

Computernetze

Übungsblatt 3

Lösung Übungsblatt 3

1.)

a)

N Stationen teilen sich einen Pure-ALOHA-Kanal mit 56kbps.
Jede Station sendet durchschnittlich alle 100s einen
1000-Bit-Frame, auch wenn der vorherige noch nicht gesendet
werden konnte. Wie hoch ist der maximale Wert für N?

Antwort:

Wir erinnern uns: Die maximale Kanalauslastung beim Pure-ALOHA-Kanal waren ja $1/2e$, also ungefähr 0,184, sprich 18,4 %.

Wenn wir eine Bandbreite von 56 kbit/s haben, dann machen 18,4 % davon ...

$$\underline{56 \text{ kbit/s} * 0,184 = 10,304 \text{ kbit/s.}}$$

Die benutzbare Bandbreite ist also 10,304 kbit/s.

Die nächste Frage ist:

Welche Bandbreite brauchen wir für EINE Station (anschließend errechnen wir, wieviele Stationen dann „reinpassen“)?

Alle 100 Sekunden wird ein 1.000-Bit-Frame gesendet. Rechnen wir das mal runter auf 1 Sekunde...

Dreisatz:

$$\begin{array}{lcl} 1.000 \text{ Bit} & \leftrightarrow & 100 \text{ Sekunden} \\ 10 \text{ Bit} & \leftrightarrow & 1 \text{ Sekunde} \end{array}$$

Für EINE Station brauchen wir also eine Bandbreite von 10 Bit/s.

Wie viele Stationen passen also rein, wenn wir eine verfügbare Bandbreite von 10,304 kBit/s, also von 10.304 Bit haben?

Dazu teilen wir einfach die verfügbare Bandbreite durch die benötigte Bandbreite für EINE Station...

$$10.304 \text{ Bit / s} / 10 \text{ Bit / s} = 1030,4 \Rightarrow \underline{1030 \text{ Stationen}}$$

Antwort:

Der maximale Wert für N ist 1030!

1.) b)

Vergleichen Sie das Delay von Pure ALOHA und Slotted ALOHA sowohl bei niedriger als auch bei hoher Last. Welches ist geringer? Erklären Sie Ihre Antwort.

niedrige Last

Pure ALOHA hat niedriges Delay

Grund: bei Slotted ALOHA muss zunächst auf den Beginn des nächsten Slots gewartet werden.

hohe Last

Slotted ALOHA hat niedrigeres Delay

Grund: Slotted ALOHA führt zu weniger Kollisionen und damit zu einer geringeren durchschnittlichen Verzögerung.

1.)

c)

10.000 Flugreservierungssysteme konkurrieren um einen einzelnen Slotted-ALOHA-Kanal. Eine durchschnittliche Station macht pro Stunde 18 Reservierungsanfragen. Ein Slot dauert 125 μ s. Wie hoch ist die ungefähre Kanalauslastung?

Schritt 1:

Wie viele Reservierungsanfragen finden pro Stunde eigentlich statt?
Dazu benutzen wir den Dreisatz...

1 Station	↔	18 Reservierungsanfragen
10.000 Stationen	↔	180.000 Reservierungsanfragen

Wir haben also 180.000 Reservierungsanfragen pro Stunde.
Wieviel sind das dann pro Sekunde?

Dreisatz:

3.600 Sekunden (1h)	↔	180.000 Reservierungsanfragen
1 Sekunde	↔	50 Reservierungsanfragen

Wir haben also pro Sekunde 50 Reservierungsanfragen.

Kommen wir nun zu den Slots:

Ein Slot dauert 125 Mikrosekunden.

Frage: Wie viele Slots haben wir dann in einer Sekunde?

Dreisatz:

125 Mikrosekunden	↔	1 Slot
1 Sekunde	↔	8000 Slots

Nun haben wir also das Verhältnis 50 Anfragen / 8.000 Slots.
Und genau das ist die Angabe über die Kanalauslastung.

Also:

Kanalauslastung:

$50/8.000 = 0,00625 = \underline{0,625\%}$.

1.)
d)

Eine große Anzahl von ALOHA-Benutzern erzeugt pro Sekunde 50 Anfragen (inkl. Originale + Retransmissions). Die Slots sind 40 ms lang.

1d) i)

Wie hoch ist die Erfolgswahrscheinlichkeit beim ersten Versuch?

Wir haben die Angabe, dass pro Sekunde 50 Anfragen vorhanden sind. Wollen wir doch mal gucken, wie viele Anfragen auf einen Slot kommen. Dazu rechnen wir mal hoch, wie viele Slots es in einer Sekunde gibt...

Dreisatz:

40 ms	↔	1 Slot
1 s	↔	25 Slots

Wir haben also 50 Anfragen auf 25 Slots. Man braucht jetzt keine höhere Mathematik, um zu erkennen, dass das also 2 Anfragen pro Slot macht.

Die Erfolgswahrscheinlichkeit beim ersten Versuch ist also $p = 0,5$.

1d) ii)

Wie hoch ist die Wahrscheinlichkeit, nach exakt k Kollisionen erfolgreich zu sein?

Antwort:

Dazu haben wir folgende Formel (Tanenbaum):

$$P_k = e^{-G}(1 - e^{-G})^{k-1}$$

P_k ist dabei die gefragte Wahrscheinlichkeit, allerdings gibt k die

ANZAHL DER VERSUCHE

an, die bis zum Erfolg nötig sind.

Da jetzt nach k Kollisionen gefragt ist, müssen wir folgendes bedenken:

Der 2. Versuch kann erst nach der 1. Kollision stattfinden.

Das heißt, wenn hier nach k KOLLISIONEN gefragt wird, dann müssen wir das k aus der Formel auf "k+1" erhöhen, denn um nach k Kollisionen erfolgreich zu sein, brauchen wir k+1 Versuche.

Also:

$$P'_k = e^{-G}(1 - e^{-G})^k$$

Weiter können wir nicht rechnen, denn die Frage ist ja nicht konkret, sondern nur allgemein gestellt...

1d)
iii)

Wie hoch ist die erwartete Anzahl von Sendeversuchen, die zum Erfolg nötig sind?

Antwort:

Dazu haben wir eine recht komplizierte Formel (Tanenbaum):

$$\sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}(1 - e^{-G})^{k-1} = e^G$$

Die müssen wir, ausnahmsweise, einfach mal so hinnehmen - sei's drum...

1e)

Messungen eines Slotted-ALOHA-Kanals mit vielen Nutzern zeigen, dass 10 Prozent der Slots unbenutzt sind.

1e)

(i) Bestimmen Sie die Kanalauslastung G .

$$P_0 = e^{-G} = 0,1$$

$$G = -\ln(0,1) = 2,3$$

1e)

ii) Wie hoch ist der Durchsatz?

$$S = Ge^{-G} = 2,3e^{-2,3} = 0,23$$

1e)

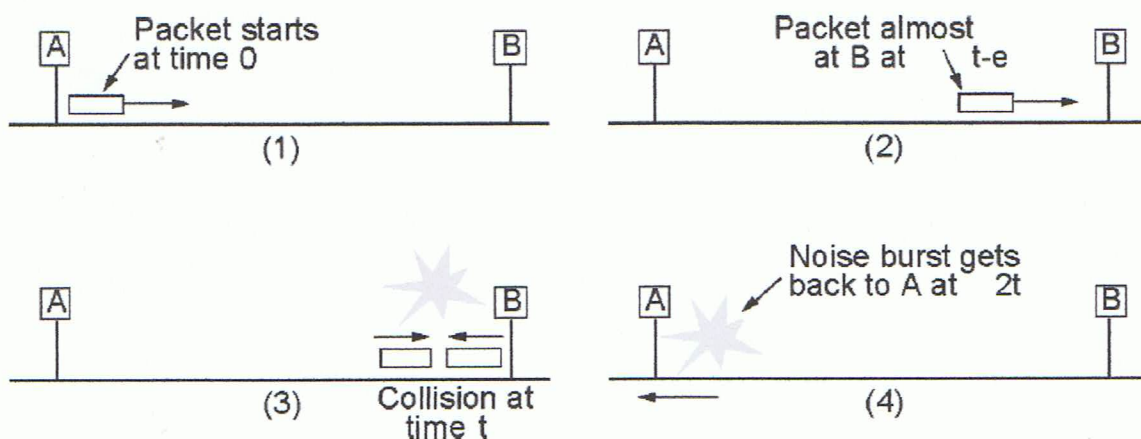
iii) Ist der Kanal über- oder unterlastet?

Da $G > 1$ ist ist der Kanal überlastet!

Minimale Paketlänge in CSMA/CD-Netzen

Gegeben sei ein CSMA/CD-Netz mit einer Datenrate von 10MBit/s. Zwei Stationen sind maximal 2,5km voneinander entfernt lokalisiert. Die Ausbreitungsgeschwindigkeit auf dem Medium sei $2 \cdot 10^8$ m/s.

2a) Warum existiert eine minimale Paketlänge?



Antwort: Station A hört während des Sendens mit, ob eine Kollision statt findet. Nehmen wir an, die Zeit, ein Paket von A zu B zu verschicken sei t und dies seien die beiden am weitesten entfernten Stationen.

Und nun nehmen wir an, Station B sendet ihr Paket ganz kurz, bevor das von Station A bei B ankommt (bis dato denkt B ja, es werde nichts gesendet). Kurz nachdem B das Paket losgeschickt hat, merkt B, dass es zu einer Kollision kommt und rücht sofort ab. Das „JAM“-Signal, also das Kollisions-Signal, wird nun auch an A gesendet. Und das dauert ja jetzt auch nochmal die ganze Zeit zurück, also nochmal fast die Zeit t .

Wenn die Station A das gesendete Paket also für die Dauer von $2t$ losschickt und die ganze Zeit kein JAM-Signal erhält, dann kann A sicher sein, dass es zu KEINER KOLLISION gekommen ist. Jetzt müssen wir noch noch wissen, wie hoch die Bitrate des Kanals ist und dann schauen, wieviele Bits in der Zeit von $2t$ gesendet werden, das ist dann die minimale Paketlänge.

2b) Wie groß ist die minimale Paketlänge in der angegebenen Konfiguration (s. Aufgabenstellung)?

(Hinweis: In diesem Beispiel entspricht die minimale Paketlänge nicht derjenigen aus dem IEEE 802.3 Standard)

Schauen wir doch erstmal, wie lange t in diesem Fall wäre, also die Zeit, ein Paket von A nach B zu schicken.

$$2,5\text{km} / 2 \cdot 10^8 \text{ m/s} = 12,5 \text{ Mikrosekunden}$$

Es hieß ja, dass wir, um eine Kollision erkennen zu können, die Dauer von $2t$ (also hin- und Rückweg) benötigen, also:

$$2 \cdot t = 2 \cdot 12,5 \text{ Mikrosekunden} = 25 \text{ Mikrosekunden}$$

Jetzt ist die nächste Frage:

Wieviele Bits werden in 25 Mikrosekunden gesendet?

$$10\text{Mbit/s} \cdot 25 \text{ Mikrosekunden} = 10 \cdot 10^6 \text{ Bit/s} \cdot 25 \cdot 10^{-6} \text{ s} = 10 \cdot 25 = \underline{250 \text{ Bit.}}$$

Die minimale Rahmenlänge beträgt also 250 Bit. Wenn wir das in Byte umrechnen wollten wären das:

$$250 / 8 = 31,25 \text{ Byte also ungefähr } 32 \text{ Byte.}$$

Wem eine Formel lieber ist:

$$\text{minimale Paketlänge} = \frac{2 \cdot l \cdot r_B}{v}$$

r_B : Datenrate
 l : max. Distanz zw. je 2 Stationen
 v : Ausbreitungsgeschwindigkeit

2c) Wie groß ist die minimale Paketlänge, wenn man die Datenrate auf 100MBit/s erhöht? Laut der Formel sehen wir, dass dann auch die minimale Paketlänge zehn mal so hoch sein muss...

2c) Welcher Nachteil ergibt sich dadurch und wie kann man diesen vermeiden?

Der Nachteil ist eine Verschwendung von Bandbreite bei kleinen Paketen (Padding)

Alternativ zu einer größeren minimalen Paketlänge kann auch die maximale Entfernung zwischen zwei Stationen verringert werden (was auch in der Realität bei 100 Mbit/s Ethernet gemacht wird, dort verringert sich dieser Abstand auf 250 m).

Aufgabe 3:

Vergleich Ethernet und Token-Ring

Zur Untersuchung von lokalen Netzen sollen die Typen Ethernet und Token-Ring betrachtet werden.

3a)

Worin liegt der grundlegende Unterschied der beiden beim Anschluss an das Medium?

Token-Ring:

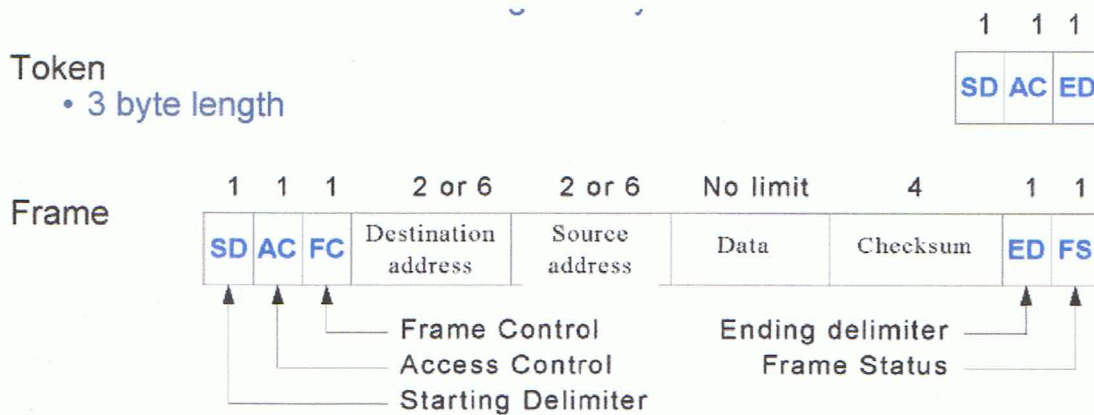
- Aktiver Anschluss an das Netz
- Regenerierung des Signals in jedem Ringadapter
- 1-Bit-Verzögerung
- Bei Ausfall eines Ringadapters: Ring unterbrochen (wenn keine zusätzliche technische Abhilfe vorhanden, z. B. durch Relais oder zentrales Verkabelungszentrum)

CSMA/CD:

- Passiver Anschluss an das Netz
- Jede Station hört alles, kann keine Daten verändern oder stoppen

3b)

Wie kann die Quittierung einer Nachricht in beiden Typen erfolgen?



Zu erst Token Ring:

Ein Rahmen des Token Ring sieht wie oben abgebildet aus.

Als vorletztes Byte haben wir den Enddelimiter. Dieser gibt das Ende des Rahmens an. Wie wir sehen, kommt anschließend 1 Byte für den Rahmenstatus (= Framestatus, FS).

Dieses Byte ist wie folgt aufgesplittet:

Frame Status (FS)

A	C	r	r	A	C	r	r
---	---	---	---	---	---	---	---

A = address-recognized bits
C = frame-copied bits
r = reserved bits

Wichtig sind hierbei die A- und C-Bits.

A=0, C=0:

Adressat ist nicht vorhanden oder nicht eingeschaltet

A=1, C=0:

Adressat ist vorhanden, aber Rahmen nicht angenommen

A=1, C=1:

Eigentliche Bestätigung: Adressat vorhanden und Rahmen wurde kopiert

Kommen wir nun zu CSMA/CD:

Bei CSMA/CD ist es so, dass der Sender wissen darf, dass sein Rahmen angekommen ist, wenn in der Zeit von $2t$ keine Kollisionsmeldung gekommen ist. Zusätzlich gibt es dann noch die Möglichkeit, den Slot danach extra für eine explizite Bestätigung freizuhalten.

3c) Welche der beiden Typen ist prinzipiell für den Realzeitbetrieb geeignet?

Besser geeignet ist der Token Ring.

Warum?

Beim CSMA/CD können Kollisionen auftreten und je nachdem, wie viel Last los ist, können mehrere Kollisionen nacheinander auftreten und da Wartezeiten zum Algorithmus gehören, die zufällig gewählt werden, kann man nie voraus sagen, wann die Daten tatsächlich spätestens beim Empfänger ankommen (wird in der nächsten Aufgabe einmal durchgerechnet).

Beim Token Ring gibt es keine Kollisionen - das macht ihn Echtzeittauglicher als CSMA/CD.

3d)

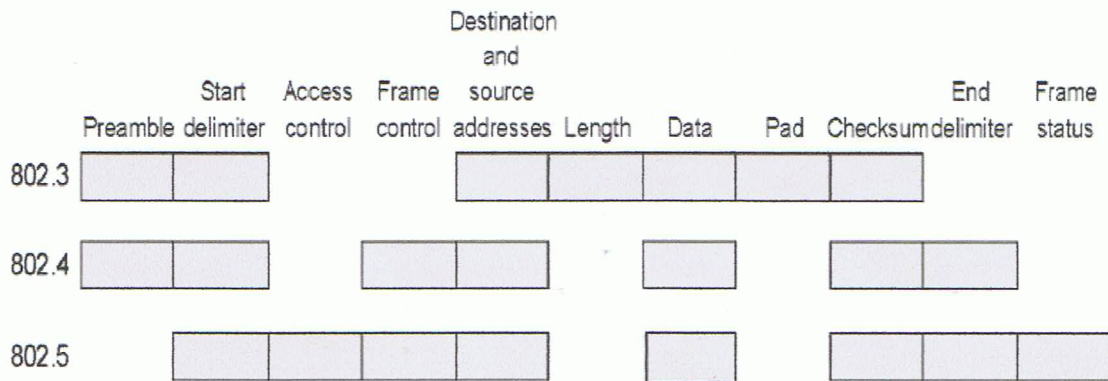
Wie wird das Ende bzw. die Länge der Rahmen in beiden Netzen erkannt (Begründung)?

Token-Ring (802.5):

End-Delimiter kennzeichnet Ende. Der End-Delimiter enthält Symbole, die in der Codierung (Differential Manchester) nicht erlaubt sind (kein Wechsel in der Mitte des Taktes).

CSMA/CD (802.3): Längengeld gibt die Länge des Datenfeldes an.

Ende des Frames: Länge plus 4 Bytes für CRC; evtl. Padding berücksichtigen.



4. Aufgabe:

Token Ring

4) Ein 16 Mbit/s-Token-Ring habe eine Token-Holding-Time von 10 ms. Der Ring verbinde 10 Stationen mit einem Abstand von jeweils 100 Metern. Die Signalausbreitungsgeschwindigkeit betrage $5 \cdot 10^7$ m/s.

4a) Nach welcher Zeit erhält eine Station spätestens das Token, wenn alle Stationen ihre Sendezeit voll ausnutzen?

Antwort:

Wir haben 10 Stationen.

Nehmen wir an, eine Station gibt ihr Token gerade ab. Wie lange dauert es also, bis sie das Token wieder bekommt?

Wenn es 10 Stationen gibt, dann haben wir also neben der einen, um die es jetzt geht, noch 9 andere. Wenn alle ihre "Sendezeit" voll ausnutzen, dann haben wir also $n-1$ mal, also in diesem Fall 9 mal die Token Holding Time (THT). Die Token Holding Time beträgt ja 10ms.

Neben 9 mal der THT haben wir ja noch die Zeit, die gebraucht wird, damit das Token einmal komplett die Runde gedreht hat, also 10 mal der Weg von Station zu Station, also n mal t_s , wenn wir sagen, dass t_s die Zeit von Station zu Station ist. Jetzt dürfen wir nicht vergessen, dass wir bei jeder Station außerdem eine 1-Bit-Verzögerung haben, also 10 mal 1-Bit-Verzögerung.

Wie groß sind jetzt die einzelnen Zeiten?

THT = 10ms.

Wie lang dauert t_s ?

Entfernung: 100 ms.

Rechnung:

$$100\text{m} / 5 \cdot 10^7 \text{ m/s} = 2 \mu\text{s}$$

$t_s \Rightarrow 2 \mu\text{s}$

Wie lange dauert die 1-Bitverzögerung?

$$1 \text{ Bit} / 16\text{MBit/s} = 0,0625 \mu\text{s} \quad (= 62,5 \text{ Nanosekunden})$$

Ein kompletter Umlauf dauert also:

9 mal Token Holding Time + 10 mal der Weg von Station zu Station und deren 1-Bit-Verzögerung,
also:

Wartezeit $W = (n-1) \cdot THT + U$; wobei U der Umlauf ist und sich aus $n \cdot (t_s + t_1)$
berechnet und t_1 für die 1-Bit-Verzögerung steht)

$$\begin{aligned} &= 9 \cdot 10 \text{ ms} + 10 \cdot (2 \mu\text{s} + 0,0625 \mu\text{s}) \\ &= 90 \text{ms} + 20,625 \mu\text{s} \\ &= \underline{90,020625 \text{ms}} \end{aligned}$$

Rechnung laut Lösung:

$$\begin{aligned} W &= (n - 1) \cdot \left(\frac{P_{max}}{K} + U \right) + U \\ &= (n - 1) \cdot THT + (n \cdot t_s + n \cdot t_1) \\ &= 9 \cdot 10 \text{ms} + 10 \cdot \frac{100 \text{m}}{5 \cdot 10^7 \frac{\text{m}}{\text{s}}} + 10 \cdot \frac{1 \text{bit}}{16 \frac{\text{Mbit}}{\text{s}}} = 90,020625 \text{ms} \end{aligned}$$

mit

K : Kapazität

n : Anzahl der Stationen

U : "Ring Circulation Time"

t_s : Verzögerung zwischen zwei Stationen

t_1 : 1-Bit Verzögerung

P_{max} : max. Paketlänge

4b) Nach welcher Zeit erhält eine Station spätestens das Token, wenn keine Station sendet?

$$W_{min} = U = (n \cdot t_s + n \cdot t_1) = 20,625 \mu\text{s}$$

Der Ring habe nun einen Durchsatz von 100 Mbit/s und es sollen 200 Stationen angeschlossen werden.

4c) Welchen Wert muss die Token-Holding-Time mindestens haben, damit die maximale Paketlänge gleich bleibt?

Die erste Frage wäre natürlich:

Wie groß ist denn eigentlich die maximale Paketlänge bisher gewesen?

Das kriegen wir durch die THT raus.

Die THT berechnet sich nämlich aus der maximalen Paketlänge, und zwar wie folgt:

$$THT = \frac{P_{max}}{K} + U \Rightarrow P_{max} = (THT - U) \cdot K$$

$$U = (n \cdot t_s + n \cdot t_1) = 20,625 \mu\text{s}$$

$$P_{max} = (10 \text{ms} - 20,625 \mu\text{s}) \cdot 16 \frac{\text{Mbit}}{\text{s}} = 159670 \text{bit}$$

Daraus lässt sich, wie oben zu sehen, die maximale Paketlänge durch Umrechnung errechnen...
Und die liegt, wie wir sehen, bei 159.670 Bit.

So, diese Länge soll sich nicht ändern, aber andere Parameter haben sich geändert, nämlich:

Durchsatz (= K) von 100 Mbit/s und es sollen
n = 200 Stationen angeschlossen werden.

Dazu folgende Berechnung:

$$P = (THT - U) \cdot K$$

Umlauf bei 200 Stationen:

$$U = 200 \times (t_s + t_1) = 200 \times \left(\frac{100m}{5 \cdot 10^7 m/s} + \frac{1 \text{ Bit}}{100 \text{ Mbit/s}} \right)$$
$$= 200 \times (2 \mu s + 0,01 \mu s) = 200 \times 2,01 \mu s = \underline{\underline{402 \mu s}}$$

Also:

$$P = (THT - U) \cdot K \quad | : K$$

$$\frac{P}{K} = THT - U \quad | + U$$

$$THT = \frac{P}{K} + U \quad \Rightarrow \text{geg. } \begin{cases} P = 159670 \text{ Bit} \\ K = 100 \text{ Mbit/s} \\ U = 402 \mu s \end{cases}$$

$$THT = \frac{159670 \text{ Bit}}{100 \text{ Mbit/s}} + 402 \mu s$$

$$= 1,5967 \text{ ms} + 402 \mu s$$

$$= 1,5967 \text{ ms} + 0,402 \text{ ms}$$

$$= \underline{\underline{1,9987 \text{ ms}}}$$

4d)

Wie lange darf das Token höchstens gehalten werden, wenn die maximale Wartezeit auf das Token gleich bleiben soll?

Dazu folgende Berechnung:

Maximale Wartezeit war $90,020625 \text{ ms}$

Formel:

$$W = (n-1) \cdot THT + (n \cdot t_s + n \cdot t_1)$$

Folgende Werte sind gegeben:

$$n = 200$$

$$t_s = 2 \mu\text{s}$$

$$t_1 = 0,01 \mu\text{s}$$

Somit (nach THT auflösen):

$$W = (n-1) \cdot THT + n(t_s + t_1) \quad | - n(t_s + t_1)$$

$$W - n(t_s + t_1) = (n-1) \cdot THT \quad | : (n-1)$$

$$THT = \frac{W - n(t_s + t_1)}{n-1}$$

mit konkreten Werten:

$$THT = \frac{90,020625 \text{ ms} - 0,402 \text{ ms}}{199} = 0,45 \text{ ms} = \underline{\underline{450 \mu\text{s}}}$$

5) Wie viele Frames pro Sekunde kann Gigabit-Ethernet verarbeiten? Betrachten Sie alle möglichen Fälle.

Datenrate: 10^9 bit/s, 4 Fälle:

- Traditionelles Halb-Duplex-Gigabit-Ethernet
- Halb-Duplex-Gigabit-Ethernet mit Carrier Extension
- Halb-Duplex-Gigabit-Ethernet mit Frame Bursting
- Voll-Duplex-Gigabit-Ethernet

Halb-Duplex-Gigabit-Ethernet

- Min. Rahmenlänge: 64 bytes = 512 bits \Rightarrow frame rate = $10^9/512 \approx 2 \cdot 10^6$ fps
- Max. Rahmenlänge: 1518 bytes = 12144 bits \Rightarrow frame rate = $10^9/12144 \approx 8,2 \cdot 10^4$ fps

Halb-Duplex-Gigabit-Ethernet mit Carrier Extension

- Min. Rahmenlänge: 512 bytes = 4096 bits \Rightarrow frame rate = $10^9/4096 \approx 2,46 \cdot 10^5$ fps
- Max. Rahmenlänge: 1518 bytes = 12144 bits \Rightarrow frame rate = $10^9/12144 \approx 8,2 \cdot 10^3$ fps

Halb-Duplex-Gigabit-Ethernet mit Frame Bursting

- Min. Rahmenlänge: 512 bytes = 4096 bits \Rightarrow frame rate = $10^9/4096 \approx 2,46 \cdot 10^5$ fps
- Max. Rahmenlänge: 8192 bytes = 65536 bits \Rightarrow frame rate = $10^9/65536 \approx 1,52 \cdot 10^4$ fps

Voll-Duplex-Gigabit-Ethernet

- Keine minimale und maximale Rahmenlänge
- Rahmenlänge hängt von der Signalstärke ab
- Studien haben gezeigt, dass CRC-32 nicht mit Rahmen >9000 bytes umgehen kann
- 9000 bytes = 72000 bits \Rightarrow frame rate = $10^9/72000 \approx 1,4 \cdot 10^4$ fps