

# P2P Audio/Video Protocol with Global Positioning Data in Real Time for Mobile Devices

José-Vicente Aguirre, Rafael Álvarez, Leandro Tortosa, and José-Francisco Vicent

**Abstract**— In this paper, we propose an original method to geoposition an audio/video stream with multiple emitters that are at the same time receivers of the mixed signal. The obtained method is suitable when a list of positions within a known area is encoded with precision tailored to the visualization capabilities of the target device. Nevertheless, it is easily adaptable to new precision requirements, as well as parameterized data precision. This method extends a previously proposed protocol, without incurring in any performance penalty.

**Keywords**— Codification, geodesic coordinates, Multiparty, Stream, VoIP, Videoconference, P2P, Security, Pocket PC, Smart Phone, PDA.

## I. INTRODUCTION

**N**OWADAYS, the increase of domestic available bandwidth and computing power is making videoconferencing a reality in situations that were unthinkable a few years ago. Applications like skype[1], qik, etc., try to take advantage of this scenario and provide new features and functionality. As an example, by using videoconferencing, people residing in distant areas can perform meetings and cooperative work successfully, despite the possible geographical restrictions.

Any application where multiple users can interact by means

Manuscript received December 29, 2008. This work was partially supported by the Spanish grants GVPRE/2008/363

José-Vicente Aguirre is with the Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Campus de Sant Vicent del Raspeig, Ap. Correos 99, E-03080, Alicante, SPAIN (e-mail: jaguirre@dccia.ua.es).

Rafael Álvarez is with the Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Campus de Sant Vicent del Raspeig, Ap. Correos 99, E-03080, Alicante, SPAIN (e-mail: ralvarez@dccia.ua.es).

Leandro Tortosa is with the Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Campus de Sant Vicent del Raspeig, Ap. Correos 99, E-03080, Alicante, SPAIN (e-mail: tortosa@dccia.ua.es).

José-Francisco Vicent is with the Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, Campus de Sant Vicent del Raspeig, Ap. Correos 99, E-03080, Alicante, SPAIN (e-mail: jvicent@dccia.ua.es).

of audio and video channels under a Peer to Peer (P2P) communication [1] [2], suffers many problems related to the available bandwidth and computing power [6] [8], which can be critical as the number of users rises. Any solution to these problems implies restrictions in the way the users interact with each other or in the quality of the communication.

The geopositioning of any type of multimedia content is a current trend that can be seen in all Internet content sharing services (picasa, qik, etc.). Being able to transmit securely the location of the audio/video stream in real time is an interesting application, especially when it does not imply any loss of quality which is already severely limited by the available resources and bandwidth.

There are several ways to encode global positioning coordinates. The geographic coordinates system uses two angular coordinates in a general spherical coordinates system used in Astronomy. The two angular coordinates are related to the following angles, measured from the center of the Earth:

- The latitude of a point on the Earth's surface is the angle (measured in degrees) between the plane of the equator and the straight line segment that joins the point to the center of the globe.
- The longitude of any point on the Earth's surface is the measure of the angle (in degrees) between the planes that contain the point, the Earth's axis and the Greenwich Meridian (adopted as reference).

Another coordinate system is the Universal Transverse Mercator (UTM). A position on the Earth is referenced in the UTM system by the UTM zone, and the easting and northing coordinate pair. The easting is the projected distance of the position from the central meridian, while the northing is the projected distance of the point from the equator. The point of origin of each UTM zone is the intersection of the equator and the zone's central meridian.

The use of a concatenation of coordinates for the transmission of all the positions of a set of nodes could be a waste of precision or possible locations, when transmitting the information of the position from many nodes simultaneously is required and the transmission bandwidth and the data size are relevant.

We have observed certain restrictions in the underlying problem, which would provide an opportunity for reducing the

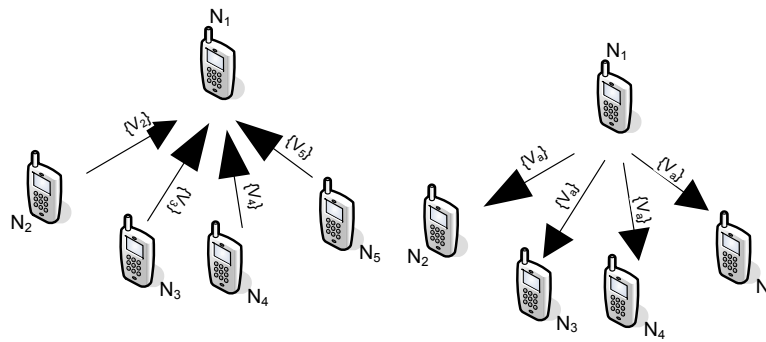


Fig. 1 Client-server Multi Party VoIP

amount of data transmitted and allowing it to be adapted to our voice/video transmission system [3].

In this paper, we propose a technique for adding a geopositioning signal corresponding to the  $N$  participants in a multi-party videoconferencing, so that the accuracy is tailored to the bandwidth unused by the audio/video channels, without producing delay. The main idea is to propose a lossy positioning information compression technique based on three factors: the minimum precision required, the range of probable positions and the number of bits available for geopositioning information. Also, we explain our P2P audio/video stream processing system to balance computational and network resources load around all machines involved in MVoIP communication.

Compared to the current state of the art, this approach provides three novelties:

- First, the protocol performs a fair load distribution of the data mixing and transmission operations, so that no machine performs more work than the others. Compared with the sequential server scheme (see Fig.1) each client has to perform only one third of the transmission and half of the mixing operations that would be necessary if that machine was the mixing server for all of the machines.
- Second, the protocol is fully distributed and self-

organizing.

- Finally, the protocol guarantees that the audio mixing phase produces the audio distribution implicitly, so that when the mixing phase ends there is no audio distribution to make because all the machines already have the audio.

It is an adequate protocol for communicating two or more machines of limited resources (mobile phones or PDAs for example) without employing a specialized server (see Fig. 1) or promoting one of the machines as server.

## II. NOTATION

We use the following notation in this paper:

- A devices ring is a subset of devices with modular sequential order and some characteristics in common.
- $R_R$  is the real devices ring.
- $R_C$  is the connected devices ring.
- $N_R$  is the total number of machines in  $R_R$
- $N_C$  or  $N$  is the total number of machines in  $R_C$ .
- $n$  is the current machine in  $R_C$ .
- $I$  is the total number of iterations of the algorithm.
- $i$  is the current iteration of the algorithm.
- $Mix_{y=0}^x(v_y)$  is a packet mixing function that mixes from  $y=0$

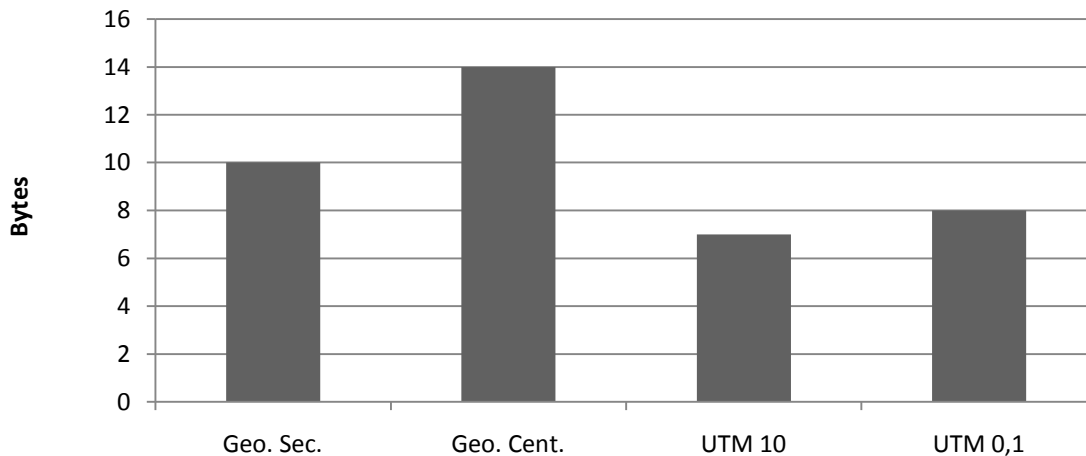


Fig. 2 Number of bytes required for coordinate representation.

	Precision	Latitude Precision	Longitude Precision	bits		bits Total	Bytes
				Lat	Lon		
Sec.	0.01	0.24 m	0.31 m	26.65	25.65	53	6.63
	0.1	2.40 m	3.08 m	23.48	22.48	46	5.75
	1	24.01 m	30.76 m	20.31	19.31	40	5.00
	10	240.08 m	307.64 m	16.98	15.98	33	4.13
Min.	1	1440.50 m	1845.83 m	14.40	13.40	28	3.50
	10	14405.00 m	18458.33 m	11.08	10.08	22	2.75
Deg.	1	86430.00 m	110750.00 m	8.49	7.49	16	2.00
	10	864300.00 m	1107500.00 m	5.17	4.17	10	1.25

Table 1 Byte and precision tradeoff for latitude and longitude in an area 40°N from the equator.

to x.

- $N$  is the total number of machines.
- $n$  is the current machine.
- $V_n$  is the current voice packet of node n.
- $V_a$  is the fully mixed voice packet ready for playback.
- Parallel  $\{\{Job1\} \{Job2\}\}$  denotes that jobs 1 and 2 are carried out in parallel.

### III. PROTOCOL SPECIFICATION

In the following, we describe the protocol specification and requirements.

#### A. Requirements

This method for embedding the geopositioning signal in an audio/video data stream in real time, although valid for many other purposes, is a part in the development of our secure communications system [3], determining the supposed initial situation with its related bounds and problem restrictions.

Firstly, the system provides an environment where the connection changes and, therefore, the number of currently connected devices with the capability to participate in an audio or audio/video communication,  $N_C$ , out of the total number of ready devices,  $N_R$ , is known and efficiently managed.

On the other hand, the system must provide mechanisms to establish and manage audio or audio/video communications in real time, ensuring the integrity, confidentiality and non-repudiation of data transmitted in the stream according to the policies specified for such communication.

Finally, the method used for mixing and transmitting all audio and video sources [3] [4] [10] [11] provides, depending on the terms of communication, a variable difference between the number of bits required by each package and the Maxim Transfer Unit (MTU) of the network, which can be used to transmit small amounts of data, like geopositioning. It is not optimal to send this data in new packages while the stream of data in real time is happening, since the high latency of data networks for mobile [5] networks would introduce significant delays in the communication noticeable to the human ear.

#### B. Position Coordinates Encoding

The amount of bytes required to represent the coordinates in different systems and different precision is shown in Fig. 2:

Geographical Coordinates with second precision (approximately, and depending on the area, corresponds to a precision of about 25m) and of cents of seconds (around 0,2m); and UTM coordinates with a precision of 10m and 0,1m.

Due to the specific requirements of the problem under study, the concatenation of all node positions could mean a waste of precision or possible locations. The different coordinate systems have been designed for global positioning; in our case we will need to position a certain number of nodes in their respective geographical locations; the further away they are located from each other, the location precision becomes less relevant while as they get closer the total possible area for location becomes smaller.

This reasoning comes from the fact that the location signal to be transmitted is meant to be a general view of the position of every node that will be represented as a set in a mobile device. Thus, precision is useless when dealing with large distances, if more precision is needed for a specific node's location then a direct query for that node's location will be performed, but this is out of the scope of this paper.

In order to achieve this, we define an action window where all nodes' location will be represented with a level of precision suitable to the visualization capabilities and free bits for transmission available. The encoding scheme is shown in Fig. 3.

In the following, we detail the different steps in the encoding.

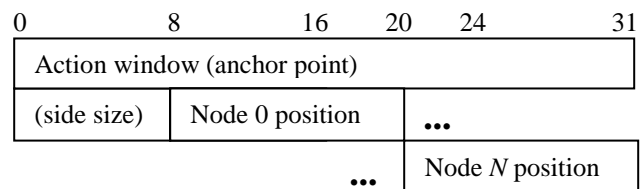


Fig. 3 Encoding bit arrangement.

#### 1) Action Windows

The action window is defined as an area of the Earth surface that contains all node locations. A huge level of precision is not required to define this window although it must be able to

adapt to large scales (whole Earth) or small ones (a town) as required.

The action window is defined by its anchor point (upper-left corner) and a certain length. The latter is used to specify size of the square containing the area. This is done in this way to reduce the number of bytes required to represent it, avoiding the need to encode two points or any unnecessary precision for its purpose.

Observing Table 1, we see that the reason to choose a minutes precision level is because it achieves a good tradeoff between precision and the amount of bytes required.

The four remaining bits in the encoding of the anchor point are used to encode the scale in which the square side length will be specified in the next byte as shown in Table 2. In this table we can observe the 16 different ways to distribute the 8 bits of the byte representing the action window side size, that have been chosen so they offer an adequate range of maximum distances and minimum precisions.

### 2) Position List

In the proposed encoding method the precision level is meant to always be less than the maximum achievable precision. In our case of study we define the maximum precision as the one with which a coordinate can be selected within our action window, visualized in full screen on a mobile device with a maximum resolution of 640x480 pixels. The maximum precision would be a single pixel, so the maximum achievable precision would be the action window side size divided by the greatest dimension of the resolution (640).

Each node will be positioned in the action window determining its position within an  $n$  by  $m$  elements grid. Since, in order to successfully represent graphically a node, a square of 10 pixels of side will be used, we consider a grid of 64x64 elements as an adequate representation precision.

Considering the required precision, one and a half bytes (12 bits) allow concatenating the different node positions while only wasting 4 bits when the number of nodes is odd.

#### C. Protocol Definition

The protocol proposed in previous papers [3] [4], consisted, regarding the transmission of an audio/video stream, of a stream establishment phase and of a transmission phase. The algorithm corresponding to the transmission phase is shown in Fig. 8, where it can be observed that it takes  $\log_2(N)$  iterations for each transmission during the sub-mixing phase.

This protocol is defined as a packet mixing and distribution algorithm in a network of  $N$  machines.

The general algorithm shows adequate packet distribution behaviour in the case that  $N = 2^l$ , but a more detailed study is necessary when this is not true.

There is a subset of these cases for which the algorithm can be adapted without any performance impact; for the rest of these cases that do impact performance, we present several possible alternatives giving as a result an adapted version of the algorithm.

### 1) General algorithm

In the case of having  $N$  machines connected in a virtual ring, with sequential numbering, so that each machine has a fixed number from 0 to  $N-1$ , we can establish the emitting and receiving nodes with

$$N_e(n, i) = n + 2^{i-1} \bmod N \quad (1)$$

and

$$N_r(n, i) = n - 2^{i-1} \bmod N, \quad (2)$$

being  $N_e(n, i)$  (see (1)) the node to which  $n$  must send  $P_e$  (see (9)) during iteration  $i$ ; and  $N_r(n, i)$  (see (2)) the node from which  $n$  must receive  $P_r$  (see (8)) during iteration  $i$ .

With the previous specifications, we can define an algorithm (see Fig. 4) whose mixing and distribution characteristics are defined in equations (3) to (7).

$$P_e(n, i) = \underset{y=0}{\text{Mix}}^{2^{i-1}-1}(V_{(n-y) \bmod N}) \quad (3)$$

$$P_r(n, i) = \underset{y=0}{\text{Mix}}^{2^{i-1}-1}(V_{((n-2^{i-1})-y) \bmod N}) \quad (4)$$

$$P(n, i) = \underset{y=0}{\text{Mix}}^{2^i-1}(V_{(n-y) \bmod N}) \quad (5)$$

$$V_a(n, i) = \underset{y=1}{\text{Mix}}^{2^i-1}(V_{(n-y) \bmod N}) \quad (6)$$

$$DV_a(n) = \underset{y=1}{\text{Mix}}^{N-1}(V_{(n-y) \bmod N}) \quad (7)$$

In this way, we can define the following:

$P_e(n, i)$  (see (3)) corresponds to the composition of the packet that node  $n$  will have to send during iteration  $i$ .

$P_r(n, i)$  (see (4)) corresponds to the composition of the packet that node  $n$  will have to receive during iteration  $i$ .

$P(n, i)$  (see (5)) is the final packet that node  $n$  will have composed after the reception of the last packet during iteration  $i$ .

$V_a(n, i)$  (see (6)) is the accumulated composition of the voice packet for playback at node  $n$  during iteration  $i$ .  $V_a$  differentiates from  $P$  in that it does not include  $V_n$ .

$DV_a(n)$  (see (7)) is the desired accumulated voice packet composition for playback at node  $n$  during iteration  $i$ .

In a recursive way, closer to the real behaviour of the algorithm, the previous functions can be defined as shown in equations (8) to (11).

$$P_r(n, i) = P_e(N_r(n, i), i) \quad (8)$$

$$P_e(n, i) = P(n, i-1) \quad (9)$$

$$P(n,i) = \begin{cases} \text{Mix}(P(n,i-1), P_r(n,i)) & \text{if } i > 0 \\ V_n & \text{if } i = 0 \end{cases} \quad (10)$$

$$V_a(n,i) = \begin{cases} \text{Mix}(V_a(n,i-1), P_r(n,i)) & \text{if } i > 0 \\ \{\} & \text{if } i = 0 \end{cases} \quad (11)$$

Employing equation ( 6 ) we can obtain the table shown in Fig. 4 that represents the audio packets mixed for playback at machine  $n$  with a total of  $N$  machines.

This table represents the values of  $V_a(n,i)$  for node  $n = N-1$  because it is the most clear case, having  $V_k$  values decreasing from  $k = N-2$  to  $k = 0$ .

Observing this table (see Fig. 6), we can extract three different cases as a function of the correspondence of the generated  $V_a$  with the  $DV_a$  (desired  $V_a$  see equation ( 7 )).

```

Function TransmitVoice (VoicePacket myVoice, int
numNodes, int myPosition)
{
  N= numNodes;
  n= myPosition;
  AllPacketReceived.add ( myVoice );
  For (i=1; i <= log2(N); i++)
  {
    NodeDestination = n + 2i-1;
    NodeOrigin = n - 2i-1;

    Parallel
    {
      {
        PacketReceive = receive ( NodeOrigin );
        AllPacketReceived.add (
          Mix (PacketReceive, AllPacketReceived [i-1] )
        );
      }

      {
        PacketSend = AllPacketReceived [i-1];
        Send(NodeDestination, PacketSend);
      }
    }
  }
}
    
```

Fig. 4 General algorithm

The first case is when  $N = 2^l$  or, more clearly, when  $N$  is a power of 2.

The second case is when  $N <> 2^l$  and  $N = 2^{l-1} + 2^x$ , where  $x < l$  or when the necessary  $P_e(n,l)$  to obtain  $DV_a(n,l)$  is a  $2^x$  mix of machines less than  $2^{l-1}$  (that would be the first case).

The third case is when  $N <> 2^l$  and  $N <> 2^{l-1} + 2^x$  where  $x < l$  or, more concisely, when the  $P_e(n,l)$  necessary to obtain  $DV_a(n,l)$  must be created with more than one packet of size  $2^x$ . In the following, we provide a detailed study of each one of these cases.

Position	Precision and bits				Min. precision	Max. distance
	Deg	min	sec	Cent sec		
0	Precision bits	1 8	1 0	1 0	1 0	1° 256°
1	Precision bits	1 0	1 8	1 0	1 0	1' 4° 16'
2	Precision bits	1 0	1 0	1 8	1 0	1" 4' 16"
3	Precision bits	1.5 8	1 0	1 0	1 0	1.5° 384°
4	Precision bits	1 0	1 0	1 12.5	12.5 3	0.12 5" 33"
5	Precision bits	1 0	1 0	2 5	25 3	0.25' 1' 6"
6	Precision bits	1 0	1 5	7.5 3	1 0	7.5" 33'
7	Precision bits	1 0	2 5	15 3	1 0	15" 1° 6'
8	Precision bits	1 5	7.5 3	1 0	1 0	7.5' 33°
9	Precision bits	2 5	15 3	1 0	1 0	15' 66°
10	Precision bits	2.75 5	20.6 3	1 0	1 0	20.6' 90° 45'
11	Precision bits	1 2	3.75 4	56 2	1 0	56" 5° 3' 45"
12	Precision bits	10 2	37.5 4	563 2	1 0	563" 50° 37' 30"
13	Precision bits	1 0	1 2	3.8 4	93.8 2	0.93 8" 5' 3.75"
14	Precision bits	1 0	15 2	56 4	1406 2	14.0 6" 1° 15' 56.25"
15	Precision bits	1 4	15 2	225 2	1 0	225" 17° 15'

Table 2 Different encodings for the action window side size

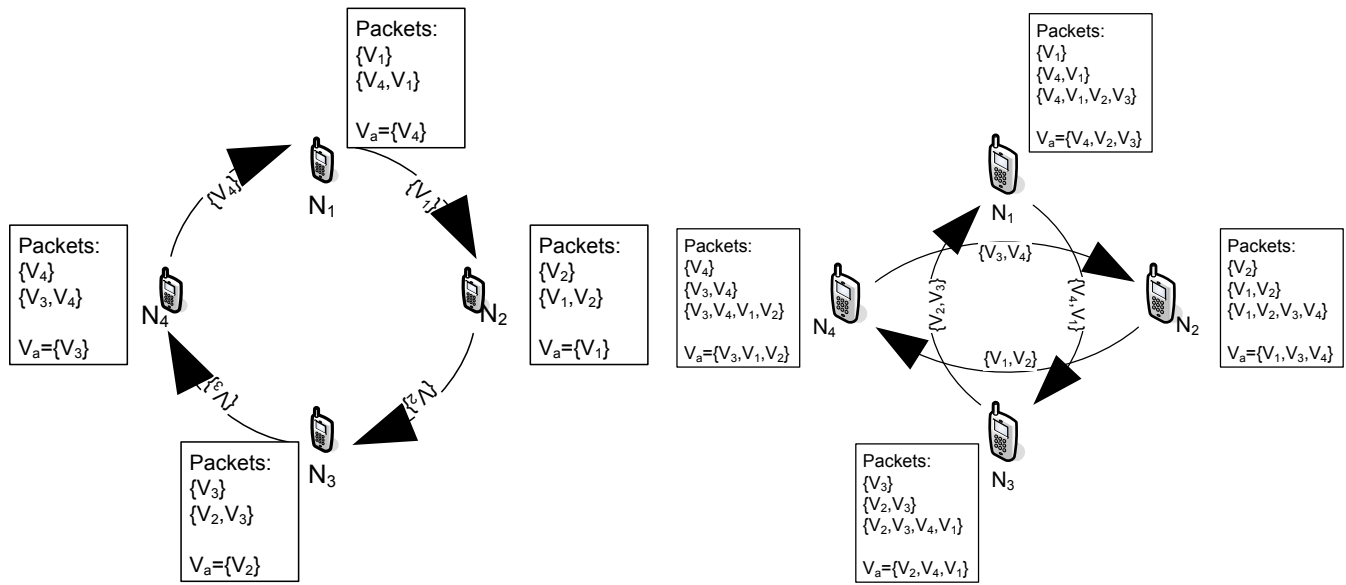


Fig. 5 Our protocol for n=4

2) Case  $N = 2I$

These are the base cases of the algorithm and do not require any modification to the general algorithm to be treated. If the number of users permitted is from 1 to k, there is a  $\lfloor \log_2(k) \rfloor / k$  probability of this case happening.

3) Case  $N < 2I$  and  $N = 2I - 1 + 2x$  where  $x < I$

In this case, with a slight modification of the general algorithm we can achieve the same performance than in the base case. To do so, it is necessary to modify  $P_e$  (equation (13))

$$x = \log_2(N - 2^{I-1}) \tag{12}$$

I	N	n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
1	2	1	0																																	
2	3	2	1	0	2																															
2	4	3	2	1	0																															
3	5	4	3	2	1	0	4	3	2																											
3	6	5	4	3	2	1	0	5	4																											
3	7	6	5	4	3	2	1	0	6																											
3	8	7	6	5	4	3	2	1	0																											
4	9	8	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2																			
4	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4																			
4	11	10	9	8	7	6	5	4	3	2	1	0	10	9	8	7	6																			
4	12	11	10	9	8	7	6	5	4	3	2	1	0	11	10	9	8																			
4	13	12	11	10	9	8	7	6	5	4	3	2	1	0	12	11	10																			
4	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	13	12																			
4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	14																			
4	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																			
5	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2			
5	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	17	16	15	14	13	12	11	10	9	8	7	6	5	4			
5	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	18	17	16	15	14	13	12	11	10	9	8	7	6			
5	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	19	18	17	16	15	14	13	12	11	10	9	8			
5	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	20	19	18	17	16	15	14	13	12	11	10			
5	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	21	20	19	18	17	16	15	14	13	12			
5	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	22	21	20	19	18	17	16	15	14			
5	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	23	22	21	20	19	18	17	16			
5	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	24	23	22	21	20	19	18			
5	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	25	24	23	22	21	20			
5	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	26	25	24	23	22			
5	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	27	26	25	24			
5	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	28	27	26			
5	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	29	28			
5	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	30			
5	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			

Fig. 6 Table of Va for diferent N

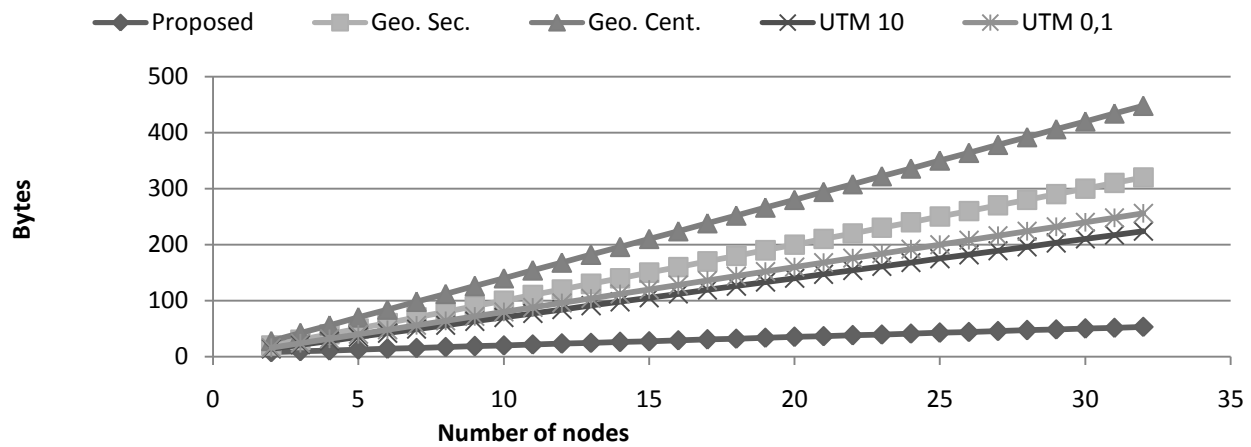


Fig. 7 Encoding sizes in bytes in relation to the number of nodes.

$$P_e(n, i) = \begin{cases} P(n, i-1) & \text{if } i < I \\ P(n, x) & \text{if } i = I \end{cases} \quad (13)$$

If the maximum number of users is from 1 to k (being k a power of 2) there is a probability of

$$\frac{\sum_{x=1}^{\lceil \log_2(k) \rceil} (x-1)}{k}$$

that this case occurs.

4) Case  $N < > 2I$  and  $N < > 2I-1 + 2x$  where  $x < I$

In this case, we cannot achieve the same performance as in the base case. A possible approach would be to add iterations to transmit the required packet sizes so that the  $V_a$  generated would be the same as the  $DV_a$ .

In the worst cases the number of iterations would be  $(\lceil \log_2(k) \rceil - 1) - 1$ . is  $\lfloor \log_2(k) \rfloor$ .

There is a

$$\frac{\lfloor \log_2(k) \rfloor}{k}$$

probability for this case to happen.

In the rest of the cases, the necessary additional iterations are within 1 and  $(\lceil \log_2(k) \rceil - 1) - 2$ .

There is a

$$k - \frac{\sum_{x=1}^{\lceil \log_2(k) \rceil} (x+1)}{k}$$

probability for this case.

5) Final Algorithm

Adding the necessary changes required for all three cases, the final algorithm is as shown in Fig. 8

```

Function TransmitVoice (VoicePacket myVoice, int
numNodes, int myPosition)
{
    N= numNodes;
    n= myPosition;
    AllPacketReceived.add ( myVoice );

    For (i=1; i < log2(N); i++)
    {
        NodeDestination = n + 2i-1;
        NodeOrigin = n - 2i-1;

        Parallel
        {
            {
                PacketReceive = receive(NodeOrigin);
                AllPacketReceived.add ( Mix (PacketReceive,
                    AllPacketReceived [i-1] ) );
            }
            {
                PacketSend = AllPacketReceived [i-1];
                Send(NodeDestination, PacketSend);
            }
        }
    }
    Float X= log2(N - 2i-1)

    If ( x ÷ [x] = 1 )
    {
        NodeDestination = n + 2i-1;
        NodeOrigin = n - 2i-1;

        Parallel
        {
            {
                PacketReceive = receive(NodeOrigin);
                AllPacketReceived.add(Mix(PacketReceive,
                    AllPacketReceived[i-1]));
            }
            {
                PacketSend = AllPacketReceived [ log2(N - 2i-1) ];
                Send(NodeDestination, PacketSend);
            }
        }
    }
}
Else
    TransmitVoice( lastPackets (VoicePacket myVoice

```

Fig. 8 Transmission phase algorithm.

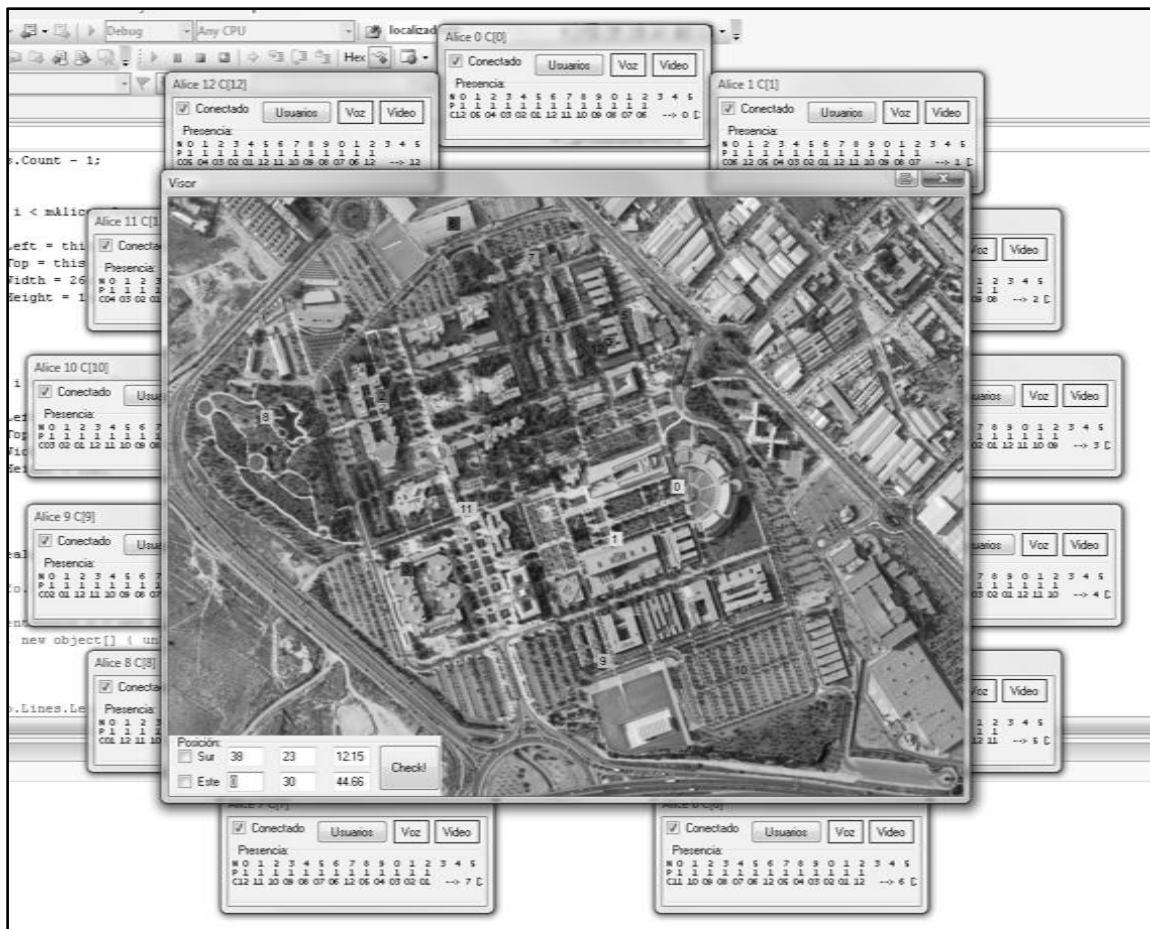


Fig. 9 Simulation with 13 Nodes.

The following additions to the protocol have been necessary in order to incorporate geopositioning to the data stream:

- Reception of each nodes position during the stream establishment phase. With these data the node that started the communication computes, as shown later, the action window parameters.
- In the sub-mixing phase, each node encapsulates the geopositioning data that it has available on each iteration within the audio/video data.
- In case that a dynamic change of the action window parameters is required during the communication, the starting node is in charge of sending the corresponding control signal encapsulated within the stream as specified in the protocol.

#### 6) Action Window Algorithm

The specification of the action window parameters is done with the following process:

- 1 Determine the minimum and maximum longitude and latitude of all node positions, obtaining two coordinates defining a window.
- 2 Extend the obtained area with a distance to the window frame. If not specified otherwise, this is taken as 300 meters.
- 3 The anchor point is obtained by rounding to the nearest position to the one calculated for the window, with a precision level of minutes. This rounding process will always try to find the upper and left-most representable point available.
- 4 By subtracting the lower right corner point with the computer anchor point, we establish the greatest distance (horizontal or vertical) as the square side length.
- 5 To represent the square size length we take the most precise representation in Table 2 that allows reaching the required distance.
- 6 The action window is the one obtained after all rounding processes.



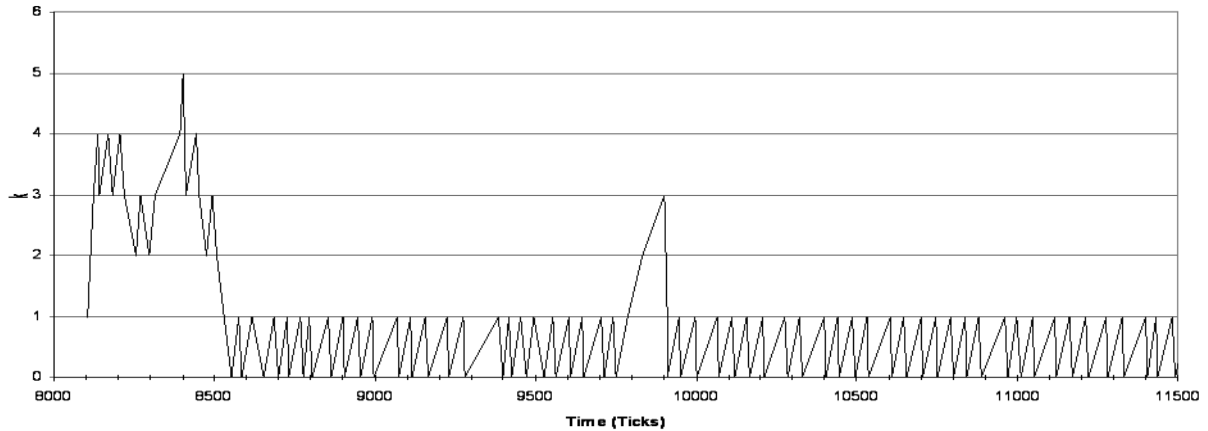


Fig. 10 Buffer state

7) Action Window Dynamic Change Detection Algorithm

In order to detect that a change of action window parameters is required the following procedure should be executed:

- 1 Determine out of all nodes, if any of them lie in the first or last row or column.
- 2 If any of them satisfies such criteria and the action window has not changed during the last hour, then the action window is changed according to the standard frame size.
- 3 If any of them satisfies this criteria and the action window has been changed less than hour ago, then the action window is changed with a frame size corresponding to the difference between a hypothetical new action window with a frame of size zero and the previous action window.
- 4 After an hour has elapsed, a new action window is computed and if the encompassed area is two magnitude units smaller the it is changed with standard frame size.

IV. RESULTS

The encapsulation of the geopositioning data produces the desired effects over the audio/video transmission in real time. Since the positioning data is transmitted using less than 50 bytes (for node sizes inferior to 30, see Fig. 7), the audio/video quality or the transmission times are not affected. Sizes over 60 bytes would provoke serious quality degradation in the video transmission without serious modifications in the stream management scheme.

In Fig. 11 we can observe that the only encoding scheme that never uses more information than what is representable is the one proposed. Since the other schemes use fixed precision, they always reach a point where more information is transmitted than what can be represented. But even in the case of dynamic precision the proposed scheme still more efficient taking up less bytes for the same precision level; as shown in Fig. 7. This size savings increase with the number of nodes.

In Fig. 12 and 10, you can see the evolution of the protocol

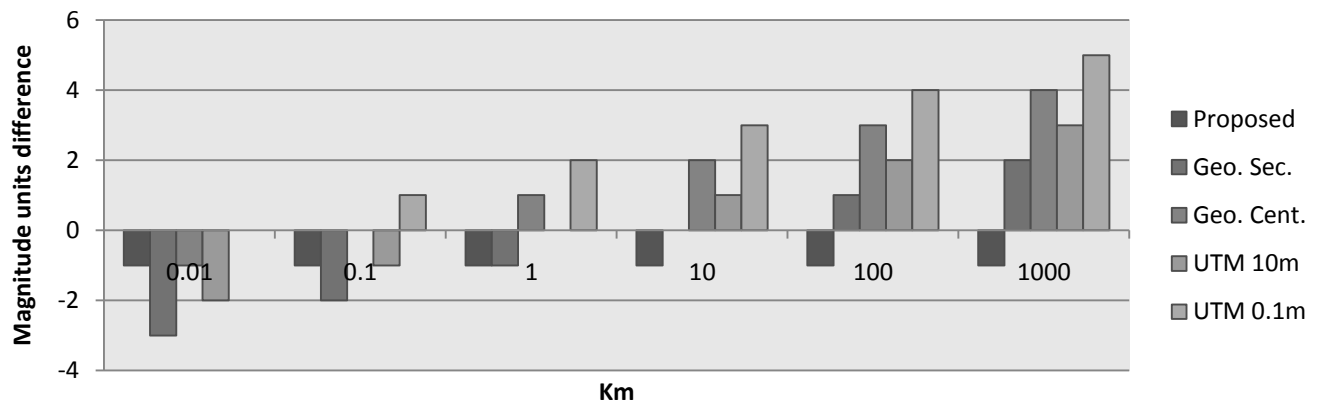


Fig. 11 Magnitude units difference between data precision and maximum achievable precision.

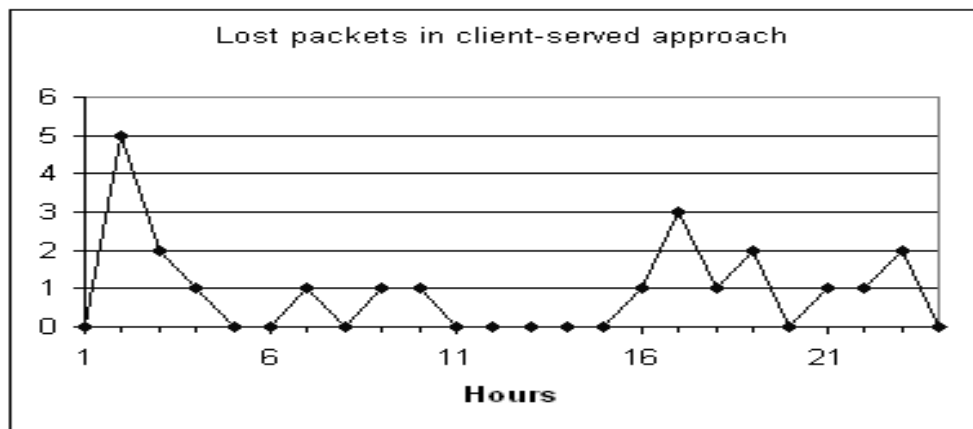


Fig. 12 Number of lost packets in a 24h hours conversation

when two nodes make a voice/video stream.

#### V. CONCLUSION

We have proposed an original method to geoposition an audio/video stream with multiple emitters that are, at the same time, consumers of the mixed signal.

The achieved method is suitable for those comes where a list of positions within a designated area is encoded with a degree of precision adjusted to the visualization capabilities; and is also easily extensible to support new requirements.

The method is designed as an extension to the previously proposed protocol; adding audio/video signal geopositioning capabilities in real time (see Fig. 9) without incurring in any significant performance penalty or loss of features.

As future research, we plan to incorporate single node high precision location queries in order to represent them with the required detail during zoom operations. Also, we plan incorporating a role based security scheme to control access to the geopositioning data.

#### REFERENCES

- [1] Salman A., Baset and Henning Schulzrinne, An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. *Technical Report CUCS-039-04*, Columbia University, 2004.
- [2] Yasusi Kanada, Multi-Context Voice Communication Controlled By using An Auditory Virtual Space, *2nd Int. Conference on Communication and Computer Networks*, 2004.
- [3] José-Vicente Aguirre, Rafael Álvarez, Leandro Tortosa and Antonio Zamora, Secure VoIP and Instant Messaging on Small PDA Devices, *WSEAS Transactions on Computers*, Vol .5, No.1, 2006, pp 171-176.
- [4] José-Vicente Aguirre, Rafael Álvarez And Antonio Zamora, Secure Mobile P2P Videoconferencing Protocol, *Eatis'07 ACM-DL Proceedings*.
- [5] M. Catalan, C. Gomez, D. Viamonte, J. Paradells, A. Calveras, F. Barcelo, TCP/IP analysis and optimization over a precommercial live UMTS network, *IEEE Communications Society / WCNC 2005*, pp. 1503-1508.
- [6] Schulzrinne H., Casner S., Frederick R., Jacobson V. *RTP: A Transport Protocol for Real-Time Applications*, IETF, RFC 3550, 2003.
- [7] RTP: A Transport Protocol for Real-Time Applications, *IETF*, RFC 3550, 2003.
- [8] Kuhn R., Walsh J., Fries S., *Security Considerations for Voice Over IP Systems*, Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, 2004.
- [9] Xiaohui G., Zhen W., Philip S., ZonYin S., Supporting MultiParty VoiceOverIP Services with PeertoPeer Stream Processing, *Proceedings of the 13th annual ACM international conference on Multimedia*, 2005, pp. 303 – 30
- [10] José-Vicente Aguirre, Rafael Álvarez, Julia Sanchez and Antonio Zamora, Broadcast Multiplexing and Subchanneling for Secure P2P Multiconferencing, *WSEAS Transactions on Computers*, Vol .6, No.3, 2007, pp 522-527.
- [11] José-Vicente Aguirre, Rafael Álvarez, José Noguera, Leandro Tortosa and Antonio Zamora, A Viability Analysis of a Secure VoIP and Instant Messaging System On a Pocket PC Platform, *WSEAS Advanced Topics in Information Security and Privacy*, 2005, pp 218-223.