

Supporting Content, Context and User Awareness in Future Internet Applications

Paweł Świątek, Krzysztof Juszczyszyn, Krzysztof Brzostowski,
Jarosław Drapała, and Adam Grzech

Institute of Computer Science, Wrocław University of Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{pawel.swiatek, krzysztof.juszczyszyn, krzysztof.brzostowski,
jaroslaw.drapala, adam.grzech}@pwr.wroc.pl

Abstract. We show how the applications utilizing a Future Internet architecture can benefit from its features like quality of service (QoS) provisioning and resources reservation. We demonstrate, how proposed applications address content, context and user awareness basing on the underlying Next Generation Network (NGN) infrastructure and how it can be used to host service-based applications.

Keywords: Future Internet, Next Generation Networks (NGN), distributed applications, quality of service (QoS), content-awareness, context-awareness, services personalization, service oriented architecture (SOA).

1 Introduction

One of the main motivations for designing new architectures for the Future Internet is to meet challenges imposed on the ICT infrastructure by new applications. These challenges include among others:

1. *Content awareness* – meaning the sensitivity of data processing and transmission methods to the content being delivered to the end-user. Content awareness may emerge in: different processing of various data streams (i.e. video encoding or sensor data encryption) and different forwarding methods (e.g. routing) for various streams.
2. *Context awareness* consisting in different treatment (in terms of forwarding and processing methods) of traffic depending on the particular use-case scenario of application generating this traffic. Context may be connected for example with the type of networking device used by a user or users geographical localization.
3. *User awareness* understood as personalization of services delivered to end-user. Personalization is achieved by means of proper choice of data processing and transmission methods according to functional and non-functional requirements stated by the user. Users requirements may be formulated explicitly or be a result of automatic recommendation which is based on the history of the application usage.

4. *Sensor networks and applications* covering such applications such as: smart energy metering, vehicle networks, intelligent building infrastructure, telemedicine, etc. Each particular telemetry application involves specific types of data processing methods and transmission of large number of small portions of data often requiring real-time or near real-time end-to-end performance.

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

Currently there is a number of approaches aiming at meeting the assumptions of the Future Internet. Depending on the proposed ICT architecture some or all of them are assured by utilization of proper networking techniques and concepts. These approaches differ among other in the layer in which the new Future Internet functionalities are accomplished. As an example consider Content Centric Network (CCN) proposed by Van Jacobson [10] where content delivery mechanism (e.g.: caching, forwarding, security, etc.) are mostly implemented at lower network layers. This revolutionary post-IP approach requires the entire networking protocol stack to be redesigned. On the other hand in the prototype of the Parallel Internet CAN (Content-Aware Network) [2] being one of the results of the Polish national project "IIP" (polish acronym for Future Internet Engineering) [4] the content is encapsulated in new frames format, but signaling messages are passed with use of the IPv6 (Internet Protocol version 6) protocol and content delivery routes calculation and caching is accomplished at the application layer. These two approaches are candidates for implementing the concepts of the internet of content and media.

One of the most mature architectures for the Future Internet is the Next Generation Network (NGN) [9]. The NGN signaling system in conjunction with the Differentiated Services (DiffServ) quality of service assurance model [3] and IPv6 networking protocol stack allows for implementation of converged all-IP multi-domain network conforming with all assumptions of the Future Internet. Sample implementation of this approach is the Parallel Internet IPv6 QoS prototype [15,16].

The main contribution of this work is the concept of how to achieve content, context and user awareness in the IPv6 QoS architecture by proper signalization in the service and transport stratum of the NGN architecture. The proposed concept is illustrated on exemplary applications designed for the IPv6 QoS system. In section 2 we give a brief overview of the architecture of the IPv6 QoS system with special focus put on the service stratum signaling system in section 2.2. Next, in section 3 we present exemplary applications and show how content, context and user awareness is achieved with use of service stratum signaling. Additionally, we show in section 3.3 how custom personalized application can be designed with use of processing and communication services already existing in the system. In section 4 we conclude our work and point out directions of future works.

2 Systems Architecture

2.1 IPv6 QoS System

In this work we consider an IPv6 QoS system architecture developed in the Polish national project IIP [4]. In this architecture it is assumed that the system consists of multiple layers each of which provides certain functionalities to the adjacent upper layer. The first layer is a physical network infrastructure which with use of virtualization techniques [7] provides to the second layer virtualized networking environment with dedicated communication and processing resources. Such virtualization allows for coexistence of multiple isolated virtual networks (called parallel internets - PI), characterized among others by different frame formats, protocol stacks and forwarding capabilities, in a single physical infrastructure.

IPv6 QoS system is one of parallel internets existing in a virtual networking environment. In general the IPv6 QoS architecture is based on coupling of the DiffServ quality of service assurance model and NGN signaling system. DiffServ is responsible for delivery to traffic flows generated by users required level of the quality of services by means of flow admission control, classification of flows to predefined traffic classes and processing of aggregated flows from different traffic classes. The NGN signaling system is used to provide end-to-end QoS guarantees by reserving necessary amount of communication resources to each particular connection request. Reservation of communication resources is performed by assignment of the request to proper DiffServ traffic class, which meets the QoS requirements for this flow.

The purpose of signaling in NGN is twofold. The first one is to reserve required communication resources and to establish an end-to-end connection between a pair of hosts in the system. This signaling is performed at the network layer in so-called *transport stratum*. Second type of signaling is performed at the application layer (*service stratum*). Service stratum signaling is in general an application specific signaling (e.g. SIP signaling) the aim of which is to configure distributed modules of an application and to process information necessary to send to transport stratum a request for communication resources reservation. Signaling can be also viewed as a middleware which separates the networking layer functionalities and application domain-specific functionalities.

2.2 Service Stratum Signaling

The task of service stratum signaling is to control the execution of distributed applications and to pass communication resources reservation requests from applications to the network. Service stratum being an intermediate layer between applications and the network which translates application specific signaling and negotiations to uniform service stratum – transport stratum interface allows for implementation of arbitrary application specific signaling schemes. This in turn allows achieving content, context and user awareness by implementation of specialized services' management mechanisms, whose task is to transparently compose and control execution of personalized complex services based on functional and non-functional requirements.

In our approach, based on the Service Oriented Architecture (SOA) paradigm, we assume that applications in the IPv6 QoS system consists of distributed loosely coupled modules (called atomic services). Execution of each application use-case scenario is performed by sending a request to an application server, which composes a specialized complex service from the available in the system atomic services. Additionally, we assume that end-to-end communication services provided by the IPv6 QoS system are also treated as atomic services and can be utilized to deliver to the user requested complex services according to SOA approach.

In order to deliver to the users requested complex services a two-stage signaling in service stratum is proposed. The task of the first stage of signaling is twofold. Firstly, based on services available in distributed communication system, it allows to compose a complex service which conforms with functional and nonfunctional requirements [8]. Secondly, it notifies each module of distributed application how and with which module they should establish communication in order to deliver requested complex service. The aim of the second stage signaling is to negotiate the details of communication between each pair of atomic services taking part in complex service delivery. Communication details depend on the functionalities of communicating services and may concern among others: data formats, audio and video codec, required transport protocol, encryption, etc. Taking into account negotiated communication details and non-functional requirements concerning requested complex service proper end-to-end communication service is requested from the IPv6 QoS system for each pair of communicating atomic services. Note that thanks to the negotiation process requested end-to-end communication services depend on the context of communication and the content being transmitted resulting in fully personalized context and content aware complex services delivery.

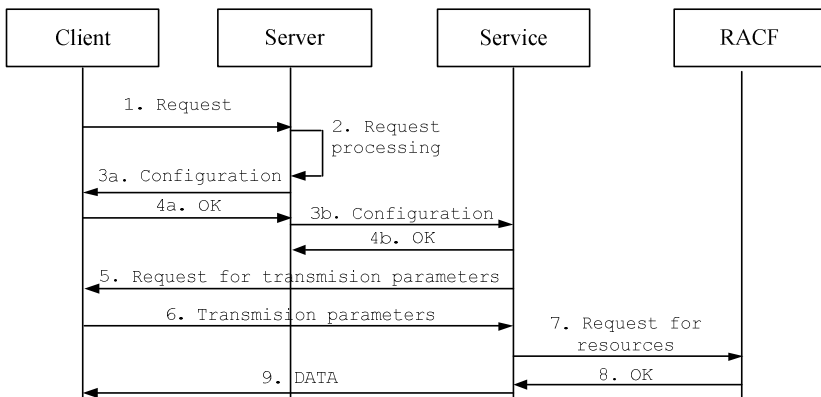


Fig. 1. Signaling messages exchange for an exemplary complex service

Fig. 1. Signaling messages exchange for an exemplary complex service. an exchange of signaling messages required to prepare an exemplary service and consisting in sending requested data from a service to a user is presented. In the first stage an arriving service request (1) is processed by server (2) in order to compose a

complex service conforming with users requirements. The result of a composition process is a set of atomic services available in distributed system which will take part in execution of request (1). In the second stage server configures all necessary atomic services (including client application) by sending them information about: source of data, data processing methods and destination of processed data (3a and 3b). In other words in this stage each service is notified from who and what will receive, what to do with received data and where to send it. After configuration is confirmed to the server (4a and 4b) each pair of communicating services negotiates values of application and communication specific parameters (5 and 6). When communicating parties agree upon type of communication appropriate end-to-end communication services (which guarantee delivery of required QoS parameters for data transfer) it is requested from the IPv6 QoS system. This is accomplished by sending a resources reservation request to the resources and admission control function (RACF) (7) with use of service control function (SCF). After confirmation of connection preparation (8) data transmission may take place (9).

The first stage of signaling beginning with sending of request (1) and ending with configuration of all services (4a and 4b) is accomplished with use of XML-RPC protocol. The second stage of signaling consisting of negotiation of values of communication and other application specific parameters (5 and 6) is accomplished with use of XMPP protocol. It is important to note, that vertical communication (signaling) of application components and network is done with use of service stratum – transport stratum interface (SCF – RACF to be exact). This means that each application component that is able to send requests to the network should be equipped with SCF module which translates application specific horizontal signaling to application independent uniform vertical signaling between service and transport stratum defined by SCF-RACF interface.

3 Future Internet Applications

3.1 SmartFit

Sustained progress and developing of infrastructure for wireless sensor networks and wearable sensors makes basis for pervasive computing systems. Such systems can operate in distributed environment where wireless sensor networks consist of huge amount of low-cost, low-power sensing nodes and many different services for data transfer, processing, storage and supporting decision making [1]. Sensing nodes can be used to build sensor networks such as Body Area Networks (BAN) or Personal Area Networks (PAN). On the other hand we have vast number of services in distributed environment facilitating the access to one or more functionalities.

SmartFit is a system adopting new technologies of pervasive computing and was designed to support endurance and technical training of either amateur and elite athletes. Application such as SmartFit must be designed to provide its functionalities “anywhere and anytime”. It means that acquired data must be transmitted between users of the system (i.e. athletes and trainers) with predefined quality level independently of their location. In order to fulfil this requirement each functionality was decomposed on small modules called atomic services. For each atomic service few different required levels of quality was defined. It means that we have different

versions of each atomic service. These different versions of atomic services are used in the process of user-centric functionality delivery with use of orchestration mechanism. User-centric functionality means that in order to compose such functionality user's specific requirements and needs are taken into account.

In Fig. 2 general architecture of SmartFit is presented. The first tier is used to sensor data acquisition. The second tier is data processing and decision making tier. The last one is presentation tier. For each tier the set of atomic services is defined. In the process of user-centric functionality composition all versions of atomic services at each tier are taken into account.

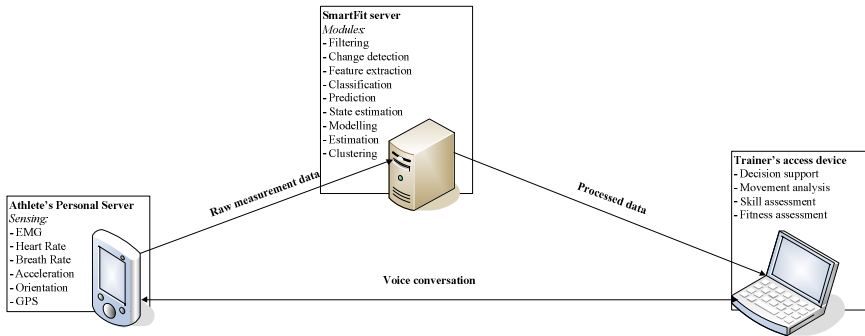


Fig. 2. Architecture of SmartFit system for distributed environment

One of the main functionality of the system is planning volume of endurance training. Because in all sports endurance training is required, this functionality is universal and can be used in the same way by athletes in various sports. The other functionality of the system is connected with endurance training monitoring. It is used to supervise correctness of the performed exercises, their right order and number of repetitions. Moreover, such functionality can be used to predict injury.

Another functionality of SmartFit system is designed to support technical training. On the contrary to endurance training, in this case, specific requirement for particular sport must be taken into account. Fortunately, architecture of SmartFit system provides mechanism for adding new functionalities supporting technical training of different sports easily. For an example use-case scenario of the SmartFit system a service for supporting a technical training in tennis was designed and implemented.

In Fig. 3 network topology configured to support technical training is presented. The main element of this network is SmartFit server which supervise the whole process of functionality delivery. This process has three phases. In the first of them the server must configure each necessary atomic service which was planned to be used in particular scenario. In considered example such atomic services facilitate physical and kinematic data acquisition from user's BAN, change detection and feature extraction for acquired data, classification based on results of feature extraction, modelling of human movements. The second phase is related to signalization between distributed modules with use of SIP and XMPP protocols. The last phase of the functionality delivery process is data transmission, processing with use of appropriate atomic services and presentation of obtained results.

The above-mentioned atomic services can be used to compose functionality supporting skill assessment and skill improvement of the elementary tennis strokes such as serve, backhand and forehand. Skill assessment allows to determine current skill level of tennis player. It can be helpful to make recommendations related to support planning of future technical training of elementary tennis strokes. Additionally, SmartFit is designed to support feedback training. It means that it is possible to see, in real-time, physiological and kinematic data from sensors placed on athlete's body during stroke performance. This feature allows tennis player and/or trainer to compare his performance to reference data acquired from his past trainings or high level elite tennis players.

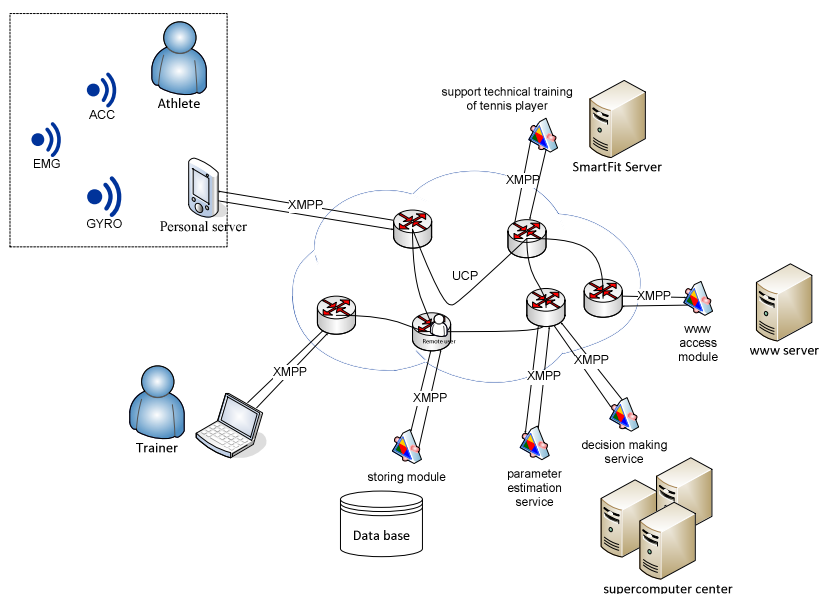


Fig. 3. Network topology supporting technical training scenario

Let us investigate the process of technical training of tennis player in details. In this use-case scenario for SmartFit athlete's BAN consists EMG, gyroscopes and accelerometers wireless units. Acquired physiological and kinematic data is transferred to the server. In skill assessment case signals from gyroscopes and accelerometers are processed in order to build relationship between wrist flexion, upper arm rotation and racquet speed. Obtained results can be compared with results captured from reference data of high level elite tennis player. Based on this data it is possible to build personalised model of improvement for a tennis player. It may be also used to make recommendation for future technical training.

In the second case feedback and learning by example trainings are combined in order to improve results of technical training. To this end acquired data from EMG, gyroscopes and accelerometers units are transferred to SmartFit server. EMG data is filtered in order to present to the trainer the sequence of muscle activation of upper

limb during strokes. Whereas signals from gyroscopes and accelerometers are used to estimate trajectory of upper limb movement during tennis strokes such as serve, forehand and backhand. In this case parameters of upper limb movement's model must be determined with use of delivered signals from gyroscope and accelerometers units. Finally, the results of trajectory estimation is visualised and delivery to the trainer and/or athlete.

In order to provide the required by the user quality of service it is necessary to apply mechanism allowing for context awareness. Context awareness incorporated in SmartFit can be used to adapt the packet size according to the user's requirements. Context information can be obtained through sensor networks e.g. measurement of heart rate during training session or/and personal server e.g. GPS or networking devices. Based on this information it is possible to predict user behaviour and his/her location. Such mechanism facilitating SmartFit system with functionalities for efficient management of network and computational resources in order to deliver system functionality with required by user's quality of service.

3.2 OnLine Lab

Virtual laboratory infrastructure should automate most tasks related to the reliable and reproducible execution of required computations [12,17]. The application Online Lab is a distributed, service-based computational laboratory benefiting from the IPv6 QoS architecture which is used to distribute computational tasks while maintaining the quality of service and user experience [13]. It allows its users, i.e. students or researchers, to access different kinds of mathematical software via Python math libraries and perform computations on the provided hardware, without the need for installing and configuring any software on a local computer. The communication mechanisms are designed for optimization of the users' Quality of Experience, measured by the response delay. The functionality of Online Lab embraces:

- access to computational services ensured by user's virtual desktop which is windowed interface opened in a Web browser,
- creation and removal of computational services with no limitations being assumed on the nature of computations – the users may freely program computational tasks in any language interpreted by running computational services,
- user profile maintenance and analysis – the users are distinguished by their profiles which hold information about their typical tasks and resource consumption.

Online Lab (OL) implements an architecture consisting of user interface (OL-UI), core server (OL-CORE), services and computational engines (OL-Services, based on the Python engine in the current prototype). OL-UI is a web service emulating a desktop and a window manager. Code is being typed into specialized data spaces - notebooks, which are executable documents executed by OL-Services.

The process of user's query execution is presented in Fig. 4. OL-Core and OL-Services belong to the service. One notebook represents one computational task. The system also may recommend notebooks of other users. The content of the notebooks is annotated with the help of domain (Math) ontology.

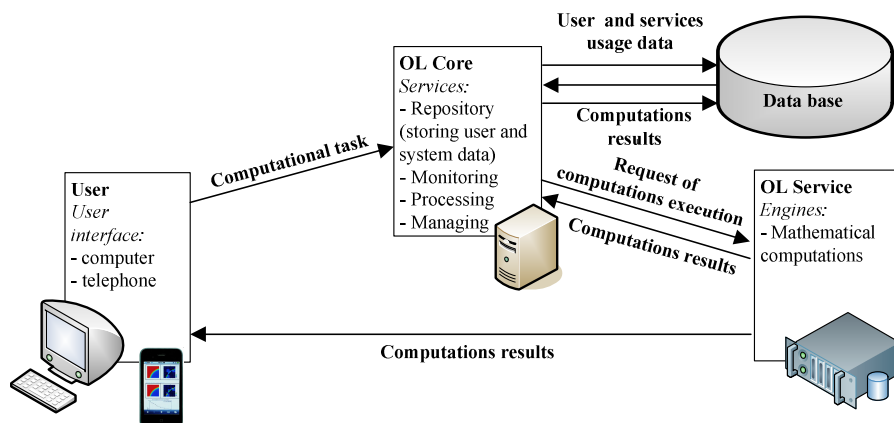


Fig. 4. The general schema of the Online Lab service execution

OL-Core is constantly monitoring the OL-Services, storing execution times and data transfers in its database. From the user point of view, in the case of computational tools, the key element of the Quality of Experience (QoE) is waiting time. The waiting time is the sum of computation time and communication times. The first is query-specific and must be taken into account as a value predicted on the basis of known history of user queries. The second depends on the volume of data and code.

Online Lab classifies user queries (computational tasks) and reserves communication services of the IIP system in order to guarantee the QoE for the user. The computational tasks are scheduled in order to minimize the waiting time of the user, which is done by computational service monitoring and dynamic configuration of communication links using the IPv6 QoS infrastructure. This approach is used to address the requirements defined in the introductory section:

1. *Content awareness* – such OL-Service is chosen to provide the minimum processing time. The data volume of the task influences the parameters used during the link reservation in IPv6 QoS system (to achieve the minimum transfer time).
2. *Context awareness* is maintained by the Load Balancer. Its task is to analyze the stream of requests and manage the negotiations with the service stratum. It is also equipped with the prediction module which forecasts the user behavior.
3. *User awareness*. The services are personalized, taking into account the user preferences, typical volumes of associated data and recommendation scheme.

Taking the above into account the general task of Online Lab is to compose a computational service, given the request stream from the users is known or predicted. All the components of the system (OL-Core and available OL-Services) are registered and have unique IDs. Once the optimal (with respect to the QoE) structure of this complex service (including the set of OL-Services and the parameters of communication links between them) is decided by the Load Balancer, the OL-Core reserves (via SCF functions, as described in sec. 2) the communication links connecting all the Online Lab services. This guarantees delivery of required QoS parameters. In the second phase the negotiation in the service stratum takes part to establish and confirm

pairwise communication between the services. After that the computational tasks are scheduled and assigned to the appropriate OL-Services by the OL-Core.

An additional unique feature of Online Lab is the possibility of implementing dedicated computational services which may be available to other applications. An example of this scenario will be sketched in the following section, where we describe the use of Online Lab service to be used by the SmartFit application.

3.3 Custom Application

The complex service model assumed for IPv6 QoS applications makes it easy to develop intelligent applications build on the basis of available communication and computational services. Intelligence lies in: distributed measurement data acquisition and processing, decision making support, ability to compose services tailored to the user's needs and profile.

The following atomic services are proposed as building blocks for complex services design. Computational services are: System Identifier (*SI*) to fit mathematical model [5] to the user, System Simulator (*SS*) that makes use of the user model to predict the user reactions for different activities, System Controller (*SC*) that gives advices to the user, Training Protocol Optimizer (*TPO*) that works out the best training protocol for the task the user is about to perform. Communication services exchange data between measurement devices, services and users end-points and, if necessary, perform additional actions, such as encryption. Using the set of atomic services, the systems is able to deliver advanced functionalities.

The system supports the user in controlling intensity of an exercise. In this custom application the data processing needed to support real-time training procedure is performed by a dedicated OL-Service which serves as a decision-making component of SmartFit application.

Typical usage scenario starts from calling Training Protocol Optimizer (*TPO*) that generates optimal training protocol and passes it to the System Controller (*SC*) and the System Simulator (*SS*). The System Controller (*SC*) uses Heart Rate signal, sent by the measurement device, to support the user in maintaining exercise intensity on the desired level. The System Simulator (*SS*) compares the user model response to signals obtained from measurement devices. After an exercise is finished, measurements are sent to the System Identifier (*SI*) to validate the model and update it.

Configuration of services depends on the sport discipline practiced by user (*context awareness*). The system reacts to long term effects caused by systematic training by adjusting the user model and the choice of services he/she needs (*user awareness*).

Note, that the above scenario is just an example of personalization scheme which relies on composition of new services on the basis of atomic services (*SI*, *SS*, *SC*, *TPO*). The final functionality of the resulting complex service is suited to the user requirements. New services, originating from different applications atomic services, may be also included and taken into account when composing complex services. For instance, when we want to deliver the same application for a diabetic user, we may compose a new service that supports physical training, taking blood glucose level into account (Fig. 5), [11]. In such a case, additional service to keep blood glucose level within normal range is executed and additional requirement to blood glucose data encryption (*content awareness*) is imposed. Additional constraint for *TPO* is defined to make sure that the training scenario generated by *TPO* will be safe for the user.

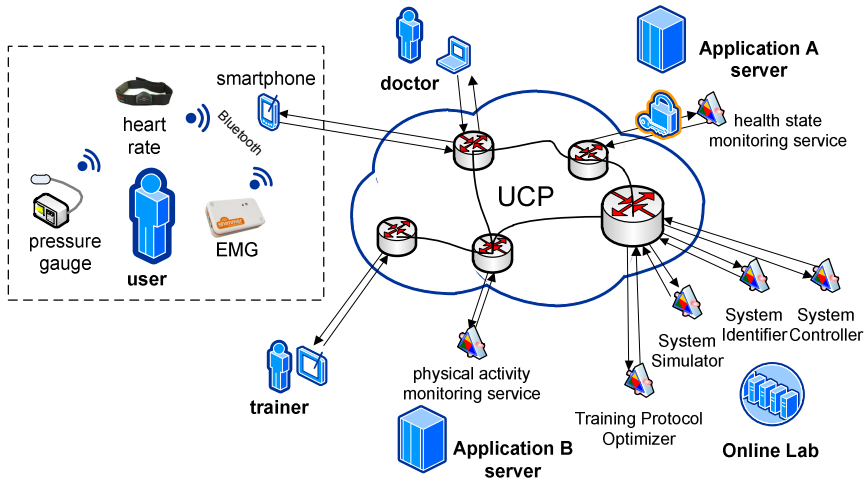


Fig. 5. Network topology for exemplary application

Described application provides context, content and user aware adaptive control system supporting the user in a real-time.

4 Conclusions

In this work we presented a general idea of delivering to end-user complex services in a distributed networking environment. The main feature of the proposed idea is that the process of complex services delivery is aware of the content being delivered, the context of the services delivery and that the delivered services are personalized for each separate end-user. In order to achieve the content, context and user awareness we proposed a general scheme for signaling system, which task is to configure distributed application modules and network resources with respect to the requirements imposed by the content being delivered, context of services delivery and specific user's needs.

The proposed signaling system was designed and implemented as a middleware between the end-user and the network, more specifically as a service stratum in the NGN architecture of IPv6 QoS system. Note, however, that the signaling system architecture does not assume any specific network architectures. The idea is to utilize communication services provided by the network layer to provide fully customizable application layer services built from computational and communication resources available in the distributed system. This means that if the application use case scenario does not require any specific quality of communication services it can be delivered to the user with use of nowadays best-effort IPv4 internet network infrastructure.

The IP QoS infrastructure and all applications presented in this work are results of the IIP project [4]. The ongoing research efforts are devoted to further integration of service-based applications within this scheme, and the testing of new software-building paradigms and concepts stemming from service customization and composition approach illustrated in sec. 3.3.

Acknowledgements. The research presented in this paper was supported by the European Union within the European Regional Development Fund program no. POIG.01.01.02-00-045/09.

Open Access. This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Alemdar, H., Ersoy, C.: Wireless Sensor Networks for healthcare: A Survey. *Computer Networks* 54, 2688–2710 (2010)
2. Bęben, A., et al.: Architecture of content aware networks in the IIP system. *Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne* 84(8/9), 955–963 (2011) (in polish)
3. Blake, S., et al.: An architecture for differentiated services. RFC2475 (1998)
4. Burakowski, W., et al.: The Future Internet Engineering Project in Poland: Goals and Achievements. In: *Future Internet Poland Conference, Poznan, Poland* (October 2011)
5. Cheng, T.M., et al.: Nonlinear Modeling and Control of Human Heart Rate Response During Exercise With Various Work Load Intensities. *IEEE Trans. on Biomedical Engineering* 55, 2499–2508 (2005)
6. Cobelli, C., et al.: Diabetes: Models, Signals, and Control. *IEEE Reviews In Biomedical Engineering* 2, 54–96 (2009)
7. Mosharaf Kabir Chowdhury, N.M., Boutaba, R.: A survey of network virtualization. *Computer Networks: The International Journal of Computer and Telecommunications Networking* 54(5), 862–876 (2010)
8. Grzech, A., Rygielski, P., Świątek, P.: Translations of Service Level Agreement in Systems Based on Service-Oriented Architectures. *Cyb. and Systems* 41, 610–627 (2010)
9. ITU-T Rec. Y. Functional requirements and architecture of next generation networks (2012)
10. Jacobson, V., Smetters, D.K., Thornton, J.D.: Networking Named Content. *CACM* 55(1), 117–124 (2012)
11. Man, C.D., et al.: Physical activity into the meal glucose-insulin model of type 1 diabetes: in silico studies. *Journal of Diabetes Science and Technology* 3(1), 56–67 (2009)
12. Noguez, J., Sucar, L.E.: A Semi-open Learning Environment for Virtual Laboratories. In: Gelbukh, A., de Alborno, Á., Terashima-Marín, H. (eds.) *MICAI 2005. LNCS (LNAI)*, vol. 3789, pp. 1185–1194. Springer, Heidelberg (2005)
13. Pautasso, C., Bausch, W., Alonso, G.: Autonomic Computing for Virtual Laboratories. In: Kohlas, J., Meyer, B., Schiper, A. (eds.) *Dependable Systems: Software, Computing, Networks. LNCS*, vol. 4028, pp. 211–230. Springer, Heidelberg (2006)
14. Świątek, J., Brzostowski, K., Tomczak, J.: Computer aided physician interview for remote control system of diabetes therapy. In: *Advances in Analysis and Decision-Making for Complex and Uncertain Systems, Baden-Baden, Germany, August 1-5*, vol. 1, pp. 8–13 (2011)
15. Tarasiuk, H., et al.: Provision of End-to-End QoS in Heterogeneous Multi-Domain Networks. *Annals of Telecommunications* 63(11) (2008)
16. Tarasiuk, H., et al.: Performance Evaluation of Signaling in the IP QoS System. *Journal of Telecommunications and Information Technology* 3, 12–20 (2011)