Prof. Dr. Thomas Schmidt
HAW Hamburg, Dept. Informatik
Raum 480a, Tel.: 42875 - 8452
Email: t.schmidt@haw-hamburg.de
Web: http://inet.haw-hamburg.de/teaching

Technik & Technologie vernetzter Systeme

Teil 1: LoWPAN Networking im IoT (1. & 2. Praktikum)

Projekt:

Gemeinsames Errichten eines Low Power Lossy Networks auf der Basis des RPL Routing-Protokolls. Durchführen einer vergleichenden Analyse sowie Messungen und Belastungstests.

Vorbereitung:

Im entstehenden Internet of Things (IoT)werden leistungsschonende Funknetze erwartet, welche Knoten von eingeschränkter Leistungskraft über Gateways mit dem Internet verbinden. Für diesen Zweck wurden folgende speziell optimierte Protokolle bzw. Protokollvarianten von der IETF entwickelt:

- Neighbor Discovery Optimization – RFC 6775
- RPL Routing – RFC 6550
- Constrained Application Protocol (CoAP) – RFC 7252

Für die weitere Praktikumsarbeit müssen Sie sich mit diesen Standardprotokollen genauer vertraut machen, um ihren Einsatz in der Praxis erproben und verifizieren zu können.

Projektschritt 1: Laboreinbindung von Gateways und Sensorknoten

Binden Sie gemäß Anleitung (Szenario 1) Ihr Gateway (RPi) und Ihren Sensorknoten in das Labornetz ein. Überprüfen Sie gem. Anleitung die Funktionsfähigkeit und Erreichbarkeit. Aktivieren Sie hiernach einen Netzwerk-Sniffer und vollziehen Sie die Selbstkonfiguration nach.

Dokumentieren Sie Ihre Beobachtungen in einem Sequenzdiagramm und vergleichen Sie den ermittelten Protokollablauf mit den Spezifikationen gem. RFC 6775.

Ist die Implementierung korrekt oder können Sie Abweichungen feststellen?

Bitte dokumentieren und diskutieren Sie Ihre Ergebnisse im Protokollteil I.

Projektschritt 2: RPL Routing im Labornetz

Erweitern Sie nun Ihre Konfiguration und den Netzaufbau gemäß Anleitung (Szenario 2), so dass die Sensorknoten Teilnehmer in einem gemeinsamen RPL-Netzwerk an dem Labor-Gateway (RPL-Root) werden. Ihr RPi soll nunmehr die Rolle eines Monitors spielen, von welchem Sie alle Sensorknoten im LoWPAN beobachten können.

Nutzen Sie in dieser Konstellation einen Netzwerk-Sniffer, um die Kontrolloperationen des RPL Routingprotokolls zu analysieren.

Welche Nachrichten und Informationen werden zwischen den RPL-Knoten ausgetauscht. Wie prüft RPL dabei die Symmetrie vorhandener Links und wählt geeignete Routen aus?

Bitte dokumentieren und diskutieren Sie Ihre Ergebnisse im Protokollteil II.

Projektschritt 3: Datenverteilung und Messung

CoAP ist ein http-ähnliches, speziell für den Einsatz in LoWPANs konzipiertes Anwendungsprotokoll. Es erlaubt das Abfragen, Anlegen und Ändern von Netzwerk-Resources mittels REST-API. Auf den bereitgestellten Sensorknoten ist das IoT-Betriebssystem RIOT installiert, weiterhin läuft darauf ein CoAP-Server, welchen Sie gem. Anleitung für ihre Analysen und Messungen verwenden können. Sie können diverse Sensordaten auslesen und die LED auf den Sensorknoten manipulieren, beobachten Sie dabei die Datenverteilung in verschieden Szenarien und führen Sie Messungen zur Untersuchung von Zuverlässigkeit sowie Zeitverhalten durch.

Konzipieren Sie für die Analyse ein geeignetes Vorgehen. Dokumentieren Sie Ihre Ergebnisse in Form von Performance-Graphen (Achsenbeschriftungen mit Einheiten!) unter aussagekräftiger Erläuterung.

Bitte dokumentieren und diskutieren Sie Ihre Ergebnisse im Protokollteil III.

Bitte senden Sie (gruppenweise) Ihr Protokoll per Email parallel an

sebastian.meiling@haw-hamburg.de
t.schmidt@haw-hamburg.de

Deadline: 17. November 2018

Anleitung zum TTvS-Praktikum: Laborversuch 1

Sebastian Meiling (sebastian.meiling@haw-hamburg.de)
Prof.Dr. Thomas Schmidt (t.schmidt@haw-hamburg.de)

WS 2019/20

1 Problem- und Zielstellung

Ziel des ersten Laborversuchs im TTvS-Praktikum ist der Aufbau und Betrieb eines *Low power Wireless Personal Area Network* (LoWPAN) auf Basis des Funkstandards IEEE 802.15.4. Aufgrund eingeschränkter Hardware-Ressourcen erfordert dies den Einsatz optimierter Standard-Protokolle, Software-Tools und Technologien, die speziell für den Einsatz im Internet der Dinge (Internet of Things, IoT) geeignet sind, dazu zählen u.a. 6LoWPAN [1], RPL [2] und CoAP [3]. Die nachfolgenden Anleitungen beschreiben die Versuchsaufbauten zur Untersuchung der 6LoWPAN Neighbor Discovery (ND) und des RPL Routing sowie zur Leistungsmessung in einem drahtlosen Sensor-Netzwerk. Dafür steht Ihnen ein IoT-Kit zur Verfügung, dieses enthält: einen Raspberry Pi als Gateway, verschiedene Sensor-Knoten, sowie Kabel und Zubehör.

2 Szenario 1: Neighbor Discovery

Die folgende Anleitung beschreibt den Versuchsaufbau (s. Abb. 1) zur Untersuchung der 6LoWPAN Neighbor Discovery bei Verwendung von link-local und unique-local IPv6-Adressen (LLA bzw. ULA).

Wichtig: Verwenden Sie die Befehle in Abschnitt 4 und beachten Sie die Hinweise in Abschnitt 5.

Abbildung 1: Aufbau eines minimalen Sensor-Netzwerks ohne Routing.

1. Starten Sie Ihren Arbeitsplatz-PC und booten Sie in das *BRV-Special*-Linux.
2. Verbinden Sie den Raspberry Pi mit dem beiliegenden Netzwerkkabel zum Arbeitsplatz-PC, nutzen Sie das freie Netzwerk-Interface (unterhalb der Grafikkarte).
3. Setzen Sie auf dem Interface `eth1` eine IPv6-Adresse mit dem Präfix `fd16:abcd:ef<XY>:2::/64` und aktivieren Sie es. *Hinweis:* ersetzen Sie `<XY>` mit der Nummer Ihres IoT-Kits und *erweitern* Sie den resultierenden Präfix zu einer gültigen IPv6-Adresse.
4. Verbinden Sie einen der beiliegenden Sensor-Knoten über Micro-USB mit Strom, verwenden Sie die linke USB-Buchse am Gerät - halten Sie den Sensor-Knoten wie in Abb. 1 dargestellt.
5. Öffnen Sie auf dem Arbeitsplatz-PC eine Shell und verbinden Sie sich per SSH auf den Raspberry Pi, die IP-Adresse ist auf dem Gerät vermerkt.
6. Versuchen Sie den Sensor-Knoten über dessen *link-lokale* IPv6-Adresse (siehe Gerät) mittels `ping6` zu erreichen. Bei Fehlern prüfen Sie die verwendeten Befehle und IP-Adressen.
7. Öffnen Sie eine weitere Shell und eine SSH-Verbindung zum Raspberry Pi, starten Sie `radvd` (siehe 4.2).
8. `radvd` weist dem Sensor-Knoten einen ULA-Präfix `fd16:abcd:ef<XY>:3::/64` zu. Testen Sie mittels `ping6`, ob Sie den Sensor-Knoten vom Raspberry Pi über dessen ULA erreichen können (s. Abb. 1).
9. Richten Sie mit `ip route add ...` auf dem Arbeitsplatz-PC eine Route zum Sensor-Knoten über den Raspberry Pi als Gateway ein. Testen Sie anschließend vom PC die Verbindung zum Sensor-Knoten.

Wenn Sie den Aufbau erfolgreich abgeschlossen und getestet haben, können Sie mittels Netzwerk-Sniffer und CoAP-Abfragen ihre Untersuchungen beginnen.

3 Szenario 2 – RPL Routing

Die folgende Anleitung beschreibt den Versuchsaufbau (s. Abb. 2) zur Untersuchung von mehr-Hop Verbindungen im IoT auf Basis des RPL Routing Protokolls.

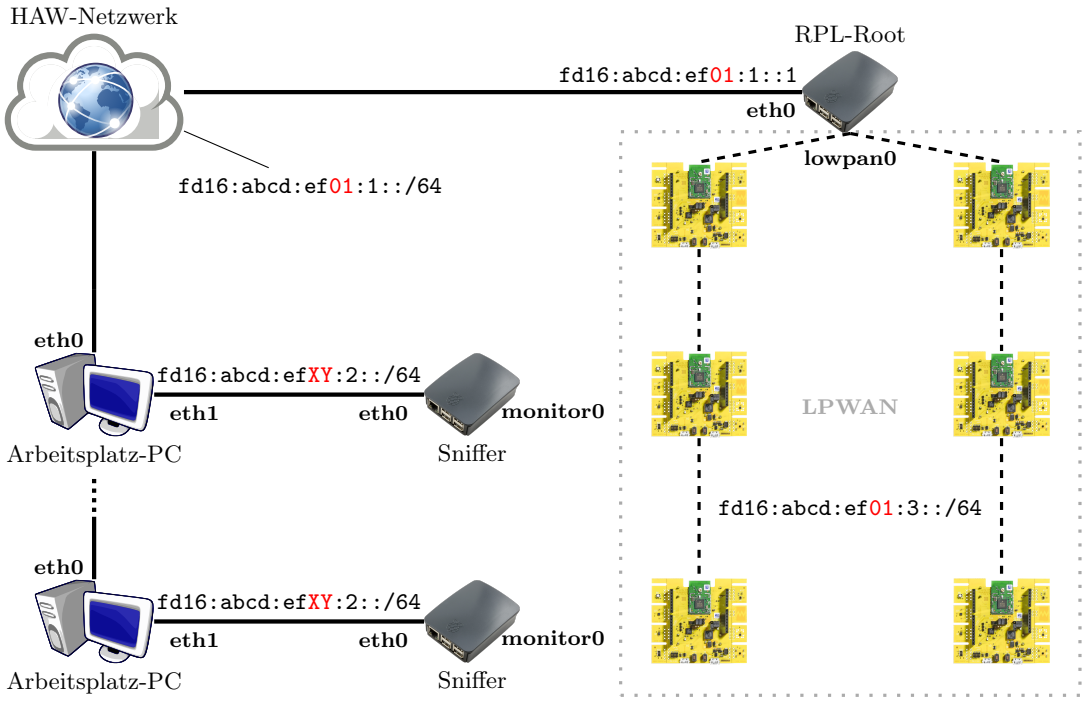

Abbildung 2: Aufbau eines Sensor-Netzwerks mit RPL Routing.

1. Erweitern Sie die Konfiguration des Netzwerk-Interface `eth0` auf dem Arbeitsplatz-PC, fügen Sie dazu eine IPv6-Adresse nach dem Schema `fd16:abcd:ef01:1::<XY>/64` hinzu (`<XY>` ist mit der Nummer ihres IoT-Kits zu ersetzen).
2. Testen Sie die Verbindung zum Raspberry Pi Gateway und RPL-Root-Knoten mittels Ping, die IPv6-Adresse lautet `fd16:abcd:ef01:1::1`.
3. Richten Sie anschließend mit dem Befehl `ip route add ...` auf dem Arbeitsplatz-PC eine statische Route für den Präfix `fd16:abcd:ef01:3::/64` über den zentralen Raspberry Pi (RPL-Root) als Gateway ein.
4. Verbinden Sie einen der beiliegenden Sensor-Knoten über Micro-USB mit Strom, verwenden Sie die linke USB-Buchse am Gerät. Drücken Sie auf dem Sensor-Knoten den rechten Taster *einmal*, um die Kanaleinstellung auf Szenario 2 zu ändern - die LED blinkt 4x schnell *rot* zur Bestätigung.
5. Testen Sie mittels Ping auf dem Arbeitsplatz die Verbindung zu ihrem Sensor-Knoten, dieser erhält vom RPL-Root eine IPv6-Adresse mit dem Präfix `fd16:abcd:ef01:3::/64`.
6. Verbinden Sie den Raspberry Pi ihres IoT-Kits wie in Abschnitt 2 beschrieben und öffnen Sie eine SSH-Verbindung.
7. Legen Sie auf dem Raspberry Pi ein Monitoring-Interface an, siehe Listing 1. Das Monitoring Interface erlaubt es den gesamten Datenverkehr im RPL Netzwerk zu beobachten.

Listing 1: Anlegen eines Monitoring-Interface über `systemd`

```
1 sudo systemctl stop lowpan
2 sudo systemctl start lowpan_monitor
```

4 Befehlsreferenz

4.1 Das `ip` Konfigurationstool

Das Tool **`ip`** [4] erlaubt u.a. die Konfiguration von IP-Interfaces und das Manipulieren der Routing-Tabelle des Linux-Kernels. Beim Setzen von IP-Adressen oder Anlegen statischer Routen ist unbedingt die korrekte Netzmaske bzw. Länge des IP-Präfixes anzugeben, da sonst als Default bei IPv4 `32` und bei IPv6 `128` verwendet wird. Folgende Befehle werden für den Laborversuch benötigt:

Listing 2: Übersicht

```
1 ip link show
2 ip link set dev <Interface> up
3 ip addr show
4 ip addr add <IP>/<Mask> dev <Interface>
5 ip [-6] route show
6 ip route add <IP>/<Mask> via <Gateway-IP>
```

1. Anzeigen aller verfügbaren Netzwerk-Interfaces
2. Ein Netzwerk-Interface aktivieren
3. Anzeigen aller IP-Adressen aller Netzwerk-Interfaces
4. Einem Netzwerk-Interface eine IP-Adresse zuweisen
5. Anzeigen aller Routen, für IPv6 Routen Option `-6`
6. Anlegen einer statischen Route über einen Gateway

4.2 Der Router Advertisement Daemon `radvd`

Der **`radvd`** ermöglicht die Verteilung von IP-Präfixen zur Auto-Konfiguration von IPv6-Adressen in einem Netzwerk, er stellt jedoch kein Routing-Protokoll bereit. Hinweis: das Programm benötigt **`root`** Rechte.

Listing 3: Übersicht

```
1 radvd
2 radvd -d 5 -m stderr -n
```

1. Startet **`radvd`** als Daemon im Hintergrund
2. Startet **`radvd`** im Vordergrund, mit Debug-Information und Ausgabe in der Shell

4.3 Netzwerk-Sniffer

Zur Analyse des Netzwerk-Verkehrs stehen Ihnen die Tools **`tshark`** (Shell) und **`wireshark`** (mit GUI) zur Verfügung. Diese können Sie wie folgt verwenden:

Listing 4: Übersicht

```
1 tshark -i <Interface> -f <Filter>
2 ssh pi@<IP> 'sudo dumpcap -P -i <Interface> -w -' | wireshark -k -i -
```

1. Startet einen Netzwerk-Sniffer in der Shell. Auf dem angegeben IP-Interface und mit dem (optionalen) Pcap-Filter [5]. z.B. `-f 'ip proto icmpv6'` oder `-f 'udp port 5683'`.
2. Startet einen Netzwerk-Sniffer auf dem Raspberry Pi und Wireshark auf dem Arbeitsplatzrechner, die Ausgabe wird per SSH an Wireshark weitergeleitet. Als Parameter muss ein Interface angegeben werden, optional kann auch hier ein PCAP-Filter angegeben werden.

4.4 Der `coap-client`

Die C-Bibliothek `libcoap` [6] enthält das Software-Tool `coap-client`, welches sich zum Testen von CoAP-Abfragen eignet und in der Shell auf dem Raspberry Pi ausgeführt werden kann. Zu beachten ist dabei, dass IPv6-Adressen in eckigen Klammern angegeben werden müssen, z.B. `[ab:cd:ef::1]`.

Listing 5: Übersicht

```
1 coap-client -m GET coap://<IP>/<Ressource>
2 echo <Wert> | coap-client -m PUT coap://<IP>/<Ressource> -f -
```

1. Abrufen einer Resource mittels `GET`.
2. Anlegen oder ändern einer Resource mittels `PUT`

5 Zusätzliche Informationen

- Beim Start des Arbeitsplatzrechners müssen Sie das Passwort für den Nutzer `networker` ändern, in der Eingabemaske müssen Sie als *erstes* das ursprüngliche Passwort eingeben und danach 2-mal ein selbstgewähltes (temporäres) Passwort.
- Das Password des Nutzers `pi` für das SSH-Login auf den Raspberry Pi lautet `TTvSprak`.
- Für die Konfiguration von IP-Interfaces und Routen sind `root` Rechte notwendig, diese können mittels `sudo <Befehl>` erlangt werden. Die Nutzer `networker` (PC) und `pi` (RPi) sind `sudo` berechtigt; auf dem Raspberry Pi muss kein Passwort eingegeben werden.
- Die in diesem Laborversuch verwendet IP-Präfixe leiten sich vom Präfix `fd16:abcd:ef::/40` und von der Nummer des IoT-Kits (`XY` $\in$ 01..25) sowie ihrer Verwendung ab. Ersetzen sie im folgenden `<XY>` jeweils gegen die Nummer des IoT-Kits (siehe auch Abb. 1 und 2):
  1. `fd16:abcd:ef01:1::/64`: Zur Verbindung des Arbeitsplatz-PCs über das Netzwerk-Interface `eth0` mit zentralen RPi-Gateway und RPL-Root.
  2. `fd16:abcd:ef<XY>:2::/64`: Zur Verbindung des PCs über `eth1` mit dem RPi des IoT-Kits.
  3. `fd16:abcd:ef<XY>:3::/64`: Zur Verbindung des Raspberry Pi mit den Sensor-Knoten.
- Bei link-local IPv6-Adressen (`fe80::/64`) muss unter Linux das IP-Interface mit angegeben werden, z.B. `fe80::ab:cd:ef:01%lowpan0`.
- Für die beiden Szenarien sind separate (Funk-)Konfigurationen vorgesehen, welche über den rechten Taster am Sensor-Knoten gewechselt werden können. Die LED blinkt *2x langsam* rot für Szenario 1 sowie *4x schnell* für Szenario 2.
- Die Sensor-Knoten erlauben die Abfrage vorhandenen CoAP-Ressourcen unter `.well-known/core` [7]. Weiterhin kann die RGB-LED mittels `PUT` manipuliert werden: mögliche Optionen sind `0|1` für AN oder AUS sowie `r|g|b` zum Schalten der einzelnen Farbkanäle. Verwenden Sie zum Erzeugen von CoAP-Anfragen den `coap-client` (s. 4.4).
- Die Software auf den Sensor-Knoten basiert auf dem Open Source IoT-Betriebsystem RIOT, weitere Information unter `https://riot-os.org` und `https://github.com/RIOT-OS`.

Literatur

[1] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," IETF, RFC 4944, September 2007.

[2] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," IETF, RFC 6550, March 2012.

[3] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF, RFC 7252, June 2014.

[4] Linux man page, "ip – show manipulate routing, devices, policy routing and tunnels," 2011. [Online]. Available: http://man7.org/linux/man-pages/man8/ip.8.html

[5] ——, "pcap-filter – packet filter syntax," 2015. [Online]. Available: http://www.tcpdump.org/manpages/pcap-filter.7.html

[6] O. Bergmann, "libcoap: C-Implementation of CoAP," 2016. [Online]. Available: https://libcoap.net/

[7] Z. Shelby, "Constrained RESTful Environments (CoRE) Link Format," IETF, RFC 6690, August 2012.