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 ALOHA

ALOHA (Hawaiian: „Hallo“) is one of the oldest media access methods and was developed in 1971 at the University of Hawaii to connect the Hawaiian Islands to a central switching station via a radio link. The two communication directions from the islands to the switching station and back were separated by frequency division duplex (FDD). Controlling media access was extremely simple: as soon as a transmitter received data, it was allowed to start transmitting. However, as no directional antennas were used and all transmitters on the islands used the same frequency, collisions could occur if two transmissions overlapped in time.

Two years later, slotted ALOHA was introduced, in which the transmitters were only allowed to start transmitting at the beginning of fixed time slots. The switching station transmitted a clock signal on the return channel for synchronization.

We now want to define our own strategy, which we call p-persistent Slotted ALOHA. If data is available, a station transmits with probability p in the next slot or delays the transmission by one slot with probability $1-p$. The following initial situation is given:

- Initially, only some of the main islands are connected to the network, i. h. $n \leq 8$[1].
- All n users are saturated, i. e. there is always data to send.
- Each user starts sending with probability p in the next possible time slot.
- The duration of a send process corresponds to the length of a time slot.

a)* What is the probability that a collision-free transmission takes place in a time slot?

Let X be the ZV, which specifies the number of stations transmitting simultaneously in the time slot in question. The transmission is collision-free if and only if $X = 1$, i. h. exactly one *any* user is transmitting. X is therefore binomially distributed with transmission probability p:

$$\Pr[X = k] = \binom{n}{k} p^k (1-p)^{n-k} \quad \Rightarrow \quad \Pr[X = 1] = \binom{n}{1} p (1-p)^{n-1} = np \cdot (1-p)^{n-1} =: f(n, p)$$

b) Determine p^* such that the probability of a collision-free transmission is maximized.

Derivation:

$$\frac{\partial f}{\partial p} = n \cdot (1-p)^{n-1} - np \cdot (n-1) \cdot (1-p)^{n-2} \overset{!}{=} 0$$

$$n \cdot (1-p)^{n-1} = np \cdot (n-1) \cdot (1-p)^{n-2}$$

$$1 - p = p \cdot (n-1)$$

$$p = \frac{1}{n}$$

[1] For large n (approx. $n > 15$) and small send probabilities, the Poisson distribution could also be used here

Sample Solution

c) Now determine the maximum channel utilization for *n* users.

$$f(n, p^*) = \left(1 - \frac{1}{n}\right)^{n-1}$$

d) Now determine the maximum channel utilization for a very large number of users.

Hint: $\lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n = e^x$

$$\lim_{n\to\infty} f(n, p^*) = \lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^{n-1} = \lim_{n\to\infty} \frac{\left(1 - \frac{1}{n}\right)^n}{\left(1 - \frac{1}{n}\right)} = \frac{1}{e} \approx 0.37$$

Sample Solution

Problem 2 ALOHA and CSMA/CD

Let there be a network (see figure 2.1) consisting of three computers which are connected to each other via a hub. The distances between the computers are approximately $d_{12} = 1\,\text{km}$ and $d_{23} = 500\,\text{m}$. Any indirect cable routing may be neglected. The transmission rate shall be $r = 100\,\text{Mbit/s}$. The relative propagation speed is $\nu = {}^{2}/_{3}$ as usual. The speed of light is given by $c_0 = 3 \cdot 10^8\,\text{m/s}$.

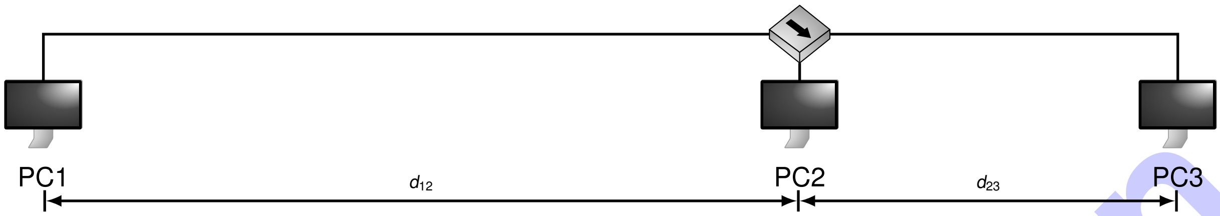


Figure 2.1

At time

- $t_0 = 0\,\text{s}$ no transmission takes place and none of the computers has data to send,
- $t_1 = 5\,\mu\text{s}$ PC1 begins to send,
- $t_2 = 15\,\mu\text{s}$ PC2 begins to send and
- $t_3 = 10\,\mu\text{s}$ PC3 begins to send

to send a frame of length 94 B each.

Sample Solution

a)* Calculate the serialisation time t_s for a message.

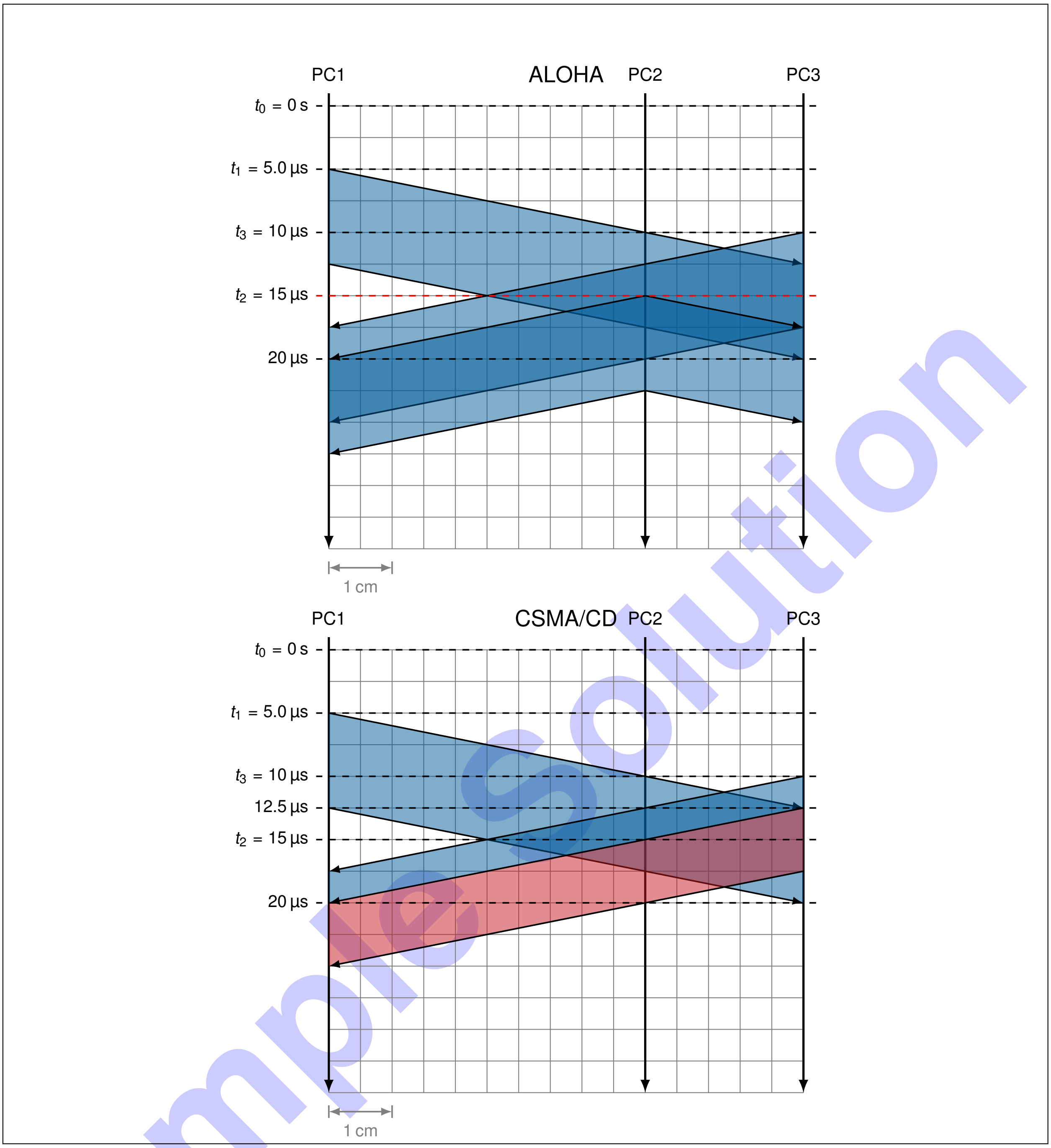
$$t_s = \frac{l}{r} = \frac{94 \cdot 8\,\text{bit}}{100 \cdot 10^6\,\text{bit/s}} = 7.52\,\mu\text{s}$$

b)* Calculate the propagation delays $t_p(1,2)$ and $t_p(2,3)$ on the two sections.

$$t_p(1,2) = \frac{d_{12}}{\nu c_0} = \frac{1000\,\text{m}}{\frac{2}{3} \cdot 3 \cdot 10^8\,\text{m/s}} = 5.0\,\mu\text{s}$$

$$t_p(2,3) = \frac{d_{23}}{\nu c_0} = \frac{500\,\text{m}}{\frac{2}{3} \cdot 3 \cdot 10^8\,\text{m/s}} = 2.5\,\mu\text{s}$$

c) For ALOHA and 1-persistent CSMA/CD respectively, draw a path-time diagram representing the transmission process in the time interval $t \in [t_0, t_0 + 30\,\mu\text{s})$. Scale: $100\,\text{m} \triangleq 5\,\text{mm}$ and $2.5\,\mu\text{s} \triangleq 5\,\text{mm}$, slot time: $\approx 5\,\mu\text{s}$



Sample Solution

Explanation: With ALOHA, the medium is not monitored. This means that at time t_2 PC2 starts transmitting, although it could already detect the transmission of PC1 and PC3. In contrast, with CSMA/CD the medium is monitored. For this reason, PC2 does not start transmitting. PC3, however, cannot yet know that PC1 is already transmitting due to the finite signal propagation speed. Therefore, a collision occurs.

At t = 12.5 µs PC3 detects the collision and aborts its own transmission. To ensure that all stations connected to the shared medium are informed of the collision, PC3 sends a *JAM signal*. This would be a 4 B long alternating bit pattern for Ethernet (the task so far is not specifically about Ethernet, but only about the underlying media access method CSMA/CD – a hint of the JAM signal is sufficient).

d) From the previous subtask it can be seen that collisions occur with both methods. In contrast to ALOHA, however, CSMA/CD does not work under the given circumstances. Why?

With ALOHA, the loss of a frame is recognised by the fact that the sender does not receive an acknowledgement. Such an acknowledgement procedure does not exist with CSMA/CD. Instead, with CSMA/CD, a sender assumes that a frame was successfully transmitted if no collision occurred during transmission.

In this case, however, PC1 has completed the transmission before the transmission or JAM signal from PC3 reaches it.PC1 therefore does not recognise the collision and wrongly assumes a successful transmission.

e) What is the condition for CSMA/CD that a node can detect a collision in time?

The serialisation time must be at least twice as long as the maximum possible propagation delay between the two nodes furthest apart.This is the only way to ensure that a node is still transmitting when it receives the „interference signal “ from the node furthest away from it, which itself started transmitting immediately before the „arrival of the first bit “.

f) For CSMA/CD, calculate the maximum distance between two computers within a collision domain as a function of the minimum frame length. Insert the values for FastEthernet ($r = 100\,\text{Mbit/s}$, $l_{\text{min}} = 64\,\text{B}$).

In the event of a collision, none of the sending nodes may end its transmission process before it has noticed the collision. Otherwise, it would assume that the transmission was successful. This means the minimum serialisation time $t_{s,\text{min}}$ of a frame must be twice the propagation delay between the two farthest stations:

$$\begin{aligned}
t_{s,\text{min}} &= 2 \cdot t_{p,\text{max}} \\
\frac{l_{\text{min}}}{r} &= 2 \cdot \frac{d_{\text{max}}}{\nu c} \\
d_{\text{max}} &= \frac{1}{2} \cdot \nu c \cdot \frac{l_{\text{min}}}{r} \\
d_{\text{max}} &= \frac{1}{2} \cdot \frac{2}{3} \cdot 3 \cdot 10^8 \frac{\text{m}}{\text{s}} \cdot \frac{64 \cdot 8\,\text{bit}}{100 \cdot 10^6\,\frac{\text{bit}}{\text{s}}} \\
d_{\text{max}} &= 10^8 \frac{\text{m}}{\text{s}} \cdot \frac{64 \cdot 8\,\text{bit}}{100 \cdot 10^6\,\frac{\text{bit}}{\text{s}}} = 512\,\text{m}
\end{aligned}$$

Sample Solution

Problem 3 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
```

Sample Solution

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.

Sample Solution

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[2] 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.

Sample Solution

[2] Jumbo frames are not considered in CNS