Proseminar Techn.Inf. Sicherheit in drahtlosen Sensornetzen (überarbeitet Januar 2009)

Thomas Weißgerber, Institut für Informatik, Freie Universität Berlin

In dieser Arbeit werden die technischen Sicherheitsanfragen bezüglich drahtloser Sensornetze erörtert. Sie entstand im Rahmen des Proseminars Technische Informatik im Wintersemester 2008 an der Freien Universität Berlin.

I. Einführung

Eine kleine Führung durch die Thematik der drahtlosen Sensornetze soll ein verwertbares Grundwissen um die Bedürfnisse und Probleme der Arbeit in diesem Bereich zur Verfügung stellen. Wer sich intensiver mit den Sensornetzen außerhalb der Sicherheitsfragen beschäftigt, wird hier kein verwertbares Material finden und diesen Anspruch erhebt die Arbeit auch überhaupt nicht.

A. Erläuterung

Ein drahtloses Sensornetz - im Folgenden kurz WSN/s (engl. wireless sensor network/s) - ist ein Verbund autonomer, hochspezialisierter Mikrorechner. Diese Rechner, untereinander drahtlos (per Funk) kommunizierend, fungieren als Knoten eines (Ad-Hoc)-Netzes (genauer: Multihop-Netz). Entscheidend ist, dass die einzelnen Knoten mit Sensoren ausgestattet sind und somit spezielle Beobachtungs-/ und Datenauswertungsaufgaben wahrnehmen.

Elementare Bestandteile der Knoten sind eine Recheneinheit (komplett, mit CPU und Speicher), ein oder mehrere physikalische Sensor/en, eine Energiequelle sowie eine Kommunikations-, (Funk)-Einheit. Man unterscheidet homogene WSN, bei denen alle Knoten identisch sind, von heterogenen WSN, die auf nochmals unterschiedlich spezialisierten Knoten aufbauen. Über Gateway-Knoten, die sowohl im WSN als auch in einem externen Netz liegen, kann auf das WSN zugegriffen werden - in Form von Informationsabruf mittels Anfragen oder per Erteilung die Knoten betreffender Befehle. Sensornetze als Ganzes sowie die einzelnen Bestandteile (Knoten) sollen üblicherweise ohne Wartung über eine möglichst lange Zeitspanne ihre Aufgabe erfüllen. Sowohl von Hard- als auch von Software-Seite aus kann man entscheidend darauf Einfluss nehmen, worauf im Allgemeinen nicht weiter eingegangen werden soll. Die Kontrolle der Zuverlässigkeit von Netzen aber - als Sicherheitsaspekt - wird in dieser Arbeit unter Anderem untersucht.

B. Aufgaben und Einsatzgebiete

Anhand der Spezifikation der WSN lassen sich mögliche Anwendungsgebiete zusammenfassen. Die Beobachtung sowie Verarbeitung von Ereignissen in einem festgelegten Areal ist das elementare Merkmal der Funktion von WSN. Als Sensoren kommen generell alle möglichen Änderungen der physikalischen Umgebung wahrnehmende Sensoren in Frage (optische, akustische, kinetische, chemische, thermische, anderweitige elektromagnetische sowie Hybrid-Sensoren). Je nach gewählter Netzabdeckung (Anzahl der Knoten*mittleres erfassbares Gebiet je Knoten) und der theoretisch frei wählbaren Größe der Sensoren lassen sich Gebiete von der Ausdehnung weniger Mikrometer bis hin zu vielen tausend Kilometern nach gewählter Präferenz überwachen. Einsatzmöglichkeiten finden sich somit unter Anderem in der Umwelttechnik, Geologie, praktischen Biologie, Sicherheitstechnik (kontextunabhängig), Verkehrsüberwachung als auch im militärischen Bereich oder in unbestimmter Zukunft in der Medizin.

C. Sicherheitsaspekte

Je nach Blickwinkel und Betrachtungsweise wird ein System unter bestimmten Prämissen analysiert und verifiziert. Das gilt natürlich ebenso für WSN. Es ergeben sich vor allem zwei Perspektiven der Thematik, nämlich zum einen die Analyse nach anwendungsspezifischen Sicherheitsanforderungen, wobei die Spezifikation der Systemsicherheit vornehmlich vom späteren Anwendungsfeld und dessen zu beachtenden Größen bestimmt wird und zum anderen die Lokalität der Analyse an sich, was heißt, dass Sicherheit sowohl als interne als auch als externe Größe mit Wirkung auf die Umwelt wahrgenommen wird. Wer sich mit WSN-Sicherheit befasst, den interessieren die Korrektheit der Arbeit des WSN (Sicherstellung der Einhaltung der Spezifikation) als geschlossenes System (eingeschlossen sei die Umwelt höchstens als Messdatenlieferant), womit sich diese Arbeit befasst, ebenso wie die möglichst vollständige Erfassung und Katalogisierung der Auswirkungen des WSN auf die Umwelt (hierbei wird das WSN als black box betrachtet), was jedoch außerhalb des technischen Rahmens liegt. Im Folgenden werden die relevanten Probleme vorgestellt und besprochen.

II. Korrektheit der Daten

Wesentliche Eigenschaft von WSN ist die Fähigkeit zur sensorischen Erfassung physikalischer Erscheinungen. Nach der Entscheidung über die spezifische technische Konzeption des WSN (Sensorik, Knotentypen, Ressourcenmanagement

usw.) stellt sich die Frage nach der Gewährleistung der korrekten Datenerfassung. Wir unterscheiden hierbei in erster Linie zwischen dem Problem der Knotenverteilung, welches die Stationierung der Knoten im zu überwachenden Gebiet unter Einbeziehung spezifischer Anforderungen des Anwenders als auch der umgebungsbedingten Voraussetzungen beschreibt, auf der einen und der Gebietsabdeckung durch vorhandene Knoten und der daraus resultierenden Qualität der Datenerfassung auf der anderen Seite. [1]
Nach der Erfassung der Daten steht deren Aufbereitung unter Berücksichtigung bestimmter, zumeist ökonomischer und anwendungsspezifischer Prämissen im Fokus unserer Betrachtung.

A. Stationierung der Knoten

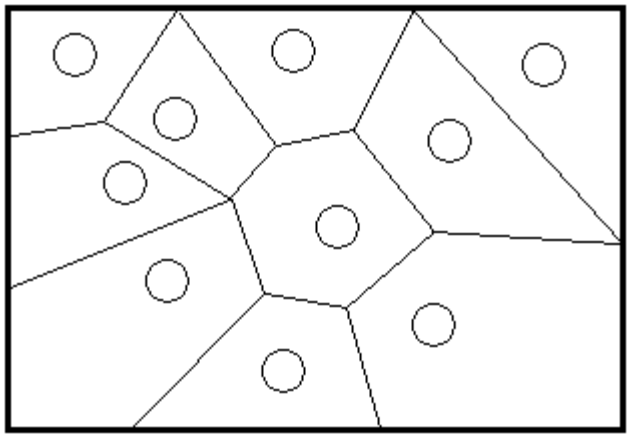
Die Verteilung der Knoten bestimmt maßgeblich und im Allgemeinen unumkehrbar über die Funktionalität des WSN. Es soll eine optimale Knotenverteilung erreicht werden, die eine maximale Gebietsabdeckung zulässt. Im Idealfall ist es also möglich, die Knoten unter Prämisse der eigenen Anforderungen gezielt zu platzieren. Das effektuiert die sensorische Arbeit. Auch gibt es Szenarien, in denen man jedoch nur begrenzten Einfluss auf den Verteilungsprozess hat, wodurch sich die Bedingungen für die Datenerfassung als zunehmend unkalkulierbar gestalten. In solchen Fällen ist es, je nach erwarteter Ausfallrate durch Fehlplatzierung, mitunter sinnvoll, das WSN zu erweitern, um die gewünschte Gebietsabdeckung zu erreichen. Außerdem ist es möglich, Sensoren mit variabler Leistung einzusetzen, um bei ungünstiger Verteilung nach vollständiger Lokalisation des Netzwerks durch Kalibrierung einzelner Sensoren eine bessere Abdeckung zu erreichen. Dabei muss bedacht werden, dass die Sensoren einen erheblichen Teil der energetischen Ressourcen aufbrauchen. Demzufolge sollte in der Konzipierung eines Netzes, das diese Technik nutzt, ein mittlerer Wert als Referenzleistung benannt werden. Erhöhte Sensorleistung führt, wenn man sie nicht entsprechend ausbalanciert (siehe IV - A), zu einer drastisch verkürzten Lebenszeit des Knotens. Mobile Knoten sind nach heutigem Stand der Technik in jeder Hinsicht unökonomisch und finden daher in diesem Kapitel keine Betrachtung.
Als allgemeines Beispiel dient ein Netzwerk mit Poisson-verteilen Knoten. [1] In der Literatur stößt man wiederholt auf Beschreibungen der Grid-Verteilung (das Gebiet ist auf gleich große Rechtecke aufgeteilt, in deren Mitte Knoten liegen), welche aber einen theoretischen Idealzustand darstellt und kaum reale Anwendung findet.

B. Datenerfassung

Wir gehen nun von einem WSN aus, dessen Knoten gesetzt und dessen Sensoren funktionstüchtig sind, was natürlich die Voraussetzung für eine sinnvolle Diskussion der elementaren Aufgabe der Sensornetze, nämlich die zuverlässige Erfassung physikalischer Ereignisse, ist. Daten über solche Ereignisse können nur dort erhoben werden, wo sie von Sensoren erfasst werden. An dieser Stelle soll der Begriff der Gebietsabdeckung definiert werden als der durch Sensoren erreichbare Anteil des zu überwachenden Gebietes. Die Knotenabdeckung beschreibt den Anteil an Knoten, der vollständig im Sensor-Radius anderer Knoten liegt. [1]
Die sensorische Erfassung wird direkt von den lokalen Gegebenheiten beeinflusst. So ist der Aufnahmeradius der Sensoren eventuell durch Hindernisse verengt, die Ausrichtung des Sensors eventuell ungünstig. Diese Beeinträchtigung kann nicht generell formalisiert werden und stellt eine bedeutende Fehlerquelle bei Berechnungen dar. Insbesondere können sich so lokale Voraussetzungen im Laufe der Zeit ändern, die jedoch aufgrund der anderweitig spezifizierten Sensoren überhaupt nicht erkannt werden. Als Beispiel diene ein (teilweise) verdeckter optischer Sensor, der somit keine verlässlichen Daten liefern kann. Die Feststellbarkeit ("detectability") beschreibt die Wahrscheinlichkeit, dass ein Ereignis von einem Sensor erfasst wird. Sie ist abhängig von der Umgebung des Knotens innerhalb des Sensorradius und von der Entfernung zum Sensor. Es haben sich zwei Modelle zur Berechnung der Feststellbarkeit herausgestellt, nämlich das "boolean sensing model" (BSM) und das "general sensing model" (GSM).
Das starre BSM beruht auf der Annahme, dass Sensoren gleicher Voraussetzungen innerhalb eines Radius ein Ereignis erfassen und außerhalb dessen nicht. Das GSM erweitert diese Methodik um einen von der Entfernung des Sensors zum Ereignis abhängigen Rückgabewert und wird folgendermaßen formalisiert [1]:

$$S(p,q)=\begin{cases}\frac{a}{\|p-q\|_2^{\beta}} & : r_0 \le \|p-q\| \le r \\ 0 & : \text{otherwise}\end{cases}$$

Es ist zu beachten, dass die Knotenabdeckung im GSM wegen des variablen Sensorwerts gleich Null ist, denn kein Knoten wird vollständig (sodass f(k)=1) von einem anderen erfasst.
Insbesondere für die Anwendung des WSN als System zur Überwachung von Eindringlingen ergibt sich daraus die Definition eines optimalen Pfades durch ein WSN.
Je nach Auslegung spricht man vom maximal breach path, welcher den Weg durch das WSN mit der geringsten Entdeckungswahrscheinlichkeit, und maximal support path, der den Weg mit der höchsten Entdeckungswahrscheinlichkeit darstellt.



Voronoi-Diagramm [1]

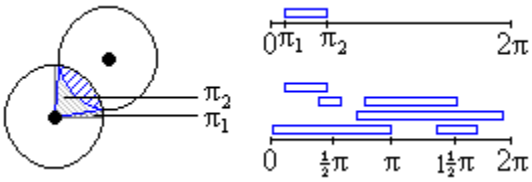
Kennt der Eindringling die Umgebungsbeschaffenheit und die Netzwerkstruktur, kann er die Wahrscheinlichkeit, entdeckt zu werden, entscheidend beeinflussen, indem er den maximal breach path wählt oder den Erfassungsgrad seines Pfades errechnet, wobei schwer formalisierbare variable Größen der Umgebung nicht beachtet werden (und i.A. gar nicht vollständig beachtet werden können) [1]:

$$\mathrm{dist}(\mathbf{p},V)=\inf\{\|\mathbf{p}-\mathbf{q}\|_2 : \mathbf{q}\in V\}$$

Im Gegenzug liefert ein solcher Algorithmus (beschrieben in [1], Seite 431, Listing 13.1) natürlich mit den errechneten worst-case-Coverage-Pfaden genau die Punkte, an denen idealerweise neue Sensoren zu platzieren sind. Aufbauend auf diesem Wissen können verschiedene Überlegungen und Berechnungen zur Bestimmung der Qualität der Erfassung angestellt werden. In einem uniformen Netzwerk (einem Netzwerk also, dessen Sensoren die gleichen Leistungsmerkmale haben und Poisson-verteilt sind) lässt sich die Sensor-Intensität an einem beliebigen Punkt q folgendermaßen formalisieren [1]:

$$C(\mathbf{q})=\sum_{s\in S} S(\mathbf{p}_s,\mathbf{q})=\sum_{s\in S}\frac{a}{\|\mathbf{p}-\mathbf{q}\|_2^{\beta}}$$

Gewöhnlich ist die Situation jedoch viel komplizierter, die Knoten sind eher zufällig verteilt und die Sensoren sind beeinträchtigt. Um also ein leistungsfähiges Protokoll für die praktische Anwendung zu entwerfen, muss man sich dieser Bedingungen statt einer zu theoretischen Betrachtung unter beinahe perfekten Umständen (2-dimensionales Gebiet) annehmen. Im Kern geht es nun darum, die Gebietsabdeckung des Netzes oder die Abdeckung an einem bestimmten Punkt zu ermitteln, ohne dass extern die Werte für jeden Knoten hinzugezogen werden müssen. Dies ist eine wichtige Funktion, insbesondere in dichten WSN und solchen, in denen aus energie- oder überwachungstaktischen Gründen Knoten unterschiedlich aktiviert/deaktiviert werden. Jeder Knoten überprüft den von ihm erfassten Bereichs auf die Abdeckung durch andere Sensoren und speichert die Informationen. Das Ergebnis soll eine Tabelle der lokalen (bis max. 2*r entfernten) Gebietsabdeckung sein. [1]



Aus den Entfernungsdaten der Knoten und den in diesen Tabellen gespeicherten Winkeln lässt sich nun für jeden Punkt die k-Coverage ermitteln, d.h. die Abdeckung durch wenigstens k Knoten. Dieses Modell lässt sich gut erweitern, insbesondere funktioniert es auch für die Annahmen, dass es sich um ein nicht uniformes Netzwerk handelt und dass nicht das BSM sondern das GSM zugrunde liegt, wobei dann nicht mehr von k-Coverage sondern von k%-Coverage gesprochen wird: Aufgrund der Formel des GSM und der ohnehin im Knoten gespeicherten Entfernungsangaben lassen sich sowohl die Erfassungswahrscheinlichkeit an einem Punkt p, der im Radius von k Knoten liegt, als die durchnschnittliche Erfassungswahrscheinlichkeit in einem ganzen Gebiet errechnen. Das k-Coverage-Problem stellt eines der Kernthemen in der WSN-Forschung dar. Es wurden viele Algorithmen vorgeschlagen, darunter auch effiziente wie der RKC/DRKC, welcher zusätzlich eine Menge an zu aktivierenden/deaktivierenden Knoten zurückgibt, um im festgelegten Bereich mindestens k-Coverage zu erlangen. [2]

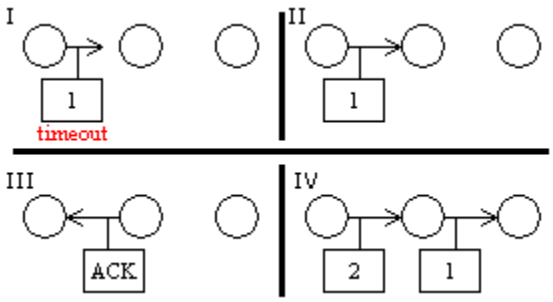
III. Datentransport

Die Anforderungen an Transportprotokolle in WSN sind von denen der Internet-Welt verschieden. WSN sind datenfixiert - anders als im Internet, wo große Mengen anonymer Daten geroutet werden müssen, passieren die Knoten eines WSN vornehmlich kleine Datenmengen meist gleichen Typs, deren Auswertung unmittelbar am Knoten im Vordergrund steht. Zumindest theoretisch sollte sich die Ressourcenknappheit auch in den Transportprotokollen widerspiegeln, jedoch stehen, wie in den Kapiteln zur Kollisions- und Staukontrolle beschrieben, nur den TCP/UDP-Äquivalenten ähnliche Methoden zur Verfügung. Wir beschreiben 3 Arten von packet loss: Nämlich Funkkanalfehler aufgrund von allg. Rauschen oder Kollision, Verlust durch Stau und Verlust aufgrund schlechter Synchronisation zwischen Sender und Empfänger (Verlust durch zu frühes Verschicken). [1]

A. Transport-Protokolle

Als grundlegendes Paradigma gilt der IEEE Standard MAC ("Media Access Control"), welcher besonderes Augenmerk auf die störungsfreie gemeinsame Nutzung des Übertragungsmediums durch viele Teilnehmer legt. Die speziellen Anforderungen an Ressourcen und die große Zahl an Netzteilnehmern (Knoten) im WSN machen den pauschalen Gebrauch der IEEE 802.11-Protokollfamilie jedoch unrentabel. Zum Datentransport zwischen Knoten können mehrere grundlegende Annahmen gemacht werden: Der Transport erfolgt entweder über einen Pfad (single-path solution) oder grundsätzlich über mehrere Pfade (multi-path-solution). Außerdem ist es sinnvoll, anwendungsspezifisch zwischen dem Transport einzelner Pakete oder Blöcke/Datenströme zu unterscheiden. Im Vordergrund steht die Erörterung der Wahrscheinlichkeit des zuverlässigen Transports. Aus der TCP-Welt wird das ACK (Acknowledgement, also die Bestätigung des Erhalts von Daten) übernommen, wobei von reinem end-to-end-ACK (von Quelle zu Ziel) aufgrund der vielfach höheren Energiekosten und des höheren Speicherbedarfs aufgrund der nötigen Pufferung mitunter vieler Pakete im Gegensatz zur hop-to-hop-ACK (von Knoten zu Knoten) abgesehen werden muss. [1] Aufgrund dieser eindeutigen Überlegenheit des hop-to-hop-ACK wird dessen Pendant im Folgenden beiseite gelassen. Insbesondere beim single-packet-delivery, beim

Versand von einzelnen Paketen also, machen NACKs (negative ACKs für nicht erhaltene Daten) keinen Sinn, denn ein eventueller Empfänger weiß nicht, wann sich an anderer Stelle ein Paket auf dem Weg zu ihm begibt, er kann keinen Zeitpunkt feststellen, an dem ihn ein für ihn vorgesehenes Paket nicht erreicht hat. Folglich bietet es sich an, auf Sender-Seite Timer zu implementieren und im Falle des ACK-Nichterhalts das Paket n-mal erneut zu senden.



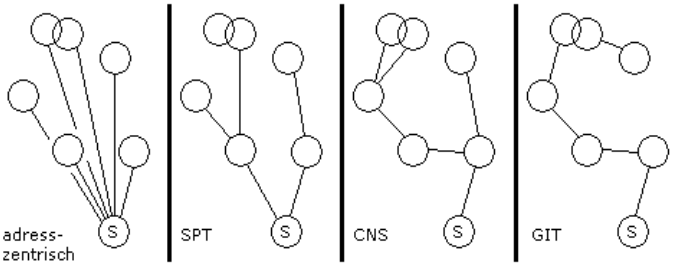
Eine Methode, mit der man aufgrund dieser Annahmen die Wahrscheinlichkeit der korrekten Lieferung des Pakets an den Endknoten errechnen kann, wird mit der HHR/HHRA in [1], vorgestellt. Die Wahrscheinlichkeit der korrekten end-to-end-Übermittlung wird hierbei als Produkt der dazwischen-liegenden hop-to-hop-Wahrscheinlichkeiten errechnet.
Betrachten wir nun den Fall des Transports mit einer multi-path-solution. Das ReInForM-Protokoll [1] setzt dabei für jeden Knoten i voraus, dass dieser die Distanz (in hops, also in Knotensprüngen) zur Senke und die Distanz der Nachbarn zur Senke kennt. Die Nachbarn werden anhand ihrer Distanz zur Senke in drei Klassen unterteilt, nämlich H- für die Knoten, die näher am Ziel sind als i, H0 für die Nachbarn, die die gleiche Entfernung haben und H+ für die, die einen hop weiter von der Senke entfernt sind. Außerdem wird angenommen, dass i seine Datenfehlerrate, also die Wahrscheinlichkeit, dass ein Paket verlorengeht, kennt. Die Anzahl der zu nutzenden Pfade wird folgendermaßen berechnet [1]:

$$P = \frac{\log(1 - r_s)}{\log(1 - (1 - e)^{n_s})}$$

Nun verteilt der Knoten i die Pfade auf die Nachbarn, wobei H- Vorrang vor H0 und H0 Vorrang vor H+ hat. Außerdem wird die Anzahl der Pfade, die von jedem dieser Knoten aus gegangen werden sollen, errechnet. Diese Daten werden nun versandt. Jeder Empfänger vergleicht sich selbst mit den mitgesandten Klassifizierungsinformationen. Gehört er bspw. zu H-, so ist die Wahrscheinlichkeit, dass er das Paket weiter-leitet, höher als würde er zu H0, H+ gehören. Danach verfährt der Knoten weiter die die Quelle. Die Zuverlässigkeit dieser Methode ist höher als die des unbeschränkten Flutens über alle Knoten auf der einen und die der single-path-delivery auf der anderen Seite. [1]
Zum korrekten Transport zwischen zwei Knoten, zur Gewährleistung der Integrität der Daten also, gibt es verschiedene Vorschläge, die jedoch alle die gemeinsame Annahme machen, dass dem Datenpaket in irgendeiner Weise zusätzliche redundante Daten zur Überprüfung der Integrität der Nachricht angefügt werden. An dieser Stelle sei zum Einen auf die Verwendung fehlererkennender/-korrigierender Codes, insbesondere aber auf den Abschnitt über die SPINS-Protokollfamilie, welche in (siehe IV - C) vorgestellt wird und auf elegante Weise die Lösung dieses mit dem Problem der zuverlässigen Authentifizierung verbindet, verwiesen.

B. Datenaggregation

In Sensornetzen werden sehr ähnliche Daten erfasst, berechnet und transportiert. Der Gedanke, redundante Daten zu erkennen und sinnvoll zu verarbeiten (d.h. evtl. fallen zu lassen) liegt daher nahe. Ein weiterer wichtiger Grund, optimale Datenaggregation zu betreiben, ist die allgemeine Ressourcenknappheit der Knoten (Energie, Speicherplatz). Desweiteren lässt sich durch kluge Datenaggregation die Zuverlässigkeit der Richtigkeit der Daten, die bspw. als Antwort auf eine Anfrage an das WSN geschickt werden, maximieren. Der Begriff des datenzentrischen Routings (DC-Routing) soll an dieser Stelle eingeführt werden. Der Grundgedanke des DC-Routing ist, dass die Datenaggregation, also die Komprimierung der Daten, während des Datenflusses zu einer Senke stattfindet. An den Knoten laufen mehrere Datenströme zusammenlaufen und werden dort entsprechend per geeigneter Aggregationsfunktion komprimiert. Bis zur Freigabe müssen die Daten natürlich im Knoten zurückgehalten werden. [3]



Der Aggregationsbaum wird am Ziel der Datenströme aufgebaut und setzt sich anwedungsspezifisch in Richtung der Quellen fort. Es lassen sich die verschiedenen Paradigmen der Anfrage an Senken umsetzen, so die Anfrage an ausgewählte Knoten, an alle Knoten und die automatische Selektion der Knoten per Methoden die auf Interest-Propagierung basieren wie "Direct Diffusion". Es ergeben sich bei geschickter Wahl des Aggregationsbaums, angepasst an die Erfordernisse, deutlich geringere Energiekosten weil weniger Übertragungen zwischen Knoten nötig sind (siehe Abb. oben). Allerdings kommt es zu höheren Verzögerungen aufgrund der Wartezeiten in den Knoten. [3]

C. Staukontrolle und Kollisionsvermeidung

Datenstaus können auftreten, wenn mehr Pakete generiert werden als das Netzwerk eigentlich verarbeiten kann. Immerhin treten die Knoten nicht als anonyme Spediteure der Daten auf (WSN sind datenzentrisch), wodurch sich kaum kalkulierbare Verzögerungszeiten, Diskrepanzen zwischen reiner Sendezeit von A nach B und der tatsächlich benötigten

Zeit inklusive der Operationen, die die Knoten zur Datenaufbereitung oder weiteren Koordination benötigen (2.2.2), einstellen. [1]
Wir gehen davon aus, dass die Knoten bereits einen geeigneten Puffer haben, jedoch ist dieser natürlich begrenzt. Ein Paket, das auf einen Knoten in einem solchen Zustand trifft, ist verloren - und mit ihm die für die Versendung aufgebrachte Energie.

> *Stau kann die Lebenszeit des Netzwerks/der Knoten sowie die Genauigkeit der Informationen verringern.*
>
> [1, Seite 459]

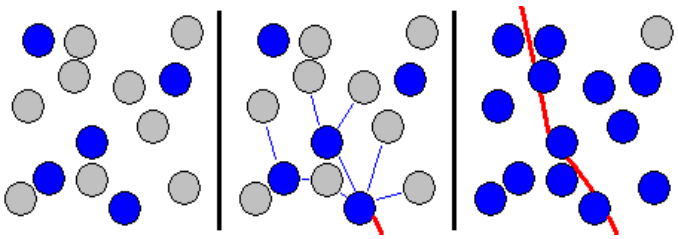
Welche Methoden zur Erkennung eines Staus bieten sich an?
Man kann den Pufferstatus abfragen und daraus Schlüsse auf einen möglichen Stau ziehen, jedoch ist diese intuitive Methode zu starr. Denn aus einem vollen Puffer kann man nicht ohne Weiteres erkennen, wie sich der Stau momentan entwickelt. Eine verbesserte Variante dieser Methode berücksichtigt neben dem Puffer an sich auch seine jüngste Veränderung. Ist der Puffer zwar voll, oder über einem bestimmten Schwellwert, wurde jedoch in jüngster Vergangenheit entlastet, so bedeutet dies, dass sich der Stau in der Auflösung befindet. [1]
Eine bessere Methode zur Stauerekennung ist das Channel Sampling. Ausgelöst wird diese Methode zur periodischen Überwachung des Sende-Kanals, wenn ein Knoten mit dem Senden von Paketen beginnen will. Die Idee ist, dass der Kanal ab dem Sendevorgang in gleichen Zeitabständen auf seine Auslastung abgetastet wird. Die Zeitspanne zwischen den Abtastvorgängen umfasst die Dauer des Sendens mehrerer Pakete. Im Laufe des Sendevorgangs ergibt sich somit ein Bild der durchschnittlichen Veränderung der Kanalauslastung, wodurch direkt auf einen eventuellen Stau geschlossen werden kann. [1]
Viele allgemeine Lösungsvorschläge zur Staukontrolle sind für den Einsatz in WSN nicht geeignet, da sie sehr stabile Netzwerke benötigen und für Sensornetze schlicht zu schwergewichtig sind. Für die Anwendung in WSN lassen sich im Wesentlichen zwei Paradigmen zur Staubehandlung herausstellen [1]: Zuerst sei die "rate control", also die Variierung der Senderate der Knoten, erwähnt. Kern dieser Idee ist, dass ein Knoten, wenn sich ein Stau anbahnt oder bereits vorliegt, seine Senderate herunterstellt, um den Kanal zu entlasten. Genauso ist es aber auch möglich, die Anzahl der Knoten, die mit dieser (hohen) Rate senden, zu variieren. Sollte der Fall eingetreten sein, dass der Puffer eines Knotens bereits voll ist, so kann dieser, da er die Daten kennt, aufgrund anwendungsspezifischer Annahmen und der damit implizierten unterschiedlichen Prioritäten der Daten entscheiden, welche Pakete (evtl. auch welche aus dem Puffer) fallen gelassen werden sollen. Zur Staubehandlung gibt es einige Lösungsvorschläge, dargestellt in [1] - daraus erweist sich CODA (Congestion Detection and Avoidance) als gute Allround-Lösung, die auf mehreren elementaren Kontrollmechanismen fußt. Insbesondere werden zwei Situationen bedacht: Wenn ein Knoten einen Stau in seiner Umgebung erkennt, sendet er einen entsprechenden Report (backpressure message, [1]) an die Quellen - an dieser Stelle kann eine Timer-gesteuerte Wartezeit eingesetzt werden. Es können zusätzliche Maßnahmen ergriffen werden wie das anwendungsspezifische Übergehen bestimmter Pakete oder die Variierung der eigenen Senderate. Ein Knoten, der eine solche Stau-Warnung erhält, geht analog vor und hat folgende Möglichkeiten: Er kann ebenfalls Pakete übergehen, seine Senderate variieren und den Report weiterleiten. So wird eine sukzessive Auflösung des Staus erreicht. Dieses Verfahren wird als "open-loop" [1] bezeichnet, da die Knoten ihrerseits keine Rückmeldung bekommen und sich die Stau-Reports nichtdeterministisch im Netzwerk fortsetzen. Ein zweiter Ansatz, der "closed-loop-regulation"-Mechanismus [1], setzt am Sensor an. Wenn dessen Senderate einen bestimmten Schwellwert (bezüglich der freien Kanakapzitäten) erreicht, nimmt er mittels eines Flag-Bits Einfluss auf das ACK-Verhalten der entsprechenden Senke. Wenn ein Sensor über einen bestimmten Zeitraum weniger als erwartet oder überhaupt keine ACKs mehr empfängt, reduziert er seine Senderate. Folglich kann auch der Zielknoten seine ACK-Rate bewusst verringern oder stoppen, um die beschriebene Reaktion der Quellknoten zu erreichen. Die Funkarchitektur birgt im Gegensatz zum kabelbasierten Datenaustausch das Risiko der Kollisionen von Datenpaketen. Dafür kann es mehrere Gründe geben. An erster Stelle sei das Hidden-Station-Problem genannt, welches auftritt, wenn ein Knoten gleichzeitig zwei Pakete verschiedener Sender empfängt - diese Pakete löschen sich gegenseitig aus. Das Exposed-Station-Problem beschreibt den Fall, dass zwei in gegenseitiger Reichweite liegende Sender gleichzeitig Pakete an verschiedene Empfänger übermitteln wollen - auch diese Pakete löschen sich gegenseitig aus, obwohl die jeweilige Sender-Empfänger-Kommunikation im Grunde nicht gestört ist. Außerdem hat ein Sender neben seinem Senderadius auch einen Störradius, in dem die versandten Pakete zwar nicht mehr lesbar sind, jedoch andere Pakete immer noch stören können. Als Lösung bietet sich das schon in 802.11 implementierte RTS/CTS-Schema ("Ready to send/Clear to send") an. [5]
Danach fragt ein Sender per RTS-Paket, in dem Adressat und Länge der zu sendenden Nachricht enthalten sind, den Empfänger, ob dieser bereit ist - ist das der Fall, geht ein CTS zurück und die eigentliche Übertragung kann beginnen. Die umliegenden Knoten erhalten das CTS ebenfalls und wissen somit, dass das Medium belegt ist, worauf sie sich ihrerseits bis zum Erhalt eines CTS bzw Ablauf eines timeouts mit dem Senden zurückhalten. Sollten zwei RTS kollidieren, worauf ihre jeweiligen Sender kein CTS bekommen, so warten diese nach einem timeout eine zufällig gewählte Zeit, bis sie ihr RTS erneut senden. Die Zeitdifferenz bewirkt, dass nun höchstwahrscheinlich nur ein Sender sein RTS losschickt, während der andere noch auf sein Timer-Signal wartet. Die Wahrscheinlichkeit, mit der nur ein RTS losgeschickt wird, kann durch eine entsprechende Wahl des möglichen Wartezeitraums erhöht werden.

IV. Stabilität und Diskretion

Die zuverlässige Lieferung von Daten ist selbstverständlich keine einmalige Anforderung. Ein Sensornetz soll über einen möglichst langen Zeitraum ohne elementare strukturelle, kostenintensive Veränderungen funktionieren. Das Kernthema ist hier das Energiemanagement der Knoten. Von der Beschädigung der Knoten als möglicher Stabilitätsfaktor wird hier aufgrund der geringen informationswissenschaftlichen Relevanz abgesehen. Gerade im Bereich der militärischen Nutzung liegt der Anspruch überdies auf Diskretion der Daten. Nun ist ein Szenario, in dem sich ein Unbefugter intimes Wissen über die Netzwerkstruktur und dessen detaillierte Funktionsweise beschafft und folglich auf die Arbeit des Netzes Einfluss nimmt oder selbst Informationen daraus bezieht, nicht abwegig. Im Folgenden werden Überlegungen zur Abwehr solcher Angriffe gemacht, wobei der Fall der Platzierung von Störsendern nicht berücksichtigt wird, da hierzu momentan aus dem Bereich der Funktechnik keine Lösungsansätze bekannt sind.

A. Energiehaushalt

Da die Knoten eines WSN auf einen begrenzten Energievorrat angewiesen sind, ergibt sich die einzige Möglichkeit der Energieeffizienkontrolle in der Variation der Energienutzung. Den größten Anteil an der zum Betriebt benötigten Energie stellen die Sensor- und die Funkeinheiten dar. Unter Gewährleistung einer ausreichenden Gebietsabdeckung können bestimmte, insbesondere mehrfach abgedeckte Knoten ihre Sensoren abschalten, um Energie einzusparen. Im Fall einer verstärkten sensorischen Aktivität der Nachbarknoten, können die entsprechenden Knoten aufgefordert werden, ihre Sensoren wieder einzuschalten, um die zuverlässige Erfassung des Ereignisses zu garantieren.

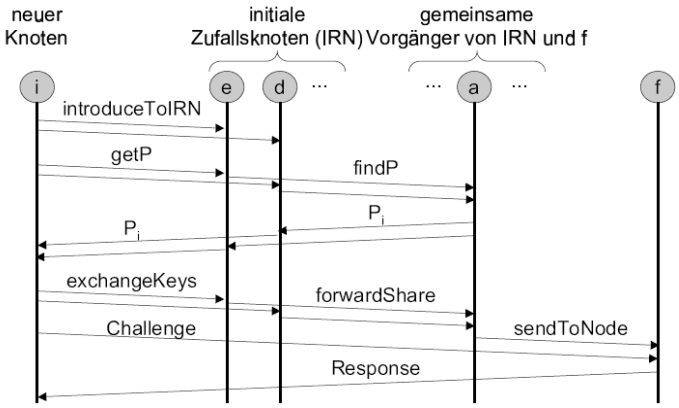


Bestimmte Anwendungsfälle lassen die manuelle Kontrolle der Sensorzustände zu, so muss die Überwachung eines bestimmten Gebietes nicht zwangsläufig die Aktivität aller Knoten des WSN erfordern. Wenn Ereignisse insbesondere an bekannten Punkten erwartet werden, deren Fortgang jedoch unbekannt ist, so können die Knoten im Randbereich solange im standby verbleiben, bis im erwarteten Kerngebiet sensorische Aktivität verzeichnet wird. Diese Methode eignet sich selbstverständlich nur, wenn zuverlässige Annahmen über das zu überwachende Gebiet und die erwarteten Ereignisse gemacht werden können. Ein Vorschlag, diese Annahmen zu verallgemeinern bieten die energierelevanten Lösungen des S-MAC-Protokolls bzw. dessen Erweiterung T-MAC: [7, 9]
S-MAC führt die Terminologie der Schlaf- und Wachzustände ein. Zugrunde liegt die Idee, dass die Knoten ihre Sensoreinheiten periodisch an- und abschalten. Um eine optimale Netzabdeckung zu behalten, ist eine Synchronisation der Knoten erforderlich, welche über SYNC-broadcasts durchgesetzt wird. Ziel ist, jedem Knoten einen festen Ablauf der sleep- und awake-Zustände zuzuweisen. Algorithmisch lösbar ist das relativ einfach durch Zuhilfenahme der in Kapitel II - B vorgestellten Methoden zur k-Coverage-Bestimmung beliebiger Punkte.
Eine Erweiterung dieses Protokolls um variable Zeiten beschreibt T-MAC. Die "Tagesabläufe" der Knoten sollen dem Datenverkehr angepasst werden. Das wird durch Hinzuziehen eines Timers erreicht, der den sich im Wachzustand befindenden Knoten noch eine bestimmte Zeit nach Erhalt des letzten Pakets auf weitere Nachrichten warten lässt, bevor der Knoten gemäß seiner Bestimmung wieder in den sleep-Modus wechselt. Dieser Zeitintervall kann ebenfalls an die bisherige Veränderung des Datendurchsatzes angepasst werden. Hierzu kann auf die für die Staukontrolle vorgestellten Methoden zur Überwachung des Datenverkehrs zugegriffen werden (siehe Kapitel III - C).
Ein sehr interessanter Punkt ist nun die Energieautarkie, was bedeutet, dass ein Knoten zumindestens theoretisch niemals aus energetischen Gründen den Dienst versagen wird, da er über einen Zugang zu einem steten lokalen Energiekreislauf verfügt.Wichtig ist, dass der Energieabfluss den Grenzwert der zur Verfügung stehenden Energie nie überschreitet.
In [9] werden diesbezüglich einige Vorschläge gemacht: Solarkollektoren finden heutzutage häufige Verwendung in energieautarken Netzen, immerhin bietet die bereits relativ weit fortgeschrittene Forschung schon effiziente Möglichkeiten. Desweiteren können halbleiterbasierte Thermowandler genutzt werden. Fraunhofer TEG stellte im Rahmen des EnAS [11] bereits ein sehr effizientes System vor, das eine Ausgangsspannung von 2-3V und 100µW Leistung bei einem Temperaturunterschied von nur 3K erzeugt. Der Piezo-Wandler basiert auf dem Effekt der Ladungsverschiebung bei mechanischer Verformung und ist ebenso wie ein System aus Spulen und Magneten, das vibrationsreaktiv ist, in der Lage im Niederspannungsbereich Energie zu erzeugen. [11]
Weiterhin wurden verschiedene Ideen zur Energiegewinnung an Flüssigkeiten vorgestellt, so nicht nur die Nutzung von fluider Bewegung durch Einsatz von Turbinen, sondern vielmehr die Ausbeutung magnetohydrodynamischer [10] oder druckdifferenzbasierter Effekte, die allerdings alles in allem heute aufgrund des hohen Materialverschleißes und der damit verbundenen niedrigen Lebensdauer des Energiewandlers für den Einsatz in WSN nicht in Frage kommen.

B. Abhörsicherheit

Die Kryptographie ist ein sehr bedeutsamer Zweig der Informatik und bietet schon detaillierte Forschungsergebnisse und vielfach erprobte Mechanismen, die man zur Diskussion der Thematik im Bereich der WSN hervorragend hinzuziehen kann. Unser Ziel unter diesem Blickwinkel ist die Verschlüsselung der übermittelten Daten, sodass fremde Zuhörer idealerweise keine Chance haben die abgehörten Funksignale inhaltlich deuten zu können. Zuerst sollen ein paar grundlegende Begriffe der Kryptographie erläutert

werden. Die ursprüngliche, zu verschlüsselnde Nachricht bezeichnet man als Klartext M, wohingegen der verschlüsselte Text als Chiffrat C bezeichnet wird. Man spricht jeweils von einer Verschlüsselungs-/Kodierungsfunktion e und einer Entschlüsselungs-/Dekodierungsfunktion d, wobei beide Funktionen abhängig von zwei Parametern sind: nämlich einem Text (M, C) und einem Schlüssel k. Folgende Identitäten lassen sich ableiten: C=e(M,k), M=d(C,k), C=e(d(C,k),k) usw. [4]. Ist der Schlüssel geheim und ausreichend groß, so ist eine sichere Kommunikation möglich. Abgesehen vom One-Time-Pad ist keine Verschlüsselungsmethode perfekt, d.h. "unknackbar", allerdings sind einige von ihnen "praktisch perfekt", was heißt, dass es unter akzeptablem Aufwand nicht möglich ist den Code zu knacken. Man unterscheidet zwei grundlegende Paradigmen [12]: Zum Einen gibt es symmetrische Verschlüsselungsverfahren. Hierbei existiert, wie bei der Begriffseinführung dargestellt, ein Schlüssel, der als Parameter für beide Funktionen dient - dieser muss also unbedingt geheim gehalten werden. Gängige Schlüssellängen sind 64 oder 128 Bit. Wichtig ist, dass M, falls es größer als der Schlüssel ist, geteilt werden muss - analog muss M, falls kleiner als der Schlüssel, aufgefüllt werden. Die versandten kodierten Blöcke sind also immer gleich groß (entsprechen der Schlüssellänge). Beispiele für solche Verfahren sind DES, 3DES, RC5, RC6, IDEA und der bekannteste Vertreter AES (Rijndael). Zum Anderen gibt es asymmetrische Verfahren ("public-key-Verfahren"), deren Besonderheit darin besteht, dass zwei verschiedene Schlüssel existieren, von denen nur derjenige, welcher zum Entschlüsseln nötig ist geheim gehalten werden muss, da sich vom öffentlichen (fürs Kodieren zuständig) nicht in angemessener Zeit auf den Dekodierschlüssel schließen lässt. Die Schlüssellängen schwanken sehr - das bekannte Verfahren RSA benutzt 512 Bit Schlüssel, andere moderne Verfahren kommen mit 100
Bit aus. Renommierte Beispiele sind RSA, El-Gammal und dessen Abwandlungen. Asymmetrische Verfahren gelten im Allg. als sicherer, allerdings ist ihr Ressourcen-verbrauch nicht zuletzt aufgrund der größeren Schlüssel sehr viel höher als der der symmetrischen Verfahren. Ein Vergleich aus [4] verdeutlicht das: Die Addition zweier 8-Bit-Zahlen auf dem represäntativen WSN-System MICA2 kostet 625 pAs an Energie. Das Verschlüsseln von 64 Bit Klartext mit RC5 kostet 1,3 µAs, eine einzige public-key-Chiffre-Operation hingegen kostet 4-28 µAs (das ist ein Faktor von 7000000-45000000 im Vergleich zur Addition). Die Wahl des Paradigmas für die so stark ressourcenbeschränkten WSN fällt somit sehr leicht, nämlich auf die symmetrischen Verfahren.
Die Wahl des genauen Algorithmus ist laut [4] erst einmal relativ egal, die Qualität seiner Implementierung äußert sich jedoch in Laufzeit und Ressourcenverbrauch (Speicher, Energie). Gefährlich ist ausdrücklich nur der Fall, dass der Schlüssel bekannt wird (die Funktionen können öffentlich sein). Daher wird für jedes Knotenpaar ein eigener Schlüssel verwendet. Problematisch ist nun genau dieser Vorgang. Außerdem, aber das fällt in den Bereich der physischen Sicherheit, zu dem aufgrund der höchst verschiedenen Anforderungs- und Umgebungsprofile keine allgemein gültigen Konzepte aufgestellt werden können [4], ist es möglich, dass ein Knoten geöffnet und dessen Speicher ausgelesen wird, wodurch auch der Schlüssel preisgegeben wird. Ein Vorschlag zielt darauf ab, im Falle eines Hardware-Zugriffs einen Selbstzerstörungsmechanismus einzuleiten - optimalerweise löst dann ein Bewegungssensor die Leerung des Speichers aus. In [4] wird das Protokoll SKEY vorgestellt, welches eine sichere Schlüsselverbreitung ermöglicht und im Folgenden im Kern erläutert werden soll: SKEY basiert induktiv auf der Annahme, dass jeder Knoten gemeinsame Schlüssel mit den Knoten auf seinem Aggregationspfad hat. Ein neu hinzustoßender Knoten erhält vom Nutzer zwei wichtige Informationen, anhand derer er sich in die bestehende Netzwerk- und verschlüsselte Kommunikationsstruktur integrieren kann: nämlich seine "Initial Random Nodes" (IRN) und Authentifizierungs-Tickets.

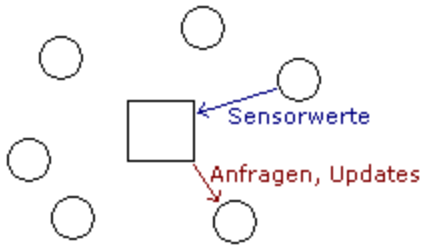


Ein neuer Knoten i meldet sich bei seinen IRN (e, d) und lässt sich von diesen seinen Aggregationspfad mitteilen (IRN suchen auf eigenen Aggregationpfaden nach Knoten, die mit i eine Aggregationsbeziehung haben (a); diese antworten mit ihrem Pfad zu i; wenn alle diese Pfade übereinstimmen, wird an i übermittelt). i will nun mit jedem Knoten, insbesondere mit Knoten f auf seinem Aggr.-Pfad Schlüssel austauschen. Dazu erzeugt er einen zufälligen Schlüssel, der aufgeteilt an die IRN verschickt wird, welche auf ihren Pfaden nach Vorgängern von f suchen und ihren Schlüsselteil über diese Vorgänger getrennt voneinander an f übermitteln. Nach abgeschlossenem Prozess wird per "Challenge-Response" überprüft, ob i und f wirklich den gleichen Schlüssel haben.

C. Authentifizierungssicherheit

Nachdem Vorschläge zur Geheimhaltung der Daten gemacht wurden, legen wir den Fokus auf die Gewährleistung von Integrität und Authentizität der Daten. Demzufolge soll es darum gehen, Methoden zu finden, die sicherstellen, dass ein Sender seine Nachricht auch wirklich abgeschickt hat, bzw. genauso abgeschickt hat wie ein anderer Knoten sie empfängt.
Die in der TCP-Welt übliche Verwendung digitaler Signaturen scheidet aus, da dieses Verfahren auf asymmetrischer Kryptographie fußt, welche - wie bereits bekannt - für die Nutzung in WSN aufgrund der Ressourcenanforderungen nicht in Frage kommt. Ein ähnliches Konzept, basierend auf

symmetrischen Verfahren, stellt die Nutzung des "message authentication code" (MAC) dar. [13]
Im Einzelnen wird hierbei die Nachricht mittels eines geheimen, Sender und Empfänger bekannten Schlüssels komprimierend (Blockchiffre-Verfahren) oder per spezieller Hashfunktion (key-Hashing) kodiert . Das Ziel ist es ausdrücklich nicht, die Informationen des Klartexts zu verschlüsseln sondern vielmehr eine den Klartext identifizierende Bitfolge, den MAC, zu erhalten, die gemeinsam mit der eigentlichen Nachricht verschickt wird. Der Empfänger verschlüsselt seinerseits die übermittelten Klartext und vergleicht das Ergebnis mit dem MAC. Sind beide Blöcke identisch, so kann mit ziemlicher Gewissheit gesagt werden, dass die Nachricht so verschickt und nicht manipuliert wurde. [15]
Oft liegt die Klartextgröße unterhalb der Blocklänge des Crypto-Verfahrens. Hier würde die Verwendung von zusätzlichen Authentifikationsblöcken doppelte Energiekosten verursachen. Es bietet sich an, die Differenz zwischen Klartextgröße und Blocklänge entsprechend zu nutzen. Hierzu müssen sich Sender und Empfänger auf ein gemeinsames Muster einigen - das kann im Rahmen des Schlüsseltauschverfahrens stattfinden -, mit welchem der freie Raum gefüllt wird und anhand dessen die Nachricht beim Dechiffrieren verifiziert werden kann. Andere Vorschläge sind das Auffüllen mit einer Sequenznummer, einem Zeitstempel oder Teilen des Klartexts. [4]
Abschließend sei die SPINS-Protkollfamilie, SNEP (Secure Network Encryption Protocol) und µTESLA (Micro Timed Efficient Stream Loss-tolerant Authentication), erwähnt, welche eine etwas ressourcenintensive, aber praktikable Allround-Lösung für die Probleme der Vertraulichkeit, Authenzität, Integrität, Aktualität und optimaler Ressourcennutzung darstellt. [14]
Die besondere zugrunde liegende Idee ist die der inhomogenen Netzstruktur: Neben den normalen Knoten gibt es Basisstationen, die auf deutlich höhere Ressourcen zugreifen können und im Allg. auch über eine bessere Energiequelle als Batterien verfügen.



Diese Struktur ermöglicht die Einstufung einer Basisstaion als sehr vertrauenswürdig, da diese per Definition nicht autonom ist und somit unter ständige Beobachtung durch den Nutzer gestellt werden kann. In SNEP besitzt jeder Knoten einen gemeinsamen Master-Schlüssel mit seiner Basisstation, aus dem die Schlüssel für die De-/Kodierung und MAC generiert werden. SNEP nutzt den RC5-Chiffre. Wichtig ist, dass Basisstation und Knoten paarweise synchrone Zähler haben. [14]

Das Besondere an µTESLA ist die Erzeugung von Asynchronität durch Festlegung einer Periode, innerhalb der ein bestimmter Master-Schlüssel gilt. Dieser Schlüssel wird regelmäßig erneuert. [13]

Literaturangabe

[1] Holger K., Willig, A.: Protocols and Architectures for Wireless Sensor Networks, Seiten 415-468; Wiley&Sons, 2007.
[2] J.Mohamed Hefeeda, Majid Bagheri: Efficient k-Coverage Algorithms for Wireless Sensor Networks; ftp://fas.sfu.ca/pub/cs/TR/2006/CMPT2006-22.pdf
[3] Sven Haidan: Datenzentrisches Routing und Directed Diffusion in drahtlosen Sensornetzen; http://i11www.iti.uni-karlsruhe.de/teaching/WS_0405/sensornetze/ausarbeitungen/haidan_DirectedDiffusion.pdf
[4] Erik-Oliver Blaß: Sicherer, aggregierender Datentransport in drahtlosen Sensornetzen; Universitätsverlag Karlsruhe, 2007.
[5] Thomas Haenselmann: Kommunikation in Sensornetzen; http://www.informatik.uni-mannheim.de/pi4.data/content/courses/2004-ws/sensornetze/radio.pdf
[6] Paul-Christian Plückhahn: Mac protocols for Sensor Networks; http://cst.mi.fu-berlin.de/teaching/SS07/19554-S-TI/reports/plueckhahn07mac-protokolle.pdf
[7] Sensys: The ACM Conference on Embedded Network Sensor Systems; http://sensys.acm.org/
[8] Dr. Faruk Bagci: Vorlesung Sensornetze, Kapitel 5 - Verbindungssicherungsschicht; http://www.informatik.uni-augsburg.de/lehrstuehle/sik/lehre/ws/sensornetze/Folien/05_DataLink.pdf
[9] Axel Bindel, Dr. Friedemann Tonner: Energie-autarke Funksensornetze; http://www.elektroniknet.de/home/automation/fachwissen/uebersicht/feldebene/sensoren-aktoren/energie-autarke-funksensornetze/druckversion/
[10] wikipedia-Artikel über Magnetohydrodynamik; http://de.wikipedia.org/wiki/Magnetohydrodynamik
[11] EnAS; http://www.energieautark.com/
[12] unbekannter Autor, FH Würzburg: Kryptographie ”Grundlagen - Terminologie”; http://www.fh-wuerzburg.de/fh/fb/all/personal/interper/WSCHNELL/Grundla.pdf
[13] Jörg Kalok: Security in Wireless Sensor Networks; http://cst.mi.fu-berlin.de/teaching/WS0607/19554-S/reports/kalok07security_slides.pdf
[14] Danat Pomerantes: Sicherheit in Sensornetzen; http://www.vs.inf.ethz.ch/edu/SS2003/DS/slides/12_sicherheit.pdf
[15] Fabian Eltz, Matthias Schubert: Message Authentication Codes; http://www.cs.uni-potsdam.de/ti/lehre/06-Kryptographie/slides/slides-05.pdf

fertiggestellt am 28.Januar 2009; Thomas Weißgerber (email: thomas.weissgerber@fu-berlin.de)