Internet of Everything

Vanessa Schindler

09.02.2016

Inhaltsverzeichnis

1 Geräteklasse

1.1 Einführung

Geräteklassen nach Mark Weiser

- Tabs cm in Kleidung
- Pads dm in der Hand
- Boards m feste interaktive Bildschirme

von Anwendungsanforderungen abhängig

1.2 Klassifikation

1.2.1 Leistungsfähigkeit

- Speicher, Rechenleistung, Übertragungsrate
- gegenseitige Abhängigkeit anderer Klassifikationsmerkmale
- IETF RFC 7228 sehr kleine + einfache Geräte
- Adressierbarkeit von Objekten (6LowPAN/coAP)

1.2.2 Energiebeschränkung

1. E0: Ereignisbeschränkt (Eventbasiertes Harvesting)
2. E1: Zeitperiode (Batterie periodisch geladen, ausgetauscht)
3. E2: Lebenszeitbeschränkung (nicht austauschbare Batterie)
4. E9: keine Beschränkung (kontinuierliche Energieversorgung)

Probleme

- Bereitstellen/Speichern elektr. Energie
- Umwandlung alternativer Energieformen der Umgebung in nutzbare elektr. Energie -> Energy Harvesting
    1. Vorteil: kostenlose Umgebungsenergie
    2. Nachteil: geringer Wirkungsgrad
    3. Nachteil: Harvesting Systeme groß

1.2.3 Deployment Modell

Aufteilung und Ausbringungsort einzelner Systemkomponenten

logisch funktionale Aufteilung mit

- Systemmodell
- Schnittstellen
- HW-Ressourcen
- Virtuelle Entitäten
- Dienste

Schwerpunkt:

- vollständiges Anwendungssystem
- Datenfluss, Verarbeitungsprozess
- Interaktion Systemkomponenten + Geräten

1.2.4 Sensoren

- Kontext (Aktivität, Position)
- Technik (Infrarot, Ultraschall, Induktion, Schall)
- physikalische Funktionsweise (magnetisch, mechanisch, elektrisch)
- logische Funktionsweise (integriert, intelligent, aktiv, passiv)
- Messgröße (Temperatur, Luftdruck)
- Infromationsgehalt (eindim, mehrdim)
- Räumlicher Bezug (Berührungslos, berührernd, gerichtet, omnidirektional)
- Anwendungsgebiet

Analog Digital Umwandlung der Messwerte notwendig. Analog Digital Converter (direkt im Mikrocontroller)

1.3 Hardware und Anwendungen

Spezialisierungsgrad sinkt bei steigender Systemgröße

- Nanonetze: Vernetzte Nanomaschinen, die jeweils eine Aufgabe erfüllen: Berechnen, Speichern, Messen, Manipulieren. Anwendungsbereiche: Biomedizin, Militär, Industrie

  Molekulare Kommunikation

  - Walkway Propagation E.coli Bakterien
  - Flow lange Strecken Pheromone
  - Diffusion Signalisierung zwischen Zellen

- Smart Dust
  - Forschungsthema
  - viele kleine dumme Knoten unaufdringlich in Umwelt integriert
  - beschränkte Hardware
  - viele Probleme: Skalierbarkeit (viele Knoten), Netzdichte (Topologiekontrolle), Energieversorgung(Batteriegröße), Entsorgung

- Sensornetze
  - Selbstorganisierend
  - geringe Leistungsaufnahme
  - Batteriebetrieb
  - Kommunikationsschnittstellen: Funk (Bluetooth Low Energy, ZigBee), i2C, SPI
  - Überwachung großer Flächen (ZebraNet): Mehrere Funkschnittstellen (kurze Entfernung Sensornetz, größere Basisstation), Sensorgröße Beobachtung nicht behindern, lange Energieversorgung
  - Intelligente verteilte Regelung(Industrieanlage): drahtlose Verbindung günstiger, Einbettung in bestehnde Systeme
  - Katastropheneinsätze

- Physical and Embedded Computing: Flexible Systeme aus eingebetteter und miniaturisierter Standardhardware
  - hohe Energieeffizienz, kostengünstig, große Sensorknoten
  - Steuerungselektronik, Home Automation, Smart Metering

- Smart und Submetering: Zeitnahe Erfassung und Steuerung von Energieverbräuchen
  - Smart Grid Vision: bessere Infos für Kunden, effizientere Nutzung Ressourcen

  - Deutsches Modell (BSI) viele Zähler, ein Gateway

- Smart Home: Hausautomation und -monitoring durch drahtlose Sensornetzwerke

  Herausforderungen

  - einfache Konfiguration vs. Sicherheit
  - Benutzerverwaltung, Zugriff
  - Robustheit, Energieeffizienz

  Lifestyle Produkte, sind im Kernbereich privater Lebensgestaltung. Abhängigkeit von externer Infrastruktur(qivicon)

- Drohnen und Roboter: Mobile Plattform mit Sensoren und Aktoren zur Messung vor Ort
  - Einsatz lebensfeindlichen Bedingungen (Radioaktivität), Umterstützung Menschen
  - Abstandssensor (Infrarot), GPS, LED, Motoren, ZigBee, WLan
  - Mobilität (Roboterschwarm, Erweiterung statischer Sensorknoten), Batterie/Enrgy Harvesting

- Wearables
  - Datenverarbeitung in Körpernähe
  - Unterstützung Alltag
  - Unauffällige Integration in nicht technische Gegenstände
  - Erfassung Umgebung
  - Neuartige Benutzerschnittstellen
  - Aufzeichnen persönlichen Verhalten, Datenfusion aus unter. Sensorquellen Positionsdaten
  - Auswertung/Anzeige Daten rudimentär möglich, löschen oft nicht vorgesehen

- Smart Phones: Steuerung Remote Geräte, Crowdsourcing, Participatory Sensing
  - Leistungsfähige Hardware
  - Leistungsfähige Kommunikationsinfrastruktur (kommerziell)
  - Kommunikationsschnittstellen (WLAN, UMTS, USB)

- Single Board Computer: Prototyping Umgebung, Haushaltsgeräte, Steuerung (Fernbedienung, Heizung)
  - anpassbare vielfältige Betriebssysteme (Android, PC-OS)
  - kostengünstige Herstellung
  - Watchdog Timer, A/D Wandler, Ethernet

- Industrie 4.0: Breites Spektrum unterschiedlicher Hardware -> Heterogenität. Drahtlose Sensorik (Batterie, Energy Harvesting), Leistungsfähige Steuer -und Regelsysteme

  Anforderungen

  - Zuverlässigkeit, Robustheit, Langlebigkeit
  - Kommunikationsschnittstellen: WirelessHART, RFID, Feldbussysteme

1.4 Geräteanbindung und Datenmodell

Große Unterschiede bei Anwendungsentwicklung zwischen Geräteklassen

1. Ressourcenbeschränkte Geräte zB. Sensorknoten
   - Anwendung eng mit BS verbunden
   - Wenig Abstraktionsebenen
   - 1 Anwendung pro Gerät
   - keine echte Nebenläufigkeit
   - einfaches Powermanagement durch Controller
   - eventbasiert und asynchrone Informationsverarbeitung
   - Hardwarenahe Mikrocontrollerprogrammierung
   - eigene Gerätetreiber
   - Interaktion Sensoren/Aktoren
2. Leistungsstarke Geräte (Smartphones)
   - App-Konzept: Software in gebündelter Form
   - Abstraktionen und Dienste BS nutzbar
   - Multithreading
   - Multicores
   - Speichermanagement, Powermanagement
   - Application-Lifecycle, Activities
   - Interaktion zwischen Anwendungen, Graphische Benutzerschnittstelle

Betriebssysteme:

- Sensornetze: TinyOS, Contiki
- Physical Computing: RISC OS, Linux/Android, Windows On Devices
- Wearables: Android Wear
- Smartphone: Android, iOS

Datenmodell: Betrachten Einzelgerät oft schwer (Systemmodell besser zur Analyse)

Kommunikation: Manipulation der Umwelt, Datenerhebung, Datenspeicherung, Benutzerschnittstelle, Datenanalyse/verarbeitung

Cloudanbindung: Integration Kleindgeräte in Cloud zur Datenhaltung. Geräte oft manuell über PC synchronisiert; Remotesteuerung

- Google PowerMeter
- Amazon Kinesis
- NETLab Toolkit
- Thingsquare

2 Privatsphäre

2.1 Einführung

Sensoren über Internet angreifbar. Beispiel RFID

- Drahtlose Übertragung Informationen zwischen Transponder und Basisstation; Nachfolger Barcode
- Erfassung Lagerbestände, neue Reisepässe, Logistik, Einzelhandel
- Eindeutige Identifikation jedes Objekts ohne Sichtkontakt
- Kill Befehl, Blocker Tags
- Authentifizierung für Zugriff (RFID Tags können einfache kryptographische Operationen)
- physische Zerstörung

  Schutz der Privatsphäre: Schutz Persönlichkeitsrechts bei Verarbeitung personenbezogener Daten (informationelle Selbstbestimmung)

  Schützenswerte Daten: personenbezogene Daten, privacy relevant, Metadaten (wer kommuniziert wann mit wem?) aus technischer Sicht Herausforderung

Säulen zum Schutz Privatsphäre: Regulierung (Datenschutzgesetze): Bundes(Landes)datenschutzgesetz, Regelungen Telekommunikationsgesetz Selbstregulierung (Tüv, trusted shop), Selbstschutz (Privacy Enhancing Technologies)
Allgemeine Schutzziele

- Schutzziel: Anforderungen an Komponenten/System, erfüllt werden müssen um schützenswerte Güter vor Bedrohungen zu schützen
  1. Confidentiality (Vertraulichkeit): keine unauthorisierte Informationsgewinnung möglich
  2. Integrity:
     - starke Integrität: nicht möglich unauthorisiert Daten zu manipulieren.
     - schwache Integrität: unauthorisierte Manipulation nicht unbemerkt
  3. Availability (Verfügbarkeit): keine unautorisierte Einschränkung der Funktionalität des Systems
- Unverkettbarkeit: personenbezogene Daten aus untersch. Kontexten für Angreifer nicht in Bezug gesetzt werden können (Datenvermeidung, Anonymisierung)
- Transparenz: Verarbeitung personenbezogener Daten nachvollziehbar, überprüfbar (Erstellung Logdateien)

- Intervenierbarkeit: betroffene Personen über Art der Erfassung und Verarbeitung personenbezogener Daten selbst bestimmen (Wahlmöglichkeiten Datenverarbeitung)

Vertrauensmodell

- vollständiges Vertrauen: Nutzer vertraut allen Entitäten des Systems komplett
- keinerlei Vertrauen: Schutz Privatsphäre durch PETs erforderlich
- Vertrauen in zentrale Instanz: Nutzer vertrauen zentralen Dienstanbieter, vertrauenswürdiger 3.Partei
- verteiltes Vertrauen: Nutzer vertraut, dass eine teilmenge der beteiligten Entitäten nicht kooperiert

Angreifermodell (hilft vergelichbarkeit protokollen): weshalb, was kann er, welche kosten für angreifer

- Angreifer "Dolev-Yao" omnipräsent im Netz
- kann nicht entschlüsseln/verschlüsseln ohne Schlüssel

Angreiferziele:

- Abhören von Daten: Daten im Sensornetz vertraulich
- Modifizieren von Daten: Daten integer
- Maskerade + Erzeugen von Daten: Daten authentisch

2.2 Ansätze zum Schutz

Anwendungsabhängige Anforderungen sind Geheimhaltung/Schutz von privaten Daten (kryptographische Verfahren) und Verwendung privater Daten zur Diensterbringung, zB Verkehrsvorhersage. Dabei nur sammeln der nötigen Daten und privatsphärengerechte Verarbeitung. Also Diensterbringung mit Prinzip der Datensparsamkeit. Es werden Samples erfasst, dies sind Messgrößen die zu einem bestimmten Zeitpunkt mit Zeitstempel beobachtet wurden. Erforderlich für Diensterbringung. Dabei können Samples nicht mit den Nutzern oder untereinander in Verbindung gebracht werden. Sie geben also nur wenig privates preis (Datensparsamkeit). Verschleierung von Samling Werten durch Herabsetzen der Präzision auf Minimum oder Störwerte hinzufügen. Zentralisierte Datensenken vermeiden, besser lokale Peer-to-Peer Netze zur Diensterbringung. Mehrere nicht kooperierende Dienstanbieter mit beschränkter Sicht. Identität der Quelle verschleiern (Pseudonyme, Datenaggregation). Unverkettbarkeit von Samples durch Samplingrate und Samplingzeitpunkte geschickt wählen (niedrige Samplingrate, zeitliche Muster vermeiden).

Kategorisierung der Ansätze

- Entwurfsstragien: Ansatz um bestimmtes Entwurfsziel zu erreichen
  1. Datenorientiert
     - MINIMISE: personenbezogene Daten auf erforderliche Minimum reduziert
     - HIDE: Personenbezogene Daten und Beziehungen verborgen
     - SEPARATE: Unterschiedliche personenbezogene Daten separat, dezentral gespeichert + verarbeitet
     - AGGREGATE: personenbezogene Daten aggregiert verarbeitet
  2. Prozessorientiert
     - INFORM: Personen über Verarbeitung personenbez. Daten informiert (Transparenz)
     - CONTROL: Personen behalten Kontrolle über Personenbezog. Daten (Intervernierbarkeit)
     - ENFORCE: rechtskonforme Datenschutzrichtlinie erstellt und durchgesetzt
     - DEMONSTRATE: Einhaltung der Datenschutzrichtlinie demonstriert werden
- Entwurfsmuster (anonyme Kommunikation)
- Privacy-enhancing technologies: Implementierung Entwurfsmuster (onion routing -> anonyme Komm.)

2.2.1 im ioE

Technologie greift stark ins private Umfeld ein, dadurch ist die Privatsphäre stärker gefährdet als im klassischen Internet. Beispiele Netatmo, Wetterstation für das Smartphone, Lifelogging Armbänder. ioE hat spezifische Herausforderungen an den technischen Schutz der Privatsphäre. Es sind heterogene Geräte, bei denen Energie, Rechenkapazität, Speicher für klassische Algos nicht ausreicht. Häufig ist keine zentrale Infrastruktur nutzbar (keine Public key Infrastruktur). Es sind mehr Daten, da viele Dinge beteiligt sind, allgegenwärtige kontinuierliche Datenerfassung. Das Vertrauensmodell ist oft unklar (gegen wen schützen, wer vertrauen) und das Angreifermodell anders (Geräte können von Angreifern korrumpiert werden). Geräte befinden sich häufig an leicht zugänglichen Stellen, Angreifer kann physisch auf Geräte zugreifen. Dadurch leicht korrumpieren, Speicher auslesen, umprogrammieren. “tamper-proof“ Hardware für Sensoren teuer. Angreifer findet Implementierungsfehler im Netzwerk Stack (Buffer Overflow) und kann Geräte fernsteuern mit Viren oder Würmern. Angreifer korrumpieren Menge von Geräten, die als byzantinische Systeme zusammenarbeiten -> Herausforderung für Kommunikationsprotokolle. Erbringe Dienst sicher in Gegenwart korrumpierter Geräte.

Angreifer = Insider Wichtig dabei ist die Anzahl der korrumpierter Geräte, je mehr

korrumpierte Geräte desto schwieriger Sicherheit. In Realität nicht alle Geräte korrumpieren, kostet Angreifer. Netz mit n Geräten Angreifer n' < n, β % korrumpiert.

Device Democracy: verteilte Cloud, sicherer Nachrichtenaustausch zwischen Geräten (lokal, global (ip)): PTP über Tor. Robuste und skalierbare Koordination von Geräten. Ohne Vertrauensanker -> dezentrale Konsistenz, Ansatz mit Blockchain Netzen (Bitcoin dezentrales Zahlungssystem ohne Vertrauensanker)

Blockchain Blockchain ist wie ein schwarzes Brett, auf dem man nur schreiben aber nicht löschen kann. Dabei hat jeder Eintrag einen Zeitstempel. Realisiert durch PTP Netz und alle Peers kennen die komplette Blockchain. Vertrauen in individuelle Peers nicht nötig, stattdessen Vertrauen dass keine Gruppe von böswilligen Nutzern existiert, die zusammen über Mehrheit einer Ressource im PTP Netz verfügt. Einträge werden Transaktionen genannt. Neue Transaktionen werden an alle Peers geflutet, sind jedoch noch nicht Teil des Blockchain. Blocks sind die eigentlichen Bausteine der Blockchain, diese enthalten mehrere Transaktionen und eine Referenz auf den vorherigen Block (Blockchain). Mining ist das Erstellen von gültigen Blocks, aber Mining ist schwer und benötigt Ressourcen (Rechenleistung, Lösen Kryptopuzzlen). Widersprüchliche Transaktionen werden beim Erstellen von Blocks ausgeschlossen. Problem bei parallel erstellten Blocks -> Blockchain-Forks. Die Auflösung ist, dass die Blockchain die längste bekannte Kette von validen Blocks ist, dies kann durch jeden Peer verifiziert werden. Einsatz zB in Namensdiensten, robuste Zeitstempel für Messwerte, finanzielle Gegenleistung durch Mikropayments. Transaktionen können beliebig komplexe Regeln enthalten, die von Minern überprüft werden (Smarte Verträge die sich von alleine ausführen).

2.2.2 Beispiele

Privatheit beim Smart Metering. Ziel ist der Verbrauch von Energie in Energienetzen (topologisch) oder kundenspezifisch (demografisch) in Echtzeit nachvollziehbar. Stromzähler senden über Internetanbindung des Kunden Messdaten an Messdienstleister, der diese Daten zur weiteren Verwendung bereitstellt. Periodische Messdaten -> Pseudonymisierung. Messdaten mittels falscher Identitäten übertragen. Aber die Pseudonymverwaltung ist aufwändig/problematisch. Profil des Pseudonyms ist gefährlich, indem einfach mittels externer Daten die Identität verknüpft wird. Übertragung problematisch, IP-Adresse kann mit Identität verknüpft werden (Webserver Logs). Andere Lösung ist die Modifikation des Energiebedarfs, durch gezieltes Laden und Entladen von Akkus. Aber das Potenzial der Verharmlsoung von Akkukapazität abhängig (kostenintensiv). Laden/Entladen verbraucht Energie und ist außerdem komplex. Soll ein konstanter Stromverbrauch oder soll randomisiert werden, leere Akkus...

Anwendungsspezifischer Ansatz

- Aggregation (über viele Haushalte, langen Zeitraum) bevor Daten übermittelt

Aggregation über Zeit einfach. Der Stromzähler aggregiert auf Register, aber es müssen vertrauenswürdige Zähler vorhanden sein. Dabei werden komplexe Tarife zB mit Nachtstrom mit mehreren Registern implementiert. Die Aggregation über mehrere Haushalte ist schwierig, da Daten die aggregiert werden darf Datensenke nicht erfahren. Mögliche TTP? Lösung ist falsche Daten an Datensenke zu schicken, nach Aggregation muss Ergebnis wieder korrekt sein. Es gibt 2 Vorgehen: Ohne Kooperation der Stromzähler untereinander und mit Kooperation.

Ohne Kooperation Hinzufügen von planbaren Rauschen. Es wird Laplace Rauschen auf Messwerte angewendet, der einzelne Rauschwert ist zufällig. Rauschwerte eliminieren sich gegenseitig bei Summenbildung vieler Teilnehmer. Jedoch sehr viele Teilnehmer notwendig. Keine feingranulare Messung möglich, nicht praktikabel.

Mit Kooperation Benötigt Protokoll zur Kommunikation zwischen Stromzählern. Anforderungen sind: Einzelne Messwerte vor MDL geschützt, nur Aggregat ermittelbar. Robuster Betrieb bei Ausfall einzelner Stromzähler oder der Kommunikationsanbindung. Realisierbarkeit auf ressourcenbeschränkter Hardware. Aktuelle Ergebnisse = kurze Dauer, kurze Messintervalle.

Homomorphe Verschlüsselung Rechnen mit verschlüsselten Daten sinnvoll möglich. Vorgehen:

- Stromzähler verschlüsseln eigenen Wert mit Public Key des MDL
- Verschlüsselter Wert entlang Baumstruktur weitergegeben
- jeder eintreffende Wert wird aggregiert mit eigenem und weitergegeben
- MDL enthält verschlüsselten Wer den er entschlüsseln kann
- kein Stromzähler erfährt Wert der anderen

Probleme kooperativer Stromzähler

- teure Kryptografie (aber ressourcenbeschränkte Hardware)
- Robustheit gegen Störung
  - Hardware/Software Probleme
  - Kommunikationsinfra
  - falsche Messergebnisse
  - Totalausfall
  - eingeschränkter schutz privatsphäre

- Strukturvorgaben (Baum) ermöglichen Attacken durch MDL (korrumpierte Stromzähler)
- Lange Dauer Kooperation pro Messwert (lange Messintervalle realisierbar)

Privatsphärengerechtes Smart Metering Protokoll: SMART-ER (Exactness, Robustness). Einteilung der Stromzähler in Gruppen, dies wird vom MDL durchgeführt. Die Gruppengröße ist konfigurierbar. Damit höhere Robustheit und weniger Overhead. Innerhalb der Gruppen Kooperation zur Ermittlung maskierter ungefährlicher Messwerte mit korrektem Aggregat. Pro Messintervall: Austausch von Zufallswerten innerhalb der Gruppen; Die Kommunikationspartner werden gespeichert; berechnen maskierter Messwerte; senden maskierter Messwerte + Abhängigkeiten an MDL; Bereinigung empfangener Messwerte durch Messdienstleister

Berechnung:

- A sendet 12 an B (A speichert alle von sich gesendeten Zahlen in einer Aggregation RS)
- A empfängt 2, 4 von B, C (A speichert alle erhaltenen Zahlen in einer Aggregation rv)
- Messwert m wird ermittelt: 7
- Datenübertragung an Messdienstleister: rv + m - RS -> 6 + 7 - 12 = 1
- Messdienstleister bildet Summe über alle verschleierten Messwerte und erhält dadurch Aggregat

Evaluation Privatsphärenschutz

- Smart Meter Privacy Break Game
- Angreifer stellt 2 Szenarien mit gleichem Aggregat
- Verteidiger wählt geheim 1 Szenario + führt Smart Metering durch
- Angreifer erhält Info entsprechend Angreifermodell (alle Kommunikation abhören, MDL korrumpieren)
- Maß Privatsphärenschutz: Wahrscheinlichkeit Angreifer richtiges Szenario (p = 0.5 -> bester Schutz, Angreifer rät; p = 1 -> kein Schutz)

MDL ist korrumpiert, dann kann Angreifer mit abgegeben, maskierten Werten arbeiten. Vrot = rv + m - RS und Vblau; Summe Vrot + Vblau gewünschtes Aggregat, dabei ist die Differenz d von Vrot - Vblau interessant. Angreifer kann nur raten. Problem MDL nimmt die Gruppenbildung vor, also ist ein Angriff mittels korrumpierter Stromzähler möglich. Dabei werden Gruppengröße - 1 korrumpierte Stromzähler benötigt. Eine Gegenmaßnahme ist die dezentrale Gruppenbildung, um den Einfluss des MDL

auf Gruppenbildung zu vermeiden. Eine Methode ist das Smart Meter Speed Dating, bei der die Gruppen von den Stromzählern dezentral selbst ermittelt werden. Hohe Robustheit durch kleine Gruppengröße, wenige Hardwareanforderungen (impl im Sensornetz), Skalierbarkeit eingeschränkt durch Speicher. Oder die Methode Elderberry, welches ein baumbasierter Ansatz mit strukturiertem P2P Overlay ist. Mit dezentraler Datensenke, wird diese entlastet. Jedoch komplexer als 1.Methode, aber sehr gut skalierbar.

Privatheit im Smart Traffic Zur Verkehrsoptimierung wird die automatische Fahrzeugnavigation entwickelt. Diese kann statisch sein, dabei ist die Routenplanung mit fixem Kartenmaterial. Adaptiv dabei wird der aktuelle Verkehrszustand mit einbezogen. Oder koordiniert, dabei wird die geplante Route anderer Verkehrsteilnehmer miteinbezogen oder eigene Pläne veröffentlicht. Es werden Sensordaten mobil erfasst (Wetter, Verkehr (adaptive Routenplanung)), die Herausforderung ist, dass die Samples auch die Position enthalten. Beispiel Vehicular Clouds Fahrzeuge als Diensterbringer (Pics-on-wheels). Herausforderung ist wie die Anfrage an geeignete Fahrzeuge weitergeleitet wird (hängt von der Position ab).

Positionsbezogene Daten Werden bei vielen Smart Traffic Anwendungen gebraucht zur Bestimmung des Verkehrszustands -> Floating Car Data.

- Lifelogging
- Location based Services: Inhalten basierend auf geografische Lokation, Check in Anwendungen, Google Latitude
- Crowdsensing

Position kann geografisch repräsentiert werden (Breiten/Längengrad) oder Kontextspezifisch (hierarchisch), mit variabler Präzision. Zeit zu der Sample erstellt, geteilt wurde. Nutzer hat potentiell Einfluss auf Samplingrate und kommuniziert mit Mobilfunk Basis (Implizit). Explizit mit GPS, Nutzer hat Einfluss auf Genauigkeit (Verschleierung). Präzise zu Nutzern zuordbare Positionssamples lassen Rückschlüsse auf Lebensgewohnheiten zu. Selbst wenn Samples nicht explizit zu Nutzern zuordbar, kann implizit Zugeordnet werden mit Kontextwissen.

Privatsphärenschutz Trennung von Positionssamples und Identitäten durch Verwendung von Pseudonymen. Volle Anonymität nicht möglich, ermöglicht unbestrafbaren Missbrauch der Dienste. Pseudonyme mit limitierter Anzahl pro Teilnehmer. Verfahren zum Ausschließen böswilliger Teilnehmer (Zuordnung durch Vertrauensanker). Vorsicht bei Pseudonymwechsel ist die Zuordnung über die Kommunikationsadresse, über Positionssamples vor und nach Wechsel. Die Zuordnung von Positionssamples zueinander verhindern -> Pseudonyme oft wechseln, effektiver wechsel nicht trivial. Verschleierung Samples. Es gibt die zeitliche Verschleierung, dabei werden Positionsupdates zufällig verspätet und die Update Intervalle verringert. Und es gibt die räumliche Verschleierung, dabei werden Positionswerte um zufälligen Faktor verschoben (geringere Präzision) oder

Dummy Werte (mehrere falsche zusammen mit echter Position). Zeitliche Verschleierung: Virtual Trip Lines. Virtuelle Induktionsspulen im Straßengraphen (unbedenkliche Orte), an denen Positionssamples geteilt werden. Mix Zones alles außer Trip Lines, geringe Samplingrate. Räumliche Verschleierung wegen Kontext Wissen und Map Matching nicht ganz so einfach. Zufällig lange Funkstille nach Pseudonymwechsel, besser ein Mix. Dies sind Mix Zonen, vordefinierte Gebiete mit hohem Verkehrsaufkommen und niedriger relativer Geschwindigkeit der Verkehrteilnehmer zueinander (Kreuzungen, Parkplätze). Verkahrsteilnehmer wechseln nur Pseudonym wenn sie Mix Zone passieren (Innerhalb Mix Zone keine Positionssamples geteilt). -> Natürliches Mixen der Pseudonyme, je mehr Mix Zonen passiert, geringer Wahrscheinlichkeit dass Angreifer Positionssamples in Verbindung bringt. Anfragen an fremde Fahrzeuge im geografischen Gebiet -> Geocast, zentraler Server schlecht. Mit Overdrive PTP Overlay Netz direkt zwischen den Fahrzeugen (Datenlokalität (entfernte Nachbarn verfälschte Pos.) und Proximity Tests (Zugehörigkeit durch selbe GSM Zelle)). Vorausschauende Planung Privacy-Preserving Cooperative Route Planning (virtuelle Tokens zur Zugangskontrolle, Promise Coins, blinde Signaturen)

3 Kommunikation

- ZigBee: Smart Homes, Connected Lighting, Utility Industry, Retail Services
- 6LoWPAN: IETF, Internet im Mittelpunkt, basiernd auf IPv6
- Cyber Physical Systems: Vernetzte Komponenten mit Sensoren und Aktuatoren, die physikalische Prozesse steuern. Vernetzte Regelschleifen. Bestandteil des IoE, Hohe Anforderungen an Safety, reibungsloses Zusammenspiel zwischen Regelkreis und Kommunikation. Time Sensitive Networking: Time-synchronized low latency streaming services (Medienzugriff, Scheduling)

3.1 Medienzugriff

3.1.1 Grundlegendes

Funk Recht hohe Unzuverlässigkeit, hohe Fehlerraten. Es is ein geteiltes Medium, Zugriff muss geregelt werden. Anforderungen an Funkbetrieb:

- Wenig Energie verbrauchen, da vernetzte Dinge eine hohe Lebenszeit haben
- Robustheit (häufige Topologieänderungen durch schlafende Systeme)
- Skalierbarkeit (große Anzahl)
- Selbstorganisation
- Sicherheit + Schutz Privatsphäre
- Niedrigere Datenraten als drahtgebunden
- Höhere Verzögerungen, Schwankungen
- Geringere Sicherheit gegen Abhören, aktive Attacken
- Regulierung der Frequenzbereiche: viele sinnvoll nutzbare Frequenzen schon belegt
- Unidirektionale Links

Signalausbreitung Im idealen Fall wird diese nicht durch Hindernisse beeinflusst. Mittlere Empfangsleistung nimmt mit 1/d2 ab. d Abstand Sender Empfänger, Ps, Pe Sende/mittlere Empfangsleistung: Pe 1/d2 Ps. Signalstärke nimmt proportional mit quadratischer Distanz ab. Proportionalitätsfaktor hängt von Frequenz ab. Unterschiedliche Bereiche

- Übertragungsbereich: Kommunikation möglich, niedrige Fehlerrate
- Erkennungsbereich: Signalerkennung möglich, keine Kommunikation

- Inferenzbereich: Signal kann nicht detektiert werden, Signal trägt zum Hintergrundrauschen bei
- sind nicht stark gegeneinander abgegrenzt, Umwelteinflüsse beeinflussen Bereiche

Semi-Broadcast-Medium sind Drahtgebunde Netze (Ethernet) oder Drahtlose Netze. Drahtgebundene Systeme sind über Broadcast- Medium verbunden (Systeme hören gesendete Dateneinheiten aller Systeme, Kollisionserkennung, CSMA/CD für Medienzuteilung). Drahtlose sind Systeme die über Semi-Broadcast-Medium verbunden sind (Systeme hören nur Dateneinheiten von Systemen in der Nähe, Sender erkennt Kollisionen nicht tritt beim Empfänger auf. Problem der versteckten und ausgelieferten Endsysteme. CSMA/CD nicht einsetzbar).

Medienzuteilung Verwendung von Zeitmultiplex, Konkurrierende Verfahren (evtl. mit festen Zeitschlitzen). Besonders ist der Energieverbrauch und die Latenz der Kommunikation zu beachten. Energie sparen indem die Funktransceiver häufig deaktiviert werden (wann empfangsbereit). Beschränkung ist dass kein Duplex Betrieb unterstützt wird (kein gleichzeitiges senden/empfangen). Unnötiger Energieverbrauch ist Kollision beim Medienzugriff -> Kollisionsvermeidung, unnötiges Lauchen (Idle Listening, System wartet auf Dateneinheit obwohl niemand sendet) und mithören (overhearing, empfängt Dateneinheiten die nicht an es gerichtet sind) -> Duty Cycling.

Kollisionsvermeidung

- Separate Signalisierung
  - Out-Of-Band Signalisierung: Auf separatem Kanal, Reservierung fester Zeitschlitze
  - In-Band Signalisierung: Auf gleichem Kanal. Beispiel Multiple access with collision avoidance (MACA) Sender sendet Request to send (RTS), Empfänger antwortet mit Clear to Send (CTS), Sender sendet. Während Handshake Kollisionen möglich, aber geringe Wahrscheinlichkeit.

Funktransceiver bei Bedarf aktivieren um drahtloses Medium auf Aktivität/Übertagung durchzuführen. Sensorknoten aktivieren Funktransceiver koordiniert -> synchrones Duty-Cycling (Zeitsynchro zwischen Knoten). Sensorknoten aktiviert Funk ohne Koordination -> asynchrones Duty-Cycling. WLAN hat hohen Energiebedarf durch Kollisionen, Idle Listening, Overhearing. Bluetooth hat langsamen aufwändigen Synchroprozess zum Verbindungsaufbau (Slow Frequency Hopping). Verbindungsverwaltung kompliziert + teuer. Synchro mehrerer Piconetze problematisch (Scatternetze). Häufige Datenverluste beim Überbrücken Piconetzgrenzen, mehrere Radio-Interfaces behebbar. Bluetooth benötigt Zertfizierung im kommerziellem Umfeld. Bluetooth smart = Bluetooth classic, Bluetooth high speed, Bluetooth low energy. Bluetooth low energy:

- Reichweite 150m

- Datenrate 1 Mbit/s optimiert für kleine Portionen von Daten (8 -27 B)
- gut für Übertragung von Zustandsinfos
- Senden Daten nur in sehr kurzen Zeitspannen
- adaptives Frequenzsprungverfahren im 2,4 GHZ Band
- viel Intelligenz im Controller
- versucht nicht dauerhaft neue Geräte zu finden -> übernimmt Master (mehr Energiebedarf)
- eingebaute Sicherheitsfunktionalität
- Linzensfrei
- optimierte Hardware (geringere Latenz, kurze Übertragungsdauer, geringe Leistungsaufnahme) -> energie-effizienterer Betrieb

Klassiche MAC Protokolle(akademisch):

- S-MAC: zeitliche Synchro der Knoten
- B-MAC: keine zeitliche Synchro der Knoten

3.1.2 Medienzugriffsprotokolle

S-MAC

- Energiebedarf gerring durch Vermeidung unnötigem Lauchen, vermeiden Kollisionen, vermeiden Overhearing
- gute Skalierbarkeit
- Autokonfiguration
- weniger Berücksichtigt: Fairness, Latenz
- Koordiniertes Schlafen vermeidet idle listening, erwachen gleichzeitig zur Kommunikation
- synchro der Systeme durch Diskretisierung der Zeitachse in Rahmen fester Länge
- Jeder Rahmen besteht aus 2 Phasen fester Länge
  - Listen Phase: Synchro + Datenaustausch anstoßen
  - Sleep Phase: System schläft oder Datenaustausch (Energie sparen)

Strukturierung der Phasen, indem die Listen Phase in Zeitschlitze gegliedert wird und pro Zeitschlitz max. 1 Dateneinheit, konkorrierender Zugriff auf Zeitschlitze. Sleep Phase ohne Gliederung. In der Listen Phase ist das System im Wachzustand, in der es noch eine Sync und eine RTS/CTS Phase gibt. In der Sync Phase wird durch SYNC Dateneinheit synchronisiert (sende sync in zufälligem Zeitschlitz, sync dateneinheit enthält Zeitspanne bis zum Beginn der nächsten Sleep Phase). In der RTS/CTS Phase wird der Datenaustausch angestoßen (wähle Zeitfenster RTS zufällig, dann Carrier Sense + RTS/CTS, max. 1 Dateneinheit). Sleep Phase entfällt falls Datenaustausch angestoßen wurde, sonst Funkschnittstelle temp. deaktivieren. Zeitdauer der Listenphase tlisten muss ausreichend lang sein um SYNC, RTS, CTS übertragen zu können. Länge ergibt sich aus MAC, PHY Schicht (datenrate) ist also nicht frei wählbar. Duty Cycle ist der Zeitanteil den ein System im Wachzustand verbringen soll., frei wählbar. Duty Cycle = tlisten/ trahmen Rahmenlänge ist die Zeitdauer eines Rahmens -> tlisten + tsleep

Synchronisation

- System muss Anfang der nächste Sleep Phase kennenlernen
- System lernt Zeitplan Nachbarn durch regelmäßigen Austausch SYNC Dateneinheiten
- falls Nachbarn nicht bekannt: System wählt eigenen Zeitplan, sonst folge gelernten Zeitplan (entstehen zeitl. synchronisierte Inseln mit eigenem Zeitplan)
- Systeme an Inseln Grenzen lernen + befolgen zusätzlich Zeitpläne benachbarter Inseln (Grenze überbrückt, mehr Energie aufwenden)

Message Passing Übertragung großer Einheiten von Nutzdaten, jedoch steigt die Bitfehlerwahrscheinlichkeit mit Länge der Dateneinheit. Message ist Menge von zusammengehörenden Nutzdaten. Message in einer Dateneinheit, große Wahrscheinlichkeit für Neuübertragung. Message in mehrere kleine Dateneinheiten, kleinere Wahrscheinlichkeit für Neuübertragung Message. Jede Dateneinheit erzeugt Overhead mit RTS/CTS. Message Passing fragmentiert Message in mehrere Dateneinheiten und überträgt alle Dateneinheiten als Burst nach einem RTS/CTS. Jede Dateneinheit einzeln bestätigt, bei Bitfehler nur fehlerhafte neu übertragen. Erweiterung Adaptive Listening:

- Nach Übertragung einer Dateneinheit zusätzliche Phase um neuen Datenaustausch anzutauschen -> Adaptive Listening Phase
- Woher weiß schlafendes System dass Übertragung abgeschlossen

Kenngrößen zur Leistungsbewertung Durchschnittlicher Energiebedarf pro Byte E= Gesamtenergiebedarf aller Systeme geteilt durch Anzahl von Senke empfangene Bytes. Durchschnittliche Ende-zu-Ende Verzögerung V = Summe aller Ende-zu-Ende Verzögerungen geteilt durch Anzahl der Dateneinheiten. Ende-zu-Ende Goodput G = Gesamtzahl von der Senke empfangene Bytes geteilt durch Zeitspanne zwischen Versenden der 1.Dateneinheit bis zum Empfang der letzten Dateneinheit an der Senke.

B-MAC Energieeffizienter Betrieb durch Kollisionsvermeidung und effizientere Kanalnutzung bei hohen und niedrigen Datenraten. Skalierbarkeit bzgl. Anzahl der Systeme, Toleranz gegenüber sich ändernder Funkbedingungen, Rekonfigurierbarkeit durch Protokoll der Vermittlungsschicht. Einfache Impl., geringe Codegröße, geringer Speicherverbrauch. Periodisches Prüfen des Kanals, statt zeitlicher Synchro. Problem der versteckten Endgeräte nicht behandelt. keine Fragmentierung großer Nachrichten.

Low Power Listening = Ruhezustand(empfangsbereit)

- System schläft meistens
- erwacht kurz um Kanal auf Daten zu überprüfen
- wach bleiben, falls Daten zum Empfang

Clear Channel Assessment = Senden

- prüfe ob Kanal frei
- übertrage Präambel und Daten

Präambel stellt sicher, dass Empfänger Übertragung nicht verpasst (Intervall zwischen aufeinanderfolgenden Kanalüberprüfungen legt Präambel Mindestdauer fest). B-MAC fast 4,5 facher Durchsatz als S-MAC unicast (Geringerer Rechenaufwand pro Dateneinheit, CCA effektiv)

- Durchsatz über Verzögerung zwischen zwei übertragenen Paketen gesteuert. B-MAC CCA Sampling Intervall, Energiebedarf wächst sublinear. S-MAC Duty-Cycle, Energiebedarf wächst linear (Overhead SYNC Periode)
- Erweiterung X-MAC: Lange Präambel oft ungünstig, sende viele kleine Präambelpakete
  - für Unicast: Integriere Zieladresse in kurze Präambelpakete

IEEE 802.15.4 Definiert physikalische Schicht und Medienzugriff, wird bei ZigBee und 6LoWPAN verwendet. Ziele:

- Kleine - mittlere Datenraten
- Moderate Verzögerungen
- Geringerer Energiebedarf
- geringe Komplexität
- Anwendungen: Sensoren, interaktive Spielzeuge, Smart Badges, Fernsteuerung, Gebäudeautomatisierung

- kombiniert zeitplan basiertes und konkurrenz basiertes Verfahren (realzeitfähigkeit durch Zeitschlitze)
- asymmetrisch: Systeme können unterschiedliche Rollen annehmen
- basiert auf CSMA/CA

2 verschiedene Typen von Netzen möglich: PTP Netze (Mesh) und Stern Netze (Star). 2 verschiedene Klassen von Systemen: vollfunktionstüchtige Systeme, eingeschränkte Systeme (Stern Netzen). Jedes Netz besitzt spezielles FFD den Koordinator, der das Netz mit periodisch versendeter Beacons synchronisiert. 2 Modi: Beacon Modus und Non-Beacon Modus.

Beacon Modus Im Stern Netz sind Systeme dem Koordinator zugeordnet, diese formulieren ein PAN. Der Koordinator führ Buch über Systeme und Vergabe von Adressen und sendet regelmäßig Beacons (PAN Identifikator). Er bearbeitet Anforderungen für garantierte Zeitschlitze und vermittelt zwischen Systemen und Peer Koordinatoren. Annahme: Koordinator ist Energie keine limitierende Ressource und übernimmt mehr Aufgaben. Die Rahmen haben eine feste Länge, bei der in der inaktiven Phase alle Systeme den Transceiver ausschalten können. Die aktive Phase hat eine Dauer von 16 Zeitschlitzen bei der der Koordinator aktiv sein muss. CAP: Contention Access Period mit Zugriff durch Slotted CSMA/CS (Konkurrierender Zugriff). GTS: Guaranteed Time Slot, müssen beim Koordinator angefordert werden(Reservierung bis Zeitschlitze wieder freigegeben, oder Koordinator feststellt dass Schlitz best. Dauer nicht benutzt wurde), max. 7 aufeinanderfolgende Zeitschlitze.

CAP Slotted CSMA/CA regelt Medienzugriff. Kanal abhören (CS) bevor auf Kanal gesendet, kein Schutz bei versteckten Systemen (kein RTS/CTS). CA Exponentieller Backoff zur Kollisionsvermeidung. CAP Zeitschlitz in Backoff Perioden (BP) untergliedert. NB Anzahl bisher durchgeführter Backoffs, CW Contention Window, BE Backoff Exponent

GTS GTS Anfrage (Unterschied in Sende/Empfangszeitschlitz), dann Antwort Koordinator in 2 Schritten: Unverzügliches Ack bestätigt Empfang und zugeordnete GTS-Zeitschlitz in nächsten Beacon bekanntgeben (verzögern falls Ressourcen nicht ausreichen)

Datenübertagung System zu Koordinator: Wenn System GTS Zeitschlitz reserviert hat, kann direkt nutzen (Dateneinheit + Inter Frame Space + ACK müssen in Zeitschlitz passen) sonst sende während CAP mittels CSMA/CA. Koordinator zu System: Falls System Zeitschlitz (receive) reserviert hat, nutzt dieser der Koordinator, System antwortet mit ACK. Sonst integriert Koordinator Sendewunsch in Beacon (Adresse des Systems im pending address Feld), System antwortet mit Datenanfrage, Koordinator sendet ACK und sendet Daten. System quittiert mit ACK. Bei Fehlern wiederholt System die Anfrage während nächstem Rahmen.

Non-Beacon Modus Keine feste Rahmenstruktur, Zugriff auf Medium mit unslotted CSMA/CA (wegen fehlender Zeitsynchro keine Zeitschlitze möglich). Koordination/Synchro nicht Teil Standards. PTP Netz Systeme können untereinander frei kommunizieren, formieren PAN. PAN Koordinator wird frei gewählt, zb 1. kommunizierender Knoten

4 Routing

Annahmen drahtlose Sensornetze:

- Adressierung anders, globales Adressierungsschema wenig praktikabel
- Concast: Daten von vielen Quellen an eine Senke
- Dissemination: Daten von einer Quelle an viele Senken -> wenig knoten zu knoten Kommunikation
- potentiell viel Redundanz (Temp.werte)
- Ressourcen stark limitiert (Energie, Speicher)

4.1 probalistische Verfahren

4.1.1 Fluten

Knoten sendet jede Dateneinheit per Broadcast an alle Nachbarn (Broadcast-MAC-Adresse). Empfangene Knoten leiten Dateneinheiten an alle ihre Nachbarn weiter. Es ist dezentral, selbstorganisierend, keine Routinetabellen, hohe Netzlast, häufig im Einsatz. Vorteil ist, dass keine Routenfindung notwendig, keine Wartung der Topologie erforderlich und keine Routingtabellen (keine Zustandshaltung/wartung). Nachteile sind dass Dateneinheiten dupliziert versendet werden (Implosion), limitierte Ressourcen nicht berücksichtigt werden (hohe Netzbelastung, viele Kollisionen, hoher Energieverbrauch). Die Dateneinheiten terminieren nicht, kreisen unendlich lange im Netz, keine Zuverlässigkeit.

Verbesserung Dupliktvermeidung Ziel ist es, dass Sensorknoten neu empfangene Dateneinheiten genau einmal weiterleiten. Alle Sensorknoten müssen Zustandshaltung über bereits weitergeleitete Dateneinheiten betreiben, Dateneinheiten eindeutig identifizierbar. Die Anforderungen sind in dezentralen, Selbstorg. WSAN nicht immer einfach erfüllbar, denn Erkennen duplizierter Dateneinheiten problematisch und beschränkter Speicherplatz. Einsatz probabilistischer Verfahren zur Duplikaterkennung.

Verbesserung begrenzter Reichweite TTL Begrenzung der Reichweite durch Time to Live, wenn Dateneinheit maximalen Hopcount erreicht hat, wird sie verworfen. Vorteile sind geringe Netzbelastung, weniger Kollisionen. Nachteile ist keine Zuverlässigkeit.

4.1.2 Gossiping

Rumor Mongering ist ähnlich der Verbreitung von Gerüchten (erzählt Gerüchte gewisse Anzahl Personen), wird angewahnt auf Weiterleitung von Dateneinheiten. Leite Dateneinheiten an zufällig ausgewählten Knoten weiter, bis ausreichend viele Knoten die Dateneinheit erhalten haben. Knoten leitet Dateneinheit mit gewisser Wahrscheinlichkeit p an Nachbarn weiter -> Broadcast auf MAC-Ebene erreicht alle 1-Hop Nachbarn.

p typisch zwischen 65 und 75 % , wenn p < 65 große Wahrscheinlichkeit, dass Dateneinheit verworfen wird vor Ziel. Es kommt nicht zu einer Implosion von Daten und hat geringeren Overhead als Fluten. Nachteile sind, dass es eine evtl. lange Übertragungszeit gibt, denn die Nachricht kann einen sehr unsinnigen Pfad zwischen Knoten nehmen. Es gibt keine Zuverlässigkeit, denn es erhalten nicht alle Knoten im Netz die Dateneinheit (feature)

Fluten und Gossiping Auf den ersten k Hops fluten, dann Gossiping mit Wahrscheinlichkeit p. Als GOSSIP(p,k) bezeichnet. GOSSIP(1,k) -> Fluten, GOSSIP(p,0) -> normales Gossiping. Erfahrungswerte p = 0,72 und k = 4 (fast alle Knoten Daten)

variierendes p Wahrscheinlichkeit p steigt, je näher Dateneinheit Ziel kommt. Voraussetzung: Jeder Knoten kennt Distanz bis zum Ziel (Ziel kann zb erfolgreich empfangene Daten quittieren, damit Distanz bekannt geben). pri: Gossiping Wahrscheinlichkeit für eine Dateneinheit auf dem i ten System Ri in Richtung Ziel. k Faktor der Wahrscheinlichkeit bestimmt. pro = (1 + k) pri-1 näher zum Ziel, (1-k)pri-1 weiter weg vom Ziel, pri-1 gleichbleibend, unbestimmt.

4.2 inhaltsbasierte Verfahren

4.2.1 Directed Diffusion

Ist ein datenzentrisches Verfahren, klassische Adressen spielen keine Rolle Daten im Mittelpunkt. Basiert auf Fluten, selbstorganisiert. Findet Pfade von Ereignisquelle zu einer Senke, unterstützt Aggregation von Daten. Daten werden durch Attribut-Wert-Paar benannt. Beobachtungsaufgabe wird als Interesse im Netz verbreitet (Diffusion), Gradient wird dadurch in den Knoten im Netz etabliert, Gemessene Ereignisse folgen mehreren Pfaden zum Interessenten (Senke) -> Verstärkung einer weniger Pfade. Geeignete Szenarien sind: Eine Datenanfrage - mehrfache/ regelmäßige Antworten (regelm. Temperaturmessung). Etablierung einer Infrastruktur zahlt sich evtl. aus (Interessen teuer, Ereignisse billig hinsichtlich Kommunikationsaufwand). Sensorknoten die eine Datenanfrage beantworten können sind unbekannt. Mehrere Sensorknoten können selbe Daten anfragen. Senken äußern Interesse an Daten (Query), Interesse wird periodisch erneuert, Soft-state und selbstorganisierend (Zeitstempel monoton erhöht). Interesse wird im Netz geflutet. Aufbau eines Pfades zur Datenquelle, speichern die Richtung aus der Interesse kam (Gradient (enthält info über Datenrate) kein Bezug zur Quelle). Zu einer Query können mehrere Gradienten existieren. Daten werden auf Rückwärtspfad gesendet, also von Ereignisquelle zu senke. Bei der Gradientenverstärkung wird nach Ereignisquellen gesucht, dabei werden anfangs Gradienten mit niedriger Datenrate etabliert (niedrige Stärke, mehrere Gradienten pro Knoten und Interesse möglich). Weiterleitung der Ereignisse erfolgt an alle Gradienten, Weiterleitung auf mehreren Pfaden, mehrfaches Senden. Verstärkung falls Ereignisquelle gefunden -> Senke startet Reinforcement Phase. Datenrate zu diesem Interesse erhöht -> Stärke Gradient steigt. Ist angeforderte Stärke höher als verfügbare -> Weiterleitung zu einem Nachbarn in Richtung Ereignisquelle. Dadurch

werden Pfade zwischen Ereignisquelle und senke etabliert und die Weiterleitung erfolgt nur noch auf diesen Pfaden, der Pfad mit geringer Verzögerung wird gewählt. Anpassung an Änderungen der Datenquelle, Topologie, Senke möglich. Directed Diffusion ist reaktives Routing (Pfad als Reaktion auf Interessenbekundung etabliert). Der Pfadaufbau geschieht in 2 Phasen: phase 1 redundantes Senden von Daten Phase 2: Reduktion der Pfade durch Gradientenverstärkung (Datentransfer in beiden Phasen). Der Datendache ist zur Dupliakterkennung da, Vermeidung von Schleifen und Anpassung der Datenrate. Nur lokale Kommunikation (keine Ende-zu Ende Kommunikation). Beim traditionellen Routing wird ein schleifunfreier Pfad aufgebaut, dann erst Datentransfer und es ist eine Ende-zu Ende Kommunikation. Directed Diffusion deutlich besser als Fluten.

- Auslieferungsrate bei Knotenausfällen
- Auslieferungsrate = Anzahl unterschiedlicher empfangener Eregnisse/ Anzahl ursprünglich versendeter Ereignisse

4.2.2 Rumor Routing

Datenzentrischer Ansatz, aber Initiative geht von Ereignissenke und quelle aus. Kompromiss zwischen Fluten von Anfragen und Fluten von Ereignissen. Es ist ein agentenbasierter Ansatz, bei dem Ereignis Agenten Pfade zu Ereignissen etablieren. Anfragen suchen im Netz nach Ereignis-Pfaden. Ereignisagenten besitzen Informationen über Ereignisse und werden bei neuem Ereignis mit gew. Wahrscheinlichkeit erzeugt. Sie wandern als langlebige Dateneinheit durchs Netz und können auf ihrem Weg Informationen lernen, max Lebensdauer (Anzahl von Hops). Sie hinterlassen Pfadinformationen in den Knoten (Ameisen). Beteiligte Sensorknoten benötigen Informationen über Nachbarn und Ereignisse. Nachbarschaftsinformation (periodische Hello Dateneinheit) und Ereignistabelle (Weiterleitungsinformation zu bekannten Ereignissen, Updates durch ankommende Ereignis Agenten, Einträge limitierte Lebensdauer). Prinzip funktioniert, weil Wahrscheinlichkeit hoch, dass sich in einem Rechteck zwei zufällige Pfade schneiden. -> Trade off zwischen Erfolgswahrscheinlichkeit und Energieverbrauch. Etablierung mehrerer Pfade benötigt mehr Energie. Rumor Routing, wenn viele Events und wenig Anfragen.

Random Walks Leite Dateneinheit an zufällig gewählten Nachbarn per Unicast weiter (Unicast-Mac-Adresse). Alternative Form ist, leite mehrere Kopien der Dateneinheit parallel an zufällige Nachbarn (Gerüchte). Random Walk sowohl für Anfragen (Suchagenten) und Ereignisagenten. Trifft Anfrage auf Ereignispfad, gezieltes Weiterleiten.

4.3 lokationsbasierte Verfahren

In vielen Anwendungen Orte adressiert werden und die Information der Quell-, Ziel-, Zwischensysteme kann Routing unterstützen (evtl. Abbildung zwischen Systemname und Position nötig, evtl. auf Routingtabelle verzichten, Positionsangaben können Routingalgos vereinfachen (Richtung)). Vorteil falls Position äquivalent zu Adressen gesehen

werden, gibt es eine implizite Ordnung auf Adressen und daher geeignet für greedy Strategien.

4.3.1 Distanzbasiertes Greedy Verfahren

Die Dateneinheit soll zu einer bekannten Position übertragen werden unter der Annahme, dass jedes System seine Position kennt. Strategie most forward within r:

- r ist Kommunikationsradius -> Nachbarn
- Greedy Verfahren: Leite Dateneinheit an Nachbar System weiter, das Ziel am nächsten ist
- Strategie ist frei von Schleifen
- Topologie bleibt unbeachtet (nicht immer kürzester Pfad)
- System am Rand der Übertragungsreichweite bevorzugt

4.3.2 Richtungsbasiertes Greedy Verfahren

Wähle Next-Hop System möglichst nahe an der idealen Richtung zum Ziel (Ideale Richtung = Verbindungsgerade), jedoch nicht frei von Schleifen. 2 Metriken: Minimaler Abstand zur Verbindungsgerade und minimaler Winkel zur Verbindungsgerade. Das Problem ist, dass Distanz und Richtungsbasierte Greedy Strategien bleiben in Sackgassen stecken (lokales Extremum).

4.3.3 GPSR

Greedy Perimeter Stateless Routing.

- Rechte Hand Regel erlaubt aus Sackgasse (Labyrinth) zu entkommen
- Weiterleitung in 2 Modi
  - Greedy Modus: Weiterleitung durch distanzbasiertes Greedy Routing solange möglich (keine Sackgasse)
  - bringt Dateneinheit schnell dem Ziel näher
  - Perimeter Modus: Weiterleitung durch Perimeter Routing falls Sackgasse vorliegt
  - Ermöglicht Entkommen aus Sackgassen
  - Nutze Rechte Hand Regel um Dateneinheit um Hindernis zu leiten
- GPSR wechselt zwischen Greedy und Perimeter Modus

Face + Perimeter

- Face = Größtmögliche Fläche einer Ebene, welche nicht von Kanten zwischen benachbarten Systemen geschnitten wird
- inferior face: face liegt vollständig innerhalb des Netzes
- exterior face: face umschließt das Netz
- Notation: folge der Knoten, welche beim Umlaufen der Face gegen Uhrzeigersinn besucht werden
- Perimeter: Menge der Systeme, die ein Face definieren

Ablauf

- Initial Greedy Modus
- Perimeter Modus: Position Eintrittspunkts in Perimeter Modus wird in Dateneinheit notiert
- Weiterleitung um face welche von der Verbindungsgeraden zwischen aktuellem System und Zielposition geschnitten wird
- leite Dateneinheit gemäß Rechte Hand Regel entlang entsprechenden Perimeter weiter (um face fläche herum)
- Wechsel in greedy Modus sobald weiterleitendes System Ziel näher ist als der Eintrittspunkt in Perimeter Modus
- Perimeter Modus erfordert Planaren Graphen (drahtlosen Netz iR nicht)

4.4 Distanzvektorbasierte Verfahren

- Reaktion auf Änderungen:
- in traditionellen Netzen: Schnelle Reparatur bei Fehlern, finden alternativer Pfade
- in low power and lossy Netzen: Überreaktion vermeiden, kann zu Instabilität und Overhead führen
- Anforderungen:
  - Adaptivität
  - Constraint-based Routing: Bestimmte Links aufgrund Randbedingungen ausschließen
  - verschiedene Verkehrsmuster: Multipoint-To-Point (Concast), point-to-multipoint (multicast), point-topoint(unicast), parallele pfade
  - Konfiguration und Management (so wenig wie möglich)

4.4.1 RPL

Routingprotokoll RPL (ripple) mit Aufbau von DODAGs (Destination-Oriented Distributed Acyclic Graphs) und Nutzung einer Objective Funktion, basiert auf Metriken und Randbedingungen. Verbindung von Sensorbasierten Zugangsnetzen zum Internet über wenige dedizierte Knoten (Wurzeln)

- up: Verkehr Richtung Internet concast
- down Verkehr Richtung Knoten. Multicast
- Gerichtete Azyklische Graphen

Konstruktion DODAG: Startet von der Wurzel, Knoten in der Nachbarschaft empfangen Nachricht, entscheiden ob Beitritt Graph oder nicht. Beitritt = Knoten besitzt Route zur Wurzel (Elternknoten). Falls Knoten Blattknoten ist keine Aktivität, falls er ein Router ist Informationen in Nachbarschaft verbreiten. Dies repräsentiert Concast-Datenfluss. Der DAG wird durch eine DODAG ID innerhalb Instanz eindeutig identifiziert. Varianten RPL Instanzen:

- Einzelner DODAG mit einer Wurzel
- mehrere umkoordinierte DODAGS unabhängigen Wurzeln (mehrere Datensammelpunkte)
- ein DODAG mit virtuellen Wurzel, die durch mehrere physikalische Wurzelknoten vertreten ist

Dynamische Anpassung DODAGs können durch Änderungen im Netz nicht mehr optimal sein (dynamische Anpassung) -> resultiert in unterschiedlichen DODAG Versionen. Anpassung wird durch Wurzel kontrolliert, diese kann Topologieänderungen bewirken (innerhalb DODAG).

Node Rank Sakrale Wert, der relative Position eines Knoten im DODAG angibt (relative position zur wurzel, wächst streng monoton nach unten (zur schleifenvermeidung), gültig innerhalb DODAG Version)

Objective Function Abstrakt formuliertes Pfadkosten-Kriterium (Metriken, Constraints). Definert wie Metriken + Constraints in Node Rank umgewandelt (welche Knoten Eltern, wieviele) -> Jeder DODAG Ausprägung einer Objective Funktion.

RPL Kontrollnachrichten

- DAG Information Object: Informationen, die Knoten ermöglichen RPL Instanzen zu entdecken, Konfigparameter, DODAG Eltern (Node Rank Sender). enthalten eigenen moderank, eigene pfadkosten (pfadkosten über besten Elternknoten). werden initial von der Wurzel gesendet, oder wenn sich der eigene noderank, pfadkosten ändern

- DAG Information Solicitation: Gezielte Anforderung eines DIO von einem Knoten
- Destination Advertisement Object: Um Infos über Routineziel entlang eines DODAG zu kommunizieren

Storing / Non storing nodes Ziel ist die Etablierung von Routinginfos für Down-Richtung (Blattknoten senden DAO in Richtung Wurzel). Problem ist der Speicheraufwand in den Knoten. Also gibt es es non-storing Noldes, die keine Routinginfo halten und Wurzel Source-Routing Info in Dateneinheit einfügt. Storing Nodes halten die Knoten Routinginfos, ein Mischbetrieb ist nicht zulässig.

Sicherheit RPL Optionen: Unsecured, Preinstalled (Vorinstallierte gemeinsame Geheimnisse auf Knoten, gesicherte RPL Nachrichten), Authentifiziert (Beitritt Blattknoten im vorinstallierten Schlüssel, als Router: Schlüssel von Authentication Authority erworben). Schutzziele: Integrität, Vertraulichkeit, Schutz vor Wiedereinspielen, Schutz vor Verzögerung

Konstruktion genau

- Knoten schicken periodisch link lokale Multicast DIO Nachrichten (Stabilität beeinflussen Periodendauer)
- Knoten hören auf DIO Nachrichten und verwenden Infos um DODAG beizutreten, verwalten
- Elternwahl aufgrund Infos in DIO Nachrichten. Elternwahl= wähle 2 Knoten, minimiere Pfadkosten. Knoten dürfen Noderank nur verkleinern
- Ergebnis: Upward Routing in Richtung DODAG Wurzel
- Wurzel selbst muss nicht bekannt sein (Moderank reicht aus, abstand über lokale dio bekannt)
- Objective Function bestimmt aus Metrik Moderank, aufgrund dessen Eltern gewählt werden -> automatisch richtiges routen wegen strenger monotonie

Konstruktion distanzvektorbasiert Es werden Pfadkosten für günstigsten Pfad zur Wurzel bekanntgegeben. Elternknoten werden solche, die Pfadkosten minimieren (Elternmenge und bevorzugter Elternknoten gewählt). Vorsicht bei Schleifen, Count-to-Infinity. Kann günstig sein hohen NodeRank zu haben: Größere Menge potentieller Elternknoten (Bessere Robustheit). Weniger Knoten können diesen Knoten als Elternknoten wählen (Schleifenvermeidung) -> Energieersparnis, weniger Weiterleitungen. Problem Greediness: count to infinity. Die Schleifenbildung ist unvermeidbar (Linkkosten können sich ständig ändern). Die maxdepth rule schränkt die Elternwahl ein. Erlaubt sind dabei alle Knoten, deren NodeRank kleiner ist als der eigene oder den eigenen

noderank nicht mehr als einen bestimmten wert überschreiten noderank(parent) < noderank(self) + DAGMaxRankIncrease (auch 0). maxdepth rule schränkt Schleifenbildung nicht ein, schränkt aber deren größe ein.

Schleifenerkennung bei Datenübertragung Immer noch möglich sind Schleifen im Datenpfad, bsw. wenn Links vorübergehend ausfallen. Dann gibt es keine unmittelbare Reaktion durch das Routingprotokoll (zu teuer). Ansatz ist Piggybacking von Routinginfos in Dateneinheiten. Setzen von Feldern im Kopf der Dateneinheit (ipv6 extension header), auch datapath validation genannt.

datapath validation/DODAG Reparatur benötigte infos: noderank absender, richtung dateneinheit. verirrte Dateneinheit -> reparatur DODAG. Lokale Strategien bei der Reparatur: Wahl eines anderen Elternknoten, Poison-and-wait (falls kein neuer Elter gewählt werden kann). Globale Strategie neue Version DODAG erzeugen: Durch die Wurzel initialisiert, Neuberechnung DODAG (Ohne EInschränkungen bei Bekanntgabe von Noderank, Elternwahl; aufgrund aktueller Link Metriken, Constraints) -> kompletter Neustart Topologie, Optimierung (Objective Function). Dies deutlich teuerer als lokale Reparatur, aber bessere Pfade

Diskussion

- MP2P meister Anteil Verkehr (RPL findet redundante, optimierte Pfade)
- P2MP selten (Kommunikation Aktoren nicht beachtet, aktives auslesen sekundär)
- P2P so gut wie niergends (Pfade nicht optimal, im non-storing mode immer über wurzel kommuniziert)
- generierung down Routen mittels DAO erzeugt Signalierungsoverhead
- Fragmentierung auf Vermittlungsschicht
  - RPL für ipv6 entworfen
- Aggregation von Adressen (storing mode rpl router Routingtabellen Einträge für gesamten sub DODAG speichern); ohne Aggregation können Routingtabellen sehr groß werden. Baumtopologie (Eltern nur noch redundant gewählt, wenn gemeinsames Präfix)

5 Topologiekontrolle

Topologie ist der Aufbau von Verbindungen zwischen Systemen. Ein dichtes Netz führt zu komplexer Topologie (Maß der Dichte = Anzahl der Nachbarn, abhängig von betrachter schicht). Idee ist Komplexität zu reduzieren durch Topologiekontrolle, darf nicht restrikitv sein: Erhaltung der Konnektivität. Topologiekotrolle:

- Flaches Netz (gleichberechtigte Systeme): Sendeleistung regulieren oder Zahl der Nachbarn regulieren
- Hierarchisches Netz (Systeme untersch. Aufgaben/Fähigkeiten): Backbone nutzen oder Cluster bilden

5.1 Flache Netze

Sendeleistung regulieren: Nicht immer maximale Sendeleistung (bezogen ganzes Netz, einzelne Verbindungen), weniger Nachbarn und Beeinflussungen (einfachere Topologie, einfaches Routing), weniger Energieverbrauch Sendevorgang. Alternativ auf einzelne Verbindungen verzichten. Sendeleistung und Zahl der direkten Nachbarn hängen unmittelbar zusammen. Problem ist die richtige Sendeleistung zu finden, abhängig von genannten Vergleichskriterien (erwünscht geringe Sendeleistung, Konnektivität und hoher Durchsatz)-> Kompromiss.

minimale/maximale Sendeleistung MMS Ziel ist die Sendeleistung der einzelnen Systeme zu minimieren. Gegeben ist die Position aller Systeme. Zentralisierter Greedy Algo:

- jedes System i, setze Sendeleistung pi = 0
- zwei teilnetze t1, t2 mit geringem abstand verbinden
- Sendeleistung pi, pj der 2 Systeme (i t1, j t2), die am dichtesten zusammen liegen so erhöhen, dass diese miteinander kommunizieren können
- Verbindungen, die Konnektivität nicht beeinflussen entfernen
- Erhält Konnektivität des Netzes
- Zentralisierter Ansatz in Sensornetzen nicht so praktikabel

5.2 hierarchisches Netz

Beim Backbone Netz werden Systeme in 2 Klassen eingeteilt. Einige Systeme bilden Backbone, Systeme außerhalb Backbone dürfen nicht direkt miteinander kommunizieren. Backbone bildet dominating set D: Jedes System entwender Teil von D oder besitzt mindestens ein System aus D als direkten Nachbarn. Backbone muss verbunden sein. Bei Cluster Netzen werden Sensoren in Gruppen (Cluster) Cx eingeteilt. Jedes System in genau einer Gruppe, Ausnahme sind Systeme die 2 oder mehr Gruppen verbinden. Clusterbildung:

- Vertreter der Gruppe (Cluster Head)
- Cluster heads bilden unabhängige Menge: 2 Cluster Heads dürfen nicht benachbart sein, gesucht ist maximal unabhängige Menge (NP-vollständiges Problem)

Maximal unhabhängige Menge Gesucht ist Attribut um Knoten zu ordnen (Knoten-ID, Energiereserven, Bewegungsgeschw.), Knoten tauschen Attribut lokal aus. Derjenige wird Clusterhead der größten Attributwert aufweißt. Cluster head benachbarte Knoten werden von der Betrachtung ausgeschlossen.

Bestimmung der Gateways Cluster Heads (C) gefunden, Ziel Cluster verbinden und Gateways finden. Jeder Cluster Head benötigt Verbindung zu allen anderen Cluster Heads die maximal 3 Schritte entfernt sind (Backbone Konnektivität sichergestellt). Lässt sich auf Steinerbaum Problem zurückführen (NP vollständig, vereinfachter Fall da C unabhängige Menge). Generelles Problem ist, dass Cluster Heads mehr belastet werden, fallen schneller aus.

LEACH Idee ist durch Rotierung der Cluster Heads die Verteilung der Last. Regelmäßige Neuverteilung der Aufgaben, führt zu gleichmäßigerem Energiebedarf der Knoten. LEACH Low Energy Clustering Hierarchy: p % der Knoten sind Clusterheaders, Aufteilung regelmäßig geändert.

- Verteilter Aufbau einer Cluster Topologie
- gleichmäßige Verteilung des Energieverbrauchs
- Möglichkeit Daten am Cluster Head zu aggregieren
- Annahme: Clusterheads kommunizieren direkt mit Basisstation

Phasen:

- Knoten bestimmen sich mit bestimmter Wahrscheinlichkeit zu Clusterhead und teilen dies Nachbarn mit
- Alle anderen Knoten orden sich dem nächsten in Reichweite befindlichen Clusterhead zu

Die Operationen von LEACH in Runden unterteilt:

- Advertisement Phase: Wahl der Cluster Heads
- Cluster Setup Phase: Aufbau der Cluster
- Steady state phase: Normaler Betrieb, Datenübertragung

In der Advertisement Phase entscheidet jeder Knoten ob er Clusterhead für aktuelle Runde r sein wird. P ist Anteil der Clusterheads im Netz (vorher festgelegt), G Menge der Knoten der Knoten die in den letzten 1/p Runden nicht in Cluster head waren, t(n) = jeder knoten n bestimmt Zufallszahl z zwischen 0 und 1 und wird Clusterhead falls z $<$ t(n).

- n $\in$ G: $\frac{P}{1-P(rmod\frac{1}{P})}$
- n $\notin$ G: 0
- jeder Knoten wird einmal Clusterhead innerhalb 1/P Runden
- Clusterheads broadcasten Entscheidung in Form Advertisement Nachricht an Nachbarknoten

Cluster Setup und Steady state Phase Bei der Cluster Setup Phase entscheiden sich die Knoten aufgrund der erhaltenen Advertisements welchen Cluster sie sich zuordnen. Entscheidung Cluster Head mitgeteilt, dieser sammelt eingehende Meldungen und erstellt einen TDMA Zeitplan den er allen Knoten im Cluster mitteilt. In der Steady State Phase senden und empfangen Knoten gemäß des Zeitplans und schalten sonst Funkchip ab (Energie). Der Clusterhead kann Funkchip nicht abschalten. Nach festem Zeitintervall d nächste Runde r + 1.

6 Datentransport

Kommunikationsformen:

- Sensoren -> Aktoren: Unicast (Zuverlässigkeit), hausautomation , industrie, autowäsche
- Senke -> Sensoren/Aktoren: Multicast/Broadcast (Kontroll/steuerinfo, Codeupdate), zuverlässigkeit
- Sensoren -> Senke: Concast (Sensormessdaten), Zuverlässigkeit, stau

Zuverlässiger Transportdienst:

- Alle Dateneinheiten kommen an
- unverändert
- reihenfolgetreu
- duplikatfrei
- keine Phantom Dateneinheiten

Protokollmechanismen: Zeitgeber, Quittungen, Sequenznummern, Sendewiederholungen (Automatic Repeat Request), Vorwährtsfehlerkorrektur (Forward Error Correction). TCP:

- TCP erkennt Ursache Paketverlust nicht (nimmt Stausituation an) -> schlechte Performance
- 100 % Zuverlässigkeit (viele Wiederholungen, head of line blocking (Latenzen), daten bei Transport nicht verändert (kein in network processing))
- verbindungsorientiert (3 wege handshake (latenzen))
- nur für Unicast

Probalistische Zuverlässigkeit: Nur ein Teil aller Dateneinheiten erreicht Senke (mt Wahrscheinlichkeit p) -> Verbraucht weniger Energie, wegen schlechter Funkverbindung.... Teil Dateneinheiten ausreichend bei redundanter Info (periodisch). Infos die Senke erreicht stimmt nicht exakt mit erfassten Info überein (durch Paketverlust, Aggregation) oder Info ist veraltet (lange Verzögerung wegen schlechter Funkverbindung).

6.1 Unicast

6.1.1 Zuverlässigkeit

Pro Hop vs. Ende-zu-Ende:

- Bei Quittungen pro Hop (Link Layer): Jeweiliges Nachbarsystem (Vorgänger) wiederholt Dateneinheit

  - Vorteil: Wiederholungen auf MAC Schicht möglich -> geringe Latenz
  - Nachteil: Zustandshaltung auf Zwischensystemen nötig
  - Viele Quittungen zwischen Nachbarn nötig, teuer
- Bei Ende-zu-Ende Quittungen: Nur ursprünglicher Sender wiederholt Dateneinheit
  - Vorteil: Kontrolle auf Transportschicht immer nötig bei zuverlässigem Dienst
  - Nachteil: Timer bestimmen
  - Dateneinheit über gesamten Pfad wiederholen

Nur positive Quittungen (ACK) sinnvoll, negative (NACK) schwierig (Empfänger weiß nichts von Dateneinheit, bei periodischem Senden sinnvoll). Stop-and-wait: Senke sendet ACK bei Empfang, Quelle sendet nach Emfang von ACK nächste Dateneinheit (Quelle wiederholt eigene Datenheit, wiederholt unendlich oft). Fallunterscheidungen stop-and-wait: Nur Ende-zu-Ende ACKs/Wiederholungen. Ende-zu-Ende und ACK/Wiederholungen pro Hop (MAC[x] jeweils bis zu x Sendewiederholungen pro Hop). Ende-zu-Ende Quittungen immer nötig, aber pro Hop Sendewiederholungen sinnvoll (hohen Bitfehlerraten). Je größer x bei MAC[x], desto später beginnt exponentieller Anstieg im Energieverbrauch (Verfahren MAC[unendlich] linearen Energieverbrauch bei steigender Anzahl Hops). Vorteile noch von pro-Hop: Energieverbrauch wird im Netz verteilt, Sender nicht in jede Sendewiederholung involviert.

Energieverbrauch fehlerhafte drahtlose Kommunikation wird durch Binary symmetric channel modelliert (jedes bit mit gleicher unabhängiger Wahrscheinlichkeit p falsch übertragen -> Erwartungswert). Bei geringen Bitfehlern lohnt der Verzicht auf pro Hop Sendewiederholungen.

6.1.2 HHR

Hop-by-Hop Reliability: Zielsetzung ist erhöhte Zuverlässigkeit ohne Quittungen. Erhöhe die Wahrscheinlichkeit für erfolgreichen Empfang ohne Quittungen und Sendewiederholungen, sende Dateneinheit sofort k mal. Wähle k so, dass Dateneinheit mit Wahrscheinlichkeit r von Senke empfangen wird ri $= 1 - pi^{ki}$. Berechnung von ki = mit Kenntnis der Paketfehlerrate pi eines Systems i zum Nachbarsystem.

mit Acknowledgements Sende bis zu ki mal, warte aber jedes mal auf Quittung. Höre auf zu senden, sobald erste Quittung empfangen oder ki erreicht. Vergleich der beiden: Pro Hop Quittungen lohnen sich erst bei hohen Paketfehlerraten.

- HHR
  - Vorteil: Übertragungseinsparung durch Quittungsverzicht
  - niedrigen Paketfehlerraten Daten weniger oft übertragen.
- HHRA

- Vorteil: hohen Paketfehlerraten auf weitere Sendevorgänge nach erfolgreicher Quittung verzichtet
- Nachteil: Overhead der Quittungen lohnt sich bei geringen Paketfehlerraten nicht

6.2 Multicast

6.2.1 PSFQ

Pump Slowly Fetch Quickly: Ziel Daten von Basisstation zu einer Menge Sensoren transportieren. Basisstation pumpt Dateneinheiten ins Sensornetz, zwischen versand von 2 Dateneinheiten gewartet. Systeme speichern Dateneinheiten, falls es eine neue ist. Weiterleitung nur falls es Sequenznummer konsistent (keine Dateneinheit fehlt). Falls Sequenznummern fehlen: Nach verlorener Dateneinheit anfragen, Fetch Operation -> lokaler Repartaturmechanismus (NACK mit fehlender Sequenznummer versenden). Vorgängersystem versendet Dateneinheit neu (empfangene Dateneinheit immer puffern). Pumpen langsam (fetchen ermöglichen), fetch schnell.

Details Verzögerung zwischen 2 Pump Operationen groß, der Vorteil ist, dass mehrere fetches dazwischen durchführbar sind. Probalistisches Weiterleiten von Dateneinheiten mit aufeinanderfolgenden Sequenznummern. Warte zufälliges Zeitintervall t, höre währenddessen Medium ab, leite nur dann weiter, falls <= 3 Nachbarn die gleiche Dateneinheit weitergeleitet haben. Dateneinheiten mit falscher Sequenznummer nicht weiterleiten, fehlende holen. Fetch Anfragen (NACK) enthalten Liste fehlender Sequenznummern, sind Broadcast. Idee: Für jede fehlende Dateneinheit im NACK.

- Systeme die fehlende Dateneinheit kennen, starten Timer t und warten, abhören Medium
- falls fehlende Dateneinheit vor Ablauf Timer gehört -> Timer abbrechen
- nicht gehört, Dateneinheit senden
- nächste Dateneinheit

Solange fetchen bis alle fehlenden Dateneinheiten empfangen, warte dazwischen jeweils Tr (oder obere Grenze k von NACK überschritten).

Evaluierung Je größer Distanz zur Quelle, desto niedriger Zustellrate. Je höher Paketfehlerrate, desto niedriger Zustellrate (Mehr Dateneinheiten/NACKs gehen verloren). PSFQ erreicht selbst bei p = 70 % noch Zustellraten > 70 % aber keine 100 % Zustellgarantie.

6.3 Concast

Datenströme können zu Stau/Überlast führen. Einige Sensoren beobachten Ereignis und senden zur Senke (hohe Systemdichte/Engpass), viele verschiedene Sensoren senden periodisch an Senke (zu kurzes Sendeintervall, Stau bei Senke). Auswirkungen Stau:

- Datenverluste -> Dateneinheit neu übertragen, kostet zusätzlich Energie
- Daten kommen verzögert an -> weniger Daten senden, abhängig von Anforderung der Anwendung

6.3.1 ESRT

Event-to-Sink Reliable Transport: In jeder Periode p mit Länge t sollen ca R Dateneinheiten zugestellt werden. Anpassen der Senderate fi zum Zeitpunkt i. Senke misst in Periode i Anzahl ri der empfangenen Dateneinheiten. Senke berechnet Zuverlässigkeit ni = ri/R. Senke berechnet fi+1 basierend auf fi und ni, fi+1 an alle Systeme per Broadcast gesendet. Senderate fi+1 wird bestimmt in Abhängigkeit der Region in der sich fi befindet (kein Stau/niedrige Zuverlässigkeit, kein Stau/optimale Zuverlässigkeit, kein Stau/ hohe Zuverlässigkeit, Stau/hohe Zuverlässigkeit, Stau/niedrige Zuverlässigkeit)

Fallunterscheidung

- Kein Stau/niedrige Zuverlässigkeit: ni $<$ 1 (weniger Dateneinheiten als gewünscht ankommen), fi $<$ fmax (fmax Rate ab der ni kleiner wird); fi+1 = fi/ni
- Optimales Verhalten: 1 - ϵ <= ni <= 1 + ϵ, fi $<$ fmax; fi+ 1 = fi
- kein Stau/Hohe Zuverlässigkeit: ni $>$ 1 (mehr Dateneinheiten als gewünscht ankommen), fi <= fmax; fi +1 = fi/2(1 + 1/ni)
- Stau/hohe Zuverlässigkeit: ni $>$ 1, fi $>$ fmax; fi+1 = fi/ni
- Stau/niedrige Zuverlässigkeit: ni <= 1, fi $>$ fmax; fi+1= fi^{ni}/k(k Anzahl Perioden in der Netz in diesem Zustand verweilt)
- konvergiert zum optimalen Fall, schnell nur wenige Perioden notwendig

6.3.2 Aggregation

Aggregationsformen:

- Keine Aggregation: Eingehende Daten direkt weitergeleitet, keine Vorverarbeitung
- Paketaggregation: Eingehende Daten zwischengespeichert, gesammelte Daten in einem großen Paket gesammelt weitergeleitet
- Datenaggregation: Eingehende Daten zwischengespeichert, gesammelte Daten werden mit Aggregationsfunktion zu einem Aggregat zusammengefasst, welches weitergeleitet wird

MRS Anzahl Messwerte die Datensenke erreichen. Ergebnisse

- TinyOS LPL: Aggregation minimal (LPL synchronisiert Daten lokal, Folgeübertragung günstiger (Präambel kürzer), kein Zusammenhang zwischen Energiebedarf und Anzahl Datenpakete
- S-MAC: Aggregation erhöht MRS (Durchsatz bei kleinen Duty-Cycle gering), Aggregation reduzuert Anzahl an Übertragungen
- Ergebnis stark abhängig von MAC Protokoll, Parameter Protokoll wichtiger
- Aggregation hilft Stausituationen aufzulösen
- Daten/Paketaggregation kein großer Unterschied: Paketagg. vorzuziehen, da alle Quelldaten vorliegen und nicht nur Aggregat
- Datenvolumen hier kein großer Einfluss Energiebedarf (Maximale Paketgröße Sensornetze)
- Anzahl Übertragungen oft wichtig

7 Systeme

Verbindung mit dem Internet -> Internet of Things (6LowPAN zur ipv6 Konnektivität, CoAP zur Unterstützung Client/Server Anwendungen, RPL (Routing im LowPAN). Anwendungsszenarien (Standardisierung im Kontext mit ZigBee). Industrial Internet mit hoher Verfügbarkeit, deterministisches Verhalten (WPAN (802.15.4e, 6TiSCH (Zusammenarbeit 6LowPAN)), LANs (Time-Sensitive Networking, DetNet))

7.1 IETF

IPv6 over Low-Power Wireless Personal Area Networks: Nutzung von IPv6 in Umgebung mit stark eingeschränkten Ressourcen. Potentiell sehr viele Sensorknoten im Einsatz (ipv4 Adressraum reicht nicht aus, daher ipv6). Knoten haben global eindeutige IPv6 Adresse, damit sind alle Knoten global erreichbar, es ist kein Gateway erforderlich, der Ende-zu-Ende Kontext bleibt erhalten. Nutzung von IPv6 Autokonfigurationsmechanismen, dies erspart Konfigurationsoverhead. Nutzung von etablierten Managementprotokollen und kein Anpassen von bestehenden Anwendungen erforderlich (UDP als Transportprotokoll).

Architektur Verbindung zwischen Inseln mit drahtlosen eingebetteten Systemen herstellen. Jede Insel ist Stub-Netz im Internet, enthält Quelle und Senken von Daten und fungiert nicht als Transitnetz für andere Quellen und Senken. LoWPAN = Insel mit Geräten mit gemeinsamer IP-Adresspräfix. 2 Typen von Geräten in einem LoPAN unterschieden: Router, führen auch Routingaufgaben aus und Host, Quelle oder Senke von Daten, sind nicht in Routing involviert.

- Simple LoWPAN: Ein Edge Router, ein Link zu anderen IP-Netzen
- Ad-hoc LowPAN: Kein Edge Router, keine Infrastruktur
- Extended LoWPAN: Mehrere Edge Router, gemeinsamer Backbone Link

Klassische Schichtenmodell im Internet:

- HTTP
- TCP/UDP
- IP
- Ethernet

Schichtenmodell von 6LoWPAN:

- CoAP
- UDP
- RPL

- IPv6 mit 6LoWPAN Adaption
- 802.15.4

Anwendung dazwischen: Interoperabel und transparent für Anwendungen

7.1.1 6LoWPAN Adaption

Annahmen:

- senden kleine Dateneinheiten
- basiert auf 802.15.4, auch NFC, ipv6 over bluetooth low energy
- Geräte leistungsfähigkeit limitiert
- batteriebetrieben
- Ausbringen der Netze ad-hoc (verschiedene Topologien, Selbstorganisation)
- Konfigurations- und Managementbedarf minimal sein

Aufgaben: IP Konnektivität herstellen (Interkonnektivität mit anderen IP-Netzen herstellen). Geringe Paketgröße -> Header Compression (IPv6 Header Compression, UDP Header Compression), da Dateneinheit möglichst in eine 802.15.4 Dateneinheit passen soll, ansonsten Fragmentierung und Reassemblierung. Geringer Konfigurationsaufwand -> 6LoWPAN-ND, Protokolle einfach zu konfigurieren, Bootstrapping neuer Netze möglichst einfach, Selbstheilung. Abbildung von Ipv6 Adressen auf IEEE 802.15.4 Adressen. LoWPAN wird auf einen IPv6 Link abgebildet, Unterstützung von Link-Layer Broadcast. Abbildung von IPv6 Multicast Adressen auf Link-Layer Broadcast Adresse (In jedem MAC Frame PAN-ID und Link-Layer Broadcast Adresse (0xffff) enthalten). Einordnen der 6LoWPAN-Adaptionsschicht: Adaptionsschicht zur Nutzung von IPv6 über 802.15.4 (auf Endgerät Network Layer, EdgeRouter Ethernet MAC). IPv6 Adressen:

- Adresstyp wird durch führende Bits einer Adresse festgelegt
- Unicast: einzelnes Interface. alles übrige (0:0:0:0:0:ffff::96)
- Anycast: Menge Interfaces, Dateneinheit an ein Interface aus Menge ausgeliefert. Local Link Unicast (Interface ID) 1111111010 (als Präfix: fe80::/10)
- Multicast: Menge Interfaces, Dateneinheiten an alle Interfaces der Menge ausgeliefert. 11111111 (als Präfix: ff00::/8)

Adressen von IEEE 802.15.4

- IEEE 64-bit Adressen: MAC Adresse des drahtlos angebundenen Geräts
- 16-bit Adressen, eindeutig im PAN: PAN Koordinator verteilt diese, vergleiche 802.15.4 im non-beacon modus

- Berechnung der IPv6 Adressen im 6LoWPAN: IPv6 Adressen mit 64 Bit Präfix und 64 Bit Interface.
- 6LoWPAN nutzt Stateless Address Autokonfiguration: Berechnung der Adresse aus LoWPAN und Interface Identifier des drahtlosen Geräts

Fragmentierung in der Adaptionsschicht nötig (802.15.4 Dateneinheit nur 127 Byte, 102 Nutzdaten), aber 802.15.4 bietet keine Fragmentierung an. Header Compression ist wünschenswert (teilweise Informationen redundant in Paketköpfen, effizientere Codierung). IPv6 Dateneinheit: IPv6-Paket = Standardkopf, Paketkopferweiterung 1 -n(optional), Daten. IPv6-Standardkopf= Source Address, Destination Address. 6LoWPAN spezifiziert eigene Paketköpfe für LoWPAN spezifische Metadaten und die Verkapselung der IPv6 Dateneinheit: A B | 6LoWPAN AB-type Header | Payload; Werte für AB bestimmen den Paketkopf Typ:

- 00 kein LoWPAN Paketkopf
- 01 Dispatch Header (Komprimierung)
- 10 Mesh Paketkopf, für 2-Schicht Routing
- 11 Fragmentierungsheader
- 6LoWPAN Header können verkettet werden
- Mesh Header + Fragment Header + Header Compression (HC1)

Fragmentierungskopf Datagram tag zur Unterscheidung der Dateneinheit bei Reassemblierung (in Kombi mit 802.15.4 Sender/Empfänger Adresse und der Datagram size). Datagram size ist Position in reassimbilierter Dateneinheit. Datagram offset zur Berechnung der restlichen Daten, kann nur Positionen die Vielfache von 8 Bytes sind beschreiben.

Header Compression Unnütze redundante Infos vermeiden, Felder mit allgemeinen Infos weglassen, Hop-by-Hop Header Compression.

- LOWPAN HC1: Komprimierung des IPv6-Kopfs
- LOWPAN HC2: Komprimierung UDP Kopfs
- Zustandslose Komprimierung: Sender und Empfänger benötigen keinen gemeinsamen Kontext
- erweiterungen um effizientere, zustandsbehaftete Komprimierungsmechanismen exisiteren

Dispatch-Header gibt nachfolgenden Paketkopftyp an: Kodierungsmöglichkeit für Protokollerweiterungen. Unterscheidung der Paketköpfe anhand Dispatch Byte: 2 Bit Dispatch Header Präfix, 6 Bit Sektor. Im Anschluss folgt direkt durch den Selektor beschriebende Paketkopf.

Komprimierung IPv6 Header IPv6 Ursprungs und Zieladresse sind link-lokale Unicast Adressen (Interface ID aus Schicht 2 abgeleitet werden). Payload length (aus Schicht 2 Payload length oder aus Datagram size im Fragmentierungskopf berechnet werden), Traffic Class und Flow Label oft 0. Next Header UDP, ICMP oder TCP. Kann auf 2 Byte Kopf komrimiert werden (+ 1 Byte Hop Limit Feld).

komprimierter UDP-Kopf

- Port Nummer: evtl. weniger Ports ausreichend
- UDP Payload Length: uU aus schicht 3 Payload length berechnet werden
- kann nicht komprimiert werden: Prüfsumme
- IPv6 Kopf und UDP Kopf können auf 6 Byte komprimiert werden.

Was passiert mit Multicast Adressen, was mit globalen Unicast Adressen? 2 weitere Komprimierungsmethoden:

- LOWPAN IPHC: Komprimierung von globalen Unicast Adressen unter Ausnutzung von Kontextinfo (wird nicht beschrieben wie Kontextinfo ausgetauscht, zb während Neighbor discovery). Komprimierung von Multicast Adressen
- LOWPAN NHC: Komprimierung der Next Header in IPv6 (sowohl UDP als auch TCP)

7.1.2 CoAP

Constrained Application Protocol: Transfer Protokoll für eingebettete/ressourcenbeschränkte Maschinen-zu-Maschinen Kommunikation. Leichtgewichtete Alternative zu HTTP. Es wird UDP genutzt (einfache Message Schicht für Sendewiederholungen), ist ein kompaktes Format. RESTful (Representational State Transfer): GET, PUT, POST, DELETE, Unterstützung von URIs. Interworking mit HTTP (einfache Proxy und Caching Umsetzung), Unterstützung von DTLS (Datagram Transport Layer Security).

CoAP Schichten

- Message Schicht: Mechanismen zur zuverlässigen Nachrichtenübertragung; 4 Typen von Nachrichten: Confirmable, non-confirmable, Acknowledgement, Reset. Übertragung von Request/Responses
- Requests/Responses: enthalten jeweils einen Methodenaufruf auf ein Objekt; 4 Methoden: Get(Aufruf Objekt), put(Speichern Objekt), post (übermitteln neuer Infos), delete. Ein Objekt durch Objektidentifikator gekennzeichnet (URIs)

Messaging Modell Eindeutige Nachrichten ID (Duplikaterkennung, Zuordnung von Quittungen), Unzuverlässige Übertragung (Nachrichtentyp non confirmable). Zuverlässige Übertragung (Quittungen und exponentieller Backoff, Nachrichtentyp Typ Confirmable, Quittungen Typ Acknowledgement). Signalisierung von Fehlern (Nachrichtentyp Reset (RST).

Request/Response Modell CoAP basiert wie HTTP auf Client/Server Modell, allerdings agieren beide Kommunikationspartner als Client Server. CoAP Request vergleichbar mit HTTP Request. Client sendet Request an Server um Aktion auf einer Ressource auszuführen. CoAP Request besteht aus:

- auszuführender Methode auf Ressource
- Identifikator auf Ressource
- Nutzdaten
- Optional: Meta Daten zum Request
- Client erzeugt zur eindeutigen Identifikation des Requests einen Token

Server antwortet mit Response und dazugehörigem Response code. CoAP Response Code vergleichbar mit HTTP Status Code. 3 Klassen:

- 2 - Success: Request erfolgreich empfangen, interpretiert, ausgeführt
- 4 - Client Error: Request enthält falsche Syntax oder nicht ausführbar
- 5 - Server Error: Servor konnte Request nicht ausführen
- Responses können sowohl Piggy-backed (im ACK) als auch in separaten Nachrichten übertragen werden

CoAP nutzt einfaches und kompaktes Nachrichtenformat.

- Versionsnummer
- Nachrichtentyp (CON, NON, ACK, RST)
- Token Länge
- Code: Identifiziert Methodenaufruf und Fehlercode
- Message ID
- Token: genutzt um zusammengehörige request/respones zu identifizieren
- Options
- Payload: Nach Payload Marker (0xff) folgt eigentlicher Inhalt

CoAP URIs nutzt coap und coaps URI Schema um Ressourcen und deren Lokation zu identifizieren (coaps = DTLS). Syntax zur Erstellung von CoAP-URIs: Coap-URI = “coap:“ “//“ host [“:“ port] path-abempty [“?“ query] Token stellt Beziehung zwischen Request und Response her. Message ID nur zwischen Nachricht und Quittung.

Sicherheit

- CoAP unterstützt Nutzung von DTLS (Integritätsschutz, Vertraulichkeit und Authentifizierung von Anwendungsdaten)
- 4 Sicherheitsmodi:
  - NoSec: DTLS nicht unterstützt
  - PreSharedKey: DTLS genutzt, vorverteilte Schlüssel
  - RawPublicKey: DTLS genutzt, asymmetrische Schlüsselpaare verwendet
  - Certificate: DTLS genutzt, X.509 Zertifikate genutzt
- auf ressourcenschwachen Geräten kommen spezielle, angepasste Cipher Suites zum Einsatz

7.2 ZigBee

Sehr stark anwendungsorientiert, Standard für Internet der Dinge. Kleine autonome Netze stehen im Mittelpunkt, keine Interoperabilität mit etablierter Infrastruktur. Ziel ist ein Standard zur Realisierung von Anwendungen mit Überwachungs- und Steuerungsaufgaben mit einheitlichen herstellerübergreifenden Schnittstellen. Annahme ist geringes Datenaufkommen mit dem Ziel eines geringem Energieverbrauchs. (Haus/Gebäudeautomation, Beleuchtungsstuerueng, Gesundheitspflege, Einzelhandel und Logistik, Smart Energy).

Architektur - Schichtenmodell

- 802.15.4
- Netzwerkschicht: Routing, Adressierung (Zuweisung und Verwaltung von 16 Bit Kurzadressen, keine Verwendung von IP), Konfiguration der Geräte (Rollenzuweisung (Koordinator, Endgerät, Router)), Netzwerkmanagement
- Anwendungsschicht:
  - Anwendungsunterstützt: zuverlässiger Transportdienst, Binding (Kopplung von Geräten), Gruppenverwaltung
  - Anwendungsrahmenwerk: Anwendungsobjekte (Anwendungsfunktionalität)
  - ZigBee Device Object: Management Dienste, Hypervisor

Anwendungsrahmenwerk Einbettung von Anwendungsobjekten, durch Unterteilung der Anwendung in bis zu 240 Anwendungsobjekte. Je Anwendungsobjekt eine Aufgabe mit eindeutiger ID, entspricht einem Endprodukt. Definition von Anwendungsprofilen: Standards für die Kommunikation zwischen den Geräten in einem Anwendungsbereich. Identifiziert über Profile-Identifier. Standardprofile werden von der ZigBee Alliance entwickelt und enthalten allgemein nützliche Anwendungsbereiche. Private Profile werden von individuellen Herstellern entwickelt und benötigen einen von der ZigBee Alliance zugeordneten Profile-Identifier. Anwendungsprofile bestehen aus:

- Menge von Geräten für Anwendung benötigt
- Menge von Clustern um Funktionalität zu implementieren
- Menge von Attributen, die den Zustand von Geräten repräsentieren
- Menge von Befehlen die Kommunikation ermöglichen
- Beschreibung welche Cluster von welchem Gerät benötigt werden
- funktionale Beschreibung für jedes Gerät

Client/Server Modell Anwendungen basieren auf Client-Server Modell. Attribute sind typischerweise Servern zugeordnet. Clients nutzen Attribute über standardisierte Cluster-Befehle.

Cluster Menge von Attributen und Befehlen. Jeweils einer Rolle (Client/Server) zugeordnet. ZigBee Standards spezifiziert keine Cluster. Cluster werden in der ZigBee Cluster Library zusammengefasst.

Geräte Device Description ist die funktionale Beschreibung eines Gerätetyps. Spezifiziert welche Cluster die Funktionalität bereitstellen.

Endpunkt Entspricht konzeptionell einem Port im Internet Modell. Die Kommunikation erfolgt immer zwischen 2 Endpunkten. Der Endpunkt ist genau einer Device Description zugeordnet. Gerät kann über mehrere Endpunkte verfügen. Maximal 256 Endpunkte möglich: Endpunkt 0 (ZigBee Device Object (Verwaltungsfunktionalität)), Endpunkte 241 - 254 (für zukünftige Anwendungen reserviert), Endpunkt 255 (Broadcast).

zuverlässiger Transportdienst Nutzt 802.15.4 (beschränkte Größe der Dateneinheiten, unzuverlässig, hohe Bitfehlerrate). Datentransport mit ZigBee ist keine eigene Schicht, verbindungslos, Fragmentierung für große Dateneinheiten, Keine TCP mäßige Zuverlässigkeit

- Vorteil: Quittierung (Ende zu Ende) nur bei Fragmentierung verpflichtend
- Vorteil: Duplikaterkennung

- Nachteil: Fehlererkennung (Schicht 2 hat aber Prüfsummen)
- Nachteil: Reihenfolgetreue (außer bei Fragmentierung)
- Nachteil: Staukontrolle
- Nachteil: Flusskontrolle (außer bei Fragmentierung)

Nutzung von Quittungen Quittungen erfolgen pro übertragener Dateneinheit, werden als spezielle Dateneinheit übertragen (kein piggy back). Sendewiederholungen nach Ablauf des Timers (max. 3 Sendewiederholungen), bei Empfang einer unerwarteten Quittung (Endpunkt, Cluster ID oder Sequenznummer stimmen nicht überein).

Fragmentierung Hier sind Quittungen verpflichtend. Max 127 Bytes auf physikalischer Schicht (802.15.4), wenn MAC Datenfeld nicht ausreicht, dann Fragmentierung. Quittungen werden dann nicht pro Fragment sondern pro Fenster verschickt (Fenstergröße 1 -8 Fragmente). Quittung beinhaltet Sequenznummer des ersten Fragments und Bitmaske (Selektives ACK/NACK durch Bitmaske). Quittungen gesendet bei erhalt des letzten Blocks im Sendefenster oder vollständigem Empfang des Sendefensters oder optional Timeout. Nicht quittierte Blöcke müssen erneut gesendet werden. Kein Versand neuer Fragmente vor vollständiger Quittierung des Fensters.

Binding Verknüpfung zwischen den Endpunkten 2 Geräte. Wird in Binding Tabelle gespeichert. Im Netz stellt ein Gerät (Cache Server) komplette Binding-Tabelle bereit. Bei der Kommunikation von Befehlen ist keine Adresse erforderlich (Quelle muss Zielgeräte nicht kennen).

Gruppenverwaltung Multicast Gruppen werden über eine Gruppen ID identifiziert. Gruppentabelle: Zuordnung Gruppen ID -> Lokale Endpunkte. Endpunkte können mehreren Endpunkten zugeordnet werden.

Geräteklassen/Rollen

- Reduced-Function Devices (RFD): Implementieren nur nötigste Teile von IEEE 802.15.4, keine direkte Kommunikation zwischen RFIDs möglich
- Full-Function Devices (FFD): Können auch als 802.15.4-PAN Koordinator agieren, RFD-RFD Kommunikation via FFD (store and forward)
- Koordinator (FFD): Übernimmt Verwaltung des ZigBee Netzwerks, beeinflusst Netztopologie, Funkkanal
- Router (FFD): Kann Dateneinheiten im Netzwerk weiterleiten
- Endgerät (RFD oder FFD): Gerät ohne Weiterleitungsfähigkeit

Routing Unterstützte Topologien: Stern, Baum, Mesh

Baumtopologie Statisches Adressierungsschema, dieses sieht für jeden Router einen eigenen Adressbereich vor(implizite Baumstruktur im Adressraum). Parameter: Max. Tiefe Baum, Max. Anzahl Kinderknoten, Max. Anzahl Kind Routern. Adressen werden dabei direkt zum Routing verwendet, Router kennen Distanz zum Koordinator und Parameter des Adressbaums (Routingentscheidung lokal mit wenig Rechenop. möglich). Falscher Teilbaum: Weiterleitung aufwärts Richtung Koordinator. Richtiger Teilbaum: Weiterleitung abwärts Richtung Ziel.

- Vorteil: Keine Pfadsuche
- Vorteil: Keine Zustandshaltung (Routingtabellen)
- Nachteile: Ausfallanfällig, keine redundanten Pfade
- Nachteile: supoptimale Pfade
- Nachteile: Keine Router mit freien Adressen in Reichweite -> Kein Beitritt zum Netz möglich
- weitere Eig. : 802.15.4 Beacon Mode möglich (duty cycling für alle Knoten) Koordinator gibt Beacon Takt vor, abhängige Router senden ihr Beacon zeitversetzt

Meshtopologie Adressen werden zufällig gewählt, Koordinator stellt sicher dass keine Adresskollisionen auftreten. Routingverfahren erforderlich zur Pfadsuche (AODV ad-hoc on demand distance vector routing). AODV einfaches Protokoll, eine Routinganfrage wird durch das Netz geflutet und etabliert zur Quelle führende Pfade. Das Zielsystem beantwortet die Anfrage und baut so einen bidirektionalen Pfad auf. Endgeräte beteiligen sich nicht an der Pfadsuche (Router antworten stellvertretend).

- Vorteil: Robust/Selbstheilend, bessere Pfade?
- Nachteil: kein IEEE 802.15.4 Beacon Mode (Koordinator und Router müssen permanent aktiv sein -> ungeeignet für batteriebetriebene Geräte), Ressourcenbedarf (Zustandshaltung und Kommunikation)
- weitere Eig.: Für many-to-one (Concast) Routing geeignet. Senke etabliert zu ihr führende Pfade, effizienter als viele einzelne Pfadsuchen.
- Pfadinfos für Source Routing nutzbar: weniger Zustand im Zwischensystem

Multicast Routing Schicht der Anwendungsunterstuutzung: Gruppenverwaltung. Netzwerkschicht: Weiterleitung der Dateneinheiten an eine Gruppe. Sender ist Mitglied der Gruppe, Broadcast mit gegrenzter Reichweite. Die Zieladresse ist Broadcastadresse, Gruppenmitglieder werten Dateneinheit aus, Nichtmitglieder leiten Dateneinheit weiter (Begrenzter Broadcast Radius Zähler wird erniedrigt). Sender ist kein Mitglied der Gruppe, dann Routing zu einem Mitglied der Gruppe (Weganfrage wird durch Router geflutet). Von diesem Mitglied dann Weiterleitung wie oben.

ZigBee Varianten

- ZigBee 2006, 2007 ZigBee IP basiert auf 6LoWPAN

ZigBee IP entstand im Zuge der Entwicklung des ZigBee Smart Energy 2.0 Profils. NIST-Vorgabe: Offener Standard, basierend auf IEEE/IETF. Definiert einen Protokollstapel auf Basis etablierter Standards (6LoWPAN und RPL, HTTP als Anwendungsschichtprotokoll erzwingt TCP als Transportprotokoll, TLS zur Absicherung). ZigBee Smart Energy 2.0 spezifiziert im wesentlichen Datenformate, Schnittstellen und Anwendungsverhalten.

7.3 Industrial Internet

Harte Echtzeit und Verfügbarkeitsanforderungen, deterministisches Verhalten und geringe Latenz. Isolierung von Verkehr unterschiedlicher Kunden. Anwendungsgebiete sind Audio/Video Streaming (Studio EInsatz) und Industrie 4.0 (Prozessleitsysteme, Maschinensteuerung). Ansätze für Internet der Dinge nutzen drahtlose Kommunikation, aber hohe Fehlerrate und geringe Robustheit. Netzwerkprobleme können schlimme Auswirkungen haben, deswegen in Industrie 4.0 keine Möglichkeit.

7.3.1 WPAN

6LoWPAN mit Integration in IP basierte Lösungen und Time-Slotted Channel Hopping (TSCH) (Medienzugriffsverfahren, mit hoher Robustheit und weniger Fehleranfällig) sind aber attraktiv. Entwicklung neuer Medienzugriffsverfahren basierend auf synchronem Zeitmuliplex. Das Medium wird in feste Zeitschlitze eingeteilt (synchrone Schlaf wachzeiten). Nutze sogennante Slotframes zur Synchro, diese bestehen aus einer festgelegten Anzahl von Zeitschlitzen. Channel Hopping ist der Wechsel der Frequenz für jede zu übertragende Dateneinheit. Sendewiederholung erfolgt auf einer anderen Frequenz, wechsel der Frequenz zufallsbasiert. Vorteile sind planbare zeitliche Belegung des Mediums, Bandbreitenzuteilung möglich. Frequenzwechsel verbessert Robusteit -> Grundlage Determinismus. Störungen durch Signalübertragung (destruktive Interferenz, Fading) und andere Sender betreffen normalerweise nicht alle Funkkanäle.

802.15.4e Variante für den Einsatz in Industrie 4.0. Neuer Betriebsmodi: Time slotted channel hopping, der ein sehr energieeffizienter, planbarer robuster Medienzugriff ist. Vermeidung Idle-Listening.

TSCH Slotframe Struktur Synchro der Geräte auf Basis Slotframe Struktur. Slotframe ist eine Menge von Slots, die sich wiederholen (S-MAC). Geräte folgen Zeitplan der vorgibt, was in jedem Zeitschlitz erfolgt. In jedem Zeitschlitz kann Gerät Senden, Empfangen oder Schlafen. Zeitplan gibt auch vor, welcher Nachbar wach ist und ob empfangen oder gesendet werden kann. Der zu verwendete Kanal (Frequenz) wird vorgegeben. Zeitschlitze lang genug um Datenübertragung und ACK und evtl. Sendewiederholung durchzuführen (impl.abhängige Länge (10ms)). TSCH betrachtet Kommunikation Schicht 2, 6TiSCH im Kontext von 6LoWPAN, Umsetzung von Scheduling.

6TiSCH 802.15.4e beschreibt wie Schicht2 einem Zeitplan folgt, nicht aber wie Zeitplan erstellt und im Betrieb angepasst wird. 6TiSCH: IPv6 over the TSCH mode of IEEE 802.15.4e. Anpassung der 6LoWPAN Architektur an Annahmen 802.15.4e (Synchro, Signalisierung), im draft Stadium. Aufgaben aus verschiedenen Teilbereichen:

- Nachbarn erkennen
- Zeitpläne erstellen und anpassen (statische Zeitpläne)
- Multi hop pfade von rpl auf Zeitpläne abbilden
- auf Topologieänderungen reagieren
- 6TiSCH Netz: kleine drahtlose Mesh Netze, verbunden über schnelles Backbone Netz
- Scheduling: Nicht von 6TiSCH standardisiert, Scheduling Instanz zentral/dezentral
- hohe Anforderungen an Synchronisation

Synchronisation Problem ist, dass Knoten sehr genau synchronisiert werden müssen (auf etwa 1 ms genau). Wiederkehrende Synchro notwendig, lokale Uhren der Sensorknoten laufen aufgrund von Quarzungenauigkeiten auseinander. Ablauf Einführung von Time Mastern. Wahl nicht vorgegeben, verschiedene Verfahren möglich. Synchronisieren Uhren nie mit anderen Knoten im Netz. 2 Möglichkeiten der Synchro für restliche Knoten:

- Paket basierte Synchronisation: Messen der Ankunftszeit der Pakete, verlängern /verkürzen der eigenen Aktivitätsphase wenn Pakete später/früher als erwartet empfangen.
- ACK-basierte Synchro: Messen der Ankunftszeit der Pakete, Unterschied der erwarteten Ankunftszeit und der gemessenen in ACK eintragen. Nach empfang des ACK kann Aktivitätsphase angepasst werden.
- keep alive Nahrichten (alle 30s) falls keine Kommunikation stattgefunden hat.

7.3.2 LAN

IEEE Time-Sensitive Networking anderer Ansatz: Echtzeit und Verfügbarkeitsgarantien mit IEEE 802.x Technologie, Schwerpunkt Ethernet, Ansätze auch auf WLAN übertragbar. Keine Änderung des Medienzugriffsverfahrens, Einhalten von Garantien durch geeignetes Scheduling. Übergeordnete Planung/Koordinierung der Medienzuteilung (Gemeinsam zu 6TiSCH). Konzept: Geringe Latenzen und hohe Zuverlässigkeit und damit keine Paketverluste in Zwischensystemen zulässig, keine großen Puffer in Zwischensystemen möglich. Explizite Reservierung von Ressourcen (Bandbreite). Keine Überlastsituation -> keine Pufferung. Leistungsgarantien nur für akzeptierte Datenströme. Datenströme mit definierter Charakterisitik (Rate, Burstfreiheit), abgeschlossenes Teilnetz,

kooperierender Zwischen und Endsysteme. Nicht TSN Datenverkehr: Best Effort. Die Vermeidung von Überlastsituationen ermöglicht deterministische Latenzen von unter 50ms bzw. 2ms über 7 Hops.

Architektur

- devices outside of TSN cloud still communicate with all other devices using legacy best effort QoS
- half duplex link cant do TSN
- Streaming QoS only guaranteed in TSN cloud
- peer device not TSN capable
- TSN cloud (defended network): TSN switch, TSN endpoint, TSN wireless bridge
- legacy switch, end point, TSN end point
- Ethernet hub, end point, TSN end point

Bausteine Grundlage Weiterleitungsmechanismen auf Schicht 2. Verschiedene Bausteine:

- Zeitsynchronisation
- Stream Reservation Protocol
- Forwarding/ Queueuing: Multipath Routing

Nichtdeterminismus von Medien/zugriffsverfahren wird nicht berücksichtigt. Zeitliche Koordinierung bei drahtlosen (geteilten) Medien erforderlich -> TSCH

8 Sicherheit

Kryptografische Bausteine

- Symmetrische Verschlüsselung (gemeinsamer Schlüssel zum ver/entschlüsseln): hohe effizienz, da meist einfache Operationen (Bit Shift, XOR), Beispiel AES (Advanced Encryption Standrard)
- Asymmentrische Verschlüsselung (öffentliche und private Schlüssel): Basis Faktorisierung von Zahlen schwer, Beispiel RSA
- Integritätssicherung: Kryptografische Hashfunktion (symmetrische Kryptografie); Digitale Signatur (asymmetrische Kryptrografie)
- Schlüsselaustausch
- Vermeide asymmetrische Krypthografie im ioE. Lösung mit symmetrischer Krypto.

8.1 Schlüsselaustausch in Sensornetzen

klassische Schlüsselaustauschprotokolle benutzen häufig zentrale Komponenten (Certificate Authorities, in Public Key Infrastrukturen), aber zentrale Komponenten uU nicht immer erreichbar. Einsatz von Diffie-Hellman wegen Ressourcenbeschränkung schwierig, Berechnungsdauer mehr als 2 Sekunden. Austausch von Schlüsseln im ioE erschwert (Ressourcenknappheit, Selbstorganisiation, Netztopologie). Verfahren zum Schlüsselaustausch:

- Single Mission Key: einfach und problematisch
- Zufallsverteilte Schlüssellisten: 2 Systeme besitzen mit gewisser Wahrscheinlichkeit einen gemeinsamen Schlüssel
- Key Infection: Angreifer darf während Schlüsselaustausch nicht anwesend sein

8.1.1 Naiver Ansatz - Single Mission Key

Alle Systeme bekommen vor ihrer Ausbringung den selben Schlüssel K, dieser wird verwendet um Kommunikation abzusichern (verschlüsseln, Integritätssicherung). Problem ist, sobald ein einziges System korrumpiert wird, ist gesamte Kommunikation im Netz unsicher. Paarweise Schlüssel mit allen Systemen. Jeder bekommt prophylaktisch jeweils einen Schlüssel zur Kommunikation mit allen n- 1 anderen Systemen. Probleme sind, dass der Speicherverbrauch bei größeren n sehr hoch ist. Wie neue Systeme im Netz, wie die Schlüssel auf alle Knoten aufteilen.

8.1.2 Eschenauer Gligor - Zufallsverteilte Schlüssellisten

Benutzer erstellt vorab offline eine große Liste von Schlüsseln, den Key Pool P. Jedes System, das dem Netz beitreten soll, bekommt eine zufällige Teilmenge dieser Schlüssel, seinen Key Ring R. Nun gilt mit einer bestimmten Wahrscheinlichkeit besitzen 2 beliebige Systeme einen gemeinsamen Schlüssel auf ihren beiden Key Ringen. System ohne zusätzliche Infrastruktur, Schlüsselfindung mit hoher Wahrscheinlichkeit, Trade off zwischen Speicherverbrauch und Sicherheit. Falls 2 Systeme keinen gemeinsamen Schlüssel auf ihren Key Ringen besitzen, lässt sich ein Schlüsselpfad zueinander konstruieren. Schlüsselpfad ist ein Pfad über mehrere Systeme hinweg, die jeweils paarweise einen gemeinsamen Schlüssel auf ihren Key Ringen besitzen. Wie erkennen 2 Systeme, dass sie über einen gemeinsamen Schlüssel verfügen? Versenden von Schlüssellisten (System x sendet Schlüsselnummer im Klartext an y, y antwortet mit gemeinsamer schlüsselnummer klartext) -> unsicher. Versenden von Klartext und Chiffrat Paaren. System x sendet Zufallszahl z und Chiffrate seiner Schlüssel an y (z, enc2(z), enc4(z), enc5(z)). System y antwortet mit (enc2(z + 1)), was x überprüfen kann -> teuer. Konstruktion der Schlüsselpfade ist das Problem, dass Systeme uU keinen gemeinsamen Schlüssel haben. System x und y tauschen über z einen gemeinsamen Schlüssel aus -> unsicher. Schlüsselpfade können über mehrere Zwischenstationen gehen -> Schlüsselgraph, muss zusammenhängend sein (mit richtiger Wahl P, R). Sonst Systeme die nie mit anderen sicher kommunizieren können. Vorschlag Konfigurationen für verschiedene P und R.

Sicherheit Angreifer kann gewissen Prozentsatz an Knoten korrumpieren und Schlüsselliste auslesen. Bereits kleine Menge korrumpierter Systeme erlaubt Mithören oder Manipulieren vieler Verbindungen. Häufig wird Schlüssel zur Absicherung mehrerer Verbindungen verwendet. Korrumpierte Systeme auf dem Schlüsselpfad kennen ausgetauschte Schlüssel -> manipulieren, abhören.

8.1.3 Key Infection

Einsatz:

- Smart Dust Environment (viele sehr günstige und zufällig angebrachte Sensorknoten)
- Eingeschränktes Angreifermodell: Angreifer in Schlüsselaustauschphase nur bedingt präsent (lokal eingeschränkt, kann nicht kompletten Netzverkehr abhören und keine aktiven Angriffe während Schlüsselaustauschphase)
- Keine Infrastruktur notwendig, kein Schlüsselaustauschserver

Ablauf des initalien Schlüsselaustauschs wird für und von allen Knoten in gegenseitiger Kommunikationsreichweite durchgeführt. Knoten i wählt einen zufälligen Schlüssel ki, dieser wird per Broadcast versendet. Knoten j wählt einen gemeinsamen Sitzungsschlüssel kji und sendet diesen zusammen mit seiner ID verschlüsselt an Knoten i. Das Key Infection Protokoll besteht aus mehreren Teilen:

- Key Infection: Initalier Schlüsselaustausch im Klartext zwischen benachbarten Knoten
- Multihop Key Exchange(optional): Schlüsselaustausch zwischen nicht benachbarten Knoten
- Secrecy Amplification(optional): Verstärkung der Sicherheit der Schlüssel durch Nutzung von Mehrwege Austausch

Geringer Speicheraufwand Schlüsselmaterial. Geringer Kommunikationsaufwand (Pro Schlüsselaustauch 2 Nachrichten, in der Praxis tauschen alle Knoten gleichzeitg aus (Kollisionen, Sendewiederholungen)). Nur eingeschränktes Angreifermodell, symmetrische Links. Neue Ansätze zunehmend auch asymmetrische Krypto.

8.2 Standardidierung

8.2.1 DICE

DTLS in Constrained Environments - Draft. Ziele sind effiziente Nutzung von DTLS in ioE. Einführung von Multicast-Security auf Basis von DTLS. Eigenschaften:

- Authentifikation zweier Kommunikationsendprodukte
- Integrität und Authentizität der Daten
- Vertraulichkeit
- Implemententierung auf ressourcenbeschränkten ioE Systemen

Erstellen der DICE DTLS Profile: Analyse von DTLS, Entwurf verschiedener effizienter Konfigurationen (Funktionalität in DTLS sehr umfangreich, Unterstützung sehr vieler Kyp. Algos) -> DICE definiert Konfiguration, die effizient und sicher ist.

TLS Grundlagen Protokollstruktur:

- IP
- TCP
- Record Protocol: Schutz der Anwendungsdaten, Vertraulichkeit, Integritätsschutz
- Handshake, CHangeCipherSpec, Alert, Application Data
- Anwendung

Symmetrische Kryptogramme zum Schutz der Vertraulichkeit, Integritätsschutz mittels (H)MAC. Schlüsselmaterial muss durch anderes Protokoll ausgehandelt werden (TLS Handshake Protocol). Record Protocol nutzbar für Protokolle höherer Schichten. Unterschiede TLS und DTLS: TLS setzt zuverlässiges Transportprotokoll voraus, Ion nutzt

häufig UDP (Paketverluste). Mechanismen zur Behandlung von Paketverlusten in DTLS integriert (explizite Sezuenznummer, sendewiederholungen, Timer, Erweiterung des Handshake Protokolls zur Fragmentierung, Reihenfolgetreue und Paketverlusterkennung). Keine direkte Verwendung von Stromchiffren mehr möglich (Reihenfolgetreue nötig).

DICE DTLS Profile Einschränkung der zu unterstützenden kryptographischen Algoritmen. Einschränkung der zu unterstützenden Authentifizierungsmechanismen (preshared keys, raw public...), Einschränkung der Protokollerweiterungen und Funktionen, dafür höhere Effizienz. Sehr viele Cipher Suites in TLS/DTLS beschreiben Kombination aus kryptographischen Mechanismen für Schlüsselaustausch, Authentifikation...viele nicht praktikabel auf ressourcenbeschränkten Systemen. Kryptographische Algorithmen für DICE Profile: Symmetrische Verschlüsselung mit AES, Hassfunktion SHA-1, Asymmetrische Kryptogr. auf Basis elliptischer Kurven(effizienter Implementierter, geringe Schlüssellänge). Authentifikation: Vorverteilte symmetrische Schlüssel (Vorverurteilung in krypt. Hardware oder Firmware). Raw Public keys (Authentifikation nicht Teil des Schlüsselaustauschs). Zertifikate (Nutzung von Diffie-Hellman auf Basis elliptischer Kurven. Zertifikatvalidierung/Prüfung kann ausgelagert werden). Zukünftige Standardisierung: Sichere Multicast Kommunikation (Austausch Gruppenschlüssel schwierig, Authentifikation Absender schwierig). Optimierte Version von IPSec.

8.2.2 DCAF

Delegates Security, bei der Authentifizierung und Autorisierung durch einen Stellvertreter durchgeführt wird. Der Stellvertreter muss vertrauenswürdig sein, entlastet ressourcenbeschränkte Systeme. Aber zusätzlicher Kommunikationsaufwand (weniger Berechnung, mehr Datenaustausch. Delegates CoAP Authentification and Authorization Framework spezifiziert einen Delegationsmechanismus für CoAP, keine eigene Protokollschicht (eingebettet in CoAP). Stellvertreter normale CoAP-Server, Kommunikation mit und zwischen Stellvertreter mit CoAP POST. Nur symmetrische Verschlüsselung auf ressourcenbeschränkten Geräten, sowohl Client als auch Server können beschränkt sein. Individuelle Autorisierung für jede URI und CoAP Methode.

Rollen

- Besitzer
- eingeschränkte Systeme: Client/Server
- vertrauensbeziehungen zwischen Besitzern werden auf Konfiguration entsprechender Zugriffsberechtigungen abgebildet (Authentifizierung mit Zertifikaten/Geheimnis)
- Stellvertreter: Client Authentication Manager (CAM) übernimmt Rolle des Clients im Autorisierungsprozess. Server Authentication Manager (SAM) Vertritt Server.

Architektur Symmetrischer Schlüssel für Client und CAM, Server und SAM. Authentifizierung Client/Server gegenüber dem Stellvertreter. Wechselseitige Authenfizierung von CAM/SAM (Zertifikat oder vergleichbare Mechanismen). Zwischen CAM/SAM TLS und zwischen Server/Client DTLS (CAM/Client DTLS, SAM/Server DTLS) Ablauf:

- Unautorisierte Anfrage Client -> Server [GET bat/status]
- Server verweist Client an SAM [Unauthorized SAM]
- Autorisierungsanfrage an CAM: SAM-URI zur Kontaktaufnahme [POST client-authoize SAM SAI]
- Angeforderte Berechtigungen (SAI Server Authorization Information)
- Weiterleitung der Ticketanfrage an SAM: CAM kann die Anfrage vorher ablehnen oder einschränken [POST SAM SAI, GET]
- Ticketausstellung durch SAM: Ticket Face (F) für den Server, Ticket Verifier (v) für den Client
- Ticket Weiterleitung an Client
- Client nutzt Verifier als summ. Schlüssel, Server prüft Anfrage (stimmen Verifier und Face überein? Anfrage entspricht den im Face enthaltenen Berechtigungen? (DTLS(key=Verifier, pskidentity=Face)

Ticket Face gewährt Zugriffsberechtigungen (SAI), zur Validieren erforderliche Infos (Nonce (N) oder Zeitstempel (TS) schützen gegen Replay-Angriffe (Angreifer recycelt Ticket)). Eine Methode zur Schlüsselableitung (G) oder einen symmetrischen Schlüssel für die Verbindung Client- Server (v) (Ticket face muss auch verschlüsselt werden). Ticket Verifier (für den Client) ist ein symmetrischer Schlüssel, eindeutig dem Ticket face zugeordnet.

Schlüsselübertragung/ableitung Der Server leitet den Verbindungsschlüssel aus dem Ticket Face ab -> Ticket Face muss an den Server übertragen werden. DCAF nutzt dazu den PSK Intentity Hint im TLS Handshake. Übertragung erfolgt im Klartext, da noch keine sichere Verbindung existiert. Schlüsselmaterial muss zusätzlich verschlüsselt werden. Schlüsselableitung:

- Ticket Face umverschlüsselt: Key = HMACserverkey(Face)
- Ticket Face verschlüsselt: Key = DecryptserverKey(Face).v. Verbindungsaufbau schlägt fehl wenn das Ticket Face manipuliert wurde (Server errechnet falschen Schlüssel).

8.2.3 ZigBee

Absicherung auf 2 Schichten möglich. Netzwerkschicht ein geteilter Schlüssel für das gesamte Netzwerk (Single Mission Key), schützt alle Übertragungen insbesondere auch Broadcast. Anwendungsunterstützungsschicht eigene Schlüssel für jede Ende-zu-Ende Verbindung (Linzschlüssel). Nur auf Unicash-Kommunikation anwendbar. Schlüsselverwaltung erfolgt durch ein Trustcenter. Problem ist die Schlüsselverteilung bei initialem Verbindungsaufbau (Übertragung im Klartext). ZigBee Home Automation Profile: Definiert Standardschlüssel, bietet nicht mehr Sicherheit als Übertragung in Sicherheit.

8.3 Reality Check

8.3.1 BMW Connected Drive

Cloud basierter Dienst zur Steuerung des Autos aus der Ferne. Cloud-Service: SMS + HTTP über GPRS/EDGE. Schutzmachanismen sind im Modem implementiert, Mikrocontroller realisiert Basisfunktionalität und überwacht Modem. Verschlüsselung AES und DES, Signaturverfahren. Schlüsselmaterial im Modem fest einprogrammiert, für alle Fahrzeuge gleich. Einfach zu extrahieren.

Angriff Ausgangspunkt ist eine Mobilfunkbasisstation (IMSI Catcher), Auto bucht sich in das gefälschte Netz ein. Angreifer weckt das Steuersignal mit einer SMS, gesichert mit einem nicht mehr geheimen Schlüssel. Auto fordert Befehle aus der Cloud an (HTTP), kein HTTPs keine Transportverschlüsselung. Angreifer antwortet mit gewünschtem Steuerbefehl. Nutzdaten sind mit bekannten Schlüsseln verschlüsselt und signiert. Auto akzeptiert Befehle nur wenn Fahrgestellnummer übereinstimmt, sonst sendet es seine Fahrgestellnummer per sms an den Angreifer. ConnectedDrive kann deaktiviert werden, kann über selben Angriff aktiviert werden (Konfigurationsupdate). Ein gemeinsamer Schlüsselsatz für alle Fahrzeuge -> Netzwerkschlüssel (Single Mission Key), Single point of Failure. Keine Authentifizierung der ConnctedDrive Cloud. Mitgliedschaft im Netzwerk genügt, Angreifbar selbst wenn Schlüsselsatz besser geschuutzt wäre (Steuergerät als Krypto-Koprozessor. Generierte Nachrichten werden von jedem anderen Steuergerät akzeptiert). Unnötige Preisgabe von zur Validieren nutzbaren Infos.

8.4 Projekte

8.4.1 Kastel

Funktionale Domäne: Komponenten des Smart Home welche gemeinsam eine bestimmte Funktion erfüllen. Kompenten sind Sensorknoten und Energiemanagement (Gateway, Router, Switches). Sicherheitszonen: Setzt Komponenten des Smart-Home mit gleichem Schutzbedarf in Vertrauensbeziehung. (intern smart home gerät und extern (internet)) Sicherheitszonen sind orthogonal zu dem Domänenmodell und physischer Topologie. Funktionale Domänen können sich über mehrere Sicherheitszonen erstrecken oder umgekehrt. Zonenübergänge müssen kontrolliert werden. Sicherheitskonzept und Security

by Design wichtig. Erfolgreiche Angriffe auf Teile des Netzes kompromitteren nicht Gesamtsicherheit.

8.4.2 Flegsens und Mose

Flegsens untersucht Möglichkeit einer sicheren und flexiblen Überwachung von Grenz und Liegenschaften mit Hilfe drahtloser Sensornetze mit Rücksicht auf Angreifer, unzuverlässige Hardware/Kommunikation, feindliche Umwelt, begrenzte Ressourcen der Sensorknoten. Betrieb von Hardware in rauer Umgebung, Protokolle zum Betrieb und Ortung von Eindringlingen. Absicherung gegen Angriffe mit IT-Mitteln. Evaluation der Hardware zum Verständnis von Praxisproblemen, Entwurf und Simulation von Protokollen. Schutzziele, Riskoanalyse und Sicherheitskonzept. Anforderungen:

- Zeittreue -> Zeitsynchronisierung
- Ortstreue -> Lokalisierung
- Lebenszeit -> Duty Cycling
- Robustheit -> Knotenausfallerkennung, Partitionserkennung
- Verfügbarkeit -> Knotenausfallerkennung, Partitionserkennung, DoS Erkennung
- Trackingprotokoll benötigt Lokalisierung und Zeitsynchro