X-MOVIE:
Transmission and Presentation of Digital Movies under X

Bernd Lamparter   Wolfgang Effelsberg

Praktische Informatik IV

University of Mannheim

lamparter@pi4.informatik.uni-mannheim.dbp.de

Abstract

We describe a system for storing, transmitting and presenting digital films in a computer network. The hardware used in the system is standard hardware, as found in typical workstations today; no special hardware is required. The movies are shown in a window of the X Window System. This allows full integration with the classical components of computer applications such as text, color graphics, menues and icons. The X-MOVIE system is based on color lookup table technology. We present a new algorithm for the gradual adaptation of the color lookup table during the presentation of the film.

1   Introduction

Modern workstations have fast processors, high resolution color graphics displays and high performance network adapters. For all three components a considerable performance improvement can be expected for the next years. Such a powerful hardware scenario enables new computer applications. Perhaps the most important one is the integration of digital films into a classical data processing environment.

For many years, computers have been used for the processing of digital films. The most important directions in research and development can be classified into three categories: computer animation [LMS91], computer simulation and digital video [LD87, MD89]. This paper concentrates on the transmission and presentation of sequences of digital images in a distributed system of standard UNIX workstations and servers. We present an experimental system called X-MOVIE which is operational at the University of Mannheim. Our goal is the presentation of digital movies in a window of the X-Window-System under UNIX [Jon88, SGN88]. Unlike other systems reported in the literature the X-MOVIE system does not require special hardware.

In Chapter 2 of the paper we describe the architecture of the X-MOVIE system. Chapter 3 classifies formats of digital still images and digital movies. In Chapter 4 we concentrate on the network requirements for the transmission of digital movies.

The integration of a movie window into the X-Window-System is described in Chapter 5. We report first experiences with our implementation in Chapter 6. Chapter 7 concludes the paper.

2   Architecture of the X-MOVIE System

The X-MOVIE System is a distributed test bed for integrated transmission and presentation of digital movies. It consists of interconnected UNIX workstations. In our current implementation, the network is a standard Ethernet. An FDDI-ring has been installed recently, and X-MOVIE is currently being ported to FDDI. The architecture of the system is shown in Figure 1.

The main components of X-MOVIE are the Movie Server, the Movie Client and the X-Client. These three components can run on one, two or three different UNIX systems depending on the requirements of the application.

The **Movie Server** is able to store and replay sequences of digital images (digital films). It maintains a movie directory. On request of the Movie Client a sequence of images is sent over the network to the Movie Client. The transmission protocol is called MTP (Movie Transmission Protocol). It was developed specifically for this purpose. The **Movie Client** is an extension of the standard X-Server of the X-Window-System. The extensions implement a new set of functions for the purpose of displaying movies in a window on the screen. Examples of new functions are XMOpenMovie, XMPlayMovie, and XMShowSinglePicture. These extensions were integrated into the source code of the X-Window-System. The third component of the system is the **X-Client**. Similar to the X-Server, it is an extention of the standard X-Client. The set of new functions mentioned above was integrated into the Xlib function library of the X-Client so that a programmer can now invoke the new movie functions just like other Xlib functions.

In this paper we concentrate on the transmission and presentation of digital movies; we do not discuss the creation of such movies here. Digitized video films can be used as well as computer generated digital movies.

3   Formats for Digital Images and Movies

3.1   Formats for Digital Images

An image digitized with high quality ("true colors") uses 24 bits of storage per pixel (one byte for each of the color components red, green and blue). About 16 million different colors can occur in one image. Experience shows that the human eye cannot recognize smaller color differences [Lut89]. However, "true color" representation needs too much storage to be practical (720 kbytes for a 600×400 pixel picture), so image compression is needed.

A very widely used form of image compression is the color lookup table technology. The graphics adapters of PC's and workstations can typically display 256 different colors at a time. The color components for each color are stored in a color lookup table (or "color map") with 256 entries. Each pixel of an image is represented by one byte which is an index into the color lookup table (CLUT) (see Figure 2).

For a specific image the colors in the CLUT can be optimized by choosing the most appropriate 256 colors out of a set of 16 million. So only 256 colors can

be presented at one time, but out of a large set of colors. With each image the corresponding CLUT must be stored. The color lookup table reduces the amount of data to one third without much loss of quality.

3.2 Compression of Images

Even with color lookup tables images are still very large, and further compression is desirable. We distinguish between lossless and lossy compression. Lossless compression implies that the decompressed image data is an exact copy of the orignal image data. Lossy compression allows small differences, and usually provides higher compression factors.

Compression algorithms can also be classified according to their time behavior. Symmetric algorithms use the same amount of time for compression and decompression whereas asymmetric algorithms use more time for compression than for decompression. Asymmetric algorithms are often useful for applications where compression can be done off-line, and the compressed images are stored for later use. Decompression is often done much more frequently, and can be done in real-time.

Image representation with a CLUT can be considered as a simple, lossy compression. Decompression is done in real-time in the graphics adapter. More sophisticated compression needs more computing power for decompression. This computing power can be provided with the following hardware approaches:

- Use a faster main processor
- Provide special hardware for image decompression in the workstation or in the color graphics adapter
- Use multiple parallel processors, i. e. transputers.

An interesting compression scheme for still images is the Color Cell Compression algorithm (CCC, [CDF$^+$86]; an extension can be found in [Pin90]). CCC compresses to 2 bits/pixel. Because of the blockwise nature of the algorithm, compression and decompression can be parallelized well. The algorithm is lossy and symmetric.

ISO is standardizing schemes for compression of both still images and movies [Gal91]: JPEG (Joint Photographic Expert Group) and MPEG (Motion Picture Expert Group). Both algorithms are lossy and symmetric. The compression factors are very high due to the Discrete Cosine Transformation step, but compression and decompression require considerable computing power. Therefore the current trend is to build special compression chips. JPEG chips are already available in the market, and MPEG chips are expected soon.

3.3 Formats for Digital Movies

A digital movie can be stored as a sequence of images, either with 24 bits per pixel or with 8 bits per pixel (as indices into a CLUT). In order to provide the impression of continuous movement, at least 25 images per second must be shown.

3.3.1 Compression for Movies

The storage requirement is already a problem with still images, and a much greater problem with movies. Fortunately new possibilities for compression arise for movies. Due to the fact that the color difference of a pixel over time is often small, differential

pixel coding can be used. Two-dimensional compression algorithms can be extended into three-dimensional ones, with time as the third dimension. Another simple scheme is to compute the difference image between two frames and then compress the difference image.

3.3.2 Movies and Color Lookup Tables

When showing a movie as a sequence of still images, each coded with a CLUT, the following problem arises: If two subsequent frames are coded with independent color lookup tables the CLUT of the second image has to be loaded into the adapter card just before the pixels of the second image. This has the effect that the first image is visible in the colors of the second image for a short time. This is visually very disturbing, even if the images have similar colors, because similar colors do not necessarily use the same CLUT entries. The problem is solved if all images of the whole movie are coded for the same CLUT. However, the whole move will then only have 256 different colors. Although 256 colors are sufficient for a single image, with an optimized CLUT, they are not sufficient for a long sequence of images, perhaps starting with a blue sky and white clouds and ending in a room with green and brown interior. A new solution must be found for showing movies with color lookup table technology.

We propose to use the following algorithm to solve this problem: For each image some of the 256 CLUT entries are not used. For example 32 CLUT entries are left free. Before loading the next image we can now load 32 new colors into the CLUT without changing the colors of the current image. The second image can use all of the old colors and 32 new colors. It will again leave 32 entries free for its successor. This leads to continous updating of the CLUT while the movie is showing (see Figure 3). In fact the amount of free entries can vary from image to image depending on the actual difference in colors between the images. From our experience the loss of a few out of 256 colors has no visible effect at a rate of 25 frames/s. This algorithm was implemented successfully in the X-MOVIE system.

4   Transmission Protocols for Images and Movies

In a UNIX environment the first attempt with network protocols will always be the Internet protocols TCP/IP [Com88].

IP is the protocol of the Network Layer (layer 3 in ISO/OSI); it offers a connectionless packet transfer between hosts. The protocol can loose packets and deliver packets out of sequence. Bit errors are detected, and the faulty packet is discarded.

TCP offers a reliable connection-oriented transport service over IP. It uses standard technology for error detection and recovery (sequence numbers and retransmission with go-back-n), and for flow control (sliding window).

A third protocol in the Internet ist the user datagram protocol (UDP). Basically it offers a programming interface for IP. Most of the applications in the Internet use TCP, some use UDP (see Table 1).

4.1 The Movie Transmission Protocol MTP

In the X-MOVIE environment a movie consists of a header, an initial CLUT and a sequence of images with their CLUT updates. The Movie Server operates on such movie data structures. On request from the Movie Client (= X Window Server) it will provide the following services:

- Store/retrieve information about a movie
- Play
- Stop
- Step forward/step backward
- Show picture $< n >$
- Slower/faster

If Movie Server and Movie Client run on different systems, a Movie Transmission Protocol (MTP) is required to transfer the sequence of images. The MTP is an application layer protocol in OSI terminology. The transmission of MTP Protocol Data Units must be accomplished by mapping them to the services of the transport subsystem. Unfortunately the transport subsystems available today are not appropriate for movie transmission because they do not provide **isochronous data flows** (sometimes called Continuous Media Data Streams, [Her91]).

4.2 Isochronous Data Flow

The term "isochronous" means a data flow between end systems with minimal delay jitter. It is not the delay as such causing the problem, but the variance in delay between subsequent packets. Two major obstacles prevent isochronous data flows: The operating systems of workstations and hosts cannot handle data traffic in real-time, and today's network protocols cause variable delays.

In order to solve the operating system problem the OS scheduler must be modified to take the requirements of real-time processing into account. Early work on this topic can be found in [ADH91, ATW+90].

As far as the network protocols are concerned, the problems are much more difficult to solve because almost all layers introduce variable delays. It starts out with media access protocols in LANs. For example, in a CSMA/CD-based protocol, carrier sensing can lead to variable delays, depending on traffic from other stations on the bus. Also, collisions can delay packet transmission in an unpredictable way. Similarly, a Token Ring LAN will have variable packet delays depending on where the token happens to be when transmission is requested, and how many priority stations are waiting for transmission. The same is true for the token-based media access control protocol of the high-speed LAN FDDI.

Conventional transport protocols are not suitable for isochronous transmission either. Both TCP and ISO TP4 use time-outs and retransmission for error correction. Of course retransmitted packets will have a longer delay than first-shot packets. Fortunately the new high-speed networks now under development, such as DQDB or Broadband ISDN based on ATM, will provide protocols for isochronous data flow [Kil91, Fun91]. In parallel new high-speed transport protocols are under development [HS91].

4.3 MTP Design Issues

For our X-MOVIE test bed we have tried to use existing transport protocols and networks for movie transmission in order to gain early experience. A Movie Transmission Protocol MTP was developed and tested over the Internet protocols. MTP is a hybrid protocol: some transmissions are reliable (i.e. acknowledged), others are unreliable. All data packets from the Movie Client to the Movie Server (such as "stop") are acknowledged. Control packets from the Movie Server to the Movie Client are also acknowledged, but image transfers are not. They require continuous, fast transmission. Data loss for a part of an image is not critical since an update will follow after 1/25 s.

In conventional transport protocols flow control techniques such as "Sliding Window" are used to prevent the receiver from being overrun by a faster sender. In isochronous transmission window-based flow control is inappropriate, and rate-based flow control must be applied. Sender and receiver have to work at the same data rate, they have to be synchronized [Ste90]. Movies arriving at the display from a network have to be synchronized with real-time. Hence a synchronisation mechanism between Movie Client, Movie Server and real-time must be implemented. Synchronization is not yet implemented in MTP.

5 Integration of Movies into a Window System

For the workstation user the movie should appear in a window on the screen, without disturbing other windows. The handling of the movie window should be similar to the handling of other windows. Movie windows are not yet supported in current window systems, such as the X Window System. An integration of movie windows can be done in the following ways:

- Hardware can be added (blue-box or realtime digitizing of analog video sources) [Bru89, Lut89]

- The window software is accelerated so that a sequence of still images can be passed to it, and it will draw them at 25 images per second

- The window system is extended by a movie service.

The goal of the X-MOVIE project is to avoid special hardware; hence we only elaborate on the other two possibilities.

5.1 Fast Transmission of Single Frames

Each window system has a mechanism to draw raster images, so in principle we can show movies by drawing images fast, one after the other [Lof90]. But each image causes considerable overhead because it invokes a large number of subroutines of the window system until it is finally presented. Also the programmer of the client has to control the flow of images explicitly. Therefore we considered this solution to be unacceptable.

5.2 Movie Service as an Extension to the Window System

A more promising approach is the integration of a movie window as a primitive of the window system. In case of X this implies full control of the movie window by the X Server process. The X Server process manages the connection establishment and release to the Movie Server and controls the movie transmission. We see many advantages compared to the solution above:

- The communication overhead between X Client and X Server is lower

- The code for movie presentation (i.e. presentation of sequences of images) does not have to be repeated in each client program

- The X server can be better adapted to the hardware, so the performance and hence the movie quality is better.

- The manipulation of the movie window (open, close, zoom, ...) works as usual (for the user and for the programmer).

The prototype of the X-Movie System is based on the latter solution. The subroutine library of the X Window System (Xlib) was extended to provide the following new functions [Kel91]:

XMListMovies(): Requests the X-Server to send a list of stored movies and images

XMFreeMovieInfo(): Frees the memory allocated with **XMListMovies**

XMOpenMovie(): Opens a movie connection

XMPlayMovie(): Starts a movie in a window

XMShowSinglePicture(): Shows a single frame of a movie

XMStopMovie(): Stops a running movie

XMDestroyMovie(): Frees all resources allocated for the movie (memory, communication channels)

6 Experience

The X-MOVIE system was implemented on IBM PS/2 Model 80 workstations under the AIX Operating System. All three components, the Movie Server, the Movie Client and the extended X Client, are now operational on PS/2s. The Movie Server was also ported to a DECSystem 5400 under ULTRIX. The PS/2s have 8514A color graphics adapters. In the current configuration the network is an Ethernet.

In our first experiments we used the TCP/IP protocols for digital film transmission. Our measurements showed that the throughput was only about 150 kbit/s. This was much lower than expected. The low transmission rate is mainly due to some of the TCP protocol mechanisms. One example is the Sliding Window flow control mechanism which adapts very slowly to the differences in speed of two participating systems. In our next experiment we used the UDP Protocol over IP. Here we could measure a throughput of approx. 2 Mbit/s. UDP has no error recovery, and approximately 10% of the packets were lost in the network. In a third experiment we used UDP and IP over a 16 Mbit/s IBM Token Ring Network. All

three X-MOVIE components ran on PS/2 systems. The throughputs measured were approximately the same as with Ethernet.

In a fourth experiment all components were installed on the same workstation. Now no real time parallelism between the different processes was possible. The frequent process switches between the UNIX processes for the three components created a processor bottleneck, and the data rate between Movie Server and Movie Client dropped to below 1 Mbit/s [Sut90, Zim90].

The color graphics adapter 8514A has a color lookup table with 256 entries (8 bits per pixel). Our experiments were done with two digital films: Film 1 had an image size of 128 by 128 pixels, film 2 had 220 by 220 pixels. The films were dithered (converted to color lookup table technology) off-line and stored on the disk of the Movie Server. Film 1 could be shown at 10 frames/s, film 2 at 3 frames/s.

In a more detailed performance analysis we investigated the chain of components in the movie data path: the magnetic disk, the main memory, the bus and the network adapter card of the Movie Server; the network adapter card of the Movie Client (= X Server), the bus, the main memory and the graphics adapter of the Movie Client. Our results were that in all configurations the network adapter cards and network transmission protocols were the bottleneck of the system.

7   Conclusions and Outlook

The X-MOVIE system is a test bed for digital image transmission and presentation in a network of UNIX workstations. It is entirely based on standard hardware and uses standard network technology and standard graphics adapters with color lookup tables.

We have presented an algorithm for the continuous adaptation of a color lookup table for digital movies. It allows the presentation of films with good color quality using standard color lookup table graphics adapters.

Experience has shown that standard Ethernet technology and TCP/IP protocols are not suitable for movie transmission. In particular the Sliding Window flow control and the Go-Back-n error recovery scheme in TCP are inappropriate. The transmission of digital movies will require new protocols which are at the same time isochronous, i.e. have a very low delay jitter, and provide high throughput.

For the presentation of movies on the workstation screen, extentions to the X-Window-System were presented. It was shown that these extensions can be implemented and used without major problems. First experience with X-MOVIE has shown that the presentation of digital movies in a network of standard UNIX workstations is feasible.

Recently we have installed RISC workstations and an FDDI ring, and we are currently porting the X-MOVIE components to this new environment. We expect that films with image sizes of 256 by 256 pixels can then be shown at 25 frames/s. In the long run we intend to integrate JPEG and MPEG compression chips into the system.

Acknowledgment

We wish to thank Prof. A. Schmitt, W. Leister, and A. Stößer from the Institut of Operating and Dialog Systems of the University of Karlsruhe from where we got film 1 (generated with the VERA Raytracing System).

References

[ADH91]   D.P. Anderson, L. Delgrossi, and R.G. Herrtwich. Process Structure and Scheduling in Real-Time Protocol Implementations. In *Kommunikation in Verteilten Systemen,* Mannheim. Informatik-Fachberichte, Springer-Verlag, 1991.

[ATW⁺90] D.P. Anderson, S. Tzou, R. Wahbe, R. Govindan, and M. Andrews. Support for Continous Media in the DASH System. In *10th International Conference on Distributed Computer Systems* Paris, May 1990.

[Bru89]   T. Brunhoff. *VEX: Video Extension to X, Version 5.5.* Textronix, Inc., 1989.

[CDF⁺86] G. Campbell, T.A. DeFanti, J. Frederikson, S.A. Joyce, A.L. Lawrence, J.A. Lindberg, and D.J. Sandin. Two bit/pixel full color encoding. *Computer Graphics,* 1986.

[Com88]   D. Comer. *Internetworking with TCP/IP.* Prentice-Hall International Editions, 1988.

[Fun91]   Oswald Fundneider. Breitband-ISDN auf Basis ATM: Das zukünftige Netz für jede Bitrate, 1991. In *Kommunikation in Verteilten Systemen,* Mannheim. Informatik-Fachberichte, Springer-Verlag, 1991.

[Gal91]   Didier Le Gall. MPEG:A Video Compression Standard for Multimedia Applications. *Communications of the ACM,* 34(4):46–58, 1991.

[Her91]   R.G. Herrtwich. Zeitkritische Datenströme in verteilten Multimedia-Systemen. In *Kommunikation in Verteilten Systemen,* Mannheim. Informatik-Fachberichte, Springer-Verlag, 1991.

[HS91]   L. Henckel and H. Stüttgen. Transportdienste in Breitbandnetzen, 1991. In *Kommunikation in Verteilten Systemen,* Mannheim. Informatik-Fachberichte, Springer-Verlag, 1991.

[Jon88]   O. Jones. *Introduction to the X-Window-System.* Prentice Hall, 1988.

[Kel91]   R. Keller. Erweiterung des X11-Servers zur Bewegtbilddarstellung. Master's thesis, Universität Mannheim, 1991.

[Kil91]   U. Killat. B-ISDN und MAN: Konkurrierende Netztechnologien. In *Kommunikation in Verteilten Systemen (Tutorial),* Mannheim. Deutsche Informatik-Akademie, 1991.

[LD87]   L. F. Ludwig and D. F. Dunn. Laboratory for Emulation and Study of Intergrated and Coordinated Media Communication. *Proc. of ACM SIGCOMM 87,* Stowe, Vermont, pages 283–291, Aug. 1987.

[LMS91]   W. Leister, H. Müller, and A. Stößer. *Fotorealistische Computeranimation.* Springer-Verlag, (Heidelberg, Berlin), 1991.

[Lof90]   G. Loff. Konzeption, Entwurf und Implementierung eines netzweiten Videodienstes. Master's thesis, Universität Karlsruhe, 1990.

[Lut89]   A. Luther. *Digital Video in the PC Environment.* McGraw Hill Book Company, New York, 1989.

[MD89]   W. E. Mackay and G. Davenport. Virtual Video Editing in Interactive Multimedia Applications. *Communications of the ACM*, 32(7):802–810, Juli 1989.

[Pin90]   M. Pins. *Analyse und Auswahl von Algorithmen zur Datenkompression unter besonderer Berücksichtigung von Bildern und Bildfolgen.* PhD thesis, Universität Karlsruhe, 1990.

[SGN88]   R. W. Scheiffler, J. Gettys, and R. Newman. *X-Window-System, C-Library and Protocol Reference.* Murray Printing Company, 1988.

[Ste90]   R. Steinmetz. Synchronisation Properties in Multimedia Systems. *IEEE Journal on Selected Areas in Communications*, 8(3):401–412, 1990.

[Sut90]   S. Sutter. *Datenformate bei der Bewegtbildübertragung über ein lokales Netz.* Studienarbeit, Lehrstuhl für Praktische Informatik IV, Universität Mannheim, 1990.

[Zim90]   R. Zimmermann. *Übertragungsprotokolle für die Bewegtbildübertragung über ein lokales Netz.* Studienarbeit, Lehrstuhl für Praktische Informatik IV, Universität Mannheim, 1990.
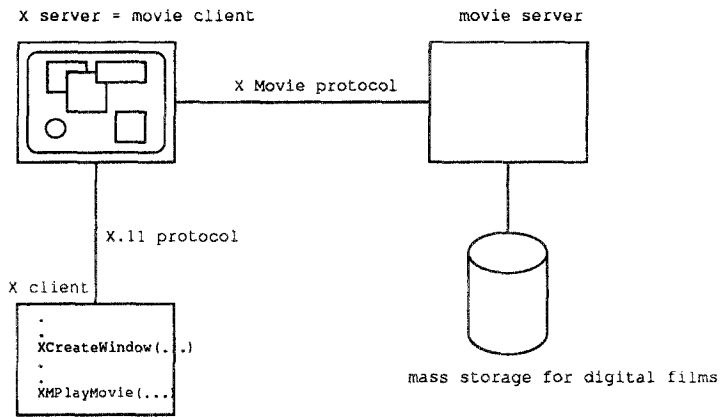
338

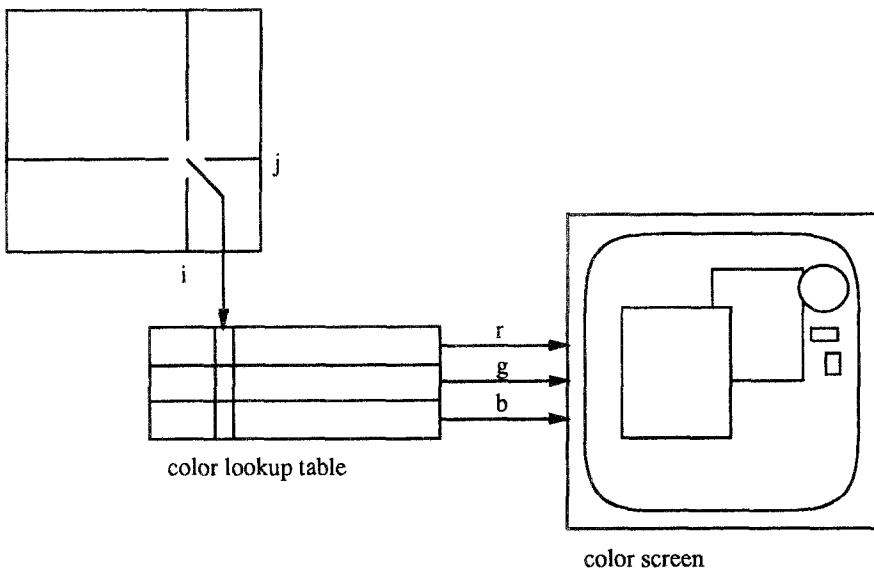Figure 1: Architecture of the X-MOVIE system
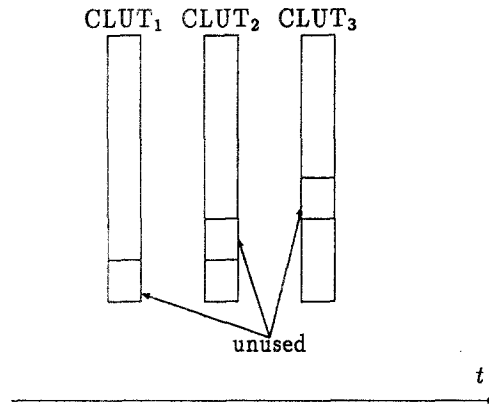

Figure 2: color lookup table technology

CLUT$_1$ CLUT$_2$ CLUT$_3$

unused

t

Figure 3: Dynamic adaptation of the CLUT

telnet, ftp, mail, rlogin, X-Windows, ...	nfs, X-MOVIE
TCP	UDP
IP	
Ethernet	Tokenring	⋯

Table 1: Protocol stack in the Internet