CS 4453 Computer Networks

Chapter 6 Multimedia Networking

2015 Winter

6.1 Video and audio on-line

The most salient characteristic of video is its high bit rate.

Another important characteristic of video is that it can be compressed, there by trading off video quality with bit rate. There are two types of redundancy in video which can be used for compression.

- Spatial redundancy. Ah image that consists of mostly white space has a high degree of redundancy and can be compressed without significantly sacrificing image quality.
- Temporal redundancy. An image and the subsequent image may be the same.

By using compression, we can create multiple versions of the save video, each at a different quality level.

Digital audio has lower bandwidth requirement comparing to the video. But it has its own characteristics.

The analog audio signal is sampled at some fixed rate, for example, 8,000 samples per second. Each of the samples is then rounded to one of a finite number of values. Each of the quantization values is represented by one byte. For playback, the digital signal can be converted back, but only in an approximation of the original signals.

By increasing the sampling rate and the number of quantization values, the decoded signal can better approximate the original analog signal. Compression are also used for the audio.

Although audio bit rates are generally much less than those of video, users are generally much more sensitive to audio glitches than video glitches.

Types of multimedia network applications

The multimedia applications can be classified into three broad categories:

- Streaming stored audio/video.
- Conversational voice-/video-over-IP.
- Streaming live audio/video.

Streaming stored audio and video

Streaming stored video has some key distinguishing features.

- Streaming. The client typically begins video play out within a few seconds after it begins receiving the video from the server. This means that the client will be playing out from one location in the video while at the same time receiving later parts of the video from the server.
- Interactivity. The user may pause, reposition forward, reposition backward, fast-forward, and so on through the video content.
- Continuous play out. Once play out of the video begins, it should proceed according to the original timing of the recording. So data must be received from the server in time for its play out.

The most important performance measure for streaming video is average throughput. By using buffering and prefetching, it is possible to provide continuous playout even when the throughput fluctuates, as long as the average throughput remains above the video rate.

For many streaming video applications, prerecorded video is stored on, and streamed from, a CDN rather than from a single data center. P2P video streaming applications are also used for many applications.

Conversational voice- and video-over-IP

Real-time conversational voice over the internet is often referred as Voice-over-IP (VoIP) or Internet telephony.

Timing considerations are important because audio and video conversational applications are highly delay-sensitive. The delay from when a user speaks or moves until the action is manifested at the other side should be less than a few hundred milliseconds. For voice, delays smaller than 150 milliseconds are not perceived by a human listener, delay between 150 and 400 milliseconds can be acceptable and delays exceeding 400 milliseconds can result in frustrating.

On the other hand, conversational multimedia application are loss-tolerant–occasional loss only causes occasional glitches in audio/video playback, and these losses can often be partially of fully concealed.

Streaming live audio and video

These applications allow a user to receive a live radio or television transmission.

Live, broadcast-like applications often have many users who receive the same audio/video program at the same time. Although the distribution of live audio/video to many receivers can be efficiently accomplished using the IP multicasting techniques, multicast distribution is more often accomplished via application-layer multicast (using P2P network or CDNs).

Streaming stored video

A common characteristic of the video streaming is the extensive use of client-side application buffering to mitigate the effects of varying end-to-end delays and varying amounts of available bandwidth between server and client.

Streaming video systems can be classified into three categories.

- UDP streaming: UDP streaming typically uses a small client-side buffer, big enough to hold less than a second of video. Before passing the video chunks to UDP, the server will encapsulate the video chunks within transport packets specially designed for transporting audio and video, using the RTP (which we will discuss later) or a similar scheme. In addition to the server-to-client video stream, the client and the server also maintain, in parallel, a separate control connection over which the client sends commands regarding session state changes. UDP streaming has some drawbacks. The unpredictable and varying amount of available bandwidth between server and client may cause problems. The additional control connection increases the cost and complexity of the transmitting. Some firewall are configured to block UDP traffic which also causes problems.

- HTTP streaming: The video is stored in an HTTP server as an ordinary file with a specific URL. Upon requesting, the server sends the video as quickly as possible. One client side, the bytes are collected in a client application buffer and start to playback when the buffer exceeds a predetermined threshold. The client also can pre-fetch video frames that are to be consumed in the future. Note that the application buffer is different from the TCP receiving buffer. The application buffer works together with the TCP buffer to smooth the playback. HTTP streaming does not require a control connection. HTTP byte-range header is used to specify the range of bytes the client currently wants to retrieve from the desired video. Using this header, the user can reposition the video palyout.

- Adaptive HTTP streaming: One shortcoming of the HTTP streaming is that the client cannot choose the version of the video even the video can be encoded into different versions from high-definitions to low-definitions. Dynamic Adaptive Streaming over HTTP (DASH) is developed for improving that problem. With DASH, each video version is stored in the HTTP server with different URL. The HTTP server also has a manifest file, which provides a URL for each version along with its bit rate. The client first requests the manifest file and learns about the various versions. Then the client selects one chunk at a time by specifying the URL.

While downloading chunks, the client also measures the received bandwidth and runs a rate determination algorithm to select the chunk to request next. DASH therefore allows the client to freely switch among different quality levels. By dynamically monitoring the available bandwidth and client buffer level, and adjusting the transmission rate with version switching, DASH can often achieve continuous playout at the best possible quality level without frame freezing or skipping.

In many implementations, the server not only stores many versions of the video but also separately stores many versions of the audio, so that the client can dynamically select both video and audio chunks, and locally synchronizes audio and video playout.

Content distribution networks

Many companies need to distribute on-demand streaming videos to clients all over the world. A single massive data center does not quite fit that purpose. Most major video-streaming companies now make use of Content Distribution Networks (CDNs).

A CDN manages servers in multiple geographically distributed locations, stores copies of the videos (and other types of Web content, including documents, images, and audio) in its servers, and attempts to direct each user request to a CDN location that will provide the best user experience.

The CDN may be a private CDN, i.e., owned by the content provider itself (for examples, Google's CDN distributes YouTube videos). The CDN may alternatively be a third-party CDN that distributes content on behalf of multiple content providers (for example, Akamai's CDN distributes Netflix and Hulu).

CDNs typically adopt one of two different server placement philosophies.

- Enter Deep. Pioneered by Akamai, it deploys server clusters in access ISPs (ISPs direct accessing end users) all over the world. The goal is to get close to end users, thereby improving user-perceived delay and throughput by decreasing the number of links and routers between the end user and then CDN cluster from which it receives content. Because of this highly distributed design, the task of maintaining and managing the clusters become challenging.

- Bring home. Taken by Limelight and other CDN companies, it brings the ISPs home by building large clusters at a smaller number of key locations and connecting these clusters using a private high-speed network. Instead of getting inside the access ISPs, these CDNs typically place each cluster at a location that is simultaneously near the point of presence of many tier-1 ISPs. Compared with the enter-deep design, the bring-home design typically results in lower maintenance and management overhead, possibly at the expense of higher delay and lower throughout to end users.

Once its clusters are in place, the CDN replicates content across its clusters. Usually, the CDN does not place a copy of every video in each cluster. Many CDNs do not push videos to their clusters but instead use a simple pull strategy: if a client requests a video from a cluster that is not storing the video, then the cluster retrieves the video and stores a copy locally while streaming the video to the client at the same time. When a cluster's storage becomes full, it removes videos that are not frequently requested.

CDN operation

Most CDNs take advantage of DNS to intercept and redirect requests. For example, suppose a content provider *LUvideo*, employs the third-party CDN company, *CCC*, to distribute its video to its customers. On the *LUvideo* web pages, each of its videos is assigned a URL that includes the string "video" and a unique identifier for the video itself; for example, Transformers 7 might be assigned `http://video.LUvideo.ca/6Y7B23V`.

Then the following steps occur:

1. The user visits the web page at *LUvideo*.
2. When the user clicks on the link `http://video.LUvideo.ca/6Y7B23V`, the user's host sends a DNS query for `video.LUvideo.ca`.
3. The user's Local DNS server (LDNS) relays the DNS query to an authoritative DNS server for LUvideo, which observes the string "video" in the hostname `video.LUvideo.ca`. To hand over the DNS query to *CCC*, instead of returning an IP address, the LUvideo authoritative DNS server returns to the LDNS a hostmane in the *CCC*'s domain, for example, `a1105.ccc.com`.

4. From this point on, the DNS query enters into *CCC*'s private DNS infrastructure. The user's LDNS then sends a second query, now for `a1105.ccc.com`, and *CCC*'s DNS system eventually returns the IP addresses of a *CCC* content server to the LDNS. It is thus here, within the *CCC*'s DNS system, that the CDN server from which the client will receive its content is specified.
5. The LDNS forwards the IP address of the content-serving CDN node to the user's host.
6. Once the client received the IP address for a *CCC* content server, it establishes a direct TCP connection with the server at the IP address and issues a HTTP GET request for the video. If DASH is used, the server will first send to the client a manifest file with s list of URLs, one for each version of the video, and the client will dynamically select chunks from the different versions.

Cluster selection strategies

Cluster selection strategy is a mechanism for dynamically directing clients to a server cluster or a data center within the CDN. As we just saw, the CDN learns the IP address of the client's LDNS server via the client's DNS lookup. After learning this IP address, the CDN needs to select an appropriate cluster based on this IP address. CDNs generally employ proprietary cluster selection strategies.

One simple strategy is to assign the client to the cluster that is geographically closest. (Using commercial geo-location databases, each LDNS IP addresses is mapped to a geographic location). Such a solution can work reasonably well for a large fraction of the clients.

However, for some clients, the solution may perform poorly, since the geographically closest cluster may not be the closest cluster along the network path. And a problem inherent with all DNS-based approaches is that some end-users are configured to use remotely located LDNSs.

Moreover, this simple strategy ignores the variation in delay and available bandwidth over time of Internet paths, always assigning the same cluster to a particular client.

One method CDNs can be used is performing periodic real-time measurements of delay and loss performance between their clusters and clients. For instance, a CDN can have each of its clusters periodically send probes to all of the LDNSs around the world. One drawback of this approach is that many LDNSs are configured to not respond to such probes.

An alternative to sending extraneous traffic for measuring path properties is to use the characteristics of recent and ongoing traffic between the clients and CDN servers. Another alternative method is to use DNS query traffic to measure the delay between clients and clusters.

A very different approach to matching clients with CDN servers is to use IP anycast (RFC 1546). The idea behind IP anycast is to have the routers in the internet route the client's packets to the "closest" cluster, as determined by BGP.

During the IP-anycast configuration stage, the CDN company assigns the same IP address to each of its clusters, and uses standard BGP to advertise this IP address from each of the different cluster locations. When a BGP router receives multiple route advertisements for this same IP address, it treats these advertisements as providing different paths to the same physical location (in fact for different physical locations). Following standard operating procedures, the BGP router will then pick the "best" router to the IP address according to its local route selection mechanism.

For example, if one BGP route is only one AS hop away from the router, and all other BGP routers are two or more AS hops away, then the BGP router would typically choose to route packets to the location that needs to traverse only one AS. This approach has the advantage of finding the cluster that is closest to the client rather than the cluster that is closest to the client's LDNS. However, the IP anycast strategy again does not take into account the dynamic nature of the internet over short time scales.

6.2 Protocols for real time applications

On-line real-time conversational applications, including VoIP and video conferencing, are very popular now. Therefore IETF and ITU have been busy for trying standard protocols.