

Mechanisms for Effective and Efficient Dissemination of Multimedia

Saraswathi Krithivasan, Sridhar Iyer
Kanwal Rekhi School of Information Technology
Indian Institute of Technology, Bombay
{saras, sri}@it.iitb.ac.in

Abstract:

Choice of mechanisms for disseminating multimedia contents to geographically dispersed participants is influenced by many factors including network topology, encoding technique, service model, and mode of transmission. Encoding techniques render the multimedia file sizes manageable while a multicast service model provides a scalable architecture to disseminate the contents. The underlying network can be a private *Intranet*, *Internet*, or *heterogeneous* which can be a combination of several network topologies. The mode of transmission can be, *synchronous*, when live or recorded contents are transmitted to multiple users simultaneously or *on-demand*, when the files are served as per users' requests.

We measure effectiveness of delivery by the Quality of Service (QoS) experienced by end users. Efficiency is measured in terms of optimal utilization of system and network resources in delivering contents. Given a specific encoding technique and a multicast service model, mechanisms for effective and efficient delivery of multimedia contents vary depending on the nature of the underlying network topology and the mode of transmission deployed.

The first objective of this paper is to survey and classify the existing mechanisms for effective and efficient dissemination of multimedia, with a view to identifying their applicability to various modes of transmission. Heterogeneous architectures comprising of satellite, terrestrial links as well as the Internet are increasingly deployed for multimedia applications like distance education. Our second objective is to analyze applicability of the surveyed mechanisms to heterogeneous architectures and motivate directions for further research. To this end, a hierarchical representation of a heterogeneous network is presented and applicability of the mechanisms in reference to this representation is discussed. A case study of an instance of a heterogeneous network implemented for the Distance Education Program (DEP) of IIT Bombay is presented to stress the practical significance of research in this area.

Key words: QoS, Multimedia dissemination, Heterogeneous networks, Synchronous transmission, On-demand transmission, Distance education

1. Introduction

The increasing popularity of applications such as live broadcasts of events, streaming stored movies, video games, video conferencing, and distance education implies that an increasing amount of multimedia content is being disseminated to users scattered at different locations. Several factors influence the design and implementation of techniques for the dissemination of multimedia contents. These factors are:

- **Network topology:** The underlying network can be Intranet, Internet or heterogeneous; An Intranet, can comprise of a LAN, satellite network, or a network which combines different network technologies, but typically controlled by a single administrative domain. Internet is a public network, which provides best effort service to all the applications sharing the resources. A heterogeneous network is a combination of private and public networks and thus can include interconnected Local Area Networks (LANs) and Wide Area Networks (WANs). Mechanisms needed for effective and efficient dissemination of multimedia vary based on the characteristics of the network, such as, resource availability, nature of links, link access mechanisms etc.
- **Encoding mechanism:** Multimedia files are huge in size compared to the data files. Several standards exist for efficient coding of the multimedia data to reduce their size. Moving Picture Experts Group (MPEG) [45] has defined a family of standards for coding audio-visual information, e.g., MPEG-1, MPEG-2, and MPEG-4 [45][46][43][29]. Encoding techniques generally trade off between the file size, resolution quality, and the complexity of the decoding algorithm. Based on the nature of the multimedia content, appropriate algorithms can be chosen that strikes a balance between these three factors.
- **Delivery mode:** Typically, the following delivery modes are used while streaming multimedia:
 - Synchronous mode:* where a streaming session is multicast to multiple receivers receiving the stream simultaneously. There are two ways in which synchronous transmission can happen:
 - *From a live source:* Here a live media stream is encoded on the fly and transmitted to receivers in real time.
 - *From a stored medium:* Here the media stream is played from a recorded source (e.g. tape, CD).In both cases, broadcast is synchronous and the clients receive the transmission simultaneously.
 - On-demand transmission:* where the media files are typically stored and served to clients as and when they request for the file.The mode of transmission places different requirements on the system and network resources. For example, synchronous transmission places huge demands on the bandwidth as all the receivers are served simultaneously; Synchronous transmission from a stored medium and on-demand delivery require storing of the media files, introducing server and storage management issues.
- **Service model:** Service model can be unicast, broadcast, or multicast. Unicast of content involves dedicated connections between the source and each of the receivers while a broadcast can be received by any receiver on the network. Typically a multicast model is assumed while delivering multimedia files to multiple clients, as serving the files to

individual receivers with a unicast stream will quickly deplete the network resources. Details of a multicast model and its associated protocols can be found in [31][33][29].

Given any combination of these factors, the mechanisms used for the delivery of multimedia contents must be:

- **Effective**, i.e., guarantee a minimum quality of reception to ensure a smooth playout of the multimedia files.
- **Efficient**, i.e., optimally use network and system resources, allowing for scalability.

Thus, the choice of mechanisms for effective and efficient delivery of multimedia becomes important. These factors that affect the dissemination of multimedia are summarized in Figure 1.

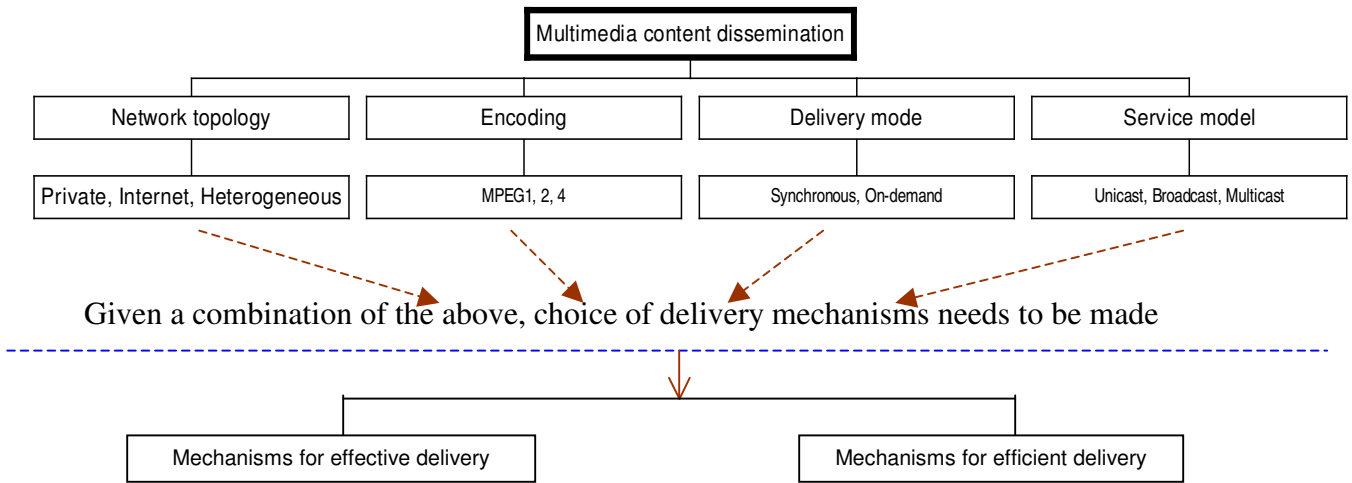


Figure 1: Factors affecting multimedia dissemination

Several survey papers have been written in the area of multimedia dissemination. However, these have dealt with one of the factors or a subset of the mechanisms discussed in this paper. For example, a survey on mechanisms for on-demand mode of delivery is presented in [29] while adaptation mechanisms are surveyed in [50]. Similarly, surveys of encoding mechanisms [43] [46], caching mechanisms [49], error-correction techniques [32], and multicast routing protocols [33] provide detailed reviews of work in these specific areas. In this paper, we have taken a global view of multimedia dissemination. We review the mechanisms representative of the techniques deployed in multimedia delivery, indicating whether a given mechanism or a class of mechanisms is suitable for the synchronous mode of delivery, on-demand mode of delivery, or both. Since we believe that heterogeneous architectures combining different network topologies would be deployed increasingly, we have presented a hierarchical representation of a heterogeneous network and discussed the applicability of the delivery mechanisms. A case study of an existing network for distance education is presented to identify the challenges that must be addressed in adapting the existing mechanisms to heterogeneous architectures.

The rest of the paper is organized as follows. Section 2 outlines the parameters for QoS, and presents the mechanisms for effective delivery. In Section 3, additional mechanisms needed

for efficient delivery of multimedia are outlined. While the existing techniques focus on the Internet, many important emerging multimedia applications such as distance education deploy heterogeneous architectures. A discussion of the applicability of delivery mechanisms to heterogeneous architectures are presented along with a case study in Section 4. Research efforts needed in this area are presented in Section 5.

2. Mechanisms for effective delivery

We say that a multimedia delivery mechanism is effective if it ensures a minimum required Quality of Service (QoS). QoS is a function of the quality of reception at the receiver. If the QoS of the received multimedia content is below the required minimum, it may not make any sense to the receiver. Whereas on a qualitative level, receiver perception is an important parameter to gauge QoS, several quantitative parameters are used to measure QoS:

- *Bandwidth*: The transmission rate of a communication link is typically referred to as the “bandwidth”, measured in bits/second. For multimedia applications, certain minimum bandwidth should be allocated to ensure the acceptable quality of reception at the receiving client.
- *Data loss*: When the network traffic increases, the queues at the routers become longer. As the buffer in a router is finite, an arriving data packet may be dropped, leading to loss of data. In order to ensure good quality of reception, loss of data packets has to be minimized.
- *Delay*: Multimedia applications are sensitive to delays. Packets have to arrive at the receiver within a certain time to be played in First In First Out (FIFO) order. In this sense, multimedia data can be termed “real-time” data. Thus, in the case of multimedia delivery, a packet that arrives late misses its turn to be played back and is hence considered lost.
- *Delay Jitter*: Due to the random nature of queuing delays in the network, packets can arrive at the receiver with varying inter packet delay. For multimedia data, this variation causes disturbances in the reception, referred to as “delay jitter”. To improve quality, it is important that inter packet arrival times are kept almost constant at the receiving client so that the resultant play out is smooth.

There are several mechanisms proposed in the literature, which enhance or ensure effectiveness of delivery. Firstly, application level correction techniques [29] can be applied to enhance the receiver perception, even when some transmission errors are encountered. We classify these as *correction-based* mechanisms and discuss them in Section 2.1. Secondly, constraints on network resources such as bandwidth allocated to the multimedia application need to be considered, while ensuring that packet loss, delay, and variation in inter packet arrival times are minimized. (Bandwidth availability can be affected either by the network conditions or constraints at the receiver’s end). We categorize mechanisms that ensure effective delivery considering the system and/or the network resources, as *resource-based* mechanisms and discuss them in section 2.2.

Resource-based mechanisms can be further categorized under the three dimensions: *adaptation mechanisms*, *reservation mechanisms*, and *hybrid mechanisms*. Adaptation mechanisms use knowledge about the network or end-user conditions to decide on the

appropriate level of resource usage. These mechanisms are discussed in section 2.2.1. *Reservation mechanisms* use admission control mechanisms to ensure that an application gets the required resources for smooth delivery. Different reservation mechanisms, which provide different levels of QoS are discussed in Section 2.2.2. *Hybrid mechanisms*, which use combinations of architectures and protocols to enhance effectiveness of multimedia delivery are discussed in Section 2.2.3.

A classification chart of the mechanisms for effective delivery is provided in Figure 2.

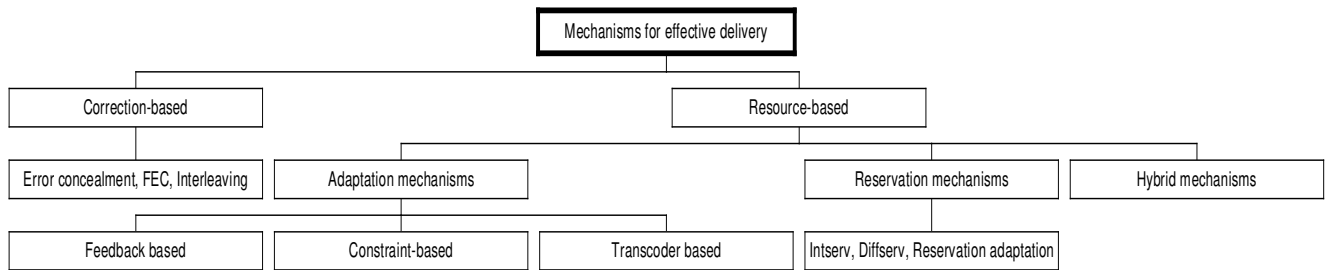


Figure 2: Classification of mechanisms for effective delivery

2.1. Correction-based mechanisms

In the case of multimedia data, a packet that arrives after its scheduled play out time is considered to be lost. Due to this property, mechanisms that are based on the retransmission of lost or corrupted data do not serve any purpose. For a detailed discussion of error recovery mechanisms, the reader is referred to [32][9][25][29]. Correction-based mechanisms either use simple error concealment mechanisms at the receiver or add redundant information at the senders to enhance the quality of reception. These are briefly discussed below.

2.1.1. Error concealment

These mechanisms are implemented at the receivers and require no support from the sender. In its simplest form, error concealment involves inserting the previous frame in the place of a lost frame. While such insertion techniques are very easy to implement, more complex interpolation-based schemes [32] can also be used to capture the changing characteristics of the content. These techniques attempt to interpolate a lost packet from packets surrounding it, to produce a replacement. Such techniques are computationally expensive and add more complexity.

2.1.2. Forward Error Correction (FEC)

Adding redundant information to the multimedia data packets to enable recovery from packet losses is the key idea behind FEC techniques. Here additional bandwidth cost is

incurred due to the redundancy. A simple FEC scheme [32][29] would be to send a block of redundant information every n packets as the $n+1^{th}$ packet. This mechanism would allow the receiver to reconstruct any one lost packet from the group of $n+1$ packets. However, this does not work if two or more packets are lost. Also, as all the $n+1$ packets have to be received before the play out can begin, this scheme increases the playout delay.

Another FEC scheme [32][29] is to send a low bit rate stream as the redundant stream. In this scheme, the sender creates two streams: a nominal stream and a low-resolution, low-bit rate stream which is the redundant stream. It appends the $n+1^{th}$ block from the redundant stream to the n^{th} block of the nominal stream to create the n^{th} packet. For example, as illustrated in Figure 3, packet 3 carries a low-resolution version of packet 2. When packet 2 is lost, the receiver can retrieve it from packet 3. Thus, whenever there is non-consecutive packet loss, the receiver can use the redundant information from the subsequent packet to recover from the loss. Although, the recovered packet will be of lower quality, overall quality can still be maintained as the stream is made up of mostly high quality packets of data with an occasional low quality packet. Another set of correction techniques use Erasure codes [29]. Erasure codes use algorithms to construct multiple independent encoded versions for the same set of data packets, which are sent as redundant stream. Using the same redundant stream, different receivers can recover from different packet losses. These techniques introduce more computational complexity and additional delays.

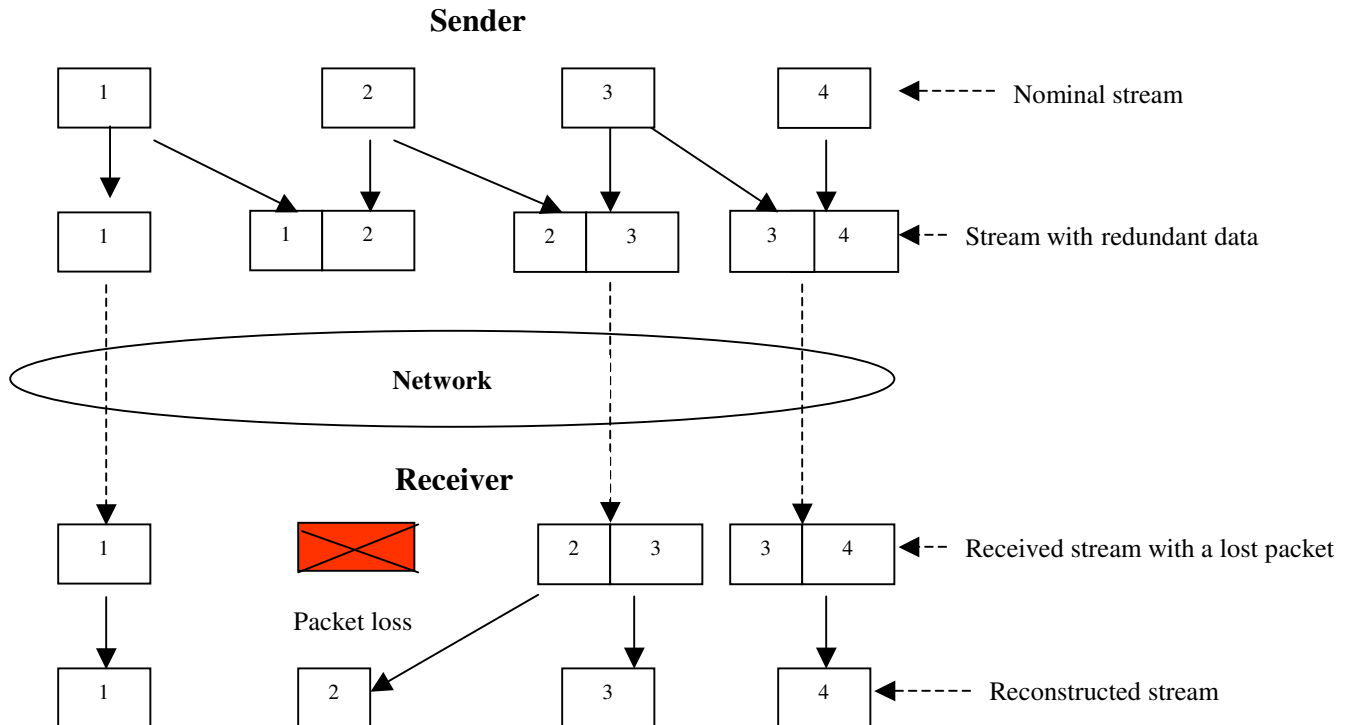


Figure 3: Forward Error Correction: An illustration

2.1.3. Interleaving

This mechanism is based on re-sequencing of the bits in a packet. By spreading the blocks that constitute a single packet over multiple packets, the perceived quality of reception can be improved. This is because when a packet is lost, instead of a single large gap, the interleaved stream results in multiple small gaps in the reconstructed stream. For example, in Figure 4 [32][29] blocks 3,7,11, and 15 are lost. However, these blocks constitute only one lost block in each of the packets (rather than losing one whole packet) the resultant quality is improved.

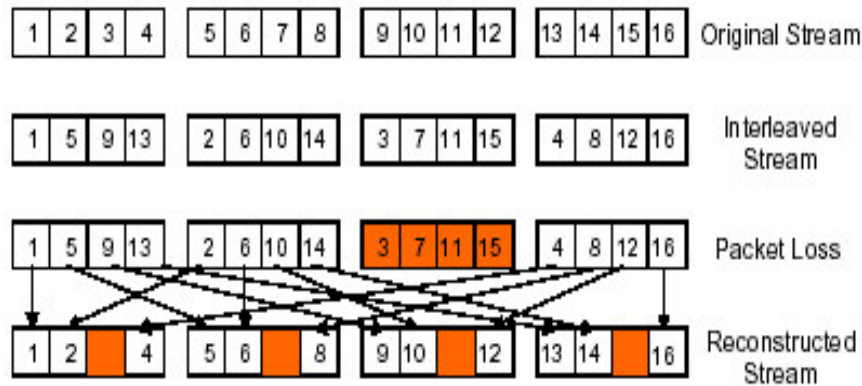


Figure 4: Interleaving: An illustration (taken from [32][29])

The main advantage of interleaving is that it does not increase the bandwidth requirement. However, it adds to the additional encoding/decoding complexity, adding to playout delays.

2.1.4. Discussion

Correction based mechanisms can be used with both synchronous and on-demand delivery of multimedia data. For example, the FEC techniques can be applied to both modes of transmission, as the original stream -- whether from a live source or from a stored medium -- can be appended with the redundant stream on the fly. Error-concealment techniques are well suited for and typically used with on-demand delivery. In the case of popular on-demand applications where several parallel streams are served, simple error-concealment mechanisms can improve QoS with very little overhead.

2.2. Resource-based mechanisms

QoS delivered to the end user depends not only on the network condition, but also on the constraints at the user's end. Thus, status of network and system resources have to be taken into account while disseminating multimedia files which are typically large in size. There are several ways in which such an objective can be achieved. One way is to use *adaptation mechanisms* wherein the source, receiver, or a transcoder¹ ensures that the resultant

¹ We refer to multimedia gateways, which implement data reduction techniques as transcoders.

reception is improved given the resource constraints. Another way is to use *reservation mechanisms*, wherein resources are reserved *a priori* to satisfy the given real-time constraints of the multimedia content. *Hybrid mechanisms*, utilizing techniques such as admission control and priority scheduling, have also been proposed. All three types of mechanisms are discussed in this section.

2.2.1. Adaptation mechanisms

The term “adaptation” refers to the fact that there is some intelligence built into the sender, receiver, or a network component using which the ideal transmission rate is chosen (by the sender, receiver, or a component such as a transcoder) for a given system and network conditions. Adaptation can be implemented by:

- Sender nodes, which collect feedback from the set of receivers they serve, categorized as *feedback-based* mechanisms.
- Receiver nodes, which use the knowledge of their resource constraints to join the appropriate group/s to receive the media file at a particular rate, categorized as *constraint-based* mechanisms.
- Network components such as transcoders which can serve a group of receivers with similar constraints, categorized as *transcoder-based* mechanisms.

There are two parts to the adaptation mechanisms based on feedback in the sender’s perspective: finding out the quality of reception at the receivers (feedback), and based on this information, taking appropriate action (adaptation). Some of the *feedback* mechanisms are discussed Section 2.2.1.1. and feedback based *adaptation* mechanisms are discussed in 2.2.1.2.

Adaptation mechanisms can also be implemented by the receivers when they are aware of their resource constraints. These *constraint based* techniques are discussed in 2.2.1.3. *Transcoder based* adaptation techniques, which use data reduction techniques to serve a group of users with similar constraints, are discussed in 2.2.1.4. The reader is referred to [50] for a detailed survey of adaptation mechanisms.

2.2.1.1. Feedback mechanisms

Feedback mechanisms are designed to provide information about the quality of reception at the receivers. These mechanisms by themselves do not implement any action based on the information. Rather they facilitate the sender to be aware of the reception quality at the receivers. Based on this, the sender can take appropriate actions to ensure acceptable quality of reception at the receivers.

The sender finds out about the quality of reception through feedback reports sent by the receivers. The receivers report several parameters such as delay, jitter, and packet loss as experienced by them. However, a major drawback of the feedback mechanisms is that when the number of receivers is large in a multicast session, each sending feedback about its quality of reception, the feedback messages may consume bandwidth causing or adding to congestion in the network This phenomenon is termed as “feedback implosion” [37]. Thus,

an important challenge in designing feedback mechanisms is to avoid feedback implosion while providing timely and accurate feedback to the sender.

Keeping the above challenge in mind, we first discuss the feedback mechanism proposed in the Real Time Control Protocol (RTCP), which is standardized as part of Real Time Transport Protocol (RTP). An alternative, based on a probability model proposed to avoid feedback implosion is also discussed.

2.2.1.1.1. RTCP

RTP [37] [10] is a standard protocol used for delivering multimedia data, prescribes two closely linked parts: (i) Real Time Transport Protocol (RTP), which provides end-to-end data delivery services, including payload type identification, sequence numbering, time stamping, and source identification. (ii) Real Time Control Protocol (RTCP), which provides the functionality to monitor the QoS and to convey information about the participants in an on-going session².

RTCP periodically distributes control packets containing quality information to all session participants through the same distribution mechanisms as the data packets. In order to avoid feedback implosion, RTCP packets utilize a small *a priori* known fraction of the session bandwidth. Session bandwidth is supplied by the session management application, when it invokes the media application. The standard recommends 5% of the session bandwidth to be reserved for control, out of which the senders and receivers are assigned 25 percent and 75 percent respectively.

The RTCP protocol works by estimating the number of participating sites in a session and uses this estimation and the average size³ of the RTCP packet to calculate the interval between two RTCP packets. Two tables, a *sender table*, with a list of senders, and a *member table*, with a list of all participants, are maintained. For details of the algorithm and the protocol, we refer the reader to [37].

Quality of reception feedback is sent by RTCP report packets, which are of two kinds: Sender Report (SR), Receiver Report (RR). Besides the packet type code, the only difference between SR and RR is that the SR includes a 20-byte sender information section. The RR contains the feedback parameters including fraction of packets lost, cumulative number of packets lost, an estimation of the inter arrival jitter of RTP data packets etc. which can be used by the sender.

RTCP deals with feedback implosion by allocating a fraction of the session bandwidth to control traffic. However, this algorithm may not scale well when the number of receivers increases. There are other mechanisms based on statistical models recommended in the literature for providing feedback. One such algorithm, which tackles the scalability problem, is discussed below.

² A multimedia session defines a set of concurrent RTP sessions among a common group of participants. Each medium (e.g. video, audio) is typically carried in a separate RTP session with its own RTCP packets.

³ $Avg_rtcp_size = \alpha * packet_size + (1 - \alpha) * Avg_rtcp_size$, where $packet_size$ is the size of the RTCP packet just received. The standard uses a value of 1/16 for α , giving more weight to the historical value.

2.2.1.1.2. A Scalable feedback mechanism

A scalable feedback control mechanism, which uses a probabilistic polling mechanism, is proposed in [6]. A series of probing messages are sent to the multicast group to elicit congestion reports from the receivers. The number of receivers who can respond in a particular round is restricted by requiring them to match a randomly generated key. The algorithm uses a random key of length 16. A receiver with a matching key responds in the following two cases: (i) Initially, when the server sends a SIZESOLICITED probe message which it uses to estimate the number of receivers, and (ii) subsequently, when the network state perceived by the receiver is worse than the current advertised state from the sender.

Given n receivers, a logarithmic relationship is shown to exist between n and the average round in which a receiver would first match the key. This allows the sender to estimate the receiver group size. Once the sender is able to come up with this estimate, the SIZESOLICITED flag is unset. Subsequently, only receivers experiencing congestion respond upon matching the random key. An estimate of the congested receivers is obtained from the number of elapsed rounds between a response to a SIZESOLICITED message, and the first response reporting congestion from a receiver.

Scalability of the algorithm arises out of the fact that the maximum discovery time of a congested receiver is independent of the number of receivers. When there are no congested receivers, the worst case congestion discovery time is given by $2^l * rtt_{max}$, where l is the length of the random key and rtt_{max} represents the worst case round trip time for a probe message.

2.2.1.2. Feedback based adaptation

Feedback mechanisms (discussed in the pervious section) provide the sender with a view of reception quality at the receivers it serves. The sender uses the feedback from the receivers to *adapt* its transmission rate to provide better QoS to its receivers [50][6][8]. Such *feedback based adaptations* are discussed in this section.

Feedback based adaptation techniques use the feedback from a receiver to place the receiver in one of three states: UNLOADED, LOADED, OR CONGESTED. The network state is determined using a threshold parameter, which gives the upper limit for the number of congested receivers. In LOADED state, the sender is sending at the maximum useful rate. When the network state is UNLOADED, the sender progressively increases its sending rate in an additive manner and when the network state is CONGESTED, the sender progressively decreases its transmission rate in a multiplicative manner. The challenges in implementing such adaptation mechanisms include choosing appropriate values for the threshold, additive rate, and multiplicative rate. The threshold value has to be chosen in such a way that a small subset of the receivers with bad reception does not cause everyone in the group to experience poorer quality.

In [6] the sender adjusts the rate between the maximum rate of the coder and a minimum rate specified by the user based on the number of congested receivers as compared to the threshold value. Here the scalable feedback mechanism discussed in 2.2.1.1.2 is used. The

dynamic QoS control mechanism proposed in [8] uses RTCP receiver reports to calculate packet loss and jitter. Appropriate action is taken by the sender based on the value of these parameters. Assuming that packet loss is induced by congestion, the algorithm defines two thresholds λ_c and λ_u , as shown in Figure 5. These thresholds are used to determine the three network states - UNLOADED, LOADED, and CONGESTED - as experienced by the receivers. The upper threshold λ_c is chosen by taking into account losses, which are acceptable, and the lower threshold λ_u is chosen to avoid QoS oscillations.

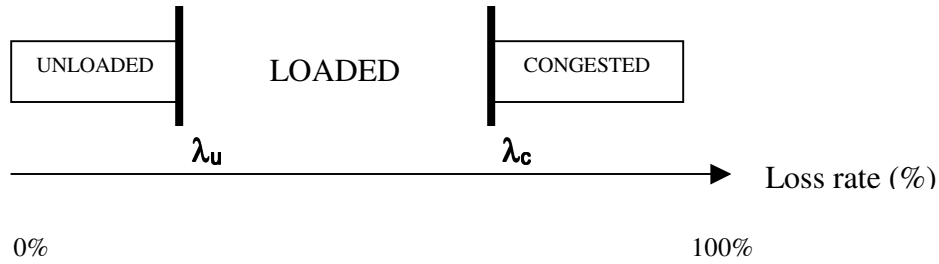


Figure 5: Sender's view of the receiver states

Based on the proportion of receivers in one of the defined states, decision to increase, decrease, or hold the bandwidth is made. For example, when $x\%$ of the receivers report packet losses (which the sender interprets as congestion), the sender reduces its rate of transmission. To guarantee minimum quality, the minimum bandwidth is set by the user. If this minimum rate itself causes packet loss, the receiver should detach from the session.

In the Loss-Delay based Adjustment Algorithm (LDA) proposed in [42], an additional parameter, *bottleneck bandwidth*, computed using the round-trip delay from the RTCP receiver report, is used for determining the additive rate increase parameter. The LDA algorithm adjusts the adaptation parameters dynamically in response to the packet loss, delay, and capacity of the connection.

Adaptation techniques discussed thus far are initiated by the sender based on the feedback from the receivers. There are situations when the receivers participating in a multicast session have different constraints. In such cases, it is better for the receivers to adapt to transmission at an acceptable rate based on their constraints. These receiver initiated adaptation techniques are discussed in the next section.

2.2.1.3. Constraint-based adaptation

We define a group of receivers with similar constraints on network and/or system resources as *homogeneous* and a group with different constraints on network and/or system resources as *heterogeneous*. Heterogeneous groups make it difficult for the feedback-based schemes to determine a single optimal transmission rate. When receivers are aware of their constraints, it is better to allow them to tune to the level of transmission individually according to their needs and capabilities. Such adaptation techniques initiated by the receivers [50] are termed *constraint-based adaptation*. A simple technique that falls under this category is to use the *simulcast* scheme [29] where the source encodes and transmits multiple copies of the media data. Each copy is encoded to provide different levels of QoS

and sent to a different multicast group. While this technique is easy to implement, due to its inefficient use of bandwidth it is generally not recommended for transmission of bandwidth intensive multimedia files.

A combination of layered encoding and layered transmission, where the source data is encoded into a number of layers, is the more commonly used mechanism with multimedia data [50]. A base layer provides the minimal QoS needed for an acceptable representation of the original data stream. Higher layers can be incrementally combined with the base layer to get higher QoS. Each of the layers is transmitted through a separate multicast group. This mechanism is referred to as *cumulative layering*. Several schemes such as RLM [30] and *Thinstreams* adaptation [53] use this mechanism. In these schemes the receiver is responsible for monitoring the network condition and adapt by adding or dropping layers.

While the above mentioned schemes are completely receiver-driven (where the sender is providing different static encoded layers), there are other schemes where the sender rate adaptation based on feedback is used in conjunction with the receiver adaptation based on constraints at the receiver's end. The Adaptive Layered Transmission (ALT) [41] and the adaptation schemes proposed in [47] along with the credit-based AMML scheme, fall under this category. In these hybrid mechanisms, the sender uses the feedback to dynamically change the encoding of the layers to provide better options for the receivers. However, these mechanisms introduce additional complexities at the senders.

Another option for receiver-initiated adaptation is provided by transcoder based adaptation discussed in the next section.

2.2.1.4. Transcoder based adaptation

Transcoder based adaptation techniques are those where a multimedia gateway (transcoder) implements adaptation for a group of receivers with similar constraints. Typically, the gateways are placed at appropriate locations in the network to convert a high bandwidth transmission into appropriate bandwidth transmission for the benefit of a group of users with constraints on bandwidth. Receivers may initially start with a particular gateway and may be allowed to adapt to network state dynamically by choosing a different gateway. Gateways can also be designed to use an adaptive rate-control algorithm to adjust their transmission rate in response to receiver feedback.

Transcoding techniques are used to serve the same underlying multimedia object at different quality levels to different users based on the users' operating constraints. The main effect of transcoding is data reduction, which helps to reduce network end-to-end delay, loss, and delay jitter for the multimedia packets. Transcoding can be done in a number of ways [43][24] including *Spatial transcoding* (to reduce the frame size), *Temporal transcoding* (to decrease the frame rate by dropping less significant frames), *Color transcoding* (to reduce the data size by decreasing the color depth). Design of the transcoding algorithm, and the placement of the gateway are important issues in transcoder based adaptation schemes. Detailed treatment of transcoder based adaptation is beyond the scope of this paper.

In the model proposed in [3] the input format is first converted into an intermediate format. The intermediate format is then encoded to produce new streams. By making provision for multiple intermediate formats, a number of encoder/decoder combinations can be supported. A scheme for automatic configuration of transcoders to support a group of receivers in a multicast group is provided in [22]. By deploying transcoders, the burden of encoding is shifted from the senders to transcoders. Also, by placing the transcoders at appropriate locations in the network, receivers with similar constraints can be served better.

2.2.1.5. Discussion

In summary, feedback based adaptation techniques are best suited for synchronous multicasting of multimedia to a set of homogeneous (with similar connectivity and system resources) receivers. These mechanisms are especially suitable for synchronous dissemination of multimedia contents because a single source can take appropriate action based on the feedback from its receivers. These mechanisms generally do not scale well for the on-demand mode of transmission. Feedback based adaptation techniques can be combined with constraint based techniques, where the sender encodes layers based on the feedback from the receivers while the receivers themselves join an appropriate layer based on their constraint. Transcoders can also implement feedback mechanisms to adapt the transcoding to suit the set of receivers they serve.

When the group of receivers is heterogeneous (with different connectivity and system resources), the constraint based and transcoder based techniques are more appropriate. These mechanisms are suitable for both synchronous and on-demand modes of transmission. Constraint based and transcoder adaptation techniques scale well as the decision to adapt is left to the receiver. Also, unlike the feedback-based adaptation techniques where quality of reception is reduced across all receivers to cater to the need of a sizable proportion of receivers facing congestion, constraint based techniques are applied individually. In this sense, these techniques are fair. Transcoder based mechanisms can be used as an alternative to the cumulative layering schemes discussed in 2.2.1.3. when receivers with similar constraints are clustered in geographically diverse locations.

2.2.2. Reservation mechanisms

The main idea behind reservation-based techniques is that QoS can be guaranteed if the required resources are pre-allocated to the application. These mechanisms work on the premise that a multimedia flow can be denied admission if its resource requirements cannot be met by the network. Reservation mechanisms can provide bounds on end-to-end delays thus guaranteeing a particular level of QoS. However, the network resources may be under-utilized when an application with reserved bandwidth has less or no data to send.

Integrated Services (Intserv)[40][52] is an architecture developed by the Internet Engineering Task Force (IETF) to provide individualized QoS guarantees to application flows. The Resource Reservation Protocol (RSVP) [7][10] is a standardized protocol for QoS based on reservation of resources on a per flow basis, which is used in the Intserv architecture. Differentiated Services (Diffserv) [5] was proposed by the Internet Engineering Task Force (IETF) to provide QoS to aggregated flows. A brief overview of

the Intserv and Diffserv architectures are provided in Sections 2.2.2.1 and 2.2.2.2 respectively. The working of the RSVP protocol is outlined in Section 2.2.2.1.1.

2.2.2.1 Intserv

Intserv, proposed in RFC 2215, is aimed at providing the required QoS based on the applications' needs. The basic characteristics of the architecture can be summarized as follows:

1. It is based on per flow reservation.
2. It requires that the applications specify their traffic and resource requirements and make reservations before starting the traffic.
3. It uses a reservation protocol to reserve resources and install reservation state along a path determined using the routing protocol.
4. It uses an admission control mechanism to ensure availability of resources at every hop before admitting a new flow.
5. It enforces resource reservation through classification of packets and scheduling mechanisms.

The architecture provides two types of services: guaranteed service, where the resources as specified by the application are allocated so that hard guarantees can be given, and Controlled Load Service (CLS), which provisions for the required resources, but does not guarantee QoS. CLS provides services, which mimics the service the application would get in a lightly loaded best effort network. Intserv uses RSVP protocol to reserve resources on a per flow basis, where per flow states are maintained at every intermediate router on the path between the source and destination. While this architecture allows for hard guarantees, it has the inherent problems of scalability. To understand the reservation mechanism used by Intserv we discuss the salient features of the reservation protocol RSVP.

2.2.2.1.1. RSVP

RSVP [7] [10] is a network control protocol that allows applications to obtain special QoS for their data flows, by reserving resources along the paths. When an application in a host requests a specific QoS for its data stream, RSVP is used to deliver the request to each router along the path of the data stream and to maintain router and host state to provide the requested service. The two principal characteristics of RSVP are:

1. It provides reservations for bandwidth in multicast trees (unicast is handled as a trivial case of multicast)
2. It requires the receivers of data flows to initiate and maintain resource reservations.

At each node, RSVP applies a local decision procedure (admission control) to the QoS request. If admission control succeeds, it sets the parameters to the packet classifier and packet scheduler to obtain the desired QoS. If admission control fails at any node, RSVP returns an error indication to the application. Even though RSVP reserves resources for simplex data streams, i.e., it reserves resources in only one direction on a link, the same application may act as both sender and receiver. For dynamic adaptability and robustness,

RSVP maintains *soft state*⁴ in the routers. For details of RSVP the reader is referred to RFC 2205 and [29].

2.2.2.2. Diffserv

Diffserv was proposed as an alternative to Intserv to address the scalability issue. The main characteristics of Diffserv are:

1. It divides traffic into small number of classes and allocates resources on a per class basis.
2. It requires packet headers to be marked with a classification, which is used by the router for providing a particular forwarding treatment termed Per Hop Behavior (PHB).
3. It provides differential treatments to classes (aggregated flows) based on packet marking done on the basis of Service Level Agreements (SLAs) between the customer and the service provider.

Diffserv architecture consists of two sets of functional elements:

1. *Edge functions* (Packet classification and traffic conditioning): The arriving packets are marked at the “edge” of the network, i.e., either at the Diffserv enabled host generating the packet or at the first Diffserv capable router that the traffic passes through.
2. *Core function* (Forwarding the packet): Based on the packet marking, a Diffserv capable router forwards the packet onto its next hop according to the per-hop behavior associated with that packet’s class.

While the Intserv and Diffserv architectures are based on different admission control and reservation techniques, there are also admission control mechanisms suggested specifically for multimedia applications. We briefly discuss the idea behind such techniques, which dynamically adjust the reservation for their flows.

2.2.2.3. Reservation adaptation

Multimedia applications are sensitive to delay jitter while they can tolerate occasional loss of data. These applications also need a minimum guaranteed QoS below which reception is not decipherable and hence not useful. Thus, it is possible to specify a minimum QoS function as well as the desired QoS the application needs. Based on these parameters admission control algorithms are developed to ensure that at least the minimum quality is provided to the existing flows while admitting a new flow. We categorize these mechanisms that dynamically adapt reserved resources under the *reservation adaptation* mechanisms. The main idea underlying these mechanisms is to maximize resource utilization while guaranteeing proper reception for all the existing applications.

An algorithm for resource reallocation when a source specifies a range for its resource requirement is presented in [35]. The applications specify (QoS min, QoS max). Initially, when there are few competing flows, applications are allocated the maximum resources

⁴ Reservation states are refreshed periodically by RSVP; when not refreshed, expiry of a timer associated with a state deletes it automatically.

required. When the resources become constrained, allocations of the existing flows are adjusted such that their minimum requirements are met. After the adjustment, if the pooled resources satisfy at least the minimum requirement of the new flow, it is admitted. Any additional bandwidth after the new flow is admitted is shared across all the flows ensuring that the system as a whole operates with the best possible resource allocation. This algorithm ensures that resources are appropriately utilized considering the tolerance that multimedia applications exhibit. However, additional computational complexity is introduced in the CAC algorithm.

2.2.2.4. Discussion

The advantage of Intserv architecture is that the architecture can support guaranteed QoS. However, per flow reservation may lead to sub-optimal use of resources and scalability problems. The Diffserv architecture, where aggregated flows belonging to specific classes are treated with specific forwarding behavior, provides a scalable mechanism. Also, Diffserv does not define specific services or service classes like Intserv. This allows for flexibility in the service models where relative and qualitative service distinctions can be built.

Intserv and Diffserv define specific architectures which fulfill the twin requirements of resource assurance and service differentiation in a network to provide QoS to applications. In the case of multimedia applications, rather than allocating or provisioning a specific amount of resources, the reserved resources can be adjusted to ensure acceptable QoS while admitting new application flows. Reservation mechanisms that dynamically adapt based on the availability of resources work well for multimedia applications.

2.2.3. Hybrid mechanisms

From the above discussions on the Intserv and Diffserv architectures, it is clear that each has its limitations. Intserv can provide guaranteed QoS at the cost of scalability issues while Diffserv provides flexible service model and scalability at the cost of no guarantee on QoS. Multimedia traffic is sensitive to delay to the extent that packets arrive at the receiver in time to be played back. However, multimedia applications generally tolerate a small number of lost packets. Given these characteristics of multimedia data, hybrid architectures, which combine the features of the Intserv and Diffserv, can serve the requirements of multimedia dissemination.

One option is to provide RSVP based guaranteed resources to applications in the Intranets (provisioned with enough bandwidth), which are connected to a Diffserv enabled backbone providing quality assurances for aggregated traffic [10]. In such architectures the border gateways between the Intserv network and the Diffserv backbone need mechanisms to map the flows to different aggregated classes. Reverse mechanisms are needed when the flows enter the Intserv network from the Diffserv backbone.

Another hybrid architecture, proposed in [24] uses the concepts of priority queuing and RSVP to offer the features of Intserv and Diffserv. The priority queuing mechanism gives multimedia traffic the highest priority over other traffic. The protocol considers two

scenarios when QoS can suffer: (i) When the network is congested (ii) End user is constrained by connectivity or system capability. The paper proposes use of transcoding to adapt the rate of transmission when either of the two scenarios is encountered. Transcoding is performed to avoid network congestion and cater to low bandwidth or to suit end users with specific support as regards to processing power and/or display resolutions. QoS is provided by using a combination of reservation, queuing, routing, and congestion avoidance mechanisms along with transcoding. Statistical information about the load and network performance at different times may be used to trigger transcoding. A signaling protocol to get information about the network load may also be used as a trigger. This architecture emulates features of Intserv and diffserv by combining well-known protocols at the different network layers.

2.2.3.1 Discussion

We have provided instances from the literature where different combinations of protocols and architectures are used for effective delivery. Considering the characteristics of multimedia applications and the factors that influence their dissemination, a thorough study of such hybrid mechanisms is warranted. This appears to be an important area for further investigation. We discuss some of the research issues in this area in Section 4.

In this section we discussed the resource-based mechanisms -- classified as adaptation mechanisms, reservation mechanisms, and hybrid mechanisms -- for effective delivery of multimedia. Adaptation mechanisms can be: (i) Implemented at the sender, based on feedback from the receivers, (ii) Facilitated by the sender and executed by the receivers, or (iii) Implemented by transcoders and joined by receivers. Reservation mechanisms are capable of providing prioritized services with or without guarantees for QoS. Guaranteeing QoS involves setting aside resources, which contributes to underutilization of resources. Mechanisms which provide service assurance than absolute guarantee, are better suited for multimedia applications. However, given that within an Intranet it is feasible to provide guaranteed service, hybrid architectures that combine various mechanisms proposed for QoS at different levels can be deployed for effective dissemination of multimedia.

A summary of the key points discussed in this section is provided in Table 1.

Effective Delivery Mechanisms	Key Points
<p>Correction-based –(Error concealment, FEC)</p>	Suitable for both modes of transmission (Simple concealment techniques especially suitable for on-demand mode)
<p>Resource-based</p> <ul style="list-style-type: none"> – Adaptation <ul style="list-style-type: none"> • Feedback-based • Constraint-based • Transcoder-based 	<p>Suitable for synchronous mode (both live and stored) and homogeneous receivers</p> <p>Suitable for both stored synchronous and on-demand modes of transmission and heterogeneous receivers</p>
<p>–Reservation (Intserv, Diffserv)</p>	Can be used with both modes, when link is not dedicated. Reservation adaptation best-suited for multimedia.
<p>– Hybrid</p>	Combing mechanisms (Reservation Adaptation); Resource assurance works well for multimedia.

Table 1: Summary of effective delivery mechanisms

3. Mechanisms for efficient delivery of multimedia

We say that a multimedia delivery mechanism is efficient if it ensures optimal use of network and system resources, allowing for scalability. There are many ways in which efficient dissemination can be effected. One way would be to store contents close to the receivers so that the access path is shorter which in turn reduces the time taken for retrieval of contents. We classify the techniques which use this idea as *storage-based* mechanisms. These mechanisms are discussed in Section 3.1. *Caching* and *Content Distribution Networks (CDNs)*, which are widely used storage mechanisms, are discussed in Sections 3.1.1 and 3.1.2 respectively.

We categorize mechanisms that are designed to service user requests with minimal latency as *service-based* mechanisms and discuss them in Section 3.2. These mechanisms are relevant only for on-demand delivery of multimedia. We classify service-based mechanisms into *stream merging* mechanisms and *navigation* mechanisms. When multiple requests for a particular media file are spread across a short interval of time, mechanisms are needed to optimally serve them while ensuring minimal play back delay. Techniques that serve this purpose are categorized under *stream merging mechanisms* and discussed in Section 3.2.1. The numerous stream merging techniques proposed in the literature are further classified into *periodic broadcast*, *patching*, and *hierarchical stream merging*. Popular multimedia applications such as on-demand streaming of movies allow the users to navigate through the media files. Users can interact with the application using VCR like pause, rewind, etc. buttons. System response time to such requests have to be within a few seconds. Techniques that aid or enhance the response times are categorized under *navigation* mechanisms and discussed in Section 3.2.2.

We refer the reader to [29] for a detailed overview of issues and trends related to on-demand streaming. Figure 6 illustrates the classification of mechanisms for efficient delivery.

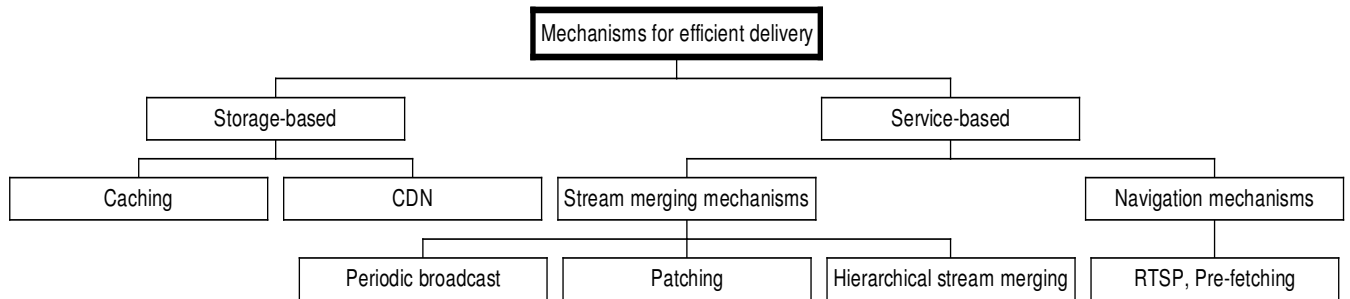


Figure 6: Classification of mechanisms for efficient delivery

3.1. Storage-based mechanisms

Bringing contents closer to the receivers in order to reduce retrieval latencies and network congestion is the key idea behind the storage-based mechanisms. These mechanisms can be classified broadly into the following two categories: *Caching* techniques and *Content Delivery Networks* (CDNs), which are discussed in this section.

3.1.1 Caching

Caching can be defined as a process by which frequently accessed data is kept locally, rather than constantly being accessed from the place where it is stored. In the context of a network, caching generally refers to a proxy-based architecture where proxy servers are deployed in different domains to serve clients in that domain. Two important factors have to be considered when caching is used for streaming multimedia: (i) The media play back has to begin within few seconds after the user’s request. (ii) Once the play back begins, it has to be continuous. In addition, caching mechanisms also have to take into account some of the unique properties of multimedia data:

- *Large size of media files:* Multimedia files are voluminous due to their high data rate and long playback duration. However, unlike web objects, media files can be segmented and individual segments can be cached. Caching some of the segments means that a client may need to access some segments directly from the server also. The question here is how one decides on which segments to place in the cache? Many algorithms [51][38][20] have been proposed for partial caching of media objects keeping the two factors (immediate and continuous playback) in mind.

These algorithms also need to ensure that the benefits of caching override the synchronization overhead between the cache and server.

- *High bandwidth requirement:* Bandwidth poses the major constraint for streaming multimedia files. A cache server may be constrained by the bottleneck bandwidth in serving clients. Use of multicast delivery in conjunction with caching [15][44], and cooperation among proxies (peer-to-peer architectures) [17][14] are ways in which the bandwidth constraint can be tackled.
- *Navigation capability through the media file:* During the typically long duration of media playback, users are allowed to interact with the application to pause, fast-forward, or rewind. Thus, access rates for different parts of the media file may be different. This introduces complexity in the cache management techniques, which need to capture the effects of the VCR like functions on the cache's contents.

For a survey of caching techniques for multimedia streaming, refer to [49]. Appropriately placed and well-managed caches can alleviate network congestion and allow more users access to multimedia files. Deploying proxies for caching multimedia contents increases the overall system capacity. However capacity is still limited by the aggregate resources of the proxies. Maintaining proxies also introduces additional deployment and management costs. Several peer-to-peer architectures, where the end-systems cooperate and share some of their resources with the proxy servers have been proposed [17][14] These mechanisms aim at increasing the over-all capacity of the system while utilizing available resources in an optimal way. Detailed discussion of these algorithms is beyond the scope of this paper.

Maintenance of proxies and cache management introduce significant overheads and need constant monitoring. Mechanisms such as CDNs are deployed increasingly where the burden is shifted to a third party service provider, who charges for these services. A brief discussion on CDNs is provided in the following section.

3.1.2. Content Distribution Networks

An alternative to caching is to employ a third-party for delivering the contents to clients. This approach termed *Content Distribution Network* (CDN) is popularized by service providers like Akamai [2]. Thousands of servers are deployed by these service providers at the Points of Presence (PoPs) of major Inetnet Service Providers (ISPs). These servers, which store the contents, are also referred to as "caches". CDNs typically use proprietary protocols to monitor the Internet traffic, direct user requests to appropriate servers, and distribute the contents. This approach reduces the load on the backbone network, which results in a better service in terms of shorter delay and smaller loss rate [17]. However, as the CDN operator charges the content provider for every megabyte served, cost-effectiveness is a major concern in this approach.

Figure 7 illustrates caching and CDN approaches. In caching, the local cache servers serve the requests from clients; however, when the cache does not contain the requested object, it will fetch it from the media server. In the CDN, a set of cooperating servers is deployed on the backbone. Requests are processed by the appropriate server.

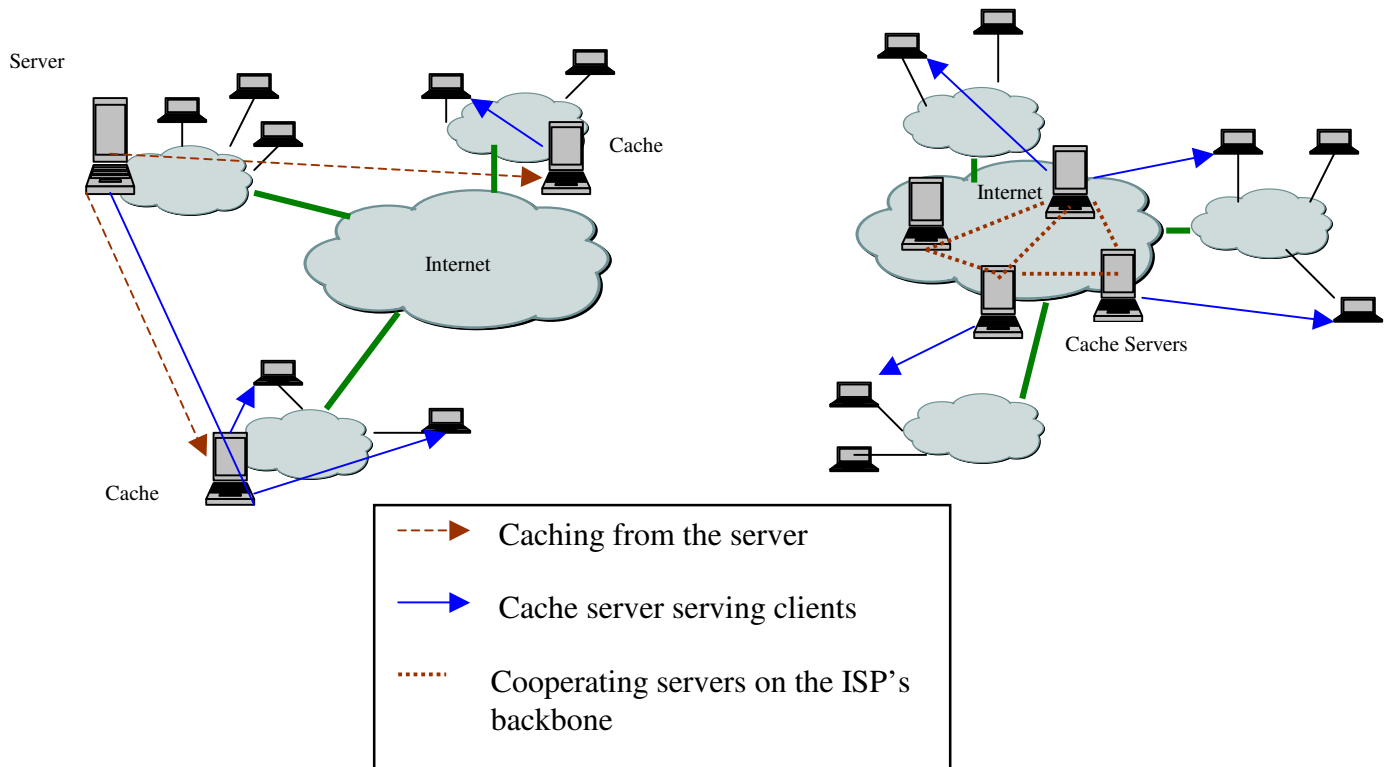


Figure 7: Caching vs. CDN

3.1.3. Discussion

When multimedia contents to be served are static over a period of time, cache management becomes simpler. Highly dynamic contents need constant monitoring and complex cache management strategies, which can be better handled by a CDN. When the receivers are clustered in different geographical regions deploying proxy servers to serve these clusters provide a scalable solution. On the other hand, when the receivers are scattered across the globe, paying for the services of a CDN makes more sense. Caching and CDNs can be deployed for both synchronous mode of transmission of stored contents as well as on-demand mode of transmission. None of the mechanisms discussed in this section are applicable to live streaming.

3.2. Service-based mechanisms

On-demand mode of transmission requires efficient service mechanisms for: (i) Serving the multimedia file upon request from any of the receivers at any time. (ii) Providing responses with minimum latencies when the receiver issues any navigation commands. We define *service-based* mechanism as those that are designed to service user requests with minimal latency. We define the following additional parameters, which are important in ensuring efficient service:

- **Start up latency:** This is time lag between the time when a file is requested and the time when the file is played.
- **Navigation latency:** This is the time taken by the media player application to respond to receiver requests such as fast-forward, rewind, etc., to navigate through the multimedia file.

Both start-up latency and navigation latency have to be minimized to best serve the receivers. When multiple requests for a particular media file are spread across a short interval of time, mechanisms are needed to optimally serve them while ensuring minimal start up latency. Techniques that serve this purpose are categorized under stream merging mechanisms and discussed in Section 3.2.1. Navigation techniques, which minimize response times to user interaction requests, are discussed in Section 3.2.2.

3.2.1. Stream merging techniques

An important challenge in delivering popular multimedia files to a large number of clients is to optimize the use of resources at the client, server, and network while ensuring the smallest possible start-up latency. In the simplistic case, the server can allocate separate channels for each of the clients. This approach will quickly exhaust the server bandwidth when the demand is high. Various techniques where client requests are batched so that the aggregated requests are served together are proposed in the literature. Serving multiple clients with a single stream is the key idea of this aggregation class of techniques, which includes Staggered broadcasting [11] and Adaptive piggybacking [16]. These techniques are not scalable when the demand for the media files is high.

Other optimization techniques assume several *server channels*⁵. Several techniques, which are based on merging the streams from multiple channels, have been proposed. An overview of these techniques can be found in [29]. These techniques optimize the resource requirements for media playback such that the quality factors are considered while serving multiple clients. The numerous stream-merging techniques proposed in the literature are categorized into three main classes: periodic broadcasts, patching, and hierarchical stream merging. These three classes of techniques are briefly discussed in the following sections.

3.2.1.1. Periodic broadcasts

In Periodic broadcasts the media file is divided into N segments with relative lengths S_1, S_2, \dots, S_n . (increasing in size) which are broadcast periodically using a pre-determined schedule. Clients can receive multiple segments concurrently, at an aggregate rate that exceeds the media playback rate. Data thus received ahead of time is stored at the client for subsequent playback. Variations of Periodic broadcast techniques can be found in [48][1][18][21]. These techniques optimize on different parameters, for example, trading off between buffer requirement at the client and the number of concurrent channels needed for smooth playback. We have summarized the key aspects of some of the periodic broadcast techniques in Table 2.

⁵ The I/O and network bandwidths required at the server to deliver one video stream in order to guarantee continuous playback at a client is defined as a server channel.

Technique	Number of segments	Buffer space requirement	Start up latency	Concurrent reception	Broadcast rate
Pyramid broadcast	N segments, increasing geometrically in size	High	Low	Two channels	Higher than media play back rate
Permutation-based pyramid broadcast	N segments, P sub segments	50% of the media file	Low	Single channel	Media play back rate
Skyscraper broadcasting	N segments – relative sizes 1, 2, 2, 5, 5, 12, 12,.....	L_n , length of the n th segment	$T / \sum L_n$	Two channels	Media play back rate
Harmonic broadcasting	Segments of equal size	Relatively higher; Out of order sequence of segments possible	Small	All channels	$1/k$ times media play back rate for segment k

Table 2: Summary of Periodic Broadcast (PB) protocols

Several enhancements have been proposed to the simple Periodic Broadcast (PB) protocols. Optimized PB protocols are proposed in [28], which are optimized under the constraint that clients receive each segment entirely before playing the segment. Optimized PB protocols are extended in [28] to encode the transmitted data using erasure codes. This enables recovery from packet loss when the loss is less than a tunable parameter p . In this reliable class of PB protocols, using the erasure code, each client reconstructs segments prior to the time the segment needs to be played.

3.2.1.2. Patching

The motivation for this technique [19][39] is to minimize the bandwidth requirement for multicasting a media file, when the client requests are spread across an interval of time. When a client joins a multicast at a time later than the start time for the multicast, the client stores the on going (later portions) stream, while the server sends a unicast “*patch*” for the missing initial part of the file. The patch stream is terminated when the initial missed portion of the media file is served to the client.

Greedy patching [9][19] and Grace patching [19] are variations of this technique. In greedy patching, a client is allowed to join the multicast at any point in time. In grace patching, if client request comes very late into the multicast session, a separate stream is started. It is shown that Grace patching can yield better results as the new stream may be used by a subsequent request for patching, more efficiently [19]. Generally patching schemes need

large buffers. The buffer requirement for greedy patching is especially high which needs to be in the order of the largest media file the client would request.

3.2.1.3. Hierarchical stream merging

In hierarchical stream merging protocols, when a client requests for a media file, a new multicast transmission is started. However, the client listens to two channels simultaneously – to its own transmission and to the most recently started transmission (for another client) before its own. The client's own stream terminates when it delivers the data missed by the earlier stream that the client is listening to. The merged clients start listening to the next most recently initiated stream (if any) with which they can merge. Thus, in this class of techniques, more and more streams are merged and served by a single transmission. Bandwidth skimming [13][28] and partitioning [28] techniques are categorized under this class. A family of reliable bandwidth skimming protocols is proposed in [28].

3.2.1.4. Discussion

Stream merging techniques optimize the network and system resources when many requests are received for the same multimedia file within a short interval of time. Such scenarios arise when popular multimedia files are requested on demand by multiple users. When several requests within a short interval of each other are eventually merged, this converges into a synchronous delivery mode with a set of receivers served by a single sender. Now receivers can send their feedback reports to the sender, which can take appropriate action. Thus, effective delivery techniques such as adaptation based on feedback can be combined with stream merging techniques. The questions that need to be answered to implement such a combination include:

- How do we choose the interval over which the streams are merged (the point where the synchronous model emerges)?
- How do the receivers know when to start sending their feedback reports?

In the next section, we discuss the mechanisms that provide user navigation through the media files and mechanisms that ensure quick responses to such user requests.

3.2.2 Navigation mechanisms

Most media player applications provide the functionality for the receiver to navigate through the media files during playback. When a receiver issues a command such as pause, rewind, etc., he/she expects the player to implement these actions with minimal latencies. We categorize the mechanisms that provide users with navigational abilities and those that optimize the response time to user requests, as navigation mechanisms. In this section we outline the Real Time Streaming Protocol (RTSP) for exchanging play back control information between the media player and the server, and some pre-fetching mechanisms deployed to reduce response latencies.

3.2.2.1. RTSP

Real-Time Session Protocol (RTSP) [36] is a control protocol for initiating and directing streaming of multimedia from media servers. It allows the media player to control the transmission of the media stream. The main functionality of RTSP is to coordinate the delivery of media objects and enable a rich set of controls. The media stream (typically, a RTP packet wrapped in UDP) uses a different port than the RTSP control messages. In other words, the data and control channels are separate (*out-of-band signaling*).

In proxy based streaming (discussed in Section 3.1.1), the RTSP control messages which enable interaction such a pause, rewind, play, etc., have to pass between the client and server through the proxy. When only a part of the file is stored in the proxy's cache, the proxy has to start the stream (when the RTSP PLAY message is received) while fetching the missing segments from the server [49]. RTSP provides a RANGE request using which such fetching can be achieved. Details of this standard can be found in [36].

3.2.2.2. Pre-fetching

Pre-fetching is the process by which a server predicts the documents that the user might visit in the near future and caches them. A pre-fetching technique based on analysis of historic data is proposed in [20]. This technique reduces the retrieval latency (when users request VCR-like functions) using past user behavior to determine which segments should be pre-fetched, in anticipation of the user interaction. The main difficulty in analyzing user behavior is that some user interactions are random and unpredictable. These random browsing behaviors are eliminated using a playtime threshold, which sets a minimum playtime below which the user action is considered random behavior. Having recorded the user behavior, and pruned the random browse patterns, association rule mining techniques [20] are used to determine the set of media segments that should be pre-fetched prior to an active session.

Another algorithm that considers the time-varying behavior (TVB) of users to pre-fetch the appropriate segments is proposed in [26]. The algorithm maps the viewing schedules and preferences of different demographic groups such as children, college students, elderly, etc., to different parts of the day. In each part of the day, the most popular videos associated with each of the groups are pre-fetched first. The authors also show that by combining the TVB-aware pre-fetch algorithm and cache replacement algorithms, retrieval latencies can be significantly reduced.

Capturing user behavior and mapping that to develop efficient pre-fetching and cache replacement techniques are still being worked on. A short discussion on the topic follows.

3.2.2.3. Discussion

Media player applications provide user-friendly interfaces to navigate through multimedia files. However, response times to user requests still have scope for improvement, especially when the traffic is heavy. Intelligent pre-fetching techniques can alleviate the problem to certain extent. An important requirement for pre-fetching is that the segment (predicted to be requested) should be fetched before its play out time. Thus, when an active stream starts

playing the requested segment, the proxy sends pre-fetching requests for the next video segments that are likely to be requested. Generally, the effectiveness of a pre-fetching approach depends on the following factors:

- Available bandwidth
- Time of pre-fetching
- Confidence and support of the rules used for predicting user behavior
- Cache size at the proxy server.

When the group of receivers is heterogeneous (with different connectivity and system resources) and the receivers use the navigational commands during playback, optimizing on resources while ensuring effective delivery is a challenge. A combination of layered encoding (to serve the heterogeneous receivers) and caching based on access probabilities of various portions of the multimedia file is proposed in [27] to achieve significant transmission cost reduction and effective delivery.

In this section we discussed the storage based and service based techniques to provide efficient delivery of multimedia. By placing cache servers at appropriate locations, applying user behavior aware pre-fetching and cache replacement techniques, and implementing appropriate stream merging techniques, multiple users requesting the same media files can be served efficiently. Here again, the combinations of mechanisms should be carefully chosen and validated to ensure efficient service with best possible use of resources. A summary of the key points discussed in this section is provided in Table 3.

Efficient Delivery Mechanisms	Key Points
Storage-based	Suitable for stored synchronous and on-demand modes of delivery
–Caching	Suitable for Static content sand clustered receivers
– CDN	Suitable for dynamic contents and scattered receivers
Service-based	Relevant only for On-demand mode.
– Stream-merging	Effective delivery mechanisms (E.g.,Feedback based adaptation) Can be used when streams are merged.
– Navigation	Knowledge of user-behaviour needed for efficient pre-fetching techniques.

Table 3: Summary of efficient delivery mechanisms

4. Extending delivery mechanisms to heterogeneous networks

The delivery mechanisms discussed in the previous sections assumed the Internet as the underlying network. Heterogeneous networks comprising of different types of interconnected networks are increasingly being deployed, especially for multimedia intensive applications such as distance education. Such networks typically exhibit the following features:

- Some parts of the heterogeneous network have dedicated resources while other parts may have to share resources. For example, nodes in a satellite network may experience no congestion if the bandwidth is pre-assigned and dedicated, while nodes connected through the Internet have to contend for resources and may have different connectivity constraints.
- The network link characteristics may be different for different parts of the heterogeneous network. For example, a node connected through a dedicated Leased Line (LL) may still experience packet losses if the channel is error prone.

We define a heterogeneous network as follows:

- S is the source that originates the media stream and is the *root* of the dissemination tree.
- S can be directly connected to clients $\{C_{11}, C_{12}, \dots, C_{1k}\}$ which are *leaf nodes* or it can be connected to *relay nodes* $\{S_1, S_2, \dots, S_m\}$ which are root nodes of *sub-trees*. These relay nodes act like sources to nodes in the sub-trees.
- Each of the sub-trees can have a structure similar to the tree based at source S.
- A directly connected node, which serves as source to a node is its *parent*. All the intermediate nodes between a leaf node and S are *ancestors* to that leaf node. These are also its *upstream* nodes. All nodes below a node are its *downstream* nodes.

Figure 8 illustrates the heterogeneous architecture represented as a hierarchical structure. Node S is the source originating the multimedia transmission. Clients C_{11}, \dots, C_{1n} are directly connected to S through a LAN. Nodes S_1, \dots, S_n are connected to S through Internet, satellite, or leased line (LL). Each of these nodes acts as source to the nodes connected to them either directly through a LAN or through Internet, satellite, or LL links. In other words, the hierarchical tree structure based at S can be repeated at each of the nodes S_1, \dots, S_n .

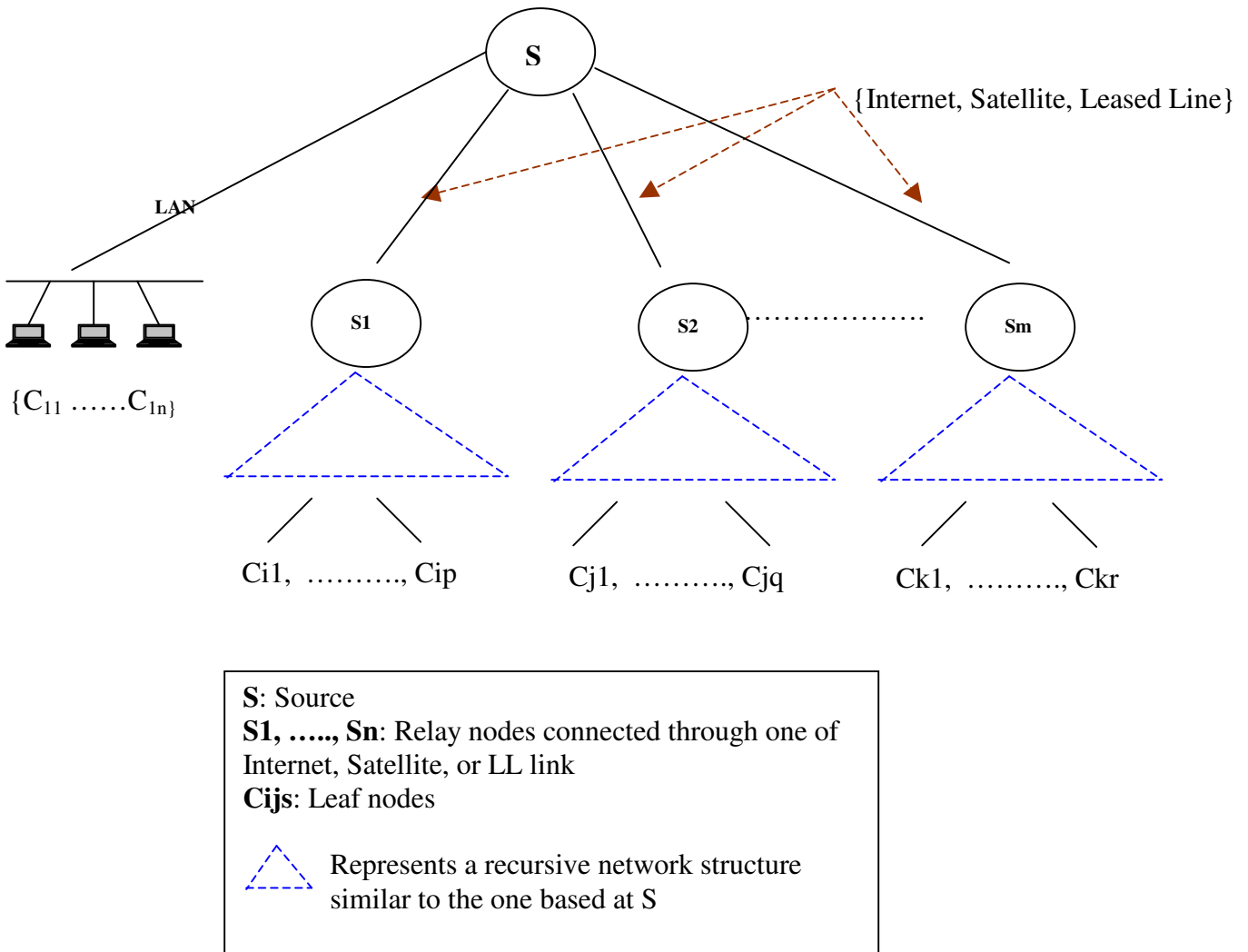


Figure 8: Hierarchical representation of a heterogeneous architecture

Using the hierarchical representation we have decomposed the heterogeneous architecture into sub-trees each having a structure similar to the original tree. This representation allows us to discuss the applicability of the mechanisms for multimedia delivery for the tree rooted at S and *recursively* apply the same logic to the sub-trees.

The ultimate goal is to provide the best possible quality of reception to the end users irrespective of the nature of their physical connectivity, while ensuring optimal use of system and network resources. In this section, preliminary mapping of the discussed protocols to the heterogeneous architecture are briefly outlined. In Section 4.1, suitability of the mechanisms for effective delivery is explored. Section 4.2 deals with mechanisms for efficient delivery of multimedia data. Case study of a heterogeneous network deployed for distance education is discussed in section 4.3. In depth exploration of applicability of the techniques and validating their effectiveness in efficiently disseminating multimedia data across a heterogeneous network are the goals of the on going work.

4.1. Applicability of mechanisms for effective delivery

We classified mechanisms for effective delivery in Section 2 (Refer to Figure 2). While extending the existing mechanisms to the heterogeneous network, we need to consider the fact that the downstream nodes may be connected through different types of links. The nodes may also face different system and connectivity constraints referred to as end-user constraints. If a source (or a relay node) is aware of the characteristics of the links to the nodes it is serving and their constraints, it can adapt a mechanism or a combination of mechanisms to best serve them. The link characteristics considered are, whether: (i) The link is dedicated or contention-based (ii) It is error-prone or not. Keeping these points in mind, we discuss the applicability of correction-based and resource-based mechanisms to the heterogeneous architecture in this section.

4.1.1. Correction based mechanisms

Recall that the application level correction techniques that can be applied to enhance the receiver perception, even when some transmission errors are encountered are categorized as correction based mechanisms. These mechanisms can be easily adapted to a heterogeneous architecture. A node connected through a link that is prone to errors or congestion (if it is a shared link), could implement an appropriate error concealment technique.

A naïve implementation of FEC techniques can be centralized. In a centralized implementation, FEC can be implemented at S (see Figure 8) and the same stream (with redundant data) can be forwarded by the relay nodes $\{S_1, S_2, \dots, S_m\}$. All the leaf nodes implement mechanisms to recover from losses using the redundant stream. Thus, every node which is an ancestor to one or more nodes, forwards the FEC stream (from source S) and every leaf node implements the algorithm to use the redundant stream to recover from losses. This simple extension will work well when most of the links in the network exhibit similar loss characteristics. When the error rates on the links vary, applying specific correction mechanisms based on the knowledge of the link errors would be more appropriate.

In a hierarchical implementation, every source node learns the characteristics of its links to the downstream nodes. Based on this knowledge, the node can decide whether a FEC mechanism is required or not. For example, source S, based on the characteristics of the links to the nodes it serves can decide to send just the data stream, or data stream with FEC for recovery from single packet loss, or a more complex FEC to recover from multiple packet losses. Given knowledge about the next hop characteristics, a source may serve different links with different mechanisms and may even adapt the mechanisms to the link characteristics dynamically.

The centralized implementation is simple but may be costly in terms of bandwidth consumption if the links in a sub-tree are dedicated and error-free. The hierarchical implementation is more complex and may not be suitable if the link characteristics alternate such that the implementation overheads are high. Further investigation is required to understand the performance of these alternatives.

4.1.2. Resource based mechanism

Recall that mechanisms that ensure effective delivery considering the system and/or the network resources are classified as resource-based mechanisms. Applicability of adaptation, reservation, and hybrid mechanisms to the heterogeneous network architecture are discussed in this section.

4.1.2.1. Adaptation mechanisms

We discussed adaptation techniques in section 2.2.1.2. In a naïve implementation of adaptation based on feedback, if RTCP is used, all the leaf node clients (C_{ij} s in Figure 8) will be sending their reports back to Source S. Through these receiver reports, the source S can find out about the reception quality at the receivers and adjust its rate of transmission accordingly. However, adaptation at source S would affect the quality for all the receivers irrespective of the cause of poor reception at the receivers, which could be due to the link characteristics, or end-user constraints. Thus, the simple extension of RTCP feedback to a heterogeneous network where the link characteristics may vary drastically may not be appropriate. For example, while traffic over a satellite link may experience no congestion if the bandwidth is pre-assigned and dedicated, traffic over the Internet links may experience congestion and/or constraints from the end-user. Thus, adaptation by the source based on feedback without the knowledge of connectivity constraints of the receivers will provide degraded quality for all the receivers, which is not desirable.

In the hierarchical model, feedback can be made local to the parent serving the node. This source can make a decision to adjust the rate of transmission based on its perception of congestion from the feedback received from the nodes it directly serves. Each of these servers could make autonomous decisions about adapting their rates. Such a mechanism requires that the group of receivers treat their upstream sender (parent) as the source, to which they send their feedback.

- Suppose the source S sends a RTP multimedia data packet to the relay node S1. This RTP header will contain source id of S.
- S1, while forwarding this packet to the next level of nodes connected to it, needs to change the source id to its own so that the feedback from the nodes it serves are directly sent to it.

Hierarchical feedback based adaptation, as discussed above works well when the nodes served by a source (at any level) are homogeneous in terms of their resources and link characteristics. If the nodes are heterogeneous, constraint based or transcoder based adaptation techniques, as opposed to feedback-based adaptation, would be more appropriate. Here again, knowing the characteristics of its next hop links, a node at every level can make the right choice of mechanisms to best serve its downstream nodes.

4.1.2.2. Reservation mechanisms

Reservation mechanisms make sense only when a node's link is shared by many different applications and hence prone to congestion. At every level of the hierarchical

representation, a source can opt to deploy reservation mechanisms over the links over which nodes contend for access to resources. The mechanisms discussed in Section 2.2.2 can be directly applied to the sub-trees where nodes are connected through shared links. However, when a source (or a relay node) is aware of the link characteristics and end-user constraints, it can combine reservation mechanisms with other mechanisms to ensure effective delivery at each of the clients. Such combination mechanisms are discussed in the next section.

4.1.2.3. Hybrid mechanisms

It is clear that a source, based on its knowledge of links downstream and the constraints faced by the nodes, can implement a combination of mechanisms to best serve them. The source can differentiate between the various nodes it serves in terms of their link characteristics and constraints, and tune the protocols to achieve effective delivery at each of these nodes. While different combinations are possible, a protocol which combines reservation adaptation with feedback is briefly outlined below.

Combining reservation adaptation with feedback: As discussed in Section 2.2.2.3, multimedia applications can tolerate variations in the QoS within a defined range (QoSmin, QoSmax). While admitting a new application, it is important to ensure that all the existing applications get at least their respective defined QoSmin. Instead of adjusting the QoS of all existing applications to a value between QoSmin and QoSmax (refer to discussion in Section 2.2.2.3), it is possible to adjust the QoS for individual applications based on the feedback received from the receivers. Thus, algorithms that combine feedback mechanisms with reservation are possible in parts of the heterogeneous network where receivers contend for network resources. The reservation protocol may be implemented with

- Strict resource allocation for time critical data,
- Range based resource allocation (which is adjusted during Connection Admission Control (CAC)) for multimedia data, and
- A mechanism such as Controlled Load Service (CLS) [40], which emulates a lightly loaded best effort network service model, for other data traffic.

Receivers send feedback to the sender informing their quality of reception. This feedback can be used by the CAC algorithm to reduce the resources allocated for applications to a value between QoSmin and QoSmax. As the adjustment is made based on feedback, resource allocation may vary among the receivers. Such an algorithm that combines reservation adaptation based on feedback can be implemented at every sub-tree of the hierarchical architecture, where feedback based mechanisms are implemented.

4.2. Applicability of mechanisms for efficient delivery

We classified mechanisms for efficient delivery in Section 3 (Refer to Figure 6). Considering the heterogeneous architecture, issues to be tackled to deliver multimedia efficiently using the storage-based and service-based techniques are discussed in this section.

4.2.1. Storage-based techniques

Recall that techniques which store the contents close to the receivers in order to reduce the retrieval latency are termed storage-based mechanisms. The hierarchical representation inherently supports caching at the relay nodes. In a naïve implementation, relay nodes can also serve as cache servers which store the contents and serve them to their downstream clients. This implementation would be very expensive if most of the relay nodes serve only a few clients or if the client requirements are dynamic.

When a source has knowledge of its downstream nodes and their requirements, it can make a choice to cache or not. If the source chooses to cache, it can also implement the most appropriate cache management technique based on this knowledge. By placing cache servers at select locations in the sub-trees, stored contents can be served efficiently with minimal cache maintenance overheads.

4.2.2. Service based techniques

In Section 3.2, we defined service-based mechanism as those that are designed to service user requests with minimal latency. When the number of end-users requesting the multimedia files on demand from the local servers increases, use of the service-based techniques would help in optimizing the system and network resources. A naïve implementation would use one of the stream merging techniques discussed in Section 3.2.1, at each of the relay nodes that serve the clients attached to it (assuming that every relay node is also a cache server).

A more intelligent implementation would use the knowledge of the nodes attached to the source (or relay nodes) to combine mechanisms to achieve efficiency. For example, by using cache servers (placed appropriately at some relay nodes which use algorithms to cache appropriate files based on user behavior) and serving files from them using stream merging techniques, better resource utilization can be achieved. Also, from the point where the merged streams are served by a single server, it will be possible to use feedback mechanisms to adapt the rate of transmission. Other combination mechanisms such as reservation adaptation with feedback also become relevant when the transmission converges to a synchronous one. Thus, by combining techniques to optimize storage and service along with effective delivery mechanisms, multimedia information can be disseminated efficiently to clients.

4.3. Discussion

From Sections 4.1.and 4.2, the following points emerge:

1. Every source (originating or forwarding the multimedia data), based on its knowledge of the link characteristics to its downstream nodes, makes autonomous decisions about the protocols needed for effective and efficient dissemination in its sub-tree.

The decision-making algorithm can be implemented as a server module. This module takes as input the link characteristics of the next hop nodes (thus constructing the view of the nodes in its sub-tree) and the end-user constraints. Based on these input values, it would decide on the mechanisms appropriate for serving the nodes in an effective and efficient manner. This module can be installed at every node, except the leaf nodes.

2. Every leaf node and relay node implements the algorithms required to support the mechanisms initiated by its parent node.

This support algorithm can be implemented as a client module. This module takes the parameters from the server module and invokes the appropriate algorithms to apply the mechanisms initiated by its parent for effective and efficient delivery.

Given the recursive nature of the hierarchical representation of the heterogeneous architecture, the server module built for source S can be implemented at every relay node and the client module can be implemented at every leaf node. Such a module enables tuning of the mechanisms at every level such that all the receivers are provided with the most effective and efficient delivery of multimedia data given their connectivity and resource constraints.

4.4. Case Study: A heterogeneous architecture for distance education

Many multimedia intensive applications such as distance education are deployed over satellite networks owing to their wide reach and inherent multicast capabilities [23][4]. To provide users with a choice of distance learning models, a heterogeneous architecture illustrated in Figure 9, is implemented at the Distance Education Program, IIT Bombay. In this section, we discuss as a case study, this heterogeneous architecture.

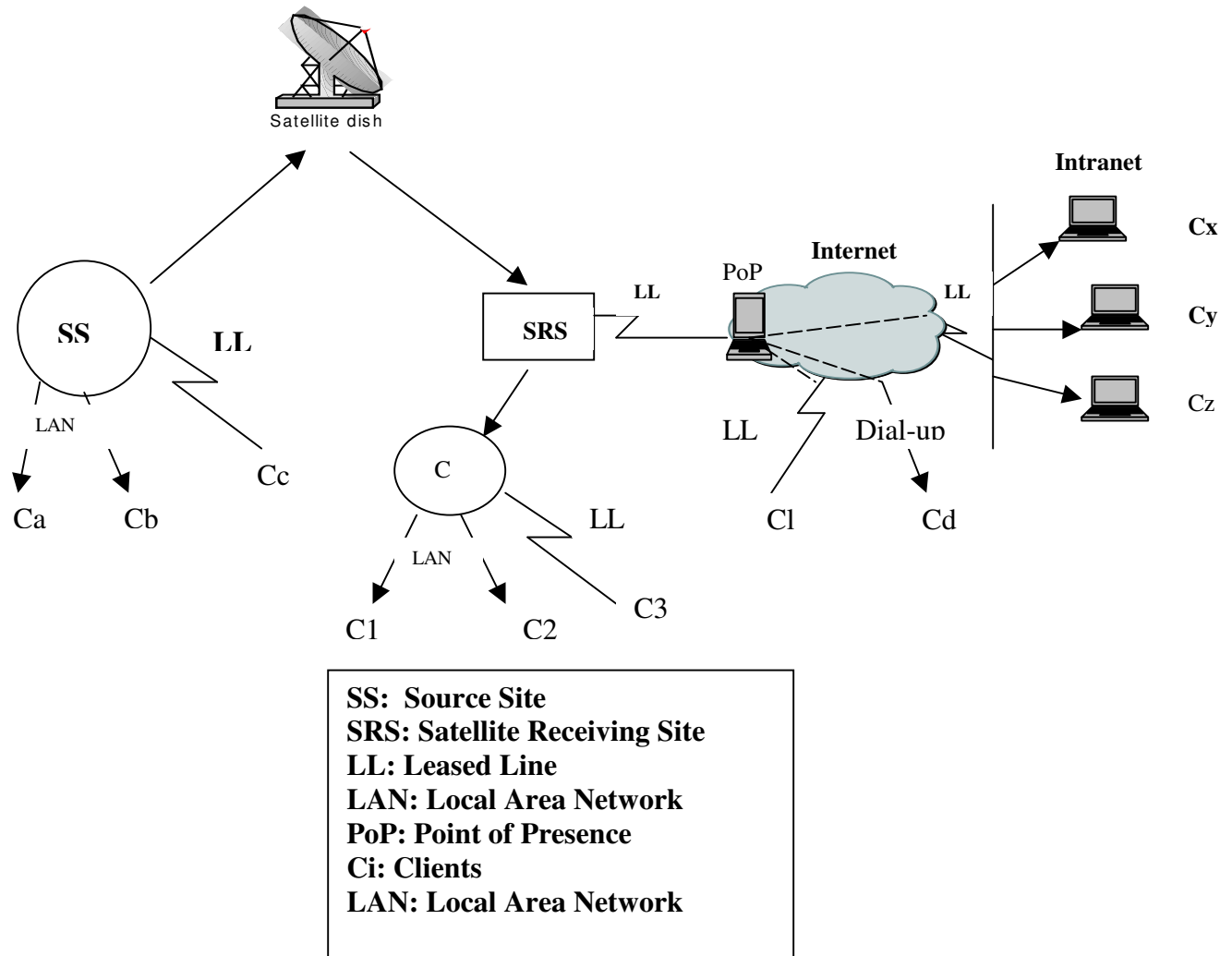


Figure 9: An instance of the heterogeneous architecture - IIT Bombay's DEP network

4.4.1. The model

The model has the following features:

- A network of satellite receiving stations is used for reaching Remote Centers (RCs). Remote Centers can also be directly connected to the source through a LAN or Leased Line (LL). Participants attend lectures at these receiving stations synchronously according to a pre-determined schedule.
- As the lectures are transmitted, they are captured and transmitted live to the servers placed at the Points of Presence (PoPs) of the Internet Service Provider (ISP).
- These servers serve the local area organizations/users connected to the Internet through the LL or dial-up connections. Transmissions can be either synchronous or on demand.

The Source Site (SS) originates the multimedia live stream. Ca, Cb, Cc are clients attached to the SS through LAN or Leased Line (LL). The stream is synchronously received at the Satellite Receiving Sites (SRSs) at different locations, from where participants attend the lectures. C1, C2, C3 are clients attached to the SRS through LAN or Leased Line (LL). PoP is a Point of presence of the Internet service Provider (ISP). The contents are cached at the PoP server, which is a node on the Internet and served over the Internet to clients connected through LL or Dial-up connections synchronously or on demand.

The model provides several options to participants:

- Synchronous delivery (live or from stored medium) at the RCs: In this case a traditional classroom environment is simulated where participants can have live interaction with faculty and peer-to-peer interaction with other participants. However, in this mode participants have to travel to the centers.
- Synchronous delivery over the Internet: This option provides flexibility of space, but not time. Introduces some discipline. However, does not provide live interaction. Off-line interaction (through e-mail and discussion forums) are possible.
- On-demand delivery over the Internet: This option provides flexibility of space and time. Live interaction with faculty is not possible. Off-line support mechanisms are possible. Suitable only for self-motivated individuals.

4.4.2. Discussion

In this model, the satellite network with a dedicated channel of 512 kbps is used for transmission of multimedia lectures from a source site SS to various geographically dispersed nodes in a multicast mode. While there is no congestion-based loss on the satellite channel, the channel may be error-prone. Some clients are also connected to the source directly over LAN or LL. While the LL is dedicated, a LAN is shared by contending applications from many users. The LAN links can be congestion-prone. Since the transmission from SS is synchronous and all the nodes connected to it receive the contents in a multicast mode, feedback mechanisms are suitable for this sub-network. Based on the feedback the source can gauge the extent to which the links are error-prone and opt to provide the appropriate FEC techniques. The receiver nodes themselves can implement simple error concealment techniques if only occasional losses are experienced. Since most of the receivers connected to SS have symmetric links, adaptation based on the feedback may not be necessary at this sub-network. Since most of the links are dedicated and transmission is synchronous, reservation mechanisms and efficient delivery mechanisms are not applicable at this sub-network level.

Each of the SRSs act as source to clients connected through LL or LAN, which is similar to how clients are connected to SS. SRSs at the different regions (cities in different states in most cases) also serve as sources to clients placed at PoPs of ISP. Through LL connections to the SRSs, contents are cached in the streaming servers placed at the PoPs of the ISP at the different regions. Clients at these regions (connected through LL or dial-up connections) can access the contents synchronously or on demand from these streaming servers over the Internet. In other words, the PoP servers serve as cache servers and sources (relay nodes), which store and serve the contents to clients connected over the Internet. This model leverages the satellite network for loading the cache servers. By locally storing the

content, the ISP can provide better QoS to clients in the PoP's local area. These clients connected over the Internet can access the multimedia lectures in the synchronous as well as on-demand mode. This sub-network is prone to congestion and end-user constraints.

In the synchronous mode, feedback mechanisms and feedback-based adaptation can be applied when the end users exhibit homogeneity. The PoP servers can provide multiple layers for the multimedia data, when the receivers are heterogeneous, allowing for constraint-based adaptation. Reservation and hybrid mechanisms are applicable to this sub-network where the clients are served over Internet links.

When the contents are accessed in the on-demand mode, issues dealing with efficient cache management, and management of the on-demand streaming have to be tackled. Based on the knowledge of user behavior and demand for the contents, appropriate stream merging techniques can be applied.

5. Conclusion

Efficient dissemination of multimedia contents over the Internet with guaranteed QoS remains a challenge. In this paper we have reviewed existing mechanisms for delivery of multimedia contents that assume Internet as the underlying network. The contributions of this review include the following:

- Classification of delivery mechanisms for multimedia contents.
- Identification of mechanisms suitable for the synchronous and on-demand delivery modes, and mechanisms that are applicable when the group of receivers is homogeneous and heterogeneous.

This review constitutes a step in the development of effective and efficient dissemination protocols for heterogeneous architectures. One application of such architecture would be distance education. For a distance education application where reaching participants at remote places and guaranteeing the required minimum quality of reception are important, satellite networks provide a scalable solution. However, heterogeneous architectures are attractive since they provide flexibility of access and choice of delivery modes to participants. A sample case study, which is an instance of a heterogeneous architecture, is presented.

The following are our research contributions in extending the reviewed mechanisms to heterogeneous architectures:

- Decomposition of the heterogeneous architecture into a hierarchical representation, which allows us to develop algorithms for implementing mechanisms for multimedia delivery for a source and *recursively* apply them for the relay nodes.
- Preliminary investigation of the applicability of the existing mechanisms to the heterogeneous network and suggestions for modifications and extensions needed to tailor them to such a network.

Implementation and validation of mechanisms suitable for the heterogeneous network of DEP is the next step of our research. Generalizing these modules for any heterogeneous network and validating them are challenges for further research in this area.

Acknowledgements:

We would like to thank Prof. Aniruddha Sahoo, IIT Bombay, for his valuable comments and suggestions. This work is partly supported by the Development Gateway Foundation (DGF) and the Ministry of communication and Information Technology (MIT), Government of India.

References

- [1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, A Permutation Based Pyramid Broadcasting Scheme for Video-on-Demand Systems. IEEE ICMCS, June 1996.
- [2] Akamai: The business Internet: <http://www.akamai.com/>
- [3] E. Amir, S. McCanne, and H. Zhang, An application level video gateway, ACM Multimedia, 1995.
- [4] K. Arya, S. Krithivasan, and S. Iyer, Satellite Based Distance Education Programme at Indian Institute of Technology, Bombay, India. Educational Media In Asia: Reviews, cases and lessons. Vancouver, Commonwealth of Learning, 2003.
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An architecture for Differentiated Services, RFC 2475, December 1998.
- [6] J. Bolot, T. Turlitti, and I. Wakeman, Scalable feedback control for multicast video distribution in the Internet, ACM SIGCOMM, 1994.
- [7] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jasmin, Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification RFC 2205, September 1997.
- [8] I. Busse, B. Deffner, and H. Schulzrinne, Dynamic QoS control of multimedia applications based on RTP, Computer communications, January 1996.
- [9] S. W. Carter and D. D. E. Long, Improving Video-on-Demand Server Efficiency Through Stream Tapping. ICCCN, September 1997.
- [10] J. Chou, and T. Neghen, RTP/RTCP, RTSP, and RSVP, Multimedia protocols for the Internet.
<http://www.cs.berkeley.edu/~randy/Courses/cs294.s02/05InternetMultimedia.ppt>
- [11] A. Dan, D. Sitaram, and P. Shahabuddin, Scheduling Policies for an On-Demand Video Server, ACM Multimedia, October 1994.
- [12] D. L. Eager, M. K. Vernon, and J. Zahorjan, Optimal and Efficient Merging Schedules for Video-on-Demand Servers, ACM Multimedia, November 1999.
- [13] D. L. Eager, M. K. Vernon, and J. Zahorjan, Bandwidth skimming: a technique for cost-effective video-on-demand, MMCN, January 2000.
- [14] F. A. L. Fuentes, An overview of media streaming over peer-to-peer networks, Technische Universität München, April 2002.
- [15] L. Gao, J. Kurose, and D. Towsley, Efficient Schemes for Broadcasting Popular Videos, NOSSDAV, 1998.

- [16] L. Golubchik, J. C. S. Lui, and R. Muntz, Reducing I/O Demand in Video-on-Demand Storage Servers, ACM Sigmetrics, May 1995.
- [17] M. M. Hefeeda, B. K. Bhargava, and D. K. Y. Yau, A Hybrid Architecture for Cost-Effective On-Demand Media Streaming, Journal of Computer Networks, 2004.
- [18] K. Hua, and S. Sheu, Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems, ACM SIGCOMM, September 1997.
- [19] K. A. Hua, Y. Cai, and S. Sheu, Patching: A Multicast Technique for True Video-on-Demand, ACM Multimedia, September 1998.
- [20] C. Huang, and T. Hsu, A user aware prefetching mechanism for video streaming, World Wide Web, Internet and web information systems, Vol.6, Number 4, December 2003.
- [21] L. Juhn, and L. Tseng, Fast Data Broadcasting and Receiving Scheme for Popular Video Service, IEEE Trans. on Broadcasting, March 1998.
- [22] L. Kouvelas, V. Hardman, and J. Crowcroft, Network adaptive continuous media applications through self organized transcoding, NOSSDAV, 1998.
- [23] S. Krithivasan and S. Iyer, To Beam or to Stream: Satellite-based vs. Streaming-based Infrastructure for Distance Education, EDMEDIA, June 2004.
- [24] R. Kumar, J. S. Rao, A. K. Turuk, S. Chattopadhyay and G. K. Rao, A Protocol to Support QoS for Multimedia Traffic over Internet with Transcoding, Department of Computer Science and Engineering, Technical Report, IIT Kharagpur, 2002.
- [25] J. Kurose and K. Ross, Computer networking: A top down approach featuring the Internet, Second edition, Addison Wesley, 2002.
- [26] K.Y. Leung, E.W.M. Wong, and K.H. Yeung, Designing efficient and robust caching algorithms for streaming-on-demand services on the Internet, World Wide Web, Internet and web information systems, Volume 7, Number 3, September 2004.
- [27] J. Liu and B. Li, A QoS-based joint scheduling and caching algorithm for multimedia objects, World Wide Web, Internet and web information systems, Volume 7, Number 3, September 2004.
- [28] A. Mahanti, D. L. Eager, M. K. Vernon, and D. Sundaram-Stukel, Scalable On-Demand Media Streaming with Packet Loss Recovery, SIGCOMM, August 2001.
- [29] A. Mahanti, On-Demand Media Streaming on the Internet: Trends and Issues, Comprehensive Examination Paper, December 2001.

- [30] S. McCanne, V. Jacobson, and M. Vetterli, Receiver-driven layered multicast, SIGCOMM symposium on communications architectures and protocols, August 1996.
- [31] P. Paul, S. V. Raghavan, Survey of Multicast Routing Algorithms and Protocols, Proceedings of the 15th international conference on Computer communication, Pages: 902 – 926, 2002.
- [32] C. Perkins, O. Hodson, and V. Hardman, A Survey of Packet Loss Recovery Techniques for Streaming Audio, IEEE Network, September/October 1998.
- [33] M. Ramalho, Intra- and Inter-Domain Multicast Routing Protocols: A Survey and Taxonomy, IEEE communications, Surveys and Tutorials, 2000.
<http://www.comsoc.org/livepubs/surveys/public/1q00issue/ramalho.html>
- [34] L. Rizzo, Effective Erasure Codes for Reliable Computer Communication Protocols. ACM Computer Communication Review, 27(2): 24–36, April 1997.
- [35] A. Sahoo, B. Devalla, Y. Guan, R. Bettati, and W. Zhao, Adaptive connection management for mission critical applications over ATM-based networks, NAECON, 1998.
- [36] H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP), RFC 2326, April 1998.
- [37] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications, RFC 3550, 2003.
- [38] S. Sen, J. Rexford, and D. Towsley. Proxy prefix caching for multimedia streams. IEEE INFOCOM, March 1999.
- [39] S. Sen, L. Gao, J. Rexford, and D. Towsley, Optimal Patching Schemes for Efficient Multimedia Streaming, NOSSDAV, June 1999.
- [40] S. Shenker, and J. Wroclawski, General characterization parameters for Integrated Service Network elements, RFC 2215, September 1997.
- [41] D. Sisalem and F. Emanuel, QoS control using adaptive layered data transmission, ACM multimedia, 1995.
- [42] D. Sisalem, and H. Schulzrinne, The Loss-Delay Based adjustment algorithm: A TCP-friendly adaptation scheme, NOSSDAV, 1998.
- [43] B. C. Smith, A Survey of Compressed Domain Processing Techniques, <http://www.uky.edu/~kiernan/DL/bsmith.html>
- [44] H. Tan, D. Eager, M. Vernon, and H. Guo, Quality of service evaluations of multicast streaming protocols, ACM Sigmetrics, 2002.

- [45] The MPEG homepage: <http://www.chiariglione.org/mpeg/>
- [46] B. Vandalore, W. Feng, R. Jain, and S. Fahmy, A Survey of Application Layer Techniques for Adaptive Streaming of Multimedia, Journal of Real-time systems, 2000.
- [47] B.J. Vickers, C. Albuquerque, and T. Suda, Adaptive multicast of multi-layered video: Rate-based and credit-based approaches, IEEE Infocom, 1998.
- [48] S. Viswanathan, and T. Imielinski, Metropolitan Area Video-on-Demand Service using Pyramid Broadcasting. Multimedia Systems, August 1996.
- [49] J. Wang, A Survey of Web Caching Schemes for the Internet, Cornell Network Research Group, 2001.
- [50] X. Wang, and H. Schulzrinne, Comparison of adaptive Internet multimedia applications, Invited paper, Special issue on distributed processing for controlling telecommunications systems, June 1999.
- [51] Y. Wang, Z. Zhang, D. Du, and D. Su, A network-conscious approach to end-to-end video delivery over wide area networks using proxy servers, IEEE INFOCOM, April 1998.
- [52] Z. Wang, Internet QoS – Architectures and mechanisms for Quality of Service, Morgan Kauffman, 2001.
- [53] L. Wu, R. Sharma, and B. Smith, Thin streams: An architecture for multicasting layered video, NOSSDAV 1997.