Technische Universität München
Lehrstuhl Informatik VIII
Prof. Dr.-Ing. Georg Carle
Christian Grothoff, Ph.D.
Stephan M. Günther, M.Sc.

Master Course Computer Networks

Homework 2

(submission until November 19th into INBOX located in front of 03.05.052)

Note: Subproblems marked by * can be solved without preceding results.

Media Access Control (CSMA CD and the Binary Exponential Backoff)

Given a network with n stations attached to hub (see Figure 1). Let's assume time is slotted and each slot lasts for 512 bit, the transmit rate is 100 Mbit/s, and CSMA/CD is being used as MAC protocol, i. e., any node that currently has data to be sent will start sending at the next time slot it detects an idle medium.

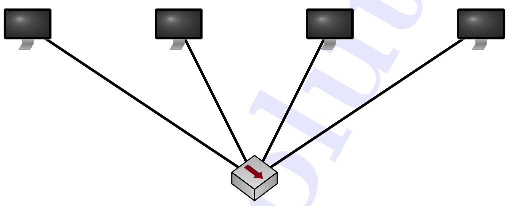


Figure 1: Sample topology

In essence, the situation described above ressembles a simplified variant of FastEthernet.

a)* What is the reasoning that FastEthernet requires a minimum frame length of 64 B?
Hint: Think about serialization delays and signal propagation.

Ethernet specifies a maximum distance of 500 m between any two nodes within a single broadcast domain. For CSMA/CD to work, a node must be able to detect a collision while still transmitting (a frame is assumed to be transmitted successfully as soon as transmission is complete). Consequently, a collision must be detected while still transmitting.
The worst case is that some station A starts transmitting and immediately before the first bit arrives at the distant station B, this station also starts transmitting. The condition is that A must still be transmitting when the first bit of station B arrives at A. Thus, we have two times the signal propagation delay between A and B.
Given the approximate speed of light in copper ($2 \cdot 10^8$ m/s), the above mentioned maximum distance, and the transmit rate of FastEthernet, we obtain

$$l_{\min} = \frac{2 \cdot 500\,\mathrm{m}}{2 \cdot 10^8\,\mathrm{m/s}} \cdot 100\,\mathrm{Mbit/s} = 62.5\,\mathrm{Byte}.$$

Taking some guard intervals into account, the minimum frame length of 64 Byte appears reasonable.

b) Why is it impossible that one node starts transmitting at time slot t and another node still detects an idle medium at $t+1$?

(Fast)Ethernet is slotted and the slot time corresponds to the time needed to transmit 512 bit, or, equivalently 64 Byte. As discussed in the previous subproblem it takes at most half this time for the signal to propagate to the farthest station. Consequently, it is made sure that after one slot time all stations know about an ongoing transmission.

In case of collisions a binary exponential backoff is being used. After the k-th collision, each node draws uniformly distributed some $x \in \{0, 1, 2, \dots, 2^k - 1\}$ and waits for x slot times before attempting to transmit again. As a simplification, we assume that any node is notified about a collision and re-draws x even if it was not involved in this collision, i. e., it was waiting due to a previous collision. Furthermore, we assume that all nodes have non-zero backlog, i. e., every node wants to transmit in the next possible time slot.

c)* Let X be the random variable that counts successive retransmits, i. e., $\Pr[X = k]$ is the probability for a successful transmission ~~in the k-th time slot after $k-1$~~ after k successive collisions. Determine the mass function of X.

We start with a slightly modified random variable $Y \in \{0, 1\}$ representing the event of a successfull transmission ($Y = 1$) given a specific k. In particular, Y does not take the history into account but only considers the probability for a successful transmission given a specific value for k.

- $\Pr[Y = 1 \mid k = 0]$:
  For $k = 0$, x is drawn from $\mathcal{X} = \{0\}$, i. e., all stations involved draw the same value and thus start transmitting at the same time. As a result, we have

  $$\Pr[Y = 1 \mid k = 0] = \begin{cases} 0 & n > 1, \\ 1 & n = 1. \end{cases}$$

  For arbitrary $k \geq 0$ and $n = 1$ we obtain $\Pr[Y = 1 \mid k \geq 0, n = 1] = 1$. Thus, we limit the further discussion to the interesting case $n > 1$.

- $\Pr[Y = 1 \mid k = 1]$:
  Now, x is drawn from $\mathcal{X} = \{0, 1\}$. Given n stations drawing some value from $\mathcal{X}$, there are 2^n different results possible. Assume that one specific sation draws 0, then the remaining $n-1$ stations must draw 1 in order to make it work. Considering that the one station above can be any of the n stations, there is a total of n different results for which no collision occurs. We thus have

  $$\Pr[Y = 1 \mid k = 1, n > 1] = \frac{n}{2^n}.$$

- $\Pr[Y = 1 \mid k = 2]$:
  Now, x is drawn from $\mathcal{X} = \{0, 1, 2, 3\}$. Given n stations drawing some value from $\mathcal{X}$, there are $4^n = 2^{2n}$ different results possible. Assume that one specific sation draws 0, then the remaining $n-1$ stations must draw some value larger than 0 in order to make it work. In the same way, if one station draws 1, then the remaining $n-1$ stations must draw some values larger than 1 and so on. This gives

  $$\Pr[Y = 1 \mid k = 2, n > 1] = \frac{n}{2^{2n}} \left(3^{n-1} + 2^{n-1} + 1\right).$$

- $\Pr[Y = 1 \mid k]$:
  In the general case for some $k > 0$, we thus obtain

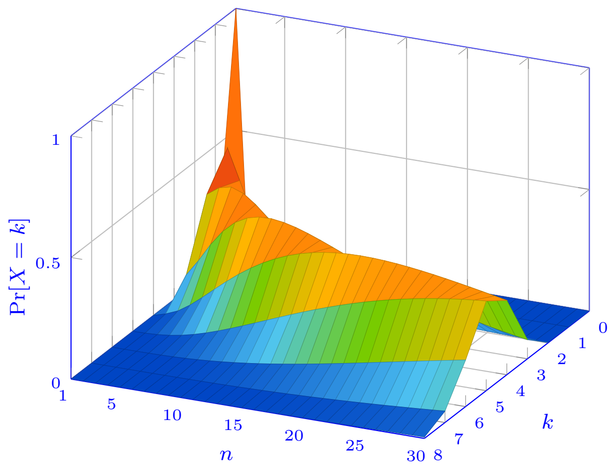
$$\Pr[Y = 1 \mid k] = \frac{n}{2^{nk}} \sum_{i=1}^{2^k-1} i^{n-1}.$$

  We are interested in $\Pr[X = k]$, i. e., the probability that transmission succeeds after k unsuccessful attempts. The difference to Y is that X takes previous unsuccessful attempts into account, e.g. the event $X = 1$ implicitly contains the result $Y = 0 \mid k = 0$. This finally gives

$$\begin{aligned}\Pr[X = k] &= \prod_{i=0}^{k-1} (1 - \Pr[Y = 1 \mid k = i]) \cdot \Pr[Y = 1 \mid k] \\ &= \prod_{i=0}^{k-1} \left(1 - \frac{n}{2^{ni}} \sum_{j=1}^{2^i-1} j^{n-1}\right) \cdot \frac{n}{2^{nk}} \sum_{i=1}^{2^k-1} i^{n-1}.\end{aligned}$$

  Looks weird and we should make sure that the result is a valid PMF. This can be done by summation over $k \geq 0$ (using a CAS) and making sure that $\Pr[X \geq 0]$ converges to 1.

d) Use a CAS[1] and visualize the PMF of X in dependency of n and k.



The plot nicely shows that the number of backoff slots (defined by k) significantly increases in n.

e) What is the expectation of X for $n = 3$ (and $k \to \infty$)?

For the expectation of X we obtain

$$\mathbb{E}[X] = \sum_{k=0}^{\infty} k \Pr[X = k] \approx 1.88.$$

Since the access procedure starts with $k = 0$ and fails for sure in this case, 3 nodes need on average 2.88 trials until one of them successfully transmits.

[1]TUM students of the faculties for Computer Science and Electrical Engineering have free access to MATLAB (see `https://matlab.rbg.tum.de`). It is also preinstalled on the "Sunhalle" accounts. But you may use any other CAS of your choice if you wish.

Classless Inter-Domain Routing **IPv4 Subnetting**

You are given the address blocks 131.159.20.0 / 22 and 131.159.36.0. / 24. Using these blocks, your task is to find a suitable subnet scheme fulfilling the requirements of Table 1.

Subnet	A	B	C	D	E
# IPs	300	300	15	40	4

Table 1: Number of required host addresses (including routers) for each subnet

a)* State the first and last address for each subnet. How many addresses per subnet can be used to address hosts?

The minimum size of the subnets (including network and broadcast addresses) for networks A – E are 512, 512, 32, 64, and 8, respectively. Thus the address blocks may be split into the following subnets:

- A: 131.159.20.0/23, 131.159.20.0 – 131.159.21.255
- B: 131.159.22.0/23, 131.159.22.0 – 131.159.23.255
- D: 131.159.36.0/26, 131.159.36.0 – 131.159.36.63
- C: 131.159.36.64/27, 131.159.36.64 – 131.159.36.95
- E: 131.159.36.80/29, 131.159.36.96 – 131.159.36.103

b)* Given the two address blocks, why is it not possible to create one single subnet containing all addresses?

The address blocks are non-continuous. The first one ends at ends at 131.159.23.255 while the second one start at 131.159.36.0. Any supernet comprising both subnets would also include the gap in between.

Note: Assuming the networks 131.159.20.0/22 and 131.159.24.0/24, it is also impossible to find a suitable supernet. The reason is that both networks together comprise the address space from 131.159.20.0 to 131.159.24.255, i. e. a total of five /24 subnets. The smallest supernet spanning this address range would, however, comprise a total of eight subnets and thus addresses not covered by the given subnets.

The reason for this is that subnet masks are always a block of n leading ones followed by $32 - n$ trailing zeros. Thus, the number of addresses within a subnet or the number of equally sized subnets within some supernet are always powers of two.