INTRODUCTION:

In recent years, there is an emerging need to distribute popular realtime streaming media on-demand to a large number of clients over the Internet. However, streaming multimedia traffic to a large number of customers imposes a high traffic load on the network. The high volume of such multimedia traffic alongside timing constraints requires a large-scale, cost-effective media distribution system.

In Internet, upon each client’s request, the media server retrieves the video and multicasts it over User Datagram Protocol (UDP) at a constant rate which equals to the drain rate at the client side. However, these UDP packets may be dropped or delayed during the transmission over the Internet, e.g. due to network congestion. Thus, a buffering scheme is usually used at the server side before sending the video file into the network in order to control the sending rate to fit the current network status and service requirements. Besides, for efficiency reasons the video must be compressed before it is sent to the network. Once the client receives the compressed video from the network, it needs to decompress the video by its local media player. Before starting to play back the video, the client places the received packets into its own buffer, so-called client buffer, which is set to balance the receiving rate and the playing rate. Finally, the video can be decoded and played back properly at the client.

However, it becomes challenging for today’s Internet to rely on such an architecture to deliver videos to a large number of users due to the following reasons.

• The traditional media streaming systems use client-server approaches to allocate a dedicated stream from a media server upon each client’s request. However, limited processing power, memory size and limited out-bound network bandwidth of the streaming server causes a limitation on the total number of concurrent clients that the system can support.

• As IP multicast is not widely deployed and is not generally available as a service for average end users, most of the existing media distribution systems rely on native unicast protocols for delivering video. However, unicast is recognized as an inefficient way of delivering multimedia services to a large number of clients. It not only wastes the network resources but also raises
scalability issues. The scalability issue results from the fact that adding Internet-scale potential users requires a commensurate amount of resource to the supplying server(s).

• Streams need to be transported reliably to the end-user across the Internet. However, the Internet is designed as a best effort network without considering application's Quality of Service (QoS) requirements. Communications between two end points are not guaranteed and packets may be lost or delayed if they traverse congested routers or links. Another reliability concern arises from the fact that only one type of entity (i.e. video server) is responsible for all clients. Thus, the server failure may take place due to instant, short- or even long-term overloads.

• The timing constraint makes the system even harder to design. Besides the timing requirements for playing video in time, the media streaming system should be able to detect and recover from failures quickly, so that the service disruption for the affected nodes is minimized.

Therefore, this work focuses on overcoming the problem of serving multimedia files to a large number of end hosts distributed across the Internet.

The peer-to-peer technology provides a promising alternative solution for multicast communication. Peer-to-peer is referred as a fully distributed network architecture which is in contrary to the traditional client-server model. In client-server model, a server is providing centralized service requested by different clients, i.e., hosts. Usually in this approach the address of the server is known by the hosts in advance. The obvious drawback of client-server model is its limited bandwidth and resource of the server which is shared by all the hosts due to which the server can be easily overwhelmed by a huge number of simultaneous hosts. Peer-to-peer model allows hosts to form an adhoc logical overlay network on top of the physical network. The links of the overlay network are logical links each of which can be mapped to a physical path in the physical network. Hosts can share information as well as bandwidth with each other through the overlay network. This requires hosts be able to perform more complex operations such as routing and overlay topology construction/maintenance. The advantages of peer-to-peer systems are obvious. First, as peers can share their bandwidth with each other, peers can contribute their bandwidth to the system. More peers join the system, more bandwidth the system has. Second, the processing is distributed, no single point of failure problem. In client-server model, the server is much more important than the hosts. If the server is down, the whole system is down as well. This is called single point of failure problem. In peer-to-peer systems, peers are of equal importance. The overlay network is formed in a distributed fashion such that the system can continue to functions properly even if some of the peers leave
the system.

Due to the explosive growth of the Internet and increasing demand for multimedia information on the web, streaming video over the Internet has received tremendous attention from academia and industry. Transmission of real-time video typically has bandwidth, delay, and loss requirements. However, the current best-effort Internet does not offer any quality of service (QoS) guarantees to streaming video. Furthermore, for video multicast, it is difficult to achieve both efficiency and flexibility.