AccessGrid-to-Go : Providing AccessGrid access on Personal Digital Assistants

Gabriel Maajid[2], Michael Thorson[1], Jason Leigh[1] (spiff@evl.uic.edu),
Kyoung Park[1], Atul Nayak[1], Paul Salva[2], Shirley Berry[2]

1- Electronic Visualization Laboratory (EVL), University of Illinois at Chicago
2- Chicago Public Schools- Medill Technical and Professional Development Center.

(www.evl.uic.edu/cavern/continuum)

Vic Viewer for PocketPC (VVP) is a software application to decode and display a packetized video stream delivered over IP networks to a PDA running the Microsoft PocketPC operating system. This was first developed in the Spring of 2001 by Michael Thorson at EVL. The software is based on VIC, an open source video conferencing application originally developed at Lawrence Berkley National Laboratory. VIC is the primary means for video distribution over the Access Grid.

The fundamental problem we are attempting to solve is: How does one display dozens of Access Grid video streams on a small screen, over a low bandwidth wireless network, with a small amount of processing power? We believe this is the problem, in general, that needs to be solved for wide spread deployment of portable wireless video conferencing.

To solve the "small display problem" the VVP user interface provides either a cropped portrait view of a 320x200 VIC video stream, or a full portrait view. To view the image in portrait mode, one has to rotate one's PocketPC by 90 degrees. Furthermore, instead of trying to display all the Access Grid video thumbnails, VVP displays a list of video streams that a user can select from, to view.

To solve the "low available bandwidth problem" VVP employs a mediating proxy that filters all incoming multicast traffic from an Access Grid venue. The proxy compiles a list of detected video streams and forwards the list to VVP. The VVP user can then select the video stream that he/she would like to watch. This then causes the proxy to only send the selected video stream to VVP.

We tested two PocketPCs, the 133MHz HP Jornada 548, and the 206MHz Compaq iPaq 3765. While both systems appeared to have the processing power to handle video streaming and decoding, the limitations were found to be in the PocketPC's operating system (WinCE3). Firstly WinCE3 had only a tiny 4Kbyte UDP buffer. This meant that traffic bursts exceeding 500kbps would result in significant packet loss. To overcome this the proxy adds a 1ms delay between the transmission of each video packet to VVP, hence smoothing the overall video stream. Secondly, the WinCE3 Winsock API did not provide an asynchronous event signaling mechanism (typically available in Win32) for notifying an application that a datagram had arrived. To solve this problem, a separate thread had to be invoked to constantly monitor incoming traffic and to notify the display routines when video was available.

The first implementation of VVP supports a variety of packet-video formats, including: Xerox Parc Network Video (NV), Motion JPEG, ITU h.261, Sun Microsystems Cell B, and BVC. The following are the performance results of VVP using H.261 and MJPEG on the Jornada and iPaq.

Source Configuration	HP Jornada	Compaq iPaq
H.261 video Native frame size 352x288 VVP Proxy VIC source quality setting 10 (default)	Maximum Frame Rate: 9 fps Maximum Data Rate: 260 kbps	Maximum Frame Rate: 15 fps Maximum Data Rate: 500 kbps
Motion jpeg video Native frame size 320x240 VVP Proxy VIC source quality setting 30 (default)	Maximum Frame Rate: 8 fps Maximum Data Rate: 260 kbps	Maximum Frame Rate: 11 fps Maximum Data Rate: 400 kbps

The current unsolved problems include:

1. Extending the proxy to support multiple connected VVP clients.
2. Integrating unidirectional audio streaming with unidirectional video streaming.
3. Integrating bidirectional audio streaming with unidirectional video streaming.
4. Integrating bidirectional audio and video streaming.

It is not clear whether the 4th goal is obtainable with the current class of PDA devices, but we strongly believe this will be possible with the release of the upcoming 400MHz PDAs (due in Spring 2002). We anticipate that we will be able to solve problem 1. by March 2002- the results of which will be presented at the Access Grid retreat.

This work is supported in part by Microsoft Corporation, Hewlett Packard Corporation, and Medill Technical and Professional Development Center and the Illinois State Board of Education Learning Technology Center 7.