

Ermittlung des Netzzustands von Funk-Netzwerken

Christopher Kramer

AG Vernetzte Systeme des Fachbereichs Informatik
Technische Universität Kaiserslautern
67653 Kaiserslautern

Zusammenfassung Für Anwendungen von Funknetzwerken sind Topologie-Informationen von großer Bedeutung. Das im Rahmen dieser Arbeit entworfene und implementierte Protokoll ermittelt selbstständig die Topologie eines Funknetzwerkes inklusive Einmessung aller Links und verteilt die gewonnenen Topologie-Informationen netzweit. Anhand der Implementierung wird das entworfene Protokoll funktional evaluiert.

Keywords: Funknetzwerke, Netzzustand, Topologie-Bestimmung, Netz-Kalibrierung

1 Einleitung

Für Anwendungen von Funknetzwerken sind Topologie-Informationen äußerst hilfreich und teilweise zwingend nötig. Es spielt meist eine Rolle, wie groß das Funknetzwerk ist, welche Knoten überhaupt vorhanden sind und welche Knoten Nachrichten austauschen können oder sich gegenseitig stören. Ist dieses Wissen vorhanden, können beispielsweise Routing- oder Scheduling-Verfahren darauf zurückgreifen, um das Netz optimal zu nutzen.

Da die Topologie eines Funknetzwerkes oft nicht einfach statisch ermittelt und angegeben werden kann, wurde im Rahmen dieses Projekts ein Protokoll entworfen und implementiert, mit dem die Knoten eines Funknetzwerkes ihre Topologie eigenständig bestimmen und untereinander austauschen.

Zunächst wird in Kapitel 2 definiert, welche Netzinformation das Protokoll bestimmen soll. Anschließend legt Kapitel 3 kurz dar, welche Annahmen gelten müssen, damit das Protokoll funktioniert. In Kapitel 4 wird das Protokoll selbst und in Kapitel 5 die Implementierung des Protokolls sowie ein beispielhafter Testlauf beschrieben. Einen Ausblick auf mögliche Verbesserungen gibt Kapitel 6 und Kapitel 7 schließt die Arbeit mit einer Zusammenfassung.

2 Ermittelte Netzzustands-Informationen

Zunächst soll festgelegt werden, welche Informationen über den Netzzustand das Protokoll bestimmen soll.

Die Topologie eines Funknetzes ergibt sich aus den vorhandenen Links zwischen den Knoten. Es wird dabei zwischen drei Linktypen unterschieden (vgl. [6]):

Kommunikations-Links: Es besteht ein Kommunikations-Link von einem Knoten A zu einem Knoten B, wenn B Nachrichten von A (direkt) korrekt empfangen kann, sofern kein anderer Knoten in Interferenzreichweite zeitgleich sendet.

Interferenz-Links: Es besteht ein Interferenz-Link von einem Knoten A zu einem Knoten B, wenn das korrekte Empfangen von Nachrichten durch B gestört wird, wenn A zeitgleich sendet. Ein Kommunikations-Link ist immer auch ein Interferenz-Link.

Sensing-Links: Es besteht ein Sensing-Link von einem Knoten A zu einem Knoten B, wenn B Übertragungsenergie von A erkennen kann. Ein Interferenz-Link ist immer auch ein Sensing-Link.

Da Links in Funknetzwerken nicht zwingend symmetrisch sein müssen [4], werden Links als (potenziell) asymmetrisch angesehen. Es kann also sein, dass von einem Knoten A ein Kommunikations-Link zu einem Knoten B besteht, von Knoten B aber nur ein Interferenz-Link zu Knoten A, beispielsweise weil B einen schwächeren Transceiver besitzt oder schlechteren räumlichen Bedingungen ausgesetzt ist.

Zusätzlich zu der Einstufung der Links in vorhandene Link-Typen soll das Protokoll die Güte von Kommunikations-Links bestimmen. So kann beispielsweise ein Routing-Protokoll diese Information nutzen, um Nachrichten bevorzugt über starke Links zu routen oder die Übertragungsleistung zu verringern. Als Maß für die Linkgüte wird die durchschnittliche Stärke des Empfangssignals (RSS) bestimmt, da diese bei modernen Transceiver-Chipsätzen gut mit der Packet-Delivery-Ratio korreliert[7].

Sind alle Links des Funknetzes bestimmt, lassen sich daraus implizit weitere Informationen über das Netz ableiten:

Vorhandene Knoten: Es werden nur Knoten zu denen Links bestehen als zum Netzwerk angehörig angesehen, daher lässt sich aus den erkannten Links direkt die Menge aller zum Netz gehöriger Knoten ablesen. Mit Hilfe dieser Information lässt sich beispielsweise exklusives Scheduling umsetzen, indem jedem vorhandenen Knoten ein Slot exklusiv zugeordnet wird.

Netzdurchmesser: Der tatsächliche Durchmesser des Funknetzes kann aus den Kommunikations-Links bestimmt werden. Mit diesem Wissen können Protokolle, deren Aufwand vom maximalen Netzdurchmesser abhängt (wie z.B. BBS [3]), effizienter betrieben werden als mit einer konservativen Abschätzung.

Jegliche ermittelte Netzinformation wird unter den Knoten ausgetauscht und ist schlussendlich allen Knoten bekannt.

3 Annahmen

Das im Rahmen dieses Projekts entwickelte Protokoll trifft einige Annahmen über das Funknetz, die für eine korrekte Funktionsweise erfüllt sein müssen.

Single-Network Property: Es muss zwischen allen Knotenpaaren ein zuverlässiger Kommunikationspfad vorhanden sein. Wäre dies nicht der Fall, könnte die ermittelte Netzzustandsinformation nicht unter allen Knoten ausgetauscht werden. Darüber hinaus darf es keine Knoten in Interferenzreichweite geben, welche ein anderes Protokoll auf dem selben Funkkanal verwenden.

Eindeutige Knoten-IDs: Es wird davon ausgegangen, dass jedem Knoten eine eindeutige Knoten-ID zugewiesen ist. Knoten-IDs müssen ganze Zahlen beginnend bei 0 sein und für einen effizienten Betrieb des Protokolls sollten die IDs möglichst nur geringe Lücken aufweisen.

Master-Knoten: Ein (beliebiger) Knoten agiert als Master, alle anderen Knoten als Slaves. Der Masterknoten übernimmt eine gesonderte Rolle bei der Synchronisation und bei der Terminierung des Protokolls.

Maximaler Netzdurchmesser bekannt: Es wird davon ausgegangen, dass eine obere Abschätzung des Netzdurchmessers bekannt ist. Es ist nicht nötig, den exakten Netzdurchmesser zu kennen (dieser lässt sich durch das Protokoll bestimmen). Es wird vorab lediglich eine obere Abschätzung benötigt.

Begrenzte Knoten-Mobilität: Knoten dürfen sich bewegen, während das Protokoll die Topologie bestimmt. Allerdings darf die Topologie nicht andauernden Veränderungen unterliegen, da sonst das Protokoll nicht terminiert.

Tick-Synchronisation mit begrenztem Tick-Offset: Das Protokoll verwendet ein Slotting um exklusive Reservierungen für die einzelnen Knoten festzulegen und benötigt dafür eine Tick-Synchronisation mit begrenztem Tick-Offset.

4 Protokoll zur Bestimmung des Netzzustands

Im Folgenden wird zunächst die grundlegende Funktionsweise des Protokolls erläutert und dann auf Details des Slottings und des Nachrichtenformats, sowie die Einstufung in die Link-Typen eingegangen.

Periodisch bestimmt das Protokoll die Größe des Adressraums der Knoten im Netzwerk (die maximale Anzahl Knoten) $n_{nodesMax}$. Hierzu kommt das Arbitrating and Cooperative Transfer Protocol (ACTP) [1] mit netzweiter Arbitrierung zum Einsatz. Es senden alle Knoten gleichzeitig ihre eigene Knoten-ID plus Eins¹

¹ Es wird angenommen dass Knoten-IDs bei 0 beginnen. Dadurch ist die Größe des Adressraums 1 größer als die maximale ID.

in Black-Burst Codierung (vgl. [2]), wobei die binäre Eins dominant kodiert wird und das höchstwertige Bit (MSB) zuerst übertragen wird. Durch das Protokoll wird somit sichergestellt, dass sich der von dem Knoten mit der höchsten Knoten-ID gesendete Wert netzweit durchsetzt und allen Knoten im Netz bekannt ist.

Nun werden jedem potenziell vorhandenen Knoten, d.h. für alle Knoten-IDs im bestimmten Adressraum, exklusiv Slots reserviert. Die Reservierung wird durch alle Knoten selbstständig berechnet (s. Kapitel 4.1), wozu lediglich die allen Knoten bekannte Größe des Adressraums $n_{nodesMax}$ benötigt wird, d.h. ein Austausch von Reservierungs-Information entfällt. Allerdings entsteht ein Overhead durch ungenutzte Reservierungen, wenn nicht alle Knoten-IDs innerhalb des Adressraums lückenlos benutzt werden.

Sämtliche Reservierungen werden an einem netzweiten Tick ausgerichtet, der durch Black-Burst-Synchronisation (BBS) [3] verfügbar gemacht wird. Das verwendete Master-basierte BBS setzt voraus, dass ein Masterknoten definiert ist und ebenfalls der maximale Netzdurchmesser bekannt ist.

Jeder vorhandene Knoten sendet nun in allen für ihn exklusiv reservierten Slots Nachrichten. Knoten, die diese Nachrichten empfangen, wissen nun, dass ein Link von diesem Knoten zu ihnen besteht. Selbst wenn der Link so schwach ist, dass die Nachricht nicht fehlerfrei empfangen werden kann (wie es bei Sensing-Links der Fall ist), ist dem Empfänger durch die exklusive Zuordnung des Slots dennoch bekannt, von welchem Knoten gesendet wurde. Empfängt ein Knoten nichts, so weiß er damit, dass kein Link zwischen dem Knoten, für den der Slot reserviert ist, und ihm besteht. Die Einstufung der Links in Linktypen erfolgt anhand der beobachteten Empfangsstärke (Details s. Kapitel 4.3).

Der netzweite Austausch der Netzinformation geschieht, indem die von den Knoten gesendeten Nachrichten mit der bisher ermittelten Netzinformation gefüllt werden.

4.1 Slotting

Das Protokoll verwendet ein dreistufiges Slotting, um die exklusiven Reservierungen effizient umzusetzen. Dieses Slotting ist beispielhaft in Abb. 1 grafisch illustriert.

Super-Slots bilden die oberste Ebene des Slottings. Sie beginnen mit einem Super-Tick, wobei es sich um einen BBS-Tick handelt, dessen Rahmen um ein spezielles Super-Tick-Flag erweitert ist. Ein Super-Slot besteht aus mehreren Makro-Slots, wobei konfiguriert werden kann, wie viele Makro-Slots einen Super-Slot bilden sollen.

Makro-Slots werden durch Makro-Ticks abgegrenzt, wobei es sich um BBS-Ticks handelt, bei denen das Super-Tick Flag entweder 0 oder 1 ist (d.h. Super-Ticks sind auch Makro-Ticks). Der erste Makro-Slot in einem Super-Slot beginnt mit

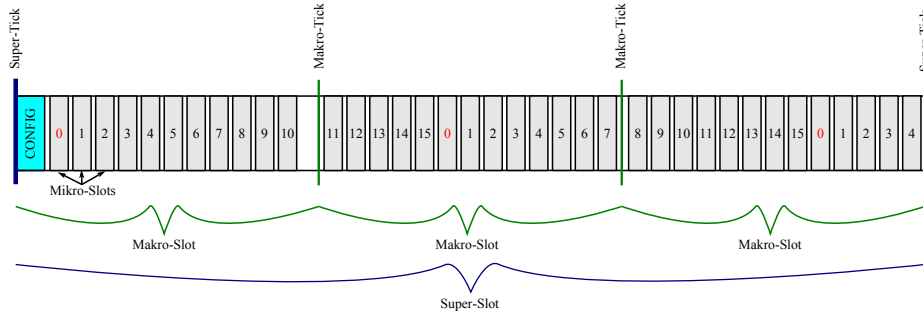


Abbildung 1. Slotting mit 3 Makro-Slots pro Super-Slot und $n_{nodesMax} = 16$ und den sich daraus ergebenden Slotzuordnungen.

der netzweiten Übertragung der Konfiguration per ACTP. Die Konfiguration umfasst in erster Linie die Größe des Adressraums, sowie Flags zur Terminierung (Details dazu in Kapitel 4.2). Dadurch, dass die Konfiguration per ACTP allen Knoten bekannt gemacht wird, können alle Knoten die exklusiven Reservierungen selbstständig bestimmen. Hierdurch wird ebenfalls ermöglicht, dass — während das Protokoll läuft — ein neuer Knoten hinzu kommt, dessen Knoten-ID außerhalb des aktuellen Adressraums liegt, und sich dadurch die exklusiven Reservierungen verändern.

Mikro-Slots werden durch einen am vorherigen Makro-Tick ausgerichteten Timer ausgelöst. Sämtliche Mikro-Slots innerhalb eines Super-Slots werden fortlaufend durchnummeriert, d.h. die Zählung läuft über Makro-Slot-Grenzen hinaus. Jeder Mikro-Slot ist exklusiv für je einen Knoten k reserviert. Die Reservierung ergibt sich aus der Mikro-Slot-Nummer n_{slot} und der maximalen Knoten-Anzahl $n_{nodesMax}$ nach Gl. (4.1) über die Modulo-Funktion.

$$k = n_{slot} \bmod n_{nodesMax} \quad (4.1)$$

Durch die netzweite Synchronisierung kann jeder Knoten die Mikro-Slot-Nummer n_{slot} abzählen und zusammen mit der maximalen Knoten-Anzahl $n_{nodesMax}$, welche netzweit übertragen wurde, kann er lokal zu jedem Mikro-Slot den Knoten k bestimmen, für den dieser exklusiv reserviert ist. Der Knoten, für den der Mikro-Slot exklusiv reserviert ist, sendet eine MEASURE-Nachricht konstanter Länge. Diese Nachricht ist kürzer als der Mikro-Slot. Anhand der Nachrichtenlänge können Knoten, welche zu weit entfernt sind, um die Nachricht korrekt zu empfangen, einen Sensing-Link erkennen. Knoten, die die Nachricht korrekt empfangen, können daran erkennen, dass ein Kommunikations- oder Interferenz-Link besteht. Darüber hinaus verteilen die Knoten ihr bisher gesammeltes Wissen über MEASURE-Nachrichten an andere Knoten. Details dazu werden in Kapitel 4.2 erläutert.

Die Verwendung von Super-Slots bringt einige Vorteile mit sich. Die Dauer eines

Makro-Slots kann nicht beliebig vergrößert werden, da eine regelmäßige BBS-Synchronisation wichtig ist, damit der Offset zwischen den Knoten nicht zu groß wird. Wäre der Adressraum nun größer als Mikro-Slots in einen Makro-Slot passen, müsste die Länge des Makro-Slots erhöht werden, um für alle Knoten Slots reservieren zu können. Dadurch, dass die Mikro-Slot-Zählung in einem Super-Slot über Makro-Slot-Grenzen hinaus läuft, ist es möglich, trotz kleinem Offset immer genügend Mikro-Slots unterbringen zu können, um für jeden Knoten mindestens einen reservieren zu können.

Des Weiteren verringert das Zusammenfassen von Makro-Slots zu Super-Slots den Verschnitt und sorgt so für eine effizientere Ausnutzung der Bandbreite. Diese wird ebenfalls dadurch verbessert, dass die relativ aufwendige Übertragung der Konfiguration nicht in jedem Makro-Slot erfolgen muss. Trotzdem kann sie regelmäßig erfolgen und es so neuen Knoten ermöglichen, dem Netz beizutreten. Je nach Größe der Makro- und Super-Slots sind für jeden einzelnen Knoten i.d.R. mehrere Sendeslots innerhalb eines Super-Slots reserviert (vgl. Abb. 1).

4.2 Nachrichtenformate

Das Protokoll verwendet zwei unterschiedliche Nachrichtentypen, die in diesem Kapitel näher beschrieben werden.

CONFIG-Nachrichten werden am Anfang eines jeden Super-Slots per ACTP mit arbiträrer Kodierung netzweit übertragen. Sie enthalten folgende Information:

- *Maximale Knoten-Anzahl* $n_{nodesMax}$ (8 Bit)
Aus der maximalen Knoten-Anzahl ergeben sich die exklusiven Reservierungen (s. Kapitel 4.1). Aus der gewählten Länge von 8 Bit ergibt sich, dass der Adressraum maximal 256 erreichen darf.
- *OnDuty-Flag*
Mit dem OnDuty-Flag wird angegeben, ob gerade das hier beschriebene Protokoll zu Netzeinmessung läuft (OnDuty=0) oder das eigentliche Anwendungsprotokoll (OnDuty=1).
- *NoTerminate-Flag* (nur wenn OnDuty=0)
Die Terminierung erfolgt, sobald alle Knoten sich einig sind, dass sie ihre Links hinreichend stabil ausgemessen haben. Die Knoten senden das NoTerminate-Flag so lange dominant 1, bis sie der Terminierung zustimmen. Nur wenn ein Knoten seine Links für die letzten $n_{stableForTerm}$ Super-Slots stabil eingestuft hat, setzt er das NoTerminateFlag 0. Erst wenn alle Knoten das Flag 0 setzen, wird terminiert und im nächsten Super-Slot das Anwendungsprotokoll gestartet (OnDuty=1).
- *MasterTerminateFlag* (nur wenn OnDuty=0)
Der Master hat das Recht, die Terminierung durch das Setzen des MasterTerminate-Flags zu erzwingen, auch wenn das NoTerminate-Flag noch durch einen Knoten gesetzt wird. Dies verhindert, dass nie terminiert werden kann, weil stetig Änderungen am Netz wahrgenommen werden. Je

nachdem, wie der Master konfiguriert ist, startet er in diesem Fall trotzdem die Anwendung (indem er im nächsten Super-Slot OnDuty=1 setzt) oder meldet einen Fehler (z.B. blinkende Fehler-LED). Zu beachten ist, dass nicht davon ausgegangen werden kann, dass alle Knoten ein einheitliches Bild der Topologie haben, falls der Master die Terminierung erzwungen hat.

MEASURE-Nachrichten werden als reguläre Rahmen per lokalem Broadcast übertragen. Sie erfüllen zwei Aufgaben: Zum einen erhalten alle Knoten in Reichweite allein dadurch, dass sie die Nachricht empfangen, die Information, dass ein Link besteht, und können anhand der Empfangsstärke den Linktyp einschätzen. Zum anderen enthält die Nachricht die Topologie-Information, die der sendende Knoten bisher entweder selbst ermittelt oder von anderen Knoten mitgeteilt bekommen hat.

Der Inhalt der MEASURE-Nachricht setzt sich wie folgt zusammen:

- *Knoten-ID des Senders*
Da die Sender-ID dem Empfänger durch die exklusiven Reservierungen bereits implizit bekannt ist, dient die Übertragung der Knoten-ID des Senders in der Nachricht lediglich der Redundanz.
- *Makro-Slot-Nummer*
Auch die Makro-Slot-Nummer innerhalb des aktuellen Super-Slots kann jeder Knoten durch die Synchronisation selbst bestimmen, die Übertragung innerhalb der Nachricht dient lediglich der Redundanz.
- *Bereits erkannte Links*
Die bereits selbst erkannten und von anderen Knoten mitgeteilten Links bilden den eigentlichen Inhalt der Nachricht. Es werden in jeder MEASURE-Nachricht Informationen zu 15 Links übertragen. Falls mehr Links zu übertragen sind, erfolgt eine Fragmentierung.
Für jeden einzelnen Link werden folgende Informationen übermittelt:
 - *Sender Knoten-ID und Empfänger Knoten-ID*
Links werden nicht als symmetrisch angenommen, daher kann sich der Link von Knoten 1 zu Knoten 2 vom Link in entgegengesetzter Richtung unterscheiden.
 - *Link-Typ*
Es werden vier unterschiedliche Link-Typen unterschieden: Kein Link (Typ 0), Sensing-Link (Typ 1), Interferenz-Link (Typ 2) und Kommunikations-Link (Typ 3).
 - *Sequenz-Nummer*
Knoten ermitteln zum einen selbst Informationen zu Links, erhalten aber auch Informationen über Links von Nachbarn. Erhält ein Knoten von mehreren Nachbarn widersprüchliche Informationen über einen Link so muss er entscheiden können, welche Information aktueller ist. Dazu vergibt der Knoten, der den Link ermittelt hat, d.h. der Zielknoten eines Links, eine Sequenznummer, die er immer dann erhöht, wenn sich die Link-Information ändert. Knoten, die Link-Informationen erhalten,

ignorieren diese, falls deren Sequenz-Nummer kleiner ist als die Sequenz-Nummer der ihnen bereits bekannten Link-Information.

- *Link-Signalstärke*

Falls der Link ein Kommunikations-Link ist, wird die mittlere Signalstärke, die auf diesem Link gemessen wurde, angefügt. Dies ermöglicht es der Anwendung, z.B. stärkere Links bevorzugt zu verwenden (z.B. für Routing).

Da MEASURE-Nachrichten als reguläre Rahmen per 802.15.4 versendet werden, enthalten sie auch implizit eine CRC16-Prüfsumme, die es erlaubt, fehlerhaft erhaltene Rahmen zu erkennen.

4.3 Einstufung in Link-Typen

Wie in Kapitel 2 beschrieben, werden Links in unterschiedliche Link-Typen eingeteilt. Die Einstufung erfolgt grundsätzlich anhand der empfangenen Nachrichten, der Empfangsstärke beim Empfang sowie im Falle von Sensing-Links anhand der empfangenen Energie sowie deren Dauer.

Wird eine Nachricht mit einer Empfangsstärke von -85 dBm oder besser empfangen, so wird der Link als Kommunikations-Link eingestuft. Wird eine Nachricht mit einer geringeren Empfangsstärke korrekt empfangen, so wird der Link trotz des korrekten Empfangs der Einzelnachricht als nicht zuverlässig und daher pessimistisch als Interferenz-Link eingestuft. In einer Untersuchung zur Korrelation von Empfangsstärke und Packet Receive Rate (PRR) mit dem auch bei unserer Implementierung (Details s. Kapitel 5) eingesetzten CC2420-Transceiver wurde gezeigt, dass bei Empfangsstärken ab -87 dBm eine PRR von mindestens 85% zu beobachten ist [7]. Basierend auf dieser Erkenntnis stufen wir einen Link mit mindestens -85 dBm Empfangsstärke als zuverlässigen Kommunikations-Link und alles, was darunter liegt, als Interferenz-Link ein.

Besteht ein Sensing-Link von einem Knoten zu einem anderen, so bemerkt die Empfängerseite gar keinen Nachrichtenempfang. Es kann also nicht die Empfangsstärke beim Nachrichtenempfang zur Einstufung des Links genutzt werden. Allerdings bemerkt die Empfängerseite auf einem Sensing-Link die Energie auf dem Medium. Durch die exklusiven Reservierungen ist klar, welcher Knoten gerade sendet. Stimmt die Zeit der Mediumbelegung unter Berücksichtigung der CCA-Verzögerung mit der Übertragungszeit einer MEASURE-Nachricht überein, so wird von einem Sensing-Link ausgegangen.

Wird weder eine Nachricht noch Energie auf dem Medium festgestellt, so geht der Knoten davon aus, dass überhaupt kein Link von dem Knoten, für den der Slot reserviert ist, besteht.

Um zu vermeiden, dass kurzfristige Änderungen der Link-Qualität dazu führen, dass sich keine stabile Sicht des Netzwerks durchsetzt und das Protokoll niemals stabil wird, wird neben dem einzelnen Nachrichtenempfang auch die bisherige Einstufung des Links berücksichtigt. Hierbei wird ein pessimistischer Ansatz verfolgt, der Links im Zweifel eher schlechter als besser einstuft.

Initial werden alle Links als Interferenz-Links angenommen. Aus Sicht der Anwendung ist ein Interferenz-Link der schlechteste Fall, er ermöglicht weder zuverlässige Kommunikation noch dass einer der Knoten sendet, während der andere von einem anderen Knoten etwas empfängt (also kein spatial reuse).

Wird ein Link nun neu eingestuft, d.h. die neue Einstufung $LinkType_{measured}$ weicht von der vorherigen Einstufung $LinkType_{current}$ ab, so hängt es nun auch von der vorherigen Einstufung ab, was geschieht.

Handelt es sich bei der Änderung um „gute Nachrichten“, also eine für die Anwendung nützlichere Situation, so wird zunächst abgewartet, ob die positive Änderung stabil bleibt. Auf diese Art werden „gute Nachrichten“ langsam propagiert. Handelt es sich dagegen um eine Verschlechterung der Situation, bricht beispielsweise ein Kommunikations-Link, so sollen diese „schlechten Nachrichten“ schneller propagiert werden.

Im Folgenden werden die einzelnen Übergänge zwischen den Link-Typen betrachtet und beschrieben, wie das Protokoll darauf reagieren soll.

Typ 2 \rightarrow *Typ 3*: Empfängt ein Knoten k_1 eine fehlerfreie Nachricht mit hoher Signalstärke von Knoten k_2 und stuft ihn neu als Kommunikations-Link ein ($LinkType_{measured}(k_2, k_1) = 3$), so sind das „gute Nachrichten“. Bevor diese propagiert werden, muss der Link noch für weitere $n_{stableSlots}$ Slots² als Kommunikations-Link beobachtet werden. Erst dann wird die neue Einstufung $LinkType_{current}(k_2, k_1) = 3$ propagiert.

Typ 2 \rightarrow *Typ 0*: Empfängt ein Knoten k_1 gar keine Nachricht von k_2 , so stuft er ihn mit $LinkType_{measured}(k_2, k_1) = 0$ ein. Hat er ihn bisher als Interferenz-Link eingestuft ($LinkType_{current}(k_2, k_1) = 2$), so sind das „gute Nachrichten“, denn ein Interferenz-Link ist wie bereits erklärt für die Anwendung der schlechteste Fall. Da „gute Nachrichten“ langsam propagiert werden, muss k_1 zunächst $n_{stableSlots}$ Slots warten, bevor er die Neueinstufung propagieren darf.

Typ 2 \rightarrow *Typ 1*: Dieser Fall ist äquivalent zum vorherigen Fall, d.h. es handelt sich um „gute Nachrichten“, die langsam propagiert werden.

Typ 3 \rightarrow *Typ 0*: Bricht ein vorhandener Kommunikations-Link von k_2 zu k_1 ($LinkType_{current}(k_2, k_1) = 3$) ab, d.h. k_1 empfängt keine Nachricht von k_2 ($LinkType_{measured}(k_2, k_1) = 0$), so sind dies „schlechte Nachrichten“. Es wird pessimistisch zunächst der schlechteste Fall angenommen, nämlich dass es sich um einen Interferenz-Link handelt ($LinkType_{current}(k_2, k_1) = 2$). Dies wird propagiert, sobald der Link-Typ für $n_{instableSlots}$ Slots vom bisher propagierten Typ (d.h. $LinkType_{current}(k_2, k_1) = 3$) abweicht. Dabei gilt $n_{instableSlots} < n_{stableSlots}$, denn „schlechte Nachrichten“ sollen schneller als „gute Nachrichten“ propagiert werden.

Sobald der Link für $n_{stableSlots}$ Slots als „kein Link“ eingestuft wurde, wird diese Information dann (als „gute Nachrichten“) propagiert ($LinkType_{current}(k_2, k_1) = 0$).

² Es werden $n_{stableSlots}$ Mikroslots, die k_2 zum Senden zugeordnet, sind abgewartet.

Typ 3 \rightarrow *Typ 2* oder *Typ 3* \rightarrow *Typ 1*: Diese Fälle sind äquivalent zum vorherigen Fall: Das Brechen eines Kommunikations-Links sind immer „schlechte Nachrichten“ und daher schneller zu propagieren.

Typ 0 \rightarrow *Typ 3*: Auf einem Link, der bisher als „kein Link“ eingestuft wurde ($LinkType_{current}(k_2, k_1) = 0$), wird eine Nachricht empfangen ($LinkType_{measured}(k_2, k_1) = 3$). Da der hinzugekommene Link zu Interferenz führen kann, sind dies zunächst „schlechte Nachrichten“, die schnell propagiert werden müssen. Wie beim Linkabbruch wird zunächst pessimistisch der schlechteste Fall angenommen, nämlich dass es sich um einen Interferenz-Link handelt. Dies wird nach $n_{instableSlots}$ Slots propagiert ($LinkType_{current}(k_2, k_1) = 2$). Erst sobald sich der Link für $n_{stableSlots}$ Slots als stabil herausgestellt hat, wird $LinkType_{current}(k_2, k_1) = 3$ propagiert.

Typ 0 \rightarrow *Typ 2*: Auch hierbei handelt es sich um „schlechte Nachrichten“, die nach $n_{instableSlots}$ Slot mit $LinkType_{current}(k_2, k_1) = 2$ propagiert werden.

Typ 1 \rightarrow *Typ 2* oder *Typ 1* \rightarrow *Typ 3*: Diese Fälle sind äquivalent zu den beiden vorherigen auch „schlechte Nachrichten“.

5 Implementierung und Tests

Eine erste Implementierung des hier beschriebenen Protokolls wurde ebenfalls im Rahmen dieses Projektes umgesetzt. Zum Einsatz kamen dabei Knoten vom Typ Imote2 [5], die mit einem CC2420 Transceiver [8] ausgestattet sind. Außerdem kam das auf dieser Plattform bereits integrierte BBS-Protokoll zur Ticksynchronisation zum Einsatz. Da die Implementierung des ACTP-Protokolls für diese Plattform zum Zeitpunkt dieser Arbeit noch nicht fertiggestellt war, verzichtet die bisher umgesetzte Implementierung auf den Versand der CONFIG-Nachrichten und setzt voraus, dass allen Knoten a priori die Größe des Adressraums bekannt ist. Zur Terminierung wurden die MEASURE-Nachrichten um ein Terminate-Flag erweitert: Sobald alle Knoten in Reichweite das Flag entsprechend setzen, wird terminiert. Da MEASURE-Nachrichten nicht netzweit übertragen werden, kann dies in großen Topologien allerdings dazu führen, dass nur einige Teile des Netzes terminieren. Sobald die Implementierung von ACTP abgeschlossen ist, soll diese den aktuellen Terminierungsmechanismus ersetzen.

Zum Testen bietet die Implementierung an, aktuelle Statusinformationen über die UART-Schnittstelle an einen PC auszugeben, über den so der aktuelle Status live eingesehen werden kann. Dazu wird am Anfang eines jeden Super-slots eine Link-Matrix ausgegeben, in der der Linkstatus aller Links abgelesen werden kann. Darüber hinaus werden Statusänderungen live ausgegeben, etwa wenn ein neuer Link erkannt wird, ein Link bricht, instabil wird oder neue Linkinformation von einem Nachbarn erhalten wurde. Darüber hinaus kann

über die LEDs das Scheduling beobachtet werden, indem je nach Makro-Slot andere LEDs leuchten und beim Senden einer Nachricht immer eine LED aufblitzt. Die Terminierung wird durch das gleichzeitige Aufleuchten aller LEDs signalisiert.

Die Implementierung wurde anhand unterschiedlicher Testfälle getestet. Dabei kamen bisher nur recht kleine Netze mit bis zu fünf Knoten und einem Netzdurchmesser von bis zu zwei Hops zum Einsatz. Im Folgenden wird einer dieser Tests anhand der vom Protokoll ausgegebenen Status-Informationen illustriert.

In diesem Test wird das Protokoll auf fünf Knoten in Single-Hop-Reichweite gestartet, wobei der Knoten mit der ID 3 als Master und die Knoten mit den IDs 7, 8, 14 und 15 als Slaves gestartet werden. Es wird eine Super-Slot-Länge von 3 Makro-Slots verwendet, eine Makro-Slot-Länge von 2 s und eine Mikro-Slot-Länge von 11 ms. Links müssen für $n_{stableSlots} = 30$ Slots stabil und für $n_{instableSlots} = 10$ Slots instabil bleiben, bevor der neue Zustand propagiert wird (s. Kapitel 4.3). Das Protokoll soll terminieren, nachdem für $n_{stableForTerm} = 3$ Super-Slots bei keinem Knoten eine Änderung mehr beobachtet wurde. Mit $n_{nodesMax} = 20$ sind maximal 20 Knoten mit den IDs 0 bis 19 erlaubt³.

Wir betrachten den Protokollablauf vom Masterknoten mit der ID 3⁴:

```

TEX Master node 3 ready
<<< 00 |01 |02 |03 |04 |05 |06 |07 |08 |09 |10 |11 |12 |13 |14 |15 |16 |17 |18 |19 |
vv -----
00 | | | | | | | | | | | | | | | | | | | | | |
01 | | | | | | | | | | | | | | | | | | | | | |
02 | | | | | | | | | | | | | | | | | | | | | |
03 | | | | 3 | | | | | | | | | | | | | | | | | |
04 | | | | | | | | | | | | | | | | | | | | | |
05 | | | | | | | | | | | | | | | | | | | | | |
06 | | | | | | | | | | | | | | | | | | | | | |
07 | | | | | | | | | | | | | | | | | | | | | |
08 | | | | | | | | | | | | | | | | | | | | | |
09 | | | | | | | | | | | | | | | | | | | | | |
10 | | | | | | | | | | | | | | | | | | | | | |
11 | | | | | | | | | | | | | | | | | | | | | |
12 | | | | | | | | | | | | | | | | | | | | | |
13 | | | | | | | | | | | | | | | | | | | | | |
14 | | | | | | | | | | | | | | | | | | | | | |
15 | | | | | | | | | | | | | | | | | | | | | |
16 | | | | | | | | | | | | | | | | | | | | | |
17 | | | | | | | | | | | | | | | | | | | | | |
18 | | | | | | | | | | | | | | | | | | | | | |
19 | | | | | | | | | | | | | | | | | | | | | |

```

Nach der Startmeldung gibt der Knoten die initiale Linkmatrix aus. Hier sind alle Links noch als Interferenz-Links (Link-Typ 2) eingestuft, da dies als worst case initial angenommen wird (vgl. Kapitel 4.3). Die einzige Ausnahme ist der

³ Würde das Protokoll selbst über CONFIG-Nachrichten $n_{nodesMax}$ bestimmen, so hätte es $n_{nodesMax} = 16$ verwendet (höchste Knoten-ID ist 15 plus Eins).

⁴ In allen Link-Matrizen wurde um die Übersichtlichkeit zu bewahren der Link-Typ 2 entfernt, d.h. leere Zellen bedeuten, dass der Link-Typ 2 ist.

Link von Knoten 3 zu sich selbst. Da jeder Knoten immer mit sich selbst kommunizieren kann, ist der Selbstlink von Anfang an immer vorhanden. Während der Arbarbeitung finden folgende Neubewertungen statt:

```

1: BROKEN 0->3 (own link)
2: BROKEN 1->3 (own link)
3: BROKEN 2->3 (own link)
5: BROKEN 4->3 (own link)
6: BROKEN 5->3 (own link)
7: BROKEN 6->3 (own link)
7: NEW 7-> 7 (received from 7), Type=3, rssi=0
7: NEW 7-> 3 (received myself), rssi=-37
8: NEW 8-> 8 (received from 8), Type=3, rssi=0
8: NEW 8-> 3 (received myself), rssi=-47
10: BROKEN 9->3 (own link)
11: BROKEN 10->3 (own link)
12: BROKEN 11->3 (own link)
13: BROKEN 12->3 (own link)
14: BROKEN 13->3 (own link)
14: NEW 14-> 14 (received from 14), Type=3, rssi=0
14: NEW 14-> 3 (received myself), rssi=-50
15: NEW 15-> 15 (received from 15), Type=3, rssi=0
15: NEW 15-> 3 (received myself), rssi=-47
17: BROKEN 16->3 (own link)
18: BROKEN 17->3 (own link)
19: BROKEN 18->3 (own link)
20: BROKEN 19->3 (own link)

```

Hier sehen wir die ersten festgestellten Änderungen. In den ersten 7 Mikro-Slots (Mikro-Slot 0 bis 6) werden keine Nachrichten empfangen, da es keine Knoten mit den entsprechenden IDs gibt. Der bisher angenommene Interferenz-Link ist also gebrochen, was jeweils im darauffolgenden Slot (Slot 1 bis 7) festgestellt wird. Im Mikro-Slot 7 wird eine Nachricht von Knoten 7 empfangen, welche die neue Information enthält, dass es einen Kommunikations-Link (Typ 3) von 7 zu 7 gibt (Selbstlink). Da eine Nachricht fehlerfrei mit einer guten Signalstärke von -37 dBm empfangen wurde, wird außerdem davon ausgegangen, dass es einen neuen Kommunikations-Link von 7 zum Knoten 3 gibt. Noch ist die Information über diesen Link aber instabil und wird daher noch nicht propagiert. In den Mikro-Slots 8 bis 20 werden dort, wo Knoten vorhanden sind, Nachrichten empfangen und neue Links angenommen. Dort, wo keine vorhanden sind, wird der angenommene Interferenz-Link als gebrochen wahrgenommen.

```

187: STOP INSTABLE 7-> 3 - (received myself)
188: STOP INSTABLE 8-> 3 - (received myself)
194: STOP INSTABLE 14-> 3 - (received myself)
195: STOP INSTABLE 15-> 3 - (received myself)

```

Die ursprünglich als instabil angenommenen Links werden nun langsam als stabil angenommen, nachdem jeder Link $n_{instableSlots} = 30$ mal unverändert wahrgenommen wurde, was entsprechend propagiert wird.

```

<<< 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
vv -----
00 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
01 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
02 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
03 | 0*| 0*| 0*| 3 | 0*| 0*| 0*| 3*| 3*| 0*| 0*| 0*| 0*| 0*| 3*| 3*| 0*| 0*| 0*| 0*|
04 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
05 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
06 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
07 |   |   |   |   |   |   |   | 3 |   |   |   |   |   |   |   |   |   |   |   |   |
08 |   |   |   |   |   |   |   |   | 3 |   |   |   |   |   |   |   |   |   |   |   |
09 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
10 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

```

```

11 | | | | | | | | | | | | | | | | | | | | | |
12 | | | | | | | | | | | | | | | | | | | | | |
13 | | | | | | | | | | | | | | | | | | | | | |
14 | | | | | | | | | | | | | | | | 3 | | | | | |
15 | | | | | | | | | | | | | | | | | 3 | | | | |
16 | | | | | | | | | | | | | | | | | | | | | |
17 | | | | | | | | | | | | | | | | | | | | | |
18 | | | | | | | | | | | | | | | | | | | | | |
19 | | | | | | | | | | | | | | | | | | | | | |

```

In der neuen Linkmatrix zum Beginn des zweiten Super-Slots sind in der Zeile des Knotens 3 selbst jetzt bereits die festgestellten Änderungen eingetragen: Kommunikations-Links von den Knoten 7, 8, 14 und 15 sowie „kein Link“ zu allen nicht vorhandenen Knoten. Alle diese Links sind noch dabei, stabil zu werden, d.h. sie wurden noch nicht so propagiert, wie sie in der Link-Matrix stehen (sondern als Interferenz-Link). Dies lässt sich an der Markierung mit einem Stern ablesen. In dieser Phase werden auch die Empfangsstärken der ankommenden Nachrichten gesammelt, um die mittlere Empfangsstärke des Links zu bestimmen.

```

247: NEW 3-> 7 (received from 7), Type=3, rssi=-31
247: STABLE 7-> 3 - STRENGTH -36(received myself)
248: NEW 3-> 8 (received from 8), Type=3, rssi=-48
248: NEW 7-> 8 (received from 8), Type=3, rssi=-33
248: STABLE 8-> 3 - STRENGTH -49(received myself)
254: NEW 3-> 14 (received from 14), Type=3, rssi=-51
254: NEW 7-> 14 (received from 14), Type=3, rssi=-52
254: NEW 8-> 14 (received from 14), Type=3, rssi=-26
254: STABLE 14-> 3 - STRENGTH -51(received myself)
255: NEW 3-> 15 (received from 15), Type=3, rssi=-52
255: NEW 7-> 15 (received from 15), Type=3, rssi=-53
255: NEW 8-> 15 (received from 15), Type=3, rssi=-56
255: NEW 14-> 15 (received from 15), Type=3, rssi=-33
255: STABLE 15-> 3 - STRENGTH -47(received myself)

```

Nach $n_{stableSlots}$ Slots werden die Links nun stabil. Zuerst wird der Link von 3 zu 7 von Knoten 7 als stabil eingestuft und daher als Kommunikations-Link (Typ 3) mit einer sehr guten Empfangsstärke von -31 dBm propagiert. Dann wird der entgegengesetzte Link von 7 zu 3 an Knoten 3 selbst stabil, hier wurden durchschnittlich -36 dBm gemessen. Das gleiche passiert nun auch mit den anderen Links.

```

267: NEW 0-> 7 (received from 7), Type=0, rssi=0
267: NEW 1-> 7 (received from 7), Type=0, rssi=0
267: NEW 2-> 7 (received from 7), Type=0, rssi=0
267: NEW 4-> 7 (received from 7), Type=0, rssi=0
267: NEW 5-> 7 (received from 7), Type=0, rssi=0
267: NEW 6-> 7 (received from 7), Type=0, rssi=0

```

Hier werden nun an Knoten 7 die Links von den Knoten mit den IDs 0, 1, 2, 4, 5, 6 stabil als „kein Link“ (Typ 0) eingestuft und propagiert. In diesem Super-Slot werden nun alle weiteren stabil werdenden Links ausgetauscht. Da hier nichts Neues passiert, betrachten wir gleich die resultierende Link-Matrix:

```

<<< 00 |01 |02 |03 |04 |05 |06 |07 |08 |09 |10 |11 |12 |13 |14 |15 |16 |17 |18 |19 |
vv -----
00 | | | | | | | | | | | | | | | | | | | | | |
01 | | | | | | | | | | | | | | | | | | | | | |
02 | | | | | | | | | | | | | | | | | | | | | |
03 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 0 |
04 | | | | | | | | | | | | | | | | | | | | | |
05 | | | | | | | | | | | | | | | | | | | | | |
06 | | | | | | | | | | | | | | | | | | | | | |
07 | 0 | 0 | 0 | 3 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 0 | 3 | 3 | 0 | 0 | 0 | 0 |

```

08		0		0		0		3		0		0		0		3		3		0		0		0		0		0		3		3		0		0		0		0		
09																																										
10																																										
11																																										
12																																										
13																																										
14		0		0		0		3		0		0		0		3		3		0		0		0		0		0		3		3		0		0		0		0		
15		0		0		0		3		0		0		0		3		3		0		0		0		0		0		3		3		0		0		0		0		
16																																										
17																																										
18																																										
19																																										

Hier ist bereits die Topologie vollständig korrekt erfasst: Es gibt symmetrische Kommunikations-Links zwischen allen Knoten 3, 7, 8, 14 und 15. Alle Links, die von nicht vorhandenen Knoten ausgehen, stehen noch auf dem Ausgangswert Interferenz-Link (Typ 2). Nachdem sich die Topologie für $n_{stableForTerm} = 3$ Super-Slots nicht verändert, meldet Knoten 3:

OK TO TERM

Dies bedeutet, dass er ab jetzt in den MEASURE-Nachrichten das Flag setzt, was signalisiert, dass er der Terminierung zustimmt. Da alle anderen Knoten ebenfalls keine Topologie-Änderungen feststellen und daher auch der Terminierung zustimmen, endet das Protokoll nun nach 6 Super-Slots (also 36 Sekunden) mit folgender Meldung:

TERMINATE BY AGREEMENT

6 Ausblick

Wie bereits im Kapitel 5 besprochen, fehlt der Implementierung noch die Verbreitung der CONFIG-Nachrichten per ACTP und damit ein zuverlässiger Terminierungs-Mechanismus.

Darüber hinaus wäre es wichtig, eine Änderung des Netzzustandes effizient zu erkennen und zu propagieren. In jedem Fall kann man das Protokoll wie es hier beschrieben wurde in regelmäßigem Abstand ausführen, um Topologie-Änderungen zu erkennen. Nutzt man dabei stets die bereits bekannte Topologie als Ausgangsbasis, müssen gar nicht viele Topologie-Änderungen übertragen werden, so dass das Protokoll schneller terminieren sollte, als es beim ersten Einmessen der Topologie der Fall war. Darüber hinaus erscheint es sinnvoll — wenn z.B. die Anwendung Topologie-Änderungen wie einen ausgefallenen Knoten feststellt — dass die Knoten eine Topologie-Aktualisierung anstoßen können. Damit von der Anwendung erkannte Topologie-Änderungen effizient propagiert werden können, müsste das Protokoll um einen entsprechenden Modus erweitert werden.

Selbstverständlich wären ausführlichere Tests mit größeren Topologien sehr interessant. Wichtig wäre vor allem zu testen, wie lange es dauert, bis das Protokoll bei einer größeren Topologie terminiert. In der Praxis wird das Protokoll außerdem immer mit Störungen durch andere Protokolle wie WLAN oder Bluetooth zu rechnen haben. Dabei müsste man testen, wie andere Protokolle sich auf die Netzeinmessung auswirken und ob man eventuell Terminierungskriterien anpassen muss, damit das Protokoll trotz Störungen terminiert.

Schlussendlich ist zu untersuchen, wie die durch das Protokoll gewonnenen

Topologie-Informationen durch Anwendungen genutzt werden und ob sich aus den Anwendungen noch weitere bisher nicht beachtete Anforderungen an das Protokoll bzw. die gewonnenen Topologie-Informationen ergeben.

7 Zusammenfassung

Das im Rahmen dieses Projekts entwickelte und implementierte Protokoll ermöglicht es, die Topologie eines Funk-Netzwerkes vollständig selbstständig zu ermitteln und einzumessen. Dabei werden die festgestellten Links nach Sensing-, Interferenz- und Kommunikations-Links klassifiziert, sowie durchschnittliche Empfangsstärken der Links ermittelt. Erste Tests des implementierten Protokolls haben gezeigt, dass das Protokoll in der Lage ist, zuverlässig innerhalb weniger Sekunden kleine Topologien auszumessen.

Literatur

1. Christmann, D., Gotzhein, R., Rohr, S.: The Arbitrating Value Transfer Protocol (AVTP) - Deterministic Binary Countdown in Wireless Multi-Hop Networks. In: Computer Communications and Networks (ICCCN), 2012 21st International Conference on. pp. 1–9 (aug 2012)
2. Christmann, D., Gotzhein, R., Kuhn, T.: Multi-hop clock synchronization in wireless ad-hoc networks. Electronic Communications of the EASST 17 (2009)
3. Gotzhein, R., Kuhn, T.: Decentralized tick synchronization for multi-hop medium slotting in wireless ad hoc networks using black bursts. In: Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on. pp. 422–431. IEEE (2008)
4. Kotz, D., Newport, C., Elliott, C.: The Mistaken Axioms of Wireless Network Research. Tech. Rep. TR2003-467, Dartmouth College (July 2003)
5. MEMSIC Inc.: Imote 2 datasheet. <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=134%3Aimote2> (2013)
6. Sobrinho, J., Krishnakumar, A.S.: Quality-of-service in ad hoc carrier sense multiple access wireless networks. Selected Areas in Communications, IEEE Journal on 17(8), 1353–1368 (1999)
7. Srinivasan, K., Levis, P.: RSSI is Under Appreciated. In: Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets) (2006), <http://www.eecs.harvard.edu/emnets/papers/levisEmnets06.pdf>
8. Texas Instruments: CC2420 datasheet (2013), <http://www.ti.com/lit/ds/symlink/cc2420.pdf>, Revision SWRS041c