios (f1) are higher than for scenarios within one domain (b1). Moreover, due to the more complex call set-up process, for particular call scenarios presented in Fig. 6 (b1 and f1) mean *CSD* times are greater than mean *CDD* times. The obtained results are as expected and confirm proper operation of the implemented simulator.

5 Conclusions

The paper presents a simulation model for a multidomain IMS/NGN telecommunications network based on the SDN concept in the transport stratum. No other simulation models for such a telecommunication network solution are provided by the scientific community. The proposed model enables a comprehensive analysis of the IMS/NGN/SDN architecture. It takes into account a wide set of service scenarios generated in both domains (registration, intra- and inter-operator calls), as well as parame-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

ters of network elements and traffic sources (including the probability of transport resource unavailability resulting in call set-up failure). The above mentioned scenarios and parameters were analyzed and verified in a laboratory testbed [13]. The output variables of the simulator are mean Call Set-up Delay ($E(CSD)$) and mean Call Disengagement Delay ($E(CDD)$) provided separately for all types of successful call scenarios. They are a subset of standardized Call Processing Performance parameters important for network users and operators.

The simulator was developed in the OMNeT++ environment and has a modular structure enabling its easy modification and extension. It was subjected to detailed functional tests at the stage of source code developing (partial tests) and after this process was completed (final tests). Using various tools, the correctness of message passing through network elements and message handling procedures in these elements (e.g. delays and changes in message fields) were checked for all service scenarios. The performed tests demonstrated that the functionality of all network elements and all assumed service scenarios are correctly implemented. Consequently, it can be stated that the described simulator reflects the phenomena taking place in real network.

Additionally, selected research results obtained using the simulator were presented. They confirmed the expected relationships between $E(CSD)$ and $E(CDD)$ values for intra- and inter-operator calls. They demonstrated that for the assumed parameter values, the modeled network is able to handle even 90 call set-up requests per second generated in each domain, without overload.

The presented research and test results allowed achieving the aim of this paper. It was demonstrated that the concept of integrating IMS/NGN and SDN works correctly and can be used in practice. However, it is necessary to comprehensively examine the properties of this concept, including the influence of SDN on quality in both IMS/NGN service and transport stratum [23], which will be carried out using the developed simulation model.

Additional research using the presented simulator is planned in order to collect experience and develop an analytical model for a multidomain IMS/NGN network based on the SDN concept in the transport stratum. This will make a step to move from analyzing the operation of this network to designing its resources in order to ensure appropriate quality parameters.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. General overview of NGN, ITU-T Recommendation Y.2001, December 2004
2. IP Multimedia Subsystem (IMS); Stage 2 (Release 18), 3GPP TS 23.228 v18.4.0, December 2023
3. Framework of Software-Defined Networking, ITU-T Recommendation Y.3300, June 2014.
4. Bolanowski, M., Gerka, A., Paszkiewicz, A., Ganzha, M., Paprzycki, M.: Application of Genetic Algorithm to Load Balancing in Networks with a Homogeneous Traffic Flow. In: Mikyška, J., de Mulatier, C., Paszynski, M., Krzhizhanovskaya, V.V., Dongarra, J.J., Sloot,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16

P.M. (eds.) Computational Science – ICCS 2023, Lecture Notes in Computer Science, vol 14074, pp. 314–321. Springer, Cham (2023)
5. Call processing performance for voice service in hybrid IP networks, ITU-T Recommendation Y.1530, November 2007
6. SIP-based call processing performance, ITU-T Recommendation Y.1531, November 2007
7. Katov, A. N., Anggorojati, B., Kyriazakos, S., Mihovska, A. D., Prasad, N. R.: Towards Internet of Services - SDN-enabled IMS Architecture for IoT Integration. In: Proceedings of 18th International Symposium on Wireless Personal Multimedia Communications, WPMC 2015, Hyderabad, India (2015)
8. Kang, S., Yoon, W.: SDN-based resource allocation for heterogeneous LTE and WLAN multi-radio networks. The Journal of Supercomputing **72**(4), 1342–1362 (2016)
9. Tranoris, C., Denazis, S., Mouratidis, N., Dowling, P., Tynan, J.: Integrating OpenFlow in IMS Networks and Enabling for Future Internet Research and Experimentation. In: Galis, A., Gavras, A. (eds.) The Future Internet. FIA 2013, Lecture Notes in Computer Science, vol. 7858, pp. 77–88. Springer, Heidelberg (2013)
10. Khairi, S., Raouyane, B., Bellafkih, M.: Towards Enhanced QoS Management SDN-based for Next Generation Networks with QoE Evaluation: IMS Use Case. Journal of Mobile Multimedia **13**(3-4), 183-196 (2017)
11. Liu, Z., Wang, Q., Lee, J.-O.: A SDN Controller Enabled Architecture for the IMS. In: Proceedings of 20th Asia-Pacific Network Operations and Management Symposium, APNOMS 2019, pp. 1-4, Matsue, Japan (2019)
12. Tang, C.-S., Twu, C.-Y., Ju, J.-H., Tsou, Y.-D.: Collaboration of IMS and SDN to enable new ICT service creation. In: Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium, pp. 1-4, Hsinchu, Taiwan (2014)
13. Kaczmarek, S., Sac, M., Bachorski, K.: Implementation of IMS/NGN Transport Stratum Based on the SDN Concept. Sensors **23**(12), 5481 (2023)
14. Rosenberg, J., et al.: SIP: Session Initiation Protocol, IETF RFC 3261, June 2002
15. Fajardo, V., et al.: Diameter Base Protocol, IETF RFC 6733, October 2012
16. OpenFlow Switch Specification Version 1.5.1, ONF TS-025, April 2015
17. Resource Control Protocol No. 1, Version 3 — Protocol at the Rs Interface between Service Control Entities and the Policy Decision Physical Entity, ITU-T Recommendation Q.3301, August 2013
18. Fielding, R., Nottingham, M., Reschke, J.: HTTP/1.1, IETF RFC 9112, June 2022
19. Open Network Operating System (ONOS) SDN Controller, https://opennetworking.org/onos/, last accessed 2024/03/01
20. Kaczmarek, S., Sac, M.: Traffic model of a multidomain IMS/NGN. Telecommunication Review and Telecommunication News (8-9), 1030-1038 (2014)
21. OMNeT++ Documentation, https://omnetpp.org/documentation, last accessed 2024/03/01
22. Kaczmarek, S., Sac, M.: Verification of the Analytical Traffic Model of a Multidomain IMS/NGN Using the Simulation Model. In: Grzech, A., et al. (eds.) Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part II, Advances in Intelligent Systems and Computing, vol. 430, pp. 109-130. Springer, Cham (2016)
23. Kaczmarek, S., Litka, J.A.: Impact of SDN Controller's Performance on Quality of Service. IEEE Access **12**, 8262-8282 (2024)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_16