VIRTUAL MEDIA CENTER: A GENERIC ARCHITECTURE FOR DISTRIBUTED VIDEO-ON-DEMAND SERVICES

by

Sunny K. Y. LEE

A report submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

Department of Computer Science & Engineering
The Chinese University of Hong Kong

April, 2004

Supervised by

Prof. Michael R. Lyu

I. Abstract

There are various technical issues in Video-on-Demand (VOD) service, such as video content allocation, load balancing, transimssion and server selection. Every topic has researched in deep details in the pass. In this report, a generic architecture for Distributed Video-On-Demand (DVOD) services is purposed, designed and implemented. This generic architecture is the result of consideration of various issues addressed in VOD service and will be focus on server selection in both of static and dynamic way. The development and implementation of this generic architecture is called Virtual Media Center (VMC). Server selection is emphasis by collecting Round-Trip-Time (RTT) between the server and client at run-time. Streaming server side resources metrics such as CPU utilization, allocated bandwidth and connected client are also monitored and reported back to VMC Web server for real-time selection process. The technologies used are base on CORBA and Web service. The benefit of CORBA and Web service technologies bring highly interoperability between distributed components and firewall friendly in the Internet. It is also increase the flexibility of different implementation for individual components. The streaming for media content is decoupled as an individual distributed entity that allowable to use heterogeneous streaming server, technology and protocol such as RTSP, MMS and HTTP etc. That is also the gold of this generic architecture that allowing to adapt new streaming technology over time.

II. Acknowledgement

The Generic Architecture for Distributed Video-on-Demand services has been benefited greatly from Prof. Michael Lyu - my project supervisor. His inspiring insight and incomparable patience guided me through my study. I would also like to express my sincerest gratitude to Prof. Lyu for his comments and revision on the script. His encouragements helped me overcome the limitations during the research.

III. Table of content

1 Introduction

Large amount of researches have been done on Video-on-demand (VOD) in the pass. Characteristics of a VOD service are bandwidth demanding and concurrent user capacity is always limited. Most researches are focus on resolving these major bottom necks by video content allocation or replication strategy), load balancing, different transmission method and server selection etc.

As the popularity of the computer network, such as the Internet, most of the VOD systems are target to provide service on computer networks. Computer network consist many autonomous computer systems and VOD service designed in distributed environment is a dominant architecture.

Computers in computer networks can be treated as an autonomous component and the whole computer network can be treated as a distributed system. The heterogeneous among distributed components consist of different hardware, computer networks and operating system. The heterogeneous components in a DVOD system may also include components developed by different programming language and different developers. DVOD system solves the heterogeneous issue by distributed middleware, such as CORBA and provides interoperability.

VOD service has many application specific issues inherited need to address and solve. The most critical one is bandwidth and storage demanding. Thus, there are many researches on video content allocation, replication strategy, transmission method, caching technique, such as use of Set-Top Box (STB), load balancing, storage read/write scheduling, best network path selection, fault tolerance and availability. A successful DVOD system must cover all these issues in its design consideration.

1.1 Contribution

In this report, a DVOD system is designed to build on distributed middleware CORBA. The design is based from consideration of all the issues inherited in VOD service and distributed systems. The selection of the best video streaming server in both static and dynamic metrics is the main focus of this design. The CORAB remote invocation call is proposed as for the client side probe for dynamic server metrics while the static metrics such as the server list will be provide from the result of a remote method invocation. The implementation of the DVOD system will be described and the CORBA IDL for CORBA remote objects definition will be listed. This DVOD system can be used as a framework for a simple and robust architecture for system extension and interoperability.

1.2 Organization of this report

Firstly, this report will discuss and summarize the related work of VOD and DVOD in the pass in section 2. Then the design of the generic architecture for distributed video-ob-demand services will be described in Section3. In section 4, the implementation details and subscriber interface will be described. Finally, section 5 give a conclusion with the advantage and disadvantage and future works.

2. Related Work

2.1. Video-on-Demand Introduction

Video-on-Demand (VOD) is a service that allows users to view any video at any time and can use VCR-like functions, such as fast forward, fast backward, pause, stop and play interactively during the viewing session.

Due to the rapid popular of computer networks in recent year, almost all the VOD system are designed to use computer networks as a delivery channel. The transmission networks properties suitable for VOS service are desired with high bandwidth and follow common protocols. The most common are IP-based network running TCP and UDP [1], [2], [3], [4], [5].

The major streaming media protocol used is the Real Time Streaming Protocol (RTSP), Real-Time Transport Protocol (RTP) and Real-Time Control Protocol (RTCP). It provides direct control of bit stream and was developed for use in unicast and multicast networks for streaming application. RTSP is in the application layer of the OSI reference model. The Resource Reservation Protocol (RSVP) can used to reserve necessary resources at routers along the transmission paths.

Audio and Video file used for streaming are usually in the following format:

- Windows Media Audio (.wma) and meta files (.wax)
- Windows Media Audio (.wmv) and meta files (.wvx)
- Windows Media files, advanced stream format (.asf) and meta files (.asx)
- MPEG-1and MPEG-2 (.mpg, .mpeg, .mpa, .mp2)
- MPEG Layer-3 audio (.mp3)
- MPEG-4
- QuickTime Movie
- RealVideo (.rm)
- RealAudio (.ra)

The most popular format used for video streaming are Windows Media files, advanced stream format, RealVideo and QuickTime. The most popular streaming players are Microsoft Media Player, RealOne and QuickTime.

The VOD service is high bandwidth demanding and limited user capacity [5]. Some high speed networks ATM and CATV, has proposed to provide commercial service such as Digital Theatre [6], [7], [8], [5], [9]. Some proposal also include wireless network, such as W-LAN, W-WAN (GPRS and 3G) in their consideration [10].

In order to facilitate the VOD service provision, users of the VOS system should use their existing equipment, such as a personal computer or TV to access the VOS service. However, some VOD services require to installing a Set-Top box (STB) in the client side to access their services (Figure 1). The major function of STB is used to provide hardware for buffering, caching and video decryption [11], [12], [9], [13].

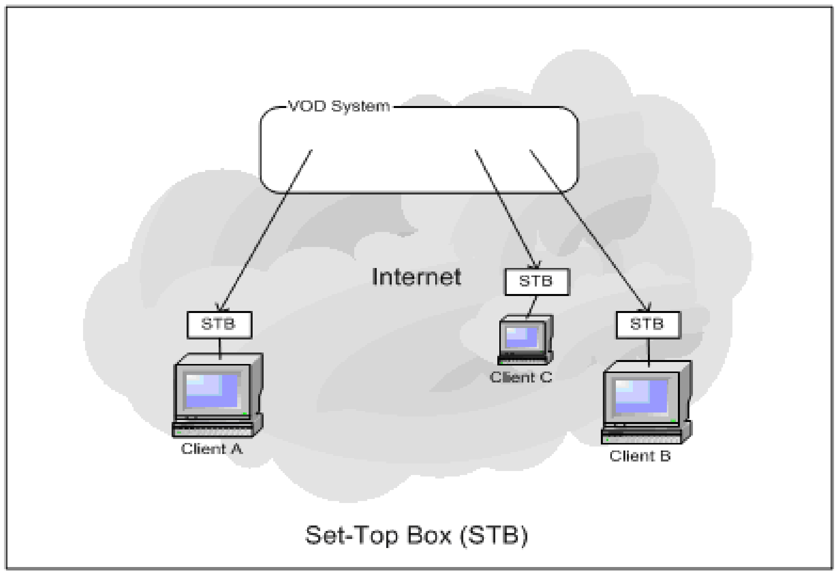


Figure 2.1 Set-Top Box for VOD Service

2.2. Common System Architecture

There are two important properties that make the design and build of a VOD system is a challenge task. The first is the on-demand property the DOVS system. It means any user can access the service at ant time. The second is that when each user accesses the service, they may view different content at the same time. These two properties cause the multicast solution like TV broadcasting is inadequate to use as a solution to delivery contents to the users. Thus, large amount of researches on VOD topic in the pass is related the solution of video contents delivery such as multi-cast, unicast, hybrid of multi-cast [14], [15], [16], [17], [18], [19]. And the second challenge is video resource arrangement, such as video content allocation, admission control and session management [20], [21], [22], [23], [24], [25], [26], [27]. This two issues will be discus the following sections.

2.3. Centralized and Distributed

Broadly speaking, the architecture can be classified in to Centralized VOD [28] and Distributed VOD systems [29], [30]. The criteria to classify a VOD system is centralized or distributed can be looking on the characteristic of its processors, control and data (Enslow 1978).

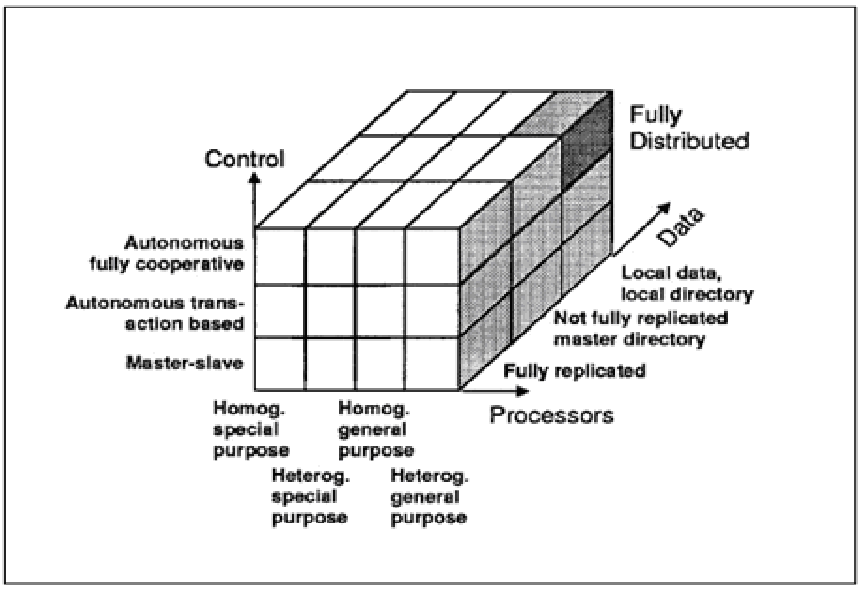


Figure 2.3.1 Distributed System Types (Enslow 1978)

The current hardware cost and bandwidth for a VOD system with centralized hardware control, video content placement and request handling is incapable to fulfill the requirement of VOD service. As a result, most of the current research and proposed VOD systems is designed in various degree of distribution. A Distributed Video-on-Demand (DVOD) system can be builds with various degree of distribution. The most common architecture is the distribution of the video content delivery by multiple streaming servers [2], [31], [32], [33], [34], [30], [35]. Figure 3 shows a common DVOD architecture.

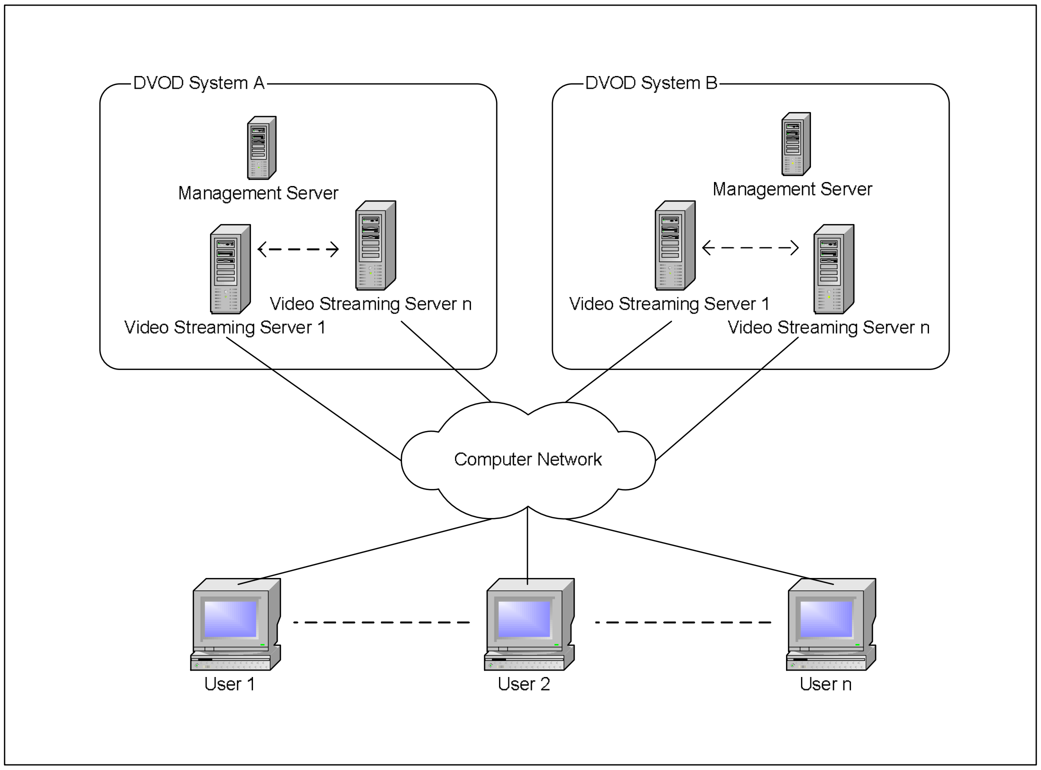


Figure 2.3.2 Common Distributed Video-on-Demand (DVOD) architecture

2.4. Video Content Allocation

Videos have to be stored in permanent storage and the size of videos is huge. It is usually partition the video collection into multiple servers. The storage for video contents can be low speed archive system or high speed storage such as RAID or disk array with programmed read/write control for video contents. In archive system, video is loading to fast read/write storage when request to watch. This is beneficial by cheap and low cost but with slow respond time. In the side is the fully high speed storage for optimal performance but with high cost. Some system stores the expected more demanding videos in high speed storage and store all other videos in archive storage to give a balance on performance and cost [34].

Besides the storage speed, video content allocation in DVOD system is an important factor which can influence greatly in the performance. For example, the location between the user and video in the network can cause respond different and cause traffic congestion to the network.

There are several strategies suggested on how to organize the videos in the storages. The allocation of videos also can base on the popularity of the videos [21], [22] or the user demand for a video [20]. Videos are also replicate to multiple servers to improve performance and increase reliability to the whole system [21], [22]. Besides considering how many copies of a video should place and where it should place, video content allocation in DVOD system must provide access transparency and location transparency to geographical scattered users as a whole.

The replication strategy in DVOD system may increase reliability but also may cast extra network traffics for update. The design of a replication strategy is always need to consider the cost induced to existing system. For example, the replicates of a video can arrange in the network topology in a tree form and replication is arranged in different level between root level and sublevels. The arrangement of this kind is a way to reduce replication update cost.

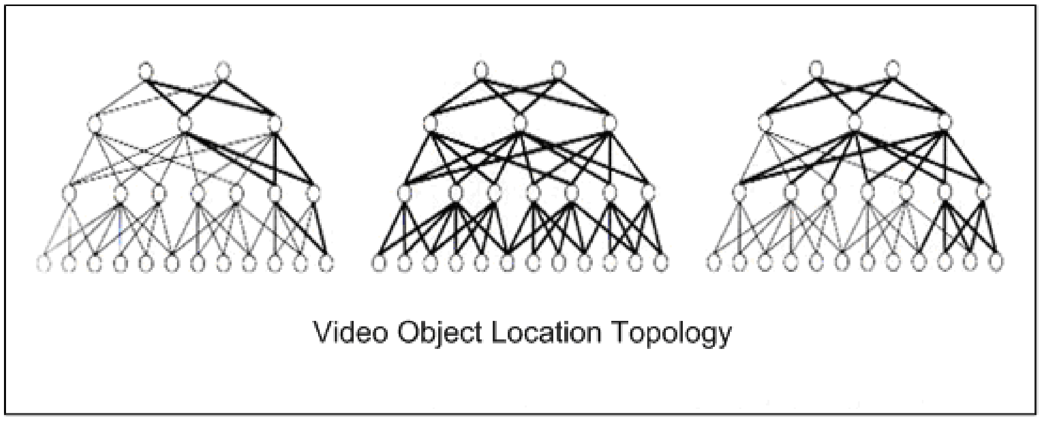


Figure 2.4 Video Replicates Allocated in Tree Form

2.5. Transmission

Delivering video content is bandwidth demanding and this is one of the major bottle-neck to limiting the user capacity. The common computer network types used to participate in DVOD system is listed in Table 1.

Type	Range	Bandwidth (Mbps)	Latency (ms)
LAN	1-2 kms	10-1000	1-10
WAN	Worldwide	0.010-600	100-500
MAN	2-50 kms	1-150	10
Wireless LAN	0.15-1.5 km	2-11	5-20
Wireless WAN	Worldwide	0.010-2	100-500
Internet	Worldwide	0.010-2	100-500

Table 1 Network Types

VOD service is mainly delivery to user with streaming technology. The bandwidth requirement and latency in a computer networks influence the capacity and quality of a VOD service. Compression algorithms are used to reduce the bandwidth requirement in video streaming. Basically, the bandwidth requirement is proportion to the video quality. The trade off for the VOD service provider is between the quality and capacity of concurrent users on the assumption of target network types.

Figure 5 shows some common video format size and bandwidth requirements.

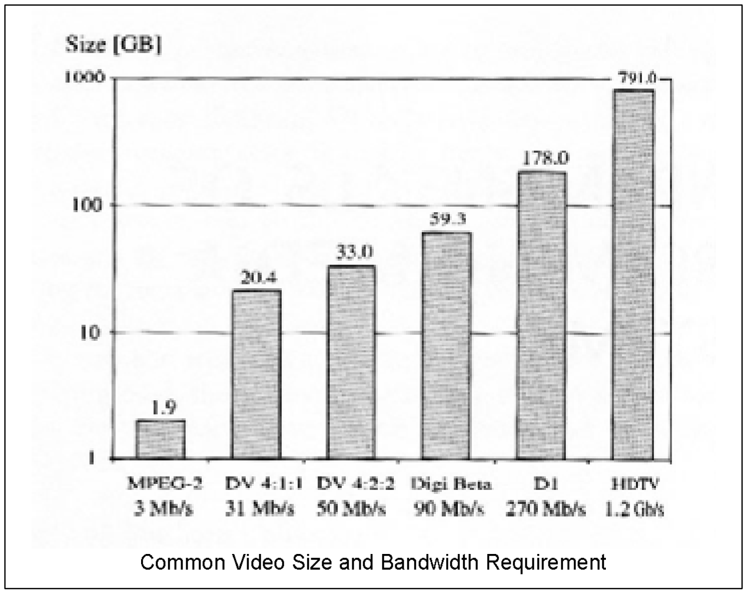


Figure 2.5.1 Common Video Bandwidth Requirements

The quality of service (QoS) in computer network ensures the bandwidth is reserved for a transmission channel. It is no solution to solve in the current equipment with most popular Internet protocol IPv4. The most common solution employs now is to use datagram, such as UDP with buffering at client side to avoid unstable transmission rate or use dedicated network such as CATV to provide VOD service.

VOD service with VCR-like function can be delivered in a unicast channel for every user, and this referred to as True VOD service. True VOD service provides short response time but cost more resources in both CPU time and bandwidth.

There are a number of researches in lecture had proposed to use a combination of unicast channels (Figure 6) and multicast channels (Figure 7) [17], [30] to increase bandwidth utilization and maximizing the user capacity. Some transmission VOD service will repeatedly broadcast selected videos and use chaining technique to provide VCR-Like functions. VOS service with this hybrid technique is referred as Near VOD. The trade off of Near VOD is limited video selection and no interactive control. But Near VOD can beneficial to saving bandwidth requirement and can provide service on providing VOD service through low bandwidth networks.

There is also other proposal such as repeatedly broadcast a video [13], [14], [19] and delivery video content to user via multi-path (Figure 8) [15], [16] or multiple sources [17], [36]. All of these are subject to improve performance and increase capacity but induce extra cost to the pre-processing of video allocation, increase complexity and decrease robustness of video allocation.

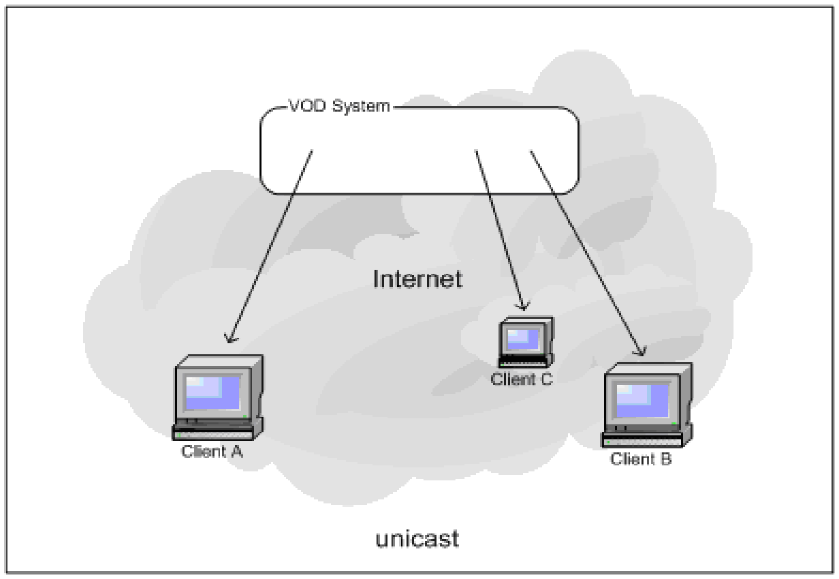


Figure 2.5.2 Unicast Transmission

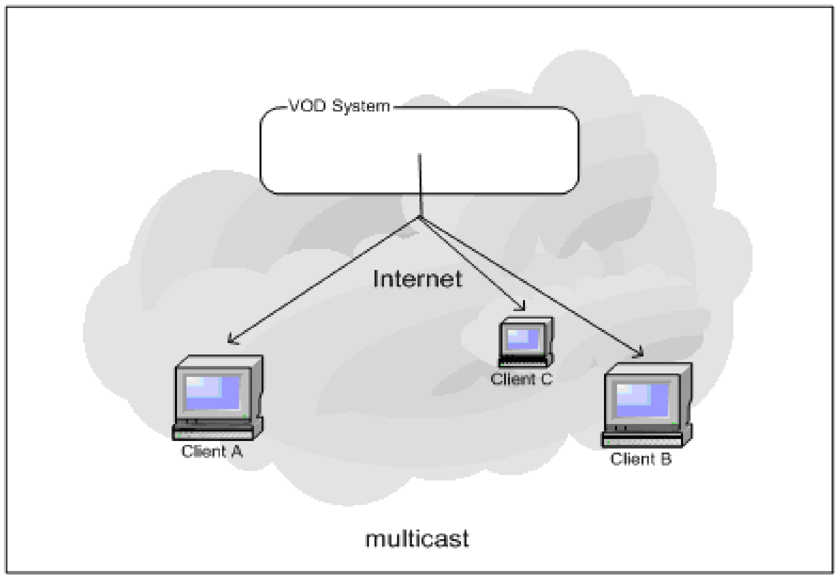


Figure 2.5.3 Multicast Transmissions

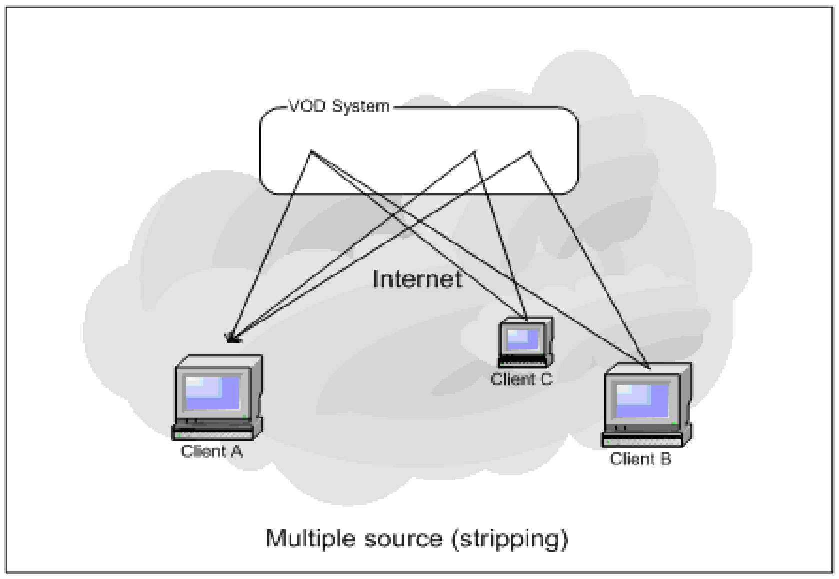


Figure 2.5.4 Multiple Source Transmission

VOD service in DVOD system enables distribution of user request to different locations. As a result, the bandwidth bottle-neck in a single network path is alleviated. The concurrent users and service quality can be improved. However, this is not sufficient for commercial requirement on concurrent user capacity. Different transmission mechanism for DVOD system is still worth to adopt.

Other minor turn up in transmission aspect is that the transmission rate can be change at run-time according to the required video quality or network condition. This technique is called Variable Bit Rate (VBR) transmission and most of the commercial streaming products are supported.

2.6. Replication

Replication can increase performance, reliability and fault tolerance. But incorrect strategy can cause increase storage, increase management cost and no lower performance. Replicate the whole video collection on-line in DVOD system is impractical. So it is usually suggested to replicate some selected video base on some allocation strategy [20], [21], [22]. The video management database, such as video catalogue, resource information etc. can be replication as a whole for fault tolerance purpose. The update issues adhere to replication on video content is minimum, it is due to stored video content is seldom to modify. The most prominent issues are the proper selection of replication source and placement. For example, in the popularity strategy, the prediction is difficult and subject to uncertainty. If the popularity threshold is update dynamically, the update cost for replication is complex and costly for large size video files over bandwidth limited network [35]. Further more, VOD service users are scatted over geographic location. If replicates not located in the shortest network path to the user, it may induce traffic jams and influence other user admission to the VOD service. As a result, replication in DVOD system can increase fault tolerance, such as replication video management data. But the over all performance is subject the uncertainty of the replicates popularity and replicates allocation.

2.7. Load Balancing

Load balancing enable the load in a DVOD system more evenly distributed. The load threshold metric used for load balancing in a DVOD system can be obtain from various resources metrics in static, statistical and dynamic ways.

The load balancing in DVOD system can be classified in to 3 levels: storage, bandwidth and server. Storage load balancing schedule optimized disk read/write control especially for large size video files base on the reading pattern of VOD service. Bandwidth load balancing concern on distributed user request to the shortest network path between the client and server to optimum bandwidth utilization. Server load balancing in VOD service is used to maximize the user capacity. Due to videos in DVOD systems are stored in different servers. Server selection is basically used to select the appropriate video server location for the video requested. If video stored in different servers may replicates according to different video allocation strategy. Besides selecting the right video providers, the server selection process can be used for load balancing purpose. In DVOD system, video allocation and replication strategies are designed to improved system performance and load balancing with server selection [37], [38], [39], [40].

2.8. Server Selection

Client Side Server Selection [37] performs server selection algorithms in the client side with various static, statistical or dynamic metrics. Dynamic estimation use small probe to detect current network resource availability. If the set of probe servers are large, statistical metrics can be use to select a subset of servers for probing.

Dynamic Server Selection [38] enables application-level congestion avoidance. Dynamic server selection consistently outperforms static policies. The selection assumes no knowledge of server location and network topologies and RTT is the basic for selection metrics.

Ping Random [39] is a two-step process for server selection where ping and traceroutes results were used to evaluate the server selection scheme. The first step of Ping Random selects the best 5 well-performing servers. The second step selects the one which is the best among the 5 servers in first step. Ping metrics are used to select a small set of servers, which are called ping-set. Then server is selected from the ping set randomly. Their experiments show that pick up server by RTT metrics performed significantly better on average than those pick up by minimizing hop count or AS count but not as well the Ping Random method. The advantage of this method is that ping is light weight and sensitive to real-time network traffic. But clients request may oscillating and collide among servers.

Random Early Migration (REM) [42] migrate request progressively as the system load increase. REM compares the current service load with threshold and decides whether request migration is needed with a certain probability, which is a function of the service load. REM will take the server load into consideration and achieve less service delay by migrate progressively. But the server for migration must contain the requested videos and the best result will be obtained if with an initialized video allocation algorithm.

Basically, the server selection criteria fall into three classes: static, statistical and dynamic [37]. The static type is based on hardware resources and configuration thresholds, such as hardware loading and connection bandwidth. These static figures take into account the resources capacity but not real time resources availability. Statistical type compute and analysis pass usage records to obtain hints on server resource estimation. This type is less reliable if the user access pattern is high variability. Dynamic type use probes to estimate resources availability on run-time. This type can provide the most up-to-date information of resources availability but add additional traffic and calculation. Among these three types of criteria, hybrid of these three types can provide more rich information but the trade off is the extra resource and slow response time.

The selection metrics used for server selection fall in to four categories. They are router, DNS, server-side and client-side [37]. The most common used metrics in VOD system is listed in Table 2.

Categories	Selection Metrics
Client-side	Geography location, Hop count, RTT, Bandwidth, Random selection, Server load metrics, Server resource configuration Prior respond time
Server-side	Video allocation, Server load metrics, Network path metric [41]

Table 2 common server selection metric

In server-side server selection, it is performed by a VOD service management server. For example, the random early migration [42] performed in the server to decide whether request migration is needed with a certain probability, which is a function of the service load is a server selection in the server side.

In client-side server selection allows the client to select the best server [37], [38] [39]. The server responding time can be used as the selection metric for the best server in dynamic way. The respond T is defined as follows [37].

$T = T_{DNS} + T_{connect} + T_{Latency} + T_{Remaining}$

Where T_{DNS} is the DNS look up time and $T_{connect}$ is the time to establish a TCP/IP connection. The DNS look up can be ignore due to it is usually streaming servers are provided by VOD system management server. The responsibility of the client-side is only need to choose the best server to fulfill its service requirement. The $T_{remaining}$ is a threshold representing the dynamic load of the server. It can be the sum of available bandwidth for the client-server connection and upon server load. $T_{latency}$ is representing the static or statistical threshold of the target server. The client send probe to each target servers to obtain their corresponding T. The smallest T provides an important threshold can be used to select as the best server to serve at run-time.

In addition to server respond time, the network path and traffic condition is also the important metrics that a client can used to calculate the best server selection. Probe is a technique used to obtain dynamic server respond time and network traffic metric without know the server location or network topology [38]. It can be used in various ways. For example, client can send probes to n servers and select the first reply immediately without waiting the rest. Or send probes to all servers, upon receiving the first probe reply, delay for half the reply time to see if other servers respond almost as quickly. After the delay, select the server with the median bandwidth among those have replied [37], Or use the smallest means of 1, 2, 3, 4 or 5 RTT measurements [38]. However, probe induced extra traffic to the network and server load. The design of probe should be carefully consider that to avoid cause impact to the system performance.

The hop count and Round-Trip Time (RTT) obtained from probe are usually important metrics used for server selection [38], [39]. Hop count provide hints to estimate the best path between the client and the server. RTT provide hints to estimate the traffic condition between the client and server. But RTT is better than hop count for prediction metrics (Figure 9) [38], [39]. As the respond time is critical in VOD service and hop count is a poor and inaccurate metric [39], RTT should takes higher priority than hop count at run-time selection. Thus, RTT can be the base for dynamic selection in client side [38].

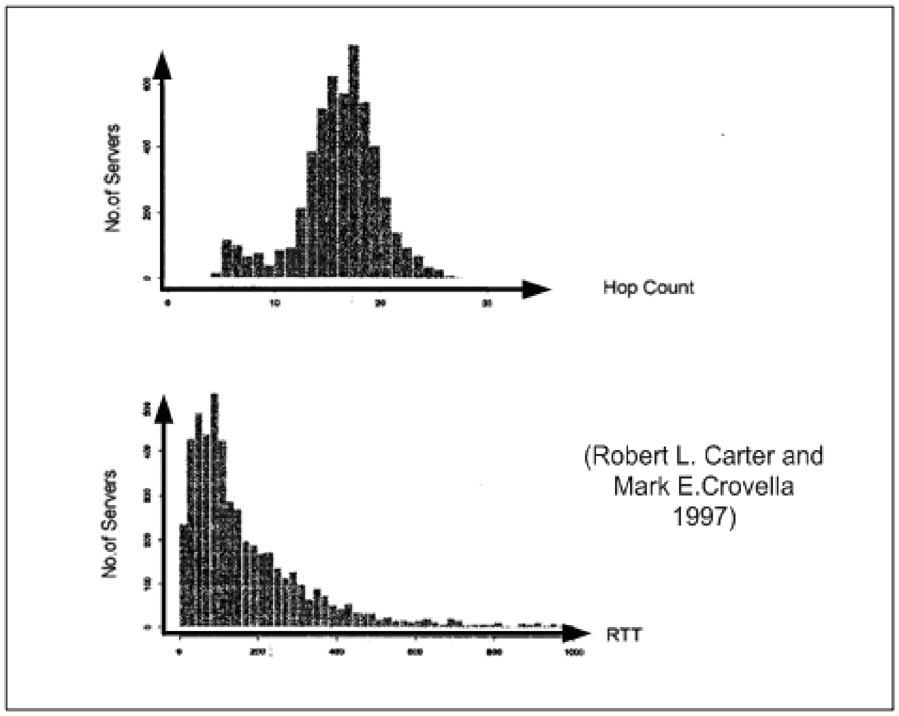


Figure 2.8 Empirical Distributions of Hops and RTT to 5262 Random Servers [38]

The method of obtaining RTT can use simply ping and small size of file transfer. Transfer time with small files give some indication of server performance, but are not always accurate [39]. The ping method shows that it is a better method to all other small file transfer method with simplicity and light weight in bandwidth [39]. In addition to these two methods, a remote method invocation can be used to provide a mixed way of metric to indicate the server respond and RTT metric in distributed system architecture. This method will be discussed in more details and implement in the proposed DVOD system in the following section of this report.

2.9. Fault Tolerance

Fault tolerance in VOD service enable failure of a component, such as a video streaming server fail, the system can still provide service but with lower video selection or performance. Fault Tolerance can increase the availability of a VOD system [2]. Replication and server selection in dynamic ways provide a mask to achieve component or server level fault tolerance. Other level of fault tolerance can be classified as in table 3.

Class	Subclass	Description
Omission failure		A server omits to respond to a request
Response failure		Server responds incorrectly to a request
	Value failure	Return wrong value
	State transmission failure	Has wrong effect on resources (for example, sets wrong values in data items)
Time failure		Response not within a specified time interval
Crash failure		Repeated omission failure: a server repeatedly fails to respond to requests until it is restarted
	Amnesia-crash	A server starts in its initial state, having forgotten its state at the time of the crash
	Pause-crash	A server restarts in the state before the crash
	Halting-crash	Server never restarts

Table 3 Characteristics of Faults

In DVOD system, hardware failure in component achieve by employing storage redundancy, such as RAID, operation system functions, such as clustering etc. to mask the component level failure. The fault tolerance in DVOD system regarding to the Omission failure, Response failure and Time failure can be masked by a distributed middleware such as CORBA. The failure result is a exception in the remote method invocation. The result to the clients may need to handle the failure exception and experience slow performance.

2.10. Admission Control

Admission control in DVOD system controls if a new request can be fulfillment or accept base on the current resources metrics. The purpose is to avoid violating the service quality of others users already in use. Admission control metrics can be from networks resources and CPU load [25], or in threshold-based polices [23], or base on blocking probability [43]. In a DVOD system, it is usually performed the management server as near the client as possible. It is because this can reduce influence to the system and sharing resources to existing users. As a result, the front end process responsible for this task need to obtains sufficient system resource metrics or threshold to make its admission control decision at run-time.

2.11. Summary

VOD service had been research topics for a long time. The proposed topic for VOD service can be grouped into five issues. They are video allocation, admission control, transmission mechanism, replicate policy and server selection. The system architecture is mainly based on distributed components across the component networks. Most proposed DVOD system deals with how to coordinate different components, such as providing video to user from different video streaming servers as a whole to obtain the capacity optimization and access transparency. But it is still few successful commercial VOD service in the markets. Beside the inherit limitation of existing network bandwidth, other non technical issues such as copy right, security and investment interest factors are also influence the popularity of VOD service. However, DVOD system is indeed provide a technical framework to using existing computer networks as a platform for VOD service development.

3 System Design

3.1. Motivation

Multimedia streaming over Internet is popular in recent years. Multimedia technologies such as compression, security and streaming protocol are still under evolution, a framework to provide a generic architecture for distributed video-on-demand (DVOD) services is needed.

As DVOD services are usually operated by different parties and some of the services may not the same. A generic architecture which can provide a standard interface for interoperability of different operators is essential for the framework.

The media contents and sources are various. The procedure for source set up such as real-time broadcast event or ad hoc monitoring for mobile users should dynamic, automatically or as simple as possible.

3.2. Design Goal

The design gold of a generic architecture for distributed video-on-demand (DVOD) is simple. It is only one principle, define clearly the system components and use standard interface between them

3.3. Solution

The ways to define components among a system is not in the topic of this report. And the interface standard among software systems are various and not possible to investigate their details here. As CORBA and Web service are both popular standards in distribute computing environment and Internet application respectively, these two standards are selected as the component interface of the generic architecture for distributed video-on-demand services. The major reasons for choosing CORBA and Web Service for the generic architecture as follows:

CORBA:

- Interoperability middleware for heterogeneous network, computer hardware, operating system and programming language
- Best for interface between heterogeneous streaming server among different technology or content provider
- Drawback: difficult to pass firewall in distributed environment

Web Service:

- Standard API for Web application
- Best for interface between different content providers
- Can pass firewall

As most current CORBA products are difficult to bypassing firewall and no standard ports for configuration, the Web service can used instead of CORBA in case of server interoperate in the Internet environment.

Logically, due to CORBA and Web service are defined as the standard interface between components, the network, computer hardware, operating system and programming language may be different. But in reality, the most normal condition may be a central Web server, a database server and different streaming servers from different venders or products. All may running in different operating system and developed by different tools.

In the user side, Internet browse is a normal access interface. The capability to decode the streaming is determinate by if the user installed the appropriate media player component for the target streaming source. Different streaming source may employ different streaming technologies. Clients must install the corresponding component or player to decode the steaming if it is not stalled before.

3.4. Design Consideration

Video Content Allocation Issues

Due to the uncertainty of video popularity prediction and replicate policy is tightly related to video allocation, the DVOD in this design assumed that video collection is partitioned on multiple streaming servers and some of videos may have multiple copies in different streaming server. As the beneficial obtain from simplicity is out weight the cost of uncertainty, no any allocation method will use.

Transmission Issues

Another assumption is that transmission is based on unicast and the whole video file is stored in a single location. This simply the implementation and less cost for video pre-processing, such as video stripping on multiple servers, adaptive chaining etc. The most important benefit is there is no need to use STB and simplify the client side setup.

Heterogeneous and Middleware Issues

The implementation is flexible in CORBA and has been chosen as the framework for streaming service [12], [44] and VOD service in mobile environment [45]. The limitation based on the supporting programming language binding in CORBA IDL and supporting platform for ORB products. It is theoretically and practically to implementation heterogeneous components in different network and operating systems. In the DVOD system proposed in this report, the two CORBA remote objects will be implemented in the UNIX platform and the client interface will implementation in Windows platform. Video streaming is decided to spawn a new thread on per view basis. In other CORBA implementation of DVOD system, the video streaming part can be a third part product with integrated control interface, such as CORBA or COM to control the delivery of streaming video to clients.

Fault Tolerance Issues

All video information, user account, resources metrics and other management data are stored in a central database installed in the Video Management Server without replication and any fault tolerance protection. The main different to practical system is that replication and redundant of database increase availability and fault tolerance.

Location Transparency Issues

The CORBA Naming Service will be used for client to the look up the Video Management Server. This provides the location transparency of the Video Management Server to the Video Clients but every client must setup to locate the CORBA Naming service in its configuration.

Resource Metrics Update Issues

The CORBA Event Service will be used to provide real time resources metrics notification to subscribed CORBA remote objects. The major resources metrics in the Event Service channel is the remote object in the Video Streaming Server. It is because it provides real time loading and resources metrics for both the Video Management Server and Video Client in the server selection process.

As a result, the design and implementation of this DVOD system is mainly focus on the admission control and server selection with both static and dynamic resource metrics. And demonstrate the use of CORBA as the development platform for DVOD system

3.5. System Architecture

3.5.1 Roles

The roles in a generic architecture for distributed video-on-demand service can be classified as follows:

- System operator
- Content provider
- Subscriber

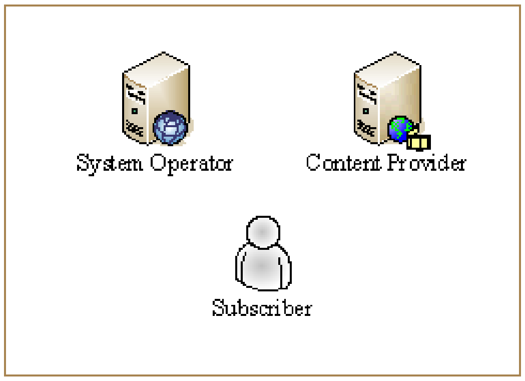


Figure 3.5.1 VMC Roles

System operator is the party who provide the browsing service to subscribers. Content providers register them service or content to system operator and subscribers can browse variety of service such as Video-On-Demand, Real-Time TV, Real-Time Web Cam etc., and choosing the content provider to serve them.

Content provider provides the media content streaming directly to subscribers. The physical media content may be a streaming file or an audio/video capture from hardware device. The source of the media content is usually called streaming server. Content provide may equip their streaming severs in their own sites. So that a system operator may consist many streaming servers from different content providers and those streaming servers may scatted in different geographic location among the Internet.

Subscriber refers to user of the system who retrieving the streaming content from any content provider through the system operator. The user will access the system operator's Web site to obtain the streaming server location and then retrieve the streaming directly from the streaming server.

3.5.2 Major Components

The components in a generic architecture for distributed video-on-demand service can be classified as follows:

- Web Server
- Remote Web Server
- Database Server
- Streaming Server

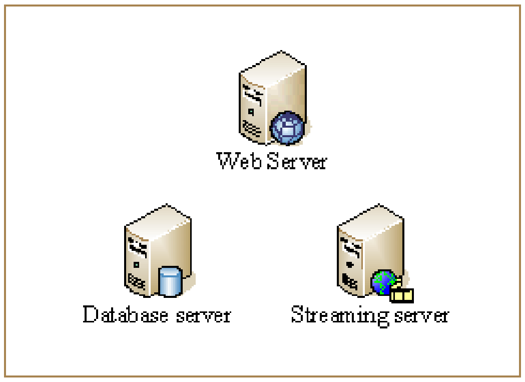


Figure 3.5.2 VMC Major Components

Web server is the access point or portal of the distributed video-on-demand services. It provide catalogue of media content to subscribers. Remote Web server is the same as the Web server except its catalogue content is come from a remote Web server. The Remote Web server is similar a proxy. The function of a Remote Web server can be for load balancing or provide different version of catalogue form, such as for mobile device etc.

Database server is the central repository for all catalogue information, system configuration and meta data in the distributed video-on-demand services.

Streaming server is the server which provides media streaming service to subscribers. Streaming server located within Intranet of the system operator is called local Streaming server. Streaming server located outside the Intranet, such as in the Internet is called Remote Streaming server. Remote streaming server is used to minimize the distance between the subscribers and the media source in order to improve bandwidth utilization and load balancing. Content provider may provide their one Streaming server group to one or several System operators.

3.5.3 Distributed Video-On-Demand Scenario

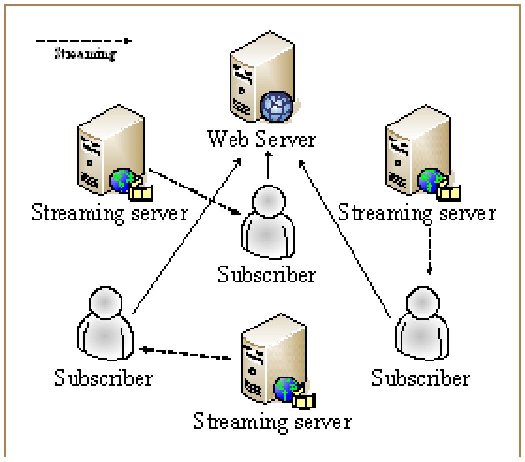


Figure 3.5.3 DVOD Scenarios

The most typical scenario for the most typical services is the distributed video-on-demand service. The configuration is usually a central Web server plus many distributed streaming servers scattered in different geographic location in the Internet. Subscribers will be distributed to the most suitable streaming server to retrieve the target media streaming.

3.5.4 Real-Time Broadcast Scenario

Basically, real-time broadcast is streaming media signals from audio/video device to subscribers. It is not only limited to Web Cam or TV turner card.

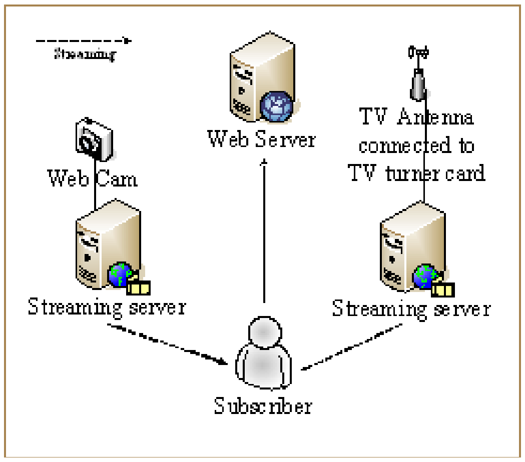


Figure 3.5.4 Real-Time Broadcast Scenarios

3.5.5 Mobile Subscriber Scenario

Remote Web server provide media content catalogue to mobile subscribers through local access point. The source of media content catalogue is consumed from another Web server which is the Web server of the system operator. The purpose of the Remote Web server is used to provide different version of catalogue to appropriate mobile device, such as Pocket PC or Smartphone. Remote Web server may also provided by system operator for load balancing purpose or operated by another partity who target to mobile device in a specific wireless coverage area.

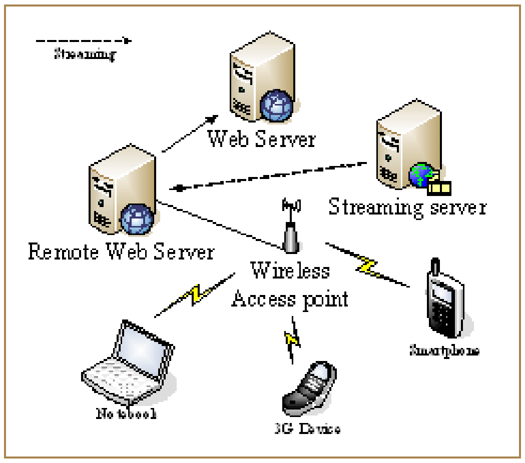


Figure 3.5.5 Mobile Subscriber Scenarios

3.5.6 Ad hoc Multimedia Streaming Scenario

Ad hoc multimedia allow mobile users to become a media content provide at anytime.

Mobile subscribers may also become a content provider when they register their Web Cam or streaming service to the system operator’s Web server.

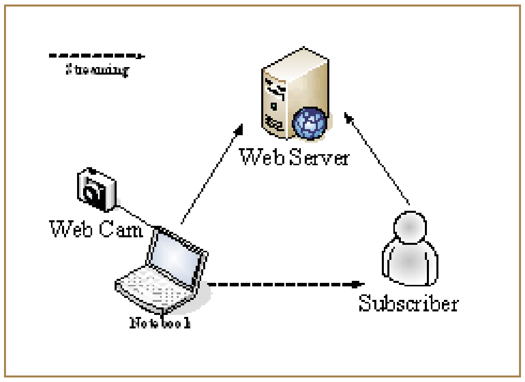


Figure 3.5.6 Ad Hoc Multimedia Streaming Scenarios

3.5.7 Local and Remote Server Scenario

The common configuration for Distributed Video-On-Demand services consist multiple Streaming servers and remote Web servers located in different geographic area in the Internet. Group of servers located in a LAN are usually protected by firewall. CORBA is proposed to be the standard component interface for interoperate behind the firewall while component interoperate outside the firewall will be proposed to use Web service

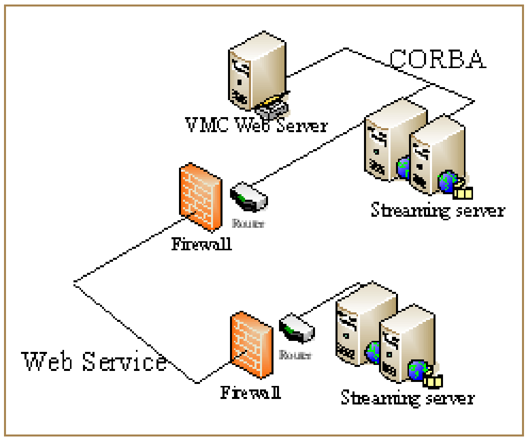


Figure 3.5.7 Local and Remote Scenarios

3.5.8 Server Selection Scenario

Server selection occurs when subscriber order to watch a multimedia streaming content. It is by default, the system operator short list the most appropriate servers which available the ordered content for subscriber. The metrics for ordering include Round-Trip-Time (RTT), Allocated Band Width for clients

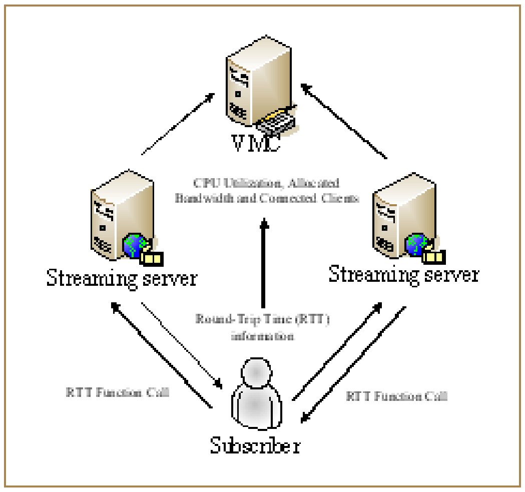

VMC
CPU Utilization, Allocated Bandwidth and Connected Clients
Streaming server
Streaming server
Round-Trip Time (RTT) information
RTT Function Call
RTT Function Call
Subscriber

Figure 3.5.8 Server Selection Scenarios

and number of connected clients.

3.5.9 RTT Collection Scenario

The optimal solution to obtain RTT figure between the subscriber and streaming server is by running an echo function in the server by the subscriber. The subscribers invoke the echo function each time it need to select the best server. But this method needs the subscriber capable to invoke the server object in the server and this is usually need installation of client software, such as install a client object and ORB in CORBA technology. The less optimal solution to obtain RTT is by collection the RTT figures by a remote Web server which is located nearest to the subscribers. The remote Web server will run the echo function periodically instead of running by subscribers.

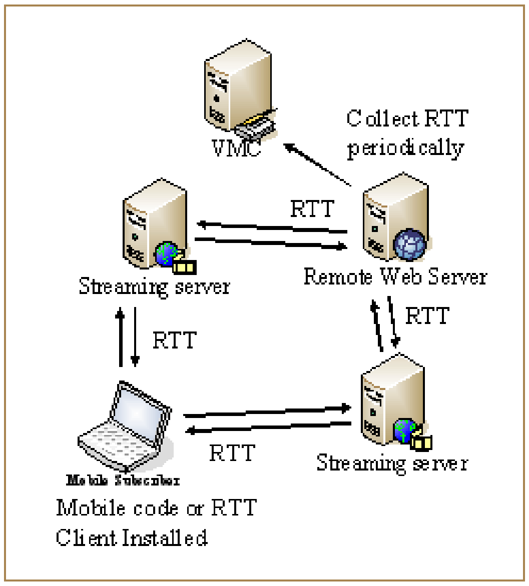


Figure 3.5.9 RTT Collection Scenarios

3.5.10 Logical Component Architecture

The logical component architecture of the generic distributed Video-On-Demand services is illustrated below. Components which distributed in different geographic location are grouped over a grey block. Some components may further separated in different machine (such as the database server) for load balancing is not illustrated here as it is not the major purpose here. The streaming server, remote Web server and client application may have multiple instances as multiple subscribers may retrieve their media contents from different streaming servers concurrently.

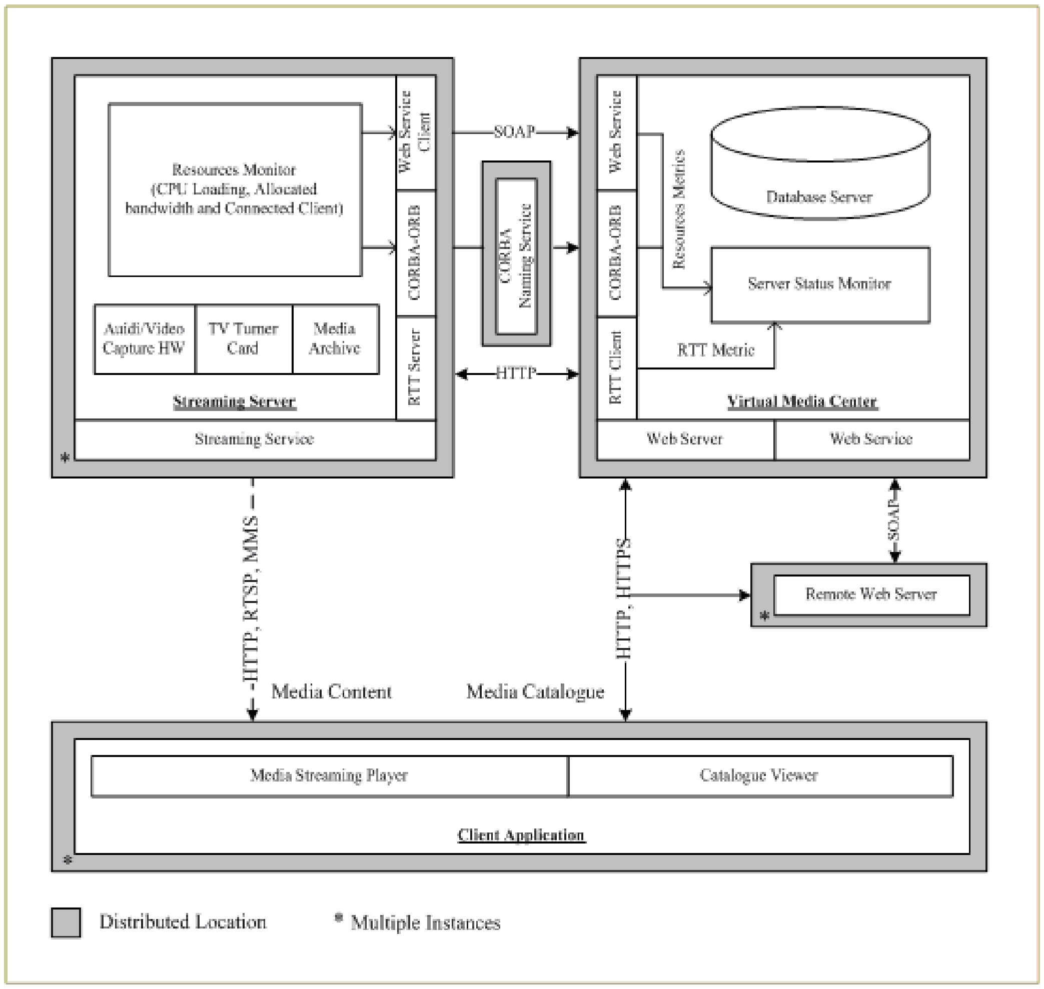


Figure 3.5.10 Distributed Component Architecture

In figure 3.5.10, it is illustrate all major components in the generic architecture. The components are mainly grouped together form 5 independent groups which are easy for running in distributed locations or machines (illustrated by grey border). The VMC Web server, Web service and CORBA naming service only need one instant within a single distributed video-on-demand services system. The streaming server and clients may have many instances. The remote Web server is optional. If it exists, there may have used to holding different version of media content catalogue (such as Pocket PC) or used to collect RTT for mobile devices (RTT Agent). RTT collection provide dynamic resources metrics and avoid install RTT client code into client’s device.

3.5.11 Component Collaboration

The collaboration between components is show in Figure 12 below.

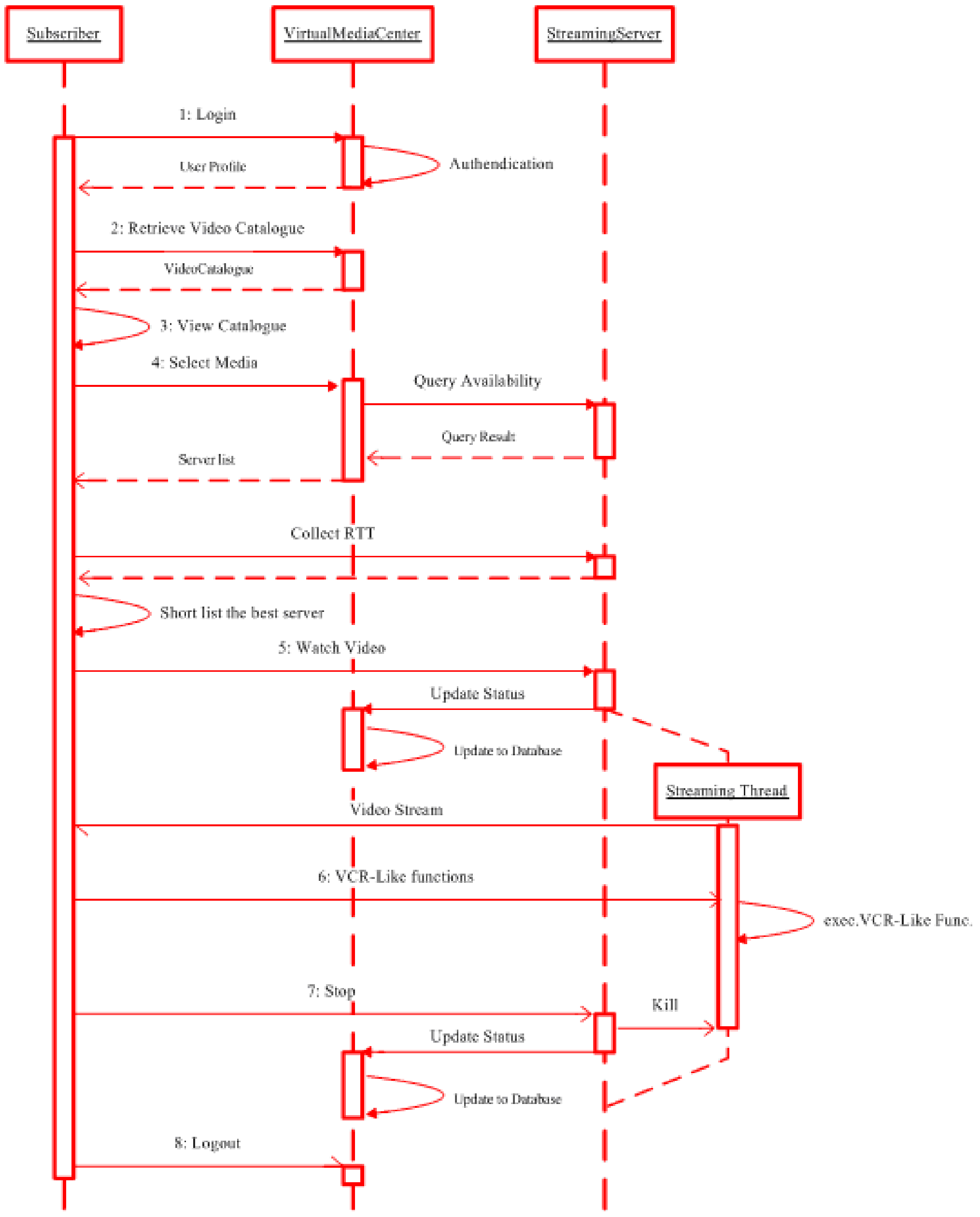
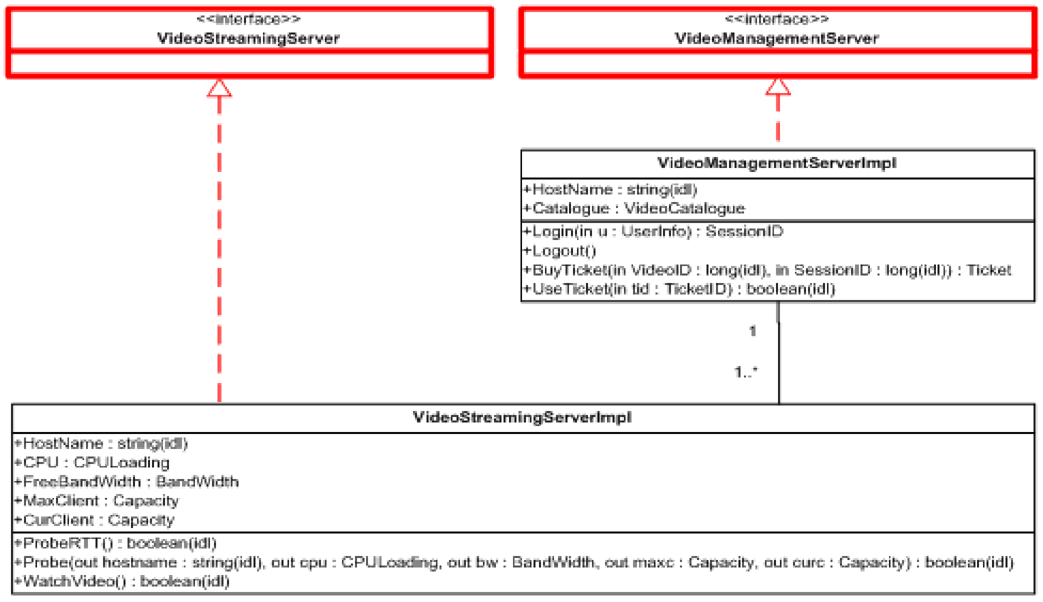


Figure 3.5.11 Sequence Diagram of a Video Session

3.6 Interface and Class Definitions

The interface for remote objects should be implemented by the supported programming language binding of CORBA in the server. The VideoManagementServerImpl and VideoStreamingServerImpl classes in Figure 13 represented the implementation class.



<<interface>>
VideoStreamingServer
<<interface>>
VideoManagementServer
VideoManagementServerImpl
+HostName : string(idl)
+Catalogue : VideoCatalogue
+Login(in u : UserInfo) : SessionID
+Logout()
+BuyTicket(in VideoID : long(idl), in SessionID : long(idl)) : Ticket
+UseTicket(in tid : TicketID) : boolean(idl)
1
1..*
VideoStreamingServerImpl
+HostName : string(idl)
+CPU : CPULoading
+FreeBandWidth : BandWidth
+MaxClient : Capacity
+CurClient : Capacity
+ProbeRTT() : boolean(idl)
+Probe(out hostname : string(idl), out cpu : CPULoading, out bw : BandWidth, out maxc : Capacity, out curc : Capacity) : boolean(idl)
+WatchVideo() : boolean(idl)

Figure 3.6 Class to implementation CORBA Interface

3.7 Data Structure

Figure 14 shows the data structures will be defined in CORBA IDL.

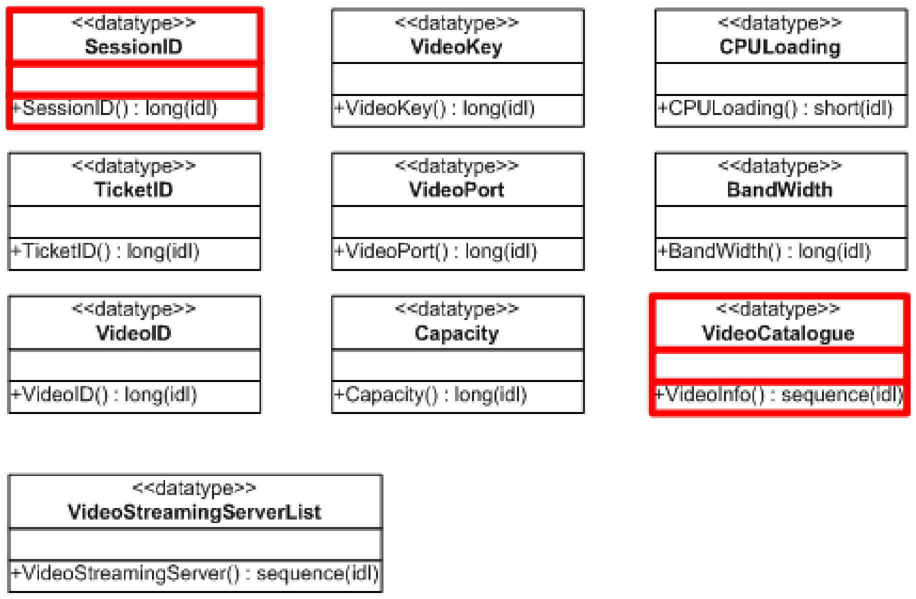
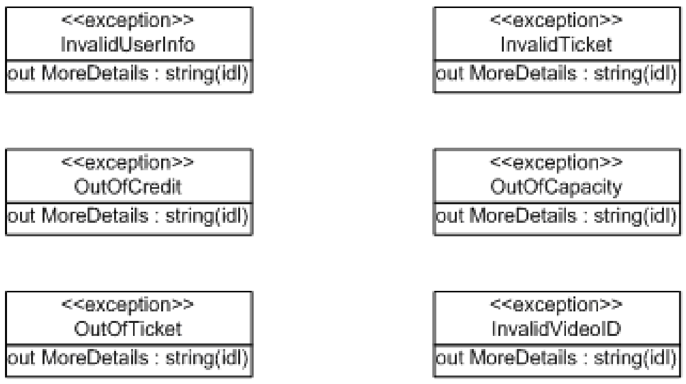


Figure 3.7 Data Structure for CORBA Remote Objects

3.8 Exceptions

Exceptions are used to inform more descriptive system error information in run-time. CORBA provide exception for remote method invocation for error handling. Figure 15 shows the exception will be raise in the CORBA interface.



<<exception>> InvalidUserInfo
out MoreDetails : string(idl)
<<exception>> InvalidTicket
out MoreDetails : string(idl)
<<exception>> OutOfCredit
out MoreDetails : string(idl)
<<exception>> OutOfCapacity
out MoreDetails : string(idl)
<<exception>> OutOfTicket
out MoreDetails : string(idl)
<<exception>> InvalidVideoID
out MoreDetails : string(idl)

Figure 3.8 Exceptions for Remote Object Methods

4 Implementation

4.1. Overview

The implementation to the Distributed Video-on-Demand services is called Virtual Media Center (VMC). The implemented services in the VMC are as follows:

- Video-on-Demand (VOD)
- Real-Time TV
- Real-Time Web Cam Monitor
- Ad Hoc Streaming
- Auto Server Selection
- CORBA API
- Web Service API

Basically, VOD, Real-Time TV and Real-Time Web Cam monitor are the same. All are streaming service in the point of subscribers view. The different is on the content. But in the back end, VOD is different to Real-Time TV and Real-Time Web Cam monitor. The source of VOD is physical file in supported streaming format, while the sources of Real-Time TV and Real-Time Web Cam monitor is audio/video signals directly captured from hardware device. Ad Hoc streaming is another form of the VOD, Real-Time TV and Real-Time Web Cam Monitor services. The different is Ad Hoc streaming emphasis the source location is mobile. The VMC mask the mobility of streaming source by providing a fixed access location. Auto server selection is important in VMC as if subscriber fail to obtain the media content from the most suitable server will cause network traffic jams and lower system capacity. The selection criteria is by short list a list of available servers from some collected resource metrics at the subscriber order a media to view. The last two services CORBA API and Web service API are application interfaces to meet the design gold to increase interoperability and bypassing firewall.

A typical implementation consist a central Web server, Web service interface, local streaming servers, remote Web server and remote streaming servers, subscribers in PC, mobile and wireless device is illustrated in the following implementation illustration.

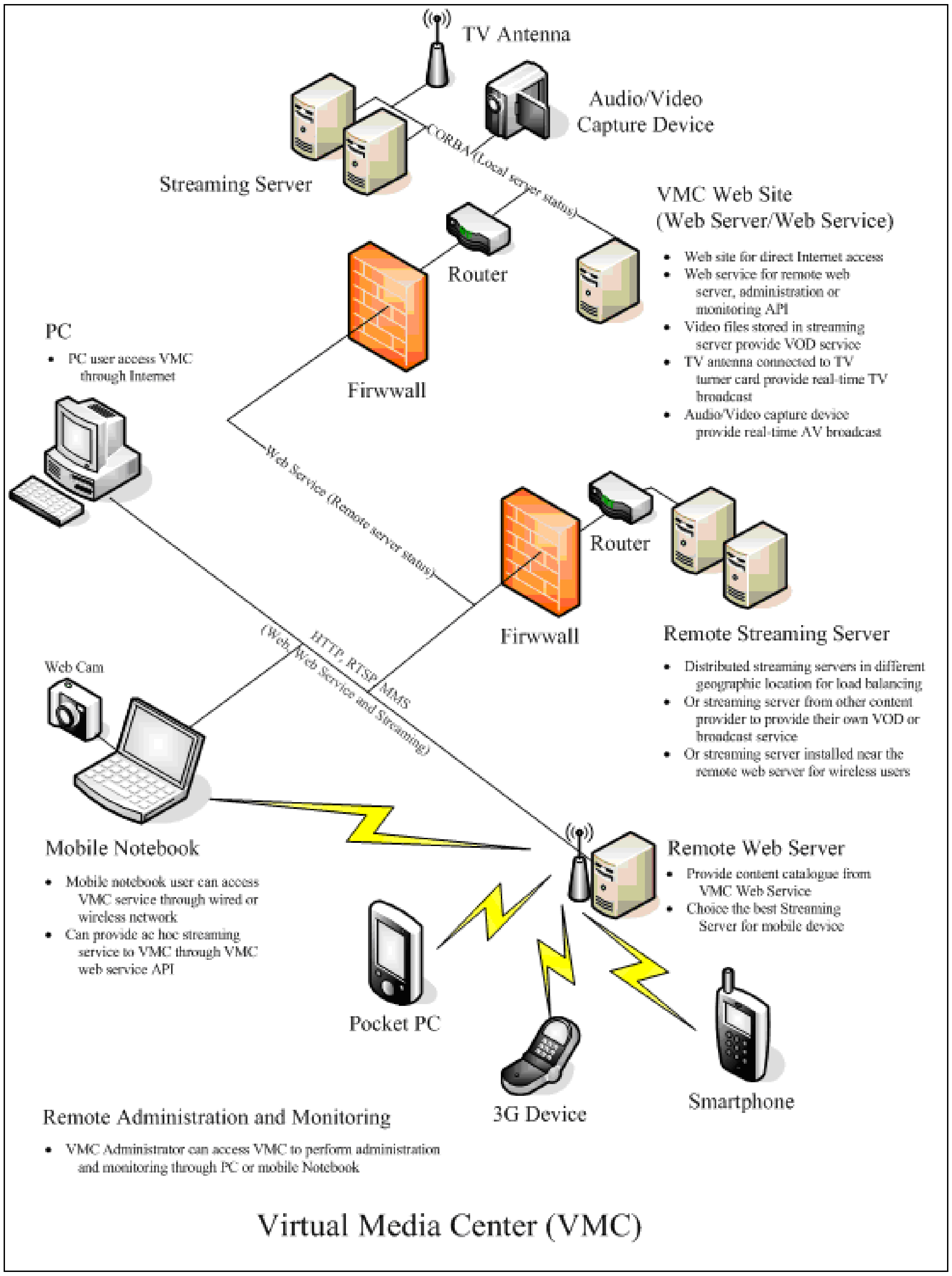

TV Antenna
Audio/Video Capture Device
Streaming Server
CORBA (Local server status)
VMC Web Site (Web Server/Web Service)
• Web site for direct Internet access
• Web service for remote web server, administration or monitoring API
• Video files stored in streaming server provide VOD service
• TV antenna connected to TV turner card provide real-time TV broadcast
• Audio/Video capture device provide real-time AV broadcast
Router
Firwwall
PC
• PC user access VMC through Internet
Web Service (Remote server status)
Router
Firwwall
Remote Streaming Server
• Distributed streaming servers in different geographic location for load balancing
• Or streaming server from other content provider to provide their own VOD or broadcast service
• Or streaming server installed near the remote web server for wireless users
HTTP, RTSP, MMS
(Web, Web Service and Streaming)
Web Cam
Mobile Notebook
• Mobile notebook user can access VMC service through wired or wireless network
• Can provide ac hoc streaming service to VMC through VMC web service API
Remote Web Server
• Provide content catalogue from VMC Web Service
• Choice the best Streaming Server for mobile device
Pocket PC
3G Device
Smartphone
Remote Administration and Monitoring
• VMC Administrator can access VMC to perform administration and monitoring through PC or mobile Notebook
Virtual Media Center (VMC)

Figure 4.1.1 A Generic Implementation Architecture

The implementation corresponding to the logical component architecture is illustrated as below.

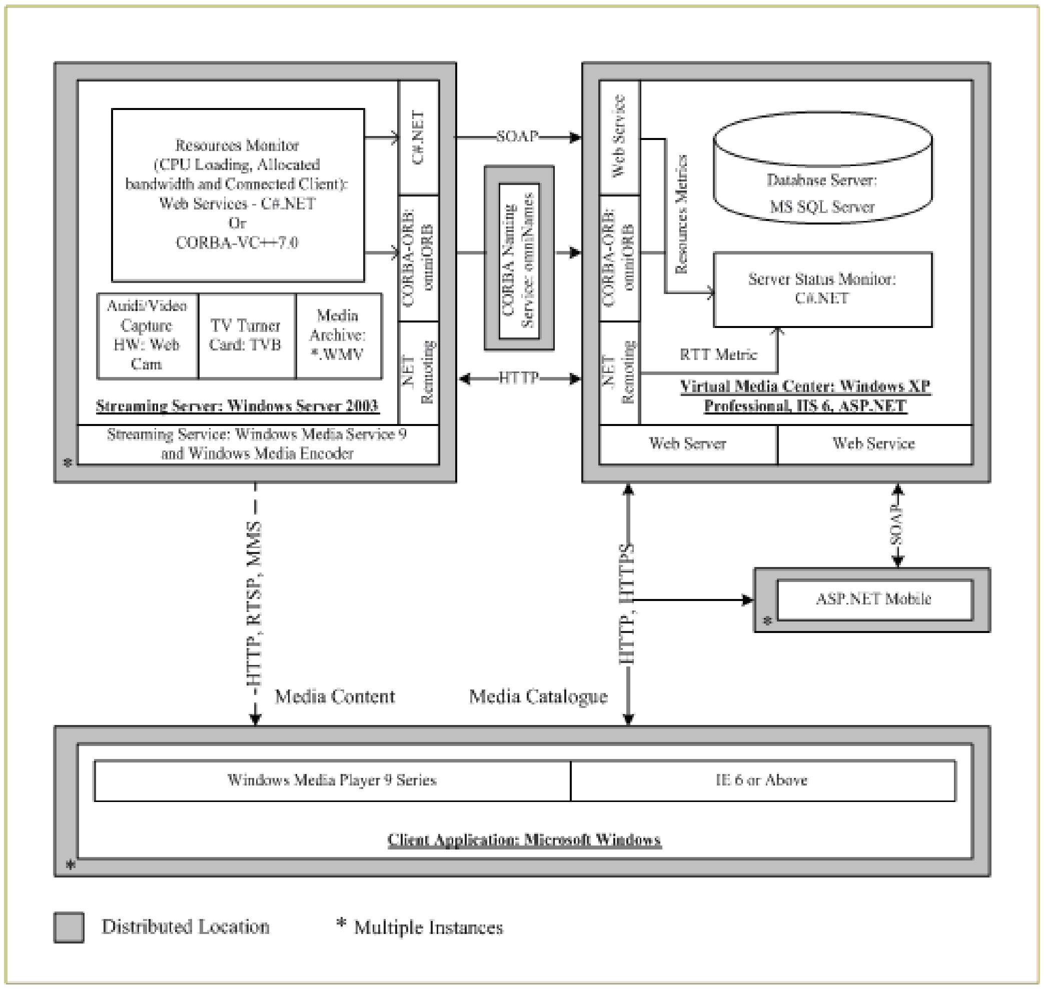

Resources Monitor (CPU Loading, Allocated bandwidth and Connected Client): Web Services - C#.NET Or CORBA-VC++7.0
C#.NET
CORBA-ORB: omniORB
.NET Remoting
Auidi/Video Capture HW: Web Cam
TV Turner Card: TVB
Media Archive: *.WMV
Streaming Server: Windows Server 2003
Streaming Service: Windows Media Service 9 and Windows Media Encoder
SOAP
HTTP
CORBA Naming Service: omniNames
Web Service
CORBA-ORB: omniORB
.NET Remoting
Resources Metrics
Database Server: MS SQL Server
Server Status Monitor: C#.NET
RTT Metric
Virtual Media Center: Windows XP Professional, IIS 6, ASP.NET
Web Server
Web Service
SOAP
ASP.NET Mobile
HTTP, RTSP, MMS
HTTP, HTTPS
Media Content
Media Catalogue
Windows Media Player 9 Series
IE 6 or Above
Client Application: Microsoft Windows
Distributed Location
* Multiple Instances

Figure 4.1.2 Distributed Component Implementation

4.2. Development Tools

The implementation of VMC is mainly target on Microsoft Windows system for both back end and front end. The development tools available in Microsoft Windows are many. The choosing criteria are depending on many factors, such as development time, cost of ownership, learning curve and maintenance cost etc. All of these are not the major issues here. As the design of the generic architecture framework is allow to chose many different implementation and work together by CORAB and Web service as the interface for distributed components.

CORBA ORB can be interoperated by different products. For simplicity, all ORB here is developed by a free product called omniORB and the CORBA naming service is choosing the same series called omniNames. As omniORB is support C++. Both CORBA Server object and client object use C++ to develop. All the others development tools are major in Microsoft Visual Studio.NET. The only thing need to mention here are both Visual Studio.NET 2002 and Visual Studio.NET 2003 are used. The reason is that the current version of omniORB (version 4.0.3) is supported to VC++7.0 only and VC++7.0 is in Visual Studio.NET 2002. And other components such as .NET Remoting, Web Service and ASP.NET are developed in Visual Studio.NET 2003. The following table summarized the development tools details.

VMC Server (Web Server and Web Service Interface)	
Web Server	Windows Server 2003 Internet Information Server 6 ASP.NET 1.1
Web Service	Visual Studio .NET 2003 C# Internet Information Server 6
VMC Monitor (RTT Collection)	C#.NET .NET Remoting
Database Server	SQL server 2000 MSDE version
CORBA Naming Service	omniORB Naming Service
CORBA Server Object	omniORB ORB Visual Studio .NET 2003 VC++7.0
Streaming Server	
VOD Service	Windows Server 2003 Microsoft Windows Media Server 9 HTTP, MMS and RTSP Protocols Windows Media File format (*.wmv)
Real-time broadcast for TV	Windows XP Professional Windows Media Encoder TV Turner Card
Real-time broadcast for Web Cam	Windows XP Professional Windows Media Encoder Web Cam
RTT Server	Visual Studio .NET 2003 C# .NET Remoting

	Windows Service
Resource Monitor – CORBA Client	omniORB ORB Visual Studio .NET 2003 VC++7.0
Resource Monitor – Web Service Client	Visual Studio .NET 2003 C#.NET
Remote Web Server	
Web Server	Windows Server 2003 Internet Information Server 6 ASP.NET 1.1 Mobile (For Pocket PC)

Client Requirement	
PC	Microsoft Windows Windows Media Player 9 Internet Explorer 6 or Above
Pocket PC	Microsoft Pocket PC 2003 Windows Media Player 9

Development Tools	
PC	Internet Information Server 6 Visual Studio .NET 2002 VC++7.0 (For omniORB) Visual Studio .NET 2003 C# ASP.NET 1.1 .NET Framework 1.1
Pocket PC	ASP.NET 1.1 Mobile .NET compact framework 1.1 Pocket PC 2003 Emulator Pocket PC 2003 Phone Edition Emulator

4.3. Database Structure

The database structure in SQL server is illustrated in the following diagram. The actual field dimension is not important here as they are depending on application different requirement and implementation.

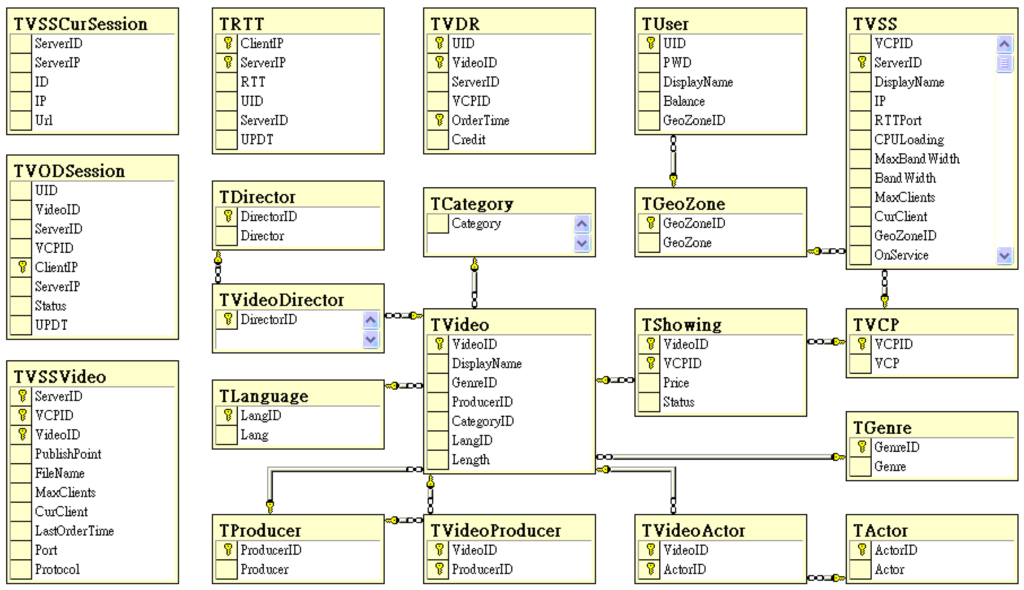


Figure 4.3 SQL Server Database Structure Diagram

These tables is designed to provide basic data architecture for support user profile, video catalogue, streaming server configuration, server resources metrics, current session monitor and media retrieval log. But it is not refined to optimal database structure design. It may be different in different application or implementation.

4.4. Database Script

The T-SQL used to create all the basic tables in SQL server is listed as follows.

```
CREATE TABLE [dbo].[TActor] (
     [ActorID] [int] NOT NULL ,
     [Actor] [nvarchar] (100) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TCategory] (
     [CategoryID] [int] NOT NULL ,
     [Category] [nvarchar] (5) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TDirector] (
     [DirectorID] [int] NOT NULL ,
     [Director] [nvarchar] (100) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TGenre] (
     [GenreID] [int] NOT NULL ,
     [Genre] [nvarchar] (50) NULL
```

```
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TGeoZone] (
     [GeoZoneID] [int] NOT NULL ,
     [GeoZone] [nvarchar] (20) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TLanguage] (
     [LangID] [int] NOT NULL ,
     [Lang] [nvarchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TProducer] (
     [ProducerID] [int] NOT NULL ,
     [Producer] [nvarchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TRTT] (
     [ClientIP] [varchar] (15) NOT NULL ,
     [ServerIP] [varchar] (15) NOT NULL ,
     [RTT] [int] NOT NULL ,
     [UID] [nvarchar] (20) NULL ,
     [ServerID] [int] NULL ,
     [UPDT] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TShowing] (
     [VideoID] [int] NOT NULL ,
     [VCPID] [int] NOT NULL ,
     [Price] [int] NULL ,
     [Status] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TUser] (
     [UID] [nvarchar] (20) NOT NULL ,
     [PWD] [nvarchar] (20) NULL ,
     [DisplayName] [nvarchar] (30) NULL ,
     [Balance] [bigint] NULL ,
     [GeoZoneID] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVCP] (
     [VCPID] [int] NOT NULL ,
     [VCP] [nvarchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVDR] (
     [UID] [nvarchar] (20) NOT NULL ,
     [VideoID] [int] NOT NULL ,
     [ServerID] [int] NULL ,
     [VCPID] [int] NULL ,
     [OrderTime] [datetime] NOT NULL ,
     [Credit] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVODSession] (
     [UID] [nvarchar] (20) NULL ,
     [VideoID] [int] NULL ,
     [ServerID] [int] NULL ,
     [VCPID] [int] NULL ,
     [ClientIP] [varchar] (15) NOT NULL ,
     [ServerIP] [varchar] (15) NOT NULL ,
     [Status] [int] NULL ,
     [UPDT] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVSS] (
     [VCPID] [int] NOT NULL ,
     [ServerID] [int] NOT NULL ,
     [DisplayName] [nvarchar] (30) NULL ,
     [IP] [varchar] (15) NULL ,
     [RTTPort] [int] NULL ,
     [CPULoading] [smallint] NULL ,
     [MaxBandWidth] [int] NULL ,
     [BandWidth] [int] NULL ,
     [MaxClients] [int] NULL ,
     [CurClient] [int] NULL ,
     [GeoZoneID] [int] NULL ,
     [OnService] [bit] NULL ,
     [AutoDetect] [bit] NULL ,
     [PrivateIP] [varchar] (15) NULL ,
     [UPDT] [datetime] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVSSCurSession] (
     [ServerID] [int] NOT NULL ,
     [ServerIP] [varchar] (15) NOT NULL ,
     [ID] [int] NULL ,
     [IP] [varchar] (15) NULL ,
     [Url] [varchar] (100) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVSSVideo] (
     [ServerID] [int] NOT NULL ,
     [VCPID] [int] NOT NULL ,
     [VideoID] [int] NOT NULL ,
     [PublishPoint] [varchar] (50) NOT NULL ,
     [FileName] [varchar] (50) NULL ,
     [MaxClients] [int] NULL ,
     [CurClient] [int] NULL ,
     [LastOrderTime] [datetime] NULL ,
     [Port] [int] NULL ,
     [Protocol] [varchar] (10) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVideo] (
     [VideoID] [int] NOT NULL ,
     [DisplayName] [nvarchar] (100) NULL ,
     [GenreID] [int] NULL ,
     [ProducerID] [int] NULL ,
     [CategoryID] [int] NULL ,
     [LangID] [int] NULL ,
     [Length] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVideoActor] (
     [VideoID] [int] NOT NULL ,
     [ActorID] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TVideoDirector] (
     [VideoID] [int] NOT NULL ,
     [DirectorID] [int] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[TVideoProducer] (
     [VideoID] [int] NOT NULL ,
     [ProducerID] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[TActor] ADD
     CONSTRAINT [PK_TActor] PRIMARY KEY  CLUSTERED
     (
          [ActorID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TCategory] ADD
     CONSTRAINT [PK_TCategory] PRIMARY KEY  CLUSTERED
     (
          [CategoryID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TDirector] ADD
     CONSTRAINT [PK_TDirector] PRIMARY KEY  CLUSTERED
     (
          [DirectorID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TGenre] ADD
     CONSTRAINT [PK_TGenre] PRIMARY KEY  CLUSTERED
     (
          [GenreID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TGeoZone] ADD
     CONSTRAINT [PK_TGeoZone] PRIMARY KEY  CLUSTERED
     (
          [GeoZoneID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TLanguage] ADD
     CONSTRAINT [PK_TLanguage] PRIMARY KEY  CLUSTERED
     (
          [LangID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TProducer] ADD
     CONSTRAINT [PK_TProducer] PRIMARY KEY  CLUSTERED
     (
          [ProducerID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TRTT] ADD
     CONSTRAINT [DF_TRTT_RTT] DEFAULT (0) FOR [RTT],
     CONSTRAINT [DF_TRTT_UPDT] DEFAULT (getdate()) FOR [UPDT],
     CONSTRAINT [PK_TRTT] PRIMARY KEY  CLUSTERED
     (
          [ClientIP],
          [ServerIP]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TShowing] ADD
     CONSTRAINT [DF_TShowing_Status] DEFAULT (0) FOR [Status],
     CONSTRAINT [PK_TPrice] PRIMARY KEY  CLUSTERED
     (
          [VideoID],
          [VCPID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TUser] ADD
     CONSTRAINT [PK_TUser] PRIMARY KEY  CLUSTERED
     (
          [UID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVCP] ADD
     CONSTRAINT [PK_TVCP] PRIMARY KEY  CLUSTERED
     (
          [VCPID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVDR] ADD
     CONSTRAINT [DF_TVDR_OrderTime] DEFAULT (getdate()) FOR [OrderTime],
     CONSTRAINT [PK_TVDR] PRIMARY KEY  CLUSTERED
     (
          [UID],
          [VideoID],
          [OrderTime]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVODSession] ADD
     CONSTRAINT [DF_TVODSession_Status] DEFAULT (0) FOR [Status],
     CONSTRAINT [DF_TVODSession_UPDT] DEFAULT (getdate()) FOR [UPDT],
     CONSTRAINT [PK_TVODSession] PRIMARY KEY  CLUSTERED
     (
          [ClientIP]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVSS] ADD
     CONSTRAINT [DF_TVSS_RTT] DEFAULT (2909) FOR [RTTPort],
     CONSTRAINT [DF_TVSS_CPULoading] DEFAULT (0) FOR [CPULoading],
     CONSTRAINT [DF_TVSS_MaxBandWidth] DEFAULT (1000) FOR [MaxBandWidth],
     CONSTRAINT [DF_TVSS_BandWidth] DEFAULT (0) FOR [BandWidth],
     CONSTRAINT [DF_TVSS_OnService] DEFAULT (0) FOR [OnService],
     CONSTRAINT [DF_TVSS_AutoDetect] DEFAULT (1) FOR [AutoDetect],
     CONSTRAINT [DF_TVSS_PrivateIP] DEFAULT ('') FOR [PrivateIP],
     CONSTRAINT [DF_TVSS_UPDT] DEFAULT (getdate()) FOR [UPDT],
     CONSTRAINT [PK_TVSS] PRIMARY KEY  CLUSTERED
     (
          [ServerID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVSSVideo] ADD
     CONSTRAINT [DF_TVSSVideo_LastOrderTime] DEFAULT (getdate()) FOR [LastOrderTime],
     CONSTRAINT [DF_TVSSVideo_Port] DEFAULT (8080) FOR [Port],
     CONSTRAINT [DF_TVSSVideo_Protocol] DEFAULT ('Default') FOR [Protocol],
     CONSTRAINT [PK_TVSSVideo] PRIMARY KEY  CLUSTERED
     (
          [ServerID],
          [VCPID],
          [VideoID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVideo] ADD
     CONSTRAINT [PK_TVideo] PRIMARY KEY  CLUSTERED
     (
          [VideoID]
     )  ON [PRIMARY]
```

```
GO

ALTER TABLE [dbo].[TVideoActor] ADD
     CONSTRAINT [PK_TVideoActor] PRIMARY KEY  CLUSTERED
     (
          [VideoID],
          [ActorID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVideoDirector] ADD
     CONSTRAINT [PK_TVideoDirector] PRIMARY KEY  CLUSTERED
     (
          [VideoID],
          [DirectorID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TVideoProducer] ADD
     CONSTRAINT [PK_TVideoProductor] PRIMARY KEY  CLUSTERED
     (
          [VideoID],
          [ProducerID]
     )  ON [PRIMARY]
GO

ALTER TABLE [dbo].[TShowing] ADD
     CONSTRAINT [FK_TPrice_TVCP] FOREIGN KEY
     (
          [VCPID]
     ) REFERENCES [dbo].[TVCP] (
          [VCPID]
     ),
     CONSTRAINT [FK_TPrice_TVideo] FOREIGN KEY
     (
          [VideoID]
     ) REFERENCES [dbo].[TVideo] (
          [VideoID]
     )
GO

ALTER TABLE [dbo].[TUser] ADD
     CONSTRAINT [FK_TUser_TGeoZone] FOREIGN KEY
     (
          [GeoZoneID]
     ) REFERENCES [dbo].[TGeoZone] (
          [GeoZoneID]
     )
GO

ALTER TABLE [dbo].[TVSS] ADD
     CONSTRAINT [FK_TVSS_TGeoZone] FOREIGN KEY
     (
          [GeoZoneID]
     ) REFERENCES [dbo].[TGeoZone] (
          [GeoZoneID]
     ),
     CONSTRAINT [FK_TVSS_TVCP] FOREIGN KEY
     (
          [VCPID]
     ) REFERENCES [dbo].[TVCP] (
          [VCPID]
     )
GO

ALTER TABLE [dbo].[TVideo] ADD
     CONSTRAINT [FK_TVideo_TCategory] FOREIGN KEY
     (
          [CategoryID]
     ) REFERENCES [dbo].[TCategory] (
          [CategoryID]
     ),
     CONSTRAINT [FK_TVideo_TGenre] FOREIGN KEY
     (
          [GenreID]
     ) REFERENCES [dbo].[TGenre] (
          [GenreID]
     ),
     CONSTRAINT [FK_TVideo_TLanguage] FOREIGN KEY
     (
          [LangID]
     ) REFERENCES [dbo].[TLanguage] (
          [LangID]
     ),
     CONSTRAINT [FK_TVideo_TProducer] FOREIGN KEY
     (
          [ProducerID]
     ) REFERENCES [dbo].[TProducer] (
          [ProducerID]
     )
GO

ALTER TABLE [dbo].[TVideoActor] ADD
     CONSTRAINT [FK_TVideoActor_TActor] FOREIGN KEY
     (
          [ActorID]
     ) REFERENCES [dbo].[TActor] (
          [ActorID]
     ),
     CONSTRAINT [FK_TVideoActor_TVideo] FOREIGN KEY
     (
          [VideoID]
     ) REFERENCES [dbo].[TVideo] (
          [VideoID]
     )
GO

ALTER TABLE [dbo].[TVideoDirector] ADD
     CONSTRAINT [FK_TVideoDirector_TDirector] FOREIGN KEY
     (
          [DirectorID]
     ) REFERENCES [dbo].[TDirector] (
          [DirectorID]
     ),
     CONSTRAINT [FK_TVideoDirector_TVideo] FOREIGN KEY
     (
          [VideoID]
     ) REFERENCES [dbo].[TVideo] (
          [VideoID]
     )
GO

ALTER TABLE [dbo].[TVideoProducer] ADD
     CONSTRAINT [FK_TVideoProductor_TProducer] FOREIGN KEY
     (
          [ProducerID]
     ) REFERENCES [dbo].[TProducer] (
          [ProducerID]
     ),
     CONSTRAINT [FK_TVideoProductor_TVideo] FOREIGN KEY
     (
          [VideoID]
     ) REFERENCES [dbo].[TVideo] (
          [VideoID]
     )
GO
```

Figure 4.4 T-SQL Table Initialization Script

4.5. CORBA ORB

It is logically free to use any CORBA ORB product for both the server and client sides. One common places to find CORBA ORB product is from the Object Management Group's (OMG) Web site (http://www.omg.org).

CORBA product can be classified in to pure commercial products, commercial with free trial period and free of charge shareware. omniORB is one of the free of charge ORB that provide light weight and high performance and binding to C++. CORBA Naming and Event service are also providing by omnoORB. Commercial support is available by Duncan Grisby's company. The CORBA Event service was written by Paul Nader and can be downloaded from omnievents.sourceforge.net. The omniORB version used in this implementation is 4.0.3. It is supported for Unix/Linux/Windows, provided binding to both Java and C++. The CORBA Event service used in this implementation is also chosen omniNames. For details of omniORB and omniNames setup, please refer to appendix at the end of this document.

4.6. CORBA IDL

CORBA IDL is independent to any CORBA ORB product. The IDL used in omniORB can be re-compile under another CORBA ORB product and generate a corresenting set of server skeleton and client stub.

CORBA IDL provides the definition of the VMC server objects. This definition then can be used to generate client stubs and server skeletons in target programming language for remote invocation and implementation. The IDL for the VMC is listed as follows.

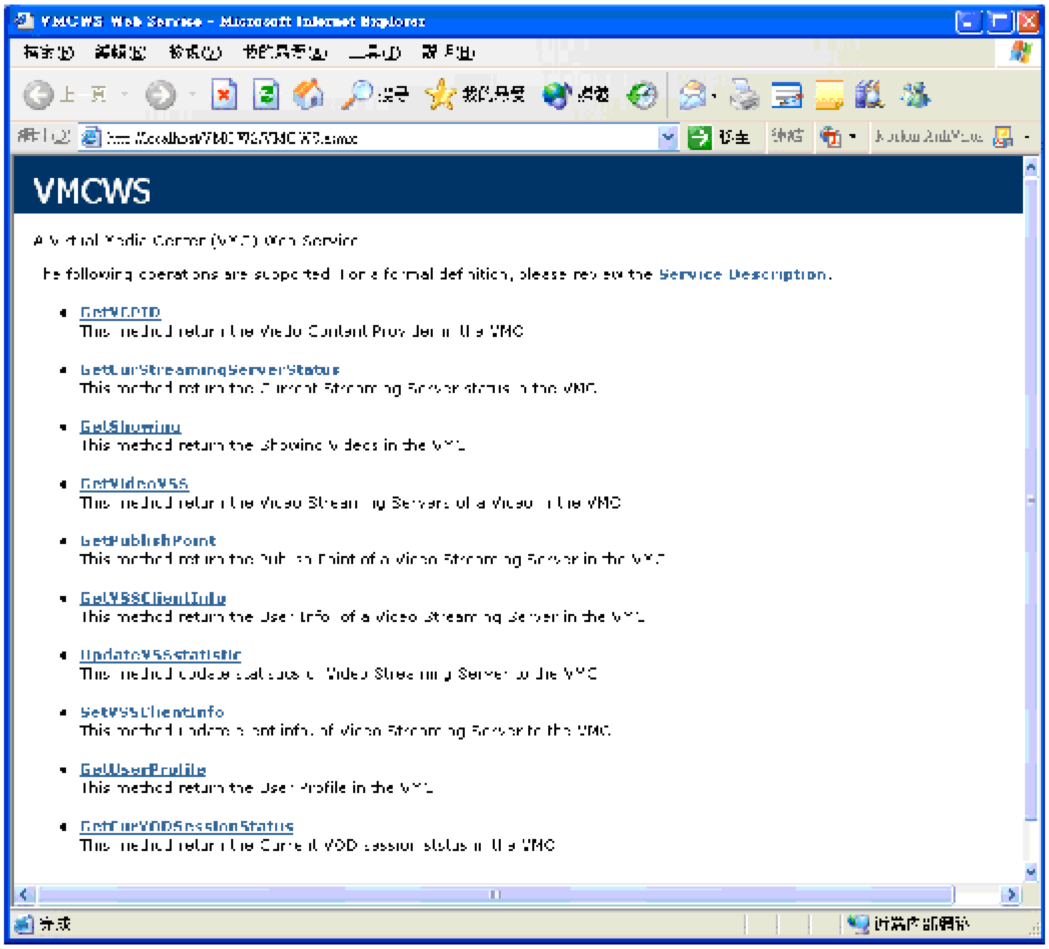
```
/*
The Chinese University of Hong Kong
Department of Computer Science & Engineering
PTMSC of COMPUTER SCIENCE (2002-2004)
Project:  Distributed Video-on-Demand (DVOD)
Supervisor:    Prof. Michael R. Lyu
Student:  Lee Ka Yan Sunny
StudentID:     02240550
Application:   Virtual Media Center (VMC)
*/

module DVODVMC {
     // Virtual Media Center (VMC)
     interface VMC {

          readonly attribute string     Name;
          readonly attribute string     HostName;
          readonly attribute string     IP;
          readonly attribute long  VCPCount;
          readonly attribute long  VSSCount;
          readonly attribute long  VideoCount;

          boolean AddVCP(in long VCPID, in string Name);

          boolean RemoveVCP(in long VCPID);

     };

     // Video Content Provider (VCP)
     interface VCP {

          boolean AddVSS(in long VCPID, in long ServerID, in string HostName, in string
IP, in string DisplayName, in long RTTPort, in long MaxBandWidth, in long MaxClient,
in long GeoZoneID);

          boolean RemoveVSS(in long VCPID, in long ServerID);

          boolean ShowingOn(in long VCPID, in long VideoID, in long Price, in long
Status);

          boolean ShowingOff(in long VCPID, in long VideoID);

     };

     // Video Streaming Server (VSS)
     interface VSS {

          boolean ServiceOn(in long ServerID);

          boolean ServiceOff(in long ServerID);

          boolean AddVideo(in long ServerID, in long VideoID, in string Title, in string
PublishPoint, in string FileName);

          boolean RemoveVideo(in long ServerID, in long VideoID, in string Title);

          boolean UpdateStatistic(in long ServerID, in long CPULoading, in long
AllocatedBandWidth, in long CurClients);

          boolean IsValidClient(in long ServerID, in string ClientIP);

          boolean UpdateClientInfo(in long ServerID, in string ClientIP, in long ID,
in string VODURL);

     };

};
```

Figure 4.6 CORBA IDL for VMC Server Objects

There are total three server objects defined. They are VMC object, VCP object and VSS object respectively. VMC provided function to add or remove a content provider. VCP provided functions to add or remove streaming server managed by a content provider and enable or disable the availability of media content. VSS provided functions to turn the streaming server on or off, add or remove a media content and update of server resources metrics (CPU utilization, allocated bandwidth and connected client). In this implementation, only the resources update function of the VSS server object is used.

4.7. Web Service

The operations provided in the VMC Web service is to providing remote Web server to obtain necessary information to provide it's local subscribers. Web service operations are basically a subset of the direct database access operation in the Web server. The name of the Web service in the implementation is called VMCWS. The VMCWS can be tested with the Internet Explorer 6 by directly access the Web service URL. The following screen capture shows the Web page which used to access the VMCWS directly.


VMCWS Web Service - Microsoft Internet Explorer
http://localhost/VMCWS/VMCWS.asmx

VMCWS

A Virtual Media Center (VMC) Web Service

The following operations are supported. For a formal definition, please review the Service Description.

- GetVCPID
  This method return the Video Content Provider in the VMC
- GetCurStreamingServerStatus
  This method return the Current Streaming Server status in the VMC
- GetShowing
  This method return the Showing Videos in the VMC
- GetVideoVSS
  This method return the Video Streaming Servers of a Video in the VMC
- GetPublishPoint
  This method return the Publish Point of a Video Streaming Server in the VMC
- GetVSSClientInfo
  This method return the User Info of a Video Streaming Server in the VMC
- UpdateVSSstatistic
  This method update statistic of Video Streaming Server to the VMC
- SetVSSClientInfo
  This method update client info. of Video Streaming Server to the VMC
- GetUserProfile
  This method return the User Profile in the VMC
- GetCurVODSessionStatus
  This method return the Current VOD session status in the VMC

Figure 4.7 VMC Web Service

The Web service illustrated in figure 4.7 is only for demonstration purpose only. In real application, it must include authentication consideration instead of anyone can free to access the server information and retrieve media contents.

4.8. Web Service Description Language (WDSL)

The Web Service Definition Language (WSDL) is used by dynamic service discovery from other parties. The target clients are other distributed video-on-demand system or remote Web server who want to being a agent to provide the video catalogue from current VMC. The WDSL of the VMCWS is listed as the following.

```
  <?xml version="1.0" encoding="utf-8" ?>
- <definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s0="http://dvod.homeip.net/VMCWS/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://dvod.homeip.net/VMCWS/"
xmlns="http://schemas.xmlsoap.org/wsdl/">
- <types>
- <s:schema elementFormDefault="qualified"
targetNamespace="http://dvod.homeip.net/VMCWS/">
  <s:import namespace="http://www.w3.org/2001/XMLSchema" />
- <s:element name="GetUserProfile">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="0" maxOccurs="1" name="UID" type="s:string" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetUserProfileResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetUserProfileResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetVCPID">
  <s:complexType />
  </s:element>
- <s:element name="GetVCPIDResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetVCPIDResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetShowing">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="VCPID" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="VideoID" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="Price" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="Status" type="s:int" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetShowingResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetShowingResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetVideoVSS">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="VCPID" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="VideoID" type="s:int" />
  <s:element minOccurs="0" maxOccurs="1" name="ClientIP" type="s:string" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetVideoVSSResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetVideoVSSResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetPublishPoint">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="ServerID" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="VideoID" type="s:int" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetPublishPointResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetPublishPointResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="UpdateVSSstatistic">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="ServerID" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="CPULoading" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="AllocatedBandWidth" type="s:int" />
  <s:element minOccurs="1" maxOccurs="1" name="CurClient" type="s:int" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="UpdateVSSstatisticResponse">
  <s:complexType />
  </s:element>
- <s:element name="SetVSSClientInfo">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="ServerID" type="s:int" />
  <s:element minOccurs="0" maxOccurs="1" name="ServerIP" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="ID" type="s:int" />
  <s:element minOccurs="0" maxOccurs="1" name="IP" type="s:string" />
  <s:element minOccurs="0" maxOccurs="1" name="Url" type="s:string" />
  <s:element minOccurs="1" maxOccurs="1" name="Clear" type="s:int" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="SetVSSClientInfoResponse">
  <s:complexType />
  </s:element>
- <s:element name="GetVSSClientInfo">
- <s:complexType>
- <s:sequence>
  <s:element minOccurs="1" maxOccurs="1" name="ServerID" type="s:int" />
  <s:element minOccurs="0" maxOccurs="1" name="ServerIP" type="s:string" />
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetVSSClientInfoResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetVSSClientInfoResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetCurVODSessionStatus">
  <s:complexType />
  </s:element>
- <s:element name="GetCurVODSessionStatusResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetCurVODSessionStatusResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
- <s:element name="GetCurStreamingServerStatus">
  <s:complexType />
  </s:element>
- <s:element name="GetCurStreamingServerStatusResponse">
- <s:complexType>
- <s:sequence>
- <s:element minOccurs="0" maxOccurs="1" name="GetCurStreamingServerStatusResult">
- <s:complexType>
- <s:sequence>
  <s:element ref="s:schema" />
  <s:any />
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:sequence>
  </s:complexType>
  </s:element>
  </s:schema>
  </types>
- <message name="GetUserProfileSoapIn">
  <part name="parameters" element="s0:GetUserProfile" />
  </message>
- <message name="GetUserProfileSoapOut">
  <part name="parameters" element="s0:GetUserProfileResponse" />
  </message>
- <message name="GetVCPIDSoapIn">
  <part name="parameters" element="s0:GetVCPID" />
  </message>
- <message name="GetVCPIDSoapOut">
  <part name="parameters" element="s0:GetVCPIDResponse" />
  </message>
- <message name="GetShowingSoapIn">
  <part name="parameters" element="s0:GetShowing" />
  </message>
- <message name="GetShowingSoapOut">
  <part name="parameters" element="s0:GetShowingResponse" />
  </message>
- <message name="GetVideoVSSSoapIn">
  <part name="parameters" element="s0:GetVideoVSS" />
  </message>
- <message name="GetVideoVSSSoapOut">
  <part name="parameters" element="s0:GetVideoVSSResponse" />
  </message>
- <message name="GetPublishPointSoapIn">
  <part name="parameters" element="s0:GetPublishPoint" />
  </message>
- <message name="GetPublishPointSoapOut">
  <part name="parameters" element="s0:GetPublishPointResponse" />
  </message>
- <message name="UpdateVSSstatisticSoapIn">
  <part name="parameters" element="s0:UpdateVSSstatistic" />
  </message>
- <message name="UpdateVSSstatisticSoapOut">
  <part name="parameters" element="s0:UpdateVSSstatisticResponse" />
  </message>
- <message name="SetVSSClientInfoSoapIn">
  <part name="parameters" element="s0:SetVSSClientInfo" />
  </message>
- <message name="SetVSSClientInfoSoapOut">
  <part name="parameters" element="s0:SetVSSClientInfoResponse" />
  </message>
- <message name="GetVSSClientInfoSoapIn">
  <part name="parameters" element="s0:GetVSSClientInfo" />
  </message>
- <message name="GetVSSClientInfoSoapOut">
  <part name="parameters" element="s0:GetVSSClientInfoResponse" />
  </message>
- <message name="GetCurVODSessionStatusSoapIn">
```

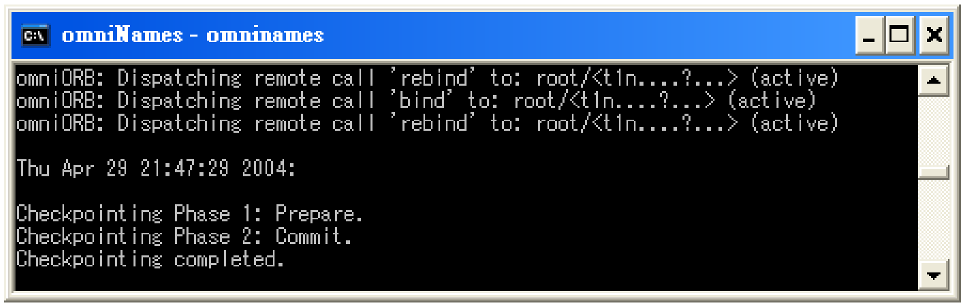
```
  <part name="parameters" element="s0:GetCurVODSessionStatus" />
  </message>
- <message name="GetCurVODSessionStatusSoapOut">
  <part name="parameters" element="s0:GetCurVODSessionStatusResponse" />
  </message>
- <message name="GetCurStreamingServerStatusSoapIn">
  <part name="parameters" element="s0:GetCurStreamingServerStatus" />
  </message>
- <message name="GetCurStreamingServerStatusSoapOut">
  <part name="parameters" element="s0:GetCurStreamingServerStatusResponse" />
  </message>
- <portType name="VMCWSSoap">
- <operation name="GetUserProfile">
  <documentation>This method return the User Profile in the VMC</documentation>
  <input message="s0:GetUserProfileSoapIn" />
  <output message="s0:GetUserProfileSoapOut" />
  </operation>
- <operation name="GetVCPID">
  <documentation>This method return the Viedo Content Provider in the VMC</documentation>
  <input message="s0:GetVCPIDSoapIn" />
  <output message="s0:GetVCPIDSoapOut" />
  </operation>
- <operation name="GetShowing">
  <documentation>This method return the Showing Videos in the VMC</documentation>
  <input message="s0:GetShowingSoapIn" />
  <output message="s0:GetShowingSoapOut" />
  </operation>
- <operation name="GetVideoVSS">
  <documentation>This method return the Video Streaming Servers of a Video in the
VMC</documentation>
  <input message="s0:GetVideoVSSSoapIn" />
  <output message="s0:GetVideoVSSSoapOut" />
  </operation>
- <operation name="GetPublishPoint">
  <documentation>This method return the Publish Point of a Video Streaming Server in the
VMC</documentation>
  <input message="s0:GetPublishPointSoapIn" />
  <output message="s0:GetPublishPointSoapOut" />
  </operation>
- <operation name="UpdateVSSstatistic">
  <documentation>This method update statistics of Video Streaming Server to the
VMC</documentation>
  <input message="s0:UpdateVSSstatisticSoapIn" />
  <output message="s0:UpdateVSSstatisticSoapOut" />
  </operation>
- <operation name="SetVSSClientInfo">
  <documentation>This method update client info. of Video Streaming Server to the
VMC</documentation>
  <input message="s0:SetVSSClientInfoSoapIn" />
  <output message="s0:SetVSSClientInfoSoapOut" />
  </operation>
- <operation name="GetVSSClientInfo">
  <documentation>This method return the User Info. of a Video Streaming Server in the
VMC</documentation>
  <input message="s0:GetVSSClientInfoSoapIn" />
  <output message="s0:GetVSSClientInfoSoapOut" />
  </operation>
- <operation name="GetCurVODSessionStatus">
  <documentation>This method return the Current VOD session ststus in the
VMC</documentation>
  <input message="s0:GetCurVODSessionStatusSoapIn" />
  <output message="s0:GetCurVODSessionStatusSoapOut" />
  </operation>
- <operation name="GetCurStreamingServerStatus">
  <documentation>This method return the Current Streaming Server status in the
VMC</documentation>
  <input message="s0:GetCurStreamingServerStatusSoapIn" />
  <output message="s0:GetCurStreamingServerStatusSoapOut" />
  </operation>
  </portType>
```

```
- <binding name="VMCWSSoap" type="s0:VMCWSSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
- <operation name="GetUserProfile">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetUserProfile"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetVCPID">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetVCPID" style="document"
/>
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetShowing">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetShowing"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetVideoVSS">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetVideoVSS"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetPublishPoint">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetPublishPoint"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="UpdateVSSstatistic">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/UpdateVSSstatistic"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="SetVSSClientInfo">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/SetVSSClientInfo"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetVSSClientInfo">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetVSSClientInfo"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetCurVODSessionStatus">
  <soap:operation soapAction="http://dvod.homeip.net/VMCWS/GetCurVODSessionStatus"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
- <operation name="GetCurStreamingServerStatus">
  <soap:operation
soapAction="http://dvod.homeip.net/VMCWS/GetCurStreamingServerStatus"
style="document" />
- <input>
  <soap:body use="literal" />
  </input>
- <output>
  <soap:body use="literal" />
  </output>
  </operation>
  </binding>
- <service name="VMCWS">
  <documentation>A Virtual Media Center (VMC) Web Service</documentation>
- <port name="VMCWSSoap" binding="s0:VMCWSSoap">
  <soap:address location="http://localhost/VMCWS/VMCWS.asmx" />
  </port>
  </service>
  </definitions>
```

Figure 4.8 Web Service WSDL Content

4.9. Programs in VMC Server

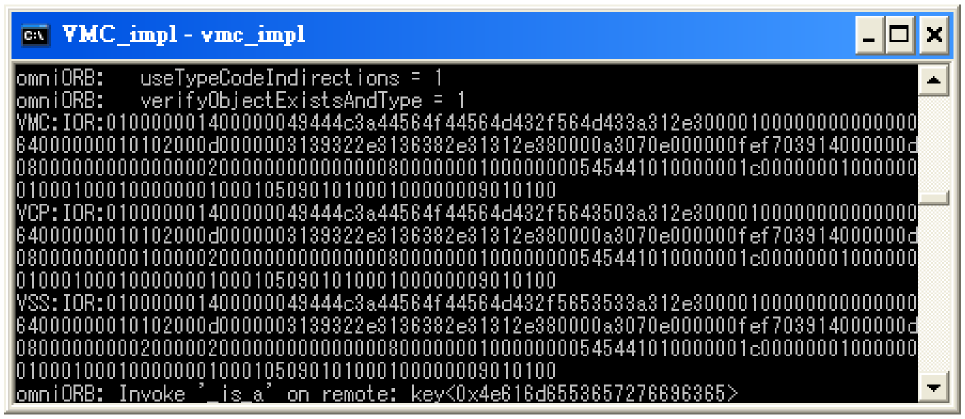
omniNames.exe is a application program execute in command line. It is used to provide CORBA naming service to the VMC. This program is a included from omniORB package.



```
omniORB: Dispatching remote call 'rebind' to: root/<t1n....?...> (active)
omniORB: Dispatching remote call 'bind' to: root/<t1n....?...> (active)
omniORB: Dispatching remote call 'rebind' to: root/<t1n....?...> (active)

Thu Apr 29 21:47:29 2004:

Checkpointing Phase 1: Prepare.
Checkpointing Phase 2: Commit.
Checkpointing completed.
```

Figure 4.9.1 **omniNames.exe** is running

VMC_impl.exe is the CORBA server object for updating of streaming server resources metrics. This program is developed by Visual Studio.NET 2002 VC++7.0.

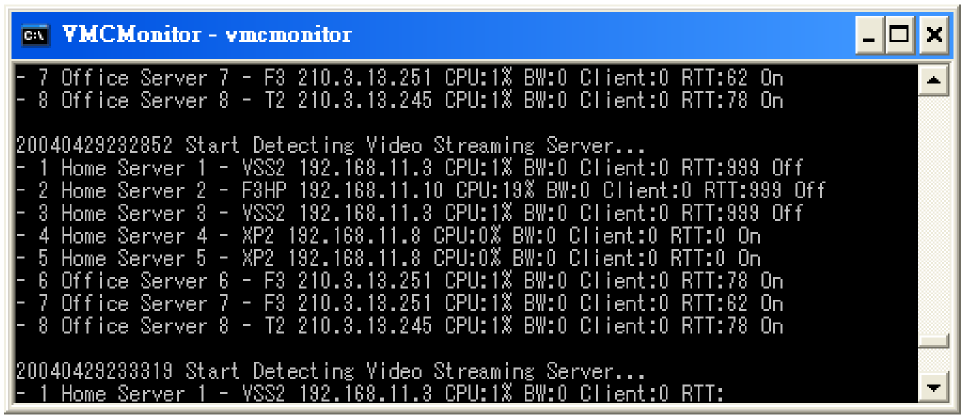


```
omniORB:   useTypeCodeIndirections = 1
omniORB:   verifyObjectExistsAndType = 1
VMC:IOR:010000001400000049444c3a44564f44564d432f564d433a312e3000010000000000000
640000000101020000d000000313932e3136382e31312e380000a3070e000000fef70391400000d
080000000000000020000000000000008000000010000000545441010000001c0000000100000
010001000100000001000105090101000100000009010100
VCP:IOR:010000001400000049444c3a44564f44564d432f5643503a312e3000010000000000000
640000000101020000d000000313932e3136382e31312e380000a3070e000000fef70391400000d
080000000001000002000000000000008000000010000000545441010000001c0000000100000
010001000100000001000105090101000100000009010100
VSS:IOR:010000001400000049444c3a44564f44564d432f5653533a312e3000010000000000000
640000000101020000d000000313932e3136382e31312e380000a3070e000000fef70391400000d
080000000002000002000000000000008000000010000000545441010000001c0000000100000
010001000100000001000105090101000100000009010100
omniORB: Invoke '_is_a' on remote: key<0x4e616d6553657276696365>
```

Figure 4.9.2 **VMC_impl.exe** is running

In Figure 4.9.2, It shows three CORBA server objects was started. They are the VMC server object, VCP server object and VSS server object.

VMCMonitor.exe is a program used to collect streaming server RTT and monitor the status of the server. If streaming fail to response the echo function call to VMCMonitor.exe, that server will be marked as off service. The echo function will be called periodically through .NET remoting technology. This program is developed by Visual Studio.NET 2003 C#.



```
VMCMonitor - vmcmonitor
- 7 Office Server 7 - F3 210.3.13.251 CPU:1% BW:0 Client:0 RTT:62 On
- 8 Office Server 8 - T2 210.3.13.245 CPU:1% BW:0 Client:0 RTT:78 On

20040429232852 Start Detecting Video Streaming Server...
- 1 Home Server 1 - VSS2 192.168.11.3 CPU:1% BW:0 Client:0 RTT:999 Off
- 2 Home Server 2 - F3HP 192.168.11.10 CPU:19% BW:0 Client:0 RTT:999 Off
- 3 Home Server 3 - VSS2 192.168.11.3 CPU:1% BW:0 Client:0 RTT:999 Off
- 4 Home Server 4 - XP2 192.168.11.8 CPU:0% BW:0 Client:0 RTT:0 On
- 5 Home Server 5 - XP2 192.168.11.8 CPU:0% BW:0 Client:0 RTT:0 On
- 6 Office Server 6 - F3 210.3.13.251 CPU:1% BW:0 Client:0 RTT:78 On
- 7 Office Server 7 - F3 210.3.13.251 CPU:1% BW:0 Client:0 RTT:62 On
- 8 Office Server 8 - T2 210.3.13.245 CPU:1% BW:0 Client:0 RTT:78 On

20040429233319 Start Detecting Video Streaming Server...
- 1 Home Server 1 - VSS2 192.168.11.3 CPU:1% BW:0 Client:0 RTT:
```

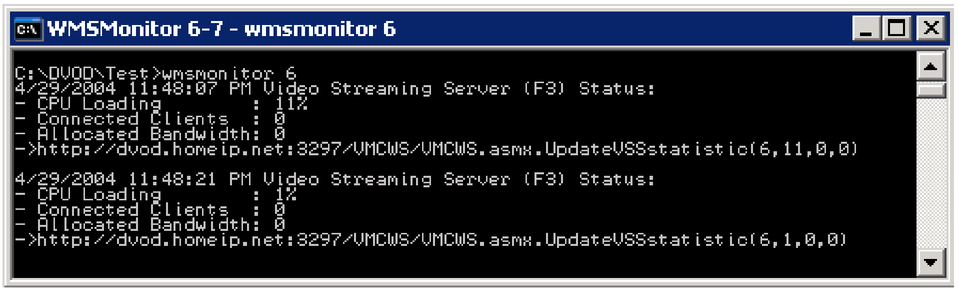
Figure 4.9.3 **VMCMonitor.exe** is running

In Figure 4.9.3, it shows the RTT times in million second and the status at the end of each logical servers. The status Off at the end means that server is no respond to the echo .NET remote function and will be marked as off service in the VMC.

All the three programs are not compulsory to run in the VMC Web server. In case of running them is different computers for load distribution, all streaming server CORBA client must addressed the **omniNames.exe** correctly in the network and **VMC_impl.exe** and **VMCMonitor.exe** must correctly connected to the database server.

4.10. Programs in Streaming Server

WMSMonitor.exe is a program used to get current server CPU utilization, allocated bandwidth and connected clients resources metrics and report to VMC through Web service interface periodically. This program is developed by Visual Studio.NET 2003 C#.

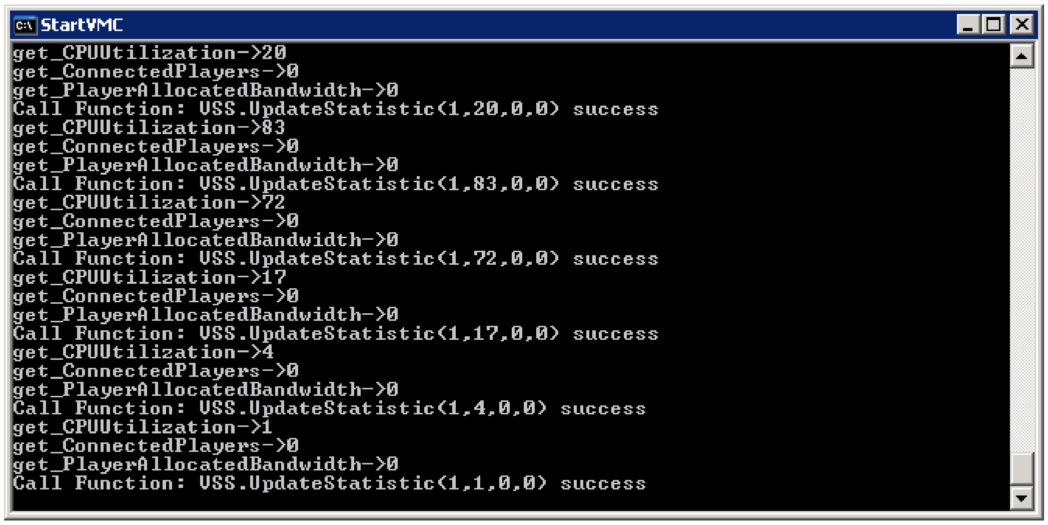


```
WMSMonitor 6-7 - wmsmonitor 6
C:\DVOD\Test>wmsmonitor 6
4/29/2004 11:48:07 PM Video Streaming Server (F3) Status:
- CPU Loading         : 11%
- Connected Clients   : 0
- Allocated Bandwidth: 0
->http://dvod.homeip.net:3297/VMCWS/VMCWS.asmx.UpdateVSSstatistic(6,11,0,0)

4/29/2004 11:48:21 PM Video Streaming Server (F3) Status:
- CPU Loading         : 1%
- Connected Clients   : 0
- Allocated Bandwidth: 0
->http://dvod.homeip.net:3297/VMCWS/VMCWS.asmx.UpdateVSSstatistic(6,1,0,0)
```

Figure 4.10.1 **WMSMonitor.exe** is running

In Figure 4.10, the **WMSMonitor.exe** gets the resources metrics and update to VMC though the displayed URL. The URL is saved in a text configuration file and can be change before start **WMSMonitor.exe.**

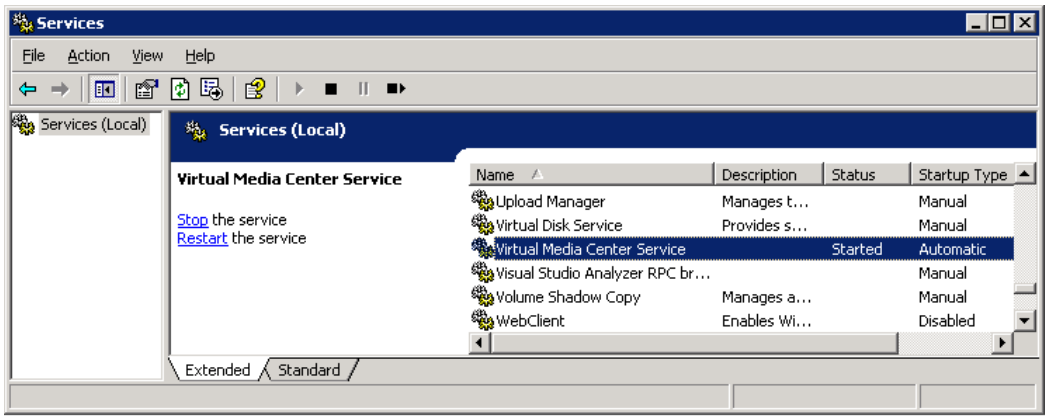
VSSMonitor.exe provides the same functions as **WMSMonitor.exe.** VSSMonitor.exe updates the resource metrics to VMC through CORBA remote object invocation. This program is developed by Visual Studio.NET 2002 VC++7.0.



```
get_CPUUtilization->20
get_ConnectedPlayers->0
get_PlayerAllocatedBandwidth->0
Call Function: VSS.UpdateStatistic(1,20,0,0) success
get_CPUUtilization->83
get_ConnectedPlayers->0
get_PlayerAllocatedBandwidth->0
Call Function: VSS.UpdateStatistic(1,83,0,0) success
get_CPUUtilization->72
get_ConnectedPlayers->0
get_PlayerAllocatedBandwidth->0
Call Function: VSS.UpdateStatistic(1,72,0,0) success
get_CPUUtilization->17
get_ConnectedPlayers->0
get_PlayerAllocatedBandwidth->0
Call Function: VSS.UpdateStatistic(1,17,0,0) success
get_CPUUtilization->4
get_ConnectedPlayers->0
get_PlayerAllocatedBandwidth->0
Call Function: VSS.UpdateStatistic(1,4,0,0) success
get_CPUUtilization->1
get_ConnectedPlayers->0
get_PlayerAllocatedBandwidth->0
Call Function: VSS.UpdateStatistic(1,1,0,0) success
```

Figure 4.10.2 **VSSMonitor.exe** is running

Windows Service called “**Virtual Media Center Service**” is the .NET remoting server object used to provide echo function called from the **VMCMonitor.exe**.



Virtual Media Center Service

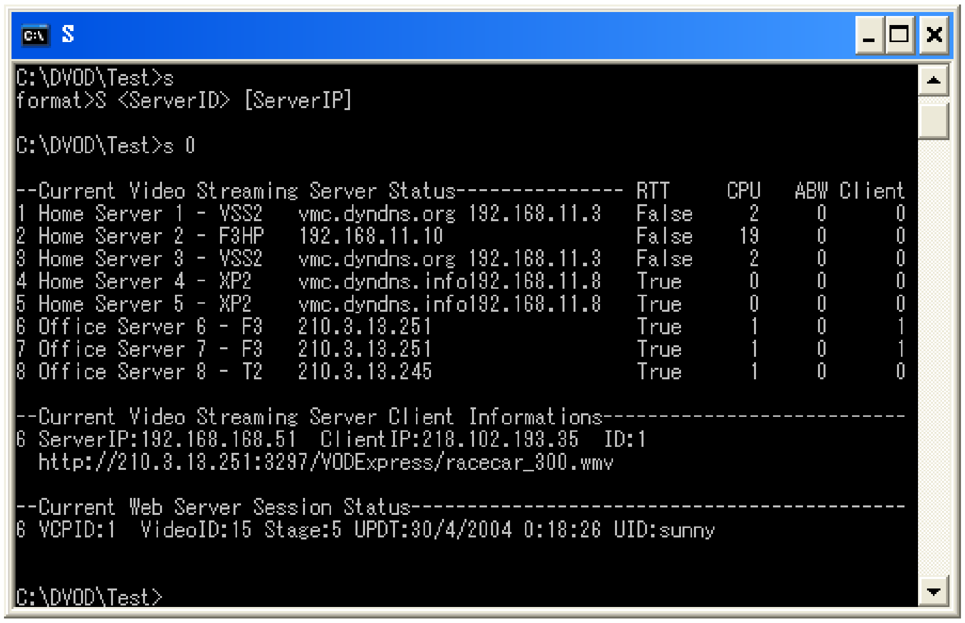
Stop the service
Restart the service

Name	Description	Status	Startup Type
Upload Manager	Manages t...		Manual
Virtual Disk Service	Provides s...		Manual
Virtual Media Center Service		Started	Automatic
Visual Studio Analyzer RPC br...			Manual
Volume Shadow Copy	Manages a...		Manual
WebClient	Enables Wi...		Disabled

Figure 4.10.3 Windows service “**Virtual Media Center Service**” is running

4.11. System Monitoring Tool

S.exe is a utility program used to display the current streaming server status, streaming client information and client activity in the Web server session.



```
C:\DVOD\Test>s
format>S <ServerID> [ServerIP]

C:\DVOD\Test>s 0

--Current Video Streaming Server Status--------------- RTT     CPU   ABW Client
1 Home Server 1 - VSS2   vmc.dyndns.org 192.168.11.3   False     2     0      0
2 Home Server 2 - F3HP   192.168.11.10                 False    19     0      0
3 Home Server 3 - VSS2   vmc.dyndns.org 192.168.11.3   False     2     0      0
4 Home Server 4 - XP2    vmc.dyndns.info192.168.11.8   True      0     0      0
5 Home Server 5 - XP2    vmc.dyndns.info192.168.11.8   True      0     0      0
6 Office Server 6 - F3   210.3.13.251                  True      1     0      1
7 Office Server 7 - F3   210.3.13.251                  True      1     0      1
8 Office Server 8 - T2   210.3.13.245                  True      1     0      0

--Current Video Streaming Server Client Informations--------------------------
6 ServerIP:192.168.168.51  ClientIP:218.102.193.35  ID:1
  http://210.3.13.251:3297/VODExpress/racecar_300.wmv

--Current Web Server Session Status-------------------------------------------
6 VCPID:1  VideoID:15 Stage:5 UPDT:30/4/2004 0:18:26 UID:sunny


C:\DVOD\Test>
```

Figure 4.11 S.exe ran and displayed the result in the console

In Figure 4.11, the parameter 0 means display the information for all streaming servers. The first part lists out the streaming server information such as IP, RTT status, CPU, BW (bandwidth) and connected clients. The RTT status false means the last .NET remoting echo function call was failed and no RTT figure update at the moment. The second part shows the client's IP and URL of the retrieving media. The last part shows the user information such as username, current selected video and status (Status 5 means watching video).

4.12. Client Interface

Subscriber access to use VMC services must provide a username and password for authentication. Please not that, for demonstration purpose, the URL for the VMC Web services VMCWS is linked in this page. The other links are a link to request open subscriber account, a link to the Remote Web server (Pocket PC version), Windows Media video demo and Windows Media Player 9 Series download. The upper “Virtual Media Center” header and the bottom advertisement bar are for decoration purpose only.

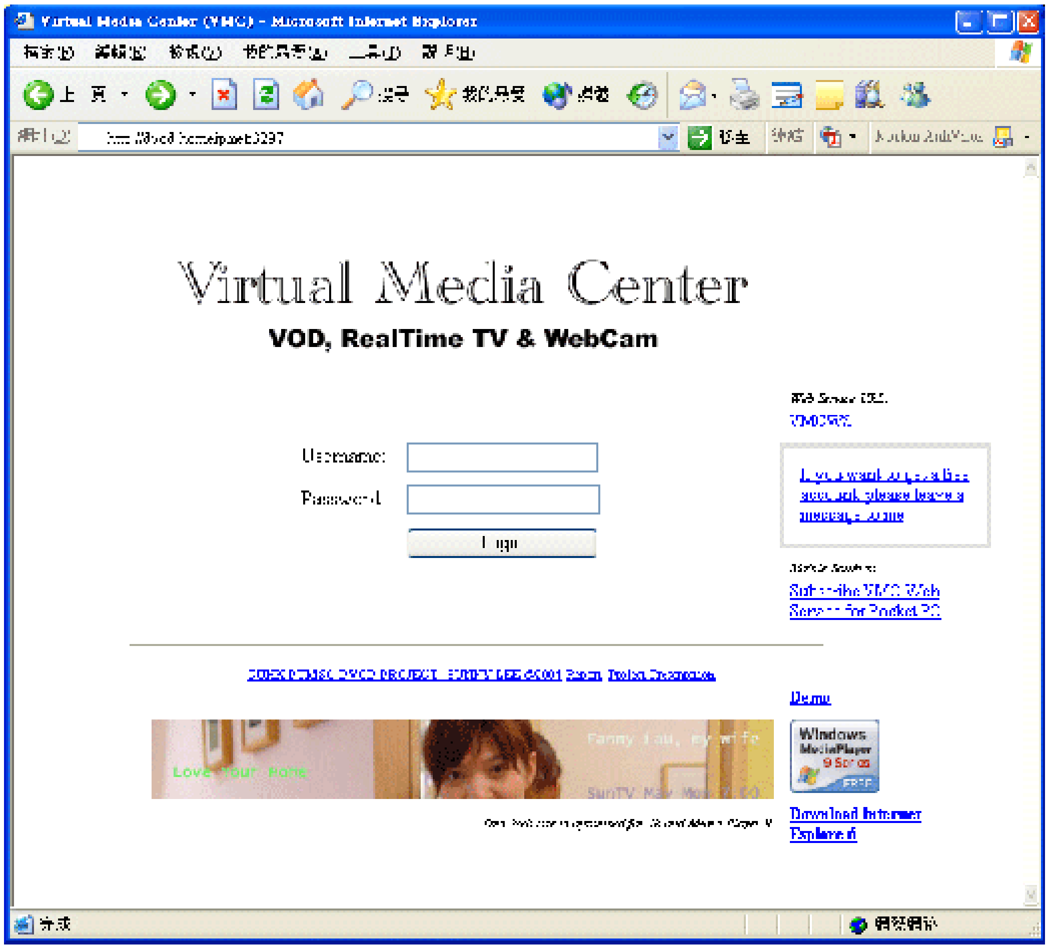
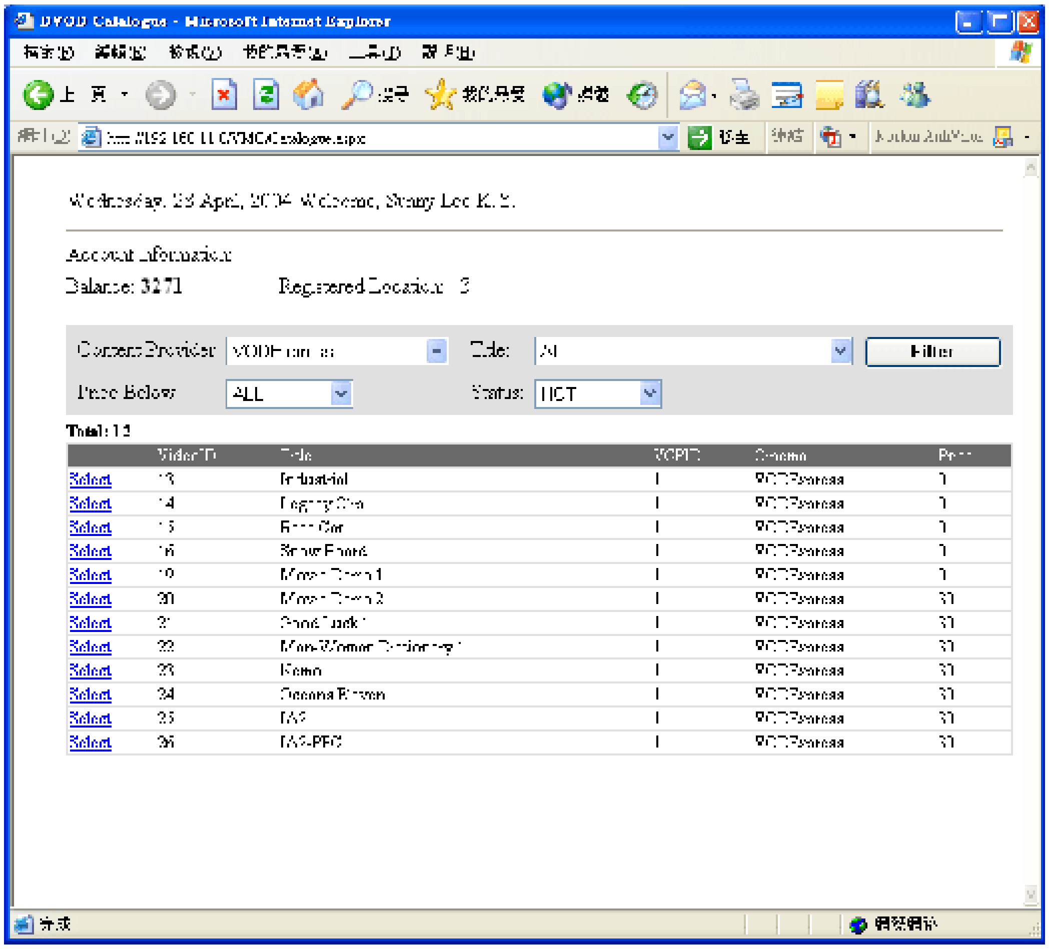

Virtual Media Center (VMC) - Microsoft Internet Explorer
Virtual Media Center
VOD, RealTime TV & WebCam
Username:
Password:
Login
If you want to get a free account, please leave a message to me
Subscribe VMC Web Server for Pocket PC
Demo
Windows MediaPlayer 9 Series FREE
Download Internet Explorer 6
Love Your Home

Figure 4.9.1 VMC home page

When subscriber login successful, the media catalogue page will be display. In the upper part of the page, there is a greeting and following the display name of the subscriber. Under the greeting is the account information. Current implementation will show the Balance and Registered location of the subscriber. Balance will be deducted every time the subscriber selected a media to watch. The Registered location is the geographic mark used for statistic server selection purpose but it is not implemented in this report.



Wednesday, 23 April, 2004 Welcome, Stany Lee K. S.

Account Information

Balance: 3271 Registered Location: 3

Content Provider: VODExpress Title: All Filter

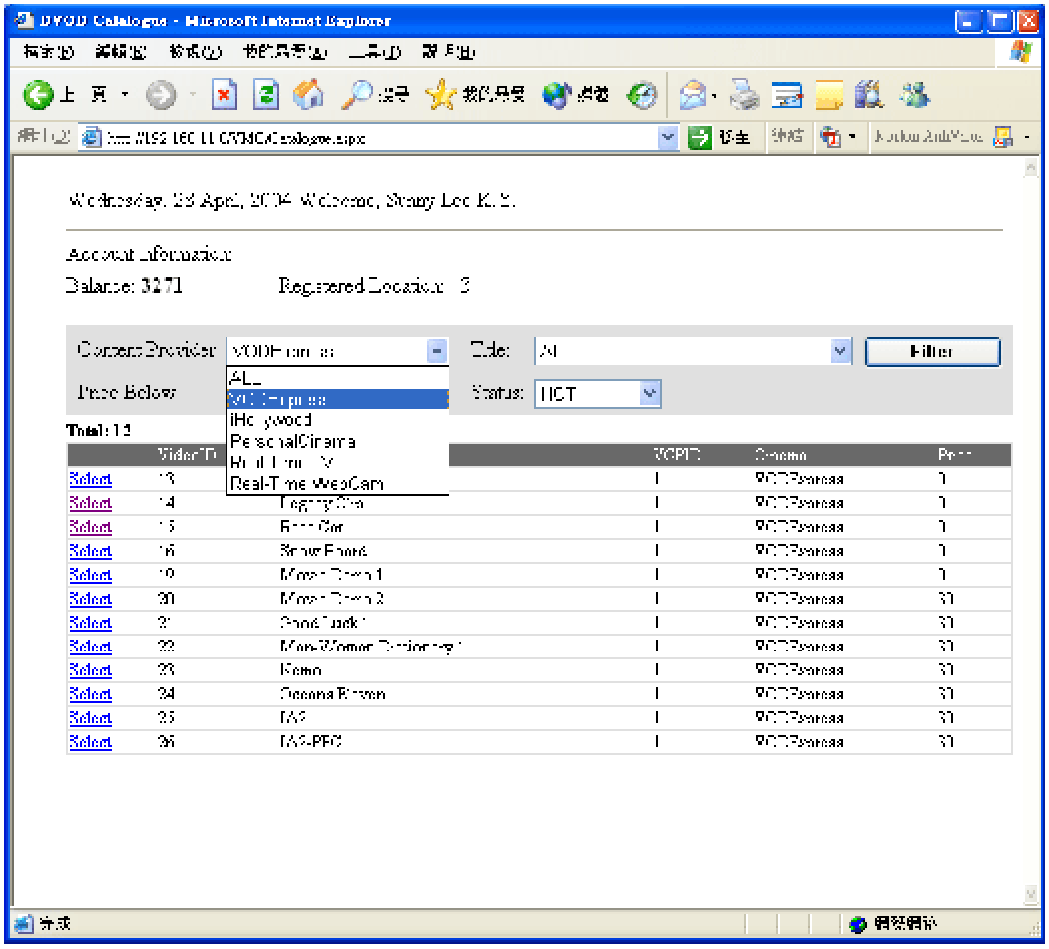
Price Below: ALL Status: HOT

Total: 12

	Video ID	Title	VOPID	Cinema	Price
Select	13	Industrial	1	VODExpress	1
Select	14	[illegible]	1	VODExpress	1
Select	15	[illegible] Car	1	VODExpress	1
Select	16	Snow Board	1	VODExpress	1
Select	19	Money Town 1	1	VODExpress	1
Select	20	Money Town 2	1	VODExpress	31
Select	21	Good Luck	1	VODExpress	31
Select	22	Man-Woman Dictionary	1	VODExpress	31
Select	23	Kemo	1	VODExpress	31
Select	24	Second Eleven	1	VODExpress	31
Select	25	IA2	1	VODExpress	31
Select	26	IA2-PRC	1	VODExpress	31

Figure 4.9.2 VMC catalogue page

As media content provider can register their content to VMC. There may more than one media content provider registered. Subscriber is usually selecting the desired content provider instead of browsing a long list of media content directly. The filter function in the catalogue page is designed for this purpose. For example, if the subscriber want to browse the media content from a provider called “VODExpress”, the step is select “VODExpress” in the Content Provider Combo box and then click the “Filter” button. Others similar options to filter are Title, Price Below and Status of a media, such as new or archive etc.



DVOD Catalogue - Microsoft Internet Explorer

Wednesday, 28 April, 2004 Welcome, Stany Lee K. S.

Account Information

Balance: 3271 Registered Location: 3

Content Provider: VODExpress

ALL

VODExpress

iHollywood

PersonalCinema

[illegible]

Real-Time WebCam

Title: All

Filter

Price Below

Status: HCT

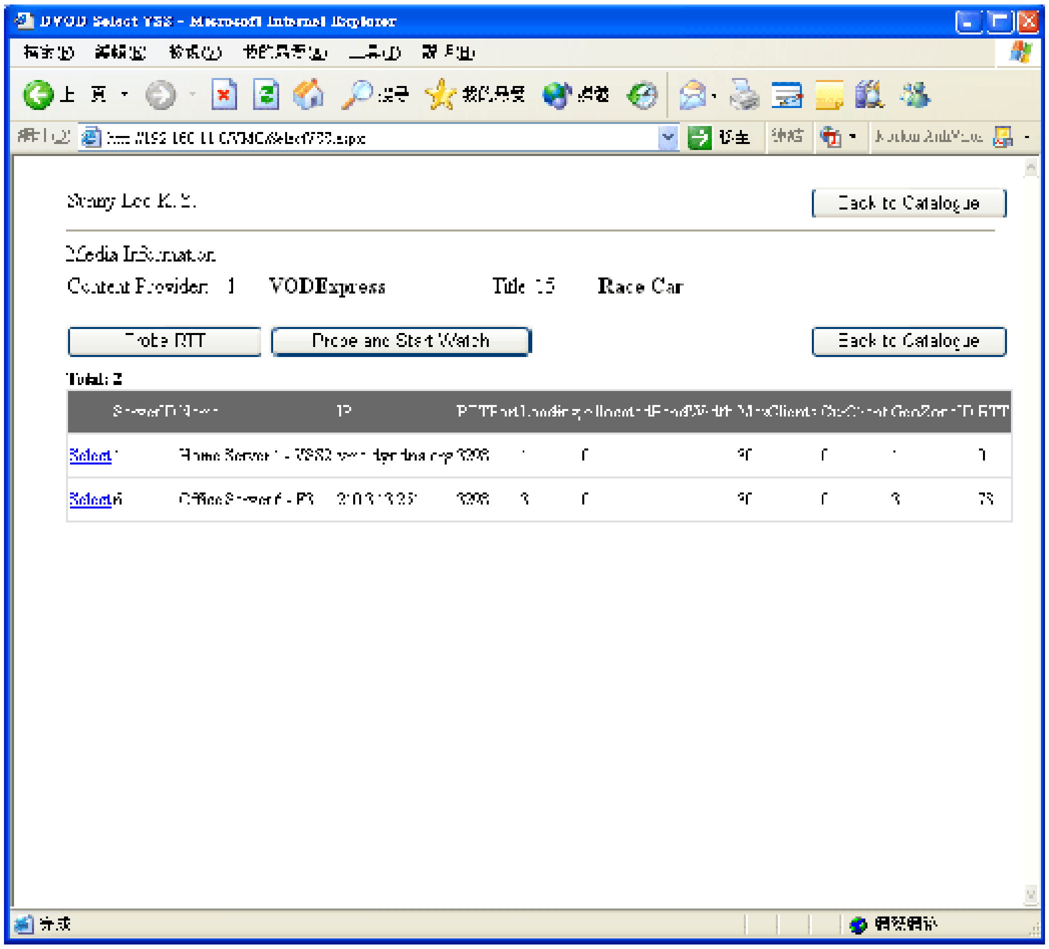
Total: 12

	VideoID		VCPID	Cinema	Price
Select	13		1	VODExpress	1
Select	14	Legacy [illegible]	1	VODExpress	1
Select	15	Race Car	1	VODExpress	1
Select	16	Snow Board	1	VODExpress	1
Select	19	Moon Town 1	1	VODExpress	1
Select	20	Moon Town 2	1	VODExpress	31
Select	21	Good Luck	1	VODExpress	31
Select	22	Man-Woman Dictionary [illegible]	1	VODExpress	31
Select	23	Kemo	1	VODExpress	31
Select	24	Ocean Bitten	1	VODExpress	31
Select	25	IA2	1	VODExpress	31
Select	26	IA2-PEC	1	VODExpress	31

Figure 4.9.3 VMC catalogue page: use filter function

When subscriber want to watch a media from a content provider, He/She should select the “Select” link in the left of the corresponding the media. Once clicked the “Select” link, the select server page will be display.

The select server page will list out all the available streaming servers which can provide the selected media content. The streaming server available will be short listed according to the most up-to-date resource metrics reported. The resources metric is reported by back-end CORAB or Web service interface periodically. Subscriber can click on the “Probe RTT” button to collect RTT figures and short list the best server again at any time. The RTT figures are also automatically collected by the VMC back-end process through .NET Remoting operations periodically.



Stany Lee K. S.

Back to Catalogue

Media Information

Content Provider: 1 VODExpress Title 15 Race Car

Probe RTT | Probe and Start Watch | Back to Catalogue

Total: 2

ServerID	Name	IP	RTTPort	Loading	allocatedBandWidth	MaxClients	CurClient	GeoZoneID	RTT
Select 1	Home Server 1 - VSS2	[illegible].dyndns.org	3398	1	0	90	0	1	0
Select 6	Office Server 6 - P3	210.3.13.23	3398	3	0	90	0	3	78

Figure 4.9.4 VMC select server page

The top most one is the best server for delivering the select media content when the select server page is loaded or the last “Probe RTT” button was clicked. Subscriber can click “Select” link of the best server (the first one) or other desired server and start watching streaming.

When click on “Select” link, the VOD session page will display and the corresponding media player component will be loaded. Subscriber will experience a litter buffering period and the selected media content will start to play once buffering complete.

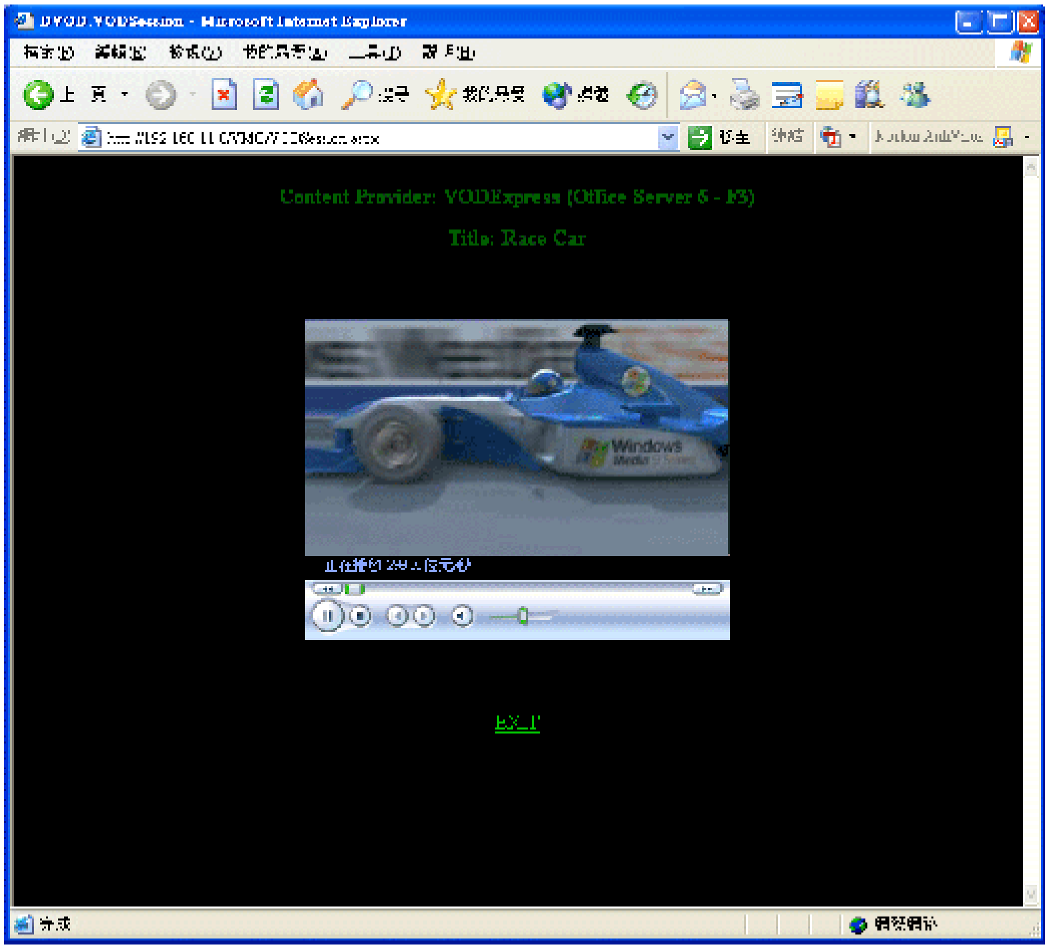


Figure 4.9.5 VMC VOD session page

The top of the VOD session page will display the name of the content provider, the server name and shows the media title in the second line.   Subscriber can execute VCR-like functions in the player during watch period.   When subscriber finish the VOD session or want to exit from VOD session, he/she can click the “EXIT” link. This link will direct the subscriber back to the server selection.   If subscriber is unsatisfied the quality of the previous server, he/she can re-select another server and try to watch if the quality is better.   Or subscriber can click the catalogue button in the server selection page to back to media catalogue page to select another media again.

5 Conclusion

5.1 Advantage

- CORBA and Web Service are well defined standard
- Independent to individual streaming technologies
- Web Interface for multiple devices: PC, mobile notebook, Pocket, Palm, 3G phone or Smart phone etc.
- Real-time server selection
- RTT independent to any underlying network path selection algorithm and easy to implement

5.2 Disadvantage

- Subscriber must installed a build in streaming player in Web browser to decode the target streaming
- Remote Web server must located nearest to subscriber to obtain the best RTT estimation for subscriber
- Streaming quality and concurrent connections are still limited to underlying bandwidth, compression technology and network equipment

5.3 Other Issues Need to Consider Further

Use of Middleware

The generic architecture for DVOD services needs to mask the heterogeneity across different computer network, operating systems, programming languages and products developed by different vendors. This advantage is obtained from employing a standard interface distributed middleware, CORBA. The trade off is the deployment of ORB to every client computers. Extra learn curve and development time is needed for the middleware. But it is worth to pay this extra effort as the middleware resolved the heterogeneity and provide better interoperability support.

Focus on Real-Time Server Selection

As VOD service is bandwidth demanding. The inherited limitation of bandwidth in existing computer networks makes bandwidth utilization is a critical issue need solve in DVOD system. This problem can be alleviated by a suitable server selection mechanism. The implemented system selecting the right video server in run-time using least CPU utilization, allocated bandwidth and connected clients. There may be other resources metrics or path selection algorithms available. All of them are need to under detail experiment and fine turn. Otherwise incorrect resource metering can cause unnecessary loading and influence system.

Use of Remote Method Invocation for Client Side Probe

The use of client side probe enables dynamic load balancing. In the DVOD system, client side perform probe by remote method invocation. This method provides the best RTT figures at run-time but may cause extra traffics. A compromise method is to use RTT agents which distributed in different geographic location to collect RTT periodically for clients. Client may pre-registered their access location or use some IP location algorithm to map the nearest RTT agent and use the RTT figures to aid server selection process.

No Assumption on Video Popularity

The video replicate and allocation strategy can increase system performance if the demand of a video and access location is predicted accurately. However, it is a difficult tasks and it is mainly rely on historical data and extrapolation. The extra cost for analysis and replication is out weight the result obtained. Another property observed from VOD service is that VOD emphasis on watch at any video at any time. The design principle on VOD service should consider every video may has the same possibility of request.

Transmission with unicast

Hybrid transmission such as multicast plus unicast, repeatedly broadcast selected videos, video batching etc. technique can improve bandwidth utilization but prohibit user interactive functions. Unicast is a simple technique but limit system capacity. Another assumption is that the video quality has high priority than system capacity in the user point of view instead of the system capacity point of view.

Set Top Box

Many VOS service architecture requires a STB. The STB provide hardware for employ different transmission method, caching, buffering and decryption. But is add extra setup cost and lack flexibility. The CORBA ORB and remote object client stubs must installed in the client side. This is less convenience than DVOD just using Internet browser or common streaming player, but it is convenience than using STB.

Fault Tolerance

Another important characteristic in distributed system is that there is no single point of failure. Failure of one component only lowers the system performance. In the current DVOD system, there is a single Video Management Server to control the admission, server selection and management of database updates. This is a major drawback in terms fault tolerance issue and may cause performance bottle-neck in the system. However, it will save cost for replicate update and simply design.

5.4 Future Works

The server selection mechanism implemented demonstrated a common practical solution in dynamic way I terms of run-time RTT probe between client and servers. The RTT collection agent improves the efficiency of static server selection mechanism and alleviate the network traffic and loading in pure run-time RTT probe mode. But considered current resources metrics are limited to CPU utilization, allocated bandwidth, connection client and RTT figures. There may be a lot of others factors can be consider and fine-tune, such as the capacity of server network bandwidth, capacity of maximum clients and quality of service (QoS) etc.

The security is another major issues need to be address in the future. Current generic architecture has partitioned the system into components by function groups. But there is no individual component responsibility for security purpose. The current implementation replies on the inhered security features of the employed streaming services (e.g. build in Windows Media Services security features). Individual security feature may be further consider and add in as a individual components. For example, if employ CORBA security service or add in individual components are worth to further study.

The copy right problem exists since the first appear of digital media, such as MP3. Video copy rights just another extension to this problem. Although digital watermarking has proposed to solve the identification of ownership and integrity of a digital video, but the ownership problem between different movies producer are a very complicated problem. There is some digital right management product, such as Microsoft was released to solve this kind of problem. However, a solution or standard from non propriety is a better than stick on some de-facto standard. The digital right management in the generic architecture is another important issues need to solve.

The current distributed video-on-demand services have established a generic architecture for multimedia streaming service in the Internet. But it is a starting point of development in terms of practical application in real world. The chosen of CORBA as the middleware provide a framework for solving heterogeneity. And Web service interface standard is going to dominate the interface standard technology in the future. These two technologies provide components in the generic architecture can be change implementation without affecting the whole system component collaboration.

6 Reference

[1] Zhou S. and King P., “A Simple Platform Independent Video on Demand Application”, School of Electrical Engineering and Telecommunications, The University of New South Wales, Australia.

[2] Anker Tal, Dolev Danny and Keidar Idit, “Fault Tolerant Video on Demand Services”, Institute of Computer Science, Lab. of Computer Science, MIT.

[3] Kao H.Y., Kuo Y.C., Huang S.T. and Wang B.F., “Design and Implementation of a PC-based Video-on-Demand System”, *in Proc. of the 12th International Conference on Information Networking (ICOIN-12)*, pp. 42-45, Tokyo, Japan.

[4] Kiyoshi Tanaka, Hideki Sakamato, Hideharu Suzuki, Kazutoshi Nishimura, “Distributed Architecture for Large-scale Video Servers”, *in Proc. of the International Conference on Information, Communications and Signal Processing (ICICS)*, vol.1, pp. 578-583, Sept 1997.

[5] Christos Bouras, Vaggelis Kapoulas, Agisilaos Konidaris, Afroditi Sevasti, “A Dynamic Distributed Video on Demand Service”, *in Proc. of the 20th International Conference on Distributed Computing Systems (ICDCS 2000).*

[6] Alexey Roytman, Israel Ben-Shaul and Israel Cidon, “DVS: A System for Distribution and Management of Global Video on Demand Services”, *in Proc. of the IEEE International Conference on Multimedia Computing and Systems*, vol. 2, pp. 343-346, June 1999.

[7] Fonseca N.L.S. and Franco C.M.R., “Using Distributed Servers to Provide Distributed Home Theatre Services”, *in Proc. of SBT/IEEE International Conference on Telecommunications Symposium (ITS’98)*, vol. 1, pp. 261-266, Aug 1998.

[8] Seabra dos Santos Daniela Alvin, Vieira Alex Borges, “Performance Analysis and Optimization of a Distributed Video On Demand Service”, *in Proc. of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS’03)*, 2003.

[9] Wu J.L., Chen J.K. and Chao H.W., "Load Shift Protocol Design in ATM-based VOD Systems", *in Proc. of ICOIN-12*, Tokyo, Japan, pp. 222-227, Jan 1998.

[10] Xiang Zhe, Zhang Qian, Zhu Wenwu, "Cache Replacement and Server Selection For Video Proxy Across Wireless Internet", *in Proc. of International Symposium on Circuits and Systems (ISCAS'03)*, vol. 2, pp. II-828-II-831, May 2003.

[11] Bakiras, S. and Li V.O.K., "Maximizing the Number of Users in an Interactive Video-on-Demand System", *IEEE Transactions on Broadcasting*, vol. 48(4), Dec 2002.

[12] Angelo Morzenti, Matteo Pradella, Matteo Rossi, Stefano Russo, Antonio Sergio, "A Case Study in Object-oriented Modeling and Design of Distributed Multimedia Applications", University di Napoli Federico II.

[13] Jehan F., Steven W., Darrell D., "A Universal Distribution Protocol For Video-on-Demand", *in Proc. of International Conference on Multimedia and Expo (ICME'00)*, vol. 1, pp. 49-52, Aug 2000.

[14] Tantaoui M.A., Hua K.A. and Sheu S., "Scalable Technique for VCR-like Interaction in Video-on-Demand Application", *in Proc. of the 22nd International Conference on Distributed Computing Systems Workshop*, pp. 246-251, July 2002.

[15] Chow Randy, Lee Chung-Wei, Liu C., "Traffic Dispersion Strategies for Multimedia Streaming", *in Proc. of the Eight IEEE Workshop on Future Trends of Distributed Computing Systems*, pp. 18-24, 31 Oct – 2 Nov 2001.

[16] Majumder Abhik, Purl Rohit and Ramchandram Kannan, "Distributed Multimedia Transmission from Multiple Servers", *in Proc. of the International Conference on Image Processing* (*ICIP'02)*, vol. 3, pp. III-177-III-180, June 2002.

[17] Chen Jen-Kai and Wu Jean-Lien C., "Adaptive Chaining Scheme for Distributed VOD Application", *IEEE Transaction on Broadcasting*, vol. 45(2), June 1999.

[18] Lim Eun-Ji, Park Seong-Ho, Hong Hyeon-Ok and Chung Ki-Dong, "A Proxy Caching Scheme for Continues Media Streams on the Internet", *in Proc. of the 15th International Conference on Information Networking* (*ICOIN'01)*, pp.

720-725, 31 Jan - 2 Feb 2001.

[19] Shyu Ing-Jye and Shieh Shiuh-Pyng, “Balancing Workload and Recovery Load on Distributed Fault-tolerant VOD Systems”, *IEEE Communications Letters,* vol. 2(10), pp. 288-290, Oct 1998.

[20] Chatschik C., Bisdikian and Baiju V. Patel, “Issues on Movie Allocation in Distributed Video-on-Demand Systems”, IBM Research Division.

[21] Zhou Xiaobo and Xu Cheung-Zhong, “Optimal Video Replication and Placement on a Cluster of Video-on-Demand Servers”, *in Proc. of IEEE International Conference on Parallel Processing (ICPP’02)*, 2002.

[22] González Sonia, Navarro Angeles, Lopez Juan and Zapata Emilio, “Load Sharing Based on Popularity in Distributed Video on Demand Systems”, *in Proc. of IEEE International Conference on Multimedia and Expo (ICME '02),* vol. 1, pp. 5-8, Aug 2002.

[23] Chan S.-H.G. and Tobagi F.A., “Threshold-based admission policies for video services “, *in Proc. of the Global Telecommunications Conference (GLOBECOM '99)*, vol. 4, 1999.

[24] Jehan F., Steven W. and Darrell D., “Combining Pay-Per-Vie and Video-on-Demand Services”, Dept. of Computer Science, University of Houston, Dept. of Computer Science, University of California.

[25] Qazzaz B., Suppi R., Cores F., Ripoll A., Hernandez P. and Luque E., “Admission Control Polices For Video On Demand Brokers”, Department of Computer Science, University Autonoma de Barcelona, Bellaterra, Spain, *IEEE* 2003.

[26] Yu Hongtao, Chor Ping Low, Anf Yacine, “Design Issues on Video-on-Demand Resources Management”, School of Electrical & Electrical Engineering, Nanyang Technological University, Singapore.

[27] Shahabi, C. and Banaei-Kashani, F., “Decentralized Resource Management for a Distributed Continuous Media Server”, *in Proc. of IEEE Transactions on Parallel and Distributed Systems,* vol. 13(11), pp. 1183-1200, Nov 2002.

[28] But Jason and Egan Greg, “Designing a Scalable Video-on-Demand System”, *in Proc. of IEEE International Conference on Communications, Circuits and Systems and West Sino Expositions*, vol. 1(29), pp. 559-565, July 2002.

[29] Mundur P., Sood A. and Simon R., “Threshold-based Admission Control for Multi-class Video-on-Demand Systems”, *in Proc. of IEEE International Conference on Performance, Computing and Communications (IPCCC '98)*, pp. 154-160, Feb 1998.

[30] Calvagna A., Puliafito A. and Vita L., “A Low-Cost/High-Performance Server for Video on Demand”, *IEEE* 1999.

[31] Chor Ping Low; Hongtao Yu; Ng, J.M.; Qingping Lin and Atif, Y., “An Efficient Algorithm for the Video Server Selection Problem”, *in Proc. of IEEE Global Telecommunications Conference (GLOBECOM’'00)*, vol. 3(27), Dec 2000.

[32] Chan Gary S.H. and Tobagi Fouad, “Distributed Servers Architecture for Networked Video Services”, *in Proc. of IEEE/ACM Transactions on Networking*, vol. 9(2), pp. 125-136, April 2001.

[33] Lee J-Y.B., “Parallel Video Servers: A Tutorial”, *in Proc. of* IEEE Multimedia, vol.5(2), pp. 20-28, June 1998.

[34] Rowe L.A., Boreczky J.S.; Berger D.A., Brubeck D.W. and Baldeschwieler J.E., “A Distributed Hierarchical Video-on-Demand System”, *in Proc. of International Conference on Image Processing*, Washington DC, October 1995.

[35] Hui C.K., Ng J.K., Wong Wai and Leung Karl R.P.H., “The Implementation of a Multi-Server Distributed MPEG Video System”, *in Proc. of the Seventh IEEE Real-Time Technology and Applications Symposium (RTAS’01)*, pp. 111-113, June 2001.

[36] Lee Jack Y.B., Wong P.C., “Performance Analysis of a Pull-Based Parallel Video Server”, *in Proc. of IEEE Transactions on Parallel and Distributed Systems*, vol. 11(12), December 2000.

[37] Dykes S.G., Robbins K.A. and Jeffery C.L., “An Empirical Evaluation of Client-side Server Selection Algorithms”, *in Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM’2000)*, vol. 3, pp.1361-1370, March 2000.

[38] Carter Robet L. and Crovella Mark E., “Server Selection Using Dynamic Path Characterization in Wide-Area Networks”, *in Proc. of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97)*, vol. 3, pp. 1014-1021, April 1997.

[39] Hanna Katrina M., Natarajan Nandini and Levine Brian Neil, “Evaluation of a Novel Two-Step Server Selection Metric”, *in Proc. of the Ninth International Conference on the Network Protocols*, pp. 290-300, Nov 2001.

[40] Mohamed Vall O., Mohanmed-Salem, “Scaling Server Selection using a Multi-Broker Architecture”, *in Proc. of the 23rd of International Conference on Distributed Computing Systems Workshop*, pp. 934-939, 2003.

[41] Amini Lisa and Schulzrinne Henning, “On Probe Strategies For Dynamic Multimedia Server Selection”, *in Proc. of IEEE International Conference on the Multimedia and Expo* (*ICME’02)*, vol. 1, pp. 393-396, 2002.

[42] Zhao Yinqing and Kuo C.-C.J., “Video-on-Demand Server System Design with Random Early Migration”, *in Proc. of the 2003 International Symposium on Circuits and Systems (ISCAS'03)*, vol. 2, May 2003.

[43] Mundur Padmavathi, Sood Arun and Simon Robert, “Threshold-Based Admission Control For Multi-Class Video-on-Demand Systems”, *in Proc. of IEEE International Conference on Performance, Computing and Communications (IPCCC'98)*, pp. 154-160, Feb 1998.

[44] Sumedh M., Nagarajan S., Douglas C. Schmidt, “Design and Performance of a CORBA Audio/Video Streaming Service”, *IEEE* 1999.

[45] Fitzpatrick T., Blair G.S., Coulson G., Davies N. and Robin P., “Software Architecture for Adaptive Distributed Multimedia Systems”, *IEEE Proc-Softw.*, vol. 145(5), pp. 163-171, Oct 1998.