

SeNeTs: Test- und Steuerungsumgebung für Software in großen drahtlosen Sensornetzen

Jan Blumenthal, Matthias Handy, Dirk Timmermann
Institut für Angewandte Mikroelektronik und Datentechnik, Universität Rostock
(jan.blumenthal, matthias.handy, dirk.timmermann)@e-technik.uni-rostock.de

Abstract: In dieser Arbeit wird die Test- und Steuerungsumgebung SeNeTs für Software in drahtlosen Sensornetzwerken vorgestellt. SeNeTs gestattet durch Nutzung eines zweiten Kommunikationskanals die Übertragung von zusätzlichen Statusinformationen der Sensorknoten ohne die primäre Anwendung zu beeinflussen und erlaubt somit eine Überprüfung der Software unter realistischen Bedingungen. SeNeTs vereinfacht überdies die Fehlersuche und stellt komfortable Update-Mechanismen für Sensorknoten-Software zur Verfügung.

1 Einleitung

Sensornetzwerke bestehen aus tausenden unbeaufsichtigt arbeitenden Sensorknoten. Test und Administration dieser Sensorknoten erfordern optimierte Software, um redundante Messinformationen rechtzeitig zu erkennen sowie auszusortieren, wodurch der Kommunikationsaufwand innerhalb des Netzwerkes minimiert wird. Während der Implementierungs- und Testphase eines Sensornetzwerkes ist es jedoch erforderlich, so viele Informationen wie möglich vom Netzwerk zu erhalten, um die korrekte Funktion der Software nachzuweisen. Zusätzlich ist es in der Testphase häufig erforderlich, eine aktualisierte Software auf die Sensorknoten zu übertragen. Für diese Anwendungsfälle wurde die Test- und Steuerungsumgebung SeNeTs für Sensornetzwerk-Software entwickelt [BH04].

2 SeNeTs - Architektur

2.1 Begriffsdefinitionen

Eine Sensorknoten-anwendung (SKA) umfasst sämtliche Software-Komponenten, die auf einem Sensorknoten installiert sind, z.B. Sensortreiber, Betriebssystem und Middleware-Komponenten wie Ortsbestimmung. Eine Sensornetzwerk-anwendung (SNWA) beschreibt die Aufgabe des gesamten Sensornetzwerkes. Sie umfasst alle Sensorknoten-anwendungen und legt das Zusammenwirken aller Systemkomponenten fest. Ein SeNeTs-Netzwerk ist ein Sensornetzwerk, das von SeNeTs gesteuert wird.

2.2 SeNeTs-Komponenten

Software für drahtlose Sensornetze wird oft nur simuliert, da echte Sensorknoten entweder noch nicht existieren oder zu teuer sind. Nachteilig ist jedoch die sequenzielle Behandlung parallel eintreffender Ereignisse, da damit die Realität nur ungenügend abgebildet wird und das Simulationsergebnis verfälscht wird. In SeNeTs laufen Sensorknotenapplikationen verteilt auf verschiedenen Host-Systemen (PCs, PDAs, Entwicklerboards). Die parallele Ausführung führt im Gegensatz zu herkömmlichen Simulation zu einer Entkopplung von Anwendung und Administrierungsumgebung.

Die Entwicklung und insbesondere der Test von verteilten Anwendungen sind schwierig, da Informationen über den internen Zustand der einzelnen Knoten zentral ausgewertet werden müssen, um die korrekte Funktionsweise des Gesamtsystems zu überwachen. Abhilfe kann das Versenden von Zustandsinformationen (Logging) in der Entwicklungsphase schaffen. Systeme mit zusätzlicher Logging- und Debuggingunterstützung beeinflussen aber das Verhalten der Applikation. Das Übertragen von Logging-Nachrichten führt automatisch zum verzögerten Übermitteln anwendungsspezifischer Daten. Besonders in drahtlosen Anwendungen mit begrenzten Ressourcen resultiert dies in einem modifizierten Zeitverhalten sowie einem frühen Ausfall von Knoten durch den erhöhten Kommunikationsaufwand und, als Konsequenz, zu falschen Ergebnissen. In einem Netzwerk aus n Knoten nimmt zudem die Kanalkapazität mit $\sqrt{(1/n)}$ ab. Das Transportmedium stellt folglich einen Flaschenhals dar [LBC⁺01]. In Sensornetzwerken mit tausenden von Knoten beeinträchtigt dieser Bottleneck-Effekt das Applikationsverhalten erheblich.

Die Architektur von SeNeTs erlaubt die Reduzierung des Flaschenhals-Effektes durch Einführung von zwei unabhängigen Kommunikationskanälen (Abb. 1). Der primäre Kommunikationskanal entspricht dem originären Kommunikationskanal der verteilten Anwendung. Er wird durch die Sensornetzwerkanwendung definiert und entspricht dem Übertragungsverfahren der Sensorknoten, z.B. Bluetooth oder Zigbee.

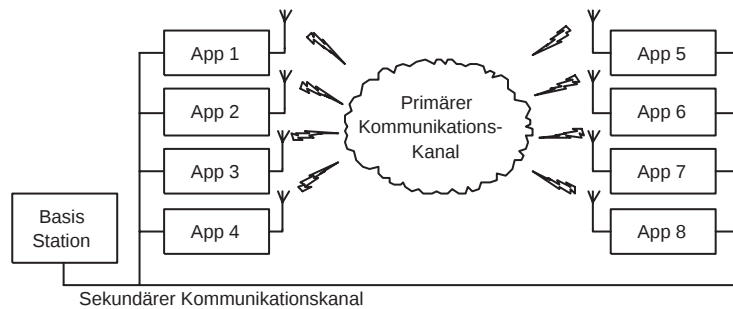


Abbildung 1: Kommunikationskanäle in SeNeTs

Der sekundäre Kommunikationskanal ist für die Administrierung der SeNeTs-Komponenten vorgesehen. Dieser Kanal überträgt Steuerungs- und Loggingnachrichten. Er ist unabhängig vom primären Kommunikationskanal und benutzt eine andere

Übertragungsmethode, z.B. Ethernet oder Ultraschall. Die Administration von verteilten und parallel arbeitenden Anwendungen erfolgt über diesen sekundären Kommunikationskanal. Dazu wird eine kaskadierte Infrastruktur aus SeNeTs-Komponenten aufgebaut. Abb. 2 zeigt die wichtigsten SeNeTs-Komponenten: Knotenanwendungen, Network-Server, Application-Server und optionale Komponenten zur Evaluierung und Visualisierung der Ergebnisse. Alle Komponenten, mit Ausnahme der Knotenanwendungen untereinander, kommunizieren ausschließlich über den sekundären Kommunikationskanal.

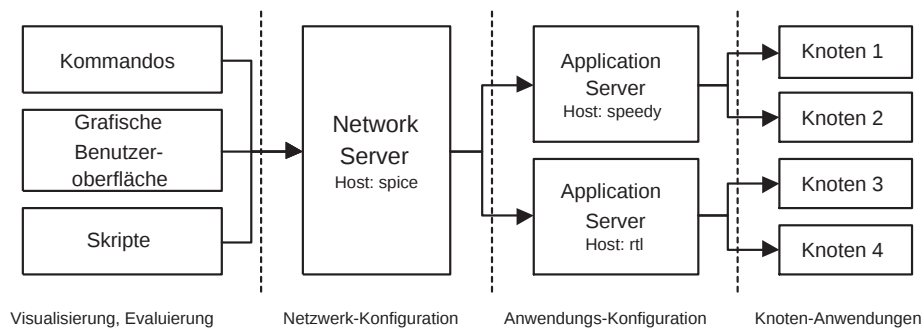


Abbildung 2: Sekundärer Kommunikationskanal in SeNeTs

2.3 Network-Server

Der Network-Server (NS) administriert komplette Sensornetzwerke und assoziierte Sensorknoten. Er startet, stoppt und überwacht Knotenanwendungen. In einem SeNeTs-Netzwerk existiert max. ein Network-Server, der aber mehrere Sensornetzwerke simultan verwalten kann. Ein Network-Server wird als Dienst des Betriebssystems gestartet und ermöglicht die Interaktion mit externen Programmen.

Der Network-Server initialisiert den sekundären Kommunikationskanal und öffnet zusätzliche Sockets. Externe Programme, wie z.B. Skripte, Webseiten oder Telnet-Clients, können sich über diese Sockets mit dem Network-Server verbinden und erlauben somit eine externe Administration.

Die verteilten Anwendungen sind über den sekundären Kommunikationskanal mit dem Network-Server verbunden. Sie senden Nachrichten über den aktuellen Zustand über diesen Kanal. Angekoppelte externe Programme, z.B. Visualisierungsapplikationen mit grafischer Ausgabe, können diese Nachrichten über installierte Callbacks empfangen.

2.4 Application-Server

Der Application-Server (AS) verwaltet die Instanzen der Knotenanwendungen, die auf einem Host-System laufen (Abb. 2). Er agiert als Brücke zwischen den Knotenanwendungen und dem Network-Server. Der Application-Server arbeitet unabhängig vom Network-Server. Beim Programmstart wird eine Pipe zur Kommunikation mit dem Network-Server geöffnet. Die Kommunikation erfolgt durch vordefinierte Kommandos, die im Application-Server empfangen werden und an die Knoten-Applikationen weitergeleitet werden. Ist die Kommunikation zum Network-Server gestört, werden die Knotenanwendungen nicht beeinflusst. In diesem Fall sind lediglich die Steuerungs- und Loggingfunktionalitäten nicht nutzbar bis der sekundäre Kommunikationskanal wiederhergestellt ist.

Wie der Network-Server startet auch der Application-Server als Dienst des Betriebssystems. In der Startphase benötigt der AS zusätzliche Konfigurationsparameter der Knotenanwendungen. Sie sind notwendig, um die benötigte Knotenhardware im Hostsystem zu referenzieren und den Knotenanwendungen zuzuweisen. Gesetzt den Fall, ein System enthält zwei lokale Bluetooth-Module, die das Funkmodul eines Sensorknotens darstellen (Abb. 2), dann benötigen die Applikationen die logischen Geräteadressen im Hostsystem, um zur Laufzeit die korrekte Hardware ansprechen zu können.

2.5 SeNeTs-Anwendung

Service-basierte Anwendungen für drahtlose Sensorknoten werden nach einem Software-Schichtenmodell (Abb. 3a) entwickelt [BHG⁺03]. Oberhalb der Hardwareschicht wird ein spezialisiertes Betriebssystem (OS) wie z.B. TinyOS installiert. [Be04]. Dieses Betriebssystem enthält Module, die die Treiber kapseln. Ein Treiber enthält Software zur Initialisierung des Messprozesses und zur Weiterleitung der Daten. Oberhalb des Betriebssystems liegt die Middleware. Sie enthält Dienste zur Datenaggregation und Positionsbestimmung.

Die SeNeTs-Anpassungsschicht besteht aus mehreren Komponenten, die einer Sensorknoten-anwendung hinzugefügt werden können, um diese auf einem Hostsystem auszuführen. Abb. 3b veranschaulicht die Anpassungsschicht, die aus den Modulen Logging, Steuerung, Debugger, SeNeTs-Hardwareabstraktion und einer optionalen Umgebungsemulati-on besteht. Durch diese zusätzlichen Komponenten wird eine realistische Test- und Steuerungsumgebung garantiert. Nach der Entwicklungs- und Testphase der Knotenanwendungen werden die optionalen SeNeTs-Komponenten durch einfache Rekompilierung bei veränderten Compiler-Schaltern entfernt.

Eine Anwendung, die aus Sensorknoten-anwendung und Komponenten der SeNeTs-Anpassungsschicht besteht, wird als SeNeTs-Anwendung bezeichnet (SeA). Der Quelltext einer Sensorknoten-anwendung wird durch das Hinzufügen der Komponenten nicht beeinflusst. Die Verknüpfung von SNA und SeA erfolgt z.B. über Macros. Es ist aber meist nicht notwendig, die Sensorknoten-anwendung an SeNeTs anzupassen.

Eine SeNeTs-Anwendung läuft als Prozess im Hostsystem. Durch die Verwendung eines

„eigenen“ Betriebssystems arbeiten die SeNeTs-Anwendungen vollständig autonom im Host-System. Sie interagieren nicht mit anderen Prozessen. Eine Ausnahme bildet die während der Startphase geöffnete Pipe zum Application-Server, mit dem Kommandos und Logging-Nachrichten ausgetauscht werden.

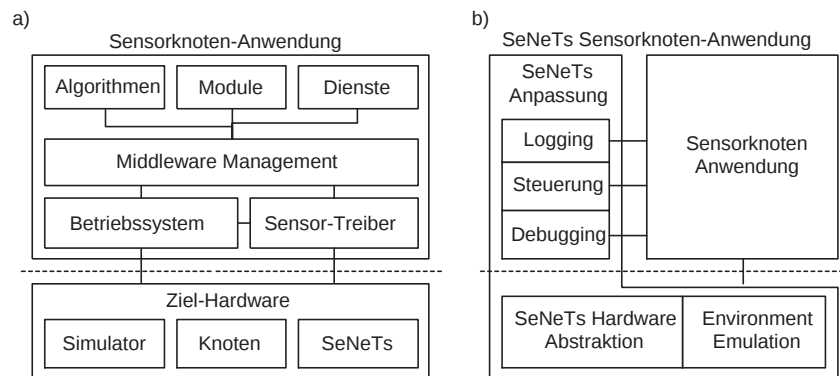


Abbildung 3: a) Software-Schichtenmodell einer Sensorknoten-Anwendung (SKA), b) Software-Schichtenmodell einer SeNeTs-Anwendung (SeA)

3 Zusammenfassung

In dieser Arbeit wurde SeNeTs vorgestellt, eine Test- und Steuerungsumgebung für große drahtlose Sensornetzwerke. SeNeTs vereinfacht den Entwicklungsprozess von Anwendungen für Sensornetzwerke und ist unabhängig von einer speziellen Sensorknoten-Plattform. SeNeTs nutzt für die Steuerung und Administration einen unabhängigen sekundären Kommunikationskanal und verhindert dadurch eine Überlastung des primären drahtlosen Kommunikationskanals der Sensorknoten.

Literatur

- [Be04] Berkeley WEBS. TinyOS. <http://today.cs.berkeley.edu/tos>. 2004.
- [BH04] Blumenthal, J. und Handy, M. The SeNeTs Project. <http://www.senets.org>. 2004.
- [BHG⁺03] Blumenthal, J., Handy, M., Golasowski, F., Haase, M., und Timmermann, D.: Wireless sensor networks - new challenges in software engineering. In: *Proceedings of 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Lisbon, Portugal. September 2003.
- [LBC⁺01] Li, J., Blake, C., Couto, D. S. J. D., Lee, H. I., und Morris, R.: Capacity of ad hoc wireless networks. In: *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking (MobiCom '01)*. Rome, Italy. July 2001.