MAC-Teilschicht

“PURE ALOHA“

Die MAC-Teilschicht ist, wie ihr Name bereits verrät eine "Teilschicht", das heißt, eine Unterschicht zur Sicherungsschicht.

Wofür steht MAC?

M : Medium
A : Access
C : Control

= Medium-Zugriffssteuerung

Es soll also der Zugriff auf ein Medium gesteuert werden.
Wenn Daten ausgetauscht werden, dann geschieht das ~~z~~ über einen Kanal. Also ist der Kanal unser Medium.
Wir befinden uns bei der MAC-Teilschicht im Broadcast-Netz, also im Rundfunk.
Im Rundfunk ist es so, dass auf einen Kanal mehrere Stationen zugreifen können.
Das heißt, der mehrfache Zugriff von verschiedenen Stationen auf einen Kanal muss gesteuert werden. Deswegen werden die Protokolle, die hier behandelt werden, auch Mehrfachzugriffsprotokolle genannt.

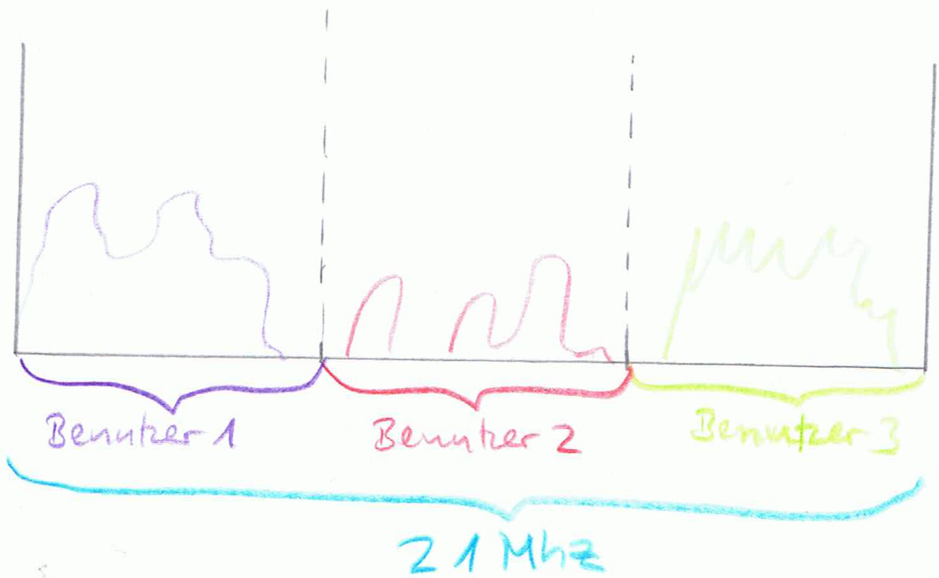
Hier ist also die Hauptfrage:

Wie kann ein Kanal mehreren Benutzern gleichzeitig zur Verfügung gestellt werden?

Eine Variante, mit der man diese Frage beantworten und dieses Problem somit behandeln könnte, wäre, das sogenannte Frequenzmultiplex-Verfahren.

Die Verfügbare Bandbreite des Kanals wird bei N Stationen einfach in N gleich große Teilabschnitte geteilt.

Wenn wir beispielsweise eine Bandbreite von 21 Mhz haben und N = 3 Stationen, die darauf zugreifen möchten, dann bekommt jede Station (jeder Benutzer) ein Drittel der Gesamtbandbreite, also jeder 7 Mhz.



Durch diese Aufteilung können die N (in diesem Fall eben 3) Kanäle jederzeit ungestört auf den Kanal zugreifen und ihre Daten übertragen.

Nun ist es aber so wie im realen Leben: Nicht immer hat jeder was zu sagen (hier: zu übertragen).

Stellen wir uns jetzt mal vor, N ist nicht 3 sondern, um es mal dramatisch genug zu machen, 21. Somit bekäme jeder ~~k~~ Benutzer eine Bandbreite von 1 Mhz zugeteilt.
Das hieße ja, die einzelnen Benutzer können nicht mehr so viel auf einmal senden.

Stellen wir uns jetzt noch vor, von diesen 21 Benutzern wollen lediglich 3 etwas senden. Von der insgesamt verfügbaren Bandbreite würde dann gerade mal ein Siebtel gebraucht werden (die anderen Benutzer nutzen ihre Bandbreite nicht, da sie ja nichts senden, geben sie aber auch nicht für andere Benutzer ~~Kanäle~~ frei).
Um es auf den Punkt zu bringen: Das wäre Bandbreitenverschwendung!

Es wäre doch genial, wenn jede sendende Station (Benutzer und Station benutz ich jetzt mal synonym) die volle Bandbreite zur Verfügung hätte. Das geht natürlich nur, wenn wir uns vom Frequenzmultiplexverfahren verabschieden.
Jetzt haben wir natürlich das Problem:
Wenn mehrere Stationen etwas senden ~~dürfen~~ wollen - wie soll das denn gehen? Stellen wir uns vor, in einem Meeting sitzen 6 Leute am Tisch und verhandeln. Da ist natürlich auch die Frage: Wer ~~sagt wann~~ spricht wann? Wenn auch nur 2 Leute gleichzeitig sprechen, ginge die Kommunikation in die Hose. Und um dieses Problem wollen wir uns jetzt kümmern...

Halten wir noch mal fest:

1.) Wir haben mehrere Stationen

2.) Wir haben nur einen Kanal

3.) Wenn zwei Stationen (oder mehrere) gleichzeitig Daten sendet, gehen diese "Datenaustausche" in die Hose. Man spricht hierbei von "Kollision".

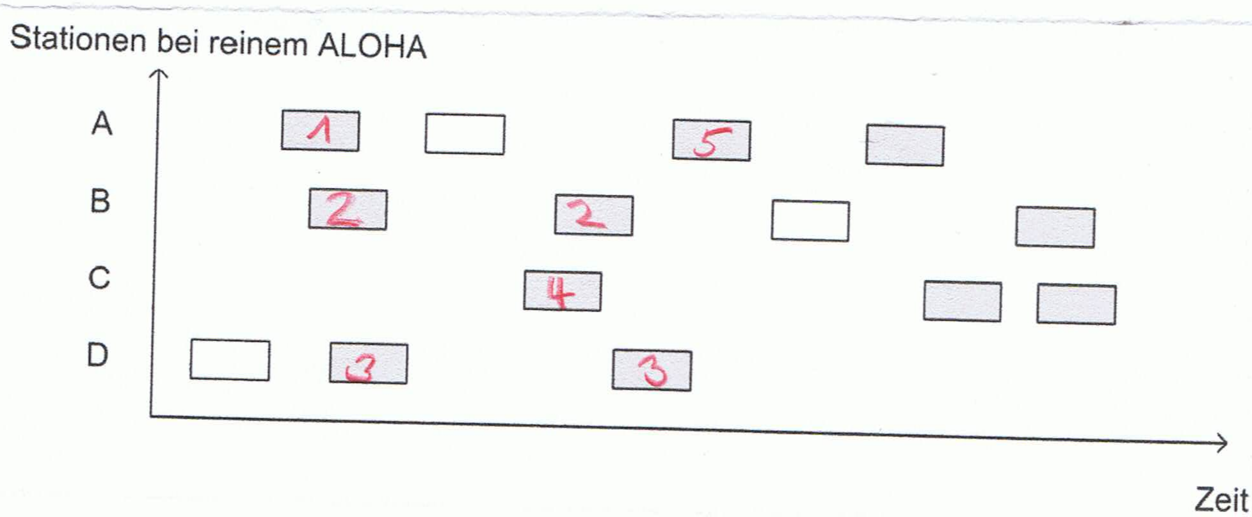
Kommen wir zum ersten "Mehrfachzugriffsprotokoll":

Das erste Protokoll wurde auf Hawaii erfunden und trägt den passenden Namen: ALOHA:
Davon gibt es zwei Versionen:
1.) Reines ALOHA (pure ALOHA)
2.) ~~S~~Unterteiltes ALOHA (slotted ALOHA)

Reines ALOHA:

Das Prinzip ist einfach: Wann immer eine Station gerade senden möchte: sie sendet einfach.
Das wäre vergleichbar mit dem Meeting-Beispiel von vorhin: Wann immer jemand etwas sagen möchte: er sagt es einfach.
Logisch, dass es, je mehr Leute am Tisch sitzen, zu Überschneidungen kommt. Zu unserer Übertragung zurück: Es kommt zu "Kollisionen". Die Wahrscheinlichkeit einer Kollision nimmt mit der Anzahl der Stationen natürlich zu.
Ob ein Rahmen, den eine Station sendet, angekommen und fehlerfrei übertragen worden ist, erfährt die sendende Station durch einen Acknowledgementrahmen.
Wenn der Rahmen kollidiert, dann wartet ~~der~~ die sendende Station eine zufällig gewählte Zeitspanne, bis sie die Übertragung neu startet.

Eine berechtigte Frage wäre natürlich:
Warum wartet die sendende Station eine zufällig gewählte Zeit?
Nehmen wir an, das Protokoll legt für einen kollidierenden Rahmen eine bestimmte Zeit fest. Nach Ablauf dieser Zeit soll der kollidierende Rahmen neu übertragen werden.
Klare Sache: Das würde dann ja stets für den anderen kollidierenden Rahmen auch gelten und nach dieser bestimmten Zeit würden exakt die beiden gleichen Rahmen erneut kollidieren.



Zum Bild:
Die grauen Rahmen sind "Kollisionsopfer", die weißen werden übertragen.
Damit wir wissen, wovon ich gleich spreche, habe ich die Rahmen, um die es gleich geht, beschriftet.
Zu erst einmal überträgt Station D erfolgreich einen Rahmen (gleich der erste, weiße Rahmen).
Als nächstes beginnt A mit dem Versenden des Rahmens (Rahmen 1). Doch dieser kollidiert mit Rahmen 2 und etwas später außerdem noch mit Rahmen 3.
Jetzt bekommen, wie vorhin erwähnt, alle drei Rahmen eine zufällig bestimmte Wartezeit zugeordnet.

Ein Glücksfall für Station A, denn dort ist die Zeit des Wartens am kürzesten und es gibt, wie wir sehen, während der Übertragungsdauer des Rahmens keine weitere Kollision.

Für Rahmen 2 und 3 sieht es dagegen nicht so gut aus (also für die Stationen B und D), denn diese kollidieren erneut, zusätzlich noch mit Rahmen 4, der von der Station C losgeschickt wurde.

Die längste Wartezeit, die zufällig ermittelt wurde, bekommt, wie wir sehen, Rahmen 3 (der nächste Übertragungsversuch ist auf dem Bild schon gar nicht mehr drauf).

Rahmen 4 hat auch eine relativ lange Wartezeit bekommen.

Rahmen 5 (von Station A) kollidiert zwischenzeitlich noch mit Rahmen 3 (von Station D), aber Rahmen 2 wird beim insgesamt nunmehr dritten Übertragungsversuch erfolgreich übertragen.

Wir sehen sofort, dass es mehr graue als weiße Rahmen, also mehr Kollisionen als erfolgreich übertragene Rahmen gibt.

Das lässt die Frage nach der Effizienz dieses Verfahrens aufkommen.

Dazu brauchen wir folgende Begriffe:

- Rahmenzeit
- Durchsatz (S)
- Versuche pro Paketzeit (G)

Beginnen wir mit der Rahmenzeit:

Rahmenzeit ist die Zeit, die benötigt wird, um den Standardrahmen mit fester Länge zu übermitteln (d.h. die Länge des Rahmens geteilt durch die Bitrate). Tanenbaum, 3. Aufl., S. 274

Durchsatz pro Rahmenzeit (=S)

Pro Rahmenzeit versuchen ja, wenn wir Pech haben, mehrere Stationen, Rahmen zu senden.
Dadurch kommt es ja zu Kollisionen.
Die Frage ist, wie viele Rahmen pro Rahmenzeit tatsächlich hergestellt werden können (ein kollidierender Rahmen wird ja nicht neu erstellt, sondern einfach erneut zu übertragen versucht).
Der Wert für S ist sinnvoller Weise einer zwischen 0 und 1 (womit wir fast eine Prozentangabe hätten).
Dass Ganze nochmal in anderen Worten:
Beim Durchsatz ist die Frage "Wieviel Rahmen werden durchschnittlich in der Rahmenzeit übertragen" entscheidend.

Also gilt: $0 < S < 1$

Versuche pro Paketzeit (=G)

Nehmen wir an, von 10 Station wollen 3 Stationen ~~ungefähr~~ zeitgleich einen Rahmen übertragen. Es käme zu einer Kollision und für alle 3 Rahmen würde jeweils ein neuer Versuch gestartet werden.
Je mehr Zeit vergeht, desto höher ist die Wahrscheinlichkeit, dass andere auch was übertragen wollen.
G gibt die durchschnittliche Anzahl an Versuchen pro Paketzeit an.

G ist stets mindestens so groß wie S. Logisch, denn jedem erfolgreich übertragenen Rahmen geht natürlich mindestens ein Versuch voraus.
Also gilt
$G \geq S$.

S war ja, wie man es auch ausdrücken kann, die Wahrscheinlichkeit, dass ein Rahmen erfolgreich übertragen wird.
Bei einem beliebigen Datenverkehr von G (wir können bei G auch von der „Belastung" des Kanals sprechen) lässt sich der Datendurchsatz S errechnen aus der gegebenen Belastung G multipliziert mit der Wahrscheinlichkeit, dass eine Übertragung erfolgreich ist.

Also:

$$\boxed{S = G \cdot P_0}$$

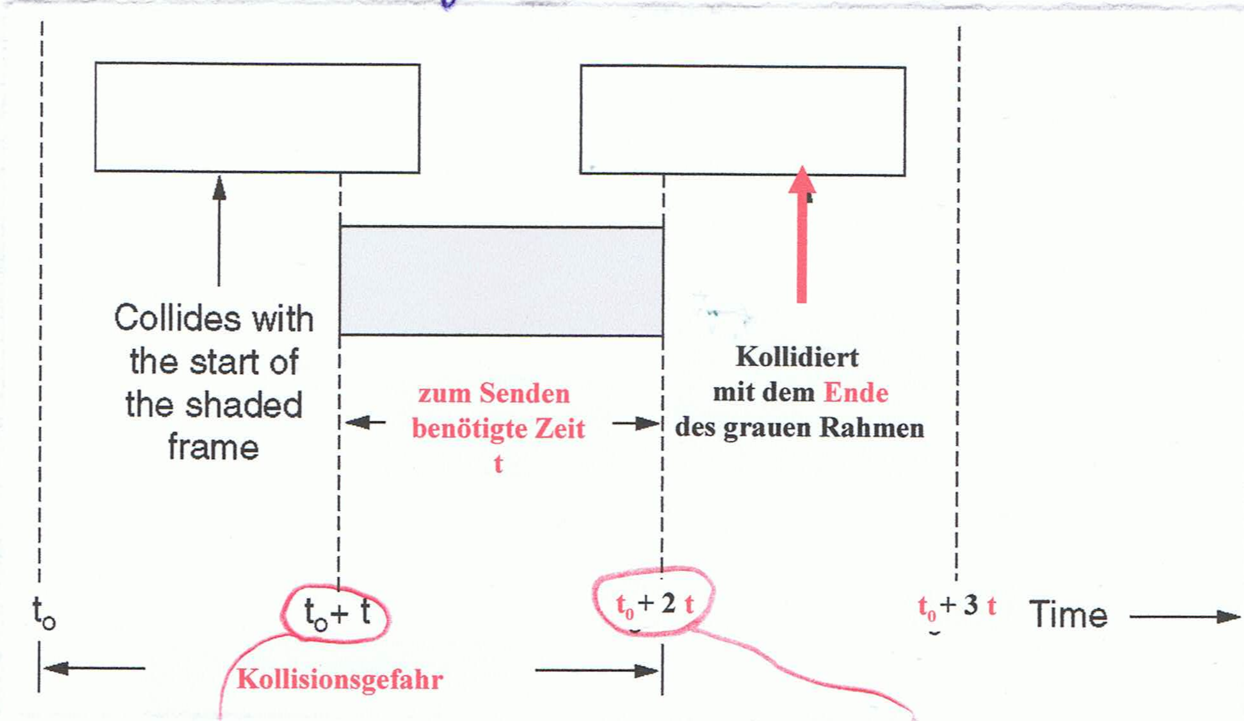
Wo bitteschön kommt jetzt P_0 her und was soll das sein? Wie angegeben, das ist die Wahrscheinlichkeit, dass ein Rahmen keine Kollision erleidet (also dass die Übertragung erfolgreich ist).

Diese Wahrscheinlichkeit ergibt sich aus der Poisson-Verteilung, die wir an dieser Stelle ausnahmsweise mal hinnehmen und nicht groß herleiten. Sie lautet:

$$\boxed{P(k) = \frac{G^k \cdot e^{-G}}{k!}}$$

Was ist jetzt schon wieder k?
k gibt die Anzahl der produzierten Rahmen pro Paketdauer t an (= Versuche, Rahmen zu übertragen).

Nehmen wir also an, die zum Senden eines Rahmens benötigte Zeit wäre t.



Der im Bild schattierte Rahmen wir zum Zeitpunkt t_0+t losgeschickt.
Wir sehen, dass dieser Rahmen mit dem linken oberen Rahmen (~~der linke weiße~~) sowie mit dem rechten oberen Rahmen (~~ebenfalls weiß~~) kollidiert.
Diese wurden im Intervall zwischen t_0 und t_0+t (linker Rahmen) und ~~dem~~ im Intervall zwischen t_0+t und t_0+2t losgeschickt.
Der schattierte Rahmen wäre problemlos übertragen worden, wenn zwischen t_0 und t_0+2t keine anderen ~~Rahmen übertragen worden~~ Übertragungsversuche passiert wären.
Der Versuch war ja k.
Wie hoch ist nun nach unserer Formel die Wahrscheinlichkeit, dass in der Paketdauer t kein Rahmen losgesendet wird?
Wir setzen dafür also für k einfach 0 ein...

$$P(k) = \frac{G^k \cdot e^{-G}}{k!}$$ allgemeine Formel

Konkret:

$$P(0) = \frac{G^0 \cdot e^{-G}}{0!} = \frac{1 \cdot e^{-G}}{1} = \underline{\underline{e^{-G}}}$$

Nun hatten wir aber gerade festgestellt, dass im Intervall von t_0 bis $t_0 + 2t$, also die doppelte ~~Paket~~ Rahmenlänge, kein Versuch gestartet werden darf.

Nach stochastischen Regeln müssen wir die Wahrscheinlichkeiten multiplizieren.

Kleiner Exkurs:
Die Wahrscheinlichkeit, eine 6 zu würfeln, ist $\frac{1}{6}$.
Die Wahrscheinlichkeit, zwei 6er in Folge zu würfeln,
Die Wahrscheinlichkeit, zwei 6er in Folge zu würfeln, ist

$$\frac{1}{6} * \frac{1}{6} = \frac{1}{36}.$$

Nach gleichem Prinzip gehen wir hier vor. Zwei mal in Folge sollen keine Rahmen gesendet werden, also multiplizieren wir die Wahrscheinlichkeiten:

$$\boxed{P(0) = e^{-G} \cdot e^{-G} = e^{-2G}}$$

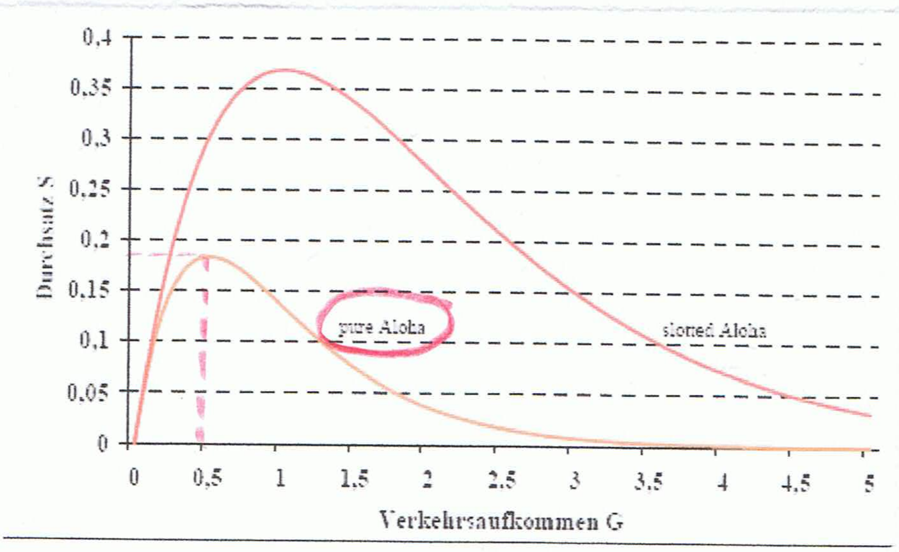
Nun hieß es vorhin, dass wir für den Durchsatz die gegebene Belastung des Kanals (also G) mit der Wahrscheinlichkeit multiplizieren, dass ein Rahmen übertragen wird, also

$$S = G \cdot P_0 \Rightarrow \boxed{S = G \cdot e^{-2G}}$$

Eine alternative Überlegung wäre, dass im Paket-Sende-Intervall nur ein Rahmen zu übertragen versucht wird und im Paketintervall davor kein Rahmen.

Das hieße dann:

$$S = P(k=1) * P(k=0) = \frac{G^1 \cdot e^{-G}}{1!} * e^{-G} = G \cdot e^{-G} \cdot e^{-G} = \underline{\underline{G \cdot e^{-2G}}}$$



Das Bild zeigt das Verhältnis zwischen der Belastung des Kanals (Verkehrsaufkommen G) und ~~S~~ dem Durchsatz S.

Wir sehen:

Der größte Durchsatz wird bei $G = 0{,}5$ erreicht ($S = \frac{1}{2e} \approx 0{,}184$).

Durch das reine ALOHA („pure ALOHA") wird also eine Kanalbelegung von gerade mal 18% erreicht.

Das ist ein bisschen wenig, deswegen ~~gab~~ gibt es eine Erweiterung, das sogenannte „unterteilte ALOHA" („slotted ALOHA")…..