4.5 Streaming Media on the Web*

Streaming media are becoming more and more popular on the World Wide Web. Whereas a *direct* streaming from a Web server to a Web client follows the principles explained in the previous sections, **Web caches** and **Web proxies** pose new challenges.

* The transparencies of this section are based on a tutorial presented by Dr. Markus Hofmann (Lucent Bell Labs) at ICNP 2004. His support is gratefully acknowledged.

What is Streaming?

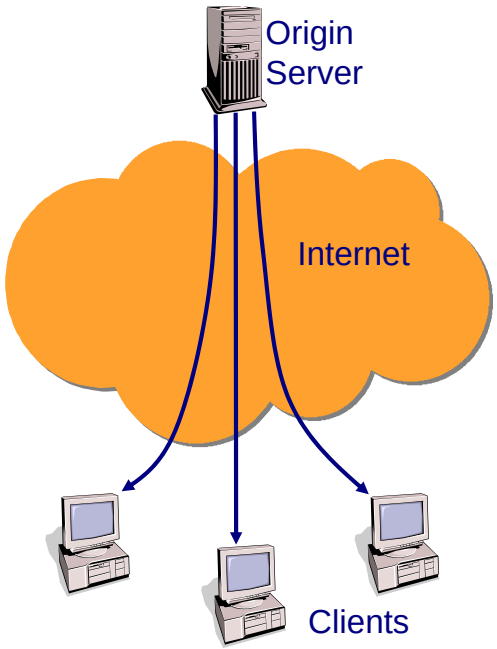
- “Streaming media” refers to media types with time constraints and a continuous data flow, mainly video and audio transmission
- Playback of streaming media starts already while data is being received, i.e., it is not necessary to download the entire file before playback starts. This is different from plain media downloads (e.g., using Napster).
- The streaming media server, advertising, subscription services, and online music shipment market is expected to grow to $14.9 billion by 2009 (source: Web page of Global Information, Inc.)

Application Examples

- Live, non-interactive applications:
  - Internet radio, news broadcasts, sport events, etc.
  - Content is typically not recorded in advance; loose delay constraints
- Live, interactive applications:
  - Teleconferencing, video/audio phones, distributed games, etc.
  - Tight delay constraints to support interactivity
- Stored, non-interactive applications (on-demand):
  - Movies over the Internet, news archives, video clips on homepages
  - Content recorded in advance; loose delay constraints

Streaming on the Internet - Today

- Current streaming technology is in a relatively primitive stage, despite great research work.
- Dominant market players are RealNetworks, Microsoft, and Apple (Quicktime)
- Most streaming applications are based on unicast transport:
  - Server load increases linearly with the number of clients.
  - Bandwidth intensive multimedia flows lead to serious network congestion.
  - Clients experience unpredictable playback quality and high start-up latency.

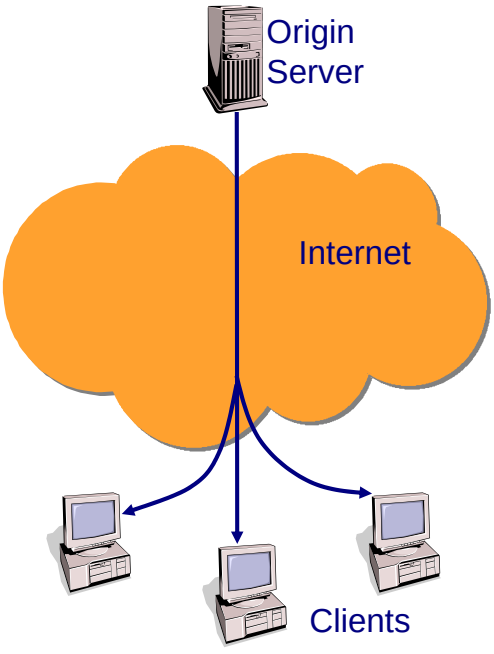


Example: Clinton's Testimony

- CNN audio and video quality became unbearable for most people at around 1:00 pm on August 17, 1998
- Link to video stream removed from CNN Web page at 1:15 pm
- Other news servers were also unreachable

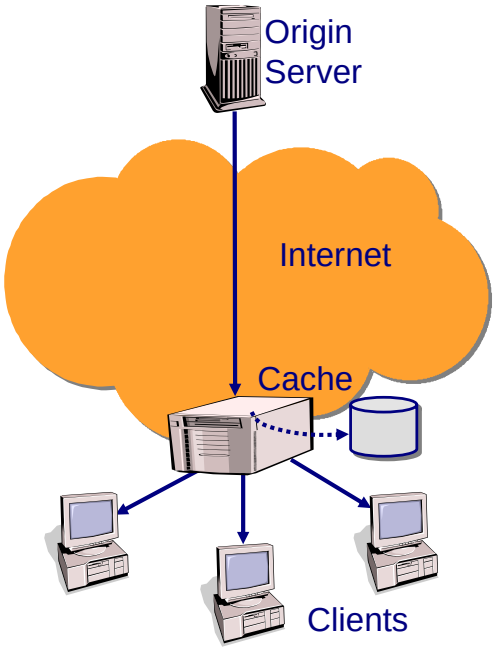
Using IP Multicast for Streaming Media?

- IP multicast replicates data packets at branching points in the network.
- Benefits of using IP multicast technology:
  - Reduction of server load,
  - Reduction of network load.
- Problems with multicast-based streaming:
  - IP multicast is not yet widely deployed (lack of a business model, management issues, does not yet work properly across domains, etc.).
  - Multicast is useful for live broadcasts; it is not directly applicable to on-demand style services (because it requires synchronous receivers).
  - Multicast does not address the problem of high start-up latency.



Caching of Streaming Media

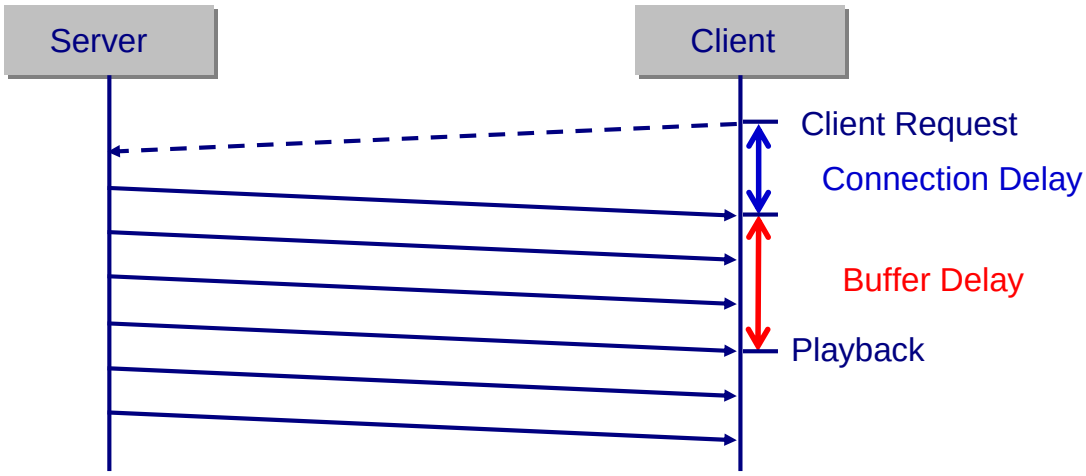
- Client requests for streaming media are handled by the local cache:
  - *Cache miss*: Request is forwarded to the origin server which starts playback through the cache; the cache relays the data and simultaneously stores the stream.
  - *Cache hit*: The cache starts playback of the requested stream.
- Benefits of caching:
  - Reduced server load,
  - Reduced network load,
  - Improved playback quality,
  - Improved start-up latency.



Caching Techniques for Streaming Media

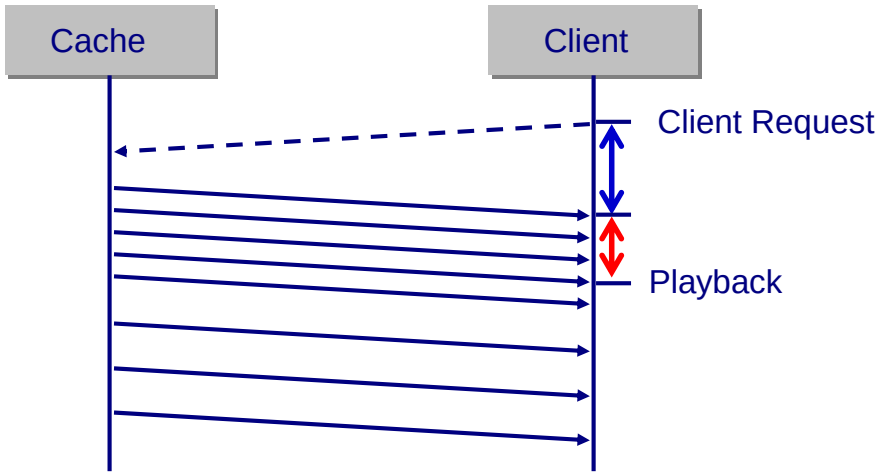
- Why is streaming media different from conventional Web traffic?
  - Different signaling protocols (HTTP vs. RTSP)
  - Object size: caching of streaming objects in their entirety does not scale to large numbers
  - Time constraints of continuous media
- Classical caching techniques are not applicable to streaming media.
- Innovative streaming solutions will adapt to user behavior and will scale with an increasing demand of streaming media.
  - “Fast prefix transfer” reduces access delay.
  - Scalability is achieved by segmentation of streaming objects.
  - Dynamic caching increases throughput through request aggregation.

Conventional Buffered Playback



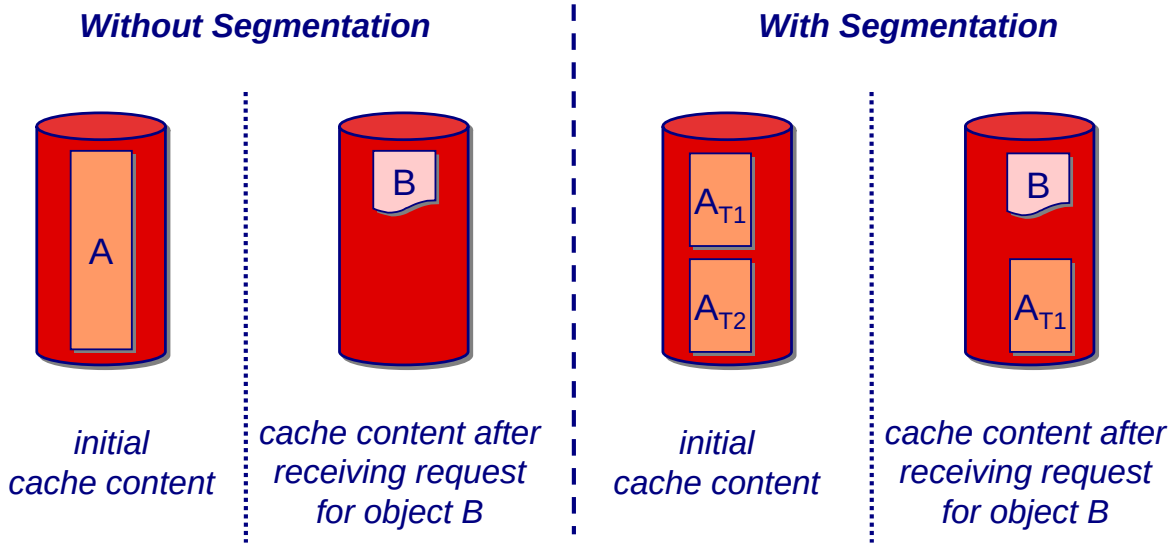
- Streaming applications maintain a playback buffer to absorb jitter.
- Playback usually starts when the playback buffer is filled completely.

Fast Prefix Transfer



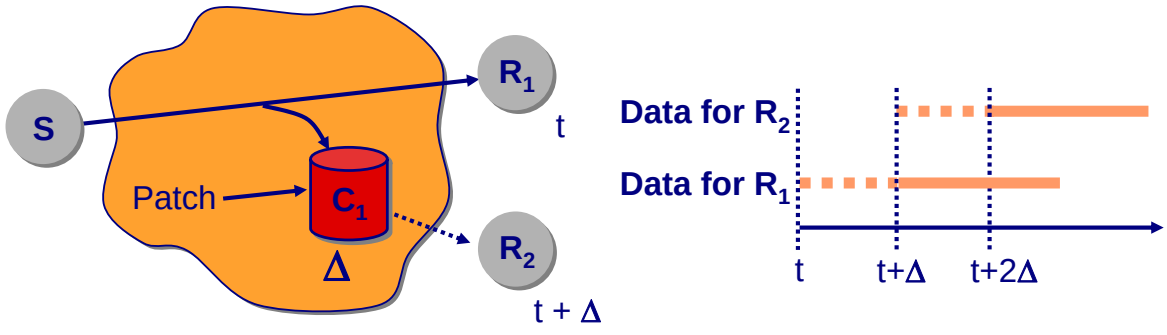
- **Fast prefix transfer** at the beginning of the data transfer decreases buffering delay.
- Knowledge of the buffer size at the client allows for optimization.

Segmentation of Streaming Objects



- Division of streaming objects into smaller segments
- Segments can be cached and replaced independently

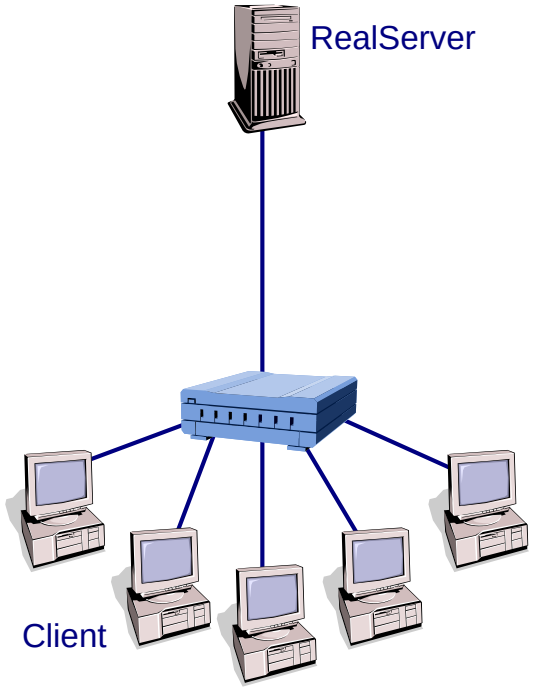
Dynamic Caching



- Asynchronous requests characterized by *temporal distance* Δ
- Dynamic cache bridges temporal distance
  - enables use of multicast even for asynchronous requests
- Size of dynamic cache independent from size of media object
- Dynamic caching requires data *patching*

Implementation

- Implementation on FreeBSD
  - Support for Real Time Streaming Protocol (RTSP) and Realtime Transport Protocol (RTP)
- Experiments:
  - 12 MPEG objects with different playback rates
  - Object selection according to the Zipf distribution
  - Time between two successive client requests: 15 sec
  - Playback buffer: 5 sec



Fast Prefix Transfer