

Kommunikation in Sensornetzen

Probleme der MAC-Schicht in Sensornetzen (Funknetzen):

Folgende Sachverhalte machen im Sinn der Energieeffizienz Probleme:

- idle listening:** Ein Knoten hört das Medium ab, obwohl kein Sender aktiv ist. Da ohne spezielle Vorkehrungen nicht bekannt ist, wann der nächste Sender aktiv wird, kann auf das Abhören nicht ohne weiteres verzichtet werden.
- overemitting:** Ein Knoten sendet eine Nachricht, wobei der Empfänger zumindest zeitweise nicht in der Lage ist, die Nachricht zu hören.
- overhearing:** Ein Knoten hört eine Nachricht (vollständig), die jedoch an einen anderen Adressaten gerichtet ist. Ohne spezielle Vorkehrungen muss die Nachricht unnötigerweise überhört werden, um die darauf folgende Nachricht zu erhalten.
- collisions:** Zwei Knoten übertragen zur gleichen Zeit. Dabei zerstören sich beide Nachrichten gegenseitig beim Empfänger. Mindestens zwei Sender verschwenden Sendeenergie, je nach Verfahren sind mögl. alle Empfänger beteiligt und es entsteht keinerlei Nutzen.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

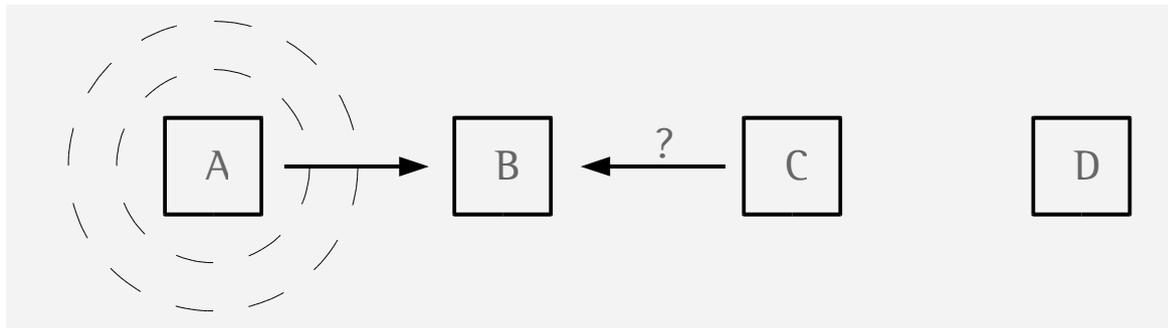
AMRIS

Kommunikation in Sensornetzen

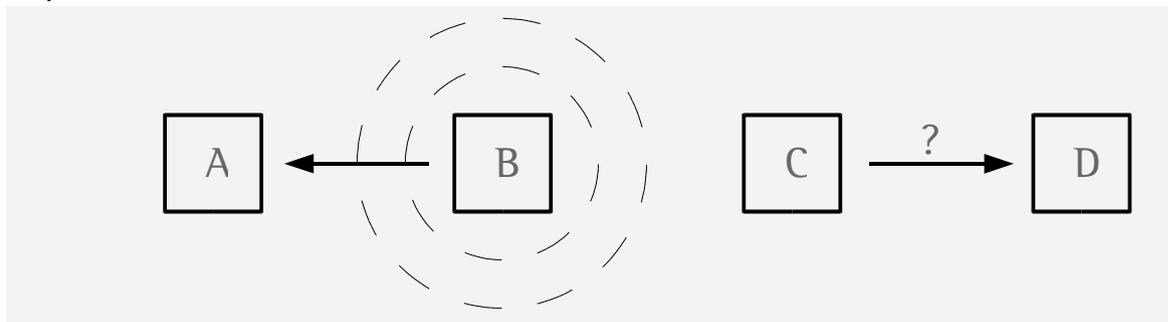
Probleme der MAC-Schicht speziell in Funknetzen:

Im Gegensatz zu kabelbasierten Kanälen ist die Sendereichweite einer Station in Funknetzen begrenzt. Hieraus entsteht das Problem, dass eine Station durch Senden in einer Entfernung Kollisionen erzeugt, die sie selbst nicht mehr detektieren kann (Hidden Station Problem) oder bei sich lokal eine Kollision erkennt, die beim Empfänger gar nicht mehr hörbar ist (Exposed Station Problem):

Hidden Station Problem:



Exposed Station Problem:



MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

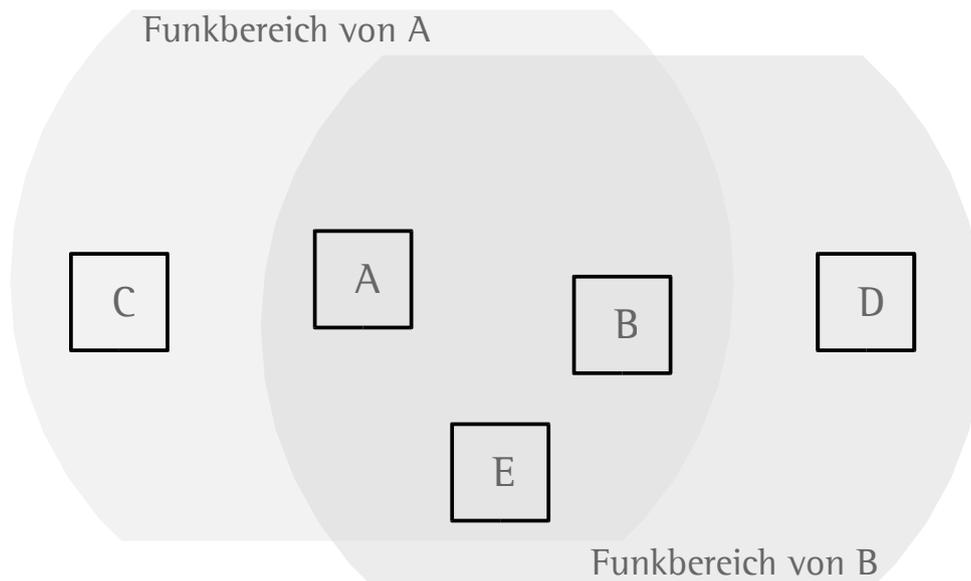
AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

Unterschied zwischen Kabel- und Äther-gebundenem Medienzugriff: Im Kabel gilt, dass die Verhältnisse beim Sender denen beim Empfänger gleichen. Der Sender kann aus seinen lokalen Beobachtungen ableiten, ob ein Versenden für den Empfänger Sinn macht. Grundsätzlich interessiert jedoch nur, ob der Empfänger ungestört hören kann. In Funknetzen kann jedoch aus den lokalen Beobachtungen nicht auf die entfernten geschlossen werden.

Lösung: Die Detektion erfolgt nicht „elektrotechnisch“, sondern muss per Protokoll gehandhabt werden.



Bemerkung: In der Realität hat ein Sender neben seinem Senderradius auch noch einen Störradius, in dem seine Pakete nicht mehr verstanden werden, jedoch immer noch Kollisionen verursachen. Auch die Kreis- bzw. Kugelumgebung um einen Sender ist eine Vereinfachung. Durch Interaktion mit der Umgebung sehen die wahren Empfangsbereiche meist anders aus.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Das RTS/CTS-Schema:

Implementiert in MACA (Medium Access with Collision Avoidance) und übernommen in 802.11 zum Ausloten eines ungestörten Empfangs:

- (1) Sender A fragt bei B an: Request to Send (RTS). Ein RTS Paket enthält den Empfänger der Nachricht und die Länge der Nachricht, die folgen soll.
 - (2) Kommt das RTS Paket beim Empfänger an, so sendet dieser: Clear to Send (CTS). Das CTS Paket enthält die aus dem RTS bekannte Länge der Nachricht, die demnächst zu erwarten ist.
 - (3) Nachdem der ursprüngliche Sender das CTS erhalten hat, wird die eigentliche Nachricht gesendet.
- Optional: (4) Der Empfänger kann anhand der Prüfsumme nachvollziehen, ob die Nachricht (wahrscheinlich) richtig angekommen ist und sendet daraufhin ein Acknowledgement (ACK).

Alternativ: Es kann auch der höheren Transportschicht überlassen werden das Fehlen der Nachricht zu detektieren und diese erneut anzufordern (bzw. durch nicht-Bestätigung den erneuten Versandt zu provozieren). Allerdings dauert dies deutlich länger und erfordert eine neue contention-Phase, (freie Medium, RTS/CTS usw.)

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Das RTS/CTS-Schema:

Ausloten eines ungestörten Empfangs

Was machen die anderen?

- (1) Ein RTS könnte mit einem weiteren RTS kollidieren, z. B. wenn E auch senden will. Beide Sender erhalten dann kein CTS, da ihre Nachricht (evtl. von ihnen selbst unbeachtet) beim Empfänger kollidiert ist. Beide warten jedoch eine zufällig gewählte Zeit bis sie annehmen, dass kein CTS mehr zu erwarten ist. Der Ungeduldigere wird zuerst einen weiteren Versuch unternehmen, der Geduldigere hört dies und hält sich zurück.
- (2) Das RTS wird z. B. auch von C gehört. C erkennt jedoch, dass er nicht der Empfänger ist. Andererseits: Würde C senden, so stört dies B nicht, da sich beide ohnehin nicht hören können. Trotzdem muss C solange ruhig halten, bis wenigstens ein mögliches CTS sicher beim Sender angekommen ist. Ansonsten würde C die Kommunikation zwischen A und B sabotieren. Jedoch: Danach könnte C versenden. Zwar versteht A dadurch sein „eigenes Wort“ nicht, dies hat aber auf den Empfang bei B keine Auswirkung.
- (3) Auch D hört das CTS von B. Da B bewußt die Länge des erwarteten Paketes in das CTS-Paket eingebaut hat weiß D, dass sie mindestens bis Ende des Nutzdatenpaketes von A still halten muss, auch dann, wenn sie dieses Paket überhaupt nicht hören kann.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Das RTS/CTS-Schema:

Ausloten eines ungestörten Empfangs

Wie wird der Zugang zum Kanal bei 802.11 geregelt?

Nach einer erfolgreichen Übertragung beginnt grundsätzlich eine neue Rahmenzeit. Nun können sendewillige Knoten ihren Sendewunsch bekannt machen. Zuvor muss jedoch noch die Zeit gewartet werden, die ein ACK benötigt. Würde das ACK nicht beim Empfänger ankommen, so müßte die letzte Übertragung als ungültig angenommen werden. Danach beginnt eine Contention-Phase mit 31 Zeitschlitzten.

Jeder Sender zieht eine Zufallszahl z im Intervall $[0,31]$ und hört den Kanal bis zu diesem Zeitschlitz ab. Wurde dieser zuvor von einem anderen Sender belegt, so stellt der spätere Sender seinen Sendewunsch zurück. Anonsten hat er den Kanal gewonnen, es sei denn ein anderer Sender hat ausgerechnet die gleiche Zahl gezogen. Dann werden sich beide RTS Pakete überlagern und ein CTS ist nicht zu erwarten.

Darüber hinaus werden die Zeitschlitzte in der Contention Phase auf 63 verdoppelt, ein weiterer Misserfolg führt wieder zu einer Verdoppelung usw. So paßt sich das Protokoll automatisch einer unterschiedlichen Anzahl von Teilnehmern an.

Warum werden diskrete Zeitschlitzte und nicht stetig verteilte Wartezeiten verwendet?

Antwort: Weil das Umschalten zwischen Empfangen und Senden eine gewisse Zeit in Anspruch nimmt. Der Zeitschlitz ist so groß gewählt, dass Schaltzeiten berücksichtigt sind.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

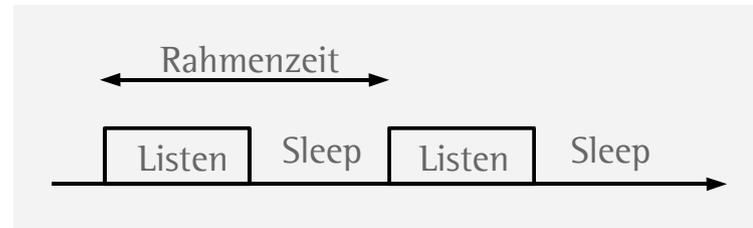
AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

S-MAC Protokoll: Eine Adaption für Sensornetze

Prinzip: Alle Knoten haben einen sog. Ablaufplan (schedule), der die Zeit in Listen- und Sleep-Perioden einteilt. Knoten versuchen sich in möglichst großen Verbänden zusammen zu finden, die diesem Schlaf- und Wachschema folgen.



MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Wie ist das Verhältnis zwischen Listen und Sleep Zeit zu optimieren?

Beteiligte Größen: Max. anfallende Datenrate (bestimmt durch natürliche Ereignisse und Sensoren), max. Senderate, Puffergröße des Knoten, tolerierbare Latenz

Listen-Periode: So kurz wie möglich, da energie-intensiv. Muss jedoch für max. anfallende Datenrate ausreichend sein. Sonst müssen Daten nach Ende der Sendeperiode verworfen werden.

Sleep-Periode: So lang wie möglich. Jedoch müssen Daten während dessen gepuffert werden, d. h. Puffergröße begrenzt die Sleep-Länge.

ges. Rahmenzeit = Listen+Sleep: Muss max. Latenzzeit gewährleisten. Andererseits: Das Ein-/Ausschalten der RF-Unit braucht auch Zeit u. Energie.

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

S-MAC Protokoll: Eine Adaption für Sensornetze

Initialisierungsphase

Synchronizer: Nach dem Ausbringen der Knoten warten alle eine zufällige Zeit und hören, ob sich ein anderer Knoten meldet. Ist dies erfolglos, so sendet der ungeduldigste Knoten eine SYNC-Nachricht. Sie enthält die Information, dass der Knoten in t Sekunden in den Schlafzustand übergeht. Der erste Knoten wird auch als Synchronizer bezeichnet, alle anderen als Follower.

Follower: Alle Knoten in der Sendenumgebung hören die initiale SYNC-Nachricht und beschließen zum gleichen Zeitpunkt in den Schlafzustand zu gehen, wie der Synchronizer. Sie warten selbst eine zufällige Zeit d und verkünden nach Ablauf der d Sekunden, dass sie in $t-d$ Zeiteinheiten schlafen werden, ebenso wie der Synchronizer. Warum die Wartezeit? Weil sonst alle zur gleichen Zeit senden würden und so sichere Kollisionen erzeugen würden. Durch die erneuten SYNCs werden auch weitere Knoten im Umkreis noch in den virtuellen Cluster aufgenommen, so diese nicht schon einem anderen Synchronizer folgen.

Alle Empfänger: Jedes SYNC-Paket teilt auch allen Empfängern mit, welche Knoten in Sendereichweite sind und welchem Ablaufplan diese angehören. Dies wird oft, jedoch nicht zwingend der eigene Ablaufplan sein. Das Wissen über die „anderen“ Wachzeiten ist zum Adressieren der Nachbarn wichtig.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

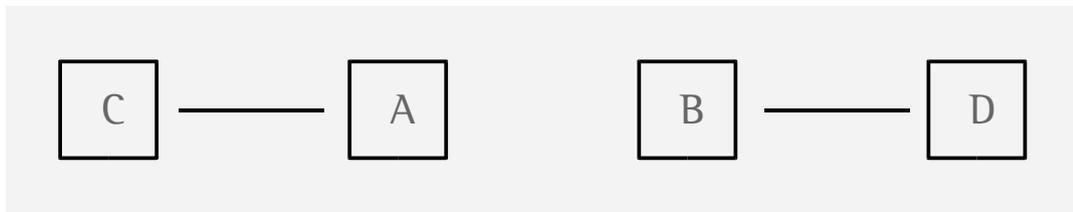
AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

S-MAC Protokoll: Eine Adaption für Sensornetze

Was geschieht mit Knoten die in Sendereichweite sind, jedoch bereits unterschiedlichen virtuellen Clustern angehören, im Beispiel unten also A und B?



Strategie 1 (Empfangen zum Zeitpunkt der anderen): Ein Knoten nimmt beide Ablaufpläne auf. Vorteil: Er kann an beiden Clustern teilnehmen und fungiert als Transponder der Nachrichten von Cluster zu Cluster transportieren kann, insbesondere auch Broadcast Nachrichten. Nachteil: Er hat muss eine doppelt so lange Wachzeit wie gewöhnliche Knoten einhalten. Obwohl er eine wichtige Funktion zwischen den Clustern hat, wird er viel früher ausfallen als andere Knoten.

Strategie 2 (Senden zum Zeitpunkt der anderen): Der Knoten bleibt zwar bei dem Zeitablauf den er zuerst angenommen hat, speichert aber dennoch die Knoten anderer Cluster in Sendereichweite, zusammen mit deren Ablaufplänen. So kann er die anderen Knoten bei Bedarf erreichen (durch Frühaufstehen). Broadcast Nachrichten anderer Cluster können so aber nicht sicher an ihn zugestellt werden.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

S-MAC Protokoll: Eine Adaption für Sensornetze

Ausnahmesituation:

Von Zeit zu Zeit muss jeder Knoten eine volle Rahmenzeit den Kanal abhören. Warum?

Weil neue Knoten hinzukommen könnten, die sich evtl. bereits mit anderen Nachbarn synchronisiert haben. Diese neuen Nachbarn können nur durch vollständiges Abhören der Rahmenzeit sicher entdeckt werden. Möglicherweise verbinden die neuen Nachbarn auch zwei Teilnetze, die sich zuvor noch nicht kannten.

Wann sollte ein Knoten eine Broadcast-Nachricht für alle versenden?

- (a) zum Zeitpunkt des eigenen Ablaufplans
- (b) zum Zeitpunkt des Ablaufplans eines Nachbarclusters?

Antwort: Immer zum Zeitpunkt des eigenen Ablaufplans. Denn zum. bei Strategie 1 ist gesichert, dass dann alle zuhören, auch die, die nicht zum eigenen Cluster gehören. Würde der Knoten zum Zeitpunkt des Ablaufplans eines Nachbarclusters senden, so gäbe es wahrscheinlich Knoten innerhalb der Sendereichweite, die schlafen.

Jedoch schläft bei Strategie 1 zum Zeitplan des Senders niemals ein Knoten in der Reichweite des Senders. Warum? Weil dieser seinen eigenen Zeitpunkt verkündet hat und alle ihn speichern konnten.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

S-MAC Protokoll: Eine Adaption für Sensornetze

Zeitsynchronisation

Die Wachphase ist in zwei Abschnitte eingeteilt, nämlich eine SYNC- und eine RTS-Phase. Die SYNC-Phase wird dazu verwendet die Timer wieder anzugleichen um eine langfristige Drift zu verhindern.

Jeder Knoten sendet in größeren Abständen ein SYNC-Paket der Art, wie es auch aus der Initialisierungsphase bekannt ist. Dieser SYNC hat (1) die Aufgabe andere Knoten neu zu synchronisieren und (2) neu hinzukommenden Knoten die Möglichkeit zu geben sich einem Cluster zuzuordnen. Der Zugang zum Medium wird in beiden Fällen über einen zufällig gewählten Wartetimer ausgehandelt. Derjenige, der das Medium zuerst belegt gewinnt es auch. In der Sync-Phase ist der erste Knoten auch der, der seine Zeit verkünden darf.



MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

S-MAC Protokoll: Eine Adaption für Sensornetze

Kollisionsvermeidung und Vermeidung von Overhearing

Prinzipiell funktioniert die Kollisionsvermeidung nach dem RTS/CTS Schema. Jedoch sollte ein Knoten immer dann in den Schlafzustand gehen, wenn ein Paket für einen fremden Adressaten gesendet wird. Dies ist möglich, da sowohl im RTS als auch im CTS Paket die Länge des anstehenden Paketes enthalten ist.

Will der Knoten noch weitere Aufgaben ausführen, sollte zumindest die Funkeinheit abgeschaltet werden. Dann muss ein Zähler, der sog. Network Allocation Vector NAV mit der Wartezeit gesetzt werden. Jede Zeiteinheit wird der NAV (z. B. Interruptgesteuert) um eins dekrementiert. Der Knoten darf sich das Senden erst dann erlauben, wenn $NAV == 0$ ist. Andernfalls ist bekannt, dass das Medium ohnehin belegt ist, auch ohne dies „physisch“ zu testen. Dies wird auch als **virtual carrier sense** bezeichnet.

Am Ende des virtual carrier sense wird das Medium tatsächlich abgehört und nach einer zufälligen Wartezeit der Versuch einer Belegung gestartet (mit RTS/CTS). Ansonsten muss wieder für die Länge des kommenden Paketes gewartet werden.

Fazit: Vermeidung von Overhearing durch virtual carrier sense

Kollisionsvermeidung durch Zufallstimer und physical carrier sense in Verbindung mit RTS/CTS Schema

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

Verbesserung T-MAC:

Grundsätzliches Problem von S-MAC sind rel. lange aktive Zeiten. Diese müssen jedoch so lange sein, um den Fall des größten Datenaufkommens handhaben zu können. In allen anderen Fällen muss man unnötig lange Wartezeiten in Kauf nehmen. Eine intuitive Lösung wäre...

einen Timeout TA zu definieren. TA sollte ein wenig länger als die längste Backoff-Time (max. zufällige Wartezeit f. Kanalzugriff) gewählt werden. Ist dieser Timeout abgelaufen, so liegt möglicherweise kein erneuter Kommunikationswunsch vor. Es ist dann nahelegend, dass sich daraufhin alle Knoten in den Sleep-Mode versetzen.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

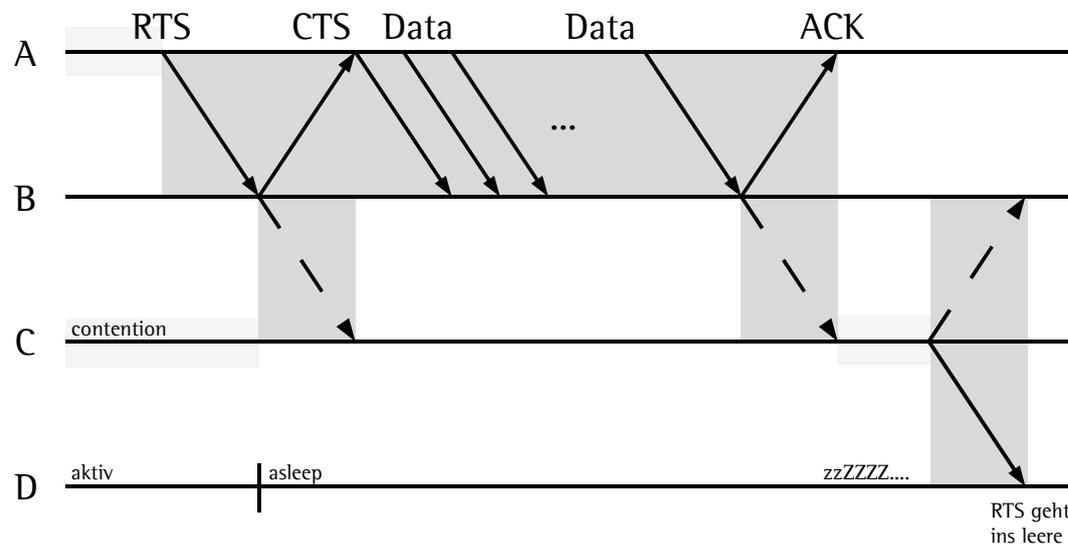
Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

Verbesserung T-MAC:

Problem: Der Ansatz funktioniert innerhalb einer 1-Hop Nachbarschaft. Gleichzeitig können aber Sendewünsche in der Nachbarschaft von Sender und Empfänger in der weiteren Umgebung entstehen.

Für die weitere Nachbarschaft ist jedoch charakteristisch, dass sie von der Kommunikation zwischen Sender und Empfänger (abhängig von der Distanz) nichts mitbekommt. Die verhinderten Sender dürfen ihre Sendewünsche auch nicht bekunden, so dass die Knoten in drei Hops Entfernung in den sleep-Mode wechseln und erst wieder in der nächsten Wachphase erreichbar sind.



Situation: A will an B senden und C will an D senden. Jedoch darf C dies nicht mitteilen und verliert D so nach einiger Zeit.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

Verbesserung T-MAC:

Lösung 1: Der vorerst verhinderte Knoten B darf seinen zukünftigen Sendewunsch doch bekunden. Im Beispiel wartet Knoten C noch das CTS von B abm sendet dann aber sein eigenes FRTS (Future Ready-To-Send). Normalerweise würde dieses mit den Daten von A kollidieren und Datenpakete bei B verstümmeln. Da A dies weiß, sendet er ein DS (Data-Send). Das DS hat keine tiefere Bedeutung und kann daher auch ohne Probleme zerstört werden. D kann für die Zeit schlafen, die in FRTS enthalten war (von B aus RTS kopiert).

MAC in Funknetzen

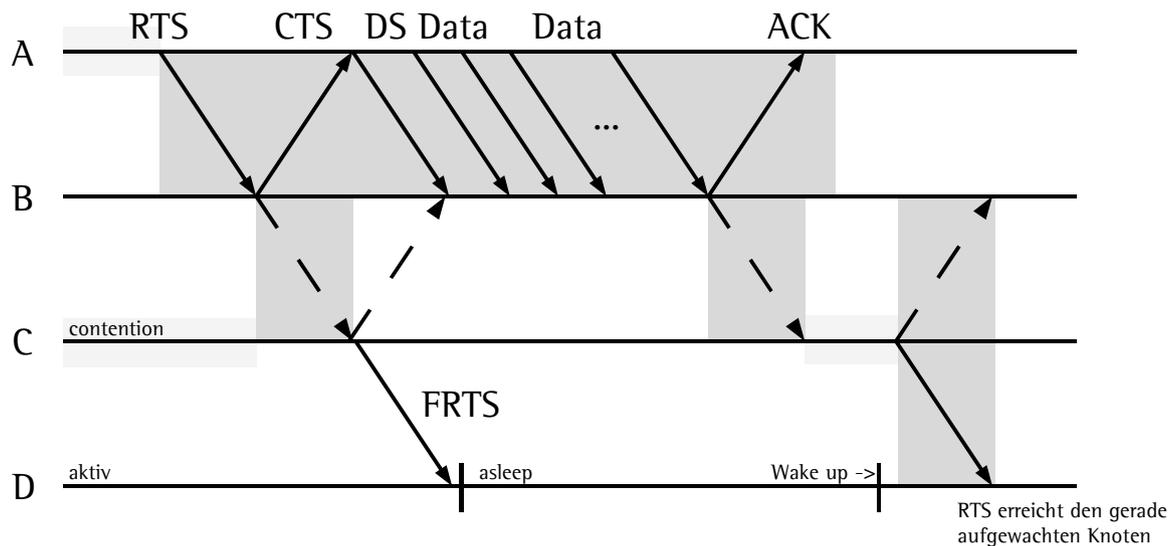
RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS



Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

Verbesserung T-MAC:

Lösung 2 (full-buffer priority): Droht der Puffer eines Knotens überzulaufen, so nimmt er einfach keine Daten mehr an, ignoriert also ein RTS und versucht selbst das Medium zu gewinnen. Welche Probleme könnten auftreten?

Der Ansatz funktioniert nur, wenn an eine Senke gesendet wird. Ansonsten ignorieren sich die Knoten gegenseitig und es kommt zu einer dead-lock Situation.

MAC in Funknetzen

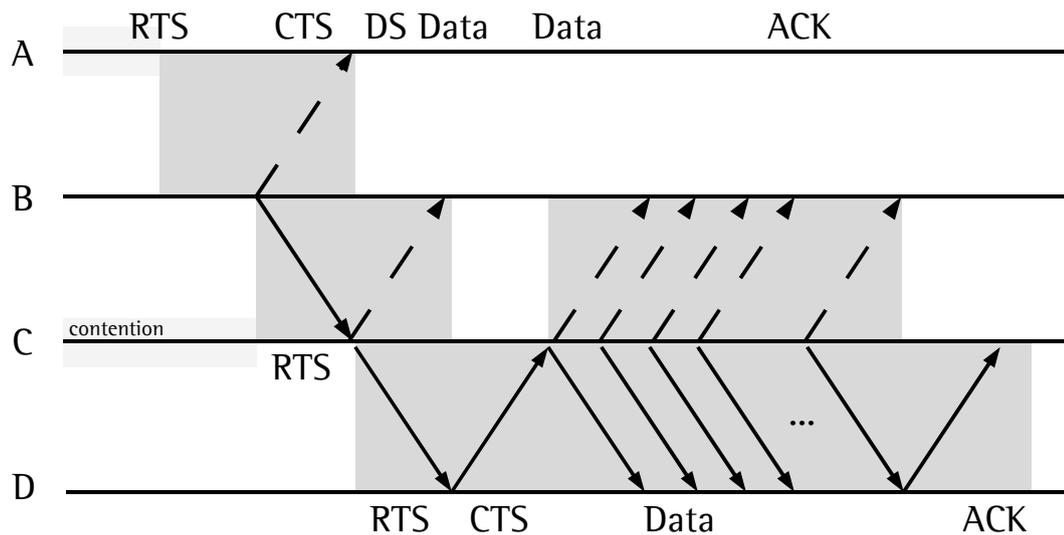
RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS



Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

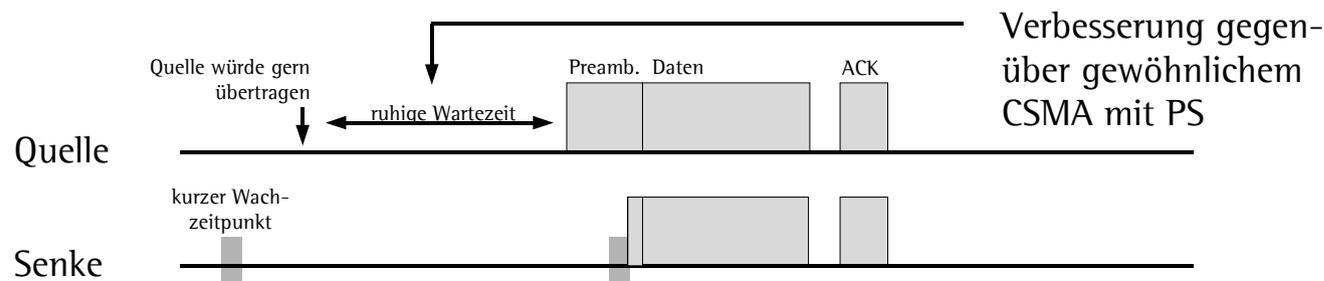
WiseMAC

WiseMAC funktioniert grundsätzlich so, wie CSMA mit Preamble Sampling.

Probleme bei Preamble Sampling

(a) Die lange Vorlaufzeit, die ein Sender senden muss, nur damit der Kanal belegt ist, ohne dass dabei relevante Information übertragen wird.

(b) In der Preamble Phase wird gewissermaßen der Empfänger „aufgesammelt“, um dann zur eigentlichen Datenübertragung zur Verfügung zu stehen. Leider werden während dessen auch alle anderen Empfänger mitgenommen, jedoch ohne Sinn. Diese können zwar den Daten-Frame „verschlafen“, müssen aber wenigstens bis zu einem evtl. RTS bzw. CTS warten. Dadurch geht viel Energie in unbeteiligten Knoten verloren.



MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

WiseMAC

Verbesserung bei WiseMAC: Jeder merkt sich den Wachzeitpunkt seiner Nachbarn, und zwar rel. zur eigenen. Erst kurz vor Erwachen des Empfängers wird der Kanal von einem Sender belegt.

Wann ein Empfänger aufwacht, teilt dieser als Zusatzinformation in seinen ACK Paketen mit. Die ACK Pakete könnten natürlich auch von zufällig wachen aber unbeteiligten Knoten zur Korrektur ihres Wissens über den Wachzeitpunkt des ACK-Versenders verwendet werden. Im Gegensatz zu S- und T-MAC hat bei WiseMAC jeder Knoten einen eigenen, zufälligen Wachzeitpunkt.

Problem der Synchronisation: Haben sich Sender und Empfänger über längere Zeit nicht gehört, sind die Uhren evtl. nicht mehr synchron. Je länger diese Zeit war, desto früher muss die Preamble-Zeit des Senders **vor** dem erwarteten Aufwachen des Empfängers beginnen.

Was ist die Obergrenze für die Preamble-Zeit? Antwort: Die Rahmenlänge.

Problem der Kollisionen: Entschließen sich mehrere Sender einen Datenrahmen an den gleichen Empfänger zu adressieren, so kollidieren die Preamble Rahmen auf jeden Fall. Daher muss die Preamble Länge noch um eine zufällige Zeit verlängert werden. Hier wählt man wieder ein Vielfaches der Umschaltzeit der RF-Unit, ebenso wie bei 802.11. Dadurch gewinnt ein einzelner Sender wahrscheinlich die so entstandene contention-Phase.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

Probleme der MAC-Schicht speziell in Funknetzen:

WiseMAC

Vorteile von WiseMAC: Unbeteiligte Sender wenig belastet, wegen kurzer Preamble
Sender weniger belastet, da nur kurze Preamble.

Nachteile: Wenn nicht mehr alle Knoten automatisch zuhören, geht auch kein Broadcast mehr (Lösung hierfür: Der Sender schaltet auf lange Preamble-Zeiten um).

In jedem Fall wachen unbeteiligte Knoten während des Daten-Frames auf. Leider können dann die Daten nicht mehr so leicht überhört werden, ja deren Länge für zufällig zugeschaltete Knoten nicht erkennbar ist. Im Mittel muss einem Daten-Rahmen, so er von einem unbeteiligten Knoten nach dem Aufwachen erwischt wurde, zu 50% gehört werden.

MAC in Funk-
netzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Die Aufgabe des AMRIS-Protokolls ist es in einem drahtlosen Ad-Hoc Netzwerk einen Multicast-Baum mit vielen Empfängern aufzubauen, die einer bestimmten Sitzung beitreten möchten. Besonders stabil ist das Protokoll bei sich bewegenden, verschwindenden und neu hinzukommenden Knoten.

Im Rahmen der Sensornetze sollen von einer Quelle eher selten große Datenmengen zu den Knoten versandt werden. Jedoch eignet sich das Protokoll auch dafür einen Verteilbaum für Anfragen aufzubauen. Knoten mit unterschiedlichen Ausstattungen können dabei verschiedenen Gruppen beitreten, z. B. je einer pro Sensor-Typ (Temperatur, Licht, Feuchtigkeit etc.).

Der Baum kann zu einer effizienten Verteilung von Anfragen genutzt werden. Insbesondere können über ihn die vor Ort erhobenen Daten auf einem optimalen Weg wieder an die Senke (meist auch die Quelle der Anfrage) zurückgesandt werden.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Aufgabe: Eine Quelle möchte in einem wireless Ad-Hoc Netzwerk Daten an potenziell viele Empfänger schicken (Aufgabe der Schicht 3)

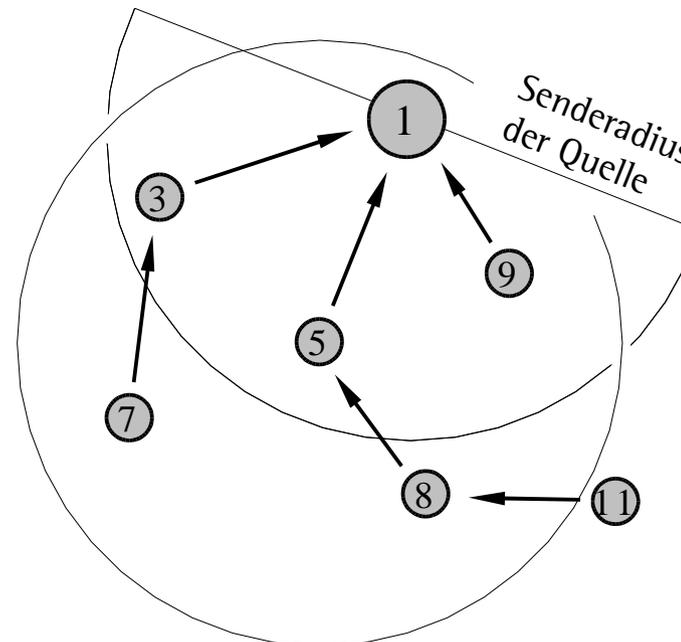
Nebenbedingung: Es soll keine Rolle spielen, ob die darunter liegende Schicht 2 (Sicherheitsschicht) nur Punkt zu Punkt Verbindungen oder auch Broadcast unterstützt

Protokoll läuft in 2 Phasen:

- (1) Initialisierungsphase
- (2) Beitritts-Phase
- (3) Maintenance-Phase

Bemerkung:

Broadcast: Nachrichten an alle
 Multicast: Nachrichten nur an Gruppe



MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Initialisierungsphase

Quelle verschickt Nachrichten der Art: NEW_SESSION(sessionID, msmID, routingMetric)

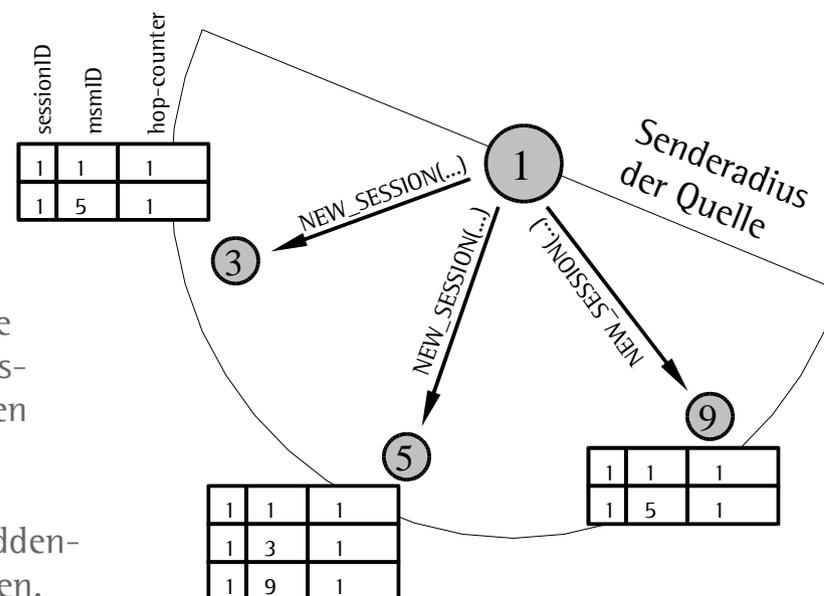
Empfänger warten eine zufällige Zeit. Der Empfänger mit der geringsten Wartezeit wird die NEW_SESSION-Nachricht wiederholen. Er ersetzt die msmID (multicast membership ID) durch einen größeren, jedoch nicht den unmittelbar folgenden Wert.

Die Lücke zwischen den IDs soll den späteren Eintritt weiterer Knoten erleichtern.

Als Routing-Metrik kann z. B. die Anzahl der Hops verwendet werden.

Jeder Teilnehmer merkt sich eine Tabelle mit allen bekannt werdenden Nachbarn. Die Einträge der Tabelle haben auch ein Verfallsdatum, falls ein Nachbar über einen längeren Zeitraum nicht gehört wurde.

Doppelte msmIDs können aufgrund des Hidden-Station Problems nur lokal vermieden werden.



MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Initialisierungsphase

Jeder Knoten ist für die Wahl seines Vaterknoten selbst verantwortlich. Er wählt hierzu einen Knoten mit einer guten Metrik, also z. B. einer kurzen Verbindung zur Quelle.

Nach Ablauf seines Timers verkündet er selbst eine NEW_SESSION-Nachricht mit der bereits erwähnten höheren msmID des Vaters. Natürlich könnten im Verlauf und verspätet noch weitere, bessere Väter auftauchen. Diese werden – ebenso wie alle Knoten – in die Nachbarschaftstabelle aufgenommen, jedoch der Vater nicht mehr gewechselt. Denn sonst würde die zuletzt versendete NEW_SESSION Nachricht nicht mehr stimmen (denn diese soll sich ja gerade nach der Vater-ID richten, plus einem Offset).

Bisher sind noch keine Knoten der Multicast-Session beigetreten. Es wurde lediglich die Nachbarschaftsinformation ausgetauscht.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Beitrittsphase

Ein Knoten X möchte der Multicast-Gruppe beitreten:

Seine Tabelle enthält Nachbarn Y mit größeren und kleineren msmIDs. Da kleinere IDs tendenziell früher vergeben werden als große (jedoch nicht zwingend), haben die Nachbarn von X mit kleineren IDs wahrscheinlich einen kürzeren Weg zur Quelle. X sortiert die Nachbarn aufsteigend nach ihrer ID und sendet einen JOIN_REQ an einen potenziellen Elternknoten Y.

Reaktion des Elternknoten Y:

Y prüft, ob er bereits ein Mitglied der Multicast-Gruppe ist. Wenn ja, sendet er ein JOIN_ACK. Er dadurch wurde Knoten X letztlich der Gruppe zugeordnet und wird später Pakete der Multicast-Session erhalten.

Ist Y nicht Teil der Gruppe, so verfährt er so, als habe er selbst den Wunsch beizutreten. Wurde in der Kette zur Quelle hin ein gültiger Teilnehmer gefunden, so baut sich die Rekursion rückwärts bis zum Knoten X wieder ab und er Zweig des Multicast Baumes wurde etabliert. Wenn sich die Knoten bewegt haben kann es zu Sackgassen kommen, die nicht zum Ziel führen!

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Maintenance-Phase:

Nach der Initialisierung schicken die Knoten regelmäßig sogenannte Beacons (=Signale, Funkfeuer), in der lediglich ihre eigenen Daten enthalten sind (session, memID, hops). Die Nachbarn hören diese Beacons und speichern sie in ihrer Nachbarschaftstabelle.

Wie bereits erwähnt hat jeder Eintrag ein Verfallsdatum. So entfällt die Nachbarschaftsinformation automatisch, wenn ein Knoten ausfällt. Grundsätzlich hat dies erst einmal keine Folgen.

Verbindungsbruch im Multicast-Baum:

Erst wenn ein Knoten die Verbindung zu seinem Vater i. d. Multicast Gruppe verliert, ist er dafür verantwortlich, die Verbindung wiederherzustellen. Dies wird als **Branch Reconstruction** bezeichnet und besteht aus zwei Phasen BR1 und BR2.

BR1:

Wieder werden wie bei der Beitrittsphase die Nachbarn angefragt (JOIN_REQ), ob eine „Vaterschaft“ zur fraglichen Multicast Gruppe in Frage kommt. Auch hier sendet der mögliche Vater im Erfolgsfall ein JOIN_ACK. Sonst muss er selbst in seiner Umgebung nach Verbindungsmöglichkeiten suchen. Knoten die nur vermitteln werden als **Intermediate-Nodes** bezeichnet.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS

Kommunikation in Sensornetzen

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Maintenance-Phase: BR1 (Fortsetzung)

Führt die Rekursion zu keinem Ergebnis werden JOIN_NACK Tokens auf dem umgekehrten Pfad verschickt. Der Initiator der eigentlich beitreten wollte erkennt dies zuletzt und geht, nachdem er einige weitere mögl. Vaterknoten angefragt hat, in Phase BR2 über. BR2 ist der Notnagel für fehlgeschlagene Versuche einen neuen Zeig in den MC-Baum einzufügen oder einen bestehenden zu reparieren.

BR2:

Bisher wurden (zumindest logisch) unicast JOIN_REQ Aufforderungen verschickt. Nun werden die gleichen Aufforderungen als Broadcast Nachrichten generiert. Dies ist unabhängig davon, ob die Schicht 2 dies unterstützt. Wenn nicht, so müssen die Nachrichten für alle Nachbarknoten dupliziert werden.

Ein Knoten Y erhält die Broadcast JOIN_REQ Nachricht. Ist er selbst bereits Teilnehmer der Gruppe, so wird er auf dem umgekehrten Weg ein JOIN_ACK (als Unicast) an den Initiator X des Requests verschicken. Ist diese Nachricht bei X angekommen, so ist damit noch kein Zweig des Baumes hergestellt. Denn es könnten eine Reihe von JOIN_ACK-Antworten eintreffen. In diesem Fall möchte sich X vorbehalten den optimalen Partner (also den kürzesten Weg zur Quelle) zu wählen. Nur auf diesem Weg wird ein JOIN_CONF verschickt, der dann tatsächlich den Zweig im MC-Baum erzeugt. Generell sollte ein Broadcast JOIN_REQ kann mit einem TTL-Counter versehen werden, um nicht das ganze Netz zu belasten.

BR2 kann auch in der Beitrittsphase verwendet werden, wenn der gewöhnliche Weg der Vatersuche fehlgeschlagen ist.

Neue Knoten die noch keine msmlD besitzen, müssen diese vor der Anforderung zum Beitritt aus der Nachbarschaftsinformation (den Beacons) erzeugen.

MAC in Funknetzen

RTS/CTS-Schema

S-MAC

T-MAC

Wise-MAC

AMRIS