**Multimedia Communication Over Distributed Systems**

Premchand Lingamgunta and Goutham Gandreddi

## Abstract

*Distributed Multimedia Systems is an area of active commercialization and research. This technology is widely viewed as the next generation technology for computers and communication networks. This paper will discuss some features of the technology, protocols related to DMS, its architecture, synchronization and scalability. Also, we will see some of the current trends in this technology.*

# 

**Introduction**

Research and development efforts in multimedia falls in two groups. One group concentrates on the stand-alone multimedia workstations and associated software and tools. The other combines multimedia with the distributed systems. The distributed multimedia system offers a **broader** spectrum of implementation possibilities in comparison to stand - alone systems. But in addition to the possibilities they all add a new dimension to the system complexity.

With the explosive growth of the Internet and dramatic increase in wireless access, there is a tremendous demand on multimedia service over wireless Internet. The third generation (3G) wireless networks, foreseen to be the enabling technology for multimedia services with 384kbps to 2Mbps bandwidth, makes it feasible for providing integrated service of data, voice, audio, and video across the wireless link. Continuous media applications, such as audio or video conferencing, Internet radio, on-line seminars, or video-on-demand systems, are becoming commonplace. These applications transmit data at regular intervals and require strict guarantees on maximum delay and minimum bandwidth. When these services are moved in wireless devices with latency over noisy wireless bottlenecks, they must be able to adapt a wide bandwidth range.

# Architecture

Distributed multimedia system consists of three basic components: an Information server, a wide area network and a multimedia client on the user site.

Traditional networks are used to provide error- free transmission. However, most multimedia applications can tolerate some errors in transmission due to corruption or packet loss without retransmission or correction.

Some of the differences in the traditional and multimedia communication are given in the table below:

|  |  |  |
| --- | --- | --- |
| Characteristics | Data Transfer | Multimedia  Transfer |
| Data rate | Low | High |
| Traffic pattern | Burst | Stream oriented,  highly burst |
| Reliability  requirements | No loss | Some loss |
| Latency time  requirements | None | Low (for  example: 20ms) |
| Mode of  communication | Point to Point | Multipoint |
| Temporal  relationship | None | Synchronized  transmission. |
|  |  |  |

From the above discussion we have noted that the Traditional networks do not suit multimedia communication. Transmission characteristics of existing Ethernet and Internet Protocols (CSMA/CD, TCP/IP) do not support the low latency, high bandwidth requirements of the audio video-based applications.

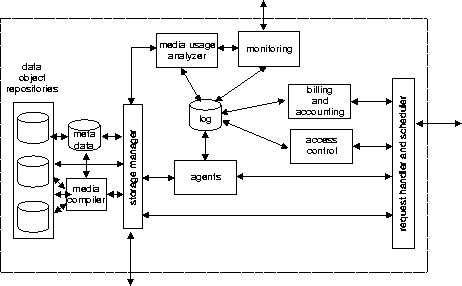
FDDI in its synchronized mode has low access latency and low jitter. FDDI also guarantees a bounded access delay and a predictable average bandwidth for synchronous traffic. However, Due to its high cost FDDI is at the moment used primarily for the backbone networks.

Asynchronous Transfer Mode (ATM) is rapidly emerging as the future protocol for multimedia communication. ATM provides great flexibility in the bandwidth allocation by assigning fixed length packets called cells, to support virtual connections. ATM can also increase the bandwidth efficiency by buffering and statistically multiplexing burst traffic at the expense of cell delay and loss. For the Internet, the Internet Engineering Task Force (IETF) is working on a TCP/IP interface for ATM.

## Multimedia Server

Current personal computers, workstations and servers are designed to handle traditional forms of data. Their performance is optimized for a scientific or transaction – oriented type of workload. These systems do not perform well for multimedia data, requiring fast data retrieval and guaranteed real time capabilities. The I/O capacity is usually a severe bottleneck.

### Components of a multimedia Server



**Figure:** Possible components and their internal relationship in a multimedia server

To satisfy the stringent system requirements the server architecture is quite complex. The above figure presents possible components in a multimedia server and their internal relations. The internal structure of the server and needed components depends on the purpose of the server. The presented server architecture is based on consideration of interactive news system requires.

The scheduler and request handler take care of the I/O data flow of the server. It handles the requests and manages the data flow to and from the server.

To be able to restrict the access only to certain users or network domains, the server must include access control. The access control manages which requests are allowed. Since many planned systems will include commercial services a billing and accounting module must be present.

The server requires administration and supervision. For this all transactions are logged in a log repository and a monitoring tool is provided. Since there will be a huge amount of log data available, means for filtering out the interesting information must exist. This could be done by a media usage analyzer, which filters out the interesting log information and presents it in an understandable format.

To manage the large amount of data a storage manager is provided. The storage manager manages the data in the object repositories and the data flow. Information about the data, so called metadata, could be saved in a separate repository, facilitating the data management and retrieval. To make it possible to serve the data in different formats, the data could be saved in some specified format and then changed to the wanted format by a media compiler. The media usage analyzer could give feedback of usage patterns to the storage manager, helping to decide what information should be stored where.

To be able to manage the data effectively and to be able to offer personalized services, the system should also include agents. The agents would select what data to present to the user, according to user preferences and access patterns. In this manner each user would get a personalized service.

To decrease server load, the processing of requests could be distributed among multiple servers. Every server would be its own entity, but they could then be managed in a centralized or distributed manner. Such

approaches need careful consideration and analysis, because of their complexity.

The design of a multimedia server needs thorough planning to meet the stringent requirements. Designing a high-performance multimedia server that can support multiple, simultaneous, full-motion video streams are still challenging. A great deal of work needs to be done in areas like real-time scheduling, parallel I/O, reliability, scalability, dynamic scheduling and caching techniques for multimedia data

**3. Comparative Analysis of Protocols**

We have some protocols which are used for real time multimedia processing. Now we will look at every aspect of each protocol.

Real Time transport protocol (RTP) is an end-to-end transport protocol. It used to support multimedia traffic on Internet. It may be implemented on top of UDP/IP or ATM, and it takes advantage of the MBONE, which allows bandwidth-efficient distribution of data to many users by eliminating redundant packet transmissions. It is already used to support many video-conferencing tools.

RTP offers a control protocol called RTCP that supports the protoco1 functionality. An RTCP message consists of a number of ‘stackable’ packets, each with its own type code and length indication. Their format is fairly similar to data packets; in particular, the type indication is at the same location. RTCP packets are multicast periodically to the same multicast group as data packets, Thus, they also serve as a liveness indicator of session members, even in the absence of transmitting media data. The functionality of RTCP is described briefly below: (1) Quality of Service monitoring and congestion control RTCP packets contain the necessary information for quality-of-service (QoS) monitoring. Since they are multicast, all session members can survey how the other participants are faring. Applications that have recently sent audio or video data generate a sender report. It contains information useful for inter-media synchronization, as well as cumulative counters for packets and bytes sent. These allow receivers to estimate the actual data rate. Session members issue receiver reports for all video or audio sources they have heard from recently. They contain information on the highest sequence number received, the number of packets lost, a measure of the inter-arrival jitter and timestamps needed to compute an estimate of the round-trip delay between sender and receiver issuing the report. (2) Inter-media synchronization The RTCP sender reports contain an indication of real-time (wall clock time) and a corresponding RTP timestamp. These two values allow the synchronization of different media, for example, lip-syncing of audio and video. (3) Identification RTF’ data packets identify their origin only through a randomly generated 32-bit identifier. For conferencing application, a bit more context is often desirable. RTCP messages contain an SDES (source description) packet, in turn containing a number of piece of information is the so-called canonical name, a globally unique identifier of the session participant. Other possible SDES items include the user’s name, email address, telephone number, application information and alert message. (4) Session size estimation and scaling RTCP packets are sent periodically by each session member. The desire for upto-date control information has to be balanced against of the desire to limit control traffic to small percentage of data traffic, even with sessions consisting of several hundred members. The control traffic load is scaled with the data traffic load so that it makes up a certain percentage of the nominal data rate (5%).

**Disadvantages of RTP in the wireless environment**

We identify several disadvantages in RTP adaptability, especially when deployed in the wireless environment. They consist of the following three main problems: (I) RTP cannot dynamically control session bandwidth in RTP, the session announcement must specify a session bandwidth that is maximal for the entire length of the multicast session. We assume that there are 10 data sources in the session and they each may generate data at 128 kbps so that the initiator must specify a session bandwidth of 10\*128=128Okbps. If, at some point in the session, there are only one source currently generating data, the bandwidth used for RTCP messages should be 5% of 128kbps (the current bandwidth used) or 6.4kbps, but instead would be calculated to be 5% of 128Okbps (the specified session bandwidth) or 64kbps. Control traffic then accounts for 50% (four times the recommended level) of the total bandwidth used for this session. These excess control messages waste bandwidth on the network and processing time at each RTP participant. This wasted bandwidth is especially a problem in the wireless environment where bandwidth is scarce. We suggest the excess messaging be eliminated by implementing dynamic session bandwidths which change with the actual bandwidth used. (2) RTCP cannot adapt the control bandwidth scalability. Currently, fixed networks can provide high bandwidth resources, such as 155Mbps ATM and lGbps Ethernet etc. But wireless links always provides very low bandwidth capabilities to the end user. So it is commonly necessary to modify the data stream seen on the wireless network to reduce its bandwidth usage. However this can not be achieved by RTCP. For example, consider this common scenario: A multicast session on the lOOMbps Ethernet (fmed networks) is made up of one data source and 10 data sinks (perhaps an on-line lecture series). The data source is transmitting MPEG-1 video at 1.2Mbps. The control bandwidth is set to 5% of 1.2Mbps or 6Okbps giving an approximate session bandwidth of 1.2Mbps. If a wireless host joins this conference, the manager must reduce the data bandwidth usage to 128kbps (teleconferencing quality) for support the wireless host. Unless the control bandwidth is also decreased, however, RTCP reports will continue to use 6Okbps of bandwidth on the wireless link. Control message would then account for 32% of the total wireless bandwidth used for this session. Clearly, this is unacceptable. (3) RTCP have not flexible message format sent Finally, consider again the scenario of 1 data source and 20 receivers. While the data source may be interested in the inception statistics of all the receivers, clearly the receivers themselves would not be. The Receiver Report, however, are multicast to all participants, wasting bandwidth for those participants who do not make use of them. We suggest a mechanism to provide more flexibility in the RTCP packet format, allowing receiver or sender reports to be removed when desired, or replaced with summary information. As wireless hosts tend to be resources, they will commonly act as receivers only and will benefit from the reduced bandwidth usage of the modified RTCP packet.

Real-time Transport Protocol security nor the Secure Real-time Transport Protocol key management mechanisms is adequate, the complexity of SRTP based on Datagram Transport Layer Security (DTLS) is too high to reduce the scope of use, so this paper designs a real-time encrypted transport mechanism, using DTLS to achieve key management and negotiating encryption algorithms. Extend the DTLS, and then achieve encryption of RTP based on the DTLS and packaging transmission. To be compared in the particular case, the method, under certain safety requirements, is better suited for the high efficiency transmission network. The mechanism provides a good foundation for real-time multipath transmission.

With the rapid development of network technology, transmitting real-time audio and video streaming in the IP network has become a concerned network application. The standard protocols for real-time audio and video streaming transmitting include the Real-Time Transport Protocol (RTP) and the Real-Time Transport Control Protocol (RTCP). RTP is the Internet standard published by the IETF, the current version of the document is RFC3550 [1]. RTP is only responsible for real-time data transmission; the main function of RTCP is providing periodic reports for the reception quality feedback and flow controlling. Traditional RTCP feedback model is that all the receivers periodically send RTCP packets to provide feedback about the current network quality of service with multicast model. With the increasing size of the multicast session, for not occupying too much network resources, RTCP packets may maintain a fixed proportion of the bandwidth. However that will extend the RTCP packet transmission interval and reduce the feedback rate.

The lack of a session to guarantee long-lived service and a function for control over the delivery of real-time streaming data in HTTP was a barrier in supporting a Video-on-Demand POD) service over the Internet. Recently, this barrier has been overcome with the introduction of Real-Time Streaming Protocol (RTSP).

The Real Time Streaming Protocol (RTSP) is a network control [protocol](https://en.wikipedia.org/wiki/Communications_protocol) designed for use in entertainment and communications systems to control [streaming media](https://en.wikipedia.org/wiki/Streaming_media) [servers](https://en.wikipedia.org/wiki/Web_server). The protocol is used for establishing and controlling media sessions between end points. Clients of media servers issue [VHS](https://en.wikipedia.org/wiki/VHS)-style commands, such as play, record and pause, to facilitate real-time control of the media streaming from the server to a client (Video On Demand) or from a client to the server (Voice Recording).

The transmission of streaming data itself is not a task of RTSP. Most RTSP servers use the [Real-time Transport Protocol](https://en.wikipedia.org/wiki/Real-time_Transport_Protocol) (RTP) in conjunction with [Real-time Control Protocol](https://en.wikipedia.org/wiki/RTCP) (RTCP) for media stream delivery. However, some vendors implement proprietary transport protocols. The RTSP server software from [RealNetworks](https://en.wikipedia.org/wiki/RealNetworks), for example, also used RealNetworks' proprietary [Real Data Transport](https://en.wikipedia.org/wiki/Real_Data_Transport) (RDT).

Real-Time Messaging Protocol (RTMP) was initially a [proprietary protocol](https://en.wikipedia.org/wiki/Proprietary_protocol) developed by [Macromedia](https://en.wikipedia.org/wiki/Macromedia) for [streaming](https://en.wikipedia.org/wiki/Streaming_media) audio, video and data over the Internet, between a [Flash](https://en.wikipedia.org/wiki/Adobe_Flash) player and a server. Macromedia is now owned by [Adobe](https://en.wikipedia.org/wiki/Adobe_Systems), which has released an incomplete version of the specification of the protocol for public use.

The RTMP protocol has multiple variations:

The "plain" protocol which works on top of and uses TCP port number 1935 by default.

RTMPE which is RTMP encrypted using Adobe's own security mechanism. While the details of the implementation are proprietary, the mechanism uses industry standard cryptographic primitives.

RTMP is [encapsulated](https://en.wikipedia.org/wiki/Encapsulation_(networking)) within [HTTP](https://en.wikipedia.org/wiki/HTTP) requests to traverse firewalls. RTMPT is frequently found utilizing cleartext requests on [TCP](https://en.wikipedia.org/wiki/Transmission_Control_Protocol) [ports](https://en.wikipedia.org/wiki/Port_(computer_networking)) 80 and 443 to bypass most corporate traffic filtering. The encapsulated session may carry plain RTMP, RTMPS, or RTMPE packets within.

RTMFP, which is RTMP over  [UDP](https://en.wikipedia.org/wiki/User_Datagram_Protocol) instead of TCP, replacing RTMP Chunk Stream. The Secure [Real-Time Media Flow Protocol](https://en.wikipedia.org/wiki/Real-Time_Media_Flow_Protocol) suite has been developed by Adobe Systems and enables end‐users to connect and communicate directly with each other (P2P).

RTMP is a TCP-based protocol which maintains persistent connections and allows low-latency communication. To deliver streams smoothly and transmit as much information as possible, it splits streams into fragments, and their size is negotiated dynamically between the client and server. Sometimes, it is kept unchanged; the default fragment sizes are 64 bytes for audio data, and 128 bytes for video data and most other data types. Fragments from different streams may then be interleaved, and [multiplexed](https://en.wikipedia.org/wiki/Multiplexed) over a single connection. With longer data chunks, the protocol thus carries only a one-byte header per fragment, so incurring very little [overhead](https://en.wikipedia.org/wiki/Overhead_(engineering)). However, in practice, individual fragments are not typically interleaved. Instead, the interleaving and multiplexing is done at the packet level, with RTMP packets across several different active channels being interleaved in such a way as to ensure that each channel meets its bandwidth, latency, and other quality-of-service requirements. Packets interleaved in this fashion are treated as indivisible and are not interleaved on the fragment level.

The RTMP defines several virtual channels on which packets may be sent and received, and which operate independently of each other. For example, there is a channel for handling RPC requests and responses, a channel for video stream data, a channel for audio stream data, a channel for out-of-band control messages (fragment size negotiation, etc.), and so on. During a typical RTMP session, several channels may be active simultaneously at any given time. When RTMP data is encoded, a packet header is generated. The packet header specifies, amongst other matters, the ID of the channel on which it is to be sent, a timestamp of when it was generated (if necessary), and the size of the packet's payload. This header is then followed by the actual payload content of the packet, which is fragmented according to the currently agreed-upon fragment size before it is sent over the connection. The packet header itself is never fragmented, and its size does not count towards the data in the packet's first fragment. In other words, only the actual packet payload (the media data) is subject to fragmentation.

RTCP works hand in hand with [RTP](https://www.3cx.com/pbx/rtp/). RTP does the delivery of the actual data, whereas RTCP is used to send control packets to participants in a call. The primary function is to provide feedback on the quality of service being provided by RTP.

RTP is originated and received on even port numbers, and the associated RTCP communication uses the next higher odd port number. It transports statistics and information such as octet and packet counts, jitter, and round-trip time. An application can use this information to control QoS parameters and choose, for example, to use a different codec.

RTCP does not provide any flow encryption or authentication methods, but such mechanisms can be implemented with the use of the Secure Real-time Transport Protocol (SRTP).

Routing - The network must decide how to trans- port packets from the source to the receiver of the flow (or receivers of the flow, in the case of multi- cast). Thus, the second component of the architecture is a routing protocol that provides quality unicast and multicast paths. There are many approaches to unicast routing, and several different approach- es to multicast routing exist as well. None of the current proposals have yet dealt sufficiently with the interaction between routing and quality of service constraints; that is the subject of future research.

Resource Reservation - For the network to deliver a quantitatively specified quality of service to a particular flow, it is usually necessary to set aside certain resources, such as a share of bandwidth or a number of buffers, for that flow. This ability to create and maintain resource reservations on each link along the transport path is the third component of the architecture. Two approaches to resource reservation are described elsewhere in this article, we describe another.

Admission Control - Because a network's re- sources are finite, it cannot grant all resource reservation requests. In order to maintain the network load at a level where all quality of service commitments can be met, the network architecture must contain an admission control algorithm that determines which reservation requests to grant and which to deny, thereby maintaining the net- work load at an appropriate level.

Packet Scheduling - After every packet trans- mission, a network switch must decide whether or not to transmit the next packet, and which is next. These decisions are controlled by the packet scheduling algorithm -which lies at the heart of any network architecture because it determines the qualities of service the network can provide. There are many proposed packet scheduling algorithms.

So finally, RTSP is a real-time streaming protocol. Meaning, you can stream whatever you want in real time. So, you can use it to stream LIVE content (no matter what it is, video, audio, text, presentation...). [RTP](http://www.ietf.org/rfc/rfc3550.txt) is a transport protocol which is used to transport media data which is negotiated over RTSP.

You use RTSP to control media transmission over RTP. You use it to setup, play, pause, teardown the stream...

So, if you want your server to just start streaming when the URL is requested, you can implement some sort of RTP-only server. But if you want more control and if you are streaming live video, you must use RTSP, because it transmits SDP and other important decoding data.

 RTMP to help Web servers stream low-latency, on-demand content across the Web efficiently. Low latency is important when you wish to view smooth video in your browser. RTMP servers, such as the Flash Media Server, also support live video transmissions and can stream audio and other types of data as well. If someone loses an Internet connection while viewing our RTMP content, the system can reconnect and resume streaming. Internet users enjoy videos that start faster and play smoothly when viewing streaming content using RTMP.

 RTCP

works hand in hand with RTP. RTP does the delivery of the actual data, whereas RTCP is used to send control packets to participants in a call.

# 4. Current Trends in DMS

Currently some of the distributed multimedia systems are:

1. **Video on Demand**: The consumer can select a video or any program on demand. The application consists of Interactive features like forward, rewind and pause.
2. **News and Reference Services:** News on Demand is similar to VOD but it provides sophisticated news retrieval and reference services that combine live and achieved video, access to textual data and still photography from various sources. The information is delivered based on a filtering criterion kept by the user.
3. **Interactive shopping and electronic commerce:** Home shopping will provide a customizable shopping environment.

Customers will be effectively and rapidly focus on the products and services that are of interest to them.

1. **Entertainment and games:** Interactive entertainment may become a frequently used service. Games will consist of simple applications that will be downloaded to the set top device thus not incurring the significant cost associated with the use of server and network facilities.
2. **Distance Learning:** Educational interactive programming and distance learning are areas where the research is going on. Current indications are there that these may become popular but not have sufficient commercial use to the providers.

# 5. Conclusion

There is no single protocol that could address all the needs for a DMS architecture, lot of factors( bandwidth, number of users, security level , plethora of devices ) come into picture in deciding which protocol is suitable for the user, so user need to sacrifice some of the features and chose which protocol best fits his/her/organization’s purpose.

DMS (Distributed Multimedia Systems) still has a vast scope to improve, with respect to the increasing demand of the users and the rate at which they are consuming the data. With the advance in networking have made some improvements in the way in which the streaming services are delivered to the user. And we need better protocols to support the requirements of the users and also the protocols need to be able to cover the wide range of media devices at the same time.

# References

[1]- A Distributed Multimedia Commu-nication System and its Applications to E-Learning - Hans L. Cycon, Thomas C. Schmidt, Matthias Wählisch, Mark Palkow and Henrik Regensburg

[2] T. C. Schmidt and M. Wählisch, “Seamless Multicast Handover in a Hierarchical Mobile IPv6 Environment (M-HMIPv6),” individual, Internet Draft – work in progress 01, February 2004.

[3]- The multicast system for seamless live multimedia in WLAN -Yoseop Woo & Iksoo Kim

[4] A.D Birrel and B.J. Nelson, “Implementing Remote Procedure Call”, ACM Trans. On Computer System, vol.2,nol, pp39-59, Feb. 1984.

[5] - A Group RPC Protocol for Distributed Systems - Chang-Seuk Lee, Kwang-Hui Lee, Jong-Kun Lee

[6]- OPTIMAL RESOURCE ALLOCATION FOR VIDEO COMMUNICATION OVER DISTRIBUTED SYSTEMS - Yifeng He and Ling Guan

[7] - Synchronization Techniques in Distributed Multimedia Presentation - Shahab Ud Din , Dick Bulterman

[8] - Network Architecture for Distributed Multimedia Systems, Lancaster

University.

[9] - DMS ’97 Pacific Workshop on Distributed Multimedia Systems.

[10] - DMS ‘2001 Seventh International Conference on Distributed Multimedia Systems.

[11] - Formal Specification of Distributed Multimedia Systems. - GS Blair, L. Blair, H. Bowman, and A. Chetwynd