



School of
Engineering

InES Institute of
Embedded Systems

Institute of Embedded Systems - InES

Analyse der Empfangshäufigkeit von Bluetooth Low Energy Paketen für iPhone Applikationen im Background-Mode

Bestimmung der optimalen Sendeparametern für Beacons in
Gemeinschaft basierten Datennetzwerken

Dominique Truninger
03.03.2015

Inhalt

1.	Einleitung.....	2
1.1	Versuchsaufbau.....	2
1.2	Sendeparameter.....	3
2.	Erste Testphase mit konstanter Sendezeit.....	5
2.1	Parametrisierung vom 16.01.2015 – Standby-Zeit und Sendeintervall	5
2.2	Parametrisierung vom 23.01.2015 – Standby-Zeit und Sendeintervall	7
2.3	Parametrisierung vom 27.01.2015 – Standby-Zeit und Sendeintervall	8
3.	Zweite Testphase mit konstanter Standby-Zeit	9
3.1	Parametrisierung vom 04.02.2015 – Sendezeit und Sendeintervall.....	9
3.2	Parametrisierung vom 09.02.2015 – Sendezeit und Sendeintervall.....	11
4.	Auswertung	12
5.	Abbildungsverzeichnis.....	12
6.	Contact	12

1. Einleitung

Heute existieren bereits viele Bluetooth Low Energy (BLE) Applikationen in den unterschiedlichen App Stores von Google, Apple und Microsoft. Die meisten dieser Applikationen setzen voraus, dass entweder eine Verbindung mit einem BLE-Device eingegangen oder die Applikation im Vordergrund betrieben werden muss. In einem gemeinschaftsbasierten Datennetzwerk wäre jedoch keines dieser beiden Szenarien praktikabel. Oft wäre es wünschenswert eine Applikation im Hintergrund zu betreiben und aktiv nach BLE-Devices zu scannen. Jedes Betriebssystem löst das aktive Scannen nach BLE-Devices anders und oft verändert sich das Verhalten einer Applikation wenn sie im Background arbeiten muss. Denn sobald eine Applikation im Background-Mode arbeitet, bekommt sie nur noch ein begrenztes Zeitfenster um seine Tasks abzuarbeiten.

Dieses Dokument befasst sich mit der Analyse und Auswertung einer Testreihe zur Bestimmung der Empfangshäufigkeit von BLE-Paketen einer iPhone Applikation im Background-Mode. Um einen Beacon möglichst energieeffizient zu implementieren, muss das Verhalten des Betriebssystems iOS und die Empfangshäufigkeit im Hintergrund bestimmt werden können. Zur Bestimmung der optimalen Sendeparameter eines Beacons wurden diverse Tests durchgeführt, welche das Send- und Empfangsverhalten einer BLE-Übertragung für das Betriebssystem iOS 7.0 überprüfen sollten.

1.1 Versuchsaufbau

Zum Testen der Empfangshäufigkeit von BLE-Paketen im Background-Mode wurde eigens eine Firmware für Beacons entwickelt, welche stündlich ein neues Test-Szenario durchspielen soll. Die dabei zu überprüfenden Parameter sind:

- die Sendeintervall (auch als Sendehäufigkeit bezeichnet)
- die Standby-Zeit (Zeit in dem der Beacon im Sleep ist)
- die Sendezeit (Zeitfenster in welchem BLE-Pakete gesendet werden)

Durch variieren dieser Parameter entstehen unterschiedliche Empfangshäufigkeiten am Smartphone, welche anschliessend analysiert und ausgewertet werden können. Abbildung 1 zeigt den Versuchsaufbau mit zwei Beacons und einem iPhone.

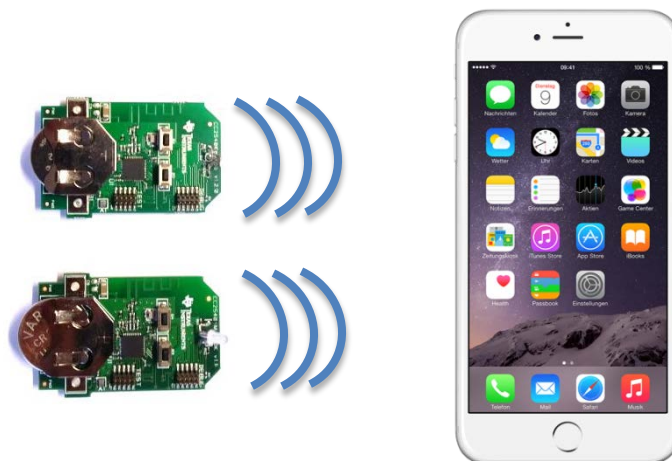


Abbildung 1: Versuchsaufbau

Jedes Testszenario wird exakt eine Stunde durchgeführt. Nach jeder Stunde wechselt einer der drei Sendeparameter den Wert. Ziel dieser Arbeit ist die Bestimmung der optimalen Sendeparameter, bei welchen die Empfangshäufigkeit am iPhone am höchsten ist (verglichen mit der verbrauchten Energie am Beacon). Die Abbildung 2 zeigt die drei Sendeparameter.

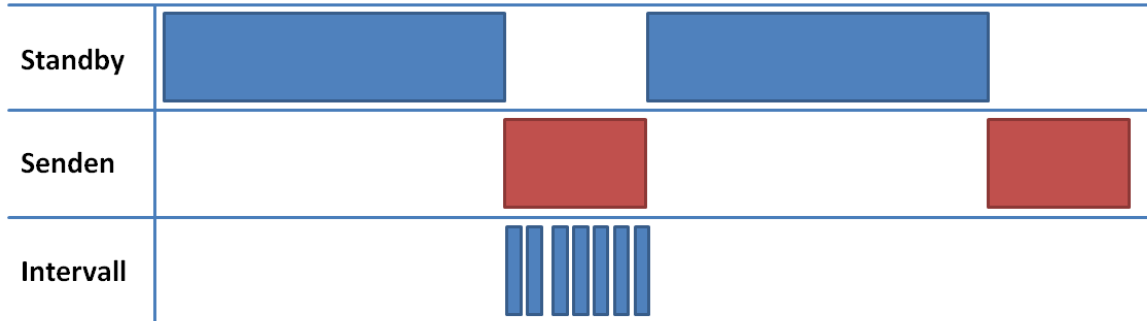


Abbildung 2: Die drei Sendeparameter – Standby-Zeit, Sendezeit und Sendeintervall

Die dazu entwickelte iOS Applikation arbeitet im Hintergrund und scannt die Umgebung nach BLE-Geräten. Wird ein BLE-Paket von einem Beacon empfangen, so werden der Zeitpunkt und die Kennung des Beacons in einer Datei festgehalten.

1.2 Sendeparameter

Die Tests wurden in zwei unterschiedliche Versuche unterteilt. In der ersten Testphase wurde die Sendezeit konstant bei einer Sekunde (1 s) gehalten. Stündlich änderte sich entweder der Wert des Sendeintervalls oder der Standby-Zeit. Tabelle 1 zeigt die Parameter der ersten Testphase. Bei einer Möglichkeit von sieben verschiedenen Zuständen für das Sendeintervall und sieben verschiedenen Zuständen für die Standby-Zeit, ergeben sich insgesamt also 49 unterschiedliche Test-Szenarien die untersucht wurden.

Sendezeit [ms]	1000	1000	1000	1000	1000	1000	1000
Sendeintervall [ms]	20	40	60	80	100	120	140
Standby-Zeit [s]	10	11	12	13	14	15	16

Tabelle 1: Variation der Standby-Zeit und des Sendeintervalls bei konstanter Sendezeit

In einem zweiten Versuch wurde die Standby-Zeit konstant bei 15 Sekunden gehalten, wobei die Sendezeit und das Sendeintervall nach Tabelle 2 verändert wurden.

Sendezeit [ms]	200	300	400	500	600	700	800
Sendeintervall [ms]	20	40	60	80	100	120	140
Standby-Zeit [s]	15	15	15	15	15	15	15

Tabelle 2: Variation des Sendeintervalls und der Sendezeit bei konstanter Standby-Zeit

Die Tabelle 3 zeigt die einzelnen Test-Szenarien für die erste Testphase. Jedes Szenario wurde genau eine Stunde durchlaufen, um eine Aussage über das Verhalten des Betriebssystems zu erhalten. Es ist zu erkennen dass die Sendezeit in der Testphase konstant bei einer Sekunde gehalten wurde (rechte Spalte – Sendezeit [s]). Bei jedem, am Smartphone, empfangenen Datenpaket wurde ein Logeintrag mit der Bezeichnung des Test-Szenarios, der Beacon-Adresse sowie der Uhrzeit und Datum abgespeichert.

Insgesamt wurden 49 Test-Szenarios durchgespielt, was demnach 49 Stunden dauerte. Aus hardwaretechnischen Gründen mussten die Tests jedoch teilweise früher beendet werden. Dauerte ein Test mehr als 49 Stunden, so begann die Software automatisch erneut und die Test Cases wurden ein zweites Mal durchlaufen.

<i>Test Case</i>	<i>Standby-Zeit [s]</i>	<i>Sendeintervall [ms]</i>	<i>Sendezeit [s]</i>
00t0i0	10	20	1
00t0i1	10	40	1
00t0i2	10	60	1
00t0i3	10	80	1
00t0i4	10	100	1
00t0i5	10	120	1
00t0i6	10	140	1
00t1i0	11	20	1
00t1i1	11	40	1
00t1i2	11	60	1
00t1i3	11	80	1
00t1i4	11	100	1
00t1i5	11	120	1
00t1i6	11	140	1
00t2i0	12	20	1
00t2i1	12	40	1
00t2i2	12	60	1
00t2i3	12	80	1
00t2i4	12	100	1
00t2i5	12	120	1
00t2i6	12	140	1
00t3i0	13	20	1
...
...

Tabelle 3: Alle Parameter und Test-Szenarien im Überblick

2. Erste Testphase mit konstanter Sendezeit

In der ersten Testphase wurde der Parameter Sendezeit konstant gehalten. Die beiden Parameter Sendeintervall und Standby-Zeit wurden stündlich variiert (siehe Tabelle 3)

2.1 Parametrisierung vom 16.01.2015 – Standby-Zeit und Sendeintervall

Der erste Test startete am 16.01.2015 um 15:00 Uhr und endete am 17.01.2015 um 08:28 Uhr. Die beiden Abbildungen 3 und 4 zeigen die empfangenen Pakete pro Test-Szenario. Es zeigt sich, dass bei den Test Cases (vergleiche Tabelle 3) mit einem Sendeintervall von 20ms die meisten Pakete empfangen wurden.

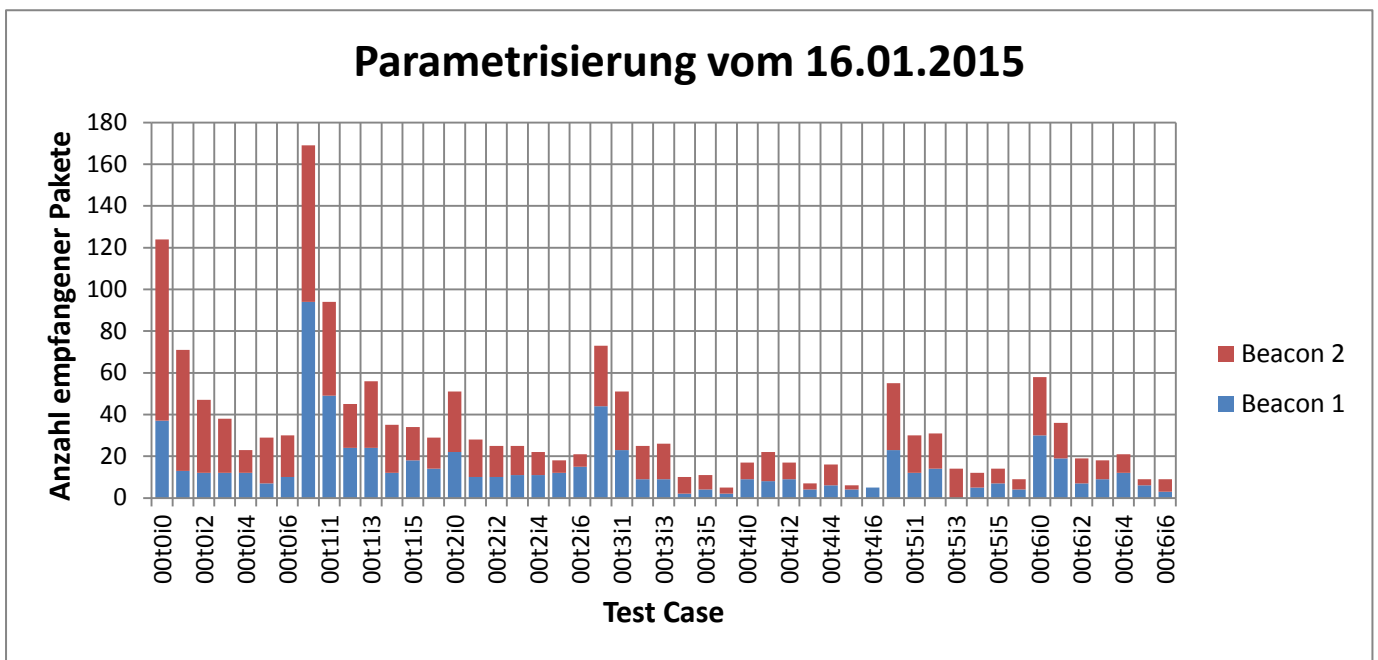


Abbildung 4: Auswertung der Logdaten (16.01.2015)

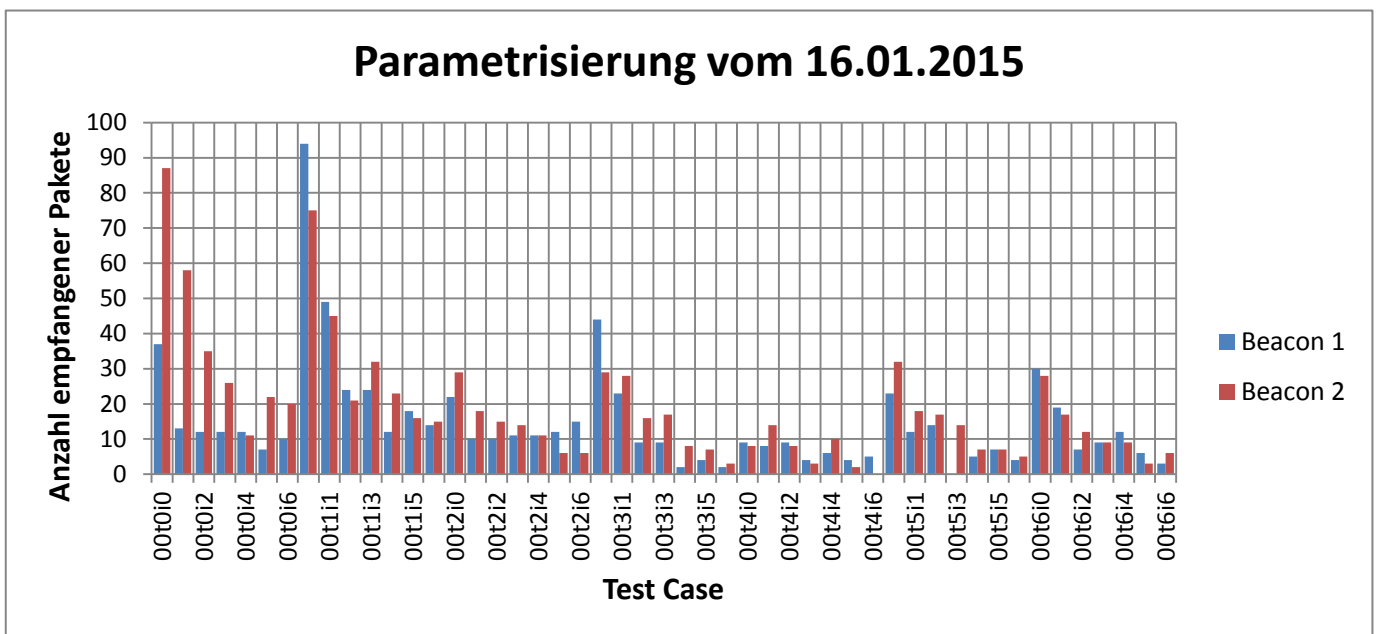


Abbildung 3: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 16.01.2015)

Da dieser Test mehr als 49 Stunden lief (nämlich 66 Stunden), wurden einige Test-Cases mehrfach durchlaufen. Dies führt zu einem erhöhten Anstieg der empfangenen Pakete in den Test Cases, die doppelt durchgeführt wurden. Die folgende Abbildung 5 zeigt die am Smartphone empfangenen Pakete sortiert nach Stunden. Es ist zu erkennen, dass der Versuch mehr als 65 Stunden lief.

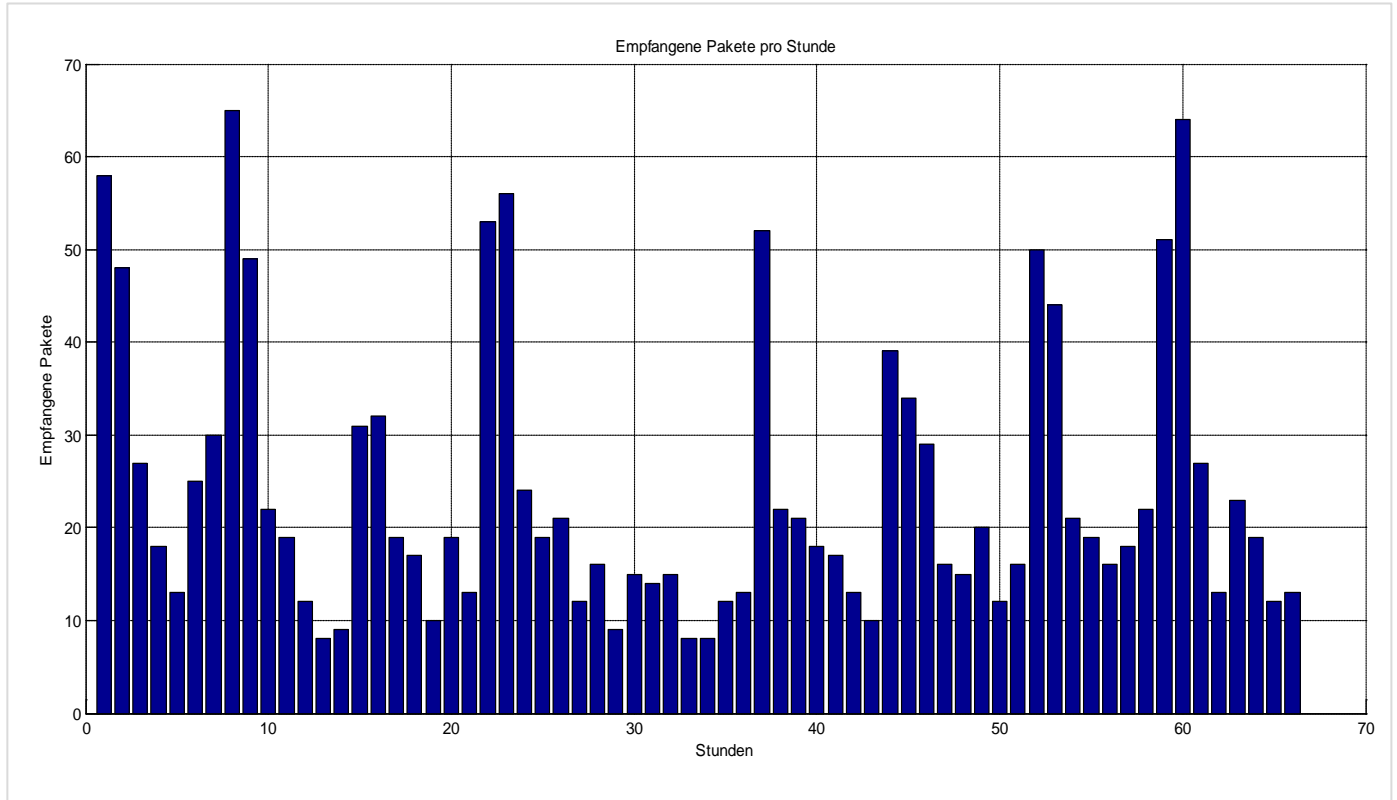


Abbildung 5: Balkendiagramm mit den empfangenen Paketen nach Stunden sortiert

Die zu erkennenden Peaks in Abbildung 5 erscheinen immer bei den Sendeeintervallen 20ms und 40ms. Bei Sendeeintervallen von 20ms oder 40ms werden im Schnitt also zwischen 40 und 75 BLE-Pakete pro Stunde empfangen. Da in diesem Test zwei Beacons eingesetzt wurden, muss die Anzahl empfangener Pakete durch zwei geteilt werden.

2.2 Parametrisierung vom 23.01.2015 – Standby-Zeit und Sendeintervall

Bei der Parametrisierung vom 23.01.2015 wurde über 90 Stunden gemessen. Da ein Testzyklus insgesamt 49 Stunden dauert, wurden auch hier einige Test-Szenarien zweimal durchlaufen. Abbildung 6 zeigt die Anzahl empfangener Pakete pro Stunde sortiert nach den jeweiligen Test Cases.

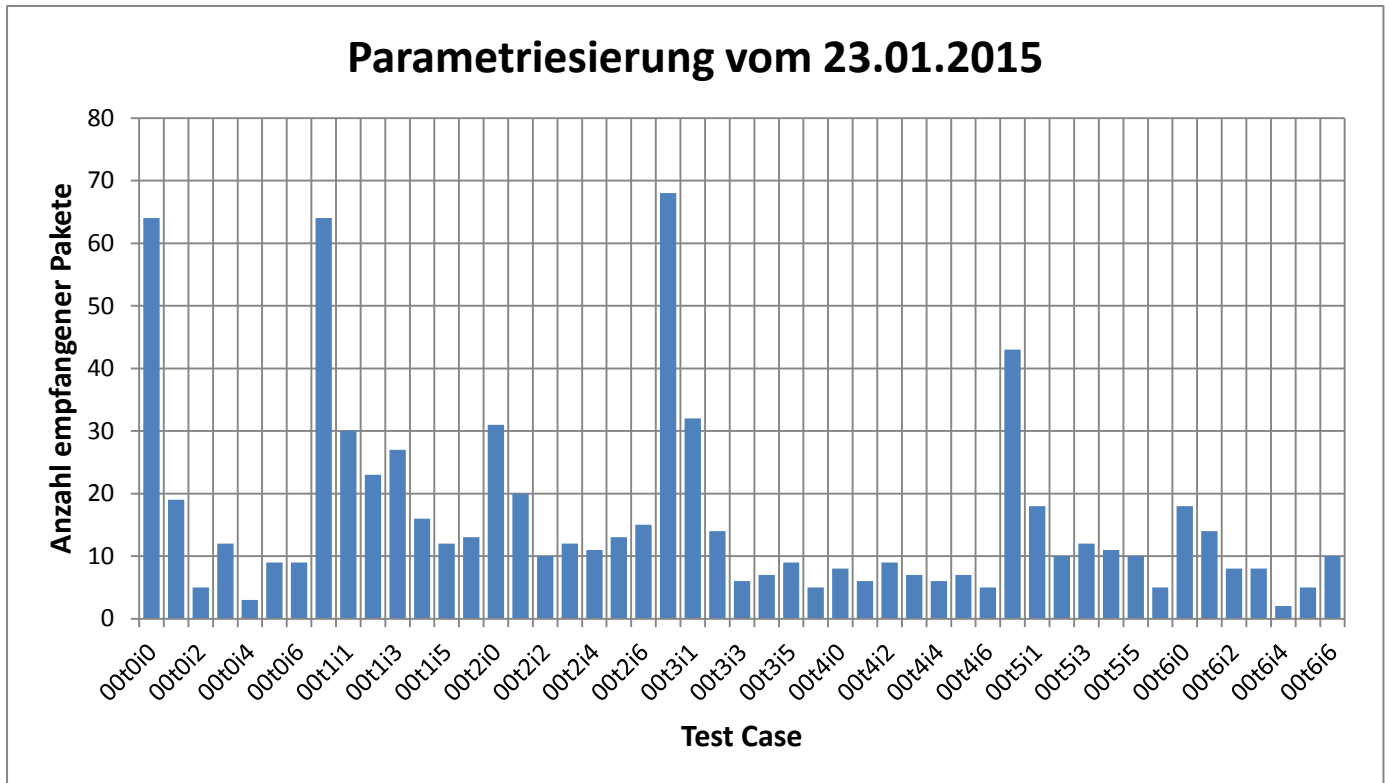


Abbildung 6: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 23.01.2015)

Interessiert man sich zum Beispiel für die Anzahl empfangener Pakete pro Stunde im Test Case 00t0i0, so muss beachtet werden, dass dieser Test Case doppelt durchgeführt wurde. Die Anzahl empfangener Pakete ist in diesem Test Case also doppelt so hoch, wie wenn das Test Szenario nur einmal durchlaufen worden wäre.

2.3 Parametrisierung vom 27.01.2015 – Standby-Zeit und Sendeintervall

Bei diesen Langzeittests waren zwei Beacons gleichzeitig aktiv. Einer der beiden wurde nach der Hälfte der Zeit ausgeschaltet. Die beiden Abbildung 7 und 8 zeigen ebenfalls maximale Ausschläge bei 20 ms Sendeintervall.

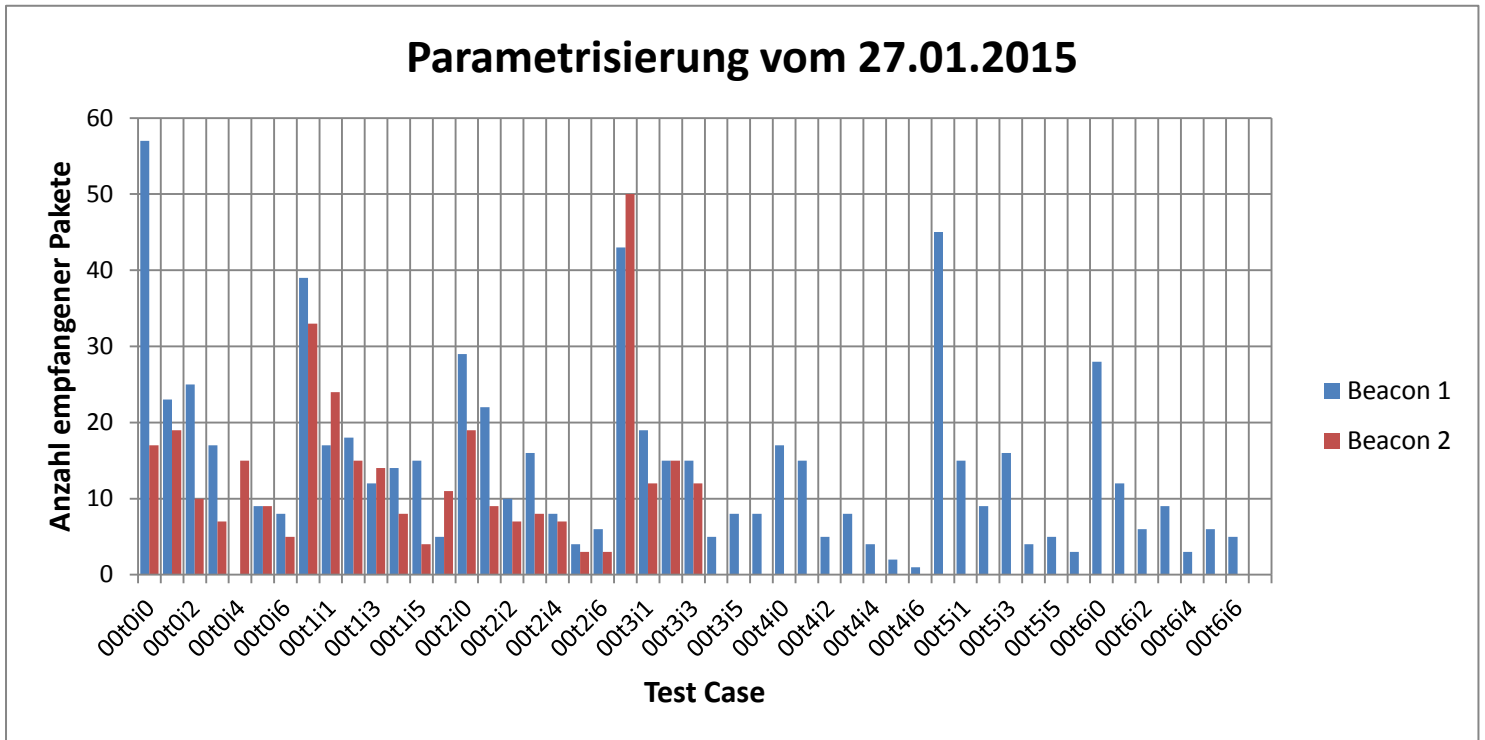


Abbildung 7: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 27.01.2015)

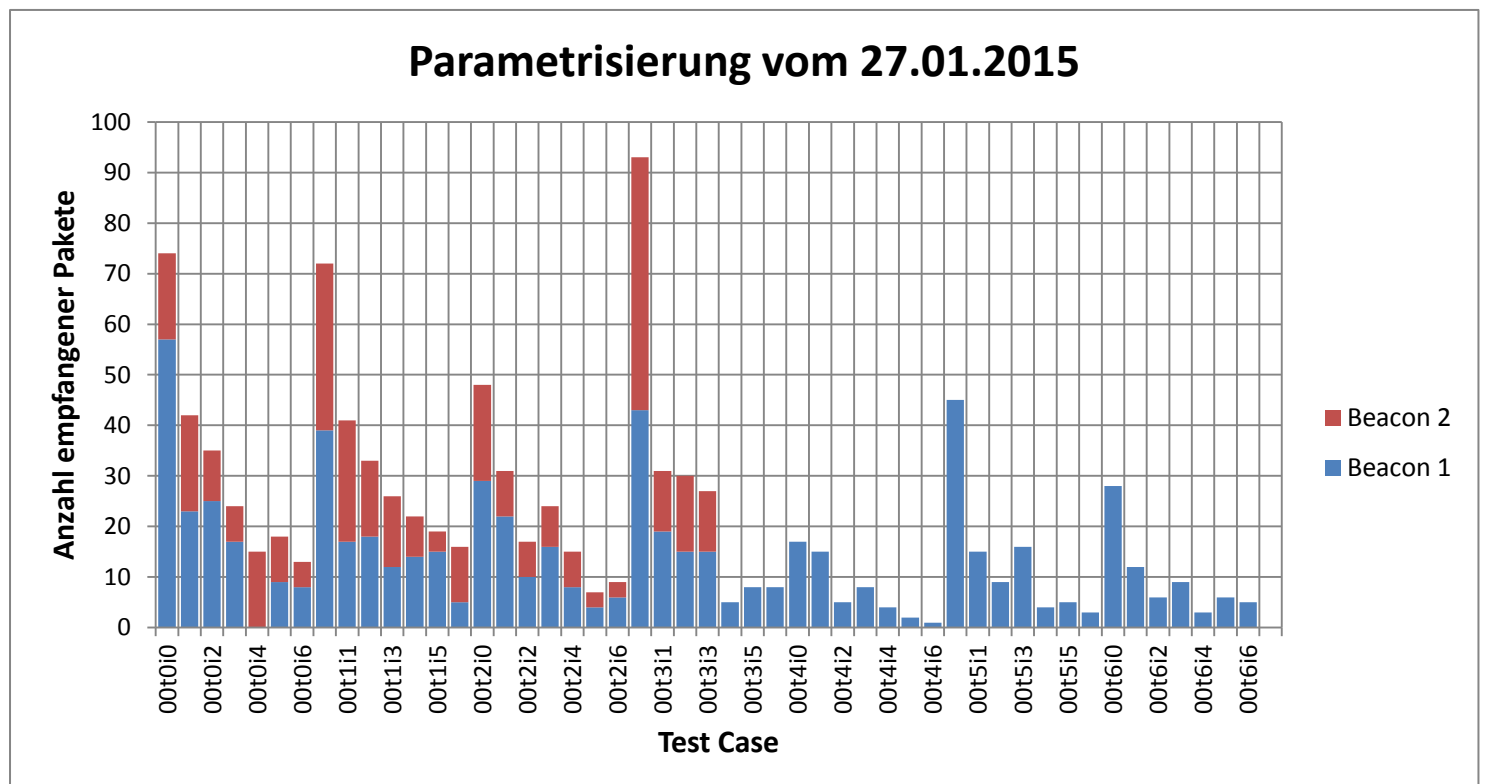


Abbildung 8: Auswertung der Logdaten (27.01.2015)

3. Zweite Testphase mit konstanter Standby-Zeit

In dieser Testphase wurde die Standby-Zeit konstant bei 15 Sekunden gehalten und nur die beiden Parameter Sendezeit und Sendeintervall variiert.

3.1 Parametrisierung vom 04.02.2015 – Sendezeit und Sendeintervall

Die Sendezeit wurde hier stündlich (von 200ms bis 800ms in 100ms Schritten) erhöht. Die Abbildung 9 zeigen einen Peak für den Test Case 00s6i2 sowie einen Peak für den Test Case 00s3i0. Der „S“ Parameter in der Bezeichnung des Test Cases steht für die Sendezeit. Nach der Formel

$$\text{Sendezeit} = 200 \text{ ms} + 100 \text{ ms} * S \quad (1)$$

lässt sich die Sendezeit berechnen. Der Peak bei 00s3i0 (s=3) wird also nach Formel (1) auf 500ms bestimmt, wobei der Peak bei 00s6i0 (s=6) auf 800ms berechnet werden kann. Das Verhältnis der beiden Peaks für die Anzahl empfangener Pakete berechnet sich zu 21:23 wobei das Verhältnis der beiden Sendezeiten bei 5:8 liegt. Vergleicht man also die aufgewendete Energie für das Senden der BLE-Pakete (nach Formel 2) in beiden Test-Szenarien mit der Anzahl empfangener Pakete, so zeigt sich, dass die effizient bei einer Sendezeit von 500 ms deutlich höher ist als bei 800 ms.

$$\text{Anzahl gesendeter Pakete pro Stunde} = \frac{3600}{\text{StandbyZeit} + \text{Sendezeit}} \cdot \frac{\text{Sendezeit}}{\text{Sendeintervall}} \quad (2)$$

Diese Werte geben einen Rückschluss auf das Verhalten des Betriebssystems. Der Scheduler scheint also der Applikation einen Zeitraum von circa 500 ms zur Verarbeitung seiner Tasks zur Verfügung zu stellen.

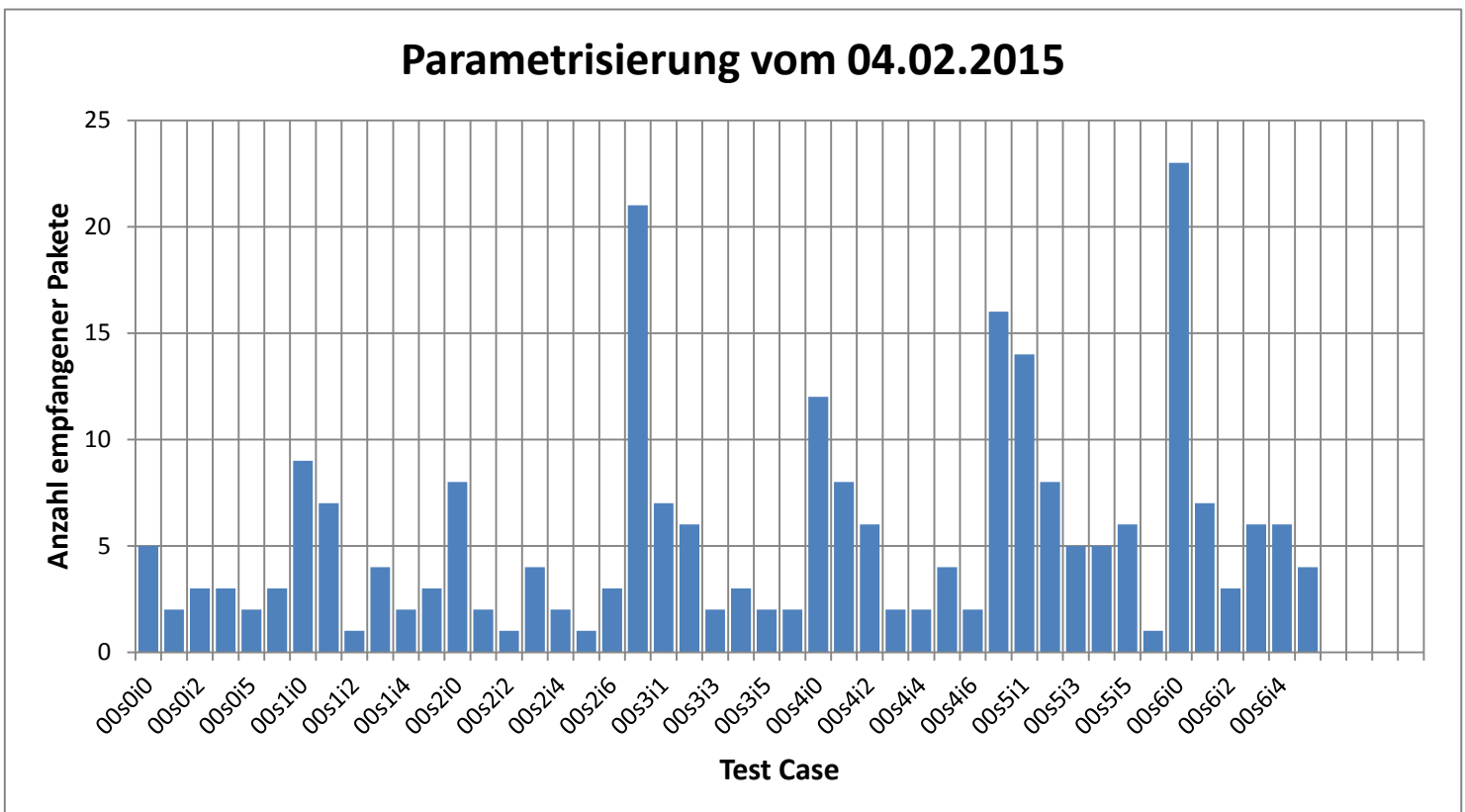


Abbildung 9: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 04.02.2015)

Die Abbildung 10 zeigt dieselben Messdaten sortiert nach den Sendeintervallen. Auch hier sind die Peaks bei 500 ms und 800 ms klar ersichtlich.

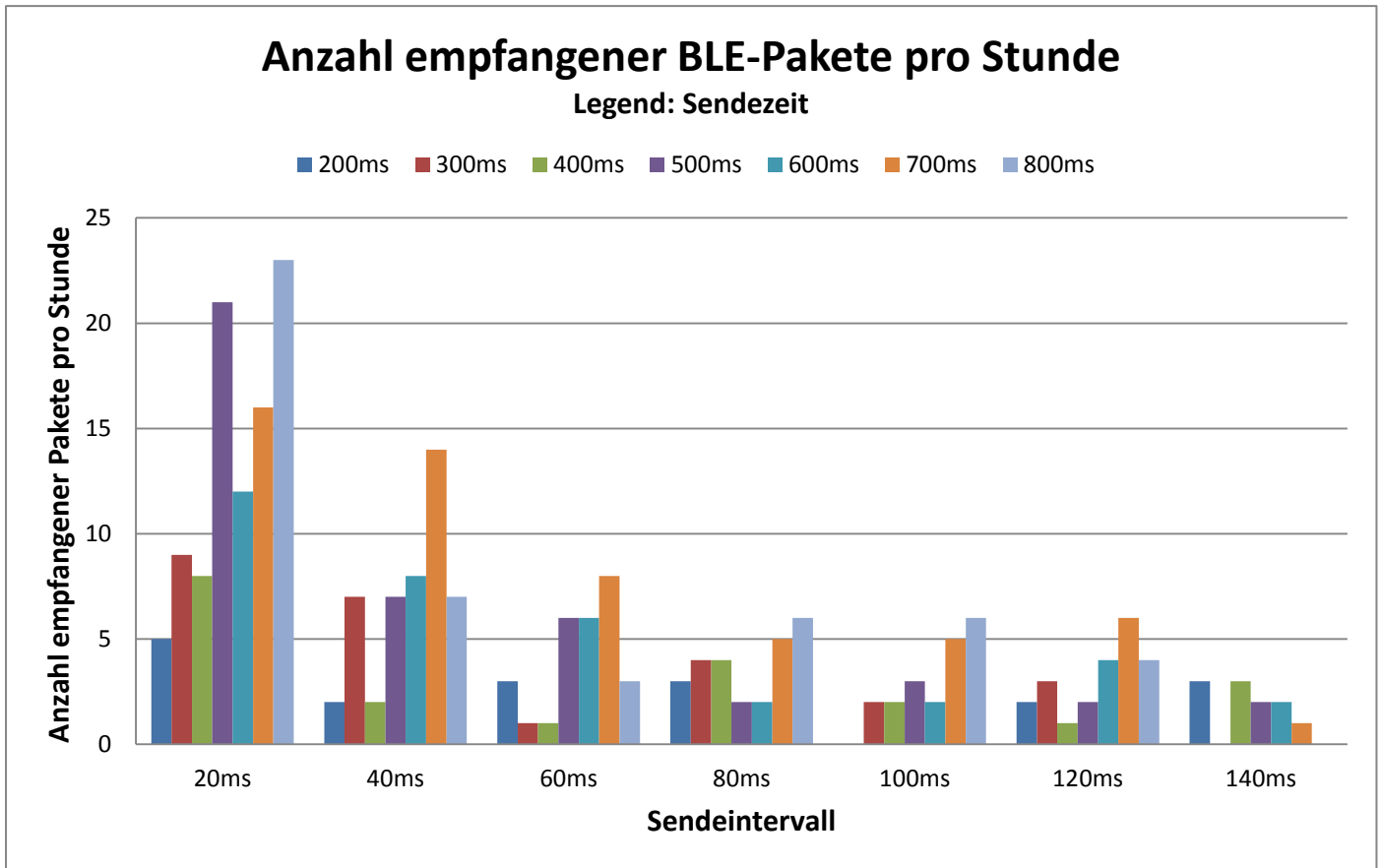


Abbildung 10: Darstellung der empfangener Pakete nach der Parametrisierung vom 04.02.2015

3.2 Parametrisierung vom 09.02.2015 – Sendezeit und Sendeintervall

Die Parametrisierung vom 09.02.2015 misst ebenfalls die Anzahl empfangener Pakete in Abhängigkeit zur Sendezeit und des Sendeintervalls. Die Messungen zeigen ein ähnliches Ergebnis wie die Messungen vom 04.02.2015. Die grössten Ausschläge aus der Abbildung 11 und Abbildung 12 zeigen sich bei 600 ms, 700 ms und 800 ms.

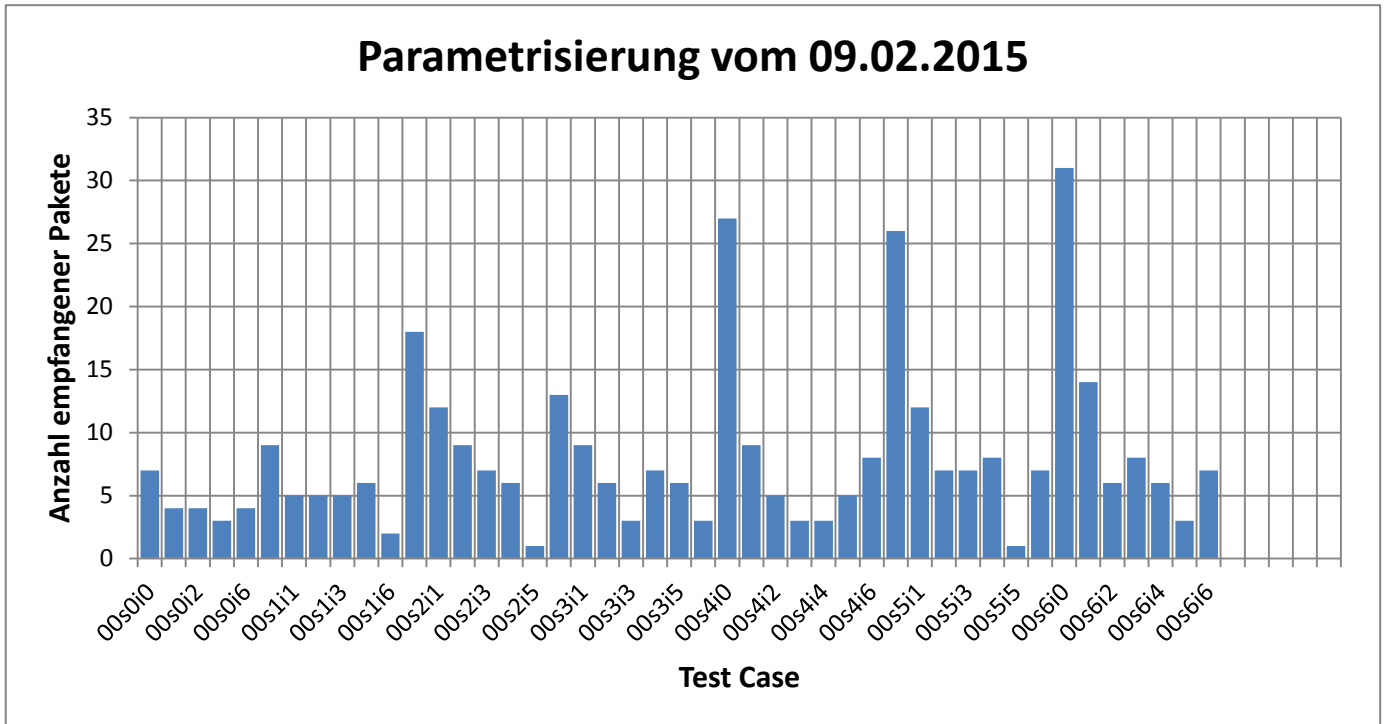


Abbildung 11: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 09.02.2015)

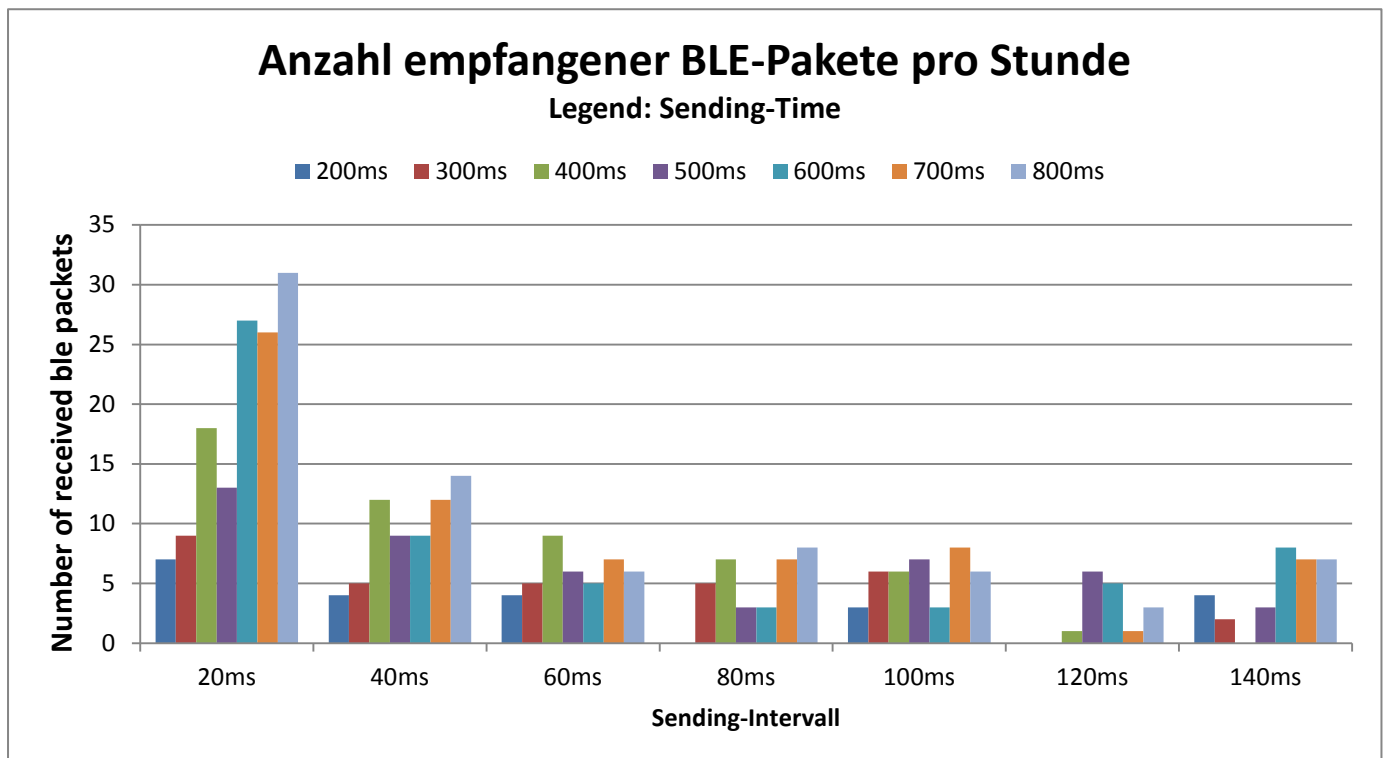


Abbildung 12: Darstellung der empfangener Pakete nach der Parametrisierung vom 09.02.2015

4. Auswertung

Die durchgeführten Messung zeigen, dass der Einfluss des Sendeintervalls und somit die Anzahl gesendeter Pakete pro Sekunde deutlich grösser ist als die Standby-Zeit oder die Sendezeit. Eine längere Standby-Zeit führt nicht direkt zu einem schlechteren Ergebnis am Empfänger. Auch eine Erhöhung der Sendezeit führt nicht zwangsläufig zu einer höheren Anzahl empfangener Pakete. Dies zeigen die Messungen vom 04.02.2015 und 09.02.2015. Ein Optimum lässt sich also bei einer Sendezeit um zwischen 500 ms und 700ms finden wobei das Sendeintervall bei 20ms gehalten werden soll.

5. Abbildungsverzeichnis

Abbildung 1: Versuchsaufbau.....	2
Abbildung 2: Die drei Sendeparameter – Standby-Zeit, Sendezeit und Sendeintervall.....	3
Abbildung 3: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 16.01.2015) ...	5
Abbildung 4: Auswertung der Logdaten (16.01.2015)	5
Abbildung 5: Balkendiagramm mit den empfangenen Paketen nach Stunden sortiert	6
Abbildung 6: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 23.01.2015) ..	7
Abbildung 7: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 27.01.2015) ..	8
Abbildung 8: Auswertung der Logdaten (27.01.2015)	8
Abbildung 9: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 04.02.2015) ...	9
Abbildung 10: Darstellung der empfangener Pakete nach der Parametrisierung vom 04.02.2015	10
Abbildung 11: Grafische Darstellung der empfangenen Pakete pro Test-Case (Messung 09.02.2015) ..	11
Abbildung 12: Darstellung der empfangener Pakete nach der Parametrisierung vom 09.02.2015	11

6. Contact

Dominique Truninger
Bachelor of Science in Electrical Engineering
Zurich University of Applied Sciences
8401 Winterthur
trug@zhaw.ch