



Patrick Dünser



Amon Grünbaum

Diplomanden	Patrick Dünser, Amon Grünbaum
Examinator	Prof. Oliver Augenstein
Experte	Dr. Peter Rost, NEC Labs Europe, Heidelberg D
Themengebiet	Kommunikationssysteme
Projektpartner	IBM Research GmbH, Rüschlikon ZH

Distributed Debugging for Wireless Sensor Networks

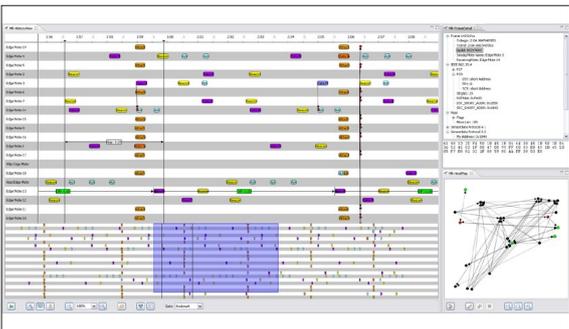
Debugging of Sensor Networks in Virtualized Environments



IBM Mote Runner



Sensor: MEMSIC Iris Mote



Übersicht Eclipse Plugin: History View, Frame Detail und Heat Map

Ausgangslage: Heutzutage sind immer mehr Produkte mit Sensoren ausgestattet, welche Umgebungsinformationen messen und nahtlos in das Informationsnetz integriert sind. Die einzelnen Sensoren können untereinander Daten austauschen und so Synergien nutzen. Dies wird als «Internet of Things» bezeichnet. Sensornetze sind jedoch kompliziert zu programmieren und zu testen, was den technologischen Fortschritt verlangsamt. IBM hat mit Mote Runner eine Plattform entwickelt, die es erlaubt, Wireless-Sensoren mittels einer Hochsprache (Java/C#) zu programmieren und zu debuggen. Mit der dazugehörigen Simulationsumgebung ist es bedeutend einfacher, Sensornetze zu testen. Um das Debugging komfortabler zu machen, war es unsere Aufgabe, die Kommunikation der verschiedenen Sensoren grafisch darzustellen.

Vorgehen/Technologien: Die Schwierigkeit beim Debuggen grosser Sensornetze besteht darin, versteckte Fehler wie Protokollverletzungen und Deadlock zu finden. Grund dafür ist, dass die Sensoren nebenläufig und zum Teil voneinander abhängig arbeiten. Unsere Visualisierung sollte deshalb die Radionachrichten zwischen den Sensoren in einem überschaubaren Raster darstellen. Um verschiedene Probleme der Sensornetze anzugehen, haben wir Werkzeuge entwickelt, die Abstände zwischen Nachrichten messen und Details der Netzwerkprotokolle anzeigen. Für die Realisierung eignete sich ein Eclipse Plugin, da bereits andere Teile von Mote Runner darauf aufsetzen. Des Weiteren musste eine Lösung gefunden werden, um auch eigene bzw. nicht integrierte Protokolle grafisch darzustellen.

Ergebnis: Entstanden ist ein Plugin, das aus mehreren Views besteht. Zum einen werden auf einer fortlaufenden Timeline die gesendeten Pakete der Sensoren visualisiert und nach Bedarf auch deren Empfänger aufgezeigt. Zum anderen werden nach dem Markieren eines Pakets zusätzliche Infos über den Inhalt des Pakets in einer separaten Ansicht aufgelistet. Zum dritten werden die Sensoren entsprechend ihrer räumlichen Position und ihrer Kommunikationspfade dargestellt. Die empfangenen Radionachrichten werden analysiert. Je nach Format der Radionachricht wird diese anders dargestellt. Der Parserbaum kann vom Benutzer konfiguriert werden, und falls noch kein Parser für sein Netzwerkprotokoll vorhanden ist, kann der Benutzer selber das API für seinen Protokollparser implementieren. Dieser kann in die Applikation eingehängt werden. Mittels eines Wizard wird der Einstieg zum Erstellen des eigenen Parsers erleichtert.