On Air: Evaluating streaming MPEG4 content across WLANs

Nicole Driscoll, Joshua Liberman, Jason Novinger, Robert Williamson

September 21, 2007

Acknowledgement

- University of Missouri Columbia Computer Science Department
- National Science Foundation
- Dr. Haibin Lu and Dr. Wenjun Zeng
- Yingnan Zu and Yiping Wu

Outline of Presentation

- Introduction
  - Video compression, wireless networks, and video streaming
- Methodologies
  - Network testbed, video tools
- Results
- Further Research

Introduction

Our project sought to evaluate whether current MPEG4 video compression schemes were adequate to allow video content to be streamed across wireless (WLANs) computer networks. To do this, it is important to understand:

- benefits of streaming video
- video compression techniques
- challenges associated with WLANs
- techniques used to evaluate video quality

Streaming video

Streaming video has many advantages over the download and watch method:

- Data streamed over UDP, avoiding the need for ACK messages and retransmitted packets
- Immediate or near-immediate start of video playback
- Ability to play, pause, skip forward and backward
- Availability of live broadcasts

Video sequences

Video sequences are essentially:

- sequences of still images displayed at a constant rate (e.g. 25 frames per second)
- these still images are literally a matrix of pixels of differing color and brightness
- each of pizel is represented by a set number of bits (often 8 bits per pixel)

Why compress?

- bandwidth constraints - dial-up data connections cannot accomodate the data rate necessary as can WLAN connections, which in turn cannot accomodate the data rate that wired Ethernet connections can
- there is existing redundant information in images that can be represented more efficiently
- humans perceive some components, like luminance, with more sensitivity than other components, like chrominance

Video compression

Compressing video sequences takes advantages of redundant information found in the sequence:

- spatial - redundant color and luminance information within one image (frame)
- temporal - redundant information between two consecutive frame in a video sequence

Spatial redundancy

Still 2-D images will have some pixels that are the same color and brightness. These images can be compressed by:

- breaking the image into 8 pixel x 8 pixel blocks
- using a DCT (discrete cosine transform) to identify highly important information

$$G_{u,v} = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{u,v}\cos[\frac{\pi}{8}(x+\frac{1}{2})u]\cos[\frac{\pi}{8}(y+\frac{1}{2})v]$$

Spatial redundancy

- reducing this numerical information by a quantizing coefficient

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & -4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- representing more common information with smaller binary "words" a Huffman code

Temporal redundancy

From frame to frame, there is often repeated data that can be transmitted just once to reduce bandwidth needs.

- frames are each either
  - I - intra frame - full image is compressed using spatial compression
  - P - predicted frame - only the differences from the previous frame are sent
  - B - bi-predicted frame - only the differences from the previous and successive frames are sent

Challenges of wireless networks

While WLANs are prized for their mobility, this comes at a cost:

- Unlike switched Ethernet, WLANs must share a broadcast medium, leading to collisions and retransmissions
- The fastest WLANs have inherently slower speeds than switched Ethernet connections

802.11b	11mbps
802.11g	54mbps
802.11n	300mbps
Ethernet	10/100/1000mbps

Challenges of wireless networks

- WLANs do not have fantastic range, with data throughput suffering as distance from the access point is increased
- Unlike shielded Ethernet cables, WLANs are susceptible to interference from common household items like microwaves and cordless telephones

Evaluating video quality

There are numerous, subjective and objective, ways to judge the quality of received video

- subjective
  - human eye
- objective
  - PSNR - Peak Signal to Noise Ratio
  - SSIM - Strucural SIMilarity

Objective video quality

- PSNR - ratio of the highest power of signal to the power of noise, expressed in dB, a score above 30.0 dB is considered high quality

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{\frac{1}{mn} \sum_{i=o}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2}$$

- SSIM - a decimal value from 0.0 to 1.0, 0.0 meaning no similarity and 1.0 being perfect

Methodologies

To evaluate video quality over wireless networks, we first had to build a network testbed using the technologies we were hoping to test.

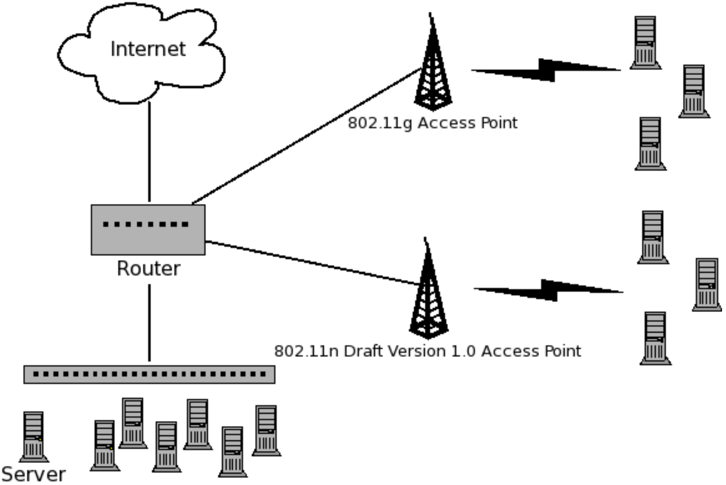
- wired and wireless local area network
- server computer running streaming server software
- client computers running streaming client software
- tools to encode video, compare received streams to originals, and hold it all together

Network hardware

The network hardware consisted of wired and wireless equipment.

- Wired Netgear router/firewall
- 24-port Cisco Ethernet switch
- Linksys WAP54G 802.11b and g access point
- Linksys WAP4400N 802.11n access point
- 13 assorted USB and PCI wireless adaptors
    - 4 PCI 802.11b and g adaptors
    - 6 USB 802.11b and g adaptors
    - 3 USB 802.11n adaptors

Network topology



Streaming server and clients

- Server
  - Typical Dell desktop PC
  - Pentium 4 CPU, 512MB RAM, 100mbps Ethernet
  - Fedora Core 6 Linux
  - Apple's Darwin Streaming Server
- Clients
  - Typical Dell desktop PCs
  - Pentium 4 CPU, 512MB RAM, 100mbps Ethernet
  - Fedora Core 6 and/or Ubuntu 7.04 Linux
  - openRTSP streaming media client from live555
  - 802.11b, g, or n wireless adaptor

Software packages

- MPEG4IP
  - mp4creator to create MPEG4 container files from elementary streams
  - mp4videoinfo to help identify lost frames
- ffmpeg
  - used to encode raw YUV videos to MPEG4 elementary streams
  - used to convert received MPEG4 stream to YUV for analysis

Software packages

- psnr tool from the evalvid package to calculate PSNR and SSIM scores for received MPEG4 streams
- Perl programming language to automate the test run process and analysis of received streams
- tcpdump to record all incoming (client) and outgoing (server) stream packets for later analysis

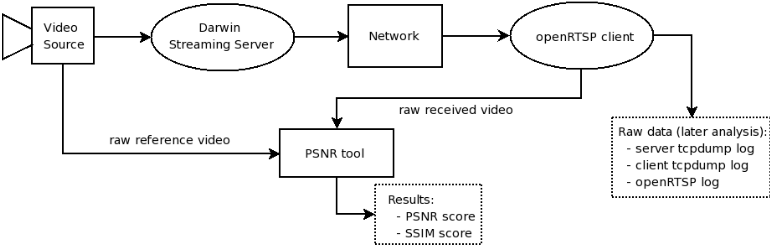
Missteps

- Videolan client (VLC)
  - Robert modified code to report statistics during test runs
  - VLC eventually proved to not be able to handle heavy streams well, prompted the switch to openRTSP
- Darwin Streaming Server
  - trouble compiling, needed to work out dependencies
  - firewall and SELinux were enabled by default on Fedora Core 6, not allowing any incoming connections

Missteps

- ffmpeg
  - existing binaries did not properly encode MP4 video
  - mp4creator could not find headers

Testing



- after the videos are streamed to the clients, they are then copied back and analyzed in comparison to the reference video

Assumptions

Our expected breaking points

Network	Max Speed	Ave. Speed	Breaking point
Ethernet	100mbps	90mbps	5-6 15mbps streams
802.11n	300mbps		20 15mbps streams?
802.11g	54mbps	27mbps	2 15mbps or 6 5mbps streams
802.11b	11mbps	5.5mbps	5 1024k or 1 5mbps stream

Let's find out what happened!

Feedback

A multitude of data

- server connected clients meter
- output from openRTSP
- tcpdump log files
- video itself
    - artifacts
    - PSNR score
    - SSIM score

Our first clue

- Percentage packet loss on the Darwin Streaming Server web interface suggested that there was significant packet losses at the beginning of all streams 5mbps and larger
- Most likely due to very large I & P frames near the beginning of the video
- On 5mbps video, issues quickly went away, lingered on 15mbps and higher

Initial results

- our initial observations tend to support our earlier assumptions that video quality is severely broken when total video throughput nears the maximum capacity of the network link
- all types of networks had issues with streams that 5mbps or larger, with early frames being dropped
- both wireless and wired networks had problems with streams 15mbps and larger, often dropping all I frames

Detailed results

- data runs were completed earlier this week
- PSNR and SSIM calculations for all received video were completed Wednesday evening
- unfortunately, not enough time to have a good view of what happened
- plan on continuuing work on data analysis during the following semester
- detailed results will be posted on the website

Further Research

- MPEG4 Part 10 (h.264) encoding
  - higher quality at a lower bitrate
- more intensive testing of 802.11n wireless
  - do wireless issues outweigh potential throughput gains
- tweaking wireless network settings for streaming video

Authors

Nicole Driscoll is a senior majoring in Computer Science at Southeast Missouri State University in Cape Girardeau, MO.

Joshua Liberman is a senior majoring in Computer Science and Mathematics at the University of Central Missouri in Warrensburg, MO.

Jason Novinger is a senior majoring in Computer Science at Truman State University in Kirksville, MO.

Robert Williamson is a senior majoring in Computer Science and Mathematics at the University of Missouri-Columbia in Columbia, MO.