Videoconferencia P2P Segura para Dispositivos Móviles

JOSÉ VICENTE AGUIRRE *RAFAEL ÁLVAREZ* *ANTONIO ZAMORA*
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante (SPAIN)
jaguirre@dccia.ua.es *ralvarez@dccia.ua.es* *zamora@dccia.ua.es*

Este trabajo ha sido parcialmente financiado por el proyecto de investigación español GV06/018

Abstract

In this paper, we present and analyze a lightweight peer-to-peer (P2P) protocol to provide multiparty videoconference services on small platforms like mobile phones or PDAs. These services provide interesting characteristics for many emerging applications that require a common audio and video channel between different users which are far apart but with an internet connection. The proposed protocol is distributed and does not require specialized servers or specific network infrastructure for its operation.

Resumen

En este artículo, se presenta y analiza una propuesta de un protocolo peer-to-peer (P2P) de bajo coste computacional que posibilita servicios de videoconferencia en plataformas reducidas (teléfonos móviles, PDAs, etc). Estos servicios brindan características interesantes para muchas aplicaciones emergentes que requieren un canal de audio y video entre diferentes usuarios alejados entre sí, pero que posean conexión a Internet. El protocolo propuesto se caracteriza por ser distribuido y no requerir servidores especializados o infraestructura especifica de red para funcionar.

1. Introducción

En la actualidad, el aumento del ancho de banda domestico y la capacidad de computo de los dispositivos móviles, esta haciendo posible el uso de video conferencia en dispositivos en los cuales hasta hace poco era impensable. El uso de la videoconferencia posibilita reuniones y trabajo cooperativo entre especialistas en diferentes áreas, a pesar de las posibles restricciones geográficas.

Toda aplicación, en la que múltiples usuarios puedan interactuar mediante canales de audio y video usando comunicación P2P [1,2], produce muchos problemas relacionados con el ancho de banda y la capacidad de cómputo, que pueden resultar críticos cuando el numero de usuarios aumenta sin control. Cualquier solución a este tipo de problemas impondrá, por necesidad, restricciones en el modo en que los usuarios interaccionan o en la calidad la comunicación.

En este artículo se propone una técnica para añadir video conferencia de múltiples usuarios en el esquema de comunicaciones multimedia seguras propuesto previamente [3,4,5], adaptándolo a la transmisión de voz que ya se realizaba en dicha propuesta.

2. Notación

En el artículo, utilizamos la siguiente notación:

- N es el número total de dispositivos.
- n es el dispositivo actual.
- I es el número total de iteraciones en el algoritmo.
- i es la iteración actual del algoritmo.
- F_n es la imagen (*frame*) generada por el dispositivo n.
- $F_a(n)$ es el conjunto de imágenes (*frames*) acumuladas en el dispositivo n.
- $\underset{y=0}{\overset{x}{Sel}}(F_y)$ es la función selectora de imágenes, que aleatoriamente selecciona una imagen F_y para $y=0$ hasta x.
- *Parallel {{Job1} {Job2}}* expresa que *job1* y *job2* son ejecutados concurrentemente.

3. Especificación del Protocolo

A continuación, describiremos los requerimientos y la especificación del protocolo.

3.1. Requerimientos

El protocolo pretende solventar las situaciones en las que N dispositivos necesitan establecer un canal común de video, de tal forma que cualquier maquina pueda ver la información de video transmitida por cualquiera de las otras maquinas en cualquier momento. La transmisión de

video se realiza garantizando un mínimo de imágenes nuevas de video por segundo.
Los requerimientos se establecen teniendo en cuenta las restricciones más estrictas tanto en el lado de los dispositivos, como en el lado del canal de comunicación, que son:

- Todos los dispositivos tienen recursos limitados, de tal forma que ninguno de ellos puede llegar a ser un servidor centralizado o un súper-nodo [1].
- El canal de comunicación, sólo se puede realizar eficientemente en paralelo un envío y recepción simultáneos.

Estas restricciones se encuentran fácilmente en los casos de comunicación de múltiples teléfonos móviles o PDAs.
El estudio del impacto de relajar algunas de estas restricciones queda fuera de los objetivos de este articulo.

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

    Parallel
    {
      {
       PacketReceive = receive ( NodeOrigin );
       AllPacketReceived.add ( PacketReceive )
      }

     {
      PacketSend = Sel(AllPacketReceived);
      Send(NodeDestination, PacketSend);
     }
    }
   }
}
```

Fig. 1 Algoritmo general

3.2. Definición

El protocolo se define como un algoritmo de distribución de paquetes en una red de *N* dispositivos.
En la figura 1, se muestra el algoritmo inicial de partida, que se modificará para permitir distribuir la frecuencia de envió de imágenes. Esto permite que se pueda garantizar un mínimo y un máximo de imágenes por segundo (fps) para cada nodo.

3.2.1. **Algoritmo general.** Teniendo *N* dispositivos conectados en un anillo virtual, con cada maquina numerada secuencialmente desde *0* hasta *N-1*, entonces podemos establecer el nodo de emisión y recepción con

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

y

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

donde $N_e(n, i)$ (ver (1)) es el nodo al que *n* debe enviar P_e (ver (3)) en la iteración *i*; y $N_r(n, i)$ (ver (2)) el nodo desde el que *n* debe recibir P_r (ver (4)) en la iteración *i*.
Con la especificación anterior, podemos definir un algoritmo (ver Fig. 1) donde la forma de seleccionar y distribuir los paquetes viene definida por las ecuaciones (3) y (4).

$$P_e(n,i) = \underset{y=0}{\overset{2^{i-1}-1}{Sel}}\left(F_{(n-y) \bmod N}\right) \qquad (3)$$

$$P_r(n,i) = \underset{y=0}{\overset{2^{i-1}-1}{Sel}}\left(F_{\left((n-2^{i-1})-y\right) \bmod N}\right) \qquad (4)$$

De tal forma, que podemos definir:

- $P_e(n,i)$ (ver (3)) es el F_x que el nodo *n* tendrá que enviar en la iteración *i*.
- $P_r(n,i)$ (ver (4)) es el F_x que el nodo *n* tendrá que recibir en la iteración *i*.

De forma recursiva, y más próxima al funcionamiento real del algoritmo, las ecuaciones anteriores pueden ser definidas tal como aparecen en las ecuaciones de la (5) a la (8).

$$P_r(n,i) = P_e(N_r(n,i),i) \qquad (5)$$

$$P_e(n,i) = Sel(P(n,i-1)) \qquad (6)$$

$$P(n,i) = \begin{cases} \bigcup\left(P(n,i-1), P_r(n,i)\right) & si \quad i > 0 \\ F_n & si \quad i = 0 \end{cases} \qquad (7)$$

$$F_a(n) = P(n,2^I) \qquad (8)$$

Empleando la ecuación (8) podemos obtener la tabla (véase Fig. 2) que representa los paquetes de video que pueden ser seleccionados para su visionado en el dispositivo *n=0*.

N	i=0	i=1	i=2	i=3
3	{0}	{ 1}	{ 2 | 0}	
4	{0}	{ 1}	{ 2 | 3}	
5	{0}	{ 1}	{ 2 | 3}	{ 4 | 0 | 1 | 2}
6	{0}	{ 1}	{ 2 | 3}	{ 4 | 5 | 0 | 1}
7	{0}	{ 1}	{ 2 | 3}	{ 4 | 5 | 6 | 0}
8	{0}	{ 1}	{ 2 | 3}	{ 4 | 5 | 6 | 7}

Fig. 2 *P(0, i)* en diferentes iteraciones

Esta tabla representa los valores de $F_a(n)$ para el nodo $n=0$ por ser el caso más significativo, al representar los valores de F_k desde $k = 0$ hasta $k = N-1$ de forma creciente.

3.2.2. Estudio de probabilidades. Viendo la figura Fig. 2 podemos comprobar las diferentes distribuciones de probabilidades para las *I* imágenes que recibimos, con respecto al total de *N* imágenes que es posible recibir.
Para el caso inicial *i=0*, tenemos la imagen propia con una probabilidad *p=1*. En el caso de *i=1*, por recibir una única imagen, también obtenemos una *p=1* para la imagen

$$F_{N_e(n,i)}.$$

Con *i=2*, al poder recibir dos posibles imágenes, que en origen han podido ser obtenidas con probabilidad *p=1*, nos queda que la probabilidad de recibir una de las posibles imágenes es *p=0,5*. Y así sucesivamente tal y como se describe en la ecuación (9).

$$p_F(x) = \begin{cases} 1 & if \quad x = 0 \\ \dfrac{1}{\lceil \log_2(x+1) \rceil} \cdot p_F\left(x - 2^{\lceil \log_2 x \rceil - 1}\right) & if \quad x > 0 \end{cases} \quad (9)$$

Donde *x* es la posición en la que se recibiría la imagen F_x en el caso de que se enviaran todas las imágenes. Tal y como esta definida la función de envío y recepción en la figura 1, *x* es la distancia del nodo suministrador de la imagen F_x al nodo n que la recibirá. Esta distancia se define en este caso como la posición contando en sentido anti-horario en el anillo de *N* dispositivos comenzando por el dispositivo *n*.
Con esta función de probabilidad, se puede observar que para cada nodo *n*, la probabilidad con la que en cada envío recibirá una F_x viene fijada por su situación en el anillo. Recibiendo siempre con máxima probabilidad $F_{n-1 \bmod N}$ y con la mínima las 2^{I-1} ultimas imágenes.
Para mejorar este caso, se propone una modificación del algoritmo de la figura 1 que permite balancear las probabilidades de recepción de cada F_x cada *N-1* envíos de imágenes.

3.2.3. Algoritmo final. Cambiando las funciones de envío y recepción por las mostradas en las ecuaciones (10) y (11) se consigue balancear la probabilidad de recepción, de modo que cada *N-1* envíos la probabilidad de recepción de cada imagen F_x por cada nodo *n* sea la misma.

$$N_r(n,i,k) = \begin{cases} n + 2^{((i+k) \bmod I)-1} \bmod N & if \quad k \le \dfrac{\lceil N-1 \rceil}{2} \\ n - 2^{((i+k) \bmod I)-1} \bmod N & if \quad k > \dfrac{\lceil N-1 \rceil}{2} \end{cases} \quad (10)$$

$$N_r(n,i,k) = \begin{cases} n - 2^{((i+k) \bmod I)-1} \bmod N & if \quad k \le \dfrac{\lceil N-1 \rceil}{2} \\ n + 2^{((i+k) \bmod I)-1} \bmod N & if \quad k > \dfrac{\lceil N-1 \rceil}{2} \end{cases} \quad (11)$$

Donde *k* es un contador de envíos cíclico desde *0* hasta *N-1*.
Con estas nuevas funciones de nodo de envío y nodo de recepción y utilizando las ecuaciones recursivas (5) a (8) se obtiene el nuevo algoritmo para el envío y recepción de imágenes en videoconferencia.
Cómo se puede observar en la Fig. 3, el algoritmo señalado sólo tiene las propiedades deseadas para los *N* impares menores que 9 y los *N* pares menores que 6. Por esta razón se propone el fijar 7 como cota máxima de usuarios simultáneos, dejando el caso *N=6* como el único en el que hay un F_x con probabilidad menor de refresco.
Cualquier cambio para mejorar estos números implica un aumento en el número de iteraciones, que ralentizaría el envío.

N	Saltos	F_a
3	1 | 2	0 -> { 1} { 2 0}
	2 | 1	0 -> { 2} { 1 0}
4	1 | 2	0 -> { 1} { 2 3}
	2 | 1	0 -> { 2} { 1 3}
	-1 | -2	0 -> { 3} { 2 1}
5	1 | 2 | 4	0 -> { 1} { 2 3} { 4 0 1 2}
	2 | 4 | 1	0 -> { 2} { 4 1} { 1 3 0 2}
	4 | 1 | 2	0 -> { 4} { 1 0} { 2 1 3 2}
	-2 | -4 | -1	0 -> { 3} { 1 4} { 4 2 0 3}
6	1 | 2 | 4	0 -> { 1} { 2 3} { 4 5 0 1}
	2 | 4 | 1	0 -> { 2} { 4 0} { 1 3 5 1}
	4 | 1 | 2	0 -> { 4} { 1 5} { 2 0 3 1}
	-2 | -4 | -1	0 -> { 4} { 2 0} { 5 3 1 5}
	-1 | -2 | -4	0 -> { 5} { 4 3} { 2 1 0 5}
7	1 | 2 | 4	0 -> { 1} { 2 3} { 4 5 6 0}
	2 | 4 | 1	0 -> { 2} { 4 6} { 1 3 5 0}
	4 | 1 | 2	0 -> { 4} { 1 5} { 2 6 3 0}
	-2 | -4 | -1	0 -> { 5} { 3 1} { 6 4 2 0}
	-1 | -2 | -4	0 -> { 6} { 5 4} { 3 2 1 0}
	-4 | -1 | -2	0 -> { 3} { 6 2} { 5 1 4 0}
	1 | 2 | 4	0 -> { 1} { 2 3} { 4 5 6 0}
8	1 | 2 | 4	0 -> { 1} { 2 3} { 4 5 6 7}
	2 | 4 | 1	0 -> { 2} { 4 6} { 1 3 5 7}
	4 | 1 | 2	0 -> { 4} { 1 5} { 2 6 3 7}
	-2 | -4 | -1	0 -> { 6} { 4 2} { 7 5 3 1}
	-1 | -2 | -4	0 -> { 7} { 6 5} { 4 3 2 1}
	-4 | -1 | -2	0 -> { 4} { 7 3} { 6 2 5 1}
	1 | 2 | 4	0 -> { 1} { 2 3} { 4 5 6 7}
	2 | 4 | 1	0 -> { 2} { 4 6} { 1 3 5 7}
	4 | 1 | 2	0 -> { 4} { 1 5} { 2 6 3 7}

Fig. 3 $F_a(0)$ del algoritmo final

4. Resultados

Los resultados del algoritmo vienen condicionados por su aplicación. Al pretender servir como extensión para video en nuestra aplicación de VoIP (ver [4]), el envío de imágenes debe adaptarse a la frecuencia de muestreo y envío del audio, ya que la calidad de la señal de audio es más critica que la de video. Tomando los valores fijados para pruebas en nuestras implementaciones (ver [3] y [4]) podemos acotar los *fps* que se obtendrán en los casos mejor y peor.

Las ecuaciones (12) a (14) representan los tiempos de muestreo, de envío y una cota superior al tiempo total requerido para el envío de audio y video.

$$SampleTime = \frac{(Tam - kAddCif)}{BpsSamp} \quad (12)$$

$$VideoSendTime = I \cdot \left(\frac{Tam}{BpsTrans} \right) \quad (13)$$

$$TotalSendTime < 2 \cdot \left(I \cdot \left(\frac{Tam}{BpsTrans} \right) \right) \quad (14)$$

Es las ecuaciones de la (12) a la (14), consideramos lo siguiente:

- *I* es el número máximo de iteraciones.
- *Tam* es el tamaño del paquete de datos a enviar expresado en bytes.
- *kAddCif* es el número de bytes de control añadidos por el cifrador.
- *BpsTrans* es la velocidad de transmisión en bytes por segundo.
- *BpsSamp* es la frecuencia de muestreo en bytes por segundo.

En esta implementación, *kAddCif* esta fijado en 48 bytes, *BpsSamp* son 11025 bytes por segundo y *Tam* son 1400 bytes. Si consideramos una velocidad de conexión de 1,5 Mbps (un limite practico para la telefonía móvil) se obtiene una transferencia de 16 fps en el mejor caso y en el peor caso una transferencia de 1/6 de 16 fps, es decir 3 fps para *N=7*.

En la Fig. 4 se puede observar el uso medio teórico del ancho de banda, para cierto número de usuarios simultáneos, incluyendo voz y datos y manteniendo los *fps* mencionados anteriormente.

5. Conclusión

Nuestro esquema es ligero, escalable y puede ser implementado en dispositivos con recursos limitados como teléfonos móviles y PDAs. Proporciona excelentes resultados con la velocidad de conexión actual (y futura) de los dispositivos móviles, dando amplias posibilidades a diferentes tipos de aplicaciones incluida la videoconferencia segura (ver [6]).

Establece un límite de 7 usuarios máximos hablando simultáneamente, que es un límite adecuado, para las aplicaciones practicas del protocolo.

Además, el protocolo es P2P, lo que limita el daño causado por el fallo de un único nodo y es más resistente a ataques de denegación de servicios.

La inclusión de técnicas para la disminución del ancho de banda utilizado, se consideran como posibles líneas de investigación futuras.

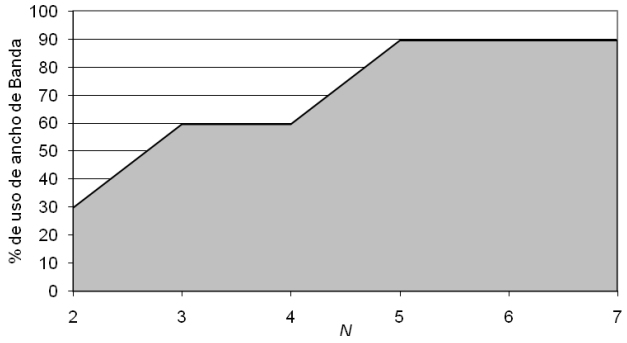

Fig. 4 Uso medio de ancho de banda

6. Referencias

[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, José Noguera, Leandro Tortosa And Antonio Zamora, "Secure VoIP and Instant Messaging on Small PDA Devices", *Transactions on Computers*, vol 5-1, 2005, pp 171-176.

[4] José-Vicente Aguirre, Rafael Álvarez, Leandro Tortosa And Antonio Zamora, "Secure Lightweight P2P Multiconferencing", *Transactions on Communications*, vol 6-1, 2007, pp 195-200.

[5] 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 - 306.

[6] Kuhn R., Walsh J., Fries S. "Security Considerations for Voice Over IP Systems", *Computer Security Division, Information Technology Laboratory*, 2004, National Institute of Standards and Technology.

[7] Schulzrinne H., Casner S., Frederick R., Jacobson V. "RTP: A Transport Protocol for Real-Time Applications", *RFC 3550*, 2003, IETF.

[8] Handley M., Schulzrinne H., Schooler E., Rosenberg J. "Session Initiation Protocol (SIP)", *The Internet Society*, 1999.