Communication Concepts, Information Theory, Math and Patterns

Unit 03 - Hands-On Networking - 2018

Prof. Dr.-Ing. Thorsten Herfet, Andreas Schmidt, Pablo Gil Pereira

Telecommunications Lab, Saarland Informatics Campus, 19th Feb. 2018

Recap

- Historical and modern **telecommunication systems**.
- Transmissions use certain **symbols** to transfer meaning.
- **Encoding** used to transmit written language.
- In **networking**, there are **challenges** galore.

At the Train Station

Through computer scientist's glasses...

- *Party A* communicates
- with *Party B*
- with *Purpose P*
- in *Environment E*
- and under *Requirements R*

Example: Alice (*A*) asks Bob (*B*) for the time (*P*) at the train station (*E*).

What are the Requirements (*R*)?

- *B*'s answer should be valid (R1).
- *E* should not be too noisy (R2).
- *A* and *B* should talk same language (R3).
- *A* has to say "Good day, may I ask you a question" first (R4).
  Because otherwise *B* will think he is impolite and refrain from answering.

Communicate

Meaning (information) is transmitted through **something else (symbols)**, which is built using e.g. **words (alphabet)**.

Examples

- Information.
  Time of day. Sensor readings. Thought: "This is interesting."
- Symbols.
  11 o'clock. Tomorrow. In general: Sentences from a language.
- Alphabet.
  All words/numbers used to describe time information. In general: Language XYZ

Time flies by...

Processing Delay (D_s)

- Enquirer: Phrase the question.
- Responder: Understand question. Look at the watch. Phrase answer.

Transmission Delay ($D_t = \frac{\text{Data Length}}{\text{Data Rate}} = \frac{L}{R}$)

- Enquirer: What time is it?
- Responder: Eleven o'clock.

Propagation Delay ($D_p = \frac{\text{Distance}}{\text{Speed of Propagation}} = \frac{d}{c}$)

- Natural bound: Speed of sound / light etc.
- Distance between mouths and ears.

Round-Trip Time (RTT)

- Time between asking the question and receiving the answer.
- $RTT = 2 \cdot D_p + 2 \cdot D_t + D_s$

Loss

What can cause misunderstandings?

- A cannot understand B because a train is approaching.
- B has a horrible accent.

How can this be solved?

- B repeats what he said before.
  Later: *Automated Repeat reQuest (ARQ)*.
- B says something like "it's eleven o'clock on the minute".
  Later: *Forwarding Error Coding (FEC)*.

Formal Sizes:

- Packet Loss Rate
- Bit Error Rate

Data Rate

What are the possible answers(A)?

- e.g. precision using hours per day, and quarter-hours of day.

How many bits would I need to encode a certain answer (s)?

- $s = ld(|A|)$

How long does it take to transmit one symbol (t)?

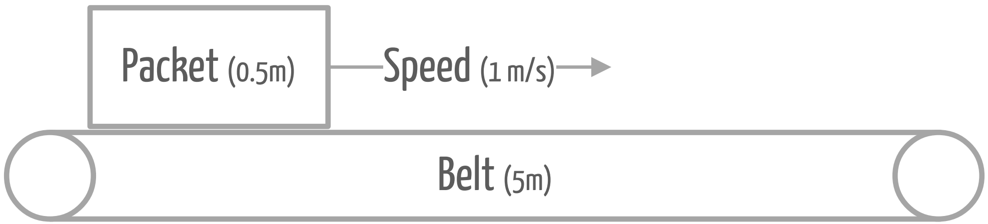
- See last slide.

Formal Sizes:

- Data Rate: $r = \frac{s}{t}$, $[s] = bit$, $[t] = sec$
- Throughput
- Goodput

Links

Analogy: Packet Conveyor Belt.



Questions:

- How long does it take for a packet to traverse the belt?
- How large is the throughput (packets per second) if we send a single packet?
- How many packets can fit the belt / can be sent back to back?
- What happens if we send more?

Data Network Links

Base Sizes:

- Length $\equiv$ **delay** $D, [D] = sec.$
- Speed $\equiv$ **throughput** $B, [B] = bps.$
  Also called bandwidth or data rate.
  Careful: Bandwidth is most of the times given in Hertz and a frequency size.
- Packet size $\equiv$ **packet length** $L, [L] = byte.$
  Largest possible packet size: **Maximum Segment Size** ($MSS, [MSS] = byte$).

Composed Sizes:

- Number of *bytes* a link can contain at any time:
  Bandwidth-Delay Product ($BDP = B \cdot D, [BDP] = byte$).
- Number of *packets* a link can contain at any time:
  Window ($N = \frac{BDP}{MSS}, [N] = 1$)

Data Network Link - By Example

- Given: $MSS = 1280byte$, $D = 200ms$, $B = 50Mbps$.
- How big is the BDP? How many packets can be sent?

$$BDP = B \cdot D = 50Mbps \cdot 200ms = 10Mbit = 1280kByte$$

$$N = \frac{BDP}{MSS} = 1024$$

Making Connections

Communication can be **connectionless** or **connection-based**.

- Consider our train station example again!
- Bob wanted Alice to establish a connection first.
  - Alice had to say "Hello" first, so that Bob isn't offended (and paying attention).
  - Usually people say "Bye" to each other when stopping conversation.
    Closing communication. Free resources.
  - Speaking with people involves polite start and end sentences.
    → **Connection-Based** transmission
- There are regular announcements (*"Please mind the gap"*):
  - Everyone who is interested can tune in and listen.
  - There is neither an establishment nor a teardown routine.
    → **Connectionless** transmission

Symbol Rate

Potential Problem: Consider Alice speaking incredibly fast (think of chipmunks). Bob cannot understand her, because his hearing system is overwhelmed.

- For many physical transmission systems, the rate at which symbols are generated is important and has to be agreed on:
- Rate automatically selected / conveyed by the transmission systems or standardized among interoperating systems.
- Has to be specified when tuning in to a communication.
  e.g. baud rate for a serial connection between PC and embedded board
- Mathematically speaking:
  - Symbol Duration: T_s
  - Symbol Rate: $f_s = \frac{1}{T_s}$

In-Order

- Assume Alice asks Bob for three different things using three separate letters (answers in brackets):
    - When did you receive this letter? *(02nd March)*
    - When did the HON course start? *(19th February)*
    - When is the HON exam? *(27th March)*
- Bob answers, but does not incorporate the original question asked.

Problems

- Post office might reorder answers.
- Alice might associate wrong answer with her questions.

How to solve this?

Solution: Include request. Identify messages (e.g. sequence numbers).

Communication Building Blocks

Technologies

Media

- Air
- Water
- Copper
- Glass
- Plastic

Approaches

- Optical
- Acoustic
- Electrical (wired and wireless)

Communication Systems

- Combine *Media* and *Approach* to a general process able to transmit a symbol.
- Examples: Telegraph, Telephone, Copper Cable, Fibre Optics, Fire Signals, ...

> More on this in detail in later lectures.

HON

Networks

- Communication Systems ...
    - are itself useful for transmitting.
    - require an established network for access and connectivity.
- Examples:
    - Fire Signal Network
    - PSTN
    - Internet

Services

- Users want something to be done (e.g. telling someone something).
- Doing something is implemented by services.
- Examples:
    - Tell Troy has fallen.
    - Configure my router in Troy.
    - Book a hotel in Troy.
    - Check the weather in Troy.

HON

Protocols

Definition (Wikipedia): *In telecommunications, a communication protocol is a* ***system of rules*** *that allow* ***two or more entities*** *of a communications system to transmit information via any kind of variation of a physical quantity. These are the rules or standard that defines the* ***syntax****,* ***semantics*** *and* ***synchronization*** *of communication and possible* ***error recovery methods****. Protocols may be implemented by hardware, software, or a combination of both.*

Protocol Specification:

- *Service* to be provided to a user of the protocol.
- *Assumptions* to be made about environment and participants.
- *Vocabulary* to be used by participants.
- *Encoding* to map elements of the vocabulary to transmittable symbols (flags etc.).
- *Behavioural Rules* to be followed by participants.

Train Station Time Query Protocol

- **Service:** Get to know the current time by asking someone.
- **Assumptions:**
- Participants can hear each other and talk the same language.
- **Vocabulary:**
- Question words.
- Time information bits (numbers (0-24), modifiers (half past), well-known times (noon, midnight), imperial specialities (a.m., p.m.)).
- **Encoding:**
  - Sounds (*W-H-A-T T-I-M-E I-S I-T*, *N-O-O-N*).
  - Gestures (*point at wrist*).
- **Behavioural Rules:** Ask politely!

Important: Define an acronym, e.g. Train Station Time Query Protocol (TSTQP).

Textual vs. Binary Protocols

Textual

`GET /resource HTTP/1.1`

- Humans can read and understand the transmitted data **easily**.
- Usually done using 7-bit ASCII characters (RFC20).
  Published 1969, but just recently became a standard in 2015.
- Inefficient (information-theory wise).
- Straightforward to extend.
  Depending on protocol grammar.

Binary

`0x450005dc` (Version: 4, Length: 20 bytes)

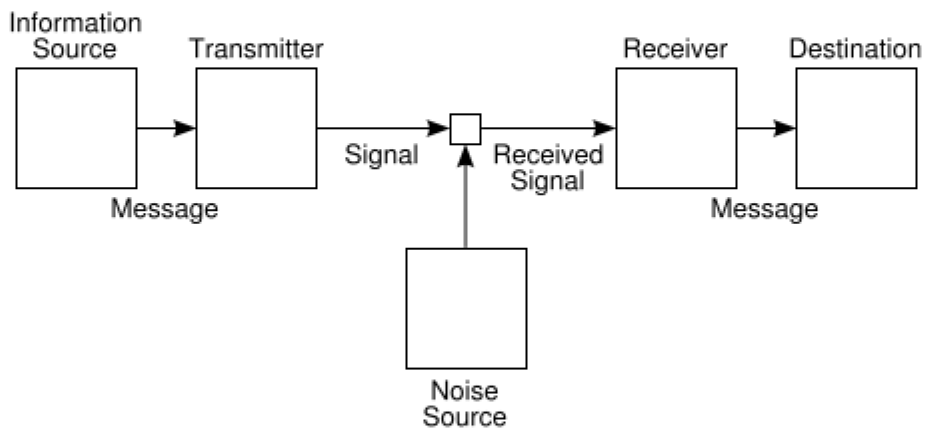
- Most humans **cannot** understand the transmitted data from the bits.
- Fields have differing length (number of bits) and type.
- Efficient.
- Potentially hard to extend.
  One way will be covered later.

⚠ Both are needed! None is generally **better** than the other.

A Quantum of Information Theory

Communication Systems

General description by Claude Elwood Shannon.



Source: *Wikipedia*

Just a Little Bit

Bit: Basic unit of information in *computing* and *digital communications*. Portmanteu of *binary digit*.

- Only two values: 0, 1.
- Compare Light Bulb: Either on or off.
- In Information Theory:
  1 Sh (Shannon)
  Bit: Data ↔ Shannon: Information

Source

Entropy

Origin: Thermodynamics. How irreversible is a system? *Ultimate chaos is inevitable!*

Shannon (Information Theory)
Receiver attempts to infer which message has been sent. Receiver is uncertain about what is sent, but anticipates certain things.

Intuition

- Likely Option $\rightarrow$ Low Entropy.
  e.g. white pixel of a document scan
- Unlikely Option $\rightarrow$ High Entropy.
  e.g. traffic light's yellow light
- More Options $\rightarrow$ Higher Entropy.
  e.g. traffic light vs. 7-segm. display

Quantification (in bits)

- Consider all options and call this Ω.
- Count them as $N = |\Omega|$.
- Calculate $log_2(N)$.
  The number of bits required to identify these options by a unique binary number.
- Entropy of Option x: $-p_x \cdot log_2(p_x)$

Inefficiency of Textual Protocols

Consider protocol involving a *command* field, which can be one of the following:

Retrieve (`GET`), Create (`ADD`), Modify (`MOD`), Delete (`DEL`)

How many bits are used for the textual and binary solution?

Textual

- 3 characters for each command.
- One ASCII char needs 7 bits.
  Often even 1 byte, but let's be fair.
- **Result:** $3 \cdot 7bit = 21bit$

Binary

- 4 different commands (0,1,2,3).
- 4 values require two bits.
  (00, 01, 10, 11)
- **Result:** $2bit$

Why are textual protocols used at all?

Efficiency is not your **only** parameter! **Compression** (see later) can bring efficiency.

Networking Math

Number Systems

Binary

- Symbols [0,1] (for EEs: High and Low)
- Base: 2
- Hardware uses this.
- Tedious to write.
- Protocol headers might have single bits of information.

Hexadecimal

- Symbols [0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]
- Base: 16
- Lazy humans that have to write binary numbers use this.
- Condensed presentation.

Decimal

- Symbols [0,1,2,3,4,5,6,7,8,9]
- Humans use this.

Example

- Binary: `101010`
- Hexadecimal: `2A`
- Decimal: `42`

Python

- Binary `0b???`
- Hexadecimal `0x???`
- Decimal `???`

Conversions

Hex <-> Bin

- One hex value <-> 4-bit sequence.
- Convert blockwise.
- Examples:
  - `0xF8` = `0b11111000`
    (`F` = `1111`, `8`= `1000`)
  - `0b01101010` = `0x6A`
    (`0110` = `6`, `1010` = `A`)

Hex / Bin -> Dec

- Multiply the symbols by the respective decimal basis and sum up.
- Example:
  - `0x1A` = 16 + 10 = 26
  - `0b1010` = 8 + 2 = 10

Dec -> Hex (n=16) / Bin (n=2)

- Process:
  1. Divide the decimal number by *n*
     Integer division.
  2. Write down the remainder.
     In hex/bin notation.
  3. Divide the result again by *n* (integer division).
  4. Repeat 2. and 3. until result is 0.
  5. The value is the digit sequence of the remainders from the last to first.
- Example:
  - 42 / 16 = 2 (Rest 10 = A)
  - 2 / 16 = 0 (Rest 2)
  - 42 = `0x2A`

Two's Complement

- Humans can decide that certain binary numbers are signed (have a sign +/-) or unsigned (always non-negative).
- Numbers are interpreted as two's complement as follows:
  - First bit indicates sign value (0 = positive, 1 = negative).

Example

`0b11010110` -> `1` * (-128) + `1` * 64 + `0` * 32 + `1` * 16 + `0` * 8 + `1` * 4 + `1` * 2 + `0` * 1 = -42

Quiz

❓ What is `0xCAFE` in binary?

A: `0b1110 1010 1110 1110`

B: `0b1100 1010 1111 1110`

C: `0b1100 1010 1100 1111`

D: `0b110 1101 0101 10111`

A Means to an End

- Endianness = byte order of multi-byte numbers
- Little-endian: LSB first, MSB last
- Big-endian: MSB first, LSB last

Example

- 4-byte value: 01 02 03 04 (hex, MSB to LSB)
- Big-Endian: 0x01 0x02 0x03 0x04
- Little-Endian: 0x04 0x03 0x02 0x01

Big-Endianness is also the **Network-Byte Order**

Quiz

Value of `0x00010000`? Assume little-endian.

A: 1	**B:** 256
C: 4096	**D:** 65536

Binary Operators

Bit A	Bit B	A & B	A | B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Examples

And	Or	Xor

`  0x1010`
`& 0x1100`
`= 0x1000`
	`  0x1010`
	`| 0x1100`
	`= 0x1110`
		`  0x1010`
		`^ 0x1100`
		`= 0x0110`

Shifts

Logical Shift

- Does not preserve the sign bit.
- All bits moved in direction.
- Vacant (now empty) bit positions are filled with zeros.
- Python: `<<` and `>>`

Arithmetic Shift / Signed Shift

- For right shifts: Instead of filling with 0s (logical), fill with sign bit.
- Not available in Python directly.

Ciruclar Shift

- Numbers wrap instead of filling.
- Not available in Python directly.

Examples

- `0b0001 << 1 = 0b0010` (left logical shift)
- `0b1010 << 1 = 0b0101` (left circular shift)
- `0b0001 << 4 = 0b0000` (left logical shift, if size is 4 bit)
- `0b1001 >> 1 = 0b0100` (right logical shift)

Quiz

What is the result of `0b11001 >> 3`?

A: `0b00111`

B: `0b11111`

C: `0b00011`

D: `0b10101`

Answer:

A: That's a circular shift.

B: That's a right arithmetic shift.

C: Correct!

D: That's nonsense.

Masking and Setting Bits

Check for last 4 bits of a bit string b

`b & 0b00001111`

Set Bit at position x (from right) to y in Bit-String b

- If `y=1`: `b |= (1 << x)`
- If `y=0`: `b &= ~(1 << x)`

Examples

- Set bit at position 2 to 1 in `0b1001`:
  `0b1001 | 0b0100 = 0b1101`

- Check if bit at position 4 is set:
  `((0b01010100 & 0b00010000) >> 4) == 1`

Metric Prefixes

- Usually orders of 1000:
  - Kilo (10^3), Mega (10^6), Giga (10^9)
  - Milli (10^{-3}), Micro (10^{-6}), Nano (10^{-9})
- In computer science (especially with bytes):
  - Kibi = 1024
  - Mebi = 1024^2

Careful: The industry likes to use metric prefixes but calculates with binary prefixes (factor 1024).

- For more information:
  - https://en.wikipedia.org/wiki/Metric_prefix
  - https://en.wikipedia.org/wiki/Kibibyte

Units

Bit

- Information.
- 8 bits are one byte.

Packet

- (More) Information.
- Sequence of bytes (seldomly bits).
- Variable length, even though parts have to be fixed.

Second

- Time.
- Used for: Round-Trip Time, Forward-Trip Time, Jitter, ...

Hertz

- Frequency.
- Used for: Packet Rate, Baud Rate, ...

Size(s) matter... even for computer scientists

Time Measures

- Latency/Delay: How long something takes in seconds.
- Forward-Trip Time (FTT): How long from source to sink?
- Round-Trip Time: FTT + way back. Sometimes 2*FTT.
- Jitter: variance of delay.

Data Rate

- $R = \frac{TransmittedData}{Time} \rightarrow [R] = bits/s$
- Throughput: Overall data rate.
- Goodput: Useful data rate (application data per time).
- Maximum Bandwidth: Maximally achievable data rate of a channel. Careful: Not Hertz!

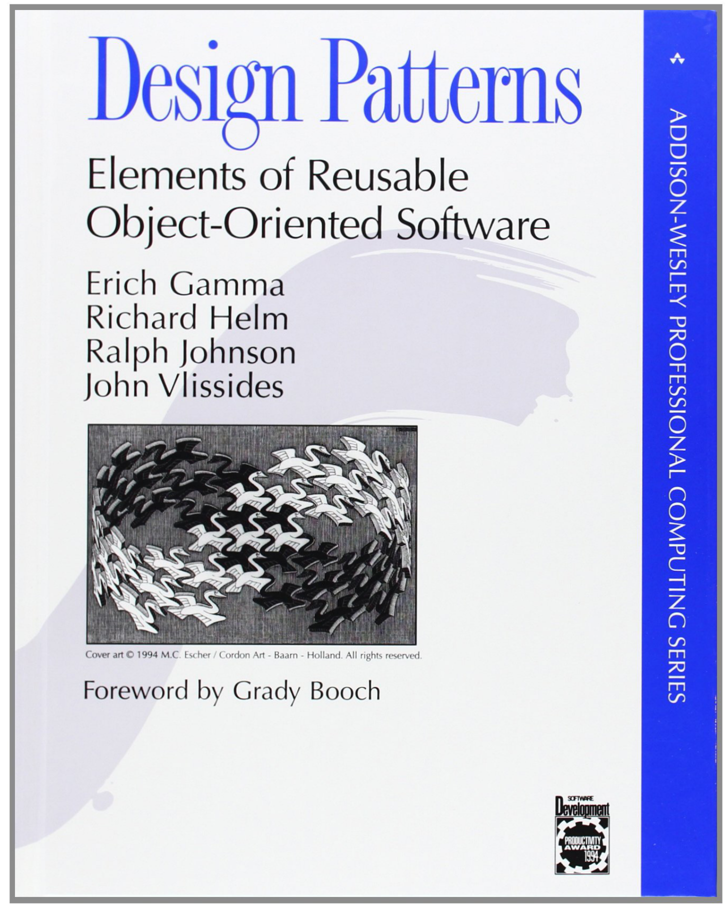
Loss

- Packet Loss Rate: $plr = \frac{LostPackets}{TransmittedPackets} \rightarrow [plr] = 1$ (usually in %)
- Bit Error Rate: $ber = \frac{WrongBits}{TotalBits} \rightarrow [ber] = 1$ (usually in %)

Patterns

Software Design Patterns

- **Motivation:** Computer Scientists tend to reinvent the wheel again and again.. and again.
- Mistakes and design weaknesses sneak in when reinventing.
- Other fields, e.g. architecture, have patterns for houses, bridges, etc.
- Proposed by the Gang-of-Four (GoF).



Pattern

Definition and Usage Advice

- In general: *Approach that can be used to solve common problems.*
- Should always be applied appropriately and only if required.
  Don't apply a pattern because you feel smart about knowing it.
- Nearly always has to be adapted to match the specific requirements.

Profile

Name: Suitable name to refer to it.

Context: When to apply?

Implementation: How to apply?

Benefits: Why is it good?

Drawbacks: What might be problems?

Variants: How is the pattern usually modified?

Similar Patterns: Which patterns can be used with this together?

Request-Response Pattern / Poll Pattern

Context: Simple, often stateless way to ask questions and get answers.

Implementation:

- Define a format for questions and answers.
- Enquirer sends requests, responder waits for it and answers accordingly.

Benefits:

- Simple protocol.

Variants:

- Long-Polling: Responses not sent immediately, but delayed until the responder has something to send (avoids poll-to-refresh).

Inverse Pattern: Push Pattern

Drawbacks:

- If the responder wants to tell the requester something, he has to wait for the request.
- Realtime updates require constantly asking the responder:
  "Has something changed." - "No." - "How about now?"

Buffer Pattern

Context:

- Persist data.
- Enforce order.
- Smoothing data rate.

Benefits:

- Simple.
- Universal.

Variants: Queue, Stack, Pipe, ...

Implementation:

- List, array, ordered-something implementation.
- Provide put() and get() methods.

Drawbacks:

- Requires storage.
- Requires fill-level management.

The Three-Letter Acronym Pattern

Context: Whenever you have to name a protocol, tool, project, ...

Benefits: Nobody will know what it is when first encountering your TLA.

Drawbacks: Nobody will know what it is when first encountering your TLA.

Implementation: Pick 3 letters (or digits) from ASCII and invent suitable words for it.

Variants:

- 4LAD (Four-Letter Acronym Definitions).

Similar Patterns:

- "Let it end with a P"-pattern.
- Recursive acronyms (GNU - GNU not Unix).

More information: RFC5513.

Wrap-Up

Questions?

Take-Home Messages

- Party A wants to communicate with Party B with Purpose P in Environment E and with Requirements R.
- **Information** is transmitted through **symbols over an alphabet**.
- **Delay**, **loss** and **data rate** are important quantifiable parameters for communication.
- Networks, composed of multiple communication systems, are used to provide a service. Participants have to follow a **protocol**.
- No networking without **math and numbers**.
- **Patterns** help to structure and recognize approaches.

Further Reading

- Kurose-Ross "Computer Networking" (Sec. 1.1, 1.3., 1.4)
- Gamma et al. "Design Patterns - Elements of Reusable Object-Oriented Software".