ABSTRACT

The interactive video (IV) market has been expected to capture a significant share of the huge potential revenues to be generated by the business and residential markets. The level of revenues generated depends on the completion rate of calls the service provider can support, no matter what the IV system or network condition. Thus, a cost-effective, scalable fault-tolerant IV system is needed to maximize the video call completion rate at an affordable cost. This article describes design methodologies for a scalable, fault-tolerant IV system and an IV system design and analysis research prototype called IVSDNA (IV System Designer and Analyzer). The IVSDNA prototype is designed to help network planners and engineers to evaluate quantitative trade-offs (in terms of network communications costs, video storage costs, and degree of system fault tolerance) between two major IV system architectures (centralized and distributed) with a variety of video distribution methods, replication strategies, and fault-tolerant access protocols.

Distributed Interactive Video System Design and Analysis

Tsong-Ho Wu, Bellcore

Ibrahim Korpeoglu, University of Maryland, College Park

Bo-Chao Cheng, Racal Inc.

The interactive video (IV) market has been expected to capture a significant share of the huge potential revenues to be generated by the business and the residential market. Applications of IV include video on demand (VoD), near video on demand (NVoD), transaction services, IV database, and IV games. The level of revenues generated depends on the completion rate of video calls that the service provider can support, no matter what the IV network or system condition is. Thus, a cost-effective, fault-tolerant IV system and its transport network constitute a crucial portion of the video service provider's plan to maximize the video call completion rate at an affordable cost.

To achieve an affordable IV system with the capability of maximizing the call completion rate, the system must have load-balancing capability under normal system conditions as well as rerouting capability when some service delivery system components fail or become congested. Furthermore, these network control functions may need to be distributed throughout the network for a large-scale IV system due to economics, performance, and reliability concerns.

This article describes methodologies for designing a scalable, fault-tolerant IV system and a software prototype (called IVSDNA, for IV system design and analysis) for architecture analysis. The IVSDNA prototype is designed to help network planners and engineers evaluate quantitative trade-offs (in terms of network communications costs, video storage costs, and the degree of system fault tolerance) between two major IV system architectures (centralized and distributed IV systems) with a variety of video distribution methods, replication strategies, and fault-tolerant access protocols. Note that this article only focuses on system design aspects. How the transport network cooperates with the proposed IV system design and reacts to transport network stress conditions is beyond the scope of this article; some solutions can be found in [1, 2].

In the remainder of the article, the following section discusses IV system architecture alternatives, and after that fault-tolerant IV system designs are described, including video distribution methods and fault-tolerant video connection rerouting protocols. The fourth and fifth sections discuss the functional structure of the IVSDNA prototype and some experimental results, respectively. A summary and some remarks are given in the final section.

INTERACTIVE VIDEO SYSTEM ARCHITECTURES

Figure 1 depicts a simplified end-to-end IV network architecture similar to that proposed by the International Telecommunications Union — Telecommunications Standardizations Sector (ITU-T) for VoD applications [3]. This architecture comprises the service control point(s) (SCP(s)); intelligent peripherals (IPs), such as video gateways; video servers, broadband transport nodes, and the access network. Video information providers (VIPs) are connected to the IV backbone network through VIP-network interfaces (VNIs), and video information users (VIUs) are connected to the access network through customer-network interfaces (CNIs). The communication flow and interfaces proposed by ITU-T for VoD applications can be found in [3]. The functional distribution of the IV system determines the system performance and communications costs. IV system designs range from the simplest (but network-resource-consuming) centralized IV system to the complex distributed IV system.

CENTRALIZED INTERACTIVE VIDEO SYSTEM ARCHITECTURE

The basic characteristic of the centralized IV system, as depicted in Fig. 2, is that the video contents are always transported on demand from the central video server (CVS) to the subscribers through the network. The CVS may be located in the

0163-6804/97/$10.00 © 1997 IEEE

VIP's headend office. The online storage and output data rate for the CVS are significantly high in order to support a moderate size of market penetration. The signaling and control processes are performed in a central manner via the SCP-IP system.

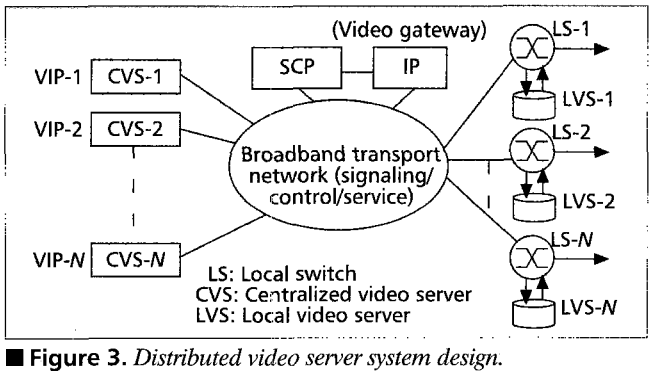
Video transport for the centralized video server system can be supported through a continuous mode or noncontinuous mode. For the continuous transport mode, the video is continuously delivered to the user's set-top box (STB) from the CVS during the connection period; thus, a dedicated video channel is needed for each subscriber. In this transport mode, no local buffer is needed in each end office's (EO's) broadband switch to held any video content. The bandwidth utilization for this case is similar to the case of using STM (synchronous transfer mode) technology to support video transport. For the noncontinuous video transport mode, local buffers are equipped in each EO's switch to hold part of the video content. There are several ways to implement the noncontinuous video transport mode, such as via a store-and-forward system [4] and a periodically staggered framing system [5].

In the centralized IV system, the video delivery is processed and transported by the CVS. If this CVS fails or becomes incapable of supporting existing video connections, these video connections will be dropped, unless there is a backup CVS that can be accessed through the network rerouting capability.

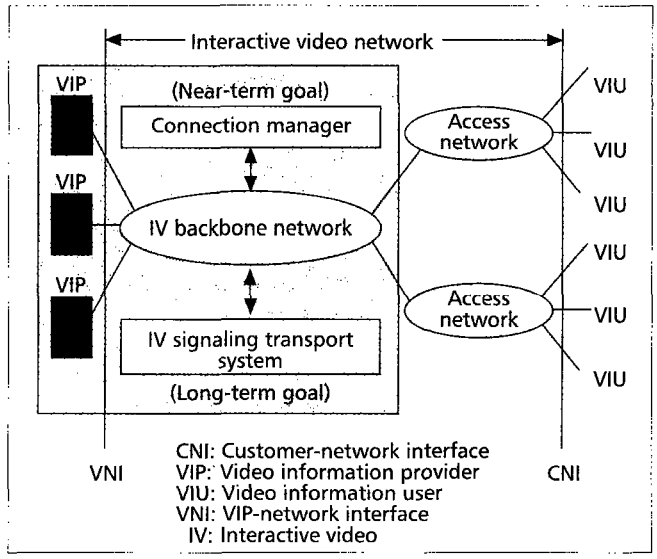
Distributed Interactive Video System Architecture

The distributed IV system (Fig. 3) is to distribute CVS function within the network using the concept of local storage. Such a local storage device is called a *local video server* (LVS). The location identification of LVSs in the network can be provided through the SCP-IP service control system, which may be a centralized or distributed system depending on network size, service penetration rate, and fault tolerance requirements. If an LVS is attached to its local switch, this LVS is called the *default video server* of that local switch. Due to economic concerns, some local switches may not have their own default local video servers. In that case, subscribers served by that local switch will be served by one LVS in its neighbor nodes or the CVS. In this distributed architecture, load balancing and connection rerouting may be required to avoid system congestion and minimize system impacts due to system failures, respectively.

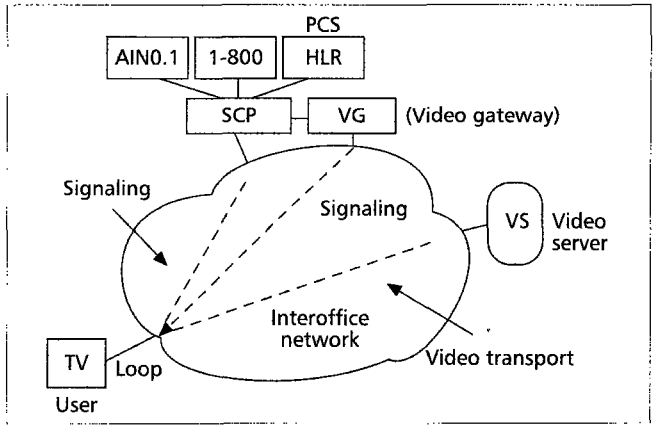
The purpose of distributing the CVS function locally is to reduce the network communications costs by allowing subscribers to access videos through their LVSs. Thus, the distributed IV system architecture design needs to be closely aligned to the subscriber access pattern and the marketing strategy. For example, if the system places the most frequently viewed "hot" videos as close to subscribers as possible, it is



■ **Figure 3.** *Distributed video server system design.*



■ **Figure 1.** *A simplified end-to-end interactive video network architecture.*



■ **Figure 2.** *Centralized interactive video system architecture.*

expected that the network communication costs associated with these hot video accesses can be significantly reduced. In this system, a video archive is still needed in case the LVS cannot provide videos requested by users. Note that each LVS may be a mini-CVS, and its video contents may be downloaded off-line from the CVS and updated periodically (e.g., two weeks). This concept is similar to today's library network where the user is first served by the local library but will be served by other local libraries if the serving local library cannot provide the service. In this case, the Library of Congress serves as the centralized remote library archives, since it stores almost all publications in the United States.

The distributed IV system can be structured in a hierarchical way for system scalability and evolution. It can start from an initial two-level system with a CVS and several LVSs to a system with as many levels of the hierarchy as needed. The number of levels needed depends on the network size, network costs, and network performance requirements (e.g., database access delay).

Compared with the sentralized video server system, the distributed video server system may have a lower average network connection cost and higher system reliability, but at the expense of a significant amount of local storage systems needed. Table 1 summarizes the differences between these two system alternatives. These high-level system trade-offs can be quantified using a system design and analysis tool like the one discussed later.

System Attributes	Centralized System	Distributed System
On-line storage requirement	very high	small
Video storage requirement	lower	higher
Network communications costs	higher	lower
Degree of fault-tolerance	lower	higher
Network control complexity	simpler	relatively complex

■ **Table 1.** *General comparison between centralized and distributed interactive video systems.*

FAULT-TOLERANT INTERACTIVE VIDEO SYSTEM DESIGNS

VIDEO DISTRIBUTION AND REPLICATION STRATEGIES

Once the IV architecture is determined (i.e., centralized or distributed), the next key design issue is how to distribute, place, and replicate videos into LVSs in a cost-effective manner, while meeting the required level of system availability. The resulting system should enable the implementation of a fault-tolerant system access protocol to support video transport under normal system conditions as well as system stress conditions (i.e., system failures, congestion, and/or overload). Note that the content stored in the video depends on the application. For example, each video may store one or more movie for VoD or NVoD applications, or an application program for high-speed Internet Web access applications.

Video Placement in the Centralized Interactive Video System — In the centralized system architecture, each VIP has one CVS in which all the videos of that VIP are placed. The video is not distributed or replicated in the network. Thus, there is not much of a decision to make about video placement in the centralized system. However, the videos in the CVS may be replicated internally, to provide parallel outputs and increase CVS reliability. Reference [6] describes a probabilistic placement method that distributes videos into multiple disks within the CVS to provide the required video throughput rate. Reference [7] proposes and analyzes a hierarchical video server system, in terms of costs and performance, which may be used to implement a large-scale video server system. These methods aim to design a reliable high-throughput video server *per se*, which is beyond the scope of this article. Here we focus only on network-wide video distribution, placement, and replication methods to meet both the cost and system reliability requirements.

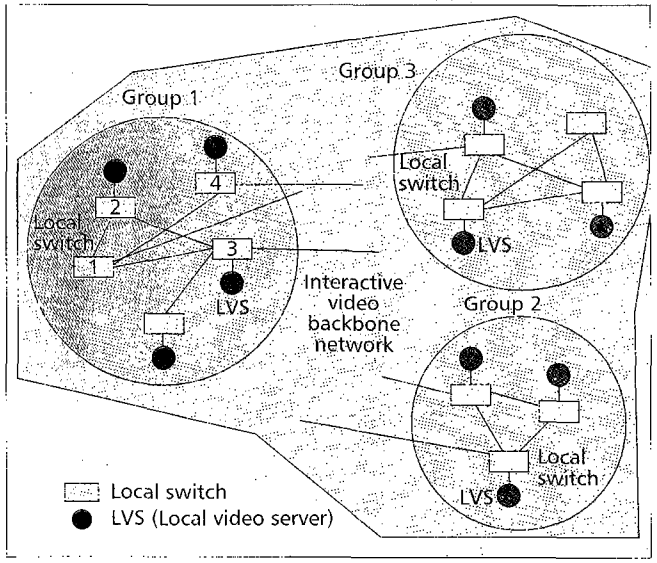
Placement and Replication in the Distributed Interactive Video System — The goal of video placement and replication is to minimize network costs, including the network communications and video storage costs, while meeting the required level of system fault tolerance (i.e., availability). The video placement strategy for the distributed IV system is applied in the entire system. However, the replication strategy is generally applied regionally or locally in order to reduce the replication cost and improve the connection rerouting delay. Because the IVSDNA deals with video placement and replication as an integrated design, video placement and replication are designed based on the "grouping" concept, described below.

Grouping Concept for Video Distribution and Replication — Figure 4 depicts an example of the video server grouping concept. In this example, the IV system is divided into three peer groups, and some user access points (i.e., local switches) may not have their own default video servers (e.g., node 1 in group 1). In this case, the user access in node 1 will be served first by the CVS or one LVS in peer group 1 (e.g., the LVS in node 3). If the local switch has a default video server (e.g., node 2 in group 1), the user access can be directly served by that LVS. Thus, two key design issues here are how many LVSs are needed and where these LVSs should be placed.

The video server group can be partitioned from an administration perspective or based on some design criteria. For example, if the design criterion is to minimize the network communication cost, the IV system may be partitioned into groups based on distance. In this case, LVSs located within a predetermined distance form a group so that local user access can be guaranteed to be served by LVSs in that peer group. In this case, some well-known clustering algorithms, such as the one discussed in [8], can be used to partition the entire system into a number of groups in a cost-effective manner. Other design criteria include community of interest, which may result in a system partition with minimum storage costs, since each local serving area in the community has the similar video access pattern. In this case, the video contents needed for each local access node in the same community are likely to be the same; thus, it may be enough to have one or two LVSs installed in that peer group.

Video Distribution and Replication Methods — Given the following input information, we discuss several methods that have been implemented in the IVSDNA (see the fourth section) for video placement and replication.

- Access statistics for each local switch: the most frequently accessed videos and their access frequencies. This information may be obtained from the marketing study or derived from past statistics over a predetermined time period.
- Grouping information: the number of peer groups with which the local switch is associated.
- Topology information: the number of local switches, the end-to-end video connection routing information, and the distance (in terms of the number of local switches on the path) for each switch pair.
- Server information: the capacities of the CVS and LVSs.
- Video information: the call holding time (in hours) and the length of each video (in gigabytes).



■ **Figure 4.** *Grouping concept for video placement and replication.*

The video distribution and replication design issue can be formulated as an integer programming problem. To simplify the computational complexity, we discuss four heuristics as follows, which have been implemented in the IVSDNA prototype (see the fourth section).

Method 1 — The first method is the most naive. It simply looks up the video access statistics for the local switch to which the server is connected. This method assumes that each local access switch has a default LVS. The videos are placed into the servers according to their access frequencies from that switch. The video most frequently accessed by local users of that switch is placed first to the default video server. The algorithm repeats the process until the default video server reaches its capacity limit.

If the grouping criterion is based on community of interest, in which each node in the peer group has a similar access pattern, it is likely the same video may be stored in almost all video servers in that peer group, resulting in an over-replicated system. If the peer group is formed based on some criteria that result in different access patterns for different nodes in the peer group, video contents stored in each LVS of the peer group may be totally different, which results in a very low level of replication, and thus low system availability.

Method 2 — Method 2 uses an algorithm similar to the optimal file allocation method described in [9]. This method distributes videos into LVSs on a per-peer-group basis. The design objective of method 2 is to minimize the communication cost subject to the server capacity constraint. The basic idea of method 2 is to place as many videos as possible into a predetermined number of video servers (of course, this number may be variable through the entire process). Method 2 does not take video replication into account. The replication version of method 2 can be found in method 4.

The algorithm is repeated for the following process in each peer group until all peer groups have been processed. Let N and M be the numbers of LVSs and videos in the peer group, respectively. Also, let c_i (Gbyte) and s_j (Gbyte) be the capacity of LVS i and the size of video j, respectively. The method initially assumes that each asynchronous transfer mode (ATM) access switch has a default video server. Let f_{ij} be the number of times that video j is requested from LVS i in a fixed time period. The method is summarized as follows:

Step 1 Obtain a weight matrix W with element $\{W_{ij} = f_{ij} \times s_j\}$ (for weighting method 1) or $W_{ij} = f_{ij}$ (for weighting method 2).

Step 2 If all videos in the peer group have been placed (i.e., the weight matrix W is empty), the algorithm stops; otherwise, execute the following procedure: For each video j, choose server i with the maximal value of f_{ij}, and place video j into server i.

Step 3 If the capacity of the chosen server i is exceeded, remove video j from server i, and go to step 4.

Step 4: Update weight matrix W by deleting row i from matrix W; go to step 2.

Method 3 — Unlike method 2, method 3 distributes and replicates the videos in a cost-effective manner. It places at most two copies of a video in different video servers in the same peer group. Since method 3 takes into account the server's capacity constraint, some videos (e.g., infrequently viewed videos) may not be duplicated. The method can easily be modified to have at most n copies of a video within the peer group, where $n > 2$.

Let N and c_i be the number of video servers and the capacity of server i in a peer group, respectively. Given weight matrix $\{f_{ij}\}$ where f_{ij} is the access frequency of movie j stored in video server i. The method is summarized as follows. Among all local video servers, choose the video with the maximum viewing frequency as the target video for placement. Place that target video into two different video servers having the maximum and next to maximum viewing frequencies for that target video, assuming the capacities of these two target servers are not exceeded. If there are more than two servers having the same maximum viewing frequency of the considered video, the algorithm randomly chooses any two of them for video placement. The algorithm then removes that video entry from the statistics of that server, and repeats the same process until all the servers are filled.

LVS 1	LVS 2	LVS 3	LVS 4
(0,20)	(1,35)	(0,25)	(3,15)
(1,19)	(2,25)	(5,20)	(2,13)
(3,14)	(0,17)	(1,10)	(0,10)
(4,12)	(6,10)	(4,7)	(5,9)

■ **Table 2.** *An example of video distribution method 3.*

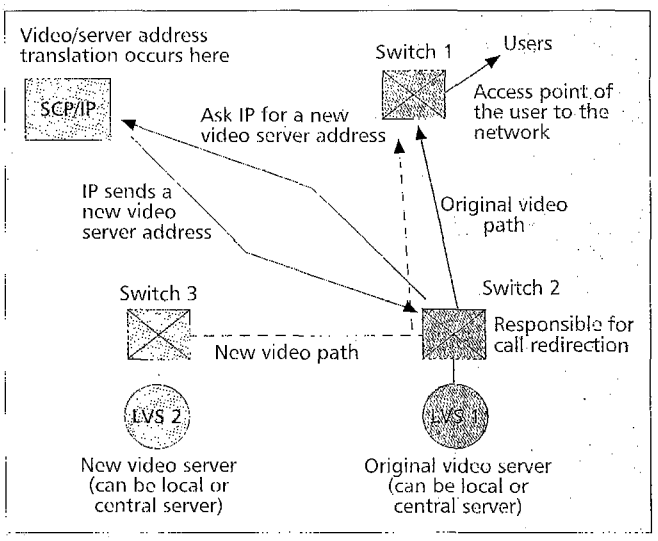
Table 2 shows an example of the algorithm used in method 3. Assume that the video access statistics are shown in the following table, and that each local serving area has its own local video server. In the table, entries (video ID, access frequency) associated with each LVS are sorted in the decreasing order of frequencies.

In Table 2, we first choose the video with the highest frequency, which is video 1 in row 1 (i.e., column LVS 2, video 1 with a frequency of 35). We place this video in LVS 2. Then we search the same video in the statistics of other servers, and choose the one with the next higher frequency. In this case it is column 1 (with frequency 19). We place video 1 in LVS 1. We then delete entries (1,35) and (1,19), from the table. The algorithm repeats the same process by searching the next target video for placement based on the updated statistics table. In this case, video 0 (column 3) and video 2 (column 2) have the same highest frequency. We randomly choose one of them, provided they have not been placed before. If these videos have been placed, they should be ignored in the process and deleted from the statistics table. In the example here, we choose video 0 in column 3 and place it into LVS 3. Similarly, we place the same video in server LVS 1 for replication. The process is repeated until all the servers are filled.

Method 4 — Method 4 uses the algorithm of method 2 to distribute videos in LVSs and a simple algorithm for video replication. Unlike method 3 that is constrained by the server capacity, method 4 does not assume the server capacity constraint in the process. Thus, it guarantees that two copies of the same video are stored in the peer group. Method 4 guarantees 100 percent availability under the single-server failure or congestion condition, at the expense of more storage devices needed. For example, the video storage requirement using method 4 is twice that of using method 2 for video placement.

Fault-Tolerant Video Connection Rerouting Protocols

In a distributed IV system, a fault-tolerant video transport protocol is needed to enable the network to continue to serve video connections in case of server failures, congestion, or other events (e.g., processing overhead). Video connection rerouting needs to be performed in a timely manner to ensure that the stringent quality of service (QoS) demands of real-



■ **Figure 5.** *Concept of the anchor rerouting protocol.*

time video connections are met. Some preliminary standard and requirement references for QoS of video connections can be found in [10, 11].

The video distribution and replication schemes described in the previous subsection ensure that the IV system has at least two copies of the video for access, including a copy of the video stored in the CVS. If an LVS fails before or after connection setup, the system should find an available backup server and re-establish the connection to that backup server as soon as possible. There are two possible fault-tolerant protocols that may be used for video connection rerouting and re-establishment.

Protocol 1: Anchor Rerouting — The first protocol implements the anchor rerouting concept, which is similar to the one used in IS-41 protocol for cellular communications networks [12]. Figure 5 depicts an example of anchor rerouting concept. Customers of local area 1 access the network through their local switch, called the *local access point*. Assume initially that some video connections for these customers are served by LVS 1 via a video path from switch 2 to switch 1 and to customers in local area 1 (Fig. 5). In this example, switch 1 is the local access point of customers, and switch 2 is the connection point of LVS 1. Switch 2 is defined as an *anchor switch* here for these video connections. This example assumes that either local area 1 does not have its default local video server or its default video server cannot serve the requests (e.g., the requested video is not in the default server).

Now assume that LVS 1 fails or is unable to serve some of existing video connections. The anchor rerouting protocol works as follows. When the anchor switch (i.e., switch 2) receives an alert message from LVS 1, it sends a request to the SCP/IP for searching an alternative video server address. The SCP/IP may or may not provide the alternative address for each connection, depending on whether the alternative server is available and can serve the request (e.g., has the video requested). If the SCP/IP returns a valid alternative video server address, the anchor switch then tries to establish a new connection from it to the alternate video server (i.e., LVS 2, which is associated with switch 3 in this example). Thus, a new connection is established (i.e., switch 3 to switch 2 to switch 1) for customers in local area 1, and switch 2 serves the call-forwarding-like feature. If the SCP/IP cannot return a valid alternate video server address for the request, the existing video connection will be dropped, and customers will be informed of service interruption.

The advantage of the anchor rerouting protocol is that it can re-establish the new connection quickly, but at the expense of inefficient network resource allocation for network reconfiguration. Note that, in this protocol, the SCP/IP will be inquired for an alternate video server address whenever a redirected connection is needed. Thus, the IP should be informed of the server unavailability quickly to avoid returning the primary video server address as the alternative video server address. The care should be taken when implementing this protocol.

The above protocol describes the rerouting procedure for an existing video connection. For a new call, its local switch will get an available video server to serve its request from the SCP/IP (i.e., via the signaling process). However, when the local switch tries to establish the video connection to that target video server, the connection may be unable to establish due to the failure of that server. This scenario may occur when the server fails, but the IP does not update its directory tree in time when the new call request arrives. In this case, the anchor switch (i.e., the local switch associated with the failed target video server) may send the inquiry message (the same as in the case for existing connections) with the server unavailability message to the IP to trigger the update of the directory tree and get an available alternate video server address from the updated directory tree, if available.

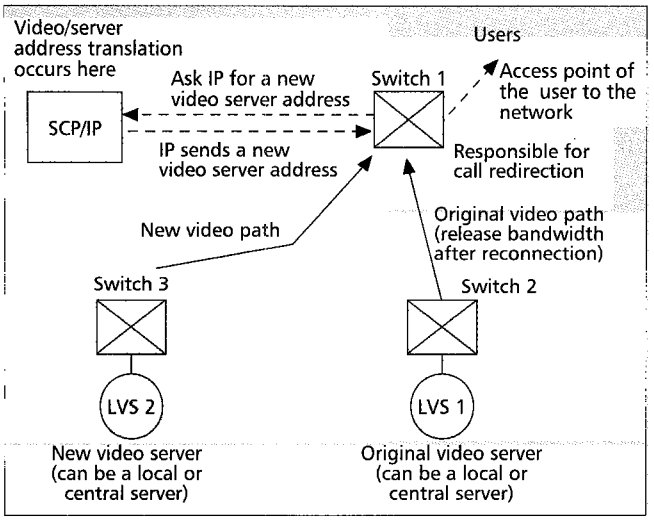
Protocol 2: Dynamic Rerouting — This protocol is similar to protocol 1, except that the local access switch is responsible for connection rerouting. The concept is similar to dynamic routing, described in [12]. Figure 6 depicts an example of dynamic routing for video connection rerouting.

In Fig. 6, the access point switch of users is switch 1. This switch initially asks the SCP/IP for a video server address for requested videos, and then establishes a connection to that server (say LVS 1 in this example). When LVS 1 becomes unavailable for supporting existing connections, some (or all) connections may need to be rerouted to other available servers. In this case, the connection point switch of LVS 1 (i.e., switch 2) informs the access point switch of the users (switch 1) of the server unavailability and which connections have been affected. In the meantime, the video bandwidth of the original connection (between switch 1 and switch 2) is released, and the network makes it available for use to other video connections. The local access point switch, switch 1, sends a reconnection request to the SCP/IP for connections needing to be redirected. If the SCP/IP returns a valid alternative video server address to switch 1 for video connections, switch 1 then re-establishes a new connection to, say, LVS 2 in this example. The new connection may not necessarily be via switch 2. If LVS 2 is available to support this new connection, the required video bandwidth will be allocated, and the remainder of the video can be provided to customers via that new connection (i.e., from switch 1 to switch 3; Fig. 6). If the SCP/IP does not return any valid video server address (e.g., no other server is available), the connection is disconnected.

The major advantage of the dynamic rerouting protocol is that it can use the network resources more efficiently than its anchor-routing (i.e., protocol 1) counterpart. However, protocol 2 may be slower than protocol 1, since the former requires releasing the bandwidth of the original video connection, requesting and re-establishing the new connection, and allocating the bandwidth on the new connection once the new alternate server is verified as being available to serve the required QoS for that connection.

Directory Search Methods in the SCP/IP System

The purpose of the directory system residing in the IP is to help locate the address of the target video server that may serve the customer request. The directory system can be a centralized or



■ **Figure 6.** *Concept of the dynamic routing protocol.*

distributed system, depending on network size and QoS requirements. The detailed directory system designs and their evolution strategies are beyond the scope of this article. Here, we only discuss the directory searching method alternatives based on the centralized directory system architecture implemented in the IVSDNA.

The basic information stored in the video directory system includes:

Video Location Information — The directory should store the location and videos for each local video server. This information should be organized in a way that helps rapid searching.

Server Status Information — The directory should know exactly whether a server is available and can serve the request at the time the call request arrives, so it does not return the address of an unavailable server as the target location for a video connection request. For load balancing among servers, an intelligent directory may also keep track of the current usage status of servers.

Alternative Server Information — The directory should store alternative server addresses upon a reconnection request.

Network Topology Information — This information should help the system to decide on a cost-effective solution if there exists more than one alternative server available to serve the request.

Statistical Information — The peripheral where the directory is implemented can keep some statistical information, like access information from users. This information could be used for the video distribution in the continuous IV system design process.

There are three directory searching methods that have been implemented in the IVSDNA:

Method 1 — Search the default video server for the requested video first. If the video is there, the SCP/IP returns the default server's address; otherwise, it returns the address of the central video server.

Method 2 — It searches the default video server first, as does method 1, but it will search other available alternate local video servers if the default video server is not available or cannot serve the request. The alternate video server searching can be in a predefined or random sequence.

Method 3 — The method is similar to method 2, except that the directory system will search the available video server that is closest to the default video server.

In general, method 1 is the simplest directory search strategy, but it may not provide cost-effective solutions for fault-tolerant video transmission. In contrast, method 3 may provide the most cost-effective solutions for fault-tolerant video transport; however, it requires storing the most information in the directory (e.g., video server information, network topology information). Method 2 is a compromise between methods 1 and 3.

In case alternative address caching is used in the fault-tolerance access protocol, the directory should be capable of providing more than one video server addresses simultaneously upon request.

Interactive Video System Designer and Analyzer (IVSDNA)

Functional Diagram of IVSDNA

The IVSDNA is an IV system design and analysis software prototype designed to analyze a variety of IV system designs for both the centralized and distributed video server system architectures. The IVSDNA is implemented based on a commercially available OPNET simulation platform. OPNET provides a window-based design system and uses a hierarchical approach to integrate network design specification, simulation, and advanced post-processing. Figure 7 depicts the functional diagram of the IVSDNA design.

The main functional modules of an IV network are local switches, video servers, and terminals (representing users). The IVSDNA consists of two major parts: design and analysis. The design part includes network model creation, statistics collection, video distribution, and replication. On the basis of these system design outputs, a fault-tolerant IV system simulator and a video addressing translation directory system may be created to analyze quantitative trade-offs (in terms of the network communications cost, the video storage cost, and the degree of system fault tolerance) for both the centralized and distributed video server systems with different design parameters (e.g., different video distribution strategies and/or directory searching methods).

Cost and Performance Model

The major output parameters from the IVSDNA include the following:

- The communication cost (the switching cost)
- The video storage cost
- The degree of fault tolerance (i.e., availability)
- The video server throughput requirement

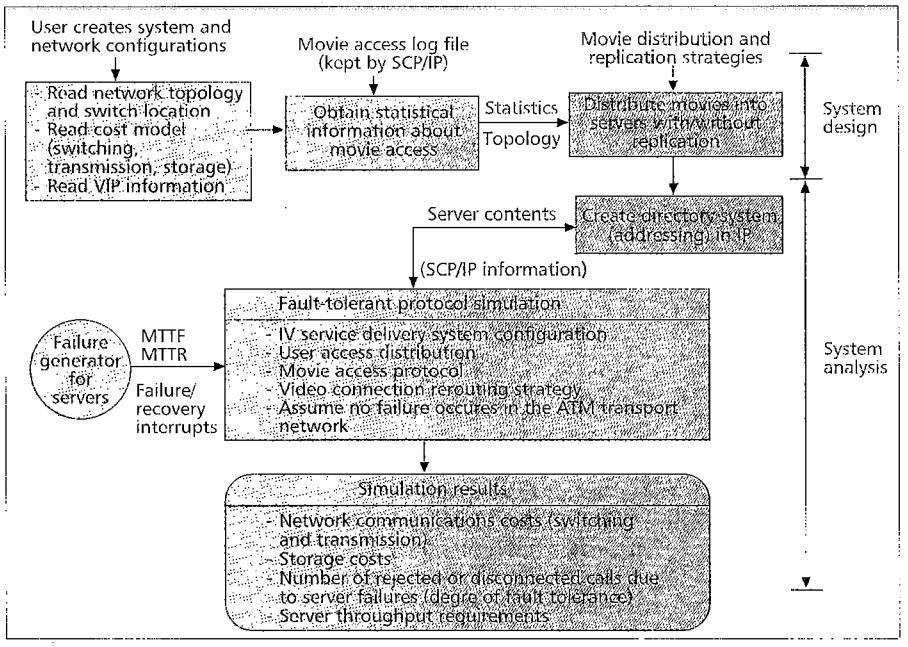
The Communication Cost — We define the communication cost of a video connection as *size x distance*, where *size* is the size of the video (in gigabits) that is transferred from the server to the user, and *distance* is the number of local switches on the video path from the server to the user. The unit of the communications cost that has been implemented in the IVSDNA prototype is *gigabits switched*, which gives the total number of gigabits switched during a video session. The size of a video can be estimated as follows:

$$\text{Size(Gbits)} = (\text{video bandwidth(Mb/s)} \times \text{connection-duration(s)}/1000)$$

For example, assume that a constant bit rate video stream of 3 Mb/s is transferred over the broadband network for 2 hr and traverses 3 switches between the user and the video server; then the communications cost is

$$\text{Cost} = (3 \text{ Mb/s} \times (2 \times 3600\text{s})/1000) \times 3 = 64.8 \text{ Gbits switched.}$$

Note that the cost in terms of gigabits switched can be easily converted to the dollar amount, if the cost per gigabit switched and the link transmission cost are known. Note that if a video path includes n switches, the number of transmission links is $(n - 1)$.



■ **Figure 7.** *Functional diagram of IVSDNA software prototype.*

The Video Storage Cost — The storage cost for each system design is the total number of gigabytes needed to store the videos in the entire IV system based on some particular system architecture being considered. This cost function can easily be converted to the dollar amount if the dollar cost for each megabyte is known.

Degree of Fault Tolerance (System Availability) — System fault tolerance is an important measure for reflecting the call completion rate. It is defined as the percentage of user requests that can be served and completed by the system during a given period of time, no matter what the system condition is (e.g., the server failure, congestion, processing overload). Note that two types of call rejects could occur. The first type of call reject occurs during the connection setup process (i.e., signaling process) due to no available server being found that may serve the user request. The second type of call reject occurs after the connection has been established, but breaks down in the middle of the session, due to conditions of system stress (e.g., server failure, congestion, processing overload).

The level of fault tolerance the system can support depends on the video distribution and replication strategies used during the IV system design phase. It also depends on the system and network control capability that may be able to search for an alternative available video server and redirect the video connection to the available alternate video server.

Server Throughput Requirement — The server throughput is defined as *the number of simultaneous video streams* that the video server can support with the committed QoS. The number of simultaneous stream multiplied by the bandwidth requirement of each stream will give an approximate value for the server throughput in terms of *megabits per second*. The total number of video streams a server can support simultaneously is limited, and the number of simultaneous streams for the same video being supported is also limited due to system engineering and video licensing constraints. If the server throughput exceeds some design threshold, the server may have to use a very-high-power computing platform or an interconnection network of parallel video storage systems to support the required throughput and QoS, which could make the server design very expensive. Thus, one of the major criteria of the IV system design is to distribute and replicate videos as evenly as possible (i.e., load balancing) to reduce the average server throughput requirement.

Experimental Results for VoD Applications

This section discusses some results that can be obtained from the OPNET-based IVSDNA prototype. These results show the capabilities that this IVSDNA system can support, including the quantitative trade-off analysis in terms of the network communications cost, video storage cost, system fault tolerance, and server throughput requirement for different IV system architectures with a variety of design methods, as described in the second and third sections. The outputs obtained from IVSDNA would help network planners and engineers to plan the IV system architecture and the associated evolution strategy. The application considered in this study is VoD.

Network and Communications Flow Models

The network model used in the study is a metropolitan local area transport area's (LATA's) core network. The model network includes 23 major hub nodes, and each hub node is equipped with an ATM switch. In this case study, we also assume that

Parameter	Centralized VoD System	Distributed VoD System
Total commun. costs (svc. and transm.) (Gby's switched)	11,391,471.92	8,332,180.49 (27% less)
Total storage (Gbytes)	771	1291 (67% more)
Peak CVS throughput (simultaneous streams)	553	320 (42% less)
Peak LVS throughput	0	43
Replication ratio	1 (no replication)	2.1 (110% more)
CVS serving ratio	1	0.56
LVS serving ratio	0	0.44
Network and system configurations: 23 ATM switches, 1 CVS (VIP)		
Simulation time: 2,000,000 s		
User request distribution: exponential with mean 100 s		
A total of 259,445 calls		
Video distribution algorithm: method 1		
Directory searching technique: method 1		

■ **Table 3.** *Cost comparison between centralized and distributed interactive video system architectures.*

the SCP/IP is a central system; thus, only one SCP/IP node is used here. The study includes analyzing two major IV system architectures: centralized and distributed. For the distributed VoD system, there are 23 ATM switches, 1 CVS, and 14 LVSs partitioned into 3 peer groups. The centralized VoD system uses the similar configuration as its distributed counterpart, except that no LVS exists. In both cases, the users access the VoD services through their local ATM switches.

The communications flow model used in this study is the one proposed in ITU-T Recommendation I.375 [3]. Note that the communication flow model of [3] used here is a generic flow model. A more updated and detailed communications protocol, DSM-CC (Digital Storage Media-Command and Control) for VoD applications that is specified by DAVIC can be found in [13, 14].

Mean of video distribution	50 s	100 s	150 s
Communications cost per call (Gb/s switched)	26.61	32.17	34.89
Peak CVS throughput (simultaneous video streams)	243	320	362
LVS serving ratio	0.59	0.44	0.36

■ **Table 4.** *Video distribution pattern effects for a distributed interactive video system.*

Simulation Parameters

The major configuration and simulation parameters used in this study are summarized as follows.

Simulation Time — In our simulations, it varies from 2,000,000 to 4,000,000 s.

Request Interarrival — The interarrival time of requests from a local service area is assumed to be exponential with a mean rate of 100 s.

Number of VIPs — For simplicity, only one VIP is assumed (thus, one CVS). The number of videos that can be supplied per VIP is assumed to be 1000.

Video Access Pattern Distribution — This shows the frequency of the access pattern for each video in a local service area. If we have a total of 1000 videos, for example, and if this video pattern distribution is exponential with a mean of 20, it means that the video with ID = 0 is the most frequently accessed, the video with ID = 1000 may be the least frequently accessed one (roughly), and the video with ID = 20 may always be accessed on average. If the mean is close to 0, the distribution curve becomes steeper, so videos with IDs close to 0 may get access more frequently than other videos. If the mean becomes larger, the distribution curve is likely to become more flat.

Video Distribution Algorithm — In a distributed server architecture, four video distribution algorithms may be used.

Directory Search Method — Three directory searching methods may be used here.

LVS and CVS Characteristics — For each simulation, the user can define the capacities of LVSs and the CVS (in gigabytes) and the mean time to failure (MTTF) and mean time to repair (MTTR) attribute values of LVSs and the CVS (in hours).

The video length used in the simulation is randomly selected, which ranges from 30–100 min. With the above simulation parameters and the network model, we discuss some results below.

Experiment 1: Cost Comparison between Centralized and Distributed VoD System Architectures

First we compare the centralized and distributed IV system architectures in terms of the network communications cost, storage cost, and server throughput requirement. Table 3 summarizes the results. Assumptions made for Table 3 include:

- A VoD network model with 23 ATM switches, 1 CVS with 1000 videos stored, and 14 LVSs within 3 peer groups.
- All local user groups (one user group is associated with one ATM switch) use the same video access pattern (i.e., the exponential distribution) with the same mean of 100 s.
- The video distribution method uses method 1 as described earlier.
- The directory search method uses method 1.
- Each simulation duration is 2,000,000 s.

The program was run on a server based on a SUN 4 processor and took about 9 hr for each simulation case (i.e., about 32,432 s for the program execution time plus 3475 s for executing system codes).

The simulation study reported in Table 3 assumes that the video is transported via the constant bit rate Motion Picture Experts Group version 2 (MPEG-2) video stream, and no statistical multiplexing is performed in the network. As shown from Table 3, the network communications cost is reduced by about 27 percent when the VoD system architecture moves from the centralized system to the distributed system, at the expense of 67 percent more video storage needed. However, the distributed VoD system architecture may have other advantages over its centralized counterpart in terms of the replication ratio (about 110 percent more) and a lower CVS throughput requirement (about 42 percent less). The higher the replication ratio is, the higher system availability the system may have. These results are based on the particular video request access ratio distribution (i.e., 56 percent access to the CVS and 44 percent access to LVSs) for the distributed IV system architecture. Thus, as the percentage of LVS access increases, it is expected that the more network communications cost can be reduced; of course, at the expense of more local video storage needed.

The results reported here are based on a 23-switch network model. If the number of switches increases but the distribution of video servers (i.e., the LVS and CVS serving ratios shown in Table 3) remains unchanged, the relative ratio of communications costs to video storage costs remains about the same. To study the steepness effect of the video distribution pattern, we run an experiment with the access mean of 50 s and 150 s (in addition to 100 s). The distribution is again exponential. Table 4 summarizes the results for a distributed VoD system.

As shown in Table 4, if the access pattern is sharp (i.e., hot videos are accessed much more frequently than others — the mean is close to video ID "0"), the communications cost becomes lower. This is because local access is occurring more often (i.e., the percentage of serving videos in that local area increases), which also results in a smaller CVS throughput requirement.

The video distribution algorithm used for Table 4 (i.e., method 1) does not consider the video replication. In the next experiment, we will analyze the cost of adding the replication option and the degree of system fault tolerance that the system can gain.

Experiment 2: Trade-offs between Costs and Fault Tolerance

This experiment studies the quantitative trade-offs between the costs of providing the fault tolerance and the degree of fault tolerance that the system can gain. Assumptions used in this study are the same as those used in experiment 1, except

Parameter	Centralized VoD system	Distributed VoD system
Total commun. costs (svc and transm.) (Gb s switched)	10,832 716.49	7,171,763.41 (34% less)
Total storage (Gbytes)	771	1790 (130% more)
Replication ratio	1	2
Total calls rejected	13,034	6360 (47% less)
Calls rejected before the connection is set up	10,682	5653 (47% less)
Calls rejected after the connection is set up	2,352	707 (70% less)
Total costs due to call rejection (Gb s switched)	55,104.70	28,175.15 (48.3% less)
CVS serving ratio	1	0.67
LVS serving ratio	0	0.6

Network and system configurations: 23 ATM switches, 1 CVS (VIP) with 1000 movies, 14 LVSs (3 groups).

Simulation time: 2,000,000 s

User request distribution: exponential with mean 100 s

A total of 259,429 calls

Video distribution algorithm: method 4 (with replication)

Directory searching technique: method 1

MTTF (CVS) 96 hr, MTTF (LVS) 48 hr, MTTR (CVS & LVS) = 2 hr

■ **Table 5.** *Cost and fault-tolerance tradeoffs.*

that here the video distribution algorithm uses method 4, which takes account of the video replication. The MTTF and MTTR for the CVS are assumed to be 96 hr and 2 hr, respectively. The MTTF and MTTR for the LVS are assumed to be 48 hr and 2 hr, respectively. The experimental results of this study are summarized in Table 5.

Similar to the first experiment, the distributed VoD system has advantages over its centralized counterpart in terms of the network communications cost (i.e., about 34 percent less), at the expense of 130 percent more storage needed. The extra storage needed here for the distributed system is to provide the high degree of system fault tolerance (i.e., its total call rejected rate is about 47 percent less than its centralized counterpart). Since the video is replicated once in the same peer group, the user should find an alternate LVS in its peer group if its default or primary server fails. In this case study, the ratio of calls served by the CVS to calls served by LVSs is 2/3.

SUMMARY

We have discussed methodologies for designing a scalable, fault-tolerant interactive video system and an OPNET-based software prototype (the IVSDNA) designed to help interactive video network planners and engineers analyze alternative interactive video system architectures with a variety of design alternatives. The system architectures implemented in the IVSDNA include the centralized and distributed interactive video systems. The design alternatives include various video placement and replication methods, and the SCP/IP directory searching techniques. The quantitative trade-offs for different interactive video system architectures with different design strategies, in terms of network communications costs, storage cost, the degree of system fault tolerance, and the server throughput requirement can be obtained by using the IVSDNA. Several experiments reported in the article have suggested that the distributed interactive video system may have advantages over the centralized interactive video system in terms of network communication costs and system availability, but at an expense of more storage systems needed. The ratio of network communications cost to video storage cost would determine the economic advantage for the considered interactive video system architectures.

REFERENCES

[1] T-H. Wu, "Emerging Technologies for Fiber Network Survivability," *IEEE Commun. Mag.*, Feb. 1995.

[2] D. Hsing, L. Kant and B. Cheng, "A Restoration System for ATM Networks," *Proc. IEEE MILCOM '96*, Oct. 1996.

[3] ITU-T Draft Rec. I.375, "Network Capabilities to Support Multimedia Services," Geneva, Switzerland, Nov. 14–25, 1994.

[4] A. D. Gleman, *et al.*, "A Stored-and -Forward Architecture for Video on Demand Service," *Canadian J. Electrical and Comp. Eng.*, vol. 18, no. 1, 1993, pp. 37–40.

[5] G. H. Petit and D. Deloddere, "A Video-On-Demand Network Architecture Optimizing Bandwidth and Buffer Storage Resources," *Proc. ISS '95*, Berlin, Germany, Apr. 1995, p. Pr6.

[6] T. D. C. Little and D. Venkatesh, "Probabilistic Assignment of Movies to Storage Devices in a Video-on-Demand System," *Proc. 4th Int'l. Wksp. on Network and Op. Sys. Support for Digital Audio and Video*, Nov. 1992, pp. 231–24.

[7] Y. Doganata and A. N. Tantawi, "A Cost/Performance Study of Video Servers with Hierarchical Storage," *IEEE*, 1994.

[8] Syslo, Deo, and Kowalik, *Discrete Optimization Algorithms*, Prentice Hall, 1983.

[9] D. Bell, *Distributed Database Systems*, Reading, MA: Addison Wesley, 1992.

[10] ITU-T Document, "Integrated Video Service (IVS) Baseline Document," SG 13, Geneva, Switzerland, Mar. 1994.

[11] Bellcore Generic Requirements, "Video Transport Over Asynchronous Transfer Mode (ATM) Generic Requirements," GR-2901-CORE, issue 1, May 1995.

[12] T-H. Wu and L-F. Chang, "Architectures for PCS Mobility Management on ATM Transport Networks," *Proc. Int'l. Conf. Universal Pers. Commun.*, Tokyo, Japan, Nov. 1995.

[13] ISO/IEC 13818-6, "Digital Storage Media Command and Control," *DIS*, July 1996.

[14] Digital Audio Visual Council, DAVIC 1.0 Specification, Jan. 1996.

BIOGRAPHIES

TSONG-HO WU [F '97] received a B.S. in mathematics from the National Taiwan University in 1976, and M.S. and Ph.D. degrees in operations research from the State University of New York at Stony Brook in 1981 and 1983, respectively. He has been with the Network Control Research Department at Bellcore, Red Bank, New Jersey, since 1986, where he is now a director responsible for research on broadband fiber network design, survivable network architectures, emerging technology applications for SONET and ATM virtual path-based networks, ATM-based control and signaling transport for video dial tones services, and PCS mobility management. From 1983 to 1986 he was with Sprint's data communications division as a senior research scientist, where he was responsible for project management and research on planning and designing a new advanced nation-wide packet-switched data network. From 1978 to 1979 he taught at the Department of Mathematics of the National Taiwan University. His current research interests include broadband IP/ATM/SONET transport and control network architectures, PCS mobility management, and distributed database system architectures for interactive video services.

IBRAHIM KORPEOGLU received his B.S. degree from Bilkent University, Ankara, Turkey, in 1994, and his M.S. degree from the University of Maryland-College Park in 1996, both in computer science. He is currently a Ph.D. student at the Mobile Computing and Multimedia Laboratory of the University of Maryland-College Park. His main research interests are in the areas of broadband networks, multimedia networking, mobile and wireless computing, and operating systems.

BO-CHAO CHENG received a B.S. in chemical engineering from the National Central University in 1984 and the Ph.D. in computer and information science from the New Jersey Institute of Technology in 1996. He joined Racal Inc. in 1996.