

Zusammenfassung

Drahtlose Sensornetzwerke (WSN) ermöglichen eine Vielzahl neuartiger Anwendungen, e.g. im Körper eingebettete Messdatenerfassung, Lebensraumerforschung und Beobachtung von Ozeandaten. Dennoch haben diese Netzwerke andere Anforderungen als konventionelle Systeme. Selbst-Konfiguration, energieeffizienter Betrieb, Kollaboration und Datenverarbeitung innerhalb des Netzwerkes sind Beispiele für wichtige Anforderungen. Um diesen Anforderungen gerecht zu werden, spielt die Systemsoftware eines Sensorknotens eine grundlegende Rolle: Sie sollte nützliche Abstraktionen zur Verfügung stellen, um die Entwicklung von Anwendungen zu ermöglichen, und gleichzeitig mit den beschränkten Ressourcen eines Sensorknotens haushalten können.

Das Spektrum der möglichen Anwendungen eines Sensorknotens beinhaltet verschiedenartige Aufgaben wie Uhrensynchronisation, Datensammlung, Signalverarbeitung und Datenfusion. Der traditionelle Ansatz auf diesem Gebiet ist es Betriebssystemkonzepte mit dramatisch reduziertem Funktionsumfang anzubieten. In dieser Arbeit präsentieren wir jedoch einen alternativen Ansatz. Unser Betriebssystem ermöglicht potenziell beliebige Funktionalität, welche sich dynamisch an das aktuelle Anforderungsprofil anpasst. Die Grundidee ist es Dienste anzubieten, die über ein Cluster von Knoten verteilt sind, anstatt ein ganzes System auf einem Knoten auszuführen.

Kooperation ist das Schlüsselwort um komplexe Aufgaben auf ressourcenbeschränkten Sensorknoten auszuführen. Unser Betriebssystem unterstützt Kooperation unter benachbarten Knoten mit Hilfe des Konzeptes verteilter Dienste. Dafür entwickeln wir eine biologisch inspirierte Heuristik verantwortlich für die Platzierung der Dienste im Sensornetzwerk. Wir entwickelten zwei Versionen der Heuristik mit unterschiedlicher Komplexität und Leistung. Beide arbeiten vollständig verteilt und basieren auf lokalen Informationen und Regeln. Die für die Organisation der Dienste nötige Kommunikation wird mit Hilfe von Stigmergy realisiert.

Im zweiten Teil dieser Arbeit stellen wir zwei Clustering-Heuristiken vor, die den Netzwerkgraphen in verbundene Untergraphen (genannt Cluster) aufteilen. Jeder Cluster beinhaltet dann eine vollständige Instanz des Betriebssystems und der Anwendung. Durch die Aufteilung des Netzwerks in Cluster wird der Organisationsaufwand reduziert, da Protokolle, die auf globaler Information beruhen, auf ein einzelnes Cluster beschränkt sind. Dieses verbessert die Skalierbarkeit eines Systems.

Das Clustering-Problem wird Minimum-Intrakommunikations-Kosten-Clustering genannt. Die Idee dahinter ist, dass eine minimale Menge von Ressourcen in einem Cluster vorhanden sein muss und die Cluster wohl-verbunden sind. Die erste Heuristik ist für Netzwerke mit einem geringen Maß an Topologieveränderungen geeignet, die zweite dagegen für Netzwerke mit mäßig vielen Veränderungen. Beide Heuristiken basieren auf dem Prinzip der Arbeitsteilung sozialer Insekten.

Wir evaluieren unsere Heuristiken mit Hilfe des Shox-Simulators für drahtlose Netzwerke. Die Dienstverteilungsheuristik war in der Lage sehr gute Zuordnungen, nahe dem Optimum, für die meisten Experimente zu erzeugen. Unsere Clusteringheuristik für Systeme mit wenigen Topologieveränderungen übertrifft eine existierende Heuristik (Expanding Ring) im Hinblick auf die Kosten in den meisten Fällen. Sie war in der Lage Cluster zu erzeugen, die maximal im Durchschnitt 1,42 mal das Optimum in allen simulierten Szenarios waren. Ferner weisen die Ergebnisse eine geringe Standardabweichung auf.

Die Heuristiken können auf verschiedene Art und Weise erweitert werden. Um die Last, die dem Clusterhead auferlegt wird, besser zu verteilen, kann das Emergente Clustering um eine Clusterhead-Rotation erweitert werden. Darüber hinaus kann eine weitere Verhandlungsphase am Ende der Heuristik eingesetzt werden um ihre Leistung zu verbessern.

Über diese Punkte hinaus ist es unser Ziel das Konzept unseres Betriebssystems mit dem Kontrollscript-Mechanismus aus virtuellen Maschinen für WSNs zu kombinieren. Dieses wird eine einfache Entwicklung datenzentrischer Skripte ermöglichen, die die umfangreiche Funktionalität unserer verteilten Dienste nutzen um noch komplexere Aufgaben zu bewältigen.