
Lokalisierung in Sensornetzwerken

Ein Projekt des Fraunhofer Instituts Integrierte Schaltungen

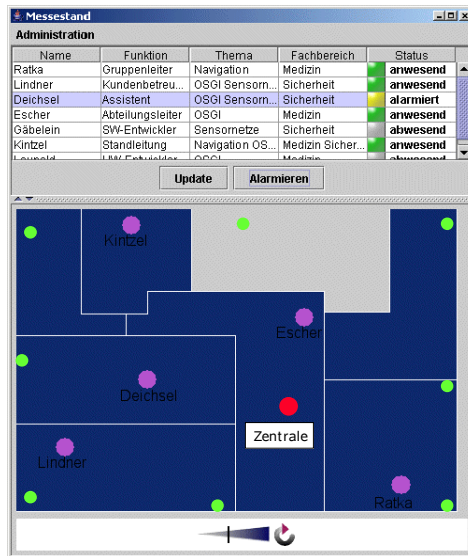
Vortrag im Rahmen der Vorlesung Organic Computing
Dr. rer. nat. Christophe Bobda, Prof. Dr. Rolf Wanka
Department of Computer Science 12
Hardware-Software-Co-Design
12.07.2005

Werner Spiegl, Christian Flügel
Fraunhofer Institut Integrierte Schaltungen

Gliederung

- Überblick
- Infrastruktur
- Sensorknoten
 - Motes (Anker, Badge, Master)
 - Protokolle
 - TinyOS, NesC
- Routing
- Positionsbestimmung
- Zentrale Berechnung am PC
- Zukunft

Überblick



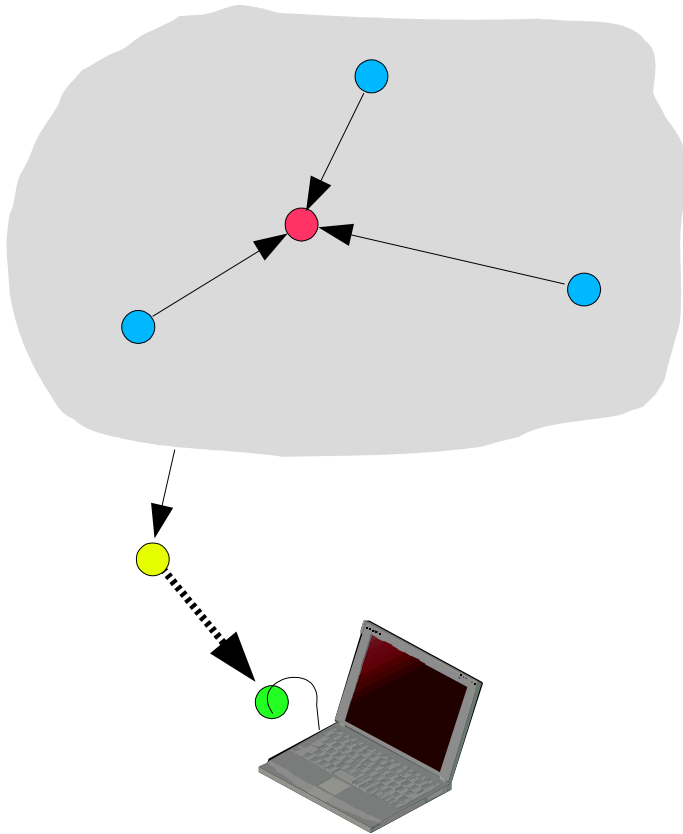
Screenshot: FhG IIS



Foto: XBow

- Sensornetzwerk zur Lokalisierung von Geräten und Personen und zur Übermittlung von Statusmeldungen (bidirektional)
- Multi-Hop Kommunikation
- Dynamisches Routing
- Zentraler Lokalisierungsalgorithmus
- Genauigkeit: ca. 40-50% der Distanz zwischen zwei Ankerknoten

Infrastruktur



- Fest installierte Ankerknoten (Beacon), mit bekannter Positionen, ermittelt
 - bei einer initialen Installation (akt. Methode)
 - über GPS-Module
 - über eine sonstige denkbare Methode
- Mobile Knoten mit unbekannter Position (Badge)
- Routingknoten
- Masterknoten an zentralem Rechner, an den alle ermittelten Daten übertragen werden zur Positionsberechnung
- Selbständige Vernetzung der Beacons, Badges und Routingknoten zu Sensornetzwerk

Mobile Node (1985)



Sensorknoten

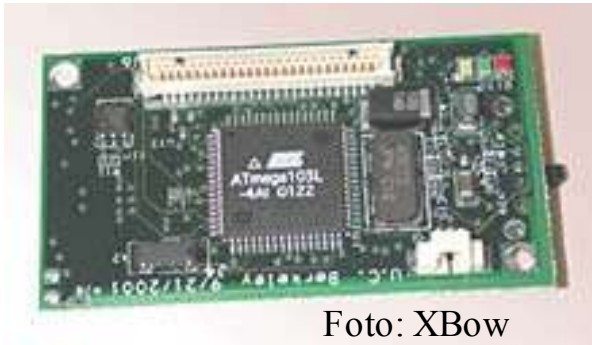


Foto: XBow

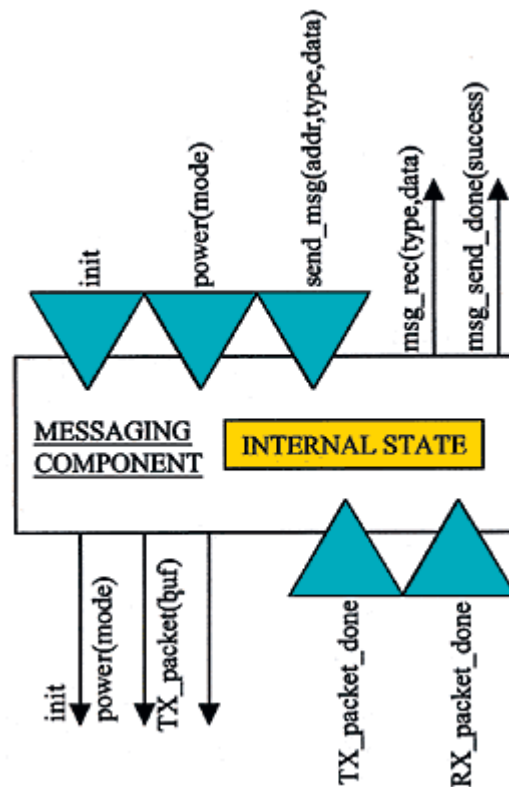


- Mica2 bzw. Mica2Dot Motes (MPR400) von Xbow.com als Beacon, Badge u. Routingknoten
- Auszug aus dem Datenblatt:
 - Abmessungen: 57mm(l) x 31mm(b) bzw. 25mm(D)
 - Prozessor: Atmega 128L 7,3MHz bzw. 4MHz
 - Speicher: 128 kB Flash, 4kB RAM, 4kB EEPROM
 - Frequenzen: 300 – 1000 Mhz (akt. 868 MHz)
 - Stromverbrauch:
 - Power Down Modus < 25 μ A
 - Empfangen mit ca. 15 – 20 mA
 - Senden mit ca. 20 – 25 mA
- Hardware mit besonderen Anforderungen an
 - den Übertragungsstandard => Proprietäres System,
 - das Betriebssystem => TinyOS und
 - die Programmiersprache => NesC

Proprietäres System:

- Angepasste Bibliotheken
- Routing von Intel: DSDV (Destination Sequenced Distance Vector)
 - Ad-Hoc-Verfahren, spontane Vernetzung der Sensorknoten
 - Proaktives Routingverfahren, Route-Updates in periodischen Abständen
- MAC-Layer auf S-MAX der University of Southern California basierend:
 - Sicherstellen von synchronisierten Schlaf-Wach-Rhythmen, Duty Cycle (Verhältnisses der aktiven Zeit zur inaktiven Zeit, akt. 10%)
 - Gegenseitige Abstimmung von Latenzzeit und Energieverbrauch in Abhängigkeit von den aktuellen Verbindungen im Netz

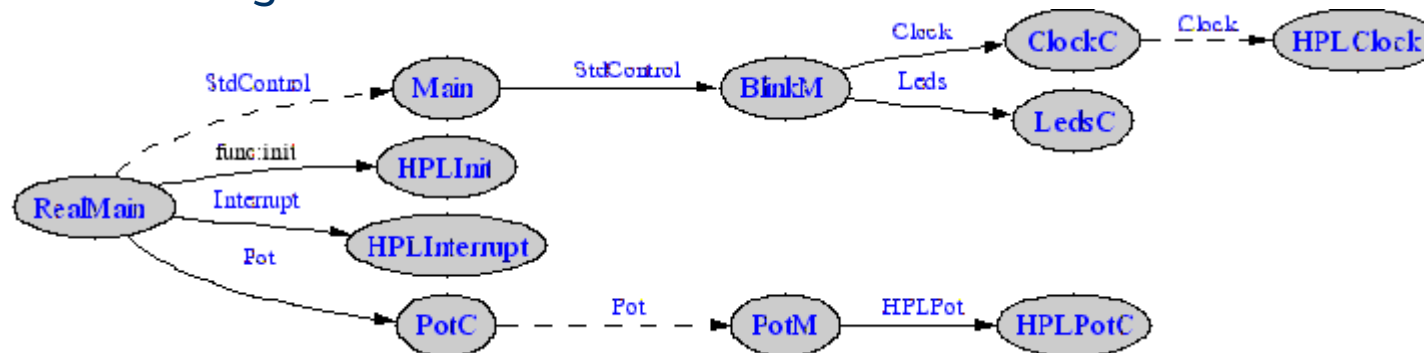
Betriebssystem: TinyOS



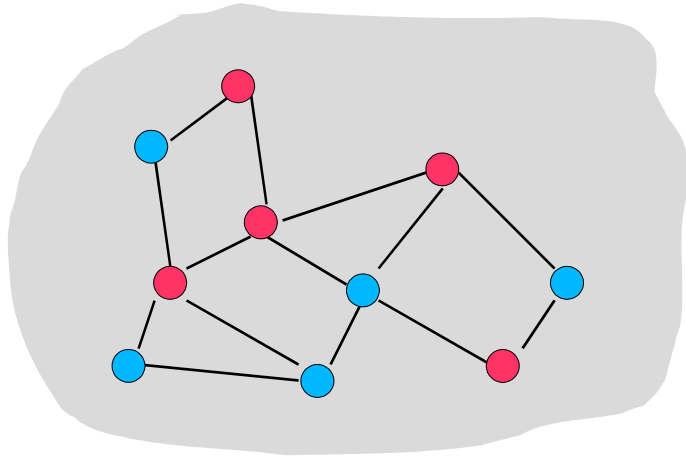
- Event-basierendes OS, entwickelt für drahtlose Sensornetzwerkknoten mit beschränkten Ressourcen
- Komponenten-basierende Architektur
- Die anpassbare Komponentenbibliothek beinhaltet Netzwerkprotokolle, Sensortreiber, etc.
- Event-gesteuerte Ausführung erlaubt sehr gute Energiekontrolle
- TinyOS is open-source
- <http://www.tinyos.net/>
- <http://sourceforge.net/projects/tinyos/>

Programmiersprache: NesC (Networked Embedded Systems C)

- Ursprünglich entwickelt im Hinblick auf die Konzepte von TinyOS, Erweiterung zu C;
- Programmierung über Module und Konfigurationen; Module nur über Schnittstellen ansprechbar; in Konfigurationen Verdrahtung von Modulen zu Programmen
- Sicherer als C, vereinfachter Programmaufbau durch das Komponentenmodell
- „Statische Sprache“ ohne dynamisches Anfordern von Speicher; dadurch einfachere und genauere Programmanalyse und Optimierungen
- <http://nesc.c.sourceforge.net/>



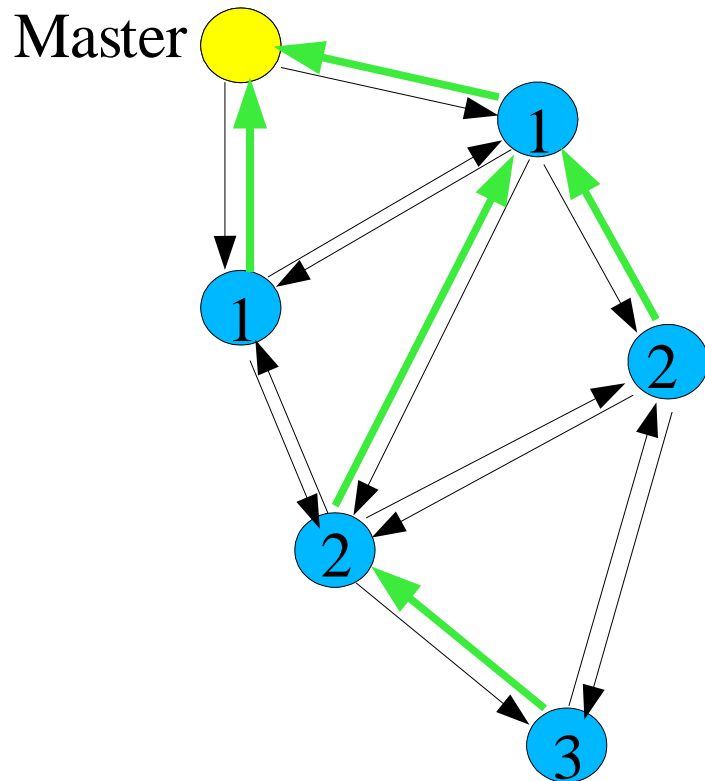
Routing im Sensornetzwerk



- bekannte Position (Beacon)
- unbekannte Position
- Nachbarschaftsbeziehung bekannt

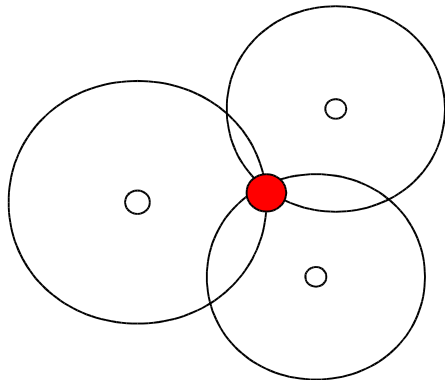
- Routen im SNW nicht festgelegt, Redundanz bei zeitweisem oder dauerhaftem Ausfall von Knoten durch:
 - Ausfall der Energieversorgung (z.B. leere Batterie)
 - Hardwaredefekt
 - Räumliche Veränderung
 - Positionsänderung der mobilen Knoten
 - Bekannte Routingprotokolle (z.B. für Internet) ungeeignet, da besondere Anforderungen im SNW gelten:
 - Selbstorganisation, Rollenfestlegung der Sensorknoten
 - Minimierung des Energieaufwands (z.B. Sendeenergie, Anzahl der Nachrichten)
 - Minimierung des Speicherbedarfs
- Anwendung spezieller Routingverfahren

Routingverfahren



- Periodische Route-Updates von Master getriggert (akt.: 5s), Flooding über Anker (optional Badges) und spezielle Routingknoten
- Jeder Knoten merkt sich nur den Vorgänger
- Verbindungsstabilität als Qualitätsmetrik:
 - Paketverlustmessung u. Entfernung
 - Zeitaufwendiger als Hop-Counting, aber zuverlässiger
 - Dauer zur Verifizierung einer neuen Route: ca. 30s

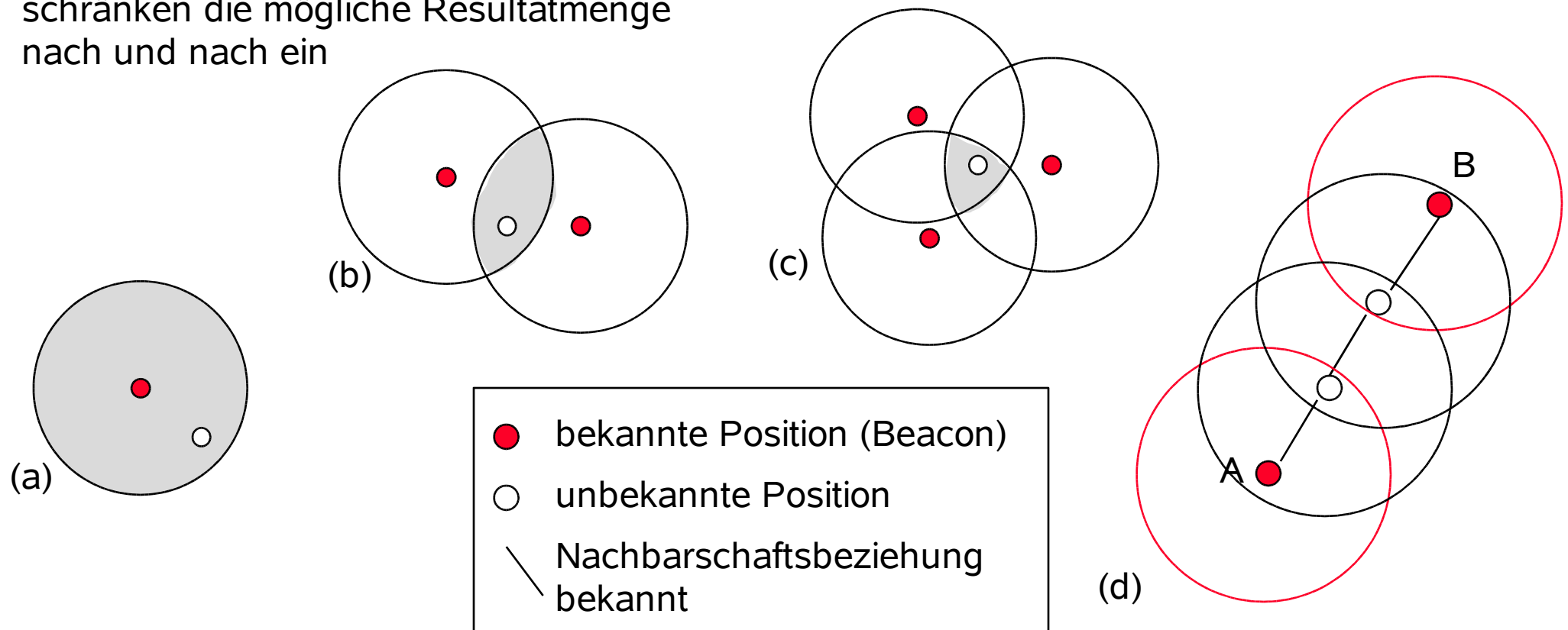
Positionsbestimmung



- Ziel: Einen bestimmten (mobilen) Sensorknoten innerhalb eines (statischen) Sensornetzwerkes mit einer gewissen Genauigkeit zu orten
- Zur Berechnung muss jeder unbekannte Knoten ausreichend Ankerknoten sehen (mind. 3)
- Ankerknoten übermitteln periodisch ein Beacon-Signal (ID des Motes)
- Badges messen die Feldstärke (RSSI) der empfangenen Beacon-Signale
- Weiterleiten der Daten per Multi-Hop (über die Anker- und Routingknoten) an den Master
- Berechnung der Positionen auf zentraler Recheneinheit
- Schwerpunkt durch RSSI gewichtet

Positionsbestimmung mit mehreren Nebenbedingungen

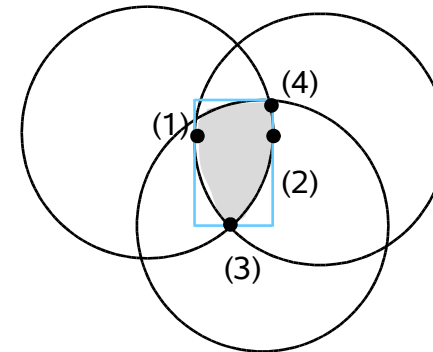
- Zusätzliche Nebenbedingungen schränken die mögliche Resultatmenge nach und nach ein



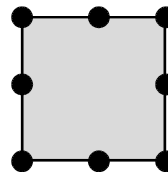
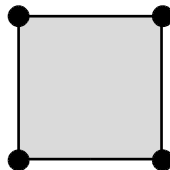
Verbesserung der Positionsbestimmung

Verbesserung der Positionsabschätzung

- Mittelpunkt des möglichen Vierecks
- Zeit ungefähr * 4

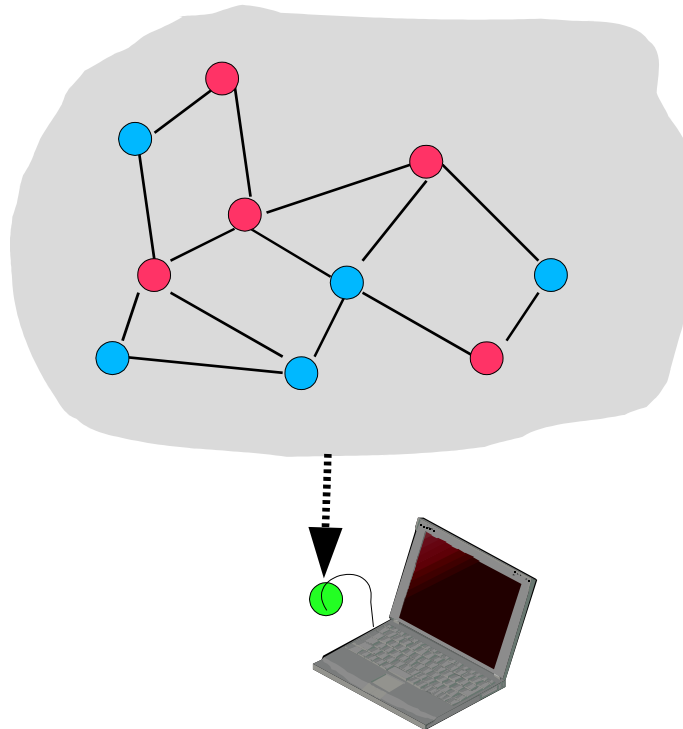


Intelligente Platzierung der Beacon-Knoten

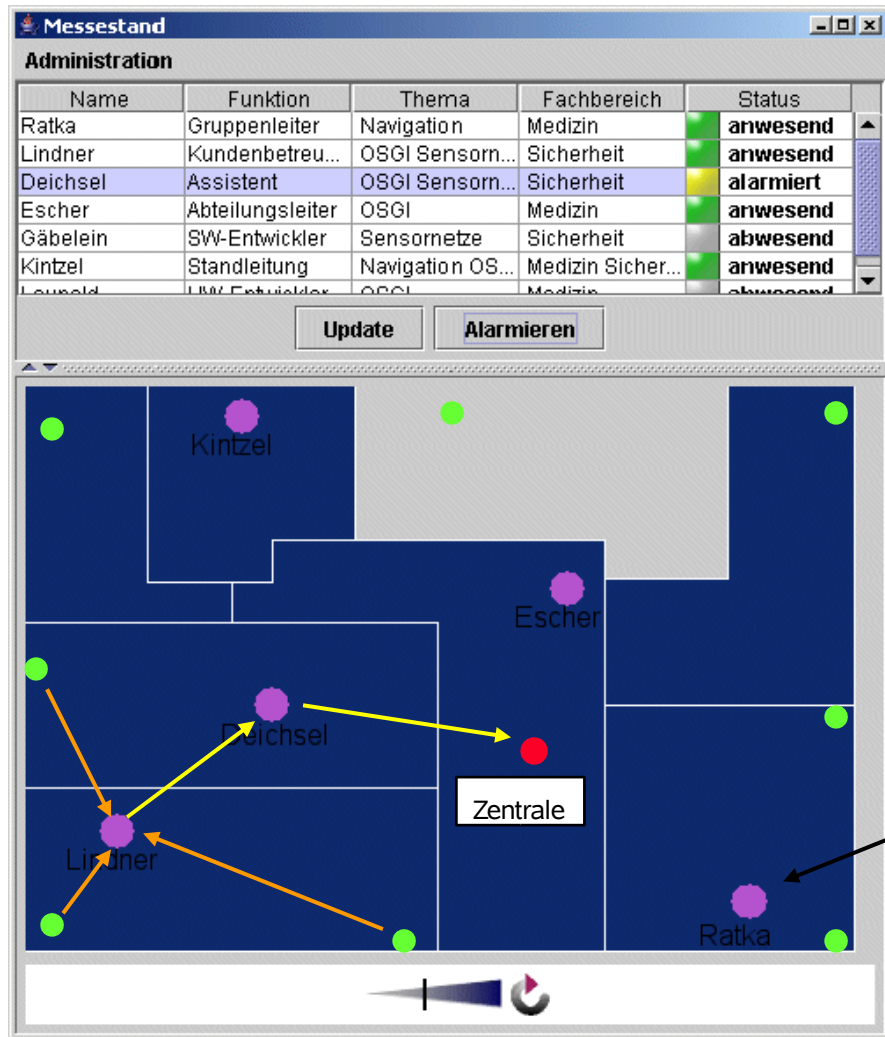


- Durch geeignete Wahl der Position der Beacon-Knoten mit angepasster Sendereichweite: $R/D < 1$
- ideal: $R = 0.86D$

Zentrale Berechnung

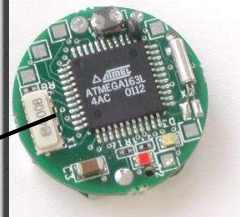


- Berechnung der Positionen auf zentraler Recheneinheit
- Möglich: Globale Sicht (des ganzen Netzwerks)
- Zentrale Berechnung von lokalen Daten:
 - Daten der jeweiligen Badges werden ausgewertet
 - Lokaler Algorithmus ausgelagert auf PC



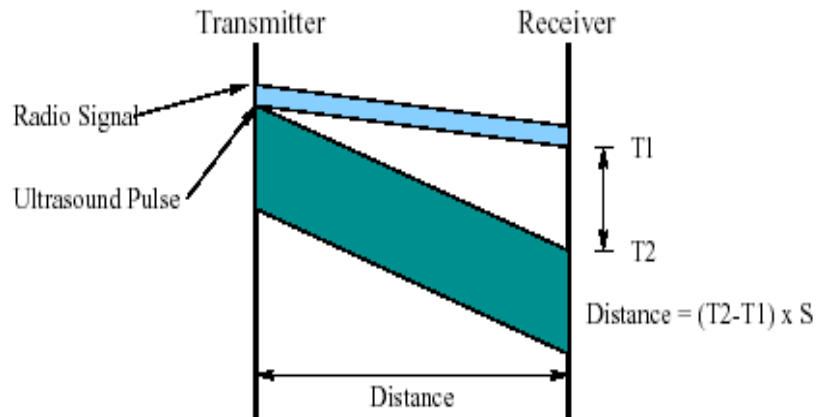
Zentrale Berechnung am PC

- Java, Serial Forwarder:
 - Masterknoten an Interface über COM-Schnittstelle mit PC
 - Stellt Server-Port zur Verfügung für unterschiedliche Clients
- Java, Client:
 - 2D-Raumgeometrie (über XML-Format bearbeit-/ importierbar)
 - Interpretation der Daten
 - Lokalisierungsalgorithmus
 - Zuweisen von Personeninformationen an Badge-ID
 - Personalstatus ablesbar
 - „Alarmieren“ von Personal in Reichweite



Badge

Zukunft



- Verfeinerung der Positionsbestimmung:
 - Messung der Signallaufzeiten, TOA, TDOA (time [difference] of arrival), Messen des Zeitunterschieds von 2 gleichzeitig gesendeten Signalen: RF- und Ultraschall (Genauigkeit beim 3m Knotenabstand: 2cm)
 - Nachteil: Erhöhung von Platzbedarf (auf dem Sensorknoten), Energiebedarf und Hardwarekosten
- Senken des Energiebedarfs:
 - Anwenden/Entwickeln günstigerer Übertragungsprotokolle => niedrigerer Duty Cycle
- Minimierung der Hardwarekosten
- Neue Projekte im Bereich Sensornetzwerke
 - ...

Danke für Ihre Aufmerksamkeit!



Fragen?