Lecture 02 - Information Theory and Networking

Information**:** what reduces uncertainty
surprisal of an even with probability p: l(p) = -log2(p) bits
Surprisal (or information content) quantifies the unexpectedness of an event, defined as the negative log-probability of an outcome (\(S(p)=-\log (p)\))

Example:

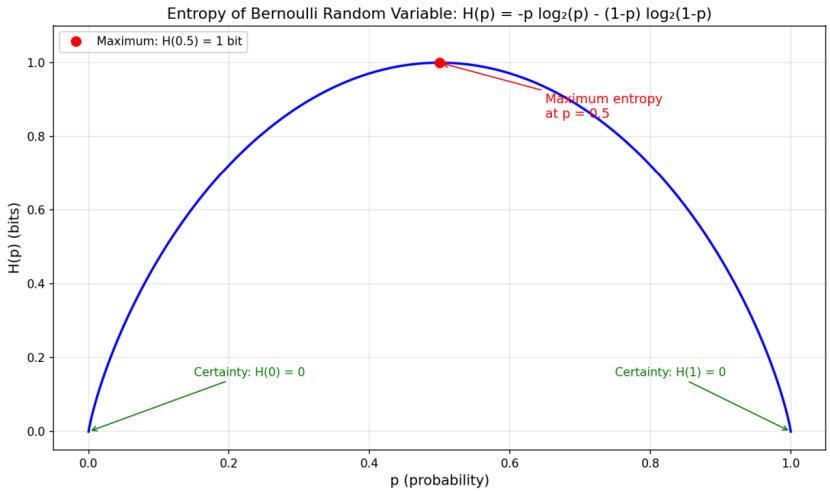
- Coin flip (p=0.5): l= 1 bit
- rolling a 6 (p=⅙): l=2/58 bits

Shannon Entropy: average information per message

H(X) = -∑n, i=1 pi log2(pi) bits

Represents:

- Minimum bits needed to represent outcome
- Fundamental limit on data compression
- Average amount of information per message
    - NOTE: INFORMATION MAKES UP A MESSAGE
- Fair coin: H = 1



Entropy: H(P):
Units of information:
Basic units:

- Bit
- Byte
- Octet

SI prefix:

- Kilo
- Mega
- Giga
- Tera

Binary prefixes:

- Kibi
- Mebi
- Gibi
- Tebi

Data =/ information

information= actual content (measured in entropy)
Data = how we represent it (measured in bits/bytes)
difference= redundancy, overhead, metadata

Tradeoffs = efficiency vs accessibility vs reliability

Example: english text

- entropy : 1.5 bits/character
- ASCII: 8 bits/char (19% efficient)
- compressed : 2 bits/char (75% efficient)

ASCII → american standard code for information interchange

- 7 bits per character …
- Extended ASCII → 8 bits

Key ranges:

- Control: 0x00-0x1F
- Digits: 0x30-0x39
- Uppercase: 0x41-0x5A
- Lowercase: 0x61-0x7A

Unicode and UTF-8:

Problem: ASCII only handles English
Unicode: 149,186+ characters

- Code points: U+0041 (A), U+4F60 ()

UTF-8: Variable-length encoding

- 1 byte: ASCII (U+0000 to U+007F)
- 2 bytes: Latin, Greek, Cyrillic, Arabic
- 3 bytes: Asian characters, emoji
- 4 bytes: Rare characters

Base64 encoding

Purpose: Represent binary data using ASCII printable characters
Alphabet: A-Z, a-z, 0-9, +, / (64 chars = 6 bits each)

Process:

1. Take 3 bytes (24 bits)
2. Split into 4 groups of 6 bits
3. Map to base64 characters

Efficiency: 33% overhead
Uses in email attachments, embedded images, API tokens

Text File Formats
Different goals → different formats

- JSON: Data interchange, self-describing, no schema needed
- TOML: Configuration files, supports comments
- YAML: Indentation-based, human-friendly configs
- CSV: Tabular data, simple but limited
- XML: Verbose but flexible, declining use

Magic Numbers (file signatures) → Bytes at file start that identify format
Purpose:

- Quick file type identification
- Prevents misinterpretation
- Used by file command

```
PNG:  89 50 4E 47 0D 0A 1A 0A
JPEG: FF D8 FF
GIF:  47 49 46 38
PDF:  25 50 44 46  (%PDF)
ZIP:  50 4B 03 04  (PK)
ELF:  7F 45 4C 46
```

IEEE 754 floating point → Standard for representing real numbers
Binary32 (float): [Sign:1][Exponent:8][Mantissa:23] = 32 bits

- Range: ±3.4 × 1038
- Precision: ~7 decimal digits

Binary64 (double): [Sign:1][Exponent:11][Mantissa:52] = 64 bits

- Range: ±1.7 × 10308
- Precision: ~15 decimal digits

Instruction Set Encoding
Principle: Frequent operations → fewer bits (like Huffman coding)

x86 variable-length encoding

- NOP: 1 byte (very common)
- PUSH EAX: 1 byte (common)
- MOV EAX,5: 5 bytes (less common)

x86 (CISC):

- Variable-length
- Better code density
- Complex decoding

ARM (RISC):

- Fixed 32-bit
- Simple decoding
- Less code density

Network Protocol Formats → Messages aligned to 32-bit boundaries

Similar to file formats:

- Magic numbers → Version/type fields
- Length fields → Total Length, Data Offset
- Checksums → Data integrity
- Variable sections → Options

Key differences:

- Ephemeral (not stored long-term)
- Strict size limits (MTU ~1500 bytes)
- Performance critical (billions/sec)
- Must maintain backward compatibility

Why 32-bit alignment? Hardware efficiency, simpler parsing

Audio: analog to Digital

Sampling → bigger bits/section better

Fastest frequency → Nyquist frequency

1. Sampling: Measure amplitude at regular intervals
    - Sampling rate (Hz): samples/second
    - Nyquist theorem: Must sample at ≥ 2× highest frequency
2. Quantization: Round to discrete values
    - Bit depth: bits per sample
    - More bits = less noise

Audio Data Rates

CD Quality:

- 44.1 kHz, 16 bits, stereo
- Data rate: 44,100 × 16 × 2 ≈ 1.4 Mb/s
- 1 minute ≈ 10.5 MB

Voice (telephone):

- 8 kHz, 8 bits, mono
- Data rate: 64 kb/s

Compression:

- Lossless (FLAC): 40-60% of original
- Lossy (MP3): 128-320 kb/s (10-30× compression)

Images

1920×1080 RGB image:

- Pixels: 1,920 × 1,080 = 2,073,600
- 24 bits/pixel (8 bits per R, G, B)
- Uncompressed: 6.2 MB

Formats:

- PNG: Lossless, 2-4× compression (photos), transparency
- JPEG: Lossy, 10-50× compression, no transparency

Typical file sizes (1920×1080):

- PNG: 0.5-3 MB
- JPEG (high): 200-500 KB
- JPEG (medium): 50-150 KB

Video

Uncompressed 1080p @ 30fps:

- 6.2 MB/frame × 30 fps = 186 MB/s = 1.49 Gb/s
- Completely impractical!

Compression techniques:

- Intra-frame: Compress each frame (like JPEG) - 10-20×
- Inter-frame: Store only differences between frames - 5-10×
- Total: 50-200× typical

Typical compressed rates:

- YouTube 1080p: 3-5 Mb/s
- Netflix 4K: 15-25 Mb/s
- Blu-ray: 20-40 Mb/s