AN APPROACH TO PROVIDE OPEN ACCESS TO PROXIMITY-BASED SERVICE[1];

Evelina Pencheva, Denitsa Kireva, Ivaylo Atanasov, Ventsislav Trifonov

Technichal University of Sofia
e-mails: enp@tu-sofia.bg; kireva@tu-sofia.bg; iia@tu-sofia.bg;vgt@tu-sofia.bg
Bulgaria

Abstract: Device-to-Device (D2D) technology allows devices to detect each other within proximity and communicate directly. Current standards define Proximity-based Service (ProSe) which supports both direct and network supported device discovery. The paper proposes an approach to open access to ProSe functionality based on network supported discovery. The open access to ProSe functionality enables creation and deployment of 3rd party applications that initiate D2D communications in case of cellular network congestion. The 3rd party applications may access ProSe functions through Web Service interfaces. The proposed interfaces are described by information flows and actual interface definition. Models of ProSe state as seen by 3rd party application and by the network are proposed, formally described and verified.

Key words: Device-to-Device Communications, Cellular Network Offloading; Web Services; Application Programming Interfaces; Finite State Machines.

1. INTRODUCTION

Device-to-Device (D2D) communications are aimed to relieve congestion in the network in environment of rapidly growing mobile data traffic. D2D communications enable cellular traffic offloading for resource intensive and interactive applications on mobile devices.

This type of communications may play a key role in multi-access edge computing, facilitating traffic offloading and effective resource sharing for UEs (User Equipments) which are close to each other. The proximity of UEs provides high bit rates, low latency, and low energy consumption.

Typical use cases for direct communications between UEs include vehicle to vehicle communications (collision avoidance and sharing of traffic information), local data services (information sharing, mobile advertising, community services, streaming

[1] The research is conducted under the grant of project DH07/10-2016, funded by National Science Fund, Ministry of Education and Science, Bulgaria.

services, mobile multiplayer gaming), optimization of network signalling (group handover of multiple users), alleviation of network load in case of congestion, public safety services, etc. [1].

Considerable research is ongoing in the field of offloading techniques. In [2], the authors present a smart base station-assisted partial-flow device-to-device offloading system that provides seamless video streaming services to clients by effectively offloading parts of the video traffic between devices in order to alleviate the cellular network traffic load. The offloading utility of communications between devices may be maximized by proposing an optimal content pushing strategy based on the user interests and sharing willingness [3]. In [4], the authors study mobility issues of data offloading radio in device-to-device communications. An offloading scheme of D2D communications on multi-radio technologies is proposed in [5]. The scheme can achieve high link spectrum efficiency without sacrificing the available terminal density. In [6], the authors formulate an optimization problem in D2D communications to maximize the number of users served and reduce the number of access points deployed while satisfying a set of system constraints. An incentive framework of D2D offloading, where the operator encourages some users acting as D2D transmitters to broadcast their popular contents to nearby region to improve operators' overall economic efficiency is proposed in [7]. The authors of [8] investigate the relationship between offloading gain of the system and energy cost of each device involved in the process.

D2D communications are based on Proximity-based Service which allows direct communication between User Equipments (UEs). As to [9], Proximity-based Service (ProSe) allows identification of ProSe-enabled UEs which are in proximity, using cellular or wireless technology, as well as establishment of communication paths between two or more ProSe-enabled UEs that are in direct communication range. The ProSe Direct Communication path could use Long term Evolution (LTE) access network or Wireless Local Area network (WLAN). In this paper, we propose an approach to open access to ProSe based on Service-oriented Architecture (SOA) using Web service technology [10]. In current standards, it is the UE that initiates proximity discovery and communication offloading. With the proposed approach, it is a 3rd party or network operator application that initiates the procedures and thus controls the ProSe-enabled UEs connectivity. An authorized application may make a decision for offloading based on access network congestion, device location, device provider credit etc. D2D offloading scheme may be useful for users which are at cell edges, inside isolated environment like basements or large buildings and require better quality of service.

The paper is structured as follows. Section 2 presents the architecture for open access to ProSe. Section 3 provides a description of the proposed Web Service including sequence diagrams which illustrate typical use cases of the Web Service functionality. Section 4 describes Web Service interfaces and the supported operations. Section 5 presents the proposed ProSe state models and their formal verification. The conclusion summarizes the author contribution and outlines ideas for future work.

2. ARCHITECTURE FOR OPEN ACCESS TO PROSE FUNCTIONS

In order to communicate with each other without intermediate nodes, UEs need to discover the presence of other UEs in their vicinity. Proximity discovery appears to be a trigger for direct communications.

The 3GPP standards define direct discovery or core network supported discovery. For the purpose of open access to ProSe functionality, we consider core network supported discovery (core network-level ProSe discovery).

The SCEF (Service Capability Exposure Function) is a functional element which provides a means to securely expose the services and capabilities provided by 3GPP network interfaces. Based on the geographical area provided by the 3rd party AS and the configuration data, the SCEF determines from which RCAF(s) to request congestion reporting. The 3rd party AS (Application Server) runs 3rd party applications that may be used for initiation of D2D communications. The RCAF (RAN Congestion Awareness Function) reports network status to the SCEF to enable decisions based on user plane congestion status. The network status includes the following information: congestion level or an indication of the "no congestion" state, and cell identity for which the congestion level is being provided. The ProSe Function is the logical function that is used for network related actions required for ProSe. It is used to provision the UE with necessary parameters in order to use the service. The ProSe Application Server saves information about PreSe users and provides mapping of ProSe related IDs. The HSS (Home Subscriber Server) stores ProSe subscription information.

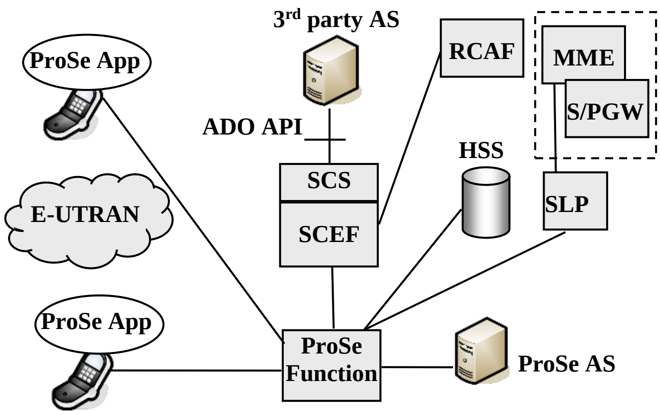

Fig. 1. Architecture for open access to ProSe functionality

The open access to ProSe functionality may be provided through APIs. We propose a Web Service API that may be used for initiation of core network-level ProSe discovery. The API abstracts the detail of ProSe functionality and data and it is defined in implementation independent form for distributed computing environment.

3. DETAILED WEB SERVICE DESCRIPTION

“Application-driven offloading” (ADO) is a service which enables applications to discover that UEs in a specific geographic area are close to each other. Using the Web Service APIs, applications may provide UEs with necessary parameters such as PLMN specific parameters in order to allow the UEs to use ProSe in specific PLNM. The applications may be notified about current cellular network congestion level.

The following scenario provides an example business use-case further to illustrate the application initiated UE discovery. There is a soccer game that is broadcasted under IPTV. The network is overloaded in the area of the stadium. Service providers can benefit of ProSe functionality to take some load off the network in the stadium area by allowing direct transmission among mobile phones and other devices. If the cellular network is overloaded, the application is notified and it may facilitate streaming service (IPTV) by initiating discovery of ProSe-enabled devices in order to form cluster and grouping data within the cluster.

A 3rd party initiated offloading, relying on Proximity based service, might be in help when pre-congestion situation is detected, or otherwise good radio conditions get abruptly worsened for long time, e.g. earthquake caused walls to fall. This flexibility allows some of the 'survived' D2D capable devices to keep being connected and thus improving the chance in localization of injured people or alerting about potential hazards like gas leaks, liquid spills, etc.

Let us imagine that a 3rd party has deployed ProSe Application, which initiates D2D communications in case of network congestion. In order to be able to initiate network offloading, the ProSe application needs to know whether there are ProSe-enabled UE in the hot area.

To obtain ProSe service a ProSe-enabled UE needs to register with the ProSe Function. Then to activate ProSe features such as EPC-level ProSe Discovery for a specific application, the UE registers the application with the ProSe Function. The ProSe Application may subscribe to receive notifications about UE ProSe registration and Application registration for ProSe, as illustrated in Fig.2. For the purpose of 3rd party initiated ProSe discovery, the UE need to provide the identification of the cell on which it is camped.

On receiving notification about network congestion, the ProSe Application determines whether there are UEs registered for ProSe within the context of a specific application in the cell which is overloaded (Fig.3).

If there are such UEs, the ProSe Application initiates proximity request in order to be alerted when the UEs are closed to each other. The ProSe Application provides UEs and Application identities, the window for the period during which the request is valid, and the range class for the application.

The ProSe function processes and validates the proximity requests as explained in [9], and activates location reporting for the respective UEs. The UEs' locations are reported periodically, based on a trigger, or a combination of both. The ProSe Application registers for proximity alerts. When ProSe Function detects that the UEs are in proximity, the ADO Web Service notifies the ProSe Application and the ProSe function informs the respective UEs that they are in proximity with each other. Location reporting is cancelled in the network.

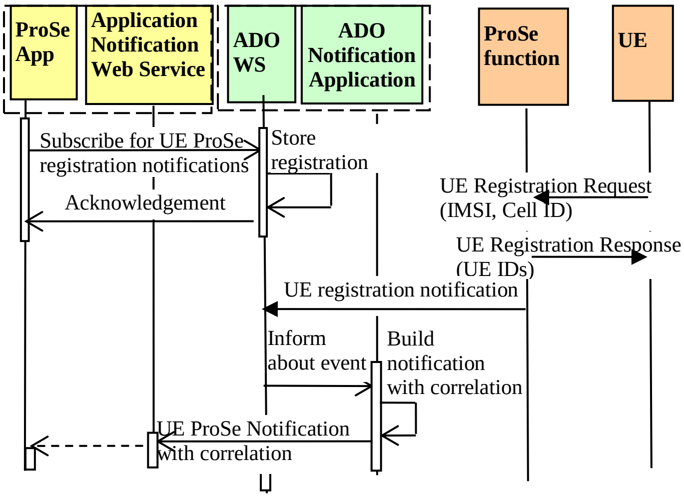

Fig. 2. ProSe Application subscription and notification about UE registration for ProSe

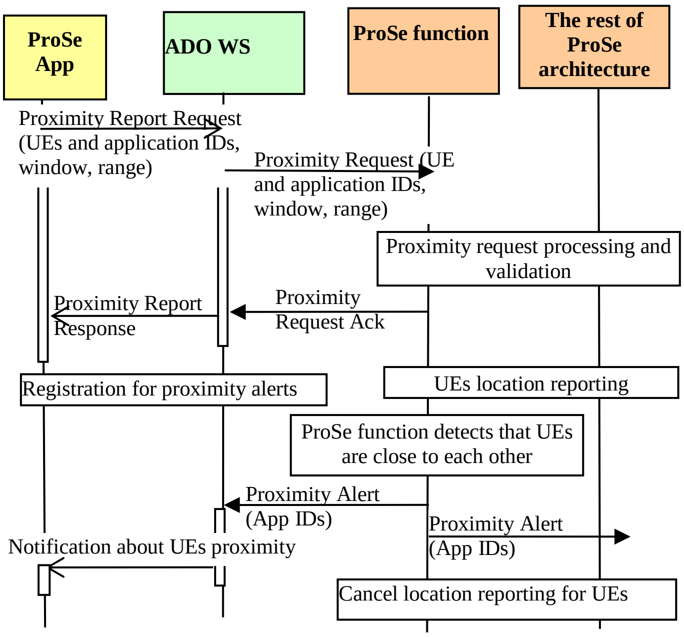

Fig. 3. ProSe Application initiated proximity request for UEs

At any time the UE or the ProSe Function may initiate UE deregistration for ProSe. For example, the UE may decide to deregister for ProSe when there are no ProSe-enabled applications activated on the UE. In case of active subscription for the UE registration status for ProSe, the ADO Web Service notifies the ProSe Application about the UE deregistration about ProSe. The ProSe Application may decide to cancel Proximity Report Request it sent earlier (e.g. when time window is closed).

4. DESCRIPTION OF WEB SERVICE INTERFACES

The ADO Web Service supports the following interfaces.

The RANUserPlaneCongestionNotificationManagement interface is used by the ProSe Applications to manage their registration for notifications about network level congestions. The startRANUserPlaneCongestionNotification operation will be used by the ProSe Application to register its interest in receiving notifications about changes in network congestion levels. The stopRANUserPlaneCongestionNotification operation will be used by the ProSe Applications to stop receiving notifications by cancelling an existing registration.

The RANUserPlaneCongestionNotification interface provides the methods for notifying the Application about the events related to changes in network level congestion. The notifyRANUserPlaneCongLevel operation will report a change in network congestion level. The operation contains cell ID and the network congestion level (e.g. 0: no congestion; 1: congestion; 2: severe congestion).

The UERegistrationNotificationManagement interface is used by ProSe Applications to manage their subscriptions for notifications about registration of Pro-enable UE and Application. It supports operation for initiation and termination of notifications about UE and application registration for ProSe (startUERegist-rationNotification and stopUERegistrationNotification).

The UERegistrationNotification is the interface that provides notifications about UE registration (notifyUERegistration operation) and application registration for ProSe (notifyAppRegistration operation). The operations provide UE and application IDs, and the cell identity. The interface also supports notifyUEDeregistration operation which notifies ProSe applications about UE deregistration including the cause for deregistration.

The ProximityDiscovery interface is used by ProSe Applications to initiate discovery of UEs which are in vicinity to each other. The reportPoximity operation includes UE and application IDs, cell ID. The cancelReportProximity operation will be used by ProSe Application to cancel ongoing discovery.

The ProximityAlertNotificationManagement is used by the ProSe applications to register their interest in receiving alerts for UEs proximity. The ProSe Application will invoke the startProximityAlertNotification operation to start notifications about UEs proximity. The operation distance between devices that shall be monitored, tracking accuracy, indicating the meters of acceptable error in tracking distance, criteria, indicating when notification should occur, frequency, indicating maximum frequency of notifications, and duration the period of time notifications are provided.

The ProximityAlertNotification interface provides operations for alerting the PreSe Application about UEs proximity. When the UEs enter into proximity, the ProSe application is notified by notifyProximityAlert operation. The operation contains application IDs and UE IDs, location information for UEs. The proximityError operation is sent to the ProSe application to indicate that the notifications for UE or the whole notifications are being cancelled by the Web Service. The application is informed by proximityEnd operation that the notifications about UEs proximity discovering have been completed.

5. PROXIMITY-BASED SERVICE STATE MODELS

Models, representing the ProSe service as seen by the application, and by the network, are proposed in the section. These models have to expose equivalent behaviour, i.e. they have to be synchronized. The synchronized behaviour of the models allows to prove in a mathematically formalized manner that the approach is consistently implementable. Mathematical formalism for equivalence of behaviour is used to generate model-based test situations IN ORDER to demonstrate compliance of a system's implementation with its specification.

The SCS/SCEF needs to maintain synchronized views on the state of ProSe as seen by the ProSe application and by the ProSe function. Fig.4 illustrates the simplified ProSe Application view on the ProSe state.

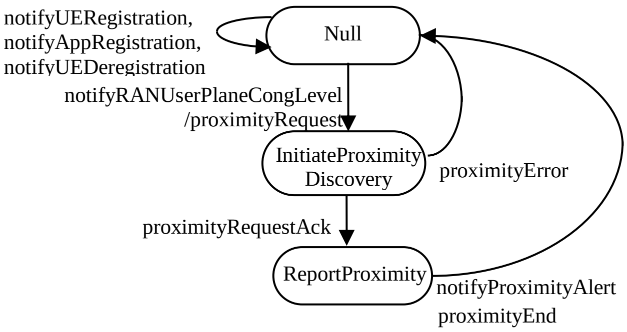

Fig. 4. Application view on the state of Proximity-based Service

The transitions in the state model are driven by methods of Web Service interfaces defined in Section 4.

In Null state, the RAN is not congested and there is no need from ProSe Application point of view to initiate offloading. In Null state, the ProSe Application may be notified about ProSe-enabled UE and/or application registration, as well as for UE deregistration. In Null state, the ProSe Application may be notified about RAN congestion and it sends to the network a proximity request to initiate in the congested cell a discovery of ProSe-enabled UE which are in proximity each other. This results in transition to InitiateProximityDiscovery state. A transition to ReportProximity state occurs when the ProSe Application receives an acknowledgement of the proximity request from the network. In ReportProximity state, the ProSe Application may be notified about UEs proximity and the result is transition to Null state. In ReportProximity state, the ProSe Application may be notified about the end of proximity alerting or an error in proximity alerting.

Fig. 5 illustrates the simplified network view on the state of ProSe. In Registration state, the ProSe function receives registrations from ProSe-enabled UE and applications. The transition to ProximityRequest state occurs, when the ProSe Application sends a request to report proximity. In IdentityMapping state, the ProSe function requests from the App Server to assign respective UE IDs for the ProSe users. In ProximityValidation state, the proximity request is validated. In LocationReporting state, UEs' locations are reported. The transition to ProximityAlerting state occurs when the ProSe function detects that the UEs are in proximity, and the ProSe function notifies the ProSe Application and the respective UEs.

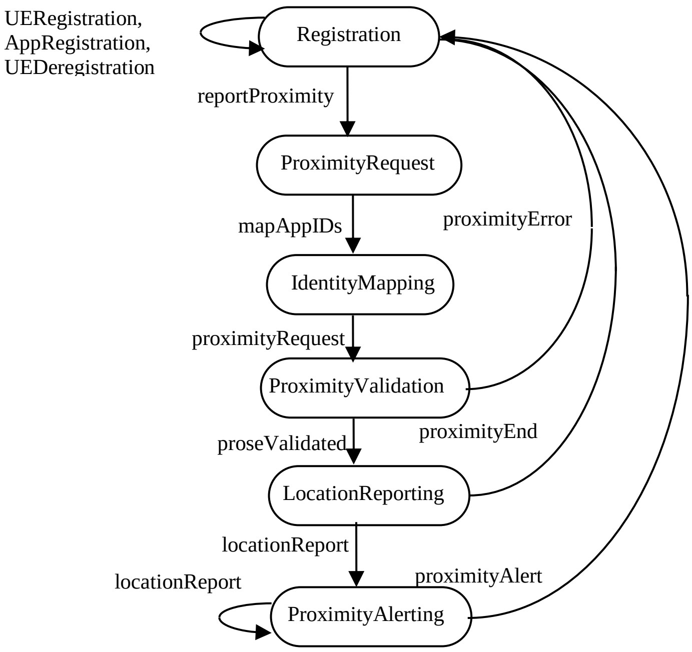

Fig. 5. ProSe function view on the state of Proximity-based Service

A detailed description of network assisted ProSe discovery may be found in [9].

Formal description of state machines is provided using the notation of Labelled Transition Systems (LTS).

Definition: A *Labelled Transition System* (LTS) is a quadruple (S, Act, $\rightarrow$, s_0), where S is countable set of states, Act is a countable set of elementary actions, $\rightarrow \subseteq S \times Act \times S$ is a set of transitions, and $s_0 \in S$ is the set of initial states.

By $T_{App} = (S_{App}, Act_{App}, \rightarrow_{App}, s_0^{App})$ it is denoted an LTS representing the ProSe Application view on the ProSe state, where:

S_{App} = {Null [s_1^A], InitiateProximityDiscovery [s_2^A], ReportProximity [s_3^A]};

Act_{App} = { notifyUERegistration [t_1^A], notifyAppRegistration [t_2^A], notifyUE-Deregistration [t_3^A], notifyRANUserPlaneCongestion [t_4^A], proximityRequestAck [t_5^A], proximityError [t_6^A], proximityEnd [t_7^A], notifyProximityAlert [t_8^A]};

$\rightarrow_{App}$ = { (s_1^A t_1^A s_1^A), (s_1^A t_2^A s_1^A), (s_1^A t_3^A s_1^A), (s_1^A t_4^A s_2^A), (s_2^A t_5^A s_3^A), (s_2^A t_6^A s_1^A), (s_3^A t_7^A s_1^A), (s_3^A t_8^A s_1^A) };

s_0^{App} = {Null}.

Short notations of state and transition names are given in brackets.

By T_N= (S_N, Act_N, $\rightarrow_N$, s_0^N) it is denoted an LTS representing the ProSe function view on the ProSe state, where:

S_N= {Registration [s_1^N], ProximityRequest [s_2^N], IdentityMapping [s_3^N], ProximityValidation [s_4^N], LocationReporting [s_5^N], ProximityAlerting [s_6^N]};

Act_N = { UERegistration [t_1^N], AppRegistration [t_2^N], UEDeregistration [t_3^N], reportProximity[t_4^N], mapAppIDs [t_5^N], proximityRequest [t_6^N], proseValidated[t_7^N], locationReport[t_8^N], proximityAlert [t_9^N], proximityEnd [t_{10}^N], proximityError [t_{11}^N] };

$\rightarrow_N$ = { (s_1^N t_1^N s_1^N), (s_1^N t_2^N s_1^N), (s_1^N t_3^N s_1^N), (s_1^N t_4^N s_2^N), (s_2^N t_5^N s_3^N), (s_3^N t_6^N s_4^N), (s_4^N t_7^N s_5^N), (s_4^N t_{11}^N s_1^N), (s_5^N t_8^N s_6^N), (s_6^N t_8^N s_6^N), (s_5^N t_{10}^N s_1^N), (s_6^N t_9^N s_1^N) };

- s_0^N = {Registration}.

Having formal description of the models representing ProSe status as seen by ProSe Application and ProSe function, we can prove that these models are synchronized i.e. they expose equivalent behaviour.

Intuitively, in terms of observed behaviour, two LTSs are equivalent if one LTS displays a final result and the other LTS displays the same result. The idea of equivalence is formalized by the concept of bisimilarity [11]. In practice, strong bisimilarity puts strong conditions for equivalence which are not always necessary. Weak bisimilarity allows internal transitions to be ignored.

Proposition: T_{App} and T_N are weakly bisimilar.

Proof: To prove the bisimilarity between two LTSs, it has to be proved that there exists a bisimilar relation between their states. We identify the following relation between the states of of T_{App} and T_N: Null and Registration, InitiateProximityDiscovery and ProximityRequest, ReportProximity and LocationReporting. This relation is denoted by U_{AppN} where U_{AppN} = {(s_1^A, s_1^N), (s_2^A, s_2^N), (s_3^A, s_5^N)}. Then:

1. The ProSe Application is notified when a ProSe-enabled UE and/or application registers with the ProSe function: for (s_1^A t_1^A s_1^A), (s_1^A t_2^A s_1^A) ∃ (s_1^N t_1^N s_1^N), (s_1^N t_2^N s_1^N).
2. The ProSe Application is notified when a ProSe-enabled UE is deregistered: for (s_1^A t_3^A s_1^A) ∃ (s_1^N t_3^N s_1^N).
3. When the ProSe Application is notified about network congestion it sends a proximity request to the network: for (s_1^A t_4^A s_2^A) ∃ (s_1^N t_4^N s_2^N).
4. The proximity request is validated: for (s_2^A t_5^A s_3^A) ∃ (s_2^N t_5^N s_3^N), (s_3^N t_6^N s_4^N), (s_4^N t_7^N s_5^N).
5. The proximity request is not validated: for (s_2^A t_6^A s_1^A) ∃ (s_2^N t_5^N s_3^N), (s_3^N t_6^N s_4^N), (s_4^N t_{11}^N s_1^N).
6. The ProSe Application is notified when the ProSe function detects that UEs are in proximity: for (s_3^A t_8^A s_1^A) ∃ (s_5^N t_8^N s_6^N), (s_6^N t_8^N s_6^N), (s_6^N t_9^N s_1^N).
7. The ProSe Application is notified that the proximity reporting is ended: for (s_3^A t_7^A s_1^A) ∃ (s_5^N t_{10}^N s_1^N).

Therefore T_{App} and T_{N} are weakly bisimilar.■

6. CONCLUSION

In this paper, we propose a Web Service that provides open access to Proximity-based service in the network. The proposed "Application-driven offloading" Web Service allows 3rd party applications to be notified about congestion in the cellular network and to initiate UE's proximity discovery in order to trigger D2D communications and offload the network. The research novelty is in delegating the control of ProSe functions to 3rd party applications. With existing standardized solutions it is the UE that decides when to switch off to D2D communications. Delegating the control to dedicated applications allows a service provider or network operator to define a policy for traffic offloading, e.g. based on radio access network congestion level, device location or even based on the available credit in case of prepaid communications.

The open access to ProSe functionality might be seen as an approach that is three-fold beneficial. The approach gain for operators is consisted mainly by its inherent capability to mitigate the congestion-related situations by additional offload procedures. The applications i.e. the third party might be involved in the very same process while pursuing the fulfilment of own quality of service promises given, especially regarding losses and latency. Finally, the indirect aspect of improvement is the one about the end-user quality of experience which is going to be affected inevitably. All this has a price of procedures complexity increment within the operator's network but it seems that gain outweighs it.

Our future work will be related to definition of RESTful interfaces for open access to Proximity-based service in order to study capabilities for deployment of the service in Multi-access Edge Computing environment.

REFERENCES

[1] Kar, U. N., Sanyal, D. K. “An overview of device-to-device communication in cellular networks,” *ScienceDirect, ICT Express*, October, 2017, pp.1-6.

[2] Park, G. S., Kim, W., Jeong S. H., Song, H. "Smart Base Station-Assisted Partial-Flow Device-to-Device Offloading System for Video Streaming Services," in *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2639-2655, Sept., 2017.

[3] Pan, Y., Pan, C., Zhu, H., Ahmed, Q. Z., Chen M., Wang, J. "Content offloading via D2D communications based on user interests and sharing willingness," *2017 IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1-6.

[4] Wang, T., Sun, Y., Song L., Han, Z. "Social Data Offloading in D2D-Enhanced Cellular Networks by Network Formation Games," in *IEEE Transactions on Wireless Communications*, vol. 14, no. 12, pp. 7004-7015, Dec. 2015.

[5] Liu, C., He, C., Meng, W. “A tractable Multi-RATs offloading Scheme on D2D Communications,” *IEEE Access*, September, vol 5, 2017, pp.20841-208-51.

[6] Abbas, N., Dawy, Z., Hajj, H., Sharafeddine S., Filali, F. "Traffic offloading with maximum user capacity in dense D2D cooperative networks," *IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1-6.

[7] Shang, B., Zhao L., Chen, K. C. "Operator’s Economy of Device-to-Device Offloading in Underlaying Cellular Networks," *IEEE Communications Letters*, vol. 21, no. 4, pp. 865-868, April 2017.

[8] Chen, B., Yang C., Molisch, A. F. "Cache-Enabled Device-to-Device Communications: Offloading Gain and Energy Cost," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4519-4536, July 2017.

[9] 3GPP TS 23.303 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Proximity-based services (ProSe); Stage 2, (Release 14), v14.1.0.

[10] Mohammadi, M. “Service oriented Architecture: A review and Evaluation of Reference Models,’ *International Journal on Information Technologies & Security*, № 4 (vol.9), 2017, pp.37-50.

[11] Pola, G., Manes, C., Schaft A., Di Benedetto, M. D. "Bisimulation Equivalence of Discrete-Time Stochastic Linear Control Systems," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, 2017, pp. 1-1.

Information about the authors:

Evelina Pencheva is with the Faculty of Telecommunications, Technical University of Sofia. She has a DSc degree in communication networks. Currently, she is Professor and her scientific research area covers multimedia networks, telecommunication protocols, and service platforms.

Denitsta Kireva is with the Faculty of Telecommunications, Technical University of Sofia. Currently, she is Assistant Professor and her scientific research covers IT and communications.

Ivaylo Atanasov is with the Faculty of Telecommunications, Technical University of Sofia. He received his DSc degree in communication networks. Currently, he is Professor and his scientific research area covers mobile networks, internet communications and protocols, and mobile applications.

Ventsislav Trifonov is with the Faculty of Telecommunications, Technical University of Sofia. He received his PhD degree in security systems. Currently, he is Associate Professor and his scientific research area covers big data analytics and autonomic behaviour.

Manuscript received on 27 December 2017