Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Computer Networking and IT Security (INHN0012)

Tutorial 5

Problem 1 Cyclic Redundancy Check (CRC)

The message `10101100` is secured using CRC as introduced in the lecture. The reduction polynomial $r(x) = x^3 + 1$ is given.

a)* What is the checksum length?

The length of the checksum in bits corresponds to the degree of the reduction polynomial, so here degree($r(x)$) = 3 bit.

b) Determine the checksum for the given message.

First, zeros are appended to the message deg($r(x)$) = 3: `10101100`**`000`**. Then divide by $r(x)$:

```
10101100000 : 1001 = 10111011 remainder 011
1001|||||||
----|||||||
001111|||||
  1001|||||
  ----|||||
  01100||||
   1001||||
   ----||||
   01010|||
    1001|||
    ----|||
    001100|
      1001|
      ----|
      01010
       1001
       ----
       0011
```

c)* Specify the transmitted bit sequence.

The transmitted bit string consists of the original message concatenated with the just calculated checksum: `10101100 011`.

The error pattern `00100000000` now occurs during the transfer.

d)* What is the received bit sequence?

The received bit sequence is the XOR of the transmitted bit sequence and the error pattern:

```
    10101100011
XOR 00100000000
---------------
    10001100011
```

e) Show that the transmission error is detected.

The received bit sequence is again divided by $r(x)$:

```
10001100011 : 1001 = 10011111 remainder 100
1001|||||||
----|||||||
0001110||||
   1001||||
   ----||||
   01110|||
    1001|||
    ----|||
    01110||
     1001||
     ----||
     01111|
      1001|
      ----|
      01101
       1001
       ----
       0100
```

The remainder `100` remains. With an error-free transfer, on the other hand, no remainder should have been left.

f)* Specify an error pattern that cannot be detected.

Multiples of the reduction polynomial cannot be detected, e.g. B. `10010000000`.

g) CRC was explicitly introduced in the lecture as an error-detecting, but not as an error-correcting code. Show that by means of CRC even 1 bit errors are not correctable in the concrete example of this task.

Argumentative The transmitted message is 11 bit long, i.e. there are a total of eleven possible 1 bit errors. However, the checksum is only 3 bit long, i.e., a maximum of seven bit errors could be distinguished since there are only seven non-zero residues. Thus, an unambiguous assignment of a remainder to a concrete bit error is not possible.

Proof by counterexample It is sufficient to find two different error patterns that produce the same remainder, because then it is not possible to infer unambiguously one of the two errors from this remainder. The error patterns `00001000000` and `000001000` both return the residue `001`.

Discussion: What about longer checksums?
An Ethernet frame has a size of up to 1518 B[1] including the checksum. This corresponds to $1518 \cdot 8 =$ 12144 possible 1 bit errors. The checksum is 32 bit long, resulting in $2^{32} - 1$ nonzero residues. Now, since $2^{32} - 1 > 12118$, a correction might be possible according to the above. However, this requires some more mathematics to be considered, since the number of possible residues and their unique assignability to 1 bit-errors depend on the structure of the reduction polynomial.
In fact, it is now the case that by means of the polynomial used in Ethernet 1 bit-errors really lead to unique residues. A correction would be possible with it, but is not used in practice with Ethernet.
So basically it depends on

- the choice of the reduction polynomial and
- the size ratio between user data and checksum

whether 1 bit errors are correctable by CRC.

[1] Jumbo frames are not considered in CNS

Problem 2 Switching and Forwarding

Given is the example network shown in Figure 2.1. In this problem, we will analyze the network with respect to the collision domains and the behavior of the network devices hub, switch and access point. The notebooks NB1 and NB2 are associated with the access point AP.

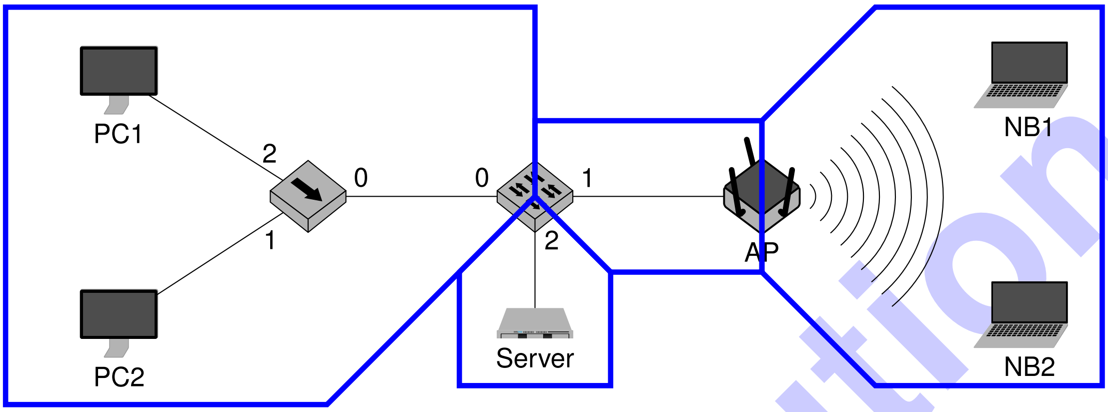

Figure 2.1: Network topology

a)* Draw the borders of all collision domains in the network in Figure 2.1.
It is important that not only the wire, but also the corresponding interface, are clearly marked as part of the collision domain.

b)* The network only consists of one broadcast domain. Why is that the case? Propose a change to the network that results in the presence multiple broadcast domains.

Routers separate broadcast domains. If the switch were replaced by a router, the network would consist of three broadcast domains (where the one including PC1 and PC2 and the one including the server are identical to the respective collision domain). Alternatively, severing a link would yield two separate local area networks with their own broadcast domains.

In the following subproblems we consider a message sent by PC1, destined for the server. The switching table of the central switch is given in Table 2.1. As a result of a previous transmission between PC2 and NB2, it already contains some entries.

MAC	Port	Reasoning
PC2.MAC	0	Previous connection
NB2.MAC	1	Previous connection
PC1.MAC	0	learned when the request is received by the switch, removed by subproblem h)
Server.MAC	2	learned when the reply is received by the switch
PC1.MAC	2	learned from malicious frame with spoofed SA from server

Table 2.1: Switching table of the switch of Figure 2.1.

c)* What is the state of the switching table, after the switch has forwarded the frame?
Update the table accordingly.

d) The server replies to the request. What is the state of the switching table after the reply reached PC1?
Update the table accordingly.

e) Which nodes will receive the reply sent by the server? What happens if a node that is not the intended destination receives the frame?

The switch knows the association PC1 ↔ Port 0 and will therefore only forward the frame to port 0. All nodes in the connected collision domain (here: PC1 and PC2) will receive the frame.
However, PC2 is not addressed as the destination. Upon receiving the frame, a NIC compares the destination address in the frame against its own address(es) and discards the frame because of an address mismatch. That is what will happen on PC2.

Now, we take a look at the connection between PC2 and NB2. NB2 wants to send some new data to PC2 and builds a frame on layer 2.

f)* Which addresses of which nodes must NB2 write into the header of the frame? What roles do these addresses have?

As a WiFi client, NB must address the access point directly. Therefore it addresses itself as source and transmitter of the message (NB2.MA is both SA and TA), the access point as receiver of the wireless transmission (AP.MAC = RA) and the actual destination node (PC2.MAC = DA).

g) Complete the header of the Ethernet frame, which is created by the access point to forward the data in the wired network. If a fields value is not specified in the instructions, choose a sensible value.

PC2.MAC	NB2.MAC	`0x0800`	Payload	FCS

An EtherType value of `0x86dd` for IPv6 is also possible. Transmission of data indicates some regular traffic over IPv4 or IPv6.

h) What would happen if the server was compromised and would send a spoofed frame, where the source address is set to the one of PC1? Update the switching table and explain the effect on the network.

(We ignore the destination address and the actual delivery of the frame)
The switch would update the entry for PC1.MAC and map it to port 2, where the server is connected. The switch would then forward all traffic destined for PC1 to the server. PC1 would no longer receive frames from the switch.
However, messages sent from PC2 to PC1 form a special case, since these two hosts are on the same collision domain. Therefore, PC1 would receive such messages directly on the local segment, but the switch would also forward them to the server.