Chapter 9

Multimedia Communication Systems

9.1 Definition of Multimedia Communication

The consideration of multimedia applications supports the view that local systems expand toward distributed solutions. Applications such as kiosks, multimedia mail, collaborative work systems, virtual reality applications and others require high-speed networks with a high transfer rate and communication systems with adaptive, lightweight transmission protocols on top of the networks.

From the communication perspective, we divide the higher layers of the Multimedia Communication System (MCS) into two architectural subsystems: *an application subsystem* and *a transport subsystem.*

9.2 Application Subsystem

Collaborative Computing

The current infrastructure of networked workstations and PCs, and the availability of audio and video at these end-points, makes it easier for people to cooperate and bridge space and time. In this way, network connectivity and end-point integration of multimedia provides users with a collaborative computing environment. Collaborative computing is generally known as Computer-Supported Cooperative Work (CSCW).

There are many tools for collaborative computing, such as electronic mail, bulletin boards (e.g., Usenet news), screen sharing tools (e.g., ShowMe from Sunsoft), text-based conferencing systems(e.g., Internet Relay Chat, CompuServe, American Online), telephone conference systems, conference rooms (e.g., VideoWindow from Bellcore), and video conference systems (e.g., MBone tools nv, vat). Further, there are many implemented CSCW systems that unify several tools, such as Rapport from AT&T, MERMAID from NEC and others.

Collaborative Dimensions

Electronic collaboration can be categorized according to three main parameters: time, user scale and control. Therefore, the collaboration space can be partitioned into a three-dimensional space.

Time

With respect to time, there are two modes of cooperative work: asynchronous and synchronous. Asynchronous cooperative work specifies processing activities that do not happen at the same time; the synchronous cooperative work happens at the same time.

User Scale

The user scale parameter specifies whether a single user collaborates with another user or a group of more than two users collaborate together. Groups can be further classified as follows:

- A group may be *static* or *dynamic* during its lifetime. A group is *static* if its participating members are pre-determined and membership does not change during the activity. A group is *dynamic* if the number of group members varies during the collaborative activity, i.e., group members can join or leave the activity at any time.
- Group members may have different roles in the CSCW, e.g., a member of a group (if he or she is listed in the group definition), a participant of a group activity (if he or she successfully joins the conference), a conference initiator, a conference chairman, a token holder or an observer.
- Groups may consist of members which have *homogeneous* or *heterogeneous* characteristics and requirements of their collaborative environment.

Control

Control during collaboration can be *centralized* or *distributed*.

Centralized control means that there is a chairman (e.g., main manger) who controls the collaborative work and every group member (e.g., user agent) reports to him or her. *Distributed control* means that every group member has control over his/her own tasks in the collaborative work and distributed control protocols are in place to provide consistent collaboration.

Other partition parameter may include *locality*, and *collaboration awareness*.

Locality partition means that a collaboration can occur either in the same place (e.g., a group meeting in an officer or conference room) or among users located in different place through tele-collaboration.

Group communication systems can be further categorized into *computer-augmented collaboration systems*, where collaboration is emphasized, and *collaboration-augmented computing systems*, where the concentrations are on computing.

Group Communication Architecture

Group communication (GC) involves the communication of multiple users in a synchronous or an asynchronous mode with centralized or distributed control.

A group communication architecture consists of a *support model*, *system model* and *interface model*. The GC support model includes group communication agents that communicate via a multi-point multicast communication network as shown in following figure.

Group communication agents may use the following for their collaboration:

Group Rendezvous

Group rendezvous denotes a method which allows one to organize meetings, and to get information about the group, ongoing meetings and other static and dynamic information.

Shared Applications

Application sharing denotes techniques which allow one to replicate information to multiple users simultaneously. The remote users may point to interesting aspects (e.g., via tele-pointing) of the information and modify it so that all users can immediately see the updated information (e.g., joint editing). Shared applications mostly belong to collaboration transparent applications.

Conferencing

Conferencing is a simple form of collaborative computing. This service provides the management of multiple users for communicating with each other using multiple media. Conferencing applications belong to collaboration-aware applications.

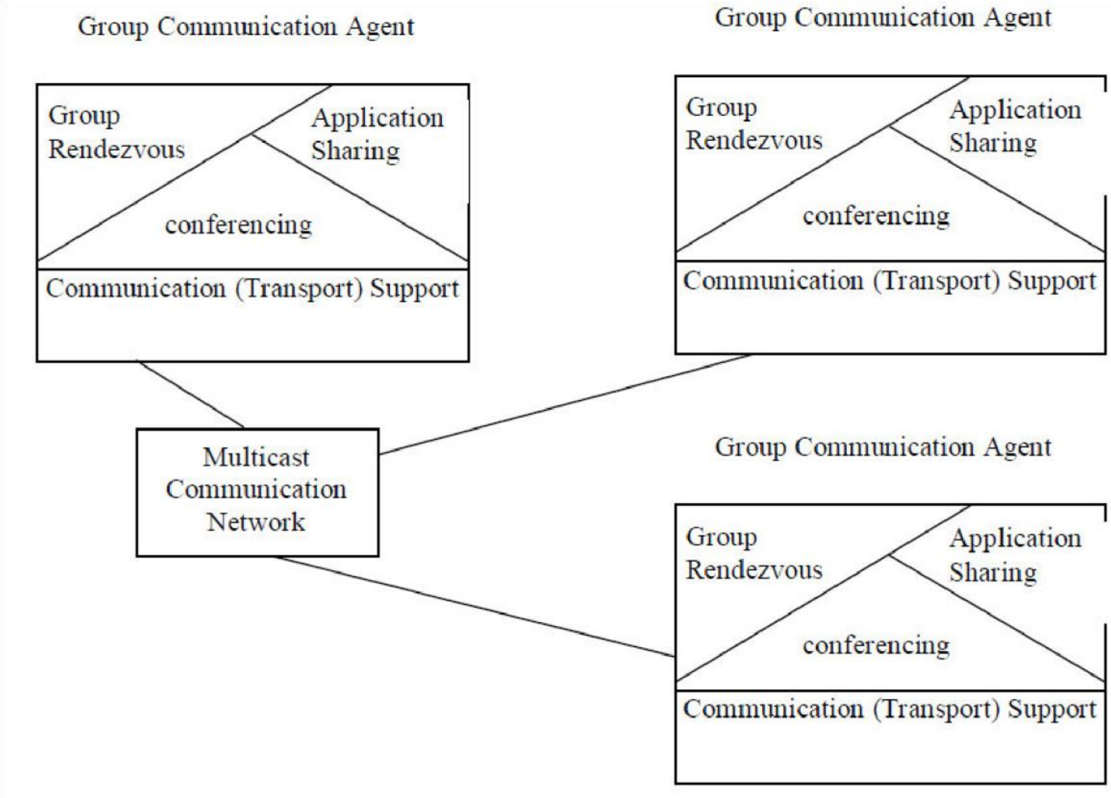
Group Communication Agent
Group Rendezvous
Application Sharing
conferencing
Communication (Transport) Support
Group Communication Agent
Group Rendezvous
Application Sharing
conferencing
Communication (Transport) Support
Multicast Communication Network
Group Communication Agent
Group Rendezvous
Application Sharing
conferencing
Communication (Transport) Support

Figure 9.1: Group communication support model

The GC system model is based on a client-server model. Clients provide user interfaces for smooth interaction between group members and the system. Servers supply functions for accomplishing the group communication work, and each server specializes in its own function.

Application Sharing Approach

Sharing applications is recognized as a vital mechanism for supporting group communication activities. Sharing applications means that when a shared application program (e.g., editor) executes any input from a participant, all execution results performed on the shared object (e.g., document text) are distributed among all the participants. Shared objects are displayed, generally, in shared windows.

Application sharing is most often implemented in collaboration-transparent systems, but can also be developed through collaboration-aware, special-purpose applications. An example of a software toolkit that assists in development of shared computer applications is Bellcore's Rendezvous system (language and architecture).

Shared applications may be used as conversational props in tele-conferencing situations for collaborative document editing and collaborative software development.

An important issue in application sharing is shared control. The primary design decision in sharing applications is to determine whether they should be centralized or replicated:

Centralized Architecture

In a centralized architecture, a single copy of the shared application runs at one site. All participants' input to the application is then distributed to all sites.

The advantage of the centralized approach is easy maintenance because there is only one copy of the application that updates the shared object. The disadvantage is high network traffic because the output of the application needs to be distributed every time.

Replicated Architecture

In a replicated architecture, a copy of the shared application runs locally at each site. Input events to each application are distributed to all sites and each copy of the shared application is executed locally at each site.

The advantages of this architecture are low network traffic, because only input events are distributed among the sites, and low response times, since all participants get their output from local copies of the application. The disadvantages are the requirement of the same execution environment for the application at each site, and the difficulty in maintaining consistency.

Conferencing

Conferencing supports collaborative computing and is also called synchronous telecollaboration.

Conferencing is a management service that controls the communication among multiple users via multiple media, such as video and audio, to achieve simultaneous face-to-face communication. More precisely, video and audio have the following purposes in a teleconferencing system:

Video is used in technical discussions to display view-graph and to indicate how many users are still physically present at a conference. For visual support, workstations, PCs or video walls can be used.

Audio is an important component in tele-conferencing for describing and clarifying visual information. Therefore, quality audio, with true full-duplex communication and echo cancellation, and possibly enhanced with spatial queues, is necessary.

Session Management

Session management is an important part of the multimedia communication architecture. It is the core part which separates the control, needed during the transport, from the actual transport. Session management is extensively studied in the collaborative computing area; therefore, we concentrate on architectural and management issues in this area.

A session management architecture is built around an entity-session manager which separates the control from the transport. By creating a reusable session manager, which is separated from the user-interface, conference-oriented tools avoid a duplication of their effort.

The session control architecture consists of the following components:

Session Manager

Session manager includes local and remote functionalities. Local functionalities may include:

Membership control management, such as participant authentication or presentation of coordinated user interfaces;

Control management for shared workspace, such as floor control

Media control management, such as intercommunication among media agents or synchronization

Configuration management, such as an exchange of interrelated QoS parameters of selection of appropriate services according to QoS; and

Conference control management, such as an establishment, modification and a closing of a conference.

Media agents

Media agents are separate from the session manager and they are not responsible for decisions specific to each type of media. The modularity allows replacement of agents. Each agent performs its own control mechanism over the particular medium, such as mute, unmute, change video quality, start sending, stop sending, etc.

Shared Workspace Agent

The shared workspace agent transmits shared objects (e.g., telepointer coordinate, graphical or textual object) among the shared application.

Session Control

Each session is described through the session state. This state information is either private or shared among all session participants. Dependent on the functions, which an application required and a session control provides, several control mechanisms are embedded in session management:

Floor control:

In a shared workspace, the floor control is used to provide access to the shared workspace. The floor control in shared application is often used to maintain data consistency.

Conference Control:

In conferencing applications, Conference control is used.

Media control:

This control mainly includes a functionality such as the synchronization of media streams.

Configuration Control:

Configuration control includes a control of media quality, QOS handling, resource availability and other system components to provide a session according to user's requirements. *Membership control*:

This may include services, for example invitation to a session, registration into a session, modification of the membership during the session etc.

9.3 Transport Subsystem

Requirements

Distributed multimedia applications put new requirements on application designers, As well as network protocol and system designers. We analyze the most important enforced requirements with respect to the multimedia transmission.

Network multimedia applications by themselves impose new requirements onto data handling in computing and communications because they need:

1. Substantial data throughput,
2. fast date forwarding,
3. service guarantees, and
4. multicasting

Data Throughput

Audio and video data resemble a stream-like behavior, and they demand, even in a compressed mode, high data throughput. In a workstation or network, several of those streams may exist concurrently demanding a high throughput.

Fast Data Forwarding

Fast data forwarding imposes a problem on end-systems where different applications exist in the same end-system, and they each require data movement ranging from normal, error-free data transmission to new time-constraint traffic types. But generally, the faster a communication system can transfer a data packet, the fewer packets need to be buffered. The requirement leads to a careful spatial and temporal resource management in the end-systems and routers/switches.

Service Guarantees

Distributed multimedia applications need service guarantees; otherwise, their acceptance does not come through as these systems, working with continuous media, compete against radio and television services. To achieve services guarantees, resource management must be used. Without resource management in end-systems and switches/routers, multimedia systems cannot provide reliable QoS to their users because transmission over unreserved resources leads to dropped or delayed packets.

Multicasting

Multicast is important for multimedia-distributed applications in terms of sharing resources like the network bandwidth and the communication protocol processing at end systems.

Transport Layer

Transport protocols, to support multimedia transmission, need to have new features and provide the following function, semi-reliability, multicasting, NAK (None-Acknowledgment)-based error recovery mechanism and rate control.

First, we present transport protocols, such as TCP and UDP, which are used in the Internet protocol stack for multimedia transmission, and secondly, we analyze new emerging transport protocols, such as RTP, XTP and other protocols, which are suitable for multimedia.

Transmission Control Protocol (TCP)

Early implementations of video conferencing applications were implemented on top of the TCP protocol. TCP provides a reliable, serial communication path, or virtual circuit, between processes exchanging a full-duplex stream of bytes. Each process is assumed to reside in an internet host that is identified by an IP address.

Each process has a number of logical, full-duplex ports through which it can set up and use as full-duplex TCP connections.

User Datagram Protocol (UDP)

UDP is a simple extension to the Internet network protocol IP that supports multiplexing of datagrams exchanged between pairs of Internet hosts. It offers only multiplexing and check summing, nothing else. Higher-level protocols using UDP must provide their own retransmission, packetization, reassembly, flow control, congestion avoidance, etc.

Real-time Transport Protocol (RTP)

RTP is an end-to-end protocol providing network transport function suitable for applications transmitting real-time data, such as audio, video or simulation data over multicast or unicast network services.

Xpress Transport Protocol (XTP)

XTP was designed to be an efficient protocol, taking into account the low error ratios and higher speeds of current networks. It is still in the process of augmentation by the XTP Form to provide a better platform for the incoming variety of applications. XTP integrates transport and network protocol functionalities to have more control over the environment in which it operates.

XTP is intended to be useful in a wide variety of environments, from real-time control systems to remote procedure calls in distributed operating systems and distributed databases to bulk data transfer. It defines for this purpose six service types: connection, transaction, unacknowledged data gram, acknowledged datagram, isochronous stream and bulk data.

In XTP, the end-user is represented by a context becoming active within an XTP implementation.

Network Layer

The requirements on the network layer for multimedia transmission are a provision of high bandwidth, multicasting, resource reservation and Qos guarantees, new routing protocols with support for streaming capabilities and new higher-capacity routers with support of integrated services.

Internet Protocol (IP)

IP provides for the unreliable carriage of datagrams form source host to destination host, possibly passing through one or more gateways (routers) and networks in the process.

Internet Group Management Protocol (IGMP)

Internet Group Management protocol (ICMP) is a protocol for managing Internet multicasting groups. It is used by conferencing applications to join and leave particular multicast group. The basic service permits a source to send datagrams to all members of a multicast group. There are no guarantees of the delivery to any or all targets in the group.

Resource reservation Protocol (RSVP)

RSVP is a protocol which transfers reservations and keeps a state at the intermediate nodes. It does not have a data transfer component. RSVP messages are sent as IP datagrams, and the router keeps "soft state", which is refreshed by periodic reservation messages. In the absence of the refresh messages, the routers delete the reservation after a certain timeout.

9.4 Quality of Service and Resource Management

Every product is expected to have a quality apart from satisfying the requirements. The quality is measured by various parameters.

Parameterization of the services is defined in ISO (International Standard Organization) standards through the notion of *Quality of Service (QoS)*. The ISO standard defines QoS as a concept for specifying how "good" the offered networking services are. QoS can be characterized by a number of specific parameters.

QoS Layering

Traditional QoS (ISO standards) was provided by the network layer of the communication system. An enhancement of QoS was achieved through inducing QoS into transport services. For MCS, the QoS notion must be extended because many other services contribute to the end-to-end service quality.

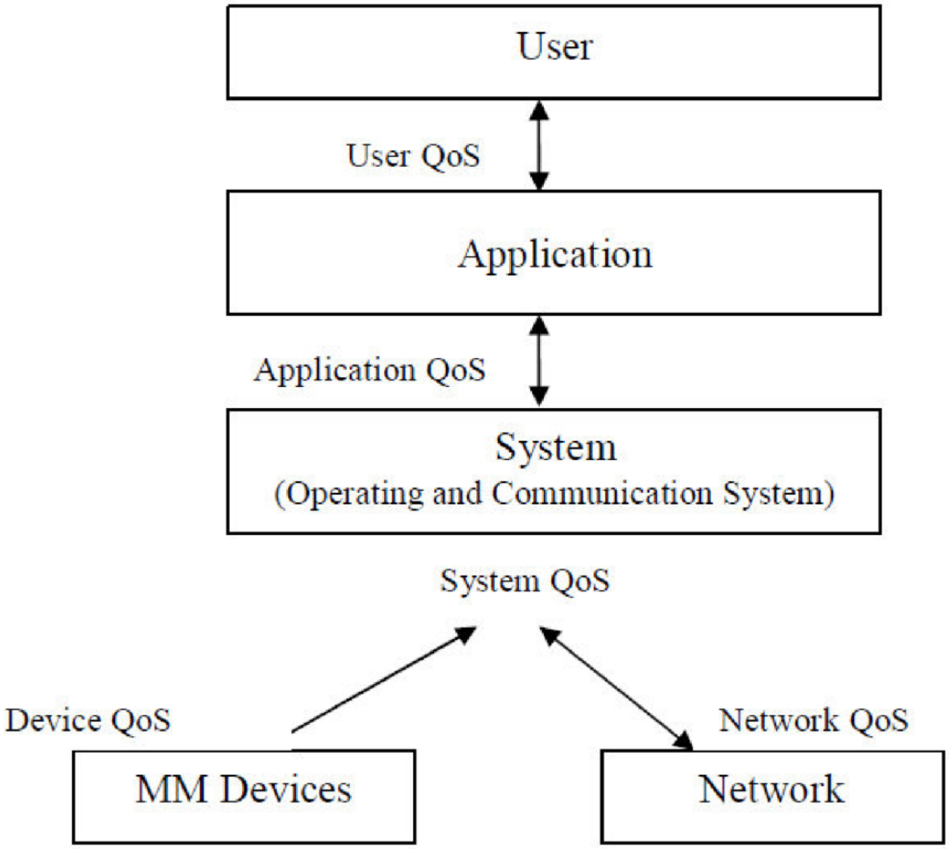

Figure 9.2: Quality of Service Layered model for the Multimedia Communication System

The MCS consists of three layers: application, system (including communication services and operating system services), and devices (network and Multimedia (MM) devices). Above the application may or may not reside a human user.

This implies the introduction of QoS in the application (application QoS), in the system (system QoS) and in the network (network QoS). In the case of having a human user, the MCS may also have a user QoS specification. We concentrate in the network layer on the network device and its QoS because it is of interest to us in the MCS. The MM devices find their representation (partially) in application QoS.

Resource Management

Multimedia systems with integrated audio and video processing are at the limit of their capacity, even with data compression and utilization of new technologies. Current computers do not allow processing of data according to their deadlines without any resource reservation and real-time process management.

Resource management in distributed multimedia systems covers several computers and the involved communication networks. It allocates all resources involved in the data transfer process between sources and sinks. In an integrated distributed multimedia system, several applications compete for system resources. This shortage of resources requires careful allocation. The system management must employ adequate scheduling algorithms to serve the requirements of the applications. Thereby, the resource is first allocated and then managed.

Resource management in distributed multimedia systems covers several computers and the involved communication networks. It allocates all resources involved in the data transfer process between sources and sinks.

Resource

A resource is a system entity required by tasks for manipulating data. Each resource has a set of distinguishing characteristics:

- There are *active* and *passive* resources. An *active* resource is, for example, the CPU or a network adapter for protocol processing; it provides a service. A *passive* resource is, for example, the main memory (buffer space) or bandwidth (link throughput); it denotes some system capabilities required by active resources.
- A resource can be either used *exclusively* by one process or *shared between various processes*. For example, a loudspeaker is an exclusive resource, whereas bandwidth is a shared resource.
- A resource that exists only once in the system is known as a *single resource*, otherwise it is a *multiple resource*. In a transputer-based multiprocessor system, the individual CPU is a multiple resource.