VoIP Based System for the Message Distribution

MIROSLAV VOZNAK, FILIP REZAC, JAROSLAV ZDRALEK
Department of Telecommunications
VSB–Technical University of Ostrava
17. listopadu 15/2172, 70833 Ostrava-Poruba
CZECH REPUBLIC
miroslav.voznak@vsb.cz, fillip.rezac@vsb.cz, jaroslav.zdralek@vsb.cz

Abstract: - Voice over IP is surely technology with enormous potential and we can find many applications utilizing VoIP as a communication platform. The paper deals with developed system based on VoIP enabling a user to distribute a voice message on mobile phones. It can be used not only for the purpose of warning in case of disasters but generally to notify people and announce an important message. The benefit of using voice message to transfer information, as opposed to other forms of alerts distribution, is that the end user is forced to pick up the call and to listen to the message – i.e. information cannot be missed or disregarded. Optimalization and estimated performance are related to the design of developed system therefore this issue is described in the paper too.

Key-Words: - Voice message, VoIP, disaster warning, Asterisk, Sipp

1 Introduction

Systems aimed at warning a large population against danger or serving as information sources are rather numerous these days and are used in many areas of human activity. Most of these systems, however, are based on sending text messages or distributing messages using a central device (siren horn, radio broadcasting). Our system is based on using SIP call generator to generate and distribute voice messages directly to the end device (IP phone, cell phone, fixed line, etc.) [1]. The benefit of such communication compared to the others is the fact that it uses a phone call and therefore it is possible to get feedback who accepted the message and to improve efficiency of alert system. The whole system will be based in the data centre of a telecommunications operator and will be accessible to the crisis centre's staff. A staff member logs into the system created by us, loads the pre-recorded alert and other parameters and sends the request. The output of the application are SIP messages which are sent into communication server, it can be based on open-source softswitch, such as Asterisk or OpenSIPS [2], [3]. The softswitch processes the messages and starts initiating calls to the end users according to the pre-defined parameters. The end user obtains a better understanding and sufficient information to solve the situation. If the end user does not answer the call (missed call, phone switched off), the system arranges to re-send the message and re-initiate the call with the end user concerned. Primarily, the system is designed to alert population to dangerous situation. Naturally, it can be used in other areas of activities.

2 Used Technology and Performance

The system has been designed as LAMP server [4], [5], [6]. The crisis centre' staff can operate it through a web interface. We suppose a co-operation with mobile operators, they are able to deliver the list of numbers located in target area [7]. The list which the staff members enter is saved in the XML format and contains two columns - telephone number of the end user and his/her name [8]. The number of rows in the list equals the number of end users. The messages are entered in the system in the .wav or .pcap format [9]. This voice message is then sent out to all end users and played once the phone is answered. The staff members may set the duration of the ringing. The call is not regarded as executed unless the end user answers the call. The remaining parameters entered into the system will be described further along with other algorithms which used these parameters. The system uses the open-source generator Sipp with pre-set call schemes in the XML format [10]. We met with Sipp in a research project in CESNET association and the acquired experience proved to be invaluable for this research [11]. To be able to process parameters entered into Sipp and XML dynamically, we applied two methods.

Using the first one, a correct parameter is assigned to the values entered into the forms by the crisis centre staff, and it is then sent to Sipp. In order to dynamically switch the telephone numbers based on the loaded list of end users, values in the XML scheme for Sipp need to change dynamically too.

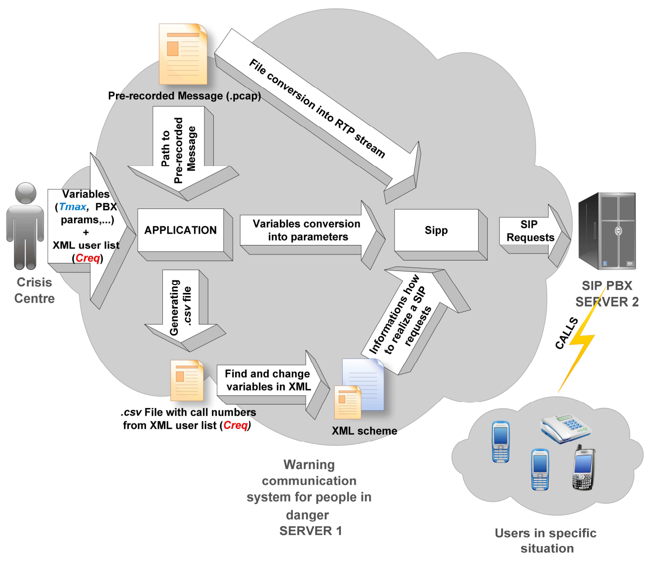

Fig.1 Alert System Scheme.

This is the purpose of the .csv file which is generated by the application directly after the list of end users has been loaded. This is where the XML scheme gets the telephone numbers and Sipp creates the SIP message for the softswitch. Fig. 1 illustrates the scheme. Another issue addressed while developing the system was how many messages and subsequent calls are we able to generate at a particular moment. This value depends on computing performance requirements of the server with the system and the softswitch, required bandwidth on the transfer line between the system and the SIP softswitch and finally maximum load of the Sipp application.

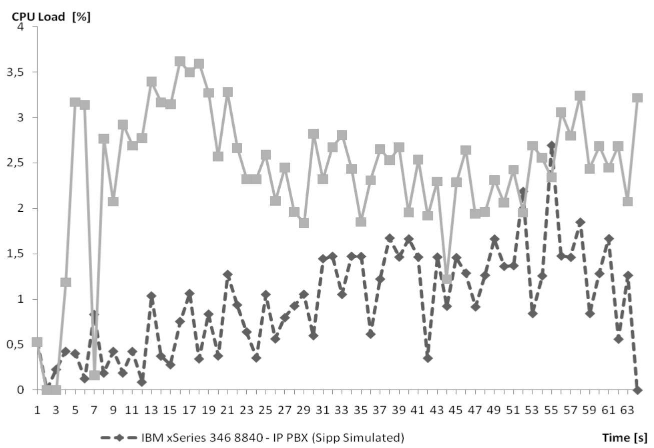

Fig.2 System and IP PBX CPU Load.

The configuration we tested consisted of Dell PowerEdge R510 with our system, the recipient was the IBM server xSeries 346 8840 with the Sipp application in the server mode which simulates the behavior of a SIP softswitch, including generating calls based on the SIP messages received and playing voice messages to end users. The test itself simulated generation of 500 SIP requests with a 60 second long pre-recorded voice message at a particular moment. The resulting values were obtained using the dstat tool. Fig. 2 shows that the percentage load of CPU of both servers has not exceeded 4 %. If we used a real SIP IP softswitch instead of Sipp application in the server mode, the load on CPU would have been higher. On the other hand, we tested using 60 second long pre-recorded voice message. In the real operation, the maximum duration of the voice message will be 30 seconds. It derives from the test that computing performance requirements on hardware do not represent a limiting factor for the maximum amount of calls generated.

In any case, we recommend installing the system and the SIP softswitch separately of two different servers in order to ensure uninterrupted and faultless operation of the whole system. The softswitch's IP address (domain), login name and account password are entered into the system by the staff of the crisis centre. As regards the required bandwidth on the transfer line between the system and the SIP softswitch, it can also be influenced by the choice of the voice codec applied for the transmission of the voice message. To calculate the real bandwidth BW_M [bps], we used formula (1) where H [b] is the sum of all headers [12].

$$BW_M = M \cdot C_R \cdot \left(1 + \frac{H_{RTP} + \sum_{j=1}^{3} H_j}{P_S}\right) \quad (1)$$

The parameter C_R [bps] is the codec's transfer speed (codec rate) and the formula (2) determines the gap between the packets with samples (timing in process of packetizing), where P_S [b] is the payload size [13].

$$\Delta t = \frac{P_S}{C_R} \quad (2)$$

The following table shows the calculation of the bandwidth for Ethernet using four most frequently used codecs. The values have been calculated for the range between 100 and 500 calls. You can see that for instance for codec G.711, the bandwidth for 500 calls is 45.2 Mbit/s. If we wanted to increase the number of calls let's say to 1 000 while using the same codec, we would get to the threshold of the transfer capacity of a standard 100 Mbit/s line. When using GSM codec, we would need to generate 5 000 calls at a particular moment to reach this threshold. These values indicate that the bandwidth may be the limiting factor for the ultimate number of calls generated and that their number reflects the codec used.

Table 1 Different Codecs and their Bandwidth.

Number of Calls	Codecs and their Bandwidth [Mbps]			
	G.711	G.729	G.723 - ACELP	GSM-EFR
100	9,04	3,44	2,4	4
200	18,08	6,88	4,8	8
300	27,12	10,32	7,2	12
400	36,16	13,76	9,6	16
500	45,2	17,2	12	20

The last factor which can influence the number of system-generated calls at a particular moment is the maximum load of the Sipp application. While testing, we established that Sipp is the biggest constraint of the whole system. This open-source tool can generate a maximum of 700 SIP requests at a particular moment without a fault. For higher values, the generated INVITE requests contain errors in structure or length. This is why we set the maximum amount of calls generated at a particular moment to 500. In order to be able to distribute even to sets of end users exceeding 500, it was necessary to divide the total amount of requested calls into subsets of 500 requests. We refer to these subsets as groups and set the interval between individual groups to 60 seconds. If we take into account that the maximum length of a pre-recorded voice message is 30 seconds and staff can only set the maximum duration of ringing to 15 seconds, the resulting time is 45 seconds. This means that the call length never exceeds 45 seconds. This indicates that our system can generate 500 calls every 60 seconds. The following chapter describes algorithms used by the system to manage the groups, missed calls and total time necessary for call distribution.

3 Used Algorithms and Methodology of System Behavior

The previous chapter states that the maximum number of calls which the system can generate at a particular moment is 500. In the text and figures below, this value is referred to as *Cmax*. Once the staff member logs in s/he can enter the XML list of end users into the system. The system immediately calculates the total number of calls to be generated *Creq*, number of groups to which the calls will be divided *Gn* and the estimated time to send all calls *Tsnd* [s].

$$Gn \cong \frac{Creg}{Cmax} \qquad (3)$$

The number of groups *Gn* is determined using formula (3), estimated time *Tsnd* using formula (4).

$$Tsnd \cong \left(\frac{Creg}{Cmax}\right)\cdot 60 \qquad (4)$$

If, for instance, staff member enters a list with 6800 end users, we will obtain the following values: *Creq* = 6800, *Gn* = 14 and *Tsnd* = 816 s.

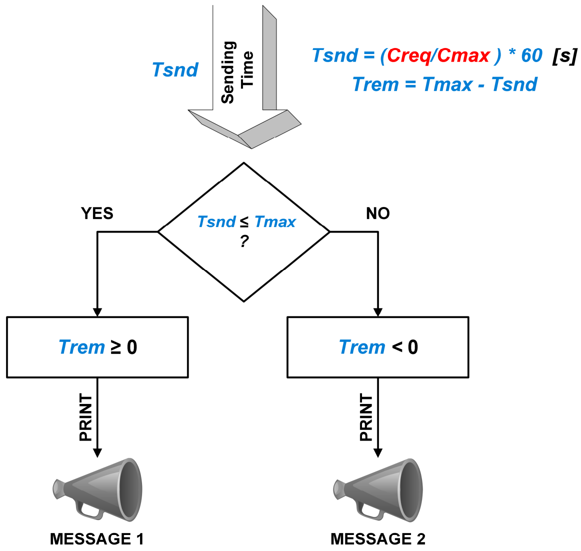

Fig.3 Time Diagram.

In case the threat of danger or natural disaster becomes real, staff has exact time models setting the maximum time limits in which all end users should be alerted. This time value is inserted into the system form and is described as the maximum time to send all planned calls *Tmax* [s]. Fig. 3 illustrates two situations which can arise after the value is inserted. In the first case where *Tsnd* ≤ *Tmax*, there is time remaining to re-send unanswered calls *Trem* [s] after all calls have been sent for the first time. In this case, the system generates the value of time remaining *Trem* and notifies the staff that during this time, the system is to automatically re-initiate calls which have not been answered in the first wave. Where the end user does not answer the call for the second time either, the calls are being re-initiated until *Trem* = 0. In the latter case, where *Tsnd* > *Tmax*, the system returns information that the time required to initiate all calls *Tsnd* is longer that the maximum time for sending requests set by staff *Tmax* and therefore it is not possible to guarantee that all end users will be contacted and requests for unanswered calls re-generated. If staff want to contact all end users, or even set aside time to re-initiate unanswered calls, they need to increase *Tmax* so that Tmax > Tsnd. This is how the staff gets an idea of the time plan to generate calls and can thus make further steps to address the situation.

Other values which staff need to enter into the ready-made forms before calls can actually start to be generated is the duration of the ringing at the end user's side, login name and password to the SIP account and of course the location of the pre-recorded voice message.

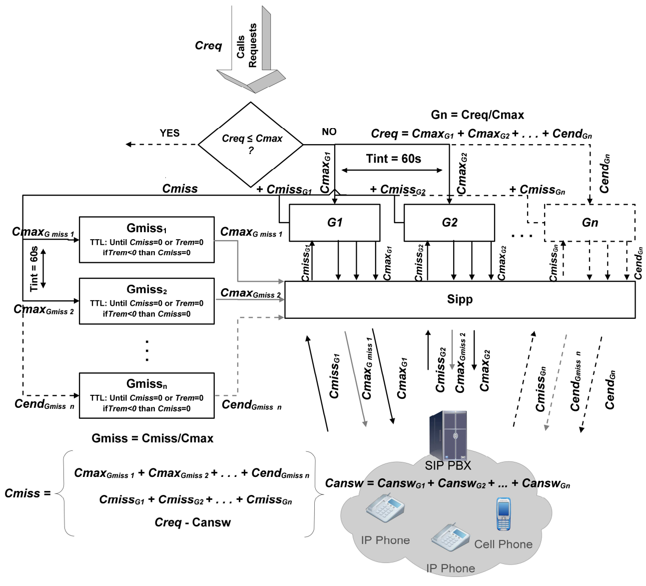

Fig.4 Calls Diagram, *Creq* > *Cmax*.

Once all forms have been filled, the system can start initiating calls according to pre-set parameters. Staff members launch the call generation by pressing the SEND button. At first, the system logs into the IP switchboard and the SIP account created for this purpose and then it starts to generate SIP requests using the Sipp application depicted on Fig. 4. Generating is carried out across individual groups (*G1* - *Gn*) with 60 seconds interval. Each group contains *Cmax* call requests, the last group contains the remaining requests (*Cend*). By adding together all *Cmax* and *Cend*, we obtain a total number of call requests *Creq* (5). Requests are dispatched to the switchboard in sequence and the switchboard starts to initiate individual calls. At this stage, the end device (cell phone, fixed line, etc.) of the end user starts to ring and keeps ringing for the time defined by staff in the system form. If the end user answers the call within this time span, a voice message containing information about the danger and how to address it is played to him/her. This call is marked as answered (*Cansw*). If the end user fails to answer the call during the defined period of ringing, or the end device is not available, the call is marked as unanswered (*Cmiss*). Once all requests across all groups have been generated, the system adds together all unanswered calls (6). If *Trem* > 0, the system starts to generate SIP requests for call previously unanswered. Where *Cmiss* > *Cmax*, such calls are again subdivided into individual groups (*Gmiss1* – *Gmissn*). The number of these groups can be determined using formula (7). Unanswered calls are re-initiated until *Trem* = 0 or until there is no unanswered calls *Cmiss* = 0.

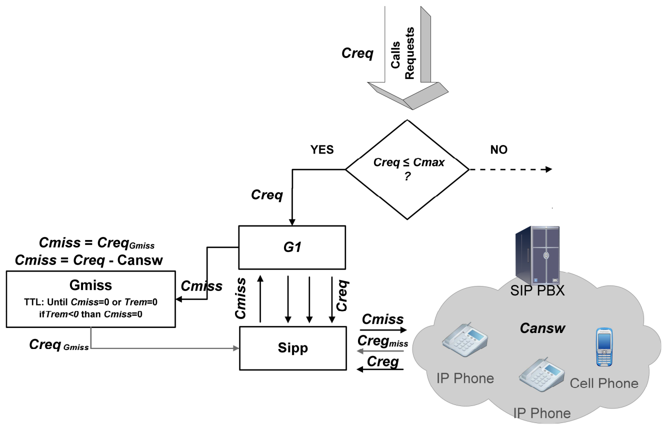

Fig.5 Calls Diagram, *Creq* ≤ *Cmax*.

4 System Efficiency and Spectrum of Utilization

While generating, the system indicates the total number of requests *Creq*, number of groups *Gn* and *Gmiss*, total number of answered calls *Cansw*, number of unanswered calls *Cmiss* and times *Tsnd* and *Trem*. If an unexpected situation occurs, the whole process can be aborted by pressing the ABORT button.

$$Creq = Cmax_{G1} + Cmax_{G1} + ... + Cend_{Gn} \tag{5}$$

$$Gmiss \cong \frac{Cmiss}{Cmax} \tag{6}$$

$$Cmiss = \left\{ \begin{matrix} Cmax_{Gmiss1} + Cmax_{Gmiss2} + ... + Cend_{Gmissn} \\ Cmiss_{G1} + Cmiss_{G2} + ... + Cmiss_{Gn} \\ Creg - Cansw \end{matrix} \right\} \tag{7}$$

Fig. 5 illustrates the situation where *Creq* ≤ *Cmax* and values of *Gn* and *Gmiss* equal 1. The definition of individual variables is the same as for Fig. 4.

The whole system was designed to distribute pre-recorded voice messages to alert people to danger or natural disaster. Its aim is to provide sufficient information to large population on how to address a particular situation in the shortest time possible in a form which cannot be missed or disregarded. Depending on the content of the pre-recorded message, the system may not only function as a part of the early warning scheme, but it can also function as the basic infrastructure for other areas of human activities. Below we provide an overview of situations in which the system could be used:

- Alert system for population safety during natural disasters (floods, fires, earthquakes, wind storms, snow breaks, dangerous substance leakages), conflicts (attacks, wars, raids, army drills).

- Information system: during traffic congestion, to distribute poll results, to distribute election results, to announce sports competition results, to broadcast news (e.g. for visually impaired).
- Advertising system: to distribute advertisements, to address voters, to announce lottery winners.

Whatever situation the system is applied for, we can always reliably calculate the time necessary to address the target group of end users. We use formula (4) where *Creq* is the total number of end users to be targeted and *Cmax* is set to 500. If it was necessary to inform all students at our university (24 058) about a certain event within 1 hour, our system would manage it. The same applies in case a natural disaster should occur – the system is able to advise community population within a certain time span depending on the population size.

The following table illustrates the time (in hours) necessary to generate all requests to selected urban community or institution.

Table 2 Sending Time for Different Population.

	Example	Population	Sending Time -*Tsnd* [h] approx.
Village	Dobešov	243	0,008
Small Town	Odry	7 467	0,25
City	Ostrava	314 666	10,5
Big City	Krakow	756 583	25
	Prague	1 285 995	43
University	VSB-TUO students	24 058	0,8

5 Conclusion and Results

We developed a system which can distribute voice messages in the form of telephone calls using VoIP technology and SIP protocol fast and efficiently. It uses open-source application Sipp and any compatible switchboard enabling for registration of an SIP account. The benefit of using voice message to transfer information as opposed to other forms of information systems is that the information can be easily perceived by target audience, thus reducing the probability of it being missed. The system is controlled through a web interface, which makes it accessible from any machine with an Internet browser. At present, the system is subject to intensive beta testing. We plan to implement several enhancements in the future, such as using voice messages in the .wav format or a more user-friendly installation tool available through the web interface. As the Sipp application develops, we expect to increase the maximum amount of generated SIP requests at a particular moment. Subsequently, the number of requests generated within a given time span will increase.

Acknowledgment:
The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 218086.

References:
[1] A. Johnston, *SIP:Understanding the Session Initiation Protocol*, Artech House Publishers; 3 edition, 2009
[2] J. Meggelen, J. Smith and L. Madsen, *Asterisk The Future of Telephony*, O'Reilly Media; 2nd edition, 2007
[3] F. E. Goncalves, *Building Telephony Systems with OpenSIPS 1.6*, Packt Publishing, 2010
[4] R. Scoular, *Apache: The Definitive Guide*, O'Reilly Media; 3rd edition, 2002
[5] P. DuBois, *MySQL Cookbook*, O'Reilly Media; 2nd edition, 2007
[6] R. Lerdorf, *Programming PHP*, O'Reilly Media; 1st edition, 2002
[7] I. Pravda, J. Vodrazka, *Voice quality planning for NGN including mobile networks*, 12th International Conference on Personal Wireless Communications (PWC 2007), Prague, Book Series: International Federation for Information Processing, vol. 245, 2007, pp: 376-383
[8] B. Marchal, *Applied XML Solutions*, Sams, 2000
[9] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. *RFC 3550, A transport protocol for real-time applications*, IETF: Technical report, 2003
[10] *SIPp*, URL: http://sipp.sourceforge.net
[11] M. Voznak, F. Rezac and J. Ruzicka, *Security Risk in IP Telephony*, CESNET Conference, Prague, 2008
[12] M. Voznak, *Voice over IP*, VSB-Technical University of Ostrava: College Textbook, 1st. ed., Ostrava, 2008, p. 176
[13] P. Polacek, I. Baronak, *Enhanced Equivalent Capacity Method*, In Proceedings of the IADIS International Conference – Informatics 2009, Algavre, Portugal, pp. 192-196