IP Multimedia Subsystem - Presence Service

LUCIA DEMETEROVA, IVAN BARONAK, ERIK CHROMY
Slovak University of Technology in Bratislava
Faculty of Electrical Engineering and Information Technology
Ilkovicova 3, 812 19 Bratislava
SLOVAKIA
luc.demeterova@gmail.com, baronak@ut.fei.stuba.sk, erik.chromy@ut.fei.stuba.sk

Abstract: The aim of the article is oriented at one of the services of IMS presence service. The Presence Service faces a great usage nowadays, which can lead to problems following the increasing number of users. Our motivation is to illustrate the possible complications which can occur and to show possible solutions. We visualized the number of messages which are created during the activity of the presence service and illustrated them into the form of graphs for better representation.

Key–Words: IP Multimedia Subsystem, NOTIFY messages, Presence Service

1 Introduction

Nowadays, IP Multimedia subsystem introduces a wide range of benefits. Besides the interworking between circuit switching and packet switching networks it also allows to implement new services without larger intervention into the network structure. The realization can be as a single server or through the inner structure.

The presence service faces up a wide range of popularity nowadays. It informs the users about the presence of other users in the network. It also creates a dynamic user profile which is visible to others and it serves for representation, sharing information and service command. The messages (which are SIP requests) have an important role during the communication. There are 3 types of messages - PUBLISH, NOTIFY, SUBSCRIBE, and each of them plays a different role during communication [1], [2].

The rest of this paper is organized as follows. Section 2 describes the architecture of IP Multimedia Subsystem. Section 3 describes the presence service. In section 4 is the analyze of messages and communication of the presence service, service optimalization, traffic analysis and traffic modelling. The results are in section 5.

2 IP Multimedia Subsystem

IP Multimedia Subsystem (IMS) is a set of requirements and specifications which are defined in 3GPP and 3GPP2. It allows the convergention of wireless and stabile access networks for creation, providing and using the multimedia services. The architecture of IMS consists of 3 layers: transport, control and application layer. It is shown in the Fig. 1.

The first layer is a transport layer. It provides the access from different types of networks (such as GPRS, UMTS, IP, PSTN) through the network transitions (such as MGW, SGW). It also transports messages from the user to the control layer through data IP network.

The control layer is the network core. It controls the communication and creates connections between users. Key components in the control layer are SIP servers and proxy servers. They have a common name Call Session Control Function (CSCF). Their function is message routing and utilization. S-CSCF (Serving CSCF) is the centre of all signalling functions in the IMS network. Due to its managing function it also governs the function of routing. P-CSCF (Proxy CSCF) has the role of the SIP proxy server for IMS UE terminal incoming and outgoing messages. It also encapsulates and verifies the SIP messages and creates data for observing other data for UE. I-CSCF (Interrogating CSCF) is situated at the edge of the network organisation domain. CSCF allows the control of QoS policy. It contains parameters obtaining the type of media, codecs and coding format, also the bandwidth, delay and packet loss. The control layer contains HSS user database, which contains user profile, information about the location and information policy. If there are two or more HSS databases in the network, the information is obtained via the SLF server.

The application layer provides services [3]–[9]. There are 3 types of servers: SIP application server, OSA application server and CAMEL application server. Application servers host and invoke the services for users and they observe the functions of SIP application servers. They are connected with HSS to upload and download user data [10]–[13]. IMS works with several protocols. The key protocols are:

- SIP (Session Initiation Protocol), which is a control protocol for IMS session. It uses UDP (User Datagram Protocol) but if necessary, it may also use the TCP (Transmission Control Protocol). It provides the VoIP connections by cooperating with other protocols. Bodies of SIP packets carry SDP (Session Description Protocol) with information about the transfer. SIP incorporates several elements of HTTP (Hypertext Transfer Protocol) and SMTP (Simple Mail Transfer Protocol).
- DIAMETER provides the AAA frame (Authentication, Authorization, Accounting).
- COPS is responsible for exchanging policy information.
- H.248/MEGACO (Media Gateway Control) is used by IMS Media Gateway for media conversion. It provides the end-to-end communication.
- RTP/RTCP, where RTP (Real-time transport protocol) is used for media and end-to-end service delivery. The RTCP (RTP Control Protocol) assignment is the quality feedback of transmission and receiving of data transferred by RTP.
- SCTP (Stream Control Transmission Protocol) is used to transfer the PSTN (Public Switched Telephone Network) signalling messages through IP networks.

3 Presence Service

Presence service receives, saves and categorizes the actual information. Its realization can be as a single server or through the inner structure which contains several servers and entities. It represents a dynamic profile of the user. The profile is visible for other users and serves for representation, sharing of information and control of services. Presence service displays the current status of the user and it informs other users about his accessibility. Nowadays, the popular way of representation of presence is through icons. They are showed to other users in the same chat. The statuses are applied through various forms, such as graphic symbols or text description. It depends on the type of communication. Presence service is a part of 3GPP Release 6. Architecture is shown in the Fig. 2. It has 3 layers: agents, Entities and servers.

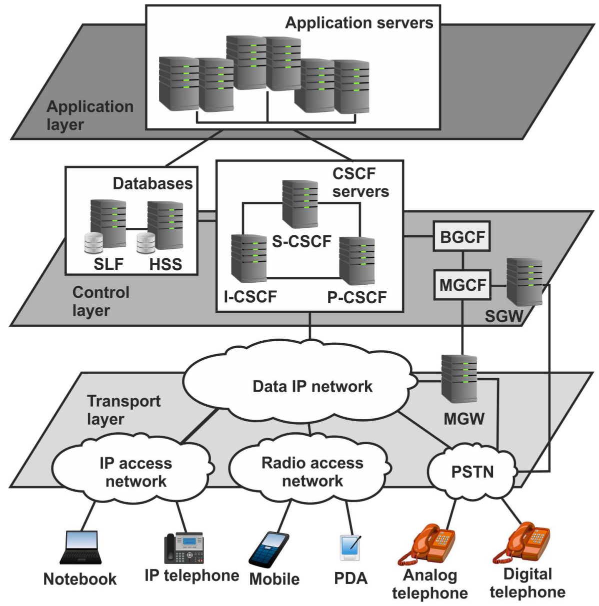

Figure 1: Architecture of IMS.

- Presence service agents collect information from different sources.
- Presence user agent collects information about the presence and supplies them to Presence server (which is the main source).
- Presence network agent obtains the information about presence from network elements.
- Presence external agent supplies the information about presence from external network and sends it in a standardized format via the reference point Pex.

Entities utilize SIP requests (UE, S-CSCF a AS) and divide them into 2 types.

- Presence service agents collect information from different sources.
- Presence user agent collects information about the presence and supplies them to Presence server (which is the main source).
- Presence network agent obtains the information about presence from network elements.
- Presence external agent supplies the information about presence from external network and sends it in a standardized format via the reference point Pex.

Entities utilize SIP requests (UE, S-CSCF a AS) and divide them into 2 types.

- Presentities inform the presence service about status changes of users.
- Watcher receives the actualizations about the user friends.

Servers collect and send the user information saved in XML documents.

- The presence server receives the messages and allocates them to the correct user.
- The resource list server creates user lists and sends their statuses.
- Application server is created to control the number of messages [10], [11], [12].

In case number of incoming messages is larger than the ability of the server to send them away, they are moved to the waiting queue. If the waiting queue is full, other incoming messages are deleted. The role of the interfaces in Presence service is to define rules between parts of Presence service. They are shown in the Fig. 2. Their functions are:

- Peu - authorisation and transfer of static user information.
- Pep - actualisation of information about the User agent, it allows to verify the User agent watchers.
- Pen - the definition of communication rules between the server of Presence network agent and the Presence server.
- Pex - functions for Presence External Agent (same as Pen for Presence network agent).
- Ph - verifies the information about the user statuses located in HSS/HLR (for Presence network agent).
- Pi - verifies the information from IMS network (for Presence network agent).
- Pc - verifies the information about mobility control from MSC (Mobile Switching Centre), (for Presence network agent).
- Pk - verifies the information about AAA events (for Presence network agent).
- Pl - verifies the information about location of the user (for Presence network agent).
- Pw - watchers obtain the information aboutstatuses of presentities due to Pw.
- Pwp - provides functions of Pw and Pep.
- Px - allows allocation of Presence server topresentity.
- Pet - allows watchers to edit the list of resources for servers.

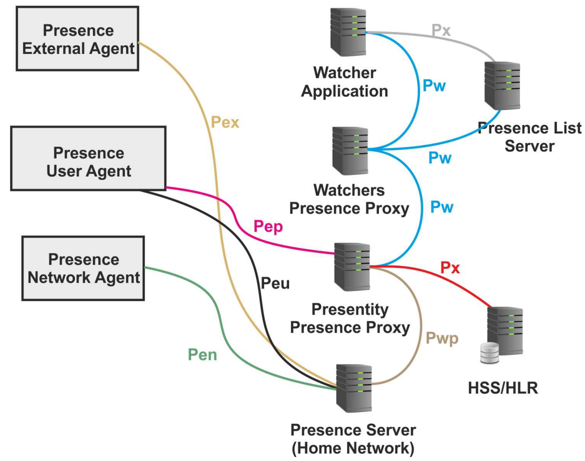

Figure 2: Architecture of presence service.

4 Messages and Communication

The presence service uses 3 types of SIP requests: PUBLISH sent when user goes online or offline (pub_login, pub_logout), modifies its status (pub_modify) or refreshes the status (pub_refresh). SUBSCRIBE sent when user initializes (sub_initial), terminates (sub_terminal) or refreshes (sub_refresh) subscribing of information from presence server. NOTIFY is sent by server, which informs the users about the status of presentity. NOTIFY has the largest representation among the SIP requests. Message body contains XML document configured by XCAP protocol. XCAP also manages the presence list of users friends. Message exchange is realized by 2 processes.

The process of publishing is shown in the Fig. 3 which has two parts: registration (messages 1-20) and status publishing (messages 21-32). During the registration, the S-CSCF is allocated to the user. User equipment (UE) represents a mobile device entering the IMS through the P-CSCF. The P-CSCF defines the destination of registration request through the S-CSCF. The information about the user profile is stored in HSS. S-CSCF sends the 401 (unauthorised) answer. After the UE receives the answer, it creates another Register request. Then the user is successfully registered. In messages 21-23, the UE sends the whole status to the application server in Public request. When the registration is finished, messages pass only through the P-CSCF and S-CSCF. The S-CSCF can locate the server because of iFC (Initial filter criteria) obtained by HSS during the registration. As soon as possible, the confirmation message 200 (OK) is sent by the server to predict re-sending the messages. In the case the status changes, UE sends another Publish request the same way as the first time. The message contains only the information about changes.

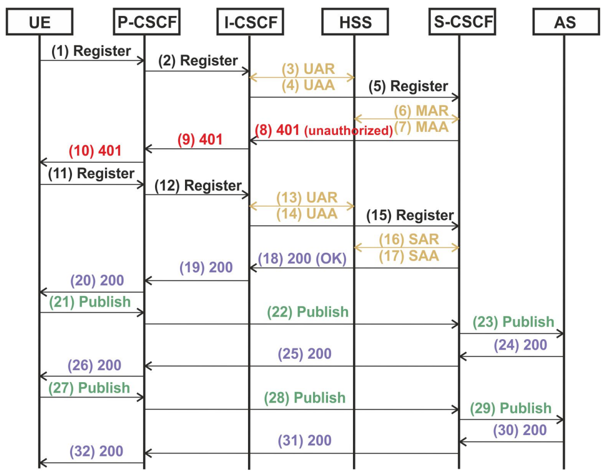

Figure 3: Process of publishing.

The process of subscribing is shown in the Fig. 4. Picture describes the situation when the watcher is in another IMS network like the presence server. The process of registration is the same as process of publishing. I-CSCF is the initial point to another IMS network. UE (watcher) creates the Subscribe request. The subscribe request contains the filter obtaining the information about what the watcher wants to see. UE enters the network through the P-CSCF and continues to S-CSCF. Then the I-CSCF finds the S-CSCF and sends the request to the AS, where the list of contacts and statuses is stored. After receiving the request, the application server confirms the user authorisation. If the authorisation is correct, the application server answers with 200 (OK). AS then sends a Notify request with information about the presence. In case some of the user statuses will change, another Notify message is sent by the server without any former request. However, the user may use several devices for subscribing its status, using its account as the only identifier is insufficient. As a solution, two overheads were created - SIP-Etag and SIP-If-Match. They are designed for identification of status. Except the first PUBLISH message, each message should contain SIP-If-Match Overhead carrying the information for refreshing, changing or deleting the status. After SIP server receives this message, it changes the user status. In the next step SIP-Etag overhead is filled with another generated identifier and answers the user with message 200 OK.

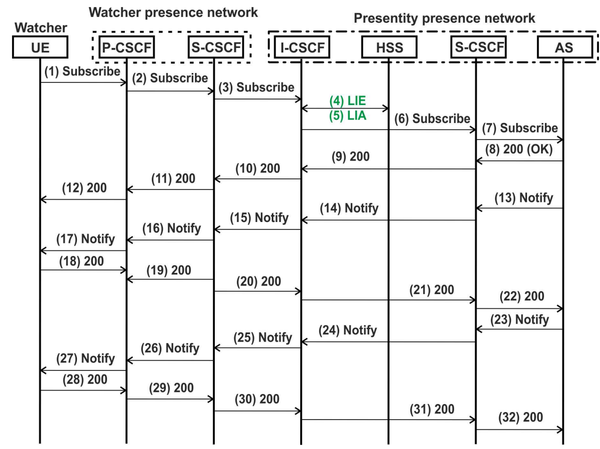

Figure 4: Process of subscribing.

4.1 Service Optimalisation

Necessary operations during status publishing and sending notifications contain PIDF/RPID XML document. These documents have a large capability due to large amount of information they contain. The watcher, subscribing the information from presentities, usually obtains one of these XML documents with every change of any status. In case a smaller device with low performance and restricted memory is used, the device may be overloaded by the large amount of information and wont be able to obtain the information or utilize it in real time. To prevent the device overload, we need to find the compromise between the amount of the sent information, frequency of notify requests and information sending bandwidth [14].

4.2 Traffic Analysis

Studies show presence service covers around 50% (and more) of signalling traffic used by IMS core. It represents heavy load and it is necessary to solve this problem. NOTIFY messages occupy the biggest traffic load storage. Because of presence service shows the actualization of user states, the traffic load depends on user behaviour characteristics in the service. Traffic load is affected by the interval of users logging in/logging out, behaviour of users in online state and also their number of contacts. Each SIP request directed to server may initiate transmission creation. The number of transmissions processed by SIP server is an important parameter which describes the service capacity. There are several types of SIP processes with different data processing. First log in generates initial PUBLISH message r_initial_pub and the refreshing PUBLISH messages r_refresh_pub, which are generated regularly until the user log off. Then the terminal PUBLISH message r_terminal_pub is sent. Result of every initial and terminal PUBLISH message is NOTIFY message. For subscribing of presence state of user, initial SUBSCRIBE message r_initial_sub is sent to the presence server. Then refreshing SUBSCRIBE message r_refresh_sub is sent. In the end of communication, the terminal SUBSCRIBE message r_terminal_sub is sent and subscribing of presence of other users is cancelled. When the user status changes, modifying PUBLISH message r_modify_pub is sent. Each of these 3 types of messages are followed by n_online_watcher NOTIFY messages. NOTIFY messages r_notify have the biggest representation in the system. The number of users logging in/logging out is given by the following equation (1):

$$r\_notify = n\_online\_watcher.(r\_pub\_login + r\_pub\_logout + r\_pub\_modify + r\_pub\_refresh) \quad (1)$$

The number of messages depends on incoming requests and on authorized watchers. NOTIFY requests are directed to waiting queue. The server sends messages in periodical intervals to prevent the network overload. If the waiting queue is filled, next incoming messages are deleted [2], [15].

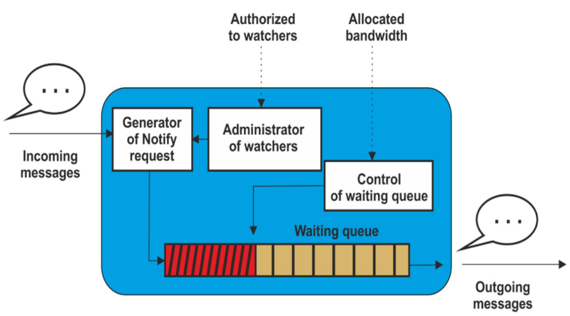

Figure 5: Creating of NOTIFY requests.

4.3 Traffic Modelling

Creation of messages at queuing systems could be shown as Markov chains and is shown in matrix, where users can be in 3 states represented by *s0*, *s1*, *s2*. *s0* represents the user who has not changed his or her status since coming online. *s1* represents the online user, who changed his status since he came online. *s2* represents the offline user. The probability of the status change by the matrix is shown:

$$\begin{pmatrix} 1-p_{01}-p_{02} & p_{01} & p_{02} \\ 1-p_{11}-p_{22} & p_{11} & p_{12} \\ p_{20} & p_{21} & 1-p_{20}-p_{21} \end{pmatrix} \quad (2)$$

Various states in matrix mean the following probabilities:

- Probability in which online presentity does not change your presence status over time Δt:

$$P(s_0, t \mid s_0, \Delta t) = 1 - p_{01} - p_{02} \quad (3)$$

- Probability in which online presentity changes your presence status over time Δt:

$$P(s_0, t \mid s_1, \Delta t) = p_{01} \quad (4)$$

- Probability in which online presentity goes offline over time Δt:

$$P(s_0, t \mid s_2, \Delta t) = p_{02} \quad (5)$$

- Probability in which online presentity does not change your presence status over time Δt:

$$P(s_1, t \mid s_0, \Delta t) = 1 - p_{11} - p_{12} \quad (6)$$

- Probability in which online presentity changes your presence status over time Δt:

$$P(s_1, t \mid s_1, \Delta t) = p_{11} \quad (7)$$

- Probability in which online presentity goes offline over time Δt:

$$P(s_1, t \mid s_2, \Delta t) = p_{12} \quad (8)$$

- Probability in which offline presentity goes online over time Δt:

$$P(s_2, t \mid s_0, \Delta t) = p_{20} \quad (9)$$

- Probability in which offline presentity changes your presence status over time Δt:

$$P(s_2, t \mid s_1, \Delta t) = p_{21} \quad (10)$$

- Probability in which offline presentity stays offline over time Δt:

$$P(s_2, t \mid s_2, \Delta t) = 1 - p_{20} - p_{21} \quad (11)$$

- Probability of transition from one state to another p_{ij} is given by exponential distribution (where *i = 0, 1, 2* and *j = 0, 1, 2*):

$$p_{ij} = \int_0^{\Delta t} \lambda_{ij}.e^{-\lambda_{ij}.x}\, dx \quad (12)$$

$$\lambda_{ij} = \frac{1}{t_{ij}} \quad (13)$$

where t_{ij} represents the average time of creation message.

5 Graphic Visualisation of Message Amount

The Fig. 6 shows creation and behaviour of PUBLISH messages while the number of online users in network increases. For this example, we used a network with 825 000 users.

The number of messages is given by the following equations:

$$pub\_modify(t) = s_0(t).P(s_0, t \mid s_1, t - \Delta t) + s_1(t).P(s_1, t - \Delta t) \quad (14)$$

$$pub\_login(t) = s_2(t).P(s_2, t \mid s_0, t - \Delta t) \quad (15)$$

$$pub\_logout(t) = s_0(t).P(s_0, t \mid s_2, t - \Delta t) + s_1(t).P(s_1, t \mid s_2, t - \Delta t) \quad (16)$$

$$pub\_refresh(t) = s_0(t).P_{Ref} + s_1(t).P_{Ref} \quad (17)$$

$$P_{Ref} = (1 - \int_0^R \frac{1}{t_m}.e^{(-\frac{1}{t_m})x}\, dx) + (1 - \int_0^R \frac{1}{t_{off}}.e^{(-\frac{1}{t_{off}})x}\, dx) \quad (18)$$

where t_m represents the average time of status changes of users and t_{off} represents the average time of offline users.

The Fig. 6 shows the number of pub_modify messages exponentially increases and the number of created messages is much higher than the number of other created requests. Pub_refresh curve has a similar behaviour, but the number of messages is lower and their creation begins later. Pub_logout messages decrease because of more online users than offline. The last curve shows the number of pub_login messages. The curve increases and fluently determines with the determination of online users.

The Fig. 7 shows the amount of NOTIFY messages created in the presence service. Because of their number which is much higher than in the Fig. 6, we used the time axis with lower units.

The number of NOTIFY messages is given by the following equation:

$$notify(t) = watchers(t).(pub\_refresh(t) + pub\_modify(t) + pub\_login(t) + pub\_logout(t)) \quad (19)$$

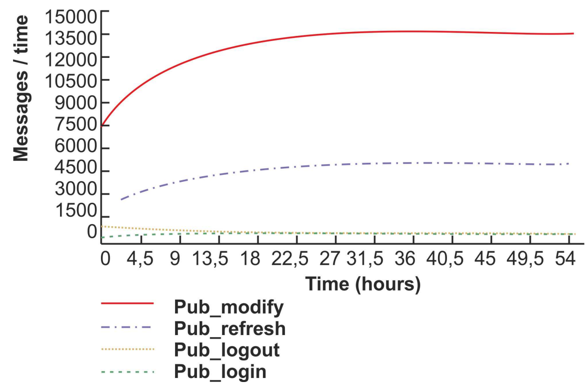

Figure 6: Creation of PUBLISH messages.

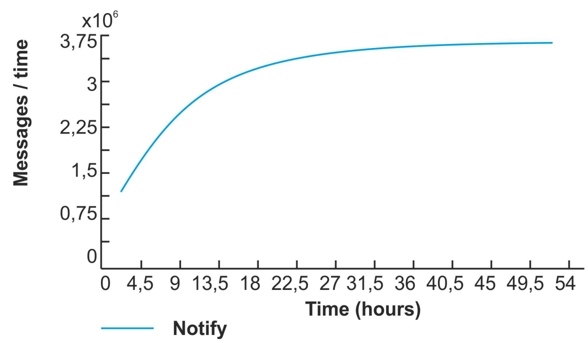

Figure 7: Creation of NOTIFY messages.

Curve of NOTIFY messages shows the exponential growth of messages with increasing time. Creation of NOTIFY messages follows every creation of PUBLISH messages, that is why their number is significantly higher.

The Fig. 8 shows the approximate comparison of creation of PUBLISH and NOTIFY messages. Because the number of NOTIFY requests is much higher than the number of PUBLISH messages, the curve rises more rapidly.

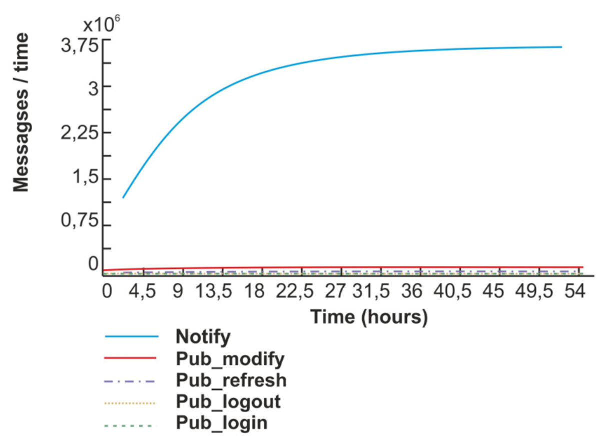

Figure 8: The comparison of created messages.

Although the comparison is only approximate, it shows the potential threat of system failure. Due to the common usage of presence service nowadays and the increasing number of users, this may cause service overload.

6 Conclusion

The presence service is being highly used nowadays. Most of the users use the applications based on presence service and the number of users is increasing. It is the reason, why servers usually confront the overload.

Due to the big number of requests waiting for processing and their deleting when the waiting queue is full, this may cause data loss and degradation of QoS in queuing systems and the presence service.

The possible solution could be the design of traffic directed to several servers. The role of the primary server is to process the incoming requests. The agent obtained in this server shows the number of incoming messages and sums up if the state of the waiting queue is fulfilled with requests. After this state, the following messages could be directed to another server located in the same cloud.

Another solution could be the system of message division due to their priority. Messages with high priority could be processed with primary server with highest performance. Messages with lower priority could be directed to server with lower performance. This solution opens the possibility of waiting queue usage much more effectively.

These solutions present wide mapping of traffic. While the applications are used by different users, these suggestions could be realized globally.

Acknowledgements: This article was created with the support of the Ministry of Education, Science, Research and Sport of the Slovak Republic within the KEGA agency project - 007STU-4/2016 Progressive educational methods in the field of telecommunications multiservice networks and VEGA agency project - 1/0462/17 Modelling of qualitative parameters in IMS networks.

References:

[1] P. Bazot, R. Huber, J. Kappel, B.S. Subramanian, E. Oguejiofor, B. Georges, C. Jackson, C. Martin and A.Sur, Developing SIP and IP Multimedia Subsystem (IMS) Applications. IBM Corp. 2007, First edition, ISBN 0738489573.

[2] D. Petras, I. Baronak and E. Chromy, Presence service in IMS. The Scientific World Journal, Volume 2013 (2013), Article ID 606790, ISSN: 1537-744X (Online), http://dx.doi.org/10.1155/2013/606790.

[3] F. De Rango, M. Tropea, P. Fazio and S. Marano, Call admission control with statistical multiplexing for aggregate MPEG traffic in a DVB-RCS satellite network. GLOBECOM '05, IEEE Global Telecommunications Conference, 2005, St. Louis, MO, USA, pp. 3231-3236, ISBN: 0-7803-9414-3.

[4] R. Roka and F. Certik, Simulation and analysis of the signal transmission in the optical transmission medium. SIMULTECH 2015 - 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Colmar, France, 2015, pp. 219-226.

[5] F. Certik and R. Roka, Possibilities for Advanced Encoding Techniques at Signal Transmission in the Optical Transmission Medium. Journal of Engineering (United States), Vol. 2016, Article number 2385372, Hindawi Publishing Corporation, ISSN 2314-4904.

[6] L. Orcik, M. Voznak, J. Rozhon, F. Rezac, J. Slachta, H. Toral-Cruz and J.C.-W. Lin, Prediction of Speech Quality Based on Resilient Backpropagation Artificial Neural Network. Wireless Personal Communications, Volume 96, Issue 4, October 2017, pp. 5375-5389, ISSN: 09296212.

[7] A. Kovac, M. Halas and M. Orgon, E-model MOS estimate improvement through jitter buffer packet loss modelling. Advances in Electrical and Electronic Engineering, Vol. 9, No. 5, 2011, VSB-Technical University of Ostrava, pp. 233-242, ISSN 1804-3119.

[8] J. Frnda, M. Voznak and L. Sevcik, Impact of packet loss and delay variation on the quality of real-time video streaming. Telecommunication Systems, Vol. 62, No. 2, 2016, Springer New York LLC, pp. 265-275, ISSN 1018-4864.

[9] M. Halas and S. Klucik, Modelling the probability density function of IPTV traffic packet delay variation. Advances in Electrical and Electronic Engineering, Vol. 10, No. 4, 2012, VSB-Technical University of Ostrava, pp. 259-263, ISSN 1804-3119.

[10] K. Al-Begain, Ch. Balakrishna, L. Angel Galindo and D.M. Fernandez, A Development

and Deployment Perspective. John Wiley & Sons, 2009, ISBN: 978-0-470-74034-7.

[11] M. Poikselka, A. Niemi, H. Khartabil and G. Mayer, The IMS: IP Multimedia Concepts and Services. 2nd Edition, John Wiley & Sons, 2007, ISBN: 978-0-470-03183-4.

[12] N. Wilkinson, Next Generation Network Services: Technologies and Strategies. John Wiley & Sons, 2002, ISBN: 978-0-471-48667-1.

[13] M. Ilyas and S.A. Ahson, IP Multimedia Subsystem (IMS) Handbook. Taylor & Francis Group, 2009. ISBN 9781420064599.

[14] G. Camarillo and M.-A. Garcia-Martin, The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds. 3rd Edition, John Wiley & Sons, 2008, ISBN: 978-0-470-51662-1.

[15] C. Chi, R. Hao, D. Wang and Z. Cao, IMS presence server: Traffic analysis & performance modelling. IEEE International Conference on Network Protocols, 2008, Orlando, FL, USA, October, 2008, pp. 63-72.