WiVision: A Wireless Video System for Real-Time Distribution and On-Demand Playback

Pradipta De Srikant Sharma Andrew Shuvalov Tzi-cker Chiueh
{prade, srikant, andrew, chiueh}@cs.sunysb.edu
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400, USA

***Abstract*— The ability to deliver digital video over wireless networks is an enabling technology for many useful applications, ranging from home entertainment and security monitoring, to enterprise messaging and military reconnaissance, and thus represents the holy grail of wireless technology development. In this paper, we describe a wireless video delivery system, WiVision, which uses IEEE 802.11 wireless LANs as the last mile for both real-time video distribution and on-demand video playback. WiVision can air both live events such as on-campus seminars and sports activities, and pre-stored video streams such as course lectures and financial analysis sessions, to mobile users, who can tune in to selected channels of their choice from their laptops or PDAs. An innovative feature of WiVision is the support for random video access based on keyword-search, where keywords are extracted from the closed-caption text embedded in TV programs. In addition, WiVision is able to broadcast video streams on the wireless link while seamlessly working with commercially available media players. This paper presents the implementation details of the real-time acquisition and network transport components of a fully operational WiVision prototype, and the results of a performance evaluation study on that prototype.**

keywords : Wi-Fi, TV-Centric Home Networks, Video, Real-Time, Media Player, Closed-Caption.

I. Introduction

The ability to access information at ones' convenience is a desired feature in any information disseminating system. Television is the most popular form of audio-visual information delivery system. However, TV programs have fixed schedules and viewers have to sit through commercials. TiVo [1] provides flexibility to the viewers to store TV programs and play them later at their convenience. In this paper, we build on the model presented by TiVo by designing a general Multimedia Distribution System where the end users can use their personal computers to watch programs aired through diverse media, such as, cable TV, close circuit televisions, VCRs, even handheld cameras. Such a setup can be very useful in scenarios, such as, broadcasting and recording class room lectures in university campuses, conferences, etc.

In recent times we have also witnessed an explosive growth of IEEE 802.11 based wireless LANs. Some of the hotspots for the wireless LANs are university campuses, hotels, airports, cafeterias, etc. Nowadays, most portable computers, including PDAs, come equipped with some type of 802.11 network interface cards. Such increasing deployment of Wi-Fi networks prompts us to reap the benefits of tetherless networking by converting the last mile of our video distribution system to wireless networks. With this wireless enhanced system in place any program can be accessed *whenever* and *wherever* a user wants it. With the ever increasing interest in pervasive computing, this is the sort of application that nicely fits into the paradigm.

In this paper we describe a real-time multimedia distribution system, *WiVision*, that can use 802.11b or Wi-Fi [2] LANs as their last mile connectivity solution. The primary objective of the WiVision project is to provide mobile users with Wi-Fi enabled terminal devices with an ability to watch live programs. Such a system is of great interest to knowledge workers such as stock brokers, who need to keep themselves abreast of the most recent business or political developments through 24-hour news channels such as CNN or CNBC. The WiVision technology can also be used to distribute digitally captured on-campus live events such as sports activities, seminars, special lectures, recital sessions, and other events of general interest in real time. Another potential use of such a system can be in surveillance networks where the closed circuit monitoring data can be streamed in real-time to the wireless handheld devices carried by security personnels.

In addition to the primary objective of distributing live programs, WiVision is also designed to provide the following features,

- *On-demand Playback* : It supports on-demand playback of pre-recorded video programs on end hosts. The pre-recorded programs can be acquired by digitizing cable TV signals and then storing them in central storage servers.
- *Text-based Navigation* : Programs enabled with closed-captions can provide navigation capability using the captions when played in playback mode. WiVision stores the closed-captions along with the video and displays the captions time-synchronized with the video. A user can scroll through the caption window to skip to the corresponding portion of the video clip.
- *Keyword Search* : In playback mode, a user can do a search for a keyword on the stored captions and choose to go to the video segment shown at that time instant.

The main challenge in designing such a system is to come up with a distributed architecture such that the acquisition of the multimedia content, the storage of this content, and later on, the distribution to the user, can all be done in a decoupled manner. Thus, the entire system is comprised of three separate components: (a) the acquisition component which digitizes the captured multimedia content into compressed streams, (b) the storage, management, and distribution component that stores the metadata and the content for future playback, as well as is designed to enable live streaming, and (c) the client software which is designed to receive the multimedia streams over wireless channels and display them. While designing the client component, we had to keep in mind that the bandwidth in wireless environment is a scarce resource. Because WiVision aims to provide a TV channel-like abstraction to end users, it broadcasts the video streams over the 802.11 networks so that mobile users can "tune in to" the appropriate streams whenever they want. However, broadcasting on wireless LAN is less reliable because the link-layer hardware does not support the ACK based retransmission mechanism as in unicast transmissions.

This paper is organized as follows; In section II we discuss the research relevant to the system we have built. In section III we describe the overall architecture of WiVision and its components. We also discuss various design decisions that were taken and the rationale behind them. In section IV we analyze the performance aspect of WiVision along with various issues that we encountered during the overall implementation. Finally, we summarize our work and discuss future enhancements that we feel are necessary for this system to be deployed effectively.

II. Related Work

Media content management and delivery is a very active research area both in industry and academia. The research is carried out on several fronts to explore different issues. The issues range from media encoding, to media distribution, media storage etc. In this section we try to compare and contrast some of the most prominent work with our system.

Berkeley Distributed Video-on-Demand [3] is one of the most comprehensive works about the storage aspect of large scale media contents. This system typically acts as a cache for media contents stored on tertiary storage devices. Andrew Swan and Lawrence Rowe [4] describe a system for video content creation, encoding and distribution over mbone multicast network. Ketan Mayer-Patel et al. [5] describe a media playback application which can access and play remotely stored media streams. All these things put together form a media distribution system over multicast networks.

Tivo [1] is a device which captures analog video signals, digitizes them and stores them on local disks for future playback. Tivo can be considered as a digital replacement of VCRs. Although Tivo does not form part of any distribution system, essentially the concept of encoding TV signals and storing them on disk is very similar to that of WiVision.

Dremedia [6] claims to have technology that combines speech recognition, image analysis, speech-to-text transcription, and the ability to organize unstructured data. It plans to use this technology for making all television content archived, indexed, and search-able. This is very similar to WiVision with an addition of search based on speech recognition and unstructured data.

The Stony Brook Video Server [7] focuses on real-time media retrieval from disks and then transferring the media streams to the ultimate clients on wired LANs with real-time guarantees. This work mainly focuses on storage and transport aspects of data.

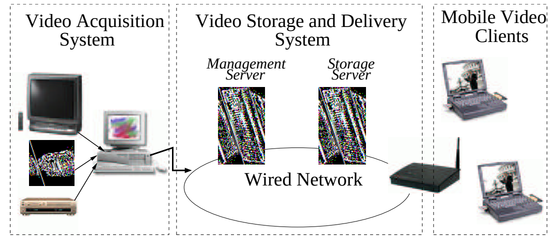

Fig. 1. *WiVision architecture. The Acquisition Server is responsible for digitizing and compressing the analog video signals in real-time. The compressed data with possible closed-caption text is sent to the central storage, management, and broadcasting servers over a TCP connection using any available physical network. The central system broadcasts this data to the interested mobile clients in real-time. The central system also supports on-demand playback along with indexing and search on closed-caption text in the video stream.*

III. System Architecture

WiVision is functionally divided into three major distributed components. These are,

- *Video Acquisition Server* responsible for converting the video signals into compressed digital streams in real-time.
- *Video Server* subsystem for storing and managing the compressed data on central servers for on-demand playback and transporting the data to the clients either in real-time or on-demand, and
- *Video Clients* on wired/wireless LANs which access the streaming data in real-time or on-demand.

The motivation behind this distributed approach stems from the advantages of possible mobility of the end components, which are, the acquisition server and the clients. Since, the source of video signal need not reside at a fixed location, the acquisition server may need to be mobile and the overall system reliability and availability expectations should not be heavily dependent on it. Considering this requirement, it becomes essential to decouple the management and long term storage aspect from the acquisition portion. The Mobile Clients are essentially dynamic components of the system which join and leave the WiVision system according to the requirements of the users handling them. This shifts the burden of overall system operation onto the central Video Server subsystem. The WiVision system architecture is shown in Figure 1.

A. Video Acquisition

The Video Acquisition Server digitizes and compresses the analog video signals in real-time and transports the encoded streams to the central storage and/or broadcast servers for further distribution. The analog video signals can be obtained from any video source such as Cable TV, for live news broadcast; a VCR, for screening of videotapes; a camcorder, for streaming live events like lectures or sports activities; or any other appropriate video source. The real-time digitization and compression is done with specialized hardware which accepts analog video signals. Further, if the video signals carry closed-caption text, these are

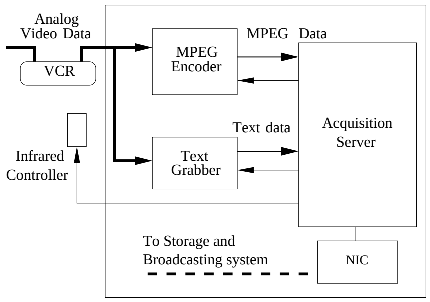

Fig. 2. *Acquisition Server Layout. The analog video signal is processed by the MPEG encoder and the text grabber device. The encoded MPEG data and the text is transported to the central video storage and broadcasting system over the network interface card using TCP. The tuning of a specific channel can be done by using an infrared device which controls the VCR used as a tuner.*

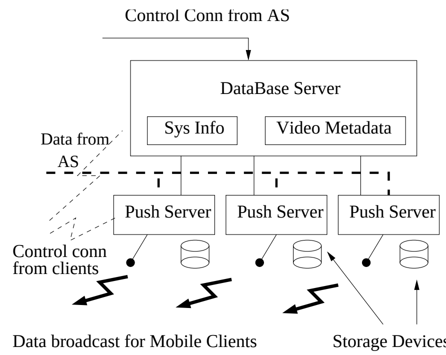

Fig. 3. *Video Server subsystem The data from the Acquisition Server is streamed to the Push Servers which broadcast this data to the interested mobile clients. The Metadata Server has a global view of the WiVision system and keeps track of the metadata corresponding to the video streams. The Acquisition Server and Clients maintain a TCP control connection with the Metadata Server. Privileged Clients can initiate recording of the video streams which is locally stored on the Push Servers.*

also required to be decoded and stored along with the digitized video streams for future indexing and referencing.

In our design, we used DarimVision *MPEGator* video encoders to convert the available analog composite video signals to MPEG-1 system layer streams. The video signal is also fed to a text-grabber device which is responsible for extracting the closed-caption text from the video signal. In order to facilitate the synchronization of closed-caption text with the video stream during playback at the clients, the closed-caption text is accompanied with timestamps. These timestamps are later used by clients to display the closed-caption text in synchronization with the video stream and provide a caption based video navigation. We use an infrared remote control device connected to the AS, and controlled by software, to tune in to the desired channels. The Acquisition Server transfers data to the Storage Server for possible playback in future. This data must be transferred reliably. Hence we have chosen TCP as the transport mechanism between the Acquisition Server and the Storage Servers. A layout of the Acquisition server and other connected components is shown in Figure 2.

B. Video Server

Video Server subsystem of WiVision is responsible for management, storage, and broadcast of the video streams and closed caption text. The Video Server receives the encoded data and closed-caption text from the Acquisition Server, manages and stores the data centrally, and distributes it to the interested clients in real-time. It also caters to on-demand playback requests from mobile clients for previously stored data.

The Video Server is functionally divided into two components , (1) a Metadata Server and (2) Push Servers. The Metadata Server is responsible for management of the metadata like, video clip information, server information, etc., about the entire WiVision distribution system. The Push Servers perform the task of storing and broadcasting stream data to the clients. A layout of the central Video Server is shown in Figure 3.

The Metadata Server manages WiVision using a relational database. The relational database holds the information about the prerecorded media streams, their locations, closed-caption text, and the entire system configuration. The prerecorded streams are stored on Push Servers in the form of movie files indexed by file ids.

Upon startup, each Acquisition Server connects to the Metadata Server and registers with it. The Metadata Server provides the Acquisition Server with a list of Push Servers to which it needs to stream the encoded data. This point onwards, the AS is treated as one of the *channels* in the distribution system. The Push Servers are responsible for transferring the video data to the mobile clients using appropriate transport protocols. The Push Servers also store the video locally for future on-demand playback. Push Servers use local disks for storing data. In future we plan to separate the storage component from streaming subsystem by making use of network attached storage such as *Phoenix* [8] which is designed specifically for video storage and proves QoS guarantees.

C. Broadcast

The streaming of real-time data to the Mobile Video Clients over wireless links is one of the most interesting issues in the implementation of WiVision. The Push Servers are responsible for broadcasting data over the last mile of the entire network. For performance reasons, Push Servers reside in the wired segment of the network. However, the real-time streaming of data is done over the wireless network. There are three options that can be considered for data transmission. (1) unicast transmission to each interested client, (2) multicast transmission to only interested clients, and (3) broadcast to all the clients and the clients can do further processing to filter out the required data. The data transmission can be carried out in an infrastructure setup of WLANs, which is essentially a single hop wireless network. Alternatively, the data can also be transmitted over a multi-hop wireless setup in order to increase the reachability and quality of transmission by increasing the density of repeaters.

The maximum bandwidth of Wi-Fi networks is 11 Mbps and the observed available bandwidth is merely 6 Mbps. The average bandwidth requirement of MPEG-1 system stream is around 1.5 Mbps. Going by these statistics, a maximum of four clients can be supported in any wireless network segment if unicast transmission was adopted. Further, these clients cannot have any QoS guarantees. This scenario is certainly not desirable.

A proper choice of data transmission would have been to use multicast addresses where multicast groups are established corresponding to each channel and mobile clients can join and leave these multicast groups according to the requirements. This would enable us to provide upto four different channels in any given wireless segment with no restriction on the maximum number of clients. But the main hindrance to this option is the lack of uniform and well documented multicast support by Wi-Fi hardware vendors. There is no clear and general mechanism that is made available by the NIC device driver to the OS to join or leave a multicast group. This forces us to resort to the last option of using broadcast addressing to reach the Mobile Clients. It is possible to send broadcast packets and still stream multiple channels in a wireless LAN segment. This can be done by sending UDP datagrams to broadcast IP address destined to specific ports. The destination ports distinguish one media channel from other. Using this technique one should be able to stream upto four channels in a wireless segment without any limitation on the number of clients.

According to 802.11b [9] specifications, broadcast packets are categorized under asynchronous data services. Asynchronous data services may experience lower quality of service compared to other packets. 802.11b specifications, in addition to ACK mechanism, also describe a fragmentation/defragmentation mechanism to increase reliability, by increasing the probability of successful transmission of the packet in cases where channel characteristics limit reception reliability for longer frames. But this ACK based reliability and fragmentation is not applicable to broadcast packets. Owing to these issues, practically the wireless segment is able to handle only upto two video channels in somewhat reliable manner. This is still better than using unicast addresses as the possible number of clients can still be very high without loading the wireless network.

In our current implementation Push Servers receive the data from acquisition servers over TCP connections. Since this data arrives from a remote location any periodicity introduced in transmissions by the Acquisition Servers may not be observed at the Push Server end. To avoid jitters and effects of network Push Servers introduce pacing in the broadcast traffic by transmitting the received data in a streamed fashion to avoid bursty traffic on the wireless LAN.

D. Video Clients

The Video Client applications run on end hosts. It is capable of playing streaming video in real-time and playback pre-recorded programs on-demand. We wanted to provide a TV-like abstraction to the user. Hence on startup, Clients fetch the list of Active Channels that are being streamed live and the list of the pre-recorded programs from the Metadata Server. For live streaming, selection of a particular channel implies listening on a port on which a Push Server is broadcasting the video. In case of playback, the client first retrieves the closed-captions corresponding to the video clip, and the video segment is unicast to each client. The closed captions are highlighted in a time synchronized manner along with the video clip. The closed captions are retrieved fully before starting the playback

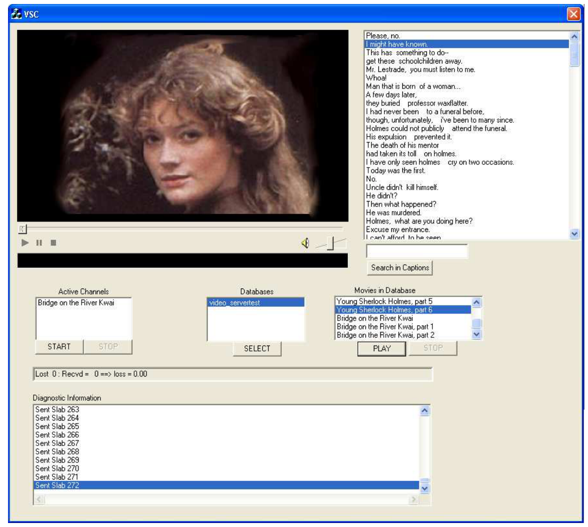

Fig. 4. *A screen-shot of client GUI. The users can navigate through a list of available channels and select the desired ones. e.g. in this screen shot the channels are displayed in active channel window. Closed-caption text, if available, is displayed in right hand text box, and one can search and navigate through this closed-caption text to reach to the desired contents.*

to provide a text-based navigation facility through the video clip for the user. One can jump to the portion of the video clip corresponding to a caption text by clicking on the caption. Searching for keywords from the captions is also possible. A screenshot of the client GUI, shown in Figure 4, displays these features.

The Video Client software is implemented on Windows platform and uses Windows Media Player (WMP) for displaying the MPEG streams. The choice of WMP is driven by the widespread use of the Windows platform for portable devices and Personal Computers at homes. However, WMP comes with its share of problems. First, it supports a limited set of protocols for streaming media. For playing streaming media on WMP, the video streaming server must either use Microsoft Media Server (MMS) protocol or it can be transfered from a webserver using HTTP. Whereas, Push Server is designed to use UDP for broadcast and unicast playback. Use of UDP is not supported by WMP. Second, WMP needs a specific header to be able to select the decoder for the MPEG stream. However, for real-time streaming this header is not present in the stream.

To circumvent the problem of limited streaming protocols and enable broadcast/unicast reception over UDP, we came up with a novel design for the client software. There are three logical components to the client software, (i) a receiver back-end which receives UDP data streamed by Push Server and puts it in a shared buffer, (ii) a lightweight proxy webserver which retrieves this data from the shared buffer and sends it to the WMP using HTTP, and (iii) the WMP embedded in the MFC-based client application which uses ActiveX control interfaces. The use of the proxy webserver solves the problem of streaming because WMP can interpret streams over HTTP connection. Figure 5 shows the components of the client software and the protocols that are used to communicate between the Push Server and the

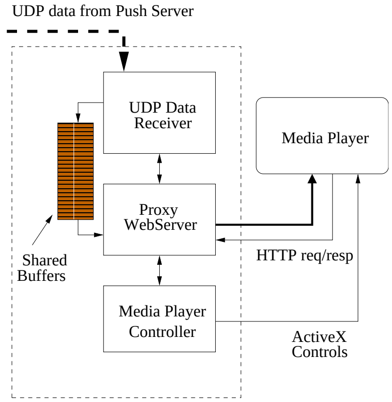

Fig. 5. *Mobile Video Client internal Architecture. The Mobile Video Client uses Windows Media Player to display the video streams. The Media Player Controller component of the client application interfaces with the Media Player through ActiveX controls. The client receives the UDP data from Push Server and places them in shared buffers. The Proxy WebServer component dispatches the data available in shared buffers to the Media Player Application which acts as an HTTP client controlled by the Media Player Controller.*

Receiver, and the Web Proxy and the Media Player.

The header required by the WMP for decoding and displaying the MPEG streams correctly is the initial MPEG pack header that is created at the beginning of the encoding process. The Push Server caches the pack header for each encoded stream. When a client requests live streaming or playback, it first fetches the header and sends it to the Media Player. This allows the WMP to decode and playback the video stream correctly.

Wireless environment is inherently lossy. It is possible to lose a few frames occasionally which may lead to a desynchronization in display of caption and video together. To resynchronize, we use the timestamps on the captions and the packets sent from the Push Server. In the event of losses we jump through the caption display corresponding to the lost video frames.

E. Home Entertainment Server

Although the system architecture of WiVision consists of multiple logical components, whether they are implemented on separate machines depends on specific application needs. For an enterprise-scale video delivery system, it is better to separate the video storage and management server from the acquisition server because multiple acquisition servers may be needed to capture different types of video streams, e.g., TV programs, live lectures, etc. However, for a home entertainment server, the video storage and management server and the acquisition server should be packaged into a single box. WiVision's ability to distribute video over wireless LAN is particularly compelling because (1) desktop machines are increasingly replaced by laptop computers and (2) almost all new laptop machines come with a built-in wireless LAN interface. As a result, a home entertainment server based on WiVision allows each household member to watch his/her favorite show from the desktop/laptop in his/her room.

It turns out that the TiVo machine may well be the best hardware platform for developing such home entertainment servers. A Tivo box can digitize and compress analog video into MPEG-2 video streams, and store them into disk storage for later playback, all in real time. In addition, TiVo boxes are

Loss Size (packets)	Slow-Changing Clip	Fast-Changing Clip
1-3	acceptable	minor perceptible glitches
4-6	minor glitches	noticeable glitches
7-10	noticeable glitches	video stalls on loss
10 +	unacceptable glitches	unacceptable pauses and clicks

TABLE I. *This table shows how a video clip will be perceived when a group of packets get lost, where the packet size used is 1472 bytes. A loss consisting of 6 packets lost in burst still gives an acceptable quality for both slow as well as fast changing clips. However, a loss of more than 10 packets will lead to a jittery video.*

relatively inexpensive compared with standard PC servers. Most importantly, it is known in the TiVo hackers community [10] that a TiVo box can be extended with an adapter that in turn connects to an Ethernet interface or an 802.11 interface. With this extensibility, one can run WiVision's video acquisition server and video storage and management server on a TiVo machine, and distribute the stored video streams through a wireless link in a near-real-time fashion. Because one can also add additional memory and disks to a TiVo machine, storage resource is in general a non-issue. Finally, the fact that Tivo also runs Linux makes porting of WiVision relatively effortless.

IV. Performance Evaluation

We analyzed the usability of wireless LANs as the last mile of video distribution in terms of perceived quality of video. We also evaluated our system to analyze the scalability in terms of supporting different number of channels. The impact of using broadcast address as a preferred addressing mechanism on the overall wireless LAN throughput was also analyzed.

A. Prototype Setup

The mobile clients we used are notebook computers with Intel Pentium class processors running MicroSoft Windows [XP/ME/98] OS with RAM varying from 64 MB to 256 MB. All mobile clients use Orinoco wireless cards as their NICs.

The Acquisition Server is implemented and installed on a Pentium-II 400MHz PC with 128MB RAM and Windows ME as the OS. The AS is connected to the central Metadata Server and Push Servers through a high speed campus-wide WAN. The input for MPEGator board comes from campus Cable TV system.

The Metadata Server is implemented on a Pentium-IV 1800 MHz machine with 256 MB of RAM with Linux Redhat 9.0 OS. The back-end relational database is Postgres and the front-end database application is implemented in JAVA and uses JDBC to interface with the database.

The system configuration of Push Servers is similar to Metadata Server. Push Servers are Pentium-IV 1600 MHz machines with 128 MB or RAM with Linux Redhat 9.0 OS. Push Servers use IDE disks to store the recordings of video streams.

Push Servers, Metadata Server, and wireless network access-points are connected with each other by a 100 Mbps Fast Ethernet switched network.

B. Quality of Video

To understand the usability of the WiVision system in a typical environment, we studied the playback and live streaming

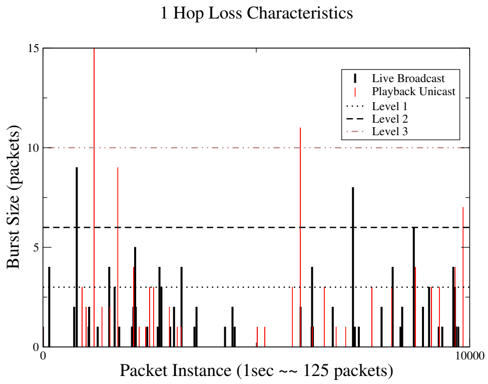

Fig. 6. *Characteristics of 1-Hop wireless channel when video is streamed live and on-demand using broadcast and unicast transfer mode respectively. It shows that most of the time, in both live streaming and playback, the number of packets lost in a burst stays below 10 packets, where packet size used is 1472 bytes. This implies that it is possible to get an acceptable video quality when the clients are in a wireless LAN setting.*

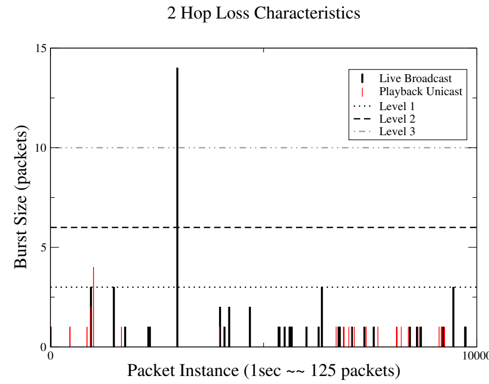

Fig. 7. *Characteristics of 2-Hop wireless channel when video is streamed live and on-demand using broadcast and unicast transfer mode respectively. The transmission characteristics in 2-Hop network are improved because of placement of additional repeaters.*

capabilities of the system on wired, as well as on 1-hop and 2-hop wireless LAN setup. In the wired case, there was no perceived loss of packet on client and both the live streaming and playback with captions were smooth. Wireless LAN is inherently prone to packet loss. In Table I, we show a relationship between the number of packets lost in burst to the perceived quality of video.

Figure 6 shows the packet loss characteristics of a 1-Hop wireless channel when we stream live data and on-demand playback. The distribution server was positioned in a room separated by two walls from the client. The difference between the two cases is in terms of using broadcast and unicast respectively. As shown in the figure, majority of the losses in broadcast are confined to a burst loss of less than 10 packets. Hence, live streaming despite having few glitches give acceptable video quality. In unicast there are a few large packet losses. However, mostly the losses stay within 10 packets and are comparatively fewer leading to a better perceived quality than that of broadcast. The different levels shown in Figure 6 correspond to different perceived qualities described in Table I.

Figure 7 shows similar characteristics for a 2-Hop channel. An additional repeater was positioned in between the client and the distribution server. The transmission characteristics were improved because of the additional repeater and the perceived quality of video was observed to be better than the 1-Hop case.

We also performed a scalability analysis of the number of possible channels over the IEEE 802.11b wireless LAN. It was possible to broadcast *two* channels with acceptable quality. The observed video quality deteriorated when the number of channels was increased to three. With four channels the observed quality was unacceptable. Thus upto 3 channels can be supported using a single channel of the wireless LAN.

V. Conclusion

In this paper we described and evaluated WiVision, a novel Real-Time Video program distribution system for wireless LANs. In addition to distributing Video programs, WiVision can be used to air various events of interest like classroom lectures, sports activities, etc. WiVision also supports on-demand playback of prerecorded events. WiVision supports a varying number of video channels but the scalability in channels comes at the expense of video quality. Going by our experience we suggest that IEEE 802.11b based wireless LANs can act as the *last mile* for Cable TV distribution if the quality of video streams is not of supreme importance.

At present, several choices for the WiVision design are constrained by the available hardware. For example, we chose MPEG-1 encoding for the video streams because MPEG-2 encoder cards are not readily available, and same for MPEG-4. Similarly, we used IEEE 802.11b for the wireless LAN. With the rapid emergence of the IEEE 802.11a, which supports 54 Mbps per channel, we can scale our system to multiple channels.

Another issue we have not addressed here is the quality degradation caused by the packet losses in the wireless channels. We have used UDP based broadcast for the transmission, but would like to shift to some reliable lightweight protocols for the final leg of data transport over wireless networks.

References

[1] [Online]. Available: http://www.tivo.com

[2] [Online]. Available: http://www.wi-fi.org

[3] L. Rowe, D. Berger, and J. Baldeschwieler, "The Berkeley Distributed Video-on-Demand System," in *Proceedings of the Sixth NEC Research Symposium*, 1996.

[4] D. Wu, A. Swan, and L. Rowe, "An Internet MBone Broadcast Management System," in *SPIE Multimedia Computing and Networking*. IEEE, January 1999.

[5] K. Patel, D. Simpson, D. Wu, and L. Rowe, "Synchronized Continuous Media Playback Through the World Wide Web," in *Demonstration at ACM Multimedia '96*, November 1996.

[6] [Online]. Available: http://www.dremedia.com

[7] M. Vernick, "The Design and Implementation of the Stony Brook Video Server," Ph.D. dissertation, SUNY at Stony Brook, December 1996.

[8] A. Raniwala, S. Sharma, A. Neogi, and T. Chiueh, "Implementation of a Fault-Tolerant Real-Time Network-Attached Storage Device," in *IEEE Symposium on Mass Storage Systems*. IEEE, 2000.

[9] IEEE, "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in *Institute of Electrical and Electronics Engineers*, November 1999.

[10] [Online]. Available: http://tivo.samba.org/