Video & Compression over Broadband for Wireless Broadband Network Engineers

by Ken Rahmes

When discussing video traffic across an IP based network, whether wired or wireless, a rule-of-thumb often overheard goes something like this: "For SDTV, a reasonable trade-off between image quality and bandwidth results in a transmission rate of about x Kbps." Especially when video surveillance networks are considered, a video data rate of 5-6 Mbps is often specified in order to accommodate multiple channels/cameras. In an effort to provide an engineer what s/he needs to comprehensively understand each fundamental phase of the process before beginning the design of data links, this paper walks through the entire process of digital video transmission. Beginning with collecting raw video from a digital video camera, subsampling and compressing the data, and encapsulating the data in the many "packages" necessary to transmit it across an IP packet based digital network, an estimate of the capacity required of the network must finally be determined.

Pixels & Lines

A modern camera captures images or pictures using a variety of sensing techniques, all of which result in an array of dots or pixels. Conceptually, one can conceive of each point or pixel in Figure 1 (below) as being represented by a combination of the colors red, green, and blue (RGB). Although light and color theory are not directly relevant to this discussion, the way the RBG signal continuously varies over time is germane. A continuously changing, or analog, signal will obviously need to be sampled if it is going to be transmitted across, and stored in, digital media. Thus, a series of snap shots, called frames, are generated. Figure 1, depicts a frame and some associated terminology.

Typically, the area of the image is rectangular and describing the dimensions of the image can be expressed in terms of columns (x-axis) X rows or lines (y-axis). Accordingly, the array or matrix in Figure 1 is expressed as: 30x14, or 30 by 14, meaning 30 columns and 14 rows. Once each pixel has been converted to a digital format, it can be represented by three numbers that represent the relative intensities of each primary color: red, green, and blue. Transmitting digital video data over a wire is accomplished by reading the three values from the first pixel, then the three values from the next pixel, etc., until the entire frame has been sent. Then, the next frame is transmitted according to the same process, then the next frame, and so forth.

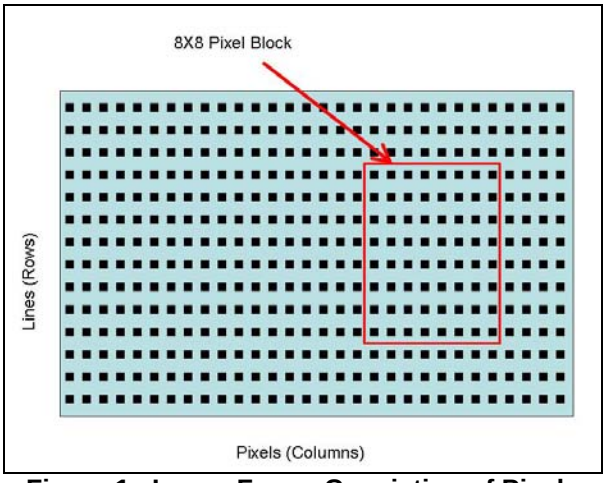


Figure 1. Image Frame Consisting of Pixels Arranged in Lines.

(There are several different ways to accomplish the "pixel reading" described above, only two of which might be encountered by the network engineer: interlaced and progressive video. In a basic sense, interlaced video breaks the pixel array into two subsets of pixels called fields where each field consists of every other line or row in the pixel array. Combining the two fields again reconstitutes the entire frame.

Progressive video just samples the entire array of pixels at once. With progressive video, there are no fields, just a single frame.)

Formats: RGB, YUV, & YCbCr

Numerous formats have been developed to represent the information contained in the RGB signal. Some of these formats can be expressed much more efficiently than others, resulting in faster transmission rates, lower storage requirements, and greater ease of compression. There are a multitude of formats in use to represent the transformed data stream and virtually every one of them is somewhat incompatible with every other! However, the distinctions are irrelevant, since once transformed, all formats transmit identically to a digital network: just a bunch of binary numbers, transmitted across some form of media as bits.

How RGB values are transformed into YUV or YCbCr values, and whether or not the process occurs before or after a frame's pixels are "quantized" into binary numbers requires a complex explanation and is beyond the purposes here. What matters most is that, after all the number crunching, the result is that YUV or YCbCr values are ultimately expressed as three binary numbers. Most often, digital video data contains 24 bits per pixel: 8 bits each for R, G, & B, or Y, Cb, & Cr. Y represents luminance and is capable of presenting a black and white signal all by itself. U (or Cb) and V (or Cr) represent chrominance and present the majority of the color. Since the human eye is much more sensitive to luminance, the chrominance is a candidate for subsampling, that is, discarding some of the information that the eye is not likely to notice as absent.

Subsampling of a frame is most often represented as ratios: 4:2:2 and 4:2:0. The first ratio indicates that the chrominance (Cb & Cr) is horizontally subsampled by a factor of two. That is, every other column in Figure 1 is skipped. The second ratio indicates that every other column and row is skipped. If frames (however unrealistically) sized as in Figure 1, (30 columns x 14 lines) are generated at the rate of 12 per second (12 fps), then the fully sampled, uncompressed bit rate will be:

4:4:4 $[30 \times 14 \times 12 \times 8]_{(Y)} + [30 \times 14 \times 12 \times (8 + 8)]_{(Cb+Cr)} = 40320 + 80640 = 120.96$ Kbps

In the case without subsampling, 30x14 pixels multiplied by 12 fps multiplied by 8 bits of intensity each for Y and Cb and Cr and summed. For the two cases of subsampling, the bit rates that have not yet been compressed are:

4:2:2 $[30 \times 14 \times 12 \times 8]_{(Y)} + [15 \times 14 \times 12 \times (8 + 8)]_{(Cb+Cr)} = 40320 + 40320 = 80.64$ Kbps

4:2:0 $[30 \times 14 \times 12 \times 8]_{(Y)} + [15 \times 7 \times 12 \times (8 + 8)]_{(Cb+Cr)} = 40320 + 20160 = 60.48$ Kbps

In the 4:2:2 case, the 30 columns have been cut in half to 15 for Cb and Cr. In the 4:2:0 case, the 14 lines have also been cut in half to 7. Note that in both cases the luminance (Y) has not been touched! Clearly, 4:2:0 subsampling, in this example, cuts the amount of data being transmitted by half. With a clearer understanding of subsampling, it is advantageous to review some of the standards (and buzz words) commonly found in digital camera specification sheets.

So far, the only three numbers doing all of the leg-work involved in "running the math" are 1) the number of horizontal pixels or columns per line, 2) the number of vertical pixels or rows per line, and 3) the frame rate. However, these values are sometimes not specifically identified in camera data sheets. Rather, various "well known" (camera) industry standards are specified and engineers are expected to know what they mean. The following section will provide some explanation of these industry standards in the context of an overview of video data compression.

Compression

Two modal video formats frequently encountered in industry specification sheets are the Common Intermediate Format (CIF) and MPEG-2/4. CIF is typically used for uncompressed formats, while MPEG-x specifically addresses compressed formats. MPEG-2 is an international standard (ISO/IEC 13818), Parts 1 & 2 of which are identical to ITU-T Rec. H.222.0 and Rec. H.262. MPEG-4 Part 2 (not to be confused with MPEG-4 Part 10) is

similar to MPEG-2 Part 2. In this paper, MPEG-2 Part 2 profiles and levels will be used to introduce some of the common industry terminology.

The Common Intermediate Format (CIF), from the ITU-T H.261 standard, is targeted at YCbCr formatted video with a frame rate of approximate-ly 30 fps and a pixel aspect ratio of about 1.22:1. Table 1 shows the CIF resolutions most often encountered. (Pixels/Line = Columns; Lines = Rows.)

Table 1. Common CIF Resolutions

Designation	Pixels/Line	Lines
SQCIF	128	96
QCIF	176	144
CIF	352	288
4CIF	704	576

Phase Alternating Line (PAL) color encoding is a scheme used in Western Europe while National Television System Committee (NTSC) color encoding is the scheme used in the United States. The differences between the two schemes are a result of the different alternating current (AC) line frequencies of the two regions (60 Hz in the U.S. vs. 50 Hz in Europe). The practical result is that PAL has a top frame rate of 25 fps, while NTSC has a top frame rate of 30 fps. Video cameras are typically capable of operating in accordance with either system.

MPEG-2 Part 2 compresses raw frames into I-frames, P-frames, and B-frames. I-frames are the anchors upon which P- and B-frames depend to achieve high levels of compression. Blocks of 8x8 pixels are run through a discrete cosine transform, redundancies in the resulting coefficients are removed, something called Huffman coding is applied, and the resulting array (or series) of numbers is transmitted across whatever media is being used. The construction of a P-frame uses 16x16 pixel blocks to search for “differences” between the previous reference frame (I- or P-frame) and the current frame. Motion vectors to account for any changes are then created. B-frames work similarly, except they use information from both the previous and following reference frames. In MPEG-2, these frames are sequenced sort of like IBBPBBPBBPBB. Such a sequence, depending on a single I-frame, is called a Group Of Pictures (GOP). Although the process is much more complex and subtle, and MPEG-4 is even more so, that’s the general idea.

Both MPEG-2 and MPEG-4 define profiles and levels in order to specify particular subsets of the complete standard. Table 2 contains some of the most important capabilities and limitations of MPEG-2 profiles and Table 3 presents similar information for MPEG-2 levels.

Table 2. MPEG-2 Profile Characteristics

Profile Name	Chrominance Ratio	Allowable Frames
Simple (SP)	4:2:0	I, P
Main (MP)	4:2:0	I, P, B
SNR Scalable (SNR)	4:2:0	I, P, B
Spatially Scalable (Spatial)	4:2:0	I, P, B
High (HP)	4:2:0 and 4:2:2	I, P, B

Table 3. MPEG-2 Level Characteristics

Level Name	Max Resolution	Some Frame Rates (fps)	Max Bit Rate (Mbps)
Low (LL)	352x288	24, 25, 30	4
Main (ML)	720x576	24, 25, 30	15
High 1440 (H-14)	1440x1152	24, 25, 30, 50, 60	60

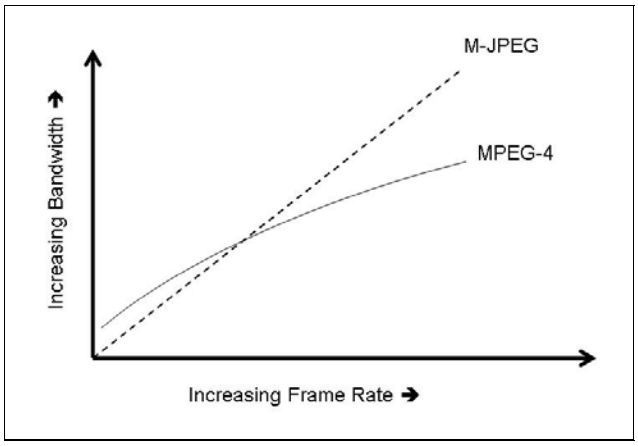
High (HL)	1920x1152	24, 25, 30, 50, 60	80

The profile defines features of the video signal, while the level constrains various quantitative characteristics of the signal. In the literature, terminology such as MP@ML is common. What matters in terms of bit rates is the chrominance ratio from Table 2, the resolutions and frame rates from Table 3, and of course, the compression achieved by the digital cosine transformation. In Table 3, the maximum bit rates are just that: the maximum allowable by the standard level (most manufacturers' chips will produce much lower data rates).

Another common format is Motion JPEG (M-JPEG). The computer resources required to process JPEG files are much lower than those needed by the various MPEG alternatives. Especially in surveillance applications, where frame rates are often low, the high quality of JPEG images combined with the low cost, simplicity, and wide availability of software capable of processing them make M-JPEG quite competitive.

Figure 2 shows the trade-off between Motion-JPEG and MPEG-4. As the frame rate increases, the bandwidth required by MPEG-4 decreases dramatically with respect to motion JPEG. However, if the frame rate is relatively low, M-JPEG can be quite competitive. Where is the crossover point? The answer depends on a large number of variables and extends beyond the current scope.

Figure 2. Motion JPEG Compared With MPEG-4.



It is important to come to terms with what can be expected from a modern digital camera in terms of compression ratios. The Axis Communications 233D (NTSC) PTZ Network Dome Camera with a resolution of 4CIF at 30 fps and with a compression ratio of 30% provides a suitable example. Axis Communications has an on-line design tool that allows engineers to estimate the bandwidth required for a camera operating in a particular mode and scenario.

Figure 3 shows a typical calculation for a camera at a busy intersection with MPEG-4, Part 2 compression.

Figure 4 shows the same situation with Motion JPEG compression.

Note the bit rate differences: MPEG-4 has an estimated bit rate of 1.54 Mbps, while M-JPEG has an estimated bit rate of 6.64 Mbps. Obviously, capacity planning must take this difference into account when estimating average and peak data rates. Industry rules of thumb generally recommend MJPEG for storage and MPEG-4 for viewing. However, according to Axis Communications support, an MPEG-4 compression ratio of 20-30% is reasonably close to MJPEG and, for many applications, is a good compromise between quality and bandwidth. In the following two figures, a 30% compression ratio has been used for both cases.

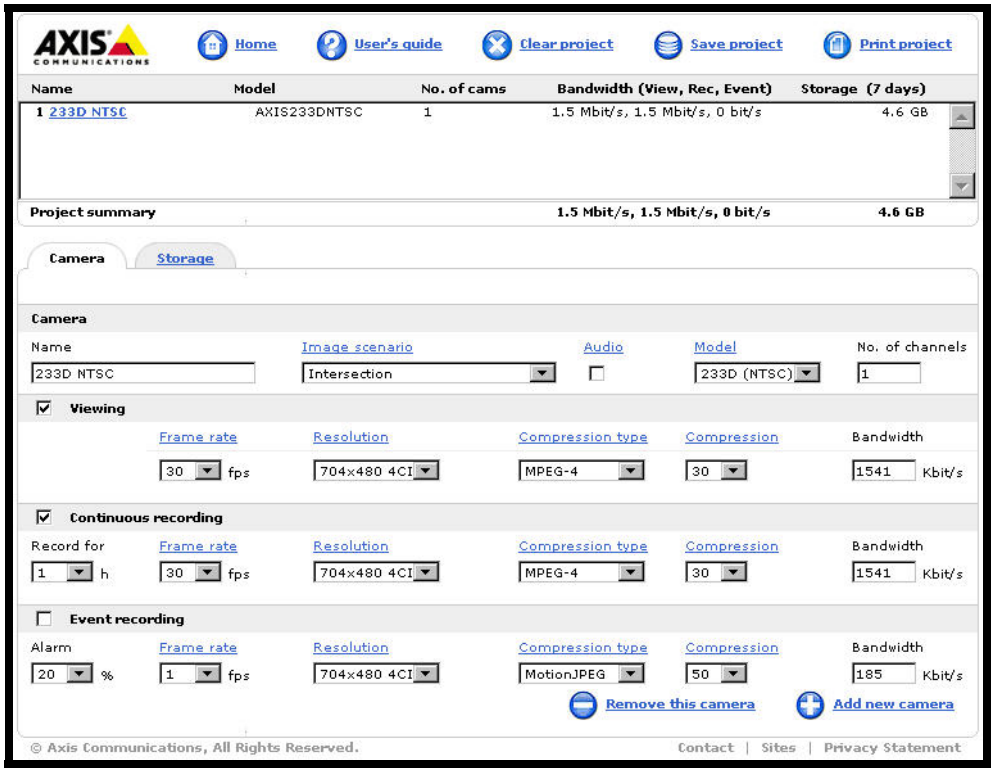


Figure 3. MPEG-4 Compression BW for the Axis 233D.

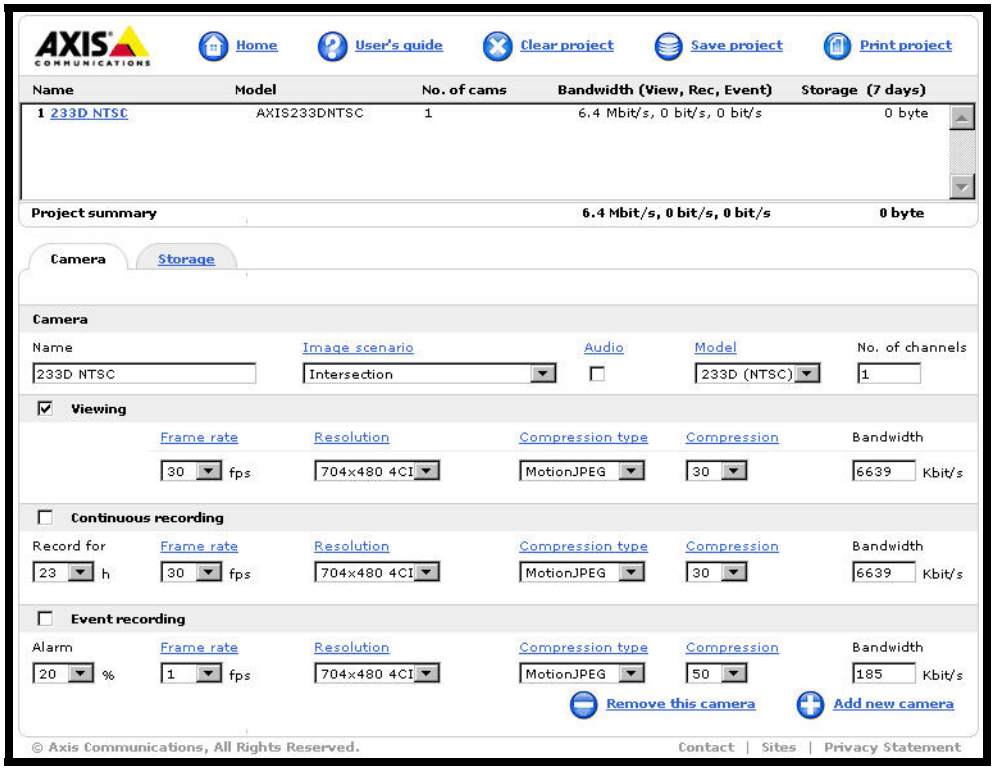


Figure 4. M-JPEG Compression BW for the Axis 233D.

The 4CIF resolution reflects 704x480 pixels per frame, and assuming 4:2:2 subsampling, the result is:

704 x 480 x (8) + 352 x 480 x (8 + 8) = 5.407 Mbits/frame.

Then, assuming a fairly detailed scene, the numbers might look something more like this:

5.407 Mbits @ 30% MPEG-4 compression = 51.367 kbits/frame.

(How was that 51.367 kbits/frame obtained from the 5.407 Mbits? Who knows? It's probably an Axis Communications proprietary digital cosine transformation and they're not about to give it up. Every manufacturer will have their compression secrets and somewhat different output video data rates. As the image scenario selected in the calculator is "Intersection," it can be assumed that there is both considerable image complexity and motion.)

Given a frame rate of 30 fps, the result is an overall bit rate of:

51.367 kbits/frame x 30 fps = 1.541 Mbps.

As may be evident, the results are derived from the Axis Communications on-line calculator (1.541 Mbps) and some arbitrary assumptions have been made. But what is important here is the process rather than the specific numbers. If one wants to determine the network capacity required for a particular camera or camera array, only a few simple computations using the manufacturer's calculator are necessary in order to obtain a reasonable estimate. The design engineer must keep in mind the high level of variance inherent in such estimates and recognize that the best results are necessarily approximate.

802.3 Encapsulation

The transmission of data across a packetized data link is a complex endeavor. Symbols represent bits that are grouped into frames, packets, segments/datagrams, and finally data. This process of encapsulation is typically referenced using the OSI Model and/or the TCP/IP Model. Table 4 presents both models, the associated terminology, and the layers for reference. It takes many layers of overhead for a compressed video bit stream to find its way across an IP network. Figure 5 shows a typical 802.3 MAC frame with the encapsulated data as just one of a virtually endless number of possible scenarios.

Table 4. The OSI & TCP/IP Models.

OSI Designation	Layer	Data Unit	TCP/IP Designation	Layer	Examples
Application	7	Data	Application (Data, Voice, Video)	5	DHCP, DNS, FTP, HTTP, RTP, L2TP
Presentation	6	Data			
Session	5	Data			
Transport	4	Segment	Transport	4	TCP, UDP
Network (IP)	3	Packet	Network (IP)	3	IP, BGP, IPsec, ARP, RIP
Data Link (MAC)	2	Frame	Data Link (MAC)	2	802.11/16, Ethernet, PPP
Physical (PHY)	1	Bits / Symbols	Physical (PHY)	1	Bits, Symbols

As shown in Figure 5, the video data is first incorporated into a Real-time Transport Protocol (RTP) packet at the TCP/IP Application layer. The RTP packet header contains such information as the format of the data, a sequence number, and timestamp. The RTP packet header is 16 bytes and in turn is wrapped in a User Datagram Protocol (UDP) segment or datagram. The UDP header is 8 bytes long. The principal advantage to using UDP datagrams is their simplicity and low overhead. UDP does not ensure delivery of a datagram. If a datagram is lost, oh well,

too bad. While the video or voice data may be lost, the end effect will probably be minimal. The point is: it is faster than TCP-like protocols that do ensure datagram delivery. The next step is the IPv4 packet encapsulation (20 bytes), or IPsec. An LLC header is often included in video protocols (8 bytes) and VLAN tags (4 bytes) may be a part of the mix. Finally, the MAC header and Cyclic Redundancy Check (CRC) checksum encapsulate everything (18 bytes).

Given the scenario of 1.541 Mbps of MPEG-4 compressed video and a "payload" of 1,456 bytes per RTP packet (Figure 3), about 132 packets per second are required to transport the data across an Ethernet network. Given the sum of the overhead per packet of 74 bytes, then the overhead in bits is 74 x 8 = 592 bits/packet. Then, 1456 + 592 = 2,048 bits/packet. So the grand total of bits per second to be transmitted over Ethernet II is:

592 bits/packet x 132 packets/second + 1,541,000 bps = 1,619,144 or approx. 1.62 Mbps.

This figure is a peak rate, of course, after all the protocol setup and handshaking has taken place and data is screaming across the Ether(net). It also does not take into account the physical layer (Layer 1) which will add a preamble and start frame delimiter (SFD) totaling 8 bytes.

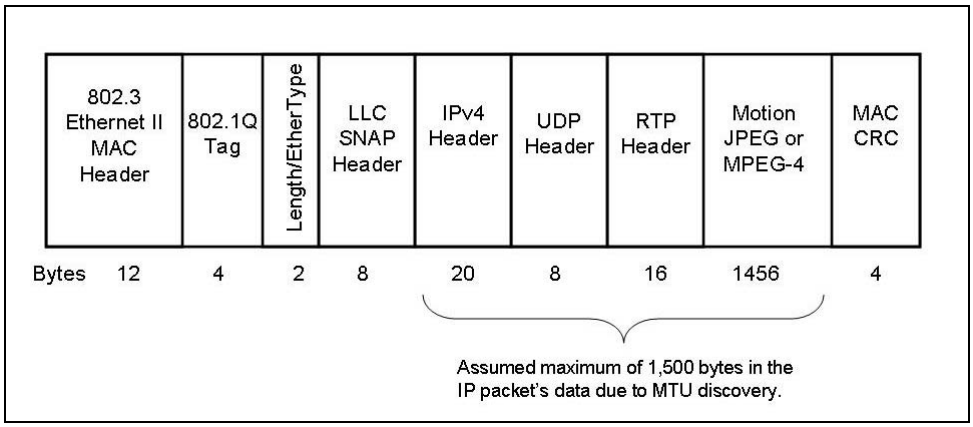


Figure 5. Ethernet Video Data Encapsulation.

802.11 Encapsulation

Data is fed to a wireless access point (AP) at the rate of about 1.62 Mbps in this scenario. What does the AP do with that data and what is the data rate required when transmitting through the mesh network?

Any router (an AP can be thought of as a wireless router) will strip off the MAC header, either remove, or more likely pass through the 802.1Q tag, and resolve the IPv4 address with the addresses in its routing table. The destination IP address will be located across the mesh network on the other side of a gateway radio or POP. While many possibilities exist, one such possibility is that nothing changes in the payload and that the only variation occurs at TCP/IP Layer 2, the MAC frame.

Instead of an Ethernet MAC, an 802.11a MAC frame header is prepended to the data and the 802.3 MAC CRC is replaced by an 802.11a Frame Check Sequence (FCS). FCS is also generally referred to as Forware Error Correction (FEC). The FCS is essentially the same as the 802.3 CRC checksum, but it has been modified to incorporate the changes made to the MAC header. The only real change from the perspective of data rates is that an 802.11 MAC header is 24-30 bytes, while the 802.3 MAC header is 12 bytes. So for each frame, 18 bytes need to be added: 8 bpB x 18 B x 132 fps = 19,008 bps. If 19.008 kbps are added to 1.62 Mbps, that gives a rough bit rate for the mesh network of 1.64 Mbps. Not a huge difference! (Just as with the Ethernet physical layer, 802.11 requires some housekeeping. In this case, it is a PLCP header and a preamble. But, when a radio manufacturer specifies a particular data rate, these PMD level issues are not included so they are not relevant to the issue of network capacity.)

In addition to the Frame Check Sequence (FCS), mentioned above, 802.11a adds a certain amount of error handling capability at the physical level, Layer 1 of the OSI and TCP/IP models. This feature is commonly referred to as Forward Error Correction (FEC). Typical ratios are 1/2, 2/3, and 3/4, where the numerator represents the number of data bits transmitted and the denominator represents the total number of bits including the FEC. Thus, for every three data bits, one additional FEC bit is included when a 3/4 FEC rate is specified. At maximum capacity – 64-QAM with 3/4 FEC – a single OFDM subchannel moves 1.125 Mbps of data. Each link has 48 data subchannels, producing 54 Mbps. Dividing by 1.64 Mbps, the estimate of the video data rate for the Axis 233D camera determined above, results in about 33 cameras transmitting MPEG-4 data simultaneously!

Does this make sense? Is this reality? Actually, no. It turns out empirically, with the error rates actually encountered (bit errors, dropped packets, etc.), that an optimistic throughput is something less than half that 54 Mbps, about 25 Mbps! This figure translates to something on the order of 12-15 cameras with MPEG-4 capability. With M-JPEG, the bit rate is 6.64 Mbps, so approximately 4 cameras are possible in the best case. These figures are all, of course, estimated "maximum" rates across a point-to-point data link! When video data is sent across a mesh network, it can degrade significantly. So engineer, beware, and as always: test, test, test.

Figure 6, shows the overall situation with a surveillance camera transmitting an 802.3/Ethernet II frame to an access point (AP) on the wireless broadband network via an Ethernet cable (usually a CAT5 cable outdoors). The AP then transmits an 802.11a frame wirelessly from the AP to another AP on the mesh network. The AP to AP transmissions continue until the final gateway AP is reached. Then, the gateway AP converts the 802.11a frame into an 802.3/Ethernet II frame and transmits it to the POP (Internet).

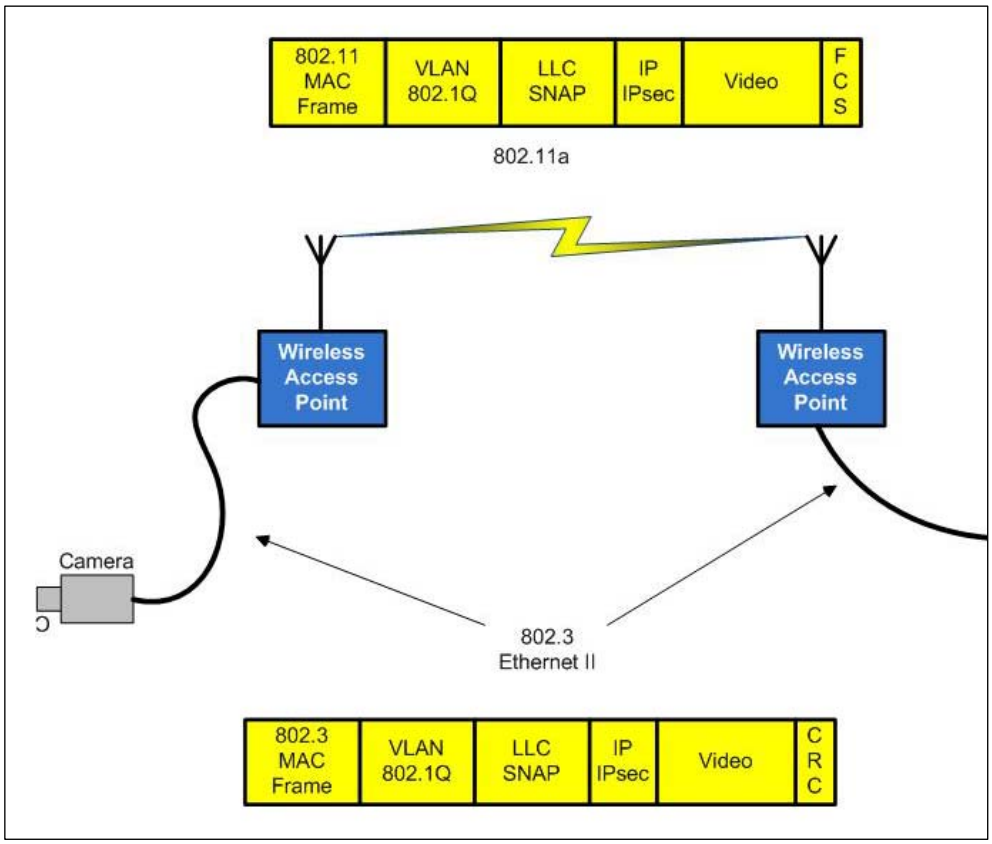


Figure 6. TCP/IP Over Two Mediums.

An AP-to-AP wireless MAC frame requires four IP addresses (see Figure 7 below). The first and second addresses are those of the receiver and transmitter on the wireless network. Then, the destination and source IP addresses follow as shown. MAC data frames in the 802.11 standard can accommodate up to 2312 bytes of data, considerably more than the 1500 bytes of the 802.3/Ethernet II standard. Unfortunately, the wireless network cannot concatenate multiple 802.3/Ethernet II frames! So, the amount of compressed video per frame remains the same.

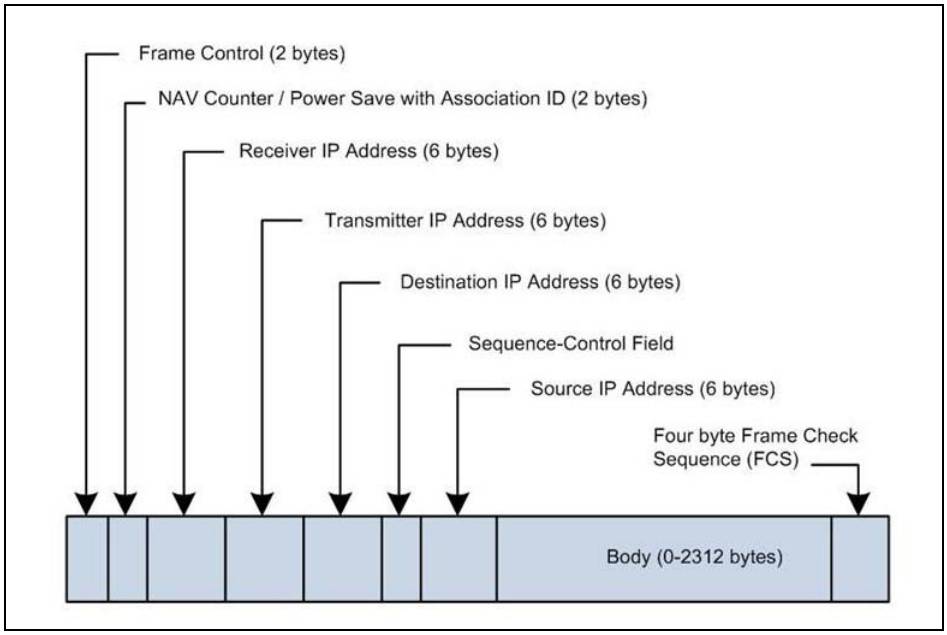


Figure 7. 802.11 Data Frame - AP to AP.

References

Axis Communications. (2008). H.264 video compression standard: New possibilities within video surveillance. Retrieved on 4/21/2008 from http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf.

Axis Communications. (n.d.). Compression standards. Retrieved on 4/15/2008 from http://www.axis.com/products/video/about_networkvideo/.compression.htm.

Proxim Wireless. (2007). Wireless solutions for security and surveillance. Retrieved from http://www.proxim.com/downloads/whitepapers/Proxim_SecuritySurveillanceWP.pdf on 4/17/2008.

Tudor, P.N. (1995). MPEG-2 Video compression. *Electronics & communication engineering journal.* Retrieved from http://www.bbc.co.uk.rd/pubs/papers/paper_14/paper_14.shtml, on 4/16/2008.

Wikipedia, the free encyclopedia. (2008). MPEG-2. Retrieved on 4/15/2008 from http://en.wikipedia.org/wiki/MPEG-2.