

Übungen zur Vorlesung
Systeme II / Rechnernetze
Sommer 2016
Blatt 7 (10 Punkte)

AUFGABE 1:

5 Punkte

1. Betrachten Sie das ALOHA-Protokoll zur Kollisionsvermeidung. Die Pakete erscheinen auf dem Medium gemäß Abbildung 1. Welche Pakete sind im ALOHA-Protokoll kollisionsfrei?

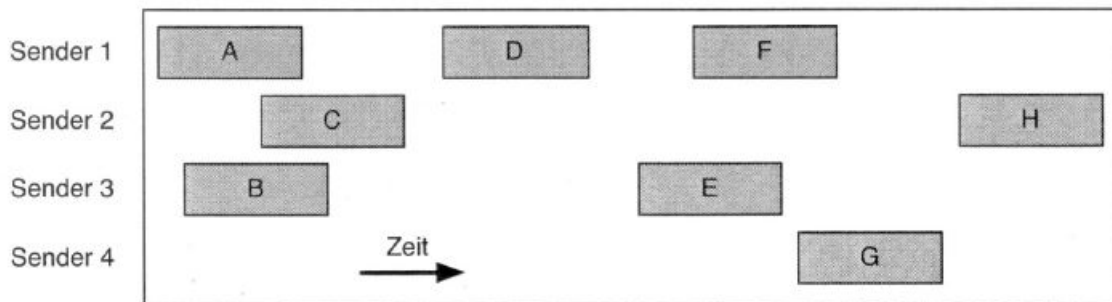


Abbildung 1: ALOHA

2. Benutzen Sie nun Slotted-ALOHA mit Zeitslots wie in Abbildung 2. Zeichnen Sie die gesendeten Pakete in die vorgegebenen Slots ein.

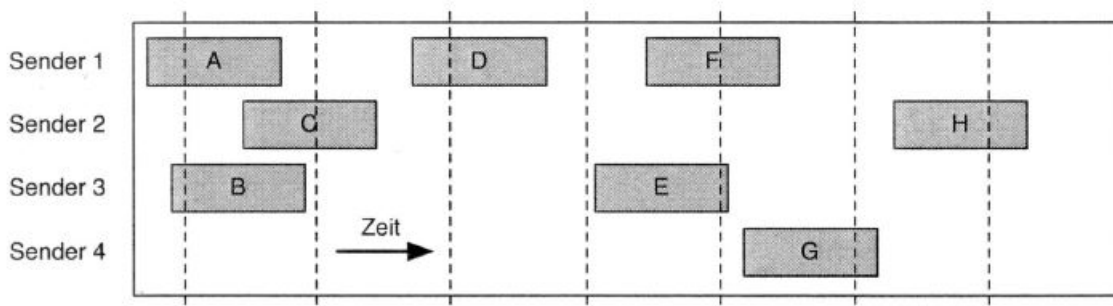


Abbildung 2: Slotted ALOHA

3. Welche Pakete sind im Slotted-ALOHA-Protokoll kollisionsfrei?
4. Welches der beiden Verfahren ist in diesem Fall, welches im Allgemeinen effektiver? Begründen Sie.
5. Wie kommen kollidierte Pakete am Ende doch noch zum Ziel?

AUFGABE 2:

5 Punkte

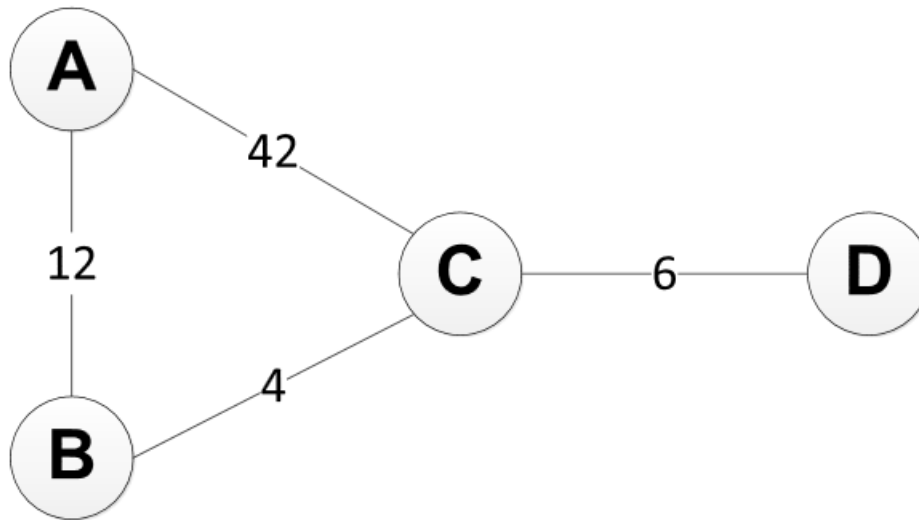


Abbildung 3: Ungerichtetes Netzwerk mit 4 Knoten

1. Führen Sie für den Graphen in Abbildung 3 den Distanzvektoralgorithmus aus, bis der Algorithmus konvergiert. Erzeugen Sie für $t = 0$ die initiale Kostenmatrix (d.h. die Tabellen enthalten nur die Kostenwerte der direkten Nachbarn). Sie können dann davon ausgehen, dass die Knoten beginnend mit A Informationen in alphabetischer Reihenfolge senden. Die Tabellen finden Sie unten UND auf der darauffolgenden Seite.
2. Löschen Sie eine Kante, um ein Count-to-Infinity Problem zu erzeugen. Was genau ist das für ein Problem? Wieso tritt es auf? Erklären Sie!

t = 0:

from A	via A	via B	via C	via D	from B	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				

from C	via A	via B	via C	via D	from D	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				

t = 1:

from A	via A	via B	via C	via D	from B	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				
from C	via A	via B	via C	via D	from D	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				

t = 2:

from A	via A	via B	via C	via D	from B	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				
from C	via A	via B	via C	via D	from D	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				

t = 3:

from A	via A	via B	via C	via D	from B	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				
from C	via A	via B	via C	via D	from D	via A	via B	via C	via D
to A					to A				
to B					to B				
to C					to C				
to D					to D				