



HOPANDFOUND

RIOT IM INTERNET OF THINGS

THORBEN SCHNIRPEL , HAUKE SONDERMANN , DENNIS KIRSCH , MARCO KOSCIELNY, ARNE RINGLEB



GLIEDERUNG

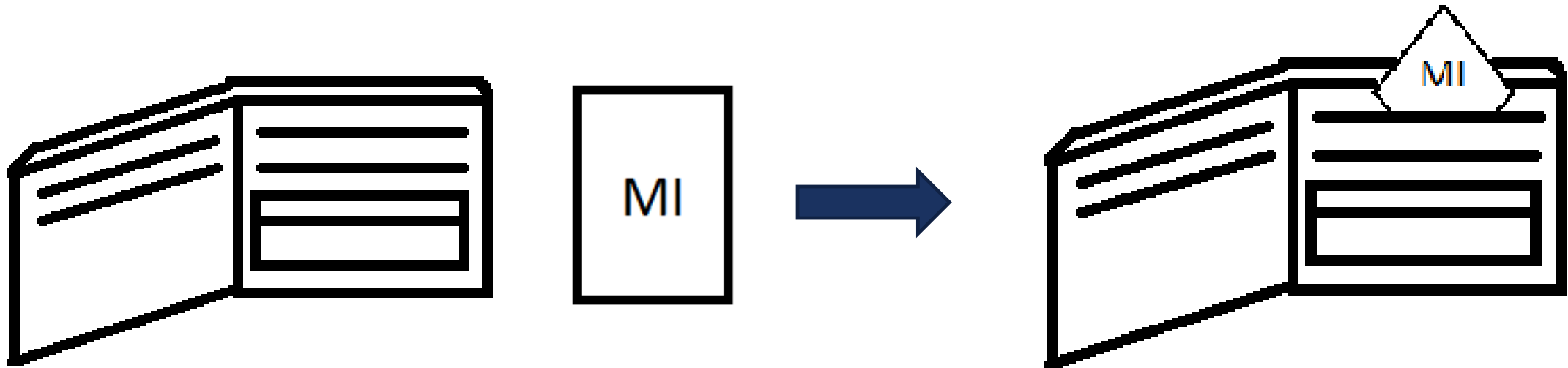
1. Grundidee von HopAndFound
2. Lösungsansatz
3. Genutzte Hardware
4. Arbeitsbereiche
5. Präsentationsaufbau
6. Nicht umgesetzte Ideen
7. Gelerntes
8. Mögliche Verbesserungen

I. GRUNDIDEE VON HOPANDFOUND

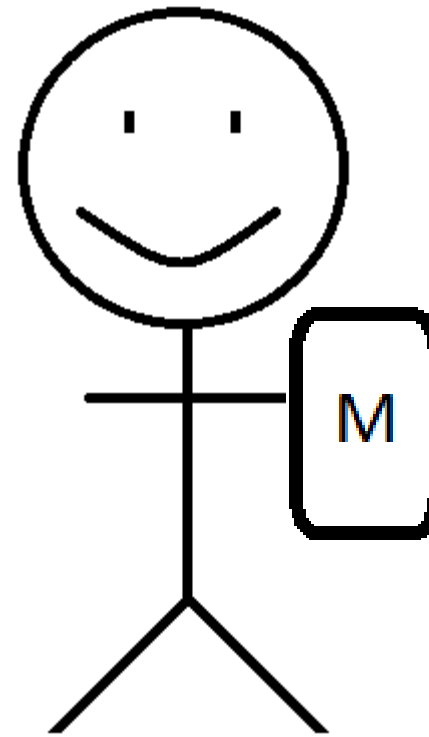
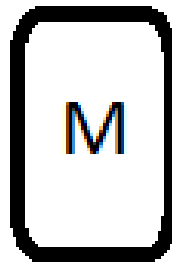
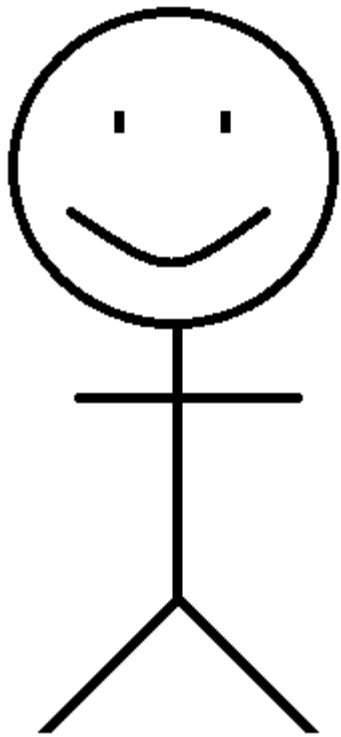
HopAndFound

- ist ein Diebstahlalarm
- soll gestohlene Gegenstände wiederfinden
- basiert auf einem stationären Netz aus Verbindungsknoten (Multihop)
- besteht aus 3 Komponenten
 - Monitored Item (MI) : Gerät, welches dem zu überwachenden Gegenstand beigelegt wird
 - Monitor (M) : Wird vom User mitgeführt und gibt Informationen über das Monitored Item
 - Node (N) : stationäre Verbindungsknoten zur Weiterleitung von Nachrichten und zur Positionsermittlung

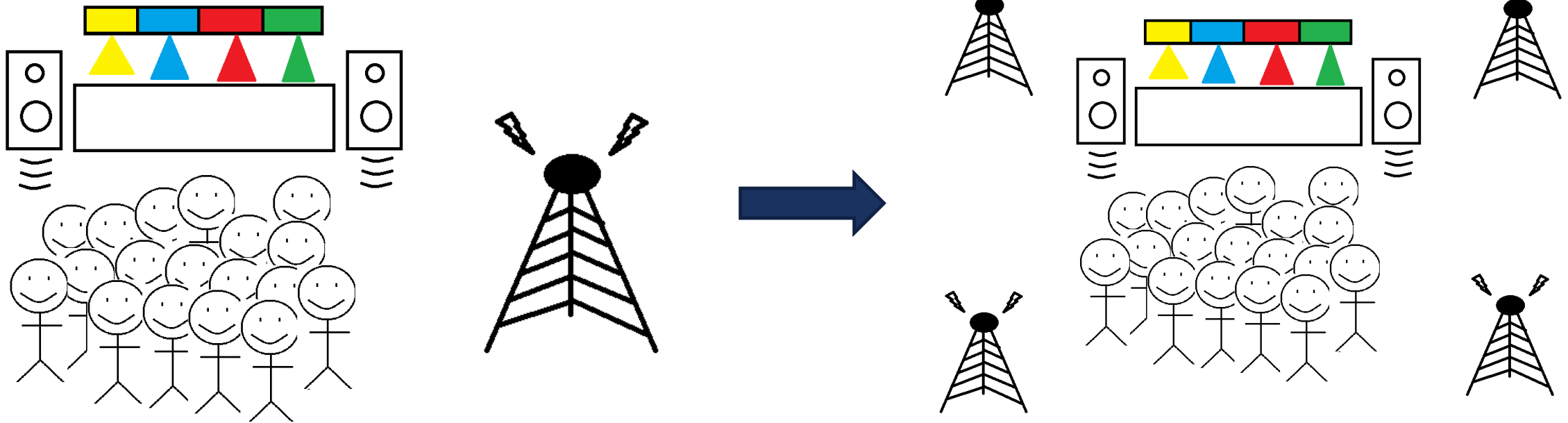
I. GRUNDIDEE VON HOPANDFOUND



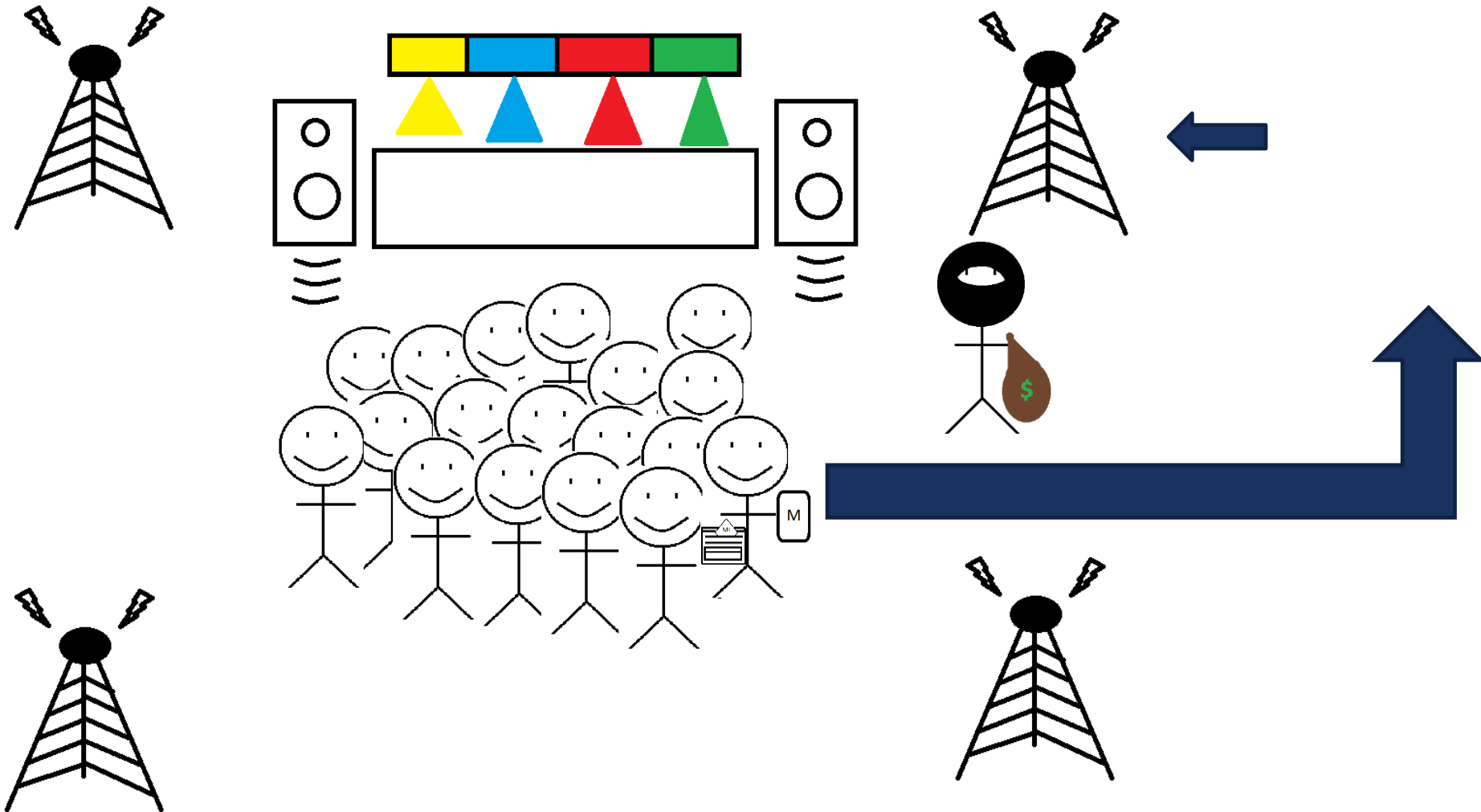
I. GRUNDIDEE VON HOPANDFOUND



I. GRUNDIDEE VON HOPANDFOUND



I. GRUNDIDEE VON HOPANDFOUND



2. LÖSUNGSANSATZ

- MI und M halten Verbindung mit Heartbeat über 6LoWPAN (heartbeat)
- Bei Abbruch des Heartbeats sendet das MI einen Multicast an alle Nodes zur Lokalisierung (localization_request)
- Die Nodes antworten mit ihrer ID (localization_reply)
- Das MI sendet einen Hilferuf mit den IDs der Nodes, die es erreicht hat (call_for_help)
- Die Nodes leiten die Nachricht per Multicast zum M weiter
- M zeigt dem Benutzer die Node IDs auf einem Display an

3. GENUTZTE HARDWARE

- MI:
 - 1x Phytec kw2x Board
- M:
 - 1x Phytec kw2x Board
 - 1x Nokia Display (pcd8544)
- Nodes:
 - 6x Phytec kw2x Board

4. ARBEITSBEREICHE

Erste Phase - Projektplanung - Projektidee bis MSI (alle) :

- Ideenfindung
- Präsentation erstellen
- GIT-Struktur
- Art der Kommunikation

Zweite Phase - Basis Implementierung bis MS2 (alle) :

- Implementierung - Kommunikationsschnittstellen
- Implementierung - Timer
- Implementierung - Dispatcher
- Implementierung - Handler

Dritte Phase wurde wie folgt aufgeteilt.

4. ARBEITSBEREICHE

Dritte Phase

Button/LED-Belegung (Hauke)

- Button
 - Node
 - Button sendet `localization_request`
 - Monitor
 - Button stellt heartbeat ein / aus
- LED
 - Node
 - LED blinkt blau beim Empfang einer `localization_request`
 - Monitor
 - LED blinkt rot, wenn `call_for_help` empfangen wird
 - LED leuchtet blau, wenn das heartbeat-Senden deaktiviert ist
 - MonitoredItem
 - LED blinkt rot, wenn `call_for_help` gesendet wird

Dokumentation (Hauke)

- GitHub readme
- GitHub Wiki

Timer (Hauke)

- Benötigt für : `heartbeat`, `bind`, `call_for_help` etc.

Dispatcher (Hauke)

4. ARBEITSBEREICHE

Routing (Arne, Marco)

- Epidemic (z.B. Gossip) Routing
 - Funktionsweise verstehen und Entscheidung treffen
 - Konzept entwickeln
- Für Selbstentwicklung entschieden
 - Minimum-Hop Routing
 - Zyklisch aktualisierte Routing Table des gesamten Netzes
 - Update bei Wegfallen oder Neuauftreten einer Node/Monitor bzw. attraktiven Verbindung
- Lauffähig umsetzen testen
 - Routing zwischen Nodes
 - Gezieltes Routing ab dem Monitored Item

4. ARBEITSBEREICHE

Destination

Hops

Next-Hop

Expiration Time

- Speicherung der günstigsten Route zu jedem bekannten Host
- Gültigkeitsverfall eines Eintrags (Expiration Time) nach 30s
- Zyklische Aktualisierung alle 9s
 - Weggefallene und neue Routen werden mitgeteilt
 - Zurücksetzen der Expiration Time
- Getriggerte Aktualisierung
 - Timer für zyklische Aktualisierung wird zurückgesetzt
 - Bei Änderung/Wegfallen/Hinzufügen einer Route
 - Zurücksetzen der Expiration Time

Die Routing Table:

```
typedef struct __attribute__((packed)) {  
    ipv6_addr_t ip_addr;  
    uint8_t hops;  
    ipv6_addr_t next_hop_addr;  
    uint32_t exp_time;  
} routing_tbl_t;
```

Update-Package:

```
typedef struct __attribute__((packed)) {  
    uint8_t type;  
    ipv6_addr_t source_addr;  
    routing_tbl_t routing_tbl[MAX_DEVICES];  
} update_t;
```

NEUER LÖSUNGSANSATZ MIT ROUTING

- MI und M halten Verbindung mit Heartbeat über 6LoWPAN (heartbeat) → kein Routing
- M nutzt Routing um sich bekanntzumachen, wertet empfangene Routing Pakete allerdings nicht aus
- Bei Abbruch des Heartbeats sendet das MI einen Multicast an alle Nodes zur Lokalisierung (localization_request)
- Die Nodes antworten mit ihrer ID und der kürzesten Route zum Monitor, sofern bekannt (localization_reply)
- Das MI sendet einen Hilferuf mit den IDs der Nodes, die es erreicht hat per Unicast über Node an M (call_for_help)
- Die Nodes leiten die Nachricht ~~per Multicast~~ per Unicast gezielt zum M weiter
- M zeigt dem Benutzer die Node IDs auf einem Display an
- Zyklisches Senden der Routing Table alle 9 sek
- Getriggertes Senden der RT falls Aktualisierungen gemacht wurden, Sendezyklus wird zurückgesetzt

Desti.	Hops	Next-Hop	Exp. Time
fe80::1	0	fe80::1	-

Desti.	Hops	Next-Hop	Exp. Time
fe80::2	0	fe80::2	-

Routing Update zyklisch

Desti.	Hops	Next-Hop	Exp. Time
fe80::3	0	fe80::3	-

Desti.	Hops	Next-Hop	Exp. Time
fe80::5	0	fe80::5	-

Desti.	Hops	Next-Hop	Exp. Time
fe80::4	0	fe80::4	-



update()
(Multicast)
zyklisch

Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-



Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30000



Routing Update zyklisch

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-



change

Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000



Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-



Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30000

Routing Update gettriggert



Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-



update()
(Multicast)
gettriggert

update()
(Multicast)
gettriggert

Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-



Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000



Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-
fe80::4	1	fe80::4	30005
fe80::2	1	fe80::2	30005

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30000

Routing Update gettriggert

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005

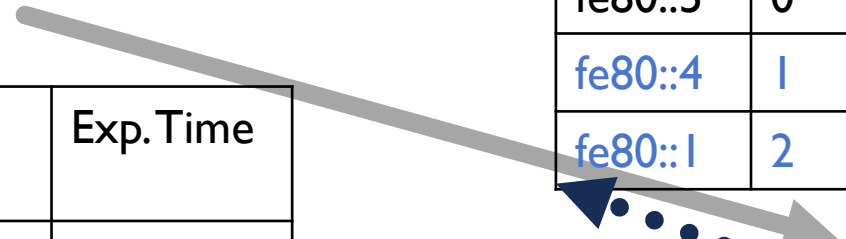
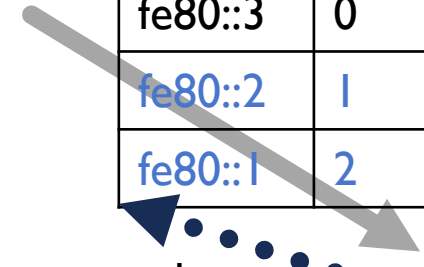
Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005

Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000

change

change

change



Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-
fe80::4	1	fe80::4	30005
fe80::2	1	fe80::2	30005

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30000

Routing Update gettriggert

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005

Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005

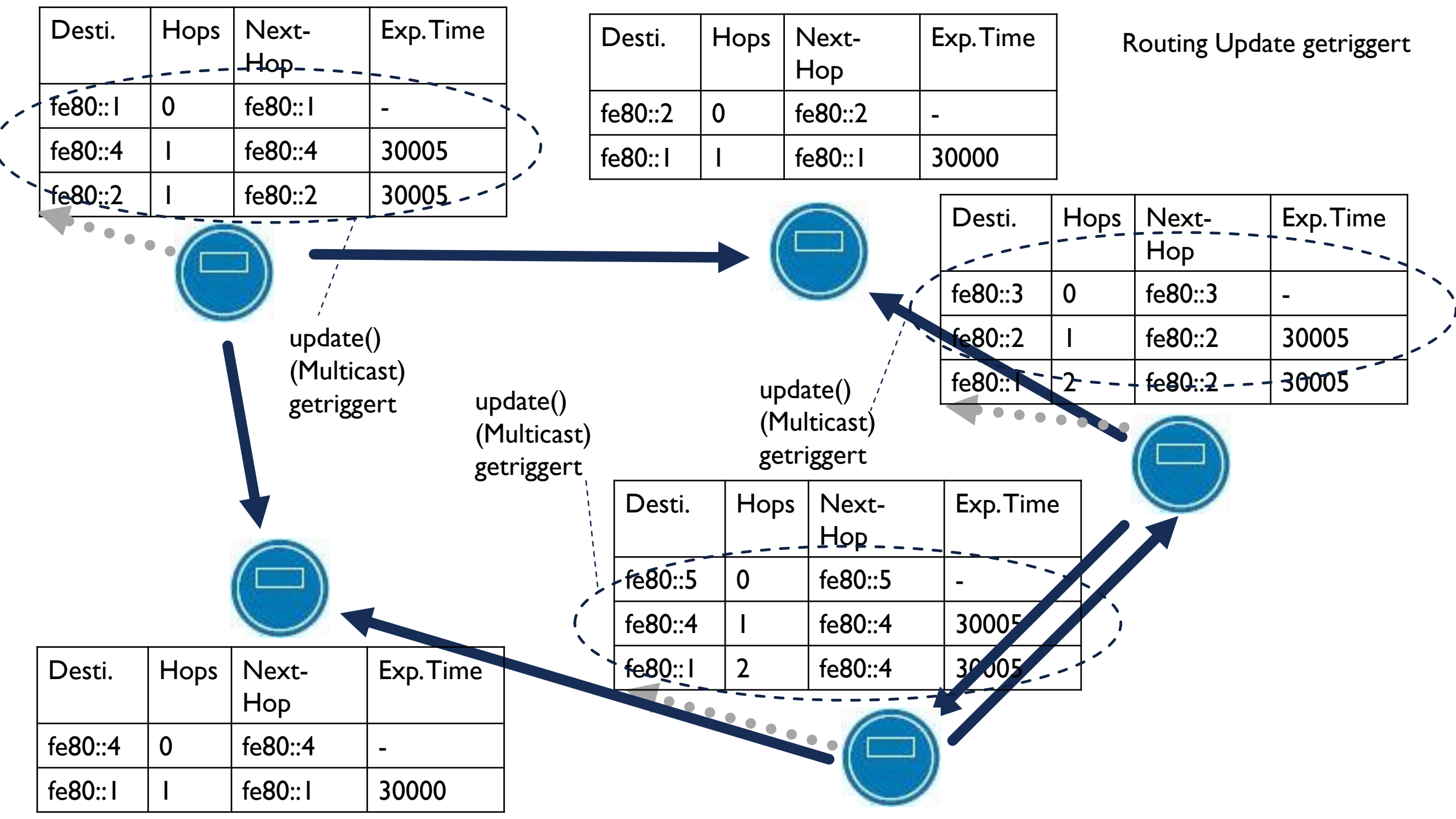
Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000



update()
(Multicast)
gettriggert

update()
(Multicast)
gettriggert

update()
(Multicast)
gettriggert

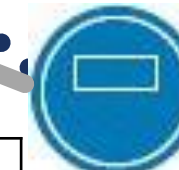


Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-
fe80::4	1	fe80::4	30005
fe80::2	1	fe80::2	30005

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30010
fe80::4	2	fe80::1	30010
fe80::3	1	fe80::3	30010

Routing Update gettriggert

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005
fe80::5	1	fe80::5	30010
fe80::4	2	fe80::5	30010



change

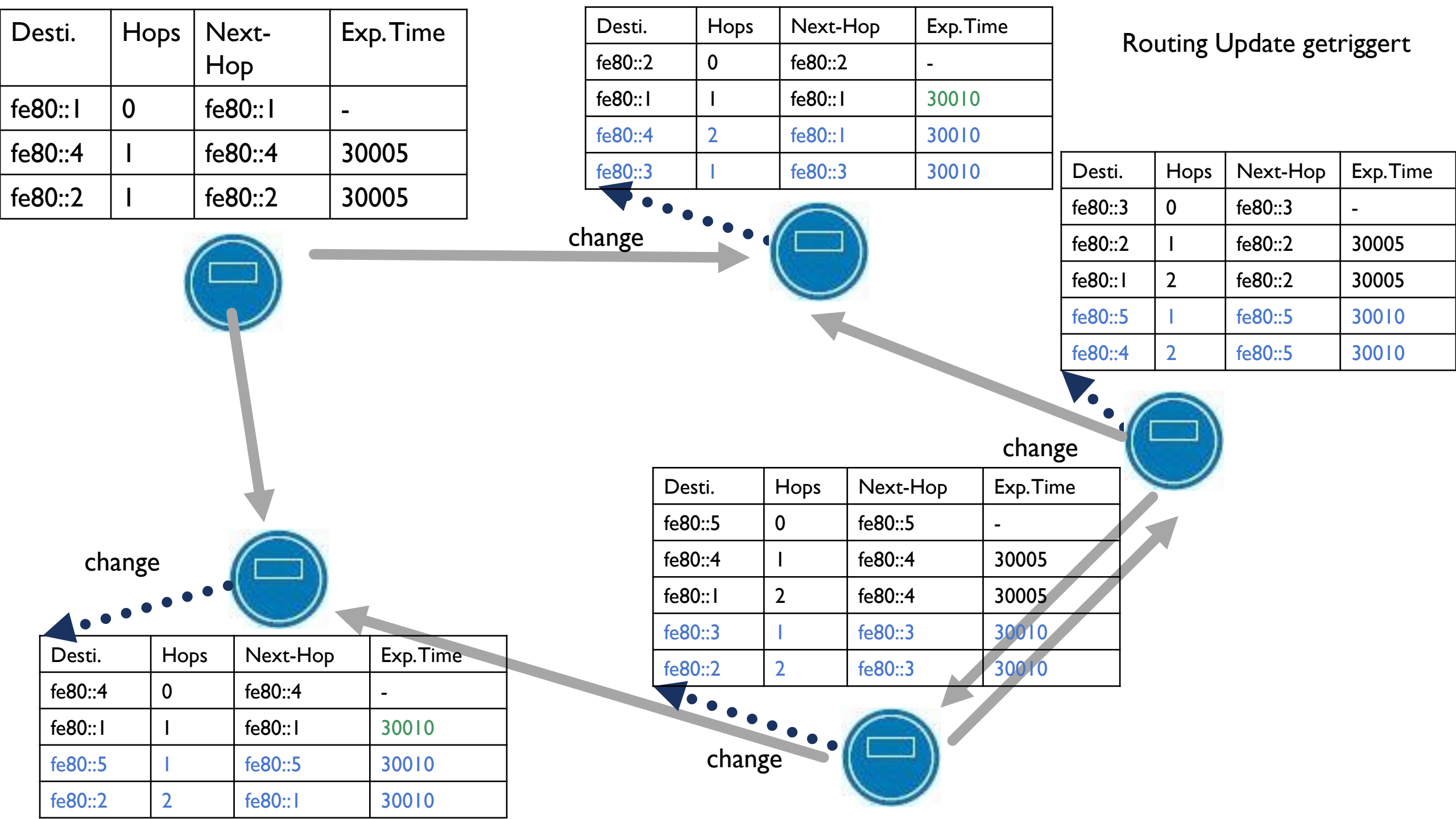
change

change

change

Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30010
fe80::5	1	fe80::5	30010
fe80::2	2	fe80::1	30010

Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005
fe80::3	1	fe80::3	30010
fe80::2	2	fe80::3	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-
fe80::4	1	fe80::4	30005
fe80::2	1	fe80::2	30005
fe80::3	2	fe80::2	30015
fe80::5	2	fe80::4	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30010
fe80::4	2	fe80::1	30010
fe80::3	1	fe80::3	30015
fe80::5	2	fe80::3	30015

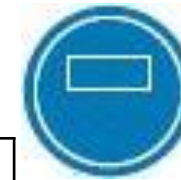
geroutete Nachricht

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005
fe80::5	1	fe80::5	30010
fe80::4	2	fe80::5	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000
fe80::5	1	fe80::5	30015
fe80::2	2	fe80::1	30010
fe80::3	2	fe80::5	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005
fe80::3	1	fe80::3	30010
fe80::2	2	fe80::3	30010

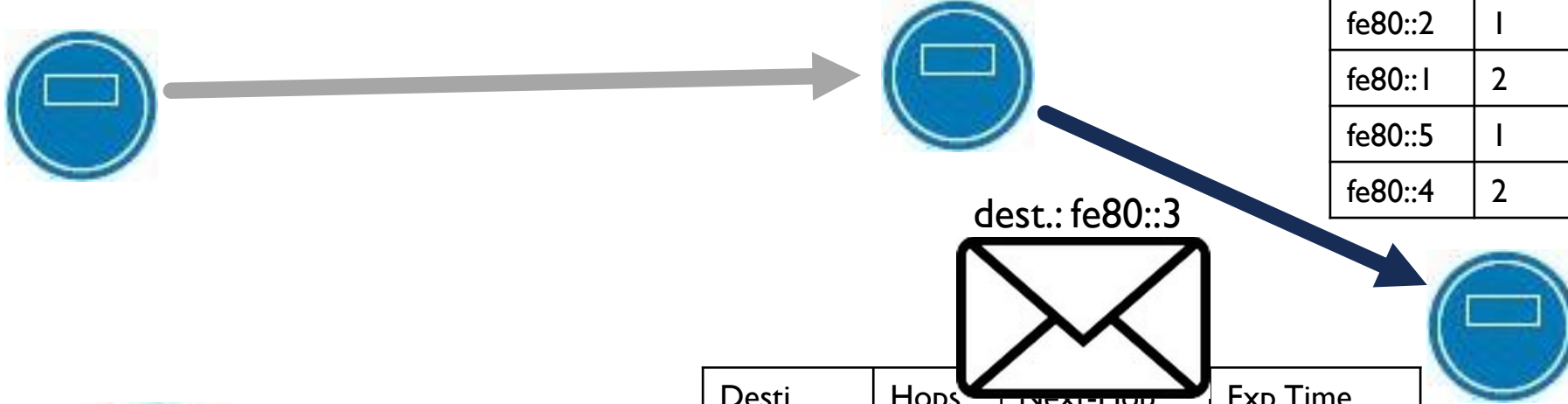


Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-
fe80::4	1	fe80::4	30005
fe80::2	1	fe80::2	30005
fe80::3	2	fe80::2	30015
fe80::5	2	fe80::4	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30010
fe80::4	2	fe80::1	30010
fe80::3	1	fe80::3	30015
fe80::5	2	fe80::3	30015

geroutete Nachricht

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005
fe80::5	1	fe80::5	30010
fe80::4	2	fe80::5	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000
fe80::5	1	fe80::5	30015
fe80::2	2	fe80::1	30010
fe80::3	2	fe80::5	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005
fe80::3	1	fe80::3	30010
fe80::2	2	fe80::3	30010

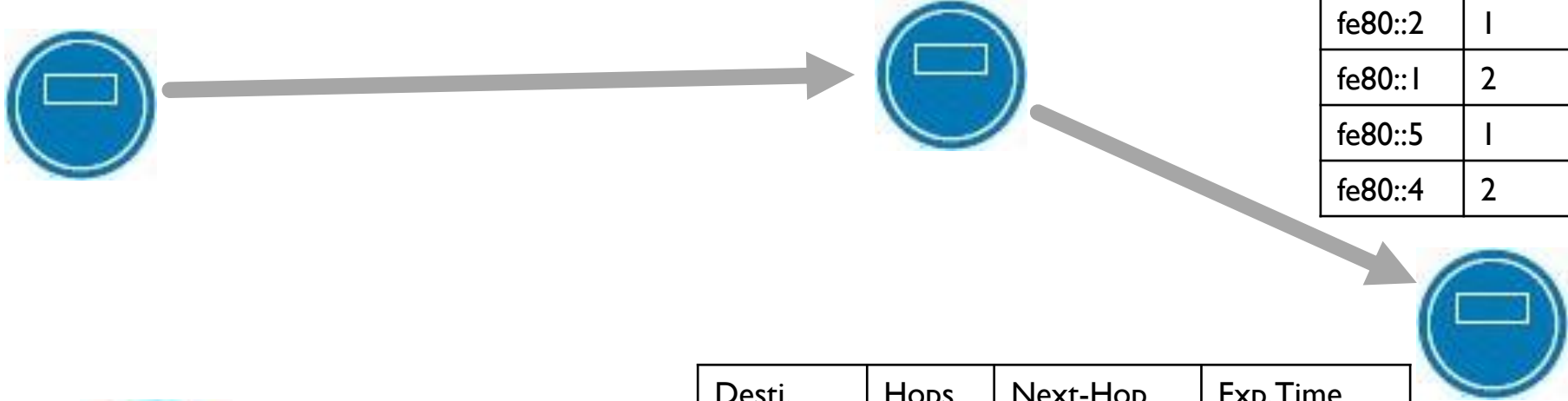


Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-
fe80::4	1	fe80::4	30005
fe80::2	1	fe80::2	30005
fe80::3	2	fe80::2	30015
fe80::5	2	fe80::4	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30010
fe80::4	2	fe80::1	30010
fe80::3	1	fe80::3	30015
fe80::5	2	fe80::3	30015

geroutete Nachricht

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005
fe80::5	1	fe80::5	30010
fe80::4	2	fe80::5	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000
fe80::5	1	fe80::5	30015
fe80::2	2	fe80::1	30010
fe80::3	2	fe80::5	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005
fe80::3	1	fe80::3	30010
fe80::2	2	fe80::3	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-



update()
(Multicast)
zyklisch



Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	30000
fe80::5	1	fe80::5	30015
fe80::2	2	fe80::1	30010
fe80::3	2	fe80::5	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	30010
fe80::4	2	fe80::1	30010
fe80::3	1	fe80::3	30015
fe80::5	2	fe80::3	30015

Routing Table Eintrag
entfernen

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005
fe80::5	1	fe80::5	30010
fe80::4	2	fe80::5	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005
fe80::3	1	fe80::3	30010
fe80::2	2	fe80::3	30010



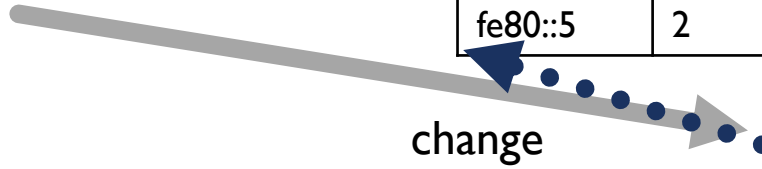
Desti.	Hops	Next-Hop	Exp.Time
fe80::1	0	fe80::1	-



change

Desti.	Hops	Next-Hop	Exp.Time
fe80::4	0	fe80::4	-
fe80::1	1	fe80::1	40000
fe80::5	1	fe80::5	30015
fe80::2	2	fe80::1	40000
fe80::3	2	fe80::5	30015

Desti.	Hops	Next-Hop	Exp.Time
fe80::2	0	fe80::2	-
fe80::1	1	fe80::1	40000
fe80::4	2	fe80::1	30010
fe80::3	1	fe80::3	30015
fe80::5	2	fe80::3	30015



Routing Table Eintrag entfernen

Desti.	Hops	Next-Hop	Exp.Time
fe80::3	0	fe80::3	-
fe80::2	1	fe80::2	30005
fe80::1	2	fe80::2	30005
fe80::5	1	fe80::5	30010
fe80::4	2	fe80::5	30010



Desti.	Hops	Next-Hop	Exp.Time
fe80::5	0	fe80::5	-
fe80::4	1	fe80::4	30005
fe80::1	2	fe80::4	30005
fe80::3	1	fe80::3	30010
fe80::2	2	fe80::3	30010



4. ARBEITSBEREICHE

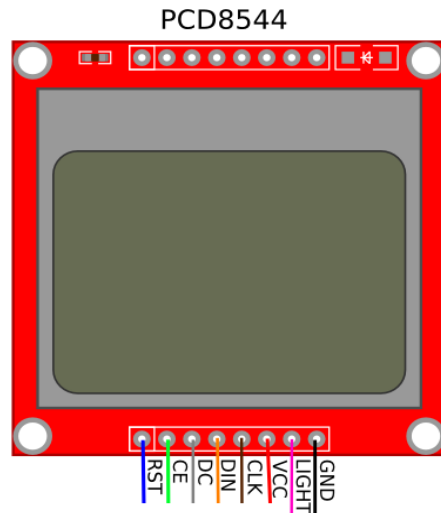
Pairing (MonitoredItem / Monitor) (Thorben):

1. MonitoredItem sendet bei Knopfdruck ein **bind** an Monitor
2. Monitor empfängt bind:
 1. Speichert die Adresse des Senders
 2. Sendet ein **bind_ack** an Sender
 3. Beginnt **heartbeats** zu Senden
3. MonitoredItem empfängt **bind_ack** und wartet anschließend auf **heartbeats**

4. ARBEITSBEREICHE

Graphische Ausgabe auf dem Nokia Display (pcd8544):

- Display wird mittels SPI angesteuert



- pcd8544.c implementiert Methoden um Zeichenketten auszugeben

- Board (pba-d-01-kw2x) : periph_conf.h

```
/* SPI 0 pin configuration */
#define SPI_0_PORT          PORTC
#define SPI_0_PORT_CLKEN() (SIM->SCGC5
#define SPI_0_AF           2

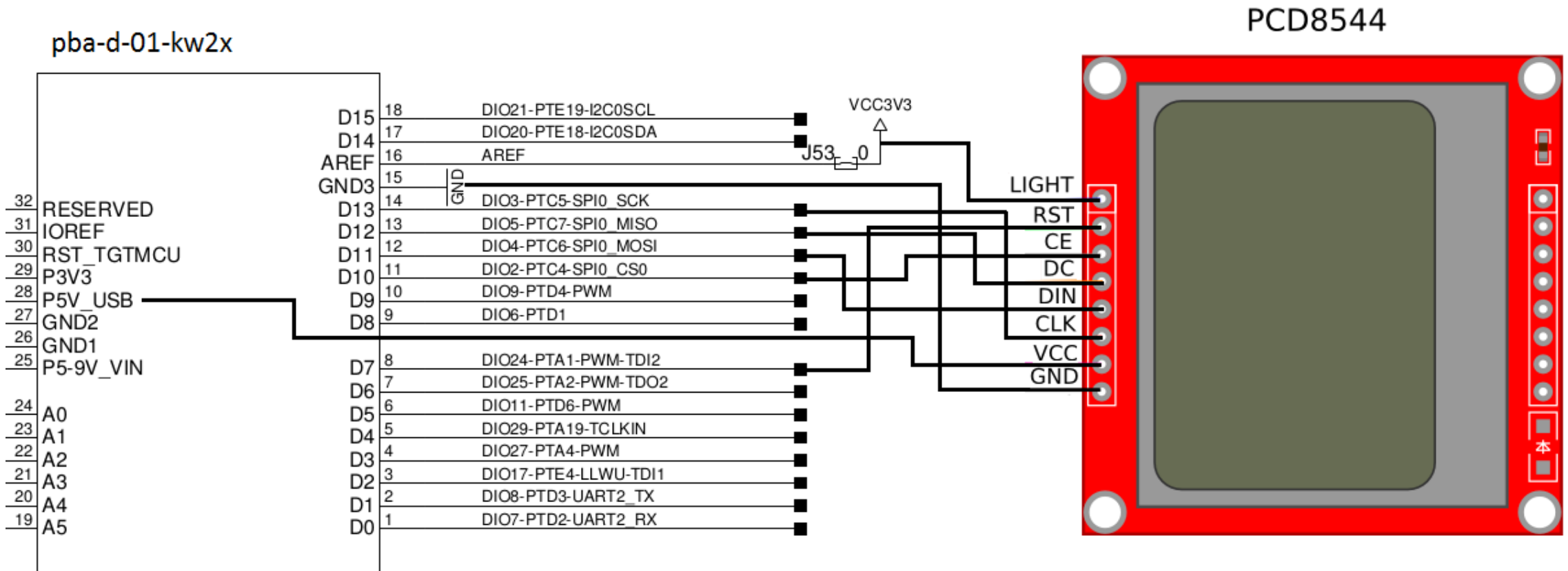
#define SPI_0_PCS0_PIN     4
#define SPI_0_SCK_PIN     5
#define SPI_0_SOUT_PIN    6
#define SPI_0_SIN_PIN     7
```

- Makefile ergänzen

```
TEST_PCD8544_SPI    ?= SPI_0
TEST_PCD8544_CS     ?= GPIO_PIN\ (2,4\
TEST_PCD8544_RESET ?= GPIO_PIN\ (0,1\
TEST_PCD8544_MODE  ?= GPIO_PIN\ (0,2\
```

4. ARBEITSBEREICHE

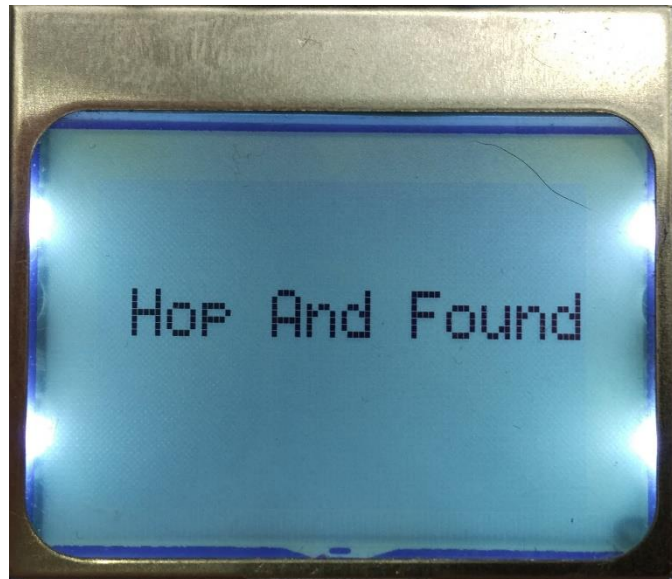
Graphische Ausgabe auf dem Nokia Display (pcd8544):



4. ARBEITSBEREICHE

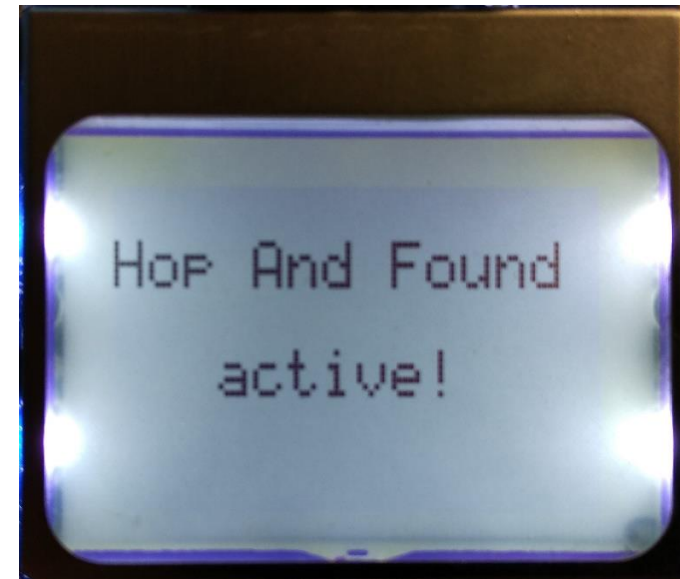
Graphische Ausgabe : Pairing (Thorben)

1. Monitor ist mit keinem MonitoredItem verbunden



2. MonitoredItem-Userbutton wurde gedrückt

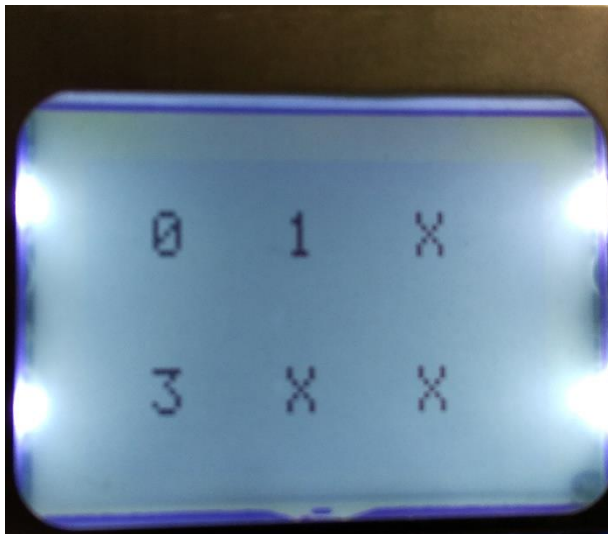
→ Monitor wird mit MonitoredItem verbunden



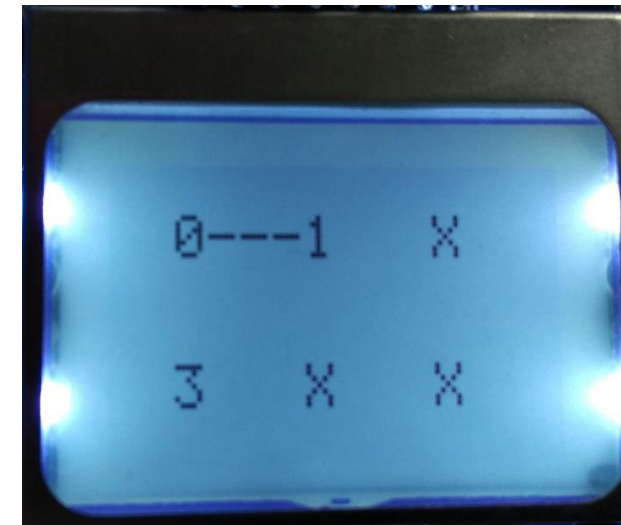
4. ARBEITSBEREICHE

Graphische Ausgabe : call_for_help (Thorben)

- Monitor erhält einen call_for_help direkt vom MonitoredItem / nur über einen Node
 - Zeigt alle vom MonitoredItem erreichbaren Nodes



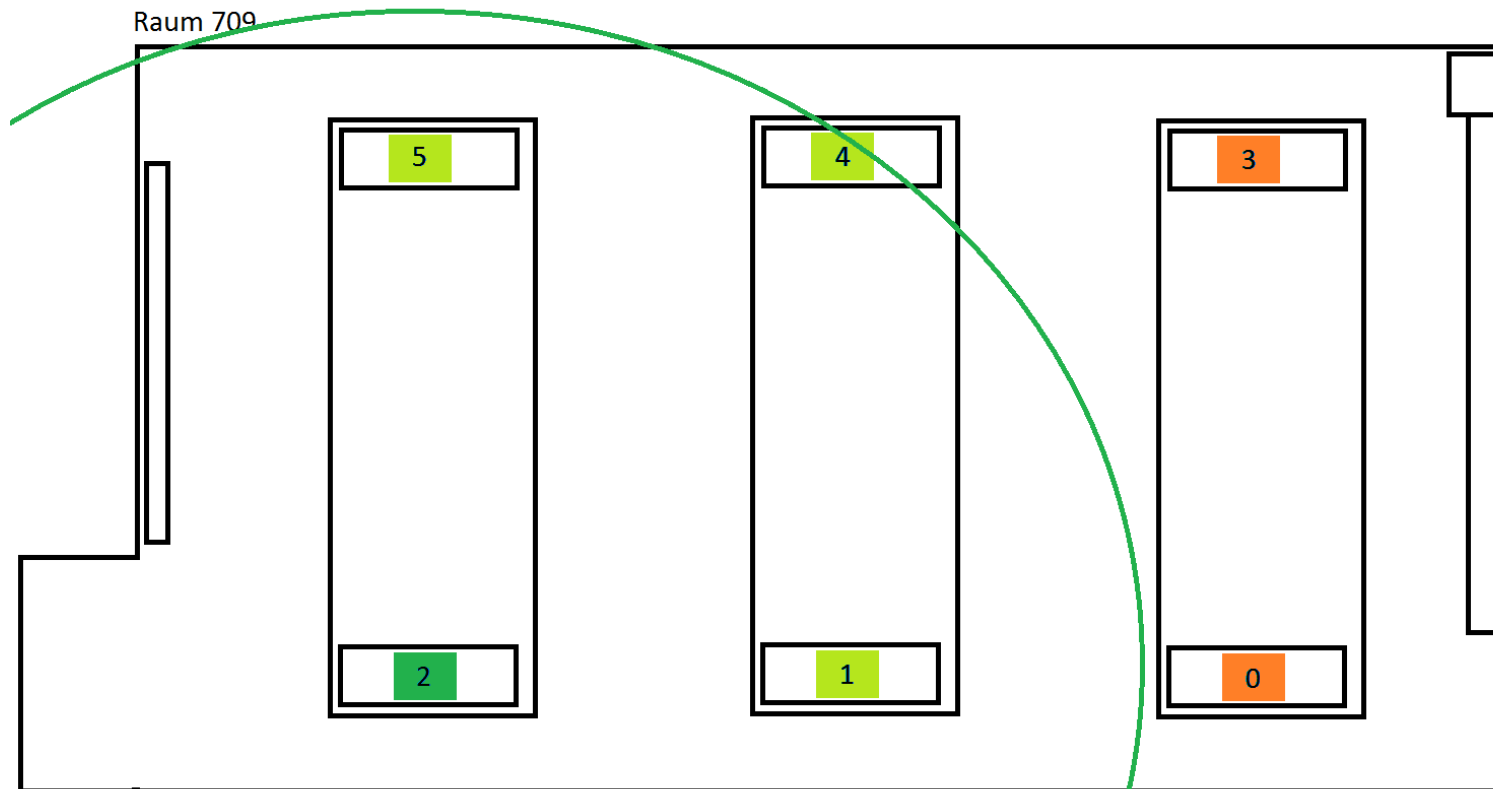
- Monitor erhält call_for_help, welcher über min. 2 Nodes weitergeleitet wurde
 - Zeigt alle vom MonitoredItem erreichbaren Nodes mit der Route des Pakets






4. ARBEITSBEREICHE

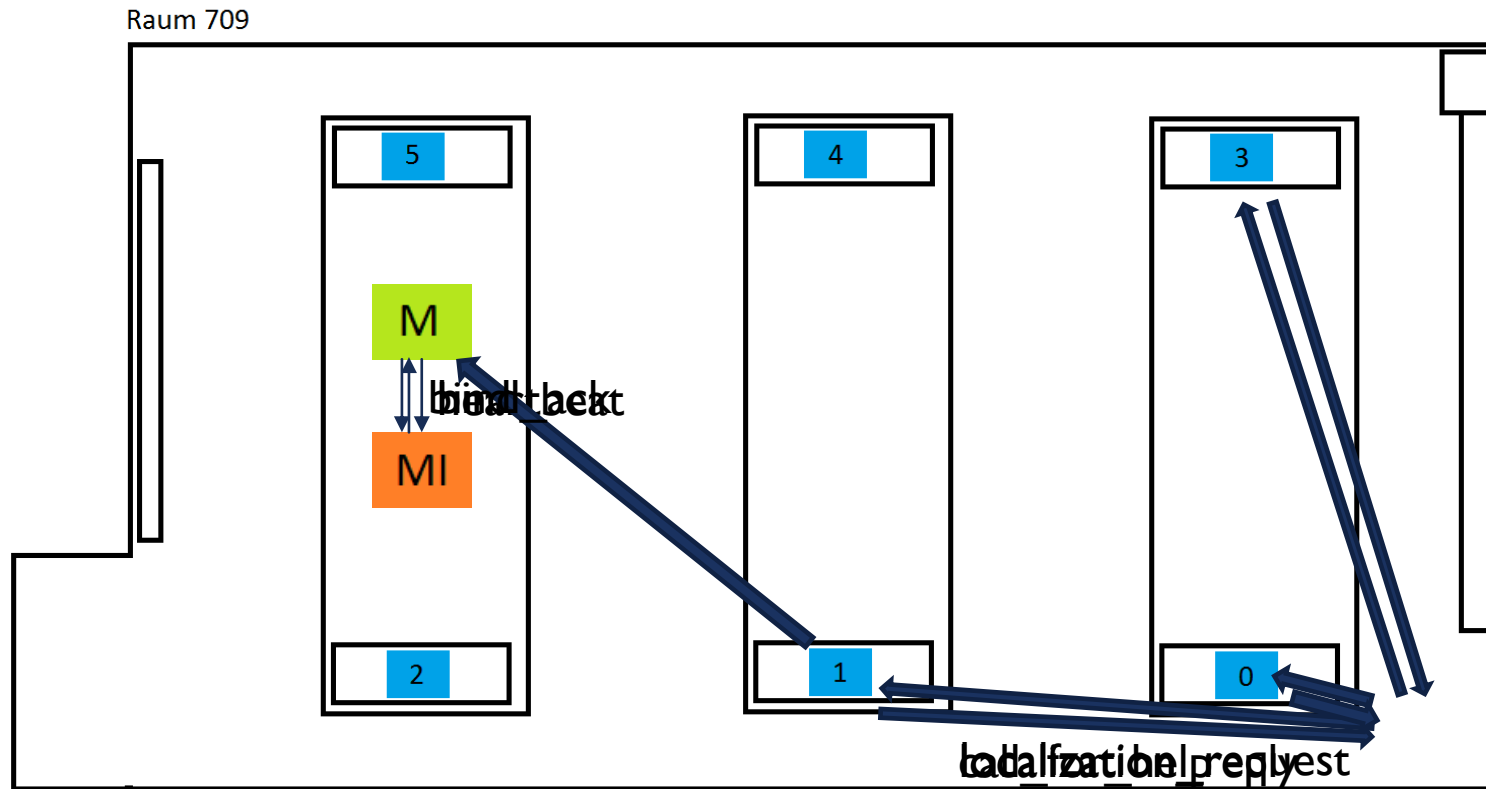
- **Graphische Ausgabe in Konsole (Dennis):**
- **Kommunikationskomponente erweitert**
 - Umstellung der Kommunikation auf Socket Implementierung
 - Konzeption und Implementierung von Sende-Funktionen
- **Thread-Implementierung**
 - Benutzung eines Handle-Threads dem Aufgaben per Message-Passing zugeteilt werden
- **Netzlust**
 - Bei Localization die Antwort der Nodes verzögern, um MI Buffer zu schonen

5. PRÄSENTATIONSAUFBAU



-  = Node von dem Signal ausgeht
-  = erreichbarer Node
-  = nicht erreichbarer Node

5. PRÄSENTATIONSAUFBAU



- X = Node X
- M = Monitor
- MI = MonitoredItem

6. NICHT UMGESetzte IDEEN

Positionsbestimmung des MI mittels

- Triangulation aus der Link-Quality, ermittelt durch erreichte Nodes bei der Localization
- Eingrenzung nach möglicher Reichweite der erreichten Nodes
- GPS

Nachrichten-Versand auf Layer2

Routing anhand von Qualitätskriterien statt Hops

- Round-Trip-Time(RTT) oder Link-Quality

7. GELERNTES

Schwierigkeiten während der Realisierung

- Routing-Entwicklung führte mehrmals zu Antwort-Schleifen
- Umsetzung bei Routing auf IPv6-Adressen komplexer als erwartet
- Übertragungsverfahren spät festgelegt
- Ansteuerung des Displays + LED + Button (GPIO doppelt belegt)

7. GELERNTES

Eingetretene Lerneffekte

- Strukturiertes Software Development
 - Für Elektrotechniker viele gute Anregungen durch Erfahrungen der Informatiker
- Gute Dokumentation hilft beim Arbeiten mit RIOT
- Beim Arbeiten mit eingebetteten Systemen sind wenig Ressourcen vorhanden

8. MÖGLICHE VERBESSERUNGEN

Routing

- Nur senden der beschriebenen Einträge einer Routing-table

Pairing

- Binden von mehreren MonitoredItems an einen Monitor

Ressourcenallokierung

- Auslagerung einiger Variablen vom Stack



FRAGEN!?

VIELEN DANK FÜR EURE AUFMERSAMKEIT

