CS 640 Introduction to Computer Networks

Lecture15

Today's lecture

- Compression
  - Lossless compression
  - Lossy compression
- Physical layer

Why use compression?

- Pros
  - Less data to transfer
  - Application sees better throughput
  - Fewer bytes to store
  - Latency might be better
- Cons
  - Data quality can degrade
  - CPU overhead (harder to compress than to uncompress)
  - Latency might be worse

- Typically done at application or data link layer
  - HTTP compression (for dynamic pages too)

Huffman codes

- Core idea: use shorter bit sequences to encode symbols that occur often
  - Symbols can be characters or something else
  - Works for some types of data
- Example: "Ramabhadran"
  - Need 8 symbols in alphabet
  - Equal size encoding
    - 11*3 = 33 bits
  - Huffman encoding
    - 4*2 + 5*3 + 2*4 = 31 bits

Symbol	Code
a	00
R	010
m	011
b	100
h	101
d	110
r	1110
n	1111

Other methods

- Run length encoding
  - Encode AAABBCDDDD as 3A2B1C4D
  - Works well for faxes
- Dictionary based methods
  - Use codes for words occurring in a dictionary
  - Words have variable lengths (may actually be a phrase)
  - Dictionary needs to be known to both sender and receiver
  - Can be static or dynamic (based on the data to compress)
  - Lempel-Ziv uses dynamic dictionaries
  - Works well for many kinds of data

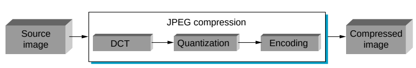
Today's lecture

- Compression
  - Lossless compression
  - Lossy compression
- Physical layer

Lossy compression overview

- Data reconstructed by the receiver is similar, but not identical to the data at the sender
- Can achieve higher compression ratios
  - User can control quality loss or compression ratio
- Used for images, audio and video
  - JPEG (images)
  - MPEG-2 video
  - MP3 audio
  - Many other formats exist (some proprietary)

JPEG

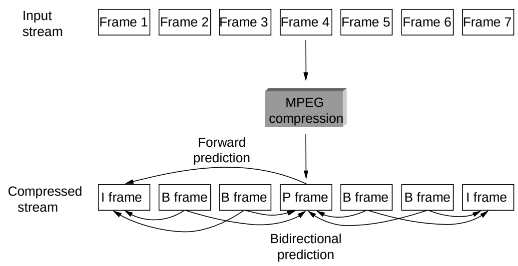


- Operates on blocks of 8*8 pixels at a time
- DCT transforms data w/o loss
  - Like transforming cartesian to polar coordinates
- Quantization drops small coefficients which represent visually unimportant information
- Encoding (Huffman+RLE)
- Compression factor ~ 30

MPEG – video compression

- Typically MPEG encoding too expensive to do online
- Video is a sequence of frames (e.g. 30/second)
  - JPEG exploits spatial locality within images (frames)
  - MPEG also exploits temporal locality – typically the next frame is somewhat similar (compression factor ~ 100)
- MPEG uses 3 types of frames
  - I frames can be decoded independently
  - P frames depend on the previous I frame
  - B frames depend on the previous and next I or P frame

MPEG frame types



CS 640

Compressing sound

- Represented as periodic samples
  - Phone quality 8 bit samples every 125 µs (64Kbps)
  - CD quality16 bit samples 23 µs (stereo 1.41 Mbps)
- MP3
  - Part of the MPEG standard
  - Divides the sound into frequency bands
  - Works on blocks of 64 to 1024 samples
  - Uses DCT, quantization, and encoding
  - Compression factor up to 12 (Layer III)

CS 640

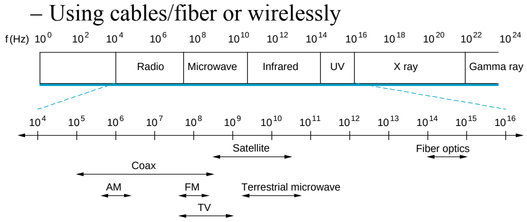
Today's lecture

- Compression
  - Lossless compression
  - Lossy compression
- Physical layer

CS 640

Physical layer

- Role: moving bits from one point to the other
  - Using electromagnetic signals or light
  - Using cables/fiber or wirelessly



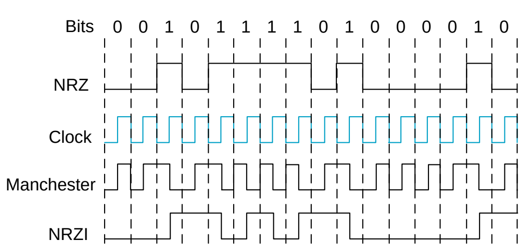
CS 640

Fundamental limit on throughput

- Shannon's theorem: $C = B \log_2(1+S/N)$
  - C is channel capacity
  - B is the width of the frequency band
  - S/N is the ratio of the power of the signal and the power of the noise
  - Actual solutions achieve throughput lower than C
- There are many ways of encoding bits into signals, theorem applies for all

CS 640

Signal encoding and synchronization



CS 640

4B/5B encoding

- 4 bits of data encoded with a 5 bit code
- Each code has at least one transition
- Efficiency 80%
  - Manchester has 50%

4 bit data	5 bit code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101