Multimedia Communications

Part D of Seminar on Current Issues and Technologies for the Internet

Dr. Junaid Ahmed Zubairi

Visiting Associate Professor

CIT, Agriculture University, Rawalpindi

August17th, 2002 8:30PM

Overview

- Multimedia requirements
- Streaming
- Phone over IP
- Recovering from Jitter and Loss
- Reference: Computer Networking by James F. Kurose and Keith W. Ross, 1st Edition, Addison Wesley 2001

Application Classes

- Typically sensitive to delay, but can tolerate packet loss (would cause minor glitches that can be concealed)
- Data contains audio and video content (“continuous media”), three classes of applications:
  - Streaming
  - Unidirectional Real-Time
  - Interactive Real-Time

Application Classes (more)

- **Stored Audio and Video Streaming**
  - Clients request audio/video files from servers and pipeline reception over the network and display
  - Clients can start playing out initial portion of the file while the file has not been completely received (it is known as streaming)
  - Interactive: user can control operation (similar to VCR: pause, resume, fast forward, rewind, etc.)
  - Delay: from client request until display start can be 1 to 10 seconds
  - Once started, the playout is continuous

Application Classes (more)

- **Unidirectional Real-Time:**
  - similar to existing TV and radio stations, but delivery on the network
  - Non-interactive, just listen/view
- **Interactive Real-Time** :
  - Phone conversation or video conference
  - More stringent delay requirement than Streaming and Unidirectional because of real-time nature
  - Video: < 150 msec acceptable
  - Audio: < 150 msec good, <400 msec acceptable

Challenges

- TCP/UDP/IP suite provides best-effort, no guarantees on expectation or variance of packet delay
- Streaming applications delay of 5 to 10 seconds is typical and has been acceptable, but performance deteriorates if links are congested (transoceanic)
- Real-Time Interactive requirements on delay and its jitter have been satisfied by over-provisioning (providing plenty of bandwidth), what will happen when the load increases?...

Challenges (more)

- Most router implementations use only First-Come-First-Serve (FCFS) packet processing and transmission scheduling
- To mitigate impact of “best-effort” protocols, we can:
  - Use UDP to avoid TCP and its slow-start phase…
  - Buffer content at client and control playback to remedy jitter
  - Adapt compression level to available bandwidth

Solution Approaches in IP Networks

- Just add more bandwidth and enhance caching capabilities (over-provisioning)!
- Need major change of the protocols :
  - Incorporate resource reservation (bandwidth, processing, buffering), and new scheduling policies
  - Set up service level agreements with applications, monitor and enforce the agreements, charge accordingly

Solution Approaches in IP Networks

- Need moderate changes ("Differentiated Services"):
  - Use two traffic classes for all packets and differentiate service accordingly
  - Charge based on class of packets
  - Network capacity is provided to ensure first class packets incur no significant delay at routers

Streaming

- Important and growing application due to reduction of storage costs, increase in high speed net access from homes, enhancements to caching and introduction of QoS in IP networks

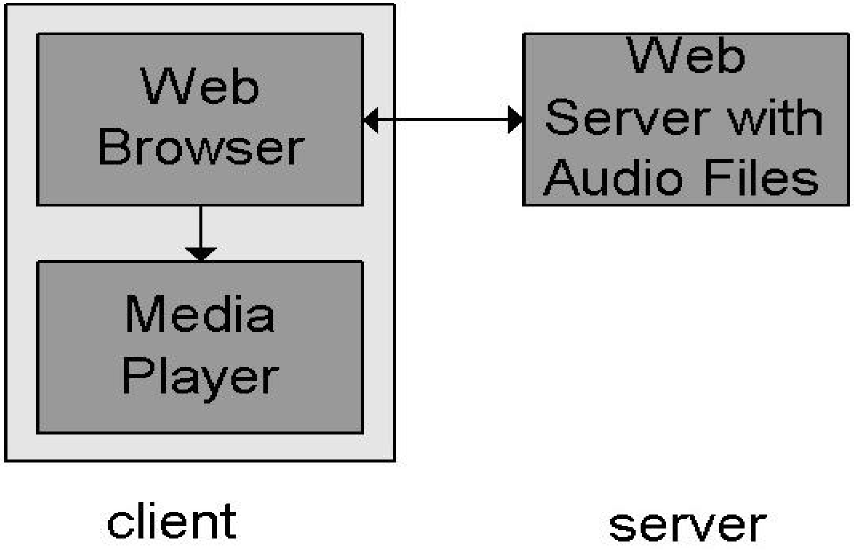
- Audio/Video file is segmented and sent over either TCP or UDP, public segmentation protocol: **Real-Time Protocol (RTP)**

Streaming

- User interactive control is provided, e.g. the public protocol **Real Time Streaming Protocol (RTSP)**
- **Helper Application:** e.g. RealPlayer; displays content, which is typically requested via a Web browser; typical functions:
  - Decompression
  - Jitter removal
  - Error correction: use redundant packets to be used for reconstruction of original stream
  - GUI for user control

Streaming From Web Servers

- Audio: in files sent as HTTP objects
- Video (interleaved audio and images in one file, or two separate files and client synchronizes the display) sent as HTTP object(s)



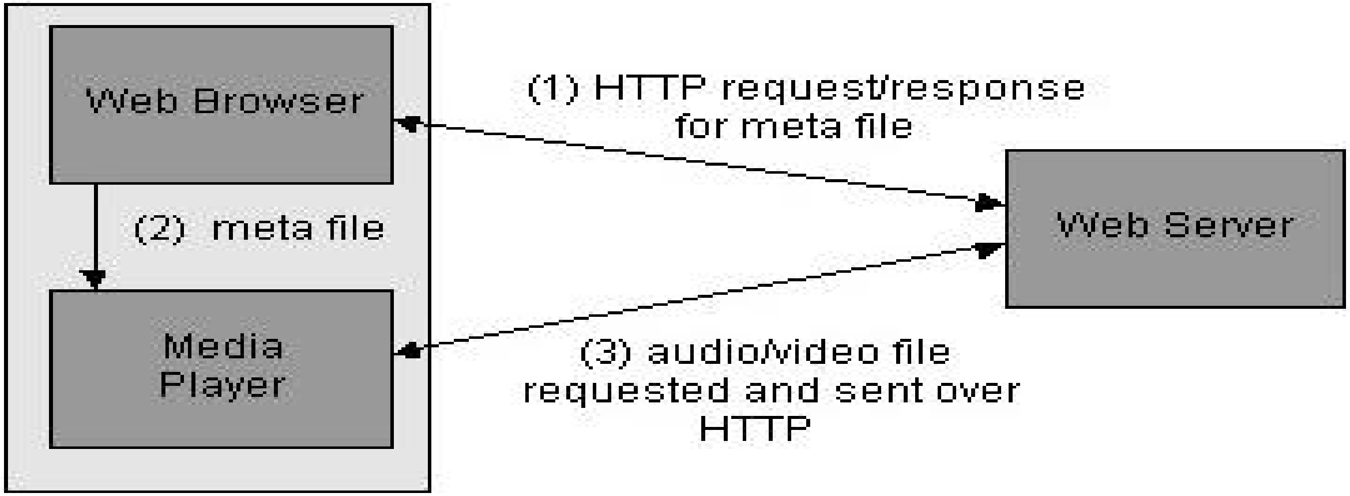
Simple Approach

- A simple architecture is to have the Browser requests the object(s) and after their reception pass them to the player or display
  - No pipelining (Entire media content must be downloaded before playing it)
- Unacceptable delays

Streaming From Web Server (more)

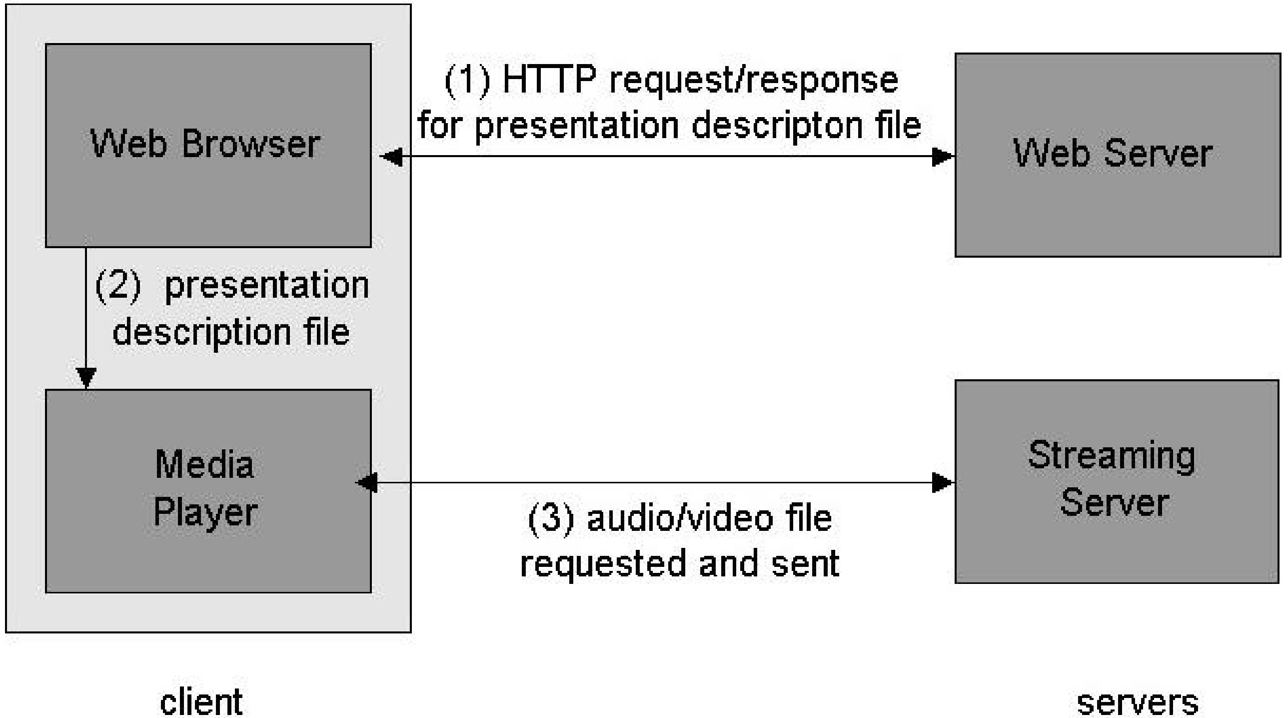
- Alternative: set up connection between server and player, then download
- Web browser requests and receives a **Meta File** (a file describing the object) instead of receiving the file itself;
- Browser launches the appropriate Player and passes it the Meta File;
- Player sets up a TCP connection with Web Server and downloads the file

Meta file requests



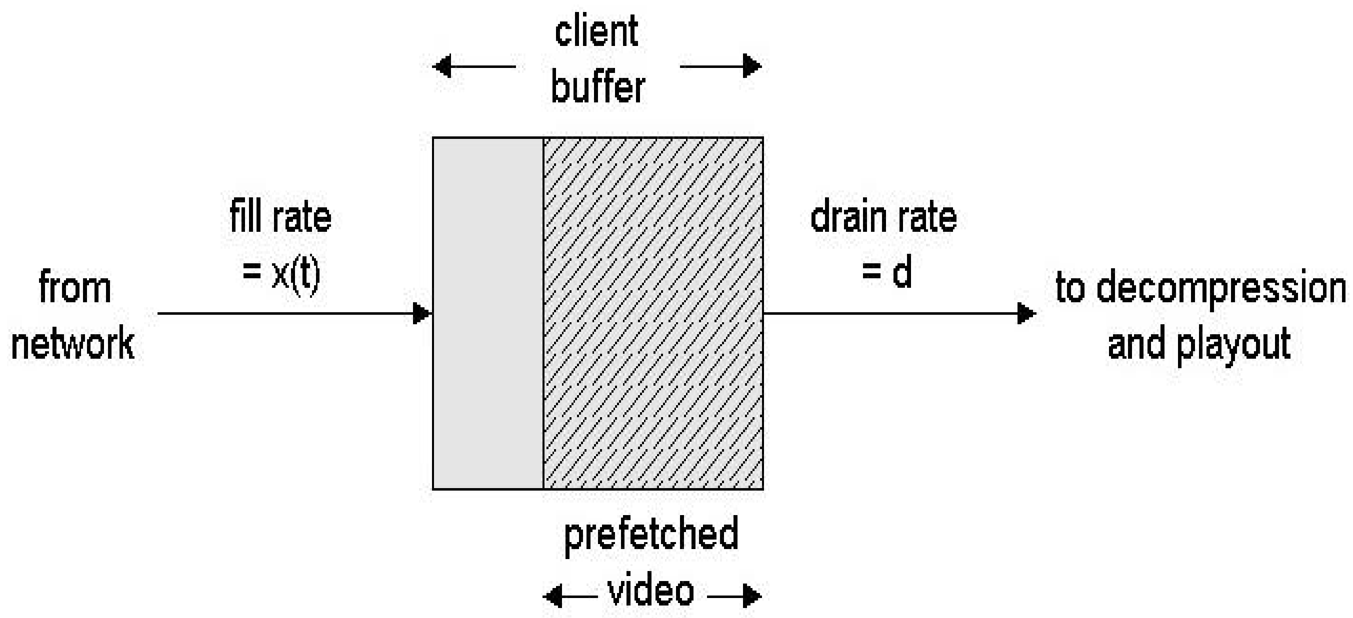
Using a Streaming Server

- This gets us around HTTP, allows a choice of UDP vs. TCP and the application layer protocol can be better tailored to Streaming; many enhancements options are possible (see next slide)



Options When Using a Streaming Server

- Using UDP, Server sends at a rate (Compression and Transmission) appropriate for client; to reduce jitter, Player buffers initially for 2-5 seconds, then starts display



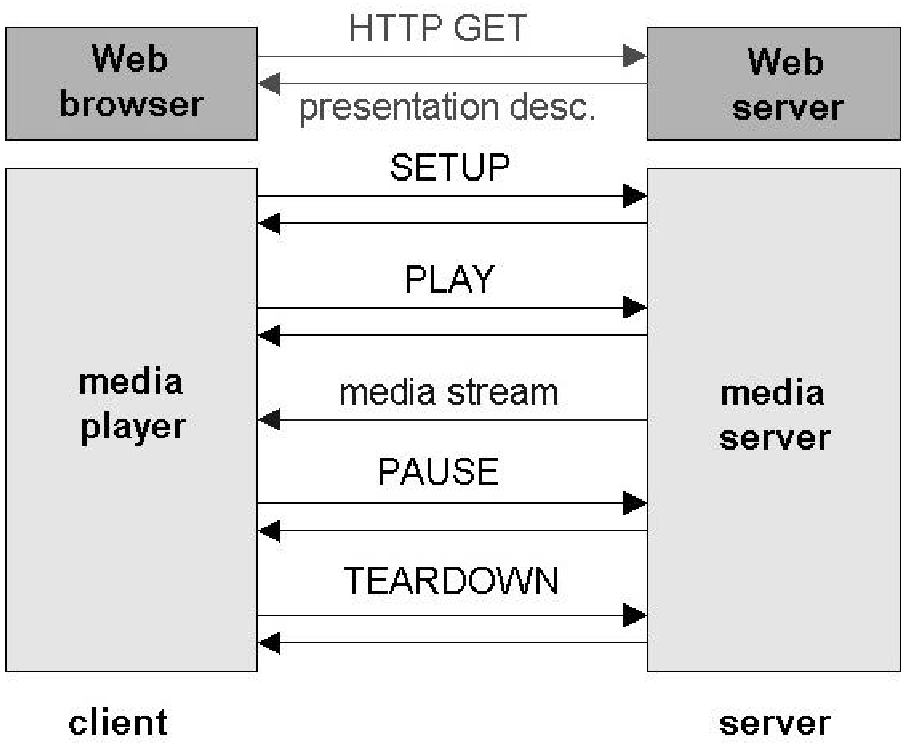
Options for Streaming

- Use TCP, and sender sends at maximum possible rate under TCP; retransmit when error is encountered; Player uses a much large buffer to smooth delivery rate of TCP

Real Time Streaming Protocol (RTSP)

- For user to control display: rewind, fast forward, pause, resume, etc…
- Out-of-band protocol (uses a separate connection for control messages)
- RFC 2326 permits use of either TCP or UDP for the control messages connection, sometimes called the RTSP Channel
- As before, meta file is communicated to web browser which then launches the Player; Player sets up an RTSP connection for control messages in addition to the connection for the streaming media

RTSP Operation



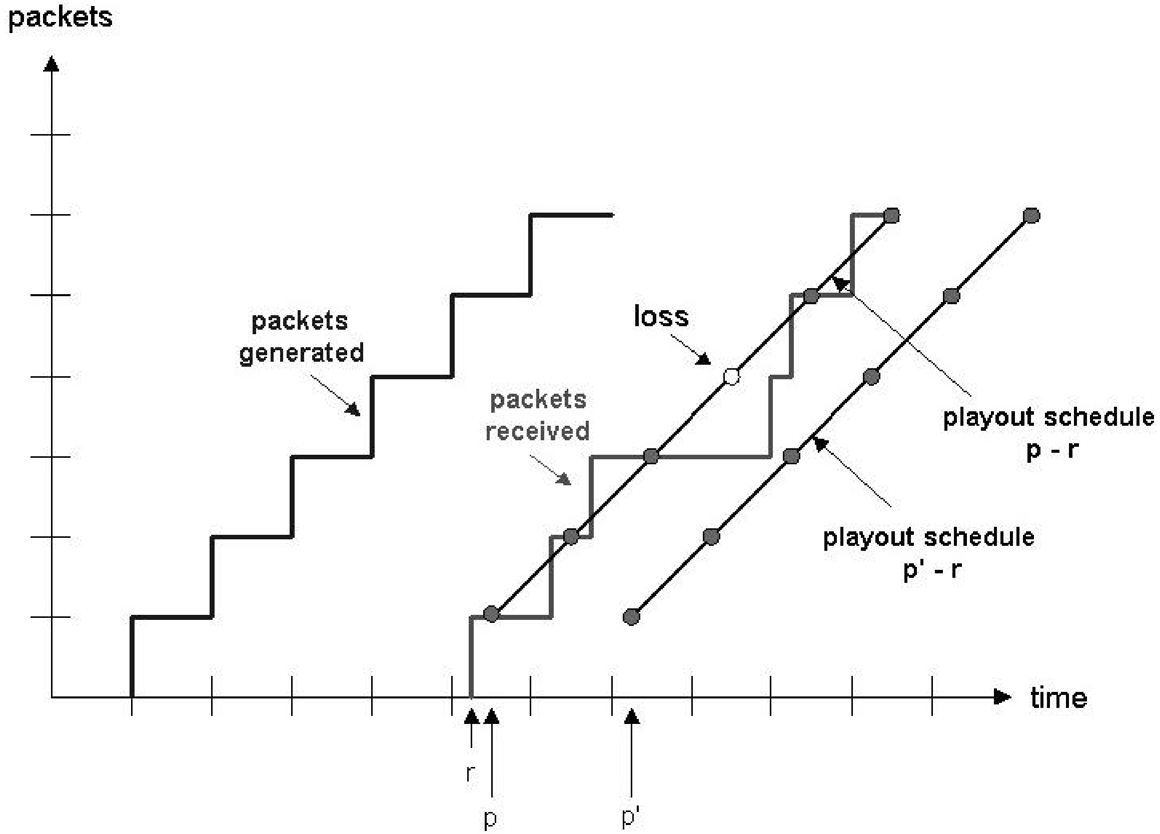
Real-Time Over IP's Best-Effort

- Internet phone applications generate packets during talk spurts
- Bit rate is 8 KBytes, and every 20 msec, the sender forms a packet of 160 Bytes + a header to be discussed below
- The coded voice information is encapsulated into a UDP packet and sent out; some packets may be lost; up to 20 % loss is tolerable; using TCP eliminates loss but at a considerable cost: variance in delay; FEC is sometimes used to fix errors and make up losses

Real-Time Over IP's Best-Effort

- End-to-end delays above 400 msec cannot be tolerated; packets that are that delayed are ignored at the receiver
- Delay jitter is handled by using timestamps, sequence numbers, and delaying playout at receivers by either a fixed or a variable amount
- With fixed playout delay, the delay should be as small as possible without missing too many packets; delay cannot exceed 400 msec

Internet Phone with Fixed Playout Delay



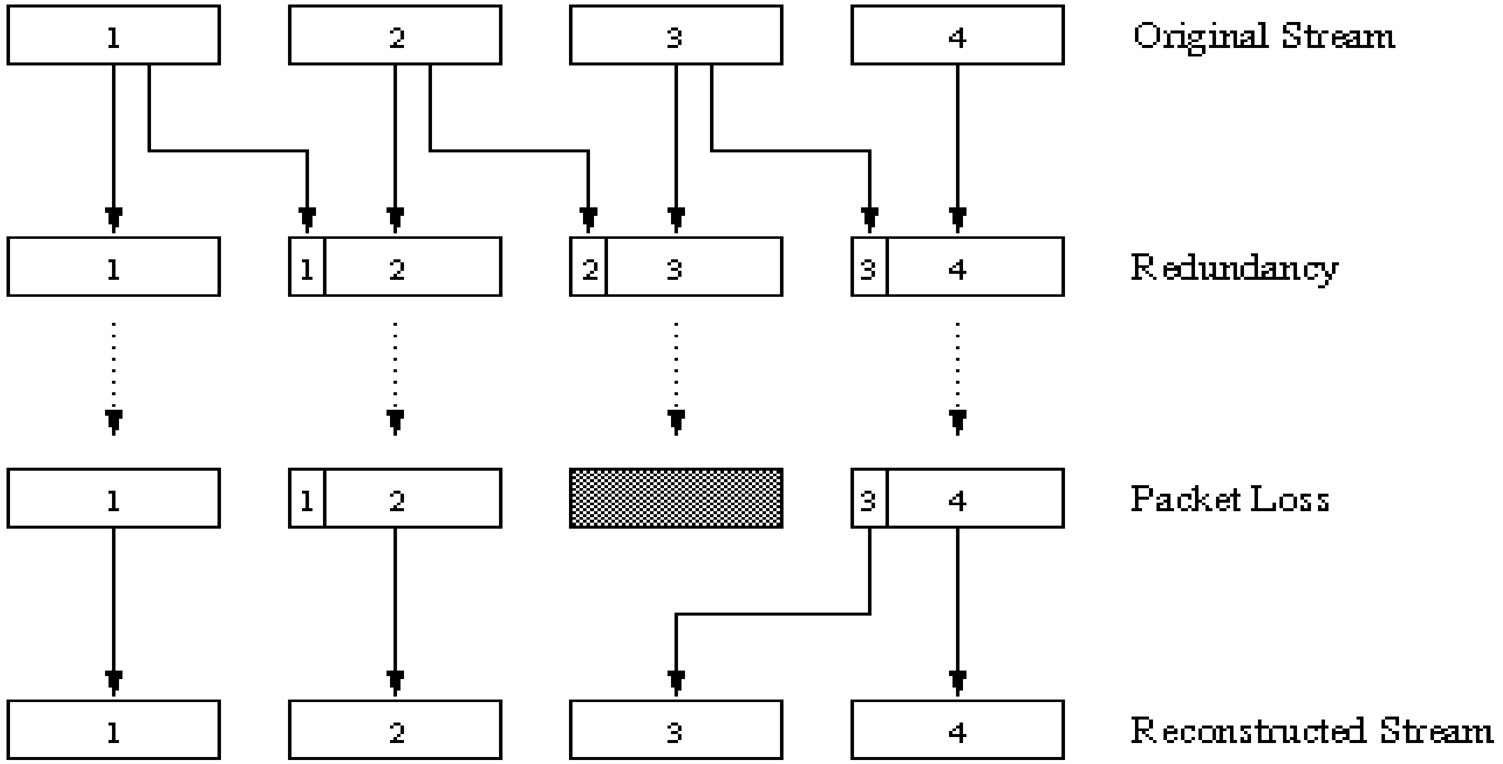
Recovery From Packet Loss

- Loss is in a broader sense: packet never arrives or arrives later than its scheduled playout time
- Since retransmission is inappropriate for Real Time applications, FEC or Interleaving are used to reduce loss impact.
- FEC is Forward Error Correction
- Simplest FEC scheme adds a redundant chunk made up of exclusive OR of a group of n chunks; redundancy is 1/n; can reconstruct if at most one lost chunk; playout time schedule assumes a loss per group

Recovery From Packet Loss

- Mixed quality streams are used to include redundant duplicates of chunks; upon loss playout available redundant chunk, albeit a lower quality one
- With one redundant chunk per group, we can recover from single losses

Piggybacking Lower Quality Stream



Interleaving

- Has no redundancy, but can cause delay in playout beyond Real Time requirements
- Divide 20 msec of audio data into smaller units of 5 msec each and interleave
- Upon loss, have a set of partially filled chunks