

# Kommunikation in Sensornetzen

## Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

Energieeffizienz des Genie Aided Aloha (best case)

Im Gegensatz zum regulären Aloha sollte man versuchen, das sog. Idle-Listening zu vermeiden, in dem der Knoten Stille überhört. Es folgt eine Abschätzung des Potenzials von Aloha im Best-Case.

Der sogenannte Genie (ein zentraler Knoten) weckt Knoten rechtzeitig immer dann, wenn der Kanal nicht frei ist, also Nachrichten eintreffen.

Damit soll die Untergrenze für den Energieverbrauch im bestmöglichen Fall abgeschätzt werden. Es bleibt die Herausforderung den zentralen Knoten durch ein realisierbares Verfahren zu ersetzen.

Genie Aided Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

Alle Knoten inkl.  
des betrachteten  
erzeugen keine  
Nachricht

$$b = 1 - 1 e^{-g(N+1)}$$

Das komplementäre Ereignis, eine Nachricht trifft ein

Der Knoten selbst  
erz. eine Nachricht

$$P_{\text{ow}}^{\text{RA}} = b_1 P_{\text{TX}} + (b - b_1) P_{\text{RX}}$$

# Kommunikation in Sensornetzen

Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

Energieeffizienz des Genie Aided Aloha

Beispiel

$$P_{\text{Grundverbrauch}} = 8 \text{ mA}$$

$$P_{\text{TX}} = 20 \text{ mA}$$

$$P_{\text{RX}} = 6 \text{ mA}$$

Eine Rahmenzeit beträgt 10ms, d. h. 100 Rahmen/Sekunde wobei 10 Knoten 1x pro Sekunde senden wollen.

$$b_1 = 1 - \frac{(1g)^0}{0!} e^{-1g}$$

$$b = 1 - 1 e^{-g(N+1)}$$

$$P_{\text{ow}}^{\text{GA}} = b_1 P_{\text{TX}} + (b - b_1) P_{\text{RX}}$$

$$g=0,01 \Rightarrow b_1 = \sim(1-0,99) = 0,01$$

$$g \times 10 = 0,1 \Rightarrow b = \sim(1-0,90) = 0,095$$

$$P^{\text{RA}} = 0,01 \times (20+8) + (0,095-0,01) \times (6+8) = 1,47 \text{ mA}$$

Batterie mit 2000 mAh reicht für  $2000 \times 60 \times 60 / 1,47 = \sim 1361$  Stunden

Die theoretisch mögliche Obergrenze für die Lebenszeit liegt bei etwa 57 Tagen.

Genie Aided Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

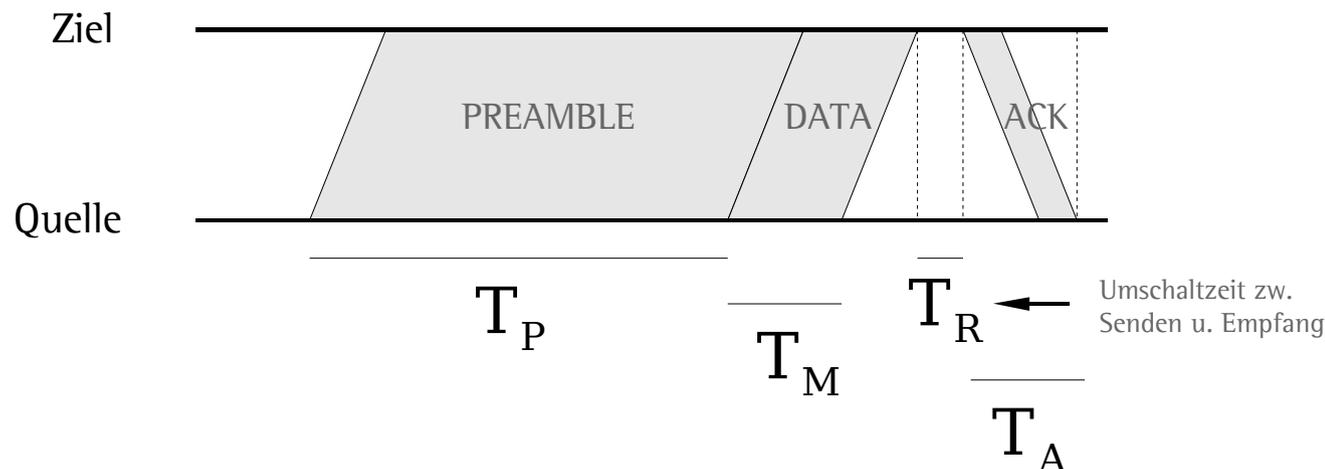
## Kommunikation in Sensornetzen

### Energieeffiziente MAC Protokolle: Aloha mit Preamble Sampling

**Idee:** In der Regel schlafen Knoten. Gesteuert durch ihre interne Uhr wachen sie jedoch nach jeder Schlafperiode auf und hören den Kanal eine minimale Zeit ab. Ist keine Aktivität vorhanden, so gehen sie sofort darauf wieder in den Schlafzustand über.

**Ein Knoten will senden:** Bevor er seine Nachricht an alle Teilnehmer senden kann, müssen alle wach sein. Um dies zu erreichen erzeugt der Sender ein ununterbrochenes Signal, die Preamble, welches mindestens eine Schlafperiode anhalten muss. Währenddessen wachen die Knoten nach und nach auf, bemerken die Aktivität auf dem Kanal und kehren daher nicht mehr in den Schlafzustand zurück. Nach einer vollen Rahmenzeit hat der Sender so alle aufwachenden Knoten „aufgesammelt“ und kann seine Nachricht verschicken.

Meist wird noch ein Acknowledgement des Empfängers vorgesehen, damit im Fehlerfall nicht nochmal ein voller Schlafzyklus verbraucht werden muss.



Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

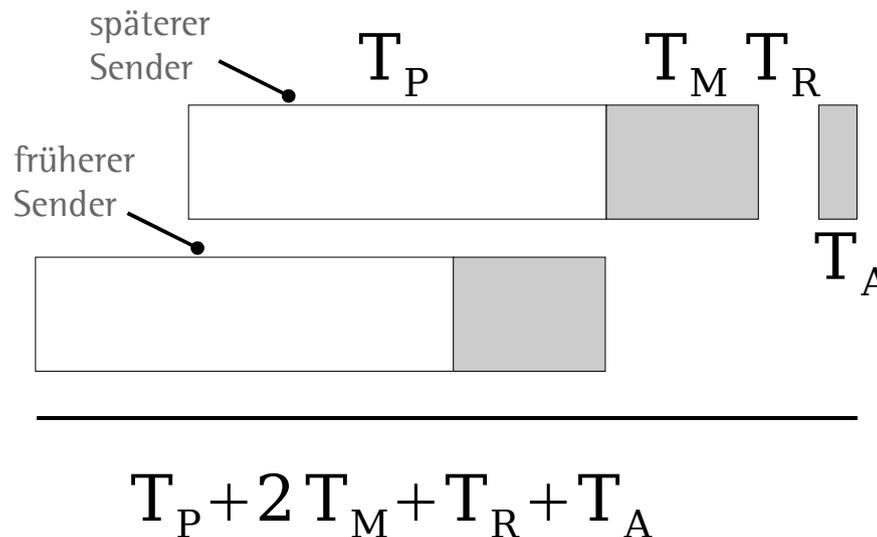
## Energieeffiziente MAC Protokolle: Aloha mit Preamble Sampling

### Energieeffizienz von Aloha mit Preamble Sampling

Hier gilt eine Besonderheit für die Zeit benötigter Still, die ein Knoten braucht um eine Nachricht fehlerfrei zu übertragen:

Ein Sendezyklus dauert grundsätzlich die Preamble Zeit  $T_P$ , die Nachrichten Zeit  $T_M$ , die Umschaltzeit  $T_R$  und die Zeit für das ACK  $T_A$ . Bei flüchtiger Betrachtung könnte man meinen, dass ein Knoten 2 dieser Zyklen benötigt, um eine Nachricht kollisionsfrei zu übertragen. Tatsächlich ist jedoch die unten angegebene Zeit  $T_P + 2T_M + T_R + T_A$  ausreichend. Man beachte zum Verständnis, dass eine Kollision innerhalb der Preamble-Zeit (zumindest für den 2. Sender) keine Probleme bereitet.

Die abgebildete Kollision ist gerade noch erlaubt, so dass der spätere Sender seine Nachricht übertragen kann, denn eine Kollision innerhalb der Preamble bereitet keine Probleme. Man beachte auch das Ausbleibende ACK des Empfängers der ersten Nachricht.



Genie Aided Aloha

Aloha m. Preamble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster Protokolle

Simulation

# Kommunikation in Sensornetzen

## Energieeffiziente MAC Protokolle: Aloha mit Preamble Sampling

Energieeffizienz des Aloha mit Preamble Sampling

Die Angaben für  $T_P, T_M, T_R$  und  $T_A$  verstehen sich in Anteilen an einer Rahmenzeit, also z. B.  $T_P=1$ , damit die Knoten 1 x pro Rahmenzeit aufwachen.

$$P_S^{PAS} = e^{-gN(T_P + 2T_M + T_R + T_A)}$$

W' für eine freie kritische Phase

$$b^{PAS} = 1 - e^{-g(N+1)(T_P + T_M + T_R + T_A)}$$

W' für eintreffende Nachricht

$$b_1^{PAS} = 1 - e^{-g(T_P + T_M + T_R + T_A)}$$

W' dass Sender Nachricht erzeugt

$$P_{ow}^{PAS} = b_1^{PAS} P_{TX} + (b^{PAS} - b_1^{PAS}) P_{RX}$$

Mittlerer Energieverbrauch

Bemerkung: Der Energieverbrauch für das kurze Aufwachen ist hier noch nicht enthalten.

Genie Aided Aloha

Aloha m. Preamble Sampling

Slotted Aloha

CSMA-Varianten

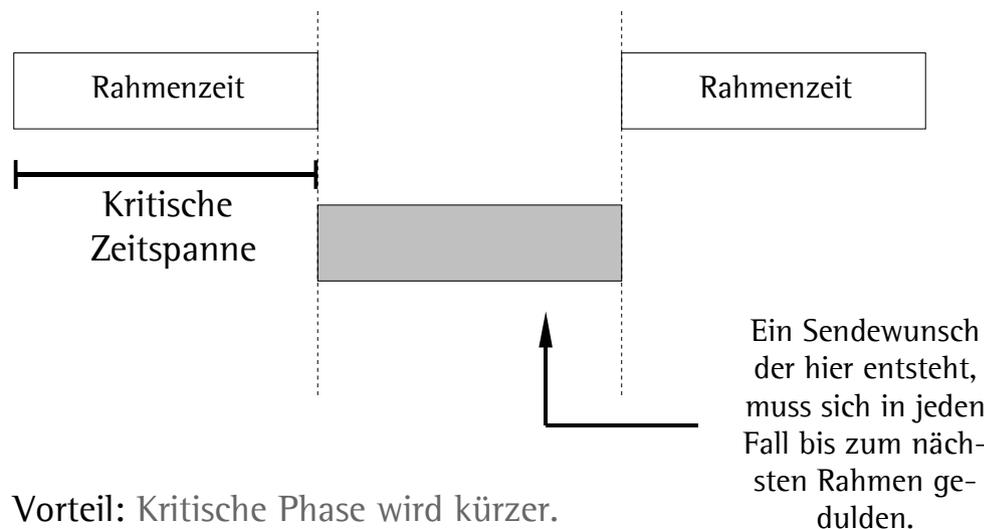
Bitmuster Protokolle

Simulation

## Kommunikation in Sensornetzen

### Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

**Slotted ALOHA:** Die Zeit ist in feste Intervalle eingeteilt. Jedes Intervall ist genau einen Rahmen lang. Nur am Anfang des Rahmens darf gesendet werden. Auch hier ist noch wenig gegenseitige Rücksichtnahme implementiert. Jedoch müssen die Teilnehmer synchronisiert sein.



**Bemerkung:** Die kritische Phase ist hier nicht der gesendete Rahmen selbst, sondern die Rahmenzeit zuvor. Denn hier entscheidet sich, ob ein oder mehr Sender den kommenden Rahmen belegen wollen. Hat ein Sender den Rahmen jedoch gewonnen, so ist dieser unkritisch, denn per Def. des Verfahrens wird dann kein weiterer Sender hinzukommen.

**Vorteil:** Kritische Phase wird kürzer.

**Nachteil:** Bilden sich zwei Sendewünsche, so ist die Kollision im nächsten Rahmen sicher. Man beachte den immer noch geringen Paketdurchsatz. Außerdem wird auch dann auf den nächsten Rahmen gewartet, wenn niemand sendet, d. h. es entsteht fast immer eine Verzögerung.

Wie verbessert sich der Paketdurchsatz für Slotted Aloha?

Genie Aided Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

## Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

### Protokolle mit Trägererkennung: 1-Persistent CSMA

Größtes Problem der ALOHA-Varianten ist, dass keine Rücksicht auf andere Teilnehmer genommen wird. Abhilfe schafft die sog. Trägererkennung (Carrier Sense Multiple Access).

Bevor eine Station sendet, wartet sie, bis der Kanal frei ist. Erst dann wird das Paket gesendet. Tritt eine Kollision auf, so greift die gewöhnliche Fehlerbehandlung. Die Station wartet eine zufällige Zeit und sendet erneut, jedoch nur wenn der Kanal frei ist. Hatten zwei Stationen zum gleichen Zeitpunkt einen Sendewunsch, so wird nach der Kollision diejenige mit der kürzeren Wartezeit ihr Paket zuerst verschicken.

Das Verfahren heißt 1-Persistent CSMA, weil mit einer Wahrscheinlichkeit von 1.0 gesendet wird, wenn der Kanal frei wird.

**Vorteil:** Flexibler als slotted ALOHA, da nicht in jedem Fall auf das Rahmenende gewartet wird. Statt dessen kann ein Knoten (bei günstigen Bedingungen) zu jedem Zeitpunkt anfangen zu senden.

**Nachteil:** Wollen zwei Stationen senden während eine Übertragung läuft, so tritt mit Sicherheit eine Kollision am Anfang des nächsten Rahmens auf. In diesem Fall gibt es keinen Vorteil gegenüber slotted ALOHA.

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

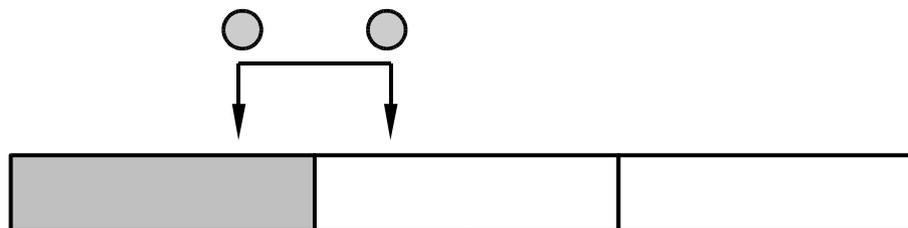
## Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

### Protokolle mit Trägererkennung: Non-Persistent CSMA

Unterschied zu 1-Persistent CSMA: Der Kanal wird nicht ständig auf das Ende der laufenden Übertragung abgehört. War der Kanal nicht frei, so wird eine zufällige Zeitspanne gewartet. Erst nach deren Ablauf wird **erneut geprüft**, ob der Kanal frei ist.

**Vorteil:** Die Wahrscheinlichkeit, dass gleichzeitig mit dem Senden begonnen wird sinkt drastisch (quasi Null), eine Kollision ist dadurch sehr unwahrscheinlich. Hier wird der Fall einer erneuten Übertragung aufgrund einer Kollision (wie bei 1-Persistent CSMA) gewissermaßen gleich vorweggenommen.

**Nachteil:** Obwohl z. B. zwei Stationen senden wollen, beginnt keine die Übertragung nach Beendigung der aktuellen Übertragung, da beide eine zufällige Zeitspanne warten. Dass das Ende einer Wartezeit genau auf den Anfang des nächsten verfügbaren Rahmens fällt, ist sehr unwahrscheinlich, wäre aber wünschenswert. In jedem Fall erhöht sich das mittlere Delay.



Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

**CSMA-Varianten**

Bitmuster  
Protokolle

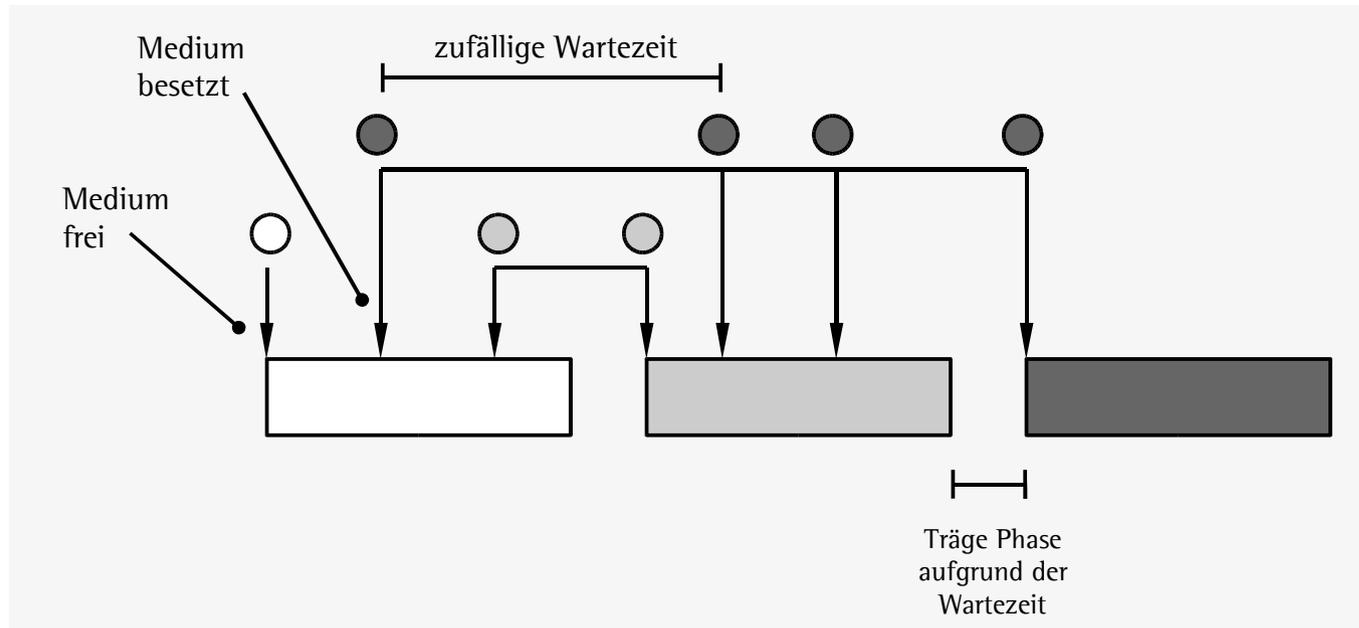
Simulation

## Kommunikation in Sensornetzen

Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

Protokolle mit Trägererkennung: Non-Persistent CSMA

○ Knoten mit Sendewunsch



Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

Protokolle mit Trägererkennung: CSMA/CD (CD=collision detection)

Eine gestörte Übertragung kann vom Sender rel. früh erkannt werden (= Kollisionserkennung), insbes. bei kabelgebundener Übertragung. Dann macht es für beide Stationen Sinn die Übertragung abubrechen, um wertvolle Bandbreite zu sparen, die andernfalls für ein Paket aufgewendet würde, das ohnehin nicht mehr zu retten ist.

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

**CSMA-Varianten**

Bitmuster  
Protokolle

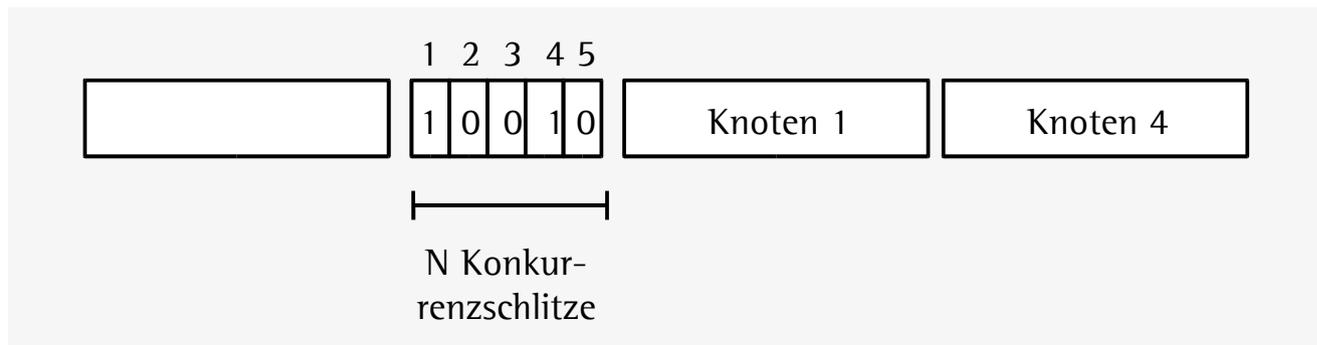
Simulation

# Kommunikation in Sensornetzen

Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

Kollisionsfreie Protokolle: Bitmuster Protokoll

Vor dem Senden wird ein kurzer Rahmen vereinbart. Jede Station hat dabei ein schmales Zeitfenster, in dem bei Sendewunsch eine 1 (ein Signal) gesetzt werden kann. Danach wird in der so vereinbarten Reihenfolge gesendet.



**Vorteil:** Kollisionen werden vollständig vermieden

**Nachteil:** Die N Konkurrenzschlitze müssen immer abgewartet werden, auch wenn nur eine Station existiert/senden will. Ist N sehr groß, geht hier ein Teil der Bandbreite verloren. Eine gute Synchronisation ist notwendig.

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

## Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

### Kollisionsfreie Protokolle: Binary Countdown

**Verfahren:** Hier bekommt jede Station eine (fortlaufende) Nummer, die z. B. mit (hier) 4 Bit darstellbar ist. Die Anzahl der Zeitslitze reduziert sich so auf 4. Es seien folgende Stationen wach:

<u>Station</u>	<u>ID</u>
4	0010
5	101
9	1001

Im Zeitschlitz des ersten Bits senden Station 5 und 9 einen Impuls, weil ihr  $2^0$  Bit gesetzt ist. Station 4 schweigt aufgrund des nicht gesetzten Bits, weiß jedoch gleichzeitig, dass sie aus dem Rennen ist, da ihr Bitpattern dem bisher gesendeten bereits nicht mehr entspricht.

Im Zeitslot des Bits  $2^1$  sendet niemand, so dass zumindest Station 5 und 9 beide noch eine Chance haben. Erst zur Zeitpunkt des  $2^2$  Bits hat Station 5 mit einem gesendeten Impuls gewonnen, da das bisherige Bitmuster schon nicht mehr für Station 9 paßt.

**Vorteil:** Kurzes Bitmuster

**Nachteil:** Nach wie vor ein Bitmuster pro Rahmenzeit, ungleiche Verteilung des Senderechts.

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

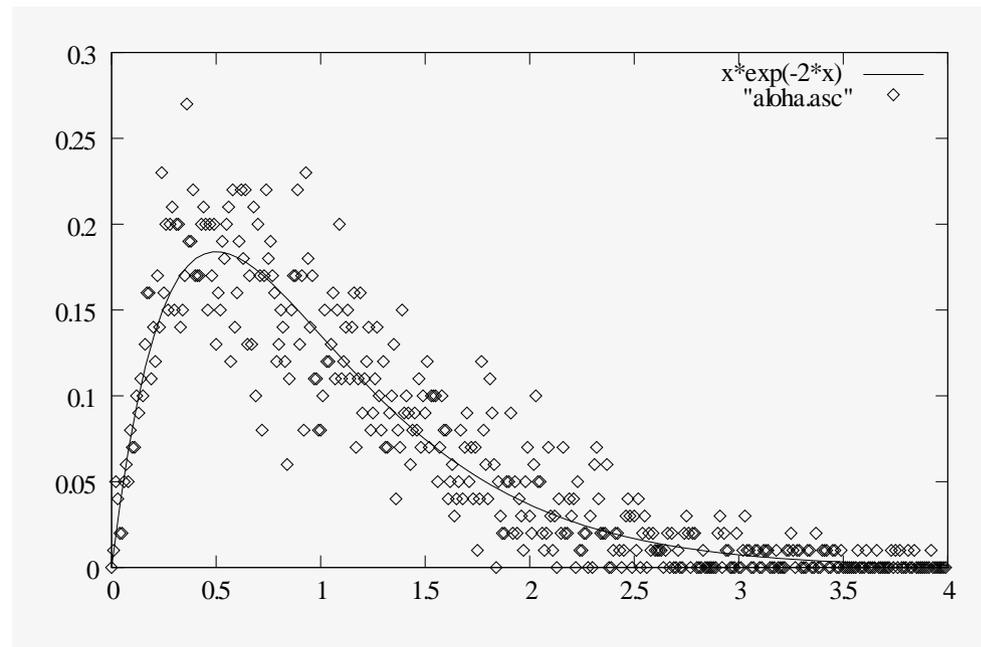
Simulation

## Kommunikation in Sensornetzen

Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

Simulation unterschiedlicher MAC-Protokolle

Nach der analytischen Überlegung soll das Aloha Verfahren mit einem kleinen Programm simuliert werden.



Bei 100 Stationen, 100 Rahmenzeiten, jeweils unterteilt in 100 Simulations-Zeiteinheiten paßte sich die Verteilung einigermaßen an die Kurve an. Warum sind in der „Praxis“ dennoch in manchen Simulationsläufen deutlich höhere oder niedrigere Datendurchsätze zu verzeichnen?

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

## Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

### Simulation unterschiedlicher MAC-Protokolle

```
const long MAX_TIME           = 10000L;
const long NO_STATIONS        = 100L;
const long FRAME_LENGTH       = 100L;
    long medium_occupied_till = -1;
    long survival_timer        = -1;
    long successful_packets     = 0;
    long no_frames              = MAX_TIME / FRAME_LENGTH;

class Station
{
public:
    void Init() {};
    void TriggerSend(long);
}; // class Station

void Station::TriggerSend(long current_time)
{
    if(current_time < medium_occupied_till) {
        survival_timer = -1;
    } // if
    else survival_timer = 0;

    medium_occupied_till = current_time + FRAME_LENGTH;
} // Station::TriggerSend
```

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation

# Kommunikation in Sensornetzen

## Klassische Medium Access Control (MAC) Protokolle für Sensorknoten?

### Simulation unterschiedlicher MAC-Protokolle

```
main()
{
    Station* station          = new Station[NO_STATIONS];

    for(long i = 0; i < NO_STATIONS; i++)
        station[i].Init();

    for(long arrival_rate = 0; arrival_rate < 400; arrival_rate += 1) {
        medium_occupied_till = -1; survival_timer = -1; successful_packets = 0;

        for(long time = 0; time < MAX_TIME; time++) {
            for(long station_index = 0; station_index < NO_STATIONS; station_index++)
                if((abs(rand()) % (100*NO_STATIONS*FRAME_LENGTH)) < arrival_rate)
                    station[station_index].TriggerSend(time);

            if(survival_timer != -1) survival_timer++;
            if(survival_timer == FRAME_LENGTH) {
                successful_packets++;
                survival_timer = -1;
            } // if
        } // for

        double overall_arrival_rate = ((double)arrival_rate)/100.0;

        cout << overall_arrival_rate << " " << (double)successful_packets/(double)
no_frames << endl;
    } // for

    delete[] station;
} // main
```

Genie Aided  
Aloha

Aloha m. Pre-  
amble Sampling

Slotted Aloha

CSMA-Varianten

Bitmuster  
Protokolle

Simulation