

Testkonzept Punkt zu Punkt Verbindung

Testumgebung und Performancevergleich von Punkt zu Punkt Verbindungen auf MAC Ebene mit BLE und IEEE802.15.4

BACHELOR THESIS - ANKLIN, BOBST, HORATH
3. Mai 2020

Fachcoach:

Matthias Meier
Manuel Di Cerbo

Team:

Raffael Anklin
Robin Bobst
Cyrill Horath

Studiengang:

Elektro- und Informationstechnik

Semester:

Frühlingssemester 2020

Inhaltsverzeichnis

1 Konzept

Für den Vergleich der 3 Mesh Netzwerkstacks Bluetooth Mesh (BT Mesh), Thread und Zigbee wird ein vom Mesh Protokoll unabhängiges Testkonzept umgesetzt welches in der Abbildung ?? als Konzeptschema dargestellt ist. Die Benchmark Slave Nodes (BSN) in der Abbildung als Sensoren und Aktoren mit unterschiedlichen Funktionalitäten dargestellt, bilden zusammen mit dem Benchmark Master Node (BMN) das zu testende Mesh Netzwerk. Innerhalb des Netzwerks wird dessen Organisation vom jeweiligen Protokoll sichergestellt. Das Testnetzwerk soll ein realitätsnahes Netzwerk nachbilden. Beispielsweise wird eine Hausautomation in einem Einfamilienhaus als Referenz angenommen in welchem jeweils nur gewisse Nodes untereinander Applikationsdaten austauschen. Ein Lichtschalter kommuniziert nur mit einer Lichtquelle und umgekehrt. Der selbe Lichtschalter tauscht jedoch keine Applikationsdaten mit dem Temperatursensor aus. Trotzdem bilden die Nodes zusammen ein Mesh Netzwerk. Diese unterschiedlichen Beziehungen innerhalb des Mesh Netzwerks sind in der Abbildung ?? bereits angedeutet und werden im Abschnitt ?? noch genauer beschrieben.

Die Benchmark Management Station (BMS) welche mit dem BMN via USB/UART kommuniziert, ist zuständig für die Verwaltung und Verarbeitung der Benchmarks. Während eines Benchmark Prozesses sollen sämtliche Messungen jedoch unabhängig von der BMS durchgeführt werden damit allfällige Latenzzeiten der USB/UART Verbindung die Resultate nicht verfälschen.

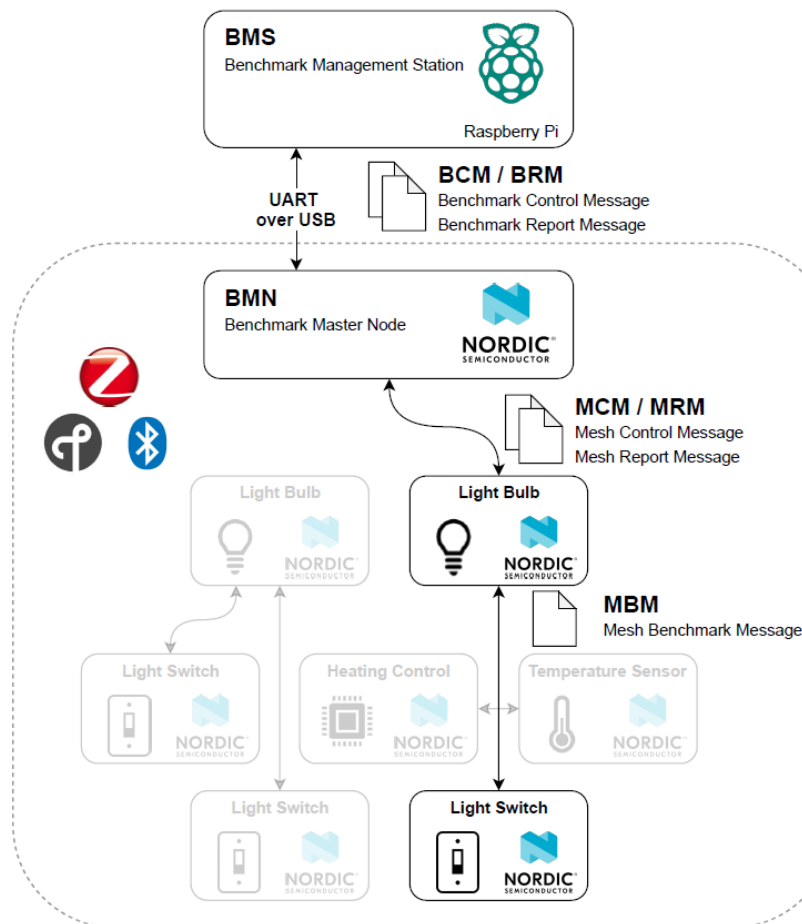


Abbildung 1.1: Konzeptschema für den Ablauf eines Mesh Benchmarks.

In der Abbildung ?? sind verschiedene Messages dargestellt. Dabei handelt es sich um die Nachrichten die zwischen den einzelnen Teilen des Testaufbaus versendet werden und schliesslich einen Benchmark ausmachen. Die Messages besitzen Funktionen:

Mesh Benchmark Message (MBM)

Die MBM ist jene Message welche die eigentlichen Messdaten produziert und diese sogleich unter den BSN (Mesh Knoten) überträgt. Anhand dieser Messages werden die Parameter gemäss der Messwerttabelle in Anhang ?? erfasst. Bei den MBM handelt es sich also eigentlich um eine Sammlung von Messages welche je nach gewünschtem Messwert in Form und Anzahl unterschiedlich ausfallen können.

Mesh Control Message (MCM)

Die MCM beinhaltet die Parameter für die Benchmarks welche vom BMN an alle BSN übertragen werden. Ausserdem werden damit Kontrollbefehle für die Benchmarks wie beispielsweise *Start/Stop* sowie *Laufzeit*, *Wiederholrate* usw. übertragen.

Mesh Report Message (MRM)

Die MRM ist jene Message welche die Messwerte von den BSN an den BMN übertragen. Gleichzeitig wird damit auch gleich signalisiert, dass die Messung abgeschlossen wurde und mögliche Fehler oder sonstige Stati übermittelt.

Benchmark Control Message (BCM)

Die BCM beschreibt die Nachrichten welche zur Steuerung eines Benchmarks von der BMS her dienen. Dabei handelt es sich um Befehle wie beispielsweise *Start/Stop*. Die BCM werden via serieller USB-UART Schnittstelle von der BMS zum BMN übertragen.

Benchmark Report Message (BRM)

Die BRM beschreiben Nachrichten welche den Status oder die Ergebnisse eines Benchmarks aus dem Mesh zurück an die BMS melden. Sie werden vom BMN initiiert und gelangen über eine die selbe USB-UART Schnittstelle wie die BCM zum BMS. Die BRM wird erst nach Abschluss des Benchmarks initiiert. Zuvor werden die Messdaten auf dem BMN zwischen gespeichert.

2 Ablauf

Ein Mesh Benchmark folgt einem klar definierten Ablauf. Die Abbildung ?? zeigt das Testkonzept in welchem auch der Ablauf eines Benchmarks bereits angedeutet ist.

1. **Benchmark User-Init:**

Auf dem Webinterface des BMS werden die gewünschten Parameter definiert und der Benchmark durch den Benutzer gestartet.

2. **Benchmark Init BMN:**

Die Parameter werden an den BMN übergeben welcher diese wiederum an alle teilnehmenden BSN weiterleitet. Mit einem Startsignal vom BMN wird der Benchmark auf den BSN gestartet.

3. **Benchmark Prozess:**

Die BSN führen den Benchmark Prozess mit den definierten Parametern aus. Dies geschieht autonom und jeweils nur zwischen den entsprechenden BSN die in einer direkten Beziehung zueinander stehen (siehe Mesh Beziehungen ??). Die entstehenden Messdaten werden auf den BSN zwischen gespeichert.

4. **Reporting:**

Nach Ablauf der Benchmark Zeit werden die Messdaten an den BMN übertragen. Dies erfolgt gesteuert durch den BMN welcher die Daten bei einem BSN nach dem anderen abfragt und direkt an das BMS weiterleitet.

5. **Finish:**

Der BMN kontrolliert ob er die Daten von sämtlichen BSN korrekt auslesen konnte und bestätigt das Ende der Messung gegenüber dem BMS.

6. **Auswertung:**

Das BMS beendet den Benchmark Vorgang, speichert die Messdaten in seiner Datenbank ab und bereitet diese grafisch auf.

3 Testszzenarien

Die Benchmarks der Mesh Protokolle sollen mit unterschiedlichen Bedingungen getestet werden wobei grundsätzlich eine reelle Anwendung nachgebaut werden soll. Zum einen gibt es unterschiedliche Beziehungen innerhalb des Mesh Netzwerks, zum anderen werden Testumgebungen unterschieden.

3.1 Mesh Beziehungen

Innerhalb eines Mesh Netzwerks können 4 Beziehungen zwischen den Nodes für die Benchmarks unterschieden werden. Üblicherweise kommen mehrere oder sogar alle 4 Beziehungen innerhalb eines Netzwerkes gleichzeitig zum Einsatz. Abbildung ?? zeigt die Beziehungen.

- Rot stellt eine einfache P2P Verbindung ohne Hop dar. Beispielsweise schaltet ein einzelner Schalter eine einzelne, definierte Lichtquelle
- Orange ist eine many-to-one Verbindung in welcher mehrere Lichtschalter die selbe Lichtquelle schalten.
- In blau ist eine klassische one-to-many Topologie dargestellt in welcher beispielsweise ein Schalter mehrere Lichtquellen bedient.
- Grün dargestellt ist eine indirekte P2P Verbindung mit. Das bedeutet, dass Schalter und Lichtquelle keine direkte Verbindung zueinander haben und daher Mesh-typisch via einem oder mehreren Hops kommuniziert.

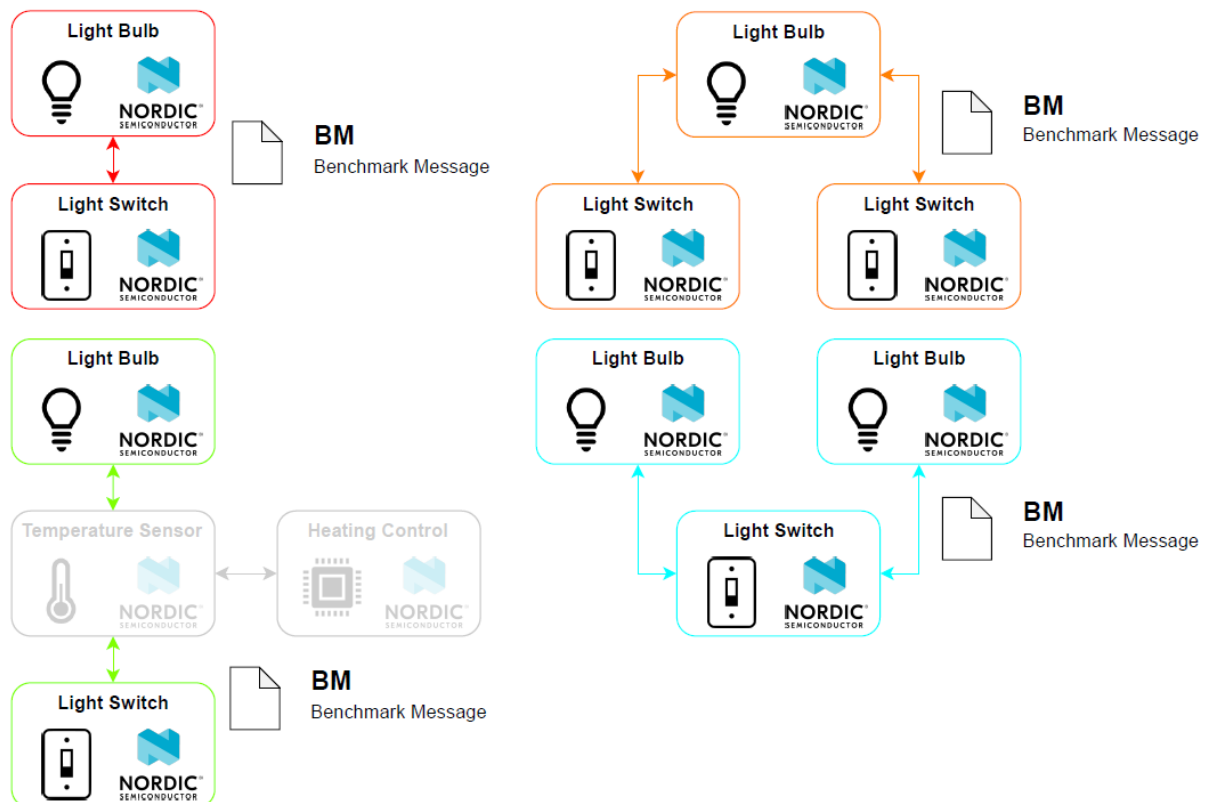


Abbildung 3.1: Beziehungen zwischen den Mesh Nodes innerhalb eines Benchmarks.

3.2 Testumgebungen

Unterschiedliche Testumgebungen sollen die Benchmarks und schlussendlich den Vergleich der 3 Mesh Protokolle aussagekräftiger machen. Abbildung ?? zeigt 5 unterschiedliche Umgebungen in denen Messungen durchgeführt werden sollen.

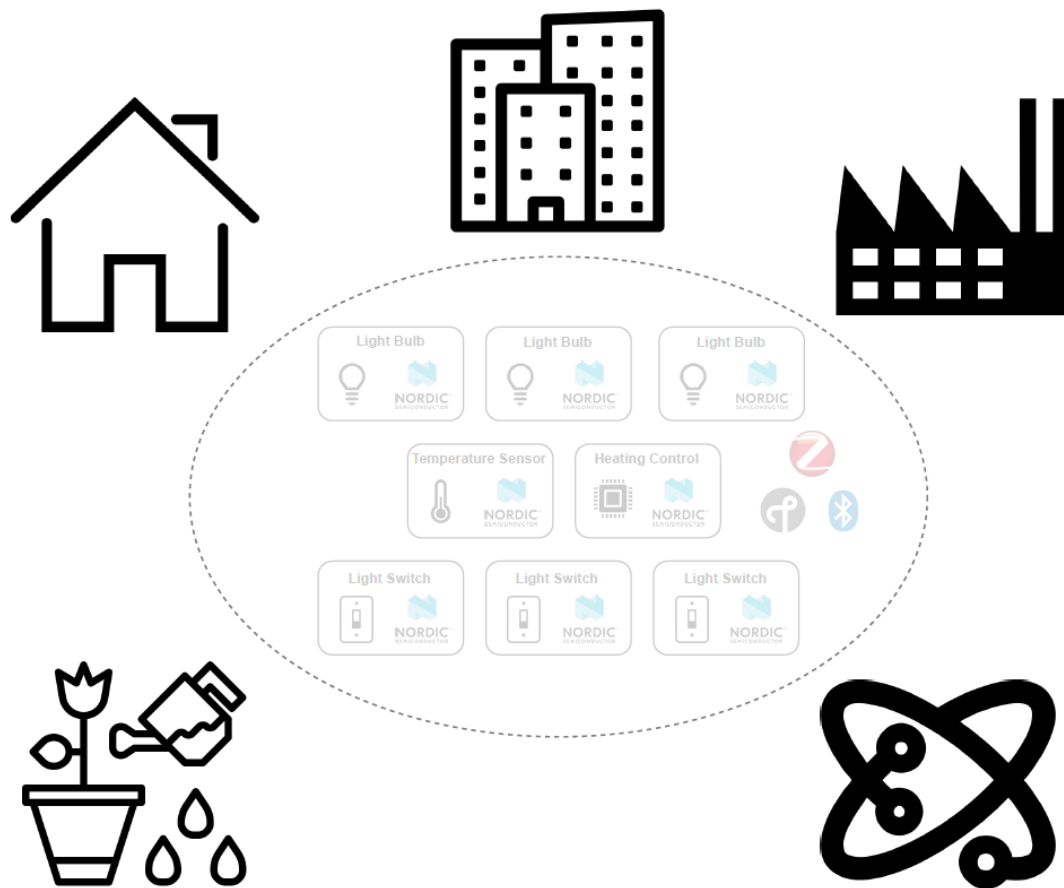


Abbildung 3.2: Mesh Netzwerk Testumgebungen

Haus

Die Testgeräte werden in einem Einfamilienhaus installiert und repräsentieren damit eine flächendeckende Heim-Automatisierung.

- Einfamilienhaus über mehrere Etagen.
- Anzahl Sensoren und Aktoren vergleichbar gross.
- Node-Dichte relativ gering.
- Keine Beeinflussung durch Nachbarsysteme zu erwarten

Wohnung

Ebenfalls als Heim-Automatisierung gedacht werden die Messungen in einer Wohnung durchgeführt.

- Wohnung über eine Etage in einem Mehrfamilienhaus
- Anzahl Sensoren und Aktoren vergleichbar gross.
- Node-Dichte höher als im Haus.
- Mögliche Störeinflüsse durch andere Systeme von Nachbarn zu erwarten.

Industrie

Um eine Industrielle Anwendung zu vergleichen erfolgt eine Messung in einem Industriebetrieb.

- Industriebetrieb mit grosser Fläche.
- Grosse Anzahl Sensoren zur Überwachung von Produktionsprozessen. Vereinzelt Aktoren zur Ansteuerung von Anlageteilen.
- Hohe Node-Dichte.
- Mögliche Störeinflüsse durch Maschinen oder Abschirmwirkung durch metallische Gegenstände zu erwarten.

Labor

Der Laboraufbau ist ein Extremtest welcher die Leistungsgrenzen der Protokollstacks ausloten soll.

- Testaufbau unter Laborbedingungen auf engstem Raum.
- Ausgeglichene Anzahl Sensoren und Aktoren.
- Sehr Hohe Node-Dichte.
- Geringe bis keine Störbeeinflussung durch die Umgebung zu erwarten.

Landwirtschaft (optional)

Für die Überwachung und Kontrolle von landwirtschaftlichen Flächen kann ein Test auf offenem Feld erfolgen.

- Landwirtschaftsfläche mit grosser Ausbreitung (z.B. Gemüseanbau).
- Grosse Anzahl Sensoren. Nur wenige bis gar keine Aktoren.
- Sehr geringe Node-Dichte mit weiten Distanzen.
- Geringe bis keine Störbeeinflussung durch die Umgebung zu erwarten.

4 Firmware

Eine Hardwareplattform. Dongle mit Akku für die BSN und DK für den BMN. Firmware dementsprechend gibt es folgende: BMN, BSN Sensor, BSN Aktor

Räffu: Mesh-5, 7, 8, 9

4.1 Signal to Noise Ratio

Das Signal to Noise Ratio beschreibt die Übertragungsqualität eines Kanals. Dabei muss das Rauschen ohne Übertragung von der Gegenseite gemessen werden P_{rauschen} . Anschliessend wird die Signalstärke von der Gegenseite gemessen P_{signal} . Der SNR wird gemäss der Formel ?? berechnet.

$$SNR = P_{\text{signal}} / P_{\text{rauschen}} \quad (4.1)$$

Gemessen wird das Signal to Noise Ratio über die verschiedenen Kanäle, welche zur Übertragung in BLE oder IEEE802.15.4 verwendet werden (siehe Abbildung ??).

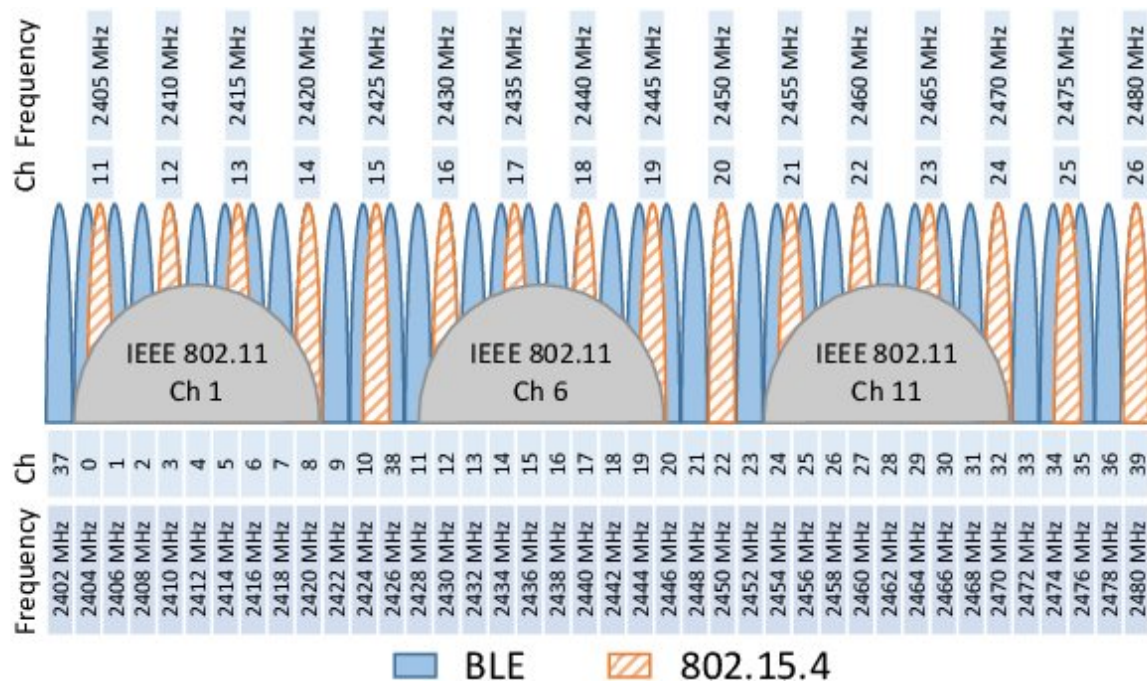


Abbildung 4.1: BLE und IEEE802.15.4 Kanäle

4.2 Durchsatz Messung

Robin: Mesh-1, 3, 4

Der Durchsatz wird gemessen, indem ein Client eine fixe vordefinierte Payload an den Server schickt. Die Zeit, die das Paket braucht um vollständig übertragen zu werden, wird gemessen. Aus dieser Zeit und der Payload wird der Durchsatz berechnet. Für die Durchsatzmessung sind zwei verschiedene Modi vorgesehen. Der erste Modus misst den Durchsatz ohne applikations Acknowledgment vom Server und der zweite Modus mit Acknowledgement. Im Bild ?? sind beide Varianten beschrieben. Die Anzahl Messungen kann vor dem Test definiert werden.

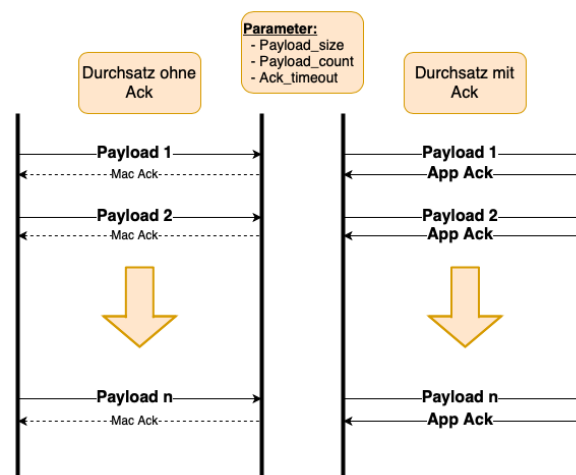


Abbildung 4.2: Konzept Durchsatzmessung

4.3 Latenz Messung

Die Latenz beschreibt die Zeit, wie lange ein Paket hat bis es vollständig übertragen wurde. Da die Messung auf applikations Ebene stattfindet, wird die Zeit vom Funktionsaufruf des Clients bis zur Bestätigung vom Server gemessen.

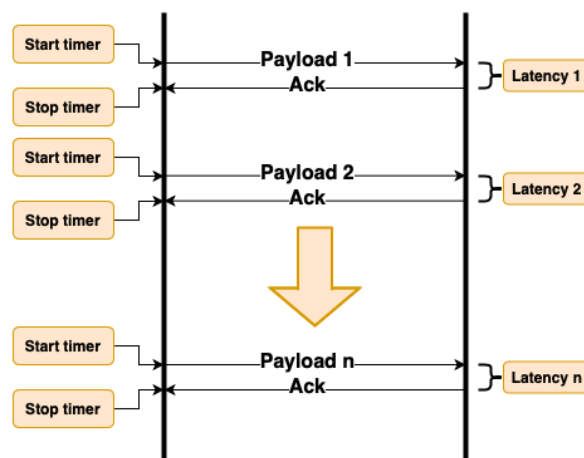


Abbildung 4.3: Konzept Latenzmessung

Cyrill: Mesh-2, 6, 10

4.4 **Parameter Mesh Test**

5 Hardware

Eine Hardwareplattform. Dongle mit Akku für die BSN und DK für den BMN. Firmware dementsprechend gibt es folgende: BMN, BSN Sensor, BSN Aktor

Für den Benchmark und den Vergleich der drei Mesh Protokolle wird mit dem nRF52840 SoC eine gemeinsame Hardwareplattform eingesetzt. Es werden die Entwicklungsboards nRF52840-Dongle für die Benchmark Slave Nodes und das nRF52840-DK für den Benchmark Master Node verwendet. Damit ist der Test einfach nachvollziehbar und kann einfach nachgebaut werden ohne dass dafür spezielle Hardware benötigt wird. Zumindest beim DK trifft dies zu 100% zu. Bei den Dongle hingegen fehlt die Speisung da diese verteilt und unabhängig von einer fixen Spannungsversorgung betrieben werden sollen. Die Speisung wird daher mit einem 500mAh Li-Po Akkupaket realisiert. Mit einem konservativ berechneten Strombedarf von 20mA ist damit also ein Testbetrieb des Mesh Netzwerks über 24 Stunden möglich.

A Messwerte Benchmark Mesh Netzwerke

Index Messung	Mesh-1	Mesh-2	Mesh-3	Mesh-4	Mesh-5	Mesh-6	Mesh-7	Mesh-8	Mesh-9	Mesh-10
Bezeichnung	Latency Time	Number of hops	Data Transmission Rate Unacknowledged	Data Transmission Rate Acknowledged	RSSI	Packet-loss	Active radio-time	Active CPU-time	Theoretical power consumption	Number of retries
Beschreibung	Bestimmung der Latenzzeit von Aktor zu Sensor über Anzahl Hops.	Bestimmung der Anzahl Hops, die eine Nachricht nehmen musste.	Bestimmen der Datenübertragungsrate (Unbestätigt)	Bestimmen der Datenübertragungsrate (Bestätigt)	Bestimmung des RSSI von verschiedenen Nodes	Bestimmung der Anzahl verlorenen Pakete	Bestimmung der Aktiven Radio Zeiten	Bestimmung der Aktiven CPU Zeit	Bestimmung der theoretischen Leistungsaufnahme.	Anzahl Retries
Messgrösse	Latenzzeit	n = Anzahl Hops	Datenübertragungsrate	Datenübertragungsrate	Empfangsleistung	Paketverlust	Zeit	Zeit	Leistung	n = Anzahl Retries
Einheit	Millisekunden (ms)	-	kBit/s	kBit/s	dBm	Verhältnis gesendete Pakete zu verlorene Pakete in %	Millisekunden (ms)	Sekunden (s)	Milliwatt (mW)	-
Vorgehen	Die Latenzzeit wird immer von einem Sensor zu einem Aktor gemessen, z.B. von einem Lichtschalter zum Licht. Wenn die Nachricht vom Sensor gesendet wird, wird ein Timestamp als Payload der Nachricht hinzugefügt. Beim Aktor werden weitere Timestamps zum Payload hinzugefügt und dem Sensor als Acknowledge zurückgeschickt. Im Sensor wird danach die Latenzzeit anhand der Timestamps berechnet.	Auf einem Node werden die Next Hop Informationen lokal gespeichert. Diese Information wird der Nachricht als Payload mitgegeben, um am Ziel Node auszuwerten wie viele Hops die Nachricht genommen hat.	Es werden Datenpakete verschiedener Länge (1Byte - ca. 1MByte) zufällig generiert. Anschliessend wird wie bei Mesh 1 eine Zeitsynchronisation durchgeführt, dabei wird zusätzlich die Grösse der Datenmenge angegeben. Nach Bestätigung der Bereitschaft beginnt der Sensor mit der Übertragung der Datenpakete. Wurde das erste Datenpaket erhalten, so wird dies gepuffert und die Empfangszeit T2 gespeichert. Ist die vollständige Datenmenge beim Node angekommen wird die Differenz aus der aktuellen Zeit und T2 gebildet. Diese bestimmt die Übertragungszeit. Anschliessend wird diese dem Sensor zurückgesendet, welcher die Datenrate aus dem Quotient der Datenmenge und Übertragungszeit bildet.	Der Ablauf ist mit T3 identisch, ausser dass der Erhalt von jedem Datenpaket (ebenfalls segmentiert) bestätigt werden muss. Die Zeitmessung ist mit der letzten Bestätigung an den Sensor abgeschlossen.	Der RSSI Wert wird von den verschiedenen Nodes erfasst und als Payload den Nachrichten mitgegeben und dem Master zugeschickt.	Die Paketnummer vom empfangen Signal wird ausgelesen und mit der Tatsächlichen Paketnummer, die in der Payload mit geliefert wird verglichen. Das Verhältnis zwischen den Werten stellt den Paketverlust dar.	Beim Einschalten und Ausschalten der Rx- / Tx-Schnittstelle wird ein Timer gestartet bzw. gestoppt, so wird die aktive Radio Zeit ermittelt.	Beim Ein- und Ausschalten der CPU soll ein Timer gestartet bzw. gestoppt werden, so wird die aktive CPU Zeit gemessen.	Anhand der gemessenen Radio und CPU Zeiten wird die Leistung berechnet.	Wird das Acknowledge nicht quittiert, wird die Nachricht erneut gesendet. Diese Anzahl Retries werden ermittelt und der Payload mitgegeben.
Störfaktoren	Umliegende Kommunikationsgeräte, welche das 2.45GHz ISM Band benutzen.									
Anzahl Wiederholungen	Periodisch									
Einstellbare Parameter	-	Anzahl Hops kann begrenzt werden	Packetsize	Packetsize	-	-	-	-	-	-
Voraussetzungen	Node muss bereit und konfiguriert sein. Zeit der Nodes muss synchronisiert.	Node muss bereit und konfiguriert sein.								
Allgemeine Bedingungen	Die Tests werden unter belastetem und unbelastetem Mesh-Netzwerk durchgeführt									