Streaming MPEG4 video over wired and wireless local area networks

Nicole Driscoll
Southeast Missouri State University
nrdriscoll1s@semo.edu

Joshua Liberman
University of Central Missouri
jsl02020@ucmo.edu

Jason Novinger
Truman State University
jnovinger@truman.edu

Robert Williamson
University of Missouri - Columbia
rdwvy3@mizzou.edu

Abstract

There is not a clear consensus on how open-standard video streaming technologies perform across wireless computer networks. Wireless networking technologies have become nearly ubiquitous, particularly in residential networks, leading consumers to expect that they will retain the performance of wired networks. Fortunately, the advances in video compression and wireless network bandwidth allow for higher-quality streaming video content over the more limited wireless network. We seek to evaluate how video, encoded using an implementation of the MPEG4 Part 2 codec, performs when streamed across a simulated residential wired and wireless computer network. In particular, we are interested in the quality of the received video and how the transmission across the network affects the subjective and objective appearance on the client computer. Our network test bed comprises nine average desktop computers equipped with a stream retrieval program, OpenRTSP, and a server running the Darwin Streaming Server from Apple Computer, connected using wired Ethernet connections and 802.11b, 802.11g, and draft 802.11n version 1.0 wireless connections. Each client was monitored while receiving a sample of raw video data encoded at one of a variety of common bit-rates to note any lost content. In addition, each client saved a copy of the video locally for later comparison with the original using the PSNR (Peak Signal to Noise Ratio) and SSIM (Structural SIMilarity) metrics. Looking strictly at established wireless standards (802.11b and g), we found that they are not capable of streaming multiple High Definition quality video streams across a wireless network link. Wired and draft 802.11n wireless connections did prove more capable of handling multiple High Definition video streams concurrently. Hopefully, our work will lead to a better understanding of the technical issues, performance, and trade-offs in home networking, thus facilitating the rapid deployment of advanced home networking services and applications.

Keywords

Video compression, wireless network technologies, video streaming, home networks.

I. Introduction

Consumers are becoming aware of the capabilities that are available to be able to stream video across networks. Web browsers originally had to download the entire file before they were able to play it back. Some of the awareness is because of You-Tube and other popular media-sharing websites. There are several advantages and disadvantages to streaming video instead of downloading. Aspects crucial to video-streaming are video compression, hinting, protocol, physical limitations and bit-rate.

We performed this research to be able to see what level of video each of the various wireless technologies could handle. We tested 802.11a, 802.11b, 802.11g, and Draft 802.11n Version 1.0. We were streaming video at various bit-rates from HDTV to VHS quality across the wireless networks and were comparing the results to our control, wired network. We ran quality analysis to judge the distortion of the videos.

Streaming video has several advantages over a download-and-watch player such as smaller memory use and near instantaneous launching, but there are also valid reasons to download a video rather then receive a stream. The ability to view multiple times without retransmission is one, as well as capturing the video quality without any packet loss, watching it as it was intended.

In order to stream a video, that video must be compressed, as raw video data is too large to transmit over current networks. The basic techniques of video compression are similar to those of image compression, with just a third dimension of locality. Each successive frame stores the changes from the previous frame. There are two different compression schemes used, known as scalable and non-scalable video compression.[2] Non-scalable video compression uses one compressed stream at one

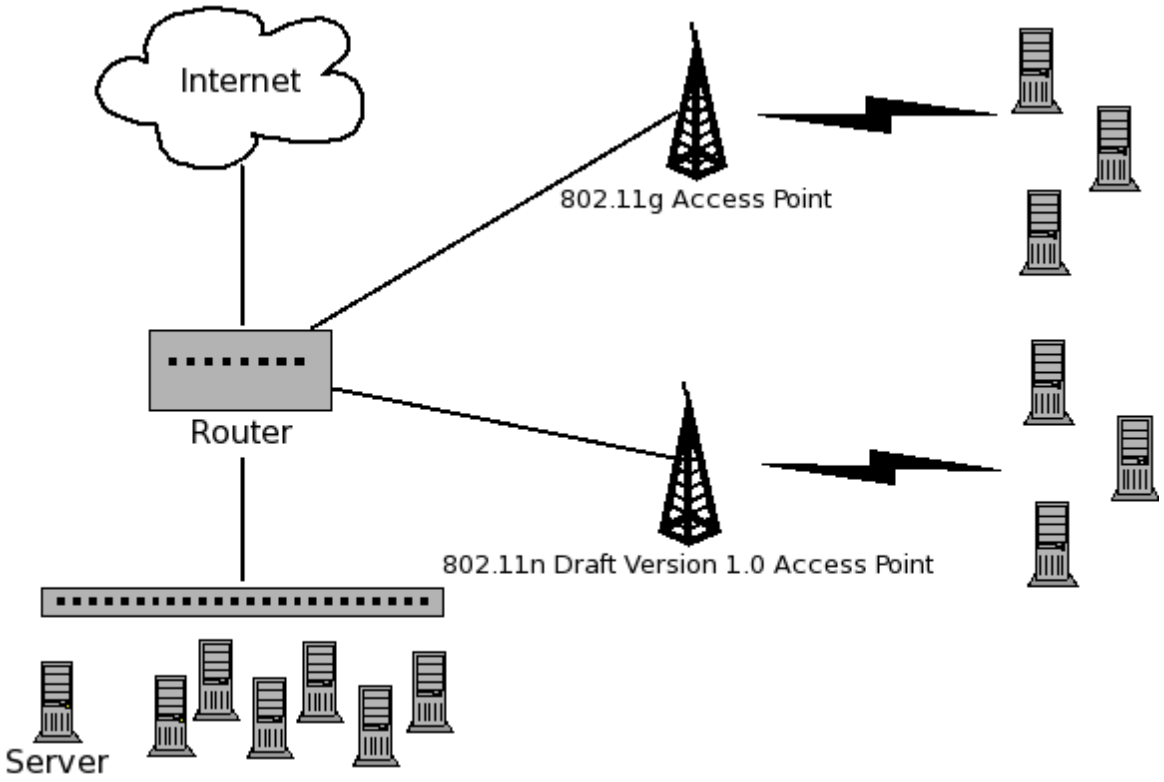

Fig. 1. A diagram of our network testbed, with wire Ethernet and wireless 802.11b, g, and draft n connections.

specific bit-rate. Scalable video compression is very flexible and can change the amount of bandwidth required to successfully stream depending on the current network conditions using multiple streams. We have used non-scalable compression on our several MPEG4 files.

Another thing that is needed to stream the video is hinting. Hinting provides the server with the needed media information, so that it is aware of what tracks are coming. There are two different types of hinting: content hinting and application hinting[2]. By using content hinting, we allow the server to know how to form the packet streams.

After hinting a file, the next important topic is the protocols used to stream multimedia. Real-time Transport Protocol(RTP) is an application layer protocol for transporting audio/video packets in over UDP. Control information is handled out-of-band by a protocol known as Real-time Transport Control Protocol. RTCP passes Quality of Service(QoS) to the server, including information like percentage of packet loss, delay, jitter, and out-of-order delivery which allows the server to adjust to changing network conditions. For this reason, streaming servers often encode videos at a variety of bit-rates so that they can adapt if network congestion is high. Clients can issue VCR-like commands to the server using the Real-time Transport Streaming Protocol(RTSP).

RTSP is an application layer protocol. It uses TCP for the transport of metadata. The basic commands for RTSP are Describe, Setup, Play, Pause, Record, and Teardown. A Describe command sent to a server includes a URL and the type of data that the client can handle. The Setup command is used to specify the port that the client will receive the data on. A Setup command is needed for each media stream, such as one for video and another for audio. The Play command will play all streams that have been initialized by the Setup command concurrently. The Teardown command ends all media streams and clears all of the clients data on the server.

If several clients connect and request high-quality media at the same time, the server can easily encounter network congestion. The server may connect to each client individually, if the server is

using Unicast, even if several clients are requesting the same file. A more intelligent server will have implemented Multicast, a technology where a single signal is sent from the server and is duplicated by routers only when required to reach its destinations.

Along with the technology considerations there are also a few physical limitations that must be taken into account when streaming media across the various types of networks. The main physical limitations to consider are the bandwidth of the network, the length and resolution of the video, and the bit-rate the video was encoded at. Bandwidth is the amount of data that can be transmitted between two given points, but includes all header information and checksums. A more accurate description of network transmission capabilities would be throughput. Throughput accounts for all links between the sender and receiver, therefore throughput is the measure of received data without the intermediate headers information. Throughput still contains some packet header information, so the measure of received data that is application-usable is called good throughput or throughput.

The size of the video depends on the length and bit-rate. Were a client to request a 30 second trailer of a high-definition video, the server would send a total of:

$$Size = \frac{30.0s \cdot 15000.0kb/s}{8,388.608kb/MiB} = 53.6MiB \qquad (1)$$

The bit-rate of a video is the amount of data per unit of time, usually a second. Bit-rates are variable and a higher rate will correspond to higher quality video assuming the same codec is used. In order for streaming video to be successful the throughput must be greater than the bit-rate. However throughput will vary depending upon network usage and bit-rate can vary depending upon the current scene. Therefore at any given moment it is possible that the bit-rate will exceed the throughput. To counter this problem, video players will buffer incoming data. This slight delay in playback ensures that if data transmission is cut off for a few microseconds or the bit-rate exceeds throughput that the user will be able to continue watching the video without interruption.

The basic principles of video encoding are designed to reduce the bit-rate without sacrificing video quality. Before this is done each frame is broken into YCbCr color space. The Y stands for luminance or brightness. The Cb and Cr are the blue and red chrominance levels, respectively. The human eye is more sensitive to luminance than to chrominance, therefore it is optimal to dedicate more bits towards brightness than color. The most common ratio used in video compression is 4:2:0. This ratio is luminance to chrominance to the ratio of the blue to red chroma. This means that the most common ratio has twice as much luminance than chrominance and an equal amount of blue and red chroma. When encoding video, each of the color spaces is handled individually.

To conserve bit rate, most video codec only encode the differences between frames rather than each entire frame. There are different types of frames such as I, P, and B frames. I-frames are intra-frames or key frames. These frames contain all the needed data to display the picture without needing knowledge of previous frames. They are often used at scene changes. These frames take up the most space and are crucial to display a video. P-frames are predicted frames which may require a significant amount of knowledge of the frame or frames directly preceding it self. They require fewer bits than I-frames. B-frames are bi-predicting frames which are similar to P-frames but may require knowledge of any frames that came before it not just the ones directly preceding it self. These frames also take up very little space since they depend so heavily upon previous frames.

Once the video has been encoded, it needs to be placed into a container. Containers allow multiple streams to be contained in one file. This is a key step as the file needs to contain hinting information that tells the streaming server how to parcel out chunks of video. MPEG4 containers can be used to include multiple audio stream (e.g. in differing languages), hint tracks, and other video streams.

The two lowest layers of networking are the physical layer and the data link layer. For wired Ethernet, the physical layer is known as the Ethernet Physical Layer and the data link layer is known as Ethernet. The Ethernet Physical Layer can vary from coaxial cable to fiber optic cable. It can

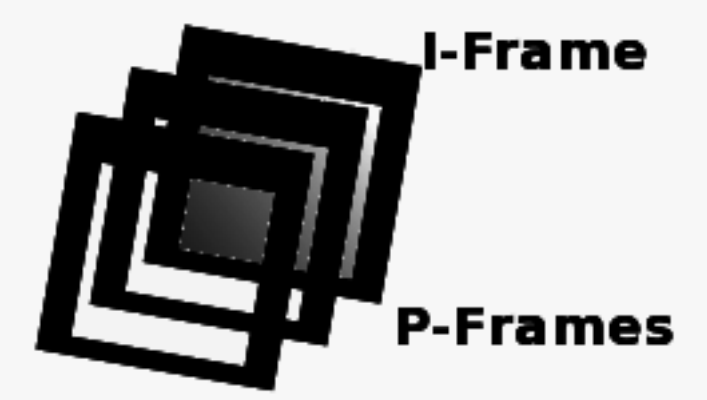

Fig. 2. A illustration of I (Intra) frame and subsequent P (Predicted) frames in an MPEG sequence.

also vary in speed, from 3Mbps to 10Gbps. Ethernet, the data link layer, consists of sending small amounts of data also known as packets. Each Ethernet station has a 48 bit MAC address. Ethernet does not have a one-to-one connection from sender to receiver. Traffic in current generation switches is routed to the selected receiver.

For wireless networks the physical layer is know as Wi-Fi and the data link layer is 802.11. There are four common protocols for 802.11, they are 802.11a, 802.11b, 802.11g, and 802.11n. Currently 802.11n can operate in the 2.4GHz and 5GHz frequency but is still in the draft stages but products based on various draft revisions are available. The 802.11b and 802.11g standards operate within the 2.4GHz, 802.11b was created first with a speed of 11Mbps and a revision known as 802.11g gave it a speed boost to 54Mbps. Most 802.11g devices are backwards compatible with 802.11b devices but have to fall back to the lower speeds. The 802.11a standard was created around the same time as 802.11b, its difference is that it works in the less crowded 5GHz frequency which allows for higher transfer rates of 54Mbps though with reduced range.

Wireless networks can operate in two different types of modes. One type deals with a central access point and the other is ad-hoc, a network formed from peer-to-peer. Wireless networks are being incorporated more and more in the business world, individual homes and other venues such as coffee shops or restaurants, college campuses, and other high-traffic areas. Today's population also has an increasing number of devices that can connect to wireless networks. These devices include but are not limited to laptops, PDAs/smart phones, portable gaming devices and portable music players.

There are many advantages and disadvantages of wireless networks. Some advantages include connectivity, cost, mobility, and convenience. Connectivity allows people to stay connected to the internet no matter where they are located depending on the location of the access point (AP). Wireless networks can allow multiple clients to be connected through a single access point. APs may be a little more expensive than wired hardware such as switches but their initial setup costs only requires a single point and they can scale without the need to buy and run additional wires. Clients are free to move within the given area of an access point or even from access point to access point. Access points can also allow for quick deployment or mobility of a network. All these factors combined increase the convenience of a client using a wireless network over wired networks.

Unfortunately wireless networks also suffer from a number of disadvantages such as security, range, reliability and speed. These disadvantages may affect the user more depending on the nature of their work. While streaming video, the security issue does not hinder our success as it would while transmitting personal information such as credit card usage, but an attacker may still attempt a denial-of-service attack by flooding the wireless medium, which would hinder legitimate packet transmissions. Range and obstructions also affect the signal strength which in turn will affect the reliability and speed. Wireless networks are subject to interference from several different types of sources,

like microwaves and cell phones, therefore performance is not guaranteed. Currently most wireless networks operate using 802.11b which has a maximum theoretical bandwidth of 11Mbps, but there is a realistic bandwidth of about 5Mbps that must be shared.

There are many possible ways to judge the quality of streamed video and they can be divided into two main categories: subjective and objective. A mean opinion score is the combination of subjective and objective video quality. The subjective video quality judgment is the way that the video appears to the human eye whereas objective video quality judgment often measures differences in the respective files. The most common kinds of objective judgment are Peak-Signal-to-Noise Ratio (PSNR) and Structural SIMularity (SSIM)[1]. PSNR captures only the raw difference between two frames and is not the most reliable form of comparison:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2 \quad (2)$$

$$PSNR = 10 \cdot \log_{10} \frac{MAX_1^2}{MSE} \quad (3)$$

The SSIM is a full-reference statistic. It compares the the original, pre-compression image to the received uncompressed image. The Structural Similarity metric reports a value from 0 to 1, with 1 representing a perfect correlation between compared images[1].

II. Methodology

VideoLAN Client (VLC) is an open source media player written in the C programming language. It was initially selected because it can be used on a number of operating systems, supported various formats of streaming video, is open source, can be started playing a video stream from the command line, and had built in statistics. The built in statistics combined with the project being open source were the two main reasons for using the software. Because statistics such as played and lost frames were already built in modifying the software to write these values out to a file with a time stamp was a simple task. When initially tested on local files it became clear that the built in statistic functions were not too reliable. In respect to bit-rate the values for the first couple seconds were always zero with the next two being too high. While these values if averaged over time were correct they were also somewhat worrisome. The real problems with VLC started appearing when using a streaming video source. The number of dropped frames and displayed frames would not add up to the total. In addition VLC seemed sluggish and slightly unresponsive when playing back streaming video.

One particular problem with VLC was its small buffer size when receiving video, causing an overflow. Whenever VLCs buffer became full, the simple response was to double the buffer size for subsequent frames. After doubling the buffer size about two or three times, VLC would finally have enough buffer to store the incoming video we were sending, but all of the data previous to this point was already lost which meant dropped frames. One early solution to this involved inserting additional blank frames to the server videos with the *cat* or concatenation command, which can append the data of one file to the end of another. In this way we were dropping these header frames without losing any actual information. While this appeared to be an ideal solution for our buffer problem, a interesting bug occurred where the comparison tools could not read the 1080p video files that had been concatenated. With these problems and no clear solution other clients were considered. Looking at other clients was considered a better use of time than trying to fix VLC. Overall the use of VLC just enforced the fact that we would need a client that could be launched from the command line.

VideoLAN Clients RTSP protocols were implemented by Live555 developers. This group had also released another program called OpenRTSP which incorporated a streamlined RTSP client that would run from the command line. This program has a lot of useful abilities and formed the core of our program list. While it could only save the video for display later, the lack of display in OpenRTSP allowed it to easily run on command line through SSH sessions for automating the stream retrieval scripts. Useful Quality of Service (QoS) statistics were captured using the Q switch and the buffer size was made ample with the b parameter. A buffer that is too small would

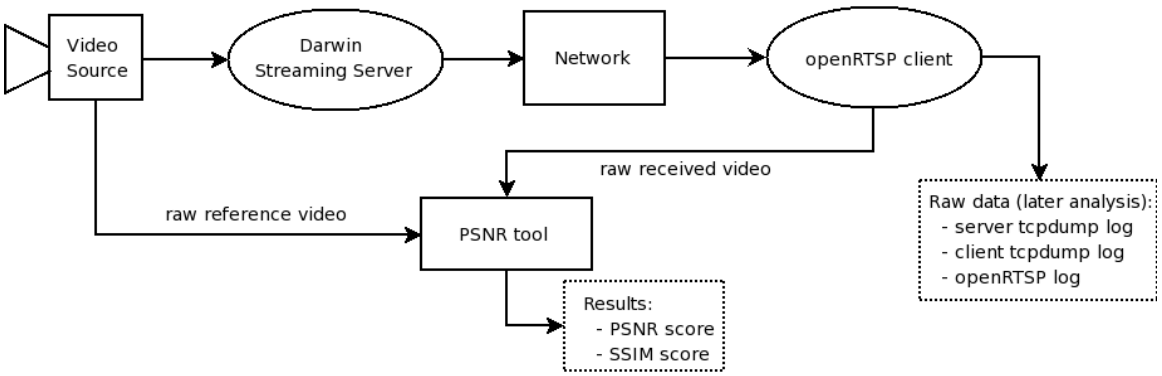

Fig. 3. A flowchart representing how video is streamed, captured, and analyzed in our network testbed.

incorporate some data loss. An example of the openRTSP invocation that we used was:

We used several scripts and a variety of Linux tools to automate the process of retrieving streams from the server and then assembling them for comparison. A Linux command *tcpdump* will output the traffic through a specified port, and comparing server to client traffic is one way to discern when packets are lost or delayed.

The server was also the master computer when it came to issuing the *ssh* or secure shell commands. The client computers were given authorization keys so that we did not need to input root passwords. This seemed to be a good example of the Master-Slave relationship type. After securing a connection, the client computers were made to request streams from the server and *scp* or secure copy them to another computer with the video comparison tools installed.

Our project aims to test the ability of different local area networking technologies, particularly wireless networks, to carry streaming video to multiple clients. We have chosen to test the streaming ability of MP4 encoded videos over these networks. In order to do this, we need four things:

1. Video (and possibly audio) content encoded in MP4 format
2. A streaming server
3. Clients to tune into and play the stream published by the server
4. A network connecting the server and client machines

In our tests, we chose to use Darwin Streaming Server, the open-source sibling of Apple Inc.'s QuickTime Streaming Server product that is included with MacOS X server. Darwin Streaming Server is capable of streaming QuickTime and MP4 encoded video files, as well as MP3 audio streams. We chose to run Darwin Streaming Server on an installation of Fedora Core 6 Linux on Intel x86 hardware. Each of our computers were equipped with Pentium 4 CPUs and 512MB RAM. Some clients had Ubuntu 7.04 Linux installed.

- 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

III. 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. Each type of network had issues with streams of 5mbps or larger, with early frames being dropped.

Both wireless and wired networks had problems with streams greater than 15mbps, often dropping all I-frames.

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


Fig. 4. On the left, a captured frame from the reference video, without any distortions. On the right, a captured frame from a corresponding received frame, shown with distortion caused by network packet loss.

IV. Conclusion

Our data seemed to confirm our hypothesis that the 802.11b & g wireless specifications were a less capable medium for streaming video content over a local area network than the wired control infrastructure. Similarly our data validated our hypothesis that draft 802.11n version 1.0 wireless specification could surpass the performance of the wired control infrastructure. Our limited number of 802.11n adapters and the little time

Video quality degrades when data packets are lost in transit. Ideally, this would not be a problem as video is often displayed at around 25 frames/second, where losing a few would be barely noticeable. Our analysis indicated that with MPEG4 Part 2 encoding, losing a single frame could have a harsh effect on the quality of the received video. This was due to the way current video encoding and compression exploits motion.

Each frame in a video can be described in terms of the difference from the previous frame. In practice, about every 12th frame is a key frame, encoding all of the information needed to display that frame, and most other frames are P-frames that encode information relative to a previous frame.

While this is effective compression, a congested network exasperates the problem usually by dropping the larger key frames first. Every time a key frame is dropped, the quality of the subsequent 11 frames degrades significantly. With higher network usage and congestion, we noted that I-frames dropped with higher probability. This conclusion suggests that a video format that encodes with a uniform frame size would have higher quality in the scenario where not all frames were transmitted.

A simple example would involve introducing some redundancy to the P-frames. Imagine cutting a frame into, say 12 blocks of equal size. Now encoding all the I-frame information into one of these blocks would not drastically increase the size of the frame, and yet within half of a second the equivalent of an I-frame will have been passed to the client. In this way, a video can be encoded without the traditional I-frames and still have corrections of the errors in its video quality.

V. Further Research

There are several derivative paths an extension on this lab could take. Our research utilized the part 2 specification of Mpeg4-Ip implemented by the *ffmpeg* tool. A similar approach as was suggested in our conclusion is being put to use by MPEG4-IP part 10. A future lab could run a comparison of the two encodings and of other video encodings streaming over a similar test bed.

Our research utilized video-only MP4 files, because audio is more difficult to judge in quality. A future lab could focus on audio encoding and streaming over a similar test bed with the goals to discern how much bitrate an audio file commonly needs to have encoded for successful and high-quality playback.

While we did get the draft 802.11n wireless network to successfully transmit, we didn't have enough clients for a legitimate stress test of the network.

One aspect we would have liked to dedicate further research to would be manipulation of our net-

work's interior structure. One of our idea's included testing how differing packet sizes affected the network load and thus the packet loss. Packets that are too small would require several transmissions and incur a lot of overhead whereas packets that are too large would unnecessarily clog the link.

There are many interesting facets for research that deal with the streaming server.

VI. Glossary

Access point (AP) - a device that connects wireless communication devices together to form a wireless network

Bandwidth - the amount of data that can be transmitted between two given points

Bit-rate - the amount of data received per second

B-frames - bi-predicting frames which are similar to P-frames but may require knowledge of any frames that came before it not just the ones directly preceding it

Chrominance the part of an image signal related to its color

Encoding - the process of transforming information from one format into another

Hinting - provides the server with the needed information

I-frames- intra-frames or key frames that contain all the needed data to display the picture, often used at scene changes

Jitter - unwanted variation of one or more signal characteristics

Latency - the time that it takes a packet of data to be sent from the sender to the receiver

Luminance - image brightness

P-frames - predicted frames which may require a significant amount of knowledge of the frame or frames directly preceding it

Peak-Signal-to-Noise ratio (PSNR) - a weighted ratio used to describe the difference between a reference image and a distorted copy

Quality of Service (QoS) - used to determine the order of packets that are forwarded based on the priority specified

User Datagram Protocol (UDP) - part of the transport layer, allows computers to send short messages, may be out of order

Real-time Protocol (RTP)- a standardized packet format for delivering audio and video over the Internet

Real-time Streaming Protocol (RTSP) - an application layer protocol that defines use of UDP streaming for the transport of data

Structural Similarity (SSIM) - a metric comparing the resemblance between a reference and a received or distorted copy

Transmission Control Protocol (TCP)- a transport layer protocol that attempts reliability and organized delivery of data

Throughput - records all links between the sender and receiver and is equal to the lowest bandwidth in the path

Video compression- a data encoding that reduces the amount of data stored in video, commonly by exploiting redundancies or removing imperceptible details.

References

[1] Zhou Wang, Conrad Bovik, Hamid Rahim Seikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 2004.

[2] D. Wu, YT Hou, W. Zhu, Y.Q. Zhang, and JM Peha. Streaming video over the Internet: approaches and directions. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(3):282–300, 2001.