

Exposé zur Studienarbeit “Visualisierung der Energie- und Kommunikationsdaten einer auf ns-2 basierenden Simulation eines Sensornetzwerkes”

(Geplante Abgabe: 31. August 2007)

Christian Czekay

10. Mai 2007

1 Motivation

Sensornetzwerke haben viele Einsatzmöglichkeiten mit jeweils verschiedenen Anforderungsprofilen. Allen Einsatzszenarien ist aber die Anforderung der Energieeffizienz, der auf den Sensorknoten laufenden Algorithmen, gemein, da es in der Natur von verteilten Sensorknoten liegt, dass sie eine autonome und sehr begrenzte Energieversorgung besitzen. Betrachtet man beispielsweise als Szenario das Ausbringen von Sensorknoten als Frühwarn- und Ersatzkommunikationsmedium im Katastrophenfall (z.B. Erdbeben[11, 4, 2]), so ist klar, dass man sich zur Energieversorgung der Knoten nicht auf das normale Stromnetz verlassen kann, da es im Ernstfall meist zu stark beschädigt wird und nicht zur Verfügung steht. Und die Verwendung von Sensornetzen als Ersatz für die ausgefallene konventionelle Kommunikationsinfrastruktur macht natürlich nur Sinn, wenn garantiert werden kann, dass die Knoten auch wirklich zur Verfügung stehen und nicht nach kurzer Zeit wegen Energiemangel ausfallen. Schließlich macht es die verteilte Natur solcher Netzwerke schwierig bis unmöglich, die Energiequelle der einzelnen Knoten manuell zu ersetzen. Wenn also Kommunikations- und Routingprotokolle für Sensorknoten mittels Simulation erprobt und evaluiert werden, ist es unbedingt nötig, den Aspekt der Energieeffizienz in Betracht zu ziehen.

2 Zielstellung

Diesen Aspekt der Netzwerksimulation durch ein geeignetes grafisches Tool zu unterstützen, ist Ziel meiner Studienarbeit. Mein Werkzeug soll den Experimentator dabei unterstützen, eine Simulation zu initialisieren und ihren Verlauf zu verfolgen. Zentraler Aspekt dabei ist, die verbleibende Energie der Knoten und ihren Energieverbrauch, sowie die Kommunikationsaktivität

zwischen den Knoten zu visualisieren. Es soll ermöglicht werden, die Energieeffizienz des simulierten Protokolls aus der Visualisierung direkt abschätzen zu können. Es wird erwartet, dass die Bewertung von Kommunikationsprotokollen hinsichtlich ihrer Energieeffizienz dadurch stark vereinfacht werden kann und es leichter wird, Protokolle unter dem Energieeffizienzaspekt zu optimieren oder neu zu entwickeln.

3 Geplante Vorgehensweise

Als Simulator soll *ns-2* [8] dienen. Zum einen ist er einer der meist verbreiteten Simulatoren für die Netzwerkkommunikation. Und zum anderen bringt er bereits grundlegende Unterstützung für die Analyse des Energieverbrauchs eines Kommunikationsprotokolls mit. *ns-2* verfügt über ein ausgereiftes Energiemodell, welches es erlaubt, die Startenergie eines Knotens und den Energieverbrauch im Idle- und Sleepmodus, beim Empfangen und Senden von Paketen und beim Erfassen von Sensordaten zu spezifizieren.

Die Unterstützung des Experimentators bei der Initialisierung einer Simulation beinhaltet die Auswahl der Anzahl der Knoten und ihrer energierelevanten Parameter, sowie die Auswahl eines geeigneten von *ns-2* nativ unterstützten oder nachträglich implementierten Kommunikationsprotokolls. Nach Abschluß der Simulation werden dann die Tracedateien des *ns-2* Simulators ausgewertet und visualisiert. Um den Experimentator in die Lage zu versetzen, die Energieeffizienz des simulierten Protokolls aus der Visualisierung direkt abzuschätzen, könnte man beispielsweise den Energielevel eines Knotens in der Größe seines Symbols und den über einen wählbaren Zeitraum gemittelten Energieverbrauch durch seine Farbe darstellen.

Die genaue Umsetzung des grafischen Teils muss noch im Verlauf der Studienarbeit evaluiert werden. Denkbar wäre z.B. eine Browser-gestützte Lösung mittels eines Javaapplets. Die Simulationsinfrastruktur müsste dann nur einmal auf einem Server eingerichtet werden und der Experimentator könnte dann von überall darauf zugreifen. Alternativen wären die Implementation als Eclipseplugin oder als Stalaloneanwendung.

4 Verwandte Arbeiten

Die Visualisierung von Simulatorergebnissen ist keine neue Aufgabe und es gibt bereits diverse Tools für diese Aufgabe.

ns-2 selbst kommt bereits mit dem Visualisierungstool *nam*, dem *Network AniMator* [3]. *nam* erzeugt eine 2D Darstellung der Knotenpositionen und der Sende- und Empfangsevents. *nam* war ursprünglich dazu gedacht, Traces von drahtgebundenen Netzwerksimulationen von *ns-2* zu visualisieren, kann inzwischen aber auch das neuere Format von drahtlosen Traces darstellen. Allerdings sind die Visualisierungsfähigkeiten in diesem Bereich noch recht

begrenzt, insbesondere gibt es keine Möglichkeit energierelevante Werte zu visualisieren.

ad-hockey[9] wurde im Rahmen des Monarchprojekts[1] entwickelt, welches ns-2 um die Fähigkeiten zur Simulation von drahtlosen Netzwerken erweiterte. Allerdings ist das letzte Release von 1999 und enthält keinerlei Unterstützung für die Visualisierung von Energieparametern.

Ein weiteres ns-2 Visualisierungswerkzeug ist *iNSpect*[6]. Es konzentriert sich darauf, die Knotenpositionen und -bewegungen sowie Packetrouten darzustellen. Außerdem ist es möglich ein ns-2 *mobility-file*, welches die Knotenbewegungen im Simulationsverlauf definiert, von ns-2 unabhängig zu visualisieren, um zu verifizieren, dass es dem gewünschten Verhalten entspricht. Aber auch iNSpect visualisiert keine energierelevanten Werte.

Mit *Huginn*[10] gibt es ein weiteres ns-2 Visualisierungstool. Es visualisiert drahtlose ns-2 Traces in 3D. Der Nutzer kann über einen GUI-Dialog flexibel bestimmen, wie bestimmte Simulationsereignisse oder Statistikwerte visualisiert werden sollen. Außerdem kann man zu beliebigen Punkten in der Simulationszeit springen und es ist (im Gegensatz zu z.B. iNSpect) nicht nötig, das gesamte Tracefile vor der Visualisierung einzulesen, was Speicher und Zeit spart. Allerdings ist das Setup des Programms und der benötigten Bibliotheken überaus kompliziert und fehleranfällig und auch bei Huginn liegt der Fokus lediglich darauf, Position und Kommunikationsaktivität der Knoten darzustellen.

TinyViz ist ein Visualisierer für den *TinyOS*[5] Simulator *TOSSIM*[7]. Während *TOSSIM* selbst die Energie von Knoten nicht simuliert, gibt es jedoch eine Erweiterung namens *PowerTOSSIM*, die entsprechende Unterstützung enthält und ein *TinyViz* Plugin mitbringt, das den Energieverbrauch der Knoten darstellt. Aufgrund der *TOSSIM* inherenten Einschränkungen ist die Energiesimulation von *PowerTOSSIM* allerdings nie völlig exakt und sie funktioniert momentan lediglich für Mica2 Knoten.

Literatur

- [1] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.
- [2] SAFER Seismic eArly warning For EuRope. <http://www.saferproject.net/>.
- [3] Deborah Estrin, Mark Handley, John S. Heidemann, Steven McCanne, Ya Xu, and Haobo Yu. Network visualization with nam, the VINT network animator. *IEEE Computer*, 33(11):63–68, 2000.

-
- [4] Joachim Fischer. Selbstorganisation im Wettlauf mit tödlichen Wellen - Start des interdisziplinären Graduiertenkollegs METRIK. *Humboldt-Spektrum*, pages 32–37, March 2006.
 - [5] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David E. Culler, and Kristofer S. J. Pister. System architecture directions for networked sensors. In *ASPLOS*, pages 93–104, 2000.
 - [6] Stuart Kurkowski, Tracy Camp, Neil Muehl, and Michael Colagrosso. A visualization and analysis tool for NS-2 wireless simulations: iNSpect. In *MASCOTS*, pages 503–506. IEEE Computer Society, 2005.
 - [7] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, pages 126–137, New York, November 5–7 2003. ACM Press.
 - [8] The ns-2 network simulator. <http://www.isi.edu/nsnam/ns/>.
 - [9] CMU Monarch Project. The CMU Monarch project's ad-hockey visualization tool for ns scenario and trace files. *Carnegie Mellon University*, August 1998.
 - [10] Björn Scheuermann, Holger Füller, Matthias Transier, Marcel Busse, Martin Mauve, and Wolfgang Effelsberg. Huginn: A 3D Visualizer for Wireless ns-2 Traces. In *Proc. of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'05)*, pages 134–150, Motreal, Canada, October 2005.
 - [11] Metrik Modellbasierte Entwicklung von Technologien für selbstorganisierende dezentrale Informationssysteme im Katastrophenmanagement (Interdisziplinäres Graduiertenkolleg am Institut für Informatik der Humboldt-Universität zu Berlin). <http://casablanca.informatik.hu-berlin.de/grk-wiki/>.