

Masterarbeit

Network Formation in Multi-Hop TSCH Netzwerken

Student
Biesinger Christoph
1350722
Masterstudiengang Informatik

Betreuer
Dario Fanucchi

Kurzfassung

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum.

Inhaltsverzeichnis

1	Einleitung	4
1.1	Problemstellung	4
1.2	Ziele	4
1.3	Lösungsansatz	4
2	Grundlagen	4
2.1	IEEE 802.15.4e Timeslotted Channel Hopping	4
2.1.1	Time Division Multiple Access TDMA	4
2.1.2	Enhanced Beacons und Information Elements	4
2.1.3	Network Formation Prozess	4
2.2	Routing Protocol for Low-Power and Lossy Networks RPL	4
2.2.1	Kontrollnachrichten	4
2.2.2	Network Formation Prozess	5
2.3	Contiki	5
2.3.1	Cooja	5
3	Relevante Arbeiten	5
3.1	The Love-Hate Relationship between IEEE802.15.4 and RPL	5
3.2	Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH	5
3.3	DeTAS: Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation	5
3.4	Minimal 6TiSCH Configuration	5
4	Design und Implementierung	5
4.1	Größe von Kontrollnachrichten	5
4.1.1	Enhanced Beacon	5
4.1.2	DODAG Information Objects	6
4.2	Lösungsansatz	6
4.2.1	Eigenschaften	6
4.2.2	Ablaufplan	7
4.2.3	Kollisionsbehandlung	7
4.2.4	Regeln	10
4.2.5	TSCH Network Formation Schedule Information Element	10
4.3	Implementierung	11
4.3.1	Pseudocode	11
4.3.2	Implementierung in Cooja	12
5	Analyse und Evaluation	12
5.1	Testprozess in Cooja/Contiki	12
5.2	Topologien	12
5.3	Leistungsmetriken	13
5.4	Network Formation mit Minimal 6TiSCH Configuration	13

5.5	Network Formation mit Algorithmus	13
5.6	Bewertung der Ergebnisse	13
6	Zusammenfassung	13
6.1	Schlussfolgerungen	13
6.2	Diskussion	13
6.3	Weiterfuehrende Arbeiten	13

1 Einleitung

Lorem ipsum dolor sit amet, [gof, mde:uml211:07b] consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

1.1 Problemstellung

Eine der Hauptaufgaben des Data Link Layers ist der Aufbau einer Knoten-zu-Knoten Datenverbindung anhand der physikalischen Infrastruktur auch Link genannt. Der logische Topologieaufbau soll daher erst auf dem Network Layer mithilfe eines Routingprotokolls bewerkstelligt werden. IEEE802.15.4 dagegen zwingt bereits auf dem Data Link Layer dem Netzwerk eine Topologie auf. Diese Topologien, namentlich Stern, Peer-to-Peer und Cluster-Tree, belegen aufgrund ihrer unterschiedlichen Eigenschaften das Routingprotokoll RPL mit Beschränkungen bei dem Topologieaufbau und damit insgesamt schlechteren Netzwerkleistungen.

Daher wird in dieser Arbeit ein Lösungsansatz mithilfe eines TSCH Scheduling Algorithmus zur Behebung dieses Problems vorgestellt.

1.2 Ziele

1.3 Lösungsansatz

2 Grundlagen

2.1 IEEE 802.15.4e Timeslotted Channel Hopping

2.1.1 Time Division Multiple Access TDMA

2.1.2 Enhanced Beacons und Information Elements

2.1.3 Network Formation Prozess

2.2 Routing Protocol for Low-Power and Lossy Networks RPL

2.2.1 Kontrollnachrichten

DIO

DAO

DIS

2.2.2 Network Formation Prozess

2.3 Contiki

2.3.1 Cooja

3 Relevante Arbeiten

3.1 The Love-Hate Relationship between IEEE802.15.4 and RPL

3.2 Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH

3.3 DeTAS: Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation

3.4 Minimal 6TiSCH Configuration

4 Design und Implementierung

4.1 Grösse von Kontrollnachrichten

Im Rahmen der Entwicklung des Algorithmus kam die ursprüngliche Idee auf, die Kontrollnachrichten von LR-WPAN und RPL innerhalb eines Paketes zu senden, weshalb die Grösse von Enhanced Beacons und DODAG Information Objects analysiert wurden.

4.1.1 Enhanced Beacon

Nach Minimal-6TiSCH beinhaltet ein minimales Enhanced Beacon folgende Teile:

Aufbau Enhanced Beacon

1. Header IE Header
2. Payload IE Header
3. MLME-SubIE TSCH Synchronization
4. MLME-SubIE TSCH Timeslot
5. MLME-SubIE Ch. Hopping
6. MLME-SubIE TSCH Slotframe and Link

4.1.2 DODAG Information Objects

4.2 Lösungsansatz

Als Lösungsansatz werden wir eine strikte Trennung von Kontrollnachrichten und normalen Datentransfer vornehmen und im weiteren nur die Kontrollnachrichten betrachten. Diese werden nach der Idee von Orchestra in Trafficplanes mit eigener Slotframestruktur eingeteilt. Dadurch kann für jede Kontrollnachricht in ihrer eigenen Trafficplane ein privater angepasster Schedule angewendet werden. Diese lokalen Trafficplanes werden anschliessend global auf die Slotframestruktur abgebildet, indem für jede Kontrollnachricht eigene wiederkehrende Timeslots innerhalb des Slotframes festgelegt werden. Dadurch ergeben sich zwei Punkte, welche betrachtet werden müssen:

1. der lokale Schedule zur Abarbeitung der Kontrollnachrichten durch Slotframestruktur
2. ein Algorithmus zur Einteilung des Timeslot- und Channel-Offset anhand der Trafficplanes.

Dieser Algorithmus soll dabei möglichst autonom in abhängigkeit zur Topologie die Erstellung der Slotframestruktur organisieren. Dafür wird beim Scheduling des Timeslot-Offsets ein Konzept aus DeTAS verwendet, indem zwischen den Trafficplanes gewechselt wird (bsp. mit 3 Trafficplanes Enhanced Beacons EB, DODAG Information Object DIO, Unicast RPL Nachrichten DOA, DIS: EB -i DIO -i DOA/DIS -i EB -i DIO -i ...) Eine solche einmalige Zusammenfassung der Kontrollnachrichten bezeichnen wir fortan als Controlmessagestack CMS.

4.2.1 Eigenschaften

-dezentraler Ansatz durch Versenden zusätzlicher Steuerungsmeldungen welche in den Kontrollnachrichten der Technologien eingebettet sind. IEEE 802.15.4e TSCh -i EB RPL -i DIO RPL -i DIS/DAO

-trennung kontrollnachrichten und datentransfer -eigener plane pro kontrollnachricht (eb, dio, dao/dis) -konfliktfreier schedule fuer kontrollnachrichten

Damit wird der globale Schedule von TSCH -feste Frequenz innerhalb eines slotframes -Timeslot Bitmap zur Angabe des lokalen Schedules -selbstständige Konflikte

- Grösse des EB Bereichs N_EB
 - N_EB hat in allen Bereichen des Netzwerks als globale Grösse den gleichen Wert. Der Wert entspricht dabei der höchsten Netzwerkdichte + 1
 - N_EB hat einen variablen Wert und ist abhängig von der Netzwerkdichte in dem jeweiligen Teilbereichen des Netzwerkes.
- Auswahl der Frequenz

- Für die Kontrollnachrichten wird pro Slotframe eine feste Frequenz vorgegeben und nur die beteiligten Links können für die Kontrollnachrichten verwendet werden. Dadurch sind die Planes
- Linkauswahl
 - Die Linkauswahl wird nach dem Zufallsprinzip ausgewählt.
 - Der erste freie Link wird verwendet.

4.2.2 Ablaufplan

Netzwerkaufbau Ablauf beim ersten Aufbau eines Netzwerkes und den benötigten Schritten zusätzlich zum normalen Prozess

1. Knoten schaltet auf TSCH Mode ein
2. Knoten initialisiert Network Formation Data Plane anhand Standard
3. Knoten sendet Enhanced Beacon mit TSCH Network Formation Schedule Information Element

Netzwerkbeitritt Prozess eines Knoten, welcher einem bestehenden PAN beitrete

1. Knoten empfängt EB
2. Knoten hört nach Regel [2] und [3] 4.2.4
3. Analyse aller TSCH Network Formation Schedule IE aus erhaltenen EBs
 - a) Lege Kontrollnachrichtenfrequenz mit momentan verwendeter Frequenz fest
 - b) Berechne weitere Links mit gleicher Kontrollnachrichtenfrequenz innerhalb des vorgegeben EB Bereich N_EB
4. Setze Links in TS Bitmap anhand TSCH Network Formation Schedule IE
5. Wähle einen freien Link nach dem Zufallsprinzip als Sendeslot aus
6. Erweitere N_EB um 1
7. Belege eigene TS Bitmap anhand Regel [4] 4.2.4
8. Empfange Links innerhalb N_EB
9. Sende eigenes Enhanced Beacon im nächsten Slotframe und belegten Timeslot

4.2.3 Kollisionsbehandlung

Im Rahmen des Schedules können zwei Arten der Kollision festgestellt werden, einmal die Kollision unterhalb und die Kollision oberhalb. Die Kollision unterhalb tritt hauptsächlich im Rahmen des Network Formation Prozess auf, wenn zwei Kindknoten gleichzeitig sich an einen Parentknoten binden wollen. Die Kollision oberhalb erscheint,

wenn ein Kindknoten sich an zwei Parentknoten binden möchte. Dieser Kollisionstyp erscheint nur, wenn der Bereich der Kontrollnachrichten anhand der lokalen Netzwerkdichte ermittelt wird.

Kollision unterhalb Hierbei erkennt ein Parentknoten, dass zwei oder mehr Kindknoten den gleichen Timeslot verwenden wollen, dabei reagieren sowohl Parentknoten und Kindknoten unterschiedlich.

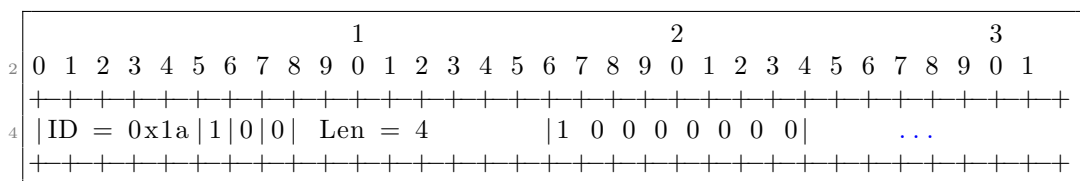
```

1  // Platzhalter
   //      n0
3  //      /  \
   //    n1    n2
5  //
   // Bitmaps fuer n0
7  // 1.SF: 10
   // 2.SF: 1C
9  // 3.SF: 1000
   // 4.SF: 1110 (n1 und n2 haben zufaellig Timeslot 2 und 3
   //      ausgewaehlt)
11

```

Aktionen durch Parentknoten 1. Parentknoten empfängt innerhalb eines freien Timeslot mehrere Enhanced Beacons und erkennt damit eine Kollision

2. erweitere N_EB um 2
3. belege Timeslot Bitmap anhand Regel [4] 4.2.4 (Timeslot wird nicht markiert)
4. setze Pending Flag
5. sende eigenes Enhanced Beacon zum nächsten Zeitpunkt

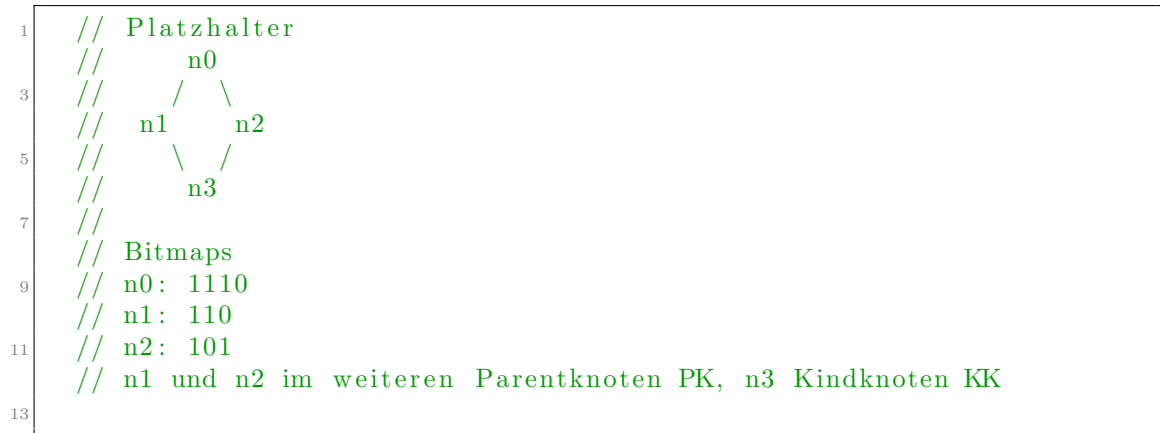


Aktionen durch Kindknoten Der Kindknoten empfängt weitere Enhanced Beacons von Parentknoten

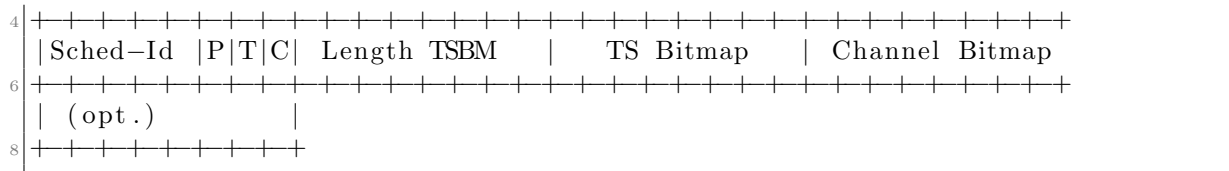
1. Knoten empfängt Enhanced Beacon von Parentknoten
2. Analyse TSCH Network Formation Schedule Information Element
 - a) Betrachte Timeslot Bitmap
 - i. Wenn eigener Timeslot markiert ist war die Kommunikation erfolgreich und der Link wird behalten

- ii. Wenn eigener Timeslot nicht markiert ist, fand eine Kollision mit fremden Knoten statt
 - A. Setze Pending Flag
 - B. Wähle einen freien Link nach dem Zufallsprinzip aus
 - iii. Wenn kein weiterer freier Timeslot zum Beitreten eines Knoten vorhanden ist, wird N_EB um 1 erweitert
 - iv. Setze Timeslot Bitmap anhand Regel [4] 4.2.4
3. Sende eigenes Enhanced beacon zum n"chstmöglichen Zeitpunkt

Kollision oberhalb Ein Kindknoten will eine Verbindung mit zwei Parentknoten erstellen, erkennt aber anhand den angebotenen Timeslots, dass immer eine Kollision mit dem jeweils anderen Parentknoten auftreten wird.



- Aktion des Kindknoten**
1. Kindknoten empfängt Enhanced Beacons von mehreren Parentknoten
 2. Analyse TSCH Network Formation Schedule Information Element
 - a) Betrachte Timeslot Bitmap
 - b) Initialisiere interne Bitmap anhand den Werten des ersten Parentknoten
 - c) Vergleiche Werte von anderen Parentknoten für jeden Timeslot
 - i. Wenn interner Bitmap den Wert 1 anzeigt und der neue Wert ebenfalls eine 1 anzeigt "ändert sich der Wert nicht.
 - ii. Wenn interne Bitmap den Wert 1 anzeigt und der neue Wert den Wert 0 aufweist, wird der Timeslot gesondert markiert.
 - iii. Wenn interne Bitmap den Wert 0 aufweist und der neue Wert den Wert 1, wird der Timeslot gesondert markiert.



Schedule-ID (5 Bit) Nummer (0x00 - 0x1f) welche einem Kontrollplane eindeutig zugeordnet wird und diesen einleitet

Bereich	Beschreibung
0x00 - 0x19	nicht verwendet
0x1a	Enhanced Beacons
0x1b	DIO-Nachrichten
0x1c	DIO/DIS-Nachrichten
0x1d - 0x1f	nicht verwendet

Pending Flag (1 Bit) ein aktives Pending Flag zeigt an, dass der Schedule momentan nicht fest zugeordnet ist und noch Probleme ausstehen (Kollision)

Type Flag (1 Bit) Beschreibt die Länge der TS Bitmap

Flag	Typ	Beschreibung	Mögliche Timeslots
0	Short Length	Länge der TS Bitmap sind 8 Bit	256
1	Long Length	Länge der TS Bitmap sind 16 Bit	65536

Channel Flag (1 Bit) liefert den Hinweis "über eine aktive Channel Bitmap"

Length TSCH Network Formation Schedule Information Element (1/2 Byte) beschreibt die Grösse der Timeslot Bitmap

Timeslot Bitmap (variable vielfache von 8 Bit) Liefert die Belegung des lokalen Schedules

Channel Bitmap (16 Bit) optionale Bitmap, zur Angabe der aktiven Channel Offsets

4.3 Implementierung

4.3.1 Pseudocode

Initialisierung

Platzhalter

Knoten beitreten

```
1 Platzhalter
```

Kollisionsbehandlung unterhalb

```
1 Platzhalter
```

...

```
1 Platzhalter
```

4.3.2 Implementierung in Cooja

c cooja

beachte datenstrukturen

5 Analyse und Evaluation

5.1 Testprozess in Cooja/Contiki

Der Testablauf wird zwei Bereiche aufgeteilt, den

- Netzwerkaufbau
- Beitrittsprozess

5.2 Topologien

Für Testzwecke wurden zwei Netzwerktopologien ausgewählt um den Algorithmus mit mehreren Anwendungsfällen zu kontrollieren

Netzwerk A Platzhalter - Durch Bild ersetzen

```
1 //
2 // n0 - n1 - n2
3 // |   |   |
4 // n3 - n4 - n5
5 // |   |   |
6 // n6 - n7 - n8
7 //
```

Netzwerk B Knoten n6 betritt das bereits etablierte Netzwerk mit zwei Parentknoten
Platzhalter - Durch Bild ersetzen



5.3 Leistungsmetriken

Network Formation Time Zeit vom ersten Enhanced Beacon, bis letzter Knoten dem Netzwerk beigetreten ist

Joining Time Zeit die ein Knoten benötigt um einem Netzwerk beizutreten (abhängig von festgelegten Regeln)

Overhead Welchen Overhead an Nachrichten erzeugt der Algorithmus

Zeitraum Kontrollnachrichten Wieviel Zeit wird für die Kontrollnachrichten reserviert

Verhalten Algorithmus “über längeren Zeitraum Wie verhalten sich die Leistungsmetriken “über einen längeren Zeitraum?

Aktivitätszeit Knoten Wie lange ist ein Knoten im Rahmen des Schedules aktiv im Verhältnis zur tatsächlichen Aktivität.

Energieverbrauch Wieviel Energie

5.4 Network Formation mit Minimal 6TiSCH Configuration

5.5 Network Formation mit Algorithmus

5.6 Bewertung der Ergebnisse

6 Zusammenfassung

6.1 Schlussfolgerungen

6.2 Diskussion

6.3 Weiterführende Arbeiten