Computer Network I

Reti di Calcolatori I

Università di Napoli Federico II – Scuola Politecnica e delle Scienze di Base
Corso di Laurea in Informatica

Riccardo Caccavale

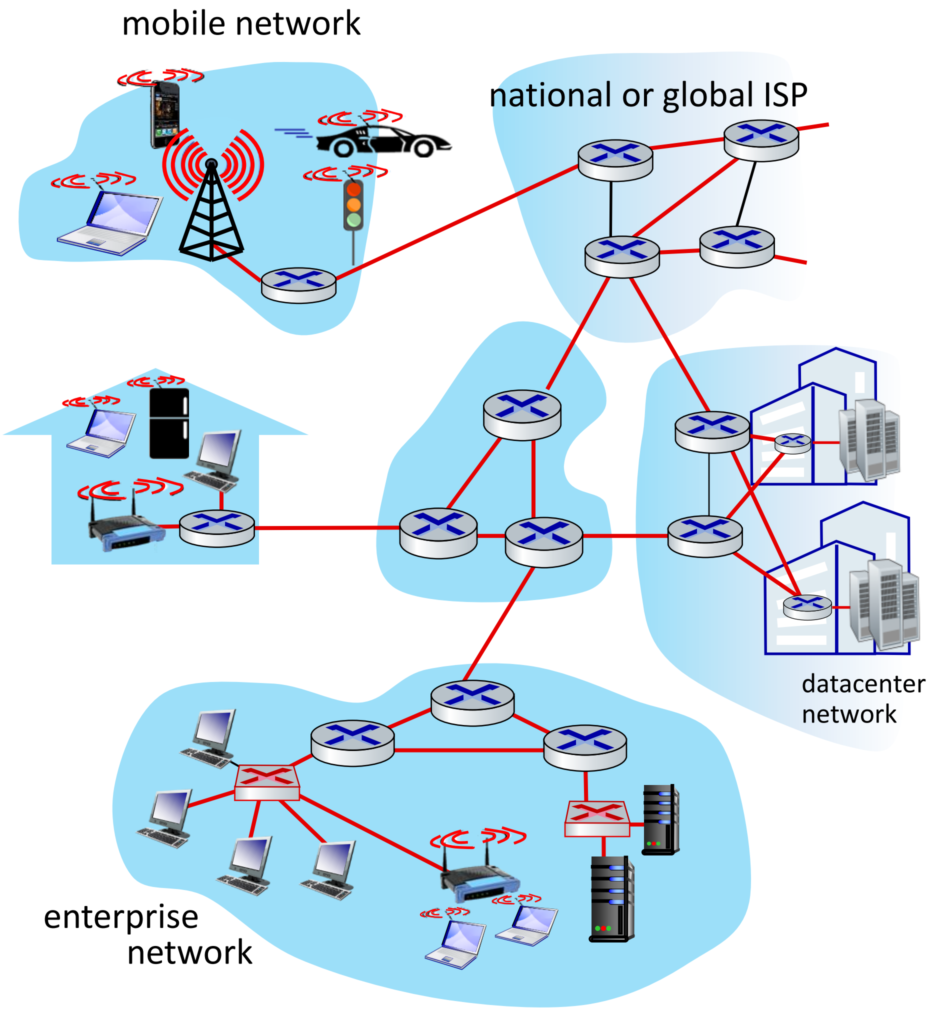
(riccardo.caccavale@unina.it)

Link Layer

From Network to Link & Physical

- We have seen that the network layer basically provides communication layer between two hosts (wherever they are).

- Link and physical layers provide communication between **two connected hosts**:
  - The **link layer** is the portion of the stack dedicated to the **transmission of packets over links** (transmission channels), from node to node of the network.
  - The **physical layer** is the portion of the stack that regulates the **structure of links** (transmission medium, connectors, cable types, etc.) and how bits are represented/transmitted along the link.

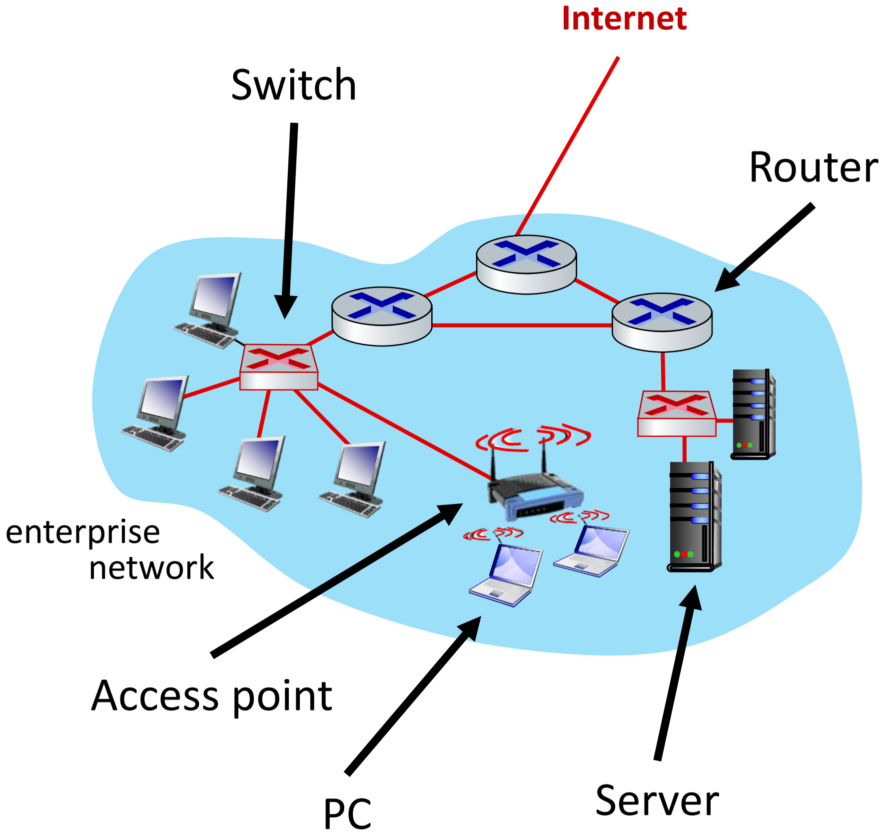


Link Layer

From Network to Link & Physical

- **Link and physical layers are strictly intertwined** (these are often grouped into a single layer called network access).
  - Some common protocols (e.g., Ethernet) cover both layers.

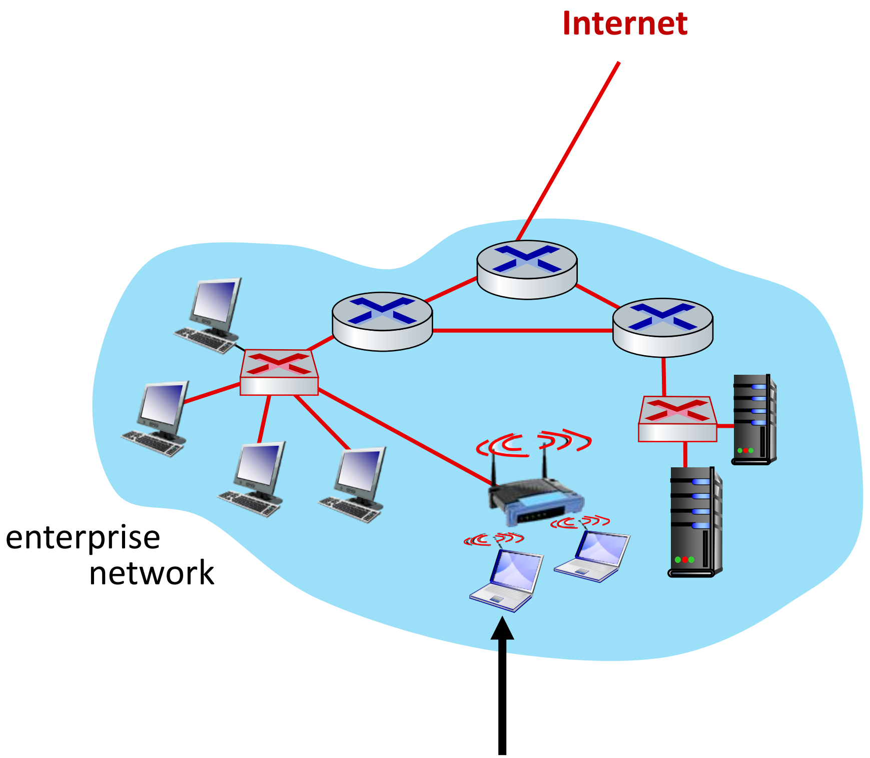
- Differently from the previous layers, **these 2 layers are present in all pieces** of the network infrastructure:
  - Nodes: hosts (PC, servers, etc.), routers, switches, hubs, WiFi access points, etc.
  - Links: wired (copper cables, optical-fiber), wireless (radiofrequencies).



Link Layer

From Network to Link & Physical

- To transmit an IP datagram, we must transfer it from source host to destination host by jumping over each of the individual links/devices along the path.
- To be transferred over a link, **datagrams are encapsulated into link-layer frames** depending on the specific type of link.
- This is **the last encapsulation**, frames are converted into signals and transferred over the link following its specification. Signals can be:
  - Electric pulses.
  - Light.
  - Radio waves.



In order to reach Internet, packets of this host have to cross 4/5 links (wireless and wired), 1 access point, 1 switch and 2/3 routers.

Link Layer

Services

- **Link layer may offer up to 4 services**: framing, link access, reliable delivery, error detection and correction.

1. **Framing**: to encapsulate **datagrams into link-layer frames** having the upper-layer datagram as payload and **specific header/tailer**. The structure of the frame depends on the link-layer/physical-layer protocols.

2. **Link access**: to define the rules **regulating the access to the link** by means of a **Medium Access Control** (MAC) protocol. Such rules depend on the link architecture:
   - For **point-to-point links** (single sender and single receiver), the **MAC protocol is simple** (or nonexistent). The sender can send a frame whenever the link is idle.
   - For **broadcast links** there is the so-called multiple access problem. Here, **the MAC protocol serves to coordinate the frame transmissions** of the many nodes.

Link Layer

Services

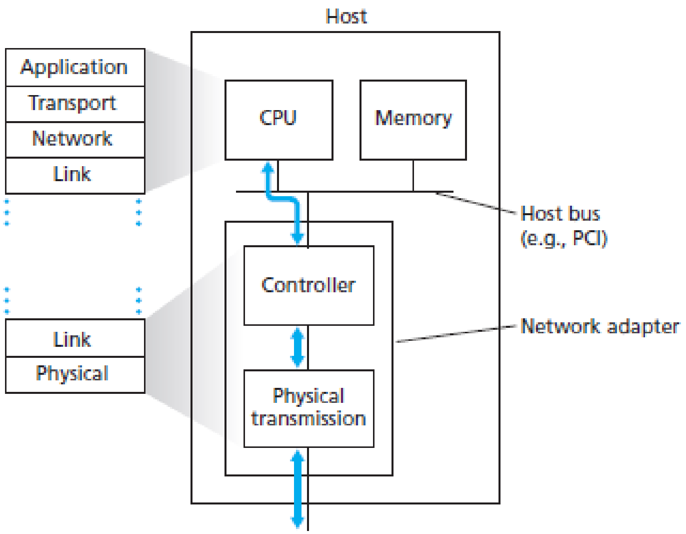
3. **Reliable delivery**: it guarantees that frames transmitted across the link are received.
   - A link-layer reliable delivery service **is often used for links that are prone to high error rates**, such as a wireless link, with the goal of correcting an error locally.
   - It may **produce strong overhead**. Since other application/transport protocols (e.g., TCP) are reliable, this feature **is not always implemented**.

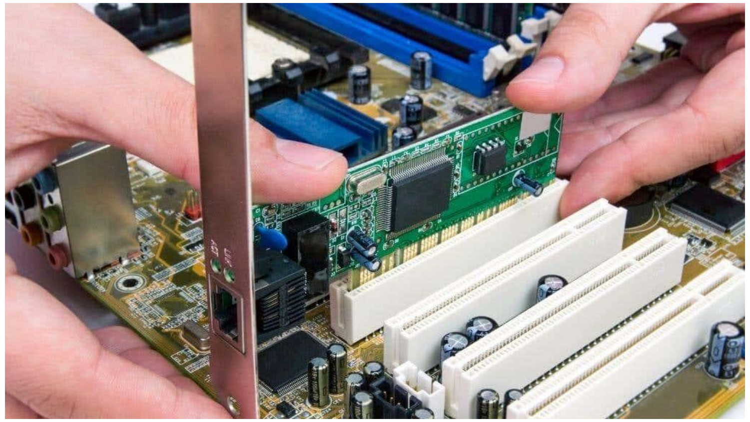
4. **Error detection and correction**: the link-layer **hardware can be affected by the bit flipping problem**. Many link-layer protocols implement more sophisticated (and hardware-embedded) checking/correction strategies in addition to the ones from upper layers (e.g., TCP and IP checksums).
   - Since there is no further encapsulation, these **checks basically cover the whole message**.
   - Hardware-embedded **checks may be a lot faster**.

Link Layer
Implementation

- In network devices (end-systems, routers, etc.) the link-layer functionalities are **implemented both software-side and (mostly) hardware-side**.

- In computers (i.e., end-systems) there are network adapters called **Network Interface Cards** (NICs) having a specialized chip (**controller**) to implement link-layer functionalities.
    - These **cards used to be separated from the motherboard** (plug-in into PCI slots), but recently this approach is changing (LAN-on-motherboard).



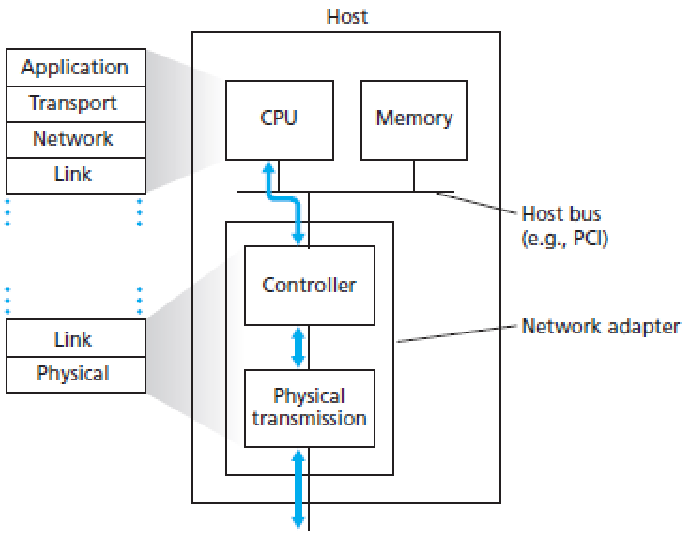


NIC plugged in a motherboard

Lattepanda mini-PC with embedded NIC

Link Layer
Implementation

- **Software**: runs on the CPU and provides high-level functionalities such as **addressing, interrupts/hardware management, handling of errors, encapsulation/decapsulation** of datagrams.
  - At link-layer we have an additional address, the **MAC address** that is used to identify the NIC.
  - The **datagrams must be stored into the memory** to be used by the upper-level protocol. They have to be:
    - **Retrieved the datagram** from memory during encapsulation (sender-side).
    - **Inserted the datagram** into the memory during decapsulation (receiver-side).



- **Hardware**: works as a typical I/O device, **converting the data into the signal** to be transmitted depending on the physical protocol (e.g., ethernet, wireless, etc.).

Link Layer
Physical Technologies

- The **link-layer protocol**, hence the resulting frame structure, **depends on the technology** of the physical link (physical layer).

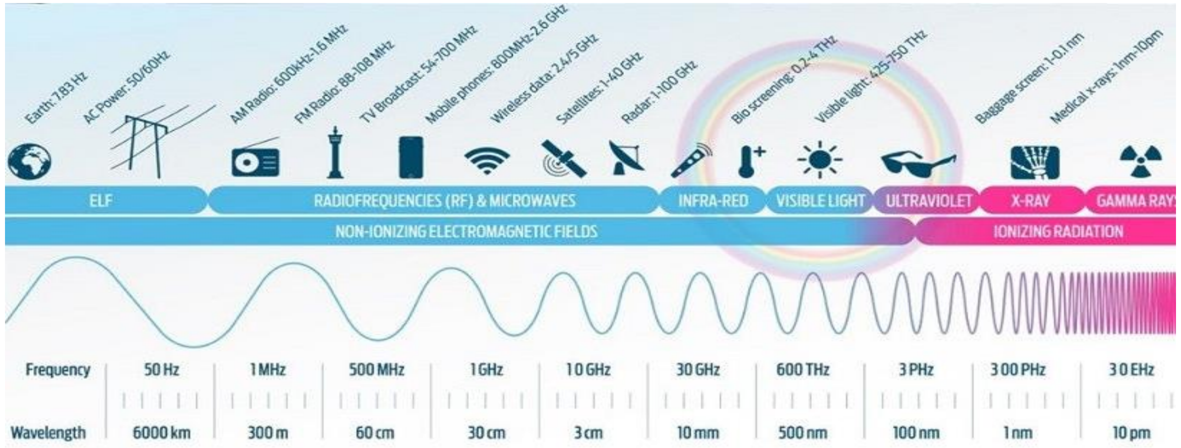
- Here some **common technologies**:
  - **Ethernet** 802.3: wired connection based on electric pulses over **4 twisted-pairs copper cables** (most common) or photons over **optic fiber** cables.
  - **WiFi** 802.11: wireless connection based on **radiofrequency** (2.4GHz, 5GHz, 6GHz).
  - **Bluetooth**: wireless connection based on **radiofrequency** (2.4GHz).

WiFi extender

Twisted-pairs

Bluetooth Adapter

Optic Fiber



WiFi/bluetooth Optic fiber

Link Layer

Physical Technologies

- Physical technologies are continuously evolving, their boundaries in term of distance/bandwidth are often updated.

- Different technologies have different pros and cons.

WiFi extender

Bluetooth Adapter

Twisted-pairs

Optic Fiber

Type	Signal	Distance	Bandwidth
Wireless	Radio	30-50 m	1.3 Gbps
Twisted pair copper cables	Electricity	30-100 m	1-25 Gbps
Fiber-optic cable	Light	100-200 km	1-400+ Gbps

Security, speed

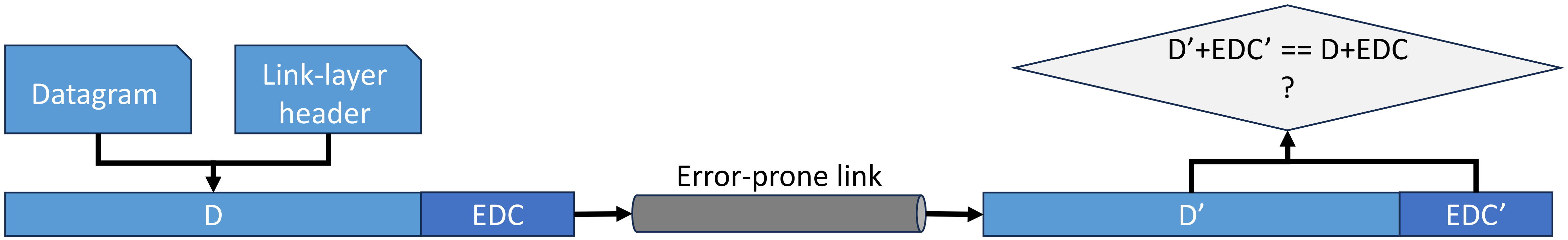
Interferences

Fragile, expansive

Link Layer

Error Detection and Correction: Problem Formulation

- In link-layer is possible to perform **bit-level error detection and correction**.
- Let's assume ***D* of size *d* being our data**, to protect it against bit errors we include in the message **some error detection and correction bits: *EDC***.
- The goal is to find **if received *D'* and *EDC'* differs from the original *D* and *EDC*** and, if so, **to possibly retrieve the initial D and EDC**.



- Note that even with the use of error-detection bits **there still may be undetected bit errors**, the receiver may be unaware that the received information contains bit errors.

Link Layer

Parity Check

- **Parity check** is perhaps the simplest form of error detection: **we include only 1 additional parity bit** to the message such that the total number of 1s among the d+1 bits of the message is even (even parity scheme) or odd (odd parity scheme).

Message	Priority bit (even scheme)	Priority bit (odd scheme)
1 0 1 0 1 1 0 0	0	1
1 1 0 1 0 0 0 0	1	0

- The **receiver need only to count the number of 1s** in the received d + 1 bits. If an odd number of 1-valued bits are found with an even parity scheme (or vice versa), the receiver knows that at least one bit error has occurred.

Link Layer

Parity Check

- Clearly **this approach only works if an odd number of bit errors have occurred**, but how likely is that?

- If the **probability of bit errors is small** and errors can be assumed to occur **independently from one bit to the next**, the probability of multiple bit errors in a packet would be extremely small, but **this is not the case of computer networks**.

- In practice, it has been observed that **errors are often clustered** together in "**bursts**" (not independent at all). Under burst error conditions, the probability of undetected errors using single-bit parity can approach 50 percent, so it is not really useful.

Link Layer

Parity Check (Two-dimensional)

- A possible way to improve this approach is to **use multiple parity bits**.

- The **two-dimensional parity** is a techniques in which message is divided into *n* rows and *m* columns, each of them having a specific parity bit.

- Here, receiver can detect that an error occurred, and also which bit, therefore **attempting a correction**.

- Two-dimensional parity can also detect (but not correct!) any combination of two errors in a packet.

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

1	0	1	0	1	1
1	0	1	0	0	0
0	1	1	1	0	1
0	0	1	0	1	0

Link Layer

Cyclic Redundancy Check (CRC)

- **Cyclic Redundancy Check** (CRC) is a widely used error detection technique. It works as follows:
  - In order to send *d* bits of data *D* the **sender and the receiver agrees on a pattern of r+1 bits called generator** (G), having the most significant bit (leftmost) at 1.
  - For a given piece of data D, the **sender will choose r additional bits** (CRC bits) to be appended to the message **such that, D+CRC is modulo-2 divisible by G** (no reminder).
  - The **receiver divides the D+CRC message by G**, if the reminder is zero, the message is correct, otherwise an error occurred.
  - This operation is implemented **by shifting G up to the most significant 1 bit of the message and performing a bit-wise XOR**.

Link Layer

Cyclic Redundancy Check (CRC)

- Let’s now consider a **simple example** of a 6-bits message D and a 3-bits CRC where:
  - d = 6
  - D = 1 0 1 1 1 0
  - r = 3
  - G = 1 0 0 1

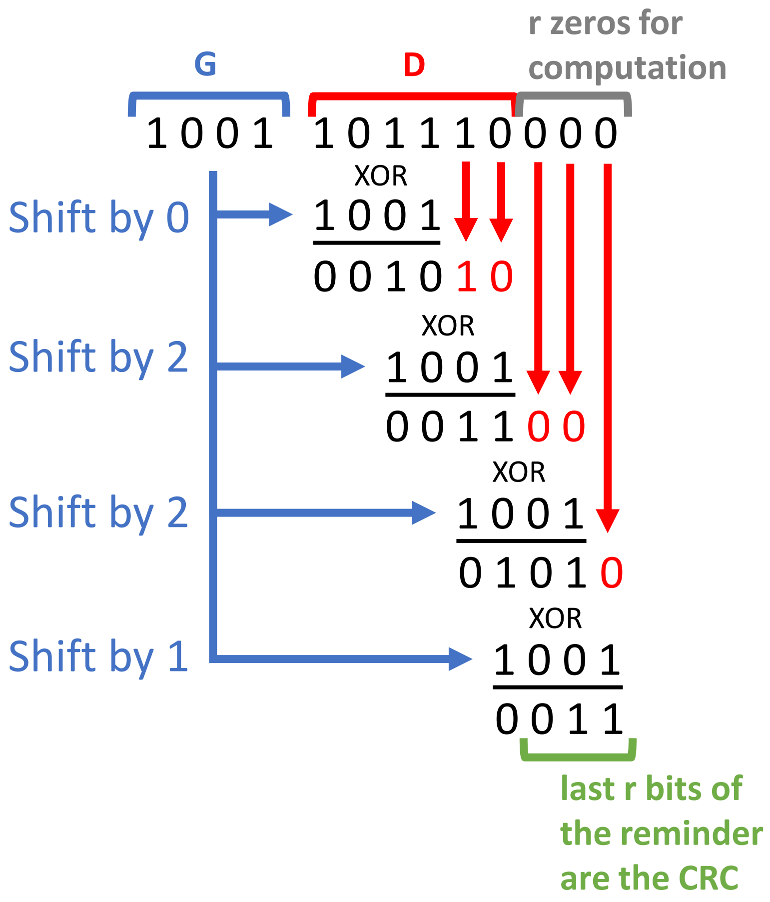
- On the **sender-side we have to create the CRC** from D and G, this CRC is attached to the message and transmitted to the receiver.

- On the **receiver-side we use D, G, and the received CRC** to understand if the received message is intact.

Link Layer

Cyclic Redundancy Check (CRC)

Sender-side calculation of CRC with r = 3

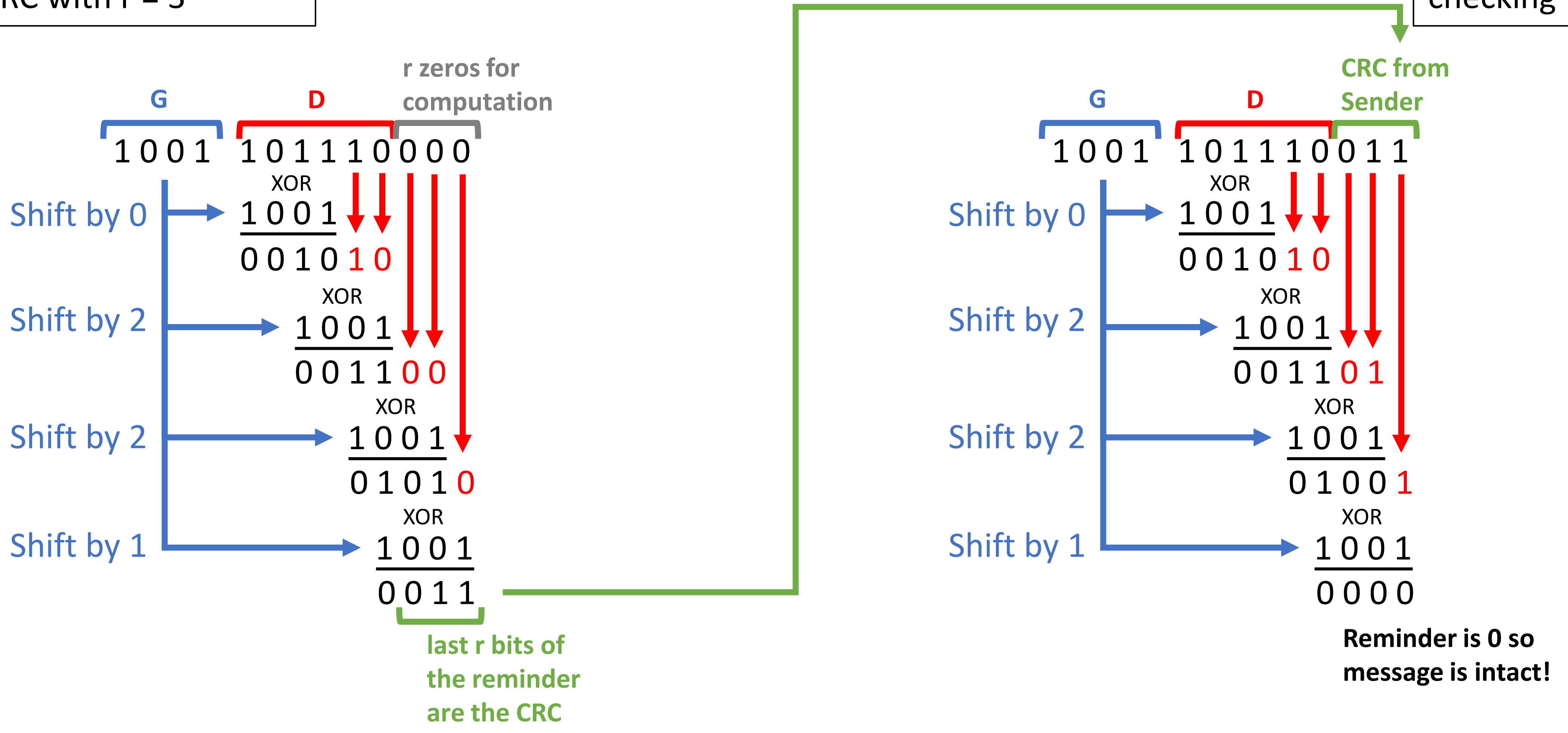


Link Layer

Cyclic Redundancy Check (CRC)

Sender-side calculation of CRC with r = 3

Receiver-side CRC checking with r = 3



Link Layer
Cyclic Redundancy Check (CRC)

- International **standards have been defined** for 8-, 12-, 16-, and 32-bit generators (i.e., r=8, r=12, r=16, r=32).
- The **CRC-32 32-bit standard**, which has been adopted in several link-level IEEE protocols, uses the following generator (33 bits):

$$G_{CRC\text{-}32} = 100000100110000010001110110110111$$

- Each of the **CRC standards can detect for sure burst errors of less than r + 1 bits**, while for **error greater than r + 1 there is a probability P** of finding the error which is:

$$P = 1 - 0.5^{r}$$

- Therefore, the **probability of finding the error increase with the increment of r**.