2012

EXPERIMENTS ON VIDEO STREAMING OVER COMPUTER NETWORKS

Steven Becker
University of Nebraska-Lincoln

EXPERIMENTS ON VIDEO STREAMING OVER COMPUTER NETWORKS

by

Steven Becker

A PROJECT REPORT

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Dr. Byrav Ramamurthy

Lincoln, Nebraska

December, 2012

Department of Computer Science & Engineering, University of Nebraska-Lincoln,
Technical Report, TR-UNL-CSE-2012-0013

EXPERIMENTS ON VIDEO STREAMING OVER COMPUTER NETWORKS

Steven Becker, M.S.

University of Nebraska, 2012

Adviser: Dr. Byrav Ramamurthy

Video traffic (including streaming video service) is dominating the Internet traffic today. Video can be streamed using a dedicated server, a content delivery network (CDN), or peer-to-peer (P2P) overlays across a network. Video can be transmitted in multiple formats and at different resolutions. Video is also being distributed to a variety of devices (fixed and mobile).

In this project, we investigate the evolution of streaming video standards over time and the corresponding effects on the Internet that have occurred as a direct result of increasing video traffic. We also examine several options for transmitting video over computer networks. In particular, we focus on streaming video from a server to a client under multiple scenarios including (1) a local network testbed, (2) a home network, (3) across Internet2 between the University of Nebraska and Rutgers University, and (4) between nodes on the global PlanetLab overlay testbed. For these scenarios, we evaluate the user video Quality of Experience (QoE) using both subjective and objective criteria. Our experiments also investigate the performance of different transcoding standards for video such as H.264, Theora, Dirac, MPEG-2, DIV3, and WMV. Compared to transmission over IP networks, the performance across a switched layer-2 path (Ethernet VLAN) is found to be superior and feasible with today's technological advances, even across the Internet2 backbone.

ACKNOWLEDGMENTS

Several individuals have been instrumental in the process towards producing this report. Foremost, my advisor Dr. Byrav Ramamurthy has expressed continual encouragement along this journey. His guidance from the perspective of professor and background knowledge in the industry has provided focus and sustained leadership in this goal of Master's project research. Similarly, Dr. Leen-Kiat Soh and Dr. Witty Srisa-an deserve appreciation for participating on the project committee.

Mr. Kent Christensen (IS, UNL) deserves recognition for his continual desire to assist in challenges that impact campus and regional networks. His expertise within Information Services has provided the luxury of extending connectivity beyond the Computer Science laboratory through Internet2 and including other educational facilities such as Rutgers University. Special thanks goes to Ivan Seskar at Rutgers University for dealing with specialized networking requests. The experiments that span multiple institutions would not be possible without his willingness to work with very customized configuration requests. Additionally, PJ Clayton (MOREnet) deserves commendation for efforts to complete the network reservations across Internet2.

We thank the US National Science Foundation (NSF) for partial support of this project through an NSF FIA grant (CNS-1040765). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In terms of bandwidth utilization, video represents the single most intensive type of traffic on the Internet [32]. Intensive gaming may require short bursts of significant bandwidth but even that pales in comparison to the general populace of Internet users and the sheer amount of video they request. Web pages themselves have an overwhelming amount of requests, but videos are on a exponential scale larger per request and dominate bandwidth by number of bits transferred [32]. A single user could surf endlessly for days without meeting the same bandwidth requirements of a single full-length movie. The lines blur significantly when attempting to differentiate these statistics because web sites often contain multimedia and even game frontends may have the necessity for a browser supporting multimedia streams to operate as designed. All of these concepts draw attention to the future of the Internet and the viability of the ever changing model. The common conception is that the Internet was designed for providing access to websites. However businesses are thriving on providing media-rich services such as TV and movie content, cloud-based application services, and even replacement for retail services. To be specific (in case you missed it), these attributes are referencing large companies such as Netflix, Microsoft, Google (owner of YouTube), and Amazon.

It would seem logical for video users to desire a full-screen experience. This can vary depending on the device used to view the content. A standard definition (SD) television comes to mind as a basic display size. Consider that a SD stream at 640x480 pixel density at the National Television System Committee (NTSC) recommended refresh rate of 30 Hz at the common color depth of 24 bits per pixel equates to about 211 Mbps if transmitted uncompressed. That bandwidth requirement goes much higher with full-definition at 1920x1080 resolution at 60 Hz – a massive 2.78 Gbps bandwidth requirement when uncompressed. As you can see, a single uncompressed video feed is out of the range of most Internet connections for home users and just a few feeds would congest most ultra fast Internet connections that a business might have as well. Compression addresses the sheer bandwidth requirement by removing the duplicate information between frames. How this is accomplished varies between encoding schemes. Similarly the desired output quality also impacts how far a source can be compressed.

There are a variety of methods to encode and compress a video stream [28, 12], and thankfully a lot of them can still save a significant amount of space over the original uncompressed source without noticeable differences between them. The result is the ability to stream video within reasonable limits with nearly imperceptible differences to the human eye from the original source. It has been a journey over time and not all encoding schemes are equally efficient.

The effects of transporting video content across networks is probably most acutely felt by individuals in the broadcasting and film/TV industry. There is a common challenge of transporting high-bandwidth video content efficiently over existing networks with the highest levels of availability. There can be serious consequences on advertising revenue and end-user experience if a live event experiences downtime during broadcasting. In that context, both availability and efficiency in transporting high-quality video are equally important. The constraints for broadcasters include short-range microwave services and costly satellite services that offer sub-par reliability and service quality, and create high latency. Router-based networks and terrestrial leased-line services are not suitable or robust enough for high-definition content to the point that many studios distribute high-definition video on tape for rendering and viewing. Tape distribution is a costly and inefficient method for high-definition video content delivery. Compression of HD content for transmission over lower-bandwidth networks does degrade signal quality but there is a valid point on what a user can interpret as perceptible differences in quality.

1.2 Diversity of Video Transport

Video can be streamed through any number of mechanisms. While the typical client-server model is still applicable, the topology can be further complicated by content delivery networks (CDN), multicast versus unicast, and live content versus video-on-demand (VOD). Transcoding requirements for a variety of devices adds to the complexity.

1.3 Contributions

Several objectives are accomplished in this report. First, we review a facilitating factor of video transport in compression. Next, we investigate the technologies that companies like Netflix, Amazon, Google, YouTube (owned by Google) use as the basis for their video transport. Additionally, we will correlate the assumptions made here to reports published by analytical organizations. The experiments performed here involve multiple transport methods, several different network topologies, and includes many of the common transcoding standards used today.

1.4 Outline of the report

The rest of the project report is organized as follows. Chapter 2 describes the historical basis of video streaming standards, related work to video streaming, and provides insight on what network consumption can tell us today. Chapter 3 describes experiments performed with video streaming applications within a diverse array of environments. Chapter 4 provides a conclusion to the project report and discusses future work.

Chapter 2

Related Work

2.1 Video Streaming Standards - MPEG

MPEG focuses on three frame types [36]. I frames represent intrapicture frames that are completely independent and are not reliant on other frames. P frames are predicted picture frames and are based on other reference frames, in particular the previous I frame. Finally, B frames are bidirectional predicted picture frames as an interpolation between I or P frames before or after it. In summary, the I frame is a compressed frame based on the original content, the P frame is the difference from the I frame, and basically the B frames fill in the gaps. Figure 2.1 shows how an original source stream is compressed through MPEG. An important point with MPEG frames is that they are not transmitted in order because of the reliance on the nearby frames. In the case of Figure 2.1, frames are logically displayed to the rendering client in I B B P B B I order. However, given that B and P frames are dependent upon I frames, the compression standard actually transmits them in I P B B I B B order. It's up to the client to arrange them correctly but this is a natural sequence given the order of dependency of the frames.

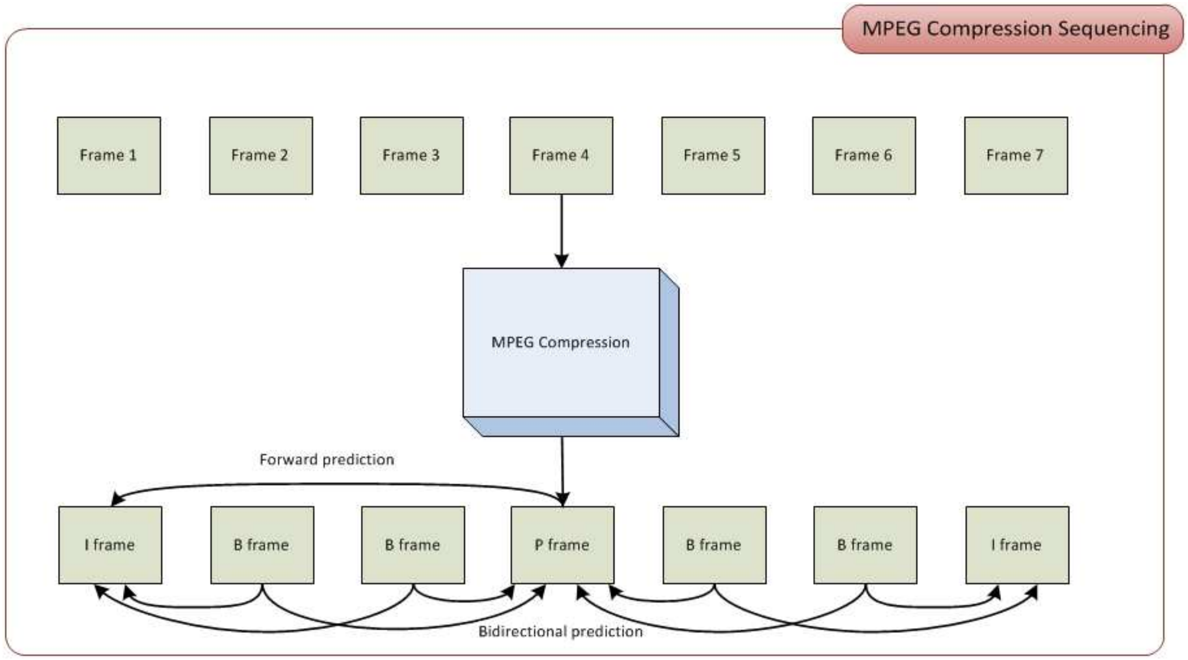

Figure 2.1: MPEG compression sequencing [36].

While video on the Internet is not new by any means, the impact has certainly evolved over time. We shall set aside any of the technical complexities of how to generate or encode a video, installing the right drivers to ensure proper playback, and converting legacy media to a current standard. Here we focus on video that could be playable on a CD. One of the most widely used compression standards in use today (MPEG) goes back to 1993 for the first iteration. The Moving Picture Experts Group established MPEG-1 with a data flow rate of up to 1.5 Mbps with 2-channel stereo audio [27]. But how many websites were available in 1993? A few hundred – total [10]. And these sites did not have the complex HTML standards that are available to developers today. One could deduce that video had a mild impact at the outset of the Internet's expansion through websites.

While high-definition content would not be as prevalent until 1998, 5.1-channel surround sound has been around since the 1980's. Computerized video seemed to trail the technological basis but rebounded quickly (as do most computer-related advancements). By 1995, MPEG-2 was standardized and included the ability to encode DVDs, which were developed that same year [27]. With data rates up to 40 Mbps, encoding in MPEG-2 was a huge step forward and included the ability to interlace all the way up to high-definition standards. Given the documented standards, Blu-ray players today are required to decode three formats including MPEG-2, MPEG-4 AVC, and SMPTE VC-1. MPEG-2 was chosen as the standard for over-the-air (OTA) digital television by Advanced Television Systems Committee (ATSC). MPEG-2 is currently very prominent throughout multiple industries. Think of all the digital signals that were actively transmitting after February 2009 when analog TV was terminated.

MPEG-3 enhancements were rolled into MPEG-2 as a new part [27]. The standard was thought to be duplicitous and therefore not much work was done to expand this recommendation. Most work focused on extending MPEG-2 or the new recommendation of MPEG-4.

By 1998, MPEG-4 was standardized and in 2003, MPEG-4 part 10 or also known as MPEG-4 Advanced Video Coding (AVC) [**?**, 27]. MPEG-4 AVC has another common name in H.264 which is the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) name given to the standard since both names equate to the synonymous joint development of the encoding scheme. MPEG-4 was actually developed to address the inability of MPEG-2 to stream on the Internet, given that MPEG-2 required so much bandwidth at the time. MPEG-4 was also developed to provide video services to mobile devices, which inherently have more bandwidth restrictions than typical landlines. While MPEG-2 has impeccable quality, MPEG-4 is more efficient while delivering the same end product. MPEG-4 does not necessarily compress video with a better formula or pack the same data into a smaller space so much as it makes assumptions in order to remove unnecessary data from the stream. Think of a moving car from one frame to the next. If the background image (buildings, atmosphere, any stationary object) does not change, then the encoding scheme only needs to include the data for the moving car. The remaining image can be predicted from neighboring frames. Picture and frame prediction allow MPEG-4 AVC to be nearly twice as efficient as MPEG-2. A standard-definition video encoded with MPEG-2 may range from 2-5 Mbps and high-definition content average 15-20 Mbps. Compare that to MPEG-4 AVC that averages 1-2 Mbps for standard-definition and only 5-10 Mbps on average for high-definition. The quality of these videos is virtually the same and believed to have imperceptible differences to the human eye. MPEG-4 AVC / H.264 is even a front runner in the forthcoming HTML5 standard, primarily competing with open-source Ogg Theora. The main benefit with Theora encoding is that it can be distributed without licensing fees [28]. The downside to MPEG-4 AVC is that it does have a patent licensing scheme attached, meaning there could be a significant cost added to selling commercially available software that provides the encoding algorithm.

The focus of coding history has focused on the ITU recommendations for moving video. There are other ITU recommendations that reference these same codecs, such as H.323 that deals with packet-based multimedia communications systems. While there is relevance in these other recommendations, they deal with systems and equipment specifics outside of the focus of moving video.

An important difference between MPEG-4 and its predecessor encoding schemes is that MPEG-4 utilizes objects while prior encoding schemes were based on the pixels themselves [33]. These objects can have intrinsic shape, texture, and motion properties. Creation of the video objects (VOs) is not specified by the encoding standard, but rather what happens to those VOs. The VOs are muxed together by a controlling unit that decides how many are transmitted, the number of layers and the level of scalability.

2.2 Video Streaming Standards - ITU

The ITU-T numerical representation for MPEG-4 AVC of H.264 [?] is quite advanced compared to the first draft of digital video encoding in H.120. Specifications for H.120 were originally published in 1984. The ITU-T H series focuses on all audio visual and multimedia systems. H.120 (characteristics of visual telephone systems, codecs for video conferencing using primary digital group transmission) was not very practical in that while it has adequate spatial resolution, the temporal quality became an issue. It became necessary to address the encoding scheme in order to stay within the boundaries of the data rate stream limitation of 1544 Kbps for NTSC and 2048 Kbps for PAL. A revision to H.120 in 1998 added background prediction and motion compensation to gain efficiency, but not to the degree that future standards would employ.

The same year (1990) that H.120 became a formal recommendation, the ITU-T also published H.261 [33] (coding of moving video, video codec for audiovisual services at p x 64 kbps where p is in the range 1-30). The reason for focusing on multiples of 64 Kbps is the fact that this recommendation is targeted at ISDN (Integrated Services Digital Network) lines that are based on those ratios. While sharing a similar maximum data rate with the prior recommendation of around 2 Mbps, H.261 could also be as efficient as 40 Kbps. The goal was to cover resolutions including CIF (352x288-pixel) and QCIF (174x144-pixel). H.261 is widely thought to be the first practical digital video coding standard and is more or less the basis for many video coding schemes today. There are multiple methodologies as the basis for

H.261 including inter-picture prediction, transform coding, and motion compensation. A unique aspect of the H.261 standard is that it only includes how to decode the video, not encoding the video stream [?]. Encoding the video is not detailed and could vary by device or application – as long as decoding functions normally then there is no need to apply unnecessary rules. Furthermore, decoders have the option of applying any post-processing steps before rendering the video to display. Video data is typically sent out by the streaming server into fixed size packets that contain the frame divisions. Moreover, the packet header may contain a sequence number, the time the packet was sent, and the relative play out time of the associate video frame – information necessary to reconstruct the video stream again on the client side.

H.263 was the next standardization formalized after MPEG-2 Part 2 (H.262). Similar to that of H.120, H.263 [33] also focused on videoconferencing from a low-bitrate perspective [9] and completed formal recommendation by late 1995 and into early 1996. The historical importance here is that a lot of Internet content such as Flash video on YouTube, Google Video, and MySpace was originally encoded in an incomplete implementation of H.263. Further enhancements that led to label changes such as H.263v3, H.263++, and H.263 2000 were the equivalent of MPEG-4 Part 2. The main improvements of note are in regard to codec extensions.

Figure 2.2 shows the efficiency of the major standards over time. The time provided in the x-axis references when that standard became formally accepted and the time slots between major encoding revisions does vary significantly. The efficiency is calculated by dividing the resolution of each frame by the data flow rate. Compression alerts the one-to-one ratio of frame pixels that can be displayed given a particular data flow rate. The rates calculated are the maximum compared to the maximum resolution that could be encoded. For example, a 1.5 Mbps data flow rate with a

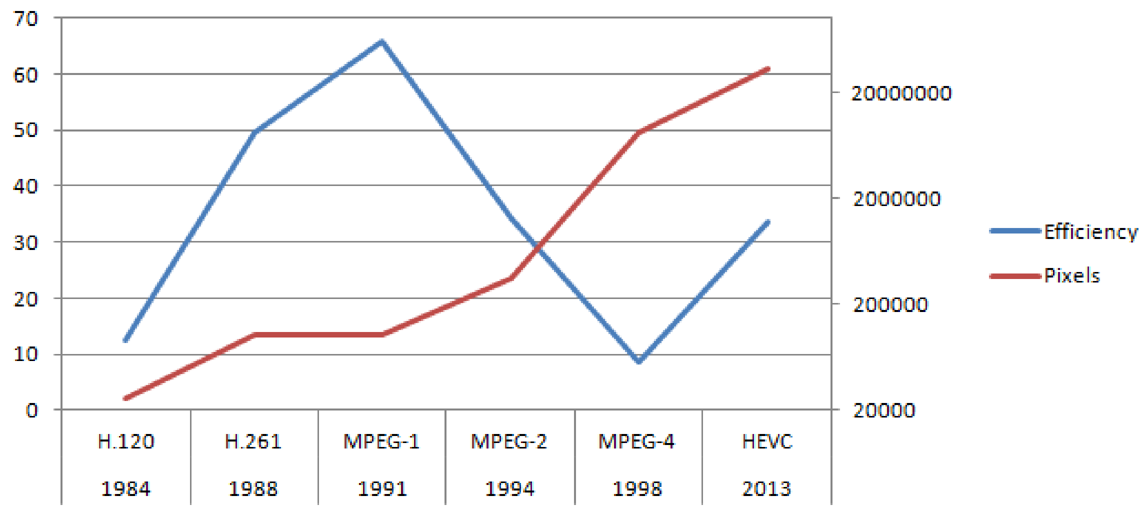

Figure 2.2: Data rate versus encoding efficiency.

maximum resolution of 352x288 pixels in MPEG-1 back in 1991 is theoretically more efficient than a 9.8 Mbps data flow rate with a maximum resolution of 720x480 pixels in MPEG-2 in 1994. However, the argument is that the increased resolution is worth the exchange for a less efficient data flow rate. As a comparison of the changes to the frame size divided by the data flow rate, the frame size itself in terms of number of pixels (frame height multiplied by frame width) is the secondary axis of Figure 2.2. Note the exponential growth in the number of pixels, which is calculated by multiplying the frame height by the frame width. What becomes apparent is that the frame size was the primary driver in the 1990's while encoding efficiency seems to be the goal before and afterwards for these major standards. Another interesting point is that besides the dual gain of efficiency and frame size on just the second major iteration, the only other time that both efficiency and frame rate increases is with HEVC to be formalized in 2013.

2.3 Compression in the Industry

Google primarily uses Flash and MP4 containers for Google Video and YouTube. Those sites accept WebM (VP8 video and Vorbis audio codec), MPEG-4, 3GPP, MOV, AVI, MPEGPS, WMV, FLV formats and transcode them accordingly. This seems appropriate given that nearly 75% of video content available is in Flash format. While this statistic may seem overwhelming, it is important to note that the very same content is usually available in H.264 as well. There is a site for YouTube that allows for HTML5 content, which removes the necessity for a Flash player plug-in to be installed with the browser. The formats provided through the HTML5-compliant site are H.264 and WebM formats. Google uses MPEG-4 AVC by default, which is one of the most recent standards. Google even has launched a series of videos in 4K resolution format (4096x3072 pixels).

Amazon digital video is available in H.264/AAC encoding within an MP4 container and VC-1/WMV9 encoding with a WMV container. These are compatible with Macintosh and Windows computers, respectively. Both encoding schemes used by Amazon have an approximate 1.5 Mbps video bit rate and 256 Kbps audio bit rate. Amazon instant video uses Flash, so a browser plugin is also necessary in this case.

First generation standard-definition Netflix content was streamed in WMV containers using a WMV3 encoding [7]. The WMV container is actually an ASF extension which is a technology that Microsoft developed as part of the Windows Media framework. A significant benefit of ASF is that content can begin streaming before the entire video is buffered. The background for the format and encoding is that Netflix partners widely accepted the Janus components in WMDRM10. This first generation content had different bit rates (0.49, 0.98, 1.56, 2.15, 3.32 Mbps) with a resolution of

740x480, which is the standard DVD resolution.

Second generation standard-definition Netflix content [7] was available through the Silverlight player and uses VC1 Advanced Profile encoding with PlayReady DRM. A key benefit above the first generation is that each Group of Pictures (GOP) header contains the frame size and resolution which allows the bit rate to adjust on the fly with fluctuating bandwidth. Moreover, the second generation content is compatible with a wider range of browsers and devices. The second generation content is available in more efficient bit rates (0.37, 0.49, 0.98, and 1.46 Mbps), thus loosening the restrictions on bandwidth requirements. Second generation high-definition (another benefit of VC1) content utilizes both 2.5 and 3.7 Mbps bit rates at up to 720p resolution. Netflix fully acknowledges their desire to limit bandwidth for high-definition content, while much better than standard definition, likely is not on par with Blu-ray discs of today. Their supposition that the bandwidth requirement would be out of the reach of the general domestic populace agrees with the general theme of this report. Resolutions of 1080i/p have only recently become available through Netflix for the same reason. Netflix has been working Dolby Digital to add 5.1 channel surround sound to their video in the last couple of years. Prior to that, Netflix content was limited by stereo audio only.

Third generation Netflix content is available above a 720p resolution and can consume up to 2.3 GB per hour [14]. This equates to a 5.23 Mbps maximum download speed requirement and coincides with the author's personal experiences in using the service. Other sections of the Netflix blog [16] provide exact measurements of 4800 kbps for video and 384 kbps for audio. This total of 5184 kbps is also in line with the maximum bandwidth requirements experienced by users and other Netflix blog notations previously mentioned. Table 2.1 shows the summary of the evolving video delivery mechanisms for Netflix.

Table 2.1: Summary of Netflix video delivery evolution [16].

textbfNetFlix Generation	textbfAttributes
First	WMV container, WMV3 encoding, 0.49 - 3.32 Mbps, 740x480 max resolution, 2-channel stereo sound
Second	Silverlight, VC1 Adv Profile encoding, PlayReady DRM, 0.37 - 3.70 Mbps, 1280x720 max resolution, on the fly adjustment for variable bandwidth, 5.1-channel surround sound
Third	5.23 Mbps, 1920x1080 and above for max resolution

2.4 Mobile Devices

Mobile devices have taken a stronger position in terms of what they can do and value added not only to entertainment but to the business sector as well. Some reports have shown that as recently as this year that smartphones have taken the lead for the first time over feature phones in terms of number of users [32]. This implies a fast growing user base for highly capable mobile devices. However there needs to be some caution when applying a label of "capable". Some devices are capable only from the moment they leave the store to the user or when they have been reset to factory defaults. While a device has the hardware components to theoretically playback a particular video file, it may be unable to do that if the operating system is not configured correctly or too many applications are overloading the available resources. Wireless providers are completely satisfied with users leaving bandwidth available. They make money off the sale of applications that can slow your device down, and subsequently prevent a user from fully utilizing the bandwidth they are paying for. Bandwidth is at a premium for wireless providers. Yet they are still making money on both fronts when a user subscribes to significant bandwidth but purchases applications that actually slow down their device.

Sandvine estimates that in the near-term, mobile users will watch streaming content nearly as frequently as is done on televisions and computers [37]. The Netflix application already consistently ranks within the top 25 free applications available on mobile devices. If Sandvine's reasonable assumption is right, then wireless carriers should forecast a significant change in their subscriber wireless usage [37].

2.5 Recent Developments

ITU-T Study Group 16, also known as the Video Coding Experts Group (VCEG), combined with Joint Technical Committee (JTC) 1 that is comprised of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) to form subcommittee 29 for coding of audio, picture, multimedia and hypermedia information. The subsequent working group 11 deals with coding of moving pictures and audio. In short, that would be ISO/IEC JTC 1/SC 29/WG 11 that is also known as MPEG. VCEG and MPEG together form the Joint Collaborative Team on Video Coding (JCT-VC). These experts convene four times a year to debate and propose international standards for compression, decompression, processing, and coded representation of moving pictures and audio [34].

JCT-VC is currently engaged in a joint call for proposals on scalable video coding extensions of High Efficiency Video Coding (HEVC). The goal for HEVC is to double compression levels provided by the latest standard in MPEG-4 AVC. You can imply that a HEVC video stream, when compared to H.264, would be half the size or double the resolution. The schedule for HEVC is shown in Table 2.2.

The standard ITU naming scheme applies to HEVC as well with the associated label of H.265 [**?**].

Table 2.2: Recent schedule for high efficiency video coding [13].

Date	**Milestone**
2012/05/11	Draft Call for Proposals
2012/05/11	Availability of Test Materials
2012/07/02	Availability of Single Layer Anchors
2012/07/21	Final Call for Proposals
2012/10/01	Submission of Deliverables to Chairs
2012/10/10-19	Evaluation of Proposals and Start of Collaborative Design

2.6 Related Work - Adaptive MPEG-4

There is a natural tendency to rely on UDP for video streaming in order to avoid buffering the lost segments for every client. UDP also has a subtle efficiency gain over TCP in that there is no necessity to have the protocol initiate flow control – the faster the data can be sent, the better. Even so, there has been other work done to create protocols to address some of the reliability issues with UDP streams found in common formats such as MPEG-4. Adaptive MPEG-4 video streaming with bandwidth estimation, or Video Transport Protocol (VTP) [3] provides end-to-end congestion control while maximizing the quality of real-time MPEG-4 video streams. VTP sends packets using UDP and adds congestion control at the application layer [3]. VTP is dependent on the receiver for the bandwidth estimation as each end destination may have different requirements. VTP allows the client to respond to the server with information that may, on the fly, adjust the sending rate and the bitrate at which the video stream is transmitted. Results from a real network testbed prove that VTP fairly shares bandwidth with TCP over congested networks in all but a few extreme cases while delivering consistent quality video [3]. Such proposals leverage ingenuity to solve difficult issues but come with their fair share of complex challenges as well. For example, it is beneficial to have a priori knowledge of typical network utilization and general link capacity. Further, VTP requires the same video sequence to be pre-encoded at several different compression levels in order to adjust the bitrate on the fly when network congestion is detected and a corresponding quality shift is necessary.

2.7 What Networks Are Telling Us

Data capacity is growing fast. Until recently communications service providers have not had good visibility or a clear picture into individual subscriber demand on their networks [31]. The overall plan to load balance and manage networks has been based on aggregate capacity demand that has long been available. A survey in OSP Magazine [31] details Internet usage data from 55,000 subscribers across a variety of access networks shows the unique dataset component of traffic traversing the access network. This is important because the access network is the only network segment that every Internet-bound packet must traverse [31]. The referenced dataset covers monthly usage, or total data downloaded in one month, for each subscriber in the network. The data from across various access networks of communication service provider networks details insight on actual bandwidth consumed by application and endpoint for every IP flow.

2.7.1 Monthly Usage

The variance for monthly usage for the 55,000 subscribers was between 0.1 GB to nearly 1000 GB. Sorting this wide range into buckets provides a baseline. The survey in OSP Magazine considers subscribers that are separated into standard usage bins [31] detailed in Table 2.3.

Figure 2.3 details the percentage of total subscribers that fit into these different usage profiles per month.

Table 2.3: Subscriber usage bins [31].

Gigabyte Usage Per Month
Less than 8 GB per month
8 to 16 GB per month
16 to 32 GB per month
32 to 64 GB per month
Greater than 64 GB per month

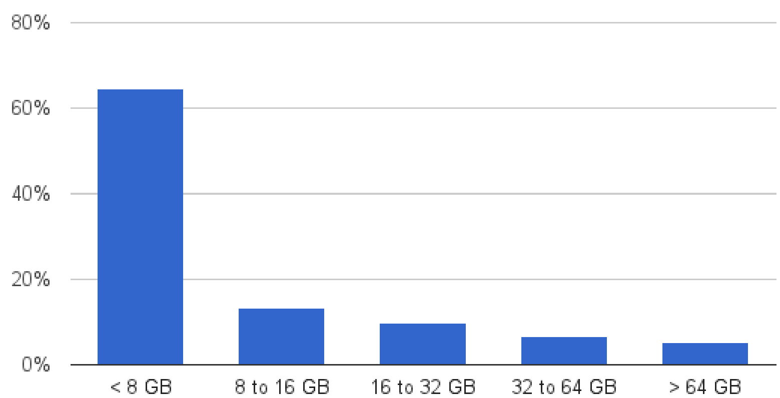

Figure 2.3: Percentage of subscribers in each usage bin [31].

When considering the lower usage bins, over 75% of subscribers use less than 16 GB per month. That leaves less than a quarter of the subscribers that fall into the last three usage bins that indicate greater than 16 GB per month. Figure 2.4 represents the same usage bins for percentage of total usage as opposed to total subscribers.

As you might correlate from Figure 2.4, the same high-usage bins ($>$ 16 GB) in Figure 2.4 represent greater than 75% of the total data downloaded and combined there is less than a quarter of the data comes from the two lower total data usage bins ($<$ 16 GB). The numbers are disproportionate in that 22% of the subscribers utilize 77% of the data. This might be rationalized by the fact that it is possible the

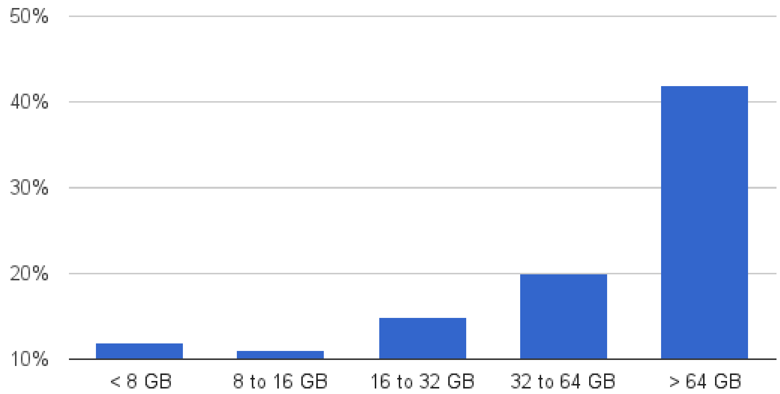

Figure 2.4: Percentage of total data [31].

22% could be paying for higher bandwidth.

2.7.2 Trending

There are tools that provide unique data about the access network in relation to the bandwidth utilized. These tools provide interesting trends that can be discovered by drilling down into the usage patterns of end users. Peak bandwidth, for example, is one of the parameters contained in the dataset for each subscriber in the network. One can make the assumption that peak bandwidth utilized is reasonably close to the provisioned bandwidth. We now further dissect the five usage bins discussed above by further dividing them into three bandwidth bins. The breakdown into assignments of 1.5 Mbps, 4 Mbps, and 8 Mbps will provide reasonable data points. Figure 2.5 is comparable to Figure 2.3 except that each usage bin is divided into the component bandwidths.

There are rational results from Figure 2.5 due to the clear relationship between

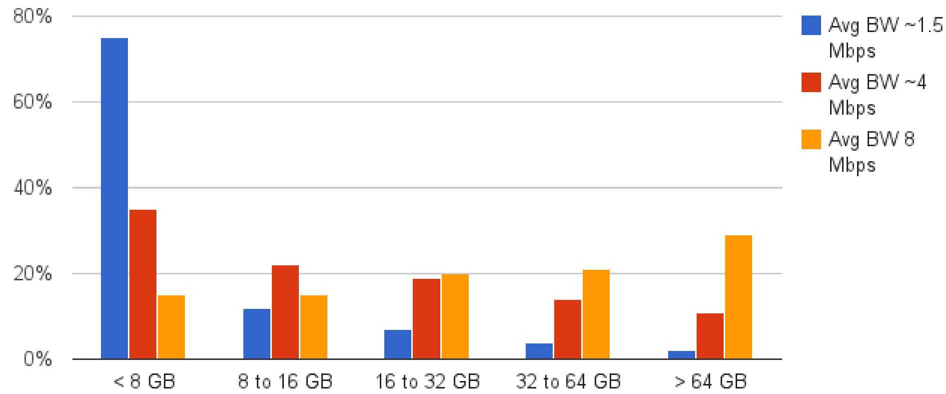

Figure 2.5: Percentage of subscribers vs provisioned bandwidth [31].

the assumed service bandwidth and monthly usage. Of the subscribers provisioned for 1.5 Mbps (blue bars), most (75%) download 8 GB or less per month, while only 2% download more than 64 GB per month. Downloading above 64 GB during any month would be implying a very high percentage of time that the full bandwidth was utilized. The trend is reversed for subscribers who are provisioned for 8 Mbps (yellow bars). Most (29%) of the high bandwidth subscribers download more than 64 GB per month, while only 15% download 8 GB or less. The implication here is that while there appears to be free bandwidth over the course of the entire month, higher bandwidth subscribers are generally using the additional bandwidth that they are paying for above low bandwidth subscribers. On the opposite end of the scale, it may be surprising to note that there is little variation in service bandwidth for subscribers using less than 64 GB per month. This could be an interesting point going forward for service providers.

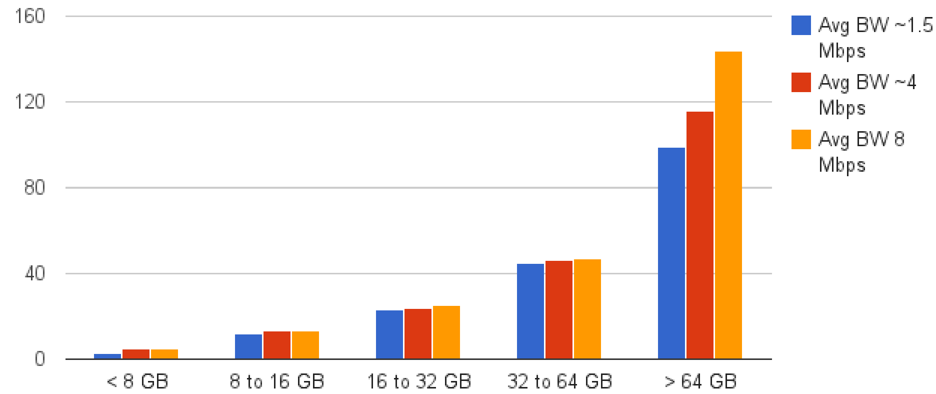

Figure 2.6: Average monthly usage vs provisioned bandwidth [31].

2.7.3 More Bandwidth

There are different stories being told between Figure 2.5 and Figure 2.6 in that the same data set is shown from different aspects. The theory is that Figure 2.5 represents what is occurring now on the network and Figure 2.6 is what will occur in the future. As you can deduce from Figure 2.5, there is a clear association between the lowest usage and the lowest bandwidth provisioned (in this case 1.5 Mbps). The same goes for the highest usage in that it is generally provisioned for the highest bandwidth (8 Mbps). You can assume that 8 Mbps service does cost more than 1.5 Mbps service, so this does appear to be fair for the subscribers. Figure 2.6, however, does show that there is nearly equal representation of each bandwidth within each particular usage bin. There is an assumption that all subscribers generally use more and more data per month and Figure 2.6 sheds some light as a potential forecasting aid to the behavior of subscribers. Let us consider that one hour of streaming video consumes about 1 GB of usage. In this example you could theorize that streaming an hour of over-the-top video (OTTV) per day would correlate to approximately 32 GB of data usage per month. Going back to Figure 2.3, we know that more than three-quarter of subscribers use less than 16 GB of data per month. The association with streaming video would be that these users generally watch less than a half hour of streaming video on average per day. As more content becomes available, it is easy to assume that the average amount of time spent on streaming video will increase as well. This is backed by statistics that show streaming video adoption is gaining in popularity along with the average stream duration. The rate of increase can be of some debate, but there is motivation from the perspective that the average time an American spends in front of a TV in a day according to The Nielsen Company is now approximately 5 hours per day [31]. Moreover, high-definition content is only becoming more prevalant with time, thus adding to the intensity of the average video stream. High-definition (HD) content is approximately four times that of a standard-definition video stream. Monthly bandwidth consumed for each subscriber would greatly increase by this fact alone without any other variables. The future provides insight to even more intensive video streams from 3D content to ultra-high definition video (UHD). UHD has several specifications [8] that range from four times that of full HD (4K, or 3840 x 2160) all the way up to sixteen times the image size of full HD (8K, or 7680 x 4320). There are some estimates that forecast 4K UHD content to be readily available in some markets by 2017 and 8K by 2022. Europe is expected to be up to 28 percent market penetration for broadcast UHD by 2025. The necessity for more bandwidth is clear at this point. Adding to the complexity is the fact that these video streams and usually unicast as opposed to broadcast or multicast. Each user's required bandwidth is impacted.

Even companies that have the sole function of providing streaming video are subject to social events and change. The Olympics was thought to be the source of a traffic drop to Netflix across the United States [19]. The estimates placed Netflix traffic down as much as 25% during the particular day in question.

2.7.4 Measuring Change

Subscribers have multiple reasons for upgrading or staying at their current service level. They might desire faster connectivity or sufficiently satisfied to stay with the existing service level, they may be displeased with their current service provider for any number of reasons including bundled services, or even be locked into contract for a set duration. Any multitude of reasons can slow down the frequency at which subscribers change or upgrade their service bandwidth. You might expect, from Figure 2.6, that subscribers who download <16 GB per month and have at least 8 Mbps service would generally have a tendency to desire more bandwidth (somewhat impatient) while those downloading >64 GB per month at only 1.5 Mbps service would still want more bandwidth but have extreme patience in doing so. Compass Flow Analyzer [5] is packaged software that has the ability to provide real-time views of bandwidth utilization on a per-subscriber, per-service, per-applications, and per-network basis. This offers entensive information where further granularity and groupings assists service providers in digesting more knowledge about their networks. What makes the dataset under discussion here is that it was taken from the access network, thus providing very accurate numbers for total monthly usage for a large number of subscribers. The comparison of percentage of total subscribers and percentage of total usage has interesting merit. A detailed snapshot of current usage is gleaned from the perspective based on subscribers, while the usage perspective provides a forecast of the future. What we can take away here is that there is a rapid increase in the amount of total usage even though the service bandwidth remains fixed.

2.8 Streaming Services

The largest component of data traffic is now generated by a single company – Netflix [18]. Netflix even surpassed BitTorrent file sharing traffic, which you would expect to still generate a significant volume. The thirst for additional bandwidth by the general subscriber base is enhanced by the fact the 29.7 percent of all peak downstream traffic in North America is sourced by Netflix. BitTorrent is still second to Netflix when taking into consideration the average of the day, 21.6 percent vs 22.2 percent respectively. An important basis in this data is the fact that BitTorrent traffic is only marginally dropping (19.2 percent to 18.8 percent over six months) over time. This means that Netflix traffic is in addition to, and not replacing, BitTorrent. While faster connectivity to the masses seems like a logical result, there are monetary factors that threaten the average user. Many Internet Service Providers are now charging in tiered bandwidth caps as opposed to unlimited data. This is in the same theme that is occurring with mobile devices. Step back a level and consider that there are real-time media providers other than Netflix, albeit not responsible for nearly the amount of traffic that Netflix is. Netflix CEO Reed Hastings believes that the rapid increase in utilization of bandwidth will result in gigabit to the home within ten years [17].

While dominating the market, the room for growth is still apparent. Netflix has 28 percent penetration within the United States [37]. While this is already a significant amount, there is clearly room to increase the percentage of subscribers. Possibly in an effort to do so, Netflix even ramped down the intensity of streaming services to Canada due to the fact that tiered-data usage is common there [15]. This allowed Canadian Netflix users to still enjoy content without exceeding their bandwidth cap limitations. The amount of network-capable devices is also rising. Almost everything you can buy has a network port or wireless functionality. And there is a high probability that if the device (television, game console, media center, mobile phone, tablet) can connect to the Internet then it also has the ability to stream video. Netflix, Vudu, Amazon Prime, and others offer a wide range of clients that permeate into a wide range of displays.

2.9 Impact on the Backbone Network

A technical report by AT&T delivers a consistent message regarding the impact of video on the Internet [32]. Figure 2.7 shows the average downstream traffic per subscriber has a stable growth pattern for a sample carrier. This appears to provide relevant data that can assist in forecasting the long-term growth of the Internet.

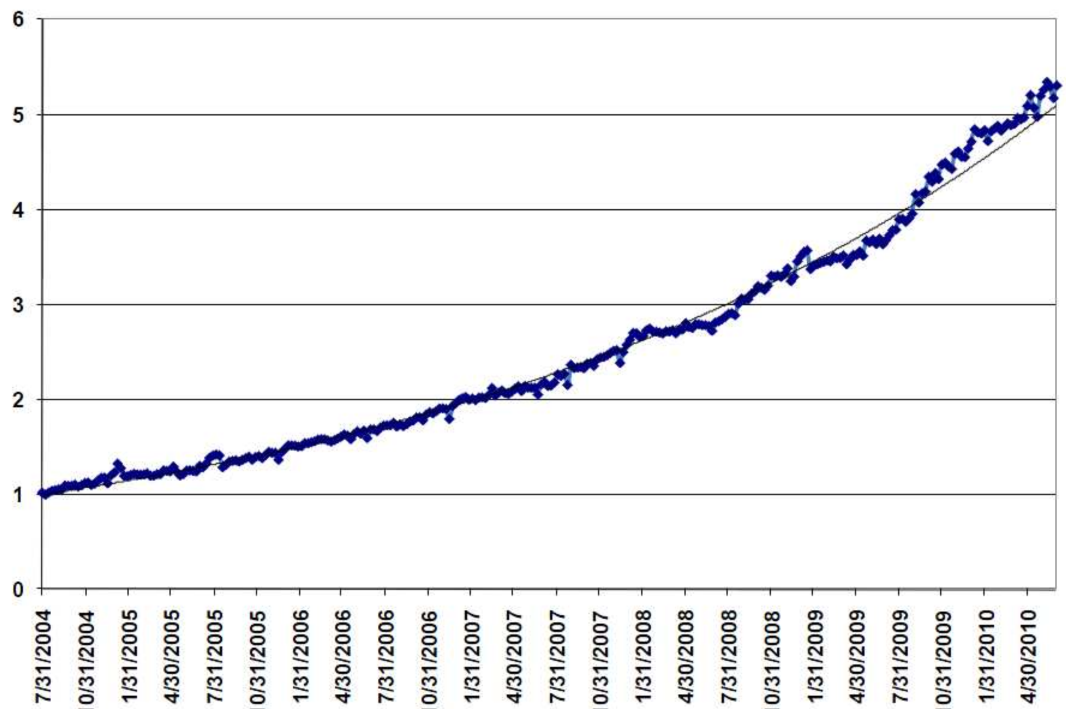

Figure 2.7: Normalized peak traffic per subscriber [32].

Figure 2.8 provides insight on the year over year growth rate of broadband penetration by geographic region. While growth rate was very high early on, it has tapered off significantly over the last couple of years [32]. Market saturation is the reasoning behind the slowed growth.

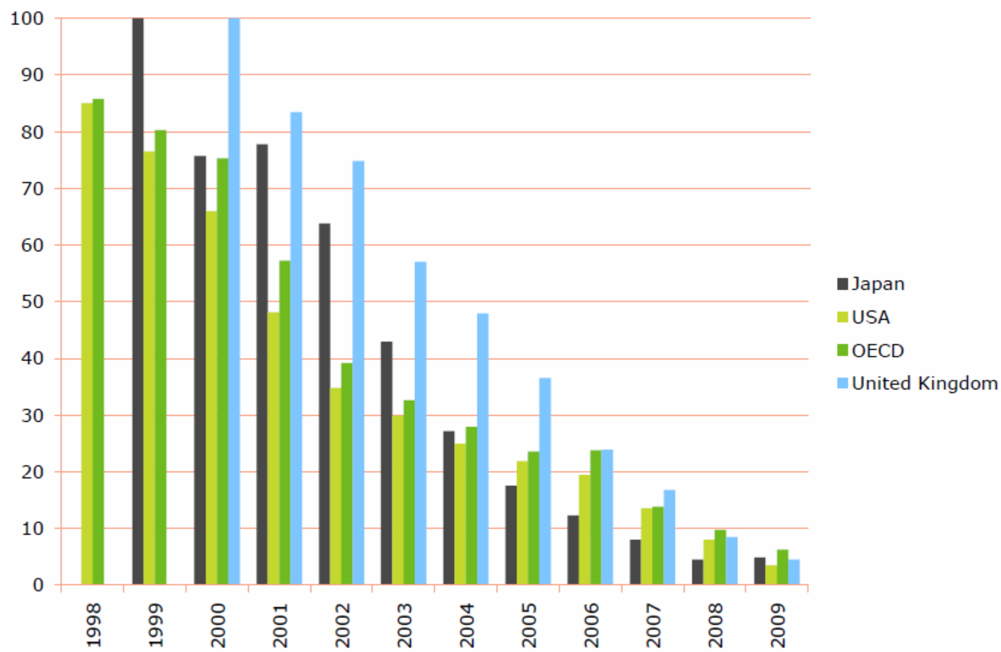

Figure 2.8: Broadband penetration rate [32].

The combination of the prior two types of growth (peak traffic per subscriber and broadband penetration) are shown in Figure 2.9. Wireless traffic (red) shows a similar pattern approximately 8 years after wireline (blue). Note that the normalized graph is in logarithmic scale. There is nearly 4 orders of magnitude growth on that backbone over the last 12 years [32].

Given the significant growth already measured, the next step is to analyze the traffic that is causing that rate of change. Figure 2.10 details the complete summary of the layer-4 traffic breakdown. It is interesting to note that P2P is decreasing as a percentage (though that does not imply a decrease in overall volume). Web/HTTP

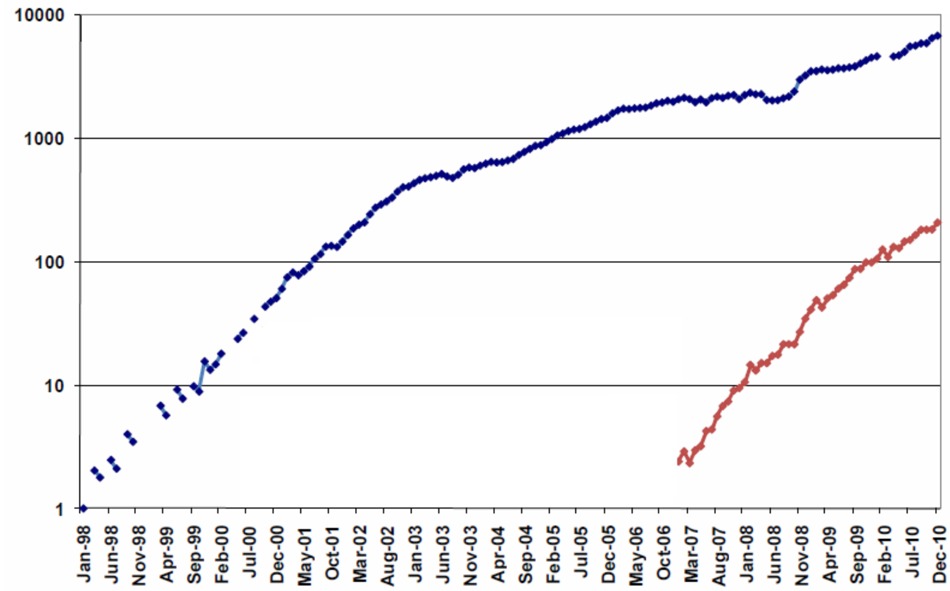

Figure 2.9: Normalized average traffic [32].

including multimedia has the largest gain over the 8 year span, followed by non-HTTP multimedia.

Given that Figure 2.10 grouped a signficant amount of traffic together, further analysis on a higher layer is required. Figure 2.11 is a stacked chart showing the normalized traffic volumes on layer-7. This separates multimedia from Web/HTTP and depicts a clear increase for Web/HTTP along with both HTTP and non-HTTP multimedia.

Video growth has an annual growth rate of 83% [32]. The implications are clear that traffic types have shifted over time to a multimedia focus. The importance of streaming video has become apparent.

To this point, we have reviewed compression, standards within the industry, and the impact to the Internet. Now the focus will shift to experiments that were performed to gain insight on a variety of technologies used in streaming video.

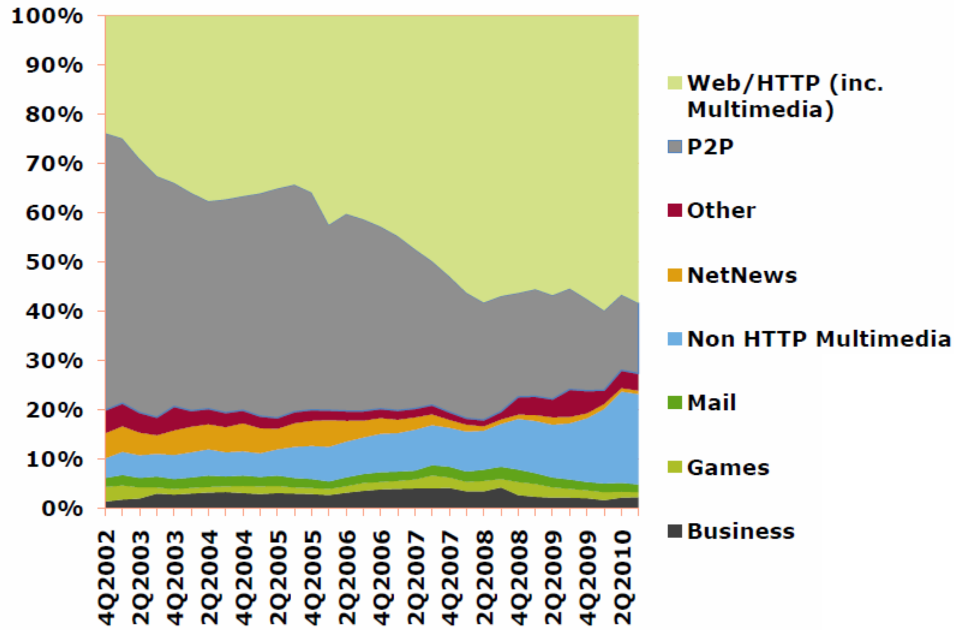

Figure 2.10: Layer-4 protocol breakdown [32].

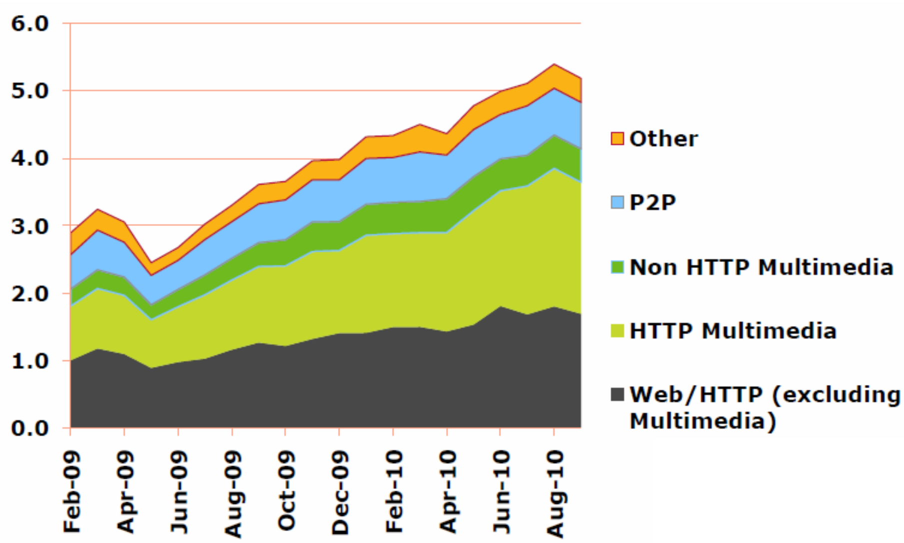

Figure 2.11: Layer-7 protocol breakdown [32].

Chapter 3

Video Streaming Applications and Environments

This chapter focuses on transcoding experiments across multiple different network testbed designs. There is a noticeable difference between the compression schemes used. Several network topologies add perspective of the impact to the network. There is a local testbed providing a subset of connectivity and network size that is established in the larger design. The local testbed includes a couple of layer 2 switches with personal computers connected to access ports on either end. A trunk port between the layer 2 switches completes the connectivity. The larger network testbed is also established through layer 2 switches but extends from the University of Nebraska to Rutgers University through ORBIT [20] lab nodes.

VideoLAN's VLC [29] software provides both client and server functionality. There are a variety of other packages that have been developed that are relevant to this topic including DLNA-compliant software that will be touched on.

3.1 Transcoding

There is a plethora of encoding standards and a variety of containers for streaming video content. It is often necessary to change formats on the fly (transcoding) in order to match functionality of the server to the client. There are many software packages available, but one of the more popular applications is the open-source solution by VideoLAN entitled VLC. VLC provides both the client and server functionality within a single application, as well as general video rendering capabilities for local files [29]. Moreover, VLC also provides transcoding options when streaming video content. VLC supports several different streaming styles including RTP, UDP, and direct over HTTP among others. For the sake of simplicity, this report focuses on the use of direct streaming over HTTP. The implication here is that the stream is based on unicast in that each client is sent a separate copy of the media stream from the server. Multicast has the desirable outcome of sending out a single copy of the video stream from the server to a group of recipients. However, multicast does have inherent complexity and may not even be available on some networks. Multicast also has a disadvantage in that video on demand services are no longer relevant since the entire group or recipients would be impacted simultaneously for every change request.

A source file with significant clarity and resolution was chosen for the streaming experiment. Movie trailers are often in high quality, high-resolution video streaming formats while keeping the duration relatively short. In this experiment, an Apple QuickTime .mov container housing a 1920x1080 resolution video file was chosen. The encoding scheme is H.264 for the sake of viewing what happens when a stream is required to transcode down in quality. The video clip being transcoded in this report comes from a freely available movie trailer of the forthcoming 2012 James Bond movie SkyFall. The video file was created on July 31, 2012 at a rate of 23.98 frames per second. The file “premiumrush2_trlr_02_1080p_dl.mov” is widely available on the Internet [26] and has a data flow rate of 6616 Kbps. The movie trailer also has AAC/MP4A audio stream data in stereo at 48000 Hz with a data flow rate of 156 Kbps, though the focus of this report is clearly on video as opposed to audio. The 130 MB file is 2 minutes and 33 seconds in length. This information is easily attainable from Linux operating systems by using commands such as “file” or “ffmpeg” to display metadata.

Figure 3.1: Video stream capture from the original source [26].

A car chase scene was chosen for the basis of the video streaming comparison between transcoding schemes. The reasoning is that there should be a significant amount of change across frames which should push the limits of encoding schemes considered not as aggressive or capable as H.264 that we utilize today. Figure 3.1 shows a slice from the original file that will be referenced in the transcoding variations going forward in this report. The method for streaming was direct over HTTP.

Given that the video stream is already in H.264 encoding within an Apple QuickTime .mov container, transcoding to H.264 within an .mp4 container should provide nearly identical video output. Figure 3.2 is a screen capture at around the same 7 second mark as from the original image capture and does reflect this theory in that the image appears flawless. There are blurred areas but that is a factor of the obvious high-speed objects within the video and not related to the transcoding scheme.

Figure 3.2: Video stream transcoded to H.264 within an MP4 container [26].

The same source video file was transcoded to the Theora encoding scheme within an OGG container in Figure 3.3. This represents both a change in the container as well as a change in the encoding scheme. There is a subtle loss in video quality. For example, the skid marks underneath the car are now slightly jagged and not as sharp as they were in the previous video stream. This marks a significant change from the prior transcoding efforts in that both the compression mechanism and the container were altered before streaming the video. The latest Theora compression algorithms [28] give nearly the same quality as H.264. This is a reasonable assumption given that the image quality is very close but not quite on the same level as the original or even the conversion to another container with the same compression.

Dirac compression is supposed to have comparable quality to MPEG-4. Figure

Figure 3.3: Video stream transcoded to Theora within an OGG container [26].

3.4 shows the resulting image after transcoding to Dirac and the results are mixed. There are components of the image, such as the tire marks, that appear more smooth than say the Theora encoding. However there is an additional artifact introduced that does not appear in any prior transcoding experiment. In specific, there is a blurred line near the top of the image but limitations on images within this report make that distinction difficult to discern. Furthermore, the license plate numbers do not appear to be rendered at the same quality either. The overall impression is that while Dirac attempts to compete with other compression schemes, it lacks the ability to produce attributes at the same level as H.264 and Theora.

Transcoding back to a prior revision of MPEG should result in a noticeable loss in quality. Figure 3.5 does show the same image snapshot from the streaming video clip, but it is obviously not the same quality as the prior transcoding schemes. In particular, using the same comparison of tire marks from the prior transcoded stream, there appears to be some data loss as the image does not have the same clarity. The subjective component here may have some impact in that the exact frame taken from

Figure 3.4: Video stream transcoded to Dirac within a TS container [26].

each transcoded clip may be representing original content or a predicted frame.

Figure 3.5: Video stream transcoded to MPEG-2 within a TS container [26].

There could be an issue of processing power or other limitations that impact the resulting clip that was transcoded to DIV3 in an ASF (Advanced Systems Format) container. In fact, it was impossible to freeze the transcoded video stream at the same spot as prior encoding schemes in order to make a direct comparison. Figure

3.6 shows a another image that was taken from within a reasonable timeframe (within 2 seconds) as all the others. It is clear even from this image from another part of the clip that there is approximately 15% data loss on the top and bottom. And the remaining image, while discernible, also has significant quality loss. This is the first transcoding process that failed to generate a smooth video stream. There could be multiple correlations to skipping frames. The data could be delayed longer than the buffer could compensate for, there could be a hardware limitation (for example, processing power or memory allocation) that prevented a smooth transition, etc.

Figure 3.6: Video stream transcoded to DIV3 within an ASF container [26].

Finally, it appears that transcoding to WMV produces a completely unrecognizable video stream. Figure 3.7 proves that there is almost complete data loss in the video stream. As suggested with the impacted video stream transcoded to DIV3, there might be underlying factors that play into the data loss. For example, how efficiently an application can encode. All of these video streams were encoded using

VideoLAN's VLC media player for the sake of consistency.

Figure 3.7: Video stream transcoded to WMV within an ASF container [26].

3.2 Local Testbed

For the sake of simplicity and to alleviate any question about connectivity, a local testbed places natural constraints on an experiment. In order to simulate the longer-distance network setup, the local testbed was configured across the same local networking nodes. Furthermore, the same layer 2 connectivity was established so as to maintain consistency with the intra-campus configuration. Two different operating systems are utilized as endpoint workstations – Windows 7 Professional at one end of the testbed and RedHat Enterprise Linux 6 on the other.

Alternately, a Samsung LN40C630 television has been added as a client display. The layout is nearly identical with the addition of the network-connected Samsung

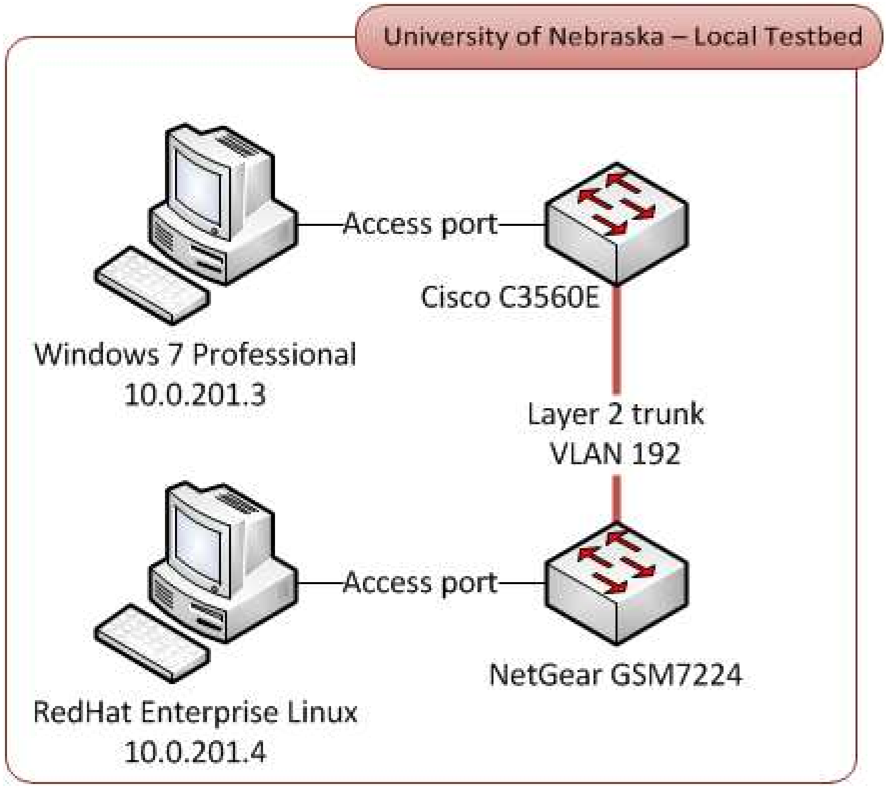

Figure 3.8: Local testbed - University of Nebraska Netlab.

television. The caveat here is that there are multiple connectivity options. Similar to the simple client-server model mentioned above, the Samsung television can be configured with a static IP address on the same private subnet. In this variant both workstations available could feasibly send streaming video to the Samsung television. The other option is to use the standard campus network ports available on the Cisco C3560E to both the Windows 7 workstation and the Samsung television. In this latter method, both devices retrieve an IP from the campus standard in using DHCP. This method also promotes the ability to use the generally available network as opposed to restricting the testbed to a private subnet. While this does allow for other campus networking access, the RedHat Enterprise Linux workstation is not available in this configuration. The NetGear GSM7224 is maintained by graduate students within the Computer Science Netlab and therefore does not extend the standard campus network. Both options are combined into Figure 3.9.

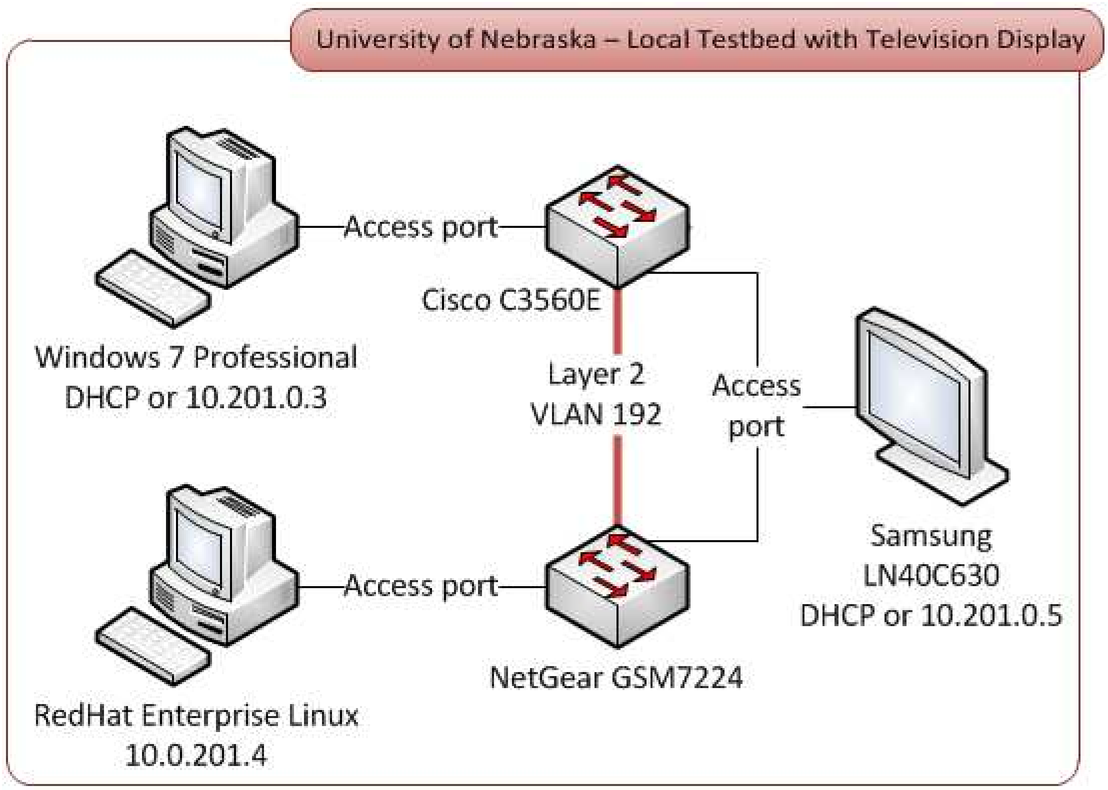

Figure 3.9: Local Testbed with television display - University of Nebraska Netlab.

3.3 Long-Distance Testbed

The long-distance testbed includes layer 2 connectivity between the University of Nebraska and Rutgers University as detailed in Figure 3.10. The unique attribute to this connectivity is that a layer 2 VLAN is used for nearly the duration of the data flow. Access ports are configured for the workstations at each end of the connection and trunk ports allowing VLAN 192 everywhere else in between. The allows both ends to establish a common subnet between then and appear as direct neighbors (single-hop) on the IP stack. Also unique to government agencies, laboratories, and education facilities is the use of Internet2 [2].

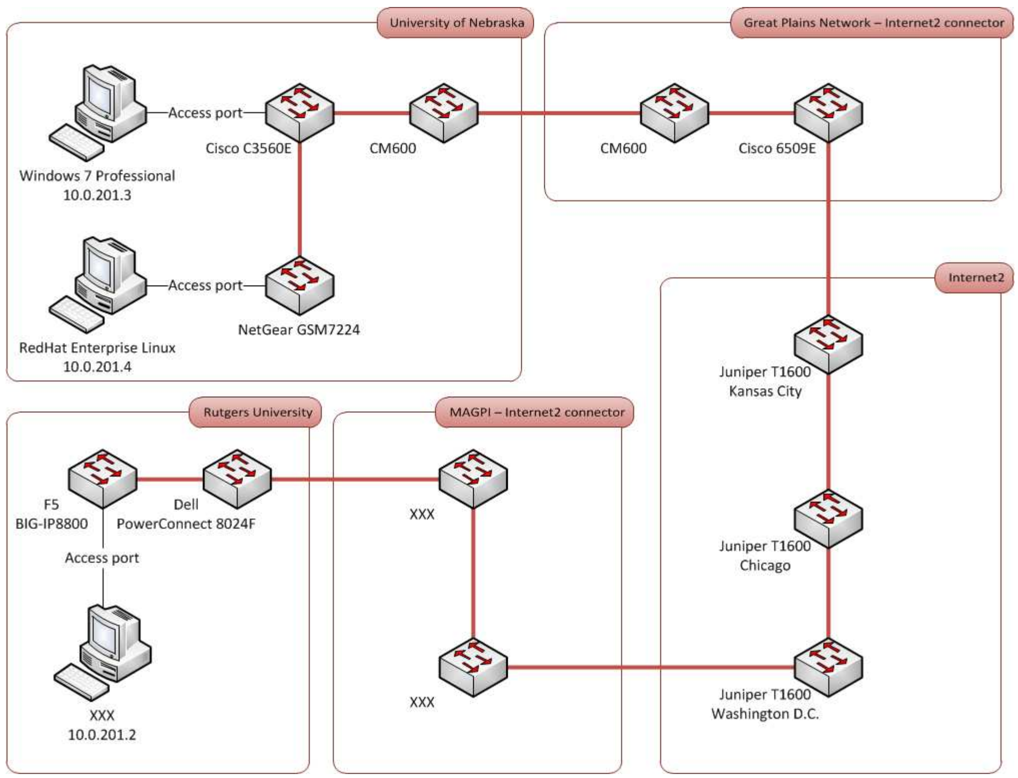

Figure 3.10: Long distance testbed from UNL to Rutgers.

3.4 Home Networking

All of these experiments have relevance in terms to the home user as well. In fact, the experimentation set was extended to include a true home networking environment. As it turns out, the author's home included Category-5 (Cat-5) cabling that connects a gigabit switch with multiple devices including personal computers (running Microsoft Windows 7 and RedHat Enterprise Linux 5), a network-connected Sharp television, and a Roku media player [25]. While wireless connectivity is becoming more prevalant in televisions as a built-in feature or optional adapter, there are still more units available with a standard network port.

3.4.1 Roku Media Player

The Roku [25] provides unique features in a home network topology. It does not store any content locally and subsequently streams all content. Similar to the client applications that exist within "smart" televisions, the Roku has individual channels for Netflix, Vudu, and Amazon. These channels are generally more advanced than the "smart" television clients. Some channels are designed for use with DLNA-compliant server applications. Testing has proved that transcoding can be performed on-the-fly on the server side to meet the client and even network bandwidth limitations.

3.4.2 Digital Living Network Alliance

A major caveat with home networking is the assumption that a local DLNA-compliant server is readily available on the same subnet. DLNA is a collaboration of media-centric organizations and was actually started by Sony [6]. Other members include AT&T, Broadcom, Cisco, Comcast, DirecTV, Dolby, DTS, Google, HP, Intel, LG, Microsoft, Panasonic, Samsung, and Verizon just to name a few of the major promoting members. The objective of DLNA is to add value to customers through standards-based interoperability. It comes down to providing a basis that multiple leading companies can utilize and feature within their products while simultaneously opening the door to universal connectivity with other products (not necessarily from the same manufacturer). The average home user is now able to deploy a set of network-connected devices that can function together with a reasonable amount of effort.

The context of reasonable effort includes the assumption that an IP-based network is available. This also implies the ability for DLNA-compliant devices to have access to a DHCP server or the user will have to manually configure the network settings. While some of the "smart" devices have built-in clients on top of the network port, such as Netflix, Vudu, and Facebook, there are other DLNA specifications that allow the device to poll the local network for streaming services.

There are DLNA specifications for mobile devices and home infrastructure devices that interface with mobile devices, but this report will focus only on the wider-range of home networking devices. DLNA specifications divide home networking devices into several classes. Table 3.1 shows the different functions of the DLNA classes.

Table 3.1: Digital Living Network Alliance (DLNA) classes.

Digital Media Server (DMS)	Store content available to DMPs and DMRs. Example: PCs and NAS.
Digital Media Player (DMP)	Find and play content from DMS. Example: TVs and game consoles. Content is pulled by the DMP.
Digital Media Renderer (DMR)	Plays content from DMC, which in turn finds content from DMS. The concept is that content is pushed to the DMR.
Digital Media Controller (DMC)	Finds content on a DMS and plays it on a DMR. Example: tablets, Wi-Fi digital cameras, PDAs.
Digital Media Printers (DMP)	Provides printing services to the DLNA home network. Example: network-enabled printers.

3.4.3 Experiments

To see what capabilities the Sharp HDTV has, multiple encoding schemes were tested from various DLNA-compatible servers on the same network. As it turns out, the Sharp TV would only recognize Apple Quicktime variants – both MP4 and the larger file equivalent of M4V extensions. Right out of the gate, this limited the test set that could be utilized. That may seem like a disappointment but the good news is that the TV does have a built-in Netflix client. Moreover, the network capabilities were referenced from the application list as DLNA, suggesting that multiple DLNA servers could be used. It turns out that this is the case and several servers running on the same Windows 7 Professional PC were all recognized by the TV. One of the assumptions in this experiment is that files are available in the correct format. Users might be disappointed to upload their digital camera videos or have content archived in another format that is not recognized by the television client.

3.4.4 Emerging Devices

There are a number of different micro-computing platforms that include HDMI, gigabit network (and Wi-Fi), USB, and SD card slots all wrapped into 3-inch form factor. These devices usually run some flavor of Linux that can be modified to support a media center application. The result is a much wider range of supported compression algorithms and containers. The downside is that this introduces another link in the chain and potentially yet another remote control to manage.

A Raspberry Pi model B [23] was ordered for the purposes of experimentation within the home networking model. However the high demand for this very recent product has produced a backlog of orders and the device is unavailable at the time of this writing.

3.5 PlanetLab

When attempting to execute experiments that require remote hosts, PlanetLab [22] lends itself as a natural fit. While mostly hosted by research institutions, PlanetLab also has nodes on Internet2's Abilene backbone through co-location and routing centers [22]. This collection of nodes is based on a common software package that facilitates updates, monitoring, auditing, and key distribution. There are over 1000 nodes across more than 500 sites. Most nodes exist in North America and Europe, but there are nodes all over the world. PlanetLab easily promotes world-wide availability for unique projects that need geographic distribution.

This report targets PlanetLab nodes within Rutgers University and the Unversity of Nebraska to form a similar network path as described earlier with layer-2 connectivity. Table 3.2 provides the details for each node used in the streaming video experiments. The location information is an approximation based on whois.net lookup query results.

Table 3.2: PlanetLab nodes.

Hostname	**IP Address**	**Location**
planetlab1.rutgers.edu	165.230.49.114	110 Frelinghuysen Road, Piscataway, NJ, USA
orbpl1.rutgers.edu	128.6.192.158	110 Frelinghuysen Road, Piscataway, NJ, USA
planetlab1.unl.edu	129.93.229.138	14th & R Street, Lincoln, NE, USA
pl2.eng.monash.edu.au	130.194.252.9	Wellington Road, Clayton, Victoria, AU
planetlab-1.sjtu.edu.cn	202.112.28.98	Tsinghua University, Beijing 100084, China
planetlab1.ucsd.edu	132.239.17.224	9500 Gilman Dr, La Jolla, CA, USA

The previously transcoded file “premiumrush2_trlr_02_1080p_dl.mov” was also used to stream via real-time transport protocol (RTP) to gauge network impact. While RedHat Fedora Core 8 is not necessarily the best desktop-oriented platform to be testing, a rather old version of VideoLAN VLC software was able to install and function properly. There are some complexities in using unsupported operating systems and software applications that are similarly aged. With the proper Yellowdog Updater, Modified (YUM) software repositories added, one can still resolve the dependencies to install VLC version 0.8.7c even though version 2.0.4 is the latest currently available.

The following code excerpt skips the details of software installation and shows the commandline execution used to start VLC in server mode.

```
vlc file:premiumrush2_trlr_02_1080p_dl.mov --sout \
  '#duplicate{dst=std{access=http,mux=ts,dst=:1234}}' --repeat
vlc file:premiumrush2_trlr_02_1080p_dl.mov --sout \
  '#rtp{mux=ts,dst=76.84.225.246,port=1234}'
```

The first code option streams a file over HTTP on port 1234 and repeats the file on an endless loop. The second code option sends an RTP video stream directly to 76.84.225.246 over port 1234. The IP address in the RTP stream correlates to the www.stbecker.com site that has a limit of 10 Mbps downstream and 1.0 Mbps upstream. The imposed bandwidth limitations provide a natural bottleneck that can impact the video stream. Two things have to occur for the RTP session to be established to the client that is using network address translation (NAT). First the traffic has to be allowed from the remote server through the local firewall at www.stbecker.com and secondly, the traffic has to be port redirected to the appropriate workstation client. At that point, the client only needs to open an RTP session against localhost on the appropriate port to view the video stream. Figure 3.11 shows the number of packets received per second on the client side over the UDP/RTP stream.

Figure 3.12 shows the number of bits transferred per second from the perspective of the client. As expected, it appears very similar to Figure 3.11. Given the client downstream limitation of 10 Mbps, we can see that this RTP video stream comes very close to that barrier. There are two points in both figures that show a drop to nearly zero (both packet and bit rate). Similarly, there are artifacts in the video stream playback at right around the 10 second and 125 second time points. There is enough

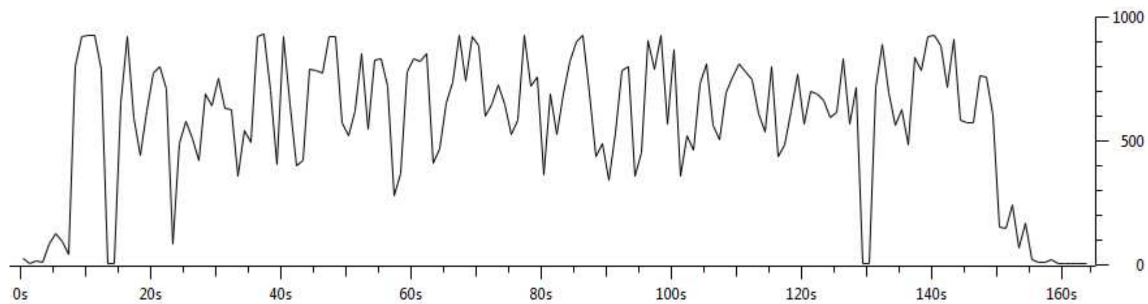

Figure 3.11: RTP packets per second.

significant data loss at these points to miss entire frames. There is a noticeable effect on the video playback.

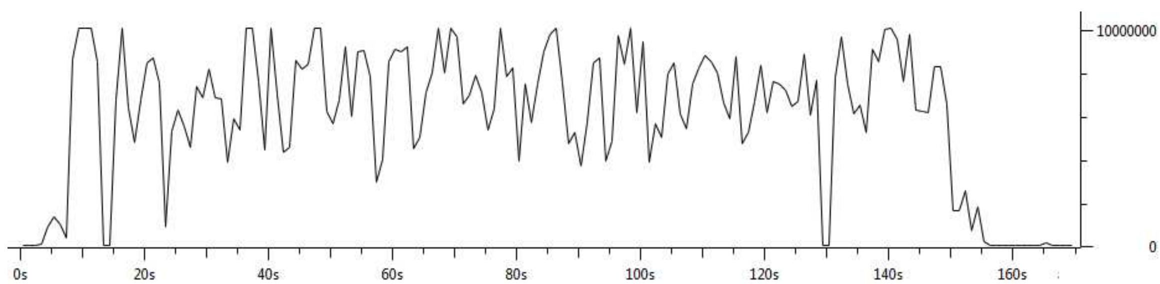

Figure 3.12: RTP bits per second.

Statistical analysis of the MPEG-4 RTP video stream is show in Figure 3.13. The display filter ensures that only packets coming from the desired source RTP video streaming server are captured and displayed. This helps place an average alongside the prior two figures. The rate for average packets per second (611 pps) and average bitrate (6.704 Mbps) seems in line with the variability over the maximums of 1000 pps and 10 Mbps. It just so happens that an Ethernet header with 14 bytes, an IP header with 20 bytes, a UDP header of 8 bytes, an RTP header of 12 bytes, and a payload of 1316 bytes totals 1370 bytes. Figure 3.13 does show the average packet size is 1370 bytes so each packet appears to be as large as possible (expected).

Display

Display filter:	ip.addr==165.230.49.114
Ignored packets:	0

Traffic	Captured	Displayed	Marked
Packets	94158	93179	0
Between first and last packet	184.868 sec	152.323 sec	
Avg. packets/sec	509.325	611.718	
Avg. packet size	1358.025 bytes	1370.000 bytes	
Bytes	127868919	127655230	
Avg. bytes/sec	691676.369	838053.468	
Avg. MBit/sec	5.533	6.704	

Figure 3.13: Video stream statistics summary.

Figure 3.14 shows that all of the 93179 packets within the video stream are within the same bucket of size classification. The packet size classification was a default metric layout provided by Wireshark [30].

Topic / Item	Count	Rate (ms)	Percent
Packet Lengths	93179	0.611718	
0-19	0	0.000000	0.00%
20-39	0	0.000000	0.00%
40-79	0	0.000000	0.00%
80-159	0	0.000000	0.00%
160-319	0	0.000000	0.00%
320-639	0	0.000000	0.00%
640-1279	0	0.000000	0.00%
1280-2559	93179	0.611718	100.00%
2560-5119	0	0.000000	0.00%
5120-	0	0.000000	0.00%

Figure 3.14: Video stream packet length.

The list of errors found in each packet is summarized in Figure 3.15. Detected transport stream frame loss is the highest error rate recorded by far. There may be multiple errors in each packet, which is the reason there are nearly half a million errors from only a hundred thousand packets.

Summary	Count
Detected TS frame loss	451685
Pointer value is too large (> remaining data length 183)	9011
Malformed Packet (Exception occurred)	6241
Malformed Packet (Exception occurred)	3851
Malformed Packet (Exception occurred)	2162
tvbuff.c:1092: failed assertion "DISSECTOR_ASSERT_NOT_REACHED"	461
Pointer value is too large (> remaining data length 103)	417
Malformed Packet (Exception occurred)	263
Malformed Packet (Exception occurred)	216
Pointer value is too large (> remaining data length 104)	213
Bad checksum	166
Malformed Packet (Exception occurred)	165
Pointer value is too large (> remaining data length 7)	148
Malformed Packet (Exception occurred)	134
Pointer value is too large (> remaining data length 3)	128

Figure 3.15: Video stream packet error summary.

Figure 3.16 provides similar statistics including the total bytes transferred and the average bitrate.

Address A	Port A	Address B	Port B	Packets
165.230.49.114	53850	192.168.200.26	search-agent	93 179

Bytes	Packets A→B	Bytes A→B	Rel Start	Duration	bps A→B
127 655 230	93 179	127 655 230	3.764901000	152.3235	6704427.75

Figure 3.16: UDP conversation summary.

Since this is an MPEG-4 video stream, there are three unique types of frames for video (this report does not provide details on the audio stream). I frames, P frames, and B frames should all be present within the video stream. Figure 3.17 shows the frametype and picture number from the beginning of the stream through 1 second. For the sake of a secondary axis overlay in the Figure 3.17, I frames have a value of 1 (least amount of prediction or none), P frames have a value of 2 (some prediction), and B frames have a value of 3 (comparatively the most prediction). As you might expect, an I frame starts the sequence followed by a string of B and P frames. What is also important to note here is that the picture number associated with each frame is not exactly linear in fashion. Zooming out to view several thousand frames would cause the difference in picture number to appear negligible within a smooth line, hence the reasoning to view only a single second with this figure. The picture number order appears to be consistently alternating frames by skipping forward two and then back one. The reasoning for this is that the decoded frame presented to the user does not need to be compressed in the same order due to the prediction dependency of the following frames. This graph also details the exact count of 24 frame in a single second which correlates to the 24 fps we would expect in the compression algorithm.

Figure 3.18 provides a similar view over a wider range of time. The alternating frame pattern is apparent between B and P frames in Figure 3.17 but only a single I frame was present. Figure 3.18 shows that I frames do appear at regular intervals but only every two and half seconds. This implies a significant amount of prediction based on these rare I frames to the other B and P frames. Also of importance is the apparent data loss between 5-7 seconds. This overlaps with the visual degradation of the video stream apparent from the VLC video application on the client side.

Figure 3.19 shows the maximum jitter for the entire stream. There is an approximate two and ahalf second transmission delay from the server to the client with a variance up to 156 milliseconds.

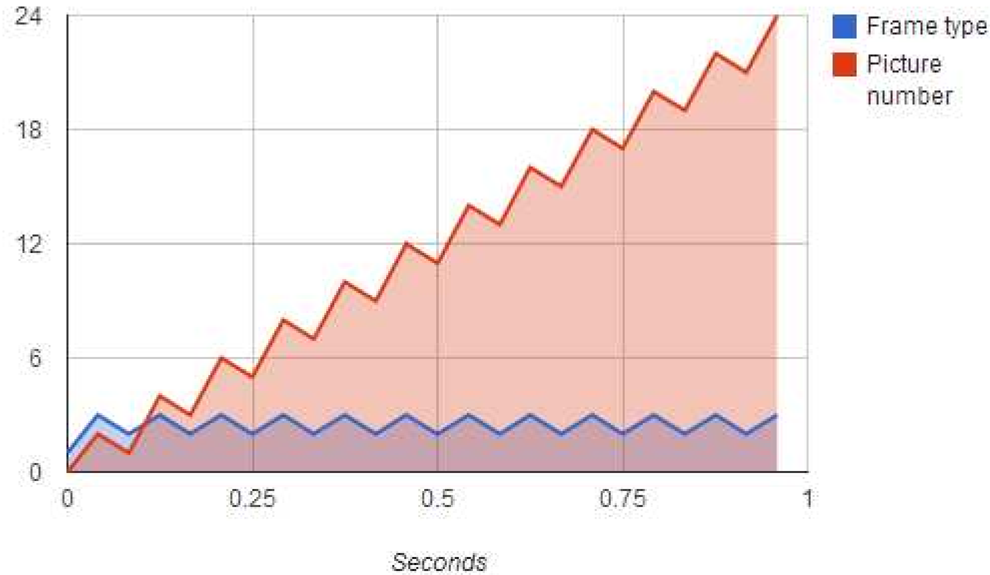

Figure 3.17: MPEG frametype and picture number by decoded time.

3.6 Layer-2 Connection with ORBIT

As described in a prior section detailing the network topologies, layer-2 (VLAN) connectivity between the University of Nebraska and Rutgers University provides the workstations on both ends to appear as local subnet peers on IP. This simplies the application setup in that there appears to be only a single hop between the client and server. In reality, there are many hops between these geographic diverse sites. Furthermore, the layer-2 connectivity provides a reservation for bandwidth that prevents other traffic from interfering.

The difference between using PlanetLab [22] across the commericial Internet and ORBIT [20] across Internet2 with a layer-2 reservation is significant at the beginning of the stream. Figure 3.20 shows the packets captured per second on the client side

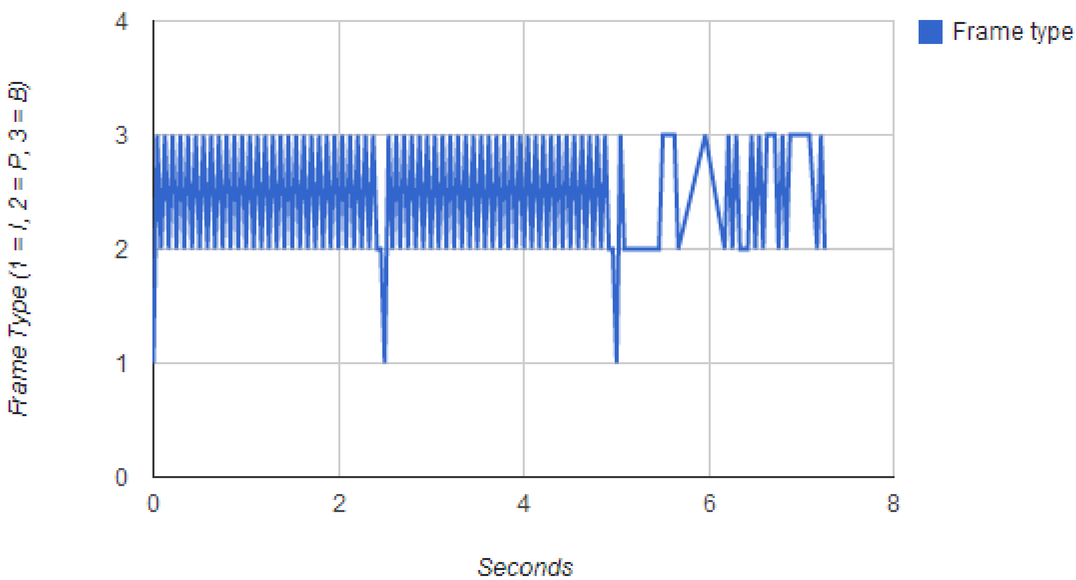

Figure 3.18: MPEG frametype by decoded time.

Analysing stream from 165.230.49.114 port 53850 to 192.168.200.26 port 1234 SSRC = 0x6B521607

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
4740	23375	2502.27	156.40	-2775.36	45572.45	SET	[Ok]
79460	35828	2409.00	152.11	-2502.16	846154.88	SET	[Ok]
4742	23376	0.78	146.64	-2775.09	45583.30	SET	[Ok]
79461	35829	0.79	142.62	-2501.90	846165.69	SET	[Ok]
4745	23377	0.78	137.49	-2774.81	45594.14	SET	[Ok]
79462	35830	0.78	133.72	-2501.64	846176.56	SET	[Ok]
4749	23378	0.99	128.90	-2774.75	45604.99	SET	[Ok]
79463	35831	0.76	125.38	-2501.35	846187.38	SET	[Ok]

Max skew = -2775.67 ms.
Total RTP packets = 101945 (expected 101945) Lost RTP packets = 8768 (8.60%) Sequence errors = 2034
Duration 152.30 s (90 ms clock drift, corresponding to 90053 Hz (+0.06%)

Figure 3.19: Network jitter and delay.

after traversing a layer-2 path from the server. There is a spike at the beginning that could not occur with the prior test from PlanetLab due to bandwidth limitations. The client application allows for 300 milliseconds for file buffering so the frontend spike is a natural result. We can theorize that the frame loss from PlanetLab to the site www.stbecker.com was the result of bandwidth limitations restricting the ability to buffer.

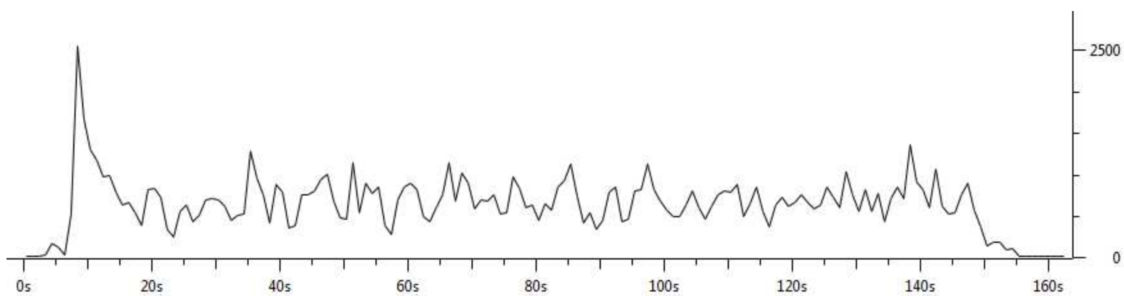

Figure 3.20: RTP packets per second over layer-2.

Figure 3.21 reinforces the transmission assumptions regarding the client application attempting to buffer the frontend.

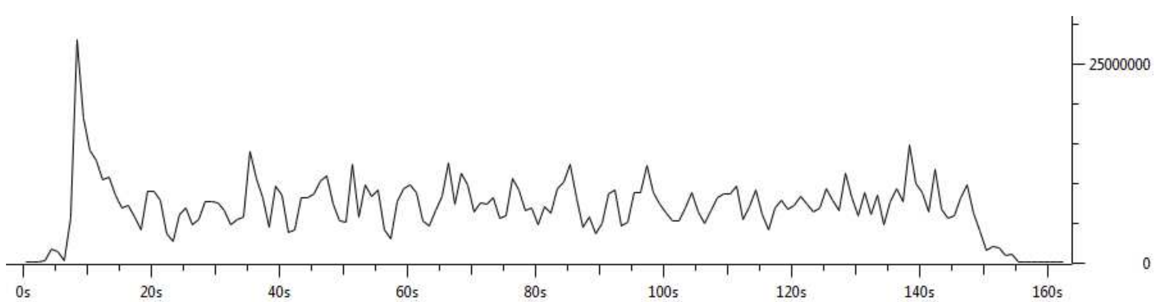

Figure 3.21: RTP bits per second over layer-2.

Figure 3.22 shows that layer-2 resulted in slightly more packets and overall byte count. Similarly, there is a higher transfer rate in terms of bandwidth utilized. This is expected given that the subjective interpretation of visually determining video quality rendered a much cleaner file.

Display

Display filter:	ip.addr==10.201.0.2
Ignored packets:	0

Traffic	Captured	Displayed	Marked
Packets	102295	101981	0
Between first and last packet	162.116 sec	151.401 sec	
Avg. packets/sec	631.000	673.581	
Avg. packet size	1365.751 bytes	1369.615 bytes	
Bytes	139709547	139674670	
Avg. bytes/sec	861789.836	922546.517	
Avg. MBit/sec	6.894	7.380	

Figure 3.22: Video stream statistics summary over layer-2.

There is almost no difference in packet length between the bandwidth-limited IP test and the layer-2 reservation as shown 3.23.

Topic / Item	Count	Rate (ms)	Percent
Packet Lengths	101981	0.673581	
0-19	0	0.000000	0.00%
20-39	0	0.000000	0.00%
40-79	0	0.000000	0.00%
80-159	31	0.000205	0.03%
160-319	0	0.000000	0.00%
320-639	0	0.000000	0.00%
640-1279	0	0.000000	0.00%
1280-2559	101950	0.673376	99.97%
2560-5119	0	0.000000	0.00%
5120-	0	0.000000	0.00%

Figure 3.23: Video stream packet length over layer-2.

The fact that the layer-2 connection shows nearly the same amount of errors in Figure 3.24, particularly those of transport stream frame loss, is unexpected.

Protocol	Summary	Count
MP2T	Detected TS frame loss	493234
MP2T	Pointer value is too large (> remaining data length 183)	9854
MPEG SECT	Malformed Packet (Exception occurred)	6872
MP2T	Malformed Packet (Exception occurred)	4330
DVB EIT	Malformed Packet (Exception occurred)	2475
MP2T	Pointer value is too large (> remaining data length 103)	490
MP2T	tvbuff.c:1092: failed assertion "DISSECTOR_ASSERT_NOT_REACHED"	440
MPEG CA	Malformed Packet (Exception occurred)	321
MPEG PAT	Malformed Packet (Exception occurred)	290
MPEG DSM-CC	Malformed Packet (Exception occurred)	259
MP2T	Pointer value is too large (> remaining data length 104)	229
DVB NIT	Malformed Packet (Exception occurred)	195
MP2T	Pointer value is too large (> remaining data length 12)	156
MP2T	Pointer value is too large (> remaining data length 9)	154
DVB SDT	Malformed Packet (Exception occurred)	152

Figure 3.24: Video stream packet error summary over layer-2.

Similar to Figure 3.22, there is comparable results in Figure 3.25 that details the UDP conversation summary from server to client over layer-2.

Address A	Port A	Address B	Port B	Packets	Bytes
10.201.0.2	57716	10.201.0.3	search-agent	101 950	139 671 500

Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
101 950	139 671 500	0	0	3.251212000	151.3761	7381428.75	N/A

Figure 3.25: UDP conversation summary over layer-2.

While the chart for MPEG frametype and picture number (not shown) for layer 2 connectivity does not change in reference from zero to one second, expanding the timeframe provides an opposing viewpoint. Figure 3.26 shows the similar pattern but without the significant data loss that was shown over layer 3.

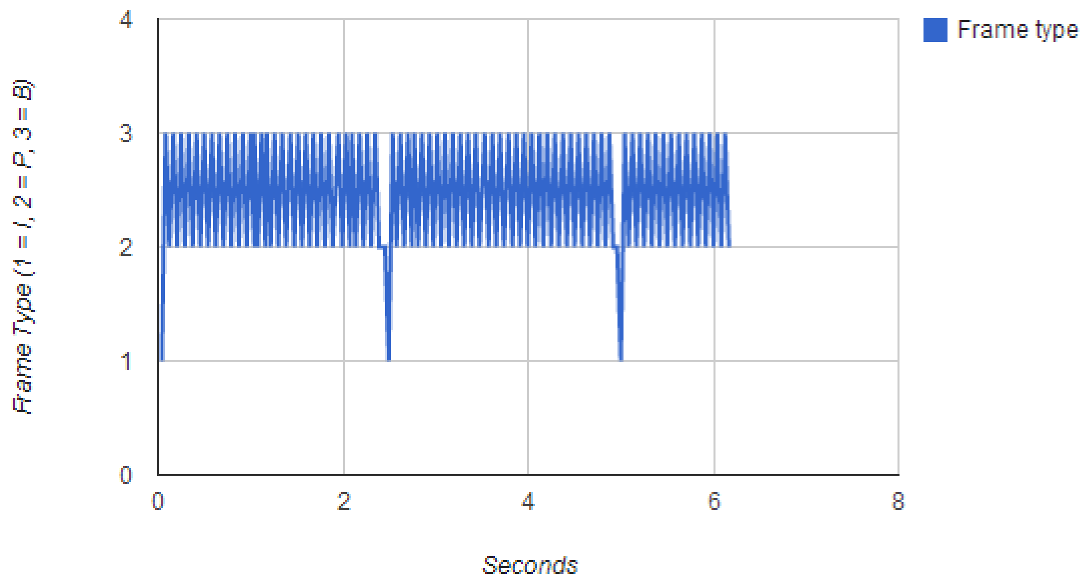

Figure 3.26: MPEG frametype over layer 2 by decoded time.

Finally, we see that the jitter described by Figure 3.27 over layer 2 is significantly less than what was observed with layer 3 with PlanetLab.

Analysing stream from 10.201.0.2 port 57716 to 10.201.0.3 port 1234 SSRC = 0x7AEB9824

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
85525	55136	2.99	1.12	0.13	920821.63	SET	[Ok]
85524	55135	0.00	1.10	1.60	920810.81	SET	[Ok]
85526	55137	1.02	1.08	0.64	920832.50	SET	[Ok]
85523	55134	3.00	1.07	0.09	920799.94	SET	[Ok]
85527	55138	1.03	1.05	1.13	920843.31	SET	[Ok]
85522	55133	0.00	1.04	1.57	920789.06	SET	[Ok]
85521	55132	3.00	1.01	0.05	920778.25	SET	[Ok]
85528	55139	1.26	1.00	1.38	920854.19	SET	[Ok]

Max skew = 2.03 ms.
Total RTP packets = 101949 (expected 101949) Lost RTP packets = 1 (0.00%) Sequence errors = 1
Duration 151.35 s (2 ms clock drift, corresponding to 90001 Hz (+0.00%)

Figure 3.27: Network jitter and delay over layer 2.

In summary, we have reviewed the efficiency of compression standards over time and how those standards have been implemented by comparing technologies used in the industry to stream video. Given those parameters, we have experimented with VideoLAN's VLC software to transcode a high-definition video stream to a variety of compression standards and used subjective reasoning (visual results) to determine quality. We have also examined multiple network topologies including those with controlled boundaries (local testbed and home network) and those with uncontrolled boundaries (PlanetLab and layer-2 VLAN connectivity to Rutgers). Finally, we took an objective look at the impact of a video stream on a computer network. We found that layer 2 reservations performed noticably better than layer 3 and that the visual imperfections were in-line with network analysis. Table 3.3 provides a brief summary of the results found between layer 2 and layer 3 connectivity.

Table 3.3: Results of network analysis when comparing layer 3 to layer 2.

Layer 3 (PlanetLab)	**Layer 2 (ORBIT)**
Visually impacted at several points within the stream.	No apparent impact.
Content limited by bandwidth threshold.	No noted limitations.
Contrained buffering.	Full buffering.
93,179 packets.	101,981 packets.
6.7 Mbps (average).	7.4 Mbps (average).
611 packets / second.	673 packets / second.
152.323 seconds.	151.401 seconds.
127,655,230 bytes.	136,674,670 bytes.
451,685 TS frame loss detections.	493,234 TS frame loss detections.
156.40 ms jitter (maximum)	1.12 ms jitter (maximum).

Chapter 4

Conclusions and Future work

4.1 Conclusions

This report has focused on experiments performed using diverse topologies and includes analysis of the effects of streaming video on computer networks. There is also discussion regarding related work in the industry including trending and forecasting, all of which indicate sustained dominance in terms of bandwidth consumed.

We have also reviewed a variety of encoding standards, particularly those recommended by the ITU and MPEG. Efficiency gains are apparent since the inception of video streaming, but other factors drive the reasoning behind Sandvine's statement that "even doubters must now agree that the age of Internet video is upon us" [37]. Market penetration of streaming video services has reached an all-time high, and yet continues to grow aggressively. One out of every four subscribers in North America already has a Netflix account, and one out of ten active subscribers on the Internet at any given moment are using Netflix. And while standard-definition content was the dominant resolution of media ten years ago, high-definition content has continued to become more prevalant ever since.

The amount of network-capable devices is also rising. Almost everything you can buy has a network port or wireless functionality. And there is a high probability that if the device (television, game console, media center, mobile phone, tablet) can connect to the Internet then it also has the ability to stream video. Netflix, Vudu, Amazon Prime, and others offer a wide range of clients that permeate into a wide range of displays. You no longer have to search specifically to find a device that offers this functionality, but rather it is a matter of how many services are offered on each device. And if the functionality already exists, users are free to adopt the services at their leisure if they have not already done so.

All of these factors lead to a natural tendency to assume there is simply a need for more bandwidth. Internet Service Providers are attempting to monetize this need by piggy-backing on the idea from mobile devices to charge for usage as opposed to unlimited data. One might speculate that Internet Service Providers are attempting to retain a foothold on over-subscribed lines by offering contracts with timeline term obligations. In the end, the natural result should be an exponential increase in the average bandwidth available to subscribers as well as an increase in the amount of time that Internet line is in use.

The assumption for more bandwidth is a given. There is another underlying theme that should be addressed. If sheer Internet bandwidth is not available, the underlying infrastructure has to be dissected to look at ways it can become more efficient. We have seen through experimentation that layer 2 reservation provides experience to the user than layer 3. Service-oriented architecture lends itself to the cause of separating particular protocols out from the rest. It is apparent that such intensive traffic as streaming video having the exact same prioritization as everything can cause immediate problems. There are already suggestions to equate quality of service into networking infrastructure [1] to provide some layer of separation. The end goal is to transform the Internet to transport multimedia more effectively.

A significant case against simply addressing the symptom (lack of bandwidth) correlates to mobile devices. Assuming that the influx of mobile traffic continues accelerating, the assumption that wireless mobile devices are the future is an easy deduction. Similar to their stationary counterparts, wireless mobile devices need to carry video streams as well. Screen resolution is also increasing as new generations of devices are introduced, which raises the bar on quality capabilities for streaming video on each device. Wireless carriers are already rapidly moving to tiered data usage plans, which effectively restricts the end user or monetizes the amount at which a user exceeds their threshold.

4.2 Future work

4.2.1 Additional Platforms

This report uses personal computer workstations on both ends to stream video as a source and receive content through application clients. There was a brief description about a Roku [25] device but there is room to expand the list of streaming devices. New developments such as the Raspberry Pi [23] would be useful to experiment with to compare the marginal cost to the capabilities of the device. Additional testing with the “smart” clients that exist in some televisions would be interesting in respect to the number of supported encoding schemes.

4.2.2 DLNA Expansion

The Digital Living Network Alliance (DLNA) provides a network basis for the home user that is relatively easy to setup. Most devices only support a portion of the official

DLNA Guidelines (as of this writing, version 1.5 is the latest). The guidelines stipulate a subset of encoding schemes to be recognized and the associated certification merely correlates to those that are fully supported. The impact is that a DLNA-certified device might work very well for a single video compression standard but not for any others. It would be useful for devices to support a wider range of codecs and containers.

While not necessarily part of the intent for DLNA, allowing certified products the ability to stream content from the Internet would expand the usage class well beyond the in-home limitations that are currently present. There are methods to deploy a DMS (Digital Media Server) such that it can retrieve and make available Internet content to a DMR (Digital Media Renderer) client. This flies in the face of the default mechanism within DLNA to search the local subnet for content servers but would open up the guideline certification to an infinite list of content on the Internet.

4.2.3 Ethernet Standards

Enhanced ethernet [4] lends itself to the conclusion that streaming video requires a separate reservation from other protocols on the Internet through priority flow control (PFC). Ethernet already has the ability to pause and restart when a switch port buffer is full, but enhanced ethernet priority flow control splits Ethernet communication into eight channels that can be paused individually for more flexibility. Priority grouping known as enhanced transmission selection (ETS) provides another layer of traffic differntiation on a per-channel basis. Thus traffic with sensitive payloads are prioritized accordingly. Congestion notification distributes congestion (traffic shaping) between the edge and core services in an attempt to keep congestion closer to the edge. All of these attributes combined correlate to lossless characteristics that are critical in demand guaranteed packet delivery.

4.2.4 Video Delivery Modes

Content delivery networks are already doing a good job of shortening the distance from the content to the client [32]. Another mechanism to address distance and processing requirements is through peer-to-peer (P2P) multimedia streaming [35, 38]. Peer-to-peer [21] relies on a non-centralized topology to spread the workload among any number of nodes and could be prioritized by bandwidth, network distance, or computing resources available. The Internet2 [2] has facilitated the reservation process for bandwidth required to deliver a transatlantic concert where musicians in Miami simultaneously exchanged live content feeds with composers in Paris [11]. Additional projects are already underway, such as Remote Media Immersion (RMI), that are utilizing multiple streams to transmit content that "dramatically surpasses the quality of today's high-definition broadcast television" [24].

Appendix A

CLI Execution

A.1 ORBIT probe1 details

```
sbecker@probe1:~$ uname -a
Linux probe1 3.2.0-31-generic-pae #50-Ubuntu SMP
  Fri Sep 7 16:39:45 UTC 2012 i686 i686 i386 GNU/Linux
sbecker@probe1:~$ cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=12.04
DISTRIB_CODENAME=precise
DISTRIB_DESCRIPTION="Ubuntu 12.04.1 LTS"
sbecker@probe1:~$ vlc --version
VLC media player 2.0.3 Twoflower (revision 2.0.2-93-g77aa89e)
VLC version 2.0.3 Twoflower (2.0.2-93-g77aa89e)
Compiled by buildd on roseapple.buildd (Jul 24 2012 22:39:41)
Compiler: gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1ubuntu5)
This program comes with NO WARRANTY, to the extent permitted by law.
```

```
You may redistribute it under the terms of
  the GNU General Public License;
see the file named COPYING for details.
Written by the VideoLAN team; see the AUTHORS file.
```

A.2 RHEL 6 - YUM Package Installation

```
su -c rpm -Uvh \
  http://download.fedoraproject.org/pub/epel/6\
  /i386/epel-release-6-7.noarch.rpm
su -c rmp -Uvh \
  http://download1.rpmfusion.org/free/el/updates/6\
  /i386/rpmfusion-free-release-6-1.noarch.rpm
rpm --import http://packages.atrpms.net/RPM-GPG-KEY.atrpms
echo "[atrpms]
name=Fedora Core $releasever - $basearch - ATrpms
baseurl=http://dl.atrpms.net/el$releasever-$basearch/atrpms/stable
gpgkey=http://ATrpms.net/RPM-GPG-KEY.atrpms
gpgcheck=1" > /etc/yum.repos.d/atrpms.repo
rpm -ivh avrc6-tejas-barot-linux-0.1.0-1.el6.x86_64.rpm
wget \
  "http://www.ask4itsolutions.com/RPMs/Scripts/vlc6-installation.sh"
chmod +x vlc6-installation.sh
./vlc6-installation.sh
yum install vlc
```

A.3 Iperf Results

```
[root@cse-brama-17 ~]# iperf -c planetlab1.rutgers.edu
------------------------------------------------------------
Client connecting to planetlab1.rutgers.edu, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 129.93.230.4 port 44753
      connected with 165.230.49.114 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec  69.9 MBytes  58.3 Mbits/sec
[root@cse-brama-17 ~]# iperf -c planetlab1.unl.edu
------------------------------------------------------------
Client connecting to planetlab1.unl.edu, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 129.93.230.4 port 58021
      connected with 129.93.229.138 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.1 sec   113 MBytes  94.5 Mbits/sec
```

Bibliography

[1] A Service-Oriented Multimedia Componentization Model, http://works.bepress.com/cgi/viewcontent.cgi?article=1057&context=jiazhang.

[2] About Internet2, http://www.internet2.edu/about/.

[3] Adaptive MPEG-4 Video Streaming with Bandwidth Estimation, http://www.cs.ucla.edu/classes/fall03/cs218/paper/Balk-ad_mpeg_w_band_est.pdf.

[4] An Enhanced Ethernet Primer, http://www.techrepublic.com/blog/datacenter/an-enhanced-ethernet-primer/3703.

[5] Compass Flow Analyze, http://www.calix.com/compass/flow_analyze/.

[6] DLNA, http://www.dlna.org.

[7] Encoding For Streaming, http://blog.netflix.com/2008/11/encoding-for-streaming.html.

[8] Got HDTV? Get Ready For UHD-TV, http://hothardware.com/News/Got-HDTV-Get-Ready-For-UHDTV/.

[9] H.263, http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=H.263.

[10] How We Got from 1 to 162 million websites on the Internet, http://royal.pingdom.com/2008/04/04/how-we-got-from-1-to-162-million-websites-on-the-internet/.

[11] Internet2 Showcase, http://www.internet2.edu/pastshowcases.cfm?start=281.

[12] ITU, http://www.itu.int/ITU-T/recommendations/index.aspx.

[13] JCT-VC HEVC, http://www.itu.int/en/ITU-T/studygroups/com16/video/Pages/jctvc.aspx.

[14] Manage Video Quality, https://account.netflix.com/HdToggle.

[15] Netflix exec: Canada's broadband caps "almost a human rights violation", http://gigaom.com/video/netflix-canada-caps-human-rights-violation/.

[16] Netflix Lowers Data Usage By 2/3 For Members In Canada, http://blog.netflix.com/2011/03/netflix-lowers-data-usage-by-23-for.html.

[17] Netflix Now The Largest Single Source of Internet Traffic In North America, http://techcrunch.com/2011/05/17/netflix-largest-internet-traffic/.

[18] Netflix Traffic Now Bigger Than BitTorrent. Has Hollywood Won?, http://gigaom.com/2011/05/17/netflix-p2p-traffic/.

[19] Netflix Traffic Plummets Due To Olympics Down By Up To 25 Percent During Games Report, http://www.huffingtonpost.com/2012/08/02/netflix-traffic-olympics_n_1732715.html.

[20] ORBIT, http://www.orbit-lab.org.

[21] Peer-to-peer, http://en.wikipedia.org/wiki/Peer-to-peer.

[22] PlanetLab, http://www.planet-lab.org.

[23] Raspberry Pi, http://www.raspberrypi.org.

[24] Remote Media Immersion, http://infolab.usc.edu/imsc/rmi/.

[25] Roku, http://www.roku.com.

[26] SkyFall Movie Trailer, http://goo.gl/790hF.

[27] The Moving Picture Experts Group, http://mpeg.chiariglione.org/standards.php.

[28] Theora, http://www.theora.org/.

[29] VLC, http://www.videolan.org/vlc/.

[30] Wireshark, http://www.wireshark.org.

[31] David Cleary. What is your access network telling you? *OSP Magazine*, 2012.

[32] Robert Doverspike and Alexandre Gerber. Traffic Types and Growth in Backbone Networks. Technical report, in Proc. of OFC/NFOEC, INVITED PAPER, March 2011.

[33] Jenq-Neng Hwang. Multimedia networking, 2009.

[34] S. Moeritz and K. Diepold. *Understanding MPEG 4: Technology and Business Insights*. Elsevier Science, 2004.

[35] Zhipeng Ouyang. On heterogeneous user demands in peer-to-peer video streaming systems.

[36] L.L. Peterson and B.S. Davie. Computer networks: A system approach. *Communications Magazine, IEEE*, 2012.

[37] Sandvine. Global internet phenomena spotlight. Technical report, Sandvine, May 2012.

[38] Miao Wang. Multi-channel Peer-to-Peer Streaming Systems as Resource Allocation Problems.