UNIVERSITY OF ALBERTA

Scalability in Media-on-demand Systems

BY

Kannan Thiruvengadam ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

DEPARTMENT OF COMPUTING SCIENCE

Edmonton, Alberta
Spring 1996

ISBN 0-612-10759-0

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Kannan Thiruvengadam

TITLE OF THESIS: Scalability in Media-on-demand Systems

DEGREE: Master of Science

YEAR THIS DEGREE GRANTED: 1996

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) . . . .
Kannan Thiruvengadam
3A-9010, 112 Street
Edmonton, Alberta
T6G 2C5

Date: 22 . Dec . '95 .

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Scalability in Media-on-demand Systems** submitted by Kannan Thiruvengadam in partial fulfillment of the requirements for the degree of Master of Science.

. . . .
Dr. Pawel Gburzynski (Supervisor)

. . . .
Dr. Wayne Grover (External)

. . . .
Dr. Janelle Harms (Examiner)

Dr. T. A. Marsland (Chair)

Date: Dec 21/95

...it is a safe prediction that in the next 50 years schools and universities will change more and more drastically than they have since they assumed their present form more than 300 years ago when they organized themselves around the printed book..

- Peter Drucker

Abstract

Scalability is a desirable quality of any system. Scalability of a media-on-demand system makes it last longer and reach more users, by virtue of its ability to cater to different levels of resources that different users have access to.

This work is an exploration of the ways to incorporate scalability in a system that delivers dynamic documents on demand to individual users over a wide area network. Specifically, we address three types of scalability.

- *Rate scalability* (the ability to control the rate of video according to the requirements of the client), is treated at depth; we suggest three ways of constructing video gateways, two of them compatible with the existing standards.

- *Component scalability* is the ability to deliver only those components of the document required by the user; this has two parts : providing optional components and providing alternative components.

  A component- and rate-scalable system can therefore be effective in providing an N-ISDN (Narrow-band Integrated Services Digital Network) user with text, images, audio and low quality video, a B-ISDN (Broadband Integrated Services Digital Network) user with all the above with the difference of high quality video, and a modem user with only text and images - all from the same composite document.

- *Content scalability* is the ability to deliver the nature and amount of information that the user requires. This quality is of interest to any information provider

who wishes to serve users with different interests. We show how object oriented information development, done using hypermedia, can make the content scalable.

In order to illustrate the above points, we have created a prototype of an open learning environment, called CyberSchool, by integrating a continuous media (audio and video) server with the World Wide Web.

Acknowledgements

My heartfelt gratitude is due to Dr. Pawel Gburzynski and Dr. Wlodek Dobosiewicz for their support and guidance during the course of this work. I thank the Faculty of Graduate Studies and Research for awarding me the the A.J. Gordin Kaplan Graduate Student Award and for nominating me for the Canadian Advanced Technology Association Scholarship. I thank Dr. Pawel Gburzynski for the travel support for presentation in LETT '95, Victoria. I am grateful to the support staff of the Department of Computing Science for various timely helps. I thank my friends for being what they are and for knowing when not to ask *How's your thesis coming?*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scalability of a system is its ability to cater to a variety of client requirements. For this reason, scalable systems reach farther and last longer than non-scalable systems. We explore the issue of scalability in ***media-on-demand*** systems. Consider a setup where ***dynamic documents*** (documents with static media like text and images, and dynamic media like audio and video) are delivered ***on demand*** (without involving local storage or latency) to customers through a communication network. In order to be specific, we assume the setup is meant to be used as a learning support environment for *open learning* (Open learning is a student-centered mode of learning).

Need for scalability

- *Difference in the levels of technology to which users have access*: B-ISDN (Broadband Integrated Services Digital Network) will become a reality sooner or later. But not every home will have B-ISDN. A number of users will have N-ISDN. Note that there are different levels of access even within N-ISDN (Narrow-band Integrated Services Digital Network); eg. BRI (Basic Rate Interface) and PRI (Primary Rate Interface) ([ISDN 95]). A larger number of users will, for a long time, be connected through POTS (plain old telephone systems) using a 14.4 Kbps or 28.8 Kbps modem. People in organizations (institutions and companies) may have access to a level different from all of the above, say a 10 Mbps Ethernet. Even though the levels of technology to which people have access continue to shift upwards, the difference in levels will always be there. Therefore, the heterogeneity of the network, on which potential clients for a media on demand system are situated, should be one of the primary concerns of an information provider.

- *Difference in learning goals*: With the advent of the online-information era, the possibilities of an open learning environment are on the rise. Open learning ([ITOLE 91]) is a student-centered mode of learning; i.e., the student decides the immediate, short-term and long-term goals of learning. Although this was originally conceived for use by the working community for professional development, there is no reason why such an environment should not be used to

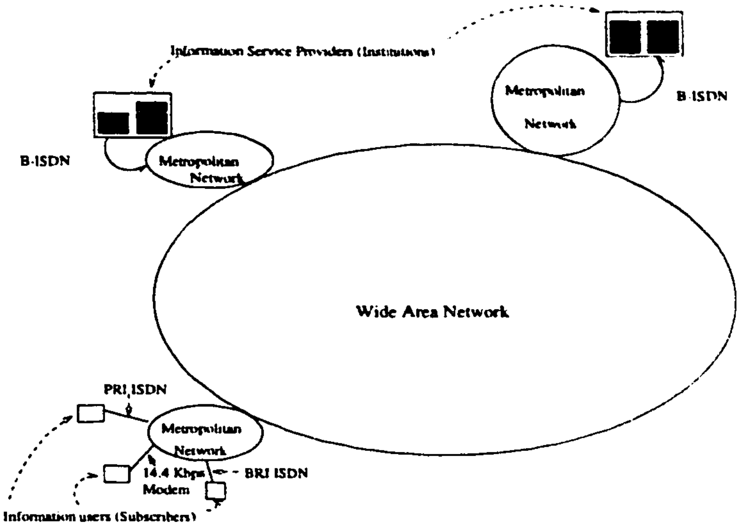


Figure 1: The heterogeneity of connections

support regular learning as well. The problem this situation gives rise to, however, is heterogeneity again, in terms of the learning goals of the users. The goal may be specific or general, need-based or curiosity-based, theoretical or practical. The nature and amount of information sought during a particular study session, may vary from user to user and from session to session. Hence, in order to make the learning environment successful, it is necessary that the information provider (educational information provider to be more precise - - what is called *institution* today) ensure support for individualization of content.

For the former need, the solution is two-fold: *component scalability*, where the user gets only those components of the documents that he desires (and his technology permits), and *rate scalability* where the user can specify the bit-rate at which the most bandwidth-intensive media, viz. video has to be delivered. As for the latter need, we suggest an object-oriented information organization scheme based on hypermedia (*Hypermedia* is means for online reading of documents containing text and images portions of which when selected by the user lead to other such documents. Formal definitions are available in Chapter 2).

Chapter 2 is devoted to content scalability. We explain why content has to be scalable and how it can be made so. The basic idea is to organize information in an object-oriented way. We also show how helpful hypermedia is as a tool in doing this. For purposes of illustration we use a model of hypermedia that captures the essential structural features special to hypermedia. Besides making the content scalable. the use of object oriented principles also improves the time and storage efficiency of integrated open hypermedia documents, (which is what future distributed information systems will be, including educational information systems).

Chapter 3 and 4 address component scalability and rate scalability respectively. Component scalability expands into two levels of choices to the user - the choice of media (from among text, images, audio and video) and, within the context of text and audio, the choice of language (or more generally, the availability of alternate components). Rate scalability refers to the user controlling the rate at which video is delivered, irrespective of the rate inherent in the recording. This paves the way for producing good quality orchestrated multimedia documents for best performance under high-tech playback conditions, and for being able to deliver it. when required. in low-tech conditions as well.

Chapter 3 brings out the difficulties in providing the user with the ability to choose from the various components of a document. Towards creating a flexible multimedia document delivery system, we explain the need to separate content from presentation, the emerging philosophy as demonstrated by the document structuring standards SGML and HyTime ([HYP 94]), and as illustrated by the markup language used on the world wide web, namely HTML.

In Chapter 4, various methods to construct video gateways (means to control bit-rate of video on the fly) are suggested. Empirical results are shown to support the theoretical comparisons of these methods. The support available in codec standards like *H.261* and *MPEG*, for 'gatewaying' video is also outlined. Scenarios other than video on demand where gateways can be used, and their interaction with the rest of the network are also discussed.

Chapter 5 is the description of a prototype of *CyberSchool*, an information technology-based learning environment. in which the above ideas have been incorporated. The prototype features remote controllable on-demand delivery of multimedia hyperdocuments and support for asynchronous interaction. The prototype has been constructed by combining the *World Wide Web* to a continuous media (audio and video) server integrated to the web through its *Common Gateway Interface*.

Chapter 6 contains our conclusions, recommendations and suggested future work.

Our contribution in this work is the following :

- Identification of the need to provide content scalability as a solution to heterogeneity of user requirements in terms of the nature and amount of information, and identification of ways and means to satisfy that need
- Identification of problems that arise while providing component scalability
- An in-depth exploration of the ways to scale the rate of unicast delivery of stored video over heterogeneous connections (video gateways)
- Design and development of a prototype of an educational application wherein the above ideas of scalability (with the exception of component scalability) have been incorporated

Support for individualization, which is in fact the ultimate form of scalability, calls for more than what we have addressed above. It involves taking into consideration the temporal, geographical, physiological, and even mental resources available to the user, besides the technological ones. Multimedia can be seen as the availability of means to reach out to a student through alternative senses, in case of impairedness. Making information accessible on demand, reduces most of the temporal problems, as well as mental issues concerning one's capacity to retain information, and speed in grasping concepts (This is because the student can study at his own pace, and refresh his memory whenever required). Geographical limitations are solved trivially by the availability of a network connecting the student with the information provider.

Chapter 2

Content Scalability

2.1 Introduction

In the context of information-delivery systems, intended for an application like home school, we define the problem of content- scalability as follows : *Users differ in the nature of the information they require, and in the amount of detail they require in any topic. The variety of requirements is so wide that it is practically impossible to prepare material for every user separately.*

To solve the problem of content-scalability, we seek a method that is economical in terms of development-cost and, at the same time, capable of catering to the wide variety of user-requirements. In this chapter, we argue that object-oriented information organization qualifies well for such a method. We identify *Hypermedia* as the means to implement that method. Besides, we develop a formal model of hypermedia, which we believe proves the suitability of hypermedia for the task at hand. Through this formalization, we also arrive at two basic guidelines for effective development of hyperdocuments.

2.2 Object-oriented information organization

First, we observe the principal advantages of object-oriented programming :

- **Code-sharability**: A *class* defined by one programmer for his own purposes can be used by another programmer - to create *objects* of that *class*, or to define another *class*. This advantage derives from the context-free nature of the definition of the initial *class* (The *class* definition is not too tied up with the first application to be usable in a different application).

- **Multi-purpose readability**: One programmer does not have to understand the whole of the code written by another programmer, to be able to build on it. For instance, the knowledge of *what the member functions do* may be sufficient, i.e., *how they do it* may not be necessary. The second programmer, is thus not forced to study that part of the code what he does not want to study, whereas, a third programmer, who wants to understand more about the program, can always do so. In other words, one can traverse the tree of objects and classes to satisfy ones own requirements, exploring any branch and reaching down to any required depth. This ability to serve multiple purposes derives from the abstraction incorporated in the program.

If information can be organized in a similar way, then evidently, the above-mentioned advantages of object-oriented programming will carry over to document development, allowing sharing of available information and multi-purpose readability.

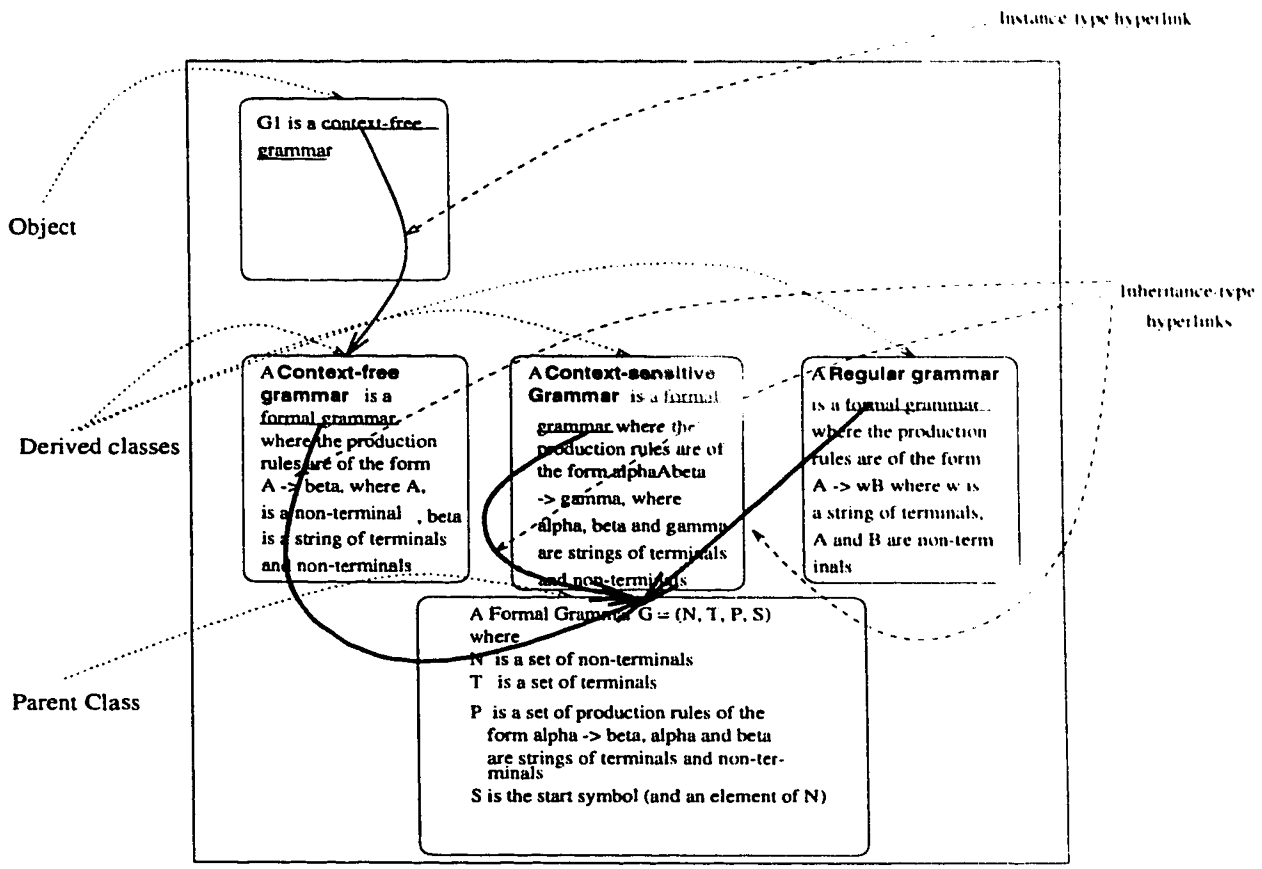
When it is difficult to structure information in a strictly object-oriented way, one can follow the general principles of abstraction and context-freedom.

2.3 Hypermedia

Hypermedia is basically a means to structure information in a non-linear fashion. The printed book is structured in a linear fashion - chapter after chapter, section after section, paragraph after paragraph, and sentence after sentence. In the hypermedia equivalent of a book, which we shall call a *hyperdocument*, information is contained in multiple plain documents; structure is imposed by the use of *hyperlinks*, each of which connects a part of a string or an image in one plain document, to another plain document.

For object-oriented organization of information, one has to

- break the entire information into classes, objects and member-functions
- use hyperlinks to connect the above pieces appropriately. *Our contention is that hypermedia can be used for this purpose.* When this is done, each hyperlink will fall into one of the following types :
  1. Encapsulation : the type of link between *what* is done and *how* it is done
  2. Instantiation : the type of link between an instance and a class (for example, *The Tale of Two Cities* and *novel*)


Instance type hyperlink
G1 is a context-free grammar
Object
Inheritance type hyperlinks
A Context-free grammar is a formal grammar where the production rules are of the form A -> beta, where A, is a non-terminal , beta is a string of terminals and non-terminals
A Context-sensitive Grammar is a formal grammar where the production rules are of the form alphaAbeta -> gamma, where alpha, beta and gamma are strings of terminals and non-terminals
A Regular grammar is a formal grammar where the production rules are of the form A -> wB where w is a string of terminals, A and B are non-terminals
Derived classes
A Formal Grammar G = (N, T, P, S) where
N is a set of non-terminals
T is a set of terminals
P is a set of production rules of the form alpha -> beta, alpha and beta are strings of terminals and non-terminals
S is the start symbol (and an element of N)
Parent Class

Object-orientation in Hyperdocument development

Figure 2: Object-orientation in Hyperdocument development.

3. Inheritance : the type of link between a parent-class and a child-class (for example, *book* and *novel*)

The direction of each type of link is as shown below :

- Encapsulation : Class to member
- Instantiation : Object to class
- Inheritance : Child-class to parent-class

Fig. 2 illustrates the proposed method of developing hyperdocuments.

If the above style is used in developing hyperdocuments, it becomes possible for users to read any node of the hyperdocument directly. Also, each user will be able to access more details as and when he needs them (At the physical level, this translates into an action like clicking on the term in question).

So far, the ways and means to organize information and to structure it so it is content-scalable, have been presented. Note that, in the process, we have proposed a style of using hypermedia. In the following section, we justify the style by presenting a formal model that captures the structural properties of hypermedia. The formalization is also meant to help hyperdocument developers gain some insight into the power and limitations of the structure of hypermedia so they can use hypermedia effectively. The idea here is to exploit the structural properties of hypermedia by working a match between structure and semantics. The formalization can also be thought of as the answer to the question 'Why should every hyperlink stand for one of the relations - inheritance, encapsulation and instance?'.

2.4 Hyperdocument development style

2.4.1 Formal Grammars and Languages

Definitions:

- *Formal Grammar*:

  A formal grammar G is denoted by (N, T, P, S), where N is a set of *non-terminals* (or *variables*), T is a set of *terminals*, P, *Productions*, and S, the *Start Symbol*. N and T should contain no intersection. S has to be one of the non-terminals.

  P consists of productions of the form $\alpha \rightsquigarrow \beta$, where α is a non-empty string of elements of $N \cup T$, and β is any string of elements of $N \cup T$.

  Eg.: Grammar $G_1 = (N_1, T_1, P_1, S_1)$, where
  $N_1 = \{S_1, B_1\}$,
  $T_1 = \{a, b\}$,

$P_1 = \{S_1 \rightsquigarrow aS_1a, B_1 \rightsquigarrow b\}$

- *Formal Language* :

  If a string γ can be obtained from S by the application of a set of productions, then γ is said to be a *sentential form* in the grammar G. If γ contains only terminals then it is referred to as *a sentence* in the language denoted by $L(G)$. $L(G)$ is nothing but the set of all such sentences.

  Eg.: Language $L(G_1) = a^n ba^n$, where $n > 0$.

- *Regular Grammar* : A *regular grammar* is a formal grammar where the production rules are all of the format $A \rightsquigarrow wB$ where A and B are non-terminals and w is a string of terminals.

  Eg.: Grammar $G_2 = (N_2, T_2, P_2, S_2)$, where
  $N_2 = \{S_2, A_2, B_2\}$,
  $T_2 = \{a, b, \epsilon\}$,
  $P_2 = \{S_2 \rightsquigarrow aA_2, A_2 \rightsquigarrow bB_2, B_2 \rightsquigarrow aB_2, B_2 \rightsquigarrow \epsilon\}$

- *Regular Language* : A *regular language* is the formal language associated with a regular grammar.

  Eg.: $L(G_2) = a^n ba^m$

- *Context-free Grammar*: A *context-free grammar* is a formal grammar where the production rules are all of the format $A \rightsquigarrow \alpha$ where A is a non-terminal and α is a string of terminals and non-terminals.

  Eg.: Grammar $G_1 = (N_1, T_1, P_1, S_1)$, where
  $N_1 = \{S_1, B_1\}$,
  $T_1 = \{a, b\}$,

$P_1 = \{S_1 \rightsquigarrow aS_1a, B_1 \rightsquigarrow b\}$

- *Context-free Language* : A *context-free language* is the formal language associated with a context-free grammar.

  Eg.: $L(G_1)$ is a context-free language.

- *Context-sensitive Grammar*: A *context-sensitive grammar* is a formal grammar where the production rules are all of the format $\alpha \rightsquigarrow \beta$ where α and β are strings of terminals and non-terminals, α not being a null string.

  Eg.: Grammar $G_3 = (N_3, T_3, P_3, S_3)$, where

  $N_3 = S_3, A_3, B_3,$

  $T_3 = s, a, b,$

  $P_3 = \{S_3 \rightsquigarrow sA_3B_3s, A_3 \rightsquigarrow a, B_3 \rightsquigarrow b, saB_3s \rightsquigarrow sabs, sA_3bs \rightsquigarrow sabs\}$

- *Context-sensitive Language*: A *context-sensitive language* is the formal language associated with a context-sensitive grammar.

  Eg.: $a^nb^nc^n$ is a context-sensitive language.

- *Context-free Graph-grammar* : A *context-free graph-grammar* is like a context-free grammar with one difference : the right hand side of each production is a graph.

  Eg.: $G_4 = (N_4, T_4, P_4, S)$, where

  $N_4 = S, A, B,$

  $T_4 = a, b,$

  P_4 is as shown in Fig. 3.

- *Graph* : A graph is a collection of *vertices* and *edges*. A *vertex* is a point and an *edge* is a line joining two vertices.

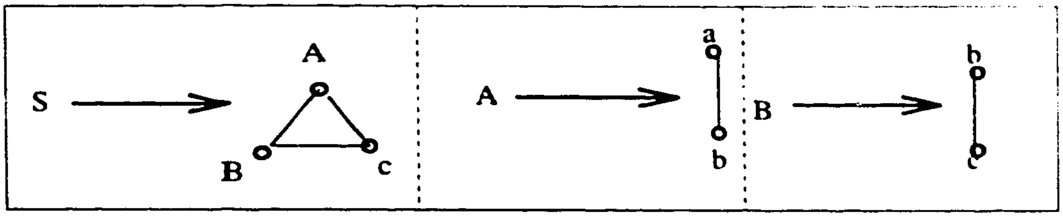


Figure 3: Production rules in a context-free graph grammar

- *Planar graph* : A planar graph is a graph that can be drawn on a two-dimensional surface with no edges crossing each other.

- *Context-free Planar-graph Grammar* : A *context-free planar-graph grammar* is a context-free graph grammar where all graphs involved are planar

  Eg. Grammar G_4 is a context-free graph-grammar.

2.4.2 Hypermedia

For examples pertaining to the following terms, see Fig. 4.

Definitions:

- *Hypertext* : Hypertext is text embedded within text. This *embedding* is realized by associating a subtext in one text document with another text document through the use of a hyperlink. In other words, a hyperlink maps a subtext from one text document to another document. Each subtext that is associated to a text document is an *anchor*.

- *Hypermedia* : Hypermedia is the combination of text and images, where both media (text and images) can contain hyperlinks.

- *Hyperdocument* : A hyperdocument is a composite document created using hypermedia. It is made of *nodes* which are plain documents, and *hyperlinks* between these nodes.

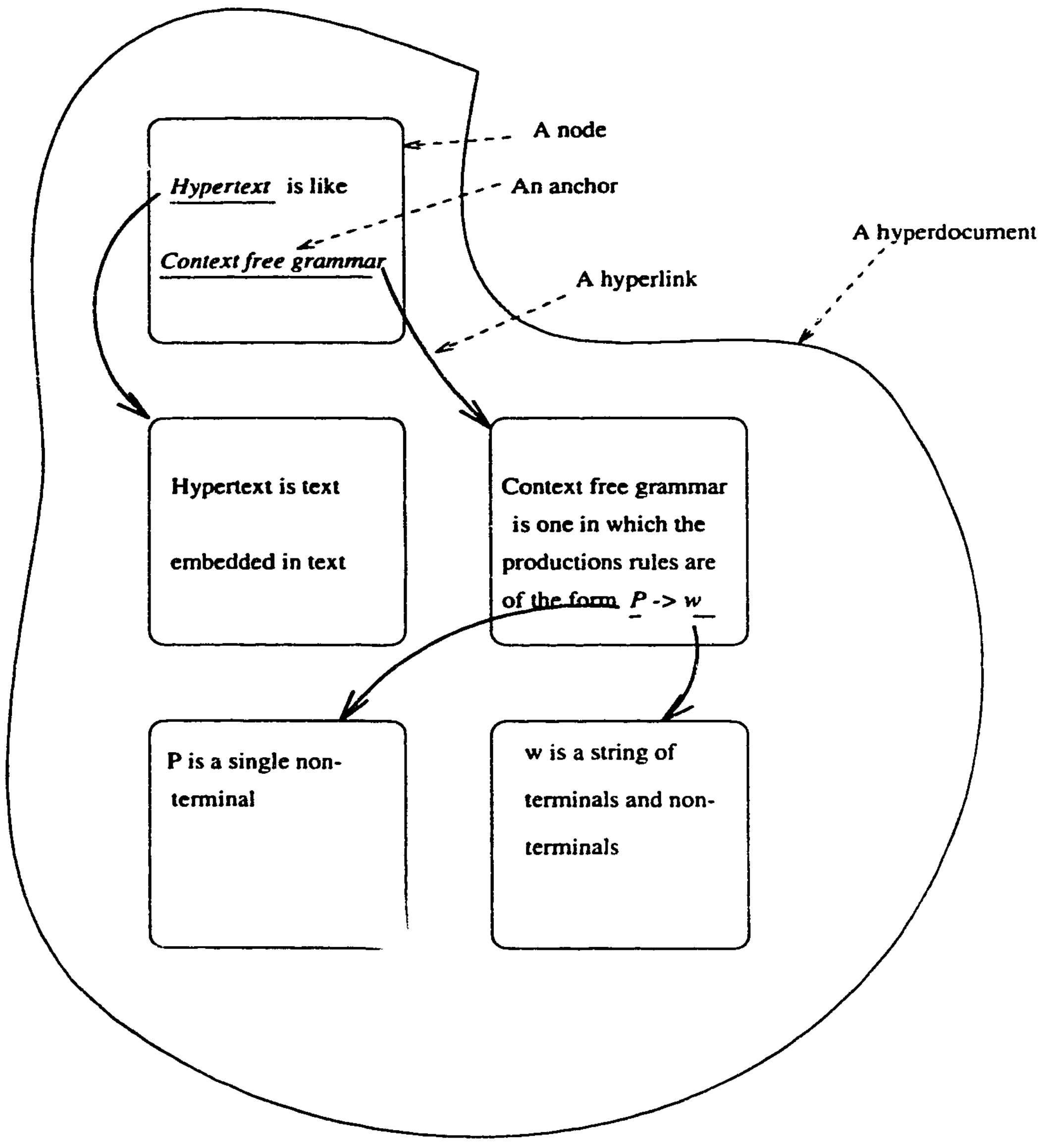


Figure 4: Components of a hyperdocument

- *Anchor* : An anchor in a hyperdocument is any subtext (or subimage) in one node of the hyperdocument the selection of which leads to another node of the hyperdocument.

- *URL* (Universal Resource Locator) : URL is the term used in the World-wide-web parlance to indicate the unique identifier of each resource (node of a hyperdocument, image etc.) made available on the web.

- *Hyperlink* : A hyperlink is a connection between a subtext (or subimage) in one node of a hyperdocument and another node.

- *Map* : A *map* is an image with several regions. A region may be clickable or non-clickable, i.e., a region can have a hyperlink associated with it.

- *Including node and Included node* : If node n_1 contains a hyperlink that leads to node n_2 then n_1 is an including node with respect to n_2 and n_2 is an included node with respect to n_1.

2.4.3 The structural relation between CFG and Hypermedia

Claim : Hypermedia and context-free grammars have the same structure.

Informal proof

- One can reach a particular node of the world-wide-web, which is a hyperdocument, in several ways (i.e. from several other nodes and by specifying the URL of the required node). This means. no matter where you invoke the URL of a particular node from, you always land in that node. In other words. the destination node is independent of the context from which its URL is invoked. This makes URLs analogous to non-terminals in a CFG, as each non-terminal produces the same string no matter what context it (the non-terminal) is present in.

- One can use any number of hyperlinks in the same node of a hyperdocument. This makes a node in a hyperdocument analogous to the right hand side of a CFG production-rule, where multiple non-terminals can be present. Considering hypertext alone, (i.e., without allowing the use of images), one can note that there is a particular order in which the anchors (clickable portions) appear. Correspondingly, there is an order in which the non-terminals appear in any CFG production rule. It is easy to extend this argument to images, thus showing the structural equivalence of hypermedia to context-free grammars.

Fig. 5 shows a hyperdocument and a structurally equivalent context-free grammar. There is a production rule in the grammar for every node in the hyperdocument.

Formal proof

Now, we transform hypertext into context-free string-grammars and vice versa, thereby establishing the structural equivalence of hypertext and context free grammars.

- *Claim : Every hyperdocument H has an equivalent context free grammar* $G = (N, T, P, S)$, where
  - S is the starting symbol
  - N is the set of non-terminals
  - T is the set of terminals
  - P is the set of production rules

  Proof:

  1. Create N by defining a non-terminal for every anchor in H.
  2. Define a new non-terminal in N for every node in H that is not referred to in any other node in H.
  3. Create T, using all lower case characters in the English alphabet. With each longest terminal string in H, associate an element of T.

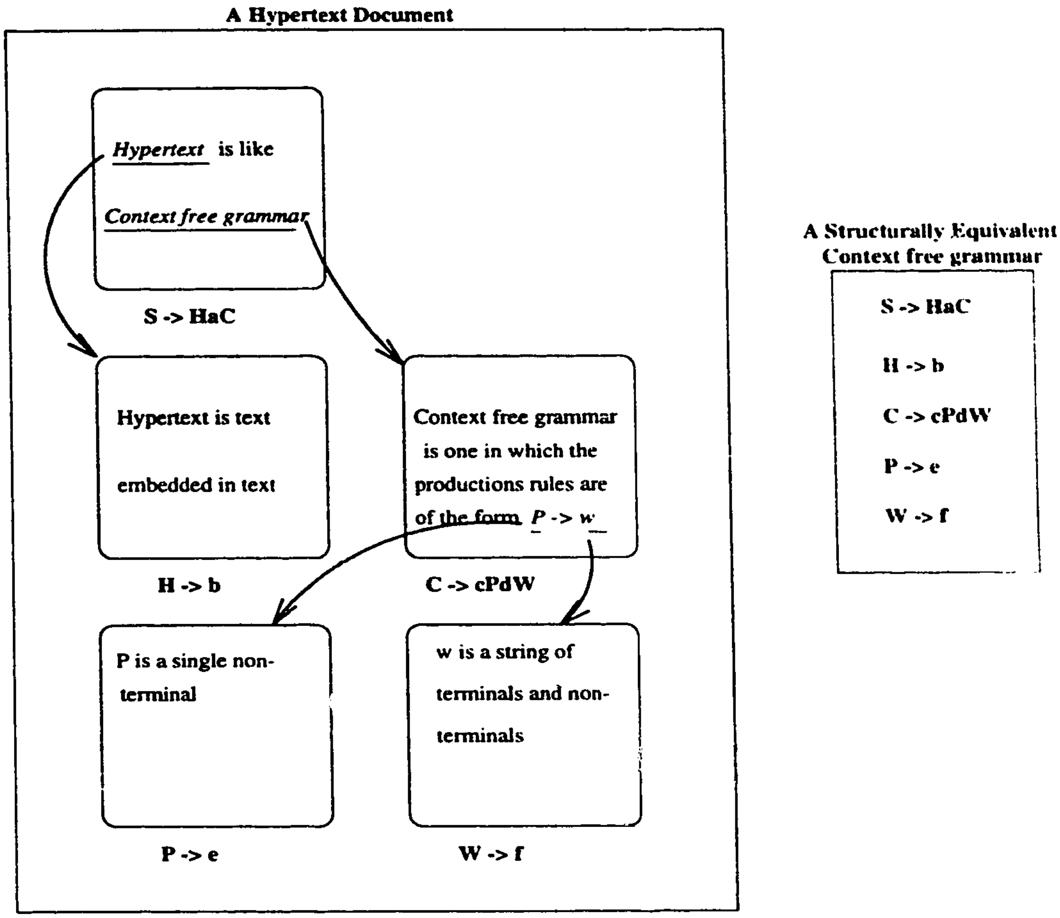


Figure 5: Context-free string-grammar and Hypertext

4. For every anchor in H, create A, by defining a production rule $A \leadsto w$. where A is the non-terminal that stands for the anchor, and w is the string of terminals and non-terminals corresponding to the node the anchor leads into.

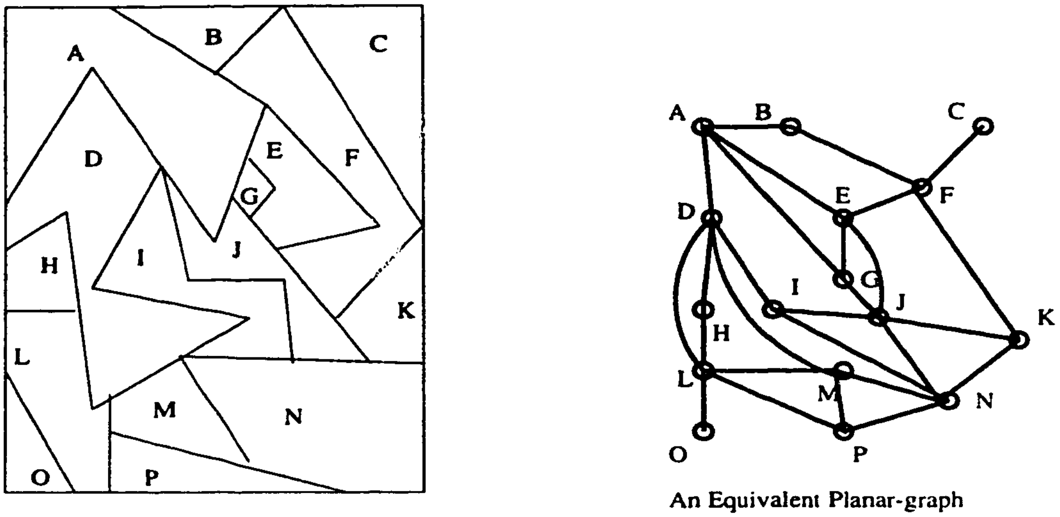
5. Define a new rule $S \leadsto w$ in H, where w is a string of those non-terminals that stand for the nodes in H that are not referenced elsewhere.

Now G is structurally equivalent to H.

Figure 5 illustrates the transformation.

- *Claim:* Every CFG G has a corresponding hyperdocument H.

  Proof:



An Equivalent Planar-graph

A Map

Figure 6: A map and its corresponding planar-graph

1. For every production $A \rightsquigarrow w$ in G, define a node in H containing w.
2. For every production $A \rightsquigarrow w$ in G, convert every occurrence of A in H into an anchor to the node containing w.

The above proof holds only for the structural equivalence of context-free string-grammars and hypertext. It is now required to show that the equivalence holds good even if the nodes of the hyperdocument are maps and not text. Note that this can be seen by considering w (mentioned in step 4 of Proof 1 and in both steps of Proof 2) a graph, instead of a string. The graph is to be derived from the corresponding map in the following way : create a vertex for each region in the map, and create an edge connecting two vertices if and only if the two corresponding regions are neighbors. Such a graph will be graph by graph-theoretic results (For every map, there is an equivalent planar-graph: See Fig. 6 for an instance of a map and its planar-graph equivalent).

2.4.4 Usefulness of the formalism

By showing that context-free grammars and hypermedia are structurally equivalent, we have in fact arrived at a method to evaluate the power of hypermedia as a tool. The power of formal grammars are well established with respect to one another. Regular grammars are less powerful than context-free grammars, in that the latter can do more than what the former can. In the same sense, context-free grammars are less powerful than context-sensitive grammars. *Hyperdocuments are to printed books, what context-free grammars are to regular grammars.* Thus hypermedia is a superior tool for information representation and presentation in comparison to printing.

In using the power of hypermedia

The fact that multiple hyperlinks can be used in each node allows a user to develop each node with one and only one central idea, leaving the next level of details to other nodes, by just including hyperlinks to those nodes from the main node. This ability to hide information is the power of hypermedia. It helps the developer to concentrate on the relation between the sub-concepts, in presenting a concept, rather than delving into the sub-concepts themselves. In other words, it is *underuse* of the power of hypermedia to have a lengthy plain document for one node of a hyperdocument. Simply put, the power of hypermedia is its support for *abstraction* (i.e. hiding information). To a reader this means depth-on-demand. One can prune the tree of information one is traversing, to meet one's immediate requirements. In other words, details are not thrust on the reader. Instead the reader asks for them and gets them, as and when the need is felt. Some may go deeper into certain branches while others may choose not to. This is the solution we propose to one part of the problem of content-scalability, namely variety of user-requirements in terms of the amount of information required.

In knowing the limitations of hypermedia

The fact that each URL decides the destination node by itself disallows the possibility of making the information in a node sensitive to the context in which its URL is used

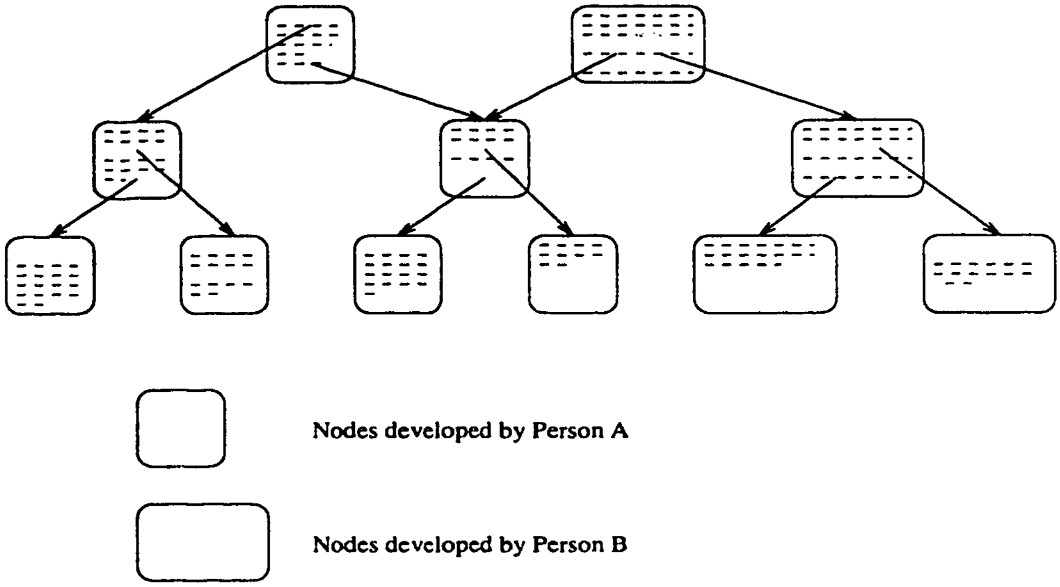


Figure 7: Document-sharing

in some other node. This limitation can be turned to the developer's advantage if the semantics of an *included* node is made independent of context in which it occurs in an *including* node. To promote such a style, a bottom-up development is preferable to a top-down one (That way there is no chance for the developer to make an included node sensitive to context. If this style is followed, it becomes possible for the same node to be included in many nodes, thereby leading to **document-sharability** (See Fig. 7). Due to the same context-free nature of the nodes, any node can be a starting point for reading. In other words, reading a particular node will not in any way require the reader to have read an including node. This is how we propose to tackle the other part of the problem of content-scalability, namely, the variety of user-requirements in terms of the nature of the information required.

Chapter 3

Component Scalability

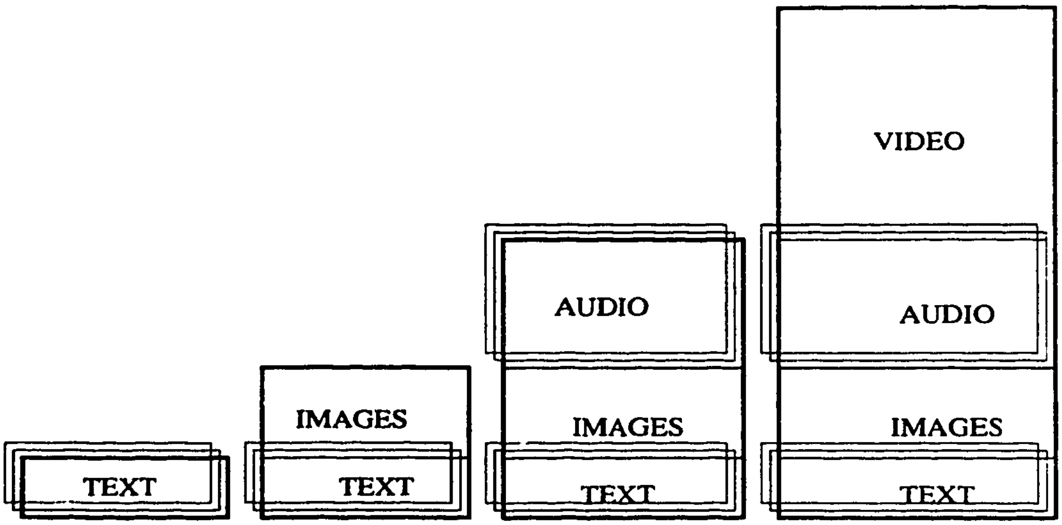


Figure 8: Typical choices of document components

3.1 Introduction

Component scalability is the ability to provide the user with any combination of the components that constitute the document. This includes both choice of media (from text, images, audio and video) and choice of language in the case of text and audio (Fig. 8). From a more generalized point of view, component scalability is about the user having *optional* and *alternative* components in the documents. The flexibility of the document structure is of primary concern here, in order to reach many levels of technology, and in order to make modifications and extensions easy for the author(s) of the document.

3.2 Implications

The main implication of the requirement of component scalability is that, since audio may have to be delivered and played without video, strong intra-medium synchronization is necessary (so every media-component is deliverable independently). The other implication is that the need for encoding inter-media synchronization information in the media themselves, and in some cases even the ease of doing this, are eliminated.

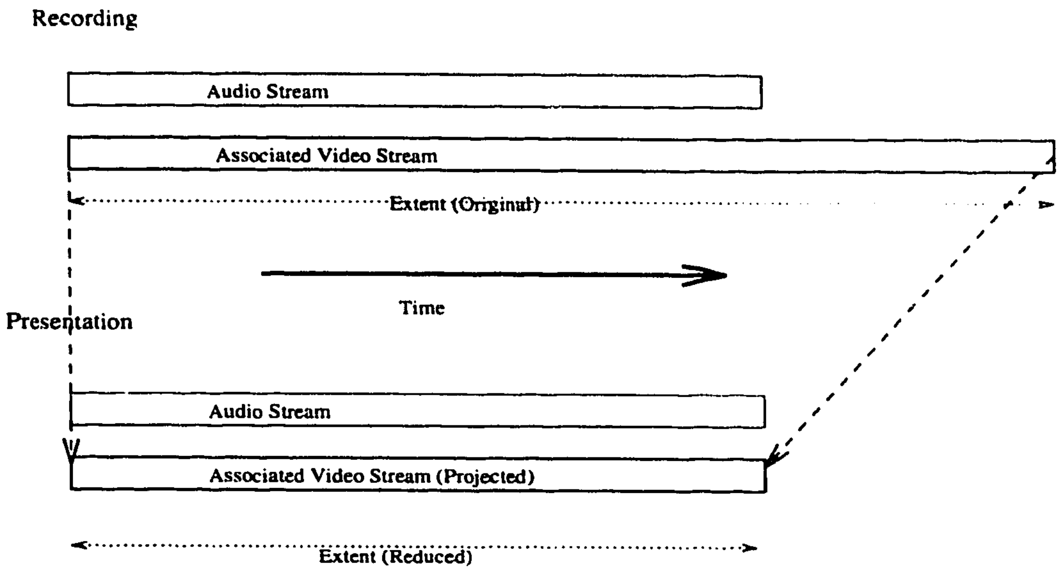


Figure 9: Event Projection

The following are the reasons for the second implication:

- **Replacement/Modification of a component:** The author may want to modify the audio encoding alone.

- **Addition of an alternate component:** The author may want to add a new audio encoding (as an alternative to the existing one - perhaps in another language).

- **Transfer time modification of a component:** By the time it is available at the receiver for playback, the video could have undergone some transcoding somewhere in between the server and the receiver, to suit the technology available for playback, as a result of which the inter-media synchronization information encoded in it becomes useless.

- **Event Projection:** The author may want to playback the video alone at a different frame rate than the one in which it was encoded. Such situations are possible (Fig. 9); a demonstration of a laboratory experiment for instance, may involve a person walking around the laboratory doing things, but doesn't have to show him walking from one place to another in real time; it might still be

> necessary to show him walking, so the student knows what sequence the various steps involved are performed in. Also, the audio corresponding to this video sequence, may be shorter than the real time of the video sequence, for which reason the author might want to run the video faster than the recording rate, fast enough to make it fit into the same duration as the audio. Note that this *time-scaling* is being done only to the video, while the audio is left as it is.

Therefore the only point in time where it is possible to find out which frame of video to associate with which sample of audio is during the playback itself. If the document specifies two components as being parallel (let us assume *parallel* here means that they have to be started at the same time, and synchronized throughout), the playback routine can make sure that is done, by matching the timing information available in the encodings (i.e. intra-media synchronization information) with its own concept of real time, and with the knowledge of time-scaling if any (faster or slower display of frames than their generation during recording).

The idea of separating the content from the presentation has become attractive of late. It enables the author of the multimedia document to make the presentation the way he wants, and not be constrained by the encoding of the components (objects that constitute the document). A version of this idea as applied to static media can be seen in HTML 2.0.

```
<IMG SRC=''pict.JPG'' WIDTH=200 HEIGHT=500 SRC=''image.jpg''>
```

No matter what the resolution of the actual image is, it will be displayed as a 200 X 500 pixel image, due to the above specification. This idea is also supported in the HyTime (Hypermedia Time-based media) document structure standard ([HYP 94]). HyTime consists of a rendition module that processes objects before they are rendered. This processing is done in an object specific way - for instance, the colors in an image can be converted to take advantage of (or to confine to) the colors available in the user's display unit. Extending this idea to dynamic media like video involves network issues: the rendition module (together with the scheduling module of HyTime which maps a source FCS [Finite Coordinate System] to a destination FCS) should communicate with the server to arrange for a reduced (or increased) frame-rate, to put the required time-scaling into effect (HyTime allows the application to do *event projection*; When the projection is along time-axis, it becomes time-scaling). This is besides what is initially agreed upon by the user as an acceptable frame rate to suit his display technology. Hence, in effect, the playback program will have to put both into play - the rendition and presentation specifications (*Rendition* is conversion of an object at playback time to suit the playback technology. *Presentation* specifications are those made by the author during the creation of the document regarding the temporal and spatial relations between components of the document, and event projection for each component).

From the above, it follows that inter-media synchronization information will not be ready until the media to be synchronized are actually available at the receiver. We conclude from this that the playback routine (which will be a HyTime Engine, if the documents delivered follow the HyTime standard) has to be able to derive inter-media synchronization information from the following :

- intra-media synchronization information available in the media
- presentation and rendition related specifications.

In the case of video, rendition also includes network level operations - the playback routine has to talk to the video gateway (through which video is routed) to make sure the characteristics of the video (frame rate for instance) conform to the playback constraints specified by the user.

Chapter 4

Rate Scalability

4.1 Introduction

In the context of Media-on-demand (MOD), we define the problem of rate-scalability as follows : *In any given video-recording the bandwidth-requirements for playback via network are inherent. Any client who can not establish a connection satisfying these requirements will, therefore, not be able to access the recording on demand.*

Our objective is to remove this limitation and make video available to even those with a lower level of resources (in terms of network bandwidth). There is no previous work addressing this very issue. Hence we investigate the problem from scratch, and develop methods to tackle it. However, a similar problem that arises while multicasting video on a network with heterogeneous branches, due to the different congestion states of the different branches, has been pointed out in [MULT 94]. Therefore, the solution we develop here is usable in both contexts. Our focus, nevertheless, is on Media-on-demand.

4.2 Solutions to rate-scalability

4.2.1 Layered coding

One way to tackle the problem is to use a layered-codec in encoding the video before recording. A layered coding typically consists of a **base layer**, which by itself can be used to produce moderate quality video, and one or more **enhancement layers** which can be used along with the base layer to enhance the quality of the video. Depending on thenetwork-bandwidth available, a user may select the base layer and an adequate number of enhancement layers (note that this number can be zero). (See Fig. 10) Note that layered-coding offers the user choice in the bit-rate requirements to be able t playback the video. This choice, however is limited. Playback at bit-rates other than the ones determined by the layers are not possible. Besides, during the playback, if the user wants to dynamically reduce or increase the bit-rate, it is not possible.

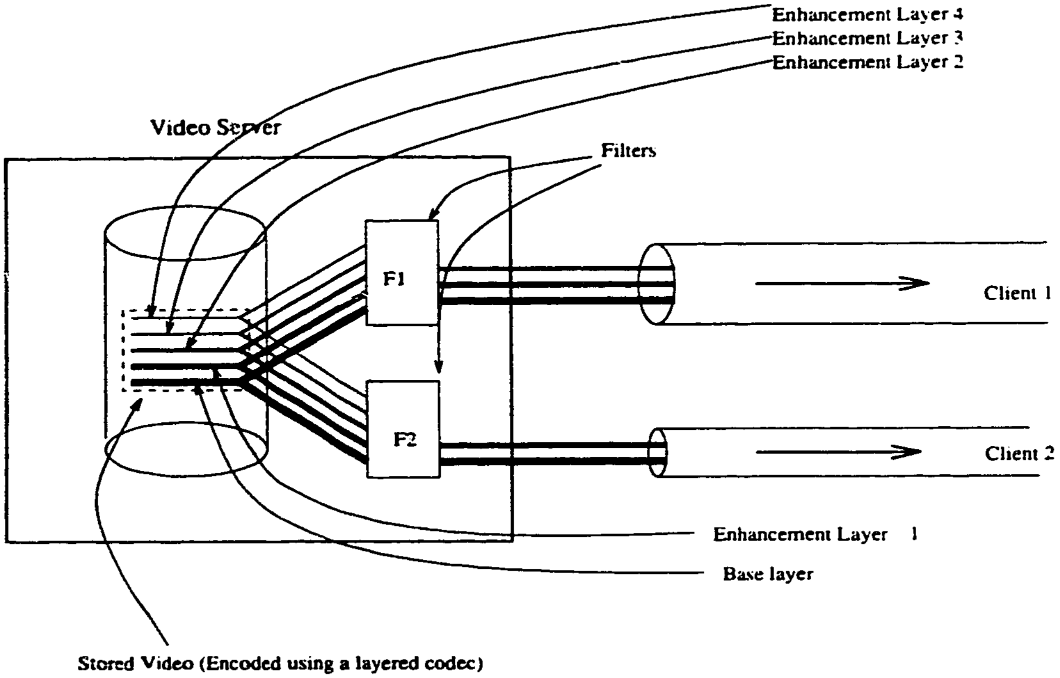


Figure 10: Layered-coding for rate-scalability in MOD

4.2.2 Video Gateways

We define a *Video Gateway* as a transport level network element that takes in video of Encoding X and outputs video of Encoding Y. A *rate-control video gateway* is one where the output rate is different from the input rate. A *code-control video gateway* is one where the output encoding is different from the input encoding. Such gateways are also called *transcoders.*

For the purposes of the problem at hand, we are concerned only with *rate-control video gateway.* More specifically, we are concerned with *ratc-reduction gateways.* The idea is to record video at the highest bit-rate at which any client might require it. and then to play the same video for any lower level through a rate-reduction gateway.

4.2.3 Comparison : Layered-codec and video gateway

Table 4.1 shows the difference between layered-codecs and video gateways as methods to achieve rate-scalability. We substantiate the claims made in the table through the

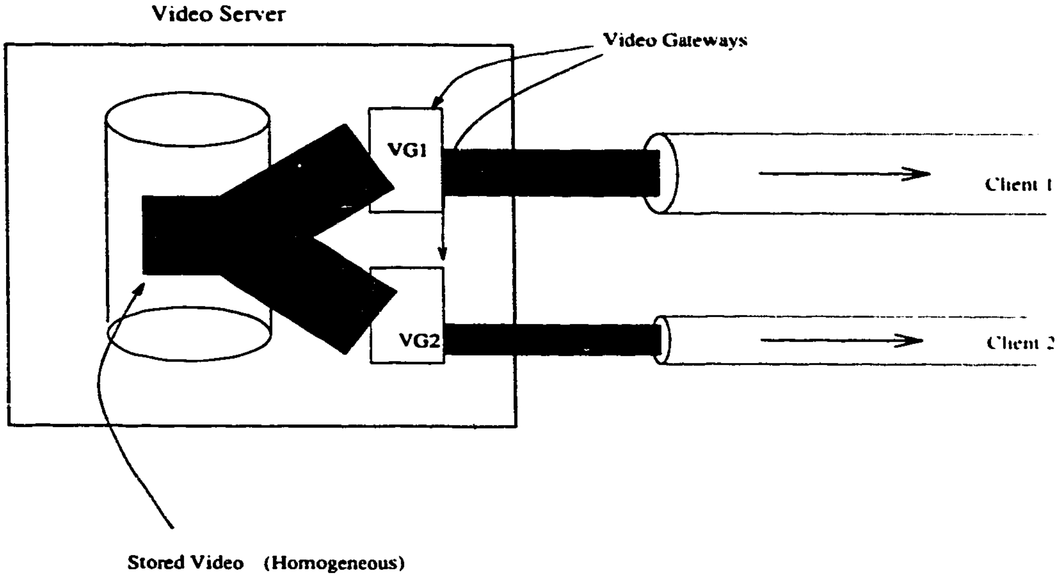


Figure 11: Video gateway for rate-scalability in MOD

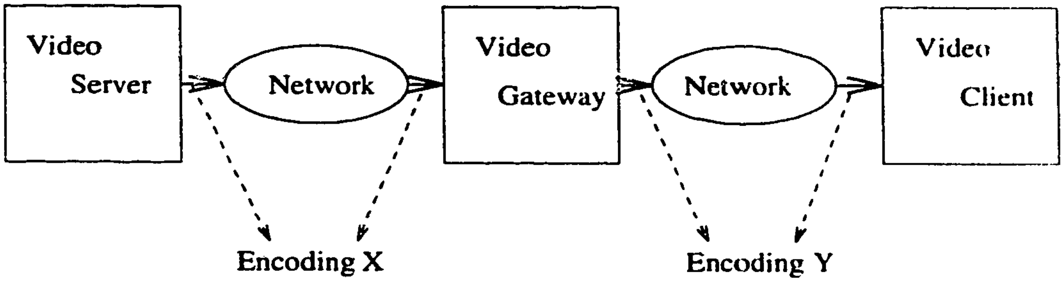


Figure 12: A General Video Gateway

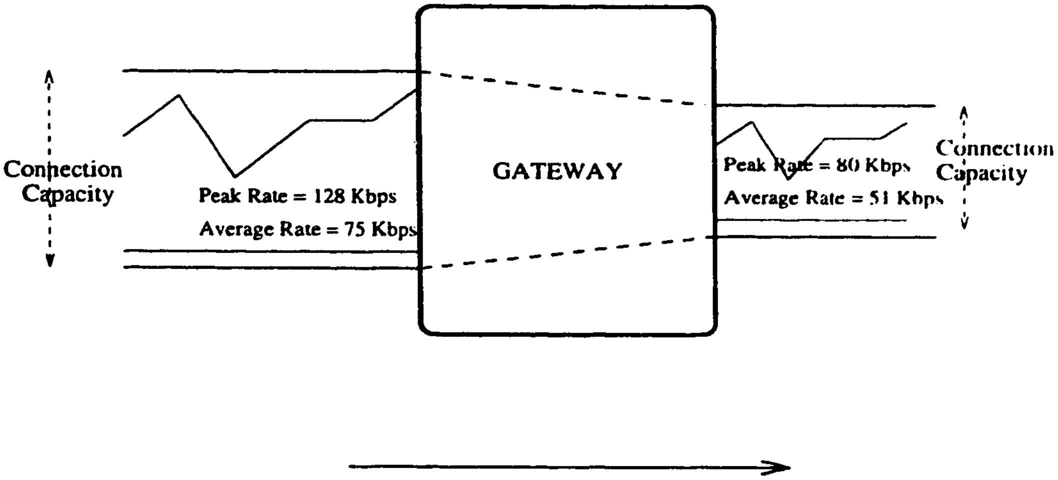


Figure 13: A Rate Control Gateway

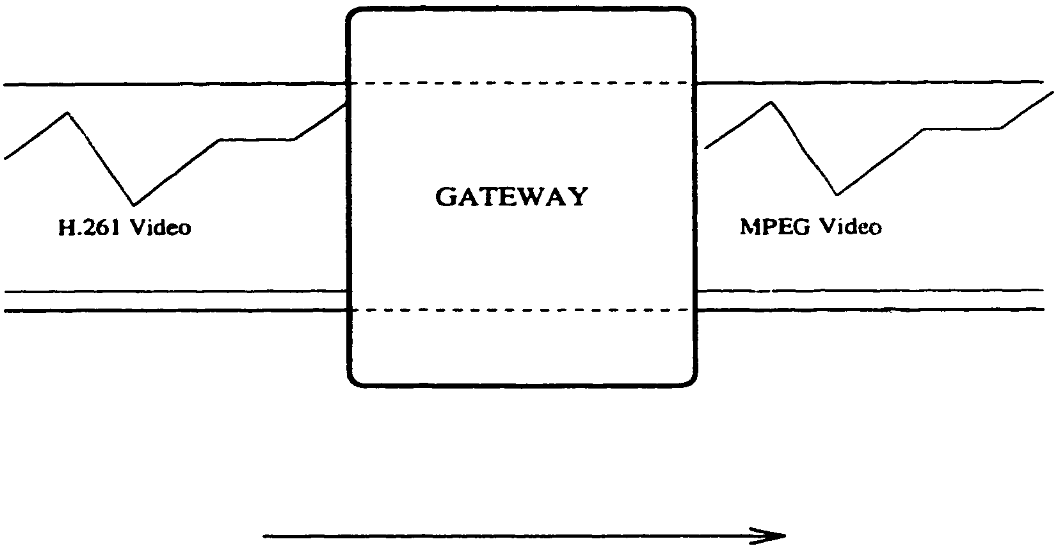


Figure 14: A Code Control Gateway (Transcoder)

following :

- Using a layered-codec to tackle difference in bit-rate requirements of various clients is a preventive mechanism in the sense, the task of separating video into layers is done during recording. Video gateways, on the other hand, are employed during playback to take the high bit-rate recorded video and convert it output the same but at the bit-rate required required by the client.

- The combined bandwidth requirement of the base and the enhancement layers is larger than the bandwidth required for non-layered video carrying the same information ([Sh 95]). This is an important drawback of this method as the whole idea behind rate-scalability is to reduce bit-rate requirements. As for video gateways, this drawback is absent, as the output video is non-layered.

- Since all possible playback-time bit-rates get fixed at the time of recording, there is no way to control the bit-rate during the playback, when layered-codecs are used. As video gateways function during playback, dynamic control is possible. (See Table 4.2)

Layered Coding	*Video Gateway*
A preventive mechanism	A curative mechanism
Bandwidth requirement more	Bandwidth requirement less
No support for dynamic control	Support for dynamic control
Narrow choice of bit-rates	Wide choice of bit-rates
No playback-time computation	Playback-time computation reqd.

Table 4.1: Layered-codec and Video gateway : Solutions for rate-scalability

- The number of enhancement-layers decided the number of options users have in terms of bit-rate required for playback. Such a limitation is absent in the case of video gateways as they can cater practically to any bit-rate require by the user by dynamically controlling the parameters used for re-encoding the input video. Table 4.2 compares the output bit-rate options made available by layered-codec and video gateways, as per the implementation of the former done in [Sh 95] and the implementation of the latter in our work (In [Sh 95], layering is done in two aspects - spatial and frequency. The input specifications tell the codec the location of the splitpoint in both aspects). Layered-coding was developed in [Sh 95] as a technique to exploit the support for prioritized traffic available in ATM networks.

- *Filters* (See Fig. 12) are used in the case of layered video to allow or disallow certain enhancement layers to pass through them, based on the bandwidth-requirements of the output channel. The job of these filters is simple as all they have to do is to identify the layer to which each packet belongs. This means minimal increase in the load on the machine on which filters are running. Video gateways. on the other hand. do a job that is as complicated as the codec itself (although certain types of gateways can be much simpler. as it will be shown later in this chapter), and therefore cause a huge increase in the load on whichever machine they are running.

Method used	Output bit-rates available
Homogeneous video	1) 6 Kbps
Layered : Spec. 1	1) 7.5 Kbps 2) 9 Kbps
Layered : Spec. 2	1) 7 Kbps 2) 9 Kbps
Layered : Spec. 3	1) 7 Kbps 2) 9 Kbps
Layered : Spec. 4	1) 7 Kbps 2) 9.5 Kbps
Video Gateway	0-6 Kbps (any rate)

Table 4.2: Bit-rate options available to user

4.3 Video Gateways : Design and Implementation

4.3.1 Rate reduction techniques

Rate control essentially includes decoding the incoming video and then reencoding it, with different encoding parameters so higher compression can be achieved. The following is an exhaustive list of operations that can be performed on a stream of video, in order to cutting down the bit-rate.

- *Spatial subsampling :* Spatial subsampling stands for selecting portions of the frame to transmit. This can be done either by cutting out a part of the frame and transmitting the cut-out, or by scaling down the dimensions of the frame. The latter, again, can be implemented depending on the application concerned. either by selecting more pixels in the regions of interest and less from the rest of the frame (a technique called *Variable Resolution*) or by selecting pixels at constant intervals along the length and breadth of the frame. (Fig. 15)

- *Temporal Subsampling :* Temporal subsampling means selecting a subset of the incoming frames to transmit by increasing the temporal sampling interval. Its effect, therefore, is the reduction of the frame rate. This omission of frames may require that each selected frame be encoded with respect to another frame that is also selected. Frames encoded this way, i.e. *with respect to* another frame are called *INTER* encoded frames. No such special care has to

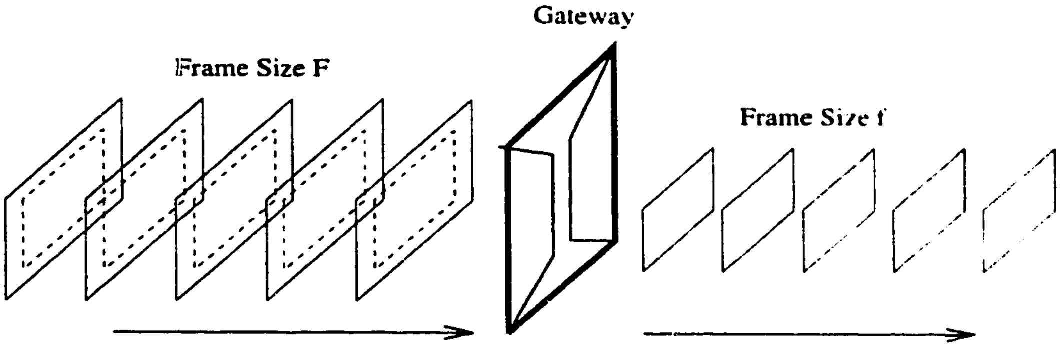


Figure 15: Spatial Subsampling

be taken, when a frame selected for transmission is an INTRA encoded frame (self-contained/complete/independent frame).

- *Amplitudinal Selection :*

  Amplitudinal selection in the pel domain stands for reducing the number of bits used to represent each pixel. Converting the image from color to grey scale, and, retaining its colored nature while reducing the number of bits to carry the color component, fall under this category. In the transform domain, amplitudinal selection stands for quantization and selection of a subset of transform coefficients from each block based on frequency, energy content, or resultant distortion, to be transmitted ([PRI 92]). This selection is referred to as *data partition*, and the following are the three ways to partition data.

  - **Frequency Truncation:**

    Selecting the first n transform coefficients

  - **Energy Thresholding:**

    Selecting all coefficients with value above a threshold n

  - **Minimum Distortion:**

    Selecting the top n coefficients based on their value

Fig. 18 shows all three of the above techniques applied to a vector of transform coefficients, with threshold= 4. The first 4 coefficients are selected in the case

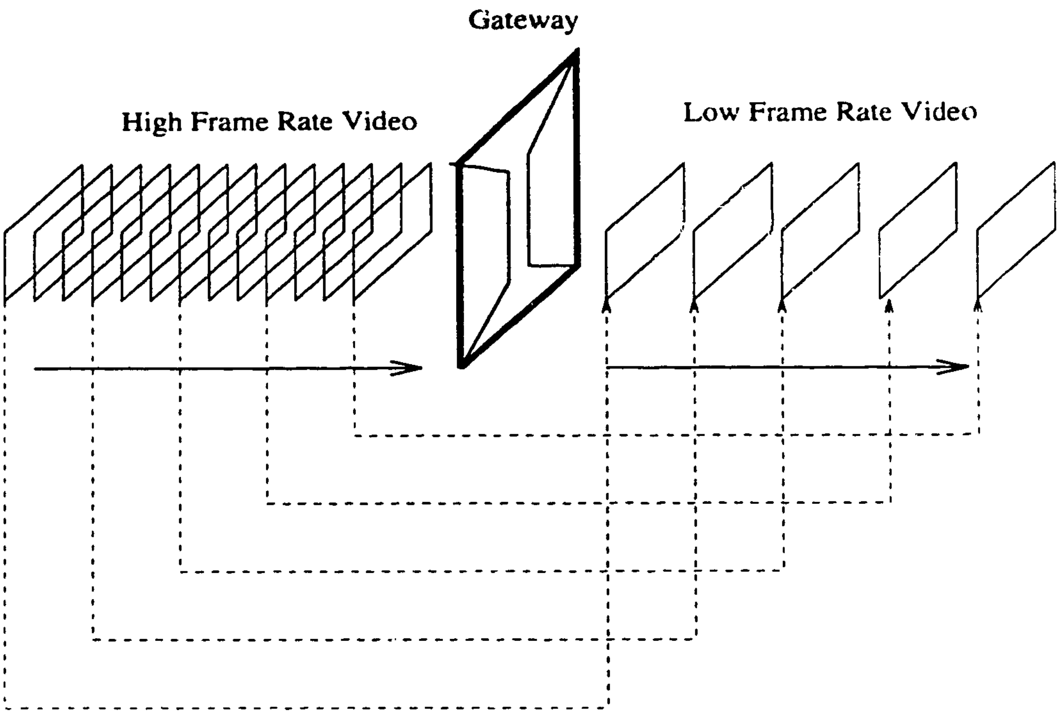


Figure 16: Temporal Subsampling

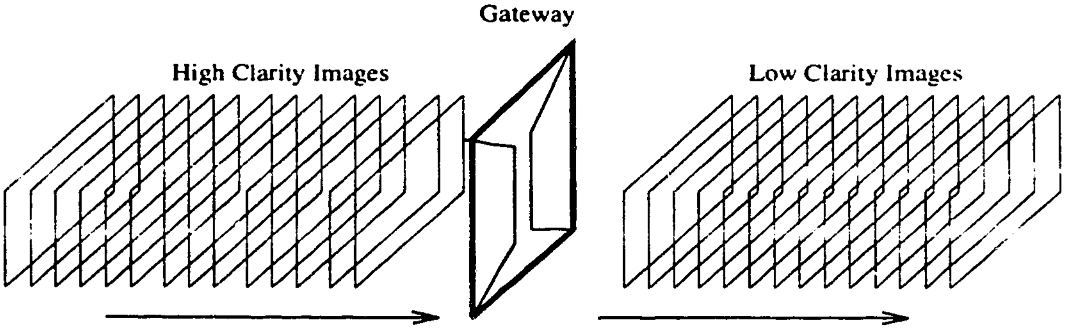


Figure 17: Amplitudinal Selection

Transform Coefficients	:	D 7 0 5 2 3 5 9 6 0
Frequency Truncation	:	D 7 0 5 2 0 0 0 0 0
Minimum Distortion	:	D 7 0 5 0 0 0 9 6 0
Energy Threshold	:	D 7 0 5 0 0 5 9 6 0

The break point used is 4

Figure 18: Data Partition

of frequency truncation, the top 4 values in minimum distortion and anything above 4 in energy thresholding. In amplitudinal *selection* the *selected* coefficients are more important than the ones not selected, considering their significance in visual importance. One can always achieve greater and greater reduction in bit-rate by filtering off more and more of the coefficients. But in so doing, it is advisable to throw away the most visually unimportant data first. The idea behind this technique is called *graceful degradation*. which means the least amount of degradation possible in the quality of the output video. for a given amount of reduction in the output bit-rate. Comparison of these three partition techniques (not with an intention to discard the less significant part. but to send it as low priority data) as applied to MPEG video has been done in [COMP 94].

4.3.2 Video Gateway Designs

1. **Pel level gateway** : This is the simplest possible design. The idea is to decode the incoming video completely, and to reencode the resulting frames. In so doing, frames can be skipped, reduced in size, or converted from color to greyscale. Any combination of these operations can be carried out right at the top level, i.e. when the complete frames in the pel domain are available. Fig.

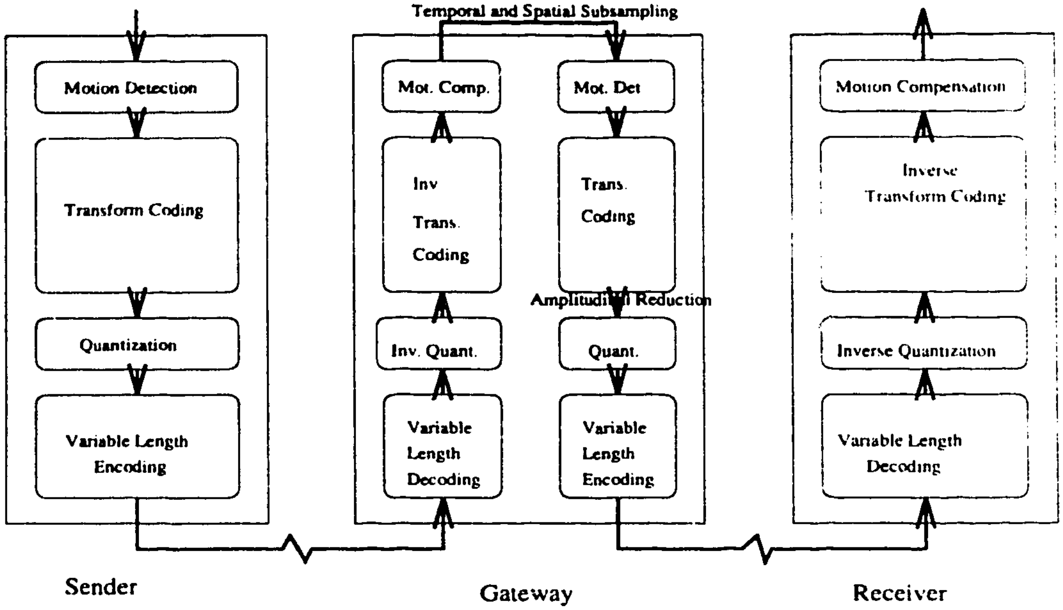


Figure 19: Pel-level Gateway

19 shows the area of action of such gateways. As the re-encoding stage in the gateway is exactly like fresh encoding, except for the fact that the frames come from the decoder and not from the camera, one can also attempt to make the motion detection part less sensitive to motion than it was when the video was originally encoded.

Note that, if further reduction in bit rate is required, it can be achieved by operating on the video in between the stages of the decompression process. For instance, after the transformation of the motion blocks to the transform domain, coarse quantization and/or data partition can be performed on the transform blocks.

The choice of the exact technique used in a gateway will have to vary based on the application. If the fidelity of individual frames is important, no gateway-related processing is advisable after the transformation stage. On the other hand, if the continuity of the video is more important than the fidelity of the individual images, then one can play with the clarity of the frames and perhaps even the frame-size, but not with the frame rate.

Merits :

- Simplicity of design: A pel-level gateway is simply a combination of a decoder and an encoder (a codec, in the reverse, in terms of operation) It is just that the input to the encoder comes from the output of the decoder, not from the camera.
- Wide choice of rate-reduction methods: Any combination of all possible rate-reduction methods (temporal subsampling, spatial subsampling and amplitudinal selection) can be applied. The first two can be done in between the output of the decoder and the input of the encoder. Amplitudinal reduction can be done once the forward transform is over.
- Low Degradation of picture quality with rate reduction: Reduction in the output bit-rate can be achieved without degrading the quality of individual frames (instead by modifying the frame rate or size). PLGs can therefore be used where the fidelity of the pictures is important.

Demerits :

- Low maximum input bit-rate: As the standard codecs code most of the frames in the INTER mode, decoding a frame invariably requires the previous frame. For this reason, irrespective of the desired bit- or frame-rate at the output of a gateway, all incoming frames have to be decoded. This together with the encoder, turns out to be highly CPU-intensive. With increase in input bit-rate, this leads to running out of CPU time to stay within the real time constraints of the gatewaying process. If bit-rates more than this limit are desired, then a full decoder-coder gateway is not helpful. This is of course a limitation only in software implementations of the gateway. If a wide range of output bit-rate (with values very close to the input rate) is required and the input bit-rate is high, only hardware versions of this type can help.

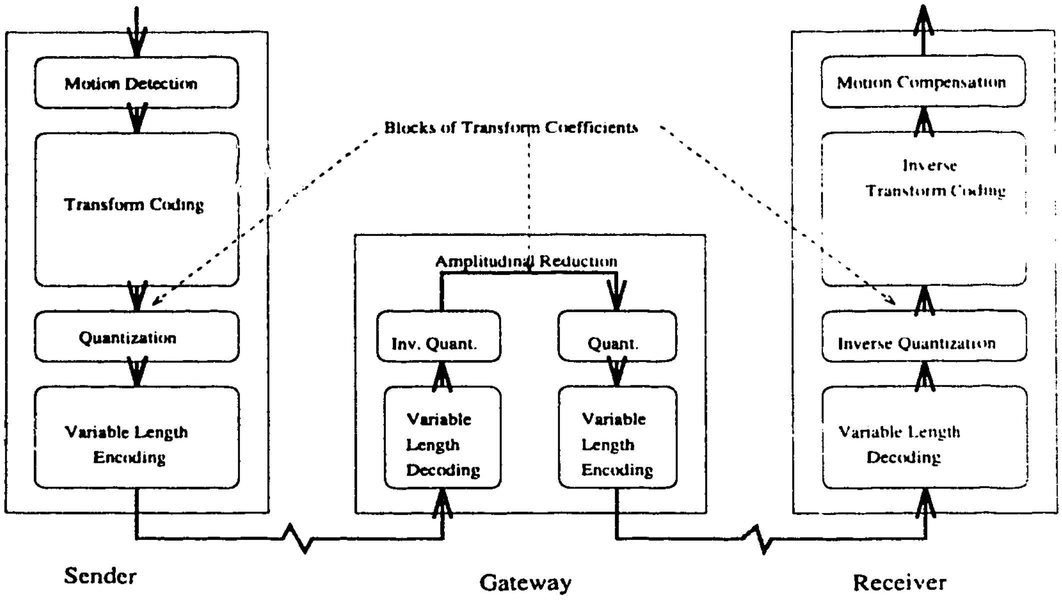


Figure 20: Transform-level Gateway

- High delay: Packets have to be accumulated until the complete frame is available. Because of this the delivery of each frame at the receiver is delayed by the time taken by the gateway to decode and encode that frame. This delay is important because it introduces difficulty in inter-media synchronization at the receiving end, as the audio samples will arrive ahead of their corresponding video frames.

2. **Transform-level gateway** : A transform-level gateway is one that does not perform forward or reverse transformation. Its operation is restricted to the transform domain. As it does not operate on frames, there is no accumulation of packets and consequently, there are no long delays introduced by the gateway.

Merits :

- High maximum input bit-rate: Because of the elimination of the transformation process, it takes an increased bit-rate at the input (in comparison with pel-level gateway), for the CPU to fail to serve both the encoder and the decoder within real time constraints.

- Low delay: The delay introduced in the video frames by the transform level gateway is much less than that introduced by the pel level gateway. This is because the transform level gateway operates on blocks (as opposed to frames), and therefore can send packets as soon as all the blocks in the packet are processed. Besides, the elimination of transformation also saves time.

Demerits :

- High Degradation of picture quality with reduction in output bit-rate: There are not many variables to modify. Amplitudinal selection is the only option. This directly shows up in the degradation of the quality of individual images, although the continuity of the video and the size of the frames of the outgoing stream will be the same as those of the incoming stream. Therefore, this method does not suit applications in which the quality of individual images is more important than the frame rate.
- Narrow choice in rate-control methods: Since each incoming block has to be transmitted, there are not many methods to apply in reducing the output bit-rate. The only things that can be modified are
  - the number of transform coefficients transmitted for each block
  - the number of significant bits of each transform coefficient that is transmitted (in other words, the coarseness of quantization)

3. **Hybrid Gateway** : In the existing codec standards (H.261 and MPEG), motion is extracted from the frame in the pel domain and then transformed to the frequency domain. This is useful, as the transformation is computationally very intensive, and doing the motion extraction before the transformation reduces the amount of transformation to be done. If transformation (of all blocks from the pel domain to the frequency domain) is done before motion extraction, then the concept of frames continues to exist even after the transformation (i.e. at

the transform level). It now becomes possible to do spatial and temporal subsampling in a gateway at the transform level itself (which means the forward and inverse transform operations are not necessary in the gateway). If we were to operate on frames at the transform level, ways to do motion detection at the frame level are needed. That motion detection can be done in the transform domain has been already shown ([DCT 89]). In fact, there are significant advantages in doing motion detection at the transform level :

- *Elimination of spurious motion detection :* The detection of spurious motion, which is unavoidable in pel level detection, due to the noise introduced by quantization, can be *completely avoided* by doing the detection after the transformation ([DCT 89]). The idea is to finely quantize a certain number of transform coefficients on the significant side, and to use only these coefficients in detecting motion.
- *Fine control over sensitivity to motion :* It is possible to choose the degree of sensitivity to motion, by limiting the frequency components that are compared while looking for motion. This can help in changing the output rate. This sensitivity, can be made to conform to the requirements of graceful degradation, by including frequency components in the increasing order, as the available output bandwidth increases.

This means that the transform level gateway can now do anything that a pel level gateway can, only faster. The price for this, is paid by the coder at the sending end and the decoder at the receiving end of the video, due to additional computation required in moving full individual frames between the frequency and the pel domains. Within the domain of asymmetric applications, however, the price paid can be considerably reduced. Recorded video can be made to undergo a post-processing stage, when it can be decoded and then re-encoded, this time doing the motion detection after the transformation. This has to be done only once and does not have any real time constraints. The price is now reduced in half; the decoder still has to cope up with transforming all blocks from the transform domain to the pel domain. Here again, a considerable elimination of the load is possible, if the decoder were to be in hardware. As the inverse transform of different blocks are data-independent (as is the forward transform), the transformation process can be performed parallelly for all blocks, thus reducing the complexity of the process.

Merits:

- Computational complexity is reduced (in comparison to pel-level gateways) because of the removal of the transformation stage.
- A choice as to which reduction techniques(s) to use, is available: Accordingly, depending on the requirements of the application, it is possible to

  (a) operate on frames, if the clarity of the output pictures is more important than the delay introduced by the gateway

  (b) operate on packets, if low delay is more important than clarity

  A hybrid gateway may also switch between the above two modes dynamically.
- There are advantages in performing motion extraction at the transform level (immunity to quantization noise and the ability to vary sensitivity to motion based on visual importance)

Demerit:

- Incompatibility with the standard codecs : As the hybrid gateway requires that motion detection be done after the transformation stage, it is incompatible with the existing standards like H.261 and MPEG.

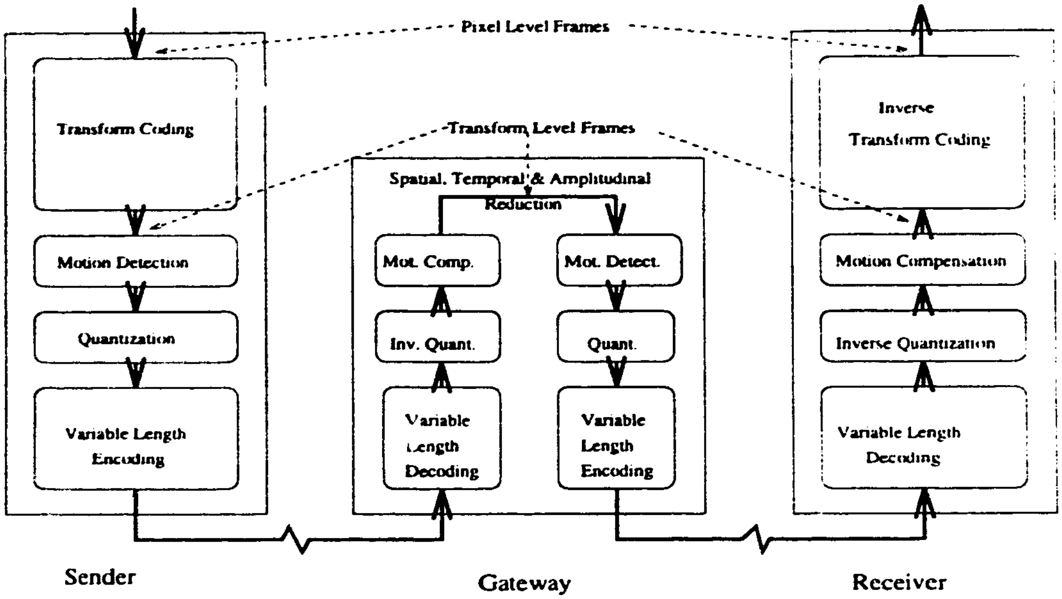


Figure 21: Hybrid Gateway

4.3.3 Video Gateway Implementations

We present here some empirical results from building gateways. The implementations are compatible with the public domain video codec *nv* ([NV 94]), used widely on the MBONE ([MBone 93]). *nv* uses Haar Transform for transform coding (as opposed to DCT which is used in the standard codecs MPEG and H.261. Computing DCT coefficients is a more intensive process that computing Haar coefficients.) All frames except the first are INTER encoded. Bandwidth control is enforced by transmitting packets into the network at appropriate intervals, and not grabbing a new frame until all packets are sent. Background data is replenished continually in order to make up for packets that may have been lost and for facilitating ‘tuning in’ in the middle of a video sequence. An INTRA encoded frame is sent every time a parameter (maximum allowed output bit rate, color/greyscale, size) is changed.

Experiments on the first two of the above three designs - pel and transform level gateways, were done. The implementation of the gateways was done for the following purposes :

- the gateways can be used in the prototype of the media-on-demand system developed as part of this work (described in Chapter 5)

- the gateways can be compared with each other to determine which one offers better quality video at a given bit-rate, and at what cost (*Quality* stands for frame-rate, frame-size and clarity of individual frames; *Cost* stands for delay encountered at the gateway and the CPU requirements of the gatewaying process)

- **Pel level gateway**:
  Fig. 22 shows our implementation of a pel level gateway. It operates on frames and therefore has to wait until all packets that belong to one frame are decoded, before beginning the encoding. As almost all frames are INTER encoded, no matter what the output frame rate is, all incoming frames have to be decoded. The implementation consists of decoding the frame (which in turn is run-length decoding followed by inverse Haar Transform) and encoding (which in turn consists of motion detection, forward Haar Transform and run-length encoding.

- **Transform level gateway**:
  Fig. 23 shows our implementation of a transport level gateway. As it operates on packets, the decoding and encoding (which comprises only of run-length decoding and run-length encoding) of each packet is done immediately after the packet arrives, in that order.

- **Experiment Equipment and Conditions**
  Experiments were performed with the equipment available in the Communication Networks Laboratory, over the available 10 Mbps Ethernet. For all experiments the size of the picture used was 320 X 240. Video was transmitted to the multicast address 224.2.0.1 so it could be received on any machine on the same LAN (including the sender), to facilitate discerning the delay introduced by the gateway, if human-perceptible and to generally be able to see the output from the gateway and its input from any machine other than the ones serving as the gateway and the original sender. The SGI Indy machine nestow was used

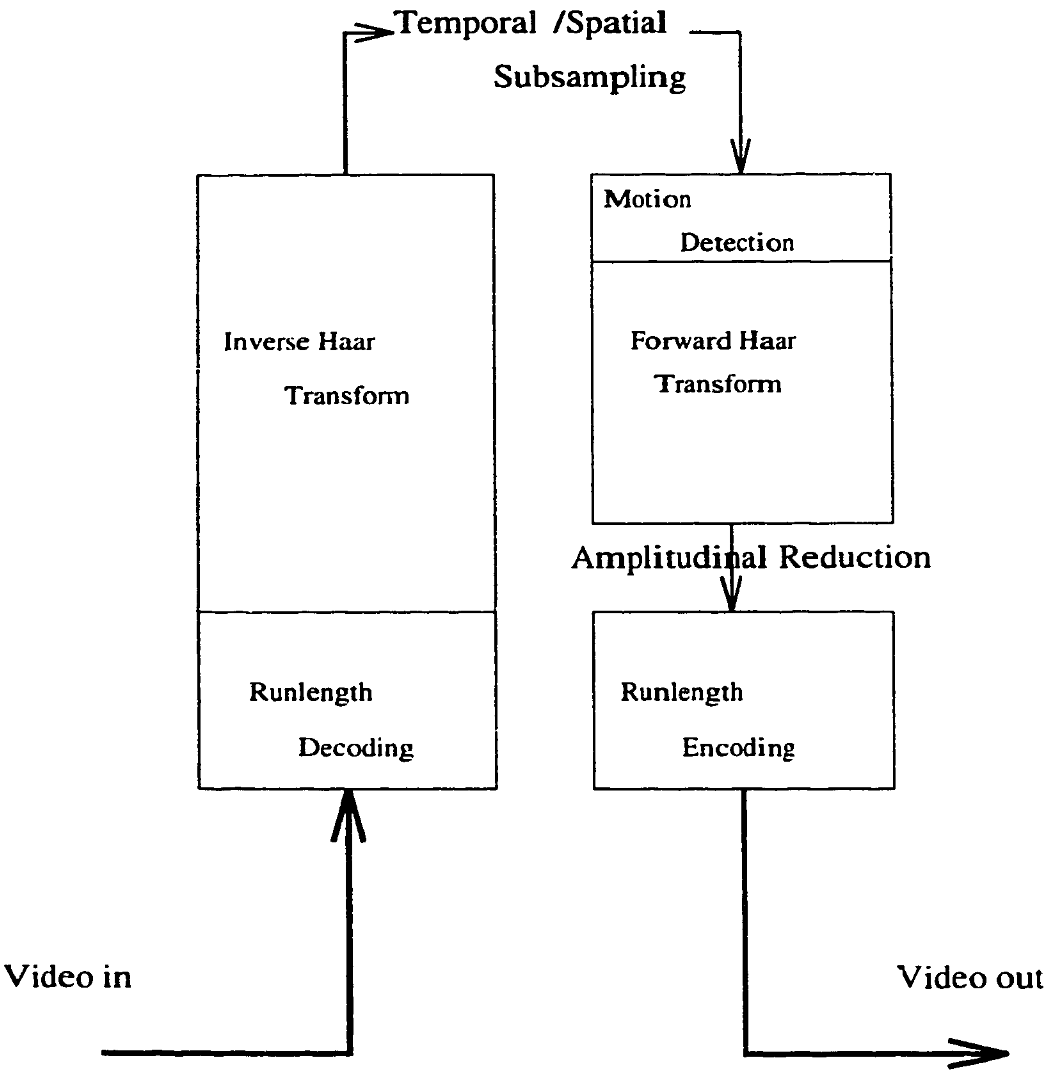


Figure 22: Pel-level Gateway Implementation

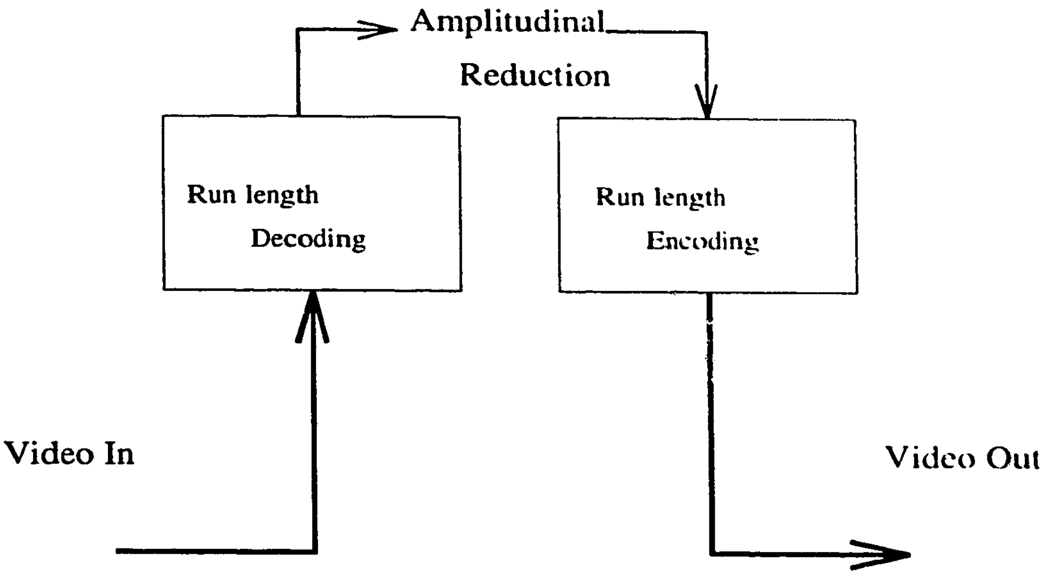
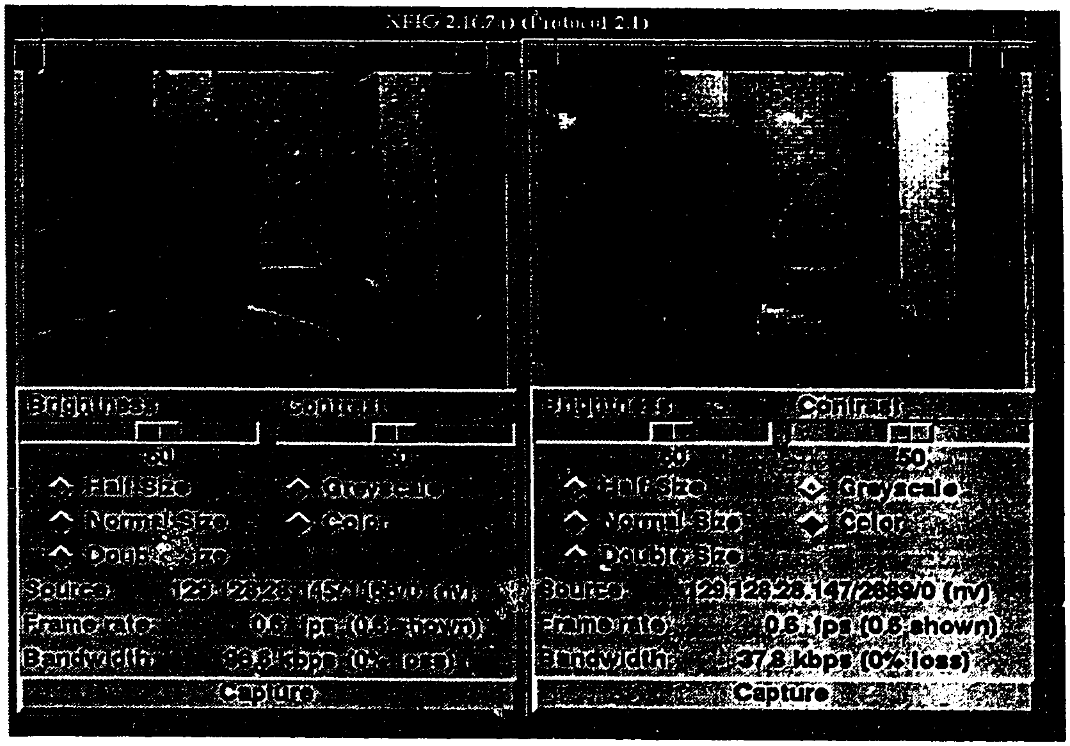


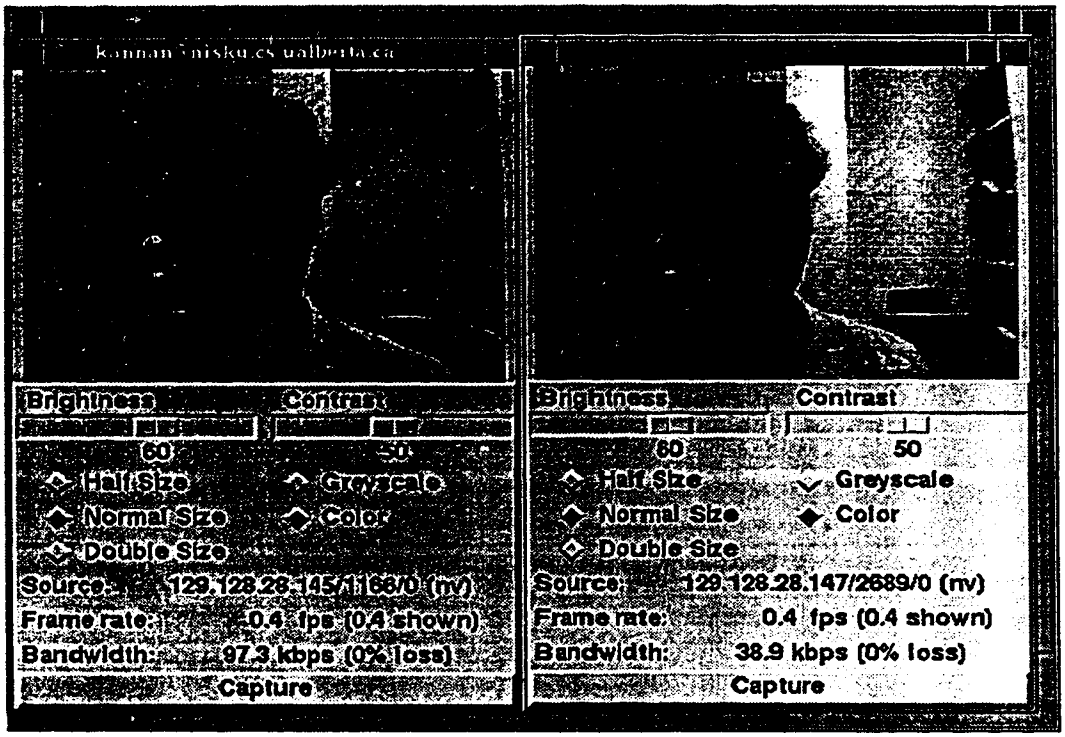
Figure 23: Transform-level Gateway Implementation for nv-video



Brightness 50 Contrast 50
Source: 129.128.28.[illegible] (nv)
Frame rate: 0.6 fps (0.6 shown)
Bandwidth: [illegible] kbps (0% loss)

Brightness 50 Contrast 50
Source: 129.128.28.147/2889/0 (nv)
Frame rate: 0.6 fps (0.6 shown)
Bandwidth: 37.8 kbps (0% loss)

Figure 24: Snapshot : Input and output video of a transform level gateway, as seen from the sender/receiver



kannan@nisku.cs.ualberta.ca

Brightness 60 Contrast 50

◇ Half Size ◇ Greyscale

◆ Normal Size ◇ Color

◇ Double Size

Source: 129.128.28.145/1166/0 (nv)

Frame rate: 0.4 fps (0.4 shown)

Bandwidth: 97.3 kbps (0% loss)

Capture

Brightness 60 Contrast 50

◇ Half Size ◇ Greyscale

◆ Normal Size ◆ Color

◇ Double Size

Source: 129.128.28.147/2689/0 (nv)

Frame rate: 0.4 fps (0.4 shown)

Bandwidth: 38.9 kbps (0% loss)

Capture

Figure 25: Snapshot : Input and output video of a pel level gateway, as seen from the sender/receiver

as the gateway. and nisku as the sender (and receiver) of the video. The video was sourced by the side-to-side full-body motion of a person seated opposite to nisku at about 3-4 feet from the camera mounted on nisku, on a rotating chair. It has to be noted that neither of these machines (nisku and nestow) have video compressing or decompressing hardware.

- **Empirical results:**

For the gateways implemented, the following quantitative aspects were observed.

1. *Delay:*

   A pel-level gateway operate on a per-frame basis. A transform-level gateway operates on a per-packet basis. Therefore, in the case of transform level gateways, the per-packet delay is the same as the per-frame delay. The delay introduced by the pel-level gateway was found to vary with the frame-rate at its input, whereas the delay due to the transform-level gateway was not affected by the input frame-rate.

   Figures 26, 27, 28 and 29 show the per-frame delay introduced by the pel-level and transform-level gateways, for three input frame-rates - 2 fps. 3 fps. 4 fps and 5 fps respectively. The same input video stream (320 X 240. *nv* encoded) was fed to both gateways.

   The values were noted for 100 consecutive frames. During delay measurement. no bit-rate reduction was being done by the gateways. In other words, the pel-level gateway decoded the incoming frames and reencoded them in exactly the same way as they were originally encoded: the transform-level gateway partially decoded the data in each packet (to be specific, run-length decoded the data) and reencoded it.

   Fig. 24 is a snap shot of the input and output from the transform level gateway, as observed from the original sender of the video (which served as the receiver as well). The pictures are identical (not considering clarity for now), showing that the gateway has taken no observable time in doing its

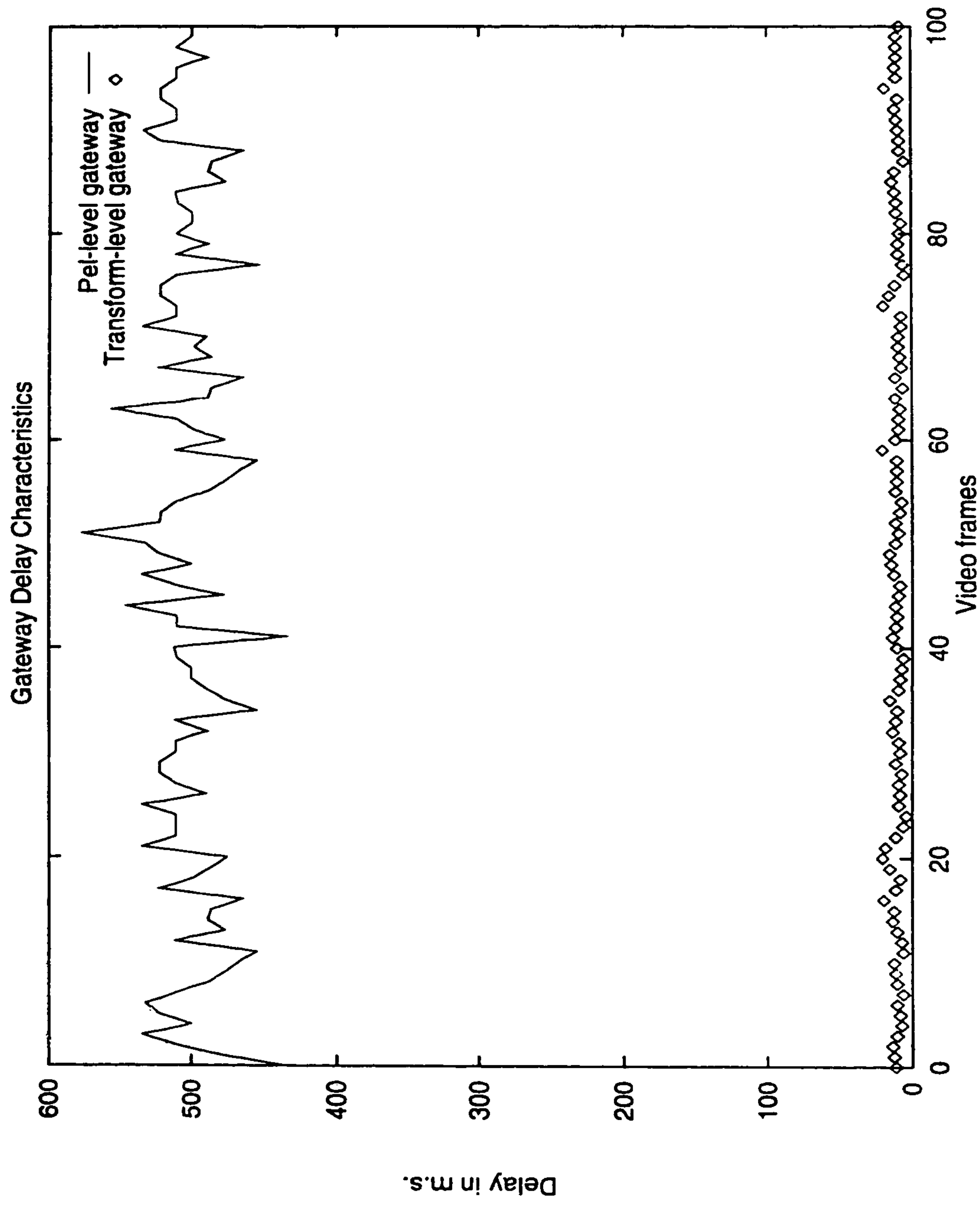


Figure 26: Delay due to gateway (Frame size : 320 X 240, Frame rate : 2 fps)

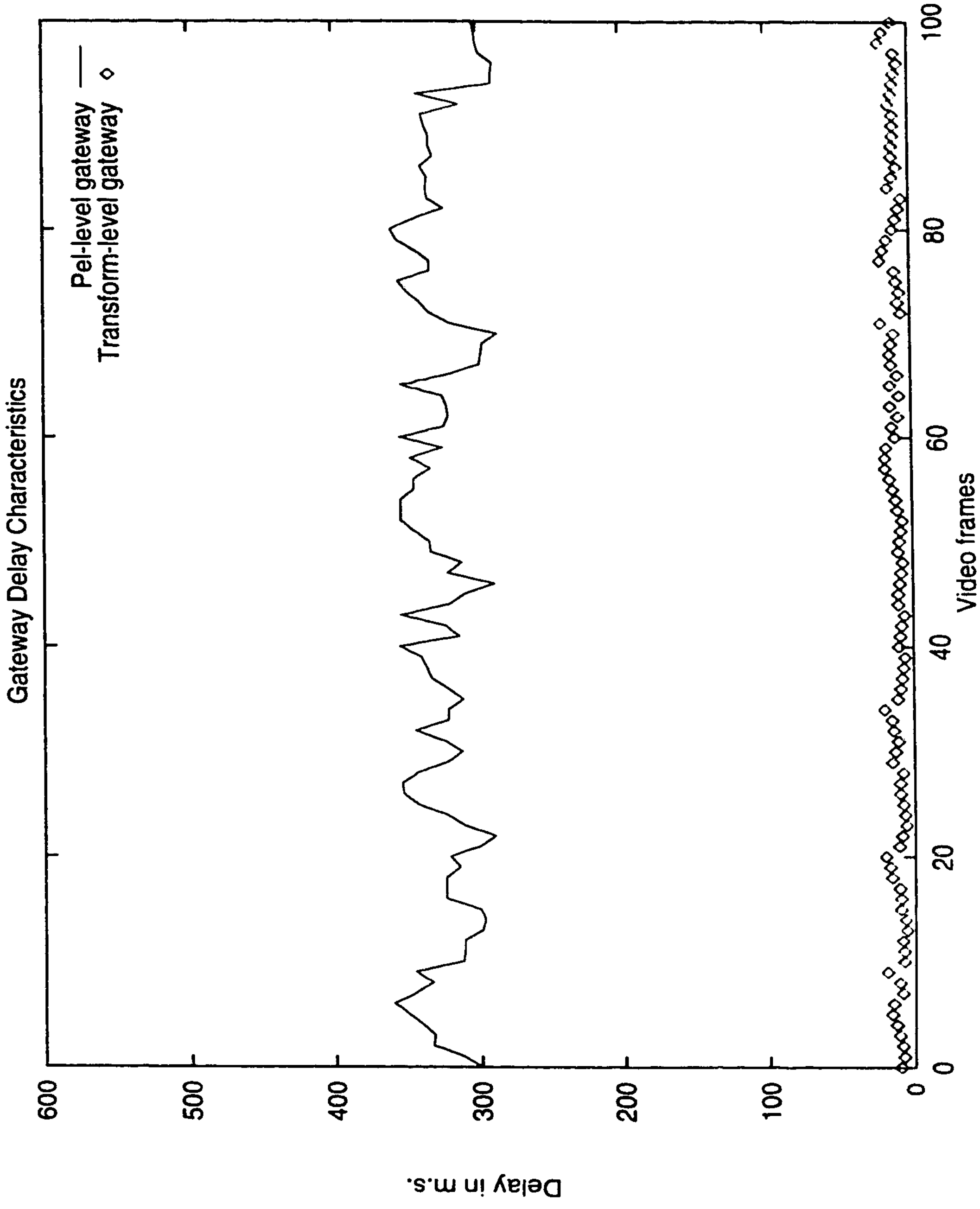


Figure 27: Delay due to gateway (Frame size : 320 X 240, Frame rate : 3 fps)

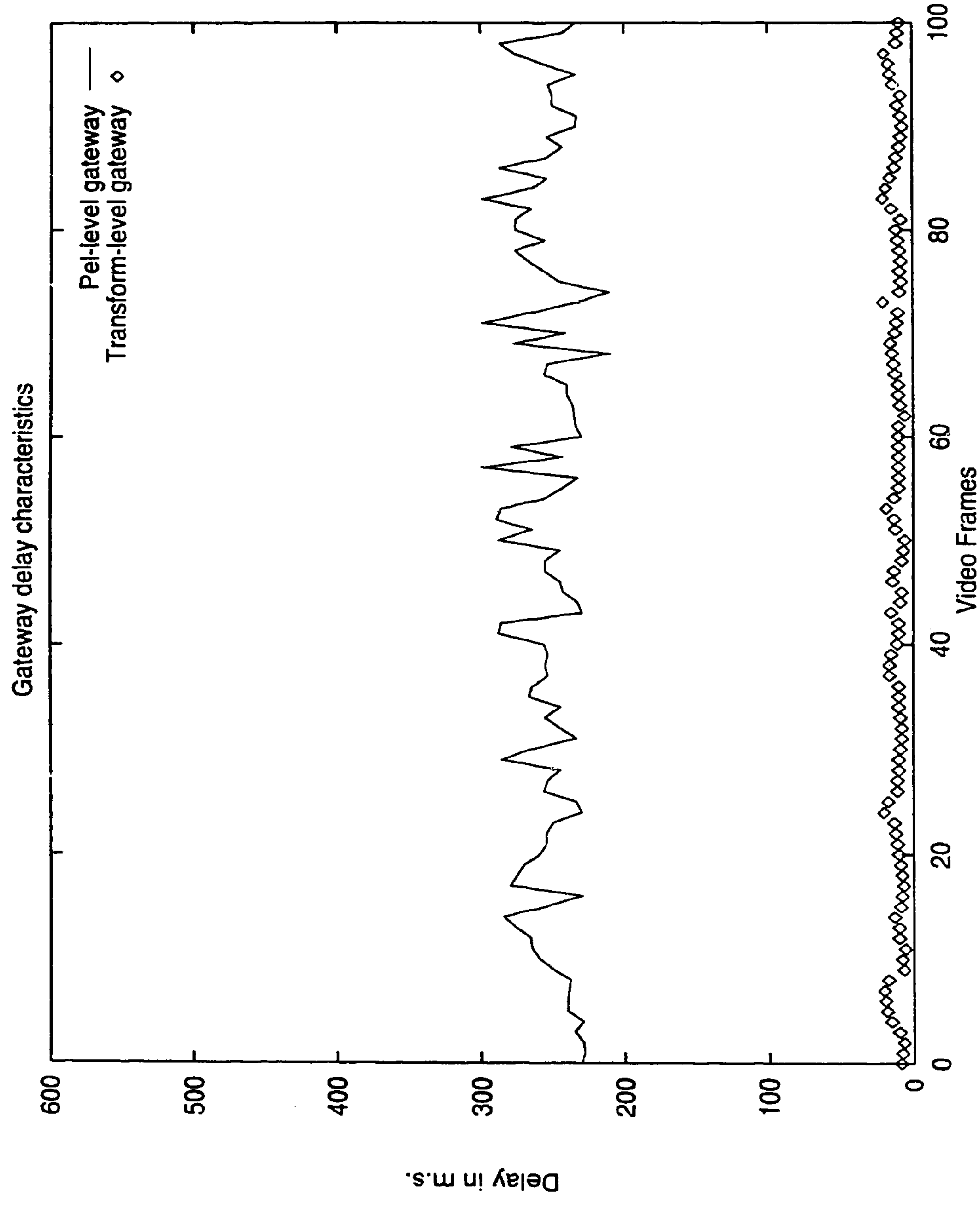


Figure 28: Delay due to gateway (Frame size : 320 X 240, Frame rate : 4 fps)

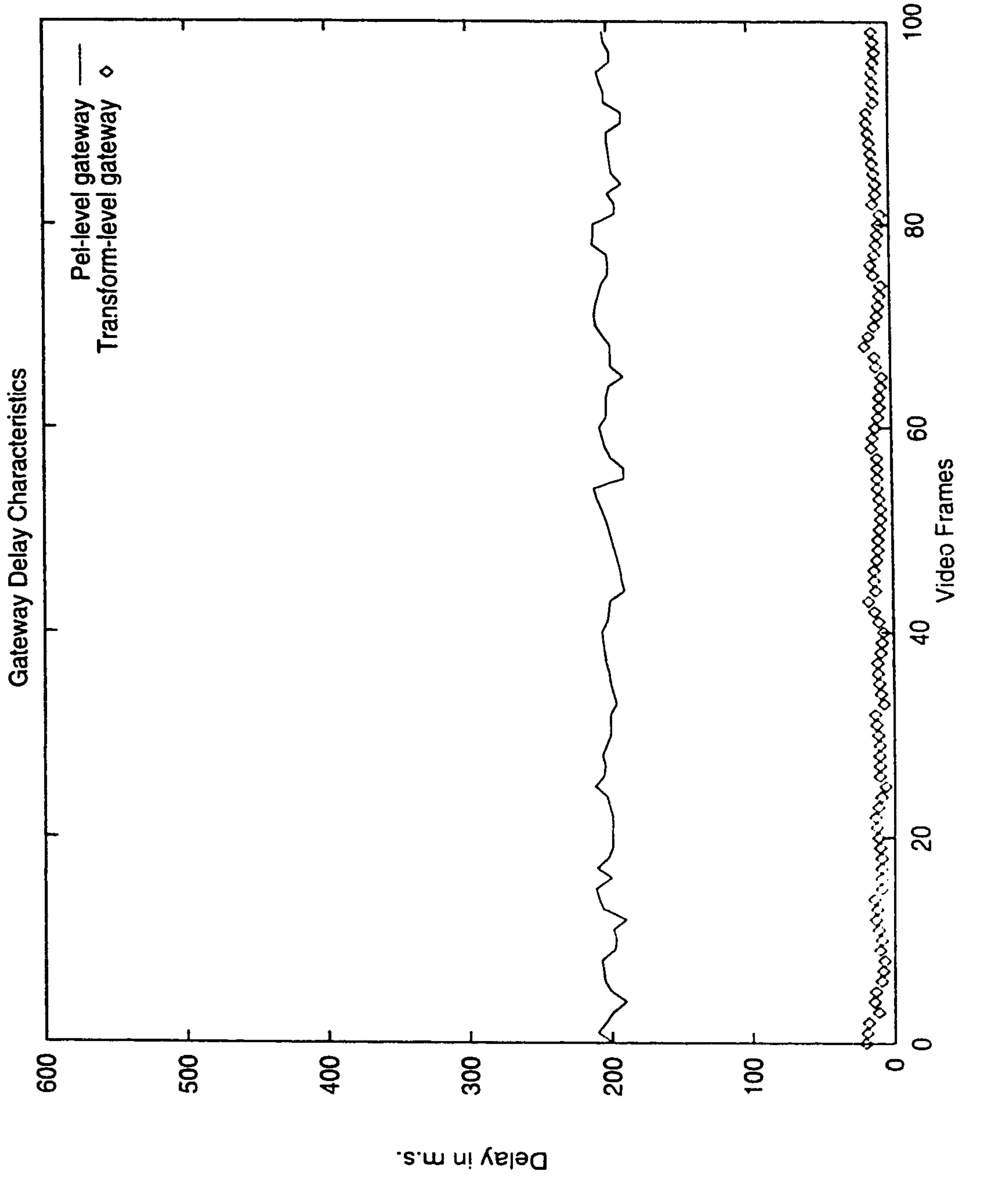


Figure 29: Delay due to gateway (Frame size : 320 X 240, Frame rate : 5 fps)

work (The delay is not human-perceptible). But in fig. 25, which shows a a snapshot of the input and output of the pel-level gateway, there is human-observable delay in the video on the right hand side (evident from the difference postures of the person in the picture).

2. *CPU intensity:*

   The computational intensity of the pel level gateway is more than double that of the transform level gateway; besides the former increases at a higher rate than the latter with increase in the input bit-rate. Figures 30, 31, 32 and 33 show the CPU time used by the gateways as a percentage of the real time elapsed during the period of measurement, for various input bit-rates. The period of measurment used was one second (A value 74 for the y-coordinate means that 740 milliseconds of the last 1000 milliseconds were used by the gateway software). This was carried on for a minute. Both gateways were simultaneously tested using the same input video stream.

3. *Quality:* There is no single established measure for the quality of motion-video. Therefore we resort to presenting some snapshots of inputs and outputs from the gateways.

   Fig. 25 shows a snapshot of the input and the output of the pel-level gateway. The input video is shown on the left. The peak bit-rate at the input was 128 Kbps. The peak bit-rate at the output was set to 40 Kbps. The clarity of the output picture is the same as that of the corresponding frame input to the gateway. Fig. 24 shows that the video output (on the right) by the transform level gateway is reduced in quality in comparison to the original.

4. Bit-rate Both gateways can operate at any given output bit-rate (i.e. from 0 Kbps to the input bit-rate).

- **Analysis of the results**

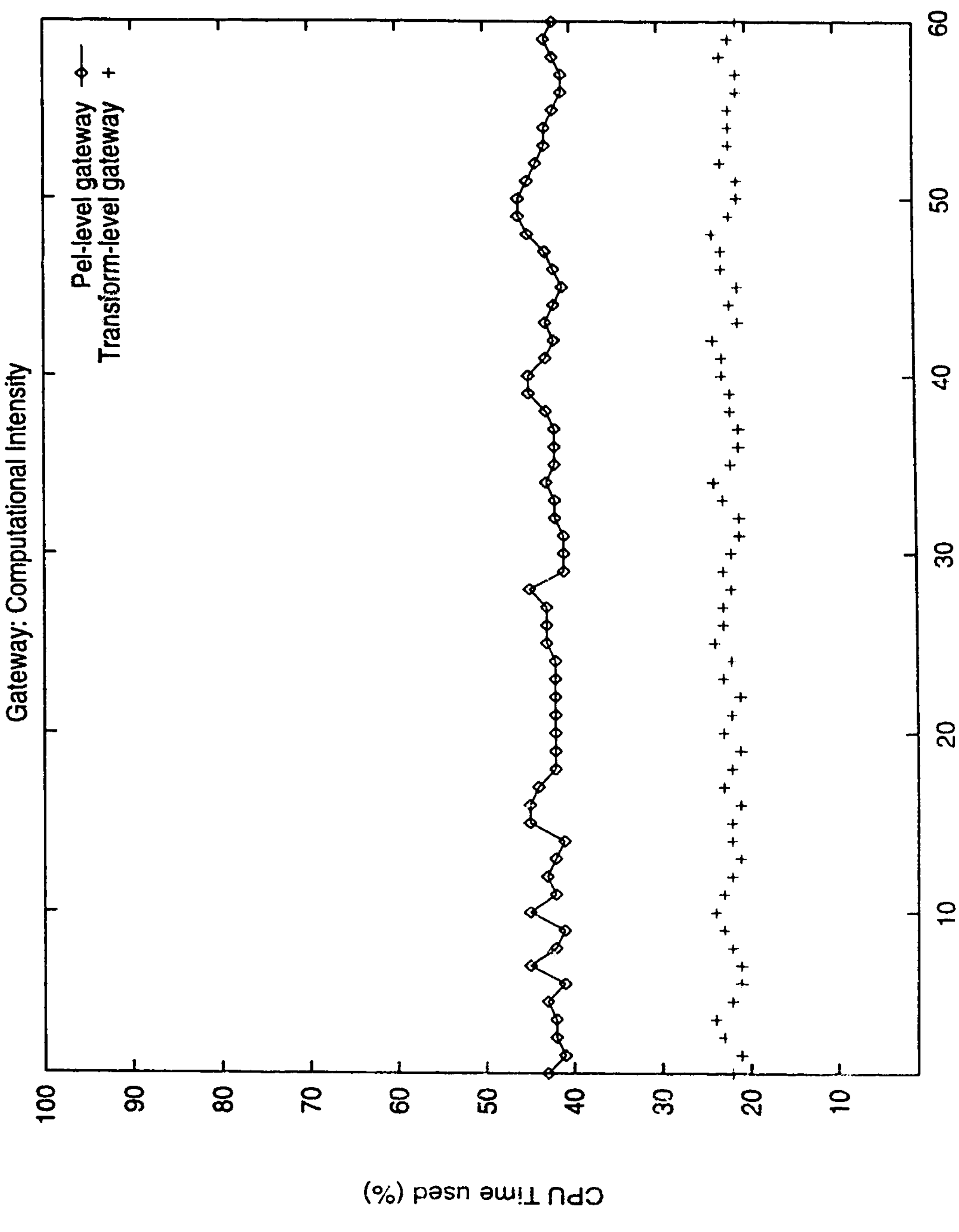


Figure 30: CPU utilization (Input bit-rate=200 Kbps)

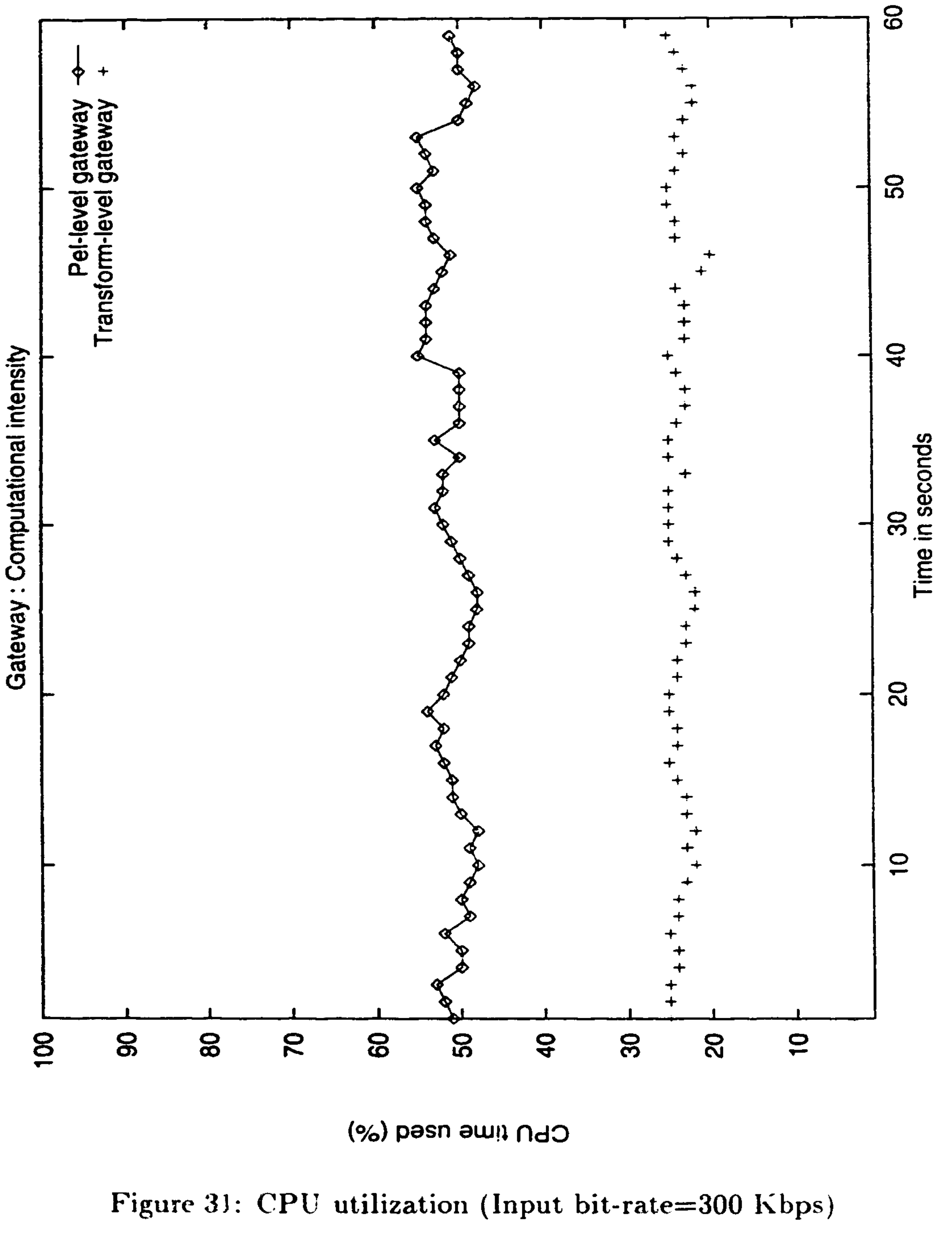


Figure 31: CPU utilization (Input bit-rate=300 Kbps)

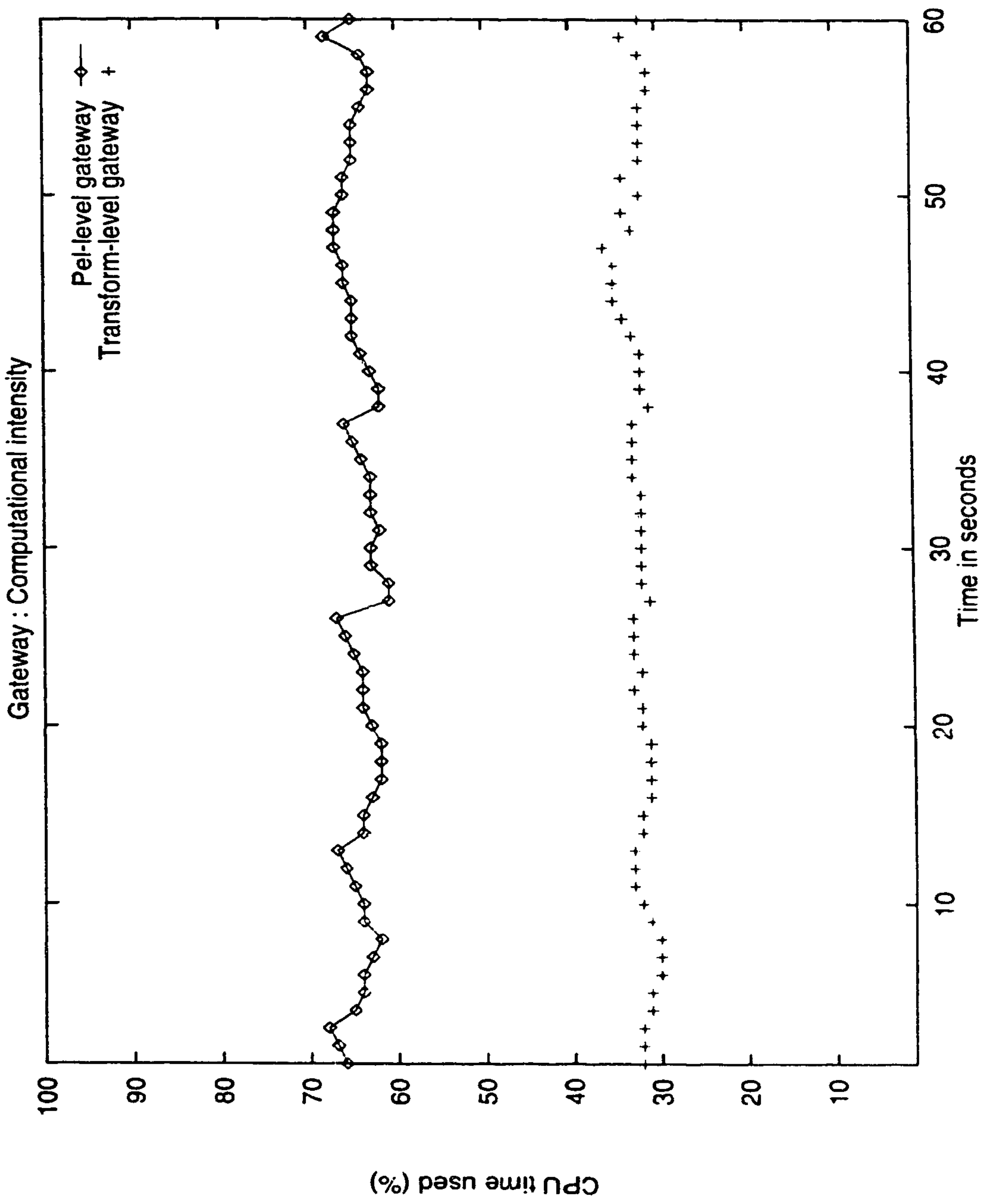


Figure 32: CPU utilization (Input bit-rate=400 Kbps)

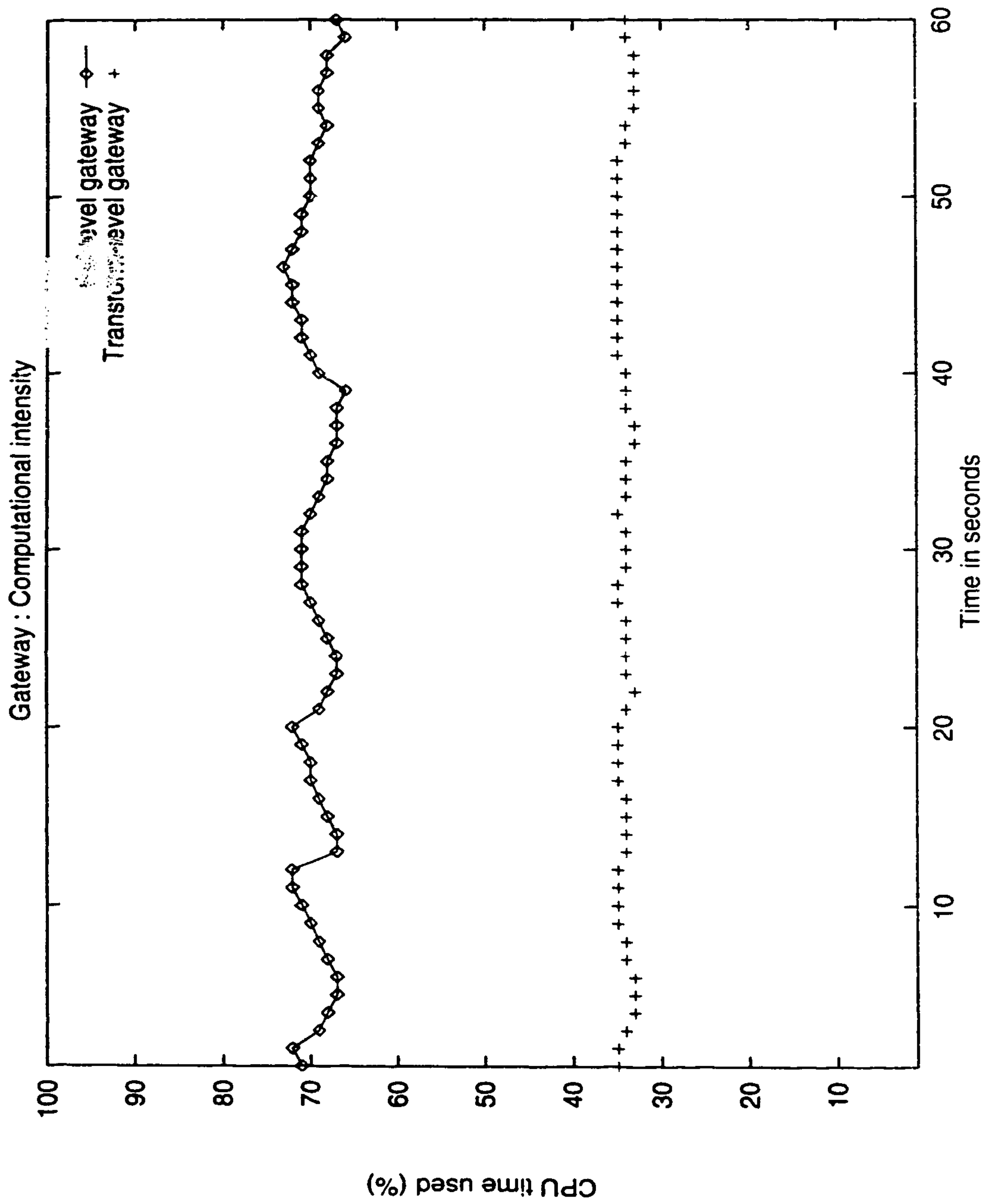


Figure 33: CPU utilization (Input bit-rate=500 Kbps)

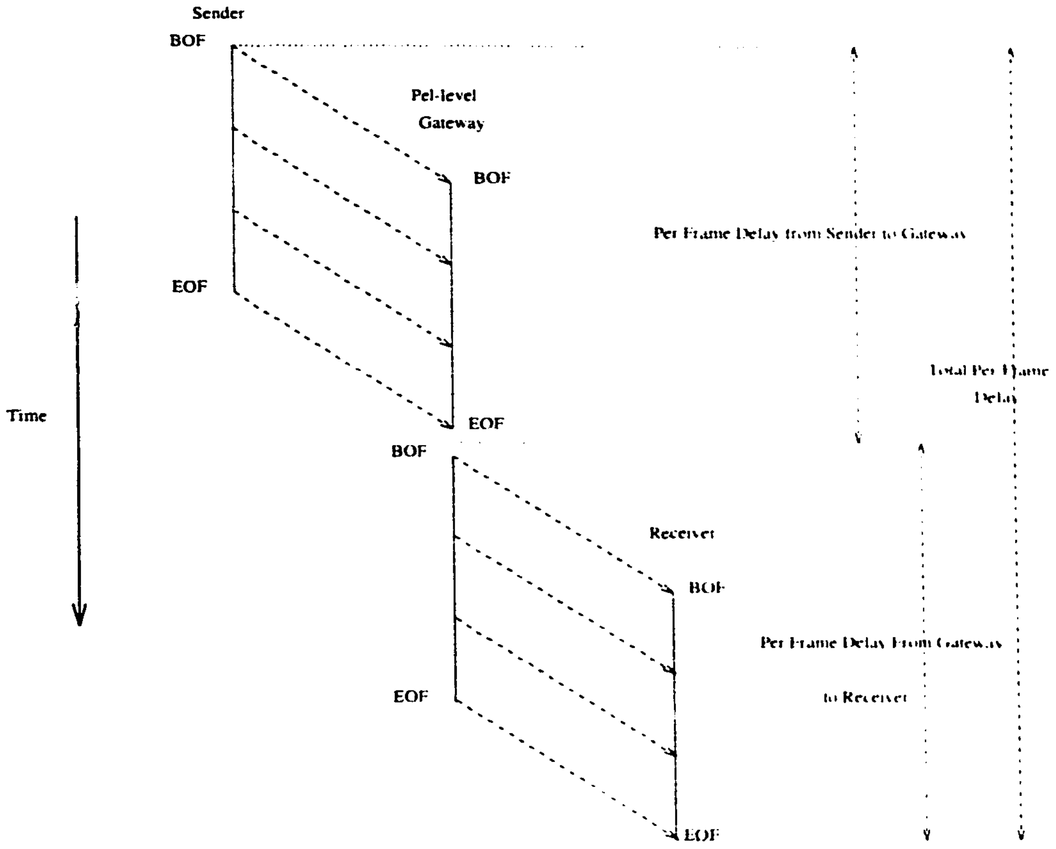


Figure 34: Per-frame Delay in Pel-level Gateway

1. *Delay:* Delay in the transform level gateway, as seen in figures 26 to 29. is due entirely to the time taken by the run length decoder and run length encoder. In the case of the pel level gateway, the encoder does not start work, until the whole frame is decoded (Fig. 34). The total real time required for decoding depends not only on the decoding routines but also on the input frame-rate. At low bit-rates, the frame-rate is also low, and the total frame-decoding time is more dependent on the frame-rate than it is on the time taken by the decoder to decode every packet. From this it follows that the delay introduced by a pel-level gateway is inversely proportional to the input frame-rate. The transform level gateway, however, operates on packets as and when they arrive, so the delay introduced by it is the sum

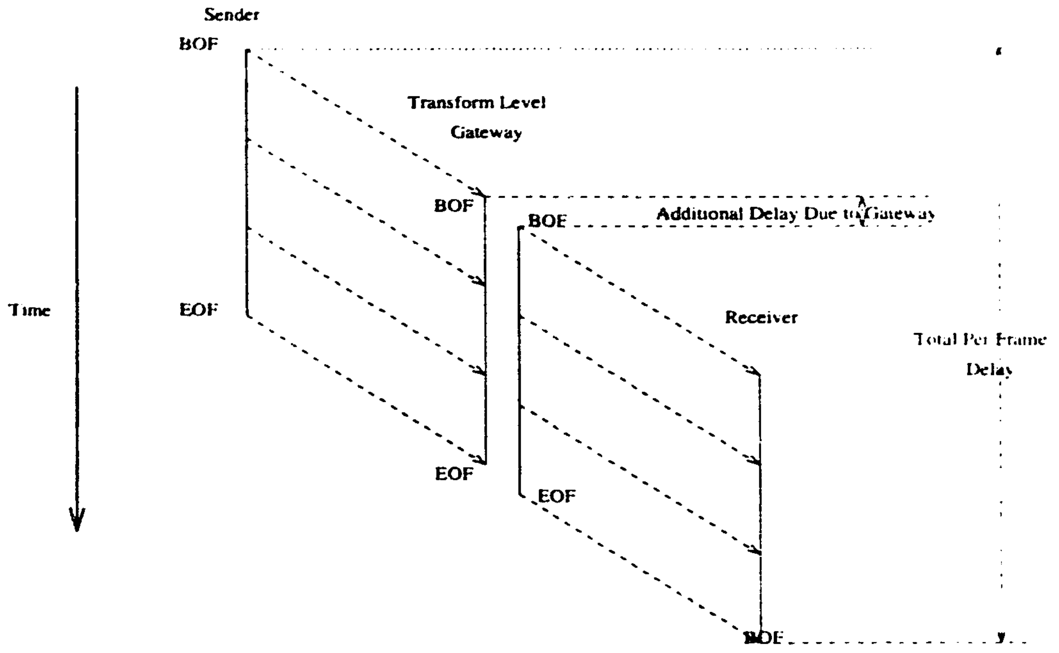


Figure 35: Per-frame Delay in Transform-level Gateway

of the time taken to run-length decode, and the time taken to reencode it. This whole process takes only a few milliseconds and does not depend on the input bit-rate or frame rate.

Fig. 36 shows the theoretical delay characteristics of the two gateways, in the form of delay vs. input frame-rate. As the delay in the pel-level gateway is the time taken for decoding each frame, it is equal to the time taken for each frame to arrive at the gateway, which is the just the reciprocal of the input frame-rate. Note that the delay due to the transform-level gateway is not affected by the input frame-rate. Note how the empirical results shown in fig. 26 through fig. 29 confirm these deductions.

2. *CPU-Utilization:* The only operations that the transform-level gateway does not do but the pel-level gateway does are Forward Haar Transform, Reverse Haar Transform and Image Differencing (The transformation part is the most CPU-intensive of the codec, more so when DCT is used in the place of Haar). In our implementation, the forward and reverse transfor-

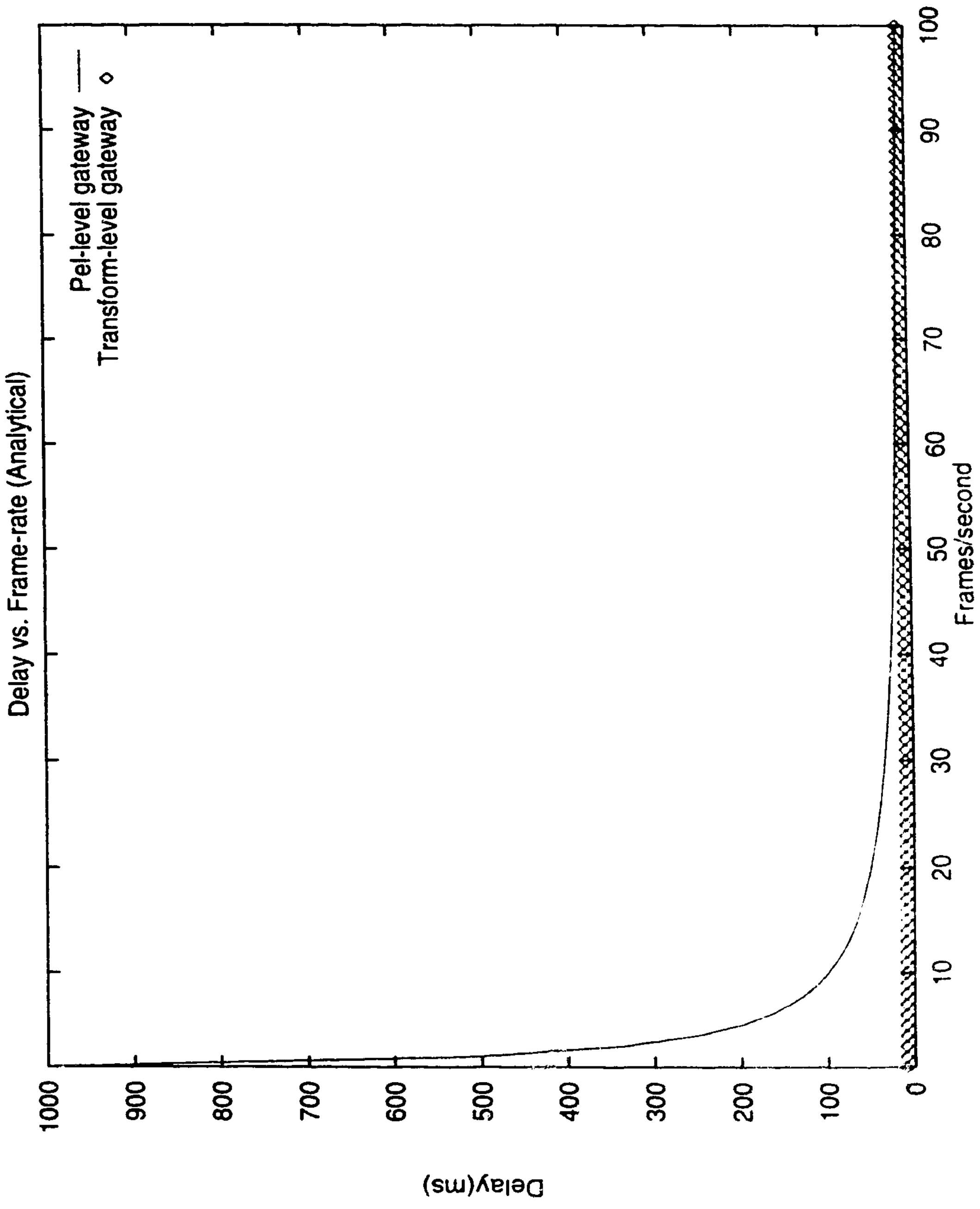


Figure 36: Delay vs. Input frame rate (Analytical)

mations consume roughly half the total CPU time required for encoding and decoding, respectively. Therefore the processing requirements of the transform-level gateway are only half of those of the pel-level gateway.

3. *Quality :* The figures (Fig. 25 and Fig. 24) show that for a given output bit-rate, the pel-level gateway produces better quality frames. Therefore. using the pel level gateway, one can achieve a more graceful degradation of the quality of the output video with decrease in the output bit-rate. than what can be achieved using a transform-level gateway. This is because the transform-level gateway cannot but operate on every block of transform coefficients, which always results in the degradation of the quality of individual frames. The pel-level gateway on the other hand. can choose to reduce frame-rate instead of reducing the clarity of individual frames. It has to be noted that the pel-level gateway is capable of doing more than what the transform-level gateway can, as far as the quality and bit-rate of the output vide are concernedo, as the former includes all modules that make up the latter. When the option to vary frame-rate is required. one has to go for pel-level gateways. If the delay and processing cost of the pel-level gateway outweigh the benefits of having that option (to vary frame-rate), then one should go for transform level gateways.

Quantitative comparison of the quality of the output video from both the gateways (using a measure like Signal-to-noise ratio) requires a tool which operates in real-time, comparing the corresponding frames from both gateways as and when they are output by the gateways. Such a tool also has to resconstruct each frame in its entirety so comparison with another frame is possible. Such complications make such a tool at least as complex as the gateway itself. For these reasons SNR comparison of the video output from the gateways could not be done. However from the design of the gateways and the snapshots of their inputs and outputs presented, it should be possible to see how the gateways relate to each other in terms of the quality of the output video. Our conclusion is that the pel-level gateway offers better output video quality at the cost of higher delay and higher computational intensity in comparison to the transform-level gateway.

4.4 Other applications of Video Gateways

The use of video gateways is not confined to applications involving Video-on-demand. At the heart of the problem (to which video gateways are the solution) are the difference between the video output from the server and the input video expected by the client and the difference in the capacities of various links that constitute the connection. One of these two situations is there in several cases, and hence video gateways form the solution in all those cases.

1. *Unicast : Client-server incompatibility*

   - **Incompatibility of the encoding format** : For both stored and live video, video gateways are required if the decoder of the receiver and the encoder of the sender do not speak the same language, i.e., do not use the same coding technique.

   - **Difference in the bit-rate** : In the case of live video the sender can adapt its coding parameters to provide video at the rate required by the user, by various means, such as increasing the time interval between subsequent frame grabs, decreasing frame size etc. Video gateways are *not* required in this case.

     The bandwidth requirements of recorded video, on the other hand, are fixed. As pointed out earlier, If a client requesting playback cannot satisfy those requirements, a rate control video gateway has to be employed. If the server does not take into consideration the particular bit-rate acceptable to the user in question, then a lot of packets will be dropped somewhere down the line, amounting to wastage of resources (The effect of such situations on end user cost has been investigated in [EOPL 93]). The user will have to pay for the lost packets as well, if the charging scheme is based on the amount of data sent by the server.

2. *Multicast :* **Multicast and heterogeneous branches** : When video (live or orchestrated) is sent down a multicast tree, because of the possible difference in the conditions and capabilities of the branches of the tree, gateways are required at the nodes to compress the video further and send it on the less capable branches.

   - **Multicast on Internet** : It has been pointed out in [MULT 94] that different congestion states of different branches of a multicast tree in the Internet require that the same video stream not be fed to the entire tree. While the source cannot generate all the different rates required by the clients, it is conceivable for the video to be passed through a box at each internal node of the tree, which produces the rates required by the child-nodes of that node. Fig. 37 shows how simulcast works; Fig. 38 shows an improvement - using layered coding; Fig. 39 shows how the situation changed with the use of video gateways.

     Solutions:

     – *Simulcasting :*

       The problem of having to cater to multiple bit-rate requirements in the case of live multicast video is solved *partially* by simultaneously multicasting the same video at two different rates to two different multicast addresses. Users are to choose one from these two rates and join the corresponding address/group. This is the way it is being done on the MBONE now. This method, has the following disadvantages :

       (a) The choice of bit-rate available to the user is limited

(b) For every new bit-rate, additional resources (especially bandwidth) are required

(c) Source has to be concerned with the variety of receiver rate-requirements

– *Layered coding:*

This is also only a *partial* solution. This solution works by sending the video in two layers, viz. **the base layer** and **the enhancement layer**. As the name suggests, the enhancement layer is merely an improvement on the quality of video that can be obtained from the base layer. Therefore, users have a choice - between a low bit rate (if only the base layer is chosen) and a high bit-rate (if both layers are chosen). A layered coding technique termed *spatial scalability mode* is part of the MPEG-2 recommendation. This method is meant to be useful also in simulcasting, and for software decoding (of the base layer). The base layer consists of samples coded at low resolution, and can be used in predicting the higher layers.

– *Using Video Gateways:*

With the use of video gateways, the situation improves as shown in the figure. Bandwidth is used economically, at the expense of additional computation at the machines running the gateways. The source can be free of the wide range of requirements of the multicast group and just produce the maximum of the bit-rates requested by the receivers. Besides, all internal nodes of the multicast tree, have to cater only to the requirements of the immediate child-nodes.

- **Multicast on ATM** : According to ATM UNI 3.0, a host wanting to connect itself to a multicast group in an ATM network cannot specify the QOS of its connection. This restriction can be removed by the use of gateways and by using multiple point to multipoint connections with each video gateway acting as a child of one of such connections and root of another. Figures 40 and 41 show both situations - with and without

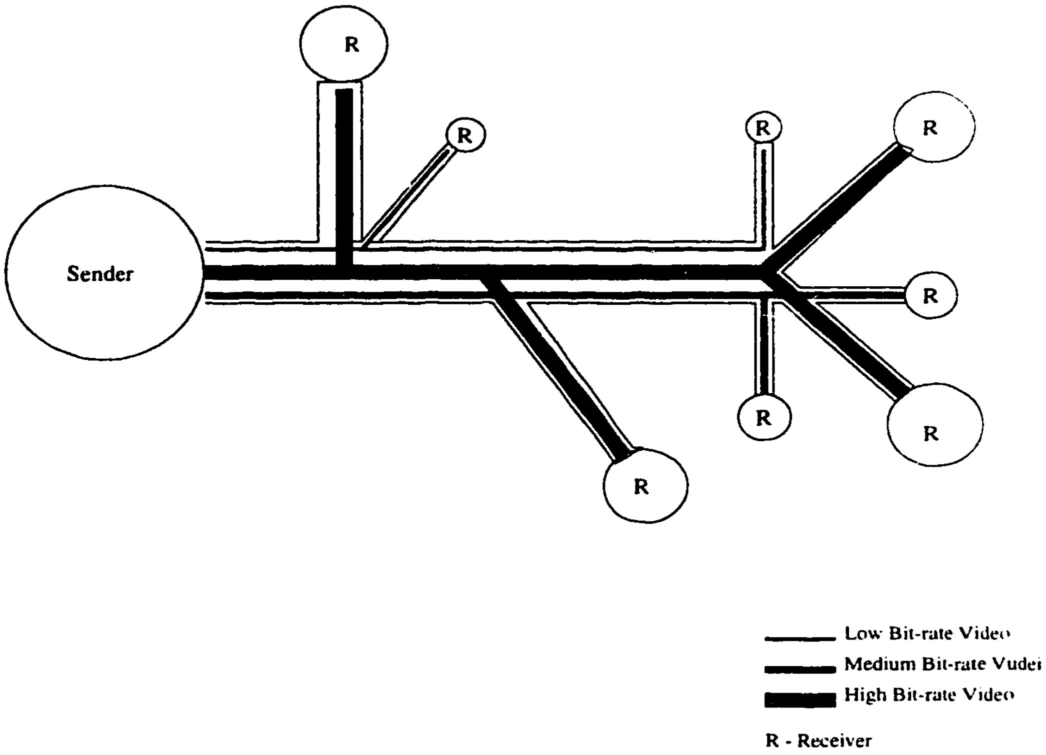


Figure 37: Simulcast : Simultaneous mutlicast of video at different bit-rates

gateways. Gateways can be predesignated machines on the network, or dynamically created as demand occurs.

3. *Unicast : Heterogeneous connection*

If the connection between the sender and the receiver is a composite one, i.e. different segments of the connection have different (temporary or permanent) characteristics, as the one shown in Fig. 42, it is desirable to employ codecs that take advantage of the specific features of the networks. For instance, in Fig. 42, the coder at the entry into the ATM network does layered coding to utilize the support offered by ATM for prioritized traffic. The decoder at the other end is capable of receiving layered video and giving out one layer so the decoder at the receiving end can process it. In the case shown in Fig. 42, the difference is in the networks that participate in the connection. There are other cases where the differences are not so permanent (for instance a credit-based scheme within

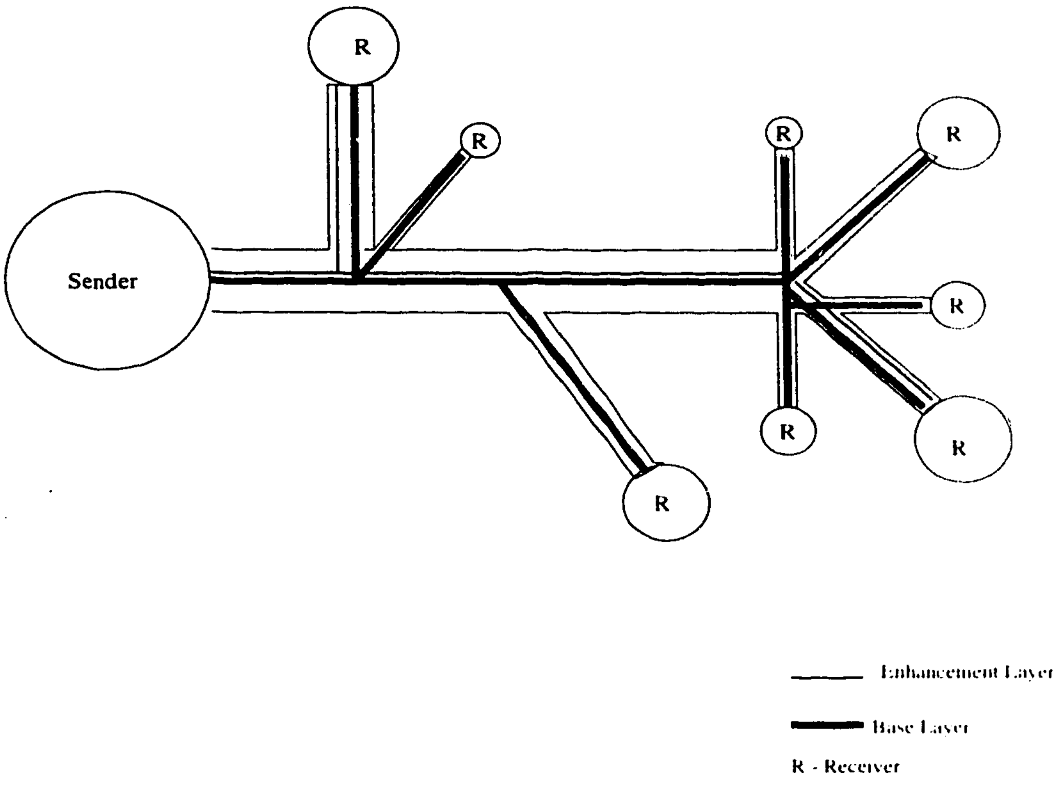


Figure 38: Layered coding

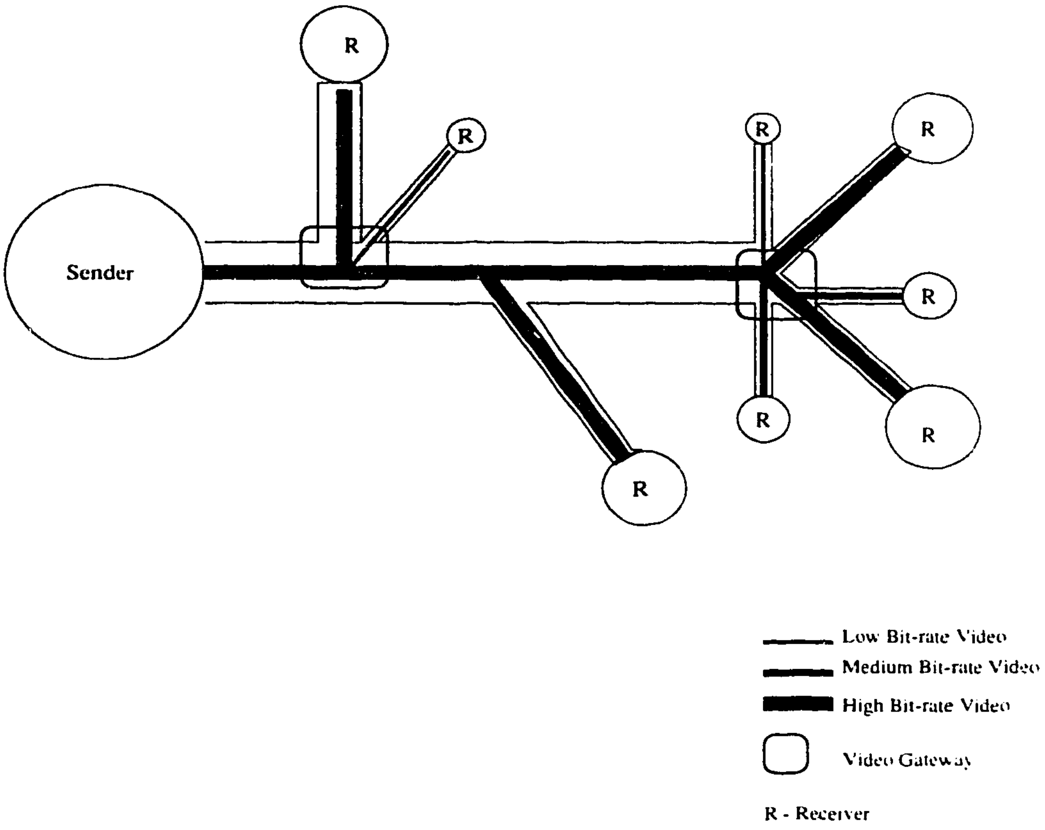


Figure 39: Multicast on Internet using Video Gateways

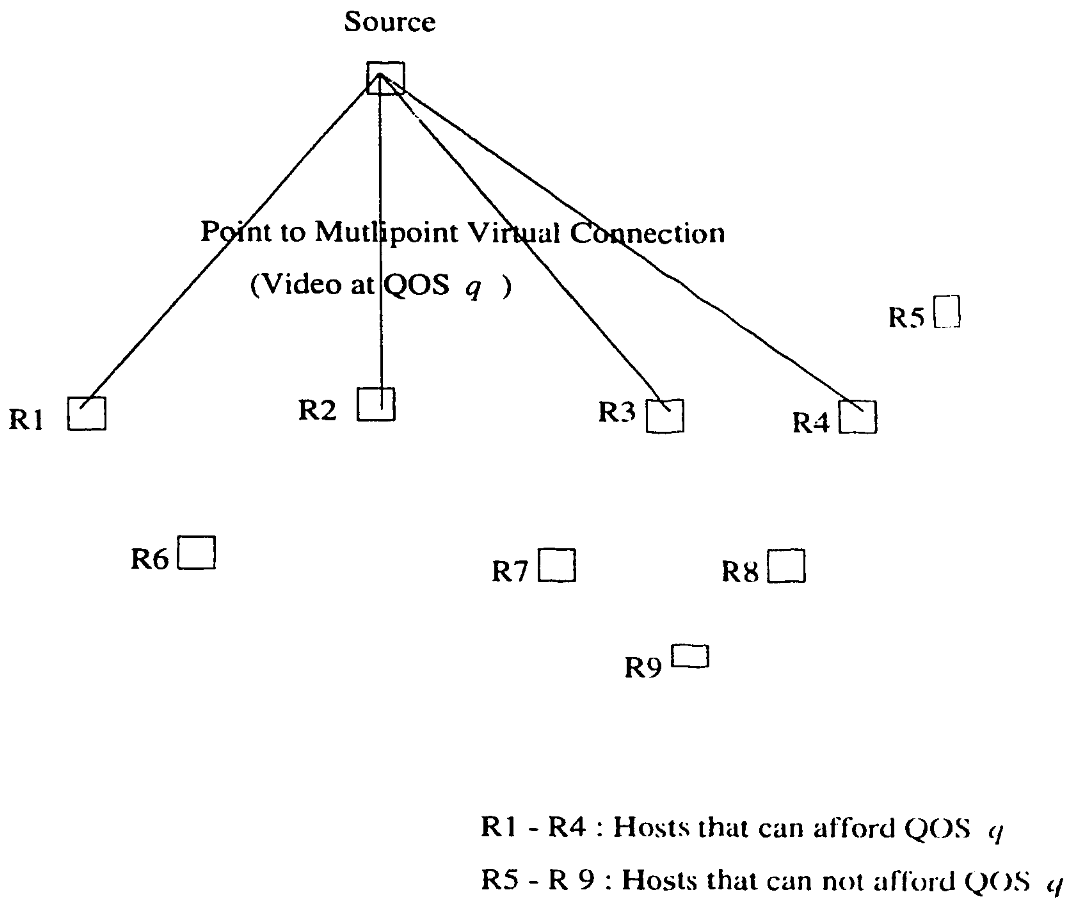
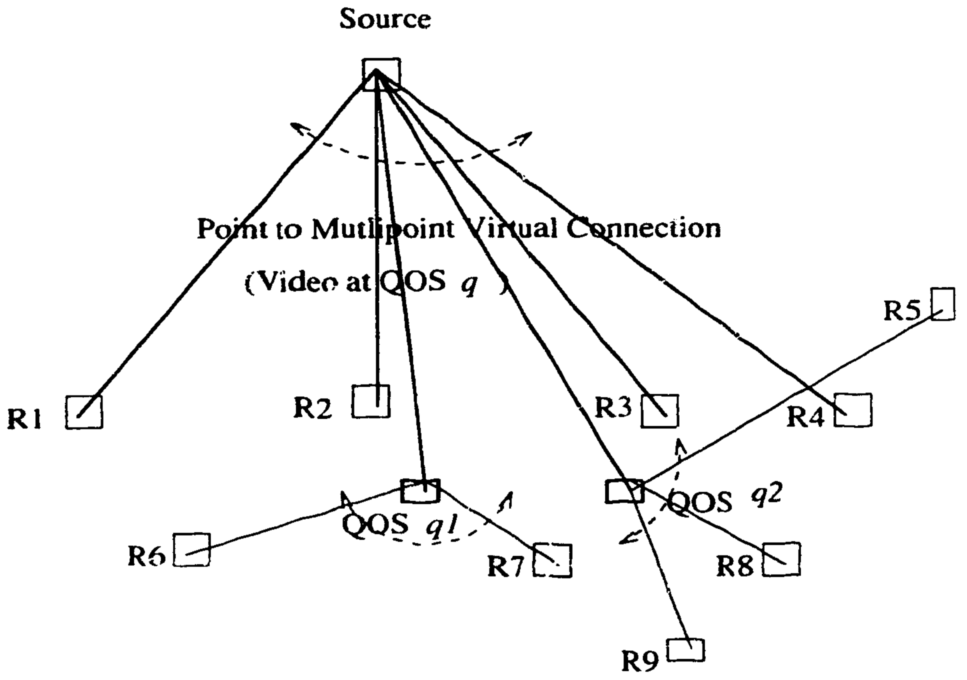

Source
Point to Mutlipoint Virtual Connection
(Video at QOS q)
R1
R2
R3
R4
R5
R6
R7
R8
R9
R1 - R4 : Hosts that can afford QOS q
R5 - R 9 : Hosts that can not afford QOS q

Figure 40: Multicast on ATM



R1 - R4 : Hosts that can afford QOS q

R5 - R 9 : Hosts that can not afford QOS q

Figure 41: Multicast on ATM using Video Gateways

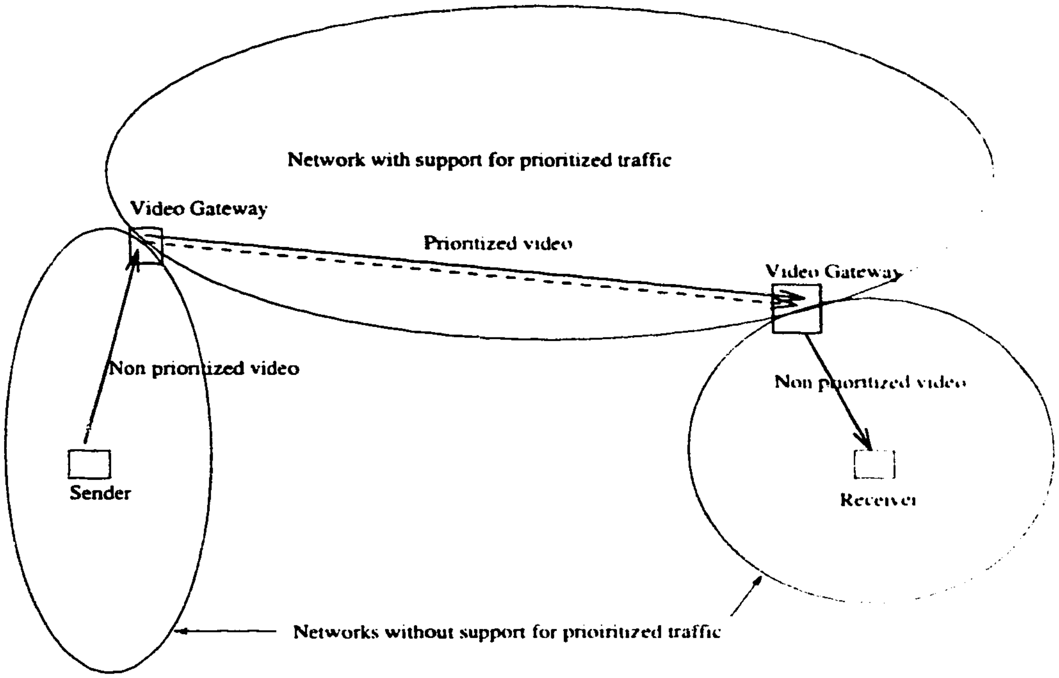


Figure 42: Unicast over heterogeneous connections

an ATM network, where the *dynamic* capacity of different individual links that constitute the connection are taken advantage of), i.e. they last only as long as the connection lasts, and video gateways can be used there as well.

4. *Computing power of the client:* A high-power machine (machine with hardware codecs or powerful CPU) can afford to take highly compressed data, thereby economizing the use of network resources. On the other hand, a lower power machine requires that the video not be highly compressed. Therefore, depending on how powerful the client machine is, video which is not meant exclusively for that client has to be 'gatewayed'.

4.5 Support for Gateways in H.261 and MPEG

- H.261

- The H.261 codec :

  H.261 uses a hybrid of DCT and DPCM. Prediction, when done, is only in the forward direction. Transformed values are quantized and then encoded using a variable length encoder. There are only two types of frames - INTRA and INTER. Only DCT is used in coding INTRA frames while both DCT and DPCM are used to encode INTER frames. Prediction is optional.

- Temporal Subsampling :

  Temporal subsampling in H.261 can be done only at the pel level, according to the way H.261 works. To do this, one can choose to omit a sequence of INTER encoded frames periodically. Although H.261 recommends the use of INTRA encoded frames (1 INTRA for every 132 INTER), the ratio of INTRA encoded frames to INTER encoded frames is very low. Hence there is no impediment in choosing frames to omit.

- Spatial Subsampling :

  This also, can be done only at the pel level.

- Amplitudinal reduction :

  All amplitudinal reduction techniques are applicable. As the frames are mostly INTER, energy threshold is the best way to keep distortion to a minimum without expending too much CPU time.

- MPEG

  - The MPEG codec: MPEG is based on motion compensation and DCT. Frames are predicted (in the forward and the backward directions) by the sender and the receiver, and only the prediction error is transmitted when the actual frame becomes available to the sender. As a result, there are kinds of frames

    1. **I** frames : INTER encoded frames
    2. **P** frames : Predictive encoded frames

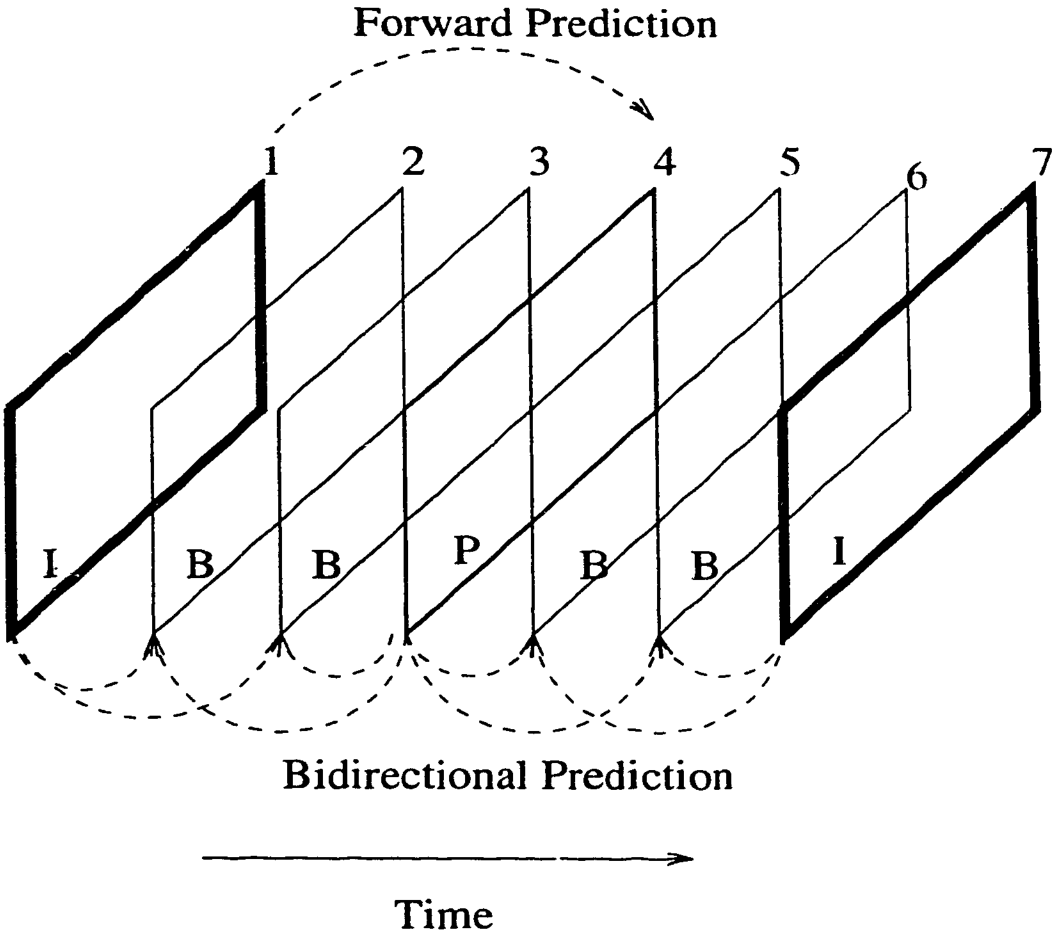


Figure 43: Frames in MPEG Video

3. **B** frames : Bidirectional Predictive encoded frames

- Temporal subsampling in MPEG: The loss of an I frame results in the loss of all subsequent frames between that I frame and the next I frame. Hence I frames are crucial for MPEG decoding. Second in importance are the P frames, as the loss of a P frame can result in the loss of the bunch of B frames that follows that B frame. The loss of a B frame does not affect the decoding of other frames in any way.

  Hence temporal selection in MPEG can only be done in one of the following two ways

  1. Drop B frames only
  2. Drop B frames and P frames

  The former results in very little decrease in the bandwidth used. as B frames take only a small fraction of the total bandwidth as it is (ratio of bandwidths B:P:I = 2:3:5). The reason is that the transmitted portion of these frames is only the error in the prediction. The error is usually minimal as the prediction is bidirectional. However. at low bandwidths. the reduction due to the elimination of B-frames may be considerable. Dropping both B and P frames can result in a considerable reduction in the bandwidth used. but it also reduces prediction correctness as many B frames are predicted based on P frames. In comparison with H.261. MPEG must be said to offer better support for temporal subsampling. This is because dropping frames does not result in additional motion being transmitted in the frames selected for transmission.

- Spatial Subsampling : As pointed out before, MPEG supports a spatial scalability mode where a low resolution video is sent in the base layer. with the help of which the high resolution part can be predicted. But this is only a layering technique. A filter can let just the base layer through. thus making it possible for a client to receive a bit-rate lower than the original. But it is not possible for the client to receive whatever bit rate it wants, because the characteristics of the base layer and the enhancement layer are decided by the sender. Therefore, a gateway is needed to meet the requirements of the client.

- Amplitudinal Subsampling : MPEG offers explicit support to quantization by allowing a different quantizer to be specified for every block. Additional use of quantization to reduce frame rate can be done by treating intra and inter blocks differently. INTRA blocks contain energy at all frequencies and therefore require fine quantization. INTER blocks, on the other hand, carry information primarily in the high frequency region. Quantization can be done coarsely to keep the bits used by the high frequency components within the visually important limits. Besides, data partition can also be employed. It has been shown that for a given bit rate, frequency truncation proves the most efficient. This is because the variable encoding stage used in MPEG encodes frequency truncated blocks more efficiently than it does mimimum distorted or energy thresholded blocks. MPEG recommends an *adaptive quantization* technique, wherein the quantization factor is recomputed for every macroblock, according to the activity of the block against the normalized activity of the frame. The effect of this step is to assign a roughly constant number of bits to each macroblock.

4.6 Other Gateway-related issues

4.6.1 Positioning and control of gateways

Where are gateways to be placed ? And who (client or server or some other program) controls the gateway ?

- *Unicast :* The gateway can be part of the server itself, as there is no point sending high bit-rate video up to a certain point down the connection, only to transmit reduced rate then onwards.

- *Multicast :* For true independence of the server from the concerns of the client, gateways have to be placed at the internal nodes of a multicast tree. One can also think of a different arrangement, where all clients wanting a particular bit-rate join the multicast group on which the gateway offering output video at that rate is itself present. As indicated previously, the gateways could be at a receiving end of a point-to-multipoint connection on which the original video is transmitted, and at the sending end on a different point-to-multipoint connection. Building up a tree structure by combining these star structures to obtain finer control, is also possible. Machines may be predesignated as gateways, so hosts can contact them for service, or gateways can be dynamically created as a new demand for a considerably different bit-rate comes in.

As for control of the gateway, there are two ways possible - one that makes the gatewaying process transparent to the server and the client, and the other that does not. Transparent gatewaying requires the network to be intelligent enough to identify and handle a specific medium (video) in a special way, while, for the same reason freeing the server and the clients of all concerns related to rate-reduction or transcoding. Non-transparent control, on the other hand, while giving the freedom of choice of the position and control of the gateway, also requires that the client and server (or at least one of them) be aware of the presence of the gateway, and take measures to counter the effects due to the gateway (delay introduced, modification of the data).

4.6.2 Effects of Additional propagational delay and Computational complexity

A gateway is a facility that makes it possible for the user to receive video at any desired rate - well. There is a price paid for this facility, however, in the form of the computation at the nodes running the gateway and the delay caused by the gatewaying process itself.

- *CPU intensity :* Hardware implementations are necessary if pel level gateways are to be employed for serious gatewaying. Transport level gateways or hybrid gateways can stand higher input bit-rates than pel level gateways can, even in software implementations, as the most CPU-greedy stage of the codec, viz. forward and reverse transform coding (the MPEG and H.261 conversion between pixels and DCT coefficients), is absent in both. Using hybrid gateways however, requires a different form of coding than the ones produced by the standard codecs.

- *Inter media synchronization problems :* This is a direct result of the delay introduced by the gateway. In an application involving the transfer of audio and video, which are meant to be synchronized during playback, the presence of a gateway can significantly affect the synchronization process. If the client is not aware of the presence of a gateway, it may be led to assume the video packets are lost. This decision may force the playback routine to play the audio alone, which means the video frames when they arrive, will be considered *out of time* and therefore discarded. This will lead to the continued *delay* of the video frames as perceived by the receiver and their consequent "loss".

Chapter 5

CyberSchool

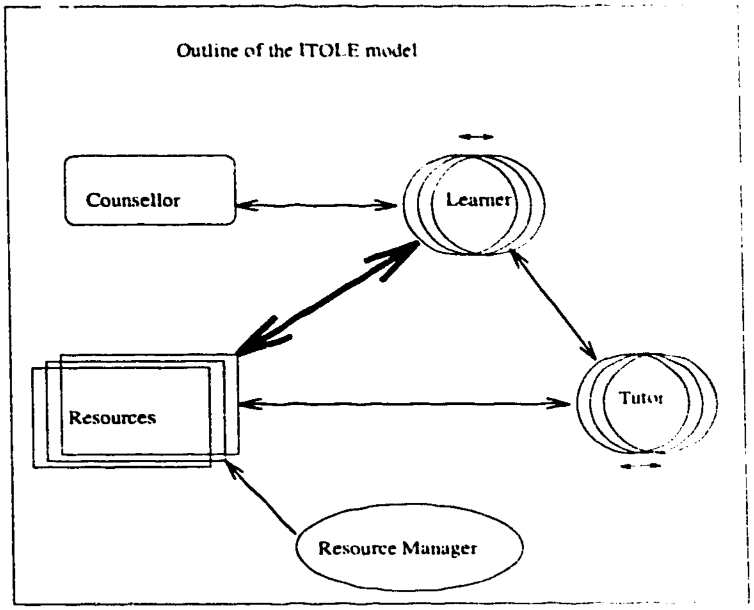


Figure 44: An Information Technology based Open Learning Environment

5.1 Introduction

CyberSchool is an information technology-based open learning environment (ITOLE). Open learning is a student-centered mode of learning. The student decides *when*, *what*, and to a considerable extent even *how* to learn. The learning goals, whether short-term or long-term or particular to a single study session, are self set. Using technology as a buffer between the student and the teacher, facilitates open learning, by eliminating the temporal and geographical dependence of the student on the teacher, which are present in campus-based education. A typical information technology based open learning environment consists of the components shown in Fig. 44. The student (learner), guided by a counsellor and assisted by a tutor, studies primarily by remotely accessing the resources available at an institution.

Individualization of the information delivered is clearly a necessary condition for an open learning environment. The need and way to individualize content are elaborated in Chapter 1. Specific aspects of individualizing the manner of delivery (choice of media, availability of alternate components and controllability of bit rate of video) are covered in Chapter 2 and Chapter 3.

This chapter describes *CyberSchool* as a vision of the way educational information services of tomorrow will work; it also presents a prototype of *CyberSchool*. This prototype is accessible from http://www.cs.ualberta.ca/~kannan/cyberschool. The prototype is meant to illustrate the ideas put forth in the previous chapters.

5.1.1 Motivation

The motivation to create such a technology-based learning environment comes from the following :

Need

The ever-increasing areas of specialization have given rise to a wide range of educational interests. The rapid growth of technology, requires the working community to continue learning. Thus, the area that a particular student wants to study. can no longer be guaranteed to be on the menu of every insititution. Even if it were, the student's interests within the area need not be necessarily the ones the professor is prepared to offer in his series of lectures. There is a real need to bring information technology to the aid of educational institutions - to create a cyber school that can provide information on all areas of interest, specific and general - to allow students to dig deep or move along a particular level of abstraction. as and when they need to do so. Careful organization of information, in particular, ensuring consistency and disallowing redundancy, is a critical requirement if we do not want to be overloaded with information. ACM ([ACM 95]) has identified a kind of learning interest where the learners are interested only in the outcomes of research projects rather than the description of the research process itself. Catering to such requirements necessitates the use of abstraction as much as possible in documentation.

Feasibility

The possibility of distributed multimedia hyperdocuments has been unquestionably established by the World Wide Web. Its open-ended nature has given rise to asynchronous group communication techniques better than those supported by USENET and LISTSERVs, as shown by HyperNews. MIME format electronic mails are another means of asynchronous communication. The MBONE is living proof for synchronous multimedia communication. As active learning only requires time and distance independent availability of the required information to the student, the above applications also suggest the possibility of a provincial, national or even global level educational infrastructure. This realization has led to the emergence of several efforts in online teaching and learning. The University of Alberta is also planning to launch technology based education systems ([UOFA 95]).

5.1.2 Specialty of CyberSchool

The only problem that increases in magnitude, as studentship is spread across different geographical and professional communities is the variety of educational requirements and the levels of resources that people have access to. This is the primary concern in the design of CyberSchool. Therefore the highlights are that

- *Use of dynamic documents:* Course material, questions and clarifications can all be in multimedia form. Text, images, audio and/or video can be used.
- *Scalable delivery:* In CyberSchool, there is consideration for the level of resources the st [illegible] as access to. Both the information delivered, and the manner of delivery are tuned to conform to the specifications of the user, regarding the network, storage and processing capacities available to the user at the time of access.

5.2 CyberSchool : The vision

We quote from [UOFA 95] the vision of the University of Alberta Learning Advantage Program. It is perhaps a less ambitious but sooner-to-be-realized version of the goal of the *CyberSchool* project.

Maria M., a single working parent who is pursuing an M.A. in Criminology, sits down at her home learning station at 9:00 in the evening. She is taking a sociology course which includes a weekly lecture on campus. She has an exam next week, and she wants to review a lecture which she heard, but during which she had some difficulty concentrating because her child was ill. The instructor's lecture notes come up on the screen on demand, she works through them for an hour. Appended to them are some self-tutoring exercises and Internet sources for further investigation, but because she still has one unresolved question, she posts an e-mail query to the instructor; he checks his mail every morning, and she will have a response when she logs on to study tomorrow night.

Maria next retrieves and reviews on screen three government documents she requested this morning from the library. She notes that Phil, a classmate, is logged on from his home in Drayton Valley. (Phil is still driving in every week to attend lectures on campus, but he hopes that by next year Drayton Valley High School will have a videoconferencing room through which it can partner with the U of A for delivery of University courses after regular school hours, as well as bringing in courses and guest lecturers from post-secondary institutions and industry to enrich the high school program, and providing various practical services such as recruitment and interviewing of students by universities, colleges, and business.) Maria and Phil are preparing for a joint presentation in three weeks, and they spend some time on-line discussing how to best use the library material. The project will be in a multimedia format which is particularly suitable to the video interview materials they have collected and are analyzing, and which they can present effectively in their multimedia classroom. At the end of the session, Maria and Phil agree on what each will post to their joint working area for review next Monday.

> Maria's final task for the evening is to book a desktop video conference for tomorrow with her departmental advisor. Her employer has arranged her work schedule so that she can take two courses next term, and she needs some assistance in selecting appropriate ones. By the time she signs on with her advisor, he will have called up on his screen her cumulative academic record and his notes about her plans for completing her degree. When they have agreed on course selection, he will be able to register her with a keystroke. She will, of course, be able pay for her courses and extend her parking permit on-line.

In CyberSchool, the student seldom knews which institution the teaching unit he is reading at the moment is coming from. Faculty from various institutions cooperate in creating and maintaining subject material, each institution being responsible for those topics for which its faculty are well known. This means not only that the students access the best material (and thereby be able to ask questions to the experts) but also that sharing the information leads to consistency and efficient use of space (storage) and time (The most desirable way of tackling information overload is to avoid redundant work). In the rest of this section we envision what CyberSchool means to the educational information provider (institution), educational information user (student) and the connection provider.

5.2.1 Information User's view

A service like CyberSchool will have to be one of the services provided by the student's communication network provider (similar to the way 1-900 services are provided by telephone companies today). Where the material itself comes from is not the student's concern. The student, however is entitled to ask questions (express doubts) to the authors of the material. The cost associated will perhaps be different for synchronous and asynchronous communication. CyberSchool serves as a learning support environment if the student is undergoing campus-based education. The working community may use CyberSchool for professional development (learning about the developments in areas of interest to them and to cope with the fast technological advancements just to be on the move in a highly competitive environment).

CyberSchool supports an exploration-based learning environment. Regular students may explore based on curiosity, while working people may use a need driven approach. It supports both by making short presentations (answering to the point), and providing additional information only on demand. Information is available on line, so the student can access it whenever and any number of times he needs it. The amount of detail available is practically infinite. Students can also develop documents by building on the already online documents, without having to repeat things for the sake of completeness. Online automatic objective type testing mechanisms and means of communicating with other students and with the teacher in both synchronous (time-dependent) and asynchronous (time-independent) fashions are also available.

5.2.2 Information provider's view

Teaching material is prepared by a team of teachers from several institutions. The material can use multimedia and can be structured using hypermedia. Dynamic media recorded for best performance in the case of good playback conditions can be delivered through much lower levels of technology as well. Preparation of the documents and most of answering students' questions are done without any realtime constraints, so both delivery of the basic material and clarification of doubts can be done slowly and steadily. The information provider pays the connection provider based on connection time, distance and QOS class. The information provider is paid by the user for each teaching unit the user accesses. Whether subsequent accesses of the same teaching units will be charged for, and whether the student is allowed to store the units locally, are policy decisions to be made by the information provider.

5.2.3 Connection Provider's view

Both the information provider and the user pay the connection provider for the connection time, distance and QOS. The responsibility of the connection provider is to provide quick, reliable connections and a variety of QOS classes.

5.3 CyberSchool : The prototype

A basic platform that can deliver a single document containing text, images, audio and video on demand, has been constructed by integrating a continuous media (audio and video) server with a world wide web server (which already carries text and images). The features of CyberSchool are built above this platform.

5.3.1 Features

- *Authoring and accessing* CyberSchool is only the infrastructure for carrying educational information. The actual study material is to be placed in it by teachers.

  Both students and teachers are users of CyberSchool, which can be seen as a (computer-mediated) communication tool, usable for educational purposes.

  - **Authoring**:

    Authoring course material boils down to authoring individual teaching units, as the course material is nothing but a hyper-structured collection of the teaching units.

    In CyberSchool, the author is not required to be on the machine that holds the secondary storage for archiving the material to be authored. In other words, *authoring can be done remotely.*

    Each teaching unit can contain text, images, audio and video. The audio and video do not have to be produced simultaneously. Information on creating teaching units is available in the next section. Hyperlinks leading to other teaching units can be embedded in the text and/or image part of each teaching unit.

- **Accessing**:

  Accessing also can be done remotely (this is, in fact, an essential part of the idea of ***access-on-demand***). Each teaching unit is identified by a URL and the student has to invoke this URL through his client program. Facilities for remotely controlling the session presentation viz. **REWIND**, **FORWARD**, **PLAY**, **PAUSE** and **STOP** are available. Refer to the documentation for more information.

Note that CyberSchool imposes no restriction on who should author or access documents.

That part of CyberSchool which allows remote authoring and on-demand access of a single multimedia document can be considered to be the kernel of CyberSchool (or its bottom layer). The application level features of CyberSchool, viz. creation of lectures (sequencing of teaching units) and interaction are built around this kernel (or, in other words, above this layer).

- *Authoring and accessing course material:*

  Teachers can create *lectures*, which provide an optional navigational guidance to the student during information-cruising. A *lecture* is a specific sequence of individual teaching units. It can also be thought of as a guided tour.

  Accessing the total course material starts by accessing the hub document (the default start node for a given topic - the one recommended by the institution providing the information). Of course the student may also directly invoke a unit located anywhere in tree of information (as opposed to the root, which is where the hub document is located), if he knows the identifier to the unit he wants. Once the student starts with a unit, then onwards accessing is either guided by the *lecture* feature which, in effect, points to the next unit to be studied, or done by the student in an exploratory fashion by invoking the hyperlinks embedded in the body of the units themselves. These links are placed in the unit by the author during creation.

- *Interaction:*

  The availability of hyperlinks is meant to minimize the need for human-human interaction by maximizing the ease and effectiveness of human-content interaction. Most of the questions (eg. What is X? How is Y done?) can be answered by the content itself, if it is carefully organized. Nevertheless, it is very likely that the student still has questions or needs clarifications. Besides, group interaction between students may also be necessary (it can serve various purposes, like sharing views, doing team-projects, etc.). To facilitate interaction in these situations, CyberSchool allows a group communication facility. The discussion can be structured in the form of a tree. Students with access to multimedia equipments can use audio and video, in asking their questions (or in responding to someone else's).

- *Testing:*

  As open-learning is by definition an active form of learning, the availability of means to do simple self-tests will help the cause of open-learning. CyberSchool provides way to do this. The tests are however only of objective-type, as the machine is required to evaluate the answers and return the results to the student immediately.

5.3.2 User Manual

The following form the client of the CyberSchool prototype.

- Mosaic
- nv
- vat

And the following form the CyberSchool server :

- a web server with the common gateway interface feature
- nv_play, nv_record
- vat_play, vat_record
- some *cgi* programs

The *_record utilities will be active during authoring, and the *_play, during access. The cgi programs coordinate the behavior of the web server and the continuous media server (i.e. the audio and video record/replay utilities) thus providing a higher level of use, viz. creation and access of files containing multimedia data.

Authoring

- *Creating a teaching unit*:

  1. Create the html component of the teaching unit using any html (or text) editor. It has to be a regular html file with the difference that you can include hyperlinks to other teaching units in the following format.

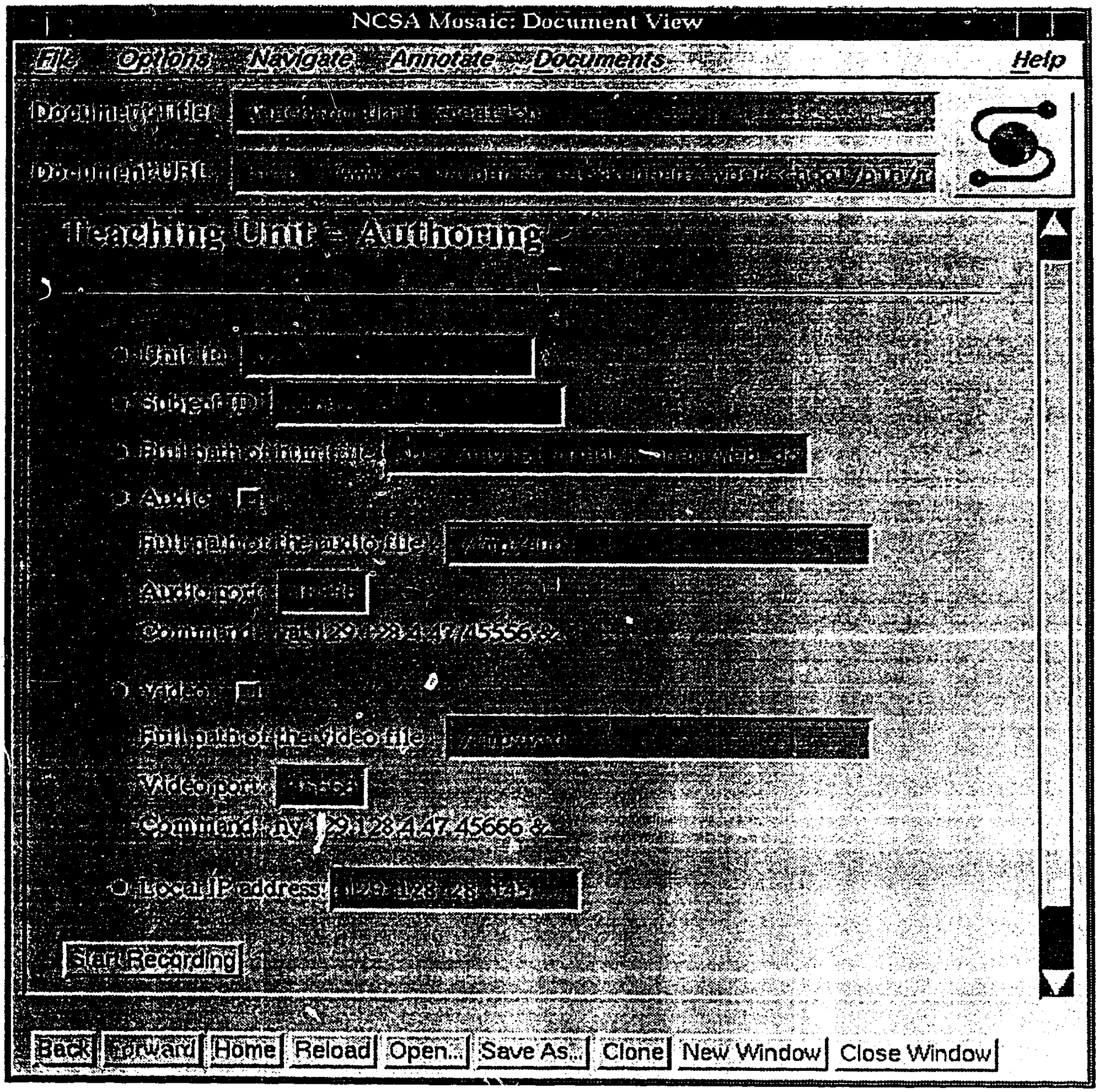
     ```
     <A HREF=''CYBERSCHOOL\_GET\_UNIT?unit_id=<unit-file-name>&
     sessionlist=<subject-name''> .. </A>
     ```

     Both *unit-file-name* and *subject-name* must be in alphanumeric characters. Now make this html document world-readable.

  2. Start Mosaic

  3. Let CYBERSCHOOL_HOME stand for the URL pointing to the home directory of CyberSchool (eg. http://www.cs.ualberta.ca/~kannan/cyberschool). Access the URL CYBERSCHOOL_HOME/bin/mrform.cgi through Mosaic. The page returned by mrform.cgi is shown in Fig. 45.

  4. Fill up the details in the form now displayed by Mosaic. The details are



Teaching Unit - Authoring

○ Unit ID: [illegible]

○ Subject ID: [illegible]

○ Full path of html file: [illegible]

○ Audio □

Full path of the audio file: [illegible]

Audio port: [illegible]

Command: vat 129.128.4.47/45556 &

○ Video □

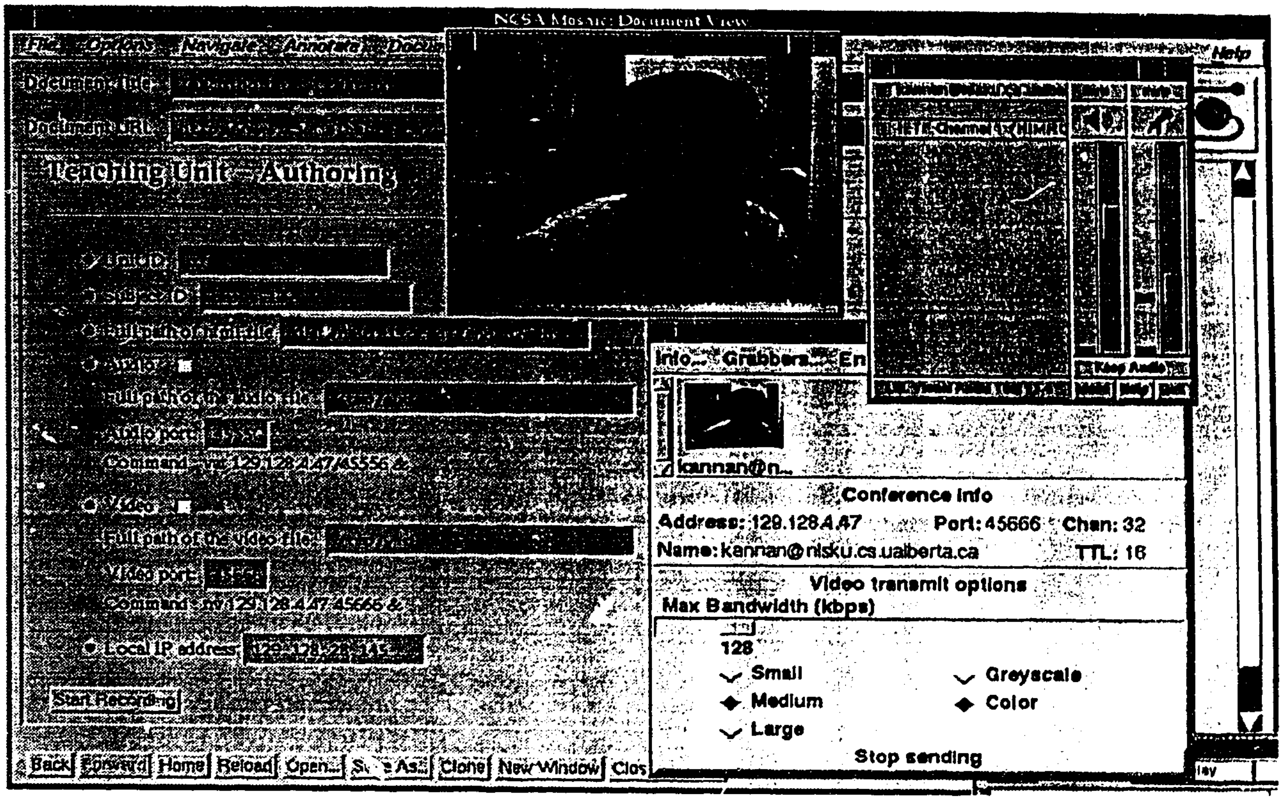
Full path of the video file: [illegible]

Video port: [illegible]

Command: nv 129.128.4.47 45666 &

○ Local IP address: 129.128.28.145

Figure 45: Creating a teaching unit : Stage 1



NCSA Mosaic: Document View

Document Title:

Document URL:

Teaching Unit - Authoring

○ Unit ID:

○ Subject ID:

○ Full path of html file:

○ Audio ■

Full path of the audio file:

Audio port: 45556

Command: vat 129.128.4.47/45556 &

○ Video ■

Full path of the video file:

Video port: 45666

Command: nv 129.128.4.47 45666 &

○ Local IP address: 129.128.28.145

Start Recording

Conference Info

Address: 129.128.4.47 Port: 45666 Chan: 32

Name: kannan@nisku.cs.ualberta.ca TTL: 16

Video transmit options

Max Bandwidth (kbps)

128

✓ Small	✓ Greyscale
◆ Medium	◆ Color
✓ Large	

Stop sending

Figure 46: Creating a teaching unit : Stage 2

- full path of the html file (created in step 1)
- full path of the audio file (to be created now)
- full path of the video file (to be created now)
- name of the unit file (alphanumeric characters only)
- name of the subject (alphanumeric characters only)
- IP address of the local machine (more precisely, of the machine from which the audio and video are to be sent for recording)

5. Start *nv* and/or *vat* on the local machine with the parameters shown in the above-mentioned form. (The complete commands themselves, as they have to be typed on your command line, are shown. Be sure to reflect any modification you make to the port numbers, in those commands) Start the transmission of video and audio from *nv* and *vat* respectively. At this point you must have a screen setup similar to the one in Fig. 46.
6. Select **Start Recording**; Immediately after that, start the audio and video sourcing (i.e., start speaking and acting). The recording proceeds only for one minute, without break. The selection of the **Start Recording** link does not make the browser display a new page (So do not wait for a new page before starting action).

Note that the directories in which the audio, video files and the unit file are to be placed, have to be *world-writable*.

- *Creating a lecture*:

  The cyberschool home directory contains two subdirectories **bin** and **docs**. The former contains the executables and the latter the course material. The teaching units are grouped in subdirectories of the **docs** directory, named after their respective subjects. Besides the units, each subject directory can also contain a file **lecture.list**, which looks like this :

```
lecture1: u1 u2 u3 u4
```

```
lecture2: u1 u4
lecture3: u3 u2
~
~
''lecture.list'' 3 lines, 54 characters
```

This feature is to enable the author of the units to provide a default sequence for the presentation of the units. The names after the ':' symbol in each line give that sequence. Note that this is only the default sequence and the actual sequence is decided at *study-time* by the student's invoking of the hyperlinks. The effect of the above sequencing procedure, that is apparent to the student, is a **NEXT** button that appears at the end of the text in each teaching unit.

Within each subject, the author can have a **hub** unit. This unit is the default starting point for exploration into that subject. It is specified in a file called *hub*. If, in accessing CyberSchool, no particular unit is specified (but a subject is), then hub unit is the one to be delivered.

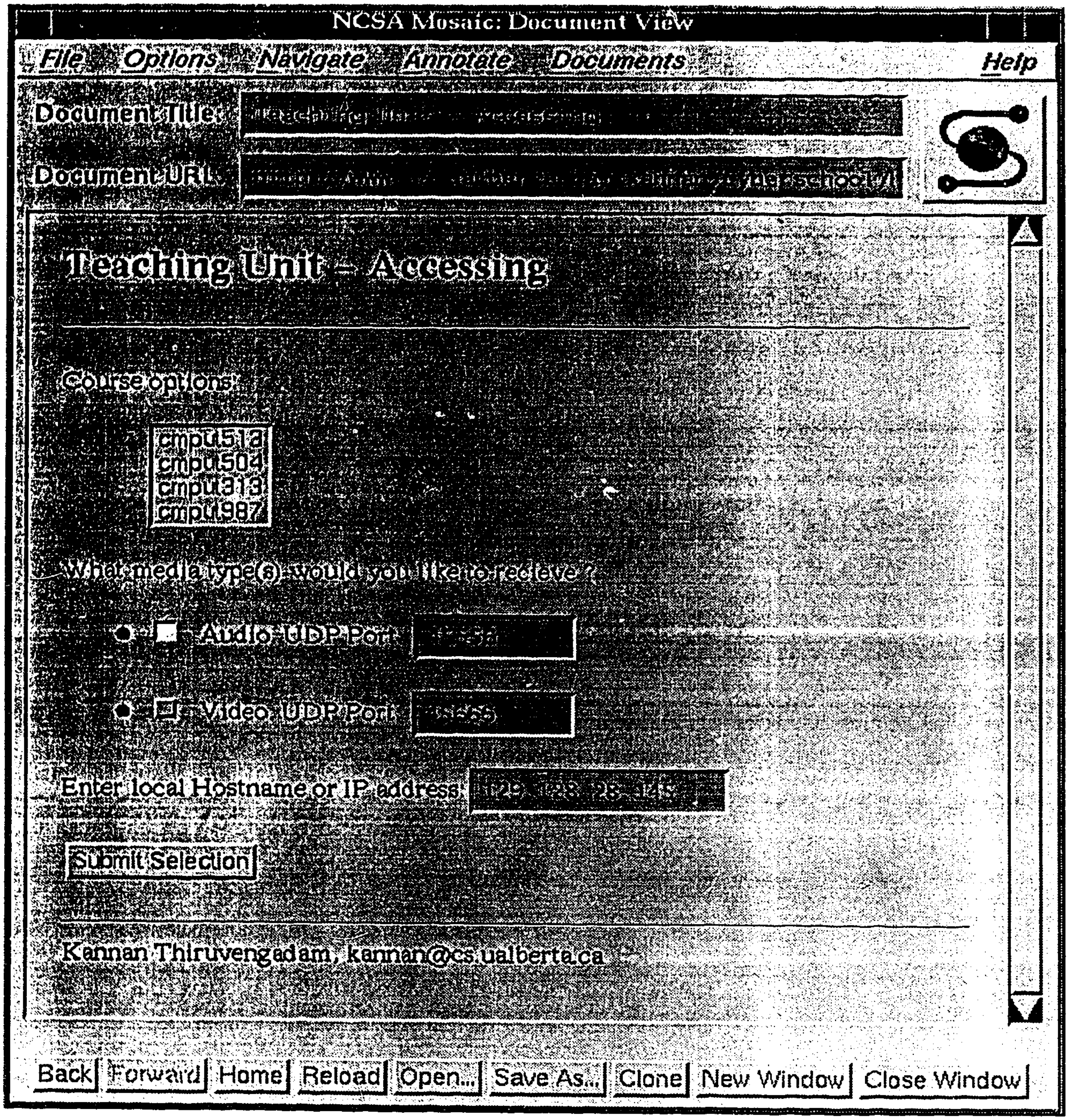
```
unit5
~
~
~
~
''hub'' 1 line, 3 characters
```

Accessing

- *Accessing a teaching unit:*

  In order to access a teaching unit from CyberSchool, you need the three utilities that make up the CyberSchool client, viz. *Mosaic*, *nv* and *vat*.

  The steps to follow, then, are the following :



Document Title: [illegible]

Document URL: [illegible] school/

Teaching Unit – Accessing

Course options:

cmput513
cmput504
cmput313
cmput987

What media type(s) would you like to recieve ?

● ■ Audio UDP Port [illegible]

● ☐ Video UDP Port [illegible]

Enter local Hostname or IP address 129.128.28.145

Submit Selection

Kannan Thiruvengadam, kannan@cs.ualberta.ca

Figure 47: User options for media selection

1. Start Mosaic
2. Let CYBERSCHOOL_HOME stand for the URL pointing to the home directory of CyberSchool; access the URL CYBERSCHOOL_HOME/bin /mediaform.cgi. The form now displayed is shown in Fig. 47
3. Select the subject.
4. Select the media you want to receive (from among audio and video).
5. Submit the form. The result of this step is shown in Fig. 48
6. Start nv and/or vat with the addresses shown in the page displayed by the web browser.
7. Select **Display Unit**. The result is as shown in Fig. 50. Fig. 49 shows another teaching unit.

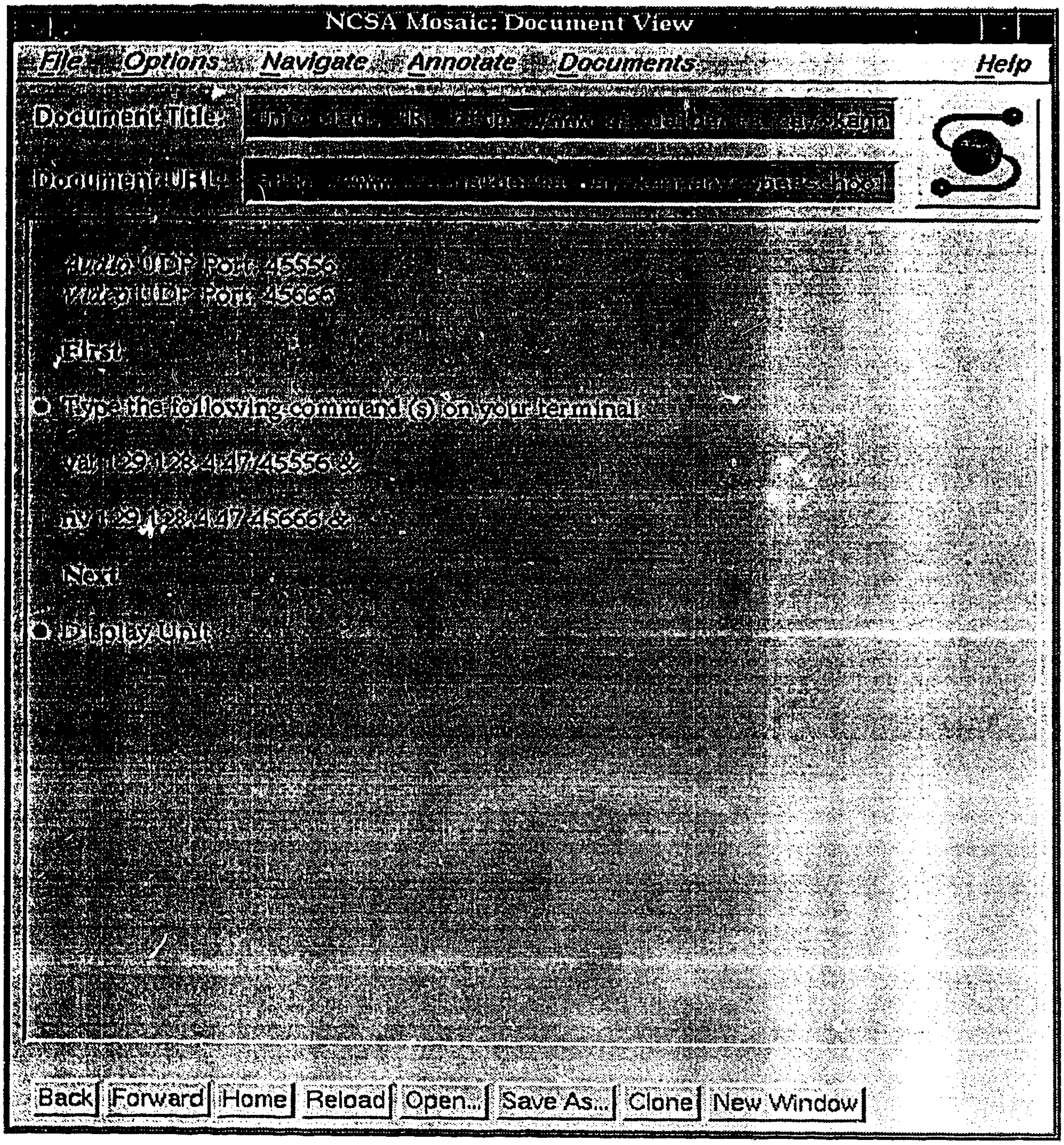
In the teaching unit shown, text, images, audio client and video client can be seen. Means to control the presentation of the audio and video are also shown. The icons at the bottom left are icons that lead to

- a bulletin board system (the ? symbol) for asynchronous group interaction
- a personal email system, for personal communication (questions to the author, or comments or suggestions)
- a search facility to locate teaching units of interest

Selecting any anchor (part of text or image that forms one end of a hyperlink) in the body of a teaching unit leads to a teaching unit on the topic identified by the anchor.

Once *nv* and *vat* are started with the appropriate addresses, the student can *directly* access any particular URL, if he knows the following

- the unit id, i.e. the file name of the unit
- the subject id, i.e. the subdirectory in which the unit is



NCSA Mosaic: Document View
File Options Navigate Annotate Documents Help
Document Title: [illegible]
Document URL: [illegible]
Back
Forward
Home
Reload
Open...
Save As...
Clone
New Window

Audio UDP Port: 45556
Video UDP Port: 45666

First:

○ Type the following command (s) on your terminal

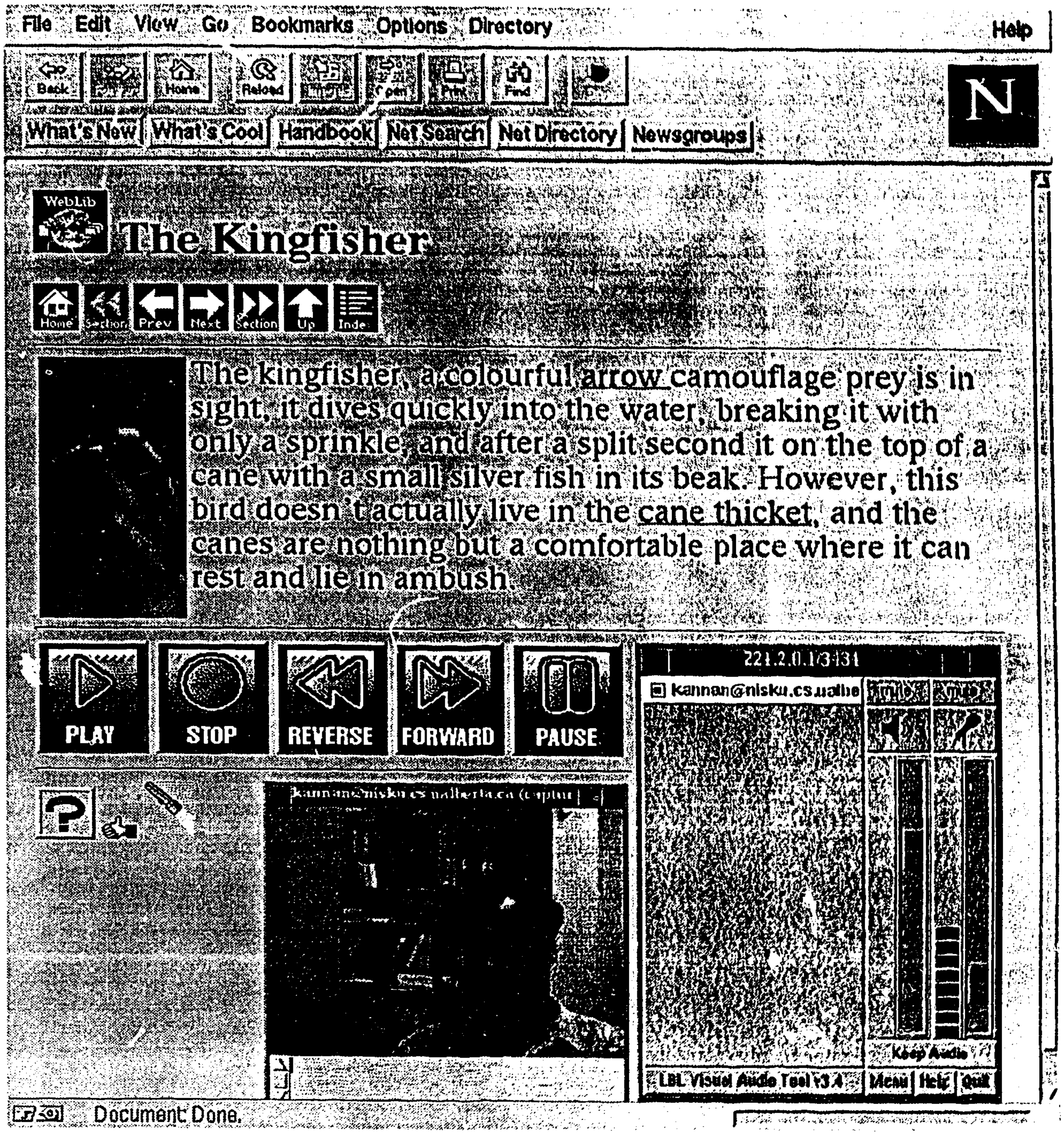
```
vat 129.128.4.47/45556 &

nv 129.128.4.47 45666 &
```

Next:

○ Display Unit

Figure 48: Result of submitting the options

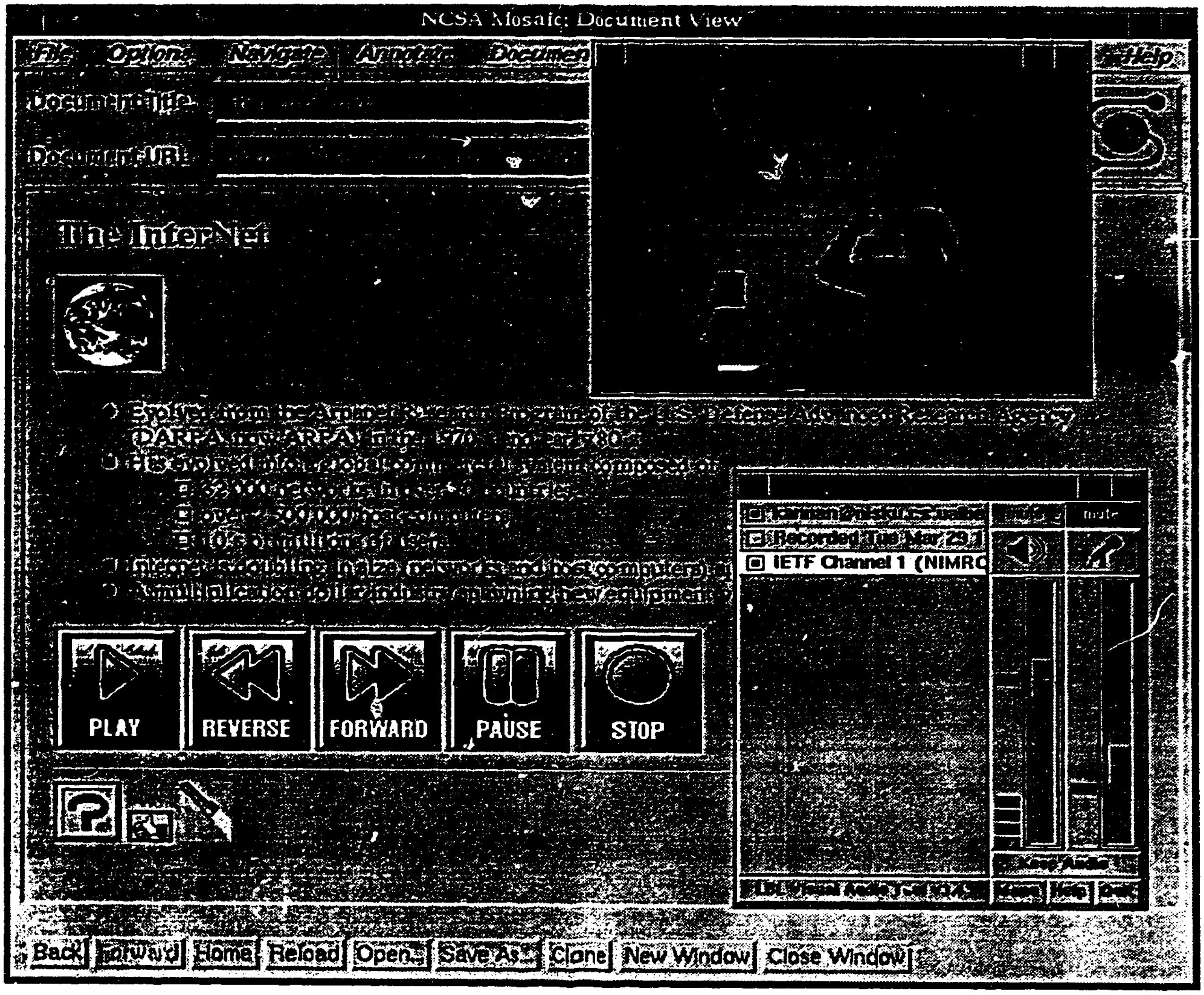


The Kingfisher

The kingfisher, a colourful arrow camouflage prey is in sight, it dives quickly into the water, breaking it with only a sprinkle, and after a split second it on the top of a cane with a small silver fish in its beak. However, this bird doesn't actually live in the cane thicket, and the canes are nothing but a comfortable place where it can rest and lie in ambush.

Document: Done.

Figure 49: A teaching unit



The InterNet

- Evolved from the Arpanet research program of the U.S. Defense Advanced Research Agency (DARPA, now ARPA) in the 1970s and early 1980s
- It has evolved into a global communication system composed of:
  - 32,000 networks in over 60 countries
  - over 2,500,000 host computers
  - 10's of millions of users
- Internet is doubling in size (networks and host computers) i
- A multi-billion dollar industry spawning new equipment

Figure 50: A teaching unit

The URL to access the required unit is of the following format.

```
CYBERSCHOOL_HOME/bin/unit.cgi?unit_id=<unit-file-name>&
&sessionlist=<subject-id>&voiceb=Audio&
videob=Video&ip_addr=<recvr-ip-addr>&auport=<recvr-audio-port>&
vidport=<recvr-video-port>
```

If *voice* is not required, *voiceb=Audio* should be excluded.If *video* is not required, *videob=Video* should be excluded.

- *Accessing a lecture* The student can directly access any unit, alongside using the lecture feature. To do this, the above URL has to be used along with another attribute that identifies the lecture wanted. The format of the URL is shown below :

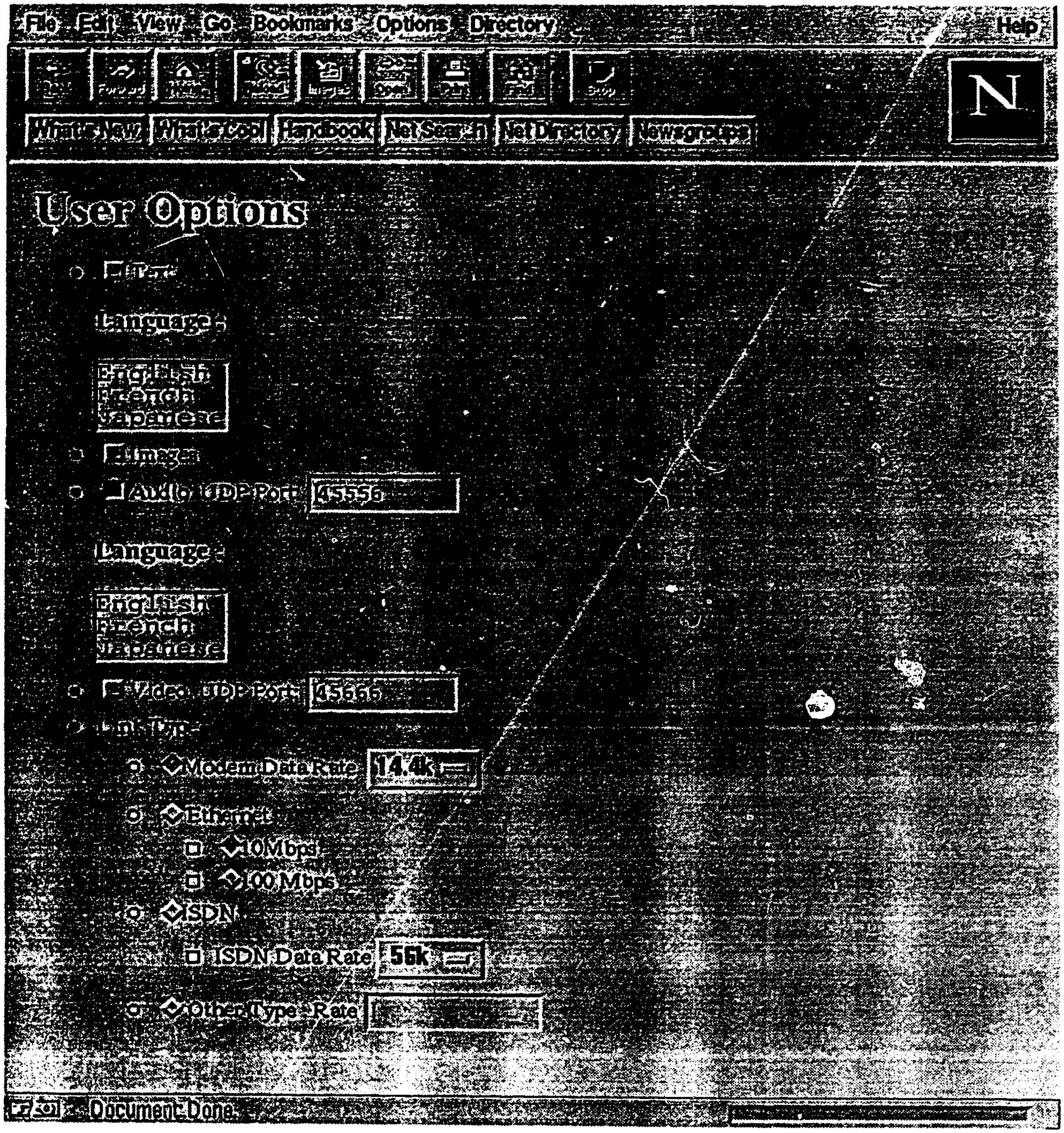
```
CYBERSCHOOL_HOME/bin/unit.cgi?unit_id=<unit-file-name>&lecture_id=1
&sessionlist=<subject-id>&voiceb=Audio&
videob=Video&ip_addr=<recvr-ip-addr>&auport=<recvr-audio-port>&
vidport=<recvr-video-port>
```

5.3.3 Implementation Status

According to the current status of the implementation, no video gateway is integrated with the prototype. If it were, then the initial specifications of the user will be more than what was shown previously. Fig. 51 captures some of the input variables that will be necessary in that case. Other variables such as greyscale vs. color picture and the size of the picture will also get included once the gateway is integrated with the rest of the system.

5.3.4 Organizing the course material

We have attempted to put together a few teaching units pertaining to *Operating Systems* trying to adhere to the hyperdocument development techniques discussed in



User Options

- ○ □ Text

  Language :

  English
  French
  Japanese

- ○ □ Images
- ○ □ Audio UDP Port: 45556

  Language :

  English
  French
  Japanese

- ○ □ Video UDP Port: 45666
- ○ Link Type
  - ◇ Modem Data Rate 14.4k
  - ◇ Ethernet
    - ◇ 10Mbps
    - ◇ 100 Mbps
  - ◇ ISDN
    - □ ISDN Data Rate 56k
  - ◇ Other Type Rate

Figure 51: User options for media selection and rate control

Chapter 1. Some of the nodes are shown here. There is a short discussion on them, and an explanation of when and why deviations from the object-oriented development may be necessary, and wha* compromises should be made in case the units developed seem too *formal* to be effective.

Discussion

The above implementation can be accessed frcm http://www.cs.ualberta.ca/~kannan/cyberschool/bin/unit.cgi?sessionlist=cmput379.

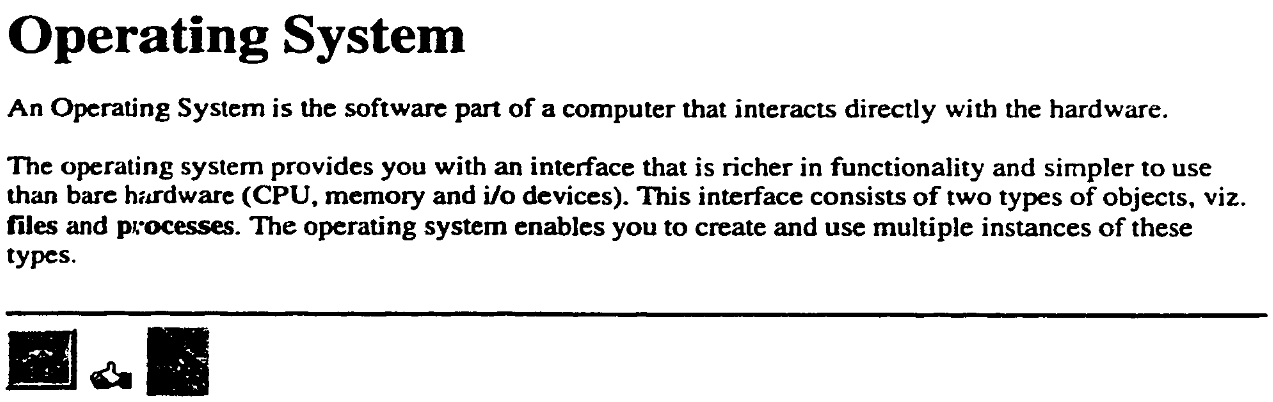
In the figures 52 through 57, bold face words indicate hyperlinks. In this implementation, links other than those mentioned in Chapter 1 (Inheritance, Encapsulation and Instance) have been used (Fig. 58). However, these additional links were used only to *divide and conquer* the problem at hand, not to physically break the document into small pieces. Therefore, these links can be considered close to *Encapsulation* type links.

There are other cases as well, where strict adherence to object oriented linking may be difficult to implement.

- In a review hyperdocument where references to parts of the reviewed document are required
- In developing tours where the author chooses to visit a sequence of teaching units for some specific purpose
- In drawing inferences, for instance infering something from the information available in two teaching units

Many times, creating a semantic network showing how every concept relates to every other concept (if there is any relation) may be necessary. Additional links made for such purposes should include

- those that refer to similar concepts (For instance the need for context-switching due to avoid starvation, the need for schemes like paging and segmentation to

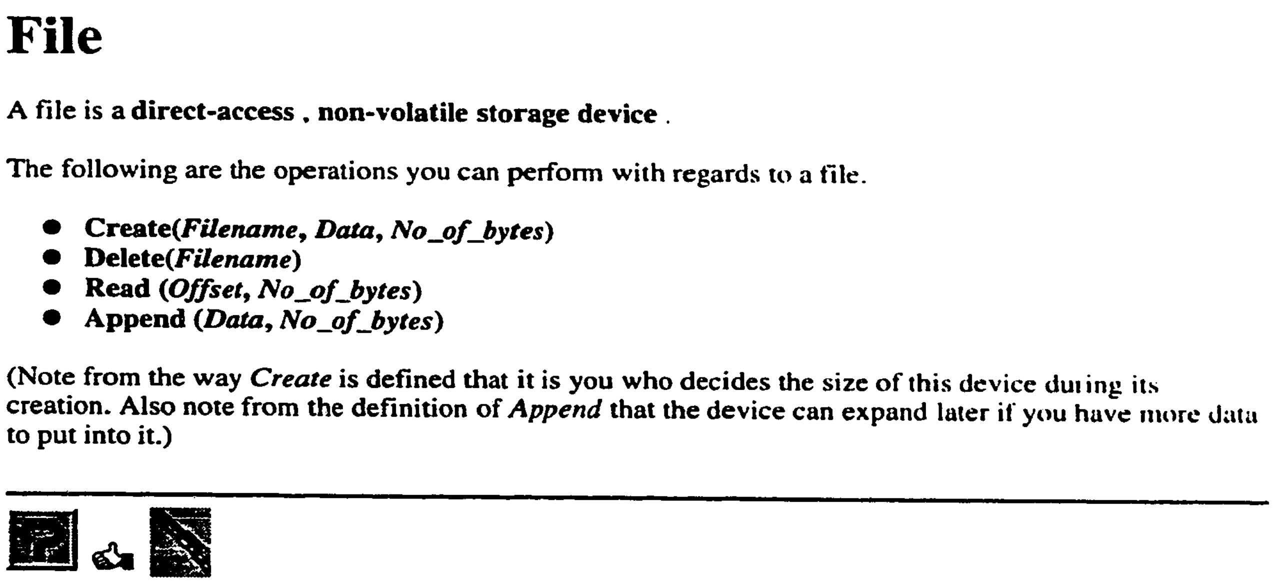


Operating System

An Operating System is the software part of a computer that interacts directly with the hardware.

The operating system provides you with an interface that is richer in functionality and simpler to use than bare hardware (CPU, memory and i/o devices). This interface consists of two types of objects, viz. **files** and **processes**. The operating system enables you to create and use multiple instances of these types.

Figure 52: TU1 : Operating System



File

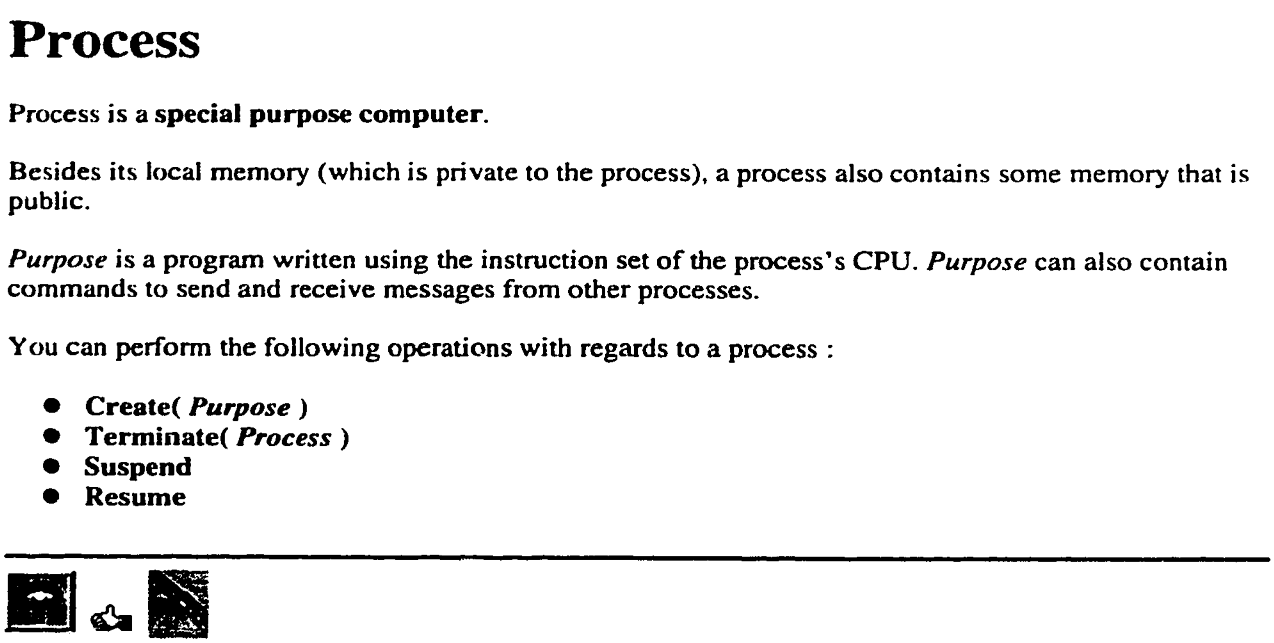
A file is a **direct-access , non-volatile storage device** .

The following are the operations you can perform with regards to a file.

- **Create(*Filename*, *Data*, *No_of_bytes*)**
- **Delete(*Filename*)**
- **Read (*Offset*, *No_of_bytes*)**
- **Append (*Data*, *No_of_bytes*)**

(Note from the way *Create* is defined that it is you who decides the size of this device during its creation. Also note from the definition of *Append* that the device can expand later if you have more data to put into it.)

Figure 53: TU11: File



Process

Process is a **special purpose computer**.

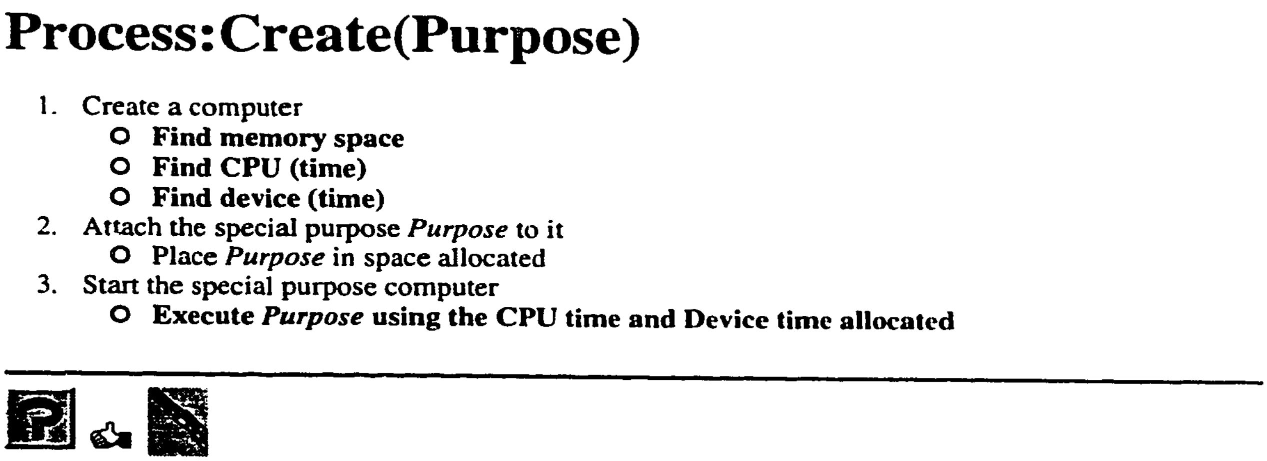
Besides its local memory (which is private to the process), a process also contains some memory that is public.

Purpose is a program written using the instruction set of the process's CPU. *Purpose* can also contain commands to send and receive messages from other processes.

You can perform the following operations with regards to a process :

- **Create(** ***Purpose*** **)**
- **Terminate(** ***Process*** **)**
- **Suspend**
- **Resume**

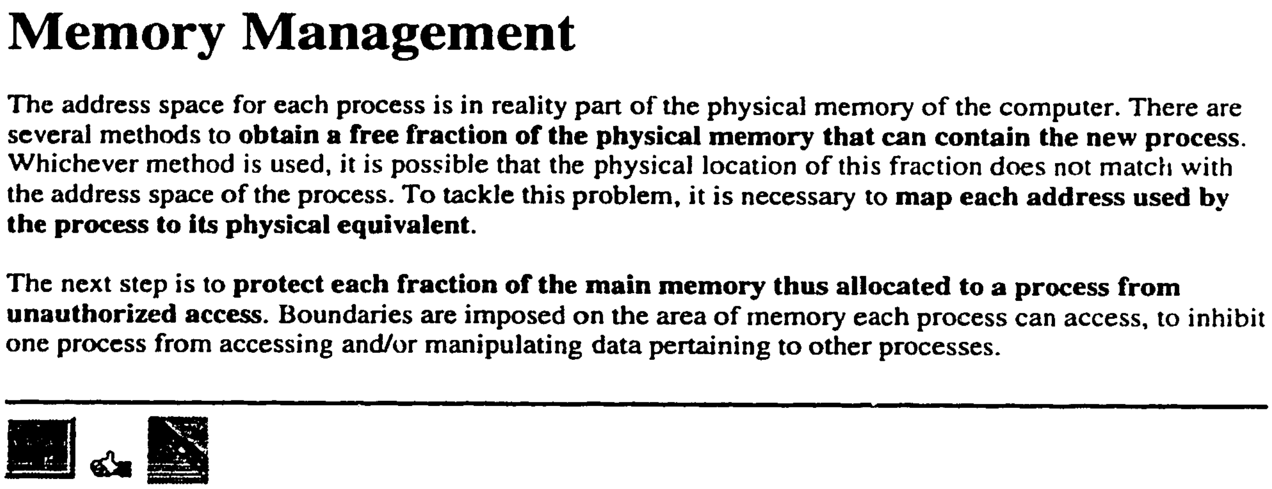
Figure 54: TU12: Process



Process:Create(Purpose)

1. Create a computer
    - **Find memory space**
    - **Find CPU (time)**
    - **Find device (time)**
2. Attach the special purpose *Purpose* to it
    - Place *Purpose* in space allocated
3. Start the special purpose computer
    - **Execute *Purpose* using the CPU time and Device time allocated**

Figure 55: TU121: Construction of a process



Memory Management

The address space for each process is in reality part of the physical memory of the computer. There are several methods to **obtain a free fraction of the physical memory that can contain the new process.** Whichever method is used, it is possible that the physical location of this fraction does not match with the address space of the process. To tackle this problem, it is necessary to **map each address used by the process to its physical equivalent.**

The next step is to **protect each fraction of the main memory thus allocated to a process from unauthorized access.** Boundaries are imposed on the area of memory each process can access, to inhibit one process from accessing and/or manipulating data pertaining to other processes.

Figure 56: TU1211: Obtaining memory for the process

CPU Management

A virtual CPU is created for each process using a fraction of the time of the real CPU. In sharing its time between the processes, if the CPU is to execute one process at a stretch, then the other processes waiting for the CPU will starve (and the users will notice). To aviod this, the CPU spends only a small interval of time on each process, at the end of which it switches over to another waiting process. Because of this discontinuity of the CPU time available to each process, some way of reminding the CPU of the status of each process becomes necessary when the process is resumed.

The combined use of time-sharing and shared-memory causes a problem. Solving this problem requires that concurrent processes be mutually excluded from entering criticial sections of the code that access the same datum. There are a few ways to ensure mutual exlusion.

Figure 57: TU1212: Obtaining CPU for the process

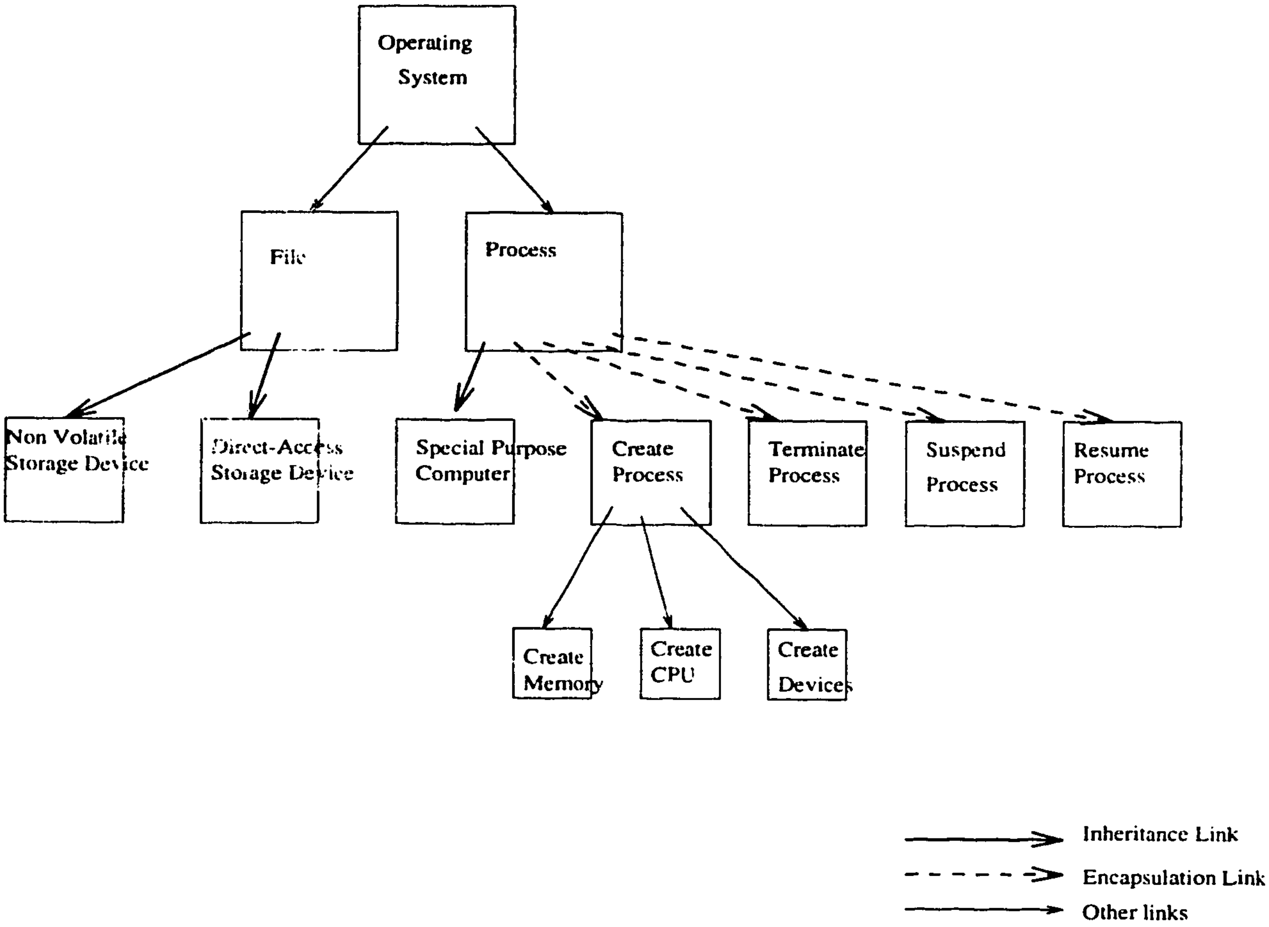


Figure 58: Instance of Hyperstructure Implementation

avoid requiring contiguous memory area are versions of the same problem, the former in the temporal domain and the latter in the spatial domain)

- those that refer to concepts that are complimentary
- those that refer to concepts that resolve apparent inconsistencies in the current node

However, one must try to arrive at a judicious combination between free-format semantic networks and object-oriented information trees, if for some reason the former seems necessary for the application at hand.

5.3.5 How the prototype works

Fig. 59 shows how the prototype works. It is made by combining audio and video utilities with the world wide web. The responsibility of the web is to carry the text and image parts of each teaching unit, and to serve as the user interface. The audio and video server utilities are connected to the web, through the Common Gateway Interface of the latter. At the receiving ends, the audio and video utilities, and the web client run independently, except for the fact that the web client is acting as the user-interface thus controlling the session presentation of the audio and video as well (PLAY, STOP etc.). Common Gateway Interface feature of the web is a way to make the user execute a program on the web server by accessing the URL of the program. In the case of CyberSchool, CGI programs do the following functions :

- invoke the audio and/or video utilities
- pass the session presentation control messages (PLAY, STOP, REW, FWD and PAUSE) from the client to the audio and/or video server.

On receiving a STOP message from the client through the common gateway interface, the audio/video server stops execution. The server is not invoked until another user request for audio/video comes in. This eliminates the need for the audio/video server to be up and running always (As the web server is up and running always. we do not need another program also to be ever-awake, as long as awakening the process through the web is possible and acceptable for the application).

Teaching units are created as plain text files containing nothing but pointers to the logical components, viz. a html file, an audio file and a video file. We refer to audio and video as dynamic media components, ane to text and images as static media components (Fig. 60). (As it must be evident, in the protoype the static media component is nothing but a HTML file). Dynamic media files are *nv* format video files and *vat* format audio files.

The web uses a connection oriented protocol (HTTP) to ensure reliable transfer of every bit of data. Audio and video are transferred using a connectionless protocol (RTP) because their timely arrival at the receiving end is more important than the loss of a few packets.

The teaching units conceptually include both hypermedia (text and images with hyperlinks) and time-based (or continuous or dynamic) media (audio and video), so we call them HyTime-like or pseudo-HyTime files. How the CGI programs split the components of teaching units into hypermedia and time-based media is shown in Fig. 60. During creation of teaching units, the flow of data is in the reverse direction.

Because of the fact that every audio and video sequence is associated only with one page of HTML, even if a user were to desert the client machine after he starts receiving audio and video, prolonged useless transmission of bandwidth-intensive media like video does not take place.

5.3.6 Scalability issues in CyberSchool : A Review

- *Content Scalability:*

  Many times it is not possible to stay strictly within the framework imposed by object-oriented principles and still to get the job done. Approaches to organize information as a semantic network must be given some consideration under such cases. Attempts have been made however, in CyberSchool, to create sample

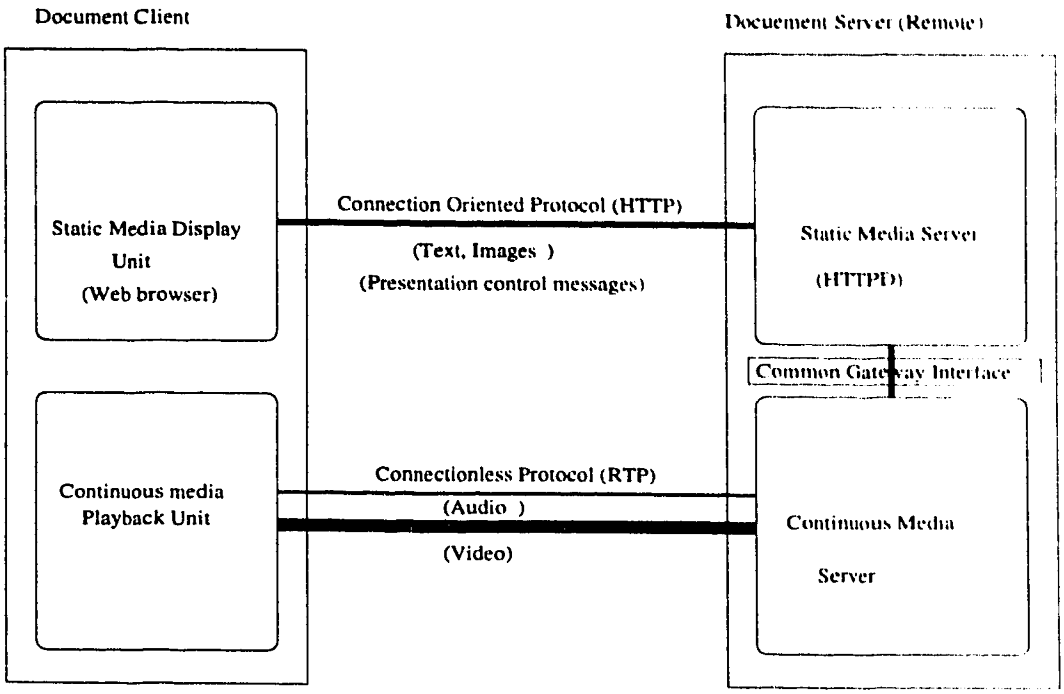

Document Client
Docuement Server (Remote)
Static Media Display Unit (Web browser)
Connection Oriented Protocol (HTTP)
(Text, Images)
(Presentation control messages)
Static Media Server (HTTPD)
Common Gateway Interface
Continuous media Playback Unit
Connectionless Protocol (RTP)
(Audio)
(Video)
Continuous Media Server

Figure 59: Components of the prototype

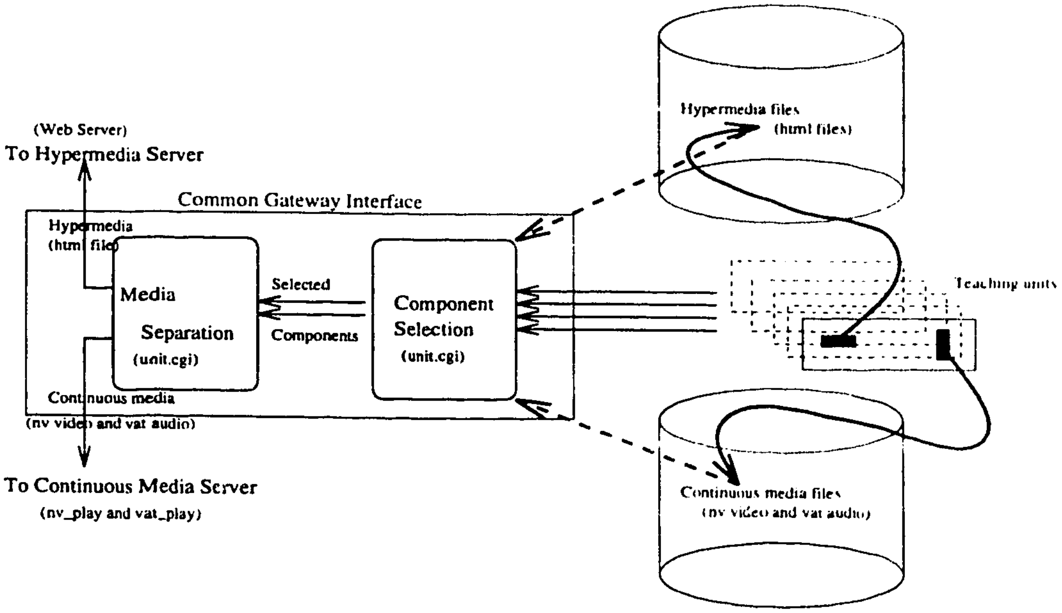

(Web Server)
To Hypermedia Server
Common Gateway Interface
Hypermedia (html files)
Media Separation (unit.cgi)
Selected Components
Component Selection (unit.cgi)
Continuous media (nv video and vat audio)
To Continuous Media Server (nv_play and vat_play)
Hypermedia files (html files)
Teaching units
Continuous media files (nv video and vat audio)

Figure 60: How the CyberSchool server works (during access)

online course material. The topic chosen is **Operating Systems**. Some of the units under this course are shown previously in this section.

- *Component Scalability:*

  The option between audio or video is the only aspect of component scalability available in CyberSchool. Synchronization, rendition and presentation specific actions are not done.

- *Rate Scalability:*

  Gateways have been successfully implemented to convert income real-time video from color to grey-scale, high frame rate to low frame rate. big to small. and high clarity to low clarity. However, no gateway is integrated with CyberSchool.

Chapter 6

Conclusion

We have investigated the support available for individualization of of information delivery. Specifically, we have arrived at ways to create hyperdocuments through formal structural models of hypermedia, and explored ways of supporting component scalability and rate-control of dynamic media. These ways have been implemented and incorporated in the prototype of CyberSchool, an open learning environment. The prototype supports on demand access of course material stored in the form of distributed online multimedia hyperdocuments, and group interaction.

6.1 Findings

- It is possible to prepare orchestrated multimedia material for high bandwidth delivery to multimedia-capable workstations, and yet, to deliver the same to lower levels of technology (less capable stations on less capable networks). This is achieved by making the documents scalable in terms of components (alternate and optional components) and in terms of the rate of video. if video is delivered. Client driven rate control of video is achieved by the use of video gateways; component level scalability is achieved by making the system capable of playing different components of the document independently (intra medium synchronization) and by incorporating encoding-independent inter-media synchronization techniques in the playback engine.

- This point is an elaboration on the issue of client-driven rate control of video: It is possible to reduce the rate of video on the fly be reducing the frame rate. frame size, changing the picture from color to greyscale and/or reducing the quality of individual frames. A gateway working at the pel level can do all of the above but involves a delay (due to its having to wait for the complete frames to arrive before doing starting to re-encode the frame) which makes inter-media synchronization at the receiving end difficult. A gateway working at the transform level, however, involves minimal delay (a few milliseconds) but is not capable of reducing the frame rate - it operates on blocks of transform coefficients, not on frames. A hybrid form of gateway, one that works at the transform-level and offers the functionality of pel-level gateway, requires the original sender of the video to send the difference between the transform coefficients of the pixel blocks instead of the transform coefficients of the difference between the pixel blocks. This requirement makes the hybrid gateway incompatible with the existing standard video codecs (H.261 and MPEG).

- Decomposition of complex lengthy documents into highly cohesive minimally coupled modules, using object oriented principles when applicable, and abstraction as a general guideline, leads individualized information delivery and node-level sharing of the documents. Hypermedia supports user-friendly presentation of the documents so decomposed.

6.2 Future Work

Overall

- Usable tools for authoring multimedia hyperdocuments (preferrably according to the HyTime Standard) have to be developed

In Rate Scalability

- The possibility of doing spatial subsampling in a transform level gateway has to be explored.

- The use of prediction at the transform level has to be assessed in comparison with the use of prediction at the pel level. The variation of transform coefficients with motion has to be studied in depth, so as to find out if it is possible to do better prediction at the transform level than at the pel level.

- Hybrid gateways should be implemented and their performance with respect to pel-level and transform-level gateway has to be evaluated. Although a hybrid gateway involves the same amount of delay as a pel-level gateway. the former can operate at higher input bit-rates.

In Component Scalability

- The nature and degree of the *necessity* and *desirability* of event projection and extent reconciliation have to be determined. Instances have to be studied for common properties, and general event projection and extent reconciliation techniques devised accordingly.
- Implementations of HyTime with focus on the scheduling and rendition modules have to be done, incorporating the above-mentioned techniques.

In Content Scalability

- The question 'whether anything and everything can be viewed in an object oriented way' has to be addressed. Guidelines to make practical compromises that eliminate the difficulty in attempting to do *unnatural* organization in inappropriate cases, have to be arrived at. It should, however. be borne in the minds of the developer that abstraction is the key idea that makes content scalability work.
- Minimal linking between units of information should be evaluated against a more user-friendly form of linking which may include references to nodes that smoothen the information available in the current node (by resolving apparent inconsistencies), nodes that provide complementary information. etc. The disadvantage of the latter (the *The lost in HyperSpace Syndrome*) has to be weighed against its user-friendliness.

6.3 Recommendations

The following recommendations are of general interest to any academic department, and of particular interest to the Department of Computing Science and the **University of Alberta Learning Advantage Program**.

- The ideas presented under Content Scalability have to be tried for various subject material, and used as learning support environment in undergraduate courses. CyberSchool's support for feedback and performance based modification of the course material has to be studied critically, with a view to assessing the amount of class room time that can be effectively replaced by CyberSchool.

- Preparation of formal documents like theses, for online placement, taking advantage of the power of hypermedia, should be encouraged. If this is a success, it can lead to wider easier and immediate access to research material, reducing redundant work and increasing the opportunity to find as much detail as desired about the work done by other people even in the absence of the original researcher. Inclusion of web-mediated demonstration (if applicable) with this documentation should also be encouraged. Such measures, besides resulting in the previously mentioned uses, will take us further into environment-friendliness, by reducing the use of paper. Software copies in disks may be obtained to avoid the *ever-changing* aspect of web documents.

- Measures to integrate presentation, demonstration and documentation of research work have to be taken. The demonstrations should be user-controllable.

- Ways of writing in hypertext (more generally, documentation using hypermedia) have to explored and formalized, to quicken the transition of documentation from paper-form to electronic form. The approaches that under utilize the context free capability of hypermedia like long plain text documents, should be discouraged. The other extreme, providing multiple levels of index into atomic data at the leaf level, though better than the previous approach (in the sense it at least uses several non-terminals in a single node), has to be discouraged as it does not give the reader the ability to prune his reading at some level of abstraction - this ability is necessary to encourage reading and to make the reading satisfy the specific requirements of the reader. Nodes of hyperdocuments have to make sense independently of other nodes, and for this purpose it is necessary to use a combination of terminals and non-terminals.

Bibliography

[ACM 95] The ACM Electronic Publishing Plan, ACM Inc. 1995, http://www.acm.org/pubs/epub_plan.html

[ATM 94] ATM, *User Network Interface 3.0*, ATM Forum, 1994.

[COMP 94] *Comparison of Priority Partition methods*, IEEE INFOCOMM, 1994.

[DCT 89] M. Ghanbari, *Two layer coding of video signals for VBR networks*, IEEE Journal on Sel. Acreas in Comm., vol. 7, no. 5, June 1989.

[DP 94] Brian DeCleene, Pramod Pancha, Magda El. Zarki, Henrik Sorenson, *Comparison of Priority Partition Methods for VBR MPEG.*

[EOPL 93] Brijesh Kumar, *Effect of Packet Losses on End-User Cost in Internetworks with Usage Based Charging*, ACM Comp. Comm. Rev. April 1993.

[FUT 91] L. Bottaci and A. Stewart, *Extending Software into the future*, Hypermedia/Hypertext and Object-oriented Databases, pp. 219, 235, Chapman and Hall, 1991.

[H261 91] Ming Liou, *Overview of the px64 Kbit/s Video Coding Standard*, CACM, vol. 34, No. 4, April 1991.

[H261 93] *Recommendation H.261: Video Codec for Audiovisual Services at p x 64 Kbit/s*, ITU-T White Book, March 1993.

[HYP 94] Steven J. Derose, David G. Durand, *Making Hypermedia Work. A User's Guide to HyTime*, pp. 253-274, Kluwer Academic Publishers, 1994.

[ITOLE 91] Bob Lewis, *Computers in higher education teaching and learning: some aspects of research and development*, The CTISS File, No. 11. Mar. 1991.

[ISDN 95] ISDN, *A User's Guide to Services, Applications and Resources in California*, http://www.pacbell.com/isdn/book/toc.html, Pacific Bell, 1995.

[JPEG 91] Gregory K. Wallace, *The JPEG Still Picture Compression Standard*. CACM, vol. 4, No. 4, April 1991.

[JPEG 93] William B. Pannebaker, Joan L. Mitchell, *JPEG Still Image Compression Standard*, pp. 29-40, Van Nostrand Reinhold, 1993.

[MBone 93] H. Erikson, *MBONE: The multicast backbone*, Proc. INET '93, San Francisco, CA, Aug. 1993.

[MPEG 91] D. J. LeGall, *MPEG: A Video Compression Standard for Multimedia Applications*, CACM, vol. 34, No. 4, April 1991.

[MULT 91] Philippe Delsarte, Benoit Macq and Dirk T.M. Slock, *Efficient Multiresolution Signal Coding via a Signal-Adapted Perfect Reconstruction Filter Pyramid*, ICASSP '91, pp. 2633-3638, May 1991.

[MULT 94] Thierry Turletti, J. C. Bolot, *Issues with multicast video distribution in heterogeneous packet networks*, Proc. 6th International Workshop on PACKET VIDEO, Portland, Oregon, 26-27 Sept. 94, pp. F3.1-3.4.

[NV 94] Ron Frederick, *Experiences with real-time software video compression*, ftp://parc-ftp.xerox.com/net-research/nv, July 1994.

[OOH 89] Kamran Parsaye, Mark Chignell, Setrag Khoshafian, Harry Wong. *Intelligent Databases: Object Oriented, Deductive Hypermedia Technologies*, John Wiley and Sons Inc., 1989.

[PRI 92] B. DeCleene and H. Sorenson, *Prioritized Subband Coding of Video for Packet-Switched Networks*, Proceedings of the 31st Midwest Symposium on Circuits and Systems, Aug. 1992.

[PRI 91] Y. Huang, N. Galatsanos, and H. Dreizen, *Priority DCT Coding of image sequences*, ICASSP '91, pp. 2629-2632, May 1991.

[SUB 91] H. Gharavi, *Multilayer Subband-Based Video Coding*, IEEE Transactions on Communications, Vol. 39, No. 9, Sept. 1991.

[Rate 93] J. C. Bolot and Thierry Turletti, *A rate control mechanism for packet video in the Internet*, Proc. IEEE INFOCOM '94, June 1994, Toronto, pp. 1216-1223.

[RTP 95] H. Schulzrinne, *RTP Profile for Audio and Video Conferences with Minimal Control*, Internet Draft, July 1995. (exp: 12/1/95)

[Sh 95] P. Gbuzynski and G. Shankar, *A layered video coding algorithm for multimedia applications over ATM*, 33rd Annual Allerton Conference on Communication, Control, and Computing, October 4-6, 1995

[UOFA 95] *The University of Alberta Learning Advantage*, The Report of the President's Ad Hoc Task Force on Technology and Learning, http://www.ualberta.ca/CNS/PUBS/LearningAdvantage.html, June 1995.