Computer Networks

Data Link Layer - Logical Link Control

Prof. Dr. Oliver Hahm

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering
oliver.hahm@fb2.fra-uas.de
https://teaching.dahahm.de

December 07, 2021

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- Address Resolution

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- Address Resolution

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- Address Resolution

Failure Causes

During the transmission of bit sequences on the physical layer errors may occur

They are typically caused by...

- Signal deformation
  - Attenuation of the transmission medium
- Noise
  - Thermal or electronic noise
- Crosstalk
  - Interference by neighboring channels
  - Capacitive coupling increases with increasing frequency
- Short-time disturbances
  - Cosmic radiation
  - Defective or insufficient insulation

Burst errors are more common than single bit errors

Typical BER values

POTS	$2 * 10^{-4}$
Radio link	$10^{-3} – 10^{-4}$
Ethernet	$10^{-9} – 10^{-10}$
Fiber	$10^{-10} – 10^{-12}$

The LLC sublayer ensures that errors are detected and handled

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- Address Resolution

Checksum

Checksum

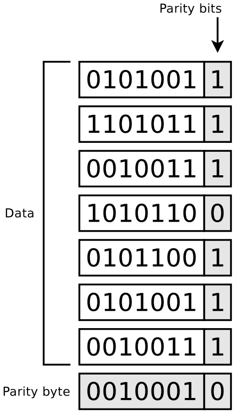
The checksum is calculated by a pre-defined algorithm for a block of data. They are typically used for the verification of the data integrity.

- For error detection, the sender attaches a checksum at each frame
- The receiver can now detect erroneous frames and discard them
- Possible checksums:
  - Parity-check codes
  - The polynomial code – Cyclic Redundancy Checks (CRCs)

Hamming Distance

- Each message (→ codeword) of n bytes contains m bytes of payload and r bytes of checksum
- Typically all 2^m data messages are allowed, but not all 2^n codewords are valid
- The minimum distance between two valid codewords is called the Hamming distance
  - In order to detect d errors, the distance needs to be $d + 1$
    - → d *flipped* bits won't create another valid codeword
  - In order to correct d errors, the distance needs to be $2d + 1$
    - → The resulting word with d *flipped* bits is still closer to the original codeword than to any other

One-dimensional Parity-check Code



- Well-suited for short blocks of data, e.g., 7-bit US-ASCII characters
- For each 7-bit section, an additional parity bit is calculated and attached to balance out the number of 1 bits in the byte
  - If the protocol defines even parity, the parity bit is used to obtain an even number of 1 bits in every byte
  - If odd parity is desired, the parity bit is used to obtain an odd number of 1 bits in every byte

$\Longrightarrow$ one-dimensional parity-check code

Two-dimensional Parity-check Code

	Data	Parity bits
Data	0101001	1
Data	1101011	1
Data	0010011	1
Data	1010110	0
Data	0101100	1
Data	0101001	1
Data	0010011	1
Parity byte	0010001	0

- For each byte exists, in addition to the parity bits, an additional parity byte
  - The content of the parity byte is calculated over each byte of the frame

  $\Longrightarrow$ two-dimensional parity-check code
- All 1-bit, 2-bit and 3-bit errors and most of the 4-bit errors can be detected via two-dimensional parity-check codes

Source: Computernetzwerke, *Larry L. Peterson*, *Bruce S. Davie*, dpunkt (2008)

Cyclic Redundancy Check (CRC)

- Bit sequences can be written as polynomials with the coefficients 0 and 1
- A frame with k bits is considered as a polynomial of degree $k-1$
  - The most significant bit is the coefficient of x^{k-1}
  - The next bit is the coefficient of x^{k-2}
  - ...
- Example: The bit sequence 10011010 corresponds to this polynomial:

$$
\begin{aligned}
M(x) &= 1 * x^7 + 0 * x^6 + 0 * x^5 + 1 * x^4 + 1 * x^3 + 0 * x^2 + 1 * x^1 + 0 * x^0 \\
&= x^7 + x^4 + x^3 + x^1
\end{aligned}
$$

Reminder: Polynomials in mathematics

A polynomial is an expression which consists of variables and coefficients and non-negative integer exponents

CRC Generator Polynomial

- The CRC specification defines a generator polynomial $C(x)$
  - The degree of the generator polynomial determines how many bit errors can be detected
- $C(x)$ is a polynomial of degree k
  - If e.g. $C(x) = x^3 + x^2 + x^0 = 1101$, then $k = 3$
    - Therefore, the degree of the generator polynomial is 3

The degree of the generator polynomial is equal to the number of bits minus one

Selection of common Generator Polynomials

- CRC-5
  Polynomial: $x^5 + x^2 + x^0$
  Representation: 0x05
  Application: USB
- CRC-8
  Polynomial:
  $x^8 + x^7 + x^5 + x^2 + x^1 + x^0$
  Representation: 0xA7
  Application: Bluetooth
- CRC-16-IBM
  Polynomial: $x^{16} + x^{15} + x^2 + x^0$
  Representation: 0x8005
  Application: Bisync, Modbus
- CRC-16-CCITT
  Polynomial: $x^{16} + x^{12} + x^5 + x^0$
  Representation: 0x1021
  Application: HDLC
- CRC-32
  Polynomial:
  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + x^0$
  Representation: 0x04C11DBB7
  Application: Ethernet

Example of Cyclic Redundancy Check (1/4)

Generator polynomial:	$x^5 + x^2 + x^0$	100101

- The generator polynomial has 6 digits
  - Therefore, five 0 bits are appended

Frame (payload):	10101
Frame with appended 0 bits:	1010100000

- The frame with the appended 0 bits is divided from the left only via XOR by the generator polynomial
  - Always start with the first common 1
  - The remainder is the checksum

XOR operation

```
1 XOR 1 = 0        0 XOR 1 = 1
1 XOR 0 = 1        0 XOR 0 = 0
```

The sender calculates the checksum

```
1010100000
100101||||
------vv||
  111100||
  100101||
  ------v|
   110010|
   100101|
   ------v
    101110
    100101
    ------
      1011 = Remainder
```

Example of Cyclic Redundancy Check (2/4)

fill the remainder

- The remainder must consist *n* bits and *n* is the degree of the generator polynomial
- If the remainder is shorter than *n*, it must be *filled* with zeros

- The checksum is appended to the payload
  - The length of the remainder must be *n* bits
  - *n* is the degree of the generator polynomial
- Result: 01011 will be appended to the frame
- Transmitted frame including checksum (code polynomial): 1010101011

Generator polynomial:	100101
Frame (payload):	10101
Frame with appended 0 bits:	1010100000
Remainder:	1011
Transferred frame (code polynomial):	1010101011

Example of Cyclic Redundancy Check (3/4)

Error-free reception

Transferred frame (code polynomial):	1010101011
Generator polynomial:	100101

- The receiver of the frame is able to verify, if the frame did arrive error-free
- By dividing (only via `XOR`) by the generator polynomial, transmissions with errors are detected
  - For division with XOR, always start with the first common 1
- If the remainder of the division is 0, then the transmission was error-free

The receiver verifies if the transmission was error-free

```
1010101011
100101||||
------vv||
  111110||
  100101||
  ------v|
   110111|
   100101|
   ------|
    100101
    100101
    ------
         0
```

Example of Cyclic Redundancy Check (4/4)

Reception of an erroneous frame

Transferred frame (code polynomial):	1110101011
Generator polynomial:	100101
Correct Transmission:	1010101011

- If the transmitted frame contains a defective bit, the remainder of the division via `XOR` not 0
- CRC *cannot* detect all errors

The receiver verifies if the transmission was erroneous

```
1110101011
100101||||
------v|||
 111110|||
 100101|||
 ------v||
  110110||
  100101||
  ------v|
   100111|
   100101|
   ------v
       101
```

Properties of CRCs

Most important characteristic

A polynomial code with r check bits will detect all burst errors of length $\leq r$

- If the error consists of a multiple of the polynomial code of the used CRC it will not be detected
- CRC-16-CCITT for example will detect
  - All single, double and three-bit errors
  - All error samples with odd number of bit errors
  - All error bursts with up to 16 bits (see above)
  - 99.997 % of all 17-bit error bursts
  - 99.998 % of all error bursts with lengths ≥ 18
- Calculation of CRC can be implemented by a simple shift register circuit in hardware

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- Address Resolution

Forward Error Correction (FEC)

- Error correction requires more redundant information to be added compared to error detection
- Upon error detection the frame typically needs to be retransmitted
- $\Rightarrow$ For somewhat reliable transmission channels simple error detection is cheaper
- $\Rightarrow$ For error-prone transmission media ($\rightarrow$ wireless communication) error-correction may be cheaper, because it reduces the amount of retransmissions
- (Forward) Error Correction can be realized via Hamming code
  - Named after the mathematician **Richard Wesley Hamming** (1915-1998)

Simple Example of Error Correction

Remember

In order to correct d errors a code needs a *Hamming distance* of $2d + 1$

- Assume a code with only four valid codewords
  - $w_1 = 0000000000$
  - $w_2 = 0000011111$
  - $w_3 = 1111100000$
  - $w_4 = 1111111111$
- $\Rightarrow$ The Hamming distance is 5
  - It can detect up to four bit errors
  - It can correct up to two bit errors
- Example:
  - If 0000000111 is received, the original must be 0000011111 (correct)
  - If 0000000000 is changed to 0000000111, the error is not corrected properly

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- Address Resolution

Reliable Transmission through Flow Control

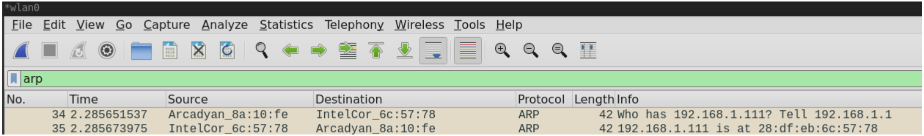
- Flow control allows the receiver to negotiate the transmission speed with the sender dynamically
  - Less powerful receivers or receivers under high load are not flooded with data
    - If a host receives data at a higher rate than it can handle it, data will get discarded and is lost
  - Concepts of flow control:
    - **Stop-and-Wait**
    - **Sliding-Window**

- Ethernet does not implement flow control mechanisms on Data Link Layer

Agenda

- Error Control
  - Failure Causes
  - Error Detection
  - Error Correction
- Flow Control
- **Address Resolution**

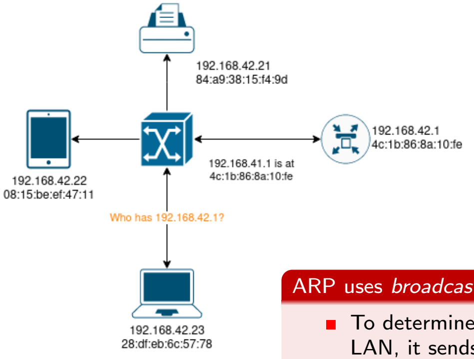
Address Resolution



- The network layer requires a mapping between physical and logical network addresses
- For IPv4 the Address Resolution Protocol (ARP) is used to resolve IPv4 addresses to MAC addresses [1]
- For IPv6 the Neighbor Discovery Protocol (NDP) accomplishes the same

[1] In fact, the original ARP specification, RFC 825, was written for IPv4 and Ethernet, but the functioning is not bound to IPv4 or any particular layer 2 protocol.

ARP and NDP



Simplified ARP message flow

NDP

In NDP routers and nodes can send proactively advertisements or be inquired via router and neighbor solicitations.

ARP uses *broadcast* messages:

- To determine the MAC address of a network device in the LAN, it sends out a MAC broadcast frame containing the IP address
- Each network device that receives the frame compares this IP address to the address assigned to it
- If a network device has this IP address, it sends an ARP response to the sender via unicast
- The original sender can now map the source MAC address of the response to the searched IP address

ARP Cache/Neighbor Cache

- The ARP cache is used to speed up the address resolution
  - It contains a table with these information for each entry:
    - Layer 3 protocol type (e.g., IPv4)
    - Layer 3 address (e.g., its IPv4 address)
    - Layer 2 address (MAC address)
    - Lifetime
  - The lifetime is set by the operating system
  - If an entry in the table is active, the lifetime is extended

The ARP cache can be displayed via `arp -n` or `ip neighbour`

```
# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.178.1            ether   9c:c7:a6:b9:32:aa   C                     wlan0
192.168.178.24           ether   d4:85:64:3b:9f:65   C                     wlan0
192.168.178.41           ether   ec:1f:72:70:08:25   C                     wlan0
192.168.178.25           ether   cc:3a:61:d3:b3:bc   C                     wlan0
```

Address resolution requests can be send manually via `arping`

You should now be able to answer the following questions:

- Which requirements need to be fulfilled to allow for error detection and correction?
- What is a CRC checksum and how does it work?
- For which purpose do we need ARP and NDP and how do they work?