PRE-PAID CHARGING SYSTEM FOR SIP-P2P COMMERCIAL APPLICATIONS

Damian Nowak[1] and Krzysztof Walkowiak[2]

[1]Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
damian.nowak.ext@googlemail.com
[2] Department of Systems and Computer Networks, Wroclaw University of Technology, Wroclaw, Poland
krzysztof.walkowiak@pwr.wroc.pl

ABSTRACT

The new SIP-P2P approach brings many advantages like the improved reliability and the ease of setup. However, Communications Service Providers (CSPs) offering VoIP services seem to be afraid to lose their profits due to the introduction of SIP-P2P. Claimed is the lack of session establishment control leading to revenue leakages. In this work we introduce a charging system for SIP-P2P that assures CSPs' revenue from using SIP-based peer-to-peer communications commercially. The proposed solution consists of a new "carrier-grade P2P network" concept, software modification of existing equipment and gives some charging application design guidelines. An application prototype was implemented to examine the concept. Additionally, experiments were done to provide the methodology for system dimensioning.

KEYWORDS

SIP-P2P, Charging System, Voice over IP, Computer Networks, P2P Systems

1. INTRODUCTION

At present, most of Communications Service Providers (CSPs) offer voice services based on the Voice over IP (VoIP) concept using the Session Initiation Protocol (SIP). The current SIP implementations work in a way, that each client, before making a call needs to register his identity in a central database called SIP Proxy. When setting up the connection between communicating parties, the proxy is used to make a translation between the B-Party (the called party) SIP URI and its IP address. Additionally, all session signalling messages sent between parties are transferred via SIP Proxy. Such a design introduces a single point of failure (when proxy is down, all communications is down). What is more, capacity and performance is not scaling very well with the amount of subscribers and the effort to set up a VoIP service using SIP is higher, than if we were using e.g. Skype, due to necessity of SIP Proxy device.

One of the latest developments in the field of VoIP solutions is the Peer-to-Peer version of the SIP called SIP-P2P. The motivation to develop SIP-P2P follows from the drawbacks of the traditional SIP described above. The concept of SIP-P2P uses a distributed hash table [1] to store the mapping between SIP URIs, associated IP address and all the other needed information. The research community in SIP-P2P area is currently very active. One of the most promising open-source projects – "39peers" is being developed with the "Google Code" project support. Paper [2] written by the "39peers" project software architects can be used as a good starting point for research in this area. Most upcoming implementations use Chord as a DHT algorithm, which assures low search times and scales pretty well. The SIP-P2P offers then a really quick and standardized method of setting up a VoIP service and allows the operators to

lower OPEX due to the fact, that SIP Proxies and their maintenance are not necessary anymore. Additionally, distributed peer-to-peer networks nature eliminates the single point of failure problem valid for traditional, centralized implementations.

With all those advantages, SIP-P2P introduces some doubts for operators, who are afraid to lose their current revenues coming from VOIP services. Since all the signalling in traditional SIP needs to pass through the SIP Proxy, it is possible to control the session establishment and supervision there, with the help of the on-line charging system (OCS), especially for pre-paid charging. The authors decided to cover pre-paid charging only due to the fact, that its support is far more complicated, than the post-paid charging. For the pre-paid mode, the system should take into consideration reservation on the accounts and solves issues related to service sessions running in parallel that makes the solution proposal preparation a lot more challenging than for the post-paid systems that usually run in off-line mode. The OCS evaluates the tariff and makes the decision, whether the session could be established and if so, what are the applicable rules. With SIP-P2P, this approach is not valid anymore, because there are not SIP Proxies. That is why sessions must be controlled on the end-user devices. This paper will guide through one of possible solutions valid for commercial use. The complete system would be proposed, that addresses these issues. The main contributions of the paper are: the carrier-grade P2P network concept, OCS solution proposal for small CSP and its performance dimensioning guidelines.

The rest of the paper is organized in the following way. In Section 2, we present common OCS requirements coming from technology, economy and law areas. Section 3 gives the reader an overview on what was already achieved in the field of charging system for services delivered in P2P networks. The carrier-grade P2P network concept together with overall charging system design is described in Section 4. In Section 5, the charging server architecture is discussed and the application running on top of it is a subject of Section 6. Section 7 is dedicated to Diameter protocol and in Section 8 the dimensioning guidelines are discussed. Section 9 contains the summary of entire work.

2. COMMON ON-LINE CHARGING SYSTEM REQUIREMENTS

On-line charging system requirements are connected to three areas: technology, economy and law. In this chapter, we will present the most important and common requirements for all these areas. The technology area defines the following requirements: high-availability, soft real-time operation and scalability in terms of capacity and new services introduction. The high availability requirement covers the charging continuity in case of any software or hardware failure and software upgrade transparency for end-users. The common carrier-grade network elements reliability reaches 99,999% [3] and the charging system is composed out of at least three modules: session supervision device, charging application and interconnecting network. The total system availability (A) is calculated as the product of component's availability A_i for each component $i \in C$

$$A = \prod_{i \in C} A_i$$

Taking this formula into consideration, it is not possible to achieve fife-nines availability time for the entire charging system. That is why 99.99% availability is required, resulting into approximately one hour downtime per year. The system availability is affected when new resources should be added to the running system, e.g. a new server joins the cluster. The design of a system should ensure minimum service disruption during hardware and software updates.

The performance requirements are related directly to the nature of services that are offered to subscribers. The typical voice service requires the delays to be less than 200ms according to [4].

The detailed dimensioning has to be done to assure, that user session would not be disturbed even in high load conditions. Additionally, the maximum traffic limit should be established not to allow new sessions to be processed when resources are exhausted, to avoid decreasing the service quality for the running session. A similar approach is used in the ATM networks during session establishment procedure – CAC [5].

The software scalability is achieved with the use for modular and layered approach, in which all the software elements could be extended or exchanged transparently for subscribers. In this way, different level of price rating complexity can be suited to CSPs needs. Currently, most of the enterprise software is built inside the application servers dealing with non-business-related tasks and such a solution should be used to keep the industry standard objective.

The economy area defines the business requirements, which involve threshold administration (e.g. session credit threshold) and rating flexibility. The rating should support all the existing services delivered with SIP and be easily extendible to introduce new services and new charging models. That is why the modular and hierarchical tariff design is a must. Following the demands of international telecommunication market, the charging application should support at least: destination based rating, time windows handling, subscriber lifecycle supervision, geographical location evaluation and the possibility to apply discounts (e.g. family tariffs). There is no point in defining here the specific business use-cases, because the use for them is depending on the particular CSP strategy. The following parameters need to be taken into consideration, when executing rating: call start time, A-Party (calling party) and B-Party URIs, geographical location, time zone (when establishing a call abroad), service quality (e.g. for streaming media), time windows administrated by CSP, A and B-Party relations (e.g. same company).

The law area defines two kinds of requirements: first are related to data and transmission's security and the second are related to accounting. The security requirements mentioned here will not be further discussed since that area is out of scope of this paper. Anyway, the Polish law (complying with UE law) defines exactly the parameters that should be stored by the operator in a CDR (Charging Data Record) related to voice service usage. The most important parameters are: E.164 numbers [6] of subscribers, consumed time, usage start time and the amount of money that was charged for the service. Typically this set of data is extended with additional parameters for statistics purposes. The proposed scope of parameters is as follows: location information, subscriber's expiry date, special offer name if one is used.

3. RELATED WORKS

The design of On-line Charging System (OCS) for services deployed using P2P networks was already presented in literature. The most important projects in this field seem to be MMAPS [7], Karma [8] and PeerMint [9]. These projects mostly aim at finding a reliable solution to solve fairness problem, common in P2P environments. The use of those systems effects in a fair balance between the data that is offered and downloaded in P2P networks. More than that, the authors' intention is to keep the charging system distributed, so the charging application along with accountancy service is spread among all the participating peers. Another feature covered by those solutions is a wide multiservice support, e.g. MMAPS solution could be used to charge for movies distribution as well as for CPU time usage for some calculations-hungry applications. In authors' opinion, the features offered by existing solutions are too sophisticated if commercial launch in multimedia environment is considered. VoIP calls supervision should be performed quick enough to make the subscriber not conscious, that the charging mechanisms work behind the scene, what results in target delays not higher than 200ms (what is human-noticeable). If we take into consideration the MMAPS project, in which the subscriber accounts are kept on at least 2 remote peers, then both of them need to be contacted during the session establishment procedure and each session reporting. The authors did a ping test in order to

check, how much time a simple message transfer lasts in typical, public IP network. The test PC was located in Wroclaw, Poland and the DNS to be reached was located in Boston, USA (Verizon Inc.). The average packet transmission time was ~150ms. If we compare that amount of time to the requested maximum total application execution time estimated to be 200ms, there is not a lot of time left to execute the code processing. That is why the existing implementations cannot be used due to their poor performance. The main problem seems to be multiple accounts synchronization. Fairness assurance implemented in Karma-based systems uses tokens to make sure, that the amount of data transmitted and received from network is balanced. However, this mechanism is completely not a case, when it comes to SIP-based multimedia services. All those circumstances make it worth and necessary to propose an OCS suitable for efficient multimedia services delivery. For more information on P2P systems refer to [10], [11].

4. P2P CARRIER-GRADE NETWORK AND OVERALL CHARGING SYSTEM DESIGN

The distributed P2P networks' nature makes them almost completely failure-proof and it is clear that this potential should be finally used commercially. However, P2P systems operate mostly in the end-user domain, which is perceived as not enough reliable from the CSPs' perspective. The main concern seems to be the end-user device software security in terms of storage and data transmission.

The proposed solution, which addresses these issues is a P2P network created and maintained entirely by the CSP. It acts here as a supplier of secure, peer devices and keeps full control over it. Then, the P2P network is built out of those peer devices, that are able to communicate with each other. The subscriber should not be allowed to get access to this extended functionality, especially to the contents and configuration of the network. From his perspective, "the black box" should allow him just to use the requested services and the way, how are they delivered should not be a point of interest. To implement these requirements, existing end-user devices (a good base could be a Wi-Fi router) should be extended with some additional functionality: SIP-P2P handler (in order to support the SIP session establishment and control over them), Diameter protocol (in order to provide a communication medium for A4C systems), encryption services and cognitive functions. What is more, "the box" could be further extended with a FemtoCell and other features, creating a reliable, common telecommunication services access point. Due to the fact, that the amount of the "boxes" in CSP network could be unmanageable without an automated management system, the SNMP v3 functionality is proposed to be implemented in the end-user devices. This standard protocol provides a high level of security due to the built-in encryption capabilities. The SNMP is already implemented in existing large-scale computer networks and supported by multiple network management systems (NMSes), so the reuse for it is a straight-forward decision.

With the use for proposed SIP-P2P-based system, some content and software is stored on the end-user devices that are typically installed in the subscriber premises. Due to this fact the data security must be taken into consideration. The authors identified three areas, where device security could lead to frauds: the content security (movies, music, etc.), the software security (deassembly prevention) and data transmission security (transmission secrecy). The most important area seems to be the device's software and data transmission, because any security violation will make it possible to make the security methods and system design public resulting in massive frauds. That is why the security in these areas should be protected by all known methods. The authors propose to achieve that with the use for data and transmission encryption. Corresponding solutions are discussed in many papers and there are already some open-source implementations that could be used out-of-the-box. The Diameter protocol [12-14] posses already some built-in encryption features that could be used. Another level of security (misuse

prevention) could be introduced with the use for the so-called "cognitive functions" [15]. A cognitive device is capable of localizing itself in the scene, in which it resides. With this knowledge, some decisions could be done, taking into consideration information retrieved from e.g. video camera or GPS receiver. With such capabilities, the device could notice if the operations taking place in its environment have some negative impact on its security. The example could be the situation, when someone tries to open the device enclosure. This should be a clear message to delete permanently all the data gathered on the device to make sure, that the software inside the box would not be analyzed and the media stored there are not distributed in unauthorized way. Another idea we propose to prevent frauds is not to keep the entire entity of content (e.g. a full movie) on a single end-user device. In this way, someone who opens the "box" will not be able to extract full content out of it. The proposed end-user device software stack resulting from the provided description is shown on Figure 1.

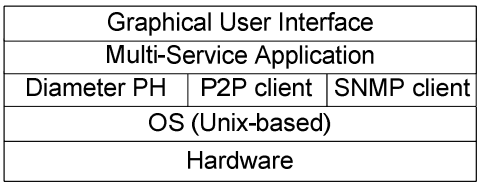

Figure 1. Layer design of P2P-enabled end-user device

To assure the proper service usage charging, peers are equipped with Diameter protocol clients and encrypted connections would be established to Diameter relays. The Diameter protocol is one of A4C protocols (Accounting, Authentication, Authorization, Audit and Charging). In many networks its predecessor, RADIUS is still used, but due to its limitations (e.g. unreliable message transfer, missing encryption and mobile user support capabilities, etc.) it is replaced by the successor. The decision to use the Diameter protocol was taken, because during last years, it became a true industry standard (3GPP standardization as the basic NGN signalling protocol) already implemented by many CSPs around the world (e.g. Vodafone group). Its popularity and standardization efforts make it easy to integrate the protocol in live network due to the fact, that the supporting network elements are available on the market. Furthermore, no extensive training is necessary and troubleshooting could be done with the use for standard tools (e.g. WireShark protocol analyzer). The Diameter Relays are introduced to reduce the amount of resources that would be necessary on the charging system front-end to handle the messages coming from potentially millions of peers. The charging load would be then transmitted to the charging system front-end, which is working as a message router and mediation device. With the use of relays, the CSP is able to aggregate the service charging traffic coming from peers existing in the relay coverage area with only one physical Diameter connection. The complete charging system architecture is shown on Figure 2.

There is one additional feature of the proposed solution worth to mention, i.e. software updates. When the charging software is considered, the software updates are crucial for assuring a correct business handling. With the proposed CSP-managed P2P network, it is possible to trigger the software update on millions of devices in a synchronized manner without any user interaction. To make sure, that the device always uses the recent software release, there should be an update-check done periodically by the device and at each startup. A potential risk here is related to the fact, that the device could be disconnected from the power source during the update and become completely unusable. That situation could be prevented by duplicating the software storage device as it is done in some PC motherboards (e.g. Gigabyte ones) currently available on the market.

Despite the power outage issue, there are also other security violation possibilities related to software updates. First of all, the release of software that has to be installed on the end-user device could be intercepted by the end-users. Additionally, in principle, the end-user is able to set up a software update server on a machine that is not controlled by the CSP. This will give the end-user the possibility to upload to his "box" a software release that contains some faults or unauthorized software what will make the frauds possible. There are also many other issues related to security, however there are out of scope of this paper.

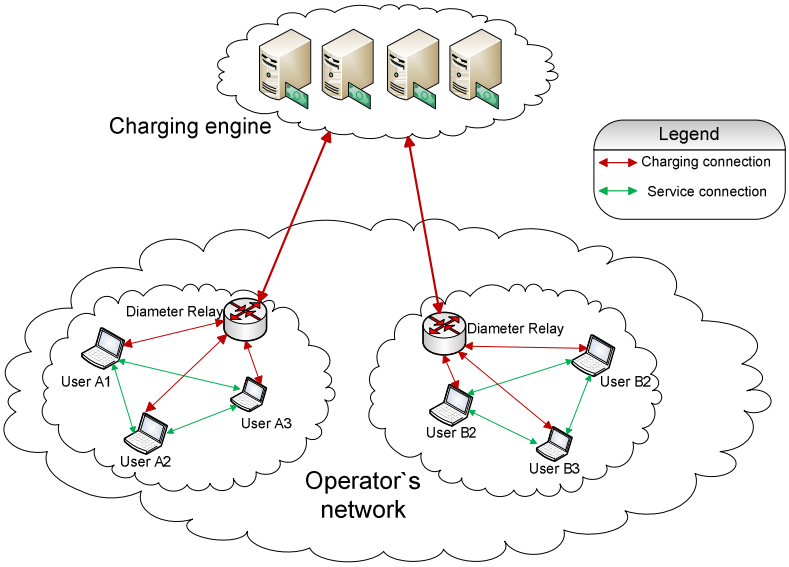

Figure 2. Overall charging system design

5. CHARGING SERVER DESIGN

In this chapter we focus on the charging server design. We propose two implementations suited to the needs of two different groups of CSPs: "small" CSPs (est. up to 500k subscribers) and "big", nation-wide CSPs (est. millions of subscribers). These two approaches are presented due to the fact, that the small CSPs mostly rely on open-source/freeware software to lower the costs. That is why the first approach will focus on delivering the demanded features using the best of available free software. On the other hand, nation-wide CSPs demand performance together with scalability and the price constraint. Thus, for nation-wide CSPs more sophisticated design will be proposed. Although the physical design of the charging system differs in terms of the potential CSP's scale (i.e. number of subscribers), the logical design presented on Figure 3 is mostly the same.

The basic software stack layer is composed of an operating system with a clustering platform. This layer differs the most between free and commercial solutions ranging from simple Linux system with open-source clustering platforms to the commercial high-available clusters. There are many publications available describing the clustering technologies, e.g. [16]. The external communication support module is in fact implemented as a mediation device. It should support a range of A4C and other protocols used for system configuration, maintenance and interoperation with 3rd party systems. The set of protocols utilized is composed of the Diameter protocol (session control), Corba-based protocols (interoperation between OCS and other operator's systems, e.g. recharging subsystem), http (basic data management) and others. The main aim of the external communications module is to translate different protocols to the internal, common message format that could be understood by all the elements of the system,

e.g. by the charging application. It is crucial for this module for flexibility reasons to be further divided into easily replaceable components (e.g. protocol handlers) allowing the CSP to adjust the system interface to the changes of his infrastructure and the industry's raising standards. The DataBase (DB) support module contains a RDBMS suited to the needs of particular operator. Small CSPs may choose simple MySQL/Postgres DBs, and nation-wide CSPs could make use out of complex (and expensive) solutions from Sun, Oracle and Microsoft. As the reader can see, there is no direct connection between the "External communication support module" and the database, meaning that all the requests coming to the server should be processed by the application before anything would be stored permanently in the DB. This design decision was made in order for the charging system to defend itself from the possible bugs in other systems. Let's assume that the subscribers found the way to recharge their account multiple times with the same voucher (what is a fault of the voucher management system, an external device from the charging system perspective). Then, without the message supervision functionality, there will be no possibility to implement a check against the maximum amount of the account. The data administration is implemented using the Web interface that is easily accessible from almost any PC out-of-the-box. Depending on the final requirements yet another device could be proposed exclusively for data administration (on-line/off-line mode). As shown on Figure 3, the subscriber profile and the tariff data administration were separated due to security and performance reasons. The charging application would a subject of separate chapter of this paper.

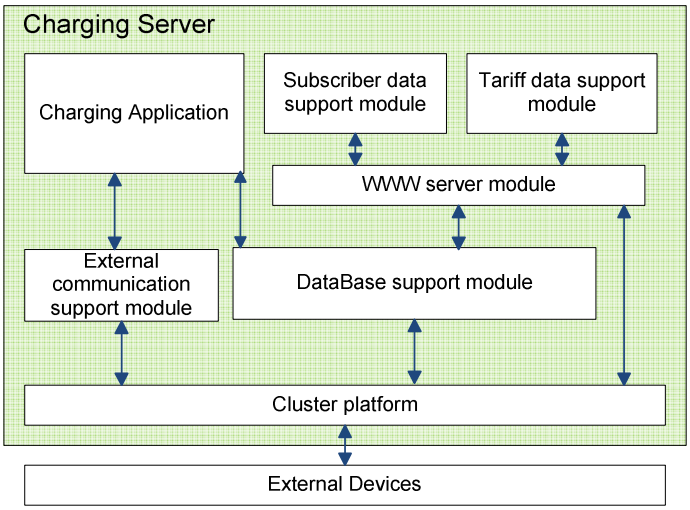

Figure 3. Overall logical charging server design

5.1. Charging Server for Small CSPs

The proposed design is oriented on providing reliable charging service to the subscribers. The performance constraint is not the dominating one here. The proposed software stack is presented on Figure 4. The base is built out of free Linux Gentoo operating system together with HeartBeat 2.1.4 clustering platform working with DRBD storage. The system was built by the authors in VMWare environment, where virtual PCs were connected through VMWare's internal network. As shown on Figure 4, two logical block storage devices were created ("DB" and "WEB") that were constantly replicated between both machines. The DRBD's "C protocol" was used, that assures, that data are always written on both cluster's nodes, before returning "success" result to OS.

HeartBeat software was used to start user applications and control the status of clustered machines. Additionally, the free MySQL 5 database storing tariff and subscriber data is started together with charging application on "Server 2" machine to deliver the highest performance

and the "Server 1" machine was used for data administration, e.g. as customer care front-end in on-line mode. The free Apache WWW server was used and the database front-end was implemented with the DaDaBIK interface creator. In fact, all the used software consists of components that are free/open-source and the only missing piece is the charging application that would be described in a separate chapter.

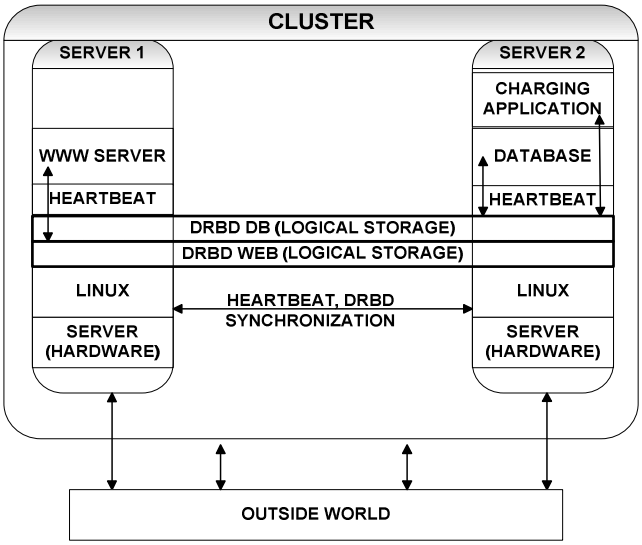

Figure 4. Cluster system for small CSPs

5.2. Charging Server for Nation-wide CSPs

The solution proposed for nation-wide CSPs differs from the one described above with the use for request processing a group of servers cluster rather than a single Server. The schema of the proposed approach is shown on Figure 5.

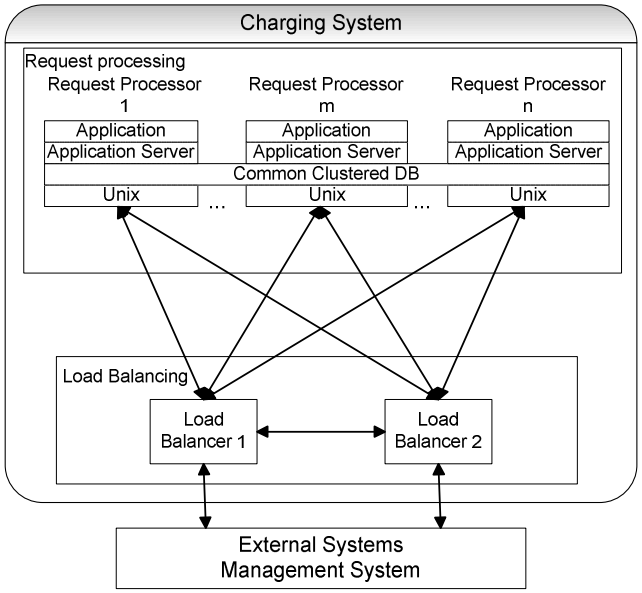

Figure 5. Nation-wide operator charging system schema

The presented charging system is composed of two main parts: a high-available load balancing module and a request processing module with multiple request processors. The first one is composed of at least two load balancing devices. For a final implementation, a proprietary solution could be chosen (appropriate devices are available from global vendors, e.g. from Cisco) or a linux-based "ldirector" working in clustered environment (e.g. HeatBeat) could be used. Except for native traffic routing, the load balancers' task is node failure detection and modifications of processors group.

The request processing engine consists of multiple servers based on commercial Unix operating system (e.g. Solaris, Red Hat). It is necessary for support purposes. On top of it, a clustered database is used, storing the profile data for all the CSP`s subscribers. The possible considered implementations are MySQL Cluster with NDB engine or Oracle RAS. These are designed exclusively for telecommunications industry needs. The use for common DB allows the request processors to be exactly the same in terms of installed software stack and tariff configuration. As a consequence each request could be sent to any available processor, reach the application layer and get processed. Due to the fact, that the proposed DBMS delivers full synchronization between all the processors together with transaction support, different requests related to particular subscriber profile could be processed in parallel on different processors.

Another possible storage environment could be one, in which each clustered processor holds its own DB. It seems that with such implementation, the DB maintenance would be easier due to lower amount of data stored in given cluster and the performance should be higher due to the fact that the application will always use the data gathered in its local resources. On the other hand, the load balancer role would have to be exchanged with some more complicated dispatching. Such dispatchers would have to implement deep packet inspection and route messages to destination clusters that make the system complicated, raising its OPEX, extending the time-to-market and lowering performance.

The application itself is meant to be running inside an application server like JBoss or GlassFish. Application development should focus on implementation of business and law requirements leaving all the technology problems (e.g.: context synchronization, database support, etc.) to be solved in some native server software. The mentioned J2EE application servers deliver the necessary standard framework for developing telco-enabled applications. For charging purposes discussed in this work, JAIN-SLEE implementation (e.g. OpenCloud`s Rhino) could be considered. This environment has some built-in features specific for charging purposes (e.g. Diameter protocol support) and some CSP extensions (e.g. Parlay/OSA gateways).

The system management (administration of tariff data and subscriber's profile audit) should be done using external systems. Some functionality is proposed to be implemented in off-line environment, e.g. profile viewing for subscribers. Additionally, separate modules are necessary for mass CSP's data administration and mediation between charging system and 3rd party software interworking with it.

6. CHARGING APPLICATION

The charging application implements all business and law requirements. The design of this application should be flexible enough to allow short time-to-market for CSP's current and future products (services). It is completely necessary for any charging application to support at least two charging modes: the so called "IEC" (Immediate Event Charging) and "SCUR" (Session Charging with Unit Reservation). Since the "IEC" is self-explanatory, the "SCUR" only would be further discussed. In that mode, after a request is received from the reporting device, some usage quota is granted and units that are to be paid for the service are reserved on subscriber's account. After the back-reporting is received, the decision has to be done, whether to charge for the service or to release the reservation. Of course, the reserved amount could be partially

charged, e.g. if the granted quota was 100 and only 60 was used with the second-based charging. Figure 6 shows steps that the charging application need to execute after a particular request is received.

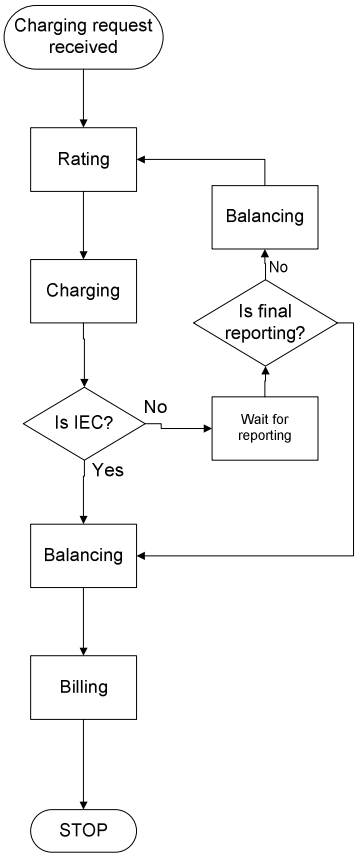

Figure 6. Charging application execution sequence

All steps presented on Figure 6 will be described in further chapters of this work. To perform any request handling some configuration is necessary, e.g. tariff definition. All those data are to be kept in the DB to be easily accessible from external systems, but reading that seldom changed data for each request seriously affects performance. That is why we propose the application design that stores the tariff data in the internal application cache, which is being refreshed at the end of each configured period of time. Due to the tariff structure design model – which will be defined later – for nation-wide CSPs we propose to use the Oracle TimesTen in-memory database. In order to keep the tariff data up-to-date, cache refresh should be executed in periodical manner. The period length should be configurable on the OCS and its estimation should be performed taking into consideration the CSP-specific application needs. For some CSPs, the period could be set to e.g. one day (because the tariff data is not changed very often) and for the others, the refresh has to be done more often (e.g. when some subscriber-specific data is stored in the cache). The data structures used there should be similar in terms of functionality to the usual database tables, so they need to implement easy way to enter, change and especially search for the data. It is necessary to implement two searching algorithms for SIP requests charging purposes: BestFit search and ExactMatch search. The first one is crucial when it comes to destination evaluation (significant tariff data reduction) and the second one is usually used for shortcodes (e.g. 112) handling. A prototype of a “Table” class is shown on Figure 7.

Table
Attributes
protected String name = null protected int columnNum = 0 protected int rowNum = 0 protected String columnNames[0..*] = null protected String columnTypes[0..*] = null protected String contents[0..*] = null
Operations
package Table(SQLData DBDesc, String Name) public String Search(String SearchString) public String[0..*] SearchMultipleRows (String SearchString) public String ListTable(Boolean PrintOnScreen)

Figure 7. Prototype of "Table" class, which is used to keep the tariff data

SubscriberData
Attributes
protected String internalID = null protected String offer= null protected Date StateEndDate = new Date(1970, 01, 01) protected boolean isLocked = true protected boolean negCredit = true protected int negCreditLimit = 0 protected String recommendedSubs = null protected int recommendedSubsNum = 0 protected String subscriptions = null protected String homeLocationIDs = null protected ArrayList accounts = new ArrayList() protected Enum SubscriberState = "Inactive"
Operations
package SubscriberData(SQLData DBData, Request request) private void ExecuteSLC() private void ShowSubscriber() public boolean CheckPermission(ServiceType ServiceType) public Boolean Charge(int AccountToCharge, int Amount, Boolean SaveToDB) public Enum getSubscriberState() public void setSubscriberState(Enum val)

Account
Attributes
public int balance public Date ExpiryDate public int number
Operations
packageAccount(int num, int bal, Date ExDate) public int GetBalance() public void AddToBalance(int Amount) public void RevomeFromBalance(int Amount) public boolean isExpired() public void ExtendExpiryDate(int Days, int Mode) public void ShowAccount()

Permissions
Attributes
public boolean isVOIPAllowed = false public boolean isIPTVAllowed = false public boolean isVODAllowed = false public boolean isMP3ServiceAllowed = false public boolean isBookServiceAllowed = false
Operations
public void ShowPermissions()

Figure 8. Proposed subscriber data design

It is obvious that for the application execution, subscriber data should be defined. There are different structures of those data, like flags (binary true/false), date fields, integer fields and some structures, like accounts defined by both: balance (as integer) and the expiry date. The example set of subscriber-specific data is presented on Figure 8.

6.1. Rating

The rating's role is to determine the amount of *units* that have to be charged for particular *amount* of service (e.g. a download of 1 mp3 file or 25 seconds of movie watching). The *units* usually refer to the lowest amount of given country's currency, but they could be as well

defined as "points", time units (e.g. seconds) and others. The *amount* of service is usually perceived as time spent using particular SIP-delivered service, but could be as well defined as amount of downloads or views depending on particular service. Anyway, the rating has to take into consideration different parameters received from reporting devices (e.g. end-user equipment) and evaluated locally (e.g. local time) to perform successful price determination. The following conditions need to be evaluated to perform rating:

- Location, from where the service is being accessed.
- Local subscriber time (for countries covered by multiple time-zones).
- The type of service, that is being accessed/used and the charging mode for that service (please note, that mode could be chosen by CSP's subscribers).
- Service-specific charging data, e.g. for VoIP service: A-Party URI, B-Party URI, codec used for voice transmission.

Due to the fact that multiple services need to be supported and the application should be flexible enough to charge ones that will be implemented in the future, the industry-standard design patterns [17] could be used, like the "Factory" pattern, that is shown on Figure 9 (for VOD Service).

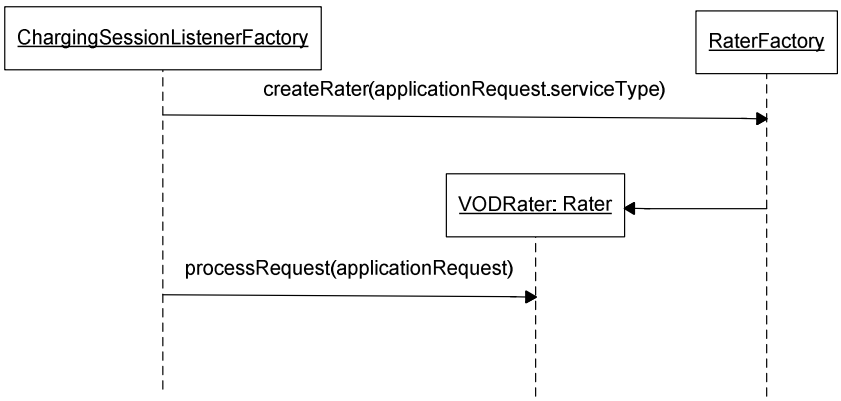

Figure 9. VODRater creation using "Factory" design pattern

After the request is delivered to the application, the object of "RaterFactory" class is requested to provide the appropriate object of a class derived from "Rater" abstract class, in this case – "VODRater" object. The choice is based on the value of the service type delivered in createRater method invocation. After that object is created, the request is passed to the rater which takes control over further processing.

The most important rating-related design decision is the approach for tariff data design. The appropriate tariff data design allows the CSP to easily create new instances of data, limits the amount of administration that is necessary to make the rating successful and clearly separates the different raters' functionality. If those conditions are fulfilled, the software is maintainable and delivers decent performance. Proposed is, that the most general data instance is the tariff identified by a name, e.g. "One". This schema is defined as a tree, in which the tariff itself is identified as a root and first-level branches are the services offered. The sub-tariffs (child tariffs) share the same schema, so a hierarchical database is created. The amount of sub-tariffs is

depending on what the tariff should offer: the basic one may contain only VoIP service, but a fully-featured tariff "VIP" could be offered to subscribers willing to benefit from all the CSP could deliver. If the subscriber tries to use a particular service, which is not covered by the tree of his tariff, then that service is not allowed for him and any usage attempt should be rejected. The leaves of the tree identify the final service-specific data, e.g. destinations for VoIP service together with assigned prices and the add-on subscriptions, the subscriber may buy. A subscription is defined here as an additional, business service, that allows the subscriber to obtain some special handling for his requests, e.g. Friends'nFamily tariff (is subscriber calls a number, that was registered as "Friend", the discount is applied). The example tariff tree is shown on Figure 10.

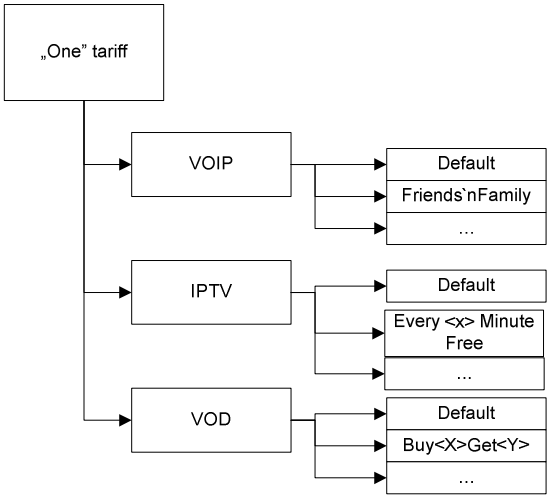

Figure 10. Example tariff tree design for services delivered with the use for SIP

6.2. Charging

The charging role is to determine the account to charge the amount defined by the rating and to make sure that all the defined balance checks are satisfied. These checks are known as thresholds, especially "SessionCreditThreshold" (SCT) and "SessionTimeThreshold" (STT) for VoIP service accomplished with the use for SIP-P2P. The first one is related to minimal amount of units the subscriber should have on his account to establish the call. It is used when the CSP offers some business service, that is free-of-charge, but the subscriber needs to have his account valid to make the call. Although the first threshold secures the CSP, using the second threshold benefits for both, i.e. the CSP and the subscriber. For instance, if the account doesn't have enough balance to set-up a session lasting longer, than it would not be established at all. In consequence, the subscriber's money would not be wasted on the service that he would not have a real chance to enjoy.

There is yet another task to be done by the charging in the case of SCUR mode: quota evaluation. As already described, the rating delivers only the price for basic amount of service and the charging has to decide, how many such amounts need to be granted to the subscribers. The estimation of minimum, maximum and optimal quota to be granted per service should be done by the CSP with the use for usage statistics, if a service is live or market demand estimation for new product. Quota estimation algorithms were discussed in [18]. Proposed charging flow for IEC is shown on Figure 11.

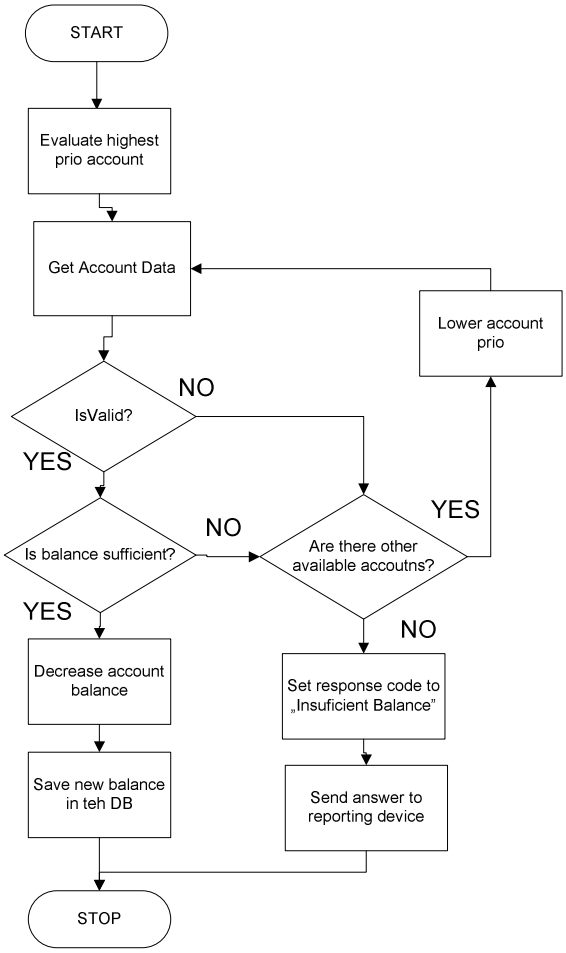

Figure 11. Charging flow for IEC

6.3. Billing

Billing is the last step of charging application execution. The role of billing is to gather the information characterizing the usage, for documentation and statistics purposes. The industry standard here is the CDR (Call Data Records) and IPDR (IP Data Records). Due to the fact that recently charging systems are being migrated to application servers, the authors propose to use the XML format for charging data records instead of CDRs and IPDRs. This decision was made, because these application servers are usually implemented using Java 2 EE technology. The XML format was used as a base since there are internet-enabled applications and some tools implemented in J2EE framework that are capable of reading, analyzing and presenting easily the data to the users. Additionally, the CSPs add content reselling to their existing business and cooperate with content providers. These companies have usually some enterprise applications that implement the IT standards like XML more than telecommunications-specific CDRs and IPDRs.

Due to the fact that different kinds of services implemented with the use for SIP-P2P gather different data to execute the rating and charging, different classes are needed to create proper billing. That is why again the "Factory" pattern is proposed to provide a flexible method of billing execution. The listing below presents the example xml file documenting the VOD service usage.

```
<xml>
<Request>
      <TimeStamp>Tue May 19 00:00:45 CEST 2009</TimeStamp>
      <RequestTime>1242684043</RequestTime>
      <RequestTimeZone>UTC-2</RequestTimeZone>
      <RequestedService>VOD</RequestedService>
      <AccessMethodType>STB</AccessMethodType>
      <UserIdentity>STB123456</UserIdentity>
      <RequestedContent>Leon</RequestedContent>
      <Quality>HD</Quality>
</Request>
<Subscriber>
      <InternalID>123456</InternalID>
      <Offer>Tariff1</Offer>
      <Accounts>{1|150|1262300399000}{4|0|1262300400000}</Accounts>
</Subscriber>
<RatingResults>
      <Prices>{200}</Prices>
      <Lengths>{0}</Lengths>
      <FinalTariff>ECONOMY;HD;HL;OffPeak</FinalTariff>
      <BillingID>13101003</BillingID>
      <ChargingMode>Event</ChargingMode>
</RatingResults>
<ChargingResults>
      <AccountID>1</AccountID>
      <ChargedAmount>200</ChargedAmount>
      <NewBalance>150</NewBalance>
</ChargingResults>
</xml>
```

7. DIAMETER PROTOCOL – SYSTEM INTERCONNECTIONS

As already described, the Diameter protocol [12] is proposed to be used for charging system interconnections. The IETF definition of Diameter gives the vendors a huge range of adaptation possibilities that is pretty useful. But on the other hand, this approach could introduce some problems for system integration in heterogeneous environments. This issue is addressed by 3GPP, that releases some standardized interfaces that are to be used in CSPs' networks. For all the services that could be implemented with the use for SIP-P2P, there is already 3GPP TS 32.260 [19] standard defined, which takes into consideration IMS services charging. Additionally, there was another standardization done specifically for the PoC service (Push-to-talk over cellular), in 3GPP TS 32.272 [20]. The authors propose to utilize that standard, when it comes to the most common VoIP service. The existence of mobility support (e.g. LocationInfo) in 3GPP makes it automatically possible to implement location dependent charging. Another advantage related to this 3GPP standardization is that the Diameter stacks are ready for use in products from multiple vendors. There are implementations existing in application servers, like IBM`s WebSphere and stand-alone ones like OpenBlox from Traffix Systems that has already been chosen for SUN's SailFin project.

The messaging in Diameter protocol is understood as exchanging the message codes and AVPs (Attribute-Value Pairs) between communicating devices. The example message flow is shown on Figure 12. For SIP-P2P implementation, the main difference to current solutions is that each reporting device connects to the charging system. For that reason, the CER (Capability Exchange Request) message is sent to find out, if the particular application (in that case Diameter-Credit-Control-Application) is supported on the charging system. After a positive answer, the CCR (Credit-Control-Request) messages are sent from reporting devices to query

for allowed usage granting and in response the CCA messages are sent (Credit-Control-Answer). At the end of conversation, the DPR (Disconnect-Peer-Request) is sent to release the dialog and DPA is sent as confirmation. That message flow is specific for SCUR charging mode (for ECUR, there is no CCRt/CCAt in the message flow).

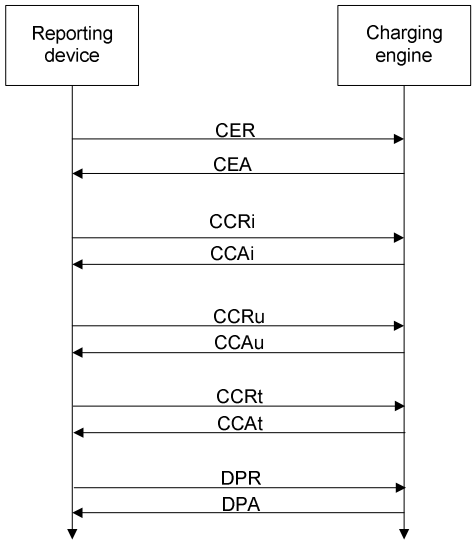

Figure 12. Example Diameter message flow

8. DIMENSIONING

In this chapter we propose a methodology for dimensioning of charging system. For our dimensioning methodology evaluation a simple cluster system was built. Results presented in this chapter refer to small CSP environments and do not give the exact numbers regarding the real application performance. The authors' intention is to point out issues that the system designer needs to take into consideration in order to provide a truly reliable system to the CSP network, rather than present some data specific to the test-cluster performance. The test bed includes two Linux-based servers installed in a VMWare environment. Dual-core CPU was used to perform the tests and each server obtained one core. A total amount of 2GB RAM was installed in the system and 512MB was assigned per each virtual machine. The charging system was simulated by a PHP script that was running inside the Apache server on top of the MySQL DB containing ~150k subscribers. The script execution involves multiple SQL commands executions; few date comparisons and rating execution. The output is a web page showing the experiment log. Intra-server connections were implemented using the VMWare internal network and a cluster was visible from externals networks on a highly-available IP address (virtual interface was bind to physical laptop's Ethernet interface). As the load generator, another laptop was used with the Linux operating system. AutoBench together with httperf were used to send the HTTP requests towards cluster. Measures were taken on that machine.

The aim of the tests was to show the maximum load that the cluster could process when one of the servers fails. During the tests, the following parameters were set for httperf tool: in one test 2000 connections are performed and only one http "get" request is sent per connection. The amount of parallel sessions was set to 25 in the first test and to 40 for the second test. The particular request was counted as processed successfully if two conditions were fulfilled: in response the HTTP code 200 (success) was received and the request processing time was lower than 200ms. The following graphs (Figure 14 and Figure 15) show the amount of successful

requests per second, when both servers are running constantly. The load was processed by cluster without any delays.

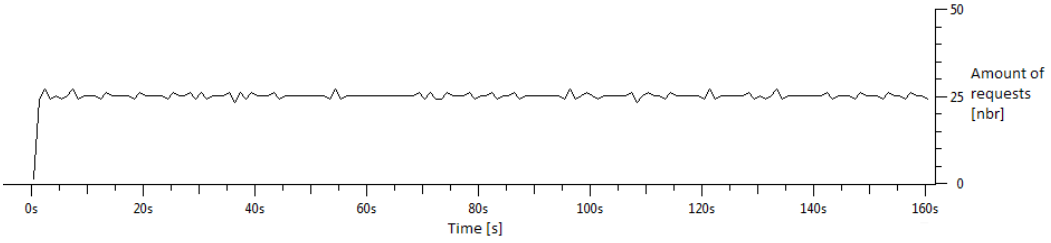

Figure 13. Amount of requests processed successfully in time - 25 requests per second

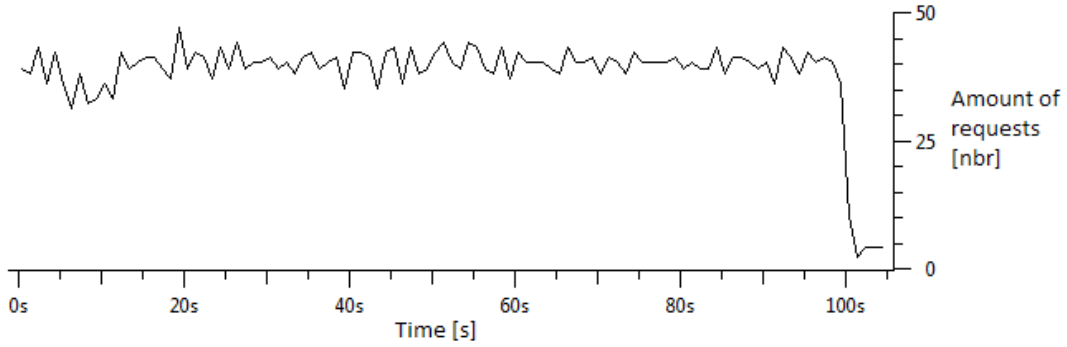

Figure 14. Amount of requests processed successfully in time - 40 requests per second

The following graphs (Figure 15 and Figure 16) show the amount of requests processed, when one of the cluster's nodes fail. Take into consideration that the HeartBeat platform was configured to start the particular service on the stand-by node ~30s after last heart beat signal is received (signals sent each 5s). As we can notice the cluster is potentially able to process successfully 45 requests per second, when both nodes are running, but when a failure happens, the cluster is not able to recover at all (as shown on Figure 16). It means that the CSP does not benefit from having that highly-available environment. The aim of building them is to make sure that if one server fails, the second one would be able to handle the load and this requirement is not fulfilled. On the other hand, when 25 requests are sent towards the cluster, its recovery mechanism works correctly and after some time needed for stabilization (~60s in this case), the load is correctly processed (as shown on Figure 15). To summarize, the designer's role is not only to perform tests showing, what is the potential of the cluster, when it is running fine. But the most important decision should be based on the fail-over mechanism testing. Almost for any device, which is used in the telecommunication industry, the total amount of subscribers that tend to use a particular device is higher than its capability, since all the subscribers do not use the services at the same time. Appropriate dimensioning could be done with the use of Erlang's formulas [21], but these calculations would give a CSP only an overview of the average load coming to the cluster. In order to make sure that the services would be handled correctly in overload situations, traffic limits should be established. This ensures that the amount of requests processed in parallel will never be higher than the limit evaluated in fail-over tests. If such overload situation happens, the authors propose to use the request gapping mechanism, which rejects automatically a given percent of all the requests coming to the system as long as the traffic limit is violated. Common call gapping algorithms were discussed in [22].

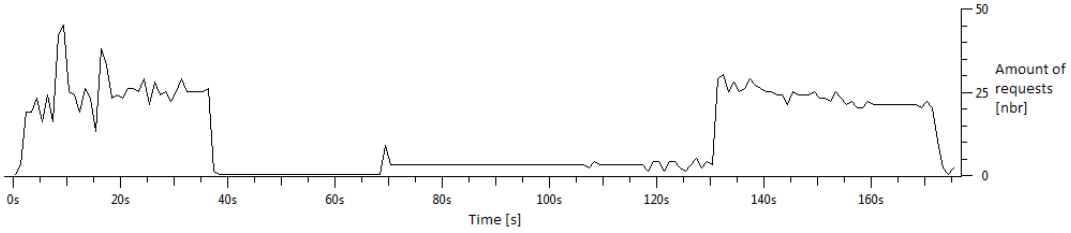

Figure 15. Amount of requests processed successfully in time - 25 requests per second

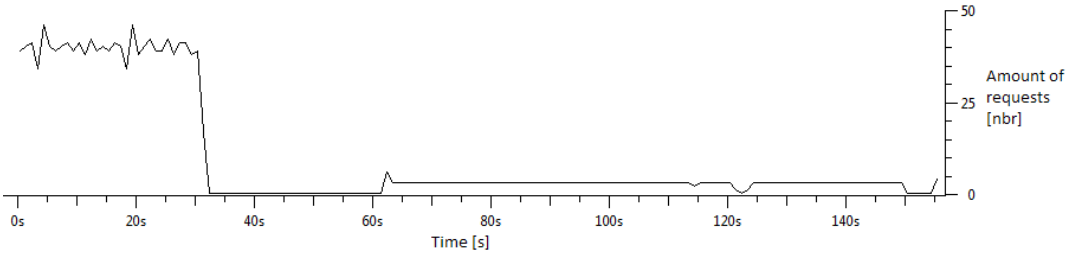

Figure 16. Amount of requests processed successfully in time - 40 requests per second

9. CONCLUSIONS

We have presented in this paper only the main principles related to charging systems design and development. Each CSP needs a system tailored to its specific business and local market requirements. That is why the proposed system would have to be further customized to satisfy all the CSPs' needs. Especially the application itself would have to be extended with some bonuses and promotion handling, that is specific to each CSP and to be honest, the set of requested features changes pretty often as the whole market does. In order to implement those modifications, a team of experts in software engineering and computer networks has to be established to further define and implement the application itself in a way that it is in line with local law and marketing principles. The engineer team needs to constantly verify its achievements to detect potential problems at each stage of development, starting at requirements definition and ending at dimensioning.

The SIP-P2P technology gives the CSPs lots of new opportunities including improvements in services reliability and reducing the OPEX, but requires them to move some parts of the charging system out of traditionally closed CSP domain. The development of SIP-P2P approach is possible with the application of presented above technologies (encryption, the use for secure Diameter protocol, etc.). However, the major problem is to convince the CSP that this is the correct way to proceed.

It seems that the key for a successful charging system design – independently of the amount of CSP's subscribers – is the use for industry standards. If standard solutions are used, it will be pretty easy to exchange or extend the system's modules when necessary, keeping all other elements compatible. This work has shown that even for small operator it is possible to build pretty powerful and flexible system using open-source and freeware. Another advantage of using this approach for building complicated systems is the fact that each element is developed by a group of professionals who focus on delivering a product capable of implementing functions in only one module of the charging system, e.g. WWW server for data presentation and configuration interface. The only missing part for the small system that has to be developed is the charging application. In this case, the implementation of industry standards like design patterns in OOP are crucial for time-to-market reduction and effective software maintenance.

Another manifestation of using standard solution is the Diameter protocol support. As the 3GPP standardization efforts show, the telco domain is really enthusiastic about using it. It allows to swap the existing expensive SS7 solutions with easier (wide spectrum of free protocol analyzers is available to be used, e.g. WireShark), better known (more and more people are familiar with IP) and cheaper IP solution.

Finally, dimensioning aspects have been discussed and as a result some methodology has been proposed. For clustered systems – to make it work without flaws in any situation – proper performance testing is absolutely necessary. More than that, it is essential for undisturbed system operation to implement some load control mechanisms in line with request gapping mechanisms. Any exception in this field will result in losing all the advantages of cluster platforms.

ACKNOWLEDGEMENTS

This work is supported by The Polish Ministry of Science and Higher Education under the grant which is being realized in years 2008-2011.

REFERENCES

[1] Balakrishnan, H., Kaashoek, M. F., Karger, D., Morris, R., Stoica, I.: Looking up data in P2P networks, MIT laboratory for Computer Science, 2003, http://www.cs.berkeley.edu/~istoica/papers/2003/cacm03.pdf.

[2] Singh, K., Schulzrinne, H.: Peer-to-Peer Internet Telephony using SIP, In the Proceedings of the International Workshop on Network and Operating System Support for Digital Audio and Video, 63 - 68, 2005.

[3] Shooman, M.: Reliability of computer systems and networks, New York, John Wiley & Sons Inc., 2002.

[4] ITU-T G.1010: End-user multimedia QoS Categories, ITU, Geneva 2002, http://www.itu.int/rec/T-REC-G.1010-200111-I/en.

[5] Vergados, D., Protopsaltis, N., Anagnostopoulos, C., Anagnostopoulos, J., Theologou, M., Protonotarios, E.: A review of Call Admission Control Schemes in Wireless ATM networks, Lecture Notes in Computer Science Vol. 2093, Springer Verlag, 459-467, 2001.

[6] ITU-T E.164: The international public telecommunication numbering plan, ITU, Geneva 2005, http://www.itu.int/rec/T-REC-E.164-200502-I/en.

[7] Gerke, J., Hausheer, D., Stiller, B.: A Generic and Modular Accounting and Charging System for Peer-to-Peer Applications, In the Proceedings of the KiVS Kurzbeiträge und Workshop, 225-228, 2005.

[8] Vishnumurthy, V., Chandrakumar, S., Sirer, E.: KARMA : A Secure Economic Framework for Peer-to-Peer Resource Sharing, Cornell University, Ithaca, 2003, http://www.cs.cornell.edu/~vivi/karma.pdf.

[9] Hausheer, D., Stiller, B.: PeerMint: Decentralized and Secure Accounting for Peer-to-Peer Applications, Lecture Notes in Computer Science Vol. 3462, Springer Verlag, 40-52, 2005.

[10] Buford, J., Yu, H., Lua, E.: P2P Networking and Applications, Morgan Kaufmann 2009.

[11] Shen, X., Yu, H., Buford J., Akon, M., (eds.): Handbook of Peer-to-Peer Networking, Springer 2009.

[12] Calhoun, P., Loughney, J., Guttman, E., Zorn, G. Arkko, J.: Diameter Base Protocol RFC3588, IETF, 2003, http://www.ietf.org/rfc/rfc3588.txt.

[13] Romascanu, D., Tschofenig, H.: Updated IANA Considerations for Diameter Command Code Allocations, RFC5719, IETF, 2010, http://www.ietf.org/rfc/rfc5719.txt.

[14] Korhonen, J., Jones, M., Morand, L., Tsou, T.: Clarifications on the Routing of Diameter Requests Based on the Username and the Realm, RFC5729, IETF, 2010, http://www.ietf.org/rfc/rfc5729.txt.

[15] Mahmoud, Q.: Cognitive Networks: Towards Self-Aware Networks, Hoboken, Wiley & Sons, 2007.

[16] Bookman, C.: Linux Clustering: Building and Maintaining Linux Clusters, Indianapolis, New Riders Publishing 2002.

[17] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Abstraction and Reuse of Object-Oriented design, In the Proceedings of the 7th European Conference on Object-Oriented Programming ECOOP`93, Lecture Notes in Computer Science Vol. 707, Springer Verlag, 406-431, 1993.

[18] Lin, Y.-B., Sou, S.-I.: Charging for Mobile All-IP Telecommunications, Indianapolis, Wiley & Sons, 2008.

[19] 3GPP TS 32.260, Valbonne, 2009, http://www.3gpp.org/ftp/Specs/html-info/32260.htm.

[20] 3GPP TS 32.272, Valbonne, 2009, http://www.3gpp.org/ftp/Specs/html-info/32272.htm.

[21] Anantharam, V., Gopinath, B., Hajela, D.: A generalization of the Erlang formula of traffic engineering, Ithaca/Morristown, Springer, 277-288, 1988.

[22] Berger, A.: Comparison of call gapping and percent blocking for overload control in distributed switching systems and telecommunications networks, Holmdel, IEEE Transactions on Communications, Vol. 39, No. 4, 574-580, 1991.

Authors

Damian Nowak was born in 1985. In 2004, he started his M.Sc. studies on the Faculty of Electronics at Wrocław Univeristy of Technology. During the studies, he was given an annual Dean's Award and twice – the Dean's Congratulations letter for excellent development in science. In 2009, he got his degree and was awarded "The Best Faculty of Electronics Graduate 2009". His main objects of interest are computer networks, on-line charging system design principles, Diameter protocol and software development methodologies. Currently, he is employed at Nokia-Siemens Networks.

Krzysztof Walkowiak was born in 1973. He received the Ph.D. degree and the D.Sc. (habilitaion) degree in computer science from the Wroclaw University of Technology, Poland, in 2000 and 2008, respectively. Currently, he is an Associate Professor at the Chair of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Technology. His research interest is mainly focused on optimization of network distributed systems like P2P systems, multicasting systems, Grid systems; network survivability; optimization of connection-oriented networks (MPLS, DWDM); application of soft-optimization techniques for design of computer networks. Prof. Walkowiak was involved in many research projects related to optimization of computer networks. Moreover, he was consulting projects for the largest Polish companies including TP SA, PZU, PKO BP, Energia Pro. Prof. Walkowiak published more than 90 scientific papers. He serves as a reviewer for many international journals including: Computer Communication, Computational Optimization and Applications, International Journal of Applied Mathematics and Computer Science, Expert Systems, Pattern Analysis and Applications, International Journal of Computer Mathematics. He is/was actively involved in many international conferences. Prof. Walkowiak is a member of IEEE and ComSoc.