IP Telephony Accounting and WAN Deployment Experience

Sven Ubik, CESNET
Prague, Czech Republic
ubik@cesnet.cz

Abstract—**Convergence of data and telephony networks is one of the current trends in telecommunications. It is supposed that all telephony traffic might be carried over a common data network infrastructure in the future.**

In accordance with this development, wide area IP telephony network has been deployed in the TEN-155 CZ network (located in Czech Republic), connected to the European network for research and education TEN-155.

In this article we describe our practical experience with deployment of IP telephony services over a wide area network. We particularly pay attention to the description of IPTA, the IP telephony accounting application, which has been developed as part of our project. IPTA differs from other IP telephony accounting applications by its modular structure, ability of using multiple accounting schemes in paralel and certain other features which we describe in this article.

Keywords—**IP Telephony Accounting, RADIUS, Voice Gateway.**

I. Introduction

IP telephony has been adopted as a way to carry telephone traffic over a common data network infrastructure. However, in order to replace classical telephony network with IP telephony, we also need means for call accounting. In this article we describe our experience with the deployment of IP telephony services over a wide area network and with the development of IP telephony accounting application.

II. IP Telephony in TEN-155 CZ

TEN-155 CZ is a national network for research and education in Czech Republic. It is connected to TEN-155, the European network for research and education. In mid-1999 we established an IP telephony group working in the TEN-155 CZ network. The aim of the group has been set to perform testing, measurement, development and practical deployment of technology for the transmission of voice over a data network. The group is financially and organizationally supported by CESNET, a not-for-profit organization which operates the TEN-155 CZ network.

The goal of the first phase of our project was to interconnect telephone exchanges of several universities across the country using IP telephony and to connect this IP telephony network to the public telephone network. We also wanted to acquire experience with the deployment of IP telephony over a wide area network and to investigate possibilities of integration of IP telephony with other services in the next phase.

III. Selecting technology

One of the requirements was to incorporate existing hardware - telephone exchanges in universities and routers in the TEN-155 CZ network. The telephone exchanges (PBXs) are of different brands and types: Ericsson MD110, Siemens Hicom 300, Alcatel and Matra, and they are equipped with different data interfaces. The network is based mainly on Cisco routers, running ATM and providing both ATM and IP services. We performed a series of experiments with various telephony equipment, particularly testing available PBX interfaces for their suitability for the connection to a data network. Upon completing these tests, we decided to use the VoIP (voice over IP) technology (as oposed to voice over ATM) and digital ISDN PRI/BRI and analog E&M and FXS interfaces.

Next, the choice of signalling had to be made. There are currently two major protocols used for connection management in IP telephony: H.323 [1] and SIP (Session Initiation Protocol) [2] together with SDP (Session Description Protocol) [3]. H.323 is currently prevailing in production implementations of VoIP services. A major advantage of SIP over H.323 is that it is simpler, messages are textual, rather than binary, similar in nature to HTTP. This makes implementation of VoIP software easier and the open architecture of SIP allows for easier integration with other services.

We performed a series of experiments with both H.323 and SIP on Cisco routers, using Microsoft Netmeeting and Cisco IP Phone 30 as H.323 clients and Columbia University sipc and Cisco IP Phone 7960 as SIP clients. We also used sipd (SIP proxy, redirect and registrar server) and SIP/H.323 gateway developed in Columbia University. While all software from Columbia University was working without problems, we ran into difficulties with Cisco IP Phone 7960 (with SIP firmware, release 1.0) and with configuration of SIP on Cisco routers. Using packet analyzer and protocol monitor in sipc we found that Cisco IP Phone 7960 was sending incorrect `From` and `To` headers in INVITE and REGISTER SIP messages. The problem was reported by Cisco as a known bug. Therefore, we decided to use H.323 initially, with hardware architecture allowing to switch to SIP later (mainly choosing routers reported to support both H.323 and SIP).

IV. Pilot project

At the end of the year 1999 the local telephone exchanges of CVUT in Prague and VSB-TU in Ostrava (two universities approximately 400 kilometers appart) were connected by the VoIP technology in a pilot project [4]. We connected universities in two other cities later. For compatibility with existing hardware deployed in our network, we inititally used Cisco AS5300 routers with the E1 ISDN/PRI interfaces to implement VoIP gateways. The configuration is shown in Fig. 1.

We did not have routers to be used as voice gateways in the other two other universitites (connected later, shown below in Fig. 1). Therefore, we took advantage of ATM connectivity between all connected universities. We configured voice gate-

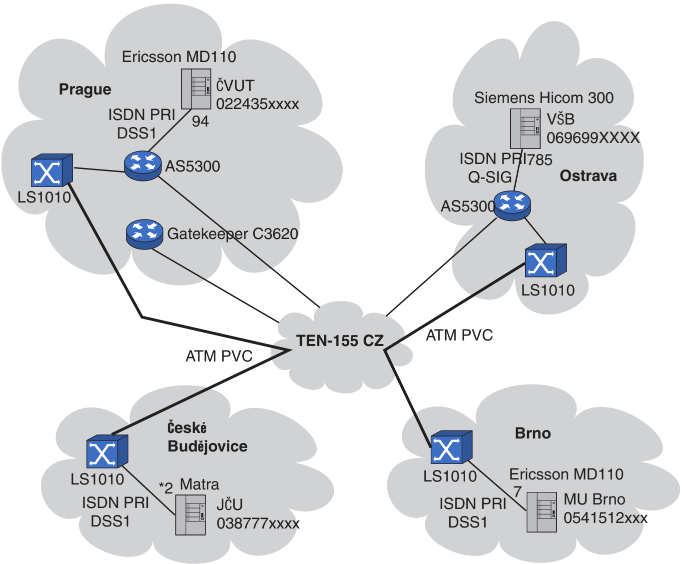

Fig. 1. Configuration of the pilot project

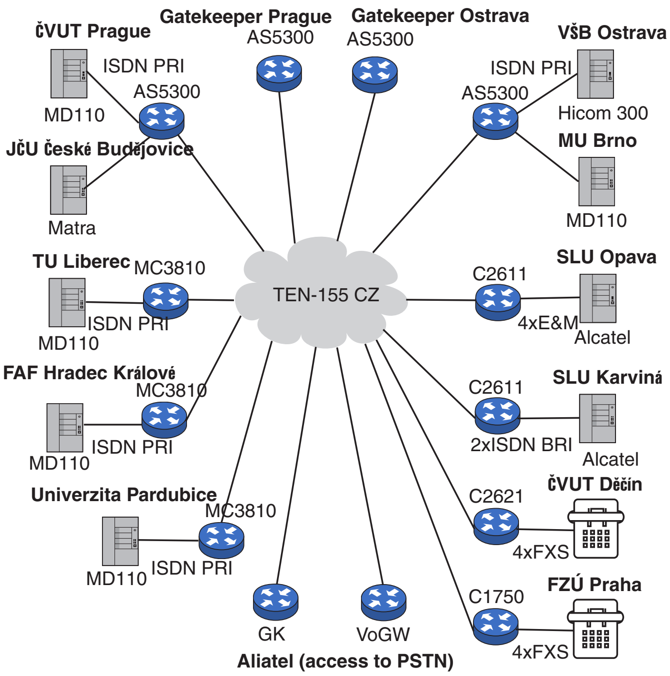

Fig. 2. Current IP telephony infrastructure in TEN-155 CZ

ways at the first two universities as having the PBXs in the latter two universities connected directly to their local ports, tunneling connections from these ports to remote PBXs through ATM PVCs. There is E1 connection between the voice gateway and ATM switch on one side and between the ATM switch and PBX on the other side.

To call from one university to another, users dial a specified prefix to escape into the IP telephony network, followed by the called number in E.164 format. We plan to enable transparent routing of calls through the IP telephony network using LCR (least cost routing) service, implemented in some PBXs. When calling from an extension in one PBX to an extension in another PBX, there is no telephone fee paid by either the caller or callee. As a part of the pilot project, an experimantal VoIP connection to CERN, Switzerland was also implemented and is currently used in both directions.

V. Current infrastructure

We are currently connecting several other universities to our IP telephony network. The infrastructure to be completed by May 2001 is depicted in Fig. 2. Although we were satisfied with Cisco AS5300 routers, we needed a less expensive alternative to build more voice gateways. Based on the requirements on voice interfaces in individual locations, we decided to use Cisco MC3810 or Cisco 2620 routers. As we intend to gradually switch from ATM to PoS on the network backbone, we plan to install separate voice gateways to two localities now connected through ATM PVC tunnels.

Currently we use G.711 codec on all VoIP connections. We also tested G.729 codec to see if people find the quality of sound acceptable. The quality seemed to be reasonable so we consider using G.729 when bandwidth becomes an issue for some localities.

We use two gatekeepers to provide fault tolerance in case one of the gatekeepers fails. This is made possible by setting two gatekeepers with different priorities in configuration of voice gateways. With some newer versions of the Cisco IOS it is possible to implement both a voice gateway and a gatekeeper on the same router. However, it requires so called enterprise feature set of the Cisco which is expensive and thus the collocation of a voice gateway and a gatekeeper on the same box could be more expensive than two separate boxes.

The IP telephony network is connected to the public telephone network and mobile phone network via the telephone exchange of CVUT in Prague. The service is provided by a telecommunication company. We have to pay to this company for the calls made from the telephone exchange of CVUT in Prague. It means that if someone from another city calls a number located in Prague, the call will be carried by the IP telephony network to Prague and it will be charged only as a local call in Prague.

VI. Accounting - a missing part

The telecommunication company currently charges us using a classical scheme, based on the call destination, duration, time of day and day of week. Even in case of another scheme, we would always need to keep track of calls initiated by all users connected to our IP telephony network. Using this information, we can determine the amounts which we will require to be paid by connected universities to settle the bill from the telecommunication company. It would be also useful to keep track about IP-only telephony calls (not continuing to the public telephony network) to check the utilization of VoIP interfaces and find potential problems.

As we did not find any existing solution satisfying all our requirements, we decided to develop a new IP telephony accounting application. We identified the following features which must be supported by the IP telephony accounting application:

- User-defined aggregations — flexible call listings from high-level summaries of aggregated VoIP traffic between connected

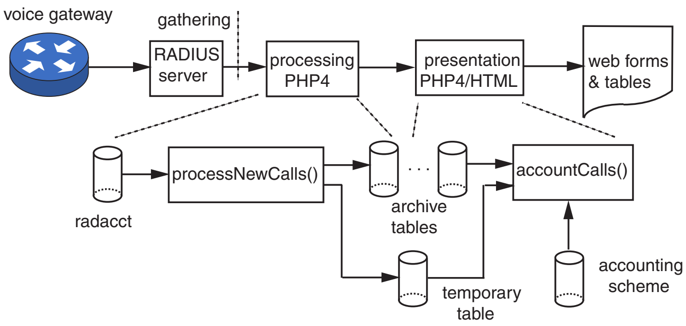

Fig. 3. Structure of the accounting application

organizations to detailed listings of individual calls

- Flexible accounting scheme — initially based on the location of the caller and callee, time of day and day of week, amount of transfered data and other parameters, with the posibility of easy modification
- Possibility of simultaneous accounting using several different accounting schemes - to evaluate alternative accounting approaches and gateway locations
- Reliability and scalability — resillience against changes of configuration of VoIP gateways, ability to store and search in reasonable speed records for calls made over several years

VII. Architecture

The structure of the accounting application is shown in Fig. 3. It consist of three parts - gathering of accounting information from voice gateways, processing of accounting information and presentation of the processed information to the user in a web-based interface. The three parts have defined interfaces between them and it is possible to replace one of the parts without the need to change the other parts. For example, it is possible to gather accounting information using different protocols from different voice gateways.

A. Gathering of accounting information

The routers which we used as voice gateways provide information about the routed calls via RADIUS accounting messages or UNIX syslog protocol. We decided to use RADIUS [5] because it provides accounting information in a more structural format than the syslog protocol, which is more suitable for further processing. RADIUS messages are sent from voice gateways, acting here as Network Access Servers (NAS), to the main and backup RADIUS servers. RADIUS is one of the standardized protocols for sending authorization and accounting information. However, the set of attributes included in the original specification do not cover all information needed in IP telephony accounting. There are two ways how to carry all necessary information.

First, the `AcctSessionId` attribute can be overloaded, carrying several pieces of accounting information in a series of strings separated by slashes [7]. An example `AcctSessionId` attribute value can look like this:

```
427/10:08:23.785 MET Thu Nov 23 2000/
Voice-OV.ten.cz/9662886D A66029F 0
42FD97B4/answer/Telephony/10:08:54.544
MET Thu Nov 23 2000/10:08:57.194 MET
Thu Nov 23 2000/10/195.113.134.240
```

Individual parts delimited by slashes mean: session identifier, call leg setup time, gateway id, connection id, call origin, call type, connect time, disconnect time, disconnect cause and remote IP address. We found that the exact format of the `AcctSessionId` attribute differs slightly in different Cisco IOS versions. However, the key parts are always present and it is possible to make parsing of this attribute resilient to small changes.

Another possibility is to use vendor-specific attributes [8]. Each piece of accounting information is carried in a separate attribute encapsulated in a vendor-specific (type 26) RADIUS attribute. An example set of vendor-specific attributes can look like this:

```
h323-gw-id=AS2-OV.ten.cz
h323-conf-id=
  4776D8CD B9A60177 0 37DBD1 74
h323-incoming-conf-id=
  4776D8CD B9A60177 0 37DBD174
h323-call-origin=answer
h323-call-type=VoIP
h323-setup-time=
  16:17:13.944 MET Mon Jan 22 2001
h323-connect-time=
  16:17:16.256 MET Mon Jan 22 2001
h323-disconnect-time=
  16:17:16.256 MET Mon Jan 22 2001
h323-disconnect-cause=10
h323-voice-quality=0
h323-remote-address=195.113.134.240
```

Some version of Cisco IOS can use only one of these two possibilities of carrying the accounting information, others can use any of them based on configuration. Therefore, the accounting application must accept both formats.

We decided to use ICRADIUS [6] server, because it has built-in support for storing RADIUS messages to a MySQL database. However, we had to modify the server in several ways. First, we added support for vendor-specific attributes, which was not included in the original server. Second, we had to change certain declarations so that the server now accepts long overloaded `AcctSessionId` attributes. The start and stop records (see below) now have to be matched based on the first part of the `AcctSessionId` attribute (the original session ID) rather than on the whole attribute value. Finally, we resolved some bugs in parsing of the configuration file.

The RADIUS server checks for duplicate messages, which can be received when the acknowledgement from the server sent back to the voice gateway gets lost, and it matches start and stop records sent by routers at the beginning and at the end of each call. However, it is sufficient to get only the stop records to obtain all needed accounting information. Therefore, we suggest that the voice gateways be configured to send the stop records only to preserve some bandwidth and processing on the server. The gatekeeper can also be configured to send RADIUS messages. However, although one of the tasks originally conceived to be performed by a gatekeeper was accounting, the RADIUS messages sent by the gatekeeper do not contain all the necessary

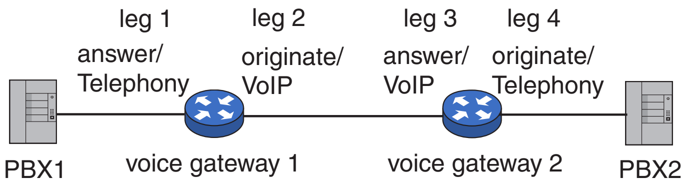

Fig. 4. Legs of a connection through two voice gateways

information about the calls. Information sent by voice gateways is more complete and sufficient, therefore we suggest that gatekeepers be configured not to send RADIUS messages.

Note the string `answer/Telephony` in the `AcctSessionId` value or `h323-call-origin` and `h323-call-type` vendor-specific attributes. These strings identify which call leg this message belongs to. Each call has either two legs, when passing through one voice gateway or four legs, when passing through two voice gateways (which is a typical case of calls from one PBX to another), see Fig. 4. Thus the RADIUS server normally receives either two or four messages for each call.

The parsed messages are stored in `radacct` table in the database, which serves as an interface between gathering and processing parts of the application.

B. Processing of accounting information

All processing and presentation of accounting information is done by PHP4 scripts. We normally use messages provided by the first voice gateway on the call route, namely the `answer/Telephony` leg for accounting of calls originated from a PBX and the `answer/VoIP` leg for accounting of calls originated by direct VoIP users (e.g., using Microsoft Netmeeting). The other two legs (`originate/Telephony` and `originate/VoIP`) are often missing some important information. An exception are calls from CERN, Switzerland where we could not configure the voice gateway (because of administrative reasons) to send RADIUS messages. Therefore, we use the `answer/VoIP` leg records provided by the second voice gateway on the call route to account these calls. However, sometimes the normally used message for a certain call leg is not available (the server did not receive the message) or it is broken in some way. In such cases we try to account the call using another leg by storing a found broken leg in a cache, looking for a correct leg of the same call (which has the same connection ID). This feature significantly improves reliablity of the accounting application. Detailed information about calls that had broken legs and about calls that could not be accounted even with assistance of the cache is logged via syslog enabling analysis of network problems.

Once a day `dailyTask()` script is automatically executed to process records in `radacct` table stored during the last 24 hours. Unfortunately, voice gateways sent messages for all attempts of calls, that is not only for calls that were made through, but also for cases when the called part did not respond. Therefore, the messages corresponding to these not-answered calls are identified and deleted from the `radacct` table by `deleteNotAnsweredCalls()` function.

```
tt CREATE TABLE tCallsDec2000 (
id INT NOT NULL AUTO_INCREMENT,
caller VARCHAR(32) NOT NULL,
callee VARCHAR(32) NOT NULL,
connectTimestamp INT NOT NULL,
duration int NOT NULL,
inputOctets int(12),
outputOctets int(12),
gwIp varchar(15),
remoteIp varchar(15),
callerMatched TINYINT,
calleeMatched TINYINT,
PRIMARY KEY (id),
UNIQUE (caller, callee, connectTimestamp)
);
```

Fig. 5. Structure of the table to store processed calls

Then, `processNewCalls()` function is called. As a first step, it identifies the caller and callee based on the IP address of the voice gateway the message came from, the identification strings sent by the voice gateway and the rewriting tables in the database. This step is necessary because in some cases the identification of the caller or callee sent by the voice gateway depends on the configuration and capabilities of the PBX and need to be rewritten to some normalized form for further processing. The normalized form we use is `<area code><number>` for users within the country and `<country code><area code><number>` for international users. Some PBXs, for example, Ericsson MD100 do not send the extension number (within an organization) of the calling user to the voice gateway by default. It has to be enabled by issuing a command separately for each extension. It can be automated by a script.

As the next step, the time of the beginning and end of the call is parsed and the duration of the call is computed. Finally, the call is inserted into one of the archive tables used to store processed calls. There is one such table for each month. This storage configuration scales well in terms of the number of records in a table and the number of tables in a database and files in the operating system. It also allows for easy backup of old tables by copying the corresponding files at the filesystem level. Definition of one of these tables is shown as an example in Fig. 5.

The cost of individual calls is not stored in the database. Only accounting schemes are stored in the database. The cost of calls is computed dynamically at the presentation time (see below). The cost can be based, for example, on the location of the callee, the duration of the call, time of day and day of week.

C. Presentation of accounting information

Accounting information is presented to the user in a web interface created dynamically also by PHP4 scripts. The user can browse through summaries at various levels of aggregation on either the caller or callee side, directly specify the caller and callee and the beginning and end of the accounting period. For example, it is possible to select a summary of all calls from CVUT (university in Prague) to numbers beginning 060, which

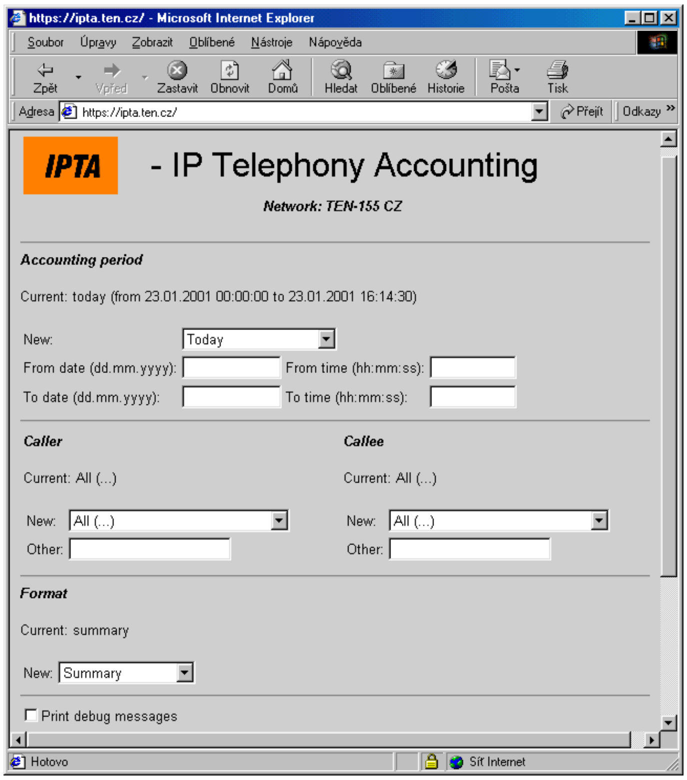

Fig. 6. Specifying accounting parameters

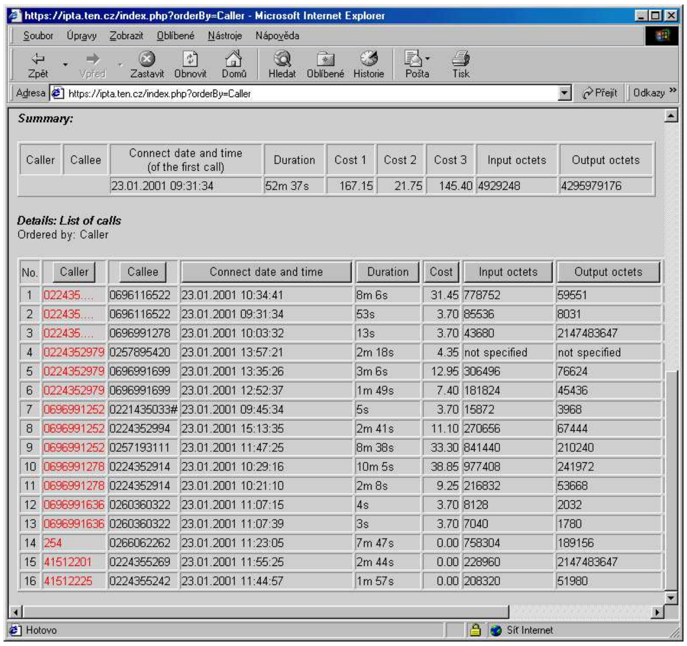

Fig. 7. Example of accounting information listing

are possible prefixes of local mobile phone operators. The form for the specification of accounting parameters is shown in Fig. 6. An example of a detailed listing of calls is shown in Fig. 7.

The user can specify the accounting scheme to be used. It is possible to account calls by several accounting schemes in parallel and the difference in cost can be presented. By combining aggregate listings and accounting schemes we can find patterns suggesting the most economical location of the gateway to the public telephone network.

The presentation part selects already processed calls from tables for individual months. Depending on the required range, calls from several tables may need to be aggregated. If the required range includes the current day, the `processNewCalls()` function is also called to obtain information about the recent calls, which have not yet been moved to the archive tables.

VIII. Conclusion and Future Directions

We successfully set up a wide area IP telephony network using Cisco routers and PBXs of different brands and types. The network is connected to PSTN and to the mobile phone network. Our experience is that H.323 is currently more reliable when using IP network based primarily on Cisco routers. However, development of SIP in Cisco IOS is under way and we plan to use SIP later to benefit from its advantages. We are currently continuing in tests of SIP-based communication using Cisco 3600 routers.

We developed an IP telephony accounting application which provides detailed information about individual calls as well as high-level summaries. The flexible structure of the accounting application allows using different methods of gathering accounting information from different voice gateways and accounting calls using multiple accounting schemes in parallel.

The application successfully gets over changes in RADIUS messages caused by changes in router configuration and it copes with missing or damaged RADIUS messages using a call leg cache. Another feature which should be available soon at least for some users (depending on the type of interface connecting the PBX and the used signaling) is the indication of the caller's name (in addition to the number) on the display of the callee's phone. We successfully tested this feature.

We plan to make experiments testing the behaviour of VoIP services running over the network with controlled QoS parameters in the cooperation with the QoS in IP working group that is implementing diffserv-based QoS in the TEN-155 CZ network.

References

[1] International Telecommunication Union, "Packet Based Multimedia Communication Systems", Recommendation H.323, Telecommunication Standardization Sector of ITU, Geneva, Switzerland, February 1998.
[2] Handley, M., Schulzrinne, H., Schooler, E., Rosenberg, J. "SIP: Session Initiation Protocol", *Request for Comments 2543*, Internet Engineering Task Force, March 1999.
[3] Handley, M., Jacobson, V. "SDP: Session Description Protocol", *Request for Comments 2327*, Internet Engineering Task Force, April 1998.
[4] Voznak, M., Bohac, M. "Experience with Operation of IP Telephony and its Integration into Current Telecommunication Infrastructure", COFAX conference, Bratislava, Slovakia, May 2000.
[5] Ligney, C. "RADIUS Accounting", *Request for Comments 2139*, Internet Engineering Task Force, June 2000.
[6] *ICRADIUS server*, http://icradius.hislora.com.au.
[7] Huff, T. "Configuration Guide for AAA Billing Features in Cisco Voice-Enabled Routers and Access Servers", Cisco Systems, 1999.
[8] "RADIUS Vendor-Specific Attributes Voice Implementation guide", Cisco Systems, 1999.